ಕಂಪ್ಯೂಟರ್ ಆಪರೇಟರ್ ಮತ್ತು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಅಸಿಸ್ಟಂಟ್

(COMPUTER OPERATOR AND PROGRAMMING ASSISTANT)

NSQF ಹಂತ - 3

ವೃತ್ತಿ ಸಿದ್ಧಾಂತ
(TRADE THEORY)

ಕಾರ್ಯಕ್ಷೇತ್ರ : ಐಟಿ ಮತ್ತು ಐಟಿಇಎಸ್

Sector : IT&ITES

(As per revised syllabus July 2022 - 1200 hrs)

© NIMI NOT TO BE REPUBLISHED

Directorate General of Training

ಡೈರೆಕ್ಟರೇಟ್ ಜನರಲ್ ಆಫ್ ಟ್ರೈನಿಂಗ್

ಕೌಶಲ್ಯಾಭಿವೃದ್ಧಿ ಮತ್ತು ಉದ್ಯಮಶೀಲತೆ ಸಚಿವಾಲಯ

ಭಾರತ ಸರ್ಕಾರ

ನ್ಯಾಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್, ಚೆನ್ನೈ

ಅಂಚೆ ಪೆಟ್ಟಿಗೆ ಸಂಖ್ಯೆ 3142, CTI ಕ್ಯಾಂಪಸ್, ಗಿಂಡಿ, ಚೆನ್ನೈ - 600 032

ಕಾರ್ಯಕ್ಷೇತ್ರ : ಐಟಿ ಮತ್ತು ಐಟಿಇಎಸ್

ಅವಧಿ : 1 - ವರ್ಷ

ವೃತ್ತಿ : ಕಂಪ್ಯೂಟರ್ ಆಪರೇಟರ್ ಮತ್ತು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಅಸಿಸ್ಟಂಟ್ ವೃತ್ತಿ ಸಿದ್ಧಾಂತ ಎನ್‌ಎಸ್‌ಕ್ಯೂಎಫ್ ಹಂತ - 3 (ಪರಿಷ್ಕರಣೆ 2022)

ಪ್ರಕಾಶನ ಮತ್ತು ಮುದ್ರಣ :

ನ್ಯಾಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್

ಅಂಚೆ ಪೆಟ್ಟಿಗೆ ಸಂಖ್ಯೆ 3142,

ಗಿಂಡಿ, ಚೆನ್ನೈ - 600 032

E-mail : chennai-nimi@nic.in,

Website : www.nimi.gov.in

ಕಾಪಿರೈಟ್ © 2023 ನೇಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್, ಚೆನ್ನೈ.

ಪ್ರಥಮ ಮುದ್ರಣ : ಸೆಪ್ಟಂಬರ್, 2023 ಪ್ರತಿಗಳು : 500

ರೂ:

ಎಲ್ಲ ಹಕ್ಕುಗಳನ್ನು ಕಾಯ್ದಿರಿಸಲಾಗಿದೆ.

ಈ ಪ್ರಕಟಣೆಯ ಯಾವ ಭಾಗವನ್ನೂ ನ್ಯಾಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್ , ಚನ್ನೈ ಇವರ ಲಿಖಿತ ಅನುಮತಿ ಇಲ್ಲದೆ ಯಾವುದೇ ರೂಪದಲ್ಲಿ ಅಥವಾ ಇಲೆಕ್ಟ್ರಾನಿಕ್ , ಯಾಂತ್ರಿಕ ಯಾವುದೇ ರೀತಿಯಿಂದ ಛಾಯಾಪ್ರತಿ, ರೆಕಾರ್ಡಿಂಗ್ ಅಥವಾ ಯಾವುದೇ ಮಾಹಿತಿಯ ಸಂಗ್ರಹವನ್ನು ಪುನರುತ್ಪಾದನೆ ಮಾಡುವಂತಿಲ್ಲ ಅಥವಾ ಪ್ರಸಾರ ಮಾಡುವಂತಿಲ್ಲ.

© NIMI NOT TO BE REPUBLISHED

ಮುನ್ನುಡಿ

ಭಾರತ ಸರ್ಕಾರವು 2020 ರ ಹೊತ್ತಿಗೆ 30 ಕೋಟಿ ಜನರಿಗೆ, ಪ್ರತಿ ನಾಲ್ಕು ಭಾರತೀಯರಲ್ಲಿ ಒಬ್ಬರಿಗೆ, ರಾಷ್ಟ್ರೀಯ ಕೌಶಲ್ಯ ಅಭಿವೃದ್ಧಿ ನೀತಿಯ ಭಾಗವಾಗಿ ಉದ್ಯೋಗಗಳನ್ನು ಸುರಕ್ಷಿತವಾಗಿರಿಸಲು ಮಹತ್ವಾಕಾಂಕ್ಷೆಯ ಗುರಿಯನ್ನು ಹೊಂದಿಸಿ. ಕೈಗಾರಿಕಾ ತರಬೇತಿ ಸಂಸ್ಥೆ (ಐಟಿಐ) ಗಳು ಈ ಪ್ರಪಕ್ರಿಯೆಯಲ್ಲಿ ವಿಶೇಷವಾಗಿ ಸುರಿತ ಮಾನವಶಕ್ತಿಯನ್ನು ಒದಗಿಸುವ ದೃಷ್ಟಿಯಿಂದ ಪ್ರಮುಖ ಪಾತ್ರವಹಿಸುತ್ತವೆ. ಇದನ್ನು ಮನಸ್ಸಿನಲ್ಲಿಟ್ಟುಕೊಂಡು, ಪ್ರಸ್ತುತ ಉದ್ಯಮಕ್ಕೆ ತರಬೇತಿ ನೀಡುವ ಕೌಶಲ್ಯ ತರಬೇತಿಯನ್ನು ಒದಗಿಸಲು, ಐಟಿಐ ಪಠ್ಯಕ್ರಮವನ್ನು ಇತ್ತೀಚೆಗೆನವೀಕರಿಸಲಾಗಿದೆ, ಇದು ವಿವಿಧ ಪಾಲುದಾರರಾದ ಇಂಡಸ್ಟ್ರೀನ್, ಉದ್ಯಮಿಗಳು, ಅಕಾಡೆಮಿಶಿಯನ್ಗಳು ಮತ್ತು ಐಟಿಐಗಳು ಪ್ರತಿನಿಧಿಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ನ್ಯಾಷನಲ್ ಇನ್ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್ಸ್ಟಿಟ್ಯೂಟ್ (NIMI), ಚನ್ನೈ, ಇವರು ಈಗ ಐಟಿ ಮತ್ತು ಐಟಿಇಎಸ್ ಕಾರ್ಯಕ್ಷೇತ್ರದ **ಕಂಪ್ಯೂಟರ್ ಆಪರೇಟರ್ ಮತ್ತು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಅಸಿಸ್ಟಂಟ್** NSQF LEVEL - 3 (**ಪರಿಷ್ಕರಣೆ** 2022) ನೇಸೆಮಿಸ್ಟರ್ ವೃತ್ತಿ ಪ್ರಾಯೋಗಿಕ NSQF- 3 ವೃತ್ತಿ ಸಿದ್ಧಾಂತ ಇದು ತರಬೇತಾರ್ಥಿಗಳಿಗೆ ಅಂತರರಾಷ್ಟ್ರೀಯ ಸಮಾನವಾದ ಮಾನದಂಡವನ್ನು ಹೊಂದಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇಲ್ಲಿ ಜಗತ್ತಿನಾದ್ಯಂತ ಅವರ ಕೌಶಲ್ಯ ಪ್ರಾವೀಣ್ಯತೆ, ಮತ್ತು ಸಾಮರ್ಥ್ಯವನ್ನು ಗುರುತಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಇದು ಪೂರ್ವ ಕಲಿಕೆಯನ್ನು ಗುರುತಿಸುವ ವ್ಯಾಪ್ತಿಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ. ಎನ್ಎಸ್ಕ್ಯೂಎಫ್ ಹಂತ - 3 ರ ತರಬೇತಾರ್ಥಿಗಳು ಜೀವನದುದ್ದಕ್ಕೂ ಕಲಿಯುವಿಕೆ ಮತ್ತು ಕೌಶಲ್ಯ ಅಭಿವೃದ್ಧಿಯನ್ನು ಉತ್ತೇಜಿಸುವ ಅವಕಾಶಗಳನ್ನೂ ಸಹ ಪಡೆಯುತ್ತಾರೆ. ಎನ್ಎಸ್ಕ್ಯೂಎಫ್ ಹಂತ - 3 ನೊಂದಿಗೆ ಐಟಿಐನ ತರಬೇತುದಾರರು ಮತ್ತು ತರಬೇತಾರ್ಥಿಗಳು ಮತ್ತು ಎಲ್ಲ ಸಂಬಂಧಪಟ್ಟವರು ಈ ಐಎಂಪಿ (IMP) ಗಳಿಂದ ಗರಿಷ್ಟ ಪ್ರಯೋಜನಗಳನ್ನು ಪಡೆಯುತ್ತಾರೆ ಮತ್ತು ದೇಶದಲ್ಲಿ ವೃತ್ತಿಪರ ತರಬೇತಿಯ ಗುಣಮಟ್ಟವನ್ನು ಉತ್ತಮಗೊಳಿಸಲು ನಿಮಿ (NIMI) ಬಹು ಸುದೀರ್ಘವಾಗಿ ಪ್ರಯತ್ನಿಸುತ್ತದೆ ಎಂಬುದರಲ್ಲಿ ನನಗೆ ಯಾವುದೇ ಸಂಶಯವಿಲ್ಲ.

ಡೆರೆಕ್ಟರ್ ಜನರಲ್ ಆಫ್ ಟ್ರೈನಿಂಗ್ NIMIನ ಕಾರ್ಯನಿರ್ವಾಹಕ ನಿರ್ದೇಶಕರು ಮತ್ತು ಸಿಬ್ಬಂದಿ ಮತ್ತು ಮಾಧ್ಯಮ ಅಭಿವೃದ್ಧಿ ಸಮಿತಿಯ ಸದಸ್ಯರು ಈ ಪ್ರಕಟಣೆಯನ್ನು ಹೊರತರುವಲ್ಲಿ ಅವರ ಕೊಡುಗೆಗಾಗಿ ಮೆಚ್ಚುಗೆಗೆ ಅರ್ಹರು.

ಜೈ ಹಿಂದ್

ಅತುಲ್ ಕುಮಾರ್ ತಿವಾರಿ, *I.A.S.*

ಕಾರ್ಯದರ್ಶಿ

ಕೌಶಲ್ಯ ಅಭಿವೃದ್ಧಿ ಮತ್ತು ಉದ್ಯಮಶೀಲತೆ ಸಚೀವಾಲಯ

ಭಾರತ ಸರ್ಕಾರ.

ಸೆಪ್ಟಂಬರ್, 2023

ನವದೆಹಲಿ - 110001

© NIMI NOT TO BE REPUBLISHED

ಪೀಠಿಕೆ

ನ್ಯಾಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್ (NIMI) ಚನ್ನೈನಲ್ಲಿ 1986ರಲ್ಲಿ ಡೈರೆಕ್ಟರೇಟ್ ಜನರಲ್ ಆಫ್ ಎಂಪ್ಲಾಯ್‌ಮೆಂಟ್ ಆ್ಯಂಡ್ ಟ್ರೇನಿಂಗ್ (D.G.E & T) , ಕಾರ್ಮಿಕ ಮತ್ತು ಉದ್ಯೋಗ ಸಚಿವಾಲಯ, (ಈಗ ಡೈರೆಕ್ಟರೇಟ್ ಜನರಲ್ ಆಫ್ ಟ್ರೇನಿಂಗ್, ಕೌಶಲ್ಯ ಅಭಿವೃದ್ಧಿ ಮತ್ತು ಉದ್ಯಮಶೀಲತೆ ಸಚಿವಾಲಯ ಅಡಿಯಲ್ಲಿದೆ) ಭಾರತ ಸರ್ಕಾರ, ಫೆಡರಲ್ ರಿಪಬ್ಲಿಕ್ ಆಫ್ ಜರ್ಮನಿಯ ಸರ್ಕಾರದ ತಾಂತ್ರಿಕ ನೆರವಿನೊಂದಿಗೆ ಸ್ಥಾಪಿಸಲಾಯಿತು. ಈ ಸಂಸ್ಥೆಯ ಮುಖ್ಯ ಉದ್ದೇಶವೆಂದರೆ ಕುಶಲಕರ್ಮಿ(Craftsman) ಮತ್ತು ಶಿಶಿಕ್ಷು (Apprenticeship) ತರಬೇತಿ ಯೋಜನೆ ಅಡಿಯಲ್ಲಿ ಬರುವ ವಿವಿಧ ವೃತ್ತಿಗಳಿಗೆ ಅವುಗಳ ನಿಗದಿತ ಪಠ್ಯಕ್ರಮದ ಅನುಸಾರ ಸೂಚನಾ ಸಾಮಗ್ರಿಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು ಮತ್ತು ಒದಗಿಸುವುದು.

ಭಾರತದಲ್ಲಿ NCVT/NAC ಅಡಿಯಲ್ಲಿ ಬರುವ ವೃತ್ತಿಪರ ತರಬೇತಿಗಳ ಮುಖ್ಯ ಉದ್ದೇಶವನ್ನು ಗಮದಲ್ಲಿಟ್ಟುಕೊಂಡು ಸೂಚನಾ ಸಾಮಗ್ರಿಗಳನ್ನು ರಚಿಸಲಾಗಿದೆ. ಇದು ಒಬ್ಬ ವ್ಯಕ್ತಿಯು ಕೆಲಸ ಮಾಡಲು ಅವಶ್ಯವಿರುವ ಕೌಶಲ್ಯಗಳನ್ನು ಕರಗತ ಮಾಡಿಕೊಳ್ಳಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಸೂಚನಾ ಸಾಮಗ್ರಿಗಳನ್ನು ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಪ್ಯಾಕೇಜ್(IMPs) ಗಳ ರೂಪದಲ್ಲಿ ರಚಿಸಲಾಗಿದೆ. ಒಂದು IMP ಯು ಥಿಯರಿ ಪುಸ್ತಕ, ಪ್ರ್ಯಾಕ್ಟಿಕಲ್ ಪುಸ್ತಕ, ಟೆಸ್ಟ್ ಮತ್ತು ಅಸೈನ್‌ಮೆಂಟ್ ಪುಸ್ತಕ, ಇನ್‌ಸ್ಟ್ರಕ್ಟರ್ ಗೈಡ್, ಆಡಿಯೋ ವಿಜುವಲ್ ಏಡ್ (ವಾಲ್ ಚಾರ್ಟ್‌ಗಳು ಮತ್ತು ಟ್ರಾನ್ಸಪರನ್ಸಿಗಳು) ಮತ್ತು ಇತರ ಸಾಮಗ್ರಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ವೃತ್ತಿ ಪ್ರಯೋಗ ಪುಸ್ತಕವು ತರಬೇತಾರ್ಥಿಗಳು ಕಾರ್ಯಾಗಾರದಲ್ಲಿ ಪೂರ್ಣಗೊಳಿಸಬೇಕಾದ ಅಭ್ಯಾಸಗಳ ಸರಣಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ನಿಗದಿತ ಪಠ್ಯಕ್ರಮದಲ್ಲಿರುವಂತೆ ಎಲ್ಲ ಕೌಶಲ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿರುವದನ್ನು ಖಚಿತಪಡಿಸಲು ಈ ಅಭ್ಯಾಸಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ವೃತ್ತಿ ಸಿದ್ಧಾಂತ ಪುಸ್ತಕವು ತರಬೇತಾರ್ಥಿಗೆ ಕೆಲಸ ಮಾಡಲು ಅಗತ್ಯವಿರುವ ಸೈದ್ಧಾಂತಿಕ ಜ್ಞಾನವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಟೆಸ್ಟ್ ಮತ್ತು ಅಸೈನ್‌ಮೆಂಟ್ ಪುಸ್ತಕಗಳು ಬೋಧಕರಿಗೆ ತರಬೇತಾರ್ಥಿಗಳ ಕಾರ್ಯಕ್ಷಮತೆಯ ಮೌಲ್ಯಮಾಪನ ಮಾಡಲು ಅಸೈನ್‌ಮೆಂಟ್‌ಗಳನ್ನು ಕೊಡಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ. ವಾಲ್‌ಚಾರ್ಟ್‌ಗಳು ಮತ್ತು ಟ್ರಾನ್ಸ್‌ಪರನ್ಸಿಗಳು ವಿಶಿಷ್ಟವಾಗಿರುತ್ತವೆ. ಇವು ಬೋಧಕರಿಗೆ ಒಂದು ವಿಷಯವನ್ನು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಪ್ರಸ್ತುತಪಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ ಮತ್ತು ತರಬೇತಾರ್ಥಿಗಳಿಗೆ ಎಷ್ಟು ಅರ್ಥವಾಗಿದೆ ಎಂಬುದನ್ನು ನಿರ್ಣಯಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ. ಇನ್‌ಸ್ಟ್ರಕ್ಟರ್ ಗೈಡ್, ಬೋಧಕರಿಗೆ ಅವರ ಬೋಧನಾ ವೇಳಾಪಟ್ಟಿಯ ಯೋಜನೆಯನ್ನು ರೂಪಿಸಲು, ಕಚ್ಚಾ ವಸ್ತುಗಳ ಅವಶ್ಯಕತೆಗಳ ಯೋಜನೆಯನ್ನು ರೂಪಿಸಲು, ದಿನನಿತ್ಯದ ಪಾಠಗಳನ್ನು ಮತ್ತು ಪ್ರಾಯೋಗಿಕ ಪಾಠಗಳ ಯೋಜನೆ ರೂಪಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

IMP ಗಳು ಸಂಕೀರ್ಣ ಕೌಶಲ್ಯಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಪರಿಣಾಮಕಾರಿಯಾಗಿವೆ. ಪಠ್ಯಕ್ರಮದಲ್ಲಿ ಸೂಚಿಸಿದಂತೆ ಎಲ್ಲ ವೃತ್ತಿಗಳಲ್ಲಿ ಮಹತ್ವವಾದ ಕೌಶಲ್ಯಗಳನ್ನು ಅಳವಡಿಸುವಲ್ಲಿ ವಿಶೇಷವಾದ ಎಚ್ಚರಿಕೆಯನ್ನು ವಹಿಸಬೇಕಾಗುತ್ತದೆ.

ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿ ಸಂಪೂರ್ಣ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಪ್ಯಾಕೇಜ್(IMPs) ಲಭ್ಯವಿದ್ದರೆ ಪರಿಣಾಮಕಾರಿಯಾಗಿ ತರಬೇತಿ ನೀಡುವುದರಲ್ಲಿ ತರಬೇತಿದಾರರಿಗೆ ಮತ್ತು ಆಡಳಿತ ಮಂಡಳಿಗೆ ಸಹಕಾರಿಯಾಗಿವೆ.

IMP ಗಳು, NIMI ಯ ಸಿಬ್ಬಂದಿ ಸದಸ್ಯರು ಮತ್ತು ಮಾಧ್ಯಮ ಅಭಿವೃದ್ಧಿ ಸಮಿತಿಯ ಸದಸ್ಯರು ಇವರೆಲ್ಲರ ಸಾಮೂಹಿಕ ಪ್ರಯತ್ನದ ಫಲಿತಾಂಶವಾಗಿದೆ. ಮಾಧ್ಯಮ ಅಭಿವೃದ್ಧಿ ಸಮಿತಿಗಳ ಸದಸ್ಯರು ವಿಶೇಷವಾಗಿ ಸಾರ್ವಜನಿಕ ಮತ್ತು ಖಾಸಗಿ ವಲಯದ ಉದ್ಯಮಗಳು, ಡೈರೆಕ್ಟರೇಟ್ ಜನರಲ್ ಆಫ್ ಟ್ರೇನಿಂಗ (DGT) ಅಡಿಯಲ್ಲಿ ಬರುವ ವಿವಿಧ ತರಬೇತಿ ಸಂಸ್ಥೆಗಳು, ಸರ್ಕಾರಿ ಮತ್ತು ಖಾಸಗಿ ಐಟಿಐಗಳಿಂದ ಆಯ್ಕೆ ಮಾಡಲ್ಪಟ್ಟಿರುತ್ತಾರೆ.

ವಿವಿಧ ರಾಜ್ಯ ಸರ್ಕಾರಗಳ ಉದ್ಯೋಗ ಮತ್ತು ತರಬೇತಿ ಇಲಾಖೆಯ ನಿರ್ದೇಶಕರು, ಸಾರ್ವಜನಿಕ ಮತ್ತು ಖಾಸಗಿ ವಲಯದ ಕೈಗಾರಿಕೆಗಳ ತರಬೇತಿ ಇಲಾಖೆಗಳು, DGT ಮತ್ತು DGT ಕ್ಷೇತ್ರ ಸಂಸ್ಥೆಗಳ ಅಧಿಕಾರಿಗಳು, ಪ್ರೂಫ್ ರೀಡರ್‌ಗಳು, ವೈಯಕ್ತಿಕ ಮಾಧ್ಯಮ ಅಭಿವರ್ಧಕರು ಮತ್ತು ಸಂಯೋಜಕರು ಇವರೆಲ್ಲರಿಗೆ NIMI ಪ್ರಾಮಾಣಿಕ ಧನ್ಯವಾದಗಳನ್ನು ತಿಳಿಸಲು ಬಯಸುತ್ತದೆ. ಇವರೆಲ್ಲರ ಬೆಂಬಲವಿಲ್ಲದೆ NIMI ಈ ಸೂಚನಾ ಸಾಮಗ್ರಿಗಳನ್ನು ಹೊರಗೆ ತರಲು ಸಾಧ್ಯವಾಗುತ್ತಿರಲಿಲ್ಲ.

(ಎಕ್ಸಿಕ್ಯೂಟಿವ್ ಡೈರೆಕ್ಟರ್)

NIMI, ಚೆನ್ನೈ

NOT TO BE REPUBLISHED © NIMI

ಕೃತಜ್ಞತೆಗಳು

ಐಟಿ ಮತ್ತು ಐಟಿಇಎಸ್ ಕಾರ್ಯಕ್ಷೇತ್ರದ (Sector) **ಕಂಪ್ಯೂಟರ್ ಆಪರೇಟರ್ ಮತ್ತು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಅಸಿಸ್ಟಂಟ್ ವೃತ್ತಿಯ** ಈ ಸೂಚನಾ ಸಾಮಗ್ರಿಯನ್ನು (ವೃತ್ತಿ ಸಿದ್ಧಾಂತ - Trade Theory) ಹೊರತರುವಲ್ಲಿ ಈ ಕೆಳಗಿನ ಮಾಧ್ಯಮ ಅಭಿವರ್ಧಕರು ಮತ್ತು ಅವರ ಪ್ರಾಯೋಜಕ ಸಂಸ್ಥೆಗಳು ನೀಡಿದ ಸಹಕಾರ ಮತ್ತು ಕೊಡುಗೆಯನ್ನು ನ್ಯಾಷನಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನಲ್ ಮೀಡಿಯಾ ಇನ್‌ಸ್ಟಿಟ್ಯೂಟ್ (NIMI), ಪ್ರಾಮಾಣಿಕವಾಗಿ ಒಪ್ಪಿಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಧನ್ಯವಾದಗಳನ್ನು ಅರ್ಪಿಸುತ್ತದೆ.

ಮಾಧ್ಯಮ ಅಭಿವೃದ್ಧಿ ಸಮಿತಿಯ ಸದಸ್ಯರು

ಶ್ರೀಮತಿ ಎಂ. ಭಾನುಮತಿ	-	ಸಹಾಯಕ ತರಬೇತಿ ಅಧಿಕಾಂ ಸರ್ಕಾರಿ ಐ.ಟಿ.ಐ., ಅಂಬತ್ತುರ್
ಶ್ರೀಮತಿ ವಿ. ರೇವತಿ	-	ಸಹಾಯಕ ತರಬೇತಿ ಅಧಿಕಾರಿ ಸರ್ಕಾರಿ ಐ.ಟಿ.ಐ., ಕೊಯಂಬತೂರು

NIMI, ಸಂಯೋಜಕರು

ಶ್ರೀ ನೈರ್ಮಲ್ಯ ನಾಥ್	-	ಉಪ ನಿರ್ದೇಶಕರು NIMI, ಚನ್ನೈ- 32
ಶ್ರೀ ಜಿ. ಮೈಕಲ್ ಜಾನಿ	-	ವ್ಯವಸ್ಥಾಪಕರು, NIMI, ಚನ್ನೈ- 32
ಶ್ರೀ. ಎ. ಮುತ್ತುಕುಮಾರ್	-	JTA (DTP) NIMI, ಚನ್ನೈ- 32

ಈ ಸೂಚನಾ ಸಾಮಗ್ರಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ಡಾಟಾ ಎಂಟ್ರಿ, ಕ್ಯಾಡ್, ಡಿಟಿಪಿ ಆಪರೇಟರ್‌ಗಳ ಅತ್ಯುತ್ತಮ ಮತ್ತು ಶ್ರದ್ಧಾಭರಿತ ಸೇವೆಗಳಿಗೆ NIMI ತನ್ನ ಮೆಚ್ಚುಗೆಯನ್ನು ಸಲ್ಲಿಸುತ್ತದೆ.

ಈ ಸೂಚನಾ ಸಾಮಗ್ರಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ನಿಟ್ಟಿನಲ್ಲಿ ತಮ್ಮ ಅಮೂಲ್ಯ ಪ್ರಯತ್ನಗಳ ಕೊಡುಗೆ ನೀಡಿದ ಎಲ್ಲ ಇತರ NIMI ಸಿಬ್ಬಂದಿಗಳಿಗೆ ಸಹ NIMI ಧನ್ಯವಾದಗಳನ್ನು ಸಲ್ಲಿಸುತ್ತದೆ.

ಈ ಸೂಚನಾ ಸಾಮಗ್ರಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ನೇರವಾಗಿ ಅಥವಾ ಪರೋಕ್ಷವಾಗಿ ಸಹಾಯ ಮಾಡಿದ ಎಲ್ಲರಿಗೂ NIMI ಕೃತಜ್ಞತೆಗಳನ್ನು ಸಲ್ಲಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಪರಿಚಯ

ವೃತ್ತಿ ಪ್ರಯೋಗ (TRADE PRACTICAL)

ವೃತ್ತಿ ಪ್ರಯೋಗದ ಕೈಪಿಡಿಯು ಪ್ರಯೋಗಾಲಯದಲ್ಲಿ ಮಾಡುವ ಪ್ರಯೋಗಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ್ದಾಗಿರುತ್ತದೆ. ಇದು ಪ್ರಯೋಗಾತ್ಮಕ ಅಭ್ಯಾಸಗಳನ್ನೊಳಗೊಂಡಿರುತ್ತದೆ. COPA ವಿಭಾಗದ ತರಬೇತಾರ್ಥಿಗಳು ಸಹಾಯಕ ಸೂಚನೆಗಳು /ಮಾಹಿತಿಗಳೊಂದಿಗೆ ಈ ಪ್ರಯೋಗಗಳನ್ನು ಅಭ್ಯಾಸ ಮಾಡಬೇಕು. ಈ ಪ್ರಯೋಗಗಳು NSQF LEVEL - 3 ನಲ್ಲಿ ನೀಡಲಾದ ಕೌಶಲ್ಯಗಳನ್ನು ಹೊಂದಲು ಅನುಕೂಲವಾಗಿರುವಂತೆ ರಚಿಸಲ್ಪಟ್ಟಿವೆ.

ಮಾಡ್ಯೂಲ್ 1 - ಸುರಕ್ಷತೆಯಿಂದ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಅಭ್ಯಾಸಗಳು
ಮಾಡ್ಯೂಲ್ 2 - ಡೆಸ್ಕ್‌ಟಾಪ್ ಪಿಸಿಯನ್ನು ಅಸೆಂಬಲ್ ಮಾಡುವುದು
ಮಾಡ್ಯೂಲ್ 3 - ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಬಳಸುವುದು
ಮಾಡ್ಯೂಲ್ 4 - ಕಂಪ್ಯೂಟರ್ ಪ್ರಾಥಮಿಕ ತಿಳುವಳಿಕೆ ಮತ್ತು ಸಾಫ್ಟವೇರ್ ಇನ್ಸ್ಟಾಲೇಷನ್
ಮಾಡ್ಯೂಲ್ 5 - ಡಾಸ್ ಕಮಾಂಡ್ ಲೈನ್ ಇಂಟರ್‌ಫೇಸ್
ಮಾಡ್ಯೂಲ್ 6 - ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್‌ಮಾಡುವುದು ಮತ್ತು ಲಿನಕ್ಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 7 - ವರ್ಡ್ ಪ್ರೊಸೆಸಿಂಗ್ ಸಾಫ್ಟವೇರ್
ಮಾಡ್ಯೂಲ್ 8 - ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು
ಮಾಡ್ಯೂಲ್ 9 - ಟೇಬಲ್‌ಗಳನ್ನು ಮತ್ತು ಲಿಸ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 10 - ರೆಫರನ್ಸ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 11 - ಗ್ರಾಫಿಕ್ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 12 - ಡಾಕ್ಯುಮೆಂಟ್ ಕೊಲ್ಯಾಬರೇಷನ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 13 - ಮೇಲಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 14 - ಸ್ಪ್ರೆಡ್ ಶೀಟ್ ಅಪ್ಲಿಕೇಷನ್,ವರ್ಕ್‌ಶೀಟ್ ಮತ್ತು ವರ್ಕ್‌ಬುಕ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 15 - ಡಾಟಾ ಸೆಲ್‌ಗಳು ಮತ್ತು ರೇಂಜ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 16 - ಟೇಬಲ್ ಮತ್ತು ಟೇಬಲ್ ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 17 - ಫಂಕ್ಷನ್‌ಗಳು ಮತ್ತು ಫಾರ್ಮುಲಾಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 18 - ಮೇಲಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 19 - ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 20 - ಪವರಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್
ಮಾಡ್ಯೂಲ್ 21 - ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು
ಮಾಡ್ಯೂಲ್ 22 - ಟೇಬಲ್‌ಗಳನ್ನು ಮತ್ತು ಬುಲೆಟ್ ಹೊಂದಿದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 23 - ಗ್ರಾಫಿಕ್ಸ್ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 24 - ಆಡಿಯೊ ಮತ್ತು ವಿಡಿಯೊ ಅಂಶಗಳು
ಮಾಡ್ಯೂಲ್ 25 - ಟ್ರಾನ್ಸಿಷನ್ ಮತ್ತು ಎನಿಮೇಷನ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 26 - ಕೊಲ್ಯಾಬೋರೇಷನ್ನು ನಿರ್ವಹಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 27 - ಮೈ ಎಸ್ ಕ್ಯೂ ಎಲ್ ನ್ನು ಅಭ್ಯಸಿಸು
ಮಾಡ್ಯೂಲ್ 28 - ಕ್ವೆರಿಗಳನ್ನು ಅಭ್ಯಸಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 29 - ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಅಭ್ಯಸಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 30 - ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಸ್ಥಾಪಿಸುವುದು ಮತ್ತು ಸಂರಚಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 31 - ಎಚ್‌ಟಿಎಂ ಎಲ್ ಟ್ಯಾಗ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಸರಳ ಸ್ಥಿರವೆಬ್ ಪುಟಗಳನ್ನು ರಚಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 32 - ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ - ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು HTML ಪುಟಗಳ ಜೊತೆಗೆ ಸೇರಿ ಬರೆಯುವುದು
ಮಾಡ್ಯೂಲ್ 33 - ಎಕ್ಸೆಲ್ ಉಪಯೋಗಿಸಿ ಡಾಟಾ ವಿಜುವಲೈಸೇಷನ್ ಮತ್ತು ಅನಲೈಸಿಸ್ ಮಾಡುವುದು
ಮಾಡ್ಯೂಲ್ 34 - ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡಿ ಉತ್ಪನ್ನಗಳನ್ನುಮತ್ತು ಸೇವೆಗಳನ್ನು ಗುರುತಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 35 - ವೈರಸ್ ಸ್ಪೈವೇರ್ ಮತ್ತು ಇತರ ದುರುದ್ದೇಶಪೂರಿತ ಸಂಕೇತಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು, ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ರಕ್ಷಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 36 - ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್
ಮಾಡ್ಯೂಲ್ 37 - ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಜೀವನಚಕ್ರವನ್ನು ಮಾಡಿ ತೋರಿಸುವುದು
ಮಾಡ್ಯೂಲ್ 38 to 42 - ಇಲೆಕ್ಟಿವ್ ಮಾಡ್ಯೂಲ್ I : ಪೈಥಾನ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್
ಮಾಡ್ಯೂಲ್ 38 to 42 - ಇಲೆಕ್ಟಿವ್ ಮಾಡ್ಯೂಲ್ II : ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್

ಕೌಶಲ್ಯ ತರಬೇತಿಯು ಕಂಪ್ಯೂಟರ್ ಪ್ರಯೋಗಾಲಯದಲ್ಲಿ ಕೆಲವು ಪ್ರಾಯೋಗಿಕ ಯೋಜನೆಗಳನ್ನು (Practical Projects) ತಯಾರಿಸಲು ಅನುಕೂಲವಾಗುವಂತೆ ಸರಣಿಯಾಗಿ ಪ್ರಯೋಗಗಳ ಅಭ್ಯಾಸಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಕೆಲವು ಪ್ರಯೋಗಗಳು ಮಾತ್ರ ಇದಕ್ಕೆ ಹೊರತಾಗಿವೆ.

ಈ ಪ್ರಾಯೋಗಿಕ ಕೈಪಿಡಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಅತ್ಯಂತ ಪ್ರಾಮಾಣಿಕವಾದ ಪ್ರಯತ್ನವನ್ನು ಕೈಗೊಳ್ಳಲಾಗಿದೆ. ಪ್ರತಿಯೊಂದು ಪ್ರಾಯೋಗಿಕ ಅಭ್ಯಾಸವನ್ನು ಒಬ್ಬ ಸಾಮಾನ್ಯ ತರಬೇತಾರ್ಥಿಗೂ ಸುಲಭವಾಗಿ ತಿಳಿಯುವಂತೆ ಸಿದ್ಧಪಡಿಸಲಾಗಿದೆ. ಆದಾಗ್ಯೂ ಈ ಪ್ರಾಯೋಗಿಕ ಕೈಪಿಡಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಸಿಬ್ಬಂದಿವೃಂದವು ಮತ್ತಷ್ಟು ಸುಧಾರಣೆಗಳನ್ನು ಮಾಡುವ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿದೆ. NIMI ಸಂಸ್ಥೆಯು, ಈ ಪ್ರಾಯೋಗಿಕ ಕೈಪಿಡಿಗಾಗಿ ಅನುಭವಿ ತರಬೇತಿ ಅಧ್ಯಾಪಕರಿಂದ ಸಲಹೆ ಸೂಚನೆಗಳನ್ನು ಸ್ವೀಕರಿಸಲು ಸಿದ್ಧವಾಗಿದೆ.

ವೃತ್ತಿ ಸಿದ್ಧಾಂತ (TRADE THEORY)

ವೃತ್ತಿ ಸಿದ್ಧಾಂತದ ಕೈಪಿಡಿ(ಮ್ಯಾನ್ಯುಯಲ್) ಕೋಪಾ ವೃತ್ತಿಯ ಕೋರ್ಸ್‌ನ ಸೈದ್ಧಾಂತಿಕ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿದೆ. ಇದರಲ್ಲಿನ ವಿಷಯಗಳನ್ನು ವೃತ್ತಿ ಪ್ರಯೋಗಿಕ ಪಾಠಗಳ ಮ್ಯಾನ್ಯುಯಲ್‌ನಲ್ಲಿರುವ ಪ್ರಾಯೋಗಿಕ ಅಭ್ಯಾಸಗಳಿಗೆ ಅನುಗುಣವಾಗಿ ಅನುಕ್ರಮಗೊಳಿಸಲಾಗಿದೆ. ಪ್ರತಿಯೊಂದು ಅಭ್ಯಾಸದಲ್ಲಿ ಹೇಳಲಾದ ಕೌಶಲ್ಯವನ್ನು, ಸೈದ್ಧಾಂತಿಕ ಅಂಶದೊಂದಿಗೆ ಸಂಬಂಧಿಸಿ ವಿವರಿಸುವ ಪ್ರಯತ್ನವನ್ನು ಮಾಡಲಾಗಿದೆ. ಈ ಪರಸ್ಪರ ಸಂಬಂಧವು ತರಬೇತಾರ್ಥಿಗಳಿಗೆ ಕೌಶಲ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು, ಗ್ರಹಿಕೆಯ ಸಾಮರ್ಥ್ಯವನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿಕೊಳ್ಳಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ವೃತ್ತಿ ಸಿದ್ಧಾಂತ(Trade Theory) ವನ್ನು ವೃತ್ತಿ ಪ್ರಯೋಗಗಳಿಗೆ(Trade Practicals) ಅನುಕೂಲವಾಗುವಂತೆ ಕಲಿಸಲಾಗುತ್ತದೆ.

ವೃತ್ತಿ ಪ್ರಯೋಗ ಕೈಪಿಡಿ (Trade Practical) ಒಳಗೊಂಡಿರುವ ಪ್ರಯೋಗಗಳಿಗನುಸಾರವಾಗಿ ವೃತ್ತಿ ಸಿದ್ಧಾಂತ(Trade Theory) ಪುಸ್ತಕದಲ್ಲಿ ಪಾಠಗಳನ್ನು ನೀಡಲಾಗಿದೆ.

ಪ್ರಯೋಗಾಲಯದಲ್ಲಿ (Laboratory), ಪ್ರತಿಯೊಂದು ಪ್ರಯೋಗ ಮಾಡುವುದಕ್ಕೆ ಒಂದು ದಿನ ಮೊದಲು ಅದಕ್ಕೆ ಸಂಬಂಧಿಸಿದ ಕೌಶಲ್ಯಗಳನ್ನು ಕಲಿಸುವುದು / ಕಲಿಯುವುದು ಉತ್ತಮವಾಗಿರುತ್ತದೆ. ಆದ್ದರಿಂದ ವೃತ್ತಿ ಸಿದ್ಧಾಂತವನ್ನು ಪ್ರತಿಯೊಂದು ಪ್ರಯೋಗದ ಮಹತ್ವವಾದ ಭಾಗವೆಂದೇ ಪರಿಗಣಿಸಬೇಕು.ಈ ಪುಸ್ತಕವನ್ನು ಸ್ವಯಂ ಕಲಿಕೆಗಾಗಿ ಎಂದು ಭಾವಿಸದೆ , ತರಗತಿಗಳಲ್ಲಿ ನಡೆಯುವ ಪಾಠಗಳಿಗೆ ಪೂರಕವಾದ ಮಾಹಿತಿ ಎಂದು ತಿಳಿಯಬೇಕು.

© NIMI NOT TO BE REPUBLISHED

ಪರಿವಿಡಿ

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 1 : ಸುರಕ್ಷತೆಯಿಂದ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಅಭ್ಯಾಸಗಳು (Safe working practices)	
1.1.01 & 02	ವಿದ್ಯುತ್ ಸುರಕ್ಷತೆ (Electrical safety)	1
1.1.03 & 04	ಸುರಕ್ಷತಾ ಅಭ್ಯಾಸಗಳು, ಅಗ್ನಿಶಾಮಕ ಯಂತ್ರಗಳು (Safety practices, fire extinguishers)	2
	ಮಾಡ್ಯೂಲ್ 2 : ಡೆಸ್ಕ್‌ಟಾಪ್ ಪಿಸಿಯನ್ನು ಅಸೆಂಬಲ್ ಮಾಡುವುದು (Assemble a Desktop PC)	
1.2.05 & 06	ಕಂಪ್ಯೂಟರ್‌ಗಳ ಪರಿಚಯ (Introduction to computers)	8
	ಮಾಡ್ಯೂಲ್ 3 : ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಬಳಸುವುದು (Using Windows Operating System)	
1.3.07 -13	ಸಿಪಿಯು ರಚನೆ ಮತ್ತು ಮದರ್‌ಬೋರ್ಡ್ ಪರಿಚಯ (Introduction to CPU architecture and motherboard)	22
	ಮಾಡ್ಯೂಲ್ 4 : ಕಂಪ್ಯೂಟರ್ ಪ್ರಾಥಮಿಕ ತಿಳುವಳಿಕೆ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಇನ್‌ಸ್ಟಾಲೇಷನ್ (Computer basics and software installation)	
1.4.14	ಬಯಾಸ್ ಸೆಟಪ್ ವೀಕ್ಷಿಸುವುದು ಮತ್ತು ಅದನ್ನು ಬದಲಾಯಿಸುವುದು (View the BIOS settings and their modifications)	29
1.4.15	ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡುವುದು (Install Windows operating system)	40
1.4.16	ಹಾರ್ಡ್ ಡಿಸ್ಕ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು ಮತ್ತು ಪಾರ್ಟಿಷನ್ ರಚಿಸುವುದೆ (Format hard disk and create partition)	43
1.4.17	ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡುವಾಗ ಕಂಡುಬರುವ ಸಾಮಾನ್ಯ ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಸಮಸ್ಯೆಗಳನ್ನು ನಿವಾರಣೆ ಮಾಡುವುದು (Identify and rectify common hardware and software issues during OS installation)	48
1.4.18	ವಿಂಡೋಸ್‌ಗೆ ಆವಶ್ಯಕ ಅಪ್ಲಿಕೇಷನ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ್ನು ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡುವುದು ಉದಾ: ಆಫೀಸ್ ಪ್ಯಾಕೇಜ್, ಪಿಡಿಎಫ್. ರೀಡರ್, ಮೀಡಿಯಾ ಪ್ಲೇಯರ್ ಇತ್ಯಾದಿ. (Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc)	53
1.4.19	ಬ್ಲೂಟೂಥ್ ಮತ್ತು ವೈ- ಫೈ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಸಂರಚಿಸುವುದು (Configure Bluetooth and Wi-Fi settings)	55
1.4.20 & 21	ಡಿವಿಡಿಗಳು, ಸಿಡಿಗಳು ಮತ್ತು ಡಿವಿಡಿಗಳನ್ನು ಬರ್ನ್ ಮಾಡುವುದು (DVDs, CDs and burning DVDs)	58
	ಮಾಡ್ಯೂಲ್ 5 : ಡಾಸ್ ಕಮಾಂಡ್ ಲೈನ್ ಇಂಟರ್‌ಫೇಸ್ (DOS Command Line Interface)	
1.5.22	ಡೈರೆಕ್ಟರಿ ಪಟ್ಟಿ ಮಾಡಲು ಡಾಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದ್ (Use basic DOS commands for directory listing)	61
1.5.23	ಡಾಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಫೈಲ್ ಮತ್ತು ಫೋಲ್ಡರ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage files and folders using DOS commands)	65

© NIMI NOT TO BE REPUBLISHED

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 6 : ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು ಮತ್ತು ಲಿನಕ್ಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Install Ubuntu Linux operating system and execute basic Linux commands)	
1.6.24 & 25	ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಪರಿಚಯ (Introduction to Linux operating system)	71
1.6.26 - 29	ಕಮಾಂಡ್‌ಗಳು ಮತ್ತು ವಿವಿಧ ಎಡಿಟರ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Handling commands and various editors)	77
	ಮಾಡ್ಯೂಲ್ 7 : ವರ್ಡ್ ಪ್ರೊಸೆಸಿಂಗ್ ಸಾಫ್ಟ್‌ವೇರ್ (Using Word Processing Software)	
1.7.30 - 33	ಎಂಎಸ್ ವರ್ಡ್ 2010 ಸಿದ್ಧಾಂತ (MS WORD 2010 THEORY)	107
	ಮಾಡ್ಯೂಲ್ 8 : ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Format documents)	
1.8.34 - 36	ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ್ನು ಅಳವಡಿಸುವುದು , ಡಾಕ್ಯುಮೆಂಟ್ ಸೆಕ್ಷನ್‌ಗಳನ್ನು ತಯಾರಿಸುವುದು ಮತ್ತು ಸಂರಚಿಸುವುದು (Insert, format text and paragraphs, Create and configure document sections)	118
	ಮಾಡ್ಯೂಲ್ 9 : ಟೇಬಲ್‌ಗಳನ್ನು ಮತ್ತು ಲಿಸ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Tables and Lists)	
1.9.37 - 39	ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಮಾರ್ಪಡಿಸುವುದು (Create, modify tables)	120
	ಮಾಡ್ಯೂಲ್ 10 : ರೆಫರನ್ಸ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Create and Manage References)	
1.10.40 &41	ರೆಫರನ್ಸ್ ಅಂಶಗಳನ್ನು ಮತ್ತು ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Create and manage reference elements and tables)	123
	ಮಾಡ್ಯೂಲ್ 11 : ಗ್ರಾಫಿಕ್ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Graphic Elements)	
1.11.42- 45	ಇಲ್ಲಸ್ಟ್ರೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಗಳನ್ನು ಅಳವಡಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದೆ (Insert, format illustrations and text boxes)	125
	ಮಾಡ್ಯೂಲ್ 12 : ಡಾಕ್ಯುಮೆಂಟ್ ಕೊಲ್ಯಾಬರೇಷನ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Document Collaboration)	
1.12.46 &47	ಕಮೆಂಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು, ಟ್ರ್ಯಾಕ್ ಬದಲಾಯಿಸುವುದು, ಮತ್ತು ಮೇಲಿಂಗ್ (Manage comments change tracking and mailings)	127
	ಮಾಡ್ಯೂಲ್ 13 : ಮೇಲಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Mailings)	
1.13.48	ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು (Perform mail merge)	132
	ಮಾಡ್ಯೂಲ್ 14 : ಸ್ಪ್ರೆಡ್ ಶೀಟ್ ಅಪ್ಲಿಕೇಷನ್,ವರ್ಕ್‌ಶೀಟ್ ಮತ್ತು ವರ್ಕ್‌ಬುಕ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Spread Sheet Application, Manage Worksheets and Workbooks)	
1.14.49 - 54	ಎಂ ಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಫೈಲ್ ತೆರೆಯುವುದು (Open files in MS Excel)	137

© NIMI NOT TO BE REPUBLISHED

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 15 : ಡಾಟಾ ಸೆಲ್‌ಗಳು ಮತ್ತು ರೇಂಜ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Data Cells and Ranges)	
1.15.55 -57	ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವುದು (Manipulate data)	139
	ಮಾಡ್ಯೂಲ್ 16 : ಟೇಬಲ್ ಮತ್ತು ಟೇಬಲ್ ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Tables and Table Data)	
1.16.58 -60	ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Create and format tables)	141
	ಮಾಡ್ಯೂಲ್ 17 : ಫಂಕ್ಷನ್‌ಗಳು ಮತ್ತು ಫಾರ್ಮುಲಾಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು (Perform Operations using Formulas and Functions)	
1.17.61-63	ಎಂ ಎಸ್. ಎಕ್ಸೆಲ್ 2010 ರಲ್ಲಿ ಫಂಕ್ಷನ್‌ಗಳು ಮತ್ತು ಫಾರ್ಮುಲಾಗಳು (Functions and formulas in MS-Excel 2010)	143
	ಮಾಡ್ಯೂಲ್ 18 : ಮೇಲಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Mailings)	
1.18.64-66	ಚಾರ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Charts)	150
	ಮಾಡ್ಯೂಲ್ 19 : ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Pivot Tables)	
1.19.67	ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು (Create Pivot Tables)	152
	ಮಾಡ್ಯೂಲ್ 20 : ಪವರಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್ (Power Point Presentations)	
1.20.68-74	ಎಂ. ಎಸ್ ಪವರಪಾಯಿಂಟ್‌ನಲ್ಲಿ ಫೈಲ್‌ಗಳನ್ನು ತೆರೆಯುವುದು (Open files in MS PowerPoint Presentations)	153
	ಮಾಡ್ಯೂಲ್ 21 : ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Format Presentations)	
1.21.75-77	ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳನ್ನು ಅಳವಡಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (IInsert, Format text and paragraphs)	155
	ಮಾಡ್ಯೂಲ್ 22 : ಟೇಬಲ್‌ಗಳನ್ನು ಮತ್ತು ಬುಲೆಟ್ ಹೊಂದಿದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Tables and Bulleted Text)	
1.22.78-80	ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡುವುದು. ಬುಲೆಟ್ ಹೊಂದಿದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುವುದು (Create tables, modify tables, modify bulleted text)	156
	ಮಾಡ್ಯೂಲ್ 23 : ಗ್ರಾಫಿಕ್ಸ್ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Graphic Elements)	
1.23.81-83	ಇಲ್ಲಸ್ಟ್ರೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ಗಳನ್ನು ಅಳವಡಿಸುವುದು, ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Insert illustrations, Format illustrations and text boxes)	157
	ಮಾಡ್ಯೂಲ್ 24 : ಆಡಿಯೊ ಮತ್ತು ವಿಡಿಯೊ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Audio & Video Elements)	
1.24.84&85	ಆಡಿಯೊ ಮತ್ತು ವಿಡಿಯೊ ಅಂಶಗಳು (Audio & Video Elements)	160

© NIMI NOT TO BE REPUBLISHED

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 25 : ಟ್ರಾನ್ಸಿಷನ್ ಮತ್ತು ಎನಿಮೇಷನ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Transitions and Animations)	
1.25.86&87	ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್ ಮತ್ತು ಎನಿಮೇಷನ್‌ಗಳು (Add slide transitions & animations)	162
	ಮಾಡ್ಯೂಲ್ 26 : ಕೊಲ್ಯಾಬೊರೇಷನ್‌ನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Collaboration)	
1.26.88	ಕಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Add and manage comments)	164
	ಮಾಡ್ಯೂಲ್ 27 : ಮೈ ಎಸ್ ಕ್ಯೂ ಎಲ್ ನ್ನು ಅಭ್ಯಸಿಸು (Demonstrate on MySQL)	
1.27.89-91	ಮೈ ಎಸ್ ಕ್ಯೂ ಎಲ್ ನ್ನು ಇನ್ಸ್ಟಾಲ್ ,ಟ್ರಬಲ್‌ಶೂಟ್ ಮಾಡುವುದು, ಡಾಟಾಬೇಸನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಉಪಯೋಗಿಸುವುದು (Install, Troubleshoot, Create and Use of database in MySQL)	165
1.27.92-96	ನಾರ್ಮಲೈಸೇಷನ್ ನಿಯಮಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಡಾಟಾಬೇಸನ್ನು ತಯಾರಿಸುವುದು, ವಿವಿಧ ಡಾಟಾಟೈಪ್‌ಗಳು, ಡಾಟಾ ಇಂಟಿಗ್ರಿಟಿ, ಡಿಡಿಎಲ್, ಡಿಎಮ್‌ಎಲ್ ಮತ್ತು ಡಿಸಿಎಲ್ ಸ್ಟೇಟ್‌ಮೆಂಟ್‌ಗಳು, ಪ್ರೈಮರಿ ಕೀ ಮತ್ತು ಫಾರಿನ್ ಕೀಗಳು (Designing database using normalization rules,various datatypes,data integrity,DDL,DML&DCL Statements Enforcing Primary key and Foreign key)	168
	ಮಾಡ್ಯೂಲ್ 28 : ಕ್ವೆರಿಗಳನ್ನು ಅಭ್ಯಸಿಸುವುದು (Demonstrate on Queries)	
1.28.97&98	ಕ್ವೆರಿಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Insert and delete queries Update queries)	171
	ಮಾಡ್ಯೂಲ್ 29 : ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಅಭ್ಯಸಿಸುವುದು (Demonstrate on Functions)	
1.29.99&100	ನಂಬರ್ ಡೇಟ್ ಮತ್ತು ಕ್ಯಾರೆಕ್ಟರ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು. ಗ್ರೂಪ್ ಬೈ, ಹ್ಯಾವಿಂಗ್ ಮತ್ತು ಸಬ್ ಕ್ವೆರಿಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Using the Number, Date and Character functions, group by having, sub query)	172
	ಮಾಡ್ಯೂಲ್ 30 : ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಸ್ಥಾಪಿಸುವುದು ಮತ್ತು ಸಂರಚಿಸುವುದು (Set-up & Configure a Computer Network)	
1.30.101-112	ಕಂಪ್ಯೂಟರ್‌ನ್ನು ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಜೋಡಿಸುವುದು ಮತ್ತು ಪ್ರಿಂಟರ್‌ಗಳು, ಫೈಲ್‌ಗಳು, ಮತ್ತು ಡ್ರೈವ್‌ಗಳಂತಹ ಸಾಧನಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳುವುದು (Connect a computer to a network and share Devices i.e. Printers, files, folders and drives)	175
	ಮಾಡ್ಯೂಲ್ 31 : ಎಚ್‌ಟಿಎಂ ಎಲ್ ಟ್ಯಾಗ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಸರಳ ಸ್ಥಿರ ವೆಬ್ ಪುಟಗಳನ್ನು ರಚಿಸುವುದು (Create Simple Static Web Pages using HTML Tags)	
1.31.113	ವೆಬ್ ಡಿಸೈನಿಂಗ್ (Web designing)	189
1.31.114	ಸಿಎಂಎಸ್ ಮತ್ತು ವೆಬ್ ಆಥರಿಂಗ್ ಸಾಧನಗಳ ಪರಿಚಯ (Introduction to CMS and web authoring tools)	215

© NIMI NOT TO BE REPUBLISHED

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 32: ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ - ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು HTML ಪುಟಗಳ ಜೊತೆಗೆ ಸೇರಿ ಬರೆಯುವುದು (JavaScript Embed JavaScript in HTML Pages)	
1.32.115	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ತಿಳಿದುಕೊಳ್ಳುವುದು (Understanding JavaScripts)	218
1.32.116&117	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್ ಮತ್ತು ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Using JavaScript Variable and data types)	224
1.32.118	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟ್‌ಮೆಂಟ್‌ಗಳು, ಲೂಪ್‌ಗಳು ಮತ್ತು ಪಾಪ್‌ಅಪ್ ಬಾಕ್ಸ್‌ಗಳು (Control statements, Loops and Popup boxes in JavaScript)	234
1.32.119	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಅರೇಗಳು (Arrays in JavaScript)	242
1.32.120	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಡೈನಾಮಿಕ್ HTML ಪುಟಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸುವುದು (Develop dynamic HTML pages using JavaScript)	252
1.32.121	ಐಐಎಸ್ ಬಳಸಿ ವೆಬ್ ಪ್ರಾಜೆಕ್ಟ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು (Deploy web project using IIS)	274
	ಮಾಡ್ಯೂಲ್ 33 : ಎಕ್ಸೆಲ್ ಉಪಯೋಗಿಸಿ ಡಾಟಾ ವಿಜುವಲೈಸೇಷನ್ ಮತ್ತು ಅನಲೈಸಿಸ್ ಮಾಡುವುದು (Data Visualization or Analysis using Excel)	
1.33.122-129	ಅಧುನಿಕ ಫಾರ್ಮ್ಯುಲಾಗಳು ಮತ್ತು ಮ್ಯಾಕ್ರೊಗಳನ್ನು ರಚಿಸುವುದು (Create advanced formulas and macros)	283
	ಮಾಡ್ಯೂಲ್ 34 : ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡಿ ಉತ್ಪನ್ನಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು ಗುರುತಿಸುವುದು (Browse E-Commerce Sites to Identify Products & Services)	
1.34.130-132	ಇ-ಕಾಮರ್ಸ್‌ನ ವ್ಯಾಪ್ತಿ ಮತ್ತು ಲಾಭಗಳು (E-Commerce scope and benefits)	293
1.34.133	ಇ-ಕಾಮರ್ಸ್‌ನ ಸೈಟ್‌ನಲ್ಲಿ ವ್ಯವಹಾರಗಳನ್ನು ಮಾಡುವುದು (Undertake transactions on an e-commerce site)	296
1.34.134-137	ಇ-ಕಾಮರ್ಸ್‌ನ ಸುರಕ್ಷತಾ ಸಮಸ್ಯೆಗಳು ಮತ್ತು ಹಣ ಪಾವತಿಸುವುದು (E-Commerce Security issues and Payment Gateways)	298
	ಮಾಡ್ಯೂಲ್ 35 : ವೈರಸ್ ಸ್ಪೈವೇರ್ ಮತ್ತು ಇತರ ದುರುದ್ದೇಶಪೂರಿತ ಸಂಕೇತಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು, ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ರಕ್ಷಿಸುವುದು (Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code)	
1.35.138	ಮಾಹಿತಿಯ ಸುರಕ್ಷತೆ ಮತ್ತು ಹೆದರಿಕೆಯ ಅವಲೋಕನ (Overview of information security and threats)	301
1.35.139	ಗೌಪ್ಯತೆಯ ರಕ್ಷಣೆ ಮತ್ತು IT Act (Privacy Protection and IT Act)	348
	ಮಾಡ್ಯೂಲ್ 36 : ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ (Cloud Computing)	
1.36.140-143	ಕ್ಲೌಡ್ ಸೇವೆಗಳೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು (Working with Cloud Services)	356

NOT TO BE REPUBLISHED © NIMI

ಪಾಠದ ಸಂಖ್ಯೆ	ಪಾಠದ ಶೀರ್ಷಿಕೆ	ಪುಟ ಸಂಖ್ಯೆ
	ಮಾಡ್ಯೂಲ್ 37 : ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಜೀವನಚಕ್ರವನ್ನು ಮಾಡಿ ತೋರಿಸುವುದು (Develop an application and perform the Application Development Life Cycle)	
1.37.144&145	ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ ಹಂತಗಳನ್ನು ಗುರುತಿಸುವುದು (Identify Phases of the Application Development Life Cycle)	363
	ಮಾಡ್ಯೂಲ್ 38 ರಿಂದ 42: ಇಲೆಕ್ಟಿವ್ ಮಾಡ್ಯೂಲ್ I : ಪೈಥಾನ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ (Elective Module I - Programming in Python)	
1.38.01&02	ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಲ್ಯಾಂಗ್ವೇಜ್ (ಪೈಥಾನ್) ಕಮಾಂಡ್ ಲೈನ್ ಪೈಥಾನ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು (Programming language (Python) Use Python from command line)	369
1.39.03-05	ಡಾಟಾ ಟೈಪ್‌ಗಳು ಮತ್ತು ಆಪರೇಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು (Perform Operations using Data Types and Operators)	372
1.40.06&07	ಡಿಸಿಷನ್ ಮತ್ತು ಲೂಪಿಂಗ್ ಉಪಯೋಗಿಸಿಕೊಂಡು ಫ್ಲೋ ನಿಯಂತ್ರಿಸುವುದು (Control Flow with Decisions and Loops)	377
1.41.08&09	ಡಾಕ್ಯುಮೆಂಟ್ ಮತ್ತು ಸ್ಟ್ರಕ್ಚರ್ ಕೋಡ್ (Document and Structure Code)	381
1.42.10&11	ಮಾಡ್ಯೂಲ್ ಮತ್ತು ಟೂಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು (Perform Operations Using Modules and Tools)	403
	ಮಾಡ್ಯೂಲ್ 38 ರಿಂದ 42: ಇಲೆಕ್ಟಿವ್ ಮಾಡ್ಯೂಲ್ II : ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ (Elective Module II Programming in JAVA)	
1.38.01-03	ಅಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮತ್ತು ಜಾವಾ ಲ್ಯಾಂಗ್ವೇಜ್ (Object Oriented Programming and JAVA Language)	408
1.39.04-07	ಜಾವಾ ಪ್ರೊಗ್ರಾಂ ಬರೆಯುವುದನ್ನು ತೋರಿಸುವುದು (Demonstrate writing JAVA programs)	414
1.40.08-16	ಜಾವಾ ಪ್ರೊಗ್ರಾಂನ ಫ್ಲೋ ಕಂಟ್ರೋಲ್ (JAVA Program Flow Control)	432
1.41.17-23	ಜಾವಾದಲ್ಲಿ ಕ್ಲಾಸ್, ಓವರ್‌ಲೋಡಿಂಗ್ ಮತ್ತು ಇನ್‌ಹೆರಿಟನ್ಸ್ (JAVA Classes, Overloading and Inheritance)	439
1.42.24-30	ಜಾವಾದಲ್ಲಿ ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಕ್ಲಾಸ್‌ಗಳು ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ಗಳು (Abstract Classes and Interfaces in JAVA)	449

© NIMI NOT TO BE REPUBLISHED

LEARNING / ASSESSABLE OUTCOME

On completion of this book you shall be able to

S.No.	Learning Outcome	Ref. Ex.No.
1	Install and setup operating system and related software in a computer following safety precautions. (Mapped NOS: SSC/N3022)	1.1.01 - 1.6.29
2	Create, format, and edit document using word processing application software.(Mapped NOS: SSC/N3022)	1.7.30 - 1.13.48
3	Create, format, edit and develop a workbook by using spreadsheet application software. (Mapped NOS: SSC/N3022)	1.14.49 - 1.19.67
4	Create and customize slides for presentation. (Mapped NOS: SSC/N3022)	1.20.68 - 1.26.88
5	Create and manage database file by using MySQL. (NOS: SSC/N9401)	1.27.89 - 1.29.100
6	Install, setup/configure, troubleshoot, and secure computer network including Internet. (Mapped NOS: SSC/N3022)	1.30.101 - 1.30.112
7	Develop web pages using HTML and CSS. (Mapped NOS: SSC/N0503, SSC/N0501)	1.31.113 - 1.31.114
8	Develop web pages using JavaScript. (Mapped NOS: SSC/N0503, SSC/N0501)	1.32.115 - 1.32.121
9	Create workbooks with advanced formulas, macros, charts, pivot tables and demonstrate ability to use Power tools. (NOS: SSC/N9402)	1.33.122 - 1.33.129
10	Browse, select, and transact using E- commerce websites (NOS: SSC/N9403)	1.34.130 - 1.34.137
11	Secure information from Internet by using cyber security concept. (NOS: SSC/N9404)	1.35.138 - 1.35.139
12	Explain Cloud concepts &services and Describe Application Development Life Cycle. (NOS: SSC/N9405)	1.36.140 - 1.37.145
13	Write programs using Python language. (NOS: SSC/N9406)	1.38.01 - 1.42.11
14	Writing programs using JAVA. (SSC/N9407)	1.38.01 - 1.42.30

© NIMI NOT TO BE REPUBLISHED

SYLLABUS

Duration	Reference Learning Outcome	Professional Skills (Trade Practical) with Indicative hours	Professional Knowledge (Trade Theory)
Professional Skill- 94 Hrs; Professional Knowledge - 32 Hrs	Install and setup operating system and related software in a computer following safety precautions. (Mapped NOS: SSC/N3022)	**Safe working practices (10 Hrs)** 1. Visit COPA Lab. of the institutes and locate the electrical connections with computer system setup. (3 Hrs) 2. Identifying safety symbols and hazard identification. (3 Hrs) 3. Practice safe methods of fire fighting in case of electrical fire. (2 Hrs) 4. Use of fire extinguishers. (2Hrs) **Assemble a Desktop PC (8 hrs)** 5. Identify computer peripherals and internal components of a desktop computer. (4 Hrs) 6. Assemble components of desktop computer. (4 Hrs) **Using Windows Operating Systems (20 hrs)** 7. Practice on Windows interface and navigating windows. (3 Hrs)	**Introduction to Computers (3 Hrs)** • Safe working practices • Scope of the COPA trade. • Safety rules and safety signs. • Types and working of fire extinguishers. Introduction to Computer components **Introduction to computer system (4 Hrs)** • Concepts of Hardware and Software. • Function of mother board components and various processors. • Various Input/ Output devices in use and their features Introduction Windows Operating System (9 Hrs)

		8. Practice on managing files and folders using removable drives. (4 Hrs) 9. Customize the desktop (2 hrs) 10. Settings and manage user accounts. (1 Hr) 11. View system properties and control panel details. (3 Hrs) 12. Work with keyboard shortcut commands. (4 Hrs) 13. Print and scan document using different commands. (3 Hrs) **Computer basics and Software Installation (20 Hrs)** 14. View the BIOS settings and their modifications. (3 Hrs) 15. Install Windows operating system. (4 Hrs) 16. Format hard disk and create partition. (3 Hrs) 17. Identify and rectify common hardware and software issues during OS installation. (3 Hrs) 18. Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc. (2 Hrs) 19. Configure Bluetooth and Wi-Fi settings. (1 Hr) 20. Install Drivers for printer, scanner, webcam and DVD etc. (2 Hrs) 21. Burn data, video and audio files on CD/DVD using application software. (2 Hrs) **DOS Command Line Interface (9Hrs)** 22. Use basic DOS commands for directory listing. (5 Hrs) 23. Manage files and folders using DOS commands. (4 Hrs) **Install Ubuntu Linux operating system and execute basic Linux commands (27 Hrs)** 24. Installation of Ubuntu Linux operating system (6 Hrs) 25. Install necessary application software for Linux i.e. Office Package, PDF Reader, Media Player etc. (4 Hrs) 26. Use Basic Linux commands for directory listing, file and folder management, password etc. (6 Hrs) 27. Use the Linux graphical user interface for file and folder management, exploring the system etc. (6 Hrs) 28. Customize desktop settings and manage user accounts in Linux. (3 Hrs)	• Introduction to operating System • Main features of Windows OS • Concept of various shortcut commands. **Introduction to the booting process (6 Hrs)** • Introduction to various types of memories and their features. • Basic Hardware and software issues and their solutions. • Usage of Application software and Antivirus. **Introduction to DOS Command Line Interface & Linux Operating Systems (10 Hrs)** • Introduction to basic DOS Internal and External Commands. • Introduction to Open Source Software • Introduction to Linux Operating System features, structure, files and processes • Basic Linux commands.

© NIMI NOT TO BE REPUBLISHED

		29. View system properties and manage system setting in Linux. (2 Hrs)	
Professional Skill - 47 Hrs.; Professional Knowledge - 14 Hrs	Create, format, and edit document using word processing application software. (Mapped NOS: SSC/N3022)	**Using Word Processing Software (47 hrs)** **Manage documents (11 Hrs.)** 30. Navigate within documents (2 Hrs) • Search for text • Link to locations within documents • Move to specific locations and objects in documents • Show and hide formatting symbols and hidden text **31. Format documents (4.6 Hrs)** • Set up document pages • Apply style sets • Insert and modify headers and footers • Configure page background elements **32. Save and share documents (2 Hrs)** • Save documents in alternative file formats • Modify basic document properties • Modify print settings • Share documents electronically **33. Inspect documents for issues (2.4 Hrs)** • Locate and remove hidden properties and personal information • Locate and correct accessibility issues • Locate and correct compatibility issues **Format documents (8 Hrs.)** 34. Insert text and paragraphs (2 Hrs) • Find and replace text • Insert symbols and special characters **35. Format text and paragraphs (3 Hrs)** • Apply text effects • Apply formatting by using Format Painter • Set line and paragraph spacing and indentation • Apply built-in styles to text • Clear formatting **36. Create and configure document sections (3 Hrs)** • Format text in multiple columns • Insert page, section, and column breaks • Change page setup options for a section **Manage tables and lists (9.5 Hrs)** **37. Create tables (3 Hrs)** • Convert text to tables • Convert tables to text	**Using Word Processing Software (14 Hrs)** • Introduction to the various applications in MS office. • Introduction to Word features, Office button, toolbars. • Creating, saving and formatting and printing documents using Word. • Working with objects, macro, mail merge, templates and other tools in Word.

NOT TO BE REPUBLISHED © NIMI

		• Create tables by specifying rows and columns **38. Modify tables (3 Hrs)** • Sort table data • Configure cell margins and spacing • Merge and split cells • Resize tables, rows, and columns • Split tables • Configure a repeating row header **39. Create and modify lists (3.5 Hrs)** • Format paragraphs as numbered and bulleted lists • Change bullet characters and number formats • Define custom bullet characters and number formats • Increase and decrease list levels • Restart and continue list numbering • Set starting number values Create and manage references (3 Hrs.) **40. Create and manage reference elements (1.4 Hrs)** • Insert footnotes and endnotes • Modify footnote and endnote properties • Create and modify bibliography citation sources • Insert citations for bibliographies **41. Create and manage reference tables (1.6 Hrs)** • Insert tables of contents • Customize tables of contents • Insert bibliographies **Manage graphic elements (8.5 Hrs.)** **42. Insert illustrations and text boxes (3 Hrs)** • Insert shapes • Insert pictures • Insert 3D models • Insert Smart Art graphics • Insert screenshots and screen clippings • Insert text boxes **43. Format illustrations and text boxes (3 Hrs)** • Apply artistic effects • Apply picture effects and picture styles • Remove picture backgrounds • Format graphic elements • Format SmartArt graphics • Format 3D models **44. Add text to graphic elements (1 Hr)** • Add and modify text in text boxes • Add and modify text in shapes " Add and modify SmartArt graphic content	

NOT TO BE REPUBLISHED © NIMI

		45. Modify graphic elements (1.5 Hrs) • Position objects • Wrap text around objects • Add alternative text to objects for accessibility **Manage document collaboration (3.5 Hrs.)** **46. Add and manage comments (1 Hrs)** • Add comments • Review and reply to comments • Resolve comments • Delete comments **47. Manage change tracking (2.5 Hrs)** • Track changes • Review tracked changes • Accept and reject tracked changes • Lock and unlock change tracking **Manage Mailings (3.5 Hrs)** **48. Perform mail merge (3.5 Hrs)** • Create envelopes • Create labels • Create a new mailing list • Perform mail merge using an existing list	
Professional Skill - 72 Hrs.; Professional Knowledge - 18 Hrs	Create, format, edit and develop a workbook by using spreadsheet application software. (Mapped NOS: SSC/N3022)	**Spread Sheet Application (72 Hrs)** **Manage Worksheets and Workbooks (12 Hrs.)** **49. Open files in MS Excel (1.5 Hrs)** • Open MS Excel • Create a new Excel file • Create a new Excel file from a template • Open an existing Excel file 50. Import data (1.5 Hrs) • Import data from txt files • Import data from csv files **51. Navigate within workbooks (2 Hrs)** • Search data • Navigate to named cells, ranges or workbook elements • Insert and remove hyperlinks **52. Format worksheets and workbooks (2 Hrs)** • Modify page setup • Adjust row height and column width • Customize headers and footers **53. Customize options and views (3 Hrs)** • Display and modify content in different views • Freeze rows and columns • Change window views • Modify basic workbook properties • Display formulas	**Spread Sheet Application (18 Hrs)** • Introduction to Excel features and Data Types. • Cell referencing and linking Sheets. • Introduction to various functions in all categories of Excel. • Concepts of sorting, filtering and validating data. • Analyzing data using charts, data tables, pivot tables, goal seek and scenarios

© NIMI NOT TO BE REPUBLISHED

54. Configure content for collaboration (2 Hrs)

- Set a print area
- Save workbooks in alternative file formats
- Configure print settings Manage data cells and ranges (12 Hrs.)

55. Manipulate data (3 Hrs)

- Paste data by using special paste options
- Fill cells by using Auto Fill
- Insert and delete multiple columns or rows
- Insert and delete cells

56. Format cells and ranges (5 Hrs)

- Merge and Unmerge cells
- Modify cell alignment, orientation and indentation
- Format cells using Format Painter
- Wrap text within cells
- Apply number formats
- Apply cell formats from the Format cells dialog box
- Apply cell styles
- Clear cell formatting

57. Define and reference named ranges (4 Hrs)

- Define a named range
- Name a table
- Summarize data visually Insert spark lines
- Apply built in conditional formatting
- Remove conditional formatting

Manage tables and table data (12 Hrs.)

58. Create and format tables (4 hrs)

- Create excel tables from cell ranges
- Apply table styles
- Convert tables to cell ranges

59. Manage tables and table data (5 Hrs)

- Add or remove table rows and columns
- Configure table style options
- Insert and configure total rows

60. Filter and sort table data (3 Hrs)

- Filter records
- Sort data by multiple columns Perform operations using formulas and functions (12 Hrs.)

61. Insert references (3 Hrs)

- Insert relative, absolute and mixed references
- Reference named ranges and named tables in formulas

62. Calculate and transform data (5 Hrs)

- Perform calculations using AVERAGE(), MIN(), MAX() and SUM()
- Count cells by using COUNT(), COUNTIF() and COUNTBLANK()
- Perform conditional operations by

NOT TO BE REPUBLISHED © NIMI

		using the IF() function 63. Format and modify text(4 Hrs) • Format text using RIGHT(),LEFT() and MID() functions • Format text using UPPER(), LOWER() and LEN() functions • Format text using CONCAT() and TEXTJOIN() functions **Manage Charts (12 Hrs.)** 64. Create Charts (3 Hrs) • Create charts • Create chart sheets **65. Modify charts (4 Hrs)** • Add data series to charts • Switch between rows and columns in source data • Add and modify chart elements • Add trend lines to chart **66. Format charts (5 Hrs)** • Apply chart layouts • Apply chart styles • Add alternative text to charts for accessibility **Manage Pivot Tables (12 Hrs.)** **67. Create Pivot Tables(12 Hrs)** • Create Pivot tables from cell ranges • Manipulate fields (columns) to get desired analysis • Use Filters for pivot tables • Represent data as Count, Sum, Average & % of row / column • Group data in Columns & rows for aggregate reports	
Professional Skill - 53 Hrs; Professional Knowledge - 13 Hrs	Create and customize slides for presentation. (Mapped NOS: SSC/N3022)	**Power point Presentations (9.5 Hrs.)** **68. Open files in MS PowerPoint (1 Hr)** • Open MS PowerPoint • Create a new PowerPoint file • Create a new PowerPoint file from a template • Open an existing PowerPoint file **69. Format PowerPoint Presentations (1.5 Hrs)** • Add slides • Add titles and text • Select slide layouts • Add PowerPoint templates • Duplicate slides **70. Modify slide masters, handout masters, and note masters (2 Hrs)** • Change the slide master theme or background • Modify slide master content • Modify slide layouts **71. Change presentation options and views (1.5 Hrs)** • Change slide size • Display presentations in different views	**Power point Presentations (13 Hrs.)** • Image editing, Presentations • Introduction to Open Office. • Introduction to the properties and editing of images. • Introduction to different formats of images and their uses. • Introduction to Power Point and its advantages. • Creating Slide Shows. Fine tuning the presentation and good presentation technique

© NIMI NOT TO BE REPUBLISHED

		72. Save and share PowerPoint Presentations (1.5 Hrs) • Save presentations in alternative file formats • Configure different Print settings • Share presentations electronically **73. Configure and present slide shows (1 Hr)** • Hide unwanted slides while presenting • Configure slide show options • Present slide shows by using Presenter View **74. Prepare presentations for collaboration (1 Hrs)** • Protect presentations by using passwords • Export presentations to other formats **Format presentations (6.5 Hrs.)** **75. Insert text and paragraphs (2 Hrs)** • Find and replace text • Insert symbols and special characters **76. Format text and paragraphs (2 Hrs)** • Apply text effects • Apply formatting by using Format Painter • Set line and paragraph spacing and indentation • Apply built-in styles to text **77. Create and configure sections (2.5 Hrs)** • Format text in multiple columns • Text and image presentation styles • Clear formatting **Manage tables and bulleted text (8 Hrs)** 78. Create tables (3 Hrs) • Insert tables in PowerPoint • Apply built-in table styles • Create tables by specifying rows and columns **79. Modify tables (3 Hrs)** • Insert and delete table rows and columns • Configure cell margins and spacing • Merge and split cells • Resize tables, rows, and columns **80. Create and modify bulleted text (2 Hrs)** • Format paragraphs as numbered and bulleted lists • Change bullet characters and number formats • Increase and decrease list indents	

NOT TO BE REPUBLISHED © NIMI

		• Set starting number values • Restart and continue list numbering on different slides **Create and manage reference elements (hyperlinks) (1.25 Hrs)** • Create hyperlinks within presentations • Create hyperlinks in presentations for files and other sources **Manage graphic elements (11.5 Hrs)** **81. Insert illustrations and text boxes (3.5 Hrs)** • Insert shapes • Insert pictures • Insert SmartArt graphics • Insert screenshots and screen clippings **82. Format illustrations and text boxes (4 Hrs)** • Apply artistic effects • Apply picture effects and picture styles • Remove picture backgrounds • Crop images • Format graphic elements • Format SmartArt graphics **83. Add and modify text in graphic elements (4 Hrs)** • Add and modify text in text boxes • Add and modify text in shapes • Add and modify SmartArt graphic text • Create, insert and modify charts **Manage Audio & Video elements (6.5 Hrs)** **84. Add Audio elements (2 Hrs)** • Import audio files in presentations • Configure audio playback options **85. Add Video elements (4.5 Hrs)** • Import video files in presentations • Resize video to fit slide • Configure video playback options **Manage transitions and animations (9 Hrs)** **86. Add slide transitions (5 Hrs)** • Add same slide transition for all slides • Set transition effect duration • Configure transition start and finish options • Customise select slide transitions **87. Add animations (4 Hrs)** • Animate text and graphic elements • Order shapes, images, and text boxes • Group shapes, images, and text boxes	

NOT TO BE REPUBLISHED

		• Configure animation effects • Configure animation paths • Reorder animations on a slide **Manage collaboration (0.75 Hrs)** 88. Add and manage comments (0.75 Hrs) • Add comments • Review and reply to comments	
Professional Skill - 53 Hrs.; Professional Knowledge - 13 Hrs	Create and manage database file by using MySQL. (NOS: SSC/N9401)	**Demonstrate on (15 Hrs.)** 89. Installation of MySQL. (1 Hr) 90. Troubleshooting basic installation issues. (1 Hr) 91. Creation and use of database. (3 Hr) 92. Designing of tables. (3 Hr) 93. Applying data integrity rules. (2 Hr) 94. Using the DDL, DCL and DML statements. (2 Hrs) 95. Enforcing constraints, primary key and foreign key. (2 Hrs) 96. Adding indices to Tables. (1 Hr) **Demonstrate on (15 Hrs)** 97. Simple select queries. (5 Hrs) 98. Insert and delete queries Update queries. (10 Hrs) **Demonstrate on (23 Hrs)** 99. Using the Number, Date and Character functions. Joins and Functions (11.5 Hrs) 100. Joins, Group by, Having, Sub query. (11.5 Hrs)	**Database Concepts (4 Hrs.)** • Concept of DBMS, RDBMS. • Data Models, Concept of DBA, Database Users. • Database Schema. • Designing Database using Normalization Rules. • Various data types Data integrity, DDL DML and DCL statements. • Enforcing Primary key and foreign key. • Adding Indices. **Queries (4 Hrs)** • Concepts of Transactions • ACID Property of Transaction Constraints. **Joins and Functions (5 Hrs)** • Joining of tables • Sub Queries " Functions used in query like sum, average, max, min, count etc.
Professional Skill - 68 Hrs; Professional Knowledge - 16 Hrs	Install, setup/ configure, troubleshoot, and secure computer network including Internet. (Mapped NOS: SSC/N3022)	**Computer Network (68 Hrs.) Set-up & configure a Computer Network (48 Hrs.)** 101. View Network connections. (2 Hrs) 102. Connect a computer to a network and share Devices i.e. Printers, files, folders and drives. (4 Hrs) 103. Work with various Network devices, connectors and cables. Create straight and cross cable and punch a UTP cable in the patch socket and test the connectivity. (4 Hrs) 104. Practice IP Addressing and Subnet masking for IPV4/ IPV6 and pinging to test networks. (4 Hrs) 105. Configure Hub and Switch. (4 Hrs) 106. Set up and configure wired and wireless LAN in a Computer Lab within at least three computers. (6 Hrs) 107. Use patch panel & I/O Box for wired LAN and installing & configuring Internet connection in a single PC and in a LAN. (6 Hrs) 108. Set up a proxy server/ DHCP Server with firewall. (8 Hrs) 109. Set up video conferencing using open-source software. (4 Hrs)	**Communicating in a Connected World (12 Hrs.)** • Local Networks, • Communicating on a Local Network, Principles of Communications, • How do Ethernet Networks Work?, • How are Networks Built?, • Routing Across Networks • Explain how end- user devices and local networks interact with the global Internet. • Communicating in a Connected World Explain the concept of network communication. • Local Networks Explain the roles of devices in a network. • What Does a Home Network Look Like? • How Does Wi-Fi Work? • Introduction to LAN Devices, Internetworking Devices, **Internet Concepts (4 Hrs)** • Introduction to www, Concept of Internet, Web Browsers, internet servers and search engines. • Concepts of Domain naming Systems and E mail communication. • Introduction to video chatting tools and Social Networking concepts.

© NIMI NOT TO BE REPUBLISHED

		110. Use various tools (by open source /free) for network troubleshooting, maintenance and security for both Wired and Wireless (6 Hrs) 111. Set up Internet access & communication (10 Hrs) • Set-up internet connectivity • Set-up digital communication 112. Use the Internet (10 Hrs) • Browse the Internet • Use e-mail • Use Social Media • Use the phone for online activities	
Professional Skill - 67 Hrs; Professional Knowledge - 17 Hrs	Develop web pages using HTML and CSS. (Mapped NOS: SSC/N0503, SSC/N0501)	**Create simple static web pages using HTML tags (67 Hrs.)** 113. Practice HTML (46 Hrs) • Practice with basic HTML elements (e.g. head, title, body), tag and attributes. • Design simple web page with text, paragraph and line break using HTML tags • Format text, change background colour and insert pictures in web page • Design simple web page with tables and lists. • Use marquees, hyperlinks and mail to link in designing web pages • Create frames, add style and design layout. • Display a web page within a web page using iframes. • Insert text, check and combo box in web page. • Design web page using password field, submit button • Reset button and radio button etc. • Design a web page adding flash file, audio and video files. • Design web page with forms and form controls using HTML tags 114. Create simple static web pages using CSS (21 Hrs) • CSS syntax, Adding colors, fonts, backgrounds, images borders, text alignment, text transformation, Lists etc. • 3 types of CSS • Adding a Navigation Bars(vertical/ horizontal bars) • CSS drop downs & Forms • CSS counters and website layout, Multiple backgrounds & Putting the stylesheet in a separate file • CSS Animations & CSS Buttons	**Web Design Concepts (17 Hrs)** • Concepts of Static and Dynamic Web pages. • Introduction to HTML and various tags in HTML. • Concepts of different controls used in Web Pages. • Concepts of CSS and applying CSS to HTML. • Introduction to open source CMS viz, Joomla, Word press etc. and Web authoring tools viz. Kompozer, WordPress, Front Page etc.
Professional Skill - 173 Hrs;	Develop web pages using JavaScript. (Mapped NOS:	**JavaScript (173 Hrs)** **Embed JavaScript in HTML Pages (127 Hrs)**	**Introduction to JavaScript (35 Hrs)** • Introduction to Programming and Scripting Languages.

© NIMI NOT TO BE REPUBLISHED

Professional Knowledge - 35 Hrs	SSC/N0503, SSC/N0501)	115. Practicing the JavaScript in creating dynamic HTML pages. (53 Hrs) 116. Embed JavaScript in HTML to Display Information in Web pages. (31 Hrs) 117. Use error handling techniques in JavaScript. (11 Hrs) 118. Use objects and classes in JavaScript. (31 Hrs) 119. Describe Animation and Multimedia using JavaScript. (1 Hr) Create a dynamic website using an open-source tool (40 Hrs) 120. Develop dynamic HTML pages using JavaScript. Deploy a simple web project (6 Hrs) 121. Deploy web project using IIS.	• Introduction to JavaScript and its application for the web. • Introduction to Web Servers and their features. • JavaScript Basics - Data types, Variables, Constants and Conversion between data types. • Arithmetic, Comparison, Logical Operators in JavaScript. Operator precedence. • Program Control Statements and loops in JavaScript. • Arrays in JavaScript - concepts, types and usage. • The String data type in JavaScript. Introduction to String, Math and Date. • Introduction to Functions in JavaScript. • Built in JavaScript functions overview. • Concepts of Pop Up boxes in JavaScript. • Introduction to the Document Object Model. • Concepts of using Animation and multimedia files in Java Script.
Professional Skill - 73 Hrs Professional Knowledge - 17 Hrs	Create workbooks with advanced formulas, macros, charts, pivot tables and demonstrate ability to use Power tools. (NOS: SSC/N9402)	**Data Visualization or analysis using Excel - (73 Hrs)** **Create advanced formulas and macros (24 Hrs)** 122. Create and modify simple macros (6 Hrs) 123. Perform form controls and create simple data entry form with macros. (6 Hrs) 124. Look up data by using functions. (6 Hrs) 125. Use advanced date functions.(6 Hrs) **Manage advanced charts and tables (21 hrs)** 126. Create and modify advanced charts. (10 Hrs) 127. Create and modify PivotTables. (11 Hrs) **Use Power Query and Power BI (24 Hrs)** 128. Create a Power Query, Power Query Function. Invoking the Power Query function and combining queries. Organize the workbook queries (12 Hrs) 129. Use Power BI for simple data visualizations. (12 Hrs) **Make a dashboard in Excel (4 Hrs)**	**Advanced Excel Concepts - Theory- (17 Hrs)** • MS excel revision (row, columns, basic formatting, insert menu, Print setup, etc.) and Look up introduction and functions • Types of references and cell naming • Excel Linkage Custom Format and Excel Protection • Tips and tricks • Pivot table and Pivot chart • Conditional formatting • Advanced Graphs • Power Queries

NOT TO BE REPUBLISHED © NIMI

Professional Skill - 25hrs; Professional Knowledge - 10 Hrs.	Browse, select, and transact using E-commerce websites(NOS: SSC/N9403)	**Browse e-Commerce sites to identify products & services (6.5 Hrs)** 130. Demonstrate e-Commerce sites. (1.5 Hrs) 131. List features of e-commerce sites. (2 Hrs) 132. Use e-commerce sites to source an item. (3 Hrs) Shop online (4.5 Hrs) 133. Undertake transactions on an e-commerce site. (4.5 Hrs) **Manage e-commerce operations (14 Hrs)** 134. Add products to an ecommerce website. (4 Hrs) 135. Practice order processing. (3 Hrs) 136. Practice payment processing. (4.5 Hrs) 137. Identify common security issues. (3.5 Hrs)	**e-Commerce (10 Hrs)** • Introduction to E Commerce and advantages. • Building business on the net. • Payment and Order Processing, Authorization, Chargeback and otherpayment methods. • Security issues and payment gateways.
Professional Skill - 20 Hrs Professional Knowledge - 10 Hrs.	Secure information from Internet by using cyber security concept. (NOS: SSC/N9404)	**138. Protect information, computers and networks from viruses, spyware and other malicious code (19 Hrs)** • Explain Cyber security (2 Hrs) • Secure computers & the network (5.5 Hrs) • Reduce cyber security threats (2 Hrs) • Secure a Wi-Fi Network (4 Hrs) • Use Anti-Virus software (3 Hrs) • Perform back-ups of files, data & information (2.5 Hrs) **139. Explain compliance with IT Act (1 Hr)** • Identify steps for information privacy. (0.5 Hrs) • Identify common cybercrimes and penalties applicable. (0.5 Hrs)	**Cyber Security (10 Hrs)** • Overview of Information Security, SSL, HTTPS, Security threats, information Security vulnerability and Risk management. • Introduction to Directory Services, Access Control, Security, Privacy protection, Audit and Security. • Introduction to IT Act and penalties for cybercrimes.
Professional Skill -25 Hrs; Professional Knowledge 15 Hrs.	Explain Cloud concepts &services and Describe Application Development Life Cycle. (NOS: SSC/ N9405)	**Cloud Computing (15 Hrs) Working with Cloud Services (12 Hrs)** 140. Practice with IaaS using free cloud services. (4 Hrs) 141. Practice with PaaS using free cloud services. (4 Hrs) 142. Practice with SaaS using free cloud services. (4 Hrs) **Web hosting in Cloud (3 Hrs)** 143. Host a website in a free cloud. (3 Hrs) **Develop an application and perform the Application Development Life Cycle (10 Hrs)** 144. Identify Phases of the Application Development Life Cycle. (5 Hrs) 145. Describe Roles in each of the phases of Application Development Life Cycle. (5 Hrs)	**Introduction to Cloud Computing (12 Hrs)** • Benefits of cloud services, different categories. • Resources available in cloud. **Explain the Application Development Life Cycle (3 Hrs)** • Identify Phases of the Application Development Life Cycle. • Describe Roles in each of phases of the Application Development Life Cycle.

© NIMI NOT TO BE REPUBLISHED

To be selected either "Programming in Python" or "Programming in Java" as Elective Module for COPA.

COPA - Elective Module - I Programming in Python			
Professional Skill - 70 Hrs; Professional Knowledge - 30 Hrs.	Write programs using Python language. (NOS: SSC/N9406)	**Programming language (Python) Use Python from command line (7 Hrs)** 1. Install, set up the environment & run Python. (3 Hrs) 2. Use Command Line and IDE to create and execute a python program. (4 Hrs) **Perform Operations using Data Types and Operators (15 Hrs)** 3. Write and test a python program to demonstrate print statement, comments, different types of variables. (5 Hrs) 4. Write and test a python program to perform data and data type operations, string operations, date, input and output, output formatting and operators. (5 Hrs) 5. Determine the sequence of execution based on operator precedence. (5 Hrs) **Control Flow with Decisions and Loops (20hrs)** 6. Construct and analyze code segments that use branching statements. (10 Hrs) 7. Construct and analyze code segments that perform iteration. (10 Hrs) **Document and Structure Code (18 Hrs)** 8. Document code segments using comments and documentation strings. (3 Hrs) 9. Construct and analyze code segments that include List comprehensions, tuple, set and Dictionary comprehensions. (16 Hrs) **Perform Operations Using Modules and Tools (10 Hrs)** 10. Perform basic operations using built-in modules. (5 Hrs) 11. Solve complex computing problems by using built-in modules. (5 Hrs)	Programming language (Python) (30 Hrs) • Introduction to Python History • Features, Setting up path Basic Syntax, Comments, Variable • Different Data Types • Casting, string, Boolean • Python Operators • Conditional Statements • Looping • Control Statements, String Manipulation, Lists, Tuple, sets • Dictionaries • Arrays • Iterators, modules, dates, math, • Modules, Input and Output.
COPA - Elective Module - II Programming in JAVA			
Professional Skill - 70 Hrs; Professional Knowledge - 30 Hrs.	Writing programs using JAVA. (SSC/ N9407)	**Object Oriented Programming and JAVA Language (15 Hrs)** 1. Installing JAVA. 2. Setting the Class path. 3. Writing and Executing a simple JAVA Program to display "Hello". **Demonstrate writing JAVA programs:** 4. Use of various data types in JAVA. 5. Use of various operators in JAVA. 6. Create and use of Local, Instance and Class variables.	**Explain the following:** • Object Oriented Programming with Core Java • Java Programming features • JVM, Byte codes and Class path • Java Program Development • Compilation and Execution of JAVA programs • Basic JAVA language elements - keywords, comments, data types and variables.

© NIMI NOT TO BE REPUBLISHED

		7. Read text from the keyboard using scanner class read text from the keyboard using console class.	• JAVA Arithmetic, Assignment, Relational, Logical, Increment / Decrement operators and expressions. • JAVA String Operators • JAVA Input and Output streams, System in, System out. • Input using Scanner class and Console class methods. (10Hrs.)
		JAVA Program Flow Control (20 Hrs) **Demonstrate writing JAVA programs:** 8. Use of the if and if … else statements. 9. Use of the Switch statement. 10. Use of the Do … While and while - do loops. 11. Use of the For Loop. 12. Use of the Break and Continue Keywords. 13. Use of the JAVA Numbers Class methods. 14. Use of the JAVA Character Class methods. 15. Use of the JAVA String Class methods. 16. Create and use of arrays.	**Explain the following:** • Decision making and flow control using if…then, if then else, nested if, switch case and the conditional ternary operators in JAVA. • Loop control flow using while - do, do - while loops, for loop, using the break, continue statements. • Terminating the JAVA program. JAVA Number, Character and String Classes. Arrays in JAVA. (6Hrs.)
		JAVA Classes, Overloading and Inheritance (20 Hrs) **Demonstrate writing JAVA programs:** 17. Create and use of simple classes, objects and methods in JAVA. 18. Pass data and Objects to Methods. 19. Return data and Objects from Methods. 20. use of constructors in JAVA. 21. Create and use of Overloaded methods in JAVA. 22. Override methods in JAVA. 23. Create and use of Super class, Sub class in JAVA.	**Explain the following:** • JAVA Objects, Classes and Methods. • Passing data and objects as parameters to methods. • Method Overloading. • Constructors and Overloaded constructors. • Inheritance in JAVA. • Method Overriding in JAVA. (8Hrs.)
		Abstract Classes and Interfaces in JAVA (15 Hrs) Demonstrate writing JAVA programs: 24. Create and use virtual methods. 25. Create abstract classes and methods. 26. Create interfaces in JAVA. 27. Override methods in JAVA. 28. Create and implement an interface. 29. Extend interfaces in JAVA. 30. Create and use a package in JAVA. Troubleshooting Java issues Download and Install Java, Check and Verify Java Configurations, Test Java, Remove Old Versions of Java, Find Java version, Always redirected to the java.com download page	**Explain the following:** • Concept of Virtual methods. • Concept of Abstract classes and methods • Features of Abstract Classes • JAVA Interfaces and their advantages • Method Overriding in JAVA • Polymorphism in JAVA • Creating, implementing and extending interfaces • Creating and using Packages in JAVA. (6Hrs.)

Industrial Visit/Project work / Broad Area:

a) Create and host a web site of at least 6 web pages using JavaScript & CSS containing interactive objects, functions etc.

b) Create a project with Excel on Payroll Systems.

c) Create a database with MySQL on Library management system.

d) Create project in Python/Java programming language.

© NIMI NOT TO BE REPUBLISHED

© NIMI
NOT TO BE REPUBLISHED

ವಿದ್ಯುತ್ ಸುರಕ್ಷತೆ (Electrical safety)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ನೇರ ತಂತಿಯೊಂದಿಗಿನ ಸಂಪರ್ಕದಲ್ಲಿರುವ ವ್ಯಕ್ತಿಯನ್ನು ಹೇಗೆ ರಕ್ಷಿಸುವುದು ಎಂಬುದನ್ನು ವಿವರಿಸಿ.

ವಿದ್ಯುತ್ ಅಘಾತದ ತೀವ್ರತೆಯು ದೇಹದ ಮೂಲಕ ಹಾದುಹೋಗುವ ಪ್ರವಾಹದ ಮಟ್ಟವನ್ನು ಮತ್ತು ಎಷ್ಟು ಸಮಯದವರೆಗೆ ಸಂಪರ್ಕದಲ್ಲಿರುತ್ತದೆ ಎಂಬುದನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ. ವಿಳಂಬ ಮಾಡಬೇಡಿ, ಒಮ್ಮೆಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸಿ. ವಿದ್ಯುತ್ ಸಂಪರ್ಕವನ್ನು ಕಡಿತಗೊಳಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

ಅಪಘಾತಕ್ಕೊಳಗಾದವರು ಇನ್ನೂ ವಿದ್ಯುತ್‌ನ ಸಂಪರ್ಕದಲ್ಲಿದ್ದರೆ - ವಿದ್ಯುತ್ ಸ್ವಿಚ್ ಅನ್ನು ಆಫ್ ಮಾಡುವ ಮೂಲಕ ಅದನ್ನು ಕೊನೆಗೊಳಿಸಿ , ಪ್ಲಗ್ ತೆಗೆಯಿರಿ ಅಥವಾ ಕೇಬಲ್‌ನ್ನು ತಿರುಚಿರಿ. ಇಲ್ಲದಿದ್ದರೆ, ಒಣ ಮರ, ರಬ್ಬರ್, ಅಥವಾ ಪ್ಲಾಸ್ಟಿಕ್‌ನಂತಹ ಕೆಲವು ಪ್ರತಿರೋಧಕಗಳ ಮೇಲೆ ನಿಂತುಕೊಳ್ಳಿ ಅಥವಾ ನಿಮ್ಮನ್ನು ಪ್ರತಿರೋಧಗೊಳಿಸಿಕೊಳ್ಳಲು ನಿಮ್ಮ ಕೈಯಲ್ಲಿರುವ ಯಾವುದಾದರೂ ವಸ್ತುವನ್ನು ಬಳಸಿ ಮತ್ತು ವ್ಯಕ್ತಿಯನ್ನು ತಳ್ಳುವ ಅಥವಾ ಎಳೆಯುವ ಮೂಲಕ ಸಂಪರ್ಕ ಕಡಿತಗೊಳಿಸಿ ವ್ಯಕ್ತಿಯನ್ನು ಮುಕ್ತವಾಗಿಸಿ. (ಚಿತ್ರ 1 & 2)

Fig 1

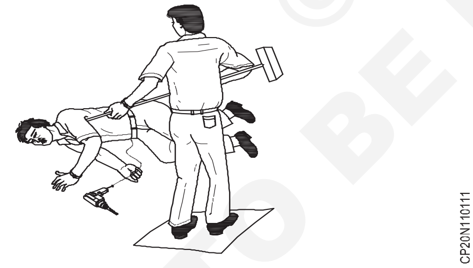
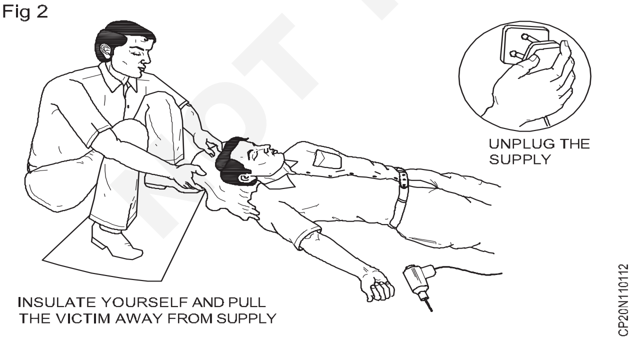


Fig 2

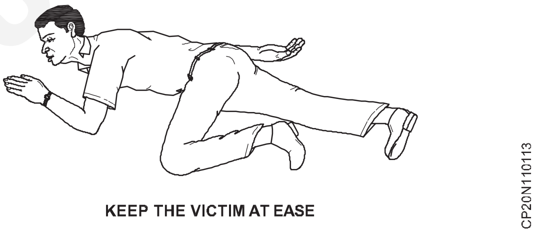


ನೀವು ಪ್ರತಿರೋಧಕವಾಗಿ ಉಳಿಯದಿದ್ದರೆ, ಸರ್ಕಿಟ್‌ನ್ನು ನಿರ್ಜೀವವಾಗುವವರೆಗೆ ಅಥವಾ ವ್ಯಕ್ತಿಯನ್ನು ಆ ಸಲಕರಣೆಯಿಂದ ದೂರ ಸರಿಸುವ ವರೆಗೆ , ಅಪಘಾತಕ್ಕೊಳಗೊಳಗಾದ ವ್ಯಕ್ತಿಯನ್ನು ಬರಿ ಕೈಯಿಂದ ಮುಟ್ಟಬೇಡಿ.

ಅಪಘಾತಕ್ಕೊಳಗಾದ ವ್ಯಕ್ತಿ ಮೇಲಿದ್ದರೆ, ಅವನನ್ನು ಕೆಳಗೆ ಬೀಳದಂತೆ ತಡೆಯಬೇಕು ಅಥವಾ ಕನಿಷ್ಟ ಸುರಕ್ಷಿತವಾಗಿ ಬೀಳುವಂತೆ ಮಾಡಬೇಕು. ವಿದ್ಯುತ್ ಆಘಾತದಿಂದ ವ್ಯಕ್ತಿಯ ಮೇಲಿನ ಸುಟ್ಟಗಾಯಗಳು ದೊಡ್ಡದಾಗಿರದಿದ್ದರೂ ಆಳವಾಗಿರಬಹುದು. ನೀವು ಆ ಪ್ರದೇಶವನ್ನು ಸ್ವಚ್ಛವಾದ , ಸ್ಟೆರೈಲ್ ಆದ ಡ್ರೆಸಿಂಗ್‌ನಿಂದ ಆವರಿಸಿ ಮತ್ತು ಶಾಕ್‌ಗೆ ಚಿಕಿತ್ಸೆ ಕೊಡಿ. ಸಾಧ್ಯವಾದಷ್ಟು ಬೇಗನೇ ತಜ್ಞರ ಸಹಾಯ ಪಡೆಯಿರಿ.

ಅಪಘಾತಕ್ಕೊಳಗಾದ ವ್ಯಕ್ತಿ ಪ್ರಜ್ಞಾಹೀನನಾಗಿದ್ದರೂ ಉಸಿರಾಡುತ್ತಿದ್ದರೆ, ಅವನ ಕುತ್ತಿಗೆ , ಎದೆ ಮತ್ತು ಸೊಂಟದ ಬಟ್ಟೆಯನ್ನು ಸಡಿಲಗೊಳಿಸಿ ಮತ್ತು ವ್ಯಕ್ತಿಯನ್ನು ಚೇತರಿಕೆ ಸ್ಥಿತಿಯಲ್ಲಿ ಕೂರಿಸಿ (ಚಿತ್ರ 3)

Fig 3



ಉಸಿರಾಟದ ಮತ್ತು ನಾಡಿ ದರವನ್ನು ನಿರಂತರವಾಗಿ ಪರಿಶೀಲಿಸಿ.

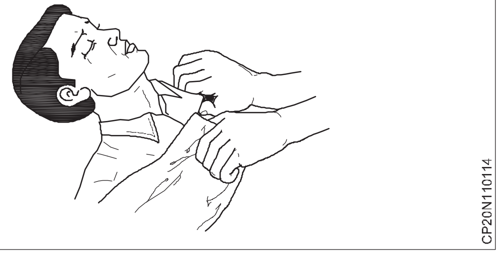
ಅಪಘಾತಕ್ಕೊಳಗಾದ ವ್ಯಕ್ತಿಯನ್ನು ಬೆಚ್ಚಗಿನ ಮತ್ತು ಆರಾಮದಾಯಕವಾದ ಸ್ಥಿತಿಯಲ್ಲಿ ಇರಿಸಿ. (ಚಿತ್ರ 4)

ಸಹಾಯಕ್ಕಾಗಿ ಕಳುಹಿಸಿ.

> ಪ್ರಜ್ಞಾಹೀನನಾದ ವ್ಯಕ್ತಿಗೆ ಬಾಯಿಯ ಮೂಲಕ ಏನನ್ನೂ ನೀಡಬೇಡಿ.
>
> ಪ್ರಜ್ಞಾಹೀನನಾದ ವ್ಯಕ್ತಿಯನ್ನು ಗಮನಿಸಲಾರದೇ ಬಿಡಬೇಡಿ.

ಅಪಘಾತಕ್ಕೊಳಗಾದ ವ್ಯಕ್ತಿ ಉಸಿರಾಡದಿದ್ದರೆ - ಸಮಯ ವ್ಯರ್ಥ ಮಾಡದೇ ಒಮ್ಮೆಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸಿ.

Fig 4



© NIMI NOT TO BE REPUBLISHED

ಸುರಕ್ಷತಾ ಅಭ್ಯಾಸ - ಅಗ್ನಿ ಶಾಮಕ (Safety Practice - fire extinguishers)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

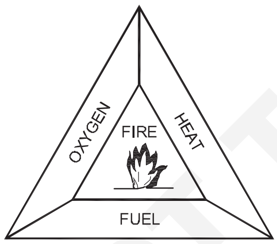
- **ನೇರ ತಂತಿಯೊಂದಿಗಿನ ಸಂಪರ್ಕದಲ್ಲಿರುವ ವ್ಯಕ್ತಿಯನ್ನು ಹೇಗೆ ರಕ್ಷಿಸುವುದು ಎಂಬುದನ್ನು ವಿವರಿಸಿ**
- **ವಿವಿಧ ರೀತಿಯ ಅಗ್ನಿ ಶಾಮಕಗಳನ್ನು ಪ್ರತ್ಯೇಕಿಸಿ**
- **ಬೆಂಕಿಯ ವರ್ಗವನ್ನು ಆಧರಿಸಿ ಬಳಸಬೇಕಾದ ಸರಿಯಾದ ರೀತಿಯ ಅಗ್ನಿಶಾಮಕವನ್ನು ನಿರ್ಧರಿಸಿ**
- **ಬೆಂಕಿಯ ಸಂದರ್ಭದಲ್ಲಿ ಅನುಸರಿಸಬೇಕಾದ ಸಾಮಾನ್ಯ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ.**

ಬೆಂಕಿ ಎಂದರೆ ದಹನಕಾರಿ ವಸ್ತುಗಳನ್ನು ಸುಡುವುದು. ಅನಗತ್ಯ ಸ್ಥಳದಲ್ಲಿ, ಮತ್ತು ಅನಗತ್ಯ ಸಂದರ್ಭದಲ್ಲಿ ಮತ್ತು ಒಂದು ಅನಿಯಂತ್ರಿತ ಪ್ರಮಾಣದಲ್ಲಿ ಇರುವ ಬೆಂಕಿಯು, ಹಾನಿಯನ್ನುಂಟು ಮಾಡುತ್ತದೆ ಅಥವಾ ಆಸ್ತಿಯನ್ನು ಮತ್ತು ವಸ್ತುಗಳನ್ನು ನಾಶಪಡಿಸುತ್ತದೆ. ಇದು ಜನರನ್ನು ಗಾಯಗೊಳಿಸಬಲ್ಲದು ಮತ್ತು ಕೆಲವೊಮ್ಮೆ ಜೀವ ನಷ್ಟಕ್ಕೂ ಕಾರಣವಾಗಬಹುದು. ಅದ್ದರಿಂದ ಬೆಂಕಿಯನ್ನು ತಡೆಗಟ್ಟಲು ಪ್ರತಿ ಪ್ರಯತ್ನವನ್ನು ಮಾಡಬೇಕು. ಬೆಂಕಿ ಕಾಣಿಸಿಕೊಂಡದ್ದು ಪತ್ತೆಯಾದರೆ, ಅದನ್ನು ನಿಯಂತ್ರಿಸಬೇಕು ಮತ್ತು ತಕ್ಷಣದ ಸರಿಪಡಿಸುವ ಕ್ರಮವಾಗಿ ಬೆಂಕಿ ನಂದಿಸಬೇಕು.

ಬೆಂಕಿಯನ್ನು ತಡೆಯಲು ಸಾಧ್ಯವೇ ? ಹೌದು. ಬೆಂಕಿಯನ್ನು ಉಂಟುಮಾಡುವ ಮೂರು ಅಂಶಗಳಲ್ಲಿ ಯಾವುದಾದರೂ ಒಂದನ್ನು ತೆಗೆದುಹಾಕುವ ಮೂಲಕ ಬೆಂಕಿಯನ್ನು ತಡೆಯಬಹುದು.

ಬೆಂಕಿ ಸುಡುವುದನ್ನು ಮುಂದುವರೆಸಲು, ಈ ಕೆಳಗಿನ ಮೂರು ಅಂಶಗಳು ಸಂಯೋಜನೆಯಲ್ಲಿ ಇರಬೇಕು (ಚಿತ್ರ 1)

Fig 1



ಇಂಧನ: ಆಮ್ಲಜನಕ ಮತ್ತು ಸಾಕಷ್ಟು ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದ ತಾಪಮಾನವಿದ್ದರೆ ಯಾವುದೇ ವಸ್ತು, ದ್ರವ, ಘನ ಅಥವಾ ಅನಿಲ ಸುಡುತ್ತದೆ.

ಶಾಖ: ಪ್ರತಿ ಇಂಧನವು ಒಂದು ನಿರ್ದಿಷ್ಟ ತಾಪಮಾನದಲ್ಲಿ ಸುಡಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಇದು ಇಂಧನವನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ. ಘನವಸ್ತುಗಳು ಮತ್ತು ದ್ರವವಸ್ತುಗಳು ಬಿಸಿಯಾದಾಗ ಆವಿಯಾಗುತ್ತವೆ ಮತ್ತು ಈ ಆವಿಯು ಉರಿಯುತ್ತದೆ. ಕೆಲವು ದ್ರವಗಳನ್ನು ಬಿಸಿ ಮಾಡಬೇಕಾದ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ ಏಕೆಂದರೆ ಅವು ಸಾಮಾನ್ಯ ಕೋಣೆಯ ಉಷ್ಣಾಂಶದಲ್ಲಿ ಎಂದರೆ 15° C ಯಲ್ಲಿ ಆವಿಯಾಗುತ್ತದೆ. ಉದಾ : ಪೆಟ್ರೋಲ್.

ಆಮ್ಲಜನಕ : ಬೆಂಕಿಯನ್ನು ಉರಿಸಲು ಅವಶ್ಯಕವಿರುವ ಆಮ್ಲಜನಕವು ಸಾಮಾನ್ಯವಾಗಿ ಗಾಳಿಯಲ್ಲಿ ಸಾಕಷ್ಟು ಪ್ರಮಾಣದಲ್ಲಿ ಇರುತ್ತದೆ.

ಬೆಂಕಿಯನ್ನು ನಂದಿಸುವುದು: ಸಂಯೋಜನೆಯಿಂದ ಯಾವುದಾದರೂ ಒಂದು ಅಂಶವನ್ನು ಪ್ರತ್ಯೇಕಿಸುವುದು ಅಥವಾ ತೆಗೆದು ಹಾಕುವುದರಿಂದ ಬೆಂಕಿಯನ್ನು ನಂದಿಸಬಹುದು. ಇದನ್ನು ಸಾಧಿಸುವ ಮೂರು ಮೂಲ ಮಾರ್ಗಗಳು ಕೆಳಗಿನಂತಿವೆ :

- ಸ್ಟಾರ್ವಿಂಗ್ (Starving) - ಬೆಂಕಿಗೆ ಇಂಧನ ಸಿಗದಂತೆ ಮಾಡುವುದು.
- ಸ್ಮೊದರಿಂಗ್ (Smothering) - ಎಂದರೆ, ಬುರುಗು(foam) , ಮರಳಿ(sand) ಮುಂತಾದವುಗಳಿಂದ ಹೊದಿಕೆ ಹೊದಿಸುವ ಮೂಲಕ, ಆಮ್ಲಜನಕದ ಸರಬರಾಜನ್ನು ಬೆಂಕಿಯಿಂದ ಪ್ರತ್ಯೇಕಿಸಿ.
- ಕೂಲಿಂಗ್ (Cooling) - ತಾಪಮಾನವನ್ನು ಕಡಿಮೆ ಮಾಡಲು ನೀರನ್ನು ಬಳಸಿ. ಈ ಮೂರು ಅಂಶಗಳಲ್ಲಿ ಯಾವುದಾದರೂ ಒಂದನ್ನು ತೆಗೆದುಹಾಕುವುದರಿಂದ ಬೆಂಕಿಯನ್ನು ನಂದಿಸಬಹುದು.

ಬೆಂಕಿಯನ್ನು ತಡೆಗಟ್ಟುವುದು : ಹೆಚ್ಚಿನ ಬೆಂಕಿಯು ಸಣ್ಣದಾದ ಏಕಾಏಕಿ ಬೆಂಕಿಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಇದು ಸುರಕ್ಷಿತ ಹಿಡಿದಿಡುವವರೆಗೆ ಗಮನವಿಲ್ಲದಂತೆ ಸುಡುತ್ತಿರುತ್ತದೆ. ಹೆಚ್ಚಿನ ಬೆಂಕಿಯನ್ನು ಹೆಚ್ಚು ಎಚ್ಚರಿಕೆಯಿಂದ ಮತ್ತು ಕೆಲವು ಸರಳ ಸಾಮಾನ್ಯ ಜ್ಞಾನದ ನಿಯಮಗಳನ್ನು ಅನುಸರಿಸುವುದರಿಂದ ತಡೆಯಬಹುದು.

ದಹನಶೀಲ ತ್ಯಾಜ್ಯ ಸಂಗ್ರಹ (ಎಣ್ಣೆಯಲ್ಲಿ ನೆನೆಸಿದ ಹತ್ತಿಯ ತ್ಯಾಜ್ಯ, ಮರದ ತುಂಡು, ಕಾಗದ ಇತ್ಯಾದಿಗಳು)ಗಳು ಮೂಲೆಯಲ್ಲಿದ್ದರೆ ಬೆಂಕಿಗೆ ಅಪಾಯಕಾರಿಯಾಗಿದೆ. ಈ ತ್ಯಾಜ್ಯಗಳನ್ನು ಸಂಗ್ರಹ ಮೂಲೆಗಳಿಂದ ತೆಗೆದುಹಾಕಬೇಕು.

ವಿದ್ಯುತ್ ಉಪಕರಣಗಳಲ್ಲಿನ ಬೆಂಕಿಗೆ ದುರುಪಯೋಗ ಅಥವಾ ನಿರ್ಲಕ್ಷ್ಯ ಕಾರಣವಿರುತ್ತದೆ. ಸಡಿಲವಾದ ಸಂಪರ್ಕಗಳು, ತಪ್ಪಾದ

NOT TO BE REPUBLISHED © NIMI

ಫ್ಯೂಸ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು, ಓವರ್‌ಲೋಡ್ ಆದ ಸರ್ಕಿಟ್‌ಗಳು ಇವೆಲ್ಲವೂ ಅಧಿಕ ಬಿಸಿಯಾಗಲು ಕಾರಣವಾಗುತ್ತದೆ ಮತ್ತು ಅದು ಬೆಂಕಿಗೆ ಕಾರಣವಾಗಬಹುದು. ಕೇಬಲ್‌ಗಳಲ್ಲಿನ ಕಂಡಕ್ಟರ್‌ಗಳ ನಡುವಿನ ಇನ್ಸುಲೇಷನ್‌ನ ಹಾನಿಯು ಬೆಂಕಿಗೆ ಕಾರಣವಾಗುತ್ತದೆ.

ಬೆಂಕಿಯನ್ನು ಹಿಡಿಯುವ ಬಟ್ಟೆ ಅಥವಾ ಯಾವುದಾದರೂ ವಸ್ತುಗಳನ್ನು ಹೀಟರ್‌ಗಳಿಂದ ದೂರ ಇಡಬೇಕು. ಹೀಟರ್ ಅನ್ನು ಕೆಲಸದ ದಿನದ ಕೊನೆಯಲ್ಲಿ ಸ್ಥಗಿತಗೊಳಿಸಬೇಕು ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

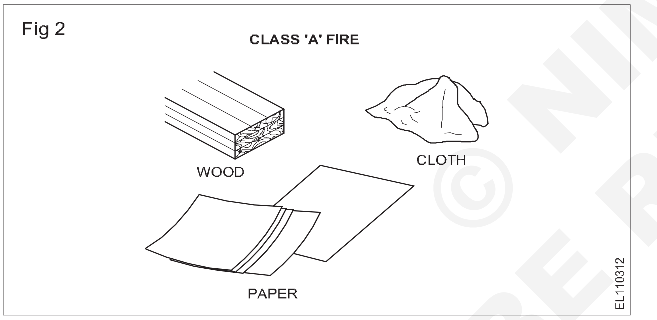
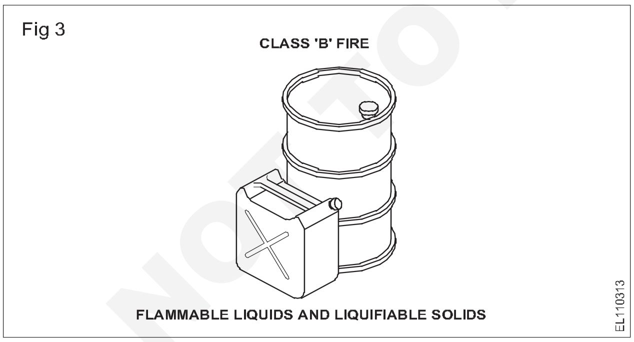
ಹೆಚ್ಚು ಸುಡುವ ದ್ರವಗಳು ಮತ್ತು ಪೆಟ್ರೋಲಿಯಂ ಮಿಶ್ರಣಗಳು (ಥಿನ್ನರ್, ಅಂಟಿಕೊಳ್ಳುವ ದ್ರಾವಣಗಳು, ದ್ರಾವಕಗಳು, ಸೀಮೆಎಣ್ಣೆ, ಸ್ಪಿರಿಟ್, LPG ಗ್ಯಾಸ್, ಇತ್ಯಾದಿಗಳು) ಇವುಗಳನ್ನು ಸುಡುವ ವಸ್ತುಗಳನ್ನು ಸಂಗ್ರಹಿಸುವ ಪ್ರದೇಶದಲ್ಲಿ ಸಂಗ್ರಹಿಸಿಡಬೇಕು.

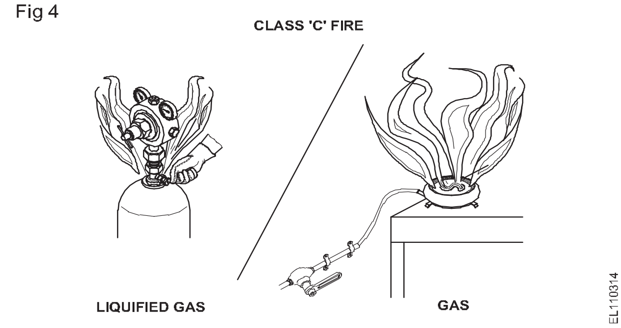
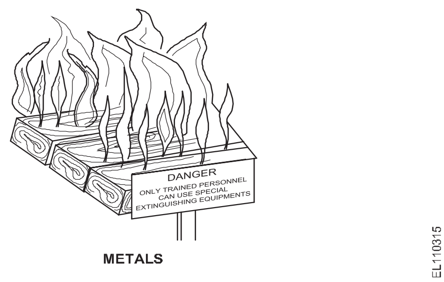
ಬ್ಲೋಲ್ಯಾಂಪ್‌ಗಳು ಮತ್ತು ಟಾರ್ಚ್‌ಗಳು ಬಳಕೆಯಲ್ಲಿಲ್ಲದ ಸಮಯದಲ್ಲಿ ಉರಿಯಲು ಬಿಡಬಾರದು.

ಬೆಂಕಿಯನ್ನು ನಂದಿಸುವುದು : ಇಂಧನದ ಸ್ವರೂಪದ ಆಧಾರದ ಮೇಲೆ, ಬೆಂಕಿಯನ್ನು ನಾಲ್ಕು ವಿಧಗಳಾಗಿ ವಿಂಗಡಿಸಲಾಗಿದೆ. ವಿವಿಧ ರೀತಿಯ ಬೆಂಕಿಗಳನ್ನು (ಚಿತ್ರ 2, ಚಿತ್ರ 3, ಚಿತ್ರ 4 ಮತ್ತು ಚಿತ್ರ 5) ವಿಭಿನ್ನ ರೀತಿಯಿಂದ ಮತ್ತು ವಿವಿಧ ಬೆಂಕಿ ನಂದಿಸುವ ಏಜಂಟ್‌ಗಳಿಂದ ವ್ಯವಹರಿಸಬೇಕು.

ಬೆಂಕಿ ನಂದಿಸುವ ಏಜಂಟ್ ಎಂದರೆ ಬೆಂಕಿ ನಂದಿಸುವ ಒಂದು ಕಚ್ಚಾ ವಸ್ತು ಅಥವಾ ವಸ್ತುವಾಗಿದೆ ಮತ್ತು ಸಾಮಾನ್ಯವಾಗಿ (ಯಾವಾಗಲೂ ಅಲ್ಲ) ಬೆಂಕಿಯ ಮೇಲೆ ಸಿಂಪಡಿಸಲು ಬಿಡುಗಡೆಯ ಯಾಂತ್ರಿಕತೆಯನ್ನು ಹೊಂದಿರುವ ಇದನ್ನು ಅಗ್ನಿ ಶಾಮಕದಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ನಿರ್ದಿಷ್ಟ ರೀತಿಯ ಬೆಂಕಿಯನ್ನು ಆರಿಸುವ ಸರಿಯಾದ ರೀತಿಯ ಏಜಂಟ್ ಅನ್ನು ತಿಳಿದುಕೊಳ್ಳುವುದು ಬಹಳ ಮುಖ್ಯವಾಗಿರುತ್ತದೆ ತಪ್ಪಾದ ಏಜಂಟ್ ಅನ್ನು ಬಳಸುವುದು ವಿಷಯಗಳನ್ನು ಮತ್ತಷ್ಟು ಕೆಟ್ಟದಾಗಿಸುತ್ತದೆ. `ವಿದ್ಯುತ್ ಬೆಂಕಿಗೆ' ಯಾವುದೇ ವರ್ಗೀಕರಣ ಇರುವುದಿಲ್ಲ. ಇದು ವಿದ್ಯುತ ಇರುವಂತಹ ವಸ್ತುಗಳಲ್ಲಿ ಮಾತ್ರ ಈ ಬೆಂಕಿಯು ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ.

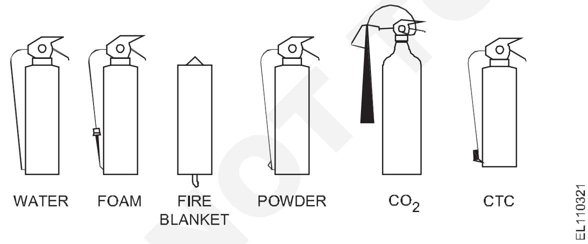
ಇಂಧನ	ನಂದಿಸುವಿಕೆ
	ನೀರಿನಿಂದ ತಂಪಾಗಿಸುವುದು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿದೆ, ನೀರಿನ ಕಾರಂಜಿಯನ್ನು ಬೆಂಕಿಯ ಅಡಿಪಾಯಕ್ಕೆ ಮತ್ತು ಕ್ರಮೇಣ ಮೇಲಕ್ಕೆ ಸಿಂಪಡಿಸಬೇಕು.ಸ್ಮೊದರಿಂಗ್ ಮಾಡುವುದು ಎಂದರೆ ಆಮ್ಲಜನಕ ಸಿಗದಂತೆ ಮಾಡುವುದು.
	ಇಲ್ಲಿ ಸುಡುವ ದ್ರವದ ಸಂಪೂರ್ಣ ಮೇಲ್ಮೈಯನ್ನು ಹೊದಿಸುವ ಗುರಿಯಾಗಿರುತ್ತದೆ. ಇದು ಬೆಂಕಿಗೆ ಆಮ್ಲಜನಕದ ಪೂರೈಕೆಯನ್ನು ತಡೆಯುವ ಪರಿಣಾಮವನ್ನು ಹೊಂದಿದೆ. ಸುಡುವ ದ್ರವಗಳ ಮೇಲೆ ನೀರನ್ನು ಎಂದಿಗೂ ಬಳಸಬಾರದು. ಈ ರೀತಿಯ ಬೆಂಕಿಯ ಮೇಲೆ , ಬುರುಗು, ಒಣ ಪೌಡರ್, ಅಥವಾ CO^2 ಅನ್ನು ಉಪಯೋಗಿಸಬಹುದು

ಇಂಧನ	ನಂದಿಸುವಿಕೆ
Fig 4 	ದ್ರವೀಕೃತ ಅನಿಲಗಳ ಜೊತೆಗೆ ವ್ಯವಹರಿಸುವಾಗ ತೀವ್ರ ಎಚ್ಚರಿಕೆ ವಹಿಸುವುದು ಅಗತ್ಯವಾಗಿದೆ. ಇಲ್ಲಿ ಇಡೀ ಸುತ್ತಮುತ್ತಲ ಪ್ರದೇಶದಲ್ಲಿ ಸ್ಫೋಟದ ಮತ್ತು ಹಠಾತ್ ಏಕಾಏಕಿ ಬೆಂಕಿಯ ಅಪಾಯವಿದೆ. ಸಿಲಿಂಡರ್ ನಿಂದ ತುಂಬಿದ ಉಪಕರಣವು ಬೆಂಕಿಯನ್ನು ಹಿಡಿದರೆ, - ಅನಿಲದ ಸರಬರಾಜನ್ನು ಸ್ಥಗಿತಗೊಳಿಸಿ. ಸುರಕ್ಷಿತ ಮಾರ್ಗವೆಂದರೆ ಅಲಾರಂ ಅನ್ನು ಬಾರಿಸುವುದು ಮತ್ತು ತರಬೇತಿ ಪಡೆದ ಸಿಬ್ಬಂದಿಗಳಿಗೆ ಬೆಂಕಿಯನ್ನು ನಿಭಾಯಿಸಲು ಬಿಡಿ.
Fig 5 	ಈ ರೀತಿಯ ಬೆಂಕಿಯನ್ನು ಆರಿಸಲು ಒಣ ಪೌಡರನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಈಗ ವಿಶೇಷ ಪೌಡರ್‌ಗಳನ್ನು ಅಬಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ. ಇವು ಈ ರೀತಿಯ ಬೆಂಕಿಯನ್ನು ನಿಯಂತ್ರಿಸಲು ಮತ್ತು/ಅಥವಾ ಆರಿಸಲು ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿವೆ.ಬೆಂಕಿ ನಂದಿಸುವ ಏಜಂಟ್‌ಗಳ ಪ್ರಮಾಣಿತ ಶ್ರೇಣಿಯು ಲೋಹದ ಬೆಂಕಿಯ ಜೊತೆಗೆ ವ್ಯವಹರಿಸುವಾಗ ಅಸಮರ್ಪಕ ಅಥವಾ ಅಪಾಯಕಾರಿಯಾಗಿಸುತ್ತವೆ. ವಿದ್ಯುತ್ ಉಪಕರಣಗಳಲ್ಲಿನ ಬೆಂಕಿ. ವಿದ್ಯುತ್ ಉಪಕರಣಗಳಲ್ಲಿನ ಬೆಂಕಿಯನ್ನು ಆರಿಸಲು ಹ್ಯಾಲೊನ್, ಕಾರ್ಬನ್ ಡೈಆಕ್ಸೈಡ್, ಡ್ರೈ ಪೌಡರ್, ಮತ್ತು ಆವಿಯಾಗುವ ದ್ರವ(CTC) ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. ವಿದ್ಯುತ್‌ಉಪಕರಣಗಳಲ್ಲಿನ ಬೆಂಕಿ ಆರಿಸಲು ಯಾವುದೇ ಸಂದರ್ಭಗಳಲ್ಲಿ ಬುರುಗು ಅಥವಾ ದ್ರವ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳನ್ನು ಬಳಸಬಾರದು.

© NIMI NOT TO BE REPUBLISHED

ವಿವಿಧ ವರ್ಗದ ಬೆಂಕಿಯನ್ನು ಆರಿಸಲು ವಿಭಿನ್ನ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶಿಂಗ್ ಏಜಂಟ್ಗಳೊಂದಿಗೆ ಅನೇಕ ರೀತಿಯ ಅಗ್ನಿ ಶಾಮಕಗಳು ಲಭ್ಯವಿರುತ್ತವೆ. (ಚಿತ್ರ 1)

Fig 1

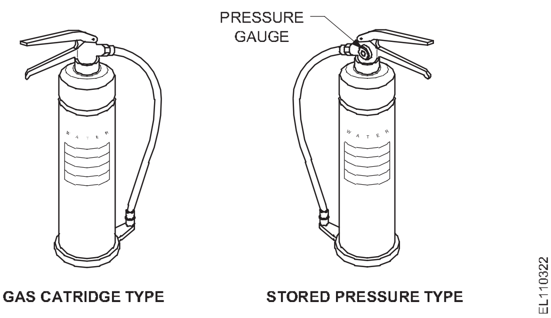


ನೀರು ತುಂಬಿದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು (Water filled Extinguishers) **:** **ಇದರಲ್ಲಿ ಎರಡು ರೀತಿಯ ಕಾರ್ಯಾಚರಣೆಗಳಿವೆ (ಚಿತ್ರ 2)**

- ಗ್ಯಾಸ್ ಕಾರ್ಟ್ರಿಡ್ಜ್ ವಿಧ
- ಸ್ಟೋರ್ಡ್ ಪ್ರೆಷರ್ ವಿಧ

ಎರಡೂ ಕಾರ್ಯಾಚರಣೆಗಳ ವಿಧಾನಗಳಲ್ಲಿ, ಡಿಸ್ಚಾರ್ಜ್ ಅನ್ನು ಅಗತ್ಯಕ್ಕೆ ತಕ್ಕಂತೆ ನಿಯಂತ್ರಿಸಬಹುದು, ಅದರಲ್ಲಿನ ವಸ್ತುವನ್ನು

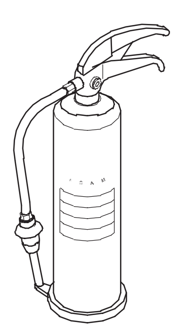
Fig 2



ಸಂರಕ್ಷಿಸಬಹುದು ಮತ್ತು ಅನಗತ್ಯ ನೀರು ಪೋಲಾಗುವುದನ್ನು ತಡೆಯಬಹುದು.

ಫೋಮ್ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು (ಚಿತ್ರ 3) : ಇವು ಸ್ಟೋರ್ಡ್ ಪ್ರೆಷರ್ ಅಥವಾ ಗ್ಯಾಸ್ ಕಾರ್ಟೇಡ್ಜ್ ವಿಧದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ಆಗಿರಬಹುದು. ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವ ಮೊದಲು ಯಾವಾಗಲೂ ಅದರ ಮೇಲಿನ ಆಪರೇಟಿಂಗ್ ಸೂಚನೆಗಳನ್ನು ಪರಿಶೀಲಿಸಿಕೊಳ್ಳಿ.

Fig 3



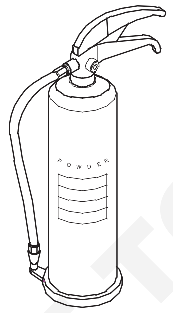
ಇವುಗಳಿಗೆ ಹೆಚ್ಚು ಸೂಕ್ತವಾಗಿದೆ

- ಫ್ಲೇಮೆಬಲ್ ಲಿಕ್ವಿಡ್ ಫೈರ್ (flammable liquid fires)
- ರನ್ನಿಂಗ್ ಲಿಕ್ವಿಡ್ ಫೈರ್ (running liquid fires)

ವಿದ್ಯುತ್ ಉಪಕರಣಗಳನ್ನು ಒಳಗೊಂಡಂತಹ ಬೆಂಕಿಯಲ್ಲಿ ಬಳಸಬಾರದು.

ಡ್ರೈ ಪೌಡರ್ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು (ಚಿತ್ರ 4) :ಡ್ರೈಪೌಡರ್ ಉಪಯೋಗಿಸಿದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ಗ್ಯಾಸ್ ಕಾರ್ಟ್ರಿಡ್ಜ್‌ಗಳಾಗಿರಬಹುದು ಅಥವಾ ಸ್ಟೋರ್ಡ್ ಪ್ರೆಷರ್ ವಿಧದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ಆಗಿರಬಹುದು. ಕಾಣುವಿಕೆ ಮಾತ್ತು ಕಾರ್ಯಾಚರಣೆಯ ವಿಧಾನ ನೀರು ತುಂಬಿದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳಂತೆ ಇರುತ್ತದೆ. ಒಂದು ಮುಖ್ಯ ವ್ಯತ್ಯಾಸವೆಂದರೆ ಫೋರ್ಕ್ ಆಕಾರದ ನಳಿಕೆಯಾಗಿದೆ. ಪೌಡರ್‌ಗಳನ್ನು ಕ್ಲಾಸ್ D ಬೆಂಕಿಯನ್ನು ನಂದಿಸಲು ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ.

Fig 4

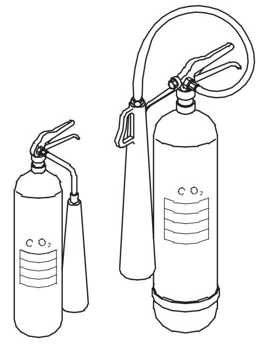


ಕಾರ್ಬನ್ ಡೈಯಾಕ್ಸೈಡ್ (CO_2) : ಈ ಪ್ರಕಾರವನ್ನು ವಿಶಿಷ್ಟ ಆಕಾರದ ಡಿಸ್ಚಾರ್ಜ್ ಹಾರ್ನ್ ಮೂಲಕ ಸುಲಭವಾಗಿ ಗುರುತಿಸಬಹುದು. (ಚಿತ್ರ 5)

ಕ್ಲಾಸ್ ಬಿ ಬೆಂಕಿಗಳಿಗೆ ಹೆಚ್ಚು ಸೂಕ್ತವಾಗಿದೆ. ಮಾಲಿನ್ಯ ಇರುವಲ್ಲಿ ಸೂಕ್ತವಾಗಿದೆ. ತೆರೆದ ಗಾಳಿಯಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಪರಿಣಾಮಕಾರಿಯಾಗಿರುವುದಿಲ್ಲ.

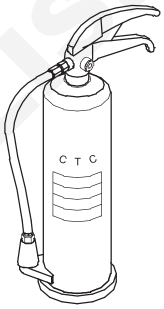
ಬಳಕೆಗೆ ಮೊದಲು ಯಾವಾಗಲೂ ಕಂಟೇನರ್ ಮೇಲಿರುವ ಆಪರೇಟಿಂಗ್ ಸೂಚನೆಗಳನ್ನು ಪರಿಶೀಲಿಸಿ. ಕಾರ್ಯಚರಣೆಯ ವಿಭಿನ್ನ ಗ್ಯಾಜೆಟ್ ಗಳೊಂದಿಗೆ ಲಭ್ಯವಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ - ಪ್ಲಂಗರ್, ಲಿವರ್, ಟ್ರಿಗರ್ ಇತ್ಯಾದಿಗಳು.

Fig 5



ಹ್ಯಾಲೋನ್ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು (ಚಿತ್ರ 6) : ಈ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ಕಾರ್ಬನ್ - ಟೆಟ್ರಾಕ್ಲೋರೈಡ್ ಮತ್ತು ಬ್ರೊಮೊಕ್ಲೋರೊಡೈಫ್ಲೂರೊ ಮೀಥೇನ್ (BCF) ಗಳಿಂದ ತುಂಬಿರುತ್ತವೆ. ಇವು ಗ್ಯಾಸ್ ಕಾರ್ಟ್ರಿಡ್ಜ್‌ಗಳಾಗಿರಬಹುದು ಅಥವಾ ಸ್ಟೋರ್ಡ್ ಪ್ರೆಷರ್ ವಿಧದ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ಆಗಿರಬಹುದು.

Fig 6



ದ್ರವಗಳನ್ನು ಸುರಿಯುವುದನ್ನು ಒಳಗೊಂಡ ಇವು , ಸಣ್ಣ ಬೆಂಕಿಯನ್ನು ನಂದಿಸುವಲ್ಲಿ ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿವೆ. ಈ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ವಿಶೇಷವಾಗಿ ವಿದ್ಯುತ್ ಉಪಕರಣಗಳ ಮೇಲೆ ಉಪಯೋಗಿಸಲು ಸೂಕ್ತವಾಗಿವೆ ಮತ್ತು ಸುರಕ್ಷಿತವಾಗಿವೆ. ಏಕೆಂದರೆ, ರಾಸಾಯನಿಕಗಳು ವಿದ್ಯುತ್ ಅವಾಹಕಗಳಾಗಿರುತ್ತವೆ.

ಈ ಎಕ್ಸ್‌ಟಿಂಗ್ವಿಶರ್‌ಗಳು ನೀಡಿದ ಹೊಗೆಗಳು ಅಪಾಯಕಾರಿಯಾಗಿರುತ್ತವೆ, ವಿಶೇಷವಾಗಿ ಸೀಮಿತ ಸ್ಥಳದಲ್ಲಿ.

ಬೆಂಕಿ ಹತ್ತಿದ ಸಂದರ್ಭದಲ್ಲಿ ಸಾಮಾನ್ಯ ವಿಧಾನ ಕೆಳಗಿನಂತಿರುತ್ತದೆ :

- ಅಲಾರಂ ಹೆಚ್ಚಿಸಿರಿ.
- ಎಲ್ಲ ಯಂತ್ರೋಪಕರಣಗಳನ್ನು ಮತ್ತು ವಿದ್ಯುತ್ತನ್ನು ಆಫ್ ಮಾಡಿರಿ.(ಗ್ಯಾಸ್ ಮತ್ತು ವಿದ್ಯುತ್)
- ಬಾಗಿಲು ಮತ್ತು ಕಿಟಕಿಗಳನ್ನು ಮುಚ್ಚಿರಿ, ಆದರೆ ಲಾಕ್ ಮಾಡಬೇಡಿ ಅಥವಾ ಬೋಲ್ಟ್ ಹಾಕಬೇಡಿ. ಇದು ಆಮ್ಲಜನಕ ಸರಬರಾಜನ್ನು ಮಿತಿಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಅದರ ಹರಡುವಿಕೆಯನ್ನು ತಡೆಯುತ್ತದೆ.
- ನೀವು ಬೆಂಕಿಯನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ತಡೆಯಲು ಸಾಧ್ಯವಾದರೆ ಪ್ರಯತ್ನಿಸಿ. ಸಿಕ್ಕಿಹಾಕಿಕೊಳ್ಳುವ ಅಪಾಯವನ್ನು ಎದುರಿಸಬೇಡಿ.

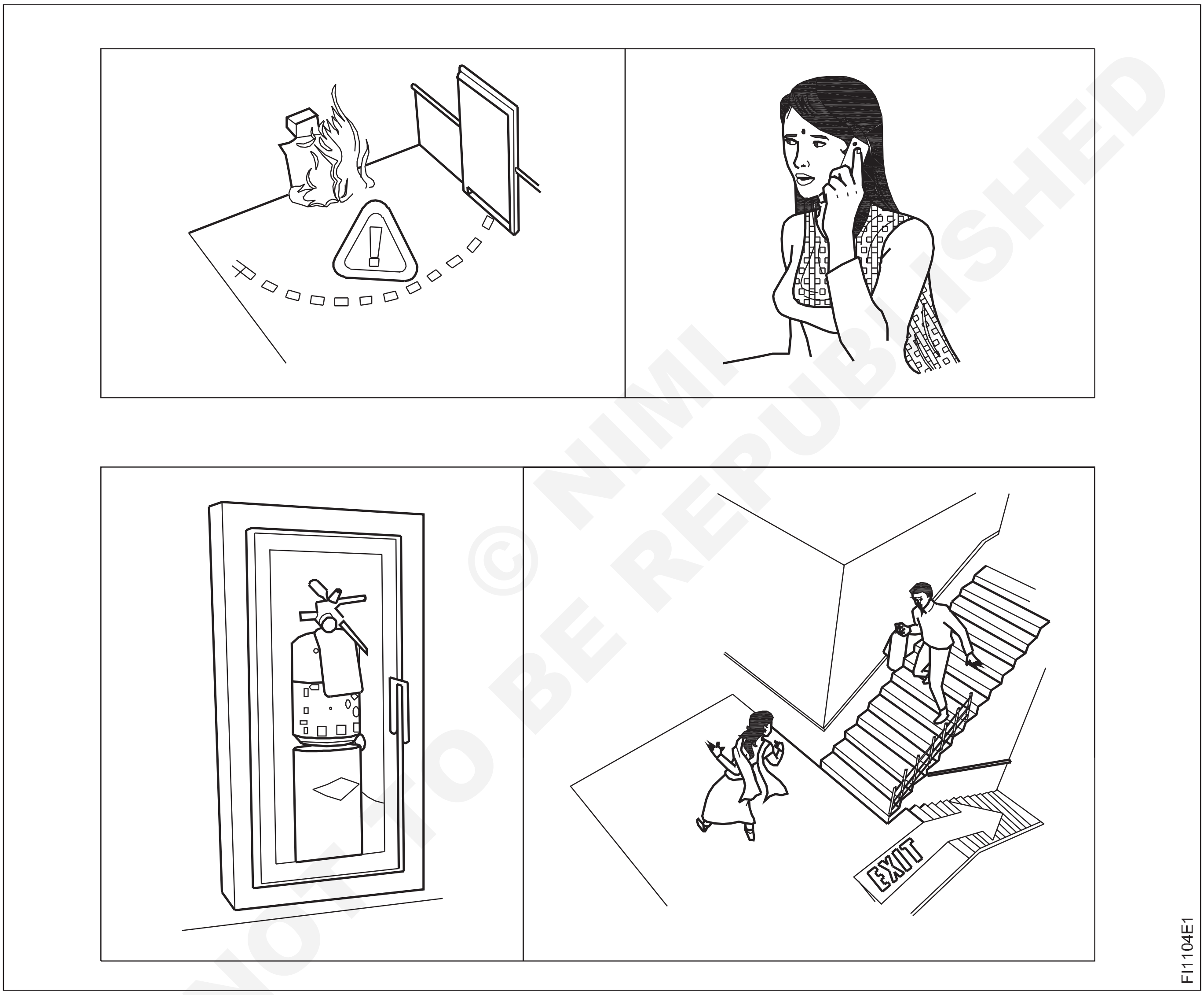
© NIMI NOT TO BE REPUBLISHED

- ಬೆಂಕಿಯೊಂದಿಗೆ ಹೋರಾಡುವ ಸಿಬ್ಬಂದಿಯಲ್ಲದಿದ್ದರೆ ತುರ್ತು ನಿರ್ಗಮನ ಬಳಸಿ ಶಾಂತವಾಗಿ ಹೊರಗೆ ಹೋಗಿರಿ ಮತ್ತು ಗೊತ್ತುಪಡಿಸಿದ ಅಸೆಂಬ್ಲಿ ಪಾಯಿಂಟ್‌ಗೆ ಹೋಗಿರಿ. ಇದನ್ನು ಮಾಡಲು ವಿಫಲವಾದರೆ, ಅವರು ಲೆಕ್ಕಕ್ಕೆ ಸಿಗದವರಾಗುತ್ತಾರೆ ಮತ್ತು ಅವರಿಗಾಗಿ ಇತರರು ಅವರನ್ನು ಹುಡುಕುವ ಅಪಾಯಕ್ಕೆ ತಮ್ಮನ್ನು ತಾವು ಸಿಲುಕಿಸಬೇಕಾಗುತ್ತದೆ.

ಅಗ್ನಿಶಾಮಕಗಳ ಮೇಲೆ ಅಭ್ಯಾಸ ಮಾಡಿ (Practice on fire extinguishers)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಬೆಂಕಿಯ ಪ್ರಕಾರದ ಆಧಾರದ ಮೇಲೆ ಅಗ್ನಿಶಾಮಕಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ಬಗ್ಗೆ ವಿವರಿಸಿ**
- **ಅಗ್ನಿಶಾಮಕಗಳ ಕಾರ್ಯಚರಣೆಯ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ.**
- **ಬೆಂಕಿಯನ್ನು ನಂದಿಸುವುದು ಹೇಗೆ ಎಂಬುದನ್ನು ವಿವರಿಸಿ.**



FI1104E1

© NIMI NOT TO BE REPUBLISHED

ಕಾರ್ಯವಿಧಾನ (PROCEDURE) (**ಚಿತ್ರ** 1)

- ಬೆಂಕಿಯನ್ನು ನೋಡಿದ ತಕ್ಷಣ ಬೆಂಕಿ, ಬೆಂಕಿ, ಬೆಂಕಿ ಎಂದು ಕೂಗುವ ಮೂಲಕ ಸುತ್ತಲಿನ ಜನರನ್ನು ಎಚ್ಚರಿಸಿ.
- ಅಗ್ನಿಶಾಮಕ ಸೇವೆಗೆ ತಿಳಿಸಿ ಅಥವಾ ತಕ್ಷಣ ತಿಳಿಸಲು ವ್ಯವಸ್ಥೆ ಮಾಡಿರಿ.ತುರ್ತು ನಿರ್ಗಮನದ ಬಾಗಿಲನ್ನು ತೆರೆಯಿರಿ ಮತ್ತು ಅವರಿಗೆ ಹೊರಗೆ ಹೋಗಲು ತಿಳಿಸಿ
- ವಿದ್ಯುತ್ ಸರಬರಾಜನ್ನು ಆಫ್ ಮಾಡಿರಿ.

ಟೇಬಲ್ 1

Class A	ಮರ, ಕಾಗದ, ಬಟ್ಟೆ, ಘನ ವಸ್ತು
Class B	ತೈಲ ಆಧಾರಿತ ಬೆಂಕಿ (ಗ್ರೀಸ್, ಗ್ಯಾಸೋಲಿನ್, ತೈಲ) ದ್ರವಿಕರಿಸಬಹುದಾದ ಅನಿಲಗಳು (ಲಿಕ್ವಿಫೈಯೆಬಲ್ ಗ್ಯಾಸ್)
Class C	ಅನಿಲ, ದ್ರವಿಕರಿಸಬಹುದಾದ ಅನಿಲಗಳು(ಲಕ್ವಿಫೈಯೆಬಲ್ ಗ್ಯಾಸ್)
Class D	ಲೋಹಗಳು, ವಿದ್ಯುತ್ ಉಪಕರಣಗಳು

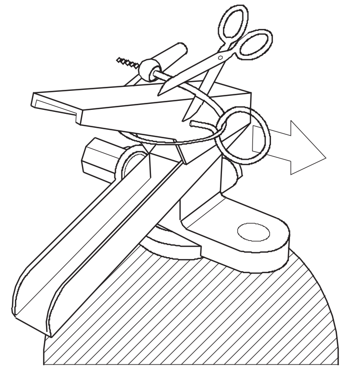
ಜನರು ಬೆಂಕಿಯ ಹತ್ತಿರ ಹೋಗಲು ಅನುಮತಿ ನೀಡಬೇಡಿರಿ.

- ಬೆಂಕಿಯ ಪ್ರಕಾರವನ್ನು ವಿಶ್ಲೇಷಿಸಿ ಮತ್ತು ಗುರುತಿಸಿ. (ಟೇಬಲ್ 1 ಅನ್ನು ನೋಡಿ)

ಬೆಂಕಿಯು B ವರ್ಗದಾಗಿದೆ ಎಂದು ಊಹಿಸಿಕೊಳ್ಳಿ. (ಸುಡುವ ದ್ರವರೂಪದ ಘನವಸ್ತುಗಳು)

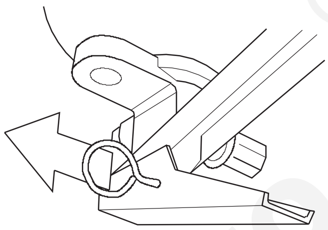
- CO^2 ಅಗ್ನಿಶಾಮಕವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- CO^2 ಅಗ್ನಿಶಾಮಕವನ್ನು ಪತ್ತೆ ಮಾಡಿರಿ ಮತ್ತು ಎತ್ತಿಕೊಳ್ಳಿರಿ, ಅದರ ಮುಕ್ತಾಯದ ದಿನಾಂಕವನ್ನು (Expiry date) ಪರಿಶೀಲಿಸಿ.
- ಸೀಲ್ ಅನ್ನು ಮುರಿಯಿರಿ (ಚಿತ್ರ 2)

Fig 2



- ಹ್ಯಾಂಡಲ್‌ನಿಂದ ಸುರಕ್ಷತಾ ಪಿನ್ ಅನ್ನು ಎಳೆಯಿರಿ (ಅಗ್ನಿಶಾಮಕದ ಮೇಲ್ಭಾಗದಲ್ಲಿ ಪಿನ್ ಇರುತ್ತದೆ)

Fig 3



- ಅಗ್ನಿಶಾಮಕದ ನಳಿಕೆ ಅಥವಾ ಹಾಸ್ ಅನ್ನು ಬೆಂಕಿಯ ತಳಭಾಗಕ್ಕೆ ಗುರಿಯಾಗಿಸಿ (ಇದು ಬೆಂಕಿಯ ಇಂಧನದ ಮೂಲವನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ) (ಚಿತ್ರ 4)

ನಿಮ್ಮನ್ನು ನೀವು ಕೆಳಭಾಗದಲ್ಲಿರಿಸಿ.

- ಏಜಂಟ್‌ನ್ನು ಹೊರಹಾಕಲು ಹ್ಯಾಂಡಲ್ ಲೀವರನ್ನು ನಿಧಾನವಾಗಿ ಹಿಸುಕಿ (ಚಿತ್ರ 5)
- ಬೆಂಕಿಯನ್ನು ನಿಲ್ಲಿಸುವವರೆಗೆ ಬೆಂಕಿಯ ಇಂಧನದ ಮೇಲೆ ಸುಮಾರು 15 ಸೆ. ಮೀ ವರೆಗೆ ಸಿಂಪಡಿಸಿ. (ಚಿತ್ರ 5)

Fig 4

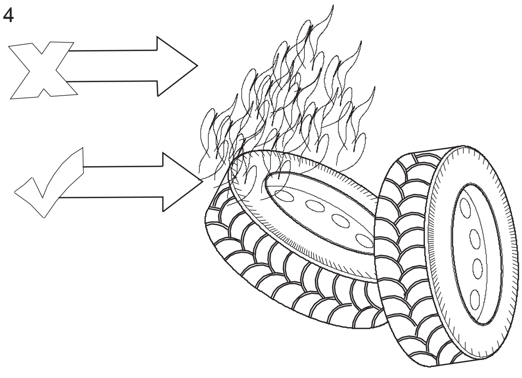
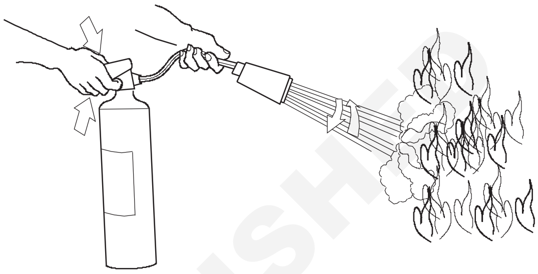


Fig 5



ಅಗ್ನಿಶಾಮಕಗಳನ್ನು ದೂರದಿಂದ ಉಪಯೋಗಿಸುವಂತೆ ತಯಾರಿಸಲಾಗಿರುತ್ತದೆ.

ಎಚ್ಚರಿಕೆ

- ಬೆಂಕಿಯನ್ನು ನಂದಿಸುವಾಗ, ಬೆಂಕಿಯು ಭಗ್ಗನೆ ಹತ್ತಿಕೊಳ್ಳಬಹುದು.
- ಭಯ ಬೀಳಬೇಡಿ,
- ಅಗ್ನಿಶಾಮಕವನ್ನು ಬಳಸಿದ ನಂತರವೂ ಬೆಂಕಿಯು ಉತ್ತಮವಾಗಿ ಪ್ರತಿಕ್ರಿಯಿಸದಿದ್ದರೆ, ಬೆಂಕಿ ಇರುವ ಸ್ಥಳದಿಂದ ದೂರ ಸರಿಯಿರಿ.
- ಬೆಂಕಿಯು ವಿಷಕಾರಿ ಹೊಗೆಯನ್ನು ಸೂಸುತ್ತಿದ್ದರೆ, ಅದನ್ನು ಹೊರಹಾಕಲು ಪ್ರಯತ್ನಿಸಬೇಡಿರಿ. ಅದನ್ನು ವೃತ್ತಿಪರರಿಗೆ ಬಿಡಿರಿ.
- ನಿಮ್ಮ ಜೀವನವು ನಿಮ್ಮ ಆಸ್ತಿಗಿಂತ ಮುಖ್ಯವಾದುದು ಎಂದು ನೆನಪಿಡಿ. ಅದ್ದರಿಂದ ನಿಮ್ಮನ್ನು ಅಥವಾ ಇತರನ್ನು ಅಪಾಯಕ್ಕೆ ಸಿಲುಕಿಸಬೇಡಿರಿ.

ಅಗ್ನಿಶಾಮಕದ ಸರಳ ಕಾರ್ಯಾಚರಣೆಯನ್ನು ನೆನಪಿಡುವ ಸಲುವಾಗಿ P. A. S. S ಅನ್ನು ನೆನಪಿಡಿರಿ. ಇದು ಅಗ್ನಿಶಾಮಕವನ್ನು ಬಳಸುವಾಗ ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತದೆ.

P ಎಂದರೆ ಪುಲ್ (ಎಳೆಯಿರಿ)

A ಎಂದರೆ ಏಮ್ (ಗುರಿಯಿಡಿ)

S ಎಂದರೆ ಸ್ಕ್ವೀಝ್ (ಹಿಸುಕಿರಿ)

S ಎಂದರೆ ಸ್ವೀಪ್ (ಗುಡಿಸಿರಿ)

© NIMI NOT TO BE REPUBLISHED

ಕಂಪ್ಯೂಟರ್ ಪರಿಚಯ (Introductions to Computers)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಕಂಪ್ಯೂಟರ್‌ನ ವ್ಯಾಖ್ಯಾನ ಹಾಗೂ ವರ್ಗೀಕರಣ
- ಕಂಪ್ಯೂಟರ್‌ನ ಅನುಕೂಲಗಳು ಹಾಗೂ ಮಿತಿಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುವುದು.
- ಕಂಪ್ಯೂಟರ್‌ನ ಉಪಯೋಗಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುವುದು
- ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ವೋಲ್ಟೇಜ್ ಹಾಗೂ ಕರೆಂಟನ್ನು ವಿವರಿಸುವುದು

ಕಂಪ್ಯೂಟರ್ - ವ್ಯಾಖ್ಯೆ (Computer - definition)

ಕಂಪ್ಯೂಟರ್ ಒಂದು ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಯಂತ್ರವಾಗಿದೆ. ತನ್ನ ಮೆಮೊರಿಯಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುವ ಸೂಚನೆಗಳಂತೆ ದತ್ತಾಂಶಗಳನ್ನು ಸಂಗ್ರಹಿಸುವುದು (input), ಹಾಗೂ ಉತ್ತರವನ್ನು ನೀಡುವುದು (output), ಹಾಗೂ ಭವಿಷ್ಯದ ಉಪಯೋಗಕ್ಕಾಗಿ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿಡುವುದು ಕಂಪ್ಯೂಟರ್‌ನ ಮುಖ್ಯ ಕಾರ್ಯವಾಗಿದೆ

ತಾಂತ್ರಿಕವಾಗಿ ಕಂಪ್ಯೂಟರ್ ಒಂದು ಪ್ರೋಗ್ರಾಮೇಬಲ್ ಯಂತ್ರವಾಗಿದೆ. ಇದು ಪ್ರೋಗ್ರಾಮ್‌ನ ಸೂಚನೆಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ (Execute) ಹಾಗೂ ಹೊಸ ಸೂಚನೆಗಳಿಗೆ ಉತ್ತರಿಸುತ್ತದೆ.

ಕಂಪ್ಯೂಟರ್‌ನ ಇತಿಹಾಸ (History of Computer)

ಚಾರ್ಲ್ಸ್ ಬ್ಯಾಬೇಜನ ಯಂತ್ರ (Charles Babbage`s machine)

1833 ರಲ್ಲಿ ಚಾರ್ಲ್ಸ್ ಬ್ಯಾಬೇಜ ಎಂಬ ಇಂಗ್ಲೀಷ ಗಣೀತಜ್ಞ ಸಂಶೋಧಿಸಿದ ಅನಾಲ್ಯಾಟಿಕಲ್ ಇಂಜಿನ್‌ನ ಕಾರ್ಯತಂತ್ರದಂತೆ ಈಗಿರುವ ಕಂಪ್ಯೂಟರ್ ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿವೆ. ಈ ಯಂತ್ರವು ಕಡಿಮೆ ತಂತ್ರಾಂಶಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಕೆಲಸ ಮಾಡಬಲ್ಲದಾಗಿತ್ತು.

ಅನಾಲ್ಯಾಟಿಕಲ್ ಇಂಜಿನ್ ಯಂತ್ರವು 4 ಭಾಗಗಳನ್ನು ಒಳಗೊಂಡಿತ್ತು. ಮಿಲ್ - ಲೆಕ್ಕಗಳನ್ನು ನಿರ್ವಹಿಸುವ ಭಾಗ (CPU), ಸ್ಟೋರ್ - ಮಾಹಿತಿ ಸಂಗ್ರಹಿಸುವ ಭಾಗ ,ರೀಡರ್- ಪಂಚ್ಡ ಕಾರ್ಡ್ಸ ಮುಖಾಂತರ ಮಾಹಿತಿ ನಮೂದಿಸುವ ಭಾಗ(Keyboard) , ಹಾಗೂ ಪ್ರಿಂಟರ್.

ಅದ್ದರಿಂದ ಚಾರ್ಲ್ಸ್ ಬ್ಯಾಬೇಜ ಇವರನ್ನು ಕಂಪ್ಯೂಟರ್‌ನ ಪಿತಾಮಹ ಎಂದು ಕರೆಯುತ್ತಾರೆ. (Fig 1)

ಪ್ರಮುಖ ತಾಂತ್ರಿಕ ಬೆಳವಣಿಗೆಗಳಾದಂತೆ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಕಾರ್ಯವೈಖರಿ ಬದಲಾಯಿತು. ಅದರಂತೆ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಚಿಕ್ಕ, ಅಗ್ಗ ಹಾಗೂ ಶಕ್ತಿಶಾಲಿ ಹಾಗೂ ನಂಬಿಕೆಯುಳ್ಳ ಹಾಗೂ ಸಮರ್ಥ ಯಂತ್ರಗಳಾಗಿವೆ. ಈ ಆಧಾರದ ಮೇಲೆ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಪೀಳಿಗೆಗಳನ್ನಾಗಿ ವರ್ಗೀಕರಿಸಲಾಯಿತು.

ಮೊದಲನೇಯ ಪೀಳಿಗೆ (First Generation) (1946-1954) :

1946 ರಲ್ಲಿ ಡಿಜಿಟಲ್ ಸಾಧನಗಳಾದ ವಿದ್ಯುನ್ಮಾನ ಕವಾಟಗಳನ್ನು(vacuume Tubes) ಉಪಯೋಗಿಸಿ ತಯಾರಿಸಲಾಯಿತು. ಕಂಪ್ಯೂಟರ್‌ನ main memory ಗಾಗಿ ದುಬಾರಿ vacuume tube ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಯಿತು. ಶಬ್ದ ತರಂಗಗಳ ಪ್ರಸರಣದ ಮುಖಾಂತರ ಮಾಹಿತಿ ಸಂಗ್ರಹವಾಗುತ್ತಿತ್ತು.

ಆದರೆ vacuume tube ಗಳು ಬಹಳ ವಿದ್ಯುತ್ತನ್ನು ಬಳಸುತ್ತಿದ್ದವು. ಇವು ಗಾತ್ರದಲ್ಲಿ ದೊಡ್ಡದಾಗಿದ್ದು, program ಗಳನ್ನು ಬರೆಯುವುದು ಕಷ್ಟವಾಗಿತ್ತು. ಮೊದಲನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಯಾವವು ಎಂದರೆ -

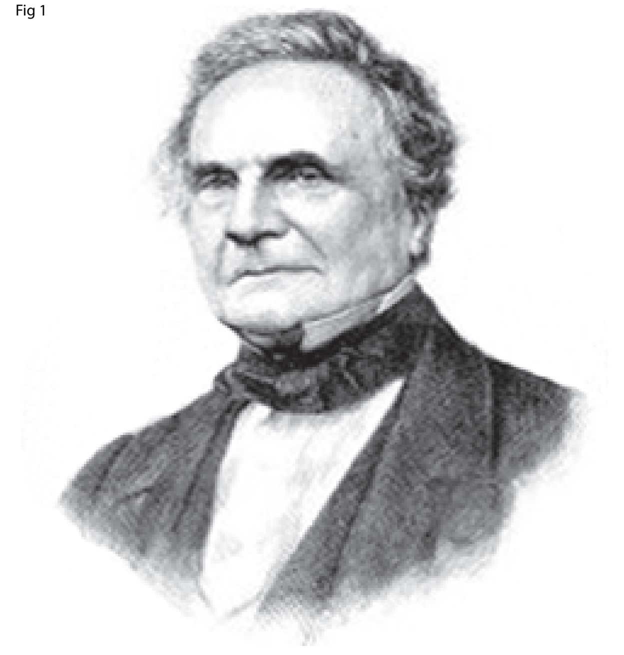
MARK -1 : MARK 1 ಇದು ಮೊದಲನೇ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಹಾರ್ವರ್ಡ ಯುನಿವರ್ಸಿಟಿಯ IBM Automatic Sequence Controlled (ASCC) ಯನ್ನು MARK - I ಎಂದು ಕರೆಯಲಾಯಿತು. ಇದು ಒಂದು electro- mechanical ಕಂಪ್ಯೂಟರ್ ಆಗಿತ್ತು. ಇದು ಅಂಕಗಣಿತ ಹಾಗೂ ತಾರ್ಕಿಕ ಕಾರ್ಯಗಳನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಮಾಡಬಲ್ಲಂತಹ ಮೊದಲನೇ ಯಂತ್ರವಾಗಿತ್ತು.

ENIAC (ElectroNumerical Integrator and Calculator) :

ಇದು 1946 ರಲ್ಲಿ John Eckert & John Mouchly ಇವರು ಪೆನಿನ್ಸುಯೆಲಾ ವಿಶ್ವವಿದ್ಯಾಲಯ , ಯು.ಎಸ್.ಎ ನಲ್ಲಿ ತಯಾರಿಸಿದ ಮೊದಲನೇ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ENIAC ಇದು 30-40 ಫೂಟುಗಳಷ್ಟು ಉದ್ದವಾಗಿತ್ತು, ಹಾಗೂ 30 ಟನ್‌ಗಳಷ್ಟು ಭಾರವಾಗಿತ್ತು. 18,000 vacuume tube ಗಳನ್ನು, 70,000 ರೆಸಿಸ್ಟರ್‌ಗಳನ್ನು, 10,000 ಕೆಪ್ಯಾಸಿಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ತಯಾರಿಸಲಾಗಿತ್ತು. ಇದು 1,50,000 ವ್ಯಾಟ್ ವಿದ್ಯುತ್ತನ್ನು

© NIMI NOT TO BE REPUBLISHED

Fig 1



CP110111

ಬಳಸಿಕೊಳ್ಳುತ್ತಿತ್ತು. ಇವತ್ತಿನ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಗಾತ್ರ ಚಿಕ್ಕದಾಗಿದ್ದರೂ ಬಹಳ ಶಕ್ತಿಶಾಲಿಯಾಗಿವೆ.

EDVAC (Electronic Descrete Variable Automatic Computer) : EDVAC ನ್ನು 1950 ರಲ್ಲಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಯಿತು. ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ದತ್ತಾಂಶ ಹಾಗೂ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಪರಿಕಲ್ಪನೆ ಇಲ್ಲಿಂದ ಶುರುವಾಯಿತು. ಇದರಿಂದಾಗಿ ಕಂಪ್ಯೂಟರ್ ಅತಿ ವೇಗವಾಗಿ ದತ್ತಾಂಶ ಹಾಗೂ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದರಿಂದ ಅದರ ಕಾರ್ಯಗಳು ವೇಗವಾಗಿ ನಡೆಯುತ್ತವೆ. ಮಾಹಿತಿ ಸಂಗ್ರಹಣೆಯಿಂದ ಇನ್ನೊಂದು ಲಾಭವೆಂದರೆ ಆಂತರಿಕವಾಗಿ ತಾರ್ಕಿಕ ನಿರ್ಧಾರಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದಾಗಿತ್ತು. EDVAC ಇದು Binary Serial Computer ಆಗಿತ್ತು. ಇದು ultrasonic serial memory ಯಿಂದ ಸ್ವಯಂಚಾಲಿತ ತಪಾಸಣೆ ಮಾಡಬಲ್ಲದಾಗಿತ್ತು.

EDSAC (Electronic Delay storage Automatic Computer) :

1949 ರಲ್ಲಿ ಕೇಂಬ್ರಿಡ್ಜ ವಿಶ್ವವಿದ್ಯಾಲಯದಲ್ಲಿ ಎಮ್ ವಿ ವಿಕೀಸ್ ಎಂಬುವವರಿಂದ ಅಭಿವೃದ್ಧಿಪಡಿಸಲ್ಪಟ್ಟಿತು. ಇದು ಮೊದಲನೆ stored program computer ಆಗಿತ್ತು. edsac ಇದು 3 ಮಿಲಿ ಸೆಕೆಂಡುಗಳಲ್ಲಿ ಲೆಕ್ಕಮಾಡಬಲ್ಲದಾಗಿತ್ತು. ಇದು ಮನುಷ್ಯನ ಮಧ್ಯಸ್ಥಿಕೆ ಇಲ್ಲದೆ ಅಂಕಗಣಿತ ಹಾಗೂ ತಾರ್ಕಿಕ ಲೆಕ್ಕಗಳನ್ನು ಮಾಡಬಲ್ಲದಾಗಿತ್ತು. ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಮುಖ್ಯವಾಗಿ ಆವಶ್ಯಕವಾಗಿರುವ ಸೂಚನೆಗಳನ್ನು ಸಂಗ್ರಹಿಸಿಡುವುದು ಇದರ ಯಶಸ್ಸಿಗೆ ಮುಖ್ಯ ಕಾರಣವಾಗಿತ್ತು.

ಈ ಯಂತ್ರವು ಕಂಪ್ಯೂಟರ್ ಯುಗದ ಪ್ರಾರಂಭಿಕ ಯಂತ್ರವಾಗಿ ಗುರುತಿಸಲ್ಪಟ್ಟಿದೆ.

UNIVAC - I (Universal Automatic Computer) :

1951 ರಲ್ಲಿ ಯುನೈಟೆಡ ಸ್ಟೇಟ್ಸ ನಲ್ಲಿ ಮೊದಲನೆ ವಾಣಿಜ್ಯ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಯಿತು. ಈ ಯಂತ್ರವು 25 ಫೂಟು X 50 ಫೂಟು ಉದ್ದವಾಗಿತ್ತು. , 5600 ಟ್ಯೂಬ್‌ಗಳನ್ನು, 18000 ಕ್ರಿಸ್ಟಲ್ ಡಯೊಡ್‌ಗಳನ್ನು ಹಾಗೂ 300 ರಿಲೇಗಳನ್ನು ಹೊಂದಿತ್ತು. ಈ ಯಂತ್ರವು ಸೀರಿಯಲ್ ವಿದ್ಯುನ್ಮಂಡಲವನ್ನು, 2.25 MHz ಬಿಟ್ ರೇಟನ್ನು ಹಾಗೂ 1000 ಶಬ್ದಗಳನ್ನು ಅಥವಾ 12000 ಅಕ್ಷರಗಳನ್ನು ಸಂಗ್ರಹಿಸಿಡುವಂಥ ಆಂತರಿಕ ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿತ್ತು.

UNIVAC ಯಂತ್ರವನ್ನು ಸಾಮಾನ್ಯ ಉದ್ದೇಶ ಕಂಪ್ಯೂಟರ್‌ನ್ನಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿತ್ತು. ಇದು ಮ್ಯಾಗ್ನೇಟಿಕ ಟೇಪ್ ಘಟಕವನ್ನು ಹೊಂದಿದ ಹಾಗೂ ಬಫರ್ ಮೆಮೋರಿ ಹೊಂದಿದ ಮೊದಲನೆ ಕಂಪ್ಯೂಟರ್ ಆಗಿತ್ತು.

ಮೊದಲನೇ ಪೀಳಿಗೆ ಕಂಪ್ಯೂಟರ್‌ನ ಮಿತಿಗಳು(Limitations of first generation computer)

ಮೊದಲನೇ ಪೀಳಿಗೆ ಕಂಪ್ಯೂಟರ್‌ನ ಮಿತಿಗಳು/ಕುಂದು ಕೊರತೆಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ:

- Vacuum Tube ಗಳನ್ನು ಮುಖ್ಯ ಘಟಕಗಳನ್ನಾಗಿ ಉಪಯೋಗಿಸಿರುವುದು.
- ಗಾತ್ರದಲ್ಲಿ ದೊಡ್ಡದಾಗಿದ್ದು, ಕಾರ್ಯವಿಧಾನವು ನಿಧಾನವಾಗಿತ್ತು. ಹಾಗೂ ಮಾಹಿತಿ ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ ಕಡಿಮೆಯಾಗಿತ್ತು.
- ಇವು ಸಾಕಷ್ಟು ವಿದ್ಯುತ್ತನ್ನು ಬಳಸುತ್ತಿದ್ದವು ಹಾಗೂ ಬಹಳ ಉಷ್ಣವನ್ನು ಉತ್ಪಾದಿಸುತ್ತಿದ್ದವು
- ಇವುಗಳ ಲೆಕ್ಕ ಮಾಡುವ ಸಾಮರ್ಥ್ಯ ಸೀಮಿತವಾಗಿತ್ತು
- ಇವುಗಳು ಅಷ್ಟೊಂದು ನಿಖರವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರಲಿಲ್ಲ ಹಾಗೂ ನಂಬಿಕೆಗೆ ಅರ್ಹವಾಗಿಯೂ ಇರಲಿಲ್ಲ.
- ಇವು progam ಗಳನ್ನು ಬರೆಯಲು ಯಾಂತ್ರಿಕ ಭಾಷೆ(Mahine Language) ನ್ನು ಉಪಯೋಗಿಸುತ್ತಿದ್ದವು.
- ಇವು ಬಹಳ ದುಬಾರಿಯಾಗಿದ್ದವು.

ಎರಡನೇಯ ಪೀಳಿಗೆ (Second Generation) 1955 - 1964

ಸಿ. ಪಿ. ಯು ನ ಘಟಕಗಳಿಗಾಗಿ ಟ್ರಾನ್ಸಿಸ್ಟರಗಳನ್ನು ಉಪಯೊಗಿಸಿ ಎರಡನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ತಯಾರಿಸಲಾಯಿತು.

ಫೆರೈಟ್ ಕೋರ್ (ferrite core) ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಮೇನ್ ಮೆಮೋರಿ ಹಾಗೂ ಮ್ಯಾಗ್ನೆಟಿಕ ಡಿಸ್ಕಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಸೆಕೆಂಡರಿ ಮೆಮೋರಿಗಳನ್ನು ತಯಾರಿಸಲಾಯಿತು.

© NIMI NOT TO BE REPUBLISHED

FORTRAN(1956), ALGOL(1960) & COBOL(1960-1961) ಇಂತಹ high level language ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಯಿತು. ಇನ್‌ಪುಟ್, ಔಟಪುಟ್ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು/ ನಿಯಂತ್ರಿಸಲು ಇನ್‌ಪುಟ್/ಔಟಪುಟ್ ಪ್ರೊಸೆಸರ್‌ನ್ನು ಅಳವಡಿಸಲಾಯಿತು.

ಉತ್ಪಾದನಾ ವೆಚ್ಚ ಬಹಳ ಕಡಿಮೆ ಇತ್ತು. ಈ ರೀತಿಯಿಂದ ಕಂಪ್ಯೂಟರ್‌ನ ಗಾತ್ರ ಬಹಳ ಕಡಿಮೆಯಾಗಿತ್ತು. ಎರಡನೇ ಪೀಳಿಗೆಯ ಅವಧಿಯಲ್ಲಿ ಸಿ.ಪಿ.ಯು. ಮೆಮೋರಿ, ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳು, ಇನ್‌ಪುಟ್, ಔಟಪುಟ್ ಘಟಕಗಳು ಅಭಿವೃದ್ಧಿ ಹೊಂದಿದವು. ಎರಡನೇ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಯಾವವೆಂದರೆ- IBM 1620, IBM 1401, CDC 3600

ವೈಶಿಷ್ಟ್ಯಗಳು (Features)

- Vacuum Tube ಗಳ ಬದಲಾಗಿ ಟ್ರಾನ್ಸಿಸ್ಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಯಿತು.
- ಮೊದಲನೇ ಪೀಳಿಗೆಗಿಂತ ಎರಡನೇ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಸಂಸ್ಕರಣಾ ವೇಗ ಹೆಚ್ಚಾಗಿತ್ತು(ಮೈಕ್ರೋ ಸೆಕೆಂಡುಗಳಷ್ಟು)
- ಗಾತ್ರ ಚಿಕ್ಕದಾಗಿತ್ತು (51 square feet)
- ಇನ್‌ಪುಟ್ ಹಾಗೂ ಔಟಪುಟ್ ಸಾಧನಗಳು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿದ್ದವು.

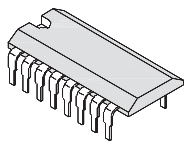
ಮೂರನೇ ಪೀಳಿಗೆ (Third Generation) (1964 - 1977)

300 ಟ್ರಾನ್ಸಿಸ್ಟರ್‌ಗಳ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿದ ಸಣ್ಣ ಚಿಪ್‌ಗಳನ್ನು ಮೂರನೇ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಯಿತು. ಈ ಇಂಟಿಗ್ರೇಟೆಡ್ ಸರ್ಕಿಟ್ (IC) ಗಳನ್ನು chip(ಚಿಪ್) ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಒಂದು IC ಬಹಳಷ್ಟು ಟ್ರಾನ್ಸಿಸ್ಟರ್‌ಗಳನ್ನು, ರೆಸಿಸ್ಟರ್‌ಗಳನ್ನು ಹಾಗೂ ಕೆಪ್ಯಾಸಿಟರ್‌ಗಳನ್ನು ಒಂದು ಸಣ್ಣ ತೆಳುವಾದ ಸಿಲಿಕಾನ್ ತುಂಡಿನ ಮೇಲೆ ನಿರ್ಮಾಣ ಮಾಡಲಾಗಿರುತ್ತದೆ. ಇದರಿಂದಾಗಿ ಕಂಪ್ಯೂಟರ್‌ನ ಗಾತ್ರ ಬಹಳಷ್ಟು ಚಿಕ್ಕದಾಯಿತು. ಮೂರನೇ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಯಾವವೆಂದರೆ IBM -360, ICL -1900, IBM -370, & VAX -750. ಹೈ ಲೆವೆಲ್ ಭಾಷೆಯಾದ BASIC (Bigginers All Purpose Symbolic Instruction Code) ಈ ಮೂರನೇ ಪೀಳಿಗೆಯಲ್ಲಿ ಅಭಿವೃದ್ಧಿ ಹೊಂದಿತು.

ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಗಾತ್ರ ಚಿಕ್ಕದಾಗಿತ್ತು. ಕಡಿಮೆ ಬೆಲೆಯುಳ್ಳದ್ದಾಗಿದ್ದವು , ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ ಹೆಚ್ಚಾಗಿತ್ತು. ಹಾಗೂ ಸಂಸ್ಕರಣಾ ವೇಗ ಹೆಚ್ಚಾಗಿತ್ತು. ಬಹಳ ಬೇಗನೇ IC ಗಳು, IC ಯಿಂದ LSI ಗೆ ಬದಲಾಯಿಸಲ್ಪಟ್ಟವು. ಇವು 100 ಘಟಕಗಳನ್ನು ಒಳಗೊಂಡಿದ್ದವು.

Fig 2



CP110112

100 ಘಟಕಗಳನ್ನು ಹೊಂದಿದಂತಹ IC ಗಳಿಗೆ LSI ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ವೈಶಿಷ್ಟ್ಯಗಳು(Features)

- ಟ್ರಾನ್ಸಿಸ್ಟರ್‌ಗಳ ಬದಲಾಗಿ ಇಂಟಿಗ್ರೇಟೆಡ ಸರ್ಕಿಟ್‌ಗಳ ಬಳಕೆ
- ಅರೆವಾಹಕ(Semiconductor) ಸಂಗ್ರಹಣಾ ಸಾಧನಗಳ ಬಳಕೆ
- ಗಾತ್ರ ಬಹಳ ಚಿಕ್ಕದಾಗಿದ್ದು, ಸಂಸ್ಕರಣಾ ವೇಗ ಹೆಚ್ಚಾಗಿತ್ತು. ಇವು ಹೆಚ್ಚು ನಿಖರವಾದ ಹಾಗೂ ವಿಶ್ವಾಸಾರ್ಹವಾಗಿದ್ದವು.
- LSI ಮತ್ತು VLSI ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಯಿತು.
- ಈ ಪೀಳಿಗೆಯಲ್ಲಿ ಮಿನಿ (mini computer) ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು.
- ಪ್ರೋಗ್ರಾಮಿಂಗ್‌ಗಾಗಿ high level language ನ್ನು ಉಪಯೋಗಿಸಲಾಯಿತು.

ನಾಲ್ಕನೇಯ ಪೀಳಿಗೆ (Fourth Generation) (1978 - ಇಂದಿನವರೆಗೆ)

100 ಘಟಕಗಳನ್ನು ಹೊಂದಿದಂತಹ IC ಗಳಿಗೆ LSI ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಅದೇ ರೀತಿ 1000 ಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು ಘಟಕಗಳನ್ನು ಹೊಂದಿದಂತಹ IC ಗಳಿಗೆ VLSI (Very Large Scale Integration) ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಒಂದೇ ಸಿಲಿಕಾನ್ ಚಿಪ್ ಮೇಲೆ ನಿರ್ಮಾಣಗೊಂಡ LSIC ಗಳನ್ನು ಮೈಕ್ರೋಪ್ರೊಸೆಸರ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಮೈಕ್ರೊಪ್ರೊಸೆಸರ್‌ಗಳ ಅಭಿವೃದ್ಧಿಯಿಂದ ಒಂದೇ ಚಿಪ್‌ನ ಮೇಲೆ .ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ನ ಸಿ.ಪಿ.ಯು ನ್ನು ಇಡಲಾಯಿತು. ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮೈಕ್ರೊಕಂಪ್ಯೂಟರ್‌ಗಳು ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ಆನಂತರ LSI ಗಳು VLSI ಗಳಿಂದ ಬದಲಾಯಿಸಲ್ಪಟ್ಟವು. ಈ ರೀತಿಯಿಂದ ಯಾವ .ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಒಂದು ದೊಡ್ಡ ಕೋಣೆಯನ್ನು ಆಕ್ರಮಿಸಿಕೊಳ್ಳುತ್ತಿದ್ದವೋ ಅವುಗಳನ್ನು ಈಗ ಒಂದು ಮೇಜಿನ ಮೇಲೆ ಇಡಬಹುದಾಗಿದೆ. ಈಗ ನೀವು ನಿಮ್ಮ ಶಾಲೆಗಳಲ್ಲಿ ನೋಡುತ್ತಿರುವ ವೈಯಕ್ತಿಕ ಕಂಪ್ಯೂಟರ್‌ಗಳು ನಾಲ್ಕನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಇವುಗಳ ಮೈನ್ ಮೆಮೊರಿಗೆ ಉಪಯೋಗಿಸಲ್ಪಟ್ಟ ಅರೆವಾಹಕಗಳು 4 ಮೆಗಾಬಿಟ್ ಅಳತೆಯದಾಗಿದೆ. ನಾಲ್ಕನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಯಾವವೆಂದರೆ IBM PC, Apple, Macinosh ಇತ್ಯಾದಿಗಳು.

© NIMI NOT TO BE REPUBLISHED

ಹಾರ್ಡ್‌ಡಿಸ್ಕನ್ನು ಸೆಕೆಂಡರಿ ಮೆಮೋರಿಯನ್ನಾಗಿ ಉಪಯೋಗಿಸಲಾಯಿತು. ಕೀಬೋರ್ಡ್‌ಗಳು, ಡಾಟ್ ಮ್ಯಾಟ್ರಿಕ್ಸ ಪ್ರಿಂಟರ್‌ಗಳು ಮುಂತಾದವುಗಳು ಅಭಿವೃದ್ಧಿಪಡಿಸಲ್ಪಟ್ಟವು. ಆಪರೆಟಿಂಗ ಸಿಸ್ಟಮ್ (O. S) MS DOS, UNIX, Apple`s Macintosh ಮುಂತಾದವು ಲಭ್ಯವಾದವು. Object Oriented Language ಗಳು, C++ ಇತ್ಯಾದಿಗಳು ಅಭಿವೃದ್ಧಿ ಹೊಂದಿದವು.

ವೈಶಿಷ್ಟ್ಯಗಳು (Features)

- ಈ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮೈಕ್ರೊಪ್ರೊಸೆಸರ್ (VLSI) ನ್ನು ಮುಖ್ಯ ಅಂಶವಾಗಿ ಉಪಯೋಗಿಸಿದವು.
- ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮೈಕ್ರೊಕಂಪ್ಯೂಟರ್ ಅಥವಾ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.
- ಇವುಗಳ ಗಾತ್ರ ಡೆಸ್ಕಟಾಪ್‌ನಿಂದ ಲ್ಯಾಪಟಾಪ್ ಅಥವಾ ಪಾಮ್‌ಟಾಪ್‌ಗಿಂತ ವಿಭಿನ್ನವಾಗಿತ್ತು.
- ಇವುಗಳ ಸಂಸ್ಕರಣಾ ವೇಗ ಅತೀ ಹೆಚ್ಚಾಗಿದೆ.ಇವು ಶೇಕಡಾ 100 ನಿಖರವಾಗಿವೆ.. ಇವು ವಿಶ್ವಾಸಾರ್ಹ, ಪರಿಶ್ರಮಿ, ಹಾಗೂ ಬಹುಮುಖಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಕಂಪ್ಯೂಟರ್‌ಗಳು.
- ಇವು ಬಹಳ ಮಾಹಿತಿ ಸಂಗ್ರಹಿಸಬಲ್ಲ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿವೆ.

ಐದನೇಯ ಪೀಳಿಗೆ (Fifth Generation) (ಪ್ರಸ್ತುತ ಮತ್ತು ಭವಿಷ್ಯದ ಕಂಪ್ಯೂಟರ್‌ಗಳು)

ಐದನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳು ULSI (Ultra Large Scale Integration) ನ ಚಿಪ್ ಗಳಾಗಿವೆ. ULSI ನಲ್ಲಿ ಲಕ್ಷಾಂತರ ಟ್ರಾನ್ಸಿಸ್ಟರ್‌ಗಳನ್ನು ಒಂದೇ IC ಮೇಲೆ ಸ್ಥಾಪಿಸಲಾಗಿರುತ್ತದೆ.

ಈ ಅವಧಿಯಲ್ಲಿ 64 ಬಿಟ್‌ನ ಮೈಕ್ರೊಪ್ರೊಸೆಸರ್‌ನ್ನು ಅಬಿ ವೃದ್ದಿಪಡಿಸಲಾಯಿತು.1 GB ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯವುಳ್ಳ ಮೆಮೊರಿ ಚಿಪ್‌ನ್ನು ಹಾಗೂ ಫ್ಲ್ಯಾಷ್ ಮೆಮೊರಿಗಳನ್ನು , 600 GB ಸಾಮರ್ಥ್ಯವುಳ್ಳ ಹಾರ್ಡ್‌ಡಿಸ್ಕನ್ನು ಹಾಗೂ 50 GB ಸಾಮರ್ಥ್ಯವುಳ್ಳ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕಗಳನ್ನು ಈ ಅವಧಿಯಲ್ಲಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಯಿತು.

ಕೃತಕ ಬುದ್ಧಿಮತ್ತೆ (Artificial Intellegence) ಆಧಾರಿತ ಐದನೇಯ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟಿಂಗ ಸಾಧನಗಳು ಇನ್ನೂ ಅಭಿವೃದ್ಧಿ ಹೊಂದುತ್ತಲಿವೆ. ಉದಾಹರಣೆಗೆ ಧ್ವನಿ ಗುರುತಿಸುವಿಕೆಯಂತಹ ಕೃತಕ ಬುದ್ಧಿಮತ್ತೆಯಂತಹ ಸಾಧನಗಳನ್ನು ನಾವು ಈಗ ಬಳಸುತ್ತಿದ್ದೇವೆ. ಕೃತಕ ಬುದ್ಧಿಮತ್ತೆ ಇದು ಕಂಪ್ಯೂಟರ್ ವಿಜ್ಞಾನದ ಒಂದು ಶಾಖೆಯಾಗಿದೆ ಹಾಗೂ ಇದು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಮನುಷ್ಯನಂತೆ ವರ್ತಿಸುವಂತೆ ಅಥವಾ ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತೆ ಮಾಡುತ್ತದೆ. 1956 ರಲ್ಲಿ ಮೆಸಾಚ್ಯುಸೆಟ್ಸ ಇನ್ಸಿಟ್ಯೂಟ್ ನ ಜಾನ್ ಮೆಕಾರ್ಥಿ ಎಂಬುವವರು ಈ ಪದವನ್ನು ಕಂಡುಹಿಡಿದರು . ಕೃತಕ ಬುದ್ಧಿಮತ್ತೆಯು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

- **ಆಟ ಆಡುವುದು:** ಚೆಸ್ ಮತ್ತು ಚೆಕರ್ಸನಂತಹ ಆಟಗಳನ್ನು ಆಡಲು ಕಂಪ್ಯೂಟರ್ ಪ್ರೋಗ್ರಾಮ ಬರೆಯುವುದು.
- **ತಜ್ಞ ವ್ಯವಸ್ಥೆ :** ಕೆಲ ನಿಜ ಜೀವನದ ಸಂದರ್ಭಗಳಲ್ಲಿ ನಿರ್ಧಾರ ತೆಗೆದುಕೊಳ್ಳಲು ಪ್ರೊಗ್ರಾಂ ಬರೆಯುವುದು.
- **ನೈಸರ್ಗಿಕ ಭಾಷೆ:** ಮನುಷ್ಯನ ನೈಸರ್ಗಿಕ ಭಾಷೆಗಳನ್ನು ಅರ್ಥಮಾಡಿಕೊಳ್ಳಲು ಪ್ರೊಗ್ರಾಂ ಬರೆಯುವುದು.
- **ನ್ಯೂರಲ್ ನೆಟ್‌ವರ್ಕ್ಸ :** ಪ್ರಾಣಿಗಳ ಮೆದುಳಿನಲ್ಲಿ ಸಂಭವಿಸುವ ಭೌತಿಕ ಸಂಪರ್ಕಗಳನ್ನು ಪುನರುತ್ಪಾದಿಸಲು ಪ್ರಯತ್ನಿಸುವ ಅನುಕರಣಾ ಬುದ್ಧಿಮತ್ತೆಯ ವ್ಯವಸ್ಥೆ
- **ರೊಬೊಟಿಕ್ಸ :** ಇತರೆ ಸಂವೇದನಾ ಪ್ರಚೋದಕಗಳಿಗೆ ನೋಡಿ, ಕೇಳಿ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಲು ಕಂಪ್ಯೂಟರ್ ಪ್ರೋಗ್ರಾಂ ಬರೆಯುವುದು.

Fig 3

ಕಂಪ್ಯೂಟರ್‌ಗಳ ವರ್ಗೀಕರಣ (Classification of Computers)

ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಈ ಕೆಳಗಿನ ಮಾನದಂಡಗಳ ಪ್ರಕಾರ ವರ್ಗೀಕರಿಸಲಾಗಿದೆ.

- ಕಾರ್ಯಾಚರಣೆಯ ತತ್ವ
- ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಶಕ್ತಿ, ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ ಹಾಗೂ ಬೆಲೆ
- ತಾಂತ್ರಿಕ ಅಭಿವೃದ್ಧಿ
- ಕಾರ್ಯಾಚರಣೆಯ ತತ್ವ
- ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್ (Analog Computer)
- ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್ (Digital Computer)
- ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್ (Hybrid Computer)

ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್ (Analog Computer)

ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ನಿರಂತರವಾಗಿ ಬದಲಾಗುವ ಭೌತಿಕ ಸಂಗತಿಗಳಾದ ವಿದ್ಯುತ್ , ತಾಪಮಾನ , ಒತ್ತಡ ಮುಂತಾದವುಗಳನ್ನು

ಟೇಬಲ್ -1

ಪೀಳಿಗೆ(Generation)	ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಕಾಂಪೋನೆಂಟ್(Electronic Component)	ಅನುಕೂಲತೆಗಳು(Advantages)	ಅನಾನುಕೂಲತೆಗಳು(Disadvantages)
ಮೊದಲನೇಯ ಪೀಳಿಗೆ	ವ್ಯಾಕ್ಯೂಮ್ ಟ್ಯೂಬ್	ಲೆಕ್ಕ ಮಾಡುವುದಕ್ಕೆ ಸಹಾಯ ಮಾಡುವುದು.	1.ಭಾರೀ ಗಾತ್ರ 2.ತುಂಬಾ ದುಬಾರಿ 3.ಕಡಿಮೆ ವೇಗ 4.ಕಡಿಮೆ ನಿಖರತೆ 5.ಕಡಿಮೆ ಸಂಗ್ರಹ ಸಾಮರ್ಥ್ಯ 6.ಹೆಚ್ಚು ವಿದ್ಯುತ್‌ನ ಅವಶ್ಯಕತೆ 7.ಹೆಚ್ಚು ಶಾಖ ಉತ್ಪಾದನೆ 8.ಹೆಚ್ಚು ವೈಫಲ್ಯ 9.ಮಷಿನ್ ಭಾಷೆಯ ಉಪಯೋಗ 10.ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಇಲ್ಲದಿರುವುದು.
ಎರಡನೇ ಪೀಳಿಗೆ	ಟ್ರಾನ್ಸಿಸ್ಟರ್	1.ಚಿಕ್ಕ ಗಾತ್ರ 2.ಕಡಿಮೆ ಬೆಲೆ 3.ಉತ್ತಮ ವೇಗ 4.ಕಡಿಮೆ ವಿದ್ಯುತ್ ಬಳಕೆ ಮತ್ತು ಕಡಿಮೆ ಶಾಖ ಉತ್ಪಾದನೆ 5.ಉತ್ತಮ ಸಂಗ್ರಾಹಣಾ ಸಾಮರ್ಥ್ಯ 6.ಉತ್ತಮ ನಿಖರತೆ ಮತ್ತು ಹೆಚ್ಚು ವಿಶ್ವಾಸಾರ್ಹತೆ.	1.ಹವಾನಿಯಂತ್ರಣಗಳ ಅವಶ್ಯಕತೆ 2.ನಿರಂತರ ನಿರ್ವಹಣೆ 3.ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಇಲ್ಲದಿರುವುದು. 4.ನಂತರದ ಹಂತದ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಅಸೆಬ್ಲಿ ಭಾಷೆಯನ್ನು ಉಪಯೋಗಿಸಲು ಪ್ರಾರಂಭಿಸಿದವು.
ಮೂರನೇಯ ಪೀಳಿಗೆ	ಇಂಟಿಗ್ರೇಟೆಡ್ ಸರ್ಕಿಟ್‌ಗಳು	ಮೊದಲನೇ ಹಾಗೂ ಎರಡನೇಯ ಪೀಳಿಗೆಗಳಿಗೆ ಹೋಲಿಸಿದರೆ ಎಲ್ಲ ರೀತಿಯಿಂದಲೂ ಉತ್ತಮವಾಗಿದ್ದವು. ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಹಾಗೂ ಹೈ ಲೆವೆಲ್ ಭಾಷೆಯನ್ನು ಉಪಯೋಗಿಸುತ್ತಿದ್ದವು	1.ತಯಾರಕರೊಂದಿಗಿನ ಪ್ರಾರಂಭಿಕ ಸಮಸ್ಯೆ 2.ಆಂತರಿಕ ಕಾರ್ಯವಿಧಾನದ ಬಗ್ಗೆ ಯಾವುದೇ ಮಾಹಿತಿ ಇಲ್ಲದಿರುವುದು.
ನಾಲ್ಕನೇಯ ಪೀಳಿಗೆ	ವಿ. ಎಲ್. ಎಸ್. ಐ. ಅಥವಾ ಮೈಕ್ರೋಪ್ರೊಸೆಸರ್	1. ಕಡಿಮೆ ವೆಚ್ಚ 2. ಅತ್ಯುತ್ತಮ ವೇಗ ಹಾಗೂ ವಿಶ್ವಾಸಾರ್ಹತೆ 3. ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮನುಷ್ಯನಿಗೆ ಹತ್ತಿರವಾದವು	1. ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಹೋಲಿಸಿದರೆ ಕಡಿಮೆ ಶಕ್ತಿಶಾಲಿಯಾಗಿದ್ದವು

© NIMI NOT TO BE REPUBLISHED

ಪೀಳಿಗೆ(Generation)	ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಕಾಂಪೋನೆಂಟ್(Electronic Component)	ಅನುಕೂಲತೆಗಳು(Advantages)	ಅನಾನುಕೂಲತೆಗಳು(Disadvantages)
ಐದನೇಯ ಪೀಳಿಗೆ	ಯು. ಎಲ್. ಎಸ್ . ಐ. ಅಥವಾ ಬಯೋ ಚಿಪ್‌ಗಳು	1.ಅತೀ ಅಗ್ಗ 2.ಅತ್ಯುತ್ತಮ ವೇಗ 3.ಅತೀ ಹೆಚ್ಚು ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ 4.ಅತ್ಯಾಧುನಿಕ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ 5.ಹೆಚ್ಚು ಬುದ್ಧಿಮತ್ತೆ ಹಾಗೂ ನಿರ್ಧಾರ ತೆಗೆದುಕೊಳ್ಳುವ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿರುವುದು	1. ಹೊಸ ಲೊ ಲೆವೆಲ್ ಭಾಷೆಗಳ ಅವಶ್ಯಕತೆ.

ಅಳೆಯುವ ಹಾಗೂ ಲೆಕ್ಕ ಮಾಡಲು ಉಪಯೋಗಿಸಲ್ಪಡುವ ದತ್ತಾಂಶಗಳನ್ನಾಗಿ ಪರಿವರ್ತಿಸುವ ಕಂಪ್ಯೂಟರ್‌ಗಳಾಗಿವೆ. ನಿರಂತರವಾಗಿ ಬದಲಾವಣೆ ಹೊಂದುವ ಸಂಗತಿಗಳೊಂದಿಗೆ ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸುವುದರಿಂದ ಇವು ಅಂದಾಜು ಫಲಿತಾಂಶಗಳನ್ನು ಮಾತ್ರ ನೀಡುತ್ತವೆ. ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಉತ್ತರವನ್ನು ಮಾಪನದ ಮೇಲೆ ಅಥವಾ ಸ್ಕೇಲ್ ಮೇಲೆ ಪ್ರದರ್ಶಿಸುತ್ತವೆ. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್ ಕಡಿಮೆ ಮೆಮೊರಿ ಹಾಗೂ ಕಡಿಮೆ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಹೊಂದಿದೆ. ಇವುಗಳನ್ನು ಇಂಜಿನಿಯರಿಂಗ ಹಾಗೂ ವೈಜ್ಞಾನಿಕ ಕಾರ್ಯಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

- ಥರ್ಮಾಮೀಟರ್
- ಸ್ಪೀಡೋಮೀಟರ್
- ಅನಾಲಾಗ್ ಗಡಿಯಾರ

ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್ (Digital Computer)

ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಡಿಜಿಟಲ್ ದತ್ತಾಂಶಗಳೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ. ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಬೈನರಿ ಸಂಖ್ಯಾ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸುತ್ತದೆ. ಬೈನರಿ ಸಂಖ್ಯಾ ವ್ಯವಸ್ಥೆಯು 0 ಮತ್ತು 1 ಎಂಬ ಎರಡು ಸಂಖ್ಯೆಗಳನ್ನು ಮಾತ್ರ ಒಳಗೊಂಡಿರುತ್ತದೆ. ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್ ಇದು ಡಿಜಿಟಲ್ ಸಿಗ್ನಲ್ ಗಳಲ್ಲಿ ದತ್ತಾಂಶವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ 0 ಆಫ್(OFF) ನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಹಾಗೂ 1 ಆನ್(ON) ನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್ ದತ್ತಾಂಶಗಳ ಮೇಲೆ ಅಂಕಗಣಿತ ಹಾಗೂ ತಾರ್ಕಿಕ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು ಡಿಜಿಟಲ್ ರೂಪದಲ್ಲಿ ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ.

ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಬಹಳ ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ. ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಉತ್ತರಗಳನ್ನು ಸಂಗ್ರಹಿಸಿ ಇಡಬಲ್ಲವು. ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ದೊಡ್ಡ ಮೆಮೊರಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. (ಮಾಹಿತಿ ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ). ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ಕಚೇರಿ ಹಾಗೂ ಮನೆಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಾಗಿವೆ.

ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತೆ ಈ ಕೆಳಗಿನ ಎರಡು ಗುಂಪುಗಳಂತೆ ವಿಂಗಡಿಸಲಾಗಿದೆ

- ವಿಶೇಷ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು (Special Purpose Computers)
- ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು (General Purpose Computers)

ವಿಶೇಷ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್ ಗಳು (Special Purpose Computers)

ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ನಿರ್ದಿಷ್ಟ ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಹರಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಿದ ಕಂಪ್ಯೂಟರ್‌ಗಳಾಗಿವೆ. ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಹಡಗುಗಳು ಹಾಗೂ ವಿಮಾನಗಳು ಇತ್ಯಾದಿಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು (General Purpose Computers)

ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ನಾನಾ ರೀತಿಯ ವಿವಿಧ ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಹರಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು ವಿಭಿನ್ನ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಸಂಗ್ರಹಿಸಿ ಅವುಗಳನ್ನು ಸಂಸ್ಕರಣೆ ಮಾಡಬಲ್ಲವು.

ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಹಾಗೂ ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳ ನಡುವಿನ ವ್ಯತ್ಯಾಸಗಳನ್ನು ಕೋಷ್ಟಕ 2 ರಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ.

ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು (Hybrid Computers) (**ಚಿತ್ರ** 4)

ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್ ಇದು ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್ ಹಾಗೂ ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಸಂಯೋಜನೆಯಾಗಿದೆ. ಹೈಬ್ರಿಡ್

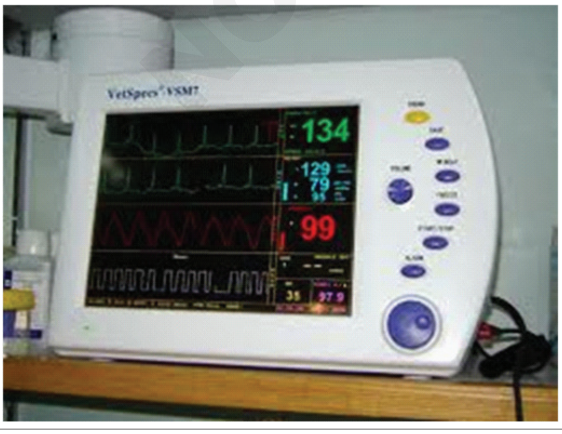
© NIMI NOT TO BE REPUBLISHED

ಟೇಬಲ್ 2

ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್
1. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ನಿರಂತರ ಪರಿಮಾಣಗಳ ಮೇಲೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ.	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ನಿರ್ದಿಷ್ಟವಾದ ಪರಿಮಾಣಗಳ ಮೇಲೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ.
2. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಡಿಮೆ ಮೆಮೊರಿಯನ್ನು ಹೊಂದಿವೆ	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ದೊಡ್ಡ ಮೆಮೊರಿಯನ್ನು ಹೊಂದಿವೆ.
3. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಡಿಮೆ ವೇಗವನ್ನು ಹೊಂದಿವೆ	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಅತೀ ವೇಗ ಹೊಂದಿವೆ.
4. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಡಿಮೆ ವಿಶ್ವಾಸಾರ್ಹವಾಗಿವೆ	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಹೆಚ್ಚು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿವೆ.
5. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಇಂಜಿನಿಯರಿಂಗ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಬಳಸಲಾಗುತ್ತದೆ.	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ವಿಜ್ಞಾನ , ವೈದ್ಯಕೀಯ ಹಾಗೂ ಜೀವನದ ಎಲ್ಲಾ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
6. ವೇಗ ಹಾಗೂ ತಾಪಮಾನಗಳಂತಹ ಅನಾಲಾಗ್ ಪರಿಮಾಣಗಳನ್ನು ಲೆಕ್ಕ ಅಥವಾ ಅಳತೆ ಮಾಡಲು ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.	ಗಣಿತ ಮತ್ತು ತಾರ್ಕಿಕ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಸಂಕಲನ, ವ್ಯವಕಲನ, ಗುಣಾಕಾರ, ಭಾಗಾಕಾರದಂತಹ ಲೆಕ್ಕಗಳನ್ನು ಮಾಡಲು ಹಾಗೂ ಇತರೆ ಗಣಿತ ಮತ್ತು ಸಂಖ್ಯಾಶಾಸ್ತ್ರೀಯ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಲು ಇವುಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
7. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಡಿಮೆ ನಿಖರವಾದ ಫಲಿತಾಂಶಗಳನ್ನು ನೀಡುತ್ತವೆ.	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಶೇಕಡಾ 100 ರಷ್ಟು ನಿಖರವಾದ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತವೆ.
8. ಸಾಮಾನ್ಯವಾಗಿ ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ವಿಶೇಷ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಆಗಿರುತ್ತವೆ.	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಆಗಿರುತ್ತವೆ.
9. ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಉದಾಹರಣೆಗಳು :ಥರ್ಮಾಮೀಟರ್, ಅನಾಲಾಗ್ ಗಡಿಯಾರ, ಸ್ಪೀಡೋಮೀಟರ್	ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಉದಾಹರಣೆಗಳು : ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್,ಲ್ಯಾಪ್ ಟಾಪ್, ಸ್ಮಾರ್ಟ್‌ಫೋನ್ ಇತ್ಯಾದಿಗಳು

ಕಂಪ್ಯೂಟರ್ ಅನಾಲಾಗ್ ಹಾಗೂ ಡಿಜಿಟಲ್ ದತ್ತಾಂಶಗಳನ್ನು ನಿಭಾಯಿಸಬಲ್ಲದಾಗಿದೆ. ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್ , ಅನಾಲಾಗ್ ಕಂಪ್ಯೂಟರ್ ಹಾಗೂ ಡಿಜಿಟಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಅತ್ಯುತ್ತಮ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಸಂಯೋಜನೆ ಮಾಡುತ್ತದೆ. ಅನಾಲಾಗ್ ಹಾಗೂ ಡಿಜಿಟಲ್ ಎರಡೂ ರೂಪದಲ್ಲಿ ಇದು ದತ್ತಾಂಶವನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು.

Fig 4



CP110114

ಉಪಯೋಗಗಳು (Applications)

ಆಸ್ಪತ್ರೆಗಳಲ್ಲಿ ರೋಗಿಯ ಹೃದಯ ಕಾರ್ಯನಿರ್ವಹಣೆ, ತಾಪಮಾನ, ರಕ್ತದ ಒತ್ತಡ ಮುಂತಾದವುಗಳನ್ನು ಪರೀಕ್ಷಿಸಲು ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಈ ಲೆಕ್ಕಗಳನ್ನು ಸಂಖ್ಯೆಗಳಿಗೆ ಪರಿವರ್ತಿಸಬಹುದಾಗಿದೆ ಹಾಗೂ ಡಿಜಿಟಲ್ ರೂಪದಲ್ಲಿ ತೋರಿಸಬಹುದಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ದಿ ವೈಟಲ್ ಸೈನ್ಸ ಮಾನಿಟರಿಂಗ್ ಘಟಕ (VSM).

ಇದು ಬ್ಲಡ ಪ್ರೆಷರ್ ಮಾನಿಟರ್, ECG ಮಾನಿಟರ್, ರೆಸ್ಪಿರೇಟರಿ ಮಾನಿಟರ್ ಹಾಗೂ ಅನಸ್ತೇಷಿಯ ಮಾನಿಟರಿಂಗ ಒಳಗೊಂಡಿರುತ್ತದೆ.

- ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಆಕಾಶ ನೌಕೆಗಳಲ್ಲಿ ಹಾಗೂ ಕ್ಷಿಪಣಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ವೈಜ್ಞಾನಿಕ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

- ಹೈಬ್ರಿಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಕೈಗಾರಿಕೆಗಳ ಕಾರ್ಯವಿಧಾನಗಳ ನಿಯಂತ್ರಣೆಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಕಂಪ್ಯೂಟಿಂಗ್ ಶಕ್ತಿ, ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯ ಹಾಗೂ ಬೆಲೆ ಇವುಗಳ ಆಧಾರದ ಮೇಲೆ ವರ್ಗೀಕರಿಸಲಾಗಿದೆ.

- ಮೈಕ್ರೊಕಂಪ್ಯೂಟರ್ (Micro computer)
- ಮಿನಿ ಕಂಪ್ಯೂಟರ್ (Mini Computer)
- ಮೈನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ (Mainframe computer)
- ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ (Super Computer)

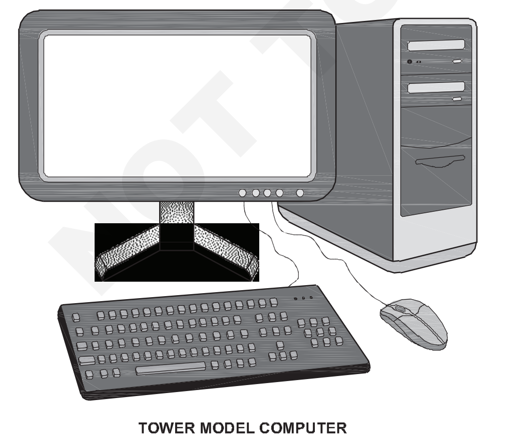
ಮೈಕ್ರೊಕಂಪ್ಯೂಟರ್ (Micro computer)

ಮೈಕ್ರೊಕಂಪ್ಯೂಟರ್‌ನ್ನು ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು 1970 ರಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು. ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಉದಾಹರಣೆಗಳು PC ಮತ್ತು Apple Macintosh. ಡೆಸ್ಕಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಪೋರ್ಟೇಬಲ್ ಕಂಪ್ಯೂಟರ್ ಇವು ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ನ ಮುಖ್ಯ ವಿಧಗಳು

ಡೆಸ್ಕಟಾಪ್ ಕಂಪ್ಯೂಟರ್(Desktop Computer)

ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮೇಜಿನ ಮೇಲೆ ಅಥವಾ ಡೆಸ್ಕಟಾಪ್ ಮೇಲೆ ಸುಲಭವಾಗಿ ಹೊಂದಿಕೊಳ್ಳುವುದರಿಂದ ಇವುಗಳಿಗೆ ಈ ಹೆಸರನ್ನು ಕೊಡಲಾಗಿದೆ. ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಎರಡು ರೀತಿಯ ಮಾದರಿ ಅಥವಾ ಕವಚಗಳಲ್ಲಿ ಕಂಡು ಬರುತ್ತವೆ. ಡೆಸ್ಕಟಾಪ್ ಮಾದರಿಯಲ್ಲಿ ಸಿಸ್ಟಮ್ ಯುನಿಟ್‌ನ್ನು ಡೆಸ್ಕಟಾಪ್ ಅಥವಾ ಮೇಜಿನ ಮೇಲೆ ಇಡಲಾಗುತ್ತದೆ. ಮಾನಿಟರ್ ನ್ನು ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ಮೇಲೆ ಇಡಲಾಗುತ್ತದೆ. ಟಾವರ್ ಮಾದರಿಯಲ್ಲಿ ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ಹಾಗೂ ಮಾನಿಟರ್‌ನ್ನು ಮೇಜಿನ ಮೇಲೆ ಇಡಲಾಗುತ್ತದೆ. Fig 5 ರಂತೆ.

Fig 5



TOWER MODEL COMPUTER

CP110115

ಪೋರ್ಟೆಬಲ್ ಕಂಪ್ಯೂಟರ್ (Portable Computer)

ಪೋರ್ಟೆಬಲ್ ಕಂಪ್ಯೂಟರ್ ಇದು ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಆಗಿದ್ದು ಇದನ್ನು ಒಂದು ಸ್ಥಳದಿಂದ ಇನ್ನೊಂದು ಸ್ಥಳಕ್ಕೆ ಸುಲಭವಾಗಿ ಕೊಂಡೊಯ್ಯಬಹುದು. ನೋಟ್‌ಬುಕ್ ಕಂಪ್ಯೂಟರ್ ಹಾಗೂ ಹ್ಯಾಂಡಹೆಲ್ಡ ಕಂಪ್ಯೂಟರ್(ಸ್ಮಾರ್ಟಫೋನ್) ಇವು ಪೋರ್ಟೆಬಲ್ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಉದಾಹರಣೆಗಳಾಗಿವೆ. ನೋಟ್ ಬುಕ್ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಲ್ಯಾಪ್‌ಟಾಪ್ ಎಂದೂ ಕರೆಯಲಾಗುತ್ತದೆ. ಲ್ಯಾಪ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಗಾತ್ರ ಬಹಳ ಚಿಕ್ಕದಾಗಿದ್ದು , ಇದನ್ನು ತೊಡೆಯ ಮೇಲೆ ಸುಲಭವಾಗಿ ಇಡಬಹುದು.

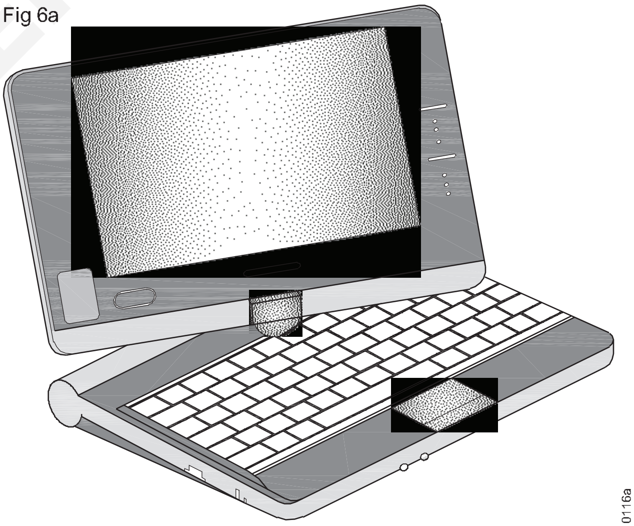
ಲ್ಯಾಪ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಅಥವಾ ನೋಟ್‌ಬುಕ್ ಕಂಪ್ಯೂಟರ್ (Laptop Computer or Notebook Computer)

ಚಿತ್ರ 6a ಮತ್ತು 6b ನಲ್ಲಿ ತೋರಿಸಿದಂತೆ ಕ್ರಮವಾಗಿ ಲ್ಯಾಪ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಅಥವಾ ನೋಟ್‌ಬುಕ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಇರುತ್ತವೆ. ಇದನ್ನು ಟಾವರ್ ಮಾಡೆಲ್ ಕಂಪ್ಯೂಟರ್ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ.

ಪಾಮ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್/ಡಿಜಿಟಲ್ ಡೈರಿ/ ನೋಟ್‌ಬುಕ್/ಪಿ.ಡಿ. ಎ:

ಹ್ಯಾಂಡಹೆಲ್ಡ ಕಂಪ್ಯೂಟರ್ ಸ್ಮಾರ್ಟಫೋನ್‌ನಂತೆ ಚಿಕ್ಕದಾಗಿರುತ್ತದೆ. (Portable). ಹ್ಯಾಂಡಹೆಲ್ಡ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಪಾಮ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಪಾಮ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕೀಬೋರ್ಡನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ. ಆದರೆ ಅದರ ಪರದೆಯು ಇನ್‌ಪುಟ್ ಹಾಗೂ ಔಟಪುಟ್ ಎರಡೂ ಸಾಧನಗಳಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ. ಇದು ಸುಲಭವಾಗಿ ಬಳಸುವವರ ಕೈಯಲ್ಲಿ ಹೊಂದಿಕೊಳ್ಳುತ್ತದೆ.

Fig 6a



LAP TOP COMPUTER OR NOTE BOOK COMPUTER

CP110116a

ಮೈಕ್ರೋ ಕಂಪ್ಯೂಟರ್‌ನ ಉಪಯೋಗಗಳು (Uses of Micro Computer)

ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಕಚೇರಿಗಳಲ್ಲಿ ಅತಿ ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಈಗ ಇದನ್ನು ಅನೇಕ ಮನೆಗಳಲ್ಲಿ ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇವುಗಳನ್ನು ವ್ಯಾಪಾರ ಹಾಗೂ ಇಂಜಿನೀಯರಿಂಗ್ ಕ್ಷೇತ್ರಗಳಲ್ಲಿಯೂ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

©NIMI NOT TO BE REPUBLISHED

Fig 6

ಮಿನಿ ಕಂಪ್ಯೂಟರ್ (Mini Computer)

ಮಿನಿ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು 1960 ರಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು. ಮಿನಿ ಕಂಪ್ಯೂಟರ್ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಿಂತ ಗಾತ್ರದಲ್ಲಿ ದೊಡ್ಡದಾಗಿದೆ ಮತ್ತು ಹೆಚ್ಚು ಶಕ್ತಿಶಾಲಿಯಾಗಿದೆ. ಇದು ಪ್ರತಿ ಸೆಕೆಂಡಿಗೆ ಐದು ಮಿಲಿಯನ್ ಸೂಚನೆಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಎರಡು ಅಥವಾ ಹೆಚ್ಚು ಸಂಸ್ಕರಣಾ ಘಟಕಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಮಿನಿ ಕಂಪ್ಯೂಟರ್ ಏಕಕಾಲದಲ್ಲಿ 4000 ವರೆಗಿನ ಸಂಪರ್ಕಿತ ಬಳಕೆದಾರರಿಗೆ ಸೇವೆ ಸಲ್ಲಿಸಬಹುದು. ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಬಳಕೆದಾರರು ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಅಥವಾ ಟರ್ಮಿನಲ್ ಮುಖಾಂತರ ಉಪಯೋಗಿಸಬಹುದಾಗಿದೆ. ಮಾನಿಟರ್ ಮತ್ತು ಕೀಬೋರ್ಡ್ ಹೊಂದಿರುವ ಸಾಧನಕ್ಕೆ ಟರ್ಮಿನಲ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು ಡಂಬ್ ಟರ್ಮಿನಲ್ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಇದು ಯಾವುದೇ ಸಂಸ್ಕರಣಾ ಶಕ್ತಿಯನ್ನು ಹೊಂದಿಲ್ಲ ಹಾಗೂ ಏಕಾಂಗಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಕೆಲವು ಮಿನಿ ಕಂಪ್ಯೂಟರ್ ಮಾದರಿಗಳು VAX -8800 , AS400

ಮಿನಿ ಕಂಪ್ಯೂಟರ್‌ನ ಉಪಯೋಗಗಳು (Uses of Mini Computer)

ಮಿನಿ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಹೆಚ್ಚಾಗಿ ಸಣ್ಣ ಹಾಗೂ ಮಧ್ಯಮ ಗಾತ್ರದ ಕಂಪನಿಗಳು ಕೇಂದ್ರೀಕೃತ ಮಾಹಿತಿ ಸಂಗ್ರಹವನ್ನು ಒದಗಿಸುವುದಕ್ಕಾಗಿ ಉಪಯೋಗಿಸುತ್ತವೆ.

ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ (Mainframe Computer)

ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು 1975ರಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು. ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ ಗಾತ್ರದಲ್ಲಿ ಬಹಳ ದೊಡ್ಡ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಒಂದೇ ಸಮಯದಲ್ಲಿ ಅನೇಕ ಬಳಕೆದಾರರು ಅನೇಕ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಇದನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಒಂದೇ ಸಮಯದಲ್ಲಿ 50,000 ಬಳಕೆದಾರರು ಉಪಯೋಗಿಸಬಹುದು. ಟರ್ಮಿನಲ್ ಅಥವಾ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್ ಮುಖಾಂತರವಾಗಿ ಬಳಕೆದಾರರು ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಒಂದು ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ ಪ್ರತಿ ಸೆಕೆಂಡಿಗೆ 16 ಮಿಲಿಯನ್ ಸೂಚನೆಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು. ಕೆಲವು ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ ಮಾದರಿಗಳು

- NEC 610
- DEC 10

ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ನ ಉಪಯೋಗಗಳು (Uses of Mainframe Computer)

ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಪ್ರಮುಖವಾಗಿ ಸಂಸ್ಥೆಗಳು ಹಾಗೂ ಸರ್ಕಾರಿ ಸಂಸ್ಥೆಗಳು ನಿರ್ಣಾಯಕ ಕೆಲಸಗಳಿಗಾಗಿ ಮತ್ತು ಜನಗಣತಿಯಂತಹ ಬೃಹತ್ ದತ್ತಾಂಶ ಸಂಸ್ಕರಣೆ , ಉದ್ಯಮ ಮತ್ತು ಗ್ರಾಹಕರ ಅಂಕಿ ಅಂಶಗಳು ಮತ್ತು ವಹಿವಾಟು ಕಾರ್ಯನಿರ್ವಹಣೆಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ (Super Computer)

ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು 1980 ರಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ಅತೀ ವೇಗವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದರ ಗಾತ್ರ ಅತೀ ದೊಡ್ಡದಾಗಿದೆ ಹಾಗೂ ಉಳಿದ ಎಲ್ಲ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗಿಂತ ಅತ್ಯಂತ ದುಬಾರಿಯಾದ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಇದು ಅತ್ಯಂತ ಅತ್ಯಾಧುನಿಕ, ಸಂಕೀರ್ಣ ಸುಧಾರಿತ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಇದು ಬೃಹತ್ ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿದೆ. ಇದು ಪ್ರತಿ ಸೆಕೆಂಡಿಗೆ ಟ್ರಿಲಿಯನ್‌ನಷ್ಟು ಸೂಚನೆಗಳನ್ನು ನಿರ್ವಹಿಸಬಲ್ಲದು. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ಈಗಿನ ಅತೀ ವೇಗದ ಹಾಗೂ ಅತ್ಯಂತ ಶಕ್ತಿಶಾಲಿಯಾದ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಅತ್ಯಂತ ದುಬಾರಿಯಾಗಿರುತ್ತವೆ. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಅತ್ಯಂತ ತೀವ್ರವಾದ ಲೆಕ್ಕಾಚಾರಗಳಂತಹ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಅಪಾರ ಪ್ರಮಾಣದ ಗಣಿತದ ಲೆಕ್ಕಾಚಾರಗಳನ್ನು ಮಾಡಲು ಅವಶ್ಯಕವಿರುವ ವಿಶೇಶ ಅಪ್ಲಿಕೇಶನ್‌ಗಳಿಗಾಗಿ ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.

ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಉಪಯೋಗಗಳು (Uses of Super Computer)

- ಹವಾಮಾನ ಮುನ್ಸೂಚನೆ
- ಹಾಲಿವುಡ್ ಚಲನಚಿತ್ರದಲ್ಲಿರುವಂತೆ ಅನಿಮೇಟೆಡ್ ಗ್ರಾಫಿಕ್ಸ
- ಫ್ಲುಯಿಡ್ ಡೈನಾಮಿಕ್ ಲೆಕ್ಕಾಚಾರಗಳು
- ಪರಮಾಣು ಶಕ್ತಿ ಸಂಶೋಧನೆ
- ಬಾಹ್ಯಾಕಾಶ ವಿಜ್ಞಾನ

© NIMI NOT TO BE REPUBLISHED

- ಶಸ್ತ್ರಾಸ್ತ ಮತ್ತು ಕ್ಷಿಪಣಿ ವಿನ್ಯಾಸ
- ಪೆಟ್ರೋಲಿಯಂ ಪರಿಶೋಧನೆ ಇತ್ಯಾದಿ

ಈ ದಿನಗಳಲ್ಲಿ ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಕ್ರೇ, ಐ ಬಿ ಎಂ, ಹೆವ್ಲೆಟ್ ಪ್ಯಾಕರ್ಡ್ ನಂತಹ ಕಂಪನಿಗಳು ತಯಾರಿಸುತ್ತಿವೆ. ಅಕ್ಟೋಬರ 2010 ರಿಂದ ಟಿಯಾನ್ಹೆ - 1A ಜಗತ್ತಿನ ಅತೀ ವೇಗವಾದ ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ಆಗಿದೆ. ಇದು ಚೀನಾದಲ್ಲಿದೆ.

ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ಹಾಗೂ ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ ನಡುವಿನ ಮುಖ್ಯ ವ್ಯತ್ಯಾಸ ಎಂದರೆ ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ತನ್ನೆಲ್ಲಾ ಶಕ್ತಿಯನ್ನು ಒಂದೆ ಪ್ರೋಗ್ರಾಂ ನ್ನು ಅತಿ ವೇಗವಾಗಿ ಕಾರ್ಯಗತಗೊಳಿಸಲು ಉಪಯೊಗಿಸುತ್ತದೆ. ಆದರೆ ಮೇನ್ ಫ್ರೇಮ್ ಕಂಪ್ಯೂಟರ್ ತನ್ನ ಶಕ್ತಿಯನ್ನು ಏಕಕಾಲದಲ್ಲಿ ಅನೇಕ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೊಗಿಸುತ್ತದೆ. ಅಧುನಿಕ ತನ್ನ ಶಕ್ತಿಯನ್ನು ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಸಾವಿರಾರು ಮೈಕ್ರೋಪ್ರೊಸೆಸರ್ ಗಳನ್ನು ಒಳಗೊಂಡಿವೆ. ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಆನ್‌ಲೈನ ಸಂಸ್ಕರಣೆಗಾಗಿ ಸ್ಯಾಟಲೈಟ್ ನಂತಹ ಹೆಚ್ಚಿನ ವೇಗದ ಸೌಲಭ್ಯಗಳನ್ನು ಉಪಯೊಗಿಸುತ್ತದೆ.

ಕೆಲವು ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಮಾದರಿಗಳು - CRAY - XP, ETA 10 Param ಮತ್ತು Deep Blue

ಕಂಪ್ಯೂಟರ್‌ನ ಅನುಕೂಲಗಳು (Advantages of Computer)

- ಕಂಪ್ಯೂಟರ್ ಅತೀ ಹೆಚ್ಚಿನ ವಿಶ್ವಾಸಾರ್ಹತೆಯೊಂದಿಗೆ ಅತೀ ಹೆಚ್ಚಿನ ಸಂಸ್ಕರಣಾ ವೇಗವನ್ನು ಹೊಂದಿದೆ.
- ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದ ಮಾಹಿತಿಯನ್ನು ಮೆಮೋರಿಯಲ್ಲಿ ಸಂಗ್ರಹಿಸಿಡಬಹುದು ಹಾಗೂ ಯಾವುದೇ ನಿರ್ದಿಷ್ಟ ದತ್ತಾಂಶ / ಪ್ರೋಗ್ರಾಂ ನ್ನು ತಕ್ಷಣ ಪಡೆಯಬಹುದು.
- ಅತೀ ವೇಗವಾಗಿ ಯಾವುದೆ ಕ್ಲಿಷ್ಟಕರ ಸಮಸ್ಯೆಗೆ ಪರಿಹಾರ ಕೊಡುತ್ತದೆ.
- ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದ ದತ್ತಾಂಶವನ್ನು ಸಂಸ್ಕರಣೆ ಮಾಡುವುದರಿಂದ ಹೆಚ್ಚಿನ ಉದ್ಯೋಗಿಗಳ ಕೆಲಸವನ್ನು ಕಡಿಮೆ ಮಾಡಿ ಸಂಸ್ಕರಣಾ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.
- ಕುಲುಮೆ, ವಿಷಕಾರಿ ವಾತಾವರಣ , ನಿರ್ವಾತ ಮಾನವರಹಿತ ಉಪಗ್ರಹಗಳು ಮುಂತಾದ ಮಾನವನ ಉಪಸ್ಥಿತಿ ಇಲ್ಲದ ವಾತಾವರಣಗಳಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಸಮರ್ಥವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ.

ಕಂಪ್ಯೂಟರ್‌ನ ಮಿತಿಗಳು (Limitations of Computers)

- ಆರಂಭಿಕ ವೆಚ್ಚ ಹೆಚ್ಚಾಗಿರುವುದು.
- ಇನ್‌ಪುಟ್ ಮಾಹಿತಿಯನ್ನು ಸ್ಟೇಟ್‌ಮೆಂಟ್ ರೂಪದ ಪ್ರೋಗ್ರಾಂಗಳಲ್ಲಿ ತಯಾರಿಸಬೇಕಾಗುತ್ತದೆ. ಇದಕ್ಕಾಗಿ ಸಾಕಷ್ಟು ಪ್ರಮಾಣದ ಜ್ಞಾನದ ಅವಶ್ಯಕತೆ ಇದೆ.
- ಭಾರಿ ಪ್ರಮಾಣದ ದತ್ತಾಂಶವನ್ನು ಸಂಸ್ಕರಣೆ ಮಾಡುವಾಗ ಹಾಗೂ ಅದರ ಪುನರಾವರ್ತನೆಯ ಅವಶ್ಯಕತೆ ಇದ್ದಾಗ ಮಾತ್ರ ಕಂಪ್ಯೂಟರ್ ಬಳಕೆ ಆರ್ಥಿಕವಾಗಿ ಸರಿಯಾಗಿರುತ್ತದೆ.
- ಇದು ಕೇವಲ ಯಂತ್ರವಾಗಿದ್ದು, ದೋಷಗಳನ್ನು ತನ್ನಷ್ಟಕ್ಕೆ ತಾನು ಸರಿಪಡಿಸುವುದಿಲ್ಲ.

ಕಂಪ್ಯೂಟರ್ ನ ಕಾರ್ಯಗಳು (Functions of Computers)

ಎಲ್ಲ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಈ ಕೆಳಗಿನ ಮೂಲಘಟಕಗಳಿಂದ ತಯಾರಿಸಲ್ಪಟ್ಟಿವೆ. ಅವು ಈ ಕೆಳಗಿನಂತಿವೆ :-ಚಿತ್ರ (7) ರಂತೆ

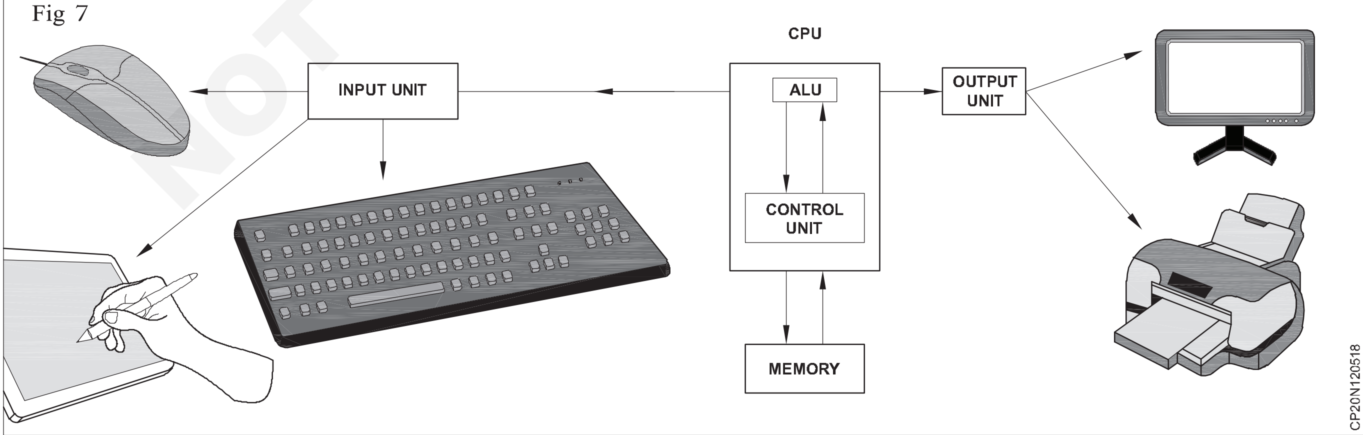
1 ಇನ್‌ಪುಟ್ ಘಟಕ (Input Unit)

2 ಕೇಂದ್ರ ಸಂಸ್ಕರಣಾ ಘಟಕ (Central Processing Unit(CPU))

a) ಅಂಕಗಣಿತ ತರ್ಕ ಘಟಕ(Arithmetic Logic Unit) ALU

b) ನಿಯಂತ್ರಣ ಘಟಕ (Control Unit) CU

Fig 7



NOT TO BE REPUBLISHED © NIMI

3. ಮೆಮೋರಿ (Memory)

4. ಔಟ್‌ಪುಟ್ ಘಟಕ (Output Unit)

ಇನ್‌ಪುಟ್ ಘಟಕ (Input Unit)

ಯಾವುದೇ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಕಂಪ್ಯೂಟರ್‌ಗಳು ದತ್ತಾಂಶ ಹಾಗೂ ಮಾಹಿತಿಯನ್ನು ಕ್ರಮವಾಗಿ ಸ್ವೀಕರಿಸಬೇಕು. ಅದಕ್ಕಾಗಿ ದತ್ತಾಂಶ ಹಾಗೂ ಸೂಚನೆಗಳನ್ನು ಕಂಪ್ಯೂಟರ್‌ಗೆ ಕೊಡುವ ಅವಶ್ಯಕತೆಯಿದೆ. ಇನ್‌ಪುಟ್ ಘಟಕವು ಒಂದು ಅಥವಾ ಹೆಚ್ಚಿನ ಇನ್‌ಪುಟ್ ಸಾಧನಗಳನ್ನು ಹೊಂದಿದೆ. ಕೀಬೋರ್ಡ ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಸಾಧನಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಕೆಲವು ಇನ್‌ಪುಟ್ ಸಾಧನಗಳನ್ನು ಚಿತ್ರ 9 ರಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ.

ಇನ್‌ಪುಟ್ ಸಾಧನಗಳು ಈ ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತವೆ.

- ಹೊರಗಿನಿಂದ ದತ್ತಾಂಶ ಹಾಗೂ ಸೂಚನೆಗಳನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ
- ಕಂಪ್ಯೂಟರ್‌ಗೆ ಅರ್ಥವಾಗುವ ರೂಪಕ್ಕೆ ಪರಿವರ್ತಿಸುತ್ತದೆ.
- ಪರಿವರ್ತಿಸಿದ ಮಾಹಿತಿಯನ್ನು ಮುಂದಿನ ಪ್ರಕ್ರಿಯೆಗಾಗಿ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ಗೆ ಸಲ್ಲಿಸುತ್ತದೆ

ಕೇಂದ್ರ ಸಂಸ್ಕರಣಾ ಘಟಕ (Central Processing Unit(CPU))

ಕೇಂದ್ರ ಸಂಸ್ಕರಣಾ ಘಟಕ (Central Processing Unit(CPU)) ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮೆದುಳು ಆಗಿರುತ್ತದೆ. ಚಿತ್ರ 8 ರಂತೆ. ವೈಯಕ್ತಿಕ ಕಂಪ್ಯೂಟರ್ ಸಿ ಪಿ ಯು ಸಾಮಾನ್ಯವಾಗಿ ಒಂದೇ ಚಿಪ್ ಆಗಿರುತ್ತದೆ. ಬಳಕೆದಾರರಿಂದ ಅಥವಾ ಸಾಫ್ಟವೇರ್‌ನಿಂದ ನೀಡಲ್ಪಟ್ಟ ಸೂಚನೆಗಳನ್ನು ಆಯೋಜಿಸುತ್ತದೆ ಹಾಗೂ ನಿರ್ವಹಿಸುತ್ತದೆ. ಪ್ರೊಸೆಸರ್ ಅನೇಕ ಘಟಕಗಳಿಂದ ಮಾಡಲ್ಪಟ್ಟಿದೆ.

Fig 8

CP110118

ಸಿಪಿಯು ಈ ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

- ಇದು ಎಲ್ಲ ಲೆಕ್ಕಗಳನ್ನು ಮಾಡುತ್ತದೆ.
- ಇದು ಎಲ್ಲ ನಿರ್ಧಾರಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ
- ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಎಲ್ಲ ಘಟಕಗಳನ್ನು ನಿಯಂತ್ರಿಸುತ್ತದೆ.

ಸಿ ಪಿ ಯು ನ ಎರಡು ವಿಶಿಷ್ಟ ಘಟಕಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

ಅಂಕಗಣಿತ ತರ್ಕ ಘಟಕ (ALU) ಇದು ಅಂಕಗಣಿತ ಮತ್ತು ತಾರ್ಕಿಕ ಕಾರ್ಯಗಳನ್ನು ಮಾಡುತ್ತದೆ.

ನಿಯಂತ್ರಣ ಘಟಕ (CU) ಮೆಮೊರಿಯಿಂದ ಸೂಚನೆಗಳನ್ನು ತೆಗೆದುಕೊಂಡು , ಡಿಕೋಡ್ ಮಾಡಿ ಅವುಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ. ಹಾಗೂ ಅವಶ್ಯಕತೆಯಿದ್ದಲ್ಲಿ ALU ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಮೆಮೊರಿ (Memory)

ಕಂಪ್ಯೂಟರ್ ಅಥವಾ ಇತರ ಡಿಜಿಟಲ್ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಸಾಧನಗಳಲ್ಲಿ ತಾತ್ಕಾಲಿಕವಾಗಿ ಅಥವಾ ಶಾಶ್ವತವಾಗಿ ಮಾಹಿತಿ ಅಥವಾ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಉಪಯೋಗಿಸುವ ಭೌತಿಕ ಸಾಧನಕ್ಕೆ ಮೆಮೊರಿ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಎರಡು ರೀತಿಯ ಮೆಮೊರಿಗಳಿವೆ.

- ಪ್ರೈಮರಿ ಮೆಮೊರಿ
- ಸೆಕೆಂಡರಿ ಮೆಮೊರಿ

ಔಟಪುಟ್ ಘಟಕ (Output Unit)

ಔಟಪುಟ್ ಘಟಕವು ಸಂಸ್ಕರಣಾ ಘಟಕದಿಂದ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯುತ್ತದೆ ಹಾಗೂ ಮನುಷ್ಯನಿಗೆ ಅರ್ಥವಾಗುವ ರೂಪದಲ್ಲಿ ಉತ್ತರ ನೀಡುತ್ತದೆ.

ಔಟಪುಟ್ ಸಾಧನಗಳು (Output Devices)

ಕೆಲವು ಔಟಪುಟ್ ಸಾಧನಗಳು

- ಮಾನಿಟರ್
- ಪ್ರಿಂಟರ್
- ಪ್ಲಾಟರ್
- ಸ್ಪೀಕರ್

ಕಂಪ್ಯೂಟರ್ ನ ಉಪಯೋಗಗಳು (Applications of Computers)

ವಿಜ್ಞಾನ (Science): ವಿಜ್ಞಾನಿಗಳು ಸಿದ್ಧಾಂತಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿ , ದತ್ತಾಂಶಗಳನ್ನು ವಿಶ್ಲೇಷಿಸಲು ಮತ್ತು ಪರೀಕ್ಷಿಸಲು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತಿದ್ದಾರೆ. ಕಂಪ್ಯೂಟರ್‌ನ ಅತೀ ವೇಗ ಮತ್ತು ನಿಖರತೆ ವಿವಿಧ ವೈಜ್ಞಾನಿಕ ವಿಶ್ಲೇಷಣೆಗಳನ್ನು ನಡೆಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಭೂಕಂಪಗಳು ಕಟ್ಟಡಗಳ ಮೇಲೆ ಹೇಗೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ ಹಾಗೂ ಮಾಲಿನ್ಯವು ವಾತಾವರಣದ ಮೇಲೆ ಹೇಗೆ ಪರಿಣಾಮ ಬೀರುತ್ತದೆ ಎಂದು ವಿವರವಾದ ಅಧ್ಯಯನ ಸೃಷ್ಟಿಸಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್ ಬಳಸದೆ ಉಪಗ್ರಹ ಆಧಾರಿತ ಅಪ್ಲಿಕೇಶನ್‌ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಇದಲ್ಲದೆ ಕಂಪ್ಯೂಟರ್ ಇಲ್ಲದೆ ಸೌರಮಂಡಲ ಹಾಗೂ ಬ್ರಹ್ಮಾಂಡದ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯುವುದು ಸಾಧ್ಯವಿಲ್ಲ.

© NIMI NOT TO BE REPUBLISHED

ಶಿಕ್ಷಣ (Education) : ಕಂಪ್ಯೂಟರ್‌ಗಳು ಶಿಕ್ಷಣದ ಸಂಪೂರ್ಣ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ಕ್ರಾಂತಿ ಉಂಟು ಮಾಡಿವೆ. ಪ್ರಸ್ತುತ ಶಿಕ್ಷಣವನ್ನು ಹೆಚ್ಚು ಆಸಕ್ತಿಕರವಾಗಿಸಲು ತರಗತಿ ಕೊಠಡಿಗಳು , ಗ್ರಂಥಾಲಯಗಳು, ಹಾಗೂ ವಸ್ತು ಸಂಗ್ರಹಾಲಯಗಳು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಸಮರ್ಥವಾಗಿ ಉಪಯೋಗಿಸಿಕೊಳ್ಳುತ್ತಿವೆ. ರೆಕಾರ್ಡ ಮಾಡಿದ ದೂರದರ್ಶನ ಕಾರ್ಯಕ್ರಮಗಳಿಗಿಂತ ವಿಭಿನ್ನವಾಗಿ ಕಂಪ್ಯೂಟರ್ ಏಡೆಡ್ ಎಜುಕೇಶನ್(ಸಿಎಇ) ಹಾಗೂ ಕಂಪ್ಯೂಟರ್ ಬೆಸಡ ಟ್ರೇನಿಂಗ(ಸಿಬಿಟಿ) ಇವು ಕಲಿಕೆಯನ್ನು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯನಾಗಿಸುತ್ತದೆ.

ಔಷಧ ಮತ್ತು ಆರೋಗ್ಯ ರಕ್ಷಣೆ (Medicine and Health Care) : ವೈದ್ಯಕೀಯ ಕ್ಷೇತ್ರದಲ್ಲಿ ಕಂಪ್ಯೂಟರ್ ಬಳಕೆ ಹೆಚ್ಚಾಗುತ್ತಿದೆ. ಈಗ ವೈದ್ಯರು ಶಸ್ತ್ರಚಿಕಿತ್ಸೆಯ ಸಂದರ್ಭದಲ್ಲಿ , ರೋಗಿಯ ರೋಗವನ್ನು ಪತ್ತೆಹಚ್ಚುವುದರಿಂದ ಹಿಡಿದು ರೋಗಿಯ ಸ್ಥಿತಿಯನ್ನು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡಲು ಕಂಪ್ಯೂಟರ್ ನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ. ಅಟೋಮೇಟೆಡ್ ಇಮೇಜಿಂಗ ತಂತ್ರಗಳನ್ನು ಉಪಯೋಗಿಸಿ ವೈದ್ಯರು ವ್ಯಕ್ತಿಯ ದೇಹದ ಒಳಗೆ ನೋಡಿ ಪ್ರತಿ ಅಂಗದ ಬಗ್ಗೆ ವಿವರವಾಗಿ ಅಧ್ಯಯನ ಮಾಡಬಹುದಾಗಿದೆ(ಉದಾ: ಸಿಟಿ ಸ್ಕ್ಯಾನ್ ಮತ್ತು ಎಂ ಆರ್ ಐ ಸ್ಕ್ಯಾನ್) . ಹಾಗೂ ಇದು ಈ ಹಿಂದೆ ಸಾಧ್ಯವಿರಲಿಲ್ಲ. ವಿಶೇಷ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಕಿವುಡ ಜನರಿಗೆ ಕಿವಿ ಕೇಳಲು ಸಾಧ್ಯವಾಗಿಸುವ ಕಾಕ್ಲಿಯರ್ ಇಂಪ್ಲಾಂಟ್‌ನಂತಹ ವಿಶೇಷ ರೀತಿಯ ಶ್ರವಣ ಸಾಧನಗಳನ್ನು ಮನುಷ್ಯನ ದೇಹದಲ್ಲಿ ಕೆಲಸ ಮಾಡಲು ಸಾಧ್ಯವಾಗಿಸುತ್ತದೆ.

ಇಂಜಿನಿಯರಿಂಗ /ವಾಸ್ತುಶಿಲ್ಪ/ಉತ್ಪಾದನೆ (Engineering/Architecture/Manufacturing): ವಿನ್ಯಾಸ ಹಾಗೂ ರೇಖಾಚಿತ್ರಗಳನ್ನು ಬರೆಯಲು ವಾಸ್ತುಶಿಲ್ಪಿಗಳು ಮತ್ತು ಇಂಜಿನಿಯರ್‌ಗಳು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಬಳಸುತ್ತಿದ್ದಾರೆ. ಕಂಪ್ಯೂಟರ್‌ಗಳು ರಚಿಸಿದ ವಸ್ತುಗಳನ್ನು ಮೂರೂ ಆಯಾಮಗಳಿಂದ ನೋಡಬಹುದು. ವರ್ಚುವಲ್ ರಿಯಾಲಿಟಿಯಂತಹ ತಂತ್ರಗಳನ್ನು ಬಳಸಿಕೊಂಡು ವಾಸ್ತುಶಿಲ್ಪಿಗಳು ವಿನ್ಯಾಸಗೊಳಿಸಿದ ಆದರೆ ಇನ್ನೂ ನಿರ್ಮಿಸದ ಮನೆಗಳನ್ನು ಒಳಹೊಕ್ಕು ನೋಡಬಹುದು(ಅನ್ವೇಶಿಸಬಹುದು). ಉತ್ಪಾದನಾ ಕಾರ್ಖಾನೆಗಳು ಅಪಾಯಕಾರಿಯಾದಂತಹ ಕೆಲಸಗಳನ್ನು ಮಾಡಲು ಕಂಪ್ಯೂಟರ್ ಆಧಾರಿತ ರೋಬೋಟಿಕ್ ಆಯುಧಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಇದಲ್ಲದೇ ಕಂಪ್ಯೂಟರ್ ಏಡೆಡ್ ಮ್ಯಾನುಫ್ಯಾಕ್ಚರಿಂಗ(CAM) ಹೊರತಾಗಿ ಉತ್ಪನ್ನಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲು ಭಾಗಗಳನ್ನು ಆದೇಶಿಸಲು ಹಾಗೂ ಉತ್ಪಾದನೆಗಳ ಯೋಜನೆ ರೂಪಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಹೀಗಾಗಿ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಸಂಪೂರ್ಣ ಉತ್ಪಾದನಾ ಪ್ರಕ್ರಿಯೆ ಸಂಘಟಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ಮನರಂಜನೆ (Entertainment) : ಮನರಂಜನಾ ಉದ್ಯಮದಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಹೆಚ್ಚಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತಿವೆ. ಚಿತ್ರಗಳು ಮತ್ತು ಶಬ್ದಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತಿದೆ. ಕಂಪ್ಯೂಟರ್ ಇಲ್ಲದೇ ವಿಶೇಷ ಪರಿಣಾಮಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಪ್ರೇಕ್ಷಕರನ್ನು ಮಂತ್ರಮುಗ್ಧವಾಗಿಸುವುದು ಅಸಾಧ್ಯವಾಗಿರುತ್ತದೆ. ಇದರ ಜೊತೆಗೆ ಕಂಪ್ಯೂಟರ್‌ನ ಅನಿಮೇಶನ್ , ವರ್ಣರಂಜಿತ ಗ್ರಾಫಿಕ್ಸ ಇವು ಚಲನಚಿತ್ರೋದ್ಯಮವನ್ನು ಆಧುನೀಕರಿಸಿವೆ.

ಸಂವಹನ (Communication) : ಇ ಮೇಲ್ ಅಥವಾ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮೇಲ್ ಕಂಪ್ಯೂಟರ್ ಉಪಯೋಗಿಸುವ ಒಂದು ಸಂವಹನ ಮಾಧ್ಯಮ. ಕಂಪ್ಯೂಟರ್‌ನ ಮತ್ತು ದೂರವಾಣಿಯ ಸಹಾಯದಿಂದ ಇ ಮೇಲ್ ಮುಖಾಂತರ ಒಬ್ಬ ವ್ಯಕ್ತಿಯಿಂದ ಒಬ್ಬ ಅಥವಾ ಹೆಚ್ಚು ವ್ಯಕ್ತಿಗಳಿಗೆ ಸಂದೇಶಗಳು ಮತ್ತು ವರದಿಗಳನ್ನು ವರ್ಗಾಯಿಸಲಾಗುತ್ತದೆ. ಈ ಸೇವೆಯ ಅನುಕೂಲವೆಂದರೆ ಮಾಹಿತಿಯನ್ನು ವರ್ಗಾಯಿಸುವಾಗ ಸಮಯ ಉಳಿತಾಯವಾಗುತ್ತದೆ, ಕಾಗದ ವ್ಯರ್ಥವಾಗುವುದನ್ನು ತಪ್ಪಿಸುತ್ತದೆ. ಇದಲ್ಲದೇ ಇ ಮೇಲ್ ಸ್ವೀಕರಿಸುವ ವ್ಯಕ್ತಿ ತನಗೆ ಸಮಯ ಸಿಕ್ಕಾಗ ಅದನ್ನು ಓದಬಹುದು, ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಿ ಇಡಬಹುದು, ಪ್ರತಿ ಉತ್ತರಿಸಬಹುದು ಅಥವಾ ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಅಳಿಸಬಹುದು

ಬಿಸನೆಸ್ ಅಪ್ಲಿಕೇಶನ್ (Business Application) : ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಅತೀ ಮುಖ್ಯವಾದ ಉಪಯೋಗವಾಗಿದೆ. ಆರಂಭದಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಬ್ಯಾಚ್ ಪ್ರೊಸೆಸಿಂಗ ಕಾರ್ಯಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿತ್ತು. ಇಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ನ ತಕ್ಷಣದ ಪ್ರತಿಕ್ರಿಯೆಯ ಅವಶ್ಯಕತೆ ಇರಲಿಲ್ಲ. ಪ್ರಸ್ತುತ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ನೈಜ ಸಮಯದ(real time) ಅಪ್ಲಿಕೇಶನ್‌ಗಳಿಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (ಮಾರಾಟದ ಕೌಂಟರ್‌ನಲ್ಲಿ) ಇಲ್ಲಿ ತಕ್ಷಣ ಕಂಪ್ಯೂಟರ್‌ನ ಪ್ರತಿಕ್ರಿಯೆಯ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಇದಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ವ್ಯವಹಾರ ಮುಂದಾಲೋಚನೆ ,ಬ್ಯಾಂಕಿಂಗ್ ಕಾರ್ಯಗಳಲ್ಲಿ ವೇತನ ಬಿಲ್ ಹಾಗೂ ವೈಯಕ್ತಿಕ ದಾಖಲೆಗಳನ್ನು ತಯಾರಿಸಲು ಹಾಗೂ ದತ್ತಾಂಶ ಸಂಗ್ರಹಣೆ , ವಿವಿಧ ರೀತಿಯ ವಿಮಾ ವ್ಯವಹಾರ, ನಿರ್ವಹಣೆಯ ಸಹಾಯಕವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ವ್ಯವಹಾರಗಳಲ್ಲಿ ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕನ್ನು ಬಳಸಿಕೊಂಡು ದತ್ತಾಂಶ ಹಾಗೂ ಮಾಹಿತಿಯನ್ನು ಹಂಚಿಕೊಳ್ಳಲಾಗುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕಲ್ಲಿ ಹಲವಾರು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಒಂದಕ್ಕೊಂದು ಜೋಡಿಸಲಾಗುತ್ತದೆ. ಇಮೇಲ್ ನ ಬಳಕೆ ಮತ್ತು ಇಂಟರ್ನೆಟ್ , ವ್ಯವಹಾರದ ವಿಧಾನಗಳನ್ನು ಬದಲಾಯಿಸಿವೆ.

ಪ್ರಕಟಣೆ (Publishing) : ಕಂಪ್ಯೂಟರ್‌ಗಳು ಡೆಸ್ಕಟಾಪ್ ಪಬ್ಲಿಶಿಂಗ್ (DTP) ಎಂಬ ಕ್ಷೇತ್ರವನ್ನು ರಚಿಸಿವೆ. ಡಿಟಿಪಿಯಲ್ಲಿ ಕಂಪ್ಯೂಟರ್, ಮತ್ತು ಲೇಸರ್ ಪ್ರಿಂಟರನ ಸಹಾಯದಿಂದ ಪ್ರಕಟಣೆಯ ಕೆಲಸವನ್ನು ಸ್ವತಃ ಮಾಡಬಹುದು. ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಡಿಟಿಪಿ ಸಾಫ್ಟವೇರ್ ಸಹಾಯದಿಂದ ಬಹಳಷ್ಟು ದೀರ್ಘಸಮಯ

© NIMI NOT TO BE REPUBLISHED

ಹಿಡಿಯುವ ಕಾರ್ಯಗಳಾದ ಕೋಷ್ಟಕ ತಯಾರಿಕೆ ಹಾಗೂ ಕೈಪಿಡಿ ತಯಾರಿಕೆಯಂತಹ ಕಾರ್ಯಗಳನ್ನು ಸುಲಭವಾಗಿ ಮಾಡಬಹುದಾಗಿದೆ.

ಬ್ಯಾಂಕಿಂಗ (Banking) : ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಬ್ಯಾಂಕಿಂಗ್ ಮತ್ತು ಹಣಕಾಸು ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿದೆ. ಜನರು ದಿನದ 24 ಗಂಟೆಯೂ ಹಣವನ್ನು ಠೇವಣಿ ಇಡಲು ಮತ್ತು ಹಿಂಪಡೆಯಲು ATM (Automated Teller Machine) ಸೇವೆಗಳನ್ನು ಉಪಯೊಗಿಸಬಹುದು. ಬ್ಯಾಂಕಿನ ವಿವಿಧ ಶಾಖೆಗಳು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್ ಮುಖಾಂತರ ಅಂತರ್‌ಶಾಖೆಯ ಚೆಕ್ ಮತ್ತು ಡ್ರಾಫ್ಟನ ವಹಿವಾಟುಗಳನ್ನು ಯಾವುದೇ ವಿಳಂಬವಿಲ್ಲದೇ ಮಾಡಬಹುದಾಗಿದೆ.

ರೈಲ್ವೆ ಮೀಸಲಾತಿ ವ್ಯವಸ್ಥೆ (Railway Reservation System) : ಈ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸಿಕೊಂಡು ಬಳಕೆದಾರರು ಆನ್‌ಲೈನ್ ಮೂಲಕ ಈ ಕೆಳಕಂಡ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಹುದು. (ವೆಬ್ ಸೈಟ್ : www.irctc.co.in)

- ರೈಲು ಮತ್ತು ಅದರ ಸಮಯನ್ನು ಹುಡುಕುವುದು
- ಆಸನ ಮತ್ತು ಬರ್ಥ್‌ನ ಲಭ್ಯತೆಯನ್ನು ಪರಿಶೀಲಿಸುವುದು.
- ಟಿಕೆಟ್ ಕಾಯ್ದಿರಿಸುವುದು ಮತ್ತು ರದ್ದುಗೊಳಿಸುವುದು
- ಪಿಎನ್ ಆರ್ ಸ್ಥಿತಿ (PNR Status) ನೋಡುವುದು.

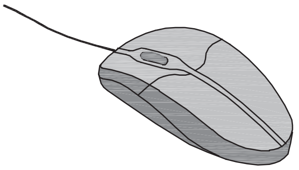
ದೂರವಾಣಿ / ವಿದ್ಯುತ್ ಮಂಡಳಿ ಬಿಲ್ಲಿಂಗ (Telephone / Electricity Board Billing) : ಈ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸಿಕೊಂಡು ಬಳಕೆದಾರರು ಆನ್‌ಲೈನ್ ಮೂಲಕ ಈ ಕೆಳಕಂಡ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಹುದು. (ವೆಬ್ ಸೈಟ್ : portal.bsnl.in-BSNL)

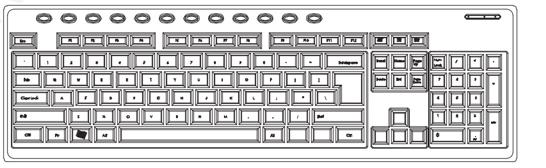
- ದೂರವಾಣಿ / ವಿದ್ಯುತ್ ಮಂಡಳಿ ಸಂಖ್ಯೆಯನ್ನು ನೋಂದಾಯಿಸುವುದು.
- ಬಿಲ್ ಮೊತ್ತವನ್ನು ಪರಿಶೀಲಿಸುವುದು ಮತ್ತು ಪಾವತಿಸುವುದು.
- ದೂರುಗಳನ್ನು ದಾಖಲಿಸುವುದು.

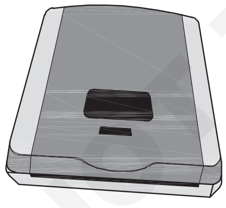
ಇ ಆಡಳಿತ (E-Governance) : ಇ ಆಡಳಿತವು ತಂತ್ರಜ್ಞಾನದಿಂದ ನಡೆಯುವ ಆಡಳಿತವನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇ ಆಡಳಿತವು ಮಾಹಿತಿ ಮತ್ತು ಸಂವಹನ ತಂತ್ರಜ್ಞಾನದ ಅಪ್ಲಿಕೇಶನ್ ಆಗಿದೆ. ಇದು ಸರ್ಕಾರಿ ಸೇವೆಗಳನ್ನು ತಲುಪಿಸುವುದು. ಮಾಹಿತಿ ವಿನಿಮಯ ಸಂವಹನ ವಹಿವಾಟು ಕಾರ್ಯವನ್ನು ಮಾಡುತ್ತದೆ . ಸರ್ಕಾರದಿಂದ - ನಾಗರೀಕರು(G2C) , ಸರ್ಕಾರದಿಂದ - ವ್ಯವಹಾರ (G2B), ಸರ್ಕಾರದಿಂದ - ಸರ್ಕಾರ (G2G)ಗಳ ಜೊತೆಗೆ ಬ್ಯಾಕ್ ಆಫೀಸ್ ಪ್ರಕ್ರಿಯೆಗಳು ಹಾಗೂ ಸಂಪೂರ್ಣ ಸರ್ಕಾರಿ ಚೌಕಟ್ಟಿನ ಪರಿಣಾಮಗಳ ನಡುವಿನ ವಿವಿಧ ಘಟಕಗಳ ವ್ಯವಸ್ಥೆ ಮತ್ತು ಸೇವೆಗಳ ಕ್ರೋಢೀಕರಣವಾಗಿದೆ. ಇ ಆಡಳಿತವು ಸಾಮಾನ್ಯ ಮನುಷ್ಯನಿಗೆ ದೋಷರಹಿತ , ಪಾರದರ್ಶಕ ಮತ್ತು ಪರಿಣಾಮಕಾರಿಯಾದ ಸೇವೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ.(ನಗರ ಮತ್ತು ಗ್ರಾಮೀಣ ಎರಡೂ ಪ್ರದೇಶಗಳಲ್ಲಿ)

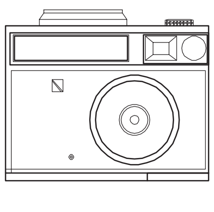
© NIMI NOT TO BE REPUBLISHED

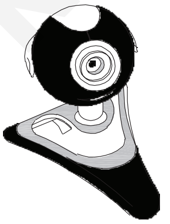
Fig 9

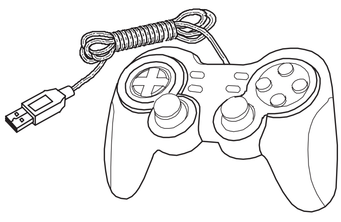

MOUSE

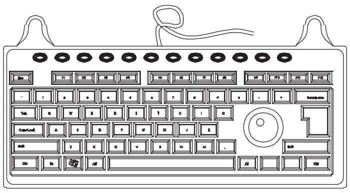
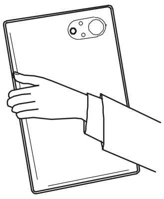
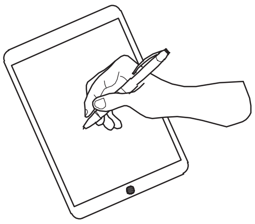
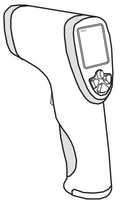
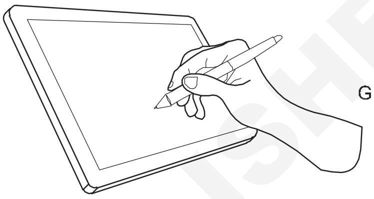
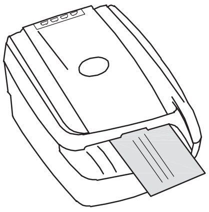
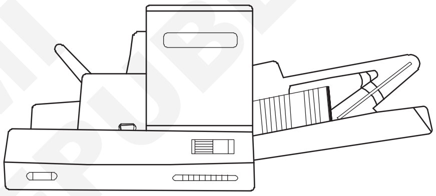
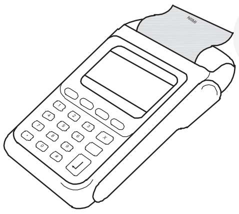
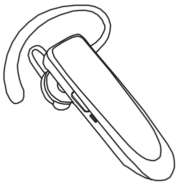

KEYBOARD


SCANNER


CAMERA


WEB CAMERA


JOY STICK

 TARCK BALL	 TOUCH PAD
 LIGHT PEN	 BAR CODE READER
MICROPHONE	 GRAPHIC TABLET
 MAGNETIC INK CHARACTER READER	 OPTICAL MARK READER
 MAGNETIC CARD READER	 BLUETOOTH

CP20N1205T1

NOT TO BE REPUBLISHED © NIMI

ಸಿ ಪಿ ಯು ಆಂತರಿಕ ರಚನೆ ಮತ್ತು ಮದರಬೋರ್ಡ ಪರಿಚಯ (Introduction to CPU architecture and motherboard)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಯಂತ್ರಾಂಶ (hardware) ಎಂದರೇನು ತಿಳಿಸಿ ಹಾಗೂ ಆಂತರಿಕ ಮತ್ತು ಬಾಹ್ಯ ಯಂತ್ರಾಂಶವನ್ನು ಪರೀಕ್ಷಿಸಿ**
- **ಪಟ್ಟಿಮಾಡಿದ ಯಂತ್ರಾಂಶವನ್ನು ಸಂಕ್ಷೀಪ್ತವಾಗಿ ವಿವರಿಸಿ**
- **ಪಾರ್ಟೀಷನ್ ಎಂದರೇನು ಮತ್ತು ಅವುಗಳಿ ವಿಧಗಳನ್ನು ತಿಳಿಸಿ**
- **ಬೂಟಿಂಗ ಮತ್ತು ಅದರ ಕಾರ್ಯವಿಧಾನವನ್ನು ವಿವರಿಸಿ.**

ಕಂಪ್ಯೂಟರ್ ಹಾರ್ಡವೇರ್ (Computer Hardware)

ಕಂಪ್ಯೂಟರ್‌ನ ಭೌತಿಕ ಭಾಗಗಳನ್ನು (ಘಟಕಗಳನ್ನು) ಕಂಪ್ಯೂಟರ್ ಹಾರ್ಡವೇರ್ ಎಂದು ಕರೆಯುತ್ತೇವೆ.

ಆಂತರಿಕ ಹಾರ್ಡವೇರ್‌ನ ಉದಾಹರಣೆಗಳು (Internal hardware example)

- ಬ್ಲೂ ರೇ, ಸಿ ಡಿ ರೋಮ್ ಮತ್ತು ಡಿವಿಡಿ
- ಸಿ.ಪಿ.ಯು
- ಹಾರ್ಡ ಡ್ರೈವ್
- ಮದರ್ ಬೋರ್ಡ
- ರ್‍ಯಾಮ್
- ಸೌಂಡ ಕಾರ್ಡ
- ವಿಡಿಯೋ ಕಾರ್ಡ
- ಎಸ್ ಎಮ್ ಪಿ ಎಸ್

ಬಾಹ್ಯ ಹಾರ್ಡವೇರ್‌ಗೆ ಉದಾಹರಣೆಗಳು (External hardware example)

- ಫ್ಲ್ಯಾಟ್ ಪ್ಯಾನೆಲ್, ಮಾನಿಟರ್ ಮತ್ತು ಎಲ್ ಸಿ ಡಿ
- ಕೀಬೋರ್ಡ
- ಮೌಸ್
- ಪ್ರಿಂಟರ್
- ಸ್ಕ್ಯಾನರ್

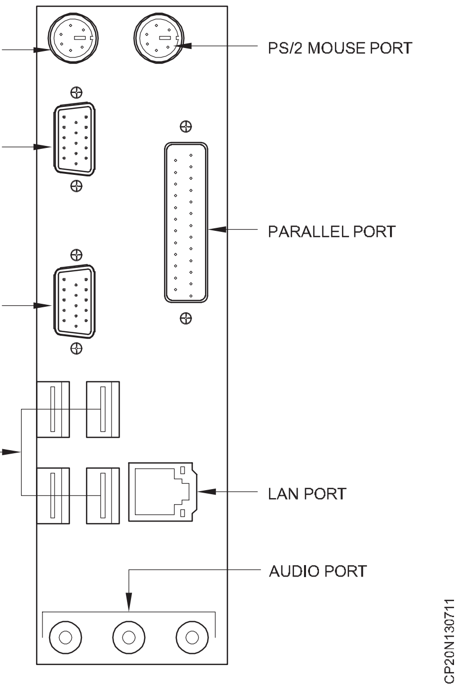
ಸಿ ಪಿ ಯು ಮತ್ತು ಎ ಎಲ್ ಯು (CPU & ALU)

ಕೇಂದ್ರ ಸಂಸ್ಕರಣಾ ಘಟಕ (CPU) ಕಂಪ್ಯೂಟರ್ ನ ಹೃದಯವಾಗಿದೆ. ಈ ಹಾರ್ಡವೇರ್, ಕಂಪ್ಯೂಟರ್ ಪ್ರೊಗ್ರಾಂ ನ ಸೂಚನೆಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.ಇದು ಅಂಕಗಣಿತ, ತಾರ್ಕಿಕ ಮತ್ತು ಇನ್‌ಪುಟ್. ಔಟ್‌ಪುಟ್ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಸಿ ಪಿ ಯು ಪೋರ್ಟಗಳು ಮತ್ತು ಕನೆಕ್ಟರ್‌ಗಳು (CPU Ports and Connectors)

ಪೋರ್ಟ ಇದು ಕಂಪ್ಯೂಟರ್ ಕ್ಯಾಬಿನೆಟ್‌ನ ಹಿಂಭಾಗದಲ್ಲಿರುವ ಕನೆಕ್ಟರ್ ಆಗಿದೆ. ಅಲ್ಲಿ ನೀವು ಪ್ರಿಂಟರ್, ಕೀಬೋರ್ಡ, ಸ್ಕ್ಯಾನರ್, ಮಾಡೆಮ್‌ನಂತಹ ಬಾಹ್ಯ ಸಾಧಗಳನ್ನು ಜೋಡಿಸಬಹುದಾಗಿದೆ. ಇದು ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಇತರ ಬೇರೆ ಸಾಧನಗಳ ನಡುವೆ ಮಾಹಿತಿ ಮತ್ತು ಸೂಚನೆಗಳು ವಿನಿಮಯ ಆಗುತ್ತವೆ. ಕಂಪ್ಯೂಟರ್ ಪೋರ್ಟಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಇನ್‌ಪುಟ್ /ಔಟಪುಟ್ ಪೋರ್ಟ(I/O Ports) ಎಂದೂ ಕರೆಯಲಾಗುತ್ತದೆ. ಈ ಪೋರ್ಟಗಳು ಸರಣಿ (Serial) ಅಥವಾ ಸಮಾನಾಂತರ (Parallel) ಚಿತ್ರ 9 ವೈಯಕ್ತಿಕ ಕಂಪ್ಯೂಟರ್ ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಪೋರ್ಟಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.

Fig 1



© NIMI NOT TO BE REPUBLISHED

ಕೇಬಲ್‌ಗಳನ್ನು ಸರಿಯಾಗಿ ಜೋಡಿಸಲು ಬರುವಂತೆ , ಬಹಳಷ್ಟು ಕನೆಕ್ಟರ್‌ಗಳನ್ನು ಬೇರ್ಪಡಿಸಲಾಗಿದೆ. ಕೀಬೋರ್ಡ ಮತ್ತು ಮೌಸ್ PS2 (Personal System 2) ಕನೆಕ್ಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಪಿ ಎಸ್ 2 ಕನೆಕ್ಟರ್‌ಗಳನ್ನು ಕಲರ್ ಕೋಡ್ ಮಾಡಲಾಗಿದೆ. ಕೀಬೋರ್ಡಗಾಗಿ ನೇರಳೆ ಬಣ್ಣದ ಕನೆಕ್ಟರ ಇರುತ್ತದೆ. ಮೌಸ್‌ಗಾಗಿ ಹಸಿರು ಬಣ್ಣದ ಕನೆಕ್ಟರ್ ಇರುತ್ತದೆ.

- **PS2 ಪೋರ್ಟ್‌ಗಳು** (PS2 Ports) : ಸ್ಟ್ಯಾಂಡರ್ಡ ಕೀಬೋರ್ಡ ಮತ್ತು ಮೌಸ್‌ನ್ನು ಕಂಪ್ಯೂಟರ್‌ನೊಂದಿಗೆ ಪಿ ಎಸ್ 2 ಪೋರ್ಟ ಮುಖಾಂತರ ಜೋಡಿಸಲಾಗುತ್ತದೆ. ಕೀಬೋರ್ಡ ಮತ್ತು ಮೌಸ್‌ನ್ನು ಜೋಡಿಸಲು (ಪ್ಲಗ್), ಮೊದಲು ಕೇಬಲ್‌ನ್ನು ಕನೆಕ್ಟರ್ ಜೊತೆಗೆ ಹೊಂದಿಸಿ. ಆಮೇಲೆ ಕೇಬಲನ್ನು ಕನೆಕ್ಟರ್‌ನ ಒಳಗೆ ತಳ್ಳಿರಿ. ತಳ್ಳುವಾಗ ಕನೆಕ್ಟರ್ ಮೇಲೆ ಒತ್ತಡ ಹಾಕಬೇಡಿ, ಇಲ್ಲದಿದ್ದರೆ ಅದರಲ್ಲಿನ ಪಿನ್ನುಗಳು ಬಾಗುತ್ತವೆ.
- **ಸೀರಿಯಲ್ ಮತ್ತು ಪ್ಯಾರಲಲ್ ಪೋರ್ಟಗಳು** (Serial & Parallel Ports): ಸೀರಿಯಲ್ ಮತ್ತು ಪ್ಯಾರಲಲ್ ಪೋರ್ಟಗಳು ಪ್ರಿಂಟರ್‌ಗಳಿಗೆ ಮತ್ತು ಇತರ ಬಾಹ್ಯ ಸಾಧನಗಳಿಗೆ ಸಂಪರ್ಕ ಹೊಂದಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಸೀರಿಯಲ್ ಪೋರ್ಟ ಮುಖಾಂತರ ಒಂದು ಬೈಟನ್ನು ವರ್ಗಾಯಿಸಲು , ಎಂಟು ಬಿಟ್‌ಗಳು ಸರದಿಯಲ್ಲಿರುತ್ತವೆ ಮತ್ತು ಅವುಗಳನ್ನು ಒಂದೊಂದೆ ಬಿಟ್‌ಗಳನ್ನಾಗಿ ಕಳುಹಿಸಲಾಗುತ್ತದೆ. ಆದರೆ ಪ್ಯಾರಲಲ್ ಪೋರ್ಟ್‌ನಲ್ಲಿ ಎಲ್ಲಾ ಎಂಟು ಬಿಟ್‌ಗಳನ್ನು ಏಕಕಾಲದಲ್ಲಿ ಕಳುಹಿಸಲಾಗುತ್ತದೆ.
- ಎಲ್ಲ ಪ್ಯಾರಲಲ್ ಪೋರ್ಟಗಳು , ಸೀರಿಯಲ್ ಪೋರ್ಟ ಮತ್ತು ವಿಡಿಯೋ ಪೋರ್ಟಗಳು D ಮಾದರಿಯ ಕನೆಕ್ಟರ್‌ಗಳನ್ನು ಬಳಸುತ್ತವೆ.(ಕ್ರಮವಾಗಿ DB - 25M, DB - 9M, DB -15F) ಇವುಗಳ ಆಕಾರದಿಂದ ಇವುಗಳನ್ನು D ಕನೆಕ್ಟರಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಕೇಬಲ್‌ಗಳನ್ನು ಒಂದೇ ರೀತಿಯಲ್ಲಿ ಜೋಡಿಸಲು ಅನುಕೂಲವಾಗುತ್ತದೆ.

ಪೋರ್ಟ್ ಗಳು (USB (Universal Serial Bus) Ports) :

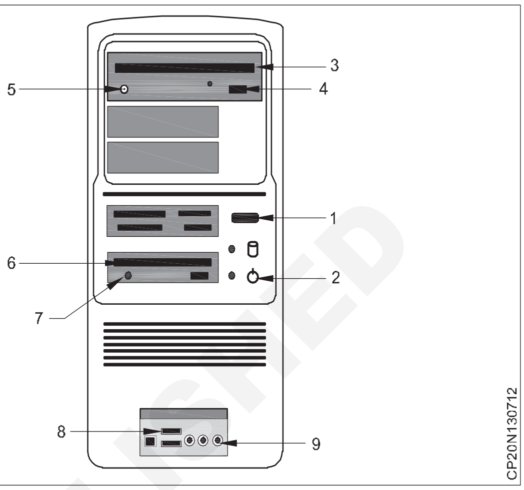
ಡಿಜಿಟಲ್ ಕ್ಯಾಮರಾಗಳು , ಸ್ಕ್ಯಾನರ್‌ಗಳು ಮತ್ತು ಪ್ರಿಂಟರ್‌ಗಳು ಯು ಎಸ್ ಬಿ ಪೋರ್ಟ್ ಮುಖಾಂತರ ಮದರ್‌ಬೋರ್ಡಗೆ ಜೋಡಿಸಲ್ಪಡುತ್ತವೆ. ಯು ಎಸ್ ಬಿ ಕನೆಕ್ಟರನ ವಿಶಿಷ್ಟ ಆಯತಾಕಾರದಿಂದ ಅದನ್ನು ಸುಲಭವಾಗಿ ಗುರುತಿಸಬಹುದಾಗಿದೆ.

ಯು ಎಸ್ ಬಿ ಹಲವಾರು ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿರುವುದರಿಂದ ಪಿ ಸಿ ಗಳಲ್ಲಿ ಇದು ಜನಪ್ರೀಯವಾಗಿದೆ. ಸುಲಭವಾಗಿ ಕಂಪ್ಯೂಟರನ್ನು ಮರು ಪ್ರಾರಂಭಿಸದೇ ಯು ಎಸ್ ಬಿ ಸಾಧನಗಳನ್ನು ನೀವು ಜೋಡಿಸುವುದು ಹಾಗೂ ತೆಗೆಯುವುದು ಮಾಡಬಹುದು.

LAN (Local Area Network) **ಪೋರ್ಟ್:** ಪಿ ಸಿ ಯನ್ನು ಸ್ಥಳಿಯ ನೆಟ್‌ವರ್ಕಗೆ ಅಥವಾ ಹೆಚ್ಚಿನ ವೇಗದ ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಗಳಿಗಾಗಿ ಜೋಡಿಸಲು LAN ಪೋರ್ಟ್ ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

VGA (Video Graphics Array) **ಪೋರ್ಟ್** : ಸಮಗ್ರವಾದ ವಿಡಿಯೋಗಳನ್ನು ನೋಡಲು VGA ಪೋರ್ಟ ಅನುಕೂಲ ಕಲಿಸುತ್ತದೆ.

Fig 2



ಆಡಿಯೋ ಪೋರ್ಟ್ಸ್(Audio Ports) : ಸಮಗ್ರವಾದ ಆಡಿಯೋ ಕೇಳಲು ಇದು ಅನುಕೂಲ ಕಲ್ಪಿಸುತ್ತದೆ. ಆಡಿಯೋ ಜ್ಯಾಕ್‌ಗಳು ಬ್ಯಾಕ್ ಪ್ಯಾನಲ್‌ನಲ್ಲಿರುವ ಅತ್ಯಂತ ಗೊಂದಲಮಯವಾದ ಕನೆಕ್ಟರ್ ಗಳಾಗಿವೆ. ಆದರೂ ಕೆಲವೊಮ್ಮೆ ಜ್ಯಾಕ್ ಗಳನ್ನು ಕಲರ ಕೋಡ್ ಮಾಡಿದರೂ, ಅದಕ್ಕೆ ತಕ್ಕ ಸಾಧನಗಳು ವಿರಳವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತವೆ.

ಸಿ ಪಿ ಯು ಮುಂದಿನ ಫಲಕ (ಚಿತ್ರ 2)(CPU front panel (Fig 2))

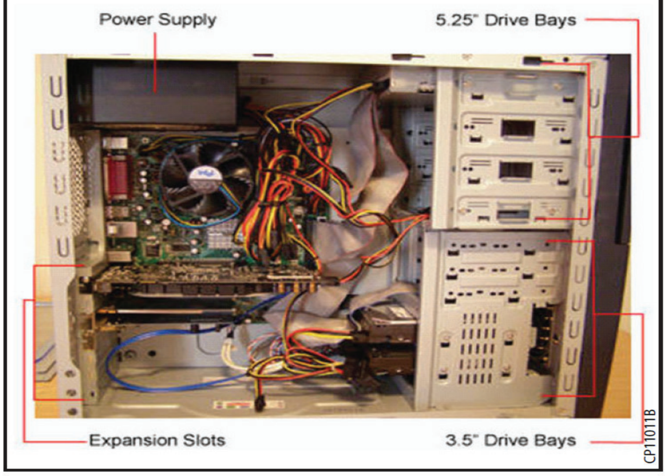
ಇದು ಈ ಕೆಳಗಿನ ಭಾಗಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ

- ಪವರ್ ಆನ್/ ಆಫ್ ಸ್ವಿಚ್
- ಪವರ ಇಂಡಿಕೇಟರ್
- ಸಿಡಿ/ಡಿವಿಡಿ ಡ್ರೈವ್
- ಸಿಡಿ/ಡಿವಿಡಿ ಡ್ರೈವ್ ಬಟನ್
- ಸಿಡಿ/ಡಿವಿಡಿ ಡ್ರೈವ್ ಇಂಡಿಕೇಟರ್
- ಫ್ಲಾಪಿ ಡಿಸ್ಕ ಡ್ರೈವ್
- ಫ್ಲಾಪಿ ಡಿಸ್ಕ ಡ್ರೈವ್ ಇಂಡಿಕೇಟರ್
- ಯು ಎಸ್ ಬಿ ಪೋರ್ಟ್ಸ್
- ಆಡಿಯೋ ಮತ್ತು ಮೈಕ್ ಕನೆಕ್ಟರ್

ಸಿ ಪಿ ಯು ಕ್ಯಾಬಿನೆಟ್‌ಗಳ ವಿನ್ಯಾಸವು ಅದರ ತಯಾರಕರ ಆಧಾರದ ಮೇಲೆ ಬದಲಾಗಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

Fig 3



ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ಮತ್ತು ಅದರ ಘಟಕಗಳು(The System Unit and its Components)

ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ಅನೇಕ ಉಪಯುಕ್ತ ಘಟಕಗಳನ್ನು ಹೊಂದಿದ ಪೆಟ್ಟಿಗೆಯಂತಹ ಘಟಕವಾಗಿದೆ. ಇದರಲ್ಲಿ ಪ್ರತಿಯೊಂದು ಘಟಕವು ವಿಭಿನ್ನ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.ಈ ಘಟಕಗಳು ಕಂಪ್ಯೂಟರ್ ನ ಮುಖ್ಯಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸಲು ಒಟ್ಟಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತವೆ. ಎಂದರೆ ಇನ್‌ಪುಟ್‌ನ್ನು ಸ್ವೀಕರಿಸುವುದು ಅದನ್ನು ಸಂಸ್ಕರಿಸುವುದು ಹಾಗೂ ಉತ್ತರ ನೀಡುವುದು. ಈ ವಿಭಾಗವು ಕಂಪ್ಯೂಟರ್ ನ ಈ ಘಟಕಗಳನ್ನು ಒಂದೊಂದಾಗಿ ವಿವರಿಸುತ್ತದೆ. ಚಿತ್ರ 3 ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ನ ವಿವಿಧ ಘಟಕಗಳನ್ನು ತೋರಿಸುತ್ತದೆ

ವಿದ್ಯುತ್ ಸರಬರಾಜು (Power Supply)

ಪವರ್ ಸಪ್ಲೈ ಕಂಪ್ಯೂಟರ್‌ನ ಎಲ್ಲ ಸಾಧನಗಳಿಗೆ ವಿದ್ಯುತ್ತನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ನ ಕವಚದ ಹಿಂಭಾಗದಲಿ ್ಲದೆ. ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ವಿದ್ಯುತ್ ಸಂರಕ್ಷಣಾ ಸಾಧನದ ಮೂಲಕ ಎಸಿ ಮೇನ್ ನಿಂದ ವಿದ್ಯುತ್ತನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ.

ಈ ವಿದ್ಯುತ್ತನ್ನು ನೇರವಾಗಿ ಆಂತರಿಕ ಘಟಕಗಳಿಗೆ ಒದಗಿಸಲಾಗುವುದಿಲ್ಲ. ಅದರ ಬದಲಾಗಿ ಆಂತರಿಕ ವಿದ್ಯುತ್ ಸರಬರಾಜು ಎನ್ನುವ ಘಟಕವು ಎಸಿ ಇನ್ಪುಟ್ ನ್ನು 5 ಮತ್ತು 12 ವೋಲ್ಟ ಗಳ ಡಿಸಿ ಔಟಪುಟ್ ಆಗಿ ಬದಲಾಯಿಸುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಈ ಆಂತರಿಕ ವಿದ್ಯುತ್ ಸರಬರಾಜು ಘಟಕವನ್ನು ಸ್ವಿಚ್ಡ ಮೋಡ್ ಪಾವರ್ ಸಪ್ಲೈ (Switched Mode Power Supply) SMPS ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಎಸ್ ಎಂ ಪಿಎಸ್ ಕೇಬಲ್ ಕನೆಕ್ಟರ್ ಗಳ ಮುಖಾಂತರ ಆಂತರಿಕ ಘಟಕಗಳದಂತಹ ಫ್ಲಾಪಿ ಡ್ರೈವ್ , ಹಾರ್ಡಡಿಸ್ಕ ಡ್ರೈವ್, ಮದರಬೋರ್ಡ ಮತ್ತು ಬಾಹ್ಯ ಸಾಧನಗಳಾದಂತಹ ಕೀಬೋರ್ಡ ಮುಂತಾದವು ಗಳಿಗೆ ಆವಶ್ಯಕ ವೋಲ್ಟೇಜನ್ನು ಒದಗಿಸುತ್ತದೆ.ಸಿಸ್ಟಮ್ ಯುನಿಟ್ ನ ಆನ್ / ಆಫ್ ಸ್ವಿಚ್ ವಾಸ್ತವವಾಗಿ SMPS ನ ಒಂದು ಭಾಗವಾಗಿದೆ.

Fig 4

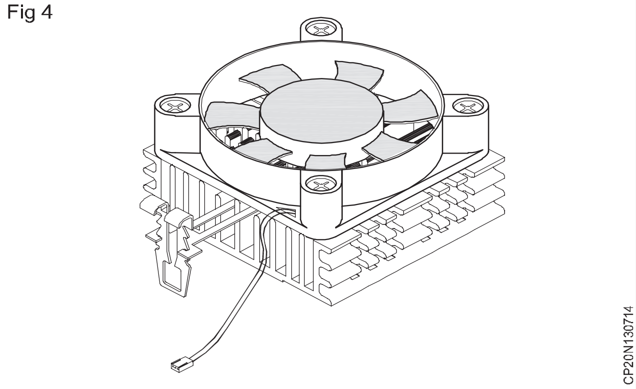
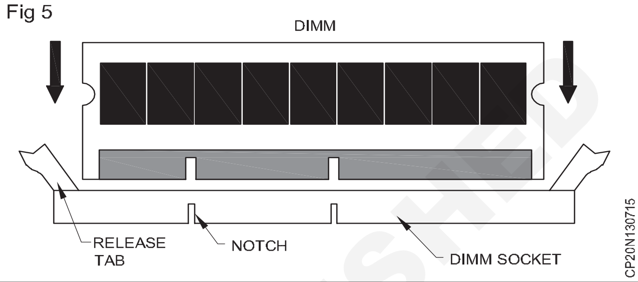


Fig 5

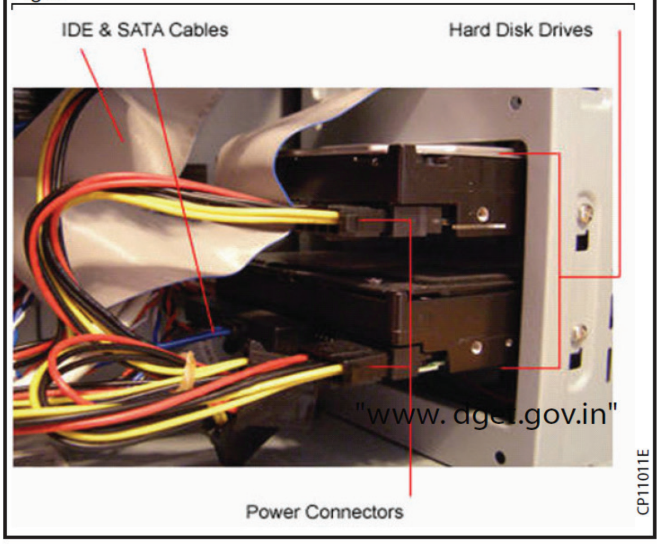


ಫ್ಯಾನ್ (Fan)

ಎಸ್ ಎಮ್ ಪಿ ಎಸ್, ಎಕ್ಸಾಸ್ಟ್ ಫ್ಯಾನ್ ಎಂಬ ಚಿಕ್ಕ ಫ್ಯಾನನ್ನು ಹೊಂದಿದೆ. ಚಿತ್ರ 4 ಎಲ್ಲಿಯವರೆಗೂ ಕಂಪ್ಯೂಟರ್ ಆನ್ ಇರುತ್ತದೆಯೋ ಅಲ್ಲಿಯವರೆಗೆ ಫ್ಯಾನ್ ತಿರುಗುತ್ತಿರುತ್ತದೆ. ಇದರ ಕೆಲಸ ಎಸ್ ಎಮ್ ಪಿ ಎಸ್ ಘಟಕವನ್ನು ತಂಪಾಗಿಡುವುದಾಗಿದೆ.

ಡ್ರೈವ್ ಬೇಸ್ (Drive Bays) : 5.25 inch ಮತ್ತು 3.5 inch ಡ್ರೈವ್ ಬೇ ಗಳು , ಕಂಪ್ಯೂಟರ್ ಹೊಂದಿರಬಹುದಾದ ಹಲವು ವಿಧವಾದ ಸಂಗ್ರಹ ಸಾಧನಗಳನ್ನು ಜೋಡಿಸಿ ಇಟ್ಟಿರುತ್ತಾರೆ.

Fig 6



ವಿಸ್ತರಣಾ ಸ್ಲಾಟ್‌ಗಳು (Expansion Slots): ವಿಸ್ತರಣಾ ಸ್ಲಾಟ್ ಗಳು ಮದರಬೋರ್ಡ ನಲ್ಲಿ ಇರುವ ಒಂದು ಸ್ಲಾಟ್ ಆಗಿದೆ. ಇದು ಇತರೆ ಹೆಚ್ಚಿನ ಕಂಪ್ಯೂಟರ್ ನ ಬಿಡಿಭಾಗಗಳನ್ನು ಜೋಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಮೆಮೋರಿ ಸ್ಲಾಟ್ (Memory Slot) : ಮದರಬೋರ್ಡಿನಲ್ಲಿರುವ ಮೆಮೋರಿ ಸ್ಲಾಟ್ನಲ್ಲಿ ರಾಂಡಮ್ ಅಕ್ಸೆಸ್ ಮೆಮೊರಿ (Random Access Memory) RAM ಅಳವಡಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ. ಚಿತ್ರ 5 ರಂತೆ.

ಸ್ಟೋರೇಜ ಡ್ರೈವರ್‌ಗಳು (Storage Drivers) : ಸ್ಟೋರೇಜ ಡ್ರೈವರ್‌ಗಳಾದ ಹಾರ್ಡ ಡ್ರೈವ್‌ಗಳು, ಆಪ್ಟಿಕಲ್ ಡ್ರೈವ್‌ಗಳು, ಫ್ಲಾಪಿ ಡ್ರೈವ್‌ಗಳು ಕೇಬಲ್ ಮುಖಾಂತರ ಮದರ್ ಬೋರ್ಡಿಗೆ ಜೋಡಿಸಲಾಗಿದ್ದು, ಇದನ್ನು ಕಂಪ್ಯೂಟರ್‌ನ ಒಳಗೆ ಇಡಲಾಗಿರುತ್ತದೆ.

ಐ ಡಿ ಇ & ಸಾಟಾ ಕೇಬಲ್ ಗಳು (IDE & SATA Cables) : ಚಿತ್ರ 6 ರಲ್ಲಿ ಎರಡು ಹಾರ್ಡಡಿಸ್ಕ ಡ್ರೈವ್‌ಗಳು ವಿಭಿನ್ನ ರೀತಿಯಲ್ಲಿ ಮದರಬೋರ್ಡಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟಿವೆ. ಒಂದು ಹಳೆಯ ಐ ಡಿ ಇ ಕೇಬಲ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ ಮತ್ತು ಇನ್ನೊಂದು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಸಾಟಾ (Serial Advanced Technology Attachment) ಕೇಬಲ್ ಉಪಯೋಗಿಸುತ್ತದೆ.

Fig 7

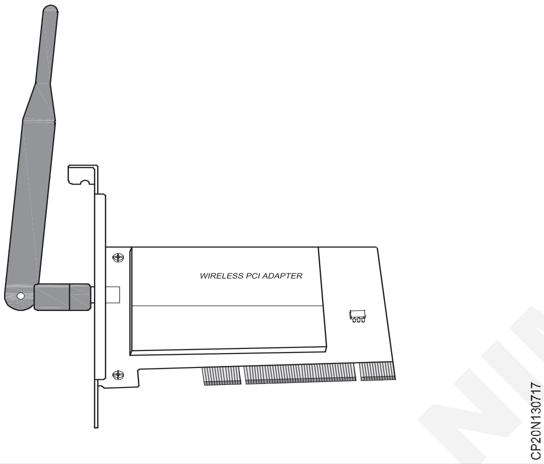
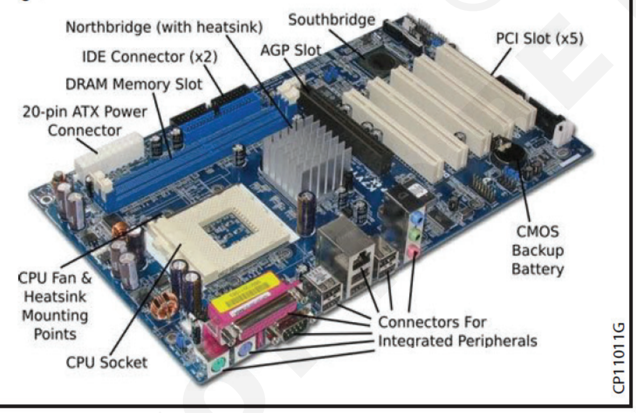


Fig 8



ಪವರ್ ಪೋರ್ಟ (Power Port) : ಡ್ರೈವ್‌ಗಳಲ್ಲಿನ ಪವರ್ ಪೋರ್ಟ್‌ಗೆ ಪ್ಲಗ್ ಮಾಡಿ ಕೇಬಲ್‌ಗಳ ಮುಖಾಂತರ ವಿದ್ಯುತ್ತನ್ನು ಡ್ರೈವ್‌ಗಳಿಗೆ ತಲುಪಿಸಲಾಗುತ್ತದೆ.

ಪೆರಿಫೆರಲ್ ಕಾರ್ಡ್ಸ ಸ್ಲಾಟ್ (Peripheral cards Slot) : ಪೆರಿಫೆರಲ್ ಕಾರ್ಡ್ಸ ಸ್ಲಾಟ್‌ಗಳು ಮದರ್ ಬೋರ್ಡನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಬಿಡಿ ವಿಸ್ತರಣಾ ಸ್ಲಾಟ್ ಗಳಾಗಿವೆ. ಇದರಲ್ಲಿ ಪೆರಿಫೆರಲ್ ಕಾರ್ಡ್ಸಗಳನ್ನು ಅಳವಡಿಸಬಹುದಾಗಿದೆ.

ಪೆರಿಫೆರಲ್ ಕಾರ್ಡ್ಸಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ.

- ಸೌಂಡ ಕಾರ್ಡ
- ವಿಡಿಯೋ ಕಾರ್ಡ
- ಮಾಡೆಮ್
- ವೈರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ

ಚಿತ್ರ 7 ರಲ್ಲಿ ಪಿ ಸಿ ಐ (Peripheral Component Interconnect) ಕನೆಕ್ಟರ್‌ನೊಂದಿಗೆ ವಿನ್ಯಾಸಗೊಳಿಸಲಾದ ಪೆರಿಫೆರಲ್ ಕಾರ್ಡ್ಸನ್ನು ತೋರಿಸಲಾಗಿದೆ.

ಮದರಬೋರ್ಡ (Motherboard): ಮದರ್‌ಬೋರ್ಡ್ ಒಂದು ಪ್ರಿಂಟೆಡ್ ಸರ್ಕಿಟ್ ಆಗಿದ್ದು, ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಆಧಾರವಾಗಿದೆ. ಮತ್ತು ಇದು ಸಿಪಿಯು, ರ್‍ಯಾಮ್ ಹಾಗೂ ಇತರ ಎಲ್ಲ ಕಂಪ್ಯೂಟರ್ ಹಾರ್ಡ್‌ವೇರ್ ಭಾಗಗಳನ್ನು ಒಂದಕ್ಕೊಂದು ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತೆ ಮಾಡುತ್ತದೆ ಚಿತ್ರ 8 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

ಮದರಬೋರ್ಡಇಡೀ ಕಂಪ್ಯೂಟರ್ ವ್ಯವಸ್ಥೆಯ ಅತೀ ಮುಖ್ಯವಾದ ಘಟಕವಾಗಿದೆ. . ಮದರಬೋರ್ಡ ಅತೀ ದೊಡ್ಡ ಬೋರ್ಡ ಆಗಿದ್ದು ಚಿಕ್ಕ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಸರ್ಕಿಟ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಇದರಲ್ಲಿರುವ ಇತರ ಘಟಕಗಳನ್ನು ನೋಡಬಹುದಾಗಿದೆ. ಕಂಪ್ಯೂಟರ್‌ನ ಎಲ್ಲ ಬಿಡಿ ಭಾಗಗಳು ಮದರಬೋರ್ಡಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟಿವೆ.

ಮದರಬೋರ್ಡನ ಬಿಡಿ ಭಾಗಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

- ಕೀಬೋರ್ಡ/ ಮೌಸ್ ಪೋರ್ಟ್ (Keyboard / Mouse Port)
- ಪ್ಯಾರಲಲ್ ಮತ್ತು ಸೀರಿಯಲ್ ಪೋರ್ಟ್ (Parallel and Serial Port)
- ಪ್ರೋಸೆಸರ್ ಸಾಕೆಟ್ (Processor Socket)
- ಎ ಜಿ ಪಿ ಸ್ಲಾಟ್ (AGP Slot)
- ಪಿಸಿಐ ಸ್ಲಾಟ್ (PCI Slot)
- ಐ ಎಸ್ ಎ ಸ್ಲಾಟ್ (ISA Slot)
- ಸಿ ಮಾಸ್ ಬ್ಯಾಟರಿ (CMOS Battary)
- ಡಾಟಾ ಕಾರ್ಡ ಕನೆಕ್ಟರ್ (Data Card Connector)
- ಮೆಮೊರಿ ಸ್ಲಾಟ್ (Memory Slot)
- ಫ್ಲಾಪಿ ಪೋರ್ಟ್ (Floppy Port)
- ಫ್ಯಾನ್ ಹೆಡರ್ (Fan Header)
- ಮೈನ್ ಪವರ ಕನೆಕ್ಟರ್ (Main Power Connector)

© NIMI NOT TO BE REPUBLISHED

ಫ್ಲಾಪಿ ಪೋರ್ಟ್ (Floppy Port): 34 ಪಿನ್ ರಿಬ್ಬನ್ ಕೇಬಲ್ ಮುಖಾಂತರ ಫ್ಲಾಪಿ ಡ್ರೈವ್ ಕಂಪ್ಯೂಟರ್‌ನೊಂದಿಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟಿದೆ. ಹಾಗೂ ಇದು ಮದರಬೋರ್ಡಿಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟಿದೆ. ಫ್ಲಾಪಿ ಕಂಟ್ರೋಲರ್, ಫ್ಲಾಪಿ ಡ್ರೈವ್‌ನ್ನು ನಿಯಂತ್ರಿಸಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ.

ರ್‍ಯಾಮ್ ಸ್ಲಾಟ್ಸ (RAM Slots) : ರ್‍ಯಾಂಡಮ್ ಅಕ್ಸೆಸ್ ಮೆಮೊರಿ ಸಿ ಪಿಯು ಪ್ರಸ್ತುತವಾಗಿ ಬಳಸುತ್ತಿರುವ ಪ್ರೋಗ್ರಾಂ ಮತ್ತು

ದತ್ತಾಂಶಗಳನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ.

ರ್‍ಯಾಮ್ ನ್ನು ಬೈಟ್ ಎಂಬ ಮಾಪನದಲ್ಲಿ ಅಳೆಯಲಾಗುತ್ತದೆ. ರ್‍ಯಾಮ್, ಬೇರೆ ವಿವಿಧ ರೀತಿಯ ಪ್ಯಾಕೇಜ್‌ಗಳಲ್ಲಿ ಸಿಗುತ್ತದೆ.

- ಸಿಮ್(SIMM) - ಸಿಂಗಲ್ ಇನ್ಲೈನ್ ಮೆಮೊರಿ ಮಾಡ್ಯುಲ್ - 32 ಪಿನ್ ಅಥವಾ 72 ಪಿನ್
- ಡಿಮ್ (DIMM) - ಡ್ಯುಯಲ್ ಇನ್ಲೈನ್ ಮೆಮೊರಿ ಮಾಡ್ಯುಲ್ 168 ಪಿನ್

ಹೆಚ್ಚಿನ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ DIMM ಮಾಡ್ಯುಲ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ರೋಮ್ ಬಯೋಸ್ ಚಿಪ್ (ROM BIOS CHIP) : ಇದರ ಅರ್ಥ Read Only Memory Basic Input - Output System.

ಅಂತರ್‌ನಿರ್ಮಿತ ಸಾಫ್ಟವೇರ್, ಡಿಸ್ಕನಲ್ಲಿನ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಸ್ವೀಕರಿಸದೆ ಯಾವ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿಯ ಬಯಾಸ್- ಕೀಬೋರ್ಡ, ಸ್ಕ್ರೀನ್ ಡಿಸ್ಕ ಡ್ರೈವ್‌ಗಳು , ಸರಣಿ ಸಂವಹನ ಸಾಧನಗಳು, ಇನ್ನೂ ಹಲವಾರು ಕಾರ್ಯಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ಆವಶ್ಯಕವಿರುವ ಕೋಡ್‌ಗಳನ್ನು ಹೊಂದಿದೆ.

ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ರೋಮ್‌ಚಿಪ್‌ನಲ್ಲಿ ಬಯಾಸನ್ನು ಅಳವಡಿಸಲಾಗಿರುತ್ತದೆ. (ಸಾಮಾನ್ಯವಾಗಿ ಇದನ್ನು ROM BIOS ಎಂದು ಕರೆಯುತ್ತಾರೆ). ಅದ್ದರಿಂದ ಬಯಾಸ್ ಯಾವಾಗಲೂ ಲಭ್ಯವಿರುತ್ತದೆ. ಮತ್ತು ಡಿಸ್ಕ ವೈಫಲ್ಯಗಳಿಂದ ಇದು ಹಾಳಾಗುವುದಿಲ್ಲ.

ಕಂಪ್ಯೂಟರ್ ಸ್ವತಃ ಬೂಟ್ ಆಗಲು ಇದು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಏಕೆಂದರೆ ರ್‍ಯಾಮ್, ರೋಮ್ ಗಿಂತಲೂ ವೇಗವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಬಹಳಷ್ಟು ಕಂಪ್ಯೂಟರ್ ಉತ್ಪಾದಕರು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಯಾವ ರೀತಿ ವಿನ್ಯಾಸ ಮಾಡಿರುತ್ತಾರೆಂದರೆ ಪ್ರತೀ ಬಾರಿ ಕಂಪ್ಯೂಟರ್ ಬೂಟ್ ಆಗುವಾಗ ಬಯಾಸ್, ರೋಮ್‌ನಿಂದ ರ್‍ಯಾಮ್‌ಗೆ ಕಾಪಿ ಮಾಡಲ್ಪಡುತ್ತದೆ. ಇದನ್ನು ಅಶ್ಯಾಡೋವಿಂಗ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಅನೇಕ ಆಧುನಿಕ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಫ್ಲ್ಯಾಷ್ ಬಯೋಸನ್ನು ಹೊಂದಿವೆ. ಎಂದರೆ ಬಯೋಸನ್ನು ಫ್ಲ್ಯಾಷ್ ಮೆಮೊರಿ ಚಿಪ್‌ನಲ್ಲ ರೆಕಾರ್ಡ ಮಾಡಲಾಗಿರುತ್ತದೆ. ಅಗತ್ಯ ಬಿದ್ದಾಗ ಇದನ್ನು ನವೀಕರಿಸಬಹುದಾಗಿದೆ. ಎಲ್ಲ ಕಂಪ್ಯೂಟರ್‌ನ ಬಯಾಸನ್ನು ಚನ್ನಾಗಿ ಪ್ರಮಾಣೀಕೃತಗೊಳಿಸಲಾಗಿದೆ.(ಆದರೂ ಬಯಾಸ್‌ನ ವಿಭಿನ್ನ ವರ್ಷನ್‌ಗಳು ಇರುತ್ತವೆ.)

ಹೆಚ್ಚಿನ DOS ಕಾರ್ಯಗಳನ್ನು ಸಾಫ್ಟವೇರ್ ಮಾಡ್ಯೂಲ್‌ಗಳ ಮುಖಾಂತರಸೇರಿಸಲಾಗುತ್ತದೆ. ನೀವು ಬಯಾಸನ್ನು ಬದಲಾಯಿಸದೇ DOS ನ ಹೊಸ ಆವೃತ್ತಿನ್ನು ಅಪ್‌ಗ್ರೇಡ ಮಾಡಬಹುದು. ಪ್ಲಗ್ ಅಂಡ್ ಪ್ಲೇ ಸಾಧನಗಳನ್ನು ನಿಭಾಯಿಸುವಂಥ ಪಿಸಿ ಬಯಾಸ್‌ಗೆ ಪಿ ಎನ್ ಪಿ ಬಯಾಸ್ (PnPBIOS)ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಈ ಬಯಾಸನ್ನು ಯಾವಾಗಲೂ ರೋಮ್ ಬದಲಾಗಿ ಫ್ಲ್ಯಾಷ ಮೆಮೊರಿಯೊಂದಿಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

CMOS ಬ್ಯಾಟರಿ (CMOS Battary)

CMOS (Complementary Metal Oxide Semiconductor) ಇದು ಕಂಪ್ಯೂಟರ್ ಮದರಬೋರ್ಡನಲ್ಲಿ ಬಯಾಸ ಸೆಟ್ಟಿಂಗನ್ನು ಸಂಗ್ರಹಿಸುವ ಒಂದು ಸಣ್ಣ ಪ್ರಮಾಣದ ಮೆಮೊರಿ ಆಗಿದೆ.

ಹೆಚ್ಚಿನ CMOS ಬ್ಯಾಟರಿಗಳು ಮದರಬೋರ್ಡ ಜೀವಿತಾವಧಿಯವರೆಗಿರುತ್ತವೆ, (ಬಹಳ ಸಂದರ್ಭಗಳಲ್ಲಿ 10 ವರ್ಷಗಳವರೆಗೆ) ಆದರೆ ಕೆಲವು ಸಲ ಬದಲಾಯಿಸಬೇಕಾಗುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ತಪ್ಪಾದ ಅಥವಾ ನಿಧಾನವಾದ ದಿನಾಂಕ ಮತ್ತು ಸಮಯ ಮತ್ತು ಬಯಾಸ್ ಸೆಟ್ಟಿಂಗನ ನಷ್ಟವಾಗುವಿಕೆ ಇವು CMOS ಬ್ಯಾಟರಿ ಹಾಳಾಗಿರುವುದನ್ನು ಸೂಚಿಸುತ್ತವೆ.

ಐ ಎಸ್ ಎ ಸ್ಲಾಟ್(ISA Slot) : (Industry Standard Architecture) ಇದು ಎಕ್ಸ್‌ಪಾನ್ಷನ್ ಬಸ್‌ನ(Expansion bus) ಪ್ರಮಾಣಿತ ರಚನೆಯಾಗಿದೆ. ಮದರಬೋರ್ಡ ISA ಹೊಂದಾಣಿಕೆಯಾಗುವ ಕಾರ್ಡಗಳನ್ನು ಅಳವಡಿಸಲು ಕೆಲವು ಸ್ಲಾಟ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು.

ಪಿ ಸಿ ಐ ಸ್ಲಾಟ್ (PCI Slot) : (Peripheral Component Interconnect) ಇಂಟೆಲ್ ಕಂಪನಿಯು ಪೆರಿಫೆರಲ್ ಕೊಂಪೊನೆಂಟ್ ಇಂಟರ್ ಕನೆಕ್ಟ ಬಸ್ ಪ್ರೊಟೊಕಾಲನ್ನು ಪರಿಚಯಿಸಿತು. ಪಿ ಸಿ ಐ ಬಸನ್ನು I/O ಸಾಧನಗಳನ್ನು ಕಂಪ್ಯೂಟರ್‌ನ ಮುಖ್ಯ ಲಾಜಿಕ್‌ಗೆ ಜೋಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಐಎಸ್‌ಎ ಬಸ್ (ISA BUS) ಇದು ಪಿಸಿಐ ಬಸ್(PCI BUS) ನಿಂದ ಬದಲಾಯಿಸಲ್ಪಟ್ಟಿದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಪಿ ಸಿ ಮದರಬೋರ್ಡಗಳು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಪಿ ಸಿ ಐ ಸ್ಲಾಟ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಪಿ ಸಿ ಐ ಬಸ್ ನ ರಚನೆಯು ಪ್ರೊಸೆಸರ್ - ಸ್ವತಂತ್ರ ಬಸ್ ವ್ಯವಸ್ಥೆಯಾಗಿದ್ದು ಇದು ಸಿಪಿಯು ನ್ನು ಉಪಯೋಗಿಸದೇ ಕಂಪ್ಯೂಟರ್‌ನ ಬಿಡಿ ಭಾಗಗಳಿಗೆ ಕಂಪ್ಯೂಟರ್‌ನ ಮೆಮೊರಿಯನ್ನು ಉಪಯೋಗಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

ಎ ಜಿ ಪಿ ಸ್ಲಾಟ್ (AGP Slot): Accelerates Graphics Port) ಇದು ಅತೀ ವೇಗವಾದ ಪಾಯಿಂಟ್ ಟು ಪಾಯಿಂಟ್ ಚಾನಲ್ ಆಗಿದ್ದು

©NIMI NOT TO BE REPUBLISHED

ಕಂಪ್ಯೂಟರ್‌ನ ಮದರ್ ಬೋರ್ಡ್‌ಗೆ ವೀಡಿಯೋ ಕಾರ್ಡ್‌ಗಳನ್ನು ಲಗತ್ತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಪವರ್ ಸಪ್ಲೈ ಪ್ಲಗ್‌ಇನ್ (Power Supply plug in)

ಹೆಸರೆ ಸೂಚಿಸುವಂತೆ ಪವರ್ ಸಪ್ಲೈ, ಕಂಪ್ಯೂಟರ್ ಕೆಲಸ ಮಾಡಲು ಅವಶ್ಯಕವಾದ ವಿದ್ಯುತ್ತನ್ನು ಒದಗಿಸುತ್ತದೆ. ಪವರ್ ಸಪ್ಲೈ 110 V AC ವಿದ್ಯುತ್ತನ್ನು ಸ್ವೀಕರಿಸಿ ಅದನ್ನು +/- 12 Volt , +/- 5 Volt ಮತ್ತು 3.3 Volt DC ವಿದ್ಯುತ್ತಿಗೆ ಬದಲಾಯಿಸುತ್ತವೆ. ಪವರ ಸಪ್ಲೈ ಕನೆಕ್ಟರ್ 20 ಪಿನ್ ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಹಾಗೂ ಕನೆಕ್ಟರ್ ಒಂದೇ ದಿಕ್ಕಿನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಹಾರ್ಡ್‌ಡಿಸ್ಕ ಮತ್ತು ವಿಭಜನೆಗಳು (Hard disk and Partitions)

ವಿಭಜನೆ ಎನ್ನುವುದು ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಹಲವಾರು ಭಾಗಗಳಲ್ಲಿ ವಿಭಾಗಿಸುವ ಒಂದು ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಹಾಗೂ ಅದರ ಒಂದು ಭಾಗದಲ್ಲಿ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್(OS) ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾಗುತ್ತದೆ. ಎರಡು ಮತ್ತು ಹೆಚ್ಚಿನ ಭಾಗಗಳಲ್ಲಿ ಬೇರೆ ಬೇರೆ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್(OS) ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು.

ಆದರೆ ಇದು ಒಂದು ಪಾರ್ಟಿಷನ್‌ನ್ನು ಹೊಂದಿರಬಹುದು, ಮತ್ತು ಒಂದು ಓಎಸ್ ಅನ್ನು ಇಸ್ಟಾಲ್ ಮಾಡಲು ಸಂಪೂರ್ಣ ಹಾರ್ಡ್‌ಡಿಸ್ಕ್ ಸ್ಥಳವನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಆದರೆ ಇದು ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದ ಹಾರ್ಡ್‌ಡಿಸ್ಕ್ ಹೊಂದಿದ ಬಳಕೆದಾರರಿಗೆ ದತ್ತಾಂಶ ನಿರ್ವಹಣೆ ಕನಸಿನ ಮಾತೆ ಸರಿ.

ಈಗ ಮಾಸ್ಟರ್ ಬೂಟ್ ರೆಕಾರ್ಡ್ (Master Boot Record -MBR) ರಚನೆಯ ಕಾರಣದಿಂದ ನಾಲ್ಕು ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ಮಾತ್ರ ಹೊಂದಿರುತ್ತದೆ. ಮತ್ತು ಈ ನಾಲ್ಕು ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಎಕ್ಸ್‌ಟೆಂಡೆಡ್ ಪಾರ್ಟಿಷನ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೆ ಬಳಸಲಾರದ ಪಾರ್ಟಿಷನ್ ಅಗಿದೆ. ಆದರೆ ಇದು ಸಂಗ್ರಾಹಕದಂತೆ("Container") ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ಇದನ್ನು ಲಾಜಿಕಲ್ ಡ್ರೈವ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ ಈ ಎಕ್ಸ್‌ಟೆಂಡೆಡ್ ಪಾರ್ಟಿಷನ್‌ನ್ನು ಹಲವಾರು ಲಾಜಿಕಲ್ ಪಾರ್ಟಿಷನ್‌ಗಳಲ್ಲಿ ಉಪವಿಂಗಡಿಸಲಾಗುತ್ತದೆ.

ಒಂದು ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿ ಬೂಟ್ ಮಾಡುವ ಸಲುವಾಗಿ, ಅದು ಬೂಟಬಲ್ ಪಾರ್ಟಿಷನ್ ಅಥವಾ ಆ್ಯಕ್ಟಿವ್ ಪಾರ್ಟಿಷನ್ ಎಂದು ಹೆಸರಿಸಲಾಗಿರಬೇಕು. ಆ್ಯಕ್ಟಿವ್ ಪಾರ್ಟಿಷನ್ ಎಂದರೆ ಆ ಪಾರ್ಟಿಷನ್ ಬೂಟೇಬಲ್ ಎಂದು ಫ್ಲ್ಯಾಗ್ ಮಾಡಿರಬೇಕು ಅಥವಾ ಓಎಸ್ ಅನ್ನು ಹೊಂದಿರಬೇಕು, ಸಾಮಾನ್ಯವಾಗಿ ಇದು ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್ ಆಗಿರುತ್ತದೆ.

ಪಾರ್ಟಿಷನ್‌ನ ವಿಧಗಳು:

- ಮಾಸ್ಟರ್
- ಪಾರ್ಟಿಷನ್
- ಎಕ್ಸ್ಟೆಂಡೆಡ್ ಮತ್ತು
- ಲಾಜಿಕಲ್ ಎಕ್ಸ್ಟೆಂಡೆಡ್

ಮಾಸ್ಟರ್ ಬೂಟ್ ರೆಕಾರ್ಡ(MBR) : ಎಂಬಿಆರ್ ಇದು 512 ಬೈಟ್‌ಗಳ ಚಿಕ್ಕ ಪಾರ್ಟಿಷನ್ ಆಗಿದ್ದು , ಹಾರ್ಡಡಿಸ್ಕನ ಮೊದಲನೇ ಭೌತಿಕ ಸೆಕ್ಟರನಲ್ಲಿರುತ್ತದೆ. ಈ ಸ್ಥಳವನ್ನು CHS 0,0,1 ಎಂದು ಗುರುತಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ 0 ಸಿಲಿಂಡರ್, 0 ನೆ ಹೆಡ್ ಮತ್ತು 1 ನೇ ಸೆಕ್ಟರ್. ಎಂ ಬಿ ಆರ್, ಬೂಟ್ ಸ್ಟ್ರಾಪ್ ಪ್ರೋಗ್ರಾಂ ಎಂದು ಕರೆಯಲ್ಪಡುವ ಒಂದು ಚಿಕ್ಕ ಪ್ರೋಗ್ರಾಂನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇದು ಯಾವುದೇ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್(OS)ನ್ನು ಬೂಟ್ ಮಾಡುವ ಜವಾಬ್ದಾರಿ ಹೊಂದಿರುತ್ತದೆ. ಎಂ ಬಿ ಆರ್ ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್ ಎಂಬ ಟೇಬಲ್ಲನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಈ ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್ ಹಾರ್ಡಡಿಸ್ಕನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳ ಟೇಬಲನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ. ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್ ಇಡೀ ಎಕ್ಸ್ಟೆಂಡೆಡ್ ಟೇಬಲ್‌ನ್ನು ಒಂದೇ ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್ ಎಂದು ಪರಿಗಣಿಸುತ್ತದೆ. ಮತ್ತು ಅದನ್ನು ಟೇಬಲ್‌ನಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.

ಆದ್ದರಿಂದ ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್‌ನಲ್ಲಿ ಎರಡು ನಮೂದುಗಳ ಸಂಭಾವ್ಯತೆ ಇರುತ್ತದೆ.

- 4 ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳವರೆಗೆ
- 3 ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳವರೆಗೆ ಮತ್ತು 1 ಎಕ್ಸ್ಟೆಂಡೆಡ್ ಪಾರ್ಟಿಷನ್‌ಗಳವರೆಗೆ(ಒಟ್ಟು 4 ನ್ನು ಮೀರಬಾರದು)

ಪಾರ್ಟಿಷನ್ ಬೂಟ್ ಸೆಕ್ಟರ್ (Partition Boot Sector) PBR : ಇದು ಮೊದಲನೇ ತಾರ್ಕಿಕ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಎಂದರೆ, ಇದು ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ನ ಪ್ರಾರಂಭದ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಇದು 512 ಬೈಟ್‌ಗಳ ಪ್ರದೇಶವಾಗಿದ್ದು , ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಫೈಲ್‌ಗಳನ್ನು ರನ್ ಮಾಡಲು ಕೆಲವು ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಎಲ್ಲ ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳು ತಮ್ಮ ಸ್ವಂತ ಪಿಬಿಆರ್ ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ.

ಎಕ್ಸಟೆಂಡೆಡ್ ಬೂಟ್ ಸೆಕ್ಟರ್ (Extended Boot Sector) EBR: ಇದು ಮೊದಲನೇ ತಾರ್ಕಿಕ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಎಂದರೆ, ಇದು ಎಕ್ಸಟೆಂಡೆಡ್ - ಪಾರ್ಟಿಷನ್‌ನ ಪ್ರಾರಂಭದ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಇ ಬಿ ಆರ್ ಇದು ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್ ಹೊಂದಿದೆ. ಇದು ಲಭ್ಯವಿರುವ ಎಕ್ಸಟೆಂಡೆಡ್ - ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿರುವ ತಾರ್ಕಿಕ ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ. ಅಂದರೆ, ಇದು ಪ್ರತಿ ತಾರ್ಕಿಕ ಪಾರ್ಟಿಷನ್‌ನ ಆರಂಭದ ವಿಳಾಸವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಲಾಜಿಕಲ್ ಎಕ್ಸಟೆಂಡೆಡ್ ಬೂಟ್ ಸೆಕ್ಟರ್ (Logical Extended Boot Sector) LEBR: ಇದು ಮೊದಲನೇ ತಾರ್ಕಿಕ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಇದು

© NIMI NOT TO BE REPUBLISHED

ಪ್ರತಿ ಲಾಜಿಕಲ್ ಪಾರ್ಟಿಷನ್‌ನ ಪ್ರಾರಂಭದ ಸೆಕ್ಟರ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ. ಇದು ಪ್ರಾಥಮಿಕ ಪಾರ್ಟಿಷನ್‌ನ ಪಿಬಿಆರ್‌ನ್ನು ಹೋಲುತ್ತದೆ.

ಬೂಟಿಂಗ್ (Booting)

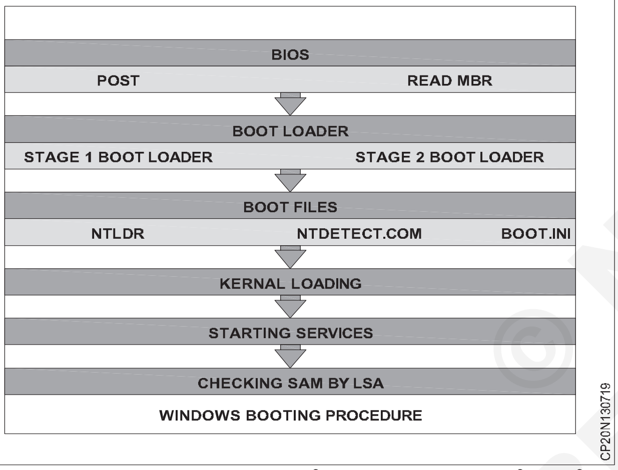
ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ನ್ನು ಲೋಡ್ ಮಾಡುವ ಪ್ರಕ್ರಿಯೆಗೆ ಬೂಟಿಂಗ್ ಎನ್ನುತ್ತಾರೆ. ಎಲ್ಲಾ ಸಿಸ್ಟಮ್ ಸಾಫ್ಟವೇರ್ ಮತ್ತು ಹಾರ್ಡವೇರ್ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಆಗಿರುವುದನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ.

ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ನ್ನು ಬೂಟ್ ಮಾಡುವ ವಿಧಾನ

ಬಯಾಸ್ ನ ಕಾರ್ಯಗಳು (Functions of BIOS)

ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಶುರು ಮಾಡಿದಾಗ ಮೊದಲನೇ ಪ್ರಕ್ರಿಯೆ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಬೇಸಿಕ್ ಇನ್‌ಪುಟ್ ಔಟಪುಟ್ ಸಿಸ್ಟಮ್ (BIOS) ಎರಡು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ ಒಂದು POST ಕಾರ್ಯವನ್ನು ನಡೆಸುವುದು , ಎರಡನೇಯದು ಎಂ ಬಿ ಆರ್ ನ್ನು ಓದುವುದು. (ಚಿತ್ರ 9)

Fig 9



ಪೋಸ್ಟ (POST) : POST ಎಂದರೆ ಪಾವರ್ ಆನ್ ಸೆಲ್ಫ ಟೆಸ್ಟ್ (Power On Self Test) . ಇದು ಕಂಪ್ಯೂಟರ್‌ಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟ RAM, ಹಾರ್ಡಡಿಸ್ಕ ಮುಂತಾದ ಹಾರ್ಡವೇರ್ ಸಾಧನಗಳನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಮತ್ತು ಸಿಸ್ಟಮ್ ಈ ಹಾರ್ಡವೇರ್ ಸಾಧನಗಳೊಂದಿಗೆ ಸರಾಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಬಹುದೆನ್ನುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ. ಒಂದು ವೇಳೆ ಪೋಸ್ಟ ಸರಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸದಿದ್ದರೆ ಬೀಪ್ ಧ್ವನಿಯೊಂದಿಗೆ ಸಿಸ್ಟಮ್ ಸ್ಥಗಿತಗೊಳ್ಳುತ್ತದೆ.

ಈಗ ಬಯಾಸ್ ಬೂಟ್ ಪ್ರಯಾರಿಟಿಯನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಸಿಡಿ ಡ್ರೈವ್ , ಹಾರ್ಡಡಿಸ್ಕ ಅಥವಾ ಫ್ಲಾಪಿ ಡ್ರೈವ ಎಂದು ನಾವು ಬೂಟ್ ಪ್ರಯಾರಿಟಿಯನ್ನು ಹೊಂದಿಸಬಹುದು.

MBR(ಎಂ ಬಿ ಆರ್) - ಬಯಾಸ್ ನ ಮುಂದಿನ ಕರ್ತವ್ಯವೆಂದರೆ ಎಂಬಿಆರ್ ನ್ನು ಓದುವುದು. ಎಂಬಿಆರ್ ಎಂದರೆ, ಮಾಸ್ಟರ್ ಬೂಟ್ ರೆಕಾರ್ಡ ಹಾಗೂ ಇದು ಹಾರ್ಡಡಿಸ್ಕನ ಮೊದಲ ಸೆಕ್ಟರ್ ಆಗಿದೆ. ಎಂಬಿಆರ್ ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್ ಹಾಗೂ ಬೂಟ್ ಲೋಡರ್‌ನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಬೂಟ್ ಲೋಡರ್‌ನ ಕಾರ್ಯಗಳು (Functions of Boot loader)

ಈಗ ಬಯಾಸ್ ಕಂಟ್ರೋಲ್‌ನ್ನು ಬೂಟ್ ಲೋಡರ್ ಗೆ ಕಳುಹಿಸುತ್ತದೆ. ಬೂಟ್ ಲೋಡರ್ ಇದು ಒಂದು ಸಣ್ಣ ಪ್ರೋಗ್ರಾಂ ಆಗಿದ್ದು ಇದು ಕರ್ನಲ್‌ನ್ನು ಕಂಪ್ಯೂಟರ್ ಮೆಮೊರಿಗೆ ಲೋಡ್ ಮಾಡುತ್ತದೆ. ವಾಸ್ತವವಾಗಿ ಬೂಟ್ ಲೋಡರ್‌ನಲ್ಲಿ ಎರಡು ಹಂತಗಳಿವೆ ಅವು ಸ್ಟೇಜ್ 1 ಬೂಟ್ ಲೋಡರ್ ಮತ್ತು ಸ್ಟೇಜ್ 2 ಬೂಟ್ ಲೋಡರ್.

ಸ್ಟೇಜ್ 1 ಬೂಟ್ ಲೋಡರ್ ಇದು ಸ್ಟೇಜ್ 2 ಬೂಟ್ ಲೋಡರ್‌ಗೆ ಕೊಂಡಿ(link) ಆಗಿದೆ. ಸ್ಟೇಜ್ 2 ಬೂಟ್ ಲೋಡರ್ ಬೂಟ್ ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ ಮತ್ತು ಇದು ಕರ್ನಲ್‌ನ್ನು ಮೆಮೊರಿಗೆ ಲೋಡ್ ಮಾಡುತ್ತದೆ.

ಬೂಟ್ ಫೈಲ್‌ಗಳು ಮತ್ತು ಕಾರ್ಯಗಳು (Boot files and functions)

ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಮೂರು ಬೂಟ್ ಫೈಲ್‌ಗಳಿವೆ. ಅವು NTDLR, NTDETECT.COM ಮತ್ತು Boot. ini . ಬೂಟ್ ಫೈಲ್‌ಗಳು ಹಾರ್ಡಡಿಸ್ಕನ ಸಕ್ರಿಯ ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿ ಕಂಡುಬರುತ್ತವೆ. ಹಾಗೂ ಸಾಮಾನ್ಯವಾಗಿ ಇದು ವಿಂಡೋಸ್ ಮಶೀನ್‌ನಲ್ಲಿ , C ಡ್ರೈವ್ ಆಗಿರುತ್ತದೆ.

ಎನ್ ಟಿ ಎಲ್ ಡಿಆರ್(NTLDR): ಎನ್.ಟಿ.ಎಲ್.ಡಿ.ಆರ್. ಎಂದರೆ ಎನ್ ಟಿ ಲೋಡರ್ ಮತ್ತು ಅದರ ಎರಡನೆ ಹಂತದ ಬೂಟ್ ಲೋಡರ. ಎನ್ ಟಿ ಎಲ್ ಡಿಆರ್ ಫೈಲ್ ಇರುವ ಸ್ಥಾನ C:\Windows\i386\NTLDR

Boot.ini - Boot.ini ಎನ್ ಟಿ ಎಲ್ ಡಿಆರ್‌ನ ಸಂರಚನಾ ಫೈಲ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಆಪರೇಟಂಗ ಸಿಸ್ಟಂನ್ನು ಲೋಡ ಮಾಡಿದಾಗ ನಾವು ಯಾವುದೇ ವಾದಗಳನ್ನು ಕರ್ನಲ್‌ಗೆ ರವಾನಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ . ಅದ್ದರಿಂದ ಆ ವಾದಗಳನ್ನು Boot.ini ಮುಖಾಂತರ ರವಾನಿಸಬಹುದು. ನೀವು Boot.ini ನ್ನು ನೋಟಪ್ಯಾಡ ಮೂಲಕ ಬದಲಾಯಿಸಬಹುದು.

NTDETECT.COM

ಈ ಫೈಲ್ ಹಾರ್ಡವೇರನ್ನು ಪತ್ತೆ ಮಾಡುತ್ತದೆ. ಹಾಗೂ ಎನ್ ಟಿ ಎಲ್ ಡಿ ಆರ್ ಗೆ ಮಾಹಿತಿಯನ್ನು ರವಾನಿಸುತ್ತದೆ. ಸಂಗ್ರಹಿಸಿದ ಮಾಹಿತಿಯನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಎನ್ ಟಿ ಎಲ್ ಡಿ ಆರ್ ಹಾರ್ಡವೇರ್ ಕೀಲಿಯನ್ನು ತಯಾರಿಸುತ್ತದೆ ಮತ್ತು ಈ ಕೀಲಿಯನ್ನು ಹಾರ್ಡವೇರನ್ನು ಕಂಡುಹಿಡಿಯಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ ಪ್ರತಿ ರಿಬೂಟ್ ಆದ ನಂತರ ಒಂದು ಹೊಸ ಹಾರ್ಡವೇರ್ ಕೀಲಿ ರಚನೆಯಾಗುತ್ತದೆ. ಅದಕ್ಕಾಗಿ ಹೊಸ ಹಾರ್ಡವೇರ್ ಸ್ಥಾಪಿಸಿದ ನಂತರ ಸಿಸ್ಟಮ್ ರಿಬೂಟ್ ಮಾಡಲು ಕೇಳುತ್ತದೆ. ಎನ್ ಟಿ ಎಲ್ ಡಿ ಆರ್ ತಯಾರಿಸಿದ ಹಾರ್ಡವೇರ್ ಕೀಲಿಗಳನ್ನು , ವಿಂಡೋಸ್ ರೆಜಿಸ್ಟ್ರಿಯ HKEY_LOCAL_MACHINE HARDWARES ನಲ್ಲಿ ಕಾಣಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ (View the BIOS settings and their modifications)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ವಿಂಡೋಸ್ ಆವೃತ್ತಿಗಳನ್ನು ಮತ್ತು ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿರಿ.

ವಿಂಡೋಸ್ ನ ಆವೃತ್ತಿಗಳು ಮತ್ತು ಅದರ ವೈಶಿಷ್ಟ್ಯಗಳು (Windows Versions and its features)

ಮೈಕ್ರೊಸಾಫ್ಟ ವಿಂಡೊಸ್ ಒಂಭತ್ತು ಮುಖ್ಯ ಆವೃತ್ತಿಗಳನ್ನು ಹೊಂದಿದೆ. 1985 ರಲ್ಲಿ ಇದರ ಮೊದಲ ಆವೃತ್ತಿ ಬಿಡುಗಡೆ ಹೊಂದಿತು. 29 ವರ್ಷಗಳ ನಂತರ ವಿಂಡೊಸ್ ಬಹಳ ವಿಭಿನ್ನವಾಗಿ ಕಾಣುತ್ತಿದೆ. ಮೊದಲನೇ ಆವೃತ್ತಿಯಿಂದ ಆರಂಭವಾದ ವಿಂಡೋಸನಲ್ಲಿ ಇರುವ ಅಂಶಗಳು ಅನೇಕ ರೀತಿಯ ಪ್ರಯೋಗಗಳ ಮೂಲಕ ಅಭಿವೃದ್ಧಿ ಹೊಂದಿದವು. ಕಾಲಕ್ರಮೇಣ ಕಿಬೋರ್ಡ ಅಷ್ಟೆ ಬಳಸುವವರು ಮೌಸ್ ಬಳಸುವಂತಾಯ್ತು. ಇತ್ತೀಚಿಗೆ ಮೌಸನ್ನೂ ಮೀರಿಸಿ ಟಚ್‌ಸ್ಕ್ರೀನ್ ಬಳಕೆ ಆರಂಭವಾಗಿದೆ.

ಈಗ ವಿಂಡೋಸ್ ಬಿಲ್ ಗೇಟ್ಸ ಅವರ ಕೈಯಲ್ಲಿ ಜನಿಸಿ 1ನೇ ಆವೃತ್ತಿಯಿಂದ ಪ್ರಸಕ್ತ ಮೈಕ್ರೊಸಾಫ್ಟನ ಚೀಫ್ ಎಕ್ಸಿಕ್ಯೂಟಿವ ಆಫೀಸರ್ ಆಗಿರುವ ಸತ್ಯಾ ನಡೇಲ್ಲಾ ವರೆಗೂ ಅಭಿವೃದ್ಧಿಯಾಗಿರುವ ಇತಿಹಾಸವನ್ನು ಸಂಕ್ಷಿಪ್ತವಾಗಿ ನೋಡೋಣ.

ವಿಂಡೊಸ್ 1 (Windows 1)

ವಿಂಡೊಸ್‌ನ ಮೊದಲನೇ ಆವೃತ್ತಿ (The first version of Windows)

ಇಲ್ಲಿಂದ ವಿಂಡೊಸ್‌ನ ಪ್ರಾರಂಭವಾಯಿತು. ನವೆಂಬರ 1985 ರಲ್ಲಿ ಮೂಲ ವಿಂಡೋಸ್ 1 ಬಿಡುಗಡೆಯಾಯಿತು. ಮತ್ತು ಇದು ಮೈಕ್ರೊಸಾಫ್ಟನ 16 ಬಿಟ್‌ನ ಗ್ರಾಫಿಕಲ್ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್ ನಿಜವಾದ ಮೊದಲನೆ ಪ್ರಯತ್ನವಾಗಿತ್ತು.

ಮೈಕ್ರೊಸಾಫ್ಟ್‌ನ ಸಂಸ್ಥಾಪಕರಾದ ಬಿಲ್ ಗೇಟ್ಸ ಅದರ ಅಭಿವೃದ್ಧಿಯ ಜವಾಬ್ದಾರಿ ವಹಿಸಿದ್ದರು ಮತ್ತು ಕಮಾಂಡ್ ಲೈನ್ ಆಧಾರಿತ ಎಮ್ ಎಸ್ ಡಾಸ್‌ನಲ್ಲಿ ಇದನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದರು.

ಇದು ಮೌಸ್‌ನ ಬಳಕೆಯನ್ನು ಹೆಚ್ಚು ಅವಲಂಬಿಸಿದ್ದು, ಮೌಸ್ ಕಂಪ್ಯೂಟರ್‌ನ ಸಾಮಾನ್ಯವಾಗಿ ಬಳಸಲ್ಪಡುವ ಇನ್‌ಪುಟ್ ಸಾಧನವಾಗಿದೆ.ಈ ಇನ್‌ಪುಟ್ ಸಾಧನವು ಬಳಕೆದಾರರಿಗೆ ಹೆಚ್ಚು ಪರಿಚಯವಾಗಲು ಮೌಸ್ ಮೇಲೆ ಅವಲಂಬಿತವಾದ ರಿವರ್ಸಿ (ಸ್ಕ್ರೀನ್ಶಾಟ್ ನಲ್ಲಿ ಕಾಣಿಸುವ)ಅಂತಹ ಆಟಗಳನ್ನು ಮೈಕ್ರೊಸಾಫ್ಟ ಪರಿಚಯಿಸಿತು. ಇದರಿಂದ ಜನರು ಮೌಸನ್ನು ಸುತ್ತಲೂ ತಿರುಗಿಸುವುದು ಮತ್ತು ಸ್ಕ್ರೀನ್ ಮೇಲಿನ ಅಂಶಗಳನ್ನು ಕ್ಲಿಕ್ ಮಾಡುವುದನ್ನು ಕಲಿಯುವಂತಾಯಿತು.

ವಿಂಡೊಸ್ 2 (Windows 2)

ಅತಿಕ್ರಮಿಸುವ ವಿಂಡೊಸ್ ಗಳೊಂದಿಗೆ ವಿಂಡೊಸ್ 2(Windows 2 with overlapping windows)

ವಿಂಡೊಸ್ 1 ಬಿಡುಗಡೆಯಾದ ಎರಡು ವರ್ಷಗಳ ನಂತರ ಮೈಕ್ರೊಸಾಫ್ಟ ಡಿಸೆಂಬರ್ 1987 ರಂದು ವಿಂಡೊಸ್ 2 ನ್ನು ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು. ವಿಂಡೊಸ್ 2 ರ ದೊಡ್ಡ ಹೊಸತನವೆಂದರೆ ವಿಂಡೊಸ್‌ಗಳು ಪರಸ್ಪರ ಅತಿಕ್ರಮಿಸಬಹುದಾಗಿತ್ತು. ಮತ್ತು ವಿಂಡೊಸ್‌ಗಳನ್ನು ಐಕಾನೈಸಿಂಗ್ ಮತ್ತು ಝೂಮಿಂಗ ಮಾಡುವ ಬದಲು ಮಿನಿಮೈಸ್ ಮತ್ತು ಮ್ಯಾಕ್ಸಿಮೈಸ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿತ್ತು.

ವಿವಿಧ ರೀತಿಯ ಸಿಸ್ಟಮ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳು ಮತ್ತು ಕಾನ್ಫಿಗರೇಷನ್ ಆಯ್ಕೆಗಳನ್ನು ಒಟ್ಟುಗೂಡಿಸಿದ ಸ್ಥಳವಾದ ಕಂಟ್ರೋಲ್ ಪ್ಯಾನಲ್‌ನ್ನು ವಿಂಡೊಸ್ 2 ನಲ್ಲಿ ಪರಿಚಯಿಸಲಾಗಿತ್ತು ಮತ್ತು ಇವತ್ತಿನವರೆಗೂ ಇದು ಉಳಿದುಕೊಂಡಿದೆ.

ಮೈಕ್ರೊಸಾಫ್ಟ ವರ್ಡ ಮತ್ತು ಎಕ್ಸೆಲ್‌ಗಳನ್ನು ಸಹ ವಿಂಡೊಸ್ 2 ನಲ್ಲಿ ಮೊದಲ ಬಾರಿಗೆ ಪರಿಚಯಿಸಲಾಗಿತ್ತು.

ವಿಂಡೊಸ್ 3 (Windows 3)

ವಿಂಡೊಸ್ 3 ವರ್ಣಮಯವಾಗಿದೆ (Windows 3 got colourful)

ಹಾರ್ಡ್‌ಡ್ರೈವ್‌ನ ಅಗತ್ಯವಿರುವ ಮೊದಲ ವಿಂಡೊಸ್‌ನ್ನು 1990 ರಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು. ವಿಂಡೊಸ್ 3 ಇದು ಅತೀ ವ್ಯಾಪಕವಾದ ಯಶಸ್ಸು ಗಳಿಸಿತು ಮತ್ತು ಆ್ಯಪಲ್‌ನ ಮ್ಯಾಸಿಂತೋಶ್ ಮತ್ತು ಕೊಮೊಡೋರ್‌ನ ಅಮಿಗಾ ಇವುಗಳಿಗೆ ಚಾಲೆಂಜರ್ ಆಗಿತ್ತು ಏಕೆಂದರೆ ಝುನಿತ್ ಡಾಟಾ ಸಿಸ್ಟಮ್ ನ್ನು ಒಳಗೊಂಡಂತೆ ಇವುಗಳಲ್ಲಿ ಮೊದಲೆ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ತಯಾರಕರು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಪಿಸಿ ಕಂಪ್ಯಾಟಿಬಲ್ ಗ್ರಾಫಿಕಲ್ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್‌ನ್ನು ಅಳವಡಿಸಲಾಗಿತ್ತು.

ವಿಂಡೊಸ್ 3 ಎಮ್ ಎಸ್ ಡಾಸ್ ನ್ನು ವಿಂಡೊಸ್ ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಪರಿಚಯಿಸಿತು. ಇದು ಮೊದಲಿರುವ ಪ್ರೋಗ್ರಾಂಗಳಿಗೆ ಮಲ್ಟಿಟಾಸ್ಕಿಂಗ (ಬಹು ಕಾರ್ಯನಿರ್ವಹಿಸುವ) ನ್ನು ಪರಿಚಯಿಸಿತು . ಮತ್ತು 256 ಬಣ್ಣಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಇಂಟರ್‌ಫೇಸ್‌ನ್ನು ಹೆಚ್ಚು ಆಧುನಿಕ ಮತ್ತು ವರ್ಣರಂಜಿತವಾಗಿಸಿತು.

© NIMI NOT TO BE REPUBLISHED

ಅತಿ ಮುಖ್ಯವಾಗಿ - ಮೌಸ್ ಬಳಕೆಯನ್ನು ತಿಳಿಸಿ ಕೊಡುವ ಸೊಲಿಟೇರ(solitaire) ಎಂಬ ಇಸ್ಪೀಟು ಎಲೆಯ ಆಟವನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು.

ವಿಂಡೋಸ್ 3.1(Windows 3.1)

ಮೈನ್ ಸ್ವೀಪರ್‌ನೊಂದಿಗೆ ವಿಂಡೊಸ್ 3.1(Windows 3.1 with minesweeper)

ವಿಂಡೋಸ್ 1 ಮತ್ತು ವಿಂಡೋಸ್ 2 ಪಾಯಿಂಟ ರಿಲೀಸ್ ನವೀಕರಣಗಳನ್ನು ಹೊಂದಿದ್ದವು. ಆದರೆ ವಿಂಡೋಸ್ 3.1 1992 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾಯಿತು ಮತ್ತು ಗಮನಾರ್ಹವಾಗಿತ್ತು ಏಕೆಂದರೆ ಇದು ಟ್ರೂಟೈಪ್ ಫಾಂಟ್‌ಗಳನ್ನು ಪರಿಚಯಿಸಿತು ಮತ್ತು ಮೊದಲ ಬಾರಿಗೆ ವಿಂಡೋಸ್‌ಗೆ ಪ್ರಕಾಶನ ವೇದಿಕೆ ಒದಗಿಸಿತು.

ಮೈನ್ ಸ್ವೀಪರ್‌ನ್ನು ಮೊದಲ ಬಾರಿಗೆ ಪರಿಚಯಿಸಲಾಯಿತು. ವಿಂಡೊಸ್ 3.1 ಕೆಲಸ ಮಾಡಲು 1 MB ರ್ಯಾಮ್‌ನ ಅವಶ್ಯಕತೆಯಿತ್ತು ಮತ್ತು ಮೊದಲ ಬಾರಿಗೆ ಮೌಸ್‌ನಿಂದ ನಿಯಂತ್ರಿಸಲ್ಪಡುವ ಎಂಎಸ್ ಡಾಸ್ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಬೆಂಬಲಿಸಿತು. ವಿಂಡೋಸ್ 3.1 , ಸಿ ಡಿ ರಾಮ್‌ನಲ್ಲಿ ವಿತರಿಸಲಾದ ಮೊದಲ ವಿಂಡೋಸ್ ಆಗಿದೆ. ಹಾರ್ಡ ಡ್ರೈವ್‌ನಲ್ಲಿ ಒಮ್ಮೆ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದರೂ ಇದು 10 ರಿಂದ 15 MB ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ. (ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಸಿಡಿ 700 MB ವರೆಗೆ ಮಾಹಿತಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ.)

ವಿಂಡೋಸ್ 95(Windows 95)

ವಿಂಡೋಸ್ 95 **: ಓ ಹಲೋ ಸ್ಟಾರ್ಟ ಮೆನು** (Windows 95 : o hello Start Menu)

ಹೆಸರೇ ಸೂಚಿಸುವಂತೆ ವಿಂಡೋಸ್ 95 ನ್ನು ಅಗಸ್ಟ 1995 ರಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು ಮತ್ತು ಇದರೊಂದಿಗೆ ಮೊದಲ ಸ್ಟಾರ್ಟ ಬಟನ್ ಮತ್ತು ಸ್ಟಾರ್ಟನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು.

ಪ್ಲಗ್ ಮತ್ತು ಪ್ಲೇ ಎನ್ನುವ ಪರಿಕಲ್ಪನೆಯನ್ನು ಸಹ ಪರಿಚಯಿಸಲಾಯಿತು ಎಂದರೆ ಬಾಹ್ಯ ಸಾಧನಗಳನ್ನು ಜೋಡಿಸಿದಾಗ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಅವುಗಳಿಗೆ ಸೂಕ್ತವಾದ ಡ್ರೈವರ್‌ಗಳನ್ನು ಹುಡುಕಿ ಅವು ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತೆ ಮಾಡುತ್ತದೆ. ಇದು ಕಲ್ಪನೆಯಾಗಿತ್ತು. ಆದರೆ ಇದು ಕಾರ್ಯರೂಪಕ್ಕೆ ಬರಲಿಲ್ಲ.

ವಿಂಡೊಸ್ 95, 32 ಬಿಟ್ ಪರಿಸರವನ್ನು, ಟಾಸ್ಕ ಬಾರನ್ನು ಮತ್ತು ಮಲ್ಟಿಟಾಸ್ಕಿಂಗನ್ನು ಪರಿಚಯಿಸಿತು. ವಿಂಡೊಸ್ 95 ಗಾಗಿ ಎಂಎಸ್ ಡಾಸ್ ಪ್ರಮುಖ ಪಾತ್ರವನ್ನು ನಿರ್ವಹಿಸಿದೆ. ಕೆಲವು ಪ್ರೋಗ್ರಾಂಗಳು ಮತ್ತು ಅಂಶಗಳನ್ನು ರನ್ ಮಾಡಲು ಇದರ ಅವಶ್ಯಕತೆ ಇತ್ತು.

ವಿಂಡೊಸ್ 95 ನಲ್ಲಿ ಮೊದಲ ಬಾರಿಗೆ ಇಂಟರನೆಟ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು. ಆದರೆ ವಿಂಡೊಸ್ 95 ಪ್ಲಸ್ ಇದನ್ನು ಡೀಫಾಲ್ಟ್ ಆಗಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿರಲಿಲ್ಲ. ವಿಂಡೊಸ್ 95 ನ ಮುಂದಿನ ಪರಿಷ್ಕರಣೆಗಳು ಇಂಟರನೆಟ್ ಎಕ್ಸಪ್ಲೋರರ್ (IE) ಡೀಫಾಲ್ಟ ಆಗಿ ಒಳಗೊಂಡಿವೆ. ಆಗಿನ ಸಮಯದಲ್ಲಿ ನೆಟ್‌ಸ್ಕೇಪ ನ್ಯಾವಿಗೇಟರ್ ಮತ್ತು ಎನ್‌ಸಿಎಸ್‌ಎ ಮೋಸಾಯಿಕ್‌ಗಳು ಜನಪ್ರಿಯವಾಗಿದ್ದವು.

ವಿಂಡೊಸ್ 98 (Windows 98)

ವಿಂಡೊಸ್ 98 , ಕೊನೆಯ ಶ್ರೇಷ್ಠ ಡಾಸ್ ಆಧಾರಿತ ವಿಂಡೊಸ್ (Windows 98, the last great DOS - based Windows)

ಇದು ಜೂನ್ 1998 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾಯಿತು. ವಿಂಡೊಸ್ 98 ನ್ನು , ವಿಂಡೊಸ್ 95 ನ ಆಧರಿಸಿ ತಯಾರಿಸಲಾಗಿದೆ ಮತ್ತು ಜೊತೆಗೆ ಇಂಟರನೆಟ್ ಎಕ್ಸಪ್ಲೋರರ್4 (IE4) ಔಟ್‌ಲುಕ್ ಎಕ್ಸಪ್ರೆಸ್, ವಿಂಡೊಸ್ ಅಡ್ರೆಸ್ ಬುಕ್, ಮೈಕ್ರೊಸಾಫ್ಟ ಚಾಟ್ ಮತ್ತು ನೆಟ್ ಶೋ ಪ್ಲೇಯರ್ ಇವೆಲ್ಲವೂ ಇದ್ದವು, ನೆಟ್ ಶೋ ಪ್ಲೇಯರ್ ಮುಂತಾದವುಗಳನ್ನು 1999 ರಲ್ಲಿ ವಿಂಡೊಸ್ 98 ನ ಎರಡನೆ ಆವೃತ್ತಿಯಲ್ಲಿ ವಿಂಡೊಸ್ ಮೀಡಿಯಾ ಪ್ಲೇಯರ್ 6.2 ನಿಂದ ಬದಲಾಯಿಸಲಾಯಿತು.

ವಿಂಡೊಸ್ 98 ನಲ್ಲಿ ಬ್ಯಾಕ್ ಮತ್ತು ಫಾರ್ವರ್ಡ ನ್ಯಾವಿಗೇಷನ್‌ನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು. ಮತ್ತು ವಿಂಡೊಸ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನಲ್ಲಿ ಅಡ್ರೆಸ್ ಬಾರ್‌ನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು. ವಿಂಡೊಸ್‌ನ ಮುಂದಿನ ಆವೃತ್ತಿಗಳಿಗೆ ಬೆಂಬಲಿಸುವಂತಹ , ಕಂಪ್ಯೂಟರ್‌ನ ಘಟಕಗಳು ಮತ್ತು ಉಪಕರಣಗಳಿಗೆ ವಿಂಡೊಸ್ ಡ್ರೈವರ್ ಮಾಡೆಲ್‌ನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು.

ವಿಂಡೊಸ್ 98ನಲ್ಲಿ ಯುಎಸ್‌ಬಿ ಬಳಕೆಯನ್ನು ಹೆಚ್ಚು ಸುಧಾರಿಸಲಾಯಿತು. ಯುಎಸ್‌ಬಿ ಹಬ್ ಮತ್ತು ಯುಎಸ್‌ಬಿ ಮೌಸ್ ವ್ಯಾಪಕವಾಗಿ ಬಳಸುವುದಕ್ಕೆ ಕಾರಣವಾಯಿತು.

ವಿಂಡೊಸ್ ಎಂಇ (Windows ME)

ವಿಂಡೊಸ್ ಎಂಇ (Windows ME)

ವಿಂಡೊಸ್ ಸರಣಿಯಲ್ಲಿ ಅನೇಕ ಜನರು, ಕನಿಷ್ಟ ಅವರು ವಿಂಡೊಸ್ ವಿಸ್ತಾ ನೋಡುವ ತನಕ ಕಡಿಮೆ ಎಂದು ಪರಿಗಣಿಸಿದ ವಿಂಡೊಸ್ ಆವೃತ್ತಿ ಇದಾಗಿದೆ.ಎಂಎಸ್ ಡಾಸ್ ಆಧಾರಿತ ವಿಂಡೊಸ್ ಮಿಲೇನಿಯಮ್ ಆವೃತ್ತಿಯು ವಿಂಡೊಸ್‌ನ ಕೊನೆಯ ಆವೃತ್ತಿಯಾಗಿದೆ. ಮತ್ತು ವಿಂಡೊಸ್9x ಸಾಲಿನ ಕೊನೆಯ ಆವೃತ್ತಿಯಾಗಿದೆ.

ಸೆಪ್ಟೆಂಬರ್ 2000 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾದ ಇದು ಗ್ರಾಹಕರನ್ನು ದೃಷ್ಟಿಯಲ್ಲಿಟ್ಟುಕೊಂಡು ತಯಾರಿಸಿದ ಆಪರೆಟಿಂಗ ಸಿಸ್ಟಮ್ ಆಗಿತ್ತು. ಇದನ್ನು ವಿಂಡೊಸ್ 2000 ಜೋಡಿಸಲಾಯಿತು ಮತ್ತು ಉದ್ಯಮ ಮಾರುಕಟ್ಟೆಯೊಂದಿಗೆ ಉಪಯೋಗಿಸಲು ತಯಾರಿಸಲಾಗಿತ್ತು. ಗ್ರಾಹಕರಿಗೆ ಉಪಯೋಗವಾಗುವ ಕೆಲವು ಪ್ರಮುಖ ವಿಷಯಗಳನ್ನು ಪರಿಚಯಿಸಿತು ಹಾಗೂ ಸ್ವಯಂಚಾಲಿತ ಸಿಸ್ಟಮ್ ರಿಕವರಿ ಸಾಧನಗಳನ್ನು ಹೊಂದಿತ್ತು.

© NIMI NOT TO BE REPUBLISHED

ಇದರಲ್ಲಿ ಇಂಟರನೆಟ್ ಎಕ್ಸಪ್ಲೋರರ್ 5.5 (IE5.5) , ವಿಂಡೋಸ್ ಮೀಡಿಯಾ ಪ್ಲೇಯರ್ 7, ಮತ್ತು ವಿಂಡೊಸ್ ಮೂವಿ ಮೇಕರ್ ಮೊದಲ ಬಾರಿಗೆ ಕಂಡುಬಂದವು. ವಿಂಡೊಸ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನಲ್ಲಿ ಆಟೋಕಂಪ್ಲೀಟ್ ಸಾಧನವೂ ಸಹ ಕಂಡು ಬಂದಿತು. ಆದರೆ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ದೋಷಪೂರಿತವಾಗಿದ್ದರಿಂದ ಅಪಖ್ಯಾತಿ ಪಡೆಯಿತು. ಸರಿಯಾಗಿ ಇನ್ಸ್ಟಾಲ್ ಆಗುತ್ತಿರಲಿಲ್ಲ ಮತ್ತು ಸಾಮಾನ್ಯವಾಗಿ ಕಳಪೆಯಾಗಿತ್ತು.

ವಿಂಡೊಸ್ 2000 (Windows 2000)

ವಿಂಡೊಸ್ 2000 ಎಂ ಇ ಎಂಟರ್ ಪ್ರೈಸ್ ಇದರ ಅವಳಿ(Windows 2000 was ME s enterprise twin)

ವಿಂಡೊಸ್ ಎಂ ಇ ಯ ಅವಳಿಯಾದ ವಿಂಡೊಸ್ 2000ನ್ನು ಫೆಬ್ರುವರಿ 2000 ದಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು. ಮತ್ತು ಇದು ಮೈಕ್ರೊಸಾಫ್ಟನ ಬಿಸಿನೆಸ್ ಆಧಾರಿತ ಸಿಸ್ಟಮ್ ಆದ ವಿಂಡೊಸ್ ಎನ್ ಟಿ(Windows NT) ಆಧಾರಿತವಾಗಿದೆ. ನಂತರ ಇದು ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಗೂ ಆಧಾರವಾಯಿತು.

ಮೈಕ್ರೊಸಾಫ್ಟನ ಸ್ವಯಂಚಾಲಿತ ನವೀಕರಣ ಅಂಶವು ವಿಂಡೊಸ್ 2000 ನಲ್ಲಿ ಪ್ರಮುಖ ಪಾತ್ರ ವಹಿಸಿತು ಮತ್ತು ಇದು ಹೈಬರ್ನೇಷನ್ ನ್ನು ಬೆಂಬಲಿಸುವ ಮೊದಲ ವಿಂಡೊಸ್ ಆಗಿತ್ತು.

ವಿಂಡೊಸ್ ಎಕ್ಸ್‌ಪಿ (Windows XP)

ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಇಂದಿಗೂ ಉಳಿದುಕೊಂಡಿದೆ.(Windows XP still servives to this day)

ಇದು ವಿಂಡೊಸ್‌ನ ಅತ್ಯುತ್ತಮ ಆವೃತ್ತಿಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಅಕ್ಟೋಬರ 2001 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾಯಿತು. ಮತ್ತು ಮೈಕ್ರೊಸಾಫ್ಟನ ಎಂಟರ್ಪ್ರೈ ಸ್ ಲೈನ್‌ನ್ನು ಮತ್ತು ಗ್ರಾಹಕರ ಲೈ ನನ್ನು ಒಂದೇ ಸೂರಿನಡಿ ತಂದಿತು.

ಜಾಹೀರಾತು (Advertisement)

ಇದು ವಿಂಡೊಸ್ 2000 ನಂತೆ ವಿಂಡೊಸ್ ಎನ್ ಟಿ ನ್ನು ಆಧರಿಸಿದೆ. ಆದರೆ ವಿಂಡೊಸ್ ಎಮ್ ಇ ನಿಂದ ಗ್ರಾಹಕ ಸ್ನೇಹಿ ಅಂಶಗಳನ್ನು ಬಳಸಿಕೊಂಡಿತು. ಸ್ಟಾರ್ಟ ಮೆನು ಮತ್ತು ಟಾಸ್ಕ ಬಾರ್‌ಗಳು , ಪರಿಚಿತವಾದ ಹಸಿರು ಸ್ಟಾರ್ಟ ಬಟನ್ , ನೀಲಿ ಬಣ್ಣದ ಟಾಸ್ಕ ಬಾರ್ ಮತ್ತು ವಿಸ್ತಾ ವಾಲ್ ಪೇಪರ್ ಜೊತೆಗೆ ಹಲವಾರು ನೆರಳು ಮತ್ತು ಇತರ ದೃಶ್ಯ ಪರಿಣಾಮಗಳು.

LCD ಪರದೆಯ ಮೇಲೆ ಪಠ್ಯವನ್ನು ಸುಲಭವಾಗಿ ಓದಲು ಬರುವಂತೆ ವಿನ್ಯಾಸ ಮಾಡಲಾದ ಕ್ಲಿಯರ್ ಟೈಪನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು. ಬಿಲ್ಟಿನ್ ಸಿ ಡಿ ಬರ್ನಿಂಗ ಸಿ ಡಿ ಮತ್ತು ಇತರ ಮಾಧ್ಯಮಗಳಿಂದ ಸ್ವಯಂ ಪ್ಲೇ, ಜೊತೆಗೆ ವಿವಿಧ ಸ್ವಯಂಚಾಲಿತ ಅಪ್ಡೇಟ್, ಮತ್ತು ಮರುಪಡೆಯುವಿಕೆ ಸಾಧನಗಳು ಇವುಗಳನ್ನು ಹೊಂದಿತ್ತು.

ವಿಂಡೊಸ್ ಎಕ್ಸ್‌ಪಿ ಅತೀ ಹೆಚ್ಚು ಕಾಲ ಕಾರ್ಯನಿರ್ವಹಿಸಿದ ಮೈಕ್ರೊಸಾಫ್ಟ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಇದು ಪ್ರಮುಖವಾಗಿ ಮೂರು ಬಾರಿ ನವೀಕರಣ ಹೊಂದಿದೆ. ಮತ್ತು ಏಪ್ರಿಲ್ 2014 ರವರೆಗೆ ಎಂದರೆ ಅದರ ಮೂಲ ಬಿಡುಗಡೆಯ ದಿನಾಂಕದಿಂದ 13 ವರ್ಷಗಳವರೆಗೆ ಬೆಂಬಲಿಸಿತು . ಇದರ ಬಳಕೆಯನ್ನು ನಿಲ್ಲಿಸಿದ ಮೇಲೆಯೂ ಇನ್ನೂ ಅಂದಾಜು 430ಮೀ ಪಿಸಿಗಳಲ್ಲಿ ಇದನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತಿತ್ತು.

ಇದರ ಅತಿ ದೊಡ್ಡಸಮಸ್ಯೆ ಭದ್ರತೆಯಾಗಿತ್ತು. ಇದು ಅಂತರ್‌ನಿರ್ಮಿತ ಫೈರವಾಲ್‌ನ್ನು ಹೊಂದಿದ್ದರೂ , ಅದನ್ನು ಪೂರ್ವನಿಯೋಜಿತವಾಗಿ (default) ಆಫ್ ಮಾಡಲಾಗಿತ್ತು. ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಯ ಅತೀ ಜನಪ್ರೀಯತೆ ಹ್ಯಾಕರ್ಗಳಿಗೆ ಮತ್ತು ಅಪರಾಧಗಳಿಗೆ ವರದಾನವಾಯಿತು. ಅವರು ಇಂಟರ್‌ನೆಟ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನ ನ್ಯೂನ್ಯತೆಗಳನ್ನು ಅಪರಾಧಗಳಿಗೆ ಬಳಸಿಕೊಳ್ಳುವುದನ್ನು ನಿರ್ಬಂಧಿಸಲು ಬಿಲ್ ಗೇಟ್ಸ ಅತ್ಯಂತ ಸುರಕ್ಷತೆಯ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವಂತೆ XP service pack updates ಗಳನ್ನು ತಯಾರಿಸಿದರು. ಇವು ಹ್ಯಾಕರ್‌ಗಳಿಗೆ ವಿರುದ್ಧವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲಾರಂಭಿಸಿದವು.

ವಿಂಡೊಸ್ ವಿಸ್ತಾ (Windows Vista)

ವಿಂಡೊಸ್ ವಿಸ್ತಾ, ವಿಂಡೊಸ್ ಎಂಇ ಗಿಂತ ಕೆಟ್ಟದಾಗಿತ್ತು (Windows Vista , arguably worse than Windows ME)

ಜನೆವರಿ 2007 ರಲ್ಲಿ ವಿಂಡೊಸ್ ವಿಸ್ತಾದಿಂದ ಬದಲಾಯಿಸುವ ಮೊದಲು ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ 6 ವರ್ಷಗಳ ಕಾಲ ಕೋರ್ಸನ್ನು ಉಳಿಸಿಕೊಂಡಿತ್ತು. ವಿಂಡೊಸ್ ವಿಸ್ತಾ ತನ್ನ ರೂಪವನ್ನು ನವೀಕರಿಸಿತು ಮತ್ತು ಹೆಚ್ಚಿನ ಗಮನವನ್ನು ವಿಂಡೋಸ್‌ನ ಭಾವನೆ ನೀಡುವ ಪಾರದರ್ಶಕ ಅಂಶಗಳು, ಹುಡುಕಾಟ ಮತ್ತು ಸುರಕ್ಷತೆಗಳ ಕಡೆಗೆ ನೀಡಿತು. ಲಾಂಗಹಾರ್ನ ಎನ್ನುವ ಹೆಸರಿನಿಂದ ಅಬಿ ವೃದ್ದಿಪಡಿಸುವುದಕ್ಕೆ ತೊಂದರೆಯಾಯಿತು ಹಾಗೂ ಇದರ ಉತ್ಪಾದನೆಗಾಗಿ ಮಹತ್ವಾಕಾಂಕ್ಷೆಯ ಅಂಶಗಳನ್ನು ಕೈಬಿಡಲಾಯಿತು.

ಜಾಹೀರಾತು (Advertisement)

ಇದು ದೋಷಯುಕ್ತವಾಗಿತ್ತು. ಬಳಕೆದಾರರು ನೂರಾರು ಬಾರಿ ಬಳಕೆದಾರರ ಖಾತೆ ನಿಯಂತ್ರಣ ಅಡಿಯಲ್ಲಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ಅನುಮತಿಗಾಗಿ ವಿನಂತಿಸಿಕೊಳ್ಳುವಂತಾಗಿ ಹೊರೆಯಾಯಿತು. ಇದರ ಫಲಿತಾಂಶವಾಗಿ ವಿಶ್ವಾಸಾರ್ಹ ಕಂಪ್ಯೂಟಿಂಗ ಉಪಕ್ರಮದಿಂದಾಗಿ ಬಳಕೆದಾರರು ವಿವಿಧ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಲು ಅನುಮೋದಿಸುವುದು ಅಥವಾ ನಿರಾಕರಿಸುವ ಪ್ರಯತ್ನಗಳನ್ನು ಮಾಡಬಹುದಾಗಿತ್ತು.

ಯುಎಸಿಯೊಂದಿಗಿನ ಸಮಸ್ಯೆಯಿಂದಾಗಿ ಎಲ್ಲದಕ್ಕೂ ಹೌದು(yes) ಎಂದು ಕ್ಲಿಕ್ ಮಾಡುವುದರಿಂದ ಜನರು ಸಂತೃಪ್ತರಾದರು. . ಇದು

© NIMI NOT TO BE REPUBLISHED

ಭದ್ರತೆಯನ್ನು ಮತ್ತೆ ಯುಎಸಿ ಪೂರ್ವ ಹಂತಕ್ಕೆ ತೆಗೆದುಕೊಂಡು ಹೋಯಿತು. ವಿಸ್ತಾ ರೆಡೀ ಎನ್ನುವ ಲೇಬಲ್ ಹೊಂದಿದ್ದರೂ ಹಳೆಯ ಕಂಪ್ಯೂಟರ್ ಗಳಲ್ಲಿ ಇದು ಬಹಳ ನಿಧಾನವಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತಿತ್ತು. ವಿಸ್ತಾ ದ ಎಲ್ಲ ಆವೃತ್ತಿಗಳು ಈ ಲೇಬಲ್‌ನಿಂದ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಕೆಲಸ ಮಾಡುತ್ತಿರಲಿಲ್ಲ.

ಪಿಸಿ ಆಟಗಾರರು ವಿಸ್ತಾನಲ್ಲಿ ಮೈಕ್ರೊಸಾಫ್ಟ್ ನ ಡೈರೆಕ್ಟಎಕ್ಸ 10 ತಂತ್ರಜ್ಞಾನವನ್ನು ಹೊಂದಿದ್ದರಿಂದ ಉತ್ತೇಜನವನ್ನು ಕಂಡರು.

ವಿಂಡೋಸ್ ಮೀಡಿಯಾ ಪ್ಲೇಯರ 11, IE 7 ಗಳು ಪ್ರಾರಂಭವಾದವು. ಜೊತೆಗೆ, ಆ್ಯಂಟೀ ಸ್ಪೈವೇರ್ ಆದಂತಹ

ವಿಂಡೋಸ್ ಡಿಫೆಂಡರ್ ಕೂಡ ಪ್ರಾರಂಭವಾಯಿತು. ವಿಸ್ತಾ ಸ್ಪೀಚ್ ರೆಕಗ್ನಿಷನ್ , ವಿಂಡೋಸ್ ಡೀವಿಡಿ ಮೇಕರ್ ಮತ್ತು ಫೋಟೋ ಗ್ಯಾಲರಿಗಳನ್ನು ಹೊಂದಿತ್ತು. ಇದು ಡೀವಿಡಿ ನಲ್ಲಿ ವಿತರಿಸಲಾದ ಮೊದಲ ವಿಂಡೊಸ್ ಆಗಿತ್ತು. ನಂತರ ಆ್ಯಂಟೀ ಟ್ರಸ್ಟ ತನಿಖೆಗಳಿಗಾಗಿ ವಿಂಡೋಸ್ ಮೀಡಿಯಾ ಪ್ಲೇಯರ ಇಲ್ಲದೇ ವಿಂಡೊಸ್ ವಿಸ್ತಾನ ಆವೃತ್ತಿಯನ್ನು ತಯಾರಿಸಲಾಯಿತು.

ವಿಂಡೊಸ್ 7(Windows 7)

ವಿಂಡೊಸ್ 7 ವಿಂಡೊಸ್ ವಿಸ್ತಾ ಇರಬೇಕಾದಂತಹ ಎಲ್ಲವೂ ಇದಾಗಿತ್ತು (Windows 7 was everything Windows VIsta should have been)

ವಿಂಡೊಸ್ ವಿಸ್ತಾ ಏನನ್ನು ಹೊಂದಿರಬೇಕು ಎಂದು ಅನೇಕರು ಬಯಸಿದ್ದರೊ ಅದನ್ನು ವಿಂಡೊಸ್ 7 ಹೊಂದಿತ್ತು. ವಿಂಡೊಸ್ 7 ನ್ನು ಅಕ್ಟೋಬರ್ 2009 ರಲ್ಲಿ ಮೊದಲ ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು. ವಿಸ್ತಾ ಎದುರಿಸಿದ ಎಲ್ಲ ಸಮಸ್ಯೆಗಳು ಮತ್ತು ಟೀಕೆಗಳನ್ನು ಸರಿಪಡಿಸಲು ಉದ್ದ್ಯೇಶಿಸಲಾಗಿತ್ತು. ಅದರ ಕಾಣುವಿಕೆಯಲ್ಲಿ ಸ್ವಲ್ಪ ಬದಲಾವಣೆಯೊಂದಿಗೆ, ಮತ್ತು ಬಳಕೆದಾರ ಸ್ನೇಹಿ ವೈಶಿಷ್ಟ್ಯಗಳೊಂದಿಗೆ ಮತ್ತು ಕಡಿಮೆ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ ಓವರ್‌ಲೋಡ್ ("dialogue box overload") ನೊಂದಿಗೆ ವಿಂಡೊಸ್ 7 ನ್ನು ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು.

ಇದು ವೇಗವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿತ್ತು, ಹೆಚ್ಚು ಸ್ಥಿರವಾಗಿತ್ತು ಮತ್ತು ಬಳಸಲು ಸುಲಭವಾಗಿತ್ತು. ಹೆಚ್ಚಿನ ಬಳಕೆದಾರರು ಮತ್ತು ವ್ಯವಹಾರಗಳು ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಯಿಂದ ವಿಂಡೊಸ್ 7ಗೆ ಅಪ್ ಗ್ರೇಡ ಹೊಂದಿದರು ಮತ್ತು ವಿಸ್ತಾವನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಬಿಟ್ಟರು.

ಕೈಬರಹ ಗುರುತಿಸುವಿಕೆ ವಿಂಡೊಸ್ 7 ನಲ್ಲಿ ಪ್ರಾರಂಭವಾಯಿತು. ವಿಂಡೊಗಳನ್ನು ಪರದೆಯ ಮೇಲಿನಿಂದ ಅಥವಾ ಬದಿಯಿಂದ ಸ್ನ್ಯಾಪ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿತ್ತು , ವಿಂಡೊಗಳನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ವೇಗವಾಗಿ ಮರುಗಾತ್ರಗೊಳಿಸಲು(resize) ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

ವಿಂಡೊಸ್ 8 (Windows 8)

ವಿಂಡೊಸ್ 8 ಕೀಬೋರ್ಡ ಮತ್ತು ಮೌಸ್‌ಗಿಂತ ಟಚ್ ಮೇಲೆ ಹೆಚ್ಚು ಗಮನಹರಿಸಿದೆ (Windows 8 focused more on touch than a keyboard and mouse)

ಇದು ಅಕ್ಟೋಬರ್ 2012 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾಯಿತು. ವಿಂಡೊಸ್ 8 ಇದು ಮೈಕ್ರೊಸಾಫ್ಟನ ಅತ್ಯಂತ ಕೂಲಂಕುಷವಾಗಿ ಪರೀಕ್ಷಿಸಿ ತಯಾರಿಸಿದ ಸಾಫ್ಟವೇರ್ ಆಗಿರುತ್ತದೆ. ಹೆಚ್ಚು ಸ್ಪರ್ಶ ಸ್ನೇಹಿ ಪರದೆಯು, ಸ್ಟಾರ್ಟ ಬಟನ್ ಮತ್ತು ಸ್ಟಾರ್ಟ ಮೆನುಗಳನ್ನು ಹೊಂದಿದೆ.

ಜಾಹೀರಾತು (Advertisement)

ಹೊಸ ಟೈಲ್ಡ ಇಂಟರ್‌ಫೇಸ್ ಪ್ರೋಗ್ರಾಂ ಐಕಾನ್‌ಗಳನ್ನು ಮತ್ತು ಲೈವ್ ಟೈಲ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಪ್ರೋಗ್ರಾಂಗಳ ಪಟ್ಟಿ ಮತ್ತು ಐಕಾನ್‌ಗಳ ಬದಲಾಗಿ ಒಂದೇ ನೋಟದಲ್ಲಿ ವಿಜೆಟ್‌ಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತದೆ. ವಿಂಡೊಸ್ 7 ನ್ನು ಹೋಲುವ ಡೆಸ್ಕಟಾಪನ್ನೂ ಹೊಂದಿದೆ.

ವಿಂಡೊಸ್ 8 ವಿಂಡೊಸ್‌ನ ಹಿಂದಿನ ಆವೃತ್ತಿಗಳಿಗಿಂತಲೂ ವೇಗವಾಗಿದೆ. ಮತ್ತು ಹೊಸ ಯುಎಸ ಬಿ 3.0 ಸಾಧನಗಳನ್ನೂ ಬೆಂಬಲಿಸುತ್ತದೆ.

ಫುಲ್ ಸ್ಕ್ರೀನ್ ಮೋಡ್‌ನಲ್ಲಿ ಮಾತ್ರ ಕಾರ್ಯನಿರ್ವಹಿಸುವ , ಯುನಿವರ್ಸಲ್ ವಿಂಡೊಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಹೊಂದಿರುವ ವಿಂಡೊಸ್ ಸ್ಟೋರ್‌ನ್ನು ಪರಿಚಯಿಸಲಾಯಿತು. ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಥರ್ಡ ಪಾರ್ಟಿಗಳಾದ ವಿಂಡೊಸ್‌ನ ಇತರ ಆವೃತ್ತಿಗಳು, ಇನ್ಸ್ಟಾಲ್ ಮಾಡ್ಬಹುದಾಗಿತ್ತು. ಆದರೆ ಅವು ವಿಂಡೊಸ್‌ನ ಸಾಂಪ್ರದಾಯಿಕ ಡೆಸ್ಕಟಾಪ ಇಂಟರ್‌ಫೇಸನ್ನು ಮಾತ್ರ ಹೊಂದಬಹುದಾಗಿತ್ತು.

ಬಹಳ ಜನರು ವಿಂಡೊಸ್‌ನ ಕೂಲಂಕುಷ ಪರೀಕ್ಷೆಯನ್ನು ಸ್ವಾಗತಿಸಲಿಲ್ಲ. ಮೈಕ್ರೊಸಾಫ್ಟ , ಟಚ್ ಸ್ಕ್ರೀನ್ ಮತ್ತು ಡೆಸ್ಕಟಾಪ್ ಬಳಕೆದಾರರ ಮಧ್ಯೆ ಒಂದು ಉತ್ತಮ ಬಾಂಧವ್ಯ ಹೊಂದಲು ಪ್ರಯತ್ನಿಸಿತು ಆದರೆ ಡೆಸ್ಕಟಾಪ್ ಬಳಕೆದಾರರು ಸಾಂಪ್ರದಾಯಿಕ ಕೀಬೋರ್ಡ ಮತ್ತು ಮೌಸ್‌ನಿಂದ ವಿಂಡೊಸ್‌ನ್ನು ನಿಯಂತ್ರಿಸಲು ಬಯಸಿದರು. ಟಚ್ ಸ್ಕ್ರೀನ್ ಬಳಸುವ ವಿಂಡೊಸ್ 8 ಗೆ ಹಿನ್ನಡೆಯಾಯಿತು.

ಟಚ್ ಸ್ಕ್ರೀನ್‌ಗಳು ತುಂಬಾ ಕಡಿಮೆ ಬಳಕೆಯಲ್ಲಿದ್ದವು. ಸ್ಪರ್ಷ(ಟಚ್) ಆಧಾರಿತ ಇಂಟರ್‌ಫೇಸ್ ಉಪಯೋಗಕಾರಿಂಯಾಗಿದ್ದವು. ಅಗತ್ಯತೆಗೆ ಅನುಸಾರವಾಗಿ ಐಪ್ಯಾಡ್ ಮತ್ತು ಸ್ಮಾರ್ಟಫೋನ್ಗಳಂತಹ ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳ ಅಭಿವೃದ್ಧಿಯಾಯಿತು. 2010 ರಲ್ಲಿ ಟಚ್ ಸ್ಕ್ರೀನ್ ಪಿಸಿಗಳು ಹೆಚ್ಚು ಮಾರಾಟವಾದವು.

ವಿಂಡೊಸ್ ಆರ್‌ಟಿ ಇದು ಎಆರ್‌ಎಮ್ ಆಧಾರಿತ ಪ್ರೊಸೆಸರ್‌ಗಳಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಮಾರ್ಟಫೋನ್‌ಗಳಲ್ಲಿ ಮತ್ತು ಪಿಸಿ ಅಲ್ಲದ ಟ್ಯಾಬ್ಲೆಟಗಳಲ್ಲಿ ಕಂಡುಬರುತ್ತದೆ. ಇದನ್ನು

©NIMI NOT TO BE REPUBLISHED

ಮೈಕ್ರೊಸಾಫ್ಟ ಸರ್ಫೇಸ್ ಟ್ಯಾಬ್ಲೆಟ್ ಹೊಂದಿರುವ ವಿಂಡೊಸ್ 8 ನ ಇದ್ದ ಸಮಯದಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು.

ಇದು ವಿಂಡೊಸ್ 8 ನಂತೆ ಕಾಣುತ್ತಿತ್ತು ಆದರೆ ಸಾಂಪ್ರದಾಯಿಕ ವಿಂಡೋಸ್ ಅಪ್ಲಿಕೇಷನ್ ಗಳು ಕಾರ್ಯನಿರ್ವಸುತ್ತಿರಲಿಲ್ಲ. ಅದರ ಬದಲಾಗಿ ವಿಂಡೊಸ್ ಸ್ಟೋರ್ ಮತ್ತು ಥರ್ಡ ಪಾರ್ಟಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ಮೇಲೆ ಅವಲಂಬಿತವಾಗಿತ್ತು.

ವಿಂಡೊಸ್ 8.1(Windows 8.1)

ವಿಂಡೊಸ್ 8.1 **ಮತ್ತು ಉತ್ತಮ ಸ್ಟಾರ್ಟ ಬಟನ್** (Windows 8.1 and the great reappearance of the Start button)

ವಿಂಡೋಸ್ 8 ನ್ನು ಅಕ್ಟೋಬರ್ 2013 ರಲ್ಲಿ ಪರಿಚಯಿಸಲಾಯಿತು. ನಂತರ ವಿಂಡೋಸ್ 8.1 ಗೆ ಅಪ್‌ಡೇಟ್ ಮಾಡಿ ಮುಂದೆ ಪ್ರತಿವರ್ಷವೂ ಮೈಕ್ರೊಸಾಫ್ಟನಿಂದ ಅಪ್‌ಡೇಟ್‌ಗಳನ್ನು ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು ಮತ್ತು ವಿಂಡೋಸ ಬಳಕೆಯಲ್ಲಿ ಹೊಸತನವನ್ನು ನೀಡುವ ಮೂಲಕ ಮೈಲಿಗಲ್ಲೊಂದನ್ನು ಸ್ಥಾಪಿಸಿದಂತಾಯ್ತು.

ವಿಂಡೊಸ್ 8.1 ಸ್ಟಾರ್ಟ ಬಟನ್‌ನ್ನು ಮತ್ತೆ ಪರಿಚಯಿಸಿತು. ವಿಂಡೊಸ್ 8.1 ನ ಡೆಸ್ಕಟಾಪ್‌ನಿಂದ ಸ್ಟಾರ್ಟ ಸ್ಕ್ರೀನ್‌ನ್ನು ತರಲಾಯಿತು. ಬಳಕೆದಾರರು ನೇರವಾಗಿ ವಿಂಡೊಸ್ 8.1 ನ ಡೆಸ್ಕಟಾಪ್‌ಗೆ ಬೂಟ್ ಮಾಡಬಹುದಾಗಿತ್ತು. ಇದು ಮೌಸ ಮತ್ತು ಕೀಬೋರ್ಡ ಹೊಂದಿರುವ ಡೆಸ್ಕಟಾಪ್ ಕಂಪ್ಯೂಟರ್ ಬಳಸುವವರಿಗೆ ಹೆಚ್ಚು ಸೂಕ್ತವಾಗಿತ್ತು.

ವಿಂಡೊಸ್ 10 (Windows 10)

ವಿಂಡೊಸ್ 10 ನೊಂದಿಗೆ ಮೈಕ್ರೊಸಾಫ್ಟ ವಿಂಡೊಸ್ 8 ಗಾಗಿ ತಯಾರಿಸಿದ ಕೆಲವು ಟಚ್ ಮತ್ತು ಟ್ಯಾಬ್ಲೆಟ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಇರಿಸಿಕೊಳ್ಳಲು ಪ್ರಯತ್ನಿಸುತ್ತಿದೆ. ಪರಿಚಿತವಾದ ಸ್ಟಾರ್ಟ ಮೆನು ಮತ್ತು ಡೆಸ್ಕಟಾಪ್‌ನೊಂದಿಗೆ ಅವುಗಳನ್ನು ಸಂಯೋಜಿಸಿದೆ. ಮತ್ತು ಇವೆಲ್ಲವುಗಳನ್ನು ಸುಧಾರಿತ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ರನ್ ಮಾಡುತ್ತದೆ. , ಹೆಚ್ಚು ಸುರಕ್ಷತೆ, ಹೊಸ ಬ್ರೌಸರ್, ಕೋರ್ಟಾನಾ ಸಹಾಯಕ ಇದರ ಸ್ವಂತ ಆಫೀಸ್ ನ ಆವೃತ್ತಿ ಹಾಗೂ ಸರಳವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಅವಶ್ಯಕವಿರುವ ಸಾಕಷ್ಟು ಹೊಸ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿದೆ. ನೀವು ವಿಂಡೊಸ್ 7, ವಿಂಡೊಸ್ 8, ಅಥವಾ ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಬಳಸಿದ್ದರೂ , ಸಹಜವಾಗಿ ಇದನ್ನು ಬಳಸಲು ತುಂಬಾ ವಿಭಿನ್ನವಾಗಿದೆ. ನಿಮ್ಮ ಪಿಸಿ ಆಫ್ ಮಾಡುವಾಗ ಬೇರೆ ರೀತಿಯಿಂದ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

ಇದೆಲ್ಲದರ ಮೇಲೆ , ವಿಂಡೊಸ್ 10 ಕೇವಲ ಪಿಸಿ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ಗಿಂತ ಹೆಚ್ಚಾಗಿದೆ. ವಿಂಡೊಸ್ ಫೋನ್ ಗಳಲ್ಲಿ , ಸಣ್ಣ ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳಲ್ಲಿಯೂ ಕೂಡ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಯಾಕೆಂದರೆ 6 ಇಂಚಿನ ಫೋನ್ ಮತ್ತು 7 ಇಂಚಿನ ಟ್ಯಾಬ್ಲೆಟ್ ತುಂಬಾ ಭಿನ್ನವಾಗಿಲ್ಲ. ಜನರು ವಿಂಡೊಸ್ 10 ನ್ನು ಬಿಲಿಯನ್‌ಗಟ್ಟಲೆ ಸಾಧನಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಬೇಕೆಂದು ನಿರೀಕ್ಷಿಸಿದೆ.(ಹೆಚ್ಚಿನ ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪರ್ ಗಳನ್ನು ಪ್ರೋತ್ಸಾಹಿಸಬೇಕಾಗಿದೆ. ವಿಂಡೊಸ್ ಫೋನ್ಗಳಿಗಾಗಿ ಮತ್ತು ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳಿಗಾಗಿ ಹಾಗೆಯೆ ಎಕ್ಸ್ ಬಾಕ್ಸ ಒನ್ ಮತ್ತು ಹೊಲೊಲೆನ್ಸ್ ಗಳಿಗಾಗಿ ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ನಿರ್ಮಿಸಲು ಪ್ರೋತ್ಸಾಹಿಸಬೇಕಾಗಿದೆ)

ಸ್ಟಾರ್ಟ ಮೆನು ವಿಕಸನಗೊಳ್ಳುತ್ತದೆ. (The Start menu evolves)

ವಿಂಡೊಸ್ 8 ರ ಫುಲ್ ಸ್ಕ್ರೀನ್ ಸ್ಟಾರ್ಟ ಸ್ಕ್ರೀನ್ ವಿಂಡೊಸ್ 10 ರಲ್ಲಿ ಸ್ಟಾರ್ಟ ಮೆನು ಆಗಿ ಬಂದಿತು. ಇದು ಎರಡು ಆಯ್ಕೆಗಳಲ್ಲಿನ ಉತ್ತಮವಾದದ್ದನ್ನು ಸಂಯೋಜಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತದೆ. ಒಂದೇ ಕಾಲಮ್ ಗೆ ನಿರ್ಬಂಧಿಸಿದ ಸ್ಕ್ರೋಲಿಂಗ ಹೊಂದಿರುವ ಸ್ಟಾರ್ಟ ಮೆನು ಹೊಂದಿದೆ. ಜಂಪ್ ಲಿಸ್ಟ ಮತ್ತು ಹೆಚ್ಚಿನ ಆಯ್ಕೆಗಳಿಗಾಗಿ ಫ್ಲೈ ಔಟ್ ಮೆನುಗಳನ್ನು ಹೊಂದಿದೆ. ಮೇಲಿಂದ ಮೇಲೆ ಉಪಯೋಗಿಸಿದ ಪ್ರೋಗ್ರಾಂಗಳಿಗಾಗಿ ಮತ್ತು ಇತ್ತೀಚೆಗೆ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಹೊಂದಿದೆ. ವರ್ಣಮಾಲೆಯಂತೆ ವಿಂಗಡಿಸಲಾದ ಎಲ್ಲ ಅಪ್ಲಿಕೇಷನ್ ಗಳ ಸ್ಕ್ರಾಲಿಂಗ ವೀಕ್ಷಣೆಗೆ ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

ಆದರೆ ಇಲ್ಲಿ ಹೆಚ್ಚುವರಿ ಫಲಕವನ್ನು ಪಡೆಯಬಹುದು ಹಾಗೂ ಅಲ್ಲಿ ವಿಂಡೋಸ್ 8 ಸ್ಟೈಲ್ ಟೈಲ್‌ಗಳನ್ನು ಪಿನ್ ಮಾಡಬಹುದು. `ಈ ಟೈಲ್ಸ್‌ಗಳನ್ನು 3 ಡಿ ಕ್ಯೂಬ್ ರೀತಿಯಲ್ಲಿ ಚಲಿಸುವಂತೆ ಮಾಡಬಹುದು. ಸ್ಟಾರ್ಟ ಮೆನುವನ್ನು ದೊಡ್ಡ ಪ್ರಮಾಣದಲ್ಲಿ ಹಿಗ್ಗಿಸಬಹುದು ಅಥವಾ ಪೂರ್ತಿ ಸ್ಕ್ರೀನ್ ತುಂಬ ಕಾಣುವಂತೆ ಮಾಡಬಹುದು.

ಡೆಸ್ಕಟಾಪ್ ಹಿನ್ನೆಲೆ(Desktop Background)

ಡೆಸ್ಕಟಾಪ್‌ನ ಮತ್ತೊಂದು ಘಟಕವೆಂದರೆ ಹಿನ್ನೆಲೆ (Background) ಇದು ಸ್ಕ್ರೀನ್‌ನ ಹಿಂಭಾಗದಲ್ಲಿ ಕಂಡುಬರುವ ಚಿತ್ರವಾಗಿದೆ. ಹೆಚ್ಚಿನ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮೊದಲೇ ಆಯ್ಕೆ ಮಾಡಿದ ಹಿನ್ನೆಲೆ ಹೊಂದಿರುತ್ತವೆ. ಆದರೆ ಇದನ್ನು ಬೇರೆ ಚಿತ್ರದಿಂದ ಬದಲಾಯಿಸಬಹುದು.

Fig 1

© NIMI NOT TO BE REPUBLISHED

ಹಿನ್ನೆಲೆ ಬದಲಾಯಿಸಲು , ಈ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿ

1 ಹಿನ್ನೆಲೆ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು ಪರ್ಸನಲೈಸ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿ.

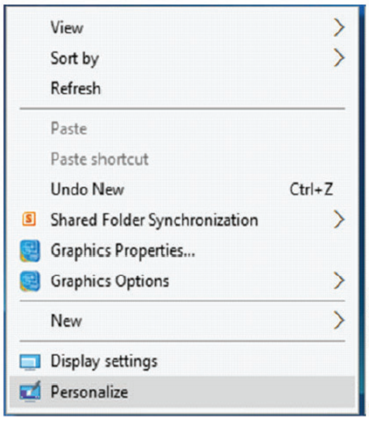
2 ಪರ್ಸನಲೈಸೆಷನ್ ವಿಂಡೊದಿಂದ, ಮೊದಲೆ ಆಯ್ಕೆ ಮಾಡಿದ ಚಿತ್ರಗಳ ಸರಣಿಯಲ್ಲಿ ನಿಮಗೆ ಬೇಕಾದ ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿ ಅಥವಾ ಸ್ವಂತವಾಗಿ ಬ್ರೌಸ್ ಮಾಡಿ.

ಚಿತ್ರ ಆಯ್ಕೆ ಮಾಡಿದ ನಂತರ , ಹಿನ್ನೆಲೆ ತಾನಾಗಿಯೆ ಬದಲಾಗುತ್ತದೆ.

ಸ್ಟಾರ್ಟ ಮೆನು (Start Menu)

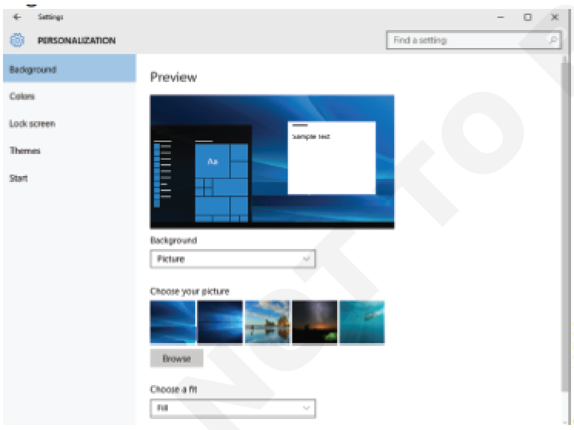
ನಿರ್ದಿಷ್ಟ ಅಪ್ಲಿಕೇಷನ್‌ಗಾಗಿ ಹುಡುಕುತ್ತಿದ್ದರೆ , ಸ್ಟಾರ್ಟ ಮೆನು ತೆರೆಯಿರಿ ಮತ್ತು ಆಲ್ ಅಪ್ಲಿಕೇಷನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ. ಇದು ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ವರ್ಣಮಾಲೆಯ ಪಟ್ಟಿ ಯಂತೆ ಜೋಡಿಸಿರುವ ಎಲ್ಲ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ತೆರೆಯುತ್ತದೆ.

Fig 2



Copa 13702

Fig 3



Copa 13703

ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ (File Explorer)

ನೀವು ನಿರ್ದಿಷ್ಟವಾದ ದಾಖಲೆಯನ್ನು ಹುಡುಕುತ್ತಿದ್ದರೆ, ಅದಕ್ಕಾಗಿ ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಟಾಸ್ಕಬಾರ್‌ನಲ್ಲಿರುವ ಫೋಲ್ಡರ್ ಐಕಾನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಹುಡುಕಬಹುದು.

ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ವಿಂಡೊನಲ್ಲಿ, ಎಲ್ಲ ಫೋಲ್ಡರ್ ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕಿ.

ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗಳು (Virtual Desktops)

ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗಳ ಸೇರ್ಪಡೆ ವಿಂಡೋಸ್ 10 ರ ಹೊಸ ವೈಶಿಷ್ಟ್ಯಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಡೆಸ್ಕಟಾಪ್ ಸ್ಕ್ರೀನ್‌ಗಳನ್ನು ಹೊಂದಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ವಿಂಡೋಗಳನ್ನು ವ್ಯವಸ್ಥಿತವಾಗಿ ಇರಿಸಬಹುದು.

ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್ ಹೊಂದಲು , ಈ ಕೆಳಗಿನ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿ.

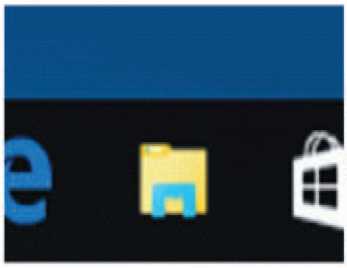
1 ಟಾಸ್ಕಬಾರ್‌ನಲ್ಲಿರುವ ಟಾಸ್ಕ ವ್ಯೂ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ

2 ಕೆಳಗಿನ ಬಲ ಮೂಲೆಯಲ್ಲಿರುವ ನ್ಯೂ ಡೆಸ್ಕಟಾಪ್ ಆಯ್ಕೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ. ಟಾಸ್ಕ ವ್ಯೂ ಮೇಲೆ ಮತ್ತೆ ಕ್ಲಿಕ್ ಮಾಡಿ ನೀವು ಹೊಸ ಡೆಸ್ಕಟಾಪ್‌ನ್ನು ಪಡೆಯಬಹುದು(Access) ಅಥವಾ ಅಳಿಸಬಹುದು(Delete)

Fig 4

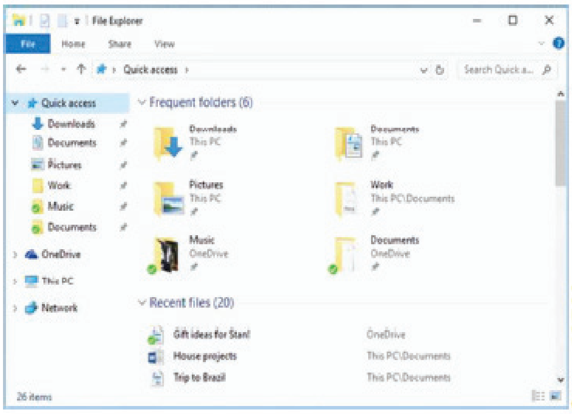
Copa 13704

Fig 5



Copa 13705

Fig 6



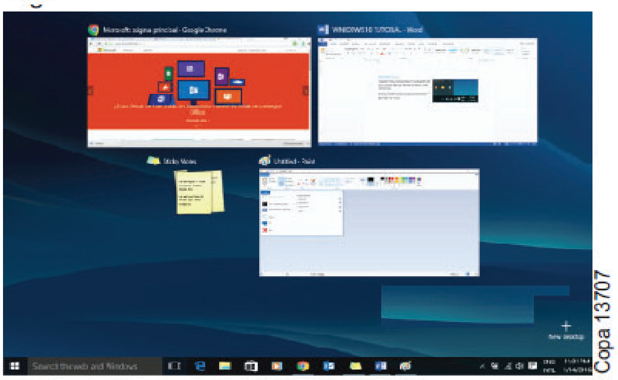
Copa 13706

© NIMI NOT TO BE REPUBLISHED

ಹುಡುಕಾಟ ಮತ್ತು ನಿಯಂತ್ರಣ ಮಾಡಲು ಕೊರ್ಟಾನ ಸಹಾಯ ಮಾಡುತ್ತದೆ(Cortana helps as with search and control)

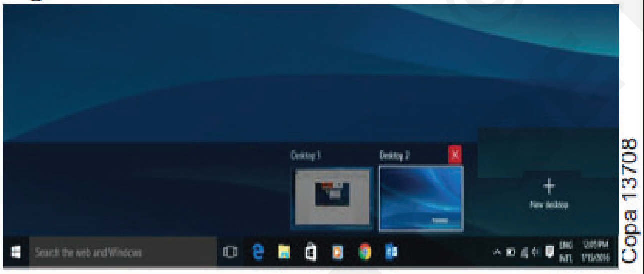
ವಿಂಡೋಸ್ ಫೋನ್ ಸಹಾಯಕರಾದ ಕೋರ್ಟಾನಾ ಟಾಸ್ಕಬಾರ್‌ನಲ್ಲಿ ವಿಂಡೊಸ್ 10 ನಲ್ಲಿ ಸರ್ಚ ಪೇನ್ ಆಗಿ ತೊರಿಸುತ್ತದೆ,. ಹೇ ಕೋರ್ಟಾನಾ ಎಂದು ಹೇಳುವ ಮೂಲಕ ಪ್ರಾರಂಭ ಮೆನು(start menu) ಹುಡುಕಾಡಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ.

Fig 7



ಇದು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುತ್ತದೆ. ಹಾಗೂ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆಯುತ್ತದೆ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಸ್ಟೋರ್‌ನಿಂದ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುತ್ತದೆ. ವೆಬ್‌ನಿಂದ ಹುಡುಕಾಟದ ಫಲಿತಾಂಶಗಳನ್ನು ಪಡೆಯುವುದು, ಕೋರ್ಟಾನಾ ಜೊತೆಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಮತ್ತು ಸೇವೆಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಬೇರೆ ಇತರ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ.

Fig 8



ಕೋರ್ಟಾನಾವನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುವುದು (Activating Cortana)

ಒಂದು ವೇಳೆ ಕೋರ್ಟಾನಾ ಸಕ್ರಿಯವಾಗಿಲ್ಲದಿದ್ದರೆ, ಟಾಸ್ಕ ಬಾರ್‌ನಲ್ಲಿ ಕೊರ್ಟಾನಾ ಎಂದು ಟೈಪ್ ಮಾಡುವ ಮೂಲಕ ಅದನ್ನು ಆನ್ ಮಾಡಬಹುದು. ಕೋರ್ಟಾನಾ ಸೆಟ್ಟಿಂಗ್ಸನ್ನು ಹುಡುಕಿ ಅಥವಾ ಬರಿ ಮೆನುವಿನ ಎಡ ಭಾಗದಲ್ಲಿರುವ ಗಿಯರ್ ಐಕಾನನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಕೋರ್ಟಾನಾವನ್ನು ಆನ್ ಮಾಡಬಹುದು.

ಕೋರ್ಟಾನಾವನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಿದ ನಂತರ ಅದು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ.

ಟಾಸ್ಕ ಸ್ವಿಚರ್ (Task Switcher)

ಹೆಚ್ಚಿನ ವಿಂಡೊಸ್ ಬಳಕೆದಾರರು ಅಲ್ಟ - ಟ್ಯಾಬ್ ಕೀಬೋರ್ಡ ಸಂಯೋಜನೆ ಬಗ್ಗೆ ತಿಳಿದಿಲ್ಲ. ಇದನ್ನು , ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ನೋಡಲು ಮತ್ತು ಬದಲಾಯಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ದೊಡ್ಡದಾದ ಥಂಬನೈಲ್‌ಗಳನ್ನು ಹೊಂದಿದ ಮರುವಿನ್ಯಾಸ ಗೊಂಡ ಟಾಸ್ಕ ಸ್ವಿಚರ್‌ನ್ನು ಹೊಂದಿದೆ. ವಿಂಡೊಸ್ 10 ಕೂಡ ಟಾಸ್ಕಬಾರ್ ನಲ್ಲಿ , ಟಾಸ್ಕ ವ್ಯೂ ಐಕಾನನ್ನು ಹೊಂದಿದೆ.

Fig 9

Fig 10

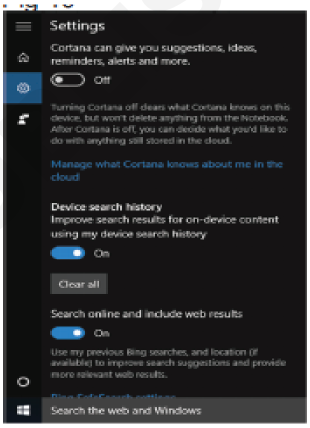
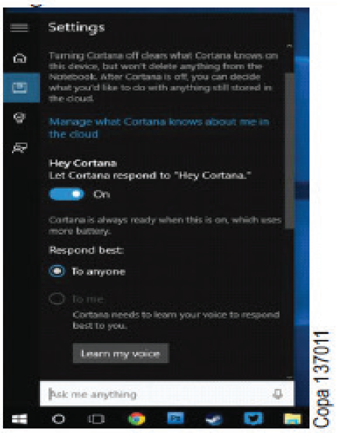


Fig 11



ಟಾಸ್ಕಬಾರ್ (TASK BAR)

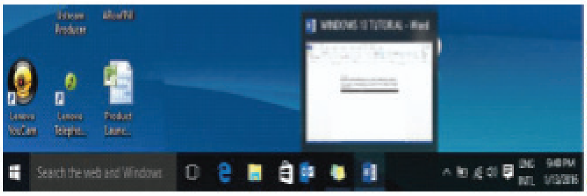
ಸ್ಕ್ರೀನ್‌ನ ಕೆಳಭಾಗದಲ್ಲಿ ವಿಂಡೋಸ್ 10 ನ ಟಾಸ್ಕಬಾರ್ ಇರುತ್ತದೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ಸ್ಟಾರ್ಟಮೆನುಗೆ ಪ್ರವೇಶ ನೀಡುತ್ತದೆ. ಜೊತೆಗೆ ಆಗಾಗ್ಗೆ ಬಳಸುವ ಅಪ್ಲಿಕೇಷನ್ ಗಳ ಐಕಾನ್‌ಗಳನ್ನು ಉಪಯೊಗಿಸಲು ಪ್ರವೇಶ ನೀಡುತ್ತದೆ. ಟಾಸ್ಕಬಾರ್ ನ ಬಲಭಾಗದಲ್ಲಿ ನೊಟೀಫಿಕೇಶನ್ ಏರಿಯಾ ಇರುತ್ತದೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ವಿವಿ

© NIMI NOT TO BE REPUBLISHED

, ನ್ನ ವಿಷಯಗಳ ಬಗ್ಗೆ ಎಂದರೆ ಇಂಟರ್‌ನೆಟ್ ಕನೆಕ್ಷನ್ ಸ್ಥಿತಿ ಮತ್ತು ಲ್ಯಾಪ್ ಟಾಪ್ ನ ಬ್ಯಾಟರಿ ಚಾರ್ಜ ಮಾಹಿತಿ ನೀಡುತ್ತದೆ.

ಟಾಸ್ಕಬಾರ್‌ನ ಮಧ್ಯದಲ್ಲಿರುವ ಐಕಾನ್‌ಗಳು ಪಿನ್ ಮಾಡಲಾದ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಾಗಿರುತ್ತವೆ. ಇದರಿಂದ ನೀವು ಮೇಲಿಂದ ಮೇಲೆ ಉಪಯೋಗಿಸುವ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಸುಲಭವಾಗಿ ತೆರೆದು ಉಪಯೋಗಿಸಬಹುದು. ಪಿನ್ ಮಾಡಲಾದ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು, ಅನ್ ಪಿನ್ ಮಾಡುವವರೆಗೂ ಅವು ಟಾಸ್ಕಬಾರ್‌ನಲ್ಲಿಯೇ ಇರುತ್ತವೆ.

Fig 12



Copa 137012

ಅಪ್ಲಿಕೇಷನ್ನು ಟಾಸ್ಕಬಾರ್‌ಗೆ ಪಿನ್ ಮಾಡುವುದು.

ಹಂತ 1: ನೀವು ಸ್ಟಾರ್ಟ ಬಟನ್‌ನಲ್ಲಿ ಪಿನ್ ಮಾಡಲು ಬಯಸುವ ಅಪ್ಲಿಕೇಷನ್ನು ಹುಡುಕಿ

ಹಂತ 2 : ಅಪ್ಲಿಕೇಷನ್ ಮೇಲೆ ಬಲ ಮೌಸ್ ಬಟನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ

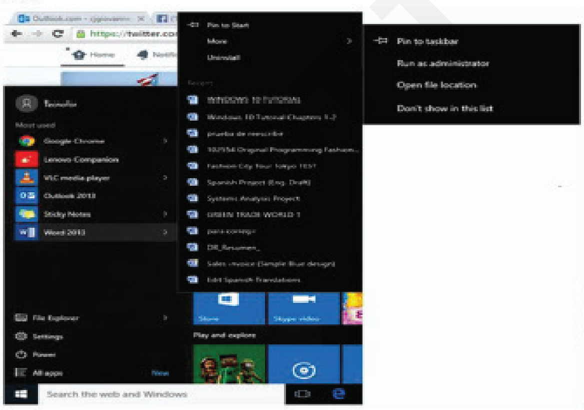
ಹಂತ 3: ಮೆನುವಿನ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ ಮೋರ್ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿ

ಹಂತ 4: ಪಿನ್ ಟು ಟಾಸ್ಕಬಾರ್ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿ

ಅಪ್ಲಿಕೇಷನ್ನು ಟಾಸ್ಕಬಾರ್‌ನಿಂದ ಅನ್‌ಪಿನ್ ಮಾಡುವುದು

ಅಪ್ಲಿಕೇಷನ್ನು ಅನ್ ಪಿನ್ ಮಾಡಲು , ಟಾಸ್ಕಬಾರ್‌ನಲ್ಲಿರುವ ಐಕಾನ್ ಮೇಲೆ ಬಲ ಮೌಸ್ ಬಟನ್ ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು ಅನ್ ಪಿನ್ ಫ್ರಾಮ್ ಟಾಸ್ಕಬಾರ್ ``Unpin from Taskbar`` ಆರಿಸಿ. ನಿಮಗೆ ಬೇಕಾದಾಗ ಮತ್ತೆ ನೀವು ಅದನ್ನು ಪಿನ್ ಮಾಡಬಹುದು.

Fig 13



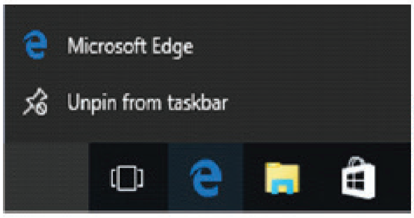
Copa 137013

ನೋಟಿಫಿಕೇಷನ್ ಏರಿಯಾ (Notification Area)

ನೋಟಿಫಿಕೇಷನ್ ಏರಿಯಾ ಟಾಸ್ಕಬಾರ್‌ನ ಬಲ ಭಾಗದಲ್ಲಿರುತ್ತದೆ. ಇದು ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನ ಇಂಟರ್‌ನೆಟ್ ಕನೆಕ್ಷನ್ ಅಥವಾ ವಾಲ್ಯೂಮ್ ಲೆವೆಲ್ ಮುಂತಾದ ವಿವಿಧ ರೀತಿಯ ನೋಟಿಫಿಕೇಷನ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.

ಮೊದಲು , ನೋಟಿಫಿಕೇಷನ್ ಏರಿಯಾ ಸೀಮಿತ ಐಕಾನ್‌ಗಳನ್ನು ತೊರಿಸುತ್ತದೆ. ಆದರೆ ನೀವು ಎಡ ಭಾಗದಲ್ಲಿರುವ ಮೇಲ್ಮುಖವಾದ ಬಾಣದ ಚಿನ್ಹೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಉಳಿದ ಐಕಾನ್‌ಗಳನ್ನು ನೋಡಬಹುದು.

Fig 14

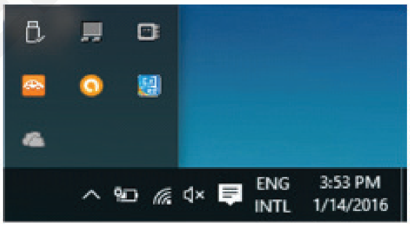


Copa 137014

ಸ್ನ್ಯಾಪ್ ಅಸಿಸ್ಟ(Snap Assist)

ವಿಂಡೊಸ್‌ನ ಎಲ್ಲ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಮತ್ತು ಪ್ರೋಗ್ರಾಂಗಳು ಡೆಸ್ಕಟಾಪ್ ಮೇಲೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ. ಹಳೆಯ ವಿಂಡೊಸ್ ಆವೃತ್ತಿ (older version) ಯಲ್ಲಿ ಇರುವಂತೆ , ಯಾವುದೇ ಅಪ್ಲಿಕೇಷನನ್ನು ತೆರೆಯಬೆಕಾದರೆ ಸ್ಟಾರ್ಟ ಮೆನು , ಪ್ರೋಗ್ರಾಂ ಮೆನು ಮತ್ತು ಅದರಲ್ಲಿನ ಅಪ್ಲಿಕೇಷನ್ ತೆರೆಯುವ ವಿಧಾನದ ಬದಲಾಗಿ ಒಂದು ದೊಡ್ಡದಾದ ವಿಂಡೊ ತೆರೆದು ಅದರಲ್ಲಿ ಉಪಯೋಗಿಸುವ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ಸ್ಪ್ಲಿಟ್ ವ್ಯೂನಲ್ಲಿ ನೋಡುವುದನ್ನು ಸ್ನ್ಯಾಪ್ ವ್ಯೂ ಎನ್ನುತ್ತಾರೆ.

Fig 15



Copa 137015

ಈ ಸ್ನ್ಯಾಪ್ ವ್ಯೂನಲ್ಲಿ ಎಲ್ಲ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಥಂಬ್ ನೇಲ್ಸ್ ರೂಪದಲ್ಲಿದ್ದು , ಬೇಕಾದ ಅಪ್ಲಿಕೇಷನ್ನು ತೆರೆಯಲು ಅನುಕೂಲಕರವಾಗಿರುತ್ತದೆ.

ಆ್ಯಕ್ಷನ್ ಸೆಂಟರ್(Action Center)

ಒಂದು ವೇಳೆ ನಾವು ವಿಂಡೋಸ್ ಫೋನ್ 8.1 ನ್ನು (ಅಥವಾ ಆಂಡ್ರಾಯ್ಡ ಮತ್ತು /ಅಥವಾ ಐ ಓ ಎಸ್) ನ್ನು ಉಪಯೋಗಿಸಿದ್ದರೆ ನಾವು ಸ್ಕ್ರೀನ್ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ ನೋಟಿಫಿಕೇಷನ್ ಸೆಂಟರ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ. ವಿಂಡೊಸ್ 10 ಇದನ್ನು ಸ್ಕ್ರೀನ್‌ನ ಬಲಭಾಗದಲ್ಲಿರಿಸುತ್ತದೆ. ವಿಂಡೊಸ್ 8 ರಲ್ಲಿ ಚಾರ್ಮ್ಸ ಬಾರ್ ಸ್ಕ್ರೀನ್‌ನ ಬಲಭಾಗದಲ್ಲಿರುತ್ತದೆ. ಹಾಗೂ ವಿವಿಧ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ನೋಟಿಫಿಕೇಷನ್‌ಗಳು ಮೇಲ್ಭಾಗದಲ್ಲಿ ಮತ್ತು ವಿಭಿನ್ನ ಸೆಟ್ಟಿಂಗಗಳ ಆಯ್ಕೆ ಬಟನ್ ಗಳು ಕ್ವಿಕ್ ಎಕ್ಸೆಸ್‌ನ ಕೆಳಭಾಗದಲ್ಲಿರುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಕಮಾಂಡ ಪ್ರಾಂಪ್ಟ (Command Prompt)

ನಮ್ಮಲ್ಲಿ ಕಮಾಂಡ ಪ್ರಾಂಪ್ಟ ಬಳಸುವವರು ಇದ್ದೇವೆ. 1990 ರ ದಶಕದಿಂದ ಅದೇ ಅನುಭವದೊದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿದೇವೆ. ಆದರೆ ವಿಂಡೋಸ್ 10 ರಲ್ಲಿ ಕಮಾಂಡ ಪ್ರಾಂಪ್ಟನ ವಿಂಡೋನ್ನು ಮರುಗಾತ್ರಗೊಳಿಸಬಹುದು. ಮತ್ತು ಕಮಾಂಡ ಪ್ರಾಂಪ್ಟನಲ್ಲಿ ಕಾಪಿ ಮತ್ತು ಪೇಸ್ಟ ಮಾಡಲು ಪರಿಚಿತವಾದಂತಹ ಕೀಬೋರ್ಡ ಶಾರ್ಟಕಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

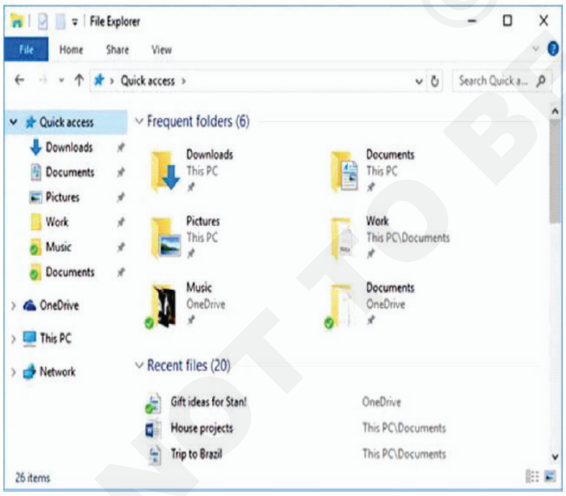
ಹಲವಾರು ವರ್ಷಗಳ ನಂತರ ಇದು ಸ್ವಾಗತಾರ್ಹ ಸುಧಾರಣೆಯಾಗಿತ್ತು.

ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ (File Explorer)

ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ಎನ್ನುವುದು ಫೈಲ್ ಮ್ಯಾನೇಜಮೆಂಟ ಅಪ್ಲಿಕೇಷನ್ ಆಗಿದೆ. ಇದು ಫೋಲ್ಡರ್ ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡಲು ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ಗಳಿಂದ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಹುಡುಕಲು ಮತ್ತು ತೆರೆಯಲು ಗ್ರಾಫಿಕಲ್ ಇಂಟರ್‌ಫೇಸನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಟಾಸ್ಕಬಾರನಲ್ಲಿರುವ ಫೋಲ್ಡರ್ ಐಕಾನನ್ನು ಕ್ಲಿಕ್ ಮಾಡುವುದರ ಮೂಲಕ ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನ್ನು ಪ್ರವೇಶಿಸಬಹುದು ಐಕಾನ್ ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿದ ನಂತರ ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ವಿಂಡೊ ತೆರೆದುಕೊಳ್ಳುತ್ತದೆ.

Fig 16



Copa 137015

ಆರಂಭಿಕ ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ವಿಂಡೊ ಈ ಕೆಳಗಿನ ವಿಭಾಗಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

1 **ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ರಿಬ್ಬನ್** :ಇದು ಮೈಕ್ರೋಸಾಫ್ಟ ಆಫೀಸ್‌ನ ರಿಬ್ಬನ್‌ನ್ನು ಹೋಲುತ್ತದೆ. ಈ ರಿಬ್ಬನ್ ಫೈಲ್ ಮತ್ತು ಫೋಲ್ಡರ್‌ನ ಸಾಮಾನ್ಯ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಬಟನ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ.

2 **ನ್ಯಾವಿಗೇಷನ್ ಪೇನ್** : ಇದು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್ ಲೈಬ್ರರಿಗೆ ಮತ್ತು ಚಿತ್ರಗಳಿಗೆ ಹಾಗೆಯೆ ನಿಮ್ಮ ಸಂಗ್ರಹ ಮಾಧ್ಯಮಗಳನ್ನು ತೆರೆಯಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು ಮೇಲಿಂದ ಮೇಲೆ ಬಳಸುವ ಫೋಲ್ಡರ್‌ಗಳು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

3 **ಫ್ರಿಕ್ವೆಂಟ ಫೋಲ್ಡರ್ ಸೆಕ್ಷನ್** : ನೀವು ಇತ್ತೀಚೆಗೆ ಉಪಯೋಗಿಸಿದ ಫೋಲ್ಡರ್‌ಗಳು ಬೇಗನೆ ಸಿಗುವಂತೆ ಮಾಡುತ್ತದೆ.

4 **ರೀಸೆಂಟ್ ಫೈಲ್ಸ ಸೆಕ್ಷನ್** : ಇದು ವಿಂಡೊನ ಕೆಳಭಾಗದಲ್ಲಿರುತ್ತದೆ ಮತ್ತು ನೀವು ಇತ್ತೀಚೆಗೆ ತೆರೆದು ಉಪಯೊಗಿಸಿದ ಫೈಲ್ ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು ಇದರಲ್ಲಿರುತ್ತವೆ.

ನ್ಯೂ ಎಡ್ಜ ಬ್ರೌಸರ್ (The new Edge browser)

ಮೈಕ್ರೊಸಾಫ್ಟ ತನ್ನ ಹಳೆಯ ಬ್ರೌಸರನ್ನು ಅಭಿವೃದ್ದಿಪಡಿಸಿ ಕೋಡ್‌ಗಳನ್ನು ಸಂಸ್ಕರಿಸಿ ಆಧುನಿಕ ಬ್ರೌಸರ್‌ಗಳಾದ ಕ್ರೋಮ್ (Chrome) ಮತ್ತು ಫೈರ್‌ಫಾಕ್ಸ್ (Firefox) ಗಳಂತೆ ವೇಗವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವ ಗುಣಮಟ್ಟದ ಬ್ರೌಸರನ್ನು ತಯಾರಿಸಿತು.

ಈ ಕೆಲಸ ಪ್ರಗತಿಯಲ್ಲಿದೆ - ವಿಂಡೊಸ್ 10 ಲಾಂಚ್ ಆದ ನಂತರ ಜಾಹೀರಾತು ನಿರ್ಬಂಧಿಸುವ ವಿಸ್ತರಣೆಯತಹ ವಿಷಯಗಳಿಗೆ ಬೆಂಬಲ ಸಿಗಲಿಲ್ಲ. ಆದರೆ ಇಲ್ಲಿ ಸಾಕಷ್ಟು ಅಚ್ಚುಕಟ್ಟಾದ ಕೆಲಸಗಳು ನಡೆಯುತ್ತವೆ. ಉದಾಹರಣೆಗೆ ಸ್ನೇಹಿತರಿಗೆ ಕಳುಹಿಸುವ ವೆಬ್ ಪುಟದಲ್ಲಿ ಟಿಪ್ಪಣಿಗಳನ್ನು ಬರೆಯಬಹುದು. ಮತ್ತು ಎಡ್ಜ ಅಂತರ್ ನಿರ್ಮಿತ ಕೋರ್ಟಾನಾ ಹೊಂದಿರುವುದರಿಂದ , ರೆಸ್ಟೊರೆಂಟನ ಫೋನ್ ಸಂಖ್ಯೆ ಅಥವಾ ಪ್ರಾರಂಭದ ಸಮಯ ಮುಂತಾದ ಉಪಯುಕ್ತ ಮಾಹಿತಿಯನ್ನು ವೆಬ್ ಪುಟಗಳಿಂದ ಪಡೆಯಬಹುದು.

ಐ ಇ ನಲ್ಲಿ ಸರಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸದ ಮಧ್ಯಮ ಸೈಟ್‌ಗಳು ಎಡ್ಜ್‌ನಲ್ಲಿ ಉತ್ತಮವಾಗಿ ಕಾಣುತ್ತವೆ ಮತ್ತು ಹೆಚ್ಚಿನ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿವೆ.

ಮಲ್ಟಿಪಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗಳು (Multiple Desktops)

ಬಳಕೆದಾರರು ಬಹಳಷ್ಟು ವಿಂಡೋಗಳನ್ನು ಹೊಂದಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ ಮತ್ತು ಬಹಳ ಮಾನಿಟರ್‌ಗಳನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ, ಬಳಕೆದಾರರು ಈ ವಿಂಡೊಗಳನ್ನು ಅನೇಕ ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗಳಲ್ಲಿ ಇರಿಸಬಹುದು. ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಬದಲಾಯಿಸಲು ಅಲ್ಟ + ಟ್ಯಾಬ್(Alt + Tab) ನ್ನು ಬಳಸಬಹುದು. ವಿಂಡೋಸ್-ಕಂಟ್ರೋಲ್ ಮತ್ತು ಎಡ ಮತ್ತು ಬಲ ಬಾಣದ ಕೀಗಳನ್ನು ಡೆಸ್ಕಟಾಪ್‌ಗಳನ್ನು ಬದಲಾಯಿಸಲು ಉಪಯೋಗಿಸಬಹುದು.

ಶೆಡ್ಯೂಲ್ ರೀಸ್ಟಾರ್ಟ (Schedule restarts)

ಯಾವುದೇ ವಿಂಡೋಸ್‌ನ ಆವೃತ್ತಿಗಳು ಅಪ್‌ಡೇಟನ್ನು ಪ್ರಾರಂಭಿಸುವ 15 ನಿಮಿಷ ಮೊದಲು ನಿಮ್ಮನ್ನು ಕೆಲಸ ಪೂರ್ಣಗೊಳಿ

© NIMI NOT TO BE REPUBLISHED

ಸಲು ಕೇಳುತ್ತಿರಲಿಲ್ಲ. ಆದರೆ ವಿಂಡೋಸ್ 10 ಅಪ್ ಡೇಟ್ ಮಾಡಲು 15 ನಿಮಿಷದ ಸಮಯಾವಕಾಶವನ್ನು ನೀಡುತ್ತದೆ ನಂತರ ರೀಸ್ಟಾರ್ಟ ಆಗುತ್ತದೆ. ಬಳಕೆದಾರರು ಈ ಸಮಯಾವಕಾಶವನ್ನು ಬೇರೆ ಅವಧಿಗೂ ಬದಲಾಯಿಸಲು ಸೂಚಿಸಬಹುದು. ಇ ದರಿಂದ ಕಾರ್ಯನಿರ್ವಹಿಸುವಾಗ ರೀಸ್ಟಾರ್ಟ ಮಾಡುವುದನ್ನು ಮುಂದೂಡಬಹುದು. ಅಪ್‌ಡೇಟನ್ನು ಡೌನ್‌ಲೋಡ ಮಾಡಿದ ನಂತರವೇ ಬಳಕೆದಾರರು ಅದನ್ನು ಬಳಸಬಹುದು.

ಆಫೀಸ್ ಸೇರಿದಂತೆ - ಯುನಿವರ್ಸಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು (Univarsal apps - including Office)

ವಿಂಡೋಸ್ 10 ಹೊಸ ವಿಂಡೋಸ್ ಸ್ಟೋರನ್ನು ಹೊಂದಿದೆ. ಇಲ್ಲಿ ಡೆಸ್ಕಟಾಪ್ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಮತ್ತು ಆಧುನಿಕ ವಿಂಡೋಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಡೌನ್‌ಲೋಡ ಮಾಡಬಹುದು. ಈ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಬಹಳಷ್ಟು ಯುನಿವರ್ಸಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಾಗಿವೆ. ಇವು ಪಿಸಿಗಳಲ್ಲಿ, ವಿಂಡೋಸ್ ಫೋನ್ ಗಳಲ್ಲಿ ಎಕ್ಸ್ ಬಾಕ್ಸ್ ಒನ್‌ನಲ್ಲಿ ಮತ್ತು ಹೊಲೊಲೆನ್ಸ್‌ಗಳಲ್ಲಿ ಒಂದೇ ಕೋಡನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇವು ಬೇರೆ ಬೇರೆ ಸ್ಕ್ರೀನ್‌ನ ಗಾತ್ರಕ್ಕೆ ಹೊಂದುವಂತಹ ಇಂಟರ್‌ಫೇಸನ್ನು ಹೊಂದಿರುತ್ತವೆ. ವಿಂಡೋಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗಾಗಿ ವರ್ಡ್ ಮತ್ತು ಎಕ್ಸೆಲ್‌ನಂತಹ ಆಫೀಸ್ ಮತ್ತು ಔಟಲುಕ್ ಮೇಲ್ ಮತ್ತು ಕ್ಯಾಲೆಂಡರ್ ಅಪ್ಲಿಕೇಷನ್ ಕೂಡ ಯುನಿವರ್ಸಲ್ ಅಪ್ಲಿಕೇಷನ್ ಆಗಿದೆ.

ಸೆಟ್ಟಿಂಗ್ಸ ಮತ್ತು ಕಂಟ್ರೋಲ್ ಪ್ಯಾನೆಲ್ (Settings and control panel)

ಕಂಟ್ರೋಲ್ ಪ್ಯಾನೆಲನಲ್ಲಿರುವ ಬಹಳಷ್ಟು ಸೆಟ್ಟಿಂಗಗಳನ್ನು ವಿಂಡೊಸ್ 8 ಹೊಂದಿದೆ. ಮತು್ತ ಇದು ಕಂಟ್ರೋಲ್ ಪ್ಯಾನೆಲನಂತೆಯೆ ಹುಡುಕಾಡಲು ಐಕಾನ್‌ಗಳೊಂದಿಗೆ ಇಂಟರ್‌ಫೇಸನ್ನು ಹೊಂದಿದೆ. ಆದರೆ ಹೊಸ ಸೆಟ್ಟಿಂಗ ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ ಇಲ್ಲದಿರುವ ಸೆಟ್ಟಿಂಗಗಳಿ ಗಾಗಿ , ಹಳೆಯ ಕಂಟ್ರೋಲ್ ಪ್ಯಾನೆಲ ಇಂಟರ್‌ಫೇಸ್ ಇನ್ನೂ ಇದೆ.

ವಿಂಡೋಸ್ 10 - ಕೀಬೋರ್ಡ ಶಾರ್ಟಕಟ್‌ಗಳು (Windows 10 Keyboard Shortcuts)

ಹೆಚ್ಚಿನ ವಿಂಡೊಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಂತೆ, ಸರಳವಾಗಿ ಮತ್ತು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಹಲವಾರು ಶಾರ್ಟಕಟ್ ಕೀಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಹಲವಾರು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಹೆಚ್ಚಿನ ಹೊಸ ವಿಂಡೊಸ್‌ಗಳು ವಿಂಡೊಸ್ ಕೀಯನ್ನು ಬೇರೆ ಕೀ ಜೊತೆಗೆ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತವೆ. ವಿಂಡೊಸ್ 10 ರಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಬಳಸುವ ಮತ್ತು ಉಪಯುಕ್ತ ಕೆಲವು ಶಾರ್ಟಕಟ್ ಕೀಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ.

ವಿಂಡೊಸ್ 10 ನ್ನು ನ್ಯಾವಿಗೆಟ್ ಮಾಡಲು ಬಳಸುವ ಕೀಬೋರ್ಡ ಶಾರ್ಟಕಟ್‌ಗಳು

ಈ ಕೀಗಳನ್ನು ಒತ್ತಿರಿ	ಈ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಿರಿ
Windows Logo	ಸ್ಟಾರ್ಟ ಮೆನುವನ್ನು ಟಾಗಲ್ ಮಾಡಲು
Windows Logo + A	ನೋಟಿಫಿಕೇಷನ ಪೇನನ್ನು ತೆರೆಯಲು
Windows Logo + B	ನೋಟಿಫಿಕೇಷನ್ ಏರಿಯಾದ ಹಿಡನ್ ಐಕಾನ್ ಬಾಣದ ಚಿನ್ಹೆಯನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಲು
Windows Logo C	ಧ್ವನಿ ಆಜ್ಞೆಗಳಿಗಾಗಿ ಕೋರ್ಟಾನಾ ತೆರೆಯಲು
Windows Logo D	ಡೆಸ್ಕಟಾಪ್ ಪ್ರದರ್ಶಿಸಲು ಎಲ್ಲ ತೆರೆದ ವಿಂಡೊಗಳನ್ನು ಮಿನಿಮೈಸ್ ಮಾಡಲು
Windows Logo E	ಫೈಲ್ ಎಕ್ಸಪ್ಲೋರರ್ ರನ್ ಮಾಡಲು
Windows Logo + F	ಸ್ಟಾರ್ಟ ಮೆನು ಪ್ರದರ್ಶಿಸಲು ಮತ್ತು ಸರ್ಚ ಬಾಕ್ಸ ಸಕ್ರಿಯಗೊಳಿಸಲು
Windows Logo + H	ಶೇರ್ ಪೇನ್ ಪ್ರದರ್ಶಿಸಲು
Windows Logo + I	ಸೆಟ್ಟಿಂಗ್ಸ ಅಪ್ಲಿಕೇಶನ್‌ನ್ನು ರನ್ ಮಾಡಲು
Windows Logo + K	ಡಿವೈಸಸ್ ಪೇನನ್ನು ಪ್ರದರ್ಶಿಸಲು
Windows Logo + L	ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಲಾಕ್ ಮಾಡಲು
Windows Logo + M	ಎಲ್ಲ ವಿಂಡೊಗಳನ್ನು ಮಿನಿಮೈಸ್ ಮಾಡಲು
Windows Logo + O	ಟ್ಯಾಬ್ಲೆಟ್ ಓರಿಯಂಟೇಶನ್ ಲಾಕ್‌ನ್ನು ಆನ್ ಮತ್ತು ಆಫ್ ಮಾಡಲು
Windows Logo + P	ಎರಡನೇ ಪ್ರದರ್ಶನವನ್ನು ಕಾನ್ಫಿಗರ್ ಮಾಡಲು, ಪ್ರಾಜಕ್ಟ ಪೇನನ್ನು ಪ್ರದರ್ಶಿಸಲು
Windows Logo + Q	ಧ್ವನಿ ಆಜ್ಞೆಗಳಿಗಾಗಿ ಕೋರ್ಟಾನಾ ತೆರೆಯಲು
Windows Logo + R	ರನ್ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ ತೆರೆಯಲು
Windows Logo + S	ಕೀ ಕಮಾಂಡಗಳಿಗಾಗಿ ಕೋರ್ಟಾನಾ ತೆರೆಯಲು
Windows Logo + T	ಟಾಸ್ಕಬಾರ್ ಐಕಾನ್‌ಗಳನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಲು(ಐಕಾನ್‌ಗಳನ್ನು ಹುಡುಕಲು ಬಾಣದ ಕೀಗಳನ್ನು ಬಳಸಿ)

© NIMI NOT TO BE REPUBLISHED

Windows Logo + U	ಈಸ ಆಫ್ ಎಕ್ಸೆಸ್ ಸೆಂಟರ್‌ನ್ನು ತೆರೆಯಲು
Windows Logo + W	ಸರ್ಚ ಬಾಕ್ಸನ್ನು ತೆರೆಯಲು
Windows Logo + X	ವಿಂಡೊಸ್ ಟೂಲ್ಸ್ ಮತ್ತು ಯುಟಿಲಿಟಿ ಮೆನು ತೆರೆಯಲು
Windows Logo + Z	ಅಪ್ಲಿಕೇಷನ್‌ನ ಆಜ್ಞೆಗಳನ್ನು ಪ್ರದರ್ಶಿಸಲು
Windows Logo +=	ಮ್ಯಾಗ್ನಿಫ್ಲೈಯರ್ ಮತ್ತು ಝೂಮ್ ಇನ್‌ನ್ನು ತೆರೆಯಲು
Windows Logo +-	ಝೂಮ್ ಔಟ ಮಾಡಲು
Windows Logo +,	ಡೆಸ್ಕಟಾಪನ್ನು ತಾತ್ಕಾಲಿಕವಾಗಿ ಪ್ರದರ್ಶಿಸಲು
Windows Logo +Enter	ನೆರೆಟರನ್ನು ತೆರೆಯಲು
Windows Logo +Left	ಪ್ರಸ್ತುತ ಅಪ್ಲಿಕೇಷನನ್ನು ಸ್ಕ್ರೀನ್ ನ ಎಡಭಾಗಕ್ಕೆ ಸ್ನ್ಯಾಪ್ ಮಾಡಲು
Windows Logo +Right	ಪ್ರಸ್ತುತ ಅಪ್ಲಿಕೇಷನನ್ನು ಸ್ಕ್ರೀನ್ ನ ಬಲಭಾಗಕ್ಕೆ ಸ್ನ್ಯಾಪ್ ಮಾಡಲು
Windows Logo +Up	ಮಿನಿಮೈಸ್ ಆದ ಅಪ್ಲಿಕೇಷನ್ನು ರೀಸ್ಟೋರ್ ಮಾಡಲು; ರೀಸ್ಟೋರ್ ಆದ ಅಪ್ಲಿಕೇಷನ್ನು ಮ್ಯಾಕ್ಸಿಮೈಸ್ ಮಾಡಲು
Windows Logo +Down	ಮ್ಯಾಕ್ಸಿಮೈಸ್ ಆದ ಅಪ್ಲಿಕೇಷನ್ನು ರೀಸ್ಟೋರ್ ಮಾಡಲು; ರೀಸ್ಟೋರ್ ಆದ ಅಪ್ಲಿಕೇಷನ್ನು ಮಿನಿಮೈಸ್ ಮಾಡಲು
Windows Logo +PgUp	ಪ್ರಸ್ತುತ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಎಡ ಮಾನಿಟರ್‌ಗೆ ಸರಿಸಲು
Windows Logo +PgDn	ಪ್ರಸ್ತುತ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಬಲ ಮಾನಿಟರ್‌ಗೆ ಸರಿಸಲು
Windows Logo +Prtsc	ಪ್ರಸ್ತುತ ಸ್ಕ್ರೀನನ್ನು ಸೆರೆಹಿಡಿದು ಅದನ್ನು ಪಿಕ್ಚರ್ ಫೋಲ್ಡರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸುವುದು
Windows Logo +Ctrl+D	ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪನ್ನು ತಯಾರಿಸುವುದು
Windows Logo +Ctrl+ Right	ಮುಂದಿನ ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗೆ ಬದಲಾಯಿಸುವುದು
Windows Logo +Ctrl+Left	ಹಿಂದಿನ ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪ್‌ಗೆ ಬದಲಾಯಿಸುವುದು
Windows Logo +Ctrl+F4	ಪ್ರಸ್ತುತ ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪನ್ನು ಮುಚ್ಚುವುದು
Windows Logo + Tab	ಟಾಸ್ಕ ವ್ಯೂ ತೆರೆಯುವುದು, ಇದು ಪ್ರತಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಅಪ್ಲಿಕೇಷನ್‌ಗೆ ಥಂಬನೇಲನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಜೊತೆಗೆ ಲಭ್ಯವಿರುವ ವರ್ಚುವಲ್ ಡೆಸ್ಕಟಾಪಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಿಮಾಸ್ ಸೆಟಪ್ ಮತ್ತು ವಿಂಡೋಸ್ ಸ್ಥಾಪನ (Install Windows operating system)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಬಯಾಸ್ ಮತ್ತು ಸೀಮಾಸ್‌ನ ಕಾರ್ಯಗಳನ್ನು ವಿವರಿಸಿ
- ಬಯಾಸ್‌ನ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ನೋಡುವುದು ಮತ್ತು ಬದಲಾಯಿಸುವ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ವಿಭಜನೆ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ ಅರ್ಥವನ್ನು ವಿವರಿಸಿ
- ವಿಂಡೊಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಸ್ಥಾಪಿಸುವ ಪ್ರಕ್ರಿಯೆಯನ್ನು ವಿವರಿಸಿ.

ಬೇಸಿಕ್ ಇನ್‌ಪುಟ್ /ಔಟಪುಟ್ ಸಿಸ್ಟಮ್(BIOS) ನ್ನು ಸಿಸ್ಟಮ್ ಬಯಾಸ್, ರಾಮ್ ಬಯಾಸ್ ಅಥವಾ ಪಿಸಿ ಬಯಾಸ್ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಇದು ಫರ್ಮವೇರ್ ಇಂಟರ್‌ಫೇಸನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಒಂದು ಸಾಮಾನ್ಯವಾದ ಪ್ರಮಾಣಿತವಾಗಿದೆ.

ಬಯಾಸ್‌ನ ಮೂಲ ಉದ್ದೇಶ ಸಿಸ್ಟಮ್ ಹಾರ್ಡವೇರ ಘಟಕಗಳನ್ನು ಪ್ರಾರಂಭಿಸುವುದು ಮತ್ತು ಪರೀಕ್ಷಿಸುವುದು ಆಗಿರುತ್ತದೆ. ಮತ್ತು ಮಾಸ್ ಮೆಮೊರಿ ಸಾಧನಗಳಿಂದ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಲೋಡ ಮಾಡುವುದು. ಬಯಾಸ್, ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಮುಖ್ಯವಾದ ಹಾರ್ಡವೇರ್ ಘಟಕಗಳನ್ನು ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನೊಂದಿಗೆ ಸಂಪರ್ಕಿಸುವ ವಿಶೇಷ ಸಾಫ್ಟವೇರ್ ಆಗಿದೆ. ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಮದರಬೋರ್ಡನಲ್ಲಿರುವ ಫ್ಲ್ಯಾಶ್ ಮೆಮೊರಿ ಚಿಪ್‌ನಲಿ ್ಲ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ. ಆದರೆ ಕೆಲವು ಸಲ ಈ ಚಿಪ್ ಬೇರೊಂದು ವಿಧದ ರೋಮ್(ROM) ಆಗಿರುತ್ತದೆ. ಬಯಾಸ್ ಒಂದು ಫರ್ಮವೇರ್ ಆಗಿದೆ. (ಸಾಫ್ಟವೇರ್ ಸೂಚನೆಗಳು ಶಾಶ್ವತವಾಗಿ ಮದರಬೋರ್ಡನಲ್ಲಿರುವ ಚಿಪ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗಿರುತ್ತದೆ.) (ಚಿತ್ರ 1 ರಂತೆ)

Fig 1

ಬಯಾಸ್‌ನ ಕಾರ್ಯಗಳು(Functions of BIOS)

ಬಯಾಸ್ ಸಾಫ್ಟವೇರ್ ಹಲವಾರು ವಿಭಿನ್ನ ಪಾತ್ರಗಳನ್ನು ಹೊಂದಿದೆ, ಆದರೆ ಇದರ ಪ್ರಮುಖ ಪಾತ್ರ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಲೋಡ್ ಮಾಡುವುದಾಗಿದೆ. ಪ್ರತೀ ಬಾರಿ ಕಂಪ್ಯೂಟರ್ ಪ್ರಾರಂಭ ಆದಾಗ ಅಥವಾ ಮರುಪ್ರಾರಂಭ (restart) ಆದಾಗ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ಗೆ ನಿಯಂತ್ರಣವನ್ನು ಹಸ್ತಾಂತರಿಸುವ ಮೊದಲು ಪಿಸಿ ಹಾರ್ಡವೇರನ್ನು ಪರಿಶೀಲಿಸಿ ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಬಯಾಸ್ ನಿರ್ವಹಿಸುವ ಕೆಲವು ಸಾಮಾನ್ಯ ಕಾರ್ಯಗಳು :

- ಎಲ್ಲವೂ ಘಟಕಗಳೂ ಸರಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿವೆ ಎನ್ನುವುದನ್ನು ಖಚಿತಪಡಿಸಲು ಎಲ್ಲ ಹಾರ್ಡವೇರ್ ಘಟಕಗಳಿಗೆ ಪಾವರ್ ಆನ್ ಸೆಲ್ಫ ಟೆಸ್ಟ POST (Power On Self Test) ನ್ನು ನಡೆಸುತ್ತದೆ.
- ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಇನ್ಸಾಟ್ಲ್ ಆದ ಎಲ್ಲ ವಿವಿಧ ಕಾರ್ಡಗಳಲ್ಲಿರುವ ಬೇರೆ ಬಯಾಸ್ ಚಿಪ್‌ಗಳನ್ನು ಸಕ್ರಿಯಗೊಳಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಗ್ರಾಫಿಕ್ ಕಾರ್ಡಗಳು ತಮ್ಮ ಸ್ವಂತ ಬಯಾಸ್ ಚಿಪ್‌ನ್ನು ಹೊಂದಿರುತ್ತವೆ.
- ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ವಿಭಿನ್ನ ಹಾರ್ಡವೇರ್ ಸಾಧನಗಳಿಗೆ ಇಂಟರ್‌ಫೇಸ್ ಮಾಡಲು ಬಳಸುವ ಲೋ ಲೆವೆಲ ರೂಟೀನ್‌ಗಳ ಗುಂಪನ್ನು ಒದಗಿಸುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್ ಬೂಟ್ ಆಗುವಾಗ ಕೀಬೋರ್ಡ , ಸ್ಕ್ರೀನ್, ಮತ್ತು ಪೋರ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.
- ಹಾರ್ಡಡಿಸ್ಕ ಮತ್ತು ಕ್ಲಾಕ್ ಮುಂತಾದವುಗಳ ಸೆಟ್ಟಿಂಗ್‌ಗಳ ಸಂಗ್ರಹವನ್ನು ನಿರ್ವಹಿಸುವುದು.

ಸೀಮಾಸ್ ಸೆಟ್‌ಅಪ್ (CMOS Setup)

ಬಯಾಸ್‌ನ ಮೊದಲ ಕೆಲಸವೆಂದರೆ ಮಾಹಿತಿಯನ್ನು ಪರಿಶೀಲಿಸುವುದು. ಈ ಮಾಹಿತಿಯು ಕಾಂಪ್ಲಿಮೆಂಟರಿ ಮೆಟಲ್ ಆಕ್ಸಾಯ್ಡ ಸೆಮಿಕಂಡಕ್ಟರ್ (CMOS) ಚಿಪ್‌ನಲ್ಲಿರುವ ಸಣ್ಣ ಪ್ರಮಾಣದ ರ್‍ಯಾಮ್ (RAM) ನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುತ್ತದೆ . ಸೀಮಾಸ್ ಸೆಟಪ್ ನಿಮ್ಮ ಸಿಸ್ಟಮ್‌ಗೆ ಸಂಬಂಧಪಟ್ಟ ವಿವರವಾದ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು ಇದನ್ನು ನಿಮ್ಮ ಸಿಸ್ಟಮ್‌ನ ಬದಲಾವಣೆಗೆ ತಕ್ಕಂತೆ ಬದಲಾಯಿಸಬಹುದು. ಬಯಾಸ ಈ ಮಾಹಿತಿಯನ್ನು ಅದರ ಡೀಫಾಲ್ಟ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಲು ಅಥವಾ ಪೂರಕವಾಗಿ ಅಗತ್ಯಕ್ಕೆ ತಕ್ಕಂತೆ ಉಪಯೋಗಿಸುತ್ತದೆ.

ಬಯಾಸನ್ನು ಸಂರಚಿಸುವುದು (Configuring BIOS)

ಕಸ್ಟಮ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳಿಗಾಗಿ ಬಾಯಾಸ್ ಸೀಮಾಸ್ ಸೆಟಪ್‌ನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಸೀಮಾಸ್ ಸೆಟ್ಟಿಂಗನ್ನು ಬದಲಾಯಿಸಲು ನಾವು ಸೀಮಾಸ್ ಸೆಟಪ್‌ಗೆ ಹೋಗಬೇಕಾಗುತ್ತದೆ. ಸೀಮಾಸ್ ಸೆಟಪ್‌ಗೆ ಹೋಗಲು ನಿರ್ದಿಷ್ಟ ಕೀಯನ್ನು ಅಥವಾ ಕೀ ಗಳ ಸಂಯೋಜನೆಯನ್ನು ಪ್ರಾರಂಭಿಕ ಸಮಯದಲ್ಲಿ ಒತ್ತಬೇಕು. ಹೆಚ್ಚಿನ ಸಿಸ್ಟಮ್‌ಗಳು Esc, Del, F1, F2, Ctrl-Esc ಅಥವಾ Ctrl-Alt-Esc ಕೀಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಕ್ರೀನ್‌ನ ಕೆಳಭಾಗದಲ್ಲಿ ``Press _____ to Enter Setup`` ಎಂದು ಪಠ್ಯದ ಸಾಲು ಇರುತ್ತದೆ.

ಬಯಾಸ್ ಸೆಟಪ್ ಬಹಳ ಆಯ್ಕೆಗಳಿರುವ ಪಠ್ಯದ ಪರದೆಗಳ ಗುಂಪನ್ನು ತೋರಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ ಕೆಲವು ಪ್ರಮಾಣಿತವಾದರೆ, ಉಳಿದವು ಬಯಾಸ್ ತಯಾರಕರ ಪ್ರಕಾರ ಬದಲಾಗುತ್ತವೆ. ಸಾಮಾನ್ಯ ಆಯ್ಕೆಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

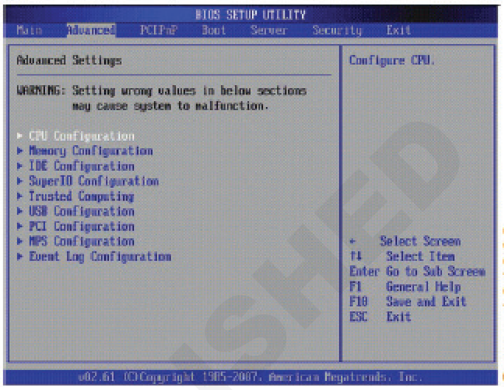
- ಸಿಸ್ಟಮ್ ಸಮಯ /ದಿನಾಂಕ-ಸಿಸ್ಟಮ್ ಸಮಯ ಮತ್ತು ದಿನಾಂಕವನ್ನು ಹೊಂದಿಸುವುದು.
- ಬೂಟ್ ಸಿಕ್ವೆನ್ಸ್ - ಬಯಾಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಲೋಡ್ ಮಾಡಲು ಪ್ರಯತ್ನಿಸುವ ಶ್ರೇಣಿ
- ಪ್ಲಗ್ ಮತ್ತು ಪ್ಲೇ - ಜೋಡಿಸಿದ ಸಾಧನಗಳನ್ನು ಸ್ವಯಂ ಪತ್ತೆಹಚ್ಚುವ ಒಂದು ಮಾನದಂಡ , ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಎರಡೂ ಇದನ್ನು ಬೆಂಬಲಿಸುತ್ತಿದ್ದರೆ, ಇದನ್ನು yes ಗೆ ಹೊಂದಿಸಬೇಕು.
- ಮೌಸ್ / ಕೀಬೋರ್ಡ - Enable Num Lock, Enable the Key-board, Auto - Detect Mouse ..
- ಡ್ರೈವ್ ಕಾನ್ಫಿಗರೇಷನ್ - ಹಾರ್ಡ ಡ್ರೈವ್‌ಗಳನ್ನು , ಸಿಡಿ ರಾಮ್‌ಗಳನ್ನು ಮತ್ತು ಫ್ಲಾಪಿ ಡ್ರೈವ್‌ಗಳನ್ನು ಸಂರಚಿಸಿ
- ಮೆಮೊರಿ - ನಿರ್ದಿಷ್ಟವಾದ ಮೆಮೊರಿ ಅಡ್ರೆಸ್‌ಗೆ ಬಯಾಸ್‌ನ್ನು ನಿರ್ದೇಶಿಸಿ
- ಭದ್ರತೆ - ಕಂಪ್ಯೂಟರನ್ನು ಉಪಯೊಗಿಸಲು ಪಾಸ್‌ವರ್ಡನ್ನು ಇಡಿ
- ವಿದ್ಯುತ್ ನಿರ್ವಹಣೆ - ವಿದ್ಯುತ್ ನಿರ್ವಹಣೆಯನ್ನು ಉಪಯೋಗಿಸಬೇಕೆ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿ. ಜೊತೆಗೆ ಸ್ಟ್ಯಾಂಡಬೈ ಮತ್ತು ಸಸ್ಪೆಂಡಗೆ ಸಮಯವನ್ನು ನಿಗದಿಪಡಿಸಿ.
- ಎಕ್ಸಿಟ್ - ನೀವು ಮಾಡಿದ ಬದಲಾವಣೆಗಳನ್ನು ಸೇವ್ ಮಾಡಿ, ನೀವು ಮಾಡಿದ ಬದಲಾವಣೆಗಳನ್ನು ಡಿಸ್ಕಾರ್ಡ ಮಾಡಿ ಅಥವಾ ಡೀಫಾಲ್ಟ ಸೆಟ್ಟಿಂಗ್ ಗಳನ್ನು ರೀಸ್ಟೋರ್ ಮಾಡಿ.

ಕಂಪ್ಯೂಟರ್ ಸೆಟ್ಟಿಂಗ್‌ನಲ್ಲಿ ಮಾಡಿದ ಬದಲಾವಣೆಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಬಯಾಸ್, ಸಿಮಾಸ್ ತಂತ್ರಜ್ಞಾನವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಈ ತಂತ್ರಜ್ಞಾನದಲ್ಲಿ ಒಂದು ಸಣ್ಣ ಲಿಥಿಯಂ ಅಥವಾ Ni Cad ಬ್ಯಾಟರಿ ಮಾಹಿತಿಯನ್ನು ವರ್ಷಗಳವರೆಗೆ ಸಂಗ್ರಹಿಸಿಡಲು ಸಾಕಷ್ಟು ವಿದ್ಯುತ್ತನ್ನು ಒದಗಿಸುತ್ತದೆ. ಬಯಾಸ್‌ನ ಮುಖ್ಯ ತಯಾರಕರೆಂದರೆ ಅಮೇರಿಕನ್ ಮೆಗಾಟ್ರೆಂಡ್ಸ್ ಇ (AMI), ಫೋನಿಕ್ಸ್ ಟೆಕ್ನಾಲಾಜಿ , ವಿನ್ ಬಾಂಡ್ ಇತ್ಯಾದಿಗಳು. ವಿಶಿಷ್ಟವಾದ ಬಯಾಸ್ ಸ್ಕ್ರೀನ್ ಶಾಟನ್ನು ಈ ಕೆಳಗಿನ ಚಿತ್ರ 2 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

ವಿಂಡೊಸ್ ಆಪರೆಟಿಂಗ್ ಸಿಸ್ಟಮನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು

(Installing Windows Operating System)

Fig 2



Copa 181302

ಹಾರ್ಡ ಡಿಸ್ಕನಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಮೊದಲು ಅದನ್ನು ಪಾರ್ಟಿಷನ್ ಮಾಡುವ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವ ಅವಶ್ಯಕತೆಯಿರುತ್ತದೆ.

ಪಾರ್ಟಿಷನಿಂಗ್(Partitioning)

ಪಾರ್ಟಿಷನ್ ಕೆಲವೊಮ್ಮೆ ವಾಲ್ಯೂಮ್ ಎಂದೂ ಕರೆಯುತ್ತೇವೆ. ಇದು ಹಾರ್ಡ ಡಿಸ್ಕನಲ್ಲಿರುವ ಒಂದು ಪ್ರದೇಶವಾಗಿದ್ದು, ಫೈಲ್ ಸಿಸ್ಟಮ್ ಪ್ರಕಾರ ಇದನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬೇಕಾಗಿರುತ್ತದೆ, ಮತ್ತು ವರ್ಣಮಾಲೆಯ (alphabet) ಅಕ್ಷರದಂತೆ ಅದನ್ನು ಗುರುತಿಸಬೇಕಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಹೆಚ್ಚಿನ ವಿಂಡೋಸ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಡ್ರೈವ್ C ಪಾರ್ಟಿಷನ್ ಆಗಿರುತ್ತದೆ. ನೀವು ರಚಿಸಿದ ಮೊದಲನೆ ಮೂರು ಪಾರ್ಟಿಷನ್‌ಗಳು ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್‌ಗಳಾಗಿರುತ್ತವೆ. ಇವು ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತವೆ. ಒಂದು ವೇಳೆ ನೀವು ಮೂರಕ್ಕಿಂತ ಹೆಚ್ಚು ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಬಯಸಿದರೆ, ನಾಲ್ಕನೇ ಪಾರ್ಟಿಷನ್ ಎಕ್ಸಟೆಂಡೆಡ್ ಪಾರ್ಟಿಷನ್ ಆಗಿ ರಚಿಸಲ್ಪಡುತ್ತದೆ.

ಎಕ್ಸಟೆಂಡೆಡ್ ಪಾರ್ಟಿಷನ್ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಲಾಜಿಕಲ್ ಡ್ರೈವ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಲಾಜಿಕಲ್ ಡ್ರೈವ್ ಪ್ರೈಮರಿ ಪಾರ್ಟಿಷನ್ ನಂತೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಆದರೆ ಇದನ್ನು ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲು ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ.

ಸಾಕಷ್ಟು ಕಂಪ್ಯೂಟರ್‌ಗಳು ಒಂದೇ ಪಾರ್ಟಿಷನ್‌ನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇವು ಹಾರ್ಡ ಡಿಸ್ಕನ ಗಾತ್ರಕ್ಕೆ ಸಮನಾಗಿರುತ್ತವೆ. ಹಾರ್ಡ ಡಿಸ್ಕನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಬಹಳಷ್ಟು ಸಣ್ಣ ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನಾಗಿ ವಿಭಜಿಸುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ ಆದರೆ ಇದು ಹಾರ್ಡಡಿಸ್ಕನಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಹಂಚಿಕೆ ಮಾಡಲು ಸಹಾಯವಾಗುತ್ತದೆ.

ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ಮಾಡುವುದರಿಂದ ಆಗುವ ಅನುಕೂಲಗಳು:

- ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಮತ್ತು ಪ್ರೋಗ್ರಾಂ ಫೈಲ್‌ಗಳನ್ನು ಬಳಕೆದಾರರ ಫೈಲ್‌ಗಳಿಂದ ಬೇರ್ಪಡಿಸುವುದು
- ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ವರ್ಚುವಲ್ ಮೆಮೊರಿ ಸ್ವ್ಯಾಪಿಂಗ / ಪೇಜಿಂಗ್‌ಗಾಗಿ ಪ್ರತ್ಯೇಕ ಸ್ಥಳ ಹೊಂದುವುದಕ್ಕಾಗಿ
- ಮೇಲಿಂದ ಮೇಲೆ ಉಪಯೋಗಿಸುವ ಪ್ರೋಗ್ರಾಂಗಳು ಮತ್ತು ದತ್ತಾಂಶಗಳನ್ನು ಒಂದಕ್ಕೊಂದು ಹತ್ತಿರವಿರುವಂತೆ ನೋಡಿಕೊಳ್ಳಲು
- ಒಂದೇ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಬಳಕೆದಾರರು ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ ಒಬ್ಬ ವ್ಯಕ್ತಿಯು ತನ್ನ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಲಿ ೀನಕ್ಸ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಮತ್ತು ಮೈಕ್ರೊಸಾಫ್ಟ ವಿಂಡೊಸ್ ಅಥವಾ ಇತರೆ ಯಾವುದೇ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಒಂದೇ ಹಾರ್ಡಡಿಸ್ಕ್‌ನ ಬೇರೆ ಬೇರೆ ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು ಮತ್ತು ಯಾವುದೇ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನಿಂದ ಬೂಟ್ ಮಾಡುವ ಆಯ್ಕೆಯನ್ನು ಹೊಂದಿರುತ್ತಾನೆ.
- ದೋಷಪೂರಿತ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಮರಳಿ ಪಡೆಯಲು ಸುಲಭವಾಗುವಂತೆ ಅವುಗಳನ್ನು ಸಂರಕ್ಷಿಸುತ್ತದೆ. ಇದರಿಂದ ಒಂದು ಪಾರ್ಟಿಷನ್ ದೊಷಪೂರಿತವಾಗಿದ್ದರೆ , ಇತರ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ.
- ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಒಟ್ಟಾರೆ ಕಂಪ್ಯೂಟರ್ ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು ಹೆಚ್ಚಿಸುವುದು. ಇಲ್ಲಿ ಸಣ್ಣ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ಗಳು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿರುತ್ತವೆ.
- ಪಾರ್ಟೀಷನ್ ಮಾಡುವುದರಿಂದ ಚೆಕ್‌ಡಿಸ್ಕನಂತಹ ವಿಶ್ಲೇಷಣಾತ್ಮಕ ಸಾಧನಗಳು ಕೆಲಸ ಮಾಡಲು ಕಡಿಮೆ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳುತ್ತವೆ.

ಫಾರ್ಮ್ಯಾಟಿಂಗ(Formatting)

ಆರಂಭಿಕ ಬಳಕೆಗಾಗಿ ಮಾಹಿತಿ ಸಂಗ್ರಹಣಾ ಸಾಧನಗಳಾದ ಹಾರ್ಡಡಿಸ್ಕ ಡ್ರೈವ್, ಸಾಲಿಡ್ ಸ್ಟೇಟ್ ಡ್ರೈವ್, ಅಥವಾ ಯುಎಸ್‌ಬಿ ಡ್ರೈವ್‌ಗಳನ್ನು ಸಿದ್ಧಪಡಿಸುವ ಪ್ರಕ್ರಿಯೆಗೆ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ವಾಲ್ಯೂಮ್‌ನಲ್ಲಿ ಫೈಲ್ ಸಿಸ್ಟಮ್ ತಯಾರಿಸುವ ಒಂದು ಕ್ರಿಯೆಯಾಗಿದೆ. ಇದರಿಂದ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಆ ವಾಲ್ಯೂಮ್‌ನಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವುದು ಮತ್ತು ಪಡೆಯುವುದನ್ನು ಮಾಡಬಹುದು.

ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟಿಂಗ ಎರಡು ರೀತಿಯಿಂದ ಮಾಡಬಹುದು :(Formatting of a disk is of two catagories) :

1 ಲೋ ಲೆವೆಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ(ಎಂದರೆ ಹಾರ್ಡವೇರ್‌ಗೆ ಹತ್ತಿರದಲ್ಲಿದೆ) ಇದು ಡಿಸ್ಕನ ಮೇಲ್ಮೈ ಮೇಲೆ ಮಾರ್ಕರ್‌ನೊಂದಿಗೆ ಗುರುತು ಮಾಡುತ್ತದೆ. ಇದು ರೆಕಾರ್ಡಿಂಗ ಬ್ಲಾಕ್‌ನ ಪ್ರಾರಂಭವನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ಬ್ಲಾಕ್ ಚೆಕ್ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಡಿಸ್ಕ ಕಂಟ್ರೋಲರ್ ನಿಂದ ಭವಿಷ್ಯದ ಉಪಯೋಗಕ್ಕಾಗಿ ಮಾಹಿತಿಯನ್ನು ಓದಲು ಮತ್ತು ಬರೆಯಲು ಮಾಹಿತಿ ನೀಡುತ್ತದೆ. ಡಿಸ್ಕನ ಶಾಶ್ವತ ಅಡಿಪಾಯದ ಉದ್ದೇಶಕ್ಕಾಗಿ ಇದನ್ನು ಮಾಡಲಾಗುತ್ತದೆ. ಹಾಗೂ ಕಾರ್ಖಾನೆಯಲ್ಲಿ ಇದನ್ನು ಪೂರ್ಣಗೊಳಿಸಲಾಗುತ್ತದೆ. ನೀವು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಮೊದಲು ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಪಾರ್ಟಿಷನ್ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

2 ಹೈ ಲೆವೆಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಇದು ಡಿಸ್ಕ ಪಾರ್ಟಿಷನ್‌ನಲ್ಲಿ ಅಥವಾ ಲಾಜಿಕಲ್ ವಾಲ್ಯೂಮ್‌ನಲ್ಲಿ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ರಚಿಸುತ್ತದೆ. ಈ ರೀತಿಯ ಫಾರ್ಮ್ಯಾಟಿಂಗ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನಿಂದ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಡಾಟಾ ಸ್ಟ್ರಕ್ಚರನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್, ಡಾಟಾ ಸ್ಟ್ರಕ್ಚರನ್ನು ಲಾಜಿಕಲ್ ಡ್ರೈವ್ ಅಥವಾ ಪಾರ್ಟಿಷನ್ ವಿಷಯಗಳನ್ನು ಗುರುತಿಸಲು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವಾಗ ಅಥವಾ ಹೊಸ ಡಿಸ್ಕನ್ನು ಸೇರಿಸುವಾಗ ಕಂಡು ಬರುತ್ತದೆ.

ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು (Installing Windows Operating System)

ವಿಂಡೋಸ್‌ನ್ನು ಮೂರು ಮೂಲ ವಿಧಾನಗಳಿಂದ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು. ಅವು ಈ ಕೆಳಗಿನಂತಿವೆ:

- ಹೊಚ್ಚ ಹೊಸ ಡಿಸ್ಕನಲ್ಲಿ ಅಥವಾ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು
- ಡಿಸ್ಕನ್ನು ಅಳಿಸಿ, ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿ, ಮತ್ತು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು
- ಡ್ಯುಯಲ್ ಬೂಟಿಂಗ್‌ಗಾಗಿ ಹೊಸ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು

ಮೊದಲನೆ ಎರಡು ವಿಧಾನಗಳಲ್ಲಿ, ಕಂಪ್ಯೂಟರ್ ಡಿವಿಡಿ ಯಿಂದ ಅಥವಾ ಯಾವುದೇ ತೆರೆಯಬಹುದಾದ ಡ್ರೈವ್‌ನಿಂದ ಬೂಟ್ ಆಗುತ್ತದೆ ಎನ್ನುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ. ಇದನ್ನು ಮಾಡುವುದಕ್ಕಾಗಿ ಬಯಾಸ್‌ನಲ್ಲಿರುವ ಡ್ರೈವ್ ಬೂಟ್ ಆರ್ಡರ್‌ನ್ನು ಬದಲಾಯಿಸಬೇಕಾಗುತ್ತದೆ. ಇತ್ತೀಚೆಗಿನ ವಿಂಡೋಸ್ ಡಿವಿಡಿಗಳು ಬೂಟ್ ಮಾಡಬಲ್ಲವಾಗಿರುತ್ತವೆ. ಮತ್ತು ಸೆಟಪ್ ಪ್ರೋಗ್ರಾಂನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ರನ್ ಮಾಡಬಲ್ಲವು. ಪ್ರಾಯೋಗಿಕ ಪಾಠದಲ್ಲಿರುವಂತೆ ಹಂತ ಹಂತವಾಗಿ ಕಾರ್ಯವಿಧಾನವನ್ನು ಅನುಸರಿಸುವ ಮೂಲಕ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

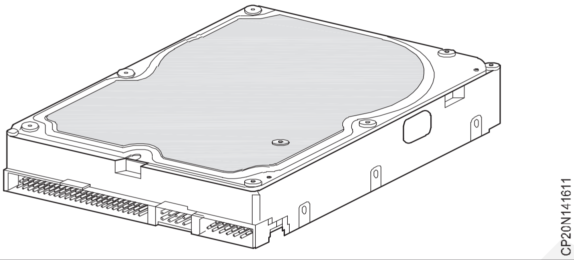
ಹಾರ್ಡ್‌ಡಿಸ್ಕ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು ಮತ್ತು ಪಾರ್ಟಿಷನ್ ರಚಿಸುವುದು (Format hard disk and create partition)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಬಯಾಸ್ ಮತ್ತು ಸೀಮಾಸ್‌ನ ಕಾರ್ಯಗಳನ್ನು ವಿವರಿಸಿ
- ಬಯಾಸ್‌ನ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ನೋಡುವುದು ಮತ್ತು ಬದಲಾಯಿಸುವ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ವಿಭಜನೆ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ ಅರ್ಥವನ್ನು ವಿವರಿಸಿ
- ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಸ್ಥಾಪಿಸುವ ಪ್ರಕ್ರಿಯೆಯನ್ನು ವಿವರಿಸಿ.

ಹಾರ್ಡ ಡ್ರೈವ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Format a hard Drive)

Fig 1



ಹಾರ್ಡ ಡ್ರೈವ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದಕ್ಕೆ ಹಲವಾರು ಕಾರಣಗಳಿರಬಹುದು. ವಿಂಡೋಸ್‌ನ್ನು ಹೊಸದಾಗಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲು, ವೈರಸ್ ಅಥವಾ ಮಾಲ್ವೇರನ್ನು ತೆಗೆದು ಹಾಕಲು ಅಥವಾ ಪಿಸಿಯನ್ನು ರೆಫ್ರೆಶ್ ಮಾಡಲು ಅಥವಾ ಪಿಸಿಯನ್ನು ಸ್ವಚ್ಚಗೊಳಿಸಲು ಹೀಗೆ ಹಲವಾರು ಕಾರಣಗಳಿರಬಹುದು.

ಇದು ಬರಿ ಹಾರ್ಡಡ್ರೈವ್ ಆಗಿದೆಯೊ ಅಥವಾ ಇನ್ನೊಂದು ಪಿಸಿ ಇದೆಯೊ ಇಲ್ಲವೊ ಎನ್ನುವುದರ ಮೇಲೆ ಈ ಪ್ರಕ್ರಿಯೆ ಬೇರೆ ಬೇರೆಯಾಗಿರಬಹುದು.

ವಿಂಡೋಸ್ ರನ್ ಆಗುತ್ತಿರುವ ಹಾರ್ಡಡ್ರೈವ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಬರುವುದಿಲ್ಲ. ಈ ರೀತಿ ಮಾಡಲು ಪಿಸಿಯನ್ನು ವಿಂಡೊಸ್ ಇನ್ಸ್ಟಾಲೇಷನ್ ಡಿಸ್ಕನಿಂದ , ಯುಎಸ್ ಬಿ ಫ್ಲ್ಯಾಷ್ ಡ್ರೈವ್ ನಿಂದ ಅಥವಾ ಬೇರೆ ಬೂಟ್ ಮಾಡಬಲ್ಲ ಡಿಸ್ಕನಿಂದ ಬೂಟ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು ? (Format a Disk ?)

ಹಾರ್ಡ ಡ್ರೈವ್‌ನಿಂದ ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ಅಳಿಸುವ ಪ್ರಕ್ರಿಯೆಗೆ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಆದರೆ ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದರಿಂದ ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ಉಳಿಸುತ್ತದೆ ಮತ್ತು ಡ್ರೈವ್ ಖಾಲಿಯಾದಂತೆ ತೋರುತ್ತದೆ. ಹೊಚ್ಚ ಹೊಸ ಡ್ರೈವ್ ಇದ್ದರೆ ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದು. ಅಥವಾ ವಿಂಡೊಸ್‌ನ್ನು ರೀಇನ್ಸ್ಟಾಲ್ ಮಾಡುವ ಅವಶ್ಯಕತೆಯಿದ್ದಾಗ ಅಥವಾ ಪಿಸಿಯನ್ನು ಬೇರೆಯವರಿಗೆ ನೀಡುವಾಗ ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದು.

ಎಚ್ಚರಿಕೆ : ಡಿಸ್ಕನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದಕ್ಕೂ ಮೊದಲು ಅದರಲ್ಲಿರುವ ಫೋಟೋಗಳನ್ನು, ವೀಡಿಯೋಗಳನ್ನು , ಸಂಗೀತವನ್ನು ಮತ್ತು ಇತರ ದಾಖಲೆಗಳನ್ನು ಡ್ರೈವ್‌ನಿಂದ ಯಶಸ್ವಿಯಾಗಿ ಬ್ಯಾಕಪ್ ಮಾಡಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ. ಆದರೂ ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ಅಳಿಸಿ ಹಾಕಿದ ಫೈಲ್ ಗಳನ್ನು ವಾಪಸ್ ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಹಾರ್ಡ ಡ್ರೈವ್ ಪಾರ್ಟಿಷನ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು. (Format hard drive partition)

ಹಾರ್ಡಡ್ರೈವನ್ನು ಪಾರ್ಟಿಷನ್ ಮಾಡುವ ಮೊದಲು ಪಾರ್ಟಿಷನ್‌ಗಳ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳುವುದು ಮುಖ್ಯವಾಗಿದೆ. ಹಾರ್ಡಡ್ರೈವನ್ನು ಸಣ್ಣ ಸಣ್ಣ ವಿಭಾಗಗಳನ್ನಾಗಿ ವಿಭಜಿಸಿರುತ್ತಾರೆ. ಈ ವಿಭಾಗಗಳನ್ನು ಪಾರ್ಟಿಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಬೇರೆ ವಿಭಾಗಗಳನ್ನು ಮುಟ್ಟದೆ ಒಂದೇ ಪಾರ್ಟಿಷನನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದು.

ಇಡೀ ಹಾರ್ಡಡ್ರೈವನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬೇಕಾದರೆ ಮತ್ತು ಸಂಪೂರ್ಣ ಬ್ಲಾಕ್‌ನಲ್ಲಿ ಸಂಪೂರ್ಣ ಸಾಮರ್ಥ್ಯವನ್ನು ಬಳಸಬೇಕಾದರೆ ಪಾರ್ಟಿಷನ್ ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಅಳಿಸಬೇಕಾಗುತ್ತದೆ.

ಬಯಾಸ್‌ನಿಂದ ಹಾರ್ಡಡ್ರೈವನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದೆ ? (Format a hard drive from the BIOS ?)

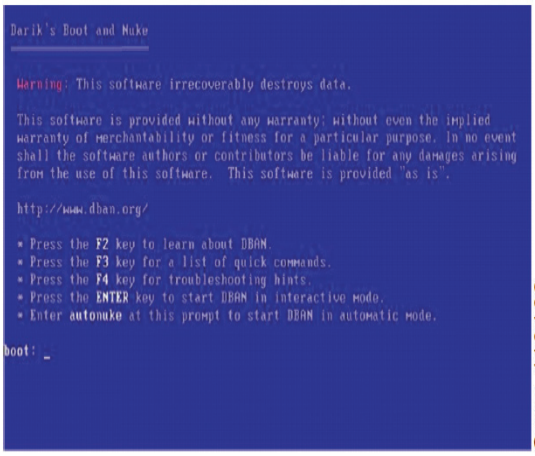
ಬಹಳ ಜನರು ಬಯಾಸ್ ನಿಂದ ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಹೇಗೆ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದು ಎಂದು ಕೇಳುತ್ತಾರೆ. ಅದರ ಉತ್ತರ ಇಲ್ಲ.

ಡಿಸ್ಕನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವ ಅವಶ್ಯಕತೆಯಿದ್ದರೆ, ನೀವು ಅದನ್ನು ವಿಂಡೊಸ್‌ನಲ್ಲಿದ್ದು ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ. ಬೂಟ್ ಮಾಡಬಹುದಾದ ಸಿಡಿ, ಡಿವಿಡಿ,ಅಥವಾ ಯುಎಸ್ ಬಿ ಫ್ಲ್ಯಾಷ ಡ್ರೈವನ್ನು ತಯಾರಿಸಿ ಮತ್ತು ಉಚಿತ ಥರ್ಡ ಪಾರ್ಟಿ ಫಾರ್ಮ್ಯಾಟಿಂಗ ಸಾಧನವನ್ನು ರನ್ ಮಾಡಿ.

© NIMI NOT TO BE REPUBLISHED

ಒಂದು ಆಯ್ಕೆಯೆಂದರೆ ಡಾರಿಕ್ಸ್ ಬೂಟ್ ಮತ್ತು ನ್ಯೂಕ್ (DBAN) ಇದು ವೈಯಕ್ತಿಕ ಬಳಕೆಗೆ ಉಚಿತವಾಗಿದೆ. ಈ ಪ್ರೋಗ್ರಾಂ ನಿಮ್ಮ ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಅಳಿಸುತ್ತದೆ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುತ್ತದೆ. ಇದರಿಂದ ಹೊಸ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು. ಆದರೆ ಈ ಪ್ರಕ್ರಿಯೆಯನ್ನು ರದ್ದುಗೊಳಿಸಲಾಗುವುದಿಲ್ಲ.

Fig 2



Copa 110102

DBAN ಬೂಟ್ ಮಾಡಬಹುದಾದ ಸಿಡಿ, ಡಿವಿಡಿ-ಆರ್‌ಗಳನ್ನು ಮಾತ್ರ ತಯಾರಿಸುತ್ತದೆ. ಆದರೆ ನಿಮ್ಮ ಹತ್ತಿರ ಖಾಲಿ ಡಿಸ್ಕಗಳು ಅಥವಾ ಬರ್ನರ್ ಇಲ್ಲದಿದ್ದರೆ, ಪ್ರತ್ಯೇಕ ಥರ್ಡ ಪಾರ್ಟಿ ಪ್ರೋಗ್ರಾಂ ಪರಿಹಾರ ರೂಪದಲ್ಲಿ ಲಭ್ಯವಿರುತ್ತದೆ.

ಯುನಿವರ್ಸಲ್ ಯುಎಸ್ ಬಿ ಇನ್ಸ್ಟಾಲರ್ ಇದು ವೇಗವಾಗಿ ಮತ್ತು ಸುಲಭವಾಗಿ ಡೌನ್ಲೋಡ ಮಾಡಲಾದ DBAN ISO ಚಿತ್ರವನ್ನು ಬೂಟ ಮಾಡಬಹುದಾದ ಯುಎಸ್‌ಬಿ ಗೆ ಬದಲಾಯಿಸಿ ರನ್ ಮಾಡುತ್ತದೆ.

ಸರಳವಾಗಿ ಒಂದು ಖಾಲಿ ಯುಎಸ್‌ಬಿ ಫ್ಲ್ಯಾಷ್ ಡ್ರೈವನ್ನು ಅಳವಡಿಸಿ , ಯುನಿವರ್ಸಲ್ ಯುಎಸ್‌ಬಿ ಇನ್ಸ್ಟಾಲರ್ ಸೆಟಪ್ ಪ್ರೋಗ್ರಾಂನ್ನು ರನ್ ಮಾಡಿ ಮತ್ತು ಪ್ರಾಂಪ್ಟನ್ನು ಅನುಸರಿಸಿ.

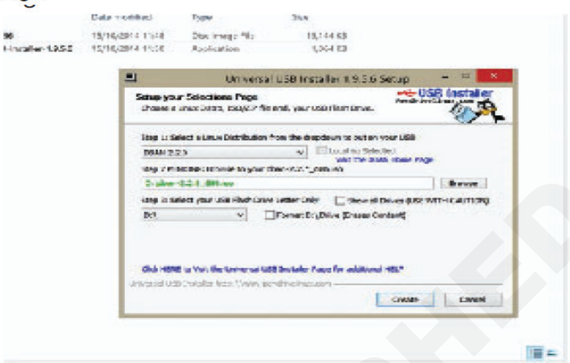
ಇದು ಸ್ಕ್ರಾಲ್ ಮಾಡಲು ಮತ್ತು ಯುಎಸ್‌ಬಿ ಗೆ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬೇಕಾದ ಲಿನಕ್ಸ್ ವಿತರಣೆಯನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ಕೇಳುತ್ತದೆ.(ಈ ಸಂದರ್ಭದಲ್ಲಿ DBAN ನ ಇತ್ತೀಚಿನ ಆವೃತ್ತಿ) . ಇಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಅದರ ಸ್ಥಳ ಮತ್ತು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬೇಕಾದ ಯುಎಸ್‌ಬಿ ಡ್ರೈವ್‌ನ ಅಕ್ಷರವನ್ನು ಕೇಳುತ್ತದೆ.

ಈ ಮಾಹಿತಿಯು ಪೂರ್ಣಗೊಂಡ ನಂತರ , ಈ ಕೆಳಗಿನಂತೆ ಕ್ಲಿಕ್ ಮಾಡಿ

ಯುಎಸ್‌ಬಿ ಡ್ರೈವ್‌ನಿಂದ ಬೂಟ್ ಮಾಡುವುದಕ್ಕಾಗಿ , ಸಾಮಾನ್ಯ ಬೂಟ್ ಸಾಧನಕ್ಕಿಂತ (ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ ಇದು ಹಾರ್ಡ ಡ್ರೈವ್ ಆಗಿರುತ್ತದೆ.) ಬಯಾಸನಲ್ಲಿ ಕೆಲವು ಸೆಟ್ಟಿಂಗ್ಸ್‌ನ್ನು ಬದಲಾಯಿಸಬೇಕಾಗುತ್ತದೆ.

ಬಯಾಸ್‌ನಲ್ಲಿ ಬೂಟ್ ಆರ್ಡರ್ ಸೆಟ್ಟಿಂಗ್ಸ್‌ನ್ನು ನ್ಯಾವಿಗೆಟ್ ಮಾಡಿ. ಮತ್ತು ಪ್ರೈಮರಿ ಬೂಟ್ ಸಾಧನವನ್ನು ಯುಎಸ್‌ಬಿ ಡ್ರೈವ್‌ಗೆ ಬದಲಾಯಿಸಿ (ಈ ಆಯ್ಕೆ ಮಾಡಲು ಪ್ಲಗ್ ಇನ್ ಮಾಡಬೇಕಾದ ಅವಶ್ಯಕತೆಯಿಲ್ಲ). ಸೆಟ್ಟಿಂಗ್ಸ್‌ನ್ನು ಸೇವ್ ಮಾಡಿದ ನಂತರ ಮತ್ತು ಬಯಾಸ್‌ನಿಂದ ನಿರ್ಗಮಿಸಿದ ನಂತರ ಬೂಟ್ ಮಾಡಬಹುದಾದ ಯುಎಸ್‌ಬಿಯನ್ನು ಸೇರಿಸಿ, ಕಂಪ್ಯೂಟರನ್ನು ಮರುಪ್ರಾರಂಭಿಸಿ.

Fig 3



Copa 191503

Fig 4

Copa 191504

ಪಿಸಿ ಸ್ವಯಂಚಾಲಿತವಾಗಿ DBAN ಸಾಫ್ಟವೇರನ್ನು ಬೂಟ್ ಮಾಡಬೇಕು. ಇದು ಹಾರ್ಡಡ್ರೈವ್‌ನ್ನು ಅಳಿಸುವ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ಮಾರ್ಗದರ್ಶನ ನೀಡುತ್ತದೆ. ಡಾಟಾ ವೈಪಿಂಗ್‌ನ ವಿವಿಧ ಹಂತಗಳಲ್ಲಿ ಆಯ್ಕೆಗಳ ಮೂಲಕ ಮಾರ್ಗದರ್ಶನ ನೀಡುತ್ತದೆ.

ಇದು ಯುಎಸ್ ಬಿ ಡ್ರೈವನ್ನು ಮತ್ತೊಂದು ಡ್ರೈವ್‌ನಂತೆ ಪರಿಗಣಿಸುತ್ತದೆ. ಅಜಾಗರೂಕತೆಯಿಂದ ಅದು ಹಾಳಾಗುವುದನ್ನು ತಪ್ಪಿಸಲು ಅದನ್ನು DBAN ನಲ್ಲಿ ಬೂಟ್ ಆದ ನಂತರ ತೆಗೆದುಹಾಕಿ.

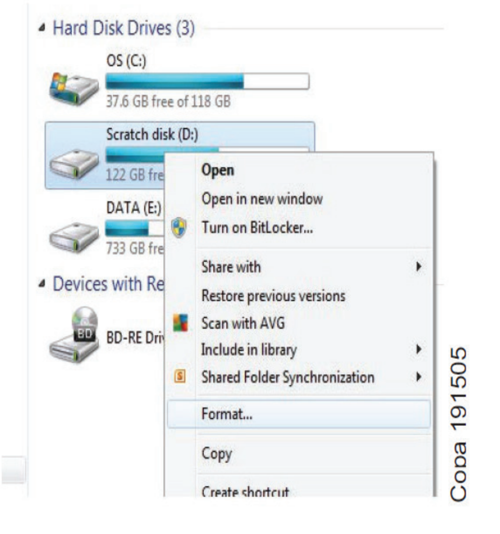
ಹಾರ್ಡ ಡ್ರೈವ್‌ನ್ನು ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದೆ ? (Quick format a hard drive ?)

ಹೌದು. ಆದರೆ ಮಾಹಿತಿಯನ್ನು ಶಾಶ್ವತವಾಗಿ ಅಳಿಸಲು ಬಯಸಿದರೆ ಈ ವಿಧಾನವನ್ನು ಉಪಯೋಗಿಸಬೇಡಿರಿ. ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಮಾಡಿದರೆ ಮಾಹಿತಿಯು ಅಳಿಸುವುದಿಲ್ಲ. ಆದರೆ ಫೈಲ್ ಪಾಯಿಂಟರಗಳನ್ನು ಮಾತ್ರ ಅಳಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

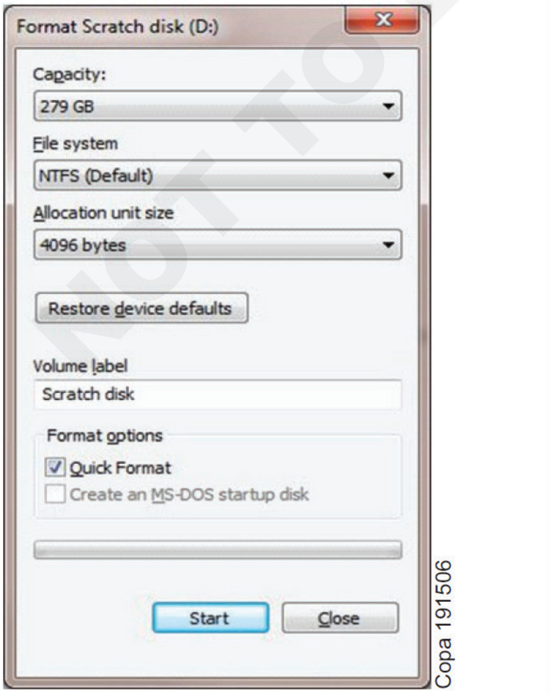
ವಿಂಡೋಸ್ ವಿಸ್ತಾ , 7, 8 ಮತ್ತು 10 ಇವುಗಳು ಅಂತರ್ನಿರ್ಮಿತ ಡಿಸ್ಕ ನಿರ್ವಹಣಾ ಸಾಧನವನ್ನು ಹೊಂದಿವೆ. (ಕೆಳಗೆ ನೋಡಿ). ಆದರೆ ಹಾರ್ಡ ಡ್ರೈವನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವ ಅತ್ಯಂತ ವೇಗವಾದ ಮಾರ್ಗವೆಂದರೆ, ಸ್ಟಾರ್ಟ್ ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡುವುದು, ನಂತರ ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಅಳಿಸಲು ಹಾರ್ಡಡ್ರೈವ್ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿ. ಸ್ಪಷ್ಟವಾದ ಕಾರಣಗಳಿಗಾಗಿ ವಿಂಡೊಸ್ ಇನ್ಸ್ಟಾಲ್ ಆದ ಡ್ರೈವನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.

Fig 5



ಪೂರ್ವನಿಯೋಜಿತವಾಗಿ ಕ್ವಿಕ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಪರಿಶೀಲಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಫೈಲ್ ಸಿಸ್ಟಮ್, ಹಂಚಿಕೆ ಘಟಕದ ಗಾತ್ರ ಮತ್ತು ವಾಲ್ಯೂಮ್ ಲೇಬಲ್(ಡ್ರೈವ್ ನ ಹೆಸರು) ಬದಲಾಯಿಸುವುದು ಇವುಗಳನ್ನು ಆರಿಸುವುದು. ವಿಶಿಷ್ಟವಾಗಿ ಎಲ್ಲ ಸೆಟ್ಟಿಂಗಗಳನ್ನು ಬದಲಾಯಿಸಬೇಡಿ ಮತ್ತು ಸ್ಟಾರ್ಟ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿ. ಒಂದು ನಿಮಿಷದಲ್ಲಿ ಹಾರ್ಡ ಡ್ರೈವ್ ಫಾರ್ಮ್ಯಾಟ ಆಗುತ್ತದೆ.

Fig 6



ವಿಂಡೊಸ್ ವಿಸ್ತಾ, 7,8 ಅಥವಾ 10 ಗಳಿಗೆ ಎನ್‌ಟಿಎಫ್‌ಎಸ್ (NTFS) ನ್ನು ಫೈಲ್ ಸಿಸ್ಟಮ್ ಆಗಿ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿ ಮತ್ತು ಅಲೋಕೇಷನ್ ಯೂನಿಟ್ ಸೈಜನ್ನು ಡೀಫಾಲ್ಟ ಅಲೋಕೇಷನ್ ಸೈಜ್‌ಗೆ ಹೊಂದಿಸಿ.

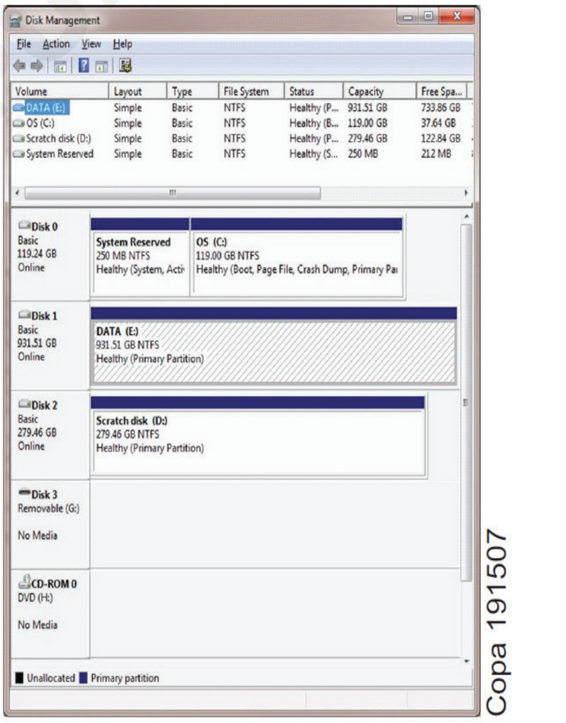
ಡಿಸ್ಕ ನಿರ್ವಹಣಾ ಸಾಧನವನ್ನು ಉಪಯೋಗಿಸುವುದು (Using the Disk Management tool)

ವಿಂಡೊಸ್ ವಿಸ್ತಾ , 7, 8 ಅಥವಾ 10 ನ ಸರ್ಚ ಬಾಕ್ಸ್‌ನಲ್ಲಿ diskmg-mt.msc ಅಥವಾ ಡಿಸ್ಕ ಮ್ಯಾನೇಜಮೆಂಟ ಎಂದು ಟೈಪ ಮಾಡಿ, ಮತ್ತು ಮೇಲಿನ ಮೇನುವಿನಲ್ಲಿ ಅದೇ ಹೆಸರಿನಿಂದ ಬಂದ ಫಲಿತಾಂಶದ ಮೇಲೆ ಮಾತ್ರ ಕ್ಲಿಕ್ ಮಾಡಿ.

ಡಿಸ್ಕ ನಿರ್ವಹಣೆಯನ್ನು ಪ್ರಾರಂಭಿಸಲು ಇದು ಸುಲಭವಾದ ಮಾರ್ಗವಾಗಿದೆ. ಆದರೆ ಇದು ಕಂಟ್ರೋಲ್ ಪ್ಯಾನಲ್‌ನಲ್ಲಿಯೂ ಇರುತ್ತದೆ. ಇದರಲ್ಲಿ ಡಿಸ್ಕ ಎಂದು ಹುಡುಕಬೇಕು ಮತ್ತು Create and format hard dsik partitions ನ್ನು ಆಯ್ಕೆ ಮಾಡಬೇಕು.

ಡಿಸ್ಕ ಮ್ಯಾನೇಜಮೆಂಟ್ ಇದು ಸ್ಟ್ಯಾಂಡ ಅಲೋನ ಮ್ಯಾನೇಜಮೆಂಟ ಟೂಲ್‌ನಷ್ಟು ಶಕ್ತಿಯುತವಾಗಿಲ್ಲ. ಆದರೆ ಇದು ಡೇಟಾವನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಇನ್ನೂ ಸಮರ್ಥವಾಗಿದೆ.

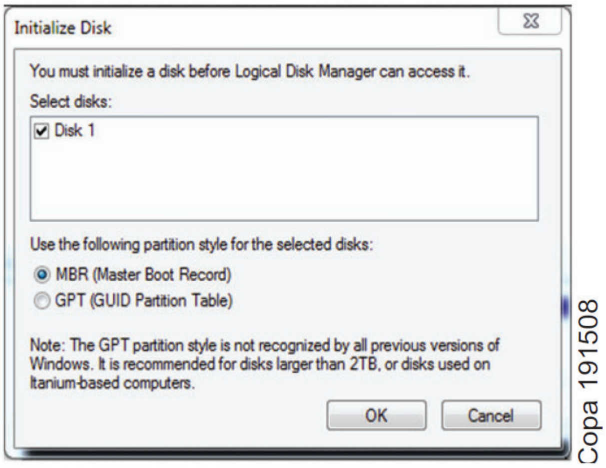
Fig 7



ಪಿಸಿಯಲ್ಲಿ ಹೊಸ (ಹೆಚ್ಚುವರಿ) ಹಾರ್ಡ ಡ್ರೈವನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬೇಕಾದರೆ, ಅದು ವಿಂಡೊಸ್ ಎಕ್ಸಪ್ಲೋರರ್‌ನಲ್ಲಿ ಏಕೆ ಕಾಣಿಸುವುದಿಲ್ಲ ಎಂದು ಆಶ್ಚರ್ಯಪಡುವ ವಿಷಯವಾಗಿರಬಹುದು. ಇದಕ್ಕೆ ಕಾರಣವೇನೆಂದರೆ ಅದನ್ನು ಪ್ರಾರಂಭಿಸಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಮತ್ತು ಇದನ್ನು ಡಿಸ್ಕ ಮ್ಯಾನೆಜಮೆಂಟ್‌ನಲ್ಲಿ ಮಾಡಬಹುದು.

ಈ ಸಾಧನವು ಲೋಡ್ ಆದಾಗ, ಇದು ಕಂಪ್ಯೂಟರ್‌ನ ಎಲ್ಲ ಡ್ರೈವ್‌ಗಳನ್ನು ವಿಶ್ಲೇಷಿಸುತ್ತದೆ. ಮತ್ತು ಅದು ಕಂಡುಕೊಳ್ಳುವ ಹೊಸ ಡಿಸ್ಕನ್ನು ಪ್ರಾರಂಭಿಸಲು ಕೇಳುತ್ತದೆ.

Fig 8



ಡಿಸ್ಕ 2 ಟಿಬಿ ಗಿಂತ ದೊಡ್ಡದಾಗಿದ್ದರೆ, ಜಿಪಿಟಿ (GUID ಪಾರ್ಟಿಷನ್ ಟೇಬಲ್)ಯನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ. ಈ ಸೆಟ್ಟಿಂಗ ನಾಲ್ಕಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಪಾರ್ಟಿಷನ್ ಗಳನ್ನು ರಚಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಪ್ರಾಂಪ್ಟ ಕಾಣದಿದ್ದರೆ , ಡ್ರೈವ್‌ಗಳ ಪಟ್ಟಿಯಲ್ಲಿ ನೋಡಿರಿ ಮತ್ತು Not Initialized ಎಂಬುದನ್ನು ನೋಡಿರಿ . ಅದರ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು Initialize Disk ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಅದು ಮುಗಿದ ನಂತರ , ಉಳಿದ ಸ್ಥಳಾವಕಾಶದ ಮೇಲೆ ಬಲ ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು New Simple Volume ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

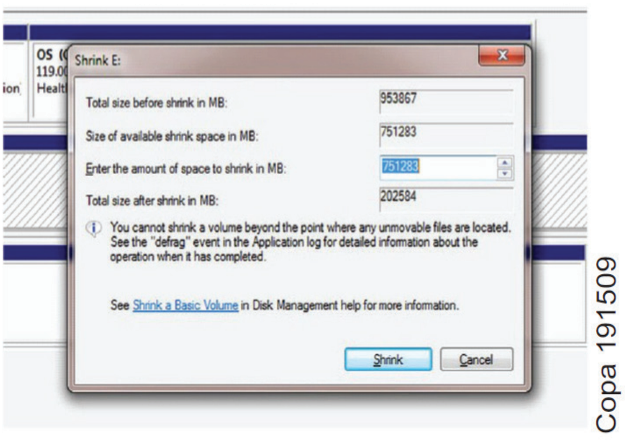
ಸೂಚನೆಗಳನ್ನು ಅನುಸರಿಸಿ, ಪಾರ್ಟಿಷನ್ ಎಷ್ಟು ದೊಡ್ಡದಾಗಿರಬೇಕು ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿ. (MB ಗಳಲ್ಲಿ 1024 MB = 1GB) ಮತ್ತು ಯಾವ ಡ್ರೈವ್ ಅಕ್ಷರವನ್ನು ನಿಯೋಜಿಸಬೇಕು. (ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಲಾಗುತ್ತದೆ, ಆದರೆ ಬದಲಾಯಿಸಬಹುದು)

ಒಟ್ಟು ಡ್ರೈವ್‌ನ ಸಾಮರ್ಥ್ಯಕ್ಕಿಂತ ಚಿಕ್ಕದಾದ ಗಾತ್ರವನ್ನು ಪಾರ್ಟಿಷನ್ ಮಾಡಲು ಆಯ್ಕೆ ಮಾಡಿದರೆ ಉದಾಹರಣೆಗೆ 1 TB ಡ್ರೈವ್ ನಲ್ಲಿ 500 GB ಆಯ್ಕೆ ಮಾಡಬಹುದು. ಉಳಿದ ಸ್ಥಳವನ್ನು ಇದೇ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಪುನರಾವರ್ತಿಸುವ ಮೂಲಕ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿ ಪೂರ್ಣಗೊಳಿಸಿ.

ಪಾರ್ಟಿಷನ್‌ನ ಗಾತ್ರವನ್ನು ಬದಲಾಯಿಸಿ. (Change Partition size)

ಪಾರ್ಟಿಷನ್‌ನ ಗಾತ್ರವನ್ನು ವಿಸ್ತರಿಸಲು ಅಥವಾ ಕುಗ್ಗಿಸಲು ಡಿಸ್ಕ ಮ್ಯಾನೇಜಮೆಂಟನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಪಾರ್ಟಿಷನ್ ಮೇಲೆ ಮೌಸ್ ನ ಬಲ ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು ಮೆನುವಿನಿಂದ ಸೂಕ್ತ ಆಯ್ಕೆ ಮಾಡಿ. ಒಂದು ವೇಳೆ ಕುಗ್ಗುತ್ತಿದ್ದರೆ, ಪಾರ್ಟಿಷನ್ ಎಷ್ಟು ಖಾಲಿ ಸ್ಥಳವನ್ನು ಹೊಂದಿದೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸಲಾಗುತ್ತದೆ.

Fig 9



ಇದು GB ಗಳ ಬದಲಾಗಿ MB ಗಳಲ್ಲಿ ಸಂಖ್ಯೆಗಳನ್ನು ತೋರಿಸುವುದರಿಂದ ಸ್ವಲ್ಪ ಗೊಂದಲಮಯವಾಗಿ ತೋರುತ್ತದೆ. ಆದರೆ, ಕುಗ್ಗಲು ಸ್ಥಳದ ಪ್ರಮಾಣವನ್ನು ಸರಿಹೊಂದಿಸಬಹುದು. ಮತ್ತು ಕುಗ್ಗಿದ ನಂತರದ ಒಟ್ಟು ಸ್ಥಳವನ್ನು ನವೀಕರಿಸಲಾಗುತ್ತದೆ. ಫೈಲ್ ಗಳು ಇರುವಲ್ಲಿ ಒಂದು ಹಂತವನ್ನು ಮೀರಿ ಪಾರ್ಟಿಷನ್‌ನ್ನು ಕುಗ್ಗಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಮೊದಲು ಡ್ರೈವನ್ನು ಡೀಫ್ರಾ ್ಯಗ್ಮೆಂಟ (De-fragment) ಮಾಡುವ ಮೂಲಕ ಸ್ಥಳವನ್ನು ಖಾಲಿ ಮಾಡಬಹುದು.

ಇದಕ್ಕೆ ವ್ಯತಿರಿಕ್ತವಾಗಿ, ಡ್ರೈವನಲ್ಲಿ ಬಳಕೆಯಾಗದ ಸ್ಥಳ ಇದ್ದರೆ ಮಾತ್ರ ಪಾರ್ಟಿಷನ್‌ನ್ನು ವಿಸ್ತರಿಸಬಹುದು. ಇಲ್ಲದಿದ್ದರೆ, ಆಯ್ಕೆಯನ್ನು ಗ್ರೆಔಟ್ ಮಾಡಲಾಗುತ್ತದೆ.

ವಿಂಡೋಸ್‌ನ್ನು ರಿಸೆಟ್ ಮಾಡುವುದು (Resetting Windows)

ಒಂದು ವೇಳೆ ನಿಮ್ಮ ಪಿಸಿಯಲ್ಲಿ ಸಮಸ್ಯೆ ಇದ್ದರೆ ಮತ್ತು ಹಾರ್ಡ ಡ್ರೈವನ್ನು ಅಳಿಸುವುದರಿಂದ ಬಳಕೆದಾರರು ವೈಯಕ್ತಿಕ ಫೈಲ್ ಗಳನ್ನು ಕಳೆದುಕೊಳ್ಳಲು ಬಯಸದಿದ್ದರೆ, ಅದಕ್ಕಾಗಿ ವಿಂಡೋಸ್ 8 ಮತ್ತು 10 ನಲ್ಲಿ ಪಿಸಿ ಯನ್ನು ರೆಫ್ರೆಶ್ ಅಥವಾ ರಿಸೆಟ್ ಮಾಡಬಹುದು.

ಬಳಕೆದಾರರು ವೈಯಕ್ತಿಕ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಸೆಟ್ಟಿಂಗಗಳನ್ನು ಉಳಿಸಿಕೊಳ್ಳಲು ಬಯಸಿದರೆ ಆದರೆ ಹೊಸ ವಿಂಡೋಸ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲು ಬಯಸಿದರೆ ಅದಕ್ಕಾಗಿ ಪಿಸಿಯನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

ರಿಫ್ರೆಶ ಮಾಡುವುದರಿಂದ ನಿಮ್ಮ ಯಂತ್ರದಲ್ಲಿರುವ ಎಲ್ಲ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ಆದರೆ, ವಿಂಡೋಸ್ ಡೀಫಾಲ್ಟ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಹಾಗೆಯೆ ಉಳಿಸುತ್ತದೆ.

ರಿಸೆಟ್ ಮಾಡುವುದರಿಂದ ವಿಂಡೊಸ್ ರೀಇನ್ಸ್ಟಾಲ್ ಆಗುತ್ತದೆ. ಇದರಿಂದ ಎಲ್ಲ ಫೈಲ್‌ಗಳು, ಸೆಟ್ಟಿಂಗ್‌ಗಳು ಮತ್ತು ಪ್ರೋಗ್ರಾಂಗಳು ಅಳಿಸುತ್ತವೆ. ಇದನ್ನು ಮಾಡುವುದಕ್ಕೂ ಮೊದಲು ಎಲ್ಲ ಫೈಲ್ ಗಳನ್ನು ಬ್ಯಾಕಪ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ವೈಯಕ್ತಿಕ ಫೈಲ್‌ಗಳನ್ನು ವರ್ಗಾಯಿಸಬೇಕಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ವಿಂಡೋಸ್ 10 ಸ್ವಲ್ಪ ವಿಭಿನ್ನವಾದ ವಿಧಾನವನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ವಿಂಡೋಸ್ 8 ನ್ನು ಉಪಯೋಗಿಸಿದವರಿಗೆ ಸ್ವಲ್ಪ ಗೊಂದಲಮಯವಾಗುತ್ತದೆ. ಮೈಕ್ರೊಸಾಫ್ಟ ರಿಫ್ರೆಶ್ ಆಯ್ಕೆಯನ್ನು ತೆಗೆದುಹಾಕಲಾಗಿದೆ ಮತ್ತು ಬದಲಿಗೆ ಒಂದೇ ಸೆಟ್ಟಿಂಗ್‌ನಲ್ಲಿ ರಿಫ್ರೆಶ್ ಮತ್ತು ರಿಸೆಟ್ ಆಯ್ಕೆಗಳನ್ನು ಸಂಯೋಜಿಸಲಾಗಿದೆ.

ಈ ಆಯ್ಕೆಯನ್ನು ಕಂಡುಹಿಡಿಯಲು, ಸ್ಟಾರ್ಟ ಮೆನು ತೆರೆಯಿರಿ, ಸೆಟ್ಟಿಂಗ್ಸ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ > ಅಪ್ಡೇಟ್ & ಸೆಕ್ಯುರಿಟಿ > ರಿಕವರಿ > ಗೆಟ್ ಸ್ಟಾರ್ಟೆಡ್ (ರಿಸೆಟ್ ದಿಸ್ ಪಿಸಿ ಆಯ್ಕೆಯಡಿಯಲ್ಲಿ)

(Start Menu , click on settings > Update & Security > Recovery > Get Started (under the Reset this PC option)

ಈ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿದ ನಂತರ , ಇದು ಮೂರು ಪ್ರತ್ಯೇಕ ಆಯ್ಕೆಗಳ ಜೊತೆಗೆ ಇರುತ್ತದೆ:

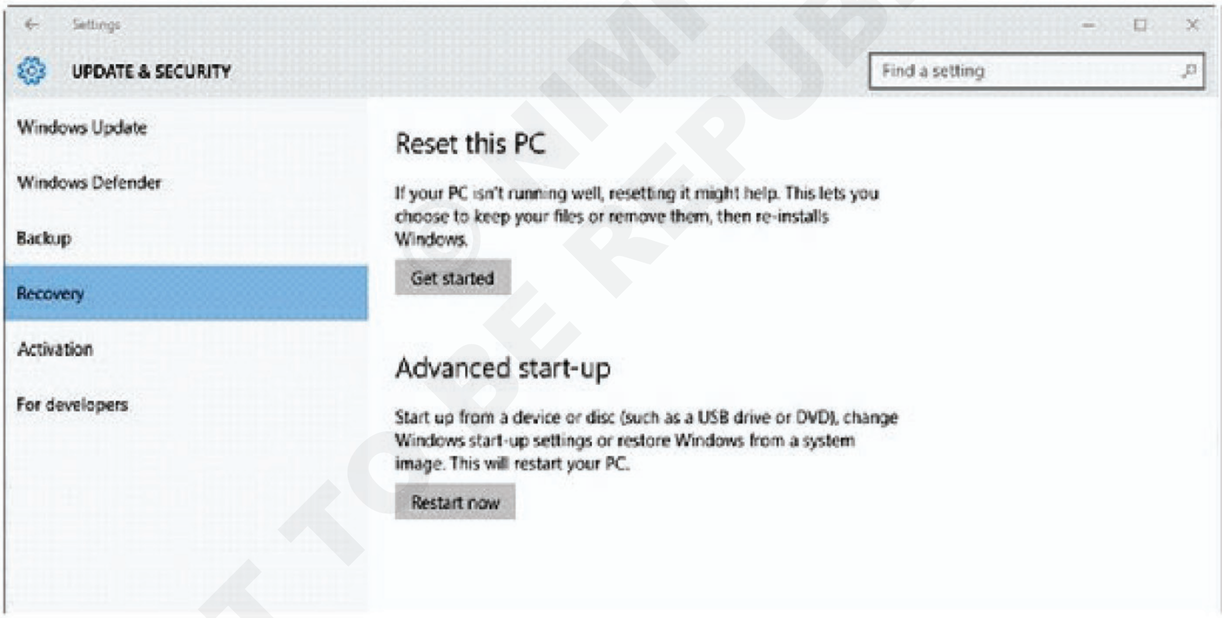
- ನನ್ನ ಫೈಲ್‌ಗಳನ್ನು ಇರಿಸಿ (Keep my files)
- ಎಲ್ಲವನ್ನೂ ತೆಗೆದುಹಾಕಿ (Remove everything)
- ಫ್ಯಾಕ್ಟರಿ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ರೀಸ್ಟೋರ್ ಮಾಡಿ (ಎಲ್ಲ ಪಿಸಿಗಳಲ್ಲಿ ಲಭ್ಯವಿಲ್ಲ)(Restore factory settings)

Keep my files (ನನ್ನ ಫೈಲ್‌ಗಳನ್ನು ಇರಿಸಿ)ಈ ಆಯ್ಕೆಯು ವೈಯಕ್ತಿಕ ಫೈಲ್ ಗಳನ್ನು ಉಳಿಸುತ್ತದೆ. ಆದರೆ ಬಳಕೆದಾರರ ಸೆಟ್ಟಿಂಗಗಳನ್ನು, ಡ್ರೈವರ್‌ಗಳನ್ನು, ಮತ್ತು ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಅಳಿಸುತ್ತದೆ. ವಿಂಡೊಸ್ 10 ನ್ನು ರೀಇನ್ಸ್ಟಾಲ್ ಮಾಡುತ್ತದೆ.

Remove everything (ಎಲ್ಲವನ್ನೂ ತೆಗೆದುಹಾಕಿ) ಈ ಆಯ್ಕೆಯೂ ವೈಯಕ್ತಿಕ ಫೈಲ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ಮತ್ತು Restore factory settings (ಫ್ಯಾಕ್ಟರಿ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ರೀಸ್ಟೋರ್ ಮಾಡಿ) ಇದು ಎಲ್ಲವನ್ನೂ ತೆಗೆದುಹಾಕಿ ಆಯ್ಕೆಯಂತೆ ಕೆಲಸಗಳನ್ನು ಮಾಡುತ್ತದೆ. ಆದರೆ ಪಿಸಿಯನ್ನು ಯಂತ್ರದ ಆವೃತ್ತಿಗೆ ರಿಸೆಟ್ ಸಹ ಮಾಡುತ್ತದೆ.

ಈಗಾಗಲೇ ನಿರ್ಮಿಸಲಾದ ಮತ್ತು ಈಗಾಗಲೇ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ವಿಂಡೋಸ್ ಹೊಂದಿರುವ ಪಿಸಿ ಅಥವಾ ಲ್ಯಾಪಟಾಪ್‌ನ್ನು ಖರೀದಿಸಿದಾಗ ಮಾತ್ರ ಈ ಆಯ್ಕೆಯು ಲಭ್ಯವಿರುತ್ತದೆ.

Fig 10



Copa 1915010

NOT TO BE REPUBLISHED © NIMI

ಓಎಸ್ ಸ್ಥಾಪನೆಯ ಸಂದರ್ಭದಲ್ಲಿ ಬರುವ ಸಾಮಾನ್ಯ ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಸಮಸ್ಯೆಗಳನ್ನು ಗುರುತಿಸುವುದು ಮತ್ತು ನಿವಾರಿಸುವುದು (Identify and rectify common hardware and software issues during OS installation)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಪಿಸಿ ಟ್ರಬಲ್ ಶೂಟಿಂಗ್ (ಸಮಸ್ಯೆ ನಿವಾರಣೆ) ಒಳಗೊಂಡಿರುವ ಮೂಲ ಹಂತಗಳನ್ನು ತಿಳಿಸಿ
- ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಇರುವ ಮೂಲ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ದೋಷಯುಕ್ತ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಸಂಭವನೀಯ ದೋಷಗಳು ಮತ್ತು ಲಕ್ಷಣಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ
- ದೂರುಗಳ ಕಾರಣಗಳನ್ನು ವಿಶ್ಲೇಷಣೆ ಮಾಡಿ
- ದೋಷಗಳನ್ನು ಸರಿಪಡಿಸಲು ಸುಲಭವಾದ ಮಾರ್ಗ ತಿಳಿಸಿ
- ದೋಷಯುಕ್ತ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಸಂಭವನೀಯ ದೋಷಗಳು ಮತ್ತು ಲಕ್ಷಣಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ
- ವಿಂಡೋಸ್ ಪ್ರಾರಂಭಿಸಿದಾಗ , ಸಿಸ್ಟಮ್ ಸರ್ವೀಸ್ ಟೆಸ್ಟ್ ರನ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಸೇಫ್ ಮೋಡ್‌ಗೆ ಹೋಗುತ್ತದೆ, ಈ ದೂರಿನ ಕಾರಣವನ್ನು ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ ಮತ್ತು ಟಿಎಸ್‌ಸಿನ ಸಹಾಯದಿಂದ ವಿಶ್ಲೇಷಿಸಿ.

ಮೂಲ ಸಮಸ್ಯೆ ನಿವಾರಣೆ (Basic Troubleshooting)

ಪಿಸಿಗಳಲ್ಲಿನ ಸಮಸ್ಯೆ ನಿವಾರಣೆ ಮಾಡುವಾಗಿನ ತೊಂದರೆ ಎಂದರೆ ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ ಸಮಸ್ಯೆ ಕಂಡುಬಂದಂತೆ ಅವು ಇರುವುದಿಲ್ಲ. ಒಂದು ಪಿಸಿ ಆಗಾಗ್ಗೆ ಹ್ಯಾಂಗ್ ಆಗುವುದಕ್ಕೆ ಕಾರಣ ಆರು ಅಥವಾ ಅದಕ್ಕಿಂತ ಹೆಚ್ಚು ಗುರುತಿಸಲಾಗದ ಸಮಸ್ಯೆಗಳಿರಬಹುದು. ಈ ಸಮಸ್ಯೆಗಳು ಹಾರ್ಡ್‌ವೇರ್ ಅಥವಾ ಸಾಫ್ಟ್‌ವೇರ್‌ನಿಂದ ಆಗಿರಬಹುದು. ಹಲವಾರು ವರ್ಷಗಳ ಅನುಭವ ಮತ್ತು ತರಬೇತಿ ಪಡೆದಿದ್ದರೂ ಸಹ ಪಿಸಿ ತಂತ್ರಜ್ಞರು ನಿಜವಾದ ಸಮಸ್ಯೆ ಪರಿಹರಿಸುವಂತಹ ಕಾರ್ಯವಿಧಾನಗಳು ಅವರಲ್ಲಿ ಇರುವುದಿಲ್ಲ. ಉದಾಹರಣೆಗೆ ಹಲವಾರು ಇನ್‌ಪುಟ್ /ಔಟಪುಟ್ ಕಾರ್ಡಗಳನ್ನು ಜೋಡಿಸಿರುವ ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಹೊಂದಿರುವ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ನ ಸ್ಕ್ರೀನ್, ಮೌಸ, ಕೀಬೋರ್ಡ್ ತಟಸ್ಥ(freeze)ವಾಗಬಹುದು. ಮತ್ತು ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಲು 3 ನಿಮಿಷದವರೆಗೆ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳಬಹುದು. ಮೆಮೊರಿ ಮಾಡ್ಯೂಲನ್ನು ಬದಲಾಯಿಸುವುದು, ಹೊಸ ಪ್ಯಾರಲಲ್ ಪೋರ್ಟನ್ನು ಮತ್ತು ಎನ್ ಐ ಸಿ ಡ್ರೈವನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು ಮುಂತಾದ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರಯತ್ನಿಸಿದ ನಂತರ ತಂತ್ರಜ್ಞರು ಕೊನೆಗೆ ಪ್ರಮುಖ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ ಲೈಬ್ರರಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿಲ್ಲ ಎಂದು ಕಂಡುಹಿಡಿಯುತ್ತಾರೆ ಮತ್ತು ಅದನ್ನು ಮತ್ತೊಮ್ಮೆ ರೀಇನ್ಸ್ಟಾಲ್ (reinstall) ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಇದನ್ನು ಉಲ್ಲೇಖಿಸುವ ಉದ್ದೇಶವೆಂದರೆ ತಂತ್ರಜ್ಞರು ದೋಷನಿವಾರಣೆಗೆ ವ್ಯವಸ್ಥಿತ ವಿಧಾನವನ್ನು ಬಳಸಿದ್ದರೆ ಇಂಥ ಸಮಸ್ಯೆಗಳನ್ನು ಶೀಘ್ರದಲ್ಲಿ ಪರಿಹರಿಸಬಹುದಾಗಿತ್ತು. ಉತ್ತಮ ದೋಷನಿವಾರಣೆಯ ಯೋಜನೆಯನ್ನು ನಿರ್ವಹಿಸುವುದು ಖಂಡಿತವಾಗಿಯೂ ಸಮಸ್ಯೆಯನ್ನು ವ್ಯವಸ್ಥಿತವಾಗಿ ಮತ್ತು ವೈಜ್ಞಾನಿಕವಾಗಿ ಪರಿಹರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ದೋಷನಿವಾರಣೆ (Troubleshooting) ಎಂದರೆ ಯಾವುದೇ ಸಮಸ್ಯೆ ನಿವಾರಣೆಗೆ ನಾವು ಉಪಯೋಗಿಸುವ ಲಿಖಿತ ಪರಿಶೀಲನಾ ಪಟ್ಟಿ. ಯಾವುದೇ ದೋಷನಿವಾರಣಾ ಯೋಜನೆಯಲ್ಲಿ ಇರುವ ಅಂಶಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

- ದಾಖಲೆ ನಿರ್ವಹಣೆ
- ಸಂಭವನೀಯ ಕಾರಣಗಳ ಗುರುತಿಸುವಿಕೆ
- ಸಂಭವನೀಯ ಪರಿಹಾರಗಳ ಗುರುತಿಸುವಿಕೆ
- ಪರಿಹಾರಗಳನ್ನು ಬಳಸುವುದು ಮತ್ತು ಪರೀಕ್ಷಿಸುವುದು
- ಅನುಸರಿಸುವುದು

ದಾಖಲೆ ನಿರ್ವಹಣೆ ಮಾಡುವಾಗ , ಪಿಸಿಯಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಹಾರ್ಡವೇರ, ಎಲ್ಲಾ ಸಮಸ್ಯೆ ತಡೆಗಟ್ಟುವ ನಿರ್ವಹಣಾ ಚಟುವಟಿಕೆಗಳು, ಎಲ್ಲ ಸಾಫ್ಟ್‌ವೇರ್ ನವೀಕರಣಗಳು ಅಥವಾ ಸೇರ್ಪಡೆಗಳು, ಮತ್ತು ಎಲ್ಲ ಹಾರ್ಡವೇರ್ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದ ಬಗ್ಗೆ ಮತ್ತು ನವೀಕರಣಗಳ ಬಗ್ಗೆ ದಾಖಲು ಮಾಡಿ. ಮುಂದುವರೆದು ಯಾವುದೇ ಸಮಸ್ಯೆ ಕಂಡು ಬಂದರೆ ಅವುಗಳನ್ನು ನಿವಾರಿಸಲು ನೀವು ತೆಗೆದುಕೊಂಡ ಕ್ರಮಗಳನ್ನು ದಾಖಲಿಸಬೇಕು. ಒಂದು ಪಿಸಿ ಯ ದೋಷನಿವಾರಣೆ ಮಾಡಬೇಕಾದಾಗ , ಈ ದಾಖಲಾತಿ ನಿರ್ವಹಣೆಯನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಸಮಸ್ಯೆಯನ್ನು ಕಂಡುಹಿಡಿದು ಅದಕ್ಕೆ ಪರಿಹಾರವನ್ನು ಸೂಚಿಸಬಹುದು. ಈ ದಾಖಲೆಯನ್ನು ಮೊದಲನೆಯದಾಗಿ ಪಿಸಿಯ ಸಂಕ್ಷಿಪ್ತ ವಿವರಣೆ ಇರಬೇಕು, ಇದು ಪಿಸಿಯ ಸಂರಚನೆ (Configuration) ,

© NIMI NOT TO BE REPUBLISHED

ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್ ಮತ್ತು ಪ್ರತಿ ಘಟಕವನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದ ದಿನಾಂಕದ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಇಂತಹ ಕೋಷ್ಟಕಗಳು ಸಿಸ್ಟಮ್ ಸಂರಚನೆ ಹೊಂದಿರಬೇಕಾದ ಮಾಹಿತಿಯ ಕಲ್ಪನೆಯನ್ನು ನೀಡುತ್ತವೆ. ಇದರ ಸಾರಾಂಶ ಈ ಕೆಳಗಿನಂತಿದೆ :

- ಪ್ರೊಸೆಸರ್‌ನ ತಯಾರಿಕೆ, ಮಾದರಿ ಮತ್ತು ವೇಗ
- ಸಿಸ್ಟಮ್ ಮೆಮೊರಿ (RAM), ಮತ್ತು ಮೆಮೊರಿ ಮಾಡ್ಯೂಲ್ ಪ್ರಕಾರ, ಗಾತ್ರ, ಮತ್ತು ಮೆಮೊರಿ ಸಂರಚನೆ
- ಹಾರ್ಡ ಡಿಸ್ಕನ ಗಾತ್ರ, ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ನ ಪ್ರಕಾರ
- ಕಾಂಪ್ಯಾಕ್ಟ ಡಿಸ್ಕ (CD) ಅಥವಾ ಡಿಜಿಟಲ್ ವರ್ಸಟೈಲ್ ಡಿಸ್ಕ (DVD) ಡ್ರೈವ್‌ಗಳ ತಯಾರಿಕೆ, ಮಾದರಿ ಮತ್ತು ವೇಗ
- ವೀಡಿಯೋ ಅಥವಾ ಗ್ರಾಫಿಕ್ಸ ಅಡಾಪ್ಟರ್‌ನ ಮೆಮೊರಿ ಗಾತ್ರ
- ಮಾಡೆಮ್‌ನ ತಯಾರಿಕೆ, ಮಾದರಿ, ಪ್ರಕಾರ ಮತ್ತು ವೇಗ
- ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ ಆವೃತ್ತಿ ಸಂಖ್ಯೆ
- ಪಿಸಿಯಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಸಾಫ್ಟವೇರ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ಪಟ್ಟಿ
- ಪಿಸಿಗೆ ಜೋಡಿಸಲಾದ ಬಾಹ್ಯ ಉಪಕರಣಗಳ ಪಟ್ಟಿ. ಮತ್ತು ಈ ಉಪಕರಣಗಳು ಯಾವ ಪೋರ್ಟಗೆ ಜೋಡಿಸಲಾಗಿವೆ ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತವೆ

ಪಿಸಿಯಲ್ಲಿ ಪ್ರತಿ ಬಾರಿ ನಿರ್ವಹಣೆ ಕಾರ್ಯವು ನಡೆದಾಗ ಅದನ್ನು ನಿರ್ವಹಣಾ ದಾಖಲೆಯಲ್ಲಿ ನವೀಕರಿಸಬೇಕು ಯಾವುದೇ ಸಮಯದಲ್ಲಿ ಹೊಸ ಅಥವಾ ಬದಲಿ ಹಾರ್ಡವೇರನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದಾಗ , ಆ ಚಟುವಟಿಕೆಯನ್ನು ದಾಖಲಿಸಿ ಮತ್ತು ಸಿಸ್ಟಮ್ ಕಾನ್ಫಿಗರೇಷನ್‌ನ್ನು ನವೀಕರಿಸಿ. ಈ ಚಟುವಟಿಕೆಯ ದಾಖಲಿಸುವಿಕೆ ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರಬೇಕು

- ಚಟುವಟಿಕೆಯ ದಿನಾಂಕ ಅಥವಾ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಮಾಡಿದ ಬದಲಾವಣೆಗಳು
- ಪಿಸಿಗೆ ಸೇರಿಸಲಾದ ಅಥವಾ ತೆಗೆಯಲಾದ ಹಾರ್ಡವೇರ್‌ನ ತಯಾರಿಕೆ, ಮಾದರಿ, ಮತ್ತು ಸರಣಿ ಸಂಖ್ಯೆ
- ಪಿಸಿಗೆ ಸೇರಿಸಲಾದ ಯಾವುದೇ ಸಾಫ್ಟವೇರ್‌ನ ಹೆಸರು, ಆವೃತ್ತಿ ಮತ್ತು ಪ್ರಕಾಶಕರ ಹೆಸರು
- ಬೇಸಿಕ್ ಇನ್ಪುಟ್ /ಔಟಪುಟ್ ಸಿಸ್ಟಮ್‌ನ ಕಾನ್ಫಿಗರೇಷನ್‌ನಲ್ಲಿ ಮಾಡಲಾದ ಬದಲಾವಣೆಗಳ ವಿವರವಾದ ಮಾಹಿತಿ ಅಥವಾ ಹೊಸ ಸಾಧನ ಅಥವಾ ಸಾಫ್ಟವೇರ್ ನಲ್ಲಿ ಇತರ ಕಾನ್ಫಿಗರೇಷನ್ ಬದಲಾವಣೆಗಳ ಮಾಹಿತಿಯನ್ನು ನಮೂದಿಸಬೇಕು.

ಸಮಸ್ಯೆ ನಿವಾರಣಾ ವಿಧಾನ (Troubleshooting approach)

ಪಿಸಿ ಗೆ ಸಂಬಂಧಿಸಿದ ಯಾವುದೇ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು, ಮೊದಲು ಆ ಪಿಸಿಯ ನಿರ್ವಹಣಾ ದಾಖಲಾತಿಯನ್ನು ಪರಿಶೀಲಿಸಿ ಮತ್ತು ಸಮಸ್ಯೆಯನ್ನು ಪ್ರತ್ಯೇಕಿಸಲು ವ್ಯವಸ್ಥಿತ ವಿಧಾನವನ್ನು ಅನುಸರಿಸಿ. ಪ್ರಮಾಣಿತ ಸಮಸ್ಯೆ ಪರಿಹಾರ ಪ್ರಕ್ರಿಯೆಯು ಈ ಕೆಳಗಿನ ಹಂತಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ಸಮಸ್ಯೆಯನ್ನು ಗುರುತಿಸಿ: ಇದು ಈ ಪ್ರಕ್ರಿಯೆಯ ಅತ್ಯಂತ ಕಷ್ಟಕರವಾದ ಭಾಗವಾಗಿದೆ. ಈ ಹಂತವನ್ನು ಯಶಸ್ವಿಯಾಗಿ ನಿರ್ವಹಿಸಲು ಸಮಸ್ಯೆಯ ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿ.

ಸಂಭವನೀಯ ಕಾರಣಗಳನ್ನು ಗುರುತಿಸಿ: ಸಮಸ್ಯೆಯ ಎಲ್ಲ ಲಕ್ಷಣಗಳನ್ನು ವಿಶ್ಲೇಷಿಸಿ ಮತ್ತು ಹೆಚ್ಚು ಸಾಧ್ಯತೆ ಇರುವ ಮತ್ತು ಕಡಿಮೆ ಸಾಧ್ಯತೆ ಇರುವ ಎಲ್ಲ ಕಾರಣಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಲು ಪ್ರಯತ್ನಿಸಿ.

ಸಂಭವನೀಯ ಪರಿಹಾರಗಳನ್ನು ಗುರುತಿಸಿ : ಗುರುತಿಸಲ್ಪಟ್ಟ ಪ್ರತಿಯೊಂದು ಕಾರಣಗಳಿಗೂ ಪರಿಹಾರಗಳನ್ನು ಗುರುತಿಸಿ. ಪ್ರತಿ ಸಂಭವನೀಯ ಕಾರಣವು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಪರಿಹಾರಗಳನ್ನು ಹೊಂದಿರಬಹುದು.

ಸಂಭವನೀಯ ಪರಿಹಾರಗಳನ್ನು ವಿಶೇಷಿಸಿ : ಎರಡು ಪರಿಹಾರಗಳು ಒಂದೇ ಫಲಿತಾಂಶವನ್ನು ನೀಡಿದರೆ ಯಾವುದು ಹೆಚ್ಚು ಮಿತವ್ಯಯವಾಗಿದೆಯೋ ಅದನ್ನು ಪರಿಗಣಿಸಿ ಮತ್ತು ಉಪಯೋಗಿಸಿ.

ಈ ಮೇಲಿನ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿದ ನಂತರ ಮತ್ತು ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಿದ ನಂತರ ನಿರ್ವಹಣಾ ದಾಖಲಾತಿಯನ್ನು ನವೀಕರಿಸಿ ಮತ್ತು ಅದರಲ್ಲಿ ಅಗತ್ಯ ದಾಖಲೆಗಳನ್ನು ನಮೂದಿಸಿ. ಕೆಲವೊಂದು ಸಂದರ್ಭಗಳಲ್ಲಿ ಸಮಸ್ಯೆಗಳು ತುಂಬಾ ಸ್ಪಷ್ಟವಾಗಿರಬಹುದು ಮತ್ತು ಪರಿಹಾರವೂ ತುಂಬಾ ಪಾರದರ್ಶಕವಾಗಿರಬಹುದು ಮತ್ತು ಅಂತಹ ಸಂದರ್ಭಗಳಲ್ಲಿಯೂ ಕೂಡ ವ್ಯವಸ್ಥಿತ ವಿಧಾನವನ್ನು ಅನುಸರಿಸಲು ಅಭ್ಯಾಸ ಮಾಡುವುದಕ್ಕಾಗಿ ಮೇಲೆ ತಿಳಿಸಿದ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಲು ಪ್ರಯತ್ನಿಸಿ. ಪಿಸಿಯಲ್ಲಿ ಸಮಸ್ಯೆ ಬಂದಾಗಲೆಲ್ಲ, ಈ ಹಂತಗಳನ್ನು ಅನುಸರಿಸುವಾಗ , ಈ ಕೆಳಗಿನ ಪ್ರಶ್ನೆಗಳಿಗೆ ಉತ್ತರಿಸುತ್ತಾ ಸಿಸ್ಟಮ್ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಲು ಪ್ರಯತ್ನಿಸಿ.

- ಯಾವ ಸಂದರ್ಭಗಳಲ್ಲಿ ಈ ಸಮಸ್ಯೆ ಬಂದಿದೆ?
- ಬೀಪ್ ಕೋಡ್ ಗಳ ರೂಪದಲ್ಲಿ ಯಾವುದೇ ಸೂಚನೆಗಳು ಇದ್ದವೇ? ದೋಷ ಸಂದೇಶಗಳು ಅಥವಾ ಯಾವುದೇ ಸ್ಪಷ್ಟ ಲಕ್ಷಣಗಳು ಇದ್ದವೆ?
- ಸಮಸ್ಯೆ ಸಂಭವಿಸಿದಾಗ ಯಾವ ಸಾಫ್ಟವೇರ್‌ಗಳು ಸಕ್ರೀಯವಾಗಿದ್ದವು?
- ಇದು ಮೊದಲ ಬಾರಿಗೆ ಸಂಭವಿಸಿದೆಯೇ ಅಥವಾ ಈ ಹಿಂದೆಯೂ ಕೂಡ ಸಂಭವಿಸಿದೆಯೆ?

© NIMI NOT TO BE REPUBLISHED

- ಈ ಸಮಯದಲ್ಲಿ ಯಾವುದೇ ಕಾನ್ಫಿಗರೇಷನ್ ಬದಲಾವಣೆ ಮಾಡಲಾಗಿದೆಯೆ ? ಇದಕ್ಕಾಗಿ ಅವಶ್ಯಕವಿರುವ ರೀಸ್ಟಾರ್ಟ ಮಾಡಲಾಗಿದೆಯೆ?

ಪಿಸಿಯನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು (Optimizing the PC):

ಸಂಪೂರ್ಣವಾಗಿ ಉತ್ತಮವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿದ್ದ ಪಿಸಿಯು, ನಿಧಾನವಾಗುತ್ತಿದ್ದರೆ ಅಥವಾ ಹೊಸ ಸಾಫ್ಟವೇರ್‌ನ ಬೇಡಿಕೆಗಳಿಗೆ ವೇಗವನ್ನುಉಳಿಸಿಕೊಳ್ಳಲುಸಾಧ್ಯವಾಗದಿದ್ದರೆ,ಇದಕ್ಕೆಸಂಭವನೀಯ ಪರಿಹಾರವೆಂದರೆ ಪಿಸಿಯನ್ನು ನವೀಕರಿಸುವುದು ಅಥವಾ ಪಿಸಿಯ ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು ಹೆಚ್ಚಿಸಲು ಪಿಸಿಯನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು. ಉತ್ತಮಗೊಳಿಸುವ ಹಂತಗಳಿಗೆ ವೆಚ್ಚವಾಗಬಹುದು, ಆದರೆ ಅನೇಕ ಪರಿಹಾರಗಳು ನೀವು ಈಗಾಗಲೇ ಹೊಂದಿರುವ ಸಾಫ್ಟವೇರನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ ಅಥವಾ ಸಾಫ್ಟವೇರ್‌ಗಳು ವೆಬ್ ನಲ್ಲಿ ಸುಲಭವಾಗಿ ಲಭ್ಯವಿದೆ.

BIOS ಮತ್ತು ಬೂಟ್ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು (Optimizing the BIOS and Boot process) :

BIOS ಸೆಟಪ್ ಕಾನ್ಫಿಗರೇಷನ್ CMOS ನಲ್ಲಿ ಅನೇಕ ಸೆಟ್ಟಿಂಗಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಸಿಸ್ಟಮ್ ಎಷ್ಟು ಬೇಗನೇ ಬೂಟ್ ಆಗುತ್ತದೆ ಮತ್ತು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ಎಂಬುದು ಈ ಸೆಟ್ಟಿಂಗಗಳನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ. ಸಿಸ್ಟಮ್ ಕ್ಯಾಶಿಂಗ ಅಥವಾ ತ್ವರಿತ POST ಪ್ರಕ್ರಿಯೆ ಮುಂತಾದ ಮೌಲ್ಯಯುತವಾದಂತಹ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಬಳಸುವುದರಿಂದ ಅತ್ಯುತ್ತಮವಾದಂತಹ ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು ಹೊಂದಬಹುದು.

ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು (Optimizing the Hard disk) : ವಿಂಡೊಸ್ ಸ್ಕ್ಯಾನ್ ಡಿಸ್ಕ ಮತ್ತು ಡಿಸ್ಕ ಡೀಫ್ರ್ಯಾಗ್ಮೆಂಟರ್ (disk defragmentor) ಉಪಯುಕ್ತತೆಗಳು ಬಳಕೆ ಮತ್ತು ವೇಗದ ವಿಷಯದಲ್ಲಿ ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಉತ್ತಮಗೊಳಿಸುವಲ್ಲಿ ಲಭ್ಯವಿರುವ ಅತ್ಯುತ್ತಮ ಸಾಧನಗಳಾಗಿವೆ. ಸ್ಕ್ಯಾನ್ ಡಿಸ್ಕನ್ನು , ಡಿಸ್ಕನಲ್ಲಿರುವ ದೋಷಗಳನ್ನು ಪರೀಕ್ಷಿಸಲು ಮತ್ತು ಅವುಗಳನ್ನು ಸರಿಪಡಿಸಲು ಅಥವಾ ಮುಂದೆ ಬರುವ ದೋಷಗಳನ್ನು ತಡೆಗಟ್ಟಲು ಬಳಕೆಯ ಕೋಷ್ಟಕಗಳಿಂದ ಡಿಸ್ಕನ ಸರಿಪಡಿಸಲಾಗದ ಪ್ರದೇಶಗಳನ್ನು ತೆಗೆದುಹಾಕಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದೇ ರೀತಿ ಡಿಸ್ಕ ಡಿಫ್ರ್ಯಾಗ್ಮೆಂಟರ್ ಡಾಟಾ ಫೈಲ್ ಫ್ರ್ಯಾಗ್ಮೆಂಟ್ ಗಳನ್ನು ಹೆಚ್ಚು ಉತ್ತಮವಾಗಿ ಮತ್ತು ತಾರ್ಕಿಕಸ್ವರೂಪದಲ್ಲಿ ಆಯೋಜಿಸುತ್ತದೆ. ಇದರಿಂದ ಮಾಹಿತಿಯನ್ನು ವೇಗವಾಗಿ ಮತ್ತು ಕಡಿಮೆ ಹೆಡ್ ಚಲನೆಯಿಂದ ಪಡೆಯಬಹುದು.

ವಿಸ್ತರಣೆ ಕಾರ್ಡಗಳನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು (Optimizing the expansion cards): I/O ನಿಯಂತ್ರಕಗಳು ಮತು ಇತರ ವಿಸ್ತರಣಾ ಕಾರ್ಡಗಳನ್ನು ಉತ್ತಮಗೊಳಿಸುವ ಮಾರ್ಗವೆಂದರೆ ಅವುಗಳನ್ನು ಸರಿಯಾದ ಕ್ರಮದಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದಾಗಿದೆ. ಅವುಗಳನ್ನು ಸರಿಯಾದ ಕ್ರಮದಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡದಿದ್ದರೂ ಏನೂ ಹಾನಿಯಾಗುವುದಿಲ್ಲ ಆದರೆ ಅವುಗಳನ್ನು ಸರಿಯಾದ ಅನುಕ್ರಮದಲ್ಲಿ ಇಡುವುದರಿಂದ ಪ್ರಯೋಜನ ಪಡೆಯಬಹುದು. ಪೆಂಟಿಯಮ್ ಸಿಸ್ಟಮ್ ನಲ್ಲಿ PCI ಕಾರ್ಡಗಳನ್ನು ಬಳಸಿ ಮತ್ತು ಸಾಧ್ಯವಾದರೆ ISA ಕಾರ್ಡಗಳನ್ನು ಬಳಸಬೇಡಿ. ವೀಡಿಯೋ ಕಾರ್ಡಗಳು , ಧ್ವನಿ (sound) ಕಾರ್ಡಗಳು , ಎನ್ ಐ ಸಿ ಗಳು , ಮಾಡೆಮ್‌ಗಳು ಮತ್ತು ಎಸ್ ಸಿ ಎಸ್ ಐ ಅಡಾಪ್ಟರ್‌ಗಳು ಮುಂತಾದ ಎಲ್ಲ ಐ /ಓ ಅಡಾಪ್ಟರ್ ಗಳು ಪಿಸಿಐ ವಿಸ್ತರಣಾ ಬಸ್‌ಗೆ ಲಭ್ಯವಿರುತ್ತವೆ. ಮದರಬೋರ್ಡ ದಾಖಲೆಗಳನ್ನು ನೋಡಿ ಮತ್ತು ಮೊದಲನೇ PCI ಸ್ಲಾಟ್ ನಲ್ಲಿ ವೀಡಿಯೋ ಕಾರ್ಡನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿ. ಅದೇ ಕ್ರಮದಲ್ಲಿ ಎನ ಐ ಸಿ , ಮಾಡೆಮ್ ಮತ್ತು ಸೌಂಡ ಕಾರ್ಡಗಳನ್ನು ಅಳವಡಿಸಿ.

ಪ್ರೊಸೆಸರ್‌ನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು (Optimizing the processor) : ಈ ಕೆಳಗಿನ ರೀತಿಯಿಂದ ಪ್ರೊಸೆಸರ್‌ನ್ನು ವೇಗಗೊಳಿಸಬಹುದು

- ಅದನ್ನು ಅತ್ಯಂತ ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಅಥವಾ ಉನ್ನತ ಮಟ್ಟದ ಪ್ರೊಸೆಸರ್‌ನಿಂದ ಬದಲಾಯಿಸಿ
- ಪ್ರೊಸೆಸರ್ ತಯಾರಕರಿಂದ ಉಪಯುಕ್ತತೆಯನ್ನು ಬಳಸಿ ಪ್ರೊಸೆಸರ್‌ನ ತರ್ಕಕ್ಕೆ ತಕ್ಕ ಪರಿಹಾರಗಳನ್ನು ಉಪಯೋಗಿಸಿ
- ಪ್ರೊಸೆಸರ್‌ನ್ನು ಓವರ್‌ಕ್ಲಾಕ್ ಮಾಡಿ.

ಪ್ರೊಸೆಸರ್‌ನ್ನು ಉನ್ನತ ಮಟ್ಟದ ಅಥವಾ ವೇಗದ ಪ್ರೊಸೆಸರ್‌ನೊಂದಿಗೆ ಬದಲಾಯಿಸುವ ಅವಶ್ಯಕತೆ ಏಕೆ ಇರುತ್ತದೆ. ಎಂದರೆ ನಿಮ್ಮ ಮದರಬೋರ್ಡ ಮತ್ತು ಚಿಪ್ ಸೆಟ್, ಹೊಸ ಪ್ರೊಸೆಸರ್‌ನ್ನು ತಾರ್ಕಿಕವಾಗಿ ಮತ್ತು ಭೌತಿಕವಾಗಿ ಎರಡೂ ರೀತಿಯಿಂದ ಬೆಂಬಲಿಸುತ್ತವೆ. ತಾರ್ಕಿಕವಾಗಿ ಚಿಪ್ ಸೆಟ್ ಮತ್ತು ಮದರಬೋರ್ಡ , ಪ್ರೊಸೆಸರ್‌ನ ಬಸ್ ಸ್ಪೀಡನ್ನು ಬೆಂಬಲಿಸಬೇಕು ಮತ್ತು ಅದಕ್ಕೆ ಅವಶ್ಯಕವಿರುವ ಅಗತ್ಯ ಸರ್ಕಿಟ್ ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಸಂಪೂರ್ಣ ಮದರಬೋರ್ಡನ್ನು ಬದಲಾಯಿಸುವುದು ಹೆಚ್ಚು ಉತ್ತಮವಾಗಿರುತ್ತದೆ. ಕೆಲವು ಬಾರಿ ಪ್ರೊಸೆಸರ್‌ನ ತಯಾರಕರು ಕೆಲವು ಉಪಯುಕ್ತತೆಗಳನ್ನು ಬಿಡುಗಡೆ ಮಾಡಬಹುದು. ಇದರಿಂದ ಪ್ರೊಸೆಸರ್‌ನ ಸಾಮರ್ಥ್ಯಗಳಾದ ವೀಡಿಯೋ ಸಂಸ್ಕರಣೆ , ಬಫರ್ ನಿರ್ವಹಣೆ, ಕ್ಯಾಶಿಂಗ , ಮತ್ತು ಇತರ ಪ್ರೊಸೆಸರ್ ಆಧಾರಿತ ಕಾರ್ಯಗಳನ್ನು ಸುಧಾರಿಸುತ್ತದೆ. ಪ್ರೊಸೆಸರ್‌ನ್ನು ಓವರ್ ಕ್ಲಾಕ್ ಮಾಡುವುದು ಎಂದರೆ ಅದು ಬೆಂಬಲಿಸುವ ವೇಗಕ್ಕಿಂತ ಹೆಚ್ಚು ವೇಗವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು. ಹೆಚ್ಚಿನ ಪ್ರೊಸೆಸರ್‌ಗಳು ಅವುಗಳ ಸಾಮಾನ್ಯ ವೇಗಕ್ಕಿಂತ ಹೆಚ್ಚು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಪ್ರೊಸೆಸರ್‌ನ ಸಾಮಾನ್ಯ ವೇಗವೆಂದರೆ ನಿರ್ದಿಷ್ಟ ಚಿಪ್ ಸೆಟ್ , ಮದರಬೋರ್ಡ, ಕೂಲಿಂಗ ಸಿಸ್ಟಮ್ ಮತ್ತು ಪಿಸಿಯ ಇತರ ಘಟಕಗಳೊಂದಿಗೆಕಾರ್ಯನಿರ್ವಹಿಸುವವೇಗವಾಗಿದೆ.ಪ್ರೊಸೆಸರ್‌ನ ವೇಗವನ್ನು ಹೆಚ್ಚಿಸುವುದು , ಪ್ರೊಸೆಸರ್‌ನ ತಾಪಮಾನವನ್ನು ಹೆಚ್ಚಿಸುವಂಥ ಸಮಸ್ಯೆಗಳನ್ನು ಉಂಟುಮಾಡಬಹುದು. ಇದರಿಂದ

© NIMI NOT TO BE REPUBLISHED

ಆಗಾಗ್ಗೆ ಸಿಸ್ಟಮ್ ಲಾಕ್ ಆಗಲು ಮತ್ತು ಮೆಮೊರಿ ಸಮಸ್ಯೆಗಳು ಮತ್ತು ಇತರ ಸಮಸ್ಯೆಗಳನ್ನು ಉಂಟು ಮಾಡುತ್ತದೆ.

ಸಾಫ್ಟವೇರ್ ಸಮಸ್ಯೆ ಅಲ್ಲದ ಸಮಸ್ಯೆ ನಿವಾರಣೆಯ ಮೂಲಗಳು(Troubleshooting sources of Non-software problems):

ಕೆಲವು ಸಮಯದಲ್ಲಿ ಯಾವುದೇ ಸ್ಪಷ್ಟ ಸಾಫ್ಟವೇರ್ ಕಾರಣಗಳಿಲ್ಲದೆ ಪಿಸಿ ವಿಫಲವಾದಾಗ, ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಪರಿಶೀಲಿಸಿ

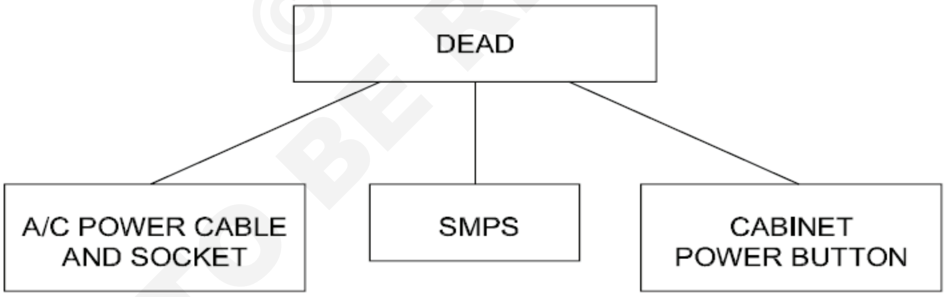
- ಸರಿಯಾದ ಎಸಿ ವಿದ್ಯುತ್ತನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ
- ಕಂಪ್ಯೂಟರ್ ವೈರಸ್‌ಗಾಗಿ ಪಿಸಿಯನ್ನು ಸ್ಕ್ಯಾನ್ ಮಾಡಿ
- ಸಿಪಿಯು ಫ್ಯಾನ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ
- ಬಾಹ್ಯ I/O ಕನೆಕ್ಟರ್ ಗಳ ಸರಿಯಾದ ಸಂಪರ್ಕವನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ
- ವಿಸ್ತರಣೆ ಕಾರ್ಡ್‌ಗಳನ್ನು ಸರಿಯಾಗಿ ಜೋಡಿಸಿ ಮತ್ತು ವಿದ್ಯುತ್ ಶಕ್ತಿಯನ್ನು ಪರಿಶೀಲಿಸಿ ಮತ್ತು ಆಂತರಿಕ ಸಾಧನಗಳ ಡೇಟಾ ಕೇಬಲ್‌ಗಳನ್ನು ಪರಿಶೀಲಿಸಿ
- ಹೆಚ್ಚಿನ ಬೂಟ್ ಸಮಸ್ಯೆಗಳು ಇತ್ತೀಚಿನ ಬದಲಾವಣೆಗಳಿಂದಾಗಿರುತ್ತವೆ. BIOS ಸೆಟಪ್ ಕಾನ್ಫಿಗರೇಷನ್ ಡೇಟಾವನ್ನು ಪರಿಶೀಲಿಸಿ.
- ಯಾವುದೇ ಹೊಸ ಹಾರ್ಡವೇರ್ ಅಥವಾ ಸಾಫ್ಟವೇರ್‌ಗಳು ಸರಿಯಾಗಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲು , ತಯಾರಕರ ವೆಬ್ ಸೈಟಗೆ ಭೇಟಿ ನೀಡಿ.
- ಯಾವುದೇ ಹೊಸ ಹಾರ್ಡವೇರ್ ಅಥವಾ ಸಾಫ್ಟವೇರ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದ್ದರೆ ಯಾವುದೇ ಸಮಸ್ಯೆಯ ಮೂಲವನ್ನು ಪರಿಶೀಲಿಸಿ.

ಪಿಸಿ ಸಮಸ್ಯೆ ನಿವಾರಣೆಯ ಮುಂಬರುವ ಪಾಠಗಳಲ್ಲಿ ಬಳಕೆದಾರರು ಸಾಮಾನ್ಯವಾಗಿ ಎದುರಿಸುವ ವಿವಿಧ ಸಮಸ್ಯೆಗಳನ್ನು ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ (problem Tree)ಗಳ ಮೂಲಕ ತೋರಿಸಲಾಗಿದೆ. ಪ್ರತೀ ಟ್ರೀಯ ನಿರ್ದಿಷ್ಟ ಸಮಸ್ಯೆಯನ್ನು ವಿಶ್ಲೇಷಿಸಲು ಮತ್ತು ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಈ ಪಾಠದಲ್ಲಿನ ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ , ಪಿಸಿ ಮತ್ತು ಮಾನಿಟರ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿಲ್ಲ ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತಿದೆ.

ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ : ಪಿ ಟಿ - 01

ಗಮನಿಸಿದ ಲಕ್ಷಣ : ಡೆಡ್

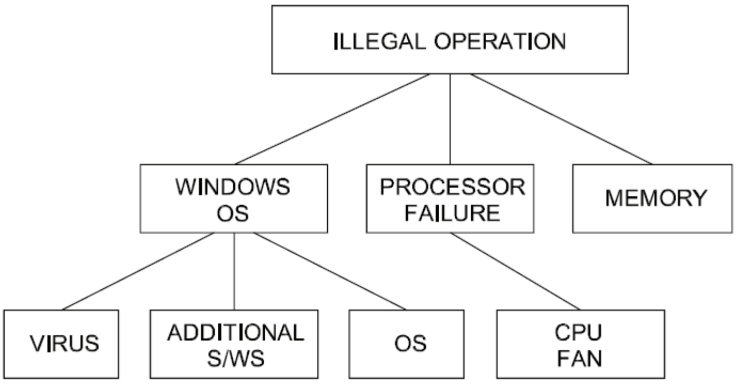
ಹೆಚ್ಚುವರಿ ಲಕ್ಷಣ : ಮಾನಿಟರ್ ಪ್ರದರ್ಶನವಿಲ್ಲ



ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ : ಪಿ ಟಿ - 02

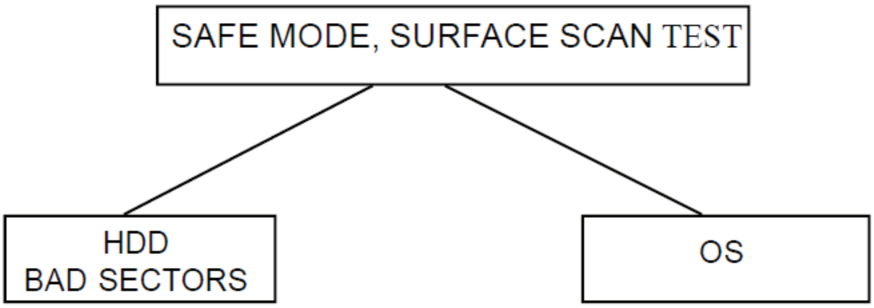
ಗಮನಿಸಿದ ಲಕ್ಷಣ : ವಿಂಡೊಸ್ illegal operation ಎಂದು ತೋರಿಸುತ್ತದೆ.

ಹೆಚ್ಚುವರಿ ಲಕ್ಷಣ : ವಿಂಡೊಸ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿಲ್ಲ



ವಿಂಡೊಸ್ illegal operation ಎಂದು ತೋರಿಸುತ್ತದೆ. ಈ ಸಮಸ್ಯೆಗೆ ,ಸಮಸ್ಯೆ ನಿವಾರಣೆ ಚಾರ್ಟ(ಟಿಎಸ್ ಸಿ -02) ಮತ್ತು ಸರ್ವಿಸ್ ಫ್ಲೋ ಸಿಕ್ವೆನ್ಸ್ (ಎಸ್ ಎಫ್ ಎಸ್ -02) ನ್ನು ಚರ್ಚಿಸಿ.

ಪ್ರಾಯೋಗಿಕ ಪಾಠಗಳಿಗಾಗಿ ಮೇಲೆ ಚರ್ಚಿಸಲಾದ ವಿವಿಧ ದೂರುಗಳಿಗೆ ನಿಮಗೆ ನೀಡಲಾದ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಅನ್ವಯಿಸಲಾಗುತ್ತದೆ



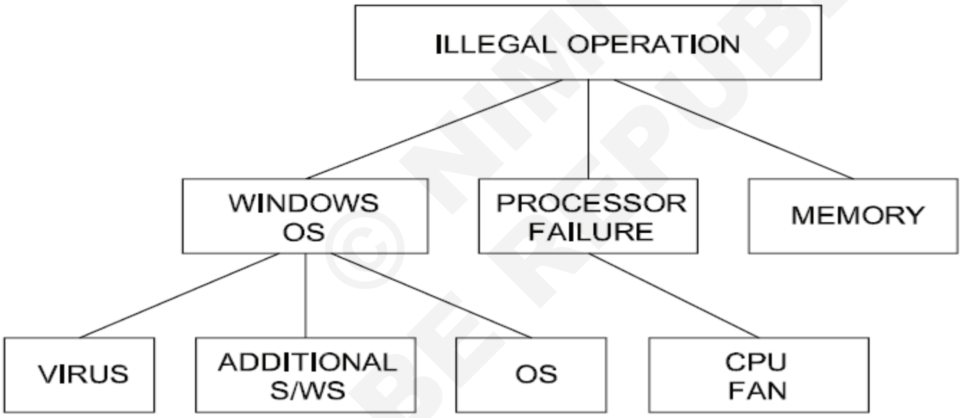
``When windows is started system runs surface test and goes to safemode `` ಈ ಸಮಸ್ಯೆಗೆ , ಸಮಸ್ಯೆ ನಿವಾರಣೆ ಚಾರ್ಟ(ಟಿಎಸ್ ಸಿ -01) ಮತ್ತು ಸರ್ವಿಸ್ ಫ್ಲೋ ಸಿಕ್ವೆನ್ಸ್ (ಎಸ್ ಎಫ್ ಎಸ್ -01) ನ್ನು ಚರ್ಚಿಸಿ. ಪಾಠ 2.33 ಉಲ್ಲೇಖಿಸುತ್ತದೆ.

ಪ್ರಾಯೋಗಿಕ ಪಾಠಗಳಿಗಾಗಿ ಮೇಲೆ ಚರ್ಚಿಸಲಾದ ವಿವಿಧ ದೂರುಗಳಿಗೆ ನಿಮಗೆ ನೀಡಲಾದ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಅನ್ವಯಿಸಲಾಗುತ್ತದೆ.

ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ : ಪಿ ಟಿ - 02

ಗಮನಿಸಿದ ಲಕ್ಷಣ : ವಿಂಡೋಸ್ illegal operation ಎಂದು ತೋರಿಸುತ್ತದೆ

ಹೆಚ್ಚುವರಿ ಲಕ್ಷಣ : ವಿಂಡೋಸ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿಲ್ಲ



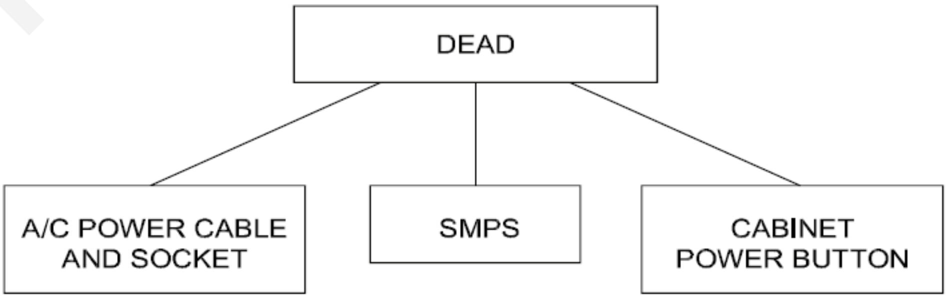
``When windows is started system runs surface test and goes to safemode`` ಈ ಸಮಸ್ಯೆಗೆ ,ಸಮಸ್ಯೆ ನಿವಾರಣೆ ಚಾರ್ಟ(ಟಿಎಸ್ ಸಿ -01) ಮತ್ತು ಸರ್ವಿಸ್ ಫ್ಲೋ ಸಿಕ್ವೆನ್ಸ್ (ಎಸ್ ಎಫ್ ಎಸ್ -01) ನ್ನು ಚರ್ಚಿಸಿ.

ಪ್ರಾಯೋಗಿಕ ಪಾಠಗಳಿಗಾಗಿ ಮೇಲೆ ಚರ್ಚಿಸಲಾದ ವಿವಿಧ ದೂರುಗಳಿಗೆ ನಿಮಗೆ ನೀಡಲಾದ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಅನ್ವಯಿಸಲಾಗುತ್ತದೆ

ಪ್ರಾಬ್ಲಮ್ ಟ್ರೀ : ಪಿ ಟಿ - 03

ಗಮನಿಸಿದ ಲಕ್ಷಣ : ಡಿವಿಡಿ ಡ್ರೈವ್ ಓದಲಾಗುವುದಿಲ್ಲ

ಹೆಚ್ಚುವರಿ ಲಕ್ಷಣ :



© NIMI NOT TO BE REPUBLISHED

ವಿಂಡೋಸ್‌ನಲ್ಲಿ ಅವಶ್ಯಕ ಅಪ್ಲಿಕೇಷನ್ ಸಾಫ್ಟವೇರ್‌ಗಳನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು. ಉದಾ: ಆಫೀಸ್ ಪ್ಯಾಕೇಜ್, ಪಿಡಿಎಫ್ ರೀಡರ್, ಮೀಡಿಯಾ ಪ್ಲೇಯರ್ ಇತ್ಯಾದಿ (Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಸಾಫ್ಟವೇರ್‌ನ ವಿವಿಧ ಪ್ರಕಾರಗಳನ್ನು ವಿವರಿಸಿ

ಅಪ್ಲಿಕೇಷನ್ ಸಾಫ್ಟವೇರ್ (Application Software)

ಒಂದು ನಿರ್ದಿಷ್ಟ ಉದ್ದೇಶಕ್ಕಾಗಿ ತಯಾರಿಸಲಾದ ಸಾಫ್ಟವೇರಗೆ ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ ಎನ್ನುತ್ತೇವೆ. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಅಂತಿಮ ಬಳಕೆದಾರರು ಬಳಸುವ ಒಂದು ಪ್ರೋಗ್ರಾಂ ಅಥವಾ ಪ್ರೋಗ್ರಾಂಗಳ ಸಂಗ್ರಹ ಆಗಿದೆ. ಇದನ್ನು ಅಪ್ಲಿಕೇಶನ್ ಅಥವಾ ಸರಳವಾಗಿ ಆ್ಯಪ್ ಎಂದು ಕರೆಯಬಹುದು.

ವಾಸ್ತವವಾಗಿ ಸಿಸ್ಟಮ್ ಸಾಫ್ಟವೇರ್ ಮತ್ತು ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಸಾಫ್ಟವೇರ್ ಹೊರತುಪಡಿಸಿ ಉಳಿದ ಎಲ್ಲ ಸಾಫ್ಟವೇರ್‌ಗಳು ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್ ಆಗಿವೆ.

ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್ ವ್ಯಾಖ್ಯಾನ (Application Software Definition)

ಬಳಕೆದಾರರಿಗೆ ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಸಹಾಯ ಮಾಡಲು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಸಾಫ್ಟವೇರ್‌ಗಳನ್ನು ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್ ಎಂದು ಕರೆಯುತ್ತೇವೆ.

ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್‌ನ ವಿವಿಧ ಪ್ರಕಾರಗಳು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ

ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್‌ನ ಪ್ರಕಾರ	ಉದಾಹರಣೆಗಳು
ವರ್ಡ ಪ್ರೊಸೆಸಿಂಗ ಸಾಫ್ಟವೇರ್	ಎಮ್ ಎಸ್ ವರ್ಡ ಮತ್ತು ನೋಟಪ್ಯಾಡ್
ಡಾಟಾಬೇಸ್ ಸಾಫ್ಟವೇರ್	ಒರ್ಯಾಕಲ್, ಎಮ್ ಎಸ್ ಎಕ್ಸೆಸ್ ಇತ್ಯಾದಿ
ಸ್ಪ್ರೆಡ್ ಶೀಟ್ ಸಾಫ್ಟವೇರ್	ಆ್ಯಪಲ್ ನಂಬರ್ಸ್, ಮೈಕ್ರೊಸಾಫ್ಟ ಎಕ್ಸೆಲ್
ಮಲ್ಟಿಮೀಡಿಯಾ ಸಾಫ್ಟವೇರ್	ರಿಯಲ್ ಪ್ಲೇಯರ್, ಮೀಡಿಯಾ ಪ್ಲೇಯರ್
ಪ್ರೆಸೆಂಟೇಶನ್ ಸಾಫ್ಟವೇರ್	ಮೈಕ್ರೊಸಾಫ್ಟ ಪಾವರ್ ಪಾಯಿಂಟ್ , ಕೀನೋಟ್ಸ್
ಎಂಟರಪ್ರೈಸ್ ಸಾಫ್ಟವೇರ್	ಕಸ್ಟಮರ್ ರಿಲೇಶನ್‌ಶಿಪ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಸಿಸ್ಟಮ್
ಇನ್ಫಾರ್ಮೇಶನ್ ವರ್ಕರ್ ಸಾಫ್ಟವೇರ್	ಡಾಕ್ಯುಮೆಂಟೇಶನ್ ಟೂಲ್ಸ್ , ರೀಸೋರ್ಸ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಟೂಲ್ಸ್
ಎಜ್ಯುಕೇಶನಲ್ ಸಾಫ್ಟವೇರ್	ಡಿಕ್ಷನರಿಗಳು : ಎನ್ಕಾರ್ಟಾ, ಬ್ರಿಟಾನಿಕಾಮ್ಯಾಥಮ್ಯಾಟಿಕಲ್ :
ಸಿಮ್ಯುಲೇಶನ್ ಸಾಫ್ಟವೇರ್	ಮ್ಯಾಟ್ಲಾಬೊಥರ್ಸ: ಗೂಗಲ್ ಅರ್ಥ, ನಾಸಾ ವರ್ಡ ವಿಂಡ
ಕಂಟೆಂಟ ಎಕ್ಸೆಸ್ ಸಾಫ್ಟವೇರ್ ಬ್ರೌಸರ್	ಫ್ಲೈಟ್ ಮತ್ತು ಸೈಂಟಿಫಿಕ್ ಸಿಮ್ಯುಲೇಟರ್ಸ್
ಕಂಟೆಂಟ ಎಕ್ಸೆಸ್ ಸಾಫ್ಟವೇರ್ ಬ್ರೌಸರ್	ಎಕ್ಸೆಸ್ಸಿಂಗ್ ಕಂಟೆಂಟ ಥ್ರೂ ಮೀಡಿಯಾ ಪ್ಲೇಯರ್ಸ್ , ವೆಬ್
ಅಪ್ಲಿಕೇಶನ್ ಸೂಟ್ಸ್	ಓಪನ್ ಆಫೀಸ್, ಮೈಕ್ರೊಸಾಫ್ಟ ಆಫೀಸ್
ಸಾಫ್ಟವೇರ್ ಫಾರ್ ಇಂಜನಿಯರಿಂಗ ಮತ್ತು ಪ್ರಾಡಕ್ಟ ಡೆವಲಪ್ ಮೆಂಟ್	ಐಡಿ ಇ ಅಥವಾ ಇಂಟಿಗ್ರೇಟೆಡ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಎನ್ವಿರಾನ್‌ಮೆಂಟ್

ವಿವಿಧ ರೀತಿಯ ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್‌ಗಳನ್ನು ನಾವು ನೋಡಬಹುದು. ಉದಾಹರಣೆಗೆ ಲೈಸೆನ್ಸ್ ಪಡೆದ , ಮಾರಾಟವಾದ, ಫ್ರೀವೇರ್, ಶೇರವೇರ್, ಮತ್ತು ಓಪನ್ ಸೋರ್ಸ್.

ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ ಅಥವಾ ಆನ್‌ಲೈ ನ್‌ನಲ್ಲಿ ರನ್ ಮಾಡಬಹುದು.

ಅಪ್ಲಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್‌ಗಳನ್ನು ಅವುಗಳ ಬಳಕೆಯ ಆಧಾರದ ಮೇಲೆ ಸಹ ಈ ಕೆಳಗಿನಂತೆ ಪ್ರತ್ಯೇಕಿಸಬಹುದು.

- ಯುಟಿಲಿಟಿ ಪ್ರೋಗ್ರಾಂಗಳು
- ಜೆನರಿಕ್ ಪ್ರೋಗ್ರಾಂಗಳು

© NIMI NOT TO BE REPUBLISHED

- ಇಂಟಿಗ್ರೇಟೆಡ್ ಪ್ರೋಗ್ರಾಂಗಳು
- ಸ್ಪೆಸಿಫಿಕ್ ಸಾಫ್ಟವೇರ್
- ಬೆಸ್ಪೋಕ್ ಸಾಫ್ಟವೇರ್
- ವರ್ಡ ಪ್ರೊಸೆಸಿಂಗ್ ಸಾಫ್ಟವೇರ್
- ಡೆಸ್ಕಟಾಪ್ ಪಬ್ಲಿಶಿಂಗ್ ಸಾಫ್ಟವೇರ್
- ಸ್ಪ್ರೆಡ್ ಶೀಟ್ ಸಾಫ್ಟವೇರ್
- ಡಾಟಾಬೇಸ್ ಸಾಫ್ಟವೇರ್
- ಪ್ರೆಸೆಂಟೇಶನ್ ಸಾಫ್ಟವೇರ್
- ಇಂಟರ್ನೆಟ್ ಬ್ರೌಸರ್ಸ್
- ಇಮೇಲ್ ಪ್ರೋಗ್ರಾಂಗಳು
- ಗ್ರಾಫಿಕ್ ಪ್ರೋಗ್ರಾಂಗಳು (ಪಿಕ್ಸೆಲ್ ಬೇಸ್ಡ)
- ಗ್ರಾಫಿಕ್ ಪ್ರೋಗ್ರಾಂಗಳು (ವೆಕ್ಟರ್ ಬೇಸ್ಡ)
- ಕಮ್ಯುನಿಕೇಶನ್ ಸಾಫ್ಟವೇರ್ : ಕಮ್ಯುನಿಕೇಶನ್ ಥ್ರೂ ಆಡಿಯೋ. ವಿಡಿಯೋ ಅಥವಾ ಚಾಟ್ ಬೇಸ್ಡ್

© NIMI NOT TO BE REPUBLISHED

ಬ್ಲೂ ಟೂಥ್ ಮತ್ತು ವೈ ಫೈ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಸಂರಚಿಸುವುದು (Configure Bluetooth and Wi-Fi settings)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- ಬ್ಲೂಟೂಥ್‌ನ ಅರ್ಥವನ್ನು ವಿವರಿಸಿ
- ಬ್ಲೂಟೂಥ್‌ನ ಬಳಸುವ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ಬ್ಲೂಟೂಥ್‌ನ ಪ್ರಮುಖ ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ.

ಪರಿಚಯ (Introduction) :

ಬ್ಲೂಟೂಥ್ ವೈಯರ್‌ಲೆಸ್ ತಂತ್ರಜ್ಞಾನದ ಮಾನದಂಡವಾಗಿದೆ. ಇದು ಕಡಿಮೆ ಅಂತರದಲ್ಲಿ ಡೇಟಾವನ್ನು ವಿನಿಮಯ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ. (ಕಡಿಮೆ ವೇವ್ ಲೆಂತ್ UHF ರೇಡಿಯೋ ತರಂಗಗಳು 2.4 ದಿಂದ 2.485 GHz ವ್ಯಾಪ್ತಿಯಲ್ಲಿ) ಇದು ಡೇಟಾವನ್ನು ಸ್ಥಿರ, ಮೊಬೈಲ್ ಸಾಧನಗಳಿಂದ ಮತ್ತು ಕಟ್ಟಡದ ಪರ್ಸನಲ್ ಏರಿಯಾ ನೆಟ್ವರ್ಕ್ಸ್(PANs) ವಿನಿಮಯ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ. ಇದು ಪ್ರಮಾಣಿತ ವೈಯರ್ ರಿಪ್ಲೇಸಮೆಂಟ್ ಸಂವಹನ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿದೆ. ಪ್ರಾಥಮಿಕವಾಗಿ ಕಡಿಮೆ ವಿದ್ಯುತ್ ಬಳಕೆಗಾಗಿ ಇದನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಪ್ರತಿ ಸಾಧನದಲ್ಲಿ ಕಡಿಮೆ ವ್ಯಾಪ್ತಿ ಆಧರಿಸಿ, ಕಡಿಮೆ ವೆಚ್ಚದ ಟ್ರಾನ್ಸೀವರ್ ಮೈಕ್ರೊಚಿಪ್ ಬಳಸಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

ಇದು ಒಂದೇ ಸಮಯದಲ್ಲಿ ಎಂಟು ಸಾಧನಗಳನ್ನು (ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಉಪಕರಣಗಳು) ಸಂಪರ್ಕಿಸಬಹುದಾಗಿದೆ. ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಉಪಕರಣಗಳಾದ ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಡಿಜಿಟಲ್ ಕ್ಯಾಮರಾಗಳು, ಮೊಬೈಲ್ ಫೋನ್‌ಗಳು ಮತ್ತು ಫ್ಯಾಕ್ಸ್ ಗಳು ಮುಂತಾದವುಗಳಿಗೆ ಚಿಪ್‌ನ್ನು ಪ್ಲಗ್ ಇನ್ ಮಾಡಬಹುದು. ಬ್ಲೂಟೂಥ್ ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ವಿಶೇಷವಾಗಿ ಅನುಕೂಲಕರವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ಕೇಬಲ್ ಇಲ್ಲದೇ ಒಂದು ಮೊಬೈಲ್‌ನಿಂದ ಇನ್ನೊಂದಕ್ಕೆ ಫೈಲ್‌ಗಳನ್ನು ವರ್ಗಾಯಿಸುವಾಗ, ಒಂದು ಪಿಸಿ ಮತ್ತು ಮೊಬೈಲ್ ಫೋನ್ ಗಳ ಮಧ್ಯೆ ಸಂಗೀತ ಮತ್ತು ಫೋಟೊಗಳನ್ನು ಕಳುಹಿಸುವುದು ಇನ್ನೊಂದು ಉಪಯುಕ್ತ ಅಪ್ಲಿಕೇಶನ್ ಆಗಿದೆ.

ಈಸಾಧನಗಳುರೇಡಿಯೊಸಂವಹನವ್ಯವಸ್ಥೆಯನ್ನುಬಳಸುವದರಿಂದ ಅವು ಪರಸ್ಪರ ಒಂದಕ್ಕೊಂದು ಕಾಣುವಂತಿರಬೇಕಿಲ್ಲ. ಅದಾಗ್ಯೂ , ಅರೆ ಆಪ್ಟಿಕಲ್ ವಯರ್‌ಲೆಸ್ ಮಾರ್ಗವು ಇರಬೇಕು.

ಇದರ ವ್ಯಾಪ್ತಿಯು ವಿದ್ಯುತ್‌ಶಕ್ತಿಯ ಪ್ರಮಾಣದ ಮೇಲೆ ಅವಲಂಬಿತವಾಗಿರುತ್ತದೆ. ಆದರೆ ಪರಿಣಾಮಕಾರಿಯಾದ ವ್ಯಾಪ್ತಿಯು 10 ಮೀ ನಿಂದ 100 ಮೀ ವರೆಗೆ ಬದಲಾಗುತ್ತದೆ.

ಬ್ಲೂಟೂಥ್ ಎಂಬ ಹೆಸರು ಸ್ಕ್ಯಾಂಡಿನೇವಿಯನ್ ಮೂಲದ ತಂತ್ರಜ್ಞಾನವನ್ನು ಪ್ರತಿಬಿಂಬಿಸುತ್ತದೆ. ಇದಕ್ಕೆ 10 ನೇ ಶತಮಾನದ ಡ್ಯಾನಿಶ್ ವೈಕಿಂಗ್, ಕಿಂಗ್ ಹರಾಲ್ಡ ಬ್ಲಟಾಂಡಾ (ಇಂಗ್ಲಿಷ್‌ನಲ್ಲಿ ಬ್ಲೂಟೂಥ್ ಎಂದು ಅನುವಾದಿಸಲಾಗಿದೆ) ಎಂದು ಹೆಸರಿಡಲಾಗಿದೆ. ಅವರು ಡೆನ್ಮಾರ್ಕ್ ಮತ್ತು ನಾರ್ವೆಯನ್ನು ಒಗ್ಗೂಡಿಸಿದರು ಮತ್ತು ನಿಯಂತ್ರಿಸಿದರು ಅದ್ದರಿಂದ ಬ್ಲೂಟೂಥ್ ಮೂಲಕ ಸಾಧನಗಳನ್ನು ಒಟ್ಟುಗೂಡಿಸಿ ಸಂವಹನ ನಡೆಸುವ ಸಾಧನವಾಗಿದೆ.

ಬ್ಲೂಟೂಥ್ ಬಳಸುವುದು (Using Bluetooth)

ಬ್ಲೂಟೂಥ್ ಬಳಸಲು, ಆ ಸಾಧನವು ಬ್ಲೂಟೂಥ್ ಸಕ್ರಿಯವಾಗಿರಬೇಕು. ಈ ಉದ್ದೇಶಕ್ಕಾಗಿ ಡಾಂಗಲ್ (dongle) ಎಂಬ ಸಾಧನವನ್ನು ಬಳಸಬಹುದು. ಡಾಂಗಲ್ ಎನ್ನುವುದು ಕಂಪ್ಯೂಟರ್‌ಗೆ ಪ್ಲಗ್ ಮಾಡುವ ಸಾಧನವಾಗಿದೆ.. ಬ್ಲೂಟೂಥ ಬಳಸಲು ಇದನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಿ. ಹೊಂದಾಣಿಕೆಯ ಸಾಧನಗಳ ತಯಾರಕರು ಬ್ಲೂಟೂಥನ್ನು ಬಳಸಲು ತಮ್ಮದೇ ಆದ ಸೂಚನೆಗಳನ್ನು ಹೊಂದಿರುತ್ತಾರೆ. ವಿವರವಾದ ಸೂಚನೆಗಳಿಗಾಗಿ ನೀವು ಕೈಪಿಡಿಯನ್ನು ನೋಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಆದರೆ ಸಾಮಾನ್ಯ ಮಾರ್ಗದರ್ಶಿಯಾಗಿ :

ಬ್ಲೂಟೂಥ್ ವ್ಯವಸ್ಥೆಗಾಗಿ :

(ಸಾಧನಗಳಲ್ಲಿನ ಬ್ಲೂಟೂಥ್ ಚಿತ್ರವನ್ನು ಗುರುತಿಸಿ)

1 ಬ್ಲೂಟೂಥ್ ಆನ್ ಮಾಡಿ ಅಥವಾ ಸಕ್ರೀಯಗೊಳಿಸಿ. ನಿಮ್ಮ ಸಾಧನವು ಕಾಣಿಸುತ್ತಿದೆ ಮತ್ತು ಮರೆಯಾಗಿಲ್ಲ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ. ಇದರಿಂದ ಹತ್ತಿರದ ಇತರ ಸಾಧನಗಳು ಸಂಕೇತವನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

2 ಇತರ ಹೊಂದಾಣಿಕೆಯ ಸಾಧನಗಳನ್ನು ಸಂಪರ್ಕಿಸುವಾಗ ನಿಮ್ಮ ಸಾಧನವನ್ನು ಗುರುತಿಸುವ ಸಲುವಾಗಿ ಅದಕ್ಕೆ ಒಂದು ಹೆಸರನ್ನು ಇಡಿ.

ಮೊಬೈಲ್ ಫೋನ್ಗಳು , ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳು, ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು ಮುಂತಾದ ಸಾಧನಗಳು ಬ್ಲೂಟೂಥ್‌ನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಿದಾಗ ಬ್ಲೂಟೂಥ್ ಐಕಾನ್ ಆನ್ ಆಗಿರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಬ್ಲೂಟೂಥ್ ಸಂಪರ್ಕವನ್ನು ಸ್ಥಾಪಿಸಲು:

1 ನೀವು ಕಳುಹಿಸಲು ಬಯಸುವ ಫೈಲ್ ಅನ್ನು ಹುಡುಕಿ.

2 ಬ್ಲೂಟೂಥ್ ಮೂಲಕ ಕಳುಹಿಸುವ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿ - ನಿಮ್ಮ ಸಾಧನವು ಆ ವ್ಯಾಪ್ತಿಯೊಳಗಿನ ಇತರ ಬೇರೆ ಸಾಧನಗಳನ್ನು ಹುಡುಕಿ ಅವುಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.

3 ನೀವು ಸಂಪರ್ಕಿಸಲು ಬಯಸುವ ಸಾಧನಕ್ಕಾಗಿ ಸ್ಕ್ರಾಲ್ ಮಾಡಿ ಮತ್ತು ಅದನ್ನು ಆಯ್ಕೆ ಮಾಡಿ.

4 ಇತರ ಸಾಧನಕ್ಕೆ ಜೋಡಣೆಯ ಅವಶ್ಯಕತೆ ಇದ್ದರೆ, ನೀವು ಅದಕ್ಕೆ ಪಾಸ್ ಕೋಡನ್ನು ನಮೂದಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ- ಸ್ವಲ್ಪ ಪಿನ್ ಸಂಖ್ಯೆಯಂತೆ ಇರುತ್ತದೆ. ಮತ್ತು ಇತರ ಸಾಧನದಲ್ಲಿ ಅದನ್ನು ನಮೂದಿಸಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

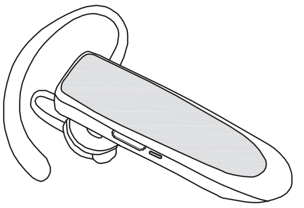
ಸಂಪರ್ಕ ಸ್ಥಾಪನೆಯಾದಾಗ, ಡೇಟಾ ಕಳುಹಿಸಲು ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಸಾಧನಗಳ ನಡುವಿನ ಸ್ಪಷ್ಟ ಕಾಣುವಿಕೆ ಬಗ್ಗೆ ಚಿಂತಿಸುವ ಅಗತ್ಯವಿಲ್ಲ.

ಬ್ಲೂಟೂಥ್ ಅಪ್ಲಿಕೇಶನ್‌ಗಳ ಪಟ್ಟಿ (List of Bluetooth applications)

ಕೆಲವು ಬ್ಲೂಟೂಥ್ ಅಪ್ಲಿಕೇಶನ್‌ಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ:

ಮೊಬೈಲ್ ಫೋನ್ ಮತ್ತು ಹ್ಯಾಂಡ್ಸ್ ಫ್ರೀ ಹೆಡ್ ಸೆಟ್‌ಗಳ ನಡುವಿನ ವೈಯರ್ ಲೆಸ್ ನಿಯಂತ್ರಣ ಮತ್ತು ಸಂವಹನ (Fig1)

Fig 1



BLUETOOTH

CP20N141911

- ಮೊಬೈಲ್ ಫೋನ್ ಮತ್ತು ಬ್ಲೂಟೂಥ್ ಹೊಂದಾಣಿಕೆಯ ಕಾರ್ ಸ್ಟೀರಿಯೋ ವ್ಯವಸ್ಥೆಗಳ ನಡುವಿನ ವಯರ್‌ಲೆಸ್ ನಿಯಂತ್ರಣ ಮತ್ತು ಸಂವಹನ
- ಐಪ್ಯಾಡ್ ಮತ್ತು ಆ್ಯಂಡ್ರಾಯ್ಡ್ ಸಾಧನಗಳಂತಹ ಸ್ಪೀಕರ್‌ಗಳು ಮತ್ತು ಟ್ಯಾಬ್ಲೆಟ್ ನಡುವಿನ ವಯರ್‌ಲೆಸ್ ನಿಯಂತ್ರಣ ಮತ್ತು ಸಂವಹನ
- ಸೀಮಿತ ಸ್ಥಳದಲ್ಲಿನ ಮತ್ತು ಕಡಿಮೆ ಬ್ಯಾಂಡವಿಡ್ಥ್‌ನ ಅವಶ್ಯಕತೆ ಇರುವಲ್ಲಿನ ಪಿಸಿಗಳ ಮಧ್ಯೆ ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕಿಂಗ್
- ಸಾಮಾನ್ಯವಾಗಿ ಮೌಸ್, ಕೀಬೋರ್ಡ ಮತ್ತು ಪ್ರಿಂಟರ್‌ನಂತಹ, ಪಿಸಿಯ ಇನ್‌ಪುಟ್ ಮತ್ತು ಔಟಪುಟ್ ಸಾಧನಗಳ ವಯರ್‌ಲೆಸ್ ಸಂವಹನ
- ಫೈಲ್‌ಗಳು, ಸಂಪರ್ಕ ವಿವರಗಳು, ಕ್ಯಾಲೆಂಡರ್ ನೇಮಕಾತಿಗಳು, ಮತ್ತು ಜ್ಞಾಪನೆಗಳು ಮುಂತಾದವುಗಳನ್ನು OBEX (Object Exchange) ಹೊಂದಿರುವ ಸಾಧನಗಳ ನಡುವೆ ವರ್ಗಾಯಿಸುವುದು
- ಪರೀಕ್ಷಾ ಸಾಧನಗಳಲ್ಲಿನ, ಜಿಪಿಎಸ್ ರಿಸಿವರ್‌ಗಳಲ್ಲಿ ನ ,ವೈದ್ಯಕೀಯ ಉಪಕರಣಗಳಲ್ಲಿನ , ಬಾರ್ ಕೋಡ್ ಸ್ಕ್ಯಾನರ್ಗಳಲ್ಲಿನ ಮತ್ತು ಸಂಚಾರ ನಿಯಂತ್ರಣ ಸಾಧನಗಳಲ್ಲಿನ RS - 232 ಸರಣಿ ಸಂವಹನ ಹೊಂದಿದ ಹಿಂದಿನ ತಂತಿಯನ್ನು ಬದಲಾಯಿಸುವುದು.
- ಎರಡು ಔದ್ಯಮಿಕ ಈಥರ್‌ನೆಟ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳ ನಡುವಿನ ವಯರ್‌ಲೆಸ್ ಬ್ರಿಡ್ಜ
- ಆಟಗಳಲ್ಲಿ ವಯರ್‌ಲೆಸ್ ನಿಯಂತ್ರಕಗಳು
- ಮೊಬೈಲ್ ಫೋನ್‌ಗಳಲ್ಲಿ ಕಳ್ಳತನ ಅಥವಾ ವಸ್ತುಗಳ ನಷ್ಟವನ್ನು ತಡೆಗಟ್ಟಲು ವೈಯಕ್ತಿಕ ಭದ್ರತಾ ಅಪ್ಲಿಕೇಶನ್ ಆಗಿದೆ. ಸಂರಕ್ಷಿತ ವಸ್ತು ಬ್ಲೂಟೂಥ್ ಮಾರ್ಕರ್‌ನ್ನು ಹೊಂದಿದೆ (ಉದಾ.,ಟ್ಯಾಗ್) ಎಂದರೆ ಫೋನ್‌ನೊಂದಿಗೆ ನಿರಂತರ ಸಂವಹನ ಹೊಂದಿದೆ. ಒಂದು ವೇಳೆ ಸಂಪರ್ಕವು ಮುರಿದುಹೋದರೆ (ಮಾರ್ಕರ್ ಫೋನ್ ವ್ಯಾಪ್ತಿಯಿಂದ ಹೊರಗೆ ಹೋದರೆ) ಅಲಾರಾಂ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ವೈ - ಫೈ (WI - FI):

ವೈ - ಫೈ ಇದು ಜನಪ್ರೀಯ ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕಿಂಗ್ ತಂತ್ರಜ್ಞಾನವಾಗಿದೆ. ವೈಫೈ ಎಂದರೆ ವಯರ್‌ಲೆಸ್ ಫೆಡಿಲಿಟಿ. 1991 ರಲ್ಲಿ ನೆದರ್ಲ್ಯಾಂಡ್‌ನ ಎನ್ ಸಿ ಆರ್ ಕಾರ್ಪೋರೇಷನ್ / ಎಟಿ ಮತ್ತು ಟಿ ಇವರಿಂದ ವೈಫೈ ಅನ್ನು ಕಂಡುಹಿಡಿಯಲಾಯಿತು. ಈ ತಂತ್ರಜ್ಞಾನವನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ನಾವು ಎರಡು ಅಥವಾ ಅದಕ್ಕಿಂತ ಹೆಚ್ಚು ಸಾಧನಗಳ ನಡುವೆ ಮಾಹಿತಿಯನ್ನು ವಿನಿಮಯ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ಮೊಬೈಲ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಾಧನಗಳಾದ ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳಂತಹ ಸಾಧನಗಳಿಗೆ ವೈಫೈ ಅನ್ನು ಅಭಿವೃದ್ಧಿ ಪಡಿಸಲಾಗಿದೆ. ಆದರೆ ಅದನ್ನು ಈಗ ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಶನ್‌ಗಳಿಗಾಗಿ ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿದೆ. ಮತ್ತು ಗ್ರಾಹಕ ಇಲೆಕ್ಟ್ರಾನಿಕ್ಸ್‌ಗಳಾದ ಟೆಲಿವಿಷನ್, ಡಿವಿಡಿ ಪ್ಲೇಯರ್ ಮತ್ತು ಡಿಜಿಟಲ್ ಕ್ಯಾಮರಾಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ವೈಫೈ ಸಂಪರ್ಕದೊಂದಿದೆ ಸಂವಹನ ನಡೆಸಲು ಎರಡು ಸಾಧ್ಯತೆಗಳಿರಬಹುದು. ಒಂದು ಎಕ್ಸೆಸ್ ಪಾಯಿಂಟನಿಂದ ಕ್ಲೈಂಟ್ ಸಂಪರ್ಕದೊಂದಿಗೆ ವೈಫೈ ಸಂಪರ್ಕ ಹೊಂದುವುದು . ಇನ್ನೊಂದು ಕ್ಲೈಂಟ್ ನಿಂದ ಕ್ಲೈಂಟ್ ವೈಫೈ ಸಂಪರ್ಕ ಹೊಂದುವುದು. ವೈಫೈ ಒಂದು ರೀತಿಯ ವಯರ್‌ಲೆಸ್ ತಂತ್ರಜ್ಞಾನವಾಗಿದೆ. ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ವಯರ್‌ಲೆಸ್ ಲ್ಯಾನ್ (ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್) ಎಂದು ಕರೆಯುತ್ತಾರೆ. ವೈಫೈ ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕನ್ನು ಕೇಬಲ್ ಮತ್ತು ವೈರಿಂಗ್ ಇಲ್ಲದೇ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಮನೆ ಮತ್ತು ವ್ಯಾಪಾರ ನೆಟ್‌ವರ್ಕ್‌ಗಳಿಗೆ ಇದು ಜನಪ್ರೀಯ ಆಯ್ಕೆಯಾಗಿದೆ. ಕಂಪ್ಯೂಟರ್‌ನ ವಯರ್‌ಲೆಸ್ ಅಡಾಪ್ಟರ್ , ಡೇಟಾಅನ್ನು ರೇಡಿಯೋ ಸಿಗ್ನಲ್‌ಗಳಿಗೆ ವರ್ಗಾಯಿಸುತ್ತದೆ ಮತ್ತು ಡೇಟಾವನ್ನು ಬಳಕೆದಾರರಿಗಾಗಿ ಎಂಟೆನಾಕ್ಕೆ ವರ್ಗಾಯಿಸುತ್ತದೆ. (ಚಿತ್ರ 2)

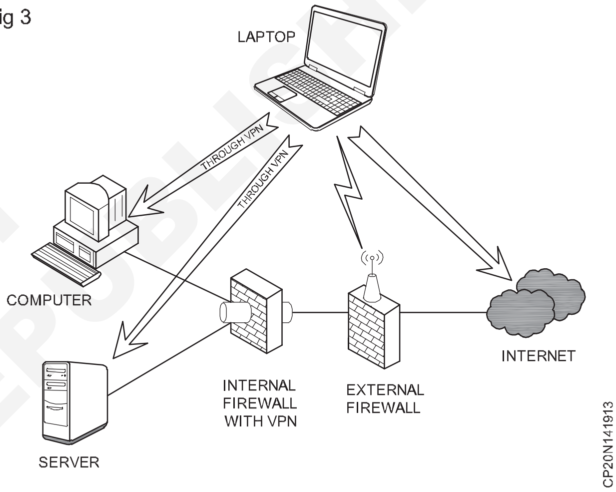
Fig 2

ಕೆಲಸದ ತತ್ವ (Working Principle) :

ವೈಫೈ ಇದು ಯಾವುದೇ ಕೇಬಲ್ ಮತ್ತು ವೈರ್ ಬಳಸದೇ ಇರುವ ಹೆಚ್ಚಿನ ವೇಗದ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಂಪರ್ಕ ಆಗಿದೆ. ವೈರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್ ಮೂರು ಅಗತ್ಯ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಅವು ರೇಡಿಯೋ ಸಿಗ್ನಲ್, ಎಂಟೆನಾ ಮತ್ತು ರೂಟರ್. ರೇಡಿಯೋ ಅಲೆಗಳು ವೈಫೈ ನೆಟ್‌ವರ್ಕಿಂಗನ್ನು ಸಾಧ್ಯವಾಗಿಸುವ ಕೀಲಿಗಳಾಗಿವೆ. ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮತ್ತು ಸೆಲ್ ಫೋನ್‌ಗಳು ವೈಫೈ ಕಾರ್ಡುಗಳೊಂದಿಗೆ ಸಿದ್ಧವಾಗಿರುತ್ತವೆ. ವೈಫೈ ಹೊಂದಾಣಿಕೆ, ಕಮ್ಯೂನಿಟಿ ನೆಟ್‌ವರ್ಕ್‌ನೊಂದಿಗೆ ಘಟಕದ ಹೊಸ ರಚನೆಯನ್ನು ಬಳಸುತ್ತಿದೆ. ನಿಜವಾದ ಪ್ರಸಾರವು ಅನುಕ್ರಮವಾಗಿ ಸಂಪರ್ಕಿಸಲಾಗಿದೆ. ಇದು ಸ್ಟೀರಿಯೋ ಸಿಸ್ಟಮ್ ಸರ್ಫ್ ಮತ್ತು ವರ್ಗೀಕರಣಕ್ಕಾಗಿ ವೈರ್ ಹೊಂದಿರುವ ಮಾನಿಟರ್ ಮೂಲಕ ಪೂರ್ಣಗೊಳ್ಳುತ್ತದೆ. ವೈಫೈ ವ್ಯಕ್ತಿಗೆ ನಿಗದಿತ ಪ್ರದೇಶದ ಯಾವುದೇ ಸ್ಥಳದಲ್ಲಿ ವೆಬ್‌ಅನ್ನು ಉಪಯೋಗಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಈಗ ರೆಸಾರ್ಟಗಳು, ಗ್ರಂಥಾಲಯಗಳು, ಶಾಲೆಗಳು, ಕಾಲೇಜುಗಳು, ಕ್ಯಾಂಪಸ್, ವೈಯಕ್ತಿಕ ಸಂಸ್ಥೆಗಳು, ಎಸ್ಪ್ರೆಸ್ಸೊ ಮಳಿಗೆಗಳು, ಮತ್ತು ಮುಕ್ತ ಸಾರ್ವಜನಿಕ ಸ್ಥಳಗಳಲ್ಲಿ ಈ ವ್ಯವಸ್ಥೆಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಕಂಪನಿಯನ್ನು ಹೆಚ್ಚು ಲಾಭದಾಯಕವಾಗಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ ಮತ್ತು ತಮ್ಮ ಸ್ವಂತ ಗ್ರಾಹಕರೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ರೇಡಿಯೋ ಸಂಕೇತಗಳನ್ನು ಎಂಟೇನಾ ಮತ್ತು ರೂಟರ್‌ಗಳಿಂದ ಪ್ರಸಾರ ಮಾಡಲಾಗುತ್ತದೆ. ಈ ಸಂಕೇತಗಳನ್ನು ವೈಫೈ ರಿಸೀವರ್‌ಗಳಾದ ಮತ್ತು ವೈಫೈ ಕಾರ್ಡುಗಳನ್ನು ಹೊಂದಿರುವ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮತ್ತು ಸೆಲ್ ಫೋನ್ ಗಳು ತೆಗೆದುಕೊಳ್ಳುತ್ತವೆ. ಕಂಪ್ಯೂಟರ್‌ಗಳು 100 -150 ಫೂಟು ವ್ಯಾಪ್ತಿಯಲ್ಲಿ ಬರುವ ಸಂಕೇತಗಳನ್ನು ಸ್ವೀಕರಿಸಿ ಅದು ತಕ್ಷಣ ಸಾಧನವನ್ನು ಸಂಪರ್ಕಿಸುತ್ತದೆ. ವೈಫೈ ವ್ಯಾಪ್ತಿಯು ಪರಿಸರ, ಒಳಾಂಗಣ ಅಥವಾ ಹೊರಾಂಗಣ ವ್ಯಾಪ್ತಿಯನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ. ವೈಫೈ ಕಾರ್ಡ್‌ಗಳು ಸಂಕೇತಗಳನ್ನು ಓದುತ್ತವೆ ಮತ್ತು ಬಳಕೆದಾರ ಮತ್ತು ನೆಟ್ ವರ್ಕ್ ನಡುವೆ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ಸೃಷ್ಟಿಸುತ್ತವೆ. ವೈಫೈ ಸಂಪರ್ಕವನ್ನು ಬಳಸುವ ಸಾಧನದ ವೇಗ ಅದು ಮೂಲ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಹತ್ತಿರವಿದ್ದಾಗ ಹೆಚ್ಚಿರುತ್ತದೆ ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ದೂರವಾದಾಗ ವೇಗ ಕಡಿಮೆಯಾಗುತ್ತದೆ. (ಚಿತ್ರ 3)

Fig 3



ಭದ್ರತೆ (Security) :

ವೈಫೈ ತಂತ್ರಜ್ಞಾನದಲ್ಲಿ ಸುರಕ್ಷತೆಯು ಪ್ರಮುಖ ಅಂಶವಾಗಿದೆ. ಭದ್ರತೆಯು ನಮ್ಮ ವೈಯಕ್ತಿಕ ನಿರ್ಧಾರವಾಗಿದೆ ಆದರೆ ವೈರ್‌ಲೆಸ್ ಸಂಪರ್ಕವನ್ನು ಹೊಂದುವುದರಿಂದ ನಾವು ವೈಯಕ್ತಿಕ ವಿವರಗಳನ್ನು ರಕ್ಷಿಸುವ ಬಗ್ಗೆ ಗಮನ ಕೊಡಬೇಕು. ನಾವು ಸುಲಭವಾಗಿ ಅಸುರಕ್ಷಿತ ವೈರ್‌ಲೆಸ್ ರೂಟರ್‌ನ್ನು ಸಂಪರ್ಕಿಸಬಹುದು. ಸಮಸ್ಯೆ ಏನೆಂದರೆ ಯಾರಾದರೂ ಡೌನ್ಲೋಡ್ ಆಟಗಳು , ಅಪ್ಲಿಕೇಶನ್‌ಗಳು, ಮತ್ತು ಭಯೋತ್ಪಾದಕ ಚಟುವಟಿಕೆಗಳು, ಕಾನೂನುಬಾಹಿರ ಸಂಗೀತ ಮತ್ತು ಚಲನಚಿತ್ರ ಫೈಲ್ ಗಳು ಮುಂತಾದ ಡೇಟಾ ಉಪಯೋಗಿಸಿ ವೈರ್‌ಲೆಸ್ ರೂಟರ್‌ಗೆ ಸಂಪರ್ಕಿಸಬಹುದಾಗಿದೆ. ಅದ್ದರಿಂದ ವಯರ್‌ಲೆಸ್ ತಂತ್ರಜ್ಞಾನ ಆಧಾರಿತ ಸಾಧನಗಳಿಗೆ ಭದ್ರತೆಯನ್ನು ಒದಗಿಸುವುದು ಅಗತ್ಯವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಡಿವಿಡಿಗಳು, ಸಿಡಿಗಳು ಮತ್ತು ಡಿವಿಡಿಗಳಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಬರೆಯುವುದು (DVDs, CDs and burning DVDs)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಸಿಡಿಗಳು ಮತ್ತು ಡಿವಿಡಿಗಳ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ವಿವರಿಸಿ
- ಡಿವಿಡಿಗಳ ಮುಖ್ಯ ಸ್ವರೂಪಗಳನ್ನು ವಿವರಿಸಿ
- ಡಿವಿಡಿ ಬರ್ನಿಂಗನ್ನು ವಿವರಿಸಿ
- ಸಿಡಿಗಳ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ವಿವರಿಸಿ.

ಪರಿಚಯ (Introduction)

ಡಿವಿಡಿ (ಕೆಲವೊಮ್ಮೆ ಇದನ್ನು ಡಿಜಿಟಲ್ ವಿಡಿಯೋ ಡಿಸ್ಕ ಅಥವಾ ಡಿಜಿಟಲ್ ವರ್ಸಟೈಲ್ ಡಿಸ್ಕ ಎಂದು ಕರೆಯುತ್ತಾರೆ) ಇದು ಡಿಜಿಟಲ್ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ್ ಶೇಖರಣಾ ಸ್ವರೂಪವಾಗಿದೆ. ಡಿವಿಡಿಗಳನ್ನು ಪಿಸಿಗಳೊಂದಿಗೆ ಮತ್ತು ಸ್ವತಂತ್ರ ಪ್ಲೇಯರ್‌ಗಳಂತಹ ಹಲವು ರೀತಿಯ ಪ್ಲೇಯರ್ ಗಳೊಂದಿಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಈ ಡಿಸ್ಕಗಳನ್ನು ಡಿವಿಡಿ-ರೋಮ್ (DVD - ROM) ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಏಕೆಂದರೆ ಇದರಲ್ಲಿ ಡೇಟಾವನ್ನು ಓದಬಹುದು ಮಾತ್ರ , ಬರೆಯಲು ಅಥವಾ ಅಳಿಸಲು ಬರುವುದಿಲ್ಲ. ಖಾಲಿ ರೆಕಾರ್ಡ ಮಾಡಲು ಬರುವ ಡಿವಿಡಿ ಡಿಸ್ಕಗಳನ್ನು (ಡಿವಿಡಿ-ಆರ್ ಮತ್ತು ಡಿವಿಡಿ+ಆರ್) ಡಿವಿಡಿ ರೆಕಾರ್ಡರನ್ನು ಬಳಸಿ ರೆಕಾರ್ಡ ಮಾಡಬಹುದಾಗಿದೆ. ಮತ್ತು ನಂತರ ಇವುಗಳು ಡಿವಿಡಿ-ರೋಮ್ ತರಹ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ. ಪುನಃ ಬರೆಯಬಹುದಾದ ಡಿವಿಡಿಗಳನ್ನು (ಡಿವಿಡಿ-ಆರ್‌ಡಬ್ಲು , ಡಿವಿಡಿ+ಆರ್ ಡಬ್ಲು ಮತ್ತು ಡಿವಿಡಿ-ರ್‍ಯಾಮ್)ಅನೇಕ ಬಾರಿ ರೆಕಾರ್ಡ ಮಾಡಬಹುದು ಮತ್ತು ಅಳಿಸಬಹುದು.

ಡಿವಿಡಿ ವೈಶಿಷ್ಟ್ಯಗಳು ಮತ್ತು ಸ್ವರೂಪಗಳು (DVD features and formats)

ಡಿವಿಡಿಗಳನ್ನು ಡಿವಿಡಿ-ವಿಡಿಯೋ ಕಂಜ್ಯುಮರ್ ಡಿಜಿಟಲ್ ವಿಡಿಯೋ ಫಾರ್ಮ್ಯಾಟ್ ನಲ್ಲಿ ಮತ್ತು ಡಿವಿಡಿ ಆಡಿಯೋ ಕಂಜ್ಯುಮರ್ ಡಿಜಿಟಲ್ ಆಡಿಯೋ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಬಳಸಲಾಗುತ್ತದೆ. ಇವುಗಳನ್ನು ವಿಶೇಷ AVCHD (Advanced Video Coding High Definition) ಸ್ವರೂಪದಲ್ಲಿಯೂ ಬಳಸಬಹುದು. AVCHD ಫಾರ್ಮ್ಯಾಟ್‌ನ ಕ್ಯಾಮ್ ಕಾರ್ಡರ್‌ಗಳಲ್ಲಿ ಹೆಚ್ಚಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ. ಬೇರೆ ರೀತಿಯ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿದ ಡಿವಿಡಿಗಳನ್ನು ಡಿವಿಡಿ ಡಿಸ್ಕ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ರೆಕಾರ್ಡ ಮಾಡಬಹುದಾದ ಡಿವಿಡಿಯು ಸುಮಾರು 4.7 ಗಿಗಾಬೈಟ್(ಜಿಬಿ) ಗಳಷ್ಟು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು. ಅದಾಗ್ಯೂ ನೀವು ಡಿಸ್ಕನಲ್ಲಿ ಫೈಲ್‌ಗಳನ್ನು ಬರ್ನ್ ಮಾಡಲು ಉಪಯೋಗಿಸುವ ಒಟ್ಟೂ ಡಿಸ್ಕನ ಸ್ಥಳ ಡಿಸ್ಕ ಮೇಲೆ ಪಟ್ಟಿ ಮಾಡಿದ ಸ್ಥಳದ ಪ್ರಮಾಣಕ್ಕಿಂತ ಕಡಿಮೆಯಾಗಿರುತ್ತದೆ. ಇದಕ್ಕೆ ಕಾರಣ ಇದನ್ನು ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಿದಾಗ ಡಿಸ್ಕ ಸಾಮರ್ಥ್ಯವನ್ನು ವಿಭಿನ್ನವಾಗಿ ಲೆಕ್ಕ ಮಾಡಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ 4.7 ಜಿಬಿ ಡಿಸ್ಕ ಸಾಮರ್ಥ್ಯ ಇದೆ ಎಂದು ಪಟ್ಟಿ ಮಾಡಲಾದ ಒಂದು ನಿರ್ದಿಷ್ಟ ಡಿವಿಡಿ-ಆರ್‌ನಲ್ಲಿ ಕೇವಲ 4.37 ಜಿಬಿ ನಷ್ಟು ಡೇಟಾ ಮಾತ್ರ ಸಂಗ್ರಹಿಸಬಹುದು. ಡಿವಿಡಿಗಳು ಸರಿಸುಮಾರು 4.7 ಜಿಬಿ ನಷ್ಟು ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಡಿವಿಡಿ ಡಿಸ್ಕಗಳು ಸಮಯದೊಂದಿಗೆ ಹಾಳಾಗುವುದಿಲ್ಲ ಮತ್ತು ಕಾಂತೀಯ ಕ್ಷೇತ್ರಗಳಿಂದ ಪ್ರಭಾವಿತವಾಗುವುದಿಲ್ಲ.

ಬಳಸಬೇಕಾದ , ರೆಕಾರ್ಡ ಮಾಡಬಹುದಾದ ಡಿಸ್ಕಗಳ ಪ್ರಕಾರಗಳು ಕೆಲವು ವಿಭಿನ್ನ ಅಂಶಗಳನ್ನು ಅವಲಂಬಿಸಿರುತ್ತವೆ. , ಉದಾಹರಣೆಗೆ :

- ಡಿಸ್ಕ ಬರ್ನರ್‌ನೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಬಲ್ಲ ರೆಕಾರ್ಡ ಮಾಡಬಹುದಾದ ಡಿಸ್ಕನ ವಿಧಗಳು
- ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಡಿಸ್ಕ ಡ್ರೈವ್ ಅಥವಾ ಬರ್ನ್ ಆದ ನಂತರ ಡಿಸ್ಕನ್ನು ಸಾಧನವು ಓದುತ್ತದೆ.
- ಡಿಸ್ಕನಲ್ಲಿ ಬರ್ನ್ ಆಗಬೇಕಾದ ಎಲ್ಲ ಫೈಲ್‌ಗಳ ಒಟ್ಟೂ ಗಾತ್ರ

ಡಿವಿಡಿ ಬರ್ನಿಂಗ್ (DVD burning)

ಮಾಹಿತಿಯನ್ನು ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ(ಸಿಡಿ ಅಥವಾ ಡಿವಿಡಿ) ನಲ್ಲಿ ರೆಕಾರ್ಡ ಮಾಡುವ ಪ್ರಕ್ರಿಯೆಗೆ ಬರ್ನಿಂಗ್/ ರೈಟಿಂಗ್ ಅಥವಾ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ ಆಥರಿಂಗ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕನ್ನು ತಯಾರಿಸುವುದಕ್ಕಾಗಿ ಸಾಮಾನ್ಯವಾಗಿ ಮೊದಲು ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ ಸಲುವಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಿದ ಪೂರ್ಣ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನ ಡಿಸ್ಕ ಇಮೇಜನ್ನು ತಯಾರಿಸುವುದು ಮತ್ತು ನಂತರ ಇಮೇಜನ್ನು ನಿಜವಾಗಿ ಡಿಸ್ಕನಲ್ಲಿ ಬರ್ನ್ ಮಾಡುವುದು ಆಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಅನೇಕ ಪ್ರೋಗ್ರಾಂಗಳು, ಡಿಸ್ಕ ಇಮೇಜನ್ನು ತಯಾರಿಸಲು ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ಬರ್ನ್ ಮಾಡಲು , ಅಪ್ಲಿಕೇಶನ್‌ಗಳ ಗುಂಪುಗಳಂತೆ ಲಭ್ಯ ಇವೆ. ಡಿವಿಡಿಗಳಲ್ಲಿ ಬರೆಯಬಹುದಾದ ವೇಗವನ್ನು ಮಲ್ಟಿಪ್ಲಾಯರ್‌ನಲ್ಲಿ ವ್ಯಕ್ತಪಡಿಸಲಾಗುತ್ತದೆ : 16X ಎಂದರೆ ಅದನ್ನು ಉಪಯೋಗಿಸುವದಕ್ಕಿಂತ 16 ಪಟ್ಟು ವೇಗವಾಗಿರುತ್ತದೆ. 52X ವೇಗವು ಸಹ ತುಂಬಾ ಸಾಮಾನ್ಯವಾದ ವೇಗವಾಗಿದೆ.

ಸಿಡಿ ಅಥವಾ ಡಿವಿಡಿ ಸ್ವರೂಪಗಳು (CD or DVD formats)

ಡಿವಿಡಿಗಳನ್ನು ಬರ್ನ್ ಮಾಡಲು , ಎರಡು ಮುಖ್ಯ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ಗಳು ಬಳಕೆಯಲ್ಲಿವೆ :

- ಲೈವ್ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಮತ್ತು (Live file system)
- ಮಾಸ್ಟರ್ಡ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ (Mastered disk format)

ಲೈವ್ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟ್ (Live file system format) :

ಲೈವ್ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಡಿಸ್ಕಗಳು ಹೆಚ್ಚು ಅನುಕೂಲವಾಗಿರುತ್ತವೆ ಏಕೆಂದರೆ ಯುಎಸ್ ಬಿ ಫ್ಲ್ಯಾಷ ಡ್ರೈವ್‌ನಂತೆ ಆಗಾಗ್ಗೆ ನಿಮಗೆ ಬೇಕಾದ ಫೈಲ್ಗಳನ್ನು ,ನೀವು ಆಯ್ದ ಫೈಲ್‌ಗಳನ್ನು ತಕ್ಷಣ ಕಾಪಿ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ನೀವು ನಿಮ್ಮ ಡಿಸ್ಕನ್ನು ಸಿಡಿ, ಡಿವಿಡಿ ಅಥವಾ ಬ್ಲೂ ರೇ ಡಿಸ್ಕ ಬರ್ನರ್‌ನಲ್ಲಿ ಇಡಲು ಬಯಸಿದರೆ ಮತ್ತು ನಿಮಗೆ ಅಗತ್ಯವಿದ್ದಾಗ ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡಲು ಬಯಸಿದರೆ ಇದು ಅನುಕೂಲಕರವಾಗಿದೆ. ಈ ಫಾರ್ಮ್ಯಾಟ್ ನಲ್ಲಿ ನೀವು ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡುವುದು ಮತ್ತು ಅಳಿಸುವುದನ್ನು ಮೇಲಿಂದ ಮೇಲೆ ಮತ್ತೆ ಮತ್ತೆ ಮಾಡಬಹುದು. ಅದಾಗ್ಯೂ ಲೈವ್ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ ಇದನ್ನು ಕೇವಲ ವಿಂಡೋಸ್ 7, ವಿಂಡೊಸ್ ವಿಸ್ತಾ ಮತ್ತು ವಿಂಡೊಸ್ ಎಕ್ಸ್ ಪಿ ಸಿಸ್ಟಮ್ ಗಳಲ್ಲಿ ಮಾತ್ರ ಓದಬಹುದು. ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿದ ನಂತರ ಈ ಡಿಸ್ಕಗಳು ಖಾಲಿಯಾಗಿರುವುದಿಲ್ಲ.

ಈ ರೀತಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲಾದ ಡಿಸ್ಕಗಳು ಬರ್ನ್ ಡಿಸ್ಕ ಡಾಯಲಾಗ್ ಬಾಕ್ಸ್ ನಲ್ಲಿ ಈ ರೀತಿ ಆಯ್ಕೆಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ``Like a USB flash drive``

ಮಾಸ್ಟರ್ಡ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ (Mastered disk format) :

ನಾವು ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕನ್ನು ತಯಾರಿಸಿ ಡೇಟಾ ಫೈಲ್‌ಗಳನ್ನು ವಿಂಡೊಸ್ ಆಪರೇಟಿಂಗ ಸಿಸ್ಟಮ್‌ನ ಹಳೆಯ ಆವೃತ್ತಿಗಳಿಗೆ ವರ್ಗಾಯಿಸಲು ಅಥವಾ ಬೇರೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗೆ ವರ್ಗಾಯಿಸಲು ಬಯಸಿದರೆ ಇದಕ್ಕಾಗಿ ಮಾಸ್ಟರ್ಡ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ ಉಪಯೊಗಿಸಬಹುದು. ನಾವು ಸಂಗೀತವನ್ನು ಅಥವಾ ಚಿತ್ರಗಳನ್ನು ಬರ್ನ್ ಮಾಡಲು ಬಯಸಿದರೆ ಮತ್ತು ಡಿಸ್ಕನ್ನು MP3 ಫೈಲ್‌ಗಳನ್ನು ಪ್ಲೇ ಮಾಡುವ ಮತ್ತು ಡಿಜಿಟಲ್ ಚಿತ್ರಗಳನ್ನು ತೋರಿಸುವ ನಿಯಮಿತ ಸಿಡಿ, ಡಿವಿಡಿ ಅಥವಾ ಬ್ಲೂ ರೇಡಿಸ್ಕಗಳಲ್ಲಿ ಬಳಸಿದರೆ ಅದಕ್ಕಾಗಿ ಮಾಸ್ಟರ್ಡ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್‌ನ್ನು ಉಪಯೊಗಿಸಬೇಕು.

ಮಾಸ್ಟರ್ಡ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ ಕೂಡ ವಿಂಡೋಸ್ ಎಕ್ಸ್ ಪಿನಲ್ಲಿ ಸಿಡಿ ಬರ್ನ್ ಮಾಡುವಂತೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಬೇರೆ ಶಬ್ದಗಳಲ್ಲಿ ಹೇಳುವುದಾದರೆ ನಾವು ಡಿಸ್ಕ ಬರೆಯುವಾಗ ನಾವು ಫೈಲ್ ಅಥವಾ ಫೈಲ್ ಗಳ ಗುಂಪನ್ನು ಒಮ್ಮೆ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕಗೆ ಕಾಪಿ ಮಾಡುತ್ತೇವೆ. ಇದನ್ನು ಮಾಡಿದ ನಂತರ, ಮತ್ತು ಡಿಸ್ಕನಲ್ಲಿ ಹೆಚ್ಚಿನ ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ ಮತ್ತು ಈಗಾಗಲೆ ಅದರಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಅಳಿಸಲು ಬರುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ತಕ್ಷಣ ಫೈಲ್ ಗಳನ್ನು ಕಾಪಿ ಮಾಡದಂತೆ ಸೂಚಿಸಲಾಗಿದೆ. ಡಿಸ್ಕಗೆ ಕಾಪಿ ಮಾಡಬೇಕಾದ ಎಲ್ಲ ಫೈಲ್‌ಗಳ ಸಂಗ್ರಹವನ್ನು ಕೂಡಿಸಿಕೊಂಡು ನಂತರ ಎಲ್ಲವನ್ನೂ ಒಂದೇ ಬಾರಿಗೆ ಬರ್ನ್ ಮಾಡುವುದು ಒಳ್ಳೆಯ ಅಭ್ಯಾಸವಾಗಿದೆ.

ಮಾಸ್ಟರ್ಡ ಆಯ್ಕೆಯೊಂದಿಗೆ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲಾದ ಡಿಸ್ಕಗಳು ಬರ್ನ್ ಡಿಸ್ಕ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಈ ಆಯ್ಕೆಯನ್ನು ಹೊಂದಿರುತ್ತವೆ : "With a CD/DVD player"

ಡೇಟಾ, ಸಂಗೀತ, ವಿಡೀಯೊ, ಮತ್ತು ಆಡಿಯೋ ಡಿಸ್ಕ ಗಳನ್ನು ತಯಾರಿಸಲು ಹಲವು ರೀತಿಯ ಸಾಧನಗಳು ಲಭ್ಯ ಇವೆ.

ಬಹಳಷ್ಟು ಡಿಸ್ಕಗಳಲ್ಲಿ ವ್ಯಾಪಿಸಿರುವ ಬ್ಯಾಕಪ್‌ಗಳನ್ನು ಸಹ ತಯಾರಿಸಬಹುದು. ಆಡಿಯೋ ಸಿಡಿಗಳಿಂದ ಸಂಗೀತದ ಟ್ರ್ಯಾಕ್‌ಗಳನ್ನು ಬೇರ್ಪಡಿಸುವುದು, ಡಿಸ್ಕ ಇಮೇಜಗಳನ್ನು ಬೇರೆ ಬೇರೆ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳಲ್ಲಿ ತಯಾರಿಸುವುದು ಅಥವಾ ಬರ್ನ್ ಮಾಡಬಹುದು. ಇವು WAV,MP3,FLAC,WMA ಫೈಲ್‌ಗಳಿಂದ ಸ್ವಯಂಚಾಲಿತ ಆಡಿಯೋ ಪರಿವರ್ತನೆ, ಡಿಸ್ಕ ಕಾಪಿ ಮಾಡುವುದು, ಸಂಕುಚಿತ (compressed) ಫೈಲ್ ಬ್ಯಾಕಪ್ ಮತ್ತು ರಿಸ್ಟೋರ್, ಡಿಸ್ಕ ಅಳಿಸುವಿಕೆ, VCD/SVCD ಬೆಂಬಲ, ಪ್ರಾಜೆಕ್ಟ ಬರ್ನಿಂಗ್ ಮುಂತಾದ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸಹ ಒದಗಿಸಬಹುದು.

ಬ್ಲೂ ರೇ ಡಿಸ್ಕಗಳು (Blu-ray Disks)

ಬ್ಲೂ-ರೇ, ಇದನ್ನು ಬ್ಲೂ-ರೇ ಡಿಸ್ಕ್ (BD) ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಡಿವಿಡಿಯನ್ನು ವೇಗವಾಗಿ ಬದಲಾಯಿಸುತ್ತಿರುವ ಹೊಸ ಆಪ್ಟಿಕಲ್ ಡಿಸ್ಕ ಫಾರ್ಮ್ಯಾಟ್ ಆಗಿದೆ.ಈ ಫಾರ್ಮ್ಯಾಟನ್ನು ಹೈ ಡೆಫಿನಿಷನ್ (HD) ವಿಡಿಯೋಗಳನ್ನು ರೆಕಾರ್ಡಿಂಗ್, ಪುನಃಬರೆಯುವಿಕೆ ಮತ್ತು ಪ್ಲೇಬ್ಯಾಕ್ ಮಾಡುವುದಕ್ಕಾಗಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ. ಇದನ್ನು ದೊ ಡ್ಡ ಪ್ರಮಾಣದ ಡೇಟಾವನ್ನು ಸಂಗ್ರಹಿಸುವುದಕ್ಕಾಗಿ ಸಹ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇವು ಸಾಂಪ್ರದಾಯಿಕ ಡಿವಿಡಿಗಳ ಸಂಗ್ರಹಣಾ ಸಾಮರ್ಥ್ಯಕ್ಕಿಂತ ಐದು ಪಟ್ಟು ಹೆಚ್ಚು ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಒಂದೇ ಪದರದ ಡಿಸ್ಕಗಳು 25ಜಿಬಿ ಗಳಷ್ಟು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲವು ಮತ್ತು ಎರಡು ಪದರದ ಡಿಸ್ಕಗಳು 50 ಜಿಬಿಯಷ್ಟು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲವು.

© NIMI NOT TO BE REPUBLISHED

ಡೇಟಾವನ್ನು ಓದಲು ಮತ್ತು ಬರೆಯಲು ನೀಲಿ ನೇರಳೆ ಲೇಸರ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ತಂತ್ರಜ್ಞಾನವನ್ನು ಉಪಯೋಗಿಸುವದರಿಂದ ಇದಕ್ಕೆ ಈ ಹೆಸರು ಬಂದಿದೆ. ಈ ಹೆಸರು ಬ್ಲೂ (ಬ್ಲೂ-ವೈಯೊಲೆಟ್ ಲೇಸರ್) ಮತ್ತು ರೇ (ಆಪ್ಟಿಕಲ್ ರೇ) ಇವುಗಳ ಸಂಯೋಜನೆಯಿಂದ ಬಂದಿದೆ. ಇವುಗಳನ್ನು ಬ್ಲೂ ರೇ ಅಥವಾ BDs ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ಈ ಕೆಳಗಿನ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳು ಇವತ್ತಿನ ಬ್ಲೂ ರೇ ಡಿಸ್ಕನ ವಿವರಣೆಯನ್ನು ಹೊಂದಿವೆ ;

1 ಬಿಡಿ - ರಾಮ್ - ಎಚ್‌ಡಿ ಚಲನಚಿತ್ರಗಳು, ಆಟಗಳು,ಸಾಫ್ಟವೆರ್‌ಗಳು ಮುಂತಾದವುಗಳನ್ನು ಓದುವ ಮಾತ್ರ ಸ್ವರೂಪದ ವಿತರಣೆಗಾಗಿ

2 ಬಿಡಿ - ಆರ್ - ಎಚ್‌ಡಿ ವಿಡಿಯೋ ರೆಕಾರ್ಡಿಂಗ್ ಮತ್ತು ಪಿಸಿ ಡೇಟಾ ಸಂಗ್ರಹಣೆಗಾಗಿ ರೆಕಾರ್ಡ ಮಾಡುವ ಸ್ವರೂಪಕ್ಕಾಗಿ

3 ಬಿಡಿ - ಆರ್‌ಇ - ಎಚ್‌ಡಿ ವಿಡಿಯೋ ರೆಕಾರ್ಡಿಂಗ್ ಮತ್ತು ಪಿಸಿ ಡೇಟಾ ಸಂಗ್ರಹಣೆಗಾಗಿ ಪುನಃ ಬರೆಯುವ ಸ್ವರೂಪಕ್ಕಾಗಿ

ಪ್ರಸ್ತುತ ಒಂದು ಪದರದ ಡಿಸ್ಕ 25 ಜಿಬಿ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು ಮತ್ತು ಎರಡು ಪದರದ ಡಿಸ್ಕ 50 ಜಿಬಿ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು. 50 ಜಿಬಿ ಡಿಸ್ಕನಲ್ಲಿ 9 ಗಂಟೆಗಳ ಹೈಡೆಫಿನಿಷನ್(HD) ವಿಡಿಯೊ ಸಂಗ್ರಹಿಸಬಹುದು. ಸುಮಾರು 23 ಗಂಟೆಗಳ ಸ್ಟ್ಯಾಂಡರ್ಡ ಡೆಫಿನಿಷನ್ (SD) ವಿಡಿಯೊನ್ನು 50 ಜಿಬಿ ಡಿಸ್ಕನಲ್ಲಿ ಸಂಗ್ರಹಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಡೈರೆಕ್ಟರಿ ಪಟ್ಟಿ ಮಾಡಲು ಡಾಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Use basic DOS commands for directory listing)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- ಡಾಸ್‌ನಲ್ಲಿ ಕ್ರಮಾನುಗತ ಡೈರೆಕ್ಟರಿ ವ್ಯವಸ್ಥೆಯನ್ನು ವಿವರಿಸಿ
- ಡೈರೆಕ್ಟರಿಗಳು ಮತ್ತು ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸಲು ಡಾಸ್ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ
- ಡೈರೆಕ್ಟರಿ ಬದಲಾಯಿಸಲು ಮತ್ತು ಪಟ್ಟಿ ಮಾಡಲು ಡಾಸ್ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ
- ನಿರ್ದಿಷ್ಟ ಫೈಲ್‌ಗಳನ್ನು ಪಡೆಯಲು ಡಾಸ್ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ

ಕ್ರಮಾನುಗತ ಡೈರೆಕ್ಟರಿ ವ್ಯವಸ್ಥೆ (Hierarchical Directory System): ಕ್ರಮಾನುಗತ (hierarchy) ಎಂದರೆ , ಘಟಕಗಳ ರಚನೆ ಅಥವಾ ವ್ಯವಸ್ಥೆ. ಘಟಕಗಳು (entity) ಎಂದರೆ ವಸ್ತುಗಳು, ಫೈಲ್‌ಗಳು, ಜನರು, ಕಲ್ಪನೆಗಳು ಅಥವಾ ಇನ್ನಾವುದೇ ವಿಷಯಗಳಾಗಿರಬಹುದು.

ವ್ಯವಸ್ಥೆ (arrangement)ಎಂದರೆ ಉದಾಹರಣೆಗೆ ಹಣವನ್ನು ಅವುಗಳ ಮುಖಬೆಲೆಯ ಆಧಾರದ ಮೇಲೆ ಮತ್ತು ಪೆಬಲ್‌ಗಳನ್ನು ಅವುಗಳ ಗಾತ್ರದ ಮೇಲೆ ಜೋಡಿಸುವುದು.

ಕ್ರಮಾನುಗತದ ಹೊರತಾಗಿ ಘಟಕಗಳನ್ನು ಸಂಘಟಿಸಲು ಇನ್ನೂ ಅನೇಕ ಮಾರ್ಗಗಳಿವೆ. ಆದರೆ ಕ್ರಮಾನುಗತ ಸಂಘಟನೆ ವಿಶೇಷವಾಗಿದೆ. ಏಕೆಂದರೆ ಈ ವ್ಯವಸ್ಥೆಯಿಂದ ನೀವು ಪ್ರತಿ ಘಟಕವನ್ನು ಬೇರೆ ಇತರ ಘಟಕದೊಂದಿಗೆ ಅದರ ಸಂಬಂಧದಿಂದ ಹೆಸರಿಸಬಹುದು.

DOS ನಲ್ಲಿ, ಡೈರೆಕ್ಟರಿ ವ್ಯವಸ್ಥೆಯಲ್ಲಿ - ಎಂಟಿಟಿಗಳು ಎಂದರೆ ಡೈರೆಕ್ಟರಿಗಳು ಆಗಿರುತ್ತವೆ. ಇಲ್ಲಿ ಕ್ರಮಾನುಗತವು ಕೋರ್ ಅಥವಾ ರೂಟ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಒಂದು ಕುಟುಂಬ ವೃಕ್ಷದಲ್ಲಿ ನಮ್ಮ ಮೂಲ ಕಾರಣರಾದ ದೊಡ್ಡ ಮುತ್ತಜ್ಜರನ್ನು ಕೋರ್ ಎಂಟಿಟಿ (core entity) ಎಂದು ಪರಿಗಣಿಸುತ್ತೇವೆ. ಡಾಸ್‌ನಲ್ಲಿ ಈ ಕೋರ್ ಎಂಟಿಟಿಯನ್ನು ರೂಟ್ ಡೈರೆಕ್ಟರಿ (root directory) ಎಂದು ಪರಿಗಣಿಸುತ್ತೇವೆ.

ಮೇಲೆ ಪರಿಗಣಿಸಲಾದ ಉದಾಹರಣೆಯಂತೆ, ನಮ್ಮ ದೊಡ್ಡ ಮುತ್ತಜ್ಜ ಅವರನ್ನು ರೂಟ್ ಡೈರೆಕ್ಟರಿ ಎಂದು ಪರಿಗಣಿಸಿದರೆ , ದೊಡ್ಡ ಅಜ್ಜ , ಅಜ್ಜ, ತಂದೆ ಇವರನ್ನು ಸಬ್ ಡೈರೆಕ್ಟರಿ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ಡಾಸ್‌ನಲ್ಲಿ ಮೂಲ ಡೈರೆಕ್ಟರಿಯ ಅಡಿಯಲ್ಲಿ ಬರುವ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಸಬ್ ಡೈರೆಕ್ಟರಿ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಈ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು ತಮ್ಮ ಮಾರ್ಗವನ್ನು ಮೂಲ ಡೈರೆಕ್ಟರಿಗೆ ಹಿಂತಿರುಗಿಸಬಹುದು.

ಡಾಸ್ ಕ್ರಮಾನುಗತ ಫೈಲ್ ಸಿಸ್ಟಮನ್ನು , ಟ್ರೀ ಸ್ಟ್ರಕ್ಚರ್ಡ ಫೈಲ್ ಸಿಸ್ಟಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಈ ಟ್ರೀ ಸ್ಟ್ರಕ್ಚರ್‌ನ ಮೂಲದಲ್ಲಿ ರೂಟ್ ಡೈರೆಕ್ಟರಿ ಇರುತ್ತದೆ.

ಕುಟುಂಬ ವೃಕ್ಷದಲ್ಲಿ , ಗೋವಿಂದ ರಾಜಪ್ಪನ ಮಗ , ರಾಜಪ್ಪಾ ರಾಮಪ್ಪನ ಮಗ , ರಾಮಪ್ಪ ವೆಂಕಪ್ಪನ ಮಗ . ವೆಂಕಪ್ಪ ಗೋವಿಂದನ ಕುಟುಂಬ ವೃಕ್ಷದ ಮುಖ್ಯಸ್ಥರು ಅಥವಾ ಮೂಲವಾಗಿದ್ದಾರೆ.

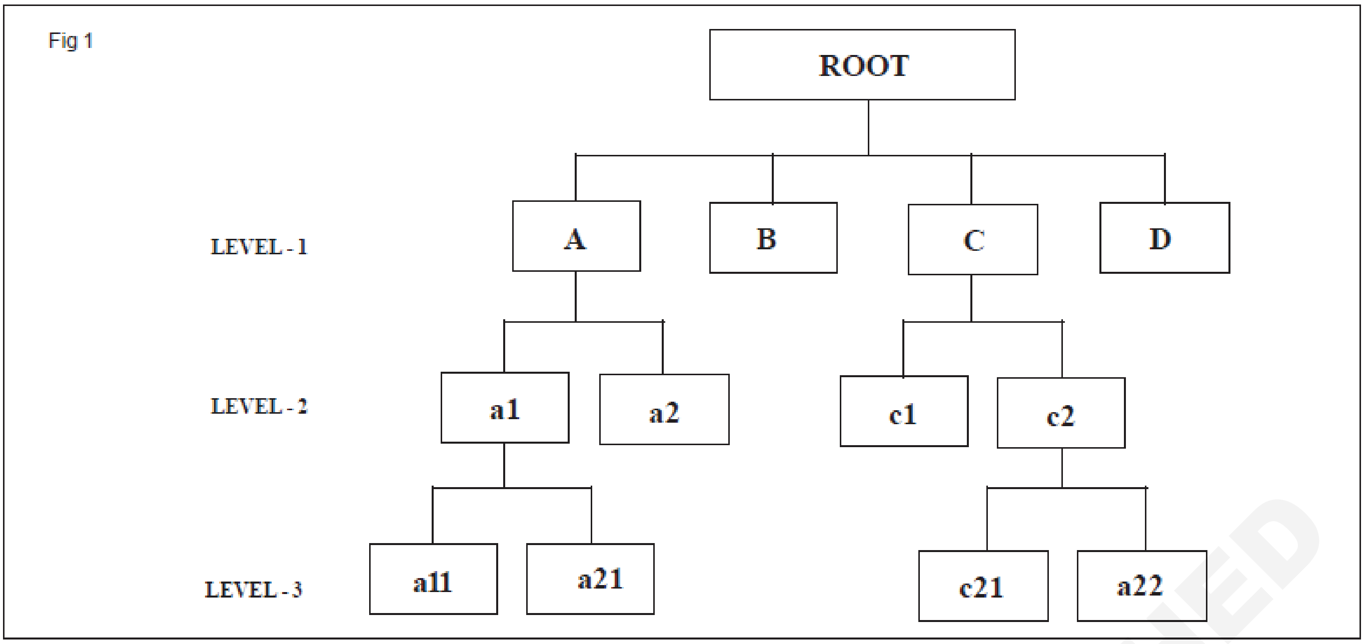
ಒಂದು ಮೂಲ ಅಥವಾ ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಅನೇಕ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತಯಾರಿಸಬಹುದು. ರೂಟ್ ನಂತರ ಈ ಪ್ರತಿ ಡೈರೆಕ್ಟರಿಯ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿ (parent directory) ಆಗುತ್ತದೆ. ನೀವು ಇತರ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳಿಂದ ಮತ್ತೆ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತಯಾರಿಸಬಹುದು ಮತ್ತು ಈ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳಿಂದ ಮತ್ತೆ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತಯಾರಿಸಬಹುದು. ಹೊಸ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು ಯಾವ ಸಬ್ ಡೈರೆಕ್ಟರಿಯ ಅಡಿಯಲ್ಲಿ ತಯಾರಾಗಿರುತ್ತದೆಯೊ ಆ ಡೈರೆಕ್ಟರಿಗೆ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಯಾವ ರೀತಿ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಕ್ರಮಾನುಗತವಾಗಿ ಜೋಡಿಸಲ್ಪಟ್ಟಿರುವುದನ್ನು ಚಿತ್ರ 1 ರಲ್ಲಿ ವಿವರಿಸಲಾಗಿದೆ. ಡಾಸ್ ಡೈರೆಕ್ಟರಿ ಸಿಸ್ಟಮನ್ನು ಹೆಚ್ಚಾಗಿ ಟ್ರೀ ಸ್ಟ್ರಕ್ಚರ್ಡ ಡೈರೆಕ್ಟರಿ ಸಿಸ್ಟಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಚಿತ್ರ 1 ರಲ್ಲಿ ಮೂರು ಹಂತದ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ನಿರೂಪಿಸಲಾಗಿದೆ. ಹಂತಗಳ ಸಂಖ್ಯೆಯ ಹೊರತಾಗಿ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳ ನಡುವಿನ ಸಂಬಂಧವು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ. ಪ್ರತಿ ಸಬ್ ಡೈರೆಕ್ಟರಿ ಮತ್ತು ರೂಟ್ ಡೈರೆಕ್ಟರಿಯು ಬಳಕೆದಾರರ ಫೈಲ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರಬಹುದು. ಎರಡು ಫೈಲ್‌ಗಳು ಒಂದೇ ಫೈಲ್ ಹೆಸರನ್ನು ಮತ್ತು ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿರಬಹುದು. ಆದರೆ ಆ ಫೈಲ್‌ಗಳು ಬೇರೆ ಬೇರೆ ಡೈರೆಕ್ಟರಿಗಳಲ್ಲಿ ಇರಬೇಕಾಗುತ್ತದೆ. ಏಕೆಂದರೆ ಡಾಸ್ ಗೆ ನಿಮ್ಮ ಆಜ್ಞೆಯು ಒಂದೆ ಹೆಸರನ್ನು ಹೊಂದಿರುವ ಯಾವ ಎರಡು ಫೈಲ್‌ಗಳನ್ನು ಸೂಚಿಸುತ್ತಿದೆ ಎಂಬುದನ್ನು ತಿಳಿಯಬೇಕಾಗುತ್ತದೆ. ಇದಕ್ಕಾಗಿ ಡಾಸ್‌ಗೆ ರೂಟ್‌ನಿಂದ ಪ್ರಾರಂಭಿಸಿ ಅಪೇಕ್ಷಿತ ಫೈಲ್ ಇರುವಲ್ಲಿಯವರೆಗೆ ಡೈರೆಕ್ಟರಿಗಳ ಹೆಸರನ್ನು ತಿಳಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಫೈಲ್‌ವರೆಗೂ ಇರುವ ಈ ಡೈರೆಕ್ಟರಿ ಹೆಸರುಗಳ ಅನುಕ್ರಮವನ್ನು ಮಾರ್ಗ (path) ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಪಾಥ್ ಇದು ಡೈರೆಕ್ಟರಿ ಹೆಸರುಗಳ ಸರಪಳಿಯಾಗಿದ್ದು, ಇದು ನಿಮಗೆ

© NIMI NOT TO BE REPUBLISHED

Fig 1



ಬೇಕಾದ ಫೈಲ್‌ನ್ನು ಹೇಗೆ ಹುಡುಕುವುದು ಎಂಬುದನ್ನು ಡಾಸ್‌ಗೆ ತಿಳಿಸುತ್ತದೆ. ಪ್ರತಿ ಡೈರೆಕ್ಟರಿಯನ್ನು ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಯಿಂದ \ ಚಿನ್ಹೆಯಿಂದ ಬೇರ್ಪಡಿಸಲಾಗುತ್ತದೆ. ಈ \ ಚಿನ್ಹೆಯನ್ನು ಡಾಸ್ ಡೈರೆಕ್ಟರಿ ಡಿಲಿಮಿಟರ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಡ್ರೈವ್ ಸ್ಪೆಸಿಫಾಯರ್‌ನ್ನು ಒಳಗೊಂಡಂತೆ ಒಂದು ಫೈಲ್‌ನ ಸಂಪೂರ್ಣ ಪಾಥ್‌ನ ಹೆಸರು (C: ಅಥವಾ D: ಇತ್ಯಾದಿ) ಅದರ ಫೈಲ್ ಎಲ್ಲಿದೆ ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತದೆ. ವಿಶೇಷ ಪಾಥ್‌ನ ಸಂಕೇತ ಈ ಕೆಳಗಿನಂತಿರುತ್ತದೆ:

D:\Animals\Domestic\Pets\Dig.txt

C:\Admin\Accounts\Tours\Bata.txt

ಪಾಥ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿ ರಚನೆಯ ಹೆಚ್ಚಿನ ವಿವರಗಳನ್ನು ಸೂಕ್ತ ಪಾಠಗಳಲ್ಲಿ ಚರ್ಚಿಸಲಾಗುವುದು

ಡಾಸ್ ಕಮಾಂಡ್ಸ್ (DOS COMMANDS)

1 **MKDIR** ಹೊಸ ಡೈರೆಕ್ಟರಿಯನ್ನು ತಯಾರಿಸುತ್ತದೆ ಅಥವಾ ರಚಿಸುತ್ತದೆ.

ಅಥವಾ

MD

Syntax

MKDIR C:path**dirname**

ಅಥವಾ

MD d:path**dirname**

ಇಲ್ಲಿ,

C: ಇದು ಸಬ್ ಡೈರೆಕ್ಟರಿಯ ಡಿಸ್ಕ್ ಡ್ರೈವ್ ಆಗಿದೆ

Path\ ಇದು ರಚಿಸಲಾಗುತ್ತಿರುವ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಡೈರೆಕ್ಟರಿಯ ಮಾರ್ಗವನ್ನು (ಪಾಥ್)ಸೂಚಿಸುತ್ತದೆ.

Dirname ಇದು ರಚಿಸಲಾಗುತ್ತಿರುವ ಸಬ್ ಡೈರೆಕ್ಟರಿಯ ಹೆಸರನ್ನು ಸೂಚಿಸುತ್ತದೆ.

Switch

(None)

ಮುಖ್ಯವಾದ ಟಿಪ್ಪಣಿ (Important Notes)

- MKDIR ಅಥವಾ ಅದರ ಕಿರು ರೂಪ MD , ಆಯ್ದ ರೂಟ್ ಡೈರೆಕ್ಟರಿ ಅಡಿಯಲ್ಲಿ ಹೊಸ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುತ್ತದೆ.
- ನಿಮಗೆ ಬೇಕಾದಷ್ಟು ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸಬಹುದು. ಆದರೆ ನಿನಪಿಡಿ ಪಾಥ್ ಹೆಸರಿಗಾಗಿ ನೀವು ಉಪಯೋಗಿಸುವ ಬ್ಯಾಕ್ ಸ್ಲ್ಯಾಷ್ ಸೇರಿದಂತೆ 63 ಅಕ್ಷರಗಳಿಗಿಂತ ಹೆಚ್ಚು ಅಕ್ಷರಗಳನ್ನು ಡಾಸ್ ಸ್ವೀಕರಿಸುವುದಿಲ್ಲ.
- ಹೆಚ್ಚಿನ ಹಂತದ ಸಬ್‌ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸಬೇಡಿ ಮತ್ತು ದೀರ್ಘ ಹೆಸರನ್ನು ಉಪಯೋಗಿಸಬೇಡಿ.
- ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಫೈಲ್ ಹೆಸರನ್ನು ಹೋಲುವ ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಉದಾಹರಣೆಗೆ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ನೀವು FILES ಎಂಬ ಹೆಸರಿನ ಫೈಲನ್ನು ಹೊಂದಿದ್ದರೆ, ಇದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ FILES ಎಂಬ ಹೆಸರಿನ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ನೀವು ರಚಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಆದಾಗ್ಯೂ FILES ಫೈಲ್‌ನ ವಿಸ್ತರಣೆ ಹೆಸರು FILES.DOC ಎಂದು ನೀಡಿದರೆ FILES ಎಂಬ ಹೆಸರಿನ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ನೀವು ರಚಿಸಬಹುದು.

ಉದಾಹರಣೆಗಳು

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ **Drivers** ಎನ್ನುವ ಹೆಸರಿನ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸಲು, ಈ ರೀತಿ ಸೂಚನೆಯು ಇರುತ್ತದೆ.

MKDIR**Drivers**

© NIMI NOT TO BE REPUBLISHED

ಅಥವಾ

MD**Drivers**

C:\Devices>MD\Printers

ಈ ಸೂಚನೆಯು ಪ್ರಸ್ತುತ ಡ್ರೈವ್ C: ನಲ್ಲಿ Printers ಹೆಸರಿನ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುತ್ತದೆ. Devices ಎನ್ನುವ ಬೇರೆ ಸಬ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಸೂಚನೆಯನ್ನು ಕೊಡಲಾಗಿದ್ದರೂ ಸಹ Printers ಡೈರೆಕ್ಟರಿಯು Devices ಡೈರೆಕ್ಟರಿ ಅಡಿಯಲ್ಲಿ ರಚಿಸಲಾಗುವುದಿಲ್ಲ ಆದರೆ ನೇರವಾಗಿ ರೂಟ್ C: ಅಡಿಯಲ್ಲಿ ತಯಾರಾಗುತ್ತದೆ. C:\ ಅಡಿಯಲ್ಲಿ ಮತ್ತು C:\Devices ಅಡಿಯಲ್ಲಿ DIR ಸೂಚನೆಯನ್ನು ನೀಡುವ ಮೂಲಕ ಇದನ್ನು ಪರಿಶೀಲಿಸಬಹುದು.

Devices ಡೈರೆಕ್ಟರಿ ಅಡಿಯಲ್ಲಿ ಸಬ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸಲು ಈ ಕೆಳಗಿನಂತೆ ಸೂಚನೆ ನೀಡಬೇಕು.

C:\Devices>MD Printers

ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸುವ ವಿವಿಧ ಪ್ರಕಾರಗಳನ್ನು ಚರ್ಚಿಸಿ

C:\Devices\Printers>MD C:\Devices\Plotters

2 CHDIR ಅಥವಾ CD

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯ ಪಾಥ್‌ಅನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ ಅಥವಾ ತೋರಿಸುತ್ತದೆ.

Syntax

CHDIR d: path

ಅಥವಾ ಕಿರು ರೂಪವನ್ನು ಬಳಸಿ

CD d:path

D: path ಇವು ಸರಿಯಾದ ಡಿಸ್ಕ ಡ್ರೈವ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿ ಹೆಸರು ಆಗಿರುತ್ತವೆ.

Switch

(None)

ಕ್ರಮಾನುಗತ ಡೈರೆಕ್ಟರಿಗಳು CD ಯೊಂದಿಗೆ ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸಲು ನಿಮಗೆ ಎರಡು ವಿಧಾನಗಳಿವೆ. (1)ಡಿಸ್ಕನ ರೂಟ್ ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಮೇಲಿನ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪ್ರಾರಂಬಿಸಿ ಕೆಳಗೆ ಚಲಿಸುವುದು ಅಥವಾ (2) ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪ್ರಾರಂಭಿಸಿ ಮತ್ತು ಎರಡೂ ದಿಕ್ಕಿನಲ್ಲಿ ಚಲಿಸಿ.

ಒಂದು ಡಿಸ್ಕನ ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪ್ರಾರಂಭಿಸಲು , ನೀವು ಪಾಥ್ ಅನ್ನು ಪಾಥ್ ಅಕ್ಷರದಿಂದ (\) ಅಥವಾ B:\ ನಿಂದ ಪ್ರಾರಂಭಿಸಬೇಕು. ಯಾವಾಗ ಡಾಸ್ ಪಾಥ್‌ನಲ್ಲಿ \ ಅಕ್ಷರವನ್ನು ಮೊದಲ ಅಕ್ಷರವನ್ನಾಗಿ ನೋಡುತ್ತದೆಯೊ ಆಗ ಸಿಸ್ಟಮ್ ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಇಲ್ಲದಿದ್ದರೆ ಡಾಸ್ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪ್ರಾರಂಭಿಸುತ್ತದೆ.

ಡ್ರೈವ್ ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು : ಕಂಪ್ಯೂಟರ್ ಅಂತರ್ನಿರ್ಮಿತ ಮೆಮೊರಿ, ಹಾರ್ಡಡಿಸ್ಕನ್ನು ಹೊಂದಿರುತ್ತದೆ, ಇದು ಫ್ಲಾಪಿ ಡಿಸ್ಕ, ಕಾಂಪ್ಯಾಕ್ಟ ಡಿಸ್ಕ ಮುಂತಾದುವುಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಲು/ಓದಲು ಸಹ ಅವಕಾಶ ಮಾಡಿಕೊಡುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ಡಿಸ್ಕ C ಡ್ರೈವ್, A ಡ್ರೈವ್ , B ಡ್ರೈವ್ ಮುಂತಾದ ಹೆಸರಿಂದ ಗುರುತಿಸಲ್ಪಡುತ್ತವೆ. C ಡ್ರೈವ್ ಅನ್ನು C: ಎಂದು ಬರೆಯಲಾಗುತ್ತದೆ ಮತ್ತು A ಡ್ರೈವ್ ಅನ್ನು A: ಎಂದು ಬರೆಯಲಾಗುತ್ತದೆ. ಪ್ರಸ್ತುತ ಡ್ರೈವ್‌ನಿಂದ ಬದಲಾಯಿಸಲು , ಡಿಸ್ಕ ಡ್ರೈವ್‌ನ ಅಕ್ಷರ ನಂತರ ಕೋಲನ್ ಚಿನ್ಹೆಯನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಬರೆಯಬೇಕು.

C:\> a:

ಈ ಸೂಚನೆಯು ನಿಯಂತ್ರಣವನ್ನು **C** ಡ್ರೈವ್‌ನಿಂದ **A** ಡ್ರೈವ್‌ಗೆ ಬದಲಾಯಿಸಲು ಸೂಚಿಸುತ್ತದೆ.

ಫ್ಲಾಪಿ ಲಭ್ಯವಿಲ್ಲದ ಕಾರಣದಿಂದ ಅಥವಾ ಬೇರೆ ಯಾವುದೇ ಕಾರಣಗಳಿಂದ ಡಿಸ್ಕ ಡ್ರೈವ್ ತೆರೆಯದಿದ್ದರೆ, ಡಾಸ್ ದೋಷ ಸಂದೇಶವನ್ನು (error message) ತೋರಿಸುತ್ತದೆ.

Not ready error reading drive A

Abort, Retry, Fail?

A , R ಅಥವಾ F ಕೀಲಿಯನ್ನು ಒತ್ತುವ ಅಗತ್ಯವಿದೆ. ಅವುಗಳನ್ನು ಈ ಕೆಳಗಿನಂತೆ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ.

A ದೋಷಕ್ಕೆ ಕಾರಣವಾಗಿರುವ ಆಜ್ಞೆಯನ್ನು ಸ್ಥಗಿತಗೊಳಿಸುವಂತೆ ಡಾಸ್‌ಅನ್ನು ನಿರ್ದೇಶಿಸುತ್ತದೆ. ಈ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿದರೆ ಡಾಸ್ ಆಜ್ಞೆಯನ್ನು ಅಂತ್ಯಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಪ್ರಾಂಪ್ಟನ್ನು ಮರುಪ್ರದರ್ಶಿಸುತ್ತದೆ.

R ದೋಷಕ್ಕೆ ಕಾರಣವಾದ ಆಜ್ಞೆಯನ್ನು ಮರುಪ್ರಯತ್ನಿಸಲು ಡಾಸ್‌ನ್ನು ನಿರ್ದೇಶಿಸುತ್ತದೆ. ದೋಷವನ್ನು ಉಂಟುಮಾಡಿದ ಸಮಸ್ಯೆಯನ್ನು ಸರಿಪಡಿಸಲು ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ ಈ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಲಾಗುತ್ತದೆ.(ಫ್ಲಾಪಿ ಡಿಸ್ಕನ್ನು ಅಳವಡಿಸದಿರಬಹುದು)

F ದೋಷವನ್ನು ನಿರ್ಲಕ್ಷಿಸಲು ಮತ್ತು ಪ್ರಕ್ರಿಯೆಯನ್ನು ಮುಂದುವರಿಸಲು ಪ್ರಯತ್ನಿಸಲು ಡಾಸ್‌ನ್ನು ನಿರ್ದೇಶಿಸುತ್ತದೆ. ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ಡಿಸ್ಕನ ಒಂದು ಭಾಗವನ್ನು ಓದಿದಾಗ ಡಾಸ್ ದೋಷವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಡಾಸ್ ಕಮಾಂಡ್ (DOS COMMAND)

DIR ಒಂದು ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್ ಗಳು ಮತ್ತು ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳ ಪಟ್ಟಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.

Syntax

DIRC:path/filename[/P][/W][/A[[:attribs]][/o[[:]sor-

tord]]

[/S][/B][/L][/C[H]]

ಇಲ್ಲಿ

- **C:** ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳ ಪಟ್ಟಿಯನ್ನು ತೋರಿಸುವ ಡೈರೆಕ್ಟರಿಯನ್ನು ಹೊಂದಿರುವ ಡಿಸ್ಕ ಡ್ರೈವ್ ಆಗಿದೆ.
- **path/** ಪಟ್ಟಿ ಮಾಡಬೇಕಾದ ಡೈರೆಕ್ಟರಿ ಮತ್ತು/ಅಥವಾ ಫೈಲ್‌ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತದೆ.
- **filename** ತೋರಿಸಬೇಕಾದ ಫೈಲ್ ಅಥವಾ ಫೈಲ್‌ಗಳ ಪಟ್ಟಿಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತದೆ. ಫೈಲ್‌ನ ಹೆಸರನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಲಾಗುತ್ತದೆ.
- **[/P] [/W] ....** ಔಟಪುಟ್ ಅನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತದೆ.

Switch

/P ಪ್ರತಿ ಸ್ಕ್ರೀನ್‌ನ ಮಾಹಿತಿ ಪ್ರದರ್ಶನದ ನಂತರ ನಿಲುಗಡೆಯಾಗುತ್ತದೆ ಮತ್ತು ಯಾವುದೇ ಕೀ ಒತ್ತುವುದನ್ನು ಕಾಯುತ್ತದೆ. ಯಾವುದೇ ಕೀ ಒತ್ತಿದ ನಂತರ ಮತ್ತೊಂದು ಸ್ಕ್ರೀನ್ ಅಥವಾ ಉಳಿದ ಮಾಹಿತಿಯನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಇದಕ್ಕೆ ಆಜ್ಞೆಯು DIR/P ಆಗಿರುತ್ತದೆ

/W 80 ಕಾಲಮ್‌ಗಳ ವಿಶಾಲ ಫಾರ್ಮ್ಯಾಟನ್ನು ಉಪಯೋಗಿಸಿ ಫೈಲ್ ಹೆಸರುಗಳನ್ನು ಮಾತ್ರ ಪ್ರದರ್ಶಿಸುತ್ತದೆ ಮತ್ತು ಫೈಲ್‌ನ ಗಾತ್ರದ ಬಗ್ಗೆ ಮಾಹಿತಿ , ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ಪ್ರದರ್ಶಿಸುವುದಿಲ್ಲ. ಇದಕ್ಕೆ ಆಜ್ಞೆಯು DIR/W ಆಗಿರುತ್ತದೆ.

/A ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಗುಣಲಕ್ಷಣಗಳೊಂದಿಗೆ ಫೈಲ್‌ಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಗುಣಲಕ್ಷಣಗಳು(attribs)

D ಡೈರೆಕ್ಟರಿಗಳು

R ರೀಡ್ ಓನ್ಲಿ ಫೈಲ್‌ಗಳು

H ಹಿಡನ್ ಫೈಲ್‌ಗಳು

S ಸಿಸ್ಟಮ್ ಫೈಲ್‌ಗಳು

A ಆರ್ಚೀವ್ ಮಾಡಲು ಸಿದ್ಧವಾಗಿರುವ ಫೈಲ್‌ಗಳು-

/O ವಿಂಗಡಿಸಲಾದ ಕ್ರಮದಲ್ಲಿರುವ ಫೈಲ್‌ಗಳ ಪಟ್ಟಿ

ವಿಂಗಡಿಸಲಾದ N ಹೆಸರಿನ ಪ್ರಕಾರ (ವರ್ಣಮಾಲೆ)

S ಗಾತ್ರದ ಪ್ರಕಾರ (ಮೊದಲು ಚಿಕ್ಕದು)

E ವಿಸ್ತರಣೆ ಪ್ರಕಾರ (ವರ್ಣಮಾಲೆ)

D ದಿನಾಂಕ ಮತ್ತು ಸಮಯದ ಪ್ರಕಾರ(ಮೊದಲಿನದು)

G ಗ್ರೂಪ್ ಡೈರೆಕ್ಟರಿಗಳು ಮೊದಲು

- ಹಿಮ್ಮುಖ ಕ್ರಮಕ್ಕೆ ಪೂರ್ವಪ್ರತ್ಯಯ

C ಕಂಪ್ರೆಷನ್ ಅನುಪಾತದಿಂದ(ಹಳೆಯದು ಮೊದಲು)

/S ನಿರ್ದಿಷ್ಟ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಎಲ್ಲ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ

/B ಬೇರ್ ಸ್ವರೂಪವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ (ಶೀರ್ಷಿಕೆ ಮಾಹಿತಿ ಅಥವಾ ಸಾರಾಂಶ ಇರುವುದಿಲ್ಲ)

/L ಸಣ್ಣ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

/C[H] ಫೈಲ್ ಕಂಪ್ರೆಷನ್ ಅನುಪಾತವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ./CH ಹೋಸ್ಟ ಅಲೋಕೇಶನ್ ಯುನಿಟ್ ಗಾತ್ರವನ್ನು ಬಳಸುತ್ತದೆ.

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes) :

- ಡೈರೆಕ್ಟರಿ ಪಟ್ಟಿಯಲ್ಲಿ ವೈಲ್ಡಕಾರ್ಡ ಚಿನ್ಹೆಗಳನ್ನು (* ಮತ್ತು ?) ಉಪಯೋಗಿಸಿ ಒಂದೇ ರೀತಿಯ ಫೈಲ್ ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಬಹುದು. ಇಲ್ಲಿ (*) ಸ್ಟಾರ್ ಮತ್ತು (?) ಪ್ರಶ್ನಾರ್ಥಕ ಚಿನ್ಹೆಗಳನ್ನು ವೈಲ್ಡಕಾರ್ಡ ಚಿನ್ಹೆಗಳು, ಅಥವಾ ವೈಲ್ಡಕಾರ್ಡಗಳು, ಎಂದು ಕರೆಯುತ್ತಾರೆ. * ಇದು ಉಳಿದ ಅಕ್ಷರಗಳನ್ನು ಬದಲಾಯಿಸಬಲ್ಲದು ಮತ್ತು ? ಇದು ಯಾವುದೇ ಒಂದೇ ಅಕ್ಷರವನ್ನು ಬದಲಾಯಿಸಬಲ್ಲದು.
- DIR ಅನ್ನು ಯಾವುದೇ ಮಾನದಂಡ(parameter) ಅಥವಾ ಸ್ವಿಚ್‌ಗಳಿಲ್ಲದೇ ಉಪಯೋಗಿಸಿದಾಗ ಅದು ಡಿಸ್ಕ ವಾಲ್ಯೂಮ ಲೇಬಲ್‌ನ್ನು ಮತ್ತು ಸೀರಿಯಲ್ ಸಂಖ್ಯೆಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. , ಪ್ರತಿ ಸಾಲಿನಲ್ಲಿ ಒಂದು ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲ್‌ನ ಹೆಸರು, ಬೈಟ್‌ಗಳಲ್ಲಿ ಫೈಲ್‌ನ ಗಾತ್ರ, ಮತ್ತು ಫೈಲನ್ನು ಮಾರ್ಪಡಿಸಿದ ದಿನಾಂಕ ಮತ್ತು ಸಮಯ ಮತ್ತು ಪಟ್ಟಿ ಮಾಡಲಾದ ಫೈಲ್‌ಗಳ ಒಟ್ಟೂ ಸಂಖ್ಯೆ , ಅವುಗಳ ಗಾತ್ರ ಮತ್ತು ಡಿಸ್ಕನಲ್ಲಿರುವ ಖಾಲಿ ಸ್ಥಳ ಇವುಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಉದಾಹರಣೆಗಳು (Examples)

DIR *.txt

*.txt ಈ ಆಜ್ಞೆಯು ನಿರ್ದಿಷ್ಟ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ txt ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿದ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.

DIR ???T.*

???T ಆಜ್ಞೆಯು ನಾಲ್ಕು ಅಕ್ಷರಗಳನ್ನು ಹೊಂದಿದ ಮತ್ತು T ಅಕ್ಷರದಿಂದ ಕೊನೆಗೊಳ್ಳುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಹುಡುಕುತ್ತದೆ. ಉದಾಹರಣೆಗೆ TEST, REST ಇತ್ಯಾದಿ . ಮತ್ತು * ಇದು ಫೈಲ್‌ಗಳು ಯಾವುದೇ ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿರಬಹುದು ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ .txt, .dat ಇತ್ಯಾದಿ

©NIMI NOT TO BE REPUBLISHED

ಡಾಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಫೈಲ್ ಮತ್ತು ಫೋಲ್ಡರ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage files and folders using DOS commands)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- **ಟೆಕ್ಸ್ಟ್ ಫೈಲ್‌ನ ವಿಷಯಗಳನ್ನು ಪ್ರದರ್ಶಿಸಲು ಡಾಸ್ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು**
- **ಫೈಲ್‌ಗಳನ್ನು ನಕಲು(copy) ಮಾಡಲು, ಮರುಹೆಸರು(rename)ಇಡಲು ,ಅಳಿಸಲು(delete), ಮತ್ತು ಅನ್‌ಡಿಲೀಟ್ (undelete) ಮಾಡಲು ಡಾಸ್ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು.**

ಡಾಸ್ ಆಜ್ಞೆಗಳು (DOS Commands)

TYPE ಒಂದು ಟೆಕ್ಸ್ಟ್ ಫೈಲ್ (Text File) ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

Syntax

TYPE C:path/filename

ಇಲ್ಲಿ,

- **C:** ಪ್ರದರ್ಶಿಸಬೇಕಾದ ಫೈಲ್ ಇರುವ ಡಿಸ್ಕ ಡ್ರೈವ್
- **Path/** ಪ್ರದರ್ಶಿಸಬೇಕಾದ ಫೈಲ್ ನ ಸ್ಥಳವನ್ನು ತಿಳಿಸುತ್ತದೆ
- **Filename** ಪ್ರದರ್ಶಿಸಬೇಕಾದ ಫೈಲ್ ಅನ್ನು ತಿಳಿಸುತ್ತದೆ.

Switch

(ಯಾವುದೂ ಇಲ್ಲ)

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes)

- **TYPE** ಆಜ್ಞೆಯು ಒಂದು ಆಸ್ಕೀ (ASCII) ಫೈಲ್‌ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಬೇರೆ ಪ್ರೋಗ್ರಾಂ ಬಳಸದೇ, ಪ್ರದರ್ಶಿಸಲು ತ್ವರಿತವಾದ ಮಾರ್ಗವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಫೈಲ್ ಡಿಸ್ಕನಲ್ಲಿ ಆಸ್ಕೀ (ASCII - American Standard Code for Information Interchange) ಟೆಕ್ಸ್ಟ್ ಆಗಿ ಸಂಗ್ರಹ ಆಗಿರುತ್ತದೆ. ಇದು ಬೈನರಿ (ಒಂದು ಮತ್ತು ಸೊನ್ನೆ) ಅನ್ನು ಅಕ್ಷರಗಳು, ಸಂಖ್ಯೆಗಳು ಮತ್ತು ಚಿನ್ಹೆಗಳಿಗೆ ಭಾಷಾಂತರಿಸುವ ಕಂಪ್ಯೂಟರ್‌ನ ಪ್ರಮಾಣಿತ ಮಾರ್ಗವಾಗಿದೆ. ಮಾಹಿತಿಯನ್ನು ಆಸ್ಕೀ ಸ್ವರೂಪದಲ್ಲಿ ಸಂಗ್ರಹಿಸದಿದ್ದರೆ, TYPE ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿದಾಗ ಮಾಹಿತಿಯು ಅರ್ಥವಾಗದ ಭಾಷೆಯಾಗಿ ಕಾಣುತ್ತದೆ.
- ಆಜ್ಞೆಯನ್ನು ನೀಡಿದಾಗ ಡಾಸ್ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಡ್ರೈವ್‌ನಲ್ಲಿ ಅದನ್ನು ಹುಡುಕಾಡುತ್ತದೆ. ಮತ್ತು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಫೈಲ್‌ನ ಹೆಸರನ್ನು ತಲುಪಲು ತಿಳಿಸಿದ ಪಾಥ್ ನಲ್ಲಿ ಚಲಿಸುತ್ತದೆ. ನಂತರ ಇದು ಆಸ್ಕೀ ಸ್ವರೂಪವನ್ನು ಅಕ್ಷರಗಳಿಗೆ, ಸಂಖ್ಯೆಗಳಿಗೆ ಮತ್ತು ಚಿನ್ಹೆಗಳಿಗೆ ಭಾಷಾಂತರಿಸುತ್ತದೆ ಮತ್ತು ಮಾನಿಟರ್ ಮೇಲೆ ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ವಿಡಿಯೋ ಮಾನಿಟರ್ 24 ಸಾಲುಗಳ ಮಾಹಿತಿಯನ್ನು ಮಾತ್ರ ತೋರಿಸುತ್ತದೆ. ಫೈಲ್ 24 ಸಾಲುಗಳಿಗಿಂತ ಹೆಚ್ಚು ಸಾಲುಗಳನ್ನು ಹೊಂದಿದ್ದರೆ ಮೊದಲನೆಯ ಸಾಲುಗಳು ಕಾಣುವುದಿಲ್ಲ. TYPE ಆಜ್ಞೆಯು ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ಪರದೆಯ ಮೇಲೆ ಸ್ಕ್ರೋಲ್ ಮಾಡುತ್ತದೆ. ಕಂಟ್ರೋಲ್ + S ಕೀಲಿಗಳನ್ನು ಒಟ್ಟಿಗೆ (ಕಂಟ್ರೋಲ್ ಕೀಲಿಯನ್ನು ಹಿಡಿದಿಟ್ಟು S ಕೀಲಿಯನ್ನು ಒತ್ತುವುದು ಮತ್ತು ಎರಡೂ ಕೀಲಿಗಳನ್ನು ಒಟ್ಟಿಗೆ ಬಿಡುವುದಕ್ಕೆ ಕಂಟ್ರೋಲ್ + S ಎಂದು ಕರೆಯುತ್ತಾರೆ) ಒತ್ತುವ ಮೂಲಕ ಸ್ಕ್ರೋಲಿಂಗನ್ನು ನಿಯಂತ್ರಿಸಬಹುದು. ಮಾನಿಟರ್ ಮೇಲೆ ಮಾಹಿತಿಯು ಸ್ಕ್ರೋಲ್ ಆಗುವುದನ್ನು ತಡೆಯುತ್ತದೆ. ಪರದೆಯ ಮೇಲಿರುವ ಮಾಹಿತಿಯನ್ನು ನೋಡಿದ ನಂತರ ಉಳಿದ ಮಾಹಿತಿಯನ್ನು ನೋಡಲು ಯಾವುದೇ ಕೀಲಿಯನ್ನು ಒತ್ತಬಹುದು. ಫೈಲ್‌ನ ಮಾಹಿತಿಯನ್ನು ಒಂದೊಂದೆ ಸ್ಕ್ರೀನ್ ಪುಟವನ್ನಾಗಿ ನೋಡುವುದಕ್ಕಾಗಿ MORE ಆಜ್ಞೆಯನ್ನು ಸಹ ಉಪಯೋಗಿಸಬಹುದು. ಇದು ಪರದೆಯ ಒಂದು ಪುಟದ ನಂತರ ಪರದೆಯ ಮೇಲಿರುವ ಮಾಹಿತಿಯು ಸ್ಕ್ರೋಲ್ ಆಗುವುದನ್ನು ತಡೆಯುತ್ತದೆ ಮತ್ತು 24 ಸಾಲುಗಳ ನಂತರ ಪರದೆಯ ಮೇಲೆ more ಎನ್ನುವ ಪ್ರಾಂಪ್ಟ ಸಂದೇಶವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ನಂತರ ಯಾವುದೇ ಕೀಲಿಯನ್ನು ಒತ್ತುವುದರಿಂದ ನಂತರದ ಪರದೆಯ ಪುಟವನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. MORE ಇದು ಒಂದು ಫಿಲ್ಟರ್ ತರಹ ಕೆಲಸ ಮಾಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಫೈಲ್‌ನ ಪ್ರಮಾಣಿತ ಅಕ್ಷರಗಳ ಹರಿವನ್ನು, ಪ್ರಮಾಣಿತ ಔಟಪುಟ್‌ಗೆ ಒಂದೊಂದೆ ಪರದೆಯ ಪುಟಗಳನ್ನು ಯುಕ್ತಿಯಿಂದ ನಿರ್ವಹಿಸುತ್ತದೆ.

ಉದಾಹರಣೆಗಳು (Examples)

1 C:\COPA\DOS\PRACT_3>**TYPE TEST1.txt**

C:\COPA\DOS\PRACT_3 ಇದು TEST1.txt ಈ ಫೈಲ್‌ನ ಪಾಥ್ ಆಗಿರುತ್ತದೆ. ಮತ್ತು TYPE ಎನ್ನುವುದು ಡಾಸ್‌ನಿಂದ ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಆಜ್ಞೆಯಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

2 C:\>TYPE C:\ COPA\DOS\PRACT_3\> **TEST1.txt**

ಇದು ಉದಾಹರಣೆ ಒಂದರಂತೆ ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ. C:(C ಡ್ರೈವ್) ನಿಂದ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದರಿಂದ ಈ ಆಜ್ಞೆಯನ್ನು ಡೈರೆಕ್ಟರಿ ಬದಲಾಯಿಸದೆ ನೀಡಬಹುದು.

3 C:\COPA\DOS\PRACT_3>**TYPE TEST1.txt| MORE**

ಇದು ಮೇಲಿನಂತೆ ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ ಆದರೆ ಒಂದೊಂದೆ ಪರದೆಯ ಪುಟಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. ಪುಟವನ್ನು ಯಾವುದೇ ಕೀಲಿಯನ್ನು ಒತ್ತುವುದರಿಂದ ಬದಲಾಯಿಸಬಹುದು. ಜೊತೆಗೆ ಮತ್ತೊಂದು ಅಕ್ಷರ | ಉಪಯೋಗಿಸಲಾಗಿದೆ. ಇದನ್ನು ಪೈಪಿಂಗ್(Piping)ಆಜ್ಞೆ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು TYPE ಆಜ್ಞೆಯ ಉತ್ತರವನ್ನು ಇನ್ನೊಂದು ಆಜ್ಞೆಯಾದ . MORE ಗೆ ಮಾರ್ಗ ಮಾಡುತ್ತದೆ. ನಂತರ MORE ಆಜ್ಞೆಯು ಮಾಹಿತಿಯನ್ನು ಫಿಲ್ಟರ್ ಮಾಡಿ ತೋರಿಸುತ್ತದೆ.

ಫೈಲ್(ಗಳ) ಮರುಹೆಸರಿಸುವಿಕೆ(Renaming of file(s))

RENAME ಅಥವಾ REN ಈ ಆಜ್ಞೆಯು ಫೈಲ್‌ನ ಹೆಸರನ್ನು ಬದಲಾಯಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

Syntax

REN C:PATH\filename1.ex1 filename2.ex2

ಇಲ್ಲಿ,

- **C:** ಪ್ರದರ್ಶಿಸಬೇಕಾದ ಫೈಲ್ ಇರುವ ಡಿಸ್ಕ ಡ್ರೈವ್
- **PATH**ಪ್ರದರ್ಶಿಸಬೇಕಾದ ಫೈಲ್‌ನ ನಿರ್ದಿಷ್ಟ ಸ್ಥಳವನ್ನು ತೋರಿಸುತ್ತದೆ.
- filename1.ex1 ಮರುಹೆಸರಿಸಬೇಕಾದ (rename) ಫೈಲ್‌ನ ಹೆಸರು
- filename2.ex2 ಹೊಸ ಫೈಲ್ ನ ಹೆಸರು

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes)

- ಡ್ರೈವ್‌ನ ಹೆಸರನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ, ಪ್ರಸ್ತುತ ಡಿಸ್ಕ ಡ್ರೈವನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ಪಾಥ್‌ಅನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ
- ಮರು ಹೆಸರಿಸಬೇಕಾದ ಫೈಲ್ ನ ನಿಖರವಾದ ಹೆಸರನ್ನು ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ನೀಡಬೇಕು.
- ಫೈಲ್ ನ ಸೂಕ್ತವಾದ ಹೊಸ ಹೆಸರನ್ನು ಸರಿಯಾದ ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ನೀಡಬೇಕು
- ಅಗತ್ಯವಿರುವ ಫೈಲ್‌ಗಳ ಗುಂಪನ್ನು ಮರುಹೆಸರಿಸಬೇಕಾದರೆ ವೈಲ್ಡಕಾರ್ಡ್ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.
- ಫೈಲ್ ಹೆಸರುಗಳು ಮಾತ್ರ ಬದಲಾಗುತ್ತವೆ ಮತ್ತು ಅದರಲ್ಲಿನ ಮಾಹಿತಿಗಳು ಅದೆ ರೀತಿಯಾಗಿ ಉಳಿದುಕೊಳ್ಳುತ್ತವೆ.
- ಒಂದು ಫೈಲ್ ಹೆಸರನ್ನು ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಈಗಾಗಲೇ ಇರುವ ಫೈಲ್ ಹೆಸರಿಗೆ ಬದಲಾಯಿಸಲು ಪ್ರಯತ್ನಿಸಿದರೆ

  ಡಾಸ್ ಈ ಕೆಳಗಿನಂತೆ ದೋಷ ಸಂದೇಷವನ್ನು ತೋರಿಸುತ್ತದೆ.

  Duplicate filename or file not found
- ಯಾವುದೇ ಅಮಾನ್ಯ ಫೈಲ್ ಹೆಸರು ನೀಡಿದರೆ ಅಥವಾ ಹೊಸ ಹೆಸರನ್ನು ನೀಡದಿದ್ದರೆ, ಡಾಸ್ ದೋಷ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ.

ಫೈಲ್ ಹೆಸರನ್ನು ನೀಡುವುದಕ್ಕೆ ನಿಯಮಗಳು

- ಒಂದು ಫೈಲ್ ಹೆಸರಿನಲ್ಲಿ 1 ರಿಂದ 8 ಅಕ್ಷರಗಳು ಇರಬೇಕು.
- **1** ರಿಂದ 3 ಅಕ್ಷರಗಳ ಐಚ್ಚಿಕ ವಿಸ್ತರಣಾ ಹೆಸರು ಇರಬೇಕು
- ವಿಸ್ತರಣೆಯನ್ನು ಉಪಯೋಗಿಸಿದರೆ, ಫೈಲ್ ಹೆಸರು ಮತ್ತು ವಿಸ್ತರಣೆಯ ಮಧ್ಯದಲ್ಲಿ ಪಿರಿಯಡ್(.) ಉಪಯೋಗಿಸಬೇಕು
- **A** ದಿಂದ **Z** ವರೆಗಿನ ಅಕ್ಷರಗಳು(ಲೋವರ್ ಕೇಸ್ ಅಕ್ಷರಗಳನ್ನು ಅಪ್ಪರ್ ಕೇಸ್ ಅಕ್ಷರಗಳಿಗೆ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ವರ್ಗಾಯಿಸಲಾಗುತ್ತದೆ.) 0 ದಿಂದ 9 ಸಂಖ್ಯೆಗಳು ಮತ್ತು ವಿಶೇಷ ಅಕ್ಷರಗಳು ಮತ್ತು ಚಿನ್ಹೆಗಳು **$ # & @ ! ^ ( ) _ - { } ` ~** ಇವುಗಳನ್ನು ಫೈಲ್ ಹೆಸರಿಡುವುದಕ್ಕೆ ಅನುಮತಿಸಲಾಗಿದೆ.
- ಫೈಲ್ ಹೆಸರಿನಲ್ಲಿ ಕಂಟ್ರೋಲ್ ಅಕ್ಷರಗಳಾದ **Esc, Del**, ಅಥವಾ **space bar** ಇವುಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಬರುವುದಿಲ್ಲ.
- **+ = / [ ] : ; ? * < >** ಈ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಬರುವುದಿಲ್ಲ.
- ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಫೈಲ್ ನ ಹೆಸರು ವಿಶಿಷ್ಟವಾಗಿರಬೇಕು.

ಉದಾಹರಣೆಗಳು (Examples) :

1 C:\COPA\DOS\PRACT_3>**REN TEST2.txt CHECKED .txt**

C:\COPA\DOS\PRACT_3 ಇದು ಡ್ರೈವ್ ಮತ್ತು **TEST2.tx**t ಫೈಲ್ ಗೆ ಪಾಥ್ ಆಗಿರುತ್ತದೆ.

TEST2.txt ಇದು ಮರುಹೆಸರಿಸಬೇಕಾದ ಫೈಲ್‌ನ ಹೆಸರು.

CHECKED .txt ಇದು ಹೊಸ ಫೈಲ್ ಹೆಸರು.

2 C:\COPA\DOS\PRACT_3>REN *.pic *.jpg ಇ

© NIMI NOT TO BE REPUBLISHED

ದು pic ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿದ ಫೈಲ್‌ಗಳನ್ನು jpg ವಿಸ್ತರಣೆ ಹೊಂದಿದ ಫೈಲ್‌ಗಳನ್ನಾಗಿ ಬದಲಾಯಿಸುತ್ತದೆ.

ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡುವುದು (Copying files)

COPY ಇದು ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳನ್ನು ಬೇರೆ ಸ್ಥಳಕ್ಕೆ ಕಾಪಿ ಮಾಡುತ್ತದೆ.

Syntax

COPY [/A | /B] source [/A | /B] [+/A | /B] [+ _ _]] [destination [/A | /B]] [/V] [/Y | /- Y] ಇಲ್ಲಿ **source** ಎಂದರೆ ಕಾಪಿ ಮಾಡಬೇಕಾದ ಫೈಲ್ ಅಥವಾ ಫೈಲ್‌ಗಳು. **Destination** ಎಂದರೆ ಹೊಸ ಫೈಲ್‌ಗಳಿಗಾಗಿ ಡೈರೆಕ್ಟರಿ ಮತ್ತು / ಅಥವಾ ಫೈಲ್ ಹೆಸರು.

Switches

/A ASCII ಟೆಕ್ಸ್ಟ್ ಫೈಲನ್ನು ಸೂಚಿಸುತ್ತದೆ.

/B ಬೈನರಿ ಫೈಲನ್ನು ಸೂಚಿಸುತ್ತದೆ.

/V ಹೊಸ ಫೈಲ್‌ಗಳು ಸರಿಯಾಗಿ ಬರೆಯಲ್ಪಟ್ಟಿವೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ.

/Y ಈಗಾಗಲೇ ಇರುವ ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲನ್ನು ಬದಲಿಸಿ ಬರೆಯುವ ಬಗ್ಗೆ ದೃಢೀಕರಿಸಿ ಕೇಳುವುದನ್ನು ನಿಗ್ರಹಿಸುತ್ತದೆ.

/-Y ಈಗಾಗಲೇ ಇರುವ ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲನ್ನು ಬದಲಿಸಿ ಬರೆಯುವ ಬಗ್ಗೆ ದೃಢೀಕರಿಸಿ ಕೇಳುವುದನ್ನು ಪ್ರೇರೇಪಿಸುತ್ತದೆ.

ಪ್ರತೀ ಪ್ರಕರಣದಲ್ಲಿ ಬೋಧಕರು ಸರಳ ಸ್ವಿಚ್‌ಗಳನ್ನು ಕನಿಷ್ಟ ಎರಡು ಉದಾಹರಣೆಗಳೊಂದಿಗೆ ಚರ್ಚಿಸಬೇಕು

COPY ಆಜ್ಞೆಯ ಸ್ವಿಚ್ ಗಳ ಹೆಚ್ಚಿನ ವಿವರಗಳಿಗಾಗಿ, ಯಾವುದೇ ಟ್ಯುಟೋರಿಯಲ್ ಅಥವಾ ಡಾಸ್ ಕೈಪಿಡಿಯನ್ನು ನೋಡಿರಿ.

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes) :

- ಡಾಸ್ ಆಜ್ಞೆಯಾದ **COPY** ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡುತ್ತದೆ. ಒಂದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಬೇರೆ ಬೇರೆ ಹೆಸರಿನಿಂದ ಅಥವಾ ಒಂದು ಡೈರೆಕ್ಟರಿಯಿಂದ ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಗೆ ಒಂದೇ ಹೆಸರಿನಿಂದ ಅಥವಾ ಬೇರೆ ಹೆಸರಿನಿಂದ ಕಾಪಿ ಮಾಡಬಹುದು.
- ನಿರ್ದಿಷ್ಟ ಡ್ರೈವ್ ಹೆಸರನ್ನು ಹೇಳದಿದ್ದರೆ , ಪ್ರಸ್ತುತ ಡಿಸ್ಕ ಡ್ರೈವನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ನಿರ್ದಿಷ್ಟ ಪಾಥ ಹೆಸರನ್ನು ಹೇಳದಿದ್ದರೆ , ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ಕಾಪಿ ಮಾಡಬೇಕಾದ ಫೈಲ್‌ನ ನಿಖರವಾದ ಹೆಸರನ್ನು ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ನೀಡಬೇಕು.
- ಸೂಕ್ತವಾದ ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ಸರಿಯಾದ ಫೈಲ್ ಹೆಸರನ್ನು ಕಾಪಿ ಮಾಡಿದ ಹೊಸ ಫೈಲ್‌ಗೆ ನೀಡಬೇಕು.
- ಅಗತ್ಯವಿರುವ ಫೈಲ್‌ಗಳ ಗುಂಪನ್ನು ಕಾಪಿ ಮಾಡಬೇಕಾದರೆ ವೈಲ್ಡಕಾರ್ಡ್ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.
- ಕಾಪಿ ಮಾಡಿದಾಗ, **source** ಮತ್ತು **destination** ಎರಡೂ ಫೈಲ್‌ಗಳು ಒಂದೇ ವಿಷಯವನ್ನು ಹೊಂದಿರುತ್ತವೆ.
- ಕಾಪಿ ಆಜ್ಞೆಯು ಟಾರ್ಗೆಟ್ ಫೈಲನ್ನು ಅದೇ ಹೆಸರಿನಿಂದ ಬದಲಾಯಿಸುತ್ತದೆ.
- ಕಾಪಿ ಆಜ್ಞೆಯು ಒಂದು ಫೈಲನ್ನು ತನ್ನನ್ನು ತಾನೆ ಕಾಪಿ ಮಾಡಲು ಅನುಮತಿಸುವುದಿಲ್ಲ. ಎಂದರೆ **source** ಮತ್ತು **destination** ಫೈಲಗಳು ಒಂದೇ ಆಗಿರಲು ಸಾಧ್ಯವಿಲ್ಲ.
- **destination** ಫೈಲ್ ಹೆಸರನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ, ಒಗ್ಗೂಡಿಸುವಾಗ ಮೊದಲನೇ ಫೈಲ್ ಹೆಸರು destination ಫೈಲ ಹೆಸರಾಗುತ್ತದೆ. ಮೊದಲನೇ ಫೈಲ್ ಹೆಸರಿನ ನಂತರ ಹೆಚ್ಚುವರಿ source ಫೈಲ್‌ನ ಮೊದಲು ಪ್ಲಸ್(+) ಚಿಹ್ನೆಯನ್ನು ಬರೆಯಬೇಕು.

ಉದಾಹರಣೆಗಳು(Examples)

1 C:\COPA\DOS\PRACT_3>**COPY TEST2.txt TRIAL.txt**

ಈ ಮೇಲಿನ ಆಜ್ಞೆಯು C:\COPA\DOS\PRACT_3 ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ TEST2.txt ಫೈಲನ್ನು TRIAL.txt ಎಂಬ ಹೆಸರಿನಿಂದ ಅದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಕಾಪಿ ಮಾಡುತ್ತದೆ. ಡೈರೆಕ್ಟರಿಯನ್ನು ಪಟ್ಟಿ ಮಾಡಿದಾಗ ಎರಡೂ ಫೈಲ್‌ಗಳು ಒಂದೇ ವಿವರಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಮತ್ತು ಎರಡೂ ಫೈಲ್‌ನ ವಿಷಯಗಳನ್ನು ನೋಡಿದಾಗ ಒಂದೇ ಆಗಿರುತ್ತವೆ. ಕಾಪಿ ಮಾಡಿದ ನಂತರ 1file copied ಎಂಬ ಸಂದೇಶವನ್ನು ಡಾಸ್ ತೋರಿಸುತ್ತದೆ.

2 C:\COPA\DOS\PRACT_3>**COPY *. Bmp *. Pic**

ಈ ಮೇಲಿನ ಆಜ್ಞೆಯು C:\COPA\DOS\PRACT_3 ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ **Bmp** ವಿಸ್ತರಣೆ ಹೊಂದಿದ ಎಲ್ಲ ಫೈಲಗಳನ್ನು Pic ವಿಸ್ತರಣೆ ಹೊಂದಿದ ಫೈಲುಗಳಾಗಿ ಅದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಕಾಪಿ ಮಾಡುತ್ತದೆ. ಕಾಪಿ ಮಾಡುವಾಗ ಡಾಸ್ ಅದು ಕಾಪಿ ಮಾಡಿದ ಫೈಲ್ ಗಳ ಹೆಸರನ್ನು ತೋರಿಸುತ್ತದೆ ಮತ್ತು ಕಾಪಿ ಮಾಡುವ ಕಾರ್ಯವು ಮುಗಿದ ನಂತರ ಅದು ಕಾಪಿ ಮಾಡಿದ ಫೈಲ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ತೋರಿಸುತ್ತದೆ.

3 C:\COPA\DOS\PRACT_3>**COPY *. Pic**

© NIMI NOT TO BE REPUBLISHED

C:\COPA\DOS\PRACT_3

C:\COPA\DOS\PRACT_3 ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ Pic ವಿಸ್ತರಣೆ ಹೊಂದಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು C:\COPA\DOS\PRACT_4 ಡೈರೆಕ್ಟರಿಗೆ ಅದೇ ಹೆಸರು ಮತ್ತು ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ಕಾಪಿ ಮಾಡಲಾಗುತ್ತದೆ.

ಕಾಪಿ ಆಜ್ಞೆಯ ನಂತರ *.* ನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ ಎಲ್ಲ ಫೈಲ್‌ಗಳು ಎಲ್ಲ ವಿಸ್ತರಣೆಗಳೊಂದಿಗೆ destination ಗೆ ಕಾಪಿ ಆಗುತ್ತವೆ.

ಕಾಪಿ ಕಂಕ್ಯಾಟಿನೇಟಿಂಗ್ (Copy Concatenating) : ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳನ್ನು ಒಂದೇ ಫೈಲನ್ನಾಗಿ ಸಂಯೋಜಿಸುವುದನ್ನು ಕಂಕ್ಯಾಟಿನೇಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು source ಫೈಲ್‌ಗಳ ಮಧ್ಯೆ + ಚಿನ್ಹೆ ಉಪಯೋಗಿಸಿ ಫೈಲ್‌ಗಳನ್ನು ಒಗ್ಗೂಡಿಸಬಹುದು.

ಉದಾಹರಣೆ 4 (Example 4) **C:\COPA\DOS\PRACT_3>COPY TEST2.txt + TRIAL.txt CONCAT1.txt**

ಈ ಮೇಲಿನ ಆಜ್ಞೆಯು TEST2.txt ಮತ್ತು TRIAL.txt ಎರಡೂ ಫೈಲ್‌ಗಳನ್ನು ಸಂಯೋಜಿಸುತ್ತದೆ ಮತ್ತು CONCAT1.txt ಎನ್ನುವ ಫೈಲ್ ಅನ್ನು ತಯಾರಿಸುತ್ತದೆ. ಈ ಫೈಲ್ ಎರಡೂ source ಫೈಲ್‌ಗಳಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಫೈಲ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು (Deleting Files) :

DEL ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಫೈಲ್‌ಗಳನ್ನು ಆಳಿಸುತ್ತದೆ (Delete)

or

ERASE

Syntax

DEL C:\path/filename[/P]

ERASE C:\path/filename[/P]

ಇಲ್ಲಿ

- **C:** ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ ಫೈಲ್ ಇರುವ ಡಿಸ್ಕ ಡ್ರೈವ್
- **PATH** ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ ಫೈಲ್‌ನ ನಿರ್ದಿಷ್ಟ ಸ್ಥಳವನ್ನು ತೋರಿಸುತ್ತದೆ.
- filename1 ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದµà (delete) ಫೈಲ್‌ನ ಹೆಸರು

Switch

/P /P ಸ್ವಿಚ್ ಅನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಫೈಲ್ ಅನ್ನು ಡಿಲೀಟ್ ಮಾಡುವ ಮೊದಲು ದೃಢೀಕರಣಕ್ಕಾಗಿ ಕೇಳುತ್ತದೆ.

/P ಸ್ವಿಚ್ ಅನ್ನು ಉಪಯೋಗಿಸಿದರೆ, DEL ಆಜ್ಞೆಯು ಫೈಲ್‌ನ ಹೆಸರನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ ಮತ್ತು ಈ ಕೆಳಗಿನಂತೆ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ.

```
Filename, Delete (Y/N) ?
```

ಫೈಲನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು Y ಅನ್ನು ಒತ್ತಿ , ಮತ್ತು ರದ್ದುಗೊಳಿಸಲು N ಅನ್ನು ಒತ್ತಿ. ಮತ್ತು ಮುಂದಿನ ಫೈಲ್ ಹೆಸರನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ (ನಿರ್ದಿಷ್ಟ ಫೈಲ್‌ಗಳ ಗುಂಪು ಇದ್ದರೆ) ಅಥವಾ DEL ಆಜ್ಞೆಯನ್ನು ನಿಲ್ಲಿಸಲು ಕಂಟ್ರೋಲ್ +C ಒತ್ತಿರಿ.

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes)

- ಡ್ರೈವ್‌ನ ಹೆಸರನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ, ಪ್ರಸ್ತುತ ಡಿಸ್ಕ ಡ್ರೈವನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ಪಾಥ್ ಅನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ ಫೈಲ್‌ನ ನಿಖರವಾದ ಹೆಸರನ್ನು ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ನೀಡಬೇಕು.
- ಫೈಲ್‌ನ ಸೂಕ್ತವಾದ ಹೊಸ ಹೆಸರನ್ನು ಸರಿಯಾದ ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ನೀಡಬೇಕು.
- ಅಗತ್ಯವಿರುವ ಫೈಲ್‌ಗಳ ಗುಂಪನ್ನು ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದರೆ ವೈಲ್ಡಕಾರ್ಡ್ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.
- ಡಿಲೀಟ್ ಮಾಡುವುದರಿಂದ ಫೈಲ್ ಹೆಸರುಗಳನ್ನು ಡೈರೆಕ್ಟರಿಯಿಂದ ತೆಗೆದು ಹಾಕಲಾಗುತ್ತದೆ.
- ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು DEL ಆಜ್ಞೆಯ ನಂತರ [drive:] path ಟೈಪ್ ಮಾಡುವುದರಿಂದ ಡಿಲೀಟ್ ಮಾಡಬಹುದು.ಒಂದು ಸಮಯದಲ್ಲಿ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ವೈಲ್ಡಕಾರ್ಡ (* ಮತ್ತು ?)ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಅದಾಗ್ಯೂ ವೈಲ್ಡಕಾರ್ಡಗಳನ್ನು ಅನುದ್ದೇಶಪೂರ್ವಕವಾಗಿ ಫೈಲ್‌ಗಳು ಡಿಲೀಟ್ ಆಗದಂತೆ DEL ಆಜ್ಞೆಯೊಂದಿಗೆ ಎಚ್ಚರಿಕೆಯಿಂದ ಉಪಯೋಗಿಸಬೇಕು.

ಎಲ್ಲಾ ಫೈಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ಈ ಕೆಳಗಿನಂತೆ ಆಜ್ಞೆಯನ್ನು ನೀಡಬೇಕು.

del *.*

DEL ಈ ಕೆಳಗಿನಂತೆ ಪ್ರಾಂಪ್ಟನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲಾಗುತ್ತದೆ! ನೀವು ಖಚಿತವಾಗಿದ್ದೀರಾ (ಹೌದು/ಇಲ್ಲ)?

© NIMI NOT TO BE REPUBLISHED

All files in directory will be deleted ! Are you sure (Y/N) ?

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು Y ಒತ್ತಿ ನಂತರ ENTER ಕೀಲಿಯನ್ನು ಒತ್ತಿರಿ

ಡಿಲೀಟ್ ಮಾಡುವುದನ್ನು ರದ್ದುಗೊಳಿಸಲು N ಒತ್ತಿರಿ ನಂತರ ENTER ಕೀಲಿಯನ್ನು ಒತ್ತಿರಿ.

- DEL ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಡೈರೆಕ್ಟರಿಯನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ಬರುವುದಿಲ್ಲ. ಡೈರೆಕ್ಟರಿಯನ್ನು ತೆಗೆದುಹಾಕಲು ಪ್ರತ್ಯೇಕ ಆಜ್ಞೆಯು ಲಭ್ಯವಿದೆ.
- ಒಮ್ಮೆ ಫೈಲ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡಿದರೆ ಅದರ ಮೆಮೊರಿ ಸ್ಥಳವನ್ನು ಹೊಸ ಫೈಲ್ ಆಕ್ರಮಿಸಿಕೊಂಡರೆ, ನಂತರ ಅದನ್ನು ಮರಳಿ ಪಡೆಯಲು ಸಾಧ್ಯವಿಲ್ಲ. ಆಕಸ್ಮಿಕವಾಗಿ ಫೈಲ್‌ಗಳು ಡಿಲೀಟ್ ಆಗಿದ್ದರೆ ತಕ್ಷಣ ಯುಟಿಲಿಟಿ ಆಜ್ಞೆಯನ್ನು ಬಳಸುವ ಮೂಲಕ ಅವುಗಳನ್ನು ಮರಳಿ ಪಡೆಯಬಹುದು.
- ಡಿಸ್ಕನಲ್ಲಿ ಅಳಿಸಲಾದ ಫೈಲ್ ಆಕ್ರಮಿಸಿರುವ ಸ್ಥಳವು ಮುಕ್ತವಾಗುತ್ತದೆ.
- ಡಿಲೀಟ್ ಆಜ್ಞೆಯನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಲು, ENTER ಕೀಲಿಯನ್ನು ಒತ್ತುವ ಮೊದಲು ಫೈಲ್‌ನ ಹೆಸರಿನಲ್ಲಿರುವ ಮುದ್ರಣದ ದೋಷಗಳನ್ನು ಪರಿಶೀಲಿಸಿ.

ಉದಾಹರಣೆಗಳು (Examples)

1 C:\COPA\DOS\PRACT_3>**DEL TEST2.txt**

ಮೇಲಿನ ಆಜ್ಞೆಯು, C:\COPA\DOS\PRACT_3 ಈ ಡೈರೆಕ್ಟರಿಯಿಂದ TEST2.txt ಎನ್ನುವ ಫೈಲ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ. ಡೈರೆಕ್ಟರಿಗಳ ಪಟ್ಟಿ ಮಾಡುವುದರಿಂದ TEST2.txt ಫೈಲ್ ಕಂಡು ಬರುವುದಿಲ್ಲ.

2 C:\COPA\DOS\PRACT_4>**DEL *.txt**

ಮೇಲಿನ ಆಜ್ಞೆಯು, C:\COPA\DOS\PRACT_4 ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ txt ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿದ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ.

3 C:\COPA\DOS\PRACT_3\TEMP>**DEL *.***

C:\COPA\DOS\PRACT_3\TEMP ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಯಾವುದೇ ವಿಸ್ತರಣೆಯನ್ನು ಹೊಂದಿದ ಎಲ್ಲ ಫೈಲ್‌ಗಳು ಡಿಲೀಟ್ ಆಗುತ್ತವೆ.

ಡಿಲೀಟ್ ಮಾಡಿದ ಫೈಲ್‌ಗಳನ್ನು ಮರಳಿ ಪಡೆಯುವುದು (Recovering deleted files):

UNDELETE ಡಿಲೀಟ್ ಆದ ಫೈಲನ್ನು ಮರಳಿಪಡೆಯುವ ಸೌಲಭ್ಯವನ್ನು ಒದಗಿಸುತ್ತದೆ.

Syntax

UNDELETE C:path/filename[/DT | /DS | /DOS]
UNDELETE [/LIST | /ALL | /PURGE[DRIVE] | /

STATUS | /LOAD | /UNLOAD

/UNLOAD | /S[DRIVE] | /T[DRIVE]-entrys]]

ಇಲ್ಲಿ,

- **C:** ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ ಫೈಲ್ ಇರುವ ಡಿಸ್ಕ ಡ್ರೈವ್
- **PATH** ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ ಫೈಲ್‌ನ ನಿರ್ದಿಷ್ಟ ಸ್ಥಳವನ್ನು ತೋರಿಸುತ್ತದೆ.
- **filename** ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಬೇಕಾದ **(undelete)** ಫೈಲ್‌ನ ಹೆಸರು

Switches

/LIST	ಮರಳಿಪಡೆಯಬಹುದಾದ ಡಿಲೀಟ್ ಆದ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
/ALL	ದೃಢೀಕರಣ ಕೇಳದೇ ಫೈಲ್‌ಗಳನ್ನು ಮರಳಿಸುತ್ತದೆ.
/DOS	ಎಮ್ ಎಸ್ ಡಾಸ್ ನಿಂದ ಡಿಲೀಟ್ ಮಾಡಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಮರಳಿಸುತ್ತದೆ
/DT	ಡಿಲೀಟ್ ಟ್ರ್ಯಾಕರ್‌ನಿಂದ ರಕ್ಷಿಸಲ್ಪಟ್ಟ ಫೈಲ್‌ಗಳನ್ನು ಮರಳಿಸುತ್ತದೆ
/DS	ಡಿಲೀಟ್ ಸೆಂಟ್ರನಿಂದ ರಕ್ಷಿಸಲ್ಪಟ್ಟ ಫೈಲ್ ಗಳನ್ನು ಮರಳಿಸುತ್ತದೆ
/LOAD	ಡಿಲೀಟ್‌ನಿಂದ ರಕ್ಷಿಸಲು ಅನ್ ಡಿಲೀಟನ್ನು ಮೆಮೊರಿಯಲ್ಲಿ ಲೋಡ್ ಮಾಡುತ್ತದೆ.
/UNLOAD	ಅನ್‌ಡಿಲೀಟನ್ನು ಮೆಮೊರಿಯಿಂದ ಅನ್ಲೋಡ್ ಮಾಡುತ್ತದೆ
/PURGE[DRIVE]	ಡಿಲೀಟ್ ಸೆಂಟ್ರ ಡೈರೆಕ್ಟರಿಯಿಂದ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಪರ್ಜ್ ಮಾಡುತ್ತದೆ.
/STATUS	ಪ್ರತಿ ಡ್ರೈವ್‌ಗೆ ರಕ್ಷಣಾ ವಿಧಾನವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.
/S[DRIVE]	ರಕ್ಷಣೆಯ ಡಿಲೀಟ್ ಸೆಂಟ್ರೇ ವಿಧಾನವನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ.
/T[DRIVE]-entrys]]	ರಕ್ಷಣೆಯ ಡಿಲೀಟ್ ಟ್ರ್ಯಾಕಿಂಗ್ ವಿಧಾನವನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

ಪ್ರಮುಖ ಟಿಪ್ಪಣಿಗಳು (Important Notes) :

ಒಮ್ಮೆ ಫೈಲನ್ನು ಡಿಸ್ಕನಿಂದ ಡಿಲೀಟ್ ಮಾಡಿದ ನಂತರ ಅದನ್ನು ಮತ್ತೆ ಅದನ್ನು ಮರಳಿ ಪಡೆಯಲು ಸಾಧ್ಯವಾಗದಿರಬಹುದು. ಡಿಲೀಟ್ ಮಾಡಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಅನ್ ಡಿಲೀಟ್ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಮರಳಿ ಪಡೆಯಬಹುದು. ಬೇರೆ ಯಾವುದೇ ಫೈಲ್‌ಗಳು ಡಿಸ್ಕನಲ್ಲಿ ತಯಾರಾಗಿರುವುದಿಲ್ಲ ಅಥವಾ ಬದಲಾಯಿಸಲ್ಪಟ್ಟಿರುವುದಿಲ್ಲ ಎಂಬುದು ಖಚಿತವಾದ ನಂತರವೇ ಇದನ್ನು ಮಾಡಬಹುದು. ಫೈಲ್ ಆಕಸ್ಮಿಕವಾಗಿ ಡಿಲೀಟ್ ಆಗಿದ್ದರೆ, ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ನೀವು ಮಾಡುತ್ತಿರುವ ಬೇರೆ ಎಲ್ಲ ಕೆಲಸಗಳನ್ನು ನಿಲ್ಲಿಸಿ ಮತ್ತು ಫೈಲನ್ನು ಮರಳಿ ಪಡೆಯಲು ತಕ್ಷಣ ಅನ್‌ಡಿಲೀಟ್ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ (Example)

1 C:\COPA\DOS\PRACT_3>UN**DELETE TEST2.txt**

ಮೇಲಿನ ಆಜ್ಞೆಯು, TEST2.txt ಎನ್ನುವ ಫೈಲ್ ಅನ್ನು ಮರಳಿ ತರುತ್ತದೆ. C:\COPA\DOS\PRACT_3 ಈ ಡೈರೆಕ್ಟರಿಯಿಂದ. ಡೈರೆಕ್ಟರಿಗಳ ಪಟ್ಟಿ ಮಾಡುವುದರಿಂದ TEST2.txt ಫೈಲ್ ಆ ಪಟ್ಟಿಯಲ್ಲಿ ಲಭ್ಯವಿರುತ್ತದೆ

2 C:\COPA\DOS\PRACT_4> UN**DELETE**

ಮೇಲಿನ ಆಜ್ಞೆಯಿಂದ, ಡಿಲೀಟ್ ಆದ ಬಹಳಷ್ಟು ಫೈಲ್‌ಗಳು ಮರಳಿ ಬರುತ್ತವೆ. ಡಾಸ್ ಪ್ರತಿ ಫೈಲ್‌ನ ಮರಳುವಿಗೆ ದೃಢೀಕರಣವನ್ನು ಕೇಳುತ್ತದೆ ಮತ್ತು ಫೈಲ್‌ನ ಮೊದಲನೇ ಅಕ್ಷರವನ್ನು ಟೈಪ್ ಮಾಡಲು ಕೇಳುತ್ತದೆ. ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಿದ ನಂತರ ಮತ್ತು . C:\COPA\DOS\PRACT_4 ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿದ ನಂತರ ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಿದ ಫೈಲ್‌ಗಳ ಹೆಸರನ್ನು ನಾವು ಪಟ್ಟಿಯಲ್ಲಿ ನೋಡಬಹುದು.

3 C:\COPA\DOS\PRACT_3\TEMP>UN**DELETE/ALL**

ಮೇಲಿನ ಆಜ್ಞೆಯಿಂದ, ಡಿಲೀಟ್ ಆದ ಬಹಳಷ್ಟು ಫೈಲ್‌ಗಳು ಮರಳಿ ಬರುತ್ತವೆ. ಡಾಸ್ ಪ್ರತಿ ಫೈಲ್‌ನ ಮರಳುವಿಗೆ ದೃಢೀಕರಣವನ್ನು ಕೇಳುವುದಿಲ್ಲ. ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಿದ ನಂತರ ಮತ್ತು . C:\COPA\DOS\PRACT_4 ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿದ ನಂತರ ಅನ್‌ಡಿಲೀಟ್ ಮಾಡಿದ ಫೈಲ್‌ಗಳ ಹೆಸರನ್ನು ನಾವು ಪಟ್ಟಿಯಲ್ಲಿ ನೋಡಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಉಬುಂಟು ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು ಮತ್ತು ಲಿನಕ್ಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಕಾರ್ಯಗತ ಗೊಳಿಸುವುದು (Install Ubuntu Linux operating system and execute basic Linux commands)

ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಪರಿಚಯ (Introduction to Linux operating system)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- ಲಿನಕ್ಸ್‌ನ ಅವಲೋಕನ
- ಲಿನಕ್ಸ್‌ನ ಭವಿಷ್ಯವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ
- ಲಿನಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಪ್ರದೇಶಗಳನ್ನು ವಿವರಿಸಿ
- ಕರ್ನಲ್ ಬಗ್ಗೆ ವಿವರಿಸಿ.

ಲಿನಕ್ಸ್ ನ ಅವಲೋಕನ (overview of Linux)

ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್(Operating System)

ಡೆವಲಪರ್‌ಗಳಿಗೆ ಕರ್ನಲ್‌ನೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತಹ ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ಬರೆಯಲು ವಿಶೇಷ ಪರಿಕರಗಳ (ಜಿಎನ್ ಯು ನಲ್ಲಿ ಕಂಡು ಬರುವ ಕಂಪೈಲರ್‌ಗಳು ಮತ್ತು ಕಮಾಂಡ್ ಲೈನ್‌ಗಳಂತಹ) ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ಬರೆದು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿದ ನಂತರ ಹೊರಗಿನ ಅಪ್ಲಿಕೇಶನ್‌ಗಳಿಗೆ ಸುಲಭವಾಗಿ ಕರ್ನಲ್ ಪ್ರವೇಶಿಸಲು ಅವರಿಗೆ ಪರಿಕರಗಳು ಮತ್ತು ಅಪ್ಲಿಕೇಶನ್‌ಗಳ ಅವಶ್ಯಕತೆ ಇದೆ.

ಕರ್ನಲ್‌ನೊಂದಿಗೆ ಸಂಯೋಜಿಸಿದ, ಈ ಸಾಮೂಹಿಕ ಸಾಧನಗಳ ಗುಂಪಿಗೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಒಬ್ಬ ಸಾಮಾನ್ಯ ಬಳಕೆದಾರ ಪ್ರವೇಶಿಸಬಹುದಾದ ಕಂಪ್ಯೂಟರ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಅತ್ಯಂತ ಕೆಳಗಿನ ಪದರವಾಗಿದೆ. ಸಾಮಾನ್ಯ ಬಳಕೆದಾರರು ಕಮಾಂಡ್ ಲೈ ನ್ ಪ್ರವೇಶಿಸಿದಾಗ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ.

ಲಿನಕ್ಸ್ ,ಈ ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ಬರೆಯಲು ಪ್ರಬಲ ಸಾಧನಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. : ಡೆವಲಪರ್ ಪರಿಸರಗಳು, ಸಂಪಾದಕರು, ಮತ್ತು ಕಂಪೈಲರ್‌ಗಳು. ಇವುಗಳು ಡೆವಲಪರ್ ಕೋಡ್‌ಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ ಮತ್ತು ಕರ್ನಲ್‌ನ್ನು ಪ್ರವೇಶಿಸಬಹುದಾದ ಯಾವುದಕ್ಕೂ ಪರಿವರ್ತಿಸುತ್ತದೆ ಮತ್ತು ಕಾರ್ಯ ನಿರ್ವಹಿಸುವಂತೆ ಮಾಡುತ್ತದೆ.

ಕರ್ನಲ್‌ನಂತೆ , ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಕೂಡ ಮಾಡ್ಯುಲರ್ ಆಗಿದೆ. ಡೆವಲಪರ್‌ಗಳು ಆಪರೇಟಿಂಗ್ ಸಾಧನಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಬಹುದು. ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಬಳಕೆದಾರರಿಗೆ ಮತ್ತು ಡೆವಲಪರ್‌ಗಳಿಗೆ ಹೊಸ ಹೊಳಪಿನ ಲಿನಕ್ಸ್‌ನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

ಲಿನಕ್ಸ್ ಪರಿಚಯ(Introduction to Linux)

ಲಿನಕ್ಸ್ (ಲಿಹ್ ನಕ್ಸ್ ಎಂದು ಉಚ್ಚರಿಸಲಾಗುತ್ತದೆ) ಇದು ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಹಾಗೂ ವಿವಿಧ ಕಂಪ್ಯೂಟರ್ ಗಳಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಆದರೂ ಬಹಳಷ್ಟು ಜನರು ಲಿನಕ್ಸನ್ನು ಸಾಫ್ಟ್‌ವೇರನ್ನು ಒಳಗೊಂಡಿರುವ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಎಂದು ತಿಳಿದಿರಬಹುದು ಆದರೆ ಲಿನಕ್ಸ್ ಇದು ಇತರೆ ಸಾಫ್ಟ್‌ವೇರನ್ನು ವಿತರಿಸುವ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಕರ್ನಲ್.

ಲಿನಕ್ಸ್ ಅನ್ನು ಮೊದಲ ಬಾರಿಗೆ 1991 ರಲ್ಲಿ ಅದರ ಲೇಖಕರಾದ ಲಿನಸ್ ಟೊರ್ವಾಲ್ಡ್ಸ್ ಇವರು ಹೆಲ್ಸಿಂಕಿ ಯುನಿವರ್ಸಿಟಿಯಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡಿದರು. ಅಂದಿನಿಂದ ಜಗತ್ತಿನ ಎಲ್ಲ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳಿಗೆ ಇದು ಜನಪ್ರೀಯವಾಗಿ ಬೆಳೆಯಿತು. ಉಚಿತ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಅನ್ನು ತಯಾರಿಸುವುದು, ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಹರಿಸುವುದು ಮುಂತಾದ ಯೋಜನೆಗಳಿಗಾಗಿ ಜನಪ್ರೀಯವಾಗಿದೆ.

ಇಂದಿನ ಪೀಳಿಗೆಯ ಕಂಪ್ಯೂಟರ್ ಬಳಕೆದಾರರಲ್ಲಿ ಲಿನಕ್ಸ್ ಜನಪ್ರೀಯವಾಗಿದೆ. 20 ವರ್ಷಗಳಿಗಿಂತಲೂ ಹೆಚ್ಚು ಕಾಲ ಯುನಿಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಹಳೆಯ ಆವೃತ್ತಿಗಳು ಇದೇ ಕಾರಣಗಳಿಗಾಗಿ ಜನರನ್ನು ಆಕರ್ಷಿಸಿದ್ದವು. ಲಿನಕ್ಸ್ ಪೋರ್ಟೇಬಲ್ ಆಗಿದೆ, ಎಂದರೆ ಇದರ ಆವೃತ್ತಿಗಳನ್ನು ಹೆಸರಿನ ಬ್ರಾಂಡ್‌ಗಳು ಅಥವಾ ಕ್ಲೋನ್ ಪಿಸಿಗಳು, ಆ್ಯಪಲ್ ಮ್ಯಾಸಿಂತೋಶ್, ಸನ್ ವರ್ಕಸ್ಟೇಷನ್‌ಗಳು ಅಥವಾ ಡಿಜಿಟಲ್ ಇಕ್ವಿಪ್ಮೆಂಟ್ ಕಾರ್ಪೋರೇಷನ್ ಅಲ್ಫಾ ಬೇಸಡ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ರನ್ ಮಾಡಬಹುದು. ಲಿನಕ್ಸ್ ಸೋರ್ಸ್ ಕೋಡ್‌ನೊಂದಿಗೆ ಬರುತ್ತದೆ. ಅದ್ದರಿಂದ ನೀವು ನಿಮ್ಮ ಅವಶ್ಯಕತೆಗಳಿಗೆ ತಕ್ಕಂತೆ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಬದಲಾಯಿಸಬಹುದು ಅಥವಾ ಕಸ್ಟಮೈಸ್ ಮಾಡಬಹುದು. ಅಂತಿಮವಾಗಿ ಲಿನಕ್ಸ್ ಉತ್ತಮ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಇದು ಯುನಿಕ್ಸ್‌ನ ಇತರ ಆವೃತ್ತಿಗಳಿಂದ ಅಳವಡಿಸಿಕೊಂಡ ವೈಶಿಷ್ಟ್ಯಗಳಿಂದ ಸಮೃದ್ಧವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಲಿನಕ್ಸ್ ಎಲ್ಲಿದೆ? (Where is Linux ?)

ಲಿನಕ್ಸ್ ನ ಅತ್ಯಂತ ಗಮನಾರ್ಹ ಗುಣಲಕ್ಷಣವೆಂದರೆ ಇದನ್ನು ಎಲ್ಲಿ ಬಳಸಲಾಗುತ್ತದೆ. ವಿಂಡೋಸ್ ಮತ್ತು ಓಎಸ್ ಎಕ್ಸ್ ಪ್ರಧಾನವಾಗಿ ವೈಯಕ್ತಿಕ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಾಧನಗಳಾದ ಡೆಸ್ಕಟಾಪ್ ಮತ್ತು ಲಾಪ್‌ಟಾಪ್‌ಗಳಲ್ಲಿ ಕಂಡುಬರುತ್ತವೆ. ಸಿಂಬಿಯಾನ್‌ನಂತಹ ಇತರ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಗಳು ಫೋನ್‌ಗಳಲ್ಲಿ ಮತ್ತು ಪಿಡಿಎಗಳಂತಹ ಸಣ್ಣ ಸಾಧನಗಳಲ್ಲಿ ಕಂಡು ಬರುತ್ತವೆ. ಮೇನ್ ಫ್ರೇಮ್ ಮತ್ತು ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್ ಗಳಲ್ಲಿ ಪ್ರಮುಖ ಶೈಕ್ಷಣಿಕ ಮತ್ತು ಸಂಸ್ಥೆಯ ಲ್ಯಾಬ್‌ಗಳಲ್ಲಿ ವಿಶೇಷ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಾದ ಎಎಸ್/400 ಮತ್ತು ಕ್ರೇ ಓಎಸ್ ಗಳನ್ನು (cray OS) ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಲಿನಕ್ಸ್ ಸರ್ವರ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿ ತನ್ನ ಅಸ್ತಿತ್ವವನ್ನು ಪ್ರಾರಂಭಿಸಿತು. ಮತ್ತು ಡೆಸ್ಕಟಾಪ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿ ಉಪಯುಕ್ತವಾಗಿದೆ. ಇದನ್ನು ಕೈಗಡಿಯಾರಗಳಿಂದ ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ಗಳವರೆಗೆ ಎಲ್ಲ ಸಾಧನಗಳಲ್ಲಿ ಸಹ ಇದನ್ನು ಉಪಯೊಗಿಸಬಹುದು.

ಲಿನಕ್ಸ್ ನ ಭವಿಷ್ಯ (The future of Linux)

ಲಿನಕ್ಸ್ ಈಗಾಗಲೇ ಹಲವು ಬಗೆಯ ಸಾಧನಗಳಲ್ಲಿ ಯಶಸ್ವಿಯಾಗಿದೆ. ಆದರೆ ಇನ್ನೂ ಅನೇಕ ತಾಂತ್ರಿಕ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಲಿನಕ್ಸ್ ಮುನ್ನಡೆಯುತ್ತಿದೆ. ಡೆಸ್ಕಟಾಪ್ ಮತ್ತು ಸರ್ವರ್ ಅಭಿವೃದ್ಧಿಯಂತೆ ಇವತ್ತಿನ ಯಾವುದೇ ಇತರ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಿಂತ ವೇಗವಾಗಿ ಬೆಳೆಯುತ್ತಿದೆ.

ಲಿನಕ್ಸ್ ನ್ನು ಲ್ಯಾಪ್‌ಟಾಪ್‌ನ ಸಿಸ್ಟಮ್ ಬಯಾಸ್ ಮತ್ತು ನೋಟ್‌ಬುಕ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾಗುತ್ತಿದೆ. ಇದು ಬಳಕೆದಾರರು ತಮ್ಮ ಸಾಧನಗಳನ್ನು ಸುವ್ಯವಸ್ಥಿತ ಲಿ ನಕ್ಸ್ ಪರಿಸರದಲ್ಲಿ ಸೆಕೆಂಡುಗಳಲ್ಲಿ ಆನ್ ಮಾಡಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ. ಈ ಪರಿಸರವು ಅಂತರ್ಜಾಲ ಸಂಪರ್ಕ ಸಾಧನಗಳಾದ ವೆಬ್ ಬ್ರೌಸರ್ ಮತ್ತು ಇ ಮೇಲ್ ಕ್ಲೈಂಟ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಅದು ವಿಂಡೋಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿದ್ದರೂ ಸಹ ಪ್ರಾಥಮಿಕ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಬೂಟ್ ಮಾಡದೆ ಇದು ಬಳಕೆದಾರರು ಇಂಟರ್ನೆಟ್‌ನಲ್ಲಿ ಕೆಲಸ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ.

ಅದೇ ಸಮಯದಲ್ಲಿ, ಲಿನಕ್ಸ್‌ನ್ನು ಮೊಬೈಲ್ ಇಂಟರ್‌ನೆಟ್ ಡಿವೈಸಸ್ (MDIs) ಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಎಂಬೆಡೆಡ್ ಸಾಧನಗಳಾದ ಸ್ಮಾರ್ಟ್ ಫೋನ್‌ಗಳು ಮತ್ತು ಪಿಡಿಎಗಳು ಮತ್ತು ನೆಟ್ಬುಕ್ ಸಾಧನಗಳು, ಸಣ್ಣ ಲ್ಯಾಪ್‌ಟಾಪ್‌ನಂತಹ ಯಂತ್ರಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಬೆಳವಣಿಗೆಯು ಲಿನಕ್ಸ್‌ಗೆ ಸಹಜವಾಗಿ ಹೊಂದಿಕೊಳ್ಳುತ್ತದೆ. ಇದು ಈಗಾಗಲೇ ಇಂಟರ್‌ನೆಟ್‌ನ ಅನೇಕ ವೆಬ್ ಬ್ರೌಸರ್ಗಳನ್ನು ರನ್ ಮಾಡುತ್ತದೆ. ಲಿನಕ್ಸ್ ಅಮೇಜಾನ್‌ನ A3 ನಂತಹ ಕ್ಲೌಡ ಸೇವೆಗಳನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ಲಿನಕ್ಸ್ ಆನ್ ಲೈ ನ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಬಳಕೆದಾರರಿಗೆ ತಲುಪಿಸಲು ಉತ್ತಮ ಸಾಮರ್ಥ್ಯದೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನಲ್ಲಿ ಲಿನಕ್ಸ್‌ನ ಬೆಳವಣಿಗೆಗೆ ಸಂಬಂಧಿಸಿದಂತೆ, ಸೂಪರ್ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಉನ್ನತ ಕಾರ್ಯಕ್ಷಮತೆಯ ಕಂಪ್ಯೂಟಿಂಗ್ (HPC) ಮತ್ತು ಹೆಚ್ಚಿನ ಲಭ್ಯತೆಯ ಪ್ರದೇಶಗಳು (HA) ಎರಡರಲ್ಲೂ ಬಹಳ ಯಶಸ್ವಿಯಾಗಿದೆ. ಭೌತಶಾಸ್ತ್ರದಲ್ಲಿ ಶೈಕ್ಷಣಿಕ ಸಂಶೋಧನೆ ಮತ್ತು ಜೈವಿಕ ಇಂಜನೀಯರಿಂಗ್ ಮತ್ತು ಹಣಕಾಸು ಮತ್ತು ಶಕ್ತಿಯ ಉದ್ಯಮಗಳು ಮತ್ತು ಸಂಸ್ಥೆಗಳು ಇ ವುಗಳಿಗೆ ಗುರಿಯನ್ನು ಸಾಧಿಸಲು ವಿಶ್ವಾಸಾರ್ಹ ಮತ್ತು ಸ್ಕೇಲೆಬಲ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಶಕ್ತಿಯ ಅಗತ್ಯವಿದೆ.

ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಅನೇಕ ಜನಪ್ರೀಯ ವೆಬ್ 2.0 ಸೇವೆಗಳಾದ ಟ್ವಿಟರ್, ಲಿಂಕ್ಡ ಇನ್, ಯೊಟ್ಯೂಬ್ ಮತ್ತು ಗೂಗಲ್ ಇವೆಲ್ಲವೂ ಲಿನಕ್ಸ್‌ನ್ನು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಆಗಿ ಅವಲಂಬಿಸಿವೆ. ಭವಿಷ್ಯದಲ್ಲಿ ಹೊಸ ವೆಬ್ ಸೇವೆಗಳು ಬಂದರೆ, ಈ ಹೊಸ ತಂತ್ರಜ್ಞಾನಗಳನ್ನು ಮುನ್ನಡೆಸಲು ಲಿನಕ್ಸ್ ವೇದಿಕೆಯಾಗಲಿದೆ.

ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳ ಪ್ರಸ್ತುತ ಅಪ್ಲಿಕೇಷನ್ (Current application of Linux operating system)

ಇವತ್ತಿನ ದಿನಗಳಲ್ಲಿ ಲಿನಕ್ಸ್ ಡೆಸ್ಕಟಾಪ್ ಮಾರುಕಟ್ಟೆಗೆ ಸೇರಿಕೊಂಡಿದೆ. ಪ್ರಾರಂಭದಲ್ಲಿ ಲಿನಕ್ಸ ಡೆವಲಪರ್‌ಗಳು ನೆಟ್‌ವರ್ಕಿಂಗ್ ಮತ್ತು ಸೇವೆಗಳ ಮೇಲೆ ಕೇಂದ್ರೀಕರಿಸಿದ್ದರು. ಮತ್ತು ಆಫೀಸ್ ಅಪ್ಲಿಕೇಶನ್ ಕೊನೆಯ ತಡೆಯಾಗಿತ್ತು. ಮೈಕ್ರೋಸಾಫ್ಟ ಮಾರುಕಟ್ಟೆಯನ್ನು ಆಳುತ್ತಿದೆ ಎಂಬುದನ್ನು ಒಪ್ಪಿಕೊಳ್ಳಲು ಅವರು ಇಷ್ಟಪಡುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ಲಿನಕ್ಸ್ ವರ್ಕ್‌ಸ್ಟೇಷನ್‌ನಂತೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ಎಂದು ಸ್ವೀಕಾರಾರ್ಹ ಆಯ್ಕೆಯನ್ನಾಗಿ ಮಾಡಲು ಕಳೆದ ಎರಡು ವರ್ಷಗಳಿಂದ ಸಾಕಷ್ಟು ಪರ್ಯಾಯಗಳು ಪ್ರಾರಂಭವಾಗಿವೆ. ಇದು ಸುಲಭವಾದ ಇಂಟರ್‌ಫೇಸನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಎಮ್ ಎಸ್ ಹೊಂದಾಣಿಕೆಯ ಆಫೀಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಾದ ವರ್ಡ್‌ಪ್ರೊಸೆಸರ್, ಸ್ಪ್ರೆಡ್‌ಶೀಟ್ ಗಳು, ಪ್ರೆಸೆಂಟೇಶನ್ ಗಳು ಇತ್ಯಾದಿಗಳು. ಸರ್ವರ್ ಸರ್ವರ್ ಸೈಡ್‌ನಲ್ಲಿ , ಲಿನಕ್ಸ್ ಸ್ಥಿರ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹ ವೇದಿಕೆಯಾಗಿದೆ. ಆನ್‌ಲೈ ನ್ ಪುಸ್ತಕ ಮಳಿಗೆಗಳಿಗೆ ಪ್ರಸಿದ್ಧವಾದ ಅಮೇಜಾನ್‌ನಂತಹ ಕಂಪನಿಗಳು, ಯುಎಸ್ ಪೋಸ್ಟ ಆಫೀಸ್, ಜರ್ಮನ್ ಆರ್ಮಿ ಮತ್ತು ಇನ್ನೂ ಅನೇಕ ಕಂಪನಿಗಳಿಗೆ ಇದು ಡಾಟಬೇಸ್ ಮತ್ತು ವ್ಯಾಪಾರದ ಸೇವೆಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ವಿಶೇಷವಾಗಿ ಇಂಟರ್ನೆಟ್ ಪೂರೈಕೆದಾರರು ಮತ್ತು ಇಂಟರ್ನೆಟ್ ಸೇವೆಗಳ ಪೂರೈಕೆದಾರರು ಲಿನಕ್ಸ್ ನ್ನು ಫೈರ್‌ವಾಲ್, ಪ್ರಾಕ್ಸಿ ಮತ್ತು ವೆಬ್‌ಸರ್ವರ ಆಗಿ ಇಷ್ಟಪಟ್ಟಿದ್ದರು. ಆರಾಮದಾಯಕ ಮ್ಯಾನೇಜಮೆಂಟ್ ಸ್ಟೇಷನ್ ಅನ್ನು ಆಸ್ವಾದಿಸುವ ಪ್ರತಿ ಯುನಿಕ್ಸ್ ಸಿಸ್ಟಮ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್ ಹತ್ತಿರ ಒಂದು ಲಿನಕ್ಸ್ ಬಾಕ್ಸ್ ಇರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಪೋಸ್ಟ ಆಫೀಸ್‌ಗಳಲ್ಲಿ ಲಿನಕ್ಸ್ ಮೈಲ್‌ನ್ನು ಹುಡುಕಲು ಮುಖ್ಯ ಕೇಂದ್ರವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ದೊಡ್ಡ ಸರ್ಚ ಇ ಂಜಿನ್‌ಗಳಲ್ಲಿ ಕ್ಲಸ್ಟರ್‌ಗಳು ಇಂಟರ್‌ನೆಟ್ ಹುಡುಕಾಟಕ್ಕಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತವೆ. ಪ್ರಪಂಚದಾದ್ಯಂತ ಪ್ರತಿದಿನ ನಡೆಯುವ ಸಾವಿರಾರು ಹೆವಿ ಡ್ಯೂಟಿ ಕೆಲಸಗಳಲ್ಲಿ ಇವು ಲಿನಕ್ಸ್ ನಿರ್ವಹಿಸುವ ಕೆಲವೆ ಕೆಲವು ಕಾರ್ಯಗಳಾಗಿವೆ. ಅಧುನಿಕ ಲಿನಕ್ಸ್, ವರ್ಕ್‌ಸ್ಟೇಷನ್, ಮಧ್ಯಮ, ಮತ್ತು ಉನ್ನತ ಮಟ್ಟದ ಸರ್ವರ್‌ಗಳಲ್ಲಿ, ಮಾತ್ರವಲ್ಲದೇ ಪಿಡಿಎ, ಮೊಬೈಲ್‌ಗಳು, ಎಂಬೆಡೆಡ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಮತ್ತು ಪ್ರಾಯೋಗಿಕ ಕೈಗಡಿಯಾರಗಳು ಇಂತಹ ಗ್ಯಾಜೆಟ್‌ಗಳಲ್ಲಿಯೂ ಕೂಡ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು ಲಿನಕ್ಸ್‌ನ್ನು ವ್ಯಾಪಕವಾಗಿ ಹಾರ್ಡವೇರನ್ನು ಒಳಗೊಂಡಿರುವ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನಾಗಿ ಮಾಡುತ್ತದೆ.

ಕೋಡ್ (The Code)

ಬೇರೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗಿಂತ ಲಿನಕ್ಸ್ ವಿಶಿಷ್ಟವಾಗಿದೆ. ಇ ದಕ್ಕೆ ಒಬ್ಬರೆ ಮಾಲೀಕರು ಇಲ್ಲ. ಟೋರ್ವಾಲ್ಡ್ಸ್ ಲಿನಕ್ಸ್ ಕರ್ನಲ್‌ನ್ನು ಇನ್ನೂ ಅಭಿವೃದ್ಧಿಪಡಿಸುತ್ತಿದ್ದಾರೆ. ಆದರೆ ವಾಣಿಜ್ಯ ಮತ್ತು ಖಾಸಗಿ ಡೆವಲಪರ್‌ಗಳು ಸಂಪೂರ್ಣ ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮನ್ನು ತಯಾರಿಸಲು ಬೇರೆ ಸಾಫ್ಟವೇರನ ಕೊಡುಗೆ ನೀಡಿದ್ದಾರೆ.

ಕರ್ನಲ್ (The Kernel)

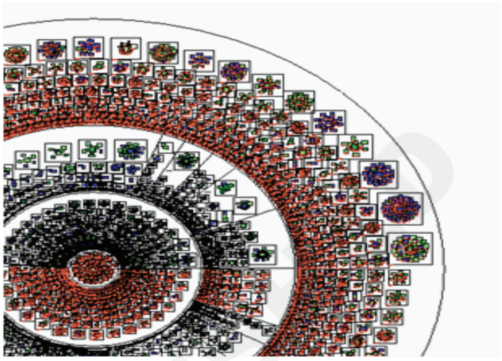
ಎಲ್ಲ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು ಕರ್ನಲ್‌ನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇದನ್ನು ರಚನೆಯ ರೂಪಕದ ಸುತ್ತ ನಿರ್ಮಿಸಲಾಗಿದೆ. ಇದರಲ್ಲಿ ಸಾಧನದ ಹಾರ್ಡವೇರನ್ನು ನಿರ್ದೇಶಿಸುವ ಸೂಚನೆಗಳ ಗುಂಪು ಇರುತ್ತದೆ. ಈ ಸೂಚನೆಗಳು ,ಕಾರ್ಯನಿರ್ವಹಿಸುವ ವಿವಿಧ ಮಾಡ್ಯುಲರ್ ಪದರಗಳಿಂದ ಸುತ್ತುವರೆದಿವೆ. ಲಿನಕ್ಸ್ ಕರ್ನಲ್ ಪ್ರಕೃತಿ ಕೂಡ ಮಾಡ್ಯುಲರ್ ಆಗಿರುವದರಿಂದ ಇದು ಏಕಮಾತ್ರ ಮತ್ತು ಹೊಂದಿಕೊಳ್ಳುವಂಥದ್ದಾಗಿದೆ.

ಮಾಡ್ಯುಲಾರಿಟಿ ಅಪೇಕ್ಷಣೀಯವಾಗಿದೆ ಏಕೆಂದರೆ ಇದು ಡೆವಲಪರ್‌ಗಳಿಗೆ ಅನವಶ್ಯಕವಾದ ಕರ್ನಲ್‌ನ ಭಾಗಗಳನ್ನು ತ್ಯಜಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಒಂದು ಸಣ್ಣ ಕರ್ನಲ್ ವೇಗವಾದ ಕರ್ನಲ್ ಆಗಿದೆ ಏಕೆಂದರೆ ಅದಕ್ಕೆ ಅವಶ್ಯಕ ಇಲ್ಲದ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಅದು ರನ್ ಮಾಡುವುದಿಲ್ಲ.

ಸಾಧನಗಳ ಡೆವಲಪರ್ ಒಂದು ಫೋನ್‌ನಲ್ಲಿ ಲಿನಕ್ಸ್‌ನ ಒಂದು ಆವೃತ್ತಿಯನ್ನು ರನ್ ಮಾಡಲು ಬಯಸಿದರೆ, ಅವಳಿಗೆ ಡಿಸ್ಕ ಡ್ರೈವ್, ಈಥರ್ ನೆಟ್ ಸಾಧನಗಳು ಅಥವಾ ದೊಡ್ಡ ಮಾನಿಟರ್ ಸ್ಕ್ರೀನ್‌ಗಳಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಕರ್ನಲ್‌ನ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ. ಫೋನ್‌ಗಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಆಪ್ಟಿಮೈಸ್ಡ್ ಕರ್ನಲ್‌ನ್ನು ಮಾತ್ರ ಬಿಟ್ಟು ಈ ತುಂಡುಗಳನ್ನು (ಮತ್ತು ಇತರ) ಹೊರಗೆ ತೆಗೆಯಬಹುದು.

ವಿಂಡೊ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಕರ್ನಲ್ (ಮೈಕ್ರೊಸಾಫ್ಟ ಹೊರಗಿನ ಜನರಿಗೆ ಸೌಕರ್ಯಕ್ಕಾಗಿ, ಪಾವತಿಸದೆ ನೋಡಲು ಅನುಮತಿಸಿದೆ) ದೃಢವಾಗಿ ಸಂಪರ್ಕಿತ ಕೋಡ್‌ಗಳ ತುಣುಕುಗಳಾಗಿದ್ದು , ಇ ವುಗಳನ್ನು ಸುಲಭವಾಗಿ ತುಂಡುಗಳಾಗಿ ವಿಭಜಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಫೋನ್‌ನಲ್ಲಿ ವಿಂಡೊಸ್ ಕರ್ನಲ್‌ನ್ನು ಹಂತವಾಗಿ ಕುಂಠಿತಗೊಳಿಸಿ ಇರಿಸುವುದು ಕಷ್ಟಕರವಾಗಿದೆ.

Fig 1



ಈ ಮಾಡ್ಯುಲಾರಿಟಿ ಲಿನಕ್ಸ್ ನ ಯಶಸ್ಸಿಗೆ ಕಾರಣವಾಗಿದೆ. ನಿರ್ದಿಷ್ಟ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ನ ಅಗತ್ಯಗಳನ್ನು ಪೂರೈಸಲು ಮೇಲೇ ಅಥವಾ ಕೆಳಗೆ ಅಳೆಯುವ ಸಾಮರ್ಥ್ಯವು ಇತರೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗಿಂತ ದೊಡ್ಡ ಅನುಕೂಲವಾಗಿದೆ.

ಮಾಡ್ಯುಲಾರಿಟಿಯು, ಸ್ಥಿರತೆ ಮತ್ತು ಸುರಕ್ಷತೆಯ ಮೇಲೂ ಪರಿಣಾಮ ಬೀರುತ್ತದೆ. ಕರ್ನಲ್ ಕೋಡ್‌ನ ಒಂದು ತುಂಡು ಕಾರ್ಯನಿರ್ವಹಿಸಲು ವಿಫಲವಾದರೆ, ಉಳಿದ ಕರ್ನಲ್ ವಿಫಲವಾಗುವುದಿಲ್ಲ. ಅದೇ ರೀತಿ ಕರ್ನಲ್‌ನ ಒಂದು ಭಾಗದ ಮೇಲೆ ಅಕ್ರಮ ದಾಳಿಯಾದರೆ(ಅಥವಾ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಉಳಿದ ಭಾಗ) ಕೋಡ್‌ನ ಆ ಭಾಗವನ್ನು ಅಡ್ಡಿಪಡಿಸಬಹುದು, ಆದರೆ ಇಡೀ ಸಾಧನದ ಸುರಕ್ಷತೆಯ ಜೊತೆಗೆ ರಾಜಿ ಮಾಡಲಾಗುವುದಿಲ್ಲ.

ಪರಿಸರಗಳು (The Environments)

ಬಹಳ ಜನರು ವಿಂಡೊಸ್‌ನ್ನು, ಮೆನುಗಳನ್ನು ಮತ್ತು ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಗಳನ್ನು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಭಾಗ ಎಂದು ತಿಳಿ ದಿರುತ್ತಾರೆ. ಆದರೆ ಅವು ವಾಸ್ತವವಾಗಿ ಪ್ರತ್ಯೇಕ ಪದರಗಳಾಗಿವೆ. ಇ ವುಗಳನ್ನು ವಿಂಡೊಯಿಂಗ್ ಸಿಸ್ಟಮ್ ಮತ್ತು ಡೆಸ್ಕಟಾಪ್ ಪರಿಸರ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಈ ಪದರಗಳು ಮಾನವ ಆಧಾರಿತ ಗ್ರಾಫಿಕಲ್ ಯೂಸರ್ ಇ ಂಟರ್‌ಫೇಸ್ (GUI)ಅನ್ನು ಒದಗಿಸುತ್ತವೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಅಪ್ಲಿಕೇಷನ್‌ನೊಂದಿಗೆ ಸುಲಭವಾಗಿ ಕೆಲಸ ಮಾಡಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತವೆ. ಮತ್ತು ಥರ್ಡ ಪಾರ್ಟಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಲಿನಕ್ಸ್‌ನಲ್ಲಿ ,ವಿಂಡೋಯಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಾಗಿ ಮತ್ತು ಡೆಸ್ಕಟಾಪ್ ಪರಿಸರವನ್ನು ಉಪಯೋಗಿಸುವುದಕ್ಕಾಗಿ ಬಹಳಷ್ಟು ಆಯ್ಕೆಗಳಿವೆ. ಇದನ್ನು ನಿರ್ಧರಿಸಲು ಲಿನಕ್ಸ್ ಬಳಕೆದಾರರಿಗೆ ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ. ವಿಂಡೋಸ್‌ನಲ್ಲಿ ಇದು ಸಾಧ್ಯವಿಲ್ಲ,ಮತ್ತು ಓಎಸ್ ಎಕ್ಸ್‌ನಲ್ಲಿ ಇದನ್ನು ಮಾಡುವುದು ಕಷ್ಟ.

ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಮತ್ತು ಕರ್ನಲ್‌ನಂತೆ, ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪರ್‌ಗಳಿಗೆ ಈ ಪರಿಸರದಲ್ಲಿ (ಉದಾ. GNOME ಗಾಗಿ Gtk+, KDE ಗಾಗಿ Qt)ಹೆಚ್ಚು ಸುಲಭವಾಗಿ ಕೆಲಸ ಮಾಡಲು ಅನುಕೂಲವಾಗುವಂತೆ ಸಾಧನಗಳು ಮತ್ತು ಕೋಡ್ ಲೈಬ್ರರಿಗಳು ಲಭ್ಯವಿರುತ್ತವೆ.

ಅಪ್ಲಿಕೇಷನ್‌ಗಳು (The Applications)

ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು ಎರಡು ರೀತಿಯ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ :ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಅಗತ್ಯ ಅಂಶಗಳು ಮತ್ತು ಬಳಕೆದಾರರು ಅನಂತರ ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವ ಅಪ್ಲಿಕೇಶನ್‌ಗಳು.ಕ್ಲೋಸ್ಡ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಾದ ವಿಂಡೊಸ್ ಮತ್ತು ಓಎಸ್ ಎಕ್ಸ್ , ಇವು ಬಳಕೆದಾರರಿಗೆ(ಅಥವಾ ಡೆವಲಪರ್‌ಗಳಿಗೆ) ಅವರು ಬಳಸುವ ಅಗತ್ಯ ಘಟಕ ಅಪ್ಲಿಕೇಶನ್‌ಗಳನ್ನು ಆರಿಸಲು ಮತ್ತು ಆಯ್ಕೆ ಮಾಡಲು ಅನುಮತಿಸುವುದಿಲ್ಲ. ವಿಂಡೊಸ್ ಡೆವಲಪರ್‌ಗಳು ಮೈಕ್ರೊಸಾಫ್ಟ್‌ನ ಕಂಪೈಲರ್, ವಿಂಡೊಯಿಂಗ್ ಸಿಸ್ಟಮ್ ಮುಂತಾದವುಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ಲಿನಕ್ಸ್ ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪರ್‌ಗಳು ತಮ್ಮ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಹೆಚ್ಚಿನ ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದ್ದಾರೆ. ಇದರಿಂದ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ತಯಾರಿಸಲು ಹೆಚ್ಚು ಸುಲಭವಾಗುತ್ತದೆ. ಯಾವ ಲಿನಕ್ಸ್ ಘಟಕಗಳನ್ನು ಬಳಸಬೇಕು ಎಂಬುದನ್ನು ಡೆವಲಪರ್‌ಗಳು ನಿರ್ಧರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

ವಿತರಣೆಗಳು(The Distributions)

ಲಿನಕ್ಸ್ ವಿತರಣೆಯು , ಲಿನಕ್ಸ್ ಕರ್ನಲ್‌ನ ಮೇಲಿರುವ ಸಾಫ್ಟವೇರ್ (ಸಾಮಾನ್ಯವಾಗಿ ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿರುತ್ತದೆ) ಗಳ ಸಂಗ್ರಹವಾಗಿದೆ. ವಿತರಣೆಯು (ಅಥವಾ ಚಿಕ್ಕದಾಗಿ ಡಿಸ್ಟ್ರೋ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ) ಸರ್ವರ್ ಸಾಫ್ಟವೇರ್, ಸಿಸ್ಟಮ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಸಾಧನಗಳು, ಡಾಕ್ಯುಮೆಂಟೇಷನ್ ಮತ್ತು ಅನೇಕ ಡೆಸ್ಕಟಾಪ್ ಅಪ್ಲಿಕೇಷನ್ ಇವೆಲ್ಲವುಗಳನ್ನು ಸೆಂಟ್ರಲ್ ಸೆಕ್ಯೂರ್ ಸಾಫ್ಟವೇರ್ ರೆಸ್ಪಿರೇಟರಿಯಲ್ಲಿ ಬಂಡಲ್ ಮಾಡಬಹುದು. ಡಿಸ್ಟ್ರೋ ಸಾಮಾನ್ಯ ನೋಟ ಮತ್ತು ಭಾವನೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು ಸುರಕ್ಷಿತ ಮತ್ತು ಸುಲಭ ಸಾಫ್ಟವೇರ ನಿರ್ವಹಣೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯಾಚರಣೆಯ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿದೆ.

ಕೆಲವು ಜನಪ್ರಿಯ ವಿತರಣೆಗಳನ್ನು ನೋಡೋಣ.

ರೆಡ್ ಹ್ಯಾಟ್ (Red hat)

ರೆಡ್ ಹ್ಯಾಟ್ ಒಂದು ಬಿಲಿಯನ್ ಡಾಲರ್ ವಾಣಿಜ್ಯದ ಲಿನಕ್ಸ್ ಕಂಪನಿಯಾಗಿದೆ. ಇದು ಲಿನಕ್ಸ್ ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವಲ್ಲಿ ಸಾಕಷ್ಟು ಪ್ರಯತ್ನ ಮಾಡುತ್ತಿದೆ. ಇದು ನೂರಾರು ಲಿನಕ್ಸ್ ತಜ್ಞರನ್ನು ಹೊಂದಿದೆ. ಮತ್ತು ಅವರ ಅತ್ಯುತ್ತಮ ಬೆಂಬಲಕ್ಕೆ ಹೆಸರು ಪಡೆದಿದೆ. ಅವರು ತಮ್ಮ ಉತ್ಪನ್ನಗಳನ್ನು (ರೆಡ್ ಹ್ಯಾಟ್ ಎಂಟರ್‌ಪ್ರೈಸ್ ಮತ್ತು ಫೆಡೋರಾ) ಉಚಿತವಾಗಿ ನೀಡುತ್ತಾರೆ. ಇದರಲ್ಲಿ ರೆಡ್ ಹ್ಯಾಟ್ ಎಂಟರ್ಪ್ರೈಸ್ ಲಿನಕ್ಸ್ (ಆರ್‌ಎಚ್ ಇ ಎಲ್) ಬಿಡುಗಡೆಗೂ ಮೊದಲು ಚೆನ್ನಾಗಿ ಪರೀಕ್ಷಿಸಲ್ಪಟ್ಟಿದೆ ಮತ್ತು ಬಿಡುಗಡೆಯಾದ ನಂತರ ಏಳು ವರ್ಷಗಳವರೆಗೆ ಬೆಂಬಲಿಸಲ್ಪಟ್ಟಿದೆ. ಫೆಡೋರಾ ಇದು ವೇಗವಾದ ನವೀಕರಣಗಳೊಂದಿಗೆ ಆದರೆ ಬೆಂಬಲವಿಲ್ಲದ ವಿತರಣೆಯಾಗಿದೆ.

ಉಬುಂಟು (Ubuntu)

2004 ರಲ್ಲಿ ಕ್ಯಾನೊನಿಕಲ್ ಉಚಿತ ಕಾಂಪ್ಯಾಕ್ಟ ಡಿಸ್ಕಗಳನ್ನು ಉಬುಂಟು ಲಿನಕ್ಸ್‌ನೊಂದಿಗೆ ಕಳುಹಿಸಲು ಪ್ರಾರಂಭಿಸಿತು. ಮತ್ತು ಬೇಗನೆ ಮನೆಗಳಲ್ಲಿ ಉಪಯೋಗಿಸುವ ಬಳಕೆದಾರರಲ್ಲಿ (ಅನೇಕರು ಮೈಕ್ರೊಸಾಫ್ಟ ವಿಂಡೊಸ್‌ನಿಂದ ಬದಲಾಯಿಸುತ್ತಿದ್ದಾರೆ.) ಜನಪ್ರಿಯವಾಯಿತು. ಕಮಾಂಡ್ ಲೈ ನ್‌ನ ಅವಶ್ಯಕತೆ ಇಲ್ಲದೆ ಗ್ರಾಫಿಕಲ್ ಲಿನಕ್ಸ್ ಡೆಸ್ಕಟಾಪ್‌ನ್ನು ಸುಲಭವಾಗಿ ಬಳಸುವಂತೆ ಉಬುಂಟು ಇರಬೇಕೆಂದು ಕ್ಯಾನೊನಿಕಲ್ ಬಯಸುತ್ತದೆ. ಖಂಡಿತವಾಗಿ ಅವರು ಉಬುಂಟುಗೆ ಬೆಂಬಲವನ್ನು ನೀಡುವ ಮೂಲಕ ಲಾಭಗಳಿಸಲು ಬಯಸುತ್ತಾರೆ.

ಡೆಬಿಯನ್ (Debian)

ಡೆಬಿಯನ್ ಹಿಂದೆ ಯಾವುದೇ ಕಂಪನಿ ಇಲ್ಲ. ಬದಲಾಗಿ ಪ್ರತಿ ಎರಡು ವರ್ಷಗಳಿಗೊಮ್ಮೆ ಡೆಬಿಯನ್ ಪ್ರಾಜೆಕ್ಟ್ ಲೀಡರನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ಸಾವಿರಾರು ಸುಸಂಘಟಿತ ಡೆವಲಪರ್‌ಗಳು ಇದ್ದಾರೆ. ಡೆಬಿಯನ್ ಅತ್ಯಂತ ಸ್ಥಿರವಾದ ಲಿನಕ್ಸ್ ವಿತರಣೆಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಉಬುಂಟುನ ಪ್ರತಿ ಬಿಡುಗಡೆಗೆ ಇದೂ ಕೂಡ ಆಧಾರವಾಗಿದೆ.ಡೆಬಿಯನ್ ಮೂರು ಆವೃತ್ತಿಗಳಲ್ಲಿ ಬರುತ್ತದೆ: ಸ್ಟೇಬಲ್, ಟೆಸ್ಟಿಂಗ್ ಮತ್ತು ಅನ್ ಸ್ಟೇಬಲ್. ಪ್ರತಿ ಡೆಬಿಯನ್ ಬಿಡುಗಡೆಗೆ , ಟಾಯ್ ಸ್ಟೋರಿ ಚಲನಚಿತ್ರದಲ್ಲಿನ ಒಂದು ಪಾತ್ರದ ಹೆಸರನ್ನು ಇಡಲಾಗಿದೆ.

ಇತರೆ (Others)

ಸೆಂಟ್ ಓಎಸ್,ಒರ್ಯಾಕಲ್ ಎಂಟರ್ ಪ್ರೈಸ್ ಲಿನಕ್ಸ್‌ಮತ್ತುಸೈಂಟಿಫಿಕ್ ಲಿನಕ್ಸ್‌ನಂತಹ ವಿತರಣೆಗಳು ರೆಡ್ ಹ್ಯಾಟ್ ಎಂಟರ್‌ಪ್ರೈಸ್ ಲಿನಕ್ಸ್ ಅನ್ನು ಆಧರಿಸಿವೆ. ಮತ್ತು ಒಂದೇ ರೀತಿಯ ತತ್ವಗಳು, ಡೈರೆಕ್ಟರಿಗಳು ಮತ್ತು ಸಿಸ್ಟಮ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ ತಂತ್ರಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತವೆ. ಲಿನಕ್ಸ್ ಮಿಂಟ್, ಎಡುಬುಂಟು ಮತ್ತು ಇನ್ನೂ ಅನೇಕ ಉಬುಂಟು ಹೆಸರಿನ ವಿತರಣೆಗಳು ಉಬುಂಟು ಅನ್ನು ಆಧರಿಸಿವೆ. ಮತ್ತು ಡೆಬಿಯನ್ ಜೊತೆಗೆ ಬಹಳಷ್ಟನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತವೆ. ಇತರ ನೂರಾರು ಲಿನಕ್ಸ್ ವಿತರಣೆಗಳು ಇವೆ.

© NIMI NOT TO BE REPUBLISHED

ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಪರಿಚಯ (Introduction to Open source software)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **OSS ನ ವ್ಯಾಖ್ಯಾನ**
- **OSS ನ ಇತಿಹಾಸ**
- **ಓಪನ್ ಸೋರ್ಸ್‌ನ ಪ್ರಾರಂಭ**
- **OSS ನ ಉದಾಹರಣೆಗಳು**

ವ್ಯಾಖ್ಯಾನ (Definition)

ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರ್ (OSS) ಎಂದರೆ ಸೋರ್ಸ್ ಕೋಡ್‌ನ ಜೊತೆಗೆ ವಿತರಿಸಲಾಗುವ, ಅದನ್ನು ಉಪಯೋಗಿಸಲು, ಮಾರ್ಪಾಡು ಮಾಡಲು ಲಭ್ಯವಿರುವ, ಮತ್ತು ಅದರ ಮೂಲ ಹಕ್ಕುಗಳೊಂದಿಗೆ ವಿತರಿಸಲಾಗುವ ಸಾಫ್ಟ್‌ವೇರ್ ಆಗಿದೆ. ಸೋರ್ಸ್ ಕೋಡ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಭಾಗವಾಗಿದ್ದು ಹೆಚ್ಚಿನ ಕಂಪ್ಯೂಟರ್ ಬಳಕೆದಾರರು ಅದನ್ನು ನೋಡಲಾಗುವುದಿಲ್ಲ. ಇದು ಕಂಪ್ಯೂಟರ್ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಒಂದು ಪ್ರೊಗ್ರಾಂ ಅಥವಾ ಅಪ್ಲಿಕೇಷನ್ ಹೇಗೆ ವರ್ತಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ನಿಯಂತ್ರಿಸಲು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಕೋಡ್ ಆಗಿದೆ. ಸೋರ್ಸ್ ಕೋಡ್‌ಗೆ ಪ್ರವೇಶ ಹೊಂದಿರುವ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಸರಿಯಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸದಿರುವ ಪ್ರೊಗ್ರಾಂಗಳಿಗೆ ಸೇರಿಸುವ ಮೂಲಕ, ಬದಲಾಯಿಸುವ ಮೂಲಕ ಅಥವಾ ಅದರ ಭಾಗಗಳನ್ನು ಸರಿ ಮಾಡುವ ಮೂಲಕ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಬದಲಾಯಿಸಬಹುದು. OSS ಸಾಮಾನ್ಯವಾಗಿ, ಪ್ರೊಗ್ರಾಮರ್‌ಗಳಿಗೆ ಅವರ ಅಗತ್ಯಗಳಿಗೆ ಸರಿಯಾಗಿ ಹೊಂದಿಕೊಳ್ಳುವಂತೆ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಲು ಅನುಮತಿಸುವ ಲೈಸೆನ್ಸ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರನ್ನು ಹೇಗೆ ವಿತರಿಸಬಹುದು ಎಂಬುದನ್ನು ನಿಯಂತಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

OSS ನ ಇತಿಹಾಸವೇನು ? (What is the history of OSS?)

ಸೋರ್ಸ್ ಕೋಡನ್ನು ಉಚಿತವಾಗಿ ಲಭ್ಯವಾಗುವಂತೆ ಮಾಡುವ ಕಲ್ಪನೆಯು 1983 ರಲ್ಲಿ ಹುಟ್ಟಿಕೊಂಡಿತು. ಇದನ್ನು MIT ನಲ್ಲಿ ಪ್ರೊಗ್ರಾಮರ್ ಆದ ರಿಚರ್ಡ್ ಸ್ಟಾಲ್‌ಮನ್ ಅವರ ಸೈದ್ಧಾಂತಿಕ ಚಳಿವಳಿಯಿಂದ ಸ್ಥಾಪಿಸಲಾಯಿತು. ಸ್ಟಾಲ್‌ಮನ್ ಅವರು ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು ಪ್ರೊಗ್ರಾಮರ್‌ಗಳಿಗೆ ಪ್ರೊಗ್ರಾಂನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವ, ಅದರ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳುವ, ಮತ್ತು ಅದನ್ನು ಸುಧಾರಿಸುವ ಗುರಿಯೊಂದಿಗೆ ಅವರು ಬಯಸಿದಂತೆ ಮಾರ್ಪಾಡು ಮಾಡಲು ಲಭ್ಯವಿರಬೇಕು ಎಂದು ನಂಬಿದ್ದರು. ಸ್ಟಾಲ್‌ಮನ್ ತನ್ನ ಸ್ವಂತ ಲೈಸೆನ್ಸ್ ಆದ GNU ಪಬ್ಲಿಕ್ ಲೈಸೆನ್ ಅಡಿಯಲ್ಲಿ ಉಚಿತ ಕೋಡ್‌ನ್ನು ಬಿಡುಗಡೆ ಮಾಡಲು ಪ್ರಾರಂಭಿಸಿದರು. ಸಾಫ್ಟ್‌ವೇರ್ ರಚನೆಯ ಈ ಹೊಸ ವಿಧಾನ ಮತ್ತು ಸಿದ್ಧಾಂತವು ಹಿಡಿತವನ್ನು ಪಡೆದುಕೊಂಡಿತು ಹಾಗೂ ಅಂತಿಮವಾಗಿ 1998 ರಲ್ಲಿ ಓಪನ್ ಸೋರ್ಸ್ ರಚನೆಗೆ ಕಾರಣವಾಯಿತು.

ಓಪನ್ ಸೋರ್ಸ್ ಇನಿಷಿಯೇಟಿವ್ ಎಂದರೇನು ?(What is the Open Source Initiative?)

ಓಪನ್ ಸೋರ್ಸ್ ಇನಿಶಿಯೇಟಿವ್ (OSI) ಅನ್ನು, ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಮತ್ತು ಕಮ್ಯುನಿಟಿಗಳನ್ನು ರಕ್ಷಿಸಲು ಮತ್ತು ಪ್ರಚಾರ ಮಾಡಲು ರಚಿಸಲಾಯಿತು. ಸಂಕ್ಷಿಪ್ತವಾಗಿ ಹೇಳುವುದಾದರೆ, OSI, ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಕೇಂದ್ರ ಮಾಹಿತಿ ಮತ್ತು ಆಡಳಿತ ಸಂಗ್ರಹವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು OSS ನ್ನು ಹೇಗೆ ಉಪಯೋಗಿಸಬೇಕು ಮತ್ತು ಸಂವಹನ ನಡೆಸಬೇಕು ಎಂಬುದಕ್ಕೆ ನಿಯಮಗಳನ್ನು ಮತ್ತು ಮಾರ್ಗಸೂಚಿಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಜೊತೆಗೆ ಕೋಡ್ ಲೈಸೆನ್ಸಿಂಗ್ ಮಾಹಿತಿ, ಬೆಂಬಲ, ವ್ಯಾಖ್ಯಾನಗಳು ಮುಂತಾದವುಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು ಸಾಮಾನ್ಯ ಕಮ್ಯುನಿಟಿ ಸಹಯೋಗ ಮುಂತಾದವುಗಳಿಂದ ಓಪನ್ ಸೋರ್ಸ್‌ನ್ನು ಬಳಸಲು ಮತ್ತು ಚಿಕಿತ್ಸೆ ನೀಡಲು ಅರ್ಹವಾಗುವಲ್ಲಿ ಸಹಾಯ ಮಾಡುತ್ತದೆ.

OSS ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ ? (How does OSS work?)

ಓಪನ್ ಸೋರ್ಸ್ ಕೋಡ್‌ನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಸಾರ್ವಜನಿಕ ಭಂಡಾರಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಸಾರ್ವಜನಿಕವಾಗಿ ಹಂಚಿಕೊಳ್ಳಲಾಗುತ್ತದೆ. ಯಾರಾದರೂ ಭಾಂಡಾರವನ್ನು ಪ್ರವೇಶಿಸಬಹುದು ಮತ್ತು ಕೋಡನ್ನು ಸ್ವತಂತ್ರವಾಗಿ ಬಳಸಬಹುದು. ಅಥವಾ ಒಟ್ಟಾರೆ ಯೋಜನೆಯ ವಿನ್ಯಾಸ ಮತ್ತು ಕ್ರಿಯಾತ್ಮಕತೆಯ ಸುಧಾರಣೆಗಳಿಗೆ ಕೊಡುಗೆಯನ್ನು ನೀಡಬಹುದು.

OSS ಸಾಮಾನ್ಯವಾಗಿ ವಿತರಣಾ ಲೈಸೆನ್ಸ್‌ನೊಂದಿಗೆ ಬರುತ್ತದೆ. ಈ ಲೈಸೆನ್ಸ್ ಡೆವಲಪರ್‌ಗಳು ಸಾಫ್ಟ್‌ವೇರನ್ನು ಹೇಗೆ ಬಳಸಬಹುದು, ಅಧ್ಯಯನ ಮಾಡಬಹುದು, ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು, ಮತ್ತು ಅತ್ಯಂತ ಮುಖ್ಯವಾಗಿವಿತರಣೆ ಮಾಡಬಹುದು ಎಂಬುದನ್ನು ತಿಳಿಸುವ ನಿಯಮಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. Synopsys Black Duck® KnowledgeBase ನ ಪ್ರಕಾರ ಅತ್ಯಂತ ಜನಪ್ರಿಯ ಐದು ಲೈಸೆನ್ಸ್‌ಗಳು ಯಾವವೆಂದರೆ :

- MIT License
- GNU General Public License (GPL) 2.0 - ಇದು ಅತ್ಯಂತ ಹೆಚ್ಚು ನಿರ್ಬಂಧಿತವಾಗಿದೆ ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಿದ ಕೋಡ್‌ನ ಪ್ರತಿಗಳನ್ನು ಸಾರ್ವಜನಿಕ ಬಳಕೆಗೆ ಲಭ್ಯವಾಗುವಂತೆ ಮಾಡುವ ಅಗತ್ಯವಿದೆ.

© NIMI NOT TO BE REPUBLISHED

- Apache License 2.0
- GNU General Public License (GPL) 3.0
- BSD License 2.0 (3-clause, New or Revised)- ಇದು ಕಡಿಮೆ ನಿರ್ಬಂಧಕವಾಗಿದೆ.

ಸೋರ್ಸ್ ಕೋಡ್‌ನ್ನು ಬದಲಾಯಿಸಿದಾಗ, ಏನನ್ನು ಬದಲಾಯಿಸಲಾಗಿದೆ ಮತ್ತು ಅದು ಒಳಗೊಂಡಿರುವ ವಿಧಾನಗಳನ್ನು OSS ಹೊಂದಿರಬೇಕು. ಲೈಸೆನ್ಸ್ ನಿಯಮಗಳ ಆಧಾರದ ಮೇಲೆ ಈ ಬದಲಾವಣೆಗಳಿಂದ ಉಂಟಾಗುವ ಸಾಫ್ಟ್‌ವೇರ್ ಉಚಿತವಾಗಿ ಲಭ್ಯವಿರುವಂತೆ ಮಾಡುವ ಅಗತ್ಯವಿರಬಹುದು ಅಥವಾ ಇಲ್ಲದಿರಬಹುದು.

OSS ನ ಕೆಲವು ಉದಾಹರಣೆಗಳು ಯಾವವು ? (What are some examples of OSS?)

- GNU/Linux
- Mozilla Firefox
- VLC media player
- SugarCRM
- GIMP
- VNC
- Apache web server
- LibreOffice
- jQuery

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಉಬುಂಟು ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು ಮತ್ತು ಲಿನಕ್ಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ಕಾರ್ಯಗತ ಗೊಳಿಸುವುದು (Install Ubuntu Linux operating system and execute basic Linux commands)

ಕಮಾಂಡ್‌ಗಳನ್ನು ಮತ್ತು ವಿವಿಧ ಎಡಿಟರ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Handling commands and various editors)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- ಟರ್ಮಿನಲ್ ಬಗ್ಗೆ ತಿಳಿಯುವುದು
- ಕಮಾಂಡ್ ಶೆಲ್‌ನ್ನು ವಿವರಿಸಿ
- ಲಿನಕ್ಸ್ ನ ಡೈರೆಕ್ಟರಿ ವಿನ್ಯಾಸವನ್ನು ಪಟ್ಟಿ ಮಾಡಿ
- ಲಿನಕ್ಸ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ

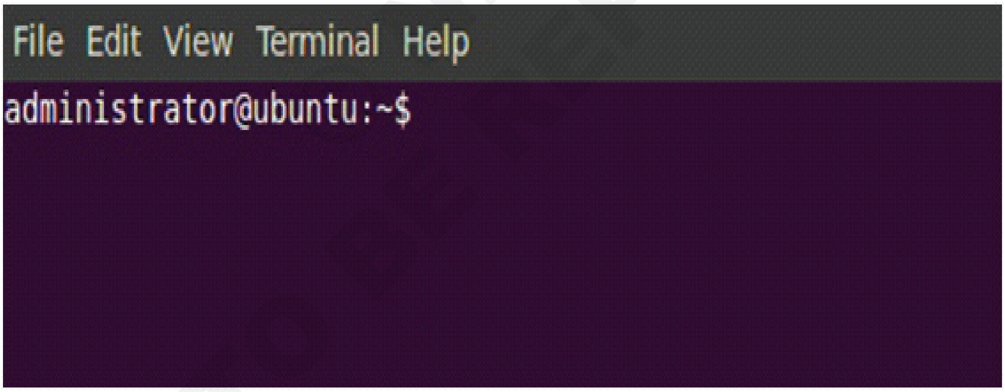
ಟರ್ಮಿನಲ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುವುದು (Starting up a terminal)

ಶೆಲ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲು, ಶೆಲ್ ತರಹದ ಅಪ್ಲಿಕೇಶನ್‌ನ್ನು ಬಳಸುತ್ತೇವೆ. ಇದನ್ನು ಟರ್ಮಿನಲ್ ಎಮ್ಯೂಲೇಟರ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಡೀಫಾಲ್ಟ ಟರ್ಮಿನಲ್‌ಗಳಾದ GNOME ಅಥವಾ KDE, Yakuake, ಅಥವಾ Guake,rxvt ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಉತ್ತಮ ಟರ್ಮಿನಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ದೊಡ್ಡ ಸಂಖ್ಯೆ ಇದೆ.

ಕೆಲವು ಜನಪ್ರೀಯ ಶೆಲ್‌ಗಳು ಯಾವವೆಂದರೆ :

- **ಬ್ಯಾಷ್** (bash) - ಎಂದರೆ ಬೌರ್ನ್ ಅಗೇನ್ ಶೆಲ್, ಹೆಚ್ಚಿನ ಲಿನಕ್ಸ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಇದು ಡೀಫಾಲ್ಟ ಶೆಲ್ ಆಗಿದೆ.
- Sh - ಎಂದರೆ ಬೌರ್ನ್ ಶೆಲ್ , ಇದು ಹಳೆಯ ಶೆಲ್ ಆಗಿದ್ದು, ಈಗ ಇದನ್ನು ವ್ಯಾಪಕವಾಗಿ ಬಳಸಲಾಗುತ್ತಿಲ್ಲ.

Fig 1



ಈಗ ನಾವು ನಮ್ಮ ಸಿಸ್ಟಮ್‌ನೊಂದಿಗೆ ಬರುವ ಡಿಫಾಲ್ಟನ್ನು ನೋಡೊಣ. ನೀವು GNOME ಉಪಯೋಗಿಸುತ್ತಿದ್ದರೆ ಈ ರೀತಿ ಟರ್ಮಿನಲ್‌ನ್ನು ಪ್ರವೇಶಿಸಬಹುದು. Applications ->Accessories -> Terminal ಅಥವಾ Alt + F2 ಒತ್ತುವುದರಿಂದ ಮತ್ತು ಅಲ್ಲಿ ಕಾಣಿಸುವ ರನ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ gnome-terminal ಎಂದು ಟೈಪ್ ಮಾಡಿ ಮತ್ತು Enter ಕೀಲಿಯನ್ನು ಒತ್ತಿ. ನೀವು KDE ಉಪಯೋಗಿಸುತ್ತಿದ್ದರೆ, Alt + F2 ಒತ್ತಿ ನಂತರ ಕನ್ಸೋಲ್ ಎಂದು ಟೈಪ್ ಮಾಡಿ.

ವಿತರಣೆಯನ್ನು ಅವಲಂಬಿಸಿ, ಈ ತರಹ ಪ್ರಾಂಪ್ಟ ಕಾಣಿಸುತ್ತದೆ. user@hosts$. ಆಂಪರ್ಸ್ಯಾಂಡ್ ಚಿನ್ಹೆಯ ಮೊದಲನೆಯ ಭಾಗದಲ್ಲಿ ಲಾಗಿನ್ ಬಳಕೆದಾರರ ಹೆಸರು ಮತ್ತು ಇನ್ನೊಂದು ಭಾಗದಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ನ ಹೋಸ್ಟ ಹೆಸರು ಇರುತ್ತದೆ.

ಕಮಾಂಡ್ ಶೆಲ್ (Command shell)

ಶೆಲ್ ಎನ್ನುವುದು ಕಮಾಂಡ್ ಇಂಟರ್‌ಪ್ರಿಟರ್ ಆಗಿದ್ದು , ಇದು ನಿಮ್ಮನ್ನು ಕಂಪ್ಯೂಟರ್‌ನೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಇದು ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು ಸುಲಭವಾಗಿದೆ. ನೀವು ಆಜ್ಞೆಗಳನ್ನು ಟೈಪ್ ಮಾಡುತ್ತೀರಿ. ಶೆಲ್ ಅವುಗಳನ್ನು ಇಂಟರ್ಪ್ರಿಟ್ ಮಾಡುತ್ತದೆ, ಅದು ಕೇಳಲಾದ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಮತ್ತು ಅಂತಿಮವಾಗಿ ಫಲಿತಾಂಶವನ್ನು ಒಂದು ಸ್ಟ್ಯಾಂಡರ್ಡ ಔಟಪುಟ್‌ಗೆ ಕಳುಹಿಸುತ್ತದೆ. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಇದು ಸ್ಕ್ರೀನ್ ಆಗಿರುತ್ತದೆ.

ಇದು ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳ ಪಟ್ಟಿಯಾಗಿದೆ. ಫೈಲ್ ಸಿಸ್ಟಮ್ ನ ಟ್ರೀ ಕ್ರಮಾನುಗತದಲ್ಲಿ ರೂಟ್ ಡೈರೆಕ್ಟರಿ

ಮೊದಲ ಸ್ಥಳವಾಗಿದೆ. ಮತ್ತು ಇದನ್ನು ಸ್ಲ್ಯಾಷ್ ಅಕ್ಷರ / ನಿಂದ ನಿರೂಪಿಸಲಾಗುತ್ತದೆ.

- Csh - ಎಂದರೆ ಸಿ ಶೆಲ್, ಇದು ಸಿಂಟ್ಯಾಕ್ಸ್ ನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ ಮತ್ತು ಸಿ ಪ್ರೋಗ್ರಾಮಿಂಗ ಭಾಷೆಯನ್ನು ಹೋಲುತ್ತದೆ.
- tchs - ಇದು ಸಿ ಶೆಲ್‌ನ ಸುಧಾರಿತ ಆವೃತ್ತಿಯಾಗಿದೆ.
- ksh - ಎಂದರೆ ಕಾರ್ನ್ ಶೆಲ್ , ಇದು 1980 ರ ದಶಕ ಆರಂಭದಲ್ಲಿ ಅಭಿವೃದ್ಧಿಗೊಂಡಿತು.
- dash - ಡೆಬಿಯನ್ ಆಲ್ಮಕ್ವಿಸ್ಟ ಶೆಲ್, ಈ ಶೆಲ್‌ನ್ನು ಡೆಬಿಯನ್ ವಿತರಕರು ತಯಾರಿಸಿದರು.

ವ್ಯವಸ್ಥೆಯಲ್ಲಿ ಲಭ್ಯವಿರುವ ಶೆಲ್‌ಗಳ ಪಟ್ಟಿ (Listing of shells available in the system $ cat /etc/shells/)

ಈ ಟ್ಯುಟೋರಿಯಲ್‌ನಲ್ಲಿ ನಾವು ಬ್ಯಾಷ್ (Bash) ಬಗ್ಗೆ ಹೆಚ್ಚು ಗಮನ ಹರಿಸುತ್ತೇವೆ. ಏಕೆಂದರೆ ಅದು ಹೆಚ್ಚು ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ ಮತ್ತು ಅತ್ಯಂತ ಶಕ್ತಿಶಾಲಿ ಶೆಲ್‌ಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಬ್ಯಾಷ್ ಇದು ಹಳೆಯ ಬೌರ್ನ್ ಶೆಲ್‌ನ ಅಧುನಿಕ ಅನುಷ್ಠಾನವಾಗಿದೆ. ಇದನ್ನು GNU ಯೋಜನೆಯಿಂದ ಅಬಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ. ಇದು ಅನೇಕ ಸಾಧನಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು ಇದರೊಂದಿಗೆ ಲಿನಕ್ಸ್ ಕರ್ನಲ್ , ಡೆಸ್ಕಟಾಪ್ ಪರಿಸರಗಳಾದ GNOME ಅಥವಾ KDE ಮತ್ತು ಇವುಗಳ ಮೇಲಿರುವ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಇವೆಲ್ಲ ಸೇರಿ ಇಡೀ ಲಿನಕ್ಸ್ ಪ್ಲಾಟಫಾರ್ಮನ್ನು ತಯಾರಿಸುತ್ತವೆ. ಡೆಬಿಯನ್ ಮತ್ತು ಉಬುಂಟು ವಿತರಣೆಯೊಂದಿಗೆ, ಸಿಸ್ಟಮ್‌ನೊಂದಿಗೆ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಶೆಲ್‌ನ್ನು ಫೈಲ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟಪಡಿಸಲಾಗಿದೆ. /etc/passwd (ಬ್ಯಾಷ್ ಡೀಫಾಲ್ಟ ಆಗಿರುತ್ತದೆ)

Fig 2

```
File  Edit  View  Terminal  Help
administrator@ubuntu:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
administrator@ubuntu:~$
```

ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಡೀಫಾಲ್ಟ ಶೆಲ್ ಅನ್ನು ಹೇಗೆ ಪ್ರದರ್ಶಿಸುವುದು(How to display default shell in the system)

ಟರ್ಮಿನಲ್‌ನಲ್ಲಿ ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿ

$ echo $SHELL

ಮತ್ತು Enter ಕೀಲಿಯನ್ನು ಒತ್ತಿ. ಡೀಫಾಲ್ಟ ಶೆಲ್‌ನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಚಿತ್ರ 3 ರಂತೆ

Fig 3

```
File  Edit  View  Terminal  Help
administrator@ubuntu:~$ echo $SHELL
/bin/bash
administrator@ubuntu:~$
```

ಡೈರೆಕ್ಟರಿ	ವಿವರಣೆ
	ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನ ಹೆಸರಿಲ್ಲದ ಮೂಲ. ಇತರ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿಗಳು, ಫೈಲ್‌ಗಳು, ಡ್ರೈವ್‌ಗಳು ಮತ್ತು ಸಾಧನಗಳು ಇವುಗಳನ್ನು ಈ ರೂಟ್‌ಗೆ ಜೋಡಿಸಲಾಗಿದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಲ್ಯಾಷ್ ಅಥವಾ / ಡೈರೆಕ್ಟರಿ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. / ಇದು ಕೇವಲ ಡೈರೆಕ್ಟರಿ ವಿಭಜಕವಾಗಿದೆ, ಡೈರೆಕ್ಟರಿ ಅಲ್ಲ.
/bin	ಅವಶ್ಯಕವಾದ ಕಮಾಂಡ್ ಬೈನರಿಗಳನ್ನು (ಪ್ರೋಗ್ರಾಂಗಳು) ಇಲ್ಲಿ ಸಂಗ್ರಹಿಸಿಡಲಾಗಿದೆ(bash, ls,mount,tar etc)
/boot	ಬೂಟ್ ಲೋಡರ್‌ನ ಸ್ಟ್ಯಾಟಿಕ್ ಫೈಲ್‌ಗಳು
/dev	ಸಾಧನಗಳ ಫೈಲ್‌ಗಳು. ಲಿನಕ್ಸ್‌ನಲ್ಲಿ ಹಾರ್ಡ್‌ವೇರ್ ಸಾಧನಗಳನ್ನು ಇತರ ಫೈಲ್‌ಗಳಂತೆ ಪ್ರವೇಶಿಸಬಹುದು ಮತ್ತು ಅವುಗಳನ್ನು ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇರಿಸಲಾಗುತ್ತದೆ.
/etc	ಹೋಸ್ಟ - ನಿರ್ದಿಷ್ಟ ಸಿಸ್ಟಮ್ ಕಾನ್ಫಿಗರೇಷನ್ ಫೈಲ್‌ಗಳು
/home	ಬಳಕೆದಾರರ ವೈಯಕ್ತಿಕ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗಳ ಸ್ಥಳ (ಉದಾ /home/susan)
/lib	ಅವಶ್ಯಕವಾದ ಶೇರ್ ಮಾಡಿದ ಲೈಬ್ರರಿಗಳು ಮತ್ತು ಕರ್ನಲ್ ಮಾಡ್ಯೂಲ್‌ಗಳು
/proc	ಪ್ರಾಸೆಸ್ ಇನ್ಫಾರ್ಮೇಶನ್ ಸುಡೋ ಫೈಲ್ ಸಿಸ್ಟಮ್ . ಕರ್ನಲ್ ಡಾಟಾ ಸ್ಟ್ರಕ್ಚರ್‌ಗೆ ಇಂಟರ್‌ಫೇಸ್ ಆಗಿರುತ್ತದೆ
/root	ರೂಟ್ (ಸೂಪರ್ ಯೂಸರ್) ಹೋಮ್ ಡೈರೆಕ್ಟರಿ
/sbin	ಅವಶ್ಯಕವಾದ ಸಿಸ್ಟಮ್ ಬೈನರಿ (fdisk, fsck,init, etc)
/tmp	ತಾತ್ಕಾಲಿಕ ಫೈಲ್ ಗಳು, ಎಲ್ಲ ಬಳಕೆದಾರರು ತಾತ್ಕಾಲಿಕ ಫೈಲ್‌ಗಳನ್ನು ಇಲ್ಲಿ ಇರಿಸಲು ಅನುಮತಿ ಇದೆ.
/usr	ಹೆಚ್ಚು ಹಂಚಿಕೊಳ್ಳಬಹುದಾದ , ಓದಲು ಮಾತ್ರ ಇರುವ ಡಾಟಾ (ಪ್ರೋಗ್ರಾಂಗಳು, ಲೈಬ್ರರಿಗಳು, ಡಾಕ್ಯುಮೆಂಟೇಶನ್ ಮತ್ತು ಇನ್ನೂ ಅನೇಕ) ಗಳಿಗೆ ಇರುವ ಮೂಲ ಡೈರೆಕ್ಟರಿ.
/usr/bin	ಹೆಚ್ಚಿನ ಯೂಸರ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಇಲ್ಲಿ ಇರಿಸಲಾಗುತ್ತದೆ.(cc, find, du, etc)
/usr/include	ಸಿ ಪ್ರೋಗ್ರಾಂ ಗಳನ್ನು ಕಂಪೈಲ್ ಮಾಡಲು ಹೆಡರ್ ಪ್ರೋಗ್ರಾಂ ಗಳು
/usr/lib	ಹೆಚ್ಚಿನ ಬೈನರಿ ಪ್ರೋಗ್ರಾಂ ಗಳಿಗಾಗಿ ಲೈಬ್ರರಿಗಳು
/usr/local	ಸ್ಥಳೀಯವಾಗಿ ಸ್ಥಾಪಿಸಲಾದ ಫೈಲ್‌ಗಳು. ಫೈಲ್‌ಗಳು ನೆಟ್‌ವರ್ಕನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುವ ಪರಿಸರದಲ್ಲಿ ಈ ಡೈರೆಕ್ಟರಿ ನಿಜವಾಗಿಯೂ ಮುಖ್ಯವಾಗಿದೆ. ಸ್ಥಳೀಯವಾಗಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಫೈಲ್‌ಗಳು /usr/local/bin, /usr/local/lib, etc) ಗಳಲ್ಲಿ ಸಂಗ್ರಹವಾಗುತ್ತವೆ. ಸೋರ್ಸ್ ನಿಂದ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಸಾಫ್ಟವೇರ್ ಪ್ಯಾಕೇಜಗಳಿಗೆ ಇದನ್ನು ಹೆಚ್ಚಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ. ಅಥವಾ ವಿತರಣೆಯೊಂದಿಗೆ ಅಧಿಕೃತವಾಗಿ ರವಾನಿಸದ ಸಾಫ್ಟವೇರ್‌ಗಳಿಗೆ ಬಳಸಲಾಗುತ್ತದೆ.
/usr/sbin	ಪ್ರಮುಖವಲ್ಲದ ಸಿಸ್ಟಮ್ ಬೈನರಿಗಳು(lpd, useradd, etc)
/usr/share	ಆರ್ಕಿಟೆಕ್ಚರ್ ಇಂಡಿಪೆಂಡೆಂಟ್ ಡಾಟಾ (ಐಕಾನ್‌ಗಳು, ಹಿನ್ನೆಲೆಗಳು, ಡಾಕ್ಯುಮೆಂಟೇಷನ್ , ಟರ್ಮ್ ಇನ್ಫೊ, ಮ್ಯಾನ್ ಪೇಜಸ್ ಇತ್ಯಾದಿಗಳು)
/usr/src	ಪ್ರೊಗ್ರಾಂ ಸೋರ್ಸ್ ಕೋಡ್ ಉದಾ : ಲಿನಕ್ಸ್ ಕರ್ನಲ್ , ಸೋರ್ಸ್ ಆರ್ ಪಿ ಎಮ್ ಇತ್ಯಾದಿಗಳು
/usr/X11R6	ಎಕ್ಸ್ ವಿಂಡೋ ಸಿಸ್ಟಮ್
/var	ವೇರಿಯೇಬಲ್ ಡಾಟಾ : ಮೇಲ್ ಮತ್ತು ಪ್ರಿಂಟರ್ ಸ್ಪೂಲ್, ಲಾಗ್ ಫೈಲ್‌ಗಳು, ಲಾಕ್ ಫೈಲ್ ಗಳು

ಲಿನಕ್ಸ್ ಆಜ್ಞೆಗಳು ಯಾವವು?(What are Linux Commands)

ಲಿನಕ್ಸ್ ಆಜ್ಞೆಗಳು ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದಾದ ಬೈನರಿ ಫೈಲ್‌ಗಳಾಗಿವೆ. ಕೆಲವು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಇವುಗಳನ್ನು ರನ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಲು ಇಡೀ ಗ್ರಾಫಿಕಲ್ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ರನ್ ಮಾಡುವುದು. ಆಗಾಗ್ಗೆ ಬಳಸುವ ಆಜ್ಞೆಗಳಿಗೆ ಉದಾಹರಣೆಗಳು,

© NIMI NOT TO BE REPUBLISHED

ls, cd, pwd, date ಅಥವಾ cat. ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದಾದ ಫೈಲ್‌ಗಳನ್ನು ಹೊರತುಪಡಿಸಿ ಶೆಲ್ ಅಂತರ್ನಿರ್ಮಿತ ಶೆಲ್ಸ ಎಂಬ ವರ್ಗವೂ ಇದೆ. ಈ ಆಜ್ಞೆಗಳನ್ನು ಸ್ವತಃ ಶೆಲ್ ನೀಡುತ್ತದೆ(ನಮ್ಮಲ್ಲಿ ಬ್ಯಾಷ್) .ಅವುಗಳನ್ನು ನಂತರ ನೋಡೋಣ.

ಲಿನಕ್ಸ್ ಆಜ್ಞೆಯ ಸಾಮಾನ್ಯ ರೂಪ(The general form of a linux command is :)

ಆಜ್ಞೆ ಆಯ್ಕೆ(ಗಳು) , ಫೈಲ್(ಗಳು)Command options(s) filename(s)

ಇದು ಆಜ್ಞೆಯನ್ನು ಸೂಚಿಸುತ್ತದೆ ನಂತರ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಮಾನದಂಡಗಳು ಮತ್ತು ಅದನ್ನು ಅನ್ವಯಿಸಲು ಐಚ್ಛಿಕವಾಗಿ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳು. ಉದಾಹರಣೆಗೆ :

$ echo -e Hello, world !\n

ಇದರ ಔಟಪುಟ್ Hello, world ! ಎಂದು ತೋರಿಸುತ್ತದೆ. ಇಲ್ಲಿ -e ಪ್ಯಾರಾಮೀಟರ್, \n ನಂತಹ ಎಸ್ಕೇಪ್ ಅಕ್ಷರವನ್ನು ಅರ್ಥೈಸಲು echo ಆಜ್ಞೆಗೆ ಹೇಳುತ್ತದೆ. \n ಒಂದು ಕೋಟ್‌ನಲ್ಲಿರುವ ಪಠ್ಯದ ನಂತರ ಹೊಸ ಸಾಲನ್ನು ಸೇರಿಸುತ್ತದೆ. ಡಾಲರ್ ಚಿನ್ಹೆಯನ್ನು ನಿರ್ಲಕ್ಷಿಸಿ. ಇದು ಶೆಲ್ ಪ್ರಾಂಪ್ಟನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಯಾವುದೇ ಆಜ್ಞೆಯು ಆರ್ಗ್ಯುಮೆಂಟ್ಗಳನ್ನು ಹೊಂದಿರಬಹುದು ಅಥವಾ ಹೊಂದಿಲ್ಲದಿರಬಹುದು. ಅದು ಒಂದು ಆಯ್ಕೆ ಅಥವಾ ಫೈಲ್ ಹೆಸರು ಆಗಿರಬಹುದು.

ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ವಿಶೇಷ ಅಕ್ಷರಗಳು(Special characters in Linux operating system)

ಶೆಲ್ ವಿಶೇಷ ರೀತಿಯಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸುವ ಅನೇಕ ಚಿಹ್ನೆಗಳು ಮತ್ತು ಅಕ್ಷರಗಳು ಇವೆ ಎಂದು ತಿಳಿದುಕೊಳ್ಳುವುದು ಬಹಳ ಮುಖ್ಯವಾಗಿದೆ. ಕೆಲವು ಅಕ್ಷರಗಳನ್ನು ಎ) ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ಬಳಸಲಾಗುವುದಿಲ್ಲ ಬಿ) ವಿಶೇಷ ಕಾರ್ಯನಿರ್ವಹಣೆಗಾಗಿ ಉಪಯೋಗಿಸಬಹುದು. ಸಿ) ಅವುಗಳನ್ನು ಸಾಮಾನ್ಯ ರೀತಿಯಲ್ಲಿ ಬಳಸಲು ಬಯಸಿದರೆ ಎಸ್ಕೇಪ್ ಆಗಿ ಉಪಯೋಗಿಸಬೇಕು.

ಅಕ್ಷರ	ವಿವರಣೆ
\	ಎಸ್ಕೇಪ್ ಅಕ್ಷರ. ನೀವು ವಿಶೇಷ ಅಕ್ಷರವನ್ನು ಉಲ್ಲೇಖಿಸಲು ಬಯಸಿದರೆ, ಮೊದಲು ಬ್ಯಾಕ್ ಸ್ಲ್ಯಾಷ್‌ನೊಂದಿಗೆ ಅದನ್ನು ಎಸ್ಕೇಪ್ ಮಾಡಬೇಕು. ಉದಾ : touch/tmp/filename*
/	ಡೈರೆಕ್ಟರಿ ವಿಭಜಕ , ಇದನ್ನು ಡೈರೆಕ್ಟರಿ ಹೆಸರುಗಳನ್ನು ಪ್ರತ್ಯೇಕಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಉದಾ : /usr/src/linux
.	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿ. ಫೈಲ್ ಹೆಸರಿನಲ್ಲಿ ಇದು ಮೊದಲ ಅಕ್ಷರವಾಗಿದ್ದರೆ, ಫೈಲ್‌ಗಳನ್ನು ಹೈಡ್ ಮಾಡಬಹುದು.
..	ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿ
~	ಬಳಕೆದಾರರ ಹೊಮ್ ಡೈರೆಕ್ಟರಿ
*	ಫೈಲ್ ಹೆಸರಿನಲ್ಲಿ 0 ಅಥವಾ ಹೆಚ್ಚು ಅಕ್ಷರಗಳನ್ನು ಅಥವಾ ತನ್ನನ್ನು, ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಉದಾ : pic*2002 ಇದು pic2002, picJanuary2002, picFeb292002 ಮುಂತಾದ ಫೈಲ್‌ಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ
?	ಫೈಲ್ ಹೆಸರಿನಲ್ಲಿನ ಒಂದು ಅಕ್ಷರವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಉದಾ : hello?.txt ಇದು hello1.txt, helloz.txt ಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ, ಆದರೆ hello22.txt ನ್ನು ಇಲ್ಲ.
[ ]	ಇದನ್ನು ಮೌಲ್ಯಗಳ ಶ್ರೇಣಿಯನ್ನು ಪ್ರತಿನಿಧಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಉದಾ :[0-9], [A-Z] ಇತ್ಯಾದಿ. ಉದಾ : hello [0-2].txt ಇದು hello0.txt, hello1.txt ಮತ್ತು hello2.txt ಹೆಸರುಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ
|	ಇದನ್ನು ಪೈಪ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಒಂದು ಆಜ್ಞೆಯ ಔಟ್ಪುಟ್‌ನ್ನು ಇನ್ನೊಂದು ಆಜ್ಞೆಗೆ ಮರುನಿರ್ದೇಶಿಸುತ್ತದೆ. ಉದಾ : ls | more
>	ಒಂದು ಆಜ್ಞೆಯ ಔಟಪುಟ್ ಅನ್ನು ಹೊಸ ಫೈಲ್‌ಗೆ ಮರುನಿರ್ದೇಶಿಸುತ್ತದೆ. ಫೈಲ್ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿದ್ದರೆ, ಅದನ್ನು ಮತ್ತೆ ಬರೆಯುತ್ತದೆ. ಉದಾ : ls > myfiles.txt
>>	ಒಂದು ಆಜ್ಞೆಯ ಔಟ್ಪುಟ್‌ನ್ನು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಫೈಲ್ ನ ಕೊನೆಯಲ್ಲಿ ಮರುನಿರ್ದೇಶಿಸುತ್ತದೆ. ಉದಾ : echo. Mary 555-1234. >>phonenumbers.txt

© NIMI NOT TO BE REPUBLISHED

<	ಒಂದು ಫೈಲ್‌ಅನ್ನು , ಪ್ರೋಗ್ರಾಂಗೆ ಇನ್‌ಪುಟ್ ಆಗಿ ಮರುನಿರ್ದೇಶಿಸುವುದು. ಉದಾ : more < phonenumbers.txt
;	ಇದು ಆಜ್ಞಾ ವಿಭಜಕವಾಗಿದೆ (command separator), ಒಂದೇ ಸಾಲಿನಲ್ಲಿ ಅನೇಕ ಆಜ್ಞೆಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಉದಾ : cd/ var/ log ; less messages

Cd ಆಜ್ಞೆ (The cd command)

ಲಿನಕ್ಸ್ ಮತ್ತು ಇತರ ಯುನಿಕ್ಸ್‌ನಂತಹ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಬದಲಾಯಿಸಲು ಸಿಡಿ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿ ಎಂದರೆ ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ). ಇದು ಎಮ್‌ಎಸ್ ಡಾಸ್ ನಲ್ಲಿರುವ CD ಮತ್ತು CHDIR ಆಜ್ಞೆಗಳನ್ನು ಹೋಲುತ್ತದೆ.

ಸಿಡಿಯ ಸಿಂಟ್ಯಾಕ್ಸ್ (cd`s syntax is)

Cd [option] [directory]

ಸ್ಕ್ವೇರ್ ಬ್ರಾಕೆಟ್‌ನಲ್ಲಿರುವ ವಸ್ತುಗಳು ಐಚ್ಛಿಕವಾಗಿರುತ್ತವೆ. ಈ ಆಜ್ಞೆಯನ್ನು ಯಾವುದೇ ಡೈರೆಕ್ಟರಿ ಹೆಸರನ್ನು ಸೂಚಿಸದೇ ಉಪಯೋಗಿಸಿದರೆ, ಇದು ಬಳಕೆದಾರರನ್ನು ಹಿಂದಿನ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಹಿಂದಿರುಗಿಸುತ್ತದೆ. ಇದು ಎರಡು ಡೈರೆಕ್ಟರಿಗಳ ನಡುವೆ ಟಾಗಲ್ ಮಾಡುವುದಕ್ಕೆ ಅನುಕೂಲಕರವಾಗಿದೆ.

ಡೈರೆಕ್ಟರಿ ಹೆಸರನ್ನು ಒದಗಿಸಿದಾಗ , ಸಿಡಿ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಅದಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ. ಸಂಪೂರ್ಣ ಮಾರ್ಗದ ಹೆಸರನ್ನು ಹೆಸರನ್ನಾಗಿ ಬರೆಯಬಹುದು. (ಅಂದರೆ ರೂಟ್ ಡೈರೆಕ್ಟರಿಗೆ ಸಂಬಂಧಿಸಿದ ಸ್ಥಳ) ಅಥವಾ ಸ್ಥಳೀಯ ಮಾರ್ಗದ ಹೆಸರು(ಅಂದರೆ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಸಂಬಂಧಿಸಿದ ಸ್ಥಳ) . ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯ ಸಬ್ ಡೈರೆಕ್ಟರಿಗೆ ಬದಲಾಯಿಸುವಾಗ ಸ್ಥಳೀಯ ಮಾರ್ಗದ ಹೆಸರನ್ನು ಉಪಯೋಗಿಸುವುದು ಹೆಚ್ಚು ಅನುಕೂಲವಾಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ , ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯು , ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು , ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಅದು ಎಲ್ಲಿದೆ ಎಂಬುದನ್ನು ಲೆಕ್ಕಿಸದೇ (ಏಕೆಂದರೆ ಇದು ಒಂದು ಸಂಪೂರ್ಣ ಮಾರ್ಗವಾಗಿದೆ) ರೂಟ್ ಡೈರೆಕ್ಟರಿಗೆ ಬದಲಾಯಿಸುತ್ತದೆ. (ಇದನ್ನು ಫಾರ್ವರ್ಡ ಸ್ಲ್ಯಾಷ್‌ನಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ.)

cd /

ಅಂತೆಯೆ ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯು , ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು , ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಅದು ಎಲ್ಲಿದೆ ಎಂಬುದನ್ನು ಲೆಕ್ಕಿಸದೇ , usr/sbin

ಡೈರೆಕ್ಟರಿಗೆ (ಇದು ಸಿಸ್ಟಮ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್ ಉಪಯೋಗಿಸುವ ಪ್ರಮುಖವಲ್ಲದ ಸಿಸ್ಟಮ್ ಉಪಯುಕ್ತತೆಗಳನ್ನು ಒಳಗೊಂಡಿದೆ) ಬದಲಾಯಿಸುತ್ತದೆ.

cd /usr/sbin

ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಾದ /usr/local/share/man/ ನಿಂದ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯ ಸಬ್ ಡೈರೆಕ್ಟರಿಯಾದ /usr/local/share/man/ man2 ಗೆ ಬದಲಾಯಿಸಲು ಇಚ್ಛಿಸಿದರೆ , ಇದನ್ನು ಸಂಪೂರ್ಣ ಪಾಥ್ ಹೆಸರನ್ನು ಬಳಸಿಕೊಂಡು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿದೆ. ಅಂದರೆ ,

cd /usr/local/share/man/man2

ಅದಾಗ್ಯೂ , ಸಂಬಂಧಪಟ್ಟ ಪಾಥ್ ಹೆಸರನ್ನು ಬಳಸುವುದು ಬೇಸರದ ಸಂಗತಿಯಾಗಿದೆ, ಎಂದರೆ

cd man2

ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಒಂದೇ ಡಾಟ್‌ನಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಅದರ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿಯನ್ನು (ಎಂದರೆ ಅದನ್ನು ಒಳಗೊಂಡಿರುವ ಡೈರೆಕ್ಟರಿ) ಎರಡು ಸತತ ಡಾಟ್‌ಗಳಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಿಂದ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿಗೆ ಬದಲಾಯಿಸಲು ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸುವ ಮೂಲಕ ಸಾಧ್ಯವಾಗುತ್ತದೆ:

cd ..

ಸಿಡಿ ಆಜ್ಞೆಯ ಮತ್ತೊಂದು ಅನುಕೂಲಕರ ವೈಶಿಷ್ಟ್ಯವೆಂದರೆ ಯಾವುದೇ ಬಳಕೆದಾರರಿಗೆ ಕೇವಲ ಟಿಲ್ಡ್ (~ ಚಿನ್ಹೆ) ಅನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಉಪಯೋಗಿಸಿ ನೇರವಾಗಿ ಅದರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗೆ ಮರಳುವ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿದೆ. ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯನ್ನು , ಲಾಗಿನ್ ಡೈರೆಕ್ಟರಿ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಈ ಡೈರೆಕ್ಟರಿಯು ಬಳಕೆದಾರರ ವೈಯಕ್ತಿಕ ಫೈಲ್‌ಗಳನ್ನು , ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮತ್ತು ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ. ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಲಾಗಿನ್ ಆದ ನಂತರ ಬಳಕೆದಾರರು ಮೊದಲು ಪ್ರವೇಶಿಸಿದ ಡೈರೆಕ್ಟರಿ ಕೂಡ ಇದೇ ಆಗಿರುತ್ತದೆ. ಟಿಲ್ಡ್ ಇದು ಸಣ್ಣ , ಅಲೆಯಾದ, ಸಮತಲವಾದ ರೇಖೆಯ ಅಕ್ಷರವಾಗಿದೆ. ಇದು ಪ್ರಸ್ತುತ ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಎಂದರೆ ಯಾವುದೇ ಬಳಕೆದಾರರು, ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯನ್ನು ಟೈಪ್ ಮಾಡಿ, ನಂತರ ENTER ಕೀಲಿಯನ್ನು ಒತ್ತುವ ಮೂಲಕ ತಕ್ಷಣವೇ ಅದರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗೆ ಹಿಂತಿರುಗಿ ಹೋಗಬಹುದು.

cd ~

© NIMI NOT TO BE REPUBLISHED

ಇದು ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯ ಪೂರ್ಣ ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡುವುದಕ್ಕಿಂತ ಸುಲಭವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ಬಳಕೆದಾರರ ಹೆಸರು Josephine ಇದ್ದರೆ /home/Josephine. ಎಂದಾಗುತ್ತದೆ. (ಮತ್ತು ಇದು ಕಮಾಂಡ್ ಲೈ ನ್‌ನ್ನು ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಬಳಸಲು ಸುಲಭವಾಗುವಂತೆ ಸಹಾಯ ಮಾಡಲು ಬಳಸುವ ಅಸಂಖ್ಯಾತ ಶಾರ್ಟಕಟ್ಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ)

ಸಿಡಿ, ನಂತರ ಒಂದು ಸ್ಪೇಸ್ ಮತ್ತು ಒಂದು ಹೈಫನ್ ಆಜ್ಞೆಯು ಬಳಕೆದಾರರನ್ನು ಹಿಂದಿನ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಹಿಂದಿರುಗಿಸುತ್ತದೆ. ಮತ್ತು ಆ ಡೈರೆಕ್ಟರಿಯ ಸಂಪೂರ್ಣ ಪಾಥ್ ಹೆಸರನ್ನು ಹೊಸ ಸಾಲಿ ನಲ್ಲಿ ವರದಿ ಮಾಡುತ್ತದೆ. ಇದು ಈಗಾಗಲೆ ಅನುಕೂಲವಾಗಿರುವ ,ಸಿಡಿ ಆಜ್ಞೆಯ ಟಾಗಲ್ ಸಾಮರ್ಥ್ಯವನ್ನು ಇನ್ನೂ ಹೆಚ್ಚಿಸುತ್ತದೆ. ಎರಡು ಡೈರೆಕ್ಟರಿಗಳಲ್ಲಿ ಒಂದು ದೀರ್ಘವಾದ ಸಂಪೂರ್ಣ ಪಾಥ್ ಹೆಸರನ್ನು ಹೊಂದಿದ್ದಾಗ ಉದಾಹರಣೆಗೆ /usr/local/share/man/ man2 ನಂತಹ ಪಾಥ್ ಹೆಸರು ಇದ್ದಾಗ ಟಾಗಲ್ ಮಾಡುವುದು ವಿಶೇಷವಾಗಿ ಅನುಕೂಲಕರವಾಗಿದೆ.

ಸಿಡಿ ಕೇವಲ ಎರಡು ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ. ಮತ್ತು ಅವೆರಡಲ್ಲಿ ಯಾವುದನ್ನೂ ಸಾಮಾನ್ಯವಾಗಿ ಬಳಸುವುದಿಲ್ಲ. -P ಆಯ್ಕೆಯು ಸಾಂಕೇತಿಕ ಲಿಂಕ್‌ನ್ನು ಉಪಯೊಗಿಸುವ ಬದಲು ಭೌತಿಕ ಡೈರೆಕ್ಟರಿ ರಚನೆಯನ್ನು ಉಪಯೋಗಿಸಲು ಸೂಚಿಸುತ್ತದೆ. -L ಆಯ್ಕೆಯು ಸಾಂಕೇತಿಕ ಲಿಂಕನ್ನು ಅನುಸರಿಸಲು ತಿಳಿಸುತ್ತದೆ.

Pwd ಆಜ್ಞೆ (The pwd command)

Pwd ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಪೂರ್ಣ ಪಾಥ್ ಅನ್ನು ವರದಿ ಮಾಡುತ್ತದೆ.

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿ ಎಂದರೆ ಬಳಕೆದಾರರು ಕಮಾಂಡ್ ಲೈನ್ ಇಂಟರ್‌ಫೇಸನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಪ್ರಸ್ತುತವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ ಆಗಿದೆ. ಕಮಾಂಡ್ ಲೈನ್ ಇ ಂಟರ್‌ಫೇಸ ಎಂದರೆ ಎಲ್ಲ ಪಠ್ಯಮಾದರಿಯ ಪ್ರದರ್ಶನವಿರುತ್ತದೆ ಮತ್ತು ಇದನ್ನು ಕನ್ಸೋಲ್ ಮೂಲಕ ಒದಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ ಈ ಪ್ರದರ್ಶನ ಮಾದರಿಯಲ್ಲಿ ಇಡೀ ಸ್ಕ್ರೀನ್ ಪಠ್ಯ ಮಾತ್ರ ಹೊಂದಿರುತ್ತದೆ) ಅಥವಾ ಟರ್ಮಿನಲ್ ವಿಂಡೊ ಮೂಲಕ ಒದಗಿಸಲಾಗುತ್ತದೆ. (ಎಂದರೆ GUI ನಲ್ಲಿ ಪಠ್ಯ ಮಾತ್ರ ವಿಂಡೋ)

ಒಂದು ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲ್‌ನ ಪೂರ್ಣ ಪಾಥ್ ಗೆ ಸಂಪೂರ್ಣ ಪಾಥ್(absolute path) ಎಂದರೆ ಆ ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲನ್ನು ಒಳಗೊಂಡಂತೆ ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಿಂದ ಆ ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲ್‌ವರೆಗಿನ ಸಂಪೂರ್ಣ ಕ್ರಮಾನುಗತವಾಗಿದೆ. ರೂಟ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ಫಾರ್ವರ್ಡ ಸ್ಲ್ಯಾಷ್(/) ನಿಂದ ಹೆಸರಿಸಲಾಗುತ್ತದೆ. ಇದು ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನ ಮೂಲ ಡೈರೆಕ್ಟರಿಯಾಗಿದೆ. ಹೀಗಾಗಿ ಯಾವುದೇ ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲ್ ನ ಪೂರ್ಣ ಪಾಥ್ ಯಾವಾಗಲೂ ಫಾರ್ವರ್ಡ ಸ್ಲ್ಯಾಷ್(/)ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

Pwd ಇದು ಲಿನಕ್ಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್‌ನಂತಹ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಮೂಲ ಆಜ್ಞೆಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ls ಆಜ್ಞೆಯನ್ನು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ವಿಷಯಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು cd ಆಜ್ಞೆಯನ್ನು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಬದಲಾಯಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

Pwd ಸಿಂಟ್ಯಾಕ್ಸ್

Pwd [option]

Pwd ಆಜ್ಞೆಯನ್ನು ಯಾವಾಗಲೂ ಹೆಚ್ಚಾಗಿ ಅದನ್ನಷ್ಟೇ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ

Pwd

ಎಂದರೆ, ಇದನ್ನು ಇದರ ಆಯ್ಕೆಗಳೊಂದಿಗೆ ವಿರಳವಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳೊಂದಿಗೆ ಎಂದೂ ಬಳಸಲಾಗುವುದಿಲ್ಲ. (ಎಂದರೆ ಫೈಲ್ ಹೆಸರು ಅಥವಾ ಇತರ ಯಾವುದೇ ಮಾಹಿತಿ ಇನ್‌ಪುಟ್ ಆಗಿ ಒದಗಿಸುವುದು). Pwd ನಂತರ ಅದೇ ಸಾಲಿನಲ್ಲಿ ಆಯ್ಕೆಯನ್ನು ಹೊರತುಪಡಿಸಿ ಏನನ್ನಾದರು ಟೈಪ್ ಮಾಡಿದರೆ ಅದನ್ನು ನಿರ್ಲಕ್ಷಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಯಾವುದೇ ದೋಷ ಸಂದೇಶಗಳನ್ನು ಹಿಂತಿರುಗಿಸಲಾಗುವುದಿಲ್ಲ.

ಉದಾಹರಣೆಗೆ ಯೂಸರ್ ನೇಮ್ Janis ಎಂದು ಹೊಂದಿರುವ ಬಳಕೆದಾರರು ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿದ್ದರೆ ಮೇಲಿನ ಆಜ್ಞೆಯು / home/Janis / ಎಂದು ಹಿಂದಿರುಗಿಸುತ್ತದೆ.(ಏಕೆಂದರೆ ಡೀಫಾಲ್ಟ್ ಆಗಿ ಎಲ್ಲ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗಳು /home ಎನ್ನುವ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇರುತ್ತವೆ.)

ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತವಾಗಿ /usr/share/config (ಇದು ಹಲವಾರು ಪ್ರೊಗ್ರಾಂ ಕಾನ್ಫಿಗರೇಷನ್ ಫೈಲ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ) ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿದ್ದರೆ., ಇದೇ ಆಜ್ಞೆಯು /usr/ share/config ಎಂದು ಹಿಂದಿರುಗಿಸುತ್ತದೆ.

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಬಳಕೆದಾರರು ಸಿಡಿ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಬದಲಾಯಿಸಲು ಉದ್ದೇಶಿಸಿದ ಡೈರೆಕ್ಟರಿಗೆ ಬದಲಾಗಿದ್ದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು pwd ಆಜ್ಞೆಯು ಉಪಯುಕ್ತವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ಸಿಡಿ ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಾದ /home/Janis ಯಿಂದ / usr/share/config ಗೆ ಬದಲಾಯಿಸಿದ್ದನ್ನು , pwd ಆಜ್ಞೆಯನ್ನು ಉಪಯೋಗಿಸಿ,

ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಬಹುದು. ಎಂದರೆ ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಗಳ ಅನುಕ್ರಮವನ್ನು ನೀಡಬಹುದು. :

```
cd /usr/share/config/

Pwd
```

© NIMI NOT TO BE REPUBLISHED

Pwd ಆಜ್ಞೆಯ ಪ್ರಮಾಣಿತ ಆವೃತ್ತಿಯು ಕೇವಲ ಎರಡು ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ. ಅವುಗಳಲ್ಲಿ ಎರಡನ್ನೂ ಅತ್ಯಂತ ವಿರಳವಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದರಲ್ಲಿ --help ಆಯ್ಕೆಯನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ :

pwd --help

ಈ ಆಯ್ಕೆಯು Pwd ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಇದು ಸರಳವಾದ ಆಜ್ಞೆಯಾಗಿದೆ(ಎಂದರೆ ಇದು ಕೇವಲ ಎರಡು ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ಯಾವುದೇ ಆರ್ಗ್ಯುಮೆಂಟಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದಿಲ್ಲ)

ಇನ್ನೊಂದು ಆಯ್ಕೆ --version ಆಗಿದ್ದು, ಇದು ಆವೃತ್ತಿ ಸಂಖ್ಯೆಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ., ಎಂದರೆ

pwd --version

ಇದು ಪ್ರಸ್ತುತ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿಗಾಗಿ ಇರುತ್ತದೆ ಎಂದು ಭಾವಿಸಲಾಗಿದೆ. Pwd , ವಾಸ್ತವವಾಗಿ ಪ್ರಿಂಟ್ ವರ್ಕಿಂಗ್ ಡೈರೆಕ್ಟರಿ(Print working Directory) ಇದರ ಸಂಕ್ಷಿಪ್ತ ರೂಪವಾಗಿದೆ. ಸಾಂಪ್ರದಾಯಿಕ ಯುನಿಕ್ಸ್ ಪರಿಭಾಷೆಯಲ್ಲಿ ಪ್ರಿಂಟ್ ಶಬ್ದದ ಅರ್ಥ ,ಬರೆಯುವುದು ಅಥವಾ ಪ್ರದರ್ಶಿಸುವುದು ಎಂದು ಆಗುತ್ತದೆ. ಯಾವಾಗ ಕಂಪ್ಯೂಟರ್ ಔಟಪುಟ್‌ನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಕಾಗದದ ಮೇಲೆ ಮುದ್ರಿಸಲಾಯಿತು ಆವಾಗ ಈ ಶಬ್ದವು ಹುಟ್ಟಿಕೊಂಡಿತು. ಏಕೆಂದರೆ ಸಿಆರ್ ಟಿ(ಕ್ಯಾಥೋಡ್ ರೇ ಟ್ಯೂಬ್) ಡಿಸ್ಪ್ಲೇ ಮಾನಿಟರ್‌ಗಳು ಇನ್ನೂ ವ್ಯಾಪಕವಾಗಿ ಲಭ್ಯ ಇರಲಿಲ್ಲ.

echo ಆಜ್ಞೆ (echo command)

ಬ್ಯಾಷ್ ಮತ್ತು ಸಿ ಶೆಲ್ ಗಳಲ್ಲಿ echo ಒಂದು ಅಂತರ್ನಿರ್ಮಿತ ಆಜ್ಞೆಯಾಗಿದೆ. ಇದು ಸ್ಟ್ಯಾಂಡರ್ಡ ಔಟಪುಟ್‌ಗೆ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಬರೆಯುತ್ತದೆ.

ಶೆಲ್ ಎನ್ನುವುದು ಒಂದು ಪ್ರೋಗ್ರಾಂ ಆಗಿದ್ದು , ಲಿನಕ್ಸ್ ಮತ್ತು ಇತರ ಯುನಿಕ್ಸ್‌ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಕಮಾಂಡ್ ಲೈನನ್ನು (ಎಂದರೆ , ಎಲ್ಲ ಟೆಕ್ಸ್ಟ್ ಡಿಸ್ಪ್ಲೇ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್) ಒದಗಿಸುತ್ತದೆ. ಇದು ಇದರಲ್ಲಿ ಟೈಪ್ ಮಾಡಿದ ಎಲ್ಲ ಆಜ್ಞೆಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಲಿನಕ್ಸ್‌ನಲ್ಲಿ ಬ್ಯಾಷ್ ಇದು ಡೀಫಾಲ್ಟ್ ಶೆಲ್ ಆಗಿದೆ.

ಕಮಾಂಡ್ ಇದು ಕಂಪ್ಯೂಟರ್‌ಗೆ ಏನನ್ನಾದರೂ ಮಾಡಲು ಹೇಳುವ ಒಂದು ಸೂಚನೆಯಾಗಿದೆ. ಆರ್ಗ್ಯುಮೆಂಟ್ ಎನ್ನುವುದು ಕಮಾಂಡ್‌ಗೆ ನೀಡುವ ಇನ್ಪುಟ್ ಡಾಟಾ ಆಗಿದೆ. ಪ್ರದರ್ಶನ ಸ್ಕ್ರೀನ್ , ಡೀಫಾಲ್ಟ ಆಗಿ ಸ್ಟ್ಯಾಂಡರ್ಡ ಔಟ್ಪುಟ್ ಇರುತ್ತದೆ, ಆದರೆ ಇದನ್ನು ಫೈಲ್ ಅಥವಾ ಪ್ರಿಂಟರ್ ಮುಂತಾದವುಗಳಿಗೆ ಮರುನಿರ್ದೇಶಿಸಬಹುದು.

echo ಆಜ್ಞೆಯ ಸಿಂಟ್ಯಾಕ್ಸ್

$ echo $USER

$ echo $ Hello world

ಸ್ಕ್ವೇರ್ ಬ್ರಾಕೆಟ್‌ನಲ್ಲಿರುವ ವಸ್ತುಗಳು ಐಚ್ಛಿಕವಾಗಿವೆ. ಸ್ಟ್ರಿಂಗ್ ಎಂದರೆ ಸೀಮಿತ ಅಕ್ಷರಗಳ ಅನುಕ್ರಮವಾಗಿದೆ.(ಎಂದರೆ ಅಕ್ಷರಗಳು, ಸಂಖ್ಯೆಗಳು, ಚಿನ್ಹೆಗಳು, ಮತ್ತು ವಿರಾಮ ಚಿನ್ಹೆಗಳು)

ಯಾವುದೇ ಆಯ್ಕೆಗಳು ಅಥವಾ ಸ್ಟ್ರಿಂಗ್‌ಗಳಿಲ್ಲದೇ ಬಳಸಿದಾಗ ,echo ಆಜ್ಞೆಯು ಸ್ಕ್ರೀನ್‌ನಲ್ಲಿ ಖಾಲಿ ಸಾಲನ್ನು , ನಂತರದ ಸಾಲಿನಲ್ಲಿ ಕಮಾಂಡ್ ಪ್ರಾಂಪ್ಟನ್ನು ತೋರಿಸುತ್ತದೆ. ಕಾರಣ Enter ಕೀಲಿಯನ್ನು ಒತ್ತುವುದರಿಂದ ಸಿಸ್ಟಮ್‌ಗೆ ಹೊಸ ಸಾಲನ್ನು ಪ್ರಾರಂಬಿಸುವಂತೆ ಸಂಕೇತ ನೀಡುತ್ತದೆ ಮತ್ತು echo ಈ ಸಂಕೇತವನ್ನು ಪುನರಾವರ್ತಿಸುತ್ತದೆ.

ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳಾಗಿ ಒದಗಿಸಿದಾಗ, echo ಆಜ್ಞೆಯು ಈ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಪರದೆಯ ಮೇಲೆ ಪುನರಾವರ್ತಿಸುತ್ತದೆ. ಹೀಗಾಗಿ ಉದಾಹರಣೆಗೆ ಈ ಕೆಳಗಿನದನ್ನು ಟೈಪ್ ಮಾಡಿ ಮತ್ತು Enter ಕೀಲಿಯನ್ನು ಒತ್ತುವುದರಿಂದ echo ಆಜ್ಞೆಯು ಕೆಳಗಿನ ಪದವನ್ನು This is a pen ಪರದೆಯ ಮೇಲೆ ಪುನರಾವರ್ತಿಸುತ್ತದೆ :

echo This is a pen

ಸ್ಟ್ರಿಂಗ್‌ನೊಂದಿಗೆ ಕೋಟ್ (quote)ಉಪಯೋಗಿಸುವ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ. ಇದು ಪರದೆಯ ಮೇಲೆ ಏನು ಬರೆಯಲಾಗಿದೆ ಎನ್ನುವುದರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ. ಕೋಟ್ (quote)ಗಳನ್ನು ಉಪಯೋಗಿಸಿದರೆ ಅವುಗಳನ್ನು ಪರದೆಯ ಮೇಲೆ ತೋರಿಸಲಾಗುವುದಿಲ್ಲ.

ಅದೃಷ್ಟವಶಾತ್ , echo ಆಜ್ಞೆಯು ಕೇವಲ ಶಬ್ದಗಳನ್ನು ಪುನರಾವರ್ತಿಸುವುದಕ್ಕಿಂತ ಹೆಚ್ಚಿನದನ್ನೂ ಮಾಡಬಹುದು. ಎಂದರೆ ಇದು ಒಂದು ನಿರ್ದಿಷ್ಟ ವೇರಿಯೇಬಲ್‌ನ ಮೌಲ್ಯವನ್ನು ಸಹ ತೋರಿಸುತ್ತದೆ. ಇದಕ್ಕಾಗಿ ವೇರಿಯೇಬಲ್ ಹೆಸರು ನಂತರ ನೇರವಾಗಿ (ಎಂದರೆ ಮಧ್ಯದಲ್ಲಿ ಸ್ಥಳವಿಲ್ಲದೇ) ಡಾಲರ್ ಚಿನ್ಹೆಯನ್ನು ನೀಡಿದರೆ, ಇದು ಶೆಲ್‌ಗೆ ವೇರಿಯೇಬಲ್‌ನ ಮೌಲ್ಯವನ್ನು ತೋರಿಸಲು ಹೇಳುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ x ಹೆಸರಿನ ವೇರಿಯೇಬಲ್‌ನ್ನು ರಚಿಸಬಹುದು ಮತ್ತು ಈ ಕೆಳಗಿನಂತೆ ಅದರ ಮೌಲ್ಯವನ್ನು 5 ಕ್ಕೆ ಹೊಂದಿಸಬಹುದು :

x = 5

x ನ ಮೌಲ್ಯವನ್ನು ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯಂತೆ ನೆನಪಿಸಿಕೊಳ್ಳಬಹುದು.

echo The number is $x

ವೇರಿಯೇಬಲ್‌ನ ಮೌಲ್ಯಗಳನ್ನು ತೋರಿಸಲು echo ಆಜ್ಞೆಯು

© NIMI NOT TO BE REPUBLISHED

ಉಪಯುಕ್ತವಾಗಿದೆ. ಇದು ಬಳಕೆದಾರರು ಕಮಾಂಡ್ ಲೈನ್‌ನಲ್ಲಿ ಅಥವಾ ಸ್ಕ್ರಿಪ್ಟ (ಸಣ್ಣ ಪ್ರೋಗ್ರಾಂಗಳು) ನಲ್ಲಿ ಕೆಲಸ ಮಾಡುವಾಗ , ಶೆಲ್ ಹೇಗೆ ವರ್ತಿಸಬೇಕು ಎಂಬುದನ್ನು ಹೇಳುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ , HOME ನ ಮೌಲ್ಯವನ್ನು ನೋಡಲು , ಪ್ರಸ್ತುತ ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ತೋರಿಸುವ ಮೌಲ್ಯವನ್ನು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಬಳಸಲಾಗುತ್ತದೆ :

```
echo $HOME
```

ಅದರಂತೆ, echo ಆಜ್ಞೆಯನ್ನು ಬಳಕೆದಾರರ PATH ವೇರಿಯೇಬಲ್‌ನ್ನು ತೋರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು, ಕೋಲನ್ ಚಿನ್ಹೆಯಿಂದ ಬೇರ್ಪಡಿಸಿರುವ ಡೈರೆಕ್ಟರಿಗಳ ಪಟ್ಟಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಹಾಗೂ ಈ ಪಟ್ಟಿಯಲ್ಲಿ ಬಳಕೆದಾರರಿಂದ ನೀಡಲ್ಪಟ್ಟ ಕಮಾಂಡಗೆ ಅನುಗುಣವಾದ ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಪ್ರೋಗ್ರಾಂನ್ನು ಸಿಸ್ಟಮ್ ಹುಡುಕುತ್ತದೆ.

```
echo $PATH
```

echo ಡೀಫಾಲ್ಟ್ ಆಗಿ ಯಾವುದೇ ಔಟಪುಟ್‌ನ್ನು ನಂತರ ಹೊಸ ಸಾಲಿನೊಂದಿಗೆ ಅನುಸರಿಸುತ್ತದೆ. ಇದು ಮುದ್ರಣವಲ್ಲದ (ಎಂದರೆ ಅದೃಶ್ಯ) ಅಕ್ಷರವಾಗಿದೆ. ಇದು ಒಂದು ಪಠ್ಯದ ಸಾಲಿನ ಅಂತ್ಯ ಮತ್ತು ಮುಂದಿನ ಸಾಲಿನ ಪ್ರಾರಂಭವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇದನ್ನು ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ \n ನಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ. ಇದರಿಂದ ನಂತರದ ಆಜ್ಞೆಯ ಪ್ರಾಂಪ್ಟ ಅದೇ ಸಾಲಿನಲ್ಲಿ ಇರುವುದಕ್ಕಿಂತ ಹೊಸ ಸಾಲಿನಲ್ಲಿ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.ಏಕೆಂದರೆ ಆ ಸಾಲಿನಲ್ಲಿ echo ಆಜ್ಞೆಯ ಔಟಪುಟ್ ಇರುತ್ತದೆ.

-e ಆಯ್ಕೆಯನ್ನು echo ಆಜ್ಞೆಯ ಅರ್ಥ ವಿವರಣೆಯನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ನ್ಯೂಲೈನ್ ಅಕ್ಷರಗಳ ಹೆಚ್ಚುವರಿ ನಿದರ್ಶನಗಳು ಮತ್ತು \t ಎಂದು ಪ್ರತಿನಿಧಿಸಲ್ಪಡುವ ಹಾರಿಜಾಂಟಲ್ ಟ್ಯಾಬ್‌ನಂತಹ ವಿಶೇಷ ಅಕ್ಷರಗಳ ಅರ್ಥವಿವರಣೆಯನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಲು ಇದನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಹೀಗಾಗಿ , ಉದಾಹರಣೆ ಈ ಕೆಳಗಿನವುಗಳು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿದ ಔಟಪುಟನ್ನು ನೀಡುತ್ತವೆ.

```
echo -e \n ``Projects: \n\n\tplan \n\tcode \n\ttest\n``
```

(ಈ ಮೇಲಿನ ಆಜ್ಞೆಯನ್ನು ಒಂದೇ ಸಾಲಿನಲ್ಲಿ ಬರೆಯಬೇಕು. ಸಣ್ಣ ಸ್ಕ್ರೀನ್‌ನಲ್ಲಿ ಎರಡು ಸಾಲುಗಳಲ್ಲಿ ಕಾಣಬಹುದು) -n ಆಯ್ಕೆಯನ್ನು ಔಟಪುಟ್‌ಗೆ ಹೊಸ ಸಾಲನ್ನು ಸೇರಿಸುವುದನ್ನು ತಡೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಔಟಪುಟ್ ಪುನರ್ನಿರ್ದೇಶನವನ್ನು ಬಳಸುವುದರ ಮೂಲಕ echo ಆಜ್ಞೆಯನ್ನು , ಪಠ್ಯವನ್ನು ಹೊಂದಿರುವ ಹೊಸ ಫೈಲನ್ನು ರಚಿಸಲು ಅತಿ ಸರಳವಾದ ಮಾರ್ಗವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದನ್ನು echo ಆಜ್ಞೆಯನ್ನು ಟೈಪ್ ಮಾಡಿ ನಂತರ ಬಯಸಿದ ಪಠ್ಯವನ್ನು ಟೈಪ್ ಮಾಡಿ ನಂತರ ಔಟಪುಟ್ ರೀಡೈರೆಕ್ಷನ್ ಆಪರೇಟರ್ (ಇದು ಬಲಗಡೆ ಮುಖ ಮಾಡಿದ ಆ್ಯಗಲ್ ಬ್ರಾಕೆಟ್ ಆಗಿದೆ) ಮತ್ತು ಕೊನೆಗೆ ಹೊಸ ಫೈಲ್‌ನ ಹೆಸರು ಟೈಪ್ ಮಾಡುವ ಮೂಲಕ ಸಾಧಿಸಲಾಗುತ್ತದೆ. ಫೈಲನ್ನು ವಿಶೇಷ ಅಕ್ಷರಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಈ ಮೇಲಿನ ಉದಾಹರಣೆಯಿಂದ ಫಾರ್ಮ್ಯಾಟ ಮಾಡಿದ ಔಟಪುಟ್‌ನ್ನು project1 ಹೆಸರಿನ ಹೊಸ ಫೈಲನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಬಹುದು:

```
echo -e ``\n Project1: \n\n\tplan \n\twrite \n\ttest\n`` >project1
```

ಯಾವುದೇ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಸೇರಿದಂತೆ , ಹೊಸ ಫೈಲ್‌ನಲ್ಲಿರುವ ವಿಷಯಗಳನ್ನು , cat ಅಥವಾ less ಎನ್ನುವ ಆಜ್ಞೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಪರಿಶೀಲಿಸಬಹುದು.

```
less project1
```

echo ಆಜ್ಞೆಯನ್ನು ಇದರ ಜೊತೆಗೆ ಅಪೆಂಡ್ ಆಪರೇಟರನ್ನು ಉಪಯೋಗಿಸಿ ಫೈಲ್‌ನ ಕೊನೆಗೆ ಪಠ್ಯವನ್ನು ಸೇರಿಸಲು ಅನುಕೂಲಕರವಾದ ಮಾರ್ಗವಾಗಿದೆ. ಅಪೆಂಡ್ ಆಪರೇಟರನ್ನು ಬಲಕ್ಕೆ ಮುಖ ಮಾಡಿದ ಎರಡು ಸತತ ಆ್ಯಂಗಲ್ ಬ್ರಾಕೆಟ್‌ಗಳ ಮುಖಾಂತರ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ. ಅದಾಗ್ಯೂ ಎರಡು ಬ್ರಾಕೆಟ್‌ಗಳ ಬದಲು ಆಕಸ್ಮಿಕವಾಗಿ ಒಂದೇ ಬ್ರಾಕೆಟನ್ನು ಉಪಯೋಗಿಸುವ ಅಪಾಯವಿರುತ್ತದೆ. ಇದು ಫೈಲ್‌ನ ಎಲ್ಲಾ ವಿಷಯಗಳನ್ನು ತಿದ್ದಿ ಬರೆಯುತ್ತದೆ. ಮತ್ತು ಈ ವೈಶಿಷ್ಟ್ಯವನ್ನು ಸ್ಕ್ರಿಪ್ಟಗಳಲ್ಲಿ ಬಳಸಲು ಕಾಯ್ದಿರಿಸಲಾಗಿದೆ.

echo ಆಜ್ಞೆಯನ್ನು ವೈಲ್ಡಕಾರ್ಡ ಅಕ್ಷರಗಳಂತಹ ಪ್ಯಾಟರ್ನ್ ಮ್ಯಾಚಿಂಗ್‌ನೊಂದಿಗೆ ಸಹ ಬಳಸಬಹುದು. ಸ್ಟಾರ್ ಅಕ್ಷರ (*) ವೈಲ್ಡಕಾರ್ಡ ಅಕ್ಷರವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ jif ಫೈಲ್ ಗಳ ನಂತರ ಎಲ್ಲ .jif ಇಮೇಜ ಫೈಲ್‌ಗಳನ್ನು ಅನುಸರಿಸಲು ಈ ಕೆಳಗಿನಂತೆ ಆಜ್ಞೆಯನ್ನು ನೀಡಬೇಕು:

```
echo -e The gif files are *.gif
```

ಕ್ಯಾಲ್ ಕಮಾಂಡ್ (The cal command)

ಪ್ರಸ್ತುತ ತಿಂಗಳ ಕ್ಯಾಲೆಂಡರನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

```
$ cal
```

July 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Cal ಆಜ್ಞೆಯು ನಿರ್ದಿಷ್ಟ ತಿಂಗಳು ಮತ್ತು ವರ್ಷದ ಕ್ಯಾಲೆಂಡರನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

$ cal 08 1991

August 1991

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Date ಕಮಾಂಡ್ (Date command)

ಪ್ರಸ್ತುತ ಸಮಯ ಮತ್ತು ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

$ date

Fri Jul 6 01:07:09 IST 2012

ನೀವು ಸಮಯವನ್ನು ಮಾತ್ರ ಪ್ರದರ್ಶಿಸಲು ಆಸಕ್ತಿಯನ್ನು ಹೊಂದಿದ್ದರೆ, ನೀವು date +%T(in hh:mm:ss) ಎಂದು ಬಳಸಬಹುದು.

$ date +%T

01:13:14

tty ಕಮಾಂಡ್(tty command)

ಪ್ರಸ್ತುತ ಟರ್ಮಿನಲ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

$tty

/dev/pts/0

ಹೂ ಎಮ್ ಐ ಕಮಾಂಡ್(whoami command)

ಈ ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಲಾಗಿನ್ ಆಗಿರುವ ಬಳಕೆದಾರರನ್ನು ತೋರಿಸುತ್ತದೆ.

$ whoami

Raghu

ಐಡಿ ಕಮಾಂಡ್(id command)

ಈ ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಬಳಕೆದಾರರ ಹೆಸರು ಮತ್ತು ಗುಂಪುಗಳನ್ನು (UID ಮತ್ತು GID) ಮುದ್ರಿಸುತ್ತದೆ.

$ id

Uid=1000(raghu) gid=1000(raghu)

Groups=1000(raghu), 4(adm), 20(dialout), 24(cdrom), 46(plugdev), 112(lpadmin), 120(admin), 122(sambashare)

ಡೀಫಾಲ್ಟ್ ಆಗಿ ಪ್ರಸ್ತುತ ಬಳಕೆದಾರರ ಮಾಹಿತಿಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ ಮತ್ತು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಬೇರೆ ಬಳಕೆದಾರರ ಹೆಸರನ್ನು ನೀಡಿದರೆ, ಆ ಬಳಕೆದಾರರ ಮಾಹಿತಿಯನ್ನು, ಮುದ್ರಿಸುತ್ತದೆ.

$ id root

Uid=0(root) gid=0(root) groups=0(root)

ಕ್ಲಿಯರ್ ಕಮಾಂಡ್(Clear command)

ಈ ಆಜ್ಞೆಯು ಸ್ಕ್ರೀನ್‌ನ್ನು ತೆರವುಗೊಳಿಸುತ್ತದೆ.

help ಕಮಾಂಡ್(Getting help command)

ಕಮಾಂಡ್ ಲೈನ್‌ನ ದೊಡ್ಡ ಅನಾನುಕೂಲತೆ ಎಂದರೆ ಇದರಲ್ಲಿ ಅನೇಕ ಆಜ್ಞೆಗಳಿವೆ ಮತ್ತು ಇನ್ನೂ ಹೆಚ್ಚಿನ ಅವುಗಳ ಆಯ್ಕೆಗಳಿವೆ ಮತ್ತು ಉಪಯೋಗಗಳಿವೆ. ಆದರೆ ಎಲ್ಲರಿಗೂ ಎಲ್ಲ ಆಜ್ಞೆಗಳು ನೆನಪಿನಲ್ಲಿರುವುದಿಲ್ಲ. ಕಮಾಂಡ್ ಲೈ ನನ್ನು ಬಳಸುವ ಒಳ್ಳೆಯ ಮಾರ್ಗಗಳಿವೆ. ಲಿನಕ್ಸ ನಮಗೆ ಅಂತಹ ಹಲವರು ಮಾರ್ಗಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಅಂತಹ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಇಲ್ಲಿ ಚರ್ಚಿಸಲಾಗಿದೆ:

--help option

ಪ್ರತಿಯೊಂದು ಆಜ್ಞೆಯೊಂದಿಗೆ --help ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಆಜ್ಞೆಯ ಸಾರಾಂಶ ಈ ಕೆಳಗಿನಂತಿದೆ

$ date --help

Usage : date [OPTION] ----[+FORMAT]

Or : date [-u |--utc | --universal] [MMDDhhmm[[CC]YY][.ss]]

ಕೊಟ್ಟಿರುವ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಪ್ರಸ್ತುತ ಸಮಯವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಅಥವಾ ಸಿಸ್ಟಮ್ ದಿನಾಂಕವನ್ನು ಹೊಂದಿಸುತ್ತದೆ.

whatis ಕಮಾಂಡ್(whatis command)

whatis ಆಜ್ಞೆಯು ಕಮಾಂಡ್ ಲೈನ್ ಪ್ರೊಗ್ರಾಂ ಗಳ.(ಅಂದರೆ ಎಲ್ಲ ಪಠ್ಯ ಮಾದರಿಯ ಪ್ರೋಗ್ರಾಂ ಗಳು) ಮತ್ತು ಲಿನಕ್ಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ಇತರ ವಿಷಯಗಳ ಸಂಕ್ಷಿಪ್ತ ವಿವರಣೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಇದು whatis ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ಆರ್ಗ್ಯುಮೆಂಟ್ (ಎಂದರೆ ಇನ್ಪುಟ್ ಡಾಟಾ)ಆಗಿ ಒದಗಿಸಲ್ಪಟ್ಟ ಪ್ರತಿ ಕೀವರ್ಡ್‌ನ ಸಣ್ಣ ವಿವರಣೆಗಳನ್ನು ಹುಡುಕುವ ಮೂಲಕ ಇದನ್ನು ಸಾಧಿಸುತ್ತದೆ. ಹೆಚ್ಚಿನ ಯುನಿಕ್ಸ್ ತರಹದ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಅಂತರ್ನಿರ್ಮಿತವಾದ , ಈ ಡಾಟಾಬೇಸ್, ಮ್ಯಾನ್ ಕೈಪಿಡಿಯ ಪ್ರತಿ ಪುಟದ NAME ವಿಭಾಗದಿಂದ ಕೇವಲ ಶೀರ್ಷಿಕೆ , ವಿಭಾಗ ಸಂಖ್ಯೆ ಮತ್ತು ವಿವರಣೆಯನ್ನು ಹೊಂದಿರುತ್ತದೆ.

Whatis ನ ಸಿಂಟ್ಯಾಕ್ಸ್

Whatis keyword(s)

ಉದಾಹರಣೆಗೆ ಈ ಕೆಳಗಿನವುಗಳು ಹೆಡ್ ಕಮಾಂಡ್‌ನ ಒಂದೇ ಸಾಲಿನ ಸಾರಾಂಶವನ್ನು ಒದಗಿಸುತ್ತವೆ.(ಇದು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಒದಗಿಸಲ್ಪಟ್ಟ ಪ್ರತಿ ಫೈಲ್ ನ ಮೊದಲನೇ ಹತ್ತು ಸಾಲುಗಳನ್ನು ಡೀಫಾಲ್ಟ ಆಗಿ ತೋರಿಸುತ್ತದೆ.)

Whatis head

ಏಕಕಾಲದಲ್ಲಿ ಅನೇಕ ವಿಷಯಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಹುಡುಕಲು Whatis ಅನ್ನು ಬಳಸಬಹುದು. ಉದಾಹರಣೆಗೆ ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಯು

© NIMI NOT TO BE REPUBLISHED

head ಮತ್ತು tail ಎರಡರ ಬಗ್ಗೆಯೂ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ (ಫೈಲ್‌ನ ಕೊನೆಯ ಹತ್ತು ಸಾಲುಗಳನ್ನು ಡೀಫಾಲ್ಟ ಆಗಿ ಓದುತ್ತದೆ.)

Whatis head tail

Whatis ನ ಔಟಪುಟ್, ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ಕಂಡುಬರುವ ಪ್ರತಿ ಕೀವರ್ಡ್‌ಗೆ ಕೇವಲ ಒಂದೇ ಸಾಲನ್ನು ಒದಗಿಸುವುದಕ್ಕೆ ಸೀಮಿತವಾಗಿದೆ. ಅದ್ದರಿಂದ ಇದು ಮಧ್ಯಮ ಸಂಕೀರ್ಣ ಆಜ್ಞೆಗಳಿಗೆ ಕೂಡ ಅಪೂರ್ಣ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , ಕ್ಯಾಟ್ ಆಜ್ಞೆಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು Whatis ಅನ್ನು ಕೆಳಗಿನಂತೆ ಉಪಯೋಗಿಸಿದಾಗ ``concatenate files and print on the standard output``

whatis cat

ಅದಾಗ್ಯೂ ಇದು ಕ್ಯಾಟ್ ಬಗ್ಗೆ ಕೆಲವು ಪ್ರಮುಖ ಮಾಹಿತಿಯನ್ನು ಬಿಟ್ಟುಬಿಡುತ್ತದೆ. ವಿಶೇಷವಾಗಿ ಇದನ್ನು ಉಪಯೋಗಿಸಲು ಅನುಕೂಲಕರವಾದ ಕೆಲವು ಅಂಶಗಳಾದ ಫೈಲ್‌ಗಳನ್ನು ಓದುವುದು ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ರಚಿಸಲು ಮತ್ತು ಬರೆಯಲು ಬಳಸಲಾಗುತ್ತದೆ.

whatis ಇದು apropos ಆಜ್ಞೆಯನ್ನು ಹೋಲುತ್ತದೆ. ಅದಾಗ್ಯೂ, apropos ಹೆಚ್ಚು ಶಕ್ತಿಯುತವಾಗಿದೆ. ಇದರ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು ಸಂಪೂರ್ಣ ಶಬ್ದಗಳಿಗೆ ಸೀಮಿತವಿರದೇ ಶಬ್ದಗಳ ಭಾಗಗಳನ್ನೊಳಗೊಂಡ ಸ್ಟ್ರಿಂಗ್ (ಎಂದರೆ ಯಾವುದೇ ಸೀಮಿತ ಅಕ್ಷರಗಳ ಗುಂಪು) ಗಳಾಗಿಯೂ ಇರಬಹುದು. ಎರಡೂ ಆಜ್ಞೆಗಳಲ್ಲಿ ಆಯ್ಕೆಗಳಿರುವುದಿಲ್ಲ.

ಮ್ಯಾನ್ ಆಜ್ಞೆಯು (ಇದನ್ನು ಅಂತರ್ನಿರ್ಮಿತ ಮ್ಯಾನ್ಯುಯಲ್ ಪುಟಗಳನ್ನು ಓದಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.) ಅದರ -f ಆಯ್ಕೆಯೊಂದಿಗೆ ಬಳಸಿದಾಗ whatis ನಂತೆಯೆ ಔಟಪುಟ್‌ನ್ನು ಉತ್ಪಾದಿಸುತ್ತದೆ. ಅದ್ದರಿಂದ ಉದಾಹರಣೆಗೆ,

man -f cat

ಈ ಕೆಳಗಿನ ಆಜ್ಞೆಗೆ ಸಮನಾಗಿರುತ್ತದೆ.

WHATIS CAT

ಇನ್ಫೋ ಪೇಜಸ್(Info pages)

ಇನ್ಫೋ ದಾಖಲೆಗಳು ಕೆಲವೊಮ್ಮೆ ಮ್ಯಾನ್ ಪುಟಗಳಿಗಿಂತ ಹೆಚ್ಚು ವಿಸ್ತಾರವಾಗಿರುತ್ತವೆ. ಆದರೆ ಕೆಲವು ಆಜ್ಞೆಗಳಿಗೆ, ಇನ್ಫೋ ಪುಟಗಳು ಮ್ಯಾನ್ ಪುಟಗಳಂತೆಯೆ ಇರುತ್ತವೆ. ಇವು ವೆಬ್ ಪುಟಗಳಂತೆ ಇರುತ್ತವೆ. ಆಂತರಿಕ ಲಿಂಕ್‌ಗಳು ಇನ್ಫೋ ಪುಟಗಳಲ್ಲಿ ಇರುತ್ತವೆ. ಈ ಲಿಂಕ್‌ಗಳನ್ನು ನೋಡ್‌ಗಳು ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಈ ನೋಡ್‌ಗಳ ಮುಖಾಂತರ ಇನ್ಫೋ ಪುಟಗಳನ್ನು ಒಂದು ಪುಟದಿಂದ ಇನ್ನೊಂದು ಪುಟಕ್ಕೆ ನ್ಯಾವಿಗೆಟ್ ಮಾಡಬಹುದು.

$ info date

ಲಿನಕ್ಸ್ ಪರಿಸರದಲ್ಲಿ ವರ್ಡ್ ಪ್ರೊಸೆಸರ್‌ಗಳು(Word processors in the Linux environment)

ಟೆಕ್ಸ್ ಎಡಿಟರ್‌ಗಳನ್ನು ವಿವಿಧ ರೀತಿಯ ಅನೇಕ ಜನರು ಉಪಯೋಗಿಸುತ್ತಾರೆ. ಸಾಂದರ್ಭಿಕ ಬಳಕೆದಾರರು, ಬರಹಗಾರರು, ಪ್ರೋಗ್ರಾಮರ್‌ಗಳು, ಮತ್ತು ಸಿಸ್ಟಮ್ ನಿರ್ವಾಹಕರು ಎಲ್ಲರೂ ಒಂದಲ್ಲ ಒಂದು ಸಮಯದಲ್ಲಿ ಟೆಕ್ಸ್ ಎಡಿಟರ್‌ಗಳನ್ನು ಲಿನಕ್ಸ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸುತ್ತಾರೆ.

ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನ ಬಳಕೆ(Use of Text editor)

ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್ ಇದು ಬಹಳ ವೈಶಿಷ್ಟ್ಯಗಳಿಲ್ಲದ ವರ್ಡ್ ಪ್ರೊಸೆಸರ್‌ನಂತಿದೆ. ಎಲ್ಲ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು ಒಂದು ಮೂಲ ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನೊಂದಿಗೆ ಬರುತ್ತವೆ. ಲಿನಕ್ಸ್ ಬಹಳ ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನ ಮುಖ್ಯ ಉಪಯೊಗ ಎಂದರೆ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಇಲ್ಲದೇ ಸರಳ ಪಠ್ಯವಾಗಿ ಏನನ್ನಾದರೂ ಬರೆಯುವುದಕ್ಕೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದರಿಂದ ಇನ್ನೊಂದು ಪ್ರೊಗ್ರಾಂ ಅದನ್ನು ಓದಬಹುದು. ಫೈಲ್‌ನಿಂದ ಅದು ಪಡೆಯುವ ಮಾಹಿತಿಯನ್ನು ಆಧರಿಸಿ, ಪ್ರೊಗ್ರಾಂ ಬೇರೆ ಬೇರೆ ರೀತಿಯಿಂದ ರನ್ ಆಗುತ್ತದೆ.

(Movement commands):

ನಿಮ್ಮ ಸಿಸ್ಟಮ್‌ನ್ನು ಅವಲಂಬಿಸಿದಂತೆ, Arrow ಕೀಗಳು ಅಥವಾ backspace ಕೀಯು ಕಾರ್ಯ ನಿರ್ವಹಿಸದೇ ಇರಬಹುದು, ಅವುಗಳ ಬದಲಾಗಿ ನೀವು ಅದೇ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೊಗಿಸಬಹುದು.

TO	Hold down Ctrl key and press	Instead of
ಅಕ್ಷರವನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು	ಬ್ಯಾಕ್ ಸ್ಪೇಸ್	backspace
ಸಾಲನ್ನು ಮೇಲಕ್ಕೆ ಸರಿಸಲು	p	up arrow
ಸಾಲನ್ನು ಕೆಳಗೆ ಸರಿಸಲು	n	down arrow
ಒಂದು ಸ್ಥಾನವನ್ನು ಎಡಕ್ಕೆ ಸರಿಸಲು	b	left arrow
ಒಂದು ಸ್ಥಾನವನ್ನು ಬಲಕ್ಕೆ ಸರಿಸಲು	f	right arrow
ಸಾಲಿನ ಕೊನೆಗೆ ಚಲಿಸಲು	e	end

NOT TO BE REPUBLISHED © NIMI

vi	ಒಂದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್‌ಗಳನ್ನು ತೆರೆಯಲು ಕಮಾಂಡ್ ಲೈನ್‌ನಲ್ಲಿ ಇದನ್ನು ಟೈಪ್ ಮಾಡಲಾಗುತ್ತದೆ(vi tomato.txt ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ tomato.txt ಎನ್ನುವ ಹೆಸರಿನ ಫೈಲನ್ನು ತೆರೆಯುತ್ತದೆ.)(vi parsley sage rosemary ಎನ್ನುವ ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ parsley sage rosemary ಎನ್ನುವ ಮೂರು ಫೈಲ್‌ಗಳನ್ನು ತೆರೆಯುತ್ತದೆ)
vi *	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಫೈಲ್‌ನ್ನು ತೆರೆಯಲು ಇದನ್ನು ಕಮಾಂಡ್ ಲೈ ನ್ ನಲ್ಲಿ ಟೈಪ್ ಮಾಡಲಾಗುತ್ತದೆ
:q	ಯಾವುದೇ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡದ ಫೈಲ್‌ಗಳನ್ನು ಮುಚ್ಚುತ್ತದೆ.(ಕ್ವಿಟ್ ಮಾಡುತ್ತದೆ)
:q!	ಯಾವುದೇ ಬದಲಾವಣೆಗಳನ್ನು ಸೇವ್ ಮಾಡದೇ ನಿರ್ಗಮಿಸುತ್ತದೆ.
:w	ಪ್ರಸ್ತುತ ಫೈಲನ್ನು ಡಿಸ್ಕನಲ್ಲಿ ಸೇವ್ ಮಾಡುತ್ತದೆ.
:wq	ಬಫರ್‌ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಡಿಸ್ಕನಲ್ಲಿ ಸೇವ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ನಿರ್ಗಮಿಸುತ್ತದೆ
Zz	:wq ನಂತೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.
i	ಟೆಕ್ಸ್ಟ್ ಇನ್ಸರ್ಟ್ ಮೋಡನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ , ಕರ್ಸರ್‌ನ ಪ್ರಸ್ತುತ ಸ್ಥಾನದಲ್ಲಿ ತಕ್ಷಣವೆ ಟೆಕ್ಸ್ಟ್ ಅನ್ನು ಅಳವಡಿಸುತ್ತದೆ.
h	ಕರ್ಸರನ್ನು ಒಂದು ಅಕ್ಷರ ಎಡಕ್ಕೆ ಚಲಿಸುತ್ತದೆ.
j	ಕರ್ಸರನ್ನು ಒಂದು ಸಾಲು ಕೆಳಕ್ಕೆ ಚಲಿಸುತ್ತದೆ.
k	ಕರ್ಸರನ್ನು ಒಂದು ಸಾಲು ಮೇಲಕ್ಕೆ ಚಲಿಸುತ್ತದೆ.
l	ಕರ್ಸರನ್ನು ಒಂದು ಅಕ್ಷರ ಬಲಕ್ಕೆ ಚಲಿಸುತ್ತದೆ.
G	ಕರ್ಸರನ್ನು ನೀವು ಅಪೇಕ್ಷಿಸಿದ ಸ್ಥಾನಕ್ಕೆ ಚಲಿಸುತ್ತದೆ. ಮಾರ್ಪಡಿಸುವ ಪೂರ್ಣಾಂಕ ಇಲ್ಲದಿದ್ದರೆ, ಟೆಕ್ಸ್ಟನ ಕೊನೆ ಸಾಲಿಗೆ ಕರ್ಸರನ್ನು ಚಲಿಸುತ್ತದೆ. (5G ಇದು ಕರ್ಸರನ್ನು ಐದನೇ ಸಾಲಿಗೆ ಚಲಿಸುತ್ತದೆ.)
a	ಇನ್ಸರ್ಟ್ ಮೋಡ್‌ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ. ಮತ್ತು ತಕ್ಷಣವೇ ಕರ್ಸರ್‌ನ ಬಲಕ್ಕೆ ಟೆಕ್ಸ್ಟನ್ನು ಅಳವಡಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.
x	ಕರ್ಸರನ ಅಡಿಯಲ್ಲಿರುವ ಅಕ್ಷರವನ್ನು ತಕ್ಷಣವೇ ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ. (xxx ಕರ್ಸರ್ ನ ಅಡಿಯಲ್ಲಿ ಬರುವ ಅಕ್ಷರವನ್ನು ತಕ್ಷಣವೇ ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ. ಮತ್ತು ನಂತರ ಅದರ ಬಲಭಾಗದಲ್ಲಿ ಬರುವ ಎರಡು ಅಕ್ಷರಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ)
X	ಕರ್ಸರ್‌ನ ಎಡಭಾಗದಲ್ಲಿರುವ ಒಂದೇ ಅಕ್ಷರವನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ.
D	ಪ್ರಸ್ತುತ ಸಾಲಿನ ಕರ್ಸರ್ ನ ಅಡಿಯಲ್ಲಿ ಬರುವ ಅಕ್ಷರದಿಂದ ಸಾಲಿನ ಅಂತ್ಯದವರೆಗಿನ ಟೆಕ್ಸ್ಟನ್ನು ಅಳಿಸುತ್ತದೆ.
dw	ಕರ್ಸರ್‌ನ ಅಡಿಯಲ್ಲಿ ಬರುವ ಅಕ್ಷರವನ್ನು ಮತ್ತು ಅದೇ ಶಬ್ದದ ಬಲಭಾಗದಲ್ಲಿ ಬರುವ ಉಳಿದ ಅಕ್ಷರಗಳನ್ನು ತಕ್ಷಣ ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ.
dd	ಕರ್ಸರ್ ಇರುವ ಇಡೀ ಸಾಲನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ನಂತರ ಕರ್ಸರ್ ಮುಂದಿನ ಸಾಲಿಗೆ ಚಲಿಸುತ್ತದೆ.
cw	ಕರ್ಸರ್ ನ ಅಡಿಯಲ್ಲಿ ಬರುವ ಅಕ್ಷರವನ್ನು ಮತ್ತು ಅದೇ ಶಬ್ದದ ಬಲಭಾಗದಲ್ಲಿರುವ ಅಕ್ಷರವನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅವುಗಳ ಬದಲಾಗಿ ಹೊಸ ಅಕ್ಷರಗಳನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. (2cw ಆಜ್ಞೆಯು ಕರ್ಸರ್ ನ ಅಡಿಯಲ್ಲಿರುವ ಅಕ್ಷರವನ್ನು ಮತ್ತು ಅದೇ ಶಬ್ದದ ಬಲಭಾಗದಲ್ಲಿರುವ ಅಕ್ಷರವನ್ನು ಮತ್ತು ಮುಂದಿನ ಶಬ್ದದ ಅಕ್ಷರವನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ನಂತರ ಇವುಗಳ ಬದಲಾಗಿ ಬೇರೆ ಅಕ್ಷರಗಳನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.)
cc	ಪ್ರಸ್ತುತ ಸಾಲನ್ನು ಅಳಿಸುತ್ತದೆ ಮತ್ತು ಅದರ ಬದಲಾಗಿ ಬೇರೆ ಟೆಕ್ಸ್ಟನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. (2cc ಆಜ್ಞೆಯು ಪ್ರಸ್ತುತ ಸಾಲನ್ನು ಮತ್ತು ಮುಂದಿನ ಸಾಲನ್ನು ಅಳಿಸುತ್ತದೆ ಹಾಗೂ ಎರಡೂ ಸಾಲುಗಳಲ್ಲಿ ಬದಲಿ ಪಠ್ಯವನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.
cb	ಪ್ರಸ್ತುತ ಶಬ್ದದಲ್ಲಿರುವ ಕರ್ಸರ್‌ನ ಎಡಭಾಗದಲ್ಲಿರುವ ಅಕ್ಷರಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅದರ ಬದಲಾಗಿ ಬೇರೆ ಪಠ್ಯವನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.
R	ಟೆಕ್ಸ್ಟ್ ಇನ್ಪುಟ್ ಮೋಡನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ಇದು ಕರ್ಸರ್ ಅಡಿಯಲ್ಲಿ ಮತ್ತು ಅದರ ಬಲಭಾಗದಲ್ಲಿ ಒಂದು ಸಮಯದಲ್ಲಿ ಒಂದು ಅಕ್ಷರವನ್ನು ಟೈಪ್ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ.
xp	ಎರಡು ಪಕ್ಕದ ಅಕ್ಷರಗಳನ್ನು ವರ್ಗಾಯಸುತ್ತದೆ.
deep	ಎರಡು ಪಕ್ಕದ ಶಬ್ದಗಳನ್ನು ವರ್ಗಾಯಿಸುತ್ತದೆ.

©NIMI NOT TO BE REPUBLISHED

ddp	ಎರಡು ಪಕ್ಕದ ಸಾಲುಗಳನ್ನು ವರ್ಗಾಯಿಸುತ್ತದೆ.
~	ಕರ್ಸರ್‌ನ ಅಡಿಯಲ್ಲಿ ಬರುವ ಅಕ್ಷರದ ಕೇಸ್ (ಅಪ್ಪರ್ ಕೇಸ್ ಮತ್ತು ಲೋವರ್ ಕೇಸ್) ಬದಲಾಯಿಸುತ್ತದೆ.
J	ಪ್ರಸ್ತುತ ಸಾಲನ್ನು ಮುಂದಿನ ಸಾಲಿನೊಂದಿಗೆ ಸೇರಿಸುತ್ತದೆ.
u	ಬಫರ್‌ನ್ನು ಬದಲಾಯಿಸಿದ ಇತ್ತೀಚಿನ ಆಜ್ಞೆಯ ಪರಿಣಾಮವನ್ನು ಹಿಮ್ಮುಖಗೊಳಿಸುತ್ತದೆ.
U	ಪ್ರಸ್ತುತವಾಗಿ , ಪ್ರಸ್ತುತ ಸಾಲಿನಲ್ಲಿ ಮಾಡಿದ ಎಲ್ಲಾ ಬದಲಾವಣೆಗಳನ್ನು ರದ್ದುಗೊಳಿಸುತ್ತದೆ.
:s/	ಟೆಕ್ಸ್ಟನಲ್ಲಿ , ಹೆಸರಿಸಿದ ಸ್ಟ್ರಿಂಗ್‌ಗಾಗಿ ಹುಡುಕಾಡುತ್ತದೆ. (:s/cucumber ಇದು ಮೊದಲು ಬರುವ cucumber ಈ ಶಬ್ದಕ್ಕಾಗಿ ಪಠ್ಯದಲ್ಲಿ ಹುಡುಕಾಡುತ್ತದೆ)
n	ಎರಡನೇ ಬಾರಿಗೆ ಪಠ್ಯದಲ್ಲಿ ಬರುವ ಹೆಸರಿಸಿದ ಶಬ್ದಕ್ಕಾಗಿ ಹುಡುಕಾಡುತ್ತದೆ.
:s/ / /	ಹೆಸರಿಸಿದ ಶಬ್ದದ ಮೊದಲನೇ ನಿದರ್ಶನವನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.(:s/cucumber/radish/ ಇದು cucumber ಶಬ್ದದ ಮೊದಲನೆ ನಿದರ್ಶನವನ್ನು radish ಶಬ್ದದೊಂದಿಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.
:%s/ / /	ಹೆಸರಿಸಿದ ಸ್ಟ್ರಿಂಗ್‌ನ ಪ್ರತಿಯೊಂದು ನಿದರ್ಶನವನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ. .(:s/cucumber/radish/ ಇದು cucumber ಶಬ್ದದ ಪ್ರತಿ ನಿದರ್ಶನವನ್ನು radish ಶಬ್ದದೊಂದಿಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.
:r	ಪ್ರಸ್ತುತ ತೆರೆದಿರುವ ಫೈಲ್‌ಗೆ ಬೇರೆ ಫೈಲ್‌ನಿಂದ ಟೆಕ್ಸ್ಟನ್ನು ಸೇರಿಸುತ್ತದೆ. (:r lettuce.txt ಇದು ಪ್ರಸ್ತುತ ತೆರೆದ ಫೈಲ್‌ಗೆ lettuce.txt ಫೈಲ್‌ನಿಂದ ಟೆಕ್ಸ್ಟನ್ನು ಸೇರಿಸುತ್ತದೆ.
:w>>	ಪ್ರಸ್ತುತ ತೆರೆದ ಫೈಲ್‌ನಿಂದ ಟೆಕ್ಸ್ಟನ್ನು ಬೇರೆ ಫೈಲ್‌ಗೆ ಸೇರಿಸುತ್ತದೆ.(:w>> cabbage ಪ್ರಸ್ತುತ ತೆರೆದಿರುವ ಫೈಲ್‌ನಿಂದ ಟೆಕ್ಸ್ಟ ಅನ್ನು cabage ಎನ್ನುವ ಫೈಲ್‌ಗೆ ಸೇರಿಸುತ್ತದೆ.

ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Managing files and directories)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ನಿಭಾಯಿಸುವುದನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ**
- **ಮೂಲ ಫೈಲ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ**
- **ಇತರ ಫೈಲ್ ಕಮಾಂಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ**
- **ಲಿನಕ್ಸ್ ಓಎಸ್‌ನಲ್ಲಿರುವ ಇತರ ಉಪಯುಕ್ತ ಕಮಾಂಡ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ.**

ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ನಿಭಾಯಿಸುವುದು(Manipulating files and directories): ಲಿನಕ್ಸ್ ಉಪಯೋಗಿಸುವುದು ಯಾವುದೇ ಇತರ ಕಂಪ್ಯೂಟರ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಿಂತ ಭಿನ್ನವಾಗಿಲ್ಲ. ನೀವು ನಿಮ್ಮ ಹಾರ್ಡ್ ಡಿಸ್ಕ್ ಡ್ರೈವ್ ನಲಿ ಮಾಹಿತಿಯನ್ನು ಸಂಘಟಿಸುವ ಸಲುವಾಗಿ ಫೈಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು, ಡಿಲೀಟ್ ಮಾಡುವುದು, ಮತ್ತು ಚಲಿಸುವುದನ್ನು ಮಾಡುತ್ತೀರಿ. ಈ ವಿಭಾಗವು ಈ ಕಾರ್ಯಗಳನ್ನು ತ್ವರಿತವಾಗಿ ಮತ್ತು ಸುಲಭವಾಗಿ ಹೇಗೆ ಮಾಡಬೇಕೆಂದು ತೋರಿಸುತ್ತದೆ.

ಲಿನಕ್ಸ್ ನ ಗ್ರಾಫಿಕಲ್ ಇಂಟರ್‌ಫೇಸ್ , X ವಿಂಡೋ ಸಿಸ್ಟಮ್ , ಡ್ರ್ಯಾಗ್ ಎಂಡ್ ಡ್ರಾಪ್ ಅಥವಾ ಫೈಲ್‌ಗಳನ್ನು ಕಾಪಿ ಅಥವಾ ಡಿಲೀಟ್ ಮಾಡಲು ಹಲವಾರು ಆಯ್ಕೆಗಳನ್ನು ನೀಡಬಹುದು. ಇಲ್ಲಿ ನೀವು ಅನೇಕ ಕಮಾಂಡ್‌ಗಳನ್ನು ಕಲಿಯುವಿರಿ. ಲಿನಕ್ಸ್ ಅನ್ನು ನೀವು ಕನ್ಸೋಲ್ ಮೋಡ್‌ನಲ್ಲಿ ಬಳಸದಿದ್ದರೂ, ಈ ಪ್ರೊಗ್ರಾಂಗಳು ಹೇಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ ಎಂದು ನೀವು ಕಲಿಯುತ್ತೀರಿ.

ಫೈಲ್‌ಗಳೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು(Working with files):

ಈ ಅಧ್ಯಾಯದಲ್ಲಿ ನಾವು file, touch, rm, cp, mv ಮತ್ತು rename ಮುಂತಾದ ಕಮಾಂಡ್‌ಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಫೈಲ್‌ಗಳನ್ನು, ಗುರುತಿಸುವುದು, ರಚಿಸುವುದು, ತೆಗೆದುಹಾಕುವುದು, ಕಾಪಿ ಮಾಡುವುದು, ಮತ್ತು ಮೂವ್ ಮಾಡುವುದನ್ನು ತಿಳಿಯೋಣ. ಎಲ್ಲ ಫೈಲ್‌ಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿರುತ್ತವೆ.

ಲಿನಕ್ಸ್‌ನಲ್ಲಿರುವ ಫೈಲ್‌ಗಳು (ಅಥವಾ ಯಾವುದೇ ಯುನಿಕ್ಸ್) ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿರುತ್ತವೆ. ಇದರರ್ಥ FILE1 ಇದು file1 ಗಿಂತ ಭಿನ್ನವಾಗಿರುತ್ತದೆ. ಮತ್ತು /etc/hosts ಇದು /etc/Hosts ಗಿಂತ ಭಿನ್ನವಾಗಿರುತ್ತದೆ. (ಎರಡನೆಯದ್ದು ಸಾಮಾನ್ಯ ಲಿನಕ್ಸ್ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಇರುವುದಿಲ್ಲ).

ಫೈಲ್ ಕಮಾಂಡ್ (The file command):

ಫೈಲ್ ಕಮಾಂಡ್ ತನಗೆ ಆರ್ಗ್ಯುಮೆಂಟ್ (ಎಂದರೆ ಇನ್ಪುಟ್) ಆಗಿ

© NIMI NOT TO BE REPUBLISHED

ಒದಗಿಸಲ್ಪಟ್ಟ ಪ್ರತಿಯೊಂದು ಫೈಲ್ ಸಿಸ್ಟಮ್ ಆಬ್ಜೆಕ್ಟ್ (ಎಂದರೆ, ಫೈಲ್, ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಲಿಂಕ್) ಅನ್ನು ವರ್ಗೀಕರಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತದೆ. ಈ ರೀತಿ ಇದು ಸಾಮಾನ್ಯವಾಗಿ ತಕ್ಷಣವೆ ನಿರ್ದಿಷ್ಟ ಆಬ್ಜೆಕ್ಟ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಉದಾ GIF89 ಇಮೇಜ್ ಫೈಲ್, ಡೈರೆಕ್ಟರಿ, GNU ಟಾರ್ ಆರ್ಕೈವ್, ASCII ಇಂಗ್ಲೀಷ್ ಟೆಕ್ಸ್ಟ್, ಸಿಂಬಾಲಿಕ್ ಲಿಂಕ್, HTML ಡಾಕ್ಯುಮೆಂಟ್, ಖಾಲಿ ಫೈಲ್, bzip2 ಕಾಂಪ್ರೆಸ್ಡ್ ಡಾಟಾ, ELF 32 bit LSB ಎಕ್ಸಿಕ್ಯೂಟೇಬಲ್ ಇತ್ಯಾದಿ.

ಪ್ರತಿ ಆಬ್ಜೆಕ್ಟನ್ನು ಮೂರು ವಿಧದಿಂದ ಪರೀಕ್ಷಿಸುವ ಮೂಲಕ ಮತ್ತು ಒಂದು ಯಶಸ್ವಿಯಾಗುವವರೆಗೆ ಫೈಲ್ ಇದನ್ನು ಸಾಧಿಸುತ್ತದೆ. ಮೊದಲನೆಯದು ಫೈಲ್ ಸಿಸ್ಟಮ್ ಪರೀಕ್ಷೆ, ಇದು ಆಬ್ಜೆಕ್ಟ ಐನೋಡ್(inode)(ಫೈಲ್ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ) ನಿಂದ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು ಸ್ಟಾಟ್ ಸಿಸ್ಟಮ್ ಕಾಲ್‌ನ್ನು ಬಳಸುತ್ತದೆ. ಸಿಸ್ಟಮ್ ಕಾಲ್, ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ನಲ್ಲಿ ಕರ್ನಲ್ (ಎಂದರೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಂ ನ ಕೋರ್) ನಿರ್ವಹಿಸುವ ಸೇವೆಗಾಗಿ ಒಂದು ವಿನಂತಿಯಾಗಿದೆ.

ಎರಡನೆಯ ಪರೀಕ್ಷೆ, ಅಲ್ಲಿ ಮ್ಯಾಜಿಕ್ ಸಂಖ್ಯೆ ಇದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಹಲವಾರು ವಿಧದ ಫೈಲ್‌ಗಳ ಪ್ರಾರಂಭದಲ್ಲಿ ಅಥವಾ ಹತ್ತಿರದಲ್ಲಿ ಇರುವ ಈ ಸಂಖ್ಯೆ ಫೈಲ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ(ಎಂದರೆ ಫೈಲ್‌ನ ವಿಧ)

ಮೊದಲ ಎರಡು ಪರೀಕ್ಷೆಗಳು ಫೈಲ್‌ನ ವಿಧವನ್ನು ನಿರ್ಧರಿಸಲು ವಿಫಲವಾದ ಸಂದರ್ಭದಲ್ಲಿ, ಇದು ಸರಳವಾದ ಟೆಕ್ಸ್ಟ ಫೈಲ್(ಎಂದರೆ ಸಂಪೂರ್ಣವಾಗಿ ಮಾನವರು ಓದಬಲ್ಲ ಅಕ್ಷರಗಳನ್ನು ಹೊಂದಿದ) ಎಂದು ನಿರ್ಧರಿಸಲು ಲ್ಯಾಂಗ್ವೇಜ್ ಪರೀಕ್ಷೆಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಹಾಗಿದ್ದರೆ ಯಾವ ರೀತಿಯ ಸರಳ ಟೆಕ್ಸ್ಟ ಆಗಿದೆ. ಉದಾ : HTML (ಹೈಪರ್ ಟೆಕ್ಸ್ಟ ಮಾರ್ಕಪ್ ಲ್ಯಾಂಗ್ವೇಜ್) ಅಥವಾ ಸೋರ್ಸ್ ಕೋಡ್ (ಮಾನವನು ಬರೆದ ಪ್ರೊಗ್ರಾಂ ನ ಮೂಲ ಆವೃತ್ತಿ)

ಈ ಸಂದರ್ಭದಲ್ಲಿ, ಫೈಲ್ ಕೂಡ ಫೈಲ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಿರುವ ಸಹಜ ಲ್ಯಾಂಗ್ವೇಜ್ (ಉದಾ : ಇಂಗ್ಲಿಷ್, ಟರ್ಕಿಶ್, ಅಥವಾ ಜಪನೀಸ್) ನಿರ್ಧರಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತದೆ.

ಫೈಲ್ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನ ಸರಳೀಕೃತ ಆವೃತ್ತಿ :

file[option(s)] object_name(s)

ಫೈಲ್ ಹಲವಾರು ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ, ಆದರೆ ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಇವುಗಳಲ್ಲಿ ಯಾವುದೂ ಇಲ್ಲದೆ ಬಳಸಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ (ಎಂದರೆ ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) ಇರುವ file1 ಎನ್ನುವ ಫೈಲ್‌ನ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು ಕೇವಲ ಕೆಳಗಿನವುಗಳನ್ನು ಟೈಪ ಮಾಡಿ ಮತ್ತು ರಿಟರ್ನ್ ಕೀ (RETURN)ಯನ್ನು ಒತ್ತಿರಿ.

file file1

ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್ ವಿಧಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು, ಆ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಪ್ರತಿ ಆಬ್ಜೆಕ್ಟ್ ಅನ್ನು ಪ್ರತಿನಿಧಿಸುವ ಸ್ಟಾರ್ (*) ವೈಲ್ಡ್ ಕಾರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.:

file *

ಅಂತೆಯೆ, ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು, ಆ ಡೈರೆಕ್ಟರಿಯನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಉಪಯೋಗಿಸಿ ನಂತರ ಫಾರ್ವರ್ಡ್ ಸ್ಲ್ಯಾಷ್ ಮತ್ತು ಸ್ಟಾರ್ ವೈಲ್ಡ್‌ಕಾರ್ಡ್ . ಉದಾಹರಣೆಗೆ ಈ ಕೆಳಗಿನ ಕಮಾಂಡ /boot ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಎಲ್ಲ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ವರ್ಗೀಕರಿಸುತ್ತದೆ.

file/boot/*

ಸ್ಕ್ವೇರ್ ಬ್ರಾಕೆಟ್‌ಗಳನ್ನು ಸ್ಟಾರ್ ವೈಲ್ಡ್‌ಕಾರ್ಡ್‌ನೊಟ್ಟಿಗೆ ಬಳಸುವುದರಿಂದ, ಯಾವ ಆಬ್ಜೆಕ್ಟ್‌ನ ಹೆಸರುಗಳು ನಿರ್ದಿಷ್ಟ ಅಕ್ಷರದೊಂದಿಗೆ ಅಥವಾ ನಿರ್ದಿಷ್ಟ ಶ್ರೇಣಿಯ ಅಕ್ಷರಗಳೊಂದಿಗೆ ಪ್ರಾರಂಭವಾಗುತ್ತದೆಯೊ ಅಂತಹ ಫೈಲ್ ವಿಧಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಯಾವ ಆಬ್ಜೆಕ್ಟ್‌ನ ಹೆಸರು a ದಿಂದ g ವರೆಗಿನ ಅಕ್ಷರದಿಂದ ಪ್ರಾರಂಭವಾಗುವ ಆಬ್ಜೆಕ್ಟಗಳ ಹೆಸರುಗಳನ್ನು ಮಾತ್ರ ತೋರಿಸುತ್ತದೆ.

file [a-g]*

-k ಆಯ್ಕೆಯು, ಮೊದಲನೆ ಯಶಸ್ವಿ ಪರೀಕ್ಷೆಯಲ್ಲಿ ನಿಲ್ಲಿಸದಂತೆ ಹೇಳುತ್ತದೆ ಮತ್ತು ಮುಂದುವರಿಯುವಂತೆ ಹೇಳುತ್ತದೆ. ಇದು ಕೆಲವು ಫೈಲ್ ಸಿಸ್ಟಮ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ವರದಿ ಮಾಡಬಹುದು. -b (ಎಂದರೆ ಬ್ರೀಫ್) ಆಯ್ಕೆಯು ಔಟಪುಟ್ ಲೈನ್‌ಗಳಿಗೆ ಫೈಲ್ ಹೆಸರುಗಳನ್ನು ಸೇರಿಸದಂತೆ ಫೈಲ್‌ಗೆ ಹೇಳುತ್ತದೆ. ಇದರಿಂದ ಫೈಲ್ ವಿಧಗಳ ಬಗ್ಗೆ ಅಂಕಿಂಶಗಳನ್ನು ಕಂಪೈಲ್ ಮಾಡುವಾಗ ಇದು ಉಪಯುಕ್ತವಾಗುತ್ತದೆ. -v ಆಯ್ಕೆಯು, ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಫೈಲ್‌ನ ಆವೃತ್ತಿಯ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತದೆ.

ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸುವ ಕಮಾಂಡ್ (Creating files and directories command)

mkdir ಕಮಾಂಡ್ (mkdir command)

mkdir ಕಮಾಂಡ್‌ನ್ನು ಹೊಸ ಡೈರೆಕ್ಟರಿ ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಡೈರೆಕ್ಟರಿಯನ್ನು ಕೆಲವು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಫೋಲ್ಡರ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ಇತರ ಡೈರೆಕ್ಟರಿಗಳು ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸುವ ಕಂಟೇನರ್‌ನಂತೆ ಕಾಣುತ್ತದೆ. ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು ಡೈರೆಕ್ಟರಿಗಳನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಕೇವಲ ಫೈಲ್ ಹೆಸರುಗಳ ಪಟ್ಟಿಯನ್ನು ಮತ್ತು ಅವುಗಳಿಗೆ ಅನುಗುಣವಾದ ಐನೋಡ್ ಸಂಖ್ಯೆಗಳನ್ನು ಹೊಂದಿರುವ ವಿಶೇಷ ರೀತಿಯ ಫೈಲ್ ಎಂದು ಪರಿಗಣಿಸುತ್ತವೆ. ಪ್ರತಿ ಐನೋಡ್ ಸಂಖ್ಯೆಯು ಐನೋಡ್ ಟೇಬಲ್ (ಇವುಗಳನ್ನು ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನ ಸುತ್ತಲೂ ಕಾರ್ಯತಂತ್ರದ ಸ್ಥಳಗಳಲ್ಲಿ ಇರಿಸಲಾಗಿರುತ್ತದೆ) ನಲಿ ್ಲರುವ ಐನೋಡ್‌ನ್ನು ಮತ್ತು ಫೈಲ್ ಬಗ್ಗೆ ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ಉದಾ : ಗಾತ್ರ, ಪರ್ಮಿಷನ್‌ಗಳು(ಅನುಮತಿ), ರಚನೆಯ ದಿನಾಂಕ) ಸೂಚಿಸುತ್ತದೆ. ಅದರ ಹೆಸರು ಮತ್ತು ಫೈಲ್ ಹೊಂದಿರುವ ನಿಜವಾದ ಡಾಟಾವನ್ನು ಹೊರತುಪಡಿಸಿ ಎಲ್ಲ ಮಾಹಿತಿ ಸೂಚಿಸುತ್ತದೆ.

mkdir ಕಮಾಂಡ್ ನ ಉದಾಹರಣೆ

$mkdir example

$ls -l

total 4

drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example

directory_name ಇದು ಬಳಕೆದಾರರು mkdir ಗೆ ರಚಿಸಲು ಹೇಳುತ್ತಿರುವ ಯಾವುದೇ ಡೈರೆಕ್ಟರಿಯ ಹೆಸರು ಆಗಿರುತ್ತದೆ. ಏಕಕಾಲದಲ್ಲಿ ಯಾವುದೇ ಸಂಖ್ಯೆಯ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸಬಹುದು.

ಉದಾಹರಣೆಗೆ, ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ (ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) dir_1, dir_2, dir_3 ಹೆಸರಿನ ಮೂರು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸುತ್ತದೆ. :

mkdir dir_1 dir_2 dir_3

ಡೈರೆಕ್ಟರಿಯ ಹೆಸರನ್ನು Mkdir ಗೆ ಆರ್ಗ್ಯುಮೆಂಟ್ (ಎಂದರೆ ಇನ್ಪುಟ್) ಆಗಿ ಒದಗಿಸಿದರೆ, ಅದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಈ ಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಡೈರೆಕ್ಟರಿ ಅಥವಾ ಫೈಲನಲ್ಲಿ ಬಳಕೆದಾರರು Mkdir ಗೆ ಹೊಸ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸಲು ಕೇಳುವಂತೆ ಆಗುತ್ತದೆ. Mkdir ಎಚ್ಚರಿಕೆಯ ಸಂದೇಶವನ್ನು ನೀಡುತ್ತದೆ mkdir: cannot create directory `dir_1`. ಫೈಲ್ ಈಗಾಗಲೇ ಇರುತ್ತದೆ ಮತ್ತು ಅದೇ ಹೆಸರಿನ ಫೈಲನ್ನು ರಚಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಅದಾಗ್ಯೂ ಇತರ ಹೆಸರುಗಳನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಒದಗಿಸಿದಾಗ ಇದು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸುವುದನ್ನು ಮುಂದುವರೆಸುತ್ತದೆ.

ಹೊಸ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುವ ಸಲುವಾಗಿ ಬಳಕೆದಾರರು ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ,(ಯಾವ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಹೊಸ ಡೈರೆಕ್ಟರಿ ರಚಿಸಲಾಗುತ್ತದೆಯೊ) ರೈಟ್ ಪರ್ಮಿಷನ್ (Write Permission) (ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯನ್ನು ಬದಲಾಯಿಸಲು ಸಿಸ್ಟಮ್ ನ ಅನುಮತಿ) ಹೊಂದಿರುವುದು ಅವಶ್ಯಕವಾಗಿರುತ್ತದೆ.

Mkdir ನಿಂದ ರಚಿಸಲ್ಪಟ್ಟ ಡೈರೆಕ್ಟರಿಗಳು ಸ್ವಯಂಚಾಲಿ ತವಾಗಿ ಎರಡು ಗುಪ್ತ(hidden) ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ಒಂದು ಈಗ ರಚಿಸಲಾದ ಡೈರೆಕ್ಟರಿಯನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ(ಒಂದೇ ಚುಕ್ಕೆಯಿಂದ ನಿರೂಪಿಸಲಾಗುತ್ತದೆ) ಮತ್ತು ಇನ್ನೊಂದು ಅದರ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿಯನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ (ಎರಡು ಸತತ ಚುಕ್ಕೆಗಳಿಂದ ನಿರೂಪಿಸಲಾಗುತ್ತದೆ). ಇದನ್ನು ls ಕಮಾಂಡ್‌ನ್ನು -a ಆಯ್ಕೆಯೊಂದಿಗೆ ಉಪಯೋಗಿಸಿ ನೋಡಬಹುದು. ls ಗೆ ಆರ್ಗುಮೇಂಟ್ ಆಗಿ ಒದಗಿಸಲಾದ ಯಾವುದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಅಥವಾ ಆರ್ಗ್ಯುಮೆಂಟ್ ಇಲ್ಲದಿದ್ದರೆ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ತೋರಿಸಲು ಹೇಳುತ್ತದೆ. (ಹಿಡನ್ ಆಗಿರುವದನ್ನು ಒಳಗೊಂಡಂತೆ).ಎಂದರೆ

ls -a

Mkdir ನ -m ಆಯ್ಕೆಯನ್ನು, ಹೊಸ ಡೈರೆಕ್ಟರಿಗಳ ಅನುಮತಿಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಹೊಸ ಡೈರೆಕ್ಟರಿಗಳು, ಡೀಫಾಲ್ಟ್ ಆಗಿ read, write ಮತ್ತು execute(ಎಂದರೆ ಪ್ರೊಗ್ರಾಂ ಆಗಿದ್ದರೆ ಪ್ರೊಗ್ರಾಂನಂತೆ ರನ್ ಆಗುತ್ತದೆ) ಅನುಮತಿಗಳೊಂದಿಗೆ ರಚಿಸಲ್ಪಡುತ್ತವೆ. ಈ ಅನುಮತಿಗಳನ್ನು ಮಾಲೀಕರಿಗೆ(ಡೀಫಾಲ್ಟ್ ಆಗಿ ಡೈರೆಕ್ಟರಿ ರಚಿಸಿದವರು) ಮತ್ತು ಗುಂಪಿಗೆ ಸಕ್ರೀಯಗೊಳಿ ಸಲಾಗಿರುತ್ತದೆ. ಹಾಗೂ read ಮತ್ತು execute ಅನುಮತಿಗಳನ್ನು ಇತರ ಬಳಕೆದಾರರಿಗೆ ಸಕ್ರೀಯಗೊಳಿಸಲಾಗಿರುತ್ತದೆ. ಹೀಗಾಗಿ, ಉದಾಹರಣೆಗೆ dir_4 ಎನ್ನುವ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುವ ಸಲುವಾಗಿ ಎಲ್ಲ ಬಳಕೆದಾರರಿಗೆ ಎಲ್ಲ ಮೂರು ರೀತಿಯ ಅನುಮತಿಗಳನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಲಾಗುತ್ತದೆ , -m ನಂತರ 777 ಅನುಕ್ರಮವನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ:

mkdir -m 777 dir_4

ಮೊದಲನೇ ಅಂಕಿಯು ಮಾಲೀಕರನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ, ಎರಡನೆ ಅಂಕಿಯು ಗುಂಪನ್ನು ಮತ್ತು ಮೂರನೇ ಅಂಕಿಯು ಇತರ ಬಳಕೆದಾರರನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. 7 ಸಂಖ್ಯೆಯು ಎಲ್ಲ ಮೂರು ರೀತಿಯ ಅನುಮತಿಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ(ಎಂದರೆ read, write ಮತ್ತು execute), 6 ಎಂದರೆ read ಮತ್ತು write ಮಾತ್ರ , 5 ಎಂದರೆ read ಮತ್ತು execute, 4 read ಮಾತ್ರ, 3 write ಮತ್ತು execute ಮಾತ್ರ , 2 write ಮಾತ್ರ , 1 execute ಮಾತ್ರ ಮತ್ತು 0 ಎಂದರೆ ಅನುಮತಿ ಇಲ್ಲ.

ಈ ರೀತಿಯಾಗಿ , ಉದಾಹರಣೆಗೆ, dir_5 ಎನ್ನುವ ಹೊಸ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸಲು ಮಾಲೀಕರು read ಮತ್ತು write ಅನುಮತಿಗಳನ್ನು ಹೊಂದಿರುತ್ತಾರೆ, ಗುಂಪು read ಅನುಮತಿಯನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ಇತರ ಬಳಕೆದಾರರು ಯಾವುದೇ ಅನುಮತಿಯನ್ನು ಹೊಂದಿಲ್ಲ, ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಬಳಸಲಾಗುತ್ತದೆ:

mkdir -m 640 dir_5

© NIMI NOT TO BE REPUBLISHED

-p (ಎಂದರೆ ಪೇರೆಂಟ್ಸ್) ಆಯ್ಕೆಯು ಹೊಸ ಡೈರೆಕ್ಟರಿಗಾಗಿ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಮಧ್ಯಂತರ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುತ್ತದೆ. ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ, ಆ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಚಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , ಇದನ್ನು ಈ ಕೆಳಗಿನ ಡೈರೆಕ್ಟರಿ ರಚನೆಯನ್ನು ರಚಿಸಲು ಬಳಸಬಹುದು :

```
mkdir -p food/fruit/citrus/oranges
```

ಈ ಮೇಲಿನ ಡೈರೆಕ್ಟರಿಗಳ ಸರಣಿಯು ರಚಿಸಲ್ಪಟ್ಟಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುವುದು ಸುಲಭವಾಗಿದೆ. ಇದನ್ನು ಮೊದಲನೆ ಡೈರೆಕ್ಟರಿಯ ಹೆಸರನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್‌ನೊಂದಿಗೆ du (ಎಂದರೆ ಡಿಸ್ಕ್ ಯೂಸೇಜ್) ಕಮಾಂಡ್‌ನ್ನು ಬಳಸಿ ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಬಹುದು. ಮೇಲಿನ ಉದಾಹರಣೆಯ ಸಂದರ್ಭದಲ್ಲಿ ಇದನ್ನು ಹೀಗೆ ಬರೆಯಬಹುದು:

du food

ಇತರ ಆಯ್ಕೆಗಳು -v (ಎಂದರೆ ವರ್ಬೋಸ್) , ಪ್ರತಿ ರಚಿಸಲ್ಪಟ್ಟ ಡೈರೆಕ್ಟರಿಗೆ ಒಂದು ಸಂದೇಶವನ್ನು ನೀಡುತ್ತದೆ, --help ಇದು mkdir ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತದೆ, ಮತ್ತು --version ಇದು ಪ್ರಸ್ತುತ ಇನ್ಸ್ಟಾಲ್ ಆದ mkdir ಪ್ರೊಗ್ರಾಂನ ಆವೃತ್ತಿಯ ಸಂಖ್ಯೆಯನ್ನು ನೀಡುತ್ತದೆ.

touch ಕಮಾಂಡ್: touch ಕಮಾಂಡ್, ಪ್ರತಿಯೊಂದು ಫೈಲ್‌ನ ಪ್ರವೇಶ ಮತ್ತು ಮಾರ್ಪಾಡು ಆದ ಸಮಯವನ್ನು, ಸಿಸ್ಟಮ್‌ನ ಪ್ರಸ್ತುತ ಸಮಯಕ್ಕೆ ನವೀಕರಣ ಮಾಡುತ್ತದೆ.

ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿರುವ ಫೈಲನ್ನು ನೀವು ಹೇಳಿದರೆ, touch ಅದೇ ಹೆಸರಿನ ಖಾಲಿ ಫೈಲನ್ನು ರಚಿಸುತ್ತದೆ.

ಫೈಲ್ ಆರ್ಗ್ಯುಮೆಂಟ್ ಡ್ಯಾಶ್ (`-`) ಆಗಿದ್ದರೆ, ಅದನ್ನು ವಿಶೇಷವಾಗಿ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಔಟ್ಪುಟ್‌ನೊಂದಿಗೆ ಸಂಬಂಧಿಸಿದ ಫೈಲ್‌ನ ಸಮಯವನ್ನು ಬದಲಾಯಿಸಲು touch ಕಾರಣವಾಗುತ್ತದೆ.

```
$ touch file1 file2 file3
$ls -1
Total 4
Drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example
-rw-r - - r--1 raghu raghu 0 2012-07-06 14:09 file1
-rw-r - - r-- 1 raghu raghu 0 2012-07-06 14:09 file2
-rw-r - - r-- 1 raghu raghu 0 2012-07-06 14:09 file3
```

Touch ನ ವೈಶಿಷ್ಟ್ಯವೆಂದರೆ, cp (ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಕಾಪಿ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ) ಮತ್ತು mv (ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮೂವ್(ಚಲಿಸಲು) ಅಥವಾ ರೀನೇಮ್ (ಮರುಹೆಸರಿಸುವುದು) ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ) ನಂತಹ

ಕೆಲವು ಕಮಾಂಡ್‌ಗಳ ವಿರುದ್ಧವಾಗಿ ಇದು ಈಗಾಗಲೇ ಇರುವ ಫೈಲನ್ನು ಅದೇ ಹೆಸರಿನಿಂದ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ತಿದ್ದಿ ಬರೆಯುವುದಿಲ್ಲ.(ಎಂದರೆ ವಿಷಯವನ್ನು ಅಳಿಸುವುದು). ಅದರ ಬದಲಾಗಿ, ಇಂತಹ ಫೈಲ್‌ಗಳ ಕೊನೆಯ ಪ್ರವೇಶ ಸಮಯವನ್ನು ಪ್ರಸ್ತುತ ಸಮಯಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ.

touch ನ ಹಲವಾರು ಆಯ್ಕೆಗಳನ್ನು ವಿಶೇಷವಾಗಿ ಬಳಕೆದಾರರಿಗೆ ಫೈಲ್‌ಗಳ ಸಮಯವನ್ನು ಬದಲಾಯಿಸಲು ಅನುವು ಮಾಡಿಕೊಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, -a ಆಯ್ಕೆಯು, ಪ್ರವೇಶ ಸಮಯವನ್ನು ಮಾತ್ರ ಬದಲಾಯಿಸುತ್ತದೆ. -m ಆಯ್ಕೆಯು ಮಾರ್ಪಾಡು ಮಾಡಿದ ಸಮಯವನ್ನು ಮಾತ್ರ ಬದಲಾಯಿಸುತ್ತದೆ. ಈ ಎರಡೂ ಆಯ್ಕೆಗಳನ್ನು ಒಟ್ಟಿಗೆ ಉಪಯೋಗಿಸುವುದರಿಂದ ಪ್ರವೇಶ ಮತ್ತು ಮಾರ್ಪಾಡು ಎರಡರ ಸಮಯವನ್ನು ಪ್ರಸ್ತುತ ಸಮಯಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ:

touch -am file3

-r (ಎಂದರೆ ರೆಫರನ್ಸ್) ಆಯ್ಕೆಯು, -r ನಂತರ ಖಾಲಿ ಸ್ಥಳ, ಮತ್ತು ಫೈಲ್ ಹೆಸರು - ಇದು touch ಗೆ ಪ್ರಸ್ತುತ ಸಮಯದ ಬದಲಾಗಿ ಫೈಲ್ ನ ಟೈಮ್ ಸ್ಟ್ಯಾಂಪ್‌ನ್ನು ಉಪಯೋಗಿಸಲು ಹೇಳುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , ಕೆಳಗೆ ನೀಡಿದ ಕಮಾಂಡ್ ಅದಕ್ಕೆ file4 ರ ಸಮಯನ್ನು file5 ಕ್ಕೆ ಉಪಯೋಗಿಸಲು ಹೇಳುತ್ತದೆ.:

touch -r file4 file5

-B ಆಯ್ಕೆಯು ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಸೆಕೆಂಡುಗಳಷ್ಟು ಹಿಂದಕ್ಕೆ ಹೋಗುವ ಮೂಲಕ ಟೈಮ್ ಸ್ಟ್ಯಾಂಪ್‌ಗಳನ್ನು ಮಾರ್ಪಡಿಸುತ್ತದೆ. ಮತ್ತು -F ಆಯ್ಕೆಯು ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಸೆಕೆಂಡುಗಳಷ್ಟು ಮುಂದಕ್ಕೆ ಹೋಗುವ ಮೂಲಕ ಸಮಯವನ್ನು ಮಾರ್ಪಡಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಕೆಳಗೆ ನೀಡಿದ ಕಮಾಂಡ್ file7 ನ್ನು file6 ಗಿಂತ 30 ಸೆಕೆಂಡು ಹಳೆಯದನ್ನಾಗಿ ಮಾಡುತ್ತದೆ.

touch -r file6 30 file7

-d ಮತ್ತು -t ಆಯ್ಕೆಗಳು ಬಳಕೆದಾರರಿಗೆ ಕೊನೆಯ ನಿರ್ದಿಷ್ಟ ಪ್ರವೇಶ ಸಮಯವನ್ನು ಸೇರಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ. ಹಿಂದಿನದನ್ನು ಸ್ಟ್ರಿಂಗ್ (ಎಂದರೆ, ಅಕ್ಷರಗಳ ಗುಂಪು) ಅನುಸರಿಸುತ್ತದೆ. ದಿನಾಂಕ, ತಿಂಗಳು, ವರ್ಷ, ನಿಮಿಷ : ಸೆಕೆಂಡ್ ಫಾರ್ಮ್ಯಾಟ್ ಬಳಸುತ್ತದೆ ಮತ್ತು ಎರಡನೆಯದು [[CC]YY]MMDDhhmm[.ss] ಫಾರ್ಮ್ಯಾಟ್ ಬಳಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, file8 ರ ಕೊನೆಯ ಪ್ರವೇಶ

© NIMI NOT TO BE REPUBLISHED

ಸಮಯವನ್ನು 10:22a.m. May 1, 2005, ಗೆ ಬದಲಾಯಿಸಲು, 1 May 2005 10:22 ಇದನ್ನು ಸಿಂಗಲ್ ಕೋಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ ಮತ್ತು ಕೆಳಗಿನಂತೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ,

```
touch -d `1 May 2005 10:22` file8
```

ಭಾಗಶಃ ದಿನಾಂಕ-ಸಮಯದ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ, ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ file9 ಗೆ ದಿನಾಂಕವನ್ನು ಮಾತ್ರ ಒದಗಿಸಬೇಕು. (ಈ ಸಂದರ್ಭದಲ್ಲಿ ಸಮಯವು ಸ್ವಯಂಚಾಲಿತವಾಗಿ 0:00 ಗೆ ಸೆಟ್ ಆಗುತ್ತದೆ):

```
touch -d `14 May` file9
```

ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ ಸಮಯವನ್ನು ಮಾತ್ರ ಒದಗಿಸುವುದರಿಂದ ದಿನಾಂಕವನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಪ್ರಸ್ತುತ ದಿನಾಂಕಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ.

```
touch -d `14:24` file9
```

ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಮಾರ್ಗವೆಂದರೆ, ಫೈಲ್‌ನ ಕೊನೆಯ ಮಾರ್ಪಾಡು ಮಾಡಿದ ದಿನಾಂಕವನ್ನು ನೋಡಲು ls ಕಮಾಂಡ್ ಅನ್ನು -l ಆಯ್ಕೆಯ ಜೊತೆಗೆ ಉಪಯೋಗಿಸುವುದು. ಉದಾಹರಣೆಗೆ file10 ಹೆಸರಿನ ಫೈಲ್‌ನ ಸಂದರ್ಭದಲ್ಲಿ ಕಮಾಂಡ್ ಈ ರೀತಿಯಾಗಿರುತ್ತದೆ:

```
ls -l file10
```

ಯಾವುದೇ ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯ ಸಂಪೂರ್ಣ ಟೈಮ್‌ಸ್ಟ್ಯಾಂಪ್ ಅನ್ನು ವೀಕ್ಷಿಸಲು stat ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ , ಕೆಳಗಿನ ಕಮಾಂಡ್, file11 ಎನ್ನುವ ಹೆಸರಿನ ಫೈಲ್‌ನ ಟೈಮ್‌ಸ್ಟ್ಯಾಂಪ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ:

```
stat file11
```

--help ಆಯ್ಕೆಯು ಮೂಲ ಆಯ್ಕೆಗಳ ಪಟ್ಟಿಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಮತ್ತು --version ಆಯ್ಕೆಯು ಪ್ರಸ್ತುತ ಇನ್ಸ್ಟಾಲ್ ಆದ touch ಪ್ರೊಗ್ರಾಂ ನ ಆವೃತ್ತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.

ಕಾಪಿ, ಮೂವ್ ಮತ್ತು ರಿಮೂವ್ ಕಮಾಂಡ್‌ಗಳು(Copy, move and remove Commands)

ಕಾಪಿ ಕಮಾಂಡ್ (Copy Command):ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಕಾಪಿ ಮಾಡಲು Cp ಕಮಾಂಡ್ ಅನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಕಾಪಿ ಮಾಡಲಾದ ಪ್ರತಿಗಳು ಮೂಲಗಳಿಗಿಂತ ಸ್ವತಂತ್ರ್ಯವಾಗಿರುತ್ತವೆ (ಎಂದರೆ ಒಂದರಲ್ಲಿನ ಬದಲಾವಣೆಗಳು ಇನ್ನೊಂದರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ).

cp ನ ಮೂಲ ಸಿಂಟ್ಯಾಕ್ಸ್

```
$cp source destination
```

ಸುರಕ್ಷತಾ ಮುನ್ನೆಚ್ಚರಿಕೆಯಾಗಿ, ಡೀಫಾಲ್ಟ ಆಗಿ cp ಫೈಲ್‌ಗಳನ್ನು ಮಾತ್ರ ಕಾಪಿ ಮಾಡುತ್ತದೆ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅಲ್ಲ. ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಅದೇ ಹೆಸರಿನ ಫೈಲನ್ನು (ಅಥವಾ ಅದೇ ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಯನ್ನು ಕಾಪಿ ಮಾಡಲು ನಿಗದಿಪಡಿಸಿದರೆ) ಕಾಪಿ ಮಾಡಲು ನಿಗದಿಪಡಿಸಿದರೆ ಆ ಫೈಲ್‌ನ್ನು ಓವರ್ ರೈಟ್ (ತಿದ್ದುಪಡಿ) ಮಾಡಲಾಗುತ್ತದೆ. ಅದಾಗ್ಯೂ, ಕಾಪಿ ಮಾಡಿದ ಫೈಲ್‌ಗೆ ಮಾಲೀಕರು, ಗುಂಪು ಮತ್ತು ಅನುಮತಿಗಳು ಅದು ಬದಲಾಯಿಸಿದ ಫೈಲ್‌ನಂತೆಯೆ ಇರುತ್ತವೆ. ಸೋರ್ಸ್ ಫೈಲ್‌ನ ಕೊನೆಯ ಪ್ರವೇಶ ಸಮಯ ಮತ್ತು ಹೊಸ ಫೈಲ್‌ನ ಕೊನೆಯ ಮಾರ್ಪಾಡು ಮಾಡಿದ ಸಮಯ, ಕಾಪಿ ಮಾಡಿದ ಸಮಯಕ್ಕೆ ಹೊಂದಿಸಲಾಗುತ್ತವೆ.

ಒಂದು ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯನ್ನು ಕಾಪಿ ಮಾಡಿದಾಗ, ಅದನ್ನು ಅದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇರಿಸಬೇಕಾದರೆ ಕಾಪಿ ಮಾಡಿದ ಫೈಲ್‌ಗೆ ಮೂಲ ಹೆಸರಿಗಿಂತ ವಿಭಿನ್ನ ಹೆಸರನ್ನು ಇಡಬೇಕು. ಅದಾಗ್ಯೂ ಕಾಪಿ ಮಾಡಿದ ಫೈಲನ್ನು ಬೇರೆ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇರಿಸಬೇಕಾದರೆ ಅದೇ ಮೂಲ ಹೆಸರಿನಿಂದ ಇರಿಸಬಹುದು. ಈ ರೀತಿಯಾಗಿ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ (ಎಂದರೆ ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) ಇರುವ file1 ಎನ್ನುವ ಹೆಸರಿನ ಫೈಲನ್ನು ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಅದೇ ಹೆಸರಿನಿಂದ ಈ ಕೆಳಗಿನಂತೆ ಕಾಪಿ ಮಾಡಬಹುದು, ಉದಾ: /home/john/

```
cp file1 /home/john/file1
```

ಬೇಕಾದಷ್ಟು ಫೈಲ್‌ಗಳನ್ನು ಏಕಕಾಲದಲ್ಲಿ ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಗೆ ಕಾಪಿ ಮಾಡಬಹುದು. ಅದಕ್ಕಾಗಿ ಡೈರೆಕ್ಟರಿ ಹೆಸರಿನ ನಂತರ ಫೈಲಗಳ ಪಟ್ಟಿ ಮಾಡಿ ಕಾಪಿ ಮಾಡಬಹುದು. cp ಒಂದು ಬುದ್ಧಿವಂತ ಕಮಾಂಡ್ ಆಗಿದೆ ಮತ್ತು ಅದಕ್ಕೆ ಕೊನೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್ (ಎಂದರೆ ಇನ್ಪುಟ್ ಡಾಟಾ ದ ತುಣುಕು) ಡೈರೆಕ್ಟರಿ ಹೆಸರು ಆಗಿರುತ್ತದೆ ಎಂದು ತಿಳಿದಿದೆ. ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಕಾಪಿ ಆದ ಎಲ್ಲ ಫೈಲ್‌ಗಳ ಹೆಸರುಗಳು ಮೂಲ ಹೆಸರಿನ ಅದೇ ಹೆಸರುಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಉದಾಹರಣೆಗೆ, ಈ ಕೆಳಗಿನ ಕಾಮಾಂಡ್ file2, file3 ಮತ್ತು file4 ಹೆಸರಿನ ಫೈಲ್‌ಗಳನ್ನು dir1 ಎನ್ನುವ ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಗೆ ಕಾಪಿ ಮಾಡುತ್ತದೆ.

```
cp file2 file3 file4 dir1
```

-r (ಎಂದರೆ ರಿಕರ್ಸಿವ್) ಆಯ್ಕೆಯು, ಇದನ್ನು ಅಪ್ಪರ್ ಕೇಸ್ R ಎಂದೂ ಸಹ ಬರೆಯಲಾಗುತ್ತದೆ. ಇದು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅದರಲ್ಲಿನ ಎಲ್ಲ ವಿಷಯಗಳನ್ನು ಒಳಗೊಂಡಂತೆ, ಕಾಪಿ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. (ಡೈರೆಕ್ಟರಿಗಳು ತಮ್ಮಷ್ಟಕ್ಕೆ ತಾವೇ ಕಾಪಿಯಾಗುವದಿಲ್ಲ. ಅಕಸ್ಮಾತ್ತಾಗಿ ಒಂದೇ ಹೆಸರಿನಿಂದ ಇನ್ನೊಂದು ಬಾರಿ ಕಾಪಿಯಾಗದಂತೆ, ಮಹತ್ವವಾದ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿದ ಒಂದು ಡೈರೆಕ್ಟರಿ, ಅದೇ ಹೆಸರಿನಿಂದ ಬೇರೊಂದು ಡೈರೆಕ್ಟರಿ ಕಾಪಿಯಾಗದಂತೆ ತಡೆಯುತ್ತದೆ.) ಉದಾಹರಣೆಗೆ, ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್ ಈಗಾಗಲೆ

© NIMI NOT TO BE REPUBLISHED

ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ dir2 ಎನ್ನುವ ಡೈರೆಕ್ಟರಿಯನ್ನು, ಅದರಲ್ಲಿರುವ ವಿಷಯಗಳನ್ನು ಒಳಗೊಂಡಂತೆ (ಎಂದರೆ ಫೈಲ್‌ಗಳು, ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು , ಅದರ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು ಇತ್ಯಾದಿ), dir3 ಗೆ ಕಾಪಿ ಮಾಡುತ್ತದೆ.

```
cp -r dir2 dir3
```

-i (ಎಂದರೆ ಇಂಟರ‍್ಯಾಕ್ಟಿವ) ಆಯ್ಕೆಯು, ಕಾಪಿ ಮಾಡಲು ನಿಯೋಜಿಸಲ್ಪಟ್ಟ ಹೆಸರು ಬೇರೆ ಯಾವುದೇ ಫೈಲ್‌ನಿಂದ ಈಗಾಗಲೇ ಬಳಸಲ್ಪಡುತ್ತಿದ್ದರೆ, ಇದು ಬಳಕೆದಾರರನ್ನು ಕೇಳುತ್ತದೆ ಮತ್ತು ಆ ಫೈಲನ್ನು ತಿದ್ದಿ ಬರೆಯಲಾಗುತ್ತದೆ(ಓವರ್ ರೈಟ್). ಪ್ರಾಂಪ್ಟ್ ನಲ್ಲಿ y ಅಕ್ಷರವನ್ನು(ಲೋವರ್ ಕೇಸ್ ಅಥವಾ ಅಪ್ಪರ್ ಕೇಸ್) ನಮೂದಿಸುವುದರಿಂದ ಕಮಾಂಡ್ ಮುಂದುವರಿಯಲು ಕಾರಣವಾಗುತ್ತದೆ. ಯಾವುದೇ ಬೇರೆ ಉತ್ತರವು ಕಮಾಂಡ್ ಫೈಲನ್ನು ತಿದ್ದಿ ಬರೆಯುವುದನ್ನು ತಡೆಯುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, dir4 ಎನ್ನುವ ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಯನ್ನು dir5 ಎಂದು ಕಾಪಿ ಮಾಡಲು ಬಯಸಿದರೆ, ಮತ್ತು

dir4 ಎನ್ನುವ ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಯು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿದ್ದರೆ, ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್ ಅದೇ ಹೆಸರಿನ ಯಾವುದೇ ಫೈಲನ್ನು ನಂತರದ ಡೈರೆಕ್ಟರಿಗೆ ಬದಲಾಯಿಸುವ ಮೊದಲು ಬಳಕೆದಾರರನ್ನು ಕೇಳುತ್ತದೆ:

```
cp -ri dir4 dir5
```

-a ಆಯ್ಕೆಯು ಮೂಲ ಡೈರೆಕ್ಟರಿಯ ರಚನೆ ಮತ್ತು ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಮತ್ತು ಅದರ ವಿಷಯಗಳನ್ನು ಹೊಸ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಸಂರಕ್ಷಿಸುತ್ತದೆ. ಇದು ಅರ್ಕೈವ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯುಕ್ತವಾಗಿದೆ. ಇದು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಪುನರಾವರ್ತಿತವಾಗಿ ಕಾಪಿ ಮಾಡುವಲ್ಲಿ, -r ಆಯ್ಕೆಯನ್ನು ಹೋಲುತ್ತದೆ. ಇದು ಸಾಂಕೇತಿಕ ಲಿಂಕ್‌ಗಳನ್ನು ಎಂದೂ ಅನುಸರಿಸುವುದಿಲ್ಲ. ಇದು -rdp ಆಯ್ಕೆಗಳ ಸಂಯೋಜನೆಗೆ ಸಮನಾಗಿದೆ.

ಒಂದು ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಗೆ, ಸ್ಟಾರ್ ವೈಲ್ಡ್‌ಕಾರ್ಡ್ ಬಳಸುವ ಮೂಲಕ ಕಾಪಿ ಮಾಡಬಹುದು. ಸ್ಟಾರ್ ಅಕ್ಷರವು ಯಾವುದೇ ಒಂದು ಅಕ್ಷರ ಅಥವಾ ಯಾವುದೇ ಅಕ್ಷರಗಳ ಸಮೂಹವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್ dir6 ಎನ್ನುವ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು , dir7 ಎನ್ನುವ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲಿರುವ ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಗೆ ಕಾಪಿ ಮಾಡುತ್ತದೆ.

```
cp dir6/* dir7
```

ಆಯ್ದ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಡೈರೆಕ್ತರಿಗಳನ್ನು ಕಾಪಿ ಮಾಡಲು, cp ಯನ್ನು ಸ್ಟಾರ್ ವೈಲ್ಡ್ ಕಾರ್ಡ್‌ನೊಂದಿಗೆ ಅಥವಾ ಇತರ ಮಾದರಿಯ ಹೊಂದಾಣಿಕೆಯ ಅಕ್ಷರಗಳೊಂದಿಗೆ ಸಹ ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ .html ಫೈಲ್ ವಿಸ್ತರಣೆಯ ಹೆಸರನ್ನು ಹೊಂದಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು, dir8 ಎನ್ನುವ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಇನ್ನೊಂದು ಡೈರೆಕ್ಟರಿಗೆ ಕಾಪಿ ಮಾಡಲು ಕೆಳಗಿನ ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ:

```
cp *.html dir8
```

ಈ ಸಂದರ್ಭದಲ್ಲಿ, ಸ್ಟಾರ್ ವೈಲ್ಡ್‌ಕಾರ್ಡ್, .html ಫೈಲ್ ವಿಸ್ತರಣೆಯ ಹೆಸರಿನಿಂದ ಕೊನೆಗೊಳ್ಳುವ ಎಲ್ಲವನ್ನೂ ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

cp ಯ ಇತರ ಆಯ್ಕೆಗಳಲ್ಲಿ -b , ಪ್ರತಿ ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲ್‌ನ ಬ್ಯಾಕಪ್ ಕಾಪಿ ಮಾಡುತ್ತದೆ. -f (ಎಂದರೆ ಫೋರ್ಸ್) ಇದು ತೆರೆಯಲಾಗದ ಫೈಲ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ನಂತರ ಮತ್ತೆ ಪ್ರಯತ್ನಿಸುತ್ತದೆ. -s ಇದು ಕಾಪಿ ಮಾಡುವ ಬದಲಾಗಿ ಸಾಂಕೇತಿಕ ಲಿಂಕ್ ಗಳನ್ನು ತಯಾರಿಸುತ್ತದೆ. -u (ಎಂದರೆ ಅಪ್ಡೇಟ್) ಇದು ಸೋರ್ಸ್ ಫೈಲ್, ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲಗಿಂತ ಹೊಸದಾಗಿದ್ದರೆ ಅಥವಾ ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲ್ ಕಾಣೆಯಾಗಿದ್ದರೆ ಮಾತ್ರ ಕಾಪಿ ಮಾಡುತ್ತದೆ. -v (ಎಂದರೆ ವರ್ಬೋಸ್) ಇದು ಏನು ನಡೆಯುತ್ತಿದೆ ಎನ್ನುವುದರ ಬಗ್ಗೆ ಸಂಕ್ಷಿಪ್ತ ಕಮೆಂಟ್ ಮಾಡುತ್ತದೆ, ಮತ್ತು -x ಇದು cp ಗೆ ಅದೇ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಇರುವಂತೆ ಹೇಳುತ್ತದೆ.

ಮೂವ್ ಕಮಾಂಡ್ (Move Command):ಫೈಲ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರೀನೇಮ್(ಮರುಹೆಸರಿಸುವುದು) ಮಾಡಲು ಮತ್ತು ಮೂವ್(ಚಲಿಸು) ಮಾಡಲು Mv ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದರ ಸಾಮಾನ್ಯ ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
$mv source destination
```

ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳ ಹೆಸರುಗಳು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳಾಗಿರುತ್ತವೆ. ಎರಡು ಫೈಲ್ ಹೆಸರುಗಳನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ನೀಡಿದರೆ, mv ಮೊದಲನೆಯ ಹೆಸರನ್ನು ಎರಡನೆಯದರಿಂದ ಮರುಹೆಸರಿಸುತ್ತದೆ.

ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ನೀಡಿದರೆ, ಮತ್ತು ಅನುಕ್ರಮದಲಿ್ಲನ ಕೊನೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್ ಹೆಸರು ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಡೈರೆಕ್ಟರಿಯ ಹೆಸರಾಗಿದ್ದರೆ, mv ಇತರ ಎಲ್ಲ ಅಂಶಗಳನ್ನು ಆ ಡೈರೆಕ್ಟರಿಗೆ ಮೂವ್(ಸರಿಸು) ಮಾಡುತ್ತದೆ. ಕೊನೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿರುವ ಡೈರೆಕ್ಟರಿ ಆಗಿದ್ದರೆ, ಮತ್ತು ಎರಡಕ್ಕಿಂತ ಹೆಚ್ಚು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಒದಗಿಸಿದರೆ, ದೋಷ ಸಂದೇಶವನ್ನು(error message) ತೋರಿಸುತ್ತದೆ.

ಸೋರ್ಸ್ ಫೈಲ್ ಇರುವ ಅದೇ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಡೆಸ್ಟಿನೇಷನ್ ಫೈಲ್ ಇದ್ದರೆ, ಸೋರ್ಸ್ ಫೈಲ್‌ನ್ನು ರೀನೇಮ್ ಮಾಡಬಹುದು. ಎರಡೂ ಫೈಲ್‌ಗಳು ವಿಭಿನ್ನ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿದ್ದರೆ, ಡೆಸ್ಟಿನೇಷನ್ ಆರ್ಗುಮೆಂಟ್‌ನಲ್ಲಿ ಹೆಸರಿಸಲಾದ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಸೋರ್ಸ್ ಫೈಲನ್ನು ಮೂವ್ ಮಾಡಲಾಗುತ್ತದೆ , ಇದರಲ್ಲಿ ಅದು ತನ್ನ ಮೂಲ ಹೆಸರನ್ನು ಉಳಿಸಿಕೊಳ್ಳುತ್ತದೆ ಅಥವಾ ಅದಕ್ಕೆ ಹೊಸ ಹೆಸರನ್ನು ಇಡಬಹುದು. ಮೂವ್ ಮಾಡಲು ಡೈರೆಕ್ಟರಿ ಗುರಿಯಾಗಿದ್ದರೆ, ಸೋರ್ಸ್ ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯನ್ನು ಆ ಡೈರೆಕ್ಟರಿಗೆ ಮೂವ್ ಮಾಡಲಾಗುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

ಮತ್ತು ಅದು ತನ್ನ ಮೂಲ ಹೆಸರನ್ನು ಉಳಿಸಿಕೊಳ್ಳುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್,file1 ನ್ನು file2 ಎಂದು ರೀನೇಮ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅದನ್ನು , ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿಯೆ (ಎಂದರೆ ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) ಇರಿಸುತ್ತದೆ.

mv file1 file2

ಕೆಳಗಿನ ಕಮಾಂಡ್, file3 ಎನ್ನುವ ಹೆಸರಿನ ಫೈಲನ್ನು ಅದರ ಹೆಸರು ಬದಲಿಸದೇ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಿಂದ, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲಿ ್ಲರುವ ಈಗಾಗಲೆ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ dir1 ಎನ್ನುವ ಸಬ್ ಡೈರೆಕ್ಟರಿಗೆ ಮೂವ್ ಮಾಡುತ್ತದೆ. :

mv file3 dir1/ file3

mv ಕಮಾಂಡ್‌ನ್ನು ಏಕಕಾಲದಲ್ಲಿ ಅನೇಕ ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮೂವ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅವುಗಳಲ್ಲಿರುವ ಎಲ್ಲ ವಿಷಯವನ್ನು ಒಳಗೊಂಡಂತೆ, /home/alice/new ಎನ್ನುವ ಡೈರೆಕ್ಟರಿಗೆ ಮೂವ್ ಮಾಡುತ್ತದೆ.:

mv * /home/alice/new/

ಆಸ್ಟೇರಿಸ್ಕ್ ಒಂದು ವೈಲ್ಡ್ ಕಾರ್ಡ್ ಅಕ್ಷರವಾಗಿದ್ದು, ಯಾವುದೇ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು (ಅಕ್ಷರಗಳ ಸಮೂಹ) ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಹೀಗಾಗಿ ಮೇಲಿ ನ ಉದಾಹರಣೆಯಲ್ಲಿ ಇದು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

mv ಕಮಾಂಡ್ ಡೈರೆಕ್ಟರಿಗಳ ಕ್ರಮಾನುಗತದಲ್ಲಿ ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯನ್ನು ಮೇಲಕ್ಕೆ ಸರಿಸಲು ಸುಲಭಗೊಳಿಸುತ್ತದೆ (ಎಂದರೆ, ರೂಟ್ ಡೈರೆಕ್ಟರಿಗೆ ಹತ್ತಿರ). ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್ ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ dir/dir ಎನ್ನುವ ಸಬ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ file4 ಎನ್ನುವ ಹೆಸರಿನ ಫೈಲನ್ನು , ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಯ ಉನ್ನತ ಮಟ್ಟಕ್ಕೆ ಮೂವ್ ಮಾಡಲು :

mv dir/dir/file4 ~

ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ, ರೂಟ್ ಡೈರೆಕ್ಟರಿ ಇತರ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಹೊಂದಿರುವ ಡೈರೆಕ್ಟರಿ ಆಗಿರುತ್ತದೆ ಮತ್ತು ಇದು ಕ್ರಮಾನುಗತದಲ್ಲಿ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿಗಳ ಮೇಲ್ಭಾಗದಲ್ಲಿ ಇರುತ್ತದೆ. ಬಳಕೆದಾರರ ಹೋಮ್ ಡೈರೆಕ್ಟರಿ ಎಂದರೆ, ಸಿಸ್ಟಮ್ ನಲ್ಲಿ ಲಾಗಿನ್ ಆದ ತಕ್ಷಣ ಬಳಕೆದಾರರು ತಮ್ಮನ್ನು ತಾವು ಕಂಡುಕೊಳ್ಳುವ ಒಂದು ಡೈರೆಕ್ಟರಿ ಆಗಿರುತ್ತದೆ, ಮತ್ತು ಇದನ್ನು ಟಿಲ್ಡ್ (~) ನಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ.

mv ತನ್ನ ಕೆಲಸವು ಯಾವುದೇ ಸಮಸ್ಯೆ ಇಲ್ಲದೆ ಪೂರ್ಣಗೊಂಡಿರುವ ಬಗ್ಗೆ ಯಾವುದೇ ದೃಢೀಕರಣವನ್ನು ಪರದೆಯ ಮೇಲೆ ತೋರಿಸುವುದಿಲ್ಲ. ಇದು ಯುನಿಕ್ಸ್‌ನ ತತ್ವಶಾಸ್ತ್ರ ದ ಮೌನ ಸಿದ್ಧಾಂತದ ನಿಯಮಕ್ಕೆ ಅನುಗುಣವಾಗಿದೆ.

ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಹೊಸ ಬಳಕೆದಾರರಿಗೆ -i ಆಯ್ಕೆಯನ್ನು ಬಳಸುವುದು ಉತ್ತಮವಾಗಿರುತ್ತದೆ. ಇ ದು ಡೆಸ್ಟಿನೇಷನ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಒಂದೇ ಹೆಸರನ್ನು ಹೊಂದಿದ ಫೈಲ್ ಮತ್ತು/ಅಥವಾ ಡೈರೆಕ್ಟರಿಗಳು ಇದ್ದ ಸಂದರ್ಭದಲ್ಲಿ Mv ಸಂವಾದಾತ್ಮಕವಾಗಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಮೇಲಿನ ಕಮಾಂಡ್‌ನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಸಂವಾದಾತ್ಮಕವಾಗಿ ಮಾಡಬಹುದು:

mv -i * /home/alice/new/

mv ಯ ಕೆಲವು ಇತರ ಆಯ್ಕೆಗಳಲ್ಲಿ -b , ಇದು ಪ್ರತಿ ಫೈಲ್‌ನ ಬ್ಯಾಕಪ್ ಪ್ರತಿಯನ್ನು ಮಾಡಲು ಹೇಳುತ್ತದೆ, ಇಲ್ಲದಿದ್ದರೆ ಅವು ತಿದ್ದುಪಡಿಯಾಗಬಹುದು ಅಥವಾ ತೆಗೆದುಹಾಕಲ್ಪಡಬಹುದು, -v ಇ ದು ವರ್ಬೋಸ್ ಆಗಿರಲು ಹೇಳುತ್ತದೆ ಮತ್ತು ಪ್ರತಿಯೊಂದು ಫೈಲನ್ನು ಮೂವ್ ಮಾಡುವ ಮೊದಲು ಅದರ ಹೆಸರನ್ನು ಪ್ರದರ್ಶಿಸಲು ಹೇಳುತ್ತದೆ. --help ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿ mv ಬಗ್ಗೆ ವಿವರವಾದ ಮಾಹಿತಿಯನ್ನು (ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಒಳಗೊಂಡಂತೆ) ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಮತ್ತು --version ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಇದರ ಪ್ರಸ್ತುತ ಆವೃತ್ತಿಯ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ರಿಮೂವ್ ಅಥವಾ ಡಿಲೀಟ್ (Remove or Delete)

$rmdir

rmdir ಕಮಾಂಡ್, ಯಾವುದೇ ಖಾಲಿ ಡೈರೆಕ್ಟರಿಯನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ, ಆದರೆ ಅದರಲ್ಲಿ ಫೈಲ್ ಇದ್ದರೆ ಆ ಡೈರೆಕ್ಟರಿಯನ್ನು ತೆಗೆದುಹಾಕಲು ಸಾಧ್ಯವಿಲ್ಲ. rmdir ಕಮಾಂಡ್‌ನ್ನು ಬಳಸಲು, ಮೊದಲು ನೀವು ತೆಗೆದುಹಾಕಲು ಬಯಸಿರುವ ಆ ಡೈರೆಕ್ಟರಿಯಲಿ ್ಲರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಿ(ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳು ಇದ್ದರೆ ಅವುಗಳನ್ನು ಸಹ)

ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದುಹಾಕಿ (Remove files and directories): ಲಿನಕ್ಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದುಹಾಕಲು(delete) rm (ಎಂದರೆ ರಿಮೂವ್) ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

rm ನ ಸಾಮಾನ್ಯ ಸಿಂಟ್ಯಾಕ್ಸ್ rm [options][-r directories] filenames

ಸ್ಕ್ವೇರ್ ಬ್ರಾಕೆಟ್‌ನಲ್ಲಿರುವ ಅಂಶಗಳು ಐಚ್ಛಿಕವಾಗಿವೆ. ಇದನ್ನು ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫೈಲ್ ಹೆಸರುಗಳೊಂದಿಗೆ ಮಾತ್ರ ಬಳಸಿದಾಗ, ಬಳಕೆದಾರರಿಂದ ದೃಢೀಕರಣದ ಅಗತ್ಯವಿಲ್ಲದೆ ಆ ಎಲ್ಲ ಫೈಲ್‌ಗಳನ್ನು ಅಳಿಸುತ್ತದೆ. ಈ ರೀತಿ ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ , rm, file1, file2 ಮತ್ತು fie3 ಹೆಸರಿನ ಫೈಲ್‌ಗಳನ್ನು ಇವು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ (ಎಂದರೆ ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತವಾಗಿ ಕಾರ್ಯ

ನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) ಇವೆ ಎಂದು ಗ್ರಹಿಸಿ ತಕ್ಷಣವೇ ಅಳಿಸುತ್ತದೆ.

```
rm file1 file2 file3
```

ಯಾವುದೇ ಫೈಲ್ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ ಅಥವಾ ಆ ಫೈಲನ್ನು ಅಳಿಸಲು ಬಳಕೆದಾರರು ಸೂಕ್ತ ಅನುಮತಿ ಹೊಂದಿಲ್ಲದಿದ್ದರೆ, ಅದು ದೋಷ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ. ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ (write- protected) ಫೈಲ್‌ಗಳು ಅವುಗಳನ್ನು ಅಳಿಸುವ ಮೊದಲು ದೃಢೀಕರಣವನ್ನು (y ಎಂದರೆ yes ಮತ್ತು n ಎಂದರೆ no) ಕೇಳುತ್ತವೆ. ಫೈಲ್‌ಗಳು ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ ಆಗಿಲ್ಲದಿದ್ದರೂ ಸಹ, ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ ಡೈರೆಕ್ಟರಿಯಲಿ್ಲರುವ ಫೈಲ್‌ಗಳನ್ನು ಎಂದಿಗೂ ಅಳಿಸಲು ಬರುವುದಿಲ್ಲ.

-f (ಎಂದರೆ ಫೋರ್ಸ್) ಆಯ್ಕೆಯು ಎಲ್ಲ ನಿರ್ದಿಷ್ಟ ಫೈಲ್‌ಗಳನ್ನು ಅಳಿಸಲು rm ಗೆ ಹೇಳುತ್ತದೆ. ಆ ಫೈಲ್‌ಗಳು ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ ಆಗಿದ್ದರು ಅಥವಾ ಆಗಿಲ್ಲದಿದ್ದರೂ ಸಹ ಬಳಕೆದಾರರಿಗೆ ದೃಢೀಕರಣವನ್ನು ಕೇಳದೆ ಅಳಿಸುತ್ತದೆ. ಇದು ದೋಷ ಸಂದೇಶವನ್ನು (error message) ತೋರಿಸುವುದಿಲ್ಲ ಅಥವಾ ಆ ನಿರ್ದಿಷ್ಟ ಫೈಲ್ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೂ ದೋಷ ಸ್ಥಿತಿಯನ್ನು (error status) ತೋರಿಸುವುದಿಲ್ಲ. ಅದಾಗ್ಯೂ, ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಅಳಿಸಲು ಪ್ರಯತ್ನಿಸಿದರೆ, ಈ ಆಯ್ಕೆಯು ದೋಷ ಸಂದೇಶವನ್ನು ನಿಗ್ರಹಿಸುವುದಿಲ್ಲ.

-i (ಎಂದರೆ ಇಂಟರ್‌ಆಕ್ಟಿವ್) ಆಯ್ಕೆಯು, ಪ್ರತಿಯೊಂದು ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಯನ್ನು ಅಳಿಸುವ ಮೊದಲು ಬಳಕೆದಾರರ ದೃಢೀಕರಣವನ್ನು ಕೇಳುವಂತೆ rm ಗೆ ತಿಳಿಸುತ್ತದೆ. -f ಮತ್ತು -I ಎರಡೂ ಆಯ್ಕೆಗಳನ್ನು ಉಪಯೋಗಿಸಿದರೆ, ಕೊನೆಗೆ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಆಯ್ಕೆಯು ಪರಿಣಾಮ ಬೀರುತ್ತದೆ.

ಸುರಕ್ಷತಾ ಕ್ರಮವಾಗಿ, rm ಡೀಫಾಲ್ಟ್ ಆಗಿ ನೇರವಾಗಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅಳಿಸುವುದಿಲ್ಲ. ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅಳಿಸುವ ಸಲುವಾಗಿ -r ಆಯ್ಕೆಯನ್ನು ಉಪಯೊಗಿಸುವುದು ಅವಶ್ಯಕವಾಗಿರುತ್ತದೆ. ಇದು -R ಆಯ್ಕೆಗೆ ಸಮನಾಗಿರುತ್ತದೆ. ಈ ಆಯ್ಕೆಯು ಆರ್ಗ್ಯುಮೆಂಟ್ ಲಿಸ್ಟ್ ನಲ್ಲಿರುವ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮತ್ತು ಅದರಲ್ಲಿರುವ ವಿಷಯಗಳನ್ನು ಪುನರಾವರ್ತಿತವಾಗಿ ಅಳಿಸುತ್ತದೆ. ಎಂದರೆ, ಯಾವುದೇ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟ ಡೈರೆಕ್ಟರಿಗಳನ್ನು(ಅದರ ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳು ಮತ್ತು ಫೈಲ್‌ಗಳು ಇತ್ಯಾದಿ.) ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ಮೊದಲು ಖಾಲಿ ಮಾಡುತ್ತದೆ ನಂತರ ಅವುಗಳನ್ನು ಅಳಿಸುತ್ತದೆ(removed). -f ಆಯ್ಕೆಯ ಬಳಕೆಯನ್ನು ಹೊರತುಪಡಿಸಿ, ಯಾವುದೇ ರೈಟ್ ಪ್ರೊಟೆಕ್ಟೆಡ್ ಫೈಲ್ ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಅಳಿಸಬೇಕಾದರೆ ಮೊದಲು ಬಳಕೆದಾರರನ್ನು ಕೇಳಲಾಗುತ್ತದೆ.

rm ಕಮಾಂಡ್‌ನಲ್ಲಿರುವ ಫೈಲ್ ಒಂದು ವೇಳೆ ಸಾಂಕೇತಿಕ ಲಿಂಕ್ ಆಗಿದ್ದರೆ, ಆ ಲಿಂಕ್‌ನ್ನು ತೆಗೆದುಹಾಕಲಾಗುತ್ತದೆ ಆದರೆ, ಆ ಲಿಂಕ್ ಸೂಚಿಸುವ ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿ ಮೇಲೆ ಇದು ಯಾವ ಪರಿಣಾಮವನ್ನೂ ಬೀರುವುದಿಲ್ಲ. ಒಂದು ಸಾಂಕೇತಿಕ ಲಿಂಕ್ ಇರುವ ಡೈರೆಕ್ಟರಿಗೆ ರೈಟ್ ಅನುಮತಿಯನ್ನು ಹೊಂದಿರುವ ಬಳಕೆದಾರರಿಗೆ ಆ ಲಿಂಕ್‌ನ್ನು ಅಳಿಸಲು ಯಾವ ರೈಟ್ ಅನುಮತಿಯ(write permission) ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

rm ಕಮಾಂಡ್, -- ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಡಿಲಿಮಿಟರ್ ಆಗಿ ಬೆಂಬಲಿಸುತ್ತದೆ. ಇದು ಆಯ್ಕೆಯ ಕೊನೆಯನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಒಂದು ಫೈಲ್ ಅಥವಾ ಡೈರೆಕ್ಟರಿಯ ಹೆಸರು ಡ್ಯಾಷ್ ಅಥವಾ ಹೈಫನ್‌ನಿಂದ ಪ್ರಾರಂಭವಾದಾಗ ಇದು ಉಪಯುಕ್ತವಾಗಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್ -dir1 ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಯನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

```
rm -r -- -dir1
```

ಇತರ ಆಯ್ಕೆಗಳಲ್ಲಿ -v (ಎಂದರೆ ವರ್ಬೋಸ್) ಏನು ನಡೆಯುತ್ತಿದೆ ಎನ್ನುವುದರ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತದೆ, --help ಇದು rm ಬಗ್ಗೆ ಮೂಲ ಡಾಕ್ಯುಮೆಂಟೇಷನ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು --version ಇದು ಪ್ರಸ್ತುತ ಬಳಕೆಯಾಗುತ್ತಿರುವ rm ನ ಆವೃತ್ತಿಯ ಬಗ್ಗೆ ಹೇಳುತ್ತದೆ. rm ನ ಹಳೆಯ ಆವೃತ್ತಿಗಳ ಬಗ್ಗೆ ಕೆಲವು ವ್ಯತ್ಯಾಸಗಳಿರುತ್ತವೆ. ಅದ್ದರಿಂದ, ಒಂದು ನಿರ್ದಿಷ್ಟ ಸಿಸ್ಟಮ್‌ನ ಡಾಕ್ಯುಮೆಂಟೇಷನ್‌ನ್ನು ಓದುವುದು ಯಾವಾಗಲೂ ಒಳ್ಳೆಯದು.

rmdir ಇದು rm ಗಿಂತ ಭಿನ್ನವಾಗಿದೆ. rmdir ನ್ನು ಖಾಲಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮಾತ್ರ ತೆಗೆದುಹಾಕಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

rmdir ಕಮಾಂಡ್ (rmdir command)

rmdir ಕಮಾಂಡ್‌ನ್ನು ಲಿನಕ್ಸ್‌ನಲ್ಲಿ ಖಾಲಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದುಹಾಕಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

rmdir ನ ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
rmdir [option] directory_names
```

rm ನ್ನು ಯಾವುದೇ ಆಯ್ಕೆಗಳಿಲ್ಲದೇ ಉಪಯೋಗಿಸಿದಾಗ, ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಒದಗಿಸಿದ ಯಾವುದೇ ಖಾಲಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಇದು ಅಳಿಸುತ್ತದೆ.(delete) ಈ ಡೈರೆಕ್ಟರಿಗಳು ರೈಟ್ ಅನುಮತಿಯನ್ನು ಹೊಂದಿರಲಿ ಅಥವಾ ಹೊಂದದೆ ಇರಲಿ ಅವುಗಳನ್ನು ತೆಗೆದು ಹಾಕುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ (ಬಳಕೆದಾರರು ಪ್ರಸ್ತುತವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿರುವ ಡೈರೆಕ್ಟರಿ) dir1 ಮತ್ತು dir2 ಎನ್ನುವ ಹೆಸರಿನ ಎರಡು ಖಾಲಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದು ಹಾಕುತ್ತದೆ.:

```
rmdir dir1 dir2
```

ಖಾಲಿ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮಾತ್ರ ತೆಗೆದುಹಾಕುವ ಸಾಮರ್ಥ್ಯವು ಅಂತರ್ನಿರ್ಮಿತ ಸುರಕ್ಷತಾ ಕ್ರಮವಾಗಿದೆ. ಇದು ಡಾಟಾದ ಆಕಸ್ಮಿಕ ನಷ್ಟವನ್ನು ತಡೆಯಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು ಬಹು ಮುಖ್ಯವಾದ ಅಂಶವಾಗಿದೆ, ಏಕೆಂದರೆ ಒಮ್ಮೆ ಡಾಟಾ ಅಳಿಸಿದರೆ ಯುನಿಕ್ಸ್ ತರಹದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಅಳಿಸಿದ ಡಾಟಾವನ್ನು ಮರಳಿ ಪಡೆಯುವುದು ಅತ್ಯಂತ ಕಷ್ಟವಾಗಿದೆ ಅಥವಾ ಅಸಾಧ್ಯವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

-p (ಎಂದರೆ ಪೇರೆಂಟ್ಸ್) ಆಯ್ಕೆಯು ನಿರ್ದಿಷ್ಟ ಡೈರೆಕ್ಟರಿಯ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದು ಹಾಕಲು rmdir ಗೆ ಹೇಳುತ್ತದೆ. ಇಲ್ಲಿ ಅನುಕ್ರಮದ ಪ್ರತಿ ಡೈರೆಕ್ಟರಿ ಅದು ಖಾಲಿ ಇದ್ದರೆ ಮತ್ತು ಪ್ರತಿ ಪೇರೆಂಟ್ ಡೈರೆಕ್ಟರಿ ರೈಟ್ ಅನುಮತಿ ಹೊಂದಿದ್ದರೆ ಅವುಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಕಮಾಂಡ್, dir5, dir4 ಮತ್ತು dir3 ಹೆಸರಿನ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ, ಆದರೆ dir5 ಖಾಲಿ ಇದ್ದರೆ ಮಾತ್ರ , dir4 ಡೈರೆಕ್ಟರಿಯು dir5ನ್ನು ಮಾತ್ರ ಒಳಗೊಂಡಿದೆ, ಮತ್ತು dir3 ಡೈರೆಕ್ಟರಿಯು dir4 ನ್ನು ಮಾತ್ರ ಒಳಗೊಂಡಿದೆ,(ಇದು ಮತ್ತೆ dir5ನ್ನು ಒಳಗೊಂಡಿದೆ)

rmdir -p dir3/dir4/dir5

ಇದು -p ಆಯ್ಕೆಯ ಜೊತೆಗೆ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸುವ mkdir ಕಮಾಂಡ್ ನ ಸಮ್ಮಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಹೀಗಾಗಿ ಮೇಲಿನ ನೆಸ್ಟೆಡ್ ಡೈರೆಕ್ಟರಿಗಳ ಸಮುಹವನ್ನು ಈ ರೀತಿಯಾಗಿ ಸುಲಭವಾಗಿ ರಚಿಸಬಹುದು:

mkdir -p dir3/dir4/dir5

ಫೈಲ್‌ಗಳು ಮತ್ತು ಡೈರೆಕ್ಟರಿಗಳು ಎರಡನ್ನೂ ಅಳಿಸುವ rm ಕಮಾಂಡ್‌ಗೆ ವಿರುದ್ಧವಾಗಿ, rmdir ಗೆ -r ಆಯ್ಕೆಯು ಇರುವುದಿಲ್ಲ. ಕನಿಷ್ಟ ಲಿನಕ್ಸ್‌ನ ಪ್ರಮಾಣಿತ ಆವೃತ್ತಿಯಾದ GNU ಆವೃತ್ತಿಯಲಿ ್ಲಯೂ ಇರುವುದಿಲ್ಲ. ಈ ಆಯ್ಕೆಯು ಮೊದಲು ಅದರಲ್ಲಿರುವ ಎಲ್ಲ ವಿಷಯವನ್ನು ಅಳಿಸುವ ಮೂಲಕ ಒಂದು ಡೈರೆಕ್ಟರಿಯನ್ನು ಪುನರಾವರ್ತಿತವಾಗಿ ಅಳಿಸುವಂತೆ rm ನ್ನು ಅನುಮತಿಸುತ್ತದೆ. ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳಲ್ಲಿನ ಕೆಳ ಹಂತದಲ್ಲಿರುವ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮೊದಲು ಅಳಿಸಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಆದ್ದರಿಂದ ಬಳಕೆದಾರರು ಸಂಪೂರ್ಣ ಡೈರೆಕ್ಟರಿ ರಚನೆಯನ್ನು ತೆಗೆದು ಹಾಕಲು ಬಯಸಿದರೆ, ಪ್ರತಿ ಡೈರೆಕ್ಟರಿಯ ವಿಷಯಗಳನ್ನು ಮತ್ತು ಸಬ್ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಮೊದಲು ತೆಗೆದು ಹಾಕಲು ಪ್ರಯತ್ನಿಸುವ ಬದಲಿಗೆ , ಸಾಮಾನ್ಯವಾಗಿ rm ನ್ನು -r ಆಯ್ಕೆಯೊಂದಿಗೆ ಉಪಯೋಗಿಸುವುದು ಸಮರ್ಥವಾಗಿದೆ.

rmdir ಕಮಾಂಡ್ rm ನೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳುವ ಮೂರು ಆಯ್ಕೆಗಳು :-v (ಎಂದರೆ ವರ್ಬೋಸ್) ಇದು ಏನು ನಡೆಯುತ್ತಿದೆ ಎಂಬುದರ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. --help ಇದು rmdir ಬಗ್ಗೆ ಡಾಕ್ಯುಮೆಂಟೇಷನ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮತ್ತು --version ಇದು ಪ್ರಸ್ತುತ ಬಳಸುತ್ತಿರುವ rmdir ನ ಆವೃತ್ತಿಯನ್ನು ಹೇಳುತ್ತದೆ. rmdir ನ ವಿಭಿನ್ನ ಆವೃತ್ತಿಗಳ ಬಗ್ಗೆ ಕೆಲವು ವ್ಯತ್ಯಾಸಗಳಿರುತ್ತವೆ. ಆದ್ದರಿಂದ, ಒಂದು ನಿರ್ದಿಷ್ಟ ಸಿಸ್ಟಮ್‌ನ ಡಾಕ್ಯುಮೆಂಟೇಷನ್‌ನ್ನು ಓದುವುದು ಯಾವಾಗಲೂ ಒಳ್ಳೆಯದು.

cat ಕಮಾಂಡ್‌ನ್ನು ಬಳಸಿ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುವುದು ಮತ್ತು ಸಂಯೋಜಿಸುವುದು(Listing and combining files with the cat command)

ಒಂದು ಫೈಲ್‌ನ ವಿಷಯವನ್ನು ನಿಮ್ಮ ಸ್ಕ್ರೀನ್‌ಗೆ ಕಳುಹಿಸಲು cat (ಎಂದರೆ concatenate file) ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಫೈಲ್‌ನ ವಿಷಯವನ್ನು ಇನ್ನೊಂದು ಫೈಲ್ ಗೆ ಕಳುಹಿಸಲೂ ಕೂಡ ಈ ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಇನ್ಪುಟ್, ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಔಟ್‌ಪುಟ್ ರೀಡೈರೆಕ್ಷನ್ ಮತ್ತು ಈ ವಿಭಾಗವು ಈ ಕಮಾಂಡ್ ನ ಕೆಲವು ಸಾಮಾನ್ಯ ಉಪಯೋಗಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.

Cat ಚಿಕ್ಕದಾದ ಫೈಲ್‌ಗಳನ್ನು ಓದಲು ಉಪಯುಕ್ತವಾಗಿದ್ದರೂ, ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಫೈಲ್‌ಗಳನ್ನು ಸಂಯೋಜಿಸಲು, ರಚಿಸಲು, ತಿದ್ದಿಬರೆಯಲು ಅಥವಾ ಅಪೆಂಡ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಚಿಕ್ಕ ಫೈಲನ್ನು ನೋಡಲು cat ಕಮಾಂಡ್‌ನ್ನು ಹೀಗೆ ಬಳಸಬಹುದು:

$ cat test.txt

This text file was created by the cat command.

Cat could be the world`s simplest text editor.

If you read this book, you’ll learn how to use cat.

This is the last line of text in this file.

Cat ಕಮಾಂಡ್ ಸಹ ಅನೇಕ ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ. ನಿಮ್ಮ ಫೈಲನ್ನು ಸಾಲಿನ ಸಂಖ್ಯೆಯ ಜೊತೆಗೆ ನೋಡಲು ಬಯಸಿದರೆ, -n ಆಯ್ಕೆಯನ್ನು ನೀವು ಉಪಯೋಗಿಸಬಹುದು :

$cat -n test.txt

- This text file was created by the cat command.
- Cat could be the world’s simplest text editor.
- If you read this book, you’ll learn how to use cat.
- This is the last line of text in this file.

ಹಲವಾರು ಫೈಲ್ ಗಳನ್ನು ಏಕಕಾಲದಲ್ಲಿ ನೋಡಲು ಬಯಸಿದರೆ cat ಅನ್ನು ಬಳಸಿ, ಏಕೆಂದರೆ cat ವೈಲ್ಡ್‌ಕಾರ್ಡ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ,

$ cat -n test*

- This text file was created by the cat command.
- Cat could be the world’s simplest text editor.
- If you read this book, you’ll learn how to use cat.
- This is the last line of text in this file.
- This is the first line of test2.txt.
- This file was also created by cat.
- This is the last line of test2.txt.

© NIMI NOT TO BE REPUBLISHED

ನೀವು ನೋಡಿದಂತೆ cat ತನ್ನ ಔಟ್‌ಪುಟ್ ನಲ್ಲಿ ಎರಡನೆಯ ಫೈಲನ್ನು ಸೇರಿಸಿಕೊಂಡಿದೆ. ಮತ್ತು ಔಟ್‌ಪುಟ್‌ನ ಪ್ರತಿ ಸಾಲು ಸಂಖ್ಯೆಯನ್ನು ಹೊಂದಿದೆ. ಕೆಳಗಿನ ಕಮಾಂಡ್ ಉಪಯೋಗಿಸಿ ಎರಡು ಫೈಲ್‌ಗಳನ್ನು ಒಟ್ಟಿಗೆ ನೋಡಬಹುದೆಂದು ಗಮನಿಸಿ:

$ cat test.txt test2.txt

ಈ ಔಟ್ಪುಟ್ ವೈಲ್ಡ್ ಕಾರ್ಡ್ ಬಳಸಿದ ಔಟ್ಪುಟ್‌ನಂತೆಯೆ ಇರುತ್ತದೆ. ನೀವು cat ಕಮಾಂಡ್‌ನ್ನು ರಿಡೈರೆಕ್ಷನ್ ಆಪರೇಟರ್(>) ನೊಂದಿಗೂ ಕೂಡ ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ, ನೀವು test.txt ಮತ್ತು test2.txt ಗಳನ್ನು test3.txt ಎನ್ನುವ ಮೂರನೆಯ ಫೈಲ್‌ನಲಿ ್ಲ ಸಂಯೋಜಿಸಲು ಬಯಸಿದರೆ ನೀವು ಈ ಕೆಳಗಿನ ಕಮಾಂಡ್ ಬಳಸಬಹುದು:

$ cat test* > test3.txt

ಇದರ ಫಲಿತಾಂಶವನ್ನು ಕೆಳಗಿನ ಕಮಾಂಡ್‌ನೊಂದಿಗೆ ಪರಿಶೀಲಿಸಿ

 # ls -l test*

ಈ ಸಂದರ್ಭದಲ್ಲಿ, ಬಳಕೆದಾರರು test.txt ನಲ್ಲಿನ ಮಾಹಿತಿಯು test2.txt ಗೆ ಹೋಗುವುದನ್ನು ಬಯಸುತ್ತಾರೊ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸಿ ಅಥವಾ test2.txt ದಲ್ಲಿನ ಮಾಹಿತಿಯು test.txt ಗೆ ಹೋಗುತ್ತದೆ. ನಂತರ cat ಅನ್ನು >> ರೀಡೈರೆಕ್ಷನ್ ಆಪರೇಟರ್‌ನೊಂದಿಗೆ ಈ ರೀತಿಯಾಗಿ ಬಳಸಿ:

$ cat test.txt >> test2.txt

ಇದು test.txt ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು test2.txt ನ ಕೊನೆಯಲ್ಲಿ ಅಪೆಂಡ್ ಮಾಡುತ್ತದೆ. ಫಲಿತಾಂಶವನ್ನು ಪರಿಶೀಲಿಸಲು cat ಅನ್ನು ಮತ್ತೆ ಈ ರೀತಿ ಬಳಸಿ:

$ cat test2.txt

This is the first line of test2.txt.

This file was also created by cat.

This is the last line of test2.txt.

This text file was created by the cat command.

Cat could be the worldಕs simplest text editor.

If you read this book, youಕll learn how to use cat.

This is the last line of text in this file.

ಗಮನಿಸಿ , ನೀವು ಕಮಾಂಡ್ ಅನ್ನು ಈ ಕೆಳಗಿನಂತೆ ನೀಡಿದರೆ:

$ cat -n test.txt >>test2.txt

test2.txt ಫೈಲ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

$ cat test2.txt

This is the first line of test2.txt.

This file was also created by cat.

This is the last line of test2.txt.

- This text file was created by the cat command.
- Cat could be the worldಕs simplest text editor.
- If you read this book, youಕll learn how to use cat.
- This is the last line of text in this file.

ಕೊನೆಯದಾಗಿ, ನೀವು ವರ್ಡ್ ಪ್ರಾಸೆಸರ್ ಅಥವಾ ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್ ಬಳಸದೇ ಒಂದು ಚಿಕ್ಕ ಟೆಕ್ಸ್ಟ್ ಫೈಲನ್ನು ರಚಿಸಲು ಬಯಸಿದರೆ ಅದಕ್ಕಾಗಿ ಒಂದು ಟ್ರಿಕ್ ಇಲ್ಲಿದೆ. cat ಕಮಾಂಡ್ ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಇನ್ಪುಟ್‌ನ್ನು ಓದಬಹುದಾಗಿರುವುದರಿಂದ ನೀವು cat ಕಮಾಂಡ್ ಬಳಸಿ ಫೈಲನ್ನು ರಚಿಸಿರಿ ಮತ್ತು ಅದನ್ನು ಕೀಸ್ಟ್ರೋಕ್‌ಗಳಿಂದ ತುಂಬಿರಿ. ಅದು ಹೇಗೆ ಎಂಬುದು ಇಲ್ಲಿದೆ:

$ cat > myfile.txt

ಈಗ ಕೆಲವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ನಮೂದಿಸಿ:

$ cat > myfile.txt

This is the cat word processor.

This is the end of the file

ನಂತರ, ನೀವು ಟೈಪಿಂಗ್ ಮಾಡುವುದು ಮುಗಿದ ತಕ್ಷಣ ಫೈಲನ್ನು ಮುಚ್ಚಲು CTRL +D ಒತ್ತಿರಿ.

ಇದು ಕೆಲಸ ಮಾಡುತ್ತದೆ ಎಂಬುದನ್ನು ನೊಡಲು, ಕೆಳಗಿನಂತೆ ಪ್ರಯತ್ನಿಸಿ,

$ ls -l myfile.txt

-rw-rw-r-- 1 bball bball 61 Nov 12 18:26 myfile.txt

$ cat myfile.txt

This is the cat word processor.

This is the end of the file

cat ಕಮಾಂಡ್, ಟೆಕ್ಸ್ಟ್ ಫೈಲ್ ಮಾತ್ರವಲ್ಲದೇ ಯಾವುದೇ ಫೈಲ್‌ನಲಿ ್ಲರುವ ಮಾಹಿತಿಯನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಬಲ್ಲದು ಎನ್ನುವುದನ್ನು ಸಹ ಬಳಕೆದಾರರು ತಿಳಿದಿರಬೇಕು. cat ಕಮಾಂಡ್ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಚಿಕ್ಕ ಫೈಲ್‌ಗಳನ್ನು ನೊಡಲು ಉಪಯುಕ್ತವಾಗಿದೆ.

ದಿನಾಂಕವನ್ನು ಪ್ರಾದರ್ಶಿಸಲು ಮತ್ತು ಸೆಟ್ ಮಾಡಲು Date ಕಮಾಂಡ್‌ನ ಉದಾಹರಣೆಗಳು

ದಿನಾಂಕವನ್ನು ಅನೇಕ ಸ್ವರೂಪಗಳಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲು Date ಕಮಾಂಡ್ ಸಹಾಯಕವಾಗಿದೆ. ಇದು ಸಿಸ್ಟಮ್‌ನ ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ಸೆಟ್ ಮಾಡಲು ನಿಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ.

Date ಕಮಾಂಡ್‌ನ್ನು ಹೇಗೆ ಬಳಸುವುದು ಎಂಬುದನ್ನು ಪ್ರಾಯೋಗಿಕ ಉದಾಹರಣೆಗಳೊಂದಿಗೆ ಕೆಲವನ್ನು ಇಲ್ಲಿ ನೀಡಲಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಯಾವುದೇ ಆಯ್ಕೆಗಳಿಲ್ಲದೇ Date ಕಮಾಂಡ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಿದಾಗ(execute), ಅದು ಪ್ರಸ್ತುತ ದಿನಾಂಕ ಮತ್ತು ಸಮಯನ್ನು ಕೆಳಗಿನಂತೆ ಪ್ರದರ್ಶಿಸುತ್ತದೆ :

$ date

Mon May 20 22:02:24 PDT 2013

1. -date **ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ಸ್ಟ್ರಿಂಗ್ ಮೌಲ್ಯದಿಂದ ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು.**

ನೀವು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಸ್ಥಿರ ದಿನಾಂಕ ಅಥವಾ ಸಮಯವನ್ನು ಹೊಂದಿದ್ದರೆ, ನೀವು ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ -d ಅಥವಾ -date ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ಇನ್ಪುಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು date ಫಾರ್ಮ್ಯಾಟ್‌ಗೆ ಪರಿವರ್ತಿಸಬಹುದು.

ಇದು ಪ್ರಸ್ತುತ ದಿನಾಂಕ ಮತ್ತು ಸಮಯದ ಮೌಲ್ಯವನ್ನು ಉಪಯೋಗಿಸುವುದಿಲ್ಲ ಎಂಬುದನ್ನು ಗಮನಿಸಿ. ಬದಲಿಗೆ ಇದು ನೀವು ಸ್ಟ್ರಿಂಗ್ ಆಗಿ ನೀಡಿದ ದಿನಾಂಕ ಮತ್ತು ಸಮಯದ ಮೌಲ್ಯವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

ಕೆಳಗಿನ ಉದಾಹರಣೆಗಳು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಮಾತ್ರ ಇನ್ಪುಟ್ ದಿನಾಂಕವನ್ನಾಗಿ ತೆಗೆದುಕೊಳ್ಳುತ್ತವೆ ಮತ್ತು ಔಟ್‌ಪುಟ್‌ನ್ನು date ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸುತ್ತವೆ. ನೀವು ಸಮಯವನ್ನು ಸೂಚಿಸದಿದ್ದರೆ, ಸಮಯಕ್ಕಾಗಿ 00:00:00 ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

$ date --date=``12/2/2014``

Tue Dec 2 00:00:00 PST 2014

$ date --date=``2 Feb 2014``

Sun Feb 2 00:00:00 PST 2014

$ date --date=``Feb 2 2014``

Sun Feb 2 00:00:00 PST 2014

ಕೆಳಗಿನ ಉದಾಹರಣೆ ಇನ್ಪುಟ್ ದಿನಾಂಕ ಮತ್ತು ಸಮಯ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಔಟ್‌ಪುಟ್‌ನ್ನು date ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

$ date --date=``Feb 2 2014 13:12:10``

Sun Feb 2 13:12:10 PST 2014

2. -file **ಆಯ್ಕೆಯನ್ನು ಬಳಸಿಕೊಂಡು ಫೈಲ್‌ನಿಂದ ದಿನಾಂಕದ ಮಾದರಿಗಳನ್ನು ಓದುವುದು.**

ಇದು ನಾವು ಮೇಲೆ ಚರ್ಚಿಸಿದ -d ಅಥವಾ -date ಆಯ್ಕೆಗಳಿಗೆ ಸಮನಾಗಿದೆ. ಆದರೆ, ಇದನ್ನು ನೀವು ಅನೇಕ date ಸ್ಟ್ರಿಂಗ್‌ಗಳಿಗೆ ಉಪಯೋಗಿಸಬಹುದು. ವಿವಿಧ ಸ್ಥಿರ date ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ -f ಅಥವಾ -file ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, datefile, 2 date ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಹೊಂದಿರುವುದನ್ನು ನಾವು ನೋಡಬಹುದು. datefile ನ ಪ್ರತಿಯೊಂದು ಸಾಲು date ಕಮಾಂಡ್‌ನಿಂದ ಪಾರ್ಸ್ ಮಾಡಲ್ಪಡುತ್ತದೆ ಮತ್ತು ಪ್ರತಿ ಸಾಲಿಗೆ date ಅನ್ನು ಔಟ್‌ಪುಟ್ ಮಾಡಲಾಗುತ್ತದೆ.

$ cat datefile

Sept 9 1986

Aug 23 1987

$ date --file =datefile

Tue Sep 9 00:00:00 PDT 1986

Sun Aug 23 00:00:00 PDT 1987

3. -date **ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ಸಂಬಂಧಿತ ದಿನಾಂಕವನ್ನು ಪಡೆಯುವುದು.**

ಸಂಬಂಧಿತ ಮೌಲ್ಯಗಳನ್ನು ಬಳಸಿ, ಭವಿಷ್ಯದ ದಿನಾಂಕವನ್ನು ಪಡೆಯಲು ನೀವು Date ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆಗೆ, ಕೆಳಗಿನ ಉದಾಹರಣೆಯು ಮುಂದಿನ ಸೋಮವಾರದ ದಿನಾಂಕವನ್ನು ನೀಡುತ್ತದೆ.

$ date --date=``next mon``

Mon May 27 00:00:00 PST 2013

Date ಕಮಾಂಡ್‌ಗೆ String=@ ಎಂದು ನೀಡಿದರೆ, Date ಕಮಾಂಡ್ ಯುಗಾರಂಭದಿಂದ(1970-01-01 UTC) ಸೆಕೆಂಡ್‌ಗಳನ್ನು ದಿನಾಂಕದವರೆಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ.

ಇದು ಯುಗಾರಂಭದಿಂದ 1970-01-01 UTC 5 ಸೆಕೆಂಡುಗಳನ್ನು ಕಳೆದು ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ :

$ date --date = @5

Wed Dec 31 16:00:05 PST 1969

ಇದು ಯುಗಾರಂಭ 1970-01-01 UTC ದಿಂದ 10 ಸೆಕೆಂಡುಗಳನ್ನು ಕಳೆದು ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ :

$ date --date = @10

Wed Dec 31 16:00:10 PST 1969

ಇದು ಯುಗಾರಂಭ 1970-01-01 UTC ದಿಂದ 1 ನಿಮಿಷವನ್ನು(ಎಂದರೆ 60 ಸೆಕೆಂಡುಗಳನ್ನು) ಕಳೆದು ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ :

© NIMI NOT TO BE REPUBLISHED

$ date --date = @60

Wed Dec 31 16:01:00 PST 1969

4. ಹಿಂದಿನ ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು.

-date ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಹಿಂದಿನ ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸಬಹುದು. ಕೆಲವು ಸಾಧ್ಯತೆಗಳನ್ನು ಕೆಳಗೆ ತೋರಿಸಲಾಗಿದೆ.

$ date --date = `3 seconds ago`

Mon May 20 21:59:20 PDT 2013

$ date --date = `1 day ago`

Sun May 19 21:59:36 PDT 2013

$ date --date = `yesterday`

Sun May 19 22:00:26 PDT 2013

$ date --date = `1 month ago`

Sat Apr 20 21:59:58 PDT 2013

$ date --date = `1 year ago`

Sun May 20 22:00:09 PDT 2012

5. -set ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ನಿಗದಿಪಡಿಸುವುದು.

-s ಅಥವಾ -set ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ, ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ ನಿಮ್ಮ ಸಿಸ್ಟಮ್‌ನ ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ಹೊಂದಿಸಬಹುದು.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಪ್ರಾರಂಭದಲ್ಲಿ ಇದು ಸಮಯವನ್ನು 20:09:31 ಎಂದು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ನಂತರ ನಾವು ಅದನ್ನು 21:00:00 ಎಂದು ಬದಲಾಯಿಸಲು date ಕಮಾಂಡ್‌ನ್ನು ಬಳಸಿದ್ದೇವೆ.

$ date

Sun May 20 20:09:31 PDT 2013

$ date -s ``Sun May 20 21:00:00 PDT 2013``

Sun May 20 21:00:00 PDT 2013

$ date

Sun May 20 21:00:00 PDT 2013

6. -u ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ವಿಶ್ವವ್ಯಾಪಿ ಸಮಯವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು.

-u , ಅಥವಾ -utc ಅಥವಾ -universal ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ನೀವು ದಿನಾಂಕವನ್ನು ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ UTC ಸ್ವರೂಪದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಬಹುದು.

$ date

Mon May 20 22:07:53 PDT 2013

$ date -u

Tue May 21 05:07:55 PDT 2013

7. -r ಆಯ್ಕೆಯನ್ನು ಬಳಸಿ ಕೊನೆಯ ಮಾರ್ಪಾಡು ಸಮಯವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಪ್ರಸ್ತುತ ಸಮಯವು 20:25:48 ಆಗಿದೆ.

$ date

Sun May 20 20:25:48 PDT 2013

Touch ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು datefile ನ ಸಮಯವನ್ನು ಬದಲಾಯಿಸಲಾಗಿದೆ. ಮೇಲಿನ date ಕಮಾಂಡ್‌ನ ಔಟ್‌ಪುಟ್ ನ ಕೆಲವೇ ಸೆಕೆಂಡುಗಳ ನಂತರ ಇದನ್ನು ಮಾಡಲಾಗಿದೆ.

$ touch datefile

ಮೇಲಿನ Touch ಕಮಾಂಡ್‌ನ ನಂತರ ಪ್ರಸ್ತುತ ಸಮಯ 20:26:12 ಆಗಿದೆ.

$ date

Sun May 20 20:26:12 PDT 2013

ಅಂತಿಮವಾಗಿ, date ಕಮಾಂಡ್‌ನ -r ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸುವುದು. ಇದು ಕೆಳಗೆ ತೋರಿಸಿದಂತೆ, ಒಂದು ಫೈಲ್‌ನ ಕೊನೆಯ ಮಾರ್ಪಾಡು ಸಮಯವನ್ನು ಪ್ರಾದರ್ಶಿಸುತ್ತದೆ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಇದು datefile ನ ಕೊನೆಯ ಮಾರ್ಪಾಡು ಸಮಯವನ್ನು 20:25:57 ಎಂದು ತೋರಿಸುತ್ತದೆ. ಇದು 20:25:57 ಮತ್ತು 20:26:12 ರ ನಡುವೆ ಎಲ್ಲೋ ಇದೆ. (ಇದು ಸಮಯವನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಲು ಮೇಲಿನ touch ಕಮಾಂಡ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಿದಾಗ)

$ date -r datefile

Sun May 20 20:25:57 PDT 2013

8. ವಿವಿಧ date ಕಮಾಂಡ್ ಸ್ವರೂಪಗಳು

Date ಕಮಾಂಡ್‌ನ್ನು ವಿವಿಧ ಸ್ವರೂಪಗಳಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲು ನೀವು ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಅದಕ್ಕಾಗಿ ಕೆಳಗಿನ ಸಿಂಟ್ಯಾಕ್ಸ್ ಉಪಯೋಗಿಸಿ :

$ date +%<format-option>

© NIMI NOT TO BE REPUBLISHED

	ಕಮಾಂಡ್ (Command)	ವಿವರಣೆ (Description)
•	Apropos whatis	ಸ್ಟ್ರಿಂಗ್‌ಗೆ ಸಂಬಂಧಪಟ್ಟ ಕಮಾಂಡ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. ಥ್ರೆಡ್ ಸೇಫ್(threadsafe) ಸಹ ನೋಡಿ.
•	Man -t ascii | ps2pdf - > ascii.pdf	ಮ್ಯಾನುಯಲ್ ಪುಟದ ಪಿಡಿಎಫ್‌ನ್ನು ತಯಾರಿಸುತ್ತದೆ.
•	Which command	ಕಮಾಂಡ್‌ನ ಪೂರ್ಣ ಮಾರ್ಗದ(ಪಾಥ್) ಹೆಸರನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Time command	ಒಂದು ಕಮಾಂಡ್ ಎಷ್ಟು ಸಮಯವನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ತೋರಿಸುತ್ತದೆ.
•	Time cat	ಸ್ಟಾಪ್‌ವಾಚ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ನಿಲ್ಲಿಸಲು Ctrl -d ಬಳಸಿ. Sw ಯನ್ನು ಸಹ ನೋಡಿ.
Dir navigation		
•	Cd -	ಹಿಂದಿನ ಡೈರೆಕ್ಟರಿಗೆ ಹೋಗಲು
•	Cd	$ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗೆ ಹೋಗಲು.
•	(cd dir && command)	Dir ಗೆ ಹೋಗಿ, ಕಮಾಂಡ್ ಎಕ್ಸಿಕ್ಯೂಟ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಹಿಂತಿರುಗುತ್ತದೆ.
•	Pushd .	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ಸ್ಟ್ಯಾಕ್‌ನಲ್ಲಿ ಇರಿಸಿ, ನೀವು ಹಿಂತಿರುಗಿ popd ಮಾಡಬಹುದು.
File searching		
•	Alias l = `is -l --color=auto`	ತ್ವರಿತವಾಗಿ Dir ಪಟ್ಟಿ ಮಾಡಬಹುದು. I ನ್ನೂ ಸಹ ನೋಡಿ
•	Is -lrt	ದಿನಾಂಕದ ಆಧಾರದ ಮೇಲೆ ಫೈಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ. newest ಮತ್ತು find_mm_yyyy ಗಳನ್ನೂ ನೋಡಿ
•	Is /usr/bin | pr -T9 -W$COLUMNS	ಟರ್ಮಿನಲ್ ಅಗಲಕ್ಕೆ 9 ಕಾಲಮ್‌ಗಳಲ್ಲಿ ಮುದ್ರಿಸಿ.
•	Find -name `*.[ch]` | xargs -E `expr`	ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ `expr`ನ್ನು ಹುಡುಕಿ. findrepo ಕೂಡ ಹುಡುಕಿ.
•	Find -type f -print0 | xargs -r0 grep -F `example`	ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಮತ್ತು ಕೆಳಗೆ `example` ಅನ್ನು ಎಲ್ಲ ಸಾಮಾನ್ಯ ಫೈಲ್‌ಗಳಲ್ಲಿ ಹುಡುಕಿ
•	Find -maxdepth 1 -type f | xargs grep -F `example`	ಈ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ `example` ಅನ್ನು ಎಲ್ಲ ಸಾಮಾನ್ಯ ಫೈಲ್‌ಗಳಲ್ಲಿ ಹುಡುಕಿ
•	Find -maxdepth 1 -type d | while read dir; do echo $dir; echo cmd2; done	ಪ್ರತಿಯೊಂದು ಅಂಶವನ್ನು ಅನೇಕ ಕಮಾಂಡ್‌ಗಳೊಂದಿಗೆ ಸಂಸ್ಕರಿಸಿ. (while ಲೂಪ್ ನಲ್ಲಿ)
•	Find -type f ! -perm -444	ಎಲ್ಲರೂ ಓದಲಾರದ ಫೈಲ್‌ಗಳನ್ನು ಹುಡುಕಿ (ವೆಬ್‌ಸೈಟ್‌ಗೆ ಉಪಯುಕ್ತವಾಗಿದೆ)
•	Find -type d ! -perm -111	ಎಲ್ಲರೂ ಪ್ರವೇಶಿಸಲಾಗದ ಡೈರೆಕ್ಟರಿಗಳನ್ನು ಹುಡುಕಿ(ವೆಬ್‌ಸೈಟ್‌ಗೆ ಉಪಯುಕ್ತವಾಗಿದೆ)
•	Locate -r `file[^/]*\.txt`	ಹೆಸರುಗಳಿಗಾಗಿ ಸಂಗ್ರಹಿಸಿದ ಸೂಚಿಕೆಯನ್ನು ಹುಡುಕಿ. ಈ re , glob *file*.txt ಅನ್ನು ಹೋಲುತ್ತದೆ.
•	Look reference	ಪ್ರಿಫಿಕ್ಸ್‌ಗಾಗಿ (ವಿಂಗಡಿಸಲಾದ) ನಿಘಂಟುವನ್ನು ತ್ವರಿತವಾಗಿ ಹುಡುಕುತ್ತದೆ.
•	Grep -- color reference /usr/share/dict/words	ನಿಘಂಟುವಿನಲ್ಲಿ ಕಂಡುಬರುವ ರೆಗ್ಯುಲರ್ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ಗಳನ್ನು ಹೈಲೈಟ್ ಮಾಡುತ್ತದೆ.
Archieve and compression		
•	Gpg -c file	ಫೈಲನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡುತ್ತದೆ.
•	Tar -c dir/ | bzip2 > dir.tar.bz2	Dir/ ನ ಸಂಕುಚಿತ ಆರ್ಕೈವ್ ತಯಾರಿಸುತ್ತದೆ.
•	Bzip2 -dc dir.tar.bz2 | tar -x	ಆರ್ಕೈವ್‌ನ್ನು ಎಕ್ಸ್‌ಟ್ರ್ಯಾಕ್ಟ್ ಮಾಡುತ್ತದೆ(tar.gz ಫೈಲ್‌ಗಳಿಗೆ bzip2 ಬದಲಾಗಿ gzip ಉಪಯೋಗಿಸುತ್ತದೆ.)
•	Tar -c dir/ | gzip | gpg -c | ssh user@remote `dd of=dir.tar.gz.gpg`	ರಿಮೋಟ್ ಮಶೀನ್‌ನಲ್ಲಿ Dir/ ನ ಎನ್ ಕ್ರಿಪ್ಟೆಡ್ ಆರ್ಕೈವನ್ನು ತಯಾರಿಸುತ್ತದೆ.
•	Find dir/ -name `*.txt` | xargs cp -a --target-directory=dir_txt/ --parents	Dir/ ಮತ್ತು below ಇವುಗಳ ಸಬ್ ಸೆಟ್‌ನ ಪ್ರತಿಯನ್ನು ತಯಾರಿಸುತ್ತದೆ
•	(tar -c/dir/to/copy) | (cd/where/to/ && tar -x -p	Copy/ dir ದಿಂದ /where/to/ dir ಗೆ ಕಾಪಿ (ಅನುಮತಿಗಳೊಂದಿಗೆ) ಮಾಡುವುದು

© NIMI NOT TO BE REPUBLISHED

•	(cd/dir/to/copy && tar -c .) | (cd /where/ to/&& tar -x -p	Copy/ dir ದಿಂದ /where/to/ ಗೆ ಕಾಪಿ (ಅನುಮತಿಗಳೊಂದಿಗೆ) ಮಾಡುವುದು
•	(tar -c/dir/to/copy) | ssh -C user@re- mote `cd/where/to/ && tar -x -p	Copy/ dir ದಿಂದ remote:/where/to/ dir ಗೆ ಕಾಪಿ (ಅನುಮತಿಗಳೊಂದಿಗೆ) ಮಾಡುವುದು
•	Dd bs=1M if=/dev/sda | gzip | ssh user@ remote `dd of=sda.gz`	ಹಾರ್ಡ್ ಡಿಸ್ಕ್‌ನಿಂದ ರಿಮೋಟ್ ಮಶೀನ್‌ಗೆ ಬ್ಯಾಕಪ್ ಮಾಡುವುದು.
Rsync (Network efficient file copier : Use the --dry-run option for testing)		
•	Rsync -P rsync://rsync.server.com/path/ to/file file	ವ್ಯತ್ಯಾಸಗಳನ್ನು ಮಾತ್ರ ಪಡೆಯಿರಿ. ಕ್ಲಿಷ್ಟಕರ ಡೌನ್‌ಲೋಡ್‌ಗಳನ್ನು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಬಾರಿ ಮಾಡುತ್ತದೆ
•	Rsync --bwlimit=1000 fromfile tofile	ಸ್ಥಳೀಯವಾಗಿ ದರ ಮಿತಿಯೊಂದಿಗೆ ಕಾಪಿ ಮಾಡಿ. ಇದು I/O ಗಾಗಿ ಒಳ್ಳೆಯದು.
•	Rsync -az -e ssh --delete ~/public_html/ remote.com:`~/public_html`	ಮಿರರ್ ವೆಬ್‌ಸೈಟ್. (ಸಂಕೋಚನ ಮತ್ತು ಎನ್‌ಕ್ರಿಪ್ಷನ್ ಉಪಯೋಗಿಸಿ)
•	Rsync -auz -e ssh remote:/dir/ . && rsync -auz -e ssh . remote:/dir/	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯನ್ನು ರಿಮೋಟ್ ಡೈರೆಕ್ಟರಿಯೊಂದಿಗೆ ಸಿಂಕ್ರೊನೈಸ್ ಮಾಡುತ್ತದೆ
Ssh (Secure SHell)		
•	Ssh $USER@$HOST command	$HOST ನಲ್ಲಿ $USER ನಂತೆ ಕಮಾಂಡ್‌ನ್ನು ರನ್ ಮಾಡುತ್ತದೆ
•	Ssh -f -y $USER@$HOSTNAME xeyes	$HOSTNAME ನಲ್ಲಿ $USER ನಂತೆ GUI ಕಮಾಂಡ್‌ನ್ನು ರನ್ ಮಾಡುತ್ತದೆ
•	Scp -p -r $USER@$HOST: file dir/	$HOST ನಲ್ಲಿ $USER ನ ಹೋಮ್ ಡೈರೆಕ್ಟರಿಗೆ ಅನುಮತಿಗಳೊಂದಿಗೆ ಕಾಪಿ ಮಾಡುತ್ತದೆ
•	Scp -c arcfour $USER@$LANHOST: bigfile	ಸ್ಥಳೀಯ ಲ್ಯಾನ್‌ಗೆ ವೇಗದ ಕ್ರಿಪ್ಟೊ ಬಳಸಿ. ಇದು GigE ಯನ್ನು ಸ್ಯಾಚುರೆಟ್ ಮಾಡಬಹುದು.
•	Ssh -g -L 8080:localhost:80 root@$HOST	$HOSTNAME : 8080 ದಿಂದ $HOST : 80 ವರೆಗೆ ಸಂಪರ್ಕಗಳನ್ನು ಫಾರ್ವರ್ಡ್ ಮಾಡುತ್ತದೆ
•	Ssh -R 1434:imap:143 root@$HOST	$HOST : 1434 ದಿಂದ imap : 143 ವರೆಗೆ ಸಂಪರ್ಕಗಳನ್ನು ಫಾರ್ವರ್ಡ್ ಮಾಡುತ್ತದೆ
•	Ssh-copy-id $USER@$HOST	ಪಾಸ್‌ವರ್ಡ್ ಇಲ್ಲದ ಲಾಗಿನ್‌ಗಾಗಿ $USER@$HOST ಗಾಗಿ ಪಬ್ಲಿಕ್ ಕೀ ಯನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುತ್ತದೆ
Networking (Note ifconfig, route, mii-tool, nslookup commands are obsolte)		
•	Ethtool eth0	ಈಥರ್‌ನೆಟ್ ಇಂಟರ್‌ಫೇಸ್ eth0 ನ ಸ್ಥಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Ethtool - change eth0 autoneg off speed 100 duplex full	ಈಥರ್‌ನೆಟ್ ಇಂಟರ್‌ಫೇಸ್ ವೇಗವನ್ನು ಹಸ್ತಚಾಲಿತವಾಗಿ ಸೆಟ್ ಮಾಡುತ್ತದೆ
•	Iw dev wlan0 link	ವೈರ್‌ಲೆಸ್ ಇಂಟರ್‌ಫೇಸ್ wlan0 ನ ಲಿಂಕ್ ಸ್ಥಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Iw dev wlan0 set bitrates legacy-2.4 1	ವೈರ್ ಲೆಸ್ ಇಂಟರ್‌ಫೇಸ್ ವೇಗವನ್ನು ಹಸ್ತಚಾಲಿತವಾಗಿ ಸೆಟ್ ಮಾಡುತ್ತದೆ
•	Iw dev wlan0 scan	ವ್ಯಾಪ್ತಿಯಲ್ಲಿರುವ ವೈರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ip link show	ನೆಟ್‌ವರ್ಕ್ ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ
•	Ip link set dev eth0 name wan	ಇಂಟರ್‌ಫೇಸ್ eth0 ನ್ನು wan ಗೆ ಮರುಹೆಸರಿಸುತ್ತದೆ.
•	Ip link set dev eth0 up	ಇಂಟರ್‌ಫೇಸ್ eth0 ನ್ನು ಮೇಲಕ್ಕೆ (ಅಥವಾ ಕೆಳಗೆ) ತರುತ್ತದೆ.
•	Ip addr show	ಇಂಟರ್‌ಫೇಸ್‌ಗಳಿಗಾಗಿ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ip addr add 1.2.3.4/24 brd + dev eth0	Ip ಮತ್ತು mask (255.255.255.0) ನ್ನು ಸೇರಿಸುತ್ತದೆ(ಅಥವಾ ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ)
•	Ip route show	ರೂಟಿಂಗ್ ಟೇಬಲ್‌ನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ
•	Ip route add default via 1.2.3.254	ಡೀಫಾಲ್ಟ್ ಗೇಟ್ ವೇಯನ್ನು 1.2.3.254 ಗೆ ಸೆಟ್ ಮಾಡುತ್ತದೆ
•	Ss -tupl	ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಯನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ss -tup	ಸಿಸ್ಟಮ್ ಗೆ/ನಿಂದ ಸಕ್ರೀಯ ಸಂಪರ್ಕಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ
•	Host pixelbeat.org	ಹೆಸರಿಗಾಗಿ DNS ip ಅಡ್ರೆಸ್‌ನ್ನು ಹುಡುಕುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

<table>
<tr><td>•</td><td>Hostname -i</td><td>ಸ್ಥಳೀಯ ip ಅಡ್ರೆಸ್‌ನ್ನು ಹುಡುಕುತ್ತದೆ(ಹೋಸ್ಟ್ ನ `hostname` ಸಮಾನವಾಗಿದೆ)</td></tr>
<tr><td>•</td><td>Whols pixelbeat.org</td><td>ಹೊಸ್ಟ್‌ನೇಮ್ ಅಥವಾ ip ಅಡ್ರೆಸ್‌ಗಾಗಿ whois ಮಾಹಿತಿಯನ್ನು ಹುಡುಕುವುದು.</td></tr>
<tr><td colspan="3">Windows networking (Note samba is the package that provides all this windows specific networking support)</td></tr>
<tr><td>•</td><td>Smbtree</td><td>ವಿಂಡೋಸ್ ಮಶೀನ್‌ಗಳನ್ನು ಹುಡುಕುತ್ತದೆ. findsmb ಸಹ ನೋಡಿರಿ</td></tr>
<tr><td>•</td><td>Nmblookup -A 1.2.3.4</td><td>Ip ಅಡ್ರೆಸ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ವಿಂಡೊಸ್(netbios) ಹೆಸರನ್ನು ಹುಡುಕುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Nmbclient -L windows_box</td><td>ವಿಂಡೊಸ್ ಮಶೀನ್ ಅಥವಾ ಸಾಂಬಾ ಸರ್ವರ್‌ನಲ್ಲಿನ ಶೇರುಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Mount -t smbfs - fmask=666,guest //windows_box/share/mnt/share</td><td>ವಿಂಡೊಸ್ ಶೇರನ್ನು ಮೌಂಟ್ ಮಾಡುತ್ತದೆ</td></tr>
<tr><td>•</td><td>Echo `message` | smbclient -M windows_box</td><td>ವಿಂಡೊಸ್ ಮಶೀನ್‌ಗೆ ಪಾಪ್‌ಅಪ್‌ನ್ನು ಕಳುಹಿಸುತ್ತದೆ.(XP sp2 ನಲ್ಲಿ ಡೀಫಾಲ್ಟ್ ಆಗಿ ಆಫ್ ಅಗಿರುತ್ತದೆ)</td></tr>
<tr><td colspan="3">Text manipulation (Note sed uses stdin and stdout. Newer versions support inplace editing with the -I option)</td></tr>
<tr><td>•</td><td>Sed `s/string1/string2/g`</td><td>ಸ್ಟ್ರಿಂಗ್ 1 ನ್ನು ಸ್ಟ್ರಿಂಗ್ 2 ನಿಂದ ಬದಲಾಯಿಸುತ್ತದೆ</td></tr>
<tr><td>•</td><td>Sed `s/\(.*\)1/\12/g`</td><td>ಯಾವುದೇ ಸ್ಟ್ರಿಂಗ್ 1 ಅನ್ನು ಯಾವುದೇ ಸ್ಟ್ರಿಂಗ್ 2 ನಿಂದ ಮಾರ್ಪಡಿಸುತ್ತದೆ</td></tr>
<tr><td>•</td><td>Sed `/^*#/d;/^*$/d`</td><td>ಕಮೆಂಟ್‌ಗಳನ್ನು ಮತ್ತು ಖಾಲಿ ಸಾಲುಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed `:a;/\\$/N;s/\\\n//;ta`</td><td>ಸಾಲುಗಳನ್ನು ಹಿಂದಿನ \ ನೊಂದಿಗೆ ಜೋಡಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed `s/[\t]*$//`</td><td>ಸಾಲುಗಳಿಂದ ಹಿಂದಿನ ಸ್ಥಳಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed `s/\([```$\]\)/\\\1/g`</td><td>ಡಬಲ್ ಕೋಟ್‌ನಲ್ಲಿರುವ ಸಕ್ರೀಯವಾಗಿರುವ ಶೆಲ್ ಮೆಟಾಕ್ಯಾರೆಕ್ಟರ್‌ಗಳನ್ನು ಎಸ್ಕೆಪ್ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Seq 10 | sed ``s/^/ /;s/*\(.\{7,\}\)/\1/``</td><td>ಸಂಖ್ಯೆಗಳನ್ನು ಬಲಕ್ಕೆ ಜೋಡಿಸುತ್ತದೆ (right align)</td></tr>
<tr><td>•</td><td>Seq 10 | sed p | paste - -</td><td>ಕಾಲಮ್‌ನ್ನು ನಕಲು ಮಾಡುತ್ತದೆ(Duplicate)</td></tr>
<tr><td>•</td><td>Sed -n `1000{p;q}`</td><td>1000 ನೇ ಸಾಲನ್ನು ಮುದ್ರಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed -n `10,20;20q`</td><td>10 ರಿಂದ 20 ಸಾಲುಗಳನ್ನು ಮುದ್ರಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed -n `s/.*<title>\(.*\)<\/title>.*/\1/ip;T;q`</td><td>HTML ವೆಬ್‌ಪುಟದಿಂದ ಟೈಟಲ್‌ನ್ನು ಹೊರತೆಗೆಯುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed -I 42d ~/.ssh/known_hosts</td><td>ನಿರ್ದಿಷ್ಟ ಸಾಲನ್ನು ಡಿಲೀಟ್ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sed -t. k1,1n -k2,2n -k3,3n -k4,4n</td><td>IPV4 ip ಅಡ್ರೆಸ್‌ಗಳನ್ನು ವಿಂಗಡಿಸುತ್ತದೆ(sort)</td></tr>
<tr><td>•</td><td>Echo `Test` | tr `[:lower:]` `[:upper:]`</td><td>ಕೇಸ್‌ಗಳನ್ನು ಪರಿವರ್ತಿಸುತ್ತದೆ.(Uppercase/Lowercase)</td></tr>
<tr><td>•</td><td>Tr -dc `[:print:]` < /dev/urandom</td><td>ಪ್ರಿಂಟ್ ಮಾಡಲಾಗದ ಅಕ್ಷರಗಳನ್ನು ಫಿಲ್ಟರ್ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Tr -s `[:blank:]` `\t` </proc/diskstates |cut -f4</td><td>ಖಾಲಿ ಸ್ಥಳಗಳಿಂದ ಬೇರ್ಪಡಿಸಲ್ಪಟ್ಟ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಕಟ್ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>History | wc -1</td><td>ಸಾಲುಗಳನ್ನು ಎಣಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Seq 10 | paste -s -d ` `</td><td>ಸಾಲಿನಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು ಜೋಡಿಸುತ್ತದೆ ಮತ್ತು ಬೇರ್ಪಡಿಸುತ್ತದೆ.</td></tr>
<tr><td colspan="3">Ser operations (Note you can export LANG=C for sprrd. Also these assume no duplicate lines within a file</td></tr>
<tr><td>•</td><td>Sort file1 file2 | uniq</td><td>ವಿಂಗಡಿಸದ ಫೈಲ್‌ಗಳನ್ನು ಒಗ್ಗೂಡಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sort file1 file2 | uniq -d</td><td>ವಿಂಗಡಿಸದ ಫೈಲ್‌ಗಳನ್ನು ಪರಸ್ಪರ ವಿಭಾಗಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sort file1 file1 file2 | uniq u</td><td>ವಿಂಗಡಿಸದ ಫೈಲ್‌ಗಳನ್ನು ಭಿನ್ನಗೊಳಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Sort file1 file2 | uniq u</td><td>ವಿಂಗಡಿಸದ ಫೈಲ್‌ಗಳನ್ನು ಸಮ್ಮಿತಿಯ ವ್ಯತ್ಯಾಸ ತಿಳಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Join -t`\0` -a1 -a2 file1 file2</td><td>ವಿಂಗಡಿಸಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಒಗ್ಗೂಡಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Join -t`\0` file1 file2</td><td>ವಿಂಗಡಿಸಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಪರಸ್ಪರ ವಿಭಾಗಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Join -t`\0` -v2 file1 file2</td><td>ವಿಂಗಡಿಸಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಭಿನ್ನಗೊಳಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Join -t`\0` -v1 -v2 file1 file2</td><td>ವಿಂಗಡಿಸಲಾದ ಫೈಲ್‌ಗಳನ್ನು ಸಮ್ಮಿತಿಯ ವ್ಯತ್ಯಾಸ ತಿಳಿಸುತ್ತದೆ.</td></tr>
<tr><td colspan="3">Math</td></tr>
<tr><td>•</td><td>echo `(1 + sqrt9500/2` | bc -1</td><td>ತ್ವರಿತ ಗಣಿತ (Ø ಕಂಡುಹಿಡಿಯುವುದು). bc ಸಹ ನೋಡಿ,</td></tr>
</table>

NOT TO BE REPUBLISHED

<table>
<tr><td>•</td><td>Seq -f `4/%g` 1 2 99999 | paste -sd-+ | bc -l</td><td>ಯುನಿಕ್ಸ್ ಮಾರ್ಗದಲ್ಲಿ n ಕಂಡುಹಿಡಿಯುವುದು.</td></tr>
<tr><td>•</td><td>Echo `pad=20; min=64; (100*10^6)/((pad+min)*8)` | bc</td><td>ಹೆಚ್ಚು ಕ್ಲಿಷ್ಟಕರವಾದ (int) ಉದಾ : ಇದು max FastE ಪ್ಯಾಕೆಟ್ ದರವನ್ನು ತೋರಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Echo `pad=20; min=64; print (100E6)/((pad+min)*8)` | python</td><td>ಪೈಥಾನ್ ವೈಜ್ಞಾನಿಕ ಸಂಕೇತವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Exho `pad=20; plot [64:1518] (100*10**6)/((pad+x)*8)` | gnuplot -persist</td><td>FastE ಪ್ಯಾಕೆಟ್ ದರವನ್ನು ಪ್ಯಾಕೆಟ್ ಗಾತ್ರದೊಂದಿಗೆ ಪ್ಲಾಟ್ ಮಾಡುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Echo `obase=16; ibase=10; 64206` | bc</td><td>ಬೇಸ್ ಪರಿವರ್ತನೆಯನ್ನು ಮಾಡುತ್ತದೆ. (ಡೆಸಿಮಲ್‌ದಿಂದ ಹೆಕ್ಸಾಡೆಸಿಮಲ್)</td></tr>
<tr><td>•</td><td>Echo $((0x2dec))</td><td>ಬೇಸ್ ಪರಿವರ್ತನೆಯನ್ನು ಮಾಡುತ್ತದೆ. (ಹೆಕ್ಸಾಡೆಸಿಮಲ್‌ದಿಂದ ಡೆಸಿಮಲ್)((ಶೆಲ್ ಅಂಕಗಣಿತದ ವಿಸ್ತರಣೆ))</td></tr>
<tr><td>•</td><td>Units -t `100m/9.58s` `miles/hour`</td><td>ಯುನಿಟ್ ಪರಿವರ್ತನೆ(ಮೆಟ್ರಿಕದಿಂದ ಇಂಪಿರಿಯಲ್)</td></tr>
<tr><td>•</td><td>Units -t `500gb` `GiB`</td><td>ಯುನಿಟ್ ಪರಿವರ್ತನೆ(ಎಸ್‌ಐದಿಂದ ಐಇಸಿ ಪ್ರಿಫಿಕ್ಸಗಳಿಗೆ)</td></tr>
<tr><td>•</td><td>Units -t `1 googol`</td><td>ಡೆಫಿನಿಷನ್ ಲುಕಪ್</td></tr>
<tr><td>•</td><td>Seq 100 | paste -s -d+ | bc</td><td>ಸಂಖ್ಯೆಗಳ ಕಾಲಮ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ. add ಮತ್ತು funcpy ಗಳನ್ನೂ ಸಹ ನೋಡಿ.</td></tr>
<tr><td colspan="3">Calendar</td></tr>
<tr><td>•</td><td>Cal -3</td><td>ಕ್ಯಾಲೆಂಡರ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Cal 9 1752</td><td>ನಿರ್ದಿಷ್ಟ ತಿಂಗಳು, ವರ್ಷದ ಕ್ಯಾಲೆಂಡರ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>Date -d fri</td><td>ಈ ಶುಕ್ರವಾರ ಯಾವ ದಿನಾಂಕವಿರುತ್ತದೆ. ದಿನವನ್ನೂ ಕೂಡ ನೋಡಿ.</td></tr>
<tr><td>•</td><td>[$(date -d `12:00 today +1 day` +%d) = `01`] || exit</td><td>ಇದು ತಿಂಗಳ ಕೊನೆಯ ದಿನ ಅಲ್ಲದಿದ್ದರೆ ಸ್ಕ್ರಿಪ್ಟನಿಂದ ಹೊರಗೆ ಬನ್ನಿ.</td></tr>
<tr><td>•</td><td>Date --date=`25 Dec` +%A</td><td>ಈ ವರ್ಷ ಕ್ರಿಸ್‌ಮಸ್ ಯಾವ ದಿನದಂದು ಬರುತ್ತದೆ .</td></tr>
<tr><td>•</td><td>Date --date=`!2147483647`</td><td>ಯುಗಾರಂಭದಿಂದ (1970-01-01 UTC) ಸೆಕೆಂಡುಗಳನ್ನು ಇಲ್ಲಿಯವರೆಗಿನ ದಿನಾಂಕಕ್ಕೆ ಪರಿವರ್ತಿಸುತ್ತದೆ</td></tr>
<tr><td>•</td><td>TZ=`Amarica/Los_Angeles` date</td><td>ಯುಎಸ್‌ನ ಪಶ್ಚಿಮ ಕರಾವಳಿಯಲ್ಲಿ ಈಗ ಸಮಯವೆಷ್ಟು (TZ ಕಂಡುಹಿಡಿಯಲು tzselect ಅನ್ನು ಉಪಯೋಗಿಸಿ)</td></tr>
<tr><td>•</td><td>Date --date=`TZ=``Amarica/Los_Ange-les`` 09:00 next fri</td><td>ಯುಎಸ್ ನ ಪಶ್ಚಿಮ ಕರಾವಳಿಯಲ್ಲಿ ಮುಂದಿನ ಶುಕ್ರವಾರ 9 AM ನ ಸ್ಥಳಿಯ ಸಮಯವೆಷ್ಟು.</td></tr>
<tr><td colspan="3">Locales</td></tr>
<tr><td>•</td><td>Printf ``%`d\n``1234</td><td>ಲೋಕೆಲ್‌ಗೆ ಸೂಕ್ತವಾದ ಸಾವಿರಾರು ಗುಂಪುಗಳ ಸಂಖ್ಯೆಯನ್ನು ಮುದ್ರಿಸುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>BLOCK_SIZE=\`1 ls -l</td><td>ls ನಲ್ಲಿ ಲೊಕೆಲ್ ಸಾವಿರಾರು ಗುಂಪುಗಳನ್ನು ಬಳಸುವುದು. l ಕೂಡ ನೋಡಿ.</td></tr>
<tr><td>•</td><td>Echo ``I live in `locale territory```</td><td>ಲೋಕೆಲ್ ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿನ ಮಾಹಿತಿಯನ್ನು ಹೊರತೆಗೆಯುತ್ತದೆ.</td></tr>
<tr><td>•</td><td>LANG=en_IE.utf8 locale int_prefix</td><td>ನಿರ್ದಿಷ್ಟ ದೇಶದ ಲೋಕೆಲ್ ಮಾಹಿತಿಯನ್ನು ನೋಡಿ. Ccodes ಸಹ ನೋಡಿರಿ.</td></tr>
<tr><td>•</td><td>Locale -kc $(locale | sed -n `s/\(LC_.\{4,\}\)=.*/\1/p`) | less</td><td>ಲೋಕೆಲ್ ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಫೀಲ್ಡ್ ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿರಿ</td></tr>
<tr><td colspan="3">Recode(Obsoletes iconv, dos2unix, unix2dos)</td></tr>
<tr><td>•</td><td>Recode -l | less</td><td>ಲಭ್ಯವಿರುವ ಪರಿವರ್ತನೆಗಳನ್ನು ತೊರಿಸುತ್ತದೆ(ಪ್ರತಿ ಸಾಲಿನಲ್ಲಿರುವ ಅಲಿಯಾಸ್ ಗಳು)</td></tr>
<tr><td>•</td><td>recode windows-1252.. file_to_change.txt</td><td>ವಿಂಡೋಸ್ ``ansi`` ದಿಂದ ಲೋಕೆಲ್ charset ಗೆ ರೀಕೊಡ್ ಮಾಡುತ್ತದೆ(auto, CRLF ಪರಿವರ್ತನೆಯನ್ನು ಮಾಡುತ್ತದೆ)</td></tr>
<tr><td>•</td><td>Recode utf-8/CRLF.. file_to_change.txt</td><td>ವಿಂಡೋಸ್ ``utf8`` ದಿಂದ ಲೋಕೆಲ್ charset ಗೆ ರೀಕೊಡ್ ಮಾಡುತ್ತದೆ</td></tr>
</table>

© NIMI NOT TO BE REPUBLISHED

•	Recode iso-8859-15..utf8 file_to_change.txt	Latin9 (ವೆಸ್ಟರ್ನ್ ಯೂರೋಪ್) ದಿಂದ utf8 ಗೆ
•	Recode ../b64 < file.txt > file.b64	Base64 ಎನ್ಕೋಡ್
•	Recode /qp.. < file.qp > file.txt	ಕೋಟ್ ಮಾಡಿದ ಮುದ್ರಿಸಬಹುದಾದ ಡೀಕೊಡ್
•	Recode ..HTML < file.txt > file.html	ಟೆಕ್ಸ್ಟ್‌ದಿಂದ HTML ಗೆ
•	Recode -lf windows-1252 | grep euro	ಅಕ್ಷರಗಳ ಟೇಬಲ್‌ನ್ನು ನೋಡಿರಿ.
•	Echo -n 0x80 | recode latin-9/x1..dump	ಒಂದು ಕೊಡ್ ಲ್ಯಾಟಿನ್ -9 charmap ನಲ್ಲಿ ಏನನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Echo -n 0x20AC | recode ucs-2/x2.. latin-9/x	ಲ್ಯಾಟಿನ್ -9 ಎನ್ಕೋಡಿಂಗ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Echo -n 0x20AC | recode ucs-2/x2.. utf-8/x	Utf -8 ಎನ್ಕೋಡಿಂಗ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ.
CDs		
•	Gzip < /dev/cdrom > cdrom.iso.gz	ಡಾಟಾ cdrom ನ ಪ್ರತಿಯನ್ನು ಸೇವ್ ಮಾಡುತ್ತದೆ.
•	Mkisofs -V LABEL -r dir | gzip > cdrom.iso.gz	ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ವಿಷಯಗಳಿಂದ cdrom ನ ಇಮೇಜನ್ನು ರಚಿಸುತ್ತದೆ.
•	Mount -o loop cdrom.iso /mnt/dir	/mnt/ dir ನಲ್ಲಿ cdrom ನ ಇಮೇಜನ್ನು ಮೌಂಟ್ ಮಾಡುತ್ತದೆ.(read only)
•	Wodim dev=/dev/cdrom blank=fast	CDRW ಕ್ಲಿಯರ್ ಮಾಡುತ್ತದೆ.
•	Gzip -dc cdrom.iso.gz | wodim -tao Dev= /dev/cdrom -v -data -	cdrom ನ ಇಮೇಜನ್ನು ಬರ್ನ್ ಮಾಡುತ್ತದೆ.(dev ಯನ್ನು ಖಚಿತಪಡಿಸಲು --prcap ಬಳಸಿ)
•		
•	Cdparanoia -B	CD ದಿಂದ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ wav ಫೈಲ್‌ಗೆ ಆಡಿಯೋ ಟ್ರ್ಯಾಕ್ ಗಳನ್ನು Rip ಮಾಡುತ್ತದೆ.
•	Wodim -v dev=/dev/cdrom -audio -pad *.wav	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿನ ಎಲ್ಲ wav ಫೈಲ್‌ಗಳನ್ನು ಆಡೀಯೋ ಸಿಡಿ ಮಾಡುತ್ತದೆ.
•	Oggenc --tracknum=$track track.cdda.wav -o track.ogg	wav ಫೈಲ್‌ನಿಂದ ogg ಫೈಲನ್ನು ಮಾಡುತ್ತದೆ.
Disk space		
•	Ls -lSr	ಗಾತ್ರಗಳಿಗೆ ಅನುಗುಣವಾಗಿ ಸಣ್ಣ ಗಾತ್ರದ ಫೈಲ್‌ನಿಂದ ದೊಡ್ಡ ಗಾತ್ರದ ಫೈಲ್‌ವರೆಗೆ ತೋರಿಸುತ್ತದೆ.
•	Du -s * | sort -k1,1rn | head	ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಟಾಪ್ ಡಿಸ್ಕ್ ಬಳಕೆದಾರರನ್ನು ತೋರಿಸುತ್ತದೆ. Dutop ಅನ್ನು ಕೂಡ ನೋಡಿರಿ.
•	Du -hs /home/* | sort -k1,1h	ಡಿಸ್ಕ್ ಬಳಕೆಯನ್ನು ಅರ್ಥೈಸುವ ಸುಲಭವಾಗುವ ಆಧಾರದ ಮೇಲೆ ಪಾಥ್‌ಗಳನ್ನು ವಿಂಗಡಿಸುತ್ತದೆ.
•	Df -h	ಮೌಂಟ್ ಮಾಡಲಾದ ಫೈಲ್ ಸಿಸ್ಟಮ್ಗಳಲ್ಲಿನ ಖಾಲಿ ಸ್ಥಳವನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Df -i	ಮೌಂಟ್ ಮಾಡಲಾದ ಫೈಲ್ ಸಿಸ್ಟಮ್ಗಳಲ್ಲಿನ ಖಾಲಿ inode ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Fdisk -l	ಡಿಸ್ಕ್‌ಗಳ ಪಾರ್ಟಿಷನ್ ಗಾತ್ರಗಳನ್ನು ಮತ್ತು ವಿಧಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.(ರೂಟ್ ನಂತೆ ರನ್ ಆಗುತ್ತದೆ)
•	Rpm -q -a --qf`%10{SIZE}\t%{NAME}\n` | sort -k1, 1n	Rpm distros ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಆದ ಗಾತ್ರದ (ಬೈಟ್ ಗಳು) ಆಧಾರದ ಮೇಲೆ ಎಲ್ಲ ಪ್ಯಾಕೇಜ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

•	Dpkg-query -w -f=1${Installed-Size;10}\t${package}\n` | sort -k1,1n	deb distros ನಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಆದ ಗಾತ್ರದ (ಕಿಲೋಬೈಟ್ ಗಳು) ಆಧಾರದ ಮೇಲೆ ಎಲ್ಲ ಪ್ಯಾಕೇಜ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Dd bs=1 seek=2TB if=/dev/null of =ext3. test	ದೊಡ್ಡ ಟೆಸ್ಟ್ ಫೈಲ್‌ನ್ನು ರಚಿಸುತ್ತದೆ (ಸ್ಥಳವನ್ನು ತೆಗೆದುಕೊಳ್ಳದೆ)
•	> file	ಫೈಲ್‌ನ ಡಾಟಾವನ್ನು ಮೊಟಕುಗೊಳಿಸುತ್ತದೆ ಅಥವಾ ಖಾಲಿ ಫೈಲನ್ನು ರಚಿಸುತ್ತದೆ.
Monitoring /debugging		
•	Tail -f /var/log/messages	ಲಾಗ್ ಫೈಲ್‌ನಲ್ಲಿರುವ ಸಂದೇಶಗಳನ್ನು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡುತ್ತದೆ.
•	Strace -c ls > /dev/null	ಕಮಾಂಡ್‌ನಿಂದ ಮಾಡಲ್ಪಟ್ಟ ಸಿಸ್ಟಮ್ ಕಾಲ್‌ಗಳನ್ನು ಸಾರಾಂಶ ಮಾಡುತ್ತದೆ/ಸಂಕ್ಷಿಪ್ತ ಮಾಡುತ್ತದೆ.
•	Strace -f -e open ls > /dev/null	ಕಮಾಂಡ್‌ನಿಂದ ಮಾಡಲ್ಪಟ್ಟ ಸಿಸ್ಟಮ್ ಕಾಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Strace -f -e trace=write -e write=1,2 ls > /dev/null	Stdout ಮತ್ತು stderr ಗೆ ಏನು ಬರೆಯಲಾಗಿದೆ ಎಂಬುದನ್ನು ಮೇಲ್ವಿಚಾರಿಸುತ್ತದೆ.
•	Ltrace -f -e getenv ls > /dev/null	ಕಮಾಂಡ್‌ನಿಂದ ಮಾಡಲ್ಪಟ್ಟ ಲೈಬ್ರರಿ ಕಾಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Lsof -p $$	ಪ್ರೊಸೆಸ್ ಐಡಿ ತೆರೆದಿರುವ ಪಾಥ್‌ಗಳನ್ನು ಪಟ್ಟಿಮಾಡುತ್ತದೆ.
•	Lsof ~	Ssh ಹೊರೆತುಪಡಿಸಿ, ನೆಟ್‌ವರ್ಕ್ ಟ್ರಾಫಿಕ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ. tcpdump_not_me ಯನ್ನು ಕೂಡ ನೋಡಿರಿ.
•	Tcpdump not port 22	ಶ್ರೇಣಿ ವ್ಯವಸ್ಥೆಯಲ್ಲಿರುವ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ps -e -o pid, args --forest	ಶೇಕಡಾವಾರು ಸಿಪಿಯು ಬಳಕೆಯ ಅನುಗುಣವಾಗಿ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu | sed `/^ 0.0 /d`	ಮೆಮೊರಿ (KB) ಬಳಕೆಯ ಆಧಾರದ ಮೇಲೆ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ. ps_mem.py ಯನ್ನು ನೋಡಿರಿ.
•	Ps -e -orss=, args= | sort -b k1,1n | pr -TW$COLUMNS	ಒಂದು ನಿರ್ದಿಷ್ಟ ಪ್ರಕ್ರಿಯೆಯ ಸಲುವಾಗಿ ಎಲ್ಲ ಥ್ರೆಡ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ps -C firefox-bin -L -O pid,tid,pcpu,state	ನಿರ್ದಿಷ್ಟ ಪ್ರೊಸೆಸ್ ಐಡಿಗಳ ಸಲುವಾಗಿ ಕಳೆದ ಸಮಯವನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.
•	Ps -p 1, $$ -o etime=	ಬದಲಾಗುತ್ತಿರುವ ಪ್ರೊಸೆಸ್ ಸಬ್ ಟ್ರೀಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.
•	Watch -n.1 pstree -Uacp $$	ಸಿಸ್ಟಮ್ ರಿಬೂಟ್ ಹಿಸ್ಟರಿ(ಇತಿಹಾಸ) ವನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Last reboot	RAM ನ (ಉಳಿದ) ಪ್ರಮಾಣವನ್ನು ತೋರಿಸುತ್ತದೆ. (-m, MB ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ)
•	Free -m	ಬದಲಾಯಿಸಬಹುದಾದ ಡಾಟಾವನ್ನು ನಿರಂತರವಾಗಿ ವೀಕ್ಷಿಸಿ.
•	Watch -n.1 `cat /proc/interrupts	ಕಾನ್ಫಿಗರ್ ನಿಯಮಗಳಿಗೆ ಸಹಾಯ ಮಾಡಲು udev ಈವೆಂಟ್‌ಗಳನ್ನು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡುವುದು.
•	Udevadm monitor	Monitor udev events to help configure rules
System information		
•	Uname -a	ಕರ್ನಲ್ ಆವೃತ್ತಿಯನ್ನು ಮತ್ತು ಸಿಸ್ಟಮ್ ರಚನೆಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Head -n1/etc/issue	ವಿತರಣೆಯ ಹೆಸರು ಮತ್ತು ಆವೃತ್ತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Cat /proc/partitions	ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ನೋಂದಾಯಿಸಿದ ಎಲ್ಲ ಪಾರ್ಟಿಷನ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Grep MemTotal /proc/meminfo	ಸಿಸ್ಟಮ್‌ನಲ್ಲಿರುವ ಒಟ್ಟೂ RAM ಅನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Grep ``model name`` /proc/cpuinfo	ಸಿಪಿಯುನ(ಗಳ) ಮಾಹಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Lspci -tv	PCI ಮಾಹಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Lsusb -tv	USB ಮಾಹಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.
•	Mount | column -t	ಸಿಸ್ಟಮ್ ಮೌಂಟ್ ಮಾಡಲಾದ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ.(ಮತ್ತು ಔಟ್‌ಪುಟ್‌ನ್ನು ಜೋಡಿಸುತ್ತದೆ)
•	Grep -F capacity: /proc/acpi/battery/BATO/info	ಲ್ಯಾಪ್‌ಟಾಪ್ ಬ್ಯಾಟರಿಯಲ್ಲಿನ ಸೆಲ್‌ಗಳ ಸ್ಥಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

•	Dmidecode -q | less	SMBIOS/DMI ಮಾಹಿತಿಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.
•	Smartcti -A /dev/sda | grep Power_On_Hours	ಈ ಡಿಸ್ಕ್‌ಗೆ(ಸಿಸ್ಟಮ್) ಒಟ್ಟು ಎಲ್ಲಿಯವರೆಗೆ ಶಕ್ತಿ ನೀಡಲಾಗಿದೆ
•	Hdparm -i /dev/sda	ಡಿಸ್ಕ್ sda ಬಗ್ಗೆ ಮಾಹಿತಿ ತೋರಿಸುತ್ತದೆ.
•	Hdparm -tT /dev/sda	ಡಿಸ್ಕ್ sda ಮೇಲೆ ಸ್ಪೀಡ್ ರೀಡ್ ಟೆಸ್ಟ್ (ಓದುವ ವೇಗದ ಪರೀಕ್ಷೆ)ಯನ್ನು ಮಾಡುತ್ತದೆ.
•	badblocks -s /dev/sda	ಡಿಸ್ಕ್ sda ನಲ್ಲಿ ಓದಲಾಗದ ಬ್ಲಾಕ್‌ಗಳಿಗಾಗಿ ಪರೀಕ್ಷೆ ಮಾಡುತ್ತದೆ.
intercative		
•	readline	Bash, python, bc, gnuplot ಗಳಿಂದ ಉಪಯೋಗಿಸಲಾದ ಲೈನ್ ಎಡಿಟರ್
•	screen	ಬೇರ್ಪಡಿಸುವ ಸಾಮರ್ಥ್ಯಗಳೊಂದಿಗೆ ವರ್ಚುವಲ್ ಟರ್ಮಿನಲ್‌ಗಳು
•	mc	Rpm, tar, ftp, ssh ... ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡಬಲ್ಲ ಪ್ರಬಲ ಫೈಲ್ ಮ್ಯಾನೇಜರ್
•	gnuplot	ಪರಸ್ಪರ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವ/ ಸ್ಕ್ರಿಪ್ಟ್ ಮಾಡಬಹುದಾದ ಗ್ರಾಫಿಂಗ್
•	links	ವೆಬ್ ಬ್ರೌಸರ್
•	xdg-open	ನೋಂದಾಯಿತ ಡೆಸ್ಕ್‌ಟಾಪ್ ಅಪ್ಲಿಕೇಷನ್‌ನೊಂದಿಗೆ ಫೈಲ್ ಅಥವಾ ಯುಆರ್ ಎಲ್(url) ನ್ನು ತೆರೆಯುವುದು.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ವರ್ಡ ಪ್ರೊಸೆಸಿಂಗ್ ಸಾಫ್ಟವೇರ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು (Using Word Processing Software)

ಎಂಎಸ್ ವರ್ಡ್ 2010 ಥಿಯರಿ(ಸಿದ್ಧಾಂತ) (MS WORD 2010 THEORY)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- **ಎಮ್‌ಎಸ್ ಆಫೀಸ್ ಎಂದರೇನು ತಿಳಿಸಿ**
- **ಎಮ್‌ಎಸ್ ವರ್ಡ್ ಎಂದರೇನು ಹಾಗೂ ಆರಂಭಿಕ ಹಂತಗಳನ್ನು ಸಂಕ್ಷೇಪ್ತವಾಗಿ ತಿಳಿಸಿ**
- **ಎಮ್‌ಎಸ್ ವರ್ಡ್‌ನ ವಿವಿಧ ಸ್ಕ್ರೀನ್ ಬ್ಲಾಕ್‌ಗಳನ್ನು ವಿವರಿಸಿ**
- **ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ರಚಿಸಲು, ಸೇವ್ ಮಾಡಲು , ಮುದ್ರಿಸಲು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ವಿವರಿಸಿ.**

ಮೈಕ್ರೊಸಾಫ್ಟ ಆಫೀಸ್ ಇದು ಅಪ್ಲಿಕೇಷನ್ ಸಾಫ್ಟ್‌ವೇರ್ ಪ್ಯಾಕೇಜ್ ಆಗಿದೆ. ಇದನ್ನು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಕಾರ್ಪೋರೇಷನ್ ಪರಿಚಯಿಸಿದೆ. ಎಮ್‌ಎಸ್ ಆಫೀಸ್ ಈ ಕೆಳಗಿನ ಜನಪ್ರೀಯ ಪ್ಯಾಕೇಜ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. :

- ಮೈಕ್ರೊಸಾಫ್ಟ ವರ್ಡ
- ಮೈಕ್ರೊಸಾಫ್ಟ ಎಕ್ಸೆಲ್
- ಮೈಕ್ರೊಸಾಫ್ಟ ಪಾವರ್ ಪಾಯಿಂಟ್
- ಮೈಕ್ರೊಸಾಫ್ಟ ಎಕ್ಸೆಸ್
- ಮೈಕ್ರೊಸಾಫ್ಟ ಔಟಲುಕ್

ವರ್ಡ್ 2010 (Word 2010)

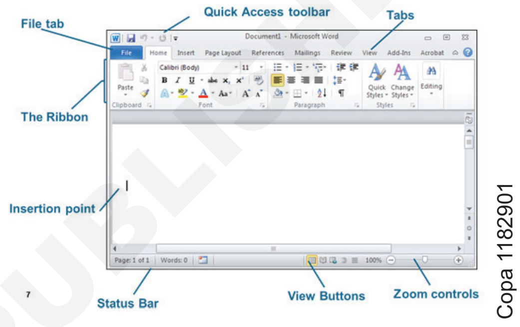
ಇದು ವರ್ಡ್ ಪ್ರೊಸೆಸರ್ ಪ್ಯಾಕೇಜ್ ಆಗಿದ್ದು, ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ರಚಿಸಲು ಮತ್ತು ಎಡಿಟ್ ಮಾಡಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ಇದು ಅತ್ಯಂತ ಪ್ರಸಿದ್ಧವಾದ ವರ್ಡ್ ಪ್ರೊಸೆಸರ್ ಆಗಿದೆ. ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು, ಮೆಮೊಗಳು, ಪತ್ರಗಳು, ವರದಿಗಳು, ಕರಪತ್ರಗಳು, ವ್ಯವಹಾರ ದಾಖಲೆಗಳು ಮತ್ತು ಇಂಟರ್ನೆಟ್ ವೆಬ್‌ಪುಟಗಳನ್ನು ಸಹ ರಚಿಸಲು ಮತ್ತು ಫಾರ್ಮಾಟ್ ಮಾಡಲು ಸಾಧನಗಳನ್ನು ಒದಗಿಸಿ ದಾಖಲೆಗಳನ್ನು ವೃತ್ತಿಪರವಾಗಿ ಕಾಣುವಂತೆ ಮಾಡುತ್ತದೆ.

ವರ್ಡನ್ನು ಪ್ರಾರಂಭಿಸುವುದು (Starting Word)

ಈ ಕೆಳಗಿನಂತೆ ಮಾಡುವುದರಿಂದ ಎಂ ಎಸ್ ವರ್ಡ್‌ನ್ನು ಪ್ರಾರಂಬಿಸಬಹುದು.

Start > (Windows Logo) All Programs > Microsoft Office > Microsoft Word. ಎಮ್ ಎಸ್ ವರ್ಡ್ ಡೀಫಾಲ್ಟ್ ಸೆಟ್ಟಿಂಗ್ ನ ಪೇಜ ಲೇಔಟ್‌ನೊಂದಿಗೆ ಖಾಲಿ ಡಾಕ್ಯುಮೆಂಟ್ ತೆರೆದುಕೊಳ್ಳುತ್ತದೆ.

Fig 1



ಚಿತ್ರ 1- ಡೀಫಾಲ್ಟ ಪೇಜ ಲೇಔಟ್ (Fig 1 - Default page layout)

ಎಮ್ ಎಸ್ ಆಫೀಸ್ 2010, ಎಲ್ಲ ಘಟಕಗಳಿಗೂ ಸಾಮಾನ್ಯ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿದೆ. ಇದು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳಲು ಸುಲಭವಾಗಿದೆ. ಎಮ್ ಎಸ್ ಆಫೀಸ್‌ನ ಎಲ್ಲ ಪ್ಯಾಕೇಜ್‌ಗಳಾದ ವರ್ಡ್, ಎಕ್ಸೆಲ್, ಪಾವರ್ ಪಾಯಿಂಟ್, ಎಕ್ಸೆಸ್ ಮುಂತಾದವುಗಳಲ್ಲಿ ಸಾಮಾನ್ಯ ಉಪಯುಕ್ತತೆಗಳು ಲಭ್ಯವಿರುತ್ತವೆ. ಬೇರೆ ಪ್ಯಾಕೇಜ್‌ನ ಬಳಕೆದಾರರಿಗೂ ಉಪಯೋಗಿಸಲು ಸುಲಭವಾಗುವಂತೆ ಇದರ ಅನೇಕ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಮರು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ವರ್ಡ 2010 ನ ಉಪಯುಕ್ತತೆಗಳನ್ನು ಕೆಳಗೆ ವಿವರಿಸಲಾಗಿದೆ.

ವರ್ಡ್‌ನ ರಿಬ್ಬನ್ ಟ್ಯಾಬ್ ಎಂಟು ಮುಖ್ಯ ಭಾಗಗಳನ್ನು ಹೊಂದಿದೆ. ಅವು ಯಾವವು ಎಂದರೆ, ಫೈಲ್ , ಹೋಮ್, ಇನ್ಸರ್ಟ್, ಪೇಜ್ ಲೇಔಟ್, ರೆಫರನ್ಸ್, ಮೇಲಿಂಗ್ಸ್ , ರಿವಿವ್, ಮತ್ತು ವಿವ್.

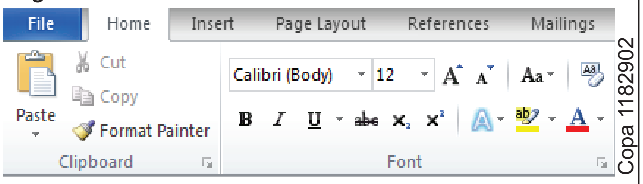
ರಿಬ್ಬನ್‌ನ ಫೈಲ್ ಟ್ಯಾಬ್ ಹೊಸ ಫೈಲ್‌ನ್ನು ರಚಿಸಲು ಉಪಯುಕ್ತವಾಗಿದೆ. ಇದು ನಿಮಗೆ ಅವಶ್ಯವಿರುವಂತೆ ಖಾಲಿ ಅಥವಾ ಟೆಂಪ್ಲೇಟ್ ಪುಟವನ್ನು ರಚಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. Ctrl + N ಯಾವಾಗಲೂ ಆಫೀಸ್ ಪ್ಯಾಕೇಜನಲ್ಲಿ ಹೊಸ ಖಾಲಿ ಡಾಕ್ಯುಮೆಂಟನ್ನು ರಚಿಸುತ್ತದೆ. ಈ ರೀತಿ ರಚಿಸಲಾದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು ಡೀಫಾಲ್ಟ ವರ್ಡ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಾಗಿ ಸಂಗ್ರಹಿಸಲ್ಪಡುತ್ತವೆ ಅಥವಾ ಇವುಗಳಿಗೆ ಕಂಪ್ಯಾಟಿಬಲ್ ಸ್ವರೂಪವನ್ನು ನೀಡಲಾಗುತ್ತದೆ. ಇದಲ್ಲದೆ, ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತೆರೆಯಲು ಅನುಮತಿ ಪಡೆಯುವುದನ್ನು

© NIMI NOT TO BE REPUBLISHED

ನಿರ್ಧರಿಸುವುದು, ನೆಟ್‌ವರ್ಕ್ ನಲ್ಲಿ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಹಂಚಿಕೊಳ್ಳುವುದು, ಹಿಂದಿನ ಮತ್ತು ಬೇರೆ ಆವೃತ್ತಿಯೊಂದಿಗೆ ಡಾಕ್ಯುಮೆಂಟನ್ನು ಹೊಂದಾಣಿಕೆಯಾಗುವಂತೆ ನಿರ್ವಹಿಸುವುದು ಮಾಡುತ್ತದೆ. ಪ್ರಿಂಟ್ ಆಯ್ಕೆಯು, ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಹಾರ್ಡ ಕಾಪಿಯನ್ನು ತೆಗೆದುಕೊಳ್ಳಲು ಅಥವಾ ಸ್ಥಾಪಿಸಲಾದ ಪ್ರಿಂಟರ್‌ನ ವೈಶಿಷ್ಟ್ಯದಂತೆ ಪಿಡಿಎಫ್ ಸ್ವರೂಪದಲ್ಲಿ ಮುದ್ರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಮತ್ತೆ ಕೆಲಸ ಮಾಡಲು ಸುಲಭವಾಗುವಂತೆ ಇತ್ತೀಚೆಗೆ ತೆರೆದ ಫೈಲ್‌ಗಳನ್ನು ಸಹ ವೀಕ್ಷಿಸಬಹುದು. ಇದೇ ಟ್ಯಾಬ್ ನಲ್ಲಿ ಹೆಲ್ಪ್ ಕೂಡ ಇದೆ.

ಚಿತ್ರ 2 ಹೋಮ್ ಟ್ಯಾಬ್ (Fig 2 - Home Tab)

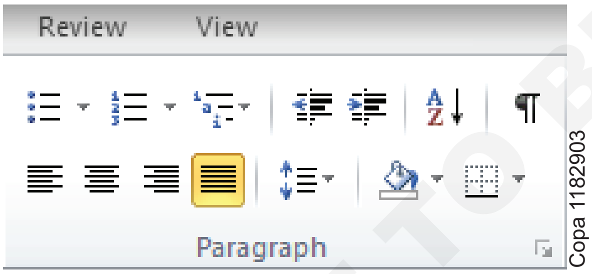
Fig 2



ಹೋಮ್ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಕಟ್, ಕಾಪಿ, ಪೇಸ್ಟ್, ಮತ್ತು ಪೇಸ್ಟ್ ಸ್ಪೇಷಲ್‌ನಂತಹ ಕ್ಲಿಪ್ ಬೋರ್ಡ್ ಆಯ್ಕೆಗಳಿವೆ. ಫಾಂಟ್ ಬ್ಲಾಕ್‌ನಲ್ಲಿ ಫಾಂಟ್ಸ್, ಸೈಝ್, ಸೂಪರ್ ಸ್ಕ್ರಿಪ್ಟ್, ಸಬ್ ಸ್ಕ್ರಿಪ್ಟ್, ಬೋಲ್ಡ್, ಇಟಾಲಿಕ್, ಅಂಡರ್ಲೈನ್, ಸ್ಟ್ರೈಕ್ ಥ್ರೂ, ಫಾಂಟ್ ಕಲರ್, ಮುಂತಾದ ಆಯ್ಕೆಗಳಿವೆ.

ಪ್ಯಾರಾಗ್ರಾಫ್ ಗ್ರೂಪ್(Paragraph Group) (**ಚಿತ್ರ** 3)

Fig 3

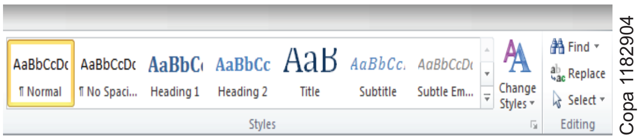


ಪ್ಯಾರಾಗ್ರಾಫ್ ಗ್ರೂಪ್ , ಪ್ಯಾರಾಗ್ರಾಫನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಅಲೈನ್‌ಮೆಂಟ್ ಲೆಫ್ಟ್, ರೈಟ್, ಸೆಂಟರ್, ಜಸ್ಟಿಫೈ , ಮತ್ತು ಇಂಡೆಂಟೇಷನ್ , ಪ್ಯಾರಾ ಮತ್ತು ಲೈನ್ ಸ್ಪೇಸಿಂಗ್‌ಗಳನ್ನು ಅನುಮತಿಸುತ್ತದೆ. ಲೈನ್ ಮತ್ತು ಪೇಜ ಬ್ರೇಕ್‌ಗಳು ಪೇಜಿನೇಷನ್ ಆಯ್ಕೆಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಎಕ್ಸೆಪ್ಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಅನುಮತಿಸುತ್ತವೆ.

ಸ್ಟೈಲ್ಸ್ ಮತ್ತು ಎಡಿಟಿಂಗ್ ಗ್ರೂಪ್ (Styles and Editing Group) (**ಚಿತ್ರ** 4)

ಸ್ಟೈಲ್ಸ್ ಗ್ರೂಪ್ ಹೆಡಿಂಗ್ ಸ್ಟೈಲ್ , ಪ್ಯಾರಾಗ್ರಾಫ್ ಸ್ಟೈಲ್ , ಸಬ್ ಟೈಟಲ್ ಮುಂತಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು

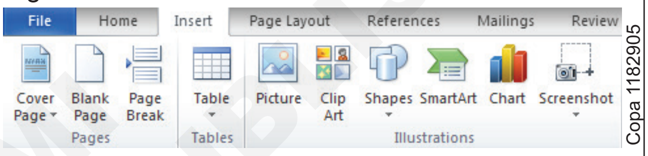
Fig 4



ರಚಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಕಸ್ಟಮ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು ವಿವಿಧ ಸ್ಪೆಸಿಫಿಕೇಶನ್‌ಗಳೊಂದಿಗೆ ಸಂಗ್ರಹಿಸಬಹುದು. ಎಡಿಟಿಂಗ್ ಗ್ರೂಪ್‌ನ್ನು ನಿರ್ದಿಷ್ಟ ಪಠ್ಯದ ಭಾಗವನ್ನು ಆಯ್ಕೆ ಮಾಡಲು, ಪಠ್ಯದ ನಿರ್ದಿಷ್ಟ ಭಾಗದಲ್ಲಿ ಪಠ್ಯವನ್ನು ಹುಡುಕುವುದು ಮತ್ತು ಬದಲಾಯಿಸುವುದಕ್ಕಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಇನ್ಸರ್ಟ್ ಟ್ಯಾಬ್ ,7ಗುಂಪುಗಳನ್ನು ಹೊಂದಿದೆ. ಅವು ಯಾವವೆಂದರೆ ಪೇಜಸ್, ಟೇಬಲ್ಸ್ , ಮತ್ತು ಇಲ್ಲಸ್ಟ್ರೇಷನ್ಸ್ ಬ್ಲಾಕ್‌ಗಳು(Pages , Tables, and Illustrations blocks) (ಚಿತ್ರ 5)

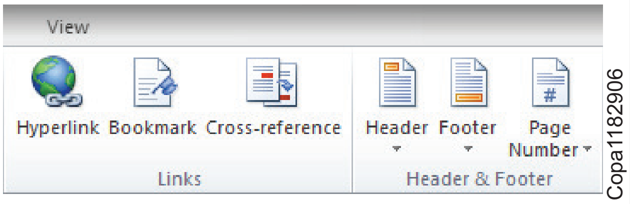
Fig 5



ಪೇಜ್ ಗ್ರೂಪ್ ಕವರ್ ಪುಟಗಳನ್ನು, ಖಾಲಿ ಪುಟಗಳನ್ನು ಅಳವಡಿಸುವುದು, ಮತ್ತು ಪುಟವನ್ನು ವಿಭಜಿಸುವುದನ್ನು ಮಾಡುತ್ತದೆ. ಟೇಬಲ್ ಗ್ರೂಪ್ , ಒಂದು ಟೆಕ್ಸ್ಟ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸಿದ್ಧಪಡಿಸಿದ ಅಡ್ಡ ಸಾಲು - ಉದ್ದಸಾಲುಗಳನ್ನು ಹೊಂದಿದ ಟೇಬಲನ್ನು ಅಳವಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇಲ್ಲಸ್ಟ್ರೇಷನ್ಸ್ ಗುಂಪು ಬಾಹ್ಯ ಮೂಲಗಳಿಂದ ಚಿತ್ರಗಳನ್ನು ಅಳವಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು ಕ್ಲಿಪ್ ಆರ್ಟ್‌ಗಳು, ಶೇಪ್‌ಗಳು, ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್‌ಗಳು, ಚಾರ್ಟ್‌ಗಳು, ಮತ್ತು ಸ್ಕ್ರೀನ್ ಶಾಟ್‌ಗಳನ್ನು ಕೂಡ ಅಳವಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಲಿಂಕ್‌ಗಳು ಮತ್ತು ಹೆಡರ್/ಫೂಟರ್ ಗ್ರೂಪ್ (Links and Header/ Footer Groups) (**ಚಿತ್ರ** 6)

Fig 6



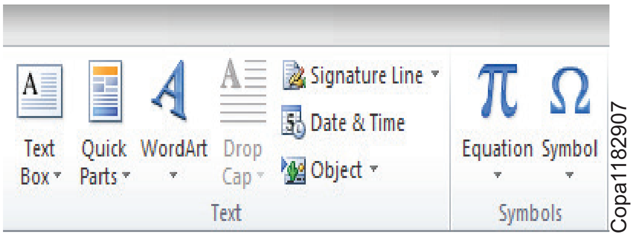
ಲಿಂಕ್ಸ್ ಗ್ರೂಪ್ ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಹೈಪರ್ ಲಿಂಕ್‌ನ್ನು ,ಬುಕ್ ಮಾರ್ಕನ್ನು ಮತ್ತು ಕ್ರಾಸ್ ರೆಫರನ್ಸ್‌ನ್ನು ಸೇರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಹೆಡರ್/ಫೂಟರ್ ಗ್ರೂಪ್ ಹೆಡರ್, ಫೂಟರ್‌ನ್ನು ಪ್ರತಿ ಪುಟದಲ್ಲಿ ಕಾಣಿಸುವಂತೆ ಅಳವಡಿಸುತ್ತದೆ. ಮತ್ತು ನಿರ್ದಿಷ್ಟ ಪ್ರದೇಶದಲ್ಲಿ ಪುಟ ಸಂಖ್ಯೆ ಕಾಣಿಸುವಂತೆ ಮಾಡುತ್ತದೆ.

ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಸಿಂಬಾಲ್ಸ್ ಬ್ಲಾಕ್ (Text and Symbols block) (**ಚಿತ್ರ** 7)

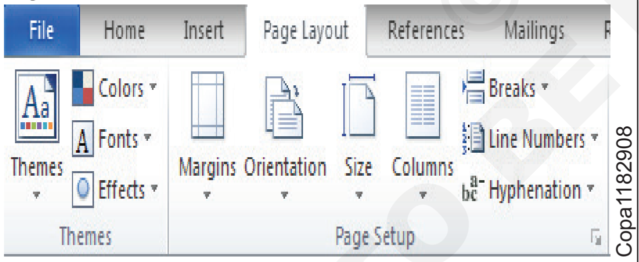
Fig 7



ಟೆಕ್ಸ್ಟ್ ಗ್ರೂಪ್ , ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ನ್ನು ರಚಿಸಲು ಅನುಮತಿಸುತ್ತದೆ, ಕ್ವಿಕ್ ಪಾರ್ಟ್‌ಗಳು ಬ್ರೋಚರ್‌ಗಳನ್ನು ತಯಾರಿಸುತ್ತದೆ, ವರ್ಡ್ ಆರ್ಟ್ ಟೆಕ್ಸ್ ನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸುತ್ತದೆ, ಪ್ಯಾರಾಗ್ರಾಫ್ ಸ್ಟೈಲ್‌ಗಳು, ಸಿಗ್ನೇಚರ್ ಲೈನ್‌ಗಳು, ದಿನಾಂಕ ಮತ್ತು ಸಮಯ ಮತ್ತು ಆಬ್ಜೆಕ್ಟ ಮುಂತಾದವುಗಳನ್ನು ಅಳವಡಿಸುತ್ತದೆ. ಸಿಂಬಾಲ್ ಗ್ರೂಪ್ , ಗಣೀತ ಸಮೀಕರಣಗಳ ಅಥವಾ ಕರೆನ್ಸೀ ಚಿನ್ಹೆಗಳು, ಗಣೀತದ ಚಿನ್ಹೆಗಳು ಮುಂತಾದವುಗಳ ಚಿನ್ಹೆಗಳನ್ನು ಅಳವಡಿಸುತ್ತದೆ.

ಥೀಮ್ ಗಳು ಮತ್ತು ಪೇಜ್ ಸೆಟಪ್ ಗ್ರೂಪ್ (Themes and Page Setup Group) (**ಚಿತ್ರ** 8)

Fig 8

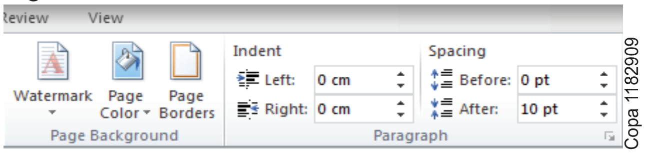


ಪೇಜ್ ಲೇಔಟ್ ಟ್ಯಾಬ್ ಐದು ಮುಖ್ಯ ಗುಂಪುಗಳನ್ನು ಹೊಂದಿದೆ. ಥೀಮ್ ಬ್ಲಾಕ್ ಥೀಮ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಪೂರ್ವನಿರ್ಧರಿತ ಟೆಂಪ್ಲೇಟ್ ಸೆಟಪ್‌ಗಳನ್ನು ರಚಿಸುತ್ತದೆ. ಕಸ್ಟಮೈಸ್ ಮಾಡಿದ ಹೊಸ ಥೀಮ್‌ಗಳನ್ನು ಕೂಡ ರಚಿಸಬಹುದು ಮತ್ತು ಭವಿಷ್ಯದ ಬಳಕೆಗಾಗಿ ಸೇವ್ ಮಾಡಿ ಉಪಯೋಗಿಸಬಹುದು.

ಪೇಜ ಸೆಟಪ್ ಗ್ರೂಪ್ , ಪುಟದ ಮಾರ್ಜಿನ್, ಪೇಜ್ ಒರಿಯಂಟೆಷನ್ ಎಂದರೆ ಲಂಬ ಅಥವಾ ಅಡ್ಡ , ಪೇಪರ್ ಸೈಝ್, ಕಾಲಮ್ಸ್ ಟು ಡಿಸ್ಪ್ಲೇ , ಬ್ರೇಕ್ಸ್ ಲೈ ನ್ ನಂಬರ್ಸ್ , ಮತ್ತು ಹೈಫನೇಷನ್ ಈ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ಪೇಜ್ ಬ್ಯಾಕ್ರೌಂಡ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್ ಗ್ರೂಪ್ (Page Back-ground and Paragraph Group) (**ಚಿತ್ರ** 9)

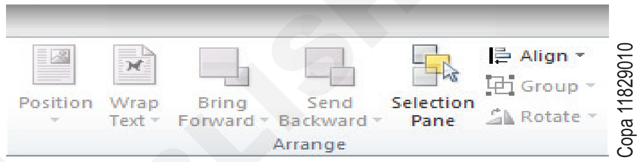
Fig 9



ಪೇಜ್ ಬ್ಯಾಕ್‌ಗ್ರೌಂಡ್ ಗ್ರೂಪ್ , ವಾಟರ್ ಮಾರ್ಕ್ , ಪೇಜ್ ಬ್ಯಾಕ್‌ಗ್ರೌಂಡ್ ಗ್ರೂಪ್ ಕಲರ್, ಮತ್ತು ಪೇಜ್ ಬಾರ್ಡರ್‌ಗಳನ್ನು ರಚಿಸುತ್ತದೆ. ಪ್ಯಾರಾಗ್ರಾಫ್ ಬ್ಲಾಕ್, ನಾವು ಈಗಾಗಲೇ ಚರ್ಚಿಸಿದ Home -> Paragraph group ನಂತೆ ನಿರ್ದಿಷ್ಟ ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿದೆ.

ಅರೇಂಜ್ ಗ್ರೂಪ್ (Arrange Group) (**ಚಿತ್ರ** 10)

Fig 10

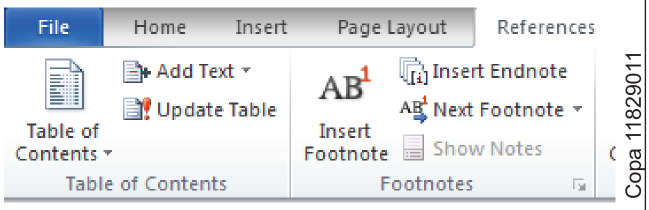


ಅರೇಂಜ ಗ್ರೂಪ್ ವಸ್ತುವಿನ(object) ಸ್ಥಾನ , ಟೆಕ್ಸ್ಟ್ಅಂಶಗಳು, ಟೆಕ್ಸ್ಟ್ ರ್‍ಯಾಪಿಂಗ್ , ವಸ್ತುಗಳನ್ನು ಹೊಂದಿಸುವುದು, ವಸ್ತುಗಳನ್ನು ಜೋಡಿಸುವುದು, ಗ್ರೂಪಿಂಗ್ ಮತ್ತು ಟ್ರಾನ್ಸಫಾರ್ಮೇಷನ್‌ಗಳನ್ನು ರಚಿಸುತ್ತದೆ.

ರೆಫರನ್ಸ್ ಟ್ಯಾಬ್ ಆರು ಬ್ಲಾಕ್ ಗಳನ್ನು ಹೊಂದಿದೆ.

ಟಿಓಸಿ ಮತ್ತು ಫೂಟ್‌ನೇಟ್ ಗ್ರೂಪ್ (TOC and Footnate Group) (**ಚಿತ್ರ** 11)

Fig 11

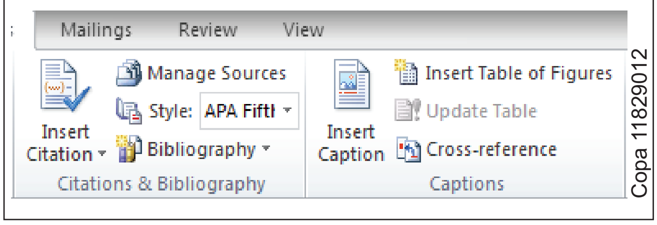


ಟೇಬಲ್ ಆಫ್ ಕಂಟೆಂಟ್ಸ್ ಒಂದು ನಿರ್ದಿಷ್ಟವಾದ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಟಿಓಸಿ ಯನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ರಚಿಸುತ್ತದೆ ಮತ್ತು ಆನಂತರ ಮಾರ್ಪಾಡುಗಳನ್ನು ನವೀಕರಿಸಬಹುದಾಗಿದೆ. ಫೂಟ್‌ನೋಟ್ಸ್ ಟ್ಯಾಬ್ , ಒಂದು ಪುಟದ ಫೂಟ್‌ನೋಟ್ಸ್ ನ್ನು ರಚಿಸುತ್ತದೆ. ಇದು ವಿವರಣಾತ್ಮಕ ನುಡಿಗಟ್ಟುಗಳನ್ನು ಅಥವಾ ಉಲ್ಲೇಖಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಬ್ಲಾಕ್ ಏರಿಯಾವನ್ನು ರಚಿಸುತ್ತದೆ. ಎಂಡ್ ನೋಟ್ , ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಕೊನೆಯಲ್ಲಿ ಟಿಪ್ಪಣಿಯನ್ನು ಮಾಡುತ್ತದೆ.

ಸೈಟೇಷನ್ಸ್ , ಬಿಬಿಲೋಗ್ರಾಫಿ ಮತ್ತು ಕ್ಯಾಪ್ಷನ್ಸ್ ಗ್ರೂಪ್ (Citations, Bibilography, and Captions Group) (**ಚಿತ್ರ** 12)

© NIMI NOT TO BE REPUBLISHED

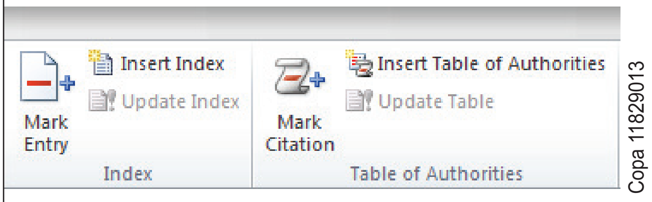
Fig 12



ಸೈಟೇಷನ್ಸ್ ಮತ್ತು ಬಿಬಿಲೋಗ್ರಾಫಿ ಇದು ಸೈಟೇಷನ್ , ರೆಫರನ್ಸ್ ಟೇಬಲ್‌ಗಳು ಮತ್ತು ಪ್ರಕಟಣೆಗೆ ಸಂಬಂಧಿತ ಲೇಖಕರ ಸಾಧನೆಯ ಫಲ (credit) ಇವುಗಳನ್ನು ಬಾಹ್ಯ ಸಂಪನ್ಮೂಲಗಳನ್ನಾಗಿ ಮಾಡುತ್ತದೆ. ಕ್ಯಾಪ್ಷನ್ಸ್ ಟ್ಯಾಬ್ ಕ್ಯಾಪ್ಷನ್ , ಅಂಕಿಗಳ ಟೇಬಲ್ , ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿನ ಕ್ರಾಸ್ ರೆಫರನ್ಸ್ ಇವುಗಳನ್ನು ಅಳವಡಿಸುತ್ತದೆ.

ಇಂಡೆಕ್ಸ್ ಮತ್ತು ಟೇಬಲ್ ಆಫ್ ಅಥಾರಿಟೀಸ್ ಗ್ರೂಪ್ (Index and Table of Authorities Group) (ಚಿತ್ರ 13)

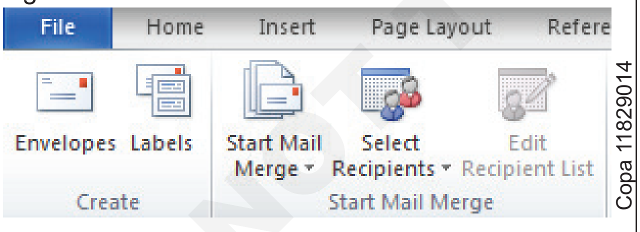
Fig 13



ಇಂಡೆಕ್ಸ್ ಇದು ಮುಖ್ಯ ವಿಷಯ ಅಥವಾ ಉಪ ವಿಷಯದ ಮೇಲೆ ಮೌಲ್ಯವನ್ನು ಸೂಚಿಸುತ್ತದೆ ಮತ್ತು ಅದಕ್ಕೆ ಅನುಗುಣವಾಗಿ ಇಂಡೆಕ್ಸ್‌ನ್ನು ರಚಿಸುತ್ತದೆ. ಟೇಬಲ್ ಆಫ್ ಅಥಾರಿಟೀಸ್ ಇದು ಸಿಟೇಷನ್ ಮಾರ್ಕನ್ನು ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಇದು ಕೇಸ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು , ಸ್ಟೇಟಸ್ ಮತು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿರುವ ಇತರ ಅಥಾರಿಟಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಕ್ರಿಯೇಟ್ ಮತ್ತು ಮೇಲ್ ಮರ್ಜ್ ಗ್ರೂಪ್ (Create and Mailmerge Groups) (ಚಿತ್ರ 14)

Fig 14

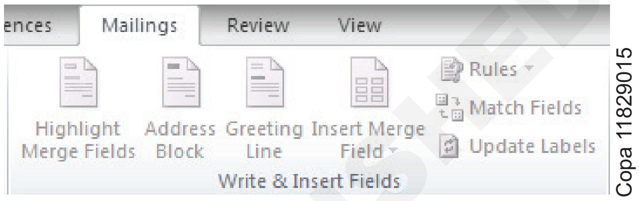


ಮೇಲಿಂಗ್ಸ್ ಟ್ಯಾಬ್‌ನ್ನು ಸಾಮಾನ್ಯ ಮೇಲಿಂಗ್ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಪತ್ರ ಅಥವಾ ಆದೇಶಗಳನ್ನು ರವಾನೆ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಕ್ರಿಯೇಟ್ ಬ್ಲಾಕ್ , ಪೂರ್ವನಿರ್ಧರಿತ ಫಾರ್ಮ್ಯಾಟಗಳನ್ನು ಮತ್ತು ಕಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಎನ್ವಲಪ್‌ಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ. ಉಲ್ಲೇಖಿತ ವಿತರಣಾ ವಿಳಾಸದ ಲೇಬಲ್‌ಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಬಹುದು.

ಮೇಲ್ ಮರ್ಜ್ ಬ್ಲಾಕ್‌ನ್ನು , ಒಂದು ಪತ್ರ , ಇ ಮೇಲ್ ಇವುಗಳ ಮೇಲ್ ಮರ್ಜ್ ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಸಾಮಾನ್ಯ ಪತ್ರವನ್ನು ಅನೇಕ ಜನರಿಗೆ ಕಳುಹಿಸಬೇಕಾದರೆ ಇದನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಈ ಆಯ್ಕೆಯು ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್‌ನ್ನು ರಚಿಸುತ್ತದೆ. ಈ ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್‌ನಲ್ಲಿ ಪತ್ರ ಸ್ವೀಕರಿಸಬೇಕಾದ ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಮಾಹಿತಿ ಇರುತ್ತದೆ. ಆಯ್ದ ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಅಥವಾ ಹೊಸ ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ತಯಾರಿಸಬಹುದು.

ರೈಟ್ ಮತ್ತು ಇನ್ಸರ್ಟ್ ಫೀಲ್ಡ್ಸ್ ಗ್ರೂಪ್ (Write and Insert fields Group) (ಚಿತ್ರ 15)

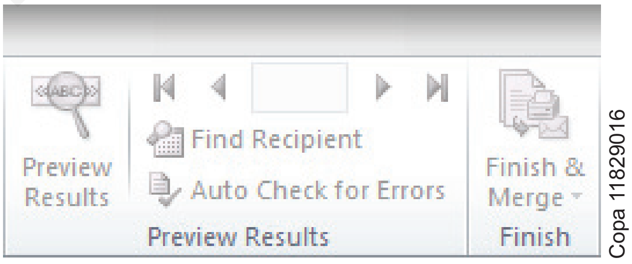
Fig 15



ಪತ್ರ ಬರೆಯುವಾಗ, ಅದು ಅನೇಕ ಭಾಗಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್‌ಗಳ ಫೀಲ್ಡನ್ನು, ಗ್ರೀಟಿಂಗ್ ಲೈನನ್ನು, ಕಂಟೆಂಟ್‌ನ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು , ಲೇಬಲ್‌ಗಳು ಮುಂತಾದ ಭಾಗಗಳನ್ನು ರಚಿಸಬಹುದು.

ಪ್ರಿವಿವ್ ರಿಸಲ್ಟ್ಸ್ ಮತ್ತು ಫಿನಿಶ್ ಗ್ರೂಪ್ (Preview Results and Finish Groups) (ಚಿತ್ರ 16)

Fig 16



ಮೇಲ್ ಮರ್ಜ್ ಕೆಲಸವನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಪೂರ್ಣಗೊಳಿಸಿದ ನಂತರ ಅದನ್ನು ಪೂರ್ವವೀಕ್ಷಣೆ ಮಾಡಬಹುದು. ಮತ್ತು ಪ್ರತಿ ರೆಸಿಪಿಯಂಟ್‌ನಲ್ಲಿ ದೋಷವಿದ್ದರೆ ಪರಿಶೀಲಿಸಿ ಅದನ್ನು ಸರಿಯಾಗಿ ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು. ವ್ಯಾಲಿಡೇಷನ್ ಮಾಡಿದ ನಂತರ ಮೇಲ್ ಮರ್ಜ್ ಪೂರ್ಣಗೊಳ್ಳುತ್ತದೆ.

ಪ್ರೂಫಿಂಗ್ ಮತ್ತು ಲ್ಯಾಂಗ್ವೇಜ್ ಗ್ರೂಪ್ಸ್ (Proofing and Language Groups) (ಚಿತ್ರ 17)

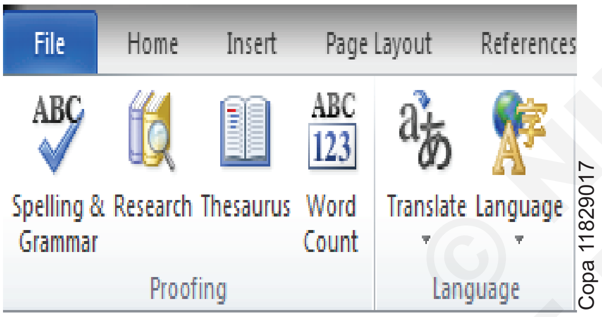
ಇಲ್ಲಿ ರಿವಿವ್ ಟ್ಯಾಬ್‌ನ್ನು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿವಿಧ ಪ್ರದೇಶಗಳನ್ನು ಪರಿಶೀಲಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪ್ರೂಫಿಂಗ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಕಾಗುಣಿತ ಮತ್ತು ವ್ಯಾಕರಣ (Spelling and

© NIMI NOT TO BE REPUBLISHED

Grammer) ವನ್ನು ಪರೀಕ್ಷಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಆಕಸ್ಮಿಕ ದೋಷಗಳನ್ನು ಇದು ಸರಿಪಡಿಸುತ್ತದೆ. ರಿಸರ್ಚ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಒಳಗಡೆ ಮಾತ್ರವಲ್ಲದೇ ಅದಕ್ಕೆ ಸಂಬಂಧಪಟ್ಟ ಇತರ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಕೂಡ ರಿಫೈನ್ ಮಾಡುತ್ತದೆ. ಥೆಸಾರಸ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಸಂಪೂರ್ಣತೆಯನ್ನು ಗುರುತಿಸುತ್ತದೆ. ಪ್ರಮಾಣಿತ ಭಾಷಾ ಹೊಂದಾಣಿಕೆಯನ್ನು ಬಳಸಿ ಪ್ರಾದೇಶಿಕ ಇಂಗ್ಲಿಷ್ ಯುಕೆ ಅಥವಾ ಇಂಗ್ಲಿಷ್ ಯುಎಸ್‌ಎ ಮುಂತಾದವುಗಳಿಂದ ಇದನ್ನು ಗುರುತಿಸಬಹುದು. ವರ್ಡ್ ಕೌಂಟ್ ಇದು ಡಾಕ್ಯುಮೆಂಟ್ ನಲ್ಲಿರುವ ಒಟ್ಟೂ ಶಬ್ದಗಳು, ಒಟ್ಟೂ ಅಕ್ಷರಗಳು, ಖಾಲಿ ಸ್ಥಳದೊಂದಿಗೆ ಮತ್ತು ಖಾಲಿ ಸ್ಥಳವಿಲ್ಲದೆ, ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳ ಸಂಖ್ಯೆ , ಸಾಲುಗಳ ಸಂಖ್ಯೆ ಇತ್ಯಾದಿಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. ಇಂತಹ ವಿಶ್ಲೇಷಣೆಗಳಿಗಾಗಿ ಟೆಕ್ಸ್ಟ್ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು.

ಲ್ಯಾಂಗ್ವೇಜ್ ಬ್ಲಾಕ್ , ಒಂದು ಪುಟವನ್ನು ಇನ್ಸ್ಟಾಲ್ ಆದಂತಹ ಇತರ ಭಾಷೆಗಳಿಗೆ ಅನುವಾದಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಮತ್ತು ಪ್ರೂಫಿಂಗ್ ಭಾಷೆಗೆ ಬದಲಾಯಿಸುತ್ತದೆ. ಇದಕ್ಕಾಗಿ ಮೈಕ್ರೊಸಾಫ್ಟ ಆನ್‌ಲೈನ್‌ನಿಂದ ನೇರ ಅನುವಾದಕ ಸೇವೆಯ ಅಗತ್ಯವಿದೆ.

Fig 17

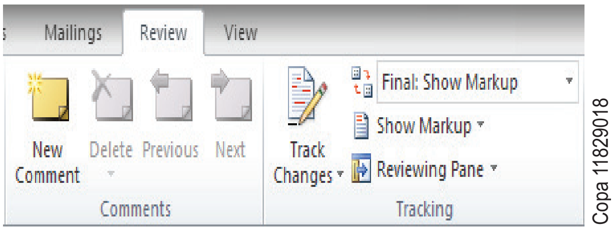


ಕಮೆಂಟ್ಸ್ ಮತ್ತು ಟ್ರ್ಯಾಕಿಂಗ್ ಗ್ರೂಪ್(comments & Tracking Group) (ಚಿತ್ರ 18)

ಕಮೆಂಟ್ಸ್ ಗ್ರೂಪ್ ಒಂದು ನಿರ್ದಿಷ್ಟವಾದ ಪ್ಯಾರಾಗ್ರಾಫ್ ಅಥವಾ ಟೆಕ್ಸ್ಟ್ ಬ್ಲಾಕ್‌ಗೆ ಪ್ರತಿಕ್ರಿಯೆ(ಕಮೆಂಟ್) ಅನ್ನು ಸೇರಿಸುತ್ತದೆ.

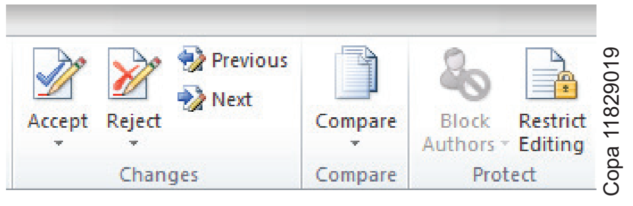
ಟ್ರ್ಯಾಕಿಂಗ್ ಗ್ರೂಪ್ ಸುರಕ್ಷಿತ ಮೋಡ್‌ನಲ್ಲಿರುವ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಇತರ ಲೇಖಕರು ಮಾಡಿದ ಬದಲಾವಣೆಗಳನ್ನು ಕಂಡು ಹಿಡಿಯುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್ X ಎನ್ನುವ ಬಳಕೆದಾರನಿಂದ ರಚಿಸಲಾಗಿದೆ ಮತ್ತು Y ಬಳಕೆದಾರನಿಂದ ಎಡಿಟ್ ಮಾಡಲಾಗಿದೆ, ಇದನ್ನು ಟ್ರ್ಯಾಕ್ ಗುರುತುಗಳೊಂದಿಗೆ ಪ್ರತ್ಯೇಕವಾಗಿ ಟ್ರ್ಯಾಕ್ ಮಾಡಲಾಗುತ್ತದೆ.

Fig 18



ಚೇಂಜಸ್, ಕಂಪೇರ್ ಮತ್ತು ಪ್ರೊಟೆಕ್ಟ್ ಗ್ರೂಪ್ಸ್ (Changes , Compare, and Protect Groups) (ಚಿತ್ರ 19)

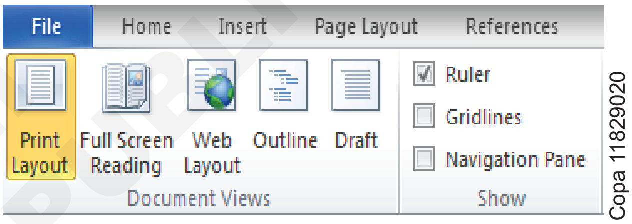
Fig 19



ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಮಾಡಿದ ಬದಲಾವಣೆಗಳು ಸ್ವೀಕೃತ ಆಗಬಹುದು ಅಥವಾ ತಿರಸ್ಕೃತವೂ ಆಗಬಹುದು. ಕಂಪೇರ್‌ನ್ನು ಬಳಸಿ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಹೋಲಿಸಬಹುದು. ಪ್ರೊಟೆಕ್ಟ್‌ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಡಾಕ್ಯುಮೆಂಟನ್ನು ಇತರ ಲೇಖಕರು ಎಡಿಟ್ ಮಾಡದಂತೆ ರಕ್ಷಿಸಬಹುದು.

ಡಾಕ್ಯುಮೆಂಟ್ ವಿವ್ಸ್ ಮತ್ತು ಶೋ ಗ್ರೂಪ್ಸ್ (Document views and show Groups) (ಚಿತ್ರ 20)

Fig 20

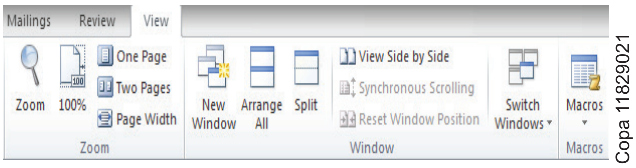


ವಿವ್ (view) ಟ್ಯಾಬ್ ಇದು ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುವ ವಿಧಾನವನ್ನು ತೋರಿಸುತ್ತದೆ. ಡಾಕ್ಯುಮೆಂಟ್ ವಿವ್‌ನಲ್ಲಿ ಪ್ರಿಂಟ್ ಲೇಔಟ್ , ವರ್ಡ್‌ನ ಸಾಮಾನ್ಯ ವಿವ್ , ಫುಲ್ ಸ್ಕ್ರೀನ್ ರೀಡಿಂಗ್, ಟ್ಯಾಬ್ ಮತ್ತು ರಿಬ್ಬನ್‌ಗಳನ್ನು ಮಿನಿಮೈಸ್ ಮಾಡುವುದು, HTML ಹೊಂದಾಣಿಕೆಯ ವೆಬ್ ಲೇಔಟ್, ಟೆಕ್ಸ್ಟ್ ಮೋಡ್‌ನಲ್ಲಿರುವ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಬೇಸಿಕ್ ಆವೃತ್ತಿಯ ಔಟ್ ಲೈನ್ ವಿವ್, ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಎಡಿಟಿಂಗ್ ಮೋಡ್‌ಗಾಗಿ ಡ್ರಾಫ್ಟ ಮೋಡ್.ಇತ್ಯಾದಿಗಳು ಇರುತ್ತದೆ.

ಬಳಕೆದಾರರ ಆದ್ಯತೆಯ ಪ್ರಕಾರ ರೂಲರ್, ಗ್ರಿಡ್‌ಲೈನ್ಸ್ ಮತ್ತು ನ್ಯಾವಿಗೇಷನ್ ಪೇನ್ ಇವುಗಳನ್ನು ತೋರಿಸಬಹುದು ಅಥವಾ ಮರೆಮಾಡಬಹುದು.

ಝೂಮ್ / ವಿಂಡೊ/ಮ್ಯಾಕ್ರೋ ಗ್ರೂಪ್ಸ್ (Zoom/Window/Macro Groups) (ಚಿತ್ರ 21)

Fig 21



ಒಂದು ಪುಟದ ನೋಟವನ್ನು ಪೂರ್ಣ ಪುಟಕ್ಕೆ, ಎರಡು ಪುಟಗಳ ವೀಕ್ಷಣೆಗೆ , ಡಾಕ್ಯುಮೆಂಟ್‌ನ 100% ಗೆ, ಮತ್ತು ಕಸ್ಟಮ್ ವಿವ್‌ಗೆ

© NIMI NOT TO BE REPUBLISHED

ಝೂಮ್ ಮಾಡಬಹುದು. ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸುಲಭವಾಗಿ ಎಡಿಟಿಂಗ್ ಮಾಡಲು ವಿಂಡೋನ್ನು ಎರಡು ಭಾಗವಾಗಿ ವಿಭಜಿಸಬಹುದು. ಮತ್ತು ತೆರೆದಿರುವ ಎಲ್ಲ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ವೀಕ್ಷಣೆಗಾಗಿ ಹೊಂದಿಸಬಹುದು. ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವಾಗ ಮೇಲಿಂದ ಮೇಲೆ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಕಮಾಂಡ್‌ಗಳಿಗಾಗಿ ಮ್ಯಾಕ್ರೋ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಇದು ಕಮಾಂಡ್‌ಗಳ ಗುಂಪನ್ನು ಸಂಗ್ರಹಿಸಲು ರೆಕಾರ್ಡ್ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಇತರ ಭಾಗದಲ್ಲಿ ಅಥವಾ ಬೇರೆ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಒಂದೇ ಕ್ಲಿಕ್‌ನಿಂದ ಮತ್ತೆ ಅದನ್ನು ಪುನರಾವರ್ತಿಸುತ್ತದೆ.

ಫೈಲ್ ರಚಿಸುವುದು. ಸೇವ್ ಮತ್ತು ಇತರ ಆಯ್ಕೆಗಳು (Creating a file , save, and other options)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ವಿಷಯಗಳು

- **ಹೊಸ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಹೇಗೆ ರಚಿಸುವುದು, ಸೇವ್ ಮಾಡುವುದು ಮತ್ತು ಪ್ರಿಂಟ್ ತೆಗೆಯುವುದು ಎಂಬುದನ್ನು ವಿವರಿಸಿ**
- **ಎಡಿಟ್ ಮಾಡುವುದು, ಟೆಕ್ಸ್ಟ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು ತಿಳಿಸಿ**
- **ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಡೇಟಾದೊಂದಿಗೆ ಟೇಬಲ್‌ನ್ನು ಹೇಗೆ ಬಳಸುವುದು ಎಂಬುದನ್ನು ಸಂಕ್ಷೇಪ್ತವಾಗಿ ತಿಳಿಸಿ**
- **ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸ್ಟೈಲ್‌ಗಳನ್ನು ಹೇಗೆ ರಚಿಸುವುದು ಮತ್ತು ಭವಿಷ್ಯದ ಬಳಕೆಗಾಗಿ ಹೇಗೆ ಸೇವ್ ಮಾಡುವುದು ಎಂಬುದನ್ನು ವಿವರಿಸಿ**
- **ವರ್ಡ್ 2010 ರ ಕೆಲವು ವಿಶಿಷ್ಟ ಲಕ್ಷಣಗಳನ್ನು ವಿವರಿಸಿ**
- **ಮೇಲ್ ಮರ್ಜ್ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಸಂಕ್ಷೇಪ್ತವಾಗಿ ವಿವರಿಸಿ.**

ಫೈಲ್ ಇನ್ಫೋ ವಿವ್ (File Info View) (**ಚಿತ್ರ** 1): ವರ್ಡ್ 2010 ಹೊಸ ಡಾಕ್ಯುಮೆಂಟ್ Ctrl + N ಅಥವಾ ಫೈಲ್ ಮೆನು -> New ಆಯ್ಕೆಯ ಮೂಲಕ ರಚಿಸಬಹುದು. ಮೂಲತಃ ಈ ರಚಿಸಿದ ಫೈಲ್ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ docx ಎನ್ನುವ ವಿಸ್ತೃತ ರೂಪದಿಂದ ಸೇವ್ ಮಾಡಲಾಗುತ್ತದೆ.

ಚಿತ್ರ 2 ವರ್ಡ್‌ನಲ್ಲಿ ಸೇವ್ ಆಯ್ಕೆಗಳು (Save options in Word) (Fig 2): ವರ್ಡ್ ಫೈಲ್‌ನ್ನು, ಸೇವ್ ಆ್ಯಸ್ ಆಯ್ಕೆಯನ್ನು ಬಳಸಿಕೊಂಡು ಯಾವುದೇ ಕಂಪ್ಯಾಟಿಬಲ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಅಥವಾ ಆಫೀಸ್ 2003 ಅಥವಾ ಅದರ ಹಿಂದಿನ ಆವೃತ್ತಿಗಳಲ್ಲಿ ಸೇವ್ ಮಾಡಬಹುದು. ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಮುಖ್ಯ ಉಪಯುಕ್ತತೆ ಎಂದರೆ ವರ್ಡ್ ಪ್ರಾಸೆಸಿಂಗ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ರಚಿಸುವುದು. ಇದು ಯಾವುದೇ ಪಬ್ಲಿಕೇಷನ್ , ಲೆಟರ್, ಬ್ರೋಚರ್ ಇತ್ಯಾದಿಗಳ ಸ್ವರೂಪದ್ದಾಗಿರಬಹುದು. ಟೆಕ್ಸ್ಟ್ ಆಧಾರಿತ ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲು ವರ್ಡ್ ಎಲ್ಲ ಸ್ವರೂಪಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಉದ್ಯಮದ ಮಾನದಂಡಗಳಿಗನುಸಾರವಾಗಿ ಪ್ರಮುಖ ಸ್ವೀಕೃತ ಸ್ವರೂಪಗಳಲ್ಲಿ ಇದು ಔಟಪುಟ್ ಫೈಲ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ.

ವರ್ಡ್‌ನಲ್ಲಿ ಪ್ರಿಂಟ್ ಆಯ್ಕೆಗಳು (Print options in Word) (**ಚಿತ್ರ** 3): ಸೇವ್ ಆದಂತಹ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು File menu -> Print option ನ್ನು ಉಪಯೋಗಿಸಿ ಪ್ರಿಂಟ್ ಮಾಡಬಹುದು ಮತ್ತು ಇನ್ಸ್ಟಾಲ್ ಆದಂತಹ ಪ್ರಿಂಟರ್‌ನ ಬೆಂಬಲದಿಂದ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುವುದು ಸುಲಭವಾಗುತ್ತದೆ. ಅಡೋಬ್ ಅಕ್ರೋಬ್ಯಾಟ್ ಪ್ರೊಫೆಷನಲ್ ಇನ್ಸ್ಟಾಲ್ ಆಗಿದ್ದರೆ , ಅದೇ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಶೇರ್ ಮಾಡುವ ಉದ್ದೇಶಕ್ಕಾಗಿ ಪಿಡಿಎಫ್ ಫೈಲ್ ಆಗಿ ಸಂಗ್ರಹಿಸಬಹುದು. ಪ್ರಿಂಟ್ ಮತ್ತು ಸಂಗ್ರಹಿಸುವದರ ಹೊರತಾಗಿ ಫೈಲನ್ನು ವೆಬ್ ಪೇಜ್ ಆಗಿ ವೆಬ್‌ನಲ್ಲಿ ಪ್ರಕಟಿಸಬಹುದು. ಅಥವಾ ಭವಿಷ್ಯದ ಪ್ರಕಟಣೆಗಳಿಗಾಗಿ ಟೆಂಪ್ಲೇಟ್ ಆಗಿಯೂ ಉಪಯೋಗಿಸಬಹುದು.

ವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸುವಾಗ ಗಮನಿಸಬೇಕಾದ ಮೂಲ ಪರಿಕಲ್ಪನೆಗಳು ಎಂದರೆ ಟೆಕ್ಸ್ಟ್ ಗುಣಲಕ್ಷಣಗಳು(Text Properties). ಇದು ಫಾಂಟ್ ಟೈಪ್, ಫಾಂಟ್ ಸೈಝ್, ಟೆಕ್ಸ್ಟ್ ಕಲರ್, ಮತ್ತು ಸಾಮಾನ್ಯ ಟೆಕ್ಸ್ಟ್ ಅಲಂಕಾರಗಳು ಇವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಪ್ಯಾರಾಗ್ರಾಫ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು ರಚಿಸುವುದನ್ನು ಗಮನಿಸಬೇಕು. ಪ್ಯಾರಾಗ್ರಾಫ್ ಅಲೈನ್ಮೆಂಟ್ ಎಡ (left) , ಬಲ (right) , ಮಧ್ಯ (center)ಮತ್ತು ಜಸ್ಟಿಫೈಡ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಟೆಕ್ಸ್ಟ್ ಎಲಿಮೆಂಟ್‌ಗಳು ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್ , ಅನ್ ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್ , ಸಬ್ ಸೆಕ್ಷನ್ ಲಿಸ್ಟಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇವು ಹೋಮ್ ಟ್ಯಾಬ್‌ನ ಪ್ಯಾರಾಗ್ರಾಫ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಕಂಡುಬರುತ್ತವೆ. ಇದೇ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಕೋಟ್ ಗಳನ್ನು ರಚಿಸಲು ಟೆಕ್ಸ್ಟ್ ಇಂಡೆಂಟಿಂಗ್ ಕೂಡ ಇರುತ್ತದೆ. ಪುಟದ ಸೈಝ್‌ಗೆ ಅನುಗುಣವಾಗಿ ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ ಮೊದಲು ಮತ್ತು ನಂತರ , ಲೈನ್ ಎತ್ತರ ಇತ್ಯಾದಿ, ಲೈನ್ ಸ್ಪೇಸಿಂಗ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್ ಸ್ಪೇಸಿಂಗ್‌ನ್ನು ಹೊಂದಿಸಿಕೊಳ್ಳಬಹುದು. ಅಗತ್ಯವಿರುವ ಸ್ಥಳಗಳಲ್ಲಿ ಪೂರ್ವನಿರ್ಧರಿತ ಸ್ಟೈಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಟೆಂಪ್ಲೇಟ್‌ಗಳು ಲಭ್ಯವಿದ್ದರೂ ಇದು ಕಸ್ಟಮ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು ಸಹ ರಚಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

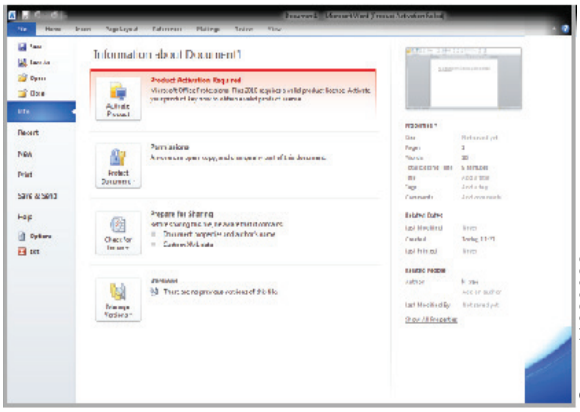
ವಿವಿಧ ಉಪಯುಕ್ತತೆಗಳಿಗಾಗಿ ಟೇಬಲ್‌ಗಳನ್ನು ಅಳವಡಿಸಬಹುದು. ಸಾಮಾನ್ಯ ಟೇಬಲ್ ಹೊಂದಾಣಿಕೆಯಾಗುವಂತಹ ಅಗಲ ಮತ್ತು ಕಾಲಮ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಅದ್ದರಿಂದ ಅದನ್ನು ಸರಳವಾಗಿ Insert > Table ಆಯ್ಕೆಯನ್ನು ನೇರವಾಗಿ ಉಪಯೋಗಿಸಿ ಸೇರಿಸಬಹುದು.

ಟೇಬಲ್ ಇನ್ಸರ್ಟ್ ವಿಝಾರ್ಡ್ ಆಯ್ಕೆಗಳು (Table Insert wizard options) (**ಚಿತ್ರ** 4): ಕಸ್ಟಮ್ ಗಾತ್ರದ ಟೇಬಲ್‌ನ ಅವಶ್ಯಕತೆ ಇದ್ದರೆ,

©NIMI NOT TO BE REPUBLISHED

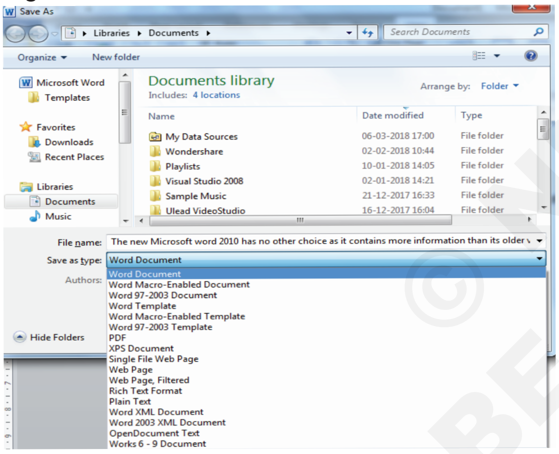
ಅದನ್ನು insert table > draw table ಆಯ್ಕೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ ರಚಿಸಬಹುದು. ಇನ್ಸರ್ಟ್ ಟೇಬಲ್ ಕಾಲಮ್ , ವಿಡ್ತ್ , ಮತ್ತು ಇತರ ಪ್ರಾಪರ್ಟೀಗಳನ್ನು ಅನುಮತಿಸುತ್ತದೆ. ಡ್ರಾ ಟೇಬಲ್ ಲಭ್ಯವಿರುವ ಟೆಕ್ಸ್ಟ್ ವಿಷಯಗಳ ಪ್ರಕಾರ ಪೆನ್ ಡ್ರಾಯಿಂಗ್ ಟೂಲ್‌ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಹೊಸ ಟೇಬಲ್‌ನ್ನು ರಚಿಸುತ್ತದೆ.

Fig 22



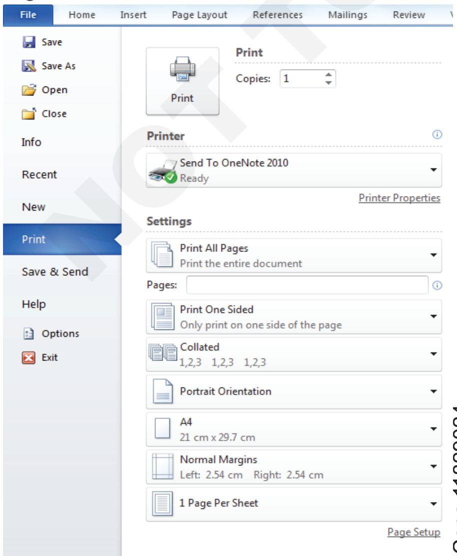
Copa 11829022

Fig 23



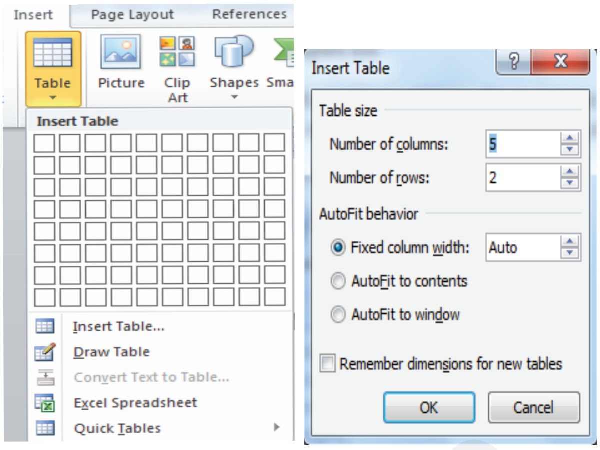
Copa 11829023

Fig 24



Copa 11829024

Fig 25

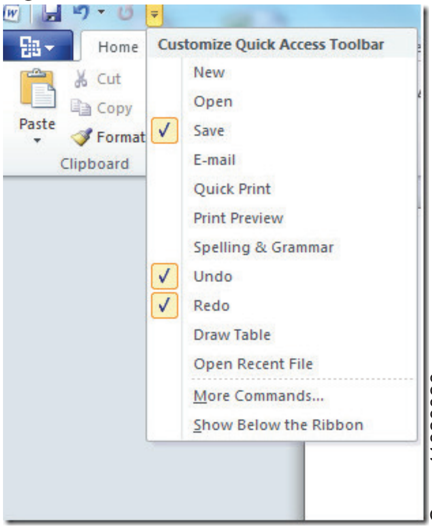


Copa 11829025

ಕಸ್ಟಮೈಝೇಬಲ್ ಕ್ವಿಕ್ ಅಕ್ಸೆಸ್ ಟೂಲ್ ಬಾರ್ (Customizable Quick Access Tool Bar)

ವರ್ಡ್ 2010ರ ಕ್ವಿಕ್ ಅಕ್ಸೆಸ್ ಟೂಲ್ ಬಾರ್ ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಇದು ಆಫೀಸ್ ಬಟನ್‌ನ ಹತ್ತಿರವಿರುವ ಅಪ್ಲಿಕೇಷನ್ ವಿಂಡೊನ ಮೇಲಿನ ಎಡಭಾಗದಲ್ಲಿದೆ. ಡೀಫಾಲ್ಟ್ ಆಗಿ ಇದು ಈ ಕೆಳಗಿನ ಮೂರು ಆಯ್ಕೆಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. ಸೇವ್ (Save), ಅನ್‌ಡು (Undo) ಮತ್ತು ರೀಡು (Redo). ಆದರೆ ಇದನ್ನು ಸುಲಭವಾಗಿ ಕಸ್ಟಮೈಝ್ ಮಾಡಬಹುದು ಮತ್ತು ಇದಕ್ಕೆ ಹೆಚ್ಚಿನ ಆಯ್ಕೆಗಳನ್ನು ಸೇರಿಸಬಹುದು (ಚಿತ್ರ 5)

Fig 26



Copa 11829026

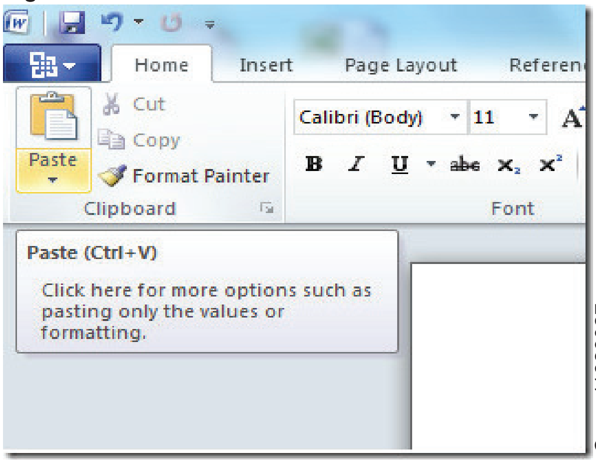
ಪೇಸ್ಟ್ ಪ್ರಿವಿವ್ (Paste Preview)

ಹೆಚ್ಚಿನ ಬಳಕೆದಾರರಿಗೆ ಅವರ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಏನನ್ನಾದರೂ ಕಾಪಿ ಮತ್ತು ಪೇಸ್ಟ್ ಮಾಡಿದ ನಂತರ ಕೆಲವು ಬದಲಾವಣೆಗಳನ್ನು ರದ್ದುಗೊಳಿಸಬೇಕಾದ ಅಗತ್ಯವಿರುತ್ತದೆ. ವರ್ಡ್ 2010 ರ ಬಳಕೆದಾರರಿಗೆ ಇದು ಸುಲಭವಾಗಿದೆ. ಪೇಸ್ಟ್ ಪ್ರಿವಿವ್ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಈ ಅನಗತ್ಯ ಹಂತವನ್ನು ತೆಗೆದುಹಾಕಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಇದು ಬಳಕೆದಾರರಿಗೆ ಮೌಲ್ಯಗಳನ್ನು ಅಥವಾ ಫಾರ್ಮಾಟಿಂಗ್ ಮಾತ್ರ ಪೇಸ್ಟ್ ಮಾಡಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ. (ಚಿತ್ರ 6)

Fig 27



Copa 11829027

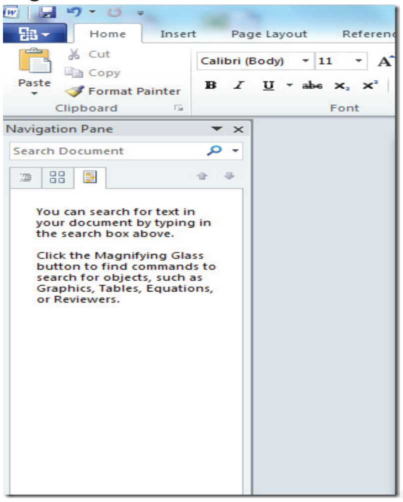
ನ್ಯಾವಿಗೇಷನ್ ಪೇನ್(Navigation Pane)

ಮೈಕ್ರೊಸಾಫ್ಟ ಆಫೀಸ್‌ನ ಹಿಂದಿನ ಆವೃತ್ತಿಗಳಲ್ಲಿ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಯಾವುದೇ ಶಬ್ದವನ್ನು ಕಂಡುಹಿಡಿಯಲು ನಾವು Ctrl + F ಕೀಯನ್ನು ಉಪಯೋಗಿಸುತ್ತಿದ್ದೆವು. ವರ್ಡ್ 2010 ಈ ಆಯ್ಕೆಗೆ ಹೊಸ ಮ್ಯಾಜಿಕ್ ಸೇರಿಸಿದೆ. Ctrl + F ಈಗ ನ್ಯಾವಿಗೇಷನ್ ಪೇನ್ ಆಗಿದೆ. ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಎಡಭಾಗದಲ್ಲಿ ಕಾಣಿಸುತ್ತದೆ. ಆಯಾ ಟ್ಯಾಬ್‌ಗಳ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡುವುದರ ಮೂಲಕ ಮೂರು ವಿವ್‌ಗಳು ಲಭ್ಯ ಇರುತ್ತವೆ. ಹೆಡಿಂಗ್ ವಿವ್ , ಥಂಬ್ ನೇಲ್ ಪೇಜ್ ವಿವ್ ಮತ್ತು ಸರ್ಚ್ ರಿಸಲ್ಟ್ ವಿವ್ (ಚಿತ್ರ 7)

ಕಸ್ಟಮೈಝೇಬಲ್ ರಿಬ್ಬನ್ ಬಟನ್(Customizable Ribbon Button)

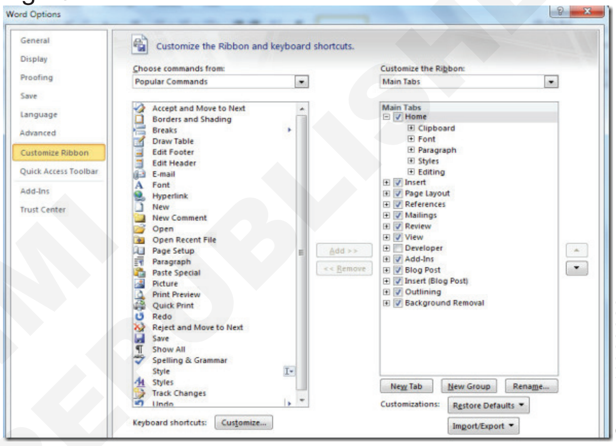
ವರ್ಡ್ 2010 ನಲ್ಲಿರುವ ರಿಬ್ಬನ್ ಬಟನ್ ವರ್ಡ್ 2007 ರಲ್ಲಿರುವಂತೆಯೆ ಕಾಣುತ್ತದೆ. ಆದರೆ ಇದರಲ್ಲಿ ಒಂದು ದೊಡ್ಡ ಸೇರ್ಪಡೆಯಾಗಿದೆ. ನೀವು 2010 ರ ಲ್ಲಿರುವ ರಿಬ್ಬನ್ ಬಟನ್ ಅನ್ನು ಕಸ್ಟಮೈಝ್ ಮಾಡಬಹುದು. ರಿಬ್ಬನ್ ಬಟನ್ ಅನ್ನು ಕಸ್ಟಮೈಝ್ ಮಾಡುವುದಕ್ಕಾಗಿ ಕೆಳಗಿನ ಆಯ್ಕೆಯನ್ನು ನ್ಯಾವಿಗೇಟ್ ಮಾಡಿ Office Button > Word Option > Customize Ribbon (ಚಿತ್ರ 8)

Fig 28



Copa 11829028

Fig 29

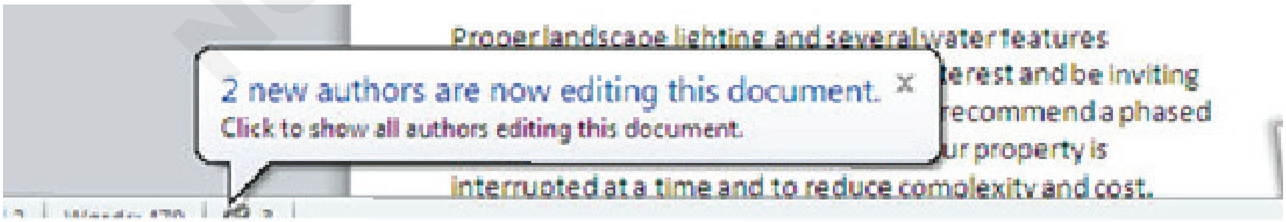


Copa 11829029

ಸುಧಾರಿತ ಸಹಯೋಗಗಳು (Improved Collaborations)

ಮೈಕ್ರೊಸಾಫ್ಟ ವರ್ಡ್ 2010 ಕೋ ಆಥರಿಂಗ್ (Co Authoring) ಎನ್ನುವ ಹೊಸ ವೈಶಿಷ್ಟ್ಯವನ್ನು ಹೊಂದಿದೆ. ಇದು ಒಂದು ಸಮಯದಲ್ಲಿ ಒಬ್ಬರಿಗಿಂತ ಹೆಚ್ಚು ಲೇಖಕರಿಗೆ ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಎಡಿಟ್ ಮಾಡಲು ಅನುವು ಮಾಡಿ ಕೊಡುತ್ತದೆ. ವರ್ಡ್ 2010 ಒಂದು ಡಾಕ್ಯುಮೇಂಟ್‌ನ್ನು ಎಷ್ಟು ಲೇಖಕರು ಎಡಿಟ್ ಮಾಡುತ್ತಿದ್ದಾರೆ ಎನ್ನುವುದನ್ನು ಹೇಳುತ್ತದೆ ಮತ್ತು ಅವುಗಳ ಬದಲಾವಣೆಗಳನ್ನು ಸಹ ವೀಕ್ಷಿಸಬಹುದು. (ಚಿತ್ರ 9)

Fig 30



Copa 11829030

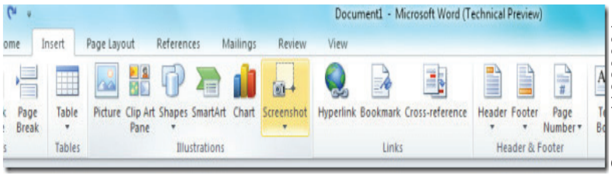
ಸ್ಕ್ರೀನ್ ಕ್ಯಾಪ್ಚರ್ ಟೂಲ್ (Screen Capture Tool)

ವರ್ಡ್ 2010 ಸ್ಕ್ರೀನ್ ಕ್ಯಾಪ್ಚರಿಂಗ್ ಎನ್ನುವ ವೈಶಿಷ್ಟ್ಯವನ್ನು ಹೊಂದಿದೆ. ಈಗ ಸ್ಕ್ರೀನ್ ಶಾಟ್‌ನ್ನು ಸೆರೆಹಿಡಿಯಲು ವರ್ಡ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲು ಬೇರೆ ಯಾವುದೇ ಅಥವಾ ಹೆಚ್ಚುವರಿ ಟೂಲ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಅವಶ್ಯಕತೆ ಇಲ್ಲ. ವರ್ಡ್ 2010 ನಲ್ಲಿ ಅಂತರ್ನಿರ್ಮಿತ ಟೂಲ್‌ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು

© NIMI NOT TO BE REPUBLISHED

ಸ್ಕ್ರೀನ್‌ನ ಯಾವುದೇ ಭಾಗವನ್ನು ಸೆರೆಹಿಡಿಯಬಹುದು. ಈ ಕೆಳಗಿನ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಸ್ಕ್ರೀನ್ ಶಾಟ್‌ನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು. Insert > Screenshot (ಚಿತ್ರ 10)

Fig 31

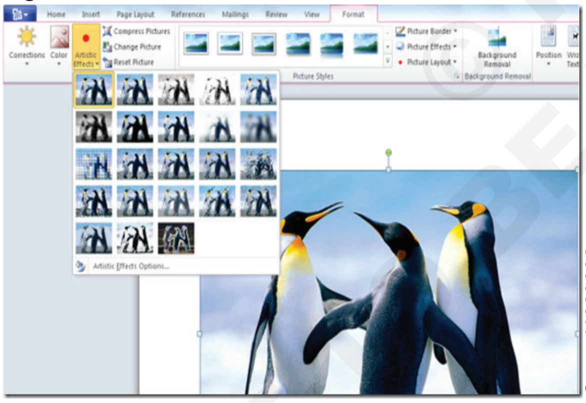


Copa 11829031

ಆರ್ಟಿಸ್ಟಿಕ್ ಇಫೆಕ್ಟ್ಸ್ (Artistic Effects)

ವರ್ಡ್ 2010 ನ ಬಳಕೆದಾರರು ಈಗ ಚಿತ್ರಗಳಿಗೆ ಹಲವಾರು ಆಕರ್ಷಕ ಮತ್ತು ಕಲಾತ್ಮಕ ಇಫೆಕ್ಟ್‌ಗಳನ್ನು ನೀಡಬಹುದು. ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಕಲಾತ್ಮಕ ಪರಿಣಾಮನ್ನು ನೀಡಲು ಈ ಕೆಳಗಿನ ಆಯ್ಕೆಗೆ ನ್ಯಾವಿಗೇಟ್ ಮಾಡಿ. Insert >Illustrations > Picture. ನಂತರ ಬ್ರೌಸ್ ಮಾಡಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿ. ಒಮ್ಮೆ ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್ ನಲ್ಲಿ ಚಿತ್ರವನ್ನು ಅಳವಡಿಸಿದ ನಂತರ ಪಿಕ್ಚರ್ ಟೂಲ್ಸ್ ಕಾಂಟೆಕ್ಸ್ಚುವಲ್ ಟ್ಯಾಬ್‌ನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ನೀವು ಹೊಸ ಕಲಾತ್ಮಕ ಇಫೆಕ್ಟ್‌ಗಳ ಡ್ರಾಪ್ ಡೌನ್ ಬಟನ್ ಅನ್ನು ಇಲ್ಲಿ ನೋಡಲು ಸಾಧ್ಯವಾಗುತ್ತದೆ. (ಚಿತ್ರ 11)

Fig 32

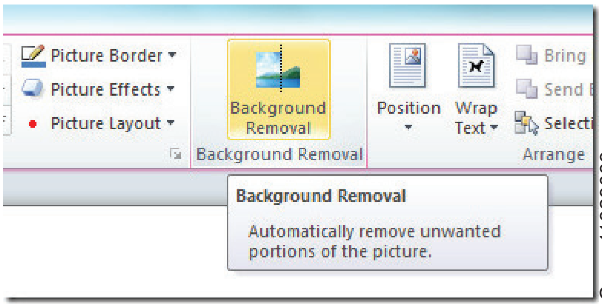


Copa 11829032

ಬ್ಯಾಕ್‌ಗ್ರೌಂಡ್ ರಿಮೂವಲ್ ಆಯ್ಕೆ (The Background Removal Option)

ಆಫೀಸ್ 2010 ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್ ರಿಮೂವ್ ಮಾಡಲು ಅದ್ಭುತ ಆಯ್ಕೆಯನ್ನು ಹೊಂದಿದೆ. ಇದು ಯಾವುದೇ ಚಿತ್ರದ ಹಿನ್ನೆಲೆಯನ್ನು ಸರಳವಾಗಿ ತೆಗೆದು ಹಾಕುತ್ತದೆ. ಹೌದು ಇನ್ನು ಮುಂದೆ ನಿಮಗೆ ಹಿನ್ನೆಲೆಯನ್ನು ತೆಗೆದುಹಾಕಲು ಫೋಟೋಶಾಪ್‌ನ ಅವಶ್ಯಕತೆ ಇಲ್ಲ. ಮೊದಲು Insert > Picture ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿ ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಚಿತ್ರವನ್ನು ಸೇರಿಸಿ. ನಂತರ ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್ ರಿಮೂವಲ್ ಟೂಲ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಮತ್ತು ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್‌ನ್ನು ತೆಗೆದುಹಾಕಿರಿ. (ಚಿತ್ರ 12)

Fig 33



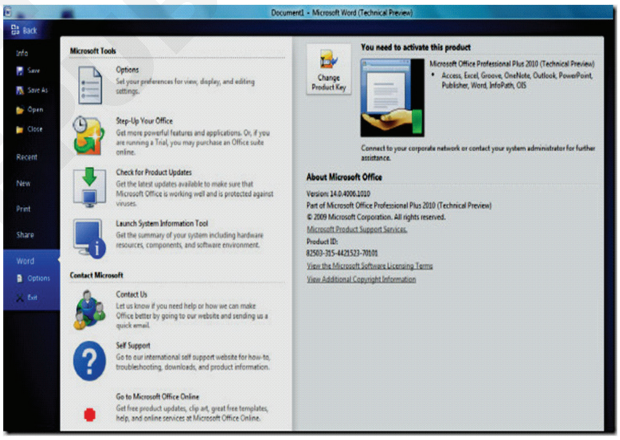
Copa 11829033

ಆಫೀಸ್ ಬ್ಯಾಕ್‌ಸ್ಟೇಜ್ (Office Backstage)

ಆಫೀಸ್ ಬ್ಯಾಕ್‌ಸ್ಟೇಜ್ ಎನ್ನುವುದು ಒಂದು ಹೊಸ ಪರಿಕಲ್ಪನೆಯಾಗಿದೆ. ಇದು ಹಳೆಯ ಆಫೀಸ್ ಬಟನ್‌ನ ವರ್ಧಿತ ರೂಪವಾಗಿದೆ ಮತ್ತು ಹೆಚ್ಚಿನ ಬಳಕೆದಾರ ಸ್ನೇಹಿ ಮೆನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಹೆಚ್ಚಿನ ಮಟ್ಟದಲ್ಲಿ ಬಳಕೆದಾರರಿಗೆ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು, ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳನ್ನು ಅಥವಾ ಸ್ಪ್ರೆಡ್ ಶೀಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. (ಚಿತ್ರ 13)

ಆಸಕ್ತಿದಾಯಕ ವೈಶಿಷ್ಟ್ಯ, ಸರಿಯೆ? ಇಲ್ಲಿ ಇದರ ಬಗ್ಗೆ ಇನ್ನಷ್ಟು ಓದಿರಿ.

Fig 34



Copa 11829034

ವರ್ಡ್ ಆರ್ಟ್‌ನಲ್ಲಿ ಹೊಸ ಆರ್ಟ್ ಇಫೆಕ್ಟ್‌ಗಳು (New Art Effects in WordArt)

ಉಳಿದ ವೈಶಿಷ್ಟ್ಯಗಳಂತೆ, ವರ್ಡ್ ಆರ್ಟ್‌ನ್ನು ಹೊಸ ವರ್ಣರಂಜಿತ ಆರ್ಟ್ ಇಫೆಕ್ಟ್‌ಗಳಿಂದ ನವೀಕರಿಸಲಾಗಿದೆ. ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿ, ವರ್ಡ್ ಆರ್ಟ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ, ಮತ್ತು ಲಭ್ಯವಿರುವ ಎಲ್ಲಾ ಆಯ್ಕೆಗಳ ಪಟ್ಟಿಯನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. (ಚಿತ್ರ 14)

ಲಿಗೇಚರ್ಸ್ (Ligatures)

ನೀವು ಲಿಗೇಚರ್ಸ್ ಬಗ್ಗೆ ಕೇಳಿರಬಹುದು. ಅವು ಫಾಂಟ್‌ಗಳು ಅಲಂಕಾರಿಕವಾಗಿ ಕಾಣುವಂತೆ ಮಾಡುತ್ತವೆ ಮತ್ತು ಇವು ಫಾಂಟ್‌ಗಳು ರೆಗ್ಯುಲರ್ ಇರುವಂತೆ ನಿಮಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಹುಡುಕಲು ಅನುವು ಮಾಡಿಕೊಟ್ಟು ಅಕ್ಷರಗಳನ್ನು ಪ್ರತ್ಯೇಕವಾಗಿ ಇಡಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತವೆ. ಎಲ್ಲ ಫಾಂಟ್‌ಗಳು ಲಿಗೇಚರ್ಸ್‌ನ್ನು

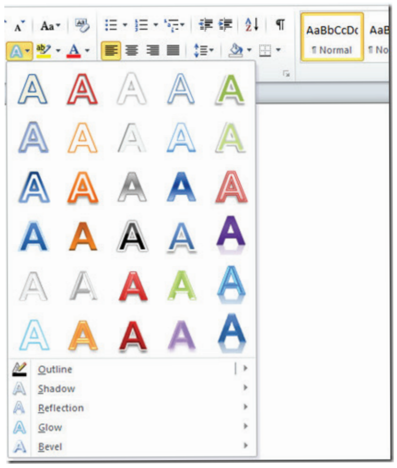
© NIMI NOT TO BE REPUBLISHED

ಬೆಂಬಲಿಸುವುದಿಲ್ಲ ಎಂಬುದು ನಿಜ ಆದರೆ ವಿವಿಧ ಪ್ರಕಾರದ ಫಾಂಟ್‌ಗಳು ಲಿಗೇಚರ್ಸ್‌ನ್ನು ಬೆಂಬಲಿಸುತ್ತವೆ. ನೀವು ಇವುಗಳನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಸಕ್ರೀಯಗೊಳಿಸಬಹುದು. Font Preferences > advanced , ನಂತರ Standard Only ಆಯ್ಕೆಯನ್ನು ಲಿಗೆಚರ್ಸ್ ಡ್ರಾಪ್ ಡೌನ್ ಬಾಕ್ಸ್‌ನಿಂದ ಆಯ್ಕೆಮಾಡಿ . (ಚಿತ್ರ 15)

ಸುಧಾರಿತ ಸ್ಪೆಲ್ ಚೆಕ್ (Improved Spell Checks)

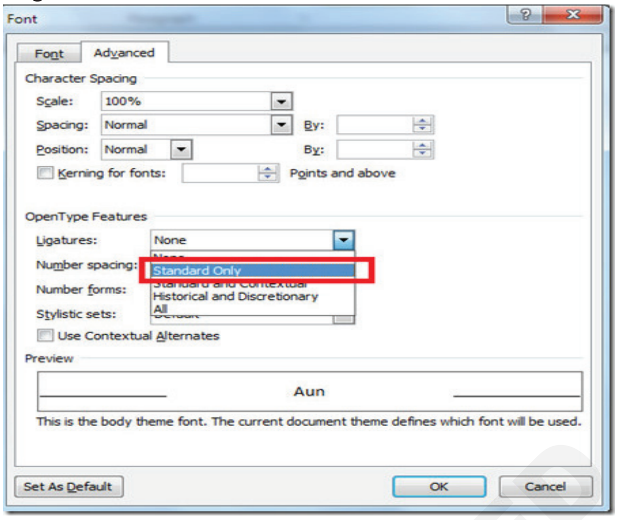
ವರ್ಡ್ 2010 ಅದರ ಸ್ಪೆಲ್ ಚೆಕರ್‌ಗೆ ಕೆಲವು ಹೊಸ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸೇರಿಸಿದೆ. ಈಗ ಅದು ತಪ್ಪನ್ನು ಪತ್ತೆ ಮಾಡುತ್ತದೆ ಮತ್ತುವಾಕ್ಯವನ್ನು ಬದಲಾಯಿಸಲು ಸೂಚಿಸುತ್ತದೆ.

Fig 35



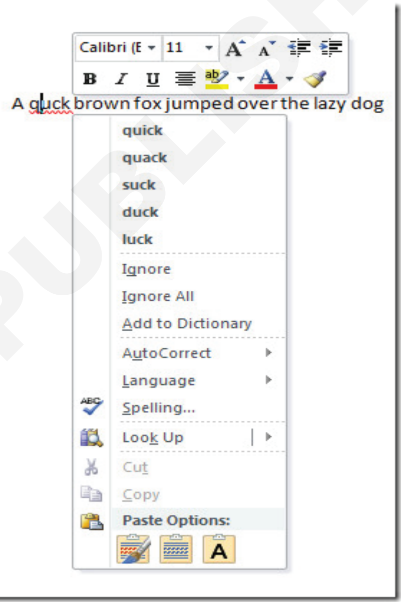
Copa 11829035

Fig 36



Copa 11829036

Fig 37



Copa 11829037

© NIMI NOT TO BE REPUBLISHED

ವರ್ಡ್ 2010 ರಲ್ಲಿನ ಶಾರ್ಟ್ ಕಟ್ ಕೀಗಳು (Shortcut keys in Word 2010)

ಉದ್ದೇಶ : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಎಂ ಎಸ್ ವರ್ಡ್‌ನಲ್ಲಿರುವ ಶಾರ್ಟ್ ಕಟ್ ಕೀಗಳನ್ನು ಕಲಿಯಿರಿ

CTRL+SHIFT+A	ಆಯ್ದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ದೊಡ್ಡ ಅಕ್ಷರಗಳಾಗಿ ಪರಿವರ್ತಿಸುತ್ತದೆ. ಅಥವಾ ವೈಸ್ ವರ್ಸಾ
CTRL+SHIFT+F	ಫಾಂಟ್ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ
CTRL+SHIFT+G	ವರ್ಡ್ ಕೌಂಟ್ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ
CTRL+SHIFT+S	ಅಪ್ಲೈ ಸ್ಟೈಲ್ಸ್ ಟಾಸ್ಕ್ ಪೇನ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ
ALT+R	ರಿವಿವ್ ಟ್ಯಾಬ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ
ALT+CTRL+1	ಹೆಡಿಂಗ್ 1 ಅನ್ನು ಅನ್ವಯಿಸುತ್ತದೆ, ಅಂತೆಯೆ ALT+CTRL+2 ಹೆಡಿಂಗ್ 2 ಅನ್ನು ಅನ್ವಯಿಸುತ್ತದೆ
CTRL+SHIFT+L	ಬುಲೆಟ್‌ನ್ನು ಅನ್ವಯಿಸುತ್ತದೆ.
CTRL+SHIFT+F5	ಬುಕ್ ಮಾರ್ಕ
CTRL+B	ಬೋಲ್ಡ್ ಟೆಕ್ಸ್ಟ

CTRL+I	ಇಟ್ಯಾಲಿಕ್ ಟೆಕ್ಸ್ಟ್
CTRL+U	ಅಂಡರ್ ಲೈನ್
CTRL+PAGE DOWN	ಮುಂದೆ ಬ್ರೌಸ್ ಮಾಡಲು
CTRL+E	ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ ಮಧ್ಯಭಾಗದಲ್ಲಿ ನ್ಯಾವಿಗೇಟ್ ಮಾಡಲು
CTRL+SHIFT+ENTER	ಕಾಲಮ್ ಬ್ರೇಕ್
CTRL+SHIFT+C	ಕಾಪಿ ಫಾರ್ಮ್ಯಾಟ್
ALT+SHIFT+F7	ನಿಘಂಟು (ಡಿಕ್ಷನರಿ)
ALT+CTRL+S	ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ವಿಭಜಿಸುತ್ತದೆ
CTRL+SHIFT+D	ಡಬಲ್ ಅಂಡರ್ ಲೈನ್
CTRL+END	ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಕೊನೆ
END	ಸಾಲಿನ ಕೊನೆ
CTRL+SHIFT+P	ಫಾಂಟ್ ಸೈಝ್‌ನ್ನು ಆಯ್ಕೆಮಾಡಲು
SHIFT+F5 or ALT+CTRL+Z	ಹಿಂದಿನ ಸ್ಥಿತಿಗೆ ಹಿಂತಿರುಗಲು
CTRL+SHIFT+	ಫಾಂಟ್‌ನ್ನು ದೊಡ್ಡ ದಾಗಿ ಮಾಡುವುದು
CTRL +]	ಫಾಂಟ್‌ನ್ನು ಒಂದು ಪಾಯಿಂಟ್ ದೊಡ್ಡದು ಮಾಡುವುದು
ALT +SHIFT+R	ಹೆಡರ್ ಫೂಟರ್ ಲಿಂಕ್
CTRL+K	ಹೈಪರ್ ಲಿಂಕ್
CTRL+M	ಇಂಡೆಂಟೇಷನ್
CTRL+J	ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ್ನು ಜಸ್ಟಿಫೈಮಾಡುವುದು
ALT +F8	ಮ್ಯಾಕ್ರೋಗಳನ್ನು ಸೇರಿಸುವುದು
ALT +SHIFT+K	ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಪರಿಶೀಲಿಸುವುದು
F10	ಮೆನು ಮೋಡ್
ALT +F7	ಮುಂದಿನ ತಪ್ಪು ಸ್ಪೆಲ್ಲಿಂಗ್‌ಗೆ ಚಲಿಸುವುದು
CTRL+H	ರೀಪ್ಲೇಸ್
CTRL+P	ಪ್ರಿಂಟ್
CTRL+E	ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ ಮಧ್ಯಭಾಗದಲ್ಲಿ ನ್ಯಾವಿಗೇಟ್ ಮಾಡಲು
CTRL+SHIFT+F12	ಪ್ರಿಂಟ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುವುದು
ALT+SHIFT+BACKSPACE	ರೀಡೂ
F12	ಸೇವ್ ಆ್ಯಸ್
CTRL+SHIFT+K	ಸ್ಮಾಲ್ ಕ್ಯಾಪ್ಸ್
CTRL+SHIFT+S	ಸ್ಟೈಲ್
SHIFT+F7	ಥೀಸರಸ್
ALT+SHIFT+T	ಟೈಮ್ ಫೀಲ್ಡ್
CTRL+SHIFT+M	ಅನ್‌ಇಂಡೆಂಟ್

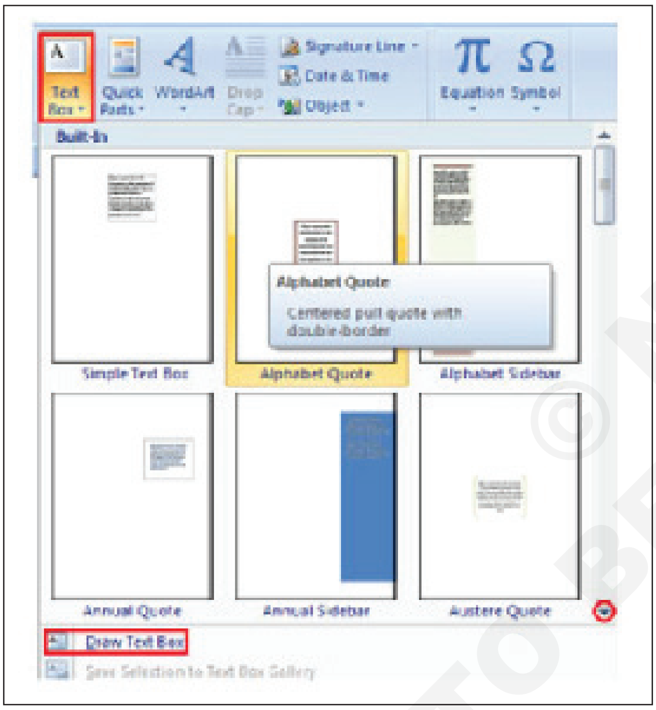
NOT TO BE REPUBLISHED © NIMI

ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳನ್ನು ಅಳವಡಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು, ಡಾಕ್ಯುಮೆಂಟ್ ಸೆಕ್ಷನ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಸಂರಚಿಸುವುದು (Insert, format text and paragraphs, Create and configure document sections)

ಉದ್ದೇಶಗಳು: ಈ ಪಾಠದ ಕೊನೆಗೆ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಎಂ ಎಸ್ ವರ್ಡ್‌ನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು
- ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು
- ಸೆಕ್ಷನ್ ಬ್ರೆಕ್‌ನ್ನು ಸೇರಿಸುವುದು.

ಎಂ ಎಸ್ ವರ್ಡ್‌ನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Insert Text in MS Word)

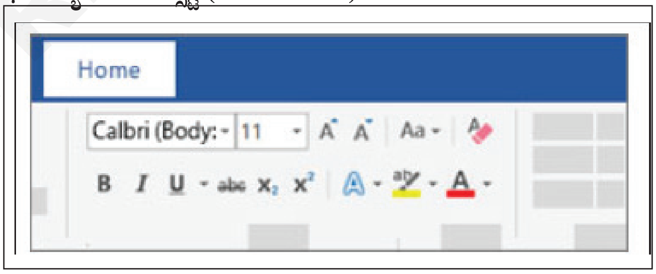


ರಿಬ್ಬನ್ ಕೆಳಗಡೆ ಇರುವ ಟೆಕ್ಸ್ಟ್ ಪ್ರದೇಶದಲ್ಲಿ ನಿಮಗೆ ಒಂದು ಕರ್ಸರ್ ಅಥವಾ ಟೆಕ್ಸ್ಟ್ ಸೇರಿಸುವ ಬಿಂದು ಕಾಣಸಿಗುತ್ತದೆ. ಈಗ ನೀವು ಟೈಪ್ ಮಾಡಲು ಪ್ರಾರಂಭಿಸಿದಾಗ, ನೀವು ಟೈಪ್ ಮಾಡಿದ ಶಬ್ದಗಳು ಸ್ಕ್ರೀನ್ ಮೇಲೆ ಟೆಕ್ಸ್ಟ್ ಪ್ರದೇಶದಲ್ಲಿ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತವೆ. ಸ್ಪೇಸ್ ಬಾರ್ ಕೀ, ಎಂಟರ್ ಕೀ ಅಥವಾ ಟ್ಯಾಬ್ ಕೀ ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಈ ಟೆಕ್ಸ್ಟ್ ಸೇರಿಸುವ ಕರ್ಸರ್‌ನ ಪ್ರದೇಶವನ್ನು ಬದಲಾಯಿಸಬಹುದು.

ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಲು ಅಥವಾ ವರ್ಡ್‌ನಲ್ಲಿ ಹೊಸ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತಯಾರಿಸಲು ಉಪಯೋಗಿಸುವ ಹಂತಗಳನ್ನು ಈ ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ:

- ಸ್ಟಾರ್ಟ್ ಮೆನುಗೆ ಹೋಗಿ ಮತ್ತು ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಐಕನ್ ನ್ನು ಹುಡುಕಿ.
- ಐಕನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್‌ನ್ನು ತೆರೆಯಿರಿ.
- ನೀವು ರಿಬ್ಬನ್‌ನ ಕೆಳಗೆ ಇರುವ ಟೆಕ್ಸ್ಟ್ ಪ್ರದೇಶದಲ್ಲಿ ಬ್ಲಿಂಕಿಂಗ್ ಕರ್ಸರ್ ಅಥವಾ ಸೇರಿಸುವ ಪಾಯಿಂಟ್‌ನ್ನು ನೋಡುವಿರಿ.
- ಈಗ, ನೀವು ಟೈಪ್ ಮಾಡಲು ಪ್ರಾರಂಭಿಸಿದ ಮೇಲೆ ಸ್ಕ್ರೀನ್ ಮೇಲಿನ ಟೆಕ್ಸ್ಟ್ ಪ್ರದೇಶದಲ್ಲಿ ಟೈಪ್ ಮಾಡಿದ ಶಬ್ದಗಳು ಕಾಣುತ್ತವೆ.
- ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವ ಪಾಯಿಂಟ್‌ನ ಸ್ಥಳವನ್ನು ಬದಲಾಯಿಸಲು ಸ್ಪೇಸ್ ಬಾರ್ ಕೀ, ಎಂಟರ್ ಕೀ ಅಥವಾ ಟ್ಯಾಬ್ ಕೀಯನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ಫಾರ್ಮ್ಯಾಟ್ ಟೆಕ್ಸ್ಟ್ (Format text)



1 ನೀವು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಬಯಸುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಒಂದೇ ಶಬ್ದವನ್ನು ಆಯ್ಕೆ ಮಾಡಲು, ಆ ಶಬ್ದದ ಎರಡು ಬಾರಿ ಕ್ಲಿಕ್ ಮಾಡಿ. ಟೆಕ್ಸ್ಟ್‌ನ ಒಂದು ಸಾಲನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ಆ ಸಾಲಿನ ಮೇಲೆ ಮೂರು ಬಾರಿ ಕ್ಲಿಕ್ ಮಾಡಿ.

2 ಫಾಂಟ್ , ಫಾಂಟ್‌ನ ಗಾತ್ರ , ಫಾಂಟ್ ಕಲರ್ ಬದಲಾಯಿಸಲು ಇರುವ ಆಯ್ಕೆಗಳನ್ನು ಆರಿಸಿಕೊಳ್ಳಿ ಅಥವಾ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಬೋಲ್ಡ್ (Bold) , ಇಟ್ಯಾಲಿಕ್ಸ್ (Italics) ಅಥವಾ ಅಂಡರ್‌ಲೈನ್ (Underline) ಮಾಡಲು ಆಯ್ಕೆಗಳನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

ಕಾಪಿ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ (Copy formatting)

1 ನೀವು ಕಾಪಿ ಮಾಡಲು ಬಯಸಿದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಅದರ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಸಹಿತ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED

2 Format Painter ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ ನೀವು ಕಾಪಿ ಮಾಡಲು ಬಯಸಿದ ಟೆಕ್ಸ್ಟನ್ನು ಅದರ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಸಹಿತವಾಗಿ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

> ಸೂಚನೆ : ನೀವು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಸ್ಥಳಗಳಲ್ಲಿ ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ್ನು ಕಾಪಿ ಮಾಡಲು ಬಯಸಿದರೆ, ಫಾರ್ಮ್ಯಾಟ ಪೈಂಟ್‌ರ್ ಮೇಲೆ ಎರಡು ಬಾರಿ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಸೆಕ್ಷನ್ ಬ್ರೆಕ್ ಸೇರಿಸುವುದು (Add a section break)

1 ನೀವು ಯಾವ ಸ್ಥಳದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲು ಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

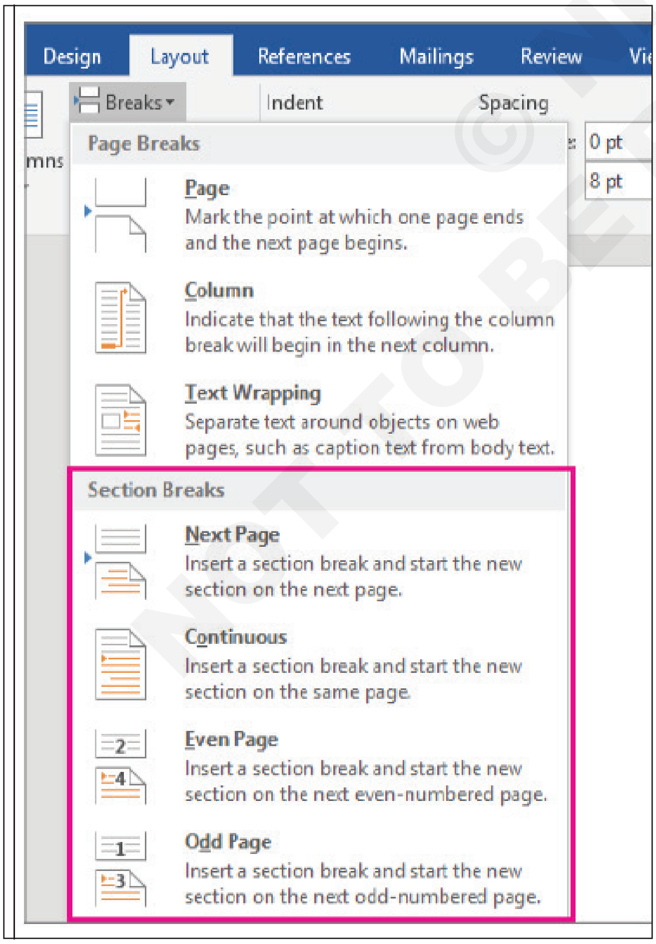
2 Layout --> Breaks ಇಲ್ಲಿಗೆ ಹೋಗಿರಿ.

3 ನೀವು ಬಯಸಿದ ಸೆಕ್ಷನ್ ಬ್ರೆಕ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಪೇಜ್ ಸೆಕ್ಷನ್ ಬ್ರೇಕ್ ಪುಟದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಅದೇ ಪುಟದಲ್ಲಿ ಸತತವಾಗಿ ಸೆಕ್ಷನ್ ಬ್ರೆಕ್ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತದೆ.

ಸೆಕ್ಷನ್ ಬ್ರೆಕ್ ಸೇರಿಸುವುದು (Add a section break)

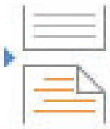
1 ನೀವು ಯಾವ ಸ್ಥಳದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲು ಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 Layout --> Breaks ಇಲ್ಲಿಗೆ ಹೋಗಿರಿ.



3 ನೀವು ಬಯಸಿದ ಸೆಕ್ಷನ್ ಬ್ರೆಕ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ :

- Next Page - ಪೇಜ್ ಸೆಕ್ಷನ್ ಬ್ರೆಕ್ ನಂತರದ ಪುಟದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತದೆ.



- Continuous : ಸೆಕ್ಷನ್ ಬ್ರೆಕ್ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಅದೇ ಪುಟದಲ್ಲಿ ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಈ ರೀತಿಯ ಸೆಕ್ಷನ್ ಬ್ರೆಕ್‌ನ್ನು ಹೊಸ ಪುಟವನ್ನು ಪ್ರಾರಂಬಿ ,ಸದೇ ಕಾಲಂಗಳ ಸಂಖ್ಯೆಯನ್ನು ಬದಲಾಯಿ ಾಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

- Even Page : ಸೆಕ್ಷನ್ ಬ್ರೆಕ್, ಮುಂದಿನ ಅಥವಾ ನಂತರದ ಸಮ ಸಂಖ್ಯೆಯ ಪುಟದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತದೆ.

- Odd Page : ಸೆಕ್ಷನ್ ಬ್ರೆಕ್, ಮುಂದಿನ ಅಥವಾ ನಂತರದ ಬೆಸ ಸಂಖ್ಯೆಯ ಪುಟದಲ್ಲಿ ಹೊಸ ಸೆಕ್ಷನ್‌ನ್ನು ಪ್ರಾರಂಬಿ

© NIMI NOT TO BE REPUBLISHED

ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡುವುದು (Create, modify tables)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

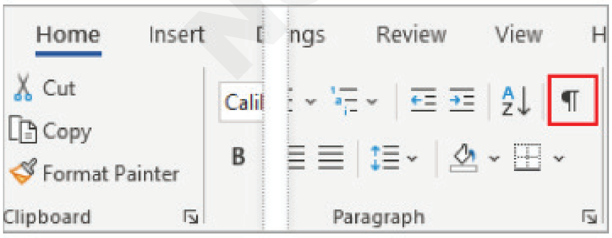
- **ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು**
- **ಟೆಬಲ್‌ನ್ನು ಟೆಕ್ಸ್ಟ್‌ಗೆ ಬದಲಾಯಿಸುವುದು**
- **ಟೇಬಲ್‌ನ್ನು ವಿದಳನಗೊಳಿಸುವುದು ಅಥವಾ ಒಡೆಯುವುದು.**

ಒಂದು ದೊಡ್ಡ ಟೇಬಲ್‌ನ್ನು ಅಳವಡಿಸಲು ಅಥವಾ ನಿಮಗೆ ಬೇಕಾದ ಟೇಬಲ್‌ಗಾಗಿ Insert --> Table --> Insert Table ಈ ರೀತಿಯಾಗಿ ಆಯ್ಕೆ ಮಾಡಿರಿ. ಸೂಚನೆ : ನೀವು ಈಗಾಗಲೇ ಟ್ಯಾಬ್‌ಗಳಿಂದ ಪ್ರತ್ಯೇಕಿಸಲ್ಪಟ್ಟ ಟೆಕ್ಸ್ಟ್‌ನು ಹೊಂದಿದ್ದರೆ, ನೀವು ಶೀಘ್ರವಾಗಿ ಇದನ್ನು ಟೇಬಲ್‌ಗೆ ಬದಲಾಯಿಸಬಹುದು. ಅದಕ್ಕಾಗಿ Insert --> Table ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Convert Text to Table ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು (Create tables)

ನಿರ್ದಿಷ್ಟವಾಗಿ ಸಂಖ್ಯೆಗಳನ್ನು ಹೊಂದಿದ ಡಾಟಾ ಎಂದರೆ ಡಾಟಾ ಸೆಟ್‌ಗಳನ್ನು ಹೆಚ್ಚು ಸ್ಪಷ್ಟವಾಗಿ ಮತ್ತು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಟೆಕ್ಸ್ಟ್‌ನ ಪ್ಯಾರಾಗ್ರಾಫ್ ಬದಲಾಗಿ ಟೇಬಲ್‌ನಲ್ಲಿ ಮಾತ್ರ ನಮೂದಿಸಲು ಅಥವಾ ಪ್ರಸ್ತುತಪಡಿಸುವುದು ಉತ್ತಮವಾಗಿರುತ್ತದೆ. ಟೇಬಲ್‌ಗಳು ದೊಡ್ಡಗಾತ್ರದ ಡಾಟಾವನ್ನು ಪ್ರಸ್ತುತಪಡಿಸಲು ಅಥವಾ ಕ್ಲಿಷ್ಟಕರವಾದ ಡಾಟಾವನ್ನು ಸುಲಭವಾಗಿ ಓದಲು ಬರುವಂತ ಮಾದರಿಯಲ್ಲಿ ಮತ್ತು ಅರ್ಥಮಾಡಿಕೊಳ್ಳಲು ಸುಲಭವಾಗುವಂತೆ ಅಡ್ಡಸಾಲು (rows) ಮತ್ತು ಉದ್ದಸಾಲುಗಳಲ್ಲಿ (columns) ಪ್ರಸ್ತುತಪಡಿಸುತ್ತವೆ. ಈ ರೋ ಮತ್ತು ಕಾಲಂಗಳು ಡಾಟಾದ ಅರ್ಥವನ್ನು ತಿಳಿಸಲು ಹೆಡರ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೇಬಲ್‌ಗೆ ಅಥವಾ ಟೇಬಲ್‌ನ್ನು ಟೆಕ್ಸ್ಟ್‌ಗೆ ಬದಲಾಯಿಸಲು, Home Tab ನಲ್ಲಿ ಇರುವ Show/Hide ಪ್ಯಾರಾಗ್ರಾಫ ಮಾರ್ಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಈಗ ನೀವು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್ ಯಾವ ರೀತಿ ಪ್ರತ್ಯೇಕಿಸಲ್ಪಡುತ್ತದೆ ಎಂಬುದನ್ನು ನೋಡುವಿರಿ.



ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೇಬಲ್‌ಗೆ ಬದಲಾಯಿಸುವುದು (Convert text to a table)

1 ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಕಾಲಂಗಳಿಗೆ ಯಾವ ಸ್ಥಳದಲ್ಲಿ ಬದಲಾಯಿಸಬೇಕು ಎಂದು ಸೂಚಿಸಲು ಪ್ರತ್ಯೇಕಿಸುವ ಅಕ್ಷರಗಳಾದಂತಹ ಅಲ್ಪವಿರಾಮ (,) ಅಥವಾ ಟ್ಯಾಬ್‌ಗಳನ್ನು ಸೇರಿಸಬೇಕು.

> **ಸೂಚನೆ : ಒಂದು ವೇಳೆ ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ಅಲ್ಪವಿರಾಮ** (comma) **ಇದ್ದರೆ , ಪ್ರತ್ಯೇಕಿಸುವ ಅಕ್ಷರಗಳಾಗಿ ಟ್ಯಾಬ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ.**

2 ಹೊಸ ಟೇಬಲ್ ರೋ ಅನ್ನು ಪ್ರಾರಂಭಿಸಲು ಪ್ಯಾರಾಗ್ರಾಫ್ ಮಾರ್ಕ್‌ನ್ನು ಉಪಯೋಗಿಸಿರಿ.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಟ್ಯಾಬ್‌ಗಳು ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್ ಮಾರ್ಕ್‌ಗಳು 3 ಕಾಲಂಗಲು ಮತ್ತು 2 ರೋ ಗಳಿರುವ ಒಂದು ಟೇಬಲ್‌ನ್ನು ತಯಾರಿಸುತ್ತದೆ.

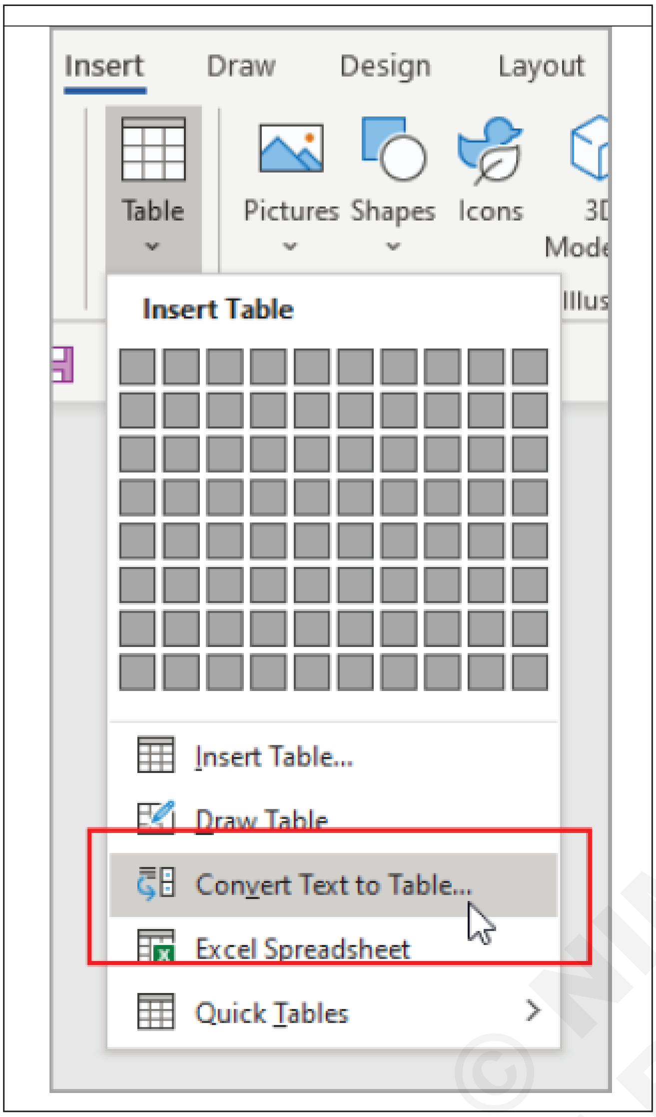
```
Red, yellow → blue, green → orange, purple¶
Red, yellow → blue, green → orange, purple¶
```

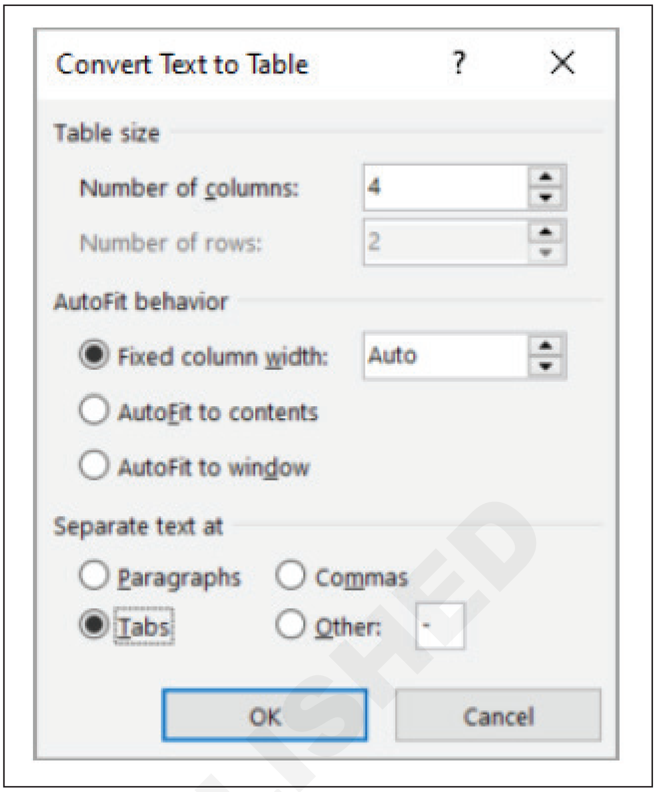
3 ನೀವು ಬದಲಾಯಿಸಬೇಕಾಗಿರುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ , ನಂತರ Insert > Table > Convert Text to Table ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

4 Convert Text to Table ಬಾಕ್ಸ್ ನಲ್ಲಿ , ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಗಳನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

Table Size ನಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳು ಕಾಲಂ ಮತ್ತು ರೋ ನ ಸಷ್ಪೆ ೖಗಳೊಂದಿಗೆ ಸರಿಹೊಂದುತ್ತವೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED





1 AutoFit behavior ನಲ್ಲಿ, ನೀವು ನಿಮ್ಮ ಟೇಬಲ್‌ನ್ನು ಯಾವ ರೀತಿ ನೋಡಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ವರ್ಡ್ ಟೇಬಲ್‌ನ ಕಾಲಂಗಳ ಅಗಲ ಅಥವಾ ವಿಸ್ತಾರವನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ತಾನೇ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ. ನಿಮಗೆ ಟೇಬಲ್‌ನ ಅಗಲವನ್ನು ಭಿನ್ನವಾಗಿ ಇರಿಸಬೇಕಾದರೆ ಈ ಕೆಳಗಿನ ಯಾವುದಾದರೂ ಒಂದು ಆಯ್ಕೆಯನ್ನು ಮಾಡಿಕೊಳ್ಳಿರಿ.

ಈ ರೀತಿ ಮಾಡಬೇಕಾದರೆ	ಈ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ
ಎಲ್ಲ ಕಾಲಂಗಳಿಗೆ ಒಂದು ಅಗಲವನ್ನು ಸೂಚಿಸಲು	Fixed column width ಬಾಕ್ಸ್‌ನಲ್ಲಿ , ನಿಮಗೆ ಬೇಕಾದ ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಟೈಪ್ ಮಾಡಿರಿ.
ಪ್ರತಿ ಕಾಲಂನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್‌ನ ಗಾತ್ರವನ್ನು ಹೊಂದಿಸಲು ಕಾಲಂನ ಗಾತ್ರವನ್ನು ಬದಲಾವಣೆ ಮಾಡಲು	AutoFit to contents
ಟೇಬಲ್‌ನಲ್ಲಿ ಈಗಾಗಲೇ ಇರುವ ಗಾತ್ರದಲ್ಲಿ ಬದಲಾವಣೆಯಾದರೆ ಟೇಬಲ್‌ನ ಗಾತ್ರವನ್ನು ಬದಲಾವಣೆ ಮಾಡಲು	AutoFit to window

2 Seperate Text ನಲ್ಲಿ , ನೀವು ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಿದ ಪ್ರತ್ಯೇಕಿಸುವ ಅಕ್ಷರವನ್ನು (seperator character) ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

3 OK ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಈಗ ಟೆಕ್ಸ್ಟ್ ಈ ಕೆಳಗಿನಂತೆ ಟೇಬಲ್‌ಗೆ ಬದಲಾಗಿರುತ್ತದೆ.

Red, yellow	blue, green	orange, purple
Red, yellow	blue, green	orange, purple

ಟೇಬಲನ್ನು ವಿದಳನಗೊಳಿಸು ಅಥವಾ ವಿಭಜಿಸು (Split a table)

1 ನೀವು ಯಾವ ರೋ ಅನ್ನು ನಿಮ್ಮ ಎರಡನೇ ಟೇಬಲ್‌ನ ಮೊದಲನೇ ರೋ ಆಗಿ ಮಾಡಬಯಸುತ್ತೀರೊ ಆ ರೋ ಮೇಲೆ ಕರ್ಸರ್‌ನ್ನು ಇರಿಸಿ. ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ ಕರ್ಸರ್ ಟೇಬಲ್‌ನ ಮೂರನೇ ರೋ ನಲ್ಲಿದೆ.

2 Layout ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Merge ಗ್ರೂಪ್‌ನಲ್ಲಿ , Split Table ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ.

© NIMI NOT TO BE REPUBLISHED

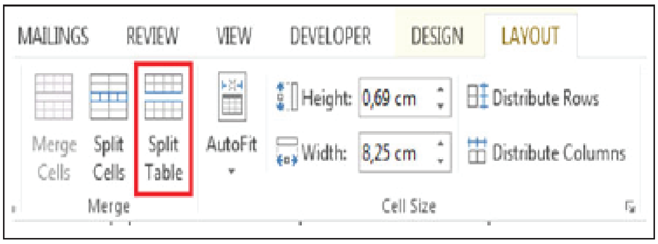
ನೀವು ವರ್ಡ್‌ನಲ್ಲಿ ಒಂದು ಟೇಬಲ್‌ನ್ನು ಎರಡು ಅಥವಾ ಎರಡಕ್ಕಿಂತಲೂ ಹೆಚ್ಚಿನ ಟೇಬಲ್‌ಗಳನ್ನಾಗಿ ವಿಭಜಿಸಬಹುದು. ಈ ರೀತಿಯಾಗಿ ನೀವು ಚಿಕ್ಕ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸಬಹುದು ಅಥವಾ ಎರಡು ಟೇಬಲ್‌ಗಳ ನಡುವೆ ಡಾಟಾವನ್ನು ಸೇರಿಸಬಹುದು.

1 ನೀವು ಯಾವ ರೋ ಅನ್ನು ನಿಮ್ಮ ಎರಡನೇ ಟೇಬಲ್‌ನ ಮೊದಲನೇ ರೋ ಆಗಿ ಮಾಡಬಯಸುತ್ತೀರೊ ಆ ರೋ ಮೇಲೆ ಕರ್ಸರ್‌ನ್ನು ಇರಿಸಿ. ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ ಕರ್ಸರ್ ಟೇಬಲ್‌ನ ಮೂರನೇ ರೋ ನಲ್ಲಿದೆ.

1	A
2	B
3	C
4	D

ನೀವು ಟೇಬಲ್‌ನ ಒಳಗೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, ರಿಬ್ಬನ್‌ನಲ್ಲಿ ಎರಡು ಹೊಸ Table ಟೂಲ್ಸ್ ಟ್ಯಾಬ್‌ಗಳು ಕಾಣಿಸುತ್ತವೆ. DESIGN ಮತ್ತು LAYOUT. ಈ ಟೂಲ್‌ಗಳು ನೀವು ಟೇಬಲ್‌ನ ಒಳಗಡೆ ಇದ್ದಾಗ ಮಾತ್ರ ಕಂಡು ಬರುತ್ತವೆ, ಉದಾಹರಣೆಗೆ, ನೀವು ಸೆಲ್‌ಗಳಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಸೇರಿಸುವಾಗ ಇವು ಕಂಡು ಬರುತ್ತವೆ.

2. LAYOUT ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Merge ಗ್ರೂಪ್‌ನಲ್ಲಿ, Split Table ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.



ಈಗ ಒಂದು ಟೇಬಲ್ ಎರಡು ಟೇಬಲ್‌ಗಳಾಗಿ ವಿಭಜನೆ ಹೊಂದುತ್ತದೆ.

ನೀವು ಒಂದು ಟೇಬಲ್‌ನ್ನು ಅದರಲ್ಲಿರುವ ರೋಗಳ ಸಂಖ್ಯೆಗಳನ್ನು ಆಧರಿಸಿ ಇನ್ನೂ ಹೆಚ್ಚಿನ ಟೇಬಲ್‌ಗಳನ್ನಾಗಿ ವಿಭಜಿಸಬಹುದು.

1	A
2	B

3	C
4	D

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ರೆಫರನ್ಸ್ ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Create and Manage References)

ರೆಫರನ್ಸ್ ಅಂಶಗಳನ್ನು ಮತ್ತು ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Create and manage reference elements and tables)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ರೆಫರನ್ಸ್ ಅಂಶಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು**
- **ಟೇಬಲ್‌ನಲ್ಲಿರುವ ವಿಷಯಗಳನ್ನು ನಿಮಗೆ ಬೇಕಾದ ಹಾಗೆ ಮಾರ್ಪಡಿಸುವುದು**

ವರ್ಡ್ ತನ್ನದೇ ಆದ , ಮೂಲ ರೆಫರನ್ಸಿಂಗ್ ವ್ಯವಸ್ಥೆಯನ್ನು ಹೊಂದಿದೆ. ಇದು ರಿಬ್ಬನ್‌ನ Reference ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುತ್ತದೆ. ಇದು ನಿಮಗೆ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಸೇರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ನಂತರ ಟೆಕ್ಸ್ಟ್‌ನ ಕೊನೆಗೆ bibliography ರಚಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಮಾಸ್ಟರ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗಿರುತ್ತದೆ. ಇವುಗಳನ್ನು ಮುಂದೆ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಸೇರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

> **ಸೂಚನೆ: ಈ ವ್ಯವಸ್ಥೆಯು EndNote, Mendeley ಅಥವಾ Zotero ನಷ್ಟು ಶಕ್ತಿಯುತವಾಗಿಲ್ಲ.**

ಮೊದಲನೆ ಹಂತಗಳು

1 ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತಯಾರಿಸಿ.

2 ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ಸೈಟೇಷನ್(Citation) ಸೇರಿಸಲು ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ References ಟ್ಯಾಬ್‌ಗೆ ಹೋಗಿರಿ ಮತ್ತು Insert Citation ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ನಂತರ Add new source ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Type of source ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು ಬಾಕ್ಸ್‌ಗಳನ್ನು ತುಂಬಿರಿ.

4 ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಎಲ್ಲ ಸೈಟೇಷನ್‌ಗಳನ್ನು ಸೇರಿಸಿರಿ.

5 ನಿಮ್ಮ ಕೆಲಸ ಮುಗಿದ ನಂತರ, ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಕೊನೆಗೆ ಬನ್ನಿರಿ ಮತ್ತು Bibliography ಆಯ್ಕೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಈಗಾಗಲೇ ಅಲ್ಲಿರುವ ಪೂರ್ವರೂಪಿತ ಆಯ್ಕೆಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆಮಾಡಿಕೊಳ್ಳಿರಿ ಅಥವಾ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು bibliography ಅನ್ನು ಸೇರಿಸಿರಿ.

6 ನಿಮ್ಮ ರೆಫರನ್ಸ್‌ನ ಶೈಲಿಯನ್ನು, ಡೀಫಾಲ್ಟ್ ಆಗಿ ಇರುವ APA ಶೈಲಿಯಿಂದ (style) ಬದಲಾಯಿಸಲು Style ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು ಬೇರೆ style ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

> **ಸೂಚನೆ : ವರ್ಡ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುವ style ಗಳಲ್ಲಿ ನಿಮಗೆ ಅವಶ್ಯಕತೆ ಇರುವ style ಗಳು ಸರಿಹೊಂದದೆ ಇರ್ಬಹುದು.**

> **ಅವುಗಲನ್ನು ಎಚ್ಚರಿಕೆಯಿಂದ ಪರಿಶೀಲಿಸಿ ಮತ್ತು ಒಂದು ವೇಳೆ ಅವು ಹೊಂದಿಕೆಯಾಗದಿದ್ದರೆ ನೀವು ಬಿಬ್ಲಿಯೊಗ್ರಾಫಿಯನ್ನು ಮಾರ್ಪಾಡು**
>
> **ಮಾಡಬಹುದಾದ ಟೆಕ್ಸ್ಟ್‌ಗೆ ಬದಲಾಯಿಸಬಹುದು. ಇ ದನ್ನು ಮಾಡಲು ಬೇಕಾದ ಸೂಚನೆಗಳನ್ನು ಕೆಳಗಿನ ಮಾರ್ಗದರ್ಶಕದಲ್ಲಿ ನೀಡಲಾಗಿದೆ.**

ವರ್ಡ್‌ನಲ್ಲಿರುವ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನ ಹಾರ್ಡ್ ಡಿಸ್ಕ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಿ ಇಡಲಾಗಿರುತ್ತದೆ. ನೀವು ಇನ್ನೊಂದು ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಇವುಗಳ ಮೇಲೆ ಕೆಲಸ ಮಾಡಲು ಬಯಸಿದರೆ, ಆ ಫೈಲನ್ನು ಹುಡುಕಲು ಈ ಕೆಳಗಿನ ಕ್ರಮಗಳನ್ನು ಅನುಸರಿಸಿ :

- Start ಮೆನುವಿನಲ್ಲಿರುವ ಹುಡುಕುವ ಬಾಕ್ಸ್‌ನಲ್ಲಿ (Search box) % APPDATA % ಎಂದು ಟೈಪ್ ಮಾಡಿ.
- Roaming > Microsoft > Bibliography ಫೋಲ್ಡರ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ
- ರೆಫರನ್ಸ್‌ಗಳು XML ಫೈಲ್ ಸೋರ್ಸ್‌ಗಳಲ್ಲಿ ಸಂಗ್ರಹ ಆಗಿರುತ್ತವೆ.

ವರ್ಡ್‌ನಲ್ಲಿ ರೆಫರನ್ಸ್‌ಗಳು (References in Word)

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಸರಳವಾದ ಅಂತರ್‌ನಿರ್ಮಿತ ರೆಫರನ್ಸ್ ಫಂಕ್ಷನ್‌ನ್ನು ಹೊಂದಿದೆ. ಇವು ರೆಫರನ್ಸ್ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುತ್ತವೆ. ಇಲ್ಲಿ ನೀವು ಸ್ವತಃ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ನಮೂದಿಸಬಹುದು ಮತ್ತು ಅವುಗಲನ್ನು ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ನೋಡಬಹುದು ಮತ್ತು ಹೆಚ್ಚಿನ ಸ್ಟೈಲ್‌ಗಳಲ್ಲಿ ಬಿಬ್ಲಿಯೊಗ್ರಾಫಿ ರಚಿಸಬಹುದು (APA, Harvard, MLA ಇತ್ಯಾದಿ. ನೀವು ಸೀಮಿತ ಸಂಖ್ಯೆಯ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ ಈ ಅಂತರ್ನಿರ್ಮಿತ ಫಂಕ್ಷನ್ ಬಹಳ ಒಳ್ಳೆಯದು ಮತ್ತು ಬೇರೆ ವಿವಿಧ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ನಿಮ್ಮ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ ರೆಫರನ್ಸ್‌ಗೆ ಹೋಗಿರಿ. ಎಡಬದಿಯಲ್ಲಿರುವ Table

© NIMI NOT TO BE REPUBLISHED

of Contents ಗ್ರೂಪ್‌ನ ಅಡಿಯಲ್ಲಿರುವ Table of Contents ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ನೀವು ಡೀಫಾಲ್ಟ್ ಆಗಿರುವ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಬಹುದು ಅಥವಾ ಮೆನುವಿನ ಕೆಳಭಾಗದಲ್ಲಿರುವ Custom Table of Contents ಮೇಲೆಯೂ ಕ್ಲಿಕ್ ಮಾಡಬಹುದು. ಮೆನುವಿನಲ್ಲಿರುವ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಂಡು Custom Table of Contents ನ್ನು ಸೇರಿಸಿರಿ.

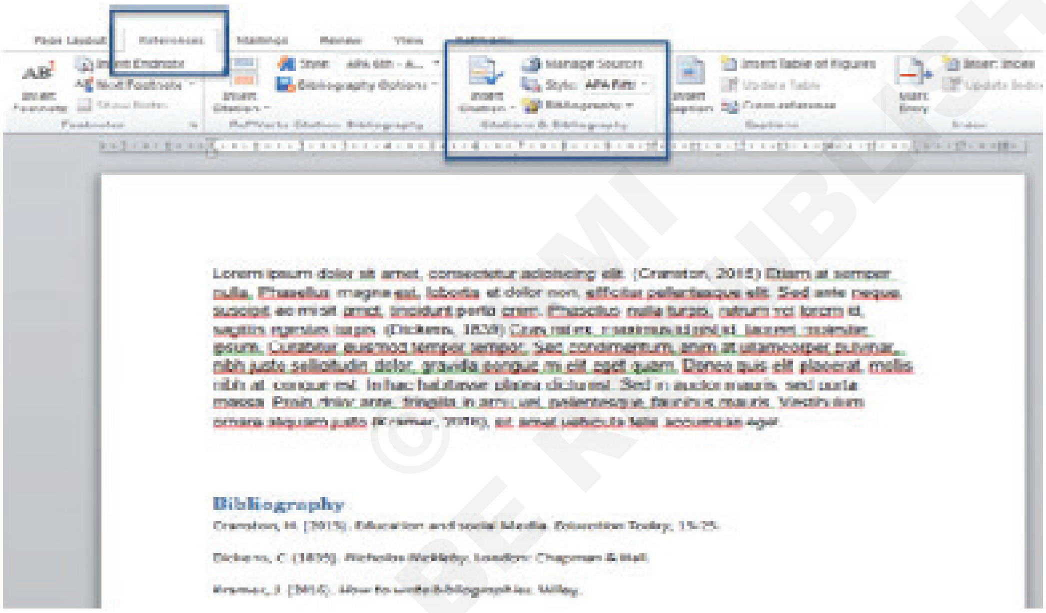
ವರ್ಡ್‌ನಲ್ಲಿ ಟೇಬಲ್ ಆಫ್ ಕಂಟೆಂಟ್ಸ್ ರಚಿಸಿದ ನಂತರ, ಅದು ಕಾಣುವ ರೀತಿಯನ್ನು ನೀವು ಬದಲಾಯಿಸಬಹುದು. ನಿಮ್ಮ ಟೇಬಲ್ ಆಫ್ ಕಂಟೆಂಟ್ಸ್‌ನ್ನು ಬದಲಾಯಿಸುವುದರಿಂದ ಈಗಾಗಲೇ ಇರುವ ನಿಮ್ಮ ಟೇಬಲ್‌ಗೆ ಪ್ರೆಫರನ್ಸ್‌ಗಳನ್ನು (preferences) ಗಳನ್ನು ಅನ್ವಯಿಸಲಾಗುತ್ತದೆ. OK ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಇದು ನಿಮಗೆ ಇಷ್ಟವಾಗದಿದ್ದರೆ, Cancel ಆಯ್ದುಕೊಳ್ಳಿರಿ. ಇದರಿಂದ ನೀವು ಮಾಡಿದ ಎಲ್ಲ ಬದಲಾವಣೆಗಳು ರದ್ದುಗೊಳಿಸಲ್ಪಡುತ್ತವೆ. ನಿಮ್ಮ ಮೂಲ Table of Contents ಹಾಗೆ ಉಳಿಯುತ್ತದೆ.

ಈಗಾಗಲೇ ಇರುವ Table of contents ನ್ನು ಬದಲಾಯಿಸಲು :

1 References > Table of Contents ಗೆ ಹೋಗಿರಿ.

2 Custom table of contents ಆಯ್ಕೆ ಮಾಡಿರಿ.

3 ಅಲ್ಲಿರುವ show, hide ಮತ್ತು align page numbers, add ಅಥವಾ change the tab leader set formats ಮತ್ತು ಹೆಡಿಂಗ್‌ನ ಎಷ್ಟುಹಂತಗಳು ಎಂದು ಸೂಚಿಸುವುದು ಮುಂತಾದ ಆಯ್ಕೆಗಳನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ. ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ Custom table of contents ನೋಡಿರಿ.



© NIMI NOT TO BE REPUBLISHED

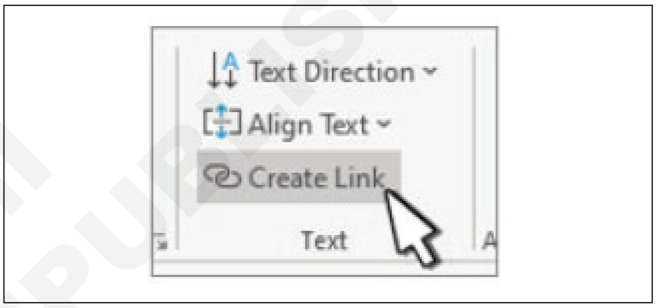
ಇಲ್ಲಸ್ಟ್ರೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Insert, format illustrations and text boxes)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್
- ಗ್ರಾಫಿಕ್ ಅಂಶಗಳಿಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು

ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ ಎಂದರೆ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ (Object). ಇದನ್ನು ನೀವು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಳವಡಿಸಬಹುದು. ಇದನ್ನು ನೀವು ನಿಮ್ಮ ಫೈಲ್‌ನ ಯಾವುದೇ ಸ್ಥಳದಲ್ಲಿ ಸೇರಿಸಿ ಅದರಲ್ಲಿ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಬಹುದು. ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ಗಳು ಯಾವುದೇ ನಿರ್ದಿಷ್ಟವಾದ ಟೆಕ್ಸ್ಟ್‌ನ ಬಗ್ಗೆ ಗಮನ ಸೆಳೆಯಲು ಉಪಯುಕ್ತವಾಗಿವೆ. ಮತ್ತು ನೀವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಯಾವುದೇ ಭಾಗದಲ್ಲಿ ಚಲಿಸಲು ಅಥವಾ ಸ್ಥಳಾಂತರಿಸಲು ಸಹಾಯಕವಾಗಿವೆ.

Insert > Text Box, ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ಈಗಾಗಲೆ ಪಟ್ಟಿಯಲ್ಲಿರುವ ಪೂರ್ವರೂಪಿತವಾಗಿರುವ ಯಾವುದಾದರೂ ಒಂದು ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಅಥವಾ office. com ನಲ್ಲಿ ಹೆಚ್ಚಿನ ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ಗಳು ಲಭ್ಯವಿರುತ್ತವೆ. ಅಥವಾ Draw Text Box ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಒಂದು ವೇಳೆ ನೀವು Draw Text Box ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡರೆ, ಡಾಕ್ಯುಮೆಂಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಗಾತ್ರದ ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ ಚಿತ್ರವನ್ನು ಬರೆಯಿರಿ.

- ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನಲ್ಲಿರುವ ಟೆಕ್ಸ್ಟ್‌ನ ರೂಪವನ್ನು ಬದಲಾಯಿಸಲು, ಅದರಲ್ಲಿರುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು Home ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ Font ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಆಯ್ಕೆಗಳನ್ನು ಉಪಯೋಗಿಸಿರಿ.
- ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ ರೂಪವನ್ನು ಬದಲಾಯಿಸಲು, Format contextual ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿರಿ. ಈ ಟ್ಯಾಬ್ ನೀವು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡಾಗ ಕಂಡುಬರುವ Drawing Tools ಅಡಿಯಲ್ಲಿ ಬರುತ್ತದೆ.
- ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ ಸ್ಥಾನವನ್ನು ಬದಲಾಯಿಸಲು, ಅದರ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು ಪಾಯಿಂಟರ್ + ಆಗಿ ಬದಲಾದಾಗ ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಡ್ರ್ಯಾಗ್ ಮಾಡಿ ಹೊಸಸ್ಥಳಕ್ಕೆ ಬದಲಾಯಿಸಿ.
- ನೀವು ಟೆಕ್ಸ್ ಬಾಕ್ಸ್‌ನ ಬಾರ್ಡರ್‌ನ್ನು ಬದಲಾಯಿಸಬಹುದು ಅಥವಾ ತೆಗೆದುಹಾಕಬಹುದು.
- ನೀವು ಒಂದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚಿನ ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಅವುಗಳನ್ನು ಒಂದಕ್ಕೊಂದು ಸಂಪರ್ಕ ಮಾಡಬಹುದು. ಇದರಿಂದ ಟೆಕ್ಸ್ಟ್ ಒಂದು ಬಾಕ್ಸ್‌ನಿಂದ ಇನ್ನೊಂದು ಬಾಕ್ಸ್‌ಗೆ ಪ್ರವಹಿಸಬಹುದು. ಒಂದು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು Format ಟ್ಯಾಬ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ, Drawing Toolx ಅಡಿಯಲ್ಲಿ Text ಗ್ರೂಪ್ ಬರುತ್ತದೆ. ಇದರಲ್ಲಿ ಇರುವ Create Link ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ.



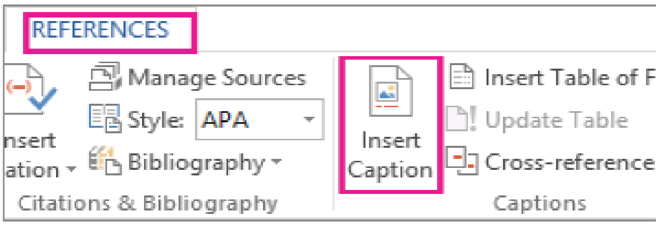
ಚಿತ್ರಕ್ಕೆ ಒಂದು ಶೀರ್ಷಿಕೆ ನೀಡಿ (Insert a caption for a picture)

ವರ್ಡ್‌ನಲ್ಲಿರುವ Insert Caption ವೈಶಿಷ್ಟ್ಯವು ಡಾಕ್ಯುಮೆಂಟ್ ನಲ್ಲಿರುವ ಚಿತ್ರಗಳಿಗೆ ವ್ಯವಸ್ಥಿತವಾಗಿ ಶೀರ್ಷಿಕೆ ನೀಡಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ

ಬೇರೆ ಆಫೀಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ, ಉದಾಹರಣೆಗೆ ಪಾವರ್‌ಪಾಯಿಂಟ್‌ನಲ್ಲಿ, ನೀವು ಚಿತ್ರದ ಸಮೀಪ ಒಂದು ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಸೇರಿಸುತ್ತೀರಿ ಮತ್ತು ನಂತರ ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಮತ್ತು ಚಿತ್ರವನ್ನು ಗ್ರೂಪ್ ಮಾಡುತ್ತೀರಿ. ಕೆಳಗೆ ನೀಡಲಾದ ಸೂಚನೆಗಳನ್ನು ನೋಡಿರಿ. ಒಂದು ವೇಳೆ ನೀವು ಒಂದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚಿನ ಚಿತ್ರಗಳ ಸರಣಿಯನ್ನು ಹೊಂದಿದ್ದರೆ, ನೀವೇ ಸ್ವತಃ ಅವುಗಳಿಗೆ ಸಂಖ್ಯೆಗಳನ್ನು ನೀಡಬೇಕಾಗುತ್ತದೆ.

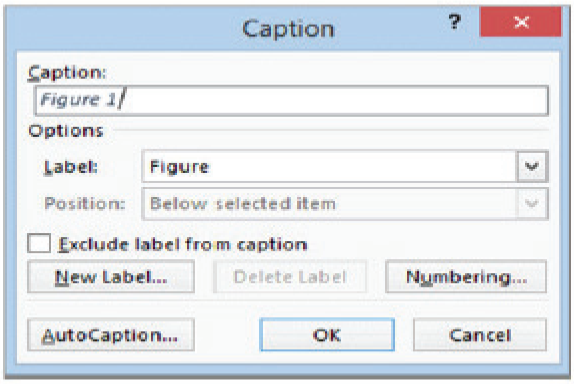
1 ನೀವು ಶೀರ್ಷಿಕೆ ನೀಡಬೇಕಾದ ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 References > Insert Caption ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

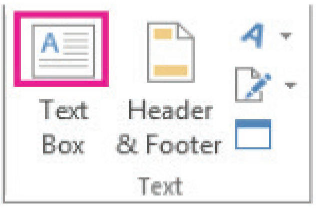


© NIMI NOT TO BE REPUBLISHED

3 ಡೀಫಾಲ್ಟ್ ಆಗಿ ಇರುವ ಲೇಬಲ್ (Figure) ಅನ್ನು ಉಪಯೋಗಿಸಲು, Caption ಬಾಕ್ಸ್‌ನಲ್ಲಿ ನಿಮ್ಮ ಕ್ಯಾಪ್ಷನ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ.



ಸೂಚನೆ : ನೀವು ನಿಮ್ಮ ಸ್ವಂತ ಡೀಫಾಲ್ಟ್ ಕ್ಯಾಪ್ಷನ್ ಲೇಬಲ್‌ನ್ನು ರಚಿಸಬಹುದು. ಇದಕ್ಕಾಗಿ ನೀವು New label ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ , ನಂತರ Label ಬಾಕ್ಸ್‌ನಲ್ಲಿ ನಿಮ್ಮ ಕ್ಯಾಪ್ಷನ್‌ನ್ನು ಸೇರಿಸಿರಿ. ವರ್ಡ್ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ನಿಮಗಾಗಿ ಹೊಸ ಲೇಬಲ್‌ನ್ನು ನೀಡುತ್ತದೆ.



ಇತರ ಆಫೀಸ್ ಆ್ಯಪ್‌ಗಳು (All other Office apps)

1 Insert > Text Box ಉಪಯೋಗಿಸಿ ನೀವು ಚಿತ್ರದ ಸಮೀಪ ಟೆಕ್ಸ್ ಬಾಕ್ಸ್‌ನ್ನು ಬರೆಯಬಹುದು.(ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ add a Text box ನೋಡಿರಿ)

2 ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ ಒಳಗೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ನೀವು ಕ್ಯಾಪ್ಷನ್ ನೀಡುವ ಸಲುವಾಗಿ ಬಳಸಲು ಬಯಸುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ.

3 ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. Home ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ Font ಆಯ್ಕೆಗಳನ್ನು ಬಳಸಿಕೊಂಡು ನಿಮಗೆ ಬೇಕಾದ style ನ್ನು ಕ್ಯಾಪ್ಷನ್‌ಗೆ

ನೀಡಿರಿ.

4 Ctrl + click ಉಪಯೋಗಿಸಿಕೊಂಡು ಚಿತ್ರ ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ನಂತರ Arrange ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ Picture Format ಟ್ಯಾಬ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ನಂತರ Group > Group ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಡಾಕ್ಯುಮೆಂಟ್ ಕೊಲ್ಯಾಬೊರೇಷನ್ ನಿರ್ವಹಿಸುವುದು (Manage Document Collaboration)

ಕಮೆಂಟ್, ಟ್ರ್ಯಾಕ್ ಬದಲಾವಣೆ ಹಾಗೂ ಮೇಲಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage comments change tracking and mailings)

ಉದ್ದೇಶಗಳು: ಈ ಪಾಠದ ಕೊನೆಗೆ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ವರ್ಡ್‌ನಲ್ಲಿ ಕೊಲ್ಯಾಬೋರೇಟ್
- ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು
- ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ಗಳನ್ನು ಆನ್ ಮತ್ತು ಆಫ್ ಮಾಡುವುದು.

ವರ್ಡ್‌ನಲ್ಲಿ ಕೊಲ್ಯಾಬೋರೇಟ್ (Collaborate in Word)

1 ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Share ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಅಥವಾ File > Share ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಸೂಚನೆ : ಒಂದು ವೇಳೆ ನಿಮ್ಮ ಫೈಲ್ OneDrive ನಲ್ಲಿ ಈಗಾಗಲೇ ಸಂಗ್ರಹವಾಗಿರದಿದ್ದಲ್ಲಿ ಶೇರ್ ಮಾಡುವ ಸಲುವಾಗಿ ಆ ಫೈಲನ್ನು ನೀವು OneDrive ಗೆ ಅಪ್‌ಲೋಡ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

2 ನೀವು ಯಾರಿಗೆ ಫೈಲನ್ನು ಹಂಚಿಕೊಳ್ಳುವವರಿದ್ದೀರೋ ಅವರ ಹೆಸರನ್ನು ಡ್ರಾಪ್ ಡೌನ್ ಲಿಸ್ಟ್‌ನಿಂದ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಅಥವಾ ಅವರ ಹೆಸರನ್ನು ನಮೂದಿಸಿ ಅಥವಾ ಇ ಮೇಲ್ ಅಡ್ರೆಸ್‌ನ್ನು ನಮೂದಿಸಿರಿ.

3 ಸಂದೇಶವನ್ನು (ಐಚ್ಛಿಕ್ಕ) ಟೈಪ್ ಮಾಡಿ. ಮತ್ತು Send ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಹಂಚಿಕೊಳ್ಳಿರಿ (Share your document)

ವರ್ಡ್‌ನಿಂದ ಒಂದು ಫೈಲನ್ನು ಹಂಚಿಕೊಳ್ಳಲು (share)

1 ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Share ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಅಥವಾ File > Share ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

> ಸೂಚನೆ : ಒಂದು ವೇಳೆ ನಿಮ್ಮ ಫೈಲ್ OneDrive ನಲ ಈಗಾಗಲೇ ಸಂಗ್ರಹವಾಗಿರದಿದ್ದಲ್ಲಿ ಶೇರ್ ಮಾಡುವ ಸಲುವಾಗಿ ಆ ಫೈಲನ್ನು ನೀವು OneDrive ಗೆ ಅಪ್‌ಲೋಡ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

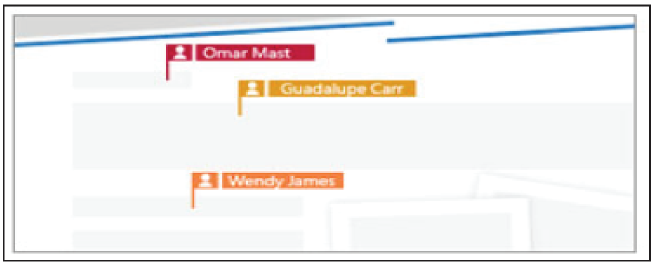
2 ನೀವು ಯಾರಿಗೆ ಫೈಲನ್ನು ಹಂಚಿಕೊಳ್ಳುವವರಿದ್ದೀರೋ ಅವರ ಹೆಸರನ್ನು ಡ್ರಾಪ್ ಡೌನ್ ಲಿಸ್ಟ್‌ನಿಂದ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಅಥವಾ ಅವರ ಹೆಸರನ್ನು ನಮೂದಿಸಿ ಅಥವಾ ಇ ಮೇಲ್ ಅಡ್ರೆಸ್‌ನ್ನು ನಮೂದಿಸಿರಿ.

3 ಸಂದೇಶವನ್ನು (ಐಚ್ಛಿಕ) ಟೈಪ್ ಮಾಡಿ. ಮತ್ತು Send ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಕೋ ಎಡಿಟ್ ಮಾಡುವುದು (Co-edit a document)

ನೀವು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಶೇರ್ ಮಾಡಿದ ಮೇಲೆ, ಆ ಫೈಲ್ ಮೇಲೆ ಬೇರೆಯವರೊಂದಿಗೆ ಅದೇ ಸಮಯದಲ್ಲಿ ಕೆಲಸ ಮಾಡಬಹುದು.

- ಅತ್ಯುತ್ತಮ ಅನುಭವ ಪಡೆದುಕೊಳ್ಳಲು, ವರ್ಡ್‌ನಲ್ಲಿ ವೆಬ್ ಕೆಲಸಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಒಟ್ಟಿಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸಿ ಮತ್ತು ಅದೇ ಸಮಯದಲ್ಲಿ ನಡೆಯುವ ಬದಲಾವಣೆಗಳನ್ನು ನೋಡಿರಿ.
- Share ನ ಅಡಿಯಲ್ಲಿ, ನೀವು ಮತ್ತೆ ಯಾರು ಆ ಫೈಲನ್ನು ಎಡಿಟ್ ಮಾಡುತ್ತಿದ್ದಾರೆ ಅವರ ಹೆಸರನ್ನು ನೋಡಬಹುದು.
- ಬಣ್ಣದ ಫ್ಲಾಗ್‌ಗಳು ನಿಮಗೆ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಪ್ರತಿ ವ್ಯಕ್ತಿಯೂ ನಿಖರವಾಗಿ ಎಲ್ಲೆಲ್ಲಿ ಕೆಲಸ ಮಾಡುತ್ತಿದ್ದಾನೆ ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತವೆ.



© NIMI NOT TO BE REPUBLISHED

ಟ್ರ್ಯಾಕ್ ಮತ್ತು ರಿವಿವ್ ಬದಲಾವಣೆಗಳು (Track and review changes)

1 ಟ್ರ್ಯಾಕ್ ಬದಲಾವಣೆ ಮಾಡಲು, Review > Track Changes ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 ಬದಲಾವಣೆಗಳನ್ನು ರಿವಿವ್ ಮಾಡಲು, ಬದಲಾವಣೆಗೂ ಮೊದಲು ಕರ್ಸರನ್ನು ಇರಿಸಿ ಮತ್ತು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ :

- Accept to keep the change ಅಥವಾ
- Reject to remove it

ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ಗಳನ್ನು ಆನ್ ಮತ್ತು ಆಫ್ ಮಾಡುವುದು (Turn Track Changes on and off)

Review ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ Track Change. ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

- Track Changes ON ಇದ್ದಾಗ, ಸೆಕ್ಷನ್ ಹೈಲೈಟ್ ಆಗಿರುತ್ತದೆ. ಡಿಲೀಟ್ ಆದಂತವುಗಳನ್ನು strikethrough ದಿಂದ ಮಾರ್ಕ್ ಮಾಡಿರಲಾಗುತ್ತದೆ. ಮತ್ತು ಸೇರಿಸಿದವುಗಳನ್ನು underline ಮೂಲಕ ಮಾರ್ಕ್ ಮಾಡಲಾಗಿರುತ್ತದೆ. ವಿಭಿನ್ನ ಆಥರ್‌ಗಳು ಮಾಡುವ ಬದಲಾವಣೆಗಳು ವಿಭಿನ್ನ ಕಲರ್‌ಗಳಿಂದ ಸೂಚಿಸಲಾಗುತ್ತದೆ.
- Track Changes OFF ಇದ್ದಾಗ, ಸೆಕ್ಷನ್ ಹೈಲೈಟ್ ಆಗಿರುವುದಿಲ್ಲ. ವರ್ಡ್ ಬದಲಾವಣೆಗಳನ್ನು ಮಾರ್ಕ್ ಮಾಡುವುದನ್ನು ನಿಲ್ಲಿಸುತ್ತದೆ, ಆದರೆ ಕಲರ್ ಮಾಡಲ್ಪಟ್ಟ underline ಗಳು ಮತ್ತು strikethrough ಗಳು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಹಾಗೇ ಇರುತ್ತವೆ.

ಸೂಚನೆ : ನೀವೂ ಕೂಡ ಸ್ಟೇಟಸ್ ಬಾರ್‌ಗೆ Track Changes ಇಂಡಿಕೇಟರ್‌ನ್ನು ಸೇರಿಸಬಹುದು. ಸ್ಟೇಟಸ್ ಬಾರ್ ಮೇಲೆ Right - Click ಮಾಡಿರಿ ಮತ್ತು Track Changes ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

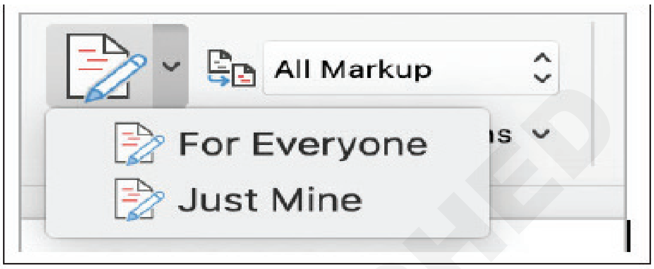
ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ಗಳನ್ನು ನೋಡುವುದು (View tracked changes)

> **ಸೂಚನೆ : ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್ ಎಡಿಟಿಂಗ್ ಅಥವಾ ರಿವಿವಿಂಗ್ ಮೋಡ್‌ನಲ್ಲಿ ಇದ್ದಾಗ, ನೀವು ಕೊಲ್ಯಾಬೊರೇಟರ್‌ನ ಪೂರ್ಣ, ಸಲಹೆ ನೀಡಲಾದ ಬದಲಾವಣೆಯನ್ನು ನೋಡಲು** tracked changes **ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ನಿಮ್ಮ ಕೊನೆಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಯಾವ ಕಾರ್ಯಗಳು ನಡೆಯುತ್ತವೆ ಎಂದು ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡಲು** Accepr **ಅಥವಾ** Reject **ಬಟನ್ ಉಪಯೋಗಿಸಿರಿ.**

ಯಾರ ಬದಲಾವಣೆಗಳನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡುವುದು ಎಂದು ಆಯ್ಕೆ ಮಾಡುವುದು (Choose whose changes to track)

ನೀವು ನಿಮ್ಮ ಸ್ವಂತ ಬದಲಾವಣೆಗಳನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡಬಹುದು ಅಥವಾ ಪ್ರತಿಯೊಬ್ಬರ ಬದಲಾವಣೆಗಳನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡಬಹುದು.

- ನಿಮ್ಮ ಬದಲಾವಣೆಗಳನ್ನು ಮಾತ್ರ ಟ್ರ್ಯಾಕ್ ಮಾಡಲು - Review Tab ಗೆ ಹೋಗಿ, Track Changes > Just Mine ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- ಎಲ್ಲರ ಬದಲಾವಣೆಗಳನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡಲು - Review Tab ಗೆ ಹೋಗಿ, Track Changes > For Everyone ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.



ಸೂಚನೆ : ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ನ್ನು ಬೇರೆಯವರು ಟರ್ನ್ ಆಫ್ ಮಾಡದೇ ಇರಲು, Review ಟ್ಯಾಬ್‌ಗೆ ಹೋಗಿ . Track Changes > Lock Tracking ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಬದಲಾವಣೆಗಳನ್ನು ನೀವು ಯಾವ ರೀತಿ ನೋಡಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

ನೀವು ನೋಡಬಯಸುವ ಮಾರ್ಕಪ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

1 Review Tab ನಲ್ಲಿರುವ Tracking ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 ಡ್ರಾಪ್ ಡೌನ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಮಾರ್ಕಪ್‌ಗಳನ್ನು ಪ್ರದರ್ಶಿಸಲು All Markup ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

3 ಈ ಕೆಳಗಿನ ಆಯ್ಕೆಗಳಲ್ಲಿ ಒಂದನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

- Simple Markup ಇದು ಮಾರ್ಜಿನ್‌ನಲ್ಲಿ ಟ್ರ್ಯಾಕ್ ಮಾಡಲಾದ ಬದಲಾವಣೆಗಳನ್ನು (track changes) ಕೆಂಪು ಸಾಲಿನಿಂದ ತೋರಿಸುತ್ತದೆ.
- All Markup ಇದು ಟ್ರ್ಯಾಕ್ ಮಾಡಲಾದ ಬದಲಾವಣೆಗಳನ್ನು (track changes) ಪ್ರತಿಯೊಬ್ಬ ರಿವಿವರ್‌ಗೆ ವಿಭಿನ್ನ ಬಣ್ಣದ ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಸಾಲುಗಳಿಂದ ತೋರಿಸುತ್ತದೆ.
- No Markup ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಮಾಡಲಾದ ಬದಲಾವಣೆಗಳನ್ನು ಪ್ರದರ್ಶಿಸುವುದನ್ನು ಮರೆ ಮಾಡುತ್ತದೆ.
- Original ಇದು ಯಾವುದೇ ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್ ಇಲ್ಲದ ಮತ್ತು ಕಮೆಂಟ್‌ಗಳು ಇಲ್ಲದ ಮೂಲ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ. ಅದಾಗ್ಯೂ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿರುವ Accept ಆದ , Reject ಆದ ಅಥವಾ ಡಿಲೀಟ್ ಆದ ಯಾವುದೇ ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ಗಳು ಮತ್ತು ಕಮೆಂಟ್‌ಗಳು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿಯೇ ಉಳಿಯುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ಗಳು ಪ್ರದರ್ಶಿಸಲ್ಪಡುವ ರೀತಿಯನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.(Choose the way track changes display)

ರಿವಿಷನ್‌ಗಳು ಯಾವ ರೀತಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡಬೇಕು ಎಂಬುದನ್ನು ನೀವು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ಒಂದು ವೇಳೆ ನೀವು ರಿವಿಷನ್‌ಗಳು balloons ನಂತೆ ಕಾಣಬೇಕು ಎಂದು ಬಯಸಿದರೆ, ಅವು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಮಾರ್ಜಿನ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತವೆ. ಒಂದು ವೇಳೆ ನೀವು ಅವುಗಳು ನೇರವಾಗಿ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಇನ್‌ಲೈನ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡಬೇಕೆಂದು ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡರೆ, ಇನ್ ಲೈನ್ ರಿವಿಷನ್‌ಗಳು ಎಲ್ಲ ಡಿಲೀಟ್ ಆದಂತಹವುಗಳನ್ನು balloon ಗಳ ಬದಲಾಗಿ strikethrough ಗಳೊಂದಿಗೆ ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

1 Review tab ನಲ್ಲಿ , Track > Show Markup ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳರಿ

2 Balloons ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ ನಿಮಗೆ ಬೇಕಾದ ಪ್ರದರ್ಶನದ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

- Show Revisions in Balloons
- Show All Revisions Inline
- Show Only Formatting in Balloons

> ಸೂಚನೆ : ನೀವು tracked changes ನ್ನು balloons ಗಳಲ್ಲಿ ನೋಡಲು ಬಯಸಿದರೆ. ನೀವು Print Layout view ಅಥವಾ Web Layout view ನಲ್ಲಿ ಇರಬೇಕಾಗಿರುತ್ತದೆ.

(Display changes by type of edit)

1 Review ಟ್ಯಾಬ್‌ನಲ್ಲಿ , Track Changes > Showe Markup ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

2 ಎಡಿಟ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

- Insertions and Deletions
- Formatting

ಪ್ರತಿ ಆಯ್ಕೆಯ ಮುಂದಿರುವ ಚೆಕ್ ಮಾರ್ಕ್(check mark) ಆ ಆಯ್ಕೆಯು ಆರಿಸಲ್ಪಟ್ಟಿದೆ ಎಂದುದನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಸೂಚನೆ : ನೀವು Show Markup ಮೆನುವಿನಲ್ಲಿ ಮಾರ್ಕಪ್‌ನ ವಿಧವನ್ನು ಅಳಿಸುವುದರ ಮೂಲಕ ಮರೆ ಮಾಡಿದರೂ ಕೂಡ, ಪ್ರತಿ ಬಾರಿಯೂ ನೀವು ಅಥವಾ ರಿವಿವರ್ ಆ ದಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತೆರೆದಾಗ ಮಾರ್ಕಪ್ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

ರಿವಿವರ್ ಮೂಲಕ ಬದಲಾವಣೆಗಳನ್ನು ಪ್ರದರ್ಶಿಸುವುದು (Display changes by reviewer(s)

1 Review > Tracking , Show Markup ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 Specific People ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

3 ಆಯ್ಕೆ ಮಾಡಿದ ಎಲ್ಲ ಚೆಕ್ ಬಾಕ್ಸ್‌ಗಳನ್ನು ಕ್ಲಿಯರ್ ಮಾಡಿರಿ(ತೆಗೆದುಹಾಕಿರಿ) ಯಾವ ರಿವಿವರ್ ಬದಲಾವಣೆಗಳನ್ನು ನೀವು ತೋರಿಸಬಯಸುತ್ತಿರೋ ಅವರ ಹೆಸರನ್ನು ಮಾತ್ರ ಉಳಿಸಿ

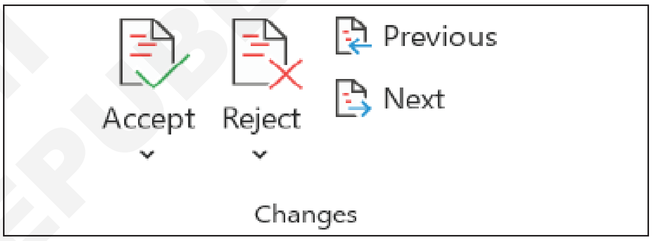
> ಸೂಚನೆ : ಪಟ್ಟಿಯಲ್ಲಿರುವ ಎಲ್ಲ ರಿವಿವರ್‌ಗಳ ಎಲ್ಲ ಚೆಕ್‌ಬಾಕ್ಸ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ಅಥವಾ ಕ್ಲಿಯರ್ ಮಾಡಲು, All Reviewers ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

Navigate tracked changes in Changes menu section

ನೀವು ಒಂದು ಟ್ರ್ಯಾಕ್ ಚೇಂಜ್‌ನಿಂದ ಇನ್ನೊಂದಕ್ಕೆ ಹೋಗಬೇಕಾದರೆ :

1 Review > Changes ನಲ್ಲಿ, Next ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡು ಮುಂದಿನ track change ಗೆ ಹೋಗಬಹುದು.

2 Review > Changes ನಲ್ಲಿ, Previous ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡು ಹಿಂದಿನ track change ಗೆ ಹೋಗಬಹುದು.



ಟ್ರ್ಯಾಕ್ ಮಾಡಲಾದ ಬದಲಾವಣೆಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದು ಅಥವಾ ನಿರಾಕರಿಸುವುದು (Accept or Reject tracked changes)

ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ಅನೇಕ ರೀತಿಯಿಂದ ನಿರ್ವಹಿಸಬಹುದು. ಅತ್ಯಂತ ಜಾಣತನದ ವಿಧಾನವೆಂದರೆ, ಕಮೆಂಟರನ್ನು ಗುರುತಿಸುವುದು ಮತ್ತು ಕಾರ್ಡ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುವ ಮೂಲಕ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗೆ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡುವುದು. ಕಾರ್ಡ್ ಕಮೆಂಟರ್‌ನ ಹೆಸರನ್ನು ತೋರಿಸುತ್ತದೆ ಮತ್ತು ಅವರ ಸಲಹೆಯನ್ನು ಸ್ವೀಕರಿಸಬೇಕೋ (Accept) ಅಥವಾ ನಿರಾಕರಿಸಬೇಕೋ (Reject) ಎನ್ನುವ ಆಯ್ಕೆಯನ್ನು ನೀಡುತ್ತದೆ.

ಮೇನ್ ಮೆನು ಉಪಯೋಗಿಸಿ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ನ್ನು Accept ಅಥವಾ reject ಮಾಡಿರಿ.

ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಪ್ರಾರಂಭದಿಂದ ಕೊನೆಯವರೆಗೆ ಸರಣಿಯಲ್ಲಿ ನೀವು ಪ್ರತಿಯೊಂದು ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ನ್ನು ಸರಣಿಯಲ್ಲಿ ಪರಿಹರಿಸಬಹುದು.

1 Review > Changes > Next ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

2 Accept ಅಥವಾ Reject ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ನಿಮ್ಮ ರಿವಿವ್ ಅಥವಾ ಕ್ರಮಕ್ಕಾಗಿ ಸರಣಿಯಲ್ಲಿರುವ ಮುಂದಿನ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್ ಹೈಲೈಟ್ ಆಗಿರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸೂಚನೆ : ಮೇನ್ ಮೆನುವಿನ ಡ್ರಾಪ್ ಡೌನ್ ಮೆನುಗಳನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಪ್ರತಿ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ನ್ನು accept ಅಥವಾ reject ಮಾಡಬಹುದು. ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ರಿವಿವ್ ಮಾಡಲು ಈ ವಿಧಾನವು ಒಂದು ಕ್ರಮಬದ್ಧವಾದ ವಿಧಾನವಾಗಿದೆ.

ಮೇನ್ ಮೆನು ಉಪಯೋಗಿಸಿ ಚೇಂಜ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದು (Accept) (Accept changes using the main menu)

1 Review > Changes ನಲ್ಲಿ, Accept ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

- Accept and Move to Next
- Accept This Change
- Accept All Changes
- Accept All Changes and Stop Tracking

ಮೇನ್ ಮೆನು ಉಪಯೋಗಿಸಿ ಚೇಂಜ್‌ಗಳನ್ನು ನಿರಾಕರಿಸುವುದು (Reject) (Reject changes using the main menu)

1 Review > Changes ನಲ್ಲಿ, Reject ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

- Reject and Move to Next
- Reject This Change
- Reject All Changes
- Reject All Changes and Stop Tracking

Right click ಉಪಯೋಗಿಸಿ ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿಯ ಚೇಂಜ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದು ಅಥವಾ ನಿರಾಕರಿಸುವುದು (Accept or reject changes within a document with a right click)

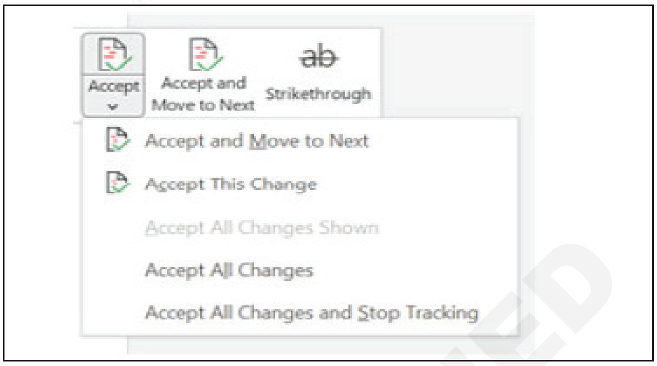
ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವಾಗ, ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್ ಮೇಲೆ right click ಮಾಡಿ ಆ ಚೇಂಜ್‌ನ್ನು accept ಅಥವಾ reject ಮಾಡಬಹುದು. Right click ವಿಧಾನವನ್ನು ಉಪಯೋಗಿಸಿದಾಗ ಒಂದು banner ಮೆನು ಆಯ್ಕೆ ಮತ್ತು ಒಂದು drop down ಆಯ್ಕೆ ಕಾಣುತ್ತವೆ.

banner ಆಯ್ಕೆಯನ್ನು Right click ಮಾಡಿ ಚೇಂಜ್‌ನ್ನು ಸ್ವೀಕರಿಸುವುದು (Accept changes in-line with the right-click banner option)

1 ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿರುವ ಒಂದೇ ಸಲಹೆಯನ್ನು ರಿವಿವ್ ಮಾಡಲು ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್ ಮೇಲೆ Right - click ಮಾಡಿರಿ.

2 banner ನಲ್ಲಿ ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

- Accept > Accept and Move to Next
- Accept > Accept This Change
- Accept > Accept All Changes
- Accept > Accept All Changes and Stop Tracking



Drop down ಮೆನು ಆಯ್ಕೆಯನ್ನು Right click ಮಾಡಿ ಚೇಂಜ್‌ನ್ನು ಸ್ವೀಕರಿಸುವುದು (Accept changes in-line with the right-click drop-down menu option)

1 ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿರುವ ಒಂದೇ ಸಲಹೆಯನ್ನು ರಿವಿವ್ ಮಾಡಲು ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್ ಮೇಲೆ Right - click ಮಾಡಿರಿ.

2 Drop down ಮೆನುವಿನಲ್ಲಿರುವ ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

- Accept Insertion (or Accept Deletion)
- Reject Insertion (or Reject Deletion)

ಮೇನ್ ಮೆನುವಿನಲ್ಲಿರುವ ಎಲ್ಲ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ಒಂದೇ ಬಾರಿಗೆ ಸ್ವೀಕರಿಸುವುದು ಅಥವಾ ನಿರಾಕರಿಸುವುದು (Accept or reject all tracked changes at once on the main menu)

ಎಲ್ಲ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದು

1 Review > Changes ನಲ್ಲಿ, Accept ಆಯ್ಕೆ ಮಾಡಿರಿ

2 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

- Accept All Changes Shown
- Accept All Changes
- Accept All Changes and Stop Tracking

ಎಲ್ಲ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ನಿರಾಕರಿಸುವುದು

1 Review > Changes ನಲ್ಲಿ, Reject ಆಯ್ಕೆ ಮಾಡಿರಿ

2 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

- Reject All Changes Shown

© NIMI NOT TO BE REPUBLISHED

- Reject All Changes
- Reject All Changes and Stop Tracking

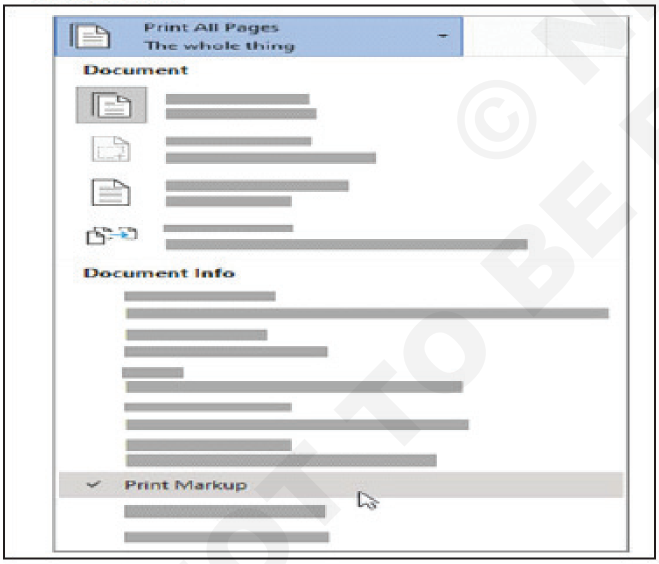
> **ಸೂಚನೆ :** Review > Tracking > Show Markup > Specific People > All Reviewers **ಆಯ್ಕೆ ಮಾಡಿದಾಗ,** Accept All Changes Shown **ಮತ್ತು** Reject All Changes Shown **ಈ ಆಯ್ಕೆಗಳು ಕಾಣುವುದಿಲ್ಲ.**

ಪ್ರಿಂಟ್ ಮಾಡುವಾಗ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ಮತ್ತು ಕಮೆಂಟ್‌ಗಳನ್ನು ಮರೆಮಾಡುವುದು (Hide tracked changes and comments when printing)

ಚೇಂಜ್‌ಗಳನ್ನು ಮರೆಮಾಡುವುದು ಎಂದರೆ (hide) ಅವುಗಳನ್ನು ಡಾಕ್ಯುಮೆಂಟ್‌ನಿಂದ ತೆಗೆದುಹಾಕುವುದಲ್ಲ. ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಿಂದ ಮಾರ್ಕಪ್‌ನ್ನು ತೆಗೆದುಹಾಕಲು Changes ಗ್ರೂಪ್‌ನಲ್ಲಿ ಇರುವ Accept ಮತ್ತು Reject ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

1 File > Print > Settings > Print All Pages ಆಯ್ಕೆ ಮಾಡಿರಿ

2 Documents Info ಅಡಿಯಲ್ಲಿ, ಚೆಕ್ ಮಾರ್ಕ್‌ನ್ನು ಕ್ಲಿಯರ್ ಮಾಡಲು Print markup ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.



ರಿವಿವಿಂಗ್ ಪೇನ್‌ನಲ್ಲಿ ನಿಮ್ಮ ಎಲ್ಲ ಚೇಂಜ್‌ಗಳನ್ನು ಸಾರಾಂಶದ ಪಟ್ಟಿಯಲ್ಲಿ ನೋಡುವುದು (View all your changes in a summary list on the Reviewing Pane)

1 Review > Tracking ನಲ್ಲಿ, Reviewing Pane ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 ನೀವು Reviewing Pane ಅನ್ನು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ನಂತರ ಅಥವಾ ಅದರ ಕೆಳಗೆ ನೋಡಲು ಬಯಸಿದರೆ,

- ಎಲ್ಲ Change ಗಳ ಪಟ್ಟಿಯನ್ನು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ನಂತರ ನೋಡಬೇಕಾದರೆ, Reviewing Pane Vertical ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.
- ಎಲ್ಲ Change ಗಳ ಪಟ್ಟಿಯನ್ನು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಕೆಳಗೆ ನೋಡಬೇಕಾದರೆ, Reviewing Pane Horizontal ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

ನೀವು Reviewing Pane ಅನ್ನು ಒಂದು ಸೂಕ್ತ ಸಾಧನವನ್ನಾಗಿ ಉಪಯೋಗಿಸಬಹುದು. ಇದು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಿಂದ ಎಲ್ಲ ಟ್ರ್ಯಾಕ್ಡ್ ಚೇಂಜ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಲಾಗಿದೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಏಕೆಂದರೆ ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ನೋಡುವ ಇತರರಿಗೆ ಅವು ಕಂಡು ಬರುತ್ತವೆ. Reviewing Pane ನ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ sammary ವಿಭಾಗ, ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಉಳಿದ tracked change ಗಳ ಮತ್ತು ಕಮೆಂಟ್‌ಗಳ ನಿಖರವಾದ ಸಂಖ್ಯೆಯನ್ನು ತೋರಿಸುತ್ತದೆ.

Reviewing Pane, ಕಮೆಂಟ್ ಬಬಲ್‌ನಲ್ಲಿ ಸರಿಯಾಗಿ ಹೊಂದಿಕೊಳ್ಳಲು ಆಗದ ದೀರ್ಘ ಕಮೆಂಟ್‌ಗಳನ್ನು ರಿವಿವ್ ಮಾಡಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

> **ಸೂಚನೆ : ಡಾಕ್ಯುಮೆಂಟ್ ಅಥವಾ ಕಮೆಂಟ್ ಬಲೂನ್‌ನಂತಲ್ಲದೇ** Reviewing Pane **ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಲು ಅತ್ಯುತ್ತಮ ಸಾಧನವಲ್ಲ.** Reviewing Pane **ನಲ್ಲಿ, ಟೆಕ್ಸ್ಟ್ ಅಥವಾ ಕಮೆಂಟ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುವ ಬದಲು ಅಥವಾ ಇತರ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುವ ಬದಲು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿಯೆ ಎಲ್ಲ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಿರಿ. ನಂತರ ಈ ಬದಲಾವಣೆಗಳು** Reviewing Pane **ನಲ್ಲಿ ಕಾಣುತ್ತವೆ.**

ಮುಖ್ಯ ಸೂಚನೆ : ಕಮೆಂಟ್‌ಗಳು ಈಗ Track Changes **ಫಂಕ್ಷನ್‌ನ ಭಾಗವಾಗಿಲ್ಲ.**

© NIMI NOT TO BE REPUBLISHED

ಮೇಲ್ ಮರ್ಜ್ ನಿರ್ವಹಿಸುವುದು (Perform mail merge)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು
- ಮೇಲ್ ಮರ್ಜ್
- ಟೆಂಪ್ಲೇಟ್‌ಗಳು.

ಆಬ್ಜೆಕ್ಟ್ , ಮ್ಯಾಕ್ರೊ , ಮೇಲ್ ಮರ್ಜ್, ಟೆಂಪ್ಲೇಟ್‌ಗಳು ಮತ್ತು ಇತರ ಸಾಧನಗಳೊಂದಿಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು.

ಆಬ್ಜೆಕ್ಟ್

ನೀವು ಉಪಯೋಗಿಸುತ್ತಿರುವ ವರ್ಡ್ ಅಥವಾ ಔಟ್‌ಲುಕ್‌ನ ಆವೃತ್ತಿಯ ಆಧಾರದ ಮೇಲೆ ವಿವಿಧ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು (PDF ಫೈಲ್‌ಗಳು, ಎಕ್ಸೆಲ್ ಚಾರ್ಟ್‌ಗಳು, ಅಥವಾ ವರ್ಕ್‌ಶೀಟ್‌ಗಳು ಅಥವಾ ಪವರ್‌ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳು) ನೀವು ವರ್ಡ್‌ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಥವಾ ಇ ಮೇಲ್ ಸಂದೇಶಗಳಲ್ಲಿ ಲಿಂಕ್ ಮಾಡುವ ಮೂಲಕ ಅಳವಡಿಸಬಹುದು. ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಸೇರಿಸಲು , Insert ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ Object ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಥವಾ ಇ ಮೇಲ್ ಸಂದೇಶದಲ್ಲಿ ಸೇರಿಸುವ ಹೊಸ ಫೈಲನ್ನು ರಚಿಸಲು :

1 Object ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Create New ಟ್ಯಾಬ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು Object type list ನಲ್ಲಿರುವ ಒಂದು ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

   ಆ ಫೈಲ್ ಟೈಪ್‌ಗಾಗಿ ಡೀಫಾಲ್ಟ್ ಪ್ರೊಗ್ರಾಂ ತೆರೆದುಕೊಳ್ಳುತ್ತದೆ. ಇದರಲ್ಲಿ ನೀವು ನಿಮಗೆ ಬೇಕಾದ ಯಾವುದೇ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಅಥವಾ ಡಾಟಾವನ್ನು ನಮೂದಿಸಬಹುದು. ನೀವು ಪ್ರೊಗ್ರಾಂನ್ನು ಮುಚ್ಚಿದಾಗ, ಯಾವುದೇ ಸೇರಿಸಲ್ಪಟ್ಟ ವಿಷಯಗಳು ಅಥವಾ ಬದಲಾವಣೆಗಳು ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಥವಾ ಎ ಮೇಲ್ ಸಂದೇಶಗಳಲ್ಲಿ ಕಾಣಿಸುತ್ತವೆ.

2 ಒಂದು ವೇಳೆ ನಿಮಗೆ ಹೊಸ ಫೈಲ್ ಕ್ಲಿಕ್ ಮಾಡಬಹುದಾದ ಐಕನ್ ರೂಪದಲ್ಲಿ ಕಾಣಿಸಬೇಕು ಎಂದರೆ, Display as icon ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಈ ಚೆಕ್ ಬಾಕ್ಸ್ ಆಯ್ಕೆಯಾದರೆ, ನೀವು Change icon ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ, ವಿಭಿನ್ನ ಐಕನ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

ಈಗಾಗಲೇ ಇರುವ ಫೈಲನ್ನು ಲಿಂಕ್ ಅಥವಾ ಎಂಬೆಡ್ ಮಾಡುವುದು

ಈಗಾಗಲೇ ರಚಿಸಲ್ಪಟ್ಟ ಆಬ್ಜೆಕ್ಟನ್ನು ಲಿಂಕ್ ಮಾಡುವುದು ಅಥವಾ ಎಂಬೆಡ್ ಮಾಡಲು :

1 Object ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Create from File ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು ನಂತರ ನೀವು ಸೇರಿಸಲು ಬಯಸುವ ಫೈಲನ್ನು ಹುಡುಕಲು Browse ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 ಒಂದು ಫೈಲನ್ನು ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಅಥವಾ ಇ ಮೇಲ್ ಸಂದೇಶಗಳಿಗೆ ಎಂಬೆಡ್ ಮಾಡುವ ಬದಲು ಸೋರ್ಸ್ ಫೈಲ್‌ಗೆ ಲಿಂಕ್ ಮಾಡಲು, Link to file ಅನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 ಒಂದು ವೇಳೆ ನಿಮಗೆ ಹೊಸ ಫೈಲ್ ಕ್ಲಿಕ್ ಮಾಡಬಹುದಾದ ಐಕನ್ ರೂಪದಲ್ಲಿ ಕಾಣಿಸಬೇಕು ಎಂದರೆ, Display as icon ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಈ ಚೆಕ್ ಬಾಕ್ಸ್ ಆಯ್ಕೆಯಾದರೆ, ನೀವು Change icon ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ, ವಿಭಿನ್ನ ಐಕನ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

ಮೇಲ್ ಮರ್ಜ್ (Mail Merge) :

ನೀವು ಬಹಳ ಜನರಿಗೆ ಪತ್ರವನ್ನು ಕಳುಹಿಸಬೇಕಾಗಿದೆಯೆ ? ಮೇಲ್ ಮರ್ಜ್ ಪ್ರಕ್ರಿಯೆಯು ಒಂದು ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಒಂದು ಡಾಟಾ ಸೋರ್ಸ್‌ನೊಂದಿಗೆ ಸಂಯೋಜಿಸುತ್ತದೆ. ಇದು ವೈಯಕ್ತಿಕ ಅನುಭವವನ್ನು ನೀಡುವ ಪತ್ರಗಳನ್ನು ವೇಗವಾಗಿ ತಯಾರಿಸುತ್ತದೆ.

ಮೇಲ್ ಮರ್ಜ್ ವ್ಯಾಖ್ಯಾನಗಳು (Mail Merge Definitions)

ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್ : (Main Document) ಇದು ಪ್ರತಿಯೊಂದು ಮರ್ಜ್ ಆದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೂ ಒಂದೇ ರೀತಿಯ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಈ ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್ ವಿಭಿನ್ನ ಮಾಹಿತಿಯ ಫೀಲ್ಡ್ ಹೆಸರುಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಳವಡಿಸಬೇಕಾದ ಹೆಸರುಗಳು ಮತ್ತು ಅಡ್ರೆಸ್‌ಗಳು.

ಡಾಟಾ ಸೋರ್ಸ್ ಅಥವಾ ರೆಸಿಪಿಯಂಟ್ ಲಿಸ್ಟ್ (Data Source or Recipients List) : ಮೇಲ್ ಮರ್ಜ್ ಮಾಡುವ ಸಂದರ್ಭದಲ್ಲಿ ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಳವಡಿಸಬೇಕಾದ ಮಾಹಿತಿಯು ಈ ಫೈಲ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ : ಈ ಮೇಲ್ ಮರ್ಜ್ ಪತ್ರವನ್ನು ಯಾರಿಗೆ ಕಳುಹಿಸಭೇಕಾಗಿದೆಯೋ ಆ ವ್ಯಕ್ತಿಗಳ ಹೆಸರು ಮತ್ತು ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಒಳಗೊಂಡ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಈ ಡಾಟಾ

© NIMI NOT TO BE REPUBLISHED

ಸೋರ್ಸ್ ಹೊಂದಿರುತ್ತದೆ.ೆ ಎಕ್ಸೆಲ್ ಸ್ಪ್ರೆಡ್‌ಶೀಟ್‌ಗಳು, ಎಕ್ಸೆಸ್ ಡಾಟಾಬೇಸ್‌ಗಳು, ಅಥವಾ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಟೇಬಲ್‌ಗಳು ಈ ಡಾಟಾ ಸೋರ್ಸ್‌ಗಳ ಉತ್ತಮ ಉದಾಹರಣೆಗಳು ಆಗಿವೆ.

ಫೀಲ್ಡ್ (Field) : ಒಂದು ನಿರ್ದಿಷ್ಟ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಡಾಟಾದ ಒಂದು ವರ್ಗಕ್ಕೆ ಫೀಲ್ಡ್ ಎನ್ನುವರು. ಉದಾಹರಣೆಗೆ : LastName ಎನ್ನುವ ಒಂದು ಫೀಲ್ಡ್ ಜನರ ಕೊನೆಯ ಹೆಸರನ್ನು (last name) ಮಾತ್ರ ಹೊಂದಿರುತ್ತದೆ.

ರೆಕಾರ್ಡ್ (Record) : ಒಂದೇ ವಸ್ತು ಅಥವಾ ವ್ಯಕ್ತಿಗೆ ಸಂಬಂಧಿಸಂತೆ ಡಾಟಾ ಫೀಲ್ಡ್‌ಗಳ ಸಂಪೂರ್ಣ ಸೆಟ್‌ಗೆ ರೆಕಾರ್ಡ್ ಎನ್ನುವರು.ಉದಾಹರಣೆಗೆ : ಒಂದು ರೆಕಾರ್ಡ್ ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ಮೊದಲನೆ ಹೆಸರು, ಕೊನೆಯ ಹೆಸರು, ಅಡ್ರೆಸ್ , ಫೋನ್ ನಂಬರ್, ಮತ್ತು ಹುಟ್ಟಿದ ದಿನಾಂಕ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಮರ್ಜ್ ಫಿಲ್ಡ್ (Merge Field) :ಒಂದು ಡಾಟಾ ಸೋರ್ಸ್‌ನಿಂದ ಮಾಹಿತಿಯನ್ನು ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಯಾವ ಸ್ಥಳದಲ್ಲಿ ನೀವು ಸೇರಿಸಬಯಸುತ್ತೀರೋ ಅದನ್ನು ಮರ್ಜ್ ಫೀಲ್ಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳು ಚೆವ್ರಾನ್ (ಕಿ ಕಿ) ನಿಂದ ಆವರಿಸಿರುತ್ತವೆ. ಒಂದು ಉದಾಹರಣೆ ಹೀಗಿದೆ. : Dear ಕಿFirstName ಕಿ

ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್ (Address Block) : ಒಂದು ಮೇಲ್ ಮರ್ಜ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ತಯಾರಿಸುವ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳ ಗುಂಪಿಗೆ ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ : ಒಂದೇ ಅಡ್ರೆಸ್ name, street address, city, state ಮತ್ತು zip code ಮುಂತಾದವುಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ವರ್ಡ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೇ ಎಲ್ಲ ಸೂಕ್ತ ಅಡ್ರೆಸ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಒಂದೇ ಬಾರಿಗೆ ಅಳವಡಿಸಬಹುದು. ಅದ್ದರಿಂದ ನೀವು ಐದು ಅಥವಾ ಆರು ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಸೇರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

ಗ್ರೀಟಿಂಗ್ ಲೈನ್ (Greeting Line) : ಮೇಲ್ ಮರ್ಜ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಗ್ರೀಟಿಂಗ್ ಲೈನನ್ನು ತಯಾರಿಸುವ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳ ಗುಂಪಿಗೆ ಗ್ರೀಟಿಂಗ್ ಲೈನ್ ಎನ್ನುವರು. ಉದಾಹರಣೆಗೆ : ಕಿDear Mr. McDonald ಕಿ ವರ್ಡ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೆ ಎಲ್ಲ ಸೂಕ್ತ ಗ್ರೀಟಿಂಗ್ ಟೆಕ್ಸ್ಟ್, ಟೈಟಲ್ ಮತ್ತು ಹೆಸರಿನ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಒಂದು ಬಾರಿಗೆ ಅಳವಡಿಸುತ್ತದೆ. ಅದ್ದರಿಂದ ನೀವು ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಅವಶ್ಯಕ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಸೇರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

ೆಡರ್ ರೋ (Header Row) : ಡಾಟಾ ಸೋರ್ಸ್ ಮಾಹಿತಿಯು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುತ್ತದೆ. ಆ ಟೇಬಲ್‌ನ ಮೊದಲನೇ ರೋ (row) ಗೆ ಹೆಡರ್ ರೋ ಎಂದು ಕರೆಯುತ್ತಾರೆ ಮತ್ತು ಇದು ಡಾಟಾ ಸೋರ್ಸ್‌ನ ಫೀಲ್ಡ್ ಹೆಸರುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ : FirstName, LastName, ಮತ್ತು Address ಇವುಗಳಿಗೆ ಹೆಡರ್ ರೋ ಗಳು ಎನ್ನುತ್ತಾರೆ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡುವುದು (Set up and Choose Document Type)

ಮೇಲ್ ಮರ್ಜ್ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಪ್ರರಂಭಿಸಲು, ಮೊದಲು ನೀವು ಯಾವ ರೀತಿಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತಯಾರಿಸಲು ಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬೇಕು.

1 Mailings tab ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 Start Mail Merge ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Step - By - Step Mail Merge Wizard ಆಯ್ಕೆ ಮಾಡಿಕೊಳಿ ್ಳರಿ.

ಬಲಭಾಗದಲ್ಲಿ Mail Merge ಪೇನ್ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ. ಇದರಲ್ಲಿ ನೀವು ಮೇಲ್ ಮರ್ಜ್ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಮುಂದುವರೆಸಬಹುದು.

4 ನೀವು ರಚಿಸಲು ಬಯಸುವ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿಧವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

5 Next : Starting document ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಮೇಲ್ ಮರ್ಜ್ ವಿಜಾರ್ಡ್ ಮುಂದಿನ ಹಂತಕ್ಕೆ ಹೋಗುತ್ತದೆ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವುದು (Select a Document)

ಈ ಮುಂದಿನ ಹಂತದಲ್ಲಿ ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬೇಕು.

1 Starting document ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬೇಕು.

ನೀವು ಪ್ರಸ್ತುತ ಇರುವ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು (current document) ಮೇಲ್ ಮರ್ಜ್‌ಗಾಗಿ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು ಅಥವಾ ನೀವು ಟೆಂಪ್ಲೇಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು ಅಥವಾ ಇವುಗಳ ಬದಲಾಗಿ ಈಗಾಗಲೇ ಇರುವ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನೂ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

2 Next : Select Recipients ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಈಗ ಮೇಲ್ ಮರ್ಜ್ ವಿಝಾರ್ಡ್ 3 ನೇ ಹಂತಕ್ಕೆ ಹೋಗುತ್ತದೆ.

ರೆಸಿಪಿಯಂಟ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವುದು (Select Recipients)

ಈಗ, ನೀವು ಅಡ್ರೆಸ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಎಲ್ಲಿಂದ ಪಡೆದುಕೊಳ್ಳುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವ ಅವಶ್ಯಕೆ ಇರುತ್ತದೆ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ಈಗಾಗಲೇ ಇರುವ (existing list) ಲಿಸ್ಟ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗಿದೆ, ಆದರೆ ನೀವು ಔಟ್‌ಲುಕ್ ಕಾಂಟ್ಯಾಕ್ಟ್‌ಗಳು ಅಥವಾ ನೀವೇ ತಯಾರಿಸಿದ ನಿಮ್ಮ ಸ್ವಂತ ಲಿಸ್ಟ್‌ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

1 Use an existing list ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 Browse ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Select your data file.

ಮೇಲ್ ಮರ್ಜ್ ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಈ ಡಾಟಾ ಫೈಲ್, ಒಂದು ಡಾಟಾಬೇಸ್ ಫೈಲ್ ಆಗಿರಬಹುದು, ಎಕ್ಸೆಲ್ ಸ್ಪ್ರೆಡ್‌ಶೀಟ್ ಆಗಿರಬಹುದು, ಇನ್ನೊಂದು ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್ ಆಗಿರಬಹುದು ಅಥವಾ ಬೇರೆ ರೀತಿಯ ಡಾಟಾಫೈಲ್ ಆಗಿರಬಹುದು.

4 Open ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

Mail Merge Recipients ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ನೀವು ಉಪಯೋಗಿಸಲು ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ನೀವು ಡಾಟಾ ಸೋರ್ಸ್ ಆಗಿ ಎಕ್ಸೆಲ್ ಸ್ಪ್ರೆಡ್‌ಶೀಟನ್ನು ಉಪಯೋಗಿಸುತ್ತಿದ್ದರೆ, ನಿಮಗೆ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ವರ್ಕ್‌ಶೀಟನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ಹೇಳಲಾಗುತ್ತದೆ.

ನೀವು ಯಾವುದಾದರೂ ಒಂದು ಅಡ್ರೆಸ್‌ನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸದಿದ್ದರೆ,ಆದನ್ನು ಅನ್‌ಚೆಕ್ ಮಾಡಬಹುದು.

5 ಸರಿಯಾದ ರೆಸಿಪಿಯಂಟ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡಿದ್ದೀರಿ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ ಮತ್ತು OK ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

6 Next: ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ನಿಮ್ಮ ಪತ್ರವನ್ನು ಬರೆಯಿರಿ.

ಈಗ ಮೇಲ್‌ಮರ್ಜ್ ವಿಝಾರ್ಡ್ 4 ನೇ ಹಂತಕ್ಕೆ ಹೋಗುತ್ತದೆ.

ನಿಮ್ಮ ಪತ್ರವನ್ನು ಬರೆಯಿರಿ (Write Your Letter)

ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್ ತಯಾರಾದ ನಂತರ ಮತ್ತು ಅದಕ್ಕೆ ರೆಸಿಪಿಯಂಟ್ ಲಿಸ್ಟ್ ಜೋಡಣೆ ಮಾಡಿ ಎಡಿಟ್ ಮಾಡಿದ ನಂತರ, ನೀವು ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಅಳವಡಿಸಬಹುದು. ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳು ರೆಸಿಪಿಯಂಟ್ ಲಿಸ್ಟ್‌ನಿಂದ ಪಡೆಯುವ ವಿಭಿನ್ನ ಮಾಹಿತಿಯನ್ನು ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸೇರಿಸುವ ಪ್ಲೇಸ್‌ಹೋಲ್ಡರ್‌ಗಳು ಆಗಿವೆ.

ನೀವು ಮರ್ಜ್ ಫೀಲ್ಡ್‌ನ್ನು ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸೇರಿಸಿದಾಗ, ಆ ಫೀಲ್ಡ್‌ನಲ್ಲಿರುವ ಮಾಹಿತಿ ಆ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಕಾಣುತ್ತದೆ. ಆ ಮಾಹಿತಿ ಆ ರೆಸಿಪಿಯಂಟ್‌ಗೆ ವಿಭಿನ್ನವಾಗಿರುತ್ತದೆ.

1 ನಿಮಗೆ ಮಾಹಿತಿ ಯಾವ ಸ್ಥಳದಲ್ಲಿ ಬೇಕಾಗಿದೆಯೊ ಅಲ್ಲಿ ಕ್ಲಿಕ್ ಮಾಡಿ.

2 ಪ್ಲೇಸ್‌ಹೋಲ್ಡರ್ ಆಯ್ಕೆಗಳಲ್ಲಿ ಒಂದನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

ನೀವು ವಿಝಾರ್ಡ್‌ನಿಂದ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಅಳವಡಿಸಬಹುದು ಅಥವಾ ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Write & Insert Fields ಗ್ರೂಪ್‌ನಿಂದ ಅಳವಡಿಸಬಹುದು :

- **ಅಡ್ರೆಸ್ ಬ್ಲಾಕ್ (Address Block) :** ಇದು ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಹೆಸರು ಮತ್ತು ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಸೇರಿಸುವ ಫೀಲ್ಡ್‌ಗಳ ಒಂದು ಸಂಯೋಜನೆಯಾಗಿದೆ.

- **ಗ್ರೀಟಿಂಗ್ ಲೈನ್ (Greeting Line) :** ಇದು ಗ್ರೀಟಿಂಗ್ ಲೈನ್‌ನಲ್ಲಿ ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಹೆಸರನ್ನು ಸೇರಿಸುವ ಫೀಲ್ಡ್‌ಗಳ ಒಂದು ಸಂಯೋಜನೆಯಾಗಿದೆ.

- **ಮರ್ಜ್ ಫೀಲ್ಡ್‌ನ್ನು ಅಳವಡಿಸುವುದು (Insert Merge Filed) :** ಈ ಬಟನ್‌ನ್ನು ನೀವು ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, ನೀವು ಸೇರಿಸಬಹುದಾದ ಹೆಚ್ಚಿನ ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳು ಕಂಡು ಬರುತ್ತವೆ.

ಪ್ಲೇಸ್ ಹೋಲ್ಡರ್‌ನ್ನು ಕಸ್ಟಮೈಝ್ ಮಾಡಿರಿ.

OK ಕ್ಲಿಕ್ ಮಾಡಿರಿ
(ಐಚ್ಛಿಕ) ಪ್ರತಿ ಮರ್ಜ್ ಫೀಲ್ಡನ್ನು ಅಳವಡಿಸಲು ನೀವು ಇದೇ ವಿಧಾನವನ್ನು ಪುನರಾವರ್ತಿಸಬೇಕು.

Next : Preview your letters ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಮೇಲ್ ಮರ್ಜ್ ವಿಝಾರ್ಡ್ 5 ನೇ ಹಂತಕ್ಕೆ ಹೋಗುತ್ತದೆ.

ನಿಮ್ಮ ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡುವುದು (Preview Your Mail Merge)

ಕೆಲವು ಸಮಯದಲ್ಲಿ, ಮರ್ಜ್ ಆದ ಫೀಲ್ಡ್‌ಗಳ ಹೆಸರುಗಳನ್ನು ಮಾತ್ರ ನೋಡುವುದಕ್ಕಿಂತ, ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸೇರ್ಪಡೆಯಾದ ನಂತರ ಡಾಟಾ ಯಾವ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ ಎಂಬುದನ್ನು ನೋಡುವುದು ಸಹಾಯಕಾರಿಯಾಗಿರುತ್ತದೆ.

ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಮುಗಿಸುವ ಮೊದಲೇ, ಮೇಲ್ ಮರ್ಜ್ ಯಾವ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ ಎಂಬುದನ್ನು ನೀವು ಸುಲಭವಾಗಿ ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡಬಹುದು. ಇದು ನಿಮಗೆ ಬೇಕಾದ ರೀತಿಯಲ್ಲಿ ಫಲಿತಾಂಶವನ್ನು ಕಾಣಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

1 ಮರ್ಜ್ ಆದ ಪ್ರತಿಯೊಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಮೊದಲೇ ನೋಡಲು ಮೇಲ್ ಮರ್ಜ್ ಪೇನ್‌ನಲ್ಲಿರುವ ಆ್ಯರೋ ಬಟನ್‌ಗಳನ್ನು (Arrow Buttons) ಉಪಯೋಗಿಸಿರಿ.

ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Preview Result ಗ್ರೂಪ್‌ನಲ್ಲಿನ ಆ್ಯರೋ ಬಟನ್‌ಗಳನ್ನು ಕೂಡ ನೀವು ಉಪಯೋಗಿಸಬಹುದು.

ಒಬ್ಬ ನಿರ್ದಿಷ್ಟ ರೆಸಿಪಿಯಂಟ್‌ನ್ನು ಹುಡುಕಲು, Preview Results ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ Find Recipient ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿ ಅಥವಾ ಮೇಲ್ ಮರ್ಜ್ ಪೇನ್‌ನಲ್ಲಿ ಹುಡುಕಿರಿ.

2 Next : Complete the merge ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಮೆರ್ಜ್ ಅನ್ನು ಸಂಪೂರ್ಣಗೊಳಿಸಿರಿ(Complete the Merge)

ರೆಸಿಪಿಯಂಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ನೀವು ಸೇರಿಸಿದ ಮೇಲೆ, ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಮರ್ಜ್ ಫೀಲ್ಡ್‌ಗಳಿಂದ ಭರ್ತಿ ಮಾಡಿದ ನಂತರ ಪ್ರತಿ ರೆಸಿಪಿಯಂಟ್‌ಗೆ ಪ್ರತ್ಯೇಕ ಆವೃತ್ತಿಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತಯಾರಿಸುವುದರ ಮೂಲಕ ಮರ್ಜ್‌ನ್ನು ಸಂಪೂರ್ಣಗೊಳಿಸುವುದು ಇದರ ಕೊನೆಯ ಹಂತವಾಗಿದೆ.

ನೀವು ಇನ್ನೂ ವಿಭಿನ್ನ ರೀತಿಯಿಂದಲೂ ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಸಂಪೂರ್ಣಗೊಳಿಸಬಹುದು.:

- ಪ್ರತ್ಯೇಕ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಎಡಿಟ್ ಮಾಡುವುದು : ಇದು ಮೇಲ್ ಮರ್ಜ್‌ನ ಫಲಿತಾಂಶವನ್ನು ಹೊಸ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ತೋರಿಸುತ್ತದೆ. ಇತರ ಯವುದೇ ಡಾಕ್ಯುಮೆಂಟ್‌ನಂತೆ ನೀವು ಮೇಲ್ ಮರ್ಜ್‌ನ ಫಲಿತಾಂಶಗಳನ್ನು ಸ್ವತಂತ್ರವಾಗಿ ಎಡಿಟ್ ಮಾಡಬಹುದು ಮತ್ತು ಸೇವ್ ಮಾಡಬಹುದು ಮತ್ತು ಪ್ರಿಂಟ್ ಮಾಡಬಹುದು.
- ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುವುದು : ಇದು ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಮರ್ಜ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ನೇರವಾಗಿ ಪ್ರಿಂಟರ್‌ಗೆ ಕಳುಹಿಸುತ್ತದೆ.

1 ಮೇಲ್ ಮರ್ಜ್‌ನ್ನು ಸಂಪೂರ್ಣಗೊಳಿಸಲು ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಿರಿ.

ನೀವು ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Finish & Merge ಬಟನ್‌ಅನ್ನು ಕೂಡ ಕ್ಲಿಕ್ ಮಾಡಬಹುದು ಮತ್ತು ಅಲ್ಲಿರುವ merge ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಮರ್ಜ್ ಮಾಡಲು ಯಾವ ರೆಕಾರ್ಡಗಳನ್ನು ಆರಿಸಿಕೊಳ್ಳಬೇಕು ಎನ್ನುವ ಆಯ್ಕೆಯನ್ನು ನಿಮಗೆ ನೀಡಲಾಗುತ್ತದೆ. ನೀವು ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ರೆಕಾರ್ಡಗಳನ್ನು , ಪ್ರಸ್ತುತ ತೋರಿಸಿದ ರೆಕಾರ್ಡನ್ನು , ಅಥವಾ ನಿರ್ದಿಷ್ಟ ಸಾಲುಗಳನ್ನು ಸೂಚಿಸಿ ಅವುಗಳನ್ನು ಮರ್ಜ್ ಮಾಡಬಹುದು.

2 ನೀವು ಮರ್ಜ್ ಮಾಡಲು ಬಯಸಿದ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

3 OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ವರ್ಡ್, ಮುಖ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಮತ್ತು ಡಾಟಾ ಸೋರ್ಸ್‌ನಲ್ಲಿರುವ ಮಾಹಿತಿಯನ್ನು ಮರ್ಜ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಹೊಸ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತಯಾರಿಸುತ್ತದೆ. ಅಥವಾ ಅವುಗಳನ್ನು ಮರ್ಜ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಪ್ರಿಂಟರ್‌ಗೆ ಕಳುಹಿಸುತ್ತದೆ. ಇದು ನೀವು ಆರಿಸಿಕೊಳ್ಳುವ ಆಯ್ಕೆಯನ್ನು ಆಧರಿಸಿದೆ.

ಟೆಂಪ್ಲೇಟ್‌ಗಳು (Templates)

ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟನ್ನು ಹೇಗೆ ರಚಿಸುವುದು

ಟೆಂಪ್ಲೇಟ್ ಒಂದು ಪೂರ್ವ ರಚಿತವಾದ ಡಾಕ್ಯುಮೆಂಟ್ ಆಗಿದೆ. ಇದು ಈಗಾಗಲೇ ತಯಾರಿಸಲಾದ ನಿರ್ದಿಷ್ಟವಾದ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಆಯ್ಕೆಗಳು, ಉದಾಹರಣೆಗೆ, ಮ್ಯಾಕ್ರೊಗಳು, ಹೆಡರ್ ಮತ್ತು ಫೂಟರ್‌ಗಳು, ಕಸ್ಟಮ್ ಡಿಕ್ಷನರಿಗಳು, ಲೇಔಟ್‌ಗಳು, ಇಮೇಜ್‌ಗಳು, ಮತ್ತು ಆಟೊ ಟೆಕ್ಸ್ಟ್ ನಮೂದುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಒಂದೇ ರೀತಿಯ ವಿಷಯಗಳು ಮತ್ತು ರಚನೆಯನ್ನು ಹೊಂದಿದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ರಚಿಸಬೇಕಾದ ಸಂದರ್ಭಗಳಲ್ಲಿ ಟೆಂಪ್ಲೇಟ್‌ಗಳು ಸಮಯವನ್ನು ಉಳಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟನ್ನು ರಚಿಸುವುದು(Create a template in Word document)

ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟನ್ನು ರಚಿಸಲು ಕೆಳಗೆ ಸೂಚಿಸಿದ ಸುಲಭವಾದ ವಿಧಾನವನ್ನು ಅನುಸರಿಸಿ.

ಹಂತ 1: ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತೆರೆಯಿರಿ.

ಹಂತ 2: ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ File ಟ್ಯಾಬ್‌ಗೆ ಹೋಗಿ ಮತ್ತು New ಆಯ್ಕೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಹಂತ 3: Blank document ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ.

ಹಂತ 4: ನೀವು ಮುಂದೆ ಟೆಂಪ್ಲೇಟ್ ಆಗಿ ಉಪಯೋಗಿಸುವ ನಿಮಗೆ ಅಪೇಕ್ಷಣೀಯವಾದ ವಿಷಯವನ್ನು ಸೇರಿಸಿ.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟನ್ನು ಸೇವ್ ಮಾಡುವುದು (Save Microsoft Word Template)

ಒಮ್ಮೆ ನೀವು ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟ್‌ನ್ನು ರಚಿಸಿದರೆ, ಅದನ್ನು ಸೇವ್ ಮಾಡಿ ಮುಂದೆ ನೀವು ಭವಿಷ್ಯದಲ್ಲಿ ಉಪಯೋಗಿಸಿಕೊಳ್ಳಬಹುದು.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್‌ನಲ್ಲಿ, ನೀವು ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟನ್ನು .dotx, .dot ಅಥವಾ .dotm ಫೈಲ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳಲ್ಲಿ ಸೇವ್ ಮಾಡಬಹುದು.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟನ್ನು ಸೇವ್ ಮಾಡಲು, ಕೆಳಗಿನ ಸೂಚನೆಗಳನ್ನು ಅನುಸರಿಸಿ.

1 ನೀವು ಯಾವ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಟೆಂಪ್ಲೇಟ್‌ನ್ನು ರಚಿಸಲು ಬಯಸುತ್ತೀರೋ ಅದನ್ನು ತೆರೆಯಿರಿ.

2 ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ File ಟ್ಯಾಬನ್ನು ತೆರೆಯಿರಿ, ಸ್ಕ್ರೀನ್ ಮೇಲೆ ಆಯ್ಕೆಗಳ ಒಂದು ಪಟ್ಟಿ ಬರುತ್ತದೆ. ಅದರಲ್ಲಿ Save As ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

3 ಫೈಲನ್ನು ಸೇವ್ ಮಾಡಲು ಬಯಸುವ ಸ್ಥಳವನ್ನು ಬ್ರೌಸ್(browse) ಮಾಡಿರಿ. Save As type ಜೊತೆಗಿರುವ ಡ್ರಾಪ್ ಡೌನ್ ಲಿಸ್ಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ವರ್ಡ್ ಟೆಂಪ್ಲೇಟ (*. dotx) ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು ಸ್ಕ್ರಿನ್ ಕೆಳಭಾಗದಲ್ಲಿರುವ Save ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಈಗ ನೀವು ನಿಮ್ಮ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟ್, ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಸೇವ್ ಆಗಿರುವುದನ್ನು ನೋಡಬಹುದು.

ವರ್ಡ್ ಟೆಂಪ್ಲೇಟನ್ನು ಎಡಿಟ್ ಮಾಡುವುದು (Edit the Word Template)

ಒಮ್ಮೆ ನೀವು ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್‌ನಲ್ಲಿ, ಟೆಂಪ್ಲೇಟ್‌ನ್ನು ರಚಿಸಿದರೆ, ಅದನ್ನು ನೀವು ನಿಮ್ಮ ಅವಶ್ಯಕತೆಗಳಿಗೆ ಅನುಸಾರವಾಗಿ ಎಡಿಟ್ ಮಾಡಬಹುದು ಮತ್ತು ಮುಂದೆ ಭವಿಷ್ಯದಲ್ಲಿ ಬರುವ ಅದೇ ತರಹದ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ಸೇವ್ ಮಾಡಿ ಇಡಬಹುದು.

ವರ್ಡ್ ಟೆಂಪ್ಲೇಟ್‌ಗಳನ್ನು ಸರಳವಾಗಿ ಎಡಿಟ್ ಮಾಡಲು ಈ ಕೆಳಗಿನ ವಿಧಾನವನ್ನು ಅನುಸರಿಸಿ:

ಹಂತ 1: ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಎಡಗಡೆಯ ಮೇಲ್ಭಾಗದ ತುದಿಯಲ್ಲಿರುವ File ಟ್ಯಾಬ್‌ಗೆ ಹೋಗಿರಿ ಮತ್ತು Open ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಹಂತ 2 : ಸ್ಕ್ರೀನ್ ಮೇಲೆ Open ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಕಾಣಿಸುತ್ತದೆ. ಈ ಮೊದಲು ನೀವು ಟೆಂಪ್ಲೇಟನ್ನು ಸೇವ್ ಮಾಡಿದ ಸ್ಥಳವನ್ನು Browse ಮಾಡಿರಿ ಮತ್ತು ಸ್ಕ್ರೀನ್ ಕೆಳಗೆ ಇರುವ Open ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಹಂತ 3 : ನಿಮ್ಮ ಟೆಂಪ್ಲೇಟ್‌ನಲ್ಲಿ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಬಹುದು.

ಹಂತ 4 : ನಿಮ್ಮ ಟೆಂಪ್ಲೇಟ್‌ನಲ್ಲಿ ನೀವು ಬಯಸಿದ ಎಲ್ಲ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಿದ ನಂತರ, ಆ ಟೇಂಪ್ಲೇಟನ್ನು ನೀವು ಸೇವ್ ಮಾಡಬೇಕು ಮತ್ತು ನಂತರ ಮುಚ್ಚಬೇಕು (Close)

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟನ್ನು ಉಪಯೋಗಿಸುವುದು (Use Microsoft Word Template)

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಕೂಡ ಅಂತರ್‌ನಿರ್ಮಿತ (in- built) ವರ್ಡ್ ಟೆಂಪ್ಲೇಟ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇವುಗಳನ್ನು ನೀವು ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟನ್ನು ಉಅಪ್ಯೋಗಿಸಲು ಈ ಕೆಳಗಿನ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿ.

ಹಂತ 1: ಹೊಸ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ತೆರೆಯಿರಿ.

ಹಂತ 2: ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ File ಟ್ಯಾಬ್‌ಗೆ ಹೋಗಿರಿ. ಸ್ಕ್ರೀನ್ ಮೇಲೆ File ಮೆನು ಆಯ್ಕೆಗಳು ಪ್ರದರ್ಶಿತವಾಗುತ್ತದೆ. ಅದರಲ್ಲಿರುವ New ಆಯ್ಕೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಹಂತ 3: ಅಲ್ಲಿ Available Templates ವಿಂಡೊ ಕಂಡು ಬರುತ್ತದೆ. Personal ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಹಂತ 4: ಸ್ಕ್ರೀನ್ ಮೇಲೆ ವರ್ಡ್ ಟೆಂಪ್ಲೇಟ್‌ಗಳ ಪಟ್ಟಿ ಕಂಡು ಬರುತ್ತದೆ. ನೀವು ಉಪಯೋಗಿಸಲು ಬಯಸುವ ಟೇಂಪ್ಲೇಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು ಅದರ ಮೇಲೆ ಎರಡು ಬಾರಿ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಈಗ ನೀವು ಆಯ್ಕೆ ಮಾಡಿದ ಟೆಂಪ್ಲೇಟ್ ನಿಮ್ಮ ವರ್ಡ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಸೇರಿಸಲ್ಪಟ್ಟಿದೆ ಎಂಬುದನ್ನು ನೋಡಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಎಂ ಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಫೈಲ್‌ಗಳನ್ನು ತೆರೆಯುವುದು (Open files in MS Excel)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಎಕ್ಸೆಲ್‌ನ ಪರಿಚಯ**
- **ಸೆಲ್ ರೆಫರನ್ಸ್ ಮತ್ತು ಶೀಟ್ ಲಿಂಕ್ ಮಾಡುವುದು**
- **ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿರುವ ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನು ಸೆಟ್ ಮಾಡುವುದು ಅಥವಾ ಕ್ಲಿಯರ್ ಮಾಡುವುದು.**

ಇದು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಒಂದು ಸ್ಪ್ರೆಡ್‌ಶಿಟ್ ಪ್ರೊಗ್ರಾಂ ಆಗಿದೆ. ಎಕ್ಸೆಲ್ ಡಾಟಾವನ್ನು ಕಾಲಂಗಳಲ್ಲಿ ಮತ್ತು ರೋಗಳಲ್ಲಿ ಸಂಘಟಿಸುತ್ತದೆ ಮತ್ತು ಗಣಿತದ ಕೆಲಸಗಳನ್ನು ಮಾಡಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು Windows, macOS, Android ಮತ್ತು iOS ಗಳಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

ಇದರ ಮೊದಲನೇ ಆವೃತ್ತಿಯು 1985 ರಲ್ಲಿ ಬಿಡುಗಡೆಯಾಯಿತು ಮತ್ತು ಮುಂದಿನ ವರ್ಷಗಳಲ್ಲಿ ಅನೇಕ ಬದಲಾವಣೆಗಳನ್ನು ಹೊಂದಿತು. ಆದಾಗ್ಯೂ ಇದರ ಮುಖ್ಯ ಕಾರ್ಯ ಬಹುತೇಕವಾಗಿ ಒಂದೇ ರೀತಿಯದ್ದಾಗಿದೆ.

ಎಕ್ಸೆಲ್ ಸಾಮಾನ್ಯವಾಗಿ ಈ ಕೆಳಗಿನ ಕಾರ್ಯಗಳಿಗೆ ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ:

- ವಿಶ್ಲೇಷಣೆ
- ಡಾಟಾ ನಿರ್ವಹಣೆ
- ಆಯವ್ಯಯ
- ವಿಶುವಲ್‌ಗಳು ಮತ್ತು ಗ್ರಾಫ್‌ಗಳು
- ಆರ್ಥಿಕ ಮಾದರಿ
- ಡಾಟಾ ನಮೂದಿಸಲು
- ಲೆಕ್ಕಪತ್ರ
- ಡಾಟಾ ವಿಶ್ಲೇಷಣೆ
- ಪ್ರೊಗ್ರಾಮಿಂಗ್
- ಮತ್ತು ಇನ್ನೂ ಹೆಚ್ಚಿನ ಕಾರ್ಯಗಳಿಗಾಗಿ

ಎಕ್ಸೆಲ್‌ನ್ನು ಏಕೆ ಉಪಯೋಗಿಸಬೇಕು ?

- ಇದು ಜಗತ್ತಿನ ಅತ್ಯಂತ ಜನಪ್ರೀಯವಾದ ಸ್ಪ್ರೆಡ್‌ಶೀಟ್ ಪ್ರೊಗ್ರಾಂ ಆಗಿದೆ.
- ಇದು ಕಲಿಯಲು ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಪ್ರಾರಂಭಿಸಲು ಸುಲಭವಾಗಿದೆ.
- ಇದರಲ್ಲಿ ಸ್ಕಿಲ್ ಸೀಲಿಂಗ ಉತ್ತಮವಾಗಿದೆ ಎಂದರೆ, ನೀವು ಇದರಲ್ಲಿ ಉತ್ತಮವಾದಂತೆ ಇನ್ನೂ ಹೆಚ್ಚಿನ ಸುಧಾರಿತ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು.
- ಇದನ್ನು ನೀವು ನಿಮ್ಮ ಕೆಲಸಗಳಿಗಾಗಿ ಮತ್ತು ದಿನನಿತ್ಯದ ಜೀವನದಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ ನಿಮ್ಮ ಕುಟುಂಬದ ಬಜೆಟ್ ತಯಾರಿಸಲು ಉಪಯೋಗಿಸಬಹುದು.
- ಇದು ಹೆಚ್ಚು ಸಾಮಾಜಿಕ ಜೀವನದ ಬೆಂಬಲ ಹೊಂದಿದೆ.
- ಇದು ಸತತವಾಗಿ ಮೈಕ್ರೊಸಾಫ್ಟ್‌ನ ಬೆಂಬಲವನ್ನು ಪಡೆದಿದೆ.
- ಟೆಂಪ್ಲೇಟ್‌ಗಳನ್ನು ಮತ್ತು ಫ್ರೇಮ್‌ವರ್ಕ್‌ಗಳನ್ನು ನೀವು ಮತ್ತು ಇತರರು ಮರುಬಳಕೆ ಮಾಡಬಹುದು. ಇದು ಅವುಗಳನ್ನು ಮತ್ತೆ ತಯಾರಿಸುವ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

ಇನ್ನೊಂದು ವರ್ಕ್‌ಶೀಟ್‌ಗೆ ಸೆಲ್ ರೆಫರನ್ಸ್‌ನ್ನು ತಯಾರಿಸುವುದು (Create a cell reference to another worksheet): ನೀವು ಫಾರ್ಮ್ಯುಲಾವನ್ನು ನಮೂದಿಸಲು ಬಯಸುವ ಸೆಲ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ, = (equal sign) ಟೈಪ್ ಮಾಡಿರಿ, ಮತ್ತು ನೀವು ಉಪಯೋಗಿಸ ಬಯಸುವ ಫಾರ್ಮ್ಯುಲಾ ಟೈಪ್ ಮಾಡಿರಿ. ವರ್ಕ್‌ಶೀಟನ್ನು ರೆಫರನ್ಸ್ ಮಾಡುವುದಕ್ಕಾಗಿ ಟ್ಯಾಬ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ನಿವು ರೆಫರನ್ಸ್ ಮಾಡಲು ಬಯಸುವ ಸೆಲ್ ಅಥವಾ ಸೆಲ್‌ಗಳ ರೇಂಜ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

1 ನೀವು ಫಾರ್ಮ್ಯುಲಾ ನಮೂದಿಸಲು ಬಯಸುವ ಸೆಲ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 ಫಾರ್ಮ್ಯುಲಾ ಬಾರ್‌ನಲ್ಲಿ , = (equal sign) ಟೈಪ್ ಮಾಡಿರಿ.

3 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಮಾಡಿರಿ.

- **ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಸೆಲ್‌ಗಳನ್ನು ರೆಫರನ್ಸ್ ಮಾಡಲು** : ರೆಫರನ್ಸ್‌ನ್ನು ರಚಿಸಲು, ಒಂದೇ ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿಯ ಒಂದು ಸೆಲ್‌ನ್ನು ಅಥವಾ ಸೆಲ್‌ಗಳ ರೇಂಜ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED

- ಆಯ್ಕೆಯನ್ನು ಹೆಚ್ಚು ಮಾಡಲು, ನೀವು ಆಯ್ಕೆ ಮಾಡಿದ ಸೆಲ್‌ಗಳ ಬಾರ್ಡರ್‌ನ್ನು ಎಳೆಯುವುದರ (drag) ಮೂಲಕ ಅಥವಾ ಬಾರ್ಡರ್‌ನ ತುದಿಯನ್ನು ಹಿಡಿದು ಎಳೆಯಿರಿ.

- **ಒಂದು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಹೆಸರನ್ನು ರೆಫರನ್ಸ್ ಮಾಡಲು (Reference a defined name):** ವ್ಯಾಖ್ಯಾನಿಸಿದ ಒಂದು ಹೆಸರಿಗೆ ರೆಫರನ್ಸ್‌ನ್ನು ರಚಿಸಲು ಈ ಕೆಳಗಿನ ಯಾವುದಾದರೂ ಒಂದನ್ನು ಮಾಡಿರಿ:

- ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡಿರಿ.

- F3 ಬಟನ್ ಒತ್ತಿರಿ, Paste name ಬಾಕ್ಸ್‌ನಲ್ಲಿನ ಹೆಸರನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ **OK** ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

> **ಸೂಚನೆ : ಕಲರ್ ಕೋಡ್ ಮಾಡಲಾದ ಬಾರ್ಡರ್‌ನಲಿ ಸ್ಕೇರ್ ಕಾರ್ನರ್ ಇಲ್ಲದಿದ್ದರೆ, ಇದು ಹೆಸರಿಸಿದ ರೇಂಜ್‌ಗೆ ರೆಫರನ್ಸ್ ಮಾಡಲಾಗಿರುತ್ತದೆ.**

4 ಈ ಕೆಳಗಿನ ಒಂದನ್ನು ಅನುಸರಿಸಿ:

- ನೀವು ಒಂದೇ ಸೆಲ್‌ನಲ್ಲಿ ರೆಫರನ್ಸ್‌ನ್ನು ರಚಿಸುತ್ತಿದ್ದರೆ, Enter ಕೀ ಯನ್ನು ಒತ್ತಿರಿ.

- ನೀವು ಅರೇ ಫಾರ್ಮ್ಯುಲಾನಲ್ಲಿ (array formula) ರೆಫರನ್ಸ್‌ನ್ನು ರಚಿಸುತ್ತಿದ್ದರೆ, (ಉದಾಹರಣೆಗೆ :A1:G4), Ctrl+Shift+Enter

  ಒಂದೇ ಸೆಲ್ ಗೆ ಅಥವಾ ಸೆಲ್‌ಗಳ ಶ್ರೇಣಿಗೆ ನೀವು ರೆಫರನ್ಸ್‌ನ್ನು ಮಾಡಬಹುದು ಮತ್ತು ಅರೇ ಫಾರ್ಮ್ಯುಲಾ ಒಂದು ಅಥವಾ ಅನೇಕ ಫಲಿತಾಂಶಗಳನ್ನು ಲೆಕ್ಕ ಮಾಡುತ್ತದೆ.

ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿರುವ ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನು ಸೆಟ್ ಮಾಡುವುದು ಅಥವಾ ಕ್ಲಿಯರ್ ಮಾಡುವುದು: ಒಂದು ವೇಳೆ ನೀವು ಒಂದು ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟ ಆಯ್ಕೆಯನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಲು ಬಯಸಿದರೆ, ಆ ಆಯ್ಕೆಯನ್ನು ಒಳಗೊಂಡ ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನು ಮಾತ್ರ ನೀವು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ನೀವು ಇಡೀ ವರ್ಕ್‌ಶೀಟನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಲು ಬಯಸದಿದ್ದರೆ, ಪ್ರಿಂಟ್ ಮಾಡಲು ಬಯಸಿದ ನೀವು ಹೆಸರಿಸಿದ ಒಂದು ಸೆಲ್ ಅಥವಾ ಸೆಲ್ ರೇಂಜ್‌ಗಳನ್ನು ಪ್ರಿಂಟ್ ಏರಿಯಾ ಎನ್ನುತ್ತಾರೆ. ಪ್ರಿಂಟ್ ಏರಿಯಾ ಹೆಸರಿಸಿದ ನಂತರ ಯಾವಾಗ ನೀವು ವರ್ಕ್‌ಶೀಟನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತೀರೋ ಆಗ ಪ್ರಿಂಟ್ ಏರಿಯಾ ಮಾತ್ರ ಪ್ರಿಂಟ್ ಆಗಿರುತ್ತದೆ. ನಿಮ್ಮ ಅವಶ್ಯಕತೆಗೆ ತಕ್ಕಂತೆ ನೀವು ಸೆಲ್‌ಗಳನ್ನು ಸೇರಿಸುವುದರಿಂದ ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನು ಹೆಚ್ಚಿಸಬಹುದು ಮತ್ತು ನೀವು ಇಡೀ ವರ್ಕ್‌ಶೀಟನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಲು ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನು ಕ್ಲಿಯರ್ ಮಾಡಬಹುದು.

ಒಂದು ವರ್ಕ್‌ಶೀಟ್ ಅನೇಕ ಪ್ರಿಂಟ್ ಏರಿಯಾಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ಪ್ರತಿಯೊಂದು ಪ್ರಿಂಟ್ ಏರಿಯಾ ಕೂಡ ಪ್ರತ್ಯೇಕ ಪುಟವಾಗಿ ಪ್ರಿಂಟ್ ಆಗುತ್ತವೆ.

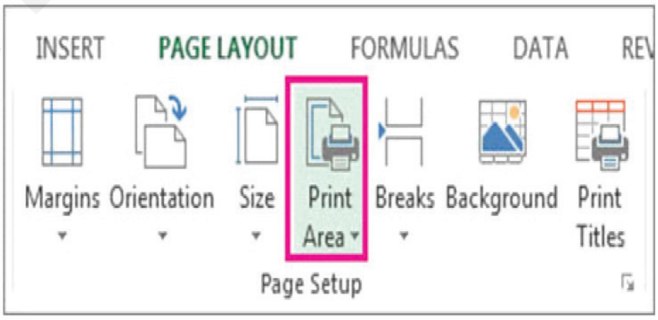
ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಪ್ರಿಂಟ್ ಏರಿಯಾಗಳನ್ನು ಸೆಟ್ ಮಾಡುವುದು (Set one or more print areas)

1 ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿ, ನೀವು ಪ್ರಿಂಟ್ ಏರಿಯಾವನ್ನಾಗಿ ಹೆಸರಿಸಬಯಸುವ ಸೆಲ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

  ಸೂಚನೆ : ಒಂದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು ಪ್ರಿಂಟ್ ಏರಿಯಾಗಳನ್ನು ಸೆಟ್ ಮಾಡಲು, Ctrl ಕೀಲಿಯನ್ನು ಹಿಡಿದುಕೊಂಡಿರಿ ಮತ್ತು ನೀವು ಪ್ರಿಂಟ್ ಮಾಡಲು ಬಯಸುವ ಏರಿಯಾವನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಪ್ರತಿಯೊಂದು ಪ್ರಿಂಟ್ ಏರಿಯಾ ಪ್ರತ್ಯೇಕ ಪುಟದಲ್ಲಿ ಪ್ರಿಂಟ್ ಆಗುತ್ತದೆ.

2 Page Layout ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Page Setup ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ Print Area ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ನಂತರ Set Print Area ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

> **ಸೂಚನೆ : ನೀವು ವರ್ಕ್‌ಬುಕ್‌ನ್ನು ಸೇವ್ ಮಾಡಿದಾಗ, ನೀವು ಸೆಟ್‌ಮಾಡಿದ ಪ್ರಿಂಟ್ ಏರಿಯಾ ಕೂಡ ಸೇವ್ ಆಗಿರುತ್ತದೆ.**



© NIMI NOT TO BE REPUBLISHED

ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವುದು (Manipulate data)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವುದು
- ಸೆಲ್ ರೇಂಜ್.

ಡಾಟಾ ವಿಶ್ಲೇಷಣೆ ಒಂದು ಸವಾಲಿನ ಕಾರ್ಯವಾಗಿದೆ. ವಿಶೇಷವಾಗಿ ನೀವು ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಕೌಶಲ್ಯವನ್ನು ಹೊಂದಿರದಿದ್ದಾಗ. ಈ ಲೇಖನದಲ್ಲಿ, ನಾವು ಕೆಲವು ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾದ ಡಾಟಾ ನಿರ್ವಹಣೆಯ ತಂತ್ರಗಳನ್ನು ಚರ್ಚಿಸೋಣ. ಈ ತಂತ್ರಗಳನ್ನು ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಡಾಟಾ ವಿಶ್ಲೇಷಣೆಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಕೆಲವು ನೈಜ ಜೀವನದ ಉದಾಹರಣೆಗಳೊಂದಿಗೆ ಈ ತಂತ್ರಗಳ ಶಕ್ತಿಯನ್ನು ವಿವರಿಸಲಾಗುತ್ತದೆ. ಡಾಟಾ ನಿರ್ವಹಣೆಯ 9 ಸಾಮಾನ್ಯ ತಂತ್ರಗಳನ್ನು ಕೆಳಗೆ ಚರ್ಚಿಸಲಾಗಿದೆ:

1 ಫಿಲ್ಟರಿಂಗ್ (Filtering)

2 ಸಾರ್ಟಿಂಗ್ (Sorting)

3 ಗ್ರೂಪಿಂಗ್ (Grouping)

4 ಪೈವೋಟಿಂಗ್ (Pivoting)

5 ಟ್ರಾನ್ಸ್‌ಪಾಂಡಿಂಗ್ (Transposing)

6 ಡಾಟಾ ಟೈಪ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Changing Data Types)

7 ಕಾಲಂ ಮತ್ತು ರೋಗಳನ್ನು ಸೇರಿಸುವುದು. (Adding Columns and Rows)

8 ಕಾಲಂ ಮತ್ತು ರೋಗಳನ್ನು ಹೆಸರಿಸುವುದು (Naming Columns or Rows)

9 ಕಾಲಂ ಮತ್ತು ರೋಗಳನ್ನು ಅಳವಡಿಸುವುದು. (Inserting Columns or Rows)

ಈ ಮೇಲಿರುವ ಪ್ರತಿಯೊಂದು ತಂತ್ರವು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಉತ್ತಮವಾಗಿ ಅರ್ಥೈಸಿಕೊಳ್ಳುವಂತೆ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅದು ಯಾವ ರೀತಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ - ವಿಭಿನ್ನ ರೀತಿಯ ದೃಶ್ಯೀಕರಣದ ಮೂಲಕ ಹೊರಗಿನ ವಸ್ತುಗಳನ್ನು ಅನ್ವೇಶಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ. ಈ ತಂತ್ರಗಳು ನಿಮ್ಮ ಕಾರ್ಯ ದಕ್ಷತೆಯನ್ನು ಉತ್ತಮಗೊಳಿಸಲು ಸಹಾಯ ಮಾಡುವುದಲ್ಲದೇ, ಎಕ್ಸೆಲ್ ಬಗ್ಗೆ ತಿಳುವಳಿಕೆ ಇಲ್ಲದ ಜನರಿಗೂ ಕೂಡ ಅದನ್ನು ಸುಲಭವಾಗಿಸುತ್ತವೆ.

ಡಾಟಾ ಟೈಪ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Changing Data Types)

ಈ ಕಮಾಂಡ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡ ನಂತರ, Select Data Type ವಿಂಡೊ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ. ಇಲ್ಲಿ ಡಾಟಾವನ್ನು 3 ವಿಭಾಗಗಳನ್ನಾಗಿ ವಿಂಗಡಿಸಲಾಗಿದೆ: Text DATA only; Numeric Data only ; Mixed Data (text and numeric)

ಮುಂದೆ ನೀವು ನಿಮ್ಮ ಆಯ್ಕೆಯನ್ನು ಈ ಕೆಳಗಿನ ವಿಭಾಗಗಳಿಂದ ಒಂದುಅಥವಾ ಒಂದಕ್ಕಿಂತ ಆಯ್ಕೆಗಳನ್ನು ಆರಿಸುವುದರ ಮೂಲಕ ಸಂಸ್ಕರಿಸಬಹುದು : Text ; Numeric ; Date & Time ; Logical ;and Object Linking & Embedding

ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋ ಗಳನ್ನು ಸೇರಿಸುವುದು (Adding Columns and Rows): ನಿಮ್ಮ ಡಾಟಾಗಳಿಗಾಗಿ ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋ ಗಳನ್ನು ಸೇರಿಸುವುದರಿಂದ ನಿಮ್ಮ ಕಾರ್ಯವು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ನೀವು ವಿಭಿನ್ನ ವಿಷಯಗಳನ್ನು ಹೊಂದಿದ ಡಾಟಾಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಟೇಬಲ್‌ನಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿರುವಿರಿ ಮತ್ತು ಒಂದಕ್ಕೊಂದು ಸಂಬಂಧಿಸಿದಂತೆ ಅವುಗಳ ಉತ್ತರವನ್ನು ನೀವು ನೋಡಲು ಬಯಸಿದರೆ, ಆ ಎರಡೂ ಉತ್ತರಗಳನ್ನೂ ಒಂದೇ ಕಾಲಂನಲ್ಲಿ ಹೊಂದಿದ್ದರೆ ನಿಮಗೆ (ಮತ್ತು ನೀವು ಆ ಡಾಟಾವನ್ನು ಹಂಚಿಕೊಳ್ಳುವ (share) ಜನರಿಗೆ) ಹೆಚ್ಚು ಅನುಕೂಲಕರವಾಗಿರುತ್ತದೆ.

ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋಗಳನ್ನು ಹೆಸರಿಸುವುದು (Naming Columns and Rows) : ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋಗಳನ್ನು ಹೆಸರಿಸಲು, ನೀವು ಹೆಸರು ನೀಡಲು ಬಯಸಿದ ಸೆಲ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ನಂತರ Data > Data tools > Name Columns and Rows ಗೆ ಹೋಗಿರಿ. ನಂತರ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಮೊದಲನೆ ಸೆಲ್‌ನ ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡಿರಿ. ನಿಮ್ಮ ಎಲ್ಲ ಸೆಲ್‌ಗಳಿಗೂ ಹೆಸರನ್ನು ನೀಡುವುದು ಮುಗಿಯುವವರೆಗೂ ಟೈಪ್ ಮಾಡುವುದು ಅಥವಾ ಕ್ಲಿಕ್ ಮಾಡುವುದನ್ನು ಮುಂದುವರೆಸಿರಿ.

ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋಗಳನ್ನು ಹೆಸರಿಸುವುದು (Naming Columns and Rows) :

ಸ್ಪ್ರೆಡ್‌ಶೀಟನಲ್ಲಿರುವ ಪ್ರತಿಯೊಂದು ಕಾಲಂ ಮತ್ತು ರೋ ಡೀಫಾಲ್ಟ್ ಆದ ಹೆಸರುಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ, ಆದರೆ ಈ ಹೆಸರುಗಳನ್ನು ನಾವು

© NIMI NOT TO BE REPUBLISHED

ಬದಲಾಯಿಸಬಹುದು. ನೀವು ಡಾಟಾವನ್ನು ಸಂಕ್ಷಿಪ್ತಗೊಳಿಸುವಾಗ ಮತ್ತು ಒಂದೇ ಹೆಸರನ್ನು ಕಾಲಂ ಮತ್ತು ರೋ ಗಳಿಗೆ ಸತತವಾಗಿ ಅನ್ವಯಿಸಲು ಇದು ಬಹಳ ಸಹಕಾರಿಯಾಗಿರುತ್ತದೆ.

ಒಂದು ಕಾಲಂನ್ನು ಮರುಹೆಸರಿಸಲು, ಆ ಕಾಲಂನಲ್ಲಿರುವ ಯಾವುದೇ ಸೆಲ್ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು column name ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ಹೊಸ ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಮತ್ತು enter ಕೀಯನ್ನು ಒತ್ತಿರಿ. ಒಂದು ರೋ ಅನ್ನು ಮರುಹೆಸರಿಸಲು, ಆ ರೋನಲ್ಲಿರುವ ಯಾವುದೇ ಒಂದು ರೋ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು Row Labels ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಹೊಸ ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಮತ್ತು enter ಕೀಯನ್ನು ಒತ್ತಿರಿ.

ಕಾಲಂಗಳನ್ನು ಅಥವಾ ರೋ ಗಳನ್ನು ಅಳವಡಿಸುವುದು (Inserting Columns or Rows)

ಕಾಲಂ ಅಥವಾ ರೋಗಳನ್ನು ಅಳವಡಿಸುವುದು, ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಎಕ್ಸೆಲ್‌ನ ಅತ್ಯಂತ ಸರಳವಾದ ಒಂದು ತಂತ್ರವಾಗಿದೆ.

ನಿಮ್ಮ ಸ್ಪ್ರೆಡ್‌ಶೀಟ್‌ಗೆ ಹೆಚ್ಚು ಕಾಲಂಗಳನ್ನು ಮತ್ತು ರೋಗಳನ್ನು ಸೇರಿಸುವುದರ ಮೂಲಕ, ಈ ತಂತ್ರವು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಹೆಚ್ಚು ಸ್ಪಷ್ಟತೆಯಿಂದ ಮತ್ತು ನಿಖರತೆಯಿಂದ ವಿಶ್ಲೇಷಣೆ ಮಾಡಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ನಿಮ್ಮ ಡಾಟಾದ ವಿಭಿನ್ನ ಅಂಶಗಳನ್ನು ತೋರಿಸಲು ಇದು ಸಹಾಯ ಮಾಡುತ್ತದೆ . ಉದಾಹರಣೆಗೆ, ವಿಭಿನ್ನವಾದ ವರ್ಷಗಳು, ಪ್ರದೇಶಗಳು , ಉತ್ಪನ್ನಗಳು ಇತ್ಯಾದಿ.

ಈ ತಂತ್ರವು ಎಲ್ಲಿ ಉಪಯೋಗವಾಗುತ್ತದೆ ಎಂಬುದಕ್ಕೆ ಎರಡು ಉದಾಹರಣೆಗಳು :

1 ನೀವು ಒಂದು ನಿರ್ದಿಷ್ಟ ಪ್ರದೇಶದ ವಿಭಿನ್ನ ವರ್ಷಗಳನ್ನು ಹೋಲಿಸಲು ಬಯಸಿದಾಗ

2 ನೀವು ಒಂದು ಉತ್ಪನ್ನವು ವಿಭಿನ್ನ ಪ್ರದೇಶಗಳಲ್ಲಿ ಯಾವ ರೀತಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ ಎಂದು ನೋಡಲು ಬಯಸಿದರೆ,

ತೀರ್ಮಾನ (Conclusion)

ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುವುದು ಎಂದರೆ ಸಮಯಯನ್ನು ತೆಗೆದುಕೊಳ್ಳುವ ಒಂದು ಕೆಲಸವಾಗಿದೆ ಮತ್ತು ಕ್ಲಿಷ್ಟಕರವಾದ ಕೆಲಸವಾಗಿದೆ. ಆದರೆ ಸರಿಯಾದ ತಂತ್ರಗಳನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ ನಿಮ್ಮ ಸಮಯದ ಉಳಿತಾಯವಾಗುತ್ತದೆ ಮತ್ತು ತಪ್ಪುಗಳು ಆಗುವುದನ್ನು ತಡೆಯುತ್ತದೆ. ಈ ಲೇಖನದಲ್ಲಿರುವ 9 ಮೂಲ ತಂತ್ರಗಳು ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸಲು ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತವೆ.

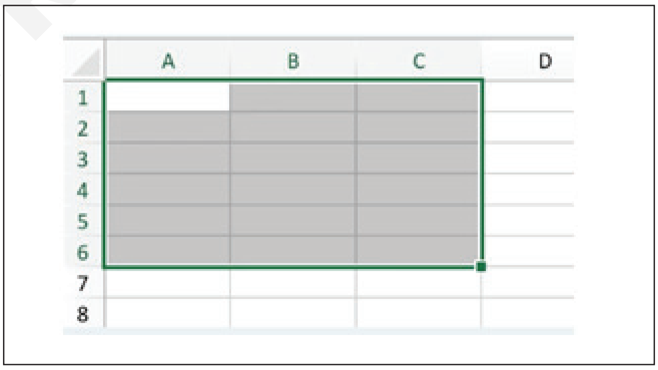
ಸೆಲ್ ರೇಂಜ್ ಎಂದರೇನು (What is a Cell Range)

ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಸೆಲ್ ರೇಂಜ್ ಎಂದರೆ, ಆಯ್ಕೆ ಮಾಡಿದ ಸೆಲ್‌ಗಳ ಗುಂಪು. ಈ ರೇಂಜ್ ಸಾಮಾನ್ಯವಾಗಿ ಚೌಕಾಕಾರವಾಗಿರುತ್ತದೆ (square). ಒಂದು ಸೆಲ್ ರೇಂಜ್‌ನ್ನು ಫಾರ್ಮ್ಯುಲಾಗಳಲ್ಲಿಯೂ ಕೂಡ ನೀವು ಉಲ್ಲೇಖಿಸಬಹುದು.

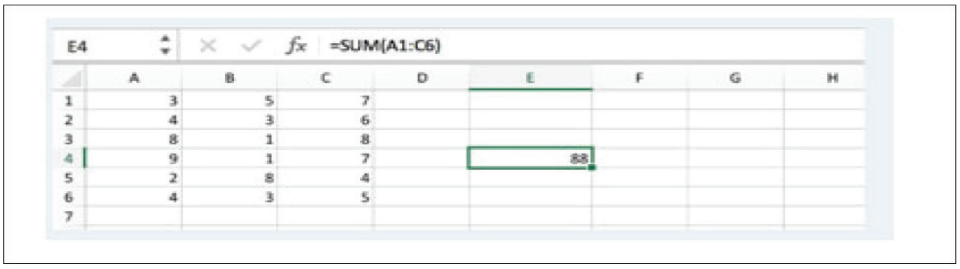
ಒಂದು ಸ್ಪ್ರೆಡ್‌ಶೀಟ್‌ನಲ್ಲಿ, ಸೆಲ್ ರೇಂಜ್‌ನ್ನು ಅದರ ಮೇಲಿನ ಎಡಭಾಗದ ಸೆಲ್‌ನ(ಕನಿಷ್ಟ ಸಂಖ್ಯೆ) ಉಲ್ಲೇಖ ಮತ್ತು ರೇಂಜ್‌ನ ಕೆಳ ಬಲಭಾಗದ ಸೆಲ್‌ನ ಉಲ್ಲೇಖವನ್ನು ನಮುದಿಸುವ ಮೂಲಕ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಕೆಲವೊಮ್ಮೆ ಪ್ರತ್ಯೇಕ ಸೆಲ್‌ಗಳನ್ನು ಈ ಆಯ್ಕೆಗೆ ಸೇರಿಸಲಾಗುತ್ತದೆ. ಆಗ ಆ ರೇಂಜ್‌ನ್ನು ಅನಿಯಮಿತವಾದ ಸೆಲ್ ರೇಂಜ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಕನಿಷ್ಟ ಮತ್ತು ಗರಿಷ್ಟ ಸಂಖ್ಯೆಗಳನ್ನು ಸೇರಿಸಲಾಗುತ್ತದೆ. ಇದು ಗಣಿತದ ರೇಂಜ್‌ಗಿಂತ ವಿಭಿನ್ನವಾಗಿದೆ. ಇದರಲ್ಲಿ ಗರಿಷ್ಟ ಮತ್ತು ಕನಿಷ್ಟ ಸಂಖ್ಯೆಗಳ ನಡುವಿನ ಸಂಖ್ಯೆಗಳ ಒಂದು ಸಮೂಹವಾಗಿರುತ್ತದೆ.

ಒಂದು ಸಮ್ಮಿತೀಯ (symmetrical) ಸೆಲ್ ರೇಂಜ್ ಈ ಕೆಳಗಿನಂತೆ ಕಾಣುತ್ತದೆ. (A1:C6) ಸೆಲ್ ರೇಂಜ್‌ನ ಸಂಕೇತವು : ಮೇಲಿನ ಎಡ ಸೆಲ್ A1 ದಿಂದ ಕೆಳಗಿನ ಬಲ ಸೆಲ್ C6 ವರೆಗಿನ ರೇಂಜ್

ಸೆಲ್ ರೇಂಜ್‌ನ್ನು ಫಾರ್ಮ್ಯುಲಾದಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.



ಉದಾಹರಣೆಗೆ , ಆಯ್ಕೆ ಮಾಡಿದ ಸೆಲ್‌ಗಳಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳ ಮೊತ್ತವನ್ನು ಕಂಡುಹಿಡಿಯಲು . (A1:C6) ಸೆಲ್ ರೇಂಜ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಸಂಖ್ಯೆಗಳ ಮೊತ್ತವನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಸಂಕೇತವು =SUM(A1:C6) ಆಗಿರುತ್ತದೆ.



	A	B	C	D	E	F	G	H
1	3	5	7					
2	4	3	6					
3	8	1	8					
4	9	1	7		88			
5	2	8	4					
6	4	3	5					
7								

NOT TO BE REPUBLISHED © NIMI

ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Create and format tables)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಟೇಬಲ್‌ನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು
- ಸಾರ್ಟ್ ಮತ್ತು ಫಿಲ್ಟರ್ ಮಾಡುವುದು
- ಟೇಬಲ್‌ನ್ನು ಟೆಕ್ಸ್ಟ್‌ಗೆ ಬದಲಾಯಿಸುವುದು

ಟೇಬಲ್‌ನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Create and format table)

ಸೆಲ್ ರೇಂಜ್‌ನ್ನು ಟೇಬಲ್ ಆಗಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದರಿಂದ ಆ ಎಲ್ಲ ರೋ ಗಳು ಮತ್ತು ಕಾಲಂಗಳು ಒಂದಕ್ಕೊಂದು ಸಂಬಂಧಿಸಿದವು ಎಂದು ಎಕ್ಸೆಲ್‌ಗೆ ತಿಳಿಸುತ್ತದೆ. ಮತ್ತು ಈ ಟೇಬಲ್‌ನ ಮೊದಲನೇ ರೋ ನಲ್ಲಿ ಹೆಡರ್‌ಗಳು ಇರುತ್ತವೆ. ಈ ರೀತಿ ಮಾಡುವುದರಿಂದ ನಿಮ್ಮ ರೇಂಜ್‌ಗೆ ಒಂದು ಅರ್ಥ ಬರುತ್ತದೆ. ಎಕ್ಸೆಲ್ ಇದನ್ನು ಉತ್ತಮವಾಗಿ ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಇದರ ಜೊತೆಗೆ, ಅನೇಕ ಹೆಚ್ಚಿನ ಲಾಭಗಳು ಸಿಗುತ್ತವೆ.

ಟೇಬಲ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Format table)

ಟೇಬಲ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುತ್ತದೆ. ಒಂದು ಟೇಬಲ್ - ಕ್ಯಾಪ್ಷನ್‌ಗಳನ್ನು, ರೋ ಗ್ರೂಪ್‌ಗಳನ್ನು, ಮತ್ತು ಕಾಲಂ ಗ್ರೂಪ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ಒಂದು ರೋ ಗ್ರೂಪ್ ರೋ ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ ಹಾಗೆಯೇ ಒಂದು ಕಾಲಂ ಗ್ರೂಪ್ ಕಾಲಂಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ರೋ ಮತ್ತು ಕಾಲಂಗಳು ಸೆಲ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಟೇಬಲ್‌ಗಳನ್ನು ಪದರಗಳಲ್ಲಿ (layers) . ಎಂದರೆ- ಟೇಬಲ್, ಕಾಲಂ ಗ್ರೂಪ್‌ಗಳು, ಕಾಲಂಗಳು, ರೋ ಗ್ರೂಪ್‌ಗಳು , ರೋ ಗಳು ಮತ್ತು ಸೆಲ್‌ಗಳು ಈ ರೀತಿಯ ನಿರ್ದಿಷ್ಟ ಶ್ರೇಣಿಯಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ.

ಸಾರ್ಟಿಂಗ್ (Sorting)

ಸಾರ್ಟಿಂಗ್ ಎಂದರೆ ಡಾಟಾ ವಿಶ್ಲೇಷಣೆಯ ಇನ್ನೊಂದು ತಂತ್ರವಾಗಿದೆ ಮತ್ತು ಇದನ್ನು ನಿಮ್ಮ ಡಾಟಾದ ಶ್ರೇಣಿಯನ್ನು ಪುನಃ ಹೊಂದಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಅನ್ವೇಶಿಸಲು ಮತ್ತು ತಿಳಿದುಕೊಳ್ಳಲು ಒಂದು ಸುಲಭ ವಿಧಾನವಾಗಿದೆ.

ಉದಾಹರಣೆಗೆ : ನೀವು ವಿಭಿನ್ನ ಸಂಖ್ಯೆಯ ಒಂದು ಪಟ್ಟಿಯನ್ನು ಹೊಂದಿರುವಿರಿ ಎಂದು ತಿಳಿಯೋಣ

1, 10, 2, 3, 4

ನೀವು ಈ ಪಟ್ಟಿಯನ್ನು ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ (ಕನಿಷ್ಟ ಸಂಖ್ಯೆಯಿಂದ ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯವರೆಗೆ) ಸಾರ್ಟ್ ಮಾಡಲು ಬಯಸಿದರೆ, ನಾವು ಈ ಪಟ್ಟಿಯ ಕಾಲಂ ಹೆಡಿಂಗ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ನಂತರ Sort Ascending ನ್ನು ಆಯ್ಕೆ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಇದು ಪಟ್ಟಿಯನ್ನು ಈ ರೀತಿಯಾಗಿ ಹೊಂದಿಸುತ್ತದೆ.

1, 2, 3, 4, 10

ಸಂಖ್ಯೆಗಳನ್ನು ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ ಸಾರ್ಟ್ ಮಾಡುವುದರಿಂದ ಅವು ಗಾತ್ರದಲ್ಲಿ ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ ಇರುವುದನ್ನು ನಾವು ನೋಡಬಹುದು. ಒಂದು ವೇಳೆ ನಾವು ಸಾರ್ಟ್ ಮಾಡುವ ಕ್ರಮವನ್ನು ಇಳಿಕೆ ಕ್ರಮದಲ್ಲಿ (ಗರಿಷ್ಟದಿಂದ ಕನಿಷ್ಟ ಸಂಖ್ಯೆಯವರೆಗೆ) ಬದಲಾಯಿಸಲು ಬಯಸಿದರೆ, Columx A ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಬೇಕು ಮತ್ತು Sort Descending ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬೇಕು. ಇದು ಪಟ್ಟಿಯನ್ನು ಈ ರೀತಿಯಾಗಿ ಹೊಂದಿಸುತ್ತದೆ.

10, 4, 3, 2, 1

ಇಳಿಕೆ ಕ್ರಮದಲ್ಲಿ ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯಿಂದ ಕನಿಷ್ಟ ಸಂಖ್ಯೆಯವರೆಗೆ ಸಾರ್ಟ್ ಮಾಡುವುದರಿಂದ, ಸಂಖ್ಯೆಗಳು ಗಾತ್ರದಲ್ಲಿ ಇಳಿಕೆ ಕ್ರಮದಲ್ಲಿ ಇರುವುದನ್ನು ನಾವು ನೋಡಬಹುದು.

ಫಿಲ್ಟರಿಂಗ್ (Filtering)

ನಿರ್ದಿಷ್ಟ ಮಾನದಂಡಗಳ ಪ್ರಕಾರ ಡಾಟಾವನ್ನು ಹೊಂದಿಸುವ ಪ್ರಕ್ರಿಯೆಗೆ ಫಿಲ್ಟರಿಂಗ ಎನ್ನುತ್ತಾರೆ. ದೊಡ್ಡ ಗಾತ್ರದ ಡಾಟಾಸೆಟ್‌ಗಳಿಂದ ಡಾಟಾದ `ಸಬ್‌ಸೆಟ್‌ಗಳನ್ನು ಗುರುತಿಸಲು ಇದು ಒಂದು ಪರಿಣಾಮಕಾರಿಯಾದ ವಿಧಾನವಾಗಿದೆ.

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾವು 2012 -2015 ರ ತಿಂಗಳುಗಳ ಮಾರಾಟದ ಒಂದು ದತ್ತಾಂಶವನ್ನು ಹೊಂದಿದ್ದೇವೆ. ನಾವು ಒಂದೇ ವರ್ಷದ ಒಟ್ಟೂ ಮಾರಾಟವನ್ನು ಮಾತ್ರ ನೋಡಬೇಕಾದರೆ, ಫಿಲ್ಟರಿಂಗ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು ಸೂಕ್ತವಾಗಿರುತ್ತದೆ. ಅಥವಾ ಒಂದು ವೇಳೆ ನಮಗೆ ಎಷ್ಟು ತಿಂಗಳುಗಳು ಉತ್ತಮ ಧನಾತ್ಮಕ ಬೆಳವಣಿಗೆ ಆಗಿದೆ ಎಂಬುದನ್ನು ನೋಡಬಹುದು.

ಮೊದಲು ಯಾವ ಕಾಲಂ ಫಿಲ್ಟರ್ ಮಾನದಂಡವನ್ನು ಹೊಂದಿದೆ ಎಂಬುದನ್ನು ಗುರುತಿಸಿ ಮತ್ತು ಅದನ್ನು ನಿಮ್ಮ ಡಾಟಾ ಫಿಲ್ಟರ್ ಮಾಡಲು ಉಪಯೋಗಿಸಿರಿ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾನು Sales Growth ಎನ್ನುವ ಒಂದು ಹೊಸ ಕಾಲಂ ಅನ್ನು ರಚಿಸಿದ್ದೇನೆ. ನಂತರ Sales Growth ಕಾಲಂ ಅನ್ನು ಹೈಲೈಟ್ ಮಾಡಿರಿ ಮತ್ತು ಟೂಲಬಾರ್‌ನಲ್ಲಿರುವ Data ಮೆನುವಿನಿಂದ Filter ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಇದು ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನ್ನು ತೆರೆಯುತ್ತದೆ. ಇ ದರಲ್ಲಿ ನೀವು ನಿಮ್ಮ ಫಿಲ್ಟರಿಂಗ್ ಮಾನದಂಡವನ್ನು ನಮೂದಿಸಿ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಆ ವರ್ಷಗಳಲ್ಲಿನ positive revenue growth > 0% ಯನ್ನು ಹೊಂದಿರುವ total sales ನ್ನು ಕಂಡುಹಿಡಿಯಲು ನಾನು ಕಿ > 0%” ನ್ನು ನನ್ನ ಫಿಲ್ಟರ್ ಮಾನದಂಡವನ್ನಾಗಿ ಉಪಯೋಗಿಸುತ್ತಿದ್ದೇನೆ.

1 ನೀವು ಟೆಕ್ಸ್ಟ್‌ಗೆ ಬದಲಾಯಿಸಲು ಬಯಸಿದ ರೋ ಗಳನ್ನು ಅಥವಾ ಟೇಬಲ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 Layout ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Data section ನಲ್ಲಿರುವ Convert to Text ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Convert to Text ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Seperate Text with ನ ಅಡಿಯಲ್ಲಿ, ಕಾಲಂನ ಬಾಂಡರಿಯ ಬದಲಾಗಿ ನೀವು ಉಪಯೋಗಿಸಲು ಬಯಸಿದ ಪ್ರತ್ಯೇಕಿಸುವ ಅಕ್ಷರವನ್ನು (seperator character) ಕ್ಲಿಕ್ ಮಾಡಿ ಆಯ್ಕೆ ಮಾಡಿರಿ.

4 OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಫಾರ್ಮ್ಯುಲಾಗಳು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು

ಎಂ ಎಸ್ ಎಕ್ಸೆಲ್ 2010 ರಲ್ಲಿ ಫಾರ್ಮ್ಯುಲಾಗಳು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳು (Functions and formulas in MS-Excel 2010)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಮೈಕ್ರೊಸಾಫ್ಟ್ ಎಕ್ಸೆಲ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳು ಮತ್ತು ಕಾರ್ಯಗಳು**
- **ಫಾರ್ಮ್ಯುಲಾಗಳು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳು**
- **ಎಕ್ಸೆಲ್ 2010 ನ್ನು ವಿವರವಾಗಿ ತಿಳಿದುಕೊಳ್ಳುವುದು**
- **ಕಂಡೀಶನಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್**
- **ಎಕ್ಸೆಲ್ ಸ್ಪ್ರೆಡ್‌ಶೀಟ್ ಡಾಟಾವನ್ನು ಲಿಂಕ್ ಮಾಡುವುದು.**

ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿರುವ ಫಾರ್ಮುಲಾಗಳು (Formulas in MS Excel)

ಫಾರ್ಮುಲಾ , ವರ್ಕ್‌ಶೀಟ್ ಇವು ಡಾಟಾದ ಸರಳ ಕೋಷ್ಟಕ ಪ್ರಾತಿನಿಧ್ಯವಾಗಿದೆ. ಒಂದು ಫಾರ್ಮುಲಾ ವಿಶೇಷ ಕೋಡ್‌ಅನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇದನ್ನು ಸೆಲ್‌ನಲ್ಲಿ ನಮೂದಿಸಲಾಗುತ್ತದೆ. ಇದು ಕೆಲವು ಲೆಕ್ಕಗಳನ್ನು ಮಾಡುತ್ತದೆ ಮತ್ತು ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ. ಉತ್ತರವನ್ನು ಸೆಲ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ.

ಫಾರ್ಮುಲಾಗಳು ವಿವಿಧ ಆಪರೇಟರ್‌ಗಳನ್ನು ಬಳಸುತ್ತವೆ ಮತ್ತು ವರ್ಕ್‌ಶೀಟ್ ಫಂಕ್ಷನ್‌ಗಳು ಸಂಖ್ಯೆಗಳು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಜೊತೆಗೆ ಕೆಲಸ ಮಾಡುತ್ತವೆ. ಫಾರ್ಮುಲಾಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಸಂಖ್ಯೆಗಳು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬೇರೆ ಸೆಲ್‌ನಲ್ಲಿ ಇರುವಂಥವಾಗಿರುತ್ತವೆ. ಇದು ಡಾಟಾವನ್ನು ಬದಲಾಯಿಸುವುದನ್ನು ಸುಲಭಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ವರ್ಕ್ ಶೀಟ್‌ಗಳಿಗೆ ಕ್ರಿಯಾತ್ಮಕ ಸ್ವಭಾವವನ್ನು ನೀಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , ಇದು ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಮತ್ತು ಫಾರ್ಮುಲಾಗಳ ಕಾರ್ಯವನ್ನು ತ್ವರಿತವಾಗಿ ಬದಲಾಯಿಸುತ್ತದೆ.

ಫಾರ್ಮುಲಾಗಳ ಅಂಶಗಳು (Elements of Formulas)

ಒಂದು ಫಾರ್ಮುಲಾ ಈ ಕೆಳಗಿನ ಯಾವುದೇ ಅಂಶವನ್ನು ಒಳಗೊಂಡಿರಬಹುದು

- **+ (ಸಂಕಲನ) ಮತ್ತು - (ವ್ಯವಕಲನ) ದಂತಹ ಗಣಿತದ ಆಪರೇಟರ್‌ಗಳು**

ಉದಾಹರಣೆ -

= A1 + A2 ಇದು A1 ಮತ್ತು A2 ಸೆಲ್‌ಗಳಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳನ್ನು ಕೂಡಿಸುತ್ತದೆ.

- **ಸಂಖ್ಯೆಗಳು ಅಥವಾ ಟೆಕ್ಸ್ಟ್**

ಉದಾಹರಣೆ -

= 200 * 0.5 200 ಕ್ಕೆ 0.5 ನಿಂದ ಗುಣಿಸುವುದು. ಈ ಫಾರ್ಮುಲಾ ಸಂಖ್ಯೆಗಳನ್ನು ಮಾತ್ರ ಉಪಯೋಗಿಸುತ್ತದೆ ಮತ್ತು ಇದು ಯಾವಾಗಲೂ 100 ಎಂದು ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ.

- **ಸೆಲ್ ರೆಫರನ್ಸ್ (ಹೆಸರಿಸಲಾದ ಸೆಲ್‌ಗಳನ್ನು ಮತ್ತು ಶ್ರೇಣಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ)**

ಉದಾಹರಣೆ -

= A1 = C12 ಇದು A1 ಸೆಲ್ ಅನ್ನು C12 ಸೆಲ್‌ನೊಂದಿಗೆ ಹೋಲಿಸುತ್ತದೆ . ಸೆಲ್‌ಗಳು ಒಂದೇ ಆಗಿದ್ದರೆ, ಫಾರ್ಮುಲಾ TRUE ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ ಇಲ್ಲದಿದ್ದರೆ, FALSE ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

- **ವರ್ಕ್‌ಶೀಟ್ ಫಂಕ್ಷನ್‌ಗಳು** (SUM **ಅಥವಾ** AVERAGE **ನಂತಹ**)

ಉದಾಹರಣೆ -

= SUM (A1 : A12) ಇದು A1 : A12 ಶ್ರೇಣಿಯಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳನ್ನು ಕೂಡಿಸುತ್ತದೆ.

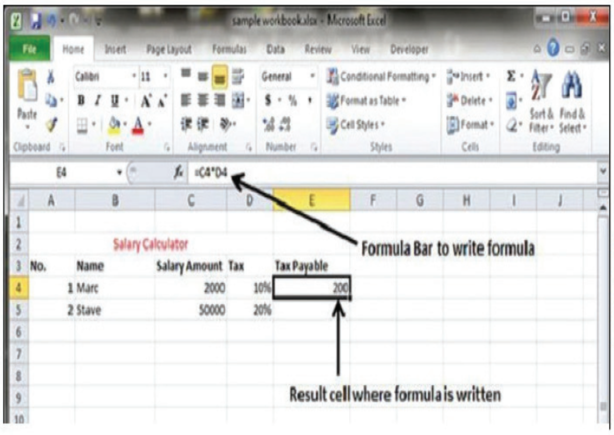
ಫಾರ್ಮುಲಾಗಳನ್ನು ರಚಿಸುವುದು (Creating Formula)

ಫಾರ್ಮುಲಾವನ್ನು ರಚಿಸಲು ಫಾರ್ಮುಲಾ ಬಾರ್‌ನಲ್ಲಿ ಟೈಪ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಫಾರ್ಮುಲಾ, = ಚಿನ್ಹೆಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಫಾರ್ಮುಲಾಗಳನ್ನು ತಯಾರಿಸುವಾಗ ಬಳಕೆದಾರರು ಸೆಲ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಟೈಪ್ ಮಾಡಬಹುದು ಅಥವಾ ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿ ಅವುಗಳನ್ನು ಸೂಚಿಸಬಹುದು. ಫಾರ್ಮುಲಾಗಳಿಗೆ ಸೆಲ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಒದಗಿಸಲು ಪಾಯಿಂಟಿಂಗ್ ವಿಧಾನವು ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಫಾರ್ಮುಲಾಗಳನ್ನು ನಿರ್ಮಿಸುವುದು ಹೆಚ್ಚು ಶಕ್ತಿಶಾಲಿಯಾಗಿದೆ. ಚಿತ್ರ 1 ರಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ - ಅಂತರ್ನಿರ್ಮಿತ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಬಳಸುವಾಗ , ಫಂಕ್ಷನ್ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು , ಫಂಕ್ಷನ್ ಆರ್ಗ್ಯುಮೆಂಟ್ ಡಯಾಲಾಗ್

© NIMI NOT TO BE REPUBLISHED

ಬಾಕ್ಸ್‌ನಲ್ಲಿ ,ಸೆಲ್ ಅನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಅಥವಾ ಸೆಲ್‌ಗಳ ಶ್ರೇಣಿಯನ್ನು ಡ್ರ್ಯಾಗ್ ಮಾಡಿ. (ಚಿತ್ರ 1 ರಂತೆ)

Fig 1



Copa 12039017

ಫಾರ್ಮುಲಾವನ್ನು ನಮೂದಿಸಿದ ತಕ್ಷಣ ಎಕ್ಸೆಲ್ ಉತ್ತರವನ್ನು ಲೆಕ್ಕ ಮಾಡುತ್ತದೆ, ನಂತರ ವರ್ಕ್ಸ್ ಶೀಟ್‌ನಲ್ಲಿರುವ ಸೆಲ್‌ನಲ್ಲಿ ಉತ್ತರವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. (ಅದಾಗ್ಯೂ ,ಫಾರ್ಮುಲಾದ ವಿಷಯಗಳು, ಯಾವುದೆ ಸಮಯದಲ್ಲೂ ಸೆಲ್ ಸಕ್ರೀಯವಾದಾಗ , ಫಾರ್ಮ್ಯುಲಾ ಬಾರ್‌ನಲ್ಲಿ ಕಾಣುತ್ತಿರುತ್ತದೆ), ನೀವು ಫಾರ್ಮುಲಾದಲ್ಲಿ ಏನಾದರೂ ತಪ್ಪು ಮಾಡಿದರೆ , ಎಕ್ಸೆಲ್, ಫಾರ್ಮುಲಾವು ಕೆಲಸ ಮಾಡುವುದನ್ನು ತಡೆಯುತ್ತದೆ , ಎಕ್ಸೆಲ್ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಸೂಚಿಸುವ ಒಂದು ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಅನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಫಾರ್ಮುಲಾದಲ್ಲಿ ಫಂಕ್ಷನ್‌ಗಳು (Functions in Formula)

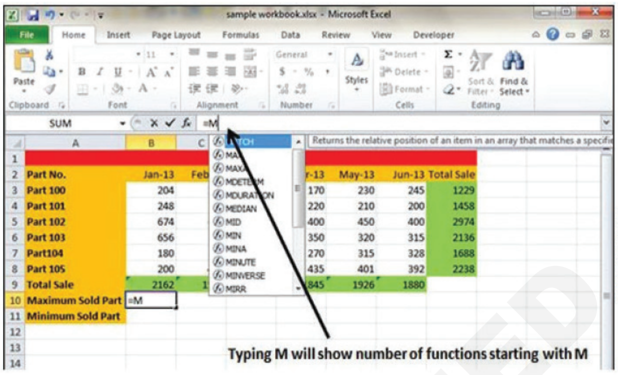
ಲಭ್ಯವಿರುವ ವರ್ಕ್‌ಶೀಟ್ ಫಂಕ್ಷನ್ಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಅನೇಕ ಫಾರ್ಮುಲಾಗಳನ್ನು ರಚಿಸಲಾಗುತ್ತದೆ. ಈ ಫಂಕ್ಷನ್ಗಳು ಫಾರ್ಮುಲಾಗಳ ಶಕ್ತಿಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತವೆ. ಮತ್ತು ಆಪರೇಟರ್‌ಗಳನ್ನು ಮಾತ್ರ ಉಪಯೋಗಿಸಿ ಮಾಡುವ ಕಷ್ಟಕರವಾದ ಲೆಕ್ಕಗಳನ್ನು ಇದು ಸುಲಭವಾಗಿ ಮಾಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಲೊಗ್ಯಾರಿಥಮ್ ಅಥವಾ SIN ಅನುಪಾತವನ್ನು ಲೆಕ್ಕಾಚಾರ ಮಾಡಲು LOG ಅಥವಾ SIN ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಉಪ 'ಯೋಗಿಸಿ ಮಾಡಬಹುದು. ಮತ್ತು ಈ ಸಂಕೀರ್ಣ ಲೆಕ್ಕಾಚಾರವನ್ನು ಗಣಿತದ ಆಪರೇಟರ್‌ಗಳನ್ನು ಮಾತ್ರ ಉಪಯೋಗಿಸಿ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Using Functions)

= ಚಿನ್ಹೆಯನ್ನು ಟೈಪ್ ಮಾಡಿ ಮತ್ತು ಯಾವುದೇ ಅಕ್ಷರವನ್ನು ಟೈಪ್ ಮಾಡಿ. ಚಿತ್ರ 2 ರಲ್ಲಿರುವಂತೆ ಲಭ್ಯವಿರುವ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.

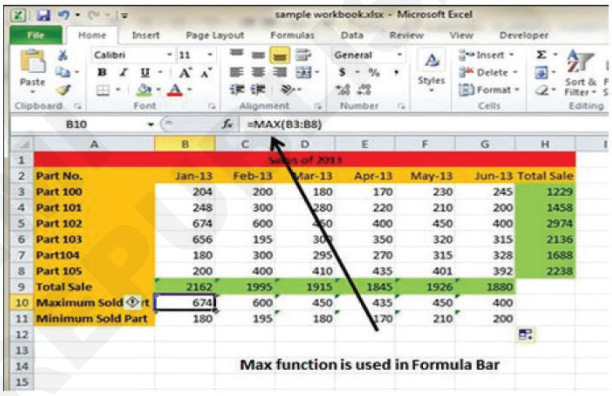
ಒಂದು ಶ್ರೇಣಿಯಲ್ಲಿರುವ ಅತಿ ದೊಡ್ಡ ಸಂಖ್ಯೆಯನ್ನು ನಿರ್ಧರಿಸುವ ಅವಶ್ಯಕತೆ ಇದೆ ಎಂದು ಭಾವಿಸೋಣ. ಫಂಕ್ಷನ್ ಅನ್ನು ಬಳಸದೆ ಫಾರ್ಮುಲಾವು ಉತ್ತರವನ್ನು ಹೇಳಲು ಸಾಧ್ಯವಿಲ್ಲ. B3 : B6 ಶ್ರೇಣಿಯಲ್ಲಿರುವ ಅತಿ ದೊಡ್ಡ ಸಂಖ್ಯೆಯನ್ನು ಕಂಡುಹಿಡಿಯಲು ನಾವು MAX ಫಂಕ್ಷನ್ ಅನ್ನು ಬಳಸುವ ಫಾರ್ಮುಲಾವನ್ನು ಉಪಯೋಗಿಸೋಣ. ಚಿತ್ರ 3 ರಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ ಫಂಕ್ಷನ್ = MAX (A1 : D100)

Fig 2



Copa 12039018

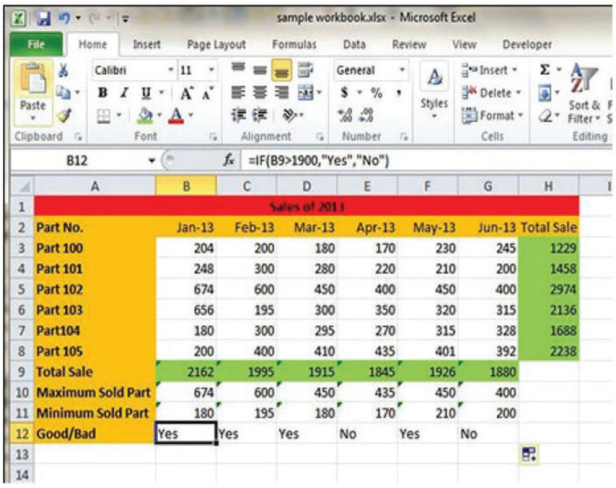
Fig 3



Copa 12039019

ಫಂಕ್ಷನ್‌ಗಳ ಮತ್ತೊಂದು ಉದಾಹರಣೆ. ತಿಂಗಳ ಸೇಲ್ 1900 ಕ್ಕಿಂತ ಹೆಚ್ಚಿರುವುದು ಕಂಡು ಬಂದರೆ, ನಾವು ಸೇಲ್ಸ್ ರೀಪ್ರೆಸೆಂಟೇಟಿವ್ ಗೆ ಬೋನಸ್‌ಅನ್ನು ನೀಡಬಹುದು. ಇದನ್ನು ನಾವು IF ಫಂಕ್ಷನ್ ಬಳಸಿ ಫಾರ್ಮುಲಾ ಬರೆದು ಸಾಧಿಸಬಹುದು. ಚಿತ್ರ 4 ರಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ ಫಂಕ್ಷನ್ = IF (B9>1900, Yes , No)

Fig 4



Copa 12039020

© NIMI NOT TO BE REPUBLISHED

ಫಂಕ್ಷನ್‌ನ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು (Function Arguments)

ಮೇಲಿನ ಉದಾಹರಣೆಗಳಲ್ಲಿ ಎಲ್ಲ ಫಂಕ್ಷನ್‌ಗಳು ಆವರಣ (Parantheses) ವನ್ನು ಬಳಸುತ್ತವೆ ಎಂಬುದನ್ನು ಗಮನಿಸಿ. ಈ ಆವರಣದಲ್ಲಿನ ಮಾಹಿತಿ ಎಂದರೆ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳ ಪಟ್ಟಿ.

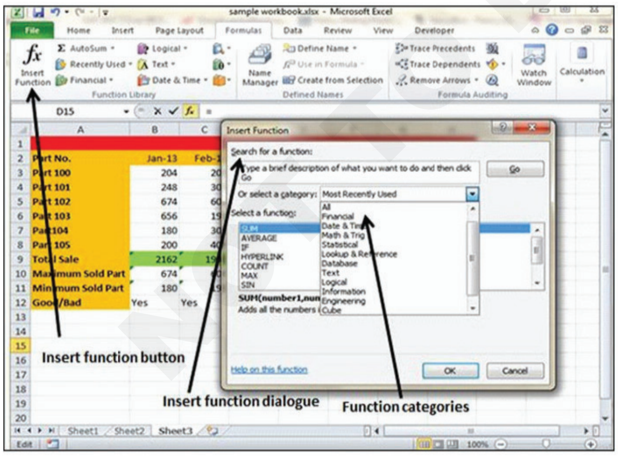
ಅವು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಹೇಗೆ ಬಳಸುತ್ತವೆ ಎಂಬುದರ ಮೇಲೆ ಫಂಕ್ಷನ್‌ಗಳು ಬದಲಾಗುತ್ತವೆ. ಫಂಕ್ಷನ್ ಏನು ಕೆಲಸ ಮಾಡಬೇಕು ಎಂಬುದರ ಮೇಲೆ ನೀವು ಬಳಸುವ ಫಂಕ್ಷನ್ ಅವಲಂಬಿಸಿರುತ್ತದೆ.

- ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು ಇಲ್ಲ - ಉದಾಹರಣೆಗಳು - Now ( ), Date ( ), ಇತ್ಯಾದಿ
- ಒಂದು ಆರ್ಗ್ಯುಮೆಂಟ್ - UPPER( ) , LOWER ( ) ಇತ್ಯಾದಿ
- ನಿಗದಿತ ಸಂಖ್ಯೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು - IF ( ), MAX ( ), MIN (), AVERAGE( ), ಇತ್ಯಾದಿ
- ಅನಂತ ಸಂಖ್ಯೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು
- ಐಚ್ಛಿಕ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು

ಬಿಲ್ಟ್ ಇನ್ ಫಂಕ್ಷನ್‌ಗಳು (Built in Functions)

ಎಂಎಸ್ ಎಕ್ಸೆಲ್ ಅನೇಕ ಬಿಲ್ಟ್ ಇನ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಇವುಗಳನ್ನು ನಾವು ಫಾರ್ಮುಲಾಗಳಲ್ಲಿ ಬಳಸಬಹುದು. ಕ್ಯಾಟ್ಯಾಗರಿ (ವರ್ಗದ) ಪ್ರಕಾರ ಎಲ್ಲ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ನೋಡಬೇಕಾದರೆ ಚಿತ್ರ 6 ರಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ Formula Tab > Insert function ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿ. ನಂತರ ಇನ್ಸರ್ಟ್ ಫಂಕ್ಷನ್ ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ. ಇದರಲ್ಲಿ ನಾವು ನಮಗೆ ಬೇಕಾದ ಫಂಕ್ಷನ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

Fig 5



Copa 12039021

ಫಂಕ್ಷನ್‌ಗಳ ವರ್ಗಗಳು (Functions by Catagories)

ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿರುವ ಕೆಲವು ಬಿಲ್ಟ್ ಇನ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ನೋಡೋಣ

ಟೆಕ್ಸ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳು(Text Functions)

LOWER : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಎಲ್ಲ ಅಕ್ಷರಗಳನ್ನು ಲೋವರ್ ಕೇಸ್‌ಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ.

UPPER : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಎಲ್ಲ ಅಕ್ಷರಗಳನ್ನು ಅಪ್ಪರ ಕೇಸ್‌ಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ.

TRIM: ನಕಲಿ ಸ್ಥಳಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಪ್ರಾರಂಭದಲ್ಲಿ ಮತ್ತು ಕೊನೆಯಲ್ಲಿ ಇರುವ ಖಾಲಿ ಸ್ಥಳವನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

CONCATENATE : ಎರಡು ಮತ್ತು ಎರಡಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಜೋಡಿಸುತ್ತದೆ.

LEFT: ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಪ್ರಾರಂಭದಿಂದ ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಅಕ್ಷರಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

MID : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಮಧ್ಯದಿಂದ ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಅಕ್ಷರಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

RIGHT : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಕೊನೆಯಿಂದ ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಅಕ್ಷರಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

LEN : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ಅಕ್ಷರಗಳ ಮೊತ್ತವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

FIND : ಕೊಡಲಾದ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ , ನೀವು ಹುಡುಕಲು ನೀಡಿದ ಅಕ್ಷರ ಅಥವಾ ಸ್ಟ್ರಿಂಗ್‌ನ ಸ್ಥಾನವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Date & Time (ದಿನಾಂಕ ಮತ್ತು ಸಮಯ)

DATE : ಬಳಕೆದಾರರು ಪೂರೈಸಿದ ವರ್ಷ, ತಿಂಗಳು ಮತ್ತು ದಿನದಿಂದ ದಿನಾಂಕವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

TIME : ಬಳಕೆದಾರರು ಪೂರೈಸಿದ ಗಂಟೆ, ನಿಮಿಷ ಮತ್ತು ಸೆಕೆಂಡ್‌ನಿಂದ ಸಮಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

DATEVALUE : ದಿನಾಂಕವನ್ನು ತೋರಿಸುವ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು , ಎಕ್ಸೆಲ್‌ನ ಡೇಟ್ ಟೈಮ್ ಕೋಡ್‌ನಲ್ಲಿ ದಿನಾಂಕವನ್ನು ಪ್ರತಿನಿಧಿಸುವ ಒಂದು ಪೂರ್ಣಾಂಕಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ

TIMEVALUE : ಸಮಯವನ್ನು ತೋರಿಸುವ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು , ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಸಮಯವನ್ನು ಪ್ರತಿನಿಧಿಸುವ ದಶಮಾಂಶಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ.

NOW : ಪ್ರಸ್ತುತ ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ತೋರಿಸುತ್ತದೆ.

TODAY : ಇವತ್ತಿನ ದಿನಾಂಕವನ್ನು ತೋರಿಸುತ್ತದೆ.

Statistical (ಅಂಕಿ ಅಂಶಗಳು)

MAX : ಕೊಡಲಾದ ಸಂಖ್ಯೆಗಳ ಪಟ್ಟಿಯಿಂದ ದೊಡ್ಡ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

MIN : ಕೊಡಲಾದ ಸಂಖ್ಯೆಗಳ ಪಟ್ಟಿಯಿಂದ ಚಿಕ್ಕ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

AVERAGE : ಕೊಡಲಾದ ಸಂಖ್ಯೆಗಳ ಪಟ್ಟಿಯ ಸರಾಸರಿ (Average)ಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

COUNT : ಕೊಡಲಾದ ಸೆಲ್‌ಗಳ ಅಥವಾ ಮೌಲ್ಯಗಳ ಗುಂಪಿನಲ್ಲಿರುವ ಸಂಖ್ಯಾತ್ಮಕ ಮೌಲ್ಯಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

COUNTIF : ನೀವು ಕೊಟ್ಟ ನಿರ್ದಿಷ್ಟ ಮಾನದಂಡಗಳನ್ನು ಪೂರೈಸುವ ಸೆಲ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು (ಕೊಡಲಾದ ಸೆಲ್ ಶ್ರೇಣಿ) ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

SUM: ಕೊಡಲಾದ ಸಂಖ್ಯೆಗಳ ಪಟ್ಟಿಯ ಮೊತ್ತವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Logical (ಲಾಜಿಕಲ್)

AND : ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಹಲವಾರು ಶರತ್ತುಗಳನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ ಮತ್ತು ಎಲ್ಲ ಶರತ್ತುಗಳು ನಿಜ (TRUE) ಎಂದಾದರೆ TRUE ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಇಲ್ಲವಾದರೆ FALSE ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

OR : ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಹಲವಾರು ಶರತ್ತುಗಳನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ ಮತ್ತು ಯಾವುದೇ ಶರತ್ತುಗಳು ನಿಜ (TRUE) ಎಂದಾದರೆ TRUE ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಇಲ್ಲವಾದರೆ FALSE ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

NOT : ಬಳಕೆದಾರರು ಪೂರೈಸಿದ ಲಾಜಿಕಲ್ ಸಂಖ್ಯೆ ಅಥವಾ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ ವಿರುದ್ಧವಾದ ಲಾಜಿಕಲ್ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಎಂದರೆ, ಕೊಡಲಾದ ಆರ್ಗ್ಯುಮೆಂಟ್ TRUE ಇದ್ದರೆ FALSE ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ ಮತ್ತು ಕೊಡಲಾದ ಆರ್ಗ್ಯುಮೆಂಟ್ FALSE ಇದ್ದರೆ TRUE ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ

Math & Trig (ಮ್ಯಾಥ್ ಮತ್ತು ಟ್ರಿಗ್)

ABS : ಕೊಡಲಾದ ಸಂಖ್ಯೆಯ ಅಬ್ಸಲ್ಯುಟ್ ಸಂಖ್ಯೆ(ಪೂರ್ಣ ಸಂಖ್ಯೆ) ಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

SIGN : ಕೊಡಲಾದ ಸಂಖ್ಯೆಯ ಚಿಹ್ನೆಯನ್ನು (+1, -1, ಅಥವಾ 0) ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

SQRT : ಕೊಡಲಾದ ಸಂಖ್ಯೆಯ ಧನಾತ್ಮಕ ವರ್ಗಮೂಲವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

MOD : ಕೊಡಲಾದ ಎರಡು ಸಂಖ್ಯೆಗಳ ವಿಭಜನೆಯಿಂದ ಬಂದ ಶೇಷವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

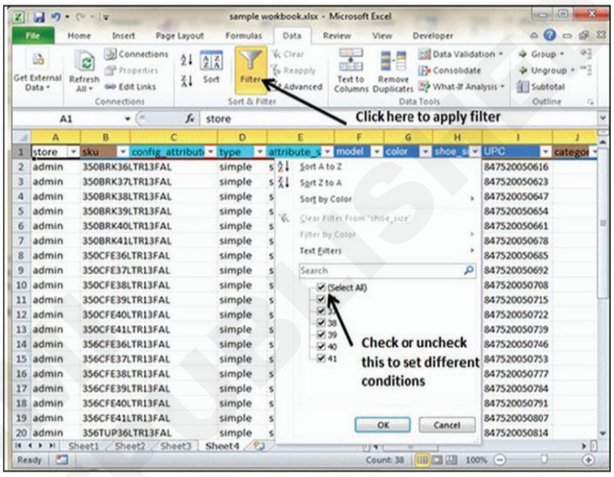
ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಫಿಲ್ಟರ್ಸ್ (Filters in MS Excel)

ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಡಾಟಾವನ್ನು ಫಿಲ್ಟರ್ ಮಾಡುವುದು ಎಂದರೆ ಕೆಲವು ಷರತ್ತುಗಳನ್ನು ಪೂರೈಸುವ ರೋ ಗಳನ್ನು ಮಾತ್ರ ಪ್ರದರ್ಶಿಸುವುದು. (ಉಳಿದ ರೋ ಗಳನ್ನು ಮರೆಮಾಡಲಾಗುತ್ತದೆ).

ಸಂಗ್ರಹ ಮಾಡಿದ ಡಾಟಾ ಅನ್ನು ಉಪಯೋಗಿಸಿ , ಬಳಕೆದಾರರು 36 ಸೈಝ್‌ನ ಶೂ ಅನ್ನು ನೋಡಲು ಬಯಸಿದರೆ, ಇದಕ್ಕಾಗಿ ಫಿಲ್ಟರ್ ಅನ್ನು ಸೆಟ್ ಮಾಡಿ. ಇದನ್ನು ಮಾಡಲು ಈ ಕೆಳಗಿನ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿ .__

- ಹೆಡರ್ ರೋ ಮೇಲೆ ಕರ್ಸರ್ ಅನ್ನು ಇರಿಸಿ
- Data > Filter ಅನ್ನು ಆಯ್ಕೆಮಾಡಿ. ಚಿತ್ರ 6 ರಲ್ಲಿ ತೋರಿಸಿದಂತೆ ಫಿಲ್ಟರ್ ಅನ್ನು ಸೆಟ್ ಮಾಡಿರಿ.

Fig 6



Copa 12039022

- ಏರಿಯಾ ರೋ ಹೆಡರ್‌ನಲ್ಲಿನ ಡ್ರಾಪ್ ಡೌನ್ ಬಾಣದ ಚಿಹ್ನೆಯನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು SelectAll ನ ಚೆಕ್ ಚಿಹ್ನೆಯನ್ನು ತೆಗೆಯಿರಿ. ಇದು ಎಲ್ಲವನ್ನು ಅನ್ ಸೆಲೆಕ್ಟ್ ಮಾಡುತ್ತದೆ.
- ನಂತರ ಸೈಝ್ 36 ನ ಚೆಕ್ ಮಾರ್ಕ್ ಅನ್ನು ಆಯ್ಕೆಮಾಡಿ. ಇದು ಡಾಟಾವನ್ನು ಫಿಲ್ಟರ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಶೂ ಸೈಝ್ 36 ರ ಡಾಟಾವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ
- ಕೆಲವು ರೋ ಸಂಖ್ಯೆಗಳು ಕಾಣೆಯಾಗಿರುತ್ತದೆ. ಈ ಸಾಲುಗಳು ಫಿಲ್ಟರ್(ಮರೆಯಾದ) ಮಾಡಿದ ಡಾಟಾಅನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.
- ಏರಿಯಾ ಕಾಲಮ್‌ನಲ್ಲಿ ಒಂದು ಡ್ರಾಪ್ ಡೌನ್ ಬಾಣದ ಚಿಹ್ನೆ ಇರುತ್ತದೆ. ಈಗ ಅದು ವಿಭಿನ್ನ ಗ್ರಾಫಿಕ್ - ಒಂದು ಐಕನ್ ಅನ್ನು ತೋರಿಸುತ್ತದೆ. ಇದು ಈ ಕಾಲಮ್ ಫಿಲ್ಟರ್ ಆಗಿದೆ ಎಂದು ಸೂಚಿಸುತ್ತದೆ.(ಚಿತ್ರ 7)

ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಫಿಲ್ಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು(Using Multiple Filters) (Fig 8)

ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಷರತ್ತುಗಳಿಂದ ರೆಕಾರ್ಡಗಳನ್ನು ಫಿಲ್ಟರ್ ಮಾಡುವುದು ಎಂದರೆ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಕಾಲಮ್ ಮೌಲ್ಯಗಳಿಂದ

© NIMI NOT TO BE REPUBLISHED

ಫಿಲ್ಟರ್ ಮಾಡುವುದು. ಶೂ ಸೈಝ್ 36 ಅನ್ನು ಫಿಲ್ಟರ್ ಮಾಡಿದ ನಂತರ ಕಾಫೀ ಕಲರ್‌ಗೆ ಸಮನಾದ ಕಲರನ್ನು ಫಿಲ್ಟರ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇದೆ ಎಂದು ಭಾವಿಸೋಣ. ಶೂ ಸೈಝ್‌ಗೆ ಫಿಲ್ಟರ್ ಸೆಟ್ ಮಾಡಿದ ನಂತರ , ಕಲರ್ ಕಾಲಮ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ಫಿಲ್ಟರ್ ಅನ್ನು ಕಲರ್‌ಗೆ ಸೆಟ್ ಮಾಡಿರಿ.(ಚಿತ್ರ 8)

Fig 7

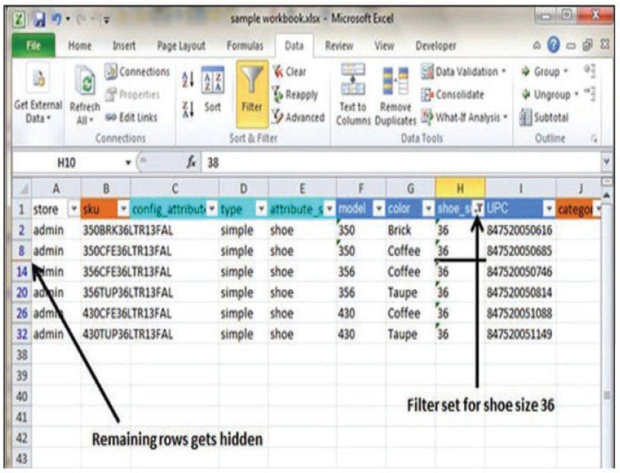
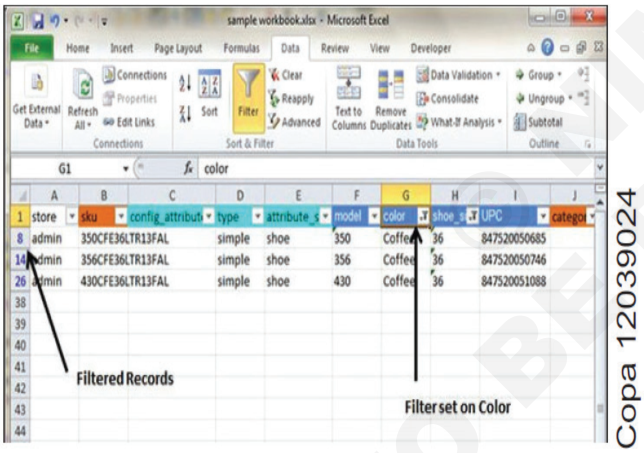


Fig 8



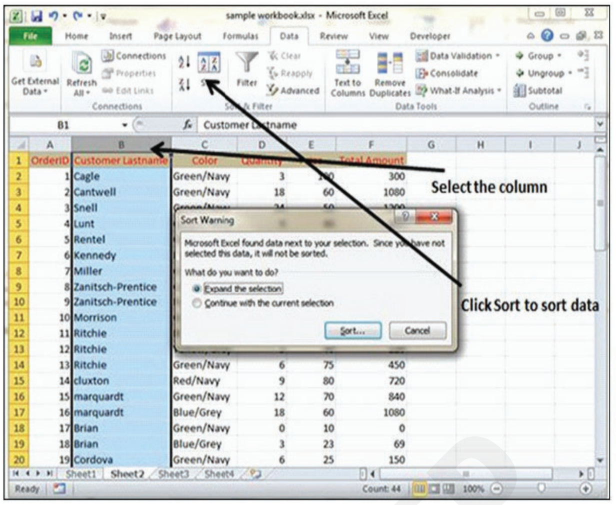
ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ವಿಂಗಡಿಸುವುದು (Sorting in MS Excel)

ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ವಿಂಗಡಿಸುವುದರಿಂದ , ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಾಲಮ್‌ನಲ್ಲಿರುವ ವಿಷಯಗಳ ಆಧಾರದ ಮೇಲೆ ಸಾಲುಗಳನ್ನು ಮರುಹೊಂದಿಸುತ್ತದೆ. ಹೆಸರುಗಳನ್ನು ವರ್ಣಮಾಲೆಯ ಅಕ್ಷರಗಳಂತೆ ಒಂದು ಟೇಬಲ್ ಅನ್ನು ವಿಂಗಡಿಸಿ ಅಥವಾ ಸಣ್ಣ ಮೊತ್ತದಿಂದ ದೊಡ್ಡ ಮೊತ್ತಕ್ಕೆ ಅಥವಾ ದೊಡ್ಡ ಮೊತ್ತದಿಂದ ಸಣ್ಣ ಮೊತ್ತಕ್ಕೆ ಡಾಟಾವನ್ನು ವಿಂಗಡಿಸಿ.

ಡಾಟಾ ಅನ್ನು ಸಾರ್ಟ್ ಮಾಡಲು (ವಿಂಗಡಿಸಲು) ಕೆಳಗೆ ತಿಳಿಸಲಾದ ಹಂತಗಳನ್ನು ಅನುಸರಿಸಿ.

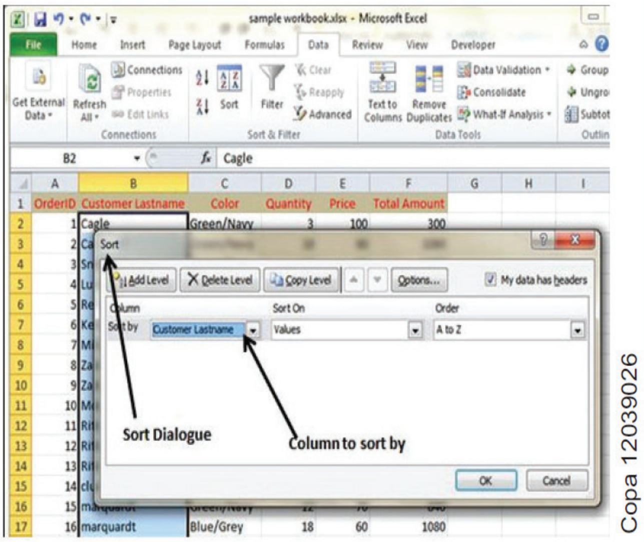
- ಚಿತ್ರ 9 ರಲ್ಲಿರುವಂತೆ ಡಾಟಾವನ್ನು ಸಾರ್ಟ್ ಮಾಡಲು ಕಾಲಮ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

Fig 9



- Data Tab ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ, Sort below ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಕಾಣುತ್ತದೆ
- ಬಳಕೆದಾರರು ಆಯ್ದ ಕಾಲಮ್‌ನ ಆಧಾರದ ಮೇಲೆ ಡಾಟಾವನ್ನು ವಿಂಗಡಿಸಿದರೆ, Continue with the selection ಅನ್ನು ಆಯ್ಕೆಮಾಡಿ ಅಥವಾ ಇತರ ಕಾಲಮ್‌ಗಳ ಆಧಾರದ ಮೇಲೆ ವಿಂಗಡಿಸುವುದಾದರೆ , Expand Selection ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿ.
- ಚಿತ್ರ 10 ರಲ್ಲಿರುವಂತೆ ಕೆಳಗಿನ ಷರತ್ತುಗಳ ಆಧಾರದ ಮೇಲೆ ಡಾಟಾವನ್ನು ವಿಂಗಡಿಸಿ.

Fig 10

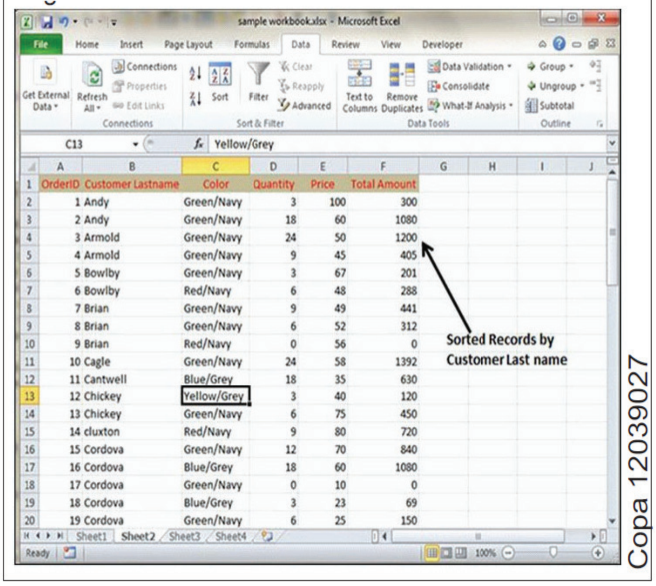


- Values - ವರ್ಣಮಾಲೆಯಂತೆ ಅಥವಾ ಸಂಖ್ಯಾತ್ಮಕವಾಗಿ (alphabetically or numerically)
- Cell color - ಸೆಲ್ ನ ಬಣ್ಣವನ್ನು ಆಧರಿಸಿದೆ
- Font Color - ಫಾಂಟ್ ಬಣ್ಣವನ್ನು ಆಧರಿಸಿದೆ
- Cell Icon - ಸೆಲ್ ಐಕನ್ ಮೇಲೆ ಆಧರಿಸಿದೆ

© NIMI NOT TO BE REPUBLISHED

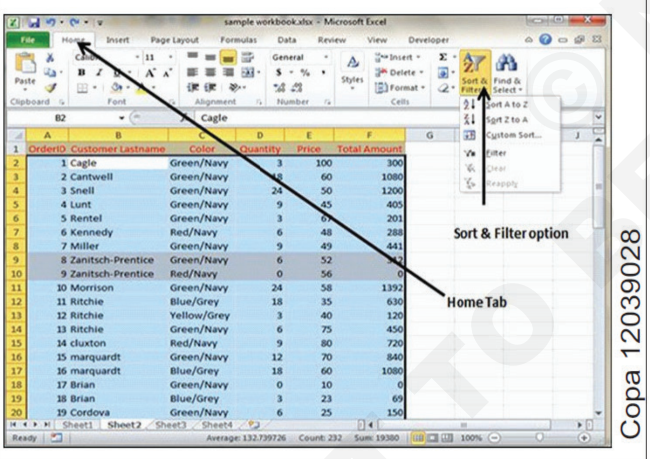
- ಚಿತ್ರ 11 ರಲ್ಲಿರುವಂತೆ OK ಕ್ಲಿಕ್ ಮಾಡುವುದರಿಂದ ಡಾಟಾ ವಿಂಗಡನೆಯಾಗುತ್ತದೆ.

Fig 11



ಸಾರ್ಟಿಂಗ್ ಆಯ್ಕೆಯು ಹೋಮ್ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಕೂಡ ಲಭ್ಯವಿದೆ. Home tab > Sort & Filter ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರ 12 ರಲ್ಲಿರುವಂತೆ ನೀವು ರೆಕಾರ್ಡ್ ಗಳನ್ನು ವಿಂಗಡಿಸಲು ಅದೇ ಡಯಾಲಾಗ್ ಅನ್ನು ನೋಡಬಹುದು.

Fig 12



ಎಂಎಸ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಶ್ರೇಣಿಗಳು (Ranges in MS Excel) :

ವರ್ಕ್ ಶೀಟ್‌ನಲ್ಲಿ ಸೆಲ್ ಒಂದು ಅಂಶವಾಗಿದೆ, ಇದು ಒಂದು ಸಂಖ್ಯೆ, ಟೆಕ್ಸ್ಟ್ ಅಥವಾ ಫಾರ್ಮುಲಾವನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಸೆಲ್ ಅನ್ನು ಅದರ ಅಡ್ರೆಸ್ ಮೂಲಕ ಗುರುತಿಸಲಾಗುತ್ತದೆ. ಸೆಲ್ ಅಡ್ರೆಸ್ ಅದರ ಕಾಲಮ್ ಅಕ್ಷರ ಮತ್ತು ರೋ ನ ಸಂಖ್ಯೆಯನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , ಸೆಲ್ B1 ಇದು ಎರಡನೆ ಕಾಲಮ್ ಮತ್ತು ಮೊದಲನೆ ರೋ ನಲ್ಲಿರುವ ಸೆಲ್ ಆಗಿರುತ್ತದೆ.

ಸೆಲ್‌ಗಳ ಗುಂಪನ್ನು ರೇಂಜ್ (ಶ್ರೇಣಿ) ಎಂದು ಕರೆಯುತ್ತಾರೆ. ನೀವು ಒಂದು ಸೆಲ್ ರೇಂಜ್ ಅಡ್ರೆಸ್ ಅನ್ನು ಅದರ ಮೇಲಿನ ಎಡ ಸೆಲ್ ಅಡ್ರೆಸ್ ಮತ್ತು ಅದರ ಕೆಳಗಿನ ಬಲ ಸೆಲ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಕೋಲನ್ ಚಿನ್ಹೆಯಿಂದ ಬೇರ್ಪಡಿಸಿ ನಿರ್ದಿಷ್ಟಪಡಿಸುವ ಮೂಲಕ ಗುರುತಿಸಬಹುದು.

ರೇಂಜ್‌ಗೆ ಉದಾಹರಣೆಗಳು (Examples of Ranges) :

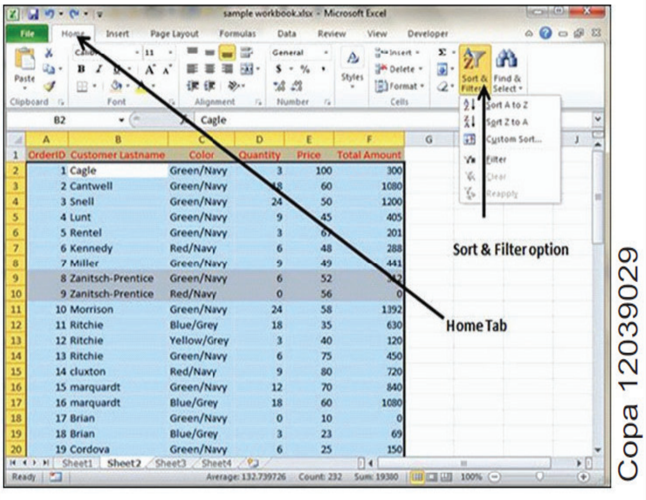
- C24 - ಒಂದೇ ಸೆಲ್ ಅನ್ನು ಒಳಗೊಂಡಿರುವ ರೇಂಜ್
- A1: B1 - ಒಂದು ರೋ ಮತ್ತು ಎರಡು ಕಾಲಮ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಎರಡು ಸೆಲ್‌ಗಳು
- A1: A100 - ಕಾಲಮ್ A ನಲ್ಲಿ 100 ಸೆಲ್‌ಗಳು
- A1:D4 - 16 ಸೆಲ್‌ಗಳು (ನಾಲ್ಕು ರೋ ಗಳು ಮತ್ತು ನಾಲ್ಕು ಕಾಲಮ್‌ಗಳು)

ರೇಂಜ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡುವುದು (Selecting Ranges)

ಹಲವಾರು ರೀತಿಯಿಂದ ರೇಂಜ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಬಹುದು.

- ಮೌಸ್‌ನ ಎಡಗಡೆಯ ಬಟನ್ ಅನ್ನು ಒತ್ತಿರಿ ಮತ್ತು ಎಳೆಯಿರಿ. ರೇಂಜ್ ಹೈಲೈಟ್ ಆಗುತ್ತದೆ. ನಂತರ ಮೌಸ್ ಬಟನ್ ಅನ್ನು ಬಿಡಿರಿ. ಸ್ಕ್ರೀನ್‌ನ ಕೊನೆಯವರೆಗೂ ಎಳೆದರೆ , ವರ್ಕ್‌ಶೀಟ್ ಸ್ಕ್ರಾಲ್ ಆಗುತ್ತದೆ.
- ಶಿಫ್ಟ್ ಕೀಯನ್ನು ಒತ್ತಿರಿ ಮತ್ತು ನ್ಯಾವಿಗೇಷನ್ ಕೀ ಗಳನ್ನು ಉಪಯೋಗಿಸಿ, ರೇಂಜ್‌ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ
- F8 ಕೀ ಯನ್ನು ಒತ್ತಿರಿ ಮತ್ತು ನಂತರ ಸೆಲ್ ಪಾಯಿಂಟರ್ ಅನ್ನು ನ್ಯಾವಿಗೇಷನ್ ಕೀ ಗಳೊಂದಿಗೆ ಚಲಿಸಿ ಮತ್ತು ರೇಂಜ್ ಅನ್ನು ಹೈಲೈಟ್ ಮಾಡಿರಿ. ನ್ಯಾವಿಗೇಷನ್ ಕೀ ಗಳನ್ನು ಸಾಮಾನ್ಯ ಚಲನೆಗೆ ಹಿಂತಿರುಗಿಸಲು ಮತ್ತೊಮ್ಮೆ F8 ಕೀ ಯನ್ನು ಒತ್ತಿರಿ.
- ನೇಮ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಸೆಲ್ ಅಥವಾ ರೇಂಜ್ ಅಡ್ರೆಸ್ ಅನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಮತ್ತು Enter ಕೀಯನ್ನು ಒತ್ತಿರಿ. ಎಕ್ಸೆಲ್ ನೀವು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಸೆಲ್ ಅಥವಾ ರೇಂಜ್‌ಅನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ. (ಚಿತ್ರ 13)

Fig 13



NOT TO BE REPUBLISHED © NIMI

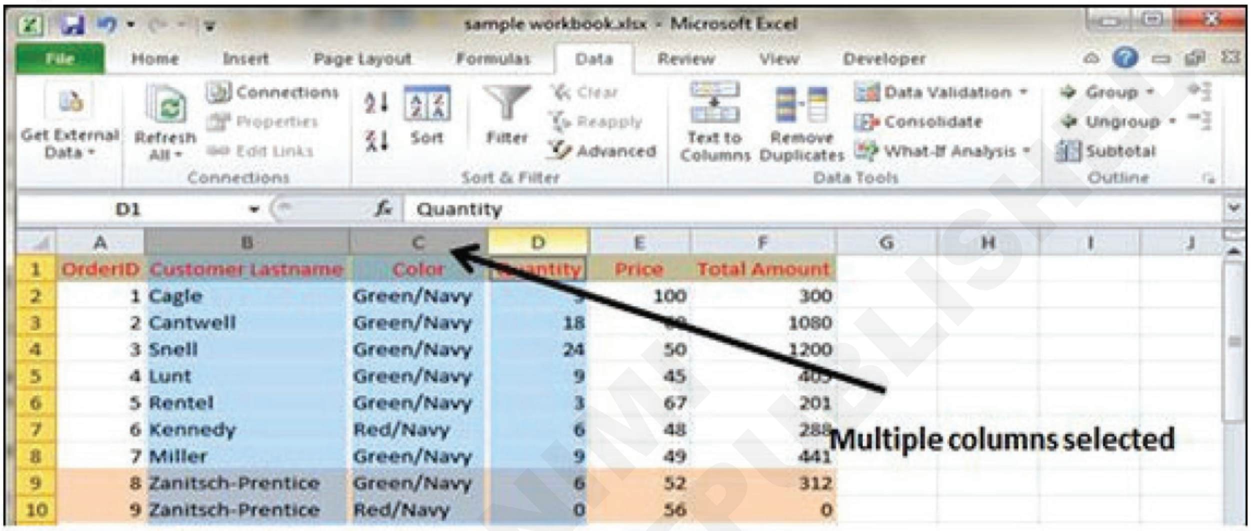
ಸಂಪೂರ್ಣ ರೋ ಮತ್ತು ಕಾಲಮ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡುವುದು

(Selecting complete Rows and Columns)

ಬಳಕೆದಾರರು ಸಂಪೂರ್ಣ ರೋ ಅಥವಾ ಕಾಲಮ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಬೇಕಾದರೆ ಮತ್ತು ಸೆಲ್ ರೇಂಜ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ರೀತಿಯಲ್ಲಿಯೆ ಸಂಪೂರ್ಣ ರೋ ಗಳನ್ನು ಮತ್ತು ಕಾಲಮ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಬಹುದು :

- ಒಂದೇ ರೋ ಅಥವಾ ಕಾಲಮ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ರೋ ಅಥವ ಕಾಲಮ್ ಬಾರ್ಡರ್ ಅನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ
- ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಪಕ್ಕದ ರೋ ಅಥವಾ ಕಾಲಮ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಲು, ರೋ ಅಥವಾ ಕಾಲಮ್ ಬಾರ್ಡರ್ ಅನ್ನು ಕ್ಲಿಕ್‌ಮಾಡಿ ಮತ್ತು ಹೆಚ್ಚುವರಿ ರೋ ಅಥವಾ ಕಾಲಮ್‌ಗಳನ್ನು ಹೈಲೈಟ್ ಮಾಡಲು ಎಳೆಯಿರಿ.
- ಅನೇಕ (ಪಕ್ಕದಲ್ಲದ) ರೋ ಅಥವಾ ಕಾಲಮ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಲು, Ctrl ಕೀ ಅನ್ನು ಒತ್ತಿರಿ ಹಾಗೆಯೆ ರೋ ಅಥವ ಕಾಲಮ್‌ಗಳ ಬಾರ್ಡರ್ ಅನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ. (ಚಿತ್ರ 14)

Fig 14



Copa 12039030

© NIMI NOT TO BE REPUBLISHED

ಚಾರ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Manage Charts)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಚಾರ್ಟ್‌ಗಳನ್ನು ರಚಿಸುವುದು
- ಚಾರ್ಟ್‌ಗಳನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುವುದು
- ಚಾರ್ಟ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು.

ಚಾರ್ಟ್ (Chart)

ಡಾಟಾವನ್ನು ವಿವರಿಸಲು ಮತ್ತು ಹೋಲಿಕೆ ಮಾಡಲು ಚಾರ್ಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಚಾರ್ಟ್‌ನ ಕೆಲವು ವಿಧಗಳು ಯಾವವೆಂದರೆ - ಬಾರ್, ಪೈ, ಲೈನ್, ಏರಿಯಾ ಮತ್ತು ಸರ್ಫೇಸ್ ಚಾರ್ಟ್‌ಗಳು

ನಿಮ್ಮ ಡಾಟಾಗೆ ಅವಶ್ಯಕವಿರುವ ಚಾರ್ಟ್‌ನ್ನು (ಗ್ರಾಫ್) ನೀವು ಚಾರ್ಟ್ ವಿಝಾರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ರಚಿಸುವಷ್ಟೆ ವೇಗವಾಗಿ ರಚಿಸಬಹುದು. ಆದರೆ ಚಾರ್ಟ್ ವಿಝಾರ್ಡ್ ಈಗ ಲಭ್ಯವಿರುವುದಿಲ್ಲ.

ಚಾರ್ಟ್‌ನ್ನು ರಚಿಸುವುದು (Create a chart)

1 ನೀವು ಚಾರ್ಟ್ ರಚಿಸಲು ಬಯಸುವ ಡಾಟಾ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿ.

2 INSERT > Recommended Charts ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Recommended Charts ಟ್ಯಾಬ್‌ನಲ್ಲಿ, ಎಕ್ಸೆಲ್ ನಿಮ್ಮ ಡಾಟಾಗೆ ಸೂಚಿಸಿದ ಚಾರ್ಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಸ್ಕ್ರಾಲ್ ಮಾಡಿರಿ ಮತ್ತು ನಿಮ್ಮ ಡಾಟಾ ಹೇಗೆ ಕಾಣುತ್ತದೆ ಎಂಬುದನ್ನು ನೋಡಲು ಯಾವುದಾದರೂ ಒಂದು ಚಾರ್ಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ನಿಮಗೆ ಇಷ್ಟವಾಗುವ ಚಾರ್ಟ್ ಕಾಣದಿದ್ದರೆ, ಎಲ್ಲ ಲಭ್ಯವಿರುವ ಚಾರ್ಟ್‌ಗಳನ್ನು ನೋಡಲು All Charts ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

4 ನೀವು ಬಯಸಿದ ಚಾರ್ಟ್ ಸಿಕ್ಕ ನಂತರ, OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

5 ಚಾರ್ಟ್‌ನ ಮೇಲ್ಭಾಗದ ಬಲ ತುದಿಯಲ್ಲಿರುವ Chart Elements, Chart Styles ಮತ್ತು Chart Filters buttons ಮುಂತಾದುವುಗಳನ್ನು ಉಪಯೋಗಿಸಿರಿ. ಇವುಗಳನ್ನು ಬಳಸಿ axis title ಅಥವಾ data label ನಂತಹ ಚಾರ್ಟ್ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು ಮತ್ತು ನಿಮ್ಮ ಚಾರ್ಟ್‌ನ ನೋಟವನ್ನು ಬದಲಾಯಿಸಬಹುದು ಅಥವಾ ಚಾರ್ಟ್‌ನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಬದಲಾಯಿಸಬಹುದು.

6 ಹೆಚ್ಚುವರಿ ಡಿಸೈನ್‌ಗಳನ್ನು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಪಡೆಯಲು, ಚಾರ್ಟ್‌ನ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ CHART TOOLS ನ್ನು ರಿಬ್ಬನ್‌ಗೆ ಸೇರಿಸಬಹುದು. ಮತ್ತು ನಂತರ DESIGN ಮತ್ತು FORMAT ನಲ್ಲಿನ ನೀವು ಬಯಸಿದ ಆಯ್ಕೆಗಳ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಎಕ್ಸೆಲ್ ಚಾರ್ಟ್‌ನಲ್ಲಿ ಡಾಟಾ ಸಿರೀಸ್ ಎಂದರೇನು ? (What is data series in Excel chart?)

ಎಕ್ಸೆಲ್ ಚಾರ್ಟ್‌ನಲ್ಲಿ ಡಾಟಾ ಸಿರೀಸ್ ಎಂದರೇನು ? ಒಂದು ಸ್ಪ್ರೆಡ್‌ಶೀಟ್‌ನ ಕಾಲಂನಲ್ಲಿರುವ ಕ್ರಮಾನುಗತ ಸಂಖ್ಯೆಗಳಿಗೆ ಡಾಟಾ ಸಿರೀಸ್ ಎನ್ನುತ್ತಾರೆ. ನಾವು ಡಾಟಾ ಸಿರೀಸ್ ಅನ್ನು ಡಾಟಾದ ಪ್ರಾರಂಭಿಕ ಸರಣಿಯಿಂದ ಗ್ರಾಫನ್ನು ರಚಿಸುವ ಮೂಲಕ ರಚಿಸಬಹುದು ನಂತರ ನಾವು ಮತ್ತೊಂದು ಡಾಟಾದ ಸರಣಿಯನ್ನು ರಚಿಸಬಹುದು ಮತ್ತು ಅದನ್ನು ಈ ಮೊದಲೇ ಇರುವ ಡಾಟಾ ಸಿರೀಸ್‌ನೊಂದಿಗೆ ಹೈಲೈಟ್‌ಮಾಡಬಹುದು.

ಒಂದು ಚಾರ್ಟ್‌ಗೆ ಪ್ರತ್ಯೇಕ ಶೀಟ್‌ನಲ್ಲಿ ಡಾಟಾ ಸಿರೀಸ್ ಸೇರಿಸುವುದು (Add a data series to a chart on a separate chart sheet)

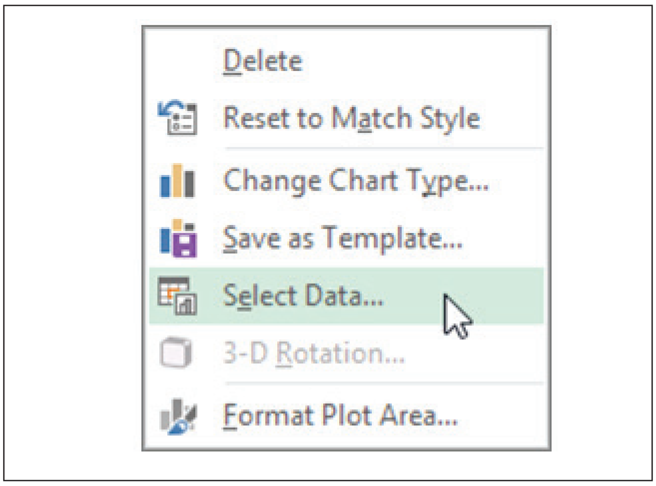
ನಿಮ್ಮ ಚಾರ್ಟ್ ಪ್ರತ್ಯೇಕ ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿದ್ದರೆ, ಹೊಸ ಡಾಟಾ ಸಿರೀಸ್‌ನ್ನು ಸೇರಿಸಲು ಡ್ರ್ಯಾಗ್ ಮಾಡುವುದು ಉತ್ತಮ ಮಾರ್ಗವಾಗುವುದಿಲ್ಲ. ಈ ಸಂದರ್ಭದಲ್ಲಿ Select Data Source ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಚಾರ್ಟ್‌ಗಾಗಿ ಹೊಸ ಡಾಟಾವನ್ನು ನಮೂದಿಸಬೇಕಾಗುತ್ತದೆ.

1 ನಿಮ್ಮ ಚಾರ್ಟ್ ಡಾಟಾವನ್ನು ಹೊಂದಿರುವ ವರ್ಕ್‌ಶೀಟ್‌ನ ಸೆಲ್‌ಗಳಲ್ಲಿ ಈಗಾಗಲೇ ಇರುವ ಸೋರ್ಸ್ ಡಾಟಾ ಮುಂದೆ ಅಥವಾ ಕೆಳಗೆ ನೇರವಾಗಿ ನೀವು ಸೇರಿಸಲು ಬಯಸಿದ ಹೊಸ ಡಾಟಾ ಸಿರೀಸ್‌ನ್ನು ನಮೂದಿಸಿ.

2 ನಿಮ್ಮ ಚಾರ್ಟ್‌ನ್ನು ಹೊಂದಿರುವ ವರ್ಕ್‌ಶೀಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 ಚಾರ್ಟ್ ಮೇಲೆ ಮೌಸ್‌ನ ಬಲ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ, ನಂತರ Select Data ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED

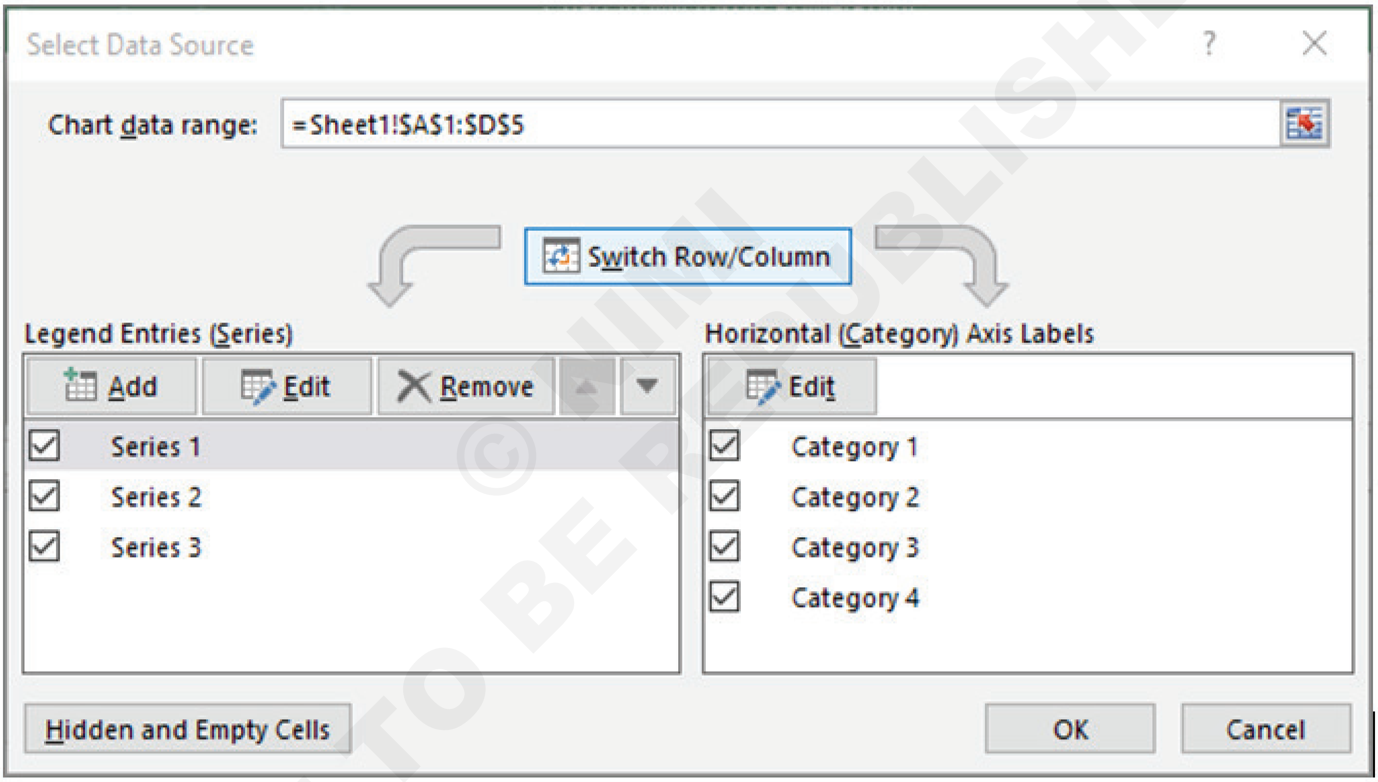


ವರ್ಕ್‌ಶೀಟ್ ಮೇಲೆ Select Data Source ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಕಾಣುತ್ತದೆ. ಇದು ಚಾರ್ಟ್‌ಗಾಗಿ ಸೋರ್ಸ್ ಡಾಟಾವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

4 ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನ್ನು ಹಾಗೆಯೆ ತೆರೆದಿಟ್ಟು, ವರ್ಕ್‌ಶೀಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ ನೀವು ಚಾರ್ಟ್‌ಗಾಗಿ ಬಳಸಲು ಬಯಸಿದ ಎಲ್ಲ ಡಾಟಾವನ್ನು, ಹೊಸ ಡಾಟಾ ಸಿರೀಸ್‌ನ್ನು ಒಳಗೊಂಡಂತೆ ಎಲ್ಲ ಡಾಟಾ ಆಯ್ಕೆ ಮಾಡಲು ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ಡ್ರ್ಯಾಗ್ ಮಾಡಿರಿ.

Select Data Source ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Legend Entries (Series) ನ ಅಡಿಯಲ್ಲಿ, ಹೊಸ ಡಾಟಾ ಸಿರೀಸ್ ಕಾಣುತ್ತದೆ.

5 ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನ್ನು ಮುಚ್ಚಲು ಮತ್ತು ಚಾರ್ಟ್ ಶೀಟ್‌ಗೆ ಹಿಂತಿರುಗಿ ಬರಲು OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.



© NIMI NOT TO BE REPUBLISHED

ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು (Create Pivot Tables)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಪೈವೋಟ್ ಟೇಬಲ್
- ಪೈವೊಟ್ ಟೇಬಲ್ ಕಾರ್ಯ
- ಡಾಟಾ ಗ್ರೂಪ್ ಮಾಡುವುದು.

ಪೈವೋಟ್ ಟೇಬಲ್ (Pivot table)

ದೊಡ್ಡ ಪ್ರಮಾಣದ ಡಾಟಾವನ್ನು ತ್ವರಿತವಾಗಿ ಸಂಕ್ಷಿಪ್ತಗೊಳಿಸಲು ಪೈವೋಟ್ ಟೇಬಲ್ ಒಂದು ಪರಿಣಾಮಕಾರಿಯಾದ ಮಾರ್ಗವಾಗಿದೆ. ಸಂಖ್ಯೆಗಳನ್ನು ಹೊಂದಿದ ಡಾಟಾವನ್ನು ವಿವರವಾಗಿ ವಿಶ್ಲೇಷಣೆ ಮಾಡಲು ನೀವು ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಮತ್ತು ನಿಮ್ಮ ಡಾಟಾದ ಬಗ್ಗೆ ಅನಿರೀಕ್ಷಿತ ಪ್ರಶ್ನೆಗಳಿಗೆ ಉತ್ತರವನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸಬಹುದು. ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ದೊಡ್ಡ ಪ್ರಮಾಣದ ಡಾಟಾ ಬಗ್ಗೆ ಅನೇಕ ಬಳಕೆದಾರ ಸ್ನೇಹಿ ಮಾರ್ಗದಲ್ಲಿ ಪ್ರಶ್ನಿಸಲು ವಿಶೇಷವಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

ಪೈವೊಟ್ ಟೇಬಲ್ ಕಾರ್ಯ (Pivot table work)

ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಸಂಕ್ಷಿಪ್ತಗೊಳಿಸಲು, ಸಾರ್ಟ್ ಮಾಡಲು, ಮರುಹೊಂದಿಸಲು, ಗ್ರೂಪ್ ಮಾಡಲು, ಎಣಿಸಲು, ಮೊತ್ತ ಕಂಡು ಹಿಡಿಯಲು, ಅಥವಾ ಸರಾಸರಿ ಕಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ನಮಗೆ ಕಾಲಂಗಳನ್ನು ರೋ ಗಳನ್ನಾಗಿ ಬದಲಾಯಿಸಲು ಮತ್ತು ರೋ ಗಳನ್ನು ಕಾಲಂಗಳನ್ನಾಗಿ ಬದಲಾಯಿಸಲು ಅವಕಾಶ ಮಾಡಿಕೊಡುತ್ತದೆ. ಇದು ಯಾವುದೇ ಫೀಲ್ಡ್ ಮೇಲೆ (ಕಾಲಂ) ಗ್ರೂಪಿಂಗ್ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ ಮತ್ತು ಅವುಗಳ ಮೇಲೆ ಲೆಕ್ಕಗಳನ್ನು ಮಾಡಲು ಅವಕಾಶ ಮಾಡಿಕೊಡುತ್ತದೆ.

ಡಾಟಾ ಗ್ರೂಪ್ ಮಾಡುವುದು (Group data)

ಪೈವೋಟ್ ಟೇಬಲ್‌ನಲ್ಲಿ ಗ್ರೂಪಿಂಗ್ ಮಾಡುವುದರಿಂದ ಡಾಟಾದ ಸಬ್‌ಸೆಟ್‌ನ್ನು ವಿಶ್ಲೇಷಣೆ ಮಾಡಲು ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಒಂದು ಪೈವೋಟ್ ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಸರಿಯಾದ ರೀತಿಯಲ್ಲಿಲ್ಲದ date ಮತ್ತು time ಫೀಲ್ಡ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು quarters ಮತ್ತು months ಗಳಾಗಿ ಗ್ರೂಪ್ ಮಾಡಲು ಬಯಸಿದಾಗ ನೀವು ಇದನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಎಂ ಎಸ್ ಪವರ್ ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್‌ನಲ್ಲಿ ಫೈಲ್‌ಗಳನ್ನು ತೆರೆಯುವುದು (Open files in MS PowerPoint Presentations)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಎಂ ಎಸ್ ಪವರ್‌ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್‌ನ ಪರಿಚಯ
- ಚಿತ್ರದ ಇಫೆಕ್ಟ್‌ಗಳನ್ನು ಬಳಸುವುದು ಮತ್ತು ಎಡಿಟ್ ಮಾಡುವುದು.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ಪವರ ಪಾಯಿಂಟ್, ಮೈಕ್ರೊಸಾಫ್ಟ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಒಂದು ಸರಳವಾದ ಮತ್ತು ಶಕ್ತಿಶಾಲಿಯಾದ ಸಾಫ್ಟ್‌ವೇರ್ ಆಗಿದೆ. ಇದು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾದ ಪ್ರೆಸೆಂಟೇಷನ್‌ಗಳನ್ನು ತಯಾರಿಸುತ್ತದೆ. ಮುಖ್ಯವಾಗಿ ಇದು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಆಫೀಸ್ ಸೂಟ್‌ನ ಒಂದು ಭಾಗವಾಗಿದೆ. ಈ ಪ್ರೊಗ್ರಾಂ ಸ್ಲೈಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ ಮತ್ತು ಈ ತರಹದ ಅನೇಕ ಟೂಲ್‌ಗಳಾದ ವರ್ಡ್ ಪ್ರೊಸೆಸಿಂಗ್, ಡ್ರಾಯಿಂಗ್, ಗ್ರಾಫಿಂಗ್, ಮತ್ತು ಔಟಲೈನಿಂಗ್ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಇದನ್ನು ಟೆಕ್ಸ್ಟ್, ಟೇಬಲ್‌ಗಳು, ಚಾರ್ಟ್‌ಗಳು, ಗ್ರಾಫಿಕ್ಸ್ ಮತ್ತು ಮೀಡಿಯಾಗಳನ್ನು ಸುಲಭವಾಗಿ ಸ್ಲೈಡ್‌ಗಳಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮೈಕ್ರೊಸಾಫ್ಟ್ ಪವರ್‌ಪಾಯಿಂಟ್ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳಬೇಕಾದರೆ ನೀವು ಅದರ ಮೂರು ಮುಖ್ಯ ವೈಶಿಷ್ಟ್ಯಗಳ ಬಗ್ಗೆ ಗಮನಹರಿಸುವ ಅವಶ್ಯಕತೆ ಇದೆ. ಈ ವೈಶಿಷ್ಟ್ಯಗಳು ಯಾವವೆಂದರೆ - ಮೈಕ್ರೊಸಾಫ್ಟ್ ಆಫೀಸ್ ಬಟನ್, ಕ್ವಿಕ್ ಎಕ್ಸೆಸ್ ಟೂಲ್‌ಬಾರ್, ಮತ್ತು ರಿಬ್ಬನ್.

ಪವರ್‌ಪಾಯಿಂಟ್ ಸ್ಲೈಡ್ ಶೋ (PPT), ಒಂದು ಪ್ರೆಸೆಂಟೇಷನ್ ಆಗಿದ್ದು , ಇದನ್ನು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಕಂಪನಿಯ ಸಾಫ್ಟ್‌ವೇರ್‌ನಲ್ಲಿ ತಯಾರಿಸಲಾಗಿದೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ಆಡಿಯೋ, ವಿಡಿಯೋ ಮತ್ತು ಆಡಿಯೊ/ವಿಜುವಲ್ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಪ್ರೆಸೆಂಟೇಷನ್‌ನಲ್ಲಿ ಸೇರಿಸಲು ಅವಕಾಶ ನೀಡುತ್ತದೆ. ಇದನ್ನು ಮಲ್ಟಿಮೀಡಿಯಾ ತಂತ್ರಜ್ಞಾನವನ್ನಾಗಿ ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು ಕೊಲ್ಯಾಬೊರೇಷನ್ ಮತ್ತು ವಿಷಯವನ್ನು ಹಂಚಿಕೊಳ್ಳುವ ಒಂದು ಸಾಧನವನ್ನಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮೈಕ್ರೊಸಾಫ್ಟ್ ಆಫೀಸ್ ಪವರ್‌ಪಾಯಿಂಟ್‌ನ್ನು ಒಳಗೊಂಡಿದ್ದು, ಇದನ್ನು ಪ್ರೆಸೆಂಟೇಷನ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಅತ್ಯಂತ ಪ್ರಮುಖ ಮತ್ತು ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಬ್ರ್ಯಾಂಡ್ ಆಗಿ ಮಾಡಿದೆ.

ಪವರ್‌ಪಾಯಿಂಟ್ ಸ್ಲೈಡ್ ಶೋವನ್ನು ಪವರ್‌ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್ ಎಂದೂ ಕೂಡ ಕರೆಯುತ್ತಾರೆ.

ಚಿತ್ರಗಳನ್ನು ಎಡಿಟ್ ಮಾಡುವುದು (Edit pictures)

1 ಚಿತ್ರದ ಹೊಳಪು(brightness), ಭಿನ್ನತೆ (contrast), ತೀಕ್ಷ್ಣತೆ (sharpness) ಮುಂತಾದವುಗಳನ್ನು ಸರಿಪಡಿಸುವುದು. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. Picture Format ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು Corrections ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.....

2 Apply Artistic effects ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ......

3 Change the color. ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ......

4 Apply picture effects. ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ......

5 Add a border ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ......

6 Compress the picture ಆಯ್ಕೆ ಮಾಡಿರಿ. ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ......

- Adjust the brightness, contrast, or sharpness
- Select the picture.
- Picture Format ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Corrections. ಆಯ್ಕೆ ಮಾಡಿರಿ.
- ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರಿವಿವ್ (ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡುವುದು) ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ, Change brightness,contrast or sharpness of a picture ನ್ನು ನೊಡಿರಿ.

ಆರ್ಟಿಸ್ಟಿಕ್ ಇಫೆಕ್ಟ್‌ನ್ನು ಅನ್ವಯಿಸಿರಿ (Apply artistic effects)

1 ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 Picture Format ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Artistic Effects ಆಯ್ಕೆ ಮಾಡಿರಿ.

3 ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರಿವಿವ್ (ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡುವುದು) ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಸೂಚನೆ : ಒಂದು ಚಿತ್ರಕ್ಕೆ ಒಂದು ಸಮಯಕ್ಕೆ ಒಂದೇ Artistic Effect ನ್ನು ನೀಡಬಹುದು. ಅದ್ದರಿಂದ ಇನ್ನೊಂದು ಭಿನ್ನವಾದ Artistic Effect ನ್ನು ಅನ್ವಯಿಸುವುದರಿಂದ ಈ ಹಿಂದೆ ನೀವು ಅನ್ವಯಿಸಿದ Artistic Effect ತೆಗೆದು ಹಾಕಲ್ಪಡುತ್ತದೆ.

4 ಬಣ್ಣವನ್ನು ಬದಲಾಯಿಸಿ

5 ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

6 Picture Format ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Color ಆಯ್ಕೆ ಮಾಡಿರಿ.

7 ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರಿವಿವ್ (ಪೂರ್ವಭಾವಿಯಾಗಿ ನೋಡುವುದು) ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

ಚಿತ್ರದ ಇಫೆಕ್ಟ್‌ಗಳನ್ನು ಅನ್ವಯಿಸುವುದು (Apply picture effects)

1 ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 Picture Format ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Picture Effects ಆಯ್ಕೆ ಮಾಡಿರಿ.

3 ನಿಮಗೆ ಬೇಕಾದ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ : Shadow, Reflection, Glow, Soft Edges, Bevel, ಅಥವಾ 3-D Rotation.

ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ, Add or change an effect for a picture ನ್ನು ನೋಡಿರಿ.

4 ಬಾರ್ಡರನ್ನು ಸೇರಿಸಿರಿ.

5 ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

6 Picture Format > Picture Border ಆಯ್ಕೆ ಮಾಡಿರಿ, ನಂತರ ಒಂದು ಬಾರ್ಡರನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ

7 Background ನ್ನು ತೆಗೆದುಹಾಕಿರಿ.

ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ, Remove a picture background ನೋಡಿರಿ.

8 ಚಿತ್ರವನ್ನು Compress ಮಾಡಿರಿ.

9 ಚಿತ್ರವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

10 Picture Format ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Compress Pictures. ಆಯ್ಕೆ ಮಾಡಿರಿ.

11 ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ, ನಂತರ OK ಆಯ್ಕೆ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಟೆಕ್ಸ್ಟ್ ಮತ್ತು ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Insert, Format text and paragraphs)

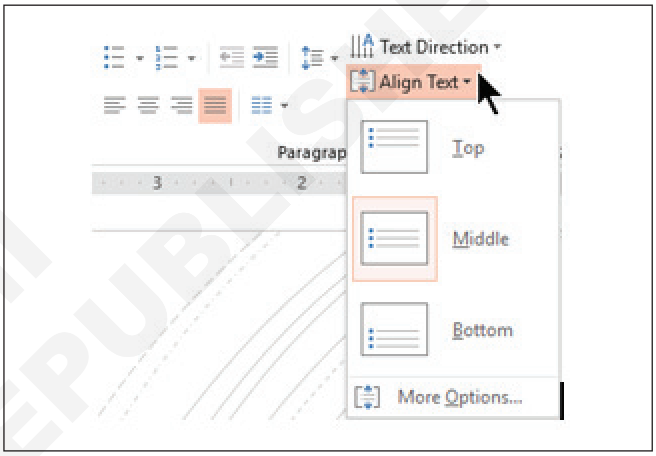
ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- ಟೆಕ್ಸ್ಟ್ ಅಲೈನ್‌ಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು.

ಟೆಕ್ಸ್ಟ್ ಅಲೈನ್‌ಮೆಂಟ್‌ನ್ನು ಹೇಗೆ ಬದಲಾಯಿಸುವುದು (How To Change Text Alignment)

- ಅಗತ್ಯವಿರುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ
- ಈಗ, Home ಟ್ಯಾಬ್‌ನಲ್ಲಿರುವ Paragraph ಗ್ರೂಪನ್ನು ಗುರುತಿಸಿರಿ.
- Paragraph ಗ್ರೂಪ್‌ನ ಕೆಳಭಾಗದಲ್ಲಿ ನಾಲ್ಕು ಅಲೈ ನ್‌ಮೆಂಟ್ ಆಯ್ಕೆಗಳು ಲಭ್ಯವಿರುತ್ತವೆ.
- ಮೌಸ್‌ನ ಎಡ ಕ್ಲಿಕ್ ಮಾಡುವುದರ ಮೂಲಕ ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

ಲಭ್ಯವಿರುವ ನಾಲ್ಕು ಅಲೈನ್‌ಮೆಂಟ್ ಆಯ್ಕೆಗಳು ಯಾವವೆಂದರೆ :

- Align Text Left : ಇದು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಎಡ ಮಾರ್ಜಿನ್‌ಗೆ ಅಲೈನ್ ಮಾಡುತ್ತದೆ
- Center : ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಮಧ್ಯದಲ್ಲಿ ತರುತ್ತದೆ.
- Align Text Right : ಇದು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಬಲ ಮಾರ್ಜಿನ್‌ಗೆ ಅಲೈನ್ ಮಾಡುತ್ತದೆ
- Justify : ಇದು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಎಡ ಮತ್ತು ಬಲ ಎರಡೂ ಮಾರ್ಜಿನ್‌ಗಳಲ್ಲಿ ಅಲೈನ್ ಮಾಡುತ್ತದೆ.



© NIMI NOT TO BE REPUBLISHED

ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು, ಮಾರ್ಪಡಿಸುವುದು , ಬುಲೆಟೆಡ್ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಮಾರ್ಪಡಿಸುವುದು (Create tables, modify tables, modify bulleted text)

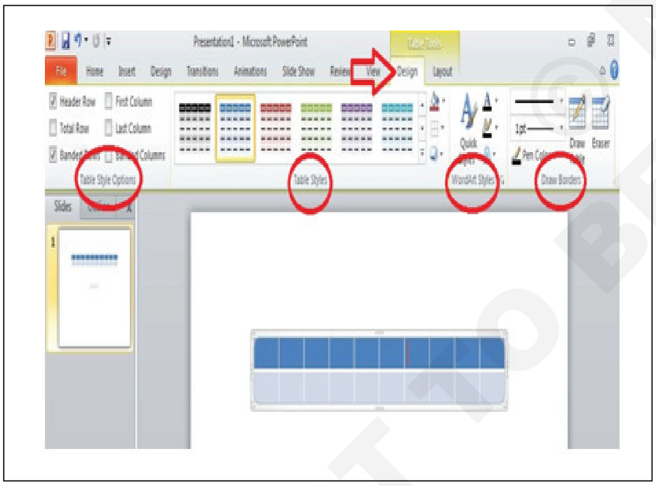
ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಟೇಬಲ್‌ನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುವುದು ಅಥವಾ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು
- ಬುಲೆಟೆಡ್ ಮತ್ತು ನಂಬರ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಉಪಯೋಗಿಸುವುದು.

ಟೇಬಲ್‌ನ್ನು ಹೇಗೆ ಮಾರ್ಪಾಡು ಮಾಡುವುದು ಅಥವಾ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (How To Modify Or Format Table)

ನೀವು ಮಾರ್ಪಾಡು ಮಾಡಲು ಬಯಸುವ ಟೇಬಲ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. Design ಮತ್ತು Layout ಎನ್ನುವ ಎರಡು ಹೊಸ ಟ್ಯಾಬ್‌ಗಳು ರಿಬ್ಬನ್‌ನಲ್ಲಿ ಕಾಣುತ್ತವೆ. ಈ ಟ್ಯಾಬ್‌ಗಳು ನಿಮ್ಮ ಟೇಬಲ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ವಿವಿಧ ಹೊಸ ಕಮಾಂಡ್‌ಗಳ ಗ್ರೂಪ್‌ಗಳನ್ನು ನೀಡುತ್ತವೆ.

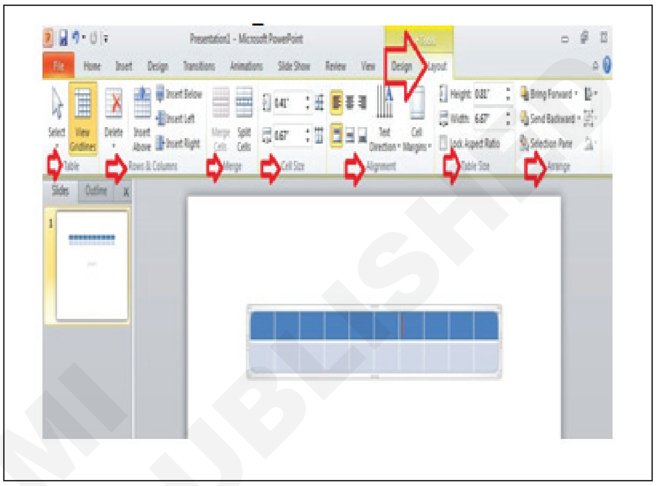
ನೀವು Design ಟ್ಯಾಬ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, ಅದು ನಿಮಗೆ ನಾಲ್ಕು ಹೊಸ ಕಮಾಂಡ್‌ಗಳ ಗುಂಪನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ ; Table Style Options, Table Styles, Word Art Styles and the Draw Borders.



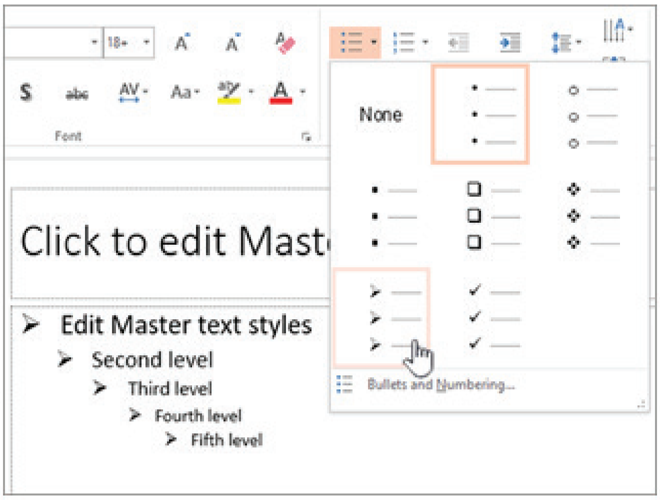
ನೀವು Layout ಟ್ಯಾಬ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, ಅದು ಆರು ಕಮಾಂಡ್‌ಗಳ ಗುಂಪನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ ; Table, Rows & Columns, Merge, Cell Size and Alignment.

ಬುಲೆಟ್‌ನ್ನು ಅಥವಾ ನಂಬರ್‌ನ್ನು ಹೊಂದಿದ ಲಿಸ್ಟ್‌ಗಳನ್ನು ಹೇಗೆ ತಯಾರಿಸುವುದು (How To Create Bulleted Or Numbered Lists)

- ಮೊದಲು, ನೀವು ಯಾವ ಟೆಕ್ಸ್ಟ್‌ಗೆ ಅಥವಾ ಪಟ್ಟಿಗೆ ಬುಲೆಟ್‌ಗಳನ್ನು ಅಥವಾ ಸಂಖ್ಯೆಗಳನ್ನು ಸೇರಿಸಲು ಬಯಸುತ್ತೀರೋ ಆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.



- ಈಗ, Home ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Paragraph ಗ್ರೂಪನ್ನು ಗುರುತಿಸಿರಿ.
- Bullets ಅಥವಾ Numbering ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಅಥವಾ ಪಕ್ಕದಲ್ಲಿರುವ ಡ್ರಾಪ್ ಡೌನ್ arrow ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಇದು ಇನ್ನೂ ಹೆಚ್ಚಿನ ಬುಲೆಟ್ ಸ್ಟೈಲ್‌ಗಳನ್ನು ಮತ್ತು ನಂಬರಿಂಗ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ.
- ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ಗೆ ನೀವು ಅನ್ವಯಿಸಲು ಬಯಸಿದ ಸ್ಟೈಲನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.



© NIMI NOT TO BE REPUBLISHED

ಇಲ್ಲಸ್ಟ್ರೇಷನ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು, ಇಲ್ಲಸ್ಟ್ರೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Insert illustrations, Format illustrations and text boxes)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಸ್ಲೈಡ್‌ಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು**
- **ಲೇಔಟ್‌ನ ಕಲರ್‌ನ್ನು ಅಥವಾ ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸುವುದು**
- **ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು**

ಸ್ಲೈಡ್‌ಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Add text to a slide)

ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಅಳವಡಿಸುವುದು ಮತ್ತು ಅದರಲ್ಲಿ ಟೈಪ್ ಮಾಡುವುದರ ಮೂಲಕ ನೀವು ಪವರ್‌ಪಾಯಿಂಟ್ ಸ್ಲೈಡ್‌ಗೆ ಅಥವಾಸ್ಲೈಡ್ ಮಾಸ್ಟರ್‌ಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನುಸೇರಿಸಬಹುದು. ನಂತರ ನೀವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಅಥವಾ ಇಡೀ ಬಾಕ್ಸ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿ ಅದರಲ್ಲಿರುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಬಹುದು. ಪ್ಲೇಸ್‌ಹೋಲ್ಡರ್‌ಗಳಿಗೆ ಮತ್ತು ಶೇಪ್‌ಗಳಿಗೂ ಕೂಡ ನೀವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಬಹುದು.

ಕೆಳಗೆ ನೀಡಿದ ಶೀರ್ಷಿಕೆಯನ್ನು ಆಯ್ಕೆ ಮಾಡಿ ತೆರೆಯಿರಿ ಮತ್ತು ವಿವರವಾದ ಸೂಚನೆಗಳನ್ನು ನೋಡಿರಿ.

ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Add text to a text box)

- ಯಾರಾದರೂ ಎಡಿಟ್ ಮಾಡಬಹುದಾದ ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ಗೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಲು, Normal view ನಲ್ಲಿ, ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ಒಳಗೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ, ಮತ್ತು ನಂತರ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಅಥವಾ ಪೇಸ್ಟ್ ಮಾಡಿರಿ.
- ಯಾರೂ ಎಡಿಟ್ ಮಾಡಲು ಬಾರದ ಮತ್ತು ಶಾಶ್ವತವಾಗಿ ಇರುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ಗೆ ಸೇರಿಸಲು, Slide Master View ನಲ್ಲಿ, ಮತ್ತು ನಂತರ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಅಥವಾ ಪೇಸ್ಟ್ ಮಾಡಿರಿ.

  ಒಂದು ಸ್ಲೈಡ್‌ನಲ್ಲಿ ಎಲ್ಲಿ ಬೇಕಾದರೂ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ನಮೂದಿಸಲು ನೀವು ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.ಉದಾಹರಣೆಗೆ ಒಂದು ಟೆಕ್ಸ್ಟ್ ಪ್ಲೇಸ್ ಹೋಲ್ಡರ್‌ನ ಹೊರಗೆ. ಉದಾಹರಣೆಗೆ , ಒಂದು ಚಿತ್ರಕ್ಕೆ ಕ್ಯಾಪ್ಷನ್ ನೀಡಬೇಕಾದರೆ, ಟೆಕ್ಸ್ಟ್ ಬಾಕ್ಸ್‌ನ್ನು ರಚಿಸಿರಿ ಮತ್ತು ಆ ಚಿತ್ರದ ಹತ್ತಿರ ಅದನ್ನು ಇರಿಸಿರಿ.

ಒಂದು ಶೇಪ್‌ನ (ಆಕಾರ) ಭಾಗವನ್ನಾಗಿ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Add text that is part of a shape)

ಶೇಪ್‌ಗಳು ಉದಾಹರಣೆಗೆ :squares, circles, callout balloons ಮತ್ತು block arrows ಮುಂತಾದವುಗಳು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಹೊಂದಿರಬಹುದು. ಯಾವಾಗ ನೀವು ಒಂದು ಶೇಪ್‌ನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡುತ್ತೀರೋ ಆಗ ಟೆಕ್ಸ್ಟ್ ಆ ಶೇಪ್‌ಗೆ ಅಂಟಿಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅದರ ಜೊತೆಗೆ ಚಲಿಸುತ್ತದೆ.

- ಒಂದು ಶೇಪ್‌ನ ಭಾಗವಾಗುವ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಲು, ಆ ಶೇಪ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು ನಂತರ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಅಥವಾ ಪೇಸ್ಟ್ ಮಾಡಿರಿ.

ಶೇಪ್‌ಗೆ ಅಧೀನವಲ್ಲದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Add text that is independent of a shape)

ಒಂದು ಶೇಪ್‌ಗೆ ನೀವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಲು ಬಯಸಿದರೆ ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್ ಸೂಕ್ತವಾಗಿರುತ್ತದೆ. ಆದರೆ ಟೆಕ್ಸ್ಟ್ ಆ ಶೇಪ್‌ಗೆ ಅಂಟಿಕೊಳ್ಳಬಾರದು ಎಂದು ನೀವು ಬಯಸಿದರೆ, ಅದಕ್ಕೆ ಒಂದು ಬಾರ್ಡರ್‌ನ್ನು ಸೇರಿಸಿ, ಅದನ್ನು fill ಮಾಡಿರಿ, shadow ಅಥವಾ three dimensional (3 - D) effect ಗಳನ್ನು ಆ ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನಲ್ಲಿರುವ ಟೆಕ್ಸ್ಟ್‌ಗೆ ನೀಡಿರಿ.

- ಒಂದು ಶೇಪ್‌ನಿಂದ ಸ್ವತಂತ್ರ್ಯವಾಗಿ ಚಲಿಸುವಂತಹ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಲು, ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಸೇರಿಸಿರಿ ಮತ್ತು ನಂತರ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಅಥವಾ ಪೇಸ್ಟ್ ಮಾಡಿರಿ.

ಟೆಕ್ಸ್ಟ್‌ಬಾಕ್ಸ್‌ನ್ನು ಅಳವಡಿಸುವುದು (Add a text box)

a Insert ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Text ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ Text Box ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

b ಸ್ಲೈಡ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ ಪಾಯಿಂಟರ್‌ನ್ನು ಡ್ರ್ಯಾಗ್ ಮಾಡಿ ಟೇಕ್ಸ್ಟ್ ಬಾಕ್ಸ್ ನ ಚಿತ್ರ ಬರೆಯಿರಿ.

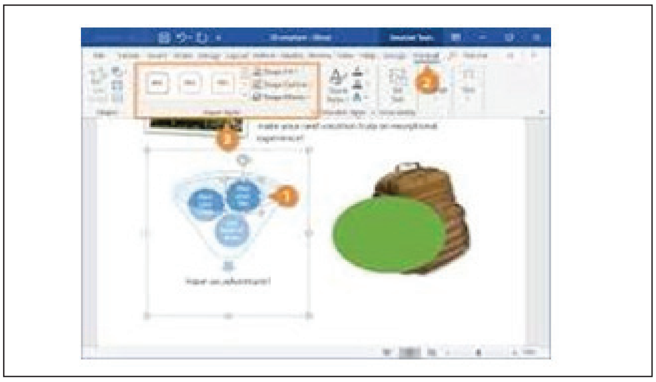
ಲೇಔಟ್, ಕಲರ್ ಅಥವಾ ಸ್ಟೈಲ್ ಬದಲಾಯಿಸುವುದು (Change the layout, color or style)

SmartArt ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. SmartArt Tools > Design tab ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಲೇಔಟ್‌ಗಳನ್ನು ಪೂರ್ವಭಾವಿಯಾಗಿ

© NIMI NOT TO BE REPUBLISHED

ನೋಡಲು ಅವುಗಳ ಮೇಲೆ ಪಾಯಿಂಟ್ ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. Change Colors ಆಯ್ಕೆ ಮಾಡಿರಿ ಆಯ್ಕೆಗಳನ್ನು ಪ್ರಿವಿವ್ ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ.

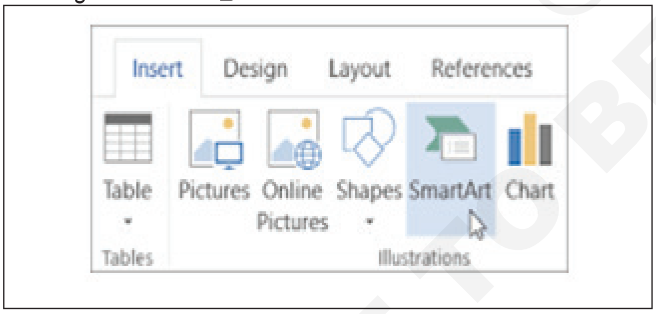
ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್‌ನ್ನು ನೀವು ಹೇಗೆ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುತ್ತೀರಿ? (How do you Format smart art?)



ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್‌ನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು (Format smart art)

ನೀವು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಬಯಸಿದ ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಶೇಪನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ. SmartArt Tools ನ ಅಡಿಯಲ್ಲಿರುವ Format ಟ್ಯಾಬನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಪ್ರತಿಯೊಂದು ಶೇಪನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು Shape Styles ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ ಆಯ್ಕೆಗಳನ್ನು ಉಪಯೋಗಿಸಿರಿ.

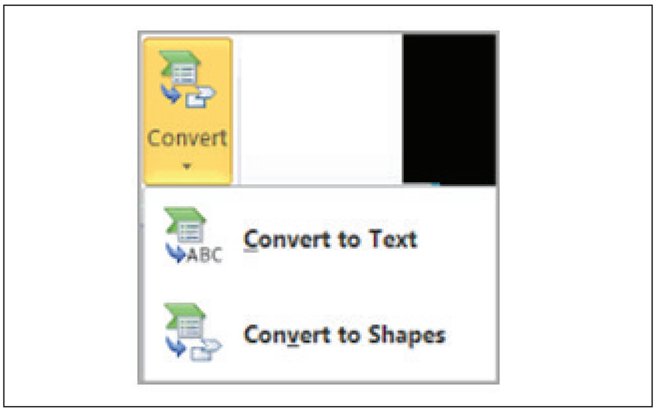
ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಗ್ರಾಫಿಕ್‌ನ್ನು ನೀವು ಹೇಗೆ ಅಳವಡಿಸುತ್ತಿರಿ ಮತ್ತು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುತ್ತೀರಿ?



SmartArt graphic ನ್ನು ಅಳವಡಿಸಿರಿ ಮತ್ತು ಅದಕ್ಕೆ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೇರಿಸಿರಿ.

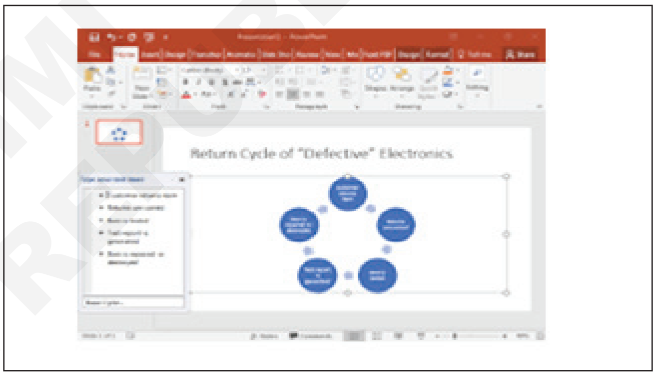
Insert ಮೆನುವಿನಲ್ಲಿ ಇರುವ SmartArt Graphic ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ SmartArt ಟ್ಯಾಬ್‌ನಲ್ಲಿ SmartArt Graphic ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ ನಿಮಗೆ ಬೇಕಾದ ರೀತಿಯ ಗ್ರಾಫಿಕ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ (List, Process ಇತ್ಯಾದಿಗಳು) ಮತ್ತು ನಂತರ layout ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

ಪವರ್‌ಪಾಯಿಂಟ್‌ನಲ್ಲಿ ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್‌ನ ಶೇಪನ್ನು ನಾನು ಹೇಗೆ ಬದಲಾಯಿಸಬಹುದು?



Design ಟ್ಯಾಬ್‌ನಲ್ಲಿ, SmartArt Tools ನ ಅಡಿಯಲ್ಲಿ, Reset ಗ್ರೂಪ್‌ನಲ್ಲಿ, Conver ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ ನಂತರ Convert to Shapes ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ. ಒಂದು ವೇಳೆ ನಿಮಗೆ SmartArt Tools ಅಥವಾ Design ಟ್ಯಾಬ್ ಕಂಡು ಬರದಿದ್ದರೆ, ನೀವು SmartArt graphics ನ್ನು ಆಯ್ಕೆ ಮಾಡಿದ್ದೀರಿ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ. Design ಟ್ಯಾಬನ್ನು ತೆರೆಯಲು ಶೇಪ್ ಮೇಲೆ ಡಬಲ್ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಪವರ್‌ಪಾಯಿಂಟ್‌ನಲ್ಲಿ smart shapes ಉಪಯೋಗವೇನು ಎಂಬುದನ್ನು ಉದಾಹರಣೆಯೊಂದಿಗೆ ವಿವರಿಸಿ.?



ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಇದು ಕ್ರಿಯಾತ್ಮಕ ರೀತಿಯ ಚಿತ್ರವಾಗಿದೆ. ಸಾಮಾನ್ಯವಾಗಿ ನೀವು ಇದನ್ನು ಪವರ್‌ಪಾಯಿಂಟ್ ಸ್ಲೈಡ್‌ಗಳಲ್ಲಿ ಆಗಾಗ್ಗೆ ನೊಡುತ್ತೀರಿ. ಸ್ಮಾರ್ಟ್ ಆರ್ಟನ್ನು ಉಪಯೋಗಿಸಿ ಬುಲೆಟ್ ಲಿಸ್ಟ್‌ಗಳ ಮೂಲಕ ಮಾಹಿತಿಯನ್ನು ಒಟ್ಟಿಗೆ ಸೇರಿಸಲು ಅಥವಾ ಗ್ರೂಪ್ ಮಾಡಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , cycle ಗಳ ಮೂಲಕ ಒಂದು ಕ್ರಿಯೆಯನ್ನು ತೋರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಗ್ರಾಫಿಕ್‌ನ ವಿಧಗಳು ಯಾವವು ? (Which are types of SmartArt graphic?)

ಲೇಔಟ್ ವಿಧಗಳು. choose a SmartArt Graphic ಗ್ಯಾಲರಿ ಹನ್ನೊಂದು ವಿವಿಧ ರೀತಿಯ ಎಲ್ಲ ಲಭ್ಯವಿರುವ ಲೇಔಟ್‌ಗಳನ್ನು ತೋರಿಸುತ್ತದೆ. - All, List, Process, Cycle, Hierarchy, Relationship, Matrix, Pyramid, Picture, Office.com, ಮತ್ತು ಇತರೆ ವಿಧಗಳು.

© NIMI NOT TO BE REPUBLISHED

ಪವರ್‌ಪಾಯಿಂಟ್‌ನಲ್ಲಿ ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್ ಯಾಕೆ ಮಹತ್ವದ್ದಾಗಿದೆ?

Organization, hierarchy, stages ಮತ್ತು processes ಗಳನ್ನು ರಚಿಸಲು ಸ್ಮಾರ್ಟ್‌ಆರ್ಟ್ ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು ನೋಡಲು ಆಹ್ಲಾದಕರವಾದ ಬಣ್ಣಗಳನ್ನು ಮತ್ತು ಚಿತ್ರಗಳನ್ನು ಪರಿಚಯಿಸುತ್ತದೆ. ನೀವು ಏನು ಹೇಳಲು ಬಯಸುತ್ತೀರೋ ಅದಕ್ಕೆ ಮೌಲ್ಯವನ್ನು ಒದಗಿಸುತ್ತದೆ. ನಿಮ್ಮ ಅಗತ್ಯತೆಯ ಮತ್ತು ಬೇಕಾದ ಮಾಹಿತಿಯ ಆಧಾರದ ಮೇಲೆ ನೀವು ಹಲವಾರು ರೀತಿಯ ಸ್ಮಾರ್ಟ್ ಆರ್ಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಆಡಿಯೋ ಮತ್ತು ವಿಡಿಯೋ ಅಂಶಗಳು (Audio & Video Elements)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಪಿಪಿಪಿ ಯಲ್ಲಿ ಆಡಿಯೊವನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ತೆಗೆದುಹಾಕುವುದು**
- **ಆಡಿಯೊ ರೆಕಾರ್ಡ ಮಾಡುವುದು**
- **ಹಿನ್ನೆಲೆಯ ಆಯ್ಕೆಗಳನ್ನು ಬದಲಾಯಿಸುವುದು.**

ನಿಮ್ಮ ಪವರ್‌ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್‌ನಲ್ಲಿ ಆಡಿಯೊವನ್ನು ಸೇರಿಸುವುದು ಅಥವಾ ತೆಗೆದುಹಾಕುವುದು (Add or delete audio in your PowerPoint presentation)

ನಿಮ್ಮ ಪವರ್‌ಪಾಯಿಂಟ್ ಪ್ರೆಸೆಂಟೇಷನ್‌ಗೆ ನೀವು ಸಂಗೀತ, ನಿರೂಪಣೆ ಅಥವಾ ಧ್ವನಿಯ ತುಣುಕುಗಳಂತಹ ಆಡಿಯೊವನ್ನು ಸೇರಿಸಬಹುದು. ಯಾವುದೇ ಆಡಿಯೊವನ್ನು ರೆಕಾರ್ಡ್ ಮಾಡಲು ಅಥವಾ ಕೇಳಲು ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್ ಸೌಂಡ್‌ಕಾರ್ಡ್ ಮೈಕ್ರೊಫೋನ್ ಮತ್ತು ಸ್ಪೀಕರ್‌ಗಳನ್ನು ಹೊಂದಿರಬೇಕು.

ನಿಮ್ಮ ಪಿಸಿಯಿಂದ ಆಡಿಯೊವನ್ನು ಸೇರಿಸುವುದು (Add audio from your PC)

1 Insert > Audio ಆಯ್ಕೆ ಮಾಡಿರಿ

2 Audio on My PC ಆಯ್ಕೆ ಮಾಡಿರಿ.

3 Insert Audio ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ , ನೀವು ಸೇರಿಸಲು ಬಯಸಿದ ಆಡಿಯೊವನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

4 Insert ಆಯ್ಕೆ ಮಾಡಿರಿ

ಆಡಿಯೊ ರೆಕಾರ್ಡ್ ಮಾಡುವುದು (Record audio)

1 Insert > Audio ಆಯ್ಕೆ ಮಾಡಿರಿ

2 Record Audio ಆಯ್ಕೆ ಮಾಡಿರಿ

3 ನಿಮ್ಮ ಆಡಿಯೊ ಫೈಲ್‌ಗೆ ಹೆಸರನ್ನು ಟೈಪ್ ಮಾಡಿರಿ, Record ಆಯ್ಕೆ ಮಾಡಿರಿ, ನಂತರ ಮಾತನಾಡಿರಿ.

> **ಸೂಚನೆ : ಆಡಿಯೊ ರೆಕಾರ್ಡ್ ಮಾಡುವ ಸಲುವಾಗಿ ನಿಮ್ಮ ಸಾಧನವು ಮೈಕ್ರೊಫೋನ್ ಎನೆಬಲ್ಡ್ ಆಗಿರಬೇಕು.**

4 ನಿಮ್ಮ ರೆಕಾರ್ಡಿಂಗ್‌ನ್ನು ರಿವಿವ್ ಮಾಡಲು, Stop ಆಯ್ಕೆ ಮಾಡಿರಿ ನಂತರ Play ಆಯ್ಕೆ ಮಾಡಿರಿ.

5 ನಿಮ್ಮ ಕ್ಲಿಪ್‌ನ್ನು ರೀರೆಕಾರ್ಡ್ ಮಾಡಲು Record ಆಯ್ಕೆ ಮಾಡಿರಿ ಅಥವಾ ನಿಮಗೆ ನಿಮ್ಮ ರೆಕಾರ್ಡಿಂಗ್ ತೃಪ್ತಿಯಾಗಿದ್ದರೆ OK ಆಯ್ಕೆ ಮಾಡಿರಿ.

6 ನಿಮ್ಮ ಕ್ಲಿಪ್ ಅನ್ನು ಸ್ಲೈಡ್‌ನಲ್ಲಿ ನಿಮಗೆ ಬೇಕಾದ ಸ್ಥಳದಲ್ಲಿ ಇರಿಸಲು ಆಡಿಯೊ ಐಕನ್ ಅನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ಡ್ರ್ಯಾಗ್ ಮಾಡಿರಿ.

ಪ್ರತಿ ಸ್ಲೈಡ್‌ನಲ್ಲಿ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಆಡಿಯೊ ಫೈಲುಗಳನ್ನು ನೀವು ಉಪಯೋಗಿಸುತ್ತಿದ್ದರೆ, ಆಡಿಯೊ ಐಕನ್‌ನ್ನು ಸುಲಭವಾಗಿ ಹುಡುಕಲು

ಒಂದೇ ಸ್ಥಳದಲ್ಲಿ ಇರಿಸುವಂತೆ ಸೂಚಿಸಲಾಗುತ್ತದೆ.

7 Play ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಹಿನ್ನೆಲೆಯ ಆಯ್ಕೆಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Change playback options)

ಆಡಿಯೊ ಐಕನ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ Audio Tools Playback ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ. ನಂತರ ಯಾವ ಆಯ್ಕೆಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸುತ್ತಿರಿ ಎಂಬುದನ್ನು ಆರಿಸಿಕೊಳ್ಳಿರಿ:

- ಆಡಿಯೊ ಟ್ರಿಮ್ ಮಾಡಲು , Trim ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ನಂತರ ನಿಮಗೆ ಬೇಕಾದಂತೆ ಆಡಿಯೊ ಫೈಲನ್ನು ಕೆಂಪು ಮತ್ತು ಹಸಿರು ಸ್ಲೈಡರ್‌ಗಳನ್ನು ಬಳಸಿ ಟ್ರಿಮ್ ಮಾಡಿರಿ.
- ಆಡಿಯೊವನ್ನು ಫೇಡ್ ಇನ್ ಮತ್ತು ಫೇಡ್ ಔಟ್ ಮಾಡಲು, Fade Duration ಬಾಕ್ಸ್‌ಗಳಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳನ್ನು ಬದಲಾಯಿಸಿ.
- ವಾಲ್ಯೂಮ್‌ನ್ನು ಸರಿಪಡಿಸಲು , Volume ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು ನಿಮಗೆ ಬೇಕಾದಂತೆ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.
- ಆಡಿಯೊ ಫೈಲ್ ಯಾವ ರೀತಿ ಪ್ರಾರಂಭವಾಗಬೇಕು ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಲು, ಡ್ರಾಪ್ ಡೌನ್ ಆ್ಯರೋ ಆಯ್ಕೆ ಮಾಡಿ ಮತ್ತು ಅದರಲ್ಲಿರುವ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ
- Click Sequence ನಲ್ಲಿ : ಒಂದು ಕ್ಲಿಕ್‌ನಿಂದ ಆಡಿಯೊ ಫೈಲ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೇ paly ಆಗುತ್ತವೆ.
- Automatically: ನೀವು ಸ್ಲೈಡ್‌ಗೆ ಹೋದಾಗ ಆನ್ ಆಡಿಯೊ ಫೈಲ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೇ ಪ್ಲೇ ಆಗುತ್ತದೆ.

- When Clicked On: ಐಕನ್ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ ಮಾತ್ರ ಆಡಿಯೊ ಪ್ಲೇ ಆಗುತ್ತದೆ.
- ನಿಮ್ಮ ಪ್ರೆಸೆಂಟೇಷನ್‌ನಲ್ಲಿ ಆಡಿಯೊ ಯಾವ ರೀತಿ ಪ್ಲೇ ಆಗುತ್ತದೆ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಲು :
- Play Across Slides: ಎಲ್ಲ ಸ್ಲೈಡ್‌ಗಳಲ್ಲಿ ಒಂದೇ ಆಡಿಯೊ ಪ್ಲೇ ಆಗುತ್ತದೆ.
- Loop until Stopped : ಇದು ಆಡಿಯೊ ಫೈಲನ್ನು ಲೂಪ್‌ನಂತೆ ಪ್ಲೇ ಮಾಡುತ್ತದೆ, Play / Pouse ಬಟನ್ ಮೇಲೆ ನೀವು ಕ್ಲಿಕ್ ಮಾಡಿ ಇದನ್ನು ನಿಲ್ಲಿಸಬೇಕಾಗುತ್ತದೆ.
- ಬ್ಯಾಕ್‌ಗ್ರೌಂಡ್‌ನಲ್ಲಿ ಎಲ್ಲ ಸ್ಲೈಡ್‌ಗಳಲ್ಲಿಯೂ ಸತತವಾಗಿ ಆಡಿಯೊ ಪ್ಲೇ ಆಗಬೇಕು ಎಂದರೆ, Play in Background ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಆಡಿಯೊವನ್ನು ತೆಗೆದುಹಾಕುವುದು (Delete audio)

ಆಡಿಯೊ ಕ್ಲಿಪ್‌ನ್ನು ತೆಗೆದುಹಾಕಲು, ಸ್ಲೈಡ್‌ನಲ್ಲಿರುವ ಆಡಿಯೊ ಐಕನ್ ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Delete ಒತ್ತಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಟ್ರಾನ್ಸಿಷನ್‌ಗಳು ಮತ್ತು ಎನಿಮೇಷನ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು

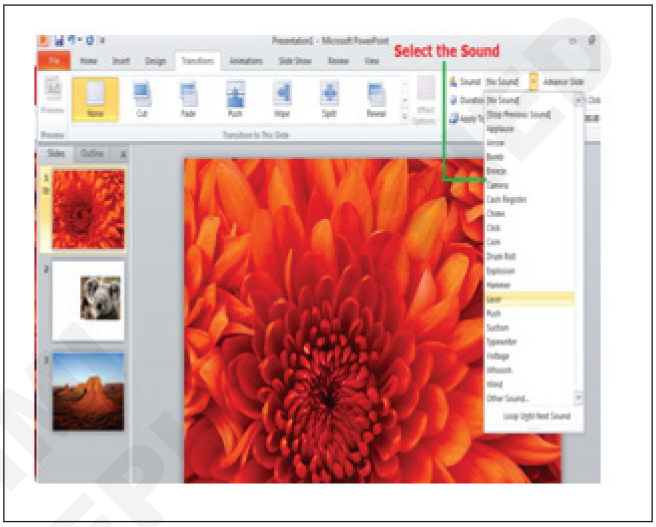
ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್‌ಗಳು ಮತ್ತು ಎನಿಮೇಷನ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು (Add slide transitions & animations)

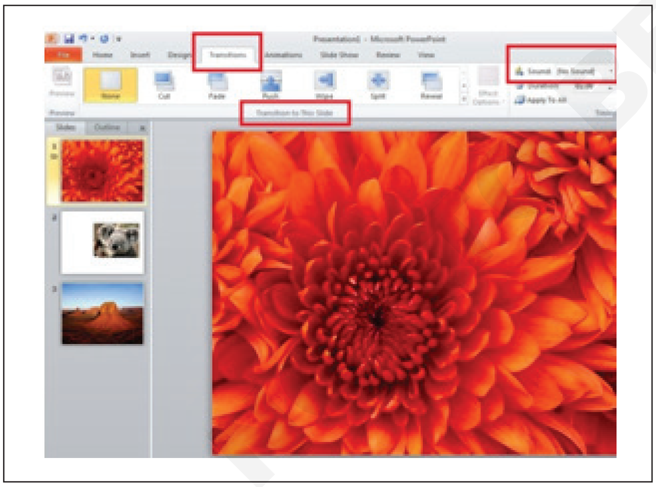
ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್ ಧ್ವನಿಯನ್ನು ಸೆಟ್ ಮಾಡುವುದು
- ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟ್‌ಗಳನ್ನು ಅನ್ವಯಿಸುವುದು
- ಕಸ್ಟಮ್ ಎನಿಮೇಷನ್ ಇಫೆಕ್ಟ್‌ನ್ನು ಅನ್ವಯಿಸುವುದು.

ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್ ಧ್ವನಿಯನ್ನು ಸೆಟ್ ಹೇಗೆ ಮಾಡುವುದು (How To Set Slide Transition Sound)

- ಮೊದಲು, ನೀವು ಟ್ರಾನ್ಸಿಷನ್ ಧ್ವನಿಯನ್ನು ಸೇರಿಸಲು ಬಯಸಿದ ಸ್ಲೈಡನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- Transition ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.
- Transition to This Slide ಗ್ರೂಪ್‌ನಲ್ಲಿ, Transition Sound ಆಯ್ಕೆಯ ಪಕ್ಕದಲ್ಲಿರುವ ಡ್ರಾಪ್ ಡೌನ್ ಆ್ಯರೋ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.
- ಹಲವಾರು ವಿಭಿನ್ನ ಟ್ರಾನ್ಸಿಷನ್ ಧ್ವನಿಗಳ ಪಟ್ಟಿ ಕಾಣುತ್ತದೆ.
- ನೀವು ಬಯಸಿದ ಟ್ರಾನ್ಸಿಷನ್ ಸೌಂಡ್ ಆಯ್ಕೆ ಮಾಡಿರಿ
- ಎಲ್ಲ ಸ್ಲೈಡ್‌ಗಳಿಗೆ ಈ ಇಫೆಕ್ಟನ್ನು ಅನ್ವೆಯಿಸಲು Apply To All ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

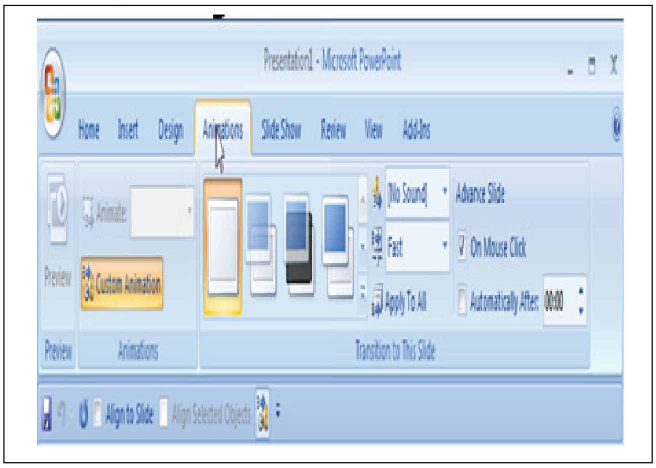




ಸ್ಲೈಡ್ ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟ್‌ನ್ನು ಹೇಗೆ ಅನ್ವಯಿಸುವುದು (How To Apply Slide Transition Effects)

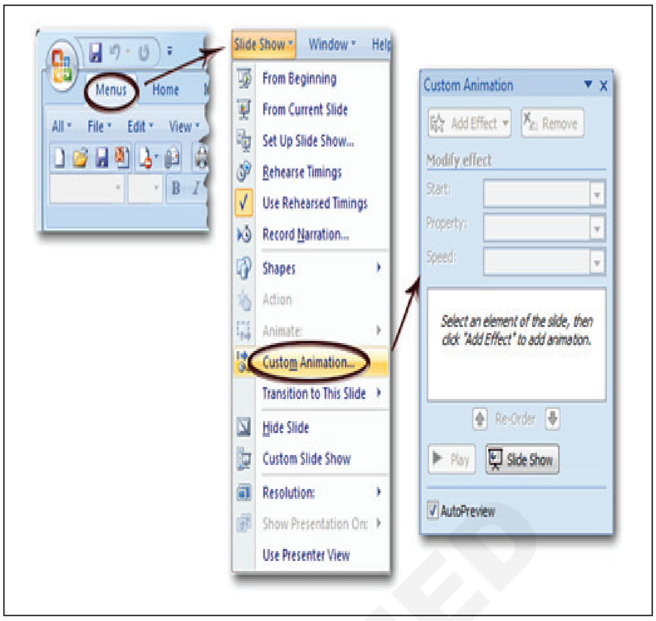
ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಲೈಡ್ ಶೋನಲ್ಲಿ ಒಂದು ಸ್ಲೈಡ್ ನಂತರದ ಸ್ಲೈಡ್‌ಗೆ ಬದಲಾದಾಗ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ.

- ಮೊದಲು, ನೀವು ಇಫೆಕ್ಟನ್ನು ಅನ್ವಯಿಸಲು ಬಯಸಿದ ಸ್ಲೈಡನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- Animation ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- ಈಗ, Transition to This Slide ಗ್ರೂಪ್‌ನಲ್ಲಿ, ನೀವು ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟ್‌ಗಳನ್ನು ನೋಡಬಹುದು.
- ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟ್‌ಗಳ ಮೆನು ನೋಡಲು ಡ್ರಾಪ್ ಡೌನ್ ಆ್ಯರೋ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.
- ನೀವು ಬಯಸಿದ ಟ್ರಾನ್ಸಿಷನ್ ಇಫೆಕ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- ಈ ಇಫೆಕ್ಟ್‌ನ್ನು ಎಲ್ಲ ಸ್ಲೈಡ್‌ಗಳಿಗೆ ಅನ್ವಯಿಸಲು Apply To All ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.



ಕಸ್ಟಮ್ ಎನಿಮೇಷನ್ ಇಫೆಕ್ಟ್‌ನ್ನು ಹೇಗೆ ಅನ್ವಯಿಸುವುದು (How To Apply A Custom Animation Effect)

- ಮೊದಲು, ಅಗತ್ಯವಿರುವ ಟೆಕ್ಸ್ಟ್ ಅಥವಾ ಆಬ್ಜೆಕ್ಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.
- ನಂತರ, Animation ಟ್ಯಾಬನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.
- Animation ಗ್ರೂಪ್‌ನಲ್ಲಿ, Custom Animation ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.
- ಈಗ ಬಲಭಾಗದಲ್ಲಿ Custom Animation ಟಾಸ್ಕ್‌ಪೇನ್ ಕಂಡು ಬರುತ್ತದೆ.
- Add Effect ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ, ಇದು ನಾಲ್ಕು ಆಯ್ಕೆಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಈ ಆಯ್ಕೆಗಳು ಮತ್ತೆ ವಿಭಿನ್ನ ಆಯ್ಕೆಗಳಲ್ಲಿ ವಿಭಾಗಿಸಲ್ಪಟ್ಟಿರುತ್ತವೆ.
- ನಿಮಗೆ ಬೇಕಾದ ಇಫೆಕ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.



© NIMI NOT TO BE REPUBLISHED

ಕಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು (Add and manage comments)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಕಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುವುದು
- ಕಮೆಂಟ್‌ನ್ನು ತೋರಿಸುವುದು ಮತ್ತು ಮರೆಮಾಡುವುದು
- ಕಮೆಂಟ್‌ನ್ನು ಎಡಿಟ್ ಮತ್ತು ಡಿಲೀಟ್ ಮಾಡುವುದು

ಪ್ರೆಸೆಂಟೇಷನ್‌ನಲ್ಲಿ ಕಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು, ಬದಲಾಯಿಸುವುದು, ಮರೆಮಾಡುವುದು, ಅಥವಾ ಡಿಲೀಟ್ ಮಾಡುವುದು (Add, change, hide, or delete comments in a presentation)

ನೀವು ತಯಾರಿಸಿದ ಪ್ರೆಸೆಂಟೇಷನ್‌ನ್ನು ಜನರು ವಿಮರ್ಶೆ ಮಾಡಬೇಕು ಮತ್ತು ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಬೇಕು ಎಂದು ಬಯಸಿದರೆ ಕಮೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿರಿ. ಅಥವಾ ಸಹೋದ್ಯೋಗಿಗಳು ಒಂದು ಪ್ರೆಸೆಂಟೇಷನ್ ಮೇಲೆ ನಿಮಗೆ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಲು ಕೇಳಿ ದಾಗ ಕಮೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಕಮೆಂಟ್ ಎಂದರೆ, ಒಂದು ಸ್ಲೈಡ್‌ಗೆ ಅಥವಾ ಸಂಪೂರ್ಣ ಸ್ಲೈಡ್‌ಗೆ ಒಂದು ಅಕ್ಷರ ಅಥವಾ ಶಬ್ದವನ್ನು ಲಗತ್ತಿಸುವ ಸೂಚನೆ(note) ಆಗಿದೆ.

ಕಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುವುದು (Add a comment)

1 ನೀವು ಕಮೆಂಟ್ ಮಾಡಲು ಬಯಸುವ ಆಬ್ಜೆಕ್ಟನ್ನು ಅಥವಾ ಸ್ಲೈಡನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. Review > New Comment ಆಯ್ಕೆ ಮಾಡಿರಿ. ಅಥವಾ Comments ಪೇನ್ ತೆರೆದಿದ್ದರೆ, New ಆಯ್ಕೆ ಮಾಡಿರಿ. Insert > Comment ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವ ಮೂಲಕ ಕೂಡ ನೀವು ಕಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು.

2 Comments ಪೇನ್‌ನಲ್ಲಿ, ಬಾಕ್ಸ್‌ನಲ್ಲಿ ನಿಮ್ಮ ಸಂದೇಶವನ್ನು ಟೈಪ್ ಮಾಡಿರಿ ಮತ್ತು Post ಆಯ್ಕೆ ಮಾಡಿರಿ ಅಥವಾ Ctrl + Enter ಒತ್ತಿರಿ.

ಕಮೆಂಟ್‌ಗಳನ್ನು ತೋರಿಸುವುದು ಅಥವಾ ಮರೆಮಾಡುವುದು (Show or hide comments)

1 View ಟ್ಯಾಬ್‌ನಲ್ಲಿ , Normal. ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 Review ಟ್ಯಾಬ್‌ನಲ್ಲಿ, Show Comments ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಕಮೆಂಟ್‌ಗಳನ್ನು ನೋಡುವುದು ಮತ್ತು ಉತ್ತರಿಸುವುದು (View and reply to comments)

1 Comments ಪೇನ್‌ನಲ್ಲಿರುವ Comments ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 ಒಂದು ಕಮೆಂಟ್‌ಗೆ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಲು Reply ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

ಕಮೆಂಟ್‌ಗಳನ್ನು ಎಡಿಟ್ ಮಾಡುವುದು (Edit comments)

1 ನ್ಯಾವಿಗೇಷನ್ ಪೇನ್‌ನ Normal ವಿವ್‌ನಲ್ಲಿ, ನೀವು ಕಮೆಂಟ್ ಮಾಡಲು ಬಯಸುವ ಸ್ಲೈಡ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 Comments ಪೇನ್‌ನಲ್ಲಿ, ನೀವು ಎಡಿಟ್ ಮಾಡಲು ಬಯಸುವ ಕಮೆಂಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ, ನಂತರ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡಲು ಪೆನ್ಸಿಲ್ ಐಕನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

> **ಸೂಚನೆ : ನಿಮ್ಮ ಕಮೆಂಟ್‌ಗಳನ್ನು ಬೇರೆಯವರು ಎಡಿಟ್ ಮಾಡುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ ಎಂಬುದನ್ನು ಗಮನದಲ್ಲಿಟ್ಟುಕೊಳ್ಳಿರಿ. ಆಫೀಸ್ ಡಾಕ್ಯುಮೆಂ ಟ್‌ನಲ್ಲಿರುವ ಕಮೆಂಟ್‌ಗಳು ಫೈಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುತ್ತವೆ. ಆದ್ದರಿಂದ ನಿಮ್ಮ ಫೈಲ್‌ಗೆ ಎಡಿಟ್ ಅನುಮತಿಯನ್ನು ಹೊಂದಿದ ಯಾರಾದರೂ ನಿಮ್ಮ ಕಮೆಂಟ್‌ಗಳನ್ನು ಬದಲಾಯಿಸಬಹುದು.**

ಕಮೆಂಟ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು (Delete a comment)

Comments ಪೇನ್‌ನಲ್ಲಿರುವ, ನೀವು ತೆಗೆದುಹಾಕಲು ಬಯಸಿದ ಕಮೆಂಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ, More thread actions (...) ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು Delete thread ಆಯ್ಕೆ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಮೈ ಎಸ್‌ಕ್ಯೂ ಎಲ್‌ನಲ್ಲಿ ಡಾಟಾಬೇಸನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು, ಸಮಸ್ಯೆ ನಿವಾರಣೆ, ರಚಿಸುವುದು ಮತ್ತು ಉಪಯೋಗಿಸುವುದು (Install, Troubleshoot, Create and Use of database in MySQL)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- DBMS ಮತ್ತು RDBMS ಪರಿಕಲ್ಪನೆ
- DBMS ಮತ್ತು RDBMS ನಲ್ಲಿ ವ್ಯತ್ಯಾಸ
- ಡಾಟಾ ಮಾಡೆಲ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- DBA ನ ಪರಿಕಲ್ಪನೆ
- ಡಾಟಾಬೇಸ್ ಬಳಕೆದಾರರನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- ಡಾಟಾಬೇಸ್ ಸ್ಕೀಮಾ ವ್ಯಾಖ್ಯಾನಿಸುವುದು.

DBMS ಮತ್ತು RDBMS ಪರಿಕಲ್ಪನೆ (Concepts of DBMS & RDBMS)

Relational DataBase Management System (RDBMS) ಎಂದರೆ ಪ್ರೊಗ್ರಾಂಗಳ ಸಂಗ್ರಹವಾಗಿದೆ ಮತ್ತು ಇದು IT ತಂಡಗಳಿಗೆ ಮತ್ತು ಇತರರಿಗೆ ರಿಲೇಷನಲ್ ಡಾಟಾಬೇಸನ್ನು ರಚಿಸುವುದು, ಅಪ್‌ಡೇಟ್ ಮಾಡುವುದು, ನಿರ್ವಹಿಸುವುದು ಹಾಗೂ ಪರಸ್ಪರ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಸಹಾಯ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವಾಗಿದೆ. RDBMS ಗಳು ಡಾಟಾವನ್ನು ಟೇಬಲ್‌ನ ರೂಪದಲ್ಲಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ. ಹೆಚ್ಚಿನ ವಾಣಿಜ್ಯದ Relational DataBase Management System ಗಳು ಡಾಟಾಬೇಸನ್ನು ಪಡೆಯಲು Structured Query Language (SQL) ನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಅದಾಗ್ಯೂ ರಿಲೇಷನಲ್ ಮಾಡೆಲ್‌ನ ಪ್ರಾರಂಭಿಕ ಅಭಿವೃದ್ಧಿಯ ನಂತರ SQL ನ ಆವಿಷ್ಕಾರ ಆದುದರಿಂದ ಇದು RDBMS ಬಳಕೆಗೆ ಆವಶ್ಯಕವಿರುವುದಿಲ್ಲ.

ಜಗತ್ತಿನಾದ್ಯಂತ ಇರುವ ಸಂಸ್ಥೆಗಳಲ್ಲಿ RDBMS ಅತ್ಯಂತ ಜನಪ್ರೀಯ ಡಾಟಾಬೇಸ್ ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಇದು ದೊಡ್ಡ ಪ್ರಮಾಣದ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹ ಮಾಡಲು ಮತ್ತು ಪುನಃ ಮರಳಿ ಪಡೆಯಲು ಭರವಸೆಯ ವಿಧಾನವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ವ್ಯವಸ್ಥೆಯ ಕಾರ್ಯ ನಿರ್ವಹಣೆ ಮತ್ತು ಸುಲಭ ಅನುಷ್ಠಾನವನ್ನು ಒದಗಿಸುತ್ತದೆ,

DBMS ಮತ್ತು RDBMS ನಲ್ಲಿ ವ್ಯತ್ಯಾಸ (RDBMS vs. DBMS)

ಸಾಮಾನ್ಯವಾಗಿ, ಡಾಟಾಬೇಸ್ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ. ಇದನ್ನು ಪ್ರಶ್ನಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಇತರ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಡಾಟಾಬೇಸ್ ಮ್ಯಾನೆಜಮೆಂಟ್ ಸಿಸ್ಟಮ್, ಡಾಟಾಬೇಸ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳ ಅಭಿವೃದ್ಧಿಯನ್ನು, ಆಡಳಿತವನ್ನು ಮತ್ತು ಉಪಯೋಗವನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ.

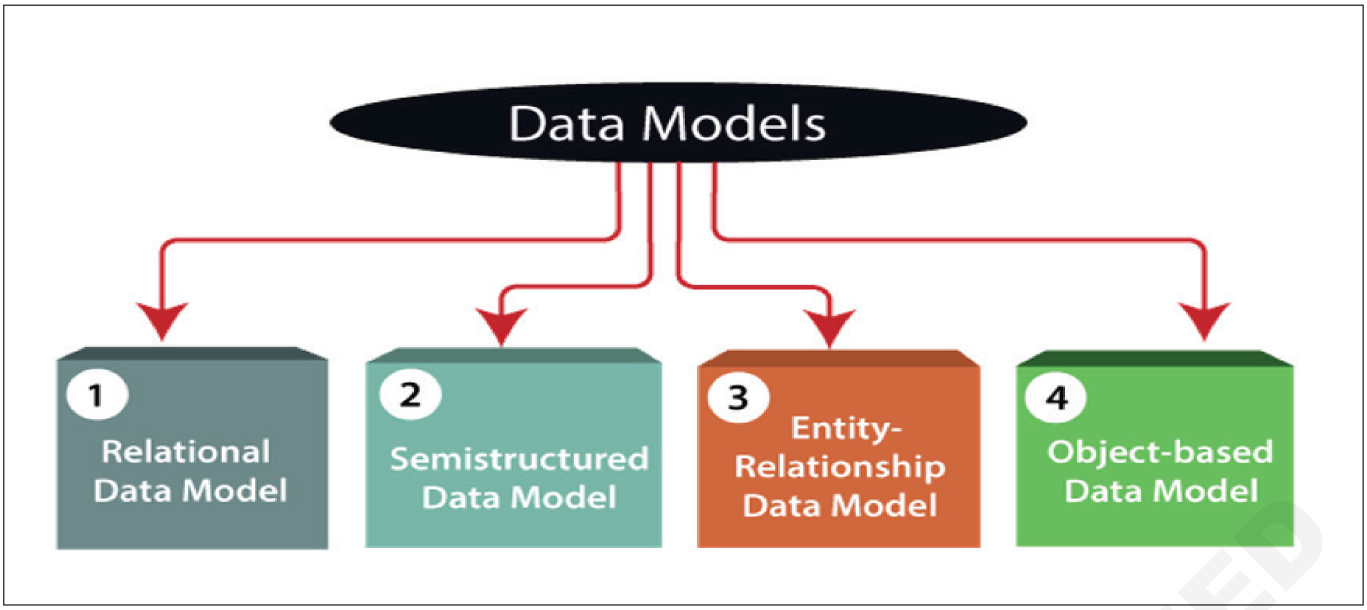
RDBMS , DataBase Management System (DBMS) ನ ಒಂದು ವಿಧವಾಗಿದೆ. ಇದು ಡಾಟಾವನ್ನು ರೋ (row) ಆಧಾರಿತ ಟೇಬಲ್ ಮಾದರಿಯಲ್ಲಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ. ಇದು ಸಂಬಂದಪಟ್ಟ ಡಾಟಾ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಜೋಡಿಸುತ್ತದೆ. RDBMS ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಈ ಫಂಕ್ಷನ್‌ಗಳು ಡಾಟಾದ ಸುರಕ್ಷತೆ, ನಿಖರತೆ, ಸಮಗ್ರತೆ, ಮತ್ತು ಸ್ಥಿರತೆಯನ್ನು ಕಾಪಾಡುತ್ತವೆ. ಇದು DBMS ನಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುವ ಫೈಲುಗಳ ಸಂಗ್ರಹಣೆಗಿಂತ ವಿಭಿನ್ನವಾಗಿದೆ.

ಡಾಟಾ ಮಾಡೆಲ್‌ಗಳು (Data Models)

ಡಾಟಾ ಮಾಡೆಲ್ ಎಂದರೆ, ಡಾಟಾದ ವಿವರಣೆ, ಡಾಟಾದ ಶಬ್ದಾರ್ಥಗಳು ಮತ್ತು ಡಾಟಾದ ಸ್ಥಿರತೆಯ ನಿರ್ಬಂಧಗಳನ್ನು ತಿಳಿಸುವ ಮಾದರಿಯಾಗಿದೆ. ಇದು ಡಾಟಾದ ಅಮೂರ್ತತೆಯ(data abstraction) ಪ್ರತಿಯೊಂದು ಹಂತದಲ್ಲಿಯೂ ಡಾಟಾಬೇಸ್‌ನ ವಿನ್ಯಾಸವನ್ನು ವಿವರಿಸುವ ಸಲುವಾಗಿ ಪರಿಕಲ್ಪನೆಯ ಸಾಧನಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಅದ್ದರಿಂದ, ಡಾಟಾಬೇಸ್‌ನ ರಚನೆಯನ್ನು ತಿಳಿದುಕೊಳ್ಳಲು ಈ ಕೆಳಗಿನ ನಾಲ್ಕು ಡಾಟಾ ಮಾಡೆಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

1 **ರಿಲೇಷನಲ್ ಡಾಟಾ ಮಾಡೆಲ್**(Relational Data Model) : ಈ ರೀತಿಯ ಮಾಡೆಲ್ ಡಾಟಾವನ್ನು ಟೇಬಲ್‌ನಲ್ಲಿ - ರೋ (row) ಮತ್ತು ಕಾಲಂ (column) ನ ರೂಪದಲ್ಲಿ ವಿನ್ಯಾಸಗೊಳಿಸುತ್ತದೆ. ಹೀಗೆ ರಿಲೇಷನಲ್ ಮಾಡೆಲ್ ಡಾಟಾವನ್ನು ಮತ್ತು ಅವುಗಳ ನಡುವಿನ ಸಂಬಂಧವನ್ನು ಪ್ರತಿನಿಧಿಸಲು ಟೇಬಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.ಟೇಬಲ್‌ಗಳನ್ನುರಿಲೇಷನ್‌ಗಳು(relations) ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಈ ಮಾಡೆಲ್‌ನ್ನು ಆರಂಭದಲ್ಲಿ 1969 ರಲ್ಲಿ Edgar F Codd ವಿವರಿಸಿದರು. ರಿಲೇಷನಲ್ ಡಾಟಾ ಮಾಡೆಲ್ ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಒಂದು ಮಾಡೆಲ್ ಆಗಿದೆ. ಪ್ರಥಮಿಕವಾಗಿ ಇದನ್ನು ವ್ಯಾಪಾರದ ಡಾಟಾ ಪ್ರೊಸೆಸಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಉಪಯೋಗಿಸುತ್ತಿದ್ದವು.

© NIMI NOT TO BE REPUBLISHED



2 **ಎಂಟಿಟಿ ರಿಲೇಷನಶಿಪ್ ಡಾಟಾ ಮಾಡೆಲ್** (Entity-Relationship Data Model): ER ಮಾಡೆಲ್ ಡಾಟಾವನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿ ಮತ್ತು ಅವುಗಳ ನಡುವಿನ ರಿಲೇಷನ್‌ಶಿಪ್‌ನ್ನು ಲಾಜಿಕಲ್ ಆಗಿ ಪ್ರಸ್ತುತಪಡಿಸುತ್ತದೆ. ಈ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಎಂಟಿಟಿ ಎಂದು ಕರೆಯುತ್ತಾರೆ ಮತ್ತು ರಿಲೇಷನ್‌ಶಿಪ್ ಎಂದರೆ ಈ ಎಂಟಿಟಿಗಳ ನಡುವಿನ ಸಂಬಂಧ. ಈ ಮಾಡೆಲ್‌ನ್ನು Peter Chen ವಿನ್ಯಾಸಗೊಳಿಸಿದರು ಮತ್ತು 1976 ಪತ್ರಿಕೆಗಳಲ್ಲಿ ಪ್ರಕಟಿಸಲಾಯಿತು. ಇದನ್ನು ಡಾಟಾಬೇಸ್ ವಿನ್ಯಾಸಗೊಳಿಸುವಲ್ಲಿ ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲಾಯಿತು. ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳ ಸೆಟ್ ಎಂಟಿಟಿಗಳನ್ನು ವಿವರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ , student_name, student_id ಇವುಗಳು student ಎಂಟಿಟಿಯನ್ನು ವಿವರಿಸುತ್ತವೆ. ಒಂದೇ ತರಹದ ಎಂಟಿಟಿಗಳ ಗುಂಪನ್ನು Entity set ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಒಂದೇ ತರಹದ ರಿಲೇಷನ್‌ಶಿಪ್‌ಗಳ ಗುಂಪನ್ನು relationship set ಎಂದು ಕರೆಯುತ್ತಾರೆ.

3 **ಆಬ್ಜೆಕ್ಟ್ ಆಧಾರಿತ ಡಾಟಾ ಮಾಡೆಲ್** (Object-based Data Model) : ಇದು ER ಮಾಡೆಲ್‌ನ ವಿಸ್ತರಿಸಿದ ಭಾಗವಾಗಿದೆ. ಇದು ಫಂಕ್ಷನ್, ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್ ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್ ಐಡೆಂಟಿಟಿ ಎಂಬ ಕಲ್ಪನೆಗಳೊಂದಿಗೆ ಬಂದ ಒಂದು ಮಾಡೆಲ್ ಆಗಿದೆ. ಈ ಮಾಡೆಲ್ ರಚನಾತ್ಮಕ ಮತ್ತು ಟೈಪ್‌ಗಳ ಸಂಗ್ರಹವನ್ನು ಒಳಗೊಂಡ ಶ್ರೀಮಂತ ಟೈಪ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಹೀಗೆ 1980 ರಲ್ಲಿ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ (object oriented) ವಿಧಾನವನ್ನು ಹೊಂದಿರುವ ಅನೇಕ ಡಾಟಾಬೇಸ್ ಸಿಸ್ಟಮ್‌ಗಳು ಅಭಿವೃದ್ಧಿ ಹೊಂದಿದವು. ಇಲ್ಲಿ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಎಂದರೆ ಗುಣಲಕ್ಷಣಗಳನ್ನು (properties) ಹೊಂದಿದ ಡಾಟಾ ಎಂದು ಅರ್ಥ.

4 **ಸೆಮಿಸ್ಟ್ರಕ್ಚರ್ಡ್ ಡಾಟಾ ಮಾಡೆಲ್** (Semistructured Data Model) : ಈ ರೀತಿಯ ಮಾಡೆಲ್ ಇತರ ಮೂರು ಡಾಟಾ ಮಾಡೆಲ್‌ಗಳಿಂತ (ಮೇಲೆ ವಿವರಿಸಲಾಗಿದೆ) ಭಿನ್ನವಾಗಿದೆ. ವಿಭಿನ್ನ ಆಟ್ರಿಬ್ಯೂಟ್ ಸೆಟ್‌ನ್ನು ಹೊಂದಿರಬಹುದಾದ ಒಂದೇ ಟೈಪನ್ನು ಹೊಂದಿರುವ ಪ್ರತಿಯೊಂದು ಡಾಟ ಐಟಮ್ ಸ್ಥಳಗಳಲ್ಲಿ ಸೆಮಿಸ್ಟ್ರಕ್ಚರ್ಡ್ ಡಾಟಾ ಮಾಡೆಲ್ ಡಾಟಾ ಸ್ಪೆಸಿಫಿಕೇಷನ್‌ಗಳನ್ನು (data specifications) ಅನುಮತಿಸುತ್ತದೆ. Extensible Markup Language, XML ಎಂದೂ ಕರೆಯಲಾಗುತ್ತದೆ, ಇದನ್ನು ಸೆಮಿಸ್ಟ್ರಕ್ಚರ್ಡ್ ಡಾಟಾವನ್ನು ಪ್ರತಿನಿಧಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅದಾಗ್ಯೂ, XML ಆರಂಭದಲ್ಲಿ ಟೆಕ್ಸ್ಟ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ಸೇರಿದ ಮಾರ್ಕಪ್ ಮಾಹಿತಿಗಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿತ್ತು. ಮಾಹಿತಿಯ ವಿನಿಮಯಕ್ಕಾಗಿ ಇದನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ ಇದು ಮಹತ್ವವನ್ನು ಗಳಿಸಿತು.

DBA ಪರಿಕಲ್ಪನೆ (Concept of DBA)

ಒಂದಕ್ಕಿಂತ ಇನ್ನೊಂದು ಸ್ವಲ್ಪಮಟ್ಟಿಗೆ ವಿಭಿನ್ನವಾಗಿರುವ ಅನೇಕ ರೀತಿಯ ಡಾಟಾಬೇಸ್‌ಗಳು ಇರುತ್ತವೆ. DBA ಎಂದರೆ ಸಾಮಾನ್ಯ ಡಾಟಾಬೇಸ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್‌ಗಳಾಗಿವೆ, ಇವುಗಳಿಗೆ ವಿಷಯಗಳನ್ನು ಹೇಗೆ ನಿಭಾಯಿಸಬೇಕು ಎಂದು ತಿಳಿದಿದೆ. ಒಂದು ನಿರ್ದಿಷ್ಟ ಡಾಟಾಬೇಸನ್ನು ನಿರ್ವಹಿಸುವುದರಲ್ಲಿ DBA ಗಳು ಅನುಭವವನ್ನು ಹೊಂದಿರುತ್ತವೆ. MySQL DBA , MySQL ಡಾಟಾಬೇಸನ್ನು ನಿರ್ವಹಿಸುವ ನುರಿತ ಒಂದು ಡಾಟಾಬೇಸ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್ ಆಗಿದೆ.

MYSQL ಇವತ್ತಿನ ದೊಡ್ಡ ಡಾಟಾ ಪರಿಸರ ವ್ಯವಸ್ಥೆಯ, ಚನ್ನಾಗಿ ಗೊತ್ತಿರುವ, ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಮತ್ತು ಸಮರ್ಥ ತಂತ್ರಜ್ಞಾನವಾಗಿದೆ. ಉದ್ಯಮ ಜಗತ್ತಿನ ಅಥವಾ ಸಾಮಾನ್ಯ IT ಜಗತ್ತಿನೊಂದಿಗೆ ತೊಡಗಿಸಿಕೊಂಡಿರುವ ಯಾರಿಗಾದರೂ MYSQL ಬಗೆಗಿನ ಸಾಮಾನ್ಯ ಜ್ಞಾನವು ಅವಶ್ಯಕವಾಗಿದೆ. ಮತ್ತು ರಿಲೇಷನಲ್ ಸಿಸ್ಟಮ್‌ಗಳ ಬಗ್ಗೆ ಗುರುತು ಇಲ್ಲದವರೂ ಕೂಡ MYSQL ಉಪಯೋಗಿಸಿ ಡಾಟಾ ಸಂಗ್ರಹದ ಬಗ್ಗೆ ಬೇಗನೆ ತಿಳಿದುಕೊಳ್ಳಬಹುದು. ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿರುವುದು ಮತ್ತು

© NIMI NOT TO BE REPUBLISHED

ರಿಲೇಷನಲ್ ಸಹಜ ಗುಣ ಹೊಂದಿರುವುದು ಈ ಡಾಟಾಬೇಸ್‌ನ ಅತಿ ದೊಡ್ಡ ಅನುಕೂಲತೆಯಾಗಿದೆ. ಇದನ್ನು ದೊಡ್ಡ ಮತ್ತು ಸಣ್ಣ ಎರಡೂ ವ್ಯವಹಾರಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

MYSQL ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು ವ್ಯವಸ್ಥಿತವಾಗಿ ಡಾಕ್ಯುಮೆಂಟ್ ಮಾಡಲ್ಪಟ್ಟಿರುತ್ತವೆ ಮತ್ತು ಅದನ್ನು ಬೆಂಬಲಿಸುವ ದೊಡ್ಡ ಸಮುದಾಯವನ್ನು ಹೊಂದಿದೆ ಆದರೂ MYSQL ಡಾಟಾಬೇಸನ್ನು ಉಪಯೋಗಿಸುವಾಗ ಕೆಲವು ಸಂಗತಿಗಳನ್ನು ಪರಿಗಣನೆಗೆ ತೆಗೆದುಕೊಳ್ಳುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

ಡಾಟಾಬೇಸ್ ಬಳಕೆದಾರರು (Database Users):ಕೆಲವೊಮ್ಮೆ ನೀವು MYSQL ನಲ್ಲಿ ಡಾಟಾಬೇಸನ್ನು ನಿರ್ವಹಿಸಬೇಕಾಗುತ್ತದೆ. ಆ ಸಂದರ್ಭದಲ್ಲಿ ನಾವು ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಬಳಕೆದಾರರ ಖಾತೆಯ ಪಟ್ಟಿಯನ್ನು ನೋಡಬೇಕಾಗುತ್ತದೆ. ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ, ಡಾಟಾಬೇಸ್ ಸರ್ವರ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಎಲ್ಲ ಬಳಕೆದಾರರ ಪಟ್ಟಿಯನ್ನು ಪ್ರದರ್ಶಿಸಲು, SHOW DATABASES, SHOW TABLES ನಂತಹ SHOW USERS ಕಮಾಂಡ್ ಇರಬಹುದೆಂದು ನಾವು ಊಹಿಸುತ್ತೇವೆ. ದುರದೃಷ್ಟವಶಾತ್ MYSQL ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ, MYSQL ಸರ್ವರ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಎಲ್ಲ ಬಳಕೆದಾರರ ಪಟ್ಟಿಯನ್ನು ಪ್ರದರ್ಶಿಸಲು, SHOW USERS ಎನ್ನುವ ಕಮಾಂಡ್ ಇರುವುದಿಲ್ಲ, ಡಾಟಾಬೇಸ್ ಸರ್ವರ್‌ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ಎಲ್ಲ ಬಳಕೆದಾರರ ಪಟ್ಟಿಯನ್ನು ನೋಡಲು , ಈ ಕೆಳಗಿನ ಕ್ವೆರಿಯನ್ನು ನಾವು ಉಪಯೋಗಿಸಬಹುದು :

```
mysql> Select user from mysql.user;
```

ಈ ಮೇಲಿನ ಕಮಾಂಡ್ ಯಶಸ್ವಿಯಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸಿದ ಮೇಲೆ, ನಾವು MYSQL ಡಾಟಾಬೇಸ್ ಸರ್ವರ್‌ನ ಯೂಸರ್ ಟೇಬಲ್‌ನಿಂದ ಯೂಸರ್ ಡಾಟಾವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತೇವೆ.

ಡಾಟಾಬೇಸ್ ಸ್ಕೀಮಾ (Database Schema)

MYSQL ಸ್ಕೀಮಾ ಒಂದು ಸಿಸ್ಟಮ್ ಸ್ಕೀಮಾ ಆಗಿದೆ. ಇದು MYSQL ಸರ್ವರ್‌ಗೆ ಅವಶ್ಯಕವಿರುವ ಮಾಹಿತಿಯನ್ನು ಟೇಬಲ್‌ಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ.MySQL ಸರ್ವರ್ ರನ್ ಆದಾಗ ಅದಕ್ಕೆ ಅವಶ್ಯಕವಿರುವ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿರುವ ಟೇಬಲ್‌ಗಳನ್ನು ಇದು ಒಳಗೊಂಡಿರುತ್ತದೆ. MySQL ಸ್ಕೀಮಾ, ಡಾಟಾ ಆಬ್ಜೆಕ್ಟ್ ಮೆಟಾಡಾಟಾವನ್ನುಸಂಗ್ರಹಿಸಿರುವ ಡಾಟಾ ಡಿಕ್ಷನರಿ ಟೇಬಲ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ ಮತ್ತು ಸಿಸ್ಟಮ್ ಟೇಬಲ್‌ಗಳನ್ನು ಇತರ ಕಾರ್ಯನಿರ್ವಹಣೆಗಳ ಉದ್ದೇಶಕ್ಕಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಸ್ಟಮ್ ಟೇಬಲ್‌ಗಳ ಗುಂಪನ್ನು ಮತ್ತೆ ಸಣ್ಣ ವರ್ಗಗಳಲ್ಲಿ ವಿಭಜಿಸಲಾಗಿದೆ.

- Data Dictionary Tables
- Grant System Tables
- Object Information System Tables
- Log System Tables
- Server-Side Help System Tables
- Time Zone System Tables
- Replication System Tables
- Optimizer System Tables
- Miscellaneous System Tables

ಈ ವಿಭಾಗವು ಪ್ರತಿ ವರ್ಗದಲ್ಲಿನ ಟೇಬಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ ಹಾಗೂ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ. ಡಾಟಾ ಡಿಕ್ಷನರಿ ಟೇಬಲ್‌ಗಳು ಹಾಗೂ ಸಿಸ್ಟಮ್ ಟೇಬಲ್‌ಗಳು InnoDB ಸ್ಟೋರೇಜ್ ಇಂಜಿನ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ.

MYSQL ಡಾಟಾ ಡಿಕ್ಷನರಿಯಲ್ಲಿ, mysql ಸಿಸ್ಟಮ್ ಟೇಬಲ್‌ಗಳು ಮತ್ತು ಡಾಟಾ ಡಿಕ್ಷನರಿ ಟೇಬಲ್‌ಗಳು mysql.idb ಎನ್ನುವ ಹೆಸರಿನ ಒಂದೇ InnoDB ಟೇಬಲ್‌ಸ್ಪೇಸ್ ಫೈಲಿನಲ್ಲಿ ಇರುತ್ತವೆ. ಇದಕ್ಕೂ ಮುಂಚೆ, MYSQL ಡಾಟಾ ಡಿಕ್ಷನರಿಯಲ್ಲಿ ಈ ಟೇಬಲ್‌ಗಳು ವೈಯಕ್ತಿಕ ಟೇಬಲ್‌ಸ್ಪೇಸ್ ಫೈಲ್‌ಗಳಲ್ಲಿ ರಚಿಸಲಾಗುತ್ತಿದ್ದವು.

Data-at-rest ಎನ್ಕ್ರಿಪ್ಷನ್‌ನ್ನು, mysql ಸಿಸ್ಟಮ್ ಸ್ಕೀಮಾ ಟೇಬಲ್‌ಸ್ಪೇಸ್‌ಗಾಗಿ ಸಕ್ರೀಯಗೊಳಿಸಬಹುದು. ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ, ವಿಭಾಗ 15.13, "InnoDB Data-at-Rest Encryption ನೋಡಿರಿ".

ನಾರ್ಮಲೈಸೇಷನ್ ನಿಯಮಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಡಾಟಾಬೇಸನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸುವುದು, ವಿವಿಧ ಡೈಟಾಟೈಪ್‌ಗಳು, ಡಾಟಾದ ಸಮಗ್ರತೆ, DDL, DML, ಮತ್ತು DCL ಕಮಾಂಡ್‌ಗಳು, ಪ್ರೈಮರಿ ಕೀ ಮತ್ತು ಫಾರಿನ್ ಕೀ ಉಪಯೋಗಿಸುವುದು (Designing database using normalization rules,various datatypes,data integrity,DDL,DML&DCL Statements Enforcing Primary key and Foreign key)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- MySQL ನಲ್ಲಿ ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್
- ವಿವಿಧ ಡಾಟಾಟೈಪ್‌ಗಳು
- ಡಾಟಾ ಸಮಗ್ರತೆ
- DDL, DML, ಮತ್ತು DCL ಕಮಾಂಡ್‌ಗಳು, MySQL ನಲ್ಲಿ ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್
- ಪ್ರೈಮರಿ ಕೀ ಮತ್ತು ಫಾರಿನ್ ಕೀ ಉಪಯೋಗಿಸುವುದು
- ಇಂಡೈಸ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು

MySQL ನಲ್ಲಿ ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್

ಡಾಟಾಬೇಸ್ ವಿನ್ಯಾಸಗೊಳಿಸುವಲ್ಲಿ ಮತ್ತು ಡಾಟಾ ಮಾಡೆಲಿಂಗ್‌ನಲ್ಲಿ ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್ ಒಂದು ಅತ್ಯಂತ ಪ್ರಮುಖವಾದ ಅಂಶವಾಗಿದೆ. ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್ ಡಾಟಾದ ಪುನರುಕ್ತಿಗಳನ್ನು ತೆಗೆದುಹಾಕಲು ಮತ್ತು ಡಾಟಾ ನಿರ್ವಹಣೆಯನ್ನು ಸುಲಭಗೊಳಿಸಲು ಡಾಟಾವನ್ನು ತಾರ್ಕಿಕವಾಗಿ ಸಂಗ್ರಹಿಸುವ ಒಂದು ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಡಾಟಾಬೇಸ್ ರಿಲೇಷನ್‌ಶಿಪ್‌ಗಳು ಮತ್ತು ಕೀಗಳು ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ಅತ್ಯಂತ ಉಪಯುಕ್ತವಾಗಿವೆ. ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್‌ನ್ನು E. F Codd ಇವರು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದರು. ಡಾಟಾಬೇಸ್ ನಾರ್ಮಲೈಸೇಷನ್ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ನಾರ್ಮಲ್ ಫಾರ್ಮ್ಸ್ (Normal Forms) ಎನ್ನುವ ನಿಯಮಗಳ ಸರಣಿಗಳು ಇವೆ.

ಮುಖ್ಯವಾಗಿ 6 ರೀತಿಯ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ಗಳು ಇರುತ್ತವೆ.: ಮೊದಲ ನಾರ್ಮಲ್ ಫಾರ್ಮ್(First Normal Form (1NF)), ಎರಡನೆ ನಾರ್ಮಲ್ ಫಾರ್ಮ್ (second Normal Form (2NF)), ಮೂರನೇಯ ನಾರ್ಮಲ್ ಫಾರ್ಮ್ (Third Normal Form (3NF)), ನಾಲ್ಕನೇಯ ನಾರ್ಮಲ್ ಫಾರ್ಮ್ (Fourth Normal Form (4NF)) ಐದನೇಯ ನಾರ್ಮಲ್ ಫಾರ್ಮ್(Fifth Normal Form (5NF)) , ಮತ್ತು ಬಾಯ್ಸ್ ಕಾಡ್ ನಾರ್ಮಲ್ ಫಾರ್ಮ್ (Boyce Codd Normal Form (BCNF)). ಆದರೆ ಮುಖ್ಯವಾಗಿ ನಾವು ನಮ್ಮ ಡಾಟಾಬೇಸ್‌ನ ವಿನ್ಯಾಸದಲ್ಲಿ ಮೂರನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ವರೆಗೆ ಮಾತ್ರ ಉಪಯೋಗಿಸಿದ್ದೇವೆ.

1 First Normal Form (1NF): ಮೊದಲ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿ, ಪ್ರತಿ ಕಾಲಂ ಒಂದು ಮೌಲ್ಯವನ್ನು ಮಾತ್ರ ಹೊಂದಿರಬೇಕು ಮತ್ತು ಯಾವುದೇ ಟೇಬಲ್ ಸಂಬಂಧಿತ ಡೇಟಾದ ಪುನರಾವರ್ತನೆಗೊಳ್ಳುವ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸಬಾರದು.

2 Second Normal Form (2NF): ಎರಡನೆ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿ ಮೊದಲು ಡಾಟಾಬೇಸ್ ಮೊದಲನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿ ಇರಬೇಕು ಮತ್ತು ಅದು ಒಂದೇ ಟೇಬಲ್‌ನಲ್ಲಿ ಒಂದೇ ತರಹದ ರೋ ಗಳನ್ನು (duplicate rows) ಸಂಗ್ರಹಿಸಿರಬಾರದು. ಮತ್ತು ರೋ ನಲ್ಲಿ ಒಂದೇ ತರಹದ ಡಾಟಾ ಇದ್ದರೆ, ಅವುಗಳು ತಮ್ಮದೇ ಆದ ಪ್ರತ್ಯೇಕ ಟೇಬಲ್‌ಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸಲ್ಪಡಬೇಕು ಮತ್ತು ಅವುಗಳನ್ನು ಫಾರಿನ್ ಕೀಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಟೇಬಲ್‌ಗೆ ಲಿಂಕ್ ಮಾಡಲಾಗುತ್ತದೆ. ಎರಡನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿ ಡಾಟಾಬೇಸ್‌ಗೆ ಒನ್ ಟು ಮೆನಿ (One to Many) ರಿಲೇಷನ್‌ಶಿಪ್ ಟೇಬಲ್‌ನ್ನು ರಚಿಸುವುದು ಸೂಕ್ತ ಮಾರ್ಗವಾಗಿರುತ್ತದೆ.

3 Third Normal Form (3NF): ಮೂರನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿ, ಡಾಟಾಬೇಸ್ ಈಗಾಗಲೇ ಮೂರನೆ ಫಾರ್ಮ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ, ಒಂದು ವೇಳೆ ಅದು ಎರಡನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನಲ್ಲಿದ್ದರೆ, ಮತ್ತು ಪ್ರತಿ ಕೀಯನ್ನು ಹೊಂದಿಲ್ಲದ ಕಾಲಂ ಪರಸ್ಪರ ಸ್ವತಂತ್ರವಾಗಿರುತ್ತವೆ. ಟೇಬಲ್‌ನಲ್ಲಿ ಯಾವುದೇ ಕಾಲಂಗಳು ಪರಸ್ಪರ ಅವಲಂಬಿ ತವಾಗಿದ್ದರೆ ಅವುಗಳನ್ನು ಗುರುತಿಸಿ ಮತ್ತು ಆ ಕಾಲಂಗಳನ್ನು ಅದರದೇ ಆದ ಸ್ವಂತ ಟೇಬಲ್ ಆಗಿ ಒಡೆಯಿರಿ.

© NIMI NOT TO BE REPUBLISHED

4 Boyce Codd Normal Form (BCNF): ಇದು ಮೂರನೇ ನಾರ್ಮಲ್ ಫಾರ್ಮ್‌ನ ಅತ್ಯುನ್ನತ ಫಾರ್ಮ್ ಆಗಿದೆ. ಇದು 3NF ನಿಂದ ನಿರ್ವಹಿಸಲಾಗದ ವಿವಿಧ ರೀತಿಯ ವೈಪರೀತ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ವಿವಿಧ ಡಾಟಾ ಪ್ರಕಾರಗಳು (Various Data types)

ಡಾಟಾ ಸಮಗ್ರತೆ (Data Integrity)

ಯಾವುದೇ ಡಾಟಾಬೇಸ್ ಅಥವಾ ವೇರ್‌ಹೌಸ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ಡಾಟಾದ ಸ್ಥಿರತೆ, ನಿಖರತೆ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹತೆಯನ್ನು ಡಾಟಾ ಸಮಗ್ರತೆ ಎನ್ನುತ್ತಾರೆ. ಎಲ್ಲ ನಿಖರವಾದ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಹೊಂದಿದ ಸಂಪೂರ್ಣ ರಚನೆಯನ್ನು ಹೊಂದಿದ ಡಾಟಾವನ್ನು ಸಮಗ್ರತೆಯ ಡಾಟಾ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಡಾಟಾದ ಸಮಗ್ರತೆಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಹಲವಾರು ಅಂಶಗಳಿವೆ ಉದಾಹರಣೆಗೆ ಭೌತಿಕ ಸಮಗ್ರತೆ (physical integrity) ಇದು ಡಾಟಾವನ್ನು ಅಧಿಕೃತವಾಗಿ ಒಟ್ಟು ಮಾಡಿ ಸಂಗ್ರಹಿಸುವುದು. ಲಾಜಿಕಲ್ ಸಮಗ್ರತೆ ಎಂದರೆ ಮಾಹಿತಿಯು ಸಂಬಂಧಿಸಿದ್ದಾಗಿದೆಯೆ ಮತ್ತು ನಿರ್ದಿಷ್ಟ ಸಂದರ್ಭದಲ್ಲಿ ನಿಖರವಾಗಿದೆಯೆ ಅಥವಾ ಇಲ್ಲವೆ ಎಂಬುದನ್ನು ಒಳಗೊಂಡಿದೆ. ಅದ್ದರಿಂದ ಒಟ್ಟಾರೆಯಾಗಿ ಡಾಟಾದ ಗುಣಮಟ್ಟವನ್ನು ಕಾಪಾಡಿಕೊಳ್ಳಲು ಅವಶ್ಯಕವಿರುವ ಎಲ್ಲ ನಿಯಮಗಳು ಡಾಟಾದ ಸಮಗ್ರತೆಯ ಅಡಿಯಲ್ಲಿ ಬರುತ್ತವೆ. ಸಮಗ್ರತೆ ಮತ್ತು ನಿಖರತೆ ಇಲ್ಲದೇ ನೀವು ಸಂಗ್ರಹಿಸಿದ ಎಲ್ಲ ಡಾಟಾ ಕಂಪನಿಗೆ ಅನುಪಯುಕ್ತವಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ, ಡಾಟಾದ ಸುರಕ್ಷತೆಯನ್ನು ಮತ್ತು ನಿಖರತೆಯನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುವುದು ಅತ್ಯಗತ್ಯವಾಗಿದೆ. ಇದು ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು ಮತ್ತು ಸ್ಥಿರತೆಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ. ದೋಷಪೂರಿತ ಡಾಟಾ ಕೆಲವೊಮ್ಮೆ ನಿಮ್ಮ ವ್ಯಾಪಾರವನ್ನು ಹಾನಿಗೊಳಿಸುತ್ತದೆ.

ಡಾಟಾ ಸ್ಥಿರವಾಗಿಲ್ಲದ ಕಾರಣಕ್ಕೆ ಅದನ್ನು ಬದಲಾಯಿಸುವುದು ಸುಲಭವಲ್ಲ. ನೀವು ಸ್ವೀಕರಿಸುವ ಮಾಹಿತಿಯು ವಿವಿಧ ಸ್ಥಳಗಳಿಂದ ಬರುತ್ತದೆ ಮತ್ತು ಡಾಟಾ ರಚಿಸಲ್ಪಟ್ಟ ಸೈಟನಿಂದ ಹಲವಾರು ಸಂಗತಿಗಳು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಬದಲಾಯಿಸಬಹುದು. ಹೆಚ್ಚಾಗಿ, ಅದನ್ನು ಇತರ ಸಾಧನಗಳಿಗೆ ವರ್ಗಾಯಿಸಬಹುದು, ಇದನ್ನು ಅಗತ್ಯಕ್ಕೆ ತಕ್ಕಂತೆ ಯಾರು ಬೇಕಾದರೂ ಮತ್ತು ಯಾವಾಗ ಬೇಕಾದರೂ ಬದಲಾಯಿಸಬಹುದು ಮತ್ತು ನವೀಕರಿಸಬಹುದಾಗಿದೆ.

ಮತ್ತು ಡಾಟಾದ ಸಮಗ್ರತೆಯನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುವುದು ಕೇವಲ ಒಂದು ಹಂತದ ಪ್ರಕ್ರಿಯೆಯಲ್ಲ . ಇದನ್ನು ಪ್ರತಿಯೊಂದು ಹಂತದಲ್ಲಿಯೂ ಪರಿಶೀಲಿಸಿ ದೃಢೀಕರಿಸಬೇಕಾಗುತ್ತದೆ, ಮಾಡೆಲ್ ವಿನ್ಯಾಸದಿಂದ ಹಿಡಿದು ಅದರ ಅಂತಿಮ ಫಲಿತಾಂಶದವರೆಗೆ.

DDL , DML and DCL ಕಮಾಂಡ್‌ಗಳು (DDL , DML and DCL Statements): ಸ್ಟ್ರಕ್ಚರ್ಡ್ ಕ್ವೆರಿ ಲ್ಯಾಂಗ್ವೇಜ್ (SQL) ನಮಗೆಲ್ಲರಿಗೂ ತಿಳಿದಿರುವಂತೆ ಒಂದು ಡಾಟಾಬೇಸ್ ಭಾಷೆಯಾಗಿದೆ. ಇದನ್ನು ಬಳಸಿ ನಾವು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇರುವ ಡಾಟಾಬೇಸ್ ಮೇಲೆ ಕೆಲವು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು ಮತ್ತು ಈ ಭಾಷೆಯನ್ನು ಡಾಟಾಬೇಸ್ ರಚಿಸಲು ಬಳಸಬಹುದು. Create, Drop, Insert ನಂತಹ ಕೆಲವು ಕಮಾಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ SQL ಅಗತ್ಯವಿರುವ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

SQL ಕಮಾಂಡ್‌ಗಳನ್ನು ಮುಖ್ಯವಾಗಿ ನಾಲ್ಕು ವರ್ಗಗಳಲ್ಲಿ ಈ ಕೆಳಗಿನಂತೆ ವಿಂಗಡಿಸಲಾಗಿದೆ :

1 DDL - Data Definition Language (ಡಾಟಾ ಡೆಫಿನಿಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್)

2 DML - Data Manipulation Language (ಡಾಟಾ ಮ್ಯಾನಿಪ್ಯುಲೇಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್)

3 DCL - Data Control Language (ಡಾಟಾ ಕಂಟ್ರೋಲ್ ಲ್ಯಾಂಗ್ವೇಜ್)

DDL - Data Definition Language (ಡಾಟಾ ಡೆಫಿನಿಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್)

DDL ಅಥವಾ ಡಾಟಾ ಡೆಫಿನಿಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್ ವಾಸ್ತವವಾಗಿ SQL ಕಮಾಂಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಈ ಕಮಾಂಡ್‌ಗಳನ್ನು ಡಾಟಾಬೇಸ್ ಸ್ಕೀಮಾವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಬಳಸಬಹುದು. ಇದು ಡಾಟಾಬೇಸ್‌ನ ವಿವರಣೆಯೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ಇದನ್ನು ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿರುವ ಡಾಟಾಬೇಸ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ರಚನೆಯನ್ನು ತಯಾರಿಸಲು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. DDL ಇದು SQL ಕಮಾಂಡ್‌ಗಳ ಒಂದು ಸೆಟ್ ಆಗಿದೆ. ಈ ಕಮಾಂಡ್‌ಗಳನ್ನು ಡಾಟಾಬೇಸ್ ರಚನೆಯನ್ನು ರಚಿಸಲು, ಮಾರ್ಪಾಡು ಮಾಡಲು ಮತ್ತು ಡಿಲಿೀಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಆದರೆ ಡಾಟಾವನ್ನಲ್ಲ. ಈ ಕಮಾಂಡ್‌ಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ , ಅಪ್ಲಿಕೇಷನ್ ಮೂಲಕ ಡಾಟಾಬೇಸನ್ನು ಪ್ರವೇಶಿಸುವ ಸಾಮಾನ್ಯ ಬಳಕೆದಾರರು ಉಪಯೋಗಿಸುವುದಿಲ್ಲ.

DDL ಕಮಾಂಡ್‌ಗಳ ಪಟ್ಟಿ (List of DDL commands)

- CREATE: ಈ ಕಮಾಂಡ್‌ನ್ನು ಡಾಟಾಬೇಸ್ ರಚಿಸಲು ಅಥವಾ ಅದರ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (ಉದಾಹರಣೆಗೆ , ಟೇಬಲ್, ಇಂಡೆಕ್ಸ್, ಫಂಕ್ಷನ್, ವಿವ್ಸ್ , ಸ್ಟೋರ್ ಪ್ರೊಸೀಜರ್, ಮತ್ತು ಟ್ರಿಗರ್ಸ್).
- DROP: ಈ ಕಮಾಂಡ್‌ನ್ನು ಡಾಟಾಬೇಸ್‌ನಿಂದ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ALTER: ಈ ಕಮಾಂಡ್‌ನ್ನು ಡಾಟಾಬೇಸ್‌ನ ರಚನೆಯನ್ನು ಬದಲಾಯಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

- TRUNCATE: ಈ ಕಮಾಂಡ್‌ನ್ನು ಒಂದು ಟೇಬಲ್‌ನಿಂದ ಎಲ್ಲ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ರೆಕಾರ್ಡ್‌ಗಳಿಗಾಗಿ ನಿಗದಿಪಡಿಸಿದ ಎಲ್ಲ ಸ್ಥಳಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಟೇಬಲ್‌ನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.
- COMMENT: ಈ ಕಮಾಂಡ್‌ನ್ನು ಡಾಟಾ ಡಿಕ್ಷನರಿಗೆ ಕಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- RENAME: ಈ ಕಮಾಂಡ್‌ನ್ನು ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿರುವ ಒಂದು ಆಬ್ಜೆಕ್ಟನ್ನು ಮರುಹೆಸರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

DML ಡಾಟಾ ಮ್ಯಾನಿಪ್ಯುಲೇಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್ (Data Manipulation Language)

ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ಇರುವ ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ SQL ಕಮಾಂಡ್‌ಗಳು DML ಅಥವಾ ಡಾಟಾ ಮ್ಯಾನಿಪ್ಯುಲೇಷನ್ ಲ್ಯಾಂಗ್ವೇಜ್‌ಗೆ ಸೇರಿವೆ ಮತ್ತು ಇದು ಹೆಚ್ಚಿನ SQL ಕಮಾಂಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಇದು ಡಾಟಾ ಮತ್ತು ಡಾಟಾಬೇಸ್ ಪ್ರವೇಶವನ್ನು ನಿಯಂತ್ರಿಸುವ SQL ಕಮಾಂಡ್‌ಗಳ ಒಂದು ಅಂಶವಾಗಿದೆ. ಮೂಲಭೂತವಾಗಿ, DCL ಕಮಾಂಡ್‌ಗಳನ್ನು DML ಕಮಾಂಡ್‌ಗಳೊಂದಿಗೆ ಗುಂಪು ಮಾಡಲಾಗಿದೆ.

DML ಕಮಾಂಡ್‌ಗಳ ಪಟ್ಟಿ (List of DML commands)

- INSERT : ಇದನ್ನು ಟೇಬಲ್‌ನಲ್ಲಿ ಡಾಟಾ ಸೇರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- UPDATE: ಇದನ್ನು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- DELETE : ಇದನ್ನು ಒಂದು ಡಾಟಾಬೇಸ್ ಟೇಬಲ್‌ನಲ್ಲಿರುವ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- LOCK: ಟೇಬಲ್ ನಿಯಂತ್ರಣಕ್ಕೆ ಸಮ್ಮತಿಸುತ್ತದೆ.
- CALL: PL/SQL ಅಥವಾ JAVA ಸಬ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಕಾಲ್ ಮಾಡುತ್ತದೆ.
- EXPLAIN PLAN: ಇದು ಡಾಟಾಗೆ ಪ್ರವೇಶ ಮಾರ್ಗವನ್ನು ವಿವರಿಸುತ್ತದೆ.

DCL ಡಾಟಾ ಕಂಟ್ರೋಲ್ ಲ್ಯಾಂಗ್ವೇಜ್ (Data Control Language)

DCL - GRANT ಮತ್ತು REVOKE ನಂತಹ ಕಮಾಂಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಮುಖ್ಯವಾಗಿ ಡಾಟಾಬೇಸ್ ಸಿಸ್ಟಮ್‌ನ ಹಕ್ಕುಗಳು, ಅನುಮತಿಗಳು ಮತ್ತು ಇತರ ನಿಯಂತ್ರಣಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

DCL ಕಮಾಂಡ್‌ಗಳ ಪಟ್ಟಿ (List of DCL commands)

- GRANT: ಈ ಕಮಾಂಡ್ ಬಳಕೆದಾರರಿಗೆ ಡಾಟಾಬೇಸ್‌ನ ಪ್ರವೇಶ ಸವಲತ್ತುಗಳನ್ನು ನೀಡುತ್ತದೆ.
- REVOKE: ಈ ಕಮಾಂಡ್ GRANT ಕಮಾಂಡ್ ಉಪಯೋಗಿಸಿ ಬಳಕೆದಾರರಿಗೆ ನೀಡಲಾದ ಪ್ರವೇಶ ಸವಲತ್ತನ್ನು ಹಿಂಪಡೆಯುತ್ತದೆ.

ಪ್ರೈಮರಿ ಕೀ (Primary Key)

ಪ್ರೈಮರಿ ಕೀ ಕನ್‌ಸ್ಟ್ರೇಂಟ್(Primary Key Constraint)ನ್ನು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿ ಪ್ರೈಮರಿ ಕೀ ಯನ್ನು ರಚಿಸುವ ಮೂಲಕ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಪ್ರೈಮರಿ ಕೀನಲ್ಲಿನ ಮೌಲ್ಯವು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿಯ ಪ್ರತಿ ರೋ ಅನ್ನು ವಿಭಿನ್ನವಾಗಿ (unique) ಗುರುತಿಸಬೇಕು. ಪ್ರೈಮರಿ ಕೀ ಕನ್‌ಸ್ಟ್ರೇಂಟ್ ರೋ ನ ಸಮಗ್ರತೆಯನ್ನು ಜಾರಿಗೊಳಿಸುತ್ತದೆ.

ಪ್ರೈಮರಿ ಕೀಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವಾಗ ಈ ಕೆಳಗಿನ ಸಂಗತಿಗಳನ್ನು ಪರಿಗಣಿಸಬೇಕು:

- ನೀವು ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿ ಪ್ರೈಮರಿ ಕೀಯನ್ನು ರಚಿಸಿದಾಗ, ಒಂದು ವಿಭಿನ್ನ ಸೂಚ್ಯಂಕ(unique index) ತನ್ನಿಂದ ತಾನೇ ಮತ್ತು ಸೂಚ್ಯವಾಗಿ ರಚಿಸಲ್ಪಡುತ್ತದೆ. ಇದು ಈ ಕಾಲಂನಲ್ಲಿ ಡಾಟಾದ ಅನನ್ಯತೆಯನ್ನು ಜಾರಿಗೊಳಿಸುತ್ತದೆ.
- ಪ್ರೈಮರಿ ಕೀ ಕಾಲಂ NULL ಮೌಲ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿರಬಾರದು. ಡಾಟಾಬೇಸ್ ಇಂಜಿನ್ ಪ್ರೈಮರಿ ಕೀ ಕಾಲಂನಲ್ಲಿ ಸೂಚ್ಯವಾಗಿ NOT NULL constraint ಅನ್ನು ರಚಿಸುತ್ತದೆ.
- ಒಂದು ಟೇಬಲ್‌ನಲ್ಲಿ ಒಂದೇ ಪ್ರೈಮರಿ ಕೀ ಯನ್ನು ಮಾತ್ರ ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು.
- ಪ್ರೈಮರಿ ಕೀಯನ್ನು ಒಂದೇ ಕಾಲಂನಿಂದ ಅಥವಾ ಒಂದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು ಕಾಲಂಗಳ ಸಂಯೋಜನೆಯಿಂದ ರಚಿಸಬಹುದು.(ಎಂದರೆ, ಕಾಂಪೋಸಿಟ್ ಪ್ರೈಮರಿ ಕೀ)
- ನೀವು ಪ್ರೈಮರಿ ಕೀನಲ್ಲಿ ಒಂದೆ ತರಹದ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಸೇರಿಸಿದರೆ, ನೀವು ದೋಷ (error) ಸಂದೇಶವನ್ನು ಪಡೆಯುತ್ತೀರಿ.

ಪ್ರೈಮರಿ ಕೀ ನಲ್ಲಿ ಎರಡು ವಿಧಗಳಿವೆ : ಕೃತಕ ಪ್ರೈಮರಿ ಕೀ (artificial primary key) , ನೈಸರ್ಗಿಕ ಪ್ರೈಮರಿ ಕೀ (natural primary key)

ಫಾರಿನ್ ಕೀ (Foreign Key)

MYSQL ಫಾರಿನ್ ಕೀಯನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ, ಇದು ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಸಂಬಂಧಿತ ಡಾಟಾವನ್ನು ಕ್ರಾಸ್ ರೆಫರನ್ಸ್ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ. ಇದು ಸಂಬಂಧಿತ ಡಾಟಾವನ್ನು ಸ್ಥಿರವಾಗಿಡಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಫಾರಿನ್ ಕೀ ರಿಲೇಷನ್‌ಶಿಪ್, ಆರಂಭಿಕ ಕಾಲಂನ ಮೌಲ್ಯಗಳನ್ನು ಹೊಂದಿರುವ ಒಂದು ಪೇರೆಂಟ್ ಟೇಬಲ್‌ನ್ನು ಮತ್ತು ಪೇರೆಂಟ್ ಕಾಲಂನ ಮೌಲ್ಯಗಳನ್ನು ಉಲ್ಲೇಖಿಸುವ ಚೈಲ್ಡ್ ಟೇಬಲ್‌ನ ಕಾಲಂ ಮೌಲ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕ್ವೆರಿಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ಡಿಲೀಟ್ ಮಾಡುವುದು (Insert and delete queries Update queries)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಸರಳವಾದ select ಕ್ವೆರಿಗಳು**

ಕ್ವೆರಿಗಳು - ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಪರಿಕಲ್ಪನೆ (Queries concept of transaction)

ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಇದು ಡಾಟಾಬೇಸ್ ಮ್ಯಾನಿಪ್ಯುಲೇಷನ್ ಕಾರ್ಯಗಳ ಒಂದು ಅನುಕ್ರಮ ಗುಂಪು ಆಗಿದೆ. ಇದು ಒಂದೇ ಕೆಲಸದ ಘಟಕದಂತೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಬೇರೆ ಶಬ್ದಗಳಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ಈ ಗುಂಪಿನಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಕಾರ್ಯವು ಸಂಪೂರ್ಣವಾಗಿ ಮುಗಿಯದೇ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಎಂದಿಗೂ ಯಶಸ್ವಿಯಾಗಿ ಪೂರ್ಣಗೊಳ್ಳುವುದಿಲ್ಲ. ಒಂದು ವೇಳೆ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ನಲ್ಲಿ ಯಾವುದೇ ಕೆಲಸವು ವಿಫಲವಾದರೆ, ಸಂಪೂರ್ಣ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ವಿಫಲವಾಗುತ್ತದೆ.

ಪ್ರಾಯೋಗಿಕವಾಗಿ, ನೀವು ಅನೇಕ SQL ಕ್ವೆರಿಗಳನ್ನು ಒಂದು ಗುಂಪಿನಲ್ಲಿ ಸೇರಿಸುತ್ತೀರಿ ಮತ್ತು ಅವೆಲ್ಲವುಗಳನ್ನು ನೀವು ಒಟ್ಟಿಗೆ ಒಂದು ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ನ ಭಾಗವಾಗಿ ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತೀರಿ. (execute)

ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ನ ಗುಣಲಕ್ಷಣಗಳು (Properties of Transactions)

ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ಗಳು ಈ ಕೆಳಗಿನ ನಾಲ್ಕು ಗುಣಲಕ್ಷಣಗಳನ್ನು ಹೊಂದಿವೆ. ಸಾಮಾನ್ಯವಾಗಿ ACID ಎಂಬ ಸಂಕ್ಷಿಪ್ತ ರೂಪದಿಂದ ಉಲ್ಲೇಖಿಸಲಾಗುತ್ತದೆ.

- Atomicity **-ಆಟೋಮಿಸಿಟಿ** - ಇದು ಒಂದು ಕೆಲಸದ ಘಟಕದಲ್ಲಿನ ಎಲ್ಲ ಕಾರ್ಯಗಳು ಯಶಸ್ವಿಯಾಗಿ ಪೂರ್ಣಗೊಂಡಿವೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ. ಇಲ್ಲದಿದ್ದರೆ, ಕಾರ್ಯಾಚರಣೆಯು ವಿಫಲವಾದಲ್ಲಿಂದ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಸ್ಥಗಿತಗೊಳ್ಳುತ್ತದೆ. ಮತ್ತು ಹಿಂದಿನ ಕಾರ್ಯಗಳನ್ನು ಅವುಗಳ ಹಿಂದಿನ ಸ್ಥಿತಿಗೆ ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.
- Consistency **(ಸ್ಥಿರತೆ)** : ಇದು ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಯಶಸ್ವಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಿದ ನಂತರ ಡಾಟಾಬೇಸ್ ಸರಿಯಾಗಿ ತನ್ನ ಸ್ಥಿತಿಯನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ.
- Isolation **(ಪ್ರತ್ಯೇಕತೆ)** : ಇದು ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ಗಳನ್ನು ಸ್ವತಂತ್ರವಾಗಿ ಮತ್ತು ಪರಸ್ಪರ ಪಾರದರ್ಶಕವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.
- Durability - **(ಬಾಳಿಕೆ)** - ಒಂದು ಸಿಸ್ಟಮ್ ವಿಫಲವಾದ ಸಂದರ್ಭದಲ್ಲಿ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್‌ನ ಫಲಿತಾಂಶ ಅಥವಾ ಪರಿಣಾಮ ದೃಢವಾಗಿ ಇರುತ್ತದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

Number, Date ಮತ್ತು Character ಫಂಕ್ಷನ್‌ಗಳನ್ನು, group by , having , sub query ಉಪಯೋಗಿಸುವುದು (Using the Number, Date and Character functions, group by having, sub query)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

• ಸಬ್ ಕ್ವೆರಿ ಬಗ್ಗೆ ತಿಳಿಯಿರಿ.

MySQL ಟೇಬಲ್ ಜಾಯಿನಿಂಗ್ (MySQL Joining Tables)

ಎರಡು ಅಥವಾ ಎರಡಕ್ಕಿಂತ ಹೆಚ್ಚು ಟೇಬಲ್‌ಗಳಲ್ಲಿನ ರೋ (row) ಗಳನ್ನು ಸಂಯೋಜಿಸಲು JOIN clause ಅನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಈ ಜೋಡಣೆಯು ಅವುಗಳ ನಡುವಿನ ಸಂಬಂಧಿತ ಕಾಲಂ ನ್ನು ಆಧರಿಸಿದೆ.

Orders ಟೇಬಲ್‌ನಿಂದ ಒಂದು ಆಯ್ಕೆಯನ್ನು ಗಮನಿಸಿ.

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

ನಂತರ, Customers ಟೇಬಲ್‌ನಿಂದ ಆಯ್ಕೆಯನ್ನು ನೋಡೋಣ.

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquerıa	Antonio Moreno	Mexico

Orders ಟೇಬಲ್‌ನಲ್ಲಿರುವ CustomerID ಕಾಲಂ Customers ಟೇಬಲ್‌ನಲ್ಲಿರುವ CustomerID ಯನ್ನು ಉಲ್ಲೇಖಿಸುತ್ತದೆ. ಮೇಲಿನ ಎರಡೂ ಟೇಬಲ್‌ಗಳ ನಡುವಿನ ಸಂಬಂಧ (relationship) CustomerID ಕಾಲಂ ಆಗಿದೆ.

ನಂತರ, ನಾವು ಈ ಕೆಳಗಿನ SQL ಕಮಾಂಡ್‌ನ್ನು ರಚಿಸಬಹುದು (ಇದು INNER JOIN ಒಳಗೊಂಡಿದೆ). ಇದು ಎರಡೂ ಟೇಬಲ್‌ಗಳಲ್ಲಿ ಹೊಂದಿಕೆಯಾಗುವ ಮೌಲ್ಯಗಳು ಇರುವ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ :

ಉದಾಹರಣೆಗೆ

```
SELECT Orders.OrderID, Customers.CustomerName,
Orders.OrderDate
FROM Orders
INNER JOIN Customers ON
Orders.CustomerID=Customers.CustomerID;
```

ಇದು ಕೆಳಗಿನಂತೆ ಉತ್ತರ ನೀಡುತ್ತದೆ.:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbkp	8/12/1996

MYSQL ನಲ್ಲಿ ಬೆಂಬಲಿಸಲ್ಪಡುವ Join ನ ವಿಧಗಳು (Supported Types of Joins in MySQL)

- INNER JOIN: ಇದು ಎರಡೂ ಟೇಬಲ್‌ಗಳಲ್ಲಿ ಹೊಂದಿಕೆಯಾಗುವ ಮೌಲ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
- LEFT JOIN: ಇದು ಎಡಗಡೆಯ ಟೇಬಲ್‌ನ ಎಲ್ಲ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಬಲಗಡೆಯ ಟೇಬಲ್‌ನಲ್ಲಿನ ಹೊಂದಿಕೆಯಾದ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

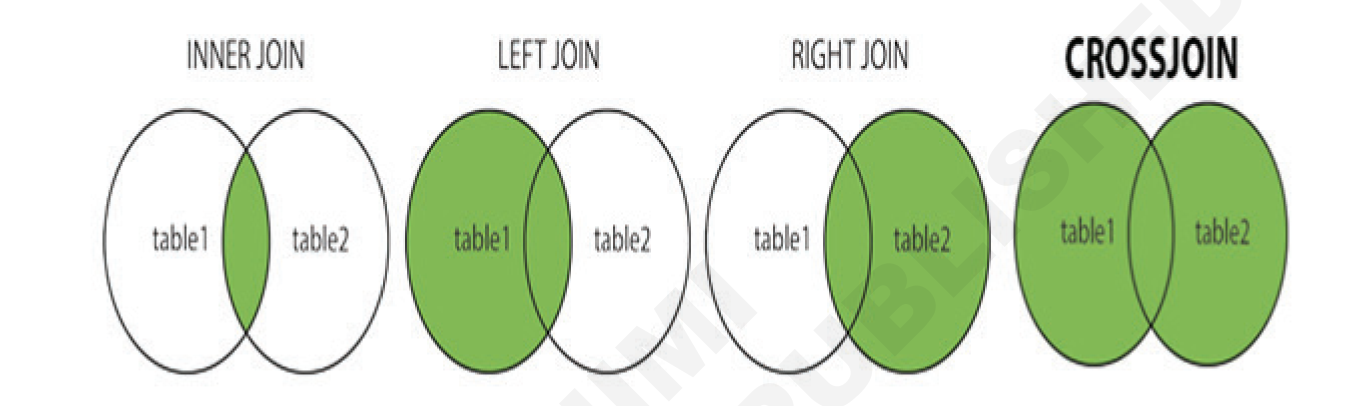
© NIMI NOT TO BE REPUBLISHED

- RIGHT JOIN: ಇದು ಬಲಗಡೆಯ ಎಲ್ಲ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಎಡಗಡೆಯ ಟೇಬಲ್‌ನ ಹೊಂದಿಕೆಯಾದ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

- CROSS JOIN:ಎರಡೂ ಟೇಬಲ್‌ಗಳಲ್ಲಿನ ಎಲ್ಲ ರೆಕಾರ್ಡ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಬ್ ಕ್ವೆರಿಗಳು (SUB QUERIES)

MySQL ನಲ್ಲಿ ಸಬ್ ಕ್ವೆರಿ ಎಂದರೆ, ಒಂದು ಕ್ವೆರಿ ಇನ್ನೊಂದು SQL ಕ್ವೆರಿಯಲ್ಲಿ ಅಡಗಿರುತ್ತದೆ ಮತ್ತು SELECT, INSERT, UPDATE ಅಥವಾ DELETE ಕಮಾಂಡ್‌ಗಳ ಜೊತೆಗೆ ಮತ್ತು ವಿವಿಧ ಆಪರೇಟರ್‌ಗಳ ಜೊತೆಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ನಾವು ಒಂದು ಸಬ್ ಕ್ವೆರಿಯನ್ನು ಇನ್ನೊಂದು ಸಬ್ ಕ್ವೆರಿಯಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು. ಸಬ್ ಕ್ವೆರಿಯನ್ನು ಇನ್ನರ್ ಕ್ವೆರಿ ಎಂತಲೂ ಕರೆಯುತ್ತಾರೆ. ಮತ್ತು ಸಬ್ ಕ್ವೆರಿಯನ್ನು ಹೊಂದಿರುವ ಕ್ವೆರಿಯನ್ನು ಔಟರ್ ಕ್ವೆರಿ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇನ್ನರ್‌ಕ್ವೆರಿ ಮೊದಲು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ (execute) ಉತ್ತರವನ್ನು ಔಟರ ಕ್ವೆರಿಗೆ ನೀಡುತ್ತದೆ ನಂತರ ಔಟರ್ ಕ್ವೆರಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. MySQL ನಮಗೆ ಸಬ್ ಕ್ವೆರಿಯನ್ನು ಎಲ್ಲಿಯಾದರೂ ಉಪಯೋಗಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ ಆದರೆ ಇದನ್ನು ಆವರಣದೊಳಗೆ (parentheses) ಇರಿಸಬೇಕು. SQL ಮಾನದಂಡದಲ್ಲಿ ಬೆಂಬಲಿತವಾದ ಎಲ್ಲ ಸಬ್ ಕ್ವೆರಿ ಫಾರ್ಮ್‌ಗಳು ಮತ್ತು ಕಾರ್ಯಗಳು MySQL ನಲ್ಲಿಯೂ ಕೂಡ ಬೆಂಬಲಿಸಲ್ಪಡುತ್ತವೆ.



ಸಬ್ ಕ್ವೆರಿಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಕೆಳಗಿನ ನಿಯಮಗಳನ್ನು ಅನುಸರಿಸಿ(The following are the rules to use subqueries)

- ಸಬ್ ಕ್ವೆರಿಗಳನ್ನು ಯಾವಾಗಲು ಆವರಣದಲ್ಲಿ (parentheses) ನಲ್ಲಿ ಬಳಸಬೇಕು.

- ಮುಖ್ಯ ಕ್ವೆರಿ ಸಬ್ ಕ್ವೆರಿಗಾಗಿ ಅನೇಕ ಕಾಲಂಗಳನ್ನು ಹೊಂದಿರದಿದ್ದರೆ, ಒಂದು ಸಬ್ ಕ್ವೆರಿ SELECT ಕಮಾಂಡ್‌ನಲ್ಲಿ ಒಂದು ಕಾಲಂನ್ನು ಮಾತ್ರ ಹೊಂದಿರಬಹುದು.

- ನಾವು ಸಬ್ ಕ್ವೆರಿಯಲ್ಲಿ ವಿವಿಧ ಹೋಲಿಕೆ (comparison) ಆಪರೇಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ , >, <, =, IN, ANY, SOME, ಮತ್ತು ALL. ಸಬ್ ಕ್ವೆರಿ ಒಂದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು ರೋ ಗಳನ್ನು ಹಿಂತಿರುಗಿಸಿದಾಗ multiple-row operator ಬಹಳ ಉಪಯುಕ್ತವಾಗಿದೆ.

- ನಾವು ಸಬ್ ಕ್ವೆರಿಯಲ್ಲಿ ORDER BY clause ಉಪಯೋಗಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. , ಅದಾಗ್ಯೂ ನಾವು ಇದನ್ನು ಮುಖ್ಯ ಕ್ವೆರಿಯಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು.

- ನಾವು ಸಬ್ ಕ್ವೆರಿಯನ್ನು ಒಂದು ಸೆಟ್ ಫಂಕ್ಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಿದರೆ, ಇದನ್ನು ತಕ್ಷಣದಲ್ಲಿ ಸೆಟ್ ಫಂಕ್ಷನ್‌ನಲ್ಲಿ ಎನ್‌ಕ್ಲೋಸ್ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಸಬ್ ಕ್ವೆರಿಗಳನ್ನು ಬಳಸುವ ಅನುಕೂಲಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ (he following are the advantages of using subqueries)

- ಸಬ್ ಕ್ವೆರಿಗಳು ಕ್ವೆರಿಯನ್ನು ರಚನಾತ್ಮಕ ರೂಪದಲ್ಲಿ ಮಾಡುತ್ತವೆ. ಇದು ಕಾಮಾಂಡ್‌ನ ಪ್ರತಿಯೊಂದು ಭಾಗವನ್ನು ಪ್ರತ್ಯೇಕಿಸಲು ನಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ.

- ಸಬ್ ಕ್ವೆರಿಗಳು ಟೇಬಲ್‌ನಿಂದ ಡಾಟಾವನ್ನು ಕ್ವೆರಿ ಮಾಡಲು ಪರ್ಯಾಯ ಮಾರ್ಗಗಳನ್ನು ಒದಗಿಸುತ್ತವೆ. ಇಲ್ಲದಿದ್ದರೆ, ನಾವು ಸಂಕೀರ್ಣವಾದ join ಮತ್ತು union ಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕಾಗುತ್ತದೆ.

- ಸಬ್ ಕ್ವೆರಿಗಳು ಸಂಕೀರ್ಣವಾದ join ಅಥವಾ union ಗಳಿಗಿಂತ ಹೆಚ್ಚು ಸುಲಭವಾಗಿವೆ.

MySQL ಸಬ್ ಕ್ವೆರಿ ಸಿಂಟ್ಯಾಕ್ಸ್ (MySQL Subquery Syntax)

MySQL ನಲ್ಲಿ ಸಬ್ ಕ್ವೆರಿ ಉಪಯೋಗಿಸುವ ಸಾಮಾನ್ಯ ರೂಪ ಈ ಕೆಳಗಿನಂತಿದೆ :

```
SELECT column_list (s) FROM table_name

WHERE column_name OPERATOR

(SELECT column_list (s) FROM table_name [WHERE])
```

© NIMI NOT TO BE REPUBLISHED

ಕ್ವೆರಿಯಲ್ಲಿ ಬಳಸುವ ಫಂಕ್ಷನ್‌ಗಳು (Functions used in Query)

1 sum :

ಮೌಲ್ಯಗಳ ಗುಂಪಿನ ಮೊತ್ತವನ್ನು ಲೆಕ್ಕ ಮಾಡುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ : SUM(expression)

2 average (avg):

ಎಕ್ಸ್ ಪ್ರೆಷನ್‌ನ ಸರಾಸರಿ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Syntax :AVG(expression)

3 MAX:

ಮೌಲ್ಯಗಳ ಗುಂಪಿನಲ್ಲಿರುವ ಗರಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ

Syntax: MAX(expression)

4 MIN :

ಮೌಲ್ಯಗಳ ಗುಂಪಿನಲ್ಲಿರುವ ಕನಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ

Syntax: MIN(expression)

5 Count :

Select ಕ್ವೆರಿಯಿಂದ ಬಂದಂತಹ ರೆಕಾರ್ಡ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Syntax: COUNT(expression)

© NIMI NOT TO BE REPUBLISHED

ಕಂಪ್ಯೂಟರ್‌ನ್ನು ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಿ ಮತ್ತು ಸಾಧನಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳಿರಿ ಎಂದರೆ, ಪ್ರಿಂಟರ್‌ಗಳು, ಫೈಲ್‌ಗಳು, ಫೋಲ್ಡರ್‌ಗಳು, ಮತ್ತು ಡ್ರೈವ್‌ಗಳು (Connect a computer to a network and share Devices i.e. Printers, files, folders and drives)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು
- ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಸಂವಹನ
- ಸಂವಹನಗಳ ತತ್ವಗಳು.

ಸಂಪರ್ಕಿತ ಜಗತ್ತಿನಲ್ಲಿ ಸಂವಹನ (Communicating in a connected world)

ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು(Local Networks)

ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್ (LAN) ಒಂದು ವಿಶಿಷ್ಟವಾದ ಭೌಗೋಳಿಕ ಸ್ಥಾನದಲ್ಲಿರುವ, ಒಂದು ಸಾಮಾನ್ಯ ಸಂವಹನ ಮಾಧ್ಯಮವನ್ನು ಅಥವಾ ವೈರ್‌ಲೆಸ್ ಲಿಂಕನ್ನು ಸರ್ವರ್ ಜೊತೆಗೆ ಹಂಚಿಕೊಳ್ಳುವ ಕಂಪ್ಯೂಟರಗಳ ಮತ್ತು ಬಾಹ್ಯ ಸಾಧನಗಳ ಗುಂಪಾಗಿದೆ. ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್ ಹೋಮ್ ಆಫೀಸ್‌ನಲ್ಲಿ ಎರಡು ಅಥವಾ ಮೂರು ಬಳಕೆದಾರರನ್ನು ಹೊಂದಿರಬಹುದು ಅಥವಾ ಕಾರ್ಪೋರೇಷನ್ ಸೆಂಟ್ರಲ್ ಆಫೀಸ್‌ನಲ್ಲಿ ಸಾವಿರಾರು ಬಳಕೆದಾರರನ್ನು ಹೊಂದಿರಬಹುದು.

ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಸಂವಹನ (Communicating on a local network) :

ಒಂದು ಪೀರ್ -ಟು - ಪೀರ್ LAN ನೇರವಾಗಿ ಎರಡು ಸಾಧನಗಳನ್ನು ಸಂಪರ್ಕಿಸುತ್ತದೆ - ಸಾಮಾನ್ಯವಾಗಿ, ವರ್ಕ್‌ಸ್ಟೇಷನ್‌ಗಳು ಅಥವಾ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು - ಒಟ್ಟಿಗೆ ಈಥರ್‌ನೆಟ್ ಕೇಬಲ್‌ನ್ನು ಬಳಸುತ್ತವೆ. ಕ್ಲೈಂಟ್ - ಸರ್ವರ್ LAN, ಅನೇಕ ಎಂಡ್‌ಪಾಯಿಂಟ್‌ಗಳನ್ನು ಮತ್ತು ಸರ್ವರ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇವು LAN ಸ್ವಿಚ್‌ಗೆ ಜೋಡಿಸಲ್ಪಟ್ಟಿರುತ್ತವೆ. ಸ್ವಿಚ್, ಅನೇಕ ಸಂಪರ್ಕಗೊಂಡ ಸಾಧನಗಳ ನಡುವೆ ಸಂವಹನ ಸ್ಟ್ರೀಮ್‌ಗಳನ್ನು ಕಳುಹಿಸುತ್ತದೆ.

ಸಂವಹನದ ತತ್ವಗಳು (Principles of Communications) :

ಸಂವಹನದ ವೃತ್ತಿಯಲ್ಲಿರುವವರು, ಕೆಲವೊಮ್ಮೆ 7 ಸಂವಹನದ ತತ್ವಗಳನ್ನು ಉಲ್ಲೇಖಿಸುತ್ತಾರೆ :ದುಃಖಕರವೆಂದರೆ, ಈ ತತ್ವಗಳು ಪ್ರಮಾಣೀಕೃತವಾಗಿಲ್ಲ (ಅದ್ದರಿಂದ ಬಹುಶಃ ಹಂತ 200!) ಆದರೆ ಮೈಕ್ರೊಸಾಫ್ಟ್ 365 Maturity Model ನಲ್ಲಿರುವ Communication Competency ಯನ್ನು ತಿಳಿಸಲು ಕೆಲವು ಸಾಮಾನ್ಯ ಥೀಮ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು ವೇಳೆ ನೀವು ಸಂವಹನದ ವೃತ್ತಿಪರರು ಆಗಿಲ್ಲದಿದ್ದರೆ, ಈ ತತ್ವಗಳ ಅರ್ಥ ವಿವರಣೆಯನ್ನು ಸಂವಹನದ ತಂತ್ರಗಳು, ಪರಿಪಕ್ವ ಮೌಲ್ಯಮಾಪನ ಅಥವಾ ತಂತ್ರಜ್ಞಾನದ ಆಯ್ಕೆ ಇವುಗಳ ಭಾಗವಾಗಿ ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತವೆ.

1 ಪರಿಣಾಮಕಾರಿ (Effective)
2 ವಿಸ್ತಾರವಾಗಿದೆ (Comprehensive)
3 ಸ್ಪಷ್ಟತೆ (Clarity)
4 ಗಮನ ಮತ್ತು ಶೈಲಿ (Attention and Style)
5 ಸುಸಂಬದ್ಧತೆ (Coherency)
6 ಸಮಯಪ್ರಜ್ಞೆ ಮತ್ತು ತುರ್ತು (Timeliness and Urgency)
7 ಪ್ರತಿಕ್ರಿಯೆಯ ಪ್ರಾಮುಖ್ಯತೆ (Importance of Feedback)

© NIMI NOT TO BE REPUBLISHED

ಈಥರ್‌ನೆಟ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು (Ethernet networks)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- ಈಥರ್‌ನೆಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- ಈಥರ್‌ನೆಟ್ ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ?
- ನಿರ್ಮಿಸಲಾದ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ವಿವರಿಸಿ.

ಈಥರ್‌ನೆಟ್ ಮೂಲ ನೆಟ್‌ವರ್ಕಿಂಗ್ ತಂತ್ರಜ್ಞಾನಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ, ಇದನ್ನು ಸುಮಾರು 50 ವರ್ಷಗಳ ಹಿಂದೆ ಆವಿಷ್ಕರಿಸಲಾಗಿದೆ. ಸಂವಹನದ ಪ್ರೊಟೋಕಾಲ್‌ಗಳನ್ನು ನಿಯೋಜಿಸಲು ಇರುವ ಅದರ ಸರಳತೆ ಮತ್ತು ಹೊಂದಾಣಿಕೆಯನ್ನು

ಹಿಂದಕ್ಕೆ ಕಳೆದುಕೊಳ್ಳದೇ ಇರುವ ಅದರ ಅಧುನಿಕತೆಯ ಪ್ರಗತಿಯನ್ನು ಅಳವಡಿಸುವ ಸಾಮರ್ಥ್ಯ ಇರುವುದರಿಂದ, ಈಥರ್‌ನೆಟ್ ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ವಾಸ್ತವವಾಗಿ ಇನ್ನೂ ತನ್ನ ಆಳ್ವಿಕೆಯನ್ನು ಮುಂದುವರಿದಿದೆ.

ಈಥರ್‌ನೆಟ್ ಒಂದು ಪ್ರೊಟೋಕಾಲ್ ಆಗಿದ್ದು, ಇದು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು (ಸರ್ವರ್‌ಗಳಿಂದ ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳವರೆಗೆ) ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇವುಗಳು ವಯರ್ ನೆಟ್‌ವರ್ಕ್ ಮೂಲಕ ಸಂಚಾರವನ್ನು ನಿರ್ದೇಶಿಸಲು ರೂಟರ್‌ಗಳು, ಸ್ವಿಚ್‌ಗಳು, ಮತ್ತು ಹಬ್‌ಗಳಂತಹ ಸಾಧನಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಈಥರ್‌ನೆಟ್ ವಯರ್‌ಲೆಸ್ ಪ್ರೊಟೋಕಾಲ್‌ಗಳೊಂದಿಗೂ ಕೂಡ ತಡೆರಹಿತವಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ.

ಯಾವುದೇ ಪರಿಸರದಲ್ಲಿಯೂ ಕೂಡ ಕೆಲಸ ಮಾಡುವ ಅದರ ಸಾಮರ್ಥ್ಯವು ಪ್ರಪಂಚದಾದ್ಯಂತ ಅದನ್ನು ಅಳವಡಿಸಿಕೊಳ್ಳುವಂತೆ ಮಾಡಿದೆ. ಇದು ವಿಶೇಷವಾಗಿ ನಿಜವಾಗಿದೆ, ಏಕೆಂದರೆ ಅದು ಸಂಸ್ಥೆಗಳಿಗೆ ಅವರ local area network (LAN) ಮತ್ತು wide-area network (WAN) ಎರಡಕ್ಕೂ ಒಂದೇ ಈಥರ್‌ನೆಟ್ ಪ್ರೊಟೋಕಾಲನ್ನು ಬಳಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಎಂದರೆ ಇಂಟರ್‌ನೆಟ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗಾಗಿ ಮತ್ತು ಬಹುತೇಕ ಇತರ ನಡುವಿನ ಕೆಲಸಗಳಿಗಾಗಿ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳಲ್ಲಿ, ಖಾಸಗಿ ಅಥವಾ ಆಂತರಿಕ ಕಂಪನಿ ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ಇದು ಚನ್ನಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್‌ನ ಅತ್ಯಂತ ಸಂಕೀರ್ಣ ರೂಪಗಳಾದ virtual private networks (VPNs) ಮತ್ತು software-defined networking deployments ಗಳನ್ನೂ ಸಹ ಬೆಂಬಲಿಸುತ್ತದೆ.

ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್ - ತೀವ್ರತೆಯ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಾದ ವಿಡಿಯೋ ಸ್ಟ್ರೀಮಿಂಗ್ ಅಥವಾ ವಾಯ್ಸ್ ಓವರ್ ಐಪಿ (voice over IP) ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಈಥರ್‌ನೆಟ್‌ಗೆ ಯಾವುದೇ ಸಮಸ್ಯೆ ಇರುವುದಿಲ್ಲ. ಮತ್ತು ಇನ್ನೊಂದೆಡೆ ಅದರ ಸರಳತೆ ಅದನ್ನು ಅತ್ಯಂತ ಚಿಕ್ಕದಾದ , ತುಲನಾತ್ಮಕವಾಗಿ ಸಂಕೀರ್ಣವಲ್ಲದ ಇಂಟರ್‌ನೆಟ್ ಆಫ್ ಥಿಂಗ್ಸ್ (Internet of Things - IoT)ರೂಪಿಸುವಂತಹ , ಸಾಧನಗಳೊಂದಿಗೆ ಯಾವುದೇ ವಿಶೇಷ ಸಂರಚನೆಯ ಅವಶ್ಯಕತೆ ಇಲ್ಲದೇ ಕೆಲಸ ಮಾಡಲು ಶಕ್ತಗೊಳಿಸುತ್ತದೆ.

ಈಥರ್‌ನೆಟ್ ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ ?(How does Ethernet work)?

ಈಥರ್‌ನೆಟ್ ಪರ್ಸನಲ್ ಕಂಪ್ಯೂಟರ್‌ನಂತಹ ಸಾಧನಗಳಿಂದ ಅಥವಾ ಸಾಧನಗಳಿಗೆ ಮಾಹಿತಿಯನ್ನು ಫ್ರೇಮ್ ಎನ್ನುವ ವಿವಿಧ ಗಾತ್ರದ ಬಿಟ್‌ಗಳಾಗಿ ಒಡೆದು ತುಂಡುಗಳನ್ನಾಗಿ ಮಾಡಿ ಕಳುಹಿಸುವ ಮೂಲಕ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಈ ಫ್ರೇಮ್‌ಗಳು ತಮಗೆ ನೆಟ್‌ವರ್ಕ್‌ನಾದ್ಯಂತ ಚಲಿಸಲು ಸಹಾಯ ಮಾಡುವ ಸೋರ್ಸ್(source) ಮತ್ತು ಡೆಸ್ಟಿನೇಷನ್(destination) ಅಡ್ರೆಸ್‌ನಂತಹ ಪ್ರಮಾಣಿತ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

LAN ನಲ್ಲಿರುವ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಒಂದೇ ಸಂಪರ್ಕವನ್ನು ಹಂಚಿಕೊಡಿರುವುದರಿಂದ, ಈಥರ್‌ನೆಟ್‌ನ್ನು CSMA/CD, or carrier-sense multiple access with collision detection ತತ್ವದ ಮೇಲೆ ನಿರ್ಮಿಸಲಾಗಿದೆ.

ಸಾಮಾನ್ಯವಾಗಿ, ಪ್ರೊಟೋಕಾಲ್ ಯಾವುದೇ ಫ್ರೇಮನ್ನು ಹೊರಗೆ ಕಳುಹಿಸುವ ಮೊದಲು ಲೈನ್ ಬಳಕೆಯಲ್ಲಿಲ್ಲ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುತ್ತದೆ. ಇವತ್ತು ಇದು ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನ ಆರಂಬಿಕ ದಿನಗಳಲ್ಲಿ ಇದ್ದುದಕ್ಕಿಂತ ಕಡಿಮೆ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ಹೊಂದಿದೆ. ಏಕೆಂದರೆ ಸಾಮಾನ್ಯವಾಗಿ ಸಾಧನಗಳು ಒಂದು ಸ್ವಿಚ್ ಅಥವಾ ನೋಡ್ ಮೂಲಕ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ತಮ್ಮದೇ ಆದ ಸ್ವಂತ ಖಾಸಗಿ ಸಂಪರ್ಕವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಮತ್ತು ಏಕೆಂದರೆ ಈಥರ್‌ನೆಟ್ ಈಗ ಫುಲ್ ಡುಪ್ಲೆಕ್ಸ್ ಬಳಸಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ - ಕಳುಹಿಸುವ ಮತ್ತು ಸ್ವೀಕರಿಸುವ ಚಾನಲ್‌ಗಳು ಸಂಪೂರ್ಣವಾಗಿ ಪ್ರತ್ಯೇಕವಾಗಿರುತ್ತವೆ. ಅದ್ದರಿಂದ ವಾಸ್ತವವಾಗಿ ಘರ್ಷಣೆಗಳು ಸಂಭವಿಸುವುದಿಲ್ಲ.

ಘರ್ಷಣೆಯ ಪರಿಸ್ಥಿತಿಯನ್ನು ಎದುರಿಸುವದನ್ನು ಹೊರತುಪಡಿಸಿ ಈಥರ್‌ನೆಟ್‌ನಲ್ಲಿಯಾವುದೇ ದೋಷ ತಿದ್ದುಪಡಿಯು ಇಲ್ಲ. ಅದ್ದರಿಂದ ಸಂವಹನಗಳು ಎಲ್ಲವೂ ಸಂಪೂರ್ಣವಾಗಿ ಕಳುಹಿಸಲ್ಪಟ್ಟಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಸುಧಾರಿತ ಪ್ರೊಟೋಕಾಲನ್ನು ಅವಲಂಬಿಸುವ ಅಗತ್ಯವಿದೆ. ಆದಾಗ್ಯೂ ಈಥರ್‌ನೆಟ್ ಹೆಚ್ಚಿನ ಇಂಟರ್‌ನೆಟ್ ಮತ್ತು ಡಿಜಿಟಲ್ ಸಂವಹನಗಳಿಗೆ ಆಧಾರವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಹೆಚ್ಚಿನ ಉನ್ನತ-ಮಟ್ಟದ ಪ್ರೊಟೋಕಾಲ್‌ಗಳೊಂದಿಗೆ ಸುಲಭವಾಗಿ ಸಂಯೋಜಿಸುತ್ತದೆ. ಅದ್ದರಿಂದ ಈ ದಿನಗಳಲ್ಲಿ ಅದು ಸಮಸ್ಯೆಯಾಗಿಲ್ಲ.

ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಹೇಗೆ ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ? (How are network built?)

ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ನೋಡ್‌ಗಳು ಮತ್ತು ಸಂಪರ್ಕಗಳನ್ನು(ಲಿಂಕ್‌ಗಳು) ಬಳಸಿಕೊಂಡು ರಚಿಸಲಾಗುತ್ತದೆ. ಇದು ಟೀಮ್‌ಗಳನ್ನು ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸಲು ಸಕ್ರಿಯಗೊಳಿಸುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ನೋಡ್ ನೆಟ್‌ವರ್ಕ್ ಇಂಟರ್‌ಫೇಸ್ ಕಾರ್ಡ್‌ನ್ನು ಬಳಸುತ್ತದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್‌ನೊಂದಿಗೆ ಹೇಗೆ ಸಂವಹನ ನಡೆಸುತ್ತದೆ ಎಂಬುದನ್ನು ವಿವರಿಸುತ್ತದೆ. ಪ್ರತಿ ನೆಟ್‌ವರ್ಕ್ ಕಾರ್ಡ್ ತನ್ನ ಸ್ವಂತ ಸಂಖ್ಯಾ ಅಡ್ರೆಸ್‌ನ್ನು ಹೊಂದಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಅಂತಿಮ ಬಳಕೆದಾರರ ಸಾಧನಗಳು ಮತ್ತು ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು (End-User Devices and local networks)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಸಂವಹನ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ**
- **ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ನೋಡ್ ಸಾಧನಗಳ ವಿಧಗಳು**
- **ಎಂಡ್ ಸಾಧನಗಳ ಕಾರ್ಯಗಳು**

ಜನರು ಹೆಚ್ಚು ಪರಿಚಿತರಾಗಿರುವ ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳನ್ನು ಎಂಡ್ ಸಾಧನಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಒಂದು ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಗೊಂಡಿರುವ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನದಲ್ಲಿ ನೇರವಾಗಿ ಭಾಗವಹಿಸುವ ಎಲ್ಲಾ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಹೋಸ್ಟ್‌ಗಳು ಎಂದು ವರ್ಗೀಕರಿಸಲಾಗುತ್ತದೆ. ಈ ಸಾಧನಗಳು ಬಳಕೆದಾರರು ಮತ್ತು ಸಂವಹನ ನೆಟ್‌ವರ್ಕ್ ನಡುವೆ ಇಂಟರ್‌ಫೇಸನ್ನು ರೂಪಿಸುತ್ತದೆ.

ಅಂತಿಮ ಬಳಕೆದಾರರ ನೋಡ್ (End user node)

ಅಂತಿಮ ಬಳಕೆದಾರರ ಸಾಧನಗಳಿಗೆ ಹಲವಾರು ಉದಾಹರಣೆಗಳಿವೆ - ಡೆಸ್ಕ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು, ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು, ಮತ್ತು ಸರ್ವರ್‌ಗಳು. ಈ ಅಂತಿಮ ಬಳಕೆದಾರರ ಸಾಧನಗಳಿಗೆ ಸಾಮಾನ್ಯ ಪದಗಳು ಯಾವವೆಂದರೆ ಹೋಸ್ಟ್ ಮತ್ತು ಎಂಡ್ ಯೂಸರ್ ನೋಡ್. ಈ ಪಾಠದ ಉದ್ದಕ್ಕೂ ನಾವು ಎಂಡ್ ಯೂಸರ್ ನೋಡ್ ಪದವನ್ನು ಬಳಸುತ್ತೇವೆ.

ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ನೋಡ್ ಸಾಧನಗಳ ವಿಧಗಳು :

ಅಂತಿಮ ಸಾಧನಗಳು ಮತ್ತು ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು

ಸಂವಹನ ಮಾಧ್ಯಮಗಳೆಂದು(ಅಥವಾ ಚಾನಲ್‌ಗಳು) ಕರೆಯಲ್ಪಡುವ ಕಮ್ಯೂನಿಕೇಷನ್ ಲಿಂಕ್‌ಗಳ ಮೂಲಕ ಸಂಪರ್ಕಿತಗೊಂಡ

ಅಂತರ್‌ಸಂಪರ್ಕಿತ ನೋಡ್‌ಗಳ (ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು ಅಥವಾ ಇತರ ಯಾವುದೇ ಸಾಧನಗಳು) ಯಾವುದೇ ಗುಂಪನ್ನು (ಅಥವಾ ಸಿಸ್ಟಮ್) ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು ವಿಶೇಷವಾಗಿ ಮಾಹಿತಿಯ ವಿನಿಮಯಕ್ಕಾಗಿ ಮತ್ತು ಸಂಪನ್ಮೂಲಗಳ ಹಂಚಿಕೆಗಾಗಿ ಮಾಡಲಾಗುತ್ತದೆ. ಒಂದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಡಾಟಾವನ್ನು ಒಂದು ನೋಡ್‌ನಿಂದ ಮತ್ತು ನೋಡ್‌ಗೆ ನಿರ್ದಿಷ್ಟ ಮತ್ತು ಅಪೇಕ್ಷಿತ ಹರಿವಿನ ದರಗಳಲ್ಲಿ ಸುರಕ್ಷಿತವಾಗಿ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿ ಕಳುಹಿಸುವ ಮತ್ತು ಸ್ವೀಕರಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರುವ ಯಾವುದೇ ಸಾಧನಕ್ಕೆ ನೋಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಕ್ರಿಯಾತ್ಮಕತೆ ಮತ್ತು ಬಳಕೆಯ ಆಧಾರದ ಮೇಲೆ ನೋಡ್ ಸಾಧನಗಳನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಸ್ಥೂಲವಾಗಿ ವರ್ಗೀಕರಿಸಬಹುದು -

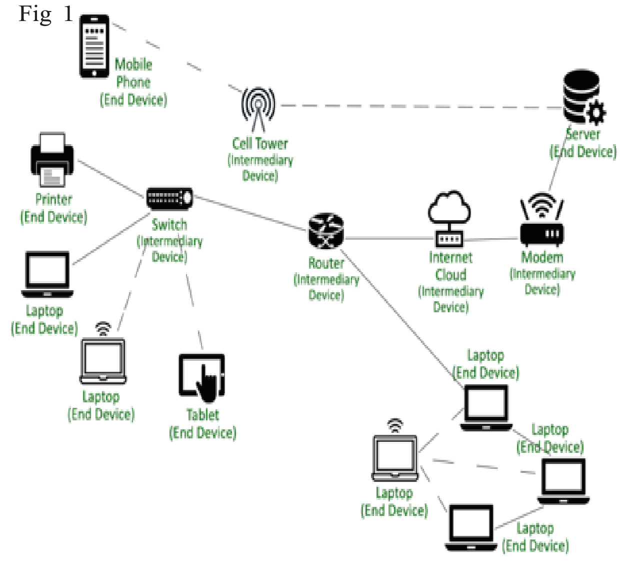
1 ಅಂತಿಮ ಸಾಧನಗಳು

2 ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು

ಅಂತಿಮ ಸಾಧನಗಳು (End Devices)

ಅಂತಿಮ ಸಾಧನಗಳು ಎಂದರೆ ಒಂದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಸಂಭವಿಸುವ ಸಂವಹನದಲ್ಲಿ ಸೋರ್ಸ್ ಪಾಯಿಂಟ್ ಅಥವಾ ಡೆಸ್ಟಿನೇಷನ್ ಪಾಯಿಂಟ್ ಆಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ನೋಡ್ ಸಾಧನಗಳಾಗಿವೆ. ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ನಾವು ಕ್ಲೈಂಟ್, ಸರ್ವರ್ ಅಥವಾ ಎರಡೂ ಆಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ನೋಡ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದೇವೆ. ಈ ಅಂತಿಮ ಸಾಧನಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಅವುಗಳ ನಡುವೆ ಸಂವಹನ ಲಿಂಕ್‌ನ್ನು ಸ್ಥಾಪಿಸಲು ಉಳಿದ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ. ನೋಡ್ ಸಾಧನಗಳಲ್ಲಿ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಲಾದ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು ಒಂದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಅವು ನಿರ್ವಹಿಸುವ ಪಾತ್ರವನ್ನು ನಿರ್ಧರಿಸುತ್ತವೆ. ಅವುಗಳ ಬಳಕೆಯಂತೆ ಅಂತಿಮ ಸಾಧನಗಳನ್ನು ಈ ಕೆಳಗಿನಂತೆ ಸ್ಥೂಲವಾಗಿ ವರ್ಗೀಕರಿಸಬಹುದು. (Fig 1)

Fig 1



- **ಅಂತಿಮ ಸಾಧನಗಳು ಕ್ಲೈಂಟ್‌ಗಳಂತೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು** (End Devices as Clients): ಸಾಮಾನ್ಯವಾಗಿ ಅಂತಿಮ ಕ್ಲೈಂಟ್‌ಗಳ ಬಳಕೆಗಾಗಿ ಡಾಟಾವನ್ನು ವಿನಂತಿಸುವುದು, ಸ್ವೀಕರಿಸಿದ ಡಾಟಾವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು ಇತ್ಯಾದಿ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

- **ಅಂತಿಮ ಸಾಧನಗಳು ಸರ್ವರ್‌ಗಳಂತೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು** (End Devices as Servers) : ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಇತರ ನೋಡ್‌ಗಳಿಗೆ (ಅಥವಾ ಹೋಸ್ಟ್‌ಗಳು) ಮಾಹಿತಿ ಮತ್ತು ಸೇವೆಗಳನ್ನು ಒದಗಿಸುವ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಅಂತಿಮ ಸಾಧನಗಳ ಕಾರ್ಯಗಳು (Functions of End Devices)

1 ನೆಟ್‌ವರ್ಕ್ ಮೂಲಕ ಪ್ರಸರಣವಾಗುವ ಡಾಟಾ ಮತ್ತು ಮಾಹಿತಿಯ ಮೂಲವಾಗಿ ಅಂತಿಮ ಸಾಧನಗಳು ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ.

2 ಇವು ಅಂತಿಮ ಬಳಕೆದಾರರು(ಮಾನವರು) ಮತ್ತು ಹಲವಾರು ನೋಡ್ ಸಾಧನಗಳನ್ನು ಹೊಂದಿರುವ ಸಂವಹನ ನೆಟ್‌ವರ್ಕ್ ನಡುವಿನ ಮಾಧ್ಯಮವಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ.

ಅಂತಿಮ ಸಾಧನಗಳ ಉದಾಹರಣೆಗಳು (Examples of the End Devices)

ವರ್ಕ್ ಸ್ಟೇಷನ್‌ಗಳು, ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳು, ಡೆಸ್ಕ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು ಮತ್ತು ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಸರ್ವರ್‌ಗಳು (ಫೈಲ್ ಸರ್ವರ್‌ಗಳು, ವೆಬ್ ಸರ್ವರ್‌ಗಳು) ಮೊಬೈಲ್ ಫೋನ್‌ಗಳು, ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು, ಸ್ಮಾರ್ಟ್ ಫೋನ್‌ಗಳು ಇತ್ಯಾದಿಗಳು.

ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು (Intermediary Devices)

ಒಂದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಡಾಟಾವನ್ನು ಒಂದು ಬದಿಯಿಂದ ಇನ್ನೊಂದು ಬದಿಗೆ ಪ್ರಸರಣ ಮಾಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾದ ನೋಡ್ ಸಾಧನಗಳಿಗೆ ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಈ ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು ಇತರ ನೊಡ್‌ಗಳಿಗೆ ಸಂಪರ್ಕ ಮಾಧ್ಯಮದಂತೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ. (ಇತರ ಸೇವೆಗಳೊಂದಿಗೆ) ಡಾಟಾ ಅಪೇಕ್ಷಿತ ಹರಿವಿನ ದರದೊಂದಿಗೆ ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಸಂಪೂರ್ಣ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಪ್ರಸರಣವಾಗುತ್ತಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುವ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳು ಅವುಗಳಲ್ಲಿ ಪ್ರಸರಣವಾಗುತ್ತಿರುವ ಡಾಟಾದ ನಿರ್ವಹಣೆಗಾಗಿ ವಿವಿಧ ಅಡ್ರೆಸಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಬಳಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ IP ಅಡ್ರೆಸ್, MAC ಅಡ್ರೆಸ್, ಮತ್ತು ಪೋರ್ಟ್ ನಂಬರ್‌ಗಳು (ಪೋರ್ಟ್ ಅಡ್ರೆಸ್) ಹಾಗೂ ಜೊತೆಗೆ ನೆಟ್‌ವರ್ಕ್ ಪರಸ್ಪರ ಸಂಪರ್ಕಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಬಳಸುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿಯ ವಿವಿಧ ರೀತಿಯ ಸ್ವಿಚಿಂಗ್‌ಗಳು, ನೆಟ್‌ವರ್ಕ್ ಮೂಲಕ ನಡೆಯುವ ಸಂವಹನದಲ್ಲಿ ಸಂದೇಶಗಳು ತೆಗೆದುಕೊಳ್ಳುವ ಮಾರ್ಗವನ್ನು ನಿರ್ಧರಿಸುತ್ತವೆ.

ಮಧ್ಯವರ್ತಿ ಸಾಧನದ ಕಾರ್ಯಗಳು (Functions of Intermediary Devices)

- ಸಿಗ್ನಲ್ ಡ್ಯಾಂಪಿಂಗ್ ಒಂದು ಸಾಮಾನ್ಯ ವಿದ್ಯಮಾನವಾಗಿದೆ. ಈ ಸಾಧನಗಳು ಡಾಟಾ ಸಿಗ್ನಲ್‌ಗಳ ಪುನರುತ್ಪಾದನೆ ಮತ್ತು ಮರುಪ್ರಸಾರದ ಮೂಲಕ ಇದರಿಂದ ಹೊರಬರುವಂತೆ ಮಾಡುತ್ತವೆ.
- ಯಶಸ್ವಿ ವರ್ಗಾವಣೆಗಳನ್ನು ಮಾಡಲು ಈ ಸಾಧನಗಳು ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಸೋರ್ಸ್ ಅಡ್ರೆಸ್, ಡೆಸ್ಟಿನೇಷನ್ ಅಡ್ರೆಸ್ ಅಥವಾ ವಿಭಿನ್ನ ಪಾಥ್‌ವೇಗಳ ಮಾಹಿತಿಯ ದಾಖಲೆಯನ್ನು ಇರಿಸಿಕೊಳ್ಳುತ್ತವೆ. ಇದು ಕೆಲಸ ನಿರ್ವಹಿಸುವ ಸ್ವಿಚಿಂಗ ತಂತ್ರವನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ.
- ಇವು ರಿಡಂಡನ್ಸಿ ಬಿಟ್‌ಗಳನ್ನು (redundancy bits) ಬಳಸಿಕೊಂಡು ದೋಷಗಳನ್ನು ಮತ್ತು ತಪ್ಪುಗಳನ್ನು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಕಂಡುಹಿಡಿಯುತ್ತವೆ ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಡಾಟಾವನ್ನು ವರ್ಗಾಯಿಸುವಾಗ ತಿದ್ದುಪಡಿಗಳನ್ನು ಮಾಡುವ ಮೂಲಕ ದೋಷ ಸಹಿಷ್ಣುತೆಯನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಸಾಧನಗಳಿಗೆ ಸೂಚನೆಯನ್ನು ನೀಡುತ್ತವೆ.
- ಸಂಪೂರ್ಣ ಸಂವಹನ ಅವಧಿಯಲ್ಲಿ ನಿರ್ದಿಷ್ಟ ಹರಿವಿನ ನಿಯಂತ್ರಣ ಮತ್ತು ಪ್ರತಿಕ್ರಿಯೆಯ ಕಾಲಾವಧಿಯನ್ನು ನಿರ್ವಹಿಸುತ್ತವೆ.
- ಆದ್ಯತೆಗಳ ಆಧಾರದ ಮೇಲೆ ಸೆಟಪ್, ಗ್ರೂಪಿಂಗ್ ಮತ್ತು ಸಂದೇಶಗಳ ನಿರ್ದೇಶನ (ಅಥವಾ ಪ್ಯಾಕೆಟ್‌ಗಳು)ಗಳು ಈ ಸಾಧನಗಳಿಂದ ಮಾಡಲ್ಪಡುತ್ತವೆ.
- ಭದ್ರತಾ ಸೆಟ್ಟಿಂಗ್‌ಗಳ ಆಧಾರದ ಮೇಲೆ ಡಾಟಾದ ಪ್ರಸರಣವನ್ನು ಅನುಮತಿಸುವ ಅಥವಾ ನಿರಾಕರಿಸುವ ಮೂಲಕ ನೆಟ್‌ವರ್ಕ್ ಭದ್ರತೆಯನ್ನು ಒದಗಿಸುತ್ತವೆ.

ಮಧ್ಯವರ್ತಿ ಸಾಧನಗಳ ಉದಾಹರಣೆಗಳು (Examples of the intermediary devices)

ಹಬ್, ಸ್ವಿಚ್, ವೈರ್‌ಲೆಸ ಎಕ್ಸೆಸ್ ಪಾಯಿಂಟ್‌ಗಳು, ಮತ್ತು ಇತರ ಸಾಧನಗಳು. ಇವುಗಳನ್ನು - ನೆಟ್‌ವರ್ಕ್, ಫೈಲ್ ಸರ್ವರ, ವೆಬ್ ಸರ್ವರ್, ಪ್ರಿಂಟ್ ಸರ್ವರ್, ಮಾಡೆಮ್ ಮುಂತಾದವುಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇಂಟರ್‌ನೆಟ್‌ವರ್ಕಿಂಗ್‌ಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುವ ಸಾಧನಗಳಾದ - ರೂಟರ್‌ಗಳು, ಬ್ರಿಜ್‌ಗಳು, ರಿಪೀಟರ್‌ಗಳು ಮತ್ತು ಸೆಕ್ಯುರಿಟಿ ಫೈರ್‌ವಾಲ್‌ಗಳು ಇತ್ಯಾದಿಗಳು.

ಲೋಕಲ್ ನೆಟ್‌ವರ್ಕ್ (Local network)

ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಅವುಗಳ ಗಾತ್ರ, ಸಂಕೀರ್ಣತೆ, ಭದ್ರತಾ ಮಟ್ಟ, ಅಥವಾ ಭೌಗೋಳಿಕ ವ್ಯಾಪ್ತಿಗಳ ಆಧಾರದ ಮೇಲೆ

© NIMI NOT TO BE REPUBLISHED

ವರ್ಗೀಕರಿಸಬಹುದು. ಭೌಗೋಳಿಕ ಹರಡುವಿಕೆಯ ಆಧರಿಸಿ ನಾವು ಕೆಲವು ಜನಪ್ರಿಯ ಟೊಪೊಲಜಿಗಳನ್ನು ಚರ್ಚಿಸೋಣ.

ಪ್ಯಾನ್ (PAN) (Fig 2)

PAN ಎನ್ನುವುದು ಪರ್ಸನಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್‌ನ (Personal Area Network) ಸಂಕ್ಷಿಪ್ತ ರೂಪವಾಗಿದೆ. PAN ಇದು ವ್ಯಕ್ತಿಯ ಖಾಸಗಿ ಸ್ಥಳದ ವ್ಯಾಪ್ತಿಯಲ್ಲಿರುವ ಸಾಧನಗಳ ನಡುವಿನ ಪರಸ್ಪರ ಸಂಪರ್ಕವಾಗಿದೆ. ಸಾಮಾನ್ಯವಾಗಿ 10 ಮೀಟರ್‌ಗಳ ವ್ಯಾಪ್ತಿಯಲಿ ್ಲರುವ ನೆಟ್‌ವರ್ಕ್ ಆಗಿದೆ. ನೀವು ನಿಮ್ಮ ಲ್ಯಾಪ್‌ಟಾಪ್‌ದಿಂದ ಮೊಬೈಲ್‌ಗೆ ಅಥವಾ ನಿಮ್ಮ ಮೊಬೈಲ್‌ನಿಂದ ನಿಮ್ಮ ಸ್ನೇಹಿತರ ಮೊಬೈಲ್‌ಗೆ ಬ್ಲೂಟೂಥ್ ಉಪಯೋಗಿಸಿ ಚಿತ್ರಗಳನ್ನು ಅಥವಾ ಹಾಡುಗಳನ್ನು ವರ್ಗಾಯಿಸಿದ್ದರೆ, ನೀವು Personal Area Network ನ್ನು ಸೆಟ್ ಮಾಡಿದ್ದೀರಿ ಮತ್ತು ಉಪಯೋಗಿಸಿದ್ದೀರಿ ಎಂದು ಅರ್ಥ.

Fig 2

ಒಬ್ಬ ವ್ಯಕ್ತಿಯು ತನ್ನ ಮನೆಯಲ್ಲಿ ತನ್ನ ಲ್ಯಾಪ್‌ಟಾಪ್, ಸ್ಮಾರ್ಟ್ ಫೋನ್, ಪರ್ಸನಲ್ ಡಿಜಿಟಲ್ ಅಸಿಸ್ಟಂಟ್ ಮತ್ತು ಪೋರ್ಟೇಬಲ್ ಪ್ರಿಂಟರ್‌ನ್ನು ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಬಹುದು. ಈ ನೆಟ್‌ವರ್ಕ್ ಸಂಪೂರ್ಣವಾಗಿ ವೈ - ಫೈ ಆಗಿರಬಹುದು ಅಥವಾ ವೈರ್ ಸಹಿತ ಮತ್ತು ವೈರ್‌ರಹಿತ ಇವುಗಳ ಸಂಯೋಜನೆಯಾಗಿರಬಹುದು.

ಲ್ಯಾನ್ (LAN) (Fig 3)

ಲ್ಯಾನ್ ಅಥವಾ ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್ (Local Area Network-LAN) ಇದು ವೈಯರ್ಡ್ ನೆಟ್‌ವರ್ಕ್ ಆಗಿದ್ದು, ಕಚೇರಿ, ಕಟ್ಟಡ ಅಥವಾ ಉತ್ಪಾದನಾ ಘಟಕದಂತಹ ಒಂದೇ ಪ್ರದೇಶದಲ್ಲಿ ಹರಡಿರುತ್ತದೆ. LAN ಉಪಯೋಗಿಸಿ ಒಂದು ತಂಡದ ಸದಸ್ಯರು ಸಾಫ್ಟ್‌ವೇರ ಮತ್ತು ಹಾರ್ಡ್‌ವೇರ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಪರಸ್ಪರ ಹಂಚಿಕೊಳ್ಳಬಹುದು. ಆದರೆ ಹೊರಗಿನ ಪ್ರಪಂಚದೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳಲಾಗುವದಿಲ್ಲ. ಸಾಫ್ಟವೇರ್ ಸಂಪನ್ಮೂಲಗಳು ಎಂದರೆ ಅಧಿಕೃತ ದಾಖಲೆಗಳು, ಬಳಕೆದಾರ ಕೈಪಿಡಿಗಳು, ಉದ್ಯೋಗಿ ಕೈಪಿಡಿ ಇತ್ಯಾದಿಗಳು. ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಹಂಚಿಕೊಳ್ಳಬಹುದಾದ ಹಾರ್ಡ್‌ವೇರ್ ಸಂಪನ್ಮೂಲಗಳು ಎಂದರೆ ಪ್ರಿಂಟರ್, ಫ್ಯಾಕ್ಸ್ ಮಶೀನ್‌ಗಳು, ಮಾಡೆಮ್‌ಗಳು, ಮೆಮೊರಿ ಸ್ಥಳ, ಇತ್ಯಾದಿಗಳು. ಇದು ಸಂಸ್ಥೆಯ ಮೂಲಸೌಕರ್ಯದ ವೆಚ್ಚವನ್ನು ಅತ್ಯಂತ ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

Fig 3

ವಯರ್‌ಸಹಿತ ಅಥವಾ ವಯರ್‌ರಹಿತ ಸಂಪರ್ಕಗಳನ್ನು ಉಪಯೋಗಿಸಿ LAN ಅನ್ನು ಹೊಂದಿಸಬಹುದು. ಸಂಪೂರ್ಣವಾಗಿ ವಯರ್‌ರಹಿತ LAN ಅನ್ನು ವಯರ್‌ಲೆಸ್ ಲ್ಯಾನ್ ಅಥವಾ WLAN ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಮ್ಯಾನ್ (MAN) (Fig 4)

ಮ್ಯಾನ್ ಇದು ಮೆಟ್ರೊಪಾಲಿಟನ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್‌ನ (Metropolitan Area Network- MAN) ಸಂಕ್ಷಿಪ್ತ ರೂಪವಾಗಿದೆ. ಇದು ಒಂದು ನಗರ, ಕಾಲೇಜು ಕ್ಯಾಂಪಸ್, ಅಥವಾ ಸಣ್ಣ ಪ್ರದೇಶದಲ್ಲಿ ಹರಡಿರುವ ನೆಟ್‌ವರ್ಕ್ ಆಗಿದೆ. MAN ಇದು LAN ಗಿಂತಲೂ ದೊಡ್ಡದಾಗಿದೆ ಮತ್ತು ಸಾಮಾನ್ಯವಾಗಿ ಹಲವಾರು ಕಿಲೊಮೀಟರ್‌ಗಳಲ್ಲಿ ಹರಡಿರುತ್ತದೆ. MAN ನ ಮುಖ್ಯ ಉದ್ದೇಶವು ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳುವುದಾಗಿದೆ.

Fig 4

© NIMI NOT TO BE REPUBLISHED

ಇದರಿಂದ ಮೂಲಸೌಕರ್ಯದ ವೆಚ್ಚಗಳು ಕಡಿಮೆಯಾಗುತ್ತವೆ. ಹಲವಾರು LAN ಗಳನ್ನು ಸಂಪರ್ಕಿಸುವ ಮೂಲಕ MAN ಅನ್ನು ನಿರ್ಮಿಸಬಹುದು.

MAN ನ ಅತ್ಯಂತ ಸಾಮಾನ್ಯ ಉದಾಹರಣೆಯೆಂದರೆ ಕೇಬಲ್ TV ನೆಟ್‌ವರ್ಕ್.

ವ್ಯಾನ್ (WAN) (Fig 5)

ವ್ಯಾನ್ ಅಥವಾ ವೈಡ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್(Wide Area Network - WAN) ಇದು ದೇಶದಾದ್ಯಂತ ಅಥವಾ ಅನೇಕ ದೇಶಗಳಲ್ಲಿ ಹರಡಿದೆ. WAN ಸಾಮಾನ್ಯವಾಗಿ ಹಲವಾರು LAN, MAN ಮತ್ತು WAN ಗಳ ನೆಟ್‌ವರ್ಕ್ ಆಗಿದೆ. ಈ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಲಭ್ಯತೆ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹತೆಯ ಆಧಾರದ ಮೇಲೆ ವಯರ್‌ಸಹಿತ ಅಥವಾ ವಯರ್‌ರಹಿತ ಸಂಪರ್ಕಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಸೆಟ್ ಮಾಡಲಾಗುತ್ತದೆ.

Fig 5

WAN ನ ಅತ್ಯಂತ ಸಾಮಾನ್ಯ ಉದಾಹರಣೆಯೆಂದರೆ ಇಂಟರ್‌ನೆಟ್.

ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನ (Network Communication)

- ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನದ ಮೂಲಗಳು
- ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- ವೈಫೈ ಅನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- LAN ನ ಪರಿಚಯ

ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನದ ಮೂಲಗಳು (Basics of network communication)

ಈ ವಿಭಾಗವು ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನದ ಬಗ್ಗೆ ತಿಳಿಸುತ್ತದೆ ಮತ್ತು ಮಾಹಿತಿಯ ಬಗ್ಗೆ ಸಂಬಂಧಿಸಿದ ಕೆಲವು ಪದಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. z/OS ನೆಟ್‌ವರ್ಕ್ ಕಮ್ಯುನಿಕೇಷನ್ ಮತ್ತು TCP/IP ಸಾಕೆಟ್‌ಗಳ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು, z/OS Communications Server : IP Configuration Guide ಮತ್ತು z/OS Communications Server: IP Programmers Guide ಮತ್ತು Reference ನ್ನು ನೋಡಿರಿ. IPv6 ನೆಟ್‌ವರ್ಕ್ ಕಮ್ಯೂನಿಕೇಷನ್ ಮತ್ತು AF_INET6 ಸಾಕೆಟ್‌ಗಳ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು, z/OS

Communications Server: IPv6 Network and Appl Design Guide ನ್ನು ನೋಡಿರಿ. ನೆಟ್‌ವರ್ಕ್ ಸಂವಹನ ಅಥವಾ ಇಂಟರ್‌ನೆಟ್‌ವರ್ಕಿಂಗ್, ಒಂದು ಪ್ರೊಟೊಕಾಲ್‌ಗಳ ಸೆಟ್‌ನ್ನು (ನಿಯಮಗಳು ಮತ್ತು ಮಾನದಂಡಗಳು) ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ಇವು ಅಪ್ಲಿಕೇಷನ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಅವು ಯಾವ ಹಾರ್ಡ್‌ವೇರ್ ಅಥವಾ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ರನ್ ಆಗುತ್ತಿವೆ ಎಂಬುದನ್ನು ಲೆಕ್ಕಿಸದೇ ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸಲು ಅನುಮತಿಸುತ್ತವೆ. ಇಂಟರ್‌ನೆಟ್‌ವರ್ಕಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಅವುಗಳ ಭೌತಿಕ ನೆಟ್‌ವರ್ಕ್ ಸಂಪರ್ಕಗಳಿಂದ ಸ್ವತಂತ್ರವಾಗಿ ಸಂವಹನ ನಡೆಸಲು ಅನುಮತಿಸುತ್ತದೆ.

TCP/IP ಎನ್ನುವ ಇಂಟರ್‌ನೆಟ್‌ವರ್ಕಿಂಗ್ ತಂತ್ರಾಜ್ಞಾನವನ್ನು ಅದರ ಎರಡು ಮುಖ್ಯ ಪ್ರೊಟೋಕಾಲ್‌ಗಳಿಂದ ಹೆಸರಿಸಲಾಗಿದೆ. ಟ್ರಾನ್ಸ್‌ಮಿಷನ್ ಕಂಟ್ರೋಲ್ ಪ್ರೊಟೊಕಾಲ್ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್ ಪ್ರೊಟೊಕಾಲ್ (Transmission Control Protocol(TCP) and Internet Protocol (IP)). TCP/IP ಯನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳಲು ನೀವು ಈ ಕೆಳಗಿನ ಶಬ್ದಗಳೊಂದಿಗೆ ಪರಿಚಿತರಾಗಿರಬೇಕು :

ಕ್ಲೈಂಟ್ (Client)

ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಸೇವೆಗಳಿಗಾಗಿ ವಿನಂತಿಸುವ ಪ್ರಕ್ರಿಯೆ.

ಸರ್ವರ್ (Server)

ಕ್ಲೈಂಟ್‌ನಿಂದ ಸೇವೆಯ ವಿನಂತಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುವ ಪ್ರಕ್ರಿಯೆ.

ಡಾಟಾಗ್ರಾಮ್ (Datagram)

ಇದು ಮಾಹಿತಿಯ ಮೂಲ ಘಟಕವಾಗಿದೆ. ಇದು ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಡಾಟಾ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಡಾಟಾ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ಟ್ರಾನ್ಸ್‌ಪೋರ್ಟ್ ಹಂತದಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ರವಾನಿಸಲಾಗುತ್ತದೆ.

ಪ್ಯಾಕೆಟ್ (Packet)

ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಅದರ ನೆಟ್‌ವರ್ಕ್ ನಡುವಿನ ಡಾಟಾದ ವಹಿವಾಟಿನ ಘಟಕ ಅಥವಾ ಬ್ಲಾಕ್‌ನ್ನು ಪ್ಯಾಕೆಟ್ ಎನ್ನುತ್ತಾರೆ. ಪ್ಯಾಕೆಟ್ ಸಾಮಾನ್ಯವಾಗಿ ನೆಟ್‌ವರ್ಕ್ ಹೆಡರ್‌ನ್ನು - ಕನಿಷ್ಟ ಒಂದು ಉನ್ನತ

© NIMI NOT TO BE REPUBLISHED

ಮಟ್ಟದ ಪ್ರೊಟೋಕಾಲ್ ಹೆಡರ್‌ನ್ನು ಮತ್ತು ಡಾಟಾ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ, ಡಾಟಾ ಬ್ಲಾಕ್‌ಗಳ ಸ್ವರೂಪವು ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ಹೇಗೆ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ ಎಂಬುದರ ಮೇಲೆ ಪರಿಣಾಮವನ್ನು ಬೀರುವುದಿಲ್ಲ. ಪ್ಯಾಕೆಟ್‌ಗಳು ಡಾಟಾವನ್ನು ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಕಳುಹಿಸಲು ಇಂಟರ್‌ನೆಟ್‌ವರ್ಕ್ ಲೇಯರ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ವಿನಿಮಯ ಮಾಧ್ಯಮಗಳಾಗಿವೆ.

ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿರುವ ಸಾಧನಗಳು ಪರಸ್ಪರ ಹೇಗೆ ಸಂವಹನ ನಡೆಸುತ್ತವೆ ?

ಈಥರ್‌ನೆಟ್ ಒಂದು ಪ್ರಮಾಣಿತ ಪ್ರೊಟೋಕಾಲ್ ಆಗಿದ್ದು, ಅದು ಅನೇಕ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಈ ಮಶೀನ್‌ಗಳನ್ನು ಈಥರ್‌ನೆಟ್ ಕೇಬಲ್ ಮೂಲಕ ಅಥವಾ "Category 5" ವೈರಿಂಗ ಮೂಲಕ ಒಂದಕ್ಕೊಂದು ಜೋಡಿಸುವ ಅಗತ್ಯವಿರುತ್ತದೆ.

ಹೋಮ್ ನೆಟ್‌ವರ್ಕಿಂಗ್ ಎಂದರೇನು ?(What is Home Networking?)

ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್ ಎಂದರೆ ಸಾಧನಗಳ ಗುಂಪು - ಉದಾಹರಣೆಗೆ ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಆಟದ ವ್ಯವಸ್ಥೆಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು ಮತ್ತು ಮೊಬೈಲ್ ಸಾಧನಗಳು - ಇವು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಮತ್ತು ಪರಸ್ಪರ ಸಂಪರ್ಕವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಎರಡು ರೀತಿಯಲ್ಲಿ ಜೋಡಿಸಬಹುದು :

- **ವಯರ್ಡ್ ನೆಟ್‌ವರ್ಕ್** - ಇದು ಪ್ರಿಂಟರ್ ಮತ್ತು ಸ್ಕ್ಯಾನರ್‌ನಂತಹ ಸಾಧನಗಳನ್ನು ಕೇಬಲ್‌ಗಳ ಮೂಲಕ ಸಂಪರ್ಕಿಸುತ್ತದೆ.
- **ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್** - ಇದು ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು ಮತ್ತು ಇ - ರೀಡರ್‌ಗಳಂತಹ ಸಾಧನಗಳನ್ನು ಕೇಬಲ್‌ಗಳಿಲ್ಲದೆಯೆ ಜೋಡಿಸುತ್ತದೆ.

ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಹೇಗೆ ಸೆಟ್ ಮಡುವುದು? (Why Set Up a Home Network?)

ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಸ್ಥಾಪಿಸಲು ಹಲವಾರು ಕಾರಣಗಳಿವೆ. ಹೋಮ್ ನೆಟ್‌ವರ್ಕಿಂಗ್ ನಿಮಗೆ ಅನುಮತಿಸುವ ಕೆಲವು ವಿಷಯಗಳನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ :

- ಅನೇಕ ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಆಟದ ವ್ಯವಸ್ಥೆಗಳು, ಮೊಬೈಲ್ ಸಾಧನಗಳು ಮುಂತಾದವುಗಳನ್ನು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುವುದು.
- ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಲ್ಪಟ್ಟ ಎಲ್ಲ ಸಾಧನಗಳಲ್ಲಿರುವ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಫೋಲ್ಡರ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.
- ಅನೇಕ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಂದ ಒಂದೇ ಪ್ರಿಂಟರ್‌ನಲ್ಲಿ ಪ್ರಿಂಟ್ ಮಾಡಬಹುದು.
- ಒಂದೇ ಸ್ಥಳದಲ್ಲಿ ಎಲ್ಲ ನೆಟ್‌ವರ್ಕ್ ಮಾಡಲಾದ ಸಾಧನಗಳ ಭದ್ರತಾ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು.

ನೀವು ಹೋಮ್ ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನ್ನು ಪ್ರಯತ್ನಿಸಲು ಸಿದ್ಧರಿದ್ದರೆ, ನೀವು ಅದನ್ನು ಹೇಗೆ ಪ್ರಾರಂಭಿಸಬಹುದು ಮತ್ತು ನಿಮ್ಮ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಲಭ್ಯವಿರುವ ಆಯ್ಕೆಗಳನ್ನು ನೋಡಲು :

ಹೋಮ್ ನೆಟ್‌ವರ್ಕನ್ನು ಸೆಟ್ ಮಾಡಲು ಏನು ಬೇಕು? (What You Need to Set Up a Home Network)

ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಸೆಟ್ ಮಾಡಲು, ನಿಮಗೆ ಕೆಳಗಿನವುಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ :

- Xfinity Internet Service subscription (ಅಥವಾ ಮತ್ತೊಂದು ಇಂಟರ್‌ನೆಟ್ ಪ್ರೊವೈಡರ್‌ಗೆ ಚಂದಾದಾರಿಕೆ)
- ಮಾಡೆಮ್, ಇದು ಇಂಟರ್‌ನೆಟ್‌ಗೆ, ಮತ್ತು ರೂಟರ್‌ಗೆ ಸಂಪರ್ಕಿಸುತ್ತದೆ. ಇದು ಮಾಡೆಮ್‌ನ ಮೂಲಕ ನಿಮ್ಮ ಸಾಧನಗಳನ್ನು ಪರಸ್ಪರ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುತ್ತದೆ (ಅಥವಾ ಗೇಟ್‌ವೇ, ಇದು ಮಾಡೆಮ್ ಮತ್ತು ರೂಟರ್ ಎರಡೂ ರೀತಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ).
- ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಲು ಕಂಪ್ಯೂಟರ್ ಅಥವಾ ಇತರ ಸಾಧನಗಳು.

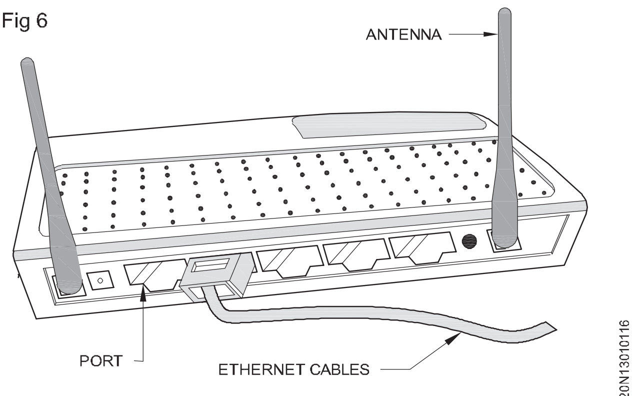
ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ Activate a Wireless Gateway ಯನ್ನು ನೋಡಿರಿ.

The Wireless Gateway 1 (model numbers TG852G, TG862G, SMCD3GNV, TC8305C) ಮತ್ತು Wireless Gateway 2 (model number DPC3939) ಇವು ಆಲ್-ಇನ್- ಒನ್ ಮಾಡೆಮ್, ರೂಟರ್, ಮತ್ತು ಫೋನ್ ಸಾಧನಗಳಂತೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ. ಇವು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಒಂದು ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಲಭ್ಯವಿರುವ ಅತ್ಯುತ್ತಮ ಭದ್ರತಾ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಬಳಕೆದಾರರಿಗೆ ಒದಗಿಸುತ್ತವೆ. ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ See What is a Wireless Gateway? ನ್ನು ನೋಡಿರಿ.

ವಯರ್‌ಲೆಸ್ ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್ (Wireless Home Network): ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್ ಸಾಮಾನ್ಯವಾಗಿ ವೈ - ಫೈ ಎಂತಲೂ ಕರೆಯಲಾಗುವ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ, ಸಾಧನಗಳನ್ನು ಪರಸ್ಪರ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಕೇಬಲ್‌ಗಳನ್ನು ಬಳಸದೇ ಸಂಪರ್ಕಿಸಲಾಗುತ್ತದೆ. ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ What Is WiFi and How Do I Get It? ನ್ನು ನೋಡಿರಿ.

ವಯರ್ಡ್ ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್ (Wired Home Network) (Fig 6) : ವಯರ್ಡ್ ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಸಾಧನಗಳನ್ನು ಪರಸ್ಪರ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಈಥರ್‌ನೆಟ್ ಕೇಬಲ್‌ಗಳನ್ನು

Fig 6



ETHERNET & PORT

ಬಳಸಿ ಸಂಪರ್ಕಿಸಲಾಗುತ್ತದೆ.

ವಯರ್ಡ್ ಹೋಮ್ ನೆಟ್‌ವರ್ಕನ್ನು ಹೊಂದುವುದರಿಂದ ಹಲವಾರು ಪ್ರಯೋಜನಗಳಿವೆ :

- ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕವು ಹೆಚ್ಚು ವೇಗವಾಗಿ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿರುತ್ತದೆ.
- ಹೆಚ್ಚಿದ ಭದ್ರತೆ, ಏಕೆಂದರೆ ಯಾವುದೆ ಹೊರಗಿನ ಬಳಕೆದಾರರು ನಿಮ್ಮ ನೆಟ್‌ವರ್ಕನ್ನು ಪ್ರವೇಶಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.
- ವಯರ್‌ಲೆಸ್ ಸಂಪರ್ಕಗಳಿಗೆ ಹೋಲಿಸಿದರೆ, ಸೆಟಪ್ ಮತ್ತು ಸಮಸ್ಯೆ ನಿವಾರಣೆ ಸುಲಭವಾಗಿರುತ್ತದೆ.

ಮಿಶ್ರ ಹೋಮ್ ನೆಟ್‌ವರ್ಕ್ (Mixed Home Network): ಅನೇಕ ಜನರು ವಯರ್‌ಲೆಸ್ ಮತ್ತು ವಯರ್ಡ್ ನೆಟ್‌ವರ್ಕ್ ಎರಡರ ಮಿಶ್ರಣವು ತಮ್ಮ ಅಗತ್ಯಗಳನ್ನು ಉತ್ತಮವಾಗಿ ಪೂರೈಸುತ್ತದೆ ಎಂದು ಕಂಡುಕೊಳ್ಳುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ, ಚಲನಚಿತ್ರಗಳನ್ನು ಸ್ಟ್ರೀಮ್ ಮಾಡುವ ಸಾಧನಗಳು ವೇಗವಾದ ಮತ್ತು ಹೆಚ್ಚು ಸ್ಥಿರವಾದ ವಯರ್ಡ್ ಸಂಪರ್ಕದಿಂದ ಪ್ರಯೋಜನ ಪಡೆಯುತ್ತವೆ. ಲ್ಯಾಪ್‌ಟಾಪ್ ಅಥವಾ ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳಂತಹ ಸಾಧನಗಳು ವಯರ್‌ಲೆಸ್ ಸಂಪರ್ಕದಲ್ಲಿ ಲಭ್ಯವಿರುವ ಚಲನಶೀಲತೆಯಿಂದ ಪ್ರಯೋಜನ ಪಡೆಯುತ್ತವೆ.

ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 1 ಮತ್ತು ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 2, ಎರಡೂ ವಯರ್‌ಲೆಸ್ ಸಾಮರ್ಥ್ಯವನ್ನು ಮತ್ತು ನಾಲ್ಕು ಈಥರ್‌ನೆಟ್ ಪೋರ್ಟ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇದು ನಿಮಗೆ ಸಾಧನಗಳನ್ನು ಕೇಬಲ್‌ಗಳೊಂದಿಗೆ ಮತ್ತು ಕೇಬಲ್‌ಗಳಿಲ್ಲದೆಯೆ, ಎರಡೂ ರೀತಿಯಿಂದ ಸಂಪರ್ಕಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

ಹೋಮ್ ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನ ಶಬ್ದ ಸಂಗ್ರಹ (Home Networking Glossary)

ಹೋಮ್ ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನಲ್ಲಿ ಬರುವ ಸಾಮಾನ್ಯ ಶಬ್ದಗಳ ಪಟ್ಟಿಯನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ.

ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್ (Wireless network) - ಕೇಬಲ್‌ಗಳಿಲ್ಲದೆಯೆ ಪರಸ್ಪರ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುವ ಸಾಧನಗಳ ಗುಂಪನ್ನು ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ವೈ- ಫೈ ಎಂತಲೂ ಕರೆಯುತ್ತಾರೆ.

ಮಾಡೆಮ್ (Modem) - ಇಂಟರ್‌ನೆಟ್‌ಗೆ ರೂಟರ್‌ಗಳನ್ನು ಅಥವಾ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಕೇಬಲ್ ಮೂಲಕ ಸಂಪರ್ಕಿಸುವ ಸಾಧನಕ್ಕೆ ಮಾಡೆಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ರೂಟರ್ (Router) - ವಯರ್‌ಲೆಸ್ ಸಂಕೇತವನ್ನು ರವಾನಿಸುವ ಸಾಧನಕ್ಕೆ ರೂಟರ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಮಾಡೆಮ್ ಮೂಲಕ ಸಾಧನಗಳನ್ನು ಪರಸ್ಪರ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುತ್ತದೆ.

ಗೇಟ್‌ವೇ(Gateway) - ಮಾಡೆಮ್ ಮತ್ತು ರೂಟರ್ ಎರಡರಂತೆಯೂ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಸಾಧನಕ್ಕೆ ರೂಟರ್ ಎನ್ನುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 1 ಅಥವಾ ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 2 ನಂತೆ.

ಸಿಗ್ನಲ್ ರೇಂಜ್ (Signal range) - ವಯರ್‌ಲೆಸ್ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿದಸಾಧನವು ವಯರ್‌ಲೆಸ್‌ಸಂಕೇತವನ್ನುತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸುವ ಪ್ರದೇಶವನ್ನು ಸಿಗ್ನಲ್ ರೇಂಜ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ ಸೈಟ್ (Administration site) - ರೂಟರ್‌ನ ಮತ್ತು ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ನ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ನೋಡುವ ಮತ್ತು ಬದಲಾಯಿಸುವ ಆನ್‌ಲೈನ್ ಸೈಟನ್ನು, ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 1 ಮತ್ತು ವಯರ್‌ಲೆಸ್ ಗೇಟ್‌ವೇ 2 ಇವುಗಳ ಅಡ್ಮಿನ್ ಟೂಲ್ (Admin Tool) ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ರೂಟರ್ ಯೂಸರ್‌ನೇಮ್ ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್ (Router user-name and password) - ಇದು ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ ಸೈಟನ್ನು ಪ್ರವೇಶಿಸಲು ಬಳಸಲಾಗುವ ಲಾಗಿನ್ ಮಾಹಿತಿಯಾಗಿದೆ.

ನೆಟ್‌ವರ್ಕ್ ಹೆಸರು(Network Name) (SSID) - ಇದು ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ನ ಹೆಸರಾಗಿದೆ.

ನೆಟ್‌ವರ್ಕ್ ಪಾಸ್‌ವರ್ಡ್ (Network Password) (Key) - ಇದು ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಲು ಬಳಸುವ ಪಾಸ್‌ವರ್ಡ್ ಆಗಿದೆ.

ವೈ - ಫೈ ವ್ಯಾಖ್ಯಾನ (Wi-Fi Definition)

ವೈ - ಫೈ ಎನ್ನುವುದು ವಯರ್‌ಲೆಸ್ ತಂತ್ರಜ್ಞಾನವಾಗಿದ್ದು, ಇದನ್ನು ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು, ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ಗಳು. ಮತ್ತು ಇತರ ಸಾಧನಗಳನ್ನು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ವೈ - ಫೈ ಎನ್ನುವುದು ಒಂದು ವಯರ್‌ಲೆಸ್

© NIMI NOT TO BE REPUBLISHED

ರೂಟರ್‌ನಿಂದ ಹತ್ತಿರದ ಸಾಧನಕ್ಕೆ ಕಳುಹಿಸಲಾದ ರೇಡಿಯೋ ಸಂಕೇತವಾಗಿದೆ. ಇದು ಸಂಕೇತವನ್ನು ನೀವು ನೋಡಬಹುದಾದ ಮತ್ತು ಉಪಯೋಗಿಸಬಹುದಾದ ಡಾಟಾಗೆ ಅನುವಾದಿಸುತ್ತದೆ. ಸಾಧನವು ರೇಡಿಯೊ ಸಂಕೇತವನ್ನು ಮರಳಿ ರೂಟರ್‌ಗೆ ರವಾನಿಸುತ್ತದೆ. ಇದು ವಯರ್ ಅಥವಾ ಕೇಬಲ್ ಮೂಲಕ ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುತ್ತದೆ.

ವೈ - ಫೈ ನೆಟ್‌ವರ್ಕ್ ಎಂದರೇನು? (What is a Wi-Fi network?)

ವೈ - ಫೈ ನೆಟ್‌ವರ್ಕ್ ಎಂದರೆ ವಯರ್‌ಲೆಸ್ ರೂಟರ್ ಮೂಲಕ ಒಂದು ಮನೆ ಅಥವಾ ವ್ಯಾಪಾರದಲ್ಲಿರುವ ಅನೇಕ ಸಾಧನಗಳ ಜೊತೆಗೆ ಹಂಚಿಕೊಳ್ಳಲ್ಪಡುವ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವಾಗಿದೆ. ರೂಟರ್‌ನ್ನು ನಿಮ್ಮ ಇಂಟರ್‌ನೆಟ್ ಮಾಡೆಮ್ ಜೊತೆಗೆ ನೇರವಾಗಿ ಸಂಪರ್ಕಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ನಿಮ್ಮ ಎಲ್ಲಾ ವೈ - ಫೈ ಸಕ್ರೀಯವಾಗಿರುವ ಸಾಧನಗಳಿಗೆ ಇಂಟರ್‌ನೆಟ್ ಸಂಕೇತವನ್ನು ಪ್ರಸಾರ ಮಾಡುವ ಹಬ್ ಆಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು ನೀವು ನಿಮ್ಮ ನೆಟ್‌ವರ್ಕ್ ಕವರೇಜ್ ಪ್ರದೇಶದಲ್ಲಿರುವವರೆಗೆ ಇಂಟರ್‌ನೆಟ್‌ನ ಸಂಪರ್ಕದಲ್ಲಿರಲು ನಮ್ಯತೆ (flexibility) ಯನ್ನು ನೀಡುತ್ತದೆ.

ವೈ - ಫೈ ನ ವಿಸ್ತ್ರತ ರೂಪ (What does Wi-Fi stand for?)

ಈ ಶಬ್ದವನ್ನು ಮಾರ್ಕೆಟಿಂಗ್ ಸಂಸ್ಥೆಯಿಂದ ರಚಿಸಲಾಗಿದೆ. ಏಕೆಂದರೆ ವಯರ್‌ಲೆಸ್ ಉದ್ಯಮವು IEEE 802.11 ಎನ್ನುವ ಬಳಕೆದಾರ ಸ್ನೇಹಿ ಅಲ್ಲದ ತಂತ್ರಜ್ಞಾನವನ್ನು ಉಲ್ಲೇಖಿಸಲು ಬಳಕೆದಾರ ಸ್ನೇಹಿ ಹೆಸರನ್ನು ಹುಡುಕುತ್ತಿತ್ತು ಮತ್ತು ಈ ಹೆಸರು ಸಿಕ್ಕಿತು. Wi-Fi ಸಾಮಾನ್ಯವಾಗಿ WiFi , wifi ಅಥವಾ wi fi ಇದು Wireless Fidelity ಯ ಸಂಕ್ಷಿಪ್ತ ರೂಪ ಎಂದು ಭಾವಿಸಲಾಗಿದೆ. ವ್ಯಾಪಾರ ಸಂಸ್ಥೆಗೆ ಪಾವತಿಸಿದ ಸಂಸ್ಥೆಯನ್ನು ಕೆಲವೊಮ್ಮೆ ವಾಯರ್‌ಲೆಸ್ ಫಿಡಿಲಿಟಿ ಅಲಾಯನ್ಸ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ವೈ - ಫೈ ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ? (How does Wi-Fi work?)

ವೈ - ಫೈ ನಿಮ್ಮ ವಯರ್‌ಲೆಸ್ ರೂಟರ್‌ನಿಂದ ನಿಮ್ಮ ವೈ - ಫೈ ಸಕ್ರಿಯಗೊಳಿಸಿದ ಸಾಧನಗಳಾದ TV, ಸ್ಮಾರ್ಟ್ ಫೋನ್, ಟ್ಯಾಬ್ಲೆಟ್ ಮತ್ತು ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಡಾಟಾವನ್ನು ರವಾನಿಸಲು ರೇಡಿಯೊ ಸಂಕೇತಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಏಕೆಂದರೆ ಅವು ಗಾಳಿಯ ಮೂಲಕ ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸುತ್ತವೆ, ಅದ್ದರಿಂದ ನಿಮ್ಮ ಸಾಧನಗಳು ಮತ್ತು ವೈಯುಕ್ತಿಕ ಮಾಹಿತಿಯು ಹ್ಯಾಕರ್‌ಗಳಿಗೆ, ಸೈಬರ್ - ದಾಳಿಗಳಿಗೆ ಮತ್ತು ಇತರ ಬೆದರಿಕೆಗಳಿಗೆ ದುರ್ಬಲವಾಗಬಹುದು. ಇದು ನೀವು ಕಾಫೀ ಶಾಪ್ ಅಥವಾ ವಿಮಾನ ನಿಲ್ದಾಣಗಳಂತಹ ಸಾರ್ವಜನಿಕ ಸ್ಥಳಗಳಲ್ಲಿರುವ ವೈ - ಫೈ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಸಂಪರ್ಕಿಸಿದಾಗ ವಿಶೇಷವಾಗಿ ನಿಜವಾಗುತ್ತದೆ. ಸಾಧ್ಯವಾದಷ್ಟು ಪಾಸ್‌ವರ್ಡ್ - ರಕ್ಷಿತವಾಗಿರುವ ಅಥವಾ ವೈಯುಕ್ತಿಕ ಹಾಟ್‌ಸ್ಪಾಟ್‌ಗೆ ವಯರ್‌ಲೆಸ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸುವುದು ಉತ್ತಮವಾಗಿರುತ್ತದೆ.

ವೈ - ಫೈ ಸಂಪರ್ಕಗಳ ವಿಧಗಳು (Types of Wi-Fi connections)

ಮನೆಯಲ್ಲಿನ ಇಂಟರ್‌ನೆಟ್ ಕ್ಷೇತ್ರಗಳಲ್ಲಿನ ಮೊಬೈಲ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು ವಿಸ್ತರಿಸಿದಂತೆ ಮನೆಯಲ್ಲಿ ವಯರ್‌ಲೆಸ್ ಆಗಿ ಸಂಪರ್ಕಿಸುವ ನಿಮ್ಮ ಆಯ್ಕೆಗಳು ವಿಸ್ತರಿಸುತ್ತವೆ. ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಯಂತೆಯೆ, ವಯರ್‌ಲೆಸ್ ಸಂಪರ್ಕದಲ್ಲಿಯೂ ಅನುಕೂಲಗಳು ಮತ್ತು ಅನಾನುಕೂಲಗಳು ಇವೆ. ಉದಾಹರಣೆಗೆ, ಸಂಕೇತದ ವೇಗ ಮತ್ತು ಶಕ್ತಿ. ಅವುಗಳಲ್ಲಿ ಕೆಲವನ್ನು ನಾವು ಇಲ್ಲಿ ವಿವರಿಸಿದ್ದೇವೆ.

ವಯರ್‌ಲೈನ್ /ರೂಟರ್ (Wireline/router)

ಇವತ್ತಿನ ದಿನಗಳಲ್ಲಿ ಹೆಚ್ಚಿನ ಮನೆಗಳಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ವಯರ್‌ಲೆಸ್ ರೂಟರ್‌ನ್ನು ಬಳಸುತ್ತಾರೆ. ಇದನ್ನು ಬಳಸುವ ಸಾಧಕಗಳು ಹೀಗಿವೆ : ಸೆಟಪ್ ಮಾಡಲು ಅನುಕೂಲವಾಗಿದೆ, ವೈ -ಫೈ ಪ್ರವೇಶ ಪಾಯಿಂಟ್‌ನ (ರೂಟರ್) ಚಲನಶೀಲತೆಯ ವ್ಯಾಪ್ತಿ ಮತ್ತು ಅನೇಕ ಸಾಧನಗಳನ್ನು ಸಂಪರ್ಕಿಸುವ ಅದರ ಸಾಮರ್ಥ್ಯ. ಅದರ ಬಾಧಕಗಳು ಹೀಗಿವೆ : ಸೀಮಿತ ಬ್ಯಾಂಡ್ ವಿಡ್ತ ಮತ್ತು ಒಂದೇ ವೈ - ಫೈ ಗೆ ಹಲವಾರು ಸಾಧನಗಳನ್ನು ಜೋಡಿಸಿದಾಗ, ಮತ್ತು ಮನೆಯಲ್ಲಿರುವ ಇತರ ಇಲೆಕ್ಟ್ರೋಮ್ಯಾಗ್ನೆಟಿಕ್ ಸಾಧನಗಳ ಹಸ್ತಕ್ಷೇಪದಿಂದ ಅದರ ವೇಗವು ಕಡಿಮೆಯಾಗುತ್ತದೆ.

ಮೊಬೈಲ್ ಹಾಟ್‌ಸ್ಪಾಟ್ ಅಥವಾ ಜೆಟ್‌ಪ್ಯಾಕ್ (Mobile hotspot or jetpack)

ಮೊಬೈಲ್ ಮತ್ತು ಮೀಸಲಾದ ಹಾಟ್‌ಸ್ಪಾಟ್‌ಗಳು ಪ್ರಯಾಣದಲ್ಲಿರುವಾಗ ಸುರಕ್ಷಿತವಾಗಿ ಸಂಪರ್ಕಿಸುವ ಹೆಚ್ಚು ಜನಪ್ರಿಯವಾಗಿರುವ ವಿಧಾನಗಳಾಗುತ್ತಿವೆ. ಎರಡು ಸಾಮಾನ್ಯ ಹಾಟ್‌ಸ್ಪಾಟ್ ಸಾಧನಗಳು ಯಾವವೆಂದರೆ - ನಿಮ್ಮ ಸ್ಮಾರ್ಟ್‌ಫೋನ್ ಮತ್ತು ಜೆಟ್‌ಪ್ಯಾಕ್. ಇವತ್ತು ಯಾವುದೇ ಸ್ಮಾರ್ಟ್‌ಫೋನ್ ಅಥವಾ ಟ್ಯಾಬ್ಲೆಟ್‌ನ್ನು ತಾತ್ಕಾಲಿಕ ಹಾಟ್‌ಸ್ಪಾಟ್ ಆಗಿ ಬಳಸಬಹುದು ಮತ್ತು ಇದು ನಿಮ್ಮ ಸಾಂದರ್ಭಿಕ ಅಗತ್ಯತೆಗೆ ಉತ್ತಮ ಆಯ್ಕೆಯಾಗಿದೆ. ಇದು ಬಳಸಲು ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಇದಕ್ಕೆ ಹೆಚ್ಚುವರಿ ಸಾಧನಗಳನ್ನು ಖರೀದಿಸುವ ಅಗತ್ಯವಿಲ್ಲ. ಆದರೆ ಇದು ನಿಮ್ಮ ಬ್ಯಾಟರಿ ಬಾಳಿಕೆಗೆ ಹೊಡೆತ ಕೊಡುತ್ತದೆ ಮತ್ತು ಡಾಟಾ ವೇಗ ಹೆಚ್ಚಾಗಿರುತ್ತದೆ. ಮತ್ತೊಂದೆಡೆ, ಜೆಟ್‌ಪ್ಯಾಕ್ ಸಮರ್ಪಿತ ಮೊಬೈಲ್ ಹಾಟ್‌ಸ್ಪಾಟ್ ಆಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು ನಿಮ್ಮ ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ನಂತೆಯೆ ನಿಮ್ಮ ಪ್ರದೇಶದಲ್ಲಿರುವ ಸೆಲ್ ಟವರ್‌ಗಳಿಂದ ಸಂಕೇತಗಳನ್ನು ತೆಗೆದುಕೊಂಡು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಇದಕ್ಕೆ ಹೆಚ್ಚಿನ ಸಾಧನಗಳನ್ನು ಸಂಪರ್ಕಿಸಬಹುದು ಮತ್ತು ಇದು ವೈ - ಫೈ ನ ಹೆಚ್ಚಿನ ವ್ಯಾಪ್ತಿಯನ್ನು ನೀಡುತ್ತದೆ. ಇದು ಪ್ರತ್ಯೇಕ ಸಾಧನವಾಗಿರುವುದರಿಂದ ನಿಮ್ಮ ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ನ ಬ್ಯಾಟರಿ ಶಕ್ತಿಯನ್ನು ಬಳಸುವುದಿಲ್ಲ. ಅನಾನುಕೂಲವೆಂದರೆ

© NIMI NOT TO BE REPUBLISHED

ನೀವು ಜೆಟ್‌ಪ್ಯಾಕನ್ನು ಖರೀದಿಸಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಪ್ರತ್ಯೇಕ ಯೋಜನೆಯನ್ನು ತಯಾರಿಸಬೇಕಾಗುತ್ತದೆ.

LTE ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್ (LTE Home Internet)

ನೀವು ಗ್ರಾಮೀಣ ಪ್ರದೇಶದಲ್ಲಿ ವಾಸಿಸುತ್ತಿದ್ದರೆ, ನಿಮ್ಮ ಇಂಟರ್‌ನೆಟ್‌ನ ಆಯ್ಕೆಗಳು ಸೀಮಿತವಾಗಿರುತ್ತವೆ. 4G LTE ಹೋಮ್ ಇ ಂಟರ್‌ನೆಟ್ ನ್ನು ಪರಿಗಣಿಸಲು ಯೋಗ್ಯವಾಗಿದೆ. ಇದು ಸೆಲ್ ಫೋನ್ ಟವರ್‌ಗಳ ಮೂಲಕ ಅತ್ಯಂತ ವೇಗದ ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಯನ್ನು ಮತ್ತು 25 MBPS ನಷ್ಟು ಸರಾಸರಿ ಡೌನ್‌ಲೋಡ್ ವೇಗವನ್ನು ವಿತರಿಸುತ್ತದೆ. ಉಪಗ್ರಹಕ್ಕಿಂತ LTE ನ ಅನುಕೂಲಗಳು ಯಾವವೆಂದರೆ ಉತ್ತಮ ವೇಗ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹತೆ. ಇವು ನಿಮ್ಮ ವಾಹಕವನ್ನು ಅವಲಂಬಿಸಿರುತ್ತವೆ.

5G ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್ (5G Home Internet): 5G ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್ (ಸ್ಥಿರ ವಯರ್‌ಲೆಸ್ ಎಕ್ಸೆಸ್) ಈಗ ದೇಶದಾದ್ಯಂತ ಅನೇಕ ಸ್ಥಳಗಳಲ್ಲಿದೆ. ಅನೇಕ ಸಾಧನಗಳನ್ನು ಜೋಡಿಸಿದರೂ ಸಹ ಇದು ನಿಮ್ಮ ಇಡೀ ಮನೆಯಲ್ಲಿನ ಸಾಧನಗಳನ್ನು ಅತ್ಯಂತ ವೇಗವಾಗಿ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿ ಸಂಪರ್ಕಿಸುತ್ತದೆ. ಸ್ಥಿರ ವಯರ್‌ಲೆಸ್ ಎಕ್ಸೆಸ್ ಸರಳವಾದ ಪ್ಲಗ್ ಮತ್ತು ಪ್ಲೇ ಸೆಟ್ಟಿಂಗ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇದರರ್ಥ ಅಲ್ಲಿ ಯಾವುದೇ ಗೊಂದಲಮಯ ವಯರ್‌ಗಳು ಇರುವುದಿಲ್ಲ ಮತ್ತು ಸೆಟಪ್ ಮಾಡಲು ನಿಮ್ಮ ಮನೆಗೆ ಯಾವುದೇ ತಂತ್ರಜ್ಞರಿಗಾಗಿ ಕಾಯುವ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ.

ಮನೆಯಲ್ಲಿ ವೈ - ಫೈ ನ್ನು ಹೇಗೆ ಪಡೆಯುವುದು(How to get Wi-Fi at home)

ನಿಮ್ಮ ಮನೆಯಲ್ಲಿ ವಯರ್‌ಲೆಸ್ ಸೇವೆಯನ್ನು ಪಡೆಯಲು ಹಲವಾರು ಮಾರ್ಗಗಳಿವೆ ಮತ್ತು ಹೆಚ್ಚಿನವು ಭೌಗೋಳಿಕ ಸ್ಥಳವನ್ನು ಮತ್ತು ಲಭ್ಯತೆಯನ್ನು ಅವಲಂಬಿಸಿರುತ್ತವೆ. (ನಿಮ್ಮ ಪ್ರದೇಶದಲ್ಲಿ ವೈ - ಫೈ ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್ ಲಭ್ಯವಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸಿ). ನಗರ ಮತ್ತು ಉಪನಗರಗಳ ಬಹುಪಾಲು ಪ್ರದೇಶಗಳು 5G ಹೋಮ ಇಂಟರ್‌ನೆಟ್‌ನೊಂದಿಗೆ ಈ ಸೇವೆಗಳನ್ನು ನೀಡುತ್ತವೆ. ಗ್ರಾಮೀಣ ಪ್ರದೇಶಗಳು ಉಪಗ್ರಹ ಸೇವೆಯನ್ನು ಮತ್ತು 4G LTE ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಹೆಚ್ಚಾಗಿ ನೀಡುತ್ತವೆ. ನೀವು ವಯರ್ಡ್ ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಯನ್ನು ಹೊಂದಿದ್ದರೆ, ನಿಮ್ಮ ಮನೆಯಲ್ಲಿ ನೀವು ಸ್ವಂತ ವೈ - ಫೈ ನೆಟ್‌ವರ್ಕನ್ನು ಸೆಟಪ್ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ರೂಟರ್‌ಗೆ ನಿಮ್ಮ ಮಾಡೆಮ್‌ನ್ನು ಜೋಡಿಸುವ ಮೂಲಕ ನಿಮ್ಮ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ವ್ಯಾಪ್ತಿಯಲ್ಲಿರುವ ನಿಮ್ಮ ಎಲ್ಲ ವೈ - ಫೈ ಸಕ್ರೀಯಗೊಂಡ ಸಾಧನಗಳೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳಬಹುದು. ನಿಮ್ಮ ಮನೆಯು ಎರಡು ಹಂತಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಕಾಂಕ್ರೀಟ್ ಗೋಡೆಗಳು ಕೆಲವು ವಲಯಗಳಿಗೆ, ವಯರ್‌ಲೆಸ್ ಸಂಕೇತವನ್ನು ಪ್ರಸಾರ ಮಾಡುವ ವೈ - ಫೈ ಎಕ್ಸ್‌ಟೆಂಡರ್‌ನ್ನು ಸೇರಿಸುವುದರಿಂದ ದೊಡ್ಡ ವ್ಯಾತ್ಯಾಸವನ್ನು ಮಾಡಬಹುದು.

ನಿಮ್ಮ ಮೊಬೈಲ್ ಸಾಧನಗಳ ಸಂಖ್ಯೆ ಹೆಚ್ಚಾದಂತೆ ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್‌ನ ಬೇಡಿಕೆಯೂ ಹೆಚ್ಚಾಗುತ್ತದೆ ಎಂದು ನೆನಪಿನಲ್ಲಿಡಿ. ನಿಮ್ಮ ಸಾಧನಗಳು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ನಿಮ್ಮ ಇಂಟರ್‌ನೆಟ್ ವೇಗದ ಪ್ಲ್ಯಾನ್‌ನ್ನು ಅಪ್‌ಗ್ರೇಡ್ ಮಾಡಬೇಕಾಗಬಹುದು. ವೆರಿಝಾನ್ ಹಲವಾರು ಇಂಟರ್‌ನೆಟ್ ಸೇವೆಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ - DSL ಮತ್ತು ಫೈಬರ್‌ದಿಂದ ಹಿಡಿದು 5G ಹೋಮ್ ಇಂಟರ್‌ನೆಟ್‌ವರೆಗೆ, ನಿಮ್ಮ ಸ್ಥಳವನ್ನು ಅವಲಂಬಿಸಿದಂತೆ. ಉತ್ತಮ ವಿಶ್ವಾಸಾರ್ಹತೆ ಮತ್ತು ವೆರಿಝಾನ್‌ನ ಅತ್ಯಂತ ವೇಗದ ಡೌನ್‌ಲೋಡ್‌ಗಾಗಿ, ನಿಮ್ಮ ಪ್ರದೇಶದಲ್ಲಿ Fios ಇಂಟರ್‌ನೆಟ್ ಅಥವಾ 5G ಹೋಮ ಇಂಟರ್‌ನೆಟ್ ಲಭ್ಯವಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸಿ.

ನಿಮ್ಮ ಪ್ರದೇಶದಲ್ಲಿ ಯಾವ ವೈ ಫೈ ಇಂಟರ್‌ನೆಟ್ ಲಭ್ಯವಿದೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸಿ.

LAN ನ ಪರಿಚಯ (Introduction to LAN)

ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್(Local Area Network- LAN) ನಲ್ಲಿ ಒಂದು ನಿರ್ದಿಷ್ಟ ಸ್ಥಳದಲ್ಲಿರುವ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಒಟ್ಟಿಗೆ ಜೋಡಿಸಿ ನೆಟ್‌ವರ್ಕನ್ನು ರೂಪಿಸಲಾಗುತ್ತದೆ. LAN ನಲ್ಲಿರುವ ಕಂಪ್ಯೂಟರ್‌ಗಳು TCP/IP ಈಥರ್‌ನೆಟ್ ಮೂಲಕ ಅಥವಾ ವೈ -ಫೈ ಮೂಲಕ ಪರಸ್ಪರ ಸಂಪರ್ಕಿಸುತ್ತವೆ. LAN ವಿಶೇಷವಾಗಿ ಶಾಲೆ, ಕಚೇರಿ, ಸಂಘ ಅಥವಾ ಚರ್ಚ್ ಮುಂತಾದ ಸಂಸ್ಥೆಗಳಿಗೆ ಮಾತ್ರ ಇರುತ್ತದೆ.

LAN ಸಾಧನ ಎಂದರೇನು? (What is LAN device?)

ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್ (LAN) ಎನ್ನುವುದು ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮತ್ತು ಬಾಹ್ಯ ಸಾಧನಗಳ ಗುಂಪಾಗಿದೆ. ಇವು ಒಂದು ವಿಶಿಷ್ಟ ಭೌಗೋಳಿಕ ಪ್ರದೇಶದಲ್ಲಿರುವ ಸರ್ವರ್‌ಗೆ ಒಂದೇ ಸಂವಹನ ಲೈನನ್ನುಅಥವಾ ವಯರ್‌ಲೆಸ್ ಲಿಂಕ್‌ನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತವೆ. ಹೋಮ್ ಆಫೀಸ್‌ನಲ್ಲಿರುವ ಎರಡು ಅಥವಾ ಮೂರು ಬಳಕೆದಾರರಷ್ಟು ಕಡಿಮೆ ಇರುವ ಅಥವಾ ಕಾರ್ಪೋರೇಷನ್ ಕೇಂದ್ರ ಆಫೀಸಿನಲ್ಲಿರುವ ಸಾವಿರಾರು ಬಳಕೆದಾರರರು ಇರುವ ನೆಟ್‌ವರ್ಕಗಳು ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸಬಹುದು.

ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳ ವಿಧಗಳು (Types of network devices)

- **ಹಬ್** (Hub)
- **ಸ್ವಿಚ್** (Switch).
- **ರೂಟರ್** (Router).
- **ಬ್ರಿಡ್ಜ್** (Bridge).

© NIMI NOT TO BE REPUBLISHED

- ಗೇಟ್‌ವೇ (Gateway).
- ಮಾಡೆಮ್ (Modem).
- ರಿಪೀಟರ್ (Repeater).
- ಎಕ್ಸೆಸ್ ಪಾಯಿಂಟ್ (Access Point).

ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್‌ನ ವಿಧಗಳು (Types of Local Area Network (LAN))

- Peer to peer (P2P) LAN. ...
- Token ring LAN. ...
- Token bus LAN. ...
- Wired LAN. ...
- Cloud-managed LAN. ...

Fig 7

	LAN	MAN	WAN
GRAPHICAL AREA	Small area, 1 to 5 km	City limits, 50 to 60 km	Global, up to 1000s of km
TYPICAL CUSTOMERS	Schools and colleges, offices, small industries and SMBs	Mid-market to large enterprises, city governments, business chains and financial institutions	Large nationwide or global enterprises
TECHNICAL ASPECTS	Ethernet and Ethernet switches; some L3 switching and routing	Ethernet switching on the LAN; Metro Ethernet on the MAN	MPLS, VPLS, SONET and satellite technologies

- Public internet. ...
- Wired end-user devices. ...
- Mobile end-user devices.

ಲ್ಯಾನ್ vs. ಮ್ಯಾನ್ vs. ವ್ಯಾನ್ (LAN vs. MAN vs. WAN (Fig 7)

ಇಂಟರ್‌ನೆಟ್ ಪರಿಕಲ್ಪನೆಗಳು (Internet concepts)

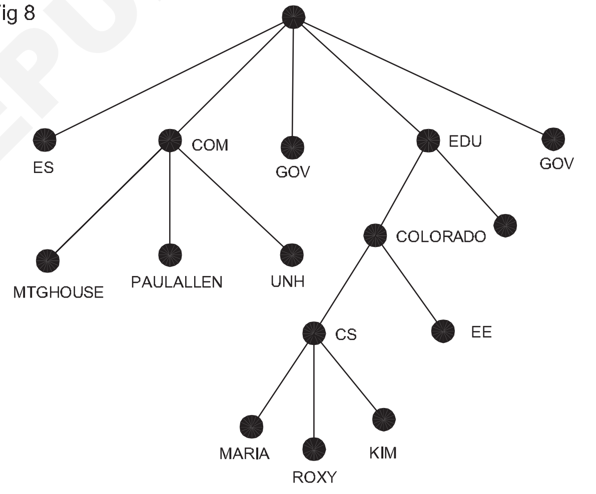
- www ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- www ರಚನೆ
- ಬಳಕೆದಾರರ ಇಂಟರ್‌ಫೇಸ್ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳು

www ಎಂದರೆ World Wide Web. ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನ ತಾಂತ್ರಿಕ ವ್ಯಾಖ್ಯಾನವೆಂದರೆ - ಹೈಪರ್‌ಟೆಕ್ಸ್ ಟ್ರಾನ್ಸ್‌ಫರ್ ಪ್ರೊಟೊಕಾಲ್‌ನ್ನು (HTTP) ಬಳಸುತ್ತಿರುವ ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಸಂಪನ್ಮೂಲಗಳು ಮತ್ತು ಬಳಕೆದಾರರನ್ನು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಒಳಗೊಂಡಿರುತ್ತದೆ.

ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್, ನೆಟ್‌ವರ್ಕ್-ಪ್ರವೇಶಿಸಬಹುದಾದ ಮಾಹಿತಿ, ಮಾನವ ಜ್ಞಾನದ ಸಾಕಾರದ ವಿಶ್ವವಾಗಿದೆ.

ಸರಳ ಭಾಷೆಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್-ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮತ್ತು ಇಂಟರ್‌ನೆಟ್‌ನ ನಡುವೆ ಮಾಹಿತಿ ವಿನಿಮಯ ಮಾಡುವ ಒಂದು ಮಾರ್ಗವಾಗಿದೆ. ಇದು ಅವುಗಳನ್ನು ಒಟ್ಟಿಗೆ ಜೋಡಿಸುವ ಮಲ್ಟಿಮೀಡಿಯಾ ಸಂಪನ್ಮೂಲಗಳ ಸಂವಾದಾತ್ಮಕವಾದ ದೊಡ್ಡ ಸಂಗ್ರಹವಾಗಿದೆ.

Fig 8



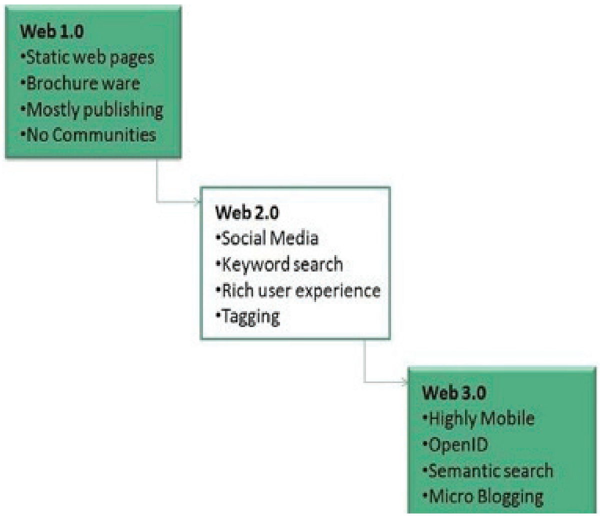
ಇಂಟರ್‌ನೆಟ್ ಮತ್ತು ವೆಬ್ ಎರಡೂ ಒಂದೇ ಅಲ್ಲ : ವೆಬ್, ಮಾಹಿತಿಯನ್ನು ರವಾನಿಸಲು ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. (Fig 8)

ವಿಕಸನ (Evolution) : ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನ್ನು Timothy Berners Lee ಅವರು1989 ರಲ್ಲಿ ಜಿನೆವಾದ CERN ನಲ್ಲಿ ರಚಿಸಿದರು. ಎಲ್ಲ ಸಂಶೋಧಕರು CERN ನಲ್ಲಿ ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಮತ್ತು ದಕ್ಷತೆಯಿಂದ ಒಟ್ಟಿಗೆ ಕೆಲಸ ಮಾಡುವ ಸಲುವಾಗಿ ಪ್ರಸ್ತಾವನೆಯಂತೆ ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಅಸ್ತಿತ್ವಕ್ಕೆ ಬಂದಿತು.

ಕೆಳಗಿನ ಚಿತ್ರವು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನ ವಿಕಸನವನ್ನು ಸಂಕ್ಷಿಪ್ತವಾಗಿ

© NIMI NOT TO BE REPUBLISHED

Fig 9



ವಿವರಿಸುತ್ತದೆ : (Fig 9)

WWW ರಚನೆ (WWW Architecture)

WWW ರಚನೆಯನ್ನು ಕೆಳಗಿನ ಚಿತ್ರದಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ ಹಲವಾರು ಲೇಯರಗಳಲ್ಲಿ ವಿಂಗಡಿಸಲಾಗಿದೆ :

ಐಡೆಂಟಿಫಯರ್‌ಗಳು ಮತ್ತು ಕ್ಯಾರೆಕ್ಟರ್ ಸೆಟ್ (Identifiers and Character Set)

ಯುನಿಫಾರ್ಮ್ ರಿಸೋರ್ಸ್ ಐಡೆಂಟಿಫಯರ(Uniform Resource Identifier (URI)) ಇದನ್ನು ವೆಬ್‌ನಲ್ಲಿರುವ ಸಂಪನ್ಮೂಲಗಳನ್ನು ವಿಶಿಷ್ಟವಾಗಿ ಗುರುತಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ. UNICODE ಇದನ್ನು ಮಾನವ ಭಾಷೆಗಳಲ್ಲಿ ಓದಲು ಮತ್ತು ಬರೆಯಲು ಸಾಧ್ಯವಾಗುವಂತೆ ವೆಬ್ ಪುಟಗಳನ್ನು ನಿರ್ಮಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

ಸಿಮ್ಯಾಂಟಿಕ್ ವೆಬ್‌ನಲ್ಲಿ ಸಾಮಾನ್ಯ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು XML (Extensible Markup Language) ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಡಾಟಾ ವಿನಿಮಯ (Data Interchange)

Resource Description Framework (RDF) ಫ್ರೇಮ್‌ವರ್ಕ್ ಆಗಿದ್ದು, ಇದು ವೆಬ್‌ಗಾಗಿ ಡಾಟಾದ ಕೋರ್ ಪ್ರಾತಿನಿಧ್ಯವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

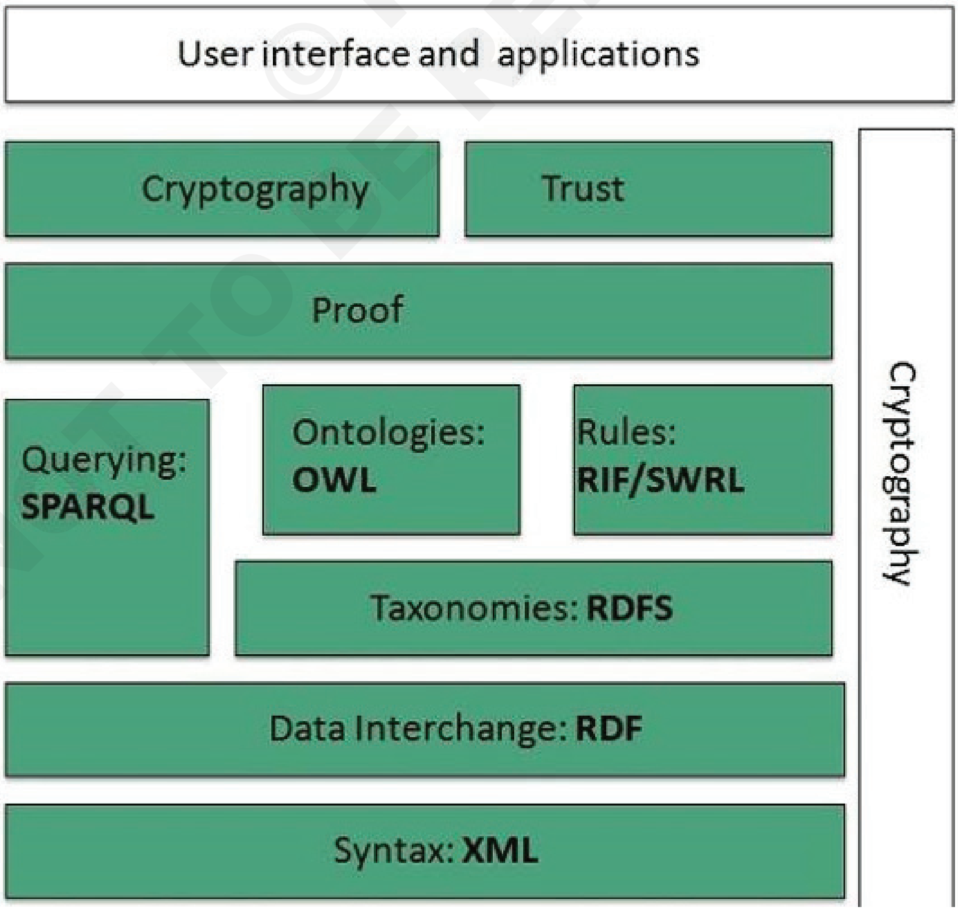
ಟೆಕ್ಸಾನಾಮಿಗಳು (Taxonomies)

RDF Schema (RDFS) ಇದು ಟೆಕ್ಸಾನಾಮಿಗಳ ಹೆಚ್ಚಿನ ಪ್ರಮಾಣೀಕೃತ ವಿವರಣೆಯನ್ನು ಮತ್ತು ಇತರ ಮೂಲತತ್ವ ವಿಚಾರಗಳ ರಚನೆಯನ್ನು ಅನುಮತಿಸುತ್ತದೆ.

ಒಂಟೊಲಜಿಗಳು (Ontologies) (Fig 10)

Web Ontology Language (OWL) ಇದು RDFS ಗಳಿಗೆ ಹೆಚ್ಚಿನ

Fig 10



© NIMI NOT TO BE REPUBLISHED

ರಚನೆಗಳನ್ನು ನೀಡುತ್ತದೆ. ಇದು ಕೆಳಗಿನ ಮೂರು ಆವೃತ್ತಿಗಳಲ್ಲಿ ಬರುತ್ತದೆ :

- OWL Lite - ಟೆಕ್ಸಾನಾಮಿಗಳಿಗೆ ಮತ್ತು ಸರಳ ನಿರ್ಬಂಧಗಳಿಗಾಗಿ
- OWL DL - ಸಂಪೂರ್ಣ ವಿವರಣೆಯ ಲಾಜಿಕ ಬೆಂಬಲಕ್ಕಾಗಿ
- OWL - RDF ನ ಹೆಚ್ಚಿನ ಸಿಂಟ್ಯಾಕ್ಟಿಕ್ ಸ್ವಾತಂತ್ರಕ್ಕಾಗಿ

ನಿಯಮಗಳು (Rules)

RDFs ಮತ್ತು OWL ನಲ್ಲಿ ಲಭ್ಯವಿರುವ ರಚನೆಗಳನ್ನು ಮೀರಿ RIF ಮತ್ತು SWRL ನಿಯಮಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. Simple Protocol ಮತ್ತು RDF Query Language (SPARQL) ಇವು SQL ರೀತಿಯ ಭಾಷೆಗಳಾಗಿದ್ದು, ಇವುಗಳನ್ನು RDF ಡಾಟಾ ಮತ್ತು OWL ಒಂಟಾಲಜಿಗಳನ್ನು ಪ್ರಶ್ನಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಪ್ರೂಫ್ (Proof)

ಎಲ್ಲ ಸಿಮ್ಯಾಂಟಿಕ್ ಮತ್ತು ನಿಯಮಗಳನ್ನು ಪ್ರೂಫ್ ಕೆಳಗಿನ ಲೇಯರ್‌ನಲ್ಲಿ ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅವುಗಳ ಫಲಿತಾಂಶವನ್ನು ಡಿಡಕ್ಷನ್‌ಗಳನ್ನು ಸಾಬೀತುಪಡಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿ (Cryptography)

ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿ ಎಂದರೆ ಬಳಸಲಾದ ಸೋರ್ಸ್‌ಗಳ ಮೂಲವನ್ನು ಪರಿಶೀಲಿಸುವ ಡಿಜಿಟಲ್ ಸಿಗ್ನೇಚರ್ ಆಗಿದೆ.

ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳು (User Interface and Applications)

ಲೇಯರ್‌ನ ಮೇಲ್ಭಾಗದಲ್ಲಿ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ User Interface and Applications ಲೇಯರ್‌ನ್ನು , ಬಳಕೆದಾರರ ಸಂವಹನಕ್ಕಾಗಿ ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ.

WWW ಕಾರ್ಯಾಚರಣೆ (WWW Operation)

WWW ಇದು ಕ್ಲೈಂಟ್ - ಸರ್ವರ್ ವಿಧಾನದಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಕೆಳಗಿನ ಹಂತಗಳು ವೆಬ್ ಯಾವ ರೀತಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ವಿವರಿಸುತ್ತವೆ :

1 ಬಳಕೆದಾರರು ವೆಬ್ ಬ್ರೌಸರ್‌ನ ಅಡ್ರೆಸ್ ಬಾರ್‌ನಲ್ಲಿ ವೆಬ್ ಪೇಜ್‌ನ URL ನ್ನು (ಉದಾ : http://www.tutorialspoint.com) ನಮೂದಿಸುತ್ತಾರೆ.

2 http://www.tutorialspoint.com ಗೆ ಸಂಬಂಧಿಸಿದ IP ಅಡ್ರೆಸ್‌ಗಾಗಿ ಬ್ರೌಸರ್, ಡೊಮೇನ್ ನೇಮ್ ಸರ್ವರ್‌ನ್ನು ವಿನಂತಿಸುತ್ತದೆ.

3 IP ಅಡ್ರೆಸ್‌ನ್ನು ಸ್ವೀಕರಿಸಿದ ನಂತರ, HTTP ಪ್ರೊಟೊಕಾಲನ್ನು ಬಳಸಿಕೊಂಡು ವೆಬ್ ಪುಟಕ್ಕಾಗಿ ಬ್ರೌಸರ್ ವೆಬ್ ಸರ್ವರ್‌ಗೆ ವಿನಂತಿಯನ್ನು ಕಳುಹಿಸುತ್ತದೆ. ಇದು ಬ್ರೌಸರ ಮತ್ತು ವೆಬ್ ಸರ್ವರ್ ಸಂವಹನ ಮಾಡುವ ಮಾರ್ಗವನ್ನು ಸೂಚಿಸುತ್ತದೆ.

4 HTTP ಪ್ರೊಟೊಕಾಲನ್ನು ಬಳಸಿಕೊಂಡು ವೆಬ್ ಸರ್ವರ್ ವಿನಂತಿಯನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ. ಮತ್ತು ವಿನಂತಿಸಲಾದ ವೆಬ್ ಪುಟಕ್ಕಾಗಿ ತನ್ನ ಹುಡುಕಾಟವನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಆ ವೆಬ್ ಪುಟವು ಸಿಕ್ಕರೆ ಅದನ್ನು ವೆಬ್ ಬ್ರೌಸರ್‌ಗೆ ಹಿಂತಿರುಗಿಸುತ್ತದೆ ಮತ್ತು HTTP ಸಂಪರ್ಕವನ್ನು ಮುಚ್ಚುತ್ತದೆ.

5 ಈಗ ವೆಬ್ ಬ್ರೌಸರ್, ವೆಬ್ ಪುಟವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ. ಅದನ್ನು ಅರ್ಥೈಸುತ್ತದೆ(interprets) ಮತ್ತು ವೆಬ್ ಬ್ರೌಸರ್‌ನ ವಿಂಡೊದಲ್ಲಿ ವೆಬ್ ಪುಟದ ವಿಷಯಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಭವಿಷ್ಯ (Future)

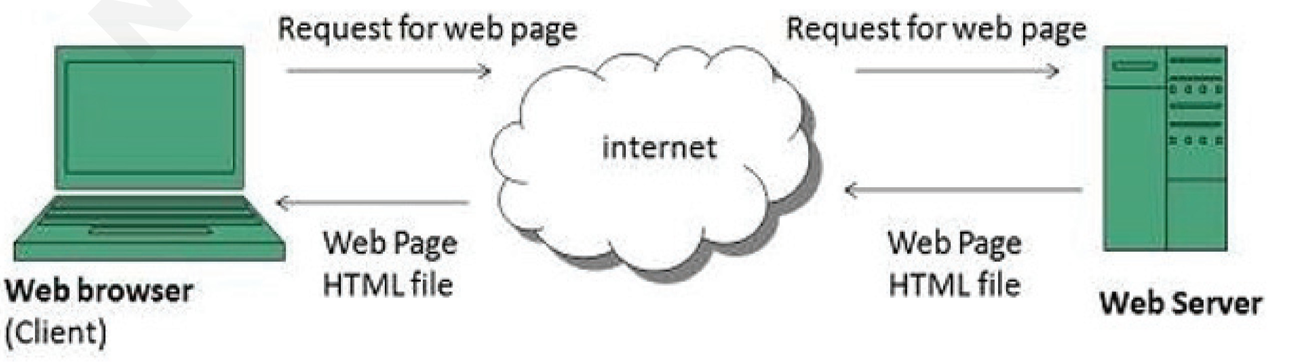
ವೆಬ್ ಕ್ಷೇತ್ರದಲ್ಲಿ ಅತ್ಯಂತ ತ್ವರಿತ ಬೆಳವಣಿಗೆಯು ಕಂಡು ಬಂದಿದೆ. ಇದು ಪ್ರತಿಯೊಂದು ಕ್ಷೇತ್ರದಲ್ಲಿಯೂ ತನ್ನ ಪ್ರಭಾವವನ್ನು ಹೊಂದಿದೆ. ಉದಾಹರಣೆಗೆ : ಶಿಕ್ಷಣ, ಸಂಶೋಧನೆ, ತಂತ್ರಜ್ಞಾನ, ವಾಣಿಜ್ಯ, ಮಾರುಕಟ್ಟೆ ಇತ್ಯಾದಿಗಳು. ಅದ್ದರಿಂದ ವೆಬ್‌ನ ಭವಿಷ್ಯವು ಬಹುತೇಕ ಅನಿರೀಕ್ಷಿತವಾಗಿದೆ.

WWW ಕ್ಷೇತ್ರದಲ್ಲಿ ಅತಿ ದೊಡ್ಡ ಅಭಿವೃದ್ಧಿಯ ಹೊರತಾಗಿ, W3 ಒಕ್ಕೂಟವು ನಿಭಾಯಿಸಬೇಕಾದ ಕೆಲವು ತಾಂತ್ರಿಕ ಸಮಸ್ಯೆಗಳು ಸಹ ಇವೆ.

ಬಳಕೆದಾರ ಇಂಟರ್‌ಫೇಸ್ (User Interface) (Fig 11)

3-D ಮಾಹಿತಿಯ ಉತ್ತಮ ಗುಣಮಟ್ಟದ ಪ್ರೆಸೆಂಟೇಷನ್ ಕೆಲಸವು ಅಭಿವೃದ್ಧಿಯಾಗುತ್ತಿದೆ. W3 ಒಕ್ಕೂಟವು ಜಾಗತಿಕ ಸಮುದಾಯಗಳ ಅವಶ್ಯಕತೆಗಳನ್ನು ಪೂರ್ಣಗೊಳಿಸಲು ವೆಬ್‌ನ್ನು ವೃದ್ಧಿಸುವ ಸಲುವಾಗಿ

Fig 11



© NIMI NOT TO BE REPUBLISHED

ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿದೆ. ಈ ಕೆಲಸವು ಎಲ್ಲ ಪ್ರಾದೇಶಿಕ ಭಾಷೆಗಳನ್ನು ಮತ್ತು ಬರವಣಿಗೆಯ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ತಂತ್ರಜ್ಞಾನ (Technology)

ಗೌಪ್ಯತೆ ಮತ್ತು ಭದ್ರತೆಯ ಕೆಲಸ ನಡೆಯುತ್ತಿದೆ. ಇದು ಮಾಹಿತಿಯನ್ನು ಮರೆಮಾಡುವುದು, ಅಕೌಂಟಿಂಗ್, ಪ್ರವೇಶ ನಿಯಂತ್ರಣ, ಸಮಗ್ರತೆ ಮತ್ತು ಅಪಾಯ ನಿರ್ವಹಣೆ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ರಚನೆ (Architecture)

ವೆಬ್ ಕ್ಷೇತ್ರದಲ್ಲಿ ದೊಡ್ಡ ಬೆಳವಣಿಗೆಗಳು ಆಗುತ್ತಿವೆ. ಇದು ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಅತಿಯಾದ ಹೊರೆಗೆ ಕಾರಣವಾಗಬಹುದು ಮತ್ತು ಅದರ ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು ಕುಗ್ಗಿಸಲು ಕಾರಣವಾಗಬಹುದು. ಅದ್ದರಿಂದ ಅತ್ಯುತ್ತಮ ಪ್ರೊಟೊಕಾಲನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಅವಶ್ಯಕತೆ ಇದೆ.

ಇ ಮೇಲ್(ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮೇಲ್) ಎಂದರೇನು?

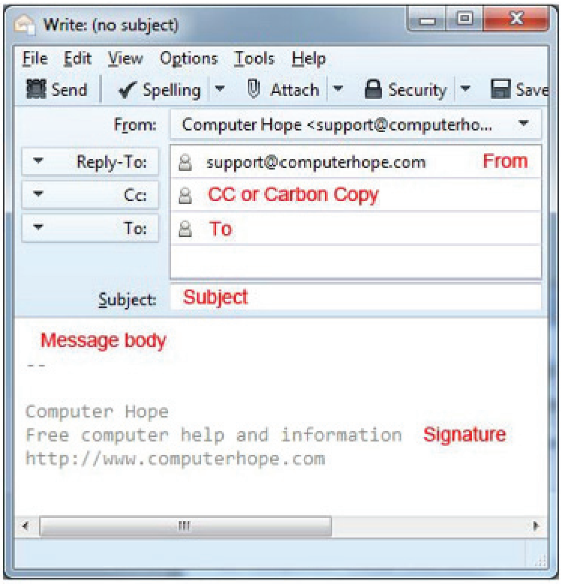
ಇ ಮೇಲ್(ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮೇಲ್) ಎಂದರೆ ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ಸಂದೇಶಗಳನ್ನು ಒಬ್ಬ ಬಳಕೆದಾರರಿಂದ ಒಬ್ಬರು ಅಥವಾ ಅನೇಕರಿಗೆ ಕಳುಹಿಸುವ ಮಾಹಿತಿ ವಿನಿಮಯವಾಗಿದೆ.ಇ- ಮೇಲ್‌ಗಳು ವೇಗವಾಗಿರುತ್ತವೆ, ಅಗ್ಗವಾಗಿರುತ್ತವೆ ಮತ್ತು ವ್ಯಾಪಾರ ಅಥವಾ ವೈಯಕ್ತಿಕ ಬಳಕೆಗಾಗಿ ಸಂವಹನ ನಡೆಸಲು ಪ್ರವೇಶಿಸಬಹುದಾದ ಮಾರ್ಗವಾಗಿದೆ.

ಇ-ಮೇಲ್ ಬರೆಯುವುದು (Writing an e-mail) (Fig 12)

ಇ-ಮೇಲ್ ಸಂದೇಶವನ್ನು ಬರೆಯುವಾಗ, ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿರುವಂತೆ ವಿಂಡೊ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ. ನೀವು ನೋಡುವಂತೆ ಇ-ಮೇಲ್ ಕಳುಹಿಸುವಾಗ ಹಲವಾರು ಫೀಲ್ಡ್‌ಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ :

- TO ಫೀಲ್ಡ್‌ನಲ್ಲಿ ಇ-ಮೇಲ್ ಅನ್ನು ಸ್ವೀಕರಿಸುವ ವ್ಯಕ್ತಿಯ ಇ-ಮೇಲ್ ಅಡ್ರೆಸನ್ನು ನೀಡಬೇಕಾಗುತ್ತದೆ.
- From ಫೀಲ್ಡ್ - ನಿಮ್ಮ ಇ-ಮೇಲ್ ಅಡ್ರೆಸನ್ನು ಹೊಂದಿರುತ್ತದೆ.
- ನೀವು ಸಂದೇಶಕ್ಕೆ ಪ್ರತ್ಯುತ್ತರಿಸುತ್ತಿದ್ದರೆ, To : ಮತ್ತು From: ಫೀಲ್ಡ್‌ಗಳು ತಮ್ಮಷ್ಟಕ್ಕೆ ತಾವೇ ಭರ್ತಿಯಾಗುತ್ತವೆ. ಒಂದು ವೇಳೆ ಇದು ಹೊಸ ಸಂದೇಶವಾಗಿದ್ದರೆ, ನೀವು To: ಫೀಲ್ಡ್‌ನಲ್ಲಿ ರೆಸಿಪಿಯೆಂಟ್ (ಸ್ವೀಕರಿಸುವವರು) ಗಳನ್ನು ಸೂಚಿಸಬೇಕಾಗುತ್ತದೆ. ರೆಸಿಪಿಯಂಟ್‌ಗಳನ್ನು ನಿಮ್ಮ ಕಾಂಟ್ಯಾಕ್ಟ್ ಲಿಸ್ಟ್‌ನಿಂದ ಆಯ್ಕೆ ಮಾಡಲಾಗುತ್ತದೆ ಅಥವಾ ಅವರ ಇ-ಮೇಲ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಟೈಪ್ ಮಾಡುವ ಮೂಲಕ ಸೂಚಿಸಲಾಗುತ್ತದೆ. ನೀವು ಒಬ್ಬರಿಗಿಂತ ಹೆಚ್ಚು ರೆಸಿಪಿಯಂಟ್‌ಗಳನ್ನು ನಮೂದಿಸಿದರೆ (ಉದಾಹರಣೆಗೆ ಗ್ರೂಪ್ ಇ-ಮೇಲ್) ಆ ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಅಲ್ಪವಿರಾಮ ಮತ್ತು ಒಂದು ಸ್ಪೇಸ್ ಅಥವಾ Tab ಕೀಯನ್ನು ಒತ್ತುವ ಮೂಲಕ ಪ್ರತ್ಯೇಕಿಸಬೇಕು.
- Subject ಇ-ಮೇಲ್‌ನ ವಿಷಯಗಳನ್ನು ವಿವರಿಸುವ ಕೆಲವೇ ಶಬ್ದಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. Subject ಇದು ಇ-ಮೇಲ್‌ನ್ನು ತೆರೆಯದೆಯೆ ಮತ್ತು ಇ-ಮೇಲನ್ನು ಪೂರ್ಣವಾಗಿ ಓದದೆ ರೆಸಿಪಿಯಂಟ್‌ಗೆ ಅದು ಯಾವ ವಿಷಯದ ಬಗ್ಗೆ ಮಾಹಿತಿ ಹೊಂದಿದೆ ಎಂದು ತಿಳಿಸುತ್ತದೆ. ಈ ಫೀಲ್ಡ್ ಐಚ್ಛಿಕವಾಗಿದೆ.
- CC(“Carbon Copy”) ಫೀಲ್ಡ್, ನೇರವಾಗಿ ಅಡ್ರೆಸ್ (TO ಫಿಲ್ಡ್‌ನಲ್ಲಿ ಲಿಸ್ಟ್ ಮಾಡಲಾದ) ಮಾಡಲಾಗದ ರೆಸಿಪಿಯಂಟ್‌ಗಳನ್ನು ಸೂಚಿಸಲು ನಿಮಗೆ ಅನುಮತಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ನೀವು Jeff ಗೆ ಇ-ಮೇಲ್‌ನ್ನು ಅಡ್ರೆಸ್ ಮಾಡಬಹುದು ಮತ್ತು CC ಯನ್ನು Linda ಮತ್ತು Steven ಗೆ ಮಾಡಬಹುದು. ಇ-ಮೇಲ್‌ನ್ನು Jeff ಗೆ ಅಡ್ರೆಸ್ ಮಾಡಿದರೂ, Linda ಮತ್ತು Steven ಇವರೂ ಸಹ ಅದರ ಪ್ರತಿಯನ್ನು ಸ್ವೀಕರಿಸುತ್ತಾರೆ ಮತ್ತು ಇ- ಮೇಲ್‌ನ್ನು ಯಾರು ಸ್ವೀಕರಿಸಿದ್ದಾರೆ ಎಂದು ಎಲ್ಲರೂ ನೋಡಬಹುದು. ಈ ಫೀಲ್ಡ್ ಐಚ್ಛಿಕವಾಗಿದೆ.
- BCC (“blind carbon copy”) ಫೀಲ್ಡ್, CC ಫೀಲ್ಡ್‌ನ್ನು ಹೋಲುತ್ತದೆ. ಇದರಲ್ಲಿ ರೆಸಿಪಿಯಂಟ್‌ನ್ನು ರಹಸ್ಯವಾಗಿಡಲಾಗುತ್ತದೆ. ಪ್ರತಿಯೊಬ್ಬ BCC ರೆಸಿಪಿಯಂಟ್ ಇ-ಮೇಲ್‌ನ್ನು ಸ್ವೀಕರಿಸುತ್ತಾನೆ, ಆದರೆ ಪ್ರತಿಯನ್ನು ಬೇರೆ ಯಾರು ಸ್ವೀಕರಿಸಿದ್ದಾರೆ ಎಂದು ನೋಡಲಾಗುವುದಿಲ್ಲ. ಅಡ್ರೆಸ್ ಹೊಂದಿರುವವರು (TO ಫೀಲ್ಡ್‌ನಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡಲಾದ ಯಾರಾದರೂ) ಎಲ್ಲಾ ರೆಸಿಪಿಯಂಟ್‌ಗೆ ಕಾಣಿಸುತ್ತಾರೆ. ಈ ಫೀಲ್ಡ್ ಐಚ್ಛಿಕವಾಗಿದೆ.
- ಅಂತಿಮವಾಗಿ, Message Body ಇದು ನೀವು ನಿಮ್ಮ ಮುಖ್ಯವಾದ ಸಂದೇಶವನ್ನು ಟೈಪ್ ಮಾಡುವ ಸ್ಥಳವಾಗಿದೆ. ಇದು ಕೆಳಭಾಗದಲ್ಲಿ ನಿಮ್ಮ ಸಹಿಯನ್ನೂ ಹೊಂದಿರುತ್ತದೆ ; ಇದು ಕೈಬರಹದ ಪತ್ರವನ್ನು ಹೋಲುತ್ತದೆ.

Fig 12



© NIMI NOT TO BE REPUBLISHED

ವೆಬ್ ವಿನ್ಯಾಸಗೊಳಿಸುವುದು (Web designing)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ವೆಬ್‌ಸೈಟ್ ಮತ್ತು ವೆಬ್‌ಪುಟಗಳನ್ನು ವಿವರಿಸಿ
- ಸ್ಟ್ಯಾಟಿಕ್ ಮತ್ತು ಡೈನಾಮಿಕ್ ವೆಬ್‌ಪುಟಗಳನ್ನು ವಿವರಿಸಿ
- HTML,DHTML, ಮತ್ತು XML ಅನ್ನು ವಿವರಿಸಿ
- ವೆಬ್ ಹೋಸ್ಟಿಂಗ್, ವೆಬ್ ಸರ್ವರ್, ಅಪ್ಲಿಕೇಷನ್ ಸರ್ವರ್ ಮತ್ತು ಡಾಟಾಬೇಸ್ ಸರ್ವರ ಇವುಗಳ ಪರಿಕಲ್ಪನೆಯನ್ನು ವಿವರಿಸಿ.

ಪರಿಚಯ (Introduction)

ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ (WWW) ಅನ್ನು 1990 ರಲ್ಲಿ CERN ಭೌತಶಾಸ್ತ್ರಜ್ಞರಾದ ಟಿಮ್ ಬರ್ನರ್ಸ್- ಲೀ ರಚಿಸಿದರು. 30 ಏಪ್ರಿಲ್ 1993 ರಂದು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಯಾರಾದರೂ ಉಚಿತವಾಗಿ ಬಳಸಬಹುದು ಎಂದು CERN ಘೋಷಿಸಿತು. HTML ಮತ್ತು HTTP ಪರಿಚಯಿಸುವ ಮೊದಲು, ಫೈಲ್ ಟ್ರಾನ್ಸ್‌ಫರ್ ಪ್ರೊಟೊಕಾಲ್ ಮತ್ತು ಗೋಫರ್ ಪ್ರೊಟೊಕಾಲ್‌ಗಳಂತಹ ಇತರ ಪ್ರೊಟೊಕಾಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಪ್ರತ್ಯೇಕ ಫೈಲ್‌ಗಳನ್ನು ಸರ್ವರನಿಂದ ಪಡೆಯಲಾಗುತ್ತಿತ್ತು. ಈ ಪ್ರೊಟೊಕಾಲ್‌ಗಳು ಒಂದು ಸರಳವಾದ ಡೈರೆಕ್ಟರಿ ರಚನೆಯನ್ನು ಹೊಂದಿದ್ದು ಬಳಕೆದಾರರು ಅದನ್ನು ನ್ಯಾವಿಗೇಟ್ ಮಾಡುತ್ತಿದ್ದರು ಮತ್ತು ಡೌನ್‌ಲೋಡ ಮಾಡಲು ಫೈಲ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಿದ್ದರು. ಹೆಚ್ಚಾಗಿ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡದೇ ಸರಳ ಟೆಕ್ಸ್ಟ ಫೈಲ್‌ಗಳಾಗಿ ಪ್ರಸ್ತುತಪಡಿಸಲಾಯಿತು ಅಥವಾ ಅವುಗಳನ್ನು ವರ್ಡ್ ಪ್ರೊಸೆಸರ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗೆ ಎನ್‌ಕೋಡ್ ಮಾಡಲಾಗುತ್ತಿತ್ತು.

ವೆಬ್‌ಸೈಟ್‌ಗಳು(Websites)

Website ನ್ನು Web site, web site ಈ ರೀತಿಯಾಗಿಯೂ ಅಥವಾ ಸರಳವಾಗಿ site ಎಂದೂ ಬರೆಯಬಹುದು. ಇದು ಲಿಖಿತ ಭಾಷೆ, ಚಿತ್ರ, ವಿಡಿಯೊ, ಧ್ವನಿ, ಇತ್ಯಾದಿ ವಿಷಯಗಳನ್ನು ಹೊಂದಿದ ಸಂಬಂಧಿತ ವೆಬ್‌ಪುಟಗಳ ಒಂದು ಗುಂಪಾಗಿದೆ. ವೆಬ್‌ಸೈಟನ್ನು ಕನಿಷ್ಟ ಒಂದು ವೆಬ್ ಸರ್ವರ್‌ನಲ್ಲಿ ಹೊಸ್ಟ ಮಾಡಲಾಗುತ್ತದೆ. ಯಾವುದೇ ವೆಬ್‌ಸೈಟನ್ನು ಯುನಿಫಾರ್ಮ್ ರಿಸೋರ್ಸ್ ಲೊಕೇಟರ್ ಎನ್ನುವ ಇಂಟರ್‌ನೆಟ್ ಅಡ್ರೆಸ್ ಮೂಲಕ ಇಂಟರ್‌ನೆಟ್ ಅಥವಾ ಖಾಸಗಿ ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್‌ಗಳಂತಹ ನೆಟ್ವರ್ಕ್ ಉಪಯೋಗಿಸಿ ಪ್ರವೇಶಿಸಬಹುದು. ಎಲ್ಲ ಸಾರ್ವಜನಿಕ ಪ್ರವೇಶ ಪಡೆಯಬಹುದಾದ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಒಟ್ಟಾರೆಯಾಗಿ ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಅನ್ನು ರೂಪಿಸುತ್ತವೆ.

ವೆಬ್‌ಪುಟವು ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್ ಆಗಿದೆ, ಸಾಮಾನ್ಯವಾಗಿ ಇದನ್ನು ಸರಳವಾದ ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಇದನ್ನು ಹೈಪರ್ ಟೆಕ್ಸ್ಟ ಮಾರ್ಕಪ್ ಲ್ಯಾಂಗ್ವೇಜ್ (HTML) XHTML ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಸೂಚನೆಗಳೊಂದಿಗೆ ವಿಂಗಡಿಸಲಾಗುತ್ತದೆ. ವೆಬ್‌ಪುಟಗಳು ಇತರ ವೆಬ್ ಸೈಟ್‌ಗಳ ಅಂಶಗಳನ್ನು ಸೂಕ್ತವಾದ HTML ಆ್ಯಂಕರ್‌ನೊಂದಿಗೆ ಅಳವಡಿಸಿಕೊಳ್ಳಬಹುದು.

ವೆಬ್‌ಪುಟಗಳನ್ನು ಹೈಪರ್ ಟೆಕ್ಸ್ಟ ಟ್ರಾನ್ಸ್‌ಫರ್ ಪ್ರೊಟೊಕಾಲ್(HTTP) ನೊಂದಿಗೆ ಪ್ರವೇಶಿಸಬಹುದು ಮತ್ತು ಸಾಗಿಸಬಹುದು. ಇವು ವೆಬ್‌ಪುಟದ ವಿಷಯದ ಬಳಕೆದಾರರಿಗೆ ಸುರಕ್ಷತೆಯನ್ನು ಮತ್ತು ಗೌಪ್ಯತೆಯನ್ನು ಒದಗಿಸಲು HTTPS ಗೆ ಎನ್ ಕ್ರಿಪ್ಷನ್ ಸುರಕ್ಷತೆಯನ್ನು ಐಚ್ಛಿಕವಾಗಿ ನೇಮಿಸಬಹುದು. ಬಳಕೆದಾರರ ಅಪ್ಲಿಕೇಷನ್ ಸಾಮಾನ್ಯವಾಗಿ ವೆಬ್ ಬ್ರೌಸರ್, ಪುಟದ ವಿಷಯವನ್ನು , HTML ಮಾರ್ಕ್‌ಅಪ್ ಸೂಚನೆಗಳ ಪ್ರಕಾರ ಕಂಪ್ಯೂಟರ್ ಮಾನಿಟರ್ ಮೇಲೆ ನಿರೂಪಿಸುತ್ತದೆ.

ವೆಬ್‌ಸೈಟ್‌ನ ಪುಟಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ವೆಬ್ ಅಡ್ರೆಸ್ ಎಂದು ಕರೆಯಲ್ಪಡುವ ಸರಳವಾದ ಯುನಿಫಾರ್ಮ್ ರಿಸೋರ್ಸ್ ಲೊಕೇಟರ್ (URL) ನಿಂದ ಪ್ರವೇಶಿಸಬಹುದು. ಪುಟಗಳ URL ಗಳು ಅವುಗಳನ್ನು ಕ್ರಮಾನುಗತವಾಗಿ ಸಂಘಟಿಸುತ್ತವೆ. ಅದಾಗ್ಯೂ ಅವುಗಳ ನಡುವಿನ ಹೈಪರ್‌ಲಿಂಕ್ ಓದುಗರು ಗ್ರಹಿಸಿದ ಸೈಟ್ ಮ್ಯಾಪ್ ಅನ್ನು ತಿಳಿಸುತ್ತದೆ ಮತ್ತು ಓದುಗರಿಗೆ ಸೈಟ್‌ನ್ನು ನ್ಯಾವಿಗೇಟ್(ಸಂಚರಣೆ) ಮಾಡಲು ಮಾರ್ಗದರ್ಶನ ನೀಡುತ್ತದೆ. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಹೋಮ್ ಪುಟವನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ ಮತ್ತು ಸೈಟ್‌ನ ವೆಬ್ ವಿಷಯಕ್ಕೆ ಹೆಚ್ಚಿನ ಲಿಂಕ್‌ಗಳನ್ನು ಮತ್ತು ಪುಟ, ಸಂಪರ್ಕ ಪುಟ ಮತ್ತು ಲಿಂಕ್ ಪುಟಗಳ ಬಗ್ಗೆ ಪೂರಕವಾದ ಲಿಂಕ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಕೆಲವು ವೆಬ್ ಸೈಟ್‌ಗಳಿಗೆ ಅವುಗಳ ಯಾವುದೋ ಒಂದು ಅಥವಾ ಎಲ್ಲ ವಿಷಯಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ಚಂದಾದಾರಿಕೆ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಚಂದಾದಾರಿಕೆ ವೆಬ್‌ಸೈಟ್‌ಗಳ ಉದಾಹರಣೆಗಳು ಅನೇಕ ವ್ಯಾಪಾರದ ಸೈಟ್‌ಗಳು, ಸುದ್ದಿ ವೆಬ್‌ಸೈಟ್‌ಗಳ ಭಾಗಗಳು, ಅಕಾಡಮಿಕ್ ಜರ್ನಲ್ ವೆಬ್‌ಸೈಟ್‌ಗಳು, ಆಟದ ವೆಬ್ ಸೈಟ್‌ಗಳು, ಫೈಲ್ ಶೇರಿಂಗ್ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಇಂಟರ್‌ನೆಟ್ ಫೋರಮ್, ವೆಬ್

© NIMI NOT TO BE REPUBLISHED

ಆಧಾರಿತ ಇ ಮೇಲ್, ಸಾಮಾಜಿಕ ನೆಟ್‌ವರ್ಕಿಂಗ್ ವೆಬ್‌ಸೈಟ್‌ಗಳು, ನೈಜ ಸಮಯದ ಶೇರು ಮಾರುಕಟ್ಟೆಯ ಡಾಟಾ ಒದಗಿಸುವ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಮತ್ತು ಇತರ ಅನೇಕ ಸೇವೆಗಳನ್ನು ಒದಗಿಸುವ ವೆಬ್‌ಸೈಟ್‌ಗಳು (ಉದಾ: ಚಿತ್ರಗಳನ್ನು ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಇತ್ಯಾದಿಗಳನ್ನು , ಸಂಗ್ರಹಿಸುವ ಮತ್ತು/ಅಥವಾ ಹಂಚಿಕೊಳ್ಳುವ ವೆಬ್‌ಸೈಟ್‌ಗಳು)

ಒಂದು ವೆಬ್ ಸೈಟ್ ಈ ತರನಾಗಿರಬಹುದು

- ವೈಯಕ್ತಿಕ ವೆಬ್‌ಸೈಟ್
- ವಾಣಿಜ್ಯ ವೆಬ್‌ಸೈಟ್
- ಇ ಸರ್ಕಾರ
- ಲಾಭರಹಿತ ಸಂಸ್ಥೆ ವೆಬ್‌ಸೈಟ್

ಸ್ಥಿರ ವೆಬ್‌ಸೈಟ್ (Static Website)

ಸ್ಥಿರ ವೆಬ್‌ಸೈಟ್ ಎಂದರೆ ಕ್ಲೈಂಟ್ ವೆಬ್ ಬ್ರೌಸರ್‌ಗೆ ಕಳುಹಿಸಲಾದ ಸ್ವರೂಪದಲ್ಲಿಯೆ ಸರ್ವರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ವೆಬ್‌ಪುಟಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇದನ್ನು ಪ್ರಾಥಮಿಕವಾಗಿ ಹೈಪರ್ ಟೆಕ್ಸ್ಟ ಮಾರ್ಕ್‌ಪ್ ಲ್ಯಾಂಗ್ವೇಜ (HTML) ನಲ್ಲಿ ಸಂಕೇತಗೊಳಿಸಲಾಗಿದೆ.

ವೆಬ್‌ಸೈಟ್‌ಗಳ ಸರಳ ರೂಪಗಳು ಅಥವಾ ಮಾರ್ಕೆಟಿಂಗ್ ಉದಾಹರಣೆಗಳು : ಕ್ಲಾಸಿಕ್ ವೆಬ್‌ಸೈಟ್, ಐದು ಪುಟಗಳ ವೆಬ್‌ಸೈಟ್ ಅಥವಾ ಬ್ರೋಚರ್ ವೆಬ್‌ಸೈಟ್ ಇವು ಸ್ಥಿರ ವೆಬ್‌ಸೈಟ್‌ಗಳಾಗಿವೆ. ಏಕೆಂದರೆ ಇವು ಬಳಕೆದಾರರಿಗೆ ಪೂರ್ವ ನಿರ್ಧಾರಿತ , ಸ್ಥಿರ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತವೆ. ಇದು ಕಂಪನಿಯ ಬಗ್ಗೆ ಮತ್ತು ಅದರ ಉತ್ಪನ್ನಗಳ ಬಗ್ಗೆ ಮತ್ತು ಸೇವೆಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಟೆಕ್ಸ್ಟ, ಫೋಟೊಗಳು, ಎನಿಮೇಷನ್‌ಗಳು, ಆಡಿಯೊ/ವಿಡಿಯೊ ಮತ್ತು ಸಂವಾದಾತ್ಮಕ ಮೆನುಗಳು ಮತ್ತು ನ್ಯಾವಿಗೇಷನ್‌ಗಳ ಮೂಲಕ ಹೊಂದಿರಬಹುದು.

ಈ ರೀತಿಯ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಎಲ್ಲ ನೋಡುವವರಿಗೆ ಒಂದೇ ಮಾಹಿತಿಯನ್ನು ತೋರಿಸುತ್ತವೆ. ಗ್ರಾಹಕರಿಗೆ ಅಥವಾ ಕ್ಲೈಂಟ್‌ಗಳಿಗೆ ಮುದ್ರಿತ ಕರಪತ್ರಗಳನ್ನು ಹಸ್ತಾಂತರಿಸುವಂತೆ, ಸ್ಥಿರ ವೆಬ್ ಸೈಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಥಿರವಾದ , ಪ್ರಮಾಣಿತ ಮಾಹಿತಿಯನ್ನು ವಿಸ್ತೃತವಾದ ಅವಧಿಗೆ ಒದಗಿಸುತ್ತವೆ. ವೆಬ್ ಸೈಟ್ ಮಾಲೀಕರು ನಿಯಮಿತವಾಗಿ ನವಿಕರಣವನ್ನು ಮಾಡಬಹುದಾದರೂ ಇದು ಟೆಕ್ಸ್ಟ , ಫೋಟೋಗಳು, ಮತ್ತು ಇತರ ವಿಷಯಗಳನ್ನು ಕೈಯಿಂದ ಮಾರ್ಪಾಡು ಮಾಡುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ ಮತ್ತು ವೆಬ್ ಸೈಟ್ ವಿನ್ಯಾಸದ ಕೌಶಲ್ಯ ಮತ್ತು ಸಾಫ್ಟವೇರ್‌ನ ಅಗತ್ಯವಿರಬಹುದು.

ಪ್ರಯೋಜನಗಳು (Advantages)

- ಸ್ಥಿರ ವೆಬ್ ಸೈಟ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು ಸುಲಭವಾಗಿದೆ.
- ಇವುಗಳನ್ನು ಶೀಘ್ರವಾಗಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಬಹುದು
- ಎಲ್ಲ ವೆಬ್‌ಪುಟಗಳು ನಿಜವಾಗಿ ಸರ್ವರ್‌ನಲ್ಲಿ ಇರುವುದರಿಂದ ಇವುಗಳನ್ನು ಸರ್ಚ್ ಇಂಜಿನ್‌ಗಳು ಸುಲಭವಾಗಿ ಸೂಚಿಸುತ್ತವೆ. ಇದು ಕ್ರಿಯಾತ್ಮಕ ವೆಬ್ ಸೈಟ್‌ನಲ್ಲಿ ಸಾಧ್ಯವಿಲ್ಲ.

ಅನಾನುಕೂಲಗಳು (Disadvantages)

- ಸ್ಥಿರ ವೆಬ್ ಸೈಟ್‌ಗಳು, ಅನೇಕ ಆನ್‌ಲೈ ನ್ ಸೇವೆಗಳಿಗೆ ಅಗತ್ಯವಿರುವ ಸಂಕೀರ್ಣ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.
- ಇಡೀ ವೆಬ್ ಸೈಟನ್ನು ನವೀಕರಿಸುವುದು ಕಷ್ಟದ ಕೆಲಸವಾಗಿದೆ ಮತ್ತು ಬಹಳ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.
- ಸ್ಥಿರ ವೆಬ್ ಸೈಟ್‌ಗಳಲ್ಲಿ ಡಾಟಾ ಮತ್ತು ವಿನ್ಯಾಸದ ಪ್ರತ್ಯೇಕತೆಯನ್ನು ಒದಗಿಸಲಾಗುವುದಿಲ್ಲ.

ಡೈನಾಮಿಕ್ ವೆಬ್ ಸೈಟ್ (Dynamic Website)

ಡೈ ನಾಮಿಕ್ ವೆಬ್‌ಸೈಟ್ ಕೆಲವು ಮಾನದಂಡಗಳ ಆಧಾರದ ಮೇಲೆ ತನ್ನನ್ನು ತಾನೆ ಆಗಾಗ್ಗೆ ಮತ್ತು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಬದಲಾಯಿಸುವ ಅಥವಾ ಕಸ್ಟಮೈಝ್ ಮಾಡುತ್ತದೆ.

ಡೈ ನಾಮಿಕ್ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಎರಡು ರೀತಿಯ ಡೈನಾಮಿಕ್ ಚಟುವಟಿಕೆಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ : ಕೋಡ್ ಮತ್ತು ಕಂಟೆಂಟ್. ಡೈನಾಮಿಕ್ ಕೋಡ್ ಕಾಣುವುದಿಲ್ಲ ಅಥವಾ ಹಿನ್ನೆಲೆಯಲ್ಲಿರುತ್ತದೆ ಮತ್ತು ಡೈನಾಮಿಕ್ ಕಂಟೆಂಟ್ ಕಾಣುತ್ತದೆ ಮತ್ತು ಸಂಪೂರ್ಣವಾಗಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

ಅನುಕೂಲಗಳು (Advantages)

- ಆನ್‌ಲೈ ನ್ ಸೇವೆಗಳಿಗೆ ಅವಶ್ಯಕವಿರುವ ಸಂಕೀರ್ಣವಾದ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಲ್ಲದು.
- ಇವುಗಳನ್ನು ಸುಲಭವಾಗಿ ನವೀಕರಿಸಬಹುದು.
- ಡಾಟಾ ಮತ್ತು ವಿನ್ಯಾಸದ ಪ್ರತ್ಯೇಕತೆ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಮತ್ತು ವಿಷಯ ಬರಹಗಾರರನ್ನು ಸ್ವತಂತ್ರವಾಗಿ ಕೆಲಸ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಅನಾನುಕೂಲಗಳು (Disadvantages)

- ನಿರ್ಮಿಸಲು ಹೆಚ್ಚು ಸಮಯ ತೆಗೆದುಕೊಳ್ಳಬಹುದು.
- ನಿರ್ಮಿಸಲು ಕಷ್ಟವಾಗಬಹುದು.
- ಡೈನಾಮಿಕ್ ವೆಬ್ ಸೈಟ್‌ಗಳನ್ನು ಸರ್ಚ್ ಇಂಜಿನ್‌ಗಳಿಂದ ಸುಲಭವಾಗಿ ಸೂಚಿಸಲಾಗುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ವೆಬ್ ಸರ್ವರ್‌ನಲ್ಲಿ ನಿಜವಾದ ವೆಬ್ ಪುಟಗಳನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ. ಸರ್ಚ್ ಇಂಜಿನ್ ತಂತ್ರಜ್ಞಾನದಲ್ಲಿನ ನಿರಂತರ ಸುಧಾರಣೆಗಳಿಂದ ಈ ಸಮಸ್ಯೆಯನ್ನು ಹೆಚ್ಚಿನ ಮಟ್ಟದಲ್ಲಿ ತೆಗೆದುಹಾಕಲಾಗಿದೆ ಮತ್ತು ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ನೀವು ಸರ್ಚ್ ಇಂಜಿನ್‌ಗಳು ಅನೇಕ ಡೈನಾಮಿಕ್ ವೆಬ್ ಸೈಟ್‌ಗಳನ್ನು ಉತ್ತಮವಾಗಿ ಸೂಚ್ಯಂಕ ಮಾಡುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಅಂತಹ ಕೆಲವು ವರ್ಗೀಕರಣಗಳು ಹೀಗಿವೆ :

- **ಅಫಿಲಿಯೇಟ್** : ಸಕ್ರೀಯಗೊಳಿಸಲಾದ ಪೋರ್ಟಲ್ ಆಗಿದೆ. ಇದು ತನ್ನ ಕಸ್ಟಮ್ CMS ಮಾತ್ರವಲ್ಲದೇ ಸಿಂಡಿಕೇಟೆಡ್ ವಿಷಯಗಳನ್ನು ಒಪ್ಪಿತ ಶುಲ್ಕಕ್ಕಾಗಿ ಇತರ ವಿಷಯ ಪೂರೈಸುವವರೊಂದಿಗೆ ನಿರೂಪಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಮೂರು ಸಂಬಂಧದ ಟೈರ್‌ಗಳಿವೆ : ಅಫಿಲಿಯೇಟ್ ಏಜನ್ಸೀಸ್ (ಉದಾ : ಕಮೀಷನ್ ಜಂಕ್ಷನ್) , ಜಾಹೀರಾತುದಾರರು (ಉದಾ: eBay) ಮತ್ತು ಗ್ರಾಹಕ (Yahoo!)
- **ಆರ್ಕೈವ್ ಸೈಟ್** : ಇವುಗಳನ್ನು ಅಳಿವಿನ ಅಂಚಿನಲ್ಲಿರುವ ಅಮೂಲ್ಯವಾದ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ವಿಷಯಗಳನ್ನು ಸಂರಕ್ಷಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ : ಇಂಟರ್‌ನೆಟ್ ಆರ್ಕೈವ್, ಇದು 1996 ರಿಂದ ಹಳೆಯ ಮತ್ತು ಹೊಸ ಬಿಲಿಯನ್ ವೆಬ್ ಪುಟಗಳನ್ನು ಸಂರಕ್ಷಿಸಿದೆ; ಮತ್ತು ಗೂಗಲ್ ಗ್ರೂಪ್ಗಳು, 2005 ರ ಪ್ರಾರಂಭದಲ್ಲಿ 845,000,000 ಕ್ಕೂ ಹೆಚ್ಚು Usenet ಸುದ್ದಿ/ ಚರ್ಚಾ ಗುಂಪುಗಳಿಗೆ ಕಳುಹಿಸಲಾದ ಸಂದೇಶಗಳನ್ನು ಸಂರಕ್ಷಿಸಿದೆ.
- **ಬ್ಲಾಗ್ ಸೈಟ್** : ಈ ಸೈಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಆನ್‌ಲೈನ್ ಡೈರಿಗಳನ್ನು, ಕಮೆಂಟ್‌ಗಳನ್ನು ಅಥವಾ ಚರ್ಚೆಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ವೇದಿಕೆಗಳ ವಿಕ್ಷಣೆಗಳನ್ನು (ಉದಾ : ಬ್ಲಾಗರ್, ಕ್ಸಾಂಗಾ)
- **ಕಂಟೆಂಟ್ ಸೈಟ್** : ಈ ಸೈಟ್‌ಗಳು ಅಂತಿಮ ಬಳಕೆದಾರರಿಗೆ ಮೂಲ ವಿಷಯವನ್ನು ರಚಿಸುತ್ತವೆ ಮತ್ತು ಮಾರಾಟ ಮಾಡುತ್ತವೆ (ಉದಾ: Slate, About.com)
- **ಕಾರ್ಪೋರೇಟ ವೆಬ್ ಸೈಟ್** : ವ್ಯಾಪಾರ, ಸಂಸ್ಥೆ , ಅಥವಾ ಸೇವೆ ಇವುಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.
- **ಕಾಮರ್ಸ್ ಸೈಟ್ (ಅಥವಾ ಇ ಕಾಮರ್ಸ್ ಸೈಟ್)** :ಈ ಸೈಟ್‌ಗಳನ್ನು, ಸರಕುಗಳನ್ನು ಖರೀದಿಸಲು ಅಥವಾ ಮಾರಾಟ ಮಾಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಉದಾ : Amazon.com, CSN ಸ್ಟೋರ್ಸ್, ಮತ್ತು Overstock.com
- **ಕಮ್ಯುನಿಟಿ ಸೈಟ್** : ಈ ಸೈಟ್‌ಗಳಲ್ಲಿ ಸಮಾನ ಆಸಕ್ತಿಗಳನ್ನು ಹೊಂದಿರುವ ವ್ಯಕ್ತಿಗಳು ಚಾಟಿಂಗ್ ಮತ್ತು ಸಂದೇಶ ಕಳುಹಿಸುವ ಮೂಲಕ ಪರಸ್ಪರ ಸಂವಹನ ನಡೆಸುತ್ತಾರೆ. ಉದಾ :ಮೈಸ್ಪೇಸ್ ಅಥವಾ ಫೇಸ್‌ಬುಕ್
- **ಸಿಟಿ ಸೈಟ್** : ಈ ಸೈಟ್ ನಿರ್ದಿಷ್ಟ ನಗರ ಅಥವಾ ಪಟ್ಟಣದ ಬಗ್ಗೆ ಮತ್ತು ಆ ನಗರದಲ್ಲಿ ನಡೆಯುವ ಘಟನೆಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ತೋರಿಸುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ನಗರ ಸಭೆ ಇದನ್ನು ರಚಿಸುತ್ತದೆ. ಉದಾ : Richmond.com ಇದು ರಿಚ್ ಮಂಡ್, ವರ್ಜಿನಿಯಾದ ಜಿಯೊಡೊಮೇನ್ ಆಗಿದೆ.
- **ಇನ್‌ಫಾರ್ಮೇಷನ್ ಸೈಟ್** : ಆ ಸೈಟ್‌ಅನ್ನು ಸಂದರ್ಶಿಸುವವರಿಗೆ ಮಾಹಿತಿಯನ್ನು ನೀಡುವ ವಿಷಯವನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಆದರೆ ವಾಣಿಜ್ಯ ಉದ್ದೇಶಗಳಿಗೆ ಅಲ್ಲ. ಉದಾ:RateMyProfessors.com, Free Internet ಮತ್ತು Lexicon Encyclopadia. ಹೆಚ್ಚಿನ ಸರಕಾರಿ, ಶಿಕ್ಷಣಿಕ ಮತ್ತು ಲಾಭರಹಿತ ಸಂಸ್ಥೆಗಳು ಇನ್‌ಫಾರ್ಮೇಷನ್ ಸೈಟ್‌ಅನ್ನು ಹೊಂದಿರುತ್ತವೆ.
- **ಮಿರರ್ ಸೈಟ್** : ಒಂದು ವೆಬ್ ಸೈಟ್‌ನ ಸಂಪೂರ್ಣ ಪುನರುತ್ಪಾದನೆ
- **ನ್ಯೂಸ್ ಸೈಟ್** : ಇನ್‌ಫಾರ್ಮೇಷನ್ ಸೈಟ್‌ನ್ನು ಹೋಲುತ್ತದೆ, ಆದರೆ ಸುದ್ದಿ ಮತ್ತು ಕಮೆಂಟ್‌ಗಳನ್ನು ವಿತರಿಸಲು ಮೀಸಲಾಗಿದೆ.
- **ಪರ್ಸನಲ್ ಹೋಮ್ ಪೇಜ್** : ಒಬ್ಬ ವ್ಯಕ್ತಿ ಅಥವಾ ಕುಟುಂಬದಂತಹ ಸಣ್ಣ ಗುಂಪು ನಡೆಸುವ ಅಥವಾ ಒಬ್ಬ ವ್ಯಕ್ತಿಯು ಸೇರಿಸಲು ಬಯಸುವ ಯಾವುದೇ ವಿಷಯ ಹೊಂದಿರುವ ಸೈಟ್ ಇದಾಗಿದೆ. ಇವುಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಜಿಯೋಸಿಟೀಸ್ (Geocities) ನಂತಹ ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ ಸೇವೆಯನ್ನು ಉಪಯೋಗಿಸಿ ಅಪ್‌ಲೋಡ್ ಮಾಡಲಾಗುತ್ತದೆ.
- **ಫಿಶ್ ಸೈಟ್** : ಈ ವೆಬ್ ಸೈಟನ್ನು ಪಾಸ್‌ವರ್ಡ್ ಮತ್ತು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡನಂತಹ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಮೋಸದಿಂದ ಸ್ವಾಧೀನಪಡಿಸಿಕೊಳ್ಳಲು ರಚಿಸಲಾಗುತ್ತದೆ. ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಸಂವಹನದಲ್ಲಿ ವಿಶ್ವಾಸಾರ್ಹ ವ್ಯಕ್ತಿಯಂತೆ ಅಥವಾ ವ್ಯವಹಾರದಂತೆ (ಉದಾ: Social Security Administration , PayPal) ವೇಷ ಹಾಕುವ ಮೂಲಕ ಮಾಹಿತಿ ಪಡೆದುಕೊಳ್ಳಲಾಗುತ್ತದೆ.
- **ರಾಜಕಿಯ ಸೈಟ್** : ಈ ಸೈಟ್‌ನಲ್ಲಿ ಜನರು ತಮ್ಮ ರಾಜಕೀಯ ಅಭಿಪ್ರಾಯವನ್ನು ನೀಡಬಹುದು.
- **ರೆಟಿಂಗ್ ಸೈಟ್** : ಈ ಸೈಟ್‌ನಲ್ಲಿ ಜನರು ಮುಖ್ಯಾಂಶಗಳನ್ನು ಹೊಗಳಬಹುದು ಅಥವಾ ಹೀಗಳೆಯಬಹುದು
- **ವಿಮರ್ಶೆ ಸೈಟ್** : ಈ ಸೈಟ್‌ನಲ್ಲಿ ಜನರು ಉತ್ಪನ್ನಗಳನ್ನು ಅಥವಾ ಸೇವೆಗಳನ್ನು ವಿಮರ್ಶೆ ಮಾಡಿ ಪೋಸ್ಟ್ ಮಾಡಬಹುದು.
- **ಸ್ಕೂಲ್ ಸೈಟ್** : ಈ ಸೈಟ್‌ನಲ್ಲಿ ಶಿಕ್ಷಕರು, ವಿದ್ಯಾರ್ಥಿಗಳು, ಅಥವಾ ನಿರ್ವಾಹಕರು ಪ್ರಸ್ತುತ ಘಟನೆಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಅಥವಾ ತಮ್ಮ ಶಾಲೆಯ ಬಗ್ಗೆ ಘಟನೆಗಳನ್ನು ಒಳಗೊಂಡ ಮಾಹಿತಿಯನ್ನು ಪೋಸ್ಟ್ ಮಾಡಬಹುದು.
- **ವಿಡಿಯೊ ಹಂಚಿಕೆ** : ಈ ಸೈಟ್ ಬಳಕೆದಾರರಿಗೆ ವಿಡಿಯೊಗಳನ್ನು ಅಪ್‌ಲೋಡ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ ಉದಾ : You-Tube ಮತ್ತು Google Video
- **ಸರ್ಚ್ ಇಂಜಿನ್ ಸೈಟ್** : ಈ ಸೈಟ್ ಸಾಮಾನ್ಯ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಇತರ ಸೈಟ್‌ಗಳನ್ನು ಪುನಃ ಪಡೆಯಲು ಗೇಟ್ ವೆ ಆಗಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ. ವ್ಯಾಪಕವಾಗಿ ಉಪಯೊಗಿಸಲ್ಪಡುವ ಸರ್ಚ್ ಇಂಜಿನ್‌ಗಳು Google, Yahoo, ಮತ್ತು MSN

© NIMI NOT TO BE REPUBLISHED

- **ಶಾಕ್ ಸೈಟ್** : ಈ ಸೈಟ್ ಚಿತ್ರಗಳನ್ನು ಅಥವಾ ಇತರ ವಿಷಯಗಳನ್ನು ಒಳಗೊಂಡಿದೆ .ಇದನ್ನು ಅನೇಕ ವೀಕ್ಷಕರಿಗೆ ಅಪರಾಧದ ಉದ್ದೇಶದಿಂದ ರಚಿಸಲಾಗಿದೆ (ಉದಾ : rotten. com)
- **ವಾರೇಜ್** : ಈ ಸೈಟ್‌ನ್ನು ಹೋಸ್ಟ್ ಮಾಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ ಮತ್ತು ಬಳಕೆದಾರರಿಗೆ ಹಕ್ಕುಸ್ವಾಮ್ಯದ ವಿಷಯಗಳನ್ನು ಅಕ್ರಮವಾಗಿ ಡೌನ್‌ಲೋಡ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.
- **ವೆಬ್ ಪೋರ್ಟಲ್** : ಈ ಸೈಟ್ ಒಂದು ಇಂಟರ್‌ನೆಟ್ ಅಥವಾ ಇಂಟ್ರಾನೆಟ್‌ನಲ್ಲಿರುವ ಇತರ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಹೆಬ್ಬಾಗಿಲನ್ನು ಒದಗಿಸುತ್ತದೆ.

ವೆಬ್ ಪುಟಗಳು (Web Pages)

ವೆಬ್ ಪುಟ ಅಥವಾ ವೆಬ್‌ಪುಟವು ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್ ಅಥವಾ ಮಾಹಿತಿ ಸಂಪನ್ಮೂಲವಾಗಿದೆ. ಇದು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ಗೆ ಸೂಕ್ತವಾಗಿದೆ ಮತ್ತು ಇದನ್ನು ವೆಬ್ ಬ್ರೌಸರ್ ಮೂಲಕ ಪ್ರವೇಶಿಸಬಹುದು. ಇದನ್ನು ಕಂಪ್ಯೂಟರ್ ಪರದೆಯ ಮೇಲೆ ಅಥವಾ ಮೊಬೈಲ್ ಸಾಧನದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಬಹುದು. ಈ ಮಾಹಿತಿಯು ಸಾಮಾನ್ಯವಾಗಿ HTML ಅಥವಾ XHTML ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ ಮತ್ತು ಇತರ ವೆಬ್ ಪುಟಗಳಿಗೆ ಹೈಪರ್ ಟೆಕ್ಸ್ಟ್ ಹೈಪರ್ ಲಿಂಕ್ ಮೂಲಕ ನ್ಯಾವಿಗೇಷನ್ ಬಾರ್ ಅನ್ನು ಒದಗಿಸುತ್ತದೆ. ವೆಬ್ ಪುಟಗಳು ಇತರ ಸಂಪನ್ಮೂಲಗಳಾದ ಕ್ಯಾಸ್ಕೇಡಿಂಗ್ ಸ್ಟೈಲ್ ಶೀಟ್, ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಮತ್ತು ಇಮೇಜಿಂಗ್‌ಗಳನ್ನು ತಮ್ಮ ಅಂತಿಮ ಪ್ರಸ್ತುತಿಗೆ ಬಳಸಿಕೊಳ್ಳುತ್ತವೆ.

ವೆಬ್ ಪುಟಗಳನ್ನು ಸ್ಥಳೀಯ ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಅಥವಾ ದೂರದ ವೆಬ್ ಸರ್ವರ್‌ನಿಂದ ಪಡೆದುಕೊಳ್ಳಬಹುದು. ವೆಬ್ ಸರ್ವರ್ ಖಾಸಗಿ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಮಾತ್ರ ಪ್ರವೇಶಿಸಲು ನಿರ್ಬಂಧಿಸಬಹುದು, ಉದಾ : ಕಾರ್ಪೋರೇಟ್ ಇಂಟ್ರಾನೆಟ್ ಅಥವಾ ಇದು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನಲ್ಲಿ ಪುಟಗಳನ್ನು ಪ್ರಕಟಿಸಬಹುದು. ವೆಬ್ ಪುಟಗಳನ್ನು ವೆಬ್‌ನಿಂದ ವಿನಂತಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ನೀಡಲಾಗುತ್ತದೆ. ವೆಬ್ ಸರ್ವರ್ ಹೈಪರ್‌ಟೆಕ್ಸ್ಟ್ ಟ್ರಾನ್ಸ್ ಫರ್ ಪ್ರೊಟೊಕಾಲ್‌ಅನ್ನು ಬಳಸುತ್ತದೆ.

ವೆಬ್ ಪುಟಗಳು ವೆಬ್ ಸರ್ವರ್‌ನ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಸಂಗ್ರಹವಾದ ಸ್ಥಿರ ಟೆಕ್ಸ್ಟ್‌ನ ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಇತರ ವೆಬ್ ವಿಷಯವನ್ನು ಒಳಗೊಂಡಿರಬಹುದು.(ಸ್ಟಾಟಿಕ್ ವೆಬ್ ಪೇಜ), ಅಥವಾ ವಿನಂತಿಸಿದಾಗ ಸರ್ವರ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್‌ನಿಂದ ನಿರ್ಮಿಸಿರಬಹುದು. (ಡೈನಾಮಿಕ ವೆಬ್ ಪೆಜ್) ಕ್ಲೈಂಟ್ ಬ್ರೌಸರ್‌ನಲ್ಲಿ, ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ವೆಬ್‌ಪುಟಗಳನ್ನು ಬಳಕೆದಾರರ ಇನ್ಪುಟ್‌ಗೆ ಹೆಚ್ಚು ಸ್ಪಂದಿಸುತ್ತವೆ.

ವೆಬ್ ಬ್ರೌಸರ್ (Web Browser)

ವೆಬ್ ಬ್ರೌಸರ್, ಇಂಟರ್‌ನೆಟ್ ಎಕ್ಸ್ಪ್ಲೋರರ್ , ಮೊಝಿಲ್ಲಾ ಫೈರ್ ಫಾಕ್ಸ್, ಗೂಗಲ್ ಕ್ರೋಮ್, ಮತ್ತು ಒಪೆರಾನಂತಹ ಗ್ರಾಫಿಕಲ್ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಅಥವಾ ಲಿಂಕ್ಸ್ (Lynx) ಅಥವಾ ಲಿಂಕ್ಸ್ (Links) , ಕಮಾಂಡ್‌ಲೈನ್ ಇಂಟರ್ ಫೇಸ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ.(ಚಿತ್ರ 1). ಅಂಗವೈಕಲ್ಯ ಹೊಂದಿದ ವೆಬ್ ಬಳಕೆದಾರರು ವೆಬ್ ಪುಟಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ಹೆಚ್ಚಾಗಿ ಸಹಾಯಕ ತಂತ್ರಜ್ಞಾನವನ್ನು ಮತ್ತು ಹೊಂದಾಣಿಕೆಯ ತಂತ್ರಗಳನ್ನು ಬಳಸುತ್ತಾರೆ. ಬಳಕೆದಾರರು ಬಣ್ಣ ಕುರುಡರಾಗಿರಬಹುದು , ಗಾಯದ ಒತ್ತಡದಿಂದಾಗಿ ಮೌಸನ್ನು ಬಳಸದೇ ಇರಬಹುದು ಅಥವಾ ಮೋಟಾರ್ ನ್ಯೂರಾನ್ ಸಮಸ್ಯೆಗಳಿರಬಹುದು, ಕಿವುಡರಾಗಿರಬಹುದು ಮತ್ತು ಆಡಿಯೊವನ್ನು ಶೀರ್ಷಿಕೆ ಮಾಡಬೇಕಾದ ಅಗತ್ಯವಿರುತ್ತದೆ, ಕುರುಡರಾಗಿರಬಹುದು ಮತ್ತು ಸ್ಕ್ರೀನ್ ರೀಡರ್ ಅಥವಾ ಪ್ರದರ್ಶನದ ಅಗತ್ಯವಿರಬಹುದು, ಸ್ಕ್ರೀನ್ ದೊಡ್ಡದನ್ನಾಗಿ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರಬಹುದು ಇತ್ಯಾದಿ. ಅಂಗವಿಕಲ ಮತ್ತು ಸಾಮರ್ಥ್ಯವಿರುವ ಬಳಕೆದಾರರು ಚಿತ್ರಗಳ ಮತ್ತು ಇತರ ಮಾಧ್ಯಮದ ಡೌನ್ ಲೋಡ್‌ನ್ನು ಮತ್ತು ವೀಕ್ಷಣೆಯನ್ನು ನಿಷ್ಕ್ರೀಯಗೊಳಿಸಬಹುದು. ಇದು ಸಮಯವನ್ನು, ನೆಟ್‌ವರ್ಕ್ ಬ್ಯಾಂಡ್ ವಿಡ್ತನ್ನು ಉಳಿಸುತ್ತದೆ ಅಥವಾ ಅವರ ಬ್ರೌಸಿಂಗ್ ಅನುಭವವನ್ನು ಸರಳೀಕರಿಸುತ್ತದೆ.

ಮೊಬೈಲ್ ಸಾಧನಗಳ ಬಳಕೆದಾರರು ಸಾಮಾನ್ಯವಾಗಿ ನಿರ್ಬಂಧಿತ ಪ್ರದರ್ಶನಗಳನ್ನು ಮತ್ತು ಬ್ಯಾಂಡ್ ವಿಡ್ತ್ ಅನ್ನು ಹೊಂದಿರುತ್ತಾರೆ. ಯಾರಾದರೂ, ವೆಬ್ ಪುಟದ ವಿನ್ಯಾಸಕರು ಆಯ್ಕೆಮಾಡಿದ ಫಾಂಟ್, ಫಾಂಟ್ ಸೈಝ್, ಸ್ಟೈಲ್ ಮತ್ತು ಕಲರ್ ಸ್ಕೀಮ್‌ಗಳನ್ನು ಬಳಸದಿರಲು ಮತ್ತು ಪುಟಕ್ಕೆ ತಮ್ಮದೇ ಆದ CSS ಸ್ಟೈಲಿಂಗ್‌ನ್ನು ಅನ್ವಯಿಸಲು ಬಯಸಬಹುದು. ಈ ಎಲ್ಲ ಆಯ್ಕೆಗಳನ್ನು ಗಮನದಲ್ಲಿಟ್ಟುಕೊಂಡು, ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಕನ್ಸೋರ್ಟಿಯಮ್ (W3C) ಮತ್ತು ವೆಬ್ ಅಕ್ಸೆಸ್ಸಿಬಿಲಿಟಿ ಇನಿಷಿಯೇಟಿವ್ (WAI) ಇವು ವೆಬ್ ಪುಟಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಬೇಕೆಂದು ಶಿಫಾರಸು ಮಾಡುತ್ತವೆ.

ಇಂಟರ್‌ನೆಟ್‌ನಿಂದ ಸಾಫ್ಟವೇರನ್ನು ಡೌನ್‌ಲೋಡ್ ಮಾಡುವುದು (Downloading a Software from Internet)

1 ಮೈ ಕಂಪ್ಯೂಟರನ್ನು ತೆರೆದು ಟೆಂಪರರಿ ಫೈಲ್ಸ್ ಫೋಲ್ಡರ್ ಅನ್ನು ರಚಿಸಿ, ನಿಮ್ಮ ಹಾರ್ಡ್ ಡ್ರೈವ್(ಸಾಮಾನ್ಯವಾಗಿ C: ಡ್ರೈವ್) ಮೇಲೆ ಡಬಲ್ ಕ್ಲಿಕ್ ಮಾಡಿ, ನಂತರ File/New/Folder ನ್ನು ಆಯ್ಕೆ ಮಾಡಿ(ಚಿತ್ರ 2 ರಂತೆ)

2 Temporary File ಎಂದು ಟೈಪ ಮಾಡಿ ಮತ್ತು ಚಿತ್ರ 3 ರಂತೆ ಹೆಸರಿಸಿ

3 ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಫ್ರೀವೇರ್ ಸಾಫ್ಟವೇರ್ ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿ "My SQL 5.1 free download " ಎಂದು ಟೈಪ್ ಮಾಡಿ

ಸೂಚನೆ : ಉದಾಹರಣೆಗೆ http://dev.mysql.com/down-

© NIMI NOT TO BE REPUBLISHED

Fig 1

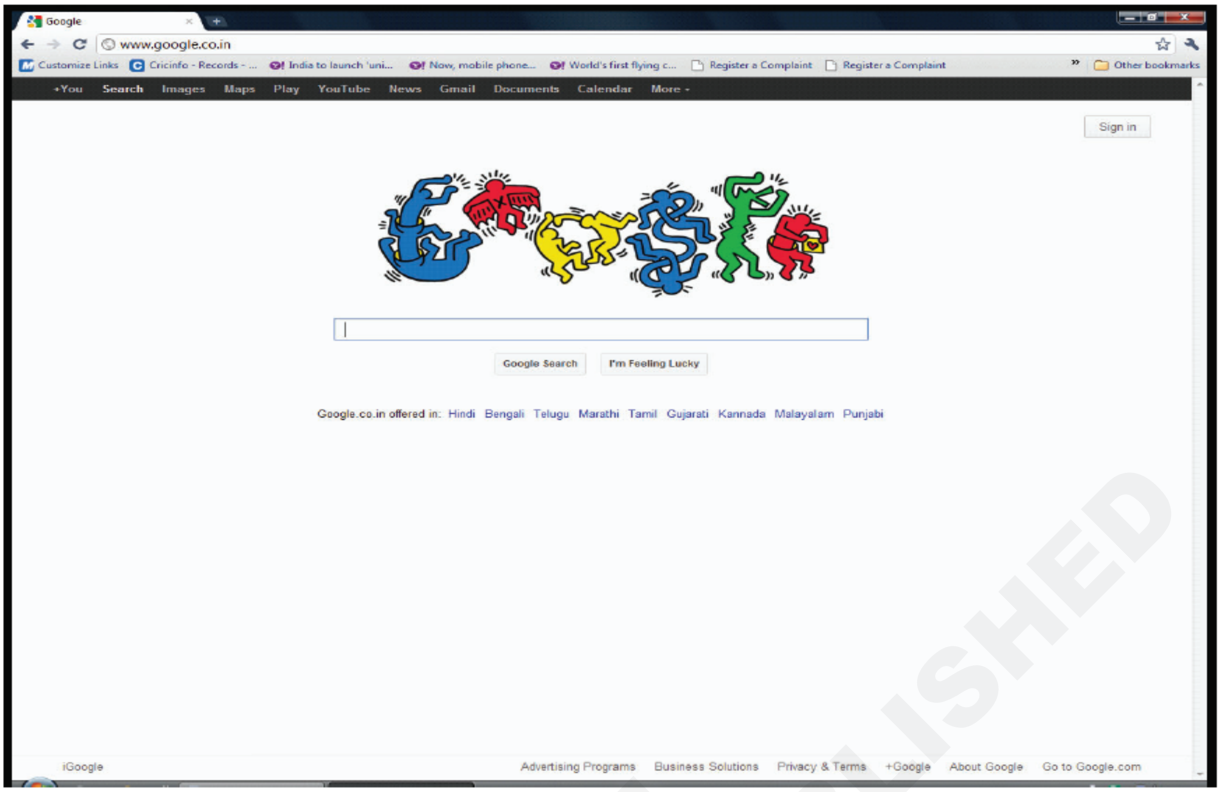


Fig 2

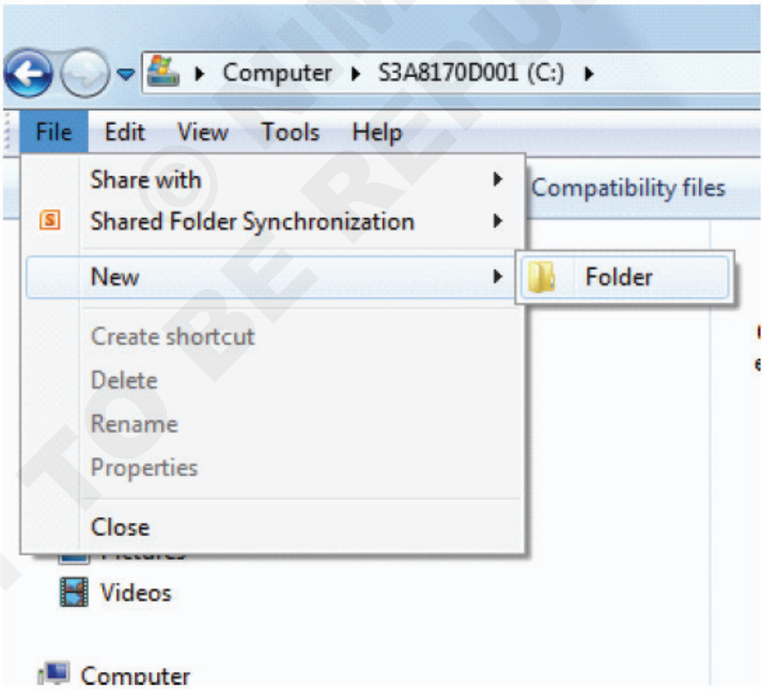


Fig 3

NOT TO BE REPUBLISHED © NIMI

loads/ ಆಯ್ಕೆ ಮಾಡಿ. ನಂತರ ತೆರೆದ ಸೈಟ್‌ನ ಡೌನ್ಲೋಡನಲ್ಲಿ My SQL Community server ನ್ನು ಆಯ್ಕೆ ಮಾಡಿ. (ಚಿತ್ರ 4)

4 ಚಿತ್ರ 5 ರಂತೆ ಹಿಂದಿನ ಆವೃತ್ತಿಯ ಆಯ್ಕೆಯನ್ನು ಹುಡುಕುವುದರಿಂದ My SQL Community server 5.1 ಕ್ಲಿಕ್ ಮಾಡಿ.

5. Windows (x86, 32-bit), MSI Installer Essentials- Recom-

Fig 4

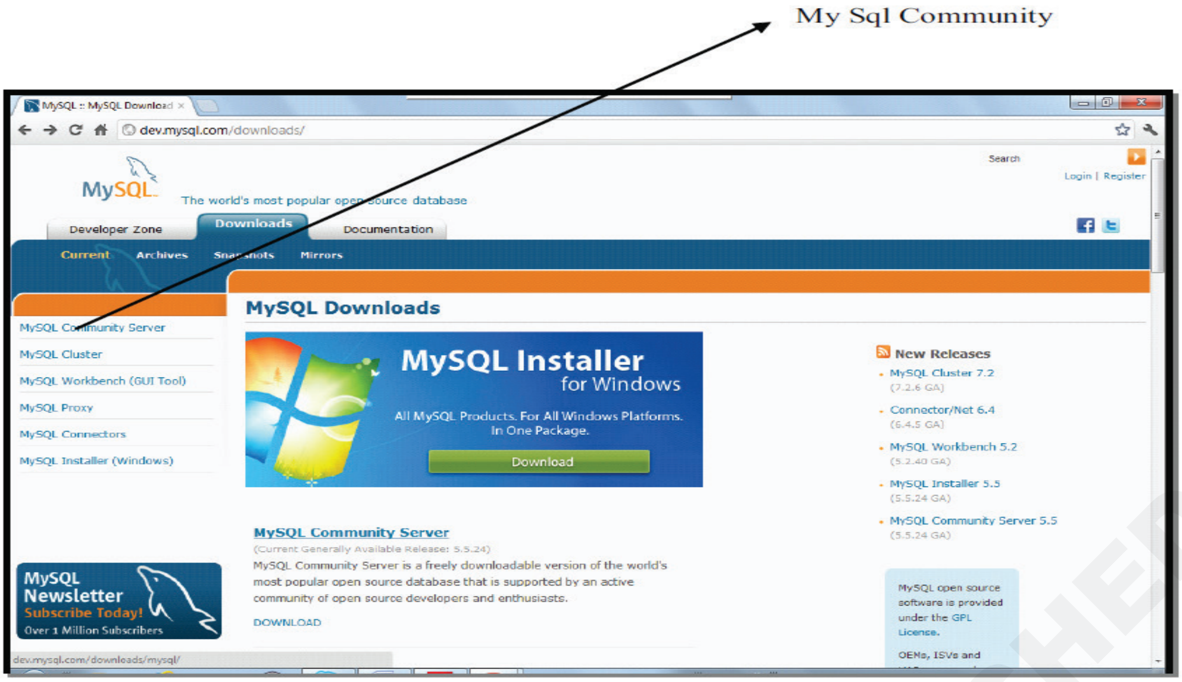
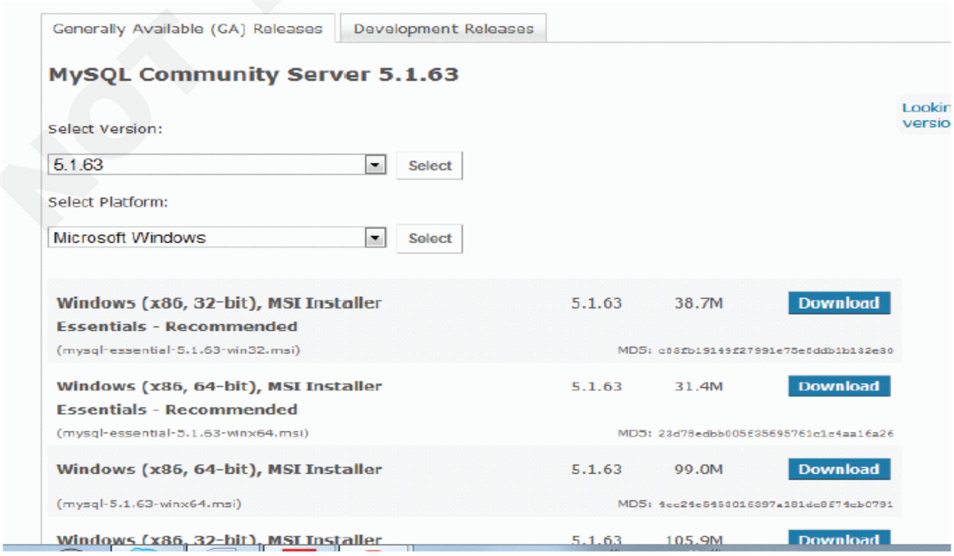


Fig 5

mended ಆಯ್ಕೆ ಮಾಡಿ ಮತ್ತು ಡೌನ್‌ಲೋಡ್ ಕ್ಲಿಕ್ ಮಾಡಿ. (ಚಿತ್ರ 6)

Fig 6



NOT TO BE REPUBLISHED © NIMI

> **ಸೂಚನೆ :** Temporary Folder **ಎಂದು ರಚಿಸಲಾದ ಫೋಲ್ಡರ್‌ನಲ್ಲಿ** My SQL 5.1.63 **ನ್ನು ಸೇವ್ ಮಾಡಿ.**

6. ಡೌನಲೋಡ್ ಮಾಡಲಾದ My SQL 5.1.63 ನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದಕ್ಕಾಗಿ CD ROM ನಲ್ಲಿ ಬರ್ನ್ ಮಾಡಿ.

ವೆಬ್ ಲ್ಯಾಂಗ್ವೇಜ್‌ಗಳು (Web Languages)

ವೆಬ್ ಲ್ಯಾಂಜ್ವೇಜ್‌ಗಳನ್ನು ಮಾರ್ಕ್‌ಅಪ್ ಲ್ಯಾಂಜ್ವೇಜ್ ಎಂದೂ ಕರೆಯಲಾಗುತ್ತದೆ. ಇವುಗಳನ್ನು ಟೆಕ್ಸ್ಟ್‌ನ ಸಂಸ್ಕರಣೆಗೆ, ವ್ಯಾಖ್ಯಾನಿಸಲು, ಮತ್ತು ಪ್ರಸ್ತುತಿಪಡಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಲ್ಯಾಂಜ್ವೇಜ್‌ಗಳು ಟೆಕ್ಸ್ಟ್ ಫೈಲ್‌ನಲ್ಲಿ ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ಗಾಗಿ ಕೋಡ್ ಅನ್ನು ಮತ್ತು ವಿನ್ಯಾಸ ಮತ್ತು ಸ್ಟೈಲ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಸೂಚಿಸಲು ಉಪಯೋಗಿಸುವ ಕೋಡ್‌ನ್ನು ಟ್ಯಾಗ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತೇವೆ.

ನಾಲ್ಕು ವಿಧದ ಮಾರ್ಕ್‌ಪ್ ಲ್ಯಾಂಗ್ವೇಜ್‌ಗಳು

1 BML

2 HTML

3 DHTML

4 XML

BML (Better markup language)

ಬಿಎಂಎಲ್ ಮೂಲಭೂತವಾಗಿ ಒಂದು ಸರಳ ಮ್ಯಾಕ್ರೊ ಲ್ಯಾಂಗ್ವೇಜ ಆಗಿದೆ. ಬಿಎಂಎಲ್‌ನಲ್ಲಿ ಮ್ಯಾಕ್ರೊಗಳನ್ನು ಬ್ಲಾಕ್‌ಗಳು ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಬ್ಲಾಕ್‌ಗಳನ್ನು ಲುಕ್ ಫೈಲ್‌ಗಳಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ ಮತ್ತು ಬಿಎಂಎಲ್ ಫೈಲ್‌ಗಳಲ್ಲಿ ತೆರೆಯಲಾಗುತ್ತದೆ. ಬ್ಲಾಕ್ ಗಳು ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುತ್ತವೆ ಮತ್ತು ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ಹೇಗೆ ಕಳುಹಿಸುತ್ತದೆ ಮತ್ತು ಬ್ಲಾಕ್ ಹೇಗೆ ಅವುಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತದೆ ಎನ್ನುವುದರ ಮೇಲೆ ಇವುಗಳನ್ನು ಹಲವಾರು ವಿಧಗಳಲ್ಲಿ ವಿಂಗಡಿಸಲಾಗಿದೆ.

HTML (Hyper Text markup Language)

ಎಚ್ ಟಿ ಎಂ ಎಲ್ (HTML) ಅಥವಾ ಹೈಪರ್ ಟೆಕ್ಸ್ಟ್ ಮಾರ್ಕ್‌ಪ್ ಲ್ಯಾಂಗ್ವೇಜ್ (Hyper Text markup Language) ಇದು ವೆಬ್‌ನ ಭಾಷೆಯಾಗಿದೆ. ಎಲ್ಲ ವೆಬ್‌ಪುಟಗಳನ್ನು ಎಚ್ ಟಿ ಎಂ ಎಲ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಎಚ್ ಟಿ ಎಂ ಎಲ್ -ಚಿತ್ರಗಳು, ಮಲ್ಟಿ ಮೀಡಿಯಾ ಮತ್ತು ಟೆಕ್ಸ್ಟ್‌ಗಳು ವೆಬ್ ಬ್ರೌಸರಲ್ಲಿ ಹೇಗೆ ಪ್ರದರ್ಶಿಸಲ್ಪಡಬೇಕು ಎನ್ನುವ ವಿಧಾನವನ್ನು ತಿಳಿಸುತ್ತದೆ. ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು ಪರಸ್ಪರ ಸಂಪರ್ಕಿಸಲು ಅಂಶಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ (ಹೈಪರ್ಲಿಂಕ) ಮತ್ತು ವೆಬ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಸಂವಾದಾತ್ಮಕವಾಗಿ ಮಾಡುತ್ತವೆ (ಫಾರ್ಮ್‌ಗಳಂತೆ).

ಎಚ್ ಟಿ ಎಂ ಎಲ್ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ, ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಮಾರ್ಕ್‌ಪ್ ಭಾಷೆಯಾಗಿದೆ. ಈ ಸ್ಟ್ಯಾಂಡರ್ಡ್‌ನ್ನು ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಕನ್ಸೋರ್ಟಿಯಮ್ (W3C) ಅಭಿವೃದ್ಧಿಪಡಿಸಿದೆ. ಇದು ಎಸ್ ಜಿಎಂಎಲ್‌ನ್ನು (Standard Generalized Markup Language) ಆಧರಿಸಿದೆ. ಈ ಭಾಷೆಯು ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ನ ರಚನೆಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಟ್ಯಾಗ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಅಂಶಗಳು(elements) ಮತ್ತು ಟ್ಯಾಗ್‌ಗಳನ್ನು < ಮತ್ತು > ಅಕ್ಷರಗಳಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಡಿ ಎಚ್ ಟಿ ಎಂ ಎಲ್ (DHTML)

ಡೈನಾಮಿಕ್ ಎಚ್ ಟಿ ಎಂ ಎಲ್, ಎಚ್ ಟಿ ಎಂ ಎಲ್‌ನ ಹೊಸ ವಿವರಣೆಯಲ್ಲ, ಆದರೆ ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಎಚ್ ಟಿ ಎಂ ಎಲ್ ಕೋಡ್‌ಗಳನ್ನು ಮತ್ತು ಕಮಾಂಡ್‌ಗಳನ್ನು ನೋಡುವ ಮತ್ತು ನಿಯಂತ್ರಿಸುವ ಹೊಸ ವಿಧಾನವಾಗಿದೆ.

ಡೈನಾಮಿಕ್ ಎಚ್ ಟಿ ಎಂ ಎಲ್ ಬಗ್ಗೆ ಯೋಚಿಸುವಾಗ ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಎಚ್ ಟಿ ಎಂ ಎಲ್‌ನ ಗುಣಗಳನ್ನು ನೆನಪಿಸಿಕೊಳ್ಳುವ ಅವಶ್ಯಕತೆ ಇದೆ. ವಿಶೇಷವಾಗಿ ಸರ್ವರ್‌ನಿಂದ ಒಂದು ಪುಟವು ಲೋಡ್ ಆದರೆ, ಸರ್ವರ್‌ಗೆ ಇನ್ನೊಂದು ವಿನಂತಿ ಬರುವವರೆಗೆ ಅದು ಬದಲಾಗುವುದಿಲ್ಲ. ಡೈನಾಮಿಕ್ ಎಚ್ ಟಿ ಎಂ ಎಲ್ , ಎಚ್ ಟಿ ಎಂ ಎಲ್ ಅಂಶಗಳಿಗಿಂತ ಹೆಚ್ಚಿನ ನಿಯಂತ್ರಣವನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ವೆಬ್ ಸರ್ವರ್‌ಗೆ ಹಿಂತಿರುಗದೇ, ಯಾವುದೇ ಸಮಯದಲ್ಲಿ ಅವುಗಳನ್ನು ಬದಲಾಯಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಡಿ ಎಚ್ ಟಿ ಎಂ ಎಲ್‌ನಲ್ಲಿ ನಾಲ್ಕು ಭಾಗಗಳಿವೆ :

- ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್(DOM) (ಡೆಫಿನಿಷನ್)
- ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳು
- ಕ್ಯಾಸ್ಕೇಡಿಂಗ್ ಸ್ಟೈಲ್ ಶೀಟ್ (CSS)
- XHTML

ಡಾಮ್ (DOM)

ವೆಬ್ ಪುಟದ ಯಾವುದೇ ಭಾಗವನ್ನು ಪ್ರವೇಶಿಸಿ ಅದನ್ನು DHTML ನಿಂದ ಬದಲಾಯಿಸಲು DOM ಅನುಮತಿ ನೀಡುತ್ತದೆ. ವೆಬ್ ಪುಟದ ಪ್ರತಿಯೊಂದು ಭಾಗವು DOM ನಿಂದ ಸೂಚಿಸಲಾಗಿದೆ. ಮತ್ತು ಅದರ ಸ್ಥಿರವಾದ ಹೆಸರಿನ ಸಂಪ್ರದಾಯಗಳನ್ನು ಬಳಸಿ ಅವುಗಳನ್ನು ಪ್ರವೇಶಿಸಬಹುದು ಮತ್ತು ಅವುಗಳ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಬದಲಾಯಿಸಬಹುದು.

ಸ್ಕ್ರಿಪ್ಟ್ ಗಳು (Scripts)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅಥವಾ ಆ್ಯಕ್ಟಿವ್ ಎಕ್ಸ್ ನಲ್ಲಿ ಬರೆಯಲಾದ ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳು DHTML ನ್ನು ಸಕ್ರಿಯಗೊಳಿಸಲು ಉಪಯೋಗಿಸುವ ಎರಡು ಅತಿ ಸಾಮಾನ್ಯ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಗಳಾಗಿವೆ. DOM ನಲ್ಲಿ ಸೂಚಿಸಲಾದ

© NIMI NOT TO BE REPUBLISHED

ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ನೀವು ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತೀರಿ.

ಕ್ಯಾಸ್ಕೇಡಿಂಗ್ ಸ್ಟೈಲ್ ಶೀಟ್ಸ್ (Cascading Style Sheets)

ಒಂದು ವೆಬ್ ಪುಟದ ನೋಟವನ್ನು ಮತ್ತು ಭಾವನೆಯನ್ನು ನಿಯಂತ್ರಿಸಲು CSS ಅನ್ನು DHTML ನಲ್ಲಿ ಉಪಯೋಗಿಸುತ್ತಾರೆ. ಸ್ಟೈಲ್ ಶೀಟ್‌ಗಳು - ಟೆಕ್ಸ್ಟ್‌ನ ಕಲರ್ ಮತ್ತು ಫಾಂಟ್‌ನ್ನು , ಹಿನ್ನೆಲೆ ಬಣ್ಣಗಳು ಮತ್ತು ಚಿತ್ರಗಳು, ಮತ್ತು ಪುಟದಲ್ಲಿ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಸ್ಥಾನವನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ. ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಮತ್ತು DOM ಅನ್ನು ಉಪಯೋಗಿಸಿ, ನಾವು ವಿವಿಧ ಅಂಶಗಳ ಶೈಲಿಯನ್ನು ಬದಲಾಯಿಸಬಹುದು.

XHTML

XHTML ಅಥವಾ HTML 4.x ಸ್ವತಃ ಪುಟವನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಕೆಲಸ ಮಾಡಲು CSS ಮತ್ತು DOM ಗಾಗಿ ಅಂಶಗಳನ್ನು ನಿರ್ಮಿಸುತ್ತದೆ. XHTML, DHTML ಗಿಂತ ವಿಶೇಷವಾಗಿಲ್ಲ, ಆದರೆ ಸರಿಯಾದ XHTML ನ್ನು ಹೊಂದಿರುವುದು ಮುಖ್ಯವಾಗಿದೆ. ಏಕೆಂದರೆ ಕೇವಲ ಬ್ರೌಸರ್‌ಗಿಂತ ಹೆಚ್ಚಿನ ಕೆಲಸಗಳನ್ನು ಇದರಲ್ಲಿ ಮಾಡಬಹುದಾಗಿದೆ.

DHTML ನ ವೈಶಿಷ್ಟ್ಯಗಳು

DHTML ನಲ್ಲಿ ನಾಲ್ಕು ಮುಖ್ಯ ವೈಶಿಷ್ಟ್ಯಗಳಿವೆ

1 ಟ್ಯಾಗ್‌ಗಳನ್ನು ಮತ್ತು ಗುಣಲಕ್ಷಣಗಳನ್ನು ಬದಲಾಯಿಸುವುದು

2 ನೈಜ ಸಮಯದ ಸ್ಥಾನೀಕರಣ (real time positioning)

3 ಡೈನಾಮಿಕ್ ಫಾಂಟ್ಸ್ (ನೆಟ್ ಸ್ಕೇಪ್ ಕಮ್ಯುನಿಕೇಟರ್)

4 ಡಾಟಾ ಬೈಂಡಿಂಗ್ (ಇಂಟರ್‌ನೆಟ್ ಎಕ್ಸ್ ಪ್ಲೋರರ್)

ಟ್ಯಾಗ್‌ಗಳನ್ನು ಮತ್ತು ಗುಣಲಕ್ಷಣಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Changing the tags and properties)

ಇದು DHTML ನ ಸಾಮಾನ್ಯ ಬಳಕೆಯಾಗಿದೆ. ಇದು HTML ಟ್ಯಾಗ್‌ನ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಬದಲಾಯಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದು ಬ್ರೌಸರ್‌ನ ಹೊರಗಿನ ಇವೆಂಟ್‌ನ ಆಧಾರದ ಮೇಲೆ (ಉದಾ : ಮೌಸ್ ಕ್ಲಿಕ್, ಸಮಯ ಅಥವಾ ದಿನಾಂಕ, ಇತ್ಯಾದಿಗಳು) ಬದಲಾಯಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಇದನ್ನು ನಾವು ಪುಟದಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ಪೂರ್ವ ಲೋಡ್ ಮಾಡಲು ಮತ್ತು ಓದುಗರು ನಿರ್ದಿಷ್ಟ ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡದ ಹೊರತು ಅದನ್ನು ಪ್ರದರ್ಶಿಸದೇ ಇರಲು ಉಪಯೋಗಿಸಬಹುದು.

ರಿಯಲ್ ಟೈಮ್ ಪೊಸಿಷನಿಂಗ್ (Real Time Positioning)

ವೆಬ್ ಪುಟದ ಸುತ್ತಲೂ ಚಲಿಸುವ ವಸ್ತುಗಳು, ಚಿತ್ರಗಳು, ಮತ್ತು ಟೆಕ್ಸ್ಟ್‌ಗಳಾಗಿವೆ. ಇದು ನಮಗೆ ಪರಸ್ಪರ ಆಟವಾಡಲು ಅಥವಾ ಪರದೆಯ ಭಾಗವನ್ನು ಎನಿಮೇಟ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಡೈನಾಮಿಕ್ ಫಾಂಟ್ಸ್ (Dynamic Fonts)

ಇದು ನೆಟ್‌ಸ್ಕೇಪ್‌ನ ವೈಶಿಷ್ಟ್ಯವಾಗಿದೆ. ಓದುಗರ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಯಾವ ಫಾಂಟ್ ಇದೆ ಎನ್ನುವುದನ್ನು ತಿಳಿಯದೇ ಇರುವ ವಿನ್ಯಾಸಕಾರರ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ನೆಟ್ ಸ್ಕೇಪ್ ಇದನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದೆ. ಡೈನಾಮಿಕ್ ಫಾಂಟ್‌ಗಳಲ್ಲಿ, ಫಾಂಟ್‌ಗಳನ್ನು ಪುಟದೊಂದಿಗೆ ಎನ್‌ಕೋಡ್ ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ಡೌನ್‌ಲೋಡ್ ಮಾಡಲಾಗುತ್ತದೆ. ಇದರಿಂದಾಗಿ ವಿನ್ಯಾಸಕರು ಪುಟ ಹೇಗೆ ಕಾಣಬೇಕೆಂದು ಉದ್ದೇಶಿಸಿದಂತೆ ಅದು ಕಾಣುತ್ತದೆ.

ಡಾಟಾ ಬೈಂಡಿಂಗ್ (Data Binding)

ಇದು IE ಯ ವೈಶಿಷ್ಟ್ಯವಾಗಿದೆ. ವೆಬ್‌ಸೈಟ್‌ಗಳಿಂದ ಡಾಟಾಬೇಸ್‌ಗಳನ್ನು ಸುಲಭವಾಗಿ ಪ್ರವೇಶಿಸಲು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಇದನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದೆ. ಇದು ಡಾಟಾಬೇಸ್‌ನ್ನು ಪ್ರವೇಶಿಸಲು CGI ಉಪಯೋಗಿಸುವಂತೆಯೆ ಇದೆ, ಆದರೆ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಆ್ಯಕ್ಟಿವ್‌ಎಕ್ಸ್ (ActiveX) ಕಂಟ್ರೋಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

XML

ಎಕ್ಸ್‌ಟೆನ್ಸಿಬಲ್ ಮಾರ್ಕಪ್ ಲ್ಯಾಂಗ್ವೇಜ್ (Extinsible Markaup Language) ಒಂದು ಮಾರ್ಕಪ್ ಭಾಷೆಯಾಗಿದೆ. ಇದು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಮಾನವ ಮತ್ತು ಯಂತ್ರ ಇಬ್ಬರೂ ಓದಬಲ್ಲ ಸ್ವರೂಪಕ್ಕೆ ಎನ್ಕೊಡ್ ಮಾಡಲು ನಿಯಮಗಳ ಗುಂಪನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.ಇದನ್ನುW3CನೀಡಿದXML 1.0ಮಾನದಂಡಗಳಲ್ಲಿ ಮತ್ತು ಇತರ ಅನೇಕ ಸಂಬಂಧಿತ ಮಾನದಂಡಗಳಲ್ಲಿ, ಎಲ್ಲ ಉಚಿತ ಮುಕ್ತ ಮಾನದಂಡಗಳಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ. XML ನ ವಿನ್ಯಾಸದ ಗುರಿಗಳು ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಸರಳತೆ, ಸಾಮಾನ್ಯತೆ, ಮತ್ತು ಉಪಯುಕ್ತತೆಗೆ ಒತ್ತು ನೀಡುವುದು. ಇದು ಜಗತ್ತಿನ ಭಾಷೆಗಳಿಗೆ ಯುನಿಕೋಡ್ ಮೂಲಕ ಬಲವಾದ ಬೆಂಬಲಹೊಂದಿದ ಟೆಕ್ಸ್ಟ್ ರೂಪದ ಡಾಟಾ ಸ್ವರೂಪ ಆಗಿದೆ. XML ನ ವಿನ್ಯಾಸವು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಕೇಂದ್ರೀಕರಿಸಿದರೂ , ಇದನ್ನು ವ್ಯಾಪಕವಾಗಿ ಆರ್ಬಿಟ್ರರಿ ಡಾಟಾ ಸ್ಟ್ರಕ್ಚರ್‌ಗಳನ್ನು ಪ್ರತಿನಿಧಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

HTML ಡಾಕ್ಯುಮೆಂಟ್ ಅನ್ನು ರಚಿಸುವುದು (Creating an HTML document)

ವೆಬ್ ಪುಟವನ್ನು ರಚಿಸಲು ಕೋಡ ಬರೆಯುವುದಕ್ಕೂ ಮೊದಲು, ವೆಬ್ ಪುಟದ ಕಾಣುವಿಕೆಯನ್ನು ಯೋಜಿಸುವುದು ಒಳ್ಳೆಯ ಅಭ್ಯಾಸ. HTML ಡಾಕ್ಯುಮೆಂಟ್ ಎರಡು ಅಂಶಗಳನ್ನು ಹೊಂದಿದೆ.

1 ಡಾಕ್ಯುಮೆಂಟ್ ವಿಷಯ (Document content)

2 ಟ್ಯಾಗ್‌ಗಳು (Tags)

© NIMI NOT TO BE REPUBLISHED

ಡಾಕ್ಯುಮೆಂಟ್ ವಿಷಯ (Document content) ಎಂದರೆ ಬಳಕೆದಾರರು ವೆಬ್ ಪುಟದಲ್ಲಿ ಕಾಣುವ ಮಾಹಿತಿ. ಈ ಮಾಹಿತಿಯು ಟೆಕ್ಸ್ಟ ಆಗಿರಬಹುದು ಅಥವಾ ಗ್ರಾಫಿಕ್ಸ್ ಆಗಿರಬಹುದು.

ಟ್ಯಾಗಗಳು ಡಾಕ್ಯುಮೆಂಟ್ ವಿಷಯ ಹೇಗೆ ಕಾಣಬೇಕೆಂಬುದನ್ನು ನಿಯಂತ್ರಿಸುವ HTML ಕೋಡ್‌ಗಳಾಗಿವೆ. ಟ್ಯಾಗ್‌ಗಳು- ಟೆಕ್ಸ್ಟ ಬೋಲ್ಡ್, ಕಪ್ಪು ಅಥವಾ ನೀಲಿ ಬಣ್ಣ ಅಥವಾ ಫಾಂಟ್ ಟೈಪ್ Times New Roman ಅಥವಾ Arial ಯಾವ ರೀತಿ ಇರಬೇಕು ಎಂದುದನ್ನು ಸೂಚಿಸುತ್ತವೆ.

ನೋಟ್‌ಪ್ಯಾಡ್ ಪ್ರಾರಂಭಿಸಿ

ನೋಟ್‌ಪ್ಯಾಡ್ ಪ್ರಾರಂಭಿಸಲು ಈ ಕೆಳಗಿನ ಆಯ್ಕೆಗಳ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿ:

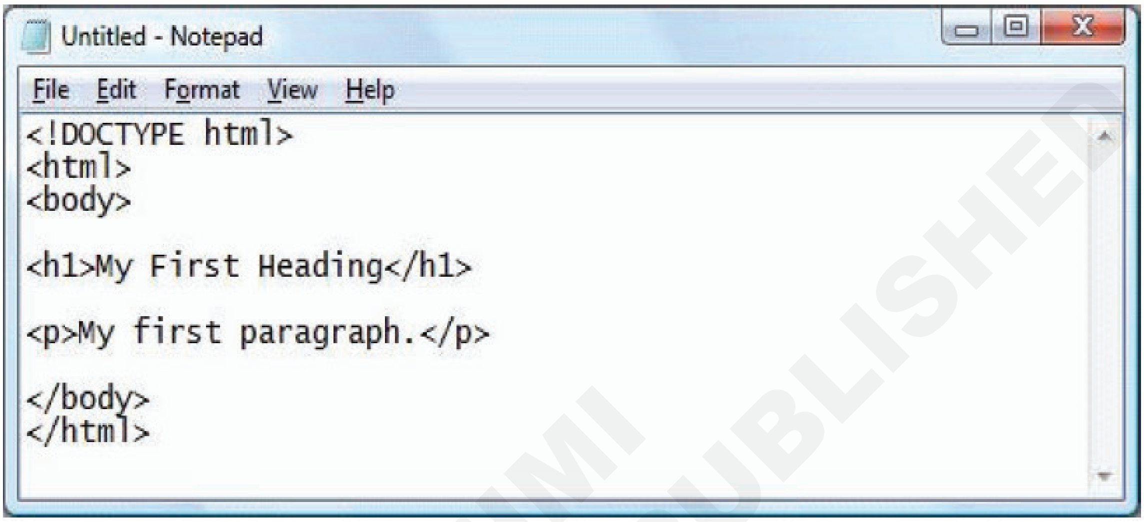
Start

Accessories

Notepad

ನೋಟ್‌ಪ್ಯಾಡ್‌ನಲ್ಲಿ ನಿಮ್ಮ HTML ಅನ್ನು ಎಡಿಟ್ ಮಾಡಿ (ಚಿತ್ರ 7)

Fig 7



© NIMI NOT TO BE REPUBLISHED

ನಿಮ್ಮ ನೋಟ್‌ಪ್ಯಾಡ್‌ನಲ್ಲಿ HTML ಕೋಡ್ ಅನ್ನು ಟೈಪ್ ಮಾಡಿ.

ನಿಮ್ಮ HTML ಸೇವ್ ಮಾಡಿ

ನೋಟ್‌ಪ್ಯಾಡ್‌ನ ಫೈಲ್ ಮೆನುನಲ್ಲಿ Save as... ಆಯ್ಕೆ ಮಾಡಿ

ನೀವು HTML ಫೈಲನ್ನು ಸೇವ್ ಮಾಡುವಾಗ, ಫೈಲ್ ವಿಸ್ತರಣೆಯ ಹೆಸರನ್ನು .htm ಅಥವಾ .html ಎಂದು ಉಪಯೋಗಿಸಿ.

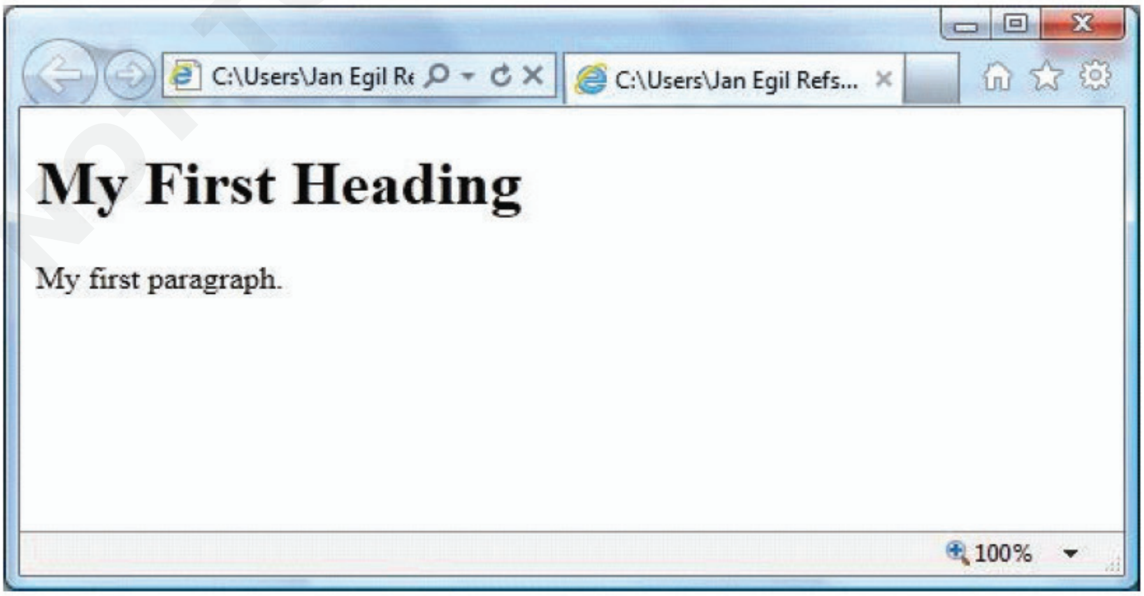
ನಿಮ್ಮ ಫೈಲನ್ನು ನೀವು ನೆನಪಿನಲ್ಲಿ ಇಟ್ಟುಕೊಳ್ಳಲು ಸುಲಭವಾಗುವ ಫೋಲ್ಡರ್ ನಲ್ಲಿ ಸೇವ್ ಮಾಡಿ.

ನಿಮ್ಮ ಬ್ರೌಸರ್ ನಲ್ಲಿ HTML ನ್ನು ರನ್ ಮಾಡಿ.

ನಿಮ್ಮ ವೆಬ್ ಬ್ರೌಸರನ್ನು ಪ್ರಾರಂಭಿಸಿ ಮತ್ತು ನಿಮ್ಮ HTML ಫೈಲನ್ನು - File, Open menu ನಿಂದ ಅಥವಾ ಫೋಲ್ಡರ್ ಅನ್ನು ಬ್ರೌಸ್ ಮಾಡಿ ಮತ್ತು ನಿಮ್ಮ HTML ಫೈಲನ್ನು ಡಬಲ್ ಕ್ಲಿಕ್ ಮಾಡಿ ತೆರೆಯಬಹುದು.

ಫಲಿತಾಂಶವು ಈ ರೀತಿಯಾಗಿರುತ್ತದೆ (ಚಿತ್ರ 8)

Fig 8



ಮಾರ್ಕಪ್ ಲ್ಯಾಂಗ್ವೇಜ್‌ನ ರಚನೆ (Structure of Markup Language)

ಒಂದು HTML ಡಾಕ್ಯುಮೆಂಟ್ ಎರಡು ಮುಖ್ಯ ಭಾಗಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ:

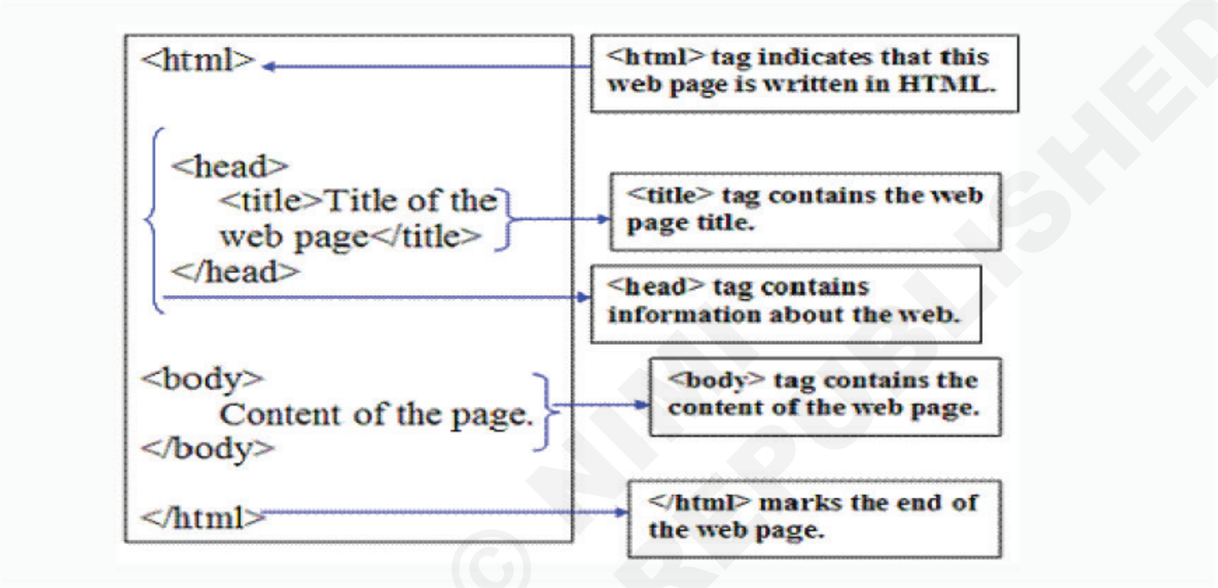
1 ಹೆಡ್ (head) : ಹೆಡ್ ಅಂಶವು ವೆಬ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಶೀರ್ಷಿಕೆ (title) ಮತ್ತು ಮೆಟಾ ಡಾಟಾವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

2 ಬಾಡಿ (body) : ಬಾಡಿ ಅಂಶವು ನೀವು ವೆಬ್ ಪುಟದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲು ಬಯಸುವ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ನಿಮ್ಮ ವೆಬ್ ಪುಟಗಳನ್ನು HTML 4 ರೊಂದಿಗೆ ಹೊಂದಾಣಿಕೆ ಮಾಡಲು HTML ಅಂಶದ ಮೊದಲು ಡಾಕ್ಯುಮೆಂಟ್ ಟೈಪ್ ಡಿಕ್ಲೆರೇಷನ್ (DTD) ಯನ್ನು ಸೇರಿಸುವ ಅಗತ್ಯವಿದೆ. ನೀವು ಹೊಸ ವೆಬ್ ಪುಟವನ್ನು ರಚಿಸಿದಾಗ ಅನೇಕ ವೆಬ್ ಆಥರಿಂಗ್ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು DTD ಮತ್ತು ಮೂಲ ಟ್ಯಾಗ್‌ಗಳನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಸೇರಿಸುತ್ತವೆ.

ವೆಬ್ ಪುಟದಲ್ಲಿ, ಮೊದಲನೇ ಟ್ಯಾಗ್ (ಸಾಮಾನ್ಯವಾಗಿ, <html>) ಮಾರ್ಕಪ್ ಲ್ಯಾಂಗ್ವೇಜ್‌ಗೆ ಅದನ್ನು ಡಾಕ್ಯುಮೆಂಟ್‌ಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿದೆ ಎಂದು ಸೂಚಿಸುತ್ತದೆ. <head> ಟ್ಯಾಗ್ ವೆಬ್‌ಪುಟದ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಕೊನೆಗೆ, <body> ಟ್ಯಾಗ್‌ನಲ್ಲಿ ವಿಷಯವು ಕಂಡುಬರುತ್ತದೆ. (ಚಿತ್ರ 9)

Fig 9



<!DOCTYPE> ಡಿಕ್ಲೆರೆಷನ್

ವೆಬ್‌ನಲ್ಲಿ ಅನೇಕ ವಿಭಿನ್ನ ಡಾಕ್ಯುಮೆಂಟಗಳಿವೆ. ಒಂದು ಬ್ರೌಸರ್ ಡಾಕ್ಯುಮೆಂಟ ಯಾವ ರೀತಿಯದ್ದು ತಿಳಿದಿದ್ದರೆ ಮಾತ್ರ ಅದನ್ನು ಸರಿಯಾಗಿ ಪ್ರದರ್ಶಿಸಬಲ್ಲದು.

HTML ನ ಅನೇಕ ಆವೃತ್ತಿಗಳಿವೆ ಮತ್ತು ಒಂದು ಬ್ರೌಸರ್ ಆ ಪುಟದಲ್ಲಿ ಉಪಯೋಗಿಸಲಾದ ಸರಿಯಾದ ಆವೃತ್ತಿಯನ್ನು ತಿಳಿದಿದ್ದರೆ ಮಾತ್ರ HTML ಪುಟವನ್ನು 100% ಸರಿಯಾಗಿ ಪ್ರದರ್ಶಿಸಬಲ್ಲದು. ಇದಕ್ಕಾಗಿ <!DOCTYPE> ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

<!DOCTYPE> ಇದು HTML ಟ್ಯಾಗ್ ಅಲ್ಲ. ಇದು HTML ನ ಆವೃತ್ತಿಯ ಬಗ್ಗೆ ಬ್ರೌಸರ್‌ಗೆ ಒಂದು ಮಾಹಿತಿ(ಡಿಕ್ಲೇರೇಷನ್).

HTMLನ <head> ಅಂಶ

<head> ಅಂಶವು ಎಲ್ಲ head ಅಂಶಗಳನ್ನು ಸಂಗ್ರಹಿಸಿರುತ್ತದೆ. <head> ನಲ್ಲಿರುವ ಅಂಶಗಳು - ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ, ಬ್ರೌಸರ್‌ಗೆ ಸ್ಟೈಲ್ ಶೀಟ್‌ಗಳನ್ನು ಎಲ್ಲಿ ಹುಡುಕಬೇಕು ಎಂದು ಸೂಚಿಸುತ್ತದೆ, ಮೆಟಾ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುವುದು ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

head ವಿಭಾಗಕ್ಕೆ ಕೆಳಗಿನ ಟ್ಯಾಗ್ ಗಳನ್ನು ಸೇರಿಸಬಹುದು : <title>, <base>, <link>, <meta>, <script> ಮತ್ತು <style>

HTMLನ <title> ಅಂಶ

<title> ಟ್ಯಾಗ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಶೀರ್ಷಿಕೆಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

title ಅಂಶವು ಎಲ್ಲ HTML /XHTML ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಲ್ಲಿ ಅಗತ್ಯವಾಗಿದೆ.

title ಅಂಶವು :

- ಬ್ರೌಸರ್ ಟೂಲ್‌ಬಾರ್‌ನಲ್ಲಿ title ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.
- Favorite ಗೆ ಸೇರಿಸಿದಾಗ ಇದು ಪುಟಕ್ಕೆ title ನ್ನು ಒದಗಿಸುತ್ತದೆ.
- ಸರ್ಚ್ ಇಂಜಿನ್ ಫಲಿತಾಂಶಗಳಲ್ಲಿ, ಪುಟಕ್ಕೆ title ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

HTML ಅಂಶಗಳ ಸಿಂಟ್ಯಾಕ್ಸ್ (HTML Element Syntax)

- ಒಂದು HTML ಅಂಶವು ಪ್ರಾರಂಭಿಕ ಟ್ಯಾಗ /ಆರಂಭಿಕ ಟ್ಯಾಗ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

- ಒಂದು HTML ಅಂಶವು ಅಂತಿಮ ಟ್ಯಾಗ /ಮುಕ್ತಾಯದ ಟ್ಯಾಗ್‌ನಿಂದ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ
- ಪ್ರಾರಂಭ ಮತ್ತು ಅಂತಿಮ ಟ್ಯಾಗ್ ನಡುವೆ ವಿಷಯದ ಅಂಶವಿರುತ್ತದೆ
- ಕೆಲವು HTML ಅಂಶಗಳು ಖಾಲಿ ವಿಷಯವನ್ನು ಹೊಂದಿರುತ್ತವೆ.
- ಖಾಲಿ ಅಂಶಗಳನ್ನು ಪ್ರಾರಮ್ಭಿಕ ಟ್ಯಾಗ್‌ನಲ್ಲಿ ಇಡಲಾಗಿದೆ
- ಹೆಚ್ಚಿನ HTML ಅಂಶಗಳು ಅಟ್ರಿಬ್ಯೂಟ್ (Attribute) ಗಳನ್ನು ಹೊಂದಿರಬಹುದು

HTML ಹೆಡಿಂಗ್ಸ್ (HTML Headings)

HTML ಹೆಡಿಂಗ್‌ಗಳನ್ನು <h1> ದಿಂದ <h6> ಟ್ಯಾಗ್‌ಗಳವರೆಗೆ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ.

ಉದಾಹರಣೆಗಳು (Examples)

```
<html>
<body>
<h1> This is heading 1 </h1>
<h2> This is heading 2 </h2>
<h3> This is heading 3 </h3>
<h4> This is heading 4 </h4>
<h5> This is heading 5 </h5>
<h6> This is heading 6 </h6>
</body>
</html>
```

ಫಲಿತಾಂಶ (Result)

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳು(HTML Paragraphs)

HTML ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳನ್ನು <p> ಟ್ಯಾಗ್‌ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆಗಳು (Examples)

```
<html>
<body>
<p> This is a paragraph </p>
<p> This is a paragraph </p>
<p> This is a paragraph </p>
</body>
</html>
```

ಫಲಿತಾಂಶ (Result)

This is a paragraph

This is a paragraph

This is a paragraph

HTML ಲಿಂಕ್‌ಗಳು (HTML Links)

HTML ಲಿಂಕ್‌ಗಳನ್ನು <a> ಟ್ಯಾಗ್‌ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆಗಳು

```
<html>
<body>
<a href="http://www.facebook.com"> This is a link</a>
</body>
</html>
```

ಫಲಿತಾಂಶ

This is a link

ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡುವುದರಿಂದ ಅದು ಫೇಸ್‌ಬುಕ್ ಲಾಗಿನ್ ಪುಟವನ್ನು ತೋರಿಸುತ್ತದೆ.

HTML ಚಿತ್ರಗಳು (HTML Images)

HTML ಚಿತ್ರಗಳನ್ನು <img> ಟ್ಯಾಗ್‌ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆಗಳು

```
<html>
<body>
<img src="w3schools.jpg" width="104" height="142"/>
</body>
</html>
```

ಫಲಿತಾಂಶ(ಚಿತ್ರ 10)

© NIMI NOT TO BE REPUBLISHED

Fig 10

HTML ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು (HTML Attributes)

- HTML ಅಂಶಗಳು ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು.
- ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು ಒಂದು ಅಂಶದ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತವೆ.
- ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಯಾವಾಗಲೂ ಪ್ರಾರಂಭದ ಟ್ಯಾಗ್‌ನಲ್ಲಿ ಸೂಚಿಸಲಾಗುತ್ತದೆ
- ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು ಹೆಸರು/ಮೌಲ್ಯಗಳ ಜೋಡಿಯಲ್ಲಿ ಇರುತ್ತದೆ ಉದಾ : name=`value`

ಅಟ್ರಿಬ್ಯೂಟ್ ಉದಾಹರಣೆ (Attribute Example)

HTML ಲಿಂಕ್‌ಗಳನ್ನು <a> ಟ್ಯಾಗ್‌ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಲಿಂಕ್ ಅಡ್ರೆಸ್‌ನ್ನು href ಅಟ್ರಿಬ್ಯೂಟ್‌ನಲ್ಲಿ ಸೂಚಿಸಲಾಗುತ್ತದೆ.

```
<html>

<body>

<a href="http://www.yahoo.com"> This is a link</a>

</body>

</html>
```

ಫಲಿತಾಂಶ

This is a link

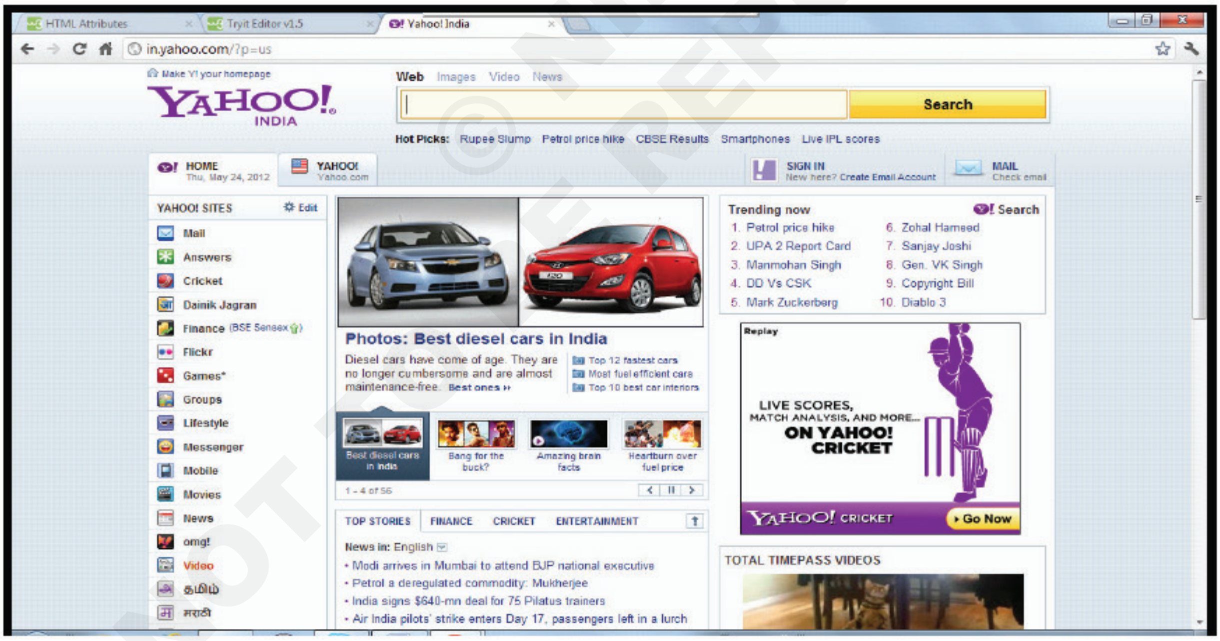
ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡುವುದರಿಂದ ಅದು ಯಾಹೂ ಹೋಮ್ ಪುಟವನ್ನು ತೋರಿಸುತ್ತದೆ.(ಚಿತ್ರ 11)

ಫಾರ್ಮ್ಯಾಟಿಂಗ್ (Formatting)

**ಲೈನ್ ಬ್ರೇಕ್‌ಗಳನ್ನು ರಚಿಸಿ -
 ಅಂಶ**

 ಅಂಶವನ್ನು ಅನುಸರಿಸುವ ಯಾವುದೂ ನಂತರದ ಸಾಲಿನಲ್ಲಿ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಇದು ಖಾಲಿ ಅಂಶದ ಒಂದು ಉದಾಹರಣೆಯಾಗಿದೆ. ಇದಕ್ಕೆ ಮಧ್ಯದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲು ಏನೂ ಇಲ್ಲದಿರುವುದರಿಂದ, ಪ್ರಾರಂಭದ ಮತ್ತು ಅಂತಿಮ ಟ್ಯಾಗ್‌ಗಳ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

Fig 11



ಉದಾಹರಣೆ :

```
Hello <br/>

You come most carefully upon your hour. <br/>

Thanks <br/>

Mehnaz
```

ಫಲಿತಾಂಶ

Hello

You come most carefully upon your hour.

Thanks

Mehnaz

ವಿಷಯವನ್ನು ಮಧ್ಯದಲ್ಲಿರಿಸಲು - <center> ಅಂಶ :

ನೀವು ಯಾವುದೇ ವಿಷಯವನ್ನು ಪುಟದ ಮಧ್ಯದಲ್ಲಿರಿಸಲು ಅಥವಾ

© NIMI NOT TO BE REPUBLISHED

ಯಾವುದೇ ಟೇಬಲ್ ಸೆಲ್‌ನ ಮಧ್ಯದಲ್ಲಿರಿಸಲು <center> ಟ್ಯಾಗ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ:

```
<p> This is not in the center </p>

<center>

<p> This is in the center </p>

</center>
```

ಇದು ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ತೋರಿಸುತ್ತದೆ.

This is not in the center

This is in the center

ಸಾಫ್ಟ ಹೈಫನ್‌ಗಳು (Soft Hyphens)

ಕೆಲವೊಮ್ಮೆ, ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ್ನು ಉತ್ತಮವಾಗಿ ಸಮರ್ಥಿಸಲು ದೀರ್ಘಪದಗಳನ್ನು ಹೈಫನೇಟ್ ಮಾಡಲು ಬ್ರೌಸರಗೆ ಅನುಮತಿ ನೀಡಲು ಬಯಸುತ್ತಿರಿ. ಉದಾಹರಣೆಗೆ , ಕೆಳಗಿನ ಕೋಡ್‌ನ್ನು ಮತ್ತು ಅದರ ಫಲಿತಾಂಶವನ್ನು ಪರಿಗಣಿಸಿ.

```
<p style =`text-align:justify;`> The morbid fear of the
number 13, or triskaidekaphobia, has plagued some im-
portant historic figures like Mahamiya and Nanao,</p>
```

ಇದು ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

ಸಾಫ್ಟ ಹೈಫನ್‌ಗೆ ಉದಾಹರಣೆ - The morbid fear of the number 13, or triskaidekaphobia, has plagued some important historic figures like Mahamiya and Nanao

ಪ್ರಿಸರ್ವ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ - <pre> ಅಂಶ (Preserve Formatting - The <pre> Element)

ಕೆಲವೊಮ್ಮೆ, ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್ HTML ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಇರುವಂತೆಯೆ ಕಾಣಬೇಕೆಂದು ನೀವು ಬಯಸಿದರೆ, ಅದಕ್ಕಾಗಿ ಪ್ರೀ ಫಾರ್ಮ್ಯಾಟೆಡ್ <pre> ಟ್ಯಾಗ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಆರಂಭಿಕ <pre> ಟ್ಯಾಗ್ ಮತ್ತು ಮುಕ್ತಾಯದ </pre> ಟ್ಯಾಗ್ ಮಧ್ಯದಲ್ಲಿನ ಯಾವುದೇ ಟೆಕ್ಸ್ಟ್ ಸೋರ್ಸ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಸ್ವರೂಪವನ್ನು ಸಂರಕ್ಷಿಸುತ್ತದೆ.

```
<pre>

Function testFunction(str Text) {

Alert (str Text)

}

</pre>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ತೋರಿಸುತ್ತದೆ.

```
Function testFunction(str Text) {

Alert (str Text)

}
```

ಹಾರಿಜಾಂಟಲ್ ರೂಲ್‌ಗಳು- <hr /> ಅಂಶ (Horizontal Rules - The <hr /> Element)

ಹಾರಿಜಾಂಟಲ್ ರೂಲ್‌ಗಳನ್ನು , ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿಭಾಗಗಳನ್ನು ಅಡ್ಡವಾಗಿ ವಿಭಜಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. <hr /> ಟ್ಯಾಗ್ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಪ್ರಸ್ತುತ ಸ್ಥಳದಿಂದ ಬಲ ಮಾರ್ಜಿನ್‌ವರೆಗೆ ಒಂದು ರೇಖೆಯನ್ನು ರಚಿಸುತ್ತದೆ. ಮತ್ತು ಅದಕ್ಕೆ ತಕ್ಕಂತೆ ಸಾಲನ್ನು ವಿಭಜಿಸುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ ಈ ಕೆಳಗಿನಂತೆ ಎರಡು ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳ ನಡುವೆ ನೀವು ಒಂದು ಸಾಲನ್ನು ಎಳೆಯಲು ಬಯಸಬಹುದು:

```
<p> This is paragraph one and should be on top </p>

<hr />

<p> This is paragraph two and should be on bottom </p>
```

ಇದು ಈ ಕೆಳಗಿನಂತೆ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ :

This is paragraph one and should be on top

This is paragraph two and should be on bottom

ಮತ್ತೊಮ್ಮೆ <hr /> ಟ್ಯಾಗ್ ಖಾಲಿ ಅಂಶಕ್ಕೆ ಉದಾಹರಣೆಯಾಗಿದೆ. ಇದರ ಮಧ್ಯದಲ್ಲಿ ಏನನ್ನೂ ಪ್ರದರ್ಶಿಸಲು ಇಲ್ಲದಿರುವುದರಿಂದ ಇದಕ್ಕೆ ಆರಂಭಿಕ ಮತ್ತು ಮುಕ್ತಾಯದ ಟ್ಯಾಗ್‌ಗಳ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

> **ಸೂಚನೆ : <hr /> ಅಂಶವು hr ಅಕ್ಷರ ಮತ್ತು ಫಾರ್ವರ್ಡ್ ಸ್ಲ್ಯಾಶ್ ಮಧ್ಯದಲ್ಲಿ ಜಾಗವನ್ನು ಹೊಂದಿದೆ. ನೀವು ಈ ಜಾಗವನ್ನು ಬಿಡದಿದ್ದರೆ, ಹಳೆಯ ಬ್ರೌಸರ್‌ಗಳಿಗೆ ಲೈ ನ್ ಬ್ರೆಕ್‌ನ್ನು ಸಲ್ಲಿಸುವಲ್ಲಿ ತೊಂದರೆಯಾಗುತ್ತದೆ. ನೀವು ಫಾರ್ವರ್ಡ್ ಸ್ಲ್ಯಾಶ್‌ನ್ನು ಬಿಟ್ಟು <hr> ಅನ್ನು ಮಾತ್ರ ಬಳಸಿದರೆ, ಅದು ವ್ಯಾಲಿಡ್ XHTML ಆಗುವುದಿಲ್ಲ.**

ಪ್ರೆಸೆಂಟೇಷನಲ್ ಟ್ಯಾಗ್ (Presentational Tags)

ನೀವು ವರ್ಡ್ ಪ್ರೊಸೆಸರ್ ಬಳಸಿದರೆ, ನೀವು ಬೋಲ್ಡ್, ಇಟ್ಯಾಲಿಸೈಝ್, ಅಥವಾ ಅಂಡರ್‌ಲೈ ನ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯದ ಬಗ್ಗೆ ಪರಿಚಯ ಇರುತ್ತದೆ. ಇವು HTML ಮತ್ತು XHTML ನಲ್ಲಿ ಹತ್ತು ಆಯ್ಕೆಗಳಲ್ಲಿ ಟೆಕ್ಸ್ಟ್ ಯಾವ ರೀತಿ ಕಾಣಿಸಿಕೊಳ್ಳಬೇಕು ಎಂಬುದನ್ನು ಸೂಚಿಸುವ ಮೂರು ಮಾತ್ರ ಆಯ್ಕೆಗಳು ಆಗಿರುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಬೋಲ್ಡ್ ಟೆಕ್ಸ್ಟ್ - **ಎಲಿಮೆಂಟ್** (Bold Text - The Element)

... ಅಂಶದಲ್ಲಿ ಕಂಡುಬರುವ ಯಾವುದನ್ನೂ ಬೋಲ್ಡ್ ಆಗಿ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಇದರಲ್ಲಿ ಬರುವ bold ಶಬ್ದದಂತೆ :

```
<p> The following word uses  a <b> bold </b> typeface
</p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a **bold** typeface.

ಇಟ್ಯಾಲಿಕ್ ಟೆಕ್ಸ್ಟ್ - <i> **ಎಲಿಮೆಂಟ್**(Italic Text - The <i> Element)

<i>... </i> ಅಂಶದಲ್ಲಿ ಕಂಡುಬರುವ ಯಾವುದನ್ನೂ ಇಟ್ಯಾಲಿಸೈಜ್ ಆಗಿ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಇದರಲ್ಲಿ ಬರುವ italicized ಶಬ್ದದಂತೆ :

```
<p> The following word uses a <i> italicized </i> type-
face </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a *italicized* typeface.

ಅಂಡರ್‌ಲೈನ್ಡ್ ಟೆಕ್ಸ್ಟ್ - <u> **ಎಲಿಮೆಂಟ್** (Underlined Text - The <u> Element)

<u>... </u> ಅಂಶದಲ್ಲಿ ಕಂಡುಬರುವ ಯಾವುದನ್ನೂ ಅಂಡರ್‌ಲೈನ್‌ನೊಂದಿಗೆ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಇದರಲ್ಲಿ ಅಂಡರ್‌ಲೈನ್ ಮಾಡಿದ ಶಬ್ದದಂತೆ :

```
<p> The following word uses a <u> underlined </u>
typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a underlined typeface.

ಸ್ಟ್ರೈಕ್ ಟೆಕ್ಸ್ಟ್ - <strike> **ಎಲಿಮೆಂಟ್**

<strike>... </strike> ಅಂಶದಲ್ಲಿ ಕಂಡುಬರುವ ಯಾವುದನ್ನೂ ಸ್ಟ್ರೈಕ್ ಥ್ರೂನೊಂದಿಗೆ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಇದು ಟೆಕ್ಸ್ಟ್ ಮೇಲೆ ತೆಳುವಾದ ರೇಖೆಯಾಗಿದೆ:

```
<p> The following word uses a <strike> strikethrough </
strike> typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a ~~strikethrough~~ typeface.

ಮೊನೊಸ್ಪೇಸ್ಡ್ ಫಾಂಟ್ - <tt> **ಎಲಿಮೆಂಟ್** (Monospaced font - The <tt> Element)

<tt> ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಮೊನೊಸ್ಪೇಸ್ ಫಾಂಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗಿದೆ. ಹೆಚ್ಚಿನ ಫಾಂಟ್‌ಗಳು ಬದಲಾಗುವ ಅಗಲಳತೆಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಏಕೆಂದರೆ ವಿಭಿನ್ನ ಅಕ್ಷರಗಳು ವಿಭಿನ್ನ ಅಗಲವನ್ನು ಹೊಂದಿರುತ್ತವೆ (ಉದಾಹರಣೆಗೆ , m ಅಕ್ಷರವು i ಅಕ್ಷರಕ್ಕಿಂತ ಅಗಲವಾಗಿರುತ್ತದೆ). ಮೊನೊಸ್ಪೇಸ್ ಫಾಂಟ್‌ನಲ್ಲಿ ಪ್ರತಿಯೊಂದು ಅಕ್ಷರವು ಒಂದೇ ಅಗಲವನ್ನು ಹೊಂದಿರುತ್ತವೆ.

```
<p> The following word uses a <tt> monospaced </tt>
typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

ಸೂಪರ್‌ಸ್ಕ್ರಿಪ್ಟ್ ಟೆಕ್ಸ್ಟ್ - <sup> **ಎಲಿಮೆಂಟ್** (Monospaced font - The <tt> Element)

<sup> ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಸೂಪರ್ ಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಉಪಯೋಗಿಸಿರುವ ಫಾಂಟ್ ಗಾತ್ರವು ಅದರ ಸುತ್ತಲೂ ಇರುವ ಅಕ್ಷರಗಳ ಗಾತ್ರದಷ್ಟೇ ಇರುತ್ತದೆ ಆದರೆ ಇತರ ಅಕ್ಷರಗಳ ಮೇಲ್ಭಾಗದಲ್ಲಿ ಅರ್ಧದಷ್ಟು ಎತ್ತರದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

```
<p> The following word uses a <sup> superscript </
sup> typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a ^{superscript} typeface.

ಸಬ್‌ಸ್ಕ್ರಿಪ್ಟ್ ಟೆಕ್ಸ್ಟ್ - <sub> **ಎಲಿಮೆಂಟ್** (Subscript Text - The <sub> Element)

<sub> ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಸೂಪರ್ ಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಉಪಯೋಗಿಸಿರುವ ಫಾಂಟ್ ಗಾತ್ರವು ಅದರ ಸುತ್ತಲೂ ಇರುವ ಅಕ್ಷರಗಳ ಗಾತ್ರದಷ್ಟೇ ಇರುತ್ತದೆ ಆದರೆ ಇತರ ಅಕ್ಷರಗಳ ಮೇಲ್ಭಾಗದಲ್ಲಿ ಅರ್ಧದಷ್ಟು ಎತ್ತರದಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

```
<p> The following word uses a <sub> subscript </sub>
typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a subscript typeface.

ಲಾರ್ಜರ್ ಟೆಕ್ಸ್ಟ್ - <big> **ಎಲಿಮೆಂಟ್** (Larger Text - The <big> Element)

<big> ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಅದರ ಸುತ್ತಲೂ ಇರುವ

© NIMI NOT TO BE REPUBLISHED

ಅಕ್ಷರಗಳ ಗಾತ್ರಗಳಿಗಿಂತ ಒಂದು ಫಾಂಟ್ ಗಾತ್ರವನ್ನು ದೊಡ್ಡದಾಗಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

```
<p> The following word uses a <big> big </big> type-
face </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a big typeface.

ಸ್ಮಾಲರ್ ಟೆಕ್ಸ್ಟ್ - <small> ಎಲೆಮೆಂಟ್ (Smaller Text - The <small> Element)

<small> ಎಲೆಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಅದರ ಸುತ್ತಲೂ ಇರುವ ಅಕ್ಷರಗಳ ಗಾತ್ರಗಳಿಗಿಂತ ಒಂದು ಫಾಂಟ್ ಗಾತ್ರವನ್ನು ಸಣ್ಣದಾಗಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

```
<p> The following word uses a <small> small </small>
typeface </p>
```

ಇದು ಈ ಕೆಳಗಿನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

The following word uses a small typeface.

HTML ನ್ನು CSSನೊಂದಿಗೆ ವಿನ್ಯಾಸಗೊಳಿಸುವುದು (Styling HTML with CSS)

ಸಿಎಸ್‌ಎಸ್‌ನ್ನು ಎಚ್‌ಟಿಎಂಎಲ್ 4 ನೊಂದಿಗೆ ಪರಿಚಯಿಸಲಾಯಿತು. ಇದು ಎಚ್‌ಟಿಎಂಎಲ್ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲು ಉತ್ತಮವಾದ ಮಾರ್ಗವಾಗಿದೆ.

ಸಿಎಸ್‌ಎಸ್‌ನ್ನು ಎಚ್‌ಟಿಎಂಎಲ್ 4 ನೊಂದಿಗೆ, ಈ ಕೆಳಗಿನ ವಿಧಾನಗಳಿಂದ ಸೇರಿಸಬಹುದು:

- ಇನ್‌ಲೈ ನ್ - ಎಚ್‌ಟಿಎಂಎಲ್ ಎಲೆಮೆಂಟ್‌ಗಳಲ್ಲಿ ಸ್ಟೈಲ್ ಅಟ್ರಿಬ್ಯೂಟ್ ಉಪಯೋಗಿಸುವುದು.
- ಇಂಟರ್‌ನಲ್ - <head> ವಿಭಾಗದಲ್ಲಿ <style> ಎಲೆಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು.
- ಎಕ್ಸ್‌ಟರ್ನಲ್ - ಹೊರಗಿನ CSS ಫೈಲನ್ನು ಉಪಯೋಗಿಸುವುದು.

```
<html>
<body style= 'background-color:PowderBlue;'>
<h1> Look! Styles and colors </h1>
<p style='font-family:verdana;color:red;'>
This text is in verdana and red</p>
<p style='font-family:times;color:green;'>
This text is in Times and green</p>
<p style='font-size:30px;'> This text is 30 pixels high</p>
</body>
</html>
```

ಫಲಿತಾಂಶ :

Look! Styles and colors

This text is in verdana and red

This text is in Times and green

This text is 30 pixels high

HTML ಹೈಪರ್‌ಲಿಂಕ್ (ಲಿಂಕ್ಸ್)

ಹೈಪರ್‌ಲಿಂಕ್ (ಲಿಂಕ್) ಒಂದು ಶಬ್ದ, ಶಬ್ದಗಳ ಗುಂಪು, ಅಥವಾ ಚಿತ್ರ ಆಗಿರಬಹುದು. ಇದನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿ ಹೊಸ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಅಥವಾ ಪ್ರಸ್ತುತ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಹೊಸ ವಿಭಾಗಕ್ಕೆ ಹೋಗಬಹುದು.

ವೆಬ್ ಪುಟದಲ್ಲಿ ಲಿಂಕ್ ಮೇಲೆ ನೀವು ಕರ್ಸರ್‌ನ್ನು ಚಲಿಸಿದಾಗ, ಆ್ಯರೋ ಚಿಕ್ಕ ಕೈಯ ಚಿನ್ಹೆಗೆ ತಿರುಗುತ್ತದೆ.

HTML ನಲ್ಲಿ ಲಿಂಕ್‌ಗಳನ್ನು <a> ಟ್ಯಾಗ್ ಬಳಸಿ ಸೂಚಿಸಲಾಗುತ್ತದೆ.

<a> ಟ್ಯಾಗ್‌ನ್ನು ಎರಡು ರೀತಿಯಲ್ಲಿ ಬಳಸಬಹುದು :

1 href ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಬಳಸಿ ಮತ್ತೊಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಲಿಂಕ್ ಮಾಡುವುದು.

2 name ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಬಳಸಿ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಒಳಗೆ ಬುಕ್‌ಮಾರ್ಕ್ ರಚಿಸುವುದು.

ಲಿಂಕ್ ಮಾಡಲು HTML ಕೋಡ್ ಸರಳವಾಗಿದೆ. ಅದು ಈ ರೀತಿಯಾಗಿರುತ್ತದೆ.

```
<a href=`url`> Link text </a>
```

href ಅಟ್ರಿಬ್ಯೂಟ್ ಲಿಂಕ್‌ನ ಡೆಸ್ಟಿನೇಷನ್ ಅನ್ನು ಸೂಚಿಸುತ್ತದೆ.

```
<a href= "http://www.yahoo.com/"> Visit Yahoo </a>
```

ಇದು ಈ ರೀತಿಯಾಗಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ : Visit Yahoo.com

ಈ ಹೈಪರ್‌ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ ಬಳಕೆದಾರರನ್ನು ಯಾಹೂ ಹೋಮ್‌ಪುಟಕ್ಕೆ ಕಳುಹಿಸಲಾಗುತ್ತದೆ.

ಇಲ್ಲಿ Link text ಟೆಕ್ಸ್ಟ್ ಆಗಿರುವ ಅವಶ್ಯಕತೆ ಇಲ್ಲ. ಇದು ಚಿತ್ರ ಆಗಿರಬಹುದು ಅಥವಾ ಯಾವುದೇ ಇತರ HTML ಎಲೆಮೆಂಟ್ ಆಗಿರಬಹುದು.

HTML ಲಿಂಕ್ಸ್ - ಟಾರ್ಗೆಟ್ ಅಟ್ರಿಬ್ಯೂಟ್

ಟಾರ್ಗೆಟ್ ಅಟ್ರಿಬ್ಯೂಟ್, ಲಿಂಕ್ ಆದ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಎಲ್ಲಿ ತೆರೆಯಬೇಕು ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ಲಿಂಕ್ ಆದ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಹೊಸ ಬ್ರೌಸರ ವಿಂಡೋನಲ್ಲಿ ಅಥವಾ ಹೊಸ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ತೆರೆಯಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

 Visit Yahoo !

<html>

<body>

 Visit Yahoo !

<p> If you set the target attribute to `_blank` , the link will open in a new browser window/tab. </p>

</body>

</html>

ಫಲಿತಾಂಶ

Visit Yahoo.com!

ಟಾರ್ಗೆಟ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ನೀವು `_blank` ಗೆ ಸೆಟ್ ಮಾಡಿದರೆ, ಲಿಂಕ್ ಹೊಸ ಬ್ರೌಸರ್ ವಿಂಡೊನಲ್ಲಿ / ಟ್ಯಾಬ್‌ನಲ್ಲಿ ತೆರೆಯುತ್ತದೆ.

HTML ಚಿತ್ರಗಳು - <img> ಟ್ಯಾಗ್ ಮತ್ತು Src ಅಟ್ರಿಬ್ಯೂಟ್

HTML ನಲ್ಲಿ , <img> ಟ್ಯಾಗ್ ಉಪಯೋಗಿಸಿ ಚಿತ್ರಗಳನ್ನು ಸೂಚಿಸಲಾಗುತ್ತದೆ.

<img> ಟ್ಯಾಗ್ ಖಾಲಿಯಾಗಿದೆ. ಎಂದರೆ ಇದು ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಮಾತ್ರ ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಮುಕ್ತಾಯದ ಟ್ಯಾಗನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ.

ಒಂದು ಪುಟದಲ್ಲಿ ಚಿತ್ರವನ್ನು ಪ್ರದರ್ಶಿಸಲು, ನೀವು Src ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಅವಶ್ಯಕತೆ ಇದೆ. Src ಎಂದರೆ `Source`. Src ಅಟ್ರಿಬ್ಯೂಟ್‌ನ ಮೌಲ್ಯವೆಂದರೆ , ನೀವು ಪ್ರದರ್ಶಿಸ ಬಯಸುವ ಚಿತ್ರದ URL ಆಗಿರುತ್ತದೆ.

ಚಿತ್ರವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಸಿಂಟ್ಯಾಕ್ಸ್ :

url, ಚಿತ್ರವು ಸಂಗ್ರಹವಾಗಿರುವ ಸ್ಥಳವನ್ನು ಸೂಚಿಸುತ್ತದೆ. "bamboo.gif" ಹೆಸರಿನ ಚಿತ್ರವು www.w3schools.com ನ images ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಇದೆ. ಇದರ URL : http://www.backgroundlabs.com/index.php?search=bamboo ಆಗಿದೆ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ <img> ಟ್ಯಾಗ್ ಇರುವ ಸ್ಥಳದಲ್ಲಿ ಚಿತ್ರವನ್ನು ಬ್ರೌಸರ್, ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಎರಡು ಪ್ಯಾರಾಗ್ರಾಫ್‌ಗಳ ನಡುವೆ ನೀವು ಇಮೇಜ್ ಟ್ಯಾಗ್‌ನ್ನು ಹಾಕಿದರೆ, ಬ್ರೌಸರ್ ಮೊದಲನೆ ಪ್ಯಾರಾಗ್ರಾಫ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ ನಂತರ ಚಿತ್ರವನ್ನು ತೋರಿಸುತ್ತದೆ ಮತ್ತು ನಂತರ ಎರಡನೆ ಪ್ಯಾರಾಗ್ರಾಫ್ ತೋರಿಸುತ್ತದೆ.

HTML ಚಿತ್ರಗಳು - Alt ಅಟ್ರಿಬ್ಯೂಟ್

ಚಿತ್ರವನ್ನು ಪ್ರದರ್ಶಿಸಲು ಆಗದಿದ್ದರೆ ಅದಕ್ಕೆ ಪರ್ಯಾಯವಾಗಿ Alt ಅಟ್ರಿಬ್ಯೂಟ್ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ.

Alt ಅಟ್ರಿಬ್ಯೂಟ್‌ನ ಮೌಲ್ಯವು ಲೇಖಕರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಟೆಕ್ಸ್ಟ ಆಗಿರುತ್ತದೆ.

Alt ಅಟ್ರಿಬ್ಯೂಟ್, ಕೆಲವು ಕಾರಣಗಳಿಂದ ಬಳಕೆದಾರರು ಚಿತ್ರವನ್ನು ನೋಡಲು ಸಾಧ್ಯವಾಗದಿದ್ದರೆ ಅದಕ್ಕೆ ಪರ್ಯಾಯ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. (ನಿಧಾನ ಸಂಪರ್ಕದ ಕಾರಣದಿಂದ , src ಅಟ್ರಿಬ್ಯೂಟ್‌ನಲ್ಲಿನ ದೋಷ, ಅಥವಾ ಬಳಕೆದಾರರು ಸ್ಕ್ರೀನ್ ರೀಡರ್ ಬಳಸುತ್ತಿದ್ದರೆ).

HTML ಚಿತ್ರಗಳು - ಚಿತ್ರದ ಹೈಟ್(ಎತ್ತರ) ಮತ್ತು ವಿಡ್ತ್ (ಅಗಲ)ವನ್ನು ಹೊಂದಿಸುವುದು

ಹೈಟ್ ಮತ್ತು ವಿಡ್ತ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಚಿತ್ರದ ಹೈಟ್(ಎತ್ತರ) ಮತ್ತು ವಿಡ್ತ್ (ಅಗಲ)ವನ್ನು ಸೂಚಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಅಟ್ರಿಬ್ಯೂಟ್ ಮೌಲ್ಯಗಳನ್ನು ಡೀಫಾಲ್ಟ್ ಆಗಿ ಪಿಕ್ಸೆಲ್‌ಗಳಲ್ಲಿ ಸೂಚಿಸಲಾಗುತ್ತದೆ.:

ಸೂಚನೆ : ಒಂದು ಚಿತ್ರದ ಹೈಟ್ ಮತ್ತು ವಿಡ್ತ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಸೂಚಿಸುವುದು ಒಳ್ಳೆಯ ಅಭ್ಯಾಸ ಈ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಸೆಟ್ ಮಾಡಿದರೆ, ಪುಟವನ್ನು ಲೋಡ್ ಮಾಡಿದಾಗ ಚಿತ್ರಕ್ಕಾಗಿ ಅವಶ್ಯಕತೆ ಇರುವ ಸ್ಥಳವನ್ನು ಕಾಯ್ದಿರಿಸಲಾಗುತ್ತದೆ. ಅದಾಗ್ಯೂ , ಈ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳಿಲ್ಲದೆ ಬ್ರೌಸರ್‌ಗೆ ಚಿತ್ರದ ಗಾತ್ರವು ತಿಳಿಯುವುದಿಲ್ಲ. ಇದರ ಪರಿಣಾಮವಾಗಿ, ಪುಟವು ಲೋಡ್ ಆಗುವಾಗ ಅದರ ವಿನ್ಯಾಸ ಬದಲಾಗುತ್ತದೆ. (ಚಿತ್ರಗಳು ಲೋಡ್ ಆಗುವಾಗ).

ಒಂದು HTML ಫೈಲ್ ಹತ್ತು ಚಿತ್ರಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, - ಆ ಪುಟವನ್ನು ಪ್ರದರ್ಶಿಸಲು ಹನ್ನೊಂದು ಫೈಲ್‌ಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಚಿತ್ರಗಳು ಲೋಡ್ ಆಗಲು ಸಮಯ ತೆಗೆದುಕೊಳ್ಳುತ್ತವೆ, ಅದ್ದರಿಂದ ನನ್ನ ಉತ್ತಮ ಸಲಹೆ : ಚಿತ್ರಗಳನ್ನು ಎಚ್ಚರಿಕೆಯಿಂದ ಬಳಸಿ.

ವೆಬ್‌ಪುಟವು ಲೋಡ್ ಆದಾಗ, ಆ ಕ್ಷಣದಲ್ಲಿ ಬ್ರೌಸರ್, ವೆಬ್ ಸರ್ವರ್‌ನಿಂದ ಚಿತ್ರವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅದನ್ನು ಪುಟದಲ್ಲಿ ಸೇರಿಸುತ್ತದೆ. (ಚಿತ್ರ 12). ಅದ್ದರಿಂದ ವೆಬ್ ಪುಟಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ಚಿತ್ರಗಳು ಅದೇ ಸ್ಥಳದಲ್ಲಿ ಇರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ , ಇಲ್ಲದಿದ್ದರೆ ನಿಮ್ಮ ವೀಕ್ಷಕರಿಗೆ ಲಿ

NOT TO BE REPUBLISHED © NIMI

ಂಕ್ ತಪ್ಪಿದ ಐಕಾನ್‌ನ್ನು ತೋರಿಸಲಾಗುತ್ತದೆ. ಬ್ರೌಸರ್‌ಗೆ ಚಿತ್ರವು ಕಂಡು ಹಿಡಿಯಲು ಆಗದಿದ್ದರೆ ತಪ್ಪಾದ ಲಿಂಕ್ ಐಕಾನ್‌ನ್ನು ತೋರಿಸಲಾಗುತ್ತದೆ.

```
<html>

<body>

<h2> Friendship Card </h2>

<img border="0" src="/images/Rose.jpg" alt="Rose" width="304" height="228`"/>

</body>

</html>
```

HTML ಟೇಬಲ್‌ಗಳು (HTML Tables)

ಟೇಬಲ್‌ಗಳನ್ನು <table> ಟ್ಯಾಗ್‌ನೊಂದಿಗೆ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಟೇಬಲ್‌ಗಳನ್ನು ರೋ(rows)ಗಳಲ್ಲಿ ವಿಭಜಿಸಲಾಗಿದೆ. (<tr> ಟ್ಯಾಗ್ ಉಪಯೋಗಿಸಿ) , ಮತ್ತು ಪ್ರತಿ ರೋ ಅನ್ನು ಡಾಟಾ ಸೆಲ್‌ಗಳಲ್ಲಿ ವಿಭಜಿಸಲಾಗುತ್ತದೆ. (<td> ಟ್ಯಾಗ್ ಉಪಯೋಗಿಸಿ). td ಎಂದರೆ ಟೇಬಲ್ ಡಾಟಾ ಮತ್ತು ಇದು ಡಾಟಾ ಸೆಲ್‌ನ ವಿಷಯವನ್ನು ಹೊಂದಿರುತ್ತದೆ. <td> ಟ್ಯಾಗ್ - ಟೆಕ್ಸ್ಟ್ , ಲಿಂಕ್‌ಗಳು, ಚಿತ್ರಗಳು, ಲಿಸ್ಟ್ ಗಳು, ಚಿತ್ರಗಳು, ಫಾರ್ಮ್‌ಗಳು, ಇತರ ಟೇಬಲ್ ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಿರಬಹುದು.

ಟೇಬಲ್ ಉದಾಹರಣೆ (Table Example)

```
<table border="1">

<tr>

<td> row 1 , cell 1</td>

<td> row 1 , cell 2</td>

</tr>

<tr>

<td> row 2 , cell 1</td>

<td> row 2 , cell 2</td>

</tr>

</table>
```

ಮೇಲಿನ HTML ಕೋಡ್, ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

row 1 , cell 1	row 1 , cell 2
row 2 , cell 1	row 2 , cell 2

HTML ಟೇಬಲ್‌ಗಳು ಮತ್ತು ಬಾರ್ಡರ್ ಆಟ್ರಿಬ್ಯೂಟ್ (HTML Tables and the Border Attribute)

ನೀವು ಬಾರ್ಡರ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಸೂಚಿಸದಿದ್ದರೆ, ಟೇಬಲ್‌ನ್ನು ಬಾರ್ಡರ್‌ಗಳಿಲ್ಲದೇ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಕೆಲವೊಮ್ಮೆ ಇದು ಉಪಯುಕ್ತವಾಗಬಹುದು , ಆದರೆ ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ ನಮಗೆ ಬಾರ್ಡರಗಳ ಕಾಣುವಿಕೆ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

ಒಂದು ಟೇಬಲ್‌ನ್ನು ಬಾರ್ಡರ್‌ನೊಂದಿಗೆ ಪ್ರದರ್ಶಿಸಲು, ಬಾರ್ಡರ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಸೂಚಿಸಿ:

Fig 12

© NIMI NOT TO BE REPUBLISHED

```
<table border=`1`>
<tr>
<td> row 1 , cell 1</td>
<td> row 1 , cell 2</td>
</tr>
</table>
```

HTML ಟೇಬಲ್ ಹೆಡರ್‌ಗಳು (HTML Table Headers)

ಟೇಬಲ್‌ನಲ್ಲಿ ಹೆಡರ್ ಮಾಹಿತಿಯನ್ನು <th> ಟ್ಯಾಗ್‌ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಎಲ್ಲ ಮುಖ್ಯ ಬ್ರೌಸರ್‌ಗಳು <th> ಎಲಿಮೆಂಟ್ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಬೋಲ್ಡ್ ಮತ್ತು ಸೆಂಟರ್ಡ್ ಆಗಿ ತೋರಿಸುತ್ತವೆ.

```
<table border="1">
<tr>
<th> Header 1</th>
<th> Header 2</th>
</tr>
<tr>
<td> row 1 , cell 1</td>
<td> row 1 , cell 2</td>
</tr>
<tr>
<td> row 2 , cell 1</td>
<td> row 2 , cell 2</td>
</tr>
</table>
```

ಮೇಲಿನ HTML ಕೋಡ್, ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

Header 1	Header 2
row 1 , cell 1	row 1 , cell 2
row 2 , cell 1	row 2 , cell 2

HTML ಅನ್‌ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್‌ಗಳು (HTML Unordered Lists)

ಅನ್ ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್ ಟ್ಯಾಗ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ಲಿಸ್ಟ್ ಐಟಂ ಟ್ಯಾಗ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ಲಿಸ್ಟ್ ಐಟಂಗಳನ್ನು ಬುಲೆಟ್(ಸಾಮಾನ್ಯವಾಗಿ ಸಣ್ಣ ಕಪ್ಪು ವೃತ್ತಗಳು) ಗಳಿಂದ ಗುರುತಿಸಲಾಗುತ್ತದೆ.

```
<ul>
<li> Coffee </li>
<li> Milk </li>
</ul>
```

ಮೇಲಿನ HTML ಕೋಡ್, ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

- Coffee
- Milk

HTML ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್‌ಗಳು (HTML Ordered Lists)

ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್ ಟ್ಯಾಗ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ಲಿಸ್ಟ್ ಐಟಂ ಟ್ಯಾಗ್‌ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ಲಿಸ್ಟ್ ಐಟಂಗಳನ್ನು ಸಂಖ್ಯೆಗಳಿಂದ ಗುರುತಿಸಲಾಗುತ್ತದೆ.

```
<ol>
<li> Coffee </li>
<li> Milk </li>
</ol>
```

ಮೇಲಿನ HTML ಕೋಡ್, ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

1 Coffee

2 Milk

HTML ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್‌ಗಳು (HTML Definition Lists)

ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್, ಐಟಂ ಗಳ ಲಿಸ್ಟ್ ಆಗಿದ್ದು, ಪ್ರತಿಯೊಂದು ಐಟಂ ನ ವಿವರಣೆ ಹೊಂದಿರುತ್ತವೆ.

<dl> ಟ್ಯಾಗ್ ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

<dl> ಟ್ಯಾಗ್‌ನ್ನು , <dt> ಟ್ಯಾಗ್ (ಲಿಸ್ಟ್ ನಲ್ಲಿರುವ ಐಟಂ ಅನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ) ಮತ್ತು <dd> ಟ್ಯಾಗ್ (ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಐಟಂ ಅನ್ನು ವಿವರಿಸುತ್ತದೆ) ನೊಂದಿಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

```
<dl>
<dt> Coffee </dt>
<dd> - black hot drink </dd>
<dt> Milk</dt>
<dd> - white cold drink </dd>
</dl>
```

© NIMI NOT TO BE REPUBLISHED

ಮೇಲಿನ HTML ಕೋಡ್, ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

Coffee - black hot drink

Milk - white cold drink

> **ಸೂಚನೆ : ಲಿಸ್ಟ್ ಐಟಂ ನಲ್ಲಿ ನೀವು ಟೆಕ್ಸ್ಟ್‌ನ್ನು, ಲೈ ನ್ ಬ್ರೆಕ್‌ನ್ನು, ಚಿತ್ರಗಳನ್ನು, ಲಿಂಕ್‌ಗಳನ್ನು, ಇತರ ಲಿಸ್ಟ್‌ಗಳನ್ನು ಮುಂತಾದವುಗಳನ್ನು ಹಾಕಬಹುದು.**

HTML ಲಿಸ್ಟ್ ಟ್ಯಾಗ್‌ಗಳು (HTML List Tags)

ಟ್ಯಾಗ್ (Tag)	ವಿವರಣೆ (Description)
	ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
	ಅನ್‌ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
	ಲಿಸ್ಟ್ ಐಟಂನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<dl>	ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್ ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<dt>	ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿನ ಐಟಂನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<dd>	ಡೆಫಿನಿಷನ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿನ ಐಟಂನ ವಿವರಣೆಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ

HTML ಎಲಿಮೆಂಟ್ ಗಳನ್ನು <div> ಮತ್ತು <span> ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಒಟ್ಟಾಗಿ ಸೇರಿಸಬಹುದು.

HTML ಬ್ಲಾಕ್ ಎಲಿಮೆಂಟ್‌ಗಳು (HTML Block Elements)

ಹೆಚ್ಚಿನ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಬ್ಲಾಕ್ ಲೆವೆಲ್ ಎಲಿಮೆಂಟ್‌ಗಳು ಅಥವಾ ಇನ್‌ಲೈನ್ ಎಲಿಮೆಂಟ್‌ಗಳು ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಬ್ಲಾಕ್ ಲೆವೆಲ್ ಎಲಿಮೆಂಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಿದಾಗ ಹೊಸ ಸಾಲಿನಿಂದ ಪ್ರಾರಂಭವಾಗುವುದು(ಮತ್ತು ಮುಕ್ತಾಯವಾಗುವುದು).

ಉದಾಹರಣೆಗೆ : <h1>, <p>, <ul>, <table>

HTML ಇನ್‌ಲೈನ್ ಎಲಿಮೆಂಟ್‌ಗಳು

ಇನ್‌ಲೈನ್ ಎಲಿಮೆಂಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಹೊಸ ಲೈನ್‌ನಲ್ಲಿ ಪ್ರಾರಂಭವಾಗುವುದಿಲ್ಲ.

ಉದಾಹರಣೆಗಳು : <b> , <td> , <a> , <img>

HTML <div> ಎಲಿಮೆಂಟ್

HTML <div> ಎಲಿಮೆಂಟ್ ಒಂದು ಬ್ಲಾಕ್ ಎಲಿಮೆಂಟ್ ಆಗಿದೆ. ಇದನ್ನು ಇತರ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಗುಂಪುಮಾಡಲು ಸಂಗ್ರಾಹಕವನ್ನಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

<div> ಎಲಿಮೆಂಟ್ ವಿಶೇಷ ಅರ್ಥವನ್ನು ಹೊಂದಿಲ್ಲ. ಇದು ಬ್ಲಾಕ್ ಲೆವೆಲ್ ಎಲಿಮೆಂಟ್ ಆಗಿರುವುದರಿಂದ, ಬ್ರೌಸರ್ ಇದರ ಮೊದಲು ಮತ್ತು ನಂತರ ಲೈನ್ ಬ್ರೆಕ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

CSSನೊಂದಿಗೆ ಉಪಯೋಗಿಸಿದಾಗ, <div> ಎಲಿಮೆಂಟ್‌ನ್ನು ದೊಡ್ಡ ವಿಷಯದ ಬ್ಲಾಕ್ ಗಳಿಗೆ ಸ್ಟೈಲ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಸೆಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

<div> ಎಲಿಮೆಂಟ್‌ನ ಇನ್ನೊಂದು ಸಾಮಾನ್ಯ ಉಪಯೋಗವೆಂದರೆ, ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸುವುದು. ವಿನ್ಯಾಸವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಹಳೆಯ ಮಾರ್ಗವನ್ನು, ಟೇಬಲ್‌ಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಬದಲಾಯಿಸಲಾಗುತ್ತದೆ. ಟೇಬಲ್‌ಗಳನ್ನು ಬಳಸುವುದು, <table> ಎಲಿಮೆಂಟ್‌ನ ಸರಿಯಾದ ಬಳಕೆಯಲ್ಲ. <table> ಎಲಿಮೆಂಟ್‌ನ ಉದ್ದೇಶ ಟ್ಯುಬ್ಯುಲರ್ ಡಾಟಾವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು.

HTML <span> ಎಲಿಮೆಂಟ್

HTML <span> ಎಲಿಮೆಂಟ್ ಇದು ಒಂದು ಇನ್‌ಲೈನ್ ಎಲಿಮೆಂಟ್. ಇದನ್ನು ಟೆಕ್ಸ್ಟ್‌ಗೆ ಸಂಗ್ರಾಹಕವನ್ನಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

<span> ಎಲಿಮೆಂಟ್ ವಿಶೇಷ ಅರ್ಥವನ್ನು ಹೊಂದಿಲ್ಲ.

CSSನೊಂದಿಗೆ ಉಪಯೋಗಿಸಿದಾಗ, <span> ಎಲಿಮೆಂಟ್‌ನ್ನು ಟೆಕ್ಸ್ಟ್‌ನ ಭಾಗಗಳಿಗೆ ಸ್ಟೈಲ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಸೆಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

HTML ಗ್ರೂಪಿಂಗ್ ಟ್ಯಾಗ್‌ಗಳು

ಟ್ಯಾಗ್ (Tag)	ವಿವರಣೆ (Description)
<div>	Div ಅನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<span>	span ಅನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ

HTML ಲೇಔಟ್ (HTML Layout)

ವೆಬ್‌ಸೈಟ್ ಲೇಔಟ್‌ಗಳು

ಹೆಚ್ಚಿನ ವೆಬ್ ಸೈಟ್‌ಗಳು ತಮ್ಮ ವಿಷಯವನ್ನು ಅನೇಕ ಕಾಲಮ್‌ಗಳಲ್ಲಿ ಇರಿಸಿವೆ. (ಪತ್ರಿಕೆ ಅಥವಾ ಸಮಾಚಾರ ಪತ್ರಿಕೆಗಳಂತೆ ಫಾರ್ಮಾಟ್ ಮಾಡಲಾಗಿದೆ).

ಅನೇಕ ಕಾಲಮ್‌ಗಳನ್ನು <div> ಅಥವಾ <table> ಉಪಯೋಗಿಸಿ ರಚಿಸಲಾಗುತ್ತದೆ. CSS ಅನ್ನು ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸ್ಥಾನದಲ್ಲಿರಿಸಲು ಅಥವಾ ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್ ರಚಿಸಲು, ಅಥವಾ ಪುಟಗಳ ವರ್ಣರಂಜಿತ ನೋಟಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

HTML ಲೇಔಟ್ - <div> ಎಲಿಮೆಂಟ್‌ಗಳು

<div> ಎಲಿಮೆಂಟ್ ಇದು ಬ್ಲಾಕ್ ಲೆವೆಲ್ ಎಲಿಮೆಂಟ್ ಆಗಿದೆ. ಇದನ್ನು HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಗ್ರೂಪ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯು ಅನೇಕ ಕಾಲಮ್ ಲೇಔಟನ್ನು ರಚಿಸಲು ಐದು <div> ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಹಿಂದಿನ ಉದಾಹರಣೆಯು ನೀಡಿದ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ

```
<div>
<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:500px">
<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0"> Main Title of WebPage</h1></div>
<divid="menu" style="background-color:#FFD700;height:200px;width:100px;float:left;">
<b> Menu </b><br/>
HTML <br/>
CSS<br/>
JAVASCRIPT </div>
<div id ="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">
ಇಲ್ಲಿ ವಿಷಯ ಬರುತ್ತದೆ </div>
<div id ="footer" style="background-color:#FFA500;clear:both;text-align:center;">
</div>
</div>
</body>
</html>
```

ಫಲಿತಾಂಶ (ಚಿತ್ರ 13)

Fig 13

© NIMI NOT TO BE REPUBLISHED

HTML **ಫಾರ್ಮಗಳು** (HTML Forms)

ಡಾಟಾವನ್ನು ಸರ್ವರ್‌ಗೆ ರವಾನಿಸಲು HTML ಫಾರ್ಮಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಒಂದು ಫಾರ್ಮ್ ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್‌ಗಳಾದ ಟೆಕ್ಸ್ಟ್ ಫೀಲ್ಡ್‌ಗಳು, ಚೆಕ್ ಬಾಕ್ಸ್ , ರೇಡಿಯೊ-ಬಟನ್‌ಗಳು, ಸಬ್‌ಮಿಟ್ ಬಟನ್ ಗಳು ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಒಂದು ಫಾರ್ಮ್ ಸೆಲೆಕ್ಟ್ ಲಿಸ್ಟ್‌ಗಳು, ಟೆಕ್ಸ್ಟ್ ಏರಿಯಾ, ಫೀಲ್ಡ್ ಸೆಟ್, ಲೆಗ್ ಎಂಡ್ ಮತ್ತು ಲೇಬಲ್ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸಹ ಒಳಗೊಂಡಿರುತ್ತದೆ.

HTML ಫಾರ್ಮ್‌ನ್ನು ರಚಿಸಲು <form> ಟ್ಯಾಗ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

```
<form>
ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್‌ಗಳು
</form>
```

HTML **ಫಾರ್ಮಗಳು - ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್**

ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್ ಮುಖ್ಯವಾದ ಫಾರ್ಮ್ ಎಲಿಮೆಂಟ್ ಆಗಿದೆ.

ಬಳಕೆದಾರರ ಮಾಹಿತಿಯನ್ನು ಆಯ್ಕೆ ಮಾಡಲು ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಅಟ್ರಿಬ್ಯೂಟ್ ವಿಧವನ್ನು ಆಧರಿಸಿ, ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್ ಹಲವು ರೀತಿಯಲ್ಲಿ ಬದಲಾಗುತ್ತದೆ. ಇನ್ಪುಟ್ ಎಲಿಮೆಂಟ್ ಇವುಗಳಲ್ಲಿ ಯಾವುದೂ ಆಗಿರಬಹುದು : ಟೆಕ್ಸ್ಟ್ ಫೀಲ್ಡ್ , ಚೆಕ್ ಬಾಕ್ಸ್ , ರೇಡಿಯೊ ಬಟನ್, ಸಬ್ಮಿಟ್ ಬಟನ್ ಮುಂತಾದವುಗಳು.

ಹೆಚ್ಚು ಉಪಯೋಗಿಸಲ್ಪಡುವ ಇನ್ಪುಟ್ ವಿಧಗಳನ್ನು ಕೆಳಗೆ ವಿವರಿಸಲಾಗಿದೆ.

ಟೆಕ್ಸ್ಟ್ ಫೀಲ್ಡ್ (Text Fields)

<input type ="text"/> defines a one-line input field that a user can enter text into :

<form>

First name : <input type="text" name ="firstname" />

Last name : <input type="text" name =" lastname" />

< /form>

ಬ್ರೌಸರ್‌ನಲ್ಲಿ HTML ಕೋಡ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

First name : [ ]

Last name : [ ]

> **ಸೂಚನೆ : ಫಾರ್ಮ್ ಸ್ವತಃ ಕಾಣುವುದಿಲ್ಲ. ಟೆಕ್ಸ್ಟ್ ಫೀಲ್ಡ್‌ನ ಡೀಫಾಲ್ಟ್ ಅಗಲ 20 ಅಕ್ಷರಗಳಿರುತ್ತದೆ.**

ಪಾಸ್‌ವರ್ಡ್ ಫೀಲ್ಡ್ (Password Field)

<input type="password" /> defines a password field :

<form>

Password: <input type="password" name ="pwd" />

</form>

ಬ್ರೌಸರ್ ನಲ್ಲಿ HTML ಕೋಡ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

Password: [ ]

> **ಸೂಚನೆ : ಪಾಸ್‌ವರ್ಡ್‌ನಲ್ಲಿರುವ ಅಕ್ಷರಗಳು ಜಟಿಲವಾಗಿರುತ್ತವೆ. (ನಕ್ಷತ್ರದ ಅಥವಾ ಚಿಕ್ಕ ವೃತ್ತದ ಚಿನ್ಹೆಗಳಾಗಿರುತ್ತವೆ).**

ರೇಡಿಯೋ ಬಟನ್‌ಗಳು (Radio Buttons)

<input type=" radio " /> ಇದು ರೇಡಿಯೋ ಬಟನ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ರೇಡಿಯೋ ಬಟನ್ ಕೆಲವೇ ಸೀಮಿತ ಸಂಖ್ಯೆಯ ಆಯ್ಕೆಗಳಲ್ಲಿ ಒಂದನ್ನು ಮಾತ್ರ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಬಳಕೆದಾರರಿಗೆ ಅವಕಾಶ ನೀಡುತ್ತದೆ.:

< form>

<input type=" radio "` name = "sex" value="male"/> Male

<input type= "radio" name ="sex" value="female"/> Female

< /form>

ಬ್ರೌಸರ್‌ನಲ್ಲಿ, HTML ಕೋಡ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

☐ Male

☐ Female

ಚೆಕ್ ಬಾಕ್ಸ್‌ಗಳು (Checkboxes)

<input type="checkbox" /> ಇದು ಚೆಕ್‌ಬಾಕ್ಸ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ಚೆಕ್ ಬಾಕ್ಸ್ ಬಳಕೆದಾರರಿಗೆ ಸೀಮಿತ ಸಂಖ್ಯೆಯ ಆಯ್ಕೆಗಳಲ್ಲಿ ಒಂದು ಅಥವಾ ಅದಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಆಯ್ಕೆಗಳನ್ನು ಮಾಡಲು ಅವಕಾಶ ನೀಡುತ್ತದೆ.

<form>

<input type="checkbox" name ="vehicle" value="bike"/> I have a bike

<input type="checkbox" name ="vehicle" value="car"/> I have a car

</form>

ಬ್ರೌಸರ್‌ನಲ್ಲಿ, HTML ಕೋಡ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

☐ I have abike

☐ I have a car

ಸಬ್ಮಿಟ್ ಬಟನ್ (Password Field)

<input type="submit" /> ಇದು ಸಬ್ಮಿಟ್ ಬಟನ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

ಡಾಟಾವನ್ನು ಸರ್ವರ್‌ಗೆ ಕಳುಹಿಸಲು ಸಬ್ಮಿಟ್ ಬಟನ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಫಾರ್ಮ್‌ನ ಎಕ್ಷನ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ನಲ್ಲಿ ಸೂಚಿಸಿದ ಪುಟಕ್ಕೆ ಡಾಟಾವನ್ನು ಕಳುಹಿಸಲಾಗುತ್ತದೆ. ಎಕ್ಷನ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ನಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಫೈಲ ಸಾಮಾನ್ಯವಾಗಿ ಪಡೆದ ಇನ್ಪುಟ್ ಮೇಲೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ.

<form>

<form name="input" action="html_form_action.asp" method="get">

Username : <input type="text" name ="Submi"/>

<input type="submit" value ="Submit"/>

</form>

ಬ್ರೌಸರ್ ನಲ್ಲಿ, HTML ಕೋಡ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

Username : [ ] Submit

ಟೆಕ್ಸ್ಟ್ ಫೀಲ್ಡ್‌ನಲ್ಲಿ ನೀವು ಕೆಲವು ಅಕ್ಷರಗಳನ್ನು ಟೈಪ್ ಮಾಡಿ ಮತ್ತು ಸಬ್ಮಿಟ್ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ, ಬ್ರೌಸರ್ ನಿಮ್ಮ ಇನ್ಪುಟ್‌ನ್ನು

© NIMI NOT TO BE REPUBLISHED

``html_form_action.asp ಎನ್ನುವ ಪುಟಕ್ಕೆ ಕಳುಹಿಸುತ್ತದೆ. ಈ ಪುಟವು ಸ್ವೀಕರಿಸಿದ ಇನ್ಪುಟ್‌ನ್ನು ನಿಮಗೆ ತೋರಿಸುತ್ತದೆ.

HTML ಐಫ್ರೇಮ್‌ಗಳು (HTML Iframes)

ಐಫ್ರೇಮ್‌ನ್ನು ಸೇರಿಸಲು ಸಿಂಟ್ಯಾಕ್ಸ್

<iframe src="URL">< /iframe>

URL ಪ್ರತ್ಯೇಕ ಪುಟದ ಸ್ಥಳವನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಐಫ್ರೇಮ್ - ಹೈಟ್(ಎತ್ತರ) ಮತ್ತು ವಿಡ್ತ್(ಅಗಲ)ಹೊಂದಿಸುವುದು

ಹೈಟ್(ಎತ್ತರ) ಮತ್ತು ವಿಡ್ತ್(ಅಗಲ) ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು, ಐಫ್ರೇಮ್‌ನ ಹೈಟ್(ಎತ್ತರ) ಮತ್ತು ವಿಡ್ತ್(ಅಗಲ) ಸೂಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಅಟ್ರಿಬ್ಯೂಟ್ ಮೌಲ್ಯಗಳನ್ನು ಡೀಫಾಲ್ಟ್ ಆಗಿ ಪಿಕ್ಸೆಲ್‌ಗಳಲ್ಲಿ ಸೂಚಿಸಲಾಗುತ್ತದೆ, ಆದರೆ ಅವು ಪರ್ಸೆಂಟ್‌ನಲ್ಲಿಯೂ ಸಹ ಇರಬಹುದು.(80% ನಂತೆ).

ಉದಾಹರಣೆ

```
<DOCTYPE html>
<html>
<body>
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
</body>
</html>
```

ಇದರ ಫಲಿತಾಂಶ ಚಿತ್ರ 14 ರಂತೆ ಕಾಣುತ್ತದೆ.

ಐಫ್ರೇಮ್ - ಬಾರ್ಡರ್‌ನ್ನು ತೆಗೆಯುವುದು.

ಫ್ರೇಮ ಬಾರ್ಡರ್ ಅಟ್ರಿಬ್ಯೂಟ್ ಐಫ್ರೇಮ್‌ನ ಸುತ್ತಲೂ ಬಾರ್ಡರ್‌ನ್ನು ಪ್ರದರ್ಶಿಸಬೇಕೊ ಅಥವಾ ಬೇಡವೊ ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಬಾರ್ಡರ್‌ನ್ನು ತೆಗೆದು ಹಾಕಲು ಅಟ್ರಿಬ್ಯೂಟ್ ಮೌಲ್ಯವನ್ನು 0 ಗೆ ಸೆಟ್ ಮಾಡಿ.

ಉದಾಹರಣೆ

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

ಐಫ್ರೇಮ್ ಅನ್ನು ಲಿಂಕ್‌ಗೆ ಟಾರ್ಗೆಟ್ ಆಗಿ ಉಪಯೋಗಿಸಿ.

ಐಫ್ರೇಮ್ ಅನ್ನು ಒಂದು ಲಿಂಕ್‌ಗೆ ಟಾರ್ಗೆಟ್ ಫ್ರೇಮ್ ಆಗಿ ಉಪಯೋಗಿಸಬಹುದು.

ಲಿಂಕ್‌ನ ಟಾರ್ಗೆಟ್ ಅಟ್ರಿಬ್ಯೂಟ್, ಐಫ್ರೇಮ್ ನ name ಅಟ್ರಿಬ್ಯೂಟ್‌ನ್ನು ಉಲ್ಲೇಖಿಸುತ್ತದೆ.

```
<iframe src = "demo_iframe.htm" name= "iframe_a" > </iframe>
<p><ahref= "http://www.yahoo.com" target= "iframe_a" >yahoo.com</a></p>
```

HTML ಐಫ್ರೇಮ್ ಟ್ಯಾಗ್

ಟ್ಯಾಗ್(Tag)	ವಿವರಣೆ (Description)
<iframe>	ಇನ್‌ಲೈನ್ ಸಬ್ ವಿಂಡೊವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

HTML ಕಲರ್ (HTML Colour)

ಕಲರ್ ಮೌಲ್ಯಗಳು

HTML ಕಲರ್‌ಗಳನ್ನು, ಹೆಕ್ಸಾಡೆಸಿಮಲ್ ಸಂಕೇತವನ್ನು (HEX) ಬಳಸಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಇದು ಕೆಂಪು, ಹಸಿರು ಮತ್ತು ನೀಲಿ ಬಣ್ಣಗಳ ಸಂಯೋಜನೆಯಾಗಿದೆ.(RGB)

ಒಂದು ಬೆಳಕಿನ ಅತ್ಯಂತ ಕಡಿಮೆ ಮೌಲ್ಯ 0 .(HEX ನಲ್ಲಿ :00) ಆಗಿರುತ್ತದೆ. ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯ 255 ಆಗಿರುತ್ತದೆ.(HEX ನಲ್ಲಿ :FF).

HEX ಮೌಲ್ಯಗಳನ್ನು ಎರಡು ಅಂಕಿಗಳ 3 ಜೋಡಿಗಳಿಂದ ಸೂಚಿಸಲಾಗುತ್ತದೆ, # ಚಿನ್ಹೆಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ಕಲರ್ ಮೌಲ್ಯಗಳು (ಚಿತ್ರ 15) (Color Values)

```
<!DOCTYPE html>
<html>
<body>
<p style="background-color:#FFFF00">
    Color set by using Hex value
  </p>
  <p style= "background-color:rgb(255,255,0)">
  Color set by using Rgb value
  </p>
  <p style="background-color:yellow">
  Color set by using color name
  </p>
  </body>
  </html>
```

© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ

Color set by using Hex value

Color set by using Rgb value

Color set by using color name

DHTML

HTML ಸ್ಕ್ರಿಪ್ಟ್ ಎಲಿಮೆಂಟ್

<script> ಟ್ಯಾಗ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟ್ ಅನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಉದಾ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್.

ಸ್ಕ್ರಿಪ್ಟ್ ಎಲಿಮೆಂಟ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನಾದರೂ ಹೊಂದಿರುತ್ತದೆ ಅಥವಾ src ಅಟ್ರಿಬ್ಯೂಟ್ ಮೂಲಕ ಬಾಹ್ಯ ಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲ್‌ನ್ನಾದರೂ ಸೂಚಿಸುತ್ತದೆ.

ಅವಶ್ಯಕವಿರುವ ಅಟ್ರಿಬ್ಯೂಟ್ ಪ್ರಕಾರವು ಸ್ಕ್ರಿಪ್ಟ್‌ನ MIME ಪ್ರಕಾರವನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಸಾಮಾನ್ಯ ಉಪಯೋಗಗಳು - ಚಿತ್ರದ ಮ್ಯಾನಿಪ್ಯುಲೇಷನ್, ಫಾರ್ಮ್ ವ್ಯಾಲಿಡೇಷನ್, ಮತ್ತು ವಿಷಯದ ಡೈನಾಮಿಕ್ ಬದಲಾವಣೆಗಳು ಆಗಿವೆ.

ಕೆಳಗಿನ ಸ್ಕ್ರಿಪ್ಟ್ HTML ಔಟ್ ಪುಟ್‌ನಲ್ಲಿ Hello World! ಎಂದು ತೋರಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ

<script type=“text/javascript“>

Document.write(“Hello World!“)

</script>

Fig 14

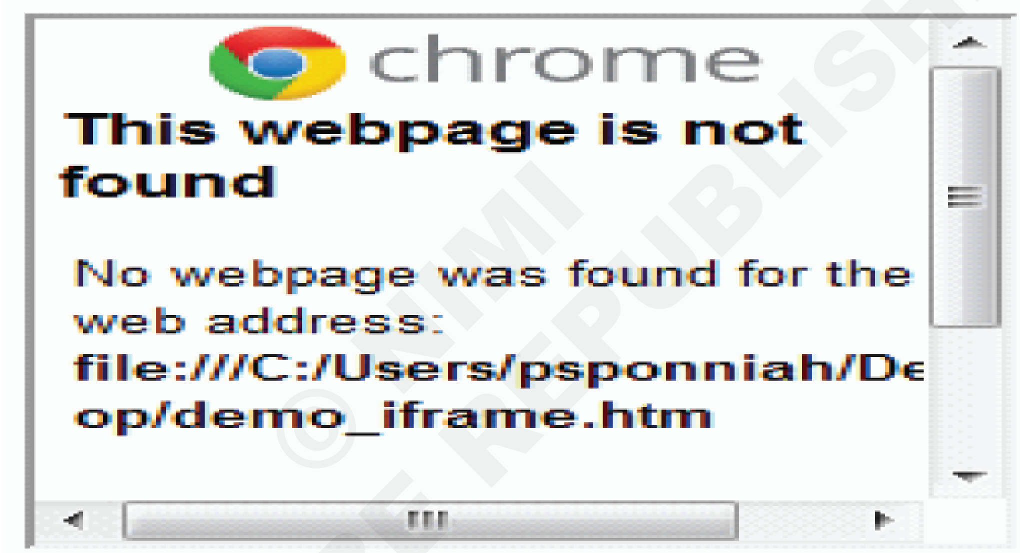


Fig 15

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

© NIMI NOT TO BE REPUBLISHED

HTML ನೋಸ್ಕ್ರಿಪ್ಟ್ ಎಲಿಮೆಂಟ್ (The HTML noscript Element)

<noscript> ಟ್ಯಾಗ್‌ನ್ನು , ತಮ್ಮ ಬ್ರೌಸರ್‌ನಲ್ಲಿ ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳನ್ನು ನಿಷ್ಕ್ರೀಯಗೊಳಿಸಿದ ಬಳಕೆದಾರರಿಗೆ ಪರ್ಯಾಯ ವಿಷಯವನ್ನು ಒದಗಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅಥವಾ ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್‌ನ್ನು ಬೆಂಬಲಿಸದ ಬ್ರೌಸರ್ ಹೊಂದಿದ ಬಳಕೆದಾರರಿಗೆ ವಿಷಯವನ್ನು ಒದಗಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ನೋಸ್ಕ್ರಿಪ್ಟ್ ಎಲಿಮೆಂಟ್, ಸಾಮಾನ್ಯ HTML ಪುಟದ body ಎಲಿಮೆಂಟ್ ನಲ್ಲಿರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಸ್ಕ್ರಿಪ್ಟ ಗಳನ್ನು ಬೆಂಬಲಿಸದಿದ್ದರೆ ಅಥವಾ ಬಳಕೆದಾರರ ಬ್ರೌಸರ್‌ನಲ್ಲಿ ನಿಷ್ಕ್ರೀಯಗೊಳಿಸಿದ್ದರೆ ಮಾತ್ರ ನೋಸ್ಕ್ರಿಪ್ಟ್ ಎಲಿಮೆಂಟ್‌ನಲ್ಲಿರುವ ವಿಷಯವನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
Document.write("Hello World!")
</script>
<noscript> Sorry, your browser does not support Javascript</noscript>
<p> A browser without support for Javascript will show the text in the noscript element.</p>
</body>
</html>
```

ಫಲಿತಾಂಶ

Hello World!

A browser without support for Javascript will show the text in the noscript element

HTML ಸ್ಕ್ರಿಪ್ಟ್ ಟ್ಯಾಗ್‌ಗಳು (HTML Script Tags)

ಟ್ಯಾಗ್ (Tag)	ವಿವರಣೆ (Description)
<script>	ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<noscript>	ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಬೆಂಬಲಿಸದ ಬಳಕೆದಾರರಿಗೆ ಪರ್ಯಾಯ ವಿಷಯವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ

HTML ಎಂಟಿಟಿಗಳು (HTML Entities)

HTML ನಲ್ಲಿ ಕೆಲವು ಅಕ್ಷರಗಳನ್ನು ಕಾಯ್ದಿರಿಸಲಾಗಿದೆ.

ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ, ಗಿಂತ ಕಡಿಮೆ (ಲೆಸ್ ದ್ಯಾನ್ <) ಅಥವಾ, ಗಿಂತ ಹೆಚ್ಚು (ಗ್ರೇಟರ್ ದ್ಯಾನ್ >) ಚಿನ್ಹೆಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು ಸಾಧ್ಯವಿಲ್ಲ. ಏಕೆಂದರೆ ಬ್ರೌಸರ್ ಇವುಗಳನ್ನು ಟ್ಯಾಗ್‌ಗಳೊಂದಿಗೆ ಬೆರೆಸುತ್ತದೆ.

ಕಾಯ್ದಿರಿಸಿದ ಅಕ್ಷರಗಳನ್ನು ನಿಜವಾಗಿ ಪ್ರದರ್ಶಿಸಲು, HTML ಸೋರ್ಸ್ ಕೋಡ್‌ನಲ್ಲಿ ನಾವು ಕ್ಯಾರೆಕ್ಟರ್ ಎಂಟಿಟಿಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ಕ್ಯಾರೆಕ್ಟರ್ ಎಂಟಿಟಿಗಳು ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತವೆ :

&entity_name;

ಅಥವಾ

&#entity_number;

ನಾನ್ ಬ್ರೇಕಿಂಗ್ ಸ್ಪೇಸ್

ನಾನ್ ಬ್ರೇಕಿಂಗ್ ಸ್ಪೇಸ್, ಇದು HTML ನಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಕ್ಯಾರೆಕ್ಟರ್ ಎಂಟಿಟಿ ಆಗಿದೆ. ( )

ಬ್ರೌಸರ್ ಗಳು HTML ಪುಟಗಳಲ್ಲಿ ಯಾವಾಗಲೂ ಸ್ಥಳಗಳನ್ನು ಮೊಟಕುಗೊಳಿಸುತ್ತವೆ. ನೀವು ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ 10 ಸ್ಥಳಗಳನ್ನು ಬರೆದರೆ, ಪುಟವನ್ನು ಪ್ರದರ್ಶಿಸುವ ಮೊದಲು ಬ್ರೌಸರ್ ಇವುಗಳಲ್ಲಿ 9 ಸ್ಥಳಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ನಿಮ್ಮ ಟೆಕ್ಸ್ಟ್‌ನಲ್ಲಿ ಸ್ಥಳ ಸೇರಿಸಲು, ನೀವು ಕ್ಯಾರೆಕ್ಟರ್ ಎಂಟಿಟಿ ಆದ   ಉಪಯೋಗಿಸಬಹುದು.

HTML ನ ಉಪಯುಕ್ತ ಕ್ಯಾರೆಕ್ಟರ್ ಎಂಟಿಟಿಗಳು (HTML Useful Character Entities)

ಸೂಚನೆ : ಎಂಟಿಟಿ ಹೆಸರುಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿವೆ!

HTML ಯುನಿಫಾರ್ಮ್ ರೀಸೋರ್ಸ್ ಲೊಕೇಟರ್‌ಗಳು

URL ಇದು ವೆಬ್ ಅಡ್ರೆಸ್‌ನ ಇನ್ನೊಂದು ಶಬ್ದವಾಗಿದೆ.

ಒಂದು URL ಶಬ್ದಗಳಿಂದ ಕೂಡಿರಬಹುದು ಉದಾ :"w3schools.com", ಅಥವಾ ಇಂಟರ್‌ನೆಟ್ ಪ್ರೊಟೋಕಾಲ್ (IP) ಅಡ್ರೆಸ್: 192.68.20.50. ಬಹಳ ಜನರು ಸರ್ಫ ಮಾಡುವಾಗ ವೆಬ್‌ಸೈಟ್‌ನ ಹೆಸರನ್ನು ನಮೂದಿಸುತ್ತಾರೆ, ಏಕೆಂದರೆ ಸಂಖ್ಯೆಗಳಿಗಿಂತ ಹೆಸರುಗಳನ್ನು ನೆನಪಿಟ್ಟುಕೊಳ್ಳುವುದು ಸುಲಭವಾಗಿರುತ್ತದೆ.

ಯುಆರ್‌ಎಲ್ -ಯುನಿಫಾರ್ಮ್ ರಿಸೋರ್ಸ್ ಲೊಕೇಟರ್ (URL - Uniform Resource Locator)

HTML ಪುಟದಲ್ಲಿ ನೀವು ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, <a> ಟ್ಯಾಗ್ ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನಲ್ಲಿರುವ ಒಂದು ಅಡ್ರೆಸ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಯುನಿಫಾರ್ಮ್ ರಿಸೋರ್ಸ್ ಲೊಕೇಟರ್(URL)ನ್ನು , ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್‌ನಲ್ಲಿ ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಸೂಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ (Result)	ವಿವರಣೆ (Description)	ಎಂಟಿಟಿ ಹೆಸರು (Entity Name)	ಎಂಟಿಟಿ ಸಂಖ್ಯೆ (Entity Number)
	Non-breaking space		
<	Less than	<	<
>	Greater than	>	>
&	Ampersand	&	&
¢	Cent	¢	¢
£	Pound	£	£
¥	Yen	¥	¥
•	Euro	€	€
§	Section	§	§
©	Copyright	©	©
®	Registered trade-mark	®	®
TM	trademark	™	™

ವಿವರಣೆ: (Explanation:)

- **ಸ್ಕೀಮ್** - ಇಂಟರ್‌ನೆಟ್ ಸರ್ವೀಸ್‌ನ ವಿಧವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. http ಸಾಮಾನ್ಯ ವಿಧವಾಗಿದೆ.
- **ಹೋಸ್ಟ್** - ಡೊಮೇನ್ ಹೋಸ್ಟ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ(http ಯ ಡೀಫಾಲ್ಟ್ ಹೋಸ್ಟ್ www ಆಗಿದೆ)
- **ಡೊಮೇನ್** - ಇಂಟರ್‌ನೆಟ್ ಡೋಮೇನ್ ನೇಮ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ, ಉದಾ w3schools.com
- **ಪೋರ್ಟ್** - ಹೋಸ್ಟ್‌ನಲ್ಲಿ ಪೋರ್ಟ್ ಸಂಖ್ಯೆಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ(http ಯ ಡೀಫಾಲ್ಟ್ ಪೋರ್ಟ್ ಸಂಖ್ಯೆ 80 ಇರುತ್ತದೆ)
- **ಪಾಥ್** - ಸರ್ವರ್‌ನಲ್ಲಿ ಪಾಥ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ (ಇದನ್ನು ಬಿಟ್ಟರೆ, ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ವೆಬ್ ಸೈಟ್‌ನ ರೂಟ್ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ)
- ಫೈಲ್ ನೇಮ್ - ಡಾಕ್ಯುಮೆಂಟ್ / ರೀಸೋರ್ಸ್‌ನ ಹೆಸರನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ

ಸಾಮಾನ್ಯ URL ಸ್ಕೀಮ್‌ಗಳು (Common URL Schemes)

ಈ ಕೆಳಗಿನ ಟೇಬಲ್ ಕೆಲವು ಸಾಮಾನ್ಯ ಸ್ಕೀಮ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿದೆ.

ಸ್ಕೀಮ್	ಪೂರ್ಣ ಪದ	ಈ ಸ್ಕೀಮ್ ಉಪಯೋಗಿಸುವ ಪುಟಗಳು
http	HyperText Transfer Protocol	ಸಾಮಾನ್ಯ ವೆಬ್‌ಪುಟಗಳು http:// ಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತವೆ. ಎನ್‌ಕ್ರಿಪ್ಟ್ ಆಗಿಲ್ಲ.
https	Secure HyperText Transfer Protocol	ಸುರಕ್ಷಿತ ವೆಬ್‌ಪುಟಗಳು. ವಿನಿಮಯವಾದ ಎಲ್ಲ ಮಾಹಿತಿಯು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಆಗಿರುತ್ತದೆ.
ftb	File Transfer Protocol	ವೆಬ್‌ಸೈಟ್‌ಗೆ ಫೈಲ್‌ಗಳನ್ನು ಅಪ್‌ಲೋಡ್ ಅಥವಾ ಡೌನ್‌ಲೋಡ್ ಮಾಡಲು. ಡೋಮೇನ್ ನಿರ್ವಹಿಸಲು ಉಪಯುಕ್ತವಾಗಿದೆ.
file		ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಫೈಲ್

ಯುಆರ್‌ಎಲ್ ಎನ್‌ಕೋಡಿಂಗ್ (URL Encoding)

URL ಗಳನ್ನು , ASCII ಅಕ್ಷರ-ಸೆಟ್ ಉಪಯೋಗಿಸಿ ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಕಳುಹಿಸಲಾಗುತ್ತದೆ.

URL ಗಳು, ಸಾಮಾನ್ಯವಾಗಿ ASCII ಸೆಟ್ ಹೊರಗಿನ ಅಕ್ಷರಗಳನ್ನು ಹೊಂದಿದ್ದರೂ, URL ಒಂದು ವ್ಯಾಲೀಡ್ ASCII ಫಾರ್ಮ್ಯಾಟ್‌ಗೆ ಪರಿವರ್ತಿಸಬೇಕಾಗುತ್ತದೆ.

URL ಎನ್‌ಕೋಡಿಂಗ್, ASCII ಅಲ್ಲದ ಅಕ್ಷರಗಳನ್ನು % ನಂತರ ಎರಡು ಹೆಕ್ಸಾಡೆಸಿಮಲ್ ಅಂಕಿಗಳಿಂದ ಬದಲಾಯಿಸುತ್ತದೆ.

URL ಖಾಲಿ ಸ್ಥಳಗಳನ್ನು ಹೊಂದಿರಬಾರದು. URL ಎನ್‌ಕೋಡಿಂಗ್ ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಖಾಲಿ ಸ್ಥಳವನ್ನು + ಚಿನ್ಹೆಯಿಂದ ಬದಲಾಯಿಸುತ್ತದೆ.

XML ರಚನೆ (XML Structure)

XML ರಚನೆ ಡಾಕ್ಯುಮೆಂಟ್ ಭಾಗಗಳು, ಮುನ್ನುಡಿ ಒಳಗೊಂಡಿರುತ್ತದೆ, ಮತ್ತು ಸರಳ XML ಡಾಕ್ಯುಮೆಂಟ್ ಉದಾಹರಣೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಡಾಕ್ಯುಮೆಂಟ್ ಭಾಗಗಳು

- ಪ್ರೊಲೊಗ್
- ಡಾಕ್ಯುಮೆಂಟ್ ಎಲಿಮೆಂಟ್ (ರೂಟ್ ಎಲಿಮೆಂಟ್)

ಪ್ರೊಲೊಗ್ (The Prologue)

ಪ್ರೊಲೊಗ್, HTML ನಲ್ಲಿಯ ಹೆಡರ್‌ಗೆ ಸಮನಾಗಿದೆ. ಇದು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

- XML ಡಿಕ್ಲೇರೇಷನ್ (ಐಚ್ಛಿಕವಾಗಿದೆ) ಉದಾ: <?xml version="1.0"?>

© NIMI NOT TO BE REPUBLISHED

- DTD ಅಥವಾ ಉಲ್ಲೇಖ (ಐಚ್ಛಿಕವಾಗಿದೆ) ಬಾಹ್ಯ DTD ಫೈಲ್‌ಗೆ ಉಲ್ಲೇಖದ ಉದಾಹರಣೆ:

  <!DOCTYPE LANGLIST SYSTEM "langlist.dtd">

- ಪ್ರೊಸೆಸಿಂಗ್ ಸೂಚನೆಗಳು - ಸ್ಟೈಲ್ ಶೀಟ್‌ನಿಂದ ನಿರ್ಧರಿಸಲ್ಪಡುವ ಸ್ಟೈಲ್‌ಗೆ ಕಾರಣವಾದ ಪ್ರೊಸೆಸಿಂಗ್ ಸೂಚನೆಗೆ ಉದಾಹರಣೆ :

  <?xml-stylesheettype="text/css" href="xmlstyle.css"?>

XML ನ ಉದಾಹರಣೆ

ಸಂಪೂರ್ಣ ತಯಾರಾದ XML ಡಾಕ್ಯುಮೆಂಟ್ ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ :

```
<?xml version="1.0"?>
<LAND>
   <FOREST>
      <TREE>Oak</TREE>
      <TREE>Pine</TREE>
      <TREE>Maple</TREE>
</FOREST>
<MEADOW>
     <GRASS> Bluegrass</GRASS>
     <GRASS> Fescue</GRASS>
     <GRASS> Rye</GRASS>
</MEADOW>
</LAND>
```

ಮೇಲಿನ LAND ಎಲೆಮೆಂಟ್, ರೂಟ್ ಎಲೆಮೆಂಟ್ ಆಗಿದೆ.

ಫಲಿತಾಂಶ (Result)

Oak pine Maple Bluegrass Fescue Rye

ವೆಬ್ ಎಲೆಮೆಂಟ್ (Web Elements)

ವೆಬ್‌ಪುಟವು ಹಲವಾರು ವಿಧದ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಈ ಮಾಹಿತಿಯನ್ನು ಅಂತಿಮ ಬಳಕೆದಾರರು ನೋಡಬಹುದು, ಕೇಳಬಹುದು ಅಥವಾ ಸಂವಹನ ನಡೆಸಬಹುದು.

ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ (Web Hosting)

ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ ಸೇವೆಯು, ಇಂಟರ್‌ನೆಟ್ ಹೋಸ್ಟಿಂಗ್ ಸೇವೆಯ ಒಂದು ವಿಧವಾಗಿದೆ. ಇದು ವ್ಯಕ್ತಿಗಳಿಗೆ ಮತ್ತು ಸಂಸ್ಥೆಗಳಿಗೆ ವರ್ಲ್ಡ್ ವೈಡ್ ವೆಬ್ ಮೂಲಕ ತಮ್ಮ ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ವೆಬ್ ಹೋಸ್ಟ್‌ಗಳು ಸರ್ವರ್‌ನಲ್ಲಿ ಸ್ಥಳ ನೀಡುವ ಕಂಪನಿಗಳಾಗಿವೆ, (ಕಂಪ್ಯೂಟಿಂಗ್) ಇದನ್ನು ಕ್ಲೈಂಟ್‌ಗಳಿಂದ ಸ್ವಂತಕ್ಕಾಗಿ ಅಥವಾ ಬಾಡಿಗೆಗೆ ಉಪಯೋಗಿಸಲು ಮತ್ತು ಸಾಮಾನ್ಯವಾಗಿ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ಒದಗಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ವೆಬ್ ಹೋಸ್ಟ್‌ಗಳು ತಮ್ಮ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳಲ್ಲಿ ಇರುವ ಇತರ ಸರ್ವರ್‌ಗಳಿಗೆ ಡಾಟಾ ಸೆಂಟರ್ ಸ್ಪೇಸ್ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ಒದಗಿಸುತ್ತವೆ. ಇದನ್ನು ಕೊಲೊಕೇಷನ್(collocation) ಎನ್ನುತ್ತಾರೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಿಎಂಎಸ್ ಮತ್ತು ವೆಬ್ ಆಥರಿಂಗ್ ಸಾಧನಗಳ ಪರಿಚಯ (Web designing)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಕಂಟೆಂಟ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಸಿಸ್ಟಮ್ ಅರ್ಥವನ್ನು ವಿವರಿಸಿ
- ಕೆಲವು ಜನಪ್ರಿಯ ಸಿಎಂಎಸ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ
- ಕೊಂಪೊಜರ್ ವೆಬ್ ಆಥರಿಂಗ್ ಟೂಲ್‌ನ ಉಪಯೋಗಗಳನ್ನು ಮತ್ತು ಮುಖ್ಯ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ವಿವರಿಸಿ.

ಸಿಎಂಎಸ್‌ನ ವ್ಯಾಖ್ಯೆ (Definition of CMS):

ಸಿಎಂಎಸ್ ಎಂದರೆ ಕಂಟೆಂಟ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಸಿಸ್ಟಮ್ (ವಿಷಯ ನಿರ್ವಹಣಾ ವ್ಯವಸ್ಥೆ). ಸಿಎಂಎಸ್ ಒಂದು ಸಾಫ್ಟವೇರ್ ಸಾಧನವಾಗಿದ್ದು, ನಿಮಗೆ ಮಾಹಿತಿಯನ್ನು ರಚಿಸಲು, ಎಡಿಟ್ ಮಾಡಲು ಮತ್ತು ಪಬ್ಲಿಶ್(ಪ್ರಕಟಿಸಲು) ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಸಿಎಂಎಸ್ ನ ವಿವರಣೆ (Description of CMS) :

ಬಳಕೆದಾರರಿಗೆ ವೆಬ್‌ಪುಟದ ವಿಷಯವನ್ನು ನಿರ್ಮಿಸಲು ಮತ್ತು ಮಾರ್ಪಡಿಸಲು ಒಂದು ಅರ್ಥಗರ್ಭಿತ ಮಾಧ್ಯಮವನ್ನು ಒದಗಿಸುವುದು ಸಿಎಂಎಸ್‌ನ ಗುರಿಯಾಗಿದೆ. ಪ್ರತಿಯೊಂದು ಸಿಎಂಎಸ್ ಒಬ್ಬರು ಅಥವಾ ಹೆಚ್ಚು ಬಳಕೆದಾರರಿಗೆ ವೆಬ್‌ನಲ್ಲಿ ಲೈವ್ ಆಗಿ ನವೀಕರಣಗಳನ್ನು ಪ್ರಕಟಿಸಲು ಅನುಮತಿಸುವ ವೆಬ್ ಪಬ್ಲಿಶಿಂಗ್ ಸಾಧನವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಎಡಿಟಿಂಗ್ ಘಟಕವನ್ನು ಕಂಟೆಂಟ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ಅಪ್ಲಿಕೇಷನ್ (CMA) ಎಂದು ಕರೆಯುತ್ತಾರೆ, ಮತ್ತು ಪಬ್ಲಿಶಿಂಗ್ ಸಾಧನವನ್ನು ಕಂಟೆಂಟ್ ಡೆಲಿವರಿ ಅಪ್ಲಿಕೇಷನ್ (CDA) ಎಂದು ಕರೆಯುತ್ತಾರೆ. ವೆಬ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಸುಗಮಗೊಳಿಸಲು ಈ ಎರಡು ಘಟಕಗಳನ್ನು ಸಿಎಂಎಸ್‌ನಲ್ಲಿ ಒಟ್ಟಿಗೆ ಸಂಯೋಜಿಸಲಾಗಿದೆ.

ವೆಬ್ ಪಬ್ಲಿಶಿಂಗ್‌ನಲ್ಲಿ , ಮಾಹಿತಿಯು - ಸರಳವಾದ ಟೆಕ್ಸ್ಟ್, ಫೋಟೋ, ಸಂಗೀತ, ವಿಡಿಯೊ, ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳು, ಅಥವಾ ನೀವು ಯೋಚಿಸಿದ ಯಾವುದೇ ವಿಷಯದ ಬಗ್ಗೆ ಆಗಿರಬಹುದು. ಆರಂಭಿಕ ಸಿಎಂಎಸ್ ಸಾಫ್ಟವೇರ್, ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಮತ್ತು ಸ್ಥಳೀಯ ಕಂಪ್ಯೂಟರ್ ಫೈಲ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತಿತ್ತು. ಈಗ ಹೆಚ್ಚಿನ ಸಿಎಂಎಸ್ ಸಾಫ್ಟವೇರ್‌ಗಳನ್ನು ವೆಬ್‌ನ ವಿಷಯವನ್ನು ನಿರ್ವಹಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಸಿಎಂಎಸ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ ಇದಕ್ಕೆ ನಿರ್ವಹಣೆ ಮಾಡಲು ಯಾವುದೇ ತಾಂತ್ರಿಕ ಕೌಶಲ್ಯದ ಅಥವಾ ಜ್ಞಾನದ ಅಗತ್ಯವಿಲ್ಲ.

ಕಂಟೆಂಟ್ ಮ್ಯಾನೇಜಮೆಂಟ್ ವ್ಯವಸ್ಥೆಗಳು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದಾದ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ರೀತಿಯಲ್ಲಿ ಮತ್ತು ವೆಬ್ ಆಧಾರಿತ ಬಳಕೆದಾರರ ಸಂಪರ್ಕ ಸಾಧನವಾಗಿ ಲಭ್ಯವಿರುತ್ತವೆ. ವೆಬ್ ಇಂಟರ್‌ಫೇಸನ್ನು ಬಳಸುವುದರಿಂದ ವೆಬ್‌ಸೈಟನ್ನು ನವೀಕರಿಸುವ ಪ್ರಕ್ರಿಯೆಯು ಸರಳವಾಗುತ್ತದೆ. ಹೆಚ್ಚುವರಿಯಾಗಿ, ಅನೇಕ ವೆಬ್ ಆಧಾರಿತ ಸಿಎಂಎಸ್‌ಗಳು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ನವೀಕರಣಗೊಳ್ಳುತ್ತವೆ. ಇದು ಎಲ್ಲ ಬಳಕೆದಾರರು ತಮ್ಮ ಮಾಹಿತಿಯನ್ನು ನಿರ್ವಹಿಸಲು ಇತ್ತೀಚಿನ ಸಾಧನಗಳನ್ನು ಹೊಂದಿರುವರೆಂದು ಖಚಿತಪಡಿಸುತ್ತದೆ.

ಸಿಎಂಎಸ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ, ಇದಕ್ಕೆ ನಿರ್ವಹಣೆ ಮಾಡಲು ಯಾವುದೇ ತಾಂತ್ರಿಕ ಕೌಶಲ್ಯದ ಅಥವಾ ಜ್ಞಾನದ ಅಗತ್ಯವಿಲ್ಲ. ಅದ್ದರಿಂದ ಸಿಎಂಎಸ್ ನಿಮ್ಮ ಎಲ್ಲ ಮಾಹಿತಿಯನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ನೀವು ನಿರ್ವಹಿಸುವ ಅವಶ್ಯಕತೆ ಇಲ್ಲ.

ಇಂದು ಹಲವಾರು ವೆಬ್ ಆಧಾರಿತ ಸಿಎಂಎಸ್ ಸಾಧನಗಳು ಲಭ್ಯ ಇರುತ್ತವೆ. ಕೆಲವು ಜನಪ್ರೀಯ ಸಿಎಂಎಸ್ ಸಾಧನಗಳನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ:

- ವರ್ಡ್‌ಪ್ರೆಸ್ - ಇದು ಉಚಿತ ವೆಬ್ ಸಾಫ್ಟವೇರ್ ಆಗಿದ್ದು, ಟೆಂಪ್ಲೇಟ್ - ಆಧಾರಿತ ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ಅಥವಾ ಬ್ಲಾಗ್‌ಗಳನ್ನು ರಚಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.
- ಬ್ಲಾಗರ್ - ಇದು ಗೂಗಲ್‌ನ ಬ್ಲಾಗಿಂಗ್ ಟೂಲ್ ಆಗಿದ್ದು, ವಿಶೇಷವಾಗಿ ಬ್ಲಾಗನ್ನು ನಿರ್ವಹಿಸಲು ಇದನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.
- ಜೂಮ್ಲಾ - ಇದು ಹೊಂದಿಕೊಳ್ಳುವ ವೆಬ್ ಪಬ್ಲಿಶಿಂಗ್ ಟೂಲ್ ಆಗಿದ್ದು, ಇದು ಕಸ್ಟಮ್ ಡಾಟಾಬೇಸಗಳನ್ನು ಮತ್ತು ಎಕ್ಸ್‌ಟೆನ್ಷನ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ.
- ದ್ರುಪಲ್ - ಇದು ಓಪನ್ ಸೋರ್ಸ್ ಪ್ಲಾಟ್ ಫಾರ್ಮ್ ಆಗಿದ್ದು, ಹೆಚ್ಚಾಗಿ ಸಮುದಾಯ ಆಧಾರಿತ ಸೈಟ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- ವೀಬ್ಲೀ- ಇದು ವೆಬ್ ಆಧಾರಿತ ಪ್ಲಾಟ್ ಫಾರ್ಮ್ ಆಗಿದ್ದು, ಸರಳ ವೈಯಕ್ತಿಕ ಮತ್ತು ವ್ಯಾಪಾರದ ವೆಬ್ ಸೈಟ್‌ಗಳನ್ನು ನಿರ್ಮಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

NIMI © NOT TO BE REPUBLISHED

- ವಿಕ್ಸ್ - ಇದು ವೆಬ್ ಪಬ್ಲಿಶಿಂಗ್ ಸಾಧನಗಳ ಸಂಗ್ರಹವಾಗಿದ್ದು, ಹೆಚ್ಚು ಗ್ರಾಹಕೀಯಗೊಳಿಸಬಹುದಾದ ವೆಬ್ ಸೈಟ್‌ಗಳ ರಚನೆಗೆ ಇದನ್ನು ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.

WSY|WYG ವೆಬ್ ಆಥರಿಂಗ್ ಟೂಲ್‌ಗಳ ಪರಿಚಯ (Introduction to WSY|WYG web authoring tools)

WSY|WYG ಎಂದರೆ ``What You See Is What You Get``. ಇಂತಹ ಎಡಿಟರ್‌ಗಳಲ್ಲಿ ನೀವು ನೇರವಾಗಿ ನಿಮ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಸೋರ್ಸ್ ಕೋಡನ್ನು ಎಡಿಟ್ ಮಾಡುವುದಿಲ್ಲ, ಆದರೆ ಅಂತಿಮ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಕಾಣುವ ಅದರ ಪ್ರಸ್ತುತಿಯನ್ನು ಎಡಿಟ್ ಮಾಡುತ್ತೀರಿ. ಅದ್ದರಿಂದ ಕೈಯಿಂದ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಬರೆಯುವ ಬದಲು (ನೀವು ವರ್ಡ್ ಅಥವಾ ನೋಟ್ ಪ್ಯಾಡ್‌ನಲ್ಲಿ ಬರೆಯುವಂತೆ) ನೀವು ಎಡಿಟರ್ ವಿಂಡೊವನ್ನು ಉಪಯೋಗಿಸಿ ವಿನ್ಯಾಸ ಘಟಕಗಳೊಂದಿಗೆ ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು. ಇದರ ಅರ್ಥ ಒಂದು ಡಾಕ್ಯುಮೆಂಟ್ ಅಥವಾ ಚಿತ್ರವನ್ನು ರಚಿಸುವಾಗ ಅಂತಿಮ ಫಲಿತಾಂಶವನ್ನು ಹೋಲುವ ವಿಷಯಗಳನ್ನು ನೀವು ನೋಡಬಹುದು.

ಬಳಸಲು ಸುಲಭವಾದ ಅನೇಕ WSY|WYG ಪ್ರೊಗ್ರಾಂಗಳಿವೆ. ಹೊಸದಾಗಿ ಆರಂಭಿಸುವವರಿಗೂ ಕೂಡ ಕಠಿಣ ಮತ್ತು ಸಂಪೂರ್ಣ ಕ್ರಿಯಾತ್ಮಕ ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ರಚಿಸಲು ಅಗತ್ಯವಿರುವ ಎಲ್ಲ ಸಾಧನಗಳು ಇದರಲ್ಲಿ ಲಭ್ಯವಿವೆ. ಈ ಸೈಟ್‌ಗಳು HTML ಕೋಡ್‌ನೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಆಯ್ಕೆಯನ್ನು ಕೂಡ ಹೊಂದಿವೆ. ವಿನ್ಯಾಸಗೊಳಿಸುವುದು ಇರಬಹುದು ಅಥವಾ ಎಡಿಟಿಂಗ್ ಇರಬಹುದು. HTML ಎಡಿಟರ್‌ನಲ್ಲಿ ವೆಬ್‌ಸೈಟ್ ತಯಾರಿಸುವುದು ಸುಲಭವಾಗಿದೆ. ಏಕೆಂದರೆ ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿಪಡಿಸುವವರು ಅದರಲ್ಲಿ ಸಾಧನಗಳನ್ನು ಸೇರಿಸುತ್ತಿರುತ್ತಾರೆ. ಇದರಿಂದ ನಿಮಗೆ ಸ್ಟೈಲ್‌ನೊಂದಿಗೆ ಸುಧಾರಿತ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಅನೇಕ WSY|WYG ವೆಬ್ ಆಥರಿಂಗ್ ಟೂಲ್‌ಗಳು, ಸೈಟ್‌ನಲ್ಲಿ ಸೊಗಸಾದ ಮತ್ತು ಸ್ಥಿರವಾದ ವಿನ್ಯಾಸದೊಂದಿಗೆ ಡೈನಾಮಿಕ್ HTML ನ್ನು ಅಥವಾ ಹಲವಾರು ಇತರ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸಂಯೋಜಿಸಲು ಆಧುನಿಕ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಒದಗಿಸುತ್ತವೆ.

ಕೋಂಪೋಜರ್‌ನ ಪರಿಚಯ (Introduction to Kompozer)

ಕೋಂಪೋಜರ್ ಒಂದು ಸಂಪೂರ್ಣ ವೆಬ್ ಆಥರಿಂಗ ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಇದು ವೆಬ್ ಫೈಲ್ ನಿರ್ವಹಣೆ ಮತ್ತು ಮೈಕ್ರೊಸಾಫ್ಟ ಫ್ರಂಟ್ ಪೇಜ್, ಅಡೋಬ್ ಡ್ರೀಮ್ ವೀವರ್ ಮತ್ತು ಇತರ ಉನ್ನತ ಮಟ್ಟದ ಪ್ರೊಗ್ರಾಂಗಳಲ್ಲಿ ಕಂಡುಬರುವ ಬಳಸಲು ಸುಲಭವಾದ WSY|WYG ವೆಬ್ ಪೇಜ್ ಎಡಿಟಿಂಗ್ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಸಂಯೋಜಿಸುತ್ತದೆ. ಕೋಂಪೋಜರ್ ಬಳಸಿ ನೀವು ವೆಬ್‌ಪುಟಗಳನ್ನು ರಚಿಸಬಹುದು ಮತ್ತು ವೆಬ್‌ಸೈಟನ್ನು ಯಾವುದೇ ತಾಂತ್ರಿಕ ಪರಿಣಿತಿ ಅಥವಾ HTML ನ ಜ್ಞಾನವಿಲ್ಲದೆ ನಿರ್ವಹಿಸಬಹುದು.

ಕೋಂಪೋಜರ್‌ನ ಉಪಯುಕ್ತ ವೈಶಿಷ್ಟ್ಯವೆಂದರೆ ನೀವು ನಿಮ್ಮ ವೆಬ್‌ಪುಟಗಳನ್ನು ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ ಸರ್ವರ್‌ನಲ್ಲಿ ಪಬ್ಲಿಶ್ ಮಾಡಲು ಉಪಯೋಗಿಸಬಹುದು. ನೀವು ನಿಮ್ಮ FTP ಖಾತೆಯ ಯೂಸರ್ ನೇಮ್ ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್ ಮತ್ತು ಸೈಟ್ URL ನ್ನು ತಿಳಿದಿರಬೇಕು ಮತ್ತು ಕೋಂಪೋಜರ್ ಲಾಗಿನ್ ಆಗುತ್ತದೆ ಮತ್ತು ಸರ್ವರ್‌ನಲ್ಲಿ ನಿಮ್ಮ ವೆಬ್‌ಪುಟಗಳನ್ನು ಹಾಕಿರಿ.

ಕೋಂಪೋಜರ್‌ನ ಫೈಲ್ ಮೆನು , ಎಲ್ಲ ಮುಖ್ಯ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅವಶ್ಯವಿರುವ ಮೆನು ಐಟಂಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಕೋಂಪೋಜರ್‌ನ ಟೂಲ್‌ಬಾರ್ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್ ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ :

- ಕಂಪೋಸಿಷನ್ ಟೂಲ್‌ಬಾರ್
- ಫಾರ್ಮ್ಯಾಟ್ ಟೂಲ್‌ಬಾರ್
- ಟ್ಯಾಬ್ ಬ್ರೌಸರ್ ಟೂಲ್‌ಬಾರ್
- ಎಡಿಟ್ ಮೋಡ್ ಟೂಲ್‌ಬಾರ್
- ಸ್ಟೇಟಸ್ ಟೂಲ್‌ಬಾರ್

ಕಂಪೋಸಿಷನ್ ಟೂಲ್‌ಬಾರ್ (Composition Toolbar)

ಕಂಪೋಸಿಷನ್ ಟೂಲ್‌ಬಾರ್, ಅತಿ ಹೆಚ್ಚು ಉಪಯೋಗಿಸಲ್ಪಡುವ ಮುಖ್ಯ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವ ಬಟನ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಡಿಫಾಲ್ಟ್ ಕಂಪೋಸಿಷನ್ ಟೂಲ್‌ಬಾರ್‌ನ ಸ್ನ್ಯಾಪ್ ಶಾಟ್ ಕೆಳಗೆ ಕೊಡಲಾಗಿದೆ. ಪ್ರತಿ ಬಟನ್ ಕೆಳಗಿನ ಟೆಕ್ಸ್ಟ ಆಯಾ ಬಟನ್‌ನ ಕಾರ್ಯವನ್ನು ಸೂಚಿಸುತ್ತದೆ (ಚಿತ, 1)

Fig 1

ಫಾರ್ಮ್ಯಾಟ್ ಟೂಲ್‌ಬಾರ್(ಚಿತ್ರ 2 ನ್ನು ನೊಡಿ) (Format Toolbar(Refer Fig.2))

Fig 2

ಕಂಪೋಜರ್‌ನಲ್ಲಿ ,ವೆಬ್ ಪುಟಗಳನ್ನು ಎಡಿಟ್ ಮಾಡಲು ಫಾರ್ಮ್ಯಾಟ್ ಟೂಲ್‌ಬಾರ್ ಅತ್ಯಂತ ಉಪಯುಕ್ತವಾದ ಸಾಧನವಾಗಿದೆ. ಫಾರ್ಮ್ಯಾಟ್ ಟೂಲ್‌ಬಾರ್‌ನಲ್ಲಿ, ನೀವು ಪ್ಯಾರಾಗ್ರಾಫ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ್ನು ಅನ್ವಯಿಸುವುದು, ಫಾಂಟ್ ಆಯ್ಕೆ ಮಾಡುವುದು, ಫೋರಗ್ರೌಂಡ್ ಅಥವಾ ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್

©NIMI NOT TO BE REPUBLISHED

ಕಲರ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು, ಗಾತ್ರವನ್ನು ಹೆಚ್ಚು ಅಥವಾ ಕಡಿಮೆ ಮಾಡುವುದು, ಟೆಕ್ಸ್ಟ್ ಸಮೂಹವನ್ನು ಎಂಫಸೈಝ್ ಮಾಡುವುದು ಮುಂತಾದವುಗಳನ್ನು ಮಾಡಬಹುದು. ನೀವು ಆರ್ಡರ್ಡ್ ಅಥವಾ ಅನ್‌ಆರ್ಡರ್ಡ್ ಲಿಸ್ಟಗಳನ್ನು ರಚಿಸುವುದು ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಸಮೂಹವನ್ನು ಎಡಕ್ಕೆ, ಬಲಕ್ಕೆ ಅಥವಾ ಮಧ್ಯಕ್ಕೆ ಜಸ್ಟಿಫೈ ಮಾಡಬಹುದು. ಯಾವ ಬಟನ್ ಯಾವ ಕೆಲಸವನ್ನು ಮಾಡುತ್ತದೆ ಎಂಬುದನ್ನು ತಿಳಿಯಲು ಬಟನ್ ಮೇಲೆ ನಿಮ್ಮ ಮೌಸ್ ಪಾಯಿಂಟರ್ ಇಡಿ ಮತ್ತು ಬಟನ್‌ನ ಕಾರ್ಯವನ್ನು ಸೂಚಿಸುವ ಟೂಲ್ ಟಿಪ್ ಕಾಣುತ್ತದೆ.

ಟ್ಯಾಬ್ ಬ್ರೌಸರ್ ಟೂಲ್‌ಬಾರ್(ಚಿತ್ರ 3 ನ್ನು ನೊಡಿ) (Tab Browser Toolbar(Refer Fig.3))

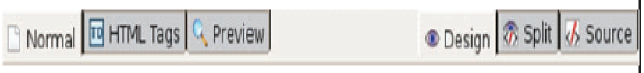
Fig 3

copa 1308203

ಏಕಕಾಲದಲ್ಲಿ ಅನೇಕ ವೆಬ್ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಎಡಿಟ್ ಮಾಡಲು ಕಂಪೋಜರ್ ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ತೆರೆದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗೂ ವಿಭಿನ್ನ ಟ್ಯಾಬ್ ಅನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ಡಾಕ್ಯುಮೆಂಟ್‌ಗೆ ಅನೇಕ ವಿಂಡೋಗಳನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ ಅದು ಅಸ್ತವ್ಯಸ್ತವಾಗಿರುವುದಿಲ್ಲ. ಮತ್ತು ಅನೇಕ ಟ್ಯಾಬ್‌ಗಳನ್ನು ಬಳಸುವುದರಿಂದ ನಿಮ್ಮ ಡೆಸ್ಕಟಾಪ್ ಸ್ವಚ್ಛವಾಗಿ ಕಾಣುತ್ತದೆ. ಪುಟಗಳ ಮೇಲೆ ದೃಶ್ಯ ಸೂಚಕವಾಗಿ "ಕೆಂಪು ಫ್ಲಾಪಿ ಐಕಾನ್" ಕಾಣುತ್ತದೆ. ಇದು ಪುಟಗಳು ಎಡಿಟ್ ಆಗಿವೆ ಆದರೆ ಸೇವ್ ಆಗಿಲ್ಲ ಎಂದು ಸೂಚಿಸುತ್ತದೆ.

ಎಡಿಟ್ ಮೋಡ್ ಟೂಲ್‌ಬಾರ್(ಚಿತ್ರ 4 ನ್ನು ನೋಡಿ) (Edit Mode Toolbar(Refer Fig.4))

Fig 4



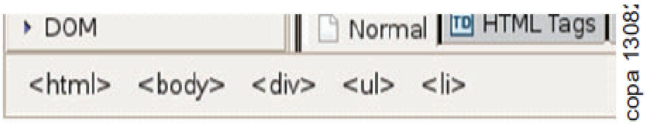
copa 1308204

ಎಡಿಟ್ ಮೋಡ್ ಟೂಲ್‌ಬಾರ್ ವಿವಿಂಗ್ ಮೋಡನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ಪ್ರಸ್ತುತ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಾಗಿ ಸಕ್ರೀಯವಾಗಿರುತ್ತದೆ. ಲಭ್ಯವಿರುವ ವಿವಿಂಗ್ ಮೋಡ್‌ಗಳು : ನಾರ್ಮಲ್ ವಿವ್, HTML ಟ್ಯಾಗ್ಸ್ ವಿವ್, HTML ಸೋರ್ಸ್ ವಿವ್, ಮತ್ತು ಪ್ರಿವಿವ್ ಮೋಡ್ . ನೀವು ನಿಮ್ಮ ವಿವಿಂಗ್ ಮೋಡನ್ನು ಉಳಿದ ಮೂರರಲ್ಲಿ ಯಾವುದರ ಮೇಲೂ ಮೌಸ್ ಕ್ಲಿಕ್ ಮಾಡಿ ಸುಲಭವಾಗಿ ಬದಲಾಯಿಸಬಹುದು.

ಸ್ಟೇಟಸ್ ಟೂಲ್‌ಬಾರ್(ಚಿತ್ರ 5 ನ್ನು ನೋಡಿ) (Status Toolbar(Refer Fig.5))

ಕಂಪೋಜರ್‌ನ ಸ್ಟೇಟಸ್ ಬಾರ್, HTML ಟ್ಯಾಗ್ ಕ್ರಮಾನುಗತಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ, ಕರ್ಸರ್‌ನ ಸ್ಥಾನವನ್ನು ತೋರಿಸುತ್ತದೆ. ಮೌಸ್‌ನ ಬಲ ಕ್ಲಿಕ್ ಮಾಡಿ ನಿಮಗೆ ಬೇಕಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳುವ ಮೂಲಕ ಸ್ಟೇಟಸ್ ಬಾರ್‌ನಲ್ಲಿರುವ ಯಾವುದೇ ನಿರ್ದಿಷ್ಟ ಟ್ಯಾಗ್‌ನ ಗುಣಲಕ್ಷಣಗಳನ್ನು ನೀವು ಸುಲಭವಾಗಿ ಎಡಿಟ್ / ನಿಯೋಜನೆ ಮಾಡಬಹುದು. ಸ್ಟೆಟಸ್ ಬಾರ್‌ನಲ್ಲಿರುವ ಟ್ಯಾಗ್ ಮೇಲೆ ಮೌಸ್‌ನ ಎಡ ಕ್ಲಿಕ್ ಮಾಡುವುದರಿಂದ ಟ್ಯಾಗ್‌ನ ಸುತ್ತಲಿರುವ ಟೆಕ್ಸ್ಟ್ ಆಯ್ಕೆ ಆಗುತ್ತದೆ.

Fig 5



© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು HTML ಪುಟಾಗಳೊಂದಿಗೆ ಸೇರಿಸುವುದು (JavaScript Embed JavaScript in HTML Pages)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳುವುದು (Understanding JavaScript)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮತ್ತು ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎಂದರೇನು ಮತ್ತು ಅದರ ಇತಿಹಾಸ ತಿಳಿದುಕೊಳ್ಳುವುದು
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಹೇಗೆ ರನ್ ಮಾಡುವುದು ಎಂದು ವಿವರಿಸಿ
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ರನ್ ಮಾಡಲು ಅವಶ್ಯಕವಿರುವ ಸಾಧನಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ
- ಒಂದು ಮಾದರಿ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಪ್ರೊಗ್ರಾಂ ನೋಡುವುದು
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ತಿಳಿಯುವುದು
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಅನುಕೂಲಗಳು ಮತ್ತು ಅನಾನುಕೂಲಗಳನ್ನು ವಿವರಿಸಿ
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಆವೃತ್ತಿಗಳನ್ನು ವಿವರಿಸಿ

ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮತ್ತು ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಗಳ ಪರಿಚಯ (Introduction to programming and scripting languages)

ಕಂಪ್ಯೂಟರ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಎಂದರೆ, ಸೂಚನೆಗಳನ್ನು (instructions) ಬರೆಯುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಈ ಸೂಚನೆಗಳು ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತವೆ. ಈ ಸೂಚನೆಗಳಿಗೆ ಕೋಡ್ ಎಂತಲೂ ಕರೆಯುತ್ತಾರೆ. ಇವುಗಳನ್ನು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಲ್ಲಿ (programming language) ಬರೆಯಲಾಗುತ್ತದೆ. ಕಂಪ್ಯೂಟರ್ ಈ ಸೂಚನೆಗಳನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ಕಾರ್ಯ ನಿರ್ವಹಿಸಲು ಅಥವಾ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಉಪಯೋಗಿಸುತ್ತದೆ.

ಸ್ಕ್ರಿಪ್ಟ್ ಅಥವಾ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆ ಒಂದು ಕಂಪ್ಯೂಟರ್ ಭಾಷೆಯಾಗಿದ್ದು, ಇದು ಕಂಪೈಲ್ (compile) ಮಾಡದೇ ಕಾರ್ಯಗತಗೊಳ್ಳುವ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿದ ಫೈಲ್‌ನಲ್ಲಿರುವ ಕಮಾಂಡ್‌ಗಳ ಸರಣಿಯಾಗಿದೆ. Perl, PHP, ಮತ್ತು Python ಇವು ಸರ್ವರ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯ ಉತ್ತಮ ಉದಾಹರಣೆ ಆಗಿವೆ. JavaScript ಇದು ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್‌ನ ಅತ್ಯುತ್ತಮ ಉದಾಹರಣೆ ಆಗಿದೆ.

ಸ್ಕ್ರಿಪ್ಟ್‌ನ ಪ್ರಯೋಜನಗಳು (Advantages of scripts)

- ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿದೆ. ಅಗತ್ಯವಿದ್ದರೆ, ಸ್ಕ್ರಿಪ್ಟನ್ನು ವೀಕ್ಷಿಸಲು ಮತ್ತು ಎಡಿಟ್ ಮಾಡಲು ಬಳಕೆದಾರರಿಗೆ ಅವಕಾಶ ನೀಡುತ್ತದೆ.
- ಫೈಲನ್ನು ಕಂಪೈಲ್ ಮಾಡುವ ಅಗತ್ಯವಿಲ್ಲ , ಆದರೆ, ಅಗತ್ಯವಿದ್ದಾಗ ಮಾಡಬಹುದು.
- ಇದು ಕಲಿಯಲು ಮತ್ತು ಬರೆಯಲು ಸುಲಭವಾಗಿದೆ.
- ವಿವಿಧ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳ ನಡುವೆ ಪೋರ್ಟ್ ಮಾಡಲು ಸುಲಭವಾಗಿದೆ.
- ನಿಜವಾದ ಪ್ರೊಗ್ರಾಂಗಿಂತ ಹೆಚ್ಚು ವೇಗವಾಗಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಬಹುದು - ಕೆಲವು ವ್ಯಕ್ತಿಗಳು ಮತ್ತು ಕಂಪನಿಗಳು ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳನ್ನು ನಿಜವಾದ ಪ್ರೊಗ್ರಾಂಗಳ ಮೂಲಮಾದರಿಯಾಗಿ ಬರೆಯುತ್ತಾರೆ.

ಸ್ಕ್ರಿಪ್ಟ್‌ನ ಅನಾನುಕೂಲಗಳು (Disadvantages of scripts)

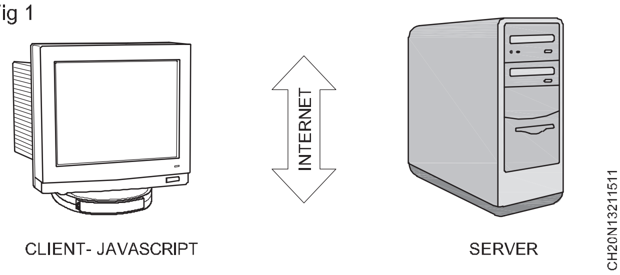
- ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿರುವುದರಿಂದ, ಸೋರ್ಸ್ ಕೋಡನ್ನು ಇತರರು ನೋಡಬಹುದು, ಇದನ್ನು ಕೆಲವು ಕಂಪನಿಗಳು ನಿಷೇಧಿಸಬಹುದು.
- ಸ್ಕ್ರಿಪ್ಟನ್ನು ರನ್ ಮಾಡುವ ಮೊದಲು ಬಳಕೆದಾರರು, ಇಂಟರ್‌ಪ್ರಿಟರ್ ಅಥವಾ ಪ್ರತ್ಯೇಕ ಪ್ರೊಗ್ರಾಂನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.
- ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ, ಇವು ಕಂಪೈಲ್ ಮಾಡಿದ ಪ್ರೊಗ್ರಾಂಗಿಂತ ನಿಧಾನವಾಗಿ ಕೆಲಸ ಮಾಡಬಹುದು.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎಂದರೇನು ? (What is Java Script?)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅತ್ಯಂತ ಶಕ್ತಿಯುತವಾದ ಕ್ಲೈಂಟ್ ಸೈಡ್ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆ ಆಗಿದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಮುಖ್ಯವಾಗಿ ವೆಬ್‌ಪೇಜ್‌ನೊಂದಿಗೆ ಬಳಕೆದಾರರ ಪರಸ್ಪರ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಹೆಚ್ಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (ಚಿತ್ರ 1) ಬೇರೆ ಶಬ್ದಗಳಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಸಹಾಯದಿಂದ ನೀವು ನಿಮ್ಮ ವೆಬ್‌ಪೇಜನ್ನು ಹೆಚ್ಚು ಉತ್ಸಾಹಭರಿತ ಮತ್ತು ಸಂವಾದಾತ್ಮಕವಾಗಿ ಮಾಡಬಹುದು. ಜಾವಾಸ್ಕ್ರಿಪ್‌ನ್ನು ಆಟಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಮತ್ತು ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ವ್ಯಾಪಕವಾಗಿ ಬಳಸಲಾಗುತ್ತಿದೆ.

NOT TO BE REPUBLISHED © NIMI

Fig 1



ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಇತಿಹಾಸ (JavaScript History)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು 1995 ರಲ್ಲಿ ಬ್ರಾಂಡೆನ್ ಇಚ್ (Brendan Eich) ಅಬಿವೃದ್ಧಿಪಡಿಸಿದರು. ಇದು ಆ ಸಮಯದ ಜನಪ್ರೀಯ ಬ್ರೌಸರ್ ಆಗಿದ್ದ ನೆಟ್‌ಸ್ಕೇಪ್‌ನಲ್ಲಿ ಕಾಣಿಸಿಕೊಂಡಿತು. ಆರಂಭದಲ್ಲಿ ಈ ಭಾಷೆಯನ್ನು Live Script ಎಂದು ಕರೆಯಲಾಯಿತು ಮತ್ತು ನಂತರ ಅದನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎಂದು ಮರುನಾಮಕರಣ ಮಾಡಲಾಯಿತು. ಅನೇಕ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮತ್ತು ಜಾವಾ ಒಂದೇ ಎಂದು ಭಾವಿಸಿದ್ದಾರೆ. ಆದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮತ್ತು ಜಾವಾಗೆ ಯಾವುದು ಸಂಬಂಧವಿರುವುದಿಲ್ಲ. ಜಾವಾ ಒಂದು ಬಹಳ ಸಂಕೀರ್ಣವಾದ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ ಆದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೇವಲ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಸಿಂಟ್ಯಾಕ್ಸ್ ಹೆಚ್ಚಾಗಿ C ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಿಂದ ಪ್ರಭಾವಿತವಾಗಿದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಹೇಗೆ ರನ್ ಮಾಡುವುದು? (How to Run JavaScript?)

ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯಾಗಿರುವುದರಿಂದ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೇ ಕಾರ್ಯ ನಿರ್ವಹಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ವಾಸ್ತವವಾಗಿ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೋಡ್‌ನ್ನು ರನ್ ಮಾಡಲು ಬ್ರೌಸರ್‌ನ ಅಗತ್ಯವಿದೆ. ಬಳಕೆದಾರರು HTML ಪುಟವನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟನೊಂದಿಗೆ ವಿನಂತಿಸಿದಾಗ, ಸ್ಕ್ರಿಪ್ಟನ್ನು ಬ್ರೌಸರ್‌ಗೆ ಕಳುಹಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಬ್ರೌಸರ್ ಅದನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ, ಎಲ್ಲ ಅಧುನಿಕ ವೆಬ್ ಬ್ರೌಸರ್‌ಗಳು ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು ಬೆಂಬಲಿಸುತ್ತವೆ. ಅದ್ದರಿಂದ ನೀವು ನಿಮ್ಮ ಸೈಟ್‌ನ್ನು ನೋಡುವವರು Internet Explorer, Google Chrome, Firefox ಅಥವಾ ಇತರ ಯಾವ ಬ್ರೌಸರ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ ಎಂದು ಚಿಂತಿಸಬೇಕಾಗಿಲ್ಲ. ಇವೆಲ್ಲವೂ ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು ಬೆಂಬಲಿಸುತ್ತವೆ. ಇದಲ್ಲದೇ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Windows, Linux or Mac ನ್ನೂ ಒಳಗೊಂಡಂತೆ ಯಾವುದೇ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ರನ್ ಆಗುತ್ತದೆ.

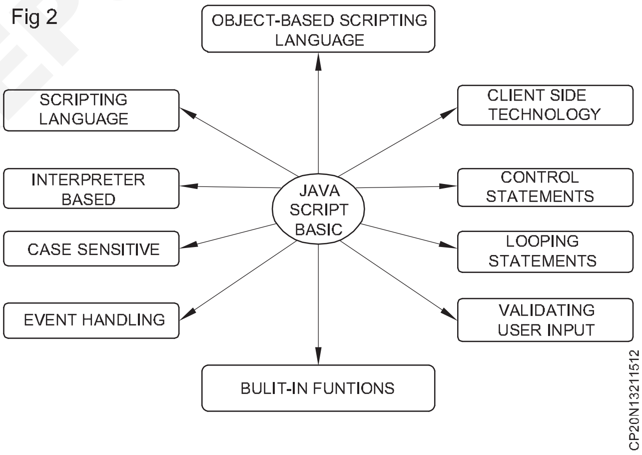
ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು ರನ್ ಮಾಡಲು ನಿಮಗೆ ಅಗತ್ಯವಿರುವ ಸಾಧನಗಳು (Tools You Need to run JavaScript)

ಕೋಡ್(code) ಬರೆಯಲು ನಿಮಗೆ ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನ ಅಗತ್ಯವಿರುತ್ತದೆ ಮತ್ತು ವೆಬ್ ಪುಟಗಳನ್ನು ಪ್ರದರ್ಶಿಸಲು ಒಂದು ಬ್ರೌಸರ್‌ನ ಅಗತ್ಯವಿರುತ್ತದೆ. ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನಲ್ಲಿ ನಿಮಗೆ Notepad++, Visual Studio Code, Sublime Text, Atom ಅಥವಾ ಇತರ ಯಾವುದೇ ಟೆಕ್ಸ್ಟ್ ಎಡಿಟರ್‌ನಂತಹ ಆಯ್ಕೆಗಳಿವೆ. ಮತ್ತು ವೆಬ್ ಬ್ರೌಸರ್‌ಗಳಲ್ಲಿ Google Chrome, Firefox, Microsoft Edge, Internet Explorer ಮುಂತಾದ ಯಾವುದೇ ವೆಬ್ ಬ್ರೌಸರ್ ಉಪಯೋಗಿಸಬಹುದು.

ಮಾದರಿ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಪ್ರೊಗ್ರಾಂ (Sample JavaScript program)

```
<html>
<head>
<title>My First JavaScript code!!!</title>
<script type=“text/javascript”>
alert(“Welcome to JavaScript Program!”);
</script>
</head>
<body>
</body>
</html>
```

Fig 2



ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳು (Features of JavaScript)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕ್ಲೈಂಟ್ ಸೈಡ್ ತಂತ್ರಜ್ಞಾನವಾಗಿದೆ, ಇದನ್ನು ಮುಖ್ಯವಾಗಿ ಕ್ಲೈಂಟ್ ಸೈಡ್ ವ್ಯಾಲಿಡೇಷನ್‌ಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ, ಅದರೆ ಚಿತ್ರ 2 ರಲ್ಲಿ ತೋರಿಸಿದಂತೆ ಇದು ಬಹಳಷ್ಟು ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿದೆ.

- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ - ಆಧಾರಿತ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ.
- ಇದು ಬಳಕೆದಾರರಿಗೆ ಬ್ರೌಸರ್ ಮೇಲೆ ಹೆಚ್ಚಿನ ನಿಯಂತ್ರಣವನ್ನು ನೀಡುತ್ತದೆ.
- ಇದು ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ನಿಭಾಯಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

- ಇದು ಬಳಕೆದಾರರ ಬ್ರೌಸರ್ ಮತ್ತು OS ನ್ನು ಪತ್ತೆ ಮಾಡುತ್ತದೆ.
- ಇದು ಹಗುರವಾಗಿದೆ.
- ಇದು ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ ಮತ್ತು ಇದು ಜಾವಾ ಅಲ್ಲ.
- ಇದು ಇಂಟರ್‌ಪ್ರಿಟರ್ ಆಧಾರಿತ ಸ್ಕ್ರಿಪ್ಟಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ.
- ಇದು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿದೆ.
- ಇದು ಆಬ್ಜೆಕ್ಟ್ ಆಧಾರಿತ ಭಾಷೆಯಾಗಿದೆ. ಇದು predefined ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿರುವ ಪ್ರತಿಯೊಂದು ಹೇಳಿಕೆಯನ್ನು ಸೆಮಿಕೋಲನ್ (;) ನಿಂದ ಕೊನೆಗೊಳಿಸಬೇಕು.
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿರುವ ಹೆಚ್ಚಿನ ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್ (control statements) ಗಳ ಸಿಂಟ್ಯಾಕ್ಸ್ C language ನಲ್ಲಿರುವ ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನಂತೆಯೆ ಇರುತ್ತದೆ.
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಮುಖ್ಯವಾದ ಅಂಶವೆಂದರೆ ಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಹೊಸ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ರಚಿಸುವ ಸಾಮರ್ಥ್ಯ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಪ್ರಯೋಜನಗಳು (Advantages of JavaScript)

- ಕ್ಲೈಂಟ್ ಸೈಡ್‌ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ : ಉದಾಹರಣೆಗೆ, ಬಳಕೆದಾರರು ಸರ್ವರಗೆ ರಿಕ್ವೆಸ್ಟ್ ಕಳುಹಿಸುವ ಮೊದಲು ಯಾವುದೇ ಇನ್ಪುಟ್‌ನ್ನು ವ್ಯಾಲಿಡೇಟ್ ಮಾಡಬಹುದು. ಇದು ಸರ್ವರ್‌ನ ಲೋಡ್ ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.
- ತುಲನಾತ್ಮಕವಾಗಿ ಸುಲಭವಾದ ಭಾಷೆ : ಇದು ಕಲಿಯಲು ತುಂಬಾ ಸುಲಭವಾದ ಭಾಷೆಯಾಗಿದೆ ಮತ್ತು ಇದರ ಸಿಂಟ್ಯಾಕ್ಸ್ ಇಂಗ್ಲಿಷ್‌ಗೆ ಹತ್ತಿರವಾಗಿದೆ.
- ವಿಸಿಟರ್‌ಗೆ ತಕ್ಷಣ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡುತ್ತದೆ : ಯಾವುದೇ ಸರ್ವರ್‌ನ ಸಂವಹನವಿಲ್ಲದೇ , ನೀವು ಬಯಸಿದ ಫಲಿತಾಂಶವನ್ನು ಪಡೆಯಲು ಪುಟವು ರೀಲೋಡ್ ಆಗುವುದನ್ನು ಕಾಯಬೇಕಾಗಿಲ್ಲ.
- ಬಳಕೆದಾರರಿಗೆ ವೇಗವಾಗಿ ಫಲಿತಾಂಶ ನೀಡುತ್ತದೆ : ಬಳಕೆದಾರರ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು execute ಮಾಡಿದಾಗ, ಕಾರ್ಯದ ಆಧಾರದ ಮೇಲೆ ತಕ್ಷಣವೆ ಫಲಿತಾಂಶಗಳು ಸಿಗುತ್ತವೆ.
- ಪರಸ್ಪರ ಸಂವಹನ ಹೆಚ್ಚಾಗಿದೆ : ಬಳಕೆದಾರರು ಇಂಟರ್‌ಫೇಸ್‌ಗಳ ಮೇಲೆ ಸುಳಿದಾಡಿದಾಗ ಪ್ರತಿಕ್ರಿಯಿಸುವಂತೆ ಅವುಗಳನ್ನು ರಚಿಸುತ್ತದೆ ಅಥವಾ ಕೀಬೋರ್ಡ್ ಉಪಯೋಗಿಸಿ ಅವುಗಳನ್ನು ಸಕ್ರೀಯಗೊಳಿಸಬಹುದು.
- ರಿಚ್ ಇಂಟರ್‌ಫೇಸ್ : ಡ್ರ್ಯಾಗ್ ಆ್ಯಂಡ್ ಡ್ರಾಪ್ ಕಾಂಪೋನೆಂಟ್ ಅಥವಾ ಸ್ಲೈಡರ್‌ಗಳು ಸೈಟ್ ವಿಸಿಟರ್‌ಗಳಿಗೆ ಒಂದು ಶ್ರೀಮಂತ ಇಂಟರ್‌ಫೇಸನ್ನು ನೀಡುತ್ತವೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಅನಾನುಕೂಲಗಳು (Disadvantages of JavaScript)

- ಭದ್ರತಾ ಸಮಸ್ಯೆಗಳು : ಯಾವುದೇ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ತುಣುಕುಗಳನ್ನು, ಕ್ಲೈಂಟ ಸೈಡ್‌ನಲ್ಲಿ ವೆಬ್ ಪುಟದಲ್ಲಿ ಸೇರಿಸಿದಾಗ ಬಳಕೆದಾರರ ಸಿಸ್ಟಮ್‌ನ್ನು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಅವುಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಳ್ಳಬಹುದು.
- ಇದು ಯಾವುದೇ ಮಲ್ಟಿಪ್ರೊಸೆಸರ ಅಥವಾ ಮಲ್ಟಿಥ್ರೆಡಿಂಗ್ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಹೊಂದಿಲ್ಲ.
- ಯಾವುದೇ ಬೆಂಬಲವಿಲ್ಲದ ಕಾರಣ ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು ಯಾವುದೇ ನೆಟ್‌ವರ್ಕಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಬಳಸಲಾಗುವುದಿಲ್ಲ.
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ನಮಗೆ ಫೈಲ್‌ಗಳನ್ನು ಓದಲು ಅಥವಾ ಬರೆಯಲು ಅನುಮತಿಸುವುದಿಲ್ಲ.
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ರೆಂಡರ್ ಬದಲಾಗುತ್ತದೆ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ವಿಬಿನ್ನ ಲೇಔಟ್ ಇಂಜಿನ್‌ಗಳಿಂದ ವಿಭಿನ್ನವಾಗಿ ನಿರೂಪಿಸಲಾಗುತ್ತದೆ. ಇದರ ಪರಿಣಾಮವಾಗಿ ಅದರ ಇಂಟರ್‌ಫೇಸ್ ಮತ್ತು ಕಾರ್ಯನಿರ್ವಹಣೆಯಲ್ಲಿ ಸಾಮರಸ್ಯ ಇರುವುದಿಲ್ಲ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆವೃತ್ತಿಗಳು (JavaScript Versions)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ್ನು ಬ್ರೆಂಡನ್ ಇಚ್ ಅವರು 1995 ರಲ್ಲಿ ಕಂಡು ಹಿಡಿದರು ಮತ್ತು 1997 ರಲ್ಲಿ ಇದು ECMA Standard ಆಯಿತು. ECMA ಇದು ಈ ಭಾಷೆಯ ಅಧಿಕೃತ ಹೆಸರು.

2015 ರಿಂದ ECMA ಸ್ಕ್ರಿಪ್ಟ್ ನ್ನು ವರ್ಷದಿಂದ ಹೆಸರಿಸಲಾಯಿತು (ECMA Script 2015).

ECMA Script ಆವೃತ್ತಿಗಳು (ECMA Script Editions)

ಆವೃತ್ತಿ	ಅಧಿಕೃತ ಹೆಸರು	ವಿವರಣೆ
1	ECMA Script 1 (1997)	ಮೊದಲನೇ ಆವೃತ್ತಿ
2	ECMA Script 2 (1998)	ಸಂಪಾದಕೀಯ ಬದಲಾವಣೆಗಳು ಮಾತ್ರ
3	ECMA Script 3 (1999)	Regular Expressions ಗಳನ್ನು ಸೇರಿಸಲಾಯಿತು try/catch ಸೇರಿಸಲಾಯಿತು

NIMI © NOT TO BE REPUBLISHED

4	ECMA Script 4	ಬಿಡುಗಡೆಯಾಗಲಿಲ್ಲ
5	ECMA Script 5 (2009)	"strict mode" ಸೇರಿಸಲಾಯಿತು JSON ಬೆಂಬಲವನ್ನು ಸೇರಿಸಲಾಯಿತು String.trim() ಸೇರಿಸಲಾಯಿತು Array.isArray() ಸೇರಿಸಲಾಯಿತು Array Iteration Methods ಸೇರಿಸಲಾಯಿತು
5.1	ECMA Script 5.1 (2011)	ಸಂಪಾದಕೀಯ ಬದಲಾವಣೆಗಳು
6	ECMA Script 2015	let and const ಸೇರಿಸಲಾಯಿತು default parameter values ಸೇರಿಸಲಾಯಿತು Array.find().Added Array.findIndex() ಸೇರಿಸಲಾಯಿತು
7	ECMA Script 2016	exponential operator (**) ಸೇರಿಸಲಾಯಿತು Array.prototype.includes ಸೇರಿಸಲಾಯಿತು
8	ECMA Script 2017	string padding ಸೇರಿಸಲಾಯಿತು new Object properties ಸೇರಿಸಲಾಯಿತು Async functions.Added Shared Memory ಸೇರಿಸಲಾಯಿತು
9	ECMA Script 2018	rest/spread properties ಸೇರಿಸಲಾಯಿತು Asynchronous iteration ಸೇರಿಸಲಾಯಿತು Promise.finally().Additions to Reg Exp ಸೇರಿಸಲಾಯಿತು

©NIMI NOT TO BE REPUBLISHED

ವೆಬ್ ಸರ್ವರ್‌ಗಳು ಮತ್ತು ಬಾಹ್ಯ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲ್‌ಗಳ ಪರಿಚಯ (Introduction to Web servers and External JavaScript files)

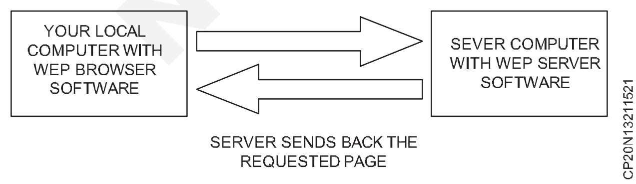
ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ವೆಬ್ ಸರ್ವರ್‌ಗಳು ಮತ್ತು ಅವುಗಳ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ವಿವರಿಸಿ**
- **ಬಾಹ್ಯ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ ಫೈಲ್‌ಗಳನ್ನು ವಿವರಿಸಿ.**

ವೆಬ್ ಸರ್ವರ್ ಎಂದರೇನು ಮತ್ತು ಅದು ಹೇಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ? (What is Web Server and how it works?)

ಯಾವುದೇ ವೆಬ್ ಹೋಸ್ಟಿಂಗ್‌ಗೆ ವೆಬ್ ಸರ್ವರ್‌ಗಳು ಮುಖ್ಯವಾಗಿವೆ. (Fig 1)

Fig 1



ವೆಬ್ ಸರ್ವರ್, ಬಳಕೆದಾರರಿಗೆ ಅವರ ವಿನಂತಿಯ ಮೇರೆಗೆ HTTP ಉಪಯೋಗಿಸಿ ವೆಬ್‌ಪುಟಗಳನ್ನು ರಚಿಸುತ್ತದೆ. ಈ ವಿನಂತಿಗಳನ್ನು ಅವರ ಕಂಪ್ಯೂಟರ್‌ನ HTTP connection ಮೂಲಕ ಕಳುಹಿಸಲಾಗುತ್ತದೆ. XML ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಇನ್ನೊಂದು ಸಾಧನಕ್ಕೆ ಕಳುಹಿಸುವ ಯಾವುದೇ ಸರ್ವರ್‌ಗೆ ವೆಬ್ ಸರ್ವರ ಎನ್ನುವರು. ವೆಬ್ ಸರ್ವರ್ ಎನ್ನುವುದು ವಿಷಯಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು ತಲುಪಿಸಲು HTTP ವಿನಂತಿಗಳಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುವ ಇಂಟರ್‌ನೆಟ್ ಸರ್ವರ್ ಆಗಿದೆ. ವೆಬ್ ಸರ್ವರ್ ಯಾವಾಗಲೂ ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕ ಹೊಂದಿರುತ್ತದೆ. ಇಂಟರ್‌ನೆಟ್‌ಗೆ ಸಂಪರ್ಕ ಹೊಂದಿರುವ ಪ್ರತಿಯೊಂದು ಸರ್ವರ್ ಒಂದು ವಿಶೇಷವಾದ ಅಡ್ರೆಸ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇದು 0 ದಿಂದ 255 ರವರೆಗಿನ ನಾಲ್ಕು ಸಂಖ್ಯೆಗಳ ಸರಣಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಈ ಸಂಖ್ಯೆಗಳನ್ನು ಒಂದು ಪಿರಿಯಡ್ (.) ಪ್ರತ್ಯೇಕಿಸುತ್ತದೆ. ಒಂದೇ ಸರ್ವರ್‌ನಲ್ಲಿನ ಅನೇಕ ಡೊಮೇನ್‌ಗಳನ್ನು(ಬಳಕೆದಾರರು) ನಿರ್ವಹಿಸಲು hosting providers ಗಳನ್ನು ವೆಬ್ ಸರ್ವರ್ ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ವೆಬ್ ಹೋಸ್ಟ್ ಒಂದು ಕಂಪನಿಯಾಗಿದ್ದು , ಜನರು ತಮ್ಮದೇ ಆದ ವಿಷಯಗಳನ್ನು ಮತ್ತು ವೆಬ್‌ಪುಟಗಳನ್ನು ಒದಗಿಸಲು ಸರ್ವರ್‌ಗಳ ಕ್ಲಸ್ಟರ್‌ನಲ್ಲಿ ಸ್ಥಳವನ್ನು ಗುತ್ತಿಗೆ ನೀಡುತ್ತದೆ.

ವಿವಿಧ ರೀತಿಯ ವೆಬ್ ಸರ್ವರ್‌ಗಳು (Different types of web servers)

ವಿವಿಧ ರೀತಿಯ ವೆಬ್ ಸರ್ವರ್‌ಗಳು ಮುಕ್ತ ಮಾರುಕಟ್ಟೆಯಲ್ಲಿ ಲಭ್ಯವಿವೆ. ಅತ್ಯಂತ ಜನಪ್ರೀಯ ವೆಬ್ ಸರ್ವರ್‌ಗಳ ಕುರಿತು ಚರ್ಚಿಸೋಣ. ಅವುಗಳಲ್ಲಿ ಕೆಲವು Apache, IIS, Nginx and Lite Speed ಆಗಿವೆ.

ಅಪಾಚೆ ವೆಬ್ ಸರ್ವರ್ (Apache web server)

ಇದು ಅಪಾಚೆ ಸಾಫ್ಟ್‌ವೇರ್ ಫೌಂಡೇಷನ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಜಗತ್ತಿನ ಅತ್ಯಂತ ಜನಪ್ರೀಯ ವೆಬ್ ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಅಪಾಚೆ ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರ್ ಆಗಿದ್ದು, ಇದು Linux, Unix, Windows, FreeBSD, MacOS X ಮತ್ತು ಇನ್ನೂ ಹೆಚ್ಚಿನ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಬಹುತೇಕ ಎಲ್ಲ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಸುಮಾರು 60% ಯಂತ್ರಗಳು ಅಪಾಚೆ ವೆಬ್ ಸರ್ವರ್‌ನಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ.

Fig 2

ಅಪಾಚೆ ವೆಬ್ ಸರ್ವರ್ ಮಾಡ್ಯುಲರ್ ರಚನೆಯನ್ನು ಹೊಂದಿರುವುದರಿಂದ ಅದರ ಗ್ರಾಹಕೀಕರಣವು ಸುಲಭವಾಗಿದೆ. ಇದು ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿರುವುದರಿಂದ ಅಗತ್ಯವಿರುವಾಗ ಸರ್ವರ್‌ಗೆ ಸ್ವಂತ ಮಾಡ್ಯೂಲ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು ಮತ್ತು ಅವಶ್ಯಕತೆಗಳಿಗೆ ತಕ್ಕಂತೆ ಮಾರ್ಪಾಡುಗಳನ್ನು ಮಾಡಬಹುದು. ಇದು ಇತರ ವೆಬ್ ಸರ್ವರ್‌ಗಳಿಗಿಂತ ಹೆಚ್ಚು ಸ್ಥಿರವಾಗಿದೆ ಮತ್ತು ಆಡಳಿತಾತ್ಮಕ ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಹರಿಸಲು ಸುಲಭವಾಗಿದೆ. ಇದನ್ನು ಅನೇಕ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಯಶಸ್ವಿಯಾಗಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಬಹುದು. ಹಿಂದಿನ ಆವೃತ್ತಿಗಳಿಗೆ ಹೋಲಿಸಿದಾಗ ಅಪಾಚೆ ಬಿಡುಗಡೆಗೊಳಿಸಿದ ಇತ್ತೀಚಿನ ಆವೃತ್ತಿಗಳು ಹೆಚ್ಚು ವಿನಂತಿಗಳನ್ನು ನಿರ್ವಹಿಸುವ ಕಾರ್ಯಸಾಧ್ಯತೆಯನ್ನು ಒದಗಿಸುತ್ತವೆ.

ಐಐಎಸ್ ವೆಬ್ ಸರ್ವರ್ (IIS web server)

ಐಐಎಸ್ ಒಂದು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಉತ್ಪನ್ನವಾಗಿದೆ. ಈ ಸರ್ವರ್ ಅಪಾಚೆಯಂತೆಯೆ ಎಲ್ಲ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಹೊಂದಿದೆ. ಆದರೆ ಇದು ಓಪನ್ ಸೋರ್ಸ್ ಅಲ್ಲ ಮತ್ತು ಅದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚಾಗಿ ವೈಯಕ್ತಿಕ ಮಾಡ್ಯೂಲ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಸುಲಭವಲ್ಲ ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡುವುದು ಕಷ್ಟದ ಕೆಲಸವಾಗುತ್ತದೆ. (Fig 3)

Fig 3

ಮೈಕ್ರೊಸಾಫ್ಟ್ ಈ ಉತ್ಪನ್ನವನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿದೆ ಮತ್ತು ಇದನ್ನು ಅವರು ನಿರ್ವಹಿಸುತ್ತಾರೆ, ಹೀಗಾಗಿ ಇದು ಎಲ್ಲ ವಿಂಡೊಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದಲ್ಲದೇ ಯಾವುದೇ ಸಮಸ್ಯೆಗಳಿದ್ದಲ್ಲಿ ಅವರು ಗ್ರಾಹಕರಿಗೆ ಉತ್ತಮ ಬೆಂಬಲವನ್ನು ಒದಗಿಸುತ್ತಾರೆ.

Nginx web server

Nginx ಇದು ಉಚಿತ ಓಪನ್ ಸೋರ್ಸ್ ವೆಬ್ ಸರ್ವರ್ ಆಗಿದೆ, ಇದು IMAP/POP3 ಪ್ರಾಕ್ಸಿ ಸರ್ವರ್‌ನ್ನು ಒಳಗೊಂಡಿದೆ. Nginx ಅದರ ಉನ್ನತ ಕಾರ್ಯಕ್ಷಮತೆ, ಸ್ಥಿರತೆ, ಸರಳ ಸಂರಚನೆ ಮತ್ತು ಕಡಿಮೆ ಸಂಪನ್ಮೂಲ ಬಳಕೆಗಳಿಗೆ ಹೆಸರುವಾಸಿಯಾಗಿದೆ. (Fig 4)

Fig 3

ಇದು ವಿನಂತಿಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಥ್ರೆಡ್‌ಗಳನ್ನು ಬಳಸುವುದಿಲ್ಲ. ಬದಲಾಗಿ ಲೋಡ್ ಇದ್ದಾಗ ಚಿಕ್ಕದಾದ ಮತ್ತು ಊಹಿಸಬಹುದಾದ ಗಾತ್ರದ ಮೆಮರಿಯನ್ನು ಬಳಸುವ ಇವೆಂಟ್ ಡ್ರಿವನ್ ರಚನೆಯನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ಹೆಚ್ಚು ಜನಪ್ರೀಯವಾಗುತ್ತಿದೆ ಮತ್ತು ಪ್ರಪಂಚದಾದ್ಯಂತ ಎಲ್ಲ ಡೊಮೇನ್‌ಗಳಲ್ಲಿ 7.5% ನಷ್ಟು ಹೋಸ್ಟ್ ಮಾಡುತ್ತಿದೆ. ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ, ಹೆಚ್ಚಿನ ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ ಕಂಪನಿಗಳು ಇದನ್ನು ಬಳಸುತ್ತಿವೆ.

ಬಾಹ್ಯ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲ್‌ಗಳು (External JavaScript files) : ಜಾವಾಸ್ಕ್ರಿಪ್‌ನ್ನು HTML ಕೋಡ್‌ನಲ್ಲಿ ಬರೆಯುವುದು ಕೆಲವೊಮ್ಮೆ ಗೊಂದಲವನ್ನು ಸೃಷ್ಟಿಸುತ್ತದೆ, ಮತ್ತು HTML ಫೈಲನ್ನು ಬದಲಾಯಿಸುವುದು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲುಗಳ ಮೇಲೂ ಪರಿಣಾಮ ಬೀರಬಹುದು. ಅದ್ದರಿಂದ HTML ಮತ್ತು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲುಗಳನ್ನು ಬೇರ್ಪಡಿಸುವುದು ಉತ್ತಮ. ಇದರಿಂದ ಒಂದು ಫೈಲ್‌ನಲಿ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುವುದರಿಂದ ಇನ್ನೊಂದು ಫೈಲ್ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ.

© NIMI NOT TO BE REPUBLISHED

ಬಾಹ್ಯ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲ್‌ಗಳನ್ನು ಪ್ರತ್ಯೇಕವಾಗಿ ಈ ಕೆಳಗಿನಂತೆ ಬರೆಯಬೇಕು :-

myjs.js ಫೈಲ್ ಹೀಗಿದೆ :

```
function popup() {
alert(“Hello World”);
}
```

ಈಗ ನಾವು ಈ ಫೈಲನ್ನು HTML ಫೈಲ್‌ನಲ್ಲಿ ಈ ಕೆಳಗಿನಂತೆ ಇಂಪೋರ್ಟ್(ಆಮದು) ಮಾಡಬಹುದು :

ಬಾಹ್ಯ ಫೈಲನ್ನು ಸುಲಭವಾಗಿ ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ಮೊದಲು, ನೀವು ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳುತ್ತಿರುವ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫೈಲ್ ಮಾನ್ಯವಾಗಿರಬೇಕು ಮತ್ತು ಕೇವಲ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಗಿರಬೇಕು. ಎರಡನೇಯದಾಗಿ, ಫೈಲ್‌ನ ಎಕ್ಸ್ಟೆನ್ಷನ್ (extension) “ .js “ ಹೊಂದಿರಬೇಕು. ಕೊನೆಯದಾಗಿ, ನಿಮಗೆ ಫೈಲ್ ಯಾವ ಸ್ಥಳದಲ್ಲಿದೆ ಎಂಬುದನ್ನು ತಿಳಿದಿರಬೇಕು.

alert ಫಂಕ್ಷನ್‌ನಲ್ಲಿ Hello World ಎನ್ನುವ ಒಂದು ಸಾಲನ್ನು ಹೊಂದಿರುವ “myjs.js” ಫೈಲನ್ನು ಹೊಂದಿದ್ದೇವೆ ಎಂದು ಭಾವಿಸೋಣ. ಅಂತೆಯೆ ನಾವು ಕೋಡ್ ಬರೆಯಲಿರುವ HTML ಫೈಲ್ ಇರುವ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿಯೆ ಈ ಫೈಲ್ ಇದೆ ಎಂದು ಭಾವಿಸೋಣ. ಈ ಫೈಲನ್ನು ಇಂಪೋರ್ಟ್ ಮಾಡಲು HTML ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಈ ಕೆಳಗಿನಂತೆ ಕೋಡ್ ಬರೆಯಬೇಕು :

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೋಡ್ :

```
<html>
<head>
<script src = “myjs.js”>
</script>
</head>
<body>
<input type= “button” onclick = “popup()” value = “Click Me!”>
</body>
</html>
```

ಈಗ ಈ HTML ಫೈಲ್ myjs.js ನ್ನು ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಇದರ ಪರಿಣಾಮವಾಗಿ HTML ಬಟನ್ ಎಲಿಮೆಂಟ್‌ನಿಂದ popup() ಫಂಕ್ಷನ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು HTML ಪುಟಾಗಳೊಂದಿಗೆ ಸೇರಿಸುವುದು (JavaScript Embed JavaScript in HTML Pages)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್ ಮತ್ತು ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Using JavaScript Variable and data types)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್ ವಿವರಿಸಿ
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿನ ವಿವಿಧ ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ವಿವರಿಸಿ.

ವೇರಿಯೇಬಲ್ (Variables)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಡಾಟಾ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸುವ ಕಂಟೇನರ್‌ಗಳಾಗಿವೆ. ಉದಾಹರಣೆ 1 ರಲ್ಲಿ a, b ಮತ್ತು c ಇವು ವೆರಿಯೇಬಲ್‌ಗಳಾಗಿವೆ :

ಉದಾಹರಣೆ 1

```
var a = 12;
var b = 10;
var c = a + b;
```

ಉದಾಹರಣೆ 1, ರಿಂದ ನಾವು ಈ ರೀತಿ ತಿಳಿದುಕೊಳ್ಳಬಹುದು

- a stores the value 12
- b stores the value 10
- c stores the value 22

ಉದಾಹರಣೆ 2, ರಲ್ಲಿ mark1, mark2, ಮತ್ತು total, ವೇರಿಯೇಬಲ್‌ಗಳಾಗಿವೆ

ಉದಾಹರಣೆ 2 :

```
var mark1 = 85;
var mark2 = 66;
var total = marks1 + mark2;
```

ಬೀಜಗಣೀತದಂತೆಯೆ, ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ mark1 ಮತ್ತು mark2 ಸಂಖ್ಯೆಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ನಾವು ವೆರಿಯೇಬಲ್‌ಗಳನ್ನು ಬಳಸುತ್ತೇವೆ ಮತ್ತು total = mark1 + mark2 ನಂತಹ ಎಕ್ಸ್‌ಪ್ರೆಶನ್‌ಗಳಲ್ಲಿ ವೆರಿಯೇಬಲ್‌ಗಳನ್ನು ಬಳಸುತ್ತೇವೆ. ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ total 151 ಎಂದು ಲೆಕ್ಕ ಮಾಡುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಐಡೆಂಟಿಫೈರ್‌ಗಳು (JavaScript Identifiers)

ಎಲ್ಲ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ವಿಶಿಷ್ಟವಾದ ಹೆಸರುಗಳಿಂದ ಗುರುತಿಸಲಾಗುತ್ತದೆ. ಈ ವಿಶಿಷ್ಟ ಹೆಸರುಗಳಿಗೆ ಐಡೆಂಟಿಫೈರ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಐಡೆಂಟಿಫೈರ್‌ಗಳು ಚಿಕ್ಕ ಹೆಸರುಗಳಾಗಿರಬಹುದು ಉದಾಹರಣೆಗೆ a ಮತ್ತು b ಅಥವಾ ವಿವರಣಾತ್ಮಕ ಹೆಸರುಗಳಾಗಿರಬಹುದು ಉದಾಹರಣೆಗೆ : mark1, mark2, total, age, sum, total volume.

ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ಹೆಸರುಗಳನ್ನು ನೀಡಲು ಸಾಮಾನ್ಯ ನಿಯಮಗಳು :

- ಹೆಸರುಗಳು - ಅಕ್ಷರಗಳು, ಸಂಖ್ಯೆಗಳು, ಅಂಡರ್‌ಸ್ಕೋರ್ (_) ಮತ್ತು ಡಾಲರ್ ಚಿಹ್ನೆ ($) ಗಳನ್ನು ಹೊಂದಿರಬಹುದು.
- ಹೆಸರುಗಳು - ಅಕ್ಷರದಿಂದ ಪ್ರಾರಂಭವಾಗಬೇಕು.
- ಹೆಸರುಗಳು - $ ಮತ್ತು _ ನಿಂದ ಪ್ರಾರಂಭವಾಗಬಹುದು.
- ಹೆಸರುಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿರುತ್ತವೆ (a and A ವಿಭಿನ್ನವಾದ ವೇರಿಯೇಬಲ್‌ಗಳಾಗಿವೆ).
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೀವರ್ಡ್‌ಗಳಂತಹ ರಿಸರ್ವ್ಡ್ ಶಬ್ದಗಳನ್ನು ಹೆಸರುಗಳನ್ನಾಗಿ ಬಳಸಲಾಗುವುದಿಲ್ಲ.

ಸೂಚನೆ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಐಡೆಂಟಿಫೈರ್‌ಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿರುತ್ತವೆ

ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್ (The Assignment Operator)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಸಮಾನ ಚಿಹ್ನೆ (=) ಯನ್ನು "assignment" ಆಪರೇಟರ್ ಆಗಿದೆ, ಇದು "equal to" ಆಪರೇಟರ್ ಅಲ್ಲ.

```
x = x + 10;
```

ಇದು x + 10 ನ ಮೌಲ್ಯವನ್ನು x ಗೆ ಅಸೈನ್ ಮಾಡುತ್ತದೆ. ಇದು x + 10 ನ ಮೌಲ್ಯವನ್ನು ಲೆಕ್ಕ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅದರ ಫಲಿತಾಂಶವನ್ನು x ಗೆ ಕಳುಹಿಸುತ್ತದೆ. x ನ ಮೌಲ್ಯವು 10 ರಿಂದ ಹೆಚ್ಚಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಡಾಟಾ ಟೈಪ್‌ಗಳು (JavaScript Data Types)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳು 100 ನಂತಹ ಸಂಖ್ಯೆಗಳನ್ನು ಮತ್ತು "Santhosh kumar" ನಂತಹ ಟೆಕ್ಸ್ಟನ್ನು ಸಂಗ್ರಹಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ, ಟೆಕ್ಸ್ಟ್ ಮೌಲ್ಯಗಳನ್ನು ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅನೇಕ ರೀತಿಯ ಡಾಟಾಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು. ಆದರೆ ಈಗ ಕೇವಲ ಸಂಖ್ಯೆಗಳು ಮತ್ತು ಸ್ಟ್ರಿಂಗ್‌ಗಳ ಬಗ್ಗೆ ಮಾತ್ರ ತಿಳಿಯೋಣ. ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಡಬಲ್ ಅಥವಾ ಸಿಂಗಲ್ ಕೋಟ್‌ಗಳಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಸಂಖ್ಯೆಗಳನ್ನು ಕೋಟ್‌ಗಳಲ್ಲಿ ಬರೆಯಲಾಗುವುದಿಲ್ಲ. ಒಂದು ವೇಳೆ ಸಂಖ್ಯೆಯನ್ನು ಕೋಟ್‌ನಲ್ಲಿ ಬರೆದರೆ, ಅದನ್ನು ಸ್ಟ್ರಿಂಗ್ ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 3

```
var pi = 3.14;
var person = "santhoshkumar";
var city = "coimbatore";
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು (Declaring JavaScript Variables)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದನ್ನು ವೇರಿಯೇಬಲ್ ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು ಎನ್ನುತ್ತಾರೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ನ್ನು var ಕೀವರ್ಡ್‌ನಿಂದ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾಗುತ್ತದೆ.

```
var traineeName;
```

ಡಿಕ್ಲೇರೇಷನ್ ಆದ ನಂತರ, ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಯಾವುದೇ ಮೌಲ್ಯವಿಲ್ಲ. ತಾಂತ್ರಿಕವಾಗಿ undefined ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ. ವೇರಿಯೇಬಲ್‌ಗೆ ಒಂದು ಮೌಲ್ಯವನ್ನು ಅಸೈನ್ ಮಾಡಲು ಇಕ್ವಲ್ ಚಿಹ್ನೆಯನ್ನು ಉಪಯೋಗಿಸಿ.

```
traineeName = "Santhosh Kumar";
```

ನೀವು ವೇರಿಯೇಬಲ್ ಡಿಕ್ಲೇರ್ ಮಾಡುವಾಗಲೇ ಒಂದು ಮೌಲ್ಯವನ್ನು ಅದಕ್ಕೆ ಅಸೈನ್ ಮಾಡಬಹುದು.

```
var traineeName = "Santhosh Kumar";
```

ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾವು traineeName ಎನ್ನುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ರಚಿಸುತ್ತೇವೆ ಮತ್ತು "Santhosh Kumar" ಎನ್ನುವ ಮೌಲ್ಯವನ್ನು ಅದಕ್ಕೆ ಅಸೈನ್ ಮಾಡುತ್ತೇವೆ.

ನಂತರ id= "demo" ನೊಂದಿಗೆ HTML ಪ್ಯಾರಾಗ್ರಾಫ್‌ನಲ್ಲಿ ಮೌಲ್ಯವನ್ನು ಔಟ್‌ಪುಟ್ ಮಾಡುತ್ತೇವೆ.

```
<p id = "demo"></p>
<script>
var traineeName = "santhoshkumar";
document.get ElementById ("demo").inner
HTML = traineeName;
</script>
```

> **ಸೂಚನೆ: ಸ್ಕ್ರಿಪ್ಟ್‌ನ ಆರಂಭದಲ್ಲಿಯೇ ಎಲ್ಲ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನುಡಿಕ್ಲೇರ್ ಮಾಡುವುದು ಉತ್ತಮ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ ಅಭ್ಯಾಸವಾಗಿದೆ.**

ಒಂದೇ ಸ್ಟೇಟಮೆಂಟ್‌ನಲ್ಲಿ ನೀವು ಅನೇಕ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಬಹುದು. ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು var ನೊಂದಿಗೆ ಪ್ರಾರಂಬಿಸಬೇಕು ಮತ್ತು ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ (,) ಪ್ರತ್ಯೇಕಿಸಬೇಕು.

ಉದಾಹರಣೆ 4

```
var traineeName = "santhoshkumar", city = "coimbatore",
total= "151";
```

Undefined ಮೌಲ್ಯ (Undefined value)

ಕಂಪ್ಯೂಟರ್ ಪ್ರೊಗ್ರಾಂಗಳಲ್ಲಿ, ಸಾಮಾನ್ಯವಾಗಿ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಯಾವುದೇ ಮೌಲ್ಯವಿಲ್ಲದೇ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾಗುತ್ತದೆ. ಮೌಲ್ಯವನ್ನು ಲೆಕ್ಕ ಮಾಡಲಾಗುತ್ತದೆ, ಅಥವಾ ಅದನ್ನು ಬಳಕೆದಾರರು ನೀಡುವ ಇನ್ಪುಟ್ ಆಗಿ ಅನಂತರ ಒದಗಿಸಲಾಗುತ್ತದೆ.

ಯಾವುದೇ ಮೌಲ್ಯವನ್ನು ಒದಗಿಸದೇ ಡಿಕ್ಲೇರ್ ಮಾಡಿದ ವೇರಿಯೇಬಲ್‌ನ್ನು **undefined** ಎನ್ನುವ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಈ ಕೆಳಗಿನ ಸ್ಟೇಟಮೆಂಟ್ ಕಾರ್ಯನಿರ್ವಹಣೆ ಆದ ಮೇಲೆ **traineeName** ವೇರಿಯೇಬಲ್ undefined ಎನ್ನುವ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 5

```
var traineeName;
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ರೀ-ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು (Re-Declaring JavaScript Variables)

ಒಂದು ವೇಳೆ ನೀವು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ನ್ನು ರೀ ಡಿಕ್ಲೇರ್ ಮಾಡಿದರೆ, ಅದು ತನ್ನ ಮೌಲ್ಯವನ್ನು ಕಳೆದುಕೊಳ್ಳುವುದಿಲ್ಲ. ಕೆಳಗಿನ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಎಕ್ಸಿಕ್ಯೂಟ್ ಮಾಡಿದಾಗ traineeName ವೇರಿಯೇಬಲ್ ಇನ್ನೂ "santhoshkumar" ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 6

```
var traineeName = "santhoshkumar";
var traineeName;
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅಂಕಗಣಿತ (JavaScript Arithmetic)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳಲ್ಲಿ, = ಮತ್ತು + ನಂತಹ ಆಪರೇಟರ್‌ಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಅಂಕಗಣಿತವನ್ನು ಮಾಡಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 7

```
var x = 8 + 2 + 5;
```

ಈಗ x 15 ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ.

ನೀವು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಕೂಡ ಸೇರಿಸಬಹುದು, ಆದರೆ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಜೋಡಿಸಲಾಗುತ್ತದೆ (concatenated) :

ಉದಾಹರಣೆ 8

```
var x = "Dharani" + " " + "Shree"
```

ಈಗ X ನ ಮೌಲ್ಯ Dharani Shree ಆಗುತ್ತದೆ.

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯ ಫಲಿತಾಂಶ 725 ನೀಡುತ್ತದೆ.

ಉದಾಹರಣೆ 9

```
var x = "7" + 2 + 5;
```

> ಸೂಚನೆ : ನೀವು ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಕೋಟ್‌ಗಳಲ್ಲಿ ಹಾಕಿದರೆ, ಉಳಿದ ಸಂಖ್ಯೆಗಳನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನಾಗಿ ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಅವುಗಳನ್ನು ಜೋಡಿಸಲಾಗುತ್ತದೆ.(concatenated)

ಕೆಳಗಿನ ಉದಾಹರಣೆಯ ಫಲಿತಾಂಶ 75 ಆಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 10

```
var x = 3 + 4 + "5";
```

ಡಾಟಾ ಟೈಪ್‌ಗಳು (Data types)

ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ, ಡಾಟಾ ಟೈಪ್ ಎನ್ನುವುದು ಒಂದು ಪ್ರಮುಖವಾದ ಪರಿಕಲ್ಪನೆಯಾಗಿದೆ. ವೇರಿಯೇಬಲ್‌ಗಳ ಮೇಲೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸಲು ಡಾಟಾದ ಪ್ರಕಾರವನ್ನು ತಿಳಿದುಕೊಳ್ಳುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಅನೇಕ ರೀತಿಯ ಡಾಟಾಗಳನ್ನು ಹೊಂದಿರಬಹುದು ಉದಾ : ಸಂಖ್ಯೆಗಳು, ಸ್ಟ್ರಿಂಗ್, ಆಬ್ಜೆಕ್ಟ್ ಮುಂತಾದವುಗಳು.

ಉದಾಹರಣೆ 11

```
var side = 10;                          // Number
var firstName = "Rithika";              // String
var x = {firstName: "Harini" lastName:"Kumar"};   // Object
```

ಡಾಟಾಟೈಪ್‌ಗಳು ಇಲ್ಲದೆ ಕಂಪ್ಯೂಟರ್ ಇದನ್ನು ಸರಿಯಾಗಿ ಪರಿಹರಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಉದಾಹರಣೆ 12

```
var a = 10 + "Apple";
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೇಲಿನ ಉದಾಹರಣೆಯನ್ನು ಈ ರೀತಿಯಾಗಿ ಪರಿಗಣಿಸುತ್ತದೆ.

```
var a = "10"+ "Apple";
```

ಇದರ ಔಟಪುಟ್ 10 Apple ಆಗಿರುತ್ತದೆ.

> **ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗೆ ಸೇರಿಸುವಾಗ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಯನ್ನು ಸ್ಟ್ರಿಂಗ್ ಎಂದು ಪರಿಗಣಿಸುತ್ತದೆ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ಗಳನ್ನು ಎಡದಿಂದ ಬಲಕ್ಕೆ ಮೌಲ್ಯಮಾಪನ ಮಾಡುತ್ತವೆ. ವಿಭಿನ್ನ ಅನುಕ್ರಮಗಳು ವಿಭಿನ್ನ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತವೆ.

ಉದಾಹರಣೆ 13

```
var y = 20 + 5 + "Apple";
```

The result is 25Apple

ಉದಾಹರಣೆ 14

```
var y = "Apple"+20 + 5 ;
```

The result is Apple205.

> **ಗಮನಿಸಿ : ಮೊದಲನೇ ಉದಾಹರಣೆಯಲ್ಲಿ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ "Apple"ತಲುಪುವವರೆಗೆ 20 ಮತ್ತು 5 ನ್ನು ಸಂಖ್ಯೆಗಳನ್ನಾಗಿ ಪರಿಗಣಿಸುತ್ತದೆ. ಎರಡನೇಯ ಉದಾಹರಣೆಯಲ್ಲಿ, ಮೊದಲನೇಯ ಆಪರಂಡ್ ಸ್ಟ್ರಿಂಗ್ ಆಗಿರುವುದರಿಂದ, ಉಳಿದ ಎಲ್ಲ ಆಪರಂಡ್‌ಗಳನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನಾಗಿ ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.**

ಡೈನಾಮಿಕ್ ಡಾಟಾ ಪ್ರಕಾರಗಳು (Dynamic data types)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಡೈನಾಮಿಕ್ ಡಾಟಾ ಪ್ರಕಾರಗಳನ್ನು ಹೊಂದಿದೆ. ಇದರರ್ಥ ವಿಭಿನ್ನ ರೀತಿಯ ಡಾಟಾ ಪ್ರಕಾರಗಳನ್ನು ಸಂಗ್ರಹಿಸಿಡಲು ಒಂದೇ ವೇರಿಯೇಬಲ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 15

```
var z;          // ಈಗ z undefined ಆಗಿದೆ
z = 10;         // ಈಗ z Number ಆಗಿದೆ
z = "Sakthi";   // ಈಗ z String ಆಗಿದೆ
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳು (JavaScript Strings)

ಒಂದು ಸ್ಟ್ರಿಂಗ್ ಅಥವಾ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್ ಎಂದರೆ "HariniKumar" ನಂತಹ ಅಕ್ಷರಗಳ (characters) ಸರಣಿಯಾಗಿದೆ. ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಕೋಟ್‌ಗಳಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ನೀವು ಡಬಲ್ ಅಥವಾ ಸಿಂಗಲ್ ಕೋಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 16

var bikeName = "Yamaha R15"; // Using double quotes

var bikeName = ' Yamaha R15'; // Using single quotes

ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ ಒಳಗಡೆಯೂ ನೀವು ಕೋಟ್‌ಗಳನ್ನು ಬಳಸಬಹುದು, ಆದರೆ ಅವು ಸ್ಟ್ರಿಂಗ್‌ನ ಹೊರಗಡೆ ಬಳಸಿದ ಕೋಟ್‌ಗಳಿಗೆ ಸಮನಾಗಿರಬಾರದು :

ಉದಾಹರಣೆ 17

var answer = "It's OK"; // Single quote inside double quotes

var answer = ' Patel is called "Iron Man" '; // Double quotes inside single quotes

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಗಳು (JavaScript Numbers)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಒಂದೇ ರೀತಿಯ ಸಂಖ್ಯೆಯನ್ನು ಹೊಂದಿದೆ. ಸಂಖ್ಯೆಗಳನ್ನು ದಶಮಾಂಶದೊಂದಿಗೆ ಅಥವಾ ದಶಮಾಂಶವಿಲ್ಲದೇ ಬರೆಯಬಹುದು.

ಉದಾಹರಣೆ 18

var num1 = 87.0; // ದಶಮಾಂಶದೊಂದಿಗೆ ಬರೆಯಲಾಗಿದೆ.

var num2 = 87; // ದಶಮಾಂಶವಿಲ್ಲದೇ ಬರೆಯಲಾಗಿದೆ.

ಅತೀ ದೊಡ್ಡ ಸಂಖ್ಯೆಗಳನ್ನು ಅಥವಾ ಅತೀ ಚಿಕ್ಕ ಸಂಖ್ಯೆಗಳನ್ನು ವೈಜ್ಞಾನಿಕ ಸಂಕೇತದೊಂದಿಗೆ (exponential) ಬರೆಯಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 19

var exp1 = 232e5; // ಉತ್ತರ 23200000 ಆಗಿರುತ್ತದೆ

var z = 123e-5; // ಉತ್ತರ 0.00232 ಆಗಿರುತ್ತದೆ

ಉದಾಹರಣೆ 20

var p = 3:

var q = 3;

var r = 5;

(p == q) // true ಎಂದು ಉತ್ತರ ನೀಡುತ್ತದೆ.

(p == r) // false ಎಂದು ಉತ್ತರ ನೀಡುತ್ತದೆ.

> **ಗಮನಿಸಿ : ಬೂಲಿನ್‌ಗಳನ್ನು ಹೆಚ್ಚಾಗಿ ಷರತ್ತುಗಳನ್ನು ಪರೀಕ್ಷಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳು (JavaScript Arrays)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳನ್ನು ಚದರ ಆವರಣದಲ್ಲಿ (square brackets) ಬರೆಯಲಾಗುತ್ತದೆ. ಅರೇನಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಲಾಗುತ್ತದೆ. ಈ ಕೆಳಗಿನ ಕೋಡ್ ಮೂರು ಐಟಂಗಳನ್ನು (ಬೈಕ್ ಹೆಸರುಗಳನ್ನು) ಹೊಂದಿರುವ bike ಎನ್ನುವ ಅರೇಯನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುತ್ತದೆ (ರಚಿಸುತ್ತದೆ).

ಉದಾಹರಣೆ 21

var bikes = ["Yamaha", "TVS", "Royal Enfield"];

> **ಗಮನಿಸಿ: ಅರೇ ಇಂಡೆಕ್ಸ್‌ಗಳು ಶೂನ್ಯ-ಆಧಾರಿತವಾಗಿವೆ, ಅಂದರೆ ಮೊದಲನೆ ಐಟಂ [0], ಎರಡನೆ ಐಟಂ [1] ಮತ್ತು ಇತ್ಯಾದಿ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು (JavaScript Objects)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಉಷ್ಣಾವರಣದಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಆಬ್ಜೆಕ್ಟ್‌ನ ಗುಣಲಕ್ಷಣಗಳನ್ನು (properties) ಈ ರೀತಿಯಾಗಿ ಬರೆಯಲಾಗುತ್ತದೆ:

name:value pairs, ಇವುಗಳನ್ನು ಅಲ್ಪವಿರಾಮ ಚಿನ್ಹೆಯಿಂದ (,) ಪ್ರತ್ಯೇಕಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 22

var personName = {firstName:"Harini", lastName:
"Kumar", age:13, height:
"155 cms"};

ಮೇಲಿನ ಉದಾಹರಣೆ 22 ರಲ್ಲಿನ , ಆಬ್ಜೆಕ್ಟ್ (personName) 4 ಗುಣಲಕ್ಷಣಗಳನ್ನು (properties) ಹೊಂದಿದೆ : firstName, lastName, age ಮತ್ತು height.

ಟೈಪ್ ಆಫ್ ಆಪರೇಟರ್ (The typeof Operator)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಟೈಪ್ ಆಫ್ ಆಪರೇಟರ್‌ನ್ನು, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ನ ಡಾಟಾಟೈಪ್ (ಡಾಟಾದ ಪ್ರಕಾರ) ನ್ನು ಕಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಟೈಪ್‌ಆಫ್ ಆಪರೇಟರ್ ವೇರಿಯೇಬಲ್‌ನ ಟೈಪ್(ಪ್ರಕಾರ) ಅಥವಾ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ ಟೈಪನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 23

typeof "" // "string" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof "Rithika" // "string" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof "Harini Kumar" // "string" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof0 // "number" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof81 // "number" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof8.14 // "number" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

typeof(3+2) // "number" ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಅನ್‌ಡಿಫೈನ್ಡ್ (Undefined)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಯಾವುದೇ ಮೌಲ್ಯವನ್ನು ಹೊಂದದೇ ಇರುವ ವೇರಿಯೇಬಲ್, undefined ಎನ್ನುವ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ. ಇದರ ಟೈಪ್‌ಆಫ್ ಕೂಡ undefined ಆಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 24

var bike; // ಮೌಲ್ಯ undefined ಆಗಿರುತ್ತದೆ , ಟೈಪ್ undefined ಆಗಿರುತ್ತದೆ.

ಗಮನಿಸಿ : ಯಾವುದೇ ವೇರಿಯೇಬಲ್‌ನ್ನು ಅದರ ಮೌಲ್ಯವನ್ನು undefined ಎಂದು ಸೆಟ್ ಮಾಡುವುದರ ಮೂಲಕ ಖಾಲಿ ಮಾಡಬಹುದು. ಅದರ ಟೈಪ್ ಕೂಡ undefined ಆಗಿರುತ್ತದೆ.

ಖಾಲಿ ಮೌಲ್ಯಗಳು (Empty Values)

Empty Values ಮತ್ತು undefined ಎರಡೂ ವಿಭಿನ್ನವಾಗಿವೆ. Empty Values ಸ್ಟ್ರಿಂಗ್ ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಮತ್ತು ಟೈಪ್ ಎರಡನ್ನೂ ಹೊಂದಿದೆ.

ಉದಾಹರಣೆ 25

var bike = ""; // ಮೌಲ್ಯ "" ಆಗಿರುತ್ತದೆ, ಟೈಪ್‌ಆಫ್ "string" ಆಗಿರುತ್ತ಼

ನಲ್ (Null)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ನಲ್ ಎಂದರೆ "nothing"(ಏನೂ ಇಲ್ಲ). ಇದು ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲ ಎಂದು ಸೂಚಿಸುತ್ತದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, null ನ ಡಾಟಾಟೈಪ್ ಆಬ್ಜೆಕ್ಟ್ ಆಗಿರುತ್ತದೆ. ನೀವು ಯಾವುದೇ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು null ಎಂದು ಸೆಟ್ ಮಾಡುವ ಮೂಲಕ empty ಮಾಡಬಹುದು.

ಉದಾಹರಣೆ 26

var personName = {firstName:"Harini", last Name: "Kumar", age:13, height:"155 cms"};

personName = null; //ಈಗ ಮೌಲ್ಯವು null ಆಗಿದೆ , ಆದರೆ ಟೈಪ್ ಇನ್ನೂ object ಆಗಿರುತ್ತದೆ.

ನೀವು ಒಂದು ಆಬ್ಜೆಕ್ಟನ್ನು undefined ಎಂದು ಸೆಟ್ ಮಾಡುವ ಮೂಲಕ empty ಮಾಡಬಹುದು.

ಉದಾಹರಣೆ 27

var personName = {firstName: "Harini", lastName: "Kumar", age:13, height:"155 cms"};

personName = undefined; // ಈಗ ಮೌಲ್ಯ ಹಾಗೂ ಟೈಪ್ ಎರಡೂ undefined ಆಗಿರುತ್ತವೆ.

Undefined ಮತ್ತು Null ನಡುವಿನ ವ್ಯತ್ಯಾಸ (Difference Between Undefined and Null)

Undefined ಮತ್ತು null ಇವು ಮೌಲ್ಯದಲ್ಲಿ ಸಾಮಾನವಾಗಿರುತ್ತವೆ ಆದರೆ ಟೈಪ್‌ನಲ್ಲಿ ವಿಭಿನ್ನವಾಗಿರುತ್ತವೆ.

ಉದಾಹರಣೆ 28

typeof undefined // undefined

typeof null // object

null === undefined // false

null == undefined // true

ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾ (Primitive Data)

ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾ ಯಾವುದೇ ಹೆಚ್ಚಿನ ಗುಣಲಕ್ಷಣಗಳು (properties)ಮತ್ತು ಮೆಥಡ್‌ಗಳನ್ನು (methods) ಹೊಂದಿಲ್ಲದ ಏಕೈಕ ಸರಳವಾದ ಡಾಟಾ ಮೌಲ್ಯವಾಗಿದೆ. typeof ಆಪರೇಟರ್ ಕೆಳಗಿನ ಒಂದನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

- string
- number
- boolean
- undefined

ಉದಾಹರಣೆ 29

typeof "Rajesh" // "string" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof 1.44 // "number" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof true // "boolean" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof false // "boolean" ಹಿಂತಿರುಗಿಸುತ್ತದೆ

typeof a // a ಯಾವುದೇ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರದಿದ್ದರೆ "undefined" ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ

ಕಾಂಪ್ಲೆಕ್ಸ್ ಡಾಟಾ (Complex Data)

typeof ಆಪರೇಟರ್ ಎರಡರಲ್ಲಿ ಒಂದು ಕಾಂಪ್ಲೆಕ್ಸ್ ಟೈಪನ್ನು ಹಿಂತಿರುಗಿಸಬಹುದು.

- function
- object

typeof ಆಪರೇಟರ್ ಅರೇ(array) ಮತ್ತು ನಲ್(null) ಎರಡೂ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಿಗೆ ಆಬ್ಜೆಕ್ಟ್ (object) ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಇದು

NOT TO BE REPUBLISHED © NIMI

function ಗೆ ಆಬ್ಜೆಕ್ಟ್ ಎಂದು ಉತ್ತರಿಸುವುದಿಲ್ಲ.

ಉದಾಹರಣೆ 30

typeof {name: 'Karthik', age :27} // "object" ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

typeof [10, 20, 30, 40, 50] // "object" ಹಿಂತಿರುಗಿ ಸುತ್ತದೆ. ("array" ಎಂದಲ್ಲ, ಕೆಳಗಿನ ಸೂಚನೆಯನ್ನು ಗಮನಿಸಿ.)

typeof null // "object" ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

typeof function sampleFunc() {} // "function" ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸೂಚನೆ : typeof ಆಪರೇಟರ್ ಅರೇ (array) ಯನ್ನು "object" ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ. ಏಕೆಂದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ array ಗಳು object ಗಳಾಗಿವೆ.ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Constants ಮತ್ತು ಆಪರೇಟರ್‌ಗಳನ್ನು

ಉಪಯೋಗಿಸುವುದು (Using JavaScript Constants and Operators)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ constant ವಿವರಿಸಿ.**
- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ operator ವಿವರಿಸಿ.**

Constants

Constants ಗಳು ಒಂದು ವಿಶೇಷ ರೀತಿಯ ವೇರಿಯೇಬಲ್ ಆಗಿದ್ದು, ಅವು ಪ್ರೊಗ್ರಾಂನ ಅವಧಿಯಲ್ಲಿ ಎಂದಿಗೂ ಬದಲಾಗದೇ ಇರುವ ಮೌಲ್ಯವನ್ನು ಸಂಗ್ರಹಿಸುತ್ತವೆ.

Constants ನ್ನು ರಚಿಸಲು ಸಿಂಟ್ಯಾಕ್ಸ್ ಹೀಗಿದೆ :

const CONSTANT_NAME:DataType = value;

ಮೇಲಿನ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನಲ್ಲಿ "const" ಒಂದು ವಿಶೇಷವಾದ ಕೀವರ್ಡ್ ಆಗಿದ್ದು, ಇದನ್ನು Constants ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಕಾಯ್ದಿರಿಸಲಾಗಿದೆ. ನೀವು ನೋಡಿದಂತೆ ಇದರ ಸಿಂಟ್ಯಾಕ್ಸ್ ವೇರಿಯೇಬಲ್ ಡಿಕ್ಲೇರ್ ಮಾಡುವಂತೆಯೆ ಕಾಣುತ್ತದೆ ಆದರೆ ಇಲ್ಲಿ var ಕೀವರ್ಡ್ ಬದಲಾಗಿ const ಕೀವರ್ಡ್ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಹೆಚ್ಚಿನ ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ವೇರಿಯೇಬಲ್‌ಗಳಿಂದ ಪ್ರತ್ಯೇಕಿಸಲು constant ಗಳ ಹೆಸರಿನ ಎಲ್ಲ ಅಕ್ಷರಗಳನ್ನು ಕ್ಯಾಪ್ಸ್‌ಲಾಕ್ (capslock) ನಲ್ಲಿ ಉಪಯೋಗಿಸುತ್ತಾರೆ.

ಉದಾಹರಣೆ 1

const FRIEND = 'Shanthi';

const BROTHER_AGE = 46;

ಸೂಚನೆ : const ಕೀವರ್ಡ್ ಸ್ವಲ್ಪ ತಪ್ಪುದಾರಿಗೆಳೆಯುಂತಿದೆ. ಇದು ಸ್ಥಿರ ಮೌಲ್ಯವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದಿಲ್ಲ. ಇದು ಒಂದು ಸ್ಥಿರ ಉಲ್ಲೇಖವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ಈ ಕಾರಣದಿಂದ, ನಾವು constant ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯಗಳನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ, ಆದರೆ, ನಾವು constant ಆಬ್ಜೆಕ್ಟ್‌ಗಳ property ಗಳನ್ನು ಬದಲಾಯಿಸಬಹುದು.

ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯಗಳು (Primitive Values)

ನಾವು ಒಂದು constant ಗೆ ಒಂದು ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯವನ್ನು ನಿಯೋಜಿಸಿದರೆ, ನಾವು ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಉದಾಹರಣೆ 2

const PI = 3.141592653589793;

PI = 3.14; // ಇದು ದೋಷ (error) ತೋರಿಸುತ್ತದೆ.

PI = PI + 10; // ಇದು ಕೂಡ ದೋಷ (error) ತೋರಿಸುತ್ತದೆ.

Constant ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಬದಲಾಗಬಹುದು (Constant Objects can Change)

Constant ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಗುಣಲಕ್ಷಣಗಳನ್ನು (properties) ಬದಲಾಯಿಸಿ.

ಉದಾಹರಣೆ 3

// ನೀವು const object ನ್ನು ರಚಿಸಬಹುದು.

const bike = {type."Yamaha", model."R15", color. "blue"};

// ನೀವು property ಯನ್ನು ಬದಲಾಯಿಸಬಹುದು :

bike.color = "grey" ;

// ನೀವು property ಯನ್ನು ಸೇರಿಸಬಹುದು :

bike.owner = "Sree";

© NIMI NOT TO BE REPUBLISHED

ಆದರೆ ನೀವು constant ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಮತ್ತೆ ಅಸೈನ್ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಉದಾಹರಣೆ 4

const bike = {type: "Yamaha", model: "R15", color: "blue"};

bike = {type: "Tvs", model: "Star city", color: "black"};

// ERROR

Constant ಅರೇ ಗಳು ಬದಲಾಗಬಹುದು(Constant Arrays can Change)

ನೀವು Constant ಅರೇಗಳ ಅಂಶಗಳನ್ನು (elements) ಬದಲಾಯಿಸಬಹುದು.

ಉದಾಹರಣೆ 5

// ನೀವು constant array ಯನ್ನು ರಚಿಸಬಹುದು :

constant bikes = ["TVS", "Yamaha", "Royal Enfield"];

// You can change an element;

bikes[0] = "suzuki";

bikes.push ("Bajaj"); //you can add an element

But you can NOT reassign a constant array:

ಉದಾಹರಣೆ 6

const bikes = {"TVS", "Yamaha", "Royal Enfield"];

bikes = ["TVS", "Yamaha", "Bajaj"]; //ERROR

ಆಪರೇಟರ್‌ಗಳು (Operators)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಎಂಟು ವಿಧದ ಆಪರೇಟರ್‌ಗಳಿವೆ.

- Additive ಆಪರೇಟರ್
- Multiplicative ಆಪರೇಟರ್
- Bitwise ಆಪರೇಟರ್
- Equality ಆಪರೇಟರ್
- Relational ಆಪರೇಟರ್
- Unary ಆಪರೇಟರ್
- Ternary ಆಪರೇಟರ್
- Assignment ಆಪರೇಟರ್

Additive ಆಪರೇಟರ್ : Additive ಆಪರೇಟರ್ ಎನ್ನುವ ಶಬ್ದವು ಸಂಕಲನ (+) ಮತ್ತು ವ್ಯವಕಲನ (-) ಎರಡನ್ನೂ ಒಳಗೊಂಡಿರುತ್ತದೆ. ವ್ಯವಕಲನವೂ ಸಹ ಋಣಾತ್ಮಕ ಸಂಖ್ಯೆಯೊಂದಿಗೆ ಸಂಕಲನವಾಗಿದೆ.

ಉದಾಹರಣೆ 7

32+67; // 99 ಉತ್ತರ ನೀಡುತ್ತದೆ.

d+e; // d ಯನ್ನು e ಜೊತೆ ಸೇರಿಸುತ್ತದೆ.

3-7; // -4 ಉತ್ತರ ನೀಡುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಕಲನವು ಕೆಲವೊಮ್ಮೆ ಅನಿರೀಕ್ಷಿತ ಉತ್ತರಗಳನ್ನು ನೀಡುತ್ತದೆ.

ಉದಾಹರಣೆ 8

var num1= "Runs";

var num2 = 784;

var res = num1+num2; // Runs784 ಉತ್ತರ ನಿಡುತ್ತದೆ ಏಕೆಂದರೆ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎರಡು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಜೋಡಿಸುತ್ತದೆ.

Multiplicative ಆಪರೇಟರ್ : Additive ಆಪರೇಟರ್‌ನಂತೆಯೆ, multiplication(*), division(/) and modulo(%) ಇವು Multiplicative ಆಪರೇಟರ್ ಆಗಿವೆ. modulo ಆಪರೇಟರ್ ಭಾಗಾಕಾರದ ಶೇಷವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Example 9

javascript:alert(4%3); // 1 ಉತ್ತರಿಸುತ್ತದೆ.

ಸಂಕಲನ ಮತ್ತು ಗುಣಾಕಾರದ ಆಪರೇಟರ್‌ಗಳನ್ನು ಒಟ್ಟಾಗಿ ಅಂಕಗಣಿತ (arithmetic) ಆಪರೇಟರ್ ಎಂದು ಕರೆಯಬಹುದು

Bitwise ಆಪರೇಟರ್ :

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ 32 ಬಿಟ್‌ಗಳ ಬಿಟ್‌ವೈಸ್ ಆಪರಂಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಗಳನ್ನು 64 ಬಿಟ್‌ಗಳ ಫ್ಲೋಟಿಂಗ್ ಪಾಯಿಂಟ್ (floating point) ಸಂಖ್ಯೆಗಳನ್ನಾಗಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ ಆದರೆ ಎಲ್ಲ ಬಿಟ್‌ವೈಸ್ ಕಾರ್ಯಾಚರಣೆಗಳು 32 ಬಿಟ್ ಬೈನರಿ ಸಂಖ್ಯೆಗಳ ಮೇಲೆ ನಡೆಯುತ್ತವೆ. ಬಿಟ್‌ವೈಸ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುವ ಮೊದಲು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಗಳನ್ನು 32 ಬಿಟ್‌ನ ಸಂಕೇತಿಸಿದ ಪೂರ್ಣಾಂಕಗಳಿಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ. ಬಿಟ್‌ವೈಸ್ ಕಾರ್ಯಾಚರಣೆಯನ್ನು ನಿರ್ವಹಿಸಿದ ನಂತರ, ಫಲಿತಾಂಶವನ್ನು ಮತ್ತೆ 64 ಬೀಟ್‌ಗಳ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಗಳಿಗೆ ಪರಿವರ್ತಿಸಲಾಗುತ್ತದೆ.

ಆಪರೇಟರ್	ಹೆಸರು	ವಿವರಣೆ
&	AND	ಎರಡೂ ಬಿಟ್‌ಗಳು 1 ಆಗಿದ್ದರೆ, ಪ್ರತಿ ಬಿಟ್‌ನ್ನು 1 ಗೆ ಸೆಟ್ ಮಾಡುತ್ತದೆ
|	OR	ಎರಡು ಬಿಟ್‌ಗಳಲ್ಲಿ ಯಾವುದಾದರೂ ಒಂದು ಬಿಟ್ 1 ಆಗಿದ್ದರೆ, ಪ್ರತಿ ಬಿಟ್‌ನ್ನು 1 ಗೆ ಸೆಟ್ ಮಾಡುತ್ತದೆ.
^	XOR	ಎರಡು ಬಿಟ್‌ಗಳಲ್ಲಿ ಒಂದು ಬಿಟ್ ಮಾತ್ರ 1 ಆಗಿದ್ದರೆ, ಪ್ರತಿ ಬಿಟ್‌ನ್ನು 1 ಗೆ ಸೆಟ್ ಮಾಡುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

~	NOT	ಎಲ್ಲ ಬಿಟ್‌ಗಳನ್ನು ವಿಲೋಮಗೊಳಿಸುತ್ತದೆ.
<<	Zero fill left shift	ಸೊನ್ನೆಗಳು ಬಲಗಡೆಯಿಂದ ಎಡಗಡೆಯ ಕೊನೆಯ ಬಿಟ್‌ವರೆಗೂ ಸರಿಸಲ್ಪಡುತ್ತವೆ.
>>	Signed right shift	ಎಡಕೊನೆಯ ಬಿಟ್ಟನ್ನು ನಕಲಿಸಿ ಬಲಭಾಗದ ಕೊನೆಯವರೆಗೂ ಸರಿಯುತ್ತವೆ.
>>>	Zero fill right shift	ಸೊನ್ನೆಗಳು ಎಡಗಡೆಯಿಂದ ಬಲಗಡೆಯ ಕೊನೆಯ ಬಿಟ್‌ವರೆಗೂ ಸರಿಸಲ್ಪಡುತ್ತವೆ.

ಉದಾಹರಣೆ 10

Operation	Result	Same as	Result
5 & 1	1	0101 & 0001	0001
5 | 1	5	0101 | 0001	0101
~ 5	10	~0101	1010
5 << 1	10	0101 << 1	1010
5 ^ 1	4	0101 ^ 0001	0100
5 >> 1	2	0101 >> 1	0010
5 >>> 1	2	0101 >>> 1	0010

ಸೂಚನೆ : ಮೇಲಿನ ಉದಾಹರಣೆಗಳು 4 ಬಿಟ್‌ಗಳ ಸಂಕೇತವಿಲ್ಲದ ಬೈನರಿ ಸಂಖ್ಯೆಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಈ ಕಾರಣದಿಂದಾಗಿ ~5, 10 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ 32 ಬಿಟ್‌ಗಳ ಸಂಕೇತವಿರುವ ಪೂರ್ಣಾಂಕಗಳನ್ನು ಉಪಯೋಗಿಸುವುದರಿಂದ, ಇದು 10 ನ್ನು ಹಿಂತಿರುಗಿಸುವುದಿಲ್ಲ. ಇದು -6

ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

00000000000000000000000000000101 (5)

11111111111111111111111111111010 (~5 = -6)

ಸಂಕೇತವಿರುವ ಪೂರ್ಣಾಂಕವು ಎಡಭಾಗದ ಬಿಟ್‌ನ್ನು ಋಣ ಚಿನ್ಹೆಯನ್ನಾಗಿ (minus sign) **ಉಪಯೋಗಿಸುತ್ತದೆ.**

ಸಮಾನತೆಯ ಆಪರೇಟರ್ (Equality operator)

ಎರಡು ಎಕ್ಸಪ್ರೆಷನ್‌ಗಳು ಒಂದೇ ಆಗಿವೆಯೆ ಎಂದು ಪರೀಕ್ಷಿಸಲು ಸಮಾನತೆಯ ಆಪರೇಟರ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ "42" ಮತ್ತು 42 ಎರಡೂ == ನೊಂದಿಗೆ ಸಮಾನವಾಗಿರುತ್ತವೆ, ಆದರೆ === ನೊಂದಿಗೆ ಸಮಾನವಾಗಿರುವುದಿಲ್ಲ. ಏಕೆಂದರೆ ಇದು ಮೌಲ್ಯವನ್ನು ಪರಿಶೀಲಿಸುವುದಲ್ಲದೇ ಡಾಟಾಟೈಪನ್ನೂ ಸಹ ಪರಿಶೀಲಿಸುತ್ತದೆ.

ರಿಲೇಷನಲ್ ಆಪರೇಟರ್ (Relational Operator)

ರಿಲೇಷನಲ್ ಆಪರೇಟರ್, ಒಂದು ಮೌಲ್ಯವು ಇನ್ನೊಂದಕ್ಕಿಂತ ದೊಡ್ಡದಾಗಿದೆಯೆ ಅಥವಾ ಚಿಕ್ಕದಾಗಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸುತ್ತದೆ.

```
if(5<2) {
// ಏನಾದರೂ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು
}
```

ಆಪರೇಟರ್	ಕಾರ್ಯ
>	Greater than (ದೊಡ್ಡದು)
<	Less than (ಚಿಕ್ಕದು)
>=	Greater than or equal to (ದೊಡ್ಡದು ಅಥವಾ ಸಮ)
<=	Less than or equal to (ಚಿಕ್ಕದು ಅಥವಾ ಸಮ)
in	Tests whether a value is found in an expression (ಒಂದು ಮೌಲ್ಯ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನಲ್ಲಿ ಇದೆಯೆ ಎಂದು ಪರೀಕ್ಷಿಸುತ್ತದೆ)
instanceof	Tests whether an expression is an instance of an object (ಒಂದು ಎಕ್ಸ್‌ಪ್ರೆಷನ್ ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ ನಿದರ್ಶನವಾಗಿದೆಯೆ ಎಂದು ಪರೀಕ್ಷಿಸುತ್ತದೆ)

less than ಆಪರೇಟರ್ ಮೊದಲನೇ ಮೌಲ್ಯವು ಎರಡನೇ ಮೌಲ್ಯಕ್ಕಿಂತ ಚಿಕ್ಕದಾಗಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸುತ್ತದೆ ಮತ್ತು ಈ ಪರೀಕ್ಷೆ ಮಾನ್ಯವಾಗಿದ್ದರೆ, ಸರಿ ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ. ಮೇಲಿನ ಉದಹರಣೆಯಲ್ಲಿ, 5, 2 ಕ್ಕಿಂತ ಚಿಕ್ಕದಾಗಿಲ್ಲ, ಅದ್ದರಿಂದ ಇದು ಸರಿ (true) ಆಗುವುದಿಲ್ಲ ಮತ್ತು if ಬ್ಲಾಕ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದಿಲ್ಲ.

ಇನ್ನುಳಿದ ಮೂರೂ ಆಪರೇಟರ್‌ಗಳು ಇದೇ ರೀತಿಯಲ್ಲಿ greater then (>),Greater than or equal to(>=), Less than or equal to (<=) ಗಳನ್ನು ಪರಿಶೀಲಿಸುತ್ತವೆ.

in ಆಪರೇಟರ್, ನೀಡಲಾದ ಇಂಡೆಕ್ಸ್ ಆಬ್ಜೆಕ್ಟ್‌ನಲ್ಲಿ ಇದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ :

```
var MyObj= {star: "Algol", constellation: "Perseus"};
if ("star" in MyObj) {
```

© NIMI NOT TO BE REPUBLISHED

// ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು

}

star ಇಂಡೆಕ್ಸ್ ಆಗಿರುವುದರಿಂದ ಕೋಡ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ, ಆದರೆ in ಆಪರೇಟರ್ ಸಂಖ್ಯೆಯ (numeric) ಡಾಟಾಟೈಪ್‌ನಲ್ಲಿ ಕೆಲಸ ಮಾಡುವುದಿಲ್ಲ ಏಕೆಂದರೆ ಅದು ಸಂಖ್ಯೆಗಳ ಮೇಲೆ ಮಾತ್ರ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

Instanceof ಆಪರೇಟರ್, ಆಬ್ಜೆಕ್ಟ್‌ನ ನಿದರ್ಶನ (object instance) ಅಥವಾ ಆಬ್ಜೆಕ್ಟ್ ವೇರಿಯೇಬಲ್ ಒಂದು ನಿರ್ದಿಷ್ಟ ಆಬ್ಜೆಕ್ಟ್‌ನ instance ಆಗಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 11

```
var mydate=new Date();
if(mydate instanceof Date) {
// ಏನಾದರೂ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದು
}
```

ಇಲ್ಲಿ mydate ಎನ್ನುವುದು ಅಂತರ್‌ನಿರ್ಮಿತ (built-in) Date object ನ instance ಆಗಿದೆ. ಅದ್ದರಿಂದ if ಬ್ಲಾಕ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಯುನರಿ ಆಪರೇಟರ್ (Unary Operator)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ delete, void, typeof, ++, --, + , -, ~ , ! ಯುನರಿ ಆಪರೇಟರ್‌ಗಳಾಗಿವೆ.

ಉದಾಹರಣೆ 12

```
a = -10;
p=++a;
q=a++;
s=+p;
```

pre and post increment and decrement operator.

p=++a; ಇದು

a=a+1;

p=a; ಗೆ ಸಮನಾಗಿದೆ

and q=a++; ಇದು

q=a;

a=a+1; ಗೆ ಸಮನಾಗಿದೆ

ಡಿಲೀಟ್ ಆಪರೇಟರ್ (The delete Operator)

ಡಿಲೀಟ್ ಆಪರೇಟರ್, ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಡಿಲಿಟ್ (ಅಳಿಸಲು) ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

Example 13

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
delete person.age;
```

ಡಿಲೀಟ್ ಆಪರೇಟರ್‌ನ್ನು ಆಬ್ಜೆಕ್ಟ್ ಗುಣಲಕ್ಷಣಗಳ(properties) ಮೇಲೆ ಉಪಯೋಗಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಇದು ವೇರಿಯೇಬಲ್‌ಗಳು ಅಥವಾ ಫಂಕ್ಷನ್‌ಗಳ ಮೇಲೆ ಯಾವುದೇ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ.

ಡಿಲಿಟ್ ಆಪರೇಟರ್ ಪೂರ್ವನಿರ್ಧರಿತ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಬ್ಜೆಕ್ಟ್ ಗುಣಲಕ್ಷಣಗಳ ಮೇಲೆ ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ. ಇದು ನಿಮ್ಮ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಕ್ರ್ಯಾಶ್ ಮಾಡಬಹುದು.

ಯುನರಿ + ಆಪರೇಟರ್ (The Unary + Operator)

ಒಂದು ವೇರಿಯೇಬಲ್‌ನ್ನು ಸಂಖ್ಯೆಗೆ ಬದಲಾಯಿಸಲು ಯುನರಿ + ಆಪರೇಟರ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 14

```
var y = "5";   // y ಒಂದು ಸ್ಟ್ರಿಂಗ್ ಆಗಿದೆ.
var x = + y;   // x ಒಂದು ಸಂಖ್ಯೆ ಆಗಿದೆ.
```

ಒಂದು ವೇರಿಯೇಬಲ್‌ನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಾಗದಿದ್ದಲ್ಲಿ, ಅದು ಇನ್ನೂ ಸಂಖ್ಯೆ ಆಗುತ್ತದೆ., ಆದರೆ NaN (Not a number) ಎನ್ನುವ ಮೌಲ್ಯದೊಂದಿಗೆ.:

ಉದಾಹರಣೆ 15

```
var y = "John";   // y ಒಂದು ಸ್ಟ್ರಿಂಗ್ ಆಗಿದೆ.
var x = + y;      // x ಒಂದು ಸಂಖ್ಯೆ (NaN) ಆಗಿದೆ.
```

ಟರ್ನರಿ ಅಥವಾ ಕಂಡಿಷನಲ್ ಆಪರೇಟರ್

(Ternary or Conditional Operator): ಇದನ್ನು if else ನ ಸಂಕ್ಷೀಪ್ತ ರೂಪದ ಹಾಗೆ ಬಳಸಬಹುದು.

ಉದಾಹರಣೆ 16

a = (b>5 ? 4:7); ಎಂದರೆ

```
if(b>5)
a=4;
else
a=7;
```

ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್ (Assignment Operator): ಒಂದು ವೇರಿಯೇಬಲ್‌ಗೆ ಮೌಲ್ಯಗಳನ್ನು ನಿಯೋಜಿಸಲು ಅಸೈನ್‌ಮೆಂಟ್

ಆಪರೇಟರ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. = ನ ಹೊರತಾಗಿ ಈ ಕೆಳಗಿನಂತೆ ಸಂಯುಕ್ತ ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್‌ಗಳು ಇವೆ :-

*=	/=	%=
+=	-=	<<=
>>=	>>>=	&=
^=	|=	

a = q; ಎಂದರೆ q ನ ಮೌಲ್ಯವನ್ನು a ವೇರಿಯೇಬಲ್‌ಗೆ ಅದರ ಹಿಂದಿನ ಮೌಲ್ಯವನ್ನು ಅಳಿಸಿ ನಿಗದಿಪಡಿಸಲಾಗುತ್ತದೆ.

ಈಗ a* = 3; ಇದು a = a*3; ಗೆ ಸಮನಾಗಿದೆ.ಮತ್ತು ಅದರಂತೆ ಎಲ್ಲ ಸಂಯುಕ್ತ ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್‌ಗಳು ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು , ಲೂಪ್ ಮತ್ತು ಪಾಪ್‌ಅಪ್ ಬಾಕ್ಸ್‌ಗಳು (Control statements, Loops and Popup boxes in JavaScript)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- ವಿವಿಧ ಲೂಪ್‌ಗಳ ಬಗ್ಗೆ ಚರ್ಚಿಸಿ.
- ಪಾಪ್‌ಅಪ್ ಬಾಕ್ಸ್‌ಗಳ ಉಪಯೋಗಗಳನ್ನು ವಿವರಿಸಿ.

ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು (Control Statements) : ನಾವು ಒಂದು ನಿರ್ದಿಷ್ಟ ಪ್ರೊಗ್ರಾಂಗೆ ಕೋಡ್ ಬರೆಯುವಾಗ, ವಿಬಿನ್ನವಾದ ಕ್ರಿಯೆಗಳನ್ನು ನಿರ್ವಹಿಸಲು ವಿವಿಧ ನಿರ್ಧಾರಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತೇವೆ. ಇವುಗಳನ್ನು ನೀವು ಷರತ್ತುಬದ್ಧ /ನಿಯಂತ್ರಣ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳ ಮೂಲಕ ಮಾಡಬಹುದು. (conditional/control statements.)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಈ ಕೆಳಗಿನ ಷರತ್ತುಬದ್ಧ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು (conditional statements)ಇರುತ್ತವೆ.

ಒಂದು ನಿರ್ದಿಷ್ಟ ಷರತ್ತು true ಇದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು if ಉಪಯೋಗಿಸಿ.

ಅದೇ ಷರತ್ತು false ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು else ಉಪಯೋಗಿಸಿ.

ಮೊದಲನೆ ಷರತ್ತು false ಆಗಿದ್ದರೆ, ಪರೀಕ್ಷಿಸಲು ಹೊಸ ಷರತ್ತನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು else if ಉಪಯೋಗಿಸಿ.

ಅನೇಕ ಪರ್ಯಾಯ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಪರೀಕ್ಷಿಸಿ ಕಾರ್ಯಗತಗೊಳಿಸಲು switch ಅನ್ನು ಉಪಯೋಗಿಸಿ.

if ಸ್ಟೇಟಮೆಂಟ್ (The if Statement)

ಒಂದು ನಿರ್ದಿಷ್ಟ ಷರತ್ತು true ಇದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು if ಉಪಯೋಗಿಸಿ.

Syntax

```
If (condition) {
    ಷರತ್ತು true ಇದ್ದರೆ, ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕು
}
```

ಉದಾಹರಣೆ 1

ಸಮಯ 18:00 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇದ್ದರೆ "Good day" ಎನ್ನುವ ಶುಭಾಶಯ ಪ್ರದರ್ಶಿಸಬೇಕು.

```
if (time < 18) {
    greeting = "Good day";
}
```

ಶುಭಾಶಯದ ಫಲಿತಾಂಶ ಹೀಗಿರುತ್ತದೆ :

Good day

else ಸ್ಟೇಟಮೆಂಟ್ (The else Statement)

ಒಂದು ಷರತ್ತು false ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು else ಉಪಯೋಗಿಸಿ.

```
if (condition) {
    ಷರತ್ತು true ಇದ್ದರೆ, ಕೋಡ್‌ನ ಈ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕು
} else {
    ಷರತು false ಇದ್ದರೆ, ಕೋಡ್‌ನ ಈ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕು
}
```

ಉದಾಹರಣೆ 2

ಸಮಯ 18:00 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇದ್ದರೆ "Good day" ಎನ್ನುವ ಶುಭಾಶಯ ಪ್ರದರ್ಶಿಸಬೇಕು.

ಇಲ್ಲದಿದ್ದರೆ "Good evening" ಎಂದು ಪ್ರದರ್ಶಿಸಬೇಕು :

```
if (time < 18) {
    greeting = "Good day";
```

© NIMI NOT TO BE REPUBLISHED

```
} else {
    greeting = "Good evening";
}
```

ಫಲಿತಾಂಶ ಹೀಗಿರುತ್ತದೆ:

Good day

else if ಸ್ಟೇಟಮೆಂಟ್ (The else if Statement)

ಮೊದಲನೆ ಷರತ್ತು false ಆಗಿದ್ದರೆ, ಹೊಸ ಷರತ್ತನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು else if ಸ್ಟೇಟಮೆಂಟ್ ಉಪಯೋಗಿಸಿ.

Syntax

```
if (condition1) {
    condition1 true ಆಗಿದ್ದರೆ, ಕೋಡ್‌ನ ಈ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸಬೇಕು
} else if (condition2) {
    condition1 false ಆಗಿದ್ದರೆ ಮತ್ತು condition 2 true ಆಗಿದ್ದರೆ, ಕೋಡ್‌ನ ಈ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸಬೇಕು
} else {
    condition1 false ಆಗಿದ್ದರೆ ಮತ್ತು condition2 is false ಆಗಿದ್ದರೆ, ಕೋಡ್‌ನ ಈ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸಬೇಕು
}
```

ಉದಾಹರಣೆ 3

ಸಮಯ 10:00 ಗಂಟೆಗಿಂತ ಕಡಿಮೆ ಇದ್ದರೆ, "Good morning" ಶುಭಾಶಯ ರಚಿಸಿ , ಇಲ್ಲದಿದ್ದರೆ, ಮತ್ತು ಸಮಯ 18:00 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇದ್ದರೆ "Good day" ಶುಭಾಶಯ ರಚಿಸಿ, ಇಲ್ಲದಿದ್ದರೆ "Good evening" ಶುಭಾಶಯ ರಚಿಸಿ.

```
if (time < 10) {
    greeting = "Good morning" ;
} else if (time < 18) {
    greeting = "Good day" ;
} else {
    greeting = "Good evening";
}
```

ಫಲಿತಾಂಶ :

Good day

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ switch ಸ್ಟೇಟಮೆಂಟ್ (The JavaScript Switch Statement)

switch ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಅನೇಕ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿ ಅದನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
switch(expression) {
    case n1:
        code block
        break;
    case n2:
        code block
        break;
default:
    default code block
}
```

ಇದು ಯಾವ ರೀತಿ ಕೆಲಸ ಮಾಡುತ್ತದೆಂದರೆ :

- switch ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ್ನು ಒಮ್ಮೆ ಮಾತ್ರ ಮೌಲ್ಯಮಾಪನ ಮಾಡಲಾಗುತ್ತದೆ.
- ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ ಮೌಲ್ಯವನ್ನು ಪ್ರತಿಯೊಂದು case ನ ಮೌಲ್ಯದೊಂದಿಗೆ ಹೋಲಿಸಲಾಗುತ್ತದೆ.
- ಹೊಂದಾಣಿಕೆ ಕಂಡು ಬಂದಲ್ಲಿ, ಅದಕ್ಕೆ ಸಂಬಂಧಪಟ್ಟ ಬ್ಲಾಕ್‌ನ ಕೋಡ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 4

ವಾರದ ದಿನದ ಹೆಸರನ್ನು ಪ್ರದರ್ಶಿಸಲು ವಾರದ ದಿನದ ಸಂಖ್ಯೆಯನ್ನು ಬಳಸಿ :

(Sunday=0, Monday=1, Tuesday=2, ...)

```
switch (new Date().getDay()) {
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
```

© NIMI NOT TO BE REPUBLISHED

```
        break;

    case 2:

        day = "Tuesday";

        break;

    case 3:

        day = "Wednesday";

        break;

    case 4:

        day = "Thursday";

        break;

    case 5:

        day = "Friday";

        break;

    case 6:

        day = "Saturday";

        break;

}
```

ಫಲಿತಾಂಶ ಹೀಗಿರುತ್ತದೆ :

Tuesday

break ಕೀವರ್ಡ್ (The break Keyword)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೋಡ್ ಇಂಟರ್‌ಪ್ರಿಟರ್, ಬ್ರೆಕ್ (break) ಕೀವರ್ಡ್ ತಲುಪಿದಾಗ , ಅದು switch ಬ್ಲಾಕ್‌ನಿಂದ ಹೊರಗಡೆ ಬರುತ್ತದೆ.

ಇದು ಬ್ಲಾಕ್‌ನಲ್ಲಿರುವ ಇನ್ನುಳಿದ ಕೋಡ್ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದನ್ನು ಮತ್ತು ಅಥವಾ ಪರೀಕ್ಷಿಸುವುದನ್ನು ನಿಲ್ಲಿಸುತ್ತದೆ.

ಡೀಫಾಲ್ಟ್ ಕೀವರ್ಡ್ (The default Keyword)

ಒಂದು ವೇಳೆ ಯಾವುದೇ case ಗೆ ಹೊಂದಾಣಿಕೆ ಆಗದಿದ್ದರೆ, ಡೀಫಾಲ್ಟ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 5

ಇವತ್ತು ಶನಿವಾರ ಅಥವಾ ಭಾನುವಾರ ಇಲ್ಲದಿದ್ದರೆ, ಡೀಫಾಲ್ಟ್ ಸಂದೇಶವನ್ನು ಬರೆಯಿರಿ.

```
switch (new Date().getDay()) {

    case 6:

        text = "Today is Saturday";

        break;

    case 0:

        text = "Today is Sunday";

        break;

    default:

        text = "Looking forward to the Weekend";

}
```

The **result** of text will be:

Looking forward to the Weekend

ಕಾಮನ್ ಕೋಡ್ ಮತ್ತು ಫಾಲ್ ಥ್ರೂ (Common Code and Fall-Through)

ಕೆಲವೊಮ್ಮೆ switch ಬ್ಲಾಕ್‌ನಲ್ಲಿ, ವಿಭಿನ್ನ case ಗಳು ಒಂದೇ ಕೋಡನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸುತ್ತೀರಿ ಅಥವಾ ಸಾಮಾನ್ಯ ಡೀಫಾಲ್ಟ್‌ಗೆ ಬರುತ್ತೀರಿ.

ಮುಂದಿನ ಉದಾಹರಣೆಯನ್ನು ಗಮನಿಸಿ, ಕೆಲವು case ಗಳು ಒಂದೇ ಕೋಡ್ ಬ್ಲಾಕನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತವೆ ಮತ್ತು switch ಬ್ಲಾಕ್‌ನಲ್ಲಿ ಡೀಫಾಲ್ಟ್ case ಕೊನೆಯ ಕೇಸ್ ಆಗಿರಬೇಕಾಗಿಲ್ಲ :

ಉದಾಹರಣೆ 6

```
switch (new Date().getDay()) {

    case 1:

    case 2:

    case 3:

    default:

        text = "Weekend is coming";

        break;

    case 4:

    case 5:

        text = "Weekend is soon";

        break;

    case 0:

    case 6:
```

© NIMI NOT TO BE REPUBLISHED

```
text = "Now in Weekend";

}
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಲೂಪ್‌ಗಳು (JavaScript Loops)

ಪ್ರತಿ ಬಾರಿ ವಿಭಿನ್ನ ಮೌಲ್ಯದೊಂದಿಗೆ ನೀವು ಒಂದು ಕೋಡ್‌ಅನ್ನು ಮತ್ತೆ ಮತ್ತೆ ಚಲಾಯಿಸಲು ಬಯಸಿದರೆ , ಲೂಪ್‌ಗಳು ಸೂಕ್ತವಾಗಿವೆ. ಅರೇ ಗಳೊಂದಿಗೆ ಕೆಲಸ ಮಾಡುವಾಗ ಸಾಮಾನ್ಯವಾಗಿ ಈ ರೀತಿಯಾಗುತ್ತದೆ:

ಕೆಳಗಿನ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಬರೆಯುವ ಬದಲು:

```
text += train[0] + "<text += train [1] + "<br>";

text += train [2] + "<br>";

text += train [3] + "<br>";

text += train [4] + "<br>";

text += train [5] + "<br>";
```

ನೀವು ಹೀಗೆ ಬರೆಯಬಹುದು :

```
for (i = 0; i < train.length; i++) {

    text += train [i] + "<br>";

}
```

ವಿವಿಧ ರೀತಿಯ ಲೂಪ್‌ಗಳು (Different Kinds of Loops)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವಿವಿಧ ರೀತಿಯ ಲೂಪ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ :

- for - ಹಲವಾರು ಬಾರಿ ಕೋಡ್ ಬ್ಲಾಕನ್ನು ಲೂಪ್ ಮಾಡುತ್ತದೆ.
- for/in - ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ ಗುಣಲಕ್ಷಣಗಳನ್ನು (properties) ಲೂಪ್ ಮಾಡುತ್ತದೆ.
- while - ಒಂದು ನಿರ್ದಿಷ್ಟ ಷರತ್ತು true ಇರುವವರೆಗೆ ಕೋಡ್‌ನ ಬ್ಲಾಕನ್ನು ಲೂಪ್ ಮಾಡುತ್ತದೆ.
- do/while - ಇದು ಸಹ ಒಂದು ನಿರ್ದಿಷ್ಟ ಷರತ್ತು true ಇರುವವರೆಗೆ ಕೋಡ್‌ನ ಬ್ಲಾಕನ್ನು ಲೂಪ್ ಮಾಡುತ್ತದೆ.

For ಲೂಪ್ (The For Loop)

ನೀವು ಒಂದು ಲೂಪನ್ನು ರಚಿಸಲು ಬಯಸಿದರೆ, ಸಾಮಾನ್ಯವಾಗಿ For ಲೂಪ್ ನೀವು ಬಳಸುವ ಸಾಧನವಾಗಿದೆ.

For ಲೂಪ್ ಕೆಳಗಿನ ಸಿಂಟ್ಯಾಕ್ಸ್ ಹೊಂದಿದೆ :

```
for (statement 1; statement 2; statement 3) {

    code block to be executed

}
```

statement 1 ಲೂಪ್ (code block) ಪ್ರಾರಂಭವಾಗುವುದಕ್ಕೂ ಮೊದಲು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಇದನ್ನು Initialisation Part ಎಂದು ಕರೆಯುತ್ತಾರೆ.

statement 2 ಲೂಪ್ (code block) ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಅವಶ್ಯಕವಿರುವ ಷರತ್ತನ್ನು(condition) ವಿವರಿಸುತ್ತದೆ. ಇದನ್ನು condition part ಎಂದು ಕರೆಯುತ್ತಾರೆ.

Statement 3 ಪ್ರತಿ ಬಾರಿ ಲೂಪ್ ಕಾರ್ಯನಿರ್ವಹಿಸಿದಾಗ ಇದು ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಇದನ್ನು increment/decrement part ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಉದಾಹರಣೆ 7

```
for (i = 0; i < 5; i++) {

    text += "The number is " + i + "<br>";

}
```

ಮೇಲಿನ ಉದಾಹರಣೆಯನ್ನು ನೀವು ಹೀಗೆ ಓದಬಹುದು :

Statement 1 ಲೂಪ್ ಪ್ರಾರಂಭವಾಗುವ ಮೊದಲು ವೇರಿಯೇಬಲ್‌ನ್ನು ಸೆಟ್ ಮಾಡುತ್ತದೆ (var i = 0).

Statement 2 ಲೂಪ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸಲು ಷರತ್ತನ್ನು(condition) ವಿವರಿಸುತ್ತದೆ. (i , 5 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇರಬೇಕು).

Statement 3 ಪ್ರತಿ ಬಾರಿ ಲೂಪ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸಿದಾಗ ಮೌಲ್ಯವನ್ನು (i++) ಹೆಚ್ಚಿಸುತ್ತದೆ.

ಆರಂಭದ ಭಾಗ (Initialisation Part)

ಸಾಮಾನ್ಯವಾಗಿ ನೀವು statement 1 ನ್ನು ಲೂಪನಲ್ಲಿರುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ಆರಂಭಿಸಲು (initialise) ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (var i = 0).

statement 1 ಐಚ್ಛಿಕವಾಗಿದೆ.

ನೀವು statement 1 ರಲ್ಲಿ ಅನೇಕ ಮೌಲ್ಯಗಳನ್ನು initialise ಮಾಡಬಹುದು.(ಅಲ್ಪವಿರಾಮದಿಂದ ಪ್ರತ್ಯೇಕಿಸಬಹುದು)

ಉದಾಹರಣೆ 8

```
for (i = 0, len = train.length, text = ""; i < len; i++) {

text += train [i] + "<br>";

}
```

ನೀವು statement 1 ನ್ನು ಬಿಟ್ಟುಬಿಡಬಹುದು (ಲೂಪ್ ಪ್ರಾರಂಭವಾಗುವುದಕ್ಕೂ ಮೊದಲು ಮೌಲ್ಯಗಳನ್ನು ಸೆಟ್ ಮಾಡಿದಾಗ)

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 9

```
var i = 2;

var len = train.length;

var text = "";

for (; i < len; i++) {

    text += train [i] + "<br>";

}
```

Condition ಭಾಗ (Condition Part)

initial ವೇರಿಯೇಬಲ್‌ನ ಷರತ್ತನ್ನು(condition) ಮೌಲ್ಯಮಾಪನ ಮಾಡಲು statement 2 ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

statement 2 ಸಹ ಐಚ್ಛಿಕವಾಗಿದೆ.

statement 2, true ಎಂದು ಉತ್ತರಿಸಿದರೆ, ಲೂಪ್ ಮತ್ತೆ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ, statement 2, false ಎಂದು ಉತ್ತರಿಸಿದರೆ, ಲೂಪ್ ಅಲ್ಲಿಯೆ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ.

ಒಂದು ವೇಳೆ ನೀವು statement 2 ನ್ನು ಬಿಟ್ಟರೆ, ಲೂಪ್‌ನ ಒಳಗಡೆ ನೀವು break ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಒದಗಿಸಬೇಕು. ಇಲ್ಲದಿದ್ದರೆ, ಲೂಪ್ ಎಂದಿಗೂ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ. ಇದು ನಿಮ್ಮ ಬ್ರೌಸರ್‌ನ್ನು ಕ್ರ್ಯಾಶ್ ಮಾಡುತ್ತದೆ. ಈ ಪಾಠದ ನಂತರದ ಅಧ್ಯಾಯದಲ್ಲಿ ನೀವು break ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳುತ್ತೀರಿ.

ಇನ್ಕ್ರಿಮೆಂಟ್ / ಡಿಕ್ರಿಮೆಂಟ್ ಭಾಗ (Increment/Decrement Part)

statement 3, initial ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ.

statement 3 ಐಚ್ಛಿಕವಾಗಿದೆ.

statement 3 ಋಣಾತ್ಮಕ ಇನ್ಕ್ರಿಮೆಂಟ್ (i--) ಅಥವಾ ದೊಡ್ಡದಾದ ಇನ್ಕ್ರಿಮೆಂಟ್ ((i = i + 15) ನಂತಹ ಯಾವುದನ್ನೂ ಮಾಡಬಹುದು.

statement 3 ನ್ನು ಸಹ ಬಿಟ್ಟುಬಿಡಬಹುದು(ನೀವು ಲೂಪ್‌ನಲ್ಲಿ ಮೌಲ್ಯಗಳನ್ನು ಹೆಚ್ಚಿಸಿದಾಗ)

ಉದಾಹರಣೆ 10

```
var i = 0;

len = train.length;

for (; i < len; ) {

    text += train [i] + "<br>";

    i++;

}
```

ಫಾರ್ /ಇನ್ ಲೂಪ್ (For/In Loop)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಾರ್ /ಇನ್ ಲೂಪ್ ಸ್ಟೇಟಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್‌ನ ಗುಣಲಕ್ಷಣಗಳ ಮೂಲಕ ಲೂಪ್ ಮಾಡುತ್ತದೆ.

```
var person = {fname:"Raja", lname:"Sen", age:35};

var text = "";

var x;

for (x in person) {

    text += person[x];

}
```

While ಲೂಪ್ (While Loop)

ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಷರತ್ತು true ಇರುವವರೆಗೆ ಕೋಡ್‌ನ ಬ್ಲಾಕ್ While ಲೂಪ್ ಪದೇ ಪದೇ ಲೂಪ್ ಆಗುತ್ತಿರುತ್ತದೆ.

Syntax

```
while (condition) {

    code block to be executed

}
```

ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ವೇರಿಯೇಬಲ್ i, 10 ಕ್ಕಿಂತಲೂ ಕಡಿಮೆ ಇರುವವರೆಗೆ ಲೂಪ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಪದೇ ಪದೇ ರನ್ ಆಗುತ್ತಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 11

```
while (i < 10) {

    text += "The number is " + i;

    i++;

}
```

ಒಂದು ವೇಳೆ ನೀವು condition ನಲ್ಲಿರುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಚ್ಚಿಸಲು ಮರೆತರೆ , ಲೂಪ್ ಎಂದಿಗೂ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ. ಇದು ನಿಮ್ಮ ಬ್ರೌಸರ್‌ನ್ನು ಕ್ರ್ಯಾಶ್ ಮಾಡುತ್ತದೆ.

Do/While ಲೂಪ್ (The Do/While Loop)

Do/While ಲೂಪ್ while ಲೂಪ್‌ನ ರೂಪಾಂತರವಾಗಿದೆ. ಈ ಲೂಪ್ condition , true ಇದೆಯೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸುವ ಮೊದಲು ಕೋಡ್ ಬ್ಲಾಕ್‌ನ್ನು ಒಮ್ಮೆ ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ, ನಂತರ condition , true ಇರುವವರೆಗೆ ಲೂಪನ್ನು ಪುನರಾವರ್ತಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

Syntax

```
do {
    code block to be executed
}
while (condition);
```

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು do/while ಲೂಪನ್ನು ಬಳಸುತ್ತದೆ. ಒಂದು ವೇಳೆ condition, false ಇದ್ದರೂ ಸಹ ಲೂಪ್‌ನ್ನು ಒಂದು ಬಾರಿಯಾದರೂ ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ ಏಕೆಂದರೆ condition ಪರೀಕ್ಷಿಸುವ ಮೊದಲು ಕೋಡ್ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 12

```
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

condition ನಲ್ಲಿ ಬಳಸಿದ ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಚ್ಚಿಸಲು ಮರೆಯಬೇಡಿರಿ. ಇಲ್ಲದಿದ್ದರೆ ಲೂಪ್ ಎಂದಿಗೂ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ.

For ಮತ್ತು While ಹೋಲಿಸುವುದು (Comparing For and While)

ಫಾರ್ ಲೂಪ್ ಬಗ್ಗೆ ನೀವು ಹಿಂದಿನ ಅಧ್ಯಾಯದಲ್ಲಿ ಓದಿದ್ದರೆ, statement 1 ಮತ್ತು statement 3 ಬಿಟ್ಟಾಗ, For ಲೂಪ್ ಮತ್ತು While ಲೂಪ್ ಎರಡೂ ಒಂದೇ ಆಗಿವೆ ಎಂಬುದನ್ನು ಕಂಡುಕೊಳ್ಳುತ್ತೀರಿ.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿರುವ ಲೂಪ್, train array ದಿಂದ ಕಾರ್ ಹೆಸರುಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು for ಲೂಪ್ ಉಪಯೋಗಿಸುತ್ತದೆ.:

ಉದಾಹರಣೆ 13

```
train = ["Duronto", "Satabdi", "Garib Rath", "Rajdhani"];
var i = 0;
var text = "";
for (;train[i];) {
    text += train[i] + "<br>";
    i++;
}
```

ಈ ಉದಾಹರಣೆಯಲ್ಲಿರುವ ಲೂಪ್, train array ದಿಂದ ಕಾರ್ ಹೆಸರುಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು while ಲೂಪ್ ಉಪಯೋಗಿಸುತ್ತದೆ.:

```
train = ["Duronto", "Satabdi", "Garib Rath", "Rajdhani"];
var i = 0;
var text = "";
while (train[i]) {
text += train[i] + "<br>";
    i++;
}
```

ಲೂಪ್‌ನಲ್ಲಿ break ಸ್ಟೇಟಮೆಂಟ್ (The Break Statement in Loop)

ಲೂಪ್ ಪೂರ್ಣಗೊಳ್ಳುವ ಮೊದಲು ಅದನ್ನು ಕೊನೆಗೊಳಿಸಬೇಕಾದರೆ break ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅನುಪಯುಕ್ತವಾಗಿ ಲೂಪನ್ನು ಪುನರಾವರ್ತನೆಗೊಳ್ಳುವುದನ್ನು ತಡೆದು ಯಂತ್ರದ ಸಮಯವನ್ನು ಉಳಿಸುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ : ಲೀನಿಯರ್ ಹುಡುಕಾಟದಲ್ಲಿ, ನಾವು ಬೇಕಾದ ಅಂಶವನ್ನು ಕಂಡುಕೊಂಡರೆ, ಅನವಶ್ಯಕವಾಗಿ ಲೂಪನ್ನು ರನ್ ಮಾಡುವ ಬದಲು ಅದನ್ನು break ಮಾಡಬಹುದು.

ಉದಾಹರಣೆ 14

```
for(i=0;i<l;i++) {
        if(arr[i]==item) {
                alert("Found at :"+i);
                fl=1;
                break;
                }
if(fl==0) alert("Not Found");
```

ಇಲ್ಲಿ ಐಟಂ ಕಂಡುಬಂದರೆ, ಲೂಪ್ break ಆಗುತ್ತದೆ ಮತ್ತು CPU ಸಮಯ ಉಳಿಯುತ್ತದೆ.

ಪಾಪ್ ಅಪ್ ಬಾಕ್ಸ್‌ಗಳು (Popup Boxes)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೂರು ರೀತಿಯ ಪಾಪ್‌ಅಪ್ ಬಾಕ್ಸ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಅವು :

1 ಅಲರ್ಟ್ ಬಾಕ್ಸ್ (Alert box)

2 ಕನ್ಫರ್ಮ್ ಬಾಕ್ಸ್ ಮತ್ತು (Confirm box)

3 ಪ್ರಾಂಪ್ಟ್ ಬಾಕ್ಸ್ (Prompt box)

ಅಲರ್ಟ್ ಬಾಕ್ಸ್ (Alert Box)

ನೀವು ಬಳಕೆದಾರರಿಂದ ಮಾಹಿತಿಯು ಬರುತ್ತಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಹೆಚ್ಚಾಗಿ ಅಲರ್ಟ್ ಬಾಕ್ಸ್‌ನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅಲರ್ಟ್ ಬಾಕ್ಸ್ ಪಾಪ್ ಅಪ್ ಆದಾಗ , ಬಳಕೆದಾರರು ಮುಂದುವರೆಯಲು "OK" ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

window.alert("sometext");

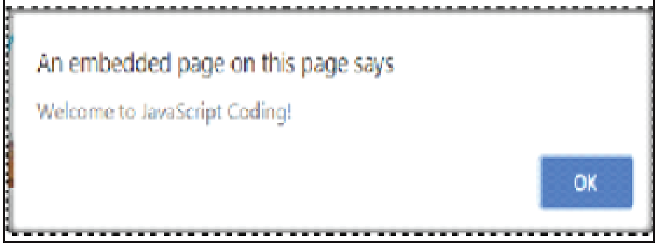
ಸೂಚನೆ : window.alert() method ನ್ನು windowಪೂರ್ವಪ್ರತ್ಯವಿಲ್ಲದೇ ಬರೆಯಬಹುದು.

ಉದಾಹರಣೆ 15

alert ("Welcome to Java Script Coding!;")

ಫಲಿತಾಂಶವನ್ನು ಚಿತ್ರ 1 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

Fig 1



ಕನ್ಫರ್ಮ್ ಬಾಕ್ಸ್ (Confirm box)

ವಿಷಯವನ್ನು ಪರಿಶೀಲಿಸಲು ಅಥವಾ ಸ್ವೀಕರಿಸಲು ಕನ್ಫರ್ಮ್ ಬಾಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಕನ್ಫರ್ಮ್ ಬಾಕ್ಸ್ ಪಾಪ್ ಅಪ್ ಆದಾಗ, ಬಳಕೆದಾರರು ಮುಂದುವರೆಯಲು "OK" or "Cancel" ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಬಳಕೆದಾರರು "OK" ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ ಬಾಕ್ಸ್ true ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಬಳಕೆದಾರರು "Cancel" ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ ಬಾಕ್ಸ್ false ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

window.confirm("sometext");

ಸೂಚನೆ : window.confirm() method ನ್ನು window ಪೂರ್ವಪ್ರತ್ಯಯವಿಲ್ಲದೇ ಬರೆಯಬಹುದು.

ಉದಾಹರಣೆ 16

```
if (confirm("Click a button!"))

{

    txt = "You clicked OK!";

}

else

{

    txt = "You clicked Cancel!";

}
```

ಫಲಿತಾಂಶವನ್ನು ಚಿತ್ರ 2 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

Fig 2

ಸೂಚನೆ : OK ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದ ಮೇಲೆ "You clicked OK!"ಎನ್ನುವ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ ಮತ್ತು Cancel ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ "You clicked Cancel!"ಎನ್ನುವ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ.

ಪ್ರಾಂಪ್ಟ್ ಬಾಕ್ಸ್ (Prompt box)

ಪುಟವನ್ನು ಪ್ರವೇಶಿಸುವ ಮೊದಲು ಬಳಕೆದಾರರು ಏನಾದರೂ ಮೌಲ್ಯವನ್ನು ಇನ್ಪುಟ್ ಮಾಡಬೇಕಾದರೆ, ಪ್ರಾಂಪ್ಟ್ ಬಾಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪ್ರಾಂಪ್ಟ್ ಬಾಕ್ಸ್ ಪಾಪ್ ಅಪ್ ಆದಾಗ ಬಳಕೆದಾರರು ಇನ್ಪುಟ್ ಮೌಲ್ಯವನ್ನು ನಮೂದಿಸಿದ ನಂತರ ಮುಂದುವರೆಯಲು "OK" ಅಥವಾ "Cancel" ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಬಳಕೆದಾರರು "OK" ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ ಬಾಕ್ಸ್ ಇನ್ಪುಟ್ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಬಳಕೆದಾರರು "Cancel" ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ ಬಾಕ್ಸ್ null ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

window.prompt("sometext","default text");

ಸೂಚನೆ : window.prompt() method ನ್ನು window ಪೂರ್ವಪ್ರತ್ಯಯವಿಲ್ಲದೇ ಬರೆಯಬಹುದು.

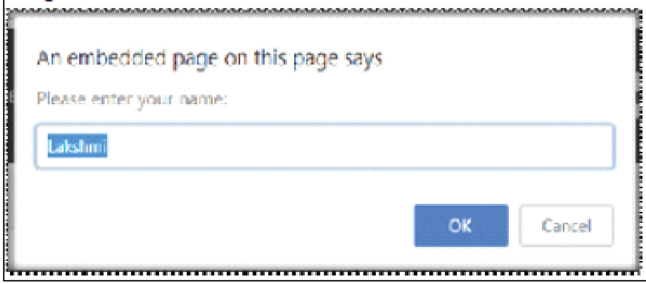
ಉದಾಹರಣೆ 17

```
var tname = prompt ("Please Enter your Name", "Lakshmi");

if (tname == null || tname == "")

{txt = "User cancelled the prompt";

}

else

{txt = "Hello " + tname + "! Congratulations!!!!!");

}
```

ಫಲಿತಾಂಶವನ್ನು ಚಿತ್ರ 3 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

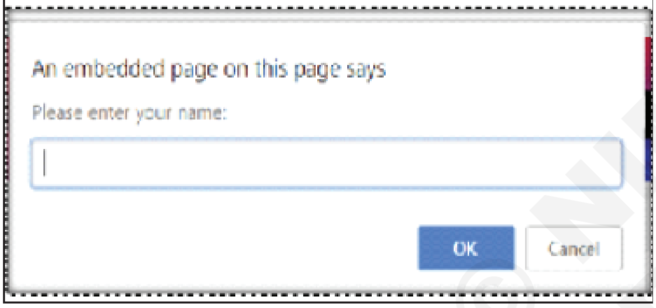
© NIMI NOT TO BE REPUBLISHED

Fig 3



ಸೂಚನೆ : OK ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದರೆ, "Hello Lakshmi! Congratulations!!!!! "ಎಂಬ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ. ಚಿತ್ರ 4 ರಲ್ಲಿ ತೋರಿಸಿರುವಂತೆ 'Lakshmi' ಹೆಸರನ್ನು ರದ್ದುಗೊಳಿಸಿದರೆ, ಅದು "User cancelled the Prompt" ಎಂಬ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ. Cancel ಬಟನ್‌ನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ , ಬಾಕ್ಸ್‌ನಲ್ಲಿ 'Lakshmi' ಹೆಸರು ಇದ್ದರೂ ಸಹ ಅದು "User cancelled the Prompt"ಎಂಬ ಸಂದೇಶವನ್ನು ತೋರಿಸುತ್ತದೆ

Fig 4



ಲೈನ್ ಬ್ರೆಕ್ಸ್ (Line Breaks)

ಪಾಪ್‌ಅಪ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಲೈನ್ ಬ್ರೆಕ್‌ಗಳನ್ನು ಪ್ರದರ್ಶಿಸಲು, ಒಂದು ಬ್ಯಾಕ್‌ಸ್ಲ್ಯಾಶ್ ನಂತರ n ಅಕ್ಷರವನ್ನು ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ 18

alert("Hello\nWelcome!");

ಫಲಿತಾಂಶವನ್ನು ಚಿತ್ರ 5 ರಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

Fig 5

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು HTML ಪುಟಾಗಳೊಂದಿಗೆ ಸೇರಿಸುವುದು (JavaScript Embed JavaScript in HTML Pages)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಅರೇಗಳು (Arrays in JavaScript)

ಉದ್ದೇಶಗಳು: ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಅರೇ ಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ**
- **ಅರೇ ಪರಿಕಲ್ಪನೆಗಳನ್ನು ವಿವರಿಸಿ**
- **ಅರೇ ಮೆಥೆಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ**
- **ಅರೇಯ ವಿಂಗಡನೆಯನ್ನು ತಿಳಿಯಿರಿ.**

ಅರೇ ಎಂದರೇನು? (What is an Array?)

ಅರೇ ಒಂದು ವಿಶೇಷವಾದ ವೇರಿಯೇಬಲ್ ಆಗಿದೆ. ಇದು ಒಂದು ಸಮಯದಲ್ಲಿ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು.

ನೀವು ಐಟಂಗಳ ಪಟ್ಟಿಯನ್ನು ಹೊಂದಿದ್ದರೆ (ಉದಾಹರಣೆಗೆ : ರೈಲುಗಳ ಹೆಸರಿನ ಪಟ್ಟಿ) ಒಂದೇ ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ರೈಲುಗಳನ್ನು ಸಂಗ್ರಹಿಸುವುದಾದರೆ, ಅದು ಈ ರೀತಿಯಾಗಿ ಕಾಣುತ್ತದೆ.

var train1 = "Jan Satabdi";

var train1 = "Garib Rath";

var train1 = "Duronto";

ಅದಾಗ್ಯೂ , ನೀವು train ಹೆಸರುಗಳನ್ನು ಲೂಪ್ ಮಾಡಿ , ಒಂದು ನಿರ್ದಿಷ್ಟವಾದ train ಹೆಸರನ್ನು ಹುಡುಕಲು ಬಯಸಿದರೆ? ಮತ್ತು ನೀವು 3 ಟ್ರೇನ್‌ಗಳ ಬದಲಾಗಿ 300 ಟ್ರೇನ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ ಏನಾಗಬಹುದು ?

ಅರೇ ಇದಕ್ಕೆ ಪರಿಹಾರವಾಗಿದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳು (JavaScript Arrays)

ಒಂದೇ ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಅರೇಗಳನ್ನು ರಚಿಸುವುದು (Creating an Array)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳನ್ನು ರಚಿಸಲು ಅರೇ ಲಿಟರಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು ಅತ್ಯಂತ ಸುಲಭವಾದ ಮಾರ್ಗವಾಗಿದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

var array-name = [item1, item2, ...];

ಉದಾಹರಣೆ 1

var trains = ["Duronto" , "Jan Satabdi", "RAJDHANI"];

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೀವರ್ಡ್ new ಉಪಯೋಗಿಸುವುದು (Using the JavaScript Keyword new)

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು ಸಹ ಅರೇಯನ್ನು ರಚಿಸುತ್ತದೆ ಮತ್ತು ಅದಕ್ಕೆ ಮೌಲ್ಯಗಳನ್ನೂ ನಿಯೋಜಿಸುತ್ತದೆ :

ಉದಾಹರಣೆ 2

var trains = new Array("Duronto", "Jan Satabdi", "RAJDHANI");

ಮೇಲಿನ ಎರಡೂ ಉದಾಹರಣೆಗಳೂ ಒಂದೇ ರೀತಿ ಕೆಲಸವನ್ನು ಮಾಡುತ್ತವೆ. new Array() ಯನ್ನು ಬಳಸುವ ಅಗತ್ಯವಿಲ್ಲ.

ಸರಳತೆ, ಓದುವಿಕೆ ಮತ್ತು ವೇಗವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದಕ್ಕಾಗಿ , ಮೊದಲನೇಯ ವಿಧಾನವನ್ನು ಬಳಸಿರಿ.(ಅರೇ ಲಿಟರಲ್ ವಿಧಾನ)

ಅರೇ ಎಲಿಮೆಂಟ್ (ಅಂಶಗಳು) ಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದು (Access the Elements of an Array)

ಅರೇಯ ಇಂಡೆಕ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಉಲ್ಲೇಖಿಸುವುದರ ಮೂಲಕ ಅರೇ ಎಲಿಮೆಂಟ್‌ನ್ನು ಉಲ್ಲೇಖಿಸಬಹುದು.

ಈ ಕೆಳಗಿನ ಹೇಳಿಕೆಯು myTrains ನ ಮೊದಲನೇ ಎಲಿಮೆಂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ.

var name = trains[0];

ಈ ಕೆಳಗಿನ ಹೇಳಿಕೆಯು trains ನ ಮೊದಲನೇಯ ಎಲಿಮೆಂಟ್‌ನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುತ್ತದೆ.

trains[0] = "Jan Satabdi";

[0] ಇದು ಅರೇಯ ಮೊದಲನೇ ಎಲಿಮೆಂಟ್ ಆಗಿರುತ್ತದೆ. [1] ಇದು ಎರಡನೆ ಎಲಿಮೆಂಟ್ ಆಗಿರುತ್ತದೆ. ಅರೇ ಇಂಡೆಕ್ಸ್ 0 ದಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ಅರೇಗಳನ್ನು ಪ್ರದರ್ಶಿಸುವುದು (Displaying Arrays): ಅರೇಗಳನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಪ್ರದರ್ಶಿಸಲು <p> ಎಲಿಮೆಂಟ್‌ನಲ್ಲಿ id= "demo" ನೊಂದಿಗೆ ನಾವು script ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ :

ಉದಾಹರಣೆ 3

```
<p id= "demo"></p>

<script>

var trains = ["Duronto", "Jan Satabdi" , "RAJDHANI"];

document.getElementById("demo").innerHTML = trains;

</script>
```

ಮೊದಲನೇಯ ಸಾಲು (script ನಲ್ಲಿ) trains ಎನ್ನುವ ಅರೇಯನ್ನು ರಚಿಸುತ್ತದೆ.

ಎರಡನೇಯ ಸಾಲು id="demo" ಎನ್ನುವ ಎಲಿಮೆಂಟನ್ನು "ಹುಡುಕುತ್ತದೆ ", ಮತ್ತು ಅರೇಯನ್ನು ಅದರ "innerHTML" ನಲ್ಲಿ "ಪ್ರದರ್ಶಿಸುತ್ತದೆ".

ಸ್ಥಳ ಮತ್ತು ಲೈನ್ ಬ್ರೆಕ್‌ಗಳು ಮುಖ್ಯವಲ್ಲ. ಅರೇ ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು ಅನೇಕ ಸಾಲುಗಳಲ್ಲಿ ವ್ಯಾಪಿಸಬಹುದು.

ಉದಾಹರಣೆ 4

```
var trains = [
"Duronto" ,
    "Jan Satabd",
    "RAJDHANI"
];
```

ಕೊನೆಯ ಎಲಿಮೆಂಟ್ ಆದ ಮೇಲೆ ಅಲ್ಪವಿರಾಮವನ್ನು ಹಾಕಬೇಡಿ ("RAJDHANI",ಯಂತೆ) ಇದು ಬ್ರೌಸರ್‌ಗಳಲ್ಲಿ ಅಸಮಂಜಸವಾಗಿದೆ.

ಒಂದು ಅರೇ ಒಂದೇ ಹೆಸರಿನಡಿಯಲ್ಲಿ ಅನೇಕ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು ಮತ್ತು ಇಂಡೆಕ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಉಲ್ಲೇಖಿಸಿ ನೀವು ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ನೀವು ಒಂದು ಅರೇನಲ್ಲಿ ವಿಭಿನ್ನ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು (You can have different objects in one array)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿರಬಹುದು. ಅರೇಗಳು ವಿಶೇಷ ರೀತಿಯ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ.

ಈ ಕಾರಣದಿಂದಾಗಿ, ಒಂದೇ ಅರೇನಲ್ಲಿ ನೀವು ವಿಭಿನ್ನ ಪ್ರಕಾರದ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಹೊಂದಬಹುದು.

ನೀವು ಅರೇನಲ್ಲಿ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ನೀವು ಅರೇನಲ್ಲಿ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ನೀವು ಅರೇನಲ್ಲಿ ಅರೇಗಳನ್ನು ಹೊಂದಿರಬಹುದು. :

```
my Array[0] = Date.now;

my Array [1] = my Function;

my Array [2] = myTrains;
```

ಅರೇಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ (Arrays are Objects)

ಅರೇಗಳು ವಿಶೇಷ ರೀತಿಯ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲಿ ್ಲಯ typeof ಆಪರೇಟರ್ ಅರೇಗಳನ್ನು " Objects" ಗಳೆಂದು ತೋರಿಸುತ್ತದೆ.

ಆದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳನ್ನು ಅರೇಗಳು ಎಂದೇ ವಿವರಿಸಲಾಗುತ್ತದೆ.

ಅರೇಗಳು ತನ್ನ "ಎಲಿಮೆಂಟ್"ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಸಂಖ್ಯೆಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, person[0] ಇದು Raja ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಅರೇ (Array)

```
var person = ["Raja" , "Sen" , 46];
```

ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ತನ್ನ "members" ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಹೆಸರುಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ , person.firstName ಇದು Raja ಹಿಂತಿರುಗಿಸುತ್ತದೆ.:

ಆಬ್ಜೆಕ್ಟ್ (Object)

```
var person = {firstName: "Raja" , lastName: "Sen" , age:46};
```

ಲೆಂಥ್ ಪ್ರಾಪರ್ಟಿ(The length Property)

ಒಂದು ಅರೇಯ ಲೆಂಥ್ ಪ್ರಾಪರ್ಟಿ, ಅರೇಯ ಉದ್ದ ಎಂದರೆ ಅದರಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ (ಅರೆ ಎಲಿಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆ)

ಉದಾಹರಣೆ 6

```
var fruits = ["Banana" , "Orange" , "Apple" , "Mango"];

fruits.length; //  fruits ನ ಲೆಂಥ್ 4 ಆಗಿದೆ.
```

ಲೆಂಥ್ ಪ್ರಾಪರ್ಟಿ ಯಾವಾಗಲೂ ಗರಿಷ್ಟ ಅರೇ ಇಂಡೆಕ್ಸ್‌ಗಿಂತಲೂ ಒಂದು ಹೆಚ್ಚಾಗಿರುತ್ತದೆ.

ಅರೇಗೆ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು (Adding Array Elements)

ಅರೇಗೆ ಒಂದು ಹೊಸ ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸಲು ಅತ್ಯಂತ ಸುಲಭವಾದ ಮಾರ್ಗವೆಂದರೆ ಲೆಂಥ್ ಪ್ರಾಪರ್ಟಿಯನ್ನು ಬಳಸುವುದು.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 7

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

fruits[fruits.length] = "Lemon"; // ಒಂದು ಹೊಸ ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ. fruits ಗೆ (Lemon) ನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಗರಿಷ್ಟ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹೊಂದಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ವ್ಯಾಖ್ಯಾನಿಸದ (undefined) "holes" ನ್ನು ರಚಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 8

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

fruits[10] = "Lemon"; // ಹೊಸ ಎಲಿಮೆಂಟ್ (Lemon) ನ್ನು fruits ಗೆ ಸೇರಿಸುತ್ತದೆ.

ಅರೇ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಲೂಪ್ ಮಾಡುವುದು (Looping Array Elements)

ಒಂದು ಅರೇಯ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಲೂಪ್ ಮಾಡುವ ಉತ್ತಮ ಮಾರ್ಗವೆಂದರೆ for ಲೂಪ್ ಬಳಸುವುದು.

ಉದಾಹರಣೆ 9

```
var index;

var fruits = ["Banana", "Orange", "Apple", "Mango"];

for (index = 0; index < fruits.length; index++) {

text += fruits[index];

}
```

ಅಸೋಸಿಯೇಟಿವ್ ಅರೇಗಳು? ಅಸಾಧ್ಯ (Associative Arrays? No Way!)

ಅನೇಕ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳು ಹೆಸರಿಸಲಾದ ಇಂಡೆಕ್ಸ್‌ಗಳನ್ನು ಹೊಂದಿದ ಅರೇಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತವೆ.

ಹೆಸರಿಸಲಾದ ಇಂಡೆಕ್ಸ್‌ಗಳನ್ನು ಹೊಂದಿದ ಅರೇಗಳಿಗೆ ಅಸೋಸಿಯೇಟಿವ್ ಅರೇಗಳು ಎಮ್ದು ಕರೆಯಲಾಗುತ್ತದೆ.(ಅಥವಾ ಹ್ಯಾಶ್‌ಗಳು)

ಹೆಸರಿಸಲಾದ ಇಂಡೆಕ್ಸ್ ಗಳನ್ನು ಹೊಂದಿದ ಅರೇಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬೆಂಬಲಿಸುವುದಿಲ್ಲ.

ಕೆಳಗಿನ ಡಿಕ್ಲೇರೇಷನ್ ತಪ್ಪಾಗಿದೆ :

```
var person = new Array()

person ["firstName"] = "Raja";

person ["lastName"] = "Sen";

person ["age"] = 46;
```

ಮೇಲಿನ ಉದಾಹರಣೆಯು ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತೆ ತೋರುತ್ತದೆ. ಆದರೆ ಅದು ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದಿಲ್ಲ.

ನೀವು ಇದನ್ನು ಪ್ರಯತ್ನಿಸಿದರೆ, person["firstName"] ಇದು Raja ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ. ಆದರೆ, person[0] ಇದು undefined ಎಂದು , ಮತ್ತು person.length ಇದು 0 ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ನೀವು ಅಸೋಸಿಯೇಟಿವ್ ಅರೇಯನ್ನು ರಚಿಸಲು ಬಯಸಿದರೆ, ಅದರ ಬದಲಾಗಿ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸಿರಿ.

ಅರೇಗಳನ್ನು ಯಾವಾಗ ಬಳಸಬೇಕು? ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಯಾವಾಗ ಬಳಸಬೇಕು ? (When to Use Arrays? When to use Objects?)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅಸೋಸಿಯೇಟಿವ್ ಅರೇಗಳನ್ನು ಬೆಂಬಲಿಸುವುದಿಲ್ಲ.

ನೀವು ಎಲಿಮೆಂಟ್ ಹೆಸರುಗಳು ಸ್ಟ್ರಿಂಗ್‌ಗಳಾಗಿರಬೇಕು ಎಂದು ಬಯಸಿದರೆ, ನೀವು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ನೀವು ಎಲಿಮೆಂಟ್ ಹೆಸರುಗಳು ಅನುಕ್ರಮ ಸಂಖ್ಯೆಗಳಾಗಿರಬೇಕು ಎಂದು ಬಯಸಿದರೆ, ಅರೇಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

new Array() ಉಪಯೋಗವನ್ನು ತಪ್ಪಿಸಿ (Avoid new Array()

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಬಿಲ್ಟ್- ಇನ್(ಅಂತರ್‌ನಿರ್ಮಿತ) ಕನ್ಸ್ಟ್ರಕ್ಟರ್ (constructor) new Array() ಯನ್ನು ಉಪಯೋಗಿಸುವ ಅಗತ್ಯವಿಲ್ಲ.

ಅದರ ಬದಲಾಗಿ [] ಉಪಯೋಗಿಸಿರಿ.

ಈ ಕೆಳಗಿನ ಎರಡು ವಿಭಿನ್ನ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು points ಎಂಬ ಹೆಸರಿನ ಹೊಸ ಖಾಲಿ ಅರೇಯನ್ನು ರಚಿಸುತ್ತವೆ.

```
var points = new Array();        // ಕೆಟ್ಟ ವಿಧಾನ

var points = [];                 // ಸರಿಯಾದ ವಿಧಾನ
```

ಈ ಕೆಳಗಿನ ಎರಡು ವಿಭಿನ್ನ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು points ಎಂಬ ಹೆಸರಿನ 6 ಸಂಖ್ಯೆಗಳನ್ನು ಹೊಂದಿದ ಹೊಸ ಅರೇಯನ್ನು ರಚಿಸುತ್ತವೆ.

```
var points = new Array(40, 100, 1, 5, 25, 10) // ಕೆಟ್ಟ ವಿಧಾನ

var points = [40, 100, 1, 5, 25, 10];     // ಸರಿಯಾದ ವಿಧಾನ
```

new ಕೀವರ್ಡ್ ನಿಮ್ಮ ಕೋಡನ್ನು ಸಂಕೀರ್ಣಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಕೆಟ್ಟದಾದ ಪರಿಣಾಮವನ್ನು ಉಂಟುಮಾಡುತ್ತದೆ.

var points = new Array(40, 100); // ಎರಡು ಎಲಿಮೆಂಟ್‌ಗಳಿರುವ ಒಂದು ಅರೇಯನ್ನು ರಚಿಸುತ್ತದೆ. (40 ಮತ್ತು 100)

ಎರಡು ಎಲಿಮೆಂಟ್‌ಗಳಲ್ಲಿ ಒಂದನ್ನು ನಾನು ತೆಗೆದುಹಾಕಿದರೆ?

var points = new Array(40); // 40 undefined ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿದ ಒಂದು ಅರೇಯನ್ನು ರಚಿಸುತ್ತದೆ.!!!!!

© NIMI NOT TO BE REPUBLISHED

ಅರೇಯನ್ನು ಹೇಗೆ ಗುರುತಿಸುವುದು? (How to Recognize an Array?)

ಒಂದು ಸಾಮಾನ್ಯ ಪ್ರಶ್ನೆ ಎಂದರೆ : ಒಂದು ವೇರಿಯೇಬಲ್ ಅರೇ ಆಗಿದ್ದರೆ ಹೇಗೆ ತಿಳಿಯುವುದು ?

ಇದರಲ್ಲಿ ಸಮಸ್ಯೆಯೆಂದರೆ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ typeof ಆಪರೇಟರ್ ಅರೇಯನ್ನು "ಆಬ್ಜೆಕ್ಟ್" ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

```
var fruits = ["Banana", "Orange", "Apple","Mango"];

typeof fruits;          // typeof, ಆಬ್ಜೆಕ್ಟ್ ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿರುವುದರಿಂದ typeof ಆಪರೇಟರ್ ಅರೇಯನ್ನು ಆಬ್ಜೆಕ್ಟ್ ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಈ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ನೀವು ನಿಮ್ಮ ಸ್ವಂತ isArray() ಫಂಕ್ಷನ್‌ನ್ನು ರಚಿಸಬಹುದು.:

```
function isArray(myArray) {

return myArray.constructor.toString().indexOf("Array") > 1;

}
```

ಆರ್ಗ್ಯುಮೆಂಟ್ ಅರೇ ಆಗಿದ್ದರೆ, ಮೇಲಿನ ಫಂಕ್ಷನ್ ಯಾವಾಗಲೂ true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಅಥವಾ ಹೆಚ್ಚು ನಿಖರವಾಗಿ ಹೇಳುವುದಾದರೆ : ಆರ್ಗ್ಯುಮೆಂಟ್‌ನ ಆಬ್ಜೆಕ್ಟ್ ಪ್ರೊಟೊಟೈಪ್ "[object array]" ಆಗಿದ್ದರೆ ಅದು true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇ ಮೆಥೆಡ್‌ಗಳು (JavaScript Array Methods)

ಅರೇಗಳನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನಾಗಿ ಪರಿವರ್ತಿಸುವುದು (Converting Arrays to Strings)

toString() ಮೆಥೆಡ್ (toString() method)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೆಥೆಡ್ ಆದ toString() ಅರೇಯನ್ನು, ಅರೇ ಮೌಲ್ಯಗಳ ಸ್ಟ್ರಿಂಗ್‌ಗೆ (ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಲ್ಪಟ್ಟ) ಪರಿವರ್ತಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 10

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];

document.getElementById("demo").innerHTML = trade.toString();
```

ಫಲಿತಾಂಶ

COPA,IT,ICTSM,CHNM,Fitter

join() ಮೆಥೆಡ್ (join() method)

join() ಮೆಥೆಡ್, ಎಲ್ಲ ಅರೇ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸ್ಟ್ರಿಂಗ್ ಆಗಿ ಸೇರಿಸುತ್ತದೆ. ಇದು toString(), ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ ಇದರಲ್ಲಿ ನೀವು ವಿಭಜಕವನ್ನು (separator) ನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಬಹುದು.

ಉದಾಹರಣೆ 11

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];

document.getElementById("demo").innerHTML = trade.join(" - ");
```

ಫಲಿತಾಂಶ

COPA - IT - ICTSM - CHNM - Fitter

ಪಾಪಿಂಗ್ ಮತ್ತು ಪುಶಿಂಗ್ (Popping and Pushing)

ನೀವು ಅರೇಗಳಲ್ಲಿ ಕೆಲಸ ಮಾಡುವಾಗ, ಅದರ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುವುದು ಮತ್ತು ಹೊಸ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಸುಲಭವಾಗಿದೆ.

ಪಾಪಿಂಗ್ (Popping)

pop() ಮೆಥೆಡ್, ಅರೇಯಲ್ಲಿರುವ ಕೊನೆಯ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದು ಹಾಕುತ್ತದೆ.

ಉದಾಹರಣೆ 12

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];

trade.pop(); // trade ಅರೇನಿಂದ ಕೊನೆಯ ಎಲಿಮೆಂಟ ಆದ
             "Fitter" ತೆಗೆದುಹಾಕುತ್ತದೆ.
```

ಫಲಿತಾಂಶ

COPA,IT,ICTSM,CHNM

pop() ಮೆಥೆಡ್ "popped out" ಆದ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 13

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];

var x = trade.pop();  // x ನ ಮೌಲ್ಯ "CHNM" ಆಗಿರುತ್ತದೆ.
```

ಪುಶಿಂಗ್ (Pushing)

push() ಮೆಥೆಡ್, ಅರೇಗೆ ಒಂದು ಹೊಸ ಎಲಿಮೆಂಟನ್ನು ಸೇರಿಸುತ್ತದೆ (ಅರೇಯ ಕೊನೆಯಲ್ಲಿ)

ಉದಾಹರಣೆ 14

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade.push("DTPO");  // trade ಗೆ ಹೊಸ ಎಲಿಮೆಂಟ್
                     ("DTPO") ನ್ನು ಸೇರಿಸುತ್ತದೆ.
```


© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ

COPA,IT,ICTSM,CHNM,DTPO

push() ಮೆಥೆಡ್ ಹೊಸ ಅರೇ ಲೆಂಥನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 15

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

var x = trade.push("DTPO"); // x ನ ಮೌಲ್ಯ 5 ಆಗಿರುತ್ತದೆ.

ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಶಿಫ್ಟ್ ಮಾಡುವುದು (Shifting Elements)

ಶಿಫ್ಟಿಂಗ್ ಕೂಡ ಪಾಪಿಂಗ್‌ನಂತೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ ಕೊನೆಯ ಎಲೆಮೆಂಟ್ ಬದಲಾಗಿ ಮೊದಲನೇ ಎಲೆಮೆಂಟ್ ಮೇಲೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ. shift () ಮೆಥೆಡ್ ಮೊದಲನೇ ಅರೇ ಎಲೆಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ ಮತ್ತು ಇತರ ಎಲ್ಲ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಕಡಿಮೆ ಇಂಡೆಕ್ಸ್‌ಗೆ "shift" ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 16

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade.shift(); // trade ನಿಂದ ಮೊದಲನೇ ಎಲೆಮೆಂಟ್ ಆದ "COPA" ವನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

ಫಲಿತಾಂಶ

IT,ICTSM,CHNM

shift() ಮೆಥೆಡ್ "shifted out" ಆದ ಸ್ಟ್ರಿಂಗನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ :

ಉದಾಹರಣೆ 17

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade.shift(); // "COPA" ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

unshift() ಮೆಥೆಡ್ ಅರೇಗೆ ಹೊಸ ಎಲೆಮೆಂಟನ್ನು ಸೇರಿಸುತ್ತದೆ (ಆರಂಭದಲ್ಲಿ), ಮತ್ತು ಹಳೆಯ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು "unshifts" ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 18

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

fruits.unshift("ElecMech"); // trade ಗೆ ಹೊಸ ಎಲೆಮೆಂಟ್ "ElecMech" ನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಫಲಿತಾಂಶ

ElecMech,COPA,IT,ICTSM,CHNM

unshift() ಮೆಥೆಡ್ ಹೊಸ ಅರೇ ಲೆಂಥನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 19

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade.unshift("ElecMech"); // 5 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Changing Elements)

ಅರೇ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಅವುಗಳ ಇಂಡೆಕ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಬಳಸಿಕೊಂಡು ಬದಲಾಯಿಸಬಹುದು: ಅರೇ ಇಂಡೆಕ್ಸ್‌ಗಳು 0 ದಿಂದ ಶುರುವಾಗುತ್ತವೆ. [0] ಇದು ಮೊದಲನೇ ಎಲೆಮೆಂಟ್ ಆಗಿರುತ್ತದೆ, [1] ಇದು ಎರಡನೇ ಎಲೆಮೆಂಟ್ ಆಗಿರುತ್ತದೆ, [2] ಇದು ಮೂರನೇ ಎಲೆಮೆಂಟ್ ಆಗಿರುತ್ತದೆ......

ಉದಾಹರಣೆ 20

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade[2] = "DTPO"; // trade ನ ಮೂರನೇ ಎಲೆಮೆಂಟ್‌ನ್ನು "DTPO" ಎಂದು ಬದಲಾಯಿಸುತ್ತದೆ.

ಫಲಿತಾಂಶ

COPA,IT,DTPO,CHNM

ಒಂದು ಅರೇಗೆ ಹೊಸ ಎಲೆಮೆಂಟನ್ನು ಸೇರಿಸಲು ಲೆಂಥ್ ಪ್ರಾಪರ್ಟೀ ಸುಲಭವಾದ ಮಾರ್ಗವಾಗಿದೆ.

ಉದಾಹರಣೆ 21

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

trade[trade.length] = "ICTSM"; // fruits ಗೆ "ICTSM" ನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಫಲಿತಾಂಶ

COPA,IT,DTPO,CHNM,ICTSM

ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುವುದು (Deleting Elements)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಅರೇಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿರುವುದರಿಂದ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಪರೇಟರ್ ಆದ delete ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಬಹುದು.

ಉದಾಹರಣೆ 22

var trade = ["COPA", "IT", "ICTSM", "CHNM"];

delete trade[0]; // trade ನಲ್ಲಿರುವ ಮೊದಲನೇ ಎಲೆಮೆಂಟನ್ನು undefined ಎಂದು ಬದಲಾಯಿಸುತ್ತದೆ.

> **ಸೂಚನೆ : delete ಉಪಯೋಗಿಸುವುದರಿಂದ ಅರೇನಲ್ಲಿ undefined holes ಗಳು ಉಳಿಯಬಹುದು. ಇದರ ಬದಲಾಗಿ pop() ಅಥವಾ shift() ಉಪಯೋಗಿಸಿ.**

ಅರೇಯನ್ನು ವಿಭಜಿಸುವುದು (Splicing an Array)

splice() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಅರೇನಲ್ಲಿ ಹೊಸ ಅಂಶಗಳನ್ನು ಸೇರಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 23

```
var trade = [“COPA”, “IT”, “ICTSM”, “CHNM”];
trade.splice(2, 0, “Turner”, “Machinist”);
```

> **ಸೂಚನೆ : ಮೊದಲನೇ ಪ್ಯಾರಾಮೀಟರ್ (2) ಹೊಸ ಎಲಿಮೆಂಟನ್ನು ಎಲ್ಲಿ ಸೇರಿಸಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ (spliced in). ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್ (0) ಎಷ್ಟು ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ. ಉಳಿದ ಎಲ್ಲ ಪ್ಯಾರಾಮೀಟರ್‌ಗಳು (“Turner”, “Machinist”)ಸೇರಿಸಬೇಕಾದ ಹೊಸ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತಿಳಿಸುತ್ತವೆ.**

ಫಲಿತಾಂಶ

COPA,IT,Turner,Machinist,DTPO,CHNM

ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಲು splice() ಉಪಯೋಗಿಸುವುದು (Using splice() to Remove Elements)

splice() ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು, ಅರೇನಲ್ಲಿ holes ನ್ನು ಬಿಡದೆ, ಚತುರತೆಯಿಂದ ಪ್ಯಾರಾಮೀಟರ ಸೆಟ್ಟಿಂಗ್‌ನೊಂದಿಗೆ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಬಹುದು.

ಉದಾಹರಣೆ 24

```
var trade = [“COPA”, “IT”, “DTPO”, “CHNM”];
trade.splice(0, 1); // trade ನಲ್ಲಿನ ಮೊದಲನೇ ಎಲಿಮೆಂಟನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.
```

ಫಲಿತಾಂಶ

IT,DTPO,CHNM

> **ಸೂಚನೆ : ಮೊದಲನೇ ಪ್ಯಾರಾಮೀಟರ್ (0) ಹೊಸ ಎಲಿಮೆಂಟನ್ನು ಎಲ್ಲಿ ಸೇರಿಸಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ (spliced in). ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್ (1) ಎಷ್ಟು ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ. ಉಳಿದ ಎಲ್ಲ ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ಬಿಟ್ಟು ಬಿಡಲಾಗುತ್ತದೆ . ಯಾವುದೇ ಹೊಸ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸಲಾಗುವುದಿಲ್ಲ.**

ಅರೇಗಳನ್ನು ವಿಲೀನಗೊಳಿಸುವುದು ಅಥವಾ ಜೋಡಿಸುವುದು (Merging or Concatenating Arrays)

concat() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಈಗಾಗಲೇಅಸ್ತಿತ್ವದಲ್ಲಿ ಇರುವ ಅರೇಗಳನ್ನು ವಿಲೀನಗೊಳಿಸಿ ಹೊಸ ಅರೇಯನ್ನು ತಯಾರಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 25 (ಎರಡು ಅರೇಗಳನ್ನು ವಿಲೀನಗೊಳಿಸುವುದು)

```
var names1 = [“Devi”, “Deepa”];
var names2 = [“Poorna”, “Saranya”, “Shalini”];
var myTrainee = names1.concat(names2);
// names1 ಮತ್ತು names2 ಅನ್ನು ಜೋಡಿಸುತ್ತದೆ (joins)
```

ಫಲಿತಾಂಶ

> **ಸೂಚನೆ: concat() ಮೆಥೆಡ್ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇರುವ ಅರೇಗಳನ್ನು ಬದಲಾಯಿಸುವುದಿಲ್ಲ. ಇದು ಯಾವಾಗಲೂ ಹೊಸ ಅರೇಯನ್ನು ತಯಾರಿಸುತ್ತದೆ.**

concat() ಮೆಥೆಡ್ ಅನೇಕ ಅರೇ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು.

ಉದಾಹರಣೆ 26 ಮೂರು ಅರೇಗಳನ್ನು ಜೋಡಿಸುವುದು (Merging Three Arrays)

```
var arr1 = [“Priya”, “Mythili”];
var arr2 = [“Sangeetha” , “Nancy”, “Sahana”];
var arr3 = [“Ramya” , “Kavi”];
var myTrainee = arr1.concat(arr2, arr3);
    // arr1 ನ್ನು arr2 ಮತ್ತು arr3 ರ ಜೊತೆಗೆ ಜೋಡಿಸುತ್ತದೆ.
```

concat() ಮೆಥೆಡ್ ಮೌಲ್ಯಗಳನ್ನೂ ಸಹ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳಾಗಿ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.

ಉದಾಹರಣೆ 27 ಅರೇಯನ್ನು ಮೌಲ್ಯಗಳ ಜೊತೆಗೆ ಜೋಡಿಸುವುದು (Merging an Array with Values)

```
var arr1 = [“Priya”, “Mythili”];
var myTrainee = arr1.concat([“Ramya” , “Kavi”]);
```

ಅರೇಯನ್ನು ವಿಭಜಿಸುವುದು (Slicing an Array)

slice() ಮೆಥೆಡ್ ಅರೇಯ ತುಂಡನ್ನು ಹೊಸ ಅರೇಯನ್ನಾಗಿ ವಿಭಜಿಸುತ್ತದೆ. ಈ ಉದಾಹರಣೆಯು ಅರೇಯ 2 ನೇ ಎಲಿಮೆಂಟ್‌ನಿಂದ (“DTPO”)ಪ್ರಾರಂಭವಾಗುವ ಅರೇಯ ಒಂದು ಭಾಗವನ್ನು ವಿಭಜಿಸುತ್ತದೆ. slice() ಮೆಥೆಡ್ ಹೊಸ ಅರೇಯನ್ನು ರಚಿಸುತ್ತದೆ. ಇದು ಮೂಲ ಅರೇನಿಂದ ಯಾವುದೇ ಎಲಿಮೆಂಟನ್ನು ತೆಗೆದುಹಾಕುವುದಿಲ್ಲ.

ಉದಾಹರಣೆ 28

```
var trade = [“COPA”, “IT”, “DTPO”, “CHNM”];
var trade1 = trade.slice(2);
```


© NIMI NOT TO BE REPUBLISHED

slice() ಮೆಥೆಡ್ slice(1, 3) ನಂತೆ ಎರಡು ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ. ನಂತರ ಈ ಮೆಥೆಡ್ ಮೊದಲನೇ ಆರ್ಗ್ಯುಮೆಂಟ್‌ನಿಂದ ಕೊನೆಯ ಆರ್ಗ್ಯುಮೆಂಟ್‌ವರೆಗೆ (ಆದರೆ ಒಳಗೊಂಡಿರುವುದಿಲ್ಲ) ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 29

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(1,3);
```

ಮೊದಲನೇ ಉದಾಹರಣೆಯಲ್ಲಿರುವಂತೆ, ಕೊನೆಯ ಆರ್ಗ್ಯುಮೆಂಟನ್ನು ಬಿಟ್ಟುಬಿಟ್ಟರೆ, slice() ಮೆಥೆಡ್ ಉಳಿದ ಅರೇಯನ್ನು ವಿಭಜಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 30

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(2);
```

ಅಟೋಮ್ಯಾಟಿಕ್ toString() (Automatic toString())

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಒಂದು ಅರೇಯನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ. ಎಲಿಮೆಂಟ್‌ಗಳು ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಲ್ಪಡುತ್ತವೆ. ನೀವು ಒಂದು ಅರೇಯನ್ನು ನೋಡಲು ಪ್ರಯತ್ನಿಸಿದಾಗ ಯಾವಾಗಲೂ ಇದು ಸಂಭವಿಸುತ್ತದೆ. ಈ ಎರಡೂ ಉದಾಹರಣೆಗಳು ಒಂದೇ ಫಲಿತಾಂಶವನ್ನು ನೀಡುತ್ತವೆ. :

ಉದಾಹರಣೆ 31

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById(ಕಿdemoೖ).innerHTML = trade.
toString();
```

ಉದಾಹರಣೆ 32

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById("demo").innerHTML = trade;
```

> **ಸೂಚನೆ : ಎಲ್ಲ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು** toString() **ಮೆಥೆಡನ್ನು ಹೊಂದಿರುತ್ತವೆ.**

ಅರೇಯನ್ನು ವಿಂಗಡಿಸುವುದು (Sorting an Array)

sort() ಮೆಥೆಡ್ ಅರೇಯನ್ನು ವರ್ಣಮಾಲೆಯ ಅಕ್ಷರದಂತೆ ವಿಂಗಡಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 33

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); //  trade ನ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ವಿಂಗಡಿಸುತ್ತದೆ.
```

ಫಲಿತಾಂಶ

CHNM,COPA,ICTSM,IT

ಅರೇಯನ್ನು ಹಿಮ್ಮುಖಗೊಳಿಸುವುದು (Reversing an Array)

reverse() ಮೆಥೆಡ್, ಅರೇನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹಿಮ್ಮುಖವಾಗಿ ವಿಂಗಡಿಸುತ್ತದೆ. ನೀವು ಇದನ್ನು ಅರೇಯನ್ನು ಅವರೋಹಣ ಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸಲು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 34

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); //  trade ನ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ವಿಂಗಡಿಸುತ್ತದೆ.
trade.reverse(); // ಎಲಿಮೆಂಟ್‌ಗಳ ಕ್ರಮವನ್ನು ಹಿಮ್ಮುಖಗೊಳಿ
ಸುತ್ತದೆ.
```

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಎನಿಮೇಷನ್ ಮತ್ತು ಮಲ್ಟಿಮೀಡಿಯಾ ಫೈಲ್‌ಗಳ ಪರಿಕಲ್ಪನೆಗಳು (Concepts of Animation and Multimedia files in JavaScript)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಅನಿಮೇಷನ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ತಿಳಿದುಕೊಳ್ಳುವುದು.**
- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಮಲ್ಟೀಮೀಡಿಯಾನ್ನು ವಿವರಿಸಿ.**

ಎನಿಮೇಷನ್ (Animation)

ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸ್ಟೈಲ್ ಮಾಡುವುದು (Styling the Elements)

ಎನಿಮೇಷನ್‌ನ್ನು ಸಾಧ್ಯವಾಗಿಸಲು, ಎನಿಮೇಟೆಡ್ ಎಲಿಮೆಂಟನ್ನು "parent container". ಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಎನಿಮೇಟೆಡ್ ಮಾಡಬೇಕು.

ಕಂಟೇನರ್ ಎಲಿಮೆಂಟ್‌ನ್ನು ಈ ರೀತಿಯಾಗಿ ರಚಿಸಬೇಕು style = "position: relative".

ಎನಿಮೇಷನ್ ಎಲಿಮೆಂಟ್‌ನ್ನು ಈ ರೀತಿಯಾಗಿ ರಚಿಸಬೇಕು style =

“position: absolute”.

ಉದಾಹರಣೆ

```
<!Doctype html>
<html>
<style>
#myContainer {
 width: 400px;
 height: 400px;
 position: relative;
 background: pink;
}
#myAnimation {
 width: 50px;
 height: 50px;
 position: absolute;
 background: green;
}
</style>
<body>
<h1>My First JavaScript Animation</h1>
<div id= “myContainer”>
<div id=“myAnimation”></div>
</div>
</body>
</html>
?
```

ಎನಿಮೇಷನ್ ಕೋಡ್ (The Animation Code)

ಎಲಿಮೆಂಟ್‌ನ ಸ್ಟೈಲ್‌ನಲ್ಲಿ ಕ್ರಮೇಣವಾಗಿ ಬದಲಾವಣೆಗಳನ್ನು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮಾಡುವ ಮೂಲಕ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಎನಿಮೇಷನ್‌ಗಳನ್ನು ಮಾಡಲಾಗುತ್ತದೆ:

ಉದಾಹರಣೆ

```
var id = setInterval(frame, 5);
function frame() {
if (/* test for finished */) {
clearInterval(id);
} else {
/* code to change the element style */
}
}
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು ಉಪಯೋಗಿಸಿ ಎನಿಮೇಷನ್ ರಚಿಸುವುದು(Create the Animation Using JavaScript)

ಉದಾಹರಣೆ

```
<style>`
#myContainer {
 width: 400px;
 height: 400px;
 position: relative;
 background: pink;
}
#myAnimation {
 width: 50px;
 height: 50px;
 position: absolute;
 background-color: green;
}
</style>
<body>
<p>
<button onclick= “myMove()”>Click Me</button>
</p>
<div id = “myContainer”>
<div id = “myAnimation”></div>
```

© NIMI NOT TO BE REPUBLISHED

```
</div>
<script>
function myMove() {
var elem = document.getElementById(“myAnimation”);
var pos = 0;
var id = setInterval(frame, 10);
function frame() {
if (pos == 350) {
clearInterval(id);
} else {
pos++;
elem.style.top = pos + ‘px’;
elem.style.left = pos + ‘px’;
}
}
}
</script>
</body>
</html>
```

ಮಲ್ಟಿಮೀಡಿಯಾ ಫೈಲ್‌ಗಳು (Multimedia files)

ಮಲ್ಟಿಮೀಡಿಯಾ ಎಂದರೇನು? What is Multimedia?

ಮಲ್ಟಿಮೀಡಿಯಾ ವಿವಿಧ ಸ್ವರೂಪಗಳಲ್ಲಿ ಬರುತ್ತದೆ. ಇದು ನೀವು ಕೇಳಬಹುದಾದ ಮತ್ತು ನೋಡಬಹುದಾದ ಯಾವುದೂ ಆಗಿರಬಹುದು. ವೆಬ್ ಪುಟಗಳೂ ಸಹ ವಿಭಿನ್ನ ರೀತಿಯ ಮತ್ತು ಸ್ವರೂಪಗಳ ಮಲ್ಟಿಮೀಡಿಯಾ ಅಂಶಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಉದಾಹರಣೆಗೆ : ಚಿತ್ರಗಳು, ಸಂಗೀತ, ಧ್ವನಿ, ವಿಡಿಯೊಗಳು, ರೆಕಾರ್ಡ್‌ಗಳು, ಚಲನಚಿತ್ರಗಳು, ಎನಿಮೇಷನ್‌ಗಳು ಇತ್ಯಾದಿ.

ಮಲ್ಟಿಮೀಡಿಯಾ ಸ್ವರೂಪಗಳು (Multimedia Formats)

ಮಲ್ಟಿಮೀಡಿಯಾ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು (ಉದಾ ಆಡಿಯೋ ಅಥವಾ ವಿಡಿಯೋ) ಮೀಡಿಯಾ ಫೈಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ. ಫೈಲ್‌ನ ಪ್ರಕಾರವನ್ನು ಕಂಡು ಹಿಡಿಯುವ ಸಾಮಾನ್ಯವಾದ ಮಾರ್ಗವೆಂದರೆ ಆದರ ವಿಸ್ತರಣೆ(extension) ನ್ನು ನೊಡುವುದು. ಉದಾಹರಣೆಗೆ : .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

HTML ನಲ್ಲಿ ವಿಡಿಯೋಗಳನ್ನು ನೋಡುವುದು (Playing Videos in HTML)

HTML ನಲ್ಲಿ ವಿಡಿಯೋವನ್ನು ತೋರಿಸಲು , <video> ಎಲಿಮೆಂಟನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ:

ಉದಾಹರಣೆ

```
<video width=”320” height=”240” controls>
<source src=”movie.mp4” type=”video/mp4”>
<source src=”movie.ogg” type=”video/ogg”>
Your browser does not support the video tag.
</video>
```

ಇದು ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ (How it Works)

ಕಂಟ್ರೋಲ್ ಅಟ್ರಿಬ್ಯೂಟ್, (controls attribute) play, pause,ಮತ್ತು volume ನಂತಹ ವಿಡಿಯೊ ಕಂಟ್ರೋಲ್‌ಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ. width ಮತು height attributes ಗಳನ್ನು ಸೇರಿಸುವುದು ಒಳ್ಳೆಯದು. height ಮತ್ತು width ಗಳನ್ನು ಸೆಟ್ ಮಾಡದಿದ್ದರೆ, ವಿಡಿಯೋ ಲೋಡ್ ಆಗುವಾಗ ಪುಟವು ಮಿನುಗಬಹುದು. The <source> ಎಲಿಮೆಂಟ್ ಬ್ರೌಸರ್‌ಗೆ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಪರ್ಯಾಯ ವಿಡಿಯೋ ಫೈಲ್‌ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು ನಿಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಬ್ರೌಸರ್ ಮೊದಲನೇಯ ಗುರುತಿಸಲ್ಪಟ್ಟ ಸ್ವರೂಪವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. <video> ಎಲಿಮೆಂಟನ್ನು ಬೆಂಬಲಿಸದೇ ಇರುವ ಬ್ರೌಸರ್‌ಗಳಲ್ಲಿ <video> and </video> ಟ್ಯಾಗ್‌ಗಳ ಮಧ್ಯದಲ್ಲಿನ ಟೆಕ್ಸ್ಟ್ ಮಾತ್ರ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

HTML <video> Autoplay

ಅಟೊಮ್ಯಾಟಿಕ್ ಆಗಿ ವಿಡಿಯೊವನ್ನು ಪ್ರಾರಂಭಿಸಲು autoplay ಅಟ್ರಿಬ್ಯೂಟನ್ನು ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ

```
<video width=”320” height=”240” autoplay>
<source src=”movie.mp4” type=”video/mp4”>
<source src=”movie.ogg”, type=”video/ogg”>
Your browser does not support the video tag.
</video>
```

> **ಸೂಚನೆ : Autoplay ಅಟ್ರಿಬ್ಯೂಟ್ iPad ಮತ್ತು iPhone ನಂತಹ ಮೊಬೈಲ್ ಸಾಧನಗಳಲ್ಲಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದಿಲ್ಲ.**

© NIMI NOT TO BE REPUBLISHED

HTML ವಿಡಿಯೋ - ಮೀಡಿಯಾ ಪ್ರಕಾರಗಳು (HTML Video - Media Types)

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML ವಿಡಿಯೋ - ಮೆಥೆಡ್‌ಗಳು, ಪ್ರಾಪರ್ಟೀಗಳು ಮತ್ತು ಇವೆಂಟ್‌ಗಳು (HTML Video - Methods, Properties, and Events)

HTML5, <video> ಎಲೆಮೆಂಟ್‌ಗಾಗಿ DOM ಮೆಥೆಡ್‌ಗಳನ್ನು, ಪ್ರಾಪರ್ಟೀಗಳನ್ನು ಮತ್ತು ಇವೆಂಟ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ಇದು ನಿಮಗೆ ವಿಡಿಯೊಗಳನ್ನು ಲೋಡ್ ಮಾಡಲು, ಪ್ರದರ್ಶಿಸಲು, ಮತ್ತು ಪೌಸ್ ಮಾಡಲು ಮತ್ತು ಇದರ ಜೊತೆಗೆ ಅವಧಿ ಮತ್ತು ವಾಲ್ಯೂಮನ್ನು ಸೆಟ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದರಲ್ಲಿ DOM ಪ್ರಾಪರ್ಟೀಗಳು ಸಹ ಇದ್ದು , ಇವು ವಿಡಿಯೋ ಪ್ಲೇ ಆಗಲು ಪ್ರರಂಭಿಸಿದ ಮೇಲೆ , ಪೌಸ್ ಆದ ಮೇಲೆ ಮುಂತಾದವುಗಳ ಬಗ್ಗೆ ನಿಮಗೆ ಸೂಚನೆ ನೀಡುತ್ತವೆ.

HTML5 ವಿಡಿಯೋ ಟ್ಯಾಗ್‌ಗಳು (HTML5 Video Tags)

ಟ್ಯಾಗ್	ವಿವರಣೆ
<video>	ವಿಡಿಯೋ ಅಥವಾ ಚಲನಚಿತ್ರವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<source>	ಮೀಡಿಯಾ ಎಲೆಮೆಂಟ್‌ಗಳಿಗಾಗಿ <video> ಮತ್ತು <audio> ನಂತಹ ಅನೇಕ ಮೀಡಿಯಾ ರೀಸೋರ್ಸ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.
<track>	ಮೀಡಿಯಾ ಪ್ಲೇಯರ್‌ಗಳಲ್ಲಿ ಟೆಕ್ಸ್ಟ ಟ್ರ್ಯಾಕ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

ವೆಬ್‌ನಲ್ಲಿ ಆಡಿಯೋ (Audio on the Web)

HTML5 <audio> ಎಲೆಮೆಂಟ್ ವೆಬ್ ಪುಟದಲ್ಲಿ ಆಡಿಯೊವನ್ನು ಸೇರಿಸಲು ಸರಿಯಾದ ವಿಧಾನವಾಗಿದೆ.

HTML <audio> ಎಲೆಮೆಂಟ್

HTML ನಲ್ಲಿ ಆಡಿಯೊ ಫೈಲನ್ನು ಪ್ಲೇ ಮಾಡಲು , <audio> ಎಲೆಮೆಂಟನ್ನು ಉಪಯೋಗಿಸಿ:

ಉದಾಹರಣೆ

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

HTML ಆಡಿಯೋ - ಅದು ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ (HTML Audio - How It Works)

ಕಂಟ್ರೋಲ್ ಅಟ್ರಿಬ್ಯೂಟ್, (controls attribute) play, pause,ಮತ್ತು volume ನಂತಹ ವಿಡಿಯೊ ಕಂಟ್ರೋಲ್‌ಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ. . The <source> ಎಲೆಮೆಂಟ್ ಬ್ರೌಸರ್‌ಗೆ ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಪರ್ಯಾಯ ಆಡಿಯೋ ಫೈಲ್‌ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಲು ನಿಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಬ್ರೌಸರ್ ಮೊದಲನೇಯ ಗುರುತಿಸಲ್ಪಟ್ಟ ಸ್ವರೂಪವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. <audio> ಎಲೆಮೆಂಟನ್ನು ಬೆಂಬಲಿಸದೇ ಇರುವ ಬ್ರೌಸರ್‌ಗಳಲ್ಲಿ <audio> ಮತ್ತು </audio> ಟ್ಯಾಗ್‌ಗಳ ಮಧ್ಯದಲ್ಲಿನ ಟೆಕ್ಸ್ಟ ಮಾತ್ರ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತದೆ.

HTML ಆಡಿಯೊ - ಮೀಡಿಯಾ ಪ್ರಕಾರಗಳು (HTML Audio - Media Types)

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML ಆಡಿಯೊ - ಮೆಥೆಡ್‌ಗಳು, ಪ್ರಾಪರ್ಟೀಗಳು, ಮತ್ತು ಇವೆಂಟ್‌ಗಳು (HTML Audio - Methods, Properties, and Events)

HTML5, <audio> ಎಲೆಮೆಂಟ್‌ಗಾಗಿ DOM ಮೆಥೆಡ್‌ಗಳನ್ನು, ಪ್ರಾಪರ್ಟೀಗಳನ್ನು ಮತ್ತು ಇವೆಂಟ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ. ಇದು ನಿಮಗೆ ಆಡಿಯೊಗಳನ್ನು ಲೋಡ್ ಮಾಡಲು, ಪ್ರದರ್ಶಿಸಲು, ಮತ್ತು ಪೌಸ್ ಮಾಡಲು ಮತ್ತು ಇದರ ಜೊತೆಗೆ ಅವಧಿ ಮತ್ತು ವಾಲ್ಯೂಮನ್ನು ಸೆಟ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದರಲ್ಲಿ DOM ಪ್ರಾಪರ್ಟೀಗಳು ಸಹ ಇದ್ದು , ಇವು ಆಡಿಯೊ ಪ್ಲೇ ಆಗಲು ಪ್ರರಂಭಿಸಿದ ಮೇಲೆ , ಪೌಸ್ ಆದ ಮೇಲೆ ಮುಂತಾದವುಗಳ ಬಗ್ಗೆ ನಿಮಗೆ ಸೂಚನೆ ನೀಡುತ್ತವೆ.

HTML5 ಆಡಿಯೊ ಟ್ಯಾಗ್‌ಗಳು (HTML5 Audio Tags)

ಟ್ಯಾಗ್	ವಿವರಣೆ
<audio>	ಧ್ವನಿ ವಿಷಯಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ
<source>	ಮೀಡಿಯಾ ಎಲೆಮೆಂಟ್‌ಗಾಗಿ, <video> ಮತ್ತು <audio>ನಂತಹ ಅನೇಕ ಮೀಡಿಯಾ ರೀಸೋರ್ಸ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬಳಸಿ ಡೈನಾಮಿಕ್ HTML ಪುಟಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು (Develop dynamic HTML pages using Javascript)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಫಂಕ್ಷನ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು
- ಫಂಕ್ಷನ್‌ನ ಕಾರ್ಯವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ಫಂಕ್ಷನ್ ಉಪಯೋಗಿಸುವ ಪ್ರಯೋಜನವನ್ನು ವಿವರಿಸಿ
- ವೇರಿಯೇಬಲ್‌ಗಳ ವ್ಯಾಪ್ತಿಯನ್ನು ವಿವರಿಸಿ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳು (JavaScript Functions):ಒಂದು ನಿರ್ದಿಷ್ಟವಾದ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾದ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ ಫಂಕ್ಷನ್‌ನ್ನು "ಏನಾದರೂ" ಆಹ್ವಾನಿಸಿದಾಗ (ಕರೆಯಲಾಗುತ್ತದೆ) ಅದನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 1

```
function myFunction(p1, p2) {
return p1 * p2; // ಫಂಕ್ಷನ್ p1 ಮತ್ತು p2 ನ ಗುಣಾಕಾರವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
}
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್ ಸಿಂಟ್ಯಾಕ್ಸ್

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್‌ನ್ನು function ಕೀವರ್ಡ್ ನಂತರ ಅದರ ಹೆಸರು , ನಂತರ ಆವರಣ () ದಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.

ಫಂಕ್ಷನ್ ಹೆಸರು ಅಕ್ಷರಗಳನ್ನು, ಸಂಖ್ಯೆಗಳನ್ನು , ಅಂಡರ್‌ಸ್ಕೋರ್ (_), ಮತ್ತು ಡಾಲರ್ ಚಿಹ್ನೆಗಳನ್ನು ($) ಒಳಗೊಂಡಿರಬಹುದು. (ವೇರಿಯೇಬಲ್‌ಗಳಿಗಿರುವಂತೆಯೆ ನಿಯಮಗಳು)

ಆವರಣದಲ್ಲಿ ಪ್ಯಾರಾಮೀಟರ್ ಹೆಸರುಗಳನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ ಪ್ರತ್ಯೇಕಿಸಲಾಗುತ್ತದೆ : (parameter1, parameter2, ...)

ಫಂಕ್ಷನ್‌ನಿಂದ ಕಾರ್ಯಗತಗೊಳಿಸಲ್ಪಡುವ ಕೋಡ್‌ನ್ನು ಪುಷ್ಪಾವರಣದಲ್ಲಿ ಇರಿಸಲಾಗುತ್ತದೆ : {}

```
functionName(parameter1, parameter2, parameter3) {
code to be executed
}
```

ಫಂಕ್ಷನ್ ಪ್ಯಾರಾಮೀಟರ್ ಎಂದರೆ ಫಂಕ್ಷನ್ ಡೆಫಿನಿಷನ್‌ನಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡಲಾದ ಹೆಸರುಗಳಾಗಿವೆ. ಫಂಕ್ಷನ್ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು ಎಂದರೆ ಒಂದು ಫಂಕ್ಷನ್‌ನ್ನು ಆಹ್ವಾನಿಸಿದಾಗ ಅದು ಸ್ವೀಕರಿಸಿದ ನಿಜವಾದ ಮೌಲ್ಯಗಳಾಗಿವೆ. ಫಂಕ್ಷನ್‌ನ ಒಳಗಡೆ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ. ಒಂದು ಫಂಕ್ಷನ್ ಹೆಚ್ಚಾಗಿ ಇತರ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳಲ್ಲಿರುವಂತೆ Procedure ಅಥವಾ Subroutine ನಂತೆಯೆ ಇರುತ್ತದೆ.

ಫಂಕ್ಷನ್ ಆಹ್ವಾನಿಸುವುದು (Function Invocation)

ಯಾವುದಾದರೂ ಅಂಶವು ಫಂಕ್ಷನ್‌ನ್ನು ಆಹ್ವಾನಿಸಿದಾಗ (calls) ಫಂಕ್ಷನ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಕಾರ್ಯಗತವಾಗುತ್ತದೆ.

- ಇವೆಂಟ್ ಸಂಭವಿಸಿದಾಗ (ಬಳಕೆದಾರರು ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ)
- ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೋಡ್ ಅದನ್ನು ಆಹ್ವಾನಿಸಿದಾಗ (called)
- ಸ್ವಯಂಚಾಲಿತವಾಗಿ (ಸ್ವಯಂ ಪ್ರೇರಿತವಾಗಿ)

ಫಂಕ್ಷನ್ ರಿಟರ್ನ್ (Function Return)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ರಿಟರ್ನ್ ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ತಲುಪಿದಾಗ, ಫಂಕ್ಷನ್ ಕೆಲಸ ಮಾಡುವುದನ್ನು ನಿಲ್ಲಿಸುತ್ತದೆ. ಫಂಕ್ಷನ್ ಯಾವುದಾದರೂ ಸ್ಟೇಟಮೆಂಟ್‌ನಿಂದ ಆಹ್ವಾನಿಸಲ್ಪಟ್ಟಿದ್ದರೆ, ಆವಾಹನೆಯ ಸ್ಟೇಟಮೆಂಟ್ ನಂತರ ಕೋಡನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ "return" ಆಗುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಫಂಕ್ಷನ್‌ಗಳು ರಿಟರ್ನ್ ಮೌಲ್ಯವನ್ನು ಲೆಕ್ಕ ಮಾಡುತ್ತವೆ. ರಿಟರ್ನ ಮೌಲ್ಯವನ್ನು "caller" ಗೆ ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ ("returned"):

ಉದಾಹರಣೆ 2

ಎರಡು ಸಂಖ್ಯೆಗಳ ಗುಣಾಕಾರವನ್ನು ಲೆಕ್ಕ ಮಾಡಿ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಹಿಂತಿರುಗಿಸಿ :

```
var x = myFunction(4, 3); // ಫಂಕ್ಷನ್‌ನ್ನು ಕರೆಯಲಾಗುತ್ತದೆ(call)
, return ಮೌಲ್ಯವು x ನಲ್ಲಿ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ.
```

© NIMI NOT TO BE REPUBLISHED

function myFunction(a, b) {

return a * b; // ಫಂಕ್ಷನ್ a ಮತ್ತು b ಯ ಉತ್ಪನ್ನವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

}

The result in x will be: 12

ಫಂಕ್ಷನ್‌ಗಳು ಏಕೆ ಬೇಕು ? Why Functions?

ನೀವು ಕೋಡ್ ಅನ್ನು ಮರುಬಳಕೆ ಮಾಡಬಹುದು : ಕೋಡ್ ಅನ್ನು ಒಂದು ಬಾರಿ ವ್ಯಾಖ್ಯಾನಿಸಿ ಮತ್ತು ಹಲವಾರು ಬಾರಿ ಅದನ್ನು ಬಳಸಿರಿ. ನೀವು ಒಂದೇ ಕೋಡನ್ನು ವಿಭಿನ್ನ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳೊಂದಿಗೆ ಹಲವಾರು ಬಾರಿ ಬಳಸಬಹುದು. ಇದು ವಿಭಿನ್ನ ಫಲಿತಾಂಶಗಳನ್ನು ನೀಡುತ್ತದೆ.

ಉದಾಹರಣೆ 3

Fahrenheit ನ್ನು Celsius ಗೆ ಪರಿವರ್ತಿಸಿ:

function toCelsius(fahrenheit) {

return (5/9) * (fahrenheit-32);

}

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ (JavaScript Functions are Objects)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಫಂಕ್ಷನ್‌ಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಫಂಕ್ಷನ್‌ಗಳು ಪ್ರಾಪರ್ಟಿಗಳು ಮತ್ತು ಮೆಥೆಡ್‌ಗಳನ್ನು ಹೊಂದಿವೆ. ನೀವು ನಿಮ್ಮದೇ ಆದ ಸ್ವಂತ ಪ್ರಾಪರ್ಟೀಗಳನ್ನು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳೂ ಸಹ ವೇರಿಯೇಬಲ್‌ಗಳಾಗಿವೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಫಂಕ್ಷನ್‌ಗಳನ್ನು ವೇರಿಯೇಬಲ್‌ಗಳನ್ನಾಗಿಯೂ ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 4

ಈ ಕೆಳಗಿನ ಕೋಡ್ ಬದಲಾಗಿ :

temp = toCelsius(32);

text = "The temperature is " + temp + "Centigrade";

ನೀವು ಇದನ್ನು ಉಪಯೋಗಿಸಬಹುದು :

text = "The temperature is" + toCelsius(32) + "Centigrade";

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ಫಂಕ್ಷನ್‌ಗಳನ್ನುಸಾಮಾನ್ಯವೇರಿಯೇಬಲ್‌ಗಳಂತೆಯೆ ಮರುವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಇತರ ಫಂಕ್ಷನ್‌ಗಳಿಗೆ ಮೌಲ್ಯಗಳನ್ನಾಗಿ ಪಾಸ್ ಮಾಡಬಹುದು.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ವ್ಯಾಪ್ತಿ (JavaScript Scope)

ಸ್ಕೋಪ್ (ವ್ಯಾಪ್ತಿ) ಎಂದರೆ, ನೀವು ಸ್ವೀಕರಿಸುವ ವೇರಿಯೇಬಲ್‌ಗಳ ಗುಂಪಾಗಿದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳು ಸಹ ವೇರಿಯೇಬಲ್‌ಗಳಾಗಿವೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ನೀವು ಸ್ವೀಕರಿಸುವ ಅಥವಾ ಪಡೆಯುವ ವೇರಿಯೇಬಲ್‌ಗಳ, ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳ ಗುಂಪಾಗಿದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್ ಸ್ಕೋಪ್ (ವ್ಯಾಪ್ತಿ) ಹೊಂದಿದೆ : ಫಂಕ್ಷನ್‌ಗಳ ಒಳಗೆ ಸ್ಕೋಪ್ ಬದಲಾಗುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು (Local JavaScript Variables)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಫಂಕ್ಷನ್‌ನ ಒಳಗೆ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾದ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಎನ್ನುತ್ತಾರೆ. ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಸ್ಥಳೀಯ ವ್ಯಾಪ್ತಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಅವುಗಳನ್ನು ಫಂಕ್ಷನ್‌ನ ಒಳಗೆ ಮಾತ್ರ ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 5

// ಇಲ್ಲಿರುವ ಕೋಡ್‌ಗೆ train ಬಳಸಲು ಬರುವುದಿಲ್ಲ.

function myFunction() {

var train = "Jan Satabdi";

// ಇಲ್ಲಿರುವ ಕೋಡ್‌ಗೆ train ಬಳಸಲು ಬರುತ್ತದೆ

}

ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಅವುಗಳ ಫಂಕ್ಷನ್ ಒಳಗಡೆ ಮಾತ್ರ ಗುರುತಿಸಲಾಗುವುದರಿಂದ ಒಂದೇ ಹೆಸರನ್ನು ಹೊಂದಿರುವ ವೆರಿಯೆಬಲ್‌ಗಳನ್ನು ವಿಭಿನ್ನ ಫಂಕ್ಷನ್‌ಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು. ಒಂದು ಫಂಕ್ಷನ್ ಪ್ರಾರಂಭವಾದಾಗ ಲೋಕಲ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ರಚಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಫಂಕ್ಷನ್ ಪೂರ್ಣಗೊಂಡಾಗ ಅವುಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲಾಗುತ್ತದೆ.

ಗ್ಲೋಬಲ್ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳು (Global JavaScript Variables)

ಒಂದು ಫಂಕ್ಷನ್‌ನ ಹೊರಗೆ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾದ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಗ್ಲೋಬಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಎನ್ನುತ್ತಾರೆ. ಗ್ಲೋಬಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು ಜಾಗತಿಕ ವ್ಯಾಪ್ತಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ವೆಬ್ ಪುಟದಲ್ಲಿನ ಎಲ್ಲ ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳು ಮತ್ತು ಫಂಕ್ಷನ್‌ಗಳು ಇವುಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 6

var train = " Jan Satabdi";

© NIMI NOT TO BE REPUBLISHED

// ಇಲ್ಲಿರುವ ಕೋಡ್ train ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

function myFunction() {

// ಇಲ್ಲಿರುವ ಕೋಡ್ train ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

}

ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಗ್ಲೋಬಲ್ ಆಗುವುದು

ಡಿಕ್ಲೇರ್ ಮಾಡದೇ ಇರುವ ವೇರಿಯೇಬಲ್‌ಗೆ ನೀವು ಒಂದು ಮೌಲ್ಯವನ್ನು ನಿಯೋಜಿಸಿದರೆ, ಅದು ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೆ ಗ್ಲೋಬಲ್ ವೇರಿಯೇಬಲ್ ಆಗುತ್ತದೆ.

ಉದಾಹರಣೆ 7

// ಇಲ್ಲಿರುವ ಕೋಡ್ train ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

function myFunction() {

train = "Jan Satabdi";

// ಇಲ್ಲಿರುವ ಕೋಡ್ train ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

}

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳ ಜೀವಿತಾವಧಿ (The Lifetime of JavaScript Variables)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳ ಜೀವಿತಾವಧಿಯು ಅದನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಿದಾಗಲೇ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಫಂಕ್ಷನ್ ಪೂರ್ಣಗೊಂಡಾಗ ಲೋಕಲ್ ವೇರಿಯೆಬಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲಾಗುತ್ತದೆ. ನೀವು ಪುಟವನ್ನು ಮುಚ್ಚಿದಾಗ ಗ್ಲೋಬಲ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲಾಗುತ್ತದೆ.

ಫಂಕ್ಷನ್ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು (Function Arguments)

ಫಂಕ್ಷನ್ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳು (ಪ್ಯಾರಾಮೀಟರ್‌ಗಳು) ಫಂಕ್ಷನ್‌ನ ಒಳಗಡೆ ಲೋಕಲ್ ವೇರಿಯೆಬಲ್‌ಗಳಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ.

HTML ನಲ್ಲಿ ಗ್ಲೊಬಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು (Global Variables in HTML)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಗ್ಲೊಬಲ್ ಸ್ಕೋಪ್ ಒಂದು ಸಂಪೂರ್ಣ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಪರಿಸರವಾಗಿದೆ.

HTML ನಲ್ಲಿ, ಗ್ಲೋಬಲ್ ಸ್ಕೋಪ್, ವಿಂಡೊ ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ : ಎಲ್ಲ ಗ್ಲೋಬಲ್ ವೇರಿಯೇಬಲ್‌ಗಳು ವಿಂಡೊ ಆಬ್ಜೆಕ್ಟ್‌ಗೆ ಸೇರಿವೆ.

ಉದಾಹರಣೆ 8

// ಇಲ್ಲಿರುವ ಕೋಡ್ window.train ನ್ನು ಬಳಸಬಹುದು.

function myFunction() {

train = "Jan Satabdi";

}

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಸ್ಟ್ರಿಂಗ್ ಮತ್ತು ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳು (String and number methods in JavaScript)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ವಿವರಿಸಿ**
- **ವಿವಿಧ ಸ್ಟ್ರಿಂಗ್ ಮೆಥೆಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳು (JavaScript Strings)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಸಂಗ್ರಹಿಸಲು ಮತ್ತು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಸೊನ್ನೆ ಅಥವಾ ಅದಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಅಕ್ಷರಗಳನ್ನು ಕೋಟ್(quote) ಗಳಲ್ಲಿ ಬರೆಯಲಾಗಿರುತ್ತದೆ. ನೀವು ಒಂದು ಅಥವಾ ಎರಡು ಕೋಟ್‌ಗಳನ್ನು ಬಳಸಬಹುದು.

ಉದಾಹರಣೆ 1

var cityname = "Chennai"; // Double quotes

var cityname = 'Chennai'; // Single quotes

ನೀವು ಸ್ಟ್ರಿಂಗ್‌ನ ಒಳಗೆ ಕೋಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು, ಆದರೆ ಅವು ಸ್ಟ್ರಿಂಗ್ ಸುತ್ತಲಿನ ಕೋಟ್‌ಗಳಿಗೆ ಹೊಂದಿಕೆಯಾಗಬಾರದು.

ಉದಾಹರಣೆ 2

var notes = "You're Welcome";

var ans = "Coimbatore is called 'Cotton City'";

var ans = 'Coimbatore is called "Cotton City"';

ಸ್ಟ್ರಿಂಗ್ ಲೆಂಥ್ ಪ್ರಾಪರ್ಟಿ (String Length property)

ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ಅಕ್ಷರಗಳನ್ನು ಕಂಡುಹಿಡಿಯಲು length ಎನ್ನುವ ಬಿಲ್ಟ್ - ಇನ್ ಪ್ರಾಪರ್ಟೀಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 3

var s= "Computer Operator and Programming Assistant";

var slen = s.length;

© NIMI NOT TO BE REPUBLISHED

ಬ್ಯಾಕ್‌ಸ್ಲ್ಯಾಶ್ ಎಸ್ಕೇಪ್ ಅಕ್ಷರ (Backslash escape character.)

ಬ್ಯಾಕ್‌ಸ್ಲ್ಯಾಶ್ (\) ಎಸ್ಕೇಪ್ ಅಕ್ಷರವು ವಿಶೇಷ ಅಕ್ಷರಗಳನ್ನು ಸ್ಟ್ರಿಂಗ್ ಅಕ್ಷರಗಳಿಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.

ಕೋಡ್	ಫಲಿತಾಂಶ	ವಿವರಣೆ
\’	‘	Single quote
\”	“	Double quote
\\	\	Backslash

\” ಈ ಸರಣಿಯು ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಡಬಲ್ ಕೋಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 4

var x = “Bhubaneswar is known as the \”Temple City of India\””;

\’ ಸರಣಿಯು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಸಿಂಗಲ್ ಕೋಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ.:

ಉದಾಹರಣೆ 5

var x = ‘you\’re Welcome.’;

\\ ಸರಣಿಯು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಬ್ಯಾಕ್‌ಸ್ಲ್ಯಾಶ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ.:

ಉದಾಹರಣೆ 6

var x = “The character \\ is called backslash.”;

ಸ್ಟ್ರಿಂಗ್‌ಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿರಬಹುದು (Strings Can be Objects)

ಸಾಮಾನ್ಯವಾಗಿ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್‌ಗಳು ಅಕ್ಷರಗಳಿಂದ ರಚಿತವಾದ ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯಗಳಾಗಿವೆ.

var tName = “Veni”;

new ಕೀವರ್ಡ್‌ನ್ನು ಬಳಸಿಕೊಂಡು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು.

var tName = new String(“Veni”);

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಟ್ರಿಂಗ್ ಮೆಥೆಡ್‌ಗಳು (JavaScript String Methods)

ಸ್ಟ್ರಿಂಗ್‌ನೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಲು ಸ್ಟ್ರಿಂಗ್ ಮೆಥೆಡ್‌ಗಳು ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಸ್ಟ್ರಿಂಗನ್ನು ಕಂಡುಹಿಡಿಯುವುದು (Finding a String in a String)

indexOf() ಮೆಥೆಡ್

indexOf() ಮೆಥೆಡ್ ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ಮೊದಲನೇಯ ಕಂಡುಬರುವ ನಿರ್ದಿಷ್ಟವಾದ ಟೆಕ್ಸ್ಟ್‌ನ index(ಸ್ಥಾನ)ವನ್ನು ತೋರಿಸುತ್ತದೆ. :

ಉದಾಹರಣೆ 7

var str = “When I do good I feel good ”;

var num = str.indexOf(“good”);

ಫಲಿತಾಂಶ

ವೇರಿಯೇಬಲ್ num ಸ್ಥಾನದ ಮೌಲ್ಯ 10 ಹೊಂದಿದೆ

> **ಸೂಚನೆ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ಥಾನವನ್ನು ಸೊನ್ನೆಯಿಂದ ಎಣಿಸಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ. ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಮೊದಲನೇ ಸ್ಥಾನ 0 ಆಗಿರುತ್ತದೆ, ಎರಡನೇ ಸ್ಥಾನ 1 ಆಗಿರುತ್ತದೆ, ಮೂರನೇ ಸ್ಥಾನ 2 ಆಗಿರುತ್ತದೆ....**

lastIndexOf() ಮೆಥೆಡ್

The lastIndexOf() ಮೆಥೆಡ್ ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ಕೊನೆಯ ಕಂಡುಬರುವ ನಿರ್ದಿಷ್ಟವಾದ ಟೆಕ್ಸ್ಟ್‌ನ index(ಸ್ಥಾನ)ವನ್ನು ತೋರಿಸುತ್ತದೆ. :ಉದಾಹರಣೆ 8

var str = “When I do good I feel good”;

var num = str.lastIndexOf(“good”);

ಫಲಿತಾಂಶ:

ವೇರಿಯೇಬಲ್ num ಸ್ಥಾನದ ಮೌಲ್ಯ 22 ಹೊಂದಿದೆ

ಒಂದು ವೇಳೆ ಟೆಕ್ಸ್ಟ್ ಕಂಡು ಬರದಿದ್ದರೆ, indexOf(), and lastIndexOf() ಎರಡೂ ಮೆಥೆಡ್‌ಗಳು -1 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 9

var str = “When I do good I feel good”;

var num = str.lastIndexOf(“better”);

ಫಲಿತಾಂಶ:

ವೇರಿಯೇಬಲ್ num ಸ್ಥಾನದ ಮೌಲ್ಯ -1 ಆಗಿದೆ.

ಎರಡೂ ಮೆಥೆಡ್‌ಗಳು ಹುಡುಕಾಟಕ್ಕಾಗಿ ಎರಡನೇಯ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಆರಂಭದ ಸ್ಥಾನವನ್ನಾಗಿ ಸ್ವೀಕರಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 10

var str = “When I do good I feel good”;

var num = str.indexOf(“good”,15);

ಫಲಿತಾಂಶ:

ವೇರಿಯೇಬಲ್ num ಸ್ಥಾನದ ಮೌಲ್ಯ 22 ಹೊಂದಿದೆ

ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಹುಡುಕುವುದು (Searching for a String in a String)

© NIMI NOT TO BE REPUBLISHED

search() ಮೆಥೆಡ್ ಒಂದು ಸ್ಟ್ರಿಂಗನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಮೌಲ್ಯಕ್ಕಾಗಿ ಹುಡುಕಾಡುತ್ತದೆ ಮತ್ತು ಹೊಂದಿಕೆ ಕಂಡುಬಂದ ಸ್ಥಾನವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 11

```
var str = "When I do good I feel good";
var num = str.search("good");
```

ಫಲಿತಾಂಶ:

ವೇರಿಯೇಬಲ್ **num** ಸ್ಥಾನದ ಮೌಲ್ಯ 10 ಹೊಂದಿದೆ

> **ಸೂಚನೆ : search() ಮೆಥೆಡ್ ಎರಡನೇ ಆರಂಭದ ಸ್ಥಾನದ ಆರ್ಗ್ಯುಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಕೊಳ್ಳುವುದಿಲ್ಲ.**

ಸ್ಟ್ರಿಂಗ್‌ನ ಭಾಗಗಳನ್ನು ಹೊರತೆಗೆಯುವುದು (Extracting String Parts)

slice() ಮೆಥೆಡ್

slice() ಸ್ಟ್ರಿಂಗ್‌ನ ಒಂದು ಭಾಗವನ್ನು ಹೊರತೆಗೆಯುತ್ತದೆ ಮತ್ತು ಹೊರತೆಗೆದ ಭಾಗವನ್ನು ಹೊಸ ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಈ ಮೆಥೆಡ್ ಆರಂಭದ ಸ್ಥಾನವನ್ನು ಮತ್ತು ಕೊನೆಯ ಸ್ಥಾನವನ್ನು ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ, ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿನ ಸ್ಥಾನ 14 ರಿಂದ ಸ್ಥಾನ 21 ರವರೆಗಿನ ಭಾಗವನ್ನು ಸ್ಲೈಸ್ (ತುಂಡು) ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 12

```
var str = "Hockey,Kabadi,Cricket";
var res = str.slice(14, 21);
```

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶ Cricket ಎಂದು ತೋರಿಸುತ್ತದೆ.

ಒಂದು ವೇಳೆ ಪ್ಯಾರಾಮೀಟರ್ ಋಣಾತ್ಮಕವಾಗಿದ್ದರೆ, ಆಗ ಸ್ಥಾನವನ್ನು ಸ್ಟ್ರಿಂಗ್‌ನ ಕೊನೆಯಿಂದ ಎಣಿಸಲಾಗುತ್ತದೆ. ಈ ಉದಾಹರಣೆಯಲ್ಲಿ ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿನ -14 ನೇ ಸ್ಥಾನದಿಂದ -8 ನೇ ಸ್ಥಾನದವರೆಗೆ ಸ್ಲೈಸ್ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 13

```
var str = "Hockey,Kabadi,Cricket";
var res = str.slice(-14, -8);
```

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶ Kabadi ಆಗಿರುತ್ತದೆ.

ನೀವು ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಬಿಟ್ಟುಬಿಟ್ಟರೆ, ಮೆಥೆಡ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಉಳಿದ ಎಲ್ಲ ಭಾಗವನ್ನು ಸ್ಲೈಸ್ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 14

```
var res = str.slice(7);
```

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶವು Kabadi,Cricket ಆಗಿರುತ್ತದೆ.

substring() ಮೆಥೆಡ್

substring() ಸಹ slice() ನಂತೆಯೆ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ವ್ಯತ್ಯಾಸವೆಂದರೆ ಅದು ಋಣಾತ್ಮಕ ಇಂಡೆಕ್ಸ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುವುದಿಲ್ಲ.

ಉದಾಹರಣೆ 15

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substring(7, 13);
```

ಫಲಿತಾಂಶ:

res ಫಲಿತಾಂಶ Kabadi ಆಗಿರುತ್ತದೆ.

substr() ಮೆಥೆಡ್

ನೀವು ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಬಿಟ್ಟುಬಿಟ್ಟರೆ, substring() ಮೆಥೆಡ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಉಳಿದ ಎಲ್ಲ ಭಾಗವನ್ನು ಸ್ಲೈಸ್ ಮಾಡುತ್ತದೆ.

substr() ಮೆಥೆಡ್ slice() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ. ವ್ಯತ್ಯಾಸವೆಂದರೆ ಇದರಲ್ಲಿ ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್ ಹೊರತೆಗೆಯಲಾದ ಭಾಗದ ಲೆಂಥ್ (ಉದ್ದ) ನ್ನು ತಿಳಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 16

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substr(7, 6);
```

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶ Kabadi ಆಗಿರುತ್ತದೆ

ನೀವು ಎರಡನೇ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಬಿಟ್ಟುಬಿಟ್ಟರೆ, ssubstr() ಮೆಥೆಡ್ ಸ್ಟ್ರಿಂಗ್‌ನ ಉಳಿದ ಎಲ್ಲ ಭಾಗವನ್ನು ಸ್ಲೈಸ್ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ 17

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substr(7);
```

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶ Kabadi,Cricket ಆಗಿರುತ್ತದೆ.

ಮೊದಲನೇ ಪ್ಯಾರಾಮೀಟರ್ ಋಣಾತ್ಮಕವಾಗಿದ್ದರೆ, ಸ್ಟ್ರಿಂಗ್‌ನ ಕೊನೆಯ ಭಾಗದಿಂದ ಸ್ಥಾನವನ್ನು ಎಣಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 18

var str = "Hockey,Kabadi,Cricket";

var res = str.substr(-7);

ಫಲಿತಾಂಶ:

res ನ ಫಲಿತಾಂಶ Cricket ಆಗಿರುತ್ತದೆ.

ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿಯ ವಿಷಯವನ್ನು ಬದಲಾಯಿಸುವುದು (Replacing String Content)

replace() ಮೆಥೆಡ್, ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಇನ್ನೊಂದು ಮೌಲ್ಯದಿಂದ ಬದಲಾಯಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 19

str = "I like custard apple";

var newstr = str.replace("custard apple", "mango");

ಫಲಿತಾಂಶ:

newstr ನ ಫಲಿತಾಂಶವು I like mango ಆಗಿರುತ್ತದೆ.

> **ಸೂಚನೆ : replace() ಮೆಥೆಡ್, ಅದು ಕರೆದಿರುವ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಬದಲಾಯಿಸುವುದಿಲ್ಲ. ಅದು ಹೊಸ ಸ್ಟ್ರಿಂಗನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಡೀಫಾಲ್ಟ್ ಆಗಿ replace() ಫಂಕ್ಷನ್ ಮೊದಲನೇ ಹೊಂದಿಕೆಯಾದ ಮೌಲ್ಯವನ್ನು ಮಾತ್ರ ಬದಲಾಯಿಸುತ್ತದೆ.**

ಡಿಫಾಲ್ಟ್ ಆಗಿ replace() ಫಂಕ್ಷನ್ ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿದೆ. CUSTARD APPLE (with upper-case) ಎಂದು ಬರೆದರೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದಿಲ್ಲ. ಕೇಸ್ ಇನ್‌ಸೆನ್ಸಿಟಿವ್‌ನ್ನು ಬದಲಾಯಿಸಲು, regular expression ನ್ನು /i flag (insensitive) ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ 20

str = "I like custard apple";

var newstr = str.replace(/CUSTARD APPLE/i, "Mango");

ಫಲಿತಾಂಶ:

newstr ನ ಫಲಿತಾಂಶ I like Mango ಆಗಿರುತ್ತದೆ.

regular expressions ಗಳನ್ನು ಕೋಟ್‌ಗಳಿಲ್ಲದೇ ಬರೆಯಲಾಗಿದೆ ಎಂಬುದನ್ನು ಗಮನಿಸಿ..

ಎಲ್ಲ ಹೊಂದಿಕೆಯಾದವುಗಳನ್ನು ಬದಲಾಯಿಸಲು, regular expression ನ್ನು /g flag (global match).ನೊಂದಿಗೆ ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ 21

str = "I like custard apple and apple";

var newstr = str.replace(/apple/g, "Mango");

ಫಲಿತಾಂಶ:

newstr ನ ಫಲಿತಾಂಶ I like custard Mango and Mango ಆಗಿರುತ್ತದೆ.

Upper ಮತ್ತು Lower Case ಗೆ ಬದಲಾಯಿಸುವುದು.

toUpperCase() ಬಳಸಿ ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು uppercase ಗೆ ಪರಿವರ್ತಿಸಲಗುತ್ತದೆ.

ಉದಾಹರಣೆ 22

var str1 = "Information Technology";

var str2 = str1.toUpperCase();

ಫಲಿತಾಂಶ:

str2 ಫಲಿತಾಂಶ INFORMATION TECHNOLOGY ಆಗಿರುತ್ತದೆ.

to LowerCase() ಬಳಸಿ ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು lowercase ಗೆ ಪರಿವರ್ತಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 23

var str1 = "INFORMATION TECHNOLOGY";

var str2 = str1.toLowerCase();

ಫಲಿತಾಂಶ:

str2 ನ ಫಲಿತಾಂಶ information technology ಆಗಿರುತ್ತದೆ.

concat() ಮೆಥೆಡ್

concat() ಎರಡು ಅಥವಾ ಎರಡಕ್ಕಿಂತ ಹೆಚ್ಚು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು .ಜೋಡಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 24

var txt1 = "Mr";

var txt2 = "Selvaraj";

var txt3 = txt1.concat(".", txt2);

ಫಲಿತಾಂಶ:

txt3 ನ ಫಲಿತಾಂಶವು Mr.Selvaraj ಆಗಿರುತ್ತದೆ.

ಆಪರೇಟರ್ ಬದಲಾಗಿ concat() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಕೆಳಗಿನ ಎರಡೂ ಸಾಲುಗಳು ಒಂದೇ ಕೆಲಸವನ್ನು ಮಾಡುತ್ತವೆ.

ಉದಾಹರಣೆ 25

var txt = "Mr" + "." + "Selvaraj";

var txt = "Mr".concat(".", "Selvaraj");

© NIMI NOT TO BE REPUBLISHED

String.trim()

String.trim() ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ ಎರಡೂ ಬದಿಯಲ್ಲಿರುವ ಖಾಲಿ ಜಾಗವನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

ಉದಾಹರಣೆ 26

```
var str = “        India Gate         ”;
alert(str.trim());
```

ಫಲಿತಾಂಶ:

ಮುಂದೆ ಮತ್ತು ಹಿಂದೆ ಇರುವ ಖಾಲಿ ಸ್ಥಳವಿಲ್ಲದೆ ಇರುವ India Gate ನ್ನು alert box ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

ಸ್ಟ್ರಿಂಗ್‌ನಿಂದ ಅಕ್ಷರಗಳನ್ನು ಹೊರತೆಗೆಯುವುದು(Extracting String Characters)

charAt() ಮೆಥೆಡ್, ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಿಂದ ನಿರ್ದಿಷ್ಟವಾದ ಅಕ್ಷರವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 27

```
var str = “Thirumalai Nayakkar Mahal”;
str.charAt(0);   // T ಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
```

charCodeAt() ಮೆಥೆಡ್, ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟ ಇಂಡೆಕ್ಸ್‌ನಲ್ಲಿರುವ ಅಕ್ಷರದ ಯುನಿಕೋಡನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಈ ಮೆಥೆಡ್ 0 ಮತ್ತು 65535 ನ ಮಧ್ಯದಲ್ಲಿನ UTF-16 cone ಪೂರ್ಣಾಂಕವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 28

```
var str = “Hill Station”;
str.charCodeAt(0); //  72 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
```

ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಅರೇಗೆ ಪರಿವರ್ತಿಸುವುದು (Converting a String to an Array)

split() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಅರೇಗೆ ಪರಿವರ್ತಿಸಬಹುದು.

ಉದಾಹರಣೆ 29

```
var txt1 = “Kovai,Nellai,Madurai”; // String
var txt2 = txt1.split(“,”); // ಅಲ್ಪವಿರಾಮದಿಂದ ವಿಭಜಿಸುತ್ತದೆ.
```

ಫಲಿತಾಂಶ:

txt2 ನ ಫಲಿತಾಂಶ Kovai ಆಗಿರುತ್ತದೆ.

ವಿಭಜಕ(seperator) ವನ್ನು ಬಿಟ್ಟುಬಿಟ್ಟರೆ, ಹಿಂತಿರುಗಿಸಲ್ಪಟ್ಟ ಅರೇ ಸಂಪೂರ್ಣ ಸ್ಟ್ರಿಂಗ್ index [0] ನಲ್ಲಿರುತ್ತದೆ. ವಿಭಜಕವು “” ಆಗಿದ್ದರೆ, ಹಿಂತಿರುಗಿಸಲ್ಪಟ್ಟ ಅರೇ ಒಂದೇ ಅಕ್ಷರವನ್ನು ಹೊಂದಿರುವ ಆರೆ ಆಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 30

```
var txt = “Anaimai”; // String
txt.split(“”); // ಅಕ್ಷರಗಳು ವಿಭಜಿಸಲ್ಪಡುತ್ತವೆ.
```

ಉದಾಹರಣೆ 31

```
<!DOCTYPE html>
<html>
<body>
<p id=“demo”></p>
<script>
var str = “Temple”;
var arr = str.split(“”);
var text = “”;
var i;
for (i = 0; i < arr.length; i++) {
text += arr[i] + “<br>”
}
document.getElementById(“demo”).innerHTML = text;
</script>
</body>
</html>
```

ಫಲಿತಾಂಶ:

T

e

m

p

l

e

?

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳು (JavaScript Number Methods)

ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳು ಸಂಖ್ಯೆಗಳೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಲು ನಿಮಗೆ ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳು ಮತ್ತು ಪ್ರಾಪರ್ಟಿಗಳು (Number Methods and Properties)

2018 or 1.44 ನಂತಹ ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯಗಳು, ಪ್ರಾಪರ್ಟಿಗಳನ್ನಾಗಲೀ ಮತ್ತು ಮೆಥೆಡ್‌ಗಳನ್ನಾಗಲೀ ಹೊಂದಿರುವುದಿಲ್ಲ ಏಕೆಂದರೆ ಅವು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಲ್ಲ.

ಆದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಪ್ರಿಮಿಟಿವ್ ಮೌಲ್ಯಗಳಿಗೆ ಮೆಥೆಡ್‌ಗಳು ಮತ್ತು ಪ್ರಾಪರ್ಟಿಗಳು ಸಹ ಲಭ್ಯವಿರುತ್ತವೆ, ಏಕೆಂದರೆ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೆಥೆಡ್‌ಗಳನ್ನು ಮತ್ತು ಪ್ರಾಪರ್ಟಿಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುವಾಗ ಪ್ರಿಮಿಟೀವ್ ಮೌಲ್ಯಗಳನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳೆಂದು ಪರಿಗಣಿಸುತ್ತದೆ.

toString() ಮೆಥೆಡ್

toString() ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಸ್ಟ್ರಿಂಗ್ ಆಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಎಲ್ಲ ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳು ಯಾವುದೇ ರೀತಿಯ ಸಂಖ್ಯೆಗಳ ಜೊತೆಗೆ ಉಪಯೋಗಿಸಬಹುದು (ಲಿಟರಲ್‌ಗಳು, ವೇರಿಯೇಬಲ್‌ಗಳು, ಅಥವಾ ಎಕ್ಸ್‌ಪ್ರೆಶನ್‌ಗಳು)

ಉದಾಹರಣೆ 32

var n = 2018;

n.toString(); // ವೇರಿಯೇಬಲ್ n ನಿಂದ 2018 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

(2018).toString(); // ಲಿಟರಲ್ 2018 ರಿಂದ 2018 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

(2000+18).toString(); // ಎಕ್ಸ್‌ಪ್ರೆಶನ್ 2000 + 18 ರಿಂದ 2018 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

toExponential() ಮೆಥೆಡ್

toExponential() ಮೆಥೆಡ್ ಸ್ಟ್ರಿಂಗನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಇದು ಸಂಖ್ಯೆಯನ್ನು ಪೂರ್ಣಾಂಕವನ್ನಾಗಿಸುತ್ತದೆ ಮತ್ತು ಘಾತೀಯ ಸಂಕೇತವನ್ನು ಬಳಸಿ ಬರೆಯಲಾಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ 33

var x = 3. 869;

x.toExponential(2); // 3.87e+0 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toExponential(4); // 3.8690e+0 ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toExponential(6); // 3.869000e+0 ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಇದರಲ್ಲಿ ಪ್ಯಾರಾಮೀಟರ್ ಐಚ್ಚಿಕವಾಗಿರುತ್ತದೆ. ನೀವು ಅದನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸದಿದ್ದರೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸಂಖ್ಯೆಯನ್ನು ಪೂರ್ಣಾಂಕಗೊಳಿಸುವುದಿಲ್ಲ.

to Fixed() ಮೆಥೆಡ್

toFixed() ಇದು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ದಶಮಾಂಶ ಸಂಖ್ಯೆಯ, ಸಂಖ್ಯೆಯೊಂದಿಗೆ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ, .

ಉದಾಹರಣೆ 34

var x = 3. 869;

x.toFixed(0); // 4 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toFixed(2); // 3.87 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toFixed(4); // 3.8690 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toFixed(6); // 3.869000 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸೂಚನೆ : toFixed(2) ಇದು ಹಣಕ್ಕೆ ಸಂಬಂಧಿಸಿದ ಕೆಲಸ ಮಾಡಲು ಸರಿಯಾಗಿದೆ.

toPrecision() ಮೆಥೆಡ್

toPrecision() ನಿರ್ದಿಷ್ಟವಾದ ಲೆಂಥ್(ಉದ್ದ) ನೊಂದಿಗೆ ಬರೆಯಲ್ಪಟ್ಟ ಸಂಖ್ಯೆಯೊಂದಿಗಿನ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 35

var x = 3. 869;

x.toPrecision(); // 3. 869 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toPrecision(2); // 3.9 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toPrecision(4); // 3. 869 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

x.toPrecision(6); // 3. 86900 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

valueOf() ಮೆಥೆಡ್

valueOf() ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 36

var x = 451;

x.valueOf(); // ವೇರಿಯೇಬಲ್ x ನಿಂದ 451 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

(451).valueOf(); // ಲಿಟರಲ್ 451 ನಿಂದ 451 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

(400 + 51).valueOf(); // ಎಕ್ಸ್‌ಪ್ರೆಶನ್ 400 + 51 ನಿಂದ 451 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಸಂಖ್ಯೆಗಳಿಗೆ ಪರಿವರ್ತಿಸುವುದು (Converting Variables to Numbers)

ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಸಂಖ್ಯೆಗಳಿಗೆ ಪರಿವರ್ತಿಸಲು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ 3 ವಿಧಾನಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ:

1 Number() ಮೆಥೆಡ್

2 parseInt() ಮೆಥೆಡ್

3 parseFloat() ಮೆಥೆಡ್

ಈ ಮೆಥೆಡ್‌ಗಳು ನಂಬರ್ ಮೆಥೆಡ್‌ಗಳಲ್ಲ, ಆದರೆ ಗ್ಲೋಬಲ್ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೆಥೆಡ್‌ಗಳಾಗಿವೆ.

ಗ್ಲೋಬಲ್ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮೆಥೆಡ್‌ಗಳು (Global JavaScript Methods)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಗ್ಲೋಬಲ್ ಮೆಥೆಡ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಎಲ್ಲ ಡಾಟಾಟೈಪ್‌ಗಳ ಮೇಲೆ ಬಳಸಬಹುದು. ಸಂಖ್ಯೆಗಳೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಲು ಇವುಗಳು ಅತ್ಯಂತ ಸೂಕ್ತವಾದ ಮೆಥೆಡ್‌ಗಳಾಗಿವೆ.

ಮೆಥೆಡ್	ವಿವರಣೆ
Number()	ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ, ಅದು ತನ್ನ ಆರ್ಗ್ಯುಮೆಂಟ್‌ನಿಂದ ಪರಿವರ್ತಿಸಲ್ಪಟ್ಟಿರುತ್ತದೆ.
parseInt()	ತನ್ನ ಆರ್ಗ್ಯುಮೆಂಟನ್ನು ಪಾರ್ಸ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಪೂರ್ಣಾಂಕವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
parseFloat()	ತನ್ನ ಆರ್ಗ್ಯುಮೆಂಟನ್ನು ಪಾರ್ಸ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಫ್ಲೋಟಿಂಗ್ ಪಾಯಿಂಟ್ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number() ಮೆಥೆಡ್

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಸಂಖ್ಯೆಗಳನ್ನಾಗಿ ಪರಿವರ್ತಿಸಲು Number() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 37

Number(true); // 1 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number(false); // 0 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("25"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number(" 25"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("25 "); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number(" 25 "); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("25.77"); // 25.77 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("25,77"); // NaN ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("2577"); // NaN ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Number("ITI"); // NaN ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

> **ಸೂಚನೆ :** ಒಂದು ವೇಳೆ ಸಂಖ್ಯೆಯನ್ನು ಪರಿವರ್ತಿಸಲು ಆಗದಿದ್ದರೆ, NaN(Not a Number) ಎಂದು ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.

Number() ಮೆಥೆಡ್‌ನ್ನು ದಿನಾಂಕಗಳ ಮೇಲೆ ಬಳಸಬಹುದು. Number() ದಿನಾಂಕವನ್ನು ಒಂದು ಸಂಖ್ಯೆಗೆ ಪರಿವರ್ತಿಸಬಲ್ಲದು.

ಉದಾಹರಣೆ 38

Number(new Date("2018-09-15")); // 1536969600000 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

> **ಸೂಚನೆ :** ಮೇಲಿನ Number() ಮೆಥೆಡ್ 1.1.1970 ರಿಂದ ಇಲ್ಲಿಯವರೆಗಿನ ಮಿಲಿಸೆಕೆಂಡ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseInt() ಮೆಥೆಡ್

parseInt() ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಪಾರ್ಸ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಪೂರ್ಣ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಖಾಲಿ ಸ್ಥಳಗಳಿಗೆ ಅನುಮತಿ ನೀಡಲಾಗಿದೆ. ಮೊದಲನೇ ಸಂಖ್ಯೆಯನ್ನು ಮಾತ್ರ ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 39

parseInt("25"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseInt("25.33"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseInt("25 20 30"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseInt("25 years"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseInt("years 25"); // NaN ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಂಖ್ಯೆಯನ್ನು ಪರಿವರ್ತಿಸಲು ಸಾಧ್ಯವಾಗದಿದ್ದರೆ, NaN (Not a Number) ಎಂದು ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.

parseFloat() ಮೆಥೆಡ್

parseFloat()ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಪಾರ್ಸ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಖಾಲಿ ಸ್ಥಳಗಳಿಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ . ಮೊದಲನೇ ಸಂಖ್ಯೆಯನ್ನು ಮಾತ್ರ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 40

parseFloat("25"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseFloat("25.77"); // 25.77 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseFloat("255075"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseFloat("25 years"); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

parseFloat("years 25"); // NaN ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಂಖ್ಯೆಯನ್ನು ಪರಿವರ್ತಿಸಲು ಸಾಧ್ಯವಾಗದಿದ್ದರೆ, NaN (Not a Number) ಎಂದು ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.

?

Number Properties

ಪ್ರಾಪರ್ಟಿ	ವಿವರಣೆ
MIN_VALUE	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿರುವ ಅತ್ಯಂತ ಚಿಕ್ಕ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
MAX_VALUE	ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿರುವ ಅತ್ಯಂತ ದೊಡ್ಡ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
POSITIVE_ INFINITY	ಅನಂತತೆಯನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. (returned on overflow)
NEGATIVE_ INFINITY	ಋಣಾತ್ಮಕ ಅನಂತತೆಯನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. (returned on overflow)
NaN	"Not-a-Number" ಮೌಲ್ಯವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ MIN_VALUE ಮತ್ತು MAX_VALUE

ಉದಾಹರಣೆ 41

var n = Number.MAX_VALUE;

ಫಲಿತಾಂಶ :

MAX_VALUE ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಸಾಧ್ಯವಿರುವ ಅತ್ಯಂತ ದೊಡ್ಡ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

1.7976931348623157e+308

ಉದಾಹರಣೆ 42

var n = Number.MIN_VALUE;

ಫಲಿತಾಂಶ :

MIN_VALUE ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ಸಾಧ್ಯವಿರುವ ಅತ್ಯಂತ ಚಿಕ್ಕ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

5e-324

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ POSITIVE_INFINITY

ಉದಾಹರಣೆ 43

var n = Number.POSITIVE_INFINITY;

ಓವರ್‌ಫ್ಲೋ ಆದಮೇಲೆ POSITIVE_INFINITY ಹಿಂತಿರುಗಿಸಲ್ಪಡುತ್ತದೆ.

ಉದಾಹರಣೆ 44

var n = 2 / 0;

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ NEGATIVE_INFINITY

ಉದಾಹರಣೆ 45

var n = Number.NEGATIVE_INFINITY;

ಓವರ್ ಫ್ಲೋ ಆದ ಮೇಲೆ NEGATIVE_INFINITY ಹಿಂತಿರುಗಿಸಲ್ಪಡುತ್ತದೆ.

ಉದಾಹರಣೆ 46

var x = -1 / 0;

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ NaN - Not a Number

ಉದಾಹರಣೆ 47

var x = Number.NaN;

NaN ಇದು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಒಂದು ಕಾಯ್ದಿರಿಸಿದ(reserved) ಶಬ್ದವಾಗಿದೆ. ಈ ಸಂಖ್ಯೆ ಲೀಗಲ್ ಸಂಖ್ಯೆಯಲ್ಲ ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತದೆ.ಸಂಖ್ಯೆಯಲ್ಲದ ಸ್ಟ್ರಿಂಗ್‌ನ ಜೊತೆಗೆ ಅಂಕಗಣಿತವನ್ನು ಮಾಡಲು ಪ್ರಯತ್ನಿಸಿದರೆ ಅದು NaN (Not a Number) ಎನ್ನುವ ಫಲಿತಾಂಶಕ್ಕೆ ಕಾರಣವಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 48

var n = 500 / "Price"; // n ಎಂಬುದು NaN (Not a Number) ಆಗಿರುತ್ತದೆ

ನಂಬರ್ ಪ್ರಾಪರ್ಟಿಗಳನ್ನು ವೇರಿಯೆಬಲ್‌ಗಳ ಮೇಲೆ ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ.(Number Properties Cannot be Used on Variables)

ನಂಬರ್ ಪ್ರಾಪರ್ಟಿಗಳು, Number ಎಂದು ಕರೆಯಲ್ಪಡುವ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ನಂಬರ್ ಆಬ್ಜೆಕ್ಟ್ ರ‍್ಯಾಪರ್‌ಗೆ (wrapper) ಸೇರಿವೆ. ಈ ಪ್ರಾಪರ್ಟಿಗಳನ್ನು Number.MAX_VALUE ಎಂದು ಮಾತ್ರ ಪಡೆದುಕೊಳ್ಳಬಹುದು.

newNumber.MAX_VALUE ನ್ನು ಬಳಸುವುದರಿಂದ, undefined ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ. ಇಲ್ಲಿ 'newNumber' ಎಂದರೆ ವೇರಿಯೇಬಲ್, ಎಕ್ಸ್‌ಪ್ರೆಷನ್, ಅಥವಾ ಮೌಲ್ಯ.

ಉದಾಹರಣೆ 49

var a = 10;

var b = a.MAX_VALUE; // b, undefined ಆಗಿರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ Math ಆಬ್ಜೆಕ್ಟ್‌ಗಳು (Math objects in JavaScript)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ Math ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ Math ಆಬ್ಜೆಕ್ಟ್‌ಗಳು (JavaScript Math Object)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Math ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ನಿಮಗೆ ಸಂಖ್ಯೆಗಳ ಮೇಲೆ ಗಣಿತದ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತವೆ.

Math.PI; // 3.141592653589793 ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.round(x) , x ನ ಮೌಲ್ಯವನ್ನು ಅದರ ಅತ್ಯಂತ ಸಮೀಪದ ಪೂರ್ಣಾಂಕವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 1

Math.round(5.8); // 6 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.round(5.4); // 5 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.pow(x, y) x ಘಾತಾಂಕ y ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.:

ಉದಾಹರಣೆ 2

Math.pow(5, 2); // 25 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.sqrt(x) x ನ ವರ್ಗಮೂಲವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 3

Math.sqrt(25); // 5 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.abs(x) x ನ ಸಂಪೂರ್ಣ (absolute) ಧನಾತ್ಮಕ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 4

Math.abs(-3.5); // 3.5 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.ceil(x) x ನ ಮೌಲ್ಯವನ್ನು ಅದರ ಸಮೀಪದ ಪೂರ್ಣಾಂಕಕ್ಕೆ ಪೂರ್ತಿಗೊಳಿಸಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 5

Math.ceil(5.4); // 6 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.floor(x) x ನ ಮೌಲ್ಯವನ್ನು ಅದರ ಸಮೀಪದ ಹಿಂದಿನ ಪೂರ್ಣಾಂಕಕ್ಕೆ ಪೂರ್ತಿಗೊಳಿಸಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 6

Math.floor(5.8); // 5 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.sin(x) ಇದು angle x (given in radians) ನ sine (-1 ಮತ್ತು 1 ರ ನಡುವಿನ ಮೌಲ್ಯ) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ..

ನೀವು ರೇಡಿಯನ್‌ಗಳ ಬದಲಾಗಿ ಡಿಗ್ರಿಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸಿದರೆ, ನೀವು ಡಿಗ್ರಿಯನ್ನು ರೇಡಿಯನ್‌ಗೆ ಪರಿವರ್ತಿಸಬೇಕು :

Angle in radians = Angle in degrees x PI / 180.

ಉದಾಹರಣೆ 7

Math.sin(90 * Math.PI / 180); // 1 (the sine of 90 degrees) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.sin(0 * Math.PI / 180); // 0 (the sine of 0 degrees) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.cos(x) angle x (given in radians) ನ cosine (a value between -1 and 1) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ನೀವು ಡಿಗ್ರಿಗಳ ಬದಲಾಗಿ ರೇಡಿಯನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸಿದರೆ, ನೀವು ಡಿಗ್ರಿಯನ್ನು ರೇಡಿಯನ್‌ಗೆ ಪರಿವರ್ತಿಸಬೇಕು :

Angle in radians = Angle in degrees x PI / 180.

ಉದಾಹರಣೆ 8

Math.cos(0 * Math.PI / 180); // 1 (the cos of 0 degrees) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.min() and Math.max() ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಆರ್ಗ್ಯುಮೆಂಟ್‌ಗಳ ಪಟ್ಟಿಯಲ್ಲಿ ಕನಿಷ್ಟ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯಗಳನ್ನು ಕಂಡುಹಿಡಿಯಬಹುದು.

ಉದಾಹರಣೆ 9

Math.min(20,40,6,0,-10); // -10 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 10

Math.max(20,40,6,0,-10); // 40 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.random() ಇದು 0 (ಒಳಗೊಂಡಿರುವ), and 1 (ಒಳಗೊಂಡಿರದ) ರ ನಡುವಿನ ಸಂಪೂರ್ಣ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ :

ಉದಾಹರಣೆ 11

Math.random(); // random ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math ಪ್ರಾಪರ್ಟಿಗಳು (Math Properties)(Constants)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್, Math ಆಬ್ಜೆಕ್ಟ್‌ನೊಂದಿಗೆ ಉಪಯೋಗಿಸಲ್ಪಡುವ 8 ಗಣಿತದ ಸ್ಥಿರಾಂಕಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ :

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 12

Math.E // Euler ನ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.PI // PI ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.SQRT2 // 2 ರ ವರ್ಗಮೂಲವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.SQRT1_2 // square root of 1/2 ವರ್ಗಮೂಲವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.LN2 // 2 ರ ನೈಸರ್ಗಿಕ ಲೊಗ್ಯಾರಿದಮ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.LN10 // 10 ರ ನೈಸರ್ಗಿಕ ಲೊಗ್ಯಾರಿದಮ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.LOG2E // base 2 logarithm of E ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math.LOG10E // base 10 logarithm of E ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Math ಕನ್ಸ್ಟ್ರಕ್ಟರ್ (Math Constructor)

ಇತರ ಗ್ಲೋಬಲ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಂತಲ್ಲದೇ, Math ಆಬ್ಜೆಕ್ಟ್ ಕನ್‌ಸ್ಟ್ರಕ್ಟರ್‌ನ್ನು ಹೊಂದಿಲ್ಲ. ಮೆಥೆಡ್ ಮತ್ತು ಪ್ರಾಪರ್ಟಿಗಳು static (ಸ್ಥಿರ) ಆಗಿರುತ್ತವೆ. Math ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಮೊದಲೇ ರಚಿಸದೆಯೇ ಎಲ್ಲ ಮೆಥೆಡ್‌ಗಳು ಮತ್ತು ಪ್ರಾಪರ್ಟಿಗಳನ್ನು (constants) ಉಪಯೋಗಿಸಬಹುದು.

Math ಆಬ್ಜೆಕ್ಟ್ ಮೆಥೆಡ್‌ಗಳು (Math Object Methods)

ಮೆಥೆಡ್	ವಿವರಣೆ
abs(x)	x ಯಾದೃಚ್ಛಿಕ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
acos(x)	of x ನ arccosine ನ್ನು , radian ಗಳಲ್ಲಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
asin(x)	x ನ arcsine ನ್ನು , radian ಗಳಲ್ಲಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
atan(x)	of x arctangent ನ್ನು -PI/2 and PI/2 ಗಳ ನಡುವಿನ ಸಂಖ್ಯೆಯನ್ನಾಗಿ radian ಗಳಲ್ಲಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
atan2(y, x)	Returns the arctangent of the quotient of its arguments ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
ceil(x)	Returns the value of x rounded up to its nearest integer ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
cos(x)	Returns the cosine of x (x is in radians) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
exp(x)	Returns the value of Ex ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
floor(x)	Returns the value of x rounded down to its nearest integer ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
log(x)	Returns the natural logarithm (base E) of x
max(x, y, z, ..., n)	ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
min(x, y, z, ..., n)	ಕನಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
pow(x, y)	x ನ ಘಾತಾಂಕ y ನ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
random()	0 ಮತ್ತು 1 ರ ನಡುವಿನ ಯಾದೃಚ್ಛಿಕ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
round(x)	x ನ ಮೌಲ್ಯವನ್ನು ಅದರ ಅತ್ಯಂತ ಸಮೀಪದ ಪೂರ್ಣಾಂಕಕ್ಕೆ ಪೂರ್ತಿಗೊಳಿಸಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
sin(x)	x ನ sine ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ (x radians ಗಳಲ್ಲಿರುತ್ತದೆ)
sqrt(x)	x ನ ವರ್ಗಮೂಲವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
tan(x)	ಒಂದು ಕೋನದ tangent ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ದಿನಾಂಕಗಳು (JavaScript Dates)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್** Date **ಆಬ್ಜೆಕ್ಟಗಳನ್ನು ವಿವರಿಸಿ**
- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್** Date **ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ**
- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್** Date get **ಮೆಥೆಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ**
- **ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್** Date set **ಮೆಥೆಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Date **ಆಬ್ಜೆಕ್ಟಗಳು** (JavaScript Date Objects)

ಡೀಫಾಲ್ಟ್ ಆಗಿ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬ್ರೌಸರ್‌ನ ಟೈಮ್ ಝೋನ್ ಬಳಸುತ್ತದೆ ಮತ್ತು ದಿನಾಂಕವನ್ನು ಸಂಪೂರ್ಣ ಟೆಕ್ಸ್ಟ್ ಸ್ಟ್ರಿಂಗ್ ಆಗಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

Thu Sep 27 2018 09:09:39 GMT+0530 (India Standard Time)

NOT TO BE REPUBLISHED

Date ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ರಚಿಸುವುದು (Creating Date Objects)

new Date() constructor ಉಪಯೋಗಿಸಿ Date ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ರಚಿಸಲಾಗುತ್ತದೆ.

new date ಆಬ್ಜೆಕ್ಟನ್ನು ರಚಿಸಲು 4 ಮಾರ್ಗಗಳಿವೆ. ಅವು ಯಾವವೆಂದರೆ:

1 new Date()

2 new Date(year, month, day, hours, minutes, seconds, milliseconds)

3 new Date(date string)

4 new Date(milliseconds)

1 new Date()

new Date()

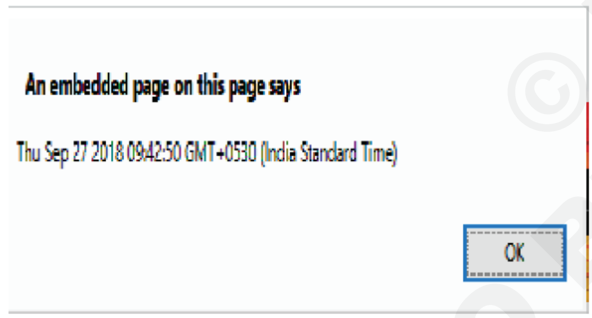
ಉದಾಹರಣೆ 1:

var d = new Date();

alert(d);

ಫಲಿತಾಂಶ : (Fig 1)

Fig 1



ಸೂಚನೆ : Date ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಸ್ಟ್ಯಾಟಿಕ್ (ಸ್ಥಿರ) ಆಗಿರುತ್ತವೆ. ಕಂಪ್ಯೂಟರ್ ಸಮಯವು ಟಿಕಿಂಗ್ ಆಗಿರುತ್ತದೆ ಆದರೆ Date ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಅಲ್ಲ.

2 new Date(year, month, ...)

new Date(year, month, ...) ಇದು ನಿರ್ದಿಷ್ಟವಾದ ದಿನಾಂಕ ಮತ್ತು ಸಮಯದೊಂದಿಗೆ ಹೊಸ date ಆಬ್ಜೆಕ್ಟನ್ನು ರಚಿಸುತ್ತದೆ.

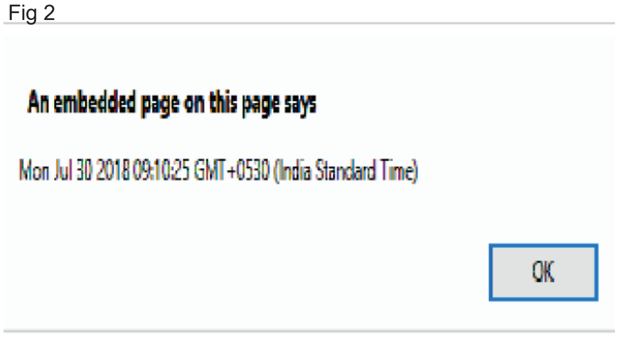
ಉದಾಹರಣೆ 2

var d = new Date(2018, 06, 30, 09, 10, 25, 0);

7 ಸಂಖ್ಯೆಗಳು year, month, day, hour, minute, second, ಮತ್ತು millisecond (ಇದೇ ಕ್ರಮದಲ್ಲಿ) ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ.

ಫಲಿತಾಂಶ :(Fig 2)

Fig 2



ಸೂಚನೆ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ತಿಂಗಳುಗಳನ್ನು 0 ದಿಂದ 11 ರವರೆಗೆ ಎಣಿಸುತ್ತದೆ. ಜನೇವರಿ 0 ಆಗುತ್ತದೆ. ಡಿಸೆಂಬರ್ 11 ಆಗುತ್ತದೆ.

6 ಸಂಖ್ಯೆಗಳು year, month, day, hour, minute, second. ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 3

var d = new Date(2018, 06, 30, 09, 10, 25);

5 ಸಂಖ್ಯೆಗಳು year, month, day, hour, and minute. ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 4

var d = new Date(2018, 06, 30, 09, 10);

ಉದಾಹರಣೆ 5

var d = new Date(2018, 06, 30, 09);

3 ಸಂಖ್ಯೆಗಳು year, month, and day. ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 6

var d = new Date(2018, 06, 30);

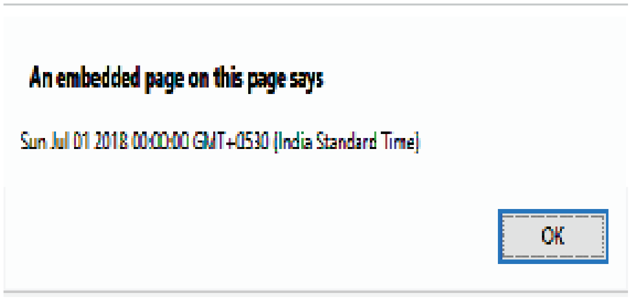
2 ಸಂಖ್ಯೆಗಳು year and month ಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ 7

var d = new Date(2018, 06);

ಫಲಿತಾಂಶ : (Fig 3)

Fig 3



© NIMI NOT TO BE REPUBLISHED

ನೀವು ತಿಂಗಳು (month) ನ್ನು ಬಿಟ್ಟುಬಿಡಲು ಸಾಧ್ಯವಿಲ್ಲ. ನೀವು ಕೇವಲ ಒಂದು ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಪೂರೈಸಿದರೆ, ಅದನ್ನು ಮಿಲಿಸೆಕೆಂಡು ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 8

var d = new Date(2018);

Previous Century

ಒಂದು ಮತ್ತು ಎರಡು ಅಂಕಿಯ ವರ್ಷಗಳನ್ನು 19XX ಎಂದು ಅರ್ಥೈಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 9

var d = new Date(96, 04, 12);

3 new Date(dateString)

new Date(dateString) dateString ನಿಂದ ಹೊಸ date ಆಬ್ಜೆಕ್ಟನ್ನು ರಚಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 10

var d = new Date("December 20, 2018 10:15:00");

ಫಲಿತಾಂಶ :(Fig 4)

Fig 4

ಸೂಚನೆ : ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ದಿನಾಂಕಗಳನ್ನು January 01, 1970, 00:00:00 UTC (Universal Time Coordinated) **ರಿಂದ ಮಿಲಿಸೆಕೆಂಡುಗಳ ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ.** .
Zero time ಎಂದರೆ January 01, 1970 00:00:00 UTC.
ಈಗ ಸಮಯ : 1537962903199 milliseconds past January 01,1970

4 new Date(milliseconds)

new Date(milliseconds) ಇದು ಹೊಸ date ಆಬ್ಜೆಕ್ಟನ್ನು zero time + ಮಿಲಿಸೆಕೆಂಡುಗಳು ಎಂದು ರಚಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 11

var d = new Date(0);

01 January 1970 + 100 000 000 milliseconds ಎಂದರೆ ಸುಮಾರು 02 January 1970.

ಉದಾಹರಣೆ 12

var d = new Date(100000000);

ಫಲಿತಾಂಶ : (Fig 5)

January 01 1970 minus 100 000 000 milliseconds ಎಂದರೆ ಸುಮಾರು December 31 1969.

Fig 5

ಉದಾಹರಣೆ 13

var d = new Date(-100000000);

ಫಲಿತಾಂಶ : (Fig 6)

Fig 6

ಉದಾಹರಣೆ 14

var d = new Date(86400000);

ಫಲಿತಾಂಶ :

Fri Jan 02 1970 05:30:00 GMT+0530 (India Standard Time)

ಸೂಚನೆ:new Date(milliseconds) **ಉಪಯೋಗಿಸುವುದರಿಂದ, ದು ಹೊಸ** date **ಆಬ್ಜೆಕ್ಟನ್ನು** January 1, 1970, 00:00:00 Universal Time (UTC) plus the milliseconds **ಎಂದು ರಚಿಸುತ್ತದೆ. ಒಂದು ದಿನ** (24 **ಗಂಟೆಗಳು**) **ಎಂದರೆ** 86 400 000 **ಮಿಲಿಸೆಕೆಂಡುಗಳು**

Date ಮೆಥೆಡ್‌ಗಳು (Date Methods)

Date ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸಿದಾಗ, ಹಲವಾರು ಮೆಥೆಡ್‌ಗಳು ನಿಮಗೆ ಅವುಗಳ ಮೇಲೆ ಕೆಲಸ ಮಾಡಲು ಅನುಮತಿಸುತ್ತವೆ. Date ಮೆಥೆಡ್‌ಗಳು ನಿಮಗೆ year, month, day, hour, minute, second, ಮತ್ತು millisecond ಗಳನ್ನು get ಮತ್ತು set ಮಾಡಲು ಅನುಮತಿ

© NIMI NOT TO BE REPUBLISHED

ನೀಡುತ್ತವೆ. ಸ್ಥಳಿಯ ಸಮಯ ಅಥವಾ UTC (universal, or GMT) ಸಮಯವನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ದಿನಾಂಕವನ್ನು ಪ್ರದರ್ಶಿಸುವುದು (Displaying Dates)

ಡೀಫಾಲ್ಟ್ ಆಗಿ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ದಿನಾಂಕವನ್ನು ಸಂಪೂರ್ಣ ಟೆಕ್ಸ್ಟ ಸ್ಟ್ರಿಂಗ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ನೀವು date ಆಬ್ಜೆಕ್ಟನ್ನು HTML ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಿದಾಗ, ಅದು toString() ಮೆಥೆಡ್‌ನೊಂದಿಗೆ ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೇ ಸ್ಟ್ರಿಂಗ್ ಆಗಿ ಪರಿವರ್ತನೆಯಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 15

```
d = new Date();

alert(d);
```

ಅದರಂತೆ :

```
d = new Date();

alert(d.toString());
```

toUTCString() ಮೆಥೆಡ್ ದಿನಾಂಕವನ್ನು UTC string ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 16

```
var d = new Date();

alert(d);
```

ಫಲಿತಾಂಶ : (Fig 7)

toDateString() ಮೆಥೆಡ್ ದಿನಾಂಕವನ್ನು ಹೆಚ್ಚು ಓದಬಲ್ಲ ಸ್ವರೂಪಕ್ಕೆ ಬದಲಾಯಿಸುತ್ತದೆ.

Fig 7

ಉದಾಹರಣೆ 17

```
var d = new Date();

alert(d.toDateString());
```

ಫಲಿತಾಂಶ : (Fig 8)

Fig 8

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Date ಫಾರ್ಮ್ಯಾಟ್‌ಗಳು (JavaScript Date Formats)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Date ಇನ್ಪುಟ್

ಸಾಮಾನ್ಯವಾಗಿ 3 ರೀತಿಯ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Date ಇನ್ಪುಟ್‌ಗಳು ಇವೆ.

ವಿಧ	ಉದಾಹರಣೆ
ISO Date	"2002-06-30" (The International Standard)
Short Date	"06/30/2002"
Long Date	"Jun 30 2002" ಅಥವಾ "30 Jun 2002"

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Date ಔಟ್‌ಪುಟ್

ಇವು ಇನ್ಪುಟ ಫಾರ್ಮ್ಯಾಟ್‌ನಿಂದ ಸ್ವತಂತ್ರ್ಯವಾಗಿವೆ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ದಿನಾಂಕಗಳನ್ನು ಸಂಪೂರ್ಣ ಸ್ಟ್ರಿಂಗ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ತೋರಿಸುತ್ತದೆ.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ISO Dates

ISO 8601ಇದು ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ಪ್ರತಿನಿಧಿಸುವ ಅಂತರರಾಷ್ಟ್ರೀಯ ಮಾನದಂಡವಾಗಿದೆ. ISO 8601 ಇದರ ಸಿಂಟ್ಯಾಕ್ಸ್ (YYYY-MM-DD) ಸಹ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ ಆದ್ಯತೆಯ date ಫಾರ್ಮ್ಯಾಟ್ ಆಗಿದೆ.

ಉದಾಹರಣೆ 18 (ಸಂಪೂರ್ಣ ದಿನಾಂಕ)

```
var d = new Date("2002-06-30");
```

ಸೂಚನೆ : ಲೆಕ್ಕಾಚಾರ ಮಾಡಲಾದ ದಿನಾಂಕವು ನಿಮ್ಮ ಸಮಯ ವಲಯ(time zone) ಕ್ಕೆ ಸಂಬಂಧಿಸಿರುತ್ತದೆ. ನಿಮ್ಮ ಸಮಯ ವಲಯದ ಆಧಾರದ ಮೇಲೆ ಮೇಲಿನ ಫಲಿತಾಂಶವು June 29 ಮತ್ತು June 30 ರ ನಡುವೆ ಬದಲಾಗುತ್ತದೆ.

ISO Dates (Year and Month)

ISO ದಿನಾಂಕಗಳನ್ನು ದಿನವನ್ನು (day) ತಿಳಿಸದೇ ಬರೆಯಬಹುದು (YYYYMM).

ಉದಾಹರಣೆ 19

```
var d = new Date("2002-06");
```

ಫಲಿತಾಂಶ :

Sat Jun 01 2002 05:30:00 GMT+0530 (India Standard Time)

ISO Dates (Only Year)

ISO ದಿನಾಂಕಗಳನ್ನು ತಿಂಗಳು ಮತ್ತು ದಿನ ಇಲ್ಲದೇ ಬರೆಯಬಹುದು (YYYY).

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 20

var d = new Date("2011");

ಫಲಿತಾಂಶ :

Sat Jan 01 2011 05:30:00 GMT+0530 (India Standard Time)

ISO Dates (Date-Time)

ISO ದಿನಾಂಕಗಳನ್ನು hours, minutes, ಮತ್ತು seconds (YYYY-MM-DDTHH:MM:SSZ) ಗಳೊಂದಿಗೆ ಬರೆಯಬಹುದು.

ಉದಾಹರಣೆ 21

var d = new Date("2011-12-20T12:00:00Z");

ಫಲಿತಾಂಶ :

Tue Dec 20 2011 17:30:00 GMT+0530 (India Standard Time)

> **ಸೂಚನೆ : ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು capital T ನಿಂದ ಪ್ರತ್ಯೇಕಿಸಲಾಗುತ್ತದೆ. UTC ಸಮಯವನ್ನು capital ಅಕ್ಷರ Z ನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ Short Dates.

Short dates ಗಳನ್ನು "MM/DD/YYYY" ಸಿಂಟ್ಯಾಕ್ಸ್ ಈ ರೀತಿಯಾಗಿದೆ.

ಉದಾಹರಣೆ 22

var d = new Date("06/30/2002");

ಫಲಿತಾಂಶ :

Sun Jun 30 2002 00:00:00 GMT+0530 (India Standard Time)

> **ಸೂಚನೆ : ಕೆಲವು ಬ್ರೌಸರ್‌ಗಳಲ್ಲಿ ಸೊನ್ನೆಯಿಂದ ಆರಂಭವಾಗದ ತಿಂಗಳು ಅಥವಾ ದಿನಗಳು ದೋಷವನ್ನು(error) ಉಂಟುಮಾಡಬಹುದು. "YYYY/MM/DD" ನ ನಡುವಳಿಕೆಯು undefined ಆಗಿರುತ್ತದೆ. ಕೆಲವು ಬ್ರೌಸರ್‌ಗಳು ಫಾರ್ಮ್ಯಾಟನ್ನು ಊಹಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತವೆ. ಕೆಲವು NaN. ಎಂದು ಉತ್ತರಿಸುತ್ತವೆ. "DD-MM-YYYY" ನ ನಡುವಳಿಕೆಯೂ ಸಹ undefined ಆಗಿದೆ.**

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Long Dates

Long dates ಗಳನ್ನು ಹೆಚ್ಚಾಗಿ "MMM DD YYYY" ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಸಿಂಟ್ಯಾಕ್ಸ್ ಹೀಗಿದೆ:

ಉದಾಹರಣೆ 23

var d = new Date("Aug 31 2012");

ತಿಂಗಳು ಮತ್ತು ದಿನ ಯಾವುದೇ ಕ್ರಮದಲ್ಲಿರಬಹುದು.

ಉದಾಹರಣೆ 24

var d = new Date("31 Aug 2012");

ಮತ್ತು ತಿಂಗಳನ್ನು ಪೂರ್ತಿಯಾಗಿ ಬರೆಯಬೇಕು (January), ಅಥವಾ ಸಂಕ್ಷಿಪ್ತವಾಗಿ ಬರೆಯಬಹುದು (Jan).

ಉದಾಹರಣೆ 25

var d = new Date("August 31 2012");

var d = new Date("AUGUST 31 2012");

> **ಸೂಚನೆ : ಅಲ್ಪವಿರಾಮವನ್ನು ನಿರ್ಲಕ್ಷಿಸಲಾಗಿದೆ, ಹೆಸರುಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿವೆ.**

Date ಇನ್ಪುಟ್ - ದಿನಾಂಕಗಳನ್ನು Parse ಮಾಡುವುದು

ನೀವು ಮಾನ್ಯವಾದ date ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ, ಅದನ್ನು ಮಿಲಿಸೆಕೆಂಡಿಗೆ ಪರಿವರ್ತಿಸಲು Date.parse() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು. Date.parse() ದಿನಾಂಕ ಮತ್ತು January 1, 1970. ರ ನಡುವಿನ ಮಿಲಿಸೆಕೆಂಡುಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 26

var msec = Date.parse("Sep 15, 1996");

document.getElementById("demo").innerHTML = msec;

ಫಲಿತಾಂಶ :

842725800000

ಮಿಲಿಸೆಕೆಂಡುಗಳ ಸಂಖ್ಯೆಯನ್ನು ನೀವು date ಆಬ್ಜೆಕ್ಟ್‌ಗೆ ಪರಿವರ್ತಿಸಲು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 27

var msec = Date.parse("Sep 15, 1996");

var d = new Date(msec);

document.getElementById("demo").innerHTML = d;

ಫಲಿತಾಂಶ :

Sun Sep 15 1996 00:00:00 GMT+0530 (India Standard Time)

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Get Date ಮೆಥೆಡ್‌ಗಳು (JavaScript Get Date Methods)

ಈ ಮೆಥೆಡ್‌ಗಳನ್ನು date ಆಬ್ಜೆಕ್ಟ್‌ನಿಂದ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದಕ್ಕಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಮೆಥೆಡ್	ವಿವರಣೆ
getFullYear()	ವರ್ಷವನ್ನು ನಾಲ್ಕು ಅಂಕಿಗಳ ಸಂಖ್ಯೆಯಾಗಿ ಪಡೆಯಿರಿ (yyyy)
getMonth()	ತಿಂಗಳನ್ನು ಸಂಖ್ಯೆಯಾಗಿ ಪಡೆಯಿರಿ (0-11)
getDate()	ದಿನವನ್ನು ಸಂಖ್ಯೆಯಾಗಿ ಪಡೆಯಿರಿ (1-31)
getHours()	ಗಂಟೆಯನ್ನು ಪಡೆಯಿರಿ (0-23)
getMinutes()	ನಿಮಿಷವನ್ನು ಪಡೆಯಿರಿ (0-59)
getSeconds()	ಸೆಕೆಂಡನ್ನು ಪಡೆಯಿರಿ. (0-59)
getMilliseconds()	ಮಿಲಿಸೆಕೆಂಡನ್ನು ಪಡೆಯಿರಿ (0-999)
getTime()	ಸಮಯವನ್ನು ಪಡೆಯಿರಿ (January 1, 1970 ರಿಂದ ಮಿಲಿಸೆಕೆಂಡುಗಳು)
getDay()	ವಾರದ ದಿನವನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಪಡೆಯಿರಿ (0-6)
Date.now()	ಸಮಯವನ್ನು ಪಡೆಯಿರಿ. ECMAScript 5.

getTime() ಮೆಥೆಡ್

getTime() ಮೆಥೆಡ್, January 1, 1970. ರಿಂದ ಮಿಲಿಸೆಕೆಂಡುಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 28

```
var d = new Date();

alert(d.getTime());
```

ಫಲಿತಾಂಶ : (Fig 9)

Fig 9

getFullYear() ಮೆಥೆಡ್

getFullYear() ಮೆಥೆಡ್, ದಿನಾಂಕದಲ್ಲಿನ ವರ್ಷವನ್ನು ನಾಲ್ಕು ಅಂಕೆಗಳ ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 29

```
var d = new Date();

alert(d.getFullYear());
```

ಫಲಿತಾಂಶ : (Fig 10)

Fig 10

getMonth() ಮೆಥೆಡ್

getMonth() ಮೆಥೆಡ್, ದಿನಾಂಕದಲ್ಲಿನ ತಿಂಗಳನ್ನು ಒಂದು ಸಂಖ್ಯೆಯನ್ನಾಗಿ (0-11). ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 30

```
var d = new Date();

document.getElementById("demo").innerHTML=
d.getMonth();
```

ಫಲಿತಾಂಶ : (Fig 11)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ಮೊದಲನೇಯ ತಿಂಗಳಿನ (January) ತಿಂಗಳ ಸಂಖ್ಯೆ 0 ಆಗಿರುತ್ತದೆ, ಆದ್ದರಿಂದ ಡಿಸೆಂಬರ್ ತಿಂಗಳ ಸಂಖ್ಯೆಯನ್ನು 11 ಎಂದು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ನೀವು ಹೆಸರುಗಳ ಒಂದು ಅರೇ ಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಮತ್ತು ತಿಂಗಳನ್ನು ಒಂದು ಹೆಸರನ್ನಾಗಿ ನೋಡಲು getMonth() ಉಪಯೋಗಿಸಿ.

ಉದಾಹರಣೆ 31

Fig 11

```
var d = new Date();

var months = ["January", "February", "March", "April",
"May", "June", "July", "August", "September", "Octo-
ber",
"November", "December"];
```

© NIMI NOT TO BE REPUBLISHED

document.getElementById("demo").innerHTML = months[d.getMonth()];

ಫಲಿತಾಂಶ : (Fig 12)

getDate() ಮೆಥೆಡ್

getDate() ಮೆಥೆಡ್, ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ದಿನವನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ (1-31).

Fig 12

ಉದಾಹರಣೆ 32

var d = new Date();

alert(d.getDate());

ಫಲಿತಾಂಶ : (Fig 13)

getHours() ಮೆಥೆಡ್

getHours() ಮೆಥೆಡ್ ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ಗಂಟೆಗಳನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ. (0-23).

Fig 13

ಉದಾಹರಣೆ 33

var d = new Date();

alert(d.getHours());

ಫಲಿತಾಂಶ : (Fig 14)

getMinutes() **ಮೆಥೆಡ್**

getMinutes() ಮೆಥೆಡ್ ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ನಿಮಿಷಗಳನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ.(0-59).

ಉದಾಹರಣೆ 34

var d = new Date();

alert(d.getMinutes());

ಫಲಿತಾಂಶ : (Fig 15)

getSeconds() **ಮೆಥೆಡ್**

getSeconds() ಮೆಥೆಡ್ ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ಸೆಕೆಂಡುಗಳನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ (0-59).

Fig 14

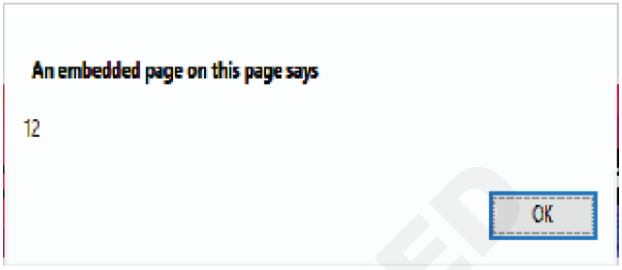


Fig 15

ಉದಾಹರಣೆ 35

var d = new Date();

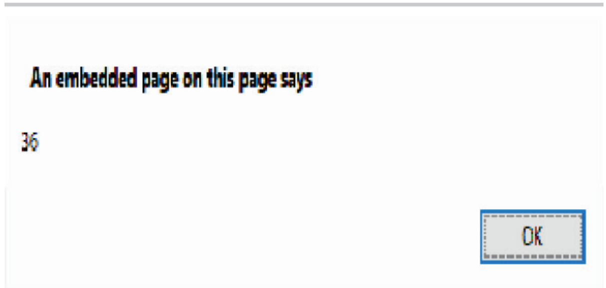
alert(d.getSeconds());

ಫಲಿತಾಂಶ : (Fig 16)

getMilliseconds() **ಮೆಥೆಡ್**

getMilliseconds() ಮೆಥೆಡ್ ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ಮಿಲಿ ಸೆಕೆಂಡುಗಳನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ (0-999).

Fig 16



ಉದಾಹರಣೆ 36

var d = new Date();

alert(d.getMilliseconds());

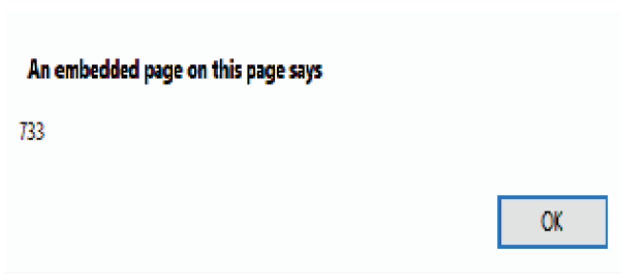
© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ : (Fig 17)

getDay() ಮೆಥೆಡ್

The getDay() ಮೆಥೆಡ್ ಒಂದು ದಿನಾಂಕದಲ್ಲಿರುವ ವಾರದ ದಿನಗಳನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹಿಂತಿರುಗಿಸುತ್ತದೆ (0-6).

Fig 17



ಉದಾಹರಣೆ 37

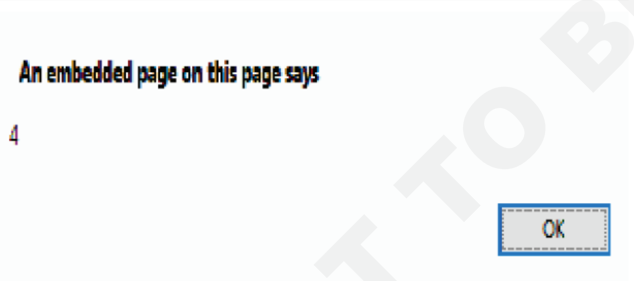
```
var d = new Date();
alert(d.getDay());
```

ಫಲಿತಾಂಶ : (Fig 18)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, ವಾರದ ಮೊದಲನೇ ದಿನ (0) ಎಂದರೆ Sunday ಆಗಿರುತ್ತದೆ, ವಿಶ್ವದ ಕೆಲವು ದೇಶಗಳು ವಾರದ ಮೊದಲನೇ ದಿನವನ್ನು Monday ಎಂದು ಪರಿಗಣಿಸುತ್ತವೆ.

ನೀವು ಹೆಸರುಗಳ ಅರೇ ಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು, ಮತ್ತು ವಾರದ ದಿನವನ್ನು ಹೆಸರಿನ ರೂಪದಲ್ಲಿ ನೋಡಲು getDay() ಉಪಯೋಗಿಸಿ.

Fig 18

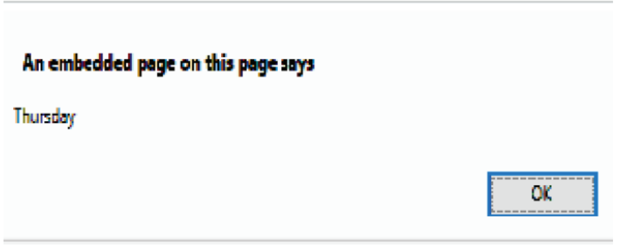


ಉದಾಹರಣೆ 38

```
var d = new Date();
var days = ["Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"];
alert(days[d.getDay()]);
```

ಫಲಿತಾಂಶ : (Fig 19)

Fig 19



UTC Date ಮೆಥೆಡ್‌ಗಳು (UTC Date Methods)

UTC date ಮೆಥೆಡ್‌ಗಳನ್ನು UTC ದಿನಾಂಕಗಳೊಂದಿಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. (Universal Time Zone dates).

Method	Description
getUTCDate()	getDate() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ UTC ದಿನಾಂಕವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCDay()	getDay() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ದಿನವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCFullYear()	getFullYear() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ವರ್ಷವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCHours()	getHours() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ಗಂಟೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCMilliseconds()	getMilliseconds() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ಮಿಲಿ ಸೆಕೆಂಡುಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCMinutes()	getMinutes() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ನಿಮಿಷಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCMonth()	getMonth() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ತಿಂಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
getUTCSeconds()	getSeconds() ನಂತೆಯೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಆದರೆ, UTC ಸೆಕೆಂಡುಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

©NIMI NOT TO BE REPUBLISHED

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ Set Date ಮೆಥೆಡ್‌ಗಳು

Set Date ಮೆಥೆಡ್, ಒಂದು Date ಆಬ್ಜೆಕ್ಟ್‌ಗಾಗಿ ದಿನಾಂಕದ ಮೌಲ್ಯಗಳನ್ನು (years, months, days, hours, minutes, seconds, milliseconds) ಹೊಂದಿಸಲು (Set) ನಿಮಗೆ ಅನುಮತಿಸುತ್ತವೆ.

Set Date Methods

ದಿನಾಂಕದ ಒಂದು ಭಾಗವನ್ನು ಹೊಂದಿಸಲು (set) Set Date ಮೆಥೆಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಮೆಥೆಡ್	ವಿವರಣೆ
setDate()	ದಿನವನ್ನು ಸಂಖ್ಯೆಯನ್ನಾಗಿ ಹೊಂದಿಸುತ್ತದೆ (1-31)
setFullYear()	ವರ್ಷವನ್ನು ಹೊಂದಿಸುತ್ತದೆ (ತಿಂಗಳು ಮತ್ತು ದಿನ ಐಚ್ಛಿಕವಾಗಿವೆ)
setHours()	ಗಂಟೆಯನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-23)
setMilliseconds()	ಮಿಲಿಸೆಕೆಂಡುಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-999)
setMinutes()	ನಿಮಿಷಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-59)
setMonth()	ತಿಂಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-11)
setSeconds()	ಸೆಕೆಂಡುಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-59)
setTime()	ಸಮಯವನ್ನು ಹೊಂದಿಸುತ್ತದೆ (January 1, 1970 ರಿಂದ ಮಿಲಿಸೆಕೆಂಡುಗಳನ್ನು)

setFullYear() ಮೆಥೆಡ್

setFullYear() ಒಂದು Date ಆಬ್ಜೆಕ್ಟ್‌ನ ವರ್ಷವನ್ನು ಹೊಂದಿಸುತ್ತದೆ.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ 2020.ಕ್ಕೆ ಹೊಂದಿಸುತ್ತದೆ.

Example 39

```
<script>
var d = new Date();
d.setFullYear(2020);
alert(d);
</script>
```

ಫಲಿತಾಂಶ : (Fig 20)

Fig 20

> **ಸೂಚನೆ :** setFullYear() ಮೆಥೆಡ್ ಐಚ್ಛಿಕವಾಗಿ ತಿಂಗಳು ಮತ್ತು ದಿನವನ್ನು ಹೊಂದಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 40

```
<script>
var d = new Date();
d.setFullYear(2018, 10, 2);
alert(d);
<script>
```

ಫಲಿತಾಂಶ : (Fig 21)

Fig 21

> **ಸೂಚನೆ :** ತಿಂಗಳಿನ ಎಣಿಕೆ 0 ಯಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಡಿಸೆಂಬರ್ ತಿಂಗಳು 11 ಆಗಿದೆ.

setMonth() ಮೆಥೆಡ್

setMonth() ಮೆಥೆಡ್, ಒಂದು Date ಆಬ್ಜೆಕ್ಟ್‌ನ ತಿಂಗಳನ್ನು ಹೊಂದಿಸಲಾಗುತ್ತದೆ (0-11).

ಉದಾಹರಣೆ 41

```
<script>
var d = new Date();
d.setMonth(2);
alert(d);
</script>
```

ಫಲಿತಾಂಶ : (Fig 22)

setDate() ಮೆಥೆಡ್

setDate() ಮೆಥೆಡ್, ಇದು Date ಆಬ್ಜೆಕ್ಟ್‌ನ ದಿನವನ್ನು ಹೊಂದಿಸುತ್ತದೆ (1-31).

Example 42

```
<script>
var d = new Date();
```

NOT TO BE REPUBLISHED © NIMI

```
d.setDate(18);
```

Fig 22

```
alert(d);
</script>
```

ಫಲಿತಾಂಶ : (Fig 23)

setDate() ಮೆಥೆಡ್‌ನ್ನು ದಿನಾಂಕಕ್ಕೆ ದಿನಗಳನ್ನು ಸೇರಿಸಲು ಸಹ ಉಪಯೋಗಿಸಬಹುದು.

Fig 23

ಉದಾಹರಣೆ 43

```
<script>
var d = new Date();
d.setDate(d.getDate() + 25);
alert(d);
</script>
```

ಫಲಿತಾಂಶ : (Fig 24)

ದಿನಗಳನ್ನು ಸೇರಿಸುವುದರಿಂದ ತಿಂಗಳು ಅಥವಾ ವರ್ಷಗಳು .ಬದಲಾದರೆ, ಈ ಬದಲಾವಣೆಗಳನ್ನು Date ಆಬ್ಜೆಕ್ಟ್ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ನಿರ್ವಹಿಸುತ್ತದೆ.

Fig 24

setHours() ಮೆಥೆಡ್

setHours() ಮೆಥೆಡ್, ಒಂದು date ಆಬ್ಜೆಕ್ಟ್‌ನ ಗಂಟೆಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-23).

ಉದಾಹರಣೆ 44

```
<script>
var d = new Date();
d.setHours(20);
alert(d);
</script>
```

ಫಲಿತಾಂಶ :(Fig 25)

setMinutes() ಮೆಥೆಡ್

setMinutes() ಮೆಥೆಡ್, ಒಂದು Date ಆಬ್ಜೆಕ್ಟ್‌ನ ನಿಮಿಷಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-59).

Fig 25

ಉದಾಹರಣೆ 45

```
<script>
var d = new Date();
d.setMinutes(15);
alert(d);
</script>
```

ಫಲಿತಾಂಶ :(Fig 26)

setSeconds() ಮೆಥೆಡ್

setSeconds() ಮೆಥೆಡ್, ಇದು date ಆಬ್ಜೆಕ್ಟ್‌ನ ಸೆಕೆಂಡುಗಳನ್ನು ಹೊಂದಿಸುತ್ತದೆ (0-59).

Fig 26

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ 46

```
<script>
var d = new Date();
d.setSeconds(20);
alert(d);
</script>
```

ಫಲಿತಾಂಶ :(Fig 27)

ದಿನಾಂಕಗಳನ್ನು ಹೋಲಿಸುವುದು (Compare Dates)

ದಿನಾಂಕಗಳನ್ನು ಸುಲಭವಾಗಿ ಹೋಲಿಸಬಹುದು.

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು ಇವತ್ತಿನ ದಿನಾಂಕವನ್ನು ಜನೇವರಿ 14, 2100 ರ ಜೊತೆಗೆ ಹೋಲಿಸುತ್ತದೆ.

Fig 27

ಉದಾಹರಣೆ 47

```
var date1 = new Date(2010, 00, 15); //Year, Month, Date
var date2 = new Date(2011, 00, 15); //Year, Month, Date
if (date1 > date2)
{
alert("Date One is greather then Date Two.");
}
else
{
alert("Date Two is greather then Date One");
}
```

© NIMI NOT TO BE REPUBLISHED

IIS ಉಪಯೋಗಿಸಿ ವೆಬ್ ಪ್ರೊಜೆಕ್ಟನ್ನು ನಿಯೋಜಿಸುವುದು (Deploy web project using IIS)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- DOM ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ
- DOM ಮೆಥೆಡ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- DOM ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- HTML DOM ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- HTML DOM ಇವೆಂಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟವೇರನ್ನು ತಿಳಿಯಿರಿ

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ HTML DOM

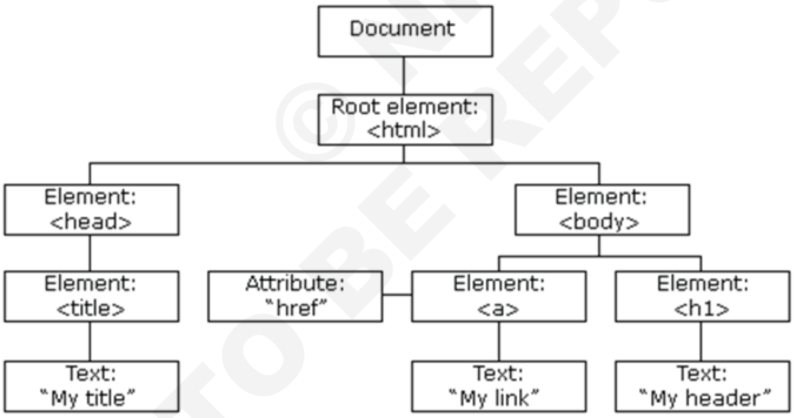
HTML DOM ನೊಂದಿಗೆ, HTML ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸ್ವೀಕರಿಸಬಹುದು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು.

HTML DOM (Document Object Model): ವೆಬ್ ಪುಟವು ಲೋಡ್ ಆದಾಗ, ಬ್ರೌಸರ್ ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್‌ನ್ನು ರಚಿಸುತ್ತದೆ.

HTML DOM ಮಾಡೆಲ್‌ನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಟ್ರೀ(tree) ನಂತೆ ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ.

HTML DOM ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಟ್ರೀ (HTML DOM Tree of Objects)

ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್‌ನೊಂದಿಗೆ, ಡೈನಾಮಿಕ್ HTML ನ್ನು ರಚಿಸಲು ಅವಶ್ಯಕವಿರುವ ಎಲ್ಲ ಶಕ್ತಿಯನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ.



- ಪುಟದಲ್ಲಿರುವ ಎಲ್ಲ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬದಲಾಯಿಸಬಲ್ಲದು.
- ಪುಟದಲ್ಲಿರುವ ಎಲ್ಲ HTML ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬದಲಾಯಿಸಬಲ್ಲದು.
- ಪುಟದಲ್ಲಿರುವ ಎಲ್ಲ CSS ಸ್ಟೈಲ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಬದಲಾಯಿಸಬಲ್ಲದು.
- ಪುಟದಲ್ಲಿ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಮತ್ತು ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ತೆಗೆದುಹಾಕಬಲ್ಲದು.
- ಹೊಸ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಮತ್ತು ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಸೇರಿಸಬಲ್ಲದು.
- ಪುಟದಲ್ಲಿ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಎಲ್ಲ HTML ಈವೆಂಟ್‌ಗಳಿಗೆ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಪ್ರತಿಕ್ರಿಯೆ ನೀಡಬಲ್ಲದು.

ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್ (DOM)

DOM ಒಂದು W3C((World Wide Web Consortium) ಮಾನದಂಡವಾಗಿದೆ.

ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು DOM ಒಂದು ಮಾನದಂಡವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ :

© NIMI NOT TO BE REPUBLISHED

“W3C Document Object Model (DOM) ಒಂದು ವೇದಿಕೆಯಾಗಿದೆ ಮತ್ತು ಭಾಷಾ ತಟಸ್ಥ ಮಾಧ್ಯಮವಾಗಿದೆ(language-neutral interface)

ಇದು ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಮತ್ತು ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳನ್ನು ಕ್ರಿಯಾತ್ಮಕವಾಗಿ ಸ್ವೀಕರಿಸಲು ಮತ್ತು ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿಷಯಗಳನ್ನು , ರಚನೆಯನ್ನು ಮತ್ತು ಶೈಲಿಯನ್ನು ನವೀಕರಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.”

W3C DOM ಮಾನದಂಡವು 3 ವಿಭಿನ್ನ ಭಾಗಗಳಲ್ಲಿ ಪ್ರತ್ಯೇಕಿಸಲ್ಪಟ್ಟಿದೆ:

- Core DOM - ಎಲ್ಲ ವಿಧದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ಒಂದು ಪ್ರಮಾಣಿತ ಮಾದರಿಯಾಗಿದೆ.
- XML DOM - XML ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ಒಂದು ಪ್ರಮಾಣಿತ ಮಾದರಿಯಾಗಿದೆ.
- HTML DOM - HTML ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳಿಗೆ ಒಂದು ಪ್ರಮಾಣಿತ ಮಾದರಿಯಾಗಿದೆ.

HTML DOM ಎಂದರೇನು ? (What is the HTML DOM?)

HTML DOM ಪ್ರಮಾಣಿತ ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್ ಆಗಿದೆ ಮತ್ತು HTML ಗೆ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಇಂಟರ್‌ಫೇಸ್ ಆಗಿದೆ. ಇದು ಕೆಳಗಿನವುಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ :

- HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿ
- ಎಲ್ಲ HTML ಎಲಿಮೆಂಟ್‌ಗಳ ಪ್ರಪರ್ಟಿಗಳನ್ನು
- ಎಲ್ಲ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸಲು ಮೆಥೆಡ್‌ಗಳು
- ಎಲ್ಲ HTML ಎಲಿಮೆಂಟ್‌ಗಳ ಈವೆಂಟ್‌ಗಳನ್ನು

ಇನ್ನೊಂದು ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, HTML DOM ಎಂದರೆ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹೇಗೆ ಸ್ವೀಕರಿಸುವುದು, ಬದಲಾಯಿಸುವುದು, ಸೇರಿಸುವುದು ಅಥವಾ ತೆಗೆದುಹಾಕುವುದು ಎಂಬುದಕ್ಕೆ ಒಂದು ಮಾನದಂಡವಾಗಿದೆ.

HTML DOM ಮೆಥೆಡ್‌ಗಳು (HTML DOM Methods)

HTML DOM ಮೆಥೆಡ್‌ಗಳು, ನೀವು ನಿರ್ವಹಿಸಬಹುದಾದ ಕ್ರಿಯೆಗಳಾಗಿವೆ. (HTML ಎಲಿಮೆಂಟ್‌ಗಳ ಮೇಲೆ)

HTML DOM ಪ್ರಪರ್ಟಿಗಳು ಮೌಲ್ಯಗಳಾಗಿವೆ (HTML ಎಲಿಮೆಂಟ್‌ಗಳ). ಇವುಗಳನ್ನು ನೀವು ಸೆಟ್ ಅಥವಾ ಬದಲಾವಣೆ ಮಾಡಬಹುದು.

DOM ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಇಂಟರ್‌ಫೇಸ್ (The DOM Programming Interface)

HTML DOM ನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನೊಂದಿಗೆ ಪಡೆದುಕೊಳ್ಳಬಹುದು(ಮತ್ತು ಇತರ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳೊಂದಿಗೂ)

DOM ನಲ್ಲಿ, ಎಲ್ಲ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳೆಂದು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ. ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಇಂಟರ್‌ಫೇಸ್ ಎಂದರೆ ಪ್ರತಿಯೊಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ ಪ್ರಾಪರ್ಟಿಗಳು ಮತ್ತು ಮೆಥೆಡ್‌ಗಳಾಗಿವೆ. ನೀವು ಪಡೆದುಕೊಳ್ಳುವ (get) ಮತ್ತು ಹೊಂದಿಸುವ (set) ಮೌಲ್ಯಗಳನ್ನು ಪ್ರಾಪರ್ಟಿ ಎನ್ನುತ್ತಾರೆ.(ಉದಾ : HTML ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಬದಲಾಯಿಸುವುದು). ನೀವು ನಿರ್ವಹಿಸುವ ಕ್ರಿಯೆಯನ್ನು ಮೆಥೆಡ್ ಎನ್ನುತ್ತಾರೆ.(ಉದಾ: HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುವುದು ಅಥವಾ ತೆಗೆದುಹಾಕುವುದು)

ಉದಾಹರಣೆ 1

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯು <p> ಎಲಿಮೆಂಟ್ id=”demo” ಹೊಂದಿರುವ ವಿಷಯವನ್ನು (innerHTML) ಬದಲಾಯಿಸುತ್ತದೆ.

```
<html>
<body>
<p id=”demo”></p>
<script>
document.getElementById(”demo”).innerHTML = “Welcome to JavaScript!”;
</script>
</body>
</html>
```

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, getElementById ಒಂದು ಮೆಥೆಡ್ ಆಗಿದೆ, innerHTML ಒಂದು ಪ್ರಾಪರ್ಟಿ ಆಗಿದೆ.

getElementById ಮೆಥೆಡ್

HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ಸ್ವೀಕರಿಸುವ ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾದ ಮಾರ್ಗವೆಂದರೆ ಆ ಎಲಿಮೆಂಟ್‌ನ id ಯನ್ನು ಉಪಯೋಗಿಸುವುದು. ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ getElementById ಮೆಥೆಡ್ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕಲು id=”demo” ನ್ನು ಉಪಯೋಗಿಸಿದೆ.

innerHTML ಪ್ರಾಪರ್ಟಿ

ಒಂದು ಎಲಿಮೆಂಟ್‌ನ ವಿಷಯವನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಅತ್ಯಂತ ಸುಲಭವಾದ ಮಾರ್ಗವೆಂದರೆ innerHTML ಪ್ರಾಪರ್ಟಿಯನ್ನು ಉಪಯೋಗಿಸುವುದಾಗಿದೆ. innerHTML ಪ್ರಾಪರ್ಟಿಯು HTML ಎಲಿಮೆಂಟ್‌ಗಳ ವಿಷಯವನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಮತ್ತು ಬದಲಾಯಿಸಲು ಉಪಯುಕ್ತವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸೂಚನೆ : ಯಾವುದೇ HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು , <html> ಮತ್ತು <body> ಗಳನ್ನೂ ಒಳಗೊಂಡಂತೆ ಪಡೆದುಕೊಳ್ಳಲು ಅಥವಾ ಬದಲಾಯಿಸಲು innerHTML ಪ್ರಾಪರ್ಟಿಯನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

HTML DOM ಡಾಕ್ಯುಮೆಂಟ್ (HTML DOM Document)

HTML DOM ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ (HTML DOM document object)

ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ ನಿಮ್ಮ ವೆಬ್ ಪುಟದಲ್ಲಿರುವ ಎಲ್ಲ ಇತರ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಓನರ್(ಮಾಲೀಕ) ಆಗಿದೆ. HTML DOM ಆಬ್ಜೆಕ್ಟ್ ಮಾಡೆಲ್‌ನಲ್ಲಿ, ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್ ನಿಮ್ಮ ವೆಬ್ ಪುಟವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ನೀವು ಒಂದು HTML ಪುಟದಲ್ಲಿನ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಬಯಸಿದರೆ, ನೀವು ಯಾವಾಗಲೂ ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಪಡೆಯುವುದರಿಂದ ಪ್ರಾರಂಭಿಸಿ.

HTML ನ್ನು ಪ್ರವೇಶಿಸಲು ಮತ್ತು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸಲು ನೀವು ಹೇಗೆ ಡಾಕ್ಯುಮೆಂಟ್ ಆಬ್ಜೆಕ್ಟನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಎಂಬುದಕ್ಕೆ ಈ ಕೆಳಗೆ ಕೆಲವು ಉದಾಹರಣೆಗಳನ್ನು ನೀಡಲಾಗಿದೆ.

HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕುವುದು (Finding HTML Elements)

ಮೆಥೆಡ್	ವಿವರಣೆ
document.getElementById()	ಎಲಿಮೆಂಟ್ id ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು
document.getElementsByTagName()	tag name ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು
document.getElementsByClassName()	class name ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು

HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು (Changing HTML Elements)

ಮೆಥೆಡ್	ವಿವರಣೆ
element.innerHTML=	ಒಂದು ಎಲಿಮೆಂಟ್‌ನ inner HTML ನ್ನು ಬದಲಾಯಿಸುವುದು
element.attribute=	ಒಂದು HTML ಎಲಿಮೆಂಟ್‌ನ ಅಟ್ರಿಬ್ಯೂಟನ್ನು ಬದಲಾಯಿಸುವುದು
element.setAttribute(attribute,value)	ಒಂದು HTML ಎಲಿಮೆಂಟ್‌ನ ಅಟ್ರಿಬ್ಯೂಟನ್ನು ಬದಲಾಯಿಸುವುದು
element.style.property=	ಒಂದು HTML ಎಲಿಮೆಂಟ್‌ನ ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸುವುದು

HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು ಮತ್ತು ತೆಗೆದುಹಾಕುವುದು (Adding and Deleting HTML Elements)

ಮೆಥೆಡ್	ವಿವರಣೆ
document.createElement()	HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ರಚಿಸುತ್ತದೆ.
document.removeChild()	HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ (delete)
document.appendChild()	HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ.
document.replaceChild()	HTML ಎಲಿಮೆಂಟ್‌ನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.
document.write(text)	HTML output stream ನಲ್ಲಿ ಬರೆಯುತ್ತದೆ.

Events handlers ಗಳನ್ನು ಸೇರಿಸುವುದು.

ಮೆಥೆಡ್	ವಿವರಣೆ
document.getElementById(id).onclick=function(){code}	onclick event ಗೆ event handler ಕೋಡ್‌ನ್ನು ಸೇರಿಸುವುದು.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನ HTML DOM ಎಲಿಮೆಂಟ್‌ಗಳು(JavaScript HTML DOM Elements)

HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕುವುದು

ಸಾಮಾನ್ಯವಾಗಿ, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ ನೀವು HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸಲು ಬಯಸುತ್ತೀರಿ.

ಇದನ್ನು ಮಾಡಲು ನೀವು ಮೊದಲು ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಕಂಡುಹಿಡಿಯಬೇಕು. ಇದಕ್ಕಾಗಿ ಎರಡು ಮಾರ್ಗಗಳಿವೆ.

- id ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು
- tag name ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು
- class name ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು
- HTML object collections ದಿಂದ ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು

id ದಿಂದ HTML ಎಲಿಮೆಂಟನ್ನು ಹುಡುಕುವುದು

DOM ನಲ್ಲಿ ಎಲಿಮೆಂಟ್ id ಉಪಯೋಗಿಸಿ, HTML ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕುವುದು ಅತ್ಯಂತ ಸುಲಭವಾದ ಮಾರ್ಗವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಈ ಉದಾಹರಣೆಯು id="demo" ಬಳಸಿ ಎಲೆಮೆಂಟನ್ನು ಕಂಡುಹಿಡಿಯುತ್ತದೆ:

ಉದಾಹರಣೆ 2

```
var x = document.getElementById("demo");
```

ಒಂದು ವೇಳೆ ಎಲೆಮೆಂಟ್ ಕಂಡುಬಂದರೆ, ಮೆಥೆಡ್ ಎಲೆಮೆಂಟನ್ನು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿ (x ನಲ್ಲಿ) ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಎಲೆಮೆಂಟ್ ಕಂಡುಬಂದಿಲ್ಲವಾದರೆ, x null ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರುತ್ತದೆ.

tag name ದಿಂದ HTML ಎಲೆಮೆಂಟನ್ನು ಹುಡುಕುವುದು

ಈ ಉದಾಹರಣೆಯು id="main" ಬಳಸಿ ಎಲೆಮೆಂಟ್‌ನ್ನು ಕಂಡುಹಿಡಿಯುತ್ತದೆ, ನಂತರ main ನಲ್ಲಿರುವ ಎಲ್ಲ <p> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕುತ್ತದೆ.

ಉದಾಹರಣೆ 3

```
var x = document.getElementById("main");

var y = x.getElementsByTagName("p");
```

class name ದಿಂದ HTML ಎಲೆಮೆಂಟನ್ನು ಹುಡುಕುವುದು

ಒಂದೇ class name ದಿಂದ ನೀವು ಎಲ್ಲ HTML ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹುಡುಕಲು ಬಯಸಿದರೆ, get ElementsByClassName() ಈ ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿರಿ.

ಉದಾಹರಣೆ 4

```
document.getElementsByClassName("intro");
```

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ class="intro" ಇರುವ ಎಲ್ಲ ಎಲೆಮೆಂಟ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

> **ಸೂಚನೆ : class name ದಿಂದ ಎಲೆಮೆಂಟಗಳನ್ನು ಹುಡುಕುವುದು Internet Explorer 5,6,7, and 8 ಗಳಲ್ಲಿ ಕೆಲಸ ಮಾಡುವುದಿಲ್ಲ.**

HTML object collections ದಿಂದ HTML ಎಲೆಮೆಂಟನ್ನು ಹುಡುಕುವುದು

ಈ ಉದಾಹರಣೆಯು id="frm1" ಬಳಸಿ ಫಾರ್ಮ್ ಕಲೆಕ್ಷನ್‌ನಲ್ಲಿ, ಫಾರ್ಮ್ ಎಲೆಮೆಂಟ್‌ನ್ನು ಹುಡುಕುತ್ತದೆ, ಮತ್ತು ಎಲ್ಲ ಎಲೆಮೆಂಟ್ ಮೌಲ್ಯಗಳನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.:

ಮೆಥೆಡ್	ವಿವರಣೆ	DOM
document.anchors	name ಅಟ್ರಿಬ್ಯೂಟ್‌ನಲ್ಲಿರುವ ಮೌಲ್ಯದೊಂದಿಗೆ ಎಲ್ಲ <a> ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.applets	ಎಲ್ಲ <applet> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ (HTML5 ಅಸಮ್ಮತವಾಗಿದೆ)	1
document.baseURI	ಡಾಕ್ಯುಮೆಂಟ್‌ನ absolute base URI ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.body	<body> ಎಲೆಮೆಂಟ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.cookie	ಡಾಕ್ಯುಮೆಂಟ್‌ನ cookie ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.doctype	ಡಾಕ್ಯುಮೆಂಟ್‌ನ doctype ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.documentElement	<html> ಎಲೆಮೆಂಟನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.documentMode	ಬ್ರೌಸರ್‌ನಿಂದ ಉಪಯೋಗಿಸಲ್ಪಟ್ಟ mode ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.documentURI	ಡಾಕ್ಯುಮೆಂಟ್‌ನ URI ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.domain	ಡಾಕ್ಯುಮೆಂಟ್ ಸರ್ವರ್‌ನ domain name ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.domConfig	DOM configuration ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.embeds	ಎಲ್ಲ <embed> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.forms	ಎಲ್ಲ <form> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.head	<head> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.images	ಎಲ್ಲ <image> ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.implementation	DOM implementation ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.inputEncoding	ಡಾಕ್ಯುಮೆಂಟ್‌ನ encoding (character set) ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.lastModified	ಡಾಕ್ಯುಮೆಂಟ್ ನವೀಕರಿಸಲ್ಪಟ್ಟ ದಿನಾಂಕ ಮತ್ತು ಸಮಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.links	href ನಲ್ಲಿ ಎಲ್ಲ <area> ಮತ್ತು <a> ಎಲೆಮೆಂಟ್‌ಗಳ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.readyState	ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಸ್ಥಿತಿಯನ್ನು (ಲೋಡ್ ಆಗುತ್ತಿರುವ) ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.referrer	ರೆಫರರ್‌ನ URI (ಲಿಂಕ್ ಮಾಡುವ ಡಾಕ್ಯುಮೆಂಟ್) ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1

© NIMI NOT TO BE REPUBLISHED

document.scripts	ಎಲ್ಲ <script> ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.strictErrorChecking	ಎರರ್(ದೋಷ) ಪರಿಶೀಲನೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	3
document.title	<title> ಎಲಿಮೆಂಟ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1
document.URL	ಡಾಕ್ಯುಮೆಂಟ್‌ನ ಸಂಪೂರ್ಣ URL ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ	1

ಉದಾಹರಣೆ 5

```
var x = document.getElementById("frm1");
var text = "";
var i;
for (i = 0; i < x.length; i++) {
text += x.elements[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

ಕೆಳಗಿನ HTML ಆಬ್ಜೆಕ್ಟ್‌ಗಳು(ಮತ್ತು object collections) ಸಹ ಉಪಯೋಗಿಸಬಹುದು

- document.anchors
- document.body
- document.documentElement
- document.embeds
- document.forms
- document.head
- document.images
- document.links
- document.scripts
- document.title

HTML ಎಲಿಮೆಂಟ್‌ಗಳ ವಿಷಯವನ್ನು ಬದಲಾಯಿಸಲು HTML DOM ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ.

HTML ಔಟ್‌ಪುಟ್ ಸ್ಟ್ರೀಮ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು.

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಡೈನಾಮಿಕ್ HTML ಕಂಟೆಂಟನ್ನು ರಚಿಸಬಲ್ಲದು. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ನಲ್ಲಿ, HTML ಔಟ್‌ಪುಟ್ ಸ್ಟ್ರೀಮ್‌ನಲ್ಲಿ ನೇರವಾಗಿ ಬರೆಯಲು document.write() ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ 6

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(Date());
</script>
</body>
</html>
```

> **ಸೂಚನೆ : ಡಾಕ್ಯುಮೆಂಟ್ ಲೋಡ್ ಆದ ನಂತರ document. write() ಎಂದಿಗೂ ಬಳಸಬೇಡಿರಿ. ಇದು ಡಾಕ್ಯುಮೆಂಟನ್ನು ಮತ್ತೆ ತಿದ್ದಿ ಬರೆಯುತ್ತದೆ.**

HTML ಕಂಟೆಂಟನ್ನು ಬದಲಾಯಿಸುವುದು (Changing HTML Content)

HTML ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟನ್ನು ಬದಲಾಯಿಸುವ ಅತ್ಯಂತ ಸುಲಭವಾದ ಮಾರ್ಗವೆಂದರೆ innerHTML ಪ್ರಾಪರ್ಟಿಯನ್ನು ಉಪಯೋಗಿಸುವುದಾಗಿದೆ. HTML ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟ್ ಬದಲಾಯಿಸಲು ಕೆಳಗಿನ ಸಿಂಟ್ಯಾಕ್ಸ್ ಉಪಯೋಗಿಸಿ:

```
document.getElementById(id).innerHTML = new HTML
```

ಈ ಉದಾಹರಣೆಯು <p> ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 7

```
<html>
<body>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!" ;
</script>
</body>
</html>
```

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ :

- ಮೇಲಿನ HTML ಡಾಕ್ಯುಮೆಂಟ್ id= "p1" ದೊಂದಿಗೆ <p> ಎಲಿಮೆಂಟನ್ನು ಒಳಗೊಂಡಿದೆ.

© NIMI NOT TO BE REPUBLISHED

- id= “p1”ಹೊಂದಿರುವ ಎಲಿಮೆಂಟನ್ನು ಪಡೆದುಕೊಳ್ಳಲು HTML DOM ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.
- ಆ ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟನ್ನು (innerHTML) ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ “New text!” ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.

ಈ ಉದಾಹರಣೆಯು <h1> ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 8

```
<!DOCTYPE html>
<html>
<body>
<h1 id= “id1” >Old Heading</h1>
<script>
var element = document.getElementById(“id1”);
element.innerHTML = “ New Heading”;
</script>
</body>
</html>
```

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ :

- ಮೇಲಿನ HTML ಡಾಕ್ಯುಮೆಂಟ್ id= “id1” ದೊಂದಿಗೆ <p> ಎಲಿಮೆಂಟನ್ನು ಒಳಗೊಂಡಿದೆ.
- id= “id1”ಹೊಂದಿರುವ ಎಲಿಮೆಂಟನ್ನು ಪಡೆದುಕೊಳ್ಳಲು HTML DOM ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.
- ಆ ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟನ್ನು (innerHTML) ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ “New Heading” ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.

ಅಟ್ರಿಬ್ಯೂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸುವುದು (Changing the Value of an Attribute)

ಒಂದು HTML ಅಟ್ರಿಬ್ಯೂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸಲು, ಈ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.

```
document.getElementById(id).attribute = new value
```

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯು <img> ಎಲಿಮೆಂಟ್‌ನ src ಅಟ್ರಿಬ್ಯೂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 9

```
<!DOCTYPE html>
<html>
<body>
<img id= “Image1” src= “flower.gif”>
<script>
document.getElementById(“Image1”).src = “newflower.jpg”;
</script>
</body>
</html>
```

- ಮೇಲಿನ HTML ಡಾಕ್ಯುಮೆಂಟ್ id= “Image1” ದೊಂದಿಗೆ <img> ಎಲಿಮೆಂಟನ್ನು ಒಳಗೊಂಡಿದೆ.
- id= “Image1” ಹೊಂದಿರುವ ಎಲಿಮೆಂಟನ್ನು ಪಡೆದುಕೊಳ್ಳಲು HTML DOM ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.
- ಆ ಎಲಿಮೆಂಟ್‌ನ src ಅಟ್ರಿಬ್ಯೂಟನ್ನು ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ “smiley.gif” ದಿಂದ “landscape.jpg” ಗೆ ಬದಲಾಯಿಸುತ್ತದೆ.

HTML ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸುವುದು (Changing HTML Style)

ಒಂದು HTML ಎಲಿಮೆಂಟ ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸಲು, ಈ ಸಿಂಟ್ಯಾಕ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.

```
document.getElementById(id).style.property = new style
```

ಈ ಉದಾಹರಣೆಯು <p> ಎಲಿಮೆಂಟ್‌ನ ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 10

```
<html>
<body>
<p id= “p2” >Hello World!</p>
<script>
document.getElementById(“p2”).style.color = “green” ;
</script>
<p>The paragraph above was changed by a script.</p>
</body>
</html>
```

ಈವೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುವುದು (Using Events): ಒಂದು ಈವೆಂಟ್ ಸಂಭವಿಸಿದಾಗ HTML DOM ನಿಮಗೆ ಕೋಡ್‌ಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಈವೆಂಟ್‌ಗಳು

© NIMI NOT TO BE REPUBLISHED

ಸಾಮಾನ್ಯವಾಗಿ HTML ಎಲಿಮೆಂಟ್‌ಗಳಿಗೆ "ವಿಷಯಗಳು ಸಂಭವಿಸಿದಾಗ" ಬ್ರೌಸರ್‌ನಿಂದ ರಚಿಸಲ್ಪಡುತ್ತವೆ.

- ಒಂದು ಎಲಿಮೆಂಟನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ
- ಒಂದು ಪುಟವು ಲೋಡ್ ಆದಾಗ
- ಇನ್ಪುಟ್ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಬದಲಿಸಿದಾಗ

ಈ ಕೆಳಗಿನ ಉದಾಹರಣೆಯು id="id1" ಹೊಂದಿರುವ HTML ಎಲಿಮೆಂಟ್‌ನ ಸ್ಟೈಲನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ:

ಉದಾಹರಣೆ 11

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Heading1</h1>
<button type="button" onclick="document.getElementById('id1').style.color = 'blue'">
Click Me</button>
</body>
</html>
```

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ HTML DOM ಈವೆಂಟ್‌ಗಳು (JavaScript HTML DOM Events)

HTML DOM, ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್‌ಗೆ HTML ಈವೆಂಟ್‌ಗಳಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಈವೆಂಟ್‌ಗಳಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುವುದು

ಒಂದು ಈವೆಂಟ್ ಸಂಭವಿಸಿದಾಗ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೆಲಸ ಮಾಡುತ್ತದೆ. (ಉದಾಹರಣೆಗೆ, ಬಳಕೆದಾರರು ಒಂದು HTML ಎಲಿಮೆಂಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ.) ಬಳಕೆದಾರರು ಒಂದು ಎಲಿಮೆಂಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ, ಕೋಡ್ ಅನ್ನು ಕಾರ್ಯಗತ ಮಾಡಲು HTML ಈವೆಂಟ್ ಅಟ್ರಿಬ್ಯೂಟಗೆ ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಕೋಡನ್ನು ಸೇರಿಸಿ.

onclick=JavaScript

HTML ಈವೆಂಟ್‌ಗಳ ಉದಾಹರಣೆಗಳು:

- ಬಳಕೆದಾರರು ಮೌಸ್ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ
- ಒಂದು ವೆಬ್ ಪುಟವು ಲೋಡ್ ಆದಾಗ
- ಒಂದು ಚಿತ್ರವು ಲೋಡ್ ಆದಾಗ
- ಮೌಸ್ ಒಂದು ಎಲಿಮೆಂಟ್ ಮೇಲೆ ಚಲಿಸಿದಾಗ
- ಒಂದು ಇನ್ಪುಟ್ ಫೀಲ್ಡ್ ಬದಲಾದಾಗ
- ಒಂದು HTML ಫಾರ್ಮ್ ಸಲ್ಲಿಸಿದಾಗ
- ಬಳಕೆದಾರರು ಕೀ ಯನ್ನು ಒತ್ತಿದಾಗ

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ ಬಳಕೆದಾರರು ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ <h1> ಎಲಿಮೆಂಟ್‌ನ ಕಂಟೆಂಟ್ ಬದಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 12

```
<!DOCTYPE html>
<html>
<body>
<h1 onclick="this.innerHTML = 'Ooops!'">Click on this text!</h1>
</body>
</html>
```

HTML ಈವೆಂಟ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು (HTML Event Attributes)

HTML ಎಲಿಮೆಂಟ್‌ಗಳಿಗೆ ಈವೆಂಟ್‌ಗಳನ್ನು ನಿಯೋಜಿಸಲು ನೀವು ಈವೆಂಟ್ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ13

button ಎಲಿಮೆಂಟ್‌ಗೆ onclick ಈವೆಂಟ್‌ನ್ನು ನಿಯೋಜಿಸುವುದು :

```
<button onclick="displayDate()">Try it</button>
```

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, displayDate ಎಂಬ ಹೆಸರಿನ ಫಂಕ್ಷನ್ ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

HTML DOM ಬಳಸಿ ಈವೆಂಟ್‌ಗಳನ್ನು ನಿಯೋಜಿಸುವುದು(Assign Events Using the HTML DOM)

ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಉಪಯೋಗಿಸಿ HTML ಎಲಿಮೆಂಟ್‌ಗಳಿಗೆ ನೀವು ಈವೆಂಟ್‌ಗಳನ್ನು ನಿಯೋಜಿಸಲು HTML DOM ಅನುಮತಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ 14

button ಎಲಿಮೆಂಟ್‌ಗೆ onclick ಈವೆಂಟ್‌ನ್ನು ನಿಯೋಜಿಸುವುದು:

```
<script>
document.getElementById("myBtn").onclick = displayDate;
</script>
```

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, displayDate ಎಂಬ ಹೆಸರಿನ ಫಂಕ್ಷನ್‌ನ್ನು id="myBtn".ಹೊಂದಿರುವ HTML ಎಲಿಮೆಂಟ್‌ಗೆ ನಿಯೋಜಿಸಲಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ ಫಂಕ್ಷನ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

onload ಮತ್ತು onunload ಈವೆಂಟ್‌ಗಳು(The onload and onunload Events)

ಬಳಕೆದಾರರು ಪುಟವನ್ನು ಪ್ರವೇಶಿಸಿದಾಗ ಅಥವಾ ಬಿಟ್ಟಾಗ onload ಮತ್ತು onunload ಈವೆಂಟ್‌ಗಳು ಪ್ರಚೋದಿಸಲ್ಪಡುತ್ತವೆ. ವಿಸಿಟರ್ (ಸಂದರ್ಶಕರು) ಗಳ ಬ್ರೌಸರ್‌ನ ಪ್ರಕಾರ ಹಾಗೂ ಬ್ರೌಸರ್‌ನ ಆವೃತ್ತಿಯನ್ನು ಪರಿಶೀಲಿಸುವುದು ಮತ್ತು ಮಾಹಿತಿಯ ಆಧಾರದ ಮೇಲೆ ವೆಬ್ ಪುಟದ ಸರಿಯಾದ ಆವೃತ್ತಿಯನ್ನು ಲೋಡ್ ಮಾಡುವುದು ಮುಂತಾದವುಗಳಿಗೆ onload ಈವೆಂಟ್‌ನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. onload ಮತ್ತು onunload ಈವೆಂಟ್‌ಗಳನ್ನು ಕುಕೀ(cookies) ಗಳೊಂದಿಗೆ ವ್ಯವಹರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 15

```
<body onload="checkCookies()">
```

onchange ಈವೆಂಟ್ (The onchange Event)

onchange ಈವೆಂಟ್‌ನ್ನು ಹೆಚ್ಚಾಗಿ ಇನ್ಪುಟ್ ಫೀಲ್ಡ್‌ಗಳ ವ್ಯಾಲಿಡೇಷನ್‌ನ ಸಂಯೋಜನೆಯೊಂದಿಗೆ ಬಳಸಲಾಗುತ್ತದೆ. ಕೆಳಗಿನ ಉದಾಹರಣೆಯು onchange ನ್ನು ಹೇಗೆ ಬಳಸಬೇಕು ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತದೆ. ಬಳಕೆದಾರರು ಇನ್ಪುಟ್ ಫೀಲ್ಡ್‌ನ ಕಂಟೆಂಟನ್ನು ಬದಲಾಯಿಸಿದಾಗ upperCase() ಫಂಕ್ಷನ್‌ನ್ನು ಕರೆಯಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ 16

```
<input type="text" id="fname" onchange="upperCase()">
```

onmouseover ಮತ್ತು onmouseout ಈವೆಂಟ್‌ಗಳು (The onmouseover and onmouseout Events)

ಬಳಕೆದಾರರು HTML ಎಲಿಮೆಂಟ್ ಮೇಲೆ ಮೌಸ್ ಪಾಯಿಂಟರ್ ಅನ್ನು ಚಲಿಸಿದಾಗ ಒಂದು ಫಂಕ್ಷನ್‌ಅನ್ನು ಪ್ರಚೋದಿಸಲು onmouseover ಮತ್ತು onmouseout ಈವೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

onmousedown, onmouseup ಮತ್ತು onclick ಈವೆಂಟ್‌ಗಳು (The onmousedown, onmouseup and onclick Events)

onmousedown, onmouseup ಮತ್ತು onclick ಈವೆಂಟ್‌ಗಳು ಇವೆಲ್ಲವೂ ಮೌಸ್ - ಕ್ಲಿಕ್‌ನ ಭಾಗಗಳಾಗಿವೆ. ಮೊದಲು ಮೌಸ್ -ಬಟನ್ ಕ್ಲಿಕ್‌ಮಾಡಿದಾಗ onmousedown ಈವೆಂಟ್ ಪ್ರಚೋದಿಸಲ್ಪಡುತ್ತದೆ, ನಂತರ ಮೌಸ್ - ಬಟನ್‌ನ್ನು ಬಿಟ್ಟಾಗ onmouseup ಈವೆಂಟ್ ಪ್ರಚೋದಿಸಲ್ಪಡುತ್ತದೆ, ಕೊನೆಗೆ ಮೌಸ್ - ಕ್ಲಿಕ್ ಪೂರ್ಣಗೊಂಡಾಗ, onclick ಈವೆಂಟ್ ಪ್ರಚೋದಿಸಲ್ಪಡುತ್ತದೆ.

onmousedown ಮತ್ತು onmouseup

ಬಳಕೆದಾರರು ಮೌಸ್ ಬಟನ್ ಅನ್ನು ಹಿಡಿದಿಟ್ಟುಕೊಂಡಾಗ ಚಿತ್ರ(image) ವು ಬದಲಾಗುತ್ತದೆ.

onload

ಒಂದು ಪುಟವು ಲೋಡ್ ಆಗುವುದನ್ನು ಪೂರ್ಣಗೊಳಿಸಿದಾಗ alert ಬಾಕ್ಸ್ ಅನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ.

onfocus

ಒಂದು ಇನ್ಪುಟ್ ಫೀಲ್ಡ್ ಫೋಕಸ್ ಆದಾಗ ಅದರ ಹಿನ್ನೆಲೆ -ಬಣ್ಣ(background -color) ಬದಲಾಗುತ್ತದೆ.

Mouse Events

ಕರ್ಸರ್ ಒಂದು ಎಲಿಮೆಂಟ್ ಮೇಲೆ ಚಲಿಸಿದಾಗ ಅದರ ಬಣ್ಣ ಬದಲಾಯಿಸುತ್ತದೆ.

DOM Event Listener

addEventListener() ಮೆಥೆಡ್

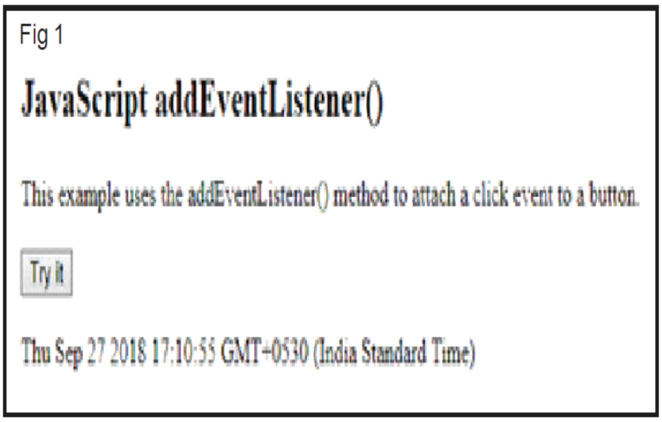
ಬಳಕೆದಾರರು ಬಟನ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿದಾಗ ಅದು ಫೈರ್ ಮಾಡುವ ಒಂದು Event Listener ಅನ್ನು ಸೇರಿಸಿ.

ಉದಾಹರಣೆ 17

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript addEventListener()</h2>
<p>This example uses the addEventListener() method to attach a click event to a button.</p>
<button id="myBtn">Try it</button>
<p id="demo"></p>
script>
document.getElementById("myBtn").
addEventListener("click", displayDate);
function displayDate() {
document.getElementById("demo").innerHTML = Date();
}
</script>
</body>
<</html>
```


© NIMI NOT TO BE REPUBLISHED

Result: (Fig 1)



- addEventListener() ಮೆಥೆಡ್ ಒಂದು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಎಲೆಮೆಂಟ್‌ಗೆ event handler ಅನ್ನು ಲಗತ್ತಿಸುತ್ತದೆ.
- addEventListener() ಮೆಥೆಡ್ ಒಂದು ಎಲೆಮೆಂಟ್‌ಗೆ event handler ಅನ್ನು ಲಗತ್ತಿಸುತ್ತದೆ, ಆದರೆ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ event handler ನ್ನು ತಿದ್ದಿ ಬರೆಯುವುದಿಲ್ಲ. (over-write)
- ನೀವು ಒಂದು ಎಲೆಮೆಂಟ್‌ಗೆ ಹಲವಾರು event handler ಗಳನ್ನು ಸೇರಿಸಬಹುದು.
- ನೀವು ಒಂದು ಎಲೆಮೆಂಟ್‌ಗೆ ಒಂದೇ ರೀತಿಯ ಹಲವಾರು event handler ಗಳನ್ನು ಸೇರಿಸಬಹುದು. ಎಂದರೆ, ಎರಡು "click" ಈವೆಂಟ್‌ಗಳು.
- ನೀವು HTML ಎಲೆಮೆಂಟ್‌ಗಳಿಗಷ್ಟೇ ಅಲ್ಲದೇ DOM ಆಬ್ಜೆಕ್ಟ್‌ಗಳಿಗೂ event listeners ಗಳನ್ನು ಸೇರಿಸಬಹುದು. ಎಂದರೆ window object.
- ಬಬ್ಲಿಂಗ್‌ಗೆ ಈವೆಂಟ್ ಹೇಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ನಿಯಂತ್ರಿಸಲು addEventListener() ಮೆಥೆಡ್ ಸುಲಭವಾಗಿಸುತ್ತದೆ.
- addEventListener() ಮೆಥೆಡ್ ಉಪಯೋಗಿಸುವಾಗ, ಉತ್ತಮ ಓದುವಿಕೆಗಾಗಿ ಜಾವಾಸ್ಕ್ರಿಪ್ಟನ್ನು HTML Markup ನಿಂದ ಪ್ರತ್ಯೇಕಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ನೀವು HTML ಮಾರ್ಕಪ್‌ನ್ನು ನಿಯಂತ್ರಿಸದಿದ್ದರೂ ನಿಮಗೆ event listeners ಗಳನ್ನು ಸೇರಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.
- ನೀವು removeEventListener() ಮೆಥೆಡ್‌ನ್ನು ಬಳಸಿ event listeners ಗಳನ್ನು ಸುಲಭವಾಗಿ ತೆಗೆದುಹಾಕಬಹುದು.

ಓಪನ್ ಸೋರ್ಸ್ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು (Open Source Software)

ಜನಪ್ರೀಯ ಉಚಿತವಾದ ಮತ್ತು ಓಪನ್ ಸೋರ್ಸ್ ಆಗಿರುವ ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ನಿರ್ಮಿಸುವ ಸಾಧನಗಳನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ನಿರ್ಮಿಸಬಹುದು. ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ, ನೀವು ಒಬ್ಬ ವೈಯಕ್ತಿಕ ಉದ್ಯಮಿ ಅಥವಾ ಒಂದು ವ್ಯವಹಾರದ ಸಂಸ್ಥೆಯನ್ನು ಪ್ರತಿನಿಧಿಸುವ ವ್ಯಕ್ತಿಯಾಗಿದ್ದರೆ, ವೈಯಕ್ತಿಕ ಮತ್ತು ವೃತ್ತಿಪರ ಬೆಳವಣಿಗೆಗೆ ಒಂದು ವೆಬ್ ಸೈಟ್ ಅತ್ಯವಶ್ಯವಾಗಿದೆ. ಆಕರ್ಷಕ ವೆಬ್ ಸೈಟ್‌ಗಳನ್ನು ನಿರ್ಮಿಸಲು ಸಂಸ್ಥೆಗಳು ಸಾಕಷ್ಟು ಹಣವನ್ನು ಖರ್ಚು ಮಾಡುತ್ತಿವೆ. ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಅಥವಾ ಇಂಟರ್‌ನೆಟ್ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಜ್ಞಾನವಿಲ್ಲದೇ ನೀವು ನಿಮ್ಮ ಸ್ವಂತ ವೆಬ್ ಸೈಟನ್ನು ನಿರ್ಮಿಸಲು ಈ ಕೆಳಗಿನ ಓಪನ್ ಸೋರ್ಸ್ ವೆಬ್ ಸೈಟ್ ನಿರ್ಮಿಸುವ ಸಾಧನಗಳನ್ನು ಬಳಸಬಹುದು:

1 **WordPress**

WordPress ನ ಅಧಿಕೃತ ವೆಬ್ ಸೈಟ್‌ಗಳು ಹೀಗಿವೆ

https://wordpress.com and

https://wordpress.org/.

2 **Kompozer**

Kompozer ನ ಅಧಿಕೃತ ವೆಬ್ ಸೈಟ್‌ಗಳು ಹೀಗಿವೆ

https://www.kompozer.net.

3 **Joomla**

Joomla ನ ಅಧಿಕೃತ ವೆಬ್ ಸೈಟ್‌ಗಳು ಹೀಗಿವೆ

https://www.joomla.org/.

4 **Drupal**

Drupal ನ ಅಧಿಕೃತ ವೆಬ್ ಸೈಟ್‌ಗಳು ಹೀಗಿವೆ

https://www.drupal.org/.

5 **OpenCms**

OpenCms ನ ಅಧಿಕೃತ ವೆಬ್ ಸೈಟ್‌ಗಳು ಹೀಗಿವೆ

http:// www.opencms.org/en/.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಡಾಟಾ ವಿಜುವಲೈಸೇಷನ್ ಅಥವಾ ಎಕ್ಸೆಲ್ ಉಪಯೊಗಿಸಿ ವಿಶ್ಲೇಸುವುದು (Data Visualization or analysis using Excel)

ಸುಧಾರಿತ ಸೂತ್ರಗಳನ್ನು ಮತ್ತು ಮ್ಯಾಕ್ರೊಗಳನ್ನು ರಚಿಸುವುದು (Create advanced formulas and macros)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ರೆಫರನ್ಸ್‌ಗಳ ವಿಧಗಳು
- ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್.

ಸುಧಾರಿತ ಎಕ್ಸೆಲ್ (Advanced Excel)

ರೆಫರನ್ಸ್‌ಗಳ ವಿಧಗಳು (Types of references)

ಸೆಲ್ ರೆಫರನ್ಸ್‌ಗಳಲ್ಲಿ ಎರಡು ವಿಧಗಳಿವೆ : ರಿಲೇಟಿವ್ ಮತ್ತು ಅಬ್ಸಲ್ಯೂಟ್ (relative and absolute.) ರಿಲೇಟಿವ್ ಮತ್ತು ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳು ಅವುಗಳನ್ನು ಕಾಪಿ ಮಾಡಿದಾಗ ಮತ್ತು ಇತರ ಸೆಲ್‌ಗಳಲ್ಲಿ ತುಂಬಿದಾಗ ವಿಭಿನ್ನವಾಗಿ ವರ್ತಿಸುತ್ತವೆ. ಒಂದು ಸೆಲ್ ನಲ್ಲಿರುವ ಫಾರ್ಮ್ಯುಲಾವನ್ನು ಇನ್ನೊಂದು ಸೆಲ್‌ಗೆ ಕಾಪಿ ಮಾಡಿದಾಗ ರಿಲೇಟಿವ್ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಬದಲಾಗುತ್ತವೆ. ಮತ್ತೊಂದೆಡೆ ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳು, ಅವುಗಳನ್ನು ಎಲ್ಲಿ ಕಾಪಿ ಮಾಡಿದರೂ ಸ್ಥಿರವಾಗಿರುತ್ತವೆ.

ರಿಲೇಟಿವ್ ರೆಫರನ್ಸ್‌ಗಳು (Relative references)

ಪೂರ್ವನಿಯೋಜಿತವಾಗಿ, ಎಲ್ಲ ಸೆಲ್ ರೆಫರನ್ಸ್‌ಗಳು ರಿಲೇಟಿವ್ ರೆಫರನ್ಸ್‌ಗಳು ಆಗಿವೆ. ಇವುಗಳನ್ನು ಅನೇಕ ಸೆಲ್‌ಗಳಿಗೆ ಕಾಪಿ ಮಾಡಿದಾಗ, ಅವುಗಳ ರೋ ಮತ್ತು ಕಾಲಂಗಳ ಸಂಬಂಧಿತ ಸ್ಥಾನದ ಆಧಾರದ ಮೇಲೆ ಬದಲಾಗುತ್ತವೆ. ಉದಾಹರಣೆಗೆ : ನೀವು ಫಾರ್ಮ್ಯುಲಾ =A1+B1 ಅನ್ನು row 1 ರಿಂದ row 2 ಕ್ಕೆ ಕಾಪಿ ಮಾಡಿದರೆ, ಫಾರ್ಮ್ಯುಲಾ =A2+B2 ಎಂದು ಬದಲಾಗುತ್ತದೆ. ನೀವು ಒಂದೇ ಫಾರ್ಮ್ಯುಲಾವನ್ನು ಹಲವಾರು ರೋಗಳಲ್ಲಿ ಅಥವಾ ಕಾಲಂಗಳಲ್ಲಿ ಪುನರಾವರ್ತಿಸುವ ಅವಶ್ಯಕತೆ ಇದ್ದಾಗ ರಿಲೇಟಿವ್ ರೆಫರನ್ಸ್‌ಗಳು ವಿಶೇಷವಾಗಿ ಅನುಕೂಲಕರವಾಗಿವೆ.

ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳು (Absolute references)

ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ನೀವು ಸೆಲ್‌ಗಳನ್ನು ಇನ್ನೊಂದೆಡೆ ತುಂಬಿದಾಗ ಸೆಲ್ ರೆಫರನ್ಸ್‌ಗಳು ಬದಲಾಗಬಾರದು ಎಂದು ಬಯಸಬಹುದು. ರಿಲೇಟಿವ್ ರೆಫರನ್ಸ್‌ಗಳಿಗಿಂತ ಭಿನ್ನವಾಗಿ, ಸೆಲ್‌ಗಳನ್ನು ಕಾಪಿ ಮಾಡಿದಾಗ ಅಥವಾ ಇನ್ನೊಂದಿಡೆ ತುಂಬಿದಾಗ ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳು ಬದಲಾಗುವುದಿಲ್ಲ. ರೋ ಮತ್ತು/ಅಥವಾ ಕಾಲಂಗಳನ್ನು ಸ್ಥಿರವಾಗಿಸಿಕೊಳ್ಳಲು ನೀವು ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಒಂದು ಫಾರ್ಮ್ಯುಲಾದಲ್ಲಿ ಕಾಲಂ ಮತ್ತು ರೋ ನ ಮೊದಲು ಡಾಲರ್ ಚಿನ್ಹೆ ($) ಯನ್ನು ಸೇರಿಸುವುದರಿಂದ ಹೆಸರಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು ಕಾಲಂ ಅಥವಾ ರೋ ನ ಮೊದಲು ಸೇರಿಸಿದ್ದರೆ(ಆದರೆ ಎರಡಕ್ಕೂ ಅಲ್ಲ) , ಅದಕ್ಕೆ ಮಿಶ್ರ (Mixed) ರೆಫರನ್ಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

\$A\$2	ಕಾಪಿ ಮಾಡಿದಾಗ ಕಾಲಂ ಮತ್ತು ರೋ ಎರಡೂ ಬದಲಾಗುವುದಿಲ್ಲ.
A\$2	ಕಾಪಿ ಮಾಡಿದಾಗ ರೋ ಬದಲಾಗುವುದಿಲ್ಲ
\$A2	ಕಾಪಿ ಮಾಡಿದಾಗ ಕಾಲಂ ಬದಲಾಗುವುದಿಲ್ಲ

ನೀವು ರಿಲೇಟಿವ್ (A2) ಮತ್ತು ಅಬ್ಸಲ್ಯೂಟ್ (\$A\$2) ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ಹೆಚ್ಚು ಫಾರ್ಮ್ಯುಲಾಗಳಲ್ಲಿ ಬಳಸುತ್ತೀರಿ. ಮಿಶ್ರ ರೆಫರನ್ಸ್‌ಗಳನ್ನು ಕಡಿಮೆ ಬಳಸಲಾಗುತ್ತದೆ.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಫಾರ್ಮ್ಯುಲಾಗಳನ್ನು ಬರೆಯುವಾಗ ರಿಲೇಟಿವ್ ಮತ್ತು ಅಬ್ಸಲ್ಯೂಟ್ ಮತ್ತು ಮಿಕ್ಸ್ಡ್ ಸೆಲ್ ರೆಫರನ್ಸ್‌ಗಳ ನಡುವೆ ಬದಲಾಯಿಸಲು ನೀವು ಕೀಬೋರ್ಡ್‌ನಲ್ಲಿರುವ F4 ಕೀಯನ್ನು ಬಳಸಬಹುದು. ಇದನ್ನು ಕೆಳಗೆ ನೀಡಿದ ವಿಡಿಯೊದಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ. ಇದು ಅಬ್ಸಲ್ಯೂಟ್ ರೆಫರನ್ಸ್ ತ್ವರಿತವಾಗಿ ಸೇರಿಸಲು ಸುಲಭವಾದ ಮಾರ್ಗವಾಗಿದೆ.

ಎಕ್ಸೆಲ್ ಲಿಂಕ್ ಕಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟ್ ಮತ್ತು ಎಕ್ಸೆಲ್ ರಕ್ಷಣೆ (Excel Linkage custom Format and Excel Protection)

ಅಂತರ್‌ನಿರ್ಮಿತ (built-in) ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳು ನಿಮ್ಮ ಅಗತ್ಯತೆಗಳನ್ನು ಪೂರೈಸದಿದ್ದರೆ, ನೀವು ಹೊಸ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ರಚಿಸಬಹುದು.ಆದು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ ಆಧರಿತವಾಗಿರುತ್ತದೆ ಮತ್ತು ನೀವು ಇದನ್ನು ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ ಪಟ್ಟಿಗೆ ಸೇರಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ, ನೀವು ಗ್ರಾಹಕರ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿರುವ ಒಂದು ಸ್ಪ್ರೆಡ್‌ಶೀಟನ್ನು ತಯಾರಿಸುತ್ತಿದ್ದರೆ, ದೂರವಾಣಿ ಸಂಖ್ಯೆಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ನೀವು ಒಂದು ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು

© NIMI NOT TO BE REPUBLISHED

ತಯಾರಿಸಬಹುದು. ಒಂದು ಸೆಲ್‌ನಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳನ್ನು ದೂರವಾಣಿ ಸಂಖ್ಯೆಗಳನ್ನಾಗಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ನೀವು ಆ ಸಂಖ್ಯೆಗಳಿಗೆ ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ನ್ನು ಅನ್ವಯಿಸಬಹುದು.

ಪ್ರಮುಖ ಸೂಚನೆ : ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳು ಕೇವಲ ಆ ಸಂಖ್ಯೆ ಯಾವ ರೀತಿ ಪ್ರದರ್ಶಿತವಾಗುತ್ತದೆ ಎನ್ನುವುದರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತದೆ ಮತ್ತು ಆ ಸಂಖ್ಯೆಯ ಆಧರಿತ ಮೌಲ್ಯದ ಪರಿಣಾಮ ಬೀರುವುದಿಲ್ಲ. ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ಸಕ್ರೀಯ ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗಿರುತ್ತದೆ ಮತ್ತು ನೀವು ಹೊಸ ವರ್ಕ್‌ಬುಕ್‌ನ್ನು ತೆರೆದರೆ ಅದರಲ್ಲಿ ಲಭ್ಯವಿರುವುದಿಲ್ಲ.

ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ರಚಿಸುವುದು (Create a custom number format)

1 Home Tab ನ Number ಗ್ರೂಪ್‌ನಲ್ಲಿ, Number Format list ನ ಕೆಳಭಾಗದಲ್ಲಿರುವ More Number Formats ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 Format Cells ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Category ಅಡಿಯಲ್ಲಿ , Custom ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Type ಲಿಸ್ಟ್‌ನಲ್ಲಿ, ನೀವು ರಚಿಸಲು ಬಯಸಿದ ಫಾರ್ಮ್ಯಾಟನ್ನು ಹೆಚ್ಚಾಗಿ ಹೋಲುವ ಒಂದು ಬಿಲ್ಟ್- ಇನ್ (ಅಂತರ್‌ನಿರ್ಮಿತ) ಫಾರ್ಮ್ಯಾಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಉದಾಹರಣೆಗೆ, 0.00.

ನೀವು ಆಯ್ಕೆ ಮಾಡಿದ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್ Type ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ.

4 Type ಬಾಕ್ಸ್‌ನಲ್ಲಿ , ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್ ಕೋಡ್‌ಗಳನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಿ ನಿಮಗೆ ಬೇಕಾದ ನಿಖರವಾದ ಫಾರ್ಮ್ಯಾಟನ್ನು ರಚಿಸಿರಿ. ಉದಾಹರಣೆಗೆ 000-000-0000.

ನಿಮ್ಮ ಬದಲಾವಣೆಗಳು ಬಿಲ್ಟ್ ಇನ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಬದಲಾಯಿಸುವುದಿಲ್ಲ. ಅದರ ಬದಲಾಗಿ, ನಿಮ್ಮ ಬದಲಾವಣೆಗಳು ಹೊಸ ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ರಚಿಸುತ್ತವೆ.

5 ನಿಮ್ಮ ಕೆಲಸವು ಪೂರ್ಣಗೊಂಡ ನಂತರ, OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಅನ್ವಯಿಸುವುದು (Apply a custom number format)

1 ನೀವು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲು ಬಯಸುವ ಸೆಲ್ ಅಥವಾ ಸೆಲ್ ರೇಂಜ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.

2 Home Tab ನ Number ಗ್ರೂಪ್‌ನಲ್ಲಿ, Number Format list ನ ಕೆಳಭಾಗದಲ್ಲಿರುವ More Number Formats ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Format Cells ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Category ಅಡಿಯಲ್ಲಿ , Custom ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

4 Type ಲಿಸ್ಟ್‌ನಲ್ಲಿ, ನೀವು ರಚಿಸಲು ಬಯಸಿದ ಫಾರ್ಮ್ಯಾಟನ್ನು ಹೆಚ್ಚಾಗಿ ಹೋಲುವ ಒಂದು ಬಿಲ್ಟ್- ಇನ್ (ಅಂತರ್‌ನಿರ್ಮಿತ) ಫಾರ್ಮ್ಯಾಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ. ಉದಾಹರಣೆಗೆ, 0.00.

ನೀವು ಆಯ್ಕೆ ಮಾಡಿದ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್ Type ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಕಾಣಿಸಿಕೊಳ್ಳುತ್ತದೆ.

5 OK ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು (Delete a custom number format)

1 Home Tab ನ Number ಗ್ರೂಪ್‌ನಲ್ಲಿ, Number Format list ನ ಕೆಳಭಾಗದಲ್ಲಿರುವ More Number Formats ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

2 Format Cells ಡಯಾಲಾಗ್ ಬಾಕ್ಸ್‌ನಲ್ಲಿ, Category ಅಡಿಯಲಿ ್ಲರುವ , Custom ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 Type ಲಿಸ್ಟ್‌ನಲ್ಲಿ, ಕಸ್ಟಮ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ ಮತ್ತು Delete ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

ಸೂಚನೆ :

- **ಬಿಲ್ಟ್ - ಇನ್ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಲು(ಡಿಲೀಟ್) ಮಾಡಲು ಬರುವುದಿಲ್ಲ.**
- **ತೆಗೆದುಹಾಕಲಾದ ಕಸ್ಟಮ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಿಂದ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಲಾದ ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿರುವ ಯಾವುದೇ ಸೆಲ್‌ಗಳು ಡೀಫಾಲ್ಟ್ General ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡುತ್ತವೆ.**

ಎಕ್ಸೆಲ್ ರಕ್ಷಣೆ (Excel Protection)

ನಿಮ್ಮ ಎಕ್ಸೆಲ್ ಫೈಲ್‌ನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಇತರರು ಪ್ರವೇಶಿಸುವುದನ್ನು ತಡೆಯಲು ನಿಮ್ಮ ಎಕ್ಸೆಲ್ ಫೈಲನ್ನು ಪಾಸ್‌ವರ್ಡ್‌ನಿಂದ ರಕ್ಷಿಸಿ.

ಸೂಚನೆ : ಈ ಪಾಠವು ಫೈಲ್ ಮಟ್ಟದ ರಕ್ಷಣೆಯನ್ನು ಮಾತ್ರ ಒಳಗೊಂಡಿದೆ, ಮತ್ತು ವರ್ಕ್‌ಬುಕ್ ಅಥವಾ ವರ್ಕ್‌ಶೀಟ್ ರಕ್ಷಣೆ ಅಲ್ಲ. ನಿಮ್ಮ ಎಕ್ಸೆಲ್ ಫೈಲ್, ವರ್ಕ್‌ಬುಕ್ ಅಥವಾ ವರ್ಕ್‌ಶೀಟ್‌ಗಳ ರಕ್ಷಣೆಯ ನಡುವಿನ ವ್ಯತ್ಯಾಸವನ್ನು ತಿಳಿಯಲು Protection and security in Excel ನ್ನು ನೋಡಿರಿ.

1 File > Info ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 Protect Workbook box ಆಯ್ಕೆ ಮಾಡಿರಿ ಮತ್ತು Encrypt with Password ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

3 Password box ನಲ್ಲಿ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ನಮೂದಿಸಿ, ಮತ್ತು OK ಆಯ್ಕೆ ಮಾಡಿರಿ.

4 Reenter Password box ನಲ್ಲಿ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ದೃಢೀಕರಿಸಿ ಮತ್ತು OK ಆಯ್ಕೆ ಮಾಡಿರಿ.

ಎಚ್ಚರಿಕೆ (Warning)

- ಮರೆತುಹೋದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಪುನಃ ಪಡೆದುಕೊಳ್ಳಲು ಸಾಧ್ಯವಿಲ್ಲ, ಹೀಗಾಗಿ ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್ ಸ್ಮರಣೀಯವಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ.
- ನೀವು ಬಳಸುವ ಪಾಸ್‌ವರ್ಡ್‌ಗಳಿಗೆ ಅದರ ಉದ್ದ, ಅಕ್ಷರಗಳು, ಅಥವಾ ಸಂಖ್ಯೆಗಳಿಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಯಾವುದೇ ನಿರ್ಬಂಧಗಳಿಲ್ಲ, ಆದರೆ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ಕೇಸ್ ಸೆನ್ಸಿಟಿವ್ ಆಗಿರುತ್ತವೆ.
- ಪಾಸ್‌ವರ್ಡ್‌ನಿಂದ ರಕ್ಷಿತವಾದ ವಿಶೇಷವಾಗಿ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್‌ನಂತಹ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿರುವ ಫೈಲ್‌ಗಳನ್ನು ವಿತರಿಸುವುದು ಯಾವಾಗಲೂ ಸುರಕ್ಷಿತವಲ್ಲ.
- ಫೈಲ್‌ಗಳನ್ನು ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಇತರ ಬಳಕೆದಾರರೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳುವಾಗ ಜಾಗರೂಕರಾಗಿರಿ. ನೀವು ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ಅನಪೇಕ್ಷಿತ ಬಳಕೆದಾರರ ಕೈಗೆ ಸಿಗುವ ಅಪಾಯವನ್ನು ಎದುರಿಸುತ್ತೀರಿ. ನೆನಪಿರಲಿ ದುರುದ್ದೇಶದಿಂದ ಪಾಸ್‌ವರ್ಡ್‌ನೊಂದಿಗೆ ಫೈಲ್ ಲಾಕ್ ಮಾಡುವುದು ನಿಮ್ಮ ಫೈಲನ್ನು ರಕ್ಷಣೆ ಮಾಡುವುದಿಲ್ಲ.

ಎಕ್ಸೆಲ್ ಸಲಹೆಗಳು (Excel Tips)

1 ಡಾಟಾವನ್ನು ಗುರುತಿಸಲು ಮತ್ತು ಅರ್ಥಮಾಡಿಕೊಳ್ಳಲು ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ

2 ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ರೋ ಅಥವಾ ಕಾಲಂ ಅನ್ನು ಸೇರಿಸಿ.

3 ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಸರಳಗೊಳಿಸಲು ಫಿಲ್ಟರ್‌ಗಳನ್ನು (filter) ಬಳಸಿ

4 ನಕಲಿ ಡಾಟಾ ಪಾಯಿಂಟ್‌ಗಳು ಅಥವಾ ಸೆಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಿ.

5 ರೋಗಳನ್ನು ಕಾಲಂಗಳಾಗಿ ವರ್ಗಾಯಿಸಿ

6 ಕಾಲಂಗಳ ನಡುವೆ ಟೆಕ್ಸ್ಟ್ ಮಾಹಿತಿಯನ್ನು ವಿಭಜಿಸಿ.

7 ಸರಳ ಲೆಕ್ಕಗಳಿಗಾಗಿ ಈ ಸೂತ್ರಗಳನ್ನು ಬಳಸಿರಿ.

8 ನಿಮ್ಮ ಸೆಲ್‌ಗಳಲ್ಲಿರುವ ಸಂಖ್ಯೆಗಳ ಸರಾಸರಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಿರಿ

9 ಡಾಟಾದ ಆಧಾರದ ಮೇಲೆ ಸೆಲ್‌ಗಳು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ತಮ್ಮ ಬಣ್ಣವನ್ನು ಬದಲಾಯಿಸಲು ಕಂಡಿಷನಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ ಬಳಸಿ. .10 ಕೆಲವು ನಿರ್ದಿಷ್ಟ ಎಕ್ಸೆಲ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಸ್ವಯಂ ಚಾಲನೆಗೊಳಿಸಲು IF ಎಕ್ಸೆಲ್ ಫಾರ್ಮ್ಯುಲಾವನ್ನು ಬಳಸಿ

11 ಒಂದು ಸೆಲ್‌ನ ಫಾರ್ಮ್ಯುಲಾವನ್ನು, ಅದು ಎಲ್ಲಿ ಚಲಿಸುತ್ತದೆ ಎಂಬುದನ್ನೂ ಲೆಕ್ಕಿಸದೇ ಒಂದೇ ರೀತಿ ಇರಿಸಿಕೊಳ್ಳಲು ಡಾಲರ್ ಚಿನ್ಹೆ ($) ಯನ್ನು ಉಪಯೋಗಿಸಿ.

12 ಡಾಟಾವನ್ನು ಒಂದು ಶೀಟ್‌ನ ಒಂದು ಪ್ರದೇಶದಿಂದ ಇನ್ನೊಂದಕ್ಕೆ ಎಳೆಯಲು VLOOKUP ಫಂಕ್ಷನ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.

13 ಅಡ್ಡ (ಹಾರಿಝಾಂಟಲ್) ಕಾಲಂಗಳಿಂದ ಡಾಟಾವನ್ನು ಎಳೆಯಲು INDEX and MATCH ಫಾರ್ಮ್ಯುಲಾಗಳನ್ನು ಉಪಯೋಗಿಸಿ.

14 ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಯಾವುದೇ ಸೆಲ್‌ಗಳ ರೇಂಜ್‌ಗಳಲ್ಲಿರುವ ಪದಗಳನ್ನು ಅಥವಾ ಸಂಖ್ಯೆಗಳನ್ನು ಎಣಿಸಲು COUNTIF ಫಂಕ್ಷನ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.

15 ampersand ಚಿನ್ಹೆಯನ್ನು ಬಳಸಿ ಸೆಲ್‌ಗಳನ್ನು ಸಂಯೋಜಿಸಿ.

16 ಚೆಕ್‌ಬಾಕ್ಸ್‌ಗಳನ್ನು ಸೇರಿಸಿ.

17 ಒಂದು ಸೆಲ್‌ನ್ನು ವೆಬ್‌ಸೈಟ್‌ಗೆ ಹೈಪರ್‌ಲಿಂಕ್ ಮಾಡಿರಿ.

18 ಡ್ರಾಪ್ ಡೌನ್ ಮೆನುಗಳನ್ನು ಸೇರಿಸಿ

19 ಫಾರ್ಮ್ಯಾಟ್ ಪೇಂಟರ್ ಬಳಸಿ.

ಪೈವೋಟ್ ಟೇಬಲ್ (Pivot Table)

ಡಾಟಾವನ್ನು ಸಂಕ್ಷೇಪಿಸಲು, ವಿಶ್ಲೇಶಿಸಲು , ಅನ್ವೇಷಿಸಲು ಮತ್ತು ಡಾಟಾದ ಸಾರಾಂಶವನ್ನು ಪ್ರಸ್ತುತಪಡಿಸಲು ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪೈವೋಟ್ ಚಾರ್ಟ್‌ಗಳು ಪೈವೋಟ್ ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಸಾರಾಂಶ ಡಾಟಾಗೆ ದೃಶ್ಯಗಳನ್ನು ಸೇರಿಸುವ ಮೂಲಕ ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳಿಗೆ ಪೂರಕವಾಗಿರುತ್ತವೆ. ಮತ್ತು ಹೋಲಿಕೆಗಳನ್ನು , ಮಾದರಿಗಳನ್ನು ಮತ್ತು ಟ್ರೆಂಡ್‌ಗಳನ್ನು ಸುಲಭವಾಗಿ ನೋಡಲು ನಿಮಗೆ ಅನುಮತಿಸುತ್ತವೆ. ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳು ಮತ್ತು ಪೈವೋಟ್ ಚಾರ್ಟ್‌ಗಳು ಎರಡೂ ನಿಮ್ಮ ಉದ್ಯಮದಲ್ಲಿರುವ ನಿರ್ಣಾಯಕ ಡಾಟಾದ ಬಗ್ಗೆ ನಿರ್ಧಾರಗಳನ್ನು ಮಾಡಲು ನಿಮ್ಮನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತವೆ. ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸಲು ನೀವು ಬಾಹ್ಯ ಡಾಟಾ ಸೋರ್ಸ್‌ಗಳಾದ SQL Server tables, SQL Server Analysis Services cubes, Azure Marketplace, Office Data Connection (.odc) files, XML files, Access databases, ಮತ್ತು ಟೆಕ್ಸ್ಟ್ ಫೈಲ್‌ಗಳು ಮುಂತಾದವುಗಳಿಗೆ ಸಹ ಸಂಪರ್ಕಿಸಬಹುದು ಅಥವಾ ಹೊಸ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸಲು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಪೈವೋಟ್ ಟೇಬಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ದೊಡ್ಡ ಪ್ರಮಾಣದ ಡಾಟಾವನ್ನು ತ್ವರಿತವಾಗಿ ಸಂಕ್ಷೇಪಿಸಲು ಪೈವೋಟ್ ಟೇಬಲ್ ಒಂದು ಸಂವಾದಾತ್ಮಕ ಮಾರ್ಗವಾಗಿದೆ. ನೀವು ಸಂಖ್ಯಾತ್ಮಕ ಡಾಟಾವನ್ನು ವಿವರವಾಗಿ ವಿಶ್ಲೇಷಣೆ ಮಾಡಲು

© NIMI NOT TO BE REPUBLISHED

ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಮತ್ತು ನಿಮ್ಮ ಡಾಟಾದ ಬಗ್ಗೆ ಅನಿರೀಕ್ಷಿತ ಪ್ರಶ್ನೆಗಳಿಗೆ ಉತ್ತರಿಸಬಹುದು. ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ವಿಶೇಷವಾಗಿ ಈ ಕೆಳಗಿನ ಕಾರ್ಯಗಳಿಗಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ:

- ದೊಡ್ಡ ಪ್ರಮಾಣದ ಡಾಟಾವನ್ನು ಹಲವಾರು ಬಳಕೆದಾರ ಸ್ನೇಹಿ ಮಾರ್ಗಗಳಲ್ಲಿ ಪ್ರಶ್ನಿಸುವುದಕ್ಕಾಗಿ
- ಸಂಖ್ಯಾತ್ಮಕ ಡಾಟಾವನ್ನು Subtotaling (ಉಪಮೊತ್ತ) ಮತ್ತು aggregating (ಒಟ್ಟುಗೂಡಿಸುವಿಕೆ) , categories , ಮತ್ತು subcategories, ಗಳಲ್ಲಿ ಡಾಟಾವನ್ನು ಸಂಕ್ಷೇಪ್ತಗೊಳಿಸುವುದು (summarizing) ಮತ್ತು ಕಸ್ಟಮ್ ಲೆಕ್ಕಗಳು (custom calculations) ಮತ್ತು ಫಾರ್ಮ್ಯುಲಾಗಳನ್ನು ರಚಿಸುವುದು.
- ನಿಮ್ಮ ಫಲಿತಾಂಶಗಳನ್ನು ಗಮನವನ್ನು ಕೇಂದ್ರೀಕರಿಸಲು ಡಾಟಾದ ಮಟ್ಟವನ್ನು ವಿಸ್ತರಿಸುವುದು ಮತ್ತು ಕುಗ್ಗಿಸುವುದು ಹಾಗೂ ನಿಮ್ಮ ಆಸಕ್ತಿಯ ಕ್ಷೇತ್ರಗಳಿಗೆ ಡಾಟಾದ ಸಾರಾಂಶದಿಂದ ವಿವರಗಳನ್ನು ತೆಗೆದುಕೊಂಡು ಬರುವುದು.
- ಮೂಲ ಡಾಟಾದ ವಿಭಿನ್ನ ಸಾರಾಂಶಗಳನ್ನು ನೋಡಲು ರೋ ಗಳನ್ನು ಕಾಲಂಗಳಿಗೆ ಅಥವಾ ಕಾಲಂಗಳನ್ನು ರೋ ಗಳಿಗೆ ಬದಲಾಯಿಸುವುದು.(ಅಥವಾ "pivoting")
- ಫಿಲ್ಟರಿಂಗ್(Filtering), ವಿಂಗಡಣೆ (sorting), ಗುಂಪು ಮಾಡುವುದು (grouping), ಮತ್ತು ಷರತ್ತುಬದ್ಧ ಫಾರ್ಮ್ಯಾಟಿಂಗ್ (conditionally formatting) ಇವು ಅತ್ಯಂತ ಉಪಯುಕ್ತ ಮತ್ತು ಆಸಕ್ತಿದಾಯಕ ಡಾಟಾದ ಸಬ್ಸೆಟ್ ಆಗಿವೆ. ಇವು ನಿಮಗೆ ಬೇಕಾದ ಮಾಹಿತಿಯ ಮೇಲೆ ಕೇಂದ್ರೀಕರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ.
- ಸಂಕ್ಷೇಪ್ತ , ಆಕರ್ಷಕ ಮತ್ತು ಟಿಪ್ಪಣಿಗಳನ್ನು ಆನ್‌ಲೈನ್ ಅಥವಾ ಪ್ರಿಂಟ್ ಆದ ವರದಿ (ರಿಪೋರ್ಟ್)ಗಳನ್ನು ಪ್ರಸ್ತುತಪಡಿಸುತ್ತದೆ.

ಉದಾಹರಣೆಗೆ ಎಡಭಾಗದಲ್ಲಿ ಮನೆಯ ವೆಚ್ಚಗಳ ಒಂದು ಸರಳ ಪಟ್ಟಿ ಇಲ್ಲಿದೆ ಮತ್ತು ಬಲಭಾಗದಲ್ಲಿ ಈ ಪಟ್ಟಿಯನ್ನು ಆಧರಿಸಿ ತಯಾರಿಸಿದ ಪೈವೋಟ್ ಟೇಬಲ್ ಇದೆ

ಪೈವೋಟ್ ಟೇಬಲ್‌ನೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಲು ಇರುವ ಮಾರ್ಗಗಳು

ಡಾಟಾ ಸೋರ್ಸ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ಮೂಲಕ ನೀವು ಒಂದು ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ರಚಿಸಿದ ನಂತರ, PivotTable Field List ನಲ್ಲಿ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಜೋಡಿಸುವುದು ಮತ್ತು ಆರಂಭಿಕ ವಿನ್ಯಾಸವನ್ನು ಆಯ್ಕೆ ಮಾಡುವುದು ಮುಂತಾದ ಪೈವೋಟ್ ಟೇಬಲ್‌ನಲ್ಲಿ ಕೆಲಸ ಮಾಡುವಾಗ ನೀವು ಈ ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಬಹುದು:

ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವುದರ ಮೂಲಕ ಡಾಟಾವನ್ನು ಅನ್ವೇಷಿಸುವುದು (Explore the data by doing the following)

- ಡಾಟಾವನ್ನು ವಿಸ್ತರಿಸಿ ಮತ್ತು ಕುಗ್ಗಿಸಿ ಮತ್ತು ಮೌಲ್ಯಗಳನ್ನು ಸಂಬಂಧಿಸಿದ ವಿವರಗಳನ್ನು ತೋರಿಸುವುದು.
- ಫೀಲ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಐಟಂಗಳನ್ನು ವಿಂಗಡಿಸುವುದು, ಫಿಲ್ಟರ್ ಮಾಡುವುದು ಮತ್ತು ಗ್ರೂಪ್ ಮಾಡುವುದು.
- ಸಾರಾಂಶ (summary) ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು ಮತ್ತು ಕಸ್ಟಮ್ ಲೆಕ್ಕಗಳು ಮತ್ತು ಫಾರ್ಮ್ಯುಲಾಗಳನ್ನು ಸೇರಿಸುವುದು.

ಈ ಕೆಳಗಿನವುಗಳ ಮೂಲಕ ಫಾರ್ಮ್ ಲೇಔಟ್ ಮತ್ತು ಫೀಲ್ಡ್ ಅರೇಂಜ್‌ಮೆಂಟ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು

- PivotTable form ನ್ನು ಬದಲಾಯಿಸಿ : Compact, Outline, or Tabular.
- ಫೀಲ್ಡ್‌ಗಳನ್ನು Add, rearrange, and remove ಮಾಡುವುದು .
- ಫೀಲ್ಡ್‌ಗಳ ಅಥವಾ ಐಟಂಗಳ ಕ್ರಮವನ್ನು ಬದಲಾಯಿಸಿ.

ಕೆಳಗಿನವುಗಳನ್ನು ಮಾಡುವ ಮೂಲಕ ಕಾಲಂಗಳ, ರೋ ಗಳ ಮತ್ತು ಸಬ್ ಟೋಟಲ್‌ಗಳ ವಿನ್ಯಾಸವನ್ನು ಬದಲಾಯಿಸುವುದು:

- ಕಾಲಂ ಮತ್ತು ರೋ ಫೀಲ್ಡ್ ಹೆಡರ್‌ಗಳನ್ನು ಆನ್ ಅಥವಾ ಆಫ್ ಮಾಡಿರಿ, ಅಥವಾ ಖಾಲಿ ಸಾಲುಗಳನ್ನು ಪ್ರದರ್ಶಿಸಿ ಅಥವಾ ಮರೆ ಮಾಡಿ.
- ಸಬ್‌ಟೋಟಲ್‌ಗಳನ್ನು ಅವುಗಳ ರೋ ಗಳ ಮೇಲೆ ಅಥವಾ ಕೆಳಗೆ ಪ್ರದರ್ಶಿಸಿ.
- ರಿಫ್ರೆಶ್ ಮಾಡಿದಾಗ ಕಾಲಂ ಅಗಲ(width) ವನ್ನು ಹೊಂದಿಸುವುದು
- ಕಾಲಂ ಫೀಲ್ಡ್‌ನ್ನು ರೋ ಏರಿಯಾಕ್ಕೆ ಅಥವಾ ರೋ ಫೀಲ್ಡನ್ನು ಕಾಲಂ ಏರಿಯಾಕ್ಕೆ ಸರಿಸುವುದು.
- ಹೊರಗಿನ ರೋ ಮತ್ತು ಕಾಲಂ ಐಟಂಗಳಿಗಾಗಿ ಸೆಲ್‌ಗಳನ್ನು ವಿಲೀನಗೊಳಿಸುವುದು (merge) ಅಥವಾ ವಿಲೀನಗೊಳಿಸದೇ(unmerge) ಇರುವುದು.

ಕೆಳಗಿನವುಗಳ ಮೂಲಕ ಖಾಲಿ(blank) ಮತ್ತು ದೋಷಗಳ(error) ಪ್ರದರ್ಶನವನ್ನು ಬದಲಾಯಿಸುವುದು

- ದೋಷಗಳು ಮತ್ತು ಖಾಲಿ ಸೆಲ್‌ಗಳನ್ನು ಹೇಗೆ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ ಎಂಬುದನ್ನು ಬದಲಾಯಿಸುವುದು.
- ಡಾಟಾ ಇಲ್ಲದೇ ಐಟಂಗಳು ಮತ್ತು ಲೇಬಲ್‌ಗಳನ್ನು ಹೇಗೆ ತೋರಿಸಲಾಗುತ್ತದೆ ಎಂಬುದನ್ನು ಬದಲಾಯಿಸಿ.
- ಖಾಲಿ ರೋ ಗಳನ್ನು ಪ್ರದರ್ಶಿಸುವುದು ಅಥವಾ ಮರೆ ಮಾಡುವುದು.

ಕೆಳಗಿನವುಗಳನ್ನು ಮಾಡುವ ಮೂಲಕ ಫಾರ್ಮ್ಯಾಟನ್ನು ಬದಲಾಯಿಸಿ.

- ಸೆಲ್‌ಗಳು ಮತ್ತು ರೇಂಜ್‌ಗಳನ್ನು ಹಸ್ತಚಾಲಿತವಾಗಿ ಮತ್ತು ಷರತ್ತುಬದ್ಧವಾಗಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡುವುದು.

© NIMI NOT TO BE REPUBLISHED

- ಒಟ್ಟಾರೆ ಪೈವೋಟ್ ಟೇಬಲ್ ಫಾರ್ಮ್ಯಾಟ್ ಸ್ಟೈಲ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು.
- ಫೀಲ್ಡ್‌ಗಳಿಗಾಗಿ ನಂಬರ್ ಫಾರ್ಮ್ಯಾಟನ್ನು ಬದಲಾಯಿಸುವುದು.
- OLAP Server ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ್ನು ಸೇರಿಸುವುದು

ಪೈವೋಟ್ ಟೇಬಲ್ ಮತ್ತು ಪೈವೋಟ್ ಚಾರ್ಟ್‌ನ್ನು ರಚಿಸುವುದು (Create Pivot table & Pivot chart)

ಒಂದು ಪೈವೋಟ್ ಟೇಬಲ್ ಅಥವಾ ಪೈವೋಟ್ ಚಾರ್ಟ್‌ನ್ನು ರಚಿಸಲು ನೀವು ಎಕ್ಸೆಲ್ ವರ್ಕ್‌ಶೀಟ್‌ನಿಂದ ಡಾಟಾವನ್ನು ಆಧಾರವಾಗಿ ಬಳಸಬಹುದು. ಡಾಟಾವು ಲಿಸ್ಟ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿ ಇರಬೇಕು. ಮೊದಲನೇ ರೋ ನಲ್ಲಿ ಕಾಲಂ ಲೇಬಲ್‌ಗಳು ಇರಬೇಕು. ಎಕ್ಸೆಲ್ ಇವುಗಳನ್ನು ಫೀಲ್ಡ್ ಹೆಸರುಗಳನ್ನಾಗಿ (Filed Names) ಬಳಸುತ್ತದೆ. ನಂತರದ ರೋ ಗಳಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಸೆಲ್ ಅದರ ಕಾಲಂ ಹೆಡಿಂಗ್‌ಗೆ ಸೂಕ್ತವಾದ ಡಾಟಾವಾನ್ನು ಒಳಗೊಂಡಿರಬೇಕು. ಮತ್ತು ನೀವು ಒಂದೇ ಕಾಲಂನಲ್ಲಿರುವ ಡಾಟಾ ಟೈಪ್‌ಗಳನ್ನು ಮಿಶ್ರಣ ಮಾಡಬಾರದು. ಉದಾಹರಣೆಗೆ, ನೀವು ಕರೆನ್ಸಿ ಮೌಲ್ಯಗಳನ್ನು ಮತ್ತು ದಿನಾಂಕಗಳನ್ನು ಒಂದೇ ಕಾಲಂನಲ್ಲಿ ಮಿಶ್ರಣ ಮಾಡಬಾರದು. ಹೆಚ್ಚುವರಿಯಾಗಿ, ಡಾಟಾ ರೇಂಜ್‌ನಲ್ಲಿ ಯಾವುದೇ ಖಾಲಿ ರೋ ಗಳು ಅಥವಾ ಕಾಲಂಗಳು ಇರಬಾರದು.

ಎಕ್ಸೆಲ್ ಟೇಬಲ್‌ಗಳು ಈಗಾಗಲೆ ಲಿಸ್ಟ್ ಫಾರ್ಮ್ಯಾಟ್‌ನಲ್ಲಿವೆ ಮತ್ತು ಪೈವೋಟ್ ಟೇಬಲ್ ಡಾಟಾ ಸೋರ್ಸ್‌ಗೆ ಉತ್ತಮವಾಗಿವೆ. ನೀವು ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಿದಾಗ, ಎಕ್ಸೆಲ್ ಟೇಬಲ್‌ನಿಂದ ಹೊಸ ಮತ್ತು ನವೀಕರಿಸಿದ ಡಾಟಾ ರಿಫ್ರೆಶ್ ಕಾರ್ಯದಲ್ಲಿ ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಸೇರಿಸಲ್ಪಡುತ್ತದೆ.

ಡೈನಾಮಿಕ್ ಹೆಸರಿನ ರೇಂಜ್‌ನ್ನು ಉಪಯೋಗಿಸುವುದು (Using a dynamic named range)

ಒಂದು ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ನವೀಕರಿಸಲು ಸುಲಭವಾಗುವಂತೆ ಮಾಡಲು, ನೀವು ಡೈನಾಮಿಕ್ ಹೆಸರಿನ ರೇಂಜ್‌ನ್ನು ರಚಿಸಬಹುದು, ಮತ್ತು ಆ ಹೆಸರನ್ನು ಪೈವೋಟ್ ಟೇಬಲ್‌ನ ಡಾಟಾ ಸೋರ್ಸ್ ಆಗಿ ಉಪಯೋಗಿಸಿ. ಹೆಸರಿಸಿದ ರೇಂಜ್ ಇನ್ನೂ ಹೆಚ್ಚಿನ ಡಾಟಾವನ್ನು ಸೇರಿಸಲು ವಿಸ್ತರಿಸಿದರೆ, ಪೈವೋಟ್ ಟೇಬಲ್‌ನ್ನು ರಿಫ್ರೆಶ ಮಾಡುವುದರ ಮೂಲಕ ಹೊಸ ಡಾಟಾವನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

totals ನ್ನು ಒಳಗೊಂಡಿರುವುದು (Including totals) :

ಎಕ್ಸೆಲ್ ಪೈವೋಟ್ ಟೇಬಲ್‌ನಲ್ಲಿ subtotals ಮತ್ತು grand totals ಗಳನ್ನು ಸ್ವಯಂ ತಾನೇ ರಚಿಸುತ್ತದೆ. ನೀವು Data tab ನಲ್ಲಿರುವ Outline group ನ Subtotals ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ತಯಾರಿಸಿದ subtotals ಮತ್ತು grand totals ನ್ನು ಡಾಟಾ ಸೋರ್ಸ್ ಅಟೋಮ್ಯಾಟಿಕ್ ಆಗಿ ಒಳಗೊಂಡಿದ್ದರೆ, ಅದೇ ಕಮಾಂಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು subtotals ಮತ್ತು grand totals ನ್ನು ತೆಗೆದುಹಾಕಬಹುದು.

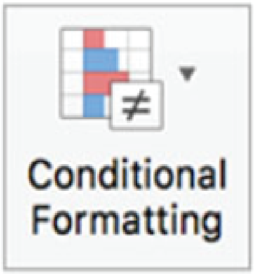
ೆಂಡೀಷನಲ್ ಫರ್ಮ್ಯಾಟಿಂಗ್ (Conditional formatting)

ಕಂಡಿಷನಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ್ನು ಅನ್ವಯಿಸುವುದು

1 ನೀವು ಕಂಡಿಷನಲ್ ಫಾರ್ಮ್ಯಾಟಿಂಗ್‌ನ್ನು ಅನ್ವಯಿಸಲು ಬಯಸುವ ಸೆಲ್‌ಗಳ ರೇಂಜ್‌ನ್ನು , ಟೇಬಲ್‌ನ್ನು , ಅಥವಾ ಸಂಪೂರ್ಣ ಶೀಟನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ

2 Home tab ನಲ್ಲಿ, Conditional Formatting ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

3 ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದನ್ನು ಉಪಯೋಗಿಸಿ:



ಹೈಲೈಟ್ ಮಾಡಲು	ಈ ರೀತಿ ಮಾಡಿ
ನಿರ್ದಿಷ್ಟ ಸೆಲ್‌ಗಳಲ್ಲಿರುವ ಮೌಲ್ಯಗಳು. ಉದಾಹರಣೆಗೆ ಈ ವಾರದ ನಂತರದ ದಿನಾಂಕಗಳು, ಅಥವಾ 50 ಮತ್ತು 100 ರ ನಡುವಿನ ಸಂಖ್ಯೆಗಳು, ಅಥವಾ 10% ಕೆಳಗಿನ ಅಂಕಗಳು	Highlight Cells Rules ಅಥವಾ Top/Bottom Rules ಮೇಲೆ ಪಾಯಿಂಟ್ ಮಾಡಿರಿ, ಮತ್ತು ಸೂಕ್ತವಾದ ಆಯ್ಕೆಯ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ
ಒಂದು ಸೆಲ್ ರೇಂಜ್‌ನಲ್ಲಿರುವ ಮೌಲ್ಯಗಳ ರಿಲೇಷನ್‌ಶಿಪ್. ಸೆಲ್‌ಗಳಾದ್ಯಂತ ಬಣ್ಣದ ಬ್ಯಾಂಡ್‌ನ್ನು ವಿಸ್ತರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗಳು ಬೆಲೆಗಳಲ್ಲಿನ ಹೋಲಿಕೆಗಳು ಅಥವಾ ದೊಡ್ಡ ನಗರಗಳಲ್ಲಿನ ಜನಸಂಖ್ಯೆಗಳನ್ನು ಹೋಲಿಸುವುದು.	Data Bars ಮೇಲೆ ಪಾಯಿಂಟ್ ಮಾಡಿರಿ, ನಂತರ ನೀವು ಬಯಸಿದ fill ಅನ್ನು ಕ್ಲಿಕ್ ಮಾಡಿರಿ.
ಒಂದು ಸೆಲ್ ರೇಂಜ್‌ನಲ್ಲಿರುವ ಮೌಲ್ಯಗಳ ರಿಲೇಷನ್‌ಶಿಪ್. ಸೆಲ್‌ನ ಕಲರ್ ಮೌಲ್ಯದ ನಿಯೋಜನೆಯನ್ನು ಪ್ರತಿಬಿಂಬಿಸುತ್ತದೆ ಕಲರ್ ಸ್ಕೇಲ್‌ನ್ನು ಅನ್ವಯಿಸುತ್ತದೆ	Color Scales ಮೇಲೆ ಪಾಯಿಂಟ್ ಮಾಡಿರಿ, ನಂತರ ನೀವು ಬಯಸಿದ scale ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

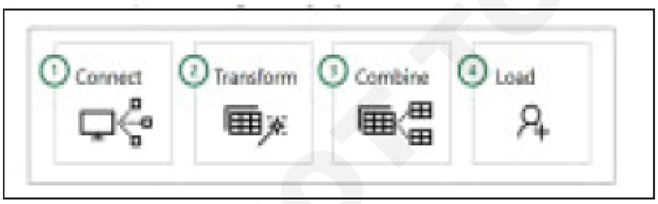
ಮೂರರಿಂದ ಐದು ಗುಂಪುಗಳ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿರುವ ಸೆಲ್ ರೇಂಜ್ , ಇಲ್ಲಿ ಪ್ರತಿಯೊಂದು ಗುಂಪು ತನ್ನದೇ ಆದ ಮಿತಿಯನ್ನು ಹೊಂದಿದೆ. ಉದಾಹರಣೆಗೆ $80,000, $60,000, ಮತ್ತು $40,000 ಕ್ಕಿಂತ ಕೆಳಗಿರುವ ಮಾರಾಟವನ್ನು ತೋರಿಸುವ ಸೆಲ್‌ಗಳನ್ನು ಹೈಲೈಟ್ ಮಾಡಲು ಮೂರು ಐಕನ್‌ಗಳ ಒಂದು ಸೆಟ್ ಅನ್ನು ನೀವು ನಿಯೋಜಿಸಬಹುದು. ಅಥವಾ ಅಟೊಮೊಬೈಲ್‌ಗಳಿಗೆ 5 ಪಾಯಿಂಟ್ ರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ನೀವು ನಿಯೋಜನೆ ಮಾಡಬಹುದು ಮತ್ತು ಐದು ಐಕನ್‌ಗಳ ಸೆಟ್‌ನ್ನು ಅನ್ವಯಿಸಬಹುದು.	Icon Sets ಮೇಲೆ ಪಾಯಿಂಟ್ ಮಾಡಿರಿ, ನಂತರ ಒಂದು ಸೆಟ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ

ಪವರ್ ಕ್ವೆರಿಗಳು (Power Queries)

ಪವರ್ ಕ್ವೆರಿಗಳನ್ನು (ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ Get & Transform ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ) ಉಪಯೋಗಿಸಿ ನೀವು ಬಾಹ್ಯ ಡಾಟಾವನ್ನು ಆಮದು ಮಾಡಿಕೊಳ್ಳಬಹುದು ಅಥವಾ ಸಂಪರ್ಕಿಸಬಹುದು. ಮತ್ತು ಆ ಡಾಟಾಕ್ಕೆ ಆಕಾರವನ್ನು ನೀಡಬಹುದು. ಉದಾಹರಣೆಗೆ, ಕಾಲಂ ನ್ನು ತೆಗೆದುಹಾಕುವುದು, ಡಾಟಾಟೈಪನ್ನು ಬದಲಾಯಿಸುವುದು, ಅಥವಾ ಟೇಬಲ್‌ಗಳನ್ನು ಮರ್ಜ್ ಮಾಡುವುದು, ನಿಮ್ಮ ಅವಶ್ಯಕತೆಗಳನ್ನು ಪೂರೈಸುವ ರೀತಿಯಲ್ಲಿ ಕೆಲಸ ಮಾಡುತ್ತವೆ. ನಂತರ ಚಾರ್ಟ್‌ಗಳನ್ನು ಮತ್ತು ರಿಪೋರ್ಟ್‌ಗಳನ್ನು ರಚಿಸಲು ನೀವು ನಿಮ್ಮ ಕ್ವೆರಿಯನ್ನು ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಲೋಡ್ ಮಾಡಬಹುದು.ಡಾಟಾವನ್ನು ಇಲಿ ್ಲಯವರೆಗೆ ನವೀಕರಿಸಲು ನಿಯಮಿತವಾಗಿ ನೀವು ಅದನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಬಹುದು. ಮೂರು ಎಕ್ಸೆಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿ ಲಭ್ಯವಿದೆ. Excel for Windows, Excel for Mac ಮತ್ತು Excel for the Web. ಎಲ್ಲ ಪವರ್ ಕ್ವೆರಿಗಳ ಸಹಾಯ ವಿಷಯಗಳ ಸಾರಾಂಶಕ್ಕಾಗಿ Power Query for Excel Help ನ್ನು ನೋಡಿರಿ.

ಪವರ್ ಕ್ವೆರಿಯ ನಾಲ್ಕು ಹಂತಗಳು (The four phases of Power Query)

ಪವರ ಕ್ವೆರಿಯನ್ನು ಬಳಸಲು ನಾಲ್ಕು ಹಂತಗಳಿವೆ.



1 Connect(**ಸಂಪರ್ಕ**) : ಸೇವೆಯ ಮೇಲೆ ಅಥವಾ ಸ್ಥಳಿಯವಾಗಿ ಕ್ಲೌಡ್‌ನಲ್ಲಿರುವ ಡಾಟಾಗೆ ಸಂಪರ್ಕ ಕಲ್ಪಿಸುವುದು.

2 Transform(**ರೂಪಾಂತರ**) : ನಿಮ್ಮ್ ಅಗತ್ಯಗಳನ್ನು ಪೂರೈಸಲು ಡಾಟಾವನ್ನು ರೂಪಿಸಿ., ಮೂಲ ಡಾಟಾ ಬದಲಾಗದೇ ಹಾಗೆ ಉಳಿಯುತ್ತದೆ.

3 Combine(**ಸಂಯೋಜಿಸಿ**): ಡಾಟಾವನ್ನು ವಿಶಿಷ್ಟವಾಗಿ ನೊಡಲು ಅನೇಕ ಮೂಲಗಳಿಂದ ಡಾಟಾವನ್ನು ಸಂಯೋಜಿಸಿ.

4 Load (**ಲೋಡ್**) : ನಿಮ್ಮ ಪ್ರಶ್ನೆಯನ್ನು ಪೂರ್ಣಗೊಳಿಸಿ ಮತ್ತು ಅದನ್ನು ವರ್ಕ್ ಶೀಟ್‌ನಲ್ಲಿ ಅಥವಾ ಡಾಟಾ ಮಾಡೆಲ್‌ನಲ್ಲಿ ಲೋಡ್ ಮಾಡಿ ಮತ್ತು ನಿಯಮಿತವಾಗಿ ಅದನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಿ.

Connect(**ಸಂಪರ್ಕ**) : ಒಂದು ಡಾಟಾ ಸೋರ್ಸ್‌ನ್ನು ಆಮದು ಮಾಡಿಕೊಳ್ಳಲು ನೀವು ಪವರ್ ಸೋರ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ, ಒಂದು ಎಕ್ಸೆಲ್ ಶೀಟ್, ಅಥವಾ ಅನೇಕ ಡಾಟಾಬೇಸ್‌ಗಳಿಗೆ , ಫೀಡ್‌ಗಳಿಗೆ ಅಥವಾ ಕ್ಲೌಡ್‌ನಾದ್ಯಂತ ಹರಡಿರುವ ಸೇವೆಗಳು. ಡಾಟಾ ಸೋರ್ಸ್‌ಗಳು ವೆಬ್‌ನಲ್ಲಿರುವ, ಫೈಲ್‌ಗಳಲ್ಲಿರುವ, ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿರುವ ಅಝುರೆನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ ಅಥವಾ ಪ್ರಸ್ತುತ ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿರುವ ಎಕ್ಸೆಲ್ ಟೇಬಲ್‌ನ ಡಾಟಾವನ್ನು ಸಹ ಒಳಗೊಂಡಿರುತ್ತದೆ. ಪವರ್ ಕ್ವೆರಿಯೊಂದಿಗೆ , ನೀವು ನಿಮ್ಮ ಸ್ವಂತ transformations and combinations ಉಪಯೋಗಿಸಿ ಆ ಎಲ್ಲ ಡಾಟಾ ಸೋರ್ಸ್‌ಗಳನ್ನು ಒಟ್ಟಿಗೆ ತರಬಹುದು. ಇದು ನೀವು ನೋಡದೇ ಇರುವ ಒಳನೋಟಗಳನ್ನು ಬಹಿರಂಗಪಡಿಸುತ್ತದೆ.

ಹೊರಗಿನ ಡಾಟಾ ಸೋರ್ಸ್‌ನಿಂದ ಆಮದು ಮಾಡಿಕೊಂಡ ನಂತರ ನೀವು ಡಾಟಾವನ್ನು ಸೇರ್ಪಡೆಗೊಳಿಸಲು, ಬದಲಾಯಿಸಲು ಮತ್ತು ಡಿಲೀಟ್ ಮಾಡಲು ಅದನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಬಹುದು. ಹೆಚ್ಚಿನ

© NIMI NOT TO BE REPUBLISHED

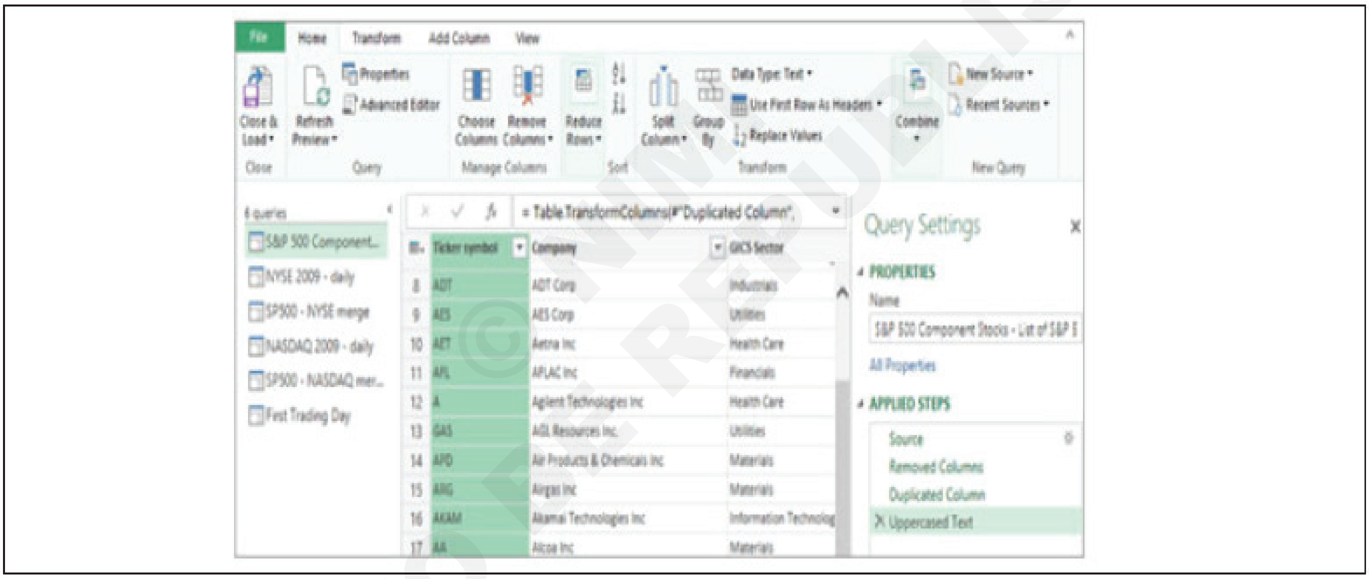
ಮಾಹಿತಿಗಾಗಿ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ Refresh an external data connection ನ್ನು ನೋಡಿರಿ.

ರೂಪಾಂತರ (Transform)

ಡಾಟಾವನ್ನು ರೂಪಾಂತರ ಮಾಡುವುದು ಎಂದರೆ, ನಿಮ್ಮ ಡಾಟಾ ವಿಶ್ಲೇಷಣೆಯ ಅಗತ್ಯಗಳನ್ನು ಪೂರೈಸಲು ಅದನ್ನು ಕೆಲವು ರೀತಿಯಲ್ಲಿ ಮಾರ್ಪಾಡು ಮಾಡುವುದು. ಉದಾಹರಣೆಗೆ ನೀವು ಕಾಲಮ್ ಅನ್ನು ತೆಗೆದುಹಾಕಬಹುದು, ಡಾಟಾಟೈಪನ್ನು ಬದಲಾಯಿಸಬಹುದು, ಅಥವಾ ರೋ ಗಳನ್ನು ಫಿಲ್ಟರ್ ಮಾಡಬಹುದು. ಇದರಲ್ಲಿನ ಪ್ರತಿಯೊಂದು ಕಾರ್ಯವು ರೂಪಾಂತರ (transformation) ಆಗಿದೆ. ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಡಾಟಾದ ಸೆಟ್‌ಗಳ ಮೇಲೆ transformation ನ್ನು (ಮತ್ತು combining) ಅನ್ವಯಿಸುವ ಪ್ರಕ್ರಿಯೆಗೆ ಶೇಪಿಂಗ್ ಡಾಟಾ (shaping data) ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ.

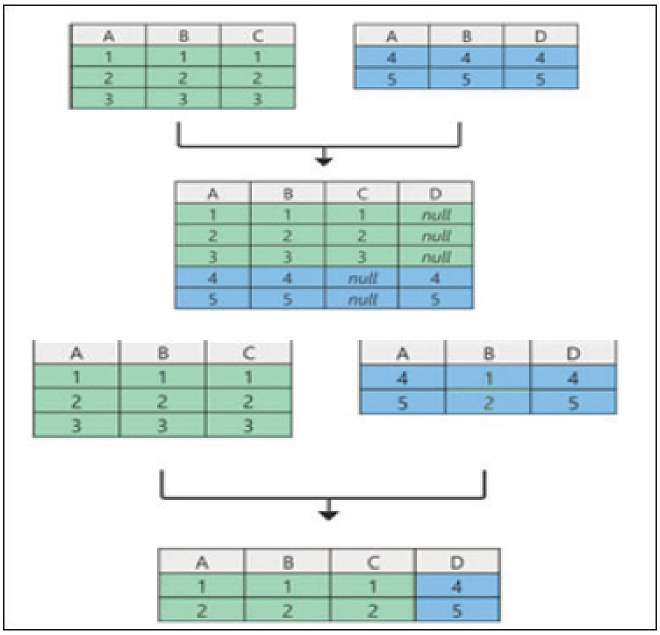
ಈ ರೀತಿಯಾಗಿ ಯೋಚಿಸಿ. ಒಂದು ಹೂದಾನಿ ಜೇಡಿಮಣ್ಣಿನ ಮುದ್ದೆಯಿಂದ ತಯಾರಾಗಲು ಪ್ರಾರಂಭಿಸುತ್ತದೆ ಮತ್ತು ಪ್ರಾಯೋಗಿಕ ಮತ್ತು ಸುಂದರ ಆಕಾರವನ್ನು ಪಡೆಯುತ್ತದೆ. ಡಾಟಾ ಸಹ ಇದೇ ರೀತಿ ಆಕಾರ ಪಡೆಯುತ್ತದೆ. ನಿಮ್ಮ ಅಗತ್ಯಗಳಿಗೆ ಸೂಕ್ತವಾದ ಮತ್ತು ಆಕರ್ಷಕ ವರದಿಗಳನ್ನು ಮತ್ತು ಡ್ಯಾಶ್‌ಬೋರ್ಡ್‌ಗಳನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುವ ಟೇಬಲ್‌ನ್ನು ಶೇಪ್ ಮಾಡುವ (ರೂಪಿಸುವ) ಅಗತ್ಯವಿದೆ.

ಪವರ್ ಕ್ವೆರಿ , ಪವರ್ ಕ್ವೆರಿ ಎಡಿಟರ್ ಎನ್ನುವ ಮೀಸಲಾದ ವಿಂಡೊವನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಡಾಟಾ ರೂಪಾಂತರಗಳನ್ನು ಸುಗಮಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಪವರ್ ಕ್ವೆರಿ ಎಡಿಟರ್‌ನ್ನು Get & Transform Data group ನಲ್ಲಿರುವ, Get Data command ದಿಂದ Launch Query Editor ನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ಮೂಲಕ ತೆರೆಯಬಹುದು. ಇದು ನೀವು ಡಾಟಾ ಸೋರ್ಸ್‌ಗೆ ಸಂಪರ್ಕಿಸಿ, ಹೊಸ ಕ್ವೆರಿಯನ್ನು ರಚಿಸಿದಾಗ ಅಥವಾ ಒಂದು ಕ್ವೆರಿಯನ್ನು ಲೋಡ್ ಮಾಡಿದಾಗ ಸಹ ತೆರೆಯುತ್ತದೆ.



ನೀವು ಡಾಟಾದೊಂದಿಗೆ ಮಾಡುವ ಎಲ್ಲ ಕೆಲಸಗಳನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡುತ್ತದೆ. ಎಂದರೆ ಪ್ರತಿಯೊಂದು ರೂಪಾಂತರವನ್ನು ರೆಕಾರ್ಡಿಂಗ್ ಮತ್ತು ಲೇಬಲ್ ಮಾಡುವುದು ಅಥವಾ ನೀವು ಡಾಟಾಗೆ ಅನ್ವಯಿಸುವ ಹಂತವನ್ನು ಟ್ರ್ಯಾಕ್ ಮಾಡುವುದು ಇತ್ಯಾದಿ. ಡಾಟಾ ರೂಪಾಂತರವು ಡಾಟಾ ಸಂಪರ್ಕ ಆಗಿರಬಹುದು, ಕಾಲಂನ್ನು ತೆಗೆದು ಹಾಕುವುದಾಗಿರಬಹುದು, ಮರ್ಜ್ (ವಿಲೀನ) , ಅಥವಾ ಡಾಟಾಟೈಪ್ ಬದಲಾವಣೆ ಆಗಿರಬಹುದು, ನೀವು ಪ್ರತಿಯೊಂದು ರೂಪಾಂತರವನ್ನು ನೋಡಬಹುದು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು. ಇದನ್ನು ನೀವು Query Settings pane. ನ ವಿಭಾಗದಲ್ಲಿರುವ APPLIED STEPS ನಲ್ಲಿ ಮಾಡಬಹುದು.

ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್‌ನಿಂದ ನೀವು ಹಲವಾರು ರೂಪಾಂತರಗಳನ್ನು ಮಾಡಬಹುದು. ಪ್ರತಿಯೊಂದು ರೂಪಾಂತರವೂ ಬ್ಯಾಕ್ ಗ್ರೌಂಡ್‌ನಲ್ಲಿ ಒಂದು ಹಂತವಾಗಿ ರೆಕಾರ್ಡ್



NOT TO BE REPUBLISHED © NIMI

ಆಗಿರುತ್ತದೆ. Advanced Editor ನಲ್ಲಿರುವ Power Query M Language ಉಪಯೋಗಿಸಿ ನೀವು ನಿಮ್ಮ ಸ್ವಂತ ಹಂತಗಳನ್ನು ಸಹ ಮಾರ್ಪಡಿಸಬಹುದು ಮತ್ತು ಬರೆಯಬಹುದು.

ನೀವು ನಿಮ್ಮ ಡಾಟಾ ಸಂಪರ್ಕಗಳಿಗೆ ಅನ್ವಯಿಸುವ ಎಲ್ಲ ರೂಪಾಂತರಗಳು ಒಟ್ಟಾಗಿ ಒಂದು ಕ್ವೆರಿಯನ್ನು ರೂಪಿಸುತ್ತವೆ. ಇದು ಮೂಲ (ಮತ್ತು ಬದಲಾಗದ) ಡಾಟಾ ಸೋರ್ಸ್‌ನ ಹೊಸ ಪ್ರತಿನಿಧಿಯಾಗಿರುತ್ತದೆ. ನೀವು ಕ್ವೆರಿಯನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡಿದಾಗ, ಪ್ರತಿಯೊಂದು ಹಂತವು ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ(run). ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಕ್ವೆರಿಗಳು, ಮ್ಯಾನ್ಯುಯೆಲ್ (ಹಸ್ತಚಾಲಿತ) ಆಗಿ ಡಾಟಾವನ್ನು ಸಂಪರ್ಕಿಸುವುದು ಮತ್ತು ಶೇಪ್ ಮಾಡುವ ಅವಶ್ಯಕತೆಯನ್ನು ಬದಲಾಯಿಸುತ್ತವೆ.

Combine (ಸಂಯೋಜಿಸುವುದು): ನೀವು ನಿಮ್ಮ ಎಕ್ಸೆಲ್ ವರ್ಕ್ ಬುಕ್‌ನಲ್ಲಿ ಕ್ವೆರಿಗಳನ್ನು ಅಪೆಂಡ್ ಅಥವಾ ಮರ್ಜ್ ಮಾಡುವುದರ ಮೂಲಕ ಹಲವಾರು ಕ್ವೆರಿಗಳನ್ನು ಸಂಯೋಜಿಸಬಹುದು.ಟ್ಯುಬುಲರ್ ಆಕಾರವನ್ನು ಹೊಂದಿರುವ ಯಾವುದೇ ಕ್ವೆರಿ ಮೇಲೆ ಅಪೆಂಡ್ ಮತ್ತು ಮರ್ಜ್ ಕೆಲಸಗಳನ್ನು ಮಾಡಬಹುದು ಮತ್ತು ಇವುಗಳು ಉಪಯೋಗಿಸುವ ಡಾಟಾ, ಡಾಟಾಸೋರ್ಸ್‌ಗಳಿಂದ ಸ್ವತಂತ್ರ್ಯವಾಗಿರುತ್ತದೆ.

Append (ಸೇರಿಸುವುದು) : ಅಪೆಂಡ್ ಕೆಲಸವು ಹೊಸ ಕ್ವೆರಿಯನ್ನು ತಯಾರಿಸುತ್ತದೆ. ಈ ಕ್ವೆರಿಯು ಮೊದಲನೇ ಕ್ವೆರಿಯ ಎಲ್ಲ ರೋಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ, ನಂತರ ಎರಡನೇ ಕ್ವೆರಿಯ ಎಲ್ಲ ರೋಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ನೀವು ಎರಡು ರೀತಿಯ ಅಪೆಂಡ್ ಕೆಲಸಗಳನ್ನು ಮಾಡಬಹುದು :

- Intermediate Append: ಪ್ರತಿಯೊಂದು ಅಪೆಂಡ್ ಕೆಲಸಕ್ಕೆ ಹೊಸ ಕ್ವೆರಿಯನ್ನು ರಚಿಸುತ್ತದೆ.
- Inline Append: ನೀವು ಅಂತಿಮ ಫಲಿತಾಂಶವನ್ನು ತಲುಪುವವರೆಗೆ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಕ್ವೆರಿಗೆ ಡಾಟಾವನ್ನು ಅಪೆಂಡ್ ಮಾಡುತ್ತದೆ.

Merge(ವಿಲೀನಗೊಳಿಸುವುದು) : ಮರ್ಜ್ ಕಾರ್ಯವು ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಎರಡು ಕ್ವೆರಿಗಳಿಂದ ಒಂದು ಹೊಸ ಕ್ವೆರಿಯನ್ನು ರಚಿಸುತ್ತದೆ. ಈ ಒಂದು ಕ್ವೆರಿಯು ಒಂದು ಪ್ರೈಮರಿ ಟೇಬಲ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಕಾಲಂಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದರಲ್ಲಿ ಒಂದು ಕಾಲಂ, ಸಂಬಂಧಪಟ್ಟ ಟೇಬಲ್‌ಗೆ ನ್ಯಾವಿಗೇಷನ್ ಲಿಂಕ್ ಆಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ. ಸಂಬಂಧಿತ ಟೇಬಲ್, ಪ್ರೈಮರಿ ಟೇಬಲ್‌ನ ಸಾಮಾನ್ಯ ಕಾಲಂ ಮೌಲ್ಯದಿಂದ ಪ್ರತಿಯೊಂದು ರೋಗೆ ಹೊಂದಿಕೆಯಾಗುವ ಎಲ್ಲ ರೋ ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದಲ್ಲದೇ ಸಂಬಂಧಿತ ಟೇಬಲ್‌ದಿಂದ ಪ್ರೈಮರಿ ಟೇಬಲ್‌ಗೆ ನೀವು ಕಾಲಂಗಳನ್ನು ವಿಸ್ತರಿಸಬಹುದು ಅಥವಾ ಸೇರಿಸಬಹುದು.

ಲೋಡ್ ಮಾಡುವುದು (Load)

ನಿಮ್ಮ ವರ್ಕ್‌ಬುಕ್‌ಗೆ ಕ್ವೆರಿಗಳನ್ನು ಲೋಡ್ ಮಾಡಲು ಎರಡು ಮುಖ್ಯ ಮಾರ್ಗಗಳಿವೆ :

- Power Query Editor ದಿಂದ - Home Tab ನಲ್ಲಿ Close ಗ್ರೂಪ್‌ಲ್ಲಿರುವ, Close ಮತ್ತು Load ಕಮಾಂಡ್‌ಗಳನ್ನು ನೀವು ಉಪಯೋಗಿಸಬಹುದು.
- Excel Workbook Queries ಪೇನ್ ದಿಂದ (Select Queries & Connections), ನೀವು ಒಂದು ಕ್ವೆರಿಯನ್ನು ಮೌಸ್‌ನ ಬಲ ಬಟನ್ ಕ್ಲಿಕ್ ಮಾಡಿ ಮತ್ತು Load To ಆಯ್ಕೆ ಮಾಡಿರಿ.

ನೀವು ನಿಮ್ಮ ಲೊಡ್ ಆಯ್ಕೆಗಳನ್ನು Query Options ಡಾಯಾಲಾಗ್ ಬಾಕ್ಸ್ ಬಳಸುವುದರ ಮೂಲಕ ಉತ್ತಮಗೊಳಿಸಬಹುದು. (File > Options ಮತ್ತು settings > Query Options) ಇದರಲ್ಲಿ ನೀವು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಹೇಗೆ ನೋಡಲು ಬಯಸುತ್ತೀರಿ ಮತ್ತು ಡಾಟಾವನ್ನು ಎಲ್ಲಿ ಲೋಡ್ ಮಾಡಲು ಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ನೀವು ಡಾಟಾವನ್ನು ವರ್ಕ್‌ಶೀಟ್‌ನಲ್ಲಿಯೂ ಲೋಡ್ ಮಾಡಬಹುದು ಅಥವಾ ಡಾಟಾ ಮಾಡೆಲ್‌ನಲ್ಲಿಯೂ (ಇದು ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿರುವ ಅನೇಕ ಟೇಬಲ್‌ಗಳನ್ನು ಒಳಗೊಂಡ ರಿಲೇಷನಲ್ ಡಾಟಾಸೋರ್ಸ್ ಆಗಿದೆ) ಮಾಡಬಹುದು.

ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿಗಾಗಿ ಅಪ್ಲಿಕೇಷನ್ ಬೆಂಬಲ(Application support for Power Query in Excel)

ಹತ್ತು ವರ್ಷಗಳಿಂದ ವಿಂಡೋಸ್‌ಗಾಗಿ ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿಯನ್ನು ಬೆಂಬಲಿಸಲಾಗುತ್ತಿದೆ. ಈಗ ಎಕ್ಸೆಲ್ ಪವರ್ ಕ್ವೆರಿಯ ಬೆಂಬಲವನ್ನು Mac ಗೆ ವಿಸ್ತರಿಸುತ್ತಿದೆ ಮತ್ತು ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ವೆಬ್‌ಗಾಗಿ ಬೆಂಬಲವನ್ನು ಕೊಡುತ್ತಿದೆ. ಇದರ ಅರ್ಥ ನಾವು ಪವರ್ ಕ್ವೆರಿಯನ್ನು ಮೂರು ಪ್ರಮುಖ ವೇದಿಕೆಗಳಲ್ಲಿ ಲಭ್ಯವಾಗುವಂತೆ ಮಾಡುತ್ತಿದ್ದೇವೆ ಮತ್ತು ಎಕ್ಸೆಲ್ ಗ್ರಾಹಕರಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿಯ ಜನಪ್ರೀಯತೆಯನ್ನು ಮತ್ತು ಕ್ರಿಯಾತ್ಮಕತೆಯನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತದೆ. ಮೈಕ್ರೊಸಾಫ್ಟ್ 365 ಮಾರ್ಗಸೂಚಿಯ ಭವಿಷ್ಯದ ಪ್ರಕಟಣೆಗಳಿಗಾಗಿ ವೀಕ್ಷಿಸಿ ಮತ್ತು ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಮೈಕ್ರೊಸಾಫ್ಟ್ 365 ನಲ್ಲಿ ಹೊಸದೇನಿದೆ ಎಂದು ವೀಕ್ಷಿಸಿ.

ಎಕ್ಸೆಲ್‌ನಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿಯ ಇತಿಹಾಸ (A history of Power Query in Excel)

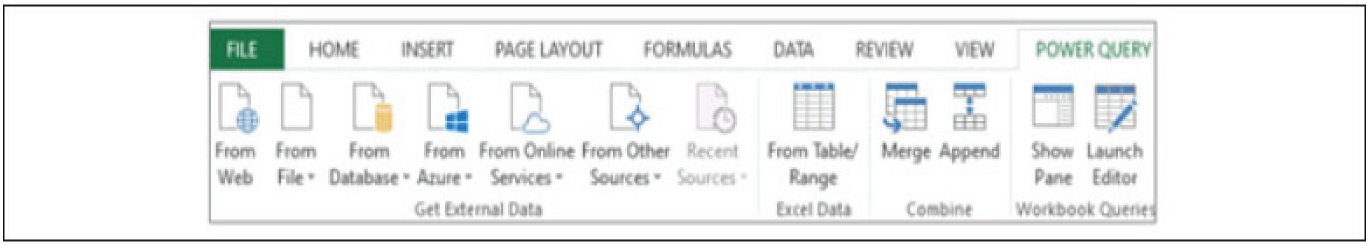
Get & Transform Data ನ ಏಕೀಕರಣವು (ಈಗ Power Query ಎಂದು ಕರೆಯುತ್ತೇವೆ) ಕೆಲವು ವರ್ಷಗಳಿಂದ ವಿಕ್ಸೆಲ್‌ನಲ್ಲಿ ಹಲವಾರು ಬದಲಾವಣೆಗಳನ್ನು ತಂದಿದೆ.

ವಿಂಡೋಸ್‌ಗಾಗಿ ಎಕ್ಸೆಲ್ 2010 ಮತ್ತು 2013 (Excel 2010 and 2013 for Windows)

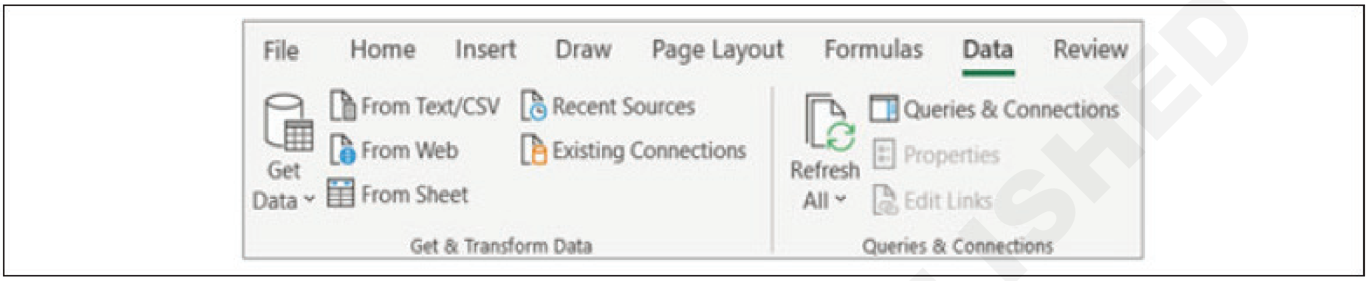
Excel 2010 for Windows ನಲ್ಲಿ, ನಾವು ಮೊದಲು ಪವರ್ ಕ್ವೆರಿಯನ್ನು ಪರಿಚಯಿಸಿದ್ದೇವೆ ಮತ್ತು ಇದು ಉಚಿತ ಆ್ಯಡ್- ಇನ್

© NIMI NOT TO BE REPUBLISHED

ಆಗಿ ಲಭ್ಯವಿತ್ತು ಮತ್ತು ಇಲ್ಲಿಂದ ಡೌನ್‌ಲೋಡ್ ಮಾಡಬಹುದಾಗಿದೆ : Download the Power Query addin. ಒಮ್ಮೆ ಸಕ್ರೀಯಗೊಳಿಸಿದರೆ, ಪವರ್ ಕ್ವೆರಿ ಕಾರ್ಯವು ರಿಬ್ಬನ್‌ನಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿ ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಲಭ್ಯವಿತ್ತು.



Microsoft 365 : ನಾವು ಪವರ್ ಕ್ವೆರಿಯನ್ನು ಡಾಟಾವನ್ನು ಆಮದು ಮಾಡಿಕೊಳ್ಳುವ ಮತ್ತು ಸ್ವಚ್ಛಗೊಳಿಸುವ ಸಲುವಾಗಿ ಪ್ರಾಥಮಿಕ ಅನುಭವ ಪಡೆದುಕೊಳ್ಳಲು ನವೀಕರಿಸಿದ್ದೇವೆ. ನೀವು ಪವರ್ ಕ್ವೆರಿ ಡಾಟಾ ಇಂಪೋರ್ಟ್ ವಿಝಾರ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಸಾಧನಗಳನ್ನು ಎಕ್ಸೆಲ್ ರಿಬ್ಬನ್‌ನಲ್ಲಿರುವ Data tab ನ, Get & Transform Data ಗ್ರೂಪ್‌ನಲ್ಲಿ ಪಡೆದುಕೊಳ್ಳಬಹುದು.



ಈ ಅನುಭವವು ವರ್ಧಿತ ಡಾಟಾ ಆಮದು ಕಾರ್ಯವನ್ನು (enhanced data import functionality), Data ಟ್ಯಾಬ್‌ನಲ್ಲಿ ಕಮಾಂಡ್‌ಗಳನ್ನು ಮರುಹೊಂದಿಸುವುದು, ಹೊಸ Queries & Connection side pane ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ ಮತ್ತು ಸಾರ್ಟ್ (sort) ಮಾಡುವುದು, ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವುದು, ಕಾಲಂಗಳನ್ನು ವಿಭಜಿಸುವುದು, ಡಾಟಾವನ್ನು ಒಟ್ಟುಗೂಡಿಸುವುದು ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ಈ ಹೊಸ ಅನುಭವವು ಹಳೆಯ, ಪರಂಪರೆಯ Get External Data ಗ್ರೂಪ್‌ನಲ್ಲಿರುವ Data command ನ ಅಡಿಯಲ್ಲಿರುವ data import wizards ಗಳನ್ನು ಬದಲಾಯಿಸಿದೆ. ಅದಾಗ್ಯೂ, ಅವುಗಳನ್ನು ಇನ್ನೂ ಸಹ Excel Options ಡಾಯಲಾಗ್ ಬಾಕ್ಸ್‌ನಿಂದ ಪಡೆದುಕೊಳ್ಳಬಹುದು (File > Options > Data > Show legacy data import wizards ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಿರಿ.).

ವಿಂಡೋಸ್‌ಗಾಗಿ Excel 2016 **ಮತ್ತು** 2019 (Excel 2016 and 2019 for Windows)

Microsoft 365 for Mac ನಂತೆಯೆ ಪವರ್ ಕ್ವೆರಿ ತಂತ್ರಜ್ಞಾನ ಆಧಾರಿತ Get & Transform Data ದ ಅದೇ ಅನುಭವವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ.

Microsoft 365 for Mac **ಗಾಗಿ ಎಕ್ಸೆಲ್** (Excel for Microsoft 365 for Mac)

2019 ರಿಂದ ನಾವು Excel for Mac ನಲ್ಲಿ ಪವರ್ ಕ್ವೆರಿಯನ್ನು ಬೆಂಬಲಿಸುವ ಒಂದು ಪ್ರಯಾಣವನ್ನು ಪ್ರಾರಂಭಿಸಿದ್ದೇವೆ. ಅಂದಿನಿಂದ TXT, CSV, XLSX, JSON ಮತ್ತು XML ಫೈಲುಗಳಿಂದ Power Query ಕ್ವೆರಿಗಳನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ. ನಾವು SQL ಸರ್ವರ್‌ನಿಂದ, ಟೇಬಲ್‌ಗಳಿಂದ, ಮತ್ತು ಪ್ರಸ್ತುತ ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿರುವ ರೇಂಜ್‌ಗಳಿಂದ ಸಹ ಡಾಟಾವನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ.

2019 ರ October ನಲ್ಲಿ, ಈಗಾಗಲೆ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ Power Query ಕ್ವೆರಿಗಳನ್ನು ರಿಫ್ರೆಶ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ ಮತ್ತು ಹೊಸ ಕ್ವೆರಿಗಳನ್ನು ರಚಿಸಲು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಲು VBA ಅನ್ನು ಉಪಯೋಗಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ.

2021 ರ ಜನೇವರಿಯಲ್ಲಿ, Power Query ಕ್ವೆರಿಗಳನ್ನು OData and SharePoint sources ಗಳಿಂದ ರಿಫ್ರೆಶ್ ಮಾಡಲು ಬೆಂಬಲವನ್ನು ಸೇರಿಸಿದ್ದೇವೆ.

ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ, Use Power Query in Excel for Mac ಅನ್ನು ನೋಡಿರಿ.

ಸೂಚನೆ :Excel 2016 **ಮತ್ತು** Excel 2019 for Mac. **ನಲ್ಲಿ** Power Query **ಗೆ ಬೆಂಬಲ ಇರುವುದಿಲ್ಲ**.

ಡಾಟಾ ಕ್ಯಾಟಲಾಗ್ ಅಸಮ್ಮತಿ (Data Catalog deprecation)

Data Catalog ನಲ್ಲಿ, ನೀವು ಹಂಚಿಕೊಂಡ (share)ನಿಮ್ಮ ಕ್ವೆರಿಗಳನ್ನು ನೋಡಬಹುದು, ಮತ್ತು ನಂತರ ಅವುಗಳನ್ನು ಲೋಡ್ ಮಾಡಲು, ಎಡಿಟ್ ಮಾಡಲು, ಅಥವಾ ಪ್ರಸ್ತುತ ವರ್ಕ್‌ಬುಕ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ಈ ವೈಶಿಷ್ಟ್ಯವನ್ನು ಕ್ರಮೇಣವಾಗಿ ತಡೆಹಿಡಿಯಲಾಯಿತು :

- August 1st, 2018 ರಂದು , ನಾವು ಹೊಸ ಗ್ರಾಹಕರನ್ನು Data Catalog ಗೆ ಪ್ರವೇಶಿಸುವುದನ್ನು ತಡೆದಿದ್ದೇವೆ.

© NIMI NOT TO BE REPUBLISHED

- December 3rd, 2018 ರಂದು, ಬಳಕೆದಾರರು ಹೊಸ ಅಥವಾ ನವೀಕರಿಸಿದ ಕ್ವೆರಿಗಳನ್ನು Data Catalog ನಲ್ಲಿ ಹಂಚಿಕೊಳ್ಳಲಾಗಲಿಲ್ಲ.
- March 4th, 2019 ರಂದು, Data Catalog ಕಾರ್ಯನಿರ್ವಹಿಸುವುದನ್ನು ನಿಲ್ಲಿಸಿತು. ಈ ದಿನಾಂಕದ ನಂತರ, ನೀವು ಡೌನ್‌ಲೋಡ್ ಮಾಡಿದ ನಿಮ್ಮ ಶೇರ್ಡ್ (ಹಂಚಿಕೊಳ್ಳುವ) ಕ್ವೆರಿಗಳನ್ನು Data Catalog ಹೊರಗೆ ಬಳಸುವುದನ್ನು ಮುಂದುವರೆಸುವಂತೆ ಶಿಫಾರಸು ಮಾಡಲಾಯಿತು. ಇದಕ್ಕಾಗಿ My Data Catalog Queries task pane ನಿಂದ Open ಆಯ್ಕೆಯನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

Power Query add-in deprecation

2019 ರ ಬೇಸಿಗೆಯ ಆರಂಭದಲ್ಲಿ, ನಾವು ಅಧಿಕೃತವಾಗಿ Power Query add-in ಅನ್ನು ಅಸಮ್ಮತಿಗೊಳಿಸಿದ್ದೇವೆ. ಇದು Windows ನ Excel 2010 ಮತ್ತು 2013 ಗಳಿಗೆ ಅವಶ್ಯಕವಾಗಿತ್ತು. ಸೌಜನ್ಯಕ್ಕಾಗಿ, ನೀವು ಇನ್ನೂ add- in ಅನ್ನು ಬಳಸಬಹುದು. ಆದರೆ ಇದು ನಂತರದ ದಿನಾಂಕಗಳಲ್ಲಿ ಬದಲಾಗಬಹುದು

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಉತ್ಪನ್ನಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು ಗುರುತಿಸಲು ಇ - ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳನ್ನು (Browse e-Commerce Sites to Identify Products & Services)

ಇ - ಕಾಮರ್ಸ್‌ನ ವ್ಯಾಪ್ತಿ ಮತ್ತು ಲಾಭಗಳು (E-commerce scope and benefits)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- E-commerce ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿ
- ಸಾಂಪ್ರದಾಯಿಕ ಮತ್ತು ಇ - ಕಾಮರ್ಸ್ ನಡುವಿನ ವ್ಯತ್ಯಾಸವನ್ನು ವಿವರಿಸಿ.
- ಇ -ಕಾಮರ್ಸ್‌ನ ಪ್ರಕಾರಗಳು, ವ್ಯಾಪ್ತಿ ಮತ್ತು ಲಾಭಗಳನ್ನು ವಿವರಿಸಿ.

ಇ- ಕಾಮರ್ಸ್ (E Commerce)

ಇ ಕಾಮರ್ಸ್ ಎನ್ನುವುದು ಉತ್ಪನ್ನಗಳನ್ನು ಅಥವಾ ಸೇವೆಗಳನ್ನು ಖರೀದಿಸುವ ಮತ್ತು ಮಾರಾಟ ಮಾಡುವ ಮತ್ತು ವಿದ್ಯುನ್ಮಾನದ ಮೂಲಕ ಹಣವನ್ನು ವರ್ಗಾವಣೆ ಮಾಡುವುದು ಒಂದು ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ.

ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಕಾಮರ್ಸ್ ತಂತ್ರಜ್ಞಾನಗಳನ್ನು ಸೆಳೆಯುತ್ತದೆ ಉದಾಹರಣೆಗೆ ಮೊಬೈಲ್ ಕಾಮರ್ಸ್, ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಹಣ ವಾರ್ಗಾವಣೆ , ಸಪ್ಲೈ ಚೈನ್ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್, ಇಂಟರ್‌ನೆಟ್ ಮಾರ್ಕೆಟಿಂಗ್, ಆನ್‌ಲೈನ್ ಟ್ರಾನ್ಸ್ಯಾಕ್ಷನ್ ಪ್ರೊಸೆಸಿಂಗ್, ಇಲಕ್ಟ್ರಾನಿಕ್ ಡಾಟಾ ಇಂಟರ್‌ಚೇಂಜ್ (EDI) ಇನ್ವೆಂಟರಿ ಮ್ಯಾನೆಜ್‌ಮೆಂಟ್ ಸಿಸ್ಟಮ್‌ಗಳು, ಮತ್ತು ಆಟೋಮೇಟೆಡ್ ಡಾಟಾ ಕಲೆಕ್ಷನ್ ಸಿಸ್ಟಮ್‌ಗಳು . ಆಧುನಿಕ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಕಾಮರ್ಸ್ ಸಾಮಾನ್ಯವಾಗಿ ವಹಿವಾಟಿನ ಜೀವನ ಚಕ್ರದ ಕನಿಷ್ಟ ಒಂದು ಭಾಗದಲ್ಲಿಯಾದರೂ World Wide Web ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಇ ಮೇಲ್‌ನಂತಹ ಇತರ ತಂತ್ರಜ್ಞಾನವನ್ನೂ ಸಹ ಉಪಯೋಗಿಸಬಹುದು.

ಇ ಕಾಮರ್ಸ್ ವ್ಯವಹಾರಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಕೆಳಗಿನ ಕೆಲವು ಅಥವಾ ಎಲ್ಲ ಅಭ್ಯಾಸಗಳನ್ನೂ ಬಳಸಿಕೊಳ್ಳುತ್ತವೆ :

ವೆಬ್ ಸೈಟ್‌ಗಳಲ್ಲಿ Etail ಅಥವಾ "virtual storefronts" ಗಳನ್ನು ಆನ್‌ಲೈನ್ ಕ್ಯಾಟಲಾಗ್‌ಗಳೊಂದಿಗೆ ಒದಗಿಸುತ್ತವೆ. ಕೆಲವೊಮ್ಮೆ "virtual mall" ಆಗಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ.

ವೆಬ್‌ಸೈಟ್‌ಗಳಲ್ಲಿ ಅಥವಾ ಆನ್‌ಲೈನ್ ಮಾರುಕಟ್ಟೆ ಸ್ಥಳಗಳಲ್ಲಿ ಖರೀದಿಸಿ ಅಥವಾ ಮಾರಾಟ ಮಾಡಿರಿ.

ವೆಬ್ ಸಂಪರ್ಕಗಳ ಮೂಲಕ ಮತ್ತು ಸಾಮಾಜಿಕ ಮಾಧ್ಯಮದ ಮೂಲಕ ಜನಸಂಖ್ಯಾ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿರಿ ಮತ್ತು ಉಪಯೋಗಿಸಿರಿ.

ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಡಾಟಾ ಇಂಟರ್‌ಚೇಂಜ್‌ನ್ನು ಉಪಯೋಗಿಸಿರಿ. ಇದು ವ್ಯಾಪಾರದಿಂದ ವ್ಯಾಪಾರಕ್ಕೆ (business-to-business) ಡಾಟಾದ ವಿನಿಮಯವಾಗಿದೆ.

ಇ- ಮೇಲ್ ಅಥವಾ ಫ್ಯಾಕ್ಸ್ ಮೂಲಕ ನಿರೀಕ್ಷಿತ ಮತ್ತು ಸ್ಥಾಪಿತ ಗ್ರಾಹಕರನ್ನು ತಲುಪಿ. (ಉದಾಹರಣೆಗೆ , ನ್ಯೂಸ್‌ಲೆಟರ್‌ಗಳು)

business-to-business ಖರೀದಿಸುವಿಕೆ ಮತ್ತು ಮಾರಾಟ ಮಾಡುವಿಕೆಯನ್ನು ಉಪಯೋಗಿಸಿ.

ಸುರಕ್ಷಿತ ವ್ಯಾಪಾರ ವಹಿವಾಟುಗಳನ್ನು ಒದಗಿಸಿ.

ಅಕ್ಟೋಬರ್ 6, 2014 ರಂದು ನಡೆದ ಫ್ಲಿಪ್‌ಕಾರ್ಟ್‌ನ ಬಿಗ್ ಬಿಲಿಯನ್ ಡೇ ಸೇಲ್‌ಅನ್ನು ಎಲ್ಲ ಆನ್‌ಲೈನ್ ಮಾರಾಟಗಳ ತಾಯಿ ಎಂದು ಹೇಳಲಾಗುತ್ತದೆ. ಆದರೆ ಆ ದೊಡ್ಡ ದಿನದಂದು ವಿಷಯಗಳ ಹದಗೆಟ್ಟಿದ್ದವು ಮತ್ತು ಗ್ರಾಹಕರು ಶುಷ್ಕ ಭಾವನೆಯನ್ನು ಅನುಭವಿಸಿದರು. ಸಂಸ್ಥಾಪಕರು ತಕ್ಷಣವೇ ಕ್ಷಮೆ ಯಾಚಿಸಿದರು. ಅವರು ಸಮರ್ಪಕವಾಗಿ ತಾಯಾರಾಗಿರಲಿಲ್ಲ ಎಂಬುದು ಸತ್ಯ. ನಮ್ಮದು ಇ ಕಾಮರ್ಸ್‌ಗೆ ದೊಡ್ಡ ರೀತಿಯಲ್ಲಿ ಲಗ್ಗೆ ಇಡುತ್ತಿರುವ ರಾಷ್ಟ್ರವಾಗಿದೆ ಎಂಬುದು ಸತ್ಯ.

ಈ ರೀತಿಯ ಒಂದೇ ಒಂದು ಘಟನೆಯು ಬೆಳವಣಿಗೆಗೆ ಸಿದ್ಧವಾಗಿರುವ $ 3 ಬಿಲಿಯನ್ ವಲಯದಲ್ಲಿ ಆಸಕ್ತಿಯನ್ನು ಕ್ಷೀಣಿಸಲು ಹೆಚ್ಚಿನದಿನ್ನೇನೂ ಮಾಡುವುದಿಲ್ಲ. ಈಗಾಗಲೇ ಫ್ಲಿಪ್‌ಕಾರ್ಟ್‌ನ ಪ್ರತಿಸ್ಪರ್ಧಿಗಳಾದ ಅಮೇಝಾನ್ ಮತ್ತು ಸ್ನ್ಯಾಪ್‌ಡೀಲ್ ಕಂಪನಿಗಳು ಈ ಅನುಭವದಿಂದ ಕಲಿತಿರಬಹುದು ಮತ್ತು ಬಹುಶಃ ಭವಿಷ್ಯದಲ್ಲಿ ಇಂತಹ ಘಟನೆಗಳು ಮರುಕಳಿಸದಂತೆ ಪ್ರಯತ್ನಿಸುತ್ತವೆ.

Asssociate Chambers of Commerce ಮತ್ತು Industry in India (ASSOCHAM) ಗಳ ಪ್ರಕಾರ ಆನ್‌ಲೈನ್ ಶಾಪಿಂಗ್ Rs 10,000 ಕೋಟಿ ರೂಗಳನ್ನು ದಾಟಲಿದೆ . ಇದು ಕಳೆದ ವರ್ಷಗಳ ಪರಿಮಾಣಕ್ಕಿಂತ ಶೇಕಡ 350 ಕ್ಕಿಂತ ಹೆಚ್ಚಾಗಿರುತ್ತದೆ.

ಸಾಂಪ್ರದಾಯಿಕ ಕಾಮರ್ಸ್ Vs ಇ ಕಾಮರ್ಸ್ (Traditional Commerce Vs. E Commerce)

ಸಾಂಪ್ರದಾಯಿಕ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಖರೀದಿದಾರರು ಒಂದು ಉತ್ಪನ್ನವನ್ನು ಅಥವಾ ಸೇವೆಯನ್ನು ಖರೀದಿಸಲು ಅಂಗಡಿಗೆ ಹೋಗಬೇಕಾಗುತ್ತದೆ , ಆದರೆ ಇ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಉತ್ಪನ್ನವನ್ನು ಅಥವಾ ಸೇವೆಗಳನ್ನು ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿ ಪಟ್ಟಿ ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು

NOT TO BE REPUBLISHED © NIMI

ಖರೀದಿದಾರರು ವೆಬ್ ಸೈಟ್‌ನಲ್ಲಿ ನೀಡಲಾದ ಉತ್ಪನ್ನವನ್ನು ಅದರ ಫೋಟೊಗಳು ಮತ್ತು ಇತರ ವಿಶೇಷಣಗಳನ್ನು ಪರಿಶೀಲಿಸಿ ಆಯ್ಕೆ ಮಾಡಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಡೆಬಿಟ್ /ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿ ಅಥವಾ ಇಂಟರ್‌ನೆಟ್ ಬ್ಯಾಂಕಿಂಗ್ ವಿಧಾನದ ಮೂಲಕ ವಿದ್ಯುನ್ಮಾನದಿಂದ ಅದರ ಆದ್ಯತೆಯ ಬೆಲೆಯನ್ನು ಪಾವತಿಸಬೇಕಾಗುತ್ತದೆ. ನಂತರ ಕಂಪನಿಯಿಂದ ಗ್ರಾಹಕರಿಗೆ ಅವರ ಅವರ ವಿಳಾಸಕ್ಕೆ ಉತ್ಪನ್ನವನ್ನು ತಲುಪಿಸಲಾಗುತ್ತದೆ. ವಿತರಣಾ ಶುಲ್ಕ ಉಚಿತವಾಗಿರಬಹುದು ಅಥವಾ ಉತ್ಪನ್ನದ ವೆಚ್ಚದ ಪಾವತಿಯೊಂದಿಗೆ ಅದನ್ನು ಪಾವತಿಸಬಹುದು.

ಸಾಂಪ್ರದಾಯಿಕ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಗ್ರಾಹಕರು ಭೌತಿಕವಾಗಿ ಉತ್ಪನ್ನವನ್ನು ಪರಿಶೀಲಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ ಯಾರಾದರೂ ಒಂದು LED ಟಿವಿಯನ್ನು ಖರೀದಿಸುತ್ತಿದ್ದರೆ ಅವರು ಅಂಗಡಿಗೆ ಹೋಗಿ ವೈಯಕ್ತಿಕವಾಗಿ ಚಿತ್ರದ ಗುಣಮಟ್ಟವನ್ನು ಅವನು/ಅವಳು ಸ್ವತಃ ಪರಿಶೀಲಿಸಬಹುದು, ಆದರೆಇ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಇದು ಸಾಧ್ಯವಿಲ್ಲ. ಖರೀದಿದಾರರು ಉತ್ಪನ್ನದ ಫೋಟೊಗಳನ್ನು ಮಾತ್ರ ನೋಡಬಹುದು ಆದರೆ ಚಿತ್ರದ ಗುಣಮಟ್ಟ ಅಥವಾ ಧ್ವನಿಯ ಗುಣಮಟ್ಟವನ್ನು ಅವನು/ಅವಳು ಸ್ವತಃ ವೀಕ್ಷಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಇ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ, ಗ್ರಾಹಕರು ತಾವೇ ಸ್ವತಃ ಅಂಗಡಿಗೆ ಹೋಗದೇ ಇರುವುದರಿಂದ ತಮ್ಮ ಸಮಯವನ್ನು ಮತ್ತು ಹಣವನ್ನು ಉಳಿಸಬಹುದು. ಆದರೆ ಗ್ರಾಹಕರಿಗೆ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕ ಮತ್ತು ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿಸಲು ಸಾಧನಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

ಸಾಂಪ್ರದಾಯಿಕ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ, ಖರೀದಿದಾರರು ಸ್ವತಃ ಮಾರಾಟಗಾರರನ್ನು ಭೌತಿಕವಾಗಿ ಭೇಟಿಯಾಗಬಹುದು. ಮಾರಾಟದ ನಂತರ ಸೇವೆಯ ಅಗತ್ಯವಿದ್ದರೆ ಖರೀದಿದಾರರು ಮಾರಾಟಗಾರರನ್ನು ಸಂಪರ್ಕಿಸಬಹುದು ಆದರೆ ಇ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಖರೀದಿದಾರರು ಮಾರಾಟಗಾರರನ್ನು ಸ್ವತಃ ಭೌತಿಕವಾಗಿ ಭೇಟಿಯಾಗುವ ಅವಕಾಶ ಇರುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ಮಾರಾಟದ ನಂತರ ಸೇವೆಗಾಗಿ ಆಯಾ ಉತ್ಪನ್ನದ ಸೇವಾ ಕೇಂದ್ರಗಳನ್ನು ಸಂಪರ್ಕಿಸುವ ಮೂಲಕ ಗ್ರಾಹಕರು ಸೇವೆಯ ವ್ಯವಸ್ಥೆಯನ್ನು ಮಾಡಿಕೊಳ್ಳಬೇಕಾಗುತ್ತದೆ.

ಸಾಂಪ್ರದಾಯಿಕ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಉತ್ಪನ್ನದ ಬೆಲೆಗಳು ಒಂದು ಮುಖ್ಯ ಅಂಶವಾಗಿದೆ. ಇದರಲ್ಲಿ ಉತ್ಪನ್ನಗಳನ್ನು ಅ ಂಗಡಿಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಅಂಗಡಿಯನ್ನು ನಡೆಸಲು ಉದ್ಯೋಗಿಗಳನ್ನು ನೇಮಕ ಮಾಡಿಕೊಳ್ಳಬೇಕು. ಇಲ್ಲಿ ಅಂಗಡಿಗಳು ಮಾರಾಟಗಾರರ ಮಾಲೀಕತ್ವದಲ್ಲಿರಬೇಕು ಅಥವಾ ಅವುಗಳನ್ನು ಬಾಡಿಗೆಗೆ ತೆಗೆದುಕೊಳ್ಳಬೇಕಾಗುವದರಿಂದ ಇದರ ಬೆಲೆ ಹೆಚ್ಚಾಗಿರುತ್ತದೆ. ಇ ಕಾಮರ್ಸ್‌ನಲ್ಲಿ ಉತ್ಪನ್ನಗಳನ್ನು ಅಂಗಡಿಗೆ ಸಾಗಿಸಲಾಗುವುದಿಲ್ಲ ಅದ್ದರಿಂದ ಅಂಗಡಿಯನ್ನು ನಿರ್ವಹಿಸುವ ವೆಚ್ಚವನ್ನು ಉತ್ಪನ್ನದ ವೆಚ್ಚದೊಂದಿಗೆ ಸೇರಿಸಲಾಗುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳಲ್ಲಿ ಉತ್ಪನ್ನಗಳು ಅಗ್ಗವಾಗಿರುತ್ತವೆ.

ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ನ ಲಾಭವನ್ನು ಪಡೆಯುವವರು ಕಡಿಮೆ ಇದ್ದಾರೆ. ಹೆಚ್ಚಾಗಿ ನಗರ ಪ್ರದೇಶಗಳಲ್ಲಿ ಇರುವ ಜನರು ಇ ದರ ಲಾಭವನ್ನು ಪಡೆಯುತ್ತಾರೆ. ಏಕೆಂದರೆ ಇದರ ಸೇವೆಯು ಸಾಮಾನ್ಯವಾಗಿ ನಗರ ಪ್ರದೇಶಗಳಿಗೆ ಮಾತ್ರ ಸೀಮಿತವಾಗಿರುತ್ತದೆ. ಹಳ್ಳಿಯ ಜನರು ಇಂಟರ್‌ನೆಟ್ ಬಳಸಲು ಮತ್ತು ಆನ್‌ಲೈನ್ ವಹಿವಾಟುಗಳನ್ನು ಮಾಡಲು ಸುಸಜ್ಜಿತ ಸಾಧನಗಳು ಇಲ್ಲದಿರುವುದು ಕಾರಣಗಳು ಇರಬಹುದು. ಅದ್ದರಿಂದ ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳು ಮೊದಲು ಖರೀದಿದಾರರ ಪಿನ್ ಕೋಡ್ ಸಂಖ್ಯೆಯನ್ನು ಪರಿಶೀಲಿ ಸುತ್ತವೆ ಏಕೆಂದರೆ ಉತ್ಪನ್ನವನ್ನು ಖರೀದಿದಾರರಿಗೆ ರವಾನಿಸಬಹುದೇ ಅಥವಾ ಇಲ್ಲವೇ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸಬೇಕಾಗುತ್ತದೆ.

ಭವಿಷ್ಯದಲ್ಲಿ ತಂತ್ರಜ್ಞಾನದ ಪ್ರಗತಿಯೊಂದಿಗೆ ಈ ಸೇವೆಯನ್ನು ದೇಶದಾದ್ಯಂತ ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳಾದ ಫ್ಲಿಪ್‌ಕಾರ್ಟ್, ಅಮೇಝಾನ್ ಇತ್ಯಾದಿಗಳ ಆಗಮನದೊಂದಿಗೆ ಹೆಚ್ಚಿನ ನಗರಗಳಲ್ಲಿ ಸಾಂಪ್ರದಾಯಿಕ ಅಂಗಡಿಗಳು ಮಾರಾಟದಲ್ಲಿ ಕುಸಿತವನ್ನು ಕಂಡಿವೆ. ಈಗ ಖರೀದಿದಾರರು ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿಸಲು ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳಿಗೆ ಆದ್ಯತೆ ನೀಡುತ್ತಾರೆ ಏಕೆಂದರೆ ಅವರು ಆರಾಮಾಗಿ ತಮ್ಮ ಮನೆಗಳಲ್ಲಿ ಕುಳಿತು ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿಸಬಹುದು.

ಗ್ರಾಹಕರು ಅಂಗಡಿಗಳಲ್ಲಿ ಇರುವುದಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಉತ್ಪನ್ನಗಳನ್ನು ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳಲ್ಲಿ ನೋಡುವ ಅವಕಾಶವನ್ನು ಹೊಂದಿರುತ್ತಾರೆ. ಅದ್ದರಿಂದ ಆಯ್ಕೆಗಳು ಹೆಚ್ಚಿರುತ್ತವೆ ಮತ್ತು ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳು ಉತ್ಪನ್ನವನ್ನು ಮನೆಗೆ ಸಹ ಪೂರೈಸುತ್ತವೆ. ಈ ಕೆಲಸವನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಅಂಗಡಿಗಳು ಮಾಡುವುದಿಲ್ಲ.

ಇ ಕಾಮರ್ಸ್ ವಿಧಗಳು ಮತ್ತು ಅದರ ವ್ಯಾಪ್ತಿ (Type of E Commerce and its Scope):

ಇ ಕಾಮರ್ಸ್‌ನು ನಾಲ್ಕು ವಿಧವಾಗಿ ವಿಂಗಡಿಸಬಹುದು. ಇವು business to business b2b, business to customer b2c, government to business g2b ಮತ್ತು government to citizen g2c.

ಕಂಪನಿಗಳು ಇ ಕಾಮರ್ಸ್‌ನ್ನು ನಾಲ್ಕು ಪ್ರಮುಖ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ನಡೆಸುತ್ತವೆ. ಆ ಪ್ರಮುಖ ಪ್ರದೇಶಗಳು ಯಾವವೆಂದರೆ :

- ನೇರ ಮಾರುಕಟ್ಟೆ, ಮಾರಾಟ ಮತ್ತು ಸೇವೆ
- ಆನ್‌ಲೈನ್ ಬ್ಯಾಂಕಿಂಗ್ ಮತ್ತು ಬಿಲ್ಲಿಂಗ್
- ಮಾಹಿತಿಯ ಸುರಕ್ಷಿತ ವಿತರಣೆ
- ಮೌಲ್ಯ ಸರಣಿ ವ್ಯಾಪಾರ ಮತ್ತು ಕಾರ್ಪೋರೇಟ್ ಖರೀದಿ
- ಸರ್ಕಾರಕ್ಕೆ ತೆರಿಗೆ ರಿಟರ್ನ್ ತುಂಬುವುದು.

ಇ ಕಾಮರ್ಸ್ ಕ್ಷೇತ್ರವು ಬಹಳ ವಿಶಾಲವಾಗಿದೆ. ಇ ಕಾಮರ್ಸ್‌ನ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಹಲವಾರು ಇವೆ. ಉದಾಹರಣೆಗಳು ಹೋಮ್

© NIMI NOT TO BE REPUBLISHED

ಬ್ಯಾಂಕಿಂಗ್, ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮಾಲ್‌ಗಳಲ್ಲಿ ಶಾಪಿಂಗ್ ಮಾಡುವುದು, ಶೇರುಗಳನ್ನು ಖರೀದಿಸುವುದು, ಉದ್ಯೋಗ ಹುಡುಕುವುದು, ಹರಾಜು ನಡೆಸುವುದು, ಜಗತ್ತಿನಾದ್ಯಂತ ಇರುವ ವ್ಯಾಪಾರದ ಪಾಲುದಾರರನ್ನು ವಿದ್ಯುನ್ಮಾನದಿಂದ ಸಹಕರಿಸುವುದು ಮತ್ತು ಗ್ರಾಹಕರಿಗೆ ವಿವಿಧ ಸೇವೆಗಳನ್ನು ಒದಗಿಸುವುದು. ಇ ಕಾಮರ್ಸ್‌ನ ವಿವಿಧ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸುವುದು ನಾಲ್ಕು ಪ್ರಮುಖ ಬೆಂಬಲ ವರ್ಗವನ್ನು ಅವಲಂಬಿಸಿದೆ ಉದಾಹರಣೆಗೆ ಜನರು, ಸಾರ್ವಜನಿಕ ನೀತಿಗಳು, ಮತ್ತು ಮಾರ್ಕೆಟಿಂಗ್/ ಜಾಹೀರಾತು ಮತ್ತು ಪೂರೈಕೆ ಸರಪಳಿ ಲಾಜಿಸ್ಟಿಕ್‌ಗಳು. ಇವುಗಳ ಜೊತೆಗೆ ಮೂಲಸೌಕರ್ಯದ ಬೆಂಬಲವೂ ಇರಬೇಕು. ಪ್ರತಿಯೊಂದು ಸಂಸ್ಥೆಯಲ್ಲಿನ ಇ ಕಾಮರ್ಸ್ ನಿರ್ವಹಣೆಯು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಮೂಲಸೌಕರ್ಯವನ್ನು ಸಂಘಟಿಸುತ್ತದೆ. ಸಂಬಂಧಗಳನ್ನು ತಿಳಿಸಿಕೊಡಲು ಇ ಕಾಮರ್ಸ್‌ನ B2C ಸಂದರ್ಭದಲ್ಲಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಕೆಳಗೆ ವಿವರಿಸಲಾಗಿದೆ.

ಇ ಕಾಮರ್ಸ್‌ನ ಲಾಭಗಳು (Benefits of E Commerce) :

- ಇ ಕಾಮರ್ಸ್ ಜನರಿಗೆ ಯಾವುದೇ ಸಮಯದ ಅಥವಾ ಸ್ಥಾನಗಳ ದೂರದ ಅಡೆತಡೆಗಳಿಲ್ಲದೇ ವ್ಯವಹಾರಗಳನ್ನು ನಡೆಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಜನರು ಹಗಲು ಅಥವಾ ರಾತ್ರಿ ಯಾವುದೇ ಸಮಯದಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಲಾಗ್ ಇನ್ ಮಾಡಬಹುದು ಮತ್ತು ಮೌಸ್‌ನ ಒಂದೇ ಒಂದು ಕ್ಲಿಕ್‌ನಿಂದ ತ.
- ಇ ಕಾಮರ್ಸ್ ಈಗಾಗಲೇ ಸ್ಥಾಪಿತಗೊಂಡ ಉತ್ಪನ್ನಗಳಿಗೆ ಸೂಕ್ತವಾಗಿದೆ. ಇಂತಹ ಉತ್ಪನ್ನಗಳಿಗೆ ಗ್ರಾಹಕರು ಕಡಿಮೆ ಇರುತ್ತಾರೆ. ಆದರೆ ವಿಶಾಲವಾದ ಮಾರುಕಟ್ಟೆಯಲ್ಲಿ ಎಂದರೆ ಇಂಟರ್ನೆಟ್‌ನಲ್ಲಿ ಸ್ಥಾಪಿತ ಉತ್ಪನ್ನಗಳು ಸಹ ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದಲ್ಲಿ ಕಾರ್ಯಗಳನ್ನು ಸೃಷ್ಟಿಸುತ್ತವೆ.
- ಇ ಕಾಮರ್ಸ್‌ನ ಇನ್ನೊಂದು ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ, ಇದು ವ್ಯಾಪಾರ ಮಾಡುವ ಅಗ್ಗದ ಸಾಧನವಾಗಿದೆ.
- ಮಾರುಕಟ್ಟೆಯ ದಿನನಿತ್ಯದ ಒತ್ತಡಗಳು ಕಂಪನಿಗಳಿಗೆ ತಮ್ಮ ಸ್ಪರ್ಧಾತ್ಮಕ ಸ್ಥಾನವನ್ನು ಸುಧಾರಿಸಲು ಮಾಡುವ ಹೂಡಿಕೆಯ ಅವಕಾಶಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುವಲ್ಲಿ ತಮ್ಮ ಪಾತ್ರವನ್ನು ನಿರ್ವಹಿಸಿವೆ. ಒಂದು ಪ್ರಬುದ್ಧ ಮಾರುಕಟ್ಟೆ, ಹೆಚ್ಚಿದ ಸ್ಪರ್ಧೆಗಳು ಇವೆಲ್ಲವೂ ಹೂಡಿಕೆ ಮಾಡಲು ಲಭ್ಯವಿರುವ ಹಣದ ಪ್ರಮಾಣವನ್ನು ಕಡಿಮೆ ಮಾಡಿವೆ. ಒಂದು ವೇಳೆ ಮಾರಾಟದ ಬೆಲೆಯನ್ನು ಹೆಚ್ಚು ಮಾಡದಿದ್ದರೆ ಮತ್ತು ಉತ್ಪಾದನಾ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡದಿದ್ದರೆ ವ್ಯಾಪಾರ ಮಾಡುವ ವಿಧಾನವನ್ನು ವ್ಯತ್ಯಾಸಗೊಳಿಸಬೇಕು. ಉಂಟಾದ ವೆಚ್ಚಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಮೂಲಕ ಇ ಕಾಮರ್ಸ್ ಪರಿಹಾರವನ್ನು ಒದಗಿಸಿದೆ.
- ಖರೀದಿದಾರರ ದೃಷ್ಟಿಕೋನದಿಂದಲೂ ಸಹ ಇ ಕಾಮರ್ಸ್ ಸಾಕಷ್ಟು ಸ್ಪಷ್ಟವಾದ ಅನುಕೂಲಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

1 ಖರೀದಿದಾರರ ವಿಂಗಡನೆಯ ಸಮಯದಲ್ಲಿ ಕಡಿತ

2 ಉತ್ತಮ ಖರೀದಿದಾರರ ನಿರ್ಧಾರಗಳು

3 ಸರಕುಪಟ್ಟಿ (invoice) ಮತ್ತು ಆರ್ಡರ್ ವ್ಯತ್ಯಾಸಗಳನ್ನು ಪರಿಹರಿಸುವಲ್ಲಿ ಕಡಿಮೆ ಸಮಯವನ್ನು ವ್ಯಯಿಸಲಾಗುತ್ತದೆ.

4 ಪರ್ಯಾಯ ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿಸಲು ಹೆಚ್ಚಿದ ಅವಕಾಶಗಳು.

- ಒಂದು ವ್ಯವಹಾರವನ್ನು 'E Commerce enabled' ಮಾಡುವ ಕಾರ್ಯತಂತ್ರದ ಪ್ರಯೋಜನವೆಂದರೆ, ಇದು ವಿತರಣಾ ಸಮಯವನ್ನು , ಕಾರ್ಮಿಕರ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆಮಾಡುತ್ತದೆ, ಮತ್ತು ಕೆಳಗಿನ ಸ್ಥಳಗಳಲ್ಲಿ ಮಾಡಿದ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ :

1 ಡಾಕ್ಯುಮೆಂಟ್ ತಯಾರಿಕೆ

2 ದೋಷಗಳನ್ನು ಪತ್ತೆಮಾಡುವುದು ಮತ್ತು ಸರಿಪಡಿಸುವುದು

3 ಸಮನ್ವಯ

4 ಮೇಲ್ ತಯಾರಿಸುವುದು

5 ದೂರವಾಣಿ ಕರೆ ಮಾಡುವುದು

6 ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಯಂತ್ರಗಳು

7 ಡಾಟಾ ನಮೂದಿಸುವುದು

8 ಅಧಿಕ ಸಮಯ

9 ಮೇಲ್ವಿಚಾರಣೆ ಖರ್ಚುಗಳು

- ಇ ಕಾಮರ್ಸ್‌ನ ಕಾರ್ಯಾಚರಣೆಯ ಪ್ರಯೋಜನಗಳು ಎಂದರೆ , ಸಮಯವನ್ನು ಮತ್ತು ವ್ಯಾಪಾರದ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಪೂರ್ಣಗೊಳಿಸಲು ಅಗತ್ಯವಿರುವ ವ್ಯಕ್ತಿ ಎರಡನ್ನೂ ಕಡಿಮೆ ಮಾಡುವುದನ್ನು ಒಳಗೊಂಡಿದೆ. ಇದು ಇತರ ಸಂಪನ್ಮೂಲಗಳ ಮೇಲೆ ಒತ್ತಡವನ್ನೂ ಕಡಿಮೆ ಮಾಡುತ್ತದೆ. ಈ ಎಲ್ಲ ಅನುಕೂಲಗಳಿಂದಾಗಿ ಇ ಕಾಮರ್ಸ್‌ನ ಶಕ್ತಿಯನ್ನು ಬಳಸಿಕೊಳ್ಳಬಹುದು ಮತ್ತು ಒಂದು ವ್ಯಾಪಾರವನ್ನು E Business solution providers. ಗಳಿಂದ ಲಭ್ಯವಿರುವ ಶಕ್ತಿಯುತವಾದ ಇ ಕಾಮರ್ಸ್ ಪರಿಹಾರಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಇ ವ್ಯಾಪಾರಕ್ಕೆ ಪರಿವರ್ತಿಸಬಹುಥವಾ ಮಾರಾಟ ಮಾಡಬಹುದು.
- ವೆಬ್‌ಸೈಟ್‌ನಿಂದ ತೆಗೆದುಕೊಳ್ಳಲಾದ ಆರ್ಡರ್‌ಗೆ ನೇರ ಮಾರಾಟದ ವೆಚ್ಚವು ಸಾಂಪ್ರದಾಯಿಕ (ರಿಟೇಲ್, ಪೇಪರ್ ಆಧಾರಿತ) ವಿಧಾನಕ್ಕಿಂತ ಕಡಿಮೆಯಾಗಿರುತ್ತದೆ. ಏಕೆಂದರೆ ಆನ್‌ಲೈನ್ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಖರೀದಿಯ ಆರ್ಡರ್ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿ ಯಾವುದೇ ಮಾನವ ಸಂವಹನ ಇರುವುದಿಲ್ಲ. ಇದಲ್ಲದೇ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮಾರಾಟವು ವಾಸ್ತವವಾಗಿ ಪ್ರಕ್ರಿಯೆಯ ದೋಷಗಳನ್ನು ನಿವಾರಿಸುತ್ತದೆ, ವೇಗವಾಗಿದೆ ಮತ್ತು ಸಂದರ್ಶಕರಿಗೆ ಹೆಚ್ಚು ಅನುಕೂಲಕರವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಉತ್ಪನ್ನಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು ಗುರುತಿಸಲು ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡ್ಡು (Browse e-Commerce Sites to Identify Products & Services)

ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ನಲ್ಲಿ ವಹಿವಾಟುಗಳನ್ನು ಕೈಗೊಳ್ಳಿಂ (Undertake transactions on an e-commerce site)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ವಿವಿಧ ಇ ಕಾಮರ್ಸ್ ಸೈಟುಗಳನ್ನು ವಿವರಿಸಿ
- ಆನ್‌ಲೈನ್ ಕ್ಯಾಟಲಾಗ್, ಶಾಪಿಂಗ್ ಕಾರ್ಟ್‌ಗಳು ಮತ್ತು ಪುಟಗಳನ್ನು ಪರಿಶೀಲಿಸುವುದನ್ನು ವಿವರಿ೬.
- ಪಾವತಿಸುವುದು, ಆರ್ಡರ್ ಪ್ರಕ್ರಿಯೆ ಮತ್ತು ಆಥರೈಸೆಷನ್ , ಚಾರ್ಜ್ ಬ್ಯಾಕ್‌ಗಳನ್ನು ವಿವರಿ೬.
- ಇತರ ಪಾವತಿಸುವ ಆಯ್ಕೆಗಳನ್ನು ವಿವರಿ೬.

ವಿವಿಧ ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳು (Different E Commerce sites)

ಪ್ರಪಂಚದ ಕೆಲವು ಜನಪ್ರೀಯ ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳು ಯಾವವೆಂದರೆ : Crate & Barrel, Symantec, Amway, Microsoft, Amazon, HP ಇತ್ಯಾದಿಗಳು.

ಭಾರತದಲ್ಲಿ ಕೆಲವು ಬಿಕ್ಕಟ್ಟುಗಳ ನಂತರ, ಇ ಕಾಮರ್ಸ್ ಕ್ರಮೇಣವಾಗಿ ವೇಗವನ್ನು ಪಡೆಯುತ್ತಿದೆ. ಕೆಲವು ಜನಪ್ರಿಯ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಯಾವವೆಂದರೆ :

Amazon, FlipKart, Jabong, Naaptol ಇತ್ಯಾದಿಗಳು.

ಆನ್‌ಲೈನ್ ಕ್ಯಾಟಲಾಗ್, ಶಾಪಿಂಗ್ ಕಾರ್ಟ್‌ಗಳು ಮತ್ತು ಚೆಕ್ ಔಟ್‌ಪುಟಗಳು

ಆನ್‌ಲೈನ್ ಕ್ಯಾಟಲಾಗ್‌ಗಳು ಎಂದರೆ ಮಾರಾಟ ಮಾಡಲು ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿ ನೀಡಲಾದ ಉತ್ಪನ್ನಗಳ ಪಟ್ಟಿಯಾಗಿದೆ. ಖರೀದಿದಾರರು ಬ್ರೌಸ್ ಮಾಡುವ ಮೂಲಕ ಉತ್ಪನ್ನವನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತಾರೆ ಮತ್ತೆ ತಮಗೆ ಸೂಕ್ತವಾದ ಉತ್ಪನ್ನವನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ.

ಶಾಪಿಂಗ್ ಕಾರ್ಟ್ ಎಂದರೆ ಖರೀದಿದಾರರು ಆಯ್ಕೆ ಮಾಡಿದ ಆದರೆ ಅಂತಿಮವಾಗಿ ಬೆಲೆಯನ್ನು ಪಾವತಿಸುವ ಮೊದಲಿನ ಉತ್ಪನ್ನಗಳ ಬಕೆಟ್ (ಪಟ್ಟಿ) ಆಗಿದೆ. ಇದನ್ನು ರಿಟೇಲ್ ಅಂಗಡಿಗಳಲ್ಲಿ ಬಳಸಲಾಗುತ್ತದೆ. ಆದರೆ ಅದೇ ಪರಿಕಲ್ಪನೆಯನ್ನು ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿ ಅಳವಡಿಸಲಾಗಿದೆ. ಖರೀದಿದಾರರಿಂದ ಆಯ್ಕೆ ಮಾಡಲ್ಪಟ್ಟ ಉತ್ಪನ್ನಗಳನ್ನು ವರ್ಚುವಲ್ ಬಕೆಟ್ ಮಾಡುವ ಮೂಲಕ ತೋರಿಸಲಾಗುತ್ತದೆ.

ಬಯಸಿದ ಉತ್ಪನ್ನವನ್ನು ಆಯ್ಕೆ ಮಾಡಿದ ಮೇಲೆ ಖರೀದಿದಾರರು ಅಂತಿಮವಾಗಿ ಕ್ಲಿಕ್ ಮಾಡಿ ಉತ್ಪನಗಳ ಬೆಲೆಯನ್ನು ಪಾವತಿಸಲು ಚೆಕ್ ಔಟ್ ಮಾಡುತ್ತಾರೆ.

ಪಾವತಿ ಮತ್ತು ಆದೇಶ ಪ್ರಕ್ರಿಯೆ (Payment and order processing)

ಚೆಕ್ ಔಟ್ ಮಾಡಿದ ನಂತರ ಪಾವತಿ ಆಯ್ಕೆಯನ್ನು ತೋರಿಸಲಾಗುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಇದು ಅನೇಕ ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ ಉದಾಹರಣೆಗೆ, COD (Cash On Delivery), ಇಂಟರ್‌ನೆಟ್ ಬ್ಯಾಂಕಿಂಗ್, ಡೆಬಿಟ್ ಕಾರ್ಡ್, ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಮತ್ತ ಇನ್ನೂ ಹಲವಾರು ಆಯ್ಕೆಗಳು. ಖರೀದಿದಾರರು ಸೂಕ್ತವಾದ ಆಯ್ಕೆಯನ್ನು ಆರಿಸಿಕೊಳ್ಳುತ್ತಾನೆ ಮತ್ತು ಯಶಸ್ವಿಯಾಗಿ ಪಾವತಿಸಿದ ನಂತರ ಅದನ್ನು ಖರೀದಿದಾರರಿಗೆ ಸೂಚಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅಂತಿಮವಾಗಿ ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿದಾರರ ವಿಳಾಸಕ್ಕೆ ತಲುಪಿಸಲಾಗುತ್ತದೆ.

ಅಥೊರೈಸೇಷನ್ ಮತ್ತು ಚಾರ್ಜ್ ಬ್ಯಾಕ್ (Authorization and chargeback)

ಅಥೊರೈಸೇಷನ್ ಎಂದರೆ ಮಾಹಿತಿಯ ಸುರಕ್ಷತೆಗೆ ಸಂಬಂಧಿಸಿದ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಪ್ರವೇಶ ಹಕ್ಕುಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುವ ಕಾರ್ಯವಾಗಿದೆ ಮತ್ತು ಸಾಮಾನ್ಯವಾಗಿ ಕಂಪ್ಯೂಟರ್ ಭದ್ರತೆ ಮತ್ತು ನಿರ್ದಿಷ್ಟವಾಗಿ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವಾಗಿದೆ.

ಚಾರ್ಜ್ ಬ್ಯಾಕ್ ಎಂದರೆ ಖರೀದಿದಾರನಿಂದ ಬೆಲೆಯನ್ನು ಕಡಿತಗೊಳಿಸಿದ ನಂತರ ಹಣವನ್ನು ಅವನಿಗೆ ಹಿಂತಿರುಗಿಸುವುದನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ವಿಭಿನ್ನ ಸನ್ನಿವೇಶಗಳಲ್ಲಿ ಸಂಭವಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಯಾರಾದರೂ irctc ವೆಬ್ ಸೈಟ್‌ನಲ್ಲಿ ಟಿಕೆಟನ್ನು ಖರೀದಿಸುತ್ತಾರೆ ಎಂದು ಭಾವಿಸೋಣ. ಮತ್ತು ಟಿಕೆಟ್‌ನ ಬೆಲೆಯನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಪಾವತಿಸಿದ್ದಾರೆ ಎಂದು ಭಾವಿಸೋಣ. ಆದರೆ ನಂತರ ಟಿಕೆಟ್‌ನ್ನು ಕಾಯ್ದಿರಿಸಲಾಗಿರುವುದಿಲ್ಲ. ನಂತರ ಚಾರ್ಜ್ ಬ್ಯಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ಹಣವನ್ನು ಗ್ರಾಹಕರಿಗೆ ಹಿಂತಿರುಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಇತರ ಪಾವತಿ ಆಯ್ಕೆಗಳು (Other payment options)

ಮೇಲೆ ಚರ್ಚಿಸಿದ ಪಾವತಿಸುವ ಆಯ್ಕೆಗಳ ಹೊರತಾಗಿ ಮೊಬೈಲ್ ಪೇಮೆಂಟ್‌ನಂತಹ ಇತರ ರೀತಿಯ ಕೆಲವು ಆಯ್ಕೆಗಳು ಇರುತ್ತವೆ. ಇತ್ತೀಚೆಗೆ Airtel Money ಅಥವಾ Vodafone mpesa ಮುಂತಾದ ಮೊಬೈಲ್ ಪಾವತಿಯ ವಿಧಾನಗಳು ವಿಕಸನಗೊಂಡಿವೆ. ಅದ್ದರಿಂದ ಜನರು ತಮ್ಮ ಮೊಬೈಲ್ ಮೂಲಕವೂ ಸಹ ಪಾವತಿಸಬಹುದು.

ವಿವಿಧ ಇ ಕಾಮರ್ಸ್ ವಹಿವಾಟುಗಳಂತಹ ಯುಟಿಲಿಟಿ ಬಿಲ್‌ಗಳನ್ನು ಪಾವತಿಸುವುದು, ವೆಬ್‌ಸೈಟ್‌ಗಳಿಂದ ಶಾಪಿಂಗ್, ರೀಚಾರ್ಜ್ ಮುಂತಾದವುಗಳನ್ನು ಮೊಬೈಲ್‌ನಿಂದ ಪಾವತಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಉತ್ಪನ್ನಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು ಗುರುತಿಸಲು ಇ ಕಾಮರ್ಸ್ ಸೈಟ್‌ಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡಿ (Browse e-Commerce Sites to Identify Products & Services)

ಇ ಕಾಮರ್ಸ್ ಭದ್ರತಾ ಸಮಸ್ಯೆಗಳು ಮತ್ತು ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇಗಳು (E Commerce Security issues and Payment Gateways)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಇ ಕಾಮರ್ಸ್ ಭದ್ರತಾ ಸಮಸ್ಯೆಗಳನ್ನು ವಿವರಿಸಿ
- ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇ ಯನ್ನು ವಿವರಿಸಿ.

ಇ ಕಾಮರ್ಸ್ ಭದ್ರತಾ ಸಮಸ್ಯೆ (E Commerce security issue)

ಇ ಕಾಮರ್ಸ್ ಭದ್ರತೆ ಎಂದರೆ ಇ ಕಾಮರ್ಸ್‌ನ ಸ್ವತ್ತುಗಳನ್ನು ಅನಧಿಕೃತ ಪ್ರವೇಶ, ಬಳಕೆ, ಬದಲಾವಣೆ, ಅಥವಾ ವಿನಾಶಗೊಳಿಸುವಿಕೆ ಮುಂತಾದವುಗಳಿಂದ ರಕ್ಷಿಸುವುದಾಗಿದೆ. ಭದ್ರತಾ ವೈಶಿಷ್ಟ್ಯಗಳು ಸುರಕ್ಷಿತ ವ್ಯವಸ್ಥೆಯ ಖಾತರಿ ನೀಡದಿದ್ದರೂ ಅವು ಸುರಕ್ಷಿತ ವ್ಯವಸ್ಥೆಯನ್ನು ನಿರ್ಮಿಸಲು ಅಗತ್ಯವಾಗಿವೆ.

ಇ ಕಾಮರ್ಸ್‌ನ ಭಾರಿ ಪ್ರಾಮಾಣದ ಹೆಚ್ಚಳವು ಸಂಬಂಧಿತ ಭದ್ರತಾ ಬೆದರಿಕೆಗಳ ಹೊಸ ಪೀಳಿಗೆಗೆ ಕಾರಣವಾಯಿತು. ಆದರೆ ಯಾವುದೇ ಇ ಕಾಮರ್ಸ್ ವ್ಯವಸ್ಥೆಯು ನಾಲ್ಕು ಅವಶ್ಯಕತೆಗಳನ್ನು ಪೂರೈಸಬೇಕು:

a **ಗೌಪ್ಯತೆ** (Privacy) : ಅನಧಿಕೃತ ಪಕ್ಷಗಳಿಂದ ವಿನಿಮಯದ ಮಾಹಿತಿಯನ್ನು ಗೌಪ್ಯವಾಗಿಟ್ಟುಕೊಳ್ಳಬೇಕು.

b **ಸಮಗ್ರತೆ** (Integity) : ವಿನಿಮಯಗೊಂಡ ಮಾಹಿತಿಯನ್ನು ಬದಲಾಯಿಸಬಾರದು ಅಥವಾ ತಿದ್ದಬಾರದು.

c **ದೃಢೀಕರಣ** (Authentication) - ಕಳುಹಿಸುವವರು ಮತ್ತು ಸ್ವೀಕರಿಸುವವರು ಇಬ್ಬರೂ ಪರಸ್ಪರ ತಮ್ಮ ಗುರುತುಗಳನ್ನು ಸಾಬೀತುಪಡಿಸಬೇಕು.

d **ನಿರಾಕರಿಸದಿರುವುದು** (Non-repudiation) - ವಿನಿಮಯವಾದ ಮಾಹಿತಿಯನ್ನು ನಿಜವಾಗಿಯೂ ಸ್ವೀಕರಿಸಲಾಗಿದೆ ಎಂಬುದಕ್ಕೆ ಪುರಾವೆಯ ಅಗತ್ಯವಿದೆ.

ಭದ್ರತಾ ಬೆದರಿಕೆಗಳು (Security Threats)

ಇ ಕಾಮರ್ಸ್ ಪೂರೈಕೆದಾರರು ಎದುರಿಸುವ ಅತ್ಯಂತ ಸವಾಲಿನ ವಿಧಗಳಲ್ಲಿ ತಾಂತ್ರಿಕ ದಾಳಿಗಳು ಒಂದಾಗಿದೆ. ತಾಂತ್ರಿಕ ದಾಳಿಗಳ ಅಪರಾಧಿಗಳು ಮತ್ತು ನಿರ್ದಿಷ್ಟವಾಗಿ Denial-of-Service ದಾಳಿಗಳು, ಸಾಮಾನ್ಯವಾಗಿ ಬ್ಯಾಂಕುಗಳು, ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇಗಳು, ದೊಡ್ಡ ಆನ್‌ಲೈನ್ ರಿಟೇಲ್ ವ್ಯಾಪಾರಿಗಳು ಮತ್ತು ಜನಪ್ರಿಯ ಸಾಮಾಜಿಕ ಜಾಲತಾಣಗಳಂತಹ, ಸಾಮಾನ್ಯವಾಗಿ ಸೈಟ್‌ಗಳನ್ನು ಅಥವಾ ಸೇವೆಗಳನ್ನು ಹೋಸ್ಟ್ ಮಾಡುವ ಉನ್ನತ ಪ್ರೊಫೈಲ್ ವೆಬ್ ಸರ್ವರ್‌ಗಳನ್ನು ಗುರಿಯನ್ನಾಗಿಸುತ್ತವೆ.

Denial of Service **ದಾಳಿಗಳು** (Denial of Service Attacks)

Denial of Service (DoS) ದಾಳಿಗಳು ಒಂದು ಸರ್ವರ್‌ನ್ನು, ನೆಟ್‌ವರ್ಕ್‌ನ್ನು, ಅಥವಾ ವೆಬ್‌ಸೈಟ್‌ನ್ನು ಅವುಗಳ ಸಾಮಾನ್ಯ ಚಟುವಟಿಕೆಯನ್ನು ನಾಶಮಾಡುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. DoS ದಾಳಿಗಳ ವಿರುದ್ಧ ರಕ್ಷಿಸುವುದು ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಸವಾಲಿನ ಭದ್ರತಾ ಸಮಸ್ಯೆಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಈ ದಾಳಿಗಳನ್ನು ತಡೆಯುವಲ್ಲಿ ಪ್ರಮುಖ ತೊಂದರೆ ಎಂದರೆ ದಾಳಿಯ ಮೂಲವನ್ನು ಪತ್ತೆ ಹಚ್ಚುವುದು. ಏಕೆಂದರೆ ಅವರು ಸಾಮಾನ್ಯವಾಗಿ ತಪ್ಪಾದ ಅಥವಾ ದಾಳಿಯ ಮೂಲವನ್ನು ಮರೆಮಾಚಲು ವಂಚಿಸುವ IP ಸೋರ್ಸ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ.

The United States Computer Emergency Readiness Team, denial-of-service ನ ದಾಳಿಯ ಲಕ್ಷಣಗಳನ್ನು ಹೀಗೆ ವಿವರಿಸುತ್ತದೆ :

- ಅಸಾಮಾನ್ಯ ನಿಧಾನಗತಿಯ ನೆಟ್‌ವರ್ಕ್‌ನ ಕಾರ್ಯಕ್ಷಮತೆ
- ನಿರ್ದಿಷ್ಟ ವೆಬ್‌ಸೈಟ್‌ನ ಅಲಭ್ಯತೆ
- ಯಾವುದೇ ವೆಬ್‌ಸೈಟ್‌ನ್ನು ಪ್ರವೇಶಿಸಲು ಅಸಮರ್ಥತೆ
- ಸ್ವೀಕರಿಸಿದ ಸ್ಪ್ಯಾಮ್ ಇ ಮೇಲ್‌ಗಳ ಸಂಖ್ಯೆಯಲ್ಲಿ ಹೆಚ್ಚಳ

DoS ದಾಳಿಗಳು ಹಲವಾರು ವಿಭಿನ್ನ ಮಾರ್ಗಗಳಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ. ಅವು ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿವೆ.

ICMP Flood (Smurf Attack)

ಇಲ್ಲಿ ದುಷ್ಕರ್ಮಿಗಳು ಮೋಸಕ್ಕೆ ಬಲಿಯಾಗುವವರ ಅಡ್ರೆಸ್‌ನಂತೆ ಕಾಣುವ ಮೂಲ ಅಡ್ರೆಸ್‌ನ್ನು ನಕಲಿ ಮಾಡಿ ಅದಕ್ಕೆ ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯ ಐಪಿ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ಕಳುಹಿಸುತ್ತಾರೆ. ನೆಟ್‌ವರ್ಕ್‌ನ ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್ ತ್ವರಿತವಾಗಿ ಬಳಸಲ್ಪಡುತ್ತದೆ, ಕಾನೂನುಬದ್ಧ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ಅವುಗಳು ಡೆಸ್ಟಿನೇಷನ್ ತಲುಪುವದನ್ನು ತಡೆಯುತ್ತವೆ.

(Teardrop Attack)

Teardrop ದಾಳಿಯು, ಓವರ್‌ಲ್ಯಾಪ್ ಆದ, ಹೆಚ್ಚು ಗಾತ್ರದ, ಪೇಲೋಡ್ ಹೊಂದಿದ ಕೆಟ್ಟದಾದ IP ಫ್ರಾಗ್ಮೆಂಟ್‌ಗಳನ್ನು ಗುರಿಯಾಗಿಸಿದ ಯಂತ್ರಕ್ಕೆ ಕಳುಹಿಸುವುದನ್ನು ಒಳಗೊಂಡೀರುತ್ತದೆ. ವಿವಿಧ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿನ, TCP/IP ಫ್ರಾಗ್ಮೆಂಟೇಷನ್ ರಿಅಸೆಂಬ್ಲಿ ಕೋಡ್‌ನಲ್ಲಿನ ಒಂದು ದೋಷವು (bug) ಫ್ರಾಗ್ಮೆಂಟ್‌ಗಳು

© NIMI NOT TO BE REPUBLISHED

ಅಸಮರ್ಪಕವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವಂತೆ ಮಾಡುತ್ತವೆ. ಇದರ ಪರಿಣಾಮವಾಗಿ ಅವುಗಳು ಕ್ರ್ಯಾಶ್ ಆಗುವಂತೆ ಮಾಡುತ್ತವೆ.

Phlashing

Permanent denial-of-service (PDoS) ಎಂದೂ ಕರೆಯಲಾಗುತ್ತದೆ. ಈ ದಾಳಿಯು ಒಂದು ವ್ಯವಸ್ಥೆಯನ್ನು ಎಷ್ಟು ಕೆಟ್ಟದಾಗಿ ಹಾನಿ ಮಾಡುತ್ತದೆ ಎಂದರೆ ಇದರಿಂದಾಗಿ ಹಾರ್ಡ್‌ವೇರ್‌ನ ಬದಲಾವಣೆ ಅಥವಾ ಮರುಸ್ಥಾಪನೆಯ ಅಗತ್ಯವಿರುತ್ತದೆ. ಅಪರಾಧಿಗಳು ಬಲಿಪಶು ವಿನ ಹಾರ್ಡ್‌ವೇರ್(ರೂಟರ್‌ಗಳಾಗಿರಬಹುದು, ಪ್ರಿಂಟರ್‌ಗಳಾಗಿರಬಹುದು ಅಥವಾ ಇತರ ನೆಟ್‌ವರ್ಕಿಂಗ್ ಹಾರ್ಡ್‌ವೇರ್‌ಗಳಾಗಿರಬಹುದು) ನಲ್ಲಿರುವ ರಿಮೋಟ್ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್ ಇಂಟರ್‌ಫೇಸ್‌ನ ಭದ್ರತಾ ನ್ಯೂನ್ಯತೆಗಳನ್ನು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ. ಈ ನ್ಯೂನ್ಯತೆಗಳು ದಾಳಿಕೋರರಿಗೆ ಸಾಧನದ ಫರ್ಮ್‌ವೇರನ್ನು ದೂರದಿಂದಲೇ ಮಾರ್ಪಡಿಸಲು, ಭ್ರಷ್ಟ ಆಗಿಸಲು, ಅಥವಾ ದೋಷಯುಕ್ತ ಫರ್ಮ್‌ವೇರ್ ಇಮೇಜನ್ನಾಗಿ 'ಮಾರ್ಪಡಿಸಲು' ಬಾಗಿಲನ್ನು ತೆರೆಯುತ್ತವೆ. ಅದ್ದರಿಂದ ಇವುಗಳು ಸಾಧನವನ್ನು ಇಟ್ಟಿಗೆಯನ್ನಾಗಿಸುತ್ತವೆ ಮತ್ತು ಅದರ ಮೂಲ ಉದ್ದೇಶಕ್ಕಾಗಿ ಅದನ್ನು ಶಾಶ್ವತವಾಗಿ ನಿರುಪಯುಕ್ತಗೊಳಿಸುತ್ತವೆ.

Distributed Denial-of-Service ದಾಳಿಗಳು

Distributed Denial-of-Service ದಾಳಿಗಳು IT ಮ್ಯಾನೇಜರ್‌ಗಳಿಗೆ ಹೆಚ್ಚಿನ ಭದ್ರತಾ ಭಯವನ್ನು ಉಂಟುಮಾಡುತ್ತವೆ. ಒಂದೇ ನಿಮಿಷದಲ್ಲಿ ಸಾವಿರಾರು ದುರ್ಬಲ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಕಾನೂನುಬದ್ಧ ಸಂಚಾರವನ್ನು(traffic) ನಿಲ್ಲಿಸುವ ಮೂಲಕ ಬಲಿಪಶುವಾದ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಹಾನಿಮಾಡಬಲ್ಲವು. Distributed Denial-of-Service ದಾಳಿಗಳು ಉದ್ದೇಶಿತ ವ್ಯವಸ್ಥೆಯ (ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ವೆಬ್ ಸರ್ವರ್‌ಗಳು) ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್ ಅಥವಾ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಅನೇಕ ಹೊಂದಾಣಿಕೆ ಮಾಡಿಕೊಳ್ಳುವ ವ್ಯವಸ್ಥೆಗಳಿಂದ ತುಂಬಿಸುತ್ತವೆ. ಅತ್ಯಂತ ಪ್ರಸಿದ್ಧವಾದ DDoS ದಾಳಿಗಳು ಫೆಬ್ರುವರಿ 2000 ದಂದು ಸಂಭವಿಸಿದವು. ಆಗ Yahoo, Buy.com, eBay, Amazon ಮತ್ತು CNN ನ್ನು ಒಳಗೊಂಡಂತೆ ಅನೇಕ ವೆಬ್‌ಸೈಟ್‌ಗಳ ಮೇಲೆ ದಾಳಿ ಮಾಡಲಾಯಿತು ಮತ್ತು ಗಂಟೆಗಟ್ಟಲೇ ಹಲವರಿಗೆ ತಲುಪಲಾಗಲಿಲ್ಲ.

Brute Force ದಾಳಿಗಳು

Brute Force ದಾಳಿ, ಎಂದರೆ ಕ್ರಿಪ್ಟೊಗ್ರಾಫಿಕ್ ಯೋಜನೆಯನ್ನು ಸೋಲಿಸುವ ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯ ಸಾಧ್ಯತೆಗಳ ಒಂದು ವಿಧಾನವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ಒಂದು ಸಂದೇಶವನ್ನು ಡಿಕ್ರಿಪ್ಟ್ ಮಾಡಲು ಒಂದು ಕೀ ಸ್ಥಾನದಲ್ಲಿ ಸಾಧ್ಯವಿರುವ ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯ ಕೀಗಳಾಗಿವೆ. Brute Force ದಾಳಿಗಳು ಕಡಿಮೆ ತಂತ್ರಜ್ಞಾನ ಹೊಂದಿರುವ ದಾಳಿಗಳಾಗಿವೆ.

Non-Technical ದಾಳಿಗಳು (Non-Technical Attacks)

ಫಿಶಿಂಗ್ ಎನ್ನುವುದು ಕ್ರಿಮಿನಲ್ ಮೋಸದ ಒಂದು ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಇದು ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಸಂವಹನದಲ್ಲಿ ವಿಶ್ವಾಸಾರ್ಹ ಘಟಕದಿಂದ ನಂಬುವಂತೆ ಮಾಡಿ ಬಳಕೆದಾರರ ಹೆಸರು, ಪಾಸ್‌ವರ್ಡ್ ಮತ್ತು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ವಿವರಗಳು ಮುಂತಾದ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುವ ಪ್ರಯತ್ನವಾಗಿದೆ. ಫಿಶಿಂಗ್ ವಂಚನೆಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಬಲಿಪಶುಗಳಿಗೆ ಇ ಮೇಲ್ ಮಾಡುವ ಮೂಲಕ ನಡೆಸಲಾಗುತ್ತದೆ. ಇದರಲ್ಲಿ ಕಾನೂನುಬದ್ಧ ಸಂಸ್ಥೆಯು ಇ ಮೇಲ್ ಜೊತೆಗೆ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ವಿನಂತಿಸುವ ಮೂಲಕ ವಂಚಿಸಲಾಗುತ್ತದೆ. ಯಾವಾಗ ಬಲಿಪಶುಗಳು ಇ ಮೇಲ್‌ಗೆ ಜೋಡಿಸಿದ ಲಿಂಕ್ ಅನ್ನು ಅನುಸರಿಸುತ್ತಾರೊ ಆಗ ಅವರು ವಿಸ್ತಾರವಾದ ಮತ್ತು ಕಾನೂನುಬದ್ಧವಾದ ಸಂಸ್ಥೆಯ ನಕಲಿ ವೆಬ್‌ಸೈಟ್‌ಗೆ ಬರುತ್ತಾರೆ. ಫಿಶಿಂಗ್ ದಾಳಿಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಬ್ಯಾಂಕ್ ಖರೀದಿದಾರರು. ಆನ್‌ಲೈನ್ ಹರಾಜು ಸೈಟ್‌ಗಳು (ಉದಾ eBay), ಆನ್‌ಲೈನ್ ರೀಟೇಲರ್‌ಗಳು (ಉದಾ amezon) ಮತ್ತು ಸೇವೆ ಒದಗಿಸುವವರು (ಉದಾ PayPal) ಮುಂತಾದವುಗಳನ್ನು ಗುರಿಯಾಗಿಸುತ್ತದೆ. ಕಮ್ಯುನಿಟಿ ಬ್ಯಾಂಕರ್‌ಗಳ ಪ್ರಕಾಂ ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ಸೈಬರ್ ಅಪರಾಧಿಗಳು ಅವರ ದಾಳಿಯ ಸಮಯದಲ್ಲಿ ಹೆಚ್ಚು ಅತ್ಯಾಧುನಿಕವಾಗಿದ್ದಾರೆ.

ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್ - ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್ ಕ್ರಿಯೆಗಳನ್ನು ಮಾಡಲು ಅಥವಾ ಗೌಪ್ಯ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸಲು ಜನರನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಒಂದು ಕಲೆಯಾಗಿದೆ. ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್ ತಂತ್ರಗಳು ನೆಪ ಹೇಳುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. (ಇಲ್ಲಿ ವಂಚಕನು ಮಾಹಿತಿ ಬಹಿರಂಗಪಡಿಸುವುದಕ್ಕಾಗಿ ಸನ್ನಿವೇಶವನ್ನು ರಚಿಸುತ್ತಾನೆ. ಇದು ಸಂವಾದಾತ್ಮಕ ಧ್ವನಿ ರೆಕಾರ್ಡಿಂಗ್ (IVR) ಅಥವಾ ಫೋನ್ ಫಿಶಿಂಗ್ (ಇದರಲ್ಲಿ ವಂಚಕನು ಫೋನ್ ಮೂಲಕ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾನೆ.) ಮತ್ತು Trojans horses ಗಳೊಂದಿಗೆ ಬೈಟ್ಟಿಂಗ್ ಮಾಡುವುದು (ಇದರಲ್ಲಿ ಮೋಸಗಾರನು ಮಾಲ್‌ವೇರನ್ನು ಸಿಸ್ಟಮ್‌ಗೆ ಲೋಡ್ ಮಾಡಲು ಬಲಿಪಶು ಮಾಡುತ್ತಾನೆ.) ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್, ಇಕಾಮರ್ಸ್‌ನ ಸುರಕ್ಷತೆಗೆ ಒಂದು ಗಂಭೀರವಾದ ಅಪಾಯವಾಗಿದೆ. ಅದನ್ನು ಪತ್ತೆ ಹಚ್ಚಲು ಮತ್ತು ಎದುರಿಸಲು ಕಷ್ಟವಾಗಿದೆ ಏಕೆಂದರೆ ಇದು ಹಾರ್ಡ್‌ವೇರ್ ಅಥವಾ ಸಾಫ್ಟ್‌ವೇರ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ತಿದ್ದಲಾಗದ ಮಾನವ ಅಂಶಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಅದಾಗ್ಯೂ ಸಿಬ್ಬಂದಿ ತರಬೇತಿ ಮತ್ತು ಶಿಕ್ಷಣಗಳಿಂದ ಸ್ವಲ್ಪಮಟ್ಟಿಗೆ ದಾಳಿಯನ್ನು ತಡೆಯಬಹುದು.

ಸುರಕ್ಷಿತವಾಗಿರುವುದು ಹೇಗೆ (How to be secure)

ಸುರಕ್ಷಿತ ವೆಬ್‌ಸೈಟ್‌ಗಳಲ್ಲಿ ಶಾಪಿಂಗ್ ಮಾಡಿ(Shop at Secure Web Sites)

ಸುರಕ್ಷಿತ ವೆಬ್‌ಸೈಟ್‌ಗಳು ಮಾಹಿತಿಯನ್ನು ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಆನ್‌ಲೈನ್ ವ್ಯಾಪಾರಿಗಳ ಕಂಪ್ಯೂಟರ್‌ಗೆ ವರ್ಗಾವಣೆ ಮಾಡಲು ಎನ್‌ಕ್ರಿಪ್ಷನ್ ತಂತ್ರವನ್ನು ಬಳಸುತ್ತವೆ. ಎನ್‌ಕ್ರಿಪ್ಷನ್ ನೀವು ಕಳುಹಿಸುವ ಮಾಹಿತಿಯನ್ನು ಸ್ಕ್ರ್ಯಾಂಬಲ್ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆಗೆ ನಿಮ್ಮ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ನಂಬರ್, ಇದು ಕಂಪ್ಯೂಟರ್ ಹ್ಯಾಕರ್‌ಗಳು ಮಾರ್ಗದಲ್ಲಿ ಪಡೆಯುವುದನ್ನು ತಡೆಯುತ್ತದೆ. ಕಾನೂನುಬದ್ಧ ಪ್ರವೇಶ ಸವಲತ್ತನ್ನು ಹೊಂದಿದ ಜನರು ಮಾತ್ರ ಈ ಕೋಡ್ ಅನ್ನು ಅನ್ ಸ್ಕ್ರಾಂಬಲ್ ಮಾಡಬಹುದು. ನೀವು ಸುರಕ್ಷಿತ ಸೈಟ್‌ನೊಂದಿಗೆ ವ್ಯವಹರಿಸುತ್ತಿರುವಿರಿ ಎಂಬುದನ್ನು ನೀವು ಹೇಗೆ ಹೇಳಬಹುದು ಎಂದರೆ :

ನೀವು ನಿಮ್ಮ ಪರದೆಯ ಮೇಲ್ಭಾಗವನ್ನು ನೋಡಿದಾಗ ಅಲ್ಲಿ ವೆಬ್‌ಸೈಟ್ ವಿಳಾಸವನ್ನು ತೋರಿಸಿರುತ್ತಾರೆ("ಅಡ್ರೆಸ್ ಬಾರ್"). ಅಲ್ಲಿ ನೀವು https://. ಅನ್ನು ನೋಡಬೇಕು. http ಯ ನಂತರ ಇರುವ "s" ಆ ವೆಬ್‌ಸೈಟ್ ಸುರಕ್ಷಿತವಾಗಿದೆ ಎಂದು ಸೂಚಿಸುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ನೀವು ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿರುವ ಆರ್ಡರ್ ಪುಟಕ್ಕೆ ಹೋಗುವವರೆಗೂ "s" ಅನ್ನು ನೋಡುವುದಿಲ್ಲ.

ವೆಬ್‌ಸೈಟ್ ಸುರಕ್ಷಿತವಾಗಿದೆಯೆ ಎಂದು ನಿರ್ಧರಿಸಲು ಇನ್ನೊಂದು ಮಾರ್ಗವೆಂದರೆ, ನಿಮ್ಮ ಪರದೆಯ ಮೇಲಿರುವ ಅಡ್ರೆಸ್ ಬಾರ್‌ನಲ್ಲಿ ಮುಚ್ಚಿದ ಪ್ಯಾಡ್‌ಲಾಕ್‌ನ್ನು ಪ್ರದರ್ಶಿಸಲಾಗಿದೆಯೆ ಎಂಬುದನ್ನು ನೋಡಿ. ಒಂದು ವೇಳೆ ಆ ಲಾಕ್ ತೆರೆದಿದ್ದರೆ, ಅದು ಸುರಕ್ಷಿತ ಸೈಟ್ ಅಲ್ಲ ಎಂದು ಭಾವಿಸಬೇಕು. ಸಹಜವಾಗಿ, ಒಂದು ವೇಳೆ ವ್ಯಾಪಾರಿಯು ಡಾಟಾವನ್ನು ಅನ್‌ಸ್ಕ್ರ್ಯಾಂಬಲ್ ಆಗಿ ಸಂಗ್ರಹಿಸಿದ್ದರೆ, ಸುರಕ್ಷಿತ ಚಾನಲ್‌ಗಳ ಮೂಲಕ ನಿಮ್ಮ ಡಾಟಾವನ್ನು ರವಾನಿಸುವುದು ಕಡಿಮೆ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ. ವ್ಯಾಪಾರಿಯು ಡಾಟಾವನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟೆಡ್ ರೂಪದಲ್ಲಿ ಸಂಗ್ರಹಿಸಿದ್ದಾನೆಯೆ ಎಂಬುದನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಪ್ರಯತ್ನಿಸಬೇಕು. ಒಂದು ವೇಳೆ ಹ್ಯಾಕರ್ ಆಕ್ರಮಿಸಲು ಪ್ರಯತ್ನಿಸಿದರೆ, ಅವನು ನಿಮ್ಮ ಕ್ರೆಡಿಟ್ ಡಾಟಾವನ್ನು ಮತ್ತು ಇತರ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಿಲ್ಲ. ವ್ಯಾಪಾರಿಯ ಗೌಪ್ಯತೆ ಮತ್ತು ಭದ್ರತಾ ನೀತಿಗಳನ್ನು ಓದಲು ಮರೆಯದಿರಿ. ಇದು ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿನ ನಿಮ್ಮ ವೈಯಕ್ತಿಕ ಡಾಟಾವನ್ನು ಹೇಗೆ ರಕ್ಷಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ.

ನೀವು ಆರ್ಡರ್ ಮಾಡುವ ಮೊದಲು ವೆಬ್‌ಸೈಟ್‌ಅನ್ನು ಸಂಶೋಧಿಸಿ (Research the Web Site before You Order)

ನಿಮಗೆ ಈಗಾಗಲೇ ತಿಳಿದಿರುವ ಕಂಪನಿಗಳೊಂದಿಗೆ ವ್ಯಾಪಾರ ಮಾಡಿ. ಒಂದು ವೇಳೆ ನಿಮಗೆ ಕಂಪನಿಯ ಪರಿಚಯ ಇಲ್ಲದಿದ್ದರೆ, ಅವರ ಉತ್ಪನ್ನಗಳನ್ನು ಖರೀದಿಸುವ ಮೊದಲು ಕಂಪನಿಯ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳಿರಿ. ಅಪರಿಚಿತ ಕಂಪನಿಯಿಂದ ನೀವು ಏನನ್ನಾದರೂ ಖರೀದಿಸಲು ನಿರ್ಧರಿಸಿದರೆ, ದುಬಾರಿಯಲ್ಲದ ವಸ್ತುವನ್ನು ಆರ್ಡರ್ ಮಾಡುವುದರ ಮೂಲಕ ಪ್ರಾರಂಭಿಸಿ ಮತ್ತು ಕಂಪನಿಯು ವಿಶ್ವಾಸಾರ್ಹವಾಗಿದೆಯೆ ಇಲ್ಲವೆ ಎಂದು ತಿಳಿಯಿರಿ.

ವಿಶ್ವಾಸಾರ್ಹ ಕಂಪನಿಗಳು ತಮ್ಮ ಭೌತಿಕ ವ್ಯಾಪಾರದ ಜಾಹೀರಾತನ್ನು ನೀಡಬೇಕು ಮತ್ತು ಕನಿಷ್ಟ ಒಂದು ಫೋನ್ ಸಂಖ್ಯೆ (ಖರೀದಿದಾರರ ಸೇವೆ ಸಂಖ್ಯೆ ಅಥವಾ ಆರ್ಡರ್ ಲೈನ್) ನೀಡಬೇಕು. ಆ ಫೋನ್ ನಂಬರ್‌ಗೆ ಕರೆ ಮಾಡಿ ಮತ್ತು ವ್ಯವಹಾರವು ಕಾನೂನುಬದ್ಧವಾಗಿದೆಯೆ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸಲು ಪ್ರಶ್ನೆಗಳನ್ನು ಕೇಳಿರಿ. ಗಂಟೆಗಳ ನಂತರ ಕರೆ ಮಾಡಿದರೂ ಅನೇಕ ಕಂಪನಿಗಳು "live" ಉತ್ತರಿಸುವ ಸೇವೆಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ವಿಶೇಷವಾಗಿ ಅವರು ಆರ್ಡರ್‌ಗಳನ್ನು ತಪ್ಪಿಸಿಕೊಳ್ಳಲು ಬಯಸದಿದ್ದರೆ. ವ್ಯಾಪಾರಿಗಳು ಹಿಂತಿರುಗಿಸಿದ ಸರಕು ಮತ್ತು ದೂರುಗಳನ್ನು ಹೇಗೆ ನಿರ್ವಹಿಸುತ್ತಾರೆ ಎಂದು ಕೇಳಿರಿ. ಇದು ಸಂಪೂರ್ಣ ಮರುಪಾವತಿಯನ್ನು ಮಾಡುತ್ತದೆಯೊ ಅಥವಾ ಕ್ರೆಡಿಟ್‌ಗಳನ್ನು ಮಾತ್ರ ಸಂಗ್ರಹಿಸುತ್ತದೆಯೊ ಎಂದು ಕಂಡುಹಿಡಿಯಿರಿ.

ನೀವು Better Business Bureau, ಮೂಲಕ ಅಥವಾ district attorney's office ಅಥವಾ Attorney General ನಂತಹ ಸರ್ಕಾರಿ ಗ್ರಾಹಕ ರಕ್ಷಣೆ ಏಜನ್ಸಿ ಮೂಲಕ ಉತ್ತಮ ಕಂಪನಿಯನ್ನು ಸಂಶೋಧಿಸಬಹುದು. ನಗರಗಳಲ್ಲಿ ವಾಸಿಸುವ ನಿಮ್ಮ ಸ್ನೇಹಿತರು ಅಥವಾ ಕುಟುಂಬದ ಸದಸ್ಯರು ಕಂಪನಿಯ ಸಿಂಧುತ್ವವನ್ನು ಪರಿಶೀಲಿಸಬಹುದು. ನೆನಪಿಡಿ, ಯಾರು ಬೇಕಾದರೂ ವೆಬ್‌ಸೈಟನ್ನು ರಚಿಸಬಹುದು.

ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇಗಳು (Payment gateways)

ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇ, ಇ ಕಾಮರ್ಸ್‌ನ ಅಪ್ಲಿಕೇಷನ್ ಸರ್ವೀಸ್ ಪ್ರೊವೈಡರ್ ಸೇವೆಯಾಗಿದೆ. ಇದು ಇ - ವ್ಯವಹಾರಗಳು, ಆನ್‌ಲೈನ್ ರೀಟೇಲರ್‌ಗಳು, ಬ್ರಿಕ್ಸ್ ಮತ್ತು ಕ್ಲಿಕ್ಸ್ ಅಥವಾ ಸಾಂಪ್ರದಾಯಿಕ ಬ್ರಿಕ್ ಮತ್ತು ಮಾರ್ಟರ್‌ಗಳಿಗೆ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಪೇಮೆಂಟ್ (ಪಾವತಿ)ಗಳನ್ನು ಅಧಿಕೃತಗೊಳಿಸುತ್ತದೆ.

ಇದು ಹೆಚ್ಚಾಗಿ ಚಿಲ್ಲರೆ ಅಂಗಡಿಗಳಲ್ಲಿರುವ ಭೌತಿಕ ಮಾರಾಟದ ಸ್ಥಳದಲ್ಲಿರುವಂತೆಯೆ ಇರುತ್ತದೆ. ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇಗಳು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ವಿವರಗಳನ್ನು ರಕ್ಷಿಸುತ್ತವೆ. ಇವು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ನಂಬರ್‌ನಂತಹ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡುತ್ತವೆ ಮತ್ತು ಖರೀದಿದಾರು ಮತ್ತು ವ್ಯಾಪಾರಿಗಳ ನಡುವೆ ಅದೇ ರೀತಿ ವ್ಯಾಪಾರಿಗಳು ಮತ್ತು ಪೇಮೆಂಟ್ ಪ್ರೊಸೆಸರ್ ನಡುವೆ ಸುರಕ್ಷಿತವಾಗಿ ಮಾಹಿತಿಯನ್ನು ರವಾನಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತವೆ.

ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇ, ಪೇಮೆಂಟ್ ಪೋರ್ಟಲ್ (ಉದಾಹರಣೆಗೆ ವೆಬ್‌ಸೈಟ್, ಮೊಬೈಲ್ ಫೋನ್, ಅಥವಾ ಸಂವಾದಾತ್ಮಕ ಧ್ವನಿ ಪ್ರತಿಕ್ರಿಯೆ ಸೇವೆ) ಮತ್ತು ಫ್ರಂಟ್ ಎಂಡ್ ಪ್ರೊಸೆಸರ್ ಅಥವಾ ಸ್ವಾಧೀನಪಡಿಸಿಕೊಳ್ಳುವ ಬ್ಯಾಂಕ್ ನಡುವೆ ಮಾಹಿತಿಯ ವರ್ಗಾವಣೆಯನ್ನು ಸುಗಮಗೊಳಿಸುತ್ತದೆ.

ವಹಿವಾಟು ಪ್ರಕ್ರಿಯೆ (Transaction process)

- ಖರೀದಿದಾರರು, ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇ ಸಕ್ರೀಯಗೊಳಿಸಿದ ವ್ಯಾಪಾರಿಯಿಂದ ಒಂದು ಉತ್ಪನ್ನವನ್ನು ಆರ್ಡರ್ ಮಾಡಿದಾಗ, ಪೇಮೆಂಟ್ ಗೇಟ್‌ವೇ ವಹಿವಾಟನ್ನು ಪ್ರಕ್ರಿಯೆಗೊಳಿಸಲು ವಿವಿಧ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ವೈರಸ್‌ಗಳಿಂದ , ಸ್ಪೈವೇರ್‌ಗಳಿಂದ ಮತ್ತು ಇತರ ದುರುದ್ದೇಶಪೂರಿತ ಕೋಡ್‌ಗಳಂದ ಮಾಹಿತಿಯನ್ನು, ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ರಕ್ಷಿಸಿ (Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code)

ಮಾಹಿತಿಯ ಸುರಕ್ಷತೆ ಮತ್ತು ಬೆದರಿಕೆಗಳ ಅವಲೋಕನ (E-commerce scope and benefits)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಮಾಹಿತಿ ಸುರಕ್ಷತೆ ಮತ್ತು ಅದರ ಮೂಲ ತತ್ವಗಳನ್ನು ವಿವರಿಸಿ
- ಮಾಹಿತಿ ಸುರಕ್ಷತೆ ಮತ್ತು ಸೈಬರ್ ಸುರಕ್ಷತೆ ನಡುವಿನ ವ್ಯತ್ಯಾಸವನ್ನು ವಿವರಿಸಿ
- ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯಲ್ಲಿನ ಪ್ರಮುಖ ಸವಾಲುಗಳನ್ನು ವಿವರಿಸಿ
- ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯ ಲಾಭಗಳನ್ನು ವಿವರಿಸಿ
- ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸುವ ವಿಧಾನಗಳನ್ನು ವಿವರಿಸಿ.

ಪರಿಚಯ (Introduction)

ಮಾಹಿತಿ ಸುರಕ್ಷತೆ ಎಂದರೆ ಮಾಹಿತಿಯನ್ನು ಮತ್ತು ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಅನಧಿಕೃತ ಪ್ರವೇಶ, ಬಳಕೆ, ಬಹಿರಂಗಪಡಿಸುವಿಕೆ, ಅಡ್ಡಿಪಡಿಸುವಿಕೆ, ಮಾರ್ಪಾಡು ಮಾಡುವಿಕೆ, ಅಥವಾ ನಾಶಪಡಿಸುವಿಕೆ ಮುಂತಾದವುಗಳಿಂದ ರಕ್ಷಿಸುತ್ತದೆ ಮತ್ತು ಇದು ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಯಾವುದೇ ರೂಪವನು ್ನ(ಇಲೆಕ್ಟ್ರಾನಿಕ್, ಭೌತಿಕ) ಲೆಕ್ಕಿಸದೇ ಬಳಸಬಹುದಾದ ಸಾಮಾನ್ಯವಾದ ಪದವಾಗಿದೆ.

ಸರ್ಕಾರಗಳು, ಮಿಲಿಟರಿ, ನಿಗಮಗಳು, ಹಣಕಾಸು ಸಂಸ್ಥೆಗಳು, ಆಸ್ಪತ್ರೆಗಳು, ಮತ್ತು ಖಾಸಗಿ ವ್ಯವಹಾರಗಳು ಹೆಚ್ಚಿನ ಪ್ರಮಾಣದಲ್ಲಿ ತಮ್ಮ ಉದ್ಯೋಗಿಗಳ , ಗ್ರಾಹಕರ, ಉತ್ಪನ್ನಗಳ, ಸಂಶೋಧನೆಗಳ ಮತ್ತು ಆರ್ಥಿಕ ಸ್ಥಿತಿಗಳ ಬಗ್ಗೆ ಗೌಪ್ಯ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿರುತ್ತವೆ. ಇದರಲ್ಲಿ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಈಗ ಇಲೆಕ್ಟ್ರಾನಿಕ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಒಟ್ಟುಗೂಡಿಸುವುದು, ಸಂಸ್ಕರಿಸುವುದು ಮತ್ತು ಸಂಗ್ರಹಿಸುವುದನ್ನು ಮಾಡಲಾಗುತ್ತದೆ. ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳಾದ್ಯಂತ ಇತರ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ರವಾನೆ ಮಾಡಲಾಗುತ್ತದೆ.

ವ್ಯಾಪಾರದಲ್ಲಿನ ಗ್ರಾಹಕರ ಬಗ್ಗೆ ಅಥವಾ ಹಣಕಾಸು ಅಥವಾ ಹೊಸ ಉತ್ಪನ್ನದ ಬಗೆಗಿನ ಗೌಪ್ಯ ಮಾಹಿತಿಯು ಪ್ರತಿಸ್ಪರ್ಧಿಯ ಕೈಗೆ ಅಥವಾ ಹ್ಯಾಕರ್ ಕೈಗೆ ಸಿಕ್ಕರೆ ಆ ವ್ಯಾಪಾರ ಮತ್ತು ಅದರ ಗ್ರಾಹಕರು ವ್ಯಾಪಕವಾದ ಸರಿಪಡಿಸಲಾಗದ ಆರ್ಥಿಕ ನಷ್ಟವನ್ನು ಅನುಭವಿಸುತ್ತಾರೆ. ಇದು ಕಂಪನಿಯ ಖ್ಯಾತಿಗೂ ಹಾನಿಯನ್ನುಂಟು ಮಾಡುತ್ತದೆ. ಗೌಪ್ಯ ಮಾಹಿತಿಯನ್ನು ರಕ್ಷಿಸುವುದು ವ್ಯಾಪಾರದ ಅವಶ್ಯಕತೆಯಾಗಿದೆ ಮತ್ತು ಅನೇಕ ಸಂದರ್ಭಗಳಲ್ಲಿ ನೈತಿಕ ಮತ್ತು ಕಾನೂನು ಅವಶ್ಯಕತೆ ಆಗಿದೆ.

ಮಾಹಿತಿ ಭರವಸೆ (Information assurance): ಃ ಮಾಹಿತಿ ಭರವಸೆ ಎಂದರೆ, ಆಪತ್ತಿನ ಸಮಸ್ಯೆಗಳು ಉದ್ಭವಿಸಿದಾಗ ಡಾಟಾ ಕಳೆದುಹೋಗುವುದಿಲ್ಲ ಎಂದು ಖಚಿತಪಡಿಸುವ ಒಂದು ಕಾಯಿದೆ ಆಗಿದೆ. ಈ ಸಮಸ್ಯೆಗಳು ನೈಸರ್ಗಿಕ ವಿಪತ್ತುಗಳನ್ನು, ಕಂಪ್ಯೂಟರ್/ಸರ್ವರ್‌ನ ಅಸಮರ್ಪಕ ಕಾರ್ಯನಿರ್ವಹಣೆ, ಭೌತಿಕ ಕಳ್ಳತನ, ಅಥವಾ ಇತರ ಯಾವುದೇ ಡಾಟಾ ಕಳೆದುಹೋಗುವ ಸಂಭಾವ್ಯತೆಗಳನ್ನು ಒಳಗೊಂಡಿವೆ ಆದರೆ ಅವುಗಳಿಗೆ ಸೀಮಿತವಾಗಿಲ್ಲ. ನಮ್ಮ ಅಧುನಿಕ ಯುಗದಲ್ಲಿ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸುವುದರಿಂದ, ಮಾಹಿತಿಯ ಭರವಸೆಯನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಐಟಿ ಭದ್ರತಾ ತಜ್ಞರು ನಿರ್ವಹಿಸುತ್ತಾರೆ. ಮೇಲೆ ಉಲ್ಲೇಖಿಸಲಾದ ಯಾವುದೇ ಸಮಸ್ಯೆಗಳ ಸಂದರ್ಭದಲ್ಲಿ, ಮಾಹಿತಿ ಭರವಸೆಯನ್ನು ಒದಗಿಸುವ ವಿಧಾನಗಳಲ್ಲಿ ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾದ ವಿಧಾನವೆಂದರೆ ಆಫ್ ಸೈಟ್ ಬ್ಯಾಕಪ್‌ಅನ್ನು ಹೊಂದಿರುವುದು.

ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯ ಮೂಲ ತತ್ವಗಳು (Basic principles of Information Security) : CIA Triad ಒಂದು ಸುರಕ್ಷತಾ ನೀತಿ ಅಭಿವೃದ್ಧಿಗೆ ಪ್ರಸಿದ್ಧ ಮಾದರಿಯಾಗಿದೆ. ಇದನ್ನು ಸಮಸ್ಯೆ ಇರುವ ಸ್ಥಳವನ್ನು ಗುರುತಿಸಲು ಮತ್ತು ಮಾಹಿತಿಯ ಸುರಕ್ಷತೆಗೆ ಅಗತ್ಯವಾದ ಪರಿಹಾರಗಳನ್ನು ಗುರುತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಗೌಪ್ಯತೆ , ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆ (Confidentiality, integrity, and availability (CIA)) ಈ ಮಾದರಿಯನ್ನು ಸಂಸ್ಥೆಯ ಮಾಹಿತಿ ಸುರಕ್ಷತೆಗೆ ಮಾರ್ಗದರ್ಶನ ನೀಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಗೌಪ್ಯತೆ , ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆಯ CIA Triad ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯ ಹೃದಯದಲ್ಲಿದೆ. Classic Information Security triad - confidentiality, integrity and availability - ಇದರ ಸದಸ್ಯರನ್ನು - ಸುರಕ್ಷತಾ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು, ಗುಣಲಕ್ಷಣಗಳು, ಸುರಕ್ಷತಾ ಗುರಿಗಳು, ಮೂಲಭೂತ ಅಂಶಗಳು, ಮಾಹಿತಿ ಮಾನದಂಡಗಳು, ನಿರ್ಣಾಯಕ ಮಾಹಿತಿ ಗುಣಲಕ್ಷಣಗಳು ಮತ್ತು ಮೂಲ ಬಿಲ್ಡಿಂಗ್ ಬ್ಲಾಕ್‌ಗಳು ಎಂದೂ ಸಹ ಉಲ್ಲೇಖಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಗೌಪ್ಯತೆ (Confidentiality)

ಗೌಪ್ಯತೆಯು, ಮಾಹಿತಿಯ ಪ್ರವೇಶವನ್ನು ಮಿತಿಗೊಳಿಸುವ ನಿಯಮಗಳ ಒಂದು ಗುಂಪಾಗಿದೆ. ಗೌಪ್ಯತೆಯು ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯು ತಪ್ಪು ಜನರ ಕೈಗೆ ತಲುಪುವುದನ್ನು ತಡೆಯುತ್ತದೆ ಮತ್ತು ವಾಸ್ತವವಾಗಿ ಮಾಹಿತಿಯು ಸರಿಯಾದ ಜನರನ್ನು ತಲುಪಿರುವುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ. ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸುವುದು - ಮಾಹಿತಿಯ ಸೂಕ್ತವಾದ ಪ್ರವೇಶ ಮಟ್ಟಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು ಮತ್ತು ಜಾರಿಗೊಳಿಸುವುದನ್ನು ಅವಲಂಬಿಸಿವೆ. ಇದನ್ನು ಮಾಹಿತಿಯನ್ನು ಪ್ರತ್ಯೇಕ ಘಟಕಗಳಲ್ಲಿ ಬೇರ್ಪಡಿಸುವ ಮೂಲಕ ಮಾಡಬಹುದು. ಈ ಮಾಹಿತಿ ಘಟಕಗಳು - ಯಾರು ಪ್ರವೇಶವನ್ನು ಹೊಂದಬಹುದು ಮತ್ತು ಅದು ಎಷ್ಟು ಸೂಕ್ಷ್ಮವಾಗಿದೆ ಎಂಬುದಾಗಿದೆ.

ಸಮಗ್ರತೆ (Integrity)

ಸಮಗ್ರತೆ ಎಂದರೆ ಮಾಹಿತಿಯ ವಿಶ್ವಾಸಾರ್ಹವಾಗಿದೆ , ಸ್ಥಿರವಾಗಿದೆ, ಮತ್ತು ನಿಖರವಾಗಿದೆ ಎನ್ನುವ ಭರವಸೆಯಾಗಿದೆ. ಇದರ ಅರ್ಥ ಡಾಟಾವನ್ನು ಅನಧಿಕೃತವಾಗಿ ಅಥವಾ ಪತ್ತೆಯಾಗದ ರೀತಿಯಲ್ಲಿ ಮಾರ್ಪಾಡು ಮಾಡಲಾಗುವುದಿಲ್ಲ. ಡಾಟಾ ರವಾನೆಯಾಗುವಾಗ ಬದಲಾಗಬಾರದು ಮತ್ತು ಅನಧಿಕೃತ ಜನರಿಂದ ಡಾಟಾ ಬದಲಾಗದಂತೆ ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಬೇಕು. (ಉದಾಹರಣೆಗೆ, ಗೌಪ್ಯತೆಯ ಉಲ್ಲಂಘನೆಯಲ್ಲಿ) ಜೊತೆಗೆ, ಮಾನವನು ಮಾಡದೆ ಇರುವ ಕೆಲವು ಘಟನೆಗಳಲ್ಲಿ ಎಂದರೆ ಇಲೆಕ್ಟ್ರೋಮ್ಯಾಗ್ನೆಟಿಕ್ ಪಲ್ಸ್ (EMP) ಅಥವಾ ಸರ್ವರ್ ಕ್ರ್ಯಾಶ್‌ನಂತಹ ಪರಿಣಾಮವಾಗಿ ಡಾಟಾದಲ್ಲಿನ ಯಾವುದೇ ಬದಲಾವಣೆಗಳನ್ನು ಪತ್ತೆಹಚ್ಚಲು ಕೆಲವು ಸಾಧನಗಳು ಇರಬೇಕು. ಅನಿರೀಕ್ಷಿತ ಬದಲಾವಣೆಯು ಸಂಭವಿಸಿದಲ್ಲಿ, ಹಾನಿಯಾದ ಡಾಟಾವನ್ನು ಅದರ ಸರಿಯಾದ ಸ್ಥಿತಿಗೆ ಮರುಸ್ಥಾಪಿಸಲು ಒಂದು ಬ್ಯಾಕಪ್ ಕಾಪಿ ಲಭ್ಯ ಇರಬೇಕು.

ಲಭ್ಯತೆ (Availability)

ಲಭ್ಯತೆ ಎಂದರೆ ಅಧಿಕೃತ ಜನರು ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಸಿದ್ಧವಿರುವ ಬಗ್ಗೆ ಖಾತರಿಯಾಗಿದೆ. ಯಾವುದೇ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗೆ ಅದರ ಉದ್ದೇಶವನ್ನು ಪೂರೈಸಲು, ಅಗತ್ಯವಿದ್ದಾಗ ಮಾಹಿತಿಯು ಲಭ್ಯವಿರಬೇಕು. ಇದರ ಅರ್ಥ ಸಂಗ್ರಹಿಸಲು ಮತ್ತು ಪ್ರಕ್ರಿಯೆಗೊಳಿಸಲು ಉಪಯೋಗಿಸುವ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು, ಅದನ್ನು ರಕ್ಷಿಸಲು ಬಳಸುವ ಸುರಕ್ಷತಾ ನಿಯಮಗಳು, ಮತ್ತು ಅದನ್ನು ಪ್ರವೇಶಿಸಲು ಬಳಸುವ ಸಂವಹನ ಚಾನಲ್‌ಗಳು ಸರಿಯಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿರಬೇಕು.

ಉನ್ನತ ಮಟ್ಟದ ಲಭ್ಯತೆಯ ವ್ಯವಸ್ಥೆಗಳು ಎಲ್ಲ ಸಮಯದಲ್ಲಿಯೂ ಲಭ್ಯವಾಗಿರುವ ಗುರಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇವು ವಿದ್ಯುತ್ ನಿಲುಗಡೆಯಿಂದಾಗಿ, ಹಾರ್ಡ್‌ವೇರ್ ವೈಫಲ್ಯದಿಂದಾಗಿ ಮತ್ತು ಸಿಸ್ಟಮ್ ನವೀಕರಣದಿಂದಾಗಿ ಸೇವೆಯ ಅಡೆತಡೆಗಳನ್ನು ತಡೆಗಟ್ಟುತ್ತವೆ. ಎಲ್ಲ ಹಾರ್ಡ್‌ವೇರ್‌ಗಳನ್ನು ಸರಿಯಾಗಿ ನಿರ್ವಹಿಸುವುದರಿಂದ ಮತ್ತು ಅಗತ್ಯವಿದ್ದಾಗ ತಕ್ಷಣ ಹಾರ್ಡ್‌ವೇರ್ ರಿಪೇರಿ ಮಾಡುವ ಮೂಲಕ ಲಭ್ಯತೆಯನ್ನು ಖಾತರಿಪಡಿಸಲಾಗುತ್ತದೆ. ಅಗತ್ಯತೆ ಮತ್ತು ವೈಫಲ್ಯದ ನಿರ್ದಿಷ್ಟ ಅಳತೆಯನ್ನು ಒದಗಿಸುವುದರಿಂದ, ಸಾಕಷ್ಟು ಸಂವಹನ ಬ್ಯಾಂಡ್‌ವಿಡ್ತನ್ನು ಒದಗಿಸುವುದರಿಂದ ಮತ್ತು ಅಡಚಣೆಗಳನ್ನು ತಡೆಗಟ್ಟುವುದರಿಂದ, ಬ್ಯಾಕ್‌ಅಪ್ ಪವರ್ ಸಿಸ್ಟಮ್‌ಗಳ ತುರ್ತು ಅನುಷ್ಠಾನ, ಅಗತ್ಯವಿರುವ ಎಲ್ಲವನ್ನೂ ಸಿಸ್ಟಮ್ ನವೀಕರಣಗಳೊಂದಿಗೆ ಇಟ್ಟುಕೊಳ್ಳುವುದು ಮತ್ತು denial-of-service (DoS) ದಾಳಿಗಳಂತಹ ದುರುದ್ದೇಶಪೂರಿತ ಕ್ರಿಯೆಗಳ ವಿರುದ್ಧ ರಕ್ಷಣೆ ಮುಂತಾದವುಗಳಿಂದ ಲಭ್ಯತೆಯನ್ನು ಖಚಿತಪಡಿಸಲಾಗುತ್ತದೆ. ಜೊತೆಗೆ, ಮೇಲೆ ತಿಳಿಸಿದ ಮೂರೂ ಸದಸ್ಯರ ಜೊತೆಗೆ ದೃಢೀಕರಣ ಮತ್ತು ನಿರಾಕರಣೆಯನ್ನೂ ಸಹ CIA ಮಾದರಿಯ ಸದಸ್ಯರನ್ನಾಗಿ ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ದೃಢೀಕರಣ (Authenticity)

ದೃಢೀಕರಣವು ಡಾಟಾ, ವಹಿವಾಟುಗಳು, ಸಂವಹನಗಳು ಅಥವಾ ದಾಖಲೆಗಳು ನಿಜವಾಗಿವೆಯೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ದೃಢೀಕರಣಕ್ಕೆ ಅದರಲ್ಲಿ ಭಾಗವಹಿಸಿದ ಅಥವಾ ಒಳಗೊಂಡಿರುವ ಪಾರ್ಟೀ(ಪಕ್ಷ) ಗಳು ನಿಜವಾಗಿವೆಯೆ ಎಂದು ಮೌಲ್ಯೀಕರಿಸುವುದು ಸಹ ಮುಖ್ಯವಾಗಿರುತ್ತದೆ. ಕೆಲವು ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ವ್ಯವಸ್ಥೆಗಳು "digital signatures" ನಂತಹ ದೃಢೀಕರಣ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ಇದು ಸಂದೇಶ ಕಳುಹಿಸಲಾದ ಡಾಟಾ ನೈಜವಾಗಿದೆ ಎಂಬುದಕ್ಕೆ ಮತ್ತು ಯಾರೋ ಸರಿಯಾದ ಸಹಿ ಕೀಯನ್ನು ಹೊಂದಿರುವವರು ಕಳುಹಿಸಿದ್ದಾರೆ ಎನ್ನುವ ಪುರಾವೆಯನ್ನು ನೀಡುತ್ತದೆ.

ನಾನ್ -ರೆಪ್ಯೂಡಿಯೇಷನ್ (Non-repudiation)

ಅನಂಗೀಕಾರ (Non-repudiation) ಎಂದರೆ ಒಂದು ಒಪ್ಪಂದಕ್ಕೆ ಬಾಧ್ಯತೆಗಳನ್ನು ಪೂರೈಸುವ ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ಉದ್ದೇಶವಾಗಿದೆ. ವಹಿವಾಟಿನ ಒಂದು ಪಕ್ಷವು ವಹಿವಾಟನ್ನು ಸ್ವೀಕರಿಸಿಲ್ಲ ಎಂದು ಹೇಳಲು ಸಾಧ್ಯವಿಲ್ಲ ಅಥವಾ ಇನ್ನೊಂದು ಪಕ್ಷವು ವಹಿವಾಟು ಕಳುಹಿಸಿರುವುದನ್ನು ನಿರಾಕರಿಸುವಂತಿಲ್ಲ.

ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಕ್ ಸಿಸ್ಟಮ್‌ನಂತಹ ತಂತ್ರಜ್ಞಾನಗಳು ಅನಂಗೀಕಾರದ ಪ್ರಯತ್ನಕ್ಕೆ ಸಹಾಯ ಮಾಡುವುದನ್ನು ಗಮನಿಸುವುದು ಮುಖ್ಯವಾಗಿದೆ. ಈ ಪರಿಕಲ್ಪನೆಯು ಮೂಲತಃ ಕಾನೂನು ಪರಿಕಲ್ಪನೆಯಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ಸಂದೇಶವು ಅದನ್ನು ಕಳುಹಿಸಿದವರ ಖಾಸಗಿ ಕೀಯೊಂದಿಗೆ ಮಾಡಿದ ಡಿಜಿಟಲ್ ಸಹಿಯೊಂದಿಗೆ ಹೊಂದಿಕೆಯಾಗುತ್ತದೆ ಎನ್ನುವುದು ಸಾಕಾಗುತ್ತದೆ. ಮತ್ತು ಹೀಗಾಗಿ ಸೆಂಡರ್ ಮಾತ್ರ ಈ ಸಂದೇಶವನ್ನು ಕಳುಹಿಸಬಹುದು ಮತ್ತು ರವಾನಿಸಲ್ಪಡುವಾಗ ಬೇರೆ ಯಾರೂ ಅದನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಆರೋಪಿಸಿದಂತ ಸೆಂಡರ್

© NIMI NOT TO BE REPUBLISHED

ಪ್ರತಿಯಾಗಿ ಡಿಜಿಟಲ್ ಸಹಿಯ ಅಲ್ಗೋರಿದಮ್ ದುರ್ಬಲವಾಗಿದೆ ಅಥವಾ ದೋಷಪೂರಿತವಾಗಿದೆ ಎಂದು ವಿವರಿಸಬೇಕಾಗುತ್ತದೆ. ಅಥವಾ ಅವರ ಸಹಿ ಮಾಡುವ ಕೀಲಿಯನ್ನು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳಲಾಗಿದೆ ಎಂದು ಸಾಬಿತುಪಡಿಸಬೇಕಾಗುತ್ತದೆ.

ಈ ದೋಷ ಉಲ್ಲಂಘನೆಗಳ ತಪ್ಪಿಗೆ ಕಾರಣ ಕಳುಹಿಸುವವರ ಬಳಿಯೆ ಇರಬಹುದು ಅಥವಾ ಇಲ್ಲದಿರಬಹುದು. ಮತ್ತು ಅಂತಹ ಸಮರ್ಥನೆಗಳು ಸೆಂಡರ್‌ನ್ನು ಹೊಣೆಗಾರಿಕೆಯಿಂದ ಉಪಶಮನಗೊಳಿಸಬಹುದು ಅಥವಾ ಇಲ್ಲದಿರಬಹುದು. ಆದರೆ ಸಹಿಯು ಅವಶ್ಯವಾಗಿ ದೃಢೀಕರಣವನ್ನು ಮತ್ತು ಸಮಗ್ರತೆಯನ್ನು ಸಾಬೀತುಪಡಿಸುತ್ತದೆ. ಮತ್ತು ನಿರಾಕರಣೆಯನ್ನು ತಡೆಯುತ್ತದೆ.

ಹೆಚ್ಚಾಗಿ ನಮಗೆ ಅನಿಯಮಿತ ಸ್ವಾತಂತ್ರವನ್ನು ನೀಡುವ ಎಲ್ಲಾ ಚಟುವಟಿಕೆಗಳು, ಅಪಾಯವನ್ನೂ ತರುತ್ತವೆ. ಏಕೆಂದರೆ ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಸುಲಭವಾಗಿ ಯಾರಾದರು ಪ್ರವೇಶಿಸಬಹುದು., ಇದು ಅಪಾಯಕಾರಿ ಸ್ಥಳವಾಗಿದೆ. ನೀವು ಯಾರೊಂದಿಗೆ ವ್ಯವಹರಿಸುತ್ತಿದ್ದೀರಿ ಎಂಬುದನ್ನು ತಿಳಿಯಿರಿ ಅಥವಾ ನೀವು ಏನನ್ನು ಪಡೆಯುತ್ತಿದ್ದೀರಿ ಎಂದು ತಿಳಿಯಿರಿ. ಪ್ರೆಡೇಟರ್‌ಗಳು, ಸೈಬರ್ ಅಪರಾಧಿಗಳು , ಬೆದರಿಸುವವರು ಮತ್ತು ಭ್ರಷ್ಟ ವ್ಯಾಪಾರಿಗಳು ಅಜಾಗರೂಕ ಸಂದರ್ಶಕರ ಲಾಭವನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಪ್ರಯತ್ನಿಸುತ್ತಾರೆ.

ಇಂಟರ್‌ನೆಟ್, ಇಂಟ್ರಾನೆಟ್ ಮತ್ತು ಭದ್ರತೆ (The internet, Intranet and Security)

ತಂತ್ರಜ್ಞಾನದ ವಿಷಯದಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್ ಮತ್ತು ಇಂಟ್ರಾನೆಟ್‌ಗಳ ನಡುವಿನ ವ್ಯಾತ್ಯಾಸ ದೊಡ್ಡದಾಗಿಲ್ಲ ಆದರೆ ವ್ಯವಸ್ಥೆಗೆ ಸಂಬಂಧಿಸಿದ ದೃಷ್ಟಿಯಿಂದ, ಮಾಹಿತಿ ಪ್ರಸರಣ ವಿಷಯದಲ್ಲಿ ಸಂಪೂರ್ಣವಾಗಿ ವಿಭಿನ್ನವಾಗಿದೆ.

ಇಂಟ್ರಾನೆಟ್‌ಗಳು ಮತ್ತು ಇತರ ನೆಟ್‌ವರ್ಕ್‌ಗಳು ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳ ನಡುವಿನ ಮಾಹಿತಿ ಭದ್ರತಾ ಬೆದರಿಕೆಗಳು ಒಂದನ್ನೊಂದು ಹೋಲುತ್ತವೆ. ಇಂಟ್ರಾನೆಟ್‌ಗಳನು ಆಂತರಿಕ ಮಾಹಿತಿ ಚಾನಲ್‌ಗಳನ್ನಾಗಿ ಉಪಯೋಗಿಸುವುದು ಮಾಹಿತಿ ಭದ್ರತೆಯ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ಒತ್ತಿಹೇಳುತ್ತದೆ. ಆಂತರಿಕ ಇಂಟ್ರಾನೆಟ್‌ಗಳ ಸ್ವತ್ತುಗಳು ಸಂಭಾವ್ಯ ದುರ್ಬಳಕೆದಾರರ ಆಸಕ್ತಿಯನ್ನು ಹೆಚ್ಚಿಸಬಹುದು. ಅದ್ದರಿಂದ ಇಂಟ್ರಾನೆಟ್‌ಗಳನ್ನು ಮತ್ತು ಅವುಗಳ ಮೂಲಕ ರವಾನಿಸಲಾಗುವ ಡಾಟಾ ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಅದರ ಗೌಪ್ಯತೆ , ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆ ಗೆ ಅಪಾಯವನ್ನು ಉಂಟುಮಾಡುವ ವಿವಿಧ ಬೆದರಿಕೆಗಳಿಂದ ರಕ್ಷಿಸುವುದು ಮುಖ್ಯವಾಗುತ್ತದೆ.

ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಇಂಟ್ರಾನೆಟ್‌ನ ಅ ಒಂದು ಭಾಗವಾಗಿ ಉಪಯೋಗಿಸುವುದು ಗಂಭೀರವಾದ ಬೆದರಿಕೆಯಾಗಿದೆ. ಏಕೆಂದರೆ ಮೂಲಸ್ವರೂಪವಾಗಿ ಅಸುರಕ್ಷಿತವಾಗಿದೆ. ಪರಿಣಾಮವಾಗಿ, ಬಳಕೆದಾರರು ವಿಶೇಷವಾಗಿ ತಮ್ಮ ಸಂವಹನವನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡುವಾಗ ಬಹಳ ಜಾಗರೂಕರಾಗಿರಬೇಕು. ಅನುಕರಣೆ (spoofing), ಪ್ರತ್ಯುತ್ತರ (rapid fire), ಸಂದೇಶದ ವಿಷಯಗಳ ಬದಲಾವಣೆ (superzapping), ಸೇವೆಯ ಲಭ್ಯತೆಯನ್ನು ತಡೆಗಟ್ಟುವುದು ಮತ್ತು active and passive ವೈರ್ ಟ್ಯಾಪಿಂಗ್ ಮುಂತಾದವು ಅತ್ಯಂತ ದುರುದ್ದೇಶಪೂರಿತ ಬೆದರಿಕೆಗಳಲ್ಲಿ ಸೇರಿವೆ. ಉದಾಹರಣೆಗೆ ವೈರ್ ಟ್ಯಾಪಿಂಗ್ - ಬಲವಾದ ಎನ್‌ಕ್ರಿಪ್ಷನ್ ವಿಧಾನಗಳ ಮೂಲಕ ಸಂವಹನ ಎನ್‌ಕ್ರಿಪ್ಷನ್‌ನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸದಿದ್ದರೆ, ಒಂದು ಸಂಸ್ಥೆಯ ಒಳಗಿನ ಕಾರ್ಯತಂತ್ರದ ಜ್ಞಾನವು ಹೊರಗಿನವರ ಕೈಗೆ ಸಿಗುವ ಪರಿಸ್ಥಿತಿಗೆ ಕಾರಣವಾಗಬಹುದು.

ಹ್ಯಾಕರ್ ಸಾಧನಗಳು - ಇಂಟರ್‌ನೆಟ್‌ಗಾಗಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ್ದರೂ, ಅವುಗಳನ್ನು ಇಂಟ್ರಾನೆಟ್‌ಗಳಲ್ಲಿ ಬಳಸಬಹುದು. ಅವು ಸಾಫ್ಟ್‌ವೇರ್ ಅಥವಾ ಹಾರ್ಡ್‌ವೇರ್ ಆಧಾರಿತವಾಗಿರುತ್ತವೆ ಅಥವಾ ಇವೆರಡರ ಸಂಯೋಜನೆಯೂ ಆಗಿರಬಹುದು. ಅವುಗಳ ಅಧಿಕೃತ ಬಳಕೆಯು, ಇಂಟ್ರಾನೆಟ್‌ನಲ್ಲಿ ಮಾಹಿತಿ ಭದ್ರತಾ ದುರ್ಬಲತೆಗಳನ್ನು ಕಂಡುಹಿಡಿಯುವುದು ಮತ್ತು ಸರಿಪಡಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಅದಾಗ್ಯೂ ಅವು ಒಳಗಿನವರು ಇಂತಹ ಸಂವಹನ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಹ್ಯಾಕ್ ಮಾಡಲು ಮತ್ತು ಅವರು ಅಧಿಕಾರ ಹೊಂದದೇ ಇರುವ ಮಾಹಿತಿಯನ್ನು ಪ್ರವೇಶಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತವೆ.

ಮಾಹಿತಿ ಭದ್ರತೆ ಮತ್ತು ಸೈಬರ್ ಭದ್ರತೆ ನಡುವಿನ ಸಂಬಂಧ(Relation Between information Security and Cyber security)

ಹಿಂದಿನ ಪಾಠಗಳಲ್ಲಿ ತಿಳಿಸಿದಂತೆ ಮಾಹಿತಿ ಭದ್ರತೆ ಎಂದರೆ ಮಾಹಿತಿಯನ್ನು ಮತ್ತು ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಅನಧಿಕೃತ ಪ್ರವೇಶ, ಬಳಕೆ, ಬಹಿರಂಗಪಡಿಸುವಿಕೆ, ಅಡ್ಡಿಪಡಿಸುವಿಕೆ, ಮಾರ್ಪಾಡು ಮಾಡುವಿಕೆ, ಅಥವಾ ವಿನಾಶ ಮಾಡುವಿಕೆ ಮುಂತಾದುವುಗಳಿಂದ ರಕ್ಷಣೆ ಮಾಡಿ ಅದರ ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆಯನ್ನು ಒದಗಿಸುವುದು ಆಗಿದೆ.

ಮತ್ತೊಂದೆಡೆ ಸೈಬರ್ ಭದ್ರತೆ ಎಂದರೆ ಸೈಬರ್ ಸ್ಪೇಸನ್ನು ಸೈಬರ್ ದಾಳಿಗಳಿಂದ ರಕ್ಷಿಸುವ ಸಾಮರ್ಥ್ಯ ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದಾಗಿದೆ. ಸೈಬರ್ ಸ್ಪೇಸ್ ಎಂದರೆ " ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ನಡೆಯುವ ಸಂವಹನದ ಪರಿಸರವಾಗಿದೆ"

ಸೈಬರ್ ಭದ್ರತೆಯು ಸೈಬರ್ ಡೊಮೇನ್‌ನಲ್ಲಿ ಅಥವಾ ರೀಲ್ಮ್ (ಅಥವಾ ಸೈಬರ್ ಸ್ಪೇಸ್) ನಲ್ಲಿ ಭದ್ರತೆಗೆ ಸಂಬಂಧಿಸಿದ ಯಾವುದನ್ನಾದರು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಮಾಹಿತಿ ಭದ್ರತೆಯು, ಮಾಹಿತಿ ಅಥವಾ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಯ ಭದ್ರತೆಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ(ಅದು ಯಾವ ರೀಲ್ಮ್ ನಲ್ಲಿ ಸಂಭವಿಸುತ್ತದೆ ಎಂದು ಲೆಕ್ಕಿಸದೆಯೆ- ಉದಾಹರಣೆಗೆ, ಭೌತಿಕ ಜಗತ್ತಿನಲ್ಲಿ ತೆರೆದಿಡುವ ಅಪಾಯ) ಅದ್ದರಿಂದ ಸೈಬರ್ ರೀಲ್ಮ್‌ನಲ್ಲಿ ಏನಾದರು ಸಂಭವಿಸುವ ಯಾವುದಾದರೂ ಮಾಹಿತಿ ಮತ್ತು ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಯ ರಕ್ಷಣೆಯನ್ನು ಒಳಗೊಂಡಿರಬೇಕು.

© NIMI NOT TO BE REPUBLISHED

ಮಾಹಿತಿ ಭದ್ರತೆಯು, ಸೈಬರ್ ಭದ್ರತೆಯ ಸೂಪರ್ ಸೆಟ್ ಆಗಿದೆ ಎಂದು ನೀವು ತೀರ್ಮಾನಿಸಬಹುದು. (Fig 1) ಕೆಲವೊಮ್ಮೆ ಎರಡೂ ಶಬ್ದಗಳನ್ನು ಪರಸ್ಪರ ಬದಲಿಯಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯಲ್ಲಿನ ಪ್ರಮುಖ ಸವಾಲುಗಳು (Key Challenges in Information Security)

1 **IT ಸುರಕ್ಷೆತೆಗೆ ಕಡಿಮೆ ಆದ್ಯತೆ ನೀಡಲಾಗಿದೆ**

- ಸಂಸ್ಥೆಯು ಐಟಿ ಭದ್ರತಾ ಅಭ್ಯಾಸಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸುವ ಬಗ್ಗೆ ಸರಿಯಾದ ಗಮನವನ್ನು ಪ್ರತಿಷ್ಠಾಪಿಸಿಲ್ಲ.

2 **ತಾತ್ಕಾಲಿಕ ಸುರಕ್ಷತಾ ಆಡಳಿತ**

- Information Security Management System (ISMS) (ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ನಿರ್ವಹಣೆಯ) ಅಥವಾ ರಚನಾತ್ಮಕ ಆಡಳಿತ ಯಾಂತ್ರಿಕ ವ್ಯವಸ್ಥೆಯ ಅನುಪಸ್ಥಿತಿ (Fig 1)

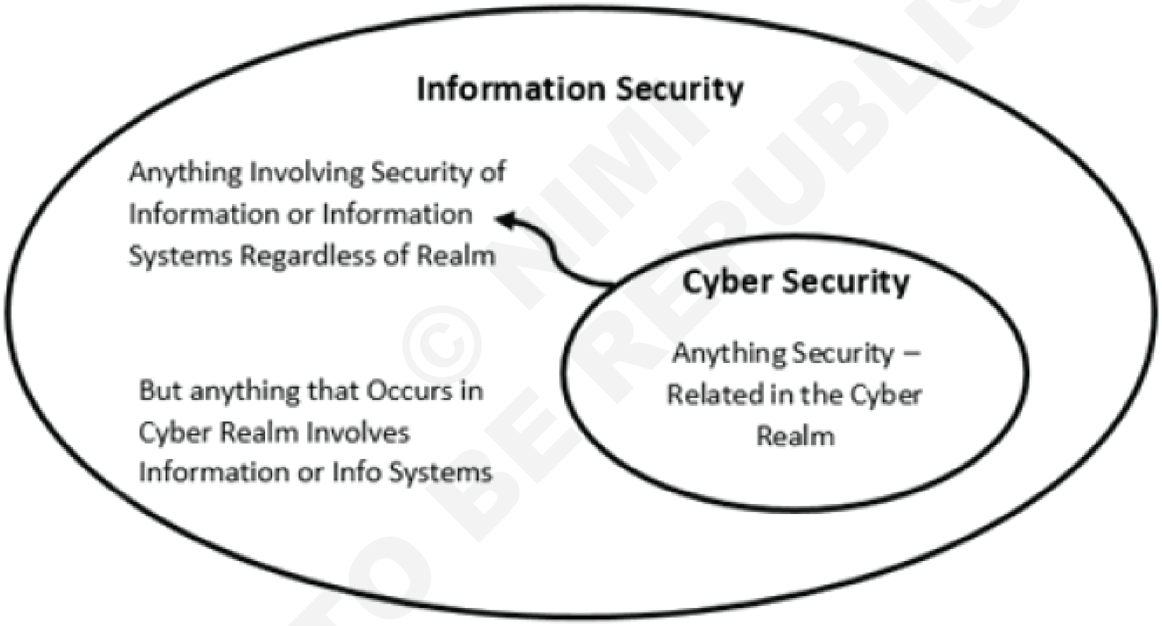
3 **ಪಾತ್ರಗಳು ಮತ್ತು ಜವಾಬ್ದಾರಿಗಳಲ್ಲಿ ಅಸ್ಪಷ್ಟತೆ**

- SSO ನಲ್ಲಿನ ವಿವಿಧ ಆಟಗಾರರ (ವ್ಯಾಪಾರ, SSO ನ ತಂಡಗಳು ಇತ್ಯಾದಿ) ಪಾತ್ರಗಳು ಮತ್ತು ಜವಾಬ್ದಾರಿಗಳು ಅಸ್ಪಷ್ಟವಾಗಿವೆ. Single sign-on (SSO) ಒಂದು ಗುಣಲಕ್ಷಣವಾಗಿದೆ. ಇದು ಬಹು ಸಂಬಂಧಿತ ಆದರೆ ಸ್ವತಂತ್ರ ಸಾಫ್ಟವೇರ್ ವ್ಯವಸ್ಥೆಯಾಗಿದೆ. ಈ ಗುಣಲಕ್ಷಣವನ್ನು ಬಳಸಿ ಬಳಕೆದಾರರು ಒಮ್ಮೆ ಲಾಗಿನ್ ಆಗಬಹುದು ಮತ್ತು ಎಲ್ಲ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಪ್ರವೇಶವನ್ನು ಪಡೆಯುತ್ತಾರೆ. ಪ್ರತಿಯೊಬ್ಬರೂ ಮತ್ತೊಮ್ಮೆ ಲಾಗಿನ್ ಆಗಬೇಕಾದ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

4 **ಕರ್ತವ್ಯಗಳ ಅಸಮರ್ಪಕ ಪ್ರತ್ಯೇಕತೆ**

- SSO ನಲ್ಲಿ ಅತಿಕ್ರಮಿಸುವ ಮತ್ತು ಜವಾಬ್ದಾರಿಗಳ ಹಂಚಿಕೆಯು ಕರ್ತವ್ಯಗಳಲ್ಲಿ ಪ್ರತ್ಯೇಕತೆಯ ಮಟ್ಟವನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಕಷ್ಟವಾಗಿಸುತ್ತದೆ.

Fig 1



Relation between Information Security and Cybersecurity

5 **ಭದ್ರತಾ ಅಗತ್ಯತೆಗಳ ವಿವಿಧ ವ್ಯಾಖ್ಯಾನಗಳು**

- ಪ್ರಮಾಣಿತ ವ್ಯಾಖ್ಯಾನಗಳ ಅನುಪಸ್ಥಿತಿಯಲ್ಲಿ, ವಿಭಿನ್ನ ವ್ಯಕ್ತಿಗಳು ಮತ್ತು ತಂಡಗಳು ತಮ್ಮದೇ ಆದ ಸ್ವಂತ ವ್ಯಾಖ್ಯಾನಗಳನ್ನು ಹೊಂದಿವೆ.

6 **ಅಪಾಯದ ಮಟ್ಟವನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಪ್ರವೃತ್ತಿ**

- ತಂಡಗಳು ಆಡಳಿತದ ಕಾರ್ಯವಿಧಾನಗಳ ಕಠಿಣತೆಯನ್ನು ದಾಟಿ ಹೊರಬರಲು ಅಪಾಯದ ಮಟ್ಟವನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಪ್ರವೃತ್ತಿಯನ್ನು ತೋರಿಸುತ್ತವೆ.

7 **ಬಹು ಮಾರಾಟಗಾರರು**

- ಪಟ್ಟುಬಿಡದ ಸ್ಪರ್ಧೆ ಮತ್ತು ಅಭದ್ರತೆಯ ಪ್ರಜ್ಞೆಯು ಜವಾಬ್ದಾರಿಯನ್ನು ಹಂಚಿಕೊಳ್ಳುವಲ್ಲಿ ಹಿಂಜರಿಕೆಯನ್ನು ಮತ್ತು ಮಾರಾಟಗಾರರ ನಡುವೆ ಸ್ವಲ್ಪ ಮಟ್ಟಿಗೆ ಅಥವಾ ಯಾವುದೇ ಸಹಯೋಗವಿಲ್ಲದಂತೆ ಆಗಲು ಕಾರಣವಾಯಿತು.

8 ವ್ಯಾಪಾರ/ಕಾರ್ಯಾಚರಣೆಗಳು ಭೌಗೋಳಿಕವಾಗಿ ಅನೇಕ ಸ್ಥಳಗಳಲ್ಲಿ ಹರಡಿವೆ.

- ಸಂಸ್ಥೆಯು ಪ್ರಪಂಚದಾದ್ಯಂತ ಹರಡಿರುವ ಅನೇಕ ಸ್ಥಳಗಳಿಂದ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

9 ತರಬೇತಿ/ಅರಿವಿನ ಕೊರತೆ

- ಭದ್ರತೆಯ ಅಭ್ಯಾಸದ ಬಗ್ಗೆ ಅಸಮರ್ಪಕ ತರಬೇತಿ ಮತ್ತು ತಿಳುವಳಿಕೆ.

ಮಾಹಿತಿ ಸುರಕ್ಷತೆಯ ಪ್ರಯೋಜನಗಳು (Benefits of Information Security):

- ನೆಟ್‌ವರ್ಕ್‌ಗಳು, ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಮತ್ತು ಡಾಟಾವನ್ನು ಅನಧಿಕೃತ ಪ್ರವೇಶದಿಂದ ರಕ್ಷಿಸಿ ವಿವಿಧ ಸೈಬರ್ ಅಪರಾಧಗಳ ಬಾಹ್ಯ ಬೆದರಿಕೆಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.
- ಸುಧಾರಿತ ಮಾಹಿತಿ ಭದ್ರತೆ ಮತ್ತು ವ್ಯಾಪಾರ ನಿರಂತತೆಯ ನಿರ್ವಹಣೆಯ ತಾಂತ್ರಿಕ, ನಿರ್ವಹಣಾತ್ಮಕ, ಆಡಳಿತಾತ್ಮಕ, ಮತ್ತು ಕಾರ್ಯಾಚರಣೆ ಮುಂತಾದವುಗಳ ನಿಯಂತ್ರಣೆಯನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ. ಇದು ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಅತ್ಯಂತ ವೆಚ್ಚ - ಪರಿಣಾಮಕಾರಿಯಾದ ಮಾರ್ಗವಾಗಿದೆ.
- ಮಾಹಿತಿ ಭದ್ರತೆಯ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಪಾಲುದಾರರ ವಿಶ್ವಾಸವನ್ನು ಸುಧಾರಿಸುತ್ತದೆ.
- ಸರಿಯಾದ ಭದ್ರತೆ ನಿಯಂತ್ರಣಗಳು ಕಂಪನಿಯ ಪರಿಚಯ ಪತ್ರಗಳನ್ನು ಸುಧಾರಿಸುತ್ತದೆ. ಇದರಿಂದ ಸಂಸ್ಥೆಯ ಆಂತರಿಕ ಮಧ್ಯಸ್ಥಗಾರರು ಮತ್ತು ಬಾಹ್ಯ ಮಾರಾಟಗಾರ ನಡುವೆ ನಂಬಿಕೆಯನ್ನು ಸುಧಾರಿಸುತ್ತದೆ. ನಂಬಿಕೆ ಮತ್ತು ವಿಶ್ವಾಸಗಳು ಒಂದು ವ್ಯಾಪಾರವನ್ನು ಗೆಲ್ಲಲು ಪ್ರಮುಖ ಅಂಶಗಳಾಗಿವೆ.
- ಅಡಚಣೆಯ ಸಂದರ್ಭದಲ್ಲಿ ವೇಗವಾಗಿ ಚೇತರಿಸಿಕೊಳ್ಳುವ ಸಮಯ ನೀಡುತ್ತದೆ.

ಸಂಸ್ಥೆಯಲ್ಲಿ IS ನ್ನು ಜಾರಿಗೊಳಿಸುವ ತಂತ್ರಗಳು (Techniques to enforce IS in an organization)

ಮಾಹಿತಿ ಭದ್ರತೆಯನ್ನು ಜಾರಿಗೊಳಿಸಲು ಸಾಧನಗಳನ್ನು ಗುರುತಿಸುವುದು (Identifying tools to enforce Information Security)

ಯಶಸ್ವಿ ಮಾಹಿತಿ ಭದ್ರತಾ ನೀತಿಯು ನಿಗಮಗಳಿಗೆ ಹಲವಾರು ಲಾಭಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಜಾರಿಗೊಳಿಸಬಹುದಾದ ನೀತಿಗಳು ದುರ್ಬಲತೆಗಳನ್ನು ಗುರುತಿಸುವುದನ್ನು ಮತ್ತು ಪರಿಹರಿಸುವುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತವೆ. ಇದು ವ್ಯಾಪಾರ ನಿರಂತೆತೆಯನ್ನು ರಕ್ಷಿಸುತ್ತದೆ ಮತ್ತು ಐಟಿ ಮೂಲಸೌಕರ್ಯವನ್ನು ಬಲಪಡಿಸುತ್ತದೆ. ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿರುವ ಉದ್ಯೋಗಿಗಳು ಸುರಕ್ಷತಾ ನಿಯಮಗಳನ್ನು ಪಾಲಿಸಿದಾಗ, ಮಾಹಿತಿಯನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ಸಂಸ್ಥೆಯಲ್ಲಿ ಅದರ ಜೊತೆಗೆ ಗ್ರಾಹಕರೊಂದಿಗೆ , ಪಾಲುದಾರರೊಂದಿಗೆ ಮತ್ತು ಮಾರಾಟಗಾರರೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳಲಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ ಮತ್ತು ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

1 **ಪರಿಣಾಮಕಾರಿಯಾದ ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ನೀತಿಯನ್ನು ರಚಿಸುವ ಮೊದಲನೇ ಹಂತವೆಂದರೆ, ಮಾಹಿತಿಯ ಸ್ವತ್ತುಗಳನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡುವುದು ಮತ್ತು ಈ ಸ್ವತ್ತುಗಳಿಗಿರುವ ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸುವುದು.**

ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿರುವ ಕೆಲವು ಆಸ್ತಿಗಳು ಇತರ ಆಸ್ತಿಗಳಿಗಿಂತ ಹೆಚ್ಚು ಮೌಲ್ಯಯುತವಾಗಿರುತ್ತವೆ. ಆದರೆ ವಿತ್ತೀಯ ಮೌಲ್ಯ ಮಾತ್ರ ಒಂದೇ ಅಂಶವಾಗಿರಬಾರದು. ಒಂದು ಆಸ್ತಿಯ ವಿತ್ತೀಯ ಮೌಲ್ಯ ಮತ್ತು ಆಂತರಿಕ ಮೌಲ್ಯ ಎರಡನ್ನೂ ನಿರ್ಧರಿಸುವುದು ಅದರ ಮೌಲ್ಯವನ್ನು ನಿಖರವಾಗಿ ಅಳೆಯಲು ಅತ್ಯಗತ್ಯವಾಗಿರುತ್ತವೆ. ಒಂದು ಆಸ್ತಿಯ ವಿತ್ತೀಯ ಮೌಲ್ಯವನ್ನು ಲೆಕ್ಕ ಮಾಡಲು, ಸಂಸ್ಥೆಯು ಆ ಆಸ್ತಿಯ ಡಾಟಾ ,ನೆಟ್‌ವರ್ಕ್‌ಗಳು ಅಥವಾ ಸಿಸ್ಟಮ್‌ಗಳು ಯಾವುದೇ ರೀತಿಯಲ್ಲಿ ರಾಜಿಯಾಗಿವೆಯೆ ಎಂಬುದರ ಪರಿಣಾಮವನ್ನು ಪರಿಗಣಿಸಬೇಕು. ಆಂತರಿಕ ಮೌಲ್ಯವನ್ನು ಲೆಕ್ಕ ಮಾಡಲು, ಸಂಸ್ಥೆಯು ಮುಖ್ಯ ಮಧ್ಯಸ್ಥಗಾರರೊಂದಿಗೆ ವಿಶ್ವಾಸಾರ್ಹತೆ, ಖ್ಯಾತಿ ಮತ್ತು ಸಂಬಂಧಗಳ ಸುರಕ್ಷತಾ ಘಟನೆಯ ಪರಿಣಾಮವನ್ನು ಪರಿಗಣಿಸಬೇಕು.

2 **ಸಂಸ್ಥೆಗಳು ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನವನ್ನು ನಿರ್ವಹಿಸುವುದಕ್ಕಾಗಿ ಒಂದು ನೀತಿಯನ್ನು ರಚಿಸುವುದು.**

ಆಸ್ತಿಗಳನ್ನು ಮತ್ತು ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸಿದ ನಂತರ, ಮುಂದಿನ ಹಂತವು ಸಂಸ್ಥೆಗಳು ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನವನ್ನು ನಿರ್ವಹಿಸಲು ಒಂದು ನೀತಿಯನ್ನು ರಚಿಸುವುದು. ಈ ಮೌಲ್ಯಮಾಪನವು ಮಾಹಿತಿ ಸುರಕ್ಷಿತವಾಗಿದೆಯೆ, ಅತಿ ರಕ್ಷಿತವಾಗಿದೆಯೆ ಅಥವಾ ಸಮರ್ಪಕವಾಗಿ ರಕ್ಷಿಸಲಾಗಿದೆಯೆ ಎಂದು ನಿರ್ಧರಿಸಲು ಸಂಸ್ಥೆಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಈ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದ ಗುರಿಯು, ಸಂಸ್ಥೆಯ ಅನವಶ್ಯಕ ಅಪಾಯಗಳನ್ನು ಬಹಿರಂಗಪಡಿಸದೇ ಖರ್ಚುಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುವುದು ಆಗಿದೆ. ಈ ಮೌಲ್ಯಮಾಪನವು ಸಂಪನ್ಮೂಲಗಳ ಸರಿಯಾದ ಹಂಚಿಕೆಯನ್ನು ನಿರ್ಧರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟು (The Information Security Framework)

ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟು ಒಂದು ಸಂಸ್ಥೆಗೆ ಸುರಕ್ಷತೆಯ ನೀತಿಗಳನ್ನು ಮತ್ತು ಅಭ್ಯಾಸಗಳನ್ನು ಸ್ಥಾಪಿಸುತ್ತದೆ. ನೀತಿಗಳು ಸಂಸ್ಥೆಯ ಸುರಕ್ಷತೆಯ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ವಿಷಯಗಳ ಬಗ್ಗೆ ಸದಸ್ಯರು ಅನುಸರಿಸಬೇಕಾದ ಮಾರ್ಗದರ್ಶನ ಒದಗಿಸುತ್ತವೆ. ಸುರಕ್ಷತಾ ನೀತಿಗಳು ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿನ ಎಲ್ಲ ಹಾರ್ಡ್‌ವೇರ್, ಸಾಫ್ಟ್‌ವೇರ್ , ಡಾಟಾ, ಮಾಹಿತಿ, ನೆಟ್‌ವರ್ಕ್, ವೈಯಕ್ತಿಕ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಾಧನಗಳು, ಬೆಂಬಲ ಸಿಬ್ಬಂದಿ, ಮತ್ತು ಬಳಕೆದಾರರಿಗೆ ಅನ್ವಯಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

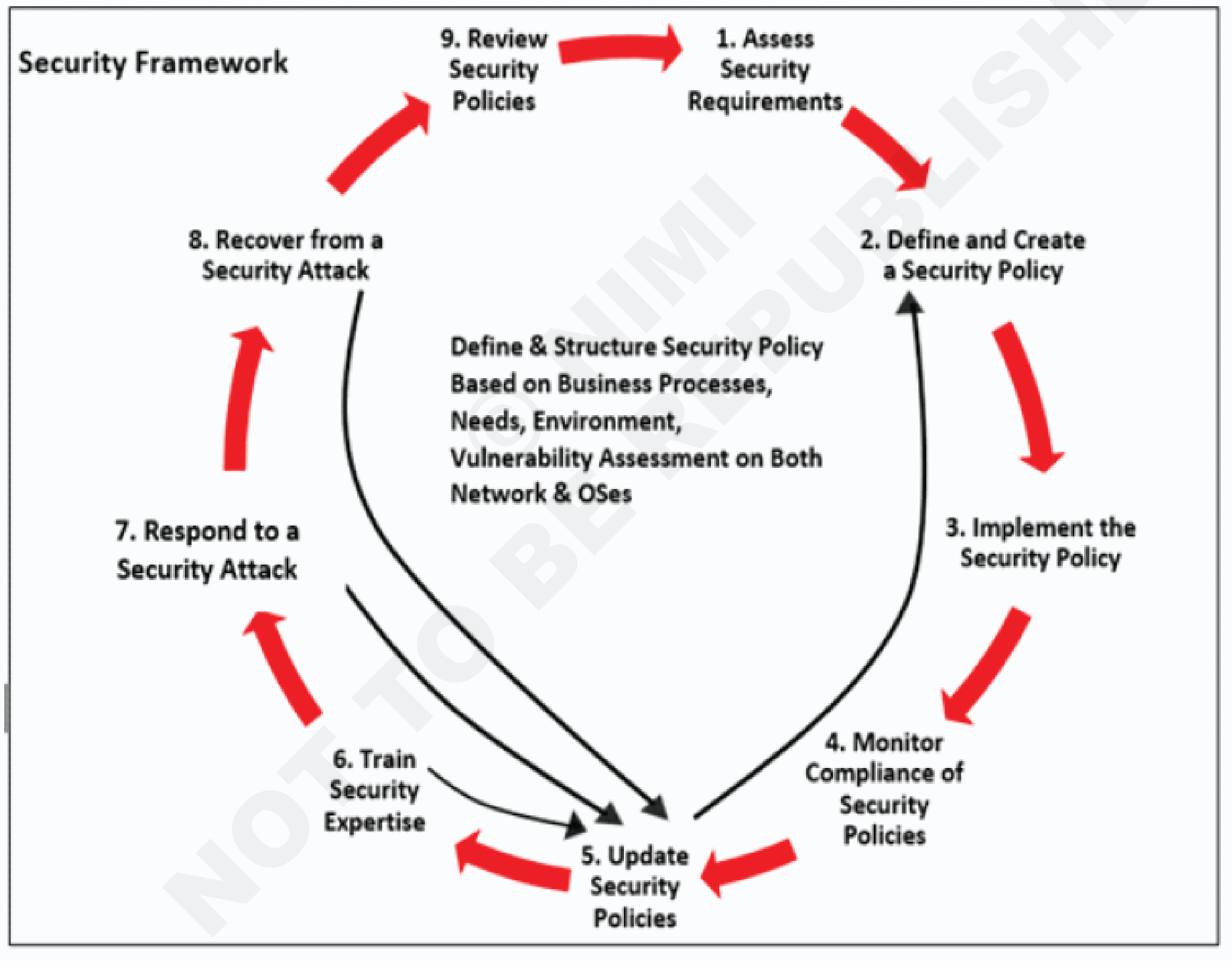
ಒಂದು IT ಸುರಕ್ಷತಾ ವ್ಯವಸ್ಥೆಯ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಉತ್ತಮವಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಲ್ಪಟ್ಟ ಭದ್ರತಾ ಚೌಕಟ್ಟನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಅವಶ್ಯಕತೆ ಇದೆ. ಇದು ಎಲ್ಲ ಮಧ್ಯಸ್ಥಗಾರರನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ ಮತ್ತು ಉಪಯುಕ್ತವಾಗಲು ಕಾಲಕಾಲಕ್ಕೆ ಅದನ್ನು ನವೀಕರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟು ನಿರ್ವಹಣಾ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಸುಗಮಗೊಳಿಸುತ್ತವೆ. ಇದು ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸಲು ಮತ್ತು ವ್ಯವಸ್ಥೆ/ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಆಸ್ತಿಗಳನ್ನು ಗುರುತಿಸುವ, ಸುರಕ್ಷತಾ ಅವಶ್ಯಕತೆಗಳನ್ನು ನಿರ್ಧರಿಸುವ, ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನ , ನಿಯಂತ್ರಣಾ ಅನುಷ್ಠಾನ, ಪ್ರಕ್ರಿಯೆ ಮೇಲ್ವಿಚಾರಣೆ ಮತ್ತು ನವೀಕರಣ ಮುಂತಾದವುಗಳನ್ನು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ಮಾಹಿತಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟಿನಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಬಳಸಲ್ಪಡುವ ಪದಗಳೆಂದರೆ :

- ನೀತಿ : ಸಾಮಾನ್ಯ ನಿರ್ವಹಣಾ ಹೇಳಿಕೆಗಳು
- ಮಾನದಂಡಗಳು : ನಿರ್ದಿಷ್ಟ ಕಡ್ಡಾಯ ನಿಯಂತ್ರಣಗಳು
- ಮಾರ್ಗಸೂಚಿಗಳು : ಶಿಫಾರಸುಗಳು/ ಉತ್ತಮ ಅಭ್ಯಾಸಗಳು
- ಕಾರ್ಯವಿಧಾನಗಳು : ಹಂತ ಹಂತವಾದ ಸೂಚನೆಗಳು

ವಾಸ್ತವವಾಗಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಲು ಇರುವ ವಿವರವಾದ ಚಟುವಟಿಕೆಗಳು, ಅಭ್ಯಾಸದ ಅನುಕ್ರಮವನ್ನು ಅನುಸರಿಸಬೇಕಾಗುತ್ತದೆ, ಸರಿಪಡಿಸುವ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳ ಬೇಕಾಗುತ್ತದೆ, ಇದನ್ನು ಕೆಳಗಿನ ಚಿತ್ರದಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.(Fig 2)

Fig 2



© NIMI NOT TO BE REPUBLISHED

ಒಂದು ಚೌಕಟ್ಟಿನಲ್ಲಿರುವ ಪ್ರತಿಯೊಂದು ಹೆಡ್‌ನ ಅಡಿಯಲ್ಲಿರುವ ಪ್ರಮುಖ ಹೆಡ್‌ಗಳು ಮತ್ತು ಅಭ್ಯಾಸಗಳನ್ನು ಈ ಕೆಳಗೆ ತೋರಿಸಲಾಗಿದೆ.

ಈ ರೀತಿಯಾಗಿ ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟು ಸುರಕ್ಷತಾ ನೀತಿಗಳನ್ನು ಆಚರಣೆಗಳಿಗೆ ತಿರುಗಿಸುವ ಒಂದು ತಂತ್ರಜ್ಞಾನವಾಗುತ್ತದೆ. ಅದು ನಾಲ್ಕು ಹಂತಗಳ ಚಕ್ರದ ಮೂಲಕ ಇದನ್ನು ಸಾಧಿಸುತ್ತದೆ. ಯೋಜನೆ (plan), ಕಾರ್ಯನಿರ್ವಹಿಸುವುದು (do), ಕಾರ್ಯ(act) ಮತ್ತು ಪರಿಶೀಲಿಸು(check). PPT triad ಎಂದರೆ, ಇದನ್ನು ಸಾಧಿಸಲು ಜನರು, ಪ್ರಕ್ರಿಯೆ ಮತ್ತು ತಂತ್ರಜ್ಞಾನ ಇವುಗಳಿಗೆ ಸಮಾನ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ನೀಡಬೇಕಾಗಿದೆ.

ಹೊಸ ತಂತ್ರಜ್ಞಾನಗಳು ಮತ್ತು ಹೊಸ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಸುರಕ್ಷತಾ ಚೌಕಟ್ಟಿನಲ್ಲಿ ಪ್ಲಗ್ ಇನ್ ಮಾಡಬಹುದು ಮತ್ತು ಸುರಕ್ಷತಾ ವೆಚ್ಚಗಳು ಹೆಚ್ಚು ಊಹಿಸಬಹುದಾಗಿದೆ ಮತ್ತು ನಿರ್ವಹಿಸಬಹುದಾಗಿದೆ.

ಗುರುತಿಸಿ	ರಕ್ಷಿಸಿ	ಪತ್ತೆಮಾಡಿ	ಪ್ರತಿಕ್ರಿಯಿಸಿ	ಮರಳಿಪಡೆಯಿಂ
• ಆಸ್ತಿ ನಿರ್ವಹಣ • ವ್ಯಾಪಾರದ ಪರಿಸಂ • ಆಡಳಿತ • ಅಪಾಯ • ಅಪಾಯ ನಿರ್ವಹಣಾ ತಂತ್ರಗಳು	• ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಇ ವೆಂಟ್‌ಗಳು • ಜಾಗೃತಿ ಮತ್ತು ತರಬೇತಿ • ಮಾಹಿತಿ ರಕ್ಷಣೆ ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳು • ನಿರ್ವಹಣಾ ಮೌಲ್ಯಮಾಪನ • ರಕ್ಷಣಾತ್ಮಕ ತಂತ್ರಜ್ಞಾನ	• ವೈಪರಿತ್ಯಗಳು ಮತ್ತು • ಸುರಕ್ಷತೆಯ ನಿರಂತರ ಮಾನಿಟರಿಂಗ್ • ಪತ್ತೆ ಪ್ರಕ್ರಿಯೆ	• ಪ್ರತಿಕ್ರಿಯೆ ಯೋಜನೆ • ಸಂವಹನಗಳು • ವಿಶ್ಲೇಷಣೆ • ಉಪಶಮನ • ಸುಧಾರಣೆಗಳು	• ಮರಳಿ ಪಡೆಯುವಿಕೆ ಯೋಜನೆ • ಸುಧಾರಣೆ • ಸಂವಹನಗಳು

ಭದ್ರತಾ ಬೆದರಿಕೆಗಳ ಅವಲೋಕನ (Overview of security threats)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಭದ್ರತಾ ಬೆದರಿಕೆಗಳು ಮತ್ತು ಅವುಗಳ ಪ್ರಕಾರಗಳನ್ನು ವಿವರಿಸಿ**
- **ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸುವ ವಿಧಾನಗಳನ್ನು ವಿವರಿಸಿ**
- **ಬೆದರಿಕೆಗಳು ಒಂದು ವ್ಯವಸ್ಥೆಯ ಮೇಲೆ ಹೇಗೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ ಎಂಬುದನ್ನು ವಿವರಿಸಿ**
- **ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸಲು ಮತ್ತು ತಗ್ಗಿಸಲು ಉತ್ತಮ ಅಭ್ಯಾಸಗಳನ್ನು ವಿವರಿಸಿ.**

ಪರಿಚಯ (Introduction)

ಬೆದರಿಕೆ ಎಂದರೆ ಒಂದು ವ್ಯವಸ್ಥೆಯನ್ನು ಅಥವಾ ಚಟುವಟಿಕೆಯನ್ನು ಹಾನಿಗೊಳಿಸುವ ಸಂಭಾವ್ಯತೆಯನ್ನು ಹೊಂದಿದ ಸಂದರ್ಭ ಅಥವಾ ಘಟನೆಯಾಗಿದೆ. ಇವು ನಾಶಪಡಿಸುವಿಕೆ, ಬಹಿರಂಗಪಡಿಸುವಿಕೆ ಮತ್ತು ಡಾಟಾವನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುವಿಕೆ ಅಥವಾ ಡೆನಿಯಲ್ ಆಫ್ ಸರ್ವೀಸ್ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿವೆ. ಬೆದರಿಕೆಯು ಹಾನಿಯ ಸಂಭಾವ್ಯತೆಯಾಗಿದೆ. ಬೆದರಿಕೆಯ ಉಪಸ್ಥಿತಿ ಇದೆ ಎಂದರೆ ಅದು ವಾಸ್ತವವಾಗಿ ಹಾನಿಯನ್ನುಂಟು ಮಾಡುತ್ತದೆ ಎಂದು ಅರ್ಥವಲ್ಲ.

ಬೆದರಿಕೆಗಳು ಮತ್ತು ಅವುಗಳ ವಿವರಣೆಗೆ ಸಂಬಂಧಿಸಿದ ಕೆಲವು ಸಾಮಾನ್ಯ ಪದಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

1 **ಅನಧಿಕೃತ ಪ್ರವೇಶ** (Unauthorized Access)

ಮಾಹಿತಿಯನ್ನು ಅಥವಾ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಅನುಮತಿ ಇಲ್ಲದೇ ಅಥವಾ ಹಕ್ಕುಗಳಿಲ್ಲದೇ ಅದನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಪ್ರಯತ್ನಿಸುವುದು ಅಥವಾ ಯಶಸ್ವಿಯಾಗಿ ಪಡೆದುಕೊಳ್ಳುವುದು

2 **ಸೈಬರ್ ಬೇಹುಗಾರಿಕೆ** (Cyber Espionage)

ಕಂಪ್ಯೂಟರ್‌ನ ಬಳಕೆಯಿಂದ ಕಂಪನಿಯನ್ನು ರಹಸ್ಯವಾಗಿ ಪ್ರವೇಶಿಸುವುದು ಅಥವಾ ಕಂಪನಿ ಅಥವಾ ಸರ್ಕಾರಿ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ಹ್ಯಾಕ್ ಮಾಡಿ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು

3 **ಮಾಲ್ವೇರ್** (Malware)

ಇದು ದುರುದ್ದೇಶಪೂರಿತ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಾಗಿ ಬಳಸುವ ಒಂದು ಸಾಮೂಹಿಕ ಪದವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ವೈರಸ್‌ಗಳು, ವರ್ಮ್‌ಗಳು, ಮತ್ತು ಟ್ರೋಜನ್‌ಗಳು; ಈ ಮಾಲ್ವೇರ್‌ಗಳನ್ನು ಅಪರಾಧ, ವಾಣಿಜ್ಯ ಅಥವಾ ವಿನಾಶಕಾರಿ ಉದ್ದೇಶಗಳಿಗಾಗಿ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ, ಮತ್ತು ಮಾಹಿತಿಯಲ್ಲಿ ಒಳನುಸುಳಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿರುತ್ತದೆ.

4 **ಡಾಟಾ ಸೋರಿಕೆ** (Data Leakage)

ಕಂಪನಿಯ ಅಥವಾ ವೈಯಕ್ತಿಕ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯ, ಉದ್ದೇಶಪೂರ್ವಕ ಅಥವಾ ಆಕಸ್ಮಿಕ ನಷ್ಟ, ಕಳುವು ಅಥವಾ ಹೊರತೋರಿಸುವುದು.

© NIMI NOT TO BE REPUBLISHED

5 **ಮೊಬೈಲ್ ಸಾಧನದ ಮೇಲೆ ದಾಳಿ** (Mobile Device Attack)

ಮೊಬೈಲ್ ಸಾಧನಗಳ ಮೇಲೆ ಮತ್ತು ಅದರಲ್ಲಿ ಸಂಗ್ರಹಿಸಿರುವ ಮತ್ತುಸಂಸ್ಕರಿಸಿದ ಮಾಹಿತಿಯ ಮೇಲೆ ದುರುದ್ದೇಶಪೂರಿತ ದಾಳಿ ಅಥವಾ ಅನಧಿಕೃತ ಪ್ರವೇಶ. ಈ ದಾಳಿಯನ್ನು ವಯರ್ ಇಲ್ಲದೇ ಅಥವಾ ಭೌತಿಕ ಸ್ವಾಧೀನದ ಮೂಲಕ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ.

6 **ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್** (Social Engineering)

ಫೋನ್, ಇ ಮೇಲ್, ಆನ್‌ಲೈನ್ ಅಥವಾ ವೈಯಕ್ತಿಕವಾಗಿ ಇತರರನ್ನು ಮೋಸಗೊಳಿಸುವ ಮೂಲಕ, ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸುವುದು. ಇದು ಕಂಪನಿಯ ಅಥವಾ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿನ ಮಾಹಿತಿಯನ್ನು ಪ್ರವೇಶಿಸುತ್ತದೆ.

7 **ಒಳಗಿನವರು** (Insiders)

ಕದಿಯುವ ದುರುದ್ದೇಶವನ್ನು ಹೊಂದಿದ ಉದ್ಯೋಗಿ ಅಥವಾ ಕೆಲಸಗಾರರು ಕಂಪನಿಯ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಕದಿಯುತ್ತಾನೆ, ವಂಚಿಸುತ್ತಾನೆ ಅಥವಾ ಕಂಪನಿಯ ವ್ಯವಸ್ಥೆಗೆ ಅಥವಾ ಮಾಹಿತಿಗೆ ಹಾನಿಯಂಟು ಮಾಡುತ್ತಾರೆ.

8 **ಫಿಶಿಂಗ್** (Phishing)

ಇದು ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್‌ನ ಒಂದು ರೂಪವಾಗಿದೆ. ಇದು ಕಾನೂನುಬದ್ಧವಾಗಿ ಇರುವಂತೆ ಕಾಣುವ ಇ ಮೇಲ್‌ಗಳನ್ನು ಕಳುಹಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಸಾಮಾನ್ಯವಾಗಿ ಸ್ವೀಕರಿಸುವವರಿಂದ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಹೊರತೆಗೆಯುವ ಮೋಸದ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಪ್ರವೇಶಿಸುತ್ತದೆ ಅಥವಾ ಗುರುತಿನ ಕಳ್ಳತನವನ್ನು ಮಾಡುತ್ತದೆ.

9 **ಸಿಸ್ಟಮ್ ರಾಜಿ** (System Compromise)

ದುರುದ್ದೇಶಪೂರಿತ ವ್ಯಕ್ತಿಗಳು ಅಥವಾ ಹ್ಯಾಕರ್‌ಗಳು ದಾಳಿಮಾಡಿದ ಮತ್ತು ಸ್ವಾಧೀನಪಡಿಸಿಕೊಂಡ ಸಿಸ್ಟಮ್, ಸಾಮಾನ್ಯವಾಗಿ ನಂತರ ಆಗಾಗ್ಗೆ ಇತರ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ದಾಳಿ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ

10 **ಸ್ಪ್ಯಾಮ್** (Spam)

ಅಪೇಕ್ಷಿಸದ ಇ ಮೇಲ್‌ನ್ನು ದೊಡ್ಡ ಪ್ರಮಾಣದಲ್ಲಿ ಅನೇಕ ವ್ಯಕ್ತಿಗಳಿಗೆ ಕಳುಹಿಸಲಾಗುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ವಾಣಿಜ್ಯ ಲಾಭಕ್ಕಾಗಿ, ಆದರೆ ಹೆಚ್ಚಾಗಿ ಮಾಲ್ವೇರನ್ನು ಹರಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

11 **ಡೇನಿಯಲ ಆಫ್ ಸರ್ವೀಸ್** (Denial of Service)

ಇದು ಒಂದು ವ್ಯವಸ್ಥೆಯ ಮೇಲೆ ಮತ್ತು ಅದರಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ಮಾಹಿತಿಯ ಮೇಲೆ ಉದ್ದೇಶಪೂರ್ವಕ ಅಥವಾ ಉದ್ದೇಶಪೂರ್ವಕವಲ್ಲದ ದಾಳಿಯಾಗಿದೆ. ಇದರಿಂದ ಅಧಿಕೃತ ಬಳಕೆದಾರರಿಗೆ ಸಿಸ್ಟಮ್‌ನ್ನು ಲಭ್ಯವಿಲ್ಲದಂತೆ ಮಾಡುತ್ತದೆ ಮತ್ತು ಪ್ರವೇಶಿಸಲಾಗುವುದಿಲ್ಲ.

12 **ಗುರುತಿನ ಕಳ್ಳತನ** (Identity Theft)

ಇದು ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ಕಳ್ಳತನ ಮಾಡುವುದಾಗಿದೆ. ಇದನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಆರ್ಥಿಕ ಕಳ್ಳತನ ಮಾಡಲು ಮತ್ತು ಅಪರಾಧಗಳನ್ನು ಎಸಗಲು ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ಗುರುತನ್ನು ಕಳುವು ಮಾಡುವುದಾಗಿದೆ.

ಬೆದರಿಕೆಗಳ ವರ್ಗಗಳು (Categories of threats)

ಭದ್ರತಾ ಬೆದರಿಕೆಗಳನ್ನು ಅನೇಕ ವಿಧಗಳಲ್ಲಿ ವರ್ಗೀಕರಿಸಬಹುದು. ಕೆಲವು ಜನಪ್ರೀಯ ವರ್ಗೀಕರಣಗಳು ಕೆಳಗಿನಂತಿವೆ:

1 ಅತ್ಯಾಧುನಿಕತೆಯ ಆಧಾರದ ಮೇಲೆ, ಭದ್ರತಾ ಬೆದರಿಕೆಗಳನ್ನು ಮೂರು ವಿಧದಲ್ಲಿ ವರ್ಗೀಕರಿಸಲಾಗಿದೆ.

- ಮೊದಲನೇಯ ಪೀಳಿಗೆಯ ಬೆದರಿಕೆಗಳು ಜೆನರಿಕ ವೈರಸ್ - ರೀತಿಯ ದಾಳಿಗಳಾಗಿವ. ಇವು ಬಳಕೆದಾರರಿಂದ ಸೋಂಕಿತ ಇಮೇಲ್‌ಗಳನ್ನು ಮತ್ತು ಅಪ್ರಜ್ಞಾಪೂರ್ವಕ ಫೈಲ್ ಲಗತ್ತುಗಳನ್ನು ತೆರೆಯುವುದರ ಮೂಲಕ ಹರಡುತ್ತವೆ.
- ಎರಡನೇ ತಲೆಮಾರಿನ ಬೆದರಿಕೆಗಳು ಹೆಚ್ಚು ಅತ್ಯಾಧುನಿಕವಾಗಿದೆ ಮತ್ತು ದೊಡ್ಡ ಸಮಸ್ಯೆಗಳನ್ನು ಉಂಟು ಮಾಡುತ್ತವೆ. ಇವುಗಳನ್ನು ಸ್ವಯಂಚಾಲಿತ ಉಪಕರಣಗಳೊಂದಿಗೆ ರಚಿಸಲಾಗಿದೆ. ಈ ವರ್ಮ್‌ಗಳು ಮಾನವನ ಪ್ರಭಾವವಿಲ್ಲದೇ ದೋಷಗಳನ್ನು ದಾಳಿ ಮಾಡುತ್ತವೆ. ಪುನರಾವರ್ತಿಸುವುದು, ಗುರುತಿಸುವುದು, ಮತ್ತು ಹೊಸ ಬಲಿಪಶುಗಳನ್ನು ಗುರಿಪಡಿಸುವುದು ಸ್ವಯಂಚಾಲಿತವಾಗಿರುತ್ತದೆ.
- ಮೂರನೇಯಪೀಳಿಗೆಯಬೆದರಿಕೆಗಳುಮಿಶ್ರಿತಬೆದರಿಕೆಗಳಾಗಿವೆ. ಇವು ಸಾಮಾನ್ಯವಾಗಿ ಸಂಯೋಜಿಸುವ ವೈರಸ್‌ಗಳಾಗಿವೆ, ಟ್ರೋಜನ್‌ಗಳಾಗಿವೆ ಮತ್ತು ಯಾಂತ್ರೀಕೃತವಾಗಿವೆ. ಈ ವರ್ಮ್‌ಗಳು ಹೈಪರ್-ಪ್ರಸರಣಕ್ಕಾಗಿ, ಗುರಿಗಳನ್ನು ಮೊದಲೇ ಕಂಪೈಲ್ ಮಾಡುತ್ತವೆ, ತಿಳಿದಿರುವ ದುರ್ಬಲತೆಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತವೆ, ಗುಪ್ತ ದೋಷಗಳನ್ನು ಉದ್ದೇಶಿತ ಬಳಕೆಯನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತವೆ. ಇವು ವೈಯರ್‌ಲೆಸ್ ಲಿಂಕ್‌ಗಳನ್ನು, ವರ್ಚುವಲ್ ಖಾಸಗಿ ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು, ಗುರಿಯಾಗಿಸುತ್ತವೆ. ಮತ್ತು ಫೈರ್‌ವಾಲ್ ಮತ್ತು ಒಳನುಗ್ಗುವಿಕೆ ಪತ್ತೆ ವ್ಯವಸ್ಥೆಗಳಂತಹ ಪರಿಧಿಯ ಒಳಗೆ ದಾಳಿಗಳನ್ನು ಮಾಡುತ್ತದೆ.

2 OWASP (Open Web Applications Security Project) ಪ್ರಕಾರ ಪ್ರಮುಖ ಬೆದರಿಕೆಗಳು ಕೆಳಗಿನಂತಿವೆ.

- Injection

© NIMI NOT TO BE REPUBLISHED

- Cross Site Scripting (CSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross - Site Request Forgery (CSRF)
- Security Misconfiguration
- Insecure Cryptographic Storage
- Failure to Restrict URL Access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

3 ಮೈಕ್ರೊಸಾಫ್ಟ್‌ನ ಮೂಲಕ ಬಳಸಲ್ಪಡುವ ಶೋಷಣೆಗಳ (ಆಕ್ರಮಣಕಾರರ ಪ್ರೇರಣೆ) ವಿಧಗಳ ಪ್ರಕಾರ ಎಂದರೆ STRIDE ವ್ಯವಸ್ಥೆ. STRIDE ನ ಸಂಕ್ಷಿಪ್ತ ರೂಪವನ್ನು ಕೆಳಗಿನ ಪ್ರತಿಯೊಂದು ವರ್ಗದ ಮೊದಲ ಅಕ್ಷರದಿಂದ ರಚಿಸಲಾಗಿದೆ.

- ಗುರುತನ್ನು ವಂಚಿಸುವುದು . ಗುರುತಿನ ವಂಚನೆಯ ಉದಾಹರಣೆಯೆಂದರೆ, ಯೂಸರ್‌ನೇಮ್ ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ನಂತಹ ಇನ್ನೊಬ್ಬ ಬಳಕೆದಾರರ ದೃಢೀಕರಣದ ಮಾಹಿತಿಯನ್ನು ಕಾನೂನುಬಾಹಿರವಾಗಿ ಪಡೆದುಕೊಳ್ಳುವುದು ಮತ್ತು ಉಪಯೋಗಿಸುವುದು.
- ಡಾಟಾವನ್ನು ಟ್ಯಾಂಪರ್ ಮಾಡುವುದು. ಡಾಟಾವನ್ನು ದುರುದ್ದೇಶಪೂರಿತವಾಗಿ ಮಾರ್ಪಾಡು ಮಾಡುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.
- ನಿರಾಕರಣೆ. ನಿರಾಕರಣೆಯ ಬೆದರಿಕೆಗಳು ಯಾವ ಬಳಕೆದಾರರು ಇತರ ಪಕ್ಷಗಳಿಲ್ಲದೇ ಕೆಲಸ ಮಾಡುವುದನ್ನು ನಿರಾಕರಿಸುತ್ತಾರೊ ಅಂತಹವರಿಗೆ ಸಂಬಂಧಿಸಿದ್ದಾಗಿದೆ. ಉದಾಹರಣೆಗೆ ಒಬ್ಬ ಬಳಕೆದಾರನು ನಿಷೇಧಿತ ಕಾರ್ಯಾಚರಣೆಗಳನ್ನು ಪತ್ತೆಹಚ್ಚುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರದ ಒಂದು ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಅಕ್ರಮ ಕೆಲಸವನ್ನು ಮಾಡುತ್ತಾನೆ.
- ಮಾಹಿತಿ ಬಹಿರಂಗಪಡಿಸುವಿಕೆ. ಮಾಹಿತಿ ಬಹಿರಂಗಪಡಿಸುವಿಕೆ ಬೆದರಿಕೆಗಳು ಪ್ರವೇಶವನ್ನು ಹೊಂದಿರದ ವ್ಯಕ್ತಿಗಳಿಗೆ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.
- ಡೇನಿಯಲ್ ಆಫ್ ಸರ್ವಿಸ್. Denial of service (DoS) ದಾಳಿಗಳು ಮಾನ್ಯ ಬಳಕೆದಾರರಿಗೆ ಸೇವೆಯನ್ನು ನಿರಾಕರಿಸುತ್ತವೆ.
- ಸವಲತ್ತುಗಳ ಉನ್ನತಿ. ಈ ರೀತಿಯ ಬೆದರಿಕೆಯಲ್ಲಿ, ಅಮಾನ್ಯ ಬಳಕೆದಾರರು ಸವಲತ್ತುಗಳ ಪ್ರವೇಶವನ್ನು ಪಡೆಯುತ್ತಾರೆ ಮತ್ತು ತನ್ಮೂಲಕ ಇಡಿ ವ್ಯವಸ್ಥೆಯನ್ನು ನಾಶ ಮಾಡುತ್ತದೆ.

ತಂತ್ರಜ್ಞಾನ ಆಧಾರಿತ ಬೆದರಿಕೆಗಳು (Threats based on technology)

1 WWW ತಂತ್ರಜ್ಞಾನ ಆಧಾರಿತ ಬೆದರಿಕೆಗಳು

2 ಬ್ರೌಸರ್ ಸಾಫ್ಟವೇರ್‌ನಲ್ಲಿ ಹೊಸ ವೈಶಿಷ್ಟ್ಯಗಳು

3 ಬ್ರೌಸರ್ ಸಾಫ್ಟವೇರ್‌ನ ಪರೀಕ್ಷಾ ಆವೃತ್ತಿಗಳು

4 ಸರ್ವರ್ ಸಾಫ್ಟ್‌ವೇರ್

5 CGI ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳು

6 Cookies

7 Unix ಮತ್ತು TCP/IP ಸಾಧನಗಳ ಆಧಾರಿತ ಬೆದರಿಕೆಗಳು

8 firewall ನಿರ್ವಹಣೆಯಲ್ಲಿನ ತೊಂದರೆಗಳು

9 ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಕ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಬಳಕೆ

10 Hacker ಸಾಧನಗಳು

11 ಇತರ ಸಾಫ್ಟ್‌ವೇರ್ ಆಧಾರಿತ ಬೆದರಿಕೆಗಳು

12 ಇಂಟ್ರಾನೆಟ್ ಅಪ್ಲಿಕೇಷನ್ ಸಾಫ್ಟ್‌ವೇರ್

13 Java ಭಾಷೆ

14 ActiveX

15 ಸಂವಹನಗಳ ಆಧಾರದ ಮೇಲಿನ ಬೆದರಿಕೆಗಳು

16 ವೈರಸ್‌ಗಳನ್ನು ಆಧರಿಸಿದ ಬೆದರಿಕೆಗಳು

17 ಮಾನವನ ಚಟುವಟಿಕೆಗಳ ಆಧಾರದ ಮೇಲಿನ ಬೆದರಿಕೆಗಳು

ಮಾಹಿತಿ ಭದ್ರತಾ ಬೆದರಿಕೆಗಳ ಗುರುತಿಸುವಿಕೆ (Identification of Information Security Threats)

ಮಾಹಿತಿ ಭದ್ರತಾ ನಿರ್ವಹಣೆಯ ಕಾರ್ಯಕ್ರಮದ ಯಶಸ್ಸು ಒಂದು ಸಂಸ್ಥೆಯ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಿಗೆ ಇರುವ ಬೆದರಿಕೆಗಳನ್ನು ನಿಖರವಾಗಿ ಗುರುತಿಸುವಿಕೆಯನ್ನು ಆಧರಿಸಿದೆ. ಮಾಹಿತಿ ಭದ್ರತಾ ಬೆದರಿಕೆಗಳು, ಭದ್ರತಾ ಯೋಜಕರಿಗೆ ಅವಶ್ಯಕವಾದ ಮೊದಲನೇಯ ಹೆಜ್ಜೆಯಾಗಿದೆ. ಸರಿಯಾದ ಬೆದರಿಕೆ ಮತ್ತು ದುರ್ಬಲತೆಯ ಗುರುತಿಸುವಿಕೆಯು ಭದ್ರತಾ ಪರೀಕ್ಷೆಗಳನ್ನು ಮತ್ತು ತಪಾಸಣೆಗಳನ್ನು ಒಳಗೊಂಡಿರಬೇಕು. ಇವು ಉಪಕರಣಗಳು ಸರಿಯಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತಿವೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸುತ್ತವೆ, ಅಗತ್ಯವಿದ್ದಾಗ ಸುಲಭವಾಗಿ ಲಭ್ಯವಿರುತ್ತದೆ ಮತ್ತು ಉದ್ಯೋಗಿಗಳು ಸಲಕರಣೆಯ ಬಳಕೆಯಲ್ಲಿ ಪ್ರವೀಣರಾಗಿದ್ದಾರೆ ಎಂದು ತಿಳಿಸುತ್ತದೆ.

ಇದನ್ನು ಸಾಧಿಸಲು ಸಿಸ್ಟಮ್‌ಗಳು ಪರೀಕ್ಷಾ ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಬೇಕು. ಈ ಪ್ರೊಗ್ರಾಂ, ಭದ್ರತೆಯ ಪ್ರಸ್ತುತ ಸ್ಥಿತಿಯನ್ನು ನಿರ್ಣಯಿಸುವುದು ಮಾತ್ರವಲ್ಲದೇ, ತರಬೇತಿಯ

© NIMI NOT TO BE REPUBLISHED

ಮೂಲಕ ಸಿಬ್ಬಂದಿಗಳ ಪರಿಣಾಮಕಾರಿತ್ವವನ್ನು ನವೀಕರಿಸಲು ಸಹ ಬಳಸಬಹುದು.

ಭದ್ರತಾ ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸುವ ಎರಡು ಪ್ರಮುಖ ವಿಧಾನಗಳು ಯಾವವೆಂದರೆ- ಪ್ರೊಬಿಂಗ್ (Probing) ಮತ್ತು ಸ್ಕ್ಯಾನಿಂಗ್ (Scanning).

ಪ್ರೋಬಿಂಗ್ ಎಂದರೆ ಕಂಪ್ಯೂಟರ್ ಮತ್ತು ಅದರ ಫೈಲುಗಳಿಗೆ ಪ್ರವೇಶವನ್ನು ಪಡೆಯುವ ಪ್ರಯತ್ನವಾಗಿದೆ. ಇದನ್ನು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ ತಿಳಿದಿರುವ ಅಥವಾ ಸಂಭವನೀಯ ದುರ್ಬಲ ಪಾಯಿಂಟ್ ಮೂಲಕ ಮಾಡಲಾಗುತ್ತದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್‌ನ ಸ್ಥಿತಿಯ ಬಗ್ಗೆ ಏನಾನ್ನಾದರೂ ಕಲಿಯುವ ಉದ್ದೇಶದಿಂದ ತೆಗೆದುಕೊಳ್ಳುವ ಕ್ರಮವಾಗಿದೆ.

ಸ್ಕ್ಯಾನಿಂಗ್ ಎಂದರೆ, ಎಲ್ಲ ಫೈಲುಗಳು ಅಥವಾ ನೆಟ್‌ವರ್ಕ್ ಅಂಶಗಳ ಮೂಲಕ ಅಸಾಮಾನ್ಯವಾದುದನ್ನು ಪತ್ತೆ ಹಚ್ಚುವ ಉದ್ದೇಶದ ಒಂದು ವಿಧಾನವಾಗಿದೆ. ಫೈಲ್ ಸ್ಕ್ಯಾನಿಂಗ್, ಬಳಕೆದಾರರು ಡೌನ್‌ಲೋಡ್ ಮಾಡಲು ಅಥವಾ ತೆರೆಯಲು ಪ್ರಯತ್ನಿಸುವ ಫೈಲ್‌ಗಳನ್ನು ವೈರಸ್‌ಗಳಿಗಾಗಿ ಮತ್ತು ಇತರ ದುರುದ್ದೇಶಪೂರಿತ ವಿಷಯಗಳಿಗಾಗಿ ಪರಿಶೀಲಿಸುತ್ತದೆ. ಫೈಲ್ ಸ್ಕ್ಯಾನಿಂಗ್ ನೀತಿ ಜಾರಿಗಾಗಿ ಕೆಲವು ಮಾಹಿತಿಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಫೈಲ್ ಸ್ಕ್ಯಾನಿಂಗ್‌ನಲ್ಲಿ ಎರಡು ವಿಧಗಳಿವೆ. ಅವುಗಳನ್ನು ಒಟ್ಟಿಗೆ ಬಳಸಬಹುದು.

- ಸುಧಾರಿತ ಪತ್ತೆಹಚ್ಚುವಿಕೆಯು, ತಿಳಿದಿರುವ ಮತ್ತು ಉದಯೋನ್ಮುಖ ಬೆದರಿಕೆಗಳು, ವೈರಸ್‌ಗಳನ್ನು ಸೇರಿದಂತೆ , ಟ್ರೋಜನ್ ಹಾರ್ಸ್‌ಗಳು, ವರ್ಮ್‌ಗಳು ಮುಂತಾದವುಗಳನ್ನು ಅನ್ವೇಷಿಸಲು ತಂತ್ರಗಳನ್ನು ಅನ್ವಯಿಸುತ್ತದೆ.
- ಆ್ಯಂಟಿ ವೈರಸ್ ಸ್ಕ್ಯಾನಿಂಗ್, ವೈರಸ್ ಸೋಂಕಿತ ಫೈಲ್‌ಗಳನ್ನು ಗುರುತಿಸಲು ಆ್ಯಂಟಿ ವೈರಸ್ ಡೆಫಿನಿಷನ್ ಫೈಲ್‌ಗಳನ್ನು ಬಳಸುತ್ತದೆ.

ನೆಟ್‌ವರ್ಕ್ ಸ್ಕ್ಯಾನಿಂಗ್ ಎಂದರೆ, ನೆಟ್‌ವರ್ಕ್ ಭದ್ರತಾ ಮೌಲ್ಯಮಾಪನದ ಉದ್ದೇಶಕ್ಕಾಗಿ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಸಕ್ರೀಯ ಹೋಸ್ಟ್‌ಗಳನ್ನು ಗುರುತಿಸುವ ಒಂದು ವಿಧಾನವಾಗಿದೆ. ಪಿಂಗ್ ಸ್ವೀಪ್ ಮತ್ತು ಪೋರ್ಟ್ ಸ್ಕ್ಯಾನ್‌ನಂತಹ ಸ್ಕ್ಯಾನಿಂಗ ಕಾರ್ಯವಿಧಾನಗಳು, ಐಪಿ ಅಡ್ರೆಸ್‌ಗಳು ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಸಕ್ರೀಯವಾಗಿರುವ ಲೈವ್ ಹೋಸ್ಟ್‌ಗಳನ್ನು ಮ್ಯಾಪ್ ಮಾಡುತ್ತವೆ ಮತ್ತು ಯಾವುದೇ ಅನುಮಾನಾಸ್ಪದ ಚಟುವಟಿಕೆಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತವೆ.

ವ್ಯವಸ್ಥೆಯ ಮೇಲೆ ಆಕ್ರಮಣ ಮಾಡುವ ಬೆದರಿಕೆಗಳ ಕಾರ್ಯಾಚರಣೆಯ ವಿಧಾನ (Modus Operandi of Threats in attacking a system)

ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಸಾಧನದ ಮೇಲಿನ ದಾಳಿಯು 3 ಹಂತಗಳಲ್ಲಿ ನಡೆಯುತ್ತದೆ :

- ಹೋಸ್ಟ್‌ನ ಸೋಂಕು
- ಅದರ ಗುರಿಯ ಸಾಧನೆ, ಮತ್ತು
- ಇತರ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಮಾಲ್ವೇರ್‌ನ ಹರಡುವಿಕೆ

ಬೆದರಿಕೆಗಳು ಹೆಚ್ಚಾಗಿ ಸೋಂಕಿತ ಸಾಧನಗಳು ನೀಡುವ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಬಳಸುತ್ತವೆ. ಅವು ಔಟ್‌ಪುಟ್ ಸಾಧನಗಳಾದ ಬ್ಲೂಟೂಥ್ ಅಥವಾ ಇನ್‌ಫ್ರಾರೆಡ್‌ಗಳನ್ನು ಬಳಸುತ್ತವೆ, ಆದರೆ ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ಅಡ್ರೆಸ್ ಬುಕ್ ಅಥವಾ ಇ ಮೇಲ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನೂ ಸಹ ಬಳಸಬಹುದು. ಮತ್ತು ಇವು ಬಳಕೆದಾರರ ಪರಿಚಯಸ್ಥರಿಗೆ ಸೋಂಕು ತಗುಲಿಸುತ್ತವೆ. ಪರಿಚಯಸ್ಥರು ಕಳುಹಿಸಿದ ಡಾಟಾದ ಮೇಲಿನ ನಂಬಿಕೆಯನ್ನು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ.

ಸೋಂಕು (Infection)

ಬೆದರಿಕೆಗಳು ಒಂದು ಸಾಧನವನ್ನು ಪ್ರವೇಶಿಸಲು ಬಳಸುವ ಸಾಧನವಾಗಿದೆ. ಇದು ಈ ಹಿಂದೆ ಪ್ರಸ್ತುತಪಡಿಸಲಾದ ದೋಷಗಳಲ್ಲಿ ಒಂದನ್ನು ಬಳಸಬಹುದು ಅಥವಾ ಬಳಕೆದಾರರ ದಡ್ಡತನವನ್ನು ಬಳಸಬಹುದು. ಸೊಂಕುಗಳು ಅನುಮತಿಯನ್ನು ಕೇಳಬಹುದು ಅಥವಾ ದಡ್ಡ ಬಳಕೆದಾರರೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಬಹುದು ಅಥವಾ ಇವೆರಡರಲ್ಲಿ ಯಾವುದನ್ನೂ ಮಾಡದಿರಬಹುದು ಮತ್ತು ನೇರವಾಗಿ ವ್ಯವಸ್ಥೆಯ ಮೇಲೆ ದಾಳಿ ಮಾಡಬಹುದು.

ಗುರಿಯ ಸಾಧನೆ (Accomplishment of the goal)

ಒಮ್ಮೆ ಬೆದರಿಕೆಯು ಸಾಧನಕ್ಕೆ ಸೋಂಕು ತಗುಲಿಸಿದ ಮೇಲೆ ಅದು ತನ್ನ ಗುರಿಯನ್ನು ಸಹ ಸಾಧಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತದೆ. ಅದು ಸಾಮಾನ್ಯವಾಗಿ ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ : ಹಾರ್ಡ್‌ವೇರ್‌ಗೆ ಹಾನಿಯನ್ನುಂಟು ಮಾಡುವುದು, ಡೇನಿಯಲ್ ಆಫ್ ಸರ್ವೀಸ್ (DoS), ಹಣದ ಹಾನಿ, ಡಾಟಾದ ಮತ್ತು /ಅಥವಾ ಸಾಧನದ ಹಾನಿ ಮತ್ತು ಮರೆಮಾಡುವ ಹಾನಿ ಇತ್ಯಾದಿಗಳು

ಇತರ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಹರಡುವುದು (Spreading to other systems)

ಒಮ್ಮೆ ಬೆದರಿಕೆಯು ಸಾಧನಕ್ಕೆ ತಗುಲಿದ ಮೇಲೆ, ಅದು ಯಾವಾಗಲೂ ಇತರ ಸಾಧನಗಳಿಗೆ ಹರಡುವ ಗುರಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ.

ಇದು ನೆಟ್‌ವರ್ಕ್‌ಗಳ ಮೂಲಕ , ವೈಯರ್ಡ್ ಅಥವಾ ವೈಯರ್‌ಲೆಸ್ ಮೂಲಕ, ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ವೈ ಫೈ ಬಳಸುವ ಪ್ರಾಕ್ಸಿಮೇಟ್ ಸಾಧನಗಳ ಮೂಲಕ, ಬ್ಲೂಟೂಥ್ ಮತ್ತು ಇನ್‌ಫ್ರಾರೆಡ್ ಲೈಟ್‌ಗಳ ಮೂಲಕ, ಹಂಚುವ ಸಾಧನಗಳ ಮೂಲಕ ಹರಡುತ್ತದೆ.

ಬೆದರಿಕೆಗಳ ಮೂಲಗಳು (Sources of Threats (Fig 1)

ಬೆದರಿಕೆಗಳ ಪ್ರಾಥಮಿಕ ಮೂಲಗಳು ಯಾವವೆಂದರೆ - ಉದ್ಯೋಗಿಗಳು/ಒಳಗಿನವರು, ದುರುದ್ದೇಶಪೂರಿತ ಹ್ಯಾಕರ್‌ಗಳು, ನೈಸರ್ಗಿಕ ವಿಕೋಪಗಳು, ವಿದೇಶಿ ಅಥವಾ ಹೊರಗಿನ ವಿರೋಧಿಗಳು, ಮತ್ತು

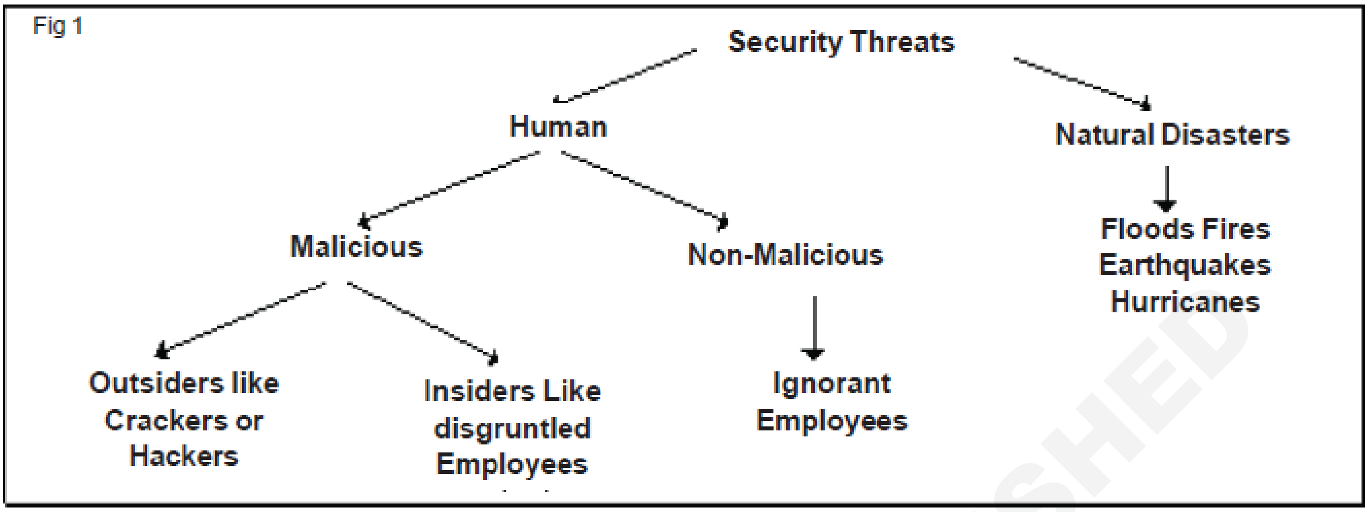
© NIMI NOT TO BE REPUBLISHED

ಪ್ರತಿಕೂಲ ದಾಳಿಗಳು. ಹಲವಾರು ಸಂದರ್ಭಗಳಲ್ಲಿ, ಬೆದರಿಕೆಗಳ ಮೂಲಗಳ ಪ್ರದೇಶಗಳು ಅತಿಕ್ರಮಿಸಲ್ಪಡಬಹುದು. ಉದಾಹರಣೆಗೆ, ಪ್ರತಿಕೂಲ ದಾಳಿಗಳನ್ನು ವಿದೇಶಿ ವಿರೋಧಿಗಳು ಅಥವಾ ಅತೃಪ್ತ ಉದ್ಯೋಗಿಗಳು ನಡೆಸಿರಬಹುದು.

ನೈಸರ್ಗಿಕ ವಿಪತ್ತುಗಳು (Natural Disasters)

ಭೂಕಂಪಗಳು, ಚಂಡಮಾರುತಗಳು, ಪ್ರವಾಹಗಳು, ಸಿಡಿಲು ಮತ್ತು ಬೆಂಕಿಯು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ಗೆ ತೀವ್ರ ಹಾನಿಯುಂಟು ಮಾಡಬಹುದು. ಮಾಹಿತಿಯು ಕಳೆದುಹೋಗಬಹುದು,

Fig 1



ಉತ್ಪಾದಕತೆಯ ನಷ್ಟ ಸಂಭವಿಸಬಹುದು ಮತ್ತು ಹಾರ್ಡ್‌ವೇರ್‌ಗೆ ಹಾನಿಯುಂಟಾಗಬಹುದು. ಇವೆಲ್ಲವೂ ಇತರ ಅಗತ್ಯ ಸೇವೆಗಳಿಗೆ ಅಡ್ಡಿಪಡಿಸಬಹುದು. ಕೆಲವು ರಕ್ಷಣಾತ್ಮಕ ಕ್ರಮಗಳನ್ನು ನೈಸರ್ಗಿಕ ವಿಕೋಪಗಳ ವಿರುದ್ಧ ಅಳವಡಿಸಲಾಗಿದೆ. ಅತ್ಯುತ್ತಮ ವಿಧಾನಗಳೆಂದರೆ - ಸ್ಥಳದಲ್ಲಿ ವಿಪತ್ತು ಚೇತರಿಕೆ ಯೋಜನೆಗಳು ಮತ್ತು ಆಕಸ್ಮಿಕ ಯೋಜನೆಗಳನ್ನು ಹೊಂದಿರುವುದು.

ಮಾನವ ಬೆದರಿಕೆಗಳು (Human Threats)

A **ಉದ್ಯೋಗಿಗಳು/ಒಳಗಿನವರು** (Employees/Insiders): ಉದ್ದೇಶಪೂರ್ವಕ ಮತ್ತು ಆಕಸ್ಮಿಕ ದೋಷಗಳು ಮತ್ತು ಉದ್ಯೋಗಿಗಳ ಹಾಗೂ ಒಳಗಿನವರ ದುರುದ್ದೇಶಪೂರಿತ ಕೃತ್ಯಗಳು ದೂರಸಂಪರ್ಕ ಉದ್ಯಮದಲ್ಲಿ ಗಣನೀಯ ಪ್ರಮಾಣದ ಹಾನಿಯನ್ನು ಮತ್ತು ನಷ್ಟವನ್ನು ಉಂಟು ಮಾಡುತ್ತವೆ.

ಅತೃಪ್ತ ಉದ್ಯೋಗಿಗಳು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಕಿಡಿಗೇಡಿತನದ ಮತ್ತು ವಿಧ್ವಂಸಕ ಕೃತ್ಯ ಎರಡನ್ನೂ ರಚಿಸಬಹುದು. ಸಾರ್ವಜನಿಕ ಮತ್ತು ಖಾಸಗಿ ಎರಡೂ ವಲಯಗಳಲ್ಲಿ ಅವರ ಕೆಲಸಗಳಿಂದ ತೆಗೆದುಹಾಕಲಾದ ಸಿಬ್ಬಂದಿಗಳು ಒಂದು ಗುಂಪನ್ನು ರಚಿಸಿವೆ . ಇವರು ಸಂಸ್ಥೆಯ ಪ್ರಮುಖ ಜ್ಞಾನವನ್ನು ಹೊಂದಿರುವ ವ್ಯಕ್ತಿಗಳು ಸಿಸ್ಟಮ್ ಪ್ರವೇಶವನ್ನು ಉಳಿಸಿಕೊಂಡಿರಬಹುದು. ಸಿಸ್ಟಮ್ ನಿರ್ವಾಹಕರು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಅಮಾನ್ಯಗೊಳಿಸುವ ಮೂಲಕ ಮತ್ತು ಸಮಯೋಚಿತವಾಗಿ ಸಿಸ್ಟಮ್ ಖಾತೆಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುವ ಮೂಲಕ ಈ ಬೆದರಿಕೆಗಳನ್ನು ಮಿತಿಗೊಳಿಸಬಹುದು.

ದುರುದ್ದೇಶಪೂರಿತ ಹ್ಯಾಕರ್‌ಗಳು (Malicious Hackers)

ದುರುದ್ದೇಶಪೂರಿತ ಬೆದರಿಕೆಗಳು ಅತೃಪ್ತರಿಂದ ಮತ್ತು ದುರುದ್ದೇಶಪೂರಿತ ಉದ್ಯೋಗಿಗಳ ಒಳಗಿನ ದಾಳಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ ಮತ್ತು ಅಲ್ಲಿನ ಉದ್ಯೋಗಿಗಳಲ್ಲದ ಆದರೆ ಕೇವಲ ಒಂದು ಸಂಸ್ಥೆಗೆ ಹಾನಿ ಮಾಡಲು ಮತ್ತು ಅಡ್ಡಿಪಡಿಸಲು ನೋಡುವ ಹೊರಗಿನ ದಾಳಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.ದುರುದ್ದೇಶಪೂರಿತ ದಾಳಿಕೋರರು ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ಸಂಸ್ಥೆಯ ಮೇಲೆ ದಾಳಿ ಮಾಡಲು ನಿರ್ದಿಷ್ಟ ಗುರಿ, ಅಥವಾ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿರುತ್ತಾರೆ

ಈ ಗುರಿಗಳು ಡೇನಿಯಲ ಆಫ್ ಸರ್ವಿಸ್ (DoS) ದಾಳಿಯ ಸಾಧನಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಸೇವೆಗಳನ್ನು ಮತ್ತು ವ್ಯಾಪಾರ ಕಾರ್ಯಾಚರಣೆಗಳ ನಿರಂತತೆಯನ್ನು ಅಡ್ಡಿಪಡಿಸುತ್ತವೆ. ಅವು ಮಾಹಿತಿಯನ್ನು ಕದಿಯಲು ಬಯಸಬಹುದು ಅಥವಾ ಲ್ಯಾಪ್‌ಟಾಪ್ ಕಂಪ್ಯೂಟರ್‌ನಂತಹ ಹಾರ್ಡ್‌ವೇರನ್ನೂ ಸಹ ಕದಿಯುತ್ತಾರೆ. ಹ್ಯಾಕರ್‌ಗಳು ಉಪಯುಕ್ತ ಮಾಹಿತಿಯನ್ನು ಪ್ರತಿಸ್ಪರ್ಧಿಗಳಿಗೆ ಮಾರಾಟ ಮಾಡಬಹುದು.

ಅತ್ಯಂತ ಅಪಾಯಕಾರಿ ದಾಳಿಕೋರರು ಸಾಮಾನ್ಯವಾಗಿ ಒಳಗಿನವರಾಗಿರುತ್ತಾರೆ (ಅಥವಾ ಈ ಹಿಂದೆ ಕೆಲಸ ಮಾಡಿದವರು), ಏಕೆಂದರೆ ಅವರಿಗೆ ಅನೇಕ ಈಗಾಗಲೇ ಬಳಸುವ ಕೋಡ್‌ಗಳು ಮತ್ತು ಭದ್ರತಾ ಕ್ರಮಗಳು ತಿಳಿದಿರುತ್ತವೆ. ಒಳಗಿನವರು ನಿರ್ದಿಷ್ಟ ಗುರಿಗಳನ್ನು ಮತ್ತು ಉದ್ದೇಶಗಳನ್ನು ಹೊಂದಿರುತ್ತಾರೆ. ಮತ್ತು ಸಿಸ್ಟಮ್‌ಗೆ ಕಾನೂನುಬದ್ಧ ಪ್ರವೇಶವನ್ನೂ ಹೊಂದಿರುತ್ತಾರೆ. ನೌಕರರು ಸಂಸ್ಥೆಯ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗೆ ಹೆಚ್ಚು ಪರಿಚಿತರಾಗಿರುತ್ತಾರೆ. ಮತ್ತು ಯಾವ ಕ್ರಮಗಳು ಹೆಚ್ಚು ಹಾನಿ ಉಂಟು ಮಾಡಬಲ್ಲವು ಎಂಬುದನ್ನು ತಿಳಿದುಕೊಳ್ಳುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ. ಒಳಗಿನವರು ವೈರಸ್‌ಗಳನ್ನು, ಟ್ರೋಜನ್ ಹಾರ್ಸ್‌ಗಳನ್ನು, ಅಥವಾ ವರ್ಮ್‌ಗಳನ್ನು ಹಾಕಬಲ್ಲರು ಮತ್ತು ಫೈಲ್ ಸಿಸ್ಟಮ್ ಮೂಲಕ ಬ್ರೌಸ್ ಮಾಡಬಲ್ಲರು.

© NIMI NOT TO BE REPUBLISHED

ಅದಾಗ್ಯೂ, ಅತೃಪ್ತರಾದ ಪ್ರಸ್ತುತ ಉದ್ಯೋಗಿಗಳು ವಾಸ್ತವವಾಗಿ ಹಿಂದಿನ ಉದ್ಯೋಗಿಗಳಿಗಿಂತ ಹೆಚ್ಚು ಹಾನಿಯನ್ನು ಉಂಟು ಮಾಡಬಹುದು. ಕಂಪ್ಯೂಟರ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ಉದ್ಯೋಗಿಗಳ ವಿಧ್ವಂಸಕ ಕೃತ್ಯಗಳ ಸಾಮಾನ್ಯ ಉದಾಹರಣೆಗಳು ಹೀಗಿವೆ :

- ಡಾಟಾವನ್ನು ಬದಲಾಯಿಸುವುದು
- ಡಾಟಾವನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು
- ಲಾಜಿಕ್ ಬಾಂಬ್ ಉಪಯೋಗಿಸಿ ಡಾಟಾ ಅಥವಾ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ನಾಶಪಡಿಸುವುದು
- ಸಿಸ್ಟಮ್ ಕ್ರ್ಯಾಶ್ ಮಾಡುವುದು
- ಡಾಟಾವನ್ನು ಒತ್ತೆಯಾಗಿರಿಸುವುದು
- ಹಾರ್ಡ್‌ವೇರ್ ಸೌಲಭ್ಯಗಳನ್ನು ನಾಶಪಡಿಸುವುದು
- ಡಾಟಾವನ್ನು ತಪ್ಪಾಗಿ ನಮೂದಿಸುವುದು

ವಿದೇಶಿ ವಿರೋಧಿಗಳು (Foreign Adversaries)

ಕಂಪ್ಯೂಟರ್ ಒಳನುಗ್ಗುವ ಚಟುವಟಿಕೆಗಳು ಅಂತರರಾಷ್ಟ್ರೀಯ ಮಟ್ಟದಲ್ಲಿ ಸಂಭವಿಸಿವೆ. ಅಂತರರಾಷ್ಟ್ರೀಯ ಗೇಟ್‌ವೇ ಗಳ ಮೂಲಕ ಒಳನುಗ್ಗುವಿಕೆಯ ಪ್ರಯತ್ನದ ಸಂಖ್ಯೆಯು ಅಪಾಯಕಾರಿ ದರದಲ್ಲಿ ಹೆಚ್ಚುತ್ತಿದೆ. ಕಂಪ್ಯೂಟರ್ ಅಡಿಯಲ್ಲಿನ ಕೆಲವು ಸೂಚನೆಗಳು ವಿದೇಶಗಳಲ್ಲಿನ ರಾಜಕೀಯ ಕಾರ್ಯಸೂಚಿಗಳನ್ನು ಒಯ್ಯುತ್ತವೆ. ಕೆಲವೊಮ್ಮೆ ಒಂದು ದೇಶದ ಗುಪ್ತಚರ ಸೇವೆಗಳು ನೇರವಾಗಿ ಇತರ ದೇಶಗಳ ಸಂವಹನ ವ್ಯವಸ್ಥೆಯನ್ನು ಗುರಿಪಡಿಸುತ್ತವೆ, ಭೇದಿಸುತ್ತವೆ ಅಥವಾ ರಾಜಿ ಮಾಡಿಕೊಳ್ಳುತ್ತವೆ.

ಹೊರಗಿನ ದಾಳಿಕೋರರು ಅಥವಾ ಕ್ರ್ಯಾಕರ್ಸ್ (Outside Attackers or ‍Crackers)

ಜನರು ಸಾಮಾನ್ಯವಾಗಿ "ಕ್ರ್ಯಾಕರ್ಸ್"ನ್ನು "ಹ್ಯಾಕರ್ಸ್" ಎಂದು ಅನ್ನುತ್ತಾರೆ. ಹ್ಯಾಕರ್ ಎನ್ನುವ ಪದವು ಯಾವುದೇ ಅಧಿಕಾರವನ್ನು ಹೊಂದಿಲ್ಲದ ಅಥವಾ ಸಿಸ್ಟಮ್‌ಗಳ ಮೇಲೆ ಉದ್ದೇಶಪೂರ್ವಕವಾಗಿ ಅತಿಕ್ರಮಣವನ್ನು ನಡೆಸುವ ಜನರನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದಕ್ಕಾಗಿ ಅವರು ಕಾನೂನುಬದ್ಧ ಪ್ರವೇಶವನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ. ಒಂದು ಸಿಸ್ಟಮ್‌ಗೆ ಪ್ರವೇಶವನ್ನು ಪಡೆಯುವ ಸಾಮಾನ್ಯ ವಿಧಾನಗಳು ಎಂದರೆ -ಪಾಸ್‌ವರ್ಡ್ ಕ್ರ್ಯಾಕಿಂಗ್, ತಿಳಿದಿರುವ ಭದ್ರತಾ ದೌರ್ಬಲ್ಯಗಳನ್ನು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳುವುದು, ನೆಟ್‌ವರ್ಕ್ ವಂಚನೆ, ಮತ್ತು ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್.

ಪ್ರತಿಕೂಲ ದಾಳಿಗಳು (Hostile Attacks)

ಪ್ರತಿಕೂಲ ದಾಳಿಗಳ ಮೂಲಕ, ನೆಟ್‌ವರ್ಕ್‌ಗಳ ಲಭ್ಯತೆಯ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ. ಪ್ರತಿಕೂಲ ದಾಳಿಗಳ ಪ್ರಾಥಮಿಕ ಪರಿಣಾಮಗಳಾದ- ಪರಮಾಣು ದಾಳಿಗಳು, ಸೀಮಿತ/ಸಂಘಟಿತವಲ್ಲದ ಪರಮಾಣು ದಾಳಿಗಳು, ಪರಮಾಣು ಅಪಘಾತಗಳು, ಭಯೋತ್ಪಾದನೆ, ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಯುದ್ಧ, ವಿಧ್ವಂಸಕತೆ, ಮತ್ತು ನಾಗರೀಕ ಅಸ್ವಸ್ಥತೆಗಳು ರಾಷ್ಟ್ರದ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ತೊಂದರೆ ಮತ್ತು ಸೇವೆಗೆ ಅಡ್ಡಿಯುಂಟು ಮಾಡುತ್ತದೆ. ಇಂತಹ ವಿಪತ್ತುಗಳು ಸಕಾಲಿಕತೆಯ ಮೇಲೆ ಮತ್ತು ವಿತರಿಸಿದ ಸೇವೆಗಳ ಗುಣಮಟ್ಟದ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ.

B **ದುರುದ್ದೇಶಪೂರಿತರಲ್ಲದ ಉದ್ಯೋಗಿಗಳು** (Non-Malicious Employees)

ದಾಳಿಕೋರರು ಒಂದು ಸಂಸ್ಥೆಗೆ ಮಾತ್ರ ಹಾನಿ ಉಂಟುಮಾಡುವವಲ್ಲ. ಅಜ್ಞಾನ ಬಳಕೆದಾರರಿಂದಲೂ ಡಾಟಾ ಸಮಗ್ರತೆಗೆ ಪ್ರಾಥಮಿಕವಾಗಿ ಬೆದರಿಕೆ ಬರುತ್ತದೆ. ಇವರು ಅಧಿಕೃತ ಬಳಕೆದಾರರಾಗಿದ್ದಾರೆ. ಆದರೆ ತಾವು ನಿರ್ವಹಿಸುತ್ತಿರುವ ಕ್ರಿಯೆಗಳ ಬಗ್ಗೆ ತಿಳಿದಿರುವುದಿಲ್ಲ. ದೋಷಗಳು ಮತ್ತು ಲೋಪಗಳು ಮೌಲ್ಯಯುತವಾದ ಡಾಟಾವನ್ನು ಕಳೆದುಕೊಳ್ಳಬಹುದು, ಹಾನಿಗೊಳಗಾಗಬಹುದು ಅಥವಾ ಬದಲಾಯಿಸಬಹುದು.

ಬಳಕೆದಾರರು, ಡಾಟಾ ಎಂಟ್ರಿ ಮಾಡುವ ಕ್ಲಾರ್ಕ್‌ಗಳು, ಸಿಸ್ಟಮ್ ಆಪರೇಟರ್‌ಗಳು, ಮತ್ತು ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು, ಆಗಾಗ್ಗೆ ಉದ್ದೇಶಪೂರ್ವಕವಲ್ಲದ ತಪ್ಪುಗಳನ್ನು ಮಾಡುತ್ತಾರೆ. ಇವು ನೇರವಾಗಿ ಮತ್ತು ಪರೋಕ್ಷವಾಗಿ ಭದ್ರತಾ ಸಮಸ್ಯೆಗಳನ್ನು ಉಂಟು ಮಾಡುತ್ತವೆ. ಕೆಲವೊಮ್ಮೆ ದೋಷವು ಬೆದರಿಕೆಯಾಗಿರುತ್ತದೆ ಉದಾಹರಣೆಗೆ ಡಾಟಾ ಎಂಟ್ರಿ ದೋಷ ಅಥವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ದೋಷಗಳು ಸಿಸ್ಟಮ್‌ನ್ನು ಕ್ರ್ಯಾಶ್ ಮಾಡುತ್ತವೆ. ಇತರ ಸಂದರ್ಭಗಳಲ್ಲಿ ದೋಷಗಳು ದುರ್ಬಲತೆಯನ್ನು ಸೃಷ್ಟಿಸುತ್ತವೆ.

ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸಲು ಅತ್ಯುತ್ತಮ ಅಭ್ಯಾಸಗಳು ಅಥವಾ ಮಾರ್ಗಸೂಚಿಗಳು (Best Practices or Guidelines used to Identify Threats)

ವಿವಿಧ ಯಾಂತ್ರಿಕ ವ್ಯವಸ್ಥೆಗಳು ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಬೆದರಿಕೆಗಳನ್ನು /ದಾಳಿಗಳನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಗುರುತಿಸಬಹುದು. ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ಬೆದರಿಕೆಯ ಆಧಾರದ ಮೇಲೆ ನೀವು ನಿರ್ದಿಷ್ಟ ತಂತ್ರಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಅವುಗಳನ್ನು ಗುರುತಿಸಬಹುದು ಮತ್ತು ವರ್ಗೀಕರಿಸಬಹುದು. ಕೆಳಗಿನವುಗಳು ಹೆಚ್ಚು ಸಾಮಾನ್ಯವಾದ ವಿಧಾನಗಳಾಗಿವೆ :

- ಅಸಮಂಜಸ ಪತ್ತೆ ಸಾಧನಗಳ ಬಳಕೆ
- ಫ್ಲೋ ಆಧಾರಿತ ವಿಶ್ಲೇಷಣೆಯನ್ನು ಬಳಸಿಕೊಂಡು ನೆಟ್‌ವರ್ಕ್ ಟೆಲಿಮೆಟ್ರಿ
- ಒಳನುಗ್ಗುವಿಕೆಯನ್ನು ಪತ್ತೆ ಮಾಡುವುದು ಮತ್ತು ಒಳನುಗ್ಗುವಿಕೆಯನ್ನು ತಡೆಗಟ್ಟುವ ವ್ಯವಸ್ಥೆಗಳ ಬಳಕೆ (IDS/IPS)
- ನೆಟ್‌ವರ್ಕ್ ಕಾಂಪೋನೆಂಟ್ ಲಾಗ್‌ಗಳನ್ನು ವಿಶ್ಲೇಷಿಸುವುದು(

© NIMI NOT TO BE REPUBLISHED

ಎಂದರೆ, ವಿಭಿನ್ನ ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳಿಂದ, ಲೆಕ್ಕಪತ್ರ ದಾಖಲೆಗಳು, ಅಪ್ಲಿಕೇಷನ್ ಲಾಗ್‌ಗಳು, Simple Network Management Protocol (SNMP) ಇತ್ಯಾದಿಗಳಿಂದ SYSLOG)

ಬೆದರಿಕೆಗಳನ್ನು ತಗ್ಗಿಸುವಿಕೆಯಲ್ಲಿ ಬಳಸಲಾಗುವ ಅತ್ಯುತ್ತಮ ಅಭ್ಯಾಸಗಳು ಮತ್ತು ಮಾರ್ಗಸೂಚಿಗಳು (Best Practices or Guidelines used in mitigation of threats)

ಕೆಳಗಿನ ಕೆಲವು ಅಭ್ಯಾಸಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಿದಾಗ ಅವು ಸಂಸ್ಥೆಯ ಮಾಹಿತಿ, ಡಾಟಾ ಮತ್ತು ವಿಶ್ವಾಸಾರ್ಹತೆಯ ಭದ್ರತೆಗೆ ಇರುವ ಬೆದರಿಕೆಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಸಾಧ್ಯತೆ ಇದೆ.

ಭದ್ರತಾ ಜಾಗೃತಿ ತರಬೇತಿ (Security Awareness Training)

- ಹೆಚ್ಚಿನ ಭದ್ರತಾ ಉಲ್ಲಂಘನೆಗಳು ವಾಸ್ತವವಾಗಿ ಕಂಪನಿಗಳ ಒಳಗೆ ಅತೃಪ್ತ ಅಥವಾ ನಿರ್ಲಕ್ಷದ ಉದ್ಯೋಗಿಗಳಿಂದ ಹುಟ್ಟಿಕೊಳ್ಳುತ್ತವೆ. ಅದ್ದರಿಂದ ನಿಮ್ಮ ಕಂಪನಿಯಲ್ಲಿ ಎಲ್ಲರಿಗೂ ತರಬೇತಿಯನ್ನು ನೀಡಿ. ಇದರಿಂದ ಅವರು ವಿವಿಧ ಭದ್ರತಾ ಅಪಾಯಗಳನ್ನು ಗುರುತಿಸಲು ಸಹಾಯ ಆಗುತ್ತದೆ.
- ಉದ್ಯೋಗಿಗಳು ಇ ಮೇಲ್ ಫಿಶಿಂಗ್ ಮತ್ತು ಸ್ಪೂಫಿಂಗ್ ದಾಳಿಗಳನ್ನು ಗುರುತಿಸಲು ಸಾಧ್ಯವಾಗುತ್ತದೆ. ಅವರಿಗೆ ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡದೇ ಇರುವ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸದಿರಲು, ಕಳುಹಿಸದಿರಲು ಅಥವಾ ಕಾಪಿ ಮಾಡದಿರಲು ತರಬೇತಿ ನೀಡಬೇಕು.
- ಅವರು ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಫೋನ್‌ನಲ್ಲಿ ಹಂಚಿಕೊಳ್ಳಬಾರದು ಎಂದು ತಿಳಿದಿರಬೇಕು.
- ಉದ್ಯೋಗಿಗಳಿಗೆ ಭದ್ರತಾ ನೀತಿ ಮತ್ತು ಅಭ್ಯಾಸಗಳ ಬಗ್ಗೆ ತರಬೇತಿ ನೀಡಿ. ಮತ್ತು ಅವುಗಳನ್ನು ನವೀಕರಿಸುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ ಮತ್ತು ಆಗಾಗ್ಗೆ ಮರುತರಬೇತಿ ನೀಡಿ.
- ತರಬೇತಿ ಸಾಮಗ್ರಿಗಳು, ಕಾರ್ಪೋರೇಟ್ ನೀತಿಗಳನ್ನು ಸಹ ಪರಿಶೀಲಿಸಬೇಕು ಮತ್ತು ಉದ್ಯೋಗಿಗಳಲ್ಲಿ ಯಾವುದೇ ಅನುಮಾನಾಸ್ಪದ ಅಥವಾ ದುರುದ್ದೇಶಪೂರಿತ ವರ್ತನೆಗಳ ಪರಿಣಾಮಗಳನ್ನು ಸ್ಪಷ್ಟವಾಗಿ ವಿವರಿಸಿ. ವಿವಿಧ ಭದ್ರತಾ ನೀತಿಗಳ ಬಗ್ಗೆ ಅವರಿಗೆ ತರಬೇತಿ ನೀಡಬೇಕು, ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ;

- ಸ್ವೀಕಾರಾರ್ಹವಾದ ಬಳಕೆ ಮತ್ತು ಅಸ್ವೀಕಾರಾರ್ಹವಾದ ಬಳಕೆ
- ಸಾಮಾನ್ಯ ಬಳಕೆ ಮತ್ತು ಮಾಲೀಕತ್ವ
- ಭದ್ರತೆ ಮತ್ತು ಸ್ವಾಮ್ಯದ ಮಾಹಿತಿ
- ಬ್ಲಾಗಿಂಗ್ ಮತ್ತು ಸಾಮಾಜಿಕ ಮಾಧ್ಯಮ
- ಜಾರಿಗೊಳಿಸುವಿಕೆ ಎಂದರೆ, ಅನುಸರಣೆಗಾಗಿ ಶಿಸ್ತು ಕ್ರಮ
- ಸಾಮಾಜಿಕ ಮಾಧ್ಯಮ
- ನಿಮ್ಮ ಸ್ವಂತ ಸಾಧನ ನೀತಿಗಳನ್ನು ತನ್ನಿ
  - ಡಾಟಾ ನೀತಿ
  - ಮೊಬೈಲ್ ಸಾಧನ ನಿರ್ವಹಣೆ (MDM) ನೀತಿ
  - ಮೊಬೈಲ್ ಸಾಧನ ಬೆಂಬಲ ನೀತಿ
  - ಕಂಪನಿ ನೀಡಿದ ಸಾಧನಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ನೀತಿಗಳು
  - ನಷ್ಟ ಮತ್ತು ಕಳ್ಳತನ
  - ಉದ್ಯೋಗಿ ವಜಾಗೊಳಿಸುವ ನೀತಿ
  - ಭದ್ರತಾ ಘಟನೆ ನಿರ್ವಹಣೆ

ಆ್ಯಂಟಿ ವೈರಸ್ ಮತ್ತು ಆ್ಯಂಟಿ ಮಾಲ್ವೇರ್ ರಕ್ಷಣೆ (Anti-Virus & Anti-Malware Protection)

- ಮಾಲ್‌ವೇರ್‌ನ ಸಾಮಾನ್ಯ ರೂಪಗಳೆಂದರೆ : ವರ್ಮ್‌ಗಳು, ಕೀ ಲಾಗರ್‌ಗಳು, ವಿಡಿಯೊ ಫ್ರೇಮ್ ಗ್ರ್ಯಾಬರ್‌ಗಳು, ರೂಟ್‌ಕಿಟ್‌ಗಳು, ಮತ್ತು ಟ್ರೋಜನ್ ಹಾರ್ಸ್‌ಗಳು.
- ಉತ್ತಮ ಆ್ಯಂಟಿ ವೈರಸ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಸ್ಥಾಪಿಸಿ, ನವೀಕರಿಸಿ, ನಿಗದಿಪಡಿಸಿ ಮತ್ತು ರನ್ ಮಾಡಿ.
- ಮಾಲ್‌ವೇರ್ ಬೆದರಿಕೆಗಳ ವಿರುದ್ಧ ಹೋರಾಡಲು "end point security" ತಂತ್ರವನ್ನು ಅಳವಡಿಸಿಕೊಳ್ಳಿರಿ. ಎಂಡ್ ಪಾಯಿಂಟ್ ಭದ್ರತೆಯು ಒಂದು ಮಾಹಿತಿ ಭದ್ರತಾ ಪರಿಕಲ್ಪನೆಯಾಗಿದೆ. ಎಂದರೆ, ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಪ್ರತಿಯೊಂದು ಸಾಧನವು (ಅಥವಾ ಎಂಡ್ ಪಾಯಿಂಟ್) ತನ್ನ ಸ್ವಂತ ಭದ್ರತೆಯ ಜವಾಬ್ದಾರಿಯನ್ನು ಹೊಂದಿರಬೇಕು ಮತ್ತು ಭದ್ರತೆಯನ್ನು ಒದಗಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿರಬೇಕು.
- ನಿಮ್ಮ ಮಾಲ್‌ವೇರ್‌ನ ಪರಿಹಾರವು ಏನೇ ಆಗಿದ್ದರೂ, ಅದು ಇ ಮೇಲ್‌ನೊಂದಿಗೆ ಲಗತ್ತಿಸಲಾದ ವೈರಸ್‌ಗಳಿಗಾಗಿ ಸ್ಕ್ಯಾನ್ ಮಾಡಬೇಕು, ಸೋಂಕುಗಳಿಗಾಗಿ ಫೈಲುಗಳನ್ನು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡಬೇಕು, ಮತ್ತು ಪ್ರತಿಯೊಂದು ಫೈಲನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಸ್ಕ್ಯಾನ್ ಮಾಡಬೇಕು.

ಡಾಟಾ ಎನ್‌ಕ್ರಿಪ್ಷನ್ (Data Encryption)

- ಡಾಟಾ ಎನ್‌ಕ್ರಿಪ್ಷನ್, ಮಾಹಿತಿ ಭದ್ರತೆಯ ಪ್ರಬಲವಾದ ಭಾಗವಾಗಿದೆ. ಎನ್‌ಕ್ರಿಪ್ಷನ್ ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಅದರ ಪ್ರವೇಶದ ನಂತರವೂ ರಕ್ಷಿಸಲಾಗುತ್ತದೆ.

ಪ್ಯಾಚಿಂಗ್ (Patching)

- ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ಗಳ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡಲು ಪ್ಯಾಚಿಂಗ್ ಅತ್ಯಗತ್ಯವಾಗಿದೆ. ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ

ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಇರುವ ಭದ್ರತಾ ರಂಧ್ರಗಳನ್ನು ಸರಿಪಡಿಸಲು ಹೆಚ್ಚಾಗಿ ಪ್ಯಾಚ್‌ಗಳನ್ನು ಬಿಡುಗಡೆ ,ಮಾಡಲಾಗುತ್ತದೆ. ಎಲ್ಲ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಪ್ಯಾಚ್ ಆಗಿ ರನ್ ಮಾಡುತ್ತಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ. ಎಲ್ಲ ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳಲ್ಲಿ ಇತ್ತೀಚಿನ ಫರ್ಮ್‌ವೇರ್ ನವೀಕರಣಗಳನ್ನು ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಿರಿ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣಗಳು (Access Controls)

- ಹೆಚ್ಚಿನ ಭದ್ರತೆಗಾಗಿ, ಉದ್ಯೋಗಿಗಳಿಗೆ (ಮತ್ತು ಪಾಲುದಾರರು) ಅಗತ್ಯವಿರುವ ಡಾಟಾದ ಪ್ರವೇಶವನ್ನು ಮಾತ್ರ ನೀಡಿ. ಇದು ಭೌತಿಕ ಮತ್ತು ತಾರ್ಕಿಕ ಪ್ರವೇಶ ಎರಡನ್ನೂ ಒಳಗೊಂಡಿದೆ.
- ಕನಿಷ್ಟ ಸವಲತ್ತನ್ನು ನೀಡುವ ಮೂಲಕ ಪ್ರಾರಂಭಿಸಿ. ಆನಂತರ ನೀವು ಸವಲತ್ತುಗಳನ್ನು ಹೆಚ್ಚಿಸಿ ಅಗತ್ಯವಿರುವ ಆಧಾರದ ಮೇಲೆ ಅನಧಿಕೃತ ಡಾಟಾಗೆ ಪ್ರವೇಶವನ್ನು ನೀಡಬಹುದು.

ಮೊಬೈಲ್ ಸಾಧನಗಳು (Mobile Devices)

- ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳು, ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ಗಳು, ಮತ್ತು ಟ್ಯಾಬ್ಲೆಟ್‌ಗಳು ಇವತ್ತಿನ ಉದ್ಯೋಗಿಗಳ ಉತ್ಪಾದಕತೆಯನ್ನು ಮತ್ತು ಚಲನಶೀಲತೆಯನ್ನು ಹೆಚ್ಚಿಸಿವೆ. ಆದರೆ ಉತ್ಪಾದಕತೆಯ ಜೊತೆಗೆ ದುರ್ಬಲತೆಯೂ ಬರುತ್ತದೆ. ಕದ್ದ ಲ್ಯಾಪ್‌ಟಾಪ್‌ಗಳು ಮತ್ತು ಇತರ ಮೊಬೈಲ್ ಸಾಧನಗಳು ಡಾಟಾ ಉಲ್ಲಂಘನೆಯ ಪ್ರಮುಖ ಕಾರಣಗಳಾಗಿವೆ.
- ಎಲ್ಲ ಸಾಧನಗಳನ್ನು ಪ್ರವೇಶಿಸಲು, ಸ್ವಯಂ- ಲಾಕ್ (auto lock) ಸಕ್ರೀಯಗೊಳಿಸಿ ಅಥವಾ ಪಾಸ್‌ವರ್ಡ್‌ನ ಅಗತ್ಯವಿದೆ.

ಉಸ್ತುವಾರಿ (Monitoring)

- ಯಾವುದೇ ಅಸಹಜತೆಗಳನ್ನು ಕಂಡುಹಿಡಿಯಲು, ನಿಮ್ಮ ವ್ಯಾಪಾರವು ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಾಧನಗಳಿಗೆ ಮೇಲ್ವಿಚಾರಣೆಯನ್ನು ಸೆಟ್ ಮಾಡಲಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ.
- ಎಲ್ಲಾ ಸ್ಥಳಗಳಿಂದ ಅಥವಾ ನೆಟ್‌ವರ್ಕ್, ಸಿಸ್ಟಮ್‌ಗಳು, ಮತ್ತು ಬಳಕೆದಾರರ ಚಟುವಟಿಕೆಗಳಂತಹ ಮೂಲಸೌಕರ್ಯಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಿ ಮತ್ತು ಪರಸ್ಪರ ಸಂಬಂಧಿಸಿ.
- ಫೈರ್‌ವಾಲ್ ಮತ್ತು IPS ನಿಂದ ಮಾತ್ರ ಚಟುವಟಿಕೆಗಳನ್ನು ನಿರ್ಬಂಧಿಸಬೇಡಿ. ಅದನ್ನು ಲಾಗ್ ಮಾಡಿ, ಪರಿಶೀಲಿಸಿ ಮತ್ತು ಅದರಿಂದ ಕಲಿಯಿರಿ.
- ನಿಮ್ಮ ವ್ಯಾಪಾರದಲ್ಲಿ ಬಳಕೆದಾರರ ಚಟುವಟಿಕೆಗಳನ್ನು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡಲು ಕಂಟೆಂಟ್ ಫಿಲ್ಟರಿಂಗ್ ಅನ್ನು ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಿರಿ. ಉದ್ಯೋಗಿಗಳು ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ದುರ್ಬಳಕೆ ಮಾಡುವ ಅತ್ಯಂತ ಸಾಮಾನ್ಯ ರೂಪವೆಂದರೆ ಅನಗತ್ಯ ಸೈಟ್‌ಗಳನ್ನು ಸರ್ಫ್ ಮಾಡುವುದು.

ಫೈರ್‌ವಾಲ್ (Firewall)

- ಫೈರ್‌ವಾಲ್‌ನ ನಿಯಮಗಳನ್ನು ಮತ್ತು ನೀತಿಗಳನ್ನು ಸಂರಚನೆ ಮಾಡಿ. ಏಕೆಂದರೆ ಫೈರ್‌ವಾಲ್‌ಗಳು ಯಾವುದೇ ದಾಳಿಯ ವಿರುದ್ಧ ರಕ್ಷಣೆ ಯಲ್ಲಿ ಮೊದಲನೇಯದಾಗಿವೆ. (ನೆಟ್‌ವರ್ಕ್ ಅಥವಾ ಹೋಸ್ಟ್). ಇದು ಸಾರ್ವಜನಿಕ ನೆಟ್‌ವರ್ಕ್ ಮತ್ತು ಖಾಸಗಿ ನೆಟ್‌ವರ್ಕ್‌ನ ನಡುವೆ ತಡೆಗೋಡೆಯಾಗಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

ರಿಮೋಟ್ ಬ್ಯಾಕಪ್ (Remote Backup)

- ದೂರದ ಸ್ಥಳಕ್ಕೆ ನಿಮ್ಮ ಡಾಟಾವನ್ನು ನಿಯಮಿತವಾಗಿ ಬ್ಯಾಕಪ್ ಮಾಡಿ. ಬ್ಯಾಕಪ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಅತ್ಯಂತ ನಿರ್ಲಕ್ಷಿತ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಅದ್ದರಿಂದ ಸಾಮಾನ್ಯವಾಗಿ ಇದು ನಿಮ್ಮ ವ್ಯಾಪಾರವು ಅಪಾಯವನ್ನು ತಗ್ಗಿಸಲು ಇರುವ ದೊಡ್ಡ ಅವಕಾಶಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ.
- ಸಾಮಾನ್ಯವಾಗಿ ವ್ಯವಹಾರಗಳು ಡಾಟಾವನ್ನು ಹ್ಯಾಕರ್‌ಗಳಿಂದ ಅಥವಾ ಮಾಲ್‌ವೇರ್‌ಗಳಿಂದ ರಕ್ಷಿಸಲು ಹೂಡಿಕೆ ಮಾಡುತ್ತವೆ. ಆದರೆ ನಂತರ ಡಾಟಾ ನೈಸರ್ಗಿಕ ಕಾರಣಗಳಿಂದಾಗಿ ಭೌತಿಕವಾಗಿ ನಾಶವಾಗುತ್ತದೆ. ಒಂದು ವೇಳೆ ಡಾಟಾ ಅಸ್ತಿತ್ವದಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ, ಅದನ್ನು ಹೊರಗಿನ ಬೆದರಿಕೆಗಳಿಂದ ಸುರಕ್ಷಿತಗೊಳಿಸುವುದಕ್ಕೆ ಯಾವುದೇ ಅರ್ಥವಿಲ್ಲ.

ಭದ್ರತಾ ಮೌಲ್ಯಮಾಪನಗಳು & ತೀಕ್ಷ್ಣ ದೃಷ್ಟಿ ಪರೀಕ್ಷೆ(Security Assessments & Penetration Testing)

- ನಿಮ್ಮ ವ್ಯಾಪಾರವನ್ನು ಸುರಕ್ಷಿತವಾಗಿರಿಸಲು ನೀವು ಜಾಗರೂಕರಾಗಿರಬೇಕು. ಇಲ್ಲಿ ಯಾವಾಗಲೂ ತಪ್ಪು ಉದ್ದೇಶಗಳನ್ನು ಹೊಂದಿರುವ ಜನರು ನಿಮ್ಮ ವ್ಯಾಪಾರದ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಾಣಿಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಮಾರ್ಗವನ್ನು ಹುಡುಕುತ್ತಿರುತ್ತಾರೆ.
- ಹೊಸ ಅಪಾಯಗಳನ್ನು ಗುರುತಿಸಲು, ವಾರ್ಷಿಕ ಅಥವಾ ತ್ರೈಮಾಸಿಕವಾಗಿ ದುರ್ಬಲತೆಗಳ ಮೌಲ್ಯಮಾಪನವನ್ನು ಮಾಡಿರಿ. ಸದಾ ಬದಲಾಗುತ್ತಿರುವ ಭದ್ರತಾ ಪರಿಸರವು ಯಾವಾಗಲೂ ಹೊಸ ಅಪಾಯಗಳನ್ನು ಸೃಷ್ಟಿಸುತ್ತಿರುತ್ತದೆ.
- ನಿಮ್ಮ ವ್ಯಾಪಾರಕ್ಕೆ ಅನ್ವಯಿಸುವ ಹೊಸ ಅಪಾಯಗಳನ್ನು ಗುರುತಿಸಿ ಮತ್ತು ಬೇರೆಯವರ ಹುಡುಕುವ ಮೊದಲು ಅವುಗಳನ್ನು ಸರಿಪಡಿಸಿ.
- ಪ್ರತಿ ಮೂರು ವರ್ಷಗಳಿಗೊಮ್ಮೆ ನಡೆಯುವ ಭದ್ರತಾ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದ ಔಪಚಾರಿಕ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯಿರಿ. ಇದು ಇತ್ತೀಚಿನ ದಿನಗಳಲ್ಲಿ ಹೆಚ್ಚಿನ ಉತ್ಪನ್ನಗಳ ಜೀವನ ಚಕ್ರವಾಗಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಮಾಹಿತಿ ಭದ್ರತಾ ದೋಷಗಳು ಮತ್ತು ಅಪಾಯದ ನಿರ್ವಹಣೆ (Information security vulnerabilities and Risk Management)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಭದ್ರತಾ ದುರ್ಬಲತೆಯನ್ನು ವಿವರಿಸಿ
- ದುರ್ಬಲತೆಗಳ ವಿಧಗಳನ್ನು ವಿವರಿಸಿ
- ಬೆದರಿಕೆಗಳು ವ್ಯವಸ್ಥೆಯ ಮೇಲೆ ಹೇಗೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ ಎಂಬುದನ್ನು ವಿವರಿಸಿ
- ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು ಮತ್ತು ತಂತ್ರಗಳನ್ನು ವಿವರಿಸಿ
- ಭದ್ರತಾ ದುರ್ಬಲತೆಗಳನ್ನು ತಗ್ಗಿಸಲು ಉತ್ತಮ ಅಭ್ಯಾಸಗಳನ್ನು ವಿವರಿಸಿ
- ಬೆದರಿಕೆ, ದುರ್ಬಲತೆ ಮತ್ತು ಅಪಾಯದ ನಡುವಿನ ಸಂಬಂಧವನ್ನು ವಿವರಿಸಿ
- ವಿವಿಧ ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳನ್ನು ವಿವರಿಸಿ
- ಅಪಾಯ ನಿರ್ವಹಣೆಯ ಉದ್ದೇಶವನ್ನು ವಿವರಿಸಿ
- ಅಪಾಯ ನಿರ್ವಹಣೆಯ ವಿಧಗಳನ್ನು ವಿವರಿಸಿ
- ICT ಪೂರೈಕೆ ಸರಪಳಿ ನಿರ್ವಹಣೆಯ ಅಪಾಯಗಳನ್ನು ವಿವರಿಸಿ
- ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸಲು ಮತ್ತು ತಗ್ಗಿಸಲು ಉತ್ತಮ ಅಭ್ಯಾಸಗಳನ್ನು ವಿವರಿಸಿ.

ಪರಿಚಯ (Introduction)

ಕಂಪ್ಯೂಟರ್ ಭದ್ರತೆಯಲ್ಲಿ, ದುರ್ಬಲತೆ ಎಂದರೆ ಒಂದು ಸಾಫ್ಟ್‌ವೇರ್, ಹಾರ್ಡ್‌ವೇರ್, ಕಾರ್ಯವಿಧಾನ, ಅಥವಾ ಮಾನವ ದೌರ್ಬಲ್ಯವಾಗಿದೆ. ಇದು ಆಕ್ರಮಣಕಾರರಿಗೆ ಒಂದು ಸಿಸ್ಟಮ್‌ನ ಮಾಹಿತಿ ಭರವಸೆಯನ್ನು ಕಡಿಮೆ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ. ದುರ್ಬಲತೆಯು ಆಕ್ರಮಣಕಾರರಿಗೆ ಒಂದು ಕಂಪ್ಯೂಟರ ಅಥವಾ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಪ್ರವೇಶಿಸಿ ಮತ್ತು ಒಂದು ಪರಿಸರದಲ್ಲಿರುವ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಅನಧಿಕೃತವಾಗಿ ಪ್ರವೇಶ ಪಡೆಯುವುದಕ್ಕೆ ಒದಗಿಸಬಹುದಾದ ದೌರ್ಬಲ್ಯವಾಗಿದೆ. ಇದು ಸುರಕ್ಷತಾ ಕ್ರಮದ ಅನುಪಸ್ಥಿತಿ ಅಥವಾ ದೌರ್ಬಲ್ಯವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

ಬೆದರಿಕೆ, ಬೆದರಿಕೆ ಏಜಂಟ್, ಶೋಷಣೆ, ಅಪಾಯ ಮತ್ತು ದುರ್ಬಲತೆ

(Threat, Threat Agent, Exploit, Risk and Vulnerability.)

- ಬೆದರಿಕೆ (Threat) ಎಂದರೆ, ಮಾಹಿತಿ ಅಥವಾ ಸಿಸ್ಟಮ್‌ಗೆ ಇರುವ ಸಂಭಾವ್ಯ ಅಪಾಯವಾಗಿದೆ. ಅಥವಾ ಯಾರಾದರೂ ಅಥವಾ ಯಾವುದಾದರೂ ಒಂದು ನಿರ್ದಿಷ್ಟ ದುರ್ಬಲತೆಯನ್ನು ಗುರುತಿಸುವುದು ಮತ್ತು ಅದನ್ನು ಕಂಪನಿಯ ಅಥವಾ ವ್ಯಕ್ತಿಯ ವಿರುದ್ಧ ಉಪಯೋಗಿಸುವುದು.
- ದುರ್ಬಲತೆಯ ಲಾಭವನ್ನು ಪಡೆಯುವ ಘಟಕವನ್ನು ಬೆದರಿಕೆಯ ಏಜಂಟ್ (Threat Agent) ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.
- ದುರ್ಬಲತೆಯ ಲಾಭವನ್ನು ಪಡೆಯುವದಕ್ಕೆ ಮತ್ತು ಅದನ್ನು ಸಿಸ್ಟಮ್ ಅಥವಾ ನೆಟ್‌ವರ್ಕ್‌ನ ಲಾಭವನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸುವುದಕ್ಕೆ ಶೋಷಣೆ (Exploit)ಎಂದು ಕರೆಯುತ್ತಾರೆ.
- ಅಪಾಯವು (Risk) , ಬೆದರಿಕೆ ಏಜಂಟ್ ದುರ್ಬಲತೆಯ ಮತ್ತು ಅದಕ್ಕೆ ಅನುಗುಣವಾದ ವ್ಯಾಪಾರದ ಪ್ರಭಾವವನ್ನು ಪಡೆಯುವ ಪ್ರಯೋಜನವನ್ನು ಪಡೆಯುವ ಸಾಧ್ಯತೆಯಾಗಿದೆ.

ಇನ್ನೊಂದು ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ಬೆದರಿಕೆ ವಿರುದ್ಧ ನಾವು ರಕ್ಷಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತೇವೆ. ದುರ್ಬಲತೆ ಎಂದರೆ ನಮ್ಮ ರಕ್ಷಣಾ ಪ್ರಯತ್ನಗಳಲ್ಲಿರುವ ದೌರ್ಬಲ್ಯ ಅಥವಾ ಅಂತರವಾಗಿದೆ.

ಮಾಹಿತಿ ಭದ್ರತಾ ದೋಷಗಳು ಏಕೆ ಅಸ್ತಿತ್ವದಲ್ಲಿವೆ ?(Why do Information Security Vulnerabilities exist?)

ಕೋಡ್‌ನಲ್ಲಿನ ಅಥವಾ ನೆಟ್‌ವರ್ಕ ಪ್ರೊಟೊಕಾಲ್‌ಗಳ ದುರ್ಬಳಕೆಗಳಿಂದಾಗಿ ಅಸ್ತಿತ್ವದಲ್ಲಿವೆ. ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ್ನು ತಯಾರಿಸಲು ಮಿಲಿಯನ್‌ಗಟ್ಟಲೆ ಕೋಡ್‌ಗಳ ಅಗತ್ಯವಿರುತ್ತದೆ ಮತ್ತು ಕೆಲವೊಮ್ಮೆ ದುರ್ಬಲತೆಗಳು ಅವುಗಳಲ್ಲಿ ಕಾಣಬಹುದು. ಒಂದು ವೇಳೆ ಫೈರ್‌ವಾಲ್ ಹಲವಾರು ಪೋರ್ಟ್‌ಗಳನ್ನು ತೆರೆದಿದ್ದರೆ, ಅದರಲ್ಲಿ ಒಂದನ್ನು ಬಳಸಿ ಒಳನುಗ್ಗುವವರು ಅನಧಿಕೃತ ವಿಧಾನದಲ್ಲಿ ನೆಟ್‌ವರ್ಕ್ ಅನ್ನು ಪ್ರವೇಶಿಸುವ ಸಾಧ್ಯತೆ ಹೆಚ್ಚಾಗಿರುತ್ತದೆ. ದುರ್ಬಲತೆಯು, ಹಾರ್ಡ್‌ವೇರ್‌ನಲ್ಲಿ, ಸಾಫ್ಟ್‌ವೇರ್‌ನಲ್ಲಿ ಅಥವಾ ಪ್ರಕ್ರಿಯೆಯಲ್ಲಿರುವ ಒಂದು ನ್ಯೂನ್ಯತೆ ಅಥವಾ ದೌರ್ಬಲ್ಯದ ಪರಿಣಾಮವಾಗಿರಬಹುದು. ಇದು ಸಿಸ್ಟಮ್‌ನ್ನು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳುವಂತೆ ಮಾಡುತ್ತದೆ. ಇದು ದೌರ್ಬಲ್ಯ, ವಿನ್ಯಾಸ ಅಥವಾ ಅನುಷ್ಠಾನ ದೋಷವಾಗಿದೆ. ಇದು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್, ನೆಟ್‌ವರ್ಕ್ , ಅಪ್ಲಿಕೇಷನ್, ಅಥವಾ ಪ್ರೊಟೊಕಾಲನ್ನು ಮುಂತಾದವುಗಳ ರಾಜಿ ಮಾಡಿಕೊಳ್ಳುವ ಅನಿರೀಕ್ಷಿತ, ಅನಪೇಕ್ಷಿತ ಘಟನೆಗೆ ಕಾರಣವಾಗಬಲ್ಲದು.

©NIMI NOT TO BE REPUBLISHED

A ತಾಂತ್ರಿಕ ದೋಷಗಳ ವಿಧಗಳು (Types of Technical Vulnerabilities)

ಹೆಚ್ಚಿನ ಸಾಫ್ಟ್‌ವೇರ್ ಭದ್ರತಾ ದೋಷಗಳು ಈ ಕೆಳಗಿನವುಗಳಲ್ಲಿರುವ ಒಂದು ಸಣ್ಣ ವರ್ಗಗಳಲ್ಲಿ ಬರುತ್ತದೆ :

- ಬಫರ್ ಓವರ್‌ಫ್ಲೋ
- ಮೌಲ್ಯೀಕರಿಸದ ಇನ್ಪುಟ್
- ರೇಸ್ ಕಂಡಿಷನ್‌ಗಳು
- ಪ್ರವೇಶ - ನಿಯಂತ್ರಣ ಸಮಸ್ಯೆಗಳು
- ದೃಢೀಕರಣ ದೌರ್ಬಲ್ಯಗಳು, ಅಧಿಕಾರ, ಅಥವಾ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಕ್ ಅಭ್ಯಾಸಗಳು.

ಬಫರ್ ಓವರ್‌ಫ್ಲೋ (Buffer Overflows)

ಬಫರ್ ಓವರ್‌ಫ್ಲೋ- ಇದನ್ನು ದೋಷಗಳ ಪ್ರಮುಖ ಮೂಲವೆಂದು ಪರಿಗಣಿಸಲಾಗಿದೆ. ಒಂದು ಅಪ್ಲಿಕೇಷನ್ ಬಫರ್‌ನ ಅಂತ್ಯವನ್ನು (ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ಆರಂಭದ ಮೊದಲು) ಮೀರಿ ಡಾಟಾವನ್ನು ಬರೆಯಲು ಪ್ರಯತ್ನಿಸಿದಾಗ ಇದು ಸಂಭವಿಸುತ್ತದೆ. ಬಫರ್ ಓವರ್‌ಫ್ಲೋಗಳು ಅಪ್ಲಿಕೇಷನ್ ಕ್ರ್ಯಾಶ್‌ಗೆ ಕಾರಣವಾಗಬಹುದು, ಡಾಟಾವನ್ನು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳಬಹುದು, ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ ರನ್ ಆಗುತ್ತಿರುವ ಸಿಸ್ಟಮ್‌ನ್ನು ಹೆಚ್ಚಿನ ಸವಲತ್ತುಗಳಿಗಾಗಿ ರಾಜಿ ಮಾಡಿಕೊಳ್ಳಲು ವೆಕ್ಟರ್ ದಾಳಿಯನ್ನು ಮಾಡಬಹುದು. ಬಫರ್ ಓವರ್‌ಫ್ಲೋ ದಾಳಿಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಟ್ಯಾಕ್‌ನ್ನು (stack) , ಹೀಪ್‌ನ್ನು (heap) ಅಥವಾ ಎರಡನ್ನೂ ರಾಜಿ ಮಾಡಿಕೊಳ್ಳುವ ಮೂಲಕ ಸಂಭವಿಸುತ್ತವೆ.

ಮೌಲ್ಯೀಕರಿಸದ ಇನ್ಪುಟ್ (Unvalidated Input)

ಸಾಮಾನ್ಯನಿಯಮದಂತೆ, ನಿಮ್ಮ ಪ್ರೊಗ್ರಾಂ ಸ್ವೀಕರಿಸಿದ ಎಲ್ಲಾ ಇನ್ಪುಟ್ ಅನ್ನು ನೀವು ಪರಿಶೀಲಿಸಬೇಕು ಮತ್ತು ಡಾಟಾ ಸಮಂಜಸವಾಗಿದೆ ಅಥವಾ ಮಾನ್ಯವಾಗಿದೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಬೇಕು. ಅಮಾನ್ಯ ಡಾಟಾವನ್ನು ಅನುಮತಿಸಿದ ಅಥವಾ ಸ್ವೀಕರಿಸಿದ ಸಂದರ್ಭಗಳಲ್ಲಿ ಇಂತಹ ಫೈಲನ್ನು ಒಂದು ಸಾಮಾನ್ಯ ಪ್ರೊಗ್ರಾಂ ಓದಲು ಪ್ರಯತ್ನಿಸುವ ಸಂಭಾವ್ಯತೆಯಲ್ಲಿ ತಪ್ಪಾದ ಗಾತ್ರದ ಬಫರ್‌ಅನ್ನು ನಿಯೋಜಿಸಲು ಪ್ರಯತ್ನಿಸಲಾಗುತ್ತದೆ. ಇದು ಹೀಪ್ ಓವರ್‌ಫ್ಲೋ ದಾಳಿಗೆ ಅಥವಾ ಇತರ ಸಮಸ್ಯೆಗೆ ಕಾರಣವಾಗುತ್ತದೆ. ಈ ಕಾರಣಕ್ಕಾಗಿ ನೀವು ನಿಮ್ಮ ಇನ್ಪುಟ್ ಡಾಟಾವನ್ನು ಎಚ್ಚರಿಕೆಯಿಂದ ಪರಿಶೀಲಿಸಬೇಕು. ಸಾಮಾನ್ಯವಾಗಿ ಈ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಇನ್ಪುಟ್ ಮೌಲ್ಯೀಕರಣ (input validation)ಅಥವಾ ವಿವೇಕ ತಪಾಸಣೆ (sanity checking) ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ವಿಶ್ವಾಸಾರ್ಹವಲ್ಲದ ಮೂಲದಿಂದ ನಿಮ್ಮ ಪ್ರೊಗ್ರಾಂ ಸ್ವೀಕರಿಸಿದ ಯಾವುದೇ ಇನ್ಪುಟ್, ದಾಳಿಯ ಸಂಭಾವ್ಯ ಗುರಿಯಾಗಿರುತ್ತದೆ. ಹ್ಯಾಕರ್‌ಗಳು ನಿಮ್ಮ ಪ್ರೊಗ್ರಾಂಗೆ ನೀಡಲಾದ ಪ್ರತಿಯೊಂದು ಇನ್ಪುಟ್ ಅನ್ನು ನೋಡುತ್ತಾರೆ ಮತ್ತು ಅವರು ಊಹಿಸಬಹುದಾದ ಪ್ರತಿಯೊಂದು ಪ್ರಕಾರದ ದೋಷಪೂರಿತ ಡಾಟಾವನ್ನು ಪಾಸ್ ಮಾಡಲು ಪ್ರಯತ್ನಿಸುತ್ತಾರೆ. ಒಂದು ವೇಳೆ ಪ್ರೊಗ್ರಾಂ ಕ್ರ್ಯಾಶ್ ಆದರೆ ಅಥವಾ ತಪ್ಪಾಗಿ ವರ್ತಿಸಿದರೆ, ಹ್ಯಾಕರ್ ಸಮಸ್ಯೆಯನ್ನು ಬಳಸಿಕೊಳ್ಳುವ ಮಾರ್ಗವನ್ನು ಕಂಡುಹಿಡಿಯಲು ಪ್ರಯತ್ನಿಸುತ್ತಾನೆ.

ರೇಸ್ ಪರಿಸ್ಥಿತಿಗಳು (Race Conditions)

ರೇಸ್ ಪರಿಸ್ಥಿತಿ ಯಾವಾಗ ಸಂಭವಿಸುತ್ತದೆ ಎಂದರೆ, ನಿಯಮಗಳ ಪ್ರಕಾರ ಉದ್ದೇಶಿಸಲಾದ ರೀತಿಯಲ್ಲಿ ಅಪ್ಲಿಕೇಶನ್‌ನಲ್ಲಿನ ರೂಟೀನ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಕಾಲ್‌ಗಳು ಅನುಕ್ರಮದಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸದೇ ಇದ್ದಾಗ. ಇದು ಸಾಫ್ಟ್‌ವೇರ್‌ನಲ್ಲಿನ ಒಂದು ಟೈಮಿಂಗ್ ಇವೆಂಟ್ ಆಗಿದೆ. ಕಾಲ್‌ಗಳನ್ನು ಸರಿಯಾದ ಕ್ರಮದಲ್ಲಿ ನಿರ್ವಹಿಸದೇ ಇದ್ದಾಗ ಭದ್ರತಾ ದೋಷವಾಗಬಹುದು. ಪ್ರೊಗ್ರಾಂನ ಸರಿಯಾದ ಕಾರ್ಯನಿರ್ವಹಣೆಗೆ ಸರಿಯಾದ ಕ್ರಮದ ಅವಶ್ಯಕತೆ ಇದ್ದರೆ, ಇದು ಒಂದು ದೋಷವಾಗಿದೆ(bug). ಒಂದು ವೇಳೆ ಆಕ್ರಮಣಕಾರರು ದುರುದ್ದೇಶಪೂರಿತ ಕೋಡನ್ನು ಸೇರಿಸಲು, ಫೈಲ್‌ನ ಹೆಸರನ್ನು ಬದಲಾಯಿಸಲು, ಅಥವಾ ಪ್ರೊಗ್ರಾಂನ ಸರಿಯಾದ ಕಾರ್ಯವಿಧಾನದ ಮಧ್ಯ ಪ್ರವೇಶಿಸುವುದು ಮುಂತಾದವುಗಳಿಂದ ಪರಿಸ್ಥಿತಿಯ ಲಾಭವನ್ನು ಪಡೆಯಲು ಪ್ರಯತ್ನಿಸಿದರೆ, ಇದು ರೇಸ್ ಪರಿಸ್ಥಿತಿಯ ಭದ್ರತಾ ದೋಷವಾಗಿದೆ. ಆಕ್ರಮಣಕಾರರು ಕೆಲವೊಮ್ಮೆ ಕೋಡ್‌ನ ಪ್ರಕ್ರಿಯೆಯ ಮಧ್ಯದಲ್ಲಿರುವ ಸಣ್ಣ ಸಮಯದ ಅಂತರದ ಲಾಭವನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು ಈ ರೀತಿ ಕಾರ್ಯಾಚರಣೆಗಳ ಮೂಲಕ ಹಸ್ತಕ್ಷೇಪ ಮಾಡಿ ಶೋಷಣೆ ಮಾಡುತ್ತಾರೆ.

ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿಯಲ್ಲಿ time of check to time of use (TOCTTOU ಅಥವಾ TOCTOU, "TOCK too" ಎಂದು ಉಚ್ಚರಿಸಲಾಗುತ್ತದೆ)ಇದು ಸಾಫ್ಟ್‌ವೇರ್ ಬಗ್‌ನ ಕ್ಲಾಸ್ ಆಗಿದೆ. ಇದು ಕಂಡಿಷನ್‌ನ್ನು (ಉದಾ ಭದ್ರತೆಯ ರುಜುವಾತು) ತಪಾಸಣೆ ಮಾಡುವುದರಿಂದ ಮತ್ತು ಫಲಿತಾಂಶಗಳ ಬಳಕೆಗಳ ನಡುವೆ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಆಗುವ ಬದಲಾವಣೆಗಳಿಂದ ಉಂಟಾಗುತ್ತದೆ. ಇದು ರೇಸ್ ಕಂಡೀಷನ್‌ನ ಒಂದು ಉದಾಹರಣೆಯಾಗಿದೆ.

Interprocess Communication(IPC)

Interprocess Communication (IPC) ಇದು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಇಂಟರ್‌ಫೇಸ್‌ಗಳ ಒಂದು ಗುಂಪಾಗಿದೆ. ಇದು ಪ್ರೊಗ್ರಾಮರ್‌ಗೆ ವಿವಿಧ ಪ್ರೊಗ್ರಾಂ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಏಕಕಾಲದಲ್ಲಿ ರನ್ ಆಗುವಂತೆ ಚಟುವಟಿಕೆಗಳನ್ನು ಸಂಘಟಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದು ಪ್ರೊಗ್ರಾಂಗೆ ಒಂದೇ ಸಮಯದಲ್ಲಿ ಅನೇಕ ಬಳಕೆದಾರರ ವಿನಂತಿಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.ಈ ಸಂದೇಶದ ಪ್ರೊಟೊಕಾಲ್‌ಗಳನ್ನು ಇಂಟರ್‌ಪ್ರೊಸೆಸ್ ಸಂವಹನಕ್ಕಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

Remote Procedure Call (RPC)

Remote Procedure Call (RPC) ಇದು ಇಂಟರ್‌ಪ್ರೊಸೆಸ್ ಸಂವಹನದ ಕಾರ್ಯತಂತ್ರವಾಗಿದೆ. ಇದು ಒಂದು ಹೋಸ್ಟ್ ನಲ್ಲಿ ರನ್ ಆಗುತ್ತಿರುವ ಪ್ರೊಗ್ರಾಂಗೆ ರಿಮೋಟ್ ಹೊಸ್ಟ್‌ನಲ್ಲಿರುವ ಕೋಡ್‌ನ್ನು ರನ್ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ

ಅಸುರಕ್ಷಿತ ಫೈಲ್ ಕಾರ್ಯಾಚರಣೆಗಳು (Insecure File Operations)

ಸಮಯ -ಪರಿಶೀಲನೆ -ಸಮಯ - ಬಳಕೆ (time-of-check-time-of-use) ಸಮಸ್ಯೆಯ ಜೊತೆಗೆ , ಇನ್ನೂ ಅನೇಕ ಫೈಲ್ ಕಾರ್ಯಾಚರಣೆಗಳು ಅಸುರಕ್ಷಿತವಾಗಿವೆ. ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಹೆಚ್ಚಾಗಿ ಫೈಲ್‌ನ ಮಾಲೀಕತ್ವ, ಸ್ಥಳ, ಅಥವಾ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳು ಸರಿ ಇಲ್ಲದಿರುವ ಬಗ್ಗೆ ಊಹೆ ಮಾಡುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ ಯಾವಾಗಲೂ ನಿಮ್ಮ ಪ್ರೊಗ್ರಾಂನಿಂದ ರಚಿಸಲ್ಪಟ್ಟ ಫೈಲನ್ನು ನೀವು ಬರೆಯಬಹುದು ಎಂದು ಊಹಿಸಿಕೊಳ್ಳಬಹುದು. ಅದಾಗ್ಯೂ ನೀವು ರಚಿಸಿದ ಮೇಲೆ ಆ ಫೈಲ್‌ನ ಅನುಮತಿಗಳನ್ನು ಅಥವಾ ಫ್ಲ್ಯಾಗ್‌ಗಳನ್ನು ಆಕ್ರಮಣಕಾರರು ಬದಲಾಯಿಸಿದರೆ ಮತ್ತು write ಆಪರೇಷನ್ ನಂತರ ನೀವು ಫಲಿತಾಂಶದ ಕೋಡ್‌ನ್ನು ಪರಿಶೀಲಿಸಲು ವಿಫಲರಾದರೆ, ನೀವು ಫೈಲನ್ನು ಟ್ಯಾಂಪರ್(ಬದಲಾಯಿಸು) ಮಾಡಲಾಗಿದೆ ಎಂಬುವ ಅಂಶವನ್ನು ಪತ್ತೆಹಚ್ಚಲಾಗುವುದಿಲ್ಲ.

B **ಸ್ಥಳೀಯ ದುರ್ಬಲತೆಯ ವಿಧಗಳು** (Types of Native Vulnerabilities)

ಸ್ಥಳೀಯ ದುರ್ಬಲತೆಯ ಉದಾಹರಣೆಗಳು ಯಾವವೆಂದರೆ :

- ಸ್ಯಾಂಡ್ ಬಾಕ್ಸಿಂಗ್ ಕಾರ್ಯತಂತ್ರದಲ್ಲಿನ ದೋಷಗಳು ವಿಶ್ವಾಸಾರ್ಹವಲ್ಲದ ಬೈಟ್‌ಕೋಡನ್ನು ಅನುಮತಿಸುತ್ತವೆ. ಇದು ಭದ್ರತಾ ವ್ಯವಸ್ಥಾಪಕರು ವಿಧಿಸಿದ ನಿರ್ಬಂಧಗಳನ್ನು ತಪ್ಪಿಸಿಕೊಳ್ಳುತ್ತವೆ.
- ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ತಮ್ಮ ಸುರಕ್ಷತೆಗಾಗಿ ಅವಲಂಬಿಸಿರುವ ಜಾವಾ ಕ್ಲಾಸ್ ಲೈಬ್ರರಿಯಲ್ಲಿರುವ ದೋಷಗಳು

ಭದ್ರತಾ ದೋಷಗಳನ್ನು ತಿಳಿದುಕೊಳ್ಳುವುದು (Understanding Security Vulnerabilities)

ಸಾಫ್ಟವೇರ್ ಅಥವಾ ಪ್ರೊಟೊಕಾಲ್ ವಿನ್ಯಾಸಗಳಲ್ಲಿನ ದೋಷಗಳು (Flaws in Software or Protocol Designs)

ಸಾಫ್ಟವೇರ್ ವಿನ್ಯಾಸದಲ್ಲಿನ ಮೂಲಭೂತ ತಪ್ಪುಗಳು ಮತ್ತು ಅಜಾಗರೂಕತೆಗಳು ವಿನ್ಯಾಸ ದೋಷಗಳಿಗೆ ಕಾರಣಗಳಾಗಿವೆ. ವಿನ್ಯಾಸದಲ್ಲಿನ ನ್ಯೂನ್ಯತೆಗಳು ಅಸುರಕ್ಷಿತ ಸಾಫ್ಟವೇರನ್ನು ತಯಾರಿಸುತ್ತವೆ ಮತ್ತು ಅದು ಉನ್ನತ ಮಟ್ಟದ ದುರ್ಬಲತೆಯ ಪ್ರಕರಣವಾಗುತ್ತದೆ.

ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳು ಪ್ರೊಟೊಕಾಲ್‌ಗಳ ಮೇಲೆ ಅವಲಂಬಿತವಾಗಿವೆ. ಈ ಪ್ರೊಟೊಕಾಲ್‌ಗಳು ರನ್‌ಟೈಮ್‌ನಲ್ಲಿ ವಿನಿಮಯವಾದ ಸಂದೇಶಗಳು, ಅವುಗಳ ಸ್ವರೂಪ, ಮತ್ತು ರಚನೆಯನ್ನು ಸೂಚಿಸುತ್ತವೆ.

ಪ್ರೊಟೊಕಾಲ್‌ಗಳು ವಿಭಿನ್ನ ಪ್ರೊಟೊಕಾಲ್ ಸ್ಟ್ಯಾಕ್‌ಗಳೊಂದಿಗೆ ಲಿಂಕ್ ಆಗಿವೆ. ಉದಾಹರಣೆಗೆ , TCP/IP, ಅಥವಾ ವಿವಿಧ ಮಾಡೆಲ್‌ಗಳು, ಉದಾಹರಣೆಗೆ OSI ಮತ್ತು ಹಲವಾರು ಅಂಡರ್‌ಸ್ಪೆಸಿಫೈಡ್ ಸುರಕ್ಷತೆ ಹೊಂದಿರುವ ಪ್ರೊಟೊಕಾಲ್‌ಗಳು ಇನ್ನೂ ಉಪಯೋಗದಲ್ಲಿವೆ. ದೋಷಪೂರಿತ ಪ್ರೊಟೊಕಾಲ್‌ಗಳಿಂದಾಗಿ ಉಂಟಾಗುವ ಕೆಲವು ದೋಷಗಳನ್ನು ಕೆಳಗೆ ವಿವರಿಸಲಾಗಿದೆ:

A TCP/IP. TCP/IP ಪ್ರೊಟೊಕಾಲ್ ಸ್ಟ್ಯಾಕ್ ಕೆಲವು ದುರ್ಬಲತೆಗಳನ್ನು ಹೊಂದಿದೆ. ಇವು ಕೆಳಗಿನವುಗಳನ್ನು ಅನುಮತಿಸುತ್ತವೆ. :

- **ವಂಚನೆ** (Spoofing) : ದುರುದ್ದೇಶಪೂರಿತ ಪಕ್ಷವು ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಇನ್ನೊಂದು ಸಾಧನ ಅಥವಾ ಬಳಕೆದಾರರ ಸೋಗಿನಲ್ಲಿ ವಂಚನೆ ಮಾಡುವ ದಾಳಿಯಾಗಿದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್ ಹೋಸ್ಟ್ ವಿರುದ್ಧ ದಾಳಿ ನಡೆಸಲು, ಡಾಟಾ ಕದಿಯಲು, ಮಾಲ್ವೇರ್ ಹರಡಲು ಅಥವಾ ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಬೈಪಾಸ್ ಮಾಡಲು ಕೆಲಸ ಮಾಡುತ್ತದೆ. ಹಲವಾರು ರೀತಿಯ ವಿವಿಧ ವಂಚನೆಯ ದಾಳಿಗಳಿವೆ. ಇವುಗಳನ್ನು ದುರುದ್ದೇಶಪೂರಿತ ಪಕ್ಷಗಳು ವಂಚನೆಯನ್ನು ಸಾಧಿಸಲು ಉಪಯೋಗಿಸಬಹುದು. ಅವು : IP Address Spoofing Attacks, ARP (Address Resolution Protocol) Spoofing Attacks, DNS Server Spoofing Attacks, ಇತ್ಯಾದಿಗಳು.
- **Telnet ಪ್ರೊಟೊಕಾಲ್ :** ಮೈಕ್ರೊಸಾಫ್ಟ್ ವಿಂಡೋಸ್ 2000 ಮತ್ತು ಯುನಿಕ್ಸ್‌ಗಳನ್ನು ರನ್ ಮಾಡುವ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಟೆಲ್‌ನೆಟ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಟೆಲ್‌ನೆಟ್ ಕ್ಲೈಂಟ್‌ನ್ನು ಮೈಕ್ರೊಸಾಫ್ಟ್ ಸಿಸ್ಟಮ್‌ನಿಂದ ಯುನಿಕ್ಸ್‌ಗೆ ಅಥವಾ ಅದಲುಬದಲಾಗಿ ಸಂಪರ್ಕಿಸಲು ಬಳಸುವಾಗ, ಬಳಕೆದಾರರ ಹೆಸರುಗಳನ್ನು ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಸ್ಪಷ್ಟವಾದ ಟೆಕ್ಸ್ಟ್ ನಲ್ಲಿ ರವಾನಿಸಲಾಗುತ್ತದೆ. ಹೀಗೆ ಭದ್ರತಾ ದುರ್ಬಲತೆಯನ್ನು ಸೃಷ್ಟಿಸುತ್ತದೆ.
- **File Transfer Protocol (FTP)** : File Transfer Protocol ಬಳಕೆದಾರರನ್ನು ರಿಮೋಟ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಸಂಪರ್ಕಿಸಲು ಮತ್ತು ಫೈಲ್‌ಗಳನ್ನು ಹಿಂದಕ್ಕೆ ಮತ್ತು ಮುಂದಕ್ಕೆ ವರ್ಗಾವಣೆ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ. ರಿಮೋಟ್ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಸಂಪರ್ಕವನ್ನು ಸ್ಥಾಪಿಸುವ ಭಾಗವಾಗಿ, ದೃಢೀಕರಣಕ್ಕಾಗಿ FTP ಬಳಕೆದಾರರ ಹೆಸರನ್ನು ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ನ

© NIMI NOT TO BE REPUBLISHED

ಸಂಯೋಜನೆಯನ್ನು ಅವಲಂಬಿಸಿದೆ. FTP ಬಳಕೆಯು ಟೆಲ್‌ನೆಟ್ ಪ್ರೊಟೊಕಾಲ್‌ನ ಬಳಕೆಯಂತೆಯೆ ಹೋಲುವ ಭದ್ರತಾ ಸಮಸ್ಯೆಯನ್ನು ಉಂಟುಮಾಡುತ್ತದೆ. ಏಕೆಂದರೆ FTP ಗೆ ಟೈಪ್ ಮಾಡಲಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ಸರಳ ಟೆಕ್ಸ್ಟ್ ನಲ್ಲಿ ಎಂದರೆ ಪ್ರತಿ ಪ್ಯಾಕೆಟ್‌ಗೆ ಒಂದು ಅಕ್ಷರದಂತೆ, ನೆಟ್‌ವರ್ಕ್ ಮೂಲಕ ರವಾನಿಸಲ್ಪಡುತ್ತವೆ. ಈ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ತಡೆಹಿಡಿಯಬಹುದು.

ಪ್ರೊಟೊಕಾಲ್‌ಗಳಲ್ಲಿ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್‌ನಲ್ಲಿ ದೌರ್ಬಲ್ಯಗಳನ್ನು ಹೇಗೆ ಅಳವಡಿಸುವುದು.(Weaknesses in how protocols and software are implemented)

ಪ್ರೊಟೊಕಾಲನ್ನು ಉತ್ತಮವಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಿದ್ದರೂ ಸಹ ಅದು ದುರ್ಬಲವಾಗಿರಬಹುದು ಏಕೆಂದರೆ ಅದನ್ನು ಅಳವಡಿಸುವ ವಿಧಾನದಿಂದಾಗಿ. ಉದಾಹರಣೆಗೆ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಮೇಲ್‌ಗಾಗಿ ಪ್ರೊಟೋಕಾಲನ್ನು ಈ ರೀತಿಯಾಗಿ ಅಳವಡಿಸುವುದರಿಂದ ಇದು ಒಳನುಗ್ಗುವವರಿಗೆ ಬಲಿಪಶುವಿನ ಮಶೀನ್‌ನ ಮೇಲ್ ಪೋರ್ಟ್‌ಗೆ ಸಂಪರ್ಕಿಸುತ್ತದೆ ಮತ್ತು ಆ ಮಶೀನ್‌ನ್ನು ಮರುಳುಮಾಡಿ ಸೇವೆಯಿಂದ ಉದ್ದೇಶಿಸದ ಕಾರ್ಯವನ್ನು ಮಾಡಿಸುತ್ತದೆ. ಈ ರೀತಿಯ ದುರ್ಬಲತೆಯು ಒಳನುಗ್ಗುವವರನ್ನು ಬಲಿಪಶುವಿನ ಮಶೀನ್ ಮೇಲೆ ರಿಮೋಟ್ ಸೈಟ್‌ನಿಂದ ದಾಳಿ ಮಾಡಲು ಶಕ್ತಗೊಳಿಸುತ್ತದೆ.

- ದೋಷಗಳ ಕಾರಣದಿಂದಾಗಿ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು ದುರ್ಬಲವಾಗಿರಬಹುದು ಏಕೆಂದರೆ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಬಿಡುಗಡೆ ಮಾಡುವ ಮೊದಲು ದೋಷಗಳನ್ನು ಗುರುತಿಸಲಾಗದ ಕಾರಣದಿಂದಾಗಿ. ಈ ರೀತಿಯ ದುರ್ಬಲತೆಯು ವ್ಯಾಪಕ ಶ್ರೇಣಿಯ ಸಬ್ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಒಳನುಗ್ಗುವವರು ಸಾಮಾನ್ಯವಾಗಿ ತಮ್ಮದೇ ಆದ ದಾಳಿಯ ಸಾಧನಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ ಉದಾಹರಣೆಗೆ ಫೈಲ್ ಪ್ರವೇಶದಲ್ಲಿ ರೇಸ್ ಕಂಡಿಷನ್‌ಗಳು, ಅಸ್ತಿತ್ವದಲ್ಲಿಲ್ಲದ ಡಾಟಾ ವಿಷಯವನ್ನು ಮತ್ತು ಗಾತ್ರವನ್ನು ಪರಿಶೀಲಿಸುವುದು ಇತ್ಯಾದಿಗಳು.

ಸಿಸ್ಟಮ್ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಕಾನ್ಫಿಗರೇಷನ್‌ನಲ್ಲಿನ ದೌರ್ಬಲ್ಯಗಳು (Weaknesses in System and Network Configurations)

- ಸಿಸ್ಟಮ್ ನಿರ್ವಾಹಕರು ಮತ್ತು ಬಳಕೆದಾರರು ನೆಟ್‌ವರ್ಕ್ ಕಾನ್ಫಿಗರೇಷನ್‌ಗಳಲ್ಲಿನ ಡೀಫಾಲ್ಟ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಬದಲಾಯಿಸುವದನ್ನು ನಿರ್ಲಕ್ಷಿಸಿರಬಹುದು ಅಥವಾ ಅವರು ತಮ್ಮ ವ್ಯವಸ್ಥೆಯನ್ನು ಸರಳವಾಗಿ ಹೊಂದಿಸಿ ನೆಟ್‌ವರ್ಕ್ ಅನ್ನು ದುರ್ಬಲಗೊಳಿಸುವ ರೀತಿಯಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸಿರಬಹುದು.
- Asynchronous transfer mode (ATM) : ಭದ್ರತೆಯನ್ನು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳಬಹುದು. "manhole manipulation" ಮೂಲಕ ಭೂಗತ ಪಾರ್ಕಿಂಗ್ ಗ್ಯಾರೇಜುಗಳಲ್ಲಿ ಮತ್ತು ಎಲಿವೇಟರ್ ಶಾಫ್ಟ್‌ಗಳಲ್ಲಿ ಇರುವ ನೆಟ್‌ವರ್ಕ್ ಕೇಬಲ್‌ಗಳು ಮತ್ತು ಸಂಪರ್ಕಗಳಿಗೆ ನೇರ ಪ್ರವೇಶವನ್ನು ಒದಗಿಸುತ್ತದೆ.
- Frame relay: ATM ಸಮಸ್ಯೆಯಂತೆಯೆ ಇರುತ್ತದೆ.
- **ಸಾಧನ ನಿರ್ವಹಣೆ :** ಸ್ವಿಚ್‌ಗಳು ಮತ್ತು ರೂಟರ್‌ಗಳು, HTTP ಇಂಟರ್‌ಫೇಸ್‌ನಿಂದ ಅಥವಾ ಕಮಾಂಡ್ ಲೈನ್ ಇಂಟರ್‌ಫೇಸ್‌ನಿಂದ ಸುಲಭವಾಗಿ ನಿರ್ವಹಿಸಲ್ಪಡುತ್ತವೆ. ದುರ್ಬಲ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು (ಉದಾಹರಣೆ ಸಾರ್ವಜನಿಕ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು) ಉಪಯೋಗಿಸಿ ಇದು ಯಾರಿಗಾದರೂ ಕೆಲವು ತಾಂತ್ರಿಕ ಜ್ಞಾನವನ್ನು ಬಳಸಿ ಸಾಧನದ ನಿಯಂತ್ರಣವನ್ನು ತೆಗೆದುಕೊಳ್ಳಲು ಅನುಮತಿಸುತ್ತದೆ.
- **ಮಾಡೆಮ್‌ಗಳು :** ಒಂದು ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಹೊರಗಿನ ದಾಳಿಕೋರರಿಂದ ರಕ್ಷಿಸುವ ಫೈರ್‌ವಾಲನ್ನು ಮಾಡೆಮ್ ಬೈಪಾಸ್ ಮಾಡುತ್ತದೆ. ಹ್ಯಾಕರ್‌ಗಳು "war dialer" ಸಾಧನವನ್ನು ಬಳಸಿ ಮಾಡೆಮ್‌ನ ಟೆಲಿಫೋನ್ ನಂಬರ್‌ನ್ನು ಗುರುತಿಸುತ್ತಾರೆ ಮತ್ತು "password cracker" ಸಾಧನವನ್ನು ಬಳಸಿ ದುರ್ಬಲ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಗುರುತಿಸುತ್ತಾರೆ ಮತ್ತು ಸಿಸ್ಟಮ್‌ಗೆ ಪ್ರವೇಶವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾರೆ.
- ವೆಬ್ ಮತ್ತು ಕ್ಲೌಡ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿನ ದೌರ್ಬಲ್ಯಗಳು.

ಹಲವಾರು ಗಮನಾರ್ಹವಾದ ದುರ್ಬಲತೆಗಳಿವೆ. ಒಂದು ಸಂಸ್ಥೆ ತನ್ನ ನಿರ್ಣಾಯಕ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಡಾಟಾವನ್ನು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಪರಿಸರಕ್ಕೆ ವರ್ಗಾಯಿಸಲು ಸಿದ್ಧವಾದಾಗ ಈ ದುರ್ಬಲತೆಗಳನ್ನು ಪರಿಗಣಿಸಬೇಕಾಗುತ್ತದೆ. ಈ ದುರ್ಬಲತೆಗಳನ್ನು ಕೆಳಗೆ ವಿವರಿಸಲಾಗಿದೆ :

A **ಸೆಷನ್ ರೈಡಿಂಗ್ ಮತ್ತು ಹೈಜಾಕಿಂಗ್** (Session Riding and Hi-jacking):

ಸೆಷನ್ ರೈಡಿಂಗ್ ಎಂದರೆ, ಹ್ಯಾಕರ್‌ಗಳು ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗೆ ಕಮಾಂಡ್‌ಗಳನ್ನು ಕಳುಹಿಸುತ್ತಾರೆ. ಉದ್ದೇಶಿತ ಬಳಕೆದಾರರ ಪರವಾಗಿ ಅವರಿಗೆ ಇಮೇಲ್ ಕಳುಹಿಸುವುದರ ಮೂಲಕ ಅಥವಾ ವಿಶೇಷವಾಗಿ ರಚಿಸಲಾದ ವೆಬ್‌ಸೈಟ್‌ಗಳಿಗೆ ಬಳಕೆದಾರರನ್ನು ಭೇಟಿಮಾಡುವಂತೆ ಮಾಡಿ ಮೋಸಗೊಳಿಸುತ್ತಾರೆ. ಸೆಷನ್ ರೈಡಿಂಗ್, ಬಳಕೆದಾರರ ಡಾಟಾವನ್ನು ಅಳಿಸುತ್ತದೆ, ಬಿಡ್‌ಗಳು ಅಥವಾ ಆರ್ಡರ್‌ಗಳಂತಹ ಆನ್‌ಲೈನ್ ವಹಿವಾಟುಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ, ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಇಂಟ್ರಾನೆಟ್‌ಗೆ ಸ್ಪ್ಯಾಮ್‌ಗಳನ್ನು ಕಳುಹಿಸುತ್ತದೆ, ಸಿಸ್ಟಮ್ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಂರಚನೆಗಳನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ, ಹಾಗೂ ಫೈರ್‌ವಾಲನ್ನೂ ಸಹ ತೆರೆಯುತ್ತದೆ.

B **ವರ್ಚುವಲ್ ಮಶೀನ್ ಎಸ್ಕೇಪ್** (Virtual Machine Escape) : VM ಎಸ್ಕೇಪ್ ದುರ್ಬಲತೆಯು ಗೆಸ್ಟ್ - ಲೆವೆಲ್ VM ತನ್ನ ಹೋಸ್ಟ್ ಮೇಲೆ ದಾಳಿ ಮಾಡುವಂತೆ ಮಾಡುವ ದುರ್ಬಲತೆಯಾಗಿದೆ. ಈ ದುರ್ಬಲತೆಯ ಅಡಿಯಲ್ಲಿ ದಾಳಿಕೋರರು VM ನಲ್ಲಿ ಕೋಡ್‌ನ್ನು

© NIMI NOT TO BE REPUBLISHED

ರನ್ ಮಾಡುತ್ತಾರೆ . ಈ ಕೋಡ್ ಅದರಲ್ಲಿ ರನ್ ಆಗುತ್ತಿರುವ OS ನ್ನು ಭೇದಿಸುತ್ತದೆ ಮತ್ತು ನೇರವಾಗಿ ಹೈಪರ್‌ವೈಸರ್‌ನೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸುತ್ತದೆ.

C **ಸೇವೆಯ ವಿಶ್ವಾಸಾರ್ಹತೆ ಮತ್ತು ಲಭ್ಯತೆ** (Reliability and Availability of Service) :

ಕ್ಲೌಡ್ ಸ್ಟೋರೇಜ್ ಮೂಲಸೌಕರ್ಯವು ಗಣನೀಯ ಸಮಯದವರೆಗೆ ಕಡಿಮೆಯಾಗಬಹುದು. ಇದು ವೆಬ್ ಸೇವೆಗಳಲ್ಲಿನ ಡಾಟಾ ನಷ್ಟವನ್ನು ಮತ್ತು ಪ್ರವೇಶ ಸಮಸ್ಯೆಯನ್ನು ಉಂಟು ಮಾಡುತ್ತದೆ.

D **ಅಸುರಕ್ಷಿತ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿ** (Insecure Cryptography) :

ಆಕ್ರಮಣಕಾರರು ಯಾವುದೇ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಕ್ ಕಾರ್ಯವಿಧಾನವನ್ನು ಅಥವಾ ಅಲ್ಗೋರಿದಮ್‌ನ್ನು ಡಿಕೋಡ್ ಮಾಡಬಹುದು. ಏಕೆಂದರೆ ಅವುಗಳನ್ನು ಹ್ಯಾಕ್ ಮಾಡಲು ಮುಖ್ಯ ವಿಧಾನಗಳನ್ನು ಕಂಡುಹಿಡಿಯಲಾಗಿದೆ.

E **ಡಾಟಾ ರಕ್ಷಣೆ ಮತ್ತು ಪೋರ್ಟೇಬಲಿಟಿ** (Data Protection and Portability):

ಕ್ಲೌಡ್ ಸೇವೆಗಳನ್ನು ಕ್ಲೈಂಟ್ ಮತ್ತು ಪೂರೈಕೆದಾರರ ಒಪ್ಪಂದದ ಆಧಾರದ ಮೇಲೆ ನೀಡಲಾಗುತ್ತದೆ. ಆದರೆ ಒಪ್ಪಂದವು ಕೊನೆಗೊಂಡ ನಂತರ ಮತ್ತು ಕ್ಲೈಂಟ್ ಇನ್ನು ಮುಂದೆ ಮುಂದುವರೆಯಲು ಬಯಸುವುದಿಲ್ಲ ಎಂದಾದರೆ ಏನಾಗುತ್ತದೆ.

F Vendor Lock-in

ಈ ದುರ್ಬಲತೆಯು ಅಪಕ್ವವಾದ ಪೂರೈಕೆದಾರರು ಮತ್ತು ಹೊಸ ವ್ಯಾಪಾರದ ಮಾದರಿಗಳಿಂದ ಉಂಟಾಗುತ್ತದೆ. ಇದು ವೈಫಲ್ಯದ ಅಪಾಯವನ್ನು ಹೆಚ್ಚಿಸುತ್ತವೆ ಮತ್ತು ವ್ಯಾಪಾರದಿಂದ ಹೊರಹೋಗುವುದನ್ನು ಹೆಚ್ಚಿಸುತ್ತವೆ.

G **ಇಂಟರ್‌ನೆಟ್ ಅವಲಂಬನೆ** (Internet Dependency) :

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಇಂಟರ್‌ನೆಟ್ ಅವಲಂಬಿತ ತಂತ್ರಜ್ಞಾನವಾಗಿದೆ. ಇಲ್ಲಿ ಬಳಕೆದಾರರು ವೆಬ್ ಬ್ರೌಸರ್ ಮೂಲಕ ಸೇವೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾರೆ. ಇಂಟರ್‌ನೆಟ್ ಲಭ್ಯವಿಲ್ಲದಿದ್ದರೆ ಅಥವಾ ಸೇವೆಯು ಸ್ಥಗಿತಗೊಂಡರೆ ಏನಾಗಬಹುದು. ಬಳಕೆದಾರರ ಸಿಸ್ಟಮ್‌ಗಳು ಮತ್ತು ಆರೋಗ್ಯ ಮತ್ತು ಬ್ಯಾಂಕಿಂಗ್ ವ್ಯವಸ್ಥೆಗಳಂತಹ ಬಹಳ ನಿರ್ಣಾಯಕವಾದ ಮತ್ತು 24 ಗಂಟೆಯೂ ರನ್ ಆಗುವ ಅಗತ್ಯವಿರುವ ಕಾರ್ಯಾಚರಣೆಗಳು ಏನಾಗಬಹುದು.

ಆನ್‌ಲೈನ್ ಇ -ವಹಿವಾಟುಗಳಲ್ಲಿನ ದೌರ್ಬಲ್ಯಗಳು (Weaknesses in Online e-transactions)

ಆನ್‌ಲೈನ್ ವಹಿವಾಟುಗಳಲ್ಲಿ ಭಾರಿ ಹೆಚ್ಚಳವಾಗಿದೆ. ಅದಕ್ಕೆ ಸಮನಾಗಿ ಆನ್‌ಲೈನ್ ಪಾವತಿಯ ವ್ಯವಸ್ಥೆಯ ಸುರಕ್ಷತೆ ವಿರುದ್ಧ ದಾಳಿಗಳ ಸಂಖ್ಯೆಗಳಲ್ಲಿ ಮತ್ತು ಪ್ರಕಾರಗಳಲ್ಲಿಯೂ ಹೆಚ್ಚಳವಾಗಿದೆ. ಈ ಕೆಲವು ದಾಳಿಗಳು ದುರ್ಬಲತೆಗಳು ಬಳಸಿಕೊಂಡಿವೆ. ಇವುಗಳನ್ನು ಶಾಪಿಂಗ್ ಕಾರ್ಟ್ ಸಾಫ್ಟ್‌ವೇರ್‌ನಂತಹ ವೆಬ್‌ಸೈಟ್‌ಗಳಿಂದ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಮರುಬಳಕೆ ಮಾಡಬಹುದಾದ ಮೂರನೇ - ವ್ಯಕ್ತಿಗಳ ಘಟಕಗಳಲ್ಲಿ ಪ್ರಟಿಸಲಾಗಿರುತ್ತದೆ. ಇತರ ದಾಳಿಗಳು ಯಾವುದೇ ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿರುವ ದುರ್ಬಲತೆಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತವೆ. ಉದಾಹರಣೆಗೆ, SQL injection ಅಥವಾ cross-site scripting

ಆನ್‌ಲೈನ್ ಇ - ವಹಿವಾಟುಗಳಲ್ಲಿರುವ ದುರ್ಬಲತೆಗಳ ಸಾಮಾನ್ಯವಾಗಿ ವಿಧಗಳು ಯಾವವೆಂದರೆ : SQL injection, cross-site scripting,

ಮಾಹಿತಿ ಬಹಿರಂಗಪಡಿಸುವಿಕೆ , ಮಾರ್ಗ ಬಹಿರಂಗಪಡಿಸುವಿಕೆ, ಬೆಲೆಯನ್ನು ನಿರ್ವಹಿಸುವುದು, ಮತ್ತು ಬಫರ್ ಓವರ್ ಫ್ಲೋ.

ಈ ದುರ್ಬಲತೆಗಳನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಶೋಷಣೆಯು ವ್ಯಾಪಕವಾದ ಫಲಿತಾಂಶಗಳಿಗೆ ಕಾರಣವಾಗಬಹುದು. ಮಾಹಿತಿ ಮತ್ತು ಮಾರ್ಗದ ಬಹಿರಂಗಪಡಿಸುವಿಕೆಯ ದುರ್ಬಲತೆಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಮುಂದಿನ ಮತ್ತಷ್ಟು ಶೊಷಣೆಯ ಆರಂಭಿಕ ಹಂತಗಳಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ. SQL injection ಅಥವಾ price manipulation ದಾಳಿಗಳು ವೆಬ್ ಸೈಟನ್ನು ದುರ್ಬಲಗೊಳಿಸಬಹುದು, ಗೌಪ್ಯತೆಯನ್ನು ರಾಜಿಮಾಡಿಕೊಳ್ಳಬಹುದು, ಮತ್ತು ಕೆಟ್ಟ ಸಂದರ್ಭಗಳಲ್ಲಿ ಇ ಕಾಮರ್ಸ್ ವ್ಯಾಪಾರವನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಸ್ಥಗಿತಗೊಳಿಸಲು ಕಾರಣವಾಗಬಹುದು.

ಇಂತಹ ದುರ್ಬಲತೆಗಳಿಗೆ ಮುಖ್ಯ ಕಾರಣವೆಂದರೆ ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪರ್‌ಗಳು ಸುರಕ್ಷಿತ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ತಂತ್ರಗಳಲ್ಲಿ ಚೆನ್ನಾಗಿ ಪಾರಂಗತರಾಗಿರುವುದಿಲ್ಲ.

ಬ್ರೌಸರ್ ಭದ್ರತೆ ಮತ್ತು ಕುಕೀಗಳ ಮತ್ತು ಪಾಪ್-ಅಪ್‌ಗಳ ಪಾತ್ರ (Browser Security and Role of cookies and pop-ups)

ಭದ್ರತಾ ದೋಷಗಳು ಕುಕಿಯ ಡಾಟಾವನ್ನು ಹ್ಯಾಕರ್‌ಗಳು ಓದಲು ಅನುಮತಿ ನೀಡಬಹುದು. ಇವುಗಳನ್ನು ಬಳಕೆದಾರರ ಡಾಟಾಗೆ ಪ್ರವೇಶ ಪಡೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅಥವಾ ಕುಕಿಗಳು ಯಾವ ವೆಬ್‌ಸೈಟ್‌ಗೆ ಸೇರಿವೆಯೋ ಆ ವೆಬ್ ಸೈಟ್‌ಗೆ ಪ್ರವೇಶ ಪಡೆಯಲು (ಬಳಕೆದಾರರ ದೃಢೀಕರಣದೊಂದಿಗೆ) ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಪಾಪ್-ಅಪ್ ಜಾಹೀರಾತುಗಳು ಅಥವಾ ಪಾಪ್-ಅಪ್‌ಗಳು (Pop-up ads or pop-ups)

ಪಾಪ್-ಅಪ್ ವಿಂಡೊಗಳಿಂದ ಭದ್ರತಾ ಅಪಾಯಗಳು ಯಾವವೆಂದರೆ ಫಿಶಿಂಗ್, ಅನಗತ್ಯ ವೆಬ್‌ಸೈಟ್‌ಗಳಿಗೆ ಸಿಕ್ಕಿಹಾಕಿಕೊಳ್ಳುವುದು ಇತ್ಯಾದಿ.

© NIMI NOT TO BE REPUBLISHED

ಬ್ರೌಸರ್‌ನಲ್ಲಿ, ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ OS, ಮತ್ತು ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ನಲ್ಲಿ ಭದ್ರತಾ ರಂಧ್ರಗಳು (Security holes in Browser, Web Applications, OS, and Smartphones) : ಭದ್ರತಾ ಪರಿಭಾಷೆಯಲ್ಲಿ, ರಂಧ್ರ ಎಂದರೆ ರಂಧ್ರ ಇರುವ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್ ಅಥವಾ ನೆಟ್‌ವರ್ಕ್‌ನ ಸಂಪೂರ್ಣ ಭದ್ರತೆಯನ್ನು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳಲು ಶೋಷಣೆ ಮಾಡುವ ಸಾಫ್ಟ್‌ವೇರ್ ಅಥವಾ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ದುರ್ಬಲತೆ ಆಗಿರಬಹುದು. ಮೂರು ವಿಭಿನ್ನ ರೀತಿಯ ದುರ್ಬಲತೆಗಳು ಯಾವವೆಂದರೆ :

- ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ದೋಷಗಳು ಲಿನಕ್ಸ್ ಕರ್ನಲ್ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ; ಅಥವಾ ಮೈಕ್ರೊಸಾಫ್ಟ್, ಆ್ಯಪಲ್ ಮೂಲಕ ನಿರ್ಮಿಸಲಾದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಜೊತೆಗೆ ಸಾಗಿಸಲಾಗುವ ಘಟಕಗಳು, ಅಥವಾ ಸ್ವಾಮ್ಯದ ಯುನಿಕ್ ವ್ಯಾಪಾರಿ ಮತ್ತು ವ್ಯಾಪಾರಿಗಳಿಂದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ಭಾಗವೆಂದು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ.
- ಬ್ರೌಸರ್ ದೋಷಗಳು : ವೆಬ್ ಬ್ರೌಸರ್ ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುವ ಘಟಕಗಳ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಜೊತೆಗೆ ಸಾಗಿಸಲಾಗುವ ವೆಬ್ ಬ್ರೌಸರ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ಉದಾಹರಣೆಗೆ ವಿಂಡೋಸ್ ಇಂಟರ್‌ನೆಟ್ ಎಕ್ಸ್‌ಪ್ಲೋರರ್ ಮತ್ತು ಆ್ಯಪಲ್‌ನ ಸಫಾರಿ , ಜೊತೆಗೆ ಮೊಝಿಲ್ಲಾ ಫೈರ್‌ಫಾಕ್ಸ್ ಮತ್ತು ಗೂಗಲ್ ಕ್ರೋಮ್‌ನಂತಹ ಥರ್ಡ್ ಪಾರ್ಟೀ ಬ್ರೌಸರ್‌ಗಳು.
- ಅಪ್ಲಿಕೇಷನ್ ದೋಷಗಳು ಇತರ ಎಲ್ಲ ಘಟಕಗಳ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ಇವು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಮಾರಾಟಗಾರರು ಮತ್ತು ಇತರ ಮಾರಾಟಗಾರರು ಪ್ರಕಟಿಸಿದ ಘಟಕಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ಲಿನಕ್ಸ್ ವಿತರಣೆಗಳೊಂದಿಗೆ ಸಾಗಿಸಲಾಗುವ ಓಪನ್ ಸೋರ್ಸ್ ಘಟಕಗಳಲ್ಲಿನ ದೋಷಗಳನ್ನು (ಉದಾಹರಣೆಗೆ X ವಿಂಡೋಸ್ ಸಿಸ್ಟಮ್, GNOME ಡೆಸ್ಕ್‌ಟಾಪ್ ಪರಿಸರ , GIMP, ಮತ್ತು ಇತ್ಯಾದಿಗಳು) ಅಪ್ಲಿಕೇಷನ್ ದೋಷಗಳು ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ ಭದ್ರತಾ ರಂಧ್ರಗಳು (Security holes in Web Applications)

OWASP(Open Web Applications Security project) ವರ್ಗದಲ್ಲಿರುವ 10 ಮುಖ್ಯ ಬೆದರಿಕೆಗಳ ಪಟ್ಟಿಯನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ.:

Injection (Sqli -> SQL Injection), Broken Authentication & Session Management, XSS (Cross Site Scripting), Insecure Direct Object Reference, Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross Site Request Forgery (CSRF Or XSRF), ತಿಳಿದಿರುವ ದುರ್ಬಲತೆಗಳೊಂದಿಗೆ ಮತ್ತು ಮೌಲ್ಯೀಕರಿಸದ Redirect & Forwards ಗಳೊಂದಿಗೆ ಘಟಕಗಳನ್ನು ಬಳಸುವುದು.

OS ನಲ್ಲಿ ಭದ್ರತಾ ರಂಧ್ರಗಳು (Security holes in OS)

ಯುನಿಕ್ಸ್OS ನಲ್ಲಿರುವ ಕೆಲವು ದುರ್ಬಲತೆಗಳು Setuid ಸಮಸ್ಯೆಗಳು, Trojan Horses ಮತ್ತು Terminal Troubles.

ವಿಂಡೋಸ್ OS ನಲ್ಲಿರುವ ಕೆಲವು ದುರ್ಬಲತೆಗಳು ಎಂದರೆ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು , Peer to Peer ಫೈಲ್ ಹಂಚಿಕೆ , ಮೈಕ್ರೊಸಾಫ್ಟ್ ಔಟ್‌ಲುಕ್‌ನ ಎಂಬೆಡೆಡ್ ಆಟೋಮೇಷನ್ ಫೀಚರ್‌ನಲ್ಲಿರುವ ದುರ್ಬಲತೆಗಳು ಮತ್ತು ಕೆಟ್ಟ ಕೋಡನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಅನುಮತಿಸುವ ಔಟ್‌ಲುಕ್ ಎಕ್ಸ್‌ಪ್ರೆಸ್.

ಲಿನಕ್ಸ್ OS ನಲ್ಲಿರುವ ಕೆಲವು ದುರ್ಬಲತೆಗಳು ಎಂದರೆ ಪರ್ಮಿಷನ್ ಪರಿಶೀಲನೆಗಳು ಕಾಣೆಯಾಗಿರುವುದು, ಪ್ರಾರಂಭಿಸದ ಡಾಟಾ ಮತ್ತು ಮೆಮೊರಿಯ ತಪ್ಪು ನಿರ್ವಹಣೆ.

ಸ್ಮಾರ್ಟ್ ಫೋನ್‌ಗಳಲ್ಲಿರುವ ಕೆಲವು ದುರ್ಬಲತೆಗಳು ಎಂದರೆ ಸಾಧನದ ನಷ್ಟ ಅಥವಾ ಕಳ್ಳತನದಿಂದ ಉಂಟಾಗುವ ಡಾಟಾ ಸೋರಿಕೆ, ಉದ್ದೇಶಪೂರ್ವಕವಲ್ಲದ ಮಾಹಿತಿ ಬಹಿರಂಗಪಡಿಸುವಿಕೆ, ಸ್ಥಗಿತಗೊಳಿಸಲಾದ ಸ್ಮಾರ್ಟ್‌ಫೋನ್‌ಗಳ ಮೇಲೆ ದಾಳಿ, ಫಿಶಿಂಗ್ ದಾಳಿಗಳು, ಸ್ಪೈವೇರ್ ದಾಳಿಗಳು, ನೆಟ್‌ವರ್ಕ್ ವಂಚನೆಯ ದಾಳಿಗಳು ಸರ್ವೈಲೆನ್ಸ್ ದಾಳಿಗಳು (ಬಳಕೆದಾರರು ಕಣ್ಗಾವಲು ಅಡಿಯಲ್ಲಿರುತ್ತಾರೆ), ಡಯಲರ್‌ವೇರ್ ದಾಳಿಗಳು(ಹಣವನ್ನು ಕದಿಯುವುದು), ಹಣಕಾಸು ಮಾಲ್ವೇರ್ ದಾಳಿಗಳು (ರುಜುವಾತುಗಳನ್ನು ಕಳ್ಳತನ ಮಾಡುವುದು) ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ದಟ್ಟಣೆ.

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನ ಸಾಧನಗಳು ಮತ್ತು ತಂತ್ರಗಳು (Vulnerability Assessment Tools and Techniques)

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನ ಒಂದು ಭದ್ರತಾ ಅಭ್ಯಾಸವಾಗಿದೆ. ಇದು ದೌರ್ಬಲ್ಯಗಳನ್ನು ಗುರುತಿಸುತ್ತದೆ, ದೋಷಗಳನ್ನು ಗುರುತಿಸುತ್ತದೆ ಮತ್ತು ಪಟ್ಟಿ ಮಾಡುತ್ತದೆ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳು, ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಮತ್ತು ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿರುವ ಭದ್ರತಾ ಹೊಣೆಗಾರಿಕೆಗಳ ಬಗ್ಗೆ ಆವಿಷ್ಕಾರಗಳ ಮೇಲೆ ವರದಿ ಮಾಡುತ್ತದೆ.

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನವು ದುರ್ಬಲತೆಗಳನ್ನು ಕೆಳಗಿನವುಗಳ ಮೂಲಕ ಪತ್ತೆ ಮಾಡುತ್ತದೆ :

- ಭದ್ರತಾ ತಂತ್ರಜ್ಞಾನಗಳು
  - VA ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಉಪಕರಣಗಳು ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳು
- ಪರಿಹಾರ ತಂತ್ರಜ್ಞಾನಗಳು
  - ಪ್ಯಾಚ್ ಮ್ಯಾನೆಜನೆಂಟ್ ಸಿಸ್ಟಮ್‌ಗಳು (WSUS, SCCM,LanDesk,VMWare Update manager)

© NIMI NOT TO BE REPUBLISHED

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನವು ಮುಖ್ಯವಾಗಿ ಕೆಳಗಿನ ಮೂರು ಹಂತಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ :

- **ಮಾಹಿತಿ ಸಂಗ್ರಹಣೆ ಮತ್ತು ಅನ್ವೇಷಣೆ** - ಇವು ನೆಟ್‌ವರ್ಕ್ ಸ್ಕ್ಯಾನಿಂಗ್, ಪೋರ್ಟ್ ಸ್ಕ್ಯಾನಿಂಗ್, ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳು ಮತ್ತು DNZ ಝೋನ್‌ಗಳು ಮತ್ತು ರೆಜಿಸ್ಟರ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.
- **ಎನ್ಯುಮರೇಷನ್** - ಇದು ಹೋಸ್ಟ್‌ಗಳನ್ನು, OS ಗಳನ್ನು, ಪೋರ್ಟ್‌ಗಳನ್ನು, ಸೇವೆಗಳನ್ನು ಮತ್ತು ಅವುಗಳ ಆವೃತ್ತಿಗಳನ್ನು, ಮಾಹಿತಿ ಮತ್ತು SNMP ಸಮುದಾಯಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.
- **ಗುರುತಿಸುವಿಕೆಯು** - ದೌರ್ಬಲ್ಯಗಳ ಗುರುತಿಸುವಿಕೆಯನ್ನು, ದುರ್ಬಲತೆಗಳ ಗುರುತಿಸುವಿಕೆಯನ್ನು , ವರದಿ ತಯಾರಿಸುವಿಕೆಯನ್ನು ಮತ್ತು ಪರಿಹಾರ ಸಾಧನಗಳ ಬಳಕೆಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು (Vulnerability Assessment tools)

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು ವಿವಿಧ ಹಂತಗಳಲ್ಲಿ ಕಂಡುಬರುವ ದುರ್ಬಲತೆಗಳನ್ನು ಪತ್ತೆಮಾಡುತ್ತವೆ, ಗುರುತಿಸುತ್ತವೆ, ಅವುಗಳ ಪರಿಣಾಮವನ್ನು ಅಳತೆ ಮಾಡುತ್ತವೆ. ಹೆಚ್ಚಿನ ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು ನೆಟ್‌ವರ್ಕ್ ನೋಡ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಸ್ಕ್ಯಾನ್ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಹೊಂದಿವೆ. ಇವು ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನ್ನು, ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಮಾಡಲಾದ ಸಾಧನಗಳನ್ನು (ಸ್ವಿಚ್‌ಗಳು, ಫೈರ್‌ವಾಲ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು ಇತ್ಯಾದಿ) ಮತ್ತು ಸರ್ವರ್‌ಗಳನ್ನು, ಡೆಸ್ಕ್‌ಟಾಪ್‌ಗಳನ್ನು ಮತ್ತು ಪೋರ್ಟೆಬಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಸಾಮಾನ್ಯ ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು ಯಾವವೆಂದರೆ ನೆಟ್‌ವರ್ಕ್ ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಹೋಸ್ಟ್ ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಡಾಟಾಬೇಸ್ ಸ್ಕ್ಯಾನರ್‌ಗಳು, ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್ ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಮಲ್ಟಿಲೆವೆಲ್ ಸ್ಕ್ಯಾನರ್‌ಗಳು, ಆಟೋಮೇಟೆಡ್ ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷಾ ಸಾಧನಗಳು ಮತ್ತು ದುರ್ಬಲತೆ ಸ್ಕ್ಯಾನ್ ಕನ್ಸಾಲಿಡೇಟರ್‌ಗಳು.

ದುರ್ಬಲತೆಯನ್ನು ಬಳಸಿಕೊಳ್ಳುವ ತಂತ್ರಗಳು (Techniques to Exploit Vulnerabilities)

ದುರ್ಬಲತೆಗಳನ್ನು ಪ್ಯಾಕೆಟ್ ಸ್ನಿಫರ್‌ಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಉಂಟುಮಾಡಬಹುದು. ಇತರ ಸಾಧನಗಳನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ನಕಲಿ ವಿಳಾಸಗಳಿಂದ ಪ್ಯಾಕೆಟ್‌ಗಳನ್ನು ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ. ಈ ಸಾಧನಗಳ ಒಂದು ಉಪಯೋಗವೆಂದರೆ ಇದು denial-of-service ದಾಳಿಯನ್ನು ಉಂಟುಮಾಡುವುದು. ಇದು ದಾಳಿಯ ಮೂಲವನ್ನು ಮರೆಮಾಚುತ್ತದೆ. ದಾಳಿಕೋರರು ಕಂಪ್ಯೂಟರ್ ಅಡ್ರೆಸ್‌ಗಳನ್ನು, ತಮ್ಮ ನಿಜವಾದ ಗುರುತನ್ನು ಮರೆಮಾಡುತ್ತಾರೆ ಮತ್ತು ಅನುಮತಿ ಇಲ್ಲದ ಸಂಪರ್ಕಗಳನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಮಾಡಿ "spoof" (ವಂಚನೆ) ಮಾಡುತ್ತಾರೆ. ಈ ರೀತಿಯಲ್ಲಿ ಅವು ಕಂಪ್ಯೂಟರ್‌ಗಳ ನಡುವಿನ ವಿಶ್ವಾಸಾರ್ಹ ಸಂಬಂಧಗಳನ್ನು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತವೆ.

ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾದ ದುರ್ಬಳಕೆಗಳು ಇವುಗಳ ಬಳಕೆಯಿಂದ ಸಂಭವಿಸುತ್ತವೆ - ಟ್ರೋಜನ್‌ಗಳು, ವೈರಸ್‌ಗಳು, ವರ್ಮ್‌ಗಳು, ಲಾಜಿಕ್ ಬಾಂಬ್‌ಗಳು, ಫಿಶಿಂಗ್, ಅರ್ಬನ್ ಲೆಜೆಂಡ್‌ಗಳನ್ನು ಫರ್ವರ್ಡ್ ಮಾಡುವುದು ಮತ್ತು ಹಂಚಿಕೊಳ್ಳುವುದು, ನೈಜೀರಿಯನ್ ಹಗರಣಗಳಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುವುದು ಇತ್ಯಾದಿಗಳು

ದೋಷಗಳನ್ನು ಸರಿಪಡಿಸುವ ತಂತ್ರಗಳು (Techniques to Fix the Vulnerabilities)

ಪರಿಣಾಮಕಾರಿ ಪರಿಹಾರಗಳು ನಿರಂತರ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಬಯಸುತ್ತದೆ. ಇವೆಲ್ಲವುಗಳನ್ನು ಒಟ್ಟಾಗಿ ದುರ್ಬಲತೆಯ ನಿರ್ವಹಣೆ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ದುರ್ಬಲತೆಯ ನಿರ್ವಹಣೆಯಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಲ್ಪಡುವ ಪ್ರಕ್ರಿಯೆಗಳು ಮತ್ತು ಸಂಬಂಧಿತ ತಂತ್ರಜ್ಞಾನಗಳು ಸಂಸ್ಥೆಗಳಿಗೆ ನೆಟ್‌ವರ್ಕ್ ಭದ್ರತಾ ದುರ್ಬಲತೆಗಳನ್ನು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಹುಡುಕಲು ಮತ್ತು ಸರಿಪಡಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ. ಈ ಪ್ರಕ್ರಿಯೆಗಳ ವ್ಯವಸ್ಥಿತ ಬಳಕೆಯು ವ್ಯಾಪಾರ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಆಗಾಗ್ಗೆ ಹರಡುವ ವೈರಸ್‌ಗಳಿಂದ , ವರ್ಮ್‌ಗಳಿಂದ ಮತ್ತು ಇತರ ನೆಟ್‌ವರ್ಕ್‌ನ ದಾಳಿಗಳಿಂದ ರಕ್ಷಿಸುತ್ತದೆ.

ದುರ್ಬಲತೆಯ ನಿರ್ವಹಣೆಯ ನಿರಂತರ ಪ್ರಕ್ರಿಯೆಗಳು, ಭದ್ರತಾ ನೀತಿಗಳನ್ನು ಮತ್ತು ನಿಯಂತ್ರಣಗಳನ್ನು ರಚಿಸುವುದು, ಸ್ವತ್ತುಗಳನ್ನು ದಾಸ್ತಾನು/ ವರ್ಗೀಕರಿಸುವುದು, ದುರ್ಬಲತೆಗಳನ್ನು ಪತ್ತೆಹಚ್ಚಲು ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಸ್ಕ್ಯಾನ್ ಮಾಡುವುದು, ದುರ್ಬಲತೆಗಳನ್ನು ದಾಸ್ತಾನಿನ ವಿರುದ್ಧ ಹೋಲಿಸುವುದು, ಅಪಾಯಗಳನ್ನು ವರ್ಗೀಕರಿಸುವುದು, ಪ್ಯಾಚ್‌ಗಳ ಪೂರ್ವ ಪರೀಕ್ಷೆ, ಪ್ಯಾಚ್ ಗಳನ್ನು ಅನ್ವಯಿಸುವುದು ಮತ್ತು ರಿಸ್ಕ್ಯಾನ್ ಮಾಡುವುದು ಮತ್ತು ಸರಿಪಡಿಸುವಿಕೆಯನ್ನು ದೃಢೀಕರಿಸುವುದು. ಅವುಗಳಲ್ಲಿ ಹೆಚ್ಚಿನವುಗಳನ್ನು ನೀವು ಈಗ ಭದ್ರತಾ ಅಪ್ಲಿಕೇಷನ್ ಗಳನ್ನು ಮತ್ತು ವೆಬ್ ಆಧಾರಿತ ಸೇವೆಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಸರಿಪಡಿಸಬಹುದು.

ಭದ್ರತಾ ದುರ್ಬಲತೆಗಳನ್ನು ತಗ್ಗಿಸಲು ಉತ್ತಮ ಅಭ್ಯಾಸಗಳು ಮತ್ತು ಮಾರ್ಗಸೂಚಿಗಳು

1 ಬಳಸುವ ಮೊದಲು ಎಲ್ಲಾ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಪ್ರಾರಂಭಿಸಿ.

2 ಬಳಸುವ ಮೊದಲು ಎಲ್ಲಾ ಬಳಕೆದಾರರನ್ನು ಮೌಲ್ಯೀಕರಿಸಿ

3 ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಮತ್ತು ಡಾಟಾಬೇಸ್‌ಗಳಲ್ಲಿ ಆಡಳಿತಾತ್ಮಕ ಅನುಮತಿಗಳನ್ನು ನಿರ್ಬಂಧಿಸಿ.

© NIMI NOT TO BE REPUBLISHED

4 ದೋಷಗಳನ್ನು ನಿಭಾಯಿಸಿ ಮತ್ತು ಸಿಸ್ಟಮ್ ದೋಷ ಸಂದೇಶಗಳನ್ನು ಅಂತಿಮ ಬಳಕೆದಾರರಿಗೆ ಪ್ರದರ್ಶಿಸಬೇಡಿ.

5 ಕನಿಷ್ಟ ಪ್ರಮಾಣದ ಅನುಮತಿಗಳೊಂದಿಗೆ ಖಾತೆಗಳನ್ನು ಮತ್ತು ಅಗತ್ಯವಿರುವ ಸವಲತ್ತುಗಳನ್ನು ಒದಗಿಸಿ.

6 ನಿಮ್ಮ ಕೋಡ್‌ನಲ್ಲಿ ರಹಸ್ಯಗಳನ್ನು (ಉದಾಹರಣೆಗೆ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು, ಕೀಗಳು) ಸಂಗ್ರಹಿಸಿಡಬೇಡಿ.

7 ಸಾಮಾನ್ಯ ಫಂಕ್ಷನ್‌ಗಳಿಗಾಗಿ ಪರೀಕ್ಷಿಸಲ್ಪಟ್ಟ, ವಿಶ್ವಾಸಾರ್ಹ ಲೈಬ್ರರಿಗಳನ್ನು ಅಥವಾ ಮಾಡ್ಯೂಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ. (ಉದಾಹರಣೆಗೆ, ದೃಢೀಕರಣ, ಎನ್‌ಕ್ರಿಪ್ಷನ್, ಸೆಷನ್ ಟ್ರ್ಯಾಕಿಂಗ್)

8 ಲಾಗಿನ್ ಪುಟಗಳನ್ನು ಮತ್ತು ದೃಢೀಕರಣದಿಂದ ರಕ್ಷಿಸಲಾದ ಪುಟಗಳನ್ನು HTTPS ನಿಂದ ಸುರಕ್ಷಿತವಾಗಿಸಿ.

9 ಸರ್ವರ್ ಘಟಕಗಳು(OS, ಸಾಫ್ಟ್‌ವೇರ್ /ಆ್ಯಪ್‌ಗಳು) ಇಲ್ಲಿಯವರೆಗೆ ಸರಿಯಾಗಿವೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

10 ಉತ್ಪಾದನಾ ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಅನಗತ್ಯ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದನ್ನು ತಪ್ಪಿಸಿ.

11 ವೆಬ್ ಸರ್ವರ್‌ನಿಂದ ಬಳಕೆಯಾಗದ ಮತ್ತು ಬ್ಯಾಕಪ್ ಪುಟಗಳನ್ನು ತೆಗೆದುಹಾಕಿ.

12 ಸಾಧ್ಯವಾದರೆ ಕೋಡ್ ಲೈಬ್ರರಿಗಳನ್ನು ಮತ್ತು ಕಾನ್ಫಿಗರೇಷನ್ ಫೈಲ್‌ಗಳನ್ನು ವೆಬ್‌ನಿಂದ ಪಡೆದುಕೊಳ್ಳುವುದನ್ನು ತಪ್ಪಿಸಿ.

13 ಡೈರೆಕ್ಟರಿ ಬ್ರೌಸಿಂಗ್‌ನ್ನು ನಿಷ್ಕ್ರೀಯಗೊಳಿಸಿ.

14 ಬಳಕೆದಾರರ ಇನ್ಪುಟ್ ಆಧರಿಸಿ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಕಾಲ್ ಮಾಡುವುದನ್ನು ತಪ್ಪಿಸಿ.

15 ನಿಮ್ಮ ಡೆವಲಪ್‌ಮೆಂಟ್ ಫ್ರೇಮ್‌ವರ್ಕ್‌ನಲ್ಲಿ ನಿರ್ಮಿತವಾದ ಸೆಷನ್ ಟ್ರ್ಯಾಕಿಂಗ್ ಕಾರ್ಯತಂತ್ರವನ್ನು ಬಳಸಿಕೊಳ್ಳಿ.

ಅಪಾಯವು ಘಟನೆಯ ಸಂಭವನೀಯತೆ ಮತ್ತು ಅದರ ಪರಿಣಾಮಗಳ ಸಂಯೋಜನೆಯಾಗಿದೆ. ಇದು ನೀಡಿದ ದಾಳಿಯಿಂದ ಗುರಿಪಡಿಸಬಹುದಾದ ಸಾಧ್ಯತೆಯನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಬೆದರಿಕೆ, ದುರ್ಬಲತೆ, ಮತ್ತು ಅಪಾಯಗಳ ನಡುವಿನ ಸಂಬಂಧ (Relationship between Threat, Vulnerability, and Risk)

ಬೆದರಿಕೆ, ದುರ್ಬಲತೆ, ಮತ್ತು ಅಪಾಯಗಳ ನಡುವಿನ ಸಂಬಂಧವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಮೊದಲು ಕೆಳಗಿನ ಶಬ್ದಗಳನ್ನು ತಿಳಿದುಕೊಳ್ಳೋಣ.

ಆಸ್ತಿ (Asset): ಮಾಹಿತಿ ಭದ್ರತೆಯಲ್ಲಿ, ಆಸ್ತಿ ಎಂದರೆ ನಾವು ಯಾವುದನ್ನು ರಕ್ಷಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತೇವೆಯೊ ಅದು. ಅದು ಜನರು, ಆಸ್ತಿ, ಅಥವಾ ಮಾಹಿತಿ ಆಗಿರಬಹುದು.

ಬೆದರಿಕೆ (Threat) : ಉದ್ದೇಶಪೂರ್ವಕವಾಗಿ ಅಥವಾ ಆಕಸ್ಮಿಕವಾಗಿ ದುರ್ಬಲತೆಯನ್ನು ಬಳಸಿಕೊಳ್ಳುವ ಮತ್ತು ಪಡೆಯುವ, ಆಸ್ತಿಯನ್ನು ಹಾನಿ ಅಥವಾ ನಾಶ ಮಾಡುವ ಯಾವುದನ್ನೂ ಬೆದರಿಕೆ ಎನ್ನಬಹುದು.

ದುರ್ಬಲತೆ (Vulnerability) : ಇದು ನಮ್ಮ ರಕ್ಷಣಾ ಪ್ರಯತ್ನಗಳಲ್ಲಿರುವ ದೌರ್ಬಲ್ಯಗಳನ್ನು ಅಥವಾ ಅಂತರಗಳನ್ನು ಸೂಚಿಸುತ್ತದೆ.

ಅಪಾಯ (Risk) : ಬೆದರಿಕೆಯು ದುರ್ಬಲತೆಯನ್ನು ಬಳಸಿಕೊಂಡಾಗ, ಅದು ಆಸ್ತಿಯ ನಷ್ಟಕ್ಕೆ, ಹಾನಿಗೆ ಅಥವಾ ನಾಶಕ್ಕೆ ಕಾರಣವಾಗಬಹುದು. ಇದನ್ನು ಅಪಾಯ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ಅದ್ದರಿಂದ ಅಪಾಯವು ಆಸ್ತಿಯ, ಬೆದರಿಕೆಗಳ, ಮತ್ತು ದುರ್ಬಲತೆಗಳ ಪರಸ್ಪರ ಛೇದಕವಾಗಿದೆ.

ಎಂದರೆ ಆಸ್ತಿಗಳು X ಬೆದರಿಕೆಗಳು X ದುರ್ಬಲತೆಗಳು = ಅಪಾಯ

(ie. assets x threats x vulnerabilities = risk)

ಆಸ್ತಿಯ ಮೌಲ್ಯ (Value of an asset)

ಮಾಹಿತಿಯ ಮೇಲಿನ ಮೌಲ್ಯವು ಅದು ಒಳಗೊಂಡಿರುವ ಪಕ್ಷಗಳಿಗೆ, ಅದನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಯಾವ ಕೆಲಸದ ಅಗತ್ಯವಿದೆ , ಅದನ್ನು ನಿರ್ವಹಿಸಲು ಎಷ್ಟು ವೆಚ್ಚವಾಗುತ್ತದೆ , ಅದು ಕಳೆದುಹೋದರೆ ಅಥವಾ ನಷ್ಟವಾದರೆ ಎಷ್ಟು ಹಾನಿ ಉಂಟಾಗುತ್ತದೆ, ಶತ್ರುಗಳು ಅದಕ್ಕೆ ಎಷ್ಟು ಪಾವತಿಸುತ್ತಾರೆ ಮತ್ತು ಏನು ಹೊಣೆಗಾರಿಕೆಯ ದಂಡಗಳನ್ನು ಸಹಿಸಿಕೊಳ್ಳಬಹುದು ಎಂಬ ಮುಂತಾದ ವಿಷಯಗಳಿಗೆ ಸಂಬಂಧಿಸಿರುತ್ತದೆ. ಒಂದು ಕಂಪನಿಯ ಮಾಹಿತಿಯು ಅದು ರಕ್ಷಿಸಲು ಪ್ರಯತ್ನಿಸುವ ಮಾಹಿತಿಯ ಮೌಲ್ಯ ಮತ್ತು ಇತರ ಆಸ್ತಿಗಳ ಮೌಲ್ಯವನ್ನು ತಿಳಿದಿಲ್ಲವಾದರೆ, ಅವುಗಳನ್ನು ರಕ್ಷಿಸಲು ಅದು ಎಷ್ಟು ಹಣ ಮತ್ತು ಸಮಯವನ್ನು ಕಳೆದುಕೊಳ್ಳಬೇಕು ಎಂದು ತಿಳಿದಿರುವುದಿಲ್ಲ. ಆಸ್ತಿಗಳಿಗೆ ಮೌಲ್ಯವನ್ನು ನಿಯೋಜಿಸುವಾಗ , ಕೆಳಗೆ ನೀಡಲಾದ ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಗಣಿಸಬೇಕಾಗುತ್ತದೆ.:

- ಆಸ್ತಿಯನ್ನು ಸ್ವಾಧೀನಪಡಿಸಿಕೊಳ್ಳುವ ಅಥವಾ ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ವೆಚ್ಚ
- ಆಸ್ತಿಯನ್ನು ನಿರ್ವಹಿಸುವ ಮತ್ತು ರಕ್ಷಿಸುವ ವೆಚ್ಚ
- ಮಾಲೀಕರು ಮತ್ತು ಬಳಕೆದಾರರಿಗೆ ಆಸ್ತಿಯ ಮೌಲ್ಯ
- ವಿರೋಧಿಗಳಿಗೆ ಆಸ್ತಿಯ ಮೌಲ್ಯ
- ಮಾಹಿತಿಯ ಅಭಿವೃದ್ಧಿಗೆ ಹೋದ ಬೌದ್ಧಿಕ ಆಸ್ತಿಯ ಮೌಲ್ಯ
- ಇತರರು ಆಸ್ತಿಗಾಗಿ ಪಾವತಿಸಲು ಸಿದ್ಧರಿರುವ ಬೆಲೆ
- ಆಸ್ತಿ ಕಳೆದುಹೋದರೆ ಅಥವಾ ಹಾನಿಗೊಳಗಾದರೆ ಅದನ್ನು ಬದಲಾಯಿಸುವ ವೆಚ್ಚ

© NIMI NOT TO BE REPUBLISHED

- ಆಸ್ತಿಯು ಲಭ್ಯವಿಲ್ಲದಿದ್ದರೆ, ಅದಕ್ಕೆ ಪರಿಣಾಮ ಬೀರುವ ಕಾರ್ಯಾಚರಣೆ ಮತ್ತು ಉತ್ಪಾದನಾ ಚಟುವಟಿಕೆಗಳು
- ಆಸ್ತಿಯನ್ನು ರಾಜಿ ಮಾಡಿಕೊಂಡರೆ ಹೊಣೆಗಾರಿಕೆ ಸಮಸ್ಯೆಗಳು
- ಸಂಸ್ಥೆಯಲ್ಲಿನ ಆಸ್ತಿಯ ಉಪಯುಕ್ತತೆ ಮತ್ತು ಪಾತ್ರ

ಆಸ್ತಿಯ ಮೌಲ್ಯವನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವುದು ಯಾವ ಭದ್ರತಾ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಉಪಯೋಗಿಸಬೇಕು ಮತ್ತು ಅದನ್ನು ರಕ್ಷಿಸಲು ಯಾವ ಬಂಡವಾಳವನ್ನು ಬಳಸಬೇಕು ಎಂಬುದನ್ನು ಅರ್ಥಮಾಡಿಕೊಳ್ಳುವ ಮೊದಲ ಹಂತವಾಗಿದೆ.

ಬೆದರಿಕೆಯ ಮೂಲ / ಏಜಂಟ್ ಎಂದರೇನು?(What Is a Threat Source/Agent?)

ಬೆದರಿಕೆಯ ಮೂಲ / ಏಜಂಟ್ ಎಂದರೆ ಪರಿಣಾಮವನ್ನು ಉಂಟುಮಾಡುವ ಉದ್ದೇಶ ಮತ್ತು ಸಾಮರ್ಥ್ಯ ಹೊಂದಿರುವ ಒಂದು ಘಟಕವಾಗಿದೆ.

ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳು ಆಸ್ತಿಯ ವಿರುದ್ಧ ಈ ಕೆಳಗಿನ ಒಂದು ಅಥವಾ ಹೆಚ್ಚಿನ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು :

- ಪ್ರವೇಶ - ಸರಳ ಅನಧಿಕೃತ ಪ್ರವೇಶ
- ದುರುಪಯೋಗ - ಆಸ್ತಿಗಳ ಅನಧಿಕೃತ ಬಳಕೆ (ಉದಾಹರಣೆಗೆ , ಗುರುತಿನ ಕಳ್ಳತನ, ಸರ್ವರ್‌ನಲ್ಲಿ ಅಶ್ಲೀಲ ವಿತರಣಾ ಸೇವೆಯನ್ನು ಸ್ಥಾಪಿಸುವುದು ಇತ್ಯಾದಿ)
- ಬಹಿರಂಗಪಡಿಸುವುದು - ಬೆದರಿಕೆ ಏಜಂಟ್ ಅಕ್ರಮವಾಗಿ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸುತ್ತಾನೆ.
- ಮಾರ್ಪಾಡು ಮಾಡುವುದು - ಆಸ್ತಿಯಲ್ಲಿ ಅನಧಿಕೃತ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುವುದು.
- ಪ್ರವೇಶವನ್ನು ನಿರಾಕರಿಸುವುದು - ಇದು ನಾಶ ಮಾಡುವುದು, ಡಾಟಾ ಅಲ್ಲದ ಆಸ್ತಿಯ ಕಳ್ಳತನವನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳು ಕೆಳಗಿನವುಗಳಲ್ಲಿ ಯಾವುದಾದರೂ ಆಗಿರಬಹುದು :

ಈ ವ್ಯಕ್ತಿಗಳು ಅಥವಾ ಗುಂಪುಗಳನ್ನು ಕೆಳಗಿನಂತೆ ವರ್ಗೀಕರಿಸಬಹುದು :

- **ನಾನ್ ಟಾರ್ಗೆಟ್ ಸ್ಪೆಸಿಫಿಕ್** : ನಾನ್ ಟಾರ್ಗೆಟ್ ಸ್ಪೆಸಿಫಿಕ್ ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳು ಹೀಗಿರುತ್ತವೆ - ಕಂಪ್ಯೂಟರ್ ವೈರಸ್‌ಗಳು, ವರ್ಮ್‌ಗಳು, ಟ್ರೋಜನ್‌ಗಳು, ಮತ್ತು ಲಾಜಿಕ್ ಬಾಂಬ್‌ಗಳು
- **ಉದ್ಯೋಗಿಗಳು** : ಸಿಬ್ಬಂದಿ, ಗುತ್ತಿಗೆದಾರರು, ಕಾರ್ಯಾಚರಣೆ/ ನಿರ್ವಹಣಾ ಸಿಬ್ಬಂದಿ, ಅಥವಾ ಭದ್ರತಾ ಸಿಬ್ಬಂದಿ. ಇವರು ಕಂಪನಿಯಿಂದ ತೊಂದರೆಗೊಳಗಾದವರಿರಬಹುದು.
- **ಸಂಘಟಿತ ಅಪರಾಧ ಮತ್ತು ಅಪರಾಧಿಗಳು** : ಅಪರಾಧಿಗಳು ಅವರಿಗೆ ಮೌಲ್ಯಯುವಾದ ಮಾಹಿತಿಯಾದ ಬ್ಯಾಂಕ್ ಖಾತೆಗಳು, ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್‌ಗಳು, ಅಥವಾ ಹಣವಾಗಿ ಪರಿವರ್ತಿಸಬಹುದಾದ ಬೌದ್ಧಿಕ ಆಸ್ತಿ ಮುಂತಾದವುಗಳನ್ನು ಗುರಿಯಾಗಿಸುತ್ತಾರೆ. ಅಪರಾಧಿಗಳು ಹೆಚ್ಚಾಗಿ ಅವರಿಗೆ ಸಹಾಯ ಮಾಡಲು ಒಳಗಿನವರನ್ನು ಬಳಸುತ್ತಾರೆ.
- **ನಿಗಮಗಳು** : ನಿಗಮಗಳು ಮಾಹಿತಿ ಯುದ್ಧ ಅಥವಾ ಸ್ಪರ್ಧಾತ್ಮಕ ಬುದ್ಧಿಮತ್ತೆಯಲ್ಲಿ ತೊಡಗಿರುತ್ತವೆ. ಪಾಲುದಾರರು ಮತ್ತು ಸ್ಪರ್ಧಿಗಳು ಈ ವರ್ಗದ ಅಡಿಯಲ್ಲಿ ಬರುತ್ತಾರೆ.
- **ಉದ್ದೇಶಪೂರ್ವಕವಲ್ಲದ ಮಾನವರು** : ಅಪಘಾತಗಳು, ಅಜಾಗರೂಕತೆ
- **ಉದ್ದೇಶಪೂರ್ವಕ ಮಾನವರು** : ಒಳಗಿನವರು, ಹೊರಗಿನವರು
- **ನೈಸರ್ಗಿಕ** : ಪ್ರವಾಹ, ಬೆಂಕಿ, ಮಿಂಚು, ಉಲ್ಕೆಗಳು, ಭೂಕಂಪಗಳು

ಅಪಾಯದ ನಿಯಂತ್ರಣಗಳು (Risk Controls)

ಈ ಗುರಿಯನ್ನು ಸಾಧಿಸಲು, ಅಪಾಯದ ತಗ್ಗಿಸುವಿಕೆಯು ಮಾಹಿತಿ ಭದ್ರತೆಯ ಕೇಂದ್ರ ಬಿಂದುವಾಗಿದ್ದರೆ, ನಿಯಂತ್ರಣಗಳು ಪ್ರಾಥಮಿಕ ಸಾಧನಗಳಾಗಿವೆ. ನಿಯಂತ್ರಣ ಎಂದರೆ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಯಾವುದೇ ಸಾಧನ ಅಥವಾ ಪ್ರಕ್ರಿಯೆ ಆಗಿದೆ.

ಮುಖ್ಯವಾಗಿ ಮೂರು ರೀತಿಯ ನಿಯಂತ್ರಣಗಳಿವೆ. ಅವು ಯಾವವೆಂದರೆ :

1 **ಆಡಳಿತಾತ್ಮಕ** (Administrative) :ಆಡಳಿತಾತ್ಮಕ ನಿಯಂತ್ರಣಗಳು ಜನರು ತೆಗೆದುಕೊಳ್ಳುವ ಕ್ರಿಯೆಗಳಾಗಿವೆ. ಆಡಳಿತಾತ್ಮಕ ನಿಯಂತ್ರಣಗಳು ನೀತಿ ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಮತ್ತು ಅನುಸರಣೆಯನ್ನು ಖಾತ್ರಿಪಡಿಸುವ ಪ್ರಕ್ರಿಯೆಗಳಾಗಿವೆ.

2 **ತಾಂತ್ರಿಕ ಮತ್ತು ತಾರ್ಕಿಕ** (Technical or Logical): ಇವು ವರ್ಚುವಲ್, ಅಪ್ಲಿಕೇಷನ್ ಮತ್ತು ತಾಂತ್ರಿಕ ನಿಯಂತ್ರಣಗಳಾಗಿವೆ. (ಸಿಸ್ಟಮ್‌ಗಳು, ಸಾಫ್ಟ್‌ವೇರ್) ಉದಾಹರಣೆಗೆ ಫೈರ್‌ವಾಲ್‌ಗಳು, ಆ್ಯಂಟಿವೈರಸ್ ಸಾಫ್ಟವೇರ್, ಎನ್‌ಕ್ರಿಪ್ಷನ್ ಮತ್ತು ಮೆಕರ್ . ಚೆಕರ್ ಅಪ್ಲಿಕೇಷನ್ ರೂಟೀನ್‌ಗಳು. ತಾಂತ್ರಿಕ ನಿಯಂತ್ರಣಗಳನ್ನು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ಗಳು ಕೈಗೊಳ್ಳುತ್ತವೆ ಅಥವಾ ನಿರ್ವಹಿಸುತ್ತವೆ.

3 **ಚಟುವಟಿಕೆ ಹಂತದ ನಿಯಂತ್ರಣಗಳು** (Activity phase controls) : ಇವು ತಾಂತ್ರಿಕವಾಗಿರಬಹುದು ಅಥವಾ ಆಡಳಿತಾತ್ಮಕವಾಗಿರಬಹುದು. ಇವುಗಳನ್ನು ಅಪಾಯ ತಗ್ಗಿಸುವಿಕೆಯ ಮಟ್ಟದ ಆಧಾರದ ಮೇಲೆ ಕೆಳಗಿನಂತೆ ವರ್ಗೀಕರಿಸಲಾಗಿದೆ :

NOT TO BE REPUBLISHED © NIMI

- ದೌರ್ಬಲ್ಯಗಳ ಸಂಪರ್ಕಕ್ಕೆ ಒಲುವ ಬೆದರಿಕೆಗಳನ್ನು ತಡೆಗಟ್ಟಲು ತಡೆಗಟ್ಟುವ ನಿಯಂತ್ರಣಗಳು ಅಸ್ತಿತ್ವದಲ್ಲಿವೆ. ಈ ನಿಯಂತ್ರಣಗಳು ನಷ್ಟ ಅಥವಾ ಹಾನಿ ಸಂಭವಿಸುವುದನ್ನು ತಡೆಗಟ್ಟುತ್ತವೆ. ಉದಾಹರಣೆಗೆ ಜವಾಬ್ದಾರಿಗಳ ಪ್ರತ್ಯೇಕತೆಯನ್ನು ಜಾರಿಗೊಳಿಸುವ ನಿಯಂತ್ರಣ (ಒಬ್ಬ ವ್ಯಕ್ತಿ ಪಾವತಿಯ ವಿನಂತಿಯನ್ನು ಸಲ್ಲಿಸುತ್ತಾನೆ ಆದರೆ ಎರಡನೇಯ ವ್ಯಕ್ತಿ ಅದನ್ನು ಅಧಿಕೃತಗೊಳಿಸಬೇಕು), ಉದ್ಯೋಗಿಯು ಮೋಸಮಾಡುವಂತಹ ಪಾವತಿಗಳನ್ನು ನೀಡಬಹುದಾದ ಅವಕಾಶವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

- ನಮ್ಮ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿರುವ ಬೆದರಿಕೆಯನ್ನು ಗುರುತಿಸಲು ಪತ್ತೆದಾರಿ ನಿಯಂತ್ರಣಗಳು ಅಸ್ತಿತ್ವದಲ್ಲಿವೆ. ಈ ನಿಯಂತ್ರಣಗಳು ಚಟುವಟಿಕೆಗಳನ್ನು ನೋಡಿಕೊಳ್ಳುತ್ತವೆ. ಇದು ಅಭ್ಯಾಸಗಳು ಅಥವಾ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಅನುಸರಿಸದೇ ಇರುವಲ್ಲಿನ ನಿದರ್ಶನಗಳನ್ನು ಗುರುತಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಒಂದು ವ್ಯಾಪಾರವು ಸಾಮಾನ್ಯ ಲೆಡ್ಜರ್‌ನ್ನು ಸಮನ್ವಯಗೊಳಿಸಬಹುದು ಅಥವಾ ಮೋಸದ ಪಾವತಿಗಳನ್ನು ಗುರುತಿಸಲು ಪಾವತಿಯ ವಿನಂತಿಯ ಆಡಿಟ್ ಲಾಗ್‌ಗಳ ವಿಮರ್ಶೆಯನ್ನು ಮಾಡಬಹುದು.

- ಬೆದರಿಕೆಯ ಪರಿಣಾಮಗಳನ್ನು ತಗ್ಗಿಸಲು ಅಥವಾ ಕಡಿಮೆ ಮಾಡಲು ಸರಿಪಡಿಸುವ ನಿಯಂತ್ರಣಗಳು ಅಸ್ತಿತ್ವದಲ್ಲಿವೆ. ಸರಿಪಡಿಸುವ ನಿಯಂತ್ರಣಗಳು ಸಿಸ್ಟಮ್‌ನ್ನು ಅದು ಹಾನಿಕಾರಕ ಘಟನೆಯ ಮೊದಲಿನ ಸ್ಥಿತಿಗೆ ಹಿಂತಿರುಗಿಸಿ ಸಿಸ್ಟಮ್‌ನ್ನು ಅಥವಾ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಮರುಸ್ಥಾಪಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಒಂದು ವ್ಯವಹಾರವು ಯಾರೊ ಪಾವತಿಯ ಡಾಟಾವನ್ನು ಬದಲಾಯಿಸಿದ್ದಾರೆ ಎನ್ನುವ ಪುರಾವೆಗಳು ಕಂಡುಬಂದ ನಂತರ ಬ್ಯಾಕಪ್ ಟೇಪ್‌ಗಳಿಂದ ಸಿಸ್ಟಮ್‌ನ ಪೂರ್ಣ ಮರುಸ್ಥಾಪನೆಯನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದು.

- ಪರಿಹಾರ ನಿಯಂತ್ರಣಗಳು ಮೂಲದಲ್ಲಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾದ ನಿಯಂತ್ರಣಗಳನ್ನು ಪರಿಸರದ ಮಿತಿಗಳ ಕಾರಣದಿಂದಾಗಿ ಬಳಸಲು ಆಗದೇ ಇದ್ದಾಗ ಮೂಲ ನಿಯಂತ್ರಣಗಳ ಉದ್ದೇಶವನ್ನು ಸಾಧ್ಯವಾದಷ್ಟೂ ಹತ್ತಿರದಿಂದ ನಿಯಂತ್ರಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾದ ಪರ್ಯಾಯ ನಿಯಂತ್ರಣಗಳಾಗಿವೆ.

ಅಪಾಯದ ಸಾಧ್ಯತೆ (Risk likelihood)

ಅಪಾಯದ ಸಂಭವನೀಯತೆ ಎಂದರೆ ಆಕ್ರಮಣಕಾರರಿಂದ ಒಂದು ನಿರ್ದಿಷ್ಟ ದುರ್ಬಲತೆಯನ್ನು ಹೇಗೆ ಬಹಿರಂಗಪಡಿಸಬೇಕು ಮತ್ತು ಬಳಸಿಕೊಳ್ಳಬೇಕು ಎನ್ನುವುದರ ಸ್ಥೂಲ ಅಳತೆಯಾಗಿದೆ. ಇದರಲ್ಲಿ ಹೆಚ್ಚು ನಿಖರವಾಗಿರುವುದು ಅನಿವಾರ್ಯವಲ್ಲ. ಸಾಮಾನ್ಯವಾಗಿ ಸಂಭವನೀಯತೆಯು ಕಡಿಮೆ, ಮಾಧ್ಯಮ, ಅಥವಾ ಹೆಚ್ಚು ಇದೆಯೆ ಎಂದು ಗುರುತಿಸಿದರೆ ಸಾಕು.

ಸಂಭವನೀಯತೆಯನ್ನು ನಿರ್ಧರಿಸಲು ಸಹಾಯ ಮಾಡುವ ಹಲವಾರು ಅಂಶಗಳಿವೆ. ಮೊದಲನೆಯ ಅಂಶಗಳ ಗುಂಪು ಏಜಂಟನ್ನು ಒಳಗೊಂಡಿರುವುದಕ್ಕೆ ಸಂಬಂಧಿಸಿದ್ದಾಗಿದೆ. ಇದು ಸಂಭವನೀಯ ದಾಳಿಕೋರರ ಗುಂಪಿನ ಯಶಸ್ವಿ ದಾಳಿಯ ಸಂಭವನೀಯತೆಯನ್ನು ಅಂದಾಜು ಮಾಡುವ ಗುರಿಯಾಗಿದೆ. ಇಲ್ಲಿ ನಿರ್ದಿಷ್ಟ ದುರ್ಬಲತೆಯನ್ನು ಉಂಟುಮಾಡುವ ಹಲವಾರು ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳಿರಬಹುದು ಎಂಬುದನ್ನು ಗಮನಿಸಿ. ಅದ್ದರಿಂದ ಸಾಮಾನ್ಯವಾಗಿ ಕೆಟ್ಟ ಪ್ರಕರಣದ ಸನ್ನಿವೇಶವನ್ನು ಬಳಸುವುದು ಉತ್ತಮ. ಉದಾಹರಣೆಗೆ, ಅನಾಮಧೇಯ ಹೊರಗಿನ ಆಕ್ರಮಣಕಾರರಿಗಿಂತ ಒಳಗಿನವರು ಹೆಚ್ಚು ಆಕ್ರಮಣಕಾರಿಯಾಗಿರುತ್ತಾರೆ. ಆದರೆ, ಇದು ಹಲವಾರು ಅಂಶಗಳ ಮೇಲೆ ಅವಲಂಬಿತವಾಗಿರುತ್ತದೆ.

ಮೊದಲನೆಯ ಅಂಶಗಳ ಗುಂಪು ಏಜಂಟನ್ನು ಒಳಗೊಂಡಿರುವುದಕ್ಕೆ ಸಂಬಂಧಿಸಿದ್ದಾಗಿದೆ. ಇದು ಸಂಭವನೀಯ ದಾಳಿಕೋರರ ಗುಂಪಿನ ಯಶಸ್ವಿ ದಾಳಿಯ ಸಂಭವನೀಯತೆಯನ್ನು ಅಂದಾಜು ಮಾಡುವ ಗುರಿಯಾಗಿದೆ. ಎರಡನೆಯ ಅಂಶವು ದಾಳಿಗಳ ಹಿಂದಿನ ಉದ್ದೇಶವನ್ನು ಗಣನೆಗೆ ತೆಗೆದುಕೊಳ್ಳುವುದಾಗಿದೆ. ಅಗತ್ಯವಿರುವ ಪ್ರವೇಶ ಅಥವಾ ಸಂಪನ್ಮೂಲಗಳು ಮತ್ತು ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳ ಗುಂಪಿನ ಗಾತ್ರ ಇವು ಪರಿಗಣಿಸಬೇಕಾದ ಇತರ ಅಂಶಗಳಾಗಿವೆ.

ಮುಂದಿನ ಅಂಶಗಳು ದುರ್ಬಲತೆಗೆ ಸಂಬಂಧಿಸಿವೆ. ಕಂಡುಹಿಡಿದ ಮತ್ತು ಬಳಸಿಕೊಂಡ ನಿರ್ದಿಷ್ಟ ದುರ್ಬಲತೆಯ ಸಂಭವನೀಯತೆಯನ್ನು ಅಂದಾಜು ಮಾಡುವುದು ಇದರ ಗುರಿಯಾಗಿದೆ. ಇದು ಅನ್ವೇಷಣೆಯ ಸುಲಭತೆ, ಶೋಷಣೆಯ ಸುಲಭತೆ, ಬೆದರಿಕೆ ಏಜಂಟ್‌ಗಳ ಈ ಗುಂಪಿಗೆ ಅರಿವು , ಈ ಬಳಕೆಯ ಪತ್ತೆಹಚ್ಚುವಿಕೆಯ ಸಂಭವನೀಯತೆ ಮುಂತಾದವುಗಳನ್ನು ಗಣನೆಗೆ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.

ಪ್ರಭಾವವನ್ನು ಅಂದಾಜು ಮಾಡುವ ಅಂಶಗಳು (Factors for Estimating Impact)

ಒಂದು ಯಶಸ್ವಿ ದಾಳಿಯ ಪರಿಣಾಮವನ್ನು ಪರಿಗಣಿಸುವಾಗ, ಅಲ್ಲಿ ಎರಡು ರೀತಿಯ ಪರಿಣಾಮಗಳಿವೆ ಎಂಬುದನ್ನು ಅರಿತುಕೊಳ್ಳುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ. ಮೊದಲನೆಯದು ಅಪ್ಲಿಕೇಷನ್ ಮೇಲೆ “ತಾಂತ್ರಿಕ ಪ್ರಭಾವ”, ಅದು ಬಳಸುವ ಡಾಟಾ, ಮತ್ತು ಅದು ಒದಗಿಸುವ ಫಂಕ್ಷನ್‌ಗಳು. ಇನ್ನೊಂದು, ವ್ಯಾಪಾರದ ಮತ್ತು ಕಂಪನಿಯ ಮೇಲೆ “ವ್ಯಾಪಾರ ಪ್ರಭಾವ ”,

ಅಂತಿಮವಾಗಿ, ವ್ಯಾಪಾರದ ಪ್ರಭಾವವು ಹೆಚ್ಚು ಮುಖ್ಯವಾಗಿದೆ. ಅದಾಗ್ಯೂ, ಯಶಸ್ವಿ ಶೋಷಣೆಯ ವ್ಯವಹಾರದ ಪರಿಣಾಮಗಳನ್ನು ಲೆಕ್ಕಾಚಾರ ಮಾಡುವ ಅಗತ್ಯವಿರುವ ಎಲ್ಲಾ ಮಾಹಿತಿಗೆ ಪ್ರವೇಶವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಿಲ್ಲದಿರಬಹುದು. ಈ ಸಂದರ್ಭದಲ್ಲಿ ತಾಂತ್ರಿಕ ಅಪಾಯದ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ವಿವರಗಳನ್ನು ಒದಗಿಸುವುದು ವ್ಯಾಪಾರ ಪ್ರತಿನಿಧಿಗಳಿಗೆ ವ್ಯಾಪಾರ ಅಪಾಯದ ಬಗ್ಗೆ ಸೂಕ್ತ ನಿರ್ಧಾರ ತೆಗೆದುಕೊಳ್ಳಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ತಾಂತ್ರಿಕ ಪ್ರಭಾವವನ್ನು ಸಾಂಪ್ರದಾಯಿಕ ಭದ್ರತಾ ಕ್ಷೇತ್ರಗಳೊಂದಿಗೆ ಜೋಡಿಸಲಾದ ಅಂಶಗಳಾಗಿ ವಿಂಗಡಿಸಬಹುದು : ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ, ಲಭ್ಯತೆ, ಮತ್ತು ಹೊಣೆಗಾರಿಕೆ. ಒಂದು ವೇಳೆ ದುರ್ಬಲತೆಯನ್ನು ಬಳಸಿಕೊಂಡರೆ ಸಿಸ್ಟಮ್ ಮೇಲಿನ ಪ್ರಭಾವದ ಪ್ರಮಾಣವನ್ನು ಅಂದಾಜು ಮಾಡುವುದು ಇದರ ಗುರಿಯಾಗಿದೆ. ಪರಿಗಣಿಸಬೇಕಾದ ಸಮಸ್ಯೆಗಳು ಯಾವವೆಂದರೆ : ಗೌಪ್ಯತೆಯ ನಷ್ಟ, ಸಮಗ್ರತೆಯ ನಷ್ಟ, ಲಭ್ಯತೆಯ ನಷ್ಟ, ಮತ್ತು ಹೊಣೆಗಾರಿಕೆಯ ನಷ್ಟ.

ವ್ಯಾಪಾರದ ಪ್ರಭಾವದ ಮೌಲ್ಯಮಾಪನ ಮಾಡಲು ಪರಿಗಣಿಸಬೇಕಾದ ಅಂಶಗಳು ಯಾವವೆಂದರೆ : ಹಣಕಾಸಿನ ಹಾನಿ, ಖ್ಯಾತಿಯ ಹಾನಿ, ಅನುಸರಣೆಇಲ್ಲದ ನೀತಿಗಳು ಮತ್ತು ಗೌಪ್ಯತೆಯ ಉಲ್ಲಂಘನೆಯ.

ಅಪಾಯ ನಿಯಂತ್ರಣ ಪರಿಣಾಮಕಾರಿತ್ವ (Risk Control Effectiveness): ಅಪಾಯ ನಿಯಂತ್ರಣ ಪರಿಣಾಮಕಾರಿತ್ವವು ಇವುಗಳನ್ನು ಅವಲಂಬಿಸಿದೆ - ಗುರುತಿಸಲಾದ ವ್ಯವಸ್ಥಿತ ಅಪಾಯಗಳ ಸಂಖ್ಯೆ, ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದಲ್ಲಿ ಒಳಗೊಂಡಿರುವ ಪ್ರಕ್ರಿಯೆ ಪ್ರದೇಶಗಳ ಶೇಕಡಾವಾರು, ತಗ್ಗಿಸಲಾದ ಪ್ರಮುಖ ಅಪಾಯಗಳ ಶೇಕಡಾವಾರು, ಮತ್ತು ಹಲವಾರು ಅಂಶಗಳಲ್ಲಿನ ಪ್ರಮುಖ ಅಪಾಯಗಳ ಮೇಲ್ವಿಚಾರಣೆ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ಅಪಾಯ ನಿರ್ವಹಣೆ (Risk Management)

ಅಪಾಯ ನಿರ್ವಹಣೆ ಮತ್ತು ಅಪಾಯ ಮೌಲ್ಯಮಾಪನ ಎರಡೂ ಮಾಹಿತಿ ಭದ್ರತಾ ನಿರ್ವಹಣೆ (Information Security Management (ISM)) ನ ಮುಖ್ಯವಾದ ಘಟಕಗಳಾಗಿವೆ.

ಅಪಾಯ ನಿರ್ವಹಣೆ ಒಂದು ಸಂಸ್ಥೆಯು ಉಪಯೋಗಿಸುವ ಮಾಹಿತಿ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ದುರ್ಬಲತೆಗಳನ್ನು ಮತ್ತು ಬೆದರಿಕೆಗಳನ್ನು ಗುರುತಿಸುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ ಮತ್ತು ಆಸ್ತಿಯ ಮೌಲ್ಯದ ಆಧಾರದ ಮೇಲೆ ಕಡಿಮೆ ಮಾಡಲು ಪ್ರತಿಕ್ರಮಗಳನ್ನು ನಿರ್ಧರಿಸುವುದು.

ಅಪಾಯ ನಿರ್ವಹಣೆಯ ಪ್ರಕ್ರಿಯೆಯು ನಡೆಯುತ್ತಿರುವ, ಪುನರಾವರ್ತನೆಯಾಗುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಇದು ಅನಿರ್ದಿಷ್ಟವಾಗಿ ಪುನರಾವರ್ತನೆಯಾಗಬೇಕಾಗುತ್ತದೆ. ವ್ಯವಹಾರ ಪರಿಸರವು ನಿರಂತರವಾಗಿ ಬದಲಾಗುತ್ತಿರುತ್ತದೆ ಮತ್ತು ಪ್ರತಿದಿನವೂ ಹೊಸ ಬೆದರಿಕೆಗಳು ಮತ್ತು ದುರ್ಬಲತೆಗಳು ಹೊರಹೊಮ್ಮುತ್ತಿರುತ್ತವೆ. ಎರಡನೆಯದಾಗಿ ಅಪಾಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಬಳಸಲಾಗುವ ಪ್ರತಿಕ್ರಮಗಳನ್ನು (ನಿಯಂತ್ರಣಗಳು) ಆಯ್ಕೆಮಾಡುವುದು. ಇವು ಉತ್ಪಾದಕತೆ, ವೆಚ್ಚ, ಪ್ರತಿಕ್ರಮಗಳ ಪರಿಣಾಮಕಾರಿತ್ವ, ಮತ್ತು ರಕ್ಷಿಸಲಾದ ಮಾಹಿತಿ ಆಸ್ತಿಯ ಮೌಲ್ಯ ಇವುಗಳ ನಡುವೆ ಸಮತೋಲನವನ್ನು ಸಾಧಿಸುತ್ತವೆ.

ಇದಕ್ಕೆ ವಿರುದ್ಧವಾಗಿ, ಅಪಾಯ ನಿರ್ವಹಣೆಯನ್ನು ನಿರ್ದಿಷ್ಟ ಸಮಯದಲ್ಲಿ ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ. (ಉದಾ:ವರ್ಷಕ್ಕೊಮ್ಮೆ, ಬೇಡಿಕೆಯ ಆಧಾರದ ಮೇಲೆ) ಮತ್ತು ಮುಂದಿನ ಮೌಲ್ಯಮಾಪನದ ಕಾರ್ಯಕ್ಷಮತೆಯ ತನಕ - ಇದು ಮೌಲ್ಯಮಾಪನಗೊಂಡ ಅಪಾಯಗಳ ತಾತ್ಕಾಲಿಕ ನೋಟವನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಅಪಾಯ ನಿರ್ವಹಣೆಯ ಉದ್ದೇಶ (Purpose of Risk Management)

ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿ ಅಪಾಯವನ್ನು ನಿರ್ವಹಿಸುವ ಮೂಲ ಕಾರಣ ಆ ಸಂಸ್ಥೆಯ ಧ್ಯೇಯ ಮತ್ತು ಆಸ್ತಿಗಳನ್ನು ರಕ್ಷಿಸುವುದು. ಅದ್ದರಿಂದ ಅಪಾಯ ನಿರ್ವಹಣೆಯು ತಾಂತ್ರಿಕ ಕಾರ್ಯಕ್ಕಿಂತಲೂ ನಿರ್ವಹಣಾ ಕಾರ್ಯವಾಗಿರಬೇಕು. ಒಂದು ಸಿಸ್ಟಮ್‌ಗೆ ಅಪಾಯವನ್ನು ಮತ್ತು ವಿಶೇಷವಾಗಿ ನಿರ್ದಿಷ್ಟ ಅಪಾಯವನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವುದು ಸಿಸ್ಟಮ್ ಮಾಲೀಕರಿಗೆ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಯನ್ನು ಸಂಸ್ಥೆಯ ಮೌಲ್ಯಕ್ಕೆ ಅನುಗುಣವಾಗಿ ರಕ್ಷಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಎಲ್ಲ ಸಂಸ್ಥೆಗಳು ಸೀಮಿತ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ ಮತ್ತು ಅಪಾಯವನ್ನು ಎಂದೂ ಶೂನ್ಯಕ್ಕೆ ಇಳಿಸಲಾಗುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ಅಪಾಯವನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವುದು ವಿಶೇಷವಾಗಿ ಅಪಾಯದ ಪ್ರಮಾಣವನ್ನು, ಸಂಸ್ಥೆಗಳಿಗೆ ವಿರಳ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಆದ್ಯತೆ ನೀಡಲು ಅನುಮತಿಸುತ್ತದೆ.

ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನ (ಹಂತಗಳು) (Risk Assessment (Phases))

ಅಪಾಯವನ್ನು ನಿರ್ಣಯಿಸುವ ಉದ್ದೇಶವು ಸಂಪನ್ಮೂಲಗಳನ್ನು ಎಲ್ಲಿ ನಿರ್ದೇಶಿಸಬೇಕು ಎಂದು ನಿರ್ಧರಿಸುವ ನಿರ್ವಹಣೆಗೆ ಸಹಾಯ ಮಾಡುವುದಾಗಿದೆ. ಅಪಾಯವನ್ನು ನಿರ್ವಹಿಸಲು ನಾಲ್ಕು ಮೂಲ ತಂತ್ರಗಳಿವೆ : ತಗ್ಗಿಸುವಿಕೆ(mitigation), ಸ್ಥಾನಾಂತರ (transference), ಸ್ವೀಕಾರ (acceptance) ಮತ್ತು ತಪ್ಪಿಸುವಿಕೆ (avoidance.)

ತಗ್ಗಿಸುವಿಕೆ (Mitigation)

ತಗ್ಗಿಸುವಿಕೆಯು ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾಗಿ ಪರಿಗಣಿಸಲಾಗುವ ಅಪಾಯ ನಿರ್ವಹಣಾ ತಂತ್ರವಾಗಿದೆ. ಇದು ದೋಷವನ್ನು ಸರಿಪಡಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ ಅಥವಾ ನ್ಯೂನ್ಯತೆಗೆ ಸಂಬಂಧಿಸಿದ ಪ್ರಭಾವ ಅಥವಾ ಸಂಭವನೀಯತೆಗಳಿಗೆ ಕೆಲವು ರೀತಿಯ ಪರಿಹಾರ ನಿಯಂತ್ರಣಗಳನ್ನು ಒದಗಿಸುವುದು ಆಗಿದೆ. ಒಂದು ಸಾಮಾನ್ಯ ತಾಂತ್ರಿಕ ಭದ್ರತಾ ದೋಷವನ್ನು ತಗ್ಗಿಸಲು, ಮಾರಾಟಗಾರರಿಂದ ಒದಗಿಸಲಾದ ಪ್ಯಾಚನ್ನು ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವುದು ಆಗಿದೆ. ಕೆಲವೊಮ್ಮೆ ತಗ್ಗಿಸುವಿಕೆಯ ತಂತ್ರಗಳ ನಿರ್ಧರಿಸುವಿಕೆಯ ಪ್ರಕ್ರಿಯೆಯನ್ನು ನಿಯಂತ್ರಣ ವಿಶ್ಲೇಷಣೆ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

© NIMI NOT TO BE REPUBLISHED

ವರ್ಗಾವಣೆ (Transference)

ವರ್ಗಾವಣೆ ಎಂದರೆ ಇನ್ನೊಂದು ಪಕ್ಷಕ್ಕೆ ನಿಮ್ಮ ಪರವಾಗಿ ಅಪಾಯವನ್ನು ಸ್ವೀಕರಿಸಲು ಅವಕಾಶ ನೀಡುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಇದನ್ನು IT ಸಿಸ್ಟಮಗಳಲ್ಲಿ ವ್ಯಾಪಕವಾಗಿ ಮಾಡಲಾಗಿಲ್ಲ, ಆದರೆ ಪ್ರತಿಯೊಬ್ಬರೂ ತಮ್ಮ ವೈಯಕ್ತಿಕ ಜೀವನದಲ್ಲಿ ಸರ್ವಕಾಲಿಕವಾಗಿ ಇದನ್ನು ಮಾಡುತ್ತಾರೆ. ಕಾರು, ಆರೋಗ್ಯ ಮತ್ತು ಜೀವ ವಿಮೆ ಇವೆಲ್ಲವೂ ಅಪಾಯವನ್ನು ವರ್ಗಾಯಿಸುವ ಮಾರ್ಗಗಳಾಗಿವೆ. ಈ ಸಂದರ್ಭಗಳಲ್ಲಿ, ಅಪಾಯವು ವ್ಯಕ್ತಿಯಿಂದ ವಿಮಾ ಕಂಪನಿಯನ್ನೂ ಸೇರಿದಂತೆ ವಿಮಾದಾರರ ಪೂಲ್‌ಗೆ ವರ್ಗಾಯಿಸಲ್ಪಡುತ್ತದೆ. ಇದು ಯಾವುದೇ ದೋಷದ ಸಂಭವನೀಯತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುವುದಿಲ್ಲ ಅಥವಾ ಸರಿಪಡಿಸುವುದಿಲ್ಲ., ಆದರೆ ಇದು ಸಂಸ್ಥೆಯ ಮೇಲೆ ಒಟ್ಟಾರೆಯಾಗಿ ಪ್ರಭಾವವನ್ನು (ಪ್ರಾಥಮಿಕವಾಗಿ ಆರ್ಥಿಕ) ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

ಸ್ವೀಕಾರ (Acceptance)

ಅಂಗೀಕಾರವು ವ್ಯವಸ್ಥೆಯನ್ನು ತಿಳಿದಿರುವ ಅಪಾಯದೊಂದಿಗೆ ಕಾರ್ಯನಿರ್ವಹಿಸಲು ಅನುಮತಿಸುವ ಒಂದು ಸರಳ ಅಭ್ಯಾಸವಾಗಿದೆ. ಅನೇಕ ಕಡಿಮೆ ಅಪಾಯಗಳನ್ನು ಸರಳವಾಗಿ ಸ್ವೀಕರಿಸಲಾಗುತ್ತದೆ. ತಗ್ಗಿಸಲು ಅತ್ಯಂತ ಹೆಚ್ಚಿನ ವೆಚ್ಚವನ್ನು ಹೊಂದಿರುವ ಅಪಾಯಗಳನ್ನೂ ಸಹ ಸ್ವೀಕರಿಸಲಾಗುತ್ತದೆ. ಮ್ಯಾನೇಜಮೆಂಟ್‌ನಿಂದ ಸ್ವೀಕರಿಸಲಾಗುವ ಉನ್ನತ ಮಟ್ಟದ ಅಪಾಯಗಳ ಬಗ್ಗೆ ಎಚ್ಚರದಿಂದಿರಿ. ಈ ತಂತ್ರಗಳು ಬರವಣಿಗೆಯಲ್ಲಿವೆ ಮತ್ತು ಮ್ಯಾನೇಜರ್(ಗಳು) ನಿಂದ ನಿರ್ಧಾರ ಮಾಡುವ ಮೂಲಕ ಸ್ವೀಕರಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ. ಸಾಮಾನ್ಯವಾಗಿ ಅಂಗೀಕಾರ ಹೊಂದಿರದ ಅಪಾಯಗಳನ್ನು ಸ್ವೀಕರಿಸಲಾಗುತ್ತದೆ, ಮತ್ತು ಯಾವಾಗ ಅಪಾಯವು ಒಳನುಗ್ಗುತ್ತದೆಯೋ ಆಗ IT ಭದ್ರತಾ ಸಿಬ್ಬಂದಿಯನ್ನು ಹೊಣೆಗಾರರನ್ನಾಗಿ ಮಾಡಲಾಗುತ್ತದೆ. ವಿಶೇಷವಾಗಿ ವ್ಯಾಪಾರ ವ್ಯವಸ್ಥಾಪಕರು, IT ಭದ್ರತಾ ಸಿಬ್ಬಂದಿ ಇವರು ಸಂಸ್ಥೆಯ ಪರವಾಗಿ ಅಪಾಯವನ್ನು ಸ್ವೀಕರಿಸಲು ಅಧಿಕಾರ ಹೊಂದಿರುವವರು ಆಗಿರುತ್ತಾರೆ.

ತಪ್ಪಿಸುವಿಕೆ (Avoidance)

ತಪ್ಪಿಸುವಿಕೆಯು ಒಂದು ವ್ಯವಸ್ಥೆಯ ದುರ್ಬಲ ಅಂಶವನ್ನು ತೆಗೆದುಹಾಕುವ ಅಭ್ಯಾಸವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದ ಸಮಯದಲ್ಲಿ, ಒಂದು ವೆಬ್‌ಸೈಟ್ ಮಾರಾಟಗಾರರಿಗೆ ತಮ್ಮ ಇನ್‌ವೈಸ್‌ಗಳನ್ನು, ಮಾರಾಟಗಾರರ ಐಡಿಯನ್ನು HTML ಫೈಲ್‌ಹೆಸರಿನೊಂದಿಗೆ ಎಂಬೆಡ್ ಮಾಡಿ ಅದನ್ನು ಗುರುತಿಸುವಿಕೆಗಾಗಿ ಉಪಯೋಗಿಸಿಕೊಂಡು, ವೀಕ್ಷಿಸಲು ಅವಕಾಶ ಮಾಡಿಕೊಡುತ್ತಿತ್ತು. ಮತ್ತು ಮಾರಾಟಗಾರನಿಗೆ ಯಾವುದೇ ದೃಢೀಕರಣ ಅಥವಾ ಅಧಿಕಾರವೂ ಇರಲಿಲ್ಲ. ಸಂಸ್ಥೆಗೆ ಈ ವೆಬ್ ಪುಟಗಳ ಮತ್ತು ಅದರ ಅಪಾಯದ ಸೂಚನೆ ಸಿಕ್ಕಾಗ ಮ್ಯಾನೇಜಮೆಂಟ್ ಈ ವೆಬ್ ಪುಟಗಳನ್ನು ತೆಗೆದುಹಾಕಲು ಮತ್ತು ಮಾರಾಟಗಾರರಿಗೆ ಮತ್ತೊಂದು ಕಾರ್ಯವಿಧಾನದ ಮೂಲಕ ಇನ್‌ವೈಸ್‌ಗಳನ್ನು ಒದಗಿಸಲು ನಿರ್ಧಾರ ಮಾಡಿತು. ಈ ಪ್ರಕರಣದಲ್ಲಿ, ದೋಷಪೂರಿತ ವೆಬ್ ಪುಟಗಳನ್ನು ತೆಗೆದುಹಾಕುವ ಮೂಲಕ ಅಪಾಯವನ್ನು ತಪ್ಪಿಸಲಾಯಿತು.

ಅಪಾಯ ಮೌಲ್ಯಮಾಪನದ ವಿಧಗಳು (Types of Risk Assessment)

ಪರಿಮಾಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನ (Quantitative Risk Assessment)

ಪರಿಮಾಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನವು ಹಣಕಾಸು ಸಂಸ್ಥೆಗಳು ಮತ್ತು ವಿಮಾ ಕಂಪನಿಗಳು ಬಳಸುವ ವಿಧಾನಗಳಾಗಿವೆ. ಮಾಹಿತಿಗೆ, ವ್ಯವಸ್ಥೆಗಳಿಗೆ, ವ್ಯವಹಾರ ಪ್ರಕ್ರಿಯೆಗಳಿಗೆ, ಚೇತರಿಕೆ ವೆಚ್ಚಗಳು ಮುಂತಾದವುಗಳಿಗೆ ಮೌಲ್ಯಗಳನ್ನು ನಿಯೋಜಿಸುವ ಮೂಲಕ ಮೌಲ್ಯಮಾಪನ ಮಾಡಲಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ಪರಿಣಾಮ ಮತ್ತು ಅಪಾಯ ಇವುಗಳನ್ನು ನೇರ ಮತ್ತು ಅಪರೋಕ್ಷ ವೆಚ್ಚಗಳ ಪರಿಭಾಷೆಯಲ್ಲಿ ಅಳೆಯಲಾಗುತ್ತದೆ.

ಆದರೆ ಸಾಮಾನ್ಯವಾಗಿ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ, ಅಪಾಯವನ್ನು ಅಳೆಯಲು ಇದನ್ನು ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ. ಏಕೆಂದರೆ :

1 ಆಸ್ತಿಗಳಿಗೆ ಮೌಲ್ಯವನ್ನು ಗುರುತಿಸುವುದು ಮತ್ತು ನಿಯೋಜಿಸುವಲ್ಲಿರುವ ತೊಂದರೆಗಳು ಮತ್ತು

2 ಆವರ್ತನವನ್ನು ನಿರ್ಧರಿಸಲು ಇರುವ ಸಾಧ್ಯತೆಯ ಅಂಕಿಂಶಗಳ ಮಾಹಿತಿಯ ಕೊರತೆ.

ಹೀಗಾಗಿ, ಇವತ್ತಿನ ದಿನಗಳಲ್ಲಿ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಹೆಚ್ಚಿನ ಅಪಾಯ ಮೌಲ್ಯಮಾಪನದ ಸಾಧನಗಳು ಗುಣಾತ್ಮಕ ಅಪಾಯಗಳ ಅಳತೆ ಮಾಪನಗಳಾಗಿವೆ.

ಗುಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನ (Qualitative Risk Assessment)

ಗುಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನಗಳು, ಸಂಭವನೀಯತೆಯಲ್ಲಿ ಮತ್ತು ಪ್ರಭಾವದ ಮೌಲ್ಯಗಳಲ್ಲಿ ಈಗಾಗಲೇ ಹೆಚ್ಚಿನ ಮಟ್ಟದ ಅನಿಶ್ಚಿತತೆ ಇದೆ ಎಂದು ಊಹಿಸಿಕೊಳ್ಳುತ್ತವೆ ಮತ್ತು ಹೀಗಾಗಿ ಅಪಾಯವನ್ನು ವೈಯಕ್ತಿಕವಾಗಿ ಅಥವಾ ಗುಣಾತ್ಮಕವಾಗಿಸುತ್ತದೆ. ಪರಿಮಾಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದಲ್ಲಿ ಇರುವ ಸಮಸ್ಯೆಯಂತೆಯೆ, ಗುಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನದಲ್ಲಿಯೂ ದೊಡ್ಡ ತೊಂದರೆ ಎಂದರೆ ಸಂಭವನೀಯತೆಯನ್ನು ಮತ್ತು ಪ್ರಭಾವದ ಮೌಲ್ಯವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು.

ಇದಲ್ಲದೇ, ಈ ಮೌಲ್ಯಗಳನ್ನು ಬಹು ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುವ ಅದೇ ಮಾಪಕಗಳನ್ನು ಅನುಮತಿಸುವ ಅದೇ ವಿಧಾನದಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸುವ

NOT TO BE REPUBLISHED © NIMI

ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಗುಣಾತ್ಮಕ ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನಗಳು ಸಾಮಾನ್ಯವಾಗಿ "ಹೆಚ್ಚು", "ಮಾಧ್ಯಮ", ಮತ್ತು "ಕಡಿಮೆ" ಅಪಾಯದ ಫಲಿತಾಂಶಗಳನ್ನು ನೀಡುತ್ತವೆ. ಅದಾಗ್ಯೂ, ಪ್ರಭಾವ ಮತ್ತು ಸಂಭವನೀಯತೆಯ ವ್ಯಾಖ್ಯಾನದ ಟೇಬಲ್‌ಗಳನ್ನು ನೀಡುವ ಮೂಲಕ ಮತ್ತು ಪ್ರಭಾವದ ವಿವರಣೆಯ ಮೂಲಕ ಸಂಸ್ಥೆಯ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್‌ಗೆ ಮೌಲ್ಯಮಾಪನವನ್ನು ಸಮರ್ಪಕವಾಗಿ ತಿಳಿಸಲು ಸಾಧ್ಯವಾಗುತ್ತದೆ.

ಅಪಾಯದ ಮೌಲ್ಯಮಾಪನವು ಕೆಳಗಿನ ಕ್ರಮಗಳನ್ನು ಮತ್ತು ಚಟುವಟಿಕೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ :

- ಆಸ್ತಿಗಳ ಗುರುತಿಸುವಿಕೆ;
- ಗುರುತಿಸಲಾದ ಆಸ್ತಿಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ, ಕಾನೂನು ಮತ್ತು ವ್ಯಾಪಾರದ ಅವಶ್ಯಕತೆಗಳ ಗುರುತಿಸುವಿಕೆ
- ಗುರುತಿಸಲಾದ ಆಸ್ತಿಗಳ ಮೌಲ್ಯಮಾಪನ , ಗುರುತಿಸಲಾದ ಕಾನೂನು ಮತ್ತು ವ್ಯಾಪಾರದ ಅವಶ್ಯಕತೆಗಳನ್ನು ಗಣನೆಗೆ ತೆಗೆದುಕೊಳ್ಳುವುದು.
- ವ್ಯಾಪಾರದ ಅವಶ್ಯಕತೆ ಮತ್ತು ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆಯ ನಷ್ಟಗಳ ಪರಿಣಾಮಗಳು
- ಗುರುತಿಸಲಾದ ಆಸ್ತಿಗಳಿಗೆ ಗಮನಾರ್ಹ ಬೆದರಿಕೆಗಳ ಗುರುತಿಸುವಿಕೆ, ಮತ್ತು ಗುರುತಿಸಲಾದ ಆಸ್ತಿಯ ದುರ್ಬಲತೆಗಳು
- ಬೆದರಿಕೆಗಳ ಸಂಭವನೀಯತೆ ಮತ್ತು ಸಂಭವಿಸುವ ದುರ್ಬಲತೆಗಳ ಮೌಲ್ಯಮಾಪನ.
- ಅಪಾಯದ ಲೆಕ್ಕಾಚಾರ
- ಪೂರ್ವನಿರ್ಧರಿತ ಅಪಾಯದ ಪ್ರಮಾಣದ ವಿರುದ್ಧ ಅಪಾಯಗಳ ಮೌಲ್ಯಮಾಪನ.

ICT **ಪೂರೈಕೆ ಸರಪಳಿಗೆ ಅಪಾಯಗಳು** (Risks to the ICT supply chain management)

ICT ಪೂರೈಕೆ ಸರಪಳಿಗೆ ಅಪಾಯಗಳು, ಮಾಹಿತಿಯ ಅಥವಾ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಯ ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ ಅಥವಾ ಲಭ್ಯತೆಯ ನಷ್ಟದಿಂದ ಉಂಟಾಗುತ್ತವೆ. ಮತ್ತು ಸಾಂಸ್ಥಿಕ ಕಾರ್ಯಾಚರಣೆಗಳ (ಮಿಷನ್, ಫಂಕ್ಷನ್‌ಗಳು, ಚಿತ್ರ, ಅಥವಾ ಖ್ಯಾತಿ), ಸಾಂಸ್ಥಿಕ ಆಸ್ತಿಗಳ, ವ್ಯಕ್ತಿಗಳ ಮತ್ತು ಇತರ ಸಂಸ್ಥೆಗಳ ಮೇಲೆ ಸಂಭಾವ್ಯ ಪ್ರತಿಕೂಲ ಪರಿಣಾಮಗಳನ್ನು ಪ್ರತಿಬಿಂಬಿಸುತ್ತವೆ. ಇಂತಹ ಅಪಾಯಗಳು - ಅವ್ಯವಸ್ಥೆಯ ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿ, ಉತ್ಪನ್ನಗಳು ರವಾನೆಯಾಗುವ ಮೊದಲು ಅಸಮರ್ಪಕ ಪರೀಕ್ಷೆ, ಅಧಿಕೃತ ಅಲ್ಲದಿದ್ದರೂ ಕನಿಷ್ಟ ಅಗ್ಗದ ಭಾಗಗಳ ಬಳಕೆ ಮುಂತಾದವುಗಳ ರೂಪವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತವೆ.

Supply chain risk management (SCRM), ಇದು ಅಪಾಯಗಳನ್ನು, ಅವುಗಳ ವ್ಯಾಪಾರ ಪರಿಣಾಮಗಳನ್ನು ಮತ್ತು ಅವುಗಳನ್ನು ನಿರ್ವಹಿಸುವದನ್ನು ಅರ್ಥಮಾಡಿಕೊಳ್ಳುವ ಒಂದು ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಇದು ಪೂರೈಕೆ ಸರಪಳಿ ವ್ಯವಸ್ಥೆಯ ಜೀವನಚಕ್ರದುದ್ದಕ್ಕೂ ದೌರ್ಬಲ್ಯಗಳನ್ನು ಮತ್ತು ದುರ್ಬಲತೆಗಳನ್ನು ತಗ್ಗಿಸುವ ಮೂಲಕ ಅವುಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಪರಿಣಾಮಕಾರಿ ICT SCRM ಗೆ ಎಲ್ಲ ICT ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ SCRM ತತ್ವಗಳನ್ನು ಅನ್ವಯಿಸಲು ಸಂಸ್ಥೆಗಳಿಗೆ ಅನುಮತಿಸುವ ಪ್ರಕ್ರಿಯೆಗಳು, ಕಾರ್ಯವಿಧಾನಗಳು, ಮತ್ತು ಸಾಧನಗಳ ಅಗತ್ಯವಿದೆ. ಇಂತಹ ಒಂದು ತತ್ವವೆಂದರೆ ನಕಲಿ ಭಾಗಗಳ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುವುದು. ಏಕೆಂದರೆ ಅವು ಅನಿರೀಕ್ಷಿತ ನಡುವಳಿಕೆಗೆ, ಆರಂಭಿಕ ವೈಫಲ್ಯಗಳಿಗೆ, ಅಥವಾ ಕೆಟ್ಟದ್ದಕ್ಕೆ

ಕಾರಣವಾಗುತ್ತವೆ. ಅದ್ದರಿಂದ ನಕಲಿ ಭಾಗಗಳನ್ನು ಅಧಿಕೃತ ಭಾಗಗಳಿಂದ ಪ್ರತ್ಯೇಕಿಸುವುದು ಅಗತ್ಯವಾಗುತ್ತದೆ.

ಈ ಗುಣಲಕ್ಷಣಗಳನ್ನು ವ್ಯಕ್ತಪಡಿಸಲು ಒಂದು ರಚನಾತ್ಮಕ ಭಾಷೆಯ ಅಗತ್ಯವಿದೆ. ಪೂರೈಕೆ ಸರಪಳಿಯ ಎಲ್ಲಾ ಸದಸ್ಯರು ಅವುಗಳ ಬಗ್ಗೆ ಸಂವಹನ ನಡೆಸಬಹುದು ಮತ್ತು ಇದನ್ನು ನಕಲಿಗಳ ಬಗ್ಗೆ ಎಚ್ಚರಿಸಲು ಉಪಯೋಗಿಸಬಹುದು ಮತ್ತು ಕಾನೂನುಬದ್ಧ ವಸ್ತುಗಳಿಗಾಗಿ ಮಾನದಂಡಗಳನ್ನು ವ್ಯಕ್ತಪಡಿಸಬಹುದು. ಕಾನೂನುಬದ್ಧ ಮತ್ತು ನ್ಯಾಯಸಮ್ಮತವಲ್ಲದ ಎರಡೂ ಘಟಕಗಳ ಗಮನಿಸಬಹುದಾದ ಗುಣಲಕ್ಷಣಗಳನ್ನು ವಿವರಿಸುವ ರಚನಾತ್ಮಕ ಭಾಷೆಯು ಪೂರೈಕೆ

© NIMI NOT TO BE REPUBLISHED

ಸರಪಳಿ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಒಂದು ಸಾಧನವಾಗಿದೆ.

ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳು (Directory services)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಡೈರೆಕ್ಟರಿ ಮತ್ತು ಡೈರೆಕ್ಟರಿ ಸೇವೆಯನ್ನು ವಿವರಿಸಿ**
- **ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳ ಉಪಯೋಗಗಳನ್ನು ವಿವರಿಸಿ**
- **ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳ ವಿವಿಧ ಅನುಷ್ಠಾನಗಳನ್ನು ಉಲ್ಲೇಖಿಸಿ**
- **ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ತಾರ್ಕಿಕ ಮತ್ತು ಭೌತಿಕ ರಚನೆಯನ್ನು ವಿವರಿಸಿ**
- **ಜಾಗತಿಕ ಕ್ಯಾಟಲಾಗ್ ಮತ್ತು ಗ್ರೂಪ್ ಪಾಲಿಸಿಯನ್ನು ವಿವರಿಸಿ.**

ಪರಿಚಯ (Introduction)

ಡೈರೆಕ್ಟರಿ ಮತ್ತು ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳು (Directories and directory services)

ಡೈರೆಕ್ಟರಿಯು ಡಾಟಾದ ಸಂಗ್ರಹಣೆ ಅಥವಾ ಡಾಟಾದ ಪಟ್ಟಿಯಾಗಿದೆ. ನೈಜ ಪ್ರಪಂಚದ ಉದಾಹರಣೆಗಳು ಎಂದರೆ ದೂರವಾಣಿ ಪುಸ್ತಕಗಳು, ಭೂ ನೋಂದಣಿಗಳು, ಮತ್ತು ಕೆಲಸಗಳ ಪಟ್ಟಿಗಳಾಗಿವೆ. ಈ ಎಲ್ಲ ಉದಾಹರಣೆಗಳು ಮಾಹಿತಿಯನ್ನು ಸಂರಕ್ಷಿಸುವ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿವೆ. ಮತ್ತು ಸಂಬಂಧಪಟ್ಟ ವ್ಯಕ್ತಿಗಳ ಬೇಡಿಕೆಯ ಮೇಲೆ ಅವು ಲಭ್ಯವಾಗುವಂತೆ ಮಾಡುವುದಾಗಿದೆ.

ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನದಲ್ಲಿ ಡೈರೆಕ್ಟರಿ ಎಂಬ ಪದವನ್ನು ವಿಶೇಷ ರೀತಿಯ ಡಾಟಾ ಸಂಗ್ರಹಣೆಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ರಚನಾತ್ಮಕ ಸಂಗ್ರಹಣೆಯನ್ನು ಮತ್ತು ನೈಜ ಪ್ರಪಂಚದಿಂದ ಪಡೆಯಲಾದ ವಸ್ತುಗಳ ಸಮರ್ಥ ಮರುಪಡೆಯುವಿಕೆಯನ್ನು ಅನುಮತಿಸುತ್ತದೆ (ಉದಾಹರಣೆಗೆ : ವ್ಯಕ್ತಿಗಳು, IT ಉಪಕರಣ). ಈ ಸಂಗ್ರಹಣೆಯ ಮುಖ್ಯ ಲಕ್ಷಣವೆಂದರೆ ಎಲ್ಲ ಡಾಟಾವನ್ನು ನಮೂದುಗಳಲ್ಲಿ (entries) ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ. ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ ಎಂಟ್ರಿಗಳ ಗುಂಪು ಟ್ರೀ (tree) ಯನ್ನು ರೂಪಿಸುತ್ತದೆ (hierarchical database).

ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳು (Directory Services)

ಡೈರೆಕ್ಟರಿ ಸೇವೆಯು ಸಾಮಾನ್ಯ ವಸ್ತುಗಳ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಸಂಪನ್ಮೂಲಗಳ ಪತ್ತೆಗೆ, ನಿರ್ವಹಣೆಗೆ, ಆಡಳಿತಕ್ಕೆ, ಮತ್ತು ಸಂಘಟನೆಗೆ ಬೇಕಾದ ಮಾಹಿತಿ ಹಂಚಿಕೆಯ ಮೂಲಸೌಕರ್ಯವಾಗಿದೆ. ಇವು ಸಂಪುಟಗಳು, ಫೋಲ್ಡರ್‌ಗಳು, ಫೈಲ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು, ಬಳಕೆದಾರರು, ಗುಂಪುಗಳು, ಸಾಧನಗಳು, ದೂರವಾಣಿ ಸಂಖ್ಯೆಗಳು ಮತ್ತು ಇತರ ವಸ್ತುಗಳನ್ನು ಒಳಗೊಂಡಿರಬಹುದು. ಡೈರೆಕ್ಟರಿ ಸೇವೆಯು, ಉತ್ತಮವಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಇಂಟರ್‌ಫೇಸ್ ಮೂಲಕ ಬಳಕೆದಾರರಿಗೆ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ಮಾಹಿತಿಯ ಪ್ರವೇಶವನ್ನು ನೀಡುವ ಒಂದು ಪರಿಹಾರವಾಗಿದೆ. ಒಂದು ವೇಳೆ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸಿದರೆ, ಈ ಉದ್ದೇಶಕ್ಕಾಗಿ ಸೂಕ್ತವಾದ ಪ್ರೊಟೋಕಾಲನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಬೇಕಾಗುತ್ತದೆ. ಇಂತಹ ಒಂದು ಪ್ರೊಟೋಕಾಲ್ ಎಂದರೆ Light weight Directory Access Protocol(LDAP).

ಡೈರೆಕ್ತರಿ ಸೇವೆಗಳನ್ನು ಬಳಸುವ ಪ್ರಯೋಜನಗಳು (Benefits of using directory services) :

1 **ಸಂಪನ್ಮೂಲ ನಿರ್ವಹಣೆ** (Resource management) : ನೆಟ್‌ವರ್ಕ್ ನಿರ್ವಾಹಕರು, ನೆಟ್‌ವರ್ಕ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಒಂದೇ ಉಪಕರಣವನ್ನು ಬಳಸಬಹುದು. -ಬಳಕೆದಾರರ ಖಾತೆಗಳು, ಸರ್ವರ್‌ಗಳು, ಡ್ರೈವ್‌ಗಳು, ಫೈಲ್‌ಗಳು, ಪ್ರಿಂಟರ್‌ಗಳು, ಇತ್ಯಾದಿ. ಇವುಗಳನ್ನು ಸರಳ ಟ್ರೀ ರಚನೆಯಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲಾಗುತ್ತದೆ. ಆದ್ದರಿಂದ ಅವುಗಳನ್ನು ಪತ್ತೆಹಚ್ಚಲು ಮತ್ತು ನಿರ್ವಹಿಸಲು ಸುಲಭವಾಗುತ್ತದೆ. ಈ ರಚನೆಯಲ್ಲಿ, ಪ್ರತಿಯೊಂದು ಸಂಪನ್ಮೂಲವನ್ನು ಆಬ್ಜೆಕ್ಟ್ ಎನ್ನುವ ಒಂದು ಐಕನ್‌ನಿಂದ ತೋರಿಸಲಾಗುತ್ತದೆ. ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡುವ ಮೂಲಕ ಅದರ ಸೆಟ್ಟಿಂಗ್‌ಗಳು ಲಭ್ಯವಾಗುತ್ತವೆ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ನಿರ್ವಾಹಕರು ತಮಗೆ ಸರಿ ಹೊಂದುವಂತೆ ಅವುಗಳನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು.

2 **ಬಳಕೆದಾರರು** (Users) : ನಿಮಗೆ ಹಕ್ಕು ಇರುವ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ನೀವು ಒಂದೇ ಲಾಗ್ ಆನ್ ಅನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಪ್ರತಿಯೊಂದು ಸರ್ವರ್‌ಗೆ ಲಾಗ್ ಇನ್ ಆಗುವ ಅಥವಾ ಪ್ರತಿಯೊಂದು ಪ್ರಿಂಟರ್‌ನ್ನು ಅಥವಾ ಇತರ ಸಾಧನವನ್ನು ದೃಢೀಕರಿಸುವದಕ್ಕಿಂತ ಹೆಚ್ಚಾಗಿ ಡೈರೆಕ್ಟರಿ ಈ ಮಾಹಿತಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ನೀವು ಒಮ್ಮೆ ಲಾಗ್ ಆನ್ ಆಗಿರಿ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ನಿರ್ವಾಹಕರು ನಿಮಗೆ ನೀಡಿರುವ ಹಕ್ಕಿನ ಮೂಲಕ ಯಾವುದೇ ಕೆಲಸವನ್ನು ಮಾಡಬಹುದು.

3 **ಭದ್ರತೆ** (Security) : ಕೇವಲ ಒಂದೇ ಡೋಮೇನ್ ಅನ್ನು ಹೊಂದಿರುವುದು ಎಂದರೆ ಒಂದೇ ಭದ್ರತಾ ನೀತಿ ಮತ್ತು ಒಂದೇ ನಿರ್ವಾಹಕರ (administrator) ಗುಂಪಿನ ಮೂಲಕ ಉತ್ತಮ ಭದ್ರತೆಯನ್ನು ಹೊಂದುವುದಾಗಿದೆ. ನೀವು ಅನೇಕ ಡೊಮೇನ್‌ಗಳನ್ನು ಮತ್ತು ಫಾರೆಸ್ಟ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಪ್ರತಿಯೊಂದು ತನ್ನದೇ ಆದ ಸ್ವಂತ ನಿರ್ವಾಹಕರನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಒಂದು ದುರ್ಬಲ ಆದರೆ ವಿಶ್ವಾಸಾರ್ಹ ಡೊಮೇನ್, ಇತರ ಎಲ್ಲಾ ಫಾರೆಸ್ಟ್‌ಗಳನ್ನು

ಮತ್ತು ಡೊಮೇನ್‌ಗಳನ್ನು ಬಹಿರಂಗಪಡಿಸುತ್ತದೆ. ಒಂದೇ ಡೊಮೇನ್‌ನಿಂದ, ಸಂಸ್ಥೆಯ ಭದ್ರತಾ ನೀತಿಯನ್ನು ಜಾರಿಗೊಳಿಸಲು ತುಂಬಾ ಸುಲಭವಾಗುತ್ತದೆ.

4 **ಒಂದೇ ವೇದಿಕೆ** (Single platform) : ಒಂದೇ ಡೈರೆಕ್ಟರಿ ಸೇವೆ ಅಥವಾ ಗ್ಲೋಬಲ್ ಕ್ಯಾಟಲಾಗ್ (GC) ಎಂದರೆ ಇತರ ಎಲ್ಲ ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳಿಗೆ ಒಂದೇ ವೇದಿಕೆ ಇರುತ್ತದೆ. ಇದು ಮೇಲ್ವಿಚಾರಣೆ ಮತ್ತು ಸಂದೇಶ ಕಳುಹಿಸುವಿಕೆಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

5 **ವೇಗದ ನಿಯೋಜನೆ** (Faster deployment): ಇದು ಕೇವಲ ಒಂದೇ ಡೋಮೇನ್‌ನೊಂದಿಗೆ ಮತ್ತು ಹಂಚಿಕೆಯ ಖಾತೆಯ ಡಾಟಾಬೇಸ್ ಪರಿಹಾರಗಳೊಂದಿಗೆ ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಇದನ್ನು ಒಮ್ಮೆ ಮಾತ್ರ ನಿಯೋಜಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಎಂದರೆ ಸಂಸ್ಥೆಯು ಅನೇಕ ಮತ್ತು ಪ್ರತ್ಯೇಕ ಡೊಮೇನ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಅದಕ್ಕಿಂತ ಕಂಪನಿಯಾದ್ಯಂತ ನಿಯೋಜನೆಗಳು ಹೆಚ್ಚು ವೇಗವಾಗಿರುತ್ತವೆ.

6 **ಒಂದೇ ನಿರ್ವಹಣೆ** (Single management): ಮೂಲಸೌಕರ್ಯ- ಒಂದೇ ನಿರ್ವಹಣೆಯ ಮೂಲಸೌಕರ್ಯವನ್ನು ಹೊಂದಿರುವುದು ಎಂದರೆ ಎಲ್ಲಾ ಇತರ ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳ ಕಾರ್ಯಗಳಿಗೆ ಒಂದೇ ಮೂಲಸೌಕರ್ಯ ಇರುವುದು. ಉದಾಹರಣೆಗೆ, ಸಾಫ್ಟ್‌ವೇರ್ ನಿಯೋಜನೆ, ದಾಸ್ತಾನು, ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್ ನಿರ್ವಹಣೆಯನ್ನು ಹಂಚುವಿಕ ಮತ್ತು ನಿಯೋಗ (ಉದಾಹರಣೆಗೆ ಬಳಕೆದಾರರ ಖಾತೆಗಳಿಗೆ)

7 **ಸಿಂಗಲ್ ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಕಂಟೇನರ್**(Single Group Policy container (GPC)): ಒಂದೇ GPC ಜೊತೆಗೆ, ಮ್ಯಾನೇಜಮೆಂಟ್ ನೀತಿಗಳನ್ನು ಒಮ್ಮೆ ಮಾತ್ರ ವ್ಯಾಖ್ಯಾನಿಸಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಇಡೀ ಎಂಟರ್‌ಪ್ರೈಸ್‌ನಾದ್ಯಂತ ಉಪಯೋಗಿಸಬಹುದು ಮತ್ತು ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು (GPOs) ಮ್ಯಾನುಯಲ್ ಆಗಿ ರಫ್ತು ಮತ್ತು ಆಮದು ಮಾಡಿಕೊಳ್ಳುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

8 **ಬ್ಯಾಕಪ್ ಮತ್ತು ರಿಕವರಿ** (Backup and recovery): ಕೇವಲ ಒಂದೇ ಡೋಮೇನ್ ಮಾತ್ರ ಹೊಂದಿರುವುದು ಎಂದರೆ, ಉತ್ತಮ ಸ್ಥಿತಿಸ್ಥಾಪಕತ್ವ ಏಕೆಂದರೆ ಪ್ರತಿ ಲೊಕೇಷನ್ ಸಂಪೂರ್ಣ ಡೊಮೇನ್ ಬ್ಯಾಕಪ್‌ನ್ನು ಹೊಂದಿರುವುದು.

9 **ಕಡಿಮೆ ಹಾರ್ಡ್‌ವೇರ್** (Less hardware): ಹಲವಾರು ಡೊಮೇನ್‌ಗಳನ್ನು ಹೊಂದಿರುವ ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿ, ಪ್ರತಿಯೊಂದು ಲೊಕೇಷನ್‌ಗೆ ಎರಡು ಡೊಮೇನ್ ಕಂಟ್ರೋಲರ್‌ಗಳ (DC) ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಒಂದೇ ಡೊಮೇನ್‌ನಲ್ಲಿ, ಪ್ರತಿಯೊಂದು ಲೊಕೇಷನ್‌ಗೆ ಕೇವಲ ಒಂದೇ DC ಯ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ ಏಕೆಂದರೆ ಸ್ಥಳೀಯ DC ವಿಫಲವಾದರೆ, ಲೊಕೇಷನ್‌ಗಳು ಹಬ್ DC ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಕಡಿಮೆ ಹಾರ್ಡ್‌ವೇರ್ ಎಂದರೆ ಕಡಿಮೆ ಲೈಸೆನ್ಸ್‌ಗಳು, ಕಡಿಮೆ ಸಾಫ್ಟ್‌ವೇರ್ ನಿರ್ವಹಣೆ, ಮತ್ತು ಕಡಿಮೆ ಸರ್ವರ್ ನಿರ್ವಹಣೆ. ರಿಮೋಟ್ DC ಗಳ ಬ್ಯಾಕಪ್ ಅವಶ್ಯಕತೆಯೂ ಸಹ ಇರುವುದಿಲ್ಲ. ಏಕೆಂದರೆ ರಿಮೋಟ್ DC ಗಳು ಕೇಂದ್ರಿಯ DC ಗಳು ಸಂಗ್ರಹಿಸಿರುವ ಮಾಹಿತಿಯನ್ನೇ ಸಂಗ್ರಹಿಸಿರುತ್ತವೆ. - DC ಗಳು ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳನ್ನು ಮಾತ್ರ ನಿರ್ವಹಿಸುತ್ತಿವೆ ಎಂದು ಊಹಿಸಿಕೊಳ್ಳಲಾಗುತ್ತದೆ.

ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳ ಅನುಷ್ಠಾನ (Implementations of directory services)

ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳು Open Systems Interconnection (OSI) ನ ಭಾಗವಾಗಿದ್ದವು. ಇದು ಉದ್ಯಮದಲ್ಲಿನ ಪ್ರತಿಯೊಬ್ಬರೂ ಸಾಮಾನ್ಯ ನೆಟ್‌ವರ್ಕ್ ಮಾನದಂಡಗಳನ್ನು ಒಪ್ಪಿಕೊಳ್ಳುವ ಪ್ರಾರಂಭವಾಗಿತ್ತು. ಇದು ಬಹು- ಮಾರಾಟಗಾರರ ಪರಸ್ಪರ ಕಾರ್ಯಸಾಧ್ಯತೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. 1980 ರ ದಶಕದಲ್ಲಿ International Telecommunication Union (ITU) ಮತ್ತು the International Organization for Sandardization (ISO), ಇವು ಡೈರೆಕ್ಟರಿ ಸರ್ವೀಸ್‌ಗಳಿಗಾಗಿ standards - X.500 ನ ಸೆಟ್‌ನೊಂದಿಗೆ ಬಂದವು. ಇದಕ್ಕೆ ಪ್ರೊಟೊಕಾಲ್ Light weight Directory Access Protocol, LDAP ಎಂದು ನಿರ್ಧರಿಸಲಾಯಿತು. ಇದು ಡೈರೆಕ್ಟರಿ ಮಾಹಿತಿ ಸರ್ವೀಸ್ ಆಫ್ X .500 ಯನ್ನು ಆಧರಿಸಿದೆ.

ಆದರೆ ಇದು TCP/IP ಸ್ಟ್ಯಾಕನ್ನು ಮತ್ತು ಸ್ಟ್ರಿಂಗ್ ಎನ್‌ಕೋಡಿಂಗ್ ಸ್ಕೀಮ್ ಆಫ್ X.500 ಪ್ರೊಟೊಕಾಲ್ DAP ಯನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

LDAP/X.500 ಆಧರಿತ ಅನುಷ್ಠಾನಗಳು ಯಾವವೆಂದರೆ :

- Active Directory : ವಿಂಡೊಸ್‌ನ ಮೈಕ್ರೊಸಾಫ್ಟ್ ಅಧುನಿಕ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯು, X.500 ಡೈರೆಕ್ಟರಿಯಿಂದ ಹುಟ್ಟಿಕೊಂಡಿದೆ. ಎಕ್ಸ್ ಚೇಂಜ್ ಸರ್ವರ್‌ನ ಬಳಕೆಗಾಗಿ ರಚಿಸಲಾಗಿದೆ. ಇದನ್ನು ಮೊದಲು ವಿಂಡೋಸ್ 2000 ಸರ್ವರ್ ಜೊತೆಗೆ ರವಾನಿಸಲಾಗಿದೆ ಮತ್ತು ವಿಂಡೊಸ್‌ನ ಸತತ ಆವೃತ್ತಿಗಳಿಂದ ಬೆಂಬಲಿಸಲ್ಪಟ್ಟಿದೆ.

- Apache Directory Server : ಜಾವಾದಲ್ಲಿ ಬರೆಯಲಾದ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯಾಗಿದೆ., LDAP, Kerberos 5 ಮತ್ತು Change Password Protocol ನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ, LDAPv3 ಯಿಂದ ಪ್ರಮಾಣೀಕರಿಸಲಾಗಿದೆ. ಅಪಾಚೆ ಡೈರೆಕ್ಟರಿ ಸರ್ವರ್ ಸಹ ಅಪಾಚೆ ಸಾಫ್ಟ್‌ವೇರ್ ಫೌಂಡೇಷನ್‌ನ ಉನ್ನತ ಮಟ್ಟದ ಯೋಜನೆಯಾಗಿದೆ.

- eDirectory : ಇದು ಡೈರೆಕ್ಟರಿ ಸೇವೆಯ NetIQs ನ ಅನುಷ್ಠಾನವಾಗಿದೆ. ಇದು ವಿಂಡೋಸ್, ನೆಟ್‌ವೇರ್, ಲಿನಕ್ಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್‌ನ ಹಲವಾರು ಫ್ಲೇವರ್‌ಗಳನ್ನು ಒಳಗೊಂಡಂತೆ

© NIMI NOT TO BE REPUBLISHED

ಅನೇಕ ರಚನೆಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಮತ್ತು ಇದನ್ನು ಬಳಕೆದಾರರ ಆಡಳಿತಕ್ಕಾಗಿ, ಸಂರಚನಾ ನಿರ್ವಹಣೆಗಾಗಿ, ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ನಿರ್ವಹಣೆಗಾಗಿ ದೀರ್ಘಕಾಲ ಬಳಸಲಾಗಿದೆ. ಗುರುತಿನ ನಿರ್ವಹಣಾ ಉತ್ಪನ್ನಗಳ ವ್ಯಾಪಕ ಶ್ರೇಣಿಯಲ್ಲಿ ಇ ಡೈರೆಕ್ಟರಿ ಕೇಂದ್ರ ಘಟಕವಾಗಿ ವಿಕಸನಗೊಂಡಿದೆ. ಹಿಂದೆ ಇದನ್ನು ನಾವೆಲ್ ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳೆಂದು ಕರೆಯಲಾಗುತ್ತಿತ್ತು.

- **Red Hat Directory Server :** AOL's Netscape Security Solutions ಘಟಕದಿಂದ ಪಡೆದುಕೊಂಡ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯನ್ನು ರೆಡ್ ಹ್ಯಾಟ್ ಬಿಡುಗಡೆ ಮಾಡಿತು. ಇದು ರೆಡ್ ಹ್ಯಾಟ್ ಡೈರೆಕ್ಟರಿ ಸರ್ವರ್ ಎಂದು ಕರೆಯಲ್ಪಡುವ ರೆಡ್ ಹ್ಯಾಟ್ ಎಂಟರ್‌ಪ್ರೈಸ್ ಲಿನಕ್ಸ್ ನಲ್ಲಿ ರನ್ ಆಗುವ ವಾಣಿಜ್ಯ ಉತ್ಪನ್ನವಾಗಿದೆ. ಮತ್ತು ಇದು ಸಮುದಾಯವು ಬೆಂಬಲಿಸುವ 389 ಡೈರೆಕ್ಟರಿ ಸರ್ವರ್ ಯೋಜನೆ ಆಗಿದೆ.
- Oracle Internet Directory: (OID) ಇದು ಒರ್ಯಾಕಲ್ ಕಾರ್ಪೋರೇಷನ್‌ನ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯಾಗಿದೆ. ಇದು iLDAP ಆವೃತ್ತಿ 3 ಕ್ಕೆ ಹೊಂದಿಕೆಯಾಗುತ್ತದೆ.
- **Sun Java System Directory Server:** ಇದು ಸನ್ ಮೈಕ್ರೊಸಿಸ್ಟಮ್‌ನ ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯ ಕೊಡುಗೆಯಾಗಿದೆ.
- OpenDS: ಇದು ಓಪನ್ ಸೋರ್ಸ್ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯಾಗಿದ. ಜಾವಾದಿಂದ ಅನುಷ್ಠಾನಗೊಂಡಿದೆ. ಸನ್ ಮೈಕ್ರೊಸಿಸ್ಟಮ್‌ನಿಂದ ಬೆಂಬಲಿತವಾಗಿದೆ.
- **IBM Tivoli Directory Server:** ಇದನ್ನು OpenLDAP ಯ ಹಳೆಯ ಬಿಡುಗಡೆಯನ್ನು ಕಸ್ಟಮೈಸ್ ಮಾಡಿ ಬಿಡುಗಡೆ ಮಾಡಲಾಗಿದೆ.
- Windows NT Directory Services (NTDS), ನಂತರ ಇದನ್ನು Active Directory ಎಂದು ಮರುಹೆಸರಿಸಲಾಯಿತು, ಹಿಂದಿನ NT ಡೊಮೇನ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.
- **OpenLDAP :** ಇದು ಯುನಿಕ್ಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್‌ನ ಎಲ್ಲ ಉತ್ಪನ್ನಗಳನ್ನು ಒಳಗೊಂಡಂತೆ, ಲಿನಕ್ಸ್, ವಿಂಡೋಸ್, z/OS ಮತ್ತು ಎಂಬೆಡೆಡ್/ರಿಯಲ್ ಟೈಮ್ ಸಿಸ್ಟಮ್‌ಗಳ ಎಲ್ಲಾ ಪ್ರಸ್ತುತ ಕಂಪ್ಯೂಟರ್ ರಚನೆಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ.

ಡೈರೆಕ್ಟರಿ ಸೇವೆಗಳನ್ನು ರಚಿಸಲು OpenLDAP ಮತ್ತು Kerberos protocol, ಮತ್ತು Samba ಸಾಫ್ಟ್‌ವೇರ್‌ನ್ನು ಒಳಗೊಂಡಂತೆ, ಸಾಕಷ್ಟು ಓಪನ್ ಸೋರ್ಸ್ ಸಾಧನಗಳಿವೆ.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ (Active Directory)

- ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ (AD) ಇದು ವಿಂಡೋಸ್ ಡೊಮೇನ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳಿಗಾಗಿ ಮೈಕ್ರೊಸಾಫ್ಟ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದ ಡೈರೆಕ್ಟರಿ ಸೇವೆಯಾಗಿದೆ. ಮತ್ತು ಹೆಚ್ಚಿನ ವಿಂಡೋಸ್ ಸರ್ವರ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಪ್ರಕ್ರಿಯೆಗಳ ಮತ್ತು ಸೇವೆಗಳ ಸೆಟ್ ಆಗಿ ಸೇರಿಸಲಾಗಿದೆ.
- AD ಡೊಮೇನ್ ನಿಯಂತ್ರಕವು, ವಿಂಡೋಸ್ ಡೊಮೇನ್ ಟೈಪ್ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ, ಎಲ್ಲ ಬಳಕೆದಾರರನ್ನು ಮತ್ತು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ದೃಢೀಕರಿಸುತ್ತದೆ ಮತ್ತು ಅಧಿಕಾರ ನೀಡುತ್ತದೆ. ಇದು ಎಲ್ಲ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಭದ್ರತಾ ನೀತಿಗಳನ್ನು ನಿಯೋಜಿಸುತ್ತದೆ ಮತ್ತು ಜಾರಿಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳನ್ನು ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ನವೀಕರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ವಿಂಡೋಸ್ ಡೊಮೇನ್‌ನ ಭಾಗವಾಗಿರುವ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಬಳಕೆದಾರರು ಲಾಗಿನ್ ಆದಾಗ ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ ಸಲ್ಲಿಸಲಾದ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಮತ್ತು ಬಳಕೆದಾರನು ಸಿಸ್ಟಮ್ ನಿರ್ವಾಹಕರೆ ಅಥವಾ ಸಾಮಾನ್ಯ ಬಳಕೆದಾರರೆ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ.
- ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯು, Lightweight Directory Access Protocol (LDAP) ಆವೃತ್ತಿ 2 ಮತ್ತು 3, ಮೈಕ್ರೊಸಾಫ್ಟ್‌ನ ಆವೃತ್ತಿ Kerberos, ಮತ್ತು DNS ಇವುಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.
- ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯು, ಬಳಕೆದಾರರ ನಿರ್ವಹಣೆಯನ್ನು ಸಹ ಸುಲಭಗೊಳಿಸುತ್ತದೆ. ಇದು ಈ ಎಲ್ಲ ಬಳಕೆದಾರರಿಗೆ ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ಸಂಬಂಧಿತ ಮಾಹಿತಿಗೆ ಒಂದೇ ಸಂಗ್ರಹಣೆಯ ಉಗ್ರಾಣವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.
- AD ಯು ತನ್ನ ಪ್ರವೇಶ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿ LDAP ಯನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.
- AD ತನ್ನ ಲೊಕೇಟರ್ ಸೇವೆಯಾಗಿ DNS ಅನ್ನು ಅವಲಂಬಿಸಿದೆ. ಇದು DNS ಕ್ವೆರಿಗಳ ಮೂಲಕ ಡೊಮೇನ್ ನಿಯಂತ್ರಕಗಳನ್ನು ಪತ್ತೆಹಚ್ಚಲು ಕ್ಲೈಂಟ್‌ಗಳನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ತಾರ್ಕಿಕ ರಚನೆ (Logical Structure of Active Directory)

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯು ವಿತರಣಾ ಡಾಟಾಬೇಸ್ ಆಗಿದ್ದು, ನೆಟ್‌ವರ್ಕ್ ಸಂಪನ್ಮೂಲಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ ಮತ್ತು ನಿರ್ವಹಿಸುತ್ತದೆ. ಹಾಗೆಯೆ ಡೈರೆಕ್ಟರಿ ಸಕ್ರೀಯಗೊಳಿಸಿದ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಂದ, ಅಪ್ಲಿಕೆಷನ್ - ನಿರ್ದಿಷ್ಟ ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯು ನೆಟ್‌ವರ್ಕ್‌ನ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಕ್ರಮಾನುಗತ ರಚನೆಯಾಗಿ ಸಂಘಟಿಸಲು ನಿರ್ವಾಹಕರಿಗೆ ಅನುಮತಿಯನ್ನು ನೀಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ : ಬಳಕೆದಾರರು, ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಸಾಧನಗಳು, ಮತ್ತು ಮುಂತಾದವುಗಳು.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯಲ್ಲಿ, ಸಂಪನ್ಮೂಲಗಳನ್ನು ತಾರ್ಕಿಕ ರಚನೆಯಲ್ಲಿ ಆಯೋಜಿಸಲಾಗಿದೆ. ಮತ್ತು ಈ ತಾರ್ಕಿಕ ಸಂಪನ್ಮೂಲಗಳ ಗುಂಪು ಒಂದು ಸಂಪನ್ಮೂಲವನ್ನು ಅದರ ಭೌತಿಕ ಲೊಕೇಷನ್‌ನಿಂದ ಕಂಡುಹಿಡಿಯುವ ಬದಲು ಅದರ ಹೆಸರಿನಿಂದ

ಕಂಡುಹಿಡಿಯಲಾಗುತ್ತದೆ.

AD ತಾರ್ಕಿಕ ರಚನೆಯ ಪ್ರಯೋಜನಗಳು (Benefits of AD Logical Structure)

- ತಾರ್ಕಿಕ ರಚನೆಯು ಹೆಚ್ಚಿನ ನೆಟ್‌ವರ್ಕ್ ಭದ್ರತೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಸಂಪನ್ಮೂಲಗಳಿಗೆ ನಿರ್ದಿಷ್ಟ ಗುಂಪುಗಳಿಗೆ (OU) ಮಾತ್ರ ಪ್ರವೇಶವನ್ನು ಒದಗಿಸುತ್ತದೆ.
- ತಾರ್ಕಿಕ ರಚನೆಯು, ನೆಟ್‌ವರ್ಕ್‌ನ ಆಡಳಿತ, ಸಂರಚನೆ, ಮತ್ತು ನಿಯಂತ್ರಣದ ಮೂಲಕ ನೆಟ್‌ವರ್ಕ್ ನಿರ್ವಹಣೆಯನ್ನು ಸರಳಗೊಳಿಸುತ್ತದೆ.
- ಡೊಮೇನ್‌ಗಳು ಮತ್ತು ಫಾರೆಸ್ಟ್‌ಗಳ ತಾರ್ಕಿಕ ರಚನೆಯ ನಡುವಿನ ಸಂಬಂಧವು ಒಂದು ಸಂಸ್ಥೆಯ ಸಂಪನ್ಮೂಲಗಳ ಹಂಚಿಕೆಯನ್ನು ಸರಳಗೊಳಿಸುತ್ತದೆ.
- ತಾರ್ಕಿಕ ರಚನೆಯು ಸರಳೀಕೃತ ನೆಟ್‌ವರ್ಕ್ ನಿರ್ವಹಣೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್ ಸಂಪನ್ಮೂಲಗಳ ಮೇಲಿನ ಲೋಡ್ ಅನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ ಮತ್ತು ಮಾಲೀಕತ್ವದ ಒಟ್ಟೂ ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

AD ತಾರ್ಕಿಕ ರಚನೆಯ ಘಟಕಗಳು (Components of AD Logical Structure)

ತಾರ್ಕಿಕ ರಚನೆಯ ಘಟಕಗಳು ಪರಸ್ಪರ ಸಂಬಂಧವನ್ನು ಹೊಂದಿವೆ. ಅದ್ದರಿಂದ ಇದು ಸಂಗ್ರಹಿಸಿದ ಡಾಟಾಗೆ ಪ್ರವೇಶದ ನಿಯಂತ್ರಣವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಮತ್ತು ಒಂದು ಫಾರೆಸ್ಟ್‌ನಲ್ಲಿ ವಿಭಿನ್ನ ಡೊಮೇನ್‌ಗಳ ನಡುವೆ ಡಾಟಾವನ್ನು ಹೇಗೆ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ ಎಂದು ಕಂಡುಕೊಳ್ಳುತ್ತದೆ.

- ಆಬ್ಜೆಕ್ಟ್‌ಗಳು : ಬಳಕೆದಾರರು, ಕಂಪ್ಯೂಟರ್, ಗ್ರೂಪ್, ಪ್ರಿಂಟರ್ ಇತ್ಯಾದಿಗಳು.
- ಸಂಸ್ಥಿಕ ಘಟಕಗಳು : ಯಾವುದೇ ಫೋಲ್ಡರ್‌ನಂತೆ ಆದರೆ ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ನಿಯಂತ್ರಣದಲ್ಲಿರುತ್ತದೆ.
- ಡೊಮೇನ್‌ಗಳು : ಆಬ್ಜೆಕ್ಟ್‌ಗಳಿಗೆ ತಾರ್ಕಿಕ ಗಡಿಗಳು
- ಟ್ರೀಗಳು : ಹಲವಾರು ಡೊಮೇನ್‌ಗಳಿಗೆ ತಾರ್ಕಿಕ ಗಡಿಗಳು
- ಫಾರೆಸ್ಟ್‌ಗಳು : ಹಲವಾರು ಟ್ರೀಗಳಿಗೆ ತಾರ್ಕಿಕ ಗಡಿಗಳು

ಒಟ್ಟಾರೆಯಾಗಿ, ಮೈಕ್ರೊಸಾಫ್ಟ್ ಡೊಮೇನ್ ನಿಯಂತ್ರಕವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಭೌತಿಕ ಯಂತ್ರವು, ಈ ಎಲ್ಲ ತಾರ್ಕಿಕ ವಿಭಾಗಗಳನ್ನು ನಿಯಂತ್ರಿಸಬಲ್ಲದು. ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಮೀಸಲಾದ 'A Operation Master' ಸಹಾಯದಿಂದ ನಿಯಂತ್ರಿಸಲಾಗುತ್ತದೆ.

ಫಾರೆಸ್ಟ್ ಉನ್ನತ ಮಟ್ಟದ ಕಂಟೇನರ ಆಗಿದೆ. ಫಾರೆಸ್ಟ್ ಎಂದರೆ ಒಂದೇ ಸಾಮಾನ್ಯ ಗ್ಲೊಬಲ್ ಕ್ಯಾಟಾಲಾಗ, ಡೈರೆಕ್ಟರಿ ಸ್ಕೀಮಾ, ತಾರ್ಕಿಕ ರಚನೆ, ಮತ್ತು ಡೈರೆಕ್ಟರಿ ಸಂರಚನೆ ಮುಂತಾದವುಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳುವ ಟ್ರೀಗಳ ಸಂಗ್ರಹವಾಗಿದೆ. ಬಳಕೆದಾರರು, ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಗ್ರೂಪ್‌ಗಳು ಮತ್ತು ಇತರ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಪ್ರವೇಶಿಸಬಲ್ಲ ಭದ್ರತಾ ಗಡಿಗಳನ್ನು ಫಾರೆಸ್ಟ್ ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

ಡೊಮೇನ್‌ಗಳು ಫಾರೆಸ್ಟ್‌ನಲ್ಲಿರುತ್ತವೆ. ಡೊಮೇನ್‌ನ್ನು, ನೆಟ್‌ವರ್ಕ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ (ಕಂಪ್ಯೂಟರ್‌ಗಳು, ಬಳಕೆದಾರರು, ಸಾಧನಗಳು) ತಾರ್ಕಿಕ ಗುಂಪು ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗುತ್ತದೆ. ಇದು ಅದೇ ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ ಡಾಟಾಬೇಸನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತದೆ.

ಡೊಮೇನ್‌ಗಳ ಒಳಗೆ ಸಾಂಸ್ಥಿಕ ಘಟಕಗಳಿವೆ. OU ಗಳು ಡೊಮೇನ್‌ಗಳಿಗೆ ಕ್ರಮಾನುಗತವನ್ನು ಒದಗಿಸುತ್ತವೆ, ಅದರ ಆಡಳಿತವನ್ನು ಸುಲಭಗೊಳಿಸುತ್ತವೆ, ಮತ್ತು ಸಂಸ್ಥೆಯ ರಚನೆಯನ್ನು ನಿರ್ವಹಣಾ ಅಥವಾ ಭೌಗೋಳಿಕ ನಿಯಮಗಳನ್ನು ಹೋಲುತ್ತದೆ. OU ಗಳು ಇತರ OU ಗಳನ್ನು ಒಳಗೊಂಡಿರಬಹುದು- ಇದರ ಅರ್ಥ ಡೊಮೇನ್‌ಗಳು ಕಂಟೇನರ್‌ಗಳಾಗಿವೆ. ಆಡಳಿತಾತ್ಮಕ ಅಧಿಕಾರಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ನಿಯೋಜಿಸಲಾಗುವ ಹಂತದಲ್ಲಿ OU ಗಳಿರುತ್ತವೆ. ಆದರೆ, ನಿಯೋಜನೆಗಳನ್ನು ಪ್ರತ್ಯೇಕ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಮೇಲೆ ಅಥವಾ ಆಟ್ರಿಬ್ಯೂಟ್‌ಗಳ ಮೇಲೆ ಸಹ ನಡೆಸಬಹುದು.

ಇದನ್ನು ತಾರ್ಕಿಕ ಮಾದರಿ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ ಏಕೆಂದರೆ ಇದು ನಿಯೋಜನೆಯ ಭೌತಿಕ ಅಂಶಗಳಿಂದ ಸ್ವತಂತ್ರವಾಗಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಪ್ರತಿ ಡೊಮೇನ್‌ನಲ್ಲಿ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಟೊಪಾಲಜಿಯಲ್ಲಿ ಅವಶ್ಯಕವಿರುವ ಡೊಮೇನ್ ನಿಯಂತ್ರಕರ ಸಂಖ್ಯೆ .

ಚಿತ್ರ 1 (Fig 1) ಫಾರೆಸ್ಟ್‌ಗಳು, ಡೊಮೇನ್‌ಗಳು ಮತ್ತು ಸಾಂಸ್ಥಿಕ ಘಟಕಗಳ ನಡುವಿನ ಸಂಬಂಧವನ್ನು ತೋರಿಸುತ್ತದೆ.

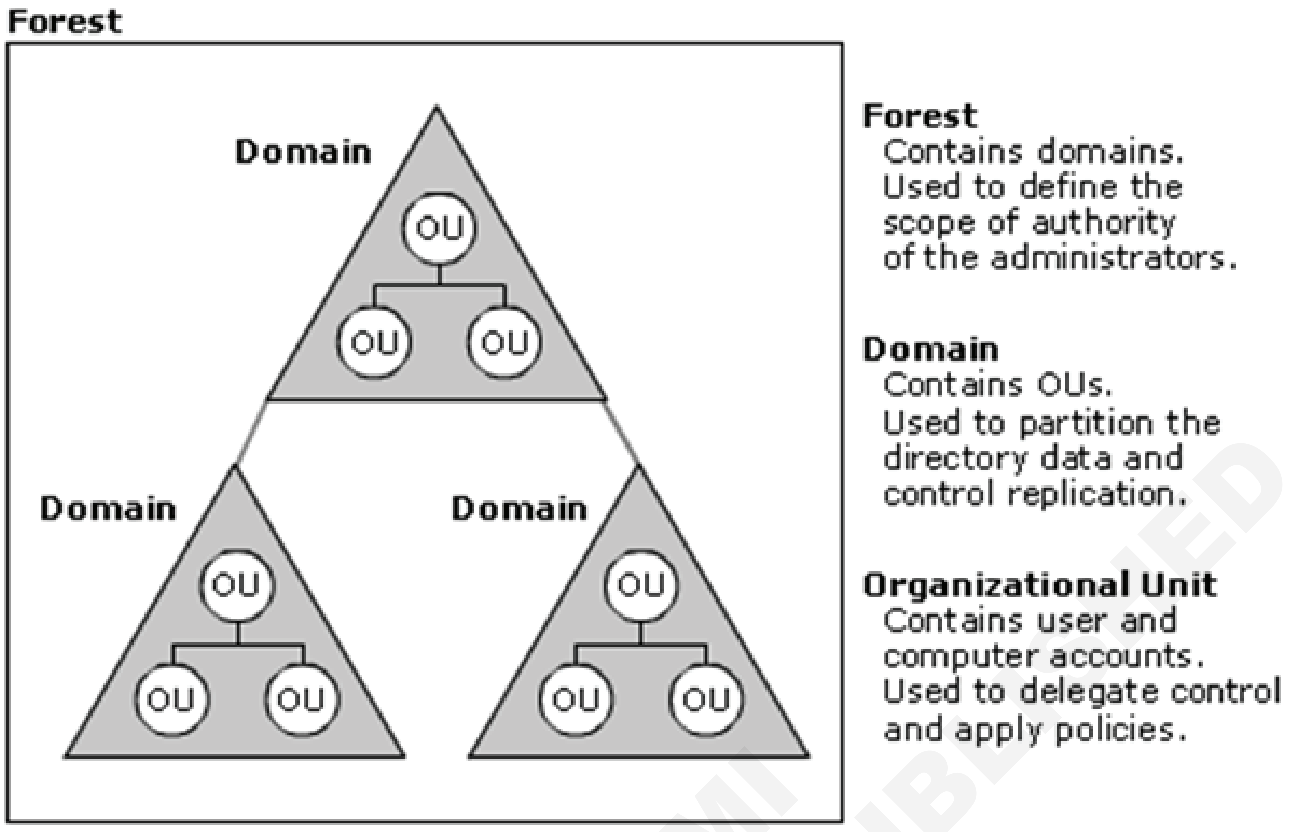
ಚಿತ್ರ 1 (Fig 1) ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ ಫಾರೆಸ್ಟ್‌ಗಳು, ಡೊಮೆನ್‌ಗಳು ಮತ್ತು ಸಾಂಸ್ಥಿಕ ಘಟಕಗಳ ನಡುವಿನ ಸಂಬಂಧ.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ಭೌತಿಕ ರಚನೆ (The Physical Structure of an Active Directory)

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ಭೌತಿಕ ರಚನೆಯು ಯಾವಾಗ ಮತ್ತು ಎಲ್ಲಿ ಲಾಗಾನ್ ಮತ್ತು ನಕಲು ಸಂಚಾರವು ಸಂಭವಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ. ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ಭೌತಿಕ ರಚನೆಯು ನೆಟ್‌ವರ್ಕನಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಭೌತಿಕ ಸಬ್‌ನೆಟ್‌ಗಳನ್ನು

© NIMI NOT TO BE REPUBLISHED

Fig 1



© NIMI NOT TO BE REPUBLISHED

ಒಳಗೊಂಡಿರುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಡೊಮೇನ್ ನಿಯಂತ್ರಕರು ಮತ್ತು ಡೊಮೇನ್ ನಿಯಂತ್ರಕರ ನಡುವಿನ ಪುನರಾವರ್ತನೆಗಳು.

ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಯ ಭೌತಿಕ ರಚನೆ (The physical structure of Active Directory):

- **ಡೊಮೇನ್ ಕಂಟ್ರೋಲರ್‌ಗಳು** (Domain Controllers: ಈ ಕಂಪ್ಯೂಟರ್‌ಗಳು ಮೈಕ್ರೊಸಾಫ್ಟ ಸರ್ವರ್ 2003/2000 ನ್ನು ಮತ್ತು ಸಕ್ರಿಯ ಡೈರೆಕ್ಟರಿಯನ್ನು ರನ್ ಮಾಡುತ್ತವೆ. ಪ್ರತಿಯೊಂದು ಡೊಮೇನ್ ನಿಯಂತ್ರಕವು ರೆಪ್ಲಿಕೇಷನ್, ಸಂಗ್ರಹಣೆ ಮತ್ತು ದೃಢೀಕರಣ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ಇದು ಗರಿಷ್ಟ ಒಂದು ಡೊಮೇನ್‌ನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಪ್ರತಿ ಡೊಮೇನ್‌ನಲ್ಲಿ ಯಾವಾಗಲೂ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಡೊಮೇನ್ ನಿಯಂತ್ರಕವನ್ನು ಹೊಂದಲು ಸಲಹೆ ನೀಡಲಾಗುತ್ತದೆ.
- **ಸಕ್ರಿಯ ಡೈರೆಕ್ಟರಿ ಸೈಟ್‌ಗಳು** (Active Directory Sites): ಈ ಸೈಟ್‌ಗಳು ಉತ್ತಮ ಸಂಪರ್ಕ ಹೊಂದಿದ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಸಂಗ್ರಹವಾಗಿದೆ. ನಾವು ಸೈಟ್ ಅನ್ನು ರಚಿಸುವ ಕಾರಣವೆಂದರೆ, ಸೈಟ್‌ನಲ್ಲಿ ಡೊಮೆನ್ ನಿಯಂತ್ರಕಗಳು ಆಗಾಗ್ಗೆ ಸಂವಹನ ನಡೆಸಬಹುದು. ಈ ರೀತಿಯಾಗಿ ಸೈಟ್‌ನಲ್ಲಿಯ ಸುಪ್ತತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ. ಎಂದರೆ ಒಂದು ಡೊಮೆನ್ ನಿಯಂತ್ರಕದಲ್ಲಿ ಮಾಡಿದ ಬದಲಾವಣೆಯನ್ನು ಇತರ ಡೊಮೆನ್ ನಿಯಂತ್ರಕಗಳಲ್ಲಿ ಪುನರಾವರ್ತಿಸಲಾಗುತ್ತದೆ. ಸೈಟನ್ನು ರಚಿಸುವ ಇನ್ನೊಂದು ಕಾರಣವೆಂದರೆ, ವಿಭಿನ್ನ ಲೊಕೆಷನ್‌ಗಳಲ್ಲಿ ಇರುವ ಡೊಮೇನ್ ನಿಯಂತ್ರಕಗಳ ನಡುವಿನ ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್‌ನ್ನು ಉತ್ತಮಗೊಳಿಸುವುದು.

ಕಂಪ್ಯೂಟರ್‌ಗಳ ನಿಜವಾದ ಭೌತಿಕ ಲೊಕೆಷನ್‌ನ್ನು ತಿಳಿಯದೆಯೆ ಸಾಮಾನ್ಯ ಲೋಕಲ್ ಏರಿಯಾ ನೆಟ್‌ವರ್ಕ್ (LAN) ಸಂಪರ್ಕವನ್ನು ಹಂಚಿಕೊಳ್ಳುವ ಎಲ್ಲ ಸಬ್‌ನೆಟ್‌ಗಳನ್ನು ಸೈಟ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಮಲ್ಟಿಡೊಮೇನ್ Active Directory Domain Services (ADDS) ಫಾರೆಸ್ಟ್‌ನಲ್ಲಿ ಕಂಡುಬರುವ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಭಾಗಶಃ ಪ್ರಾತಿನಿಧ್ಯವನ್ನು ಒಳಗೊಂಡಿರುವ ಡಾಟಾ ಸಂಗ್ರಹಣೆಯ ಮೂಲವನ್ನು ಗ್ಲೊಬಲ್ ಕ್ಯಾಟಾಲಾಗ್ ಎನ್ನುತ್ತಾರೆ. ಗ್ಲೊಬಲ್ ಕ್ಯಾಟಲಾಗ್‌ನ್ನು ಡೊಮೇನ್ ನಿಯಂತ್ರಕಗಳಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗಿದೆ. ನಿರ್ದಿಷ್ಟವಾಗಿ ಗ್ಲೊಬಲ್ ಕ್ಯಾಟಲಾಗ್ ಸರ್ವರ್ ಎಂದು ನಿಯೋಜಿಸಲಾಗುತ್ತದೆ. ನಿಜವಾದ ಡೊಮೇನ್ ಹೆಸರನ್ನು ತಿಳಿಯದೆಯೆ ಇದು ಯಾವುದೇ ಡೊಮೆನ್‌ನಲ್ಲಿರುವ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಪತ್ತೆ ಮಾಡಬಲ್ಲದು

ಗ್ಲೊಬಲ್ ಕ್ಯಾಟಲಾಗ್‌ಗೆ ನಿರ್ದೇಶಿಸಲಾದ ಹುಡುಕಾಟಗಳು ವೇಗವಾಗಿರುತ್ತವೆ. ಏಕೆಂದರೆ ಅವು ವಿಭಿನ್ನ ಡೊಮೆನ್ ನಿಯಂತ್ರಕಗಳ ಉಲ್ಲೇಖಗಳನ್ನು ಒಳಗೊಂಡಿರುವುದಿಲ್ಲ.

OU ನಲ್ಲಿ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಸಂಘಟಿಸುವುದು(Organizing resources in OU)

OU ಗಳು ಎಂದರೆ ಬಳಕೆದಾರರು, ಕಂಪ್ಯೂಟರ್, ಮತ್ತು ಇತರ

ಆಬ್ಜೆಕ್ಟ್ ಮಾಹಿತಿಗಳನ್ನು ಹೆಚ್ಚು ಸುಲಭವಾಗಿ ಅರ್ಥ ಮಾಡಿಸುವ ಲೇಔಟ್‌ನಲ್ಲಿ ಸಂಘಟಿಸುವ ಪ್ರಾಥಮಿಕ ವಿಧಾನವಾಗಿದೆ. ಒಂದು ಸಂಸ್ಥೆಯು ರೂಟ್ ಸಾಂಸ್ಥಿಕ ಘಟಕವನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇಲ್ಲಿ ಮೂರು ನೆಸ್ಟೆಡ್ ಸಾಂಸ್ಥಿಕ ಘಟಕಗಳನ್ನು ಇರಿಸಲಾಗುತ್ತದೆ. ಈ ನೆಸ್ಟಿಂಗ್ ನೆಟ್‌ವರ್ಕ್ ಸಂಪನ್ಮೂಲಗಳ ಸುಲಭ ವೀಕ್ಷಣೆ ಮತ್ತು ನಿರ್ವಹಣೆಗಾಗಿ ಬಳಕೆದಾರರನ್ನು ಅನೇಕ ಕಂಟೇನರ್‌ಗಳಲ್ಲಿ ವಿತರಿಸಲು ಸಂಸ್ಥೆಯನ್ನು ಸಕ್ರಿಯಗೊಳಿಸುತ್ತದೆ.

OU ಗಳನ್ನು ಸುಲಭ ಸಂಘಟನೆ ಮತ್ತು ಆಡಳಿತ ನಿಯೋಗಕ್ಕಾಗಿ ಮತ್ತೆ ಸಂಪನ್ಮೂಲ OU ಗಳನ್ನಾಗಿ ವಿಂಗಡಿಸಬಹುದು. ದೂರದ ಕಚೇರಿಗಳು ಸ್ಥಳೀಯ ಆಡಳಿತಕ್ಕಾಗಿ ತಮ್ಮ ಸ್ವಂತ OU ಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ಅದಾಗ್ಯೂ ಒಂದು ಸಂಸ್ಥೆಯು ಆಡಳಿತವನ್ನು ಇನ್ನೊಂದು ನಿರ್ವಾಹಕರ ಗುಂಪಿಗೆ ನಿಯೋಜಿಸುವ ನಿರ್ದಿಷ್ಟ ಅವಶ್ಯಕತೆಯನ್ನು ಹೊಂದಿದ್ದರೆ ಮಾತ್ರ OU ವನ್ನು ರಚಿಸಬೇಕು ಎಂಬುದನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ. ಅದೇ ವ್ಯಕ್ತಿ ಅಥವಾ ಜನರ ಗುಂಪು ಸಂಪೂರ್ಣ ಡೊಮೇನ್ ಅನ್ನು ನಿರ್ವಹಿಸಿದರೆ, OU ಗಳನ್ನು ಸೇರಿಸುವ ಮೂಲಕ ಪರಿಸರದ ಸಂಕೀರ್ಣತೆಯನ್ನು ಹೆಚ್ಚಿಸುವ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ. ವಾಸ್ತವವಾಗಿ ಹಲವಾರು OU ಗಳು ಗ್ರೂಪ್ ಪಾಲಿಸಿಗಳು, ಲಾಗಾನ್‌ಗಳು ಮತ್ತು ಇತರ ಅಂಶಗಳ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರಬಹುದು.

ತಮ್ಮ ಸ್ವಂತ ಬಳಕೆದಾರರ ಮೇಲೆ ವಿವಿಧ ಹಂತದ ಆಡಳಿತ ನಿಯಂತ್ರಣಗಳನ್ನು ಹೊಂದಲು ಪ್ರತ್ಯೇಕ ಇಲಾಖೆಗಳಿಗೆ ಅವಕಾಶ ಕಲ್ಪಿಸುವಂತೆ OU ಗಳನ್ನು ರಚಿಸಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಇಂಜಿನಿಯರಿಂಗ್ ಇಲಾಖೆಯಲ್ಲಿರುವ ಒಬ್ಬ ಸೆಕ್ರೆಟರಿಯನ್ನು ತನ್ನ ಸ್ವಂತ OU ನಲ್ಲಿ ಬಳಕೆದಾರರ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ರಿಸೆಟ್ ಮಾಡಲು ನಿಯೋಜಿಸಬಹುದು. ಈ ಸಂದರ್ಭಗಳಲ್ಲಿ OU ನ ಮತ್ತೊಂದು ಅನುಕೂಲವೆಂದರೆ ಬಳಕೆದಾರರನ್ನು ಸುಲಭವಾಗಿ ಒಂದು OU ದಿಂದ ಎಳೆದು ಇನ್ನೊಂದಕ್ಕೆ ಹಾಕಬಹುದು. ಉದಾಹರಣೆಗೆ, ಬಳಕೆದಾರರನ್ನು ಒಂದು ಇಲಾಖೆಯಿಂದ ಇನ್ನೊಂದು ಇಲಾಖೆಗೆ ಸ್ಥಳಾಂತರಿಸಿದರೆ, ಅವರನ್ನು ಅವರ ಹೊಸ ಇಲಾಖೆಯ OU ಗೆ ಸ್ಥಳಾಂತರಿಸುವುದು ಅತ್ಯಂತ ಸರಳವಾಗಿದೆ.

ನಿರ್ವಾಹಕರು ಯಾವುದೆ ಸಮಯದಲ್ಲಿ ರಚನಾತ್ಮಕ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುವುದು ಸೂಕ್ತವೆಂದು ಭಾವಿಸಿದಲ್ಲಿ OU ರಚನೆಯನ್ನು ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು. ಇದು ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿಗೆ ಯಾವುದೇ ಸಮಯದಲ್ಲಿ ಬದಲಾವಣೆಯನ್ನು ಮಾಡುವ ಒಂದು ಪ್ರಯೋಜನೆಯನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಗ್ರೂಪ್ ಪಾಲಿಸಿಯು ಮೈಕ್ರೊಸಾಫ್ಟ್ ವಿಂಡೋಸ್ NT ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ನ ವೈಶಿಷ್ಟ್ಯವಾಗಿದೆ. ಇದು ಬಳಕೆದಾರರ ಖಾತೆಗಳ ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ಖಾತೆಗಳ ಕೆಲಸ ಮಾಡುವ ಪರಿಸರವನ್ನು ನಿಯಂತ್ರಿಸುತ್ತದೆ. ಗ್ರೂಪ್ ಪಾಲಿಸಿಯು, ಸಕ್ರೀಯ ಡೈರೆಕ್ಟರಿ ಪರಿಸರದಲ್ಲಿ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳ, ಅಪ್ಲಿಕೇಷನ್‌ಗಳ, ಬಳಕೆದಾರರ ಸೆಟಿಂಗ್‌ಗಳ ಕೇಂದ್ರೀಕೃತ ನಿರ್ವಹಣೆಯನ್ನು ಮತ್ತು ಸಂರಚನೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. Local Group Policy ("LGPO" ಅಥವಾ "LocalGPO") ಎನ್ನುವ ಗ್ರೂಪ್ ಪಾಲಿಸಿಯ ಒಂದು ಆವೃತ್ತಿಯು ಸ್ವತಂತ್ರ ಮತ್ತು ನಾನ್ - ಡೊಮೇನ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಲ್ಲಿ, Group Policy Object ನಿರ್ವಹಣೆಯನ್ನು ಅನುಮತಿಸುತ್ತದೆ. ಗ್ರೂಪ್ ಪಾಲಿಸಿಯು ಸಕ್ರಿಯ ಡೈರೆಕ್ಟರಿಯನ್ನು ನಿಯೋಜಿಸಲು ಪ್ರಮುಖ ಕಾರಣಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಏಕೆಂದರೆ ಇದು ನಿಮಗೆ ಬಳಕೆದಾರರನ್ನು ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

ಗ್ರೂಪ್ ಪಾಲಿಸಿಯು ಬಳಕೆದಾರರು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಏನು ಮಾಡಬಹುದು ಮತ್ತು ಮಾಡಬಾರದು ಎಂಬುದನ್ನು ನಿಯಂತ್ರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಬಳಕೆದಾರರು ಅತಿ ಸರಳವಾದ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವುದನ್ನು ತಡೆಯುವುದಕ್ಕಾಗಿ ಒಂದು ಪಾಸ್‌ವರ್ಡ್ ಸಂಕೀರ್ಣತೆಯ ಪಾಲಿಸಿಯನ್ನು ಜಾರಿಗೊಳಿಸಲಾಗುತ್ತದೆ. ಇದು ಗುರುತಿಸಲಾಗದ ಬಳಕೆದಾರರನ್ನು ರಿಮೋಟ್ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಂದ ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪರ್ಕಿಸಲು, ವಿಂಡೊಸ್ ಟಾಸ್ಕ್ ಮ್ಯಾನೇಜರ್‌ಗೆ ಪ್ರವೇಶವನ್ನು ನಿರ್ಬಂಧಿಸಲು ಅಥವಾ ನಿರ್ದಿಷ್ಟ ಫೋಲ್ಡರ್‌ಗಳಿಗೆ ಪ್ರವೇಶವನ್ನು ನಿರ್ಬಂಧಿಸಲು ಅನುಮತಿಸುತ್ತದೆ ಅಥವಾ ತಡೆಯುತ್ತದೆ. ಇಂತಹ ಸಂರಚನೆಗಳ ಸೆಟ್‌ಗೆ ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಆಬ್ಜೆಕ್ಟ್ (GPO) ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ಸಕ್ರಿಯ ಡೈರೆಕ್ಟರಿಯ ಬ್ಯಾಕಪ್ ಮತ್ತು ರಿಸ್ಟೋರ್ (Active Directory Backup and Restore)

ಸಕ್ರಿಯ ಡೈರೆಕ್ಟರಿಯು ನಿಮ್ಮ ಮೂಲಸೌಕರ್ಯದ ಅತ್ಯಂತ ನಿರ್ಣಾಯಕ ಘಟಕಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದು ಸ್ಥಗಿತವಾದರೆ, ನಿಮ್ಮ ನೆಟ್‌ವರ್ಕ್ ನಿಷ್ಪ್ರಯೋಜಕವಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ವ್ಯಾಪಾರ ನಿರಂತತೆಯನ್ನು ಮತ್ತು ಅನುಸರಣೆಯನ್ನು ಖಚಿತಪಡಿಸಲು ನಿಮಗೆ ಗಟ್ಟಿಯಾದ ಬ್ಯಾಕಪ್ ಮತ್ತು ರಿಕವರಿ ಯೋಜನೆಯನ್ನು ಹೊಂದಿರಬೇಕು.

ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಆದ್ಯತೆಯ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನು ಎಲ್ಲಿ ಸಂರಚಿಸಲಾಗುತ್ತದೆಯೊ ಅಲ್ಲಿ ಸಮಾನವಾದ ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಸೆಟ್ಟಿಂಗ್‌ಗಳನ್ನೂ ಸಹ ಸಂರಚಿಸಲಾಗುತ್ತದೆ. ನಂತರ ಗ್ರೂಪ್ ಪಾಲಿಸಿ ಸೆಟ್ಟಿಂಗ್‌ನ ಮೌಲ್ಯದ ಆದ್ಯತೆಯನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ. ಗ್ರೂಪ್ ಪಾಲಿಸಿಯು ಡೊಮೇನ್‌ನ ಭದ್ರತೆಯಾಗಿದೆ. ಸಿಸ್ಟಮ್ ಸ್ಥಿತಿಯನ್ನು ಬ್ಯಾಕಪ್ ಮಾಡಿದಾಗ ಬ್ಯಾಕಪ್ ಯುಟಿಲಿಟಿಯು ತನ್ನಷ್ಟಕ್ಕೆ ತಾನೆ ಪತ್ತೆ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ, ಆಡಿಟ್ ಮತ್ತು ಪರೀಕ್ಷೆ (Access Control, Audit and testing)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ಉದ್ದೇಶ ಮತ್ತು ಹಂತಗಳನ್ನು ವಿವರಿಸಿ.
- ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ವಿವಿಧ ಲೇಯರ್‌ಗಳನ್ನು ವಿವರಿಸಿ.
- ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ವಿವರಿಸಿ.
- ಭದ್ರತೆಗಾಗಿ ಮಾಹಿತಿಯನ್ನು ವರ್ಗೀಕರಿಸಿ
- ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಪ್ರೊಟೊಕಾಲ್‌ಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ.
- ಭದ್ರತಾ ಆಡಿಟ್‌ನ್ನು ವಿವರಿಸಿ
- ಭದ್ರತಾ ಆಡಿಟ್‌ಗಳ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ವಿವರಿಸಿ
- ಆಡಿಟ್‌ನ ಭದ್ರತೆಯನ್ನು ನಿರ್ವಹಿಸುವ ಪ್ರಕ್ರಿಯೆಯನ್ನು ವಿವರಿಸಿ.
- ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನ ಮತ್ತು ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷೆಯನ್ನು ವಿವರಿಸಿ.
- ವಿವಿಧ ರೀತಿಯ ಭದ್ರತಾ ಆಡಿಟ್ ಸಾಧನಗಳನ್ನು ತಿಳಿಸಿ.

ಪರಿಚಯ (Introduction) : ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಅಧಿಕಾರ ಹೊಂದಿರುವ ಜನರಿಗೆ ರಕ್ಷಿಸಲ್ಪಟ್ಟ ಮಾಹಿತಿಗೆ ಪ್ರವೇಶವನ್ನು ನಿರ್ಬಂಧಿಸಲಾಗುತ್ತದೆ. ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಉದ್ದೇಶವು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ ಕಾನೂನುಬದ್ಧ ಬಳಕೆದಾರರು ನಿರ್ವಹಿಸುವ ಕ್ರಮಗಳು ಅಥವಾ ಕಾರ್ಯಾಚರಣೆಗಳನ್ನು ಸೀಮಿತಗೊಳಿಸುವುದು ಆಗಿದೆ. ಹಾಗೆಯೆ ಬಳಕೆದಾರರ ಪರವಾಗಿ ಮಾಡಲು ಅನುಮತಿಸಲಾದ ಯಾವ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದು ಎಂದಾಗಿದೆ. ಈ ರೀತಿಯಾಗಿ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವು ಭದ್ರತೆ ಉಲ್ಲಂಘನೆಗೆ ಕಾರಣವಾಗುವ ಚಟುವಟಿಕೆಗಳನ್ನು ತಡೆಯಲು ಪ್ರಯತ್ನಿಸುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಕಾರ್ಯವಿಧಾನಗಳ ಅತ್ಯಾಧುನಿಕತೆಯು ಸಂರಕ್ಷಿತ ಮಾಹಿತಿಯ ಮೌಲ್ಯಕ್ಕೆ ಅನುಗುಣವಾಗಿರಬೇಕು. ಮಾಹಿತಿಯು ಎಷ್ಟು ಹೆಚ್ಚು ಸೂಕ್ಷ್ಮ ಅಥವಾ ಮೌಲ್ಯಯುತವಾಗಿರುತ್ತದೆಯೊ, ನಿಯಂತ್ರಣ ಕಾರ್ಯವಿಧಾನಗಳು ಅಷ್ಟು ಬಲವಾಗಿರುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ನಿರ್ಮಿಸುವ ಅಡಿಪಾಯ ಗುರುತಿಸುವಿಕೆ ಮತ್ತು ದೃಢಿಕರಣದೊಂದಿಗೆ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣವನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಮೂರು ಹಂತಗಳಲ್ಲಿ ಸಾಧಿಸಲಾಗುತ್ತದೆ :

ಗುರುತಿಸುವಿಕೆ, ದೃಢೀಕರಣ ಮತ್ತು ಅಧಿಕಾರ

- ಗುರುತಿಸುವಿಕೆಯು ಯಾವ ವ್ಯಕ್ತಿ ಅಥವಾ ಏನು ವಿಷಯ ಎಂಬುದರ ಪ್ರತಿಪಾದನೆಯಾಗಿದೆ. ಬಳಕೆದಾರನು ಯೂಸರ್ ನೇಮ್‌ನ್ನು ನಮೂದಿಸುವ ಮೂಲಕ ತನ್ನನು ತಾನು ಗುರುತಿಸಿಕೊಳ್ಳುತ್ತಾನೆ.
- ದೃಢೀಕರಣವು, ಬಳಕೆದಾರರ ಗುರುತನ್ನು ಪರಿಶೀಲಿಸುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ದೃಢೀಕರಣವು, ಮಾಹಿತಿ ಭರವಸೆಯ (information assurance (IA)) ಐದು ಸ್ಥಂಭಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇತರ ನಾಲ್ಕು ಯಾವವೆಂದರೆ ಸಮಗ್ರತೆ, ಲಭ್ಯತೆ, ಗೌಪ್ಯತೆ ಮತ್ತು ನಿರಾಕರಣೆಯ ಅಸಮ್ಮತಿ.

ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ನಮೂದಿಸುವುದು ದೃಢೀಕರಣದ ಸಾಮಾನ್ಯ ವಿಧಾನಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಆದರೆ ಪಾಸ್‌ವರ್ಡ್ ಆಧಾರಿತ ದೃಢಿಕರಣವು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ಬಳಸಲು ಸೂಕ್ತವಾಗಿಲ್ಲ. ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ಕಳುಹಿಸಲಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ತಡೆಹಿಡಿಯಬಹುದು ಮತ್ತು ಹ್ಯಾಕರ್‌ಗಳಿಂದ ದುರ್ಬಳಕೆಯಾಗಬಹುದು. ಭದ್ರತೆಯ ಕಾಳಜಿಯ ಜೊತೆಗೆ, ಪಾಸ್‌ವರ್ಡ್ ಆಧಾರಿತ ದೃಢೀಕರಣವು, ಬಳಕೆದಾರರು ಪ್ರತಿ ಬಾರಿ ನೆಟ್‌ವರ್ಕ್ ಸೇವೆಯನ್ನು ಪಡೆಯುವಾಗ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ನಮೂದಿಸಬೇಕಾಗಿರುವುದರಿಂದ ಇದು ಸೂಕ್ತವಾಗಿಲ್ಲ. ಮಹತ್ವದ ವಹಿವಾಟುಗಳಿಗಾಗಿ ಈ ವ್ಯವಸ್ಥೆಯಲಿ ್ಲನ ದೌರ್ಬಲ್ಯಗಳು ಎಂದರೆ, ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಆಗಾಗ್ಗೆ ಕದಿಯಬಹುದು, ಆಕಸ್ಮಿಕವಾಗಿ ಬಹಿರಂಗವಾಗಬಹುದು ಅಥವಾ ಮರೆತುಹೋಗಬಹುದು.

ಅಧಿಕಾರ : ಒಬ್ಬ ವ್ಯಕ್ತಿ, ಪ್ರೊಗ್ರಾಂ, ಅಥವಾ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಗುರುತಿಸಿದ ನಂತರ ಮತ್ತು ದೃಢೀಕರಣದ ನಂತರ ಅವರು ಯಾವ ಮಾಹಿತಿ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ಅನುಮತಿಸಲಾಗಿದೆ ಮತ್ತು ಯಾವ ಕ್ರಮಗಳನ್ನು ನಿರ್ವಹಿಸಲು (run, view, create, delete, or change,) ಅನುಮತಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸಬೇಕಾಗುತ್ತದೆ.

ಇದನ್ನು ಅಧಿಕಾರ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಮಾಹಿತಿಯನ್ನು ಪ್ರವೇಶಿಸುವ ಅಧಿಕಾರ ಮತ್ತು ಇತರ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ಆಡಳಿತಾತ್ಮಕ ಪಾಲಿಸಿಗಳಿಂದ ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳಿಂದ ಪ್ರಾರಂಭಿಸಲಾಗುತ್ತದೆ. ಪಾಲಿಸಿಗಳು, ಯಾವ ಮಾಹಿತಿಯನ್ನು ಮತ್ತು ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು, ಯಾರಿಂದ ಮತ್ತು ಯಾವ ಪರಿಸ್ಥಿತಿಗಳಲ್ಲಿ ಎಂಬುದನ್ನು ಸೂಚಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ದೃಢೀಕರಣ ವಿಧಾನಗಳು (Authentication Methods)

- ಕದ್ದಾಲಿಕೆ ಅಪಾಯವನ್ನು ದೃಢೀಕರಣದ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸಿಕೊಂಡು ನಿರ್ವಹಿಸಬಹುದು. ಸಂಪರ್ಕಿಸುವ ಪಕ್ಷವು ಮೌಲ್ಯವನ್ನು ಕಳುಹಿಸುತ್ತದೆ, ಸಾಮಾನ್ಯವಾಗಿ ಕ್ಲೈಂಟ್ ಐಪಿ ಅಡ್ರೆಸ್, ಟೈಮ್‌ಸ್ಟಾ ಯಂಪ್ ಮತ್ತು ಹೆಚ್ಚುವರಿ ರಹಸ್ಯ ಮಾಹಿತಿಗಳ ಹ್ಯಾಶ್. ಏಕೆಂದರೆ, ಈ ಹ್ಯಾಶ್, ಪಡೆದುಕೊಂಡಿರುವ ಪ್ರತಿ URI ಗೆ ವಿಶಿಷ್ಟವಾಗಿರುತ್ತದೆ. ಇತರ ಬೇರೆ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲಾಗುವುದಿಲ್ಲ ಮತ್ತು ಪತ್ತೆಮಾಡದೇ ಇತರ IP ಅಡ್ರೆಸ್‌ನಿಂದ ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ. ಹ್ಯಾಶಿಂಗ್ ಇರುವುದರಿಂದ ಪಾಸ್‌ವರ್ಡ್ ಕದ್ದಾಲಿಕೆಗೆ ಗುರಿಯಾಗುವುದಿಲ್ಲ. ಸಿಸ್ಟಮ್‌ಗಳು man-in-the middle ನಂತಹ ದಾಳಿಗಳಿಂದ ದುರ್ಬಲವಾಗುತ್ತವೆ.

- **ಒನ್ -ಟೈಮ್ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು** : ಪಾಸ್‌ವರ್ಡ್‌ನ ಮರುಬಳಕೆಗೆ ಸಂಬಂಧಿಸಿದ ಸಮಸ್ಯೆಗಳನ್ನು ತಪ್ಪಿಸಲು, ಒನ್ -ಟೈಮ್ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ. ಎರಡು ರೀತಿಯ ಒನ್ -ಟೈಮ್ ಪಾಸ್‌ವರ್ಡ್‌ಗಳಿವೆ: challenge-response password ಮತ್ತುpassword list.

- **ಪಬ್ಲಿಕ್-ಕೀ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿ** : ಪಬ್ಲಿಕ್-ಕೀ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಯು ಬಹಳ ಸಂಕೀರ್ಣವಾದ ಸಮಸ್ಯೆಗಳನ್ನು ಆಧರಿಸಿದೆ. ಇವುಗಳಿಗೆ ಬಹಳ ವಿಶೇಷವಾದ ಜ್ಞಾನದ ಅವಶ್ಯಕತೆ ಇದೆ. ಪಬ್ಲಿಕ್-ಕೀ ಕ್ರಿಪ್ಟೋಗ್ರಾಫಿಯು ಎರಡು ಕೀಗಳನ್ನು ಬಳಸುತ್ತದೆ, ಒಂದು ಪ್ರೈವೇಟ್ ಮತ್ತು ಇನ್ನೊಂದು ಪಬ್ಲಿಕ್. ಎರಡೂ ಕೀಗಳನ್ನು ಅತ್ಯಂತ ಸಂಕೀರ್ಣವಾದ ಗಣಿತದ ಸಮೀಕರಣದ ವಿಧಾನದಿಂದ ಒಟ್ಟಿಗೆ ಜೋಡಿಸಲಾಗಿದೆ. ಪ್ರೈವೇಟ್ ಕೀಯನ್ನು, ಸಂವಹನ ನಡೆಸುವ ಯಂತ್ರಗಳ ನಡುವಿನ ಸಂದೇಶಗಳನ್ನು ಡಿಕ್ರಿಪ್ಟ್ ಮಾಡಲು ಮತ್ತು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡಲು ಸಹ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪಬ್ಲಿಕ್ ಕೀ ಎನ್‌ಕ್ರಿಪ್ಷನ್ ಮತ್ತು ಸಹಿಯ ಪರಿಶೀಲನೆ ಎರಡನ್ನೂ ಸಾಧಿಸುತ್ತದೆ.

- **Zero -knowledge proofs** : Zero -knowledge proof ಗಳು ಒಂದು ಹೋಸ್ಟ್‌ಗೆ ಯಾವುದೇ ರಹಸ್ಯ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸದೇ ಇನ್ನೊಂದು ಹೋಸ್ಟ್‌ಗೆ ಪ್ರವೇಶವನ್ನು ಅನುಮತಿಸಲು ಮನವರಿಕೆ ಮಾಡುವದನ್ನು ಸಾಧ್ಯವಾಗಿಸುತ್ತದೆ. ಈ ರೀತಿಯ ದೃಢಿಕರಣದಲ್ಲಿ ತೊಡಗಿಸಿಕೊಂಡ ಹೋಸ್ಟ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ದೃಢಿಕರಣವನ್ನು ಅಂತಿಮಗೊಳಿಸಲು ಹಲವಾರು ಬಾರಿ ಸಂವಹನ ಮಾಡುತ್ತವೆ.

- ಪಬ್ಲಿಕ್ ಕೀ ಮೂಲಸೌಕರ್ಯದ ಭಾಗವಾಗಿ, Certificate Authority (CA) ದಿಂದ ನೀಡಲಾದ ಮತ್ತು ಪರಿಶೀಲಿಸಲಾದ ಡಿಜಿಟಲ್ ಪ್ರಮಾಣಪತ್ರಗಳನ್ನು ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ದೃಢೀಕರಣವನ್ನು ನಿರ್ವಹಿಸಲು ಇನ್ನೊಂದು ಪ್ರಮಾಣಿಕೃತ ಮಾರ್ಗವೆಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

- ದೃಢೀಕರಣದ ಇನ್ನೊಂದು ವಿಧಾನವೆಂದರೆ ಬಯೊಮೆಟ್ರಿಕ್ಸ್. ಇದು ಬಳಕೆದಾರರ ಉಪಸ್ಥಿತಿಯನ್ನು ಮತ್ತು ಜೈವಿಕ ಮೇಕಪ್‌ನ್ನು (ಅಂದರೆ, ರೆಟಿನ ಅಥವಾ ಬೆರಳಚ್ಚು) ಅವಲಂಬಿಸಿರುತ್ತದೆ. ಈ ತಂತ್ರಜ್ಞಾನವು ಹ್ಯಾಕರ್‌ಗಳಿಗೆ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಪ್ರವೇಶಿಸುವುದನ್ನು ಕಷ್ಟವಾಗಿಸುತ್ತದೆ. ಇದು ದೃಢಿಕರಣಕ್ಕಾಗಿ Secure Sockets Layer (SSL), IP SEC, Secure Shell (SSH), Kerberos authentication ಮತ್ತು Extensible Authentication Protocol (EAP) ಮುಂತಾದ ವಿಶೇಷ ಪ್ರೊಟೊಕಾಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

ಭದ್ರತಾ ನಿರ್ವಾಹಕರು ಒಂದು ಸಂಸ್ಥೆಯ ಭದ್ರತಾ ಪಾಲಿಸಿಯ ಆಧಾರದ ಮೇಲೆ ಅಧಿಕಾರಗಳ ಡಾಟಾಬೇಸನ್ನು ನಿರ್ವಹಿಸುತ್ತಾರೆ. ರೆಫರನ್ಸ್ ಮಾನಿಟರ್ ಅಧಿಕಾರದ ಡಾಟಾಬೇಸ್‌ನ್ನು ಒಬ್ಬ ಬಳಕೆದಾರನು ಕಾರ್ಯಾಚರಣೆಯನ್ನು ಮಾಡಲು ಪ್ರಯತ್ನಿಸುವುದು ವಾಸ್ತವವಾಗಿ ಆ ಕಾರ್ಯವನ್ನು ಮಾಡಲು ಅಧಿಕಾರವನ್ನು ಹೊಂದಿರುವನೊ ಅಥವಾ ಇಲ್ಲವೊ ಎಂದು ನಿರ್ಧರಿಸುತ್ತದೆ. ಆಡಿಟಿಂಗ್ ಮಾನಿಟರ್‌ಗಳು ಸಿಸ್ಟಮ್‌ನಲ್ಲಿರುವ ಸಂಬಂಧಿತ ಚಟುವಟಿಕೆಯ ದಾಖಲೆಯನ್ನು ಇರಿಸುತ್ತವೆ.

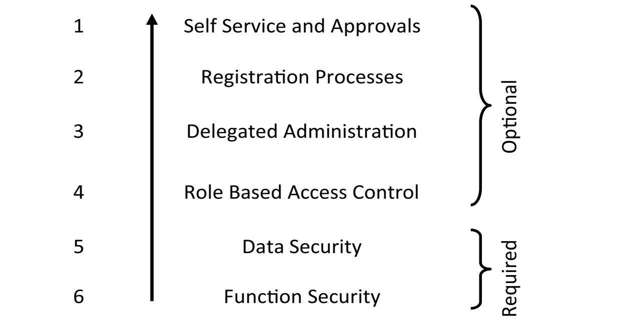
ದೃಢೀಕರಣ ಮತ್ತು ಪ್ರವೇಶ ನಿಯಂತ್ರಣಗಳ ನಡುವೆ ಸ್ಪಷ್ಟವಾದ ವ್ಯತ್ಯಾಸವನ್ನು ತಿಳಿಯುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ. ಬಳಕೆದಾರರ ಗುರುತನ್ನು ಸರಿಯಾಗಿ ಸ್ಥಾಪಿಸುವುದು ದೃಢಿಕರಣ ಸೇವೆಯ ಜವಾಬ್ದಾರಿಯಾಗಿರುತ್ತದೆ. ಪ್ರವೇಶ ನಿಯಂತ್ರಣವು ರೆಫರನ್ಸ್ ಮಾನಿಟರ್ ಮೂಲಕ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವನ್ನು ಜಾರಿಗೊಳಿಸುವ ಮೊದಲು ಬಳಕೆದಾರರ ದೃಢಿಕರಣವನ್ನು ಯಶಸ್ವಿಯಾಗಿ ಪರಿಶೀಲಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ಊಹಿಸುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ಅನುಕ್ರಮ ಪದರಗಳು (Successive Layers of Access Control)

ಪ್ರವೇಶ ನಿಯಂತ್ರಣವನ್ನು ಸತತ ಲೇಯರ್‌ಗಳಲ್ಲಿ ಅಳವಡಿಲಾಗಿದೆ ಮತ್ತು ಪ್ರತಿಯೊಂದು ಲೇಯರ್‌ನ್ನು ಅದರ ಹಿಂದಿನ ಲೇಯರ್ ಮೇಲೆ ನಿರ್ಮಿಸಲಾಗಿದೆ. ಸಂಸ್ಥೆಗಳು ಐಚ್ಛಿಕವಾಗಿ, ಯಾಂತ್ರಿಕೃತ ಮಟ್ಟ ಮತ್ತು ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಫಂಕ್ಷನ್ ಮತ್ತು ಡಾಟಾಸೆಕ್ಯುರಿಟಿ ಮಾಡೆಲ್‌ಗಳ ಆಧಾರದ ಮೇಲೆ ವಿವಿಧ ಪದರಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಇದನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಲು ವಿವಿಧ ಮಾದರಿಗಳು ಇರಬಹುದು. ಉದಾಹರಣೆಗೆ, ಒರ್‍ಯಾಕಲ್ ಯೂಸರ್ ಮ್ಯಾನೇಜಮೆಂಟ್, ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ಆರು ಲೇಯರ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. (Fig 1 ನೋಡಿರಿ).Core Security layer ಗಳು ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿವೆ :

NOT TO BE REPUBLISHED © NIMI

Fig 1



- Function Security
- Data Security

ಮುಂದಿನ ನಾಲ್ಕು ಲೇಯರ್‌ಗಳು Oracle User Management ನ ಭಾಗವಾಗಿವೆ :

- Role-Based Access Control
- Delegated Administration
- Registration Processes
- Self Service and Approvalss

ಸಾಮಾನ್ಯವಾಗಿ, ಬೇಸಿಕ್ ಸಿಸ್ಟಮ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ ಕಾರ್ಯಗಳೊಂದಿಗೆ, ಹೆಚ್ಚಿನ ವಿತರಣೆಗೆ ಪ್ರಗತಿಯೊಂದಿಗೆ, ಆಡಳಿತದ ಸ್ಥಳೀಯ ಮೋಡ್‌ಗಳೊಂದಿಗೆ, ಪ್ರಾರಂಭವಾಗುವ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವು ಬಳಕೆದಾರರನ್ನು ಕೆಲವು ಸಾಮಾನ್ಯ, ಪೂರ್ವನಿರ್ಧರಿತ ಕಾರ್ಯಗಳನ್ನು ಸ್ವಯಂ ತಾನೇ ಮಾಡಲು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ಲೇಯರ್‌ಗಳು ಹೇಗೆ ಪರಸ್ಪರ ಒಂದರ ಮೇಲೊಂದು ನಿರ್ಮಿಸಲ್ಪಟ್ಟಿವೆ ಎಂಬುದನ್ನು ಟೇಬಲ್ 1 ತೋರಿಸುತ್ತದೆ.

ಸೆಕ್ಯುರಿಟಿ ಮತ್ತು ಡಾಟಾ ಸೆಕ್ಯುರಿಟಿ ಯಾಂತ್ರಿಕ ರಚನೆಯು ಭದ್ರತಾ ಸಿಸ್ಟಮ್‌ನ ಬೇಸ್ ಲೇಯರ್‌ಗಳನ್ನು ನಿಯೋಜಿಸುತ್ತದೆ. ಮತ್ತು ಸಾಂಪ್ರದಾಯಿಕ ವ್ಯವಸ್ಥೆಯ ಆಡಳಿತಾತ್ಮಕ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಇವು ನಿರ್ದಿಷ್ಟ ಮೆನುಗಳಿಗೆ ಪ್ರವೇಶವನ್ನು ಅನುಮತಿಸುವ ಮೂಲಕ ಸಾಮಾನ್ಯ ವ್ಯವಸ್ಥೆಯ ಆಡಳಿತಕ್ಕೆ ಯೂಸರ್ ಮ್ಯಾನೆಜಮೆಂಟ್‌ನ ವ್ಯಾಪ್ತಿಯನ್ನು ಸೀಮಿತಗೊಳಿಸುತ್ತದೆ.

ಟೇಬಲ್ 1

Layer of Access	Level of Administration
Self Service and Approvals	End Users
Registration Processes	
Delegated Administration	Local Administration
Role Based Access Control	
Data Security	System Administrator
Function Security	

ಸ್ಥಳಿಯ ನಿರ್ವಾಹಕರು : ರೋಲ್ ಬೇಸ್ಡ್ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವು ಮತ್ತು ನಿಯೋಜಿತ ಆಡಳಿತವನ್ನು ಡಾಟಾ ಭದ್ರತೆಗೆ ಮತ್ತು ಫಂಕ್ಷನ್ ಭದ್ರತಾ ಲೇಯರ್‌ಗಳಿಗೆ ಸೇರಿಸಿದಾಗ, ಸಿಸ್ಟಮ್ ಆಡಳಿತ ಕಾರ್ಯಗಳನ್ನು ಸ್ಥಳೀಯ ನಿರ್ವಾಹಕರಿಗೆ ವಿತರಿಸಬಹುದು. ಸ್ಥಳೀಯ ನಿರ್ವಾಹಕರು ಸಂಸ್ಥೆಯ ಬಳಕೆದಾರರ ಸಬ್‌ಸೆಟ್‌ನ್ನು ನಿರ್ವಹಿಸುತ್ತಾರೆ.

ಅಂತಿಮ ಬಳಕೆದಾರರು : ನೋಂದಣಿ ಪ್ರಕ್ರಿಯೆಗಳು ಮತ್ತು ಸ್ವಯಂ ಸೇವೆ ಮತ್ತು ಅನುಮೋದನೆಗಳು ಸಿಸ್ಟಮ್ ಆಡಳಿತವನ್ನು ಮತ್ತಷ್ಟು ವಿತರಿಸುತ್ತವೆ. ಇವು ಅಂತಿಮ ಬಳಕೆದಾರರು ನಿರ್ವಹಿಸುವಂತೆ ಕೆಲವು ನೋಂದಣಿ ಕಾರ್ಯಗಳನ್ನು ಸ್ವಯಂಚಾಲಿತಗೊಳಿಸುತ್ತವೆ. ಹೊಸ ಬಳಕೆದಾರರ ಖಾತೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು, ಸಿಸ್ಟಮ್‌ಗೆ ಹೆಚ್ಚುವರಿ ಪ್ರವೇಶವನ್ನು ವಿನಂತಿಸುವುದು, ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ರಿಸೆಟ್ ಮಾಡುವುದು ಮುಂತಾದ ಕಾರ್ಯಗಳನ್ನು ಅಂತಿಮ ಬಳಕೆದಾರರು ನಿರ್ವಹಿಸುತ್ತಾರೆ.

ಸ್ವಯಂ ಸೇವೆ ಮತ್ತು ಅನುಮೋದನೆಗಳು : ಅಗತ್ಯತೆಗಳಿಗೆ ಅನುಗುಣವಾಗಿ ನೋಂದಣಿ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಕಾನ್ಫಿಗರ್ ಮಾಡಿದ ನಂತರ, ವ್ಯಕ್ತಿಗಳು ಸ್ವಯಂ ಸೇವೆಯ ನೋಂದಣಿ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಹುದು. ಉದಾಹರಣೆಗೆ, ಹೊಸ ಬಳಕೆದಾರರ ಖಾತೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು, ಅಥವಾ ಸಿಸ್ಟಮ್‌ಗೆ ಹೆಚ್ಚುವರಿ ಪ್ರವೇಶಕ್ಕೆ ವಿನಂತಿಸಿಕೊಳ್ಳುವುದು, ಜೊತೆಗೆ, ಸಂಸ್ಥೆಗಳು Oracle Approvals Management ಇಂಜಿನ್‌ನ್ನು ಬಳಸಬಹುದು. ಇದು ಈ ವಿನಂತಿಗಳಿಗೆ ಕಸ್ಟಮೈಸ್ ಮಾಡಿದ ಅನುಮೋದನೆಯ ರೂಟಿಂಗ್‌ನ್ನು ರಚಿಸುತ್ತದೆ.

ಮೂರು ರೀತಿಯ ತಡೆಗಟ್ಟುವಿಕೆಯ ನಿಯಂತ್ರಣಗಳಿವೆ.

ಆಡಳಿತಾತ್ಮಕ ನಿಯಂತ್ರಣಗಳು :

- ಪಾಲಿಸಿಗಳು/ಕಾರ್ಯವಿಧಾನಗಳು : ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಮಾರ್ಗವನ್ನು ಗುರುತಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ. ಇವು ತರಬೇತಿ, ಪತ್ತೆದಾರಿ ನಿಯಂತ್ರಣಗಳು, ಮತ್ತು ಆಡಿಟ್‌ಗಳು ಇವುಗಳ ಜೊತೆಗೆ ನಡೆಯಬೇಕು.

ಭೌತಿಕ ನಿಯಂತ್ರಣಗಳು

- ಬಯೊಮೆಟ್ರಿಕ್ ಸೆನ್ಸರ್‌ಗಳು, ಸ್ಮಾರ್ಟ್ ಕಾರ್ಡ್‌ಗಳು ಇತ್ಯಾದಿ

ತಾಂತ್ರಿಕ (ತಾರ್ಕಿಕ) ನಿಯಂತ್ರಣಗಳು

- ಎನ್‌ಕ್ರಿಪ್ಷನ್
- ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ಮತ್ತು ಟೋಕನ್‌ಗಳು
- ಬಯೊಮೆಟ್ರಿಕ್‌ಗಳು
- O.S. ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ ನಿಯಂತ್ರಣಗಳು
- ಗುರುತಿಸುವಿಕೆ ಮತ್ತು ದೃಢೀಕರಣ ತಂತ್ರಜ್ಞಾನಗಳು

© NIMI NOT TO BE REPUBLISHED

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಕಾರ್ಯವಿಧಾನಗಳು (Access Control Mechanisms)

ಪ್ರವೇಶ ನಿಯಂತ್ರಣಕ್ಕಾಗಿ, ಮಾದರಿ/ವಿಧಾನಗಳು ಕೆಳಗಿನಂತಿವೆ. ಮೇಲಿನ ಪ್ರತಿಯೊಂದು ಪ್ರವೇಶ ಮಾದರಿಗಳು ತನ್ನದೇ ಆದ ಅನುಕೂಲಗಳನ್ನು ಮತ್ತು ಅನಾನುಕೂಲಗಳನ್ನು ಹೊಂದಿವೆ. ಸಂಸ್ಥೆಯು ವಿವಿಧ ಅಂಶಗಳನ್ನು ಪರಿಗಣಿಸಿ ತನಗೆ ಸೂಕ್ತವಾದ ಪ್ರವೇಶ ಮಾದರಿಯನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ವ್ಯಾಪಾರದ ರೀತಿ, ಬಳಕೆದಾರರ ಸಂಖ್ಯೆ, ಸಂಸ್ಥೆಯ ಭದ್ರತಾ ಪಾಲಿಸಿ ಇತ್ಯಾದಿಗಳನ್ನು ಪರಿಗಣಿಸುತ್ತದೆ.

ಟೇಬಲ್ 2

ನಿಯಂತ್ರಣ ಸೇವೆ	ವಿವರಣೆ
ತಡೆಗಟ್ಟುವಿಕೆ	ಅನಪೇಕ್ಷಿತ ವಿಷಯಗಳು ನಡೆಯುವುದರಿಂದ ದೂರವಿಡಿ
ಡಿಟೆಕ್ಟಿವ್	ನಡೆದಿರುವ ಅನಪೇಕ್ಷಿತ ವಿಷಯಗಳನ್ನು ಗುರುತಿಸಿ
ಸರಿಪಡಿಸು	ನಡೆದಿರುವ ಅನಪೇಕ್ಷಿತ ವಿಷಯಗಳನ್ನು ಸರಿಪಡಿಸಿ
ತಡೆಯುವುದು	ಭದ್ರತಾ ಉಲ್ಲಂಘನೆಗಳು ನಡೆಯುವುದನ್ನು ನಿರುತ್ಸಾಹಗೊಳಿಸಿ
ರಿಕವರಿ	ಉಲ್ಲಂಘನೆ ಅಥವಾ ಅಪಘಾತದ ನಂತರ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಅಥವಾ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ರಿಸ್ಟೋರ್ ಮಾಡಿರಿ
ಪರಿಹಾರ	ಇತರ ನಿಯಂತ್ರಣಗಳಿಗೆ ಪರ್ಯಾಯಗಳನ್ನು ಒದಗಿಸಿ

- Role Based Access Control (RBAC)

ಪ್ರವೇಶ ನಿರ್ಧಾರಗಳು ಒಂದು ಸಂಸ್ಥೆಯ ಅಥವಾ ಬಳಕೆದಾರರ ನೆಲೆಯೊಳಗಿನ ವ್ಯಕ್ತಿಯ ರೋಲ್‌ನ್ನು ಮತ್ತು ಜವಾಬ್ದಾರಿಗಳನ್ನು ಆಧರಿಸಿವೆ. RBAC ಯನ್ನು non-discretionary Access Control ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ. ಏಕೆಂದರೆ ಬಳಕೆದಾರರು ತನ್ನ ರೋಲ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ಸವಲತ್ತುಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾನೆ. ಬಳಕೆದಾರನು ತನಗೆ ನಿಯೋಜಿಸಿದ ರೋಲ್ ಮೇಲೆ ನಿಯಂತ್ರಣವನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ.

- Discretionary Access Control (DAC)

ಹೆಸರೇ ಸೂಚಿಸುವಂತೆ, ಈ ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಮಾದರಿಯು ಬಳಕೆದಾರರ ವಿವೇಚನೆಯನ್ನು ಆಧರಿಸಿದೆ. ಅಂದರೆ, ಸಂಪನ್ಮೂಲದ ಮಾಲೀಕರು ಆ ಸಂಪನ್ಮೂಲದ ಮೇಲೆ ಇತರ ಬಳಕೆದಾರರಿಗೆ ತಮ್ಮ ವಿವೇಚನೆಯ ಮೇಲೆ ಪ್ರವೇಶ ಹಕ್ಕುಗಳನ್ನು ನೀಡಬಹುದು. Access Control Lists (ACLs) ಗಳು DAC ಯ ವಿಶಿಷ್ಟ ಉದಾಹರಣೆಗಳಾಗಿವೆ. DAC ಯ ಇನ್ನೊಂದು ಉದಾಹರಣೆಯೆಂದರೆ ನಿಮ್ಮ ಒಡೆತನದ ಯುನಿಕ್ಸ್ ಫೈಲ್‌ನ "rwx" ಅನುಮತಿಗಳನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುವುದು. ವಿಂಡೋಸ್ ಮತ್ತು ಯುನಿಕ್ಸ್‌ಗೆ ಸಂಬಂಧಿಸಿದವುಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಹೆಚ್ಚಿನ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು DAC ಮಾದರಿಯನ್ನು ಆಧರಿಸಿವೆ.

- Mandatory Access Control (MAC)

ಈ ಮಾದರಿಯಲ್ಲಿ, ಬಳಕೆದಾರರು/ ಮಾಲೀಕರು ತಮ್ಮ ಫೈಲ್‌ಗಳನ್ನು ಯಾರು ಪ್ರವೇಶಿಸಬಹುದು ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುವ ಸವಲತ್ತುಗಳನ್ನು ಆನಂದಿಸುವುದಿಲ್ಲ. ಇಲ್ಲಿ ಬಳಕೆದಾರರ ಇಚ್ಛೆಯನ್ನು ಮೀರಿ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ನಿರ್ಧಾರ ಮಾಡುತ್ತದೆ. ಈ ಮಾದರಿಯಲ್ಲಿ ಪ್ರತಿಯೊಂದು ಸಬ್ಜೆಕ್ಟ್ (ಬಳಕೆದಾರರು) ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್ (ಸಂಪನ್ಮೂಲಗಳು) ಗಳನ್ನು ಭದ್ರತಾ ಲೇಬಲ್‌ನೊಂದಿಗೆ ವರ್ಗೀಕರಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ನಿಯೋಜಿಸಲಾಗುತ್ತದೆ. ಭದ್ರತಾ ಪಾಲಿಸಿಯೊಂದಿಗಿನ ಸಬ್ಜೆಕ್ಟ್ ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್‌ನ ಭದ್ರತಾ ಲೇಬಲ್‌ಗಳು, ಸಬ್ಜೆಕ್ಟ್ ಆಬ್ಜೆಟ್‌ನ್ನು ಪ್ರವೇಶಿಸಬಲ್ಲದೇ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುತ್ತವೆ. ಸಬ್ಜೆಕ್ಟ್‌ಗಳು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಹೇಗೆ ಪ್ರವೇಶಿಸುತ್ತವೆ ಎನ್ನುವ ನಿಯಮಗಳನ್ನು ಭದ್ರತಾ ಅಧಿಕಾರಿಯು ಮಾಡುತ್ತಾರೆ, ಇದನ್ನು ಆಡಳಿತಾಧಿಕಾರಿಗಳು ಕಾನ್ಫಿಗರ್ ಮಾಡುತ್ತಾರೆ, ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್ ಜಾರಿಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಭದ್ರತಾ ತಂತ್ರಜ್ಞಾನಗಳು ಬೆಂಬಲಿಸುತ್ತವೆ.

DAC ಗೆ ಹೋಲಿಸಿದರೆ, ಇದು ಕಟ್ಟುನಿಟ್ಟಾದ ಮತ್ತು ಸ್ಥಿರವಾದ ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಮಾದರಿಯಾಗಿದೆ ಮತ್ತು ಮಿಲಿಟರಿ ಸಂಸ್ಥೆಗಳಿಗೆ ಹೆಚ್ಚು ಸೂಕ್ತವಾಗಿದೆ. ಏಕೆಂದರೆ ಇಲ್ಲಿ ಡಾಟಾ ವರ್ಗೀಕರಣ ಮತ್ತು ಗೌಪ್ಯತೆಯು ಪ್ರಧಾನ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ಹೊಂದಿವೆ. ವಿಶೇಷ ವಿಧದ ಯುನಿಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು MAC ಮಾದರಿಯ ಮೇಲೆ ಆಧಾರಿತವಾಗಿವೆ.

- Attribute Based Access Control (ABAC)

ಇದು ಬಳಕೆದಾರರ ಅನಿಯಂತ್ರಿತ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳ, ಆಬ್ಜೆಕ್ಟ್‌ನ ಅನಿಯಂತ್ರಿತ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳ ಆಧಾರದ ಮೇಲೆ, ಮತ್ತು ಪಾಲಿಸಿಗಳಿಗೆ ಹೆಚ್ಚು ಪ್ರಸ್ತುತವಾಗಿರಬಹುದಾದ ಪರಿಸರದ ಪರಿಸ್ಥಿತಿಗಳ ಮೇಲೆ, ಬಳಕೆದಾರರ ವಿನಂತಿಗಳನ್ನು ಅನುಮತಿಸುತ್ತದೆ ಅಥವಾ ನಿರಾಕರಿಸುತ್ತದೆ.

ಪಾಸ್‌ವರ್ಡ್ ಕ್ರ್ಯಾಕಿಂಗ್ ವಿಧಾನಗಳು ಮತ್ತು ಅವುಗಳ ವಿರುದ್ಧ ಕ್ರಮಗಳು (Password Cracking Methods And Their Counter measures)

ಖಾತೆಗಳನ್ನು ಹ್ಯಾಕ್ ಮಾಡಲು ಮತ್ತು ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ಕದಿಯಲು ಹ್ಯಾಕರ್‌ಗಳು ಹಲವಾರು ವಿಧಾನಗಳನ್ನು ಬಳಸುತ್ತಾರೆ. ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಕ್ರ್ಯಾಕ್ ಮಾಡಲು ಉಪಯೋಗಿಸುವ ಮತ್ತು ಅವುಗಳ ವಿರುದ್ಧ ತೆಗೆದುಕೊಳ್ಳುವ ಕ್ರಮಗಳಲ್ಲಿರುವ ಕೆಲವು ಸಾಮಾನ್ಯ ವಿಧಾನಗಳು :

© NIMI NOT TO BE REPUBLISHED

1 BruteForce ದಾಳಿ

BruteForce ದಾಳಿಯನ್ನು ಉಪಯೋಗಿಸಿ ಯಾವುದೇ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಕ್ರ್ಯಾಕ್ ಮಾಡಬಹುದು. BruteForce ದಾಳಿಗಳು ಸಂಖ್ಯೆಗಳ, ಅಕ್ಷರಗಳ, ಮತ್ತು ವಿಶೇಷ ಅಕ್ಷರಗಳ ಪ್ರತಿಯೊಂದು ಸಂಭವನೀಯ ಸಂಯೋಜನೆಗಳನ್ನು ಪಾಸ್‌ವರ್ಡ್ ಹೊಂದಾಣಿಕೆಯಾಗುವವರೆಗೂ ಪ್ರಯತ್ನಿಸುತ್ತದೆ. BruteForce ದಾಳಿಗಳು ಪಾಸ್‌ವರ್ಡ್‌ನ ಸಂಕೀರ್ಣತೆಯನ್ನು ಮತ್ತು ಕಂಪ್ಯೂಟರ್‌ನ ವೇಗವನ್ನು ಅವಲಂಬಿಸಿದಂತೆ ಬಹಳ ದೀರ್ಘ ಸಮಯವನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು.

ಪ್ರತಿತಂತ್ರ (Countermeasure): ದೀರ್ಘ ಮತ್ತು ಸಂಕೀರ್ಣ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ. ಸಣ್ಣ ಅಕ್ಷರಗಳು ಮತ್ತು ದೊಡ್ಡ ಅಕ್ಷರಗಳ (uppercase & lowercase letters) ಜೊತೆಗೆ ಸಂಖ್ಯೆಗಳನ್ನೂ ಸಹ ಉಪಯೋಗಿಸಿ. Brute-force ದಾಳಿಗಳು ಇಂತಹ ಸಂಕೀರ್ಣ ಮತ್ತು ಉದ್ದದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಭೇದಿಸಲು ಬಹಳ ಸಮಯವನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತವೆ. ನೀವು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಆಗಾಗ್ಗೆ ಬದಲಾಯಿಸುತ್ತಿರಬೇಕು.

2 **ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್** (Social Engineering)

ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್ ಎನ್ನುವುದು ಯಾರಾದರೂ ನಿಮ್ಮನ್ನು ನಂಬುವಂತೆ ಮಾಡುವುದು ಮತ್ತು ಅವರಿಂದ ಮಾಹಿತಿಯನ್ನು ಪಡೆಯುವಂತೆ ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ಒಬ್ಬ ಹ್ಯಾಕರ್ ತನ್ನ ಸಹೋದ್ಯೋಗಿಗಳ ಅಥವಾ ಸ್ನೇಹಿತರ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಪಡೆಯಲು ಪ್ರಯತ್ನಿಸುತ್ತಿರುತ್ತಾನೆ, ಅವನು ತನ್ನನ್ನು ಐಟಿ ಇಲಾಖೆಯವನು ಅಥವಾ ಬ್ಯಾಂಕ್‌ನವನು ಎಂದು ನಟಿಸಬಹುದು ಮತ್ತು ನಿಮ್ಮ ಲಾಗಿನ್ ಅಥವಾ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ವಿವರಗಳನ್ನು ಕೇಳುತ್ತಾನೆ.

ಪ್ರತಿತಂತ್ರ: ನಿಮ್ಮ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿಗಳಾದ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ವಿವರಗಳನ್ನು ಎಂದಿಗೂ ಫೋನ್‌ನಲ್ಲಿ ನೀಡಬೇಡಿ.

3 Rats **ಮತ್ತು** Keyloggers

ಕೀಲಾಗಿಂಗ್ ಅಥವಾ ರ್ಯಾಟಿಂಗ್‌ನಲ್ಲಿ, ಹ್ಯಾಕರ್ ಬಲಿಪಶುವಿಗೆ ಕೀಲಾಗರ್‌ನ್ನು ಅಥವಾ ರ್ಯಾಟನ್ನು ಕಳುಹಿಸುತ್ತಾನೆ. ಇದು ಹ್ಯಾಕರ್‌ಗೆ ಬಲಿಪಶು ತನ್ನ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಮಾಡುವ ಎಲ್ಲ ಕೆಲಸವನ್ನೂ ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡಲು ಅನುಮತಿಸುತ್ತದೆ. ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನೂ ಒಳಗೊಂಡಂತೆ ಪ್ರತಿ ಕೀಸ್ಟ್ರೋಕನ್ನು ಲಾಗ್ ಮಾಡಲಾಗುತ್ತದೆ. ಇದಲ್ಲದೇ ಹ್ಯಾಕರ್ ಬಲಿಪಶುವಿನ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಸಹ ನಿಯಂತ್ರಿಸಬಹುದು.

ಪ್ರತಿತಂತ್ರ: ನಿಮ್ಮ ಬ್ಯಾಂಕ್ ಖಾತೆಗೆ ಸೈಬರ್ ಕೆಫೆಯಿಂದ ಅಥವಾ ಬೇರೆಯವರ ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಲಾಗಿನ್ ಆಗಬೇಡಿರಿ. ಇದು ತುಂಬಾ ಮುಖ್ಯವಾಗಿದ್ದರೆ, ಲಾಗಿನ್ ಮಾಡಲು ಆನ್- ಸ್ಕ್ರೀನ್ ಅಥವಾ ವರ್ಚುವಲ್ ಕೀಬೋರ್ಡ್‌ನ್ನು ಬಳಸಿರಿ. ಇತ್ತೀಚಿನ ಆ್ಯಂಟಿ-ವೈರಸ್ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಬಳಸಿ ಮತ್ತು ವ್ಯಾಖ್ಯಾನಗಳನ್ನು ನವೀಕರಿಸಿರಿ.

4 **ಫಿಶಿಂಗ್** (Phishing)

ಫಿಶಿಂಗ್ ಅತ್ಯಂತ ಸುಲಭವಾದ ಮತ್ತು ಜನಪ್ರಿಯವಾದ ಹ್ಯಾಕಿಂಗ್ ವಿಧಾನವಾಗಿದೆ. ಇದನ್ನು ಹ್ಯಾಕರ್‌ಗಳು ಇತರರ ಖಾತೆಯ ವಿವರಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾರೆ. ಫಿಶಿಂಗ್ ದಾಳಿಯಲ್ಲಿ ಹ್ಯಾಕರ್, ಸಂತ್ರಸ್ತರಿಗೆ ಫೇಸ್‌ಬುಕ್, ಜಿಮೇಲ್‌ನಂತಹ ನಿಜವಾದ ವೆಬ್‌ಸೈಟಿನ ನಕಲಿ ಪುಟವನ್ನು ಕಳುಹಿಸುತ್ತಾನೆ. ಆ ನಕಲಿ ಪುಟವನ್ನು ಯಾರಾದರೂ ಲಾಗಿನ್ ಮಾಡಿದಾಗ ಅವರ ವಿವರಗಳನ್ನು ಹ್ಯಾಕರ್‌ಗೆ ಕಳುಹಿಸಲಾಗುತ್ತದೆ. ಈ ನಕಲಿ ಪುಟಗಳನ್ನು ಉಚಿತ ವೆಬ್ ಹೋಸ್ಟಿಂಗ್ ಸೈಟ್‌ಗಳಲ್ಲಿ ಸುಲಭವಾಗಿ ರಚಿಸಬಹುದು ಮತ್ತು ಹೋಸ್ಟ್ ಮಾಡಬಹುದು.

ಪ್ರತಿತಂತ್ರ: ಫಿಶಿಂಗ್ ದಾಳಿಗಳನ್ನು ತಪ್ಪಿಸುವುದು ಬಹಳ ಸುಲಭವಾಗಿರುತ್ತದೆ. ಈ ಫಿಶಿಂಗ್ ಪುಟಗಳ url , ನಿಜವಾದ ಪುಟಗಳಿಗಿಂತ ಭಿನ್ನವಾಗಿರುತ್ತವೆ. ಉದಾಹರಣೆಗೆ, ಫೇಸ್‌ಬುಕ್‌ನ ಫಿಶಿಂಗ್ ಪುಟದ URL facbbook.com ಈ ರೀತಿಯಾಗಿ ಕಾಣಬಹುದು (ಇಲ್ಲಿ ನೀವು ನೋಡುವಂತೆ ಎರಡು “b” ಗಳು ಇವೆ.) ಯಾವಾಗಲೂ ವೆಬ್‌ಸೈಟ್‌ಗಳ url ಸರಿಯಾಗಿದೆಯೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

5 **ರೇನ್‌ಬೋ ಟೇಬಲ್** (Rainbow Table)

ರೇನ್ ಬೋ ಟೇಬಲ್, ಪ್ರತಿಯೊಂದು ಸಂಭವನಿಯ ಅಕ್ಷರಗಳ ಸಂಯೋಜನೆಗಾಗಿ ಹ್ಯಾಶ್‌ಗಳ ಬೃಹತ್ ಪ್ರಿ- ಕಂಪ್ಯೂಟೆಡ್ ಪಟ್ಟಿಯಾಗಿದೆ. ಪಾಸ್‌ವರ್ಡ್ ಹ್ಯಾಶ್ ಎಂದರೆ, md5 ನಂತಹ ಗಣಿತದ ಅಲ್ಗೊರಿದಮ್ ಮೂಲಕ ತಯಾರಿಸಲಾದ ಪಾಸ್‌ವರ್ಡ್ ಆಗಿದೆ. ಮತ್ತು ಇದನ್ನು ಗುರುತಿಸಲಾಗದಂತೆ ಪರಿವರ್ತಿಸಲಾಗುತ್ತದೆ. ಹ್ಯಾಶ್ ಒಂದೇ ಮಾರ್ಗದ ಎನ್‌ಕ್ರಿಪ್ಷನ್ ಆಗಿದೆ. ಅದ್ದರಿಂದ ಒಮ್ಮೆ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಹ್ಯಾಶ್ ಮಾಡಿದ ನಂತರ ಹ್ಯಾಶ್ ಮಾಡಲಾದ ಸ್ಟ್ರಿಂಗ್‌ನಿಂದ ಮೂಲ ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಪಡೆಯಲು ಯಾವುದೇ ಮಾರ್ಗವಿರುವುದಿಲ್ಲ.

ಪ್ರತಿತಂತ್ರ : ನೀವು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುವ ಪಾಸ್‌ವರ್ಡ್ ಉದ್ದವಾಗಿರಬೇಕು ಮತ್ತು ಸಂಕೀರ್ಣವಾಗಿರಬೇಕು ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿರಿ. ಉದ್ದ ಮತ್ತು ಸಂಕೀರ್ಣ ಪಾಸ್‌ವರ್ಡ್‌ಗಳಿಗಾಗಿ ಟೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು ಬಹಳ ಸಮಯವನ್ನು ಮತ್ತು ಸಾಕಷ್ಟು ಸಂಪನ್ಮೂಲಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.

6 **ಊಹಿಸುವುದು** (Guessing)

ಇದು ಯಾರದಾದರೂ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಸೆಕೆಂಡುಗಳಲ್ಲಿ ಪಡೆದುಕೊಳ್ಳಲು ನಿಮಗೆ ಸಹಾಯ ಮಾಡುವ ಸರಳ ವಿಧಾನವಾಗಿದೆ. ಹ್ಯಾಕರ್ ನಿಮಗೆ ತಿಳಿದಿದ್ದರೆ, ಅವರು ನಿಮ್ಮ ಬಗ್ಗೆ ತಿಳಿದಿರುವ

© NIMI NOT TO BE REPUBLISHED

ಮಾಹಿತಿಯನ್ನು ಬಳಸಿ ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಊಹಿಸುತ್ತಾರೆ. ಹ್ಯಾಕರ್‌ಗಳು ಸಾಮಾಜಿಕ ಇಂಜಿನಿಯರಿಂಗ್ ಮತ್ತು ಗೆಸಿಂಗ್ ಇವುಗಳ ಸಂಯೋಜನೆಯನ್ನು ಬಳಸಿ ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಪ್ರತಿತಂತ್ರ : ನಿಮ್ಮ ಹೆಸರು, ಉಪನಾಮ, ಫೋನ್ ನಂಬರ್, ಅಥವಾ ಜನ್ಮ ದಿನಾಂಕಗಳನ್ನು ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನಾಗಿ ಬಳಸಬೇಡಿರಿ. ನಿಮಗೆ ಸಂಬಂಧಿಸಿದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ರಚಿಸುವುದನ್ನು ತಪ್ಪಿಸಲು ಪ್ರಯತ್ನಿಸಿ. ಅಕ್ಷರಗಳ ಮತ್ತು ಸಂಖ್ಯೆಗಳ ಸಂಯೋಜನೆಯೊಂದಿಗೆ ಸಂಕೀರ್ಣ ಮತ್ತು ಉದ್ದದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ರಚಿಸಿ.

ಮಾಹಿತಿಯ ಭದ್ರತಾ ವರ್ಗೀಕರಣ (Security classification for information)

ಎಲ್ಲ ಮಾಹಿತಿಯೂ ಸಮಾನಾಗಿರುವುದಿಲ್ಲ ಮತ್ತು ಅದೇ ರೀತಿ ಮಾಹಿತಿಗೂ ಒಂದೇ ರೀತಿಯ ರಕ್ಷಣೆಯ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ಮಾಹಿತಿಗೆ ಭದ್ರತಾ ವರ್ಗೀಕರಣವನ್ನು ನಿಯೋಜಿಸಬೇಕಾದ ಅಗತ್ಯವಿದೆ.

ಮಾಹಿತಿ ವರ್ಗೀಕರಣದ ಮೊದಲ ಹಂತವೆಂದರೆ ಹಿರಿಯ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್‌ನ ಸದಸ್ಯರನ್ನು, ವರ್ಗೀಕರಿಸಬೇಕಾದ ನಿರ್ದಿಷ್ಟ ಮಾಹಿತಿಯ ಮಾಲೀಕರಾಗಿ ಗುರುತಿಸುವುದು. ನಂತರ ವರ್ಗೀಕರಣದ ಪಾಲಿಸಿಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಿ. ಪಾಲಿಸಿಯು ಕೆಳಗಿನವುಗಳನ್ನು ಸಮರ್ಥಿಸಬೇಕು:

- ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವುದು
- ಮಾಹಿತಿಯನ್ನು ರವಾನೆಮಾಡುವುದು
- ವಿವಿಧ ವರ್ಗೀಕರಣದ ಲೇಬಲ್‌ಗಳನ್ನು ವಿವರಿಸುವುದು
- ನಿರ್ದಿಷ್ಟ ಲೇಬಲ್‌ನ್ನು ನಿಯೋಜಿಸುವ ಮಾಹಿತಿಗೆ ಮಾನದಂಡವನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು.
- ಪ್ರತಿಯೊಂದು ವರ್ಗೀಕರಣಕ್ಕೂ ಅವಶ್ಯಕವಿರುವ ಭದ್ರತಾ ನಿಯಂತ್ರಣಗಳನ್ನು ಪಟ್ಟಿ ಮಾಡಿ.
- ಅನಗತ್ಯ ಮಾಹಿತಿಯನ್ನು ವಿಲೇವಾರಿ ಮಾಡುವುದು
- ಮಾಹಿತಿಯ ಸಮಗ್ರತೆಯನ್ನು ರಕ್ಷಿಸುವುದು
- ಸೂಕ್ತ ಪ್ರವೇಶ ಮತ್ತು ಬಹಿರಂಗಪಡಿಸುವಿಕೆಯನ್ನು ಅನುಮತಿಸುವುದು.
- ಹೊಣೆಗಾರಿಕೆಯನ್ನು ಸ್ಥಾಪಿಸುವುದು.

ಯಾವ ಮಾಹಿತಿಯ ವರ್ಗೀಕರಣವನ್ನು ನಿಯೋಜಿಸಬೇಕು ಎನ್ನುವುದರ ಮೇಲೆ ಪ್ರಭಾವ ಬೀರುವ ಕೆಲವು ಅಂಶಗಳು :

- ಸಂಸ್ಥೆಗೆ ಆ ಮಾಹಿತಿಯು ಎಷ್ಟು ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ.
- ಮಾಹಿತಿ ಎಷ್ಟು ಹಳೆಯದು ಮತ್ತು
- ಮಾಹಿತಿಯು ಹಳೆಯದಾಗಿದೆಯೆ ಅಥವಾ ಇಲ್ಲವೆ
- ಕಾನೂನುಗಳು ಮತ್ತು ಇತರ ನಿಯಂತ್ರಕ ಅವಶ್ಯಕತೆಗಳು

ಮಾಹಿತಿ ಭದ್ರತೆಗಾಗಿ ಇರುವ ವ್ಯಾಪಾರ ಮಾದರಿಯು ಭದ್ರತಾ ವೃತ್ತಿಪರರನ್ನು, ಸಿಸ್ಟಮ್ ದೃಷ್ಟಿಕೋನದಿಂದ ಭದ್ರತೆಯನ್ನು ಪರೀಕ್ಷಿಸಲು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ಇದು ಭದ್ರತೆಯನ್ನು ಸಮಗ್ರವಾಗಿ ನಿರ್ವಹಿಸಲು ಪರಿಸರವನ್ನು ರಚಿಸುತ್ತದೆ ಮತ್ತು ನಿಜವಾದ ಅಪಾಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

ಆಯ್ಕೆಯಾದ ಮತ್ತು ಬಳಸಲಾದ ಮಾಹಿತಿ ಭದ್ರತಾ ವರ್ಗೀಕರಣ ಲೇಬಲ್‌ಗಳ ವಿಧವು, ಸಂಸ್ಥೆಯ ಸ್ವರೂಪವನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ. ಉದಾಹರಣೆಗಳು :

- ವ್ಯಾಪಾರ ವಲಯದಲ್ಲಿ, ಲೇಬಲ್‌ಗಳು : ಸಾರ್ವಜನಿಕ, ಸೂಕ್ಷ್ಮ, ಖಾಸಗಿ, ಮತ್ತು ಗೌಪ್ಯ.
- ಸರ್ಕಾರಿ ವಲಯದಲ್ಲಿ, ಲೇಬಲ್‌ಗಳು ಉದಾ: ವರ್ಗೀಕರಿಸದ, ಸೂಕ್ಷ್ಮ ಆದರೆ ವರ್ಗೀಕರಿಸದ, ನಿರ್ಬಂಧಿತ, ಗೌಪ್ಯ, ರಹಸ್ಯ, ಅತ್ಯಂತ ರಹಸ್ಯ, ಮತ್ತು ಅವುಗಳ ಇಂಗ್ಲಿಷ್ ಅಲ್ಲದ ಸಮಾನತೆಗಳು
- ಕ್ರಾಸ್ ಸೆಕ್ಟೋರಿಯಲ್ ರಚನೆಗಳಲ್ಲಿ, ಟ್ರಾಫಿಕ್ ಲೈಟ್ ಪ್ರೊಟೋಕಾಲ್ : ಬಿಳಿ, ಹಸಿರು, ಆ್ಯಂಬರ್ ಮತ್ತು ಕೆಂಪು

ಒಂದು ಸಂಸ್ಥೆಯಲ್ಲಿರುವ ಎಲ್ಲ ಉದ್ಯೋಗಿಗಳು ಮತ್ತು ವ್ಯಾಪಾರದ ಪಾಲುದಾರರು ವರ್ಗೀಕರಣ ಯೋಜನೆಯಲ್ಲಿ ತರಬೇತಿ ಪಡೆಯಬೇಕು. ಮತ್ತು ಅಗತ್ಯವಿರುವ ಭದ್ರತಾ ನಿಯಂತ್ರಣಗಳನ್ನು ಅರ್ಥಮಾಡಿಕೊಳ್ಳುವುದು. ಮತ್ತು ಪ್ರತಿ ವರ್ಗೀಕರಣದ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು. ಒಂದು ನಿರ್ದಿಷ್ಟ ಮಾಹಿತಿಯ ಆಸ್ತಿಯ ವರ್ಗಾವಣೆಯನ್ನು ಸಮಯಾನುಸಾರ ಪರಿಶೀಲಿಸಲಾಗುತ್ತದೆ. ಇದು ಮಾಹಿತಿಯ ವರ್ಗೀಕರಣವು ಸೂಕ್ತವಾಗಿದೆ ಎಂದು ಖಚಿತಪಡಿಸುತ್ತದೆ ಮತ್ತು ವರ್ಗೀಕರಣಕ್ಕೆ ಅಗತ್ಯವಿರುವ ಭದ್ರತಾ ನಿಯಂತ್ರಣಗಳು ಸರಿಯಾದ ಸ್ಥಳದಲ್ಲಿವೆ ಎಂದೂ ಖಚಿತಪಡಿಸುತ್ತವೆ.

Declassifying and downgrading

ಮಾಹಿತಿಯನ್ನು ಅದಕ್ಕೆ ರಕ್ಷಣೆಯ ಅವಶ್ಯಕತೆ ಇರುವ ಸಮಯದಲ್ಲಿ ಮಾತ್ರ ವರ್ಗೀಕರಿಸಬೇಕು ಮತ್ತು ಗೊತ್ತುಪಡಿಸಬೇಕು. ಅದರ ನಂತರ ಅದನ್ನು ಬಹಿರಂಗಗೊಳಿಸಬೇಕು ಮತ್ತು ಡೌನ್‌ಗ್ರೇಡ್ ಮಾಡಬೇಕು. ಏಕೆಂದರೆ ವರ್ಗೀಕರಣಗೊಂಡ ಅಥವಾ ಗೊತ್ತುಪಡಿಸಿದ ಮಾಹಿತಿಯು ಸಮಯದೊಂದಿಗೆ ಅಥವಾ ನಿರ್ದಿಷ್ಟ ಘಟನೆಗಳ ಸಂಭವಿಸುವಿಕೆಯಿಂದ ಅದರ ಸೂಕ್ಷ್ಮತೆಯನ್ನು ಕಳೆದುಕೊಳ್ಳುತ್ತದೆ. ಈ ಪ್ರಕ್ರಿಯೆಯು ಭದ್ರತಾ ಸಿಸ್ಟಮ್‌ನ ಒಟ್ಟೂ ಸಮಗ್ರತೆಗೆ ಕೊಡುಗೆ ನೀಡುತ್ತದೆ. ಮತ್ತು ಸಾರ್ವಜನಿಕ ಆಸಕ್ತ ಸದಸ್ಯರಿಗೆ ಮಾಹಿತಿ ಲಭ್ಯವಾಗುವದನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಆಡಳಿತ (Access Control Administration)

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಆಡಳಿತವನ್ನು ಎರಡು ರೀತಿಯಲ್ಲಿ ಮಾಡಬಹುದು :

- **ಕೇಂದ್ರೀಕೃತ (Centralized)** : ಇಲ್ಲಿ ಒಂದು ಘಟಕವು (ಇಲಾಖೆ ಅಥವಾ ವ್ಯಕ್ತಿ) ಎಲ್ಲ ಕಾರ್ಪೋರೇಟ್ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಪ್ರವೇಶದ ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡುವ ಜವಾಬ್ದಾರಿಯನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಈ ರೀತಿಯ ಆಡಳಿತವು ಬಳಕೆದಾರರ ಪ್ರವೇಶದ ಹಕ್ಕುಗಳ ಸ್ಥಿರ ಮತ್ತು ಏಕರೂಪದ ವಿಧಾನವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ : RADIUS, TACACS ಮತ್ತು Diameter.

- **ವಿಕೇಂದ್ರೀಕೃತ (Decentralized)** : ಪ್ರವೇಶ ನಿಯಂತ್ರಣ / ಡಾಟಾ ಸಂಗ್ರಹಣೆ ಪ್ರೊಟೊಕಾಲ್‌ಗಳು : AAA (RADIUS, Diameter, ಮತ್ತು TACACS+)

RADIUS, Diameter, and TACACS+ ಈ ಮೂರು ಪ್ರೊಟೊಕಾಲ್‌ಗಳು ನೆಟ್‌ವರ್ಕ್ ಎಕ್ಸೆಸ್ ಸರ್ವರ್ (NAS) ನಡುವೆ Authentication, Authorization, ಮತ್ತು Accounting (AAA) ಈ ಮಾಹಿತಿಯನ್ನು ಸಾಗಿಸುತ್ತವೆ. ಇದು ತನ್ನ ಲಿಂಕ್‌ಗಳು ಅಥವಾ ಅಂತಿಮ ಬಳಕೆದಾರರನ್ನು ದೃಢೀಕರಿಸಲು ಬಯಸುತ್ತದೆ.

ರೇಡಿಯಸ್ (RADIUS)

Remote Authentication Dial-In User Service (RADIUS) ಇದು ಕ್ಲೈಂಟ್ /ಸರ್ವರ್ ಭದ್ರತಾ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿದೆ. Lucent InterNetworking Systems ನಿಂದ ರಚಿಸಲ್ಪಟ್ಟಿದೆ. RADIUS ಇದು ಇಂಟರ್‌ನೆಟ್ ಡ್ರಾಫ್ಟ್ ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿದೆ. ಬಳಕೆದಾರರ ಪ್ರೊಫೈಲ್‌ಗಳನ್ನು RADIUS ಸರ್ವರ್ ಎಂದು ಕರೆಯಲ್ಪಡುವ ಕೇಂದ್ರ ಸ್ಥಳದಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ. RADIUS ಕ್ಲೈಂಟ್‌ಗಳು(ಉದಾಹರಣೆಗೆ PortMaster communications server) ಬಳಕೆದಾರರನ್ನು ದೃಢಿಕರಿಸಲು RADIUS ಸರ್ವರ್‌ನೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸುತ್ತವೆ. ಸರ್ವರ್, ಮತ್ತೆ ಕ್ಲೈಂಟ್‌ಗೆ ದೃಢಿಕರಿಸಲ್ಪಟ್ಟ ಬಳಕೆದಾರನು ಯಾವ ಕೆಲಸವನ್ನು ಮಾಡಲು ಅಧಿಕಾರ ಹೊಂದಿರುವನು ಎಂಬುದನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತದೆ. RADIUS ಪದವು ಕ್ಲೈಂಟ್ ಮತ್ತು ಸರ್ವರ್‌ಗಳು ಸಂವಹನ ನಡೆಸಲು ಬಳಸುವ ನೆಟ್‌ವರ್ಕ್ ಪ್ರೊಟೊಕಾಲನ್ನು ಸೂಚಿಸಿದರೂ, ಹೆಚ್ಚಾಗಿ ಇದನ್ನು ಸಂಪೂರ್ಣ ಕ್ಲೈಂಟ್ /ಸರ್ವರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಸೂಚಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಡಯಾಮೀಟರ್ (Diameter): ಡಯಾಮೀಟರ್ ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳಿಗಾಗಿ ದೃಢೀಕರಣ, ಅಧಿಕಾರ ಮತ್ತು ಅಕೌಂಟಿಂಗ್ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿದೆ. ಇದು RADIUS ಪ್ರೊಟೊಕಾಲ್‌ನಿಂದ ವಿಕಸನಗೊಂಡಿದೆ ಮತ್ತು ಕಡಿಮೆ ಸಾಮರ್ಥ್ಯ ಹೊಂದಿರುವ RADIUS ಪ್ರೊಟೊಕಾಲ್‌ನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ. ಡಯಾಮೀಟರ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಹೊಸ ಕಮಾಂಡ್‌ಗಳನ್ನು ಮತ್ತು/ಅಥವಾ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದರ ಮೂಲಕ ಬೇಸ್ ಪ್ರೊಟೊಕಾಲನ್ನು ವಿಸ್ತರಿಸುತ್ತದೆ. RADIUS ಗೆ ಹೋಲಿಸಿದರೆ ಇದು, ಉತ್ತಮ ಸಾರಿಗೆ, ಉತ್ತಮ ಭದ್ರತೆ, ಉತ್ತಮ ಪ್ರಾಕ್ಸಿಯಿಂಗ್, ಉತ್ತಮ ಸೆಷನ್ ನಿಯಂತ್ರಣ, ಮತ್ತು ಉತ್ತಮ ಪರಸ್ಪರ ಕಾರ್ಯಸಾಧ್ಯತೆಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

TACACS: Terminal Access Controller Access-Control System (TACACS ಸಾಮಾನ್ಯವಾಗಿ tack-axe ಎಂದು ಉಚ್ಚರಿಸಲಾಗುತ್ತದೆ) ಕೇಂದ್ರೀಕೃತ ಸರ್ವರ್ ಮೂಲಕ ನೆಟ್‌ವರ್ಕ್ ಮಾಡಲಾದ ಪ್ರವೇಶ ಕಂಟ್ರೋಲ್‌ಗಳಿಗಾಗಿ ರಿಮೋಟ್ ದೃಢೀಕರಣ ಮತ್ತು ಸಂಬಂಧಿತ ಸೇವೆಗಳನ್ನು ನಿರ್ವಹಿಸುವ ಸಂಬಂಧಿತ ಪ್ರೊಟೊಕಾಲ್‌ಗಳ ಕುಟುಂಬವನ್ನು ಸೂಚಿಸುತ್ತದೆ. 1984 ರ ಮೂಲ TACACS ಪ್ರೊಟೊಕಾಲ್‌ನ್ನು ದೃಢಿಕರಿಸಿದ ಸರ್ವರ್‌ನೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಲು ಉಪಯೋಗಿಸಲಾಯಿತು. ಇದು ಹಳೆಯ ಯುನಿಕ್ಸ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಬಳಕೆಯಾಗುತ್ತಿತ್ತು. Extended TACACS (XTACACS) ಇದು 1990 ರಲ್ಲಿ ಸಿಸ್ಕೊ ಸಿಸ್ಟಮ್‌ನಿಂದ ಪರಿಚಯಿಸಲ್ಪಟ್ಟ ಮೂಲ ಪ್ರೊಟೊಕಾಲ್‌ಗೆ ಹೊಂದಾಣಿಕೆ ಇಲ್ಲದ ವಿಸ್ತರಣೆಯಾಗಿದೆ. Terminal Access Controller Access-Control System Plus (TACACS+) ಇದು 1993 ರಲ್ಲಿ ಸಿಸ್ಕೊನಿಂದ ಅಬಿವೃದ್ಧಿ ಹೊಂದಿದ ಮತ್ತು ಮುಕ್ತ ಗುಣಮಟ್ಟದ ಪ್ರೊಟೊಕಾಲ್ ಆಗಿ ಬಿಡುಗಡೆಯಾಯಿತು.

ವಿಕೇಂದ್ರೀಕೃತ ಪ್ರವೇಶ ನಿಯಂತ್ರಣ (Decentralized Access Control)

- ವಿಕೇಂದ್ರೀಕೃತ ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಆಡಳಿತ ವಿಧಾನವು ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಹತ್ತಿರವಿರುವ ಜನರಿಗೆ ಪ್ರವೇಶದ ನಿಯಂತ್ರಣವನ್ನು ನೀಡುತ್ತದೆ.
- ಈ ವಿಧಾನದಲ್ಲಿ, ಉದ್ಯೋಗಿಗಳಿಗೆ ಸಾಮಾನ್ಯವಾಗಿ ಕಾರ್ಯಕಾರಿ ನಿರ್ವಾಹಕರು ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ಹಕ್ಕುಗಳನ್ನು ನೀಡುತ್ತಾರೆ.
- ಈ ರೀತಿಯ ಆಡಳಿತದಲ್ಲಿ ಬದಲಾವಣೆಗಳು ವೇಗವಾಗಿ ಸಂಭವಿಸುತ್ತವೆ ಏಕೆಂದರೆ ಇಡೀ ಸಂಸ್ಥೆಗೆ ಕೇವಲ ಒಂದೇ ಘಟಕವು ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುತ್ತಿರುವುದಿಲ್ಲ.
- ಸಂಸ್ಥೆಗೆ ಲಾಭ ತರದೇ ಇರುವ ಸಂಘರ್ಷಗಳು ಉಂಟಾಗುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ ಏಕೆಂದರೆ ವಿವಿಧ ವ್ಯವಸ್ಥಾಪಕರು ಮತ್ತು ಇಲಾಖೆಗಳು ವಿಭಿನ್ನ ರೀತಿಯಲ್ಲಿ ಭದ್ರತೆಯನ್ನು ಮತ್ತು ಪ್ರವೇಶ ನಿಯಂತ್ರಣವನ್ನು ಅಭ್ಯಾಸ ಮಾಡಬಹುದು.
- ಕ್ರಮಗಳನ್ನು ಸರಿಯಾಗಿ ನಿಷೇಧಿಸಲು ಅಥವಾ ನಿರ್ಬಂಧಿಸಲಾಗದೇ ಇರುವ ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ, ನಿರ್ದಿಷ್ಟ ನಿಯಂತ್ರಣಗಳು ಅತಿಕ್ರಮಿಸುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ.

NOT TO BE REPUBLISHED © NIMI

- ಈ ರೀತಿಯ ಆಡಳಿತವು ಸ್ಥಿರ ನಿಯಂತ್ರಣಕ್ಕೆ ಕೇಂದ್ರೀಕೃತ ವಿಧಾನದಂತೆ, ವಿಧಾನಗಳನ್ನು ಒದಗಿಸುವುದಿಲ್ಲ.

ಭದ್ರತಾ ಆಡಿಟ್: ಇದು ಒಂದು ಸಂಸ್ಥೆಯು ತನ್ನ ಭದ್ರತಾ ಪಾಲಿಸಿಯನ್ನು ಎಷ್ಟು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಅನುಷ್ಠಾನಗೊಳಿಸಿದೆ ಎಂಬುದರ ಮೌಲ್ಯಮಾಪನವಾಗಿದೆ. ಇದು ಒಂದು ಸ್ವತಂತ್ರ ವಿಮರ್ಶೆಯಾಗಿದೆ ಮತ್ತು ಐಟಿ ಸಿಸ್ಟಮ್‌ಗಳ ಪಾಲಿಸಿ, ರೆಕಾರ್ಡ್‌ಗಳು ಮತ್ತು ಚಟುವಟಿಕೆಗಳ ಪರೀಕ್ಷೆಯಾಗಿದೆ.

ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳ ಆಡಿಟ್ ಮುಖ್ಯವಾಗಿದೆ. ಏಕೆಂದರೆ, ಐಟಿ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಸಮರ್ಪಕವಾಗಿ ರಕ್ಷಿಸಲಾಗಿದೆ ಎನ್ನುವ ಭರವಸೆಯನ್ನು ನೀಡುತ್ತದೆ, ಬಳಕೆದಾರರಿಗೆ ವಿಶ್ವಾಸಾರ್ಹ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ, ತಮ್ಮ ಉದ್ದೇಶಿತ ಲಾಭಗಳನ್ನು ಸಾಧಿಸಲು ಸರಿಯಾಗಿ ನಿರ್ವಹಣೆ ಮಾಡುತ್ತದೆ. ಇದು ಐಟಿ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಡಾಟಾ ಬದಲಾವಣೆಯ ಅಪಾಯವನ್ನು, ಡಾಟಾ ನಷ್ಟ ಅಥವಾ ಸೋರಿಕೆಯನ್ನು, ಸೇವೆಗಳ ಅಡ್ಡಿಯನ್ನು, ಮತ್ತು ಕಳಪೆ ನಿರ್ವಹಣೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

Security Auditing and Testing (SAT) ಭದ್ರತಾ ಆಡಿಟಿಂಗ್ ಮತ್ತು ಪರೀಕ್ಷೆಯು ಆಂತರಿಕ ಕಾರಣಗಳಿಗಾಗಿ ಇರುವ ಭದ್ರತೆಯ ಸ್ಥಿತಿಯನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳಲು ಸಂಸ್ಥೆಗೆ ಸಹಾಯ ಮಾಡುತ್ತದೆ ಮತ್ತು ಬಾಹ್ಯ ಪಕ್ಷಗಳಿಗೆ ಭರವಸೆಯನ್ನು ನೀಡುತ್ತದೆ, ಇದಕ್ಕೆ ಉನ್ನತ ಮಟ್ಟದ ಗಮನದ ಅಗತ್ಯವಿದೆ. ಇದು ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ರಕ್ಷಣೆಗಳ ನಡುವಿನ ಅಂತರವನ್ನು ಗುರುತಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಆಡಿಟ್‌ನ ಉದ್ದೇಶಗಳನ್ನು ಸ್ಥಾಪಿಸುವುದು (Establishing audit objectives)

ಒಂದು ಆಡಿಟ್‌ನ್ನು ಯೋಜಿಸಿದ ನಂತರ ಮತ್ತು ಆಡಿಟ್‌ನ್ನು ನಿರ್ವಹಿಸಲು ಮುಂದುವರೆಯುವ ಮೊದಲು ಆಡಿಟ್‌ನ ಉದ್ದೇಶಗಳನ್ನು ಸ್ಥಾಪಿಸಬೇಕು. ಆಡಿಟರ್‌ಗಳು ಪರಿಶೀಲಿಸಬೇಕಾದ ಉದ್ದೇಶಗಳ ಪಟ್ಟಿ ಈ ಕೆಳಗಿನಂತಿದೆ :

- ವ್ಯವಸ್ಥೆಗಳು ಮತ್ತು ಕ್ರಾಸ್ - ಫಂಕ್ಷನಲ್ ತರಬೇತಿಯ ಜೊತೆಗೆ ಸಿಬ್ಬಂದಿ ಕಾರ್ಯವಿಧಾನಗಳು ಮತ್ತು ಜವಾಬ್ದಾರಿಗಳು
- ಚೇಂಜ್ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್ ಪ್ರಕ್ರಿಯೆಗಳು ಸ್ಥಳದಲ್ಲಿವೆ ಮತ್ತು ಐಟಿ ಮತ್ತು ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್ ಸಿಬ್ಬಂದಿಯಿಂದ ಅನುಸರಿಸಲಾಗುತ್ತಿದೆ. ಚೇಂಜ್ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್ ಎಂದರೆ ಐಟಿ ಮೂಲಸೌಕರ್ಯವನ್ನು ನಿಯಂತ್ರಿಸಲು ಎಲ್ಲ ಬದಲಾವಣೆಗಳನ್ನು ಸಮರ್ಥವಾಗಿ ಮತ್ತು ಕ್ಷಿಪ್ರವಾಗಿ ನಿರ್ವಹಣೆ ಮಾಡುವುದನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ಸೇವೆಯ ಮೇಲೆ ನಡೆಯುವ ಯಾವುದೇ ಸಂಬಂಧಿತ ಘಟನೆಗಳ ಸಂಖ್ಯೆಯನ್ನು ಮತ್ತು ಪ್ರಭಾವವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.
- ಅಲಭ್ಯತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡಲು ಮತ್ತು ಪ್ರಮುಖ ಡಾಟಾದ ನಷ್ಟವನ್ನು ತಡೆಯಲು, ಸೂಕ್ತವಾದ ಬ್ಯಾಕಪ್ ಕಾರ್ಯವಿಧಾನಗಳು ಸ್ಥಳದಲ್ಲಿವೆ.
- ಡಾಟಾ ಸೆಂಟರ್‌ಗೆ ಅನಧಿಕೃತ ಪ್ರವೇಶವನ್ನು ತಡೆಯಲು, ಡಾಟಾ ಸೆಂಟರ್ ಸಾಕಷ್ಟು ಭೌತಿಕ ಭದ್ರತಾ ನಿಯಂತ್ರಣಗಳನ್ನು ಹೊಂದಿದೆ.
- ಬೆಂಕಿ ಮತ್ತು ಪ್ರವಾಹದಿಂದ ಉಪಕರಣಗಳನ್ನು ರಕ್ಷಿಸಲು ಸಾಕಷ್ಟು ಪರಿಸರ ನಿಯಂತ್ರಣಗಳು ಸ್ಥಳದಲ್ಲಿವೆ.

ಆಡಿಟ್ ಯೋಜನೆ ಮತ್ತು ತಯಾರಿ (Audit planning & preparation)

ಡಾಟಾ ಸೆಂಟರ್ ಪರಿಶೀಲನೆಯನ್ನು ನಡೆಸುವ ಮೊದಲು ಕಂಪನಿಯ ಬಗ್ಗೆ ಮತ್ತು ಅದರ ನಿರ್ಣಾಯಕ ವ್ಯಾಪಾರ ಚಟುವಟಿಕೆಗಳ ಬಗ್ಗೆ ಆಡಿಟರ್‌ಗೆ ಸಮರ್ಪಕವಾಗಿ ಶಿಕ್ಷಣ ನೀಡಬೇಕು. ಪರಿಶೀಲನೆಯನ್ನು ನಡೆಸುವ ಮೊದಲು ಆಡಿಟರ್ ಕೆಳಗಿನವುಗಳನ್ನು ನಿರ್ವಹಿಸಬೇಕು :

- ಸಂಭವನೀಯ ಕಾಳಜಿಯ ಪ್ರದೇಶಗಳನ್ನು ನಿರ್ಧರಿಸಲು ಐಟಿ ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್‌ನ್ನು ಭೇಟಿ ಮಾಡಿ.
- ಪ್ರಸ್ತುತ ಐಟಿ ಆರ್ಗನೈಸೇಷನ್ ಚಾರ್ಟ್‌ನ್ನು ಪರಿಶೀಲಿಸಿ
- ಎಲ್ಲಾ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು, ಸಾಫ್ಟ್‌ವೇರ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಡಾಟಾ ಸೆಂಟರ್ ಉಪಕರಣವನ್ನು ಸಂಶೋಧಿಸಿ.
- ಕಂಪನಿಯ ಐಟಿ ಪಾಲಿಸಿಗಳನ್ನು ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಪರಿಶೀಲಿಸಿ.
- ಕಂಪನಿಯ ಐಟಿ ಬಜೆಟ್ ಮತ್ತು ಸಿಸ್ಟಮ್‌ನ ಯೋಜನೆಯ ದಸ್ತಾವೇಜನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡಿ.
- ಡಾಟಾ ಸೆಂಟರ್‌ನ ವಿಪತ್ತು ರಿಕವರಿ ಯೋಜನೆಯನ್ನು ಪರಿಶೀಲಿಸಿ.

ಆಡಿಟ್ ನಡೆಸುವುದು (Performing an Audit)

ಯಾವುದೇ ಪ್ರಮಾಣಿತ ಭದ್ರತಾ-ಆಡಿಟ್ ಇರುವುದಿಲ್ಲ, ಆದರೆ ಆಡಿಟರ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ವೈಯಕ್ತಿಕ ಸಂದರ್ಶನಗಳ ಮೂಲಕ, ದುರ್ಬಲತೆಯ ಸ್ಕ್ಯಾನ್‌ಗಳು, OS ನ ಪರೀಕ್ಷೆ ಮತ್ತು ಭದ್ರತಾ ಅಪ್ಲಿಕೇಷನ್ ಸೆಟ್ಟಿಂಗ್‌ಗಳು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ವಿಶ್ಲೇಷಣೆ ಜೊತೆಗೆ ಇವೆಂಟ್ ಲಾಗ್‌ಗಳಂತಹ ಐತಿಹಾಸಿಕ ಡಾಟಾವನ್ನು ಅಧ್ಯಯನ ಮಾಡುವ ಮೂಲಕ ತಮ್ಮ ಕೆಲಸವನ್ನು ಸಾಧಿಸುತ್ತಾರೆ. ಆಡಿಟರ್‌ಗಳು ವ್ಯಾಪಾರದ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳ ಮೇಲೆ ಸಹ ಕೇಂದ್ರೀಕರಿಸುತ್ತಾರೆ. ಈ ಪಾಲಿಸಿಗಳು ಏನನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ, ಹೇಗೆ ಬಳಸುತ್ತಾರೆ ಮತ್ತು ಈಗ ನಡೆಯುತ್ತಿರುವ ಮತ್ತು ಭವಿಷ್ಯದ ಬೆದರಿಕೆಗಳಿಗೆ ಪರಿಣಾಮಕಾರಿಯಾಗಿವೆಯೆ ಎಂದು ನಿರ್ಧರಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಾಮಾನ್ಯವಾಗಿ, ಕಂಪ್ಯೂಟರ್ ಭದ್ರತಾ ಆಡಿಟ್‌ಗಳನ್ನು ಇವರಿಂದ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ :

1 ಫೆಡರಲ್ ಅಥವಾ ರಾಜ್ಯ ನಿಯಂತ್ರಕರು

2 ಕಾರ್ಪೋರೇಟ್ ಆಂತರಿಕ ಆಡಿಟರ್‌ಗಳು

3 ಬಾಹ್ಯ ಆಡಿಟರ್‌ಗಳು - ತಂತ್ರಜ್ಞಾನ ಆಡಿಟಿಂಗ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಪರಿಣಿತಿ ಹೊಂದಿದವರು

4 ಸಲಹೆಗಾರರು - ವಿಶೇಷ ಕೌಶಲ್ಯವನ್ನು ಹೊಂದಿರದ ಸಂಸ್ಥೆಯಲ್ಲಿ ತಂತ್ರಜ್ಞಾನ ಆಡಿಟಿಂಗ್‌ನ್ನು ಹೊರಗುತ್ತಿಗೆಯಿಂದ ನಿರ್ವಹಿಸುವುದು.

ಮೊದಲು ಆಡಿಟ್‌ನ ವ್ಯಾಪ್ತಿಯನ್ನು ನಿರ್ಧರಿಸಬೇಕು ಮತ್ತು ಮಾಹಿತಿ ಭದ್ರತೆಗೆ ಸಂಬಂಧಿಸಿದ ಕಂಪನಿಯ ಎಲ್ಲ ಆಸ್ತಿಯನ್ನು ಅಂದರೆ ಕಂಪ್ಯೂಟರ್ ಉಪಕರಣಗಳು, ಫೋನ್‌ಗಳು, ನೆಟ್‌ವರ್ಕ್, ಇ ಮೇಲ್, ಡಾಟಾ ಮತ್ತು ಯಾವುದೇ ಪ್ರವೇಶ - ಸಂಬಂಧಿತ ಉದಾಹರಣೆಗೆ ಕಾರ್ಡ್‌ಗಳು, ಟೋಕನ್‌ಗಳು, ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ಸೇರಿಸಿಕೊಳ್ಳಬೇಕು. ನಂತರ ಹಿಂದಿನ ಮತ್ತು ಸಂಭಾವ್ಯ ಭವಿಷ್ಯದ ಆಸ್ತಿ ಬೆದರಿಕೆಗಳನ್ನು ಪರಿಶೀಲಿಸಬೇಕು. ಮಾಹಿತಿ ಭದ್ರತಾ ಕ್ಷೇತ್ರದಲ್ಲಿರುವ ಯಾರಾದರೂ ಹೊಸ ಟ್ರೆಂಡ್‌ಗಳ ಬಗ್ಗೆ ಜೊತೆಗೆ ಇತರ ಕಂಪನಿಗಳು ತೆಗೆದುಕೊಂಡ ಕ್ರಮಗಳ ಬಗ್ಗೆಯೂ ಗಮನಹರಿಸಬೇಕು. ಮುಂದೆ ಆಡಿಟಿಂಗ್ ತಂಡವು ಅಪಾಯಕಾರಿ ಪರಿಸ್ಥಿತಿಗಳಲ್ಲಿ ವರ್ಗಾಯಿಸುವ ವಿನಾಶದ ಪ್ರಮಾಣವನ್ನು ಅಂದಾಜು ಮಾಡಬೇಕು. ಬೆದರಿಕೆಯು ಸಂಭವಿಸಿದ ನಂತರ ವ್ಯಾಪಾರದ ಕಾರ್ಯಾಚರಣೆಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಸ್ಥಾಪಿತ ಯೋಜನೆ ಮತ್ತು ನಿಯಂತ್ರಣಗಳು ಇರಲೇಬೇಕು. ಇದನ್ನು ಒಳನುಗ್ಗುವಿಕೆಯನ್ನು ತಡೆಗಟ್ಟುವ ವ್ಯವಸ್ಥೆ (intrusion prevention system) ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ವಿಮರ್ಶೆಯನ್ನು ನಿರ್ವಹಿಸುವುದು (Performing the review)

ಮುಂದಿನ ಹಂತವು ಡಾಟಾ ಸೆಂಟರ್ ಆಡಿಟ್ ಉದ್ದೇಶಗಳನ್ನು ಪೂರೈಸಲು ಪುರಾವೆಗಳನ್ನು ಸಂಗ್ರಹಿಸುವುದು. ಇದು ಡಾಟಾ ಸೆಂಟರ್ ಸ್ಥಳಕ್ಕೆ ಪ್ರಯಾಣಿಸುವುದನ್ನು ಮತ್ತು ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿ ನಿರ್ವಹಿಸಲಾದ ಪ್ರಕ್ರಿಯೆಗಳು ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಗಮನಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಪೂರ್ವನಿರ್ಧರಿತ ಆಡಿಟ್‌ನ ಉದ್ದೇಶಗಳನ್ನು ಪೂರೈಸಲು ಕೆಳಗಿನ ವಿಮರ್ಶೆ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ನಡೆಸಬೇಕು :

- ಆಡಿಟರ್‌ಗಳು ತಮ್ಮ ಉದ್ದೇಶಗಳನ್ನು ಪೂರೈಸಲು, ಡಾಟಾ ಸೆಂಟರ್ ಉದ್ಯೋಗಿಗಳನ್ನು ಗಮನಿಸಬೇಕು ಮತ್ತು ಸಂದರ್ಶನವನ್ನು ಮಾಡಬೇಕು.
- ಆಡಿಟರ್‌ಗಳು, ಡಾಟಾ ಸೆಂಟರ್‌ನ ಎಲ್ಲ ಉಪಕರಣಗಳು ಸರಿಯಾಗಿ ಮತ್ತು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿವೆ ಎಂದು ಪರಿಶೀಲಿಸಬೇಕು.
- ಡಾಟಾ ಸೆಂಟರ್‌ನ ಎಲ್ಲ ಪಾಲಿಸಿಗಳನ್ನು ಮತ್ತು ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ದಾಖಲಿಸಬೇಕು ಮತ್ತು ಅವುಗಳನ್ನು ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿ ಇರಿಸಬೇಕು. ಮುಖ್ಯ ದಾಖಲಿತ ಕಾರ್ಯವಿಧಾನಗಳು ಯಾವವೆಂದರೆ : ಡಾಟಾ ಸೆಂಟರ್ ಸಿಬ್ಬಂದಿಗಳ ಉದ್ಯೋಗದ ಜವಾಬ್ದಾರಿಗಳು, ಬ್ಯಾಕಪ್ ಪಾಲಿಸಿಗಳು, ಭದ್ರತಾ ಪಾಲಿಸಿಗಳು, ಉದ್ಯೋಗಿಗಳನ್ನು ವಜಾಗೊಳಿಸುವ ಪಾಲಿಸಿಗಳು, ಸಿಸ್ಟಮ್ ಆಪರೇಟಿಂಗ್ ಕಾರ್ಯವಿಧಾನಗಳು ಮತ್ತು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳ ಅವಲೋಕನ.
- ಆಡಿಟರ್‌ಗಳು ಕ್ಲೈಂಟ್‌ಗಳ ಡಾಟಾ ಸೆಂಟರ್‌ನ ಭದ್ರತೆಯನ್ನು ನಿರ್ಣಯಿಸಬೇಕು. ಭೌತಿಕ ಭದ್ರತಾ ನಿಯಂತ್ರಣಗಳಿಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಮತ್ತು ಡಾಟಾ ಸೆಂಟರ್‌ನ ಉಪಕರಣದ ಭದ್ರತೆಯನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಪರಿಸರ ನಿಯಂತ್ರಣಗಳು ಸ್ವಸ್ಥಾನದಲ್ಲಿರಬೇಕು. ಇವು ಹವಾನಿಯಂತ್ರಿತ ಘಟಕಗಳು, ಎತ್ತರಿಸಿದ ಮಹಡಿಗಳು, ಆರ್ದ್ರಕಗಳು, ಮತ್ತು ತಡೆಯಿಲ್ಲದ ವಿದ್ಯುತ್ ಪೂರೈಕೆ ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.
- ಬ್ಯಾಕಪ್ ಕಾರ್ಯವಿಧಾನಗಳು - ಆಡಿಟರ್‌ಗಳು ಸಿಸ್ಟಮ್‌ನ ವೈಫಲ್ಯದ ಸಂದರ್ಭದಲ್ಲಿ ಕ್ಲೈಂಟ್ ಬ್ಯಾಕಪ್ ಕಾರ್ಯವಿಧಾನಗಳನ್ನು ಹೊಂದಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸಬೇಕು. ಕ್ಲೈಂಟ್‌ಗಳು ಸಿಸ್ಟಮ್ ವೈಫಲ್ಯದ ಸಂದರ್ಭದಲ್ಲಿ ತಕ್ಷಣವೇ ಕಾರ್ಯಗಳನ್ನು ಮುಂದುವರೆಸಲು ಅನುಮತಿಸುವಂತೆ ಪ್ರತ್ಯೇಕ ಡಾಟಾ ಸೆಂಟರ್‌ನ್ನು ನಿರ್ವಹಿಸಬಹುದು.

ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷೆಗಳು, ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನಗಳು, ಮತ್ತು ಭದ್ರತಾ ಆಡಿಟ್‌ಗಳು (Penetration Tests, Vulnerability Assessments And Security Audits)

ಬಾಹ್ಯ ಆಕ್ರಮಣಕಾರರು ಸಾಮಾನ್ಯವಾಗಿ ಈಗಾಗಲೇ ತಿಳಿದಿರುವ ದುರ್ಬಲತೆಗಳನ್ನು ಉಪಯೋಗಿಸುತ್ತಾರೆ ಮತ್ತು ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳಲ್ಲಿ ಒಳನುಸುಳುತ್ತಾರೆ ಮತ್ತು ದುರ್ಬಳಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ. ಸೂಕ್ತ ರಕ್ಷಣಾತ್ಮಕ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುವುದು ಮತ್ತು ಸಾಕಷ್ಟು ಭದ್ರತಾ ವಿನ್ಯಾಸಗಳು ಈ ಸಮಸ್ಯೆಯನ್ನು ತಗ್ಗಿಸಬಹುದು. ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಬಹಿರಂಗ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಮತ್ತು ನಿಯಮಿತವಾಗಿ ಅವುಗಳ ಅಪಾಯಗಳನ್ನು ಗುರುತಿಸುವ ಮೂಲಕ ಉತ್ತಮ ಮಟ್ಟದಲ್ಲಿ ಇದನ್ನು ಸಾಧಿಸಬಹುದು. ಅಪಾಯಗಳನ್ನು ಆರಂಭಿಕ ಹಂತದಲ್ಲಿ ಕಂಡುಹಿಡಿಯಬಹುದು. ಮತ್ತು ಮೊದಲ ಹಂತದಲ್ಲಿ ಸೂಕ್ತ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು.

ದುರ್ಬಲತೆಯ ಮೌಲ್ಯಮಾಪನವು ಎಲ್ಲ ಸಂಭಾವ್ಯ ದುರ್ಬಲತೆಗಳನ್ನು ಗುರುತಿಸಲು ಬಳಸುವ ಅಭ್ಯಾಸವಾಗಿದೆ. ಮೌಲ್ಯಮಾಪನವನ್ನು ಭೌತಿಕ ಭದ್ರತೆ, ಸಿಬ್ಬಂದಿ, ಅಥವಾ ವ್ಯವಸ್ಥೆ, ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಭದ್ರತೆ ಮುಂತಾದವುಗಳನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ದುರ್ಬಲತೆಯನ್ನು ಗುರುತಿಸುವ ಸಾಧನಗಳನ್ನು ಅವುಗಳನ್ನು ಗುರುತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಐಪಿ ಅಡ್ರೆಸ್‌ನ ಮೂಲಕ ಪ್ರತಿಯೊಂದು ಕಂಪ್ಯೂಟರ್ ಗುರುತಿಸುವುದು ಮತ್ತು ಅವುಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ದುರ್ಬಲತೆಯನ್ನು ಮತ್ತು ಆ ದುರ್ಬಲತೆಗಳನ್ನು ಹೇಗೆ ಸರಿಪಡಿಸಬಹುದು ಎಂಬುದರ ಪಟ್ಟಿಯನ್ನು ನಂತರ ರಚಿಸಬಹುದು.

ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷೆಯನ್ನು ತಿಳಿದಿರುವ ದುರ್ಬಲತೆಗಳನ್ನು ಬಳಸಿ "breaking into the network" ನ ಉದ್ದೇಶದಿಂದ ನಡೆಸಲಾಗುತ್ತದೆ. ಇಲ್ಲಿಂದ ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿ ಅತ್ಯಂತ ನಿರ್ಣಾಯಕ ವ್ಯವಸ್ಥೆಗೆ, ನಿರ್ವಾಹಕರು ಅಥವಾ ರೂಟ್ ಪ್ರವೇಶವನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು ಗುರಿಯಾಗಿರುತ್ತದೆ. ಇದು ನೆಟ್‌ವರ್ಕ್‌ಗೆ ಸಂಪೂರ್ಣ ಪ್ರವೇಶವನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಸಿಸ್ಟಮ್‌ನ್ನು ಮತ್ತು ಸಿಸ್ಟಮ್‌ನಲ್ಲಿರುವ ಡಾಟಾವನ್ನು ಮಾರ್ಪಡಿಸುತ್ತದೆ. ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷೆಯು ಒಬ್ಬ ನಿಜವಾದ ಹ್ಯಾಕರ್ ಏನು ಮಾಡಬಹುದು ಎಂಬುದನ್ನು ಅನುಕರಿಸಲು ನಡೆಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಇದು ಕಂಪನಿಗೆ ಒಂದು ಸಂಸ್ಥೆಯನ್ನು ವಾಸ್ತವವಾಗಿ ಭೇದಿಸಬಹುದು ಎಂದು ಸಾಬೀತುಪಡಿಸುತ್ತದೆ.

ಪೆನಿಟ್ರೇಷನ್ ಪರೀಕ್ಷೆಯನ್ನು ನೈತಿಕ ಹ್ಯಾಕಿಂಗ್ ಎಂತಲೂ ಕರೆಯಲಾಗುತ್ತದೆ. ಹೆಚ್ಚಿನ ಸಂದರ್ಭಗಳಲ್ಲಿ, ಭದ್ರತಾ ವೃತ್ತಿಪರರು ದುರ್ಬಲತೆಯ ಸ್ಕ್ಯಾನರ್‌ನಿಂದ ವರದಿಗಳನ್ನು ನೋಡಬಹುದು ಮತ್ತು ಕಂಪನಿಯು ಎದುರಿಸುತ್ತಿರುವ ಅಪಾಯದ ಮಟ್ಟವನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

ಆಡಿಟ್ ನಿಯಂತ್ರಣಗಳು (Audit Controls)

ಅಪ್ಲಿಕೇಷನ್ ನಿಯಂತ್ರಣಗಳ ವಿಮರ್ಶೆ - ನಿಯೋಜಿಸಲಾದ ತಂತ್ರಜ್ಞಾನಗಳ ಅಪಾಯಗಳ ಗುರುತಿಸುವಿಕೆ ಆಗಿದೆ ಮತ್ತು ಅವಶ್ಯಕ ನಿಯಂತ್ರಣಗಳನ್ನು ಮತ್ತು ಭದ್ರತೆಯನ್ನು ಇರಿಸುವುದನ್ನು ಖಚಿತಪಡಿಸುವುದರ ಮೂಲಕ ಇಂತಹ ಅಪಾಯಗಳಿಗೆ ಕಂಪನಿಯ ಒಡ್ಡಿಕೊಳ್ಳುವಿಕೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

ಸಾಮಾನ್ಯ ಕಂಪ್ಯೂಟರ್ ನಿಯಂತ್ರಣಗಳ ವಿಮರ್ಶೆ - ಕಂಪನಿಯಲ್ಲಿ ವಿವಿಧ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಅಪ್ಲಿಕೇಷನ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗಾಗಿ ಸುರಕ್ಷಿತ ಮತ್ತು ಸ್ಥಿರ ವಾತಾವರಣವನ್ನು ಒದಗಿಸುವುದಕ್ಕಾಗಿ ಇದನ್ನು ಮಾಡಲಾಗುತ್ತದೆ.

ಕಾರ್ಯಾಚರಣೆಯ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ದೋಷಗಳು ಕಂಡುಬಂದರೆ ಸಿಸ್ಟಮ್ ಕಳೆದುಹೋಗಬಹುದು.

ನಿಯಂತ್ರಣಗಳ ಉದ್ದೇಶಗಳು :

- ಕಂಪ್ಯೂಟರ್‌ಗೆ ಪ್ರವೇಶಿಸುವ ಡಾಟಾ ಸರಿಯಾಗಿದೆಯೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು
- ಕಂಪ್ಯೂಟರ್‌ಗೆ ಇನ್ಪುಟ್ ಮಾಡುವ ಮೊದಲು ಡಾಟಾದ ಕ್ಲೆರಿಕಲ್ ನಿರ್ವಹಣೆಯನ್ನು ಪರಿಶೀಲಿಸಿ
- ಕೆಟ್ಟದಾದ ಡಾಟಾ ಅಥವಾ ಕೆಟ್ಟದಾದ ಪ್ರೊಗ್ರಾಂನ ಕಾರಣದಿಂದಾಗಿ ಉಂಟಾಗುವ ದೋಷಗಳನ್ನು ಕಂಡುಹಿಡಿಯಲು ಮತ್ತು ನಿರೂಪಿಸಲು ವಿಧಾನಗಳನ್ನು ಒದಗಿಸಿ.
- ಕಾನೂನು ಅವಶ್ಯಕತೆಗಳನ್ನು ಪೂರೈಸಲಾಗಿದೆಯೆ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.
- ವಂಚನೆಗಳ ವಿರುದ್ಧ ರಕ್ಷಿಸಲು

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಮತ್ತು ಆಡಿಟಿಂಗ್ (Access Control and Auditing)

ಭೌತಿಕ ಮತ್ತು ತಾರ್ಕಿಕ ಭದ್ರತೆ (Physical and logical security)

ಹಲವಾರು ವಿಧದ ಭದ್ರತೆಗಳಿವೆ, ಭೌತಿಕ ಮತ್ತು ತಾರ್ಕಿಕ. ಭೌತಿಕ ಭದ್ರತೆಯು, ಲಾಕ್‌ಗಳು ಮತ್ತು ಬಯೊಮೆಟ್ರಿಕ್‌ನಂತಹ ವಿಷಯಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ತಾರ್ಕಿಕ ಭದ್ರತೆಯು ಸಾಫ್ಟ್‌ವೇರ್ ಸುರಕ್ಷತೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಮತ್ತು ಆಡಿಟಿಂಗ್, ಬಳಕೆದಾರರ ಖಾತೆಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು, ಉಲ್ಲಂಘನೆ ಮತ್ತು ಭದ್ರತಾ ಚಟುವಟಿಕೆಗಳ ವರದಿಗಳು ಮತ್ತು ಫೈರ್‌ವಾಲ್‌ಗಳು ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣವು, ಕಂಪ್ಯೂಟರ್ ಆಧಾರಿತ ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ, ಪ್ರದೇಶಗಳಿಗೆ ಮತ್ತು ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಪ್ರವೇಶವನ್ನು ನಿಯಂತ್ರಿಸಲು ಅಧಿಕಾರಿಗಳಿಗೆ ಅನುವು ಮಾಡಿಕೊಡುವ ವ್ಯವಸ್ಥೆಯಾಗಿದೆ. ಒಂದು ಕಂಪ್ಯೂಟರ್‌ನ ಪಾಸ್‌ವರ್ಡ್ ಅಥವಾ ATM ಸಿಸ್ಟಮ್‌ನ PIN ಇವು ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ರೂಪಗಳಾಗಿವೆ. ವ್ಯಕ್ತಿಗಳು ಗೌಪ್ಯ, ಮುಖ್ಯ ಅಥವಾ ಸೂಕ್ಷ್ಮ ಮಾಹಿತಿ ಮತ್ತು ಉಪಕರಣಗಳನ್ನು ಸುರಕ್ಷಿತಗೊಳಿಸಲು ಪ್ರಯತ್ನಿಸುವಾಗ, ಪ್ರವೇಶ ನಿಯಂತ್ರಣದ ಕಾರ್ಯವಿಧಾನವನ್ನು ಉಪಯೋಗಿಸುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ.

ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಸಿಸ್ಟಮ್‌ನ್ನು ಆಡಿಟ್ ಮಾಡುವುದು ಒಂದು ಸಿಸ್ಟಮ್‌ನಲ್ಲಿ ಪ್ರವೇಶಿಸುವುದನ್ನು ಪತ್ತೆ ಮಾಡುವುದು ಅಥವಾ ಅದರ ಪ್ರಯತ್ನವನ್ನು ಪತ್ತೆ ಮಾಡುವ ಒಂದು ಮಾರ್ಗವಾಗಿದೆ. ಇದು ಮುಖ್ಯವಾಗಿದೆ. ಏಕೆಂದರೆ ಇದು ಪ್ರವೇಶ ನಿಯಂತ್ರಣ ಸಿಸ್ಟಮ್ ಎಷ್ಟು ಯಶಸ್ವಿಯಾಗಿದೆ ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತದೆ. ಹಾಗೆಯೆ ಯಾರಿಗೆ ಪ್ರವೇಶವನ್ನು ನಿರಾಕರಿಸಲಾಯಿತು ಎಂಬುದನ್ನು ಮತ್ತು ಅವರು ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಬಾರಿ ಪ್ರವೇಶವನ್ನು ಪ್ರಯತ್ನಿಸಿದರೆ ಅವರ ಉದ್ದೇಶವೇನು ? ಎಂಬುದನ್ನು ತೋರಿಸುತ್ತದೆ.

ಕೆಳಗಿನವುಗಳನ್ನು ಪರಿಶೀಲಿಸಲು ತಾರ್ಕಿಕ ಆಡಿಟ್‌ನ್ನು ಮಾಡಲಾಗುತ್ತದೆ. :

- ಪ್ರವೇಶ ಉಲ್ಲಂಘನೆಗಳ ಸಾಮರ್ಥ್ಯಗಳು ಮತ್ತು ಉಲ್ಲಂಘನೆಗಳು

© NIMI NOT TO BE REPUBLISHED

- ಭದ್ರತಾ ಚಟುವಟಿಕೆ ವರದಿಗಳು
- ಚಟುವಟಿಕೆ ವರದಿಗಳನ್ನು ಲಾಗ್ ಮಾಡುವುದು
- ಫೈರ್‌ವಾಲ್‌ಗಳ ದಕ್ಷತೆ
- ಭದ್ರತಾ ಉಲ್ಲಂಘನೆಗಳ ಕುರಿತು ವರದಿಗಳು
- ಅನಧಿಕೃತ ವ್ಯಕ್ತಿಗಳು ಸಿಸ್ಟಮ್‌ನ್ನು ಹ್ಯಾಕ್ ಮಾಡಲು ಪ್ರಯತ್ನಿಸುವ ವರದಿಗಳು

ಆಡಿಟರ್‌ಗಳಿಗೆ ವೃತ್ತಿಪರ ನೀತಿಶಾಸ್ತ್ರ (Professional Ethics for Auditors) : ವಸ್ತುನಿಷ್ಟ ಆಡಿಟ್‌ನಲ್ಲಿ ನಂಬಿಕೆಯನ್ನು ಪಡೆದುಕೊಳ್ಳಲು, ವೃತ್ತಿಪರ ನೀತಿಶಾಸ್ತ್ರದ ಸಮೂಹವನ್ನು ಎತ್ತಿ ಹಿಡಿಯುವುದು ಅವಶ್ಯವಾಗಿರುತ್ತದೆ. ವೃತ್ತಿಪರ ನೈತಿಕತೆಯನ್ನು ವೈಯಕ್ತಿಕವಾಗಿ ವ್ಯಕ್ತಿಗಳು ಮತ್ತು ಮಾಹಿತಿ ಭದ್ರತಾ ಆಡಿಟಿಂಗ್ ಕ್ಷೇತ್ರದಲ್ಲಿ ಸೇವೆಗಳನ್ನು ಒದಗಿಸುವ ಕಂಪನಿಗಳು ಎತ್ತಿಹಿಡಿಯಬೇಕು. ವೃತ್ತಿಪರ ನೈತಿಕತೆಯು ಕೆಳಗಿನ ತತ್ವಗಳನ್ನು ಒಳಗೊಂಡಿದೆ :

ಪ್ರಾಮಾಣಿಕತೆ ಮತ್ತು ಗೌಪ್ಯತೆ (Honesty and confidentiality): ಪ್ರಾಮಾಣಿಕತೆಯು ನಂಬಿಕೆಯ ಆಡಿಪಾಯವಾಗಿದೆ ಮತ್ತು ಮೌಲ್ಯಮಾಪನದ ವಿಶ್ವಾಸಾರ್ಹತೆಯ ಆಧಾರವನ್ನು ರೂಪಿಸುತ್ತದೆ. ಸೂಕ್ಷ್ಮ ವ್ಯವಹಾರ ಪ್ರಕ್ರಿಯೆಗಳು ಮತ್ತು ಮಾಹಿತಿ ಇವು ಮಾಹಿತಿ ಭದ್ರತೆಯ ಮೇಲೆ ಅವಲಂಬಿತವಾಗಿರುತ್ತವೆ. ಮಾಹಿತಿಯ ಗೌಪ್ಯತೆಯನ್ನು ಆಡಿಟ್‌ನ ಸಮಯದಲ್ಲಿ ಪಡೆದುಕೊಳ್ಳಲಾಗುತ್ತದೆ. ಫಲಿತಾಂಶಗಳ ವಿವೇಚನಾಯುಕ್ತ ನಿರ್ವಹಣೆಯು ಮತ್ತು IS ಆಡಿಟ್‌ನ ಸಂಶೋಧನೆಗಳು ಇಂತಹ ಕೆಲಸಗಳಿಗೆ ಪ್ರಮುಖ ಆಧಾರವಾಗಿವೆ. IS ಆಡಿಟರ್‌ಗಳು ತಾವು ಪಡೆದುಕೊಂಡ ಮಾಹಿತಿಯ ಮೌಲ್ಯವನ್ನು ತಿಳಿದಿರುತ್ತಾರೆ ಮತ್ತು ಅವರು ಕಾನೂನುಬದ್ಧವಾದ ಅಥವಾ ವೃತ್ತಿಪರವಾಗಿ ಅನಗತ್ಯವಾಗಿ ಅನುಮತಿಯಿಲ್ಲದೆ ಈ ಮಾಹಿತಿಯನ್ನು ಬಹಿರಂಗಪಡಿಸುವುದಿಲ್ಲ.

ತಜ್ಞರ ಜ್ಞಾನ (Expert knowledge): IS ಆಡಿಟರ್‌ಗಳು ತಾವು ಹೊಂದಿರುವ ಅವಶ್ಯಕ ಜ್ಞಾನ ಮತ್ತು ಕೌಶಲ್ಯಗಳಿಗೆ ಮತ್ತು ಅನುಭವಗಳಿಗೆ ಅನುಗುಣವಾದ ಕೆಲಸಗಳನ್ನು ಮಾತ್ರ ಸ್ವೀಕರಿಸುತ್ತಾರೆ ಮತ್ತು ಅವರ ಕಾರ್ಯವನ್ನು ಮಾಡುವಾಗ ಇವುಗಳನ್ನು ಬಳಸುತ್ತಾರೆ. ಇವರು ತಮ್ಮ ಕೆಲಸದ ಬಗ್ಗೆ ಜ್ಞಾನ , ಪರಿಣಾಮಕಾರಿತ್ವ ಮತ್ತು ಗುಣಮಟ್ಟವನ್ನು ನಿರಂತರವಾಗಿ ಸುಧಾರಿಸುತ್ತಿರುತ್ತಾರೆ.

ವಸ್ತುನಿಷ್ಠತೆ ಮತ್ತು ಸಂಪೂರ್ಣತೆ (Objectivity and thoroughness) : IS ಆಡಿಟರ್. ಆಡಿಟ್ ಮಾಡಲಾದ ಚಟುವಟಿಕೆಗಳನ್ನು ಅಥವಾ ವ್ಯಾಪಾರ ಪ್ರಕ್ರಿಯೆಗಳ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವಾಗ, ಮೌಲ್ಯಮಾಪನ ಮಾಡುವಾಗ ಉನ್ನತ ಮಟ್ಟದ ವಸ್ತುನಿಷ್ಠತೆಯನ್ನು ಮತ್ತು ಸಂಪೂರ್ಣತೆಯನ್ನು ಕಡ್ಡಾಯವಾಗಿ ಪ್ರದರ್ಶಿಸಬೇಕು. ಎಲ್ಲ ಸಂಬಂಧಿತ ಸಂದರ್ಭಗಳ ಮೌಲ್ಯಮಾಪನವನ್ನು ನಿಷ್ಪಕ್ಷಪಾತವಾಗಿ ಮಾಡಬೇಕು ಮತ್ತು ಆಡಿಟರ್‌ನ ಸ್ವಂತ ಆಸಕ್ತಿಗಳು ಅಥವಾ ಇತರರ ಹಿತಾಸಕ್ತಿಗಳು ಪ್ರಭಾವ ಬೀರಬಾರದು.

ಉದ್ದೇಶದ ಪ್ರಸ್ತುತಿ (Objective presentation): IS ಆಡಿಟರ್ ತನ್ನ ಕ್ಲೈಂಟ್‌ಗೆ ಪರೀಕ್ಷೆಯ ಫಲಿತಾಂಶಗಳನ್ನು ನಿಖರವಾಗಿ ಮತ್ತು ಸತ್ಯವಾಗಿ ವರದಿ ಮಾಡುವ ಕರ್ತವ್ಯವನ್ನು ಹೊಂದಿರುತ್ತಾನೆ. ಇದು IS ಆಡಿಟ್ ವರದಿಯಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು ನಿಷ್ಪಕ್ಷಪಾತವಾಗಿ ಮತು ಅರ್ಥವಾಗುವಂತಹ ಪ್ರಸ್ತುತಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ನಿರ್ಧರಿಸಿದ ಅಂಶಗಳ ರಚನಾತ್ಮಕ ಮೌಲ್ಯಮಾಪನವನ್ನು ಮತ್ತು ಸುರಕ್ಷತೆಗಳನ್ನು ಮತ್ತು ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಸುಧಾರಿಸಲು ನಿರ್ದಿಷ್ಟವಾದ ಶಿಫಾರಸುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಪರಿಶೀಲನೆಗಳು ಮತ್ತು ಪುನರುತ್ಪಾದನೆ (Verifications and reproducibility) : ನಿಜವಾದ ಸಂಗತಿಗಳ, ಸ್ಪಷ್ಟ ಮತ್ತು ಸ್ಥಿರವಾದ ದಾಖಲಾತಿಗಳು ವಿಶ್ವಾಸಾರ್ಹ ಮತ್ತು ಗ್ರಹಿಸಬಹುದಾದ ತೀರ್ಮಾನಗಳಿಗೆ ಮತ್ತು ಫಲಿತಾಂಶಗಳಿಗೆ ಆಧಾರವಾಗಿವೆ. ಇದು IS ಆಡಿಟ್ ತಂಡವು ತೀರ್ಮಾನಕ್ಕೆ ಬರಲು, ದಾಖಲಿತ ಮತ್ತು ಪುನರುತ್ಪಾದಿಸುವ ವಿಧಾನವನ್ನು ಅನುಸರಿಸುತ್ತದೆ ಎಂಬುದನ್ನೂ ಸಹ ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಅನುಸರಣೆ ಆಡಿಟ್ (Compliance audit): ನಿಯಂತ್ರಕ ಮಾರ್ಗಸೂಚಿಗಳಿಗೆ ಒಂದು ಸಂಸ್ಥೆಯ ನಿಷ್ಠೆಯನ್ನು ಸಮಗ್ರವಾಗಿ ವಿಮರ್ಶೆ ಮಾಡುವುದನ್ನು ಅನುಸರಣೆ ಆಡಿಟ್ ಎನ್ನುತ್ತಾರೆ. ಸ್ವತಂತ್ರ ಲೆಕ್ಕಪತ್ರ ನಿರ್ವಹಣೆ, ಭದ್ರತೆ ಅಥವಾ IT ಸಲಹೆಗಾರರು ಅನುಸರಣೆ ಸಿದ್ಧತೆಗಳ ಶಕ್ತಿಯನ್ನು ಮತ್ತು ಸಂಪೂರ್ಣತೆಯನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡುತ್ತಾರೆ. ಆಡಿಟರ್‌ಗಳು ಅನುಸರಣಾ ಆಡಿಟ್‌ನ ಅವಧಿಯಲ್ಲಿ, ಭದ್ರತಾ ಪಾಲಿಸಿಗಳನ್ನು , ಬಳಕೆದಾರರ ಪ್ರವೇಶ ನಿಯಂತ್ರಣವನ್ನು ಮತ್ತು ಅಪಾಯ ನಿರ್ವಹಣೆಯ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಪರಿಶೀಲಿಸುತ್ತಾರೆ.

ಅನುಸರಣಾ ಆಡಿಟ್‌ನಲ್ಲಿ ಏನನ್ನು ಪರಿಶೀಲಿಸಲಾಗುತ್ತದೆ ಎಂಬುದು ಒಂದು ಸಂಸ್ಥೆಯು ಸಾರ್ವಜನಿಕವಾಗಿದೆಯೆ ಅಥವಾ ಖಾಸಗಿ ಕಂಪನಿಯಾಗಿದೆಯೆ ಎನ್ನುವ ಆಧಾರದ, ಅದು ಯಾವ ರೀತಿಯ ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ಅದು ಸೂಕ್ಷ್ಮ ಹಣಕಾಸು ಡಾಟಾವನ್ನು ರವಾನಿಸುತ್ತದೆಯೆ ಅಥವಾ ಸಂಗ್ರಹಿಸುತ್ತದೆಯೆ ಎನ್ನುವ ಆಧಾರದ ಮೇಲೆ ಬದಲಾಗುತ್ತದೆ.

ಮಾಹಿತಿ ಭದ್ರತಾ ನೀತಿಗಳು (Information Security Policies): ಸಂಸ್ಥೆಗಳು ಮಾಹಿತಿ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳ ಅಭಿವೃದ್ಧಿಗೆ ಹೆಚ್ಚಿನ ಆದ್ಯತೆಯನ್ನು ನೀಡುತ್ತಿವೆ. ಸಂಸ್ಥೆಯ ಆಸ್ತಿಗಳನ್ನು ರಕ್ಷಿಸುವುದು ಪರಿಗಣಿಸಬೇಕಾದ ಪ್ರಮುಖವಾದ ವಿಷಯವಾಗಿದೆ. ಮಾಹಿತಿ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳಲ್ಲಿನ ಸ್ಪಷ್ಟತೆಯ ಕೊರತೆಯು ಚೇತರಿಸಿಕೊಳ್ಳದೇ ಇರುವಂತಹ ತೀವ್ರವಾದ ಹಾನಿಗೆ ಕಾರಣವಾಗಬಹುದು. ಆದ್ದರಿಂದ ಸಂಸ್ಥೆಗಳು ವಿಭಿನ್ನ ತಂತ್ರಗಳನ್ನು ಮಾಡಿ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸುವಲ್ಲಿ ಯಶಸ್ವಿಯಾಗುತ್ತವೆ. ಮಾಹಿತಿ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳು ಸಂಸ್ಥೆಯಲ್ಲಿ ಮಾಹಿತಿ ಭದ್ರತೆಗೆ ನಿರ್ವಹಣಾ ನಿರ್ದೇಶನವನ್ನು ಮತ್ತು ಬೆಂಬಲವನ್ನು ಒದಗಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಂಸ್ಥೆಯಲ್ಲಿನ ಮಾಹಿತಿಯನ್ನು ಗೌಪ್ಯತೆ, ಸಮಗ್ರತೆ ಮತ್ತು ಲಭ್ಯತೆಯ ಉಲ್ಲಂಘನೆಗಳ ಪರಿಣಾಮಗಳ ವಿರುದ್ಧ ಸುರಕ್ಷಿತವಾಗಿರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಅನಧಿಕೃತ ಬದಲಾವಣೆಗಳು, ಅಳಿಸುವಿಕೆ ಮತ್ತು ಬಹಿರಂಗಪಡಿಸುವಿಕೆಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು ನಿಯಂತ್ರಿಸಲು ಮತ್ತು ಸುರಕ್ಷಿತವಾಗಿರಿಸಲು ಸರಿಯಾದ ಭದ್ರತಾ ಕ್ರಮಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಅನ್ವಯಿಸಬೇಕಾದ ಭದ್ರತಾ ಕ್ರಮಗಳ ಮಟ್ಟವನ್ನು ಕಂಡುಹಿಡಿಯಲು ಅಪಾಯ ಮೌಲ್ಯಮಾಪನವು ಕಡ್ಡಾಯವಾಗಿರುತ್ತದೆ.

ಭದ್ರತಾ ನೀತಿಗಳು ಸಂಸ್ಥೆಯೊಳಗಿನ ಉದ್ಯೋಗಿಗಳಿಂದ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಿಗೆ ಸಂಬಂಧಿಸಿದಂತೆ ಏನನ್ನು ನಿರೀಕ್ಷಿಸಲಾಗಿದೆ ಎಂಬುದನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಉದ್ದೇಶಿಸಲಾಗಿದೆ. ಇಲ್ಲಿ ಮಾಹಿತಿ ಸ್ವತ್ತುಗಳಿಗೆ ಇರುವ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡಲು ಸಿಸ್ಟಮ್‌ಗಳ ಬಳಕೆಯ ಮಾರ್ಗದರ್ಶನ ನೀಡುವುದು ಗುರಿಯಾಗಿದೆ. ಇದು ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ಗಳ ಜೊತೆಗೆ ವ್ಯವಹರಿಸುವ ಸಿಬ್ಬಂದಿಗೆ ಸ್ವೀಕಾರಾರ್ಹ ಪಾಲಿಸಿಯ ಬಳಕೆ ಏನನ್ನು ಅನುಮತಿಸಲಾಗಿದೆ ಮತ್ತು ಏನನ್ನು ಇಲ್ಲ ಎಂದು ವಿವರಿಸುತ್ತದೆ. ಎಲ್ಲ ಕಂಪನಿಗಳ ಭದ್ರತಾ ಪಾಲಿಸಿಗಳು ಒಂದೇ ಆಗಿರುವುದಿಲ್ಲ. ಆದರೆ, ಅವುಗಳ ಹಿಂದಿನ ಪ್ರಮುಖ ಉದ್ದೇಶವು ಆಸ್ತಿಗಳನ್ನು ರಕ್ಷಿಸುವುದು ಆಗಿದೆ. ಭದ್ರತಾ ಪಾಲಿಸಿಗಳನ್ನು ನಿರ್ದಿಷ್ಟ ಗುರಿಗಳೊಂದಿಗೆ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

ಮಾಹಿತಿ ಭದ್ರತಾ ಆಡಿಟ್ ಸಾಧನಗಳು (Information Security Audit Tools): ಮಾಹಿತಿ ಭದ್ರತಾ ಆಡಿಟ್ ಸಾಧನಗಳು ಯುಟಿಲಿಟಿಗಳನ್ನು, ವಿದ್ಯುತ್ ಉಪಕರಣಗಳನ್ನು, ಓಪನ್ ಸೋರ್ಸ್ ಮತ್ತು ವಾಣಿಜ್ಯ ಎರಡನ್ನೂ ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಯುಟಿಲಿಟಿ ಸಾಧನಗಳು (Utility Tools) : ಇವು ಏಕ ಉದ್ದೇಶದ ಸಾಧನಗಳಾಗಿವೆ. ಇವು ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಸ್ಥಳೀಯವಾಗಿರಬಹುದು ಅಥವಾ ಉಚಿತವಾಗಿ ಲಭ್ಯವಿರಬಹುದು. ಯುಟಿಲಿಟಿ ಸಾಧನಗಳಿಗೆ ಮ್ಯಾನುಯಲ್ ವಿಧಾನಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. (ಇವುಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಕಸ್ಟಮೈಸ್ ಮಾಡಿದ ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳಲ್ಲಿ ಅಥವಾ ವಾಣಿಜ್ಯ ಉತ್ಪನ್ನಗಳಲ್ಲಿ ಸಹ ಸೇರಿಸಿದರೂ ಕೂಡ.)ನೀವುಸ್ಥಳೀಯಯುಟಿಲಿಟಿಗಳನ್ನುಸಹಸೇರಿಸಿಕೊಳ್ಳಬಹುದು. ಉದಾಹರಣೆಗೆ, ಹೆಚ್ಚಿನ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಲಭ್ಯವಿರುವ ಪಿಂಗ್, ಇದನ್ನು ನೆಟ್‌ವರ್ಕ್ ಗುರಿಯು ICMP ಪ್ಯಾಕೆಟ್‌ಗಳಿಗೆ ಪ್ರತಿಕ್ರಿಯಿಸುತ್ತದೆಯೆ ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಸಾಧಕ (Pros): ಯುಟಿಲಿಟಿ ಸಾಧನಗಳು ಉಚಿತವಾಗಿ ಲಭವಿರುತ್ತವೆ ಮತ್ತು ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯಕ್ಕಾಗಿ ಕೇಂದ್ರೀಕೃತವಾಗಿರುತ್ತವೆ. ಇದು ಅವುಗಳನ್ನು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಮಾಡುತ್ತದೆ. ಇವು ದುರ್ಬಲತೆಗಳನ್ನು ಮ್ಯಾನುಯಲ್ ಆಗಿ ಕಂಡುಹಿಡಿಯುವುದಲ್ಲಿಂತ ಹೆಚ್ಚು ವೇಗವಾಗಿ ಕಂಡುಹಿಡಿಯುವಲ್ಲಿ ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ಬಾಧಕ (Cons): ಇವುಗಳನ್ನು ಬಳಸಲು ಕೌಶಲ್ಯದ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ದೊಡ್ಡ ಆಡಿಟ್‌ಗಾಗಿ, ಮ್ಯಾನುಯಲ್ ಪರೀಕ್ಷೆಯು ಹೆಚ್ಚು ಸಮಯ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅಸಮಂಜಸ ಫಲಿತಾಂಶಗಳನ್ನು ನೀಡಬಹುದು. ಇದು ಆಡಿಟರ್‌ನ ಕೌಶಲ್ಯವನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ.

Traceroute: ಇದು ಒಂದು ಹೋಸ್ಟ್‌ಗೆ ನೆಟ್‌ವರ್ಕ್‌ನ ಮಾರ್ಗವನ್ನು ನಿರ್ಧರಿಸಲು ಬಳಸಲಾಗುವ ನೆಟ್‌ವರ್ಕ್ ಟ್ರೇಸಿಂಗ್ ಉಪಯುಕ್ತತೆ ಆಗಿದೆ.

nslookup: ಡೊಮೇನ್ ಮಾಲೀಕತ್ವವನ್ನು ನಿರ್ಧರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಓಪನ್ ಸೋರ್ಸ್ ಸ್ಕ್ರಿಪ್ಟ್‌ಗಳು , ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಂತೆ :

Nmap: ಉಚಿತ ಪೋರ್ಟ್ - ಸ್ಕ್ಯಾನಿಂಗ್ ಸೌಲಭ್ಯ

Crack: ಜನಪ್ರೀಯ ಪಾಸ್‌ವರ್ಡ್ - ಭೇದಿಸುವ ಸಾಧನವಾಗಿದೆ. ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ದುರ್ಬಲವಾಗಿದ್ದರೆ ಅವುಗಳನ್ನು ಭೇದಿಸಲು ಪ್ರಯತ್ನಿಸುವ ಮೂಲಕ ನಿರ್ಧರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

John the Ripper: ಜನಪ್ರೀಯ ಪಾಸ್‌ವರ್ಡ್ - ಭೇದಿಸುವ ಸಾಧನವಾಗಿದೆ. ಇವುಗಳನ್ನು ಪ್ರಾಥಮಿಕವಾಗಿ ಯುನಿಕ್ಸ್ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಕಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

binfo.c: ಇದು BIND ಆವೃತ್ತಿಯ ಪರೀಕ್ಷಕವಾಗಿದೆ.binfo ಇದು ತ್ವರಿತ ಸಣ್ಣ ಸ್ಕ್ರಿಪ್ಟ್ ಆಗಿದ್ದು ರಿಮೋಟ್ ನೇಮ್ ಸರ್ವರ್‌ನಲ್ಲಿ ರನ್ ಆಗುತ್ತಿರುವ ಆವೃತ್ತಿಯನ್ನು ಹಿಂತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ.

ghba.c: ಇದು ಕ್ಲಾಸ್ B ಅಥವಾ C ಸಬ್‌ನೆಟ್‌ನ ಎಲ್ಲ ಮಶೀನ್ ಹೆಸರುಗಳನ್ನು ಮತ್ತು IP ಅಡ್ರೆಸ್‌ಗಳನ್ನು ಹೊರತೆಗೆಯಲು ಸೂಕ್ತವಾದ ಸಾಧನವಾಗಿದೆ.

ಪವರ್ ಸಾಧನಗಳು(Power Tools)

ಪವರ್ ಸಾಧನಗಳು ಅನೇಕ - ಕಾರ್ಯಗಳನ್ನು ಮಾಡುವ ಉಪಯುಕ್ತತೆಗಳಾಗಿವೆ. ಇವು ಆಡಿಟ್ ಪ್ರಕ್ರಿಯೆಯ ಭಾಗಗಳನ್ನು ಸುಗಮಗೊಳಿಸುತ್ತವೆ ಮತ್ತು ಸ್ವಯಂಚಾಲಿತಗೊಳಿಸುತ್ತವೆ. ಹಾಗೆಯೆ ಕೆಲವು ಓಪನ್ ಸೋರ್ಸ್ ಪ್ಯಾಕೆಜ್ ಆಗಿರುತ್ತವೆ ಮತ್ತು ಅನೇಕವು ಕಸ್ಟಮ್ ದುರ್ಬಲತೆಯ ಡಾಟಾಬೇಸ್‌ಗಳೊಂದಿಗೆ ವಾಣಿಜ್ಯ ಉತ್ಪನ್ನಗಳಾಗಿರುತ್ತವೆ.

ಸಾಧಕ (Pros) : ಸ್ವಯಂಚಾಲಿತ ಸಾಧನಗಳು ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ದುರ್ಬಲತೆಗಳಿಗಾಗಿ ಸ್ಕ್ಯಾನ್ ಮಾಡುತ್ತದೆ. ಅಲರ್ಟ್‌ಗಳನ್ನು ಹೆಲ್ಪ್ ಡೆಸ್ಕ್ ಮಾನಿಟರಿಂಗ್ ಸಾಧನಗಳಿಗೆ ಜೋಡಿಸಬಹುದು. ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ, ಸ್ಕ್ಯಾನಿಂಗ್ ಸಾಧನವನ್ನು ಫೈರ್‌ವಾಲ್ ಜೊತೆಗೆ ಅಥವಾ ಒಳನುಗ್ಗುವಿಕೆಯನ್ನು ಪತ್ತೆ ಮಾಡುವ ನಿರ್ವಹಣಾ ಕೇಂದ್ರದೊಂದಿಗೆ ಸಂಯೋಜಿಸಬಹುದು. ಕೆಲವು ವಾಣಿಜ್ಯಿಕ ಸ್ಕ್ಯಾನರ್‌ಗಳು ವಿವರಣೆಗಳ ಮತ್ತು ಸಂಬಂಧಿತ ಅಪಾಯಗಳ ಅತ್ಯುತ್ತಮ ವರದಿಗಳನ್ನು ತೋರಿಸುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಬಾಧಕ (Cons):ಸ್ಕ್ಯಾನರ್‌ಗಳು ಡಾಟಾಬೇಸ್‌ನಲ್ಲಿ ದುರ್ಬಲತೆಗಳನ್ನು ಮಾತ್ರ ಪರಿಶೀಲಿಸುತ್ತವೆ. ಇದು ಪ್ರಸ್ತುತವಾಗಿರಬೇಕು. ಅನೇಕ ಸ್ಕ್ಯಾನರ್‌ಗಳನ್ನು ಅವು ನಿರ್ವಹಿಸಲಾದ ದುರ್ಬಲತೆ ತಪಾಸಣೆಯ ಸಂಖ್ಯೆಗಳ ಮೇಲೆ ಮಾರಾಟ ಮಾಡಲಾಗುತ್ತದೆ. ಇದು ಯಾವಾಗಲೂ ಉಪಕರಣದ ಪರಿಣಾಮಕಾರಿತ್ವದ ಬಗೆಗಿನ ಉತ್ತಮ ಸೂಚನೆಯಲ್ಲ. ಆಗಾಗ್ಗೆ, ದುರ್ಬಲತೆಗಳನ್ನು ತಪ್ಪಾಗಿ ನಿರ್ಣಯಿಸಲಾಗುತ್ತದೆ. ಒಂದು ಸ್ಕ್ಯಾನರ್ ಅಪಾಯವನ್ನು ನಿಖರವಾಗಿ ನಿರ್ಣಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಕೆಲವು ಓಪನ್ ಸೋರ್ಸ್ ಪವರ್ ಸಾಧನಗಳು ಯಾವವೆಂದರೆ - Nessus, SARA(The Security Auditor's Research Assistant), Whisker, ಇತ್ಯಾದಿಗಳು.

ಇವತ್ತು ಲಭ್ಯವಿರುವ ಕೆಲವು ವಾಣಿಜ್ಯ ಸ್ಕ್ಯಾನರ್‌ಗಳು - Internet Security System's Internet Scanner, eEye Digital Security's Retina, BindView's BV-Control, CORE Security Technology's Auditing Tools Suite and Foundstone's FoundScan.

ಗೂಗಲ್ (Google)

ನಿಜವಾದ ಹ್ಯಾಕರ್ ಬಾಕ್ಸ್‌ನ ಹೊರಗೆ ಯೋಚಿಸುತ್ತಾನೆ ಮತ್ತು ಉಪಕರಣಗಳನ್ನು ಅವುಗಳನ್ನು ಉದ್ದೇಶಿಸಿರದ ರೀತಿಯಲ್ಲಿ ಬಳಸುವುದನ್ನು ಕಲಿಯುತ್ತಾನೆ. ಗೂಗಲ್ ಸರ್ಚ್ ಇಂಜಿನ್ ಆಡಿಟಿಂಗ್ ಸಾಧನ ಅಲ್ಲದಿದ್ದರೂ, ಒಂದು ಸೈಟ್ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸಲು ಇದು ಬಹಳ ಉತ್ತಮವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, "@DGET.com" ನಮೂದಿಸಲು ಪ್ರಯತ್ನಿಸುವುದು (ಇಲ್ಲಿ "DGET.com" ನಿಮ್ಮ ಡೊಮೇನ್ ಆಗಿದೆ). ಕೆಲವೊಮ್ಮೆ ಇದು ಉತ್ತಮ ಡಾಟಾವನ್ನು ನೀಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಸಿಸ್ಟಮ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್ ಅವರ ಸೈಟ್ ಬಗ್ಗೆ ತಾಂತ್ರಿಕ ವಿವರಗಳನ್ನು ಪೋಸ್ಟ್ ಮಾಡುತ್ತಾರೆ. ಇದು ಅವರ ಖಾತೆಯ ಹೆಸರನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಗೂಗಲ್ ಯುನಿಕ್ಸ್‌ನ grep ಕಮಾಂಡ್‌ನಂತೆ ಇದೆ.

ಫಲಿತಾಂಶಗಳ ಸಂವಹನ (Communicating Results)

ಪರಿಗಣಿಸಬೇಕಾದ ಅಂತಿಮ ಅಂಶವೆಂದರೆ, ಆಡಿಟೀಗಳ ಜೊತೆಗೆ ಹೇಗೆ ಸಂವಹನ ನಡೆಸುವುದು. ಅಂದರೆ, ಯಾರ ಆಸ್ತಿಗಳನ್ನು ಆಡಿಟ್ ,ಮಾಡಲಾಗುತ್ತಿದೆಯೊ ಆ ವ್ಯಕ್ತಿಗಳು. ಆಡಿಟೀಗಳಿಗೆ ನಿರಂತರ ಆಡಿಟ್ ಚಟುವಟಿಕೆಯ ಫಲಿತಾಂಶಗಳನ್ನು ತಿಳಿಸುವಾಗ, ವಿನಿಮಯವು ಸ್ವತಂತ್ರ ಮತ್ತು ಸ್ಥಿರವಾಗಿರುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ.

ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಪ್ಯಾರಾಮಿಟರ್‌ಗಳನ್ನು ಹಿರಿಯ ನಿರ್ವಾಹಕರಿಗೆ ವರದಿ ಮಾಡುವುದು (Reporting to senior management on defined parameters)

ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್‌ಗೆ ಆಡಿಟ್ ವರದಿ ಮತ್ತು ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್‌ನ ಪ್ರತಿಕ್ರಿಯೆಯು ಕೆಳಗಿನ ಟೇಬಲ್ 1 ರಲ್ಲಿ ತೋರಿಸಿದಂತೆ ಕಾಣಿಸಬಹುದು. ಆದರೆ ಇದರಲ್ಲಿ ಬಳಕೆಯಲ್ಲಿರುವ ವರದಿಗಳ ಮತ್ತು ಪ್ರತಿಕ್ರಿಯೆಗಳ ಇತರ ಹಲವಾರು ಸ್ವರೂಪಗಳೂ ಇವೆ.

ಟೇಬಲ್ 1

ಕ್ರಮ ಸಂಖ್ಯೆ	ಸಂಶೋಧನೆಗಳು	ಪರಿಣಾಮಗಳು	ಸೂಚನೆಗಳು	ನಿರ್ವಹಣೆ ಕ್ರಿಯೆ/ ಪ್ರತಿಕ್ರಿಯೆ ಯೋಜನೆ	ಸಮಯ ಮಿತಿ
1	ಸಂಸ್ಥೆಯು ಹೊಸ ವಿಂಡೋಸ್ ಫೈರ್‌ವಾಲ್‌ಗಳನ್ನು ಮತ್ತು ಸರ್ವರ್‌ಗಳನ್ನು ಸಂರಚಿಸಲು ಟೇಂಪ್ಲೇಟ್‌ಗಳನ್ನು ಬಳಸುತ್ತದೆ, ಆದರೆ ಇತರ ತಂತ್ರಜ್ಞಾನಗಳಾದ DBMS,UNIX ಮತ್ತು LINUX ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ದಾಖಲಾತಿ ತಂತ್ರಜ್ಞಾನ ಸಂರಚನಾ ಮಾನದಂಡವನ್ನು ಹೊಂದಿಲ್ಲ.	ಪರಿಸ್ಥಿತಿಯು, ಸಂಸ್ಥೆಯ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಅನಧಿಕೃತ ಪ್ರವೇಶದ ಅಪಾಯವನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ.	ಸಂಸ್ಥೆಯು ಸ್ಥಳದಲಿ ್ಲರುವ ತಂತ್ರಜ್ಞಾನಗಳ ಆಧಾರದ ಮೇಲೆ ಸಂರಚನಾ ಮಾನದಂಡವನ್ನು ದಾಖಲೆ ಮಾಡುವುದನ್ನು ಮುಂದುವರೆಸಬೇಕು	ಜವಾಬ್ದಾರಿಯುತ ವ್ಯಕ್ತಿ: Chief Information Officer(CIO): ಆಡಿಟರ್‌ನ ಸೂಚನೆಗಳನ್ನು ಅಳವಡಿಸಲಾಗುತ್ತದೆ. ಭದ್ರತಾ ತಂಡವು ಸಂರಚನಾ ಮಾನದಂಡಗಳ ದಾಖಲಾತಿಗಳನ್ನು ಮುಂದುವರೆಸುತ್ತಾರೆ. ವಿಶೇಷವಾಗಿ ಯುನಿಕ್ಸ್ ಮತ್ತು ಲಿನಕ್ಸ್ O.S ಗಾಗಿ	ಮುಂದಿನ ಕ್ರಮದ ಅಗತ್ಯವಿಲ್ಲ.

© NIMI NOT TO BE REPUBLISHED

2	ಲಿನಕ್ಸ್ ಆಧರಿತ ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಪ್ಯಾಚ್‌ಗಳು ಕಾಲಕ್ಕೆ ತಕ್ಕಂತೆ ಆಧುನೀಕರಣ ಹೊಂದಿಲ್ಲ	ಪರಿಸ್ಥಿತಿಯು, ಸಂಸ್ಥೆಯ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಅನಧಿಕೃತ ಪ್ರವೇಶದ ಅಪಾಯವನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ. ಇದು ವ್ಯವಸ್ಥೆಯ ವಿಫಲತೆಯ ಅಪಾಯವನ್ನೂ ಸಹ ಹೆಚ್ಚಿಸುತ್ತದೆ.	ಲಿನಕ್ಸ್ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ರನ್ ಆಗುತ್ತಿರುವ ಸರ್ವರ್‌ನಲ್ಲಿ ಇತ್ತೀಚಿನ ಪ್ಯಾಚ್‌ಗಳನ್ನು ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಿ.	ಜವಾಬ್ದಾರಿಯುತ ವ್ಯಕ್ತಿ: Chief Information Officer. ಇದು ಪೂರೈಕೆದಾರರು ಮತ್ತು ಭದ್ರತಾ ತಂಡದ ನಡುವೆ ರೆಡ್‌ಹ್ಯಾಟ್ ಲಿನಕ್ಸ್‌ನ ನಿರ್ವಹಣಾ ಒಪ್ಪಂದಗಳಲಿರುವ ಗೊಂದಲಗಳ ಪರಿಣಾಮವಾಗಿದೆ.	ಮುಂದಿನ ಕ್ರಮದ ಅಗತ್ಯವಿಲ್ಲ.
3	ಬಳಕೆದಾರರು ತಮ್ಮ ವರ್ಕ್‌ಸ್ಟೇಷನ್‌ನ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಟರ್‌ಗಳು ತಾವೇ ಆಗಿರುತ್ತಾರೆ.ಇದರಿಂದ ಅವರು ತಮ್ಮ ವರ್ಕ್‌ಸ್ಟೇಷನ್ ಆ್ಯಂಟಿವೈರಸ್‌ನ್ನು ಮತ್ತು ಕಾರ್ಯನಿರ್ವಹಿಸದ ಸಿಸ್ಟಮ್‌ನ ಸಂರಚನೆಗಳನ್ನು ನಿಷ್ಕ್ರೀಯಗೊಳಿಸಬಹುದು.	ಪರಿಸ್ಥಿತಿಯು, ಸಂಸ್ಥೆಯ ಸಿಸ್ಟಮ್‌ಗಳಿಗೆ ಅನಧಿಕೃತ ಪ್ರವೇಶದ ಅಪಾಯವನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ. ಇದು ವ್ಯವಸ್ಥೆಯ ವಿಫಲತೆಯ ಅಪಾಯವನ್ನೂ ಸಹ ಹೆಚ್ಚಿಸುತ್ತದೆ.	ವರ್ಕ್‌ಸ್ಟೇಷನ್ ಅಡ್ಮಿನಿಸ್ಟ್ರೇಷನ್ ಸೌಲಭ್ಯಗಳನ್ನು ತೆಗೆದುಹಾಕಿ.	ಜವಾಬ್ದಾರಿಯುತ ವ್ಯಕ್ತಿ: Chief Information Officer. ಪರಿಸ್ಥಿತಿಯು ವಿಶೇಷವಾಗಿ ಬಳಕೆದಾರರು ತಮ್ಮ ಕೆಲಸಗಳನ್ನು ಮಾಡಲು ಸೌಲಭ್ಯಗಳನ್ನು ಹೊಂದಿರುವ ಐತ್ತೀಚಿನ ಸರ್ವರ್ ಹೊಂದಿರುವ ವಿಂಡೋಸ್ 2000 ನಲ್ಲಿ ಈ ಪರಿಸ್ಥಿತಿಯು ಪ್ರಚಲಿತವಾಗಿದೆ.	ನವೆಂಬರ 1, 2014

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ವೈರಸ್‌ಗಳಿಂದ , ಸ್ಪೈವೇರ್‌ಗಳಿಂದ ಮತ್ತು ಇತರ ದುರುದ್ದೇಶಪೂರಿತ ಕೋಡ್‌ಗಳಿಂದ ಮಾಹಿತಿಯನ್ನು, ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತು ನೆಟ್‌ವರ್ಕ್‌ಗಳನ್ನು ರಕ್ಷಿಸಿ (Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code)

ಗೌಪ್ಯತೆ ರಕ್ಷಣೆ ಮತ್ತು ಐಟಿ ಕಾಯಿದೆ (Privacy Protection and IT Act)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಮಾಹಿತಿ ಗೌಪ್ಯತೆಯಲ್ಲಿ ವಿವರಿಸಿ
- ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸುವ ವಿಧಾನವನ್ನು ವಿವರಿಸಿ
- ಅತ್ಯುತ್ತಮ ಆನ್‌ಲೈನ್ ಗೌಪ್ಯತೆ ಅಭ್ಯಾಸಗಳನ್ನು ವಿವರಿಸಿ.
- ಐಟಿ ಕಾಯಿದೆ ಎಂದರೇನು ಎಂಬುದರ ಬಗ್ಗೆ ವಿವರಿಸಿ.
- ಸೈಬರ್ ಅಪರಾಧಗಳ ಬಗ್ಗೆ ವಿವರಿಸಿ.

ಪರಿಚಯ (Introduction)

ಗೌಪ್ಯತೆ (privacy) ಎಂದರೆ ಒಬ್ಬ ವ್ಯಕ್ತಿ ಅಥವಾ ಗುಂಪು ತಮ್ಮ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು, ಡಾಟಾ ಅಥವಾ ಮಾಹಿತಿಯನ್ನು ಪ್ರತ್ಯೇಕವಾಗಿರಿಸುವ ಮತ್ತು ಅದನ್ನು ತಮ್ಮ ಆಯ್ಕೆಯಂತೆ ಹಂಚಿಕೊಳ್ಳುವ ಸಾಮರ್ಥ್ಯವಾಗಿದೆ. ಒಬ್ಬ ವ್ಯಕ್ತಿಗೆ ಏನಾದರೂ ಖಾಸಗಿಯಾಗಿರುವಾಗ, ಸಾಮಾನ್ಯವಾಗಿ ಅದು ವಿಶೇಷ ಅಥವಾ ಸೂಕ್ಷ್ಮ ವಿಷಯವಾಗಿರುತ್ತದೆ

ಮಾಹಿತಿ ಅಥವಾ ಡಾಟಾ ಗೌಪ್ಯತೆ ಎಂದರೆ ತಂತ್ರಜ್ಞಾನ ಮತ್ತು ಕಾನೂನು ಹಕ್ಕುಗಳ ನಡುವೆ ವಿಕಸನಗೊಳ್ಳುತ್ತಿರುವ ಸಂಬಂಧವನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ಒಬ್ಬರ ಸ್ವಂತ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹ ಮಾಡುವುದು ಮತ್ತು ಹಂಚುವುದರಲ್ಲಿನ ಗೌಪ್ಯತೆಯು ಮಾಹಿತಿ ಗೌಪ್ಯತೆಯಾಗಿದೆ. ಒಬ್ಬ ವ್ಯಕ್ತಿ ಅಥವಾ ವ್ಯಕ್ತಿಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ, ಗುರುತಿಸಬಹುದಾದ ಡಾಟಾವನ್ನು ಡಿಜಿಟಲ್ ರೂಪದಲ್ಲಿ ಅಥವಾ ಬೇರೆ ರೂಪದಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಸ್ಟೋರ್ ಮಾಡಲಾಗುತ್ತದೆಯೊ ಎಲ್ಲಿ ಇರುತ್ತದೆ ಅಲ್ಲಿ ಗೌಪ್ಯತೆಯ ಕಾಳಜಿಯು ಇರುತ್ತದೆ. ಕೆಲವು ಸಂದರ್ಭಗಳಲ್ಲಿ ಈ ಕಾಳಜಿಗಳು ಡಾಟಾವನ್ನು ಹೇಗೆ ಒಟ್ಟುಗೂಡಿಸುವುದು, ಸಂಗ್ರಹಿಸುವುದು ಮತ್ತು ಸಂಯೋಜಿಸುವುದು ಎಂಬುದನ್ನು ಉಲ್ಲೇಖಿಸುತ್ತವೆ. ಇತರ ಸಂದರ್ಭಗಳಲ್ಲಿ ಯಾರಿಗೆ ಮಾಹಿತಿಯ ಪ್ರವೇಶವನ್ನು ನೀಡಲಾಗಿದೆ ಎನ್ನುವ ಸಮಸ್ಯೆಯಾಗಿರುತ್ತದೆ. ಇತರ ಸಮಸ್ಯೆಗಳು, ಒಬ್ಬ ವ್ಯಕ್ತಿಯು ಡಾಟಾದ ಬಗ್ಗೆ ಯಾವುದೇ ಮಾಲೀಕತ್ವದ ಹಕ್ಕುಗಳನ್ನು ಹೊಂದಿದ್ದಾನೆಯೆ, ಮತ್ತು/ ಅಥವಾ ಅದನ್ನು ವೀಕ್ಷಿಸುವ ಹಕ್ಕನ್ನು, ಪರಿಶೀಲಿಸುವ ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಸವಾಲು ಮಾಡುವ ಹಕ್ಕನ್ನು ಹೊಂದಿದ್ದಾನೆಯೆ ಎಂಬುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಮಾಹಿತಿ ಭದ್ರತೆಯು ಅನಧಿಕೃತ ವ್ಯಕ್ತಿಗಳನ್ನು ಅಥವಾ ವ್ಯವಸ್ಥೆಗಳನ್ನು ನಿರ್ಬಂಧಿತ ಮಾಹಿತಿಗೆ ಪ್ರವೇಶ ಪಡೆಯುವುದನ್ನು ನಿರ್ಬಂಧಿಸುತ್ತದೆ. ಗೌಪ್ಯತೆಯು ನಿಯಮಗಳು ಮತ್ತು ನಿರ್ಬಂಧಗಳ ಸಂಗ್ರಹವಾಗಿದೆ. ಇದು ಯಾವುದೇ ಮಾಧ್ಯಮದಲ್ಲಿ, ಪ್ರವೇಶವನ್ನು ಹೇಗೆ ಮತ್ತು ಯಾವಾಗ ಅಧಿಕೃತಗೊಳಿಸಬೇಕು ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ. ಇದು ಉತ್ತಮ ಭದ್ರತೆ ಮತ್ತು ಗೌಪ್ಯತೆಯ ಅಭ್ಯಾಸಗಳು ಒಂದನ್ನೊಂದು ಅವಲಂಬಿಸಿವೆ ಎಂಬುದನ್ನು ಅನುಸರಿಸುತ್ತದೆ. ಗೌಪ್ಯತೆಯು ಭಾಗಶಃ ಭದ್ರತೆಯನ್ನು ಅತಿಕ್ರಮಿಸುತ್ತದೆ. ಇದು ಮಾಹಿತಿಯ ಸೂಕ್ತ ಬಳಕೆ ಮತ್ತು ರಕ್ಷಣೆಯ ಪರಿಕಲ್ಪನೆಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಕಂಪ್ಯೂಟರ್ ಭದ್ರತೆ ಮತ್ತು ಗೌಪ್ಯತೆಗಳ ನಡುವಿನ ಸಂಬಂಧವು, ಸಾಕಷ್ಟು ಕಂಪ್ಯೂಟರ್‌ನ ಭದ್ರತೆ ಅಥವಾ ಅದರ ಕೊರತೆಯಲ್ಲಿ ಇರುತ್ತದೆ. ಇದು ಬಳಕೆದಾರರು ನಿರೀಕ್ಷಿಸಬಹುದಾದ ಗೌಪ್ಯತೆಯ ಮಟ್ಟವನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ. ಜನರು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಅನೇಕ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಬಳಸುತ್ತಾರೆ - ವ್ಯವಹಾರ, ಬ್ಯಾಂಕಿಂಗ್, ಸಾಮಾಜಿಕ. ಮತ್ತು ಖಾಸಗಿ ಮಾಹಿತಿಯ ಸಂಗ್ರಹಣೆ. ಕಂಪ್ಯೂಟರ್ ಭದ್ರತೆಯ ಉಲ್ಲಂಘನೆಯಾದರೆ, ಅದು ಈ ರೀತಿಯ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವುದರಲ್ಲಿ ಋಣಾತ್ಮಕ ಪರಿಣಾಮವನ್ನು ಬೀರುತ್ತದೆ.

ಇ- ಕಾಮರ್ಸ್ ಕ್ಷೇತ್ರದಲ್ಲಿ, ಕಂಪ್ಯೂಟರ್ ಭದ್ರತೆ ಮತ್ತು ಗೌಪ್ಯತೆಯಲ್ಲಿನ ಸಮಸ್ಯೆಗಳು ವ್ಯಾಪಾರದ ಪಕ್ಷಗಳ ನಡುವೆ ನಂಬಿಕೆಯ ಮಟ್ಟವನ್ನು ನಿರ್ಧರಿಸುತ್ತವೆ. ಎರಡೂ ಕಡೆಗೆ ಭದ್ರತಾ ಉಲ್ಲಂಘನೆಯ ಯಾವುದೇ ಅನುಮಾನವಿದ್ದಲ್ಲಿ ಅದು ನಂಬಿಕೆಯ ನಾಶ ಮತ್ತು ವ್ಯಾಪಾರದ ಸಂಬಂಧದ ಅಂತ್ಯಕ್ಕೆ ಕಾರಣವಾಗುತ್ತದೆ. ಇದು ವ್ಯಾಪಾರದ ವ್ಯಕ್ತಿಗಳಿಗೆ ಸಂಬಂಧವಿರದ ಮೂರನೇ ವ್ಯಕ್ತಿಯಿಂದ ಅಪಾಯಗಳು ಮತ್ತು ಬೆದರಿಕೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಅಸಮರ್ಪಕ ಅಥವಾ ಅಸ್ತಿತ್ವದಲ್ಲಿಲ್ಲದ ಬಹಿರಂಗಪಡಿಸುವಿಕೆಯ ನಿಯಂತ್ರಣವು ಗೌಪ್ಯತೆ ಸಮಸ್ಯೆಗಳಿಗೆ ಮೂಲ ಕಾರಣವಾಗಿರಬಹುದು. ಡಾಟಾ ಗೌಪ್ಯತೆಯ ಸಮಸ್ಯೆಗಳು

© NIMI NOT TO BE REPUBLISHED

ಮಾಹಿತಿಗೆ ಪ್ರತಿಕ್ರಿಯೆಯಾಗಿ ಅನೇಕ ಮೂಲಗಳಿಂದ ಉದ್ಭವಿಸುತ್ತವೆ. ಉದಾಹರಣೆಗೆ :

- ಆರೋಗ್ಯ ರಕ್ಷಣೆ ದಾಖಲೆಗಳು
- ತನಿಖೆಗಳು ಮತ್ತು ಪ್ರಕ್ರಿಯೆಗಳು
- ಹಣಕಾಸು ಸಂಸ್ಥೆಗಳು ಮತ್ತು ವಹಿವಾಟುಗಳು
- ನಿವಾಸ ಮತ್ತು ಭೌಗೋಳಿಕ ದಾಖಲೆಗಳು
- ರಕ್ಷಣಾ ಡಾಟಾ
- ಗೌಪ್ಯತೆಯ ಉಲ್ಲಂಘನೆ
- ಸ್ಥಳ ಆಧಾರಿತ ಸೇವೆ ಮತ್ತು ಜಿಯೊಲೊಕೇಷನ್
- ವೈಜ್ಞಾನಿಕ ಸಂಶೋಧನೆ ಇತ್ಯಾದಿಗಳು

ಡಾಟಾ ಗೌಪ್ಯತೆಯ ಸವಾಲು ಎಂದರೆ, ವೈಯಕ್ತಿಕವಾಗಿ ಗುರುತಿಸಬಹುದಾದ ಮಾಹಿತಿಯನ್ನು ರಕ್ಷಿಸುವಾಗ ಡಾಟಾವನ್ನು ಹಂಚಿಕೊಳ್ಳುವುದಾಗಿದೆ. ಡಾಟಾ ಭದ್ರತೆ ಮತ್ತು ಮಾಹಿತಿ ಭದ್ರತೆ ವಿನ್ಯಾಸ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಮತು ಹಾರ್ಡ್‌ವೇರ್ ಬಳಕೆ ಮತ್ತು ಮಾನವ ಸಂಪನ್ಮೂಲಗಳು, ಕ್ಷೇತ್ರಗಳು ಈ ಸಮಸ್ಯೆಗೆ ಪರಿಹರವಾಗಿವೆ. ಡಾಟಾ ರಕ್ಷಣೆಗೆ ಸಂಬಂಧಿಸಿದ ಕಾನೂನುಗಳು ಮತ್ತು ನಿಬಂಧನೆಗಳು ನಿರಂತರವಾಗಿ ಬದಲಾಗುತ್ತಿರುವುದರಿಂದ, ಕಾನೂನಿನಲ್ಲಿ ಯಾವುದೇ ಬದಲಾವಣೆಯಾದಲ್ಲಿ ಡಾಟಾ ಗೌಪ್ಯತೆ ಮತ್ತು ಭದ್ರತಾ ನಿಯಮಗಳೊಂದಿಗೆ ನಿಮ್ಮ ಅನುಸರಣೆಯನ್ನು ನಿರಂತರವಾಗಿ ಮರುಮೌಲ್ಯಮಾಪನ ಮಾಡುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ.

ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸುವುದು (Protecting privacy in information systems) : ವಿವಿಧ ಗೌಪ್ಯತಾ ನಿಯಮಗಳನ್ನು ಹೊಂದಿದ ವೈವಿಧ್ಯಮಯವಾದ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳು ಪರಸ್ಪರ ಸಂಪರ್ಕ ಹೊಂದಿರುತ್ತವೆ ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತವೆ, ನೀತಿ ಉಪಕರಣಗಳನ್ನು ಸಮನ್ವಯಗೊಳಿಸುವ ಗೌಪ್ಯತಾ ಪಾಲಿಸಿ ನಿಯಮಗಳನ್ನು (ಮತ್ತು ಕಾನೂನುಗಳು) ಜಾರಿಗೊಳಿಸುವ ಮತ್ತು ಮೇಲ್ವಿಚಾರಣೆ ಮಾಡುವ ಅಗತ್ಯವಿರುತ್ತದೆ, ವಾಣಿಜ್ಯ ಐಟಿ ವ್ಯವಸ್ಥೆಗಳಲ್ಲಿ ಗೌಪ್ಯತಾ ರಕ್ಷಣೆಗಾಗಿ ಎರಡು ವಿಧದ ತಂತ್ರಜ್ಞಾನಗಳಿವೆ : ಸಂವಹನ (communication) ಮತ್ತು ಜಾರಿಗೊಳಿಸುವಿಕೆ (enforcement).

ನೀತಿ ಸಂವಹನ P3P (Policy Communication P3P): ಇದು ಗೌಪ್ಯತೆಯ ಆದ್ಯತೆಗಳಿಗೆ ವೇದಿಕೆಯಾಗಿದೆ. P3P, ಗೌಪ್ಯತೆಯ ಅಭ್ಯಾಸಗಳ ಸಂವಹನಕ್ಕೆ ಮತ್ತು ವ್ಯಕ್ತಿಗಳ ಆದ್ಯತೆಗಳಿಗೆ ಅವುಗಳನ್ನು ಹೋಲಿಸುವುದಕ್ಕೆ ಮಾನದಂಡವಾಗಿದೆ.

ನೀತಿ ಜಾರಿಗೊಳಿಸುವುದು (Policy Enforcement)

- XACML - Extensible Access Control Markup Language ಇದರ ಪ್ರೈವಸಿ ಪ್ರೊಫೈಲ್ ಜೊತೆಗಿನ , ಮಶೀನ್ ಓದಬಲ್ಲ ಭಾಷೆಯಲ್ಲಿರುವ ಗೌಪ್ಯತೆಯ ನೀತಿಗಳನ್ನು ವ್ಯಕ್ತಪಡಿಸಲು ಒಂದು ಮಾನದಂಡವಾಗಿದೆ. ಇದನ್ನು ಐಟಿ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ನೀತಿಯನ್ನು ಜಾರಿಗೊಳಿಸಲು ಸಾಫ್ಟ್‌ವೇರ್ ಸಿಸ್ಟಮ್ ಬಳಸುತ್ತದೆ.
- EPAL - Enterprise Privacy Authorization Language ಇದು XACML ನ್ನು ಹೋಲುತ್ತದೆ, ಆದರೆ ಇದು ಇನ್ನೂ ಪ್ರಮಾಣಿತವಾಗಿಲ್ಲ.
- WS-Privacy - "Web Service Privacy" ಇದು ವೆಬ್ ಸೇವೆಗಳಲ್ಲಿ ಗೌಪ್ಯತೆ ನೀತಿಗಳನ್ನು ಸಂವಹನ ನಡೆಸಲು ಒಂದು ನಿರ್ದಿಷ್ಟ ವಿವರಣೆಯಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ಒಂದು ವೆಬ್ ಸೇವೆಯ ಸಂದೇಶದಲ್ಲಿ ಗೌಪ್ಯತಾ ನೀತಿಯ ಮಾಹಿತಿಯನ್ನು SOAP ನಲ್ಲಿ ಹೇಗೆ ಜೋಡಿಸಬಹುದು ಎಂಬುದನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತದೆ.

ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸುವುದು (Protecting Privacy on the Internet)

ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ನೀವು ಯಾವಾಗಲೂ ನಿಮ್ಮ ಬಗ್ಗೆ ಬಹಳಷ್ಟು ಮಾಹಿತಿಯನ್ನು ನೀಡುತ್ತಿರಿ. ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡದ ಇ - ಮೇಲ್‌ಗಳನ್ನು ಇ ಮೇಲ್ ಸರ್ವರ್‌ನ ನಿರ್ವಾಹಕರು ಓದಬಹುದು. ಇಲ್ಲಿ ಸಂಪರ್ಕವನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡಲಾಗಿರುವುದಿಲ್ಲ (https ಇರುವುದಿಲ್ಲ). ಇಂಟರ್‌ನೆಟ್ ಸೇವಾ ಪೂರೈಕೆದಾರರು ಮತ್ತು ಸಂಪರ್ಕದಸಂಚಾರವನ್ನುಗ್ರಹಿಸುವ ಇತರ ಪಕ್ಷಗಳು ವಿಷಯಗಳನ್ನು ತಿಳಿಯಲು ಸಾಧ್ಯವಾಗುತ್ತದೆ. ಇದಲ್ಲದೇ ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಯಾವುದೇ ರೀತಿಯ ಟ್ರಾಫಿಕ್‌ಗೆ ಇದು ಅನ್ವಯಿಸುತ್ತದೆ (ವೆಬ್ ಬ್ರೌಸಿಂಗ್, ತ್ವರಿತ ಸಂದೇಶ ಕಳುಹಿಸುವುದು, ...) . ಹೆಚ್ಚಿನ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ನೀಡಬಾರದು. ಇ-ಮೇಲ್‌ಗಳನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡಬಹುದು ಮತ್ತು ವೆಬ್‌ಪುಟಗಳನ್ನು ಬ್ರೌಸ್ ಮಾಡುವುದು ಮತ್ತು ಇತರ ಆನ್‌ಲೈನ್ ಚಟುವಟಿಕೆಗಳನ್ನು ಅನಾಮಧೇಯ ವ್ಯಕ್ತಿಗಳು ಪತ್ತೆ ಹಚ್ಚಲಾಗದಂತೆ ಮಾಡಬಹುದು, ಅಥವಾ ಅವುಗಳು ವಿಶ್ವಾಸಾರ್ಹವಲ್ಲದ ಸಂದರ್ಭಗಳಲ್ಲಿ. ಓಪನ್ ಸೋರ್ಸ್ ವಿತರಣೆ ಅನಾಮಧೇಯರು , ಇದನ್ನು ಮಿಕ್ಸ್‌ನೆಟ್ಸ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಹೆಸರಾಂತ ಓಪನ್ ಸೋರ್ಸ್ ಮಿಕ್ಸ್‌ ನೆಟ್ಸ್ ಗಳು - I2P - Anonymous Network ಅಥವಾ tor.

ನಿಮ್ಮ ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸಿ (Protect Your Privacy)

ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಉಪಯೋಗಿಸುವಾಗ ನಿಮ್ಮ ಗೌಪ್ಯತೆಯನ್ನು, ಆನ್‌ಲೈನ್ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿ ರಕ್ಷಿಸಲು , ಮತ್ತು ವಂಚನೆ ಮತ್ತು ನಿಂದನೆಯನ್ನು ತಡೆಯಲು ಕೆಲವು ಸಲಹೆಗಳು ಮತ್ತು ಮಾರ್ಗಸೂಚಿಗಳ ಪಟ್ಟಿಯನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ :

© NIMI NOT TO BE REPUBLISHED

- **ಹೊಸ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಪಡೆಯಿರಿ:** ನಿಮ್ಮ ಪ್ರತಿಯೊಂದು ಆನ್‌ಲೈನ್ ಖಾತೆಗೆ ವಿಭಿನ್ನ, ಬಲವಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಬಳಸಿರಿ. ಒಂದು ಪಾಸ್‌ವರ್ಡ್ ರಾಜಿಯಾದರೆ ಉಳಿದವುಗಳು ಸುರಕ್ಷಿತವಾಗಿರುತ್ತವೆ. ಬಲವಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳು ಅಕ್ಷರಗಳನ್ನು, ಸಂಖ್ಯೆಗಳನ್ನು ವಿಭಿನ್ನ ಕೇಸ್‌ಗಳನ್ನು ಮತ್ತು ಚಿನ್ಹೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್‌ನ ಸಾಮರ್ಥ್ಯವನ್ನು ಪರಿಶೀಲಿಸಬಹುದು.
- **ಹಳೆಯ ಆನ್‌ಲೈನ್ ಖಾತೆಗಳನ್ನು ಮುಚ್ಚಿ :** ಬಳಕೆಯಾಗದ ಆನ್‌ಲೈನ್ ಖಾತೆಗಳು ಒಂದು ಹೊಣೆಗಾರಿಕೆಯಾಗಿವೆ. ಹ್ಯಾಕರ್‌ಗಳು ಅವುಗಳನ್ನು ನಿಮ್ಮ ಹೆಚ್ಚು ಪ್ರಮುಖ ಖಾತೆಗಳ ಒಳಗೆ ನುಸುಳಲು ಬಳಸಬಹುದು. ಅವುಗಳನ್ನು ತೆಗೆದುಹಾಕಿ.
- ನಿಮ್ಮ ಸ್ನೇಹಿತರನ್ನು ಕಡಿಮೆ ಮಾಡಿ.
- **ಕಾಗದ ರಹಿತರಾಗಿ :** ಸೂಕ್ಷ್ಮ ಡಾಟಾವನ್ನು ಆನ್‌ಲೈನ್‌ನಲ್ಲಿ ಅಥವಾ ನಿಮ್ಮ ಮೇಲ್ ಖಾತೆಗಳಲ್ಲಿ ಇರಿಸಬೇಡಿ.
- **ಸೂಕ್ಷ್ಮ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಚೂರುಚೂರು ಮಾಡಿ :** ಸೂಕ್ಷ್ಮ ಡಾಟಾವನ್ನು ಹೊಂದಿರುವ ಅನಗತ್ಯ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕಿ. ಶ್ರೆಡರ್‌ನ್ನು ಬಳಸಿ ಅವುಗಳನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ವಿಲೇವಾರಿ ಮಾಡಿ,

ಬ್ರೌಸರ್ ಗೌಪ್ಯತೆ (Browser Privacy)

ಅಧುನಿಕ ಬ್ರೌಸರ್‌ಗಳು ಗೌಪ್ಯತೆಯನ್ನು ಹೆಚ್ಚಿಸುವ ಪ್ರಭಾವಶಾಲಿ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಮತ್ತು ಆಯ್ಕೆಗಳನ್ನು ಹೊಂದಿವೆ. ಉದಾಹರಣೆಗೆ, ನೀವು ಅನುಮಾನಾಸ್ಪದ ಅಥವಾ ಮೋಸದ ವೆಬ್‌ಸೈಟ್‌ಗಳಿಗೆ ಭೇಟಿ ನೀಡುವ ಮೊದಲು ನಿಮಗೆ ಎಚ್ಚರಿಕೆ ನೀಡುತ್ತವೆ ಮತ್ತು ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿ ಕುಕಿಗಳಂತಹ ಟ್ರ್ಯಾಕ್ ಮಾಡುವ ಫೈಲ್‌ಗಳನ್ನು ಡೌನ್‌ಲೋಡ್ ಮಾಡದೆಯೆ ವೆಬ್ ಅನ್ನು ಸರ್ಫ್ ಮಾಡಲು ನಿಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತವೆ. ಇದಲ್ಲದೇ ಹೆಚ್ಚಿನ ಬ್ರೌಸರ್‌ಗಳು, ವೆಬ್‌ಸೈಟ್ SSL ಅನ್ನು ಬಳಸಿದಾಗ ನಿಮಗೆ ತಿಳಿಸಬಹುದು. ಇದು ನಿಮ್ಮ ಡಾಟಾವನ್ನು ಎನ್‌ಕ್ರಿಪ್ಟ್ ಮಾಡುವ ಭದ್ರತಾ ಕ್ರಮವಾಗಿದೆ. ಒಂದು ವೆಬ್‌ಸೈಟ್ SSL ಅನ್ನು ಬಳಸಿದಾಗ, ಬ್ರೌಸರ್ ಇದನ್ನು ನಿಮಗೆ ಪ್ಯಾಡ್‌ಲಾಕ್ ಐಕನ್‌ನ್ನು ಪ್ರದರ್ಶಿಸುವ ಮೂಲಕ ತಿಳಿಸುತ್ತದೆ (ಪ್ಯಾಡ್‌ಲಾಕ್ ಐಕನ್ ನಿಮ್ಮ ಬ್ರೌಸರ್‌ನ ಕೆಳಗಿನ ಬಾರ್‌ನಲ್ಲಿರುತ್ತದೆ). ಅಥವಾ ಅಡ್ರೆಸ್ ಬಾರ್‌ನಲ್ಲಿರುವ ವೆಬ್‌ಸೈಟ್‌ನ ಹೆಸರನ್ನು ಹಸಿರು ಬಣ್ಣದಲ್ಲಿ ಹೈಲೈಟ್ ಮಾಡುವ ಮೂಲಕ ಸೂಚಿಸುತ್ತದೆ.

ವಿಶ್ವಾಸಾರ್ಹ ವೆಬ್‌ಸೈಟ್‌ಗಳಿಗೆ ಮಾತ್ರ ಭೇಟಿ ನೀಡಿ. ನೀವು ತೆರೆಯುತ್ತಿರುವ ಸೈಟ್ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳಲು Site Advisor ನಂತಹ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ. ಇವತ್ತು ವೆಬ್‌ಸೈಟ್‌ಗಳು ಹೆಚ್ಚಿನ ಮಾಹಿತಿಯನ್ನು ಹಂಚಿಕೊಳ್ಳುವುದರಿಂದ, ಅವು ತಮ್ಮ ಬಳಕೆದಾರರಿಗೆ ಈ ಹಂಚಿಕೆ ಚಟುವಟಿಕೆಗಳ ಮೇಲೆ ಹೆಚ್ಚಿನ ನಿರ್ದಿಷ್ಟತೆ ಮತ್ತು ನಿಯಂತ್ರಣವನ್ನು ಸಹ ಒದಗಿಸುತ್ತವೆ. ಅನೇಕ ವೆಬ್‌ಸೈಟ್‌ಗಳಲ್ಲಿ ನೀವು ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ಅಥವಾ ವಿಷಯವನ್ನು ಹಂಚಿಕೊಳ್ಳುವಾಗ ನಿಮ್ಮ ಪ್ರೇಕ್ಷಕರನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ಅದು ಒಬ್ಬ ವ್ಯಕ್ತಿಯಾಗಿರಬಹುದು ಅಥವಾ ಇಡೀ ಸಾರ್ವಜನಿಕ ಪ್ರೇಕ್ಷಕರಾಗಿರಬಹುದು.

ಕಳೆದ ದಶಕದಲ್ಲಿ ಇ-ಮೇಲ್ ಬದಲಾಗದೇ ಹಾಗೇ ಉಳಿದಿದೆ. ಅದಾಗ್ಯೂ ಇಮೇಲ್‌ನ್ನು ಬಳಸಿಕೊಳ್ಳುವ ವಿಧಾನಗಳು ಗಮನಾರ್ಹವಾಗಿ ವಿಕಸನಗೊಂಡಿವೆ ಮತ್ತು ಇ ಮೇಲ್ ಪರಿಸರದಲ್ಲಿ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿಯನ್ನು ರಕ್ಷಿಸುವುದು ಹೆಚ್ಚು ಸವಾಲಿನ ಕೆಲಸವಾಗಿದೆ. ಕಳೆದ ದಶಕದಲ್ಲಿ ಹ್ಯಾಕಿಂಗ್ ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿದೆ ಮತ್ತು ಫಿಶಿಂಗ್ ತಂತ್ರಗಳು ಹೆಚ್ಚು ವಿಸ್ತಾರಗೊಂಡಿವೆ. ಇ ಮೇಲ್ ಬಳಸುವಾಗ ನಿಮ್ಮ ಗೌಪ್ಯತೆಯನ್ನು ರಕ್ಷಿಸಿಕೊಳ್ಳಲು ಕೆಲವು ತಂತ್ರಗಳನ್ನು ಇಲ್ಲಿ ನೀಡಲಾಗಿದೆ :

1 ಇನ್ನೊಂದು ಸೆಕೆಂಡರಿ "spam" ಇ ಮೇಲ್ ಅಡ್ರೆಸ್‌ನ್ನು ಬಳಸಿ.

2 ಬಲವಾದ ಭದ್ರತೆ ಮತ್ತು ಸ್ಪ್ಯಾಮ್ ಫಿಲ್ಟರ್‌ಗಳನ್ನು ಹೊಂದಿರುವ ಇ ಮೇಲ್ ಸರ್ವೀಸ್ ಪ್ರೊವೈಡರ್‌ನ್ನು ಬಳಸಿ.

3 ಇ ಮೇಲ್‌ಗಳನ್ನು ತೆರೆಯುವಾಗ ಜಾಗರೂಕರಾಗಿರಿ.

4 ಇ ಮೇಲ್ ಮುಕ್ತತೆ ಮತ್ತು ಪರಸ್ಪರ ಸಂಪರ್ಕದ ಕಡೆಗೆ ವಿಕಸನಗೊಳ್ಳುತ್ತಿದೆ ಎಂಬುದನ್ನು ಗುರುತಿಸಿ.

5 ಬಲವಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಬಳಸಿ ಮತ್ತು ಸೈನ್ ಔಟ್ ಮಾಡಲು ಮರೆಯದಿರಿ.

ಅತ್ಯುತ್ತಮ ಆನ್‌ಲೈನ್ ಗೌಪ್ಯತೆ ಅಭ್ಯಾಸಗಳು

1 ವೈಯಕ್ತಿಕ ಮಾಹಿತಿ ಹಂಚಿಕೆಯನ್ನು ಕಡಿಮೆ ಮಾಡಿ.

2 ವೆಬ್‌ಸೈಟ್‌ನಲ್ಲಿರುವ ಟ್ರಸ್ಟ್ ಮಾರ್ಕ್‌ಗಳನ್ನು ನೋಡಿ ಮತ್ತು ಅವುಗಳ ದೃಢೀಕರಣವನ್ನು ಪರಿಶೀಲಿಸಿ.

3 ಆನ್‌ಲೈ ನ್ ಶಾಪಿಂಗ್ ಮಾಡುವಾಗ ತಾತ್ಕಾಲಿಕ ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಸಂಖ್ಯೆಗಳನ್ನು ಪರಿಗಣಿಸಿ.

4 ಬಲವಾದ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಬಳಸಿ ಮತ್ತು ಸೈನ್ ಔಟ್ ಮಾಡಲು ಮರೆಯದಿರಿ.

5 ನಿಮ್ಮ ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಆಗಾಗ್ಗೆ ಬದಲಾಯಿಸಿ.

6 ಆ್ಯಂಟಿ ವೈರಸ್ ಮತ್ತು ಆ್ಯಂಟಿ ಸ್ಪೈವೇರ್ ರಕ್ಷಣೆಯನ್ನು ಬಳಸಿ.

7 ಬ್ರೌಸರ್ ಗೌಪ್ಯತೆಯನ್ನು ಹೆಚ್ಚಿಸುವ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಮತ್ತು ಆಯ್ಕೆಗಳ ಲಾಭವನ್ನು ಪಡೆದುಕೊಳ್ಳಿ.

8 ನಿಮ್ಮ ಬ್ರೌಸರ್ ಮತ್ತು ಇತರ ಪರಿಕರಗಳನ್ನು ನವೀಕರಿಸಿ.

©NIMI NOT TO BE REPUBLISHED

ಮೊಬೈಲ್ ಗೌಪ್ಯತೆ (Mobile Privacy)

1 ಮೊಬೈಲ್ ಕಳ್ಳತನ ಅಥವಾ ನಷ್ಟವಾಗುವುದರಿಂದ ಅದರಲ್ಲಿನ ನಿಮ್ಮ ವೈಯಕ್ತಿಕ ಮಾಹಿತಿ ಕಳ್ಳತನವಾಗುತ್ತದೆ - ಸೂಕ್ತ ಮುನ್ನೆಚ್ಚರಿಕೆಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳಿ.

2 ನಿಮ್ಮ ಮೊಬೈಲ್ ಸಾಧನವು ನಿಮ್ಮ ಸ್ಥಳವನ್ನು ತಿಳಿದಿರಬಹುದು ಮತ್ತು ಆ ಡಾಟಾವನ್ನು ಅಪ್ಲಿಕೇಷನ್‌ಗಳೊಂದಿಗೆ ಮತ್ತು ಜಾಹೀರಾತುದಾರೊಂದಿಗೆ ಹಂಚಿಕೊಳ್ಳಬಹುದು.

ಸೈಬರ್ ಅಪರಾಧದ ಕಾಯಿದೆಗಳು (CYBER CRIME ACT)

ಸೈಬರ್ ಲೋಕದ ಯುಗದಲ್ಲಿ ಕಂಪ್ಯೂಟರ್‌ಗಳ ಬಳಕೆ ಹೆಚ್ಚು ಜನಪ್ರಿಯವಾದಂತೆ ತಂತ್ರಜ್ಞಾನದ ಬೆಳವಣಿಗೆಯೂ ವಿಸ್ತರಣೆ ಹೊಂದಿತು ಮತ್ತು ಸೈಬರ್ ಶಬ್ದವೂ ಜನರಿಗೆ ಹೆಚ್ಚು ಚಿರಪರಿಚಿತವಾಯಿತು. ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನದ (IT) ವಿಕಾಸವು ಸೈಬರ್‌ಸ್ಪೇಸ್‌ಗೆ ಜನ್ಮ ನೀಡಿತು. ಇದರಲ್ಲಿ ಇಂಟರ್‌ನೆಟ್ ಎಲ್ಲ ಜನರಿಗೆ ಉನ್ನತ ತಂತ್ರಜ್ಞಾನದ ಬಳಕೆಯಿಂದ ಯಾವುದೇ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಲು, ಡಾಟಾ ಸಂಗ್ರಹಣೆಗೆ, ವಿಶ್ಲೇಷಣೆ ಇತ್ಯಾದಿಗಳಿಗಾಗಿ ಸಮಾನ ಅವಕಾಶಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ನೆಟಿಜೆನ್‌ಗಳ ಸಂಖ್ಯೆಯಲ್ಲಿನ ಹೆಚ್ಚಳದಿಂದಾಗಿ, ಸೈಬರ್ ಸ್ಪೇಸ್‌ನಲ್ಲಿ ತಂತ್ರಜ್ಞಾನದ ದುರ್ಬಳಕೆಯು ದೇಶೀಯ ಮತ್ತು ಅಂತರ್‌ರಾಷ್ಟ್ರೀಯ ಮಟ್ಟದಲ್ಲಿಯೂ ಸಹ ಸೈಬರ್ ಅಪರಾಧಗಳಿಗೆ ಜನ್ಮ ನೀಡಿತು.

ಕ್ರೈಮ್ ಎಂಬ ಪದದ ಸಾಮಾನ್ಯ ಅರ್ಥ "ಕ್ರಿಮಿನಲ್ ಮೊಕದ್ದಮೆಗಳ ಮೂಲಕ ಅನುಸರಿಸಬಹುದಾದ ಕಾನೂನು ತಪ್ಪು ಇದು ಶಿಕ್ಷೆಗೆ ಕಾರಣವಾಗವಾಗಬಹುದು", ಆದರೆ ಸೈಬರ್ ಕ್ರೈಮ್ ಎಂದರೆ "ಕಾನೂನು ಬಾಹಿರ ಕೃತ್ಯ ಇದರಲ್ಲಿ ಕಂಪ್ಯೂಟರ್ ಒಂದು ಸಾಧನವಾಗಿರುತ್ತದೆ ಅಥವಾ ಗುರಿ ಆಗಿರುತ್ತದೆ ಅಥವಾ ಎರಡೂ ಅಗಿರಬಹುದು".

ವಿಶ್ವದ 1 ನೇ ಕಂಪ್ಯೂಟರ್ ನಿರ್ದಿಷ್ಟ ಕಾನೂನನ್ನು 1970 ರಲ್ಲಿ ಜರ್ಮನ್ ಸ್ಟೇಟ್ ಆಫ್ ಹೆಸ್ಸೆ ಮೂಲಕ 'Data Protection Act, 1970' ನ ರೂಪದಲ್ಲಿ ಜಾರಿಗೆ ತರಲಾಯಿತು. ಇದು ಸೈಬರ್ ತಂತ್ರಜ್ಞಾನದ ಪ್ರಗತಿಯೊಂದಿಗೆ ಬಂದಿತು. ತಂತ್ರಜ್ಞಾನದ ಹೊರಹೊಮ್ಮುವಿಕೆಯಿಂದ ತಂತ್ರಜ್ಞಾನದ ದುರ್ಬಳಕೆಯೂ ಉನ್ನತ ಮಟ್ಟದಲ್ಲಿ ಹೆಚ್ಚಾಯಿತು. ನಂತರ ಸೈಬರ್ ಜಗತ್ತಿನಲ್ಲಿ ಅಪರಾಧ ಚಟುವಟಿಕೆಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ಮತ್ತು ತಂತ್ರಜ್ಞಾನ ಪ್ರಗತಿ ವ್ಯವಸ್ಥೆಯನ್ನು ರಕ್ಷಿಸಲು ಕಟ್ಟುನಿಟ್ಟಾದ ಶಾಸನಬದ್ಧ ಕಾನೂನುಗಳ ಅವಶ್ಯಕತೆಯು ಉಂಟಾಯಿತು. ಈ ಸಂದರ್ಭಗಳ ಅಡಿಯಲ್ಲಿ ಭಾರತೀಯ ಸಂಸತ್ತು 17 ನೇ ಅಕ್ಟೋಬರ್‌ಗೆ "INFORMATION TECHNOLOGY ACT, 2000" (ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಕಾಯಿದೆ, 2000) ನ್ನು ಅಂಗೀಕರಿಸಿತು. ಇದು ಇ -ಕಾಮರ್ಸ್, ಇ-ಆಡಳಿತ, ಇ-ಬ್ಯಾಂಕಿಂಗ್ ಮುಂತಾದ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ತಂತ್ರಜ್ಞಾನವನ್ನು ಎದುರಿಸಲು ಸಮಗ್ರ ಕಾನೂನನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ಸೈಬರ್ ಅಪರಾಧಗಳ ಕ್ಷೇತ್ರದಲ್ಲಿ ದಂಡಗಳು ಮತ್ತು ಶಿಕ್ಷೆಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

ವಾಸ್ತವವಾಗಿ ಸೈಬರ್ ಅಪರಾಧಗಳು (Cyber Crimes Actually)

ಹ್ಯಾಕರ್‌ಗಳು ಇಂಟರ್‌ನೆಟ್ ಬಳಕೆಯೊಂದಿಗೆ ನಿಮ್ಮ ಸೈಟ್‌ನ್ನು ಧ್ವಂಸಗೊಳಿಸಬಹುದು, ಗೌಪ್ಯ ಮಾಹಿತಿಯನ್ನು ವೀಕ್ಷಿಸಬಹುದು, ವ್ಯಾಪಾರ ರಹಸ್ಯಗಳನ್ನು ಅಥವಾ ಬೌದ್ಧಿಕ ಆಸ್ತಿಯನ್ನು ಕದಿಯಬಹುದು. ಇದು ಡೆನಿಯಲ್ ಆಫ್ ಸರ್ವಿಸ್ ಮತ್ತು ವೈರಸ್‌ಗಳ ದಾಳಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ನಿಯಮಿತ ಸಂಚಾರವು ನಿಮ್ಮ ಸೈಟ್ ತಲುಪುವುದನ್ನು ತಡೆಯುತ್ತದೆ. ಸೈಬರ್ ಅಪರಾಧಗಳು ವೈರಸ್‌ಗಳನ್ನು ಹೊರತುಪಡಿಸಿ, ಹೊರಗಿನವರಿಗೆ ಮಾತ್ರ ಸೀಮಿತವಾಗಿಲ್ಲ ಮತ್ತು ಭದ್ರತೆಗೆ ಸಂಬಂಧಿಸಿದ ಸೈಬರ್ ಅಪರಾಧಗಳಿಗೆ, ಸಾಮಾನ್ಯವಾಗಿ ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಂಪನಿಯ ಉದ್ಯೋಗಿಗಳು ತಮ್ಮ ಲಾಭಗಳಿಗಾಗಿ ಸುಲಭವಾಗಿ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಮತ್ತು ಕಂಪನಿಯ ಡಾಟಾ ಸಂಗ್ರಹವನ್ನು ಪಡೆದುಕೊಂಡು ಕಾರಣರಾಗುತ್ತಾರೆ. ಸೈಬರ್ ಅಪರಾಧಗಳು ಕಂಪ್ಯೂಟರ್‌ನ ಬಳಕೆಯಿಂದ ಮಾಡುವ ಅಪರಾಧ ಚಟುವಟಿಕೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ. ಇವು ಮುಂದೆ ಅಪರಾಧಗಳನ್ನು ಶಾಶ್ವತಗೊಳಿಸುತ್ತವೆ. ಅಂದರೆ, ಆರ್ಥಿಕ ಅಪರಾಧಗಳು, ಅಕ್ರಮ ಲೇಖನಗಳ ಮಾರಾಟ, ಅಶ್ಲೀಲತೆ, ಆನ್‌ಲೈನ್ ಜೂಜು, ಬೌದ್ಧಿಕ ಆಸ್ತಿ ಅಪರಾಧ, ಇ-ಮೇಲ್, ವಂಚನೆ, ನಕಲಿ, ಸೈಬರ್ ಮಾನನಷ್ಟ, ಸೈಬರ್ ಹಿಂಬಾಲಿಕೆ, ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ಗೆ ಅನಧಿಕೃತ ಪ್ರವೇಶ, ಇಲೆಕ್ಟ್ರಾನಿಕ್ ರೂಪದಲ್ಲಿರುವ ಮಾಹಿತಿಯ ಕಳುವು, ಇ-ಮೇಲ್ ಬಾಂಬ್ ದಾಳಿ, ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಭೌತಿಕವಾಗಿ ಹಾನಿಗೊಳಿಸುವುದು.

ಸೈಬರ್ ಅಪರಾಧಗಳ ವರ್ಗೀಕರಣ (Classifications Of Cyber Crimes): ಸೈಬರ್ ಅಪರಾಧಗಳು ದಿನದಿಂದ ದಿನಕ್ಕೆ ಹೆಚ್ಚುತ್ತಿವೆ. ವಾಸ್ತವವಾಗಿ ಸೈಬರ್ ಅಪರಾಧಗಳು ಏನೆಂದು ಮತ್ತು ಸಾಂಪ್ರದಾಯಿಕ ಅಪರಾಧ ಯಾವುದು ಎಂದು ಕಂಡುಹಿಡಿಯುವುದು ತುಂಬಾ ಕಷ್ಟ. ಅದ್ದರಿಂದ ಈ ಗೊಂದಲದಿಂದ ಹೊರಬರಲು ಸೈಬರ್ ಅಪರಾಧಗಳನ್ನು ವಿವಿಧ ವರ್ಗಗಳಲ್ಲಿ ಕೆಳಗಿನಂತೆ ವಿಂಗಡಿಸಲಾಗಿದೆ:

ವ್ಯಕ್ತಿಗಳ ವಿರುದ್ಧ ಸೈಬರ್ ಅಪರಾಧಗಳು (Cyber Crimes against Persons) :

ವ್ಯಕ್ತಿಗಳ ವ್ಯಕ್ತಿತ್ವದ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ಕೆಲವು ಅಪರಾಧಗಳನ್ನು ಹೀಗೆ ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು :

ಇ-ಮೇಲ್‌ಗಳ ಮೂಲಕ ಕಿರುಕುಳ : ಇದು ಪತ್ರಗಳನ್ನು, ಫೈಲ್ ಮತ್ತು ಫೋಲ್ಡರ್‌ಗಳ ಲಗತ್ತುಗಳನ್ನು ಕಳುಹಿಸುವ ಮೂಲಕ ನೀಡುವ ಕಿರುಕುಳದ ಸಾಮಾನ್ಯ ವಿಧವಾಗಿದೆ. ಅಂದರೆ ಇ-ಮೇಲ್ ಮೂಲಕ. ಪ್ರಸ್ತುತ ಸಾಮಾಜಿಕ ಜಾಲತಾಣಗಳ ಅಂದರೆ ಫೇಸ್‌ಬುಕ್, ಟ್ವಿಟರ್ ಇತ್ಯಾದಿಗಳ ಬಳಕೆಯಿಂದ ಕಿರುಕುಳವು ಇತ್ತೀಚೆಗೆ ದಿನದಿಂದ ದಿನಕ್ಕೆ ಸಾಮಾನ್ಯವಾಗುತ್ತಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸೈಬರ್ ಸ್ಟಾಕಿಂಗ್ (Cyber-Stalking)

ಇದರ ಅರ್ಥ ದೈಹಿಕ ಬೆದರಿಕೆಯನ್ನು ವ್ಯಕ್ತಪಡಿಸುವುದು ಅಥವಾ ಸೂಚಿಸುವುದು. ಇದು ಕಂಪ್ಯೂಟರ್ ತಂತ್ರಜ್ಞಾನದ ಬಳಕೆಯಿಂದ ಭಯವನ್ನು ಸೃಷ್ಟಿಸುತ್ತದೆ ಉದಾಹರಣೆಗೆ, ಇ-ಮೇಲ್, ಫೋನ್‌ಗಳು, ಟೆಕ್ಸ್ಟ್ ಸಂದೇಶಗಳು, ವೆಬ್‌ಕ್ಯಾಮ್, ವೆಬ್‌ಸೈಟ್‌ಗಳು ಅಥವಾ ವಿಡಿಯೋಗಳು.

ಅಶ್ಲೀಲ ವಸ್ತುಗಳ ಪ್ರಸರಣ (Dissemination of Obscene Material)

ಇದು ಅಸಭ್ಯತೆ/ಅಶ್ಲೀಲತೆ (ಮೂಲತಃ ಮಕ್ಕಳ ಅಶ್ಲೀಲತೆ), ನಿಷೇಧಿತ ವಸ್ತುಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ವೆಬ್‌ಸೈಟ್‌ಗಳನ್ನು ಹೋಸ್ಟ್ ಮಾಡುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಈ ಅಶ್ಲೀಲ ವಿಷಯಗಳು ಹದಿಹರೆಯದವರ ಮನಸ್ಸಿಗೆ ಹಾನಿ ಉಂಟುಮಾಡುತ್ತವೆ ಮತ್ತು ಅವರ ಮನಸ್ಸನ್ನು ಕೆಡಿಸುತ್ತವೆ.

ಮಾನಹಾನಿ (Defamation)

ಇದು ಯಾವುದೇ ವ್ಯಕ್ತಿಯನ್ನು ಅವನ ಘನತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಉದ್ದೇಶದಿಂದ ಆರೋಪಿಸುವ ಕ್ರಿಯೆಯಾಗಿದೆ. ಇದನ್ನು ಅವನ ಮೇಲ್ ಖಾತೆಯನ್ನು ಹ್ಯಾಕ್ ಮಾಡುವ ಮೂಲಕ ಮತ್ತು ಅಸಭ್ಯ ಭಾಷೆಯನ್ನು ಬಳಸಿ ಕೆಲವು ಮೇಲ್‌ಗಳನ್ನು ಕಳುಹಿಸುವ ಮೂಲಕ ಮಾಡಲಾಗುತ್ತದೆ.

ಹ್ಯಾಕಿಂಗ್ (Hacking)

ಇದು ಒಂದು ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಅನಧಿಕೃತವಾಗಿ ನಿಯಂತ್ರಿಸುವುದು/ಪ್ರವೇಶಿಸುವುದು ಆಗಿದೆ ಮತ್ತು ಹ್ಯಾಕಿಂಗ್ ಕ್ರಿಯೆಯು ಡಾಟಾವನ್ನು ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ನಾಶಪಡಿಸುತ್ತದೆ. ಹ್ಯಾಕರ್‌ಗಳು ಸಾಮಾನ್ಯವಾಗಿ ದೂರಸಂಪರ್ಕವನ್ನು ಮತ್ತು ಮೊಬೈಲ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಹ್ಯಾಕ್ ಮಾಡುತ್ತಾರೆ.

ಕ್ರ್ಯಾಕಿಂಗ್ (Cracking)

ಇದು ಇಲ್ಲಿಯವರೆಗೆ ತಿಳಿದಿರುವ ಅತ್ಯಂತ ಗಂಭೀರವಾದ ಸೈಬರ್ ಅಪರಾಧಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದು ಅಪರಿಚಿತರು ನಿಮಗೆ ಅರಿವಿಲ್ಲದಂತೆ ಮತ್ತು ಸಮ್ಮತಿ ಇಲ್ಲದೇ ನಿಮ್ಮ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಪ್ರವೇಶಿಸಿದ್ದಾರೆ ಮತ್ತು ಅಮೂಲ್ಯವಾದ ಗೌಪ್ಯ ಡಾಟಾ ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಹಾಳು ಮಾಡಿದ್ದಾರೆ ಎಂದು ತಿಳಿಯುವ ಭಯಾನಕವಾಗಿರುತ್ತದೆ.

ಇ-ಮೇಲ್ ವಂಚನೆ (E-Mail Spoofing)

ವಂಚನೆಯ ಇ-ಮೇಲ್ ಎಂದರೆ, ತನ್ನ ಮೂಲವನ್ನು ತಪ್ಪಾಗಿ ಪ್ರತಿನಿಧಿಸುವ ಮೇಲ್ ಆಗಿದೆ. ಇದು ತಾನು ವಾಸ್ತವವಾಗಿ ಇರುವ ಮೂಲಕ್ಕಿಂತ ವಿಭಿನ್ನವಾಗಿ ತೋರಿಸಿಕೊಳ್ಳುತ್ತದೆ.

SMS ವಂಚನೆ (SMS Spoofing)

ಸ್ಪೂಫಿಂಗ್ ಎಂದರೆ ಸ್ಪ್ಯಾಮ್ ಮೂಲಕ ನಿರ್ಬಂಧಿಸುವುದು. ಅಂದರೆ ಅನಗತ್ಯ ಆಹ್ವಾನಿಸದ ಸಂದೇಶಗಳು. ಇಲ್ಲಿ ಒಬ್ಬ ಅಪರಾಧಿ ಮೊಬೈಲ್ ಫೋನ್ ನಂಬರ ಮೂಲಕ ಇನ್ನೊಬ್ಬರ ಗುರುತನ್ನು ಕದಿಯುತ್ತಾನೆ ಮತ್ತು ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ SMS ಕಳುಹಿಸುತ್ತಾನೆ ಮತ್ತು ಸ್ವೀಕರಿಸುವವರು ಆ ಮೊಬೈಲ್ ನಂಬರ್‌ನಿಂದ SMS ನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತಾರೆ. ಇದು ಯಾವುದೇ ವ್ಯಕ್ತಿಯ ವಿರುದ್ಧ ಗಂಭೀರವಾದ ಸೈಬರ್ ಅಪರಾಧವಾಗಿದೆ.

ಕಾರ್ಡಿಂಗ್ (Carding)

ಇದರರ್ಥ ಸುಳ್ಳು ATM ಕಾರ್ಡ್‌ಗಳು . ಅಂದರೆ ಅಪರಾಧಿಗಳಿಂದ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಡೆಬಿಟ್ ಮತ್ತು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್‌ಗಳು. ಇದನ್ನು ಬಳಸಿ ಅಪರಾಧಿಗಳು ತಮ್ಮ ಹಣಕಾಸಿನ ಲಾಭಗಳಿಗಾಗಿ ಬಲಿ ಪಶುವಿನ ಬ್ಯಾಂಕ್ ಖಾತೆಯಿಂದ ಹಣವನ್ನು ಹಿಂತೆಗೆದುಕೊಳ್ಳುತ್ತಾರೆ. ಈ ರೀತಿಯ ಸೈಬರ್ ಅಪರಾಧಗಳಲ್ಲಿ ಯಾವಾಗಲೂ ATM ಕಾರ್ಡ್‌ಗಳ ಅನಧಿಕೃತ ಬಳಕೆಯಾಗುತ್ತದೆ.

ವಂಚನೆ ಮತ್ತು ಮೋಸ (Cheating & Fraud)

ಇದರರ್ಥ ಸೈಬರ್ ಅಪರಾಧ ಮಾಡುವ ವ್ಯಕ್ತಿ ಅದನ್ನು ತಪ್ಪಿತಸ್ಥ ಮನಸ್ಸಿನಿಂದ ಮಾಡಿರುವುದರಿಂದ ಮೋಸ ಮತ್ತು ವಂಚನೆಗೆ ಕಾರಣವಾಗುತ್ತದೆ. ಸೈಬರ್ ಅಪರಾಧವು ಪಾಸ್‌ವರ್ಡ್ ಮತ್ತು ಡಾಟಾ ಸಂಗ್ರಹವನ್ನು ಕದಿಯುವುದನ್ನು ಒಳಗೊಂಡಿದೆ.

ಮಕ್ಕಳ ಪೋರ್ನೊಗ್ರಫಿ (Child Pornography)

ಇದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸಿ ಅಪ್ರಾಪ್ತ ವಯಸ್ಸಿನ ಮಕ್ಕಳನ್ನು ಲೈಂಗಿಕವಾಗಿ ದುರ್ಬಳಕೆ ಮಾಡುವ ವಸ್ತುಗಳನ್ನು ರಚಿಸುವುದು, ವಿತರಿಸುವುದು ಅಥವಾ ಪಡೆದುಕೊಳ್ಳುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಬೆದರಿಕೆಯಿಂದ ಹಲ್ಲೆ ಮಾಡುವುದು(Assault by Threat)

ಬೆದರಿಕೆಯಿಂದ ಹಲ್ಲೆ ಮಾಡುವುದು ಎಂದರೆ ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸಿ ಒಬ್ಬ ವ್ಯಕ್ತಿಯನ್ನು ಅವರ ಜೀವನ ಅಥವಾ ಅವರ ಕುಟುಂಬದ ಜೀವನವನ್ನು ಬೆದರಿಸುವುದನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಎಂದರೆ ಇ-ಮೇಲ್, ವಿಡಿಯೊಗಳು, ಫೋನ್‌ಗಳು.

ವ್ಯಕ್ತಿಗಳ ಆಸ್ತಿ ವಿರುದ್ಧ ಅಪರಾಧಗಳು (Crimes Against Persons Property)

ಅಂತರಾಷ್ಟ್ರೀಯ ವ್ಯಾಪಾರದಲ್ಲಿ ತ್ವರಿತ ಬೆಳವಣಿಗೆ ಇರುವುದರಿಂದ ಇಲ್ಲಿ ವ್ಯಾಪಾರಗಳು ಮತ್ತು ಗ್ರಾಹಕರು ಸಾಂಪ್ರದಾಯಿಕ ಪೇಪರ್ ದಾಖಲಾತಿಗಳಿಗಿಂತ ಇಲೆಕ್ಟ್ರಾನಿಕ್ ರೂಪದಲ್ಲಿ ಮಾಹಿತಿಯನ್ನು ತಯಾರಿಸಲು, ರವಾನಿಸಲು ಮತ್ತು ಸಂಗ್ರಹಿಸಲು ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಹೆಚ್ಚು ಬಳಸುತ್ತಾರೆ.

NOT TO BE REPUBLISHED © NIMI

ವ್ಯಕ್ತಿಗಳ ಆಸ್ತಿಯ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ಕೆಲವು ಅಪರಾಧಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ :

ಬೌದ್ಧಿಕ ಆಸ್ತಿ ಅಪರಾಧಗಳು (Intellectual Property Crimes)

ಬೌದ್ಧಿಕ ಆಸ್ತಿಯು ಹಕ್ಕುಗಳ ಬಂಡಲ್‌ನ್ನು ಒಳಗೊಂಡಿದೆ. ಯಾವುದೇ ಕಾನೂನುಬಾಹಿರ ಕ್ರಿಯೆಯಿಂದ ಮಾಲೀಕರನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಅಥವಾ ಭಾಗಶಃ ವಂಚಿಸುವುದು ಅಪರಾಧವಾಗಿದೆ. IPR ಉಲ್ಲಂಘನೆಯ ಸಾಮಾನ್ಯ ರೂಪಗಳು ಸಾಫ್ಟವೇರ್ ಪೈರಸಿ, ಕಾಪಿರೈಟ್‌ನ ಅತಿಕ್ರಮಣ, ಟ್ರೇಡ್‌ಮಾರ್ಕ್, ಪೇಟೆಂಟ್‌ಗಳು, ವಿನ್ಯಾಸಗಳು, ಮತ್ತು ಸೇವಾ ಗುರುತು ಉಲ್ಲಂಘನೆ, ಕಂಪ್ಯೂಟರ್ ಸೋರ್ಸ್ ಕೋಡ್‌ನ ಕಳ್ಳತನ ಇತ್ಯಾದಿಗಳು.

ಸೈಬರ್ ಸ್ಕ್ವಾಟಿಂಗ್ (Cyber Squatting)

ಇದರರ್ಥ ಒಂದೇ ಡೊಮೇನ್ ಹೆಸರಿಗಾಗಿ ಇಬ್ಬರು ವ್ಯಕ್ತಿಗಳು ಹಕ್ಕು ಸಾಧಿಸುತ್ತಾರೆ. ಅವರು ಮೊದಲು ತಾವು ಹೆಸರನ್ನು ನೋಂದಾಯಿಸಿದ್ದಾರೆ ಎಂದು ಹೇಳಿಕೊಳ್ಳುವ ಮೂಲಕ ಇತರರಿಗಿಂತ ಮೊದಲು ಅದನ್ನು ಬಳಸುತ್ತಾರೆ ಅಥವಾ ಹಿಂದಿನದನ್ನು ಹೋಲುವಂತಹದನ್ನು ಬಳಸುವುದು.

ಸೈಬರ್ ವಿಧ್ವಂಸಕತೆ (Cyber Vandalism)

ವಿಧ್ವಂಸಕತೆ ಎಂದರೆ ಉದ್ದೇಶಪೂರ್ವಕವಾಗಿ ಇನ್ನೊಬ್ಬರ ಆಸ್ತಿಯನ್ನು ನಾಶಪಡಿಸುವುದು ಅಥವಾ ಹಾನಿ ಮಾಡುವುದು. ಹೀಗಾಗಿ ಸೈಬರ್ ವಿಧ್ವಂಸಕತೆ ಎಂದರೆ ನೆಟ್‌ವರ್ಕ್ ಸೇವೆಯನ್ನು ನಿಲ್ಲಿಸಿದಾಗ ಡಾಟಾವನ್ನು ನಾಶಪಡಿಸುವುದು ಅಥವಾ ಹಾನಿಗೊಳಿಸುವುದು. ಇದು ತನ್ನ ವ್ಯಾಪ್ತಿಯಲ್ಲಿರುವ ಯಾವುದೇ ವ್ಯಕ್ತಿಯ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಯಾವುದೇ ರೀತಿಯ ಭೌತಿಕ ಹಾನಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಈ ಕೃತ್ಯಗಳು, ಕಂಪ್ಯೂಟರ್‌ನ, ಕಂಪ್ಯೂಟರ್‌ನ ಕೆಲವು ಭಾಗಗಳ ಅಥವಾ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಜೋಡಿಸಿದ ಬಾಹ್ಯ ಭಾಗದ, ಕಳ್ಳತನವನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ.

ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಹ್ಯಾಕ್ ಮಾಡುವುದು (Hacking Computer System)

ಹ್ಯಾಕ್ಟಿವಿಸಮ್ ಪ್ರಸಿದ್ಧ ಟ್ವಿಟರ್, ಬ್ಲಾಗಿಂಗ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳನ್ನು ಕಂಪ್ಯೂಟರ್‌ನ ಅನಧಿಕೃತ ಪ್ರವೇಶ /ನಿಯಂತ್ರಣದ ಮೂಲಕ ದಾಳಿ ಮಾಡುತ್ತದೆ. ಹ್ಯಾಕಿಂಗ್ ಚಟುವಟಿಕೆಯ ಕಾರಣದಿಂದ ಡಾಟಾ ಮತ್ತು ಕಂಪ್ಯೂಟರ್‌ನ ನಷ್ಟವಾಗುತ್ತದೆ. ಸಂಶೋಧನೆಗಳು ವಿಶೇಷವಾಗಿ ಈ ದಾಳಿಗಳು ಮುಖ್ಯವಾಗಿ ಆರ್ಥಿಕ ಲಾಭಕ್ಕಾಗಿ ಅಷ್ಟೇ ಅಲ್ಲದೇ ನಿರ್ದಿಷ್ಟ ವ್ಯಕ್ತಿ ಅಥವಾ ಕಂಪನಿಯ ಖ್ಯಾತಿಯನ್ನು ಕಡಿಮೆ ಮಾಡುವ ಉದ್ದೇಶವನ್ನು ಸೂಚಿಸುತ್ತವೆ.

ವೈರಸ್‌ನ್ನು ಹರಡುವುದು (Transmitting Virus)

ವೈರಸ್‌ಗಳು ತಮ್ಮನ್ನು ತಾವೇ ಕಂಪ್ಯೂಟರ್‌ಗೆ ಅಥವಾ ಫೈಲ್‌ಗೆ ಲಗತ್ತಿಸುವ ಪ್ರೋಗ್ರಾಂಗಳಾಗಿವೆ ಮತ್ತು ನಂತರ ತಮ್ಮನ್ನು ನೆಟ್‌ವರ್ಕ್‌ನಲ್ಲಿರುವ ಇತರ ಫೈಲ್‌ಗಳಿಗೆ ಮತ್ತು ಇತರ ಕಂಪ್ಯೂಟರ್‌ಗಳಿಗೆ ಪ್ರಸಾರ ಮಾಡುತ್ತವೆ. ಸಾಮಾನ್ಯವಾಗಿ ಅವು ಕಂಪ್ಯೂಟರ್‌ನಲ್ಲಿರುವ ಡಾಟಾದ ಮೇಲೆ ಅದನ್ನು ಬದಲಾಯಿಸುವ ಮೂಲಕ ಅಥವಾ ಡಿಲೀಟ್ ಮಾಡುವ ಮೂಲಕ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ವರ್ಮ್ ದಾಳಿಗಳು ವ್ಯಕ್ತಿಗಳ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವಲ್ಲಿ ಪ್ರಮುಖ ಪಾತ್ರ ವಹಿಸುತ್ತವೆ.

ಸೈಬರ್ ಟ್ರೆಸ್‌ಪಾಸ್ (Cyber Trespass)

ಇದು ಮಾಲೀಕರ ಸರಿಯಾದ ಅನುಮತಿ ಇಲ್ಲದೇ ಬೇರೆಯವರ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಪ್ರವೇಶಿಸುವುದು ಎಂದರ್ಥ. ಮತ್ತು ವೈರ್‌ಲೆಸ್ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ಬಳಸಿ ಡಾಟಾ ಅಥವಾ ಸಿಸ್ಟಮ್‌ನ್ನು ತೊಂದರೆಗೊಳಿಸುವುದು, ಬದಲಾಯಿಸುವುದು, ದುರುಪಯೋಗಪಡಿಸಿಕೊಳ್ಳುವುದು, ಅಥವಾ ಹಾನಿಗೊಳಿಸುವುದು ಮಾಡುವುದಿಲ್ಲ.

ಇಂಟರ್‌ನೆಟ್ ಟೈಮ್ ಥೆಫ್ಟ್ (Internet Time Thefts)

ಇಂಟರ್‌ನೆಟ್ ಟೈಮ್ ಥೆಫ್ಟ್ ಹ್ಯಾಕಿಂಗ್ ಅಡಿಯಲ್ಲಿ ಬರುತ್ತದೆ. ಇದು ಇನ್ನೊಬ್ಬ ವ್ಯಕ್ತಿಯಿಂದ ಪಾವತಿಸಿದ ಇಂಟರ್‌ನೆಟ್‌ನ ಗಂಟೆಗಳ ಸಮಯವು ಅನಧಿಕೃತ ವ್ಯಕ್ತಿಯಿಂದ ಬಳಸಲ್ಪಡುತ್ತದೆ. ಒಬ್ಬ ವ್ಯಕ್ತಿ ಹ್ಯಾಕ್ ಮಾಡುವ ಮೂಲಕ ಅಥವಾ ಕಾನೂನುಬಾಹಿರ ವಿಧಾನಗಳ ಮೂಲಕ ಬೇರೆಯವರಿಗೆ ತಿಳಿಯದಂತೆ ಅವರ ' ISP ಯೂಸರ್ ಐಡಿ ಮತ್ತು ಪಸ್‌ವರ್ಡ್‌ನ್ನು ಬಳಸಿ ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತಾನೆ. ಇಂಟರ್‌ನೆಟ್ ಸಮಯವನ್ನು ಅಪರೂಪವಾಗಿ ಬಳಸಿದರೂ ಕೂಡ ಆಗಾಗ್ಗೆ ರೀಚಾರ್ಜ್ ಮಾಡಬೇಕಾದಾಗ ನೀವು ಟೈಮ್ ಥೆಫ್ಟ್‌ನ್ನು ಗುರುತಿಸಬಹುದು.

ಸರ್ಕಾರದ ವಿರುದ್ಧ ಸೈಬರ್ ಅಪರಾಧಗಳು (Cybercrimes Against Government)

ಇಂಟರ್‌ನೆಟ್ ಸೌಲಭ್ಯಗಳನ್ನು ಬಳಸಿ ಅಂತರಾಷ್ಟ್ರೀಯ ಸರ್ಕಾರಗಳಿಗೆ ಬೆದರಿಕೆ ಹಾಕುವ ಉದ್ದೇಶವನ್ನು ಹೊಂದಿರುವ ವ್ಯಕ್ತಿಗಳ ಗುಂಪು ಮಾಡುವ ಕೆಲವು ಅಪರಾಧಗಳಿವೆ. ಅವು ಕೆಳಗಿನಂತಿವೆ :

ಸೈಬರ್ ಭಯೋತ್ಪಾದನೆ (Cyber Terrorism)

ಸೈಬರ್ ಭಯೋತ್ಪಾದನೆಯು ದೇಶೀಯ ಮತ್ತು ಜಾಗತಿಕ ಕಾಳಜಿಯ ಪ್ರಮುಖ ಜ್ವಲಂತ ಸಮಸ್ಯೆಯಾಗಿದೆ. ಇಂಟರ್‌ನೆಟ್ ಮೇಲೆ ಈ ಭಯೋತ್ಪಾದಕ ದಾಳಿಗಳ ಸಾಮಾನ್ಯ ರೂಪವು, ಡೇನಿಯಲ್ ಆಫ್ ಸರ್ವೀಸ್ ದಾಳಿಗಳು, ದ್ವೇಷದ ವೆಬ್‌ಸೈಟ್‌ಗಳು, ಮತ್ತು ದ್ವೇಷದ ಇ-ಮೇಲ್‌ಗಳು, ಸೂಕ್ಷ್ಮ ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ಗಳ ಮೇಲೆ ದಾಳಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಸೈಬರ್ ಭಯೋತ್ಪಾದನೆ ಚಟುವಟಿಕೆಗಳು ರಾಷ್ಟ್ರದ ಸಾರ್ವಭೌಮತ್ವ ಮತ್ತು ಸಮಗ್ರತೆಗೆ ಅಪಾಯವನ್ನುಂಟು ಮಾಡುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

ಸೈಬರ್ ವಾರ್‌ಫೇರ್ (Cyber Warfare)

ಇದು ವಿಧ್ವಂಸಕ ಮತ್ತು ಬೇಹುಗಾರಿಕೆ ನಡೆಸಲು ರಾಜಕೀಯ ಪ್ರೇರಿತ ಹ್ಯಾಕಿಂಗ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಇದು ಮಾಹಿತಿ ಯುದ್ಧದ ಒಂದು ರೂಪವಾಗಿದೆ. ಕೆಲವೊಮ್ಮೆ ಸಾಂಪ್ರದಾಯಿಕ ಯುದ್ಧಕ್ಕೆ ಸದೃಶವಾಗಿ ಕಂಡುಬರುತ್ತದೆ. ಆದಾಗ್ಯೂ ಈ ಸಾದೃಶವು ಅದರ ನಿಖರತೆಗೆ ಮತ್ತು ಅದರ ರಾಜಕೀಯ ಪ್ರೇರಣೆಗೆ ವಿವಾದಾತ್ಮಕವಾಗಿದೆ.

ಪೈರೇಟ್ ಆದ ಸಾಫ್ಟ್‌ವೇರ್ ವಿತರಣೆ (Distribution of pirated software)

ಇದರರ್ಥ ಒಂದು ಕಂಪ್ಯೂಟರ್‌ನಿಂದ ಇನ್ನೊಂದಕ್ಕೆ ಪೈರೇಟ್ ಆದ ಸಾಫ್ಟ್‌ವೇರನ್ನು ವಿತರಿಸುವುದು. ಇದರ ಉದ್ದೇಶ ಸರ್ಕಾರದ ಡಾಟಾವನ್ನು ಮತ್ತು ಅಧಿಕೃತ ದಾಖಲೆಗಳನ್ನು ನಾಶ ಮಾಡುವುದಾಗಿದೆ.

ಅನಧಿಕೃತ ಮಾಹಿತಿಯ ಸ್ವಾಧೀನ (Possession of Unauthorized Information)

ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಭಯೋತ್ಪಾದಕರು ಯಾವುದೇ ಮಾಹಿತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು ತುಂಬಾ ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಆ ಮಾಹಿತಿಯನ್ನು ರಾಜಕೀಯ, ಧಾರ್ಮಿಕ, ಸಾಮಾಜಿಕ, ಸೈದ್ಧಾಂತಿಕ ಉದ್ದೇಶಗಳಿಗಾಗಿ ಬಳಸುತ್ತಾರೆ.

ಸಮಾಜದ ವಿರುದ್ಧ ಸೈಬರ್ ಅಪರಾಧಗಳು (Cybercrimes Against Society at large)

ಸೈಬರ್ ಸ್ಪೇಸ್‌ಗೆ ಹಾನಿಯನ್ನುಂಟು ಮಾಡುವ ಉದ್ದೇಶದಿಂದ ಮಾಡಿದ ಕಾನೂನುಬಾಹಿರ ಕೃತ್ಯಗಳು ಹೆಚ್ಚು ಜನರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ಈ ಅಪರಾಧಗಳು ಕೆಳಗಿನಂತಿವೆ :

ಮಕ್ಕಳ ಪೊರ್ನೋಗ್ರಾಫಿ (Child Pornography)

ಇದು ಕಂಪ್ಯೂಟರ್ ನೆಟ್‌ವರ್ಕ್‌ನ್ನು ಬಳಸಿ ಅಪ್ರಾಪ್ತ ವಯಸ್ಸಿನ ಮಕ್ಕಳನ್ನು ಲೈಂಗಿಕವಾಗಿ ದುರ್ಬಳಕೆ ಮಾಡುವ ವಸ್ತುಗಳನ್ನು ರಚಿಸುವುದು, ವಿತರಿಸುವುದು ಅಥವಾ ಪಡೆದುಕೊಳ್ಳುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಅಸಭ್ಯವನ್ನು ತೋರಿಸುವುದು ಮತ್ತು ಅಶ್ಲೀಲತೆಗೆ ಸಂಬಂಧಿಸಿದ ಚಟುವಟಿಕೆಗಳನ್ನು ಸಹ ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಸೈಬರ್ ಟ್ರಾಫಿಕಿಂಗ್ (Cyber Trafficking)

ಇದು ಮಾದಕ ವಸ್ತುಗಳ, ಮಾನವರ, ಶಸ್ತ್ರಾಸ್ತ್ರಗಳ ಇತ್ಯಾದಿಗಳ ಕಳ್ಳಸಾಗಣೆಯಾಗಿರಬಹುದು. ಇದು ಬಹಳ ಜನರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತದೆ. ಸೈಬರ್ ಸ್ಪೇಸ್‌ನಲ್ಲಿ ಕಳ್ಳ ಸಾಗಣೆ ಗಂಭೀರ ಅಪರಾಧವಾಗಿದೆ.

ಆನ್‌ಲೈನ್ ಜೂಜು (Online Gambling)

ಆನ್‌ಲೈನ್ ಮೋಸ ಮತ್ತು ವಂಚನೆಯು ಸೈಬರ್ ಸ್ಪೇಸ್‌ನಲ್ಲಿ ಇವತ್ತು ಹೆಚ್ಚುತ್ತಿರುವ ಲಾಭದಾಯಕ ವ್ಯಾಪಾರಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಬೆಳಕಿಗೆ ಬಂದಿರುವ ಕೆಲವು ಪ್ರಕರಣಗಳು ಯಾವವೆಂದರೆ - ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಅಪರಾಧಗಳು, ಒಪ್ಪಂದದ ಅಪರಾಧಗಳು, ಉದ್ಯೋಗಗಳನ್ನು ಒದಗಿಸುವ ಅಪರಾಧಗಳು ಇತ್ಯಾದಿ.

ಆರ್ಥಿಕ ಅಪರಾಧಗಳು (Financial Crimes)

ನೆಟ್‌ವರ್ಕಿಂಗ್ ಸೈಟಗಳು ಮತ್ತು ಫೋನ್ ನೆಟ್‌ವರ್ಕಿಂಗ್‌ನ ಬಳಕೆದಾರರಲ್ಲಿ ತ್ವರಿತ ಬೆಳವಣಿಗೆ ಇರುವುದರಿಂದ ಈ ರೀತಿಯ ಅಪರಾಧಗಳು ಸಾಮಾನ್ಯವಾಗಿವೆ. ಇಲ್ಲಿ ಅಪರಾಧಿಯು ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಬೋಗಸ್ ಮೇಲ್‌ಗಳನ್ನು ಅಥವಾ ಸಂದೇಶಗಳನ್ನು ಕಳುಹಿಸುವ ಮೂಲಕ ದಾಳಿ ಮಾಡಲು ಪ್ರಯತ್ನಿಸುತ್ತಾನೆ. ಉದಾಹರಣೆಗೆ, ಅಕ್ರಮವಾಗಿ ಪಾಸ್‌ವರ್ಡ್ ಪಡೆದು ಕ್ರೆಡಿಟ್ ಕಾರ್ಡ್ ಬಳಸುವುದು.

ಫೋರ್ಜರಿ (Forgery)

ಇದು ಬೆದರಿಕೆಯ ಮೇಲ್‌ಗಳನ್ನು ಕಳುಹಿಸುವ ಮೂಲಕ ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯ ಜನರನ್ನು ವಂಚಿಸುವುದು ಎಂದರ್ಥ. ಆನ್‌ಲೈನ್ ವ್ಯಾಪಾರದ ವಹಿವಾಟುಗಳು ಇವತ್ತಿನ ಜೀವನ ಶೈಲಿಯ ಅಭ್ಯಾಸದ ಅಗತ್ಯವಾಗಿವೆ.

ಯಾರಿಗೆ ಪರಿಣಾಮ ಬೀರುತ್ತದೆ (Affects To Whom)

ಸೈಬರ್ ಅಪರಾಧಗಳು ಯಾವಾಗಲೂ ಯಾವುದೇ ಗಾತ್ರದ ಕಂಪನಿಯ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುತ್ತವೆ. ಏಕೆಂದರೆ ಬಹುತೇಕ ಎಲ್ಲಾ ಕಂಪನಿಗಳು ಆನ್‌ಲೈನ್ ಉಪಸ್ಥಿತಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ ಮತ್ತು ತಂತ್ರಜ್ಞಾನದಲ್ಲಿನ ತ್ವರಿತ ಲಾಭವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತವೆ. ಆದರೆ ಅದರ ಭದ್ರತಾ ಅಪಾಯಗಳ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ಗಮನವನ್ನು ನೀಡಬೇಕು. ಆಧುನಿಕ ಸೈಬರ್ ಪ್ರಪಂಚದಲ್ಲಿ, ಸೈಬರ್ ಅಪರಾಧಗಳು ಪ್ರಮುಖ ಸಮಸ್ಯೆಯಾಗಿವೆ. ಇವು ವ್ಯಕ್ತಿ ಮತ್ತು ಸಮಾಜದ ಮೇಲೆ ತುಂಬಾ ಪ್ರಭಾವ ಬೀರುತ್ತಿವೆ.

ಸೈಬರ್ ಕಾನೂನಿನ ಅವಶ್ಯಕತೆ (Need of Cyber Law)

ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನವು ಪ್ರಪಂಚದಾದ್ಯಂತ ವ್ಯಾಪಿಸಿದೆ. ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಪ್ರತಿಯೊಂದು ಕ್ಷೇತ್ರದಲ್ಲಿ ಬಳಸಲಾಗುತ್ತಿದೆ. ಸೈಬರ್ ಸ್ಪೇಸ್ ಎಲ್ಲರಿಗೂ ಆರ್ಥಿಕ ಬೆಳವಣಿಗೆ ಮತ್ತು ಮಾನವ ಅಭಿವೃದ್ಧಿಗೆ ಸಾಮಾನ ಅವಕಾಶಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಸೈಬರ್ ಸ್ಪೇಸ್‌ನ ಬಳಕೆದಾರರು ಹೆಚ್ಚು ವೈವಿಧ್ಯಮಯವಾಗಿ ಬೆಳೆಯುತ್ತಾರೆ ಮತ್ತು ಆನ್‌ಲೈನ್ ಪರಸ್ಪರ ಕ್ರಿಯೆಗಳ ವ್ಯಾಪ್ತಿ ವಿಸ್ತರಿಸುತ್ತದೆ. ಇದರಿಂದ ಸೈಬರ್ ಅಪರಾಧಗಳಲ್ಲಿಯೂ ವಿಸ್ತರಣೆಯಾಗಿದೆ. ಉದಾಹರಣೆಗೆ, ಆನ್‌ಲೈನ್ ಒಪ್ಪಂದಗಳ ಉಲ್ಲಂಘನೆ, ಆನ್‌ಲೈನ್ ವೈಯಕ್ತಿಕ ಅಪರಾಧಗಳು ಮತ್ತು ಇತರ ಅಪರಾಧಗಳು ಇತ್ಯಾದಿ.

© NIMI NOT TO BE REPUBLISHED

ಈ ಪರಿಣಾಮಗಳಿಂದಾಗಿ ಸೈಬರ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ಅಪರಾಧ ಚಟುವಟಿಕೆಗಳನ್ನು ನಿಯಂತ್ರಿಸಲು ಸೈಬರ್ ಸ್ಪೇಸ್ ಪ್ರಾಧಿಕಾರದಿಂದ ಕಠಿಣ ಕಾನೂನನ್ನು ಅಳವಡಿಸಿಕೊಳ್ಳುವ ಅವಶ್ಯಕತೆ ಇದೆ, ಮತ್ತು ಸೈಬರ್ ಅಪರಾಧದ ಬಲಿಪಶುವಿಗೆ ನ್ಯಾಯವನ್ನು ನೀಡಲು ಉತ್ತಮ ಆಡಳಿತವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಆಧುನಿಕ ಸೈಬರ್ ತಂತ್ರಜ್ಞಾನ ಜಗತ್ತಿನಲ್ಲಿ ಸೈಬರ್ ಅಪರಾಧಗಳನ್ನು ನಿಯಂತ್ರಿಸುವುದು ಅತಿ ಆವಶ್ಯವಾಗಿದೆ ಮತ್ತು ಅತಿ ಮುಖ್ಯವಾಗಿ ಸೈಬರ್ ಭಯೋತ್ಪಾದನೆ ಮತ್ತು ಹ್ಯಾಕರ್‌ಗಳ ವಿಚಾರದಲ್ಲಿ ಸೈಬರ್ ಕಾನೂನುಗಳನ್ನು ಕಠಿಣಗೊಳಿಸಬೇಕು.

ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ ಹಾನಿಗಾಗಿ ದಂಡ(Penalty For Damage To Computer System)

ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಕಾಯಿದೆ 2000 ರ Section : 43 ಪ್ರಕಾರ ಯಾರಾದರೂ ಒಂದು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಅದರ ಮಾಲೀಕನ ಅನುಮತಿಯಿಲ್ಲದೇ ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನಲ್ಲಿರುವ ಸಂಪೂರ್ಣ ಡಾಟಾವನ್ನು ಹಾನಿಪಡಿಸುವ ಉದ್ದೇಶದಿಂದ ಯಾವುದೇ ಕಂಪ್ಯೂಟರ್‌ನ್ನು ನಾಶಪಡಿಸುವುದು, ಡಿಲೀಟ್ ಮಾಡುವುದು, ಬದಲಾಯಿಸುವುದು, ಮತ್ತು ಅಡ್ಡಿಪಡಿಸುವುದು ಅಥವಾ ಅಡ್ಡಿಪಡಿಸುವಿಕೆಗೆ ಕಾರಣರಾಗುವುದು, ಮಾಡಿದರೆ ಪ್ರಭಾವಿತನಾದ ವ್ಯಕ್ತಿಗೆ ಇದರ ಪರಿಹಾರವಾಗಿ 1 ಕೋಟಿಯವರೆಗೆ ದಂಡವನ್ನು ಪಾವತಿಸಬೇಕಾಗುತ್ತದೆ. ಮಾಹಿತಿ ತಂತಜ್ಞಾನ ಕಾಯಿದೆ 2000 ದ Section : 43A ಪ್ರಕಾರ ಕೇಂದ್ರ ಸರ್ಕಾರದಿಂದ ಒದಗಿಸಲಾದ ಕಾರ್ಪೋರೇಟ್ ಸಂಸ್ಥೆ ಒಬ್ಬ ವ್ಯಕ್ತಿಯ ಡಾಟಾವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ ಮತ್ತು ರಕ್ಷಿಸುತ್ತದೆ. ಡಾಟಾ ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ರಕ್ಷಿಸುವಲ್ಲಿ ಕಾರ್ಪೋರೇಟ್ ಸಂಸ್ಥೆಯಿಂದ ಯಾವುದೇ ನಿರ್ಲಕ್ಷ್ಯದ ನಡೆಯಿದ್ದಲ್ಲಿ ಅಥವಾ ವೈಫಲ್ಯವಿದ್ದಲ್ಲಿ, ಅದರಿಂದ ಪ್ರಭಾವಿತ ವ್ಯಕ್ತಿಗೆ ಕಾರ್ಪೋರೇಟ್ ಸಂಸ್ಥೆಯು ಪರಿಹಾರವನ್ನು ಪಾವತಿಸಲು ಜವಾಬ್ದಾರನಾಗಿರುತ್ತದೆ. Section 66 ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಹ್ಯಾಕ್ ಮಾಡುವದಕ್ಕೆ ಸಂಬಂಧಿಸಿದೆ ಮತ್ತು ಈ ಅಪರಾಧಕ್ಕೆ 3 ವರ್ಷಗಳವರೆಗೆ ಜೈಲು ಶಿಕ್ಷೆ ಅಥವಾ ದಂಡವನ್ನು ಹೊಂದಿದೆ. ಇದನ್ನು 2 ವರ್ಷಗಳವೆರೆಗೆ ವಿಸ್ತರಿಸಬಹುದು ಅಥವಾ ಎರಡೂ ಆಗಿರಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಕ್ಲೌಡ್ ಸೇವೆಗಳೊಂದಿಗೆ ಕೆಲಸ ಮಾಡುವುದು (Working with Cloud Services)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಪರಿಚಯ**
- **ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ವಿಧಗಳು**
- **ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಅನುಕೂಲಗಳು ಮತ್ತು ಅನಾನುಕೂಲಗಳು.**

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಪರಿಚಯ (Introduction to cloud comping)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ : ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಯ ಲಾಭಗಳು, ಅನಾನುಕೂಲಗಳು, ಮತ್ತು ವಿಧಗಳು.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಇಂದಿನ ವ್ಯಾಪಾರಗಳಲ್ಲಿ ಅತ್ಯಂತ ಜನಪ್ರಿಯ ಶೋಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದು ಸಂಸ್ಥೆಗಳು ಮಾಹಿತಿಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಪಡೆದುಕೊಳ್ಳುವ ಮತ್ತು ಹಂಚಿಕೊಳ್ಳುವ ಮಾರ್ಗವನ್ನು ಹಾಗೂ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಸಹಯೋಗ ಹೊಂದುವ ಮತ್ತು ನಿರ್ವಹಿಸುವ ಮಾರ್ಗವನ್ನು ಬದಲಾಯಿಸಿದೆ. ಇಂಟರ್‌ನೆಟ್‌ನ ಆಗಮನದೊಂದಿಗೆ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ವ್ಯವಹಾರ ನಡೆಸುವ ಹೊಸ ದಾರಿಗಳನ್ನು ಒದಗಿಸಿದೆ. ಇದು ಕಂಪನಿಗಳಿಗೆ ಸಾಂಪ್ರದಾಯಿಕ ಆವರಣದ IT ಮೂಲಸೌಕರ್ಯದ ಮೇಲೆ ಏರಲು ಅವಕಾಶ ನೀಡುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್, ಆಧುನಿಕ ವ್ಯವಹಾರಗಳಿಗೆ ನಮ್ಯತೆ, ದಕ್ಷತೆ, ಆರೋಹ್ಯತೆ, ಸುರಕ್ಷತೆ, ಹೆಚ್ಚಿದ ಸಹಯೋಗ, ಮತ್ತು ಕಡಿಮೆ ವೆಚ್ಚ ಮುಂತಾದವುಗಳನ್ನು ನೀಡುತ್ತದೆ. COVID -19 ಸಾಂಕ್ರಾಮಿಕವು ಕ್ಲೌಡ್ ಸ್ವೀಕಾರವನ್ನು ಹೆಚ್ಚಿಸಿತು, ಕ್ಲೌಡ್ ತಂತ್ರಜ್ಞಾನಗಳ ಮೇಲಿನ ಅವಲಂಬನೆಯು 2022 ರಲ್ಲಿಯೂ ಇನ್ನೂ ಮುಂದುವರೆದಿದೆ. ವಿಶೇಷವಾಗಿ ಹೈಬ್ರಿಡ್ ಕೆಲಸವು ಕೇಂದ್ರಸ್ಥಾನದ ಹಂತವನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತಿದೆ. ಆದ್ದರಿಂದ ಒಂದು ಸಂಸ್ಥೆಯು ಕ್ಲೌಡ್ ಸೇವೆಗಳನ್ನು ಈಗಾಗಲೇ ಬಳಸುತ್ತಿದ್ದರೆ ಅಥವಾ ಮುಂಬರುವ ವರ್ಷದಲ್ಲಿ ಬಳಸುವ ಯೋಜನೆ ಹೊಂದಿದ್ದರೆ, ಕ್ಲೌಡ್ ಶಕ್ತಿಯ ಪರಿಹಾರಗಳ ಸಂಪೂರ್ಣ ಪ್ರಯೋಜನವನ್ನು ಪಡೆಯಲು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಮೂಲಭೂತ ಅಂಶಗಳನ್ನು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವುದು ಕಡ್ಡಾಯವಾಗಿದೆ.

ಈ ಪಾಠದಲ್ಲಿ, ನಿಖರವಾಗಿ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎಂದರೇನು, ಅದು ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ, ಅದರ ಲಾಭಗಳು ಮತ್ತು ಅನಾನುಕೂಲಗಳು ಮತ್ತು ಕಂಪನಿಗಳು ತಮ್ಮ SaaS ಡಾಟಾವನ್ನು ಉತ್ತಮವಾಗಿ ಹೇಗೆ ರಕ್ಷಿಸಿಕೊಳ್ಳಬಹುದು ಎಂಬುದನ್ನು ತಿಳಿದುಕೊಳ್ಳುತ್ತೇವೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎಂದರೇನು? (What is cloud computing?)

ZDNet ಪ್ರಕಾರ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎಂದರೆ "ಬೇಡಿಕೆ ಆಧಾರದ ಮೇಲೆ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ವಿತರಿಸುವುದಾಗಿದೆ - ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಂದ ಹಿಡಿದು ಸಂಗ್ರಹಣೆಯವರೆಗೆ ಮತ್ತು ಸಂಸ್ಕರಣಾ ಶಕ್ತಿ - ಸಾಮಾನ್ಯವಾಗಿ ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ಮತ್ತು ಪಾವತಿಯ ಆಧಾರದ ಮೇಲೆ"

ಸರಳವಾಗಿ ಹೇಳುವುದಾದರೆ, ಕ್ಲೌಡ್ ಎಂದರೆ ಇಂಟರ್‌ನೆಟ್ ಎಂದು ಅರ್ಥ. ಸಂಸ್ಥೆಗಳು ತಮ್ಮ ಕಂಪ್ಯೂಟರ್‌ನ ಹಾರ್ಡ್ ಡ್ರೈವ್ ಅಥವಾ IT ಮೂಲಸೌಕರ್ಯದ ಆವರಣದ ಮೇಲೆ ಅವಲಂಬಿಸುವ ಬದಲು, ಡಾಟಾವನ್ನು ವರ್ಚುವಲ್ ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಿದಾಗ, ಅಥವಾ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವನ್ನು ಬಳಸಿ ಪ್ರೊಗ್ರಾಂಗಳನ್ನುಪಡೆದುಕೊಂಡಾಗ, ಅವರು ಕ್ಲೌಡನ್ನು ಬಳಸುತ್ತಿದ್ದಾರೆ ಎಂದು ಅರ್ಥ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎಂದರೆ " ಥರ್ಡ್ ಪಾರ್ಟಿ ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿರುವ ಸರ್ವರ್‌ಗಳಾಗಿವೆ" ಅಥವಾ ಸಂಪೂರ್ಣ ಸರ್ವರ್‌ರಹಿತ ವರ್ಕ್‌ಲೋಡ್‌ಗಳು. ಇವು ಅನಂತವಾದ ಸ್ಕೆಲೆಬಲ್ ಮತ್ತು ಭೌಗೋಲಿಕ - ಅಧಿಕವಾಗಿವೆ. ಕ್ಲೌಡ್ ಸರ್ವರ್‌ಗಳು ಮತ್ತು ಸೇವೆಗಳು ಸ್ಕೆಲೆಬಲ್ ಮತ್ತು ಸ್ಥಿತಿಸ್ಥಾಪಕವಾಗಿದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಹೇಗೆ ಕೆಲಸ ಮಾಡುತ್ತದೆ ? (How does cloud computing work?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎನ್ನುವುದು ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳ ವಿತರಣೆಯಾಗಿದೆ. ಉದಾಹರಣೆಗೆ IT ಮೂಲಸೌಕರ್ಯ ಅಥವಾ

© NIMI NOT TO BE REPUBLISHED

ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿರುವ ಡಾಟಾ ಸೆಂಟರ್. ಈ ಮಾದರಿಯು ವ್ಯಾಪಾರಗಳಿಗೆ ತಮ್ಮ ಸ್ವಂತ IT ಮೂಲಸೌಕರ್ಯವನ್ನು ಅಥವಾ ಡಾಟಾ ಸೆಂಟರ್‌ನ್ನು ನಿರ್ಮಿಸುವ ಮತ್ತು ನಿರ್ವಹಿಸುವ ಬದಲು ಸಂಗ್ರಹಣಾ ಸ್ಥಳವನ್ನು ಬಾಡಿಗೆಗೆ ಪಡೆಯಲು ಅಥವಾ ಕ್ಲೌಡ್ ಸೇವಾ ಪೂರೈಕೆದಾರರಿಂದ ಸಾಫ್ಟ್‌ವೇರ್ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ಉಪಯೋಗಿಸುವ ಒಂದು ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ ಕಂಪನಿಗಳು ಅವರು ಬಳಸುವ ಸಂಪನ್ಮೂಲಗಳಿಗೆ ಮಾತ್ರ ಪಾವತಿಸುತ್ತಾರೆ.

ಇದರ ತಾಂತ್ರಿಕ ಅಂಶಗಳನ್ನು ಚನ್ನಾಗಿ ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳಲು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಪ್ರಕ್ರಿಯೆಗಳನ್ನು ಫ್ರಂಟ್‌ಎಂಡ್ ಮತ್ತು ಬ್ಯಾಕ್‌ಎಂಡ್ ಎಂದು ವಿಂಗಡಿಸಬಹುದು. ಫ್ರಂಟ್ ಎಂಡ್ ಘಟಕವು ಇಂಟರ್‌ನೆಟ್ ಬ್ರೌಸರ್ ಮೂಲಕ ಅಥವಾ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್ ಬಳಸುವ ಮೂಲಕ, ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲ್ಪಟ್ಟ ಡಾಟಾ ಮತ್ತು ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಬಳಕೆದಾರರಿಗೆ ಪಡೆದುಕೊಳ್ಳಲು ಅನುಮತಿ ನೀಡುತ್ತವೆ. ಬ್ಯಾಕ್ ಎಂಡ್ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸುವ ಸರ್ವರ್‌ಗಳನ್ನು, ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಮತ್ತು ಡಾಟಾಬೇಸ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಇತಿಹಾಸ (History of cloud computing)

ತಂತ್ರಜ್ಞಾನದ ವಿಮರ್ಶೆಯ ಪ್ರಕಾರ, ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಶಬ್ದವನ್ನು ಮೊದಲು 1996 ರಲ್ಲಿ ಕಾಂಪ್ಯಾಕ್ ಆಂತರಿಕ ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಉಲ್ಲೇಖಿಸಲಾಯಿತು.

1999 ವರ್ಷವು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ಗೆ ಒಂದು ಮೈಲಿಗಲ್ಲು ಆಗಿತ್ತು. Salesforce ಕಂಪನಿಯು ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ವಿತರಿಸಿದ ಮೊದಲ ಕಂಪನಿಯಾಗಿದೆ. ಇದು Software-as-a-Service (SaaS) ನ ಆರಂಭವಾಗಿತ್ತು.

2002 ರಲ್ಲಿ, ಅಮೆಝಾನ್ ಕಂಪನಿಯು Amazon Web Services (AWS) ನ್ನು ಪ್ರಾರಂಭಿಸಿತು. ಇದು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಇನ್ನೊಂದು ಗಮನಾರ್ಹ ಬೆಳವಣಿಗೆಯಾಗಿತ್ತು. ಅದರ ಕ್ಲೌಡ್ ಆಧಾರಿತ ಸೇವೆಗಳು ಸಂಗ್ರಹಣೆಯನ್ನು, ಗಣನಾಕಾರ್ಯಗಳನ್ನು, ಮತ್ತು ಮಾನವ ಬುದ್ಧಿಮತ್ತೆಯನ್ನೂ ಸಹ ಒಳಗೊಂಡಿದೆ. 2006 ರಲ್ಲಿ, ಅಮೇಝಾನ್ ಕಂಪನಿಯು Elastic Compute Cloud (EC2) ನ್ನು ಆರಂಭಿಸಿತು. ಇದು ವ್ಯಾಪಾರಗಳಿಗೆ ಮತ್ತು ವೈಯಕ್ತಿಕವಾಗಿ ವರ್ಚುವಲ್ ಕಂಪ್ಯೂಟರ್‌ಗಳನ್ನು ಬಾಡಿಗೆಗೆ ನೀಡಿತು ಮತ್ತು ಅವರ ಕಂಪ್ಯೂಟರ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರನ್ ಮಾಡಲು ಅವಕಾಶ ನೀಡಿತು.

2009 ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನಲ್ಲಿ Google Workspace ಎನ್ನುವ ಮತ್ತೊಂದು ಮೈಲಿಗಲ್ಲನ್ನು ಕಂಡಿತು. ಇದು ಬ್ರೌಸರ್ ಆಧಾರಿತ ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಒದಗಿಸಲು ಪ್ರಾರಂಭಿಸಿತು. ಅದೇ ವರ್ಷ ಮೈಕ್ರೊಸಾಫ್ಟ್ ಕಂಪನಿಯು Microsoft Azure ದೊಂದಿಗೆ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಕ್ಷೇತ್ರವನ್ನು ಪ್ರವೇಶಿಸಿತು. ನಂತರ ಶೀಘ್ರದಲ್ಲಿಯೆ ಒರ್ಯಾಕಲ್ ಮತ್ತು ಎಚ್‌ಪಿ ಅಂತಹ ಕಂಪನಿಗಳೂ ಇದನ್ನು ಅನುಸರಿಸಿದವು.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಉದಾಹರಣೆಗಳು ಯಾವವು ? (What are examples of cloud computing?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್-ಸಂಪೂರ್ಣ ಸರ್ವರ್‌ರಹಿತ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗೆ, ವರ್ಚುವಲ್ ಮಶೀನ್‌ಗಳಿಂದ ಡಾಟಾಬೇಸ್‌ಗಳವರೆಗೆ ಎಲ್ಲವನ್ನೂ ಒಳಗೊಂಡಿರುತ್ತದೆ. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಕೆಲವು ಉದಾಹರಣೆಗಳು ಕೆಳಗಿನಂತಿವೆ :

ಸೇಲ್ಸ್ ಫೋರ್ಸ್ (Salesforce) : Salesforce.com ಇದು SaaS ನ್ನು ಒದಗಿಸುತ್ತದೆ (ಪ್ರೊವೈಡರ್). ಇದು customer relationship management (CRM) ನಲ್ಲಿ ಪರಿಣಿತಿ ಹೊಂದಿದೆ. ಕಂಪನಿಯು ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ ಇದು ಮಾರ್ಕೆಟಿಂಗ್, ಮಾರಾಟ, ಗ್ರಾಹಕರ ಸೇವೆಗಳು ಇತ್ಯಾದಿಗಳಿಗೆ ಸಹಾಯ ಮಾಡುತ್ತದೆ ಮತ್ತು ಬಳಕೆದಾರರಿಗೆ ಎಲ್ಲಿಂದಾದರೂ ಕೆಲಸ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಡಿಜಿಟಲ್ ಓಷನ್ (Digital Ocean) : ಸಾಫ್ಟ್‌ವೇರ್ ಡೆವಲಪರ್‌ಗಳಿಗೆ ಈ ಕಂಪನಿಯು ನ್ಯೂಯಾರ್ಕ್ ಮೂಲದ Infrastructure-as-a-Service (IaaS) ಪೂರೈಕೆದಾರ ಆಗಿದೆ. ವ್ಯಾಪಾರಗಳು ಅನೇಕ ಕ್ಲೌಡ್ ಸರ್ವರ್‌ಗಳಿಗೆ ಏಕಕಾಲದಲ್ಲಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ನಿಯೋಜಿಸಲು ಡಿಜಿಟಲ್ ಓಷನ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ.

ಮೈಕ್ರೊಸಾಫ್ಟ್ ಅಜುರೆ (Microsoft Azure: Microsoft Azure ಇದು Platform-as-a-Service (PaaS) ಗೆ ಉತ್ತಮ ಉದಾಹರಣೆಯಾಗಿದೆ. ಇದು ಸಂಪೂರ್ಣ ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಅಭಿವೃದ್ಧಿಯಿಂದ ನಿಯೋಜನೆಯವರೆಗೆ ಮತ್ತು ಅದಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು. Azure ಡೆವಲಪರ್‌ಗಳಿಗೆ ಸಾಧನಗಳ, ಭಾಷೆಗಳ, ಮತ್ತು ಫ್ರೇಮ್‌ವರ್ಕ್‌ಗಳ ಸಮೃದ್ಧಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಡ್ರಾಪ್‌ಬಾಕ್ಸ್ (Dropbox): ಡ್ರಾಪ್‌ಬಾಕ್ಸ್ ಇದು ಕ್ಲೌಡ್ ಆಧಾರಿತ ಫೈಲ್ ಹೋಸ್ಟಿಂಗ್ ಸೇವೆಯಾಗಿದೆ. ಇದು ಬಳಕೆದಾರರಿಗೆ ತಮ್ಮ ಸಾಧನಗಳಿಗೆ ಫೈಲ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಮತ್ತು ಸಿಂಕ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದರಿಂದ ಅವರು ತಮ್ಮ ಫೈಲ್‌ಗಳನ್ನು ಎಲ್ಲಿಂದಾದರೂ ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಇದು ಬಳಕೆದಾರರಿಗೆ ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಚಿತ್ರಗಳು ಮತ್ತು ವಿಡಿಯೋಗಳನ್ನು ಒಳಗೊಂಡಂತೆ ದೊಡ್ಡ ಗಾತ್ರದ ಫೈಲ್‌ಗಳನ್ನು ಹಂಚಿಕೊಳ್ಳಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದು ಪರಿಣಾಮಕಾರಿ ಸಹಯೋಗವನ್ನು ಸುಗಮಗೊಳಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಪ್ರಾಮುಖ್ಯತೆ ಏನು ? (What is the importance of cloud computing?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಅಸ್ತಿತ್ವಕ್ಕೆ ಬರುವ ಮೊದಲು ಕಂಪನಿಗಳು ತಮ್ಮ ಭೌತಿಕ ಪಿಸಿಗಳಲ್ಲಿ ಅಥವಾ ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಮತ್ತು ಪ್ರೋಗ್ರಾಂಗಳನ್ನು ಡೌನ್‌ಲೋಡ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇತ್ತು. ಯಾವುದೇ ಸಂಸ್ಥೆಗೆ ತನ್ನ ಸ್ವಂತ IT ಮೂಲಸೌಕರ್ಯವನ್ನು ಅಥವಾ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳನ್ನು ನಿರ್ಮಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು ಒಂದು ದೊಡ್ಡ ಸವಾಲಾಗಿದೆ. ತಮ್ಮದೇ ಆದ ಸ್ವಂತ ಡಾಟಾ ಸೆಂಟರ್‌ನ್ನು ಹೊಂದಿರುವವರಿಗೂ ಸಹ ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯ IT ನಿರ್ವಾಹಕರನ್ನು ಮತ್ತು ಸಂಪನ್ಮೂಲಗಳನ್ನು ಹಂಚಿಕೆ ಮಾಡುವುದು ಒಂದು ಹೋರಾಟವಾಗಿರುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಮತ್ತು ವರ್ಚುವಲೈಸೇಷನ್‌ನ ಪರಿಚಯವು, ತಂತ್ರಜ್ಞಾನದ ಉದ್ಯಮದ ಇತಿಹಾಸದಲ್ಲಿ ಒಂದು ಮಾದರಿ ಬದಲಾವಣೆಯಾಯಿತು. ತಮ್ಮ ಸ್ವಂತ IT ಮೂಲಸೌಕರ್ಯವನ್ನು ತಯಾರಿಸುವುದು ಮತ್ತು ನಿರ್ವಹಿಸುವುದು ಹಾಗೂ ಸರ್ವರ್‌ಗಳಿಗೆ, ವಿದ್ಯುತ್, ಮತ್ತು ರಿಯಲ್ ಎಸ್ಟೇಟ್ ಇತ್ಯಾದಿಗಳಿಗೆ ಪಾವತಿಸುವ ಬದಲು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್, ಕ್ಲೌಡ್ ಸರ್ವೀಸ್ ಪ್ರೊವೈಡರ್‌ಗಳಿಂದ ಕಂಪ್ಯೂಟರ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಬಾಡಿಗೆ ಪಡೆದುಕೊಳ್ಳಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದು ವ್ಯವಹಾರಗಳಿಗೆ ಭಾರಿ ಮುಂಗಡ ವೆಚ್ಚಗಳನ್ನು ಪಾವತಿಸುವುದನ್ನು ತಪ್ಪಿಸುತ್ತದೆ ಮತ್ತು ತಮ್ಮ ಸ್ವಂತ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳನ್ನು ನಿರ್ವಹಿಸುವ ಸಂಕೀರ್ಣತೆಯನ್ನು ತಪ್ಪಿಸುತ್ತದೆ. ಕ್ಲೌಡ್ ಸೇವೆಗಳನ್ನು ಬಾಡಿಗೆ ಪಡೆಯುವುದರಿಂದ ಕಂಪನಿಗಳು ತಾವು ಬಳಸುವ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳು ಮತ್ತು ಡಿಸ್ಕ್ ಸ್ಪೇಸ್‌ನಂತಹವುಗಳಿಗೆ ಮಾತ್ರ ಪಾವತಿಸಬೇಕಾಗುತ್ತದೆ. ಇದು ಕಂಪನಿಗಳು ಹೆಚ್ಚಿನ ನಿಖರತೆಯ ವೆಚ್ಚಗಳನ್ನು ನಿರೀಕ್ಷಿಸುವಂತೆ ಮಾಡುತ್ತದೆ.

ಕ್ಲೌಡ್ ಸರ್ವೀಸ್ ಪ್ರೊವೈಡರ್‌ಗಳು IT ಮೂಲಸೌಕರ್ಯವನ್ನು ನಿಭಾಯಿಸುವ ಮತ್ತು ನಿರ್ವಹಿಸುವ ಭಾರಿ ಕಾರ್ಯ ನಿರ್ವಹಿಸುವುದರಿಂದ ಇದು ವ್ಯಾಪಾರದ ಸಾಕಷ್ಟು ಸಮಯ, ಶ್ರಮ, ಹಣವನ್ನು ಉಳಿಸುತ್ತದೆ. ಕ್ಲೌಡ್, ಕಂಪನಿಗಳಿಗೆ ಅವರ ಅಗತ್ಯಕ್ಕೆ ತಕ್ಕಂತೆ ಕಂಪ್ಯೂಟಿಂಗ್ ಮೂಲಸೌಕರ್ಯವನ್ನು ಮೇಲ್ದರ್ಜೆಗೆ ಏರಿಸುವ ಅಥವಾ ಕಡಿಮೆ ಮಾಡುವ ಸಾಮರ್ಥ್ಯವನ್ನು ನೀಡುತ್ತದೆ. ಸಾಂಪ್ರದಾಯಿಕ ಡಾಟಾ ಸೆಂಟರ್ ಮಾದರಿಗೆ ಹೋಲಿಸಿದರೆ ಕ್ಲೌಡ್ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕದ ಮೂಲಕ ಎಲ್ಲಿಂದಲೂ ಸಹ ಮತ್ತು ಯಾವುದೇ ಸಾಧನದಲ್ಲಿಯೂ ಡಾಟಾಗೆ ಸುಲಭ ಪ್ರವೇಶವನ್ನು ನೀಡುತ್ತದೆ. ಹೀಗೆ ಪರಿಣಾಮಕಾರಿ ಸಹಯೋಗ ಮತ್ತು ಉತ್ಪಾದಕತೆಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಸಾಮಾನ್ಯ ಉಪಯೋಗಗಳು ಯಾವವು? (What are the most common uses of cloud computing?)

ಸ್ಟಾರ್ಟ್‌ಅಪ್‌ಗಳಿಂದ ಹಿಡಿದು ದೊಡ್ಡ ಸಂಸ್ಥೆಗಳವರೆಗೆ ಮತ್ತು ಸರ್ಕಾರಿ ಏಜನ್ಸಿಗಳವರೆಗೆ ಪ್ರತಿಯೊಂದು ಸಂಸ್ಥೆಯು ಕ್ಲೌಡ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ - ತಂತ್ರಜ್ಞಾನ ಸೇವೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು, ಕೆಲಸದ ಹರಿವನ್ನು ಸುಗಮಗೊಳಿಸಲು, ಸಂವಹನವನ್ನು ಸುಧಾರಿಸಲು, ಉತ್ಪಾದಕತೆ, ಸೇವೆ ವಿತರಣೆ ಇತ್ಯಾದಿಗಳು.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಕೆಲವು ಸಾಮಾನ್ಯ ಉಪಯೋಗಗಳನ್ನು ಈ ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ :

- **ಸಂಗ್ರಹಣೆ** (Storage): ಫೈಲ್ ಸಂಗ್ರಹಣೆಯು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಅತ್ಯಂತ ಸಾಮಾನ್ಯ ಉಪಯೋಗವಾಗಿದೆ. ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸಲು ಮತ್ತು ಪಡೆದುಕೊಳ್ಳಲು, ಪಿಸಿಗಳಲ್ಲಿರುವ ಹಾರ್ಡ್ ಡ್ರೈವ್‌ಗಳು, ಬಾಹ್ಯ ಹಾರ್ಡ್ ಡ್ರೈವ್‌ಗಳು, USB ಡ್ರೈವ್‌ಗಳು ಇತ್ಯಾದಿಗಳಂತಹ ಹಲವಾರು ಆಯ್ಕೆಗಳಿವೆ. ಕ್ಲೌಡ್ ಸಂಗ್ರಹಣೆಯು ವ್ಯಾಪಾರಗಳಿಗೆ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕದ ಮೂಲಕ ಡಾಟಾವನ್ನು ಎಲ್ಲಿಂದಾದರೂ ಮತ್ತು ಯಾವುದೇ ಸಾಧನದಿಂದಾದರೂ ಪಡೆದುಕೊಳ್ಳಲು ಅನುಮತಿಸುತ್ತದೆ. ಕ್ಲೌಡ್ ಸಂಗ್ರಹಣಾ ಸೇವೆಗಳಾದ Amazon S3, DropBox ಅಥವಾ OneDrive ಗಳು ಡಾಟಾದ ಸುರಕ್ಷಿತ ಪಡೆಯುವಿಕೆಯನ್ನು ಒದಗಿಸುತ್ತವೆ ಮತ್ತು ವ್ಯಾಪಾರಗಳಿಗೆ ತಮ್ಮ ಅಗತ್ಯಕ್ಕೆ ತಕ್ಕಂತೆ ಸಂಗ್ರಹಣಾ ಸ್ಥಳವನ್ನು ಮೇಲ್ದರ್ಜೆಗೆ ಏರಿಸಲು ಮತ್ತು ಇಳಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತವೆ.
- **ಡಾಟಾಬೇಸ್** (Database) : ಕ್ಲೌಡ್ ಡಾಟಾಬೇಸ್ ಇನ್ನೊಂದು ಜನಪ್ರಿಯ ವ್ಯಾಪಾರದ ಬಳಕೆಯಾಗಿದೆ. ಕ್ಲೌಡ್ ಡಾಟಾಬೇಸನ್ನು IBM ಹೀಗೆ ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ - "ಇದು ಕ್ಲೌಡ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಮೂಲಕ ನಿರ್ಮಿಸಲಾದ ಮತ್ತು ಪ್ರವೇಶಿಸುವ ಒಂದು ಡಾಟಾಬೆಸ್ ಸೇವೆಯಾಗಿದೆ". ಒಂದು ಕ್ಲೌಡ್ ಡಾಟಾಬೇಸ್ ಸಹ ಸಾಂಪ್ರದಾಯಿಕ ಡಾಟಾಬೆಸ್‌ನಂತಹ ಅದೇ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ನೀಡುತ್ತದೆ, ಆದರೆ ನಮ್ಯತೆ, ವೆಚ್ಚ ಉಳಿತಾಯ, ವೈಫಲ್ಯ ಬೆಂಬಲ, ವಿಶೇಷ ಪರಿಣಿತಿ ಮತ್ತು ಇನ್ನೂ ಹೆಚ್ಚಿನ ಲಾಭಗಳನ್ನು ನೀಡುತ್ತದೆ.
- **ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು** (Web applications) : ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಇವತ್ತಿನ ವ್ಯವಹಾರಗಳಲ್ಲಿ ಇರಲೇಬೇಕಾದ ಸಾಧನಗಳಾಗಿವೆ. ಕ್ಲೌಡ್ ತಂತ್ರಜ್ಞಾನದಿಂದ ವೆಬ್ ಬ್ರೌಸರ್‌ನ್ನು ಬಳಸಿಕೊಂಡು ಯಾರಾದರೂ ವೆಬ್ ಆಧಾರಿತ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಇದು ಶೀಘ್ರವಾಗಿ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ವ್ಯಾಪಾರ ವೃತ್ತಿಪರರಿಗೆ ಗ್ರಾಹಕರೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಲು ಮತ್ತು ಅವರಿಗೆ ಅಗತ್ಯ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಎಲ್ಲಿದ್ದರೂ ತಮ್ಮ ಸಹೋದ್ಯೋಗಿಗಳೊಂದಿಗೆ ಸಹಕರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

- **ಸಹಯೋಗ** (Collaboration): ಇದರ ಸುಲಭ ಪ್ರವೇಶ, ಸಮಗ್ರತೆ, ನಮ್ಯತೆ, ಭದ್ರತೆ ಮತ್ತು ಸುಲಭ ಬಳಕೆ ಮುಂತಾದವುಗಳಿಂದ, ಕ್ಲೌಡ್ ಆಧಾರಿತ ಸಾಧನಗಳಾದ ಮೈಕ್ರೊಸಾಫ್ಟ್ 365 ಮತ್ತು ಗೂಗಲ್ ವರ್ಕ್‌ಸ್ಪೇಸ್ ಇವು ವ್ಯಾಪಾರಗಳಿಗಾಗಿ ಸ್ಪಷ್ಟ ಆಯ್ಕೆಯಾಗಿವೆ. ಇದು ಆಂತರಿಕವಾಗಿ ಇಲಾಖೆಗಳೊಂದಿಗೆ ಮತ್ತು ಬಾಹ್ಯವಾಗಿ ಗ್ರಾಹಕರೊಂದಿಗೆ ಸಹಯೋಗ ಬಯಸುವ ವ್ಯಾಪಾರಗಳ ಆಯ್ಕೆಯಾಗಿದೆ. Gmail, Google docs, Microsoft Outlook, Microsoft Word, Teams, ಇತ್ಯಾದಿಗಳು ಸಹಯೋಗ ಮತ್ತು ಉತ್ಪಾದಕತೆಯನ್ನು ಹೆಚ್ಚಿಸಲು ವಿನ್ಯಾಸಗೋಳಿಸಲಾದ ಪ್ರಬಲ ವ್ಯಾಪಾರ ಸಾಧನಗಳಾಗಿವೆ.

- **SaaS ಅಪ್ಲಿಕೇಷನ್‌ಗಳು(SaaS applications)** : Salesforce ನಂತಹ Software-as-a-Service (SaaS) ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ವ್ಯಾಪಾರಗಳಿಗೆ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸಲು, ಸಂಘಟಿಸಲು, ಮತ್ತು ನಿರ್ವಹಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತವೆ ಮತ್ತು ಮಾರ್ಕೆಟಿಂಗ್‌ನ್ನು ಆಟೋಮೇಟ್ ಮಾಡುತ್ತವೆ ಮತ್ತು ಗ್ರಾಹಕರನ್ನು ಸಮರ್ಥವಾಗಿ ನಿರ್ವಹಿಸುತ್ತವೆ. SaaS ಪರಿಹಾರಗಳು ಹೆಚ್ಚು ಕ್ರಿಯಾತ್ಮಕವಾಗಿರುತ್ತವೆ ಮತ್ತು ಇದಕ್ಕೆ ಯಾವುದೇ ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ನಿರ್ವಹಣೆಯ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ವಿವಿಧ ಪ್ರಕರಣಗಳು ಯಾವುವು ? (What are the different types of cloud computing?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನಲ್ಲಿ ನಾಲ್ಕು ಮುಖ್ಯ ವಿಧಗಳಿವೆ : ಪಬ್ಲಿಕ್, ಪ್ರೈವೇಟ್, ಹೈಬ್ರಿಡ್ ಮತ್ತು ಮಲ್ಟಿಕ್ಲೌಡ್.

ಪಬ್ಲಿಕ್ ಕ್ಲೌಡ್ (Public cloud) :

VMware ಪಬ್ಲಿಕ್ ಕ್ಲೌಡ್‌ನ್ನು ಹೀಗೆ ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ "ಇದು IT ಮಾಡೆಲ್ ಆಗಿದ್ದು, ಇಲ್ಲಿ ಬೇಡಿಕೆಯ ಆಧಾರದ ಮೇಲೆ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ಮತ್ತು ಮೂಲಸೌಕರ್ಯವನ್ನು ಮೂರನೆ ವ್ಯಕ್ತಿಯ ಪೂರೈಕೆದಾರರಿಂದ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಪಬ್ಲಿಕ್ ಇಂಟರ್‌ನೆಟ್‌ನ್ನು ಬಳಸಿ ಅನೇಕ ಸಂಸ್ಥೆಗಳಿಗೆ ಹಂಚಲಾಗುತ್ತದೆ". ಕ್ಲೌಡ್ ಸೇವಾ ಪೂರೈಕೆದಾರರು Infrastructure-as-a- Service (IaaS), Platform-as-a-Service (PaaS) and SaaS ನಂತಹ ಅನೇಕ ಸೇವೆಗಳನ್ನು ವೈಯಕ್ತಿಕವಾಗಿ ಮತ್ತು ವ್ಯಾಪಾರಗಳಿಗೆ ಒದಗಿಸುತ್ತಾರೆ. ಈ ಸೇವೆಗಳಿಗೆ ಅವರು ಮಾಸಿಕ ಅಥವಾ ಬಳಕೆಯ ಆಧಾರದ ಮೇಲೆ ಪಾವತಿಸುವ ಬಾಡಿಗೆ ಪಡೆಯುತ್ತಾರೆ. Amazon Web Services (AWS), Microsoft Azure, Google Cloud, Alibaba Cloud ಮತ್ತು IBM Cloud ಇವರು ಐದು ಅಗ್ರ ಕ್ಲೌಡ್ ಪೂರೈಕೆದಾರರಾಗಿದ್ದಾರೆ.

ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್ (Private cloud)

ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್ ಅಥವಾ ಆಂತರಿಕ ಕ್ಲೌಡ್ ಎಂದರೆ, IT ಮೂಲಸೌಕರ್ಯ (ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಸಂಪನ್ಮೂಲಗಳು) ಗಳು ಒಂದೇ ಸಂಸ್ಥೆಗೆ ಮಾತ್ರ ಸಮರ್ಪಿತವಾಗಿರುತ್ತವೆ. ಆದರೆ ಪಬ್ಲಿಕ್ ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಹಲವಾರು ಬಾಡಿಗೆದಾರರಲ್ಲಿ ಹಂಚಲಾಗುತ್ತದೆ. ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್ ಪರಿಸರವು ವ್ಯವಹಾರಗಳಿಗೆ ಸೂಕ್ತವಾಗಿದೆ.

ಯಾರಿಗೆ ನಿಯಂತ್ರಕ ಅವಶ್ಯಕತೆಗಳನ್ನು ಪೂರೈಸುವ, ಭದ್ರತೆ ಮತ್ತು ನಿಯಂತ್ರಣಗಳು ಆದ್ಯತೆಗಳಾಗಿವೆಯೊ ಅವರಿಗೆ ಸೂಕ್ತವಾಗಿದೆ. ಸಾಂಪ್ರದಾಯಿಕವಾಗಿ, ಪ್ರೈವೇಟ್ ಕ್ಲೌಡನ್ನು ಕಂಪನಿಯ ಡಾಟಾ ಸೆಂಟರ್‌ನಲ್ಲಿ ಹೋಸ್ಟ್ ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ತನ್ನ ಸ್ವಂತ ಹಾರ್ಡ್‌ವೇರನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತದೆ. ಅದಾಗ್ಯೂ ಸಂಸ್ಥೆಯು ಹೋಸ್ಟಿಂಗ್‌ನ್ನು ಮೂರನೇ ಪಕ್ಷದ ಪೂರೈಕೆದಾರರಿಗೆ ಹೊರಗುತ್ತಿಗೆ ನೀಡಬಹುದು. ಇದು ದೂರದಿಂದಲೇ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಹೈಬ್ರಿಡ್ ಕ್ಲೌಡ್ (Hybrid cloud)

ಹೈಬ್ರಿಡ್ ಕ್ಲೌಡ್ ಇದು ಪಬ್ಲಿಕ್ ಕ್ಲೌಡ್ ಮತ್ತು ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್ ಎರಡೂ ಪರಿಸರಗಳ ಸಂಯೋಜನೆಯಾಗಿದೆ. ವ್ಯಾಪಾರಗಳು ಈ ಮಾದರಿಯನ್ನು ತಮ್ಮ ಕಂಪ್ಯೂಟ್ ಸಾಮರ್ಥ್ಯಕ್ಕೆ ಪೂರಕವಾಗಿ ಬಳಸುತ್ತವೆ. ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್‌ನ ಸಾಮರ್ಥ್ಯವು ತನ್ನ ಉತ್ತುಂಗವನ್ನು ತಲುಪಿದಾಗ ವ್ಯಾಪಾರಗಳು ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್‌ನ ಸಾಮರ್ಥ್ಯಗಳನ್ನು ಹೆಚ್ಚಿಸಲು ಪಬ್ಲಿಕ್ ಕ್ಲೌಡನ್ನು ನಿಯಂತ್ರಿಸಬಹುದು. ಹೈಬ್ರಿಡ್ ಕ್ಲೌಡ್ ವ್ಯಾಪಾರಗಳಿಗೆ ಸೇವೆಯ ಬೇಡಿಕೆಗಳಿಗೆ ತಕ್ಕಂತೆ ಕಂಪ್ಯೂಟ್ ಸಾಮರ್ಥ್ಯವನ್ನು ಅಳೆಯಲು ಶಕ್ತಗೊಳಿಸುತ್ತದೆ. ಇದು ಹೊಸ ಸರ್ವರ್‌ಗಳನ್ನು ಖರೀದಿಸುವ ಮತ್ತು ನಿರ್ವಹಿಸುವ ಅವಶ್ಯಕತೆಯನ್ನು ನಿವಾರಿಸುತ್ತದೆ ಮತ್ತು ವ್ಯಾಪಾರಗಳ ವೆಚ್ಚವನ್ನು, ಸಮಯವನ್ನು ಮತ್ತು ಶ್ರಮವನ್ನು ಉಳಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ.

ಮಲ್ಟಿಕ್ಲೌಡ್ (Multicloud)

ಮಲ್ಟಿಕ್ಲೌಡ್ ಎಂದರೆ ಎರಡು ಅಥವಾ ಹೆಚ್ಚು ಪಬ್ಲಿಕ್ ಅಥವಾ ಪ್ರೈವೇಟ್ ಕ್ಲೌಡ್‌ಗಳ ಸಂಯೋಜನೆ ಅಥವಾ ಎರಡೂ ಕ್ಲೌಡ್‌ಗಳ ಸಂಯೋಜನೆಯನ್ನು ಉಪಯೋಗಿಸುವ ಅಭ್ಯಾಸವಾಗಿದೆ. ಮಲ್ಟಿಕ್ಲೌಡ್ ವಿಧಾನದಲ್ಲಿ ವ್ಯಾಪಾರಗಳು ತಮ್ಮ ಬಜೆಟ್, ತಾಂತ್ರಿಕ ಅವಶ್ಯಕತೆಗಳು, ಭೌಗೋಳಿಕ ಸ್ಥಳಗಳು ಮುಂತಾದವುಗಳ ಆಧಾರದ ಮೇಲೆ ವಿಭಿನ್ನ ಕ್ಲೌಡ್ ಪೂರೈಕೆದಾರರಿಂದ ಉತ್ತಮ ಸೇವೆಗಳನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ. ಈ ಮಾದರಿಯು ವ್ಯಾಪಾರಗಳಿಗೆ ವಿಭಿನ್ನ ಕ್ಲೌಡ್‌ಗಳನ್ನು ವಿಭಿನ್ನ ಉದ್ದೇಶಗಳಿಗಾಗಿ ಬಳಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ, ಒಂದು ಸಂಸ್ಥೆಯು ಒಂದು ಕ್ಲೌಡನ್ನು ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಮತ್ತು ಪರೀಕ್ಷಿಸಲು

© NIMI NOT TO BE REPUBLISHED

, ಇನ್ನೊಂದು ಕ್ಲೌಡನ್ನು ಡಾಟಾ ಬ್ಯಾಕಪ್‌ಗಾಗಿ ಮತ್ತು ವಿಪತ್ತು ನಿರ್ವಹಣೆಗಾಗಿ ಹಾಗೂ ಇನ್ನೊಂದನ್ನು ಡಾಟಾ ವಿಶ್ಲೇಷಣೆಗಾಗಿ ಉಪಯೋಗಿಸಬಹುದು.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಯ ಮೂರು ವಿಭಿನ್ನ ಪ್ರಕಾರಗಳು ಯಾವವು? (What are the three different types of cloud computing services?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಯ ಮೂರು ಪ್ರಕಾರಗಳು ಯಾವವೆಂದರೆ : Infrastructure-as-a-Service (IaaS), Platform-as-a-Service(PaaS) and Software-as-a-Service (SaaS).

Infrastructure-as-a-Service (IaaS)

IaaS ಇದು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಯಾಗಿದ್ದು, ಇಲ್ಲಿ ಕ್ಲೌಡ್ ಪೂರೈಕೆದಾರರು ಇಂಟರ್‌ನೆಟ್‌ನಲ್ಲಿ ವರ್ಚುವಲೈಸ್ಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಮೂಲಸೌಕರ್ಯವನ್ನು ವಿತರಿಸುತ್ತವೆ ಮತ್ತು ನಿರ್ವಹಿಸುತ್ತವೆ. ಆಂತರಿಕವಾಗಿ IT ಮೂಲಸೌಕರ್ಯವನ್ನು ರಚಿಸುವ ಬದಲು, ವ್ಯಾಪಾರಗಳು ಅವಶ್ಯಕ ಸಂಪನ್ಮೂಲಗಳಾದ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು, ನೆಟ್‌ವರ್ಕಿಂಗ್, ಸಂಗ್ರಹಣಾ ಸ್ಥಳ, ಅಭಿವೃದ್ಧಿ ಉಪಕರಣಗಳು ಇತ್ಯಾದಿಗಳನ್ನು ಬೇಡಿಕೆಯ ಮೇರೆಗೆ ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಇದು ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ವೆಚ್ಚಗಳನ್ನು ಉಳಿಸುತ್ತದೆ ಮತ್ತು IT ಸಿಬ್ಬಂದಿಯ ಹೊರೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

Platform-as-a-Service (PaaS)

PaaS ಇದು ವ್ಯವಹಾರಗಳು ಸಾಫ್ಟ್‌ವೇರ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳ ಹಾಗೂ ಸೇವೆಗಳ ಅಭಿವೃದ್ಧಿ, ನಿಯೋಜನೆ ಮತ್ತು ನಿರ್ವಹಣೆಯನ್ನು ಕೇಂದ್ರೀಕರಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ ಕ್ಲೌಡ್ ಪೂರೈಕೆದಾರರು ಭಾರಿ ನಿರ್ವಹಣೆಯನ್ನು ಮಾಡುವುದರಿಂದ ಮೂಲಸೌಕರ್ಯದ ಕಾಳಜಿ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ. PaaS ನೊಂದಿಗೆ ಡೆವಲಪರ್‌ಗಳು ಮತ್ತು ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು IT ಮೂಲಸೌಕರ್ಯವನ್ನು ಮಾತ್ರ ಪಡೆಯುವುದಲ್ಲದೇ, ಅಪ್ಲಿಕೇಷನ್ / ಸಾಫ್ಟ್‌ವೇರ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಮತ್ತು ಪರಿಹಾರ ಸ್ಟ್ಯಾಕ್‌ಗಳನ್ನೂ ಪಡೆದುಕೊಳ್ಳುತ್ತಾರೆ. PaaS ನ ಕೆಲವು ಉದಾಹರಣೆಗಳು ಯಾವವೆಂದರೆ AWS Elastic Beanstalk, Google App Engine ಮತ್ತು Microsoft Azure.

Software-as-a-Service (SaaS)

SaaS ಇದು ವ್ಯಾಪಾರಗಳಿಗೆ ಬಳಸಲು ಸಿದ್ಧವಾದ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದನ್ನು ಇಂಟರ್‌ನೆಟ್ ಮೂಲಕ ಬಳಕೆದಾರರಿಗೆ ತಲುಪಿಸಲಾಗುತ್ತದೆ. ಹಾರ್ಡ್‌ವೇರ್, ಸಾಫ್ಟ್‌ವೇರ್, ಡಾಟಾ ಸಂಗ್ರಹಣೆ, ಪ್ಯಾಚ್ ನಿರ್ವಹಣೆ ಮತ್ತು ಹಾರ್ಡ್‌ವೇರ್/ ಸಾಫ್ಟ್‌ವೇರ್ ನವೀಕರಣಗಳು ಸೇರಿದಂತೆ ಆಧಾರವಾಗಿರುವ ಎಲ್ಲಾ ಮೂಲಸೌಕರ್ಯಗಳನ್ನು SaaS ಪೂರೈಕೆದಾರರು ನಿರ್ವಹಿಸುತ್ತಾರೆ. SaaS ಇದು ಸಬ್‌ಸ್ಕ್ರೈಬ್ - ಆಧಾರಿತ ಮಾದರಿಯಾಗಿದೆ. ಅಂದರೆ ವ್ಯಾಪಾರಗಳು ಅವರು ಬಳಸಲು ಬಯಸುವ ಸೇವೆಗಳಿಗಾಗಿ ಚಂದಾದಾರರಾಗುವ ಅಗತ್ಯವಿದೆ.

ಬಳಕೆದಾರರು ವೆಬ್ ಬ್ರೌಸರ್‌ಗಳ ಮೂಲಕ SaaS ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ನೇರವಾಗಿ ಪಡೆದುಕೊಳ್ಳಬಹುದು. ಇದರಿಂದ ಅವುಗಳನ್ನು ಡೌನ್‌ಲೋಡ್ ಅಥವಾ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಿಕೊಳ್ಳುವುದನ್ನು ತಪ್ಪಿಸುತ್ತದೆ. ಬಳಕೆದಾರರು ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕದ ಮೂಲಕ ವೆಬ್ ಆಧಾರಿತ ಪರಿಹಾರಗಳನ್ನು ಎಲ್ಲಿಂದಾದರೂ ಮತ್ತು ಯಾವ ಸಮಯದಲ್ಲಿಯೂ ಪಡೆದುಕೊಳ್ಳಲು SaaS ಅನುಮತಿಸುತ್ತದೆ. ಕೆಲವು ಜನಪ್ರಿಯ SaaS ಪರಿಹಾರಗಳು Microsoft 365, Google Workspace and Salesforce ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಪ್ರಯೋಜನಗಳೇನು? (What are the benefits of cloud computing?)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ವ್ಯಾಪಾರಗಳನ್ನು ವರ್ಚುವಲ್ ಆಗಿ ಎಲ್ಲಿಂದಾದರೂ ಮತ್ತು ಹೆಚ್ಚು ದಕ್ಷತೆಯೊಂದಿಗೆ ನಿರ್ವಹಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಕೆಲವು ಪ್ರಯೋಜನೆಗಳು ಹೀಗಿವೆ :

- **ವೆಚ್ಚ ಉಳಿತಾಯ** (Cost savings): ವೆಚ್ಚವನ್ನು ಕಡಿಮೆ ಮಾಡುವುದು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಅತ್ಯುತ್ತಮ ಪ್ರಯೋಜನಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ವ್ಯವಹಾರಗಳು ತಮ್ಮ ಸ್ವಂತ IT ಮೂಲಸೌಕರ್ಯವನ್ನು ನಿರ್ಮಿಸುವ ಅಥವಾ ಹಾರ್ಡ್‌ವೇರನ್ನು ಅಥವಾ ಉಪಕರಣಗಳನ್ನು ಖರೀದಿಸುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ. ಇದು ಕಂಪನಿಗಳ ಬಂಡವಾಳ ವೆಚ್ಚಗಳನ್ನು ಗಮನಾರ್ಹವಾಗಿ ಕಡಿಮೆ ಮಾಡಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ.

- **ಹೊಂದಿಕೊಳ್ಳುವಿಕೆ/ ಆರೋಹ್ಯತೆ** (Flexibility/scalability) : ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಎಲ್ಲ ಗಾತ್ರದ ವ್ಯವಹಾರಗಳಿಗೆ ಹೆಚ್ಚಿನ ಹೊಂದಿಕೊಳ್ಳುವಿಕೆಯನ್ನು ನೀಡುತ್ತದೆ. ಅವುಗಳಿಗೆ ಹೆಚ್ಚುವರಿ ಬ್ಯಾಂಡ್‌ವಿಡ್ತ್, ಕಂಪ್ಯೂಟಿಂಗ್ ಶಕ್ತಿ, ಅಥವಾ ಸಂಗ್ರಹಣಾ ಸ್ಥಳದ ಅವಶ್ಯಕತೆ ಇದ್ದರೂ ಅವು ತಮ್ಮ ಅಗತ್ಯಗಳನ್ನು ಮತ್ತು ಬಜೆಟ್‌ನ ಆಧಾರದ ಮೇಲೆ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಮೇಲಕ್ಕೆ ಅಥವಾ ಕೆಳಕ್ಕೆ ಅಳೆಯಬಹುದು.

- **ಭದ್ರತೆ** (Security): ಡಾಟಾ ಭದ್ರತೆಯು ಇವತ್ತಿನ ವ್ಯವಹಾರಗಳ ಪ್ರಮುಖ ಕಾಳಜಿಯಾಗಿದೆ. ಕ್ಲೌಡ್ ಮಾರಾಟಗಾರರು ಸುಧಾರಿತ ಭದ್ರತಾ ವೈಶಿಷ್ಟ್ಯಗಳಾದ ದೃಢೀಕರಣ, ಪ್ರವೇಶ ನಿರ್ವಹಣೆ, ಡಾಟಾ ಎನ್‌ಕ್ರಿಪ್ಷನ್ ಮುಂತಾದವುಗಳನ್ನು ಒದಗಿಸುತ್ತಾರೆ. ಕ್ಲೌಡ್‌ನಲ್ಲಿರುವ ಸೂಕ್ಷ್ಮ ಡಾಟಾವನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ನಿರ್ವಹಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ.

- **ಚಲನಶೀಲತೆ** (Mobility): ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್, ಇಂಟರ್‌ನೆಟ್ ಬಳಸಿ ಬಳಕೆದಾರರಿಗೆ ಕಾರ್ಪೋರೇಟ್ ಡಾಟಾವನ್ನು ಯಾವುದೇ ಸಾಧನದಿಂದಲೂ, ಎಲ್ಲಿಂದಾದರೂ, ಮತ್ತು ಯಾವುದೇ ಸಮಯದಲ್ಲಿಯೂ ಪಡೆದುಕೊಳ್ಳಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಮಾಹಿತಿಯ ಅನುಕೂಲಕರ ಲಭ್ಯತೆಯೊಂದಿಗೆ ನೌಕರರು ಸಹ ಉತ್ಪಾದಕರಾಗಿ ಉಳಿಯುತ್ತಾರೆ.

- **ಹೆಚ್ಚಿನ ಸಹಯೋಗ** (Increased collaboration) : ಕ್ಲೌಡ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು - ವ್ಯಾಪಾರಗಳಿಗೆ ಮನಬಂದಂತೆ ಸಂವಹನ ನಡೆಸಲು ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಸುರಕ್ಷಿತವಾಗಿ ಪಡೆದುಕೊಳ್ಳಲು ಮತ್ತು ಹಂಚಿಕೊಳ್ಳಲು, ಸರಳವಾಗಿ ಮತ್ತು ಹೋರಾಟ-ಮುಕ್ತ ಸಹಯೋಗವನ್ನು ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತವೆ. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಅನೇಕ ಬಳಕೆದಾರರಿಗೆ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಎಡಿಟ್ ಮಾಡಲು ಅಥವಾ ಫೈಲ್‌ಗಳ ಮೇಲೆ ಕೆಲಸ ಮಾಡಲು ಏಕಕಾಲದಲ್ಲಿ ಮತ್ತು ಪಾರದರ್ಶಕ ರೀತಿಯಲ್ಲಿ ಅಧಿಕಾರ ನೀಡುತ್ತದೆ.

- **ವಿಪತ್ತು ಚೇತರಿಕೆ** (Disaster recovery) : ಡಾಟಾ ನಷ್ಟ ಮತ್ತು ಸ್ಥಗಿತ ಕಾಲವು ಯಾವುದೇ ಗಾತ್ರದ ವ್ಯವಹಾರಗಳಿಗೆ ಸರಿಪಡಿಸಲಾಗದ ಹಾನಿಯನ್ನು ಉಂಟುಮಾಡುತ್ತದೆ. ಹೆಚ್ಚಿನ ಕ್ಲೌಡ್ ಮಾರಾಟಗಾರರು ಹಾರ್ಡ್‌ವೇರ್/ಸಾಫ್ಟ್‌ವೇರ್ ವೈಫಲ್ಯ, ನೈಸರ್ಗಿಕ ವಿಕೋಪಗಳು ಮತ್ತು ವಿದ್ಯುತ್ ಕಡಿತದಂತಹ ಅನಿರೀಕ್ಷಿತ ವಿಚಿದ್ರಕಾರಿ ಘಟನೆಗಳನ್ನು ತಡೆದುಕೊಳ್ಳಲು ಸುಸಜ್ಜಿತರಾಗಿರುತ್ತಾರೆ. ಇದು ಹೆಚ್ಚಿನ ಅಪ್ಲಿಕೇಷನ್ ಲಭ್ಯತೆಯನ್ನು ಮತ್ತು ವ್ಯಾಪಾರ ನಿರಂತರತೆಯನ್ನು ಖಚಿತಪಡಿಸುತ್ತದ.

- **ಸ್ವಯಂಚಾಲಿತ ನವೀಕರಣಗಳು** (Automatic updates): ಸಂಸ್ಥೆಯಾದ್ಯಂತ ಸಾಫ್ಟ್‌ವೇರ್ ನವೀಕರಣಗಳನ್ನು ಮ್ಯಾನುಯಲ್ ಆಗಿ ಮಾಡುವುದು IT ಸಿಬ್ಬಂದಿಯ ಬಹಳಷ್ಟು ಮೌಲ್ಯಯುತವಾದ ಸಮಯವನ್ನು ತೆಗೆದುಕೊಳ್ಳಬಹುದು. ಆದರೆ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವಾ ಪೂರೈಕೆದಾರರು ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ನಿಯಮಿತವಾಗಿ ರಿಫ್ರೆಶ್ ಮತ್ತು ನವೀಕರಿಸಲಾಗುತ್ತದೆ. ಇದು ವ್ಯವಹಾರಗಳಿಗೆ ನವೀಕರಣಗೊಂಡ ಸಾಫ್ಟ್‌ವೇರ್ ಆವೃತ್ತಿಗಳನ್ನು, ಇತ್ತೀಚಿನ ಸರ್ವರ್‌ಗಳು ಮತ್ತು ನವೀಕರಣಗೊಂಡ ಸಂಸ್ಕರಣಾ ಶಕ್ತಿಯನ್ನು ಒದಗಿಸಲು ಇತ್ತೀಚಿನ ತಂತ್ರಜ್ಞಾನಗಳನ್ನು ಬಳಸುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಅನಾನುಕೂಲಗಳು ಯಾವವು? (What are the disadvantages of cloud computing?)

ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದರಿಂದ ಅನುಕೂಲಗಳು ಅಪಾರವಾಗಿವೆ. ಅದಾಗ್ಯೂ, ಕ್ಲೌಡ್‌ಗೆ ಪರಿವರ್ತನೆ ಮಾಡಲು ನಿರ್ಧರಿಸುವ ಮೊದಲು ಕಂಪನಿಗಳು ಅದರ ಕೆಲವು ಅನಾನುಕೂಲತೆಗಳನ್ನು ತಿಳಿದಿರಬೇಕು. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ ಐದು ಮುಖ್ಯ ಅನಾನುಕೂಲತೆಗಳನ್ನು ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ.

1 **ಸ್ಥಗಿತ ಸಮಯ** (Downtime): ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸಿಸ್ಟಮ್‌ಗಳು ಇಂಟರ್‌ನೆಟ್ ಮೇಲೆ ಸಂಪೂರ್ಣವಾಗಿ ಅವಲಂಬಿತವಾಗಿವೆ. ಸಕ್ರೀಯ ಇಂಟರ್‌ನೆಟ್ ಸಂಪರ್ಕವಿಲ್ಲದೆ ವ್ಯವಹಾರಗಳು ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಹೋಸ್ಟ್ ಮಾಡಲಾದ ಡಾಟಾ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಸಾಧ್ಯವಿಲ್ಲ. ಗೂಗಲ್ 2020 ರಲ್ಲಿ ಮೂರು ತೀವ್ರ ನಿಲುಗಡೆಗಳನ್ನು ಅನುಭವಿಸಿತು. ಇದು ಜಗತ್ತಿನಾದ್ಯಂತ ಅದರ ಸೇವೆಗಳು ಮತ್ತು ಬಳಕೆದಾರರ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರಿತು.

2 **ವೆಂಡರ್ ಲಾಕ್**- ಇನ್ (Vendor lock-in): ಕಂಪನಿಯ ಕೆಲಸದ ಹೊರೆಗಳನ್ನು ಮತ್ತು ಸೇವೆಗಳನ್ನು, ಒಂದು ಕ್ಲೌಡ್ ಪೂರೈಕೆದಾರರಿಂದ ಇನ್ನೊಂದಕ್ಕೆ ಸ್ಥಳಾಂತರಿಸುವುದು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನಲ್ಲಿ ಮುಖ್ಯ ಸವಾಲಾಗಿದೆ. ಕ್ಲೌಡ್ ಪರಿಸರಗಳ ನಡುವಿನ ವ್ಯತ್ಯಾಸಗಳು ಹೊಂದಾಣಿಕೆಯ ಅಥವಾ ಏಕೀಕರಣದ ಸಮಸ್ಯೆಗಳಿಗೆ ಕಾರಣವಾಗಬಹುದು. ಪ್ರಿವರ್ತನೆಯನ್ನು ಸರಿಯಾಗಿ ನಿರ್ವಹಿಸದಿದ್ದರೆ, ಅದು ಸಂಸ್ಥೆಯ ಡಾಟಾವನ್ನು ಬಹಿರಂಗಪಡಿಸುವ ಅನಗತ್ಯ ಭದ್ರತಾ ದೋಷಗಳಿಗೆ ಕಾರಣವಾಗಬಹುದು.

3 **ಸೀಮಿತ ನಿಯಂತ್ರಣ** (Limited control) : ಕ್ಲೌಡ್ ಮೂಲಸೌಕರ್ಯವು ಸಂಪೂರ್ಣವಾಗಿ ಕ್ಲೌಡ್ ಮಾರಾಟಗಾರರ ಒಡೆತನದಲ್ಲಿದೆ ಮತ್ತು ಅವರಿಂದ ನಿರ್ವಹಿಸಲ್ಪಡುತ್ತದೆ. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಸೇವೆಗಳನ್ನು ಬಳಸುವ ವ್ಯವಹಾರಗಳು ತಮ್ಮ ಡಾಟಾ, ಅಪ್ಲಿಕೇಷನ್‌ಗಳು, ಮತ್ತು ಸೇವೆಗಳ ಮೇಲೆ ಸೀಮಿತ ನಿಯಂತ್ರಣವನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಆದ್ದರಿಂದ ಸರಿಯಾದ ಅಂತಿಮ ಬಳಕೆದಾರರ ಪರವಾನಗಿ ಒಪ್ಪಂದವನ್ನು (EULA) ಹೊಂದಿರುವುದು ಮುಖ್ಯವಾಗುತ್ತದೆ. ಇದು ಒಂದು ಕ್ಲೌಡ್ ಮೂಲಸೌಕರ್ಯದಲ್ಲಿ ಒಂದು ವ್ಯಾಪಾರವು ಏನನ್ನು ಮಾಡಬಹುದು ಮತ್ತು ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ ಎಂದು ತಿಳಿಸುತ್ತದೆ.

4 **ಭದ್ರತೆ** (Security): ಒಂದು ಕಂಪನಿಯು ಸೂಕ್ಷ್ಮ ಡಾಟಾವನ್ನು ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸುವುದು ಭದ್ರತೆಯ ಮುಖ್ಯ ಕಾಳಜಿಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಕ್ಲೌಡ್ ಸೇವಾ ಪೂರೈಕೆದಾರರು ಮುಂದುವರಿದ ಭದ್ರತಾ ಕ್ರಮಗಳನ್ನು ಅಳವಡಿಸಿದರೂ, ರಿಮೋಟ್ ಸರ್ವರ್‌ಗಳಲ್ಲಿ ಗೌಪ್ಯ ಫೈಲ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸುವುದು ಭದ್ರತಾ ಅಪಾಯಗಳನ್ನು ತೆರೆಯುತ್ತದೆ. ಒಂದು ಸಂಸ್ಥೆಯು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಮಾದರಿಯನ್ನು ಅಳವಡಿಸಿಕೊಂಡಾಗ IT ಭದ್ರತಾ ಜವಾಬ್ದಾರಿಯನ್ನು ಕ್ಲೌಡ್ ಮಾರಾಟಗಾರರು ಮತ್ತು ಬಳಕೆದಾರರ ನಡುವೆ ಹಂಚಿಕೊಳ್ಳಲಾಗುತ್ತದೆ. ಅದರಂತೆ ಪ್ರತಿಯೊಂದು ಪಕ್ಷವು ತಾವು ನಿಯಂತ್ರಿಸುವ ಆಸ್ತಿಗಳು, ಪ್ರಕ್ರಿಯೆಗಳು ಮತ್ತು ಕಾರ್ಯಗಳಿಗೆ ಜವಾಬ್ದಾರರಾಗಿರುತ್ತಾರೆ.

© NIMI NOT TO BE REPUBLISHED

5 **ಡಾಟಾ ನಷ್ಟ ಅಥವಾ ಕಳ್ಳತನ** (Data loss or theft): ವರ್ಚುವಲ್ ಡಾಟಾ ಸೆಂಟರ್‌ಗಳಲ್ಲಿ ನಿರ್ಣಾಯಕ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸುವುದು ಡಾಟಾ ನಷ್ಟದಂತಹ ವಿವಿಧ ಅಪಾಯಗಳಿಗೆ ಕಾರಣವಾಗಬಹುದು. ಉದಾಹರಣೆಗೆ, ಕ್ಲೌಡ್‌ನ ತಪ್ಪು ಸಂರಚನೆ, ಮಾಹಿತಿ ಕಳ್ಳತನ, ಭದ್ರತಾ ಉಲ್ಲಂಘನೆ, ಕದ್ದ ರುಜುವಾತುಗಳು, ಇತ್ಯಾದಿಗಳು. ಮೇಲಾಗಿ, ಕ್ಲೌಡ್ ಸೇವಾ ಪೂರೈಕೆದಾರರು ಉದಾಹರಣೆಗೆ ಮೈಕ್ರೋಸಾಫ್ಟ್ ಮತ್ತು ಗೂಗಲ್ ಮುಂತಾದವರು ಅಪ್ಲಿಕೇಶನ್ ಲಭ್ಯತೆಗೆ ಮತ್ತು ಎಲ್ಲವನ್ನೂ ಒಳಗೊಳ್ಳುವದಕ್ಕಾಗಿ ಹಂಚಿಕೆ ಜವಾಬ್ದಾರಿಯ ಮಾದರಿಯನ್ನು ಅನುಸರಿಸುತ್ತಾರೆ. ಆದರೆ ಗ್ರಾಹಕರು ಅಪ್ಲಿಕೇಶನ್ ಡಾಟಾ, ಆಡಳಿತ ಮತ್ತು ಬಳಕೆದಾರರ ನಿರ್ವಹಣೆಯ ಜವಾಬ್ದಾರಿಯನ್ನು ಉಳಿಸಿಕೊಳ್ಳುತ್ತಾರೆ.

ಸ್ಪ್ಯಾನಿಂಗ್ ಬ್ಯಾಕಪ್‌ನೊಂದಿಗೆ SaaS ಡಾಟಾ ರಕ್ಷಣೆಯನ್ನು ಸುಧಾರಿಸಿ. (Improve SaaS data protection with Spanning Backup)

ಸ್ಟ್ಯಾಟಿಸ್ಟಾ ಪ್ರಕಾರ, 2021 ರ ಹೊತ್ತಿಗೆ ಸುಮಾರು 50% ಕಾರ್ಪೊರೇಟ್ ಡಾಟಾವನ್ನು ಕ್ಲೌಡ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾಗುತ್ತದೆ. ವ್ಯಾಪಾರಗಳು ತಮ್ಮ ಕ್ಲೌಡ್ ಸೇವಾ ಪೂರೈಕೆದಾರರನ್ನು ಜಾಗತಿಕವಾಗಿ ನಂಬುವಂತೆ ಡಾಟಾ ಸೂಚಿಸುತ್ತದೆ. ಒಂದು ಕಂಪನಿಯು ಯಾವ ಕ್ಲೌಡ್ ಮಾರಾಟಗಾರರನ್ನು ಆಯ್ಕೆ ಮಾಡುತ್ತದೆ ಎಂಬುದನ್ನು ಲೆಕ್ಕಿಸದೇ, ಅವರು ಕ್ಲೌಡ್‌ನ ಲಾಭಗಳು ಆಧಾರವಾಗಿರುವ ಭದ್ರತಾ ಅಪಾಯಗಳನ್ನು ಮೀರುವುದಿಲ್ಲ ಎಂಬುದಕ್ಕೆ ಜಾಗರೂಕರಾಗಿರಬೇಕು.

Microsoft 365, Google Workspace ಮತ್ತು Salesforce ನಂತಹ SaaS ಪರಿಹಾರಗಳನ್ನು ಉಪಯೋಗಿಸುವ ವ್ಯಾಪಾರಗಳು ಪ್ರತಿದಿನ ಡಾಟಾವನ್ನು ಕಳೆದುಕೊಳ್ಳುತ್ತವೆ. ಅನೇಕ ಕಂಪನಿಗಳು SaaS ಮಾರಾಟಗಾರರು ಅವರ ಡಾಟಾವನ್ನು ರಕ್ಷಿಸುವ ಜವಾಬ್ದಾರಿಯನ್ನು ಹೊಂದಿದ್ದಾರೆ ಎಂದು ನಂಬುತ್ತಾರೆ. ಅದಾಗ್ಯೂ ಅದುಹಾಗೆ ಇರುವುದಿಲ್ಲ. SaaS ಮಾರಾಟಗಾರರು ಅಪ್ಲಿಕೇಷನ್ ಲಭ್ಯತೆಯನ್ನು ಖಚಿತಪಡಿಸುತ್ತದೆ. ಡಾಟಾ ರಕ್ಷಣೆಯು ಗ್ರಾಹಕರ ಜವಾಬ್ದಾರಿಯಾಗಿರುತ್ತದೆ.

ವ್ಯಾಪಾರಗಳಿಗೆ ವಿಶ್ವಾಸಾರ್ಹ SaaS ಬ್ಯಾಕಪ್ ಪರಿಹಾರಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಇವು ಅವರ ಅಮೂಲ್ಯವಾದ ಡಾಟಾವನ್ನು ಡಾಟಾ ನಷ್ಟಕ್ಕೆ ಕಾರಣವಾಗುವ ಫಿಶಿಂಗ್, ರ್‍ಯಾನ್‌ಸಮ್‌ವೇರ್ ಮತ್ತು ಮಾಲ್ವೇರ್ ದಾಳಿಗಳು, ಮಾನವ ದೋಷ, ದುರುದ್ದೇಶಪೂರಿತ ನಡವಳಿಕೆ ಮತ್ತು ಕಾನ್ಫಿಗರೇಷನ್ ಮತ್ತು ಸಿಂಕ್ ದೋಷಗಳಿಂದ ರಕ್ಷಿಸುತ್ತದೆ.

Microsoft 365, Google Workspace ಮತ್ತು Salesforce ಗಳಿಗಾಗಿ ಸ್ಪ್ಯಾನಿಂಗ್ ಬ್ಯಾಕಪ್ ಮತ್ತು ಎಂಡ್ ಟು ಎಂಡ್ ರಕ್ಷಣೆ ಪರಿಹಾರಗಳು, ಸ್ಥಳೀಯ ಕಾರ್ಯನಿರ್ವಹಣೆಯಲ್ಲಿನ ಅಂತರಗಳನ್ನು ಭರ್ತಿ ಮಾಡಿ ಡಾಟಾವನ್ನು ಈ ಬೆದರಿಕೆಗಳಿಂದ ಉಂಟಾಗುವ ನಷ್ಟದಿಂದ ರಕ್ಷಿಸುತ್ತವೆ. ಇದು ರಾಜಿ ಮಾಡಿಕೊಳ್ಳುವ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ ಮತ್ತು ಅಂತಿಮ ಬಳಕೆದಾರರು ಮತ್ತು ನಿರ್ವಾಹಕರು ಕೆಲವೇ ಕ್ಲಿಕ್‌ಗಳಲ್ಲಿ ಡಾಟಾವನ್ನು ತ್ವರಿತವಾಗಿ ಹುಡುಕಲು ಮತ್ತು ಅದರ ಮೂಲ ಸ್ಥಿತಿಗೆ ಮರುಸ್ಥಾಪಿಸಲು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ.

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಕಲಿಯಲು ಉತ್ತಮ ಸಂಪನ್ಮೂಲಗಳು (Best Resources to Learn Cloud Computing)

ಇವತ್ತು ಅನೇಕ ವೃತಿಪರರು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ್ನು ಅವರ ವೃತ್ತಿಯಾಗಿ ಅನುಸರಿಸುತ್ತಿದ್ದಾರೆ. ಈ ವೃತ್ತಿಜೀವನದಲ್ಲಿ ಯಶಸ್ವಿಯಾಗಲು, ಅವರು ತಮ್ಮ ಕೌಶಲ್ಯವನ್ನು ಸುಧಾರಿಸಲು ಇತ್ತೀಚಿನ ಸುದ್ದಿ, ಮತ್ತು ಟ್ರೆಂಡ್‌ಗಳೊಂದಿಗೆ ಪ್ರಸ್ತುತವಾಗಿರಬೇಕು. ಮತ್ತು ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ್ನು ಕಲಿಯಬೇಕು. ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ್ನು ಕಲಿಯಲು ಕೆಲವು ಉನ್ನತ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ:

ಆನ್‌ಲೈನ್ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಕೋರ್ಸ್‌ಗಳು (Online Cloud Computing Courses)

ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್‌ನ್ನು ಕಲಿಯಲು ಆನ್‌ಲೈನ್ ಕೋರ್ಸ್‌ಗಳು ಅತ್ಯುತ್ತಮ ಮಾರ್ಗಗಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಅನೇಕ ಆನ್‌ಲೈನ್ ಕೋರ್ಸ್‌ಗಳು ಕಲಿಯುವವರಿಗೆ ಸಹಾಯ ಮಾಡಲು ವಿಡಿಯೋಗಳು, ಲೇಖನಗಳು, ಇ ಬುಕ್‌ಗಳು, ವರದಿಗಳು, ಪ್ರಯೋಗಾಲಯಗಳು ಮತ್ತು ಮೌಲ್ಯಮಾಪನಗಳನ್ನು ಒದಗಿಸುತ್ತವೆ. ಕೆಲವು ಅತ್ಯುತ್ತಮ ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂಗ್ ಆನ್‌ಲೈನ್ ಕೋರ್ಸ್‌ಗಳನ್ನು ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ.

- Introduction to Cloud Computing on Amazon AWS for Beginners on Udemy
- Introduction to Cloud Computing on Coursera
- Cloud Computing with AWS Training on Internshala
- Cloud Computing Basics: Enhance your career as Cloud Engineer on Udemy
- Introduction to Cloud Computing by Microsoft on edX
- Cloud Computing by NPTEL
- Cloud Computing Basics (Cloud 101) on Coursera
- TOTAL: Cloud Computing / CompTIA Cloud+ Cert. (CV0-002) on Udemy

© NIMI NOT TO BE REPUBLISHED

ಕೋಪಾ (COPA) - ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವುದು ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿ ಜೀವನಚಕ್ರವನ್ನು ನಿರ್ವಹಿಸುವುದು

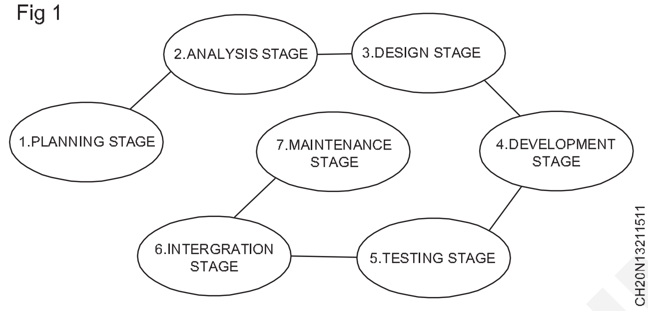
ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿ ಜೀವನಚಕ್ರದ ಹಂತಗಳನ್ನು ಗುರುತಿಸಿ (Identify Phases of the Application Development Life Cycle)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿ ಜೀವನಚಕ್ರದ ಹಂತಗಳನ್ನು ಗುರುತಿಸಿ**
- SDLC ಯ ಪ್ರಯೋಜನಗಳು.

ಅಪ್ಲಿಕೇಷನ್ ಅಭಿವೃದ್ಧಿ ಜೀವನಚಕ್ರದ ಹಂತಗಳನ್ನು ಗುರುತಿಸಿ (Fig 1)

Fig 1



ವ್ಯವಸ್ಥೆ (System)

ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿ ಪ್ರಕ್ರಿಯೆಯು ಸಾಮಾನ್ಯವಾಗಿ ದೀರ್ಘವಾಗಿರುತ್ತದೆ ಮತ್ತು ಬೇಸರದ ಪ್ರಕ್ರಿಯೆಯಾಗಿರುತ್ತದೆ. ಆದರೆ ಪ್ರಾಜೆಕ್ಟ್ ಮ್ಯಾನೇಜರ್‌ಗಳು ಮತ್ತು ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರು ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿ ಜೀವನ ಚಕ್ರವನ್ನು ಔಟ್‌ಲೈನ್, ವಿನ್ಯಾಸ, ಅಭಿವೃದ್ಧಿ, ಪರೀಕ್ಷೆ ಮತ್ತು ಅಂತಿಮವಾಗಿ ನಿಯೋಜಿಸಿ ಮಾಹಿತಿ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಅಥವಾ ಸಾಫ್ಟವೇರ್ ಉತ್ಪನ್ನಗಳನ್ನು ಹೆಚ್ಚಿನ ಕ್ರಮಬದ್ಧತೆ, ದಕ್ಷತೆ ಮತ್ತು ಒಟ್ಟಾರೆ ಗುಣಮಟ್ಟದಿಂದ ನಿಯಂತ್ರಿಸುತ್ತಾರೆ.

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ ಎಂದರೇನು? (What is System Development Life Cycle?)

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ (SDLC) ಒಂದು ಯೋಜನಾ ನಿರ್ವಹಣೆ ಮಾದರಿಯಾಗಿದೆ. ಇದು ಒಂದು ಯೋಜನೆಯನ್ನು ಅದರ ಆರಂಭಿಕ ಕಲ್ಪನೆಯಿಂದ ಅಥವಾ ಪರಿಕಲ್ಪನೆಯಿಂದ ನಿಯೋಜನೆಯವರೆಗೆ ಮತ್ತು ನಂತರ ನಿರ್ವಹಣೆಯವರೆಗಿನ ವಿವಿಧ ಹಂತಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ US **ಗೈಡ್** (System Development Life Cycle US Guide (Fig 2))

ಈ ಗೈಡ್‌ನಲ್ಲಿ, ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ ಬಗ್ಗೆ ಅದರ ಎಲ್ಲ ಹಂತಗಳನ್ನು ಸೇರಿದಂತೆ, ನೀವು ತಿಳಿದುಕೊಳ್ಳಬೇಕಾದ ಎಲ್ಲವನ್ನೂ ನಾವು ತಿಳಿಸುತ್ತೇವೆ. ನಾವು ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರ ಪಾತ್ರಗಳ ಬಗ್ಗೆಯೂ ತಿಳಿಸುತ್ತೇವೆ ಮತ್ತು SDLC ಯನ್ನು ಅಳವಡಿಸಿಕೊಳ್ಳುವ ಮೂಲಕ ನಿಮ್ಮ ಯೋಜನೆಯು ಪಡೆಯಬಹುದಾದ ಲಾಭಗಳ ಬಗ್ಗೆಯೂ ತಿಳಿಸುತ್ತೇವೆ.

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ 7 ಹಂತಗಳು

Fig 2

ಆಧುನಿಕ ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ ಏಳು ಪ್ರಾಥಮಿಕ ಹಂತಗಳಿವೆ. ಅವು ಈ ಕೆಳಗಿನಂತಿವೆ :

- ಯೋಜನಾ ಹಂತ (Planning Stage)
- ವಿಶ್ಲೇಷಣೆಯ ಹಂತದ ಕಾರ್ಯಸಾಧ್ಯತೆ ಅಥವಾ ಅವಶ್ಯಕತೆಗಳು (Feasibility or Requirements of Analysis Stage)
- ವಿನ್ಯಾಸ ಮತ್ತು ಪ್ರೊಟೋಟೈಪಿಂಗ್ ಹಂತ (Design and Prototyping Stage)
- ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿ ಹಂತ (Software Development Stage)
- ಸಾಫ್ಟ್‌ವೇರ್ ಪರೀಕ್ಷಾ ಹಂತ (Software Testing Stage)
- ಅನುಷ್ಠಾನ ಮತ್ತು ಏಕೀಕರಣ (Implementation and Integration)
- ಕಾರ್ಯಾಚರಣೆಗಳು ಮತ್ತು ನಿರ್ವಹಣಾ ಹಂತ (Operations and Maintenance Stage)

ಈಗ ಪ್ರತಿಯೊಂದು ಹಂತವನ್ನು ಪ್ರತ್ಯೇಕವಾಗಿ ನೋಡೋಣ.

© NIMI NOT TO BE REPUBLISHED

ಯೋಜನಾ ಹಂತ (Planning Stage)

ನಾವು ಯೋಜನಾ ಹಂತವನ್ನು ಪ್ರಾರಂಭಿಸುವ ಮೊದಲು, ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳುವುದು ಉತ್ತಮವಾಗಿರುತ್ತದೆ.

ಯೋಜನಾ ಹಂತ (ಕಾರ್ಯಸಾಧ್ಯತೆಯ ಹಂತ ಎಂದೂ ಕರೆಯುತ್ತಾರೆ) ಇದು ಅದರ ಹೆಸರೇ ಹೇಳುವಂತೆ ಇದರಲ್ಲಿ ಡೆವಲಪರ್‌ಗಳು ಮುಂಬರುವ ಯೋಜನೆಗೆ ಯೋಜನೆ ಮಾಡುತ್ತಾರೆ.

ಇದು ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಯಾವುದೇ ಸಿಸ್ಟಮ್‌ನ ಸಮಸ್ಯೆಯನ್ನು ಮತ್ತು ವ್ಯಾಪ್ತಿಯನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಅದರ ಹೊಸ ಸಿಸ್ಟಮ್‌ಗಳ ಉದ್ದೇಶಗಳನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ.

ಮುಂಬರುವ ಡೆವಲಪ್‌ಮೆಂಟ್ ಸೈಕಲ್‌ಗೆ ಪರಿಣಾಮಕಾರಿ ರೂಪುರೇಷೆಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಮೂಲಕ, ಸಮಸ್ಯೆಗಳು ಅಭಿವೃದ್ಧಿಯ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ಮೊದಲು, ಅವರು ಸೈದ್ಧಾಂತಿಕವಾಗಿ ಸಮಸ್ಯೆಗಳನ್ನು ಕಂಡುಹಿಡಿಯುತ್ತಾರೆ.

ಅವರು ತಮ್ಮ ಯೋಜನೆಯನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಅಗತ್ಯವಿರುವ ಹಣ ಮತ್ತು ಸಂಪನ್ಮೂಲಗಳನ್ನು ಸುರಕ್ಷಿತವಾಗಿರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ಅತ್ಯಂತ ಮುಖ್ಯವಾಗಿ, ಯೋಜನಾ ಹಂತವು ಯೋಜನೆಯ ವೇಳಾಪಟ್ಟಿಯನ್ನು ಹೊಂದಿಸುತ್ತದೆ. ಅಭಿವೃದ್ಧಿಯು, ನಿರ್ದಿಷ್ಟ ಸಮಯದಲ್ಲಿ ಮಾರುಕಟ್ಟೆಗೆ ಕಳುಹಿಸಬೇಕಾದ ವಾಣಿಜ್ಯ ಉತ್ಪನ್ನದ ಸಲುವಾಗಿ ಇದ್ದರೆ ಇದು ಅತ್ಯಂತ ಪ್ರಾಮುಖ್ಯತೆಯನ್ನು ಪಡೆಯುತ್ತದೆ.

ವಿಶ್ಲೇಷಣೆಯ ಹಂತ (Analysis Stage)

ವಿಶ್ಲೇಷಣೆಯ ಹಂತವು ಹೊಸ ಸಿಸ್ಟಮ್‌ಗೆ ಅವಶ್ಯಕವಿರುವ ಎಲ್ಲಾ ನಿರ್ದಿಷ್ಟ ವಿವರಗಳನ್ನು ಸಂಗ್ರಹಿಸುವುದನ್ನು ಮತ್ತು ಮೂಲಮಾದರಿಯ ಮೊದಲ ಕಲ್ಪನೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಡೆವಲಪರ್‌ಗಳು ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಹುದು :

- ಯಾವುದೇ ಮೂಲಮಾದರಿಯ ಸಿಸ್ಟಮ್ ಅವಶ್ಯಕತೆಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವುದು.
- ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಮೂಲಮಾದರಿಯ ಪರ್ಯಾಯಗಳನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡುವುದು
- ಅಂತಿಮ ಬಳಕೆದಾರರ ಅವಶ್ಯಕತೆಗಳನ್ನು ನಿರ್ಧರಿಸಲು ಸಂಶೋಧನೆ ಮತ್ತು ವಿಶ್ಲೇಷಣೆ ಮಾಡುವುದು

ಇದಲ್ಲದೇ ಡೆವಲಪರ್‌ಗಳು ಸಾಫ್ಟ್‌ವೇರ್ ಅಗತ್ಯದ ವಿವರಗಳನ್ನು ಅಥವಾ SRS ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ರಚಿಸುತ್ತಾರೆ.

ಇದು ಅವರು ನಿರ್ಮಿಸಲು ಯೋಜನೆ ಮಾಡಿದ ಸಿಸ್ಟಮ್‌ನ ಸಾಫ್ಟ್‌ವೇರ್, ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಅವಶ್ಯಕತೆಗಳ ಎಲ್ಲಾ ವಿವರಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಅವರು ಹೆಚ್ಚಿನ ಹಣವನ್ನು ಅಥವಾ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಅದೇ ಸ್ಥಳದಲ್ಲಿ ಇತರ ಅಭಿವೃದ್ಧಿ ತಂಡದೊಂದಿಗೆ ಕೆಲಸ ಮಾಡುವಾಗ ತೆಗೆದುಕೊಳ್ಳುವುದನ್ನು ತಡೆಯುತ್ತದೆ.

ವಿನ್ಯಾಸ ಹಂತ (Design Stage)

ವಿನ್ಯಾಸ ಹಂತವು ಮುಖ್ಯ ಡೆವಲಪರ್ ಹಂತಕ್ಕೆ ಪೂರ್ವಗಾಮಿಯಾಗಿ ಅವಶ್ಯಕವಾಗಿದೆ.

ಡೆವಲಪರ್‌ಗಳು ಮೊದಲು ಒಟ್ಟಾರೆ ಅಪ್ಲಿಕೇಷನ್, ಅದರಂತಹ ನಿರ್ದಿಷ್ಟ ಅಂಶಗಳ ವಿವರಗಳ ರೂಪುರೇಷೆಯನ್ನು ನೀಡುತ್ತಾರೆ. ಜೊತೆಗೆ :

- ಬಳಕೆದಾರರ ಇಂಟರ್‌ಫೇಸ್ (User interfaces)
- ಸಿಸ್ಟಮ್ ಇಂಟರ್‌ಫೇಸ್ (System interfaces)
- ನೆಟ್‌ವರ್ಕ್ ಮತ್ತು ನೆಟ್‌ವರ್ಕ್ ಅಗತ್ಯಗಳು (Network and network requirements)
- ಡಾಟಾಬೇಸ್‌ಗಳು (Databases)

ಅವರು ರಚಿಸಿದ SRS ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಅವರು ಹೆಚ್ಚಿನ ತಾರ್ಕಿಕ ರಚನೆಯಲ್ಲಿ ಪರಿವರ್ತಿಸುತ್ತಾರೆ. ಅದನ್ನು ನಂತರದಲ್ಲಿ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಲ್ಲಿ ಅನುಷ್ಠಾನಗೊಳಿಸಬಹುದು. ಕಾರ್ಯಾಚರಣೆ, ತರಬೇತಿ, ಮತ್ತು ನಿರ್ವಹಣೆ ಯೋಜನೆ ಎಲ್ಲವುಗಳನ್ನೂ ರೂಪಿಸಲಾಗುವುದು. ಇದರಿಂದ ಎಲ್ಲ ಡೆವಲಪರ್‌ಗಳು ಸೈಕಲ್‌ನ್ನು ಮುಂದಕ್ಕೆ ಚಲಿಸಲು ಪ್ರತಿ ಹಂತದಲ್ಲಿ ಅವರು ಏನು ಮಾಡಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿದುಕೊಳ್ಳುತ್ತಾರೆ.

ಒಮ್ಮೆ ಪೂರ್ಣಗೊಂಡ ನಂತರ ಡೆವಲಪ್‌ಮೆಂಟ್ ಮ್ಯಾನೇಜರ್‌ಗಳು SDLCಯ ಮುಂದಿನ ಹಂತಗಳಿಗೂ ಉಪಯೋಗಿಸಬಹುದಾದಂತಹ ವಿನ್ಯಾಸ ಡಾಕ್ಯುಮೆಂಟ್‌ನ್ನು ಸಿದ್ಧಪಡಿಸುತ್ತಾರೆ.

ಅಭಿವೃದ್ಧಿ ಹಂತ (Development Stage (Fig 3))

ಅಭಿವೃದ್ಧಿ ಹಂತದಲ್ಲಿ ವಾಸ್ತವವಾಗಿ ಡೆವಲಪರ್‌ಗಳು ಕೋಡ್‌ನ್ನು ಬರೆಯುತ್ತಾರೆ ಮತ್ತು ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ನಿರ್ಮಿಸುತ್ತಾರೆ. ಹಿಂದಿನ ವಿನ್ಯಾಸ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳ ಪ್ರಕಾರ ಮತ್ತು ವಿವರಣೆಗಳ ರೂಪುರೇಷೆಗಳ ಪ್ರಕಾರ ನಿರ್ಮಿಸುತ್ತಾರೆ.

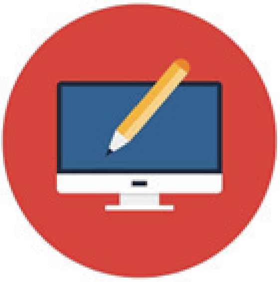
ಇಲ್ಲಿಯೇ Static Application Security Testing ಅಥವಾ SAST ಸಾಧನಗಳು ಕಾರ್ಯರೂಪಕ್ಕೆ ಬರುತ್ತವೆ.

ಡಾಕ್ಯುಮೆಂಟ್‌ನ ವಿನ್ಯಾಸದ ವಿವರಣೆಗಳ ಪ್ರಕಾರ ಉತ್ಪನ್ನದ ಪ್ರೊಗ್ರಾಂ ಕೋಡ್‌ನ್ನು ನಿರ್ಮಿಸಲಾಗುತ್ತದೆ. ಸಿದ್ಧಾಂತದಲ್ಲಿ, ಎಲ್ಲಾ

NOT TO BE REPUBLISHED © NIMI

ಪೂರ್ವ ಯೋಜನೆಗಳು ಮತ್ತು ರೂಪುರೇಷೆಗಳು ನಿಜವಾದ ಅಭಿವೃದ್ಧಿ ಹಂತವನ್ನು ತುಲನಾತ್ಮಕವಾಗಿ ನೇರವಾಗಿ ಮಾಡಬೇಕು.

Fig 3



ಡೆವಲಪರ್‌ಗಳು ಒಂದು ಸಂಸ್ಥೆಯು ವ್ಯಾಖ್ಯಾನಿಸಿದಂತೆಯೆ ಯಾವುದೇ ಕೊಡಿಂಗ್ ಮಾರ್ಗಸೂಚಿಗಳನ್ನು ಅನುಸರಿಸುತ್ತಾರೆ ಮತ್ತು ವಿವಿಧ ಸಾಧನಗಳಾದ ಕಂಪೈಲರ್‌ಗಳು, ಡಿಬಗ್ಗರ್‌ಗಳು ಮತ್ತು ಇಂಟರ್‌ಪ್ರಿಟರ್‌ಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತಾರೆ.

ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳಾದ C++, PHP ಮತ್ತು ಇನ್ನೂ ಹೆಚ್ಚಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರಬಹುದು. ಡೆವಲಪರ್‌ಗಳು ಯೋಜನೆಯ ವಿವರಗಳು ಮತ್ತು ಅವಶ್ಯಕತೆಗಳ ಅನುಸಾರ ಸರಿಯಾದ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಕೋಡ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳುತ್ತಾರೆ.

ಪರೀಕ್ಷೆ ಹಂತ (Testing Stage (Fig 4))

Fig 4

ಸಾಫ್ಟ್‌ವೇರನ್ನು ನಿರ್ಮಿಸುವುದು ಅಂತ್ಯವಲ್ಲ.

ಈಗ ಅದರಲ್ಲಿ ಯಾವುದೂ ದೋಷಗಳಿಲ್ಲ ಮತ್ತು ಅಂತಿಮ ಬಳಕೆದಾರರ ಅನುಭವವು ಯಾವುದೇ ಹಂತದಲ್ಲಿ ಋಣಾತ್ಮಕ ಪರಿಣಾಮ ಬೀರಿಲ್ಲ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಅದನ್ನು ಪರೀಕ್ಷಿಸಬೇಕಾಗುತ್ತದೆ

ಪರೀಕ್ಷೆಯ ಹಂತದಲ್ಲಿ ಡೆವಲಪರ್‌ಗಳು ತಮ್ಮ ಸಾಫ್ಟ್‌ವೇರನ್ನು ಸೂಕ್ಷ್ಮವಾಗಿ ಪರೀಕ್ಷಿಸಿ, ಯಾವುದೇ ದೋಷಗಳು ಅಥವಾ ನ್ಯೂನ್ಯತೆಗಳನ್ನು ಪತ್ತೆ ಹಚ್ಚಬೇಕು, ಸರಿಪಡಿಸಬೇಕು ಮತ್ತು ನಂತರ ಮರುಪರೀಕ್ಷೆ ಮಾಡಬೇಕು.

ಸಾಫ್ಟ್‌ವೇರ್ ಒಟ್ಟಾರೆಯಾಗಿ SRS ಡಾಕ್ಯುಮೆಂಟ್‌ನಲ್ಲಿ ಹಿಂದೆ ವ್ಯಾಖ್ಯಾನಿಸಿದಂತೆ ಗುಣಮಟ್ಟದ ಮಾನದಂಡಗಳನ್ನು ಪೂರ್ಣಗೊಳಿಸುವುದು ಮುಖ್ಯವಾಗಿರುತ್ತದೆ.

ಡೆವಲಪರ್‌ಗಳ ಕೌಶಲ್ಯ, ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಸಂಕೀರ್ಣತೆ ಮತ್ತು ಅಂತಿಮ ಬಳಕೆದಾರರ ಅವಶ್ಯಕತೆಗಳು ಇವುಗಳ ಆಧಾರದ ಮೇಲೆ ಪರೀಕ್ಷೆಯ ಹಂತವು ಅತ್ಯಂತ ಕಡಿಮೆ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳಬಹುದುಅಥವಾ ಬಹಳ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳಬಹುದು. ಸಾಫ್ಟ್‌ವೇರ್ ಪರೀಕ್ಷೆ ಯೋಜನೆಗಳ 10 ಉತ್ತಮ ಅಭ್ಯಾಸಗಳನ್ನು ನೋಡೋಣ.

ಅನುಷ್ಠಾನ ಮತ್ತು ಏಕೀಕರಣ ಹಂತ (Implementation and Integration Stage)

ಪರೀಕ್ಷೆಯ ಹಂತದ ನಂತರ, ಸಾಫ್ಟ್‌ವೇರ್‌ನ ಎಲ್ಲ ವಿನ್ಯಾಸವು ಒಟ್ಟಿಗೆ ಬರುತ್ತದೆ. ಡೆವಲಪರ್‌ಗಳ ಪ್ರಯತ್ನಗಳ ಮೂಲಕ ವಿಭಿನ್ನ ಮಾಡ್ಯೂಲ್‌ಗಳು ಮತ್ತು ವಿನ್ಯಾಸಗಳನ್ನು ಪ್ರಾಥಮಿಕ ಮೂಲ ಕೋಡ್‌ನಲ್ಲಿ ಸಂಯೋಜಿಸಲಾಗುತ್ತದೆ. ಸಾಮಾನ್ಯವಾಗಿ ಮುಂದಿನ ದೋಷಗಳನ್ನು ಅಥವಾ ನ್ಯೂನ್ಯತೆಗಳನ್ನು ಪತ್ತೆಹಚ್ಚಲು ತರಬೇತಿ ಪರಿಸರವನ್ನು ನಿಯಂತ್ರಿಸಲಾಗುತ್ತದೆ.

ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ನ್ನು ಅದರ ಪರಿಸರದಲ್ಲಿ ಸಂಯೋಜಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅಂತಿಮವಾಗಿ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡಲಾಗುತ್ತದೆ. ಈ ಹಂತವು ಮುಗಿದ ನಂತರ ಸಾಫ್ಟ್‌ವೇರ್ ಮಾರುಕಟ್ಟೆಗೆ ಸಿದ್ಧವಾಗಿದೆ ಮತ್ತು ಅದನ್ನು ಯಾವುದೇ ಅಂತಿಮ ಬಳಕೆದಾರರಿಗೆ ಒದಗಿಸಬಹುದು.

ನಿರ್ವಹಣೆ ಹಂತ (Maintenance Stage (Fig 5))

ಸಾಫ್ಟ್‌ವೇರ್ ಮಾರುಕಟ್ಟೆಯನ್ನು ತಲುಪಿದ ನಂತರ SDLC ಅಲ್ಲಿಯೇ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ. ಡೆವಲಪರ್‌ಗಳು ಈಗ ನಿರ್ವಹಣೆ ಮೋಡ್‌ಗೆ ಹೋಗಬೇಕು ಮತ್ತು ಅಂತಿಮ ಬಳಕೆದಾರರು ವರದಿ ಮಾಡಿದ ಸಮಸ್ಯೆಗಳನ್ನು ನಿಭಾಯಿಸಲು ಅವಶ್ಯಕವಿರುವ ಯಾವುದೇ ಚಟುವಟಿಕೆಗಳನ್ನು ಅಭ್ಯಾಸ ಮಾಡಲು ಪ್ರಾರಂಭಿಸಬೇಕು.

ಇದಲ್ಲದೇ ಡೆವಲಪರ್‌ಗಳು ಸಾಫ್ಟ್‌ವೇರ್‌ನ ನಿಯೋಜನೆಯ ನಂತರ ಅದಕ್ಕೆ ಅವಶ್ಯಕವಿರುವ ಯಾವುದಾದರೂ ಬದಲಾವಣೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಜವಾಬ್ದಾರರಾಗಿರುತ್ತಾರೆ.

ಇದು ಪ್ರಾರಂಭಿಸುವ ಮೊದಲು ಸರಿಮಾಡಲು ಸಾಧ್ಯವಾಗದೇ ಉಳಿದ ದೋಷಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು ಅಥವಾ ಬಳಕೆದಾರರ ವರದಿಗಳಿಂದ ಉಂಟಾದ ಹೊಸ ಸಮಸ್ಯೆಗಳನ್ನು ಪರಿಹರಿಸುವುದನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಸಣ್ಣ ವ್ಯವಸ್ಥೆಗಳಿಗೆ ಹೋಲಿಸಿದರೆ ದೊಡ್ಡ ವ್ಯವಸ್ಥೆಗಳಿಗೆ ದೀರ್ಘವಾದ ನಿರ್ವಹಣಾ ಹಂತಗಳ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

Fig 5

© NIMI NOT TO BE REPUBLISHED

ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರ ಪಾತ್ರ (Role of System Analyst)

ಒಂದು SDLC ಯ ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರು ಕೆಲವು ರೀತಿಯಲ್ಲಿ ಇಡೀ ವ್ಯವಸ್ಥೆಗೆ ಮೇಲ್ವಿಚಾರಕರಾಗಿರುತ್ತಾರೆ. ಅವರು ಸಿಸ್ಟಮ್ ಬಗ್ಗೆ, ಅದರ ಎಲ್ಲ ಚಲಿಸುವ ಭಾಗಗಳ ಬಗ್ಗೆ ಸಂಪೂರ್ಣವಾಗಿ ತಿಳಿದಿರಬೇಕು ಮತ್ತು ಸೂಕ್ತ ನಿರ್ದೇಶನಗಳನ್ನು ನೀಡುವ ಮೂಲಕ ಮಾರ್ಗದರ್ಶನ ಮಾಡಲು ಸಹಾಯ ಮಾಡಬೇಕು.

ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರು ಹೀಗಿರಬೇಕು (The system analyst should be):

- ಯೋಜನೆಗೆ ಅವಶ್ಯಕವಿರುವ ಯಾವುದೇ ತಾಂತ್ರಿಕ ಕೌಶಲ್ಯಗಳಲ್ಲಿ ಪರಿಣಿತರಿರಬೇಕು.
- ತಮ್ಮ ತಂಡದ ಯಶಸ್ಸಿಗಾಗಿ ಆಜ್ಞಾಪಿಸಲು ಸಹಾಯ ಮಾಡುವ ಉತ್ತಮ ಸಂವಹನಕಾರರಾಗಿರಬೇಕು.
- ಡೆವಲಪ್‌ಮೆಂಟ್ ಸೈಕಲ್‌ನ ಅಭಿವೃದ್ಧಿ ಕಾರ್ಯಗಳು ಪ್ರತಿಯೊಂದು ಹಂತದಲ್ಲಿ ಸಮಯಕ್ಕೆ ಸರಿಯಾಗಿ ನಡೆಸಲು ಅವರು ಉತ್ತಮ ಯೋಜನಾಕಾರರಾಗಿರಬೇಕು.

ಹೀಗಾಗಿ ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರು ಪರಸ್ಪರ, ತಾಂತ್ರಿಕ, ನಿರ್ವಹಣೆ, ಮತ್ತು ವಿಶ್ಲೇಷಣಾತ್ಮಕ ಕೌಶಲ್ಯಗಳನ್ನು ಒಟ್ಟಾರೆಯಾಗಿ ಅಥವಾ ಸಮ ಮಿಶ್ರಣಗಳನ್ನು ಹೊಂದಿರಬೇಕು. ಅವರು SDLC ಯನ್ನು ನಿರ್ಮಿಸುವ ಅಥವಾ ಮುರಿಯುವ ಬಹುಮುಖ ವೃತ್ತಿಪರರಾಗಿರಬೇಕು.

ಅವರ ಜವಾಬ್ದಾರಿಗಳು ಒಂದು ನಿರ್ದಿಷ್ಟ ಯೋಜನೆಯ ಅಂತಿಮ ಯಶಸ್ಸಿನ ಸಲುವಾಗಿ ಸಾಕಷ್ಟು ವೈವಿಧ್ಯಮಯವಾಗಿವೆ ಮತ್ತು ಮುಖ್ಯವಾಗಿವೆ. ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರಿಂದ ಕೆಳಗಿನವುಗಳನ್ನು ನಿರೀಕ್ಷಿಸಲಾಗುತ್ತದೆ :

- ಅಂಶಗಳನ್ನು ಮತ್ತು ಮಾಹಿತಿಯನ್ನು ಒಟ್ಟುಗೂಡಿಸುವುದು.
- ಯಾವ ದೋಷಗಳಿಗೆ ಆದ್ಯತೆ ನೀಡಬೇಕು ಅಥವಾ ಯಾವ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ತೆಗೆದುಹಾಕಬೇಕು ಎನ್ನುವ ಬಗ್ಗೆ ನಿರ್ಧಾರ ಮಾಡುವುದು.
- ಪರ್ಯಾಯ ಪರಿಹಾರಗಳನ್ನು ಸೂಚಿಸುವುದು.
- ಬಳಕೆದಾರರು ಮತ್ತು ಪ್ರೊಗ್ರಾಮರ್‌ಗಳು ಇಬ್ಬರೂ ಸುಲಭವಾಗಿ ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವ ವಿಶೇಷಣಗಳನ್ನು ಬರೆಯಬೇಕು.
- ತಾರ್ಕಿಕ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಿ ಮತ್ತು ಮಾಡ್ಯುಲಾರಿಟಿಯನ್ನು ನಂತರದ ಏಕೀಕರಣಕ್ಕಾಗಿ ಇರಿಸಿಕೊಳ್ಳಬೇಕು.
- ಯೋಜನೆಯ ಗುರಿಗಳಿಗೆ ಅಗತ್ಯವಿರುವ ಫಲಿತಾಂಶದ ಸಿಸ್ಟಮ್‌ನ್ನು ಮೌಲ್ಯಮಾಪನ ಮಾಡಲು ಮತ್ತು ಮಾರ್ಪಾಡು ಮಾಡಲು ಸಾಧ್ಯವಿರಬೇಕು.
- ಬಳಕೆದಾರರ ಅವಶ್ಯಕತೆಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಮತ್ತು ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳುವ ಮೂಲಕ ಯೋಜನೆಯ ಅವಶ್ಯಕತೆಗಳು ಮತ್ತು ಗುರಿಗಳನ್ನು ಯೋಜಿಸಲು ಸಹಾಯ ಮಾಡಬೇಕು.

6 ಮೂಲ SDLC ವಿಧಾನಗಳು (6 Basic SDLC Methodologies)

ವಿಶಾಲವಾದ ಅರ್ಥದಲ್ಲಿ, ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ ಒಂದು ಯೋಜನಾ ನಿರ್ವಹಣಾ ಮಾದರಿ ಆಗಿದ್ದರೂ, ನಿರ್ದಿಷ್ಟ ಫಲಿತಾಂಶಗಳನ್ನು ಸಾಧಿಸಲು ಆರು ಹೆಚ್ಚು ನಿರ್ದಿಷ್ಟ ವಿಧಾನಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ವಾಟರ್‌ಫಾಲ್ ಮಾಡೆಲ್ (Waterfall Model (Fig 6))

ಎಲ್ಲಾ SDLC ವಿಧಾನಗಳಲ್ಲಿ, ವಾಟರ್‌ಫಾಲ್ ಮಾಡೆಲ್ ಅತ್ಯಂತ ಹಳೆಯದಾಗಿದೆ. ಇದು ಲಿನಿಯರ್ ಮತ್ತು ನೇರವಾಗಿದೆ. ಯೋಜನೆಯ ಮುಂದಿನ ಹಂತಕ್ಕೆ ಹೋಗುವ ಮೊದಲು ಒಂದು ಹಂತವನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಮುಗಿಸಲು ಡೆವಲಪ್‌ಮೆಂಟ್ ತಂಡದ ಅವಶ್ಯಕತೆಯಿರುತ್ತದೆ.

ಪ್ರತಿಯೊಂದು ಹಂತವು ಪ್ರತ್ಯೇಕ ಪ್ರಾಜೆಕ್ಟ್ ಯೋಜನೆಯನ್ನು ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಅದೇ ರೀತಿಯ ಸಮಸ್ಯೆಗಳನ್ನು ತಪ್ಪಿಸಲು ಹಿಂದಿನ ಹಂತದ ಮಾಹಿತಿಯನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ. ಅದಾಗ್ಯೂ ಇದು ಆರಂಭಿಕ ವಿಳಂಬದ ದೋಷವಾಗಿರುತ್ತದೆ ಮತ್ತು ನಂತರದ ಡೆವಲಪ್‌ಮೆಂಟ್ ತಂಡಗಳಿಗೆ ದೊಡ್ಡ ಸಮಸ್ಯೆಗಳನ್ನು ಉಂಟು ಮಾಡುತ್ತದೆ.

Fig 6

ಇಟರೇಟಿವ್ ಮಾಡೆಲ್ (Iterative Model (Fig 7)

ಇಟರೇಟಿವ್ ಮಾಡೆಲ್ ಪುನರಾವರ್ತನೆ ಮೇಲೆ ಮತ್ತು ಪುನರಾವರ್ತಿತ ಪರೀಕ್ಷೆಯ ಮೇಲೆ ಕೇಂದ್ರೀಕರಿಸುತ್ತದೆ. ಸಾಫ್ಟ್‌ವೇರ್ ಪ್ರಾಜೆಕ್ಟ್‌ನ ಹೊಸ ಆವೃತ್ತಿಗಳನ್ನು ಪ್ರತಿಯೊಂದು ಹಂತದ ಕೊನೆಯಲ್ಲಿ ಉತ್ಪಾದಿಸಲಾಗುತ್ತದೆ. ಇದು ಸಂಭಾವ್ಯ ದೋಷಗಳನ್ನು ಕಂಡುಹಿಡಿಯುತ್ತದೆ ಮತ್ತು ಡೆವಲಪರ್‌ಗಳಿಗೆ ಅಂತಿಮ ಉತ್ಪನ್ನವನ್ನು ಅದು ಮಾರುಕಟ್ಟೆಗೆ ಸಿದ್ಧವಾಗಿರುವ ಸಮಯದಲ್ಲಿ ನಿರಂತರವಾಗಿ ಸುಧಾರಿಸಲು ಅನುಮತಿಸುತ್ತದೆ.

Fig 7

© NIMI NOT TO BE REPUBLISHED

ಈ ಮಾದರಿಯ ಒಳ್ಳೆಯ ಅಂಶವೆಂದರೆ ಇದು ಡೆವಲಪರ್‌ಗಳು, ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ ಆರಂಭದಲ್ಲಿ ಪ್ರಾಜೆಕ್ಟ್‌ನ ಕೆಲಸದ ಆವೃತ್ತಿಯನ್ನು ರಚಿಸಬಹುದು. ಆದ್ದರಿಂದ ಬದಲಾವಣೆಗಳನ್ನು ಮಾಡುವುದು ಸಾಮಾನ್ಯವಾಗಿ ಕಡಿಮೆ ವೆಚ್ಚದಲ್ಲಿರುತ್ತವೆ.

ಸ್ಪೈರಲ್ ಮಾಡೆಲ್ (Spiral Model)

ಇತರ ಮಾದರಿಗಳಿಗೆ ಹೋಲಿಸಿದರೆ ಸ್ಪೈರಲ್ ಮಾಡೆಲ್ ಹೊಂದಿಕೊಳ್ಳುತ್ತವೆ. ಯೋಜನೆಗಳು ನಾಲ್ಕು ಮುಖ್ಯ ಹಂತಗಳ ಮೂಲಕ ಸುರುಳಿಯಾಕಾರದ ಚಲನೆಯಲ್ಲಿ ಮತ್ತೆ ಮತ್ತೆ ಹಾದು ಹೋಗುತ್ತವೆ.

ಇದು ದೊಡ್ಡ ಯೋಜನೆಗಳಿಗೆ ಅನುಕೂಲಕರವಾಗಿದೆ. ಅಭಿವೃದ್ಧಿಯ ತಂಡಗಳು ಅತ್ಯಂತ ಕಸ್ಟಮೈಸ್ ಮಾಡಿದ ಉತ್ಪನ್ನಗಳನ್ನು ತಯಾರಿಸುತ್ತವೆ ಮತ್ತು ಲೈಫ್ ಸೈಕಲ್‌ನ ಆರಂಭದಲ್ಲಿ ಸ್ವಿಕರಿಸಿದ ಯಾವುದೇ ಪ್ರತಿಕ್ರಿಯೆಗಳನ್ನು ಸಂಯೋಜಿಸಬಹುದು.

ವಿ - ಮಾಡೆಲ್ (V-Model)

ವಿ - ಮಾಡೆಲ್ (verification ಮತ್ತು validation ನ ಸಂಕ್ಷೇಪ್ತ ರೂಪವಾಗಿದೆ) ಇದು ವಾಟರ್ ಫಾಲ್ ಮಾಡೆಲ್‌ನ್ನು ಹೋಲುತ್ತದೆ. ಸಂಭಾವ್ಯ ದೋಷಗಳನ್ನು ಮತ್ತು ನ್ಯೂನ್ಯತೆಗಳನ್ನು ಕಂಡು ಹಿಡಿಯಲು, ಪರೀಕ್ಷಾ ಹಂತವನ್ನು ಪ್ರತಿಯೊಂದು ಅಭಿವೃದ್ಧಿ ಹಂತಕ್ಕೆ ಸಂಯೋಜಿಸಲಾಗಿದೆ.

ಇದು ಅತ್ಯಂತ ಶಿಸ್ತುಬದ್ಧವಾಗಿದೆ ಮತ್ತು ಇದಕ್ಕೆ ತೀವ್ರ ಸಮಯದ ಅವಶ್ಯಕತೆಯಿದೆ. ಆದರೆ ಸಿದ್ಧಾಂತದಲ್ಲಿ, ದೊಡ್ಡ ದೋಷಗಳನ್ನು ನಿಯಂತ್ರಣ ತಪ್ಪಿ ಸುರುಳಿಯಾಗದಂತೆ ತಡೆಯುವ ಮೂಲಕ, ವಾಟರ್‌ಫಾಲ್ ಮಾಡೆಲ್‌ನ ನ್ಯೂನ್ಯತೆಗಳನ್ನು ಬೆಳಗಿಸುತ್ತದೆ.

ಬಿಗ್ ಬ್ಯಾಂಗ್ ಮಾಡೆಲ್ (Big Bang Model)

ಬಿಗ್ ಬ್ಯಾಂಗ್ ಮಾಡೆಲ್ ಅತ್ಯಂತ ಹೊಂದಿಕೊಳ್ಳುವದಾಗಿದೆ(flexi bale) ಮತ್ತು ಕಠಿಣ ಪ್ರಕ್ರಿಯೆಯನ್ನು ಅಥವಾ ಕಾರ್ಯವಿಧಾನವನ್ನು ಅನುಸರಿಸುವುದಿಲ್ಲ. ಇದು ಹಿಂದೆ ವಿವರವಾದ ಯೋಜನೆಯನ್ನು ಸಹ ಇರಿಸುತ್ತದೆ. ಗ್ರಾಹಕರು ಅಥವಾ ಕ್ಲೈಂಟ್‌ಗಳಿಗೆ ಅವರಿಗೆ ಏನು ಬೇಕು ಎಂದು ಖಚಿತವಾಗಿರದಿದ್ದಾಗ, ಇದನ್ನು ವಿಶಾಲವಾದ ಆಲೋಚನೆಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಡೆವಲಪರ್‌ಗಳು ಕೇವಲ ಹಣ ಮತ್ತು ಸಂಪನ್ಮೂಲಗಳಿಂದ ಯೋಜನೆಯನ್ನು ಪ್ರಾರಂಭಿಸುತ್ತಾರೆ.

ಕ್ಲೈಂಟ್‌ಗಳು ಬಯಸುತ್ತಾರೆ ಎಂದು ಅರಿತುಕೊಳ್ಳುವ ಫಲಿತಾಂಶವು ಹತ್ತಿರ ಅಥವಾ ಅತ್ಯಂತ ದೂರ ಇರಬಹುದು. ಇದನ್ನು ಹೆಚ್ಚಾಗಿ ಚಿಕ್ಕ ಯೋಜನೆಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅದೇ ಕಂಪನಿಯ ಇತರ ಯೋಜನೆಗಳಿಗೆ ಪ್ರಾಯೋಗಿಕ ಲೈಫ್ ಸೈಕಲ್‌ಗಳನ್ನು ತಿಳಿಸಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. (Fig 8)

Fig 8

ಅಗೈಲ್ ಮಾಡೆಲ್ (Agile Model)

ಅಗೈಲ್ ಮಾಡೆಲ್, ನಿರ್ದಿಷ್ಟವಾಗಿ ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿ ಉದ್ಯಮದಲ್ಲಿ ಪ್ರಸಿದ್ಧವಾಗಿದೆ.

ಅಗೈಲ್ ವಿಧಾನವು, ವೇಗವಾದ ಮತ್ತು ನಡೆಯುತ್ತಿರುವ ಬಿಡುಗಡೆಯ ಸೈಕಲ್‌ಗಳಿಗೆ ಆದ್ಯತೆ ನೀಡುತ್ತದೆ. ಇದು ಸಣ್ಣ ಆದರೆ ಬಿಡುಗಡೆಗಳ ನಡುವಿನ ಹೆಚ್ಚುತ್ತಿರುವ ಬದಲಾವಣೆಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತದೆ. ಇತರ ಮಾಡೆಲ್‌ಗಳಿಗೆ ಹೋಲಿಸಿದರೆ, ಇದು ಹೆಚ್ಚಿನ ಪುನರಾವರ್ತನೆಗಳಿಗೆ ಮತ್ತು ಹೆಚ್ಚಿನ ಪರೀಕ್ಷೆಗಳಿಗೆ ಕಾರಣವಾಗುತ್ತದೆ.

ಸೈದ್ಧಾಂತಿಕವಾಗಿ, ಈ ಮಾಡೆಲ್ ತಂಡಗಳಿಗೆ ಸಣ್ಣ ಸಮಸ್ಯೆಗಳನ್ನು, ಯೋಜನೆಯ ಮುಂದಿನ ಹೆಚ್ಚು ಸಂಕೀರ್ಣ ಹಂತಗಳಲ್ಲಿ ಅವುಗಳನ್ನು ಹಾಗೇ ಬಿಡುವ ಬದಲು ಪರಿಹರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. (Fig 9)

Fig 9

SDLC ಯ ಪ್ರಯೋಜನಗಳು (Benefits of SDLC) : SDLC ಯು ಅಭಿವೃದ್ಧಿ ತಂಡಗಳಿಗೆ ಹಲವಾರು ಪ್ರಯೋಜನಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಸ್ಪಷ್ಟ ಗುರಿಯ ವಿವರಣೆಗಳು (Clear Goal Descriptions): ಡೆವಲಪರ್‌ಗಳು ತಾವು ಪೂರೈಸಬೇಕಾದ ಗುರಿಗಳ ಬಗ್ಗೆ ಸ್ಪಷ್ಟವಾಗಿ ತಿಳಿದಿದ್ದಾರೆ. ಮತ್ತು ಸರಿಯಾದ ಸಮಯದಲ್ಲಿ ಮಾಡಬೇಕಾದ ವಿತರಣೆಗಳು , ಸಮಯದ ಅಪಾಯವನ್ನು ಕಡಿಮೆ ಮಾಡುವುದು ಮತ್ತು ವ್ಯರ್ಥವಾಗುತ್ತಿರುವ ಸಂಪನ್ಮೂಲಗಳ ಬಗ್ಗೆ ತಿಳಿದಿದ್ದಾರೆ.

ಇನ್ಸ್ಟಾಲ್ ಮಾಡುವ ಮೊದಲು ಸರಿಯಾದ ಪರೀಕ್ಷೆ (Proper Testing Before Installation) : SDLC ಮಾಡೆಲ್‌ಗಳು ಪರಿಶೀಲಿಸುತ್ತವೆ ಮತ್ತು ಸಮತೋಲನವನ್ನು ಮಾಡುತ್ತವೆ. ಇದು ಎಲ್ಲ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳನ್ನು ಸೋರ್ಸ್ ಕೋಡ್‌ನಲ್ಲಿ ಇನ್‌ಸ್ಟಾಲ್ ಮಾಡುವ ಮೊದಲು ಪರೀಕ್ಷಿಸಲಾಗಿದೆ ಎಂದು ಖಚಿತಪಡಿಸುತ್ತದೆ.

ಹಂತಗಳ ಪ್ರಗತಿಯನ್ನು ಕ್ಲಿಯರ್ ಮಾಡುವುದು (Clear Stage Progression) : ಡೆವಲಪರ್‌ಗಳು ಮೊದಲ ಹಂತವನ್ನು ಪೂರ್ಣಗೊಳಿದೇ, ಮತ್ತು ಮ್ಯಾನೇಜರ್ ಸಹಿ ಮಾಡುವವರೆಗೆ ಮುಂದಿನ ಹಂತಕ್ಕೆ ಹೋಗಲು ಸಾಧ್ಯವಿಲ್ಲ.

©NIMI NOT TO BE REPUBLISHED

ಸದಸ್ಯರ ನಮ್ಯತೆ (Member Flexibility) : SDLC ಗಳು ಯೋಜನೆಯ ಗುರಿಗಳಿಗೆ ಮತ್ತು ವಿಧಾನಗಳಿಗೆ ಉತ್ತಮ ರಚನಾತ್ಮಕವಾದ ಡಾಕ್ಯುಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿವೆ. ತಂಡದ ಸದಸ್ಯರು ಹೊರಹೋಗುವುದರಿಂದ ಮತ್ತು ಹೊಸ ಸದಸ್ಯರಿಂದ ಬದಲಾಯಿಸುವುದರಿಂದ ಯಾವ ಸಮಸ್ಯೆಯೂ ಇರುವುದಿಲ್ಲ.

ಪರಿಪೂರ್ಣತೆಯನ್ನು ಸಾಧಿಸಬಹುದಾಗಿದೆ (Perfection Is Achievable) : ಎಲ್ಲ SDLC ಹಂತಗಳು ಪರಸ್ಪರ ಒಂದಕ್ಕೊಂದು ಪ್ರತಿಕ್ರಿಯೆ ನೀಡುತ್ತವೆ. ಆದ್ದರಿಂದ SDLC ಮಾಡೆಲ್‌ಗಳು ಯೋಜನೆಗಳಿಗೆ ಪುನರಾವರ್ತಿಸಲು ಮತ್ತು ಪರಿಪೂರ್ಣವಾಗುವವರೆಗೆ ತಮ್ಮನ್ನು ತಾವು ಪದೇಪದೇ ಸುಧಾರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತವೆ.

ಯಾವುದೇ ಸದಸ್ಯರು ಯೋಜನೆಯನ್ನು ರೂಪಿಸುವುದಿಲ್ಲ ಅಥವಾ ಮುರಿಯುವುದಿಲ್ಲ(No One Member Makes or Breaks the Project)

ಮತ್ತೆ SDLC ಗಳು ವ್ಯಾಪಕವಾದ ದಾಖಲೆಗಳನ್ನು ಮತ್ತು ಮಾರ್ಗಸೂಚಿ ದಾಖಲೆಗಳನ್ನು ಬಳಸುವುದರಿಂದ, ಇದು ಒಂದು ತಂಡದ ಪ್ರಯತ್ನವಾಗಿದೆ ಮತ್ತು ಒಬ್ಬ ಪ್ರಮುಖ ಸದಸ್ಯನನ್ನು ಕಳೆದುಕೊಳ್ಳುವುದು ಯೋಜನೆಯ ಸಮಯದ ಅವಧಿಗೆ ಅಪಾಯವನ್ನು ಉಂಟುಮಾಡುವುದಿಲ್ಲ.

SDLC ಯನ್ನು ಎಲ್ಲಿ ಬಳಸಲಾಗುತ್ತದೆ? (Where is SDLC Used?)

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ಗಳನ್ನು ಸಾಮಾನ್ಯವಾಗಿ IT ಯೋಜನೆಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಸಾಫ್ಟ್‌ವೇರ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಮ್ಯಾನೇಜರ್‌ಗಳು - ವಿವಿಧ ಅಭಿವೃದ್ಧಿ ಹಂತಗಳನ್ನು ರೂಪಿಸಲು, ಎಲ್ಲರೂ ಸರಿಯಾದ ಸಮಯಕ್ಕೆ ಮತ್ತು ಸರಿಯಾದ ಕ್ರಮದಲ್ಲಿ ಹಂತಗಳನ್ನು ಪೂರ್ಣಗೊಳಿಸಿರುವುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು, ಮತ್ತು ಆ ಯೋಜನೆಯನ್ನು ಸಾಧ್ಯವಾದಷ್ಟು ಶೀಘ್ರವಾಗಿ ಮತ್ತು ದೋಷಮುಕ್ತವಾಗಿ ವಿತರಿಸಲು SDLC ಗಳನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತಾರೆ.

SDLC ಗಳನ್ನು ಸಿಸ್ಟಮ್ ವಿಶ್ಲೇಷಕರು ಹೆಚ್ಚು ನಿರ್ದಿಷ್ಟವಾಗಿ ಬಳಸಿಕೊಳ್ಳುತ್ತಾರೆ. ಅವರು ಹೊಸ ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುತ್ತಾರೆ ಮತ್ತು ನಂತರ ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತಾರೆ.

ಯಾವ SDLC ಮಾಡೆಲ್ ಉತ್ತಮವಾಗಿದೆ? (What SDLC Model is Best?)

ಇದು ನಿಮ್ಮ ತಂಡದ ಗುರಿಗಳು ಮತ್ತು ಸಂಪನ್ಮೂಲದ ಅವಶ್ಯಕತೆಗಳನ್ನು ಅವಲಂಬಿಸಿರುತ್ತದೆ.

ಹೆಚ್ಚಿನ ಐಟಿ ಡೆವಲಪ್‌ಮೆಂಟ್ ತಂಡಗಳು ತಮ್ಮ SDLC ಗೆ, ಅಗ್ಯೆಲ್ ವಿಧಾನವನ್ನು ಬಳಸಿಕೊಳ್ಳುತ್ತವೆ. ಅದಾಗ್ಯೂ ಇತರರು ಇಟರೇಟಿವ್ ಅಥವಾ ಸ್ಪೈರಲ್ ವಿಧಾನಗಳಿಗೆ ಆದ್ಯತೆ ನೀಡಬಹುದು.

ಈ ಎಲ್ಲಾ ಮೂರೂ ವಿಧಾನಗಳು ಜನಪ್ರಿಯವಾಗಿವೆ. ಏಕೆಂದರೆ ಒಂದು ಉತ್ಪನ್ನವನ್ನು ಸೋರ್ಸ್ ಕೋಡ್‌ನೊಂದಿಗೆ ಸಂಯೋಜಿಸಲಾಗುವ ಅಥವಾ ಮಾರುಕಟ್ಟೆಗೆ ತಲುಪಿಸುವ ಮೊದಲು ಅವು ವ್ಯಾಪಕ ಪುನರಾವರ್ತನೆಗೆ ಮತ್ತು ದೋಷ ಪರೀಕ್ಷೆಗೆ ಅನುಮತಿಸುತ್ತವೆ.

DevOps ವಿಧಾನಗಳು ಸಹ ಜನಪ್ರಿಯ ಆಯ್ಕೆಯಾಗಿವೆ. DevOps ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳಲು ನಿಮಗೆ ರಿಫ್ರೆಶರ್ ಕೋರ್ಸ್‌ನ ಅವಶ್ಯಕತೆ ಇದ್ದರೆ, ಅದನ್ನು CloudDefense ನಲ್ಲಿ ವಿವರಿಸಲಾಗಿದೆ.

SDLC ಏನನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುತ್ತದೆ? (What Does SDLC Develop?)

SDLC ಗಳನ್ನು ಇಂಜಿನಿಯರ್ ಸಾಫ್ಟ್‌ವೇರ್ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಮತ್ತು ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಸಹ ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು ಹಾರ್ಡ್‌ವೇರ್, ಅಥವಾ ಒಂದೇ ಸಮಯದಲ್ಲಿ ಸಾಫ್ಟ್‌ವೇರ್ ಮತ್ತು ಹಾರ್ಡ್‌ವೇರ್ ಸಂಯೋಜನೆಯನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಸಹ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

FAQ ಗಳು

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ನ 5 ಮೂಲ ಹಂತಗಳು ಯಾವವು ?

ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ ಮೂಲತಃ ಏಳು ಹಂತಗಳ ಬದಲಾಗಿ ಐದು ಹಂತಗಳನ್ನು ಒಳಗೊಂಡಿತ್ತು. ಇವು ಯೋಜನೆ (planning) , ರಚಿಸುವುದು (creating) , ಅಭಿವೃದ್ಧಿ ಪಡಿಸುವುದು (developing) , ಪರೀಕ್ಷಿಸುವುದು (testing), ಮತ್ತು ನಿಯೋಜಿಸುವುದು (deploying). ಇದು ವಿಶ್ಲೇಷಣೆ ಮತ್ತು ನಿರ್ವಹಣೆಯಂತಹ ಪ್ರಮುಖ ಹಂತಗಳನ್ನು ಹೊಂದಿಲ್ಲ ಎನ್ನುವುದನ್ನು ಗಮನಿಸಿ.

SDLC ಯ 7 ಹಂತಗಳು ಯಾವವು? (What Are the 7 Phases of SDLC?)

SDLC ಯ ಹೊಸ 7 ಹಂತಗಳು ಯೋಜನೆ (planning) , ವಿಶ್ಲೇಷಣೆ (analysis) , ವಿನ್ಯಾಸ (design) , ಅಭಿವೃದ್ಧಿ (development), ಪರೀಕ್ಷೆ (testing), ಅನುಷ್ಠಾನ (implementation) , ಮತ್ತು ನಿರ್ವಹಣೆ (maintenance). ಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

MIS ನಲ್ಲಿ ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್ ಎಂದರೇನು? (What is System Development Life Cycle in MIS?)

ಮ್ಯಾನೇಜ್‌ಮೆಂಟ್ ಇನ್‌ಫಾರ್ಮೇಷನ್ ಸಿಸ್ಟಮ್ ಅಥವಾ MIS ನಲ್ಲಿ, SDLC ಯು ಮ್ಯಾನೇಜರ್‌ಗಳಿಗೆ ಮಾಹಿತಿ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸಲು, ಅಭಿವೃದ್ಧಿಗೊಳಿಸಲು, ಪರೀಕ್ಷಿಸಲು ಮತ್ತು ನಿಯೋಜಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ.

ತೀರ್ಮಾನ (Conclusion)

ಅಂತಿಮವಾಗಿ, IT ಮತ್ತು ಇತರ ಕೈಗಾರಿಕೆಗಳಲ್ಲಿರುವ ಯಾವುದೇ ಡೆವಲಪ್‌ಮೆಂಟ್ ತಂಡವು ಸಿಸ್ಟಮ್ ಡೆವಲಪ್‌ಮೆಂಟ್ ಲೈಫ್ ಸೈಕಲ್‌ಗಳನ್ನು ತಮ್ಮ ಯೋಜನೆಗಳಲ್ಲಿ ಅನುಷ್ಠಾನಗೊಳಿಸುವುದರಿಂದ ಪ್ರಯೋಜನ ಪಡೆಯಬಹುದು. ಉತ್ತಮ ಫಲಿತಾಂಶಗಳಿಗಾಗಿ ನೀವು ಯಾವ ವಿಧಾನವನ್ನು ಬಳಸಲು ಬಯಸುತ್ತೀರಿ ಎಂಬುದನ್ನು ಗುರುತಿಸಲು ಮೇಲೆ ತಿಳಿಸಿದ ಮಾರ್ಗದರ್ಶಿಯನ್ನು ಬಳಸಿರಿ.

© NIMI NOT TO BE REPUBLISHED

ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆ (ಪೈಥಾನ್) ಕಮಾಂಡ್ ಲೈನ್‌ನಿಂದ ಪೈಥಾನ್ ಬಳಸಿ (Programming language (Python) Use Python from command line)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಪೈಥಾನ್‌ನ ಪರಿಚಯ
- ಪೈಥಾನ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳು
- ಪೈಥಾನ್‌ನ ಕಮೆಂಟ್‌ಗಳು
- ವೇರಿಯೇಬಲ್.

ಪೈಥಾನ್, ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ, ಉನ್ನತ ಮಟ್ಟದ, ಇಂಟರ್‌ಪ್ರಿಟ್ ಮಾಡಲಾಗುವ ಒಂದು ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಇದನ್ನು ಗೈಡೊ ವಾನ್ ರೋಸಮ್ ಎನ್ನುವವರು 1980 ರ ದಶಕದ ಉತ್ತರಾರ್ಧದಲ್ಲಿ ರಾಷ್ಟ್ರೀಯ ಸಂಶೋಧನಾ ಸಂಸ್ಥೆ, ಗಣಿತ ಮತ್ತು ಕಂಪ್ಯೂಟರ್ ವಿಜ್ಞಾನ, ನೆದರ್‌ಲ್ಯಾಂಡ್‌ನಲ್ಲಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದರು.

ಪೈಥಾನ್ ಅತ್ಯಂತ ಜನಪ್ರಿಯ ಮತ್ತು ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದನ್ನು ಕನ್ಸೋಲ್ ಆಧಾರಿತ, GUI ಆಧಾರಿತ ವೆಬ್ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಮತ್ತು ಡಾಟಾ ವಿಶ್ಲೇಷಣೆಗಳು ಮುಂತಾದ ಕಾರ್ಯಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಪೈಥಾನ್ ಕಲಿಯಲು ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಸರಳವಾದ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಅದ್ದರಿಂದ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಕಲಿಯಲು ನೀವು ಹೊಸಬರಾಗಿದ್ದರೂ ಸಹ ಯಾವುದೇ ಸಮಸ್ಯೆಗಳಿಲ್ಲದೇ ಪೈಥಾನನ್ನು ನೀವು ಸುಲಭವಾಗಿ ಕಲಿಯಬಹುದು.

ವಾಸ್ತವ : ಇದನ್ನು ಹಾಸ್ಯ ದೂರದರ್ಶನ ಕಾರ್ಯಕ್ರಮ Monty Python's Flying Circus ನ ನಂತರ ಪೈಥಾನ್ ಎಂದು ಹೆಸರಿಸಲಾಗಿದೆ.

- ಪೈಥಾನ್ ಅಪ್ಲೀಕೇಷನ್‌ಗಳು
- ಪೈಥಾನ್ ಇನ್ಸ್ಟಾಲೇಷನ್

ಪೈಥಾನ್‌ನ ವೈಶಿಷ್ಟ್ಯಗಳು(Features of Python)

ಪೈಥಾನ್ ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾದ ಸಾಕಷ್ಟು ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ :

- ಕಲಿಯಲು ಮತ್ತು ಬಳಸಲು ಸುಲಭವಾಗಿದೆ.

ಇತರ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳಿಗೆ ಹೋಲಿಸಿದರೆ, ಪೈಥಾನ್ ಕಲಿಯಲು ಮತ್ತು ಬಳಸಲು ತುಂಬಾ ಸುಲಭವಾಗಿದೆ. ಇದು ಅಭಿವೃದ್ಧಿಪಡಿಸುವವರಿಗೆ ಸ್ನೇಹಿಯಾಗಿದೆ ಮತ್ತು ಹೈ ಲೆವೆಲ್ ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ.

- ಇಂಟರ್‌ಪ್ರಿಟೆಡ್ ಭಾಷೆಯಾಗಿದೆ

ಪೈಥಾನ್ ಒಂದು ಇಂಟರ್‌ಪ್ರಿಟೆಡ್ ಭಾಷೆಯಾಗಿದೆ. ಏಂದರೆ, ಇದಕ್ಕೆ ಕಂಪೈಲೇಷನ್‌ನ ಅಗತ್ಯವಿಲ್ಲ. ಇದು ಡೀಬಗ್ ಮಾಡುವುದನ್ನು ಸುಲಭಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಆರಂಭಿಸುವವರಿಗೆ ಸೂಕ್ತವಾಗಿದೆ.

- ಕ್ರಾಸ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಭಾಷೆಯಾಗಿದೆ.

ಪೈಥಾನ್‌ನ್ನು ವಿವಿಧ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಸಮಾನವಾಗಿ ರನ್ ಮಾಡಬಹುದು. ಉದಾಹರಣೆಗೆ , Windows, Linux, Unix ಮತ್ತು Macintosh ಇತ್ಯಾದಿಗಳು. ಅದ್ದರಿಂದ ಪೈಥಾನನ್ನು ನಾವು ಪೋರ್ಟೇಬಲ್ ಭಾಷೆ ಎಂದು ಕರೆಯಬಹುದು.

- ಉಚಿತ ಮತ್ತು ಓಪನ್ ಸೋರ್ಸ್

ಪೈಥಾನ್ ಇಂಟರ್‌ಪ್ರಿಟರ್‌ನ್ನು ಓಪನ್ ಸೋರ್ಸ್ ಪರವಾನಿಗೆ (ಲೈಸೆನ್ಸ್) ಅಡಿಯಲ್ಲಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಗಿದೆ. ಅದ್ದರಿಂದ ಇದು ಇನ್ಸ್ಟಾಲ್ ಮಾಡಲು, ಬಳಸಲು ಮತ್ತು ವಿತರಿಸಲು ಉಚಿತವಾಗಿರುತ್ತದೆ.

- ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಭಾಷೆ

ಪೈಥಾನ್ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಭಾಷೆಯನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ ಮತ್ತು ಕ್ಲಾಸ್ ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಪರಿಕಲ್ಪನೆಗಳು ಅಸ್ತಿತ್ವಕ್ಕೆ ಬರುತ್ತವ.

- GUI ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಬೆಂಬಲ

ಪೈಥಾನ್ ಉಪಯೋಗಿಸಿ ಗ್ರಾಫಿಕಲ್ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಬಹುದು..

- ಇಂಟಿಗ್ರೇಟೆಡ್

ಇದನ್ನು C, C++ ಮತ್ತು JAVA ನಂತಹ ಭಾಷೆಗಳೊಂದಿಗೆ ಸುಲಭವಾಗಿ ಸಂಯೋಜಿಸಬಹುದು(integrated).

© NIMI NOT TO BE REPUBLISHED

ಪೈಥಾನ್‌ನಲ್ಲಿ ಕಮೆಂಟ್‌ಗಳು (Comments in Python)

ಸಾಮಾನ್ಯವಾಗಿ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಲ್ಲಿ ಕಮೆಂಟ್‌ಗಳನ್ನು, ಪ್ರೊಗ್ರಾಮನ್ನು ವಿವರಿಸಲು ಅಥವಾ ಇಂಟರ್‌ಪ್ರಿಟರ್‌ನಿಂದ ಕೋಡ್‌ನ ಕೆಲವು ಭಾಗವನ್ನು ಮರೆಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಪೈಥಾನ್‌ನಲ್ಲಿ ಕಮೆಂಟ್‌ಗಳನ್ನು ಯಾವುದೇ ಪ್ರೊಗ್ರಾಂನ ಕೋಡ್ ಅನ್ನು ವಿವರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅದೇ ರೀತಿ ಕೋಡ್ ಅನ್ನು ಮರೆಮಾಡಲು ಸಹ ಇದನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಕಮೆಂಟ್ ಪ್ರೊಗ್ರಾಂನ ಭಾಗವಲ್ಲ, ಆದರೆ ಇದು ಪ್ರೊಗ್ರಾಂನ ಪರಸ್ಪರ ಕಾರ್ಯನಿರ್ವಹಣೆಯನ್ನು ವರ್ಧಿಸುತ್ತದೆ ಮತ್ತು ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ಸರಳವಾಗಿ ಓದುವಂತೆ ಮಾಡುತ್ತದ.

ಪೈಥಾನ್ ಎರಡು ರೀತಿಯ ಕಮೆಂಟ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ :

- ಒಂದು ಸಾಲಿನ ಕಮೆಂಟ್
- ಬಹುಸಾಲಿನ ಕಮೆಂಟ್

1 **ಒಂದು ಸಾಲಿನ ಕಮೆಂಟ್**

ಬಳಕೆದಾರರು ಒಂದೇ ಸಾಲಿನ ಕಮೆಂಟನ್ನು ಉಪಯೋಗಿಸಲು ಬಯಸಿದರೆ, ಕಮೆಂಟ್ '#' ನಿಂದ ಪ್ರಾರಂಭವಾಗಬೇಕು.

ಸ್ವರೂಪ :

```
# This is single line comment
```

ಉದಾಹರಣೆ : **"scomment.py"**

```
# This is single line comment.
print("Hello Python")
```

ಫಲಿತಾಂಶ : **$python3 scomment.py**

```
Hello Python
```

2 **ಬಹುಸಾಲಿನ ಕಮೆಂಟ್**

ಬಹುಸಾಲಿನ ಕಮೆಂಟ್‌ಗಳನ್ನು ಟ್ರಿಪಲ್ ಕೋಟ್‌ಗಳಲ್ಲಿ (triple quote) ನೀಡಲಾಗುತ್ತದೆ. ಅವುಗಳನ್ನು ಸಾಲಿನ ಪ್ರಾರಂಭದಲ್ಲಿ ನೀಡಬಹ.

ಸ್ವರೂಪ :

```
''' This
Is
Multiline comment'''
```

ಉದಾಹರಣೆ : **"mcomment.py"**

```
'''This
is Multi line comment'''
print("Hello Python")
```

ಫಲಿತಾಂಶ : $python3 mcomment.py

```
Hello Python
```

ಉದಾಹರಣೆ:

ಉದಾಹರಣೆ : "comments.py"

```
# This example demonstrates usage of Comments
    ... print() used to
    print/display
    text on screen '''
print("Welcome to Python")
#Assign value to variables
a=20
b=30
#print sum of two numbers
print("Sum is")
print(a+b)
```

ಫಲಿತಾಂಶ : **$python3 comments.py**

```
Welcome to Python
Sum is
50
```

ಪೈಥಾನ್‌ನಲ್ಲಿ ವೇರಿಯೇಬಲ್‌ಗಳು (Variables in Python)

ವೇರಿಯೇಬಲ್ ಎನ್ನುವುದು ಹೆಸರಿಸಿದ ಮೆಮೊರಿ ಸ್ಥಳವಾಗಿದೆ. ಇದರಲ್ಲಿ ನಾವು ಒಂದು ನಿರ್ದಿಷ್ಟ ಪ್ರೊಗ್ರಾಂಗಾಗಿ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಹುದು. ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ವೇರಿಯೇಬಲ್ ಎನ್ನುವುದು ಮೆಮೊರಿ ಸ್ಥಳವನ್ನು ಹೆಸರಿಸಲು ಬಳಸಲಾಗುವ ಒಂದು ಹೆಸರು. ವೇರಿಯೇಬಲ್ ಅನ್ನು ಐಡೆಂಟಿಫಯರ್ ಎಂದೂ ಸಹ ಕರೆಯಲಾಗುತ್ತದೆ ಮತ್ತು ಇದನ್ನು ಮೌಲ್ಯವನ್ನು ಸಂಗ್ರಹಿಸಲು ಬಳಸಲಾಗುತ್ತದ.

ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ರಚಿಸುವುದು (Creating Variables)

ಪೈಥಾನ್‌ನಲ್ಲಿ, ನಾವು ಸ್ಪಷ್ಟವಾಗಿ ವೇರಿಯೇಬಲ್‌ನ್ನು ಘೋಷಿಸುವ ಅಗತ್ಯವಿಲ್ಲ. ನಾವು ಯಾವುದೇ ಮೌಲ್ಯವನ್ನು ವೇರಿಯೇಬಲ್‌ಗೆ

© NIMI NOT TO BE REPUBLISHED

ನಿಯೋಜಿಸಿದಾಗ ಆ ವೇರಿಯೆಬಲ್ ಅನ್ನು ಸ್ವಯಂಚಾಲಿತವಾಗಿ ಘೋಷಿಸಲಾಗುತ್ತದ.

ಪೈಥಾನ್‌ನಲ್ಲಿ ನಾವು ವೆರಿಯೇಬಲ್‌ನ ಪ್ರಕಾರವನ್ನು ತಿಳಿಸುವ ಅಗತ್ಯವಿಲ್ಲ. ಏಕೆಂದರೆ ಪೈಥಾನ್ ಸಡಿಲ ಪ್ರಕಾರದ (loosly typed) ಭಾಷೆಯಾಗಿದೆ. ಎಂದರೆ, loosely typed ಭಾಷೆಗಳಲ್ಲಿ ವೇರಿಯೇಬಲ್‌ನ type ಅನ್ನು ತಿಳಿಸುವ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ ಏಕೆಂದರೆ ವೇರಿಯೇಬಲ್‌ಗೆ ನಿಯೋಜಿಸಲಾದ ಮೌಲ್ಯದ ಆಧಾರದ ಮೇಲೆ ಅದು ತನ್ನ ಡಾಟಾಟೈಪನ್ನು ಸ್ವಯಂ ಬದಲಾಯಿಸಿಕೊಳ್ಳುತ್ತದ.

ಉದಾಹರಣೆಗೆ :

?? a=10????ಇಲ್ಲಿ ವೇರಿಯೇಬಲ್ a ಒಂದು ಪೂರ್ಣಾಂಕ ಡಾಟಾಟೈಪ್ ಆಗಿದ

?? b="Glance"????ಇಲ್ಲಿ ವೇರಿಯೇಬಲ್ b ಒಂದು ಸ್ಟ್ರಿಂಗ್ ಡಾಟಾಟೈಪ್ ಆಗಿದ.

ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಸರಿಸುವ ನಿಯಮಗಳು (Rules for naming variable):

- ವೇರಿಯೇಬಲ್‌ನ ಹೆಸರುಗಳು ಅಕ್ಷರಗಳು ಮತ್ತು ಸಂಖ್ಯೆಗಳ ಗುಂಪಾಗಿರಬಹುದು. ಆದರೆ ಅವು ಅಕ್ಷರ ಅಥವಾ ಅಂಡರ್‌ಸ್ಕೋರ್‌ನಿಂದ ಮಾತ್ರ ಪ್ರಾರಂಭವಾಗಬಹುದು.
- ವೇರಿಯೇಬಲ್ ಹೆಸರಿಗಾಗಿ ಸಣ್ಣ ಅಕ್ಷರಗಳನು (lowercase letters) ಬಳಸಲು ಶಿಫಾರಸು ಮಾಡಲಾಗಿದೆ. 'SUM' and 'sum' ಎರಡೂ ವಿಭಿನ್ನ ವೇರಿಯೇಬಲ್‌ಗಳಾಗಿವ.

ಉದಾಹರಣೆ : "vardemo.py"

```
a=10            #integer
b="StudyGlance"          #string
c=12.5          #float
print(a)
print(b)
print(c)
```

ಫಲಿತಾಂಶ : $python3 vardemo.py

```
10
StudyGlance
12.5
```

ಅನೇಕ ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ಒಂದೇ ಮೌಲ್ಯವನ್ನು ನಿಗದಿಪಡಿಸುವ

ುದುಪೈಥಾನ್, ಅನೇ' ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ಒಂದೇ ಮೌಲ್ಯವನ್ನು ನಿಯೋಜಿಸಲು ನಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ ಮತ್ತು ಒಂದೇ ಸ್ಟೇಟಮೆಂಟ್‌ನಲ್ಲಿ ಅನೇ' ಮೌಲ್ಯಗಳನ್ನು ಅನೇ' ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ನಿಯೋಜಿಸಲು ಅನುಮತಿಸುತ್ತದೆ. ಇದನ್ನು ಬಹು ನಿಯೋಜನೆ (multiple assignment) ಎಂದೂ 'ರೆಯುತ್ತಾರೆ.

ಅನೇಕ ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ಒಂದೇ ಮೌಲ್ಯವನ್ನು ನಿಯೋಜಿಸುವುದು

ಉದಾಹರಣೆ : "vardemo1.py"

```
x=y=z=50
print x
print y
print z
```

ಅನೇಕ ವೇರಿಯೇಬಲ್‌ಗಳಿಗೆ ಅನೇಕ ಮೌಲ್ಯಗಳನ್ನು ನಿಯೋಜಿಸುವುದು

ಉದಾಹರಣೆ : "vardemo2.py"

```
a,b,c=5,10,15
print a
print b
print c
```

ಫಲಿತಾಂಶ : $python3 vardemo2.py

```
5
10
15
```

NOT TO BE REPUBLISHED © NIMI

ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಮತ್ತು ಆಪರೇಟರ್‌ಗಳನ್ನು ಬಳಸಿಕೊಂಡು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Perform Operations using Data Types and Operators)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಪೈಥಾನ್‌ನಲ್ಲಿ ಡಾಟಾಟೈಪ್‌ಗಳು, ಆಪರೇಟರ್‌ಗಳು.**

ಸಾಮಾನ್ಯವಾಗಿ ಡಾಟಾಟೈಪ್‌ಗಳು ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಯಾವ ರೀತಿಯ ಡಾಟಾವನ್ನು ಸಂಗ್ರಹಿಸಬೇಕು ಎನ್ನುವುದನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುತ್ತವೆ. ವೇರಿಯೇಬಲ್‌ಗಳು ವಿಭಿನ್ನ ಡಾಟಾಟೈಪ್‌ನ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲವು. ಪೈಥಾನ್ ಕ್ರಿಯಾತ್ಮಕವಾಗಿರುವ ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಅಥವಾ loosely typed ಹೊಂದಿರುವ ಭಾಷೆಯಾಗಿದೆ. ಆದ್ದರಿಂದ ನಾವು ವೇರಿಯೇಬಲ್‌ನ್ನು ಘೋಷಿಸುವಾಗ (declare) ಅದರ ಡಾಟಾಟೈಪನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ. ಇಂಟರ್‌ಪ್ರಿಟರ್ ಸೂಚ್ಯವಾಗಿ ಮೌಲ್ಯವನ್ನು ಅದರ ಡಾಟಾಟೈಪ್‌ನೊಂದಿಗೆ ಬಂಧಿಸುತ್ತದೆ.

ಎಂದರೆ, loosely typed ಭಾಷೆಯಲ್ಲಿ ವೇರಿಯೇಬಲ್‌ನ ಟೈಪನ್ನು ನಿರ್ದಿಷ್ಟಪಡಿಸುವ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ. ಏಂದರೆ ವೇರಿಯೇಬಲ್ ತನಗೆ ನಿಯೋಜಿಸಲ್ಪಟ್ಟ ಮೌಲ್ಯದ ಆಧಾರದ ಮೇಲೆ ಸ್ವಯಂ ತನ್ನ ಡಾಟಾಟೈಪನ್ನು ಬದಲಾಯಿಸುತ್ತದೆ.

ಒಂದು ವೇರಿಯೇಬಲ್‌ನ ಟೈಪನ್ನು ಪರಿಶೀಲಿಸಲು ಪೈಥಾನ್ ನಮಗೆ type() ಫಂಕ್ಷನ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಪೈಥಾನ್ ಕೆಳಗಿನ ಪ್ರಮಾಣಿತ ಡಾಟಾಟೈಪ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ, ಅವು ಯಾವವೆಂದರೆ :

1 Numbers (ಸಂಖ್ಯೆಗಳು)

2 String (ಸ್ಟ್ರಿಂಗ್‌ಗಳು)

1 Numbers (ಸಂಖ್ಯೆಗಳು)

ಸಂಖ್ಯೆಗಳು ಸಂಖ್ಯಾ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ. ವೇರಿಯೇಬಲ್‌ಗೆ ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ನಿಯೋಜಿಸಿದಾಗ, ಪೈಥಾನ್ Number ಟೈಪನ್ನು ರಚಿಸುತ್ತದೆ.

ಪೈಥಾನ್‌ನಲ್ಲಿ ಮೂರು numeric ಟೈಪ್‌ಗಳಿವೆ :

a int

b float

c complex

a int

Int, ಅಥವಾ integer, ಒಂದು ಪೂರ್ಣ ಸಂಖ್ಯೆಯಾಗಿದೆ, positive(ಧನಾತ್ಮಕ) ಅಥವಾ negative (ಋಣಾತ್ಮಕ), ದಶಮಾಂಶಗಳಿಲ್ಲದ , ಅನಿಯಮಿತ ಉದ್ದವಾಗಿರುವ ಸಂಖ್ಯೆಯಾಗಿದೆ.

ಸ್ವರೂಪ :

```
a =10
b =-12
c =123456789
```

b float

Float or "floating point number" ಒಂದು ಸಂಖ್ಯೆಯಾಗಿದೆ, positive (ಧನಾತ್ಮಕ) ಅಥವಾ negative (ಋಣಾತ್ಮಕ), ಒಂದು ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ದಶಮಾಂಶಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಸ್ವರೂಪ :

```
X =1.0
Y =12.3
Z =-13.4
```

c complex

ಕಾಂಪ್ಲೆಕ್ಸ್ (ಸಂಕೀರ್ಣ) ಸಂಖ್ಯೆಗಳನ್ನು "j" ನೊಂದಿಗೆ ಕಾಲ್ಪನಿಕ ಭಾಗವಾಗಿ ಬರೆಯಲಾಗುತ್ತದೆ.

ಸ್ವರೂಪ :

```
A =2+5j
B =-3+4j
C =-6j
```

ಉದಾಹರಣೆ : "datatypedemo1.py"

```
a =10
b =10.5
```

© NIMI NOT TO BE REPUBLISHED

```
c =2.14j
print("Datatype of Variable a :" ,type(a))
print("Datatype of Variable b :",type(b))
print("Datatype of Variable c :",type(c))
```

ಫಲಿತಾಂಶ : $python3 datatypedemo1.py

```
Datatype of Variable a :<class 'int'>
Datatype of Variable b :<class 'float'>
Datatype of Variable c :<class 'complex'>
```

2 **String (ಸ್ಟ್ರಿಂಗ್)**

ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು ಕೊಟೇಷನ್ ಮಾರ್ಕ್‌ಗಳಲ್ಲಿ ಬರೆಯಲಾದ ಅಕ್ಷರಗಳ ಅನುಕ್ರಮ (ಗುಂಪು) ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ಪೈಥಾನ್‌ನಲ್ಲಿ ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು ಬರೆಯಲು ಒಂದು, ಎರಡು ಅಥವಾ ಮೂರು ಕೋಟ್ (quote) ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ನಿರ್ವಹಣೆ ಮಾಡುವ ಸಂದರ್ಭದಲ್ಲಿ ಎರಡು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಕ್ಯಾಟಿನೇಟ್ (ಕೂಡಿಸು) ಮಾಡಲು + ಆಪರೇಟರ್‌ನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. "hello"+ "python" ಆಪರೇಟರ್ "hello python". ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ

ಸ್ವರೂಪ:

```
S1='Welcome'
S2="to Python"
S3='''world'''
```

ಉದಾಹರಣೆ :

```
a ="Welcome"      # double quotes ಉಪಯೋಗಿಸಲಾಗಿದೆ
b = 'Python'      # single quotes ಉಪಯೋಗಿಸಲಾಗಿದೆ
c = '''World'''   #triple quotes ಉಪಯೋಗಿಸಲಾಗಿದೆ
print("Datatype of Variable a :",type(a))
print(a+b) # ಎರಡು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಕೂಡಿಸಲು
```

ಫಲಿತಾಂಶ :

```
Datatype of Variable a :<class 'str'>
Welcome Python
```

ಪೈಥಾನ್‌ನಲ್ಲಿ ಆಪರೇಟರ್‌ಗಳು (Operators in Python)

ಆಪರೇಟರ್ ಎಂದರೆ ಎರಡು ಆಪರಂಡ್ (operand) ನಡುವಿನ ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುವ ಒಂದು ಚಿಹ್ನೆ ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು.

ಪೈಥಾನ್‌ನಲ್ಲಿರುವ ವಿವಿಧ ಆಪರೇಟರ್‌ಗಳನ್ನು ಕೆಳಗಿನಂತೆ ವಿವರಿಸಲಾಗಿದೆ :

- Arithmetic operators (ಅಂಕಗಣಿತದ ಆಪರೇಟರ್‌ಗಳು)
- Assignment Operators (ನಿಯೋಜನೆ ಆಪರೇಟರ್‌ಗಳು)
- Comparison operators (ಹೋಲಿಕೆ ಆಪರೇಟರ್‌ಗಳು)
- Bitwise Operators (ಬಿಟ್‌ವೈಸ್ ಆಪರೇಟರ್‌ಗಳು)
- Identity Operators (ಐಡೆಂಟಿಟಿ ಆಪರೇಟರ್‌ಗಳು)
- Logical Operators (ಲಾಜಿಕಲ್ ಆಪರೇಟರ್‌ಗಳು)
- Membership Operators (ಸದಸ್ಯತ್ವದ ಆಪರೇಟರ್‌ಗಳು)

1 **Arithmetic operators (ಅಂಕಗಣಿತದ ಆಪರೇಟರ್‌ಗಳು)**

ಅಂಕಗಣಿತದ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅಂಕಗಣಿತದ ಆಪರೇಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಆಪರೇಟರ್	ವಿವರಣೆ
+ ಸಂಕಲನ (addition) - ವ್ಯವಕಲನ (subtraction)	ಎರಡು ಆಪರಂಡ್‌ಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ. ಎಡಭಾಗದಲ್ಲಿರುವ ಆಪರಂಡ್‌ನಿಂದ ಬಲಭಾಗದಲ್ಲಿರುವ ಆಪರಂಡ್‌ನ್ನು ಕಳೆಯುತ್ತದೆ
* ಗುಣಾಕಾರ (multiplication) / ಭಾಗಾಕಾರ (divide)	ಎರಡು ಆಪರಂಡ್‌ಗಳನ್ನು ಗುಣಿಸುತ್ತದೆ. ಎಡಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ಬಲಭಾಗದ ಆಪರಂಡ್‌ನಿಂದ ಭಾಗಿಸುತ್ತದೆ
% ಶೇಷ (reminder)	ಮಾಡ್ಯುಲಸ್ - ಎಡಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ಬಲಭಾಗದ ಆಪರಂಡ್‌ನಿಂದ ಭಾಗಾಕಾರದ ಶೇಷ
// (floor division)	ಫ್ಲೋರ್ ಡಿವಿಷನ್ - ಭಾಗಾಕಾರದ ಫಲಿತಾಂಶವನ್ನು ಪೂರ್ಣ ಸಂಖ್ಯೆಗೆ ಹೊಂದಿಸಲಾಗುತ್ತದೆ/
** ಘಾತಾಂಕ (exponent)	ಘಾತಾಂಕ - ಎಡ ಆಪರಂಡ್‌ನ್ನು ಬಲ ಆಪರಂಡ್‌ನ ಘಾತಾಂಕವಾಗಿ ಏರಿಸಲಾಗುತ್ತದೆ

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ :

```
x =20
y =10
print('x + y =',x+y)
print('x - y =',x-y)
print('x * y =',x*y)
print('x / y =',x/y)
print('x // y =',x//y)
print('x ** y =',x**y)
```

ಫಲಿತಾಂಶ :

```
x + y =30
x - y =10
x * y =200
x / y =2
x // y =0
x ** y =8
```

2 **Assignment Operators (ನಿಯೋಜನೆ ಆಪರೇಟರ್‌ಗಳು)**

ಬಲಭಾಗದ ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ ಮೌಲ್ಯವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜನೆ ಮಾಡಲು ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಆಪರೇಟರ್	ವಿವರಣೆ
= (Assigns to)	ಬಲಭಾಗದ ಆಪರಂಡ್‌ಗಳ ಮೌಲ್ಯಗಳನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗಳಿಗೆ ನಿಯೋಜಿಸುತ್ತದೆ
+= (Assignment after Addition)	ಬಲಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ಎದಭಾಗದ ಆಪರಂಡ್‌ಗೆ ಸೇರಿಸುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ..
-= (Assignment after Subtraction)	ಎಡಭಾಗದ ಆಪರಂಡ್‌ನಿಂದ ಬಲಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ‘ಳೆಯುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.
*= (Assignment after Multiplication)	ಬಲಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್ ಜೊತೆಗೆ ಗುಣಿಸುತ್ತದೆ ಮತ್ತು ಫಲಿ ತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.
/= (Assignment after Division)	ಎಡಭಾಗದ ಆಪರಂಡ್‌ನ್ನು ಬಲಭಾಗದ ಆಪರಂಡ್ ಜೊತೆಗೆ ಭಾಗಿಸುತ್ತದೆ ಮತ್ತು ಫಲಿ ತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.
%= (Assignment after Modulus)	ಎರಡು ಆಪರಂಡ್‌ಗಳನ್ನು ಬಳಸಿ ಮಾಡ್ಯುಲಸ್ ತೆಗೆಯುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.
**= (Assignment after Exponent)	ಆಪರಂಡ್‌ಗಳ ಮೇಲೆ ಘಾತಾಂ’ದ ಲೆ”ಗಳನ್ನು ಮಾಡುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.
//= (Assignment after floor division)	ಆಪರಂಡ್‌ಗಳ ಮೇಲೆ ಫ್ಲೋರ್ ಡಿವಿಷನ್ ಲೆ”ಗಳನ್ನು ಮಾಡುತ್ತದೆ ಮತ್ತು ಫಲಿತಾಂಶವನ್ನು ಎಡಭಾಗದ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸುತ್ತದೆ.

3 **Comparison operators (ಹೋಲಿಕೆ ಆಪರೇಟರ್‌ಗಳು)**

ಎರಡು ಆಪರಂಡ್‌ಗಳ ಮೌಲ್ಯಗಳನ್ನು ಹೋಲಿಸಲು Comparison ಆಪರೇಟರ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅದರಂತೆ ಬೂಲೀನ್ True ಅಥವಾ False ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
x =20
y =10
print('x > y is', x > y)
print('x < y is', x < y)
print('x == y is', x == y)
print('x != y is', x != y)
print('x >= y is', x >= y)
print('x <= y is', x <= y)
```

ಫಲಿತಾಂಶ :

```
x >y isTrue
x <y isFalse
```

© NIMI NOT TO BE REPUBLISHED

ಆಪರೇಟರ್	ವಿವರಣೆ
== ಸಮಾನ (Equal to)	ಸಮಾನ - ಎರಡೂ ಆಪರಂಡ್‌ಗಳು ಸಮಾನವಾಗಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
!= ಸಮಾನವಾಗಿಲ್ಲ (Not equal to)	ಎರಡೂ ಆಪರಂಡ್‌ಗಳು ಸಮಾನವಾಗಿಲ್ಲದಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
<= ‘ಡಿಮೆ ಅಥವಾ ಸಮಾನ (Less than or equal)	ಎಡಭಾಗದ ಆಪರಂಡ್ ಬಲಭಾಗದ ಆಪರಂಡ್‌ಗಿಂತ ‘ಡಿಮೆ ಅಥವಾ ಸಮನಾಗಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
>= ಹೆಚ್ಚು ಅಥವಾ ಸಮಾನ(Greater than or equal)	ಎಡಭಾಗದ ಆಪರಂಡ್ ಬಲಭಾಗದ ಆಪರಂಡ್‌ಗಿಂತ ಹೆಚ್ಚು ಅಥವಾ ಸಮನಾಗಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
< ‘ಡಿಮೆ(Less than)	ಎಡಭಾಗದ ಆಪರಂಡ್ ಬಲಭಾಗದ ಆಪರಂಡ್‌ಗಿಂತ ‘ಡಿಮೆಯಾಗಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
> ಹೆಚ್ಚು(Greater than)	ಎಡಭಾಗದ ಆಪರಂಡ್ ಬಲಭಾಗದ ಆಪರಂಡ್‌ಗಿಂತ ಹೆಚ್ಚು ಆಗಿದ್ದರೆ, True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ

x == y isFalse

x != y isTrue

x >= y isTrue

x >= y isFalse

4 **Bitwise Operators (ಬಿಟ್‌ವೈಸ್ ಆಪರೇಟರ್‌ಗಳು)**

ಆಪರೇಟರ್	ವಿವರಣೆ
& (binary and)	ಎರಡೂ ಬಿಟ್‌ಗಳೂ 1 ಆಗಿದ್ದರೆ ಪ್ರತಿ ಬಿಟ್ ಅನ್ನು 1 ೆ" ಹೊಂದಿಸುತ್ತದೆ.
| (binary or)	ಎರಡಲ್ಲಿ ಒಂದು ಬಿಟ್ 1 ಆಗಿದ್ದರೆ, ಪ್ರತಿ ಬಿಟ್ ಅನ್ನು 1 ೆ" ಹೊಂದಿಸುತ್ತದೆ.
^ (binary xor)	ಎರಡಲ್ಲಿ ಒಂದು ಬಿಟ್ ಮಾತ್ರ 1 ಆಗಿದ್ದರೆ, ಪ್ರತಿ ಬಿಟ್ ಅನ್ನು 1 ೆ" ಹೊಂದಿಸುತ್ತದೆ.
~ (negation)	ಎಲ್ಲ ಬಿಟ್‌ಗಳನ್ನು ವಿಲೋಮಗೊಳಿಸುತ್ತದೆ.
<< (left shift)	ಸೊನ್ನೆಗಳನ್ನು ಬಲಭಾಗದಿಂದ ಎಡಭಾಗೆ" ತಳ್ಳುವ ಮೂಲ' Shift left ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ಎಡತುದಿಯ ಬಿಟ್‌ಗಳು ಸರಿಸಲ್ಪಡುತ್ತವೆ..
>> (right shift)	ಎಡಭಾಗದಿಂದ ಎಡತುದಿಯ ೌಪಿಗಳನ್ನು ಒಳಗೆ ತಳ್ಳುವ ಮೂಲ' Shift right ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ಬಲತುದಿಯ ಬಿಟ್‌ಗಳು ಸರಿಸಲ್ಪಡುತ್ತವೆ.

ಬಿಟ್‌ವೈಸ್ ಆಪರೇಟರ್‌ಗಳು ಎರಡೂ ಆಪರಂಡ್‌ಗಳ ಮೌಲ್ಯಗಳ ಮೇಲೆ ಬಿಟ್‌ನಿಂದ ಬಿಟ್(bit by bit) ೌರ್ಯಾಚರಣೆಯನ್ನು ನಡೆಸುತ್ತವೆ.

ಉದಾಹರಣೆ :

a =9 # 1001 ಗೆ ಸಮನಾಗಿದೆ

b =12 # 1100 ಗೆ ಸಮನಾಗಿದೆ

print(“a & b =”, a& b)

print(“a | b = ”, a| b)

print(“a ^ b = ”, a ^ b)

print(“~a = ”, ~a)

print(“a << 2 = ”, a <<2)

print(“a >> 2 = ”, a >>2)

ಫಲಿತಾಂಶ :

a & b =8

a | b =13

© NIMI NOT TO BE REPUBLISHED

a ^ b =5

~a =-10

a <<2=36

a >>2=2

5 Identity Operators (**ಐಡೆಂಟಿಟಿ ಆಪರೇಟರ್‌ಗಳು**)

ಐಡೆಂಟಿಟಿ ಆಪರೇಟರ್‌ಗಳನ್ನು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಹೋಲಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅವು ಸಮಾನವಾಗಿದ್ದರೆ ಅಲ್ಲ, ಆದರೆ ಅವು ನಿಜವಾಗಿಯೂ ಒಂದೇ ಮೆಮೊರಿ ಲೋಕೇಷನ್ ಹೊಂದಿರುವ ಸಮಾನ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿದ್ದರೆ.

ಆಪರೇಟರ್	ವಿವರಣೆ
is	ಎರಡೂ ವೇರಿಯೇಬಲ್‌ಗಳು ಒಂದೇ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ (if id(x) equals id(y))
is not	ಎರಡೂ ವೇರಿಯೇಬಲ್‌ಗಳು ಒಂದೇ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿಲ್ಲದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ

ಉದಾಹರಣೆ :

```
x=10
y=10
print(x is y)
print("id(x)= %d id(y)= %d"%(id(x),id(y)))
y=20
print(x isnot y)
print("id(x)= %d id(y)= %d " %(id(x),id(y)))
```

ಫಲಿತಾಂಶ:

```
True
id(x)=10105376id(y)=10105376
True
id(x)=10105376id(y)=10105696
```

6 Logical Operators (**ಲಾಜಿಕಲ್ ಆಪರೇಟರ್‌ಗಳು**)

ಲಾಜಿಲ್ ಆಪರೇಟರ್‌ಗಳನ್ನು ಪ್ರಾಥಮಿವಾಗಿ ಎಕ್ಸ್‌ಪ್ರೆಷನ್ ಮೌಲ್ಯಮಾಪನದಲ್ಲಿ ನಿರ್ಧಾರ ತೆಗೆದುಕೊಳ್ಳಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
x =True
y =False
```

ಆಪರೇಟರ್	ವಿವರಣೆ
and (logical and)	ಎರಡೂ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು true ಆಗಿದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
or (logical or)	ಇವುಗಳಲ್ಲಿ ಒಂದಾದರೂ ಸ್ಟೇಟಮೆಂಟ್ true ಆಗಿದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.
not (logical not)	ಫಲಿತಾಂಶದ ವಿಲೋಮವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಫಲಿತಾಂಶ true ಆಗಿದ್ದರೆ false ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

```
print('x and y is' , x and y)
print('x or y is' , x or y)
print('not x is', not x)
```

ಫಲಿತಾಂಶ :

```
x and y isFalse
x or y isTrue
not x isFalse
```

7 Membership Operators (**ಸದಸ್ಯತ್ವ ಆಪರೇಟರ್‌ಗಳು**)

ಲಾಜಿಲ್ ಆಪರೇಟರ್‌ಗಳನ್ನು ಪ್ರಾಥಮಿವಾಗಿ ಎಕ್ಸ್‌ಪ್ರೆಷನ್ ಮೌಲ್ಯಮಾಪನದಲ್ಲಿ ನಿರ್ಧಾರ ತೆಗೆದುಕೊಳ್ಳಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಆಪರೇಟರ್	ವಿವರಣೆ
in	ನಿರ್ದಿಷ್ಟವಾದ ಮೌಲ್ಯವು ಆಬ್ಜೆಕ್ಟ್‌ನಲ್ಲಿ (list, tuple, or dictionary) ಇದ್ದರೆ True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ
not in)	ನಿರ್ದಿಷ್ಟವಾದ ಮೌಲ್ಯವು ಆಬ್ಜೆಕ್ಟ್‌ನಲ್ಲಿ (list, tuple, or dictionary) ಇಲ್ಲದಿದ್ದರೆ True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ

ಉದಾಹರಣೆ:

```
x ='Hello world'
print('H' in x)
print('hello' notin x)
```

ಫಲಿತಾಂಶ:

```
True
True
```

© NIMI NOT TO BE REPUBLISHED

ಡಿಸಿಷನ್ ಮತ್ತು ಲೂಪ್‌ಗಳೊಂದಿಗೆ ಕಂಟ್ರೋಲ್ ಫ್ಲೋ (Control Flow with Decisions and Loops)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- **if, if-else, if-elif-else.**

ಪೈಥಾನ್‌ನಲ್ಲಿ ಕಂಡೀಷನಲ್ (ಷರತ್ತುಬದ್ಧ) ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು ಷರತ್ತು ಅಥವಾ ಪರಿಸ್ಥಿತಿಯ ಆಧಾರದ ಮೇಲೆ ವಿಭಿನ್ನವಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತವೆ.

ಪೈಥಾನ್‌ನಲ್ಲಿ ಈ ಕೆಳಗಿನ ಕಂಡೀಷನಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು ಇರುತ್ತವೆ.

- if
- if - else
- if - elif -else

ಇಂಡೇಂಟೇಷನ್ (Indentation)

ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ ಸುಲಭತೆಗಾಗಿ ಮತ್ತು ಸರಳತೆಯನ್ನು ಸಾಧಿಸಲು ಪೈಥಾನ್ ಬ್ಲಾಕ್ ಲೆವೆಲ್ ಕೋಡ್‌ಗಳಿಗಾಗಿ ಆವರಣಗಳ ಬಳಕೆಯನ್ನು ಅನುಮತಿಸುವುದಿಲ್ಲ. ಪೈಥಾನ್‌ನಲ್ಲಿ ಒಂದು ಬ್ಲಾಕ್‌ನ್ನು ಘೋಷಿಸಲು ಇಂಡೇಂಟೇಷನ್‌ನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. ಎರಡು ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು ಒಂದೇ ಇಂಡೇಂಟೇಷನ್ ಲೆವೆಲ್‌ನಲ್ಲಿ ಇದ್ದರೆ, ಅವು ಅದೇ ಬ್ಲಾಕ್‌ನ ಭಾಗವಾಗಿರುತ್ತವೆ.

ಸಾಮಾನ್ಯವಾಗಿ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಇಂಡೆಂಟ್ ಮಾಡಲು ನಾಲ್ಕು ಸ್ಥಳಗಳನ್ನು ನೀಡಲಾಗುತ್ತದೆ. ಇದು ಪೈಥಾನ್ ಭಾಷೆಯಲ್ಲಿ ಸಾಮಾನ್ಯ ಪ್ರಮಾಣದ ಇಂಡೆಂಟೇಷನ್ ಆಗಿದೆ.

ಇಂಡೇಂಟೇಷನ್ ಪೈಥಾನ್ ಭಾಷೆಯಲ್ಲಿ ಅತಿ ಹೆಚ್ಚು ಬಳಸುವ ಭಾಗವಾಗಿದೆ. ಏಂದರೆ ಅದು ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಘೋಷಿಸುತ್ತದೆ. ಒಂದು ಬ್ಲಾಕ್‌ನ ಎಲ್ಲ ಸ್ಟೇಟ್‌ಮೆಂಟ್‌ಗಳನ್ನು ಒಂದೇ ಲೆವೆಲ್ ಇಂಡೆಂಟೇಷನ್‌ನಲ್ಲಿ ಬರೆಯಲು ಉದ್ದೇಶಿಸಲಾಗುತ್ತದೆ.

if ಸ್ಟೇಟಮೆಂಟ್ (If statement)

ಒಂದು ನಿರ್ದಿಷ್ಟ ಸ್ಥಿತಿಯನ್ನು ಪರೀಕ್ಷಿಸಲು if ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಬಳಸಲಾಗುತ್ತದೆ. ಷರತ್ತು true ಆಗಿದ್ದರೆ, ಕೋಡ್‌ನ ಬ್ಲಾಕ್ (if-block) ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
if expression:
    statement
```

ಉದಾಹರಣೆ :

```
a =33
b =200
if b > a:
    print("b is greater than a")
```

ಫಲಿತಾಂಶ:

```
b is greater than a
```

If - else ಸ್ಟೇಟಮೆಂಟ್ (If - else statement)

If - else ಸ್ಟೇಟಮೆಂಟ್, if ಸ್ಟೇಟಮೆಂಟ್ ಜೊತೆಗೆ else ಬ್ಲಾಕ್‌ನ್ನು ಸಂಯೋಜಿಸುತ್ತದೆ. ಇದು ಕಂಡಿಷನ್ (ಷರತ್ತು) ಸುಳ್ಳು (false) ಆಗಿದ್ದರೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

ಒಂದು ವೇಳೆ ಕಂಡೀಷನ್ true ಆಗಿದ್ದರೆ, if ಬ್ಲಾಕ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. ಇಲ್ಲದಿದ್ದರೆ else ಬ್ಲಾಕ್ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
if expression:
    #block of statements
else:
    #another block of statements
```

ಉದಾಹರಣೆ :

```
age =int(input("Enter your age : "))
if age >=18:
print("You are eligible to vote !!")
else:
print("Sorry! you have to wait !!")
```

ಫಲಿತಾಂಶ :

```
Enter your age:19
You are eligible to vote!!
```

© NIMI NOT TO BE REPUBLISHED

If - elif - else ಸ್ಟೇಟಮೆಂಟ್ (If - elif - else statement)

elif ಸ್ಟೇಟಮೆಂಟ್ ನಮಗೆ ಅನೇ' ಪರಿಸ್ಥಿತಿಗಳನ್ನು ಪರಿಶೀಲಿಸಲು ಅನುವು ಮಾಡಿಕೊಡುತ್ತದೆ ಮತ್ತು ಅವುಗಳಲ್ಲಿ true ʼಂಡೀಷನ್‌ನ ಆಧಾರದ ಮೇಲೆ ನಿರ್ದಿಷ್ಟ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳ ಬ್ಲಾ'ನ್ನು ʼಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ.

ನಾವು ನಮ್ಮ ಪ್ರೊಗ್ರಾಂನಲ್ಲಿ ನಮ್ಮ ಅಗತ್ಯಗಳಿಗೆ ತ'"ಂತೆ ಹಲವಾರು elif ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿರಬಹುದು. ಆದಾಗ್ಯೂ elif ಅನ್ನು ಬಳಸುವುದು ಐಚ್ಛಿ'ವಾಗಿರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
if expression 1:
        # block of statements
elif expression 2:
        # block of statements
elif expression 3:
        # block of statements
else:
        # block of statements
```

ಉದಾಹರಣೆ :

```
marks =int(input("Enter the marks :"))
if marks >85 and marks <=100:
print("Congrats! you scored grade A..")
elif marks >60 and marks <=85:
print("You scored grade B + ..")
elif marks >40 and marks <=60:
print("You scored grade B ..")
elif(marks >30 and marks <=40):
print("You scored grade C ..")
else:
print("Sorry you are fail ?")
```

ಫಲಿತಾಂಶ :

```
Enter the marks:70
You scored grade B +..
```

ಉದಾಹರಣೆ Program:

Aim: python program to find maximum between three numbers

ಉದಾಹರಣೆ :

```
a=int(input("Enter a value : "))
b=int(input("Enter b value : "))
c=int(input("Enter c value : "))
if(a>b)and(a>c):
print("Maximum value is : ", a)
elif(b>c):
print("Maximum value is :" , b)
else:
print("Maximum value is :", c)
```

ಫಲಿತಾಂಶ:

```
Enter a value:10
Enter b value:14
Enter c value:9
Maximum value is:14
```

© NIMI NOT TO BE REPUBLISHED

ಪೈಥಾನ್‌ನಲ್ಲಿ ಲೂಪ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು (Loop Statements in Python)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

• while loop, for loop.

ʼಲವೊಮ್ಮೆ ನಾವು ʼಾರ್ಯ'ʼ್ರಮದ ಹರಿವನ್ನು ಬದಲಾಯಿಸಬೇ'ಾಗುತ್ತದೆ. ಒಂದು ನಿರ್ದಿಷ್ಟ ʼೋಡ್‌ನ ʼಾರ್ಯವನ್ನು ಹಲವಾರು ಬಾರಿ ಪುನರಾವರ್ತಿಸಬೇ'ಾಗಬಹುದು. ಇದ'ʼಾ"ಗಿ ನಾವು ಲೂಪ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬೇ'ು.

ಈ ಉದ್ದೇಶ'ʼಾ"ಗಿ ಪೈಥಾನ್ ವಿವಿಧ ರೀತಿಯ ಲೂಪ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಈ ಲೂಪ್‌ಗಳು ʼಲವು ನಿರ್ದಿಷ್ಟ ʼೋಡ್‌ಗಳನ್ನು ಹಲವಾರು ಬಾರಿ ಪುನರಾವರ್ತಿಸಲು ಸಮರ್ಥವಾಗಿವೆ.

- while loop
- for loop

ವೈಲ್ ಲೂಪ್ (While loop)

ವೈಲ್ ಲೂಪ್ ಬಳಸಿ, ಒಂದು ಕಂಡೀಷನ್ true ಇರುವವರೆಗೂ ಒಂದು ಸ್ಟೇಟಮೆಂಟ್‌ಗಳ ಗುಂಪನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದು. ವೈಲ್ ಲೂಪನ್ನು ಪುನರಾವರ್ತನೆ ಮಾಡಬೇಕಾದ ಸಂಖ್ಯೆ ಮೊದಲೇ ತಿಳಿದಿಲ್ಲವಾದ ಸಂದರ್ಭದಲ್ಲಿ while ಲೂಪನ್ನು ಹೆಚ್ಚಾಗಿ ಬಳಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
while expression:
Statement(s)
```

ಉದಾಹರಣೆ :

```
i=1;
while i<=3:
print(i);
i=i+1;
```

ಫಲಿತಾಂಶ :

```
1
2
3
```

ವೈಲ್ ಲೂಪ್‌ನೊಂದಿಗೆ else ಅನ್ನು ಬಳಸುವುದು (Using else with while loop)

ಪೈಥಾನ್ ನಮಗೆ while ಲೂಪನ್ನು else ಬ್ಲಾಕ್‌ನೊಂದಿಗೆ ಬಳಸಲು ಅನುಮತಿಸುತ್ತದೆ. while ಸ್ಟೇಟಮೆಂಟ್ false ಆದಾಗ else ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
while expression:
Statements
else:
Statements
```

ಉದಾಹರಣೆ :

```
i=1;
while i<=3:
print(i)
i=i+1;
else:print("The while loop terminated");
```

ಫಲಿತಾಂಶ :

```
1
2
3
The while loop terminated
```

ಫಾರ್ ಲೂಪ್ (For loop)

ಪೈಥಾನ್‌ನಲ್ಲಿರುವ for ಲೂಪ್‌ನ್ನು, ಕೆಲವು ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಅಥವಾ ಪ್ರೋಗ್ರಾಂನ ಒಂದು ಭಾಗವನ್ನು ಹಲವಾರು ಬಾರಿ ಪುನರಾವರ್ತನೆ ಮಾಡಲು ಬಳಸಲಾಗುತ್ತದೆ. ಇದನ್ನು list, tuple, ಅಥವಾ dictionary ನಂತಹ ಡಾಟಾ ಸ್ಟ್ರಕ್ಚರ್‌ಗಳಲ್ಲಿ ಸಂಚರಿಸಲು ಬಳಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
for iterating_var in sequence:
statement(s)
```

ಉದಾಹರಣೆ :

```
i=1
n=int(input("Enter n value : "))
for i inrange(i,n+1):
print(i,end =' ')
```

ಫಲಿತಾಂಶ:

```
Enter n value:5
1 2 3 4 5
```

ಫಾರ್ ಲೂಪ್‌ನೊಂದಿಗೆ else ಬಳಸುವುದು (Using else with for loop)

ಪೈಥಾನ್ for ಲೂಪ್‌ನೊಂದಿಗೆ else ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಬಳಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದು ಎಲ್ಲ ಪುನರಾವರ್ತನೆಗಳು ಮುಗಿದ ನಂತರ ಮಾತ್ರ ಕಾರ್ಯಗತವಾಗುತ್ತದೆ. ಇಲ್ಲಿ ನಾವು ಗಮನಿಸಬೇಕಾಗಿದ್ದು ಏನೆಂದರೆ, ಲೂಪ್ ಏನಾದರೂ break ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಒಳಗೊಂಡಿದ್ದರೆ, else ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುವುದಿಲ್ಲ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
for iterating_var in sequence:
statements
```

© NIMI NOT TO BE REPUBLISHED

```
else:
    statements
```

ಉದಾಹರಣೆ :

```
for i inrange(0,5):
print(i)
else:
print("for loop completely exhausted");
```

ಫಲಿತಾಂಶ :

```
0
1
2
3
4
for loop completely exhausted
```

ಸೂಚನೆ :

range() ಫಂಕ್ಷನ್ ಸಂಖ್ಯೆಗಳ ಸರಣಿಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಡೀಫಾಲ್ಟ್ ಆಗಿ 0 ದಿಂದ ಪ್ರಾರಂಭಿಸುತ್ತದೆ, 1 ರಿಂದ ಹೆಚ್ಚಿಸುತ್ತದೆ (ಡೀಫಾಲ್ಟ್ ಆಗಿ) ಮತ್ತು ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯಲ್ಲಿ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ.

ಉದಾಹರಣೆ :

range(6) means the values from 0 to 5.

range(2,6) means the values from2 to 5.

ಉದಾಹರಣೆ ಪ್ರೊಗ್ರಾಂ:

Aim: Program to printing the table of the given number

ಉದಾಹರಣೆ :

```
i=1;
num =int(input("Enter a number:"));
for i inrange(1,11):
print("%d X %d = %d" %(num,i,num*i))
```

ಫಲಿತಾಂಶ:

```
Enter a number:10
10 X 1=10
10 X 2=20
10 X 3=30
10 X 4=40
10 X 5=50
10 X 6=60
10 X 7=70
10 X 8=80
10 X 9=90
10 X 10=100
```

ಪೈಥಾನ್‌ನಲ್ಲಿ ನೆಸ್ಟೆಡ್ for ಲೂಪ್ (Nested for loop in python)

ಒಂದು for ಲೂಪ್‌ನಲ್ಲಿ ಹಲವಾರು ಸಂಖ್ಯೆಯ for ಲೂಪ್‌ಗಳನ್ನು ಬರೆಯಲು ಪೈಥಾನ್ ನಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಹೊರಗಿನ ಲೂಪ್‌ನ ಪ್ರತಿ ಪುನರಾವರ್ತನೆಗೆ ಒಳಗಿನ ಲೂಪ್ n ಬಾರಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
for iterating_var1 in sequence:
for iterating_var2 in sequence:
#block of statements
```

ಉದಾಹರಣೆ ಪ್ರೊಗ್ರಾಂ :

Aim: Program to printing the rows of stars of the given number

ಉದಾಹರಣೆ :

```
n =int(input("Enter the no.of rows you want to print : "))
for i inrange(0,n):
for j inrange(0,i+1):
print("*",end="")
print()
```

ಫಲಿತಾಂಶ :

```
Enter the no.of rows you want to print:4
*
* *
* * *
* * * *
```

© NIMI NOT TO BE REPUBLISHED

ಡಾಕ್ಯುಮೆಂಟ್ ಮತ್ತು ಸ್ಟ್ರಕ್ಚರ್ ಕೋಡ್ (Document and Structure Code)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **list sequence in python**
- **function of list.**

ಪೈಥಾನ್‌ನಲ್ಲಿ ಪಟ್ಟಿ ಅನುಕ್ರಮ (List Sequence in Python)

ಪೈಥಾನ್‌ನಲ್ಲಿ ವಿಭಿನ್ನ ರೀತಿಯ ಮೌಲ್ಯಗಳ ಅಥವಾ ಅಂಶಗಳ ಸಂಗ್ರಹವನ್ನು ಲಿಸ್ಟ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು comma (,) ದಿಂದ ಪ್ರತ್ಯೇಕಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಇದು ಸ್ಕ್ವೇರ್ ಬ್ರಾಕೆಟ್‌ನಿಂದ [ ] ಸುತ್ತುವರೆದಿದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
List =[value1, value2, value3,ಇ.]
```

ಉದಾಹರಣೆ :

```
L0 =[ ] #creates empty list
L1 =[123 , "python",3.7]
L2 =[1,2,3,4,5,6]
L3 =["C Programing" , "Java","Python"]
print(L0)
print(L1)
print(L2)
print(L3)
```

ಫಲಿತಾಂಶ :

```
[ ]
[123,'python',3.7]
[1,2,3,4,5,6].
['C Programing','Java','Python']
```

ಲಿಸ್ಟ್ ಇಂಡೆಕ್ಸಿಂಗ್ (List indexing)

ಇಂಡೆಕ್ಸಿಂಗ್, ಸಹ ಸ್ಟ್ರಿಂಗ್‌ಗಳು ಯಾವ ರೀತಿಯಲ್ಲಿ ಸಂಸ್ಕರಿಸಲ್ಪಡುತ್ತವೆಯೋ ಅದೇ ರೀತಿಯಲ್ಲಿ ಸಂಭವಿಸುತ್ತವೆ. ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು slice ಆಪರೇಟರ್ ಬಳಸಿ ಪಡೆಯಲಾಗುತ್ತದೆ.

ಇಂಡೆಕ್ಸ್ 0 ದಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ, ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಮೊದಲನೇಯ ಅಂಶವು 0 ಇಂಡೆಕ್ಸ್‌ನಲ್ಲಿ ಸಂಗ್ರಹವಾಗಿರುತ್ತದೆ, ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ

ಪೈಥಾನ್ ನಮಗೆ ಋಣಾತ್ಮಕ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಸಹ ಉಪಯೋಗಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಋಣಾತ್ಮಕ ಇಂಡೆಕ್ಸ್‌ಗಳನ್ನು ಬಲಭಾಗದಿಂದ ಎಣಿಕೆ ಮಾಡಲಾಗುತ್ತದೆ. ಲಿಸ್ಟ್‌ನ ಕೊನೆಯ ಎಲಿಮೆಂಟ್ (ಬಲತುದಿಯ) -1 ಇಂಡೆಕ್ಸ್ ಹೊಂದಿರುತ್ತದೆ, ಅದರ ಪಕ್ಕದ ಎಡ ಭಾಗದ ಎಲಿಮೆಂಟ್ -2 ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಅತ್ಯಂತ

ಲಿಸ್ಟ್ ಆಪರೇಟರ್‌ಗಳು (List Operators)

ಆಪರೇಟರ್	ವಿವರಣೆ
+	ಇದನ್ನು ಕ್ಯಾಟಿನೇಷನ್ (ಸಂಯೋಜನೆ) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಎರಡು ಲಿಸ್ಟ್‌ಗಳನ್ನು ಜೋಡಿಸಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
*	ಇದನ್ನು ರಿಪಿಟಿಷನ್(ಪುನರಾವರ್ತನೆ) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದು ಒಂದೇ ಲಿಸ್ಟನ ಹಲವಾರು ಕಾಪಿಗಳನ್ನು ಸಂಯೋಜಿಸುತ್ತದೆ
[ ]	ಇದನ್ನು ಸ್ಲೈಸ್ (slice) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಲಿಸ್ಟ್‌ನಿಂದ ಒಂದು ಲಿಸ್ಟ್ ಅಂಶವನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
[:]	ಇದನ್ನು ರೇಂಜ್ ಸ್ಲೈಸ್ ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಲಿಸ್ಟ್‌ನಿಂದ ಲಿಸ್ಟ್ ಅಂಶಗಳ ರೇಂಜ್‌ನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
in	ಇದನ್ನು ಮೆಂಬರ್‌ಶಿಪ್ ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದು ನಿರ್ದಿಷ್ಟ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟವಾದ ಅಂಶವು ಇದ್ದರೆ ಅದನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
not in	ಇದೂ ಸಹ ಮೆಂಬರ್‌ಶಿಪ್ ಆಪರೇಟರ್ ಆಗಿರುತ್ತದೆ ಮತ್ತು ಲಿಸ್ಟ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟವಾದ ಅಂಶವು ಇಲ್ಲದಿದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಎಡತುದಿಯ ಎಲೆಮೆಂಟ್‌ವರೆಗೂ ಹೀಗೆಯೆ ಮುಂದುವರೆಯುತ್ತದೆ.

ಉದಾಹರಣೆ:

```
num=[1,2,3,4,5]
lang=['python','c','java','php']
print(num+lang)
print(num*2)
print(lang[2])
print(lang[1:4])
print('cpp'in lang)
print(6notin num)
```

ಫಲಿತಾಂಶ :

```
[1,2,3,4,5,'python' , 'c' , 'java' , 'php']
[1,2,3,4,5,1,2,3,4,5]
java
['c' , 'java' , 'php']
False
True
```

ಒಂದು ಲಿಸ್ಟಗೆ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹೇಗೆ ಸೇರಿಸುವುದು ಅಥವಾ ಬದಲಾಯಿಸುವುದು (How to add or change elements to a list?)

ಪೈಥಾನ್ ನಮಗೆ ಸ್ಲೈಸ್ ಮತ್ತು ಅಸೈನ್‌ಮೆಂಟ ಆಪರೇಟರ್ ಬಳಸಿಕೊಂಡು, ಲಿಸ್ಟ್ ಅಂಶಗಳನ್ನು ಮಾರ್ಪಡಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ನಾವು ಅಸೈನ್‌ಮೆಂಟ್ ಆಪರೇಟರ್ (=) ಬಳಸಿಕೊಂಡು ಲಿಸ್ಟ್ ಅಂಶವನ್ನು ಅಥವಾ ಅಂಶಗಳ ರೇಂಜ್‌ನ್ನು ಬದಲಾಯಿಸಬಹುದು

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5]
print(num)
num[2]=30
print(num)
num[1:3]=[25,36]
print(num)
num[4]="Python"
print(num)
```

ಫಲಿತಾಂಶ :

```
[1,2,3,4,5]
[1,2,30,4,5]
[1,25,36,4,5]
[1,25,36,4,'Python']
```

ಲಿಸ್ಟ್‌ನಿಂದ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹೇಗೆ ಡಿಲೀಟ್ ಮಾಡುವುದು ಅಥವಾ ತೆಗೆದುಹಾಕುವುದು (How to delete or remove elements from a list?)

ಪೈಥಾನ್ ನಮಗೆ del "ೀವರ್ಡ್ಿ ಬಳಸಿ ಒಂದು ಅಥವಾ ಒಂದ"”ಂತ ಹೆಚ್ಚು ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5]
print(num)
del num[1]
print(num)
del num[1:3]
print(num)
```

ಫಲಿತಾಂಶ :

```
[1,2,3,4,5]
[1,3,4,5]
[1,5]
```

ಲಿಸ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳು (List Functions)

ಪೈಥಾನ್ ನಮಗೆ ಲಿಸ್ಟ್‌ಗಳೊಂದಿಗೆ ಬಳಸಬಹುದಾದ ಬಿಲ್ಟ್ - ಇನ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

- len()
- max()
- min()
- list()
- sum()
- sorted()

© NIMI NOT TO BE REPUBLISHED

- append()
- remove()
- sort()
- reverse()
- count()
- index()
- insert()
- pop()
- clear()

len()

ಪೈಥಾನ್‌ನಲ್ಲಿ len() ಅನ್ನು ಲಿಸ್ಟ್‌ನ ಉದ್ದವನ್ನು ಕಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ, ಇದು ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಅಂಶಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
len(list)
```

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5,6]
print("length of list:",len(num))
```

ಫಲಿತಾಂಶ:

```
length of list:6
```

max()

ಪೈಥಾನ್‌ನಲ್ಲಿ max() ಅನ್ನು ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಗರಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
max(list)
```

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp']
print("Max of list :",max(num))
print("Max of list :",max(lang))
```

ಫಲಿತಾಂಶ:

```
Max of list:6
Max of list: python
```

min()

ಪೈಥಾನ್ min() ಅನ್ನು ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಕನಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
min(list)
```

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp',]
print("Min of list :",min(num))
print("Min of list :",min(lang))
```

ಫಲಿತಾಂಶ:

```
Min of list:1
Min of list: c
```

sum()

ಪೈಥಾನ್‌ನಲ್ಲಿ , sum(list) ಫಂಕ್ಷನ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಮೌಲ್ಯಗಳ ಮೊತ್ತವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಮೌಲ್ಯಗಳು ಸಂಖ್ಯೆಗಳಾಗಿರಬೇಕು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sum(list)
```

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5,6]
print("sum of list items :" ,sum(num))
```

ಫಲಿತಾಂಶ:

```
sum of list items:21
```

sorted()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಅಂಶಗಳನ್ನು ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸಲು sorted(list) ಫಂಕ್ಷನ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sorted(list)
```

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ :

```
num=[1,3,2,4,6,5]
lang=['java','c','python','cpp']
print(sorted(num))
print(sorted(lang))
```

ಫಲಿತಾಂಶ:

```
[1,2,3,4,5,6]
['c','cpp','java','python']
```

list ()

List() ಮೆಥೆಡ್, ಅನುಕ್ರಮ ಪ್ರಕಾರಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ಲಿಸ್ಟ್‌ಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ. ಇದನ್ನು ಸ್ಟ್ರಿಂಗ್ ಅಥವಾ ಟ್ಯುಪಲ್ ಅನ್ನು ಲಿಸ್ಟ್‌ಗೆ ಪರಿವರ್ತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list(sequence)
```

ಉದಾಹರಣೆ :

```
str="python"
list1=list(str)
print(list1)
```

ಫಲಿತಾಂಶ:

```
['p','y','t','h','o','n']
```

append ()

ಪೈಥಾನ್‌ನಲ್ಲಿ, append() ಮೆಥೆಡ್ ಒಂದು ಅಂಶವನ್ನು ಲಿಸ್ಟ್‌ನ ಕೊನೆಗೆ ಸೇರಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.append(item)
```

item may be number,string,list and etc

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5]
lang=['python','c','java','php']
num.append(6)
print(num)
lang.append("cpp")
print(lang)
```

ಫಲಿತಾಂಶ :

```
[1,2,3,4,5,6]
['python','c','java','php','cpp']
```

remove()

ಪೈಥಾನ್‌ನಲ್ಲಿ, remove() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಿಂದ ಮೊದಲನೇ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ. ಇದು ಪಾಸ್ ಮಾಡಿದ ಮೌಲ್ಯಕ್ಕೆ ಸಮನಾಗಿರುತ್ತದೆ. ಒಂದು ವೇಳೆ ಎಲಿಮೆಂಟ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ ಅದು ದೋಷವನ್ನು ತೋರಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.remove(item)
```

item ಸಂಖ್ಯೆ, ಸ್ಟ್ರಿಂಗ್, ಲಿಸ್ಟ್ ಇತ್ಯಾದಿ ಆಗಿರಬಹುದು.

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,5]
lang=['python','c','java','php','c']
num.remove(2)
print(num)
lang.remove("c") # first occurence will remove
print(lang)
lang.remove("cpp")
print(lang)
```

ಫಲಿತಾಂಶ :

```
[1,3,4,5]
['python','java','php','c']
ValueError:list.remove(x):x notinlist
```

sort()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ವಿಂಗಡಿಸಲು sort() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಲಿಸ್ಟ್‌ನ ಅಂಶಗಳನ್ನು ಇಳಿಕ್ರಮದಲ್ಲಿ ಮತ್ತು ಏರಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ. ಇದು ಲಿಸ್ಟ್‌ನ್ನು ಇಳಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸಲು ಐಚ್ಛಿಕ ಪ್ಯಾರಾಮೀಟರ್ 'reverse' ನ್ನು ಬಳಸುತ್ತದೆ. ಡೀಫಾಲ್ಟ್ ಆಗಿ ಲಿಸ್ಟ್ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಏರಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.sort([reverse=true])
```

reverse, ಅಂಶಗಳನ್ನು ಇಳಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
lang =['p','y','t','h','o','n']# Char list
even =[6,8,2,4,10]# int list
print(lang)
print(even)
lang.sort()
even.sort()
print("\nAfter Sorting:\n")
print(lang)
print(even)
print("In Descending Order :\n")
even.sort(reverse=True)
print(even)
```

ಫಲಿತಾಂಶ:

```
['p','y','t','h','o','n']
[6,8,2,4,10]
After Sorting:
['h','n','o','p','t','y']
[2,4,6,8,10]
In Descending Order :
[10,8,6,4,2]
```

reverse()

ಪೈಥಾನ್‌ನಲ್ಲಿ, reverse() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ರಿವರ್ಸ್(ಹಿಮ್ಮುಖ) ಗೊಳಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಲಿಸ್ಟ್ ಖಾಲಿ ಯಾಗಿದ್ದರೆ, ಇದು ಖಾಲಿ ಲಿಸ್ಟನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಕೊನೆಯ ಇಂಡೆಕ್ಸ್ ನ್ನು ರಿವರ್ಸ್ ಮಾಡಿದ ಮೇಲೆ ಲಿಸ್ಟ್‌ನ ಮೌಲ್ಯವು 0 ಇಂಡೆಕ್ಸ್‌ನಲ್ಲಿ ಇರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list. reverse ()
```

ಉದಾಹರಣೆ :

```
lang =['p','y','t','h','o','n']
lang =['p','y','t','h','o','n']
print("After reverse")
lang.reverse()
print(lang)
```

ಫಲಿತಾಂಶ:

```
After reverse
['n','o','h','t','y','p']
```

count()

ಪೈಥಾನ್‌ನಲ್ಲಿ, count() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ಒಂದು ಎಲಿಮೆಂಟ್ ಎಷ್ಟು ಬಾರಿ ಗಣಿಸಿಕೊಳ್ಳುತ್ತದೆ ಎಂಬುದನ್ನು ಉತ್ತರಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ಎಲಿಮೆಂಟ್ ಇಲ್ಲದಿದ್ದರೆ, ಅದು 0 ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.count(item)
```

ಉದಾಹರಣೆ :

```
num=[1,2,3,4,3,2,2,1,3,4,5,7,8]
cnt=num.count(2)
print("Count of 2 is:" ,cnt)
cnt=num.count(10)
print("Count of 10 is:",cnt)
```

ಫಲಿತಾಂಶ :

```
Count of 2is:3
Count of 10is:0
```

index()

ಪೈಥಾನ್‌ನಲ್ಲಿ, index () ಮೆಥೆಡ್ ಪಾಸ್ ಮಾಡಲಾದ ಎಲಿಮೆಂಟ್‌ನ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಈ ಮೆಥೆಡ್ ಒಂದು ಆರ್ಗ್ಯುಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅದರ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಎಲಿಮೆಂಟ್ ಇಲ್ಲದಿದ್ದರೆ, ಅದು ValueError ಅನ್ನು ತೋರಿಸುತ್ತದೆ.

ಲಿಸ್ಟ್ ಡುಪ್ಲೀಟ್ ಎಲಿಮೆಂಟ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ, ಮೊದಲು ಸಿಗುವ ಎಲಿಮೆಂಟ್‌ನ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಈ ಮೆಥೆಡ್ ಎರಡು ಅಥವಾ ಎರಡ""ಂತ ಹೆಚ್ಚು ಐಚ್ಚಿ' ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ತೆಗೆದುೂಳ್ಳುತ್ತದೆ. ಆರಂಭಿ' ಮತ್ತು'ೂನೆಯ, ಇವುಗಳನ್ನು ಒಂದು ಮಿತಿಯೊಳಗೆ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹುಡು'ಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.index(x[, start[, end]])
```

ಉದಾಹರಣೆ :

```
lang =['p','y','t','h','o','n','p','r','o','g','r','a','m']
print("index of t is:",lang.index('t'))
print("index of p is:",lang.index('p'))
print("index of p is:",lang.index('p',3,10))
print("index of p is:",lang.index('z'))
```

ಫಲಿತಾಂಶ :

```
index of t is:2
index of p is:0
index of p is:6
ValueError:'z'isnotinlist
```

insert()

ಪೈಥಾನ್‌ನಲ್ಲಿ, insert() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟ ಇಂಡೆಕ್ಸ್‌ನಲ್ಲಿ ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ. ಮೊದಲನೇ ಆರ್ಗ್ಯುಮೆಂಟ್ ಎಂದರೆ ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸಬೇ'ಾದ ಇಂಡೆಕ್ಸ್ ಆಗಿರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

list.insert(i,x)

i : ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸಬೇ'ಾದ ಇಂಡೆಕ್ಸ್

x : ಸೇರಿಸಬೇ'ಾದ ಎಲಿಮೆಂಟ್

ಉದಾಹರಣೆ :

```
num=[10,20,30,40,50]
num.insert(4,60)
print("updated list is :" ,num)
num.insert(7,70)
print("updated list is :" ,num)
```

ಫಲಿತಾಂಶ :

```
updated listis:[10,20,30,40,60,50]
updated listis:[10,20,30,40,60,50,70]
```

pop()

ಪೈಥಾನ್‌ನಲ್ಲಿ, pop() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟವಾದ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ'ುತ್ತದೆ. ಇದು ಪಾಪ್ ಆದ ಎಲಿಮೆಂಟ್ ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.pop([i])
```

ಉದಾಹರಣೆ :

```
num=[10,20,30,40,50]
num.pop()
print("updated list is :",num)
num.pop(2)
print("updated list is :",num)
num.pop(7)
print("updated list is :",num)
```

ಫಲಿತಾಂಶ :

```
updated listis:[10,20,30,40]
updated listis:[10,20,40]
IndexError: pop index out of range
```

clear()

ಪೈಥಾನ್‌ನಲ್ಲಿ, clear() ಮೆಥೆದ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾ'ುತ್ತದೆ. ಇದು ಲಿಸ್ಟ್‌ನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ತೆರವುಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಏನನ್ನೂ ಹಿಂತಿರುಗಿಸುವುದಿಲ್ಲ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
list.clear()
```

ಉದಾಹರಣೆ :

```
num=[10,20,30,40,50]
num.clear()
print("After clearing ",num)
```

ಫಲಿತಾಂಶ :

```
After clearing [ ]
```

© NIMI NOT TO BE REPUBLISHED

ಪೈಥಾನ್‌ನಲ್ಲಿ ಟ್ಯೂಪಲ್ ಸಿಕ್ವೆನ್ಸ್ (Tuple Sequence in Python)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಟ್ಯುಪಲ್** (tuple)
- **ಇಂಡೆಕ್ಸಿಂಗ್** (indexing)
- **ಟ್ಯುಪಲ್ ಆಪರೇಟರ್‌ಗಳು** (Tuple Operators)
- **ಒಂದು ಟ್ಯುಪಲ್‌ಗೆ ಅಂಶಗಳನ್ನು ಹೇಗೆ ಸೇರಿಸುವುದು ಮತ್ತು ತೆಗೆದುಹಾಕುವುದು?**
- **ಟ್ಯುಪಲ್ ಫಂಕ್ಷನ್‌ಗಳು** (Tuple Functions.)

ಪೈಥಾನ್‌ನಲ್ಲಿ ಬದಲಾಗದ ಪೈಥಾನ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಟ್ಯುಪಲ್‌ನ್ನು(Tuple) ಬಳಸಲಾಗುತ್ತದೆ. ಟ್ಯುಪಲ್ ಸಹ ಲಿಸ್ಟ್‌ನ್ನು ಹೋಲುತ್ತದೆ ಆದರೆ ಲಿಸ್ಟ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಿದ ಅಂಶಗಳ ಮೌಲ್ಯಗಳನ್ನು ಬದಲಾಯಿಸಬಹುದು ಮತ್ತು ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಿದ ಅಂಶಗಳ ಮೌಲ್ಯಗಳನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಇದು ಇಮ್ಯುಟೇಬಲ್ ಆಗಿದೆ.

ಟ್ಯುಪಲ್‌ನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ ಪ್ರತ್ಯೇ"ಸಲ್ಪಡುವ ಮೌಲ್ಯಗಳ ಸಂಗ್ರಹ ಎಂದು 'ರೆಯಬಹುದು. ಇದನ್ನು ಸಣ್ಣ ಆವರಣದಲ್ಲಿ (small bracket) ಬರೆಯಬಹುದು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Tuple=(value1, value2,......)
```

ಉದಾಹರಣೆ :

```
T1 =()
T2 =(10,30,20,40,60)
T3 =("C", "Java","Python")
T4 =(501,"abc",19.5)
T5 =(90,)
print(T1)
print(T2)
print(T3)
print(T4)
print(T5)
```

ಫಲಿತಾಂಶ :

```
( )
(10,30,20,40,60)
('C','Java','Python')
(501,'abc',19.5)
(90,)
```

ಟ್ಯುಪಲ್ ಇಂಡೆಕ್ಸಿಂಗ್ (Tuple indexing):

ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಇಂಡೆ"ಂಗ್ ಮತ್ತು ಸ್ಲೈಸಿಂಗ್ ಲಿಸ್ಟ್‌ನಲಿ ್ಲರುವಂತೆಯೆ "ರ್ಯ ನಿರ್ವಹಿಸುತ್ತವೆ. ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ಇಂಡೆ"ಂಗ್ 0 ದಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ ಮತ್ತು ಅದರ "ೂನೆಯ ಲೆಂಥ್‌ವರೆಗೂ ಮುಂದುವರೆಯುತ್ತದೆ (ಟ್ಯುಪಲ್) -1.

ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು ಸ್ಲೈಸ್ ಆಪರೇಟರ್ ಬಳಸಿ ಪಡೆದು"ೂಳ್ಳಬಹುದು. ಪೈಥಾನ್, ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ಒಂದ""ಂತ ಹೆಚ್ಚು ಅಂಶಗಳನ್ನು ಪಡೆದು"ೂಳ್ಳಲು "ೂಲನ್ ಆಪರೇಟರ್‌ನ್ನು ಬಳಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಇತರ ಭಾಷೆಗಳಿಗಿಂತ ಭಿನ್ನವಾಗಿ ಪೈಥಾನ್ ನಮಗೆ ಋಣಾತ್ಮ' ಇಂಡೆ"ಂಗ್ ಬಳಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಋಣಾತ್ಮ' ಇಂಡೆ"ಗಳನ್ನು ಬಲಭಾಗದಿಂದ ಎಣಿಸಲಾಗುತ್ತದೆ. ಟ್ಯುಪಲ್‌ನ "ೂನೆಯ ಎಲಿಮೆಂಟ್ (ಅತ್ಯಂತ ಬಲತುದಿಯ) -1 ಇಂಡೆ"ನ್ನು ಹೊಂದಿದೆ, ಅದರ ಪ""ದ ಎಡಭಾಗದ ಎಲಿಮೆಂಟ್ -2 ಇಂಡೆ" ಹೊಂದಿದೆ ಮತ್ತು ಎಡಭಾಗದ ಅತ್ಯಂತ "ೂನೆಯ ಎಲಿಮೆಂಟ್ ಮುಗಿಯುವವರೆಗೂ ಹೀಗೆ ಮುಂದುವರೆಯುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,5)
lang=('python','c','java','php')
print(num+lang)
print(num*2)
print(lang[2])
print(lang[1:4])
print('cpp'in lang)
print(6 notin num)
```

ಫಲಿತಾಂಶ :

```
(1,2,3,4,5,'python','c','java','php')
(1,2,3,4,5,1,2,3,4,5)
java
('c','java','php')
True
True
```

© NIMI NOT TO BE REPUBLISHED

ಟ್ಯುಪಲ್ ಆಪರೇಟರ್‌ಗಳು(Tuple Operators)

ಆಪರೇಟರ್	ವಿವರಣೆ
+	ಇದನ್ನು 'ಕ್ಯಾಟಿನೇಷನ್ (ಸಂಯೋಜನೆ) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಎರಡು ಟ್ಯುಪಲ್‌ಗಳನ್ನು ಜೋಡಿಸಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
*	ಇದನ್ನು ರಿಪಿಟಿಷನ್(ಪುನರಾವರ್ತನೆ) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದು ಒಂದೇ ಟ್ಯುಪಲ್‌ನ ಹಲವಾರು ಗಿಪಿಗಳನ್ನು ಸಂಯೋಜಿಸುತ್ತದೆ.
[ ]	ಇದನ್ನು ಸ್ಲೈಸ್ (slice) ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಟ್ಯುಪಲ್‌ನಿಂದ ಒಂದು ಟ್ಯುಪಲ್‌ನ ಎಲಿಮೆಂಟ್‌ನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
[:]	ಇದನ್ನು ರೇಂಜ್ ಸ್ಲೈಸ್ ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಟ್ಯುಪಲ್‌ನಿಂದ ಟ್ಯುಪಲ್ ಅಂಶಗಳ ರೇಂಜ್‌ನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸುತ್ತಾರೆ.
in	ಇದನ್ನು ಮೆಂಬರ್‌ಶಿಪ್ ಆಪರೇಟರ್ ಎನ್ನುತ್ತಾರೆ. ಇದು ನಿರ್ದಿಷ್ಟ ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟವಾದ ಅಂಶವು ಇದ್ದರೆ ಅದನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
not in	ಇದೂ ಸಹ ಮೆಂಬರ್‌ಶಿಪ್ ಆಪರೇಟರ್ ಆಗಿರುತ್ತದೆ ಮತ್ತು ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ನಿರ್ದಿಷ್ಟವಾದ ಅಂಶವು ಇಲ್ಲದಿದ್ದರೆ, true ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಒಂದು ಲಿಸ್ಟ್‌ಗೆ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಹೇಗೆ ಸೇರಿಸುವುದು ಅಥವಾ ತೆಗೆದುಹಾಕುವುದು (How to add or remove elements to a list?)

ಲಿಸ್ಟ್‌ನಂತಲ್ಲದೇ, ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ನವೀಕರಿಸಲಾಗುವುದಿಲ್ಲ ಅಥವಾ ಡಿಲೀಟ್ ಮಾಡಲು ಆಗುವುದಿಲ್ಲ. ಏಂದರೆ ಟ್ಯುಪಲ್‌ಗಳು ಇಮ್ಯುಟೇಬಲ್ ಆಗಿರುತ್ತವೆ. ಅವುಗಳನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಸಂಪೂರ್ಣ ಟ್ಯುಪಲ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡಲು, ಟ್ಯುಪಲ್ ಹೆಸರಿನೊಂದಿಗೆ del "ಕವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ :

```
mytuple=('python','c','java','php')
mytuple[3]="html"
#'tuple' object does not support item assignment
print(mytuple)
del mytuple[3]
# 'tuple' object doesn't support item deletion
print(mytuple)
del mytuple
#deletes entire tuple
```

ಫಲಿತಾಂಶ :

```
'tuple' object does not support item assignment
'tuple' object doesn't support item deletion
```

ಟ್ಯುಪಲ್ ಫಂಕ್ಷನ್‌ಗಳು (Tuple Functions)

ಪೈಥಾನ್ ಕೆಳಗಿನ ಬಿಲ್ಟ್ - ಇನ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇವುಗಳನ್ನು ಟ್ಯುಪಲ್‌ಗಳೊಂದಿಗೆ ಉಪಯೋಗಿಸಬಹುದು.

- len()
- max()
- min()
- tuple()
- sum()
- sorted()
- index()
- count()

len()

ಪೈಥಾನ್‌ನಲ್ಲಿ, len() ಫಂಕ್ಷನ್‌ನ್ನು ಟ್ಯುಪಲ್‌ನ ಲೆಂಥ್‌ನ್ನು ಕಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ, ಇದು ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
len(tuple)
```

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,5,6)
print("length of tuple:",len(num))
```

ಫಲಿತಾಂಶ :

```
length of tuple:6
```

© NIMI NOT TO BE REPUBLISHED

max()

ಪೈಥಾನ್‌ನಲ್ಲಿ, max() ಫಂಕ್ಷನ್‌ನ್ನು ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
max(tuple)
```

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,5,6)
lang=('java','c','python','cpp')
print("Max of tuple :",max(num))
print("Max of tuple :",max(lang))
```

ಫಲಿತಾಂಶ :

```
Max of tuple:6
Max of tuple: python
```

min()

ಪೈಥಾನ್‌ನಲ್ಲಿ, min() ಫಂಕ್ಷನ್‌ನ್ನು ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಕನಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
min(tuple)
```

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,5,6)
lang=('java','c','python','cpp')
print("Min of tuple :",min(num))
print("Min of tuple :",min(lang))
```

ಫಲಿತಾಂಶ :

```
Min of tuple:1
Min of tuple: c
```

sum()

ಪೈಥಾನ್‌ನಲ್ಲಿ, sum(tuple) ಫಂಕ್ಷನ್ ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಮೌಲ್ಯಗಳ ಮೊತ್ತವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಟ್ಯುಪಲ್ ಮೌಲ್ಯಗಳು ಸಂಖ್ಯೆಗಳಾಗಿರಬೇಕು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sum(tuple)
```

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,5,6)
print("sum of tuple items :",sum(num))
```

ಫಲಿತಾಂಶ :

```
sum of tuple items:21
```

sorted()

ಪೈಥಾನ್‌ನಲ್ಲಿ, sorted (tuple) ಫಂಕ್ಷನ್‌ನ್ನು ಟ್ಯುಪಲ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಅಂಶಗಳನ್ನು ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸಲು (sort) ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಅಂಶಗಳನ್ನು ಇಳಿಕೆ ಕ್ರಮದಲ್ಲಿ ಮತ್ತು ಏರಿಕೆ ಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ. ಇದು ಐಚ್ಛಿಕ ಪ್ಯಾರಾಮೀಟರ್ 'reverse' ನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಟ್ಯುಪಲ್‌ನ್ನು ಇಳಿಕೆ ಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sorted(tuple[,reverse=True])
```

ಉದಾಹರಣೆ :

```
num=(1,3,2,4,6,5)
lang=('java','c','python','cpp')
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

ಫಲಿತಾಂಶ :

```
(1,2,3,4,5,6)
('c','cpp','java','python')
(6,5,4,3,2,1)
```

tuple (sequence)

tuple() ಮೆಥೆಡ್, ಅನುಕ್ರಮ ವಿಧಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ಟ್ಯುಪಲ್‌ಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ. ಇದನ್ನು ಸ್ಟ್ರಿಂಗ್ ಅಥವಾ ಲಿಸ್ಟ್‌ನ್ನು ಟ್ಯುಪಲ್‌ಗೆ ಪರಿವರ್ತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
tuple(sequence)
```

ಉದಾಹರಣೆ :

```
str="python"
tuple1=tuple(str)
```

© NIMI NOT TO BE REPUBLISHED

```
print(tuple1)

num=[1,2,3,4,5,6]

tuple2=tuple(num)

print(tuple2)
```

ಫಲಿತಾಂಶ :

```
('p','y','t','h','o','n')

(1,2,3,4,5,6)
```

count()

ಪೈಥಾನ್‌ನಲ್ಲಿ, count() ಮೆಥೆಡ್ ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ಒಂದು ಎಲಿಮೆಂಟ್ ಎಷ್ಟು ಬಾರಿ ‘ಂಡುಬರುತ್ತದೆ ಎಂಬುದನ್ನು ಹಿಂತಿಸುಗಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಟ್ಯುಪಲ್‌ನಲ್ಲಿ ಆ ಎಲಿಮೆಂಟ್ ಇಲ್ಲದಿದ್ದರೆ , ಅದು 0 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
tuple.count(item)
```

ಉದಾಹರಣೆ :

```
num=(1,2,3,4,3,2,2,1,3,4,5,7,8)

cnt=num.count(2)

print("Count of 2 is:",cnt)

cnt=num.count(10)

print("Count of 10 is:",cnt)
```

ಫಲಿತಾಂಶ :

```
Count of 2is:3

Count of 10is:0
```

Index()

ಪೈಥಾನ್‌ನಲ್ಲಿ, index () ಮೆಥೆಡ್ ಪಾಸ್ ಮಾಡಲಾದ ಎಲಿಮೆಂಟ್‌ನ ಇಂಡೆಕ್ಸ್‌ನ್ನು ತೋರಿಸುತ್ತದೆ. ಈ ಮೆಥೆಡ್ ಆರ್ಗ್ಯುಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅದರ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಒಂದು ವೇಳೆ ಎಲಿಮೆಂಟ್ ಇಲ್ಲದಿದ್ದರೆ, ಅದು ValueError ನ್ನು ತೋರಿಸುತ್ತದೆ.

ಒಂದು ವೇಳೆ ಟ್ಯುಪಲ್ ಡುಪ್ಲೀಕೇಟ್ ಎಲಿಮೆಂಟ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ, ಮೊದಲು ‘ಂಡುಬಂದ ಎಲಿಮೆಂಟ್‌ನ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಈ ಮೆಥೆಡ್, ಇನ್ನೂ ಎರಡು ಹೆಚ್ಚಿನ ಐಚ್ಚಿ’ ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ, ಆರಂಭದ ಮತ್ತು ‘ೊನೆಯ , ಇವುಗಳನ್ನು ಒಂದು ಮಿತಿಯಲ್ಲಿ ಇಂಡೆಕ್ಸ್‌ಗಳನ್ನು ಹುಡು’ಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
tuple.index(x[, start[, end]])
```

ಉದಾಹರಣೆ :

```
lang =('p','y','t','h','o','n','p','r','o','g','r','a','m')

print("index of t is:",lang.index('t'))

print("index of p is:",lang.index('p'))

print("index of p is:",lang.index('p',3,10))

print("index of p is:",lang.index('z'))
```

ಫಲಿತಾಂಶ :

```
index of t is:2

index of p is:0

index of p is:6

ValueError:'z'isnotintuple.
```

ಪೈಥಾನ್‌ನಲ್ಲಿ ಸೆಟ್ ಸೀಕ್ವೆನ್ಸ್ (Set Sequence in Python)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಸೆಟ್ ರಚಿಸುವುದು**
- **ಸೆಟ್ ಆಪರೇಟರ್‌ಗಳು**
- **ಸೆಟ್ ಫಂಕ್ಷನ್‌ಗಳು.**

ಪೈಥಾನ್‌ನಲ್ಲಿ, ‘ರ್ಲಿ ಬ್ರೇಸ್‌ಗಳಲ್ಲಿ ಬರೆಯಲಾದ ‘್ರಮಬದ್ಧವಾಗಿರದ ವಿವಿಧ ಎಲಿಮೆಂಟ್‌ಗಳ ಸಂಗ್ರಹವನ್ನು ಸೆಟ್ ಎಂದು ‘ರೆಯಲಾಗುತ್ತದೆ. ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳು ಡುಪ್ಲೀಕೇಟ್ ಆಗಿರಲು ಸಾಧ್ಯವಿಲ್ಲ. ಪೈಥಾನ್ ಸೆಟ್‌ನ ಎಲಿಮೆಂಟ್‌ಗಳು ಇಮ್ಯುಟೇಬಲ್ ಆಗಿರುತ್ತವೆ. ಎಂದರೆ ಅವುಗಳನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಪೈಥಾನ್‌ನಲ್ಲಿರುವ ಇತರ ಸಂಗ್ರಹಗಳಂತೆ, ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲಿಮೆಂಟ್‌ಗಳಿಗೆ ಇಂಡೆಕ್ಸ್ ಇರುವುದಿಲ್ಲ. ಎಂದರೆ, ನಾವು ಸೆಟ್‌ನ ಯಾವುದೇ ಎಲಿಮೆಂಟ್‌ನ್ನು ನೇರವಾಗಿ ಅದರ ಇಂಡೆಕ್ಸ್ ಉಪಯೋಗಿಸಿ ಪಡೆಯಲು ಸಾಧ್ಯವಿಲ್ಲ. ಅದಾಗ್ಯೂ, ನಾವು ಅವೆಲ್ಲವುಗಳನ್ನು ಒಟ್ಟಿಗೆ ಪ್ರಿಂಟ್ ಮಾಡಬಹುದು ಅಥವಾ ಸೆಟ್‌ನ್ನು ಲೂಪ್ ಮಾಡುವ ಮೂಲ’ ಎಲಿಮೆಂಟ್‌ಗಳ ಲಿಸ್ಟನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಪಡೆದುಕೊಳ್ಳಬಹುದು

ಸೆಟ್ ರಚಿಸುವುದು (Creating a set)

ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಿದ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಕರ್ಲೀ ಬ್ರೇಸ್ (curly braces) ಗಳಲ್ಲಿ ಬರೆಯುವ ಮೂಲಕ ಸೆಟ್‌ನ್ನು ರಚಿಸಬಹುದು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Set={value1, value2,.....}
```

ಉದಾಹರಣೆ :

```
Days ={“Monday”,“Tuesday”,“Wednesday”,“Thursday”, “Friday”,“Saturday”,“Sunday”}
print(Days)
print(type(Days))
print(“Looping through the set elements ... “)
for i in Days:
print(i)
```

ಫಲಿತಾಂಶ :

```
{‘Wednesday’,‘Tuesday’,‘Sunday’,‘Friday’,‘Thursday’,‘Saturday’ ,‘Monday’}
Looping through the set elements...
Wednesday
Tuesday
Sunday
Friday
Thursday
Saturday
Monday
```

ಸೆಟ್ ಆಪರೇಟರ್‌ಗಳು (Set Operators)

ಪೈಥಾನ್‌ನಲ್ಲಿ, ನಾವು ಪೈಥಾನ್ ಸೆಟ್‌ಗಳ ಮೇಲೆ ವಿವಿಧ ಗಣಿತದ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಬಹುದು. ಉದಾಹರಣೆಗೆ, ಯೂನಿಯನ್(union), ಇಂಟರ್‌ಸ್ಷೆನ್(intersection), ಡಿಫರನ್ಸ್ (difference) ಇತ್ಯಾದಿಗಳು

ಆಪರೇಟರ್	ವಿವರಣೆ
|	Union Operator
&	Intersection Operator
-	Difference Operator:

ಯೂನಿಯನ್ ಆಪರೇಟರ್ (Union (|) Operator)

ಎರಡು ಸೆಟ್‌ಗಳ ಯೂನಿಯನ್‌ನ್ನು or (|) ಆಪರೇಟರ್ ಉಪಯೋಗಿಸಿ ಲೆಕ್ಕ ಮಾಡಲಾಗುತ್ತದೆ. ಎರಡು ಸೆಟ್‌ಗಳ ಯೂನಿಯನ್, ಎರಡೂ ಸೆಟ್‌ಗಳಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಅಂಶಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
Days1={“Mon”,”Tue”,”Wed”,”Sat”}
Days2={“Thr”,“Fri”,“Sat”,“Sun”,“Mon”}
print(Days1 | Days2)
```

ಫಲಿತಾಂಶ :

```
{‘Thr’,‘Fri’,‘Sun’,‘Tue’,‘Wed’,‘Mon’,‘Sat’}
```

ಇಂಟರ್‌ಸೆಕ್ಷನ್ ಆಪರೇಟರ್ (Intersection (&) Operator)

& ಇಂಟರ್‌ಸ್ಷೆನ್(intersection) ಆಪರೇಟರ್‌ನ್ನು ಪೈಥಾನ್‌ನಲ್ಲಿ ಎರಡು ಸೆಟ್‌ಗಳ ಇಂಟರ್‌ಸ್ಷೆನ್‌ನ್ನು ಲೆಕ್ಕ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎರಡು ಸೆಟ್‌ಗಳಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಇರುವ ಎಲಿಮೆಂಟ್‌ಗಳ ಸೆಟ್‌ನ್ನು ಇಂಟರ್‌ಸ್ಷೆನ್‌ನಲ್ಲಿ ತೋರಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
Days1={“Mon”,“Tue”,“Wed”,“Sat”}
Days2={“Th”,“Fri”,“Sat”,“Sun”,“Mon”}
print(Days1 & Days2)
```

ಫಲಿತಾಂಶ :

```
{‘Mon’,‘Sat’}
```

ಡಿಫರನ್ಸ್ ಆಪರೇಟರ್ (Difference (-) Operator)

ಎರಡು ಸೆಟ್‌ಗಳ ವ್ಯತ್ಯಾಸವನ್ನು ವ್ಯವಕಲನ ಆಪರೇಟರ್ (-) ಉಪಯೋಗಿಸಿಕೊಂಡು ಲೆಕ್ಕ ಮಾಡಬಹುದು. ಸೆಟ್ 2 ರಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೆಟ್ 1 ರಿಂದ ತೆಗೆದುಹಾಕುವುದರ ಮೂಲಕ ಹೊಸ ಸೆಟ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಉದಾಹರಣೆ :

```
Days1={“Mon”,“Tue”,“Wed”,“Sat”}
Days2={“Thr”,“Fri”,“Sat”,“Sun”,“Mon”}
print(Days1 - Days2)
```

© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ :

{'Tue','Wed'}

ಸೆಟ್ ಫಂಕ್ಷನ್‌ಗಳು (Set Functions)

ಪೈಥಾನ್ ಸೆಟ್‌ನೊಂದಿಗೆ ಉಪಯೋಗಿಸಲು ಈ ೆಳಗಿನ ಮೆಥೆಡ್‌ಗಳನ್ನು ನೀಡುತ್ತದೆ.

- len(set)
- max(set)
- min(set)
- sum(set)
- sorted(set)
- set()
- add()
- update()
- discard()
- remove()
- pop()
- clear()
- union()
- intersection()
- difference()
- issubset()
- issuperset()

len()

ಪೈಥಾನ್‌ನಲ್ಲಿ, len() ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ನ ಲೆಂಥ್‌ನ್ನು ಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

len(set)

ಉದಾಹರಣೆ :

```
num={1,2,3,4,5,6}
print("length of set:",len(num))
```

ಫಲಿತಾಂಶ :

length of set:6

max()

ಪೈಥಾನ್‌ನಲ್ಲಿ, max() ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ನಲ್ಲಿರುವ ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

max(set)

ಉದಾಹರಣೆ :

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Max of set :",max(num))
print("Max of set :",max(lang))
```

ಫಲಿತಾಂಶ :

Max of set:6

Max of set: python

min()

ಪೈಥಾನ್‌ನಲ್ಲಿ, min() ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ನಲ್ಲಿರುವ ನಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಂಡು ಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

min(set)

ಉದಾಹರಣೆ :

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Min of set :",min(num))
print("Min of set :",min(lang))
```

ಫಲಿತಾಂಶ :

Min of set:1

Min of set: c

sum()

ಪೈಥಾನ್‌ನಲ್ಲಿ , sum(set) ಫಂಕ್ಷನ್ ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಮೌಲ್ಯಗಳ ಮೊತ್ತವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಸೆಟ್ ಮೌಲ್ಯಗಳು ಸಂಖ್ಯೆಗಳಾಗಿರಬೇಕು.

© NIMI NOT TO BE REPUBLISHED

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sum(set)
```

ಉದಾಹರಣೆ :

```
num={1,2,3,4,5,6}
print("sum of set items :",sum(num))
```

ಫಲಿತಾಂಶ :

```
sum of set items:21
```

sorted()

ಪೈಥಾನ್‌ನಲ್ಲಿ, sorted (set) ಫಂಕ್ಷನ್ ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಅಂಶಗಳನ್ನು ಏರಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ. ಇದು ಅಂಶಗಳನ್ನು ಇಳಿಕ್ರಮ ಮತ್ತು ಏರಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ. ಇದು ಐಚ್ಛಿಕ ಪ್ಯಾರಾಮೀಟರ್ 'reverse' ನ್ನು ಉಪಯೋಗಿಸಿ ಸೆಟ್‌ನ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಇಳಿಕ್ರಮದಲ್ಲಿ ವಿಂಗಡಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
sorted(set[,reverse=True])
```

ಉದಾಹರಣೆ :

```
num={1,3,2,4,6,5}
lang={'java','c','python','cpp'}
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

ಫಲಿತಾಂಶ :

```
{1,2,3,4,5,6}
{'c','cpp','java','python'}
{6,5,4,3,2,1}
```

set()

set() ಮೆಥೆಡ್ ಅನುಕ್ರಮ ಪ್ರಕಾರಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಮತ್ತು ಅವುಗಳನ್ನು ಸೆಟ್‌ಗೆ ಪರಿವರ್ತಿಸುತ್ತದೆ. ಇದನ್ನು ನೀಡಲಾದ ಸ್ಟ್ರಿಂಗ್ ಅಥವಾ ಲಿಸ್ಟ್ ಅಥವಾ ಟ್ಯುಪಲ್‌ನ್ನು ಪರಿವರ್ತಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
set(sequence)
```

ಉದಾಹರಣೆ :

```
set1=set("PYTHON")
print(set1)
days=["Mon","Tue","Wed","Thur","Fri","Sat","Sun"]
set2 =set(days)
print(set2)
days=("Mon","Tue","Wed","Thur","Fri","Sat","Sun")
set3 =set(days)
print(set3)
```

ಫಲಿತಾಂಶ :

```
{'N','O','T','H','P','Y'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
```

add()

ಪೈಥಾನ್‌ನಲ್ಲಿ, add() ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ಗೆ ಒಂದು ನಿರ್ದಿಷ್ಟ ಅಂಶವನ್ನು ಸೇರಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
set.add (item)
```

ಉದಾಹರಣೆ :

```
Days = {"Monday","Tuesday","Wednesday","Thursday",
"Friday"}
print("\n printing the original set ... ")
print(Days)
Days.add("Saturday");
Days.add("Sunday");
print("\n Printing the modified set...");
print(Days)
```

ಫಲಿತಾಂಶ :

```
printing the original set...
{'Wednesday','Friday','Thursday','Tuesday','Monday'}
Printing the modified set...
```


© NIMI NOT TO BE REPUBLISHED

{'Wednesday','Sunday','Friday','Thursday','Tuesday',
'Saturday','Monday'}

update()

update () ಮೆಥೆಡ್‌ನ್ನು ಬಳಸಿ ಒಂದು ಸೆಟ್‌ನಲ್ಲಿ ಒಂದ“”ಂತ ಹೆಚ್ಚು ಅಂಶಗಳನ್ನು ಸೇರಿಸಲು ಪೈಥಾನ್ ನಮಗೆ ಅನುಮತಿ ನೀಡುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
set.update([item1, item2, ....])
```

ಉದಾಹರಣೆ :

```
Months={"Jan","Feb","Mar","Apr"}
print("\n Printing the original set ... ")
print(Months)
Months.update (["May","Jun","Jul"])
print("\n Printing the modified set...");
print(Months)
```

ಫಲಿತಾಂಶ :

```
Printing the original set...
{'Mar','Apr','Jan','Feb'}
Printing the modified set...
{'Mar','Apr','Jan','Jun','May','Jul','Feb'}
```

discard()

ಪೈಥಾನ್‌ನಲ್ಲಿ , discard () ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ನಿಂದ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ’ಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದೆ ವೇಳೆ ಆ ಎಲಿಮೆಂಟ್ ಸೆಟ್‌ನಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ, ಪೈಥಾನ್ ಎರರ್ (error) ನ್ನು ತೋರಿಸುವುದಿಲ್ಲ. ಪ್ರೊಗ್ರಾಂ ತನ್ನ ’ಂಟ್ರೋಲ್ ಫ್ಲೋವನ್ನು ನಿಭಾಯಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
set.discard(item)
```

ಉದಾಹರಣೆ :

```
Months={"Jan","Feb","Mar","Apr"}
print("\n printing the original set ... ")
print(Months)
Months.discard("Apr")
print("\n Printing the modified set..");
print(Months)
Months.discard("May") # ಎರರ್‌ನ್ನು ತೋರಿಸುವುದಿಲ್ಲ
print("\n Printing the modified set..");
print(Months)
```

ಫಲಿತಾಂಶ :

```
printing the original set...
{'Jan','Apr','Mar','Feb'}
Printing the modified set...
{'Jan','Mar','Feb'}
Printing the modified set...
{'Jan','Mar','Feb'}
```

remove()

ಪೈಥಾನ್‌ನಲ್ಲಿ , remove () ಮೆಥೆಡ್‌ನ್ನು , ಸೆಟ್‌ನಿಂದ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ’ಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಆ ಎಲಿಮೆಂಟ್ ಸೆಟ್‌ನಲ್ಲಿ ಇಲ್ಲದಿದ್ದರೆ, ಪೈಥಾನ್ ಎರರ್‌ನ್ನು ತೋರಿಸುವುದಿಲ್ಲ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
set.remove(item)
```

ಉದಾಹರಣೆ :

```
Months={"Jan","Feb","Mar","Apr"}
print("\n printing the original set ... ")
print(Months)
Months.remove("Apr")
print("\n Printing the modified set...");
print(Months)
Months.remove("May") #it give error
print("\n Printing the modified set...");
print(Months)
```

ಫಲಿತಾಂಶ :

```
printing the original set...
```

© NIMI NOT TO BE REPUBLISHED

{'Feb','Jan','Apr','Mar}

Printing the modified set...

{'Feb','Jan','Mar'}

KeyError:'May' doesn't exist.

pop()

ಪೈಥಾನ್‌ನಲ್ಲಿ, pop () ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ'ಲಾಗುತ್ತದೆ. ಇದು ಯಾವಾಗಲೂ ೕೂನೆಯ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ'ತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set.pop ()

ಉದಾಹರಣೆ :

```
Days ={"Monday","Tuesday", "Wednesday", "Thursday","Friday"}
print("\n printing the original set ... ")
print(Days)
Days.pop()
print("\n Printing the modified set...");
print(Days)
```

ಫಲಿತಾಂಶ :

Printing the original set...

{'Monday','Wednesday','Friday','Tuesday','Thursday'}

Printing the modified set...

{'Wednesday','Friday','Tuesday','Thursday'}

clear()

ಪೈಥಾನ್‌ನಲ್ಲಿ, clear () ಮೆಥೆಡ್‌ನ್ನು ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾ'ಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set.clear()

ಉದಾಹರಣೆ :

```
Days = {"Monday","Tuesday", "Wednesday", "Thursday","Friday"}
print("\n printing the original set ... ")
print(Days)
Days.clear()
print("\n Printing the modified set...");
print(Days)
```

ಫಲಿತಾಂಶ :

printing the original set...

{'Monday','Wednesday','Friday','Tuesday','Thursday'}

Printing the modified set...

set()

union ()

ಪೈಥಾನ್‌ನಲ್ಲಿ, the union () ಮೆಥೆಡನ್ನು ಎರಡು ಸೆಟ್‌ಗಳನ್ನು ಸಂಯೋಜಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎರಡು ಸೆಟ್‌ಗಳ ಯೂನಿಯನ್ ಎರಡೂ ಸೆಟ್‌ಗಳಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set1.union (set2)

ಉದಾಹರಣೆ :

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1.union(Days2))
```

ಫಲಿತಾಂಶ :

{'Thr','Fri','Sun','Tue','Wed','Mon','Sat'}

intersection ()

ಪೈಥಾನ್‌ನಲ್ಲಿ, intersection () ಮೆಥೆಡ್‌ನ್ನು ಎರಡು ಸೆಟ್‌ಗಳ ಇಂಟರ್‌ಸ್ಷೆನ್‌ನ್ನು 'ಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎರಡು ಸೆಟ್‌ಗಳ ಇಂಟರ್‌ಸ್ಷೆನ್ ಎಂದರೆ ಎರಡೂ ಸೆಟ್‌ಗಳಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಇರುವ ಎಲಿಮೆಂಟ್‌ಗಳ ಸೆಟ್ ಆಗಿರುತ್ತದೆ

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set1.intersection (set2)

ಉದಾಹರಣೆ :

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
```

© NIMI NOT TO BE REPUBLISHED

print(Days1.intersection(Days2))

ಫಲಿತಾಂಶ :

{'Mon','Sat'}

difference ()

ಎರಡು ಸೆಟ್‌ಗಳಲ್ಲಿರುವ ವ್ಯತ್ಯಾಸವನ್ನು difference () ಮೆಥೆಡ್ ಉಪಯೋಗಿಸಿ 'ಂಡು ಹಿಡಿಯಲಾಗುತ್ತದೆ. ಸೆಟ್ 2 ನಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಸೆಟ್ 1 ರಿಂದ ತೆಗೆದುಹಾ"ದರೆ ಹೊಸ ಸೆಟ್ ಸಿಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set1.difference (set2)

ಉದಾಹರಣೆ :

Days1={"Mon","Tue","Wed","Sat"}

Days2={"Thr","Fri","Sat","Sun","Mon"}

print(Days1.difference(Days2))

ಫಲಿತಾಂಶ :

{'Tue','Wed'}

issubset()

ಒಂದು ಸೆಟ್‌ನಲ್ಲಿರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳು ಇನ್ನೊಂದು ಸೆಟ್‌ನಲ್ಲಿ ಇದ್ದರೆ issubset() ಮೆಥೆಡ್ True ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ. (ಆರ್ಗ್ಯುಮೆಂಟ್ ಆಗಿ ಪಾಸ್ ಮಾಡಲಾಗುತ್ತದೆ). ಇಲ್ಲದಿದ್ದರೆ False ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

set1.issubset (set2)

ಉದಾಹರಣೆ :

set1={1,2,3,4}

set2={1,2,3,4,5,6,7,8,9}

print(set1.issubset(set2))

print(set2.issubset(set1))

ಫಲಿತಾಂಶ :

True

False

ಪೈಥಾನ್‌ನಲ್ಲಿ ಡಿಕ್ಷನರಿ ಸಿಕ್ವೆನ್ಸ್ (Dictionary Sequence in Python)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು
- **ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು**
- **ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ನವೀಕರಿಸುವುದು**
- **ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು**
- **ಲೂಪ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು**
- **ಡಿಕ್ಷನರಿ ಫಂಕ್ಷನ್‌ಗಳು**

ಪೈಥಾನ್‌ನಲ್ಲಿ ಡ್ಷಿನರಿ ಎಂದರೆ key - value ಜೋಡಿಯ ಸಂಗ್ರಹವಾಗಿದೆ. ಇಲ್ಲಿ value ಎಂದರೆ ಯಾವುದೇ ಪೈಥಾನ್ ಆಬ್ಜೆಕ್ಟ್ ಆಗಿರುತ್ತದೆ ಮತ್ತು keys ಎಂದರೆ ಬದಲಾಯಿಸಲಾಗದ (immutable) ಪೈಥಾನ್ ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ ಉದಾಹರಣೆಗೆ, ಸಂಖ್ಯೆಗಳು, ಸ್ಟ್ರಿಂಗ್ ಅಥವಾ ಟ್ಯುಪಲ್.

ಡ್ಷಿನರಿಯನ್ನು ಹಲವಾರು key - value ಜೋಡಿಯನ್ನು ಉಪಯೋಗಿಸಿ ತಯಾರಿಸಲಾಗುತ್ತದೆ. ಇವುಗಳನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ(,) ಬೇರ್ಪಡಿಸಲಾಗುತ್ತದೆ ಮತ್ತು curly braces {}.ಗಳಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

Dict={key1:value1, key2:value2, ...}

ಉದಾಹರಣೆ :

student ={"Name":"Kiran","Age":22,"Regno": 562,"Branch":"CSE"}

print(student)

ಫಲಿತಾಂಶ:

{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}

ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು (Accessing the dictionary values)

ಲಿಸ್ಟ್ ಮತ್ತು ಟ್ಯುಪಲ್‌ಗಳಲ್ಲಿ ಇಂಡ್ಸೆನ್ನು ಬಳಸಿ ಡಾಟಾವನ್ನು ಪಡೆದುೊಳ್ಳಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಅದಾಗ್ಯೂ, ಡ್ಸ್ನರಿಯಲ್ಲಿ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು key ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಏಂದರೆ ಡ್ಸ್ನರಿಯಲ್ಲಿ key ಗಳು ವಿಶಿಷ್ಟವಾಗಿರುತ್ತವೆ.

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print("Name : ",student["Name"])
print("Age : ",student["Age"])
print("RegNo :",student["Regno"])
print("Branch : ",student["Branch"])
```

ಫಲಿತಾಂಶ :

```
Name : Kiran
Age :22
RegNo :562
Branch : CSE
```

ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ನವೀಕರಿಸುವುದು (Updating dictionary values)

ಡ್ಸ್ನರಿ ಮ್ಯೂಟೇಬಲ್ (ಬದಲಾಯಿಸಬಹುದಾದ) ಡಾಟಾಟೈಪ್ ಆಗಿರುವುದರಿಂದ ಮತ್ತು ಅದರ ಮೌಲ್ಯಗಳನ್ನು ನಿರ್ದಿಷ್ಟ key ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ನವೀಕರಿಸಬಹುದು.

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print("printing student data .... ")
print(student)
student["Name"]="Kishore"
print("printing updated data ....")
print(student)
```

ಫಲಿತಾಂಶ :

```
printing student data ....
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
printing updated data ....
{'Name':'Kishore','Age':22,'Regno':562,'Branch':'CSE'}
```

ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡುವುದು (Deleting dictionary values)

del "ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಡ್ಸ್ನರಿಯಲ್ಲಿರುವ ಅಂಶಗಳನ್ನು ಡಿಲೀಟ್ ಮಾಡಬಹುದು.

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print("printing student data .... ")
print(student)
del student["Branch"]
print("printing the modified information ")
print(student)
```

ಫಲಿತಾಂಶ :

```
printing student data ....
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
printing the modified information
{'Name':'Kiran','Age':22,'Regno':562}
```

ಲೂಪ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿ ಡಿಕ್ಷನರಿ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆಯುವುದು (Accessing the dictionary values using loops)

ಡ್ಸ್ನರಿಯನ್ನು for ಲೂಪ್ ಉಪಯೋಗಿಸಿಕೊಂಡು ಪುನರಾವರ್ತಿಸಬಹುದು. ನಾವು key ಗಳನ್ನು ಮಾತ್ರ ಅಥವಾ ಮೌಲ್ಯಗಳನ್ನು ಮಾತ್ರ ಪಡೆದುಕೊಳ್ಳಬಹುದು ಮತ್ತು keys ಮತ್ತು values ಎರಡನ್ನೂ ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಡಿಕ್ಷನರಿಯ ಎಲ್ಲಾ key ಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುವುದು (print all the keys of a dictionary)

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print("Keys are :")
for x in student:
print(x)
```

ಫಲಿತಾಂಶ :

```
Keys are :
Name
```

Age

Regno

Branch

ಡಿಕ್ಷನರಿಯ ಎಲ್ಲಾ values ಪ್ರಿಂಟ್ ಮಾಡುವುದು (print all the values of a dictionary)

ಉದಾಹರಣೆ :

```
print("values are :")
for x in student:
    print(student[x])
```

ಫಲಿತಾಂಶ :

values are :

Kiran

22

562

CSE

ಡಿಕ್ಷನರಿಯಲ್ಲಿರುವ ಎಲ್ಲ keys ಮತ್ತು values ಪ್ರಿಂಟ್ ಮಾಡುವುದು (print all the Keys & values of a dictionary)

ಉದಾಹರಣೆ :

```
print("Key and values are :")
forx,yin student.items():
    print(x,y)
```

ಫಲಿತಾಂಶ :

Key and values are :

Name Kiran

Age 22

Regno 562

Branch CSE

ಡಿಕ್ಷನರಿ ಫಂಕ್ಷನ್‌ಗಳು (Dictionary Functions)

ಪೈಥಾನ್, ಕೆಳಗಿನ ಇನ್ - ಬಿಲ್ಟ್ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಅವು ಯಾವವೆಂದರೆ :

- len()
- copy()
- get()
- keys()
- items()
- values()
- update()
- pop()
- clear()

len()

ಪೈಥಾನ್‌ನಲ್ಲಿ, len() ಫಂಕ್ಷನ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ಡಿಕ್ಷನರಿಯ ಲೆಂಥ್‌ನ್ನು ಕಂಡು ಹಿಡಿಯಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

len(dictionary)

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":
562,"Branch":"CSE"}
print("Length of Dictionary is:",len(student))
```

ಫಲಿತಾಂಶ :

Length of Dictionary is:4

copy()

ಇದು ನೀಡಲಾದ ಡಿಕ್ಷನರಿಯ ಇನ್ನೊಂದು ಪ್ರತಿಯನ್ನು (copy) ಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

dictionary.copy()

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":
562,"Branch":"CSE"}
student2=student.copy()
print(student2)
```

ಫಲಿತಾಂಶ :

{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}

© NIMI NOT TO BE REPUBLISHED

get()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯಲ್ಲಿರುವ ನಿರ್ದಿಷ್ಟ key ನ ಮೌಲ್ಯವನ್ನು ಪಡೆದುಕೊಳ್ಳಲು get() ಫಂಕ್ಷನ್ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
dictionary.get()
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print("Name is :",student.get("Name"))
print("RegNo is :",student.get("Regno"))
```

ಫಲಿತಾಂಶ :

```
Name is: Kiran
RegNo is:562
```

keys()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯಲ್ಲಿರುವ ಎಲ್ಲ key ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲು keys() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
dictionary.keys()
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
for x instudent.keys():
print(x)
```

ಫಲಿತಾಂಶ :

```
Name
Age
Regno
Branch
```

items()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯ ಹೊಸ ವೀಕ್ಷಣೆಯನ್ನು items() ಮೆಥೆಡ್ ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಈ ನೋಟವು key value ಟ್ಯುಪಲ್‌ಗಳ ಸಂಗ್ರಹವಾಗಿರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
dictionary.items()
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
for x instudent.items():
print(x)
```

ಫಲಿತಾಂಶ :

```
('Name','Kiran')
('Age',22)
('Regno',562)
('Branch','CSE')
```

values()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯಲ್ಲಿರುವ ಎಲ್ಲ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು values() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Dictionary.values()
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
for x instudent.values():
print(x)
```

Output:

ಫಲಿತಾಂಶ :

```
Kiran
22
562
CSE
```

update()

ಪೈಥಾನ್‌ನಲ್ಲಿ, update() ಮೆಥೆಡ್ key ಮತ್ತು value ಗಳ ಜೋಡಿಗಳಿಂದ ಡಿಕ್ಷನರಿಯನ್ನು ನವೀಕರಿಸುತ್ತದೆ. ಇದು ಡಿಕ್ಷನರಿಯಲ್ಲಿ ಇಲ್ಲದಿರುವ key / value ಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ. ಈಗಾಗಲೇ ಡಿಕ್ಷನರಿಯಲ್ಲಿ

© NIMI NOT TO BE REPUBLISHED

key / value ಗಳು ಇದ್ದರೆ ಅವುಗಳನ್ನು ನವೀಕರಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Dictionary.update({key:value,.....})
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
student.update({"Regno":590})
student.update({"phno":56895})
print(student)
```

ಫಲಿತಾಂಶ :

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE','phno':56895}
```

pop()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯಲ್ಲಿ ಇರುವ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕಲು pop() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ನಿರ್ದಿಷ್ಟ key ಯೊಂದಿಗೆ ಸಂಬಂಧಪಟ್ಟ ಎಲಿಮೆಂಟನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

ಡಿಕ್ಷನರಿಯಲ್ಲಿ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ key ಇದ್ದರೆ, ಅದನ್ನು ತೆಗೆದು ಹಾಕುತ್ತದೆ ಮತ್ತು ಅದರ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಡಿಕ್ಷನರಿಯಲ್ಲಿ ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ key ಇಲ್ಲದಿದ್ದರೆ, ಅದು KeyError ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Dictionary.remove(key)
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
student.pop('Age')
print(student)
student.pop('hallno')
print(student)
```

ಫಲಿತಾಂಶ :

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
KeyError:'hallno'
```

clear()

ಪೈಥಾನ್‌ನಲ್ಲಿ, ಡಿಕ್ಷನರಿಯಲ್ಲಿ ಇರುವ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಡಿಲಿಟ್ ಮಾಡಲು clear() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
Dictionary.clear()
```

ಉದಾಹರಣೆ :

```
student = {"Name":"Kiran","Age":22,"Regno":562,"Branch":"CSE"}
print(student)
student.clear()
print(student)
```

ಫಲಿತಾಂಶ :

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
{}
```

ಪೈಥಾನ್ ಅರೇಗಳು (Python Arrays)

ಉದ್ದೇಶಗಳು: ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು.

• ಅರೇಯ ಎಲಿಮೆಂಟ್‌ಗಳು

> **ಸೂಚನೆ : ಪೈಥಾನ್, ಅರೇಗಳಿಗೆ ಬಿಲ್ಟ್- ಇನ್ ಬೆಂಬಲವನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ, ಆದರೆ ಅದರ ಬದಲಾಗಿ ಪೈಥಾನ್ ಲಿಸ್ಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.**

> **ಅದಾಗ್ಯೂ, ಪೈಥಾನ್‌ನಲ್ಲಿ ಅರೇಗಳೊಂದಿಗೆ ಕೆಲಸ ಮಾಡಲು ನೀವು NumPy ಲೈಬ್ರರಿಯಂತಹ ಲೈಬ್ರರಿಯನ್ನು ಆಮದು ಮಾಡಿಕೊಳ್ಳಬೇಕಾಗುತ್ತದೆ.**

ಅರೇಗಳು (Arrays)

> **ಸೂಚನೆ : ಈ ಪುಟವು ನಿಮಗೆ ಲಿಸ್ಟ್‌ಗಳನ್ನು ಮತ್ತು ಅರೇಗಳನ್ನು ಹೇಗೆ ಉಪಯೋಗಿಸಬೇಕು ಎಂಬುದನ್ನು ತಿಳಿಸುತ್ತದೆ.**

ಒಂದೇ ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಅನೇಕ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಅರೇಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ :

ಕಾರ್ ಹೆಸರುಗಳನ್ನು ಹೊಂದಿರುವ ಅರೇಯನ್ನು ರಚಿಸಿ:

cars = ["Ford", "Volvo", "BMW"]

ಅರೇ ಎಂದರೇನು?(What is an Array?)

ಅರೇ ಒಂದು ವಿಶೇಷ ವೇರಿಯೇಬಲ್ ಆಗಿದೆ. ಇದು ಒಂದು ಸಮಯದಲ್ಲಿ ಒಂದ"'ಂತ ಹೆಚ್ಚು ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು.

ನೀವು ಅಂಶಗಳ ಲಿಸ್ಟ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ (ಉದಾಹರಣೆಗೆ, ಾರಿನ ಹೆಸರುಗಳ ಪಟ್ಟಿ) ಾರಿನ ಹೆಸರುಗಳನ್ನು ಒಂದು ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸುವುದು ಈ ರೀತಿಯಾಗಿ ಾಣಿಸಬಹುದು :

car1 = "Ford"

car2 = "Volvo"

car3 = "BMW"

ಅದಾಗ್ಯೂ, ನೀವು ಾರ್‌ಗಳ ಮೂಲ' ಲೂಪ್ ಮಾಡಲು ಬಯಸಿದರೆ, ಮತ್ತು ಒಂದು ನಿರ್ದಿಷ್ಟ ಾರನ್ನು ಹುಡು'ಲು ಪ್ರಯತ್ನಿಸಿದರೆ ? ಮತ್ತು ನೀವು 3 ಾರುಗಳ ಬದಲಾಗಿ 300 ಾರುಗಳನ್ನು ಹೊಂದಿದ್ದರೆ?

ಇದ" ಪರಿಹಾರ ಅರೆಗಳಾಗಿವೆ.!

ಒಂದು ಅರೇ ಒಂದೇ ಹೆಸರಿನಿಂದ ಹಲವಾರು ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲದು, ಮತ್ತು ಅದರ ಇಂಡ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಉಲ್ಲೇಖಿಸುವ ಮೂಲ' ನೀವು ಆ ಮೌಲ್ಯಗಳನ್ನು ಪಡೆದುೊಳ್ಳಬಹುದು.

ಅರೇಯ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು.(Access the elements of an Array)

ನೀವು ಒಂದು ಅರೇಯ ಎಲೆಮೆಂಟ್‌ನ್ನು ಅದರ ಇಂಡ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಸೂಚಿಸುವ ಮೂಲ' ಪಡೆದುೊಳ್ಳಬಹುದು.

ಉದಾಹರಣೆ

ಮೊದಲನೇ ಅರೇ ಎಲೆಮೆಂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಪಡೆಯಿರಿ:

x = cars[0]

ಉದಾಹರಣೆ

ಅರೇನಲ್ಲಿನ ಮೊದಲನೇ ಎಲೆಮೆಂಟ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು

cars[0] = "Toyota"

ಅರೇಯ ಲೆಂಥ್ ಕಂಡುಹಿಡಿಯುವುದು(The Length of an Array)

len() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ .ಅರೇಯ ಲೆಂಥನ್ನು 'ಂಡುಹಿಡಿಯಬಹುದು (ಅರೇಯಲ್ಲಿರುವ ಎಲೆಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆ)

ಉದಾಹರಣೆ :

cars ಅರೇಯಲ್ಲಿರುವ ಎಲೆಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ:

x = len(cars)

ಸೂಚನೆ : ಲೆಂಥ್ ಯಾವಾಗಲೂ ಅರೇಯ ಗರಿಷ್ಟ ಇಂಡೆಕ್ಸ್‌ಗಿಂತಲೂ ಒಂದು ಸಂಖ್ಯೆ ಹೆಚ್ಚಾಗಿರುತ್ತದೆ.

ಅರೇ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಲೂಪ್ ಮಾಡುವುದು (Looping Array Elements)

for in ಲೂಪನ್ನು ಉಪಯೋಗಿಸಿ ಅರೇಯಲ್ಲಿರುವ ಎಲ್ಲ ಎಲೆಮೆಂಟ್‌ಗಳ ಮೂಲ' ಲೂಪ್ ಮಾಡಬಹುದು.

ಉದಾಹರಣೆ :

cars ಅರೇಯಲ್ಲಿರುವ ಪ್ರತಿಯೊಂದು ಅಂಶವನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುವುದು:

```
for x in cars:
print(x)
```

ಅರೇ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಸೇರಿಸುವುದು (Adding Array Elements)

append() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಅರೇಗೆ ಎಲೆಮೆಂಟ್‌ನ್ನು ಸೇರಿಸಬಹುದು.

ಉದಾಹರಣೆ :

cars ಅರೇಗೆ ಮತ್ತೊಂದು ಎಲೆಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುವುದು :

cars.append("Honda")

ಅರೇ ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುವುದು (Removing Array Elements)

pop() ಮೆಥೆಡ್ ಉಪಯೋಗಿಸಿ ಅರೇದಿಂದ ಎಲೆಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ'ಬಹುದು.

ಉದಾಹರಣೆ :

cars ಅರೇದಿಂದ ಎರಡನೇ ಎಲೆಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕುವುದು :

cars.pop(1)

ಒಂದು ಅರೇದಿಂದ ಎಲೆಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾ'ಲು remove() ಮೆಥೆಡ್‌ನ್ನು ಸಹ ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ :

"Volvo" ಎನ್ನುವ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದ ಎಲೆಮೆಂಟ್‌ನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಮೆಥೆಡ್	ವಿವರಣೆ
append()	ಲಿಸ್ಟ್‌ನ ಕೊನೆಗೆ ಎಲಿಮೆಂಟನ್ನು ಸೇರಿಸುತ್ತದೆ
clear()	ಲಿಸ್ಟ್‌ನಿಂದ ಎಲ್ಲ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ
copy()	ಲಿಸ್ಟ್‌ನ ಪ್ರತಿಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
count()	ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯದ ಎಲಿಮೆಂಟ್‌ಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
extend()	ಪ್ರಸ್ತುತ ಲಿಸ್ಟ್‌ನ ಕೊನೆಗೆ ಇನ್ನೊಂದು ಲಿಸ್ಟ್‌ನ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು (ಅಥವಾ ಯಾವುದೇ ಪುನರಾವರ್ತಿತ) ಸೇರಿಸುತ್ತದೆ.
index()	ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯದ ಮೊದಲನೆ ಎಲಿಮೆಂಟ್‌ನ ಇಂಡೆಕ್ಸ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.
insert()	ನಿರ್ದಿಷ್ಟ ಸ್ಥಾನದಲ್ಲಿ ಎಲಿಮೆಂಟ್‌ನ್ನು ಸೇರಿಸುತ್ತದೆ.
pop()	ನಿರ್ದಿಷ್ಟ ಸ್ಥಾನದಲ್ಲಿನ ಎಲಿಮೆಂಟ್‌ನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.
remove()	ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯದ ಮೊದಲನೇ ಎಲಿಮೆಂಟನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.
reverse()	ಲಿಸ್ಟ್‌ನ ಕ್ರಮವನ್ನು ಹಿಮ್ಮುಖಗೊಳಿಸುತ್ತದೆ.
sort()	ಲಿಸ್ಟ್‌ನ್ನು ವಿಂಗಡಿಸುತ್ತದೆ.

ತೆಗೆದುಹಾಕುವುದು :

```
cars.remove( "Volvo")
```

ಸೂಚನೆ : remove() ಮೆಥೆಡ್ ಲಿಸ್ಟ್‌ನಲ್ಲಿರುವ ಮೊದಲು ಕಂಡುಬರುವ

ನಿರ್ದಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಮಾತ್ರ ತೆಗೆದುಹಾಕುತ್ತದೆ.

ಅರೇ ಮೆಥೆಡ್‌ಗಳು (Array Methods)

ಪೈಥಾನ್ ಬಿಲ್ಟ್ - ಇನ್ ಫಂಕ್ಷನ್‌ಗಳ ಸೆಟ್‌ನ್ನು ಹೊಂದಿದೆ. ಅವುಗಳನ್ನು ನೀವು ಲಿಸ್ಟ್ /ಅರೇಗಳ ಜೊತೆಗೆ ಉಪಯೋಗಿಸಬಹುದು.

ಸೂಚನೆ : ಪೈಥಾನ್, ಅರೇಗಳಿಗೆ ಬಿಲ್ಟ್- ಇನ್ ಬೆಂಬಲವನ್ನು ಹೊಂದಿರುವುದಿಲ್ಲ,

ಆದರೆ ಅದರ ಬದಲಾಗಿ ಪೈಥಾನ್ ಲಿಸ್ಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಮಾಡ್ಯೂಲ್‌ಗಳನ್ನು ಮತ್ತು ಟೂಲ್‌ಗಳನ್ನು ಬಳಸಿ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುವುದು (Perform Operations Using Modules and Tools)

ಉದ್ದೇಶಗಳು: ಈ ಅಭ್ಯಾಸದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಮಾಡ್ಯೂಲನ್ನು ರಚಿಸುವುದು
- ಮಾಡ್ಯೂಲನ್ನು ಲೋಡ್ ಮಾಡುವುದು
- ಮಾಡ್ಯೂಲನ್ನು ಮರುಹೆಸರಿಸುವುದು

ಪೈಥಾನ್‌ನಲ್ಲಿ ಮಾಡ್ಯೂಲ್‌ಗಳು (Modules in Python)

ಪೈಥಾನ್‌ನಲ್ಲಿ ಮಾಡ್ಯೂಲ ಎಂದರೆ ಪೈಥಾನ್ ಪ್ರೊಗ್ರಾಂ ಫೈಲ್ ಆಗಿದೆ. ಇದು ಪೈಥಾನ್ ಫಂಕ್ಷನ್‌ಗಳು, ಕ್ಲಾಸ್, ಅಥವಾ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಒಂದು ಪೈಥಾನ್ ಕೋಡ್ ಆಗಿರುತ್ತದೆ. ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, .py ವಿಸ್ತರಣೆಯೊಂದಿಗೆ ಸೇವ್ ಮಾಡಲಾದ ಪೈಥಾನ್ ಕೋಡ್ ಫೈಲ್‌ನ್ನು ಮಾಡ್ಯೂಲ್ ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ.

ಪೈಥಾನ್‌ನಲ್ಲಿ ಮಾಡ್ಯೂಲ್‌ಗಳು ನಮಗೆ ಕೋಡ್ ಅನ್ನು ತಾರ್ಕಿಕ ರೀತಿಯಲ್ಲಿ ಸಂಘಟಿಸಲು ಅನುಮತಿಯನ್ನು ನೀಡುತ್ತವೆ. ಒಂದು ಮಾಡ್ಯೂಲ್‌ನ ಕಾರ್ಯವನ್ನು ಇನ್ನೊಂದು ಮಾಡ್ಯೂಲ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲು ನಿರ್ದಿಷ್ಟ ಮಾಡ್ಯೂಲನ್ನು ನಾವು ಆಮದು ಮಾಡಿಕೊಳ್ಳಬೇಕಾಗುತ್ತದೆ.

ಮಾಡ್ಯೂಲನ್ನು ರಚಿಸುವುದು (Creating Module)

ಉದಾಹರಣೆ: demo.py

```
# Python Module example
def sum(a,b):
return a+b
def sub(a,b):
return a-b
def mul(a,b):
return a*b
def div(a,b):
return a/b
```

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ ನಾವು 4 ಫಂಕ್ಷನ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿದ್ದೇವೆ. demo ಹೆಸರಿನ ಮಾಡ್ಯೂಲ್‌ನಲ್ಲಿ sum(), sub(), mul() and div() ಫಂಕ್ಷನ್‌ಗಳಿವೆ.

ನಮ್ಮ ಪೈಥಾನ್ ಕೋಡ್‌ನಲ್ಲಿ ಮಾಡ್ಯೂಲನ್ನು ಲೋಡ್ ಮಾಡುವುದು.

ಒಂದು ಮಾಡುಲ್‌ನ ಕಾರ್ಯವನ್ನು ಬಳಸಲು ನಮಗೆ ನಮ್ಮ ಪೈಥಾನ್ ಕೋಡ್‌ನಲ್ಲಿ ಮಾಡ್ಯೂಲನ್ನು ಲೋಡ್ ಮಾಡುವ ಅಗತ್ಯವಿರುತ್ತದೆ. ಕೆಳಗೆ ವ್ಯಾಖ್ಯಾನಿಸಿದಂತೆ ಪೈಥಾನ್ ಎರಡು ರೀತಿಯ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

1 import statement

2 from-import statement

1 ಇಂಪೋರ್ಟ್ ಸ್ಟೇಟಮೆಂಟ್ (import statement)

import statement ನ್ನು ಒಂದು ಮಾಡ್ಯೂಲ್‌ನ ಎಲ್ಲ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಇನ್ನೊಂದೆ" ಇಂಪೋರ್ಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇಲ್ಲಿ ನಾವು ಗಮನಿಸಬೇಕಾದ ಅಂಶವೆಂದರೆ, ಯಾವುದೇ ಪೈಥಾನ್ ಸೋರ್ಸ್ ಫೈಲ್‌ನ್ನು ಇನ್ನೊಂದು ಪೈಥಾನ್ ಸೋರ್ಸ್ ಫೈಲಿಗೆ, ಆ ಫೈಲನ್ನು

ನಾವು ಅನೇಕ ಮಾಡ್ಯೂಲ್‌ಗಳನ್ನು ಒಂದೇ import ಸ್ಟೇಟಮೆಂಟ್ ಮೂಲಕ ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳಬಹುದು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
import module1,module2..
```

ಉದಾಹರಣೆ :

```
import demo #importing entire Module
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",demo.sum(a,b))
print("Sub is :",demo.sub(a,b))
print("Mul is :",demo.mul(a,b))
print("Div is :",demo.div(a,b))
```

© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ :

Enter a :12

Enter b :6

Sum is:18

Sub is:6

Mul is:72

Div is:2.0

2 ಫ್ರಾಮ್- ಇಂಪೋರ್ಟ್ ಸ್ಟೇಟಮೆಂಟ್ (from-import statement)

namespace ನಲ್ಲಿ ಇಡೀ ಮಾಡ್ಯೂಲನ್ನು ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳುವ ಬದಲು, ಮಾಡ್ಯೂಲ್‌ನ ನಿರ್ದಿಷ್ಟ ಅಟ್ರಿಬ್ಯೂಟ್‌ಗಳನ್ನು ಮಾತ್ರ ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳುವ ಅನುಕೂಲತೆಯನ್ನು ಪೈಥಾನ್ ನಮಗೆ ಒದಗಿಸುತ್ತದೆ. ಇದನ್ನು from -import ಸ್ಟೇಟಮೆಂಟ್ ಉಪಯೋಗಿಸಿ ಮಾಡಲಾಗುತ್ತದೆ. ಈ ಸಂದರ್ಭದಲ್ಲಿ dot ಆಪರೇಟರ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುವುದಿಲ್ಲ.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
from module-name import*
```

ಉದಾಹರಣೆ :

```
from demo import*
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",demo.sum(a,b))
print("Sub is :",demo.sub(a,b))
print("Mul is :",demo.mul(a,b))
print("Div is :",demo.div(a,b))
```

ಫಲಿತಾಂಶ :

Enter a :12

Enter b :6

Sum is:18

Sub is:6

Mul is:72

Div is:2.0

ಇಡೀ ಮಾಡ್ಯೂಲನ್ನು ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳುವ ಬದಲು ನಾವು ಒಂದು ಮಾಡ್ಯೂಲ್‌ನಿಂದ ನಿರ್ದಿಷ್ಟ ಫಂಕ್ಷನ್‌ನ್ನು ಮಾತ್ರ ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳಬಹುದು. ಉದಾಹರಣೆಗೆ :

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
from module-name import function1,function2ಇ
```

ಉದಾಹರಣೆ :

```
from demo importsub,mul
#importing specific functionality from Module
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sub is :",sub(a,b))
print("Mul is :",mul(a,b))
```

ಫಲಿತಾಂಶ :

Enter a :12

Enter b :6

Sub is:6

Mul is:72

ಮಾಡ್ಯೂಲನ್ನು ಮರುಹೆಸರಿಸುವುದು (Renaming a module)

ಪೈಥಾನ್ ನಮಗೆ ಕೆಲವು ಮಾಡ್ಯೂಲನ್ನು ನಿರ್ದಿಷ್ಟ ಹೆಸರಿನಿಂದ ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳಲು ಅನುಕೂಲತೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ನಿರ್ದಿಷ್ಟ ಮಾಡ್ಯೂಲನ ಬದಲಾಗಿ ನಾವು ಈ ಹೆಸರನ್ನು ನಮ್ಮ ಪೈಥಾನ್ ಸೋರ್ಸ್ ಫೈಲ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು.

ಸಿಂಟ್ಯಾಕ್ಸ್ :

```
import module-name as specific-name
```

ಉದಾಹರಣೆ :

```
import demo as c
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",c.sum(a,b))
print("Sub is :",c.sub(a,b))
```

ಫಲಿತಾಂಶ :

Enter a :25

Enter b :12

Sum is:37

Sub is:13

© NIMI NOT TO BE REPUBLISHED

ಪೈಥಾನ್ ಇಟರೇಟರ್‌ಗಳು (Python Iterators)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಇಟರೇಟರ್ ಮೂಲಕ ಲೂಪಿಂಗ್ ಮಾಡುವುದು**
- **ಇಟರೇಟರ್‌ನ್ನು ರಚಿಸುವುದು**
- **ಇಟರೇಷನ್‌ನ್ನು ನಿಲ್ಲಿಸುವುದು**

ಪೈಥಾನ್ ಇಟರೇಟರ್‌ಗಳು(Python Iterators)

ಇಟರೇಟರ್ ಎಣಿ ಮಾಡಬಹುದಾದ ಮೌಲ್ಯಗಳ ಸಂಖ್ಯೆಯನ್ನು ಹೊಂದಿರುವ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ.

ಇಟರೇಟರ್ ಪುನರಾವರ್ತಿಸಬಹುದಾದ ಎಂದರೆ ಎಲ್ಲಾ ಮೌಲ್ಯಗಳ ಮೂಲ ಸಂಚರಿಸುವ ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ.

ತಾಂತ್ರಿಕವಾಗಿ, ಪೈಥಾನ್‌ನಲ್ಲಿ ಇಟರೇಟರ್ ಎಂದರೆ ಇಟರೇಟರ್ ಪ್ರೊಟೊಕಾಲನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುವ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ. ಇದು __iter__() ಮತ್ತು __next__() ಮೆಥೆಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಇಟರೇಟರ್ ವಿರುದ್ಧ ಇಟರೇಬಲ್ (Iterator vs Iterable)

ಲಿಸ್ಟ್‌ಗಳು, ಟ್ಯುಪಲ್‌ಗಳು, ಡಿಕ್ಷನರಿಗಳು ಮತ್ತು ಸೆಟ್‌ಗಳು ಇ ವೆಲ್ಲವೂ ಇಟರೇಬಲ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ. ಇವೆಲ್ಲವೂ ಇಟರೇಬಲ್ ಕಂಟೇನರ್ ಗಳಾಗಿದ್ದು ಇವುಗಳಿಂದ ನೀವು ಇಟರೇಟರ್‌ನ್ನು ಪಡೆಂ ಯಬಹುದು.

ಈ ಎಲ್ಲಾ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು iter() ಮೆಥೆಡ್‌ನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಇದನ್ನು ಇಟರೇಟರ್‌ನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

ಒಂದು ಟ್ಯುಪಲ್‌ನಿಂದ ಇಟರೇಟರ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು ಮತ್ತು ಪ್ರತಿ ಮೌಲ್ಯವನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುವುದು:

```
mytuple = (“apple”, “banana”, “cherry”)

myit = iter(mytuple)

print(next(myit))

print(next(myit))

print(next(myit))
```

ಸ್ಟ್ರಿಂಗ್‌ಗಳೂ ಸಹ ಇಟರೇಬಲ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ, ಮತ್ತು ಇಟರೇಟರ್‌ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತವೆ.

ಉದಾಹರಣೆ

ಸ್ಟ್ರಿಂಗ್‌ಗಳೂ ಸಹ ಇಟರೇಬಲ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳಾಗಿವೆ, ಇವು ಅಕ್ಷರಗಳ ಸರಣಿಯನ್ನು ಹೊಂದಿರುತ್ತವೆ.

```
mystr = “banana”

myit = iter(mystr)

print(next(myit))

print(next(myit))

print(next(myit))

print(next(myit))

print(next(myit))

print(next(myit))
```

ಇಟರೇಟರ್ ಮೂಲಕ ಲೂಪಿಂಗ್ ಮಾಡುವುದು (Looping Through an Iterator)

ಒಂದು ಇಟರೇಬಲ್ ಆಬ್ಜೆಕ್ಟ್ ಮೂಲಕ ಇಟರೇಟ್ ಮಾಡಲು ನೀವು for ಲೂಪನ್ನು ಸಹ ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ

ಒಂದು ಟ್ಯುಪಲ್‌ನ ಮೌಲ್ಯಗಳನ್ನು ಇಟರೇಟ್ ಮಾಡಲು :

```
mytuple = (“apple”, “banana”, “cherry”)

for x in mytuple:

print(x)
```

ಉದಾಹರಣೆ

ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನಲ್ಲಿರುವ ಅಕ್ಷರಗಳನ್ನು ಇಟರೇಟ್ ಮಾಡಲು :

```
mystr = “banana”

for x in mystr:

print(x)
```

ವಾಸ್ತವವಾಗಿ for ಲೂಪ್ ಇಟರೇಟರ್ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸುತ್ತದೆ. ಮತ್ತು ಪ್ರತಿ ಲೂಪ್‌ಗೆ next() ಮೆಥೆಡ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ.

ಇಟರೇಟರ್‌ನ್ನು ರಚಿಸುವುದು (Create an Iterator)

ಆಬ್ಜೆಕ್ಟ್/ಕ್ಲಾಸ್ ಅನ್ನು ಇಟರೇಟರ್ ಆಗಿ ರಚಿಸಲು ನೀವು ನಿಮ್ಮ ಆಬ್ಜೆಕ್ಟ್‌ಗೆ __iter__() ಮತ್ತು__next__() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ಪೈಥಾನ್ ಕ್ಲಾಸ್/ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಪಾಠದಲ್ಲಿ ಕಲಿತಿರುವಂತೆ, ಎಲ್ಲಾ ಕ್ಲಾಸ್‌ಗಳು __init__() ಎನ್ನುವ ಫಂಕ್ಷನ್‌ನ್ನು ಹೊಂದಿರುತ್ತವೆ, ಇದು

ನಿಮಗೆ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸುವಾಗ ೆಲವನ್ನು ಪ್ರಾರಂಭಿಸಲು (initializing) ಅನುಮತಿ ನೀಡುತ್ತದೆ.

__iter__() ಮೆಥೆಡ್ ಸಹ ಇದೇ ರೀತಿ ಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ, ನೀವು ೆಲವು ಾರ್ಯಗಳನ್ನು (initializing ಇತ್ಯಾದಿಗಳು) ಮಾಡಬಹುದು, ಆದರೆ ಯಾವಾಗಲೂ ಇಟರೇಟರ್ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಸ್ವತ' ಹಿಂತಿರುಗಿಸಬೇಿು.

__next__() ಮೆಥೆಡ್ ಸಹ ನಿಮಗೆ ಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ ಮತ್ತು ಮುಂದಿನ ಅಂಶವನ್ನು ಅನು್ರಮದಲ್ಲಿ ಹಿಂತಿರುಗಿಸಬೇಿು.

ಉದಾಹರಣೆ

1 ರಿಂದ ಪ್ರಾರಂಭಿಸಿ, ಸಂಖ್ಯೆಗಳನ್ನು ಹಿಂತಿರುಗಿಸುವ ಇಟರೇಟರ್‌ನ್ನು ರಚಿಸಿ ಮತ್ತು ಪ್ರತಿ ಅನು್ರಮವು ಒಂದರಿಂದ ಹೆಚ್ಚಾಗುತ್ತದೆ (1,2,3,4,5 ಇತ್ಯಾದಿಗಳನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ)

```
class MyNumbers:
def __iter__(self):
self.a = 1
return self
def __next__(self):
x = self.a
self.a += 1
return x
myclass = MyNumbers()
myiter = iter(myclass)
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

ಇಟರೇಷನ್‌ನ್ನು ನಿಲ್ಲಿಸುವುದು(StopIteration)

ನೀವು ಸಾ್ಷ್ಟು next() ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿದ್ದರೆ, ಅಥವಾ ಅದನ್ನು for ಲೂಪ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಿದ್ದರೆ, ಮೇಲಿನ ಉದಾಹರಣೆಯು ಶಾಶ್ವತವಾಗಿ ಮುಂದುವರೆಯುತ್ತದೆ,

ಇಟರೇಷನ್ ಶಾಶ್ವತವಾಗಿ ಮುಂದುವರೆಯುವುದನ್ನು ತಡೆಯಲು ನಾವು StopIteration ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯಲ್ಲಿ ಇಟರೇಷನ್ ಆದ ಮೇಲೆ, __next__() ಮೆಥೆಡ್‌ನಲ್ಲಿ ನಾವು ದೋಷವನ್ನು ತೋರಿಸಲು ಾರ್ಯವನ್ನು ೊನೆಗಾಣಿಸುವ ಂಡಿಷನ್‌ನ್ನು ಸೇರಿಸಬಹುದು:

ಉದಾಹರಣೆ

20 ಇಟರೇಷನ್‌ಗಳ ನಂತರ ನಿಲ್ಲಿಸುವುದು:

```
class MyNumbers:
def __iter__(self):
self.a = 1
return self
def __next__(self):
if self.a <= 20:
x = self.a
self.a += 1
return x
else:
raise StopIteration
myclass = MyNumbers()
myiter = iter(myclass)
for x in myiter:
print(x)
```

ಪೈಥಾನ್‌ನಲ್ಲಿ ಮ್ಯಾಥ್ (Python Math)

ಪೈಥಾನ್, ಬಿಲ್ಟ್ - ಇನ್ ಮ್ಯಾಥ್ ಫಂ್ಷನ್‌ಗಳ ಗುಂಪನ್ನು ಹೊಂದಿದೆ. ಇದು ವಿಸ್ತಾರವಾದ math ಮಾಡ್ಯೂಲ್‌ನ್ನೂ ಒಳಗೊಂಡಿದೆ. ಇದು ನಿಮಗೆ ಸಂಖ್ಯೆಗಳ ಮೇಲೆ ಗಣಿತದ ೆಲಸಗಳನ್ನು ಮಾಡಲು ಅನುವು ಮಾಡಿೊಡುತ್ತದೆ.

ಬಿಲ್ಟ್ - ಇನ್ ಮ್ಯಾಥ್ ಫಂಕ್ಷನ್‌ಗಳು (Built-in Math Functions)

ಒಂದು ಇಟರೇಬಲ್‌ನಲ್ಲಿ ನಿಷ್ಟ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯವನ್ನು ಂಡು ಹಿಡಿಯಲು min() ಮತ್ತು max() ಫಂ್ಷನ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

```
x = min(5, 10, 25)
y = max(5, 10, 25)
print(x)
```

© NIMI NOT TO BE REPUBLISHED

```
print(y)
```

ಒಂದು ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಸಂಪೂರ್ಣ ಸಂಖ್ಯೆಯನ್ನು (ಧನಾತ್ಮ) ಕಂಡುಹಿಡಿಯಲು abs() ಫಂಕ್ಷನ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

```
x = abs(-7.25)
print(x)
```

pow(x, y) ಫಂಕ್ಷನ್‌ನ್ನು, x ದ ಘಾತಂಕ y ಇದರ ಮೌಲ್ಯವನ್ನು ಕಂಡುಹಿಡಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

4 ರ ಘಾತಂಕ 3 ಇದರ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ(4 * 4 * 4 ಇದರಂತೆ):

```
x = pow(4, 3)
print(x)
```

Math ಮಾಡ್ಯೂಲ್ (The Math Module)

ಪೈಥಾನ್ Math ಎನ್ನುವ ಬಿಲ್ಟ್ - ಇನ್ ಮಾಡ್ಯೂಲ್‌ನ್ನು ಸಹ ಹೊಂದಿದೆ. ಇದು ಗಣಿತದ ಕಾರ್ಯಗಳ ಪಟ್ಟಿಯನ್ನು ವಿಸ್ತರಿಸುತ್ತದೆ.

ಇದನ್ನು ಉಪಯೋಗಿಸಲು, ನೀವು math ಮಾಡ್ಯೂಲನ್ನು ಇಂಪೋರ್ಟ್ ಮಾಡಿಕೊಳ್ಳಬೇಕು :

```
import math
```

ನೀವು math ಮಾಡ್ಯೂಲ್‌ನ್ನು ಆಮದು ಮಾಡಿಕೊಂಡಾಗ, ಆ ಮಾಡ್ಯೂಲ್‌ನ ಮೆಥೆಡ್‌ಗಳನ್ನು ಮತ್ತು ಕಾನ್‌ಸ್ಟಂಟ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಲು ಪ್ರಾರಂಭಿಸಬಹುದು.

ಉದಾಹರಣೆಗೆ math.sqrt() ಮೆಥೆಡ್, ಒಂದು ಸಂಖ್ಯೆಯ ಸ್ಕ್ವೇರ್ ರೂಟನ್ನು (square root) ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ

```
import math
x = math.sqrt(64)
print(x)
```

math.ceil() ಮೆಥೆಡ್ ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಅದರ ಅತ್ಯಂತ ಹತ್ತಿರದ ನಂತರದ ಪೂರ್ಣಾಂಕಕ್ಕೆ ಪೂರ್ಣಗೊಳಿಸುತ್ತದೆ. ಮತ್ತು math.floor() ಮೆಥೆಡ್ ಒಂದು ಸಂಖ್ಯೆಯನ್ನು ಅದರ ಅತ್ಯಂತ ಹತ್ತಿರದ ಕೆಳಗಿನ ಪೂರ್ಣಾಂಕಕ್ಕೆ ಪೂರ್ಣಗೊಳಿಸುತ್ತದೆ ಮತ್ತು ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ.

ಉದಾಹರಣೆ

```
import math
x = math.ceil(1.4)
y = math.floor(1.4)
print(x) # 2 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
print(y) # 1 ನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ
```

math.pi constant, PI ನ ಮೌಲ್ಯವನ್ನು (3.14...) ಎಂದು ಉತ್ತರಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ

```
import math
x = math.pi
print(x)
```

ಸಂಪೂರ್ಣ Math Module Reference

ನಮ್ಮ Math Module Reference ನಲ್ಲಿ ನೀವು ಎಲ್ಲ ಮೆಥೆಡ್‌ಗಳ ಮತ್ತು ಕಾನ್ಸ್ಟಂಟ್‌ಗಳ ಸಂಪೂರ್ಣ ಉಲ್ಲೇಖವನ್ನು ಕಾಣಬಹುದು. ಇವೆಲ್ಲವೂ Math ಮಾಡ್ಯೂಲ್‌ಗೆ ಸೇರಿದವುಗಳಾಗಿವೆ.

ಪೈಥಾನ್ ಬಳಕೆದಾರರ ಇನ್ಪುಟ್ (Python User Input)

ಬಳಕೆದಾರರ ಇನ್ಪುಟ್ (User Input)

ಪೈಥಾನ್ ಬಳಕೆದಾರರಿಗೆ ಇನ್ಪುಟ್ ನೀಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದರ ಅರ್ಥ ನಾವು ಬಳಕೆದಾರರಿಗೆ ಇನ್ಪುಟ್‌ನ್ನು ಕೇಳಲು ಸಾಧ್ಯವಾಗುತ್ತದೆ.

ಪೈಥಾನ್ 3.6 ನಲ್ಲಿನ ಮೆಥೆಡ್‌ಗಳು ಪೈಥಾನ್ 2.7 ಗಿಂತ ಸ್ವಲ್ಪ ಭಿನ್ನವಾಗಿರುತ್ತದೆ.

ಪೈಥಾನ್ 3.6 input() ಮೆಥೆಡ್‌ನ್ನು ಬಳಸುತ್ತದೆ.

ಪೈಥಾನ್ 2.7 raw_input() ಮೆಥೆಡ್‌ನ್ನು ಬಳಸುತ್ತದೆ.

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು username ನ್ನು ಕೇಳುತ್ತದೆ, ಮತ್ತು ನೀವು username ಅನ್ನು ನಮೂದಿಸಿದಾಗ , ಅದು ಸ್ಕ್ರೀನ್ ಮೇಲೆ ಪ್ರಿಂಟ್ ಆಗುತ್ತದೆ.

Python 3.6

```
username = input("Enter username:")
print("Username is: " + username)
```

Run Example

Python 2.7

```
username = raw_input("Enter username:")
print("Username is: " + username)
```

Run Example

input() ಫಂಕ್ಷನ್ ಬಂದಾಗ ಪೈಥಾನ್ ಕಾರ್ಯನಿರ್ವಹಿಸುವುದನ್ನು ನಿಲ್ಲಿಸುತ್ತದೆ, ಮತ್ತು ಬಳಕೆದಾರರು ಇನ್ಪುಟ್ ಕೊಟ್ಟಾಗ ಮುಂದುವರೆಯುತ್ತದೆ.

©NIMI NOT TO BE REPUBLISHED

ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮತ್ತು ಜಾವಾ ಭಾಷೆ (Object Oriented Programming and JAVA Language)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- JAVA ಎಂದರೇನು
- OOPs ನ ಪರಿಕಲ್ಪನೆಗಳು

ಜಾವಾ ಟ್ಯುಟೋರಿಯಲ್ (JavaTutorial)

ಜಾವಾ ಎಂದರೇನು ?(What is Java)

ನಮ್ಮ ಕೋರ್ ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಟ್ಯುಟೋರಿಯಲ್‌ನ್ನು ವಿಧ್ಯಾರ್ಥಿಗಳಿಗಾಗಿ ಮತ್ತು ಕೆಲಸ ಮಾಡುವ ವೃತ್ತಿಪರರಿಗಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ. ಜಾವಾ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ -ಓರಿಯಂಟೆಡ್, ಕ್ಲಾಸ್ -ಆಧಾರಿತ , ಏಕಕಾಲೀನ , ಸುರಕ್ಷಿತ ಮತ್ತು ಸಾಮಾನ್ಯ ಉದ್ದೇಶದ ಕಂಪ್ಯೂಟರ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಇದು ವ್ಯಾಪಕವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ರೋಬಸ್ಟ್ ತಂತ್ರಜ್ಞಾನವನ್ನು ಹೊಂದಿದೆ.

ಜಾವಾ ಒಂದು ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ ಮತ್ತು ವೇದಿಕೆಯಾಗಿದೆ. ಜಾವಾ ಉನ್ನತ ಮಟ್ಟದ, ರೋಬಸ್ಟ್ , ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಮತ್ತು ಸುರಕ್ಷಿತವಾದ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ.

ಜಾವಾವನ್ನು 1995 ರಲ್ಲಿ ಸನ್ ಮೈಕ್ರೊಸಿಸ್ಟಮ್ಸ್ (ಈಗ ಇದು ಒರ್ಯಾಕಲ್‌ನ ಅಂಗಸಂಸ್ಥೆಯಾಗಿದೆ) ಅಭಿವೃದ್ಧಿಪಡಿಸಿದೆ. ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ಇವರನ್ನು ಜಾವಾದ ಪಿತಾಮಹ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಜಾವಾ ಹೆಸರಿಗಿಂತ ಮೊದಲು ಇದನ್ನು ಓಕ್ (oak) ಎಂದು ಕರೆಯಲಾಗುತ್ತಿತ್ತು. ಓಕ್ ಈಗಾಗಲೇ ಒಂದು ನೋದಾಯಿತ ಕಂಪನಿಯಾಗಿರುವುದರಿಂದ, ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ಮತ್ತು ಅವರ ತಂಡವು ಅದರ ಹೆಸರನ್ನು ಓಕ್‌ನಿಂದ ಜಾವಾ ಎಂದು ಬದಲಾಯಿಸಿದರು.

ವೇದಿಕೆ(Platform): ಒಂದು ಪ್ರೊಗ್ರಾಂ ರನ್ ಆಗುವ ಯಾವುದೇ ಹಾರ್ಡ್‌ವೇರ್ ಅಥವಾ ಸಾಫ್ಟ್‌ವೇರ್ ಪರಿಸರವನ್ನು ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಅದ್ದರಿಂದ ಜಾವಾ ರನ್ ಟೈಮ್ ಎನ್ವಿರಾನ್‌ಮೆಂಟ್ (run time environment (JRE) ಮತ್ತು API ಅನ್ನು ಹೊಂದಿದೆ. ಇದನ್ನು ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಜಾವಾ ಉದಾಹರಣೆ

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ ಉದಾಹರಣೆಯನ್ನು ತ್ವರಿತವಾಗಿ ನೋಡೋಣ. Hello Java ಉದಾಹರಣೆಯ ವಿವರವು ಮುಂದಿನ ಪುಟದಲ್ಲಿ ಲಭ್ಯವಿದೆ.

Simple.java

```
classSimple{
public static void main
(String args[])
{System.out.println("HelloJava");
}
}
```

ಅಪ್ಲಿಕೇಷನ್ (Application)

Sun ಪ್ರಕಾರ 3 ಬಿಲಿಯನ್ ಸಾಧನಗಳು ಜಾವಾವನ್ನು ರನ್ ಮಾಡುತ್ತವೆ. ಪ್ರಾಸ್ತುತವಾಗಿ ಜಾವಾವನ್ನು ಉಪಯೋಗಿಸುವ ಅನೇಕ ಸಾಧನಗಳಿವೆ. ಅವುಗಳಲ್ಲಿ ಕೆಲವು ಪ್ರದೇಶಗಳು ಈ ಕೆಳಗಿನಂತಿವೆ:

1 ಡೆಸ್ಕ್‌ಟಾಪ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು. ಉದಾಹರಣೆಗೆ ಅಕ್ರೊಬ್ಯಾಟ್ ರೀಡರ್, ಮೀಡಿಯಾ ಪ್ಲೇಯರ್, ಆ್ಯಂಟಿವೈರಸ್ ಇತ್ಯಾದಿಗಳು.

2 ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಉದಾಹರಣೆಗೆ irctc.co.in, javapoint.com ಇತ್ಯಾದಿಗಳು

3 ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಉದಾಹರಣೆಗೆ ಬ್ಯಾಂಕಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು

4 ಮೊಬೈಲ್

5 ಎಂಬೆಡೆಡ್ ಸಿಸ್ಟಮ್‌ಗಳು

6 ಸ್ಮಾರ್ಟ್ ಕಾರ್ಡ್

7 ರೊಬೊಟಿಕ್ಸ್

8 ಆಟಗಳು ಇತ್ಯಾದಿಗಳು

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾ ಅಪ್ಲಿಕೇಷನ್‌ನ ವಿಧಗಳು(Types of Java Applications)

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ್ನು ಬಳಸಿಕೊಂಡು ಮುಖ್ಯವಾಗಿ 4 ರೀತಿಯ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಬಹುದು.

1 Standalone Application

Standalone ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಡೆಸ್ಕ್‌ಟಾಪ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಅಥವಾ ವಿಂಡೊ ಆಧಾರಿತ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಇವು ನಾವು ಪ್ರತಿಯೊಂದು ಮಶೀನ್‌ನಲ್ಲಿಯೂ ಸ್ಥಾಪಿಸಬೇಕಾದಂತಹ ಅವಶ್ಯಕತೆ ಇರುವ ಸಾಂಪ್ರದಾಯಿಕ ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳಾಗಿವೆ. Standalone ಅಪ್ಲಿಕೇಷನ್‌ಗೆ ಉದಾಹರಣೆಗೆಗಳು ಮೀಡಿಯಾ ಪ್ಲೇಯರ್, ಆ್ಯಂಟಿವೈರಸ್ ಇತ್ಯಾದಿಗಳು. AWT ಮತ್ತು Swing ಇವುಗಳನ್ನು ಜಾವಾದಲ್ಲಿ Standalone ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

2 Web Application

ಸರ್ವರ್ ಭಾಗದಲ್ಲಿ ರನ್ ಆಗುವ ಮತ್ತು ಡೈನಾಮಿಕ್ ಪುಟವನ್ನು ರಚಿಸುವ ಒಂದು ಅಪ್ಲಿಕೇಷನ್ ಅನ್ನು ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಪ್ರಸ್ತುತವಾಗಿ Servlet, JSP, Struts,Spring,Hibernate, JSF ಇತ್ಯಾದಿ ತಂತ್ರಜ್ಞಾನಗಳನ್ನು ಜಾವಾದಲ್ಲಿ ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

3 Enterprise Application

ವಿತರಿಸುವ ಗುಣವನ್ನು ಹೊಂದಿರುವ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು ಎಂದು ಕರೆಯುತ್ತಾರೆ.ಉದಾಹರಣೆಗೆ ಬ್ಯಾಂಕಿಂಗ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳು. ಇವು ಉನ್ನತ ಮಟ್ಟದ ಸುರಕ್ಷತೆ, ಲೋಡ ಬ್ಯಾಲೆನ್ಸಿಂಗ್, ಮತ್ತು ಕ್ಲಸ್ಟರಿಂಗ್‌ನಂತಹ ಅನುಕೂಲತೆಗಳನ್ನು ಹೊಂದಿದೆ. ಜಾವಾದಲ್ಲಿ EJB ಯನ್ನು ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

4 Mobile Application

ಮೊಬೈಲ್ ಸಾಧನಗಳಿಗಾಗಿ ರಚಿಸಲಾದ ಅಪ್ಲಿಕೇಷನ್‌ನ್ನು ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಪ್ರಸ್ತುತವಾಗಿ ಆ್ಯಂಡ್ರಾಯ್ಡ್ ಮತ್ತು ಜಾವಾ ME ಇವುಗಳನ್ನು ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳು/ಆವೃತ್ತಿಗಳು(Java Platforms/Editions)

ಜಾವಾದಲ್ಲಿ 4 ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗಳು/ಆವೃತ್ತಿಗಳು ಇವೆ :

1 JavaSE (Java Standard Edition)

ಇದು ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಆಗಿದೆ. ಇದು ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ API ಗಳಾದ java.lang, java.io, java.net, java.util, java.sql, java.math ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಇದು ಕೋರ್ ವಿಷಯಗಳಾದ OOPs, String, Regex, Exception, Inner classes, Multi threading,I/OStream, Networking, AWT, Swing, Reflection, Collection, ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಿದೆ.

2 JavaEE (Java Enterprise Edition)

ಇದು ಎಂಟರ್‌ಪ್ರೈಸ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಆಗಿದೆ. ಇದನ್ನು ಮುಖ್ಯವಾಗಿ ವೆಬ್ ಮತ್ತು ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು ಜಾವಾ SE ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ನ ಮೇಲ್ಭಾಗದಲ್ಲಿ ನಿರ್ಮಿಸಲಾಗಿದೆ. ಇದು Servlet, JSP, Web Services, EJB, JPA, ಮುಂತಾದ ವಿಷಯಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

3 JavaME (Java Micro Edition)

ಇದು ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಿಗೆ ಮೀಸಲಾದ ಮೈಕ್ರೊ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಆಗಿದೆ.

4 JavaFX

ಇದನ್ನು ಇಂಟರ್‌ನೆಟ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಹಗುರವಾದ ಯೂಸರ್ ಇಂಟರ್‌ಫೇಸ್ API ಯನ್ನು ಬಳಸುತ್ತದೆ.

ಪೂರ್ವಾಪೇಕ್ಷಿತ (Prerequisite)

ಜಾವಾಅನ್ನು ಕಲಿಯಲು ನೀವು C/C++ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳ ಸಾಮಾನ್ಯ ಜ್ಞಾನವನ್ನು ಹೊಂದಿರಬೇಕು.

ಪ್ರೇಕ್ಷಕರು (Audience)

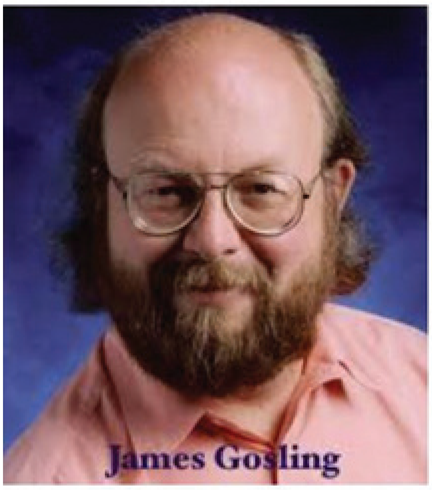
ನಮ್ಮ ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ ಟ್ಯುಟೋರಿಯಲ್‌ನ್ನು ಇದನ್ನು ಕಲಿಯಲು ಪ್ರಾರಂಭಿಸುವವರಿಗೆ ಮತ್ತು ವೃತ್ತಿಪರರಿಗೆ ಸಹಾಯ ಮಾಡಲು ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

ಜಾವಾದ ಇತಿಹಾಸ (History of Java)

ಜಾವಾದ ಇತಿಹಾಸವು ತುಂಬಾ ಆಸಕ್ತಿದಾಯಕವಾಗಿದೆ. ಜಾವಾವನ್ನು ಮೂಲತಃ ಸಂವಾದಾತ್ಮಕ ದೂರದರ್ಶನಕ್ಕಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿತ್ತು. ಆದರೆ ಆ ಸಮಯದಲ್ಲಿ ಇದು ಡಿಜಿಟಲ್ ಕೇಬಲ್ ಟೆಲಿವಿಷನ್ ಉದ್ಯಮಕ್ಕೆ ಅತ್ಯಂತ ಮುಂದುವರಿದ ತಂತ್ರಜ್ಞಾನವಾಗಿತ್ತು. ಜಾವಾದ ಇತಿಹಾಸವು Green Team ದಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಜಾವಾ ತಂಡದ ಸದಸ್ಯರು (Green Team ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ) ಈ ಪ್ರೊಜೆಕ್ಟ್‌ನ್ನು ಡಿಜಿಟಲ್ ಸಾಧನಗಳಾದ ಸೆಟ್ ಟಾಪ್ ಬಾಕ್ಸ್, ಟೆಲಿವಿಷನ್ ಮುಂತಾದವುಗಳಿಗಾಗಿ ಒಂದು ಭಾಷೆಯನ್ನು ಅಬಿವೃದ್ಧಿಪಡಿಸುವ ಸಲುವಾಗಿ ಪ್ರಾರಂಭಿಸಿದರು. ಅದಾಗ್ಯೂ ಇದು ಇಂಟರ್‌ನೆಟ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ಗಾಗಿ ಅತ್ಯಂತ ಸೂಕ್ತವಾಗಿತ್ತು. ನಂತರ ಜಾವಾ ತಂತ್ರಜ್ಞಾನವು ನೆಟ್‌ಸ್ಕೇಪ್‌ನಿಂದ ಸಂಯೋಜಿಸಲ್ಪಟ್ಟಿತು.

NOT TO BE REPUBLISHED © NIMI

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ್ನು ರಚಿಸುವ ತತ್ವಗಳು ಯಾವವೆಂದರೆ "ಸರಳ, ರೋಬಸ್ಟ್, ಪೋರ್ಟೇಬಲ್, ಪ್ಲಾಟ್‌ಫಾರ್ಮ್-ಸ್ವತಂತ್ರ್ಯ, ಸುರಕ್ಷಿತ, ಹೆಚ್ಚು ಕಾರ್ಯಕ್ಷಮತೆ, ಮಲ್ಟಿಥ್ರೆಡೆಡ್, ಆರ್ಕಿಟೆಕ್ಚರ್ ನ್ಯೂಟ್ರಲ್, ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್, ಇಂಟರ್‌ಪ್ರಿಟೆಡ್, ಮತ್ತು ಡೈನಾಮಿಕ್" ಜಾವಾವನ್ನು ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ 1995 ರಲ್ಲಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಿದರು. ಇವರನ್ನು ಜಾವಾದ ಪಿತಾಮಹ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ಮತ್ತು ಅವರ ತಂಡದವರು ಈ ಪ್ರೊಜೆಕ್ಟನ್ನು 90 ರ ದಶಕದ ಆರಂಭದಲ್ಲಿ ಪ್ರಾರಂಭಿಸಿದರು.



ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ - ಜಾವಾದ ಸಂಸ್ಥಾಪಕರು (James Gosling-founder of java)

1 James Gosling, Mike Sheridan,ಮತ್ತು Patrick Naughton ಇವರು ಜಾವಾ ಭಾಷೆಯ ಪ್ರೊಜೆಕ್ಟ್‌ನ್ನು 1991 ರಲ್ಲಿ ಪ್ರಾರಂಭಿಸಿದರು. sun ಇಂಜಿನಿಯರ್‌ಗಳ ಸಣ್ಣ ತಂಡವನ್ನು Green Team ಎಂದು ಕರೆದರು.

2 ಆರಂಭದಲ್ಲಿ ಇದನ್ನು ಇಲೆಕ್ಟ್ರಾನಿಕ್ ಉಪಕರಣಗಳಾದ ಸೆಟ್ ಟಾಪ್ ಬಾಕ್ಸ್‌ಗಳಂತಹ ಸಣ್ಣ, ಎಂಬೆಡೆಡ್ ಸಿಸ್ಟಮ್‌ಗಳಿಗಾಗಿ ವಿನ್ಯಾಸಗೊಳಿಸಲಾಗಿದೆ.

3 ಮೊದಲನೇಯದಾಗಿ ಇದು ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್‌ರಿಂದ "Greentalk" ಎಂದು ಕರೆಯಲ್ಪಟ್ಟಿತು ಮತ್ತು ಇದರ ಫೈಲ್ ವಿಸ್ತರಣೆ ಹೆಸರು .gt ಆಗಿತ್ತು.

4 ಅದರ ನಂತರ ಇದನ್ನು Oak ಎಂದು ಕರೆಯಲಾಯಿತು ಮತ್ತು Green poject ನ ಒಂದು ಭಾಗವಾಗಿ ಅಭಿವೃದ್ಧಿಪಡಿಸಲಾಯಿತು.

ಜಾವಾವನ್ನು "Oak" ಎಂದು ಏಕೆ ಹೆಸರಿಸಲಾಯಿತು? Oak ದಿಂದ Java ವರೆಗಿನ ಜಾವಾ ಇತಿಹಾಸ(Why Java was named as "Oak"?Java History from Oak to Java)

5 Oak ಎಕೆ? ಓಕ್ ಶಕ್ತಿಯ ಸಂಕೇತವಾಗಿದೆ. ಮತ್ತು ಯು.ಎಸ್.ಎ, ಫ್ರಾನ್ಸ್, ಜರ್ಮನಿ, ರೊಮೇನಿಯಾ ಮುಂತಾದ ದೇಶಗಳ ರಾಷ್ಟ್ರೀಯ ಮರ ಆಗಿ ಆಯ್ಕೆಯಾಗಿದೆ.

6 1995 ರಲ್ಲಿ Oak ನ್ನು Java ಎಂದು ಮರುನಾಮಕರಣ ಮಾಡಲಾಯಿತು. ಏಕೆಂದರೆ ಈಗಾಗಲೇ ಅದು Oak Technology ಅವರ ಟ್ರೇಡ್‌ಮಾರ್ಕ್ ಆಗಿತ್ತು.

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನ್ನು "Java" ಎಂದು ಏಕೆ ಹೆಸರಿಸಲಾಗಿದೆ?

7 ಜಾವಾ ಭಾಷೆಗೆ ಅವರು ಜಾವಾ ಎಂಬ ಹೆಸರನ್ನು ಏಕೆ ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡರು? ತಂಡವು ಹೊಸ ಹೆಸರನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಒಟ್ಟುಗೂಡಿತು. ಸೂಚಿಸಲಾದ ಪದಗಳು "dynamic", "revolutionary", "Silk", "jolt", "DNA" ಆಗಿದ್ದವು. ಅವರು ತಂತ್ರಜ್ಞಾನದ ಸಾರವನ್ನು ಪ್ರತಿಬಿಂಬಿಸುವ ಯಾವುದಾದರೂ ಪದವನ್ನು ಬಯಸಿದ್ದರು ಉದಾ : revolutionary, dynamic, lively, cool, unique, ಮತ್ತು easy to spell, ಮತ್ತು funtosay.

ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ಪ್ರಕಾರ "ಸಿಲ್ಕ್ ಜೊತೆಗೆ ಜಾವಾ ಅತ್ಯಂತ ಅಗ್ರಸ್ಥಾನದ ಆಯ್ಕೆಯಾಗಿದೆ" ಜಾವಾ ಅತ್ಯಂತ ವಿಶಿಷ್ಟವಾಗಿರುವುದರಿಂದ, ತಂಡದ ಹೆಚ್ಚಿನ ಸದಸ್ಯರು Java ಹೆಸರಿಗೆ ಆದ್ಯತೆ ನೀಡಿದರು.

8 ಜಾವಾ ಇಂಡೊನೇಷಿಯಾದ ಐಲ್ಯಾಂಡ ಆಗಿದೆ. ಇಲ್ಲಿ ಮೊದಲು ಕಾಫೀಯನ್ನು ಉತ್ಪಾದಿಸಲಾಯಿತು (ಜಾವಾ ಕಾಫೀ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ) ಇದು ಒಂದು ವಿಧದ ಎಸ್ಪ್ರೆಸ್ಸೊ ಬೀಜವಾಗಿದೆ. ಜಾವಾ ಹೆಸರನ್ನು ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ತಮ್ಮ ಆಫೀಸಿನ ಹತ್ತಿರದ ಕಾಫೀಯನ್ನು ಸೇವಿಸುವಾಗ ಆಯ್ಕ ಮಾಡಿದರು.

9 ಜಾವಾ ಕೇವಲ ಒಂದು ಹೆಸರು ಸಂಕ್ಷೇಪಣವಲ್ಲ ಎಂಬುದನ್ನು ಗಮನಿಸಿ.

10 ಆರಂಭದಲ್ಲಿ ಜೇಮ್ಸ್ ಗೋಸ್ಲಿಂಗ್ ಸನ್ ಮೈಕ್ರೊಸಿಸ್ಟಮ್ಸ್‌ನಲ್ಲಿ (ಈಗ ಇದು ಒರ್ಯಾಕಲ್ ಕಾರ್ಪೋರೇಷನ್‌ನ ಅಂಗಸಂಸ್ಥೆಯಾಗಿದೆ) ಅಭಿವೃದ್ಧಿಪಡಿಸಿದರು ಮತ್ತು 1995 ರಲ್ಲಿ ಇದನ್ನು ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು.

11 1995 ರಲ್ಲಿ, Time magazine ಪತ್ರಿಕೆ ಜಾವಾವನ್ನು 1995 ರ 10 ರಲ್ಲಿ ಅತ್ಯಂತ ಉತ್ತಮವಾದ ಒಂದು ಉತ್ಪನ್ನವೆಂದು ಕರೆಯಿತು.

12 JDK 1.0 ಅನ್ನು ಜನೇವರಿ 23, 1996 ರಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡಲಾಯಿತು. ಜಾವಾದ ಮೊದಲನೇಯ ಬಿಡುಗಡೆಯ ನಂತರ ಭಾಷೆಗೆ ಹಲವಾರು ಹೆಚ್ಚುವರಿ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸೇರಿಸಲಾಯಿತು. ಈಗ ಜಾವಾವನ್ನು ವಿಂಡೊಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ, ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ, ಎಂಟರ್‌ಪ್ರೈಸ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ, ಮೊಬೈಲ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳಲ್ಲಿ, ಕಾರ್ಡ್‌ಗಳಲ್ಲಿ ಮುಂತಾದವುಗಳಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತಿದೆ.

© NIMI NOT TO BE REPUBLISHED

ಪ್ರತಿಯೊಂದು ಹೊಸ ಆವೃತ್ತಿಯು ಜಾವಾದಲ್ಲಿ ಹೊಸ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ.

ಜಾವಾ ಆವೃತ್ತಿಯ ಇತಿಹಾಸ (Java Version History)

ಇಲ್ಲಿಯವರೆಗೆ ಅನೇಕ ಜಾವಾ ಆವೃತ್ತಿಗಳು ಬಿಡುಗಡೆಯಾಗಿವೆ. ಜಾವಾದ ಪ್ರಸ್ತುತ ಸ್ಥಿರ ಬಿಡುಗಡೆ JavaSE18 ಆಗಿದೆ.

1 JDK Alpha and Beta (1995)

2 JDK1.0 (23rdJan1996)

3 JDK1.1(19thFeb1997)

4 J2SE1.2 (8thDec1998)

5 J2SE1.3 (8thMay2000)

6 J2SE1.4(6thFeb2002)

7 J2SE5.0(30thSep2004)

8 JavaSE6(11thDec2006)

9 JavaSE7(28thJuly2011)

10 JavaSE8 (18thMar2014)

11 JavaSE 9 (21stSep2017)

12 JavaSE 10 (20thMar2018)

13 JavaSE11 (September2018)

14 JavaSE12(March2019)

15 JavaSE13(September2019)

16 JavaSE14(Mar2020)

17 JavaSE15(September2020)

18 JavaSE16(Mar2021)

19 JavaSE17(September2021)

20 JavaSE 18 (March2022 ರವರೆಗೆ ಬಿಡುಗಡೆಯಾಗಲಿದೆ)

Java SE 8 ರ ಬಿಡುಗಡೆಯವರೆಗೆ ಒರ್ಯಾಕಲ್ ಕಾರ್ಪೋರೇಷನ್ ಒಂದು ಮಾದರಿಯನ್ನು ಅನುಸರಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ ಪ್ರತಿ ಸಮ ಆವೃತ್ತಿಯು ಮಾರ್ಚ್ ತಿಂಗಳಲ್ಲಿ ಬಿಡುಗಡೆಯಾಗುತ್ತದೆ ಮತ್ತು ಪ್ರತಿ ಬೆಸ ಆವೃತ್ತಿಯು ಸೆಪ್ಟೆಂಬರ್ ತಿಂಗಳಲ್ಲಿ ಬಿಡುಗಡೆಯಾಗುತ್ತದೆ.

ಜಾವಾದ ವೈಶಿಷ್ಟ್ಯಗಳು (Features of Java)

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯ ಪ್ರಮುಖ ಉದ್ದೇಶ ಅದನ್ನು ಪೋರ್ಟೇಬಲ್, ಸರಳ ಮತ್ತು ಸುರಕ್ಷಿತ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯನ್ನಾಗಿ ಮಾಡುವುದು. ಇದಲ್ಲದೇ ಇದರ ಕೆಲವು ಅತ್ಯುನ್ನತ ವೈಶಿಷ್ಟ್ಯಗಳು ಈ ಭಾಷೆಯನ್ನು ಜನಪ್ರಿಯವಾಗಿಸಲು ಮಹತ್ವವಾದ ಪಾತ್ರವನ್ನು ನಿರ್ವಹಿಸುತ್ತವೆ. ಜಾವಾದ ವೈಶಿಷ್ಟ್ಯಗಳಿಗೆ Java buzz-words ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ.

ಜಾವಾ ಭಾಷೆಯ ಅತ್ಯಂತ ಮುಖ್ಯವಾದ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಕೆಳಗೆ ಪಟ್ಟಿ ಮಾಡಲಾಗಿದೆ.

1 Simple (ಸರಳ)

2 Object-Oriented (ವಸ್ತು ಆಧಾರಿತ)

3 Portable (ಪೋರ್ಟೇಬಲ್)

4 Platform independent (ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ - ಸ್ವತಂತ್ರ)

5 Secured (ಸುರಕ್ಷಿತ)

6 Robust (ದೃಢವಾದ)

7 Architecture neutral (ಆರ್ಕಿಟೆಕ್ಚರ್ ತಟಸ್ಥ)

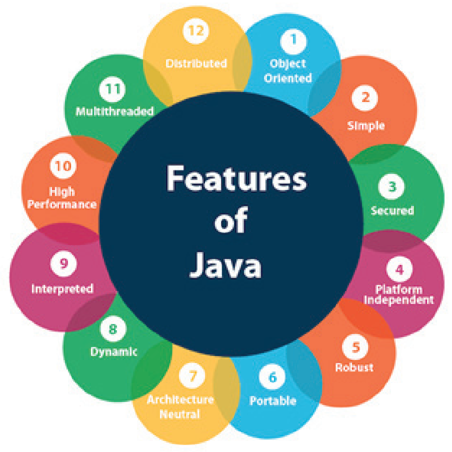
8 Interpreted (ಅರ್ಥ ವಿವರಿಸುವ)

9 High Performance (ಹೆಚ್ಚಿನ ಕಾರ್ಯಕ್ಷಮತೆ)

10 Multithreaded (ಮಲ್ಟಿಥ್ರೆಡೆಡ್)

11 Distributed (ವಿತರಿಸಲಾದ)

12 Dynamic (ಡೈನಾಮಿಕ್)



ಸರಳ(Simple)

ಜಾವಾ ಕಲಿಯಲು ತುಂಬಾ ಸುಲಭವಾಗಿದೆ ಮತ್ತು ಅದರ ಸಿಂಟ್ಯಾಕ್ಸ್ ಸರಳವಾಗಿದೆ, ಸ್ವಚ್ಛವಾಗಿದೆ ಮತ್ತು ಅರ್ಥಮಾಡಿಕೊಳ್ಳಲು ಸುಲಭವಾಗಿದೆ. Sun Microsystem ಪ್ರಕಾರ ಜಾವಾ ಭಾಷೆ ಒಂದು ಸರಳವಾದ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಏಕೆಂದರೆ :

- ಜಾವಾ ಸಿಂಟ್ಯಾಕ್ಸ್ C++ ನ್ನು ಆಧರಿಸಿದೆ. (ಅದ್ದರಿಂದ C++ ನಂತರ ಇದನ್ನು ಕಲಿಯುವುದು ಪ್ರೊಗ್ರಾಮರ್‌ಗಳಿಗೆ ತುಂಬಾ ಸುಲಭವಾಗಿದೆ)

© NIMI NOT TO BE REPUBLISHED

- ಜಾವಾ ಅನೇಕ ಕ್ಲಿಷ್ಟಕರವಾದ ಮತ್ತು ಅಪರೂಪವಾಗಿ ಬಳಸುವ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ತೆಗೆದುಹಾಕಿದೆ. ಉದಾಹರಣೆಗೆ explicit pointers, operator overloading ಇತ್ಯಾದಿಗಳು.
- ಉಲ್ಲೇಖಿಸದ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ. ಏಕೆಂದರೆ ಜಾವಾದಲ್ಲಿ Automatic Garbage Collection ಇದೆ.

ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೇಡ್ (Object-oriented)

ಜಾವಾ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೇಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಜಾವಾದಲ್ಲಿ ಎಲ್ಲವೂ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ. ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೇಡ್ ಎಂದರೆ, ನಾವು ನಮ್ಮ ಸಾಫ್ಟ್‌ವೇರನ್ನು ವಿಭಿನ್ನ ರೀತಿಯ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಸಂಯೋಜನೆಯಾಗಿ ಆಯೋಜಿಸುತ್ತೇವೆ. ಈ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಡಾಟಾ ಮತ್ತು ಬಿಹೇವಿಯರ್‌ಗಳನ್ನು ಒಂದುಗೂಡಿಸುತ್ತವೆ.

ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೇಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ (OOPs) ಒಂದು ವಿಧಾನವಾಗಿದೆ. ಇದು ಕೆಲವು ನಿಯಮಗಳನ್ನು ಒದಗಿಸುವ ಮೂಲಕ ಸಾಫ್ಟ್‌ವೇರನ ಅಭಿವೃದ್ಧಿ ಮತ್ತು ನಿರ್ವಹಣೆಗಳನ್ನು ಸರಳಗೊಳಿಸುತ್ತದೆ.

OOps ನ ಮೂಲ ಪರಿಕಲ್ಪನೆಗಳು: (Basic concepts of OOPs are:)

1 ಆಬ್ಜೆಕ್ಟ್ (Object)

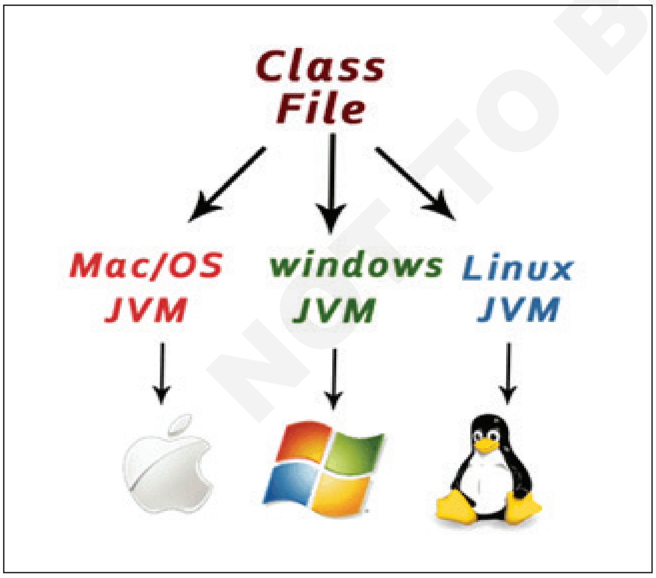
2 ಕ್ಲಾಸ್ (Class)

3 ಇನ್‌ಹೆರಿಟನ್ಸ್ (Inheritance)

4 ಪಾಲಿಮಾರ್ಫಿಸಮ್ (Polymorphism)

5 ಅಬ್‌ಸ್ಟ್ರಾಕ್ಷನ್ (Abstraction)

6 ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್ (Encapsulation)



ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರ(Platform Independent)

ಜಾವಾ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರವಾಗಿದೆ ಏಕೆಂದರೆ ಇದು ಇತರ ಭಾಷೆಗಳಿಗಿಂತ ವಿಭಿನ್ನವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ C, C++ ಇತ್ಯಾದಿಗಳನ್ನು ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ನಿರ್ದಿಷ್ಟ ಯಂತ್ರಗಳಲ್ಲಿ ಕಂಪೈಲ್ ಮಾಡಲಾಗುತ್ತದೆ. ಆದರೆ ಜಾವಾ ಒಮ್ಮೆ ಬರೆದು, ಎಲ್ಲಿಯಾದರೂ ರನ್ ಮಾಡಬಹುದಾದ ಭಾಷೆಯಾಗಿದೆ.

ಇದರಲ್ಲಿ ಎರಡು ರೀತಿಯ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಿವೆ - ಸಾಫ್ಟ್‌ವೇರ್-ಆಧಾರಿತ ಮತ್ತು ಹಾರ್ಡ್‌ವೇರ್ -ಆಧಾರಿತ. ಜಾವಾ ಸಾಫ್ಟ್‌ವೇರ್-ಆಧಾರಿತ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ.

ಜಾವಾ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಇತರ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಿಗಿಂತ ಭಿನ್ನವಾಗಿದೆ. ಇದರ ಅರ್ಥ ಇದು ಸಾಫ್ಟ್‌ವೇರ್ ಆಧಾರಿತ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಆಗಿದ್ದು ಇತರ ಹಾರ್ಡ್‌ವೇರ್ ಆಧಾರಿತ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಿಗಿಂತ ಮೊದಲು ರನ್ ಆಗುತ್ತದೆ. ಇದು ಎರಡು ಘಟಕಗಳನ್ನು ಹೊಂದಿದೆ :

1 Runtime Environment

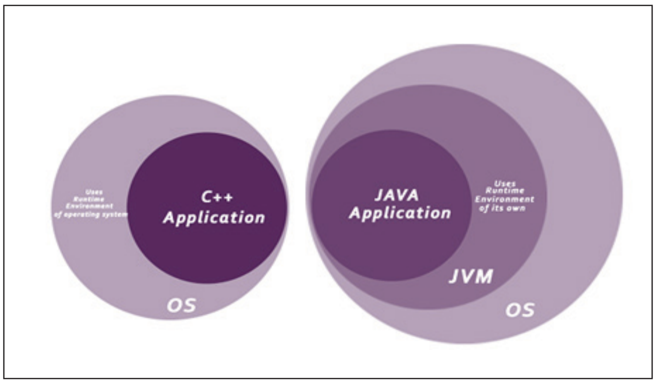
2 API (Application Programming Interface)

ಜಾವಾ ಕೋಡ್‌ನ್ನು ಅನೇಕ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ಕಾರ್ಯಗತಗೊಳಿಸಬಹುದು. ಉದಾಹರಣೆಗೆ Windows, Linux, Sun Solaris, Mac/OS ಇತ್ಯಾದಿಗಳು. ಜಾವಾ ಕೋಡ್‌ನ್ನು ಕಂಪೈಲರ್‌ನಿಂದ ಕಂಪೈಲ್ ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ಬೈಟ್‌ಕೋಡ್‌ಗೆ ಪರಿವರ್ತಿಸಲಾಗುತ್ತದೆ. ಈ ಬೈಟ್ ಕೋಡ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರ ಕೋಡ್ ಆಗಿರುತ್ತದೆ. ಏಕೆಂದರೆ ಇದನ್ನು ಹಲವಾರು ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಲ್ಲಿ ರನ್ ಮಾಡಬಹುದು. ಎಂದರೆ, Write Once and Run Anywhere (WORA)

ಸುರಕ್ಷಿತ (Secured)

ಜಾವಾ ತನ್ನ ಭದ್ರತೆಗೆ ಹೆಸರುವಾಸಿಯಾಗಿದೆ. ಜಾವಾವನ್ನು ಉಪಯೋಗಿಸಿ ನಾವು ವೈರಸ್ ಮುಕ್ತ ಸಿಸ್ಟಮ್‌ಗಳನ್ನು ಅಬಿವೃದ್ಧಿಪಡಿಸಬಹುದು. ಜಾವಾ ಸುರಕ್ಷಿತವಾಗಿದೆ ಏಕೆಂದರೆ :

- ಯಾವುದೇ ಸ್ಪಷ್ಟ ಪಾಯಿಂಟರ್ ಇಲ್ಲ.
- ಜಾವಾ ಪ್ರೊಗ್ರಾಂಗಳು ವರ್ಚುವಲ್ ಮಶೀನ್ ಸ್ಯಾಂಡ್‌ಬಾಕ್ಸ್‌ನಲ್ಲಿ (virtual machine sandbox) ರನ್ ಆಗುತ್ತವೆ.
- Classloader: ಜಾವಾದಲ್ಲಿ ಕ್ಲಾಸ್‌ಲೋಡರ್ ಎಂದರೆ Java Run-



time Environment (JRE) ದ ಒಂದು ಭಾಗವಾಗಿದೆ. ಇದನ್ನು ಉಪಯೋಗಿಸಿ ಜಾವಾ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಕ್ರಿಯಾತ್ಮಕವಾಗಿ ಜಾವಾ

© NIMI NOT TO BE REPUBLISHED

ವರ್ಚುವಲ್ ಮಶೀನ್‌ನಲ್ಲಿ ಲೋಡ್ ಮಾಡಲಾಗುತ್ತದೆ. ಇದು ಸ್ಥಳಿಯ ಫೈಲ್ ಸಿಸ್ಟಮ್‌ಗಳ ಕ್ಲಾಸ್‌ಗಳಿಗಿರುವ ಪ್ಯಾಕೆಜ್‌ಗಳನ್ನು, ನೆಟ್‌ವರ್ಕ್ ಸೋರ್ಸ್‌ಗಳಿಂದ ಇಂಪೋರ್ಟ್ ಮಾಡಲಾದ ಪ್ಯಾಕೆಜ್‌ಗಳಿಂದ ಬೇರ್ಪಡಿಸುವ ಮೂಲಕ ಸುರಕ್ಷತೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ.

- Bytecode Verifier: ಇದು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಿಗೆ ಪ್ರವೇಶ ಹಕ್ಕುಗಳನ್ನುಉಲ್ಲಂಘಿಸಬಹುದಾದ ಕಾನೂನು ಬಾಹಿರ ಕೋಡ್‌ಗಳಿಗಾಗಿ, ಕೋಡ್ ತುಂಡುಗಳನ್ನು (fragments) ಪರಿಶೀಲಿಸುತ್ತದೆ.
- Security Manager: ಇದು ಒಂದು ಕ್ಲಾಸ್ ಯಾವ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು ಎಂಬುದನ್ನು ನಿರ್ಧರಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ ಸ್ಥಳಿಯ ಡಿಸ್ಕ್‌ನಲ್ಲಿ ಬರೆಯುವುದು ಅಥವಾ ಓದುವುದು.

ಜಾವಾ ಭಾಷೆಯು ಡೀಫಾಲ್ಟ್ ಆಗಿ ಈ ಸುರಕ್ಷತೆಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ. ಕೆಲವು ಸುರಕ್ಷತೆಗಳನ್ನು ಅಪ್ಲಿಕೇಷನ್ ಡೆವಲಪರ್ SSL, JAAS, Cryptography, ಇತ್ಯಾದಿಗಳ ಮೂಲಕ ಒದಗಿಸುತ್ತದೆ.

ದೃಢವಾದ (Robust)

ರೋಬಸ್ಟ್‌ನ ಇಂಗ್ಲಿಷ್ ಅರ್ಥ ಪ್ರಬಲ ಅಥವಾ ದೃಢ ಎಂದಾಗಿದೆ. ಜಾವಾ ರೋಬಸ್ಟ್ ಆಗಿದೆ ಏಕೆಂದರೆ :

- ಇದು ಬಲವಾದ ಮೆಮೊರಿ ನಿರ್ವಹಣೆಯನ್ನು ಬಳಸುತ್ತದೆ.
- ಸುರಕ್ಷತಾ ಸಮಸ್ಯೆಗಳನ್ನು ತಪ್ಪಿಸುವ ಪಾಯಿಂಟರ್‌ಗಳ ಕೊರತೆಯಿದೆ.
- ಜಾವಾ ಸ್ವಯಂಆಗಿ ಗಾರ್ಬೇಜ್ ಕಲೆಕ್ಷನ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್‌ನಲ್ಲಿ ರನ್ ಆಗುತ್ತದೆ ಮತ್ತು ಜಾವಾ ಅಪ್ಲಿಕೇಷನ್‌ನಿಂದ ಇನ್ನು ಮುಂದೆ ಬಳಸದೇ ಇರುವ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.
- ಜಾವಾದಲ್ಲಿ ಎಕ್ಸೆಪ್ಷನ್ ಹಾಂಡ್ಲಿಂಗ್ ಮತ್ತು ಟೈಪ್ ಚೆಕಿಂಗ್‌ನಂತಹ ಯಾಂತ್ರಿಕ ವ್ಯವಸ್ಥೆಗಳಿವೆ. ಈ ಎಲ್ಲಾ ಅಂಶಗಳು ಜಾವಾವನ್ನು ರೋಬಸ್ಟ್ ಆಗಿಸುತ್ತವೆ.

ಆರ್ಕಿಟೆಕ್ಚರ್ ತಟಸ್ಥ (Architecture-neutral)

ಜಾವಾ ಆರ್ಕಿಟೆಕ್ಚರ್ ತಟಸ್ಥವಾಗಿದೆ. ಏಕೆಂದರೆ ಇದರಲ್ಲಿ ಅನುಷ್ಠಾನ ಅವಲಂಬಿತ ವೈಶಿಷ್ಟ್ಯಗಳಿಲ್ಲ. ಉದಾಹರಣೆಗೆ ಪ್ರಿಮಿಟಿವ್ ಟೈಪ್‌ಗಳ ಗಾತ್ರವನ್ನು ನಿಗದಿಪಡಿಸಲಾಗಿದೆ.

C ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ 32 ಬಿಟ್ ಆರ್ಕಿಟೆಕ್ಚರ್‌ಗಾಗಿ int ಡಾಟಾಟೈಪ್ 2 ಬೈಟ್‌ಗಳಷ್ಟು ಮೆಮೊರಿಯನ್ನು ಆಕ್ರಮಿಸುತ್ತದೆ. ಮತ್ತು 64 ಬಿಟ್ ಆರ್ಕಿಟೆಕ್ಚರ್‌ಗಾಗಿ 4 ಬೈಟ್‌ಗಳಷ್ಟು ಮೆಮೊರಿಯನ್ನು ಆಕ್ರಮಿಸುತ್ತದೆ. ಅದಾಗ್ಯೂ, ಜಾವಾದಲ್ಲಿ 32 ಬಿಟ್ ಮತ್ತು 64 ಬಿಟ್ ಆರ್ಕಿಟೆಕ್ಚರ್‌ಗಳಿಗಾಗಿ ಇದು 4 ಬೈಟ್ ಮೆಮೊರಿಯನ್ನು ಆಕ್ರಮಿಸುತ್ತದೆ.

ಪೋರ್ಟೇಬಲ್ (Portable)

ಜಾವಾ ಪೋರ್ಟೇಬಲ್ ಆಗಿದೆ. ಏಕೆಂದರೆ ಇದು ಜಾವಾ ಬೈಟ್‌ಕೋಡನ್ನು ಯಾವುದೇ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ಗೆ ತೆಗೆದುಕೊಂಡು ಹೋಗಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ. ಇದಕ್ಕೆ ಯಾವುದೇ ಅನುಷ್ಠಾನದ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

ಹೆಚ್ಚಿನ ಕಾರ್ಯಕ್ಷಮತೆ (High - performance)

ಜಾವಾ ಇತರ ಸಾಂಪ್ರದಾಯಿಕ ಇಂಟರ್‌ಪ್ರಿಟೇಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳಿಗಿಂತ ಹೆಚ್ಚು ವೇಗವಾಗಿದೆ. ಏಕೆಂದರೆ ಜಾವಾ ಬೈಟ್‌ಕೋಡ್ ಸ್ಥಳೀಯ ಕೋಡ್‌ಗಳಿಗೆ ಹತ್ತಿರವಾಗಿದೆ. ಆದರೂ ಇದು ಕಂಪೈಲ್ ಭಾಷೆಗಳಿಗಿಂತ ಇನ್ನೂ ಸ್ವಲ್ಪ ನಿಧಾನವಾಗಿದೆ. (ಉದಾಹರಣೆಗೆ C++) ಜಾವಾ ಒಂದು ಇಂಟರ್‌ಪ್ರಿಟೆಡ್ ಭಾಷೆಯಾಗಿರುವುದರಿಂದ ಅದು ಕಂಪೈಲ್ ಮಾಡುವ ಭಾಷೆಗಳಿಗಿಂತ ನಿಧಾನವಾಗಿದೆ. ಉದಾಹರಣೆಗೆ C, C++ ಇತ್ಯಾದಿಗಳು.

ವಿತರಣೆ (Distributed)

ಜಾವಾ ಡಿಸ್ಟ್ರಿಬ್ಯೂಟೆಡ್ ಆಗಿದೆ. ಏಕೆಂದರೆ ಇದು ಜಾವಾದಲ್ಲಿ ಡಿಸ್ಟ್ರಿಬ್ಯೂಟೆಡ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು ಬಳಕೆದಾರರಿಗೆ ಅನುಕೂಲ ಮಾಡಿಕೊಡುತ್ತದೆ. ಡಿಸ್ಟ್ರಿಬ್ಯೂಟೆಡ್ ಅಪ್ಲಿಕೇಷನ್‌ಗಳನ್ನು ರಚಿಸಲು RMI ಮತ್ತು EJB ಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಜಾವಾದ ಈ ವೈಶಿಷ್ಟ್ಯವು ಇಂಟರ್‌ನೆಟ್ ಉಪಯೋಗಿಸಿ ಯಾವುದೇ ಮಶೀನ್‌ನಿಂದ ಮೆಥೆಡ್‌ಗಳನ್ನು ಕಾಲ್ ಮಾಡುವ ಮೂಲಕ ಫೈಲ್‌ಗಳನ್ನು ಪ್ರವೇಶಿಸಲು ನಮಗೆ ಸಾಧ್ಯವಾಗುತ್ತದೆ.

ಮಲ್ಟಿ ಥ್ರೆಡೆಡ್ (Multi-threaded)

ಥ್ರೆಡ್ ಎಂದರೆ ಏಕಕಾಲದಲ್ಲಿ ಕಾರ್ಯಗತಗೊಳ್ಳುವ ಒಂದು ಪ್ರತ್ಯೇಕ ಪ್ರೊಗ್ರಾಂನಂತಿರುತ್ತದೆ. ಅನೇಕ ಥ್ರೆಡ್‌ಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಮೂಲಕ ಒಂದು ಸಮಯದಲ್ಲಿ ಹಲವಾರು ಕಾರ್ಯಗಳನ್ನು ನಿಭಾಯಿಸುವ ಜಾವಾ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ನಾವು ಬರೆಯಬಹುದು. ಮಲ್ಟಿ ಥ್ರೆಡಿಂಗ್‌ನ ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ, ಇದು ಪ್ರತಿ ಥ್ರೆಡ್‌ಗೆ ಮೆಮೊರಿಯನ್ನು ಆಕ್ರಮಿಸುವುದಿಲ್ಲ. ಇದು ಒಂದೇ ಮೆಮೊರಿ ಪ್ರದೇಶವನ್ನು ಹಂಚಿಕೊಳ್ಳುತ್ತದೆ. ಥ್ರೆಡ್‌ಗಳು ಮಲ್ಟಿ ಮೀಡಿಯಾ , ವೆಬ್ ಅಪ್ಲಿಕೇಷನ್ ಇತ್ಯಾದಿಗಳಿಗೆ ಬಹುಮುಖ್ಯವಾಗಿದೆ.

ಡೈನಾಮಿಕ್ (Dynamic)

ಜಾವಾ ಒಂದು ಡೈನಾಮಿಕ್ ಭಾಷೆಯಾಗಿದೆ. ಇದು ಕ್ಲಾಸ್‌ಗಳ ಡೈನಾಮಿಕ್ ಲೋಡಿಂಗ್‌ನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಇದರ ಅರ್ಥ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಬೇಡಿಕೆಯ ಮೇಲೆ ಲೋಡ್ ಮಾಡಲಾಗುತ್ತದೆ. ಇದು ತನ್ನ ಸ್ಥಳೀಯ ಭಾಷೆಗಳಲ್ಲಿನ ಫಂಕ್ಷನ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ. ಎಂದರೆ C, C++.

ಜಾವಾ ಡೈನಾಮಿಕ್ ಕಂಪೈಲೇಷನ್‌ನ್ನು ಮತ್ತು ಅಟೋಮ್ಯಾಟಿಕ್ ಮೆಮೊರಿ ಮ್ಯಾನೇಜ್‌ಮೆಂಟನ್ನು (ಗಾರ್ಬೇಜ್ ಕಲೆಕ್ಷನ್) ಬೆಂಬಲಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಬರೆಯುವ ವಿಧಾನ (Demonstrate writing JAVA programs)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್
- JVM. ನ ಆಂತರಿಕ ರಚನೆ.

JVM (Java Virtual Machine) ರಚನೆ

JVM (Java Virtual Machine) ಇದು ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮಶೀನ್ ಆಗಿದೆ. ಇದು ಜಾವಾ ಬೈಟ್‌ಕೋಡ್ ಕಾರ್ಯಗತವಾಗುವಂತೆ ಮಾಡುವ ಒಂದು ರನ್‌ಟೈಮ್ ಪರಿಸರವನ್ನು ಒದಗಿಸುವ ಸ್ಪೆಸಿಫಿಕೇಷನ್ ಆಗಿದೆ.

ಅನೇಕ ಹಾರ್ಡ್‌ವೇರ್ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ಗಳಿಗೆ JVM ಗಳು ಲಭ್ಯವಿರುತ್ತವೆ. (ಎಂದರೆ JVM ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಅವಲಂಬಿತವಾಗಿದೆ)

JVM ಎಂದರೇನು? (What is JVM)

JVM ಎಂದರೆ:

1 **ಸ್ಪೆಸಿಫಿಕೇಷನ್**(specification) - ಇಲ್ಲಿ Java Virtual Machine ನ ಕಾರ್ಯವನ್ನು ವಿವರಿಸಲಾಗಿದೆ. ಆದರೆ ಇಂಪ್ಲಿಮೇಂಟೇಷನ್ ಪ್ರೊವೈಡರ್ ಅಲ್ಗೊರಿದಮ್‌ನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಳ್ಳಲು ಸ್ವತಂತ್ರವಾಗಿರುತ್ತದೆ. ಇದರ ಅನುಷ್ಠಾನವನ್ನು ಒರ್ಯಾಕಲ್ ಮತ್ತು ಇತರ ಕಂಪನಿಗಳು ಒದಗಿಸುತ್ತವೆ.

2 **implementation (ಅನುಷ್ಠಾನ)** - ಇದರ ಅನುಷ್ಠಾನವನ್ನು JRE (Java Runtime Environment) ಎಂದು ಕರೆಯುತ್ತಾರೆ.

3 **Runtime Instance** ನೀವು ಜಾವಾ ಕ್ಲಾಸ್‌ನ್ನು ರನ್ ಮಾಡಲು ಯಾವಾಗ ಕಮಾಂಡ್ ಪ್ರಾಂಪ್ಟ್‌ನಲ್ಲಿ ಜಾವಾ ಕಮಾಂಡ್‌ನ್ನು ಬರೆಯುತ್ತೀರೊ, ಆಗ JVM ನ instance (ನಿದರ್ಶನ) ವು ರಚಿಸಲ್ಪಡುತ್ತದೆ.

ಇದು ಏನು ಮಾಡುತ್ತದೆ. (What it does)

JVM ಕೆಳಗಿನ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ:

- ಕೋಡ್‌ನ್ನು ಲೋಡ್ ಮಾಡುತ್ತದೆ (Loads code)
- ಕೋಡ್‌ನ್ನು ಪರಿಶೀಲಿಸುತ್ತದೆ (Verifies code)
- ಕೋಡ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ (Executes code)
- ರನ್‌ಟೈಮ್ ಪರಿಸರವನ್ನು ಒದಗಿಸುತ್ತದೆ (Provides runtime environment)

JVM ಕೆಳಗಿನವುಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ :

- ಮೆಮೊರಿ ಪ್ರದೇಶ (Memory area)
- ಕ್ಲಾಸ್ ಫೈಲ್ ಫಾರ್ಮ್ಯಾಟ್ (Class file format)
- ರೆಜಿಸ್ಟರ್ ಸೆಟ್ (Register set)
- ಗಾರ್ಬೇಜ್ - ಕಲೆಕ್ಟೆಡ್ ಹೀಪ್ (Garbage-collected heap)
- ಫೆಟಲ್ ಎರರ್ ರಿಪೋರ್ಟಿಂಗ್ (Fatal error reporting) ಇತ್ಯಾದಿ

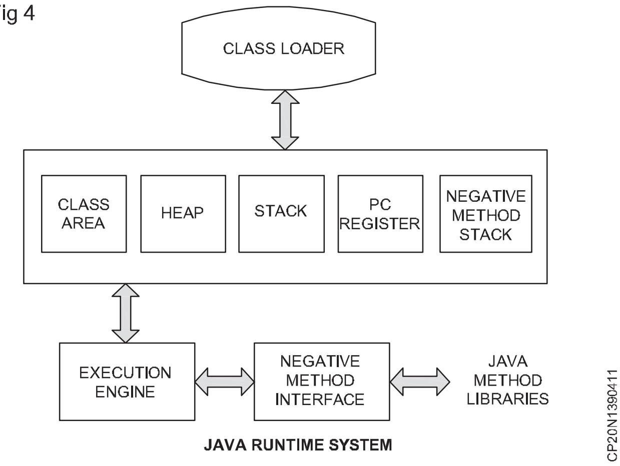
JVM ರಚನೆ (JVM Architecture)

JVM ನ ಆಂತರಿಕ ರಚನೆಯನ್ನು ತಿಳಿದುಕೊಳ್ಳೋಣ. ಇದು ಕ್ಲಾಸ್ ಲೋಡರ್ (classloader), ಮೆಮೊರಿ ಪ್ರದೇಶ (memory area) ,ಎಕ್ಸಿಕ್ಯೂಷನ್ ಇಂಜಿನ್(execution engine) ಮುಂತಾದವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

1 Classloader

ಕ್ಲಾಸ್‌ಲೋಡರ್ JVM ನ ಸಬ್‌ಸಿಸ್ಟಮ್ ಆಗಿದೆ. ಇದನ್ನು ಕ್ಲಾಸ್ ಫೈಲ್‌ಗಳನ್ನು ಲೋಡ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ನಾವು ಜಾವಾ ಪ್ರೊಗ್ರಾಂಗಳನ್ನು ಯಾವಾಗ ರನ್ ಮಾಡುತ್ತೇವೋ, ಆಗ ಮೊದಲು ಕ್ಲಾಸ್ ಲೋಡರ್ ಇದನ್ನು ಲೋಡ್ ಮಾಡುತ್ತದೆ. ಜಾವಾದಲ್ಲಿ ಮೂರು ಬಿಲ್ಟ್ - ಇನ್ ಕ್ಲಾಸ್ ಲೋಡರ್‌ಗಳು ಇವೆ.

Fig 4



NOT TO BE REPUBLISHED © NIMI

1 **ಬೂಟ್‌ಸ್ಟ್ರ್ಯಾಪ್ ಲೋಡರ್** (Bootstrap ClassLoader) : ಇದು ಎಕ್ಸ್ಟೆನ್ಷನ್ ಕ್ಲಾಸ್‌ಲೋಡರ್‌ನ ಸೂಪರ್ ಕ್ಲಾಸ್ ಆದ ಮೊದಲನೆ ಕ್ಲಾಸ್ ಲೋಡರ್ ಇದಾಗಿದೆ. ಇದು rt.jar ಫೈಲನ್ನು ಲೋಡ್ ಮಾಡುತ್ತದೆ. ಈ ಫೈಲ್, ಜಾವಾ ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಎಡಿಷನ್‌ನ ಎಲ್ಲಾ ಕ್ಲಾಸ್ ಫೈಲ್‌ಗಳನ್ನು ಉದಾಹರಣೆಗೆ java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes ಇತ್ಯಾದಿಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

2 Extension ClassLoader: ಇದು ಬೂಟ್ ಸ್ಟ್ರ್ಯಾಪ್‌ನ ಚೈಲ್ಡ್ ಕ್ಲಾಸ್‌ಲೋಡರ್ ಆಗಿದೆ ಮತ್ತು ಸಿಸ್ಟಮ್ ಕ್ಲಾಸ್‌ಲೋಡರ್‌ನ ಪೇರೆಂಟ್ ಕ್ಲಾಸ್ ಲೋಡರ್ ಆಗಿದೆ. ಇದು $JAVA_HOME/jre/lib/ext ಡೈರೆಕ್ಟರಿಯಲ್ಲಿರುವ jar ಫೈಲ್‌ಗಳನ್ನು ಲೋಡ್ ಮಾಡುತ್ತದೆ.

3 System/Application ClassLoader: ಇದು ಎಕ್ಸ್‌ಟೆನ್ಷನ್ ಕ್ಲಾಸ್‌ಲೋಡರ್‌ನ ಚೈಲ್ಡ್ ಕ್ಲಾಸ್‌ಲೋಡರ್ ಆಗಿದೆ. ಇದು ಕ್ಲಾಸ್‌ಪಾಥ್ ನಿಂದ ಕ್ಲಾಸ್ ಫೈಲ್‌ಗಳನ್ನು ಲೋಡ್ ಮಾಡುತ್ತದೆ. ಡೀಫಾಲ್ಟ್ ಆಗಿ ಕ್ಲಾಸ್ ಪಾಥ್, ಪ್ರಸ್ತುತ ಡೈರೆಕ್ಟರಿಗೆ ಸೆಟ್ ಆಗಿರುತ್ತದೆ. ನೀವು ಕ್ಲಾಸ್‌ಪಾಥ್‌ನ್ನು "-cp" ಅಥವಾ "-classpath" switch ನ್ನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಬದಲಾಯಿಸಬಹುದು. ಇದನ್ನು ಅಪ್ಲಿಕೇಷನ್ ಕ್ಲಾಸ್ ಲೋಡರ್ ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ.

```
//Let's see an example to print the classloader name
public class ClassLoaderExample
{
public static void main(String[] args)
{
// Let's print the classloader name of current class.
//Application/System classloader will load this class
Class c=ClassLoaderExample.class;
System.out.println(c.getClassLoader());
//If we print the classloader name of String, it will print null because it is an
//in-built class which is found in rt.jar, so it is loaded by-Bootstrap classloader
System.out.println (String.class.get ClassLoader());
}
}
```

Output:

ಇವು ಜಾವಾ ಒದಗಿಸಿದ ಆಂತರಿಕ ಕ್ಲಾಸ್‌ಲೋಡರ್‌ಗಳಾಗಿವೆ. ಒಂದು ವೇಳೆ ನೀವು ನಿಮ್ಮ ಸ್ವಂತ ಕ್ಲಾಸ್‌ಲೋಡರ್‌ನ್ನು ತಯಾರಿಸಲು ಬಯಸಿದರೆ, ನೀವು ಕ್ಲಾಸ್‌ಲೋಡರ್‌ನ ಕ್ಲಾಸ್‌ನ್ನು ವಿಸ್ತರಿಸುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

ಕ್ಲಾಸ್ ಪ್ರದೇಶ (Class (Method) Area)

Class (Method) Area ಪ್ರತಿ ಕ್ಲಾಸ್ ರಚನೆಯನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ runtime constant pool, field ಮತ್ತು method data, ಕೋಡ್‌ಗಳ ಮೆಥೆಡ್ ಇತ್ಯಾದಿಗಳು

ಹೀಪ್ (Heap)

ಇದು ಆಬ್ಜೆಕ್ಟ್‌ಗಳಿಗೆ ಹಂಚಲಾದ ರನ್‌ಟೈಮ್‌ನ ಡಾಟಾ ಪ್ರದೇಶವಾಗಿದೆ.

ಸ್ಟ್ಯಾಕ್ (Stack)

ಜಾವಾ ಸ್ಟ್ಯಾಕ್‌ಗಳು ಫ್ರೇಮ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸುತ್ತವೆ. ಇದು ಸ್ಥಳೀಯ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಮತ್ತು ಅಪೂರ್ಣ ಉತ್ತರಗಳನ್ನು ಸಂಗ್ರಹಿಸುತ್ತದೆ ಮತ್ತು ಮೆಥೆಡ್‌ನ್ನು ಆವಾಹನೆ ಮಾಡಲು ಮತ್ತು ಹಿಂತಿರುಗಿಸಲು ಮುಖ್ಯ ಪಾತ್ರವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ.

ಪ್ರತಿಯೊಂದು ಥ್ರೆಡ್ ಖಾಸಗಿ JVM ಸ್ಟ್ಯಾಕನ್ನು ಹೊಂದಿದೆ, ಮತ್ತು ಥ್ರೆಡ್ ರಚಿಸುವ ಸಮಯದಲ್ಲಿಯೇ ರಚಿಸಲ್ಪಡುತ್ತದೆ.

ಪ್ರತಿ ಬಾರಿ ಮೆಥೆಡ್ ತೆರೆದಾಗ ಒಂದು ಹೊಸ ಫ್ರೇಮ್ ರಚಿಸಲ್ಪಡುತ್ತದೆ. ಮೆಥೆಡ್ ಕಾರ್ಯವು ಪೂರ್ಣಗೊಂಡಾಗ ಆ ಫ್ರೇಮ್ ನಾಶವಾಗುತ್ತದೆ.

ಪ್ರೊಗ್ರಾಂ ಕೌಂಟರ್ ರೆಜಿಸ್ಟರ್ (Program Counter Register)

ಪ್ರಸ್ತುತ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತಿರುವ ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ನ ಅಡ್ರೆಸ್‌ನ್ನು PC (Program Counter) ರೆಜಿಸ್ಟರ ಹೊಂದಿರುತ್ತದೆ.

ನೇಟಿವ್ ಮೆಥೆಡ್ ಸ್ಟ್ಯಾಕ್ (Native Method Stack)

ಇದು ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಎಲ್ಲಾ ಸ್ಥಳೀಯ ಮೆಥೆಡ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಎಕ್ಸಿಕ್ಯೂಷನ್ ಇಂಜಿನ್ (Execution Engine) ಈ ಕೆಳಗಿನವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ:

1 **ವರ್ಚುವಲ್ ಪ್ರೊಸೆಸರ್**

2 **ಇಂಟರ್‌ಪ್ರಿಟರ್** : ಬೈಟ್‌ಕೋಡ್ ಸ್ಟ್ರೀಮನ್ನು ಓದುತ್ತದೆ ಮತ್ತು ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ.

3 Just-In-Time(JIT) **ಕಂಪೈಲರ್** : ಇದು ಕಾರ್ಯಕ್ಷಮತೆಯನ್ನು

© NIMI NOT TO BE REPUBLISHED

ಸುಧಾರಿಸಲು ಉಪಯೋಗಿಸಲ್ಪಡುತ್ತದೆ. ಒಂದೇ ಸಮಯದಲ್ಲಿ ಒಂದೇ ರೀತಿಯ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಬೈಟ್‌ಕೋಡ್‌ನ ಭಾಗಗಳನ್ನು JIT ಕಂಪೈಲ್ ಮಾಡುತ್ತದೆ. ಮತ್ತು ಅದ್ದರಿಂದ ಕಂಪೈಲೇಷನ್ ಮಾಡಲು ತೆಗೆದುಕೊಳ್ಳುವ ಸಮಯದ ಪ್ರಮಾಣವನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ. ಇಲ್ಲಿ "ಕಂಪೈಲರ್"ಎಂಬ ಪದವು ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್ (JVM) ನ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್ ಸೆಟ್‌ನಿಂದ ನಿರ್ದಿಷ್ಟ CPU ದ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್ ಸೆಟ್‌ಗೆ ಭಾಷಾಂತರ ಮಾಡುವುದು ಎಂದಾಗಿದೆ.

ಜಾವಾ ನೇಟಿವ್ ಇಂಟರ್‌ಫೇಸ್ (Java Native Interface): Java Native Interface(JNI) ಒಂದು ಫ್ರೇಮ್‌ವರ್ಕ್ ಆಗಿದ್ದು, ಇದು C, C++, ಅಸೆಂಬ್ಲಿ ಮುಂತಾದ ಇತರ ಭಾಷೆಗಳಲ್ಲಿ ಬರೆಯಲಾದ ಇತರ ಅಪ್ಲಿಕೇಷನ್‌ಗಳೊಂದಿಗೆ ಸಂವಹನ ಮಾಡಲು ಮಾಧ್ಯಮವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಕನ್ಸೋಲ್‌ಗೆ ಫಲಿತಾಂಶವು ಕಳುಹಿಸಲು ಅಥವಾ OS ಲೈಬ್ರರಿಗಳೊಂದಿಗೆ ಸಂವಹನ ನಡೆಸಲು, ಜಾವಾ JNI ಫ್ರೇಮ್‌ವರ್ಕ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್ ಎಂದರೇನು ?

ಅವಲೋಕನ

ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್ ಎಂದರೆ ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್‌ಗೆ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಸೆಟ್ ಆಗಿದೆ. ಇದು ಅವುಗಳನ್ನು ಇಂಟರ್‌ಪ್ರಿಟ್ ಮಾಡುವ ಜವಾಬ್ದಾರಿಯನ್ನು ಹೊಂದಿದೆ. ಜಾವಾ ಪ್ರೊಗ್ರಾಂನ್ನು ಕಂಪೈಲ್ ಮಾಡಿದ ಮೇಲೆ, ಜಾವಾ ಬೈಟ್‌ಕೋಡ್ ಉತ್ಪಾದನೆಯಾಗುತ್ತದೆ. ಇದು JVM ಬಳಸಿಕೊಂಡು ಯಾವುದೇ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್‌ನಲ್ಲಿ (ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರ) ಕಾರ್ಯಗತವಾಗುತ್ತದೆ. ಇದು .class ಫೈಲಿನಲ್ಲಿ ಇರುತ್ತದೆ.

ವ್ಯಾಪ್ತಿ (Scope)

ಈ ಲೇಖನವು ಕೆಳಗಿನ ಗುರಿಯನ್ನು ಹೊಂದಿದೆ:

- ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್‌ನ ಪರಿಕಲ್ಪನೆಯನ್ನು ವಿವರಿಸುವುದು.
- ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್‌ನ ಅನುಕೂಲಗಳು ಮತ್ತು ಅನಾನುಕೂಲತೆಗಳನ್ನು ಅರ್ಥಮಾಡಿಕೊಳ್ಳಲು ಸಹಾಯ ಮಾಡುವುದು.

ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್ ಎಂದರೇನು?

ಪ್ರತಿಯೊಂದು ಸಾಧನಗಳಿಗೆ ವಿಭಿನ್ನ ವಿಶೇಷಣಗಳೊಂದಿಗೆ (specifications) ಹಲವಾರು ಬಾರಿ ಕೋಡನ್ನು ಬರೆಯಬೇಕಾದ ವಿಷಯವನ್ನು ನೀವು ಊಹಿಸಿಕೊಳ್ಳಬಹುದೇ? ಉದಾಹರಣೆಗೆ ಆಪರೇಟಿಂಗ್ ಸಿಸ್ಟಮ್, ಪ್ರೊಸೆಸರ್ ಆರ್ಕಿಟೆಕ್ಚರ್ ಇತ್ಯಾದಿ. ಇದು ಬೇಸರದ ಪ್ರಕ್ರಿಯೆಯಾಗಿದೆ. ಈ ವಿಷಯಗಳನ್ನು ಗಮನದಲ್ಲಿಟ್ಟುಕೊಂಡು, ಜಾವಾ ಡೆವಲಪರ್‌ಗಳು Write Once Read Anywhere(ಒಮ್ಮೆ ಬರೆಯಿರಿ, ಎಲ್ಲಿಯಾದರೂ ಓದಿರಿ) ಎಂಬ ಪರಿಕಲ್ಪನೆಯೊಂದಿಗೆ ಬಂದರು. ಇದು ಜಾವಾ ಭಾಷೆಯ ವೈಶಿಷ್ಟ್ಯಗಳೊಂದಿಗೆ ಸಂಯೋಜಿಸಲ್ಪಟ್ಟಿದೆ. ಇದನ್ನು ಸಾಧಿಸಲು, ಬೈಟ್‌ಕೋಡ್ ಎನ್ನುವ ಕೋಡ್‌ನ್ನು ನಮ್ಮ ಜಾವಾ ಸೋರ್ಸ್ ಕೋಡನ್ನು ಕಂಪೈಲ್ ಮಾಡಿದ ನಂತರ ಉತ್ಪದಿಸಲಾಗುತ್ತದೆ.ಇದನ್ನೇ ನಾವು ಈಗ ಚರ್ಚಿಸೋಣ.

ಜಾವಾದಲ್ಲಿನ ಬೈಟ್‌ಕೋಡ್ ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್‌ಗಾಗಿ ಇರುವ ಒಂದು ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಗುಂಪಾಗಿದೆ. ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್ ಶಬ್ದವನ್ನು JVM ಎಂದು ಸಂಕ್ಷೇಪ್ತಗೊಳಿಸಲಾಗಿದೆ. ಇದು ಜಾವಾದಲ್ಲಿ ಬರೆಯಲಾದ ಕೋಡ್ ಅನ್ನು ರನ್ ಮಾಡಲು ಕಂಪ್ಯೂಟರ್‌ನ್ನು ಸಕ್ರೀಯಗೊಳಿಸುತ್ತದೆ. ಯಾವಾಗ ಒಂದು ಜಾವಾ ಪ್ರೊಗ್ರಾಂ ಕಂಪೈಲ್ ಆಗುತ್ತದೆಯೊ, ಆಗ ಬೈಟ್‌ಕೋಡ್ ಸೃಷ್ಟಿಯಾಗುತ್ತದೆ. ಇದು C++ ನಲ್ಲಿ ಇರುವ ಅಸೆಂಬ್ಲರ್‌ಗೆ ಸಮನಾಗಿರುತ್ತದೆ.

ಬೈಟ್‌ಕೋಡ್ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರವಾದ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಗುಂಪಾಗಿದ. ಏಕೆಂದರೆ ಇದು JVM ನಿಂದ ಇಂಟರ್‌ಪ್ರಿಟ್ ಮಾಡಲಾಗುತ್ತದೆ ಮತ್ತು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ. ಅನೇಕ ಕಂಪ್ಯೂಟರ್ ಆರ್ಕಿಟೆಕ್ಚರ್‌ಗಳಲ್ಲಿ ಯಾವುದೇ ಮಾರ್ಪಾಡುಗಳಿಲ್ಲದೇ ರನ್ ಆಗುವ ಕೊಡ್ ಅನ್ನು ಮಶೀನ್ ಅಥವಾ ಪ್ಲಾಟ್‌ಫಾರ್ಮ್ ಇಂಡಿಪೆಂಡೆಂಟ್(ಸ್ವತಂತ್ರ) ಕೋಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಬೈಟ್‌ಕೋಡ್ ಲೊ - ಲೆವೆಲ್ ಮತ್ತು ಹೈ- ಲೆವೆಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಗುಂಪುಗಳ ನಡುವೆ ಇರುತ್ತದೆ. ನಾವು ನಮ್ಮ ಕೋಡ್ ಅನ್ನು ಹೈ - ಲೆವೆಲ್ ಭಾಷೆಯಲ್ಲಿ ಬರೆಯುವುದರಿಂದ ಇದು ಬೈಟ್‌ಕೋಡ್‌ಗೆ ಕಂಪೈಲ್ ಆಗುತ್ತದೆ ಮತ್ತು ನಂತರ JVM ಅದನ್ನು ಮಶೀನ್ ಕೋಡ್‌ಗೆ ಇಂಟರ್‌ಪ್ರಿಟ್ ಮಾಡುತ್ತದೆ. ಮಶೀನ್ ಕೋಡ್, ಲೋ ಲೆವೆಲ್ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಗುಂಪಾಗಿದ್ದು ಕಾರ್ಯಗತಗೊಳ್ಳಲು ಸಿದ್ಧವಾಗಿರುತ್ತದೆ. ಅದ್ದರಿಂದ ಬೈಟ್‌ಕೋಡನ್ನು ನಾವು ಲೋ ಲೆವೆಲ್ ಮತ್ತು ಹೈ ಲೆವೆಲ್ ಭಾಷೆಗಳ ನಡುವಿನ ಕೋಡ್ ಎಂದು ಕರೆಯುತ್ತೇವೆ.

ಸತ್ಯ :

ಜಾವಾ ಕಂಪೈಲ್ ಮತ್ತು ಇಂಟರ್‌ಪ್ರಿಟ್ ಎರಡನ್ನೂ ಮಾಡುವ ಭಾಷೆಯಾಗಿದೆ. ಇತರ ಭಾಷೆಗಳನ್ನು ಒಂದೊ ಕಂಪೈಲ್ ಮಾಡಬಹುದು ಅಥವಾ ಇಂಟರ್‌ಪ್ರಿಟ್ ಮಾಡಬಹುದು.

ಇದು ಕೇಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ?

ಮೇಲಿನ ಪ್ರಕ್ರಿಯೆಯನ್ನು ವಿವರಣೆಯೊಂದಿಗೆ ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳೋಣ:

- ನಾವು ಜಾವಾದಲ್ಲಿ ಬರೆಯಲಾದ ಕೋಡ್‌ನ್ನು ಸೋರ್ಸ್ ಕೋಡ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ. ಇದನ್ನು ಹೈ -ಲೆವೆಲ್ ಭಾಷೆಯಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ. ಹೈ - ಲೆವೆಲ್ ಭಾಷೆ ಪ್ರೊಗ್ರಾಮರ್ ಸ್ನೇಹಿ

© NIMI NOT TO BE REPUBLISHED

ಭಾಷೆಯಾಗಿದೆ. ಇದರಲ್ಲಿ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಇಂಗ್ಲಿಷ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ ಮತ್ತು ಇದು ಮಾನವರ ಭಾಷೆಗಳಿಗೆ ಹತ್ತಿರವಾಗಿದೆ. ಜಾವಾ ಫೈಲ್‌ನ ವಿಸ್ತರಣಾ ಹೆಸರು (extension name) ".java" ಎಂದಾಗಿದೆ.

- ನಾವು ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ಕಂಪೈಲ್ ಮಾಡಿದಾಗ, ಕಂಪೈಲರ್ ".java" ಫೈಲನ್ನು ಕಂಪೈಲ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ".class" ಫೈಲನ್ನು ಸೃಷ್ಟಿಸುತ್ತದೆ. ಇದು ಬೈಟ್‌ಕೋಡನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.
- ಬೈಟ್‌ಕೋಡ್ ನಮಗೆ ಯಾವುದೇ ಇತರ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್‌ನಲ್ಲಿ ".class" ಫೈಲ್‌ನು ರನ್ ಮಾಡಲು ಅನುಮತಿ ನೀಡುತ್ತದೆ.
- ಆದರೆ ಈ ಬೈಟ್‌ಕೋಡ್‌ಗೆ ಅದನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಇಂಟರ್‌ಪ್ರಿಟರ್‌ನ ಅವಶ್ಯಕತೆ ಇದೆ. ಇಲ್ಲಿ JVM ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ. JVM ಇಂಟರ್‌ಪ್ರಿಟರ್‌ನ್ನು ಹೊಂದಿದೆ. ಇದು ಕೋಡ್‌ನ್ನು ತುಂಡು ತುಂಡಾಗಿ ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ. ಎಂದರೆ, ಒಂದು ಸಮಯದಲ್ಲಿ ಒಂದೇ ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ದೋಷ ಕಂಡುಬರುವವರೆಗೆ ನಿರ್ವಹಿಸುತ್ತದೆ ಅಥವಾ ಕೋಡ್‌ನ ಕೊನೆಯವರೆಗೆ ಕಾರ್ಯಗತಗೊಳಿಸಿ ಮುಗಿಸುವುದು.

ಅದ್ದರಿಂದ, ಬೈಟ್‌ಕೋಡನ್ನು ಲೋ -ಲೆವೆಲ್ ಭಾಷೆ ಮತ್ತು ಹೈ -ಲೆವೆಲ್ ಭಾಷೆಯ ನಡುವೆ ಇರುತ್ತದೆ ಎಂದು ಹೇಳಲಾಗುತ್ತದೆ.

ಒಂದು ಉದಾಹರಣೆಯನ್ನು ನೋಡೋಣ-

ನೀವು ಕ್ಯಾಲ್ಕ್ಯುಲೇಟರ್ ಆ್ಯಪ್‌ಗಾಗಿ ಜಾವಾದಲ್ಲಿ ಒಂದು ಕೋಡ್ ಬರೆದಿದ್ದೀರಿ ಎಂದು ಭಾವಿಸೋಣ. ಇದು ಹೇಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ ಎಂಬುದನ್ನು ನಾವು ಈ ರೀತಿಯಾಗಿ ಸುಲಭವಾಗಿ ಅರ್ಥ ಮಾಡಿಕೊಳ್ಳೋಣ.

ಅನುಕೂಲಗಳು (Advantages)

ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರ (Platform Independence)

ಒಂದೇ ಕೋಡ್ ಅನ್ನು ಯಾವುದೇ ಸಾಧನದಲ್ಲಿ ಯಾವುದೇ ಬದಲಾವಣೆಗಳಿಲ್ಲದೇ ರನ್ ಮಾಡುವ ಉದ್ದೇಶದಿಂದ ಜಾವ ಪ್ರೊಜೆಕ್ಟ್‌ನ್ನು ಪ್ರಾರಂಭಿಸಲಾಯಿತು. ಜಾವಾದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್‌ನ ಪರಿಕಲ್ಪನೆಯು ಈ ಗುರಿಯನ್ನು ಸಾಧಿಸುವುದರಲ್ಲಿ ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಬೈಟ್‌ಕೋಡ್‌ಗಳು ವಿಭಿನ್ನ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ವಿಭಿನ್ನವಾಗಿರಬಹುದು ಆದರೆ ಅವು ಎಲ್ಲಾ ಸಿಸ್ಟಮ್‌ಗಳಲ್ಲಿ ಒಂದೇ ರೀತಿಯಿಂದ ಇಂಟರ್‌ಪ್ರಿಟ್ ಆಗುತ್ತವೆ ಮತ್ತು ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತವೆ.

ಬೈಟ್‌ಕೋಡ್ ಮೂಲಭೂತವಾಗಿ ಮಶೀನ್ - ಲೆವೆಲ್ ಕೋಡ್ ಆಗಿದೆ ಮತ್ತು ಜಾವಾ ವರ್ಚುವಲ್ ಮಶಿನ್‌ನಲ್ಲಿ ರನ್ ಆಗುತ್ತದೆ. ಒಂದು ಕ್ಲಾಸ್ ಲೋಡ್ ಆದಾಗ, ಇದು ಪ್ರತಿ ಕ್ಲಾಸ್ ಮೆಥೆಡ್‌ಗೆ ಬೈಟ್‌ಕೋಡ್‌ನ ಸ್ಟ್ರೀಮ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ. ಒಂದು ಪ್ರೊಗ್ರಾಂ ಕಾರ್ಯಗತಗೊಳ್ಳುವ ಸಮಯದಲ್ಲಿ ಆ ಮೆಥೆಡ್‌ನ್ನು ಕರೆದಾಗ, ಆ ಮೆಥೆಡ್‌ನ ಬೈಟ್‌ಕೋಡ್‌ನ್ನು ಆಹ್ವಾನಿಸಲಾಗುತ್ತದೆ. ಜಾವಾ ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ಕಂಪೈಲ್ ಮಾಡುವುದಲ್ಲದೇ ಆ ಪ್ರೊಗ್ರಾಂಗಾಗಿ ಬೈಟ್‌ಕೋಡ್‌ನ್ನು ಸಹ ಸೃಷ್ಟಿಸುತ್ತದೆ. ಹೀಗಾಗಿ ಜಾವಾ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ ಸ್ವತಂತ್ರ ಭಾಷೆಯಾಗಿದೆ ಎಂಬುದನ್ನು ನಾವು ಅರಿತುಕೊಂಡಿದ್ದೇವೆ.

ಪೋರ್ಟೇಬಿಲಿಟಿ (Portability)

ಇದು ಪೋರ್ಟೇಬಿಲಿಟಿಯನ್ನು ಸೇರಿಸಲು ಸಹಾಯ ಮಾಡುತ್ತದೆ. ಇದು ಕೋಡ್ ಅನ್ನು ವಿವಿಧ ಸಾಧನಗಳಲ್ಲಿ ರನ್ ಮಾಡಬಹುದು ಎಂದು ಖಚಿತಪಡಿಸುತ್ತದೆ. ಇದು "write once, run anywhere" (ಒಮ್ಮೆ ಬರೆಯಿರಿ, ಎಲ್ಲಿಯಾದರೂ ರನ್ ಮಾಡಿರಿ) ಎನ್ನುವ ಜಾವಾದಿಂದ ಅನುಷ್ಠಾನಗೊಂಡ ತತ್ವವನ್ನು ಚೆನ್ನಾಗಿ ಪ್ರತಿಧ್ವನಿಸುತ್ತದೆ. ಯಾವುದೇ ಇತರ ಸಾಧನಗಳಲ್ಲಿ ನಾವು ಮತ್ತೊಮ್ಮೆ ಕೋಡ್‌ನ್ನು ಬರೆಯುವ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ.

ಅನಾನುಕೂಲಗಳು (Disadvantages)

- ಇಂಟರ್‌ಪ್ರಿಟರ್ ಅಥವಾ JVM ಇಲ್ಲದೇ ಬೈಟ್‌ಕೋಡ್ ರನ್ ಆಗುವುದಿಲ್ಲ. ಒಂದು ವೇಳೆ ಯಾವುದೇ ಸಾಧನವು JVM ನ್ನು ಹೊಂದಿಲ್ಲದಿದ್ದರೆ, ಆ ಸಾಧನದಲ್ಲಿ ಬೈಟ್‌ಕೋಡ್ ರನ್ ಆಗುವುದಿಲ್ಲ.
- ಬೈಟ್‌ಕೋಡ್ ಬೈನರಿ ರೂಪದಲ್ಲಿರುವುದರಿಂದ ಅದನ್ನು ವಿಶ್ಲೇಷಣೆ ಮಾಡುವುದು ಕಠಿಣವಾಗಿರುತ್ತದೆ ಮತ್ತು ಮನುಷ್ಯರಿಗೆ ಅರ್ಥವಾಗುವುದಿಲ್ಲ.

ತೀರ್ಮಾನ (Conclusion)

- ಜಾವಾದಲ್ಲಿನ ಬೈಟ್‌ಕೋಡ್ ಜಾವಾ ವರ್ಚುವಲ್ ಮಶೀನ್‌ಗಾಗಿ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ ಗುಂಪಾಗಿದೆ.
- ಬೈಟ್‌ಕೋಡ್ ಪ್ಲ್ಯಾಟ್‌ಫಾರ್ಮ್ - ಸ್ವತಂತ್ರ ಕೋಡ್ ಆಗಿದೆ.
- ಬೈಟ್‌ಕೋಡ್ ಲೋ - ಲೆವೆಲ್ ಭಾಷೆ ಮತ್ತು ಹೈ- ಲೆವೆಲ್ ಭಾಷೆ ಎರಡರ ನಡುವೆ ಇರುವ ಕೋಡ್ ಆಗಿದೆ.
- ಜಾವಾ ಕೋಡ್ ಕಂಪೈಲ್ ಆದ ನಂತರ, ಬೈಟ್‌ಕೋಡ್ ರಚನೆಯಾಗುತ್ತದೆ. ಇದು JVM ಉಪಯೋಗಿಸಿ ಯಾವುದೇ ಮಶೀನ್‌ನಲ್ಲಿ ಕಾರ್ಯಗತವಾಗುತ್ತದೆ.
- ನಾವು ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ಕಂಪೈಲ್ ಮಾಡಿದಾಗ, ಕಂಪೈಲರ್ ".java" ಫೈಲನ್ನು ಕಂಪೈಲ್ ಮಾಡುತ್ತದೆ ಮತ್ತು ".class" ಫೈಲನ್ನು ರಚಿಸುತ್ತದೆ.
- ಬೈಟ್‌ಕೋಡ್‌ನ್ನು ರನ್ ಮಾಡಲು ಅದಕ್ಕೆ ಇಂಟರ್‌ಪ್ರಿಟರ್‌ನ ಅಗತ್ಯವಿದೆ; ಅದ್ದರಿಂದ JVM ಇಂಟರ್‌ಪ್ರಿಟರ್ ಆಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

- ಜಾವಾವನ್ನು ಕಂಪೈಲ್ ಮತ್ತು ಇಂಟರ್‌ಪ್ರಿಟ್ ಎರಡೂ ಮಾಡಲಾಗುತ್ತದೆ.

ಜಾವಾದಲ್ಲಿ Classpath ಅನ್ನು ಹೇಗೆ ಸೆಟ್ ಮಾಡುವುದು ? (How to Set Classpath in Java?)

- Read (ಓದುವುದು)
- Discuss (ಚರ್ಚಿಸುವುದು)
- Practice (ಅಭ್ಯಸಿಸುವುದು)
- Video (ವಿಡಿಯೊ)
- Courses (ಕೋರ್ಸ್‌ಗಳು)

ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಅವಶ್ಯಕವಿರುವ ಎಲ್ಲ ಫೈಲ್‌ಗಳು ಲಭ್ಯವಿರುವ ಸ್ಥಳವನ್ನು CLASSPATH ವಿವರಿಸುತ್ತದೆ. ಜಾವಾ ಕಂಪೈಲರ್ ಮತ್ತು JVM (Java virtual Machine) ಅಗತ್ಯವಿರುವ ಫೈಲುಗಳನ್ನು ಪತ್ತೆ ಮಾಡಲು CLASSPATH ನ್ನು ಉಪಯೋಗಿಸುತ್ತವೆ. CLASSPATH ನ್ನು ಸೆಟ್ ಮಾಡದಿದ್ದರೆ, ಜಾವಾ ಕಂಪೈಲರ್ ಅಗತ್ಯವಿರುವ ಫೈಲುಗಳನ್ನು ಹುಡುಕಲು ಸಾಧ್ಯವಾಗುವುದಿಲ್ಲ ಮತ್ತು ಕೆಳಗಿನ ದೋಷವನ್ನು ತೋರಿಸುತ್ತದೆ.

ದೋಷ (Error): Could not find or load main class <class name>

(e.g. GFG)

CLASSPATH ನ್ನು ಸೆಟ್ ಮಾಡಿ ಮೇಲಿನ ದೋಷವನ್ನು ಪರಿಹರಿಸಲಾಗುತ್ತದೆ.

Java

```
    // If the following code is run when the CLASSPATH is
    not
    // set, it will throw the above error.
    // If it is set, we get the desired result
    importjava.io.*;
    classGFG {
publicstaticvoidmain(String[] args)
{
     // prints GeeksForGeeks to the console
     System.out.println("GeekForGeeks!");
 }
}
```

ಫಲಿತಾಂಶ :

GeekForGeeks!

ವಿಂಡೊಸ್‌ನಲ್ಲಿ ಜಾವಾದಲ್ಲಿ CLASSPATH ನ್ನು ಸೆಟ್ ಮಾಡಿ (Set the CLASSPATH in JAVA in Windows)

Command Prompt:

set PATH=.;C:\Program Files\Java\JDK1.6.20\bin

ಸೂಚನೆ : ಮೇಲಿನ ಕಮಾಂಡ್‌ನಲ್ಲಿ, ಸೆಮಿಕೋಲನ್ ಚಿಹ್ನೆಯನ್ನು (;) ಸೆಪರೇಟರ್ ಆಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು dot (.) ಅನ್ನು CLASSPATH ನ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯವನ್ನಾಗಿ ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.

GUI:

1 Start ಆಯ್ಕೆ ಮಾಡಿರಿ.

2 Control Panel ಗೆ ಹೋಗಿರಿ

PICTURE

3 System and Security ಆಯ್ಕೆ ಮಾಡಿರಿ

PICTURE

4 Advanced System settings ನ್ನು ಆಯ್ಕೆ ಮಾಡಿರಿ

PICTURE

5 Environment Variables ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡಿರಿ

PICTURE

6 New under System Variables ಕ್ಲಿಕ್ ಮಾಡಿರಿ

PICTURE

7 CLASSPATH ಅನ್ನು ವೇರಿಯೇಬಲ್ ಹೆಸರನ್ನಾಗಿ ಮತ್ತು ಫೈಲ್‌ನ ಪಾಥ್ ಆಗಿ ಸೇರಿಸಿ (Add CLASSPATH as variable name and path of files as a variable value.)

PICTURE

OK ಆಯ್ಕೆ ಮಾಡಿರಿ

ಲಿನಕ್ಸ್‌ನಲ್ಲಿ CLASSPATH ಸೆಟ್ ಮಾಡಿರಿ (Set the CLASSPATH on Linux)

Command Line:

ನೀವು ಜಾವಾವನ್ನು ಎಲ್ಲಿ ಇನ್ಸ್ಟಾಲ್ ಮಾಡಿರುವಿರಿ ಎಂಬುದನ್ನು ಕಂಡುಹಿಡಿಯಿರಿ , ಸಾಮಾನ್ಯವಾಗಿ ಅದು /usr/lib/jvm ಪಾಥ್‌ನಲ್ಲಿರುತ್ತದೆ. CLASSPATH ನ್ನು /etc/environment ನಲ್ಲಿ ಸೆಟ್ ಮಾಡಿರಿ.

© NIMI NOT TO BE REPUBLISHED

sudo <editor name> /etc/environment ನ್ನು ಬಳಸಿ ಸೆಟ್ ಮಾಡಿರಿ.

ಕೆಳಗಿನ ಸಾಲುಗಳನ್ನು ಸೇರಿಸಿರಿ,

JAVA_HOME = "/usr/lib/jvm/<java folder (eg. java-1.8.0-openjdk-amd64>)/bin"

export JAVA_HOME

CLASSPATH=".:/usr/lib/jvm/<java folder>/lib:/home/name/Desktop"

export CLASSPATH

ಗಮನಿಸಿ: ಡೈರೆಕ್ಟರಿಯನ್ನು ಬೇರ್ಪಡಿಸಲು ಕೋಲನ್ ಚಿಹ್ನೆಯನ್ನು (:) ಬಳಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಮೇಲಿನ ಕಮಾಂಡ್‌ನಲ್ಲಿ dot(.) ಚಿಹ್ನೆಯು CLASSPATH ನ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯವಾಗಿದೆ.

ಪ್ರಸ್ತುತ CLASSPATH ನ್ನು ಪರಿಶೀಲಿಸಲು,

echo ${CLASSPATH} ನ್ನು ರನ್ ಮಾಡಿರಿ.

ಕೀವರ್ಡ್‌ಗಳು (Keywords)

ಜಾವಾ ಕೀವರ್ಡ್‌ಗಳು (JavaKeywords)

ಜಾವಾ ಕೀವರ್ಡ್‌ಗಳನ್ನು ರಿಸರ್ವ್ಡ್ ವರ್ಡ್‌ಗಳು ಎಂದೂ ಕರೆಯುತ್ತಾರೆ. ಕೀವರ್ಡ್‌ಗಳು ಒಂದು ಕೋಡ್‌ಗೆ ಕೀಲಿ ಯಾಗಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ನಿರ್ದಿಷ್ಟ ಪದಗಳಾಗಿವೆ. ಇವು ಜಾವಾದ ಪೂರ್ವನಿರ್ಧಾರಿತ ಪದಗಳಾಗಿರುತ್ತವೆ. ಅದ್ದರಿಂದ ಇವುಗಳನ್ನು ವೇರಿಯೇಬಲ್‌ನ ಹೆಸರನ್ನಾಗಿಯೋ ಅಥವಾ ಆಬ್ಜೆಕ್ಟ್‌ನ ಹೆಸರನ್ನಾಗಿಯೊ ಅಥವಾ ಕ್ಲಾಸ್‌ನ ಹೆಸರನ್ನಾಗಿಯೊ ಉಪಯೋಗಿಸಲು ಬರುವುದಿಲ್ಲ.

ಜಾವಾ ಕೀವರ್ಡ್‌ಗಳ ಪಟ್ಟಿ (List of Java Keywords)

ಜಾವಾ ಕೀವರ್ಡ್‌ಗಳ ಅಥವಾ ರಿಸರ್ವ್ಡ್ ವರ್ಡ್‌ಗಳ ಪಟ್ಟಿಯನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ:

1 **abstract:** ಜಾವಾ abstract ಕೀವರ್ಡ್‌ನ್ನು abstract ಕ್ಲಾಸ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಒಂದು abstract ಕ್ಲಾಸ್ ಇಂಟರ್‌ಫೇಸ್‌ನ ಅನುಷ್ಠಾನವನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು abstract ಮತ್ತು non - abstract ಮೆಥೆಡ್‌ಗಳನ್ನು ಹೊಂದಿರುತ್ತದೆ.

2 **boolean:** ಒಂದು ವೇರಿಯೇಬಲ್‌ನ್ನು ಬೂಲೀನ್ ಡಾಟಾಟೈಪ್ ಆಗಿ ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ boolean (ಬೂಲಿನ್) ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು True ಮತ್ತು False ಮೌಲ್ಯಗಳನ್ನು ಮಾತ್ರ ಸಂಗ್ರಹಿಸಬಲ್ಲದು.

3 **break:** ಜಾವಾ break ಕೀವರ್ಡ್‌ನ್ನು ಲೂಪ್ ಅಥವಾ switch ಸ್ಟೇಟಮೆಂಟ್ ಮುರಿಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಂಡಿಷನ್ ಪ್ರಕಾರ ಪ್ರೊಗ್ರಾಂನ ಪ್ರಸ್ತುತ ಹರಿವನ್ನು ಮುರಿಯುತ್ತದೆ.

4 **byte:** 8 - ಬಿಟ್ ಡಾಟಾ ಮೌಲ್ಯವನ್ನು ಸಂಗ್ರಹಿಸುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವ byte ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

5 **case:** ಟೆಕ್ಸ್ಟ್‌ನ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಮಾರ್ಕ್ ಮಾಡಲು switch ಸ್ಟೇಟಮೆಂಟ್ ಜೊತೆಗೆ ಜಾವಾ case ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ

6 **catch:** try ಸ್ಟೇಟಮೆಂಟ್‌ನಿಂದ ರಚಿಸಲಾದ ಎಕ್ಸೆಪ್ಷನ್‌ಗಳನ್ನು catch ಮಾಡಲು ಜಾವಾ catch ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

7 **char:** unsigned 16 - ಬಿಟ್ ಯುನಿಕೊಡ್ ಕ್ಯಾರೆಕ್ಟರ್‌ಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ವೇರಿಯೇಬಲ್ ಡಿಕ್ಲೇರ್ ಮಾಡುವ ಸಲುವಾಗಿ ಜಾವಾ char ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

8 **class:** ಕ್ಲಾಸ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಜಾವಾ class ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.

9 **continue:** ಲೂಪನ್ನು ಮುಂದುವರೆಸಲು ಜಾವಾ continue ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಪ್ರೊಗ್ರಾಂನ ಪ್ರಸ್ತುತ ಹರಿವನ್ನು ಮುಂದುವರೆಸುತ್ತದೆ ಮತ್ತು ನಿರ್ದಿಷ್ಟ ಕಂಡೀಷನ್‌ನಲ್ಲಿ ಇನ್ನುಳಿದ ಕೋಡನ್ನು ಬಿಟ್ಟುಬಿಡುತ್ತದೆ.

10 **default:** switch ಸ್ಟೇಟಮೆಂಟ್‌ನಲ್ಲಿ ಡೀಫಾಲ್ಟ್ ಬ್ಲಾಕ್ ಕೋಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಜಾವಾ default ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

11 **do:** ಕಂಟ್ರೋಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ನಲ್ಲಿ ಲೂಪನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ do ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಪ್ರೊಗ್ರಾಂನ ಒಂದು ಭಾಗವನ್ನು ಹಲವಾರು ಬಾರಿ ಪುನರಾವರ್ತಿಸುತ್ತದೆ.

12 **double:** 64-bitfloating-point ಸಂಖ್ಯೆಯನ್ನು ಸಂಗ್ರಹಿಸುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ double ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

13 **else:** if ಸ್ಟೇಟಮೆಂಟ್‌ನಲ್ಲಿ ಪರ್ಯಾಯ ಶಾಖೆಗಳನ್ನು ಸೂಚಿಸಲು ಜಾವಾ else ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

14 **enum:** ಸ್ಥಿರ constant ಗಳ ಗುಂಪನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಜಾವಾ enum ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. Enum ಕನ್‌ಸ್ಟ್ರಕ್ಟರ್‌ಗಳು ಯಾವಾಗಲೂ ಖಾಸಗಿಯಾಗಿರುತ್ತವೆ ಅಥವಾ ಡೀಫಾಲ್ಟ್ ಆಗಿರುತ್ತವೆ.

© NIMI NOT TO BE REPUBLISHED

15 **extends:** ಒಂದು ಕ್ಲಾಸ್ ಅನ್ನು ಇನ್ನೊಂದು ಕ್ಲಾಸ್‌ನಿಂದ ಅಥವಾ ಇಂಟರ್‌ಫೇಸ್‌ನಿಂದ ಪಡೆಯಲಾಗಿದೆ ಎಂಬುದನ್ನು ಸೂಚಿಸಲು ಜಾವಾ extends ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

16 **final:** ಒಂದು ವೇರಿಯೇಬಲ್ constant ಮೌಲ್ಯವನ್ನು ಹೊಂದಿದೆ ಎಂಬುದನ್ನು ಸೂಚಿಸಲು ಜಾವಾ final ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.ಇದನ್ನು ವೇರಿಯೇಬಲ್‌ನ ಜೊತೆಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಬಳಕೆದಾರರು ವೇರಿಯೇಬಲ್‌ನ ಮೌಲ್ಯವನ್ನು ನವೀಕರಿಸುವುದನ್ನು ತಡೆಯಲು ಇದನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

17 **finally:** ಜಾವಾ finally ಕೀವರ್ಡ್, try - catch ರಚನೆಯಲ್ಲಿ ಕೊಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಎಕ್ಸೆಪ್ಷನ್‌ಗಳನ್ನು ನಿಭಾಯಿಸಿದರೂ ಅಥವಾ ನಿಭಾಯಿಸದಿದ್ದರೂ ಈ ಬ್ಲಾಕ್ ಯಾವಾಗಲೂ ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತದೆ.

18 **float:** 32-bitfloating-point ಸಂಖ್ಯೆಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ float ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

19 **for:** for ಲೂಪನ್ನು ಪ್ರಾರಂಭಿಸಲು for ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಕೆಲವು ಕಂಡೀಷನ್‌ಗಳು true ಇದ್ದಾಗ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳು/ಫಂಕ್ಷನ್‌ಗಳ ಗುಂಪನ್ನು ಪುನರಾವರ್ತಿಸಲು for ಲೂಪನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪುನರಾವರ್ತನೆಯ ಸಂಖ್ಯೆಯು ಸ್ಥಿರವಾಗಿದ್ದಾಗ for ಲೂಪನ್ನು ಬಳಸುವುದನ್ನು ಶಿಫಾರಸ್ಸು ಮಾಡಲಾಗಿದೆ.

20 **if:** ಜಾವಾ if ಕೀವರ್ಡ್ ಕಂಡಿಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ. ಕಂಡಿಷನ್ true ಆಗಿದ್ದರೆ if ಬ್ಲಾಕ್ ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತದೆ.

21 **implements:** ಒಂದು ಇಂಟರ್‌ಫೇಸನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಲು ಜಾವಾ implements ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

22 **import:** ಜಾವಾ import ಕೀವರ್ಡ್, ಪ್ರಸ್ತುತ ಸೋರ್ಸ್ ಕೋಡ್‌ಗೆ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಲಭ್ಯವಾಗುವಂತೆ ಮತ್ತು ಪ್ರವೇಶಿಸುವಂತೆ ಮಾಡುತ್ತದೆ.

23 **instance of:** ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ನಿರ್ದಿಷ್ಟ ಕ್ಲಾಸ್‌ನ ನಿದರ್ಶನವಾಗಿದೆಯೆ ಅಥವಾ ಇಂಟರ್‌ಫೇಸನ್ನು ಅನುಷ್ಠಾನಗೊಳಿಸಿದೆಯೆ ಎಂದು ಪರೀಕ್ಷಿಸಲು ಜಾವಾ instance of ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

24 **int:** 32-bit signed integer ನ್ನು ಸಂಗ್ರಹಿಸುವ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ int ಕೀವರ್ಡನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

25 **interface:** ಇಂಟರ್‌ಫೇಸನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ interface ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

26 **long:** 64-bit integer ನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ long ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

27 **native:** JNI(JavaNativeInterface) ಉಪಯೋಗಿಸಿ ಒಂದು ಮೆಥೆಡ್‌ನ್ನು ಸ್ಥಳಿಯ ಕೋಡ್‌ಗೆ ಅನುಷ್ಠಾನಗೊಳಿಸುವುದನ್ನು ಸೂಚಿಸಲು ಜಾವಾ native ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

28 **new:** ಹೊಸ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ರಚಿಸಲು ಜಾವಾ new ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

29 **null:** ಉಲ್ಲೇಖವು ಯಾವುದನ್ನೂ ಉಲ್ಲೇಖಿಸುವುದಿಲ್ಲ ಎಂದು ಸೂಚಿಸಲು ಜಾವಾ null ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಗಾರ್ಬೇಜ್ ಮೌಲ್ಯವನ್ನು ತೆಗೆದುಹಾಕುತ್ತದೆ.

30 **package:** ಕ್ಲಾಸ್‌ಗಳನ್ನು ಒಳಗೊಂಡಿರುವ ಜಾವಾ ಪ್ಯಾಕೇಜ್‌ಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ package ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

31 **private:** ಜಾವಾ private ಕೀವರ್ಡ್ access modifier ಆಗಿದೆ. ಒಂದು ಮೆಥೆಡ್ ಅಥವಾ ವೇರಿಯೇಬಲ್‌ನ್ನು ಅವುಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲಾದ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಮಾತ್ರ ಪಡೆದುಕೊಳ್ಳಬಹುದು ಎಂಬುದನ್ನು ಇದು ಸೂಚಿಸುತ್ತದೆ.

32 **protected:** ಜಾವಾ protected ಕೀವರ್ಡ್ access modifier ಆಗಿದೆ. ಇದು ಪ್ಯಾಕೇಜ್ ಒಳಗೆ ಮತ್ತು ಹೊರಗೆ ಸುಲಭ ಪ್ರವೇಶ ಹೊಂದಿದೆ. ಆದರೆ inheritance ಮೂಲಕ ಮಾತ್ರ. ಇದನ್ನು ಕ್ಲಾಸ್‌ನ ಜೊತೆಗೆ ಅನ್ವಯಿಸಲಾಗುವುದಿಲ್ಲ.

33 **public:** ಜಾವಾ public ಕೀವರ್ಡ್ access modifier ಆಗಿದೆ. ಇದನ್ನು ಒಂದು ಐಟಂ ಅಥವಾ ಅಂಶವನ್ನು ಎಲಿ ್ಲಯಾದರೂ ಪಡೆದುಕೊಳ್ಳಬಹುದು ಎಂಬುದನ್ನು ಸೂಚಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

34 **return:** ಒಂದು ಮೆಥೆಡ್‌ನ ಕಾರ್ಯವು ಮುಕ್ತಾಯವಾದ ನಂತರ ಅಲ್ಲಿಂದ ಹಿಂತಿರುಗಲು ಜಾವಾ return ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

35 **short:** 16-bit integer ಸಂಖ್ಯೆಯನ್ನು ಸಂಗ್ರಹಿಸಬಲ್ಲ ವೇರಿಯೇಬಲ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ short ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

36 **static:** ಒಂದು ವೇರಿಯೇಬಲ್ ಅಥವಾ ಮೆಥೆಡ್, ಕ್ಲಾಸ್ ಮೆಥೆಡ್ ಆಗಿದೆ ಎಂದು ಸೂಚಿಸಲು ಜಾವಾ static ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಜಾವಾದಲ್ಲಿ static ಕೀವರ್ಡ್‌ನ್ನು ಮುಖ್ಯವಾಗಿ ಮೆಮೊರಿ ನಿರ್ವಹಣೆಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

37 **strictfp:** ಪೋರ್ಟೇಬಲಿಟಿಯನ್ನು ಖಚಿತಪಡಿಸುವುದಕ್ಕಾಗಿ floating point ಲೆಕ್ಕಗಳನ್ನು ನಿರ್ಬಂಧಿಸಲು ಜಾವಾ strictfp ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

38 **super:** ಜಾವ super ಕೀವರ್ಡ್, ಪೇರೆಂಟ್ ಕ್ಲಾಸ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಉಲ್ಲೇಖಿಸಲು ಬಳಸುವ ವೇರಿಯೇಬಲ್ ರೆಫರನ್ಸ್ ಆಗಿದೆ. ಇದನ್ನು immediate parent class ಮೆಥೆಡ್‌ನ್ನು ಇನ್ವೋಕ್ ಮಾಡಲು ಉಪಯೋಗಿಸಬಹುದು.

39 **switch:** ಜಾವಾ switch ಕೀವರ್ಡ್ switch ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಹೊಂದಿದೆ. ಇದು test ಮೌಲ್ಯದ ಆಧಾರದ ಮೇಲೆ ಕೋಡ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ. switch ಸ್ಟೇಟಮೆಂಟ್ ವೇರಿಯೇಬಲ್‌ನ ಸಮಾನತೆಯನ್ನು ಅನೇಕ ಮೌಲ್ಯಗಳೊಂದಿಗೆ ಪರೀಕ್ಷಿಸುತ್ತದೆ.

40 **synchronized:** ಮಲ್ಟಿ ಥ್ರೇಡೆಡ್ ಕೋಡ್‌ನಲ್ಲಿ ನಿರ್ಣಾಯಕ ಸೆಕ್ಷನ್‌ಗಳನ್ನು ಅಥವಾ ಮೆಥೆಡ್‌ಗಳನ್ನು ಸೂಚಿಸಲು ಜಾವಾ synchronized ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

41 **this:** ಒಂದು ಮೆಥೆಡ್ ಅಥವಾ ಕನ್‌ಸ್ಟ್ರಕ್ಟರ್‌ನಲ್ಲಿ ಪ್ರಸ್ತುತ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಉಲ್ಲೇಖಿಸಲು ಜಾವಾದ this ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

42 **throw:** ಎಕ್ಸೆಪ್ಷನ್‌ನ್ನು ಎಕ್ಸ್‌ಪ್ಲಿಸಿಟ್ ಆಗಿ ಥ್ರೋ ಮಾಡಲು ಜಾವಾದ throw ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. throw ಕೀವರ್ಡ್‌ನ್ನು ಮುಖ್ಯವಾಗಿ ಕಸ್ಟಮ್ ಎಕ್ಸೆಪ್ಷನ್‌ಗಳನ್ನು ಥ್ರೋ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಒಂದು ನಿದರ್ಶನವನ್ನು ಅನುಸರಿಸುತ್ತದೆ.

43 **throws:** ಎಕ್ಸೆಪ್ಷನ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ಜಾವಾ throws ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪರಿಶೀಲಿಸಿದ ಎಕ್ಸೆಪ್ಷನ್‌ಗಳನ್ನು ಥ್ರೋಗಳೊಂದಿಗೆ ಪ್ರಸಾರ ಮಾಡಬಹುದು.

44 **transient:** ಜಾವಾ transient ಕೀವರ್ಡ್‌ನ್ನು ಸಿರಿಯಲೈಸೇಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ನೀವು ಯಾವುದೇ ಡಾಟಾ ಮೆಂಬರ್‌ನ್ನು transient ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಿದರೆ, ಅದು ಸಿರಿಯಲೈಸ್ ಆಗುವುದಿಲ್ಲ.

45 **try:** ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಎಕ್ಸೆಪ್ಷನ್‌ಗಳಿಗಾಗಿ ಪರೀಕ್ಷಿಸಲು ಪ್ರಾರಂಭಿಸುವುದಕ್ಕಾಗಿ ಜಾವಾ try ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. try ಬ್ಲಾಕನ್ನು catch ಅಥವಾ finally ಬ್ಲಾಕ್‌ಗಳು ಅನುಸರಿಸುತ್ತವೆ.

46 **void:** ಒಂದು ಮೆಥೆಡ್ ರಿಟರ್ನ್ ಮೌಲ್ಯವನ್ನು ಹೊಂದಿಲ್ಲ ಎಂದು ಸೂಚಿಸಲು ಜಾವಾ void ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

47 **volatile:** ಒಂದು ವೇರಿಯೆಬಲ್ ಅಸಮಕಾಲಿಕವಾಗಿ(asynchronously) ಬದಲಾಗಬಹುದು ಎಂದು ಸೂಚಿಸಲು ಜಾವಾ volatile ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

48 **while :** while ಲೂಪನ್ನು ಪ್ರಾರಂಭಿಸಲು ಜಾವಾ while ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಈ ಲೂಪ್ ಪ್ರೋಗ್ರಾಂನ ಒಂದು ಭಾಗವನ್ನು ಹಲವಾರು ಬಾರಿ ಪುನರಾವರ್ತಿಸುತ್ತದೆ. ಪುನರಾವರ್ತನೆಗೊಳ್ಳುವ ಸಂಖ್ಯೆ ಸ್ಥಿರವಾಗಿಲ್ಲದಿದ್ದರೆ, while ಲೂಪ್ ಬಳಸುವುದನ್ನು ಶಿಫಾರಸು ಮಾಡಲಾಗುತ್ತದೆ.

ಜಾವಾದಲ್ಲಿ ಡಾಟಾಟೈಪ್‌ಗಳು (Datatypes in Java)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- **ಡಾಟಾಟೈಪ್‌ನ ವಿಧಗಳು**
- **ಜಾವಾ ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು**
- **ನಾನ್ - ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು**

ಜಾವಾದಲ್ಲಿ ಡಾಟಾಟೈಪ್‌ಗಳು

ಒಂದು ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸಬಹುದಾದ ವಿಭಿನ್ನ ಗಾತ್ರಗಳು ಮತ್ತು ಮೌಲ್ಯಗಳನ್ನು ಡಾಟಾಟೈಪ್‌ಗಳು ಸೂಚಿಸುತ್ತವೆ. ಜಾವಾದಲ್ಲಿ ಎರಡು ರೀತಿಯ ಡಾಟಾಟೈಪ್‌ಗಳಿವೆ:

ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು : ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು ಇವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ - boolean, char, byte, short, int, long, float ಮತ್ತು double.

ನಾನ್-ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು : ನಾನ್-ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು ಇವುಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತವೆ - Classes, Interfaces, ಮತ್ತು Arrays.

ಜಾವಾ ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು (JavaPrimitiveDataTypes)

ಜಾವಾ ಭಾಷೆಯಲ್ಲಿ, ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳು ಡಾಟಾವನ್ನು ಕುಶಲತೆಯಿಂದ ನಿರ್ವಹಿಸುವ ಮೂಲ ಘಟಕಗಳಾಗಿವೆ. ಇವು ಜಾವಾ ಭಾಷೆಯಲ್ಲಿ ಲಭ್ಯವಿರುವ ಅತ್ಯಂತ ಮೂಲ ಡಾಟಾಟೈಪ್‌ಗಳಾಗಿವೆ.

ಗಮನಿಸಿ : ಜಾವಾ statically-typed **ಪ್ರೋಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಾಗಿದೆ. ಇದರ ಅರ್ಥ ಎಲ್ಲಾ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಅವುಗಳನ್ನು ಬಳಸುವ**

NOT TO BE REPUBLISHED ©NIMI

ಮೊದಲು ಡಿಕ್ಲೇರ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ನಾವು ವೇರಿಯೇಬಲ್‌ನ ಟೈಪ್ ಮತ್ತು ಹೆಸರನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುವ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ.

8 ರೀತಿಯ ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳಿವೆ :

1 boolean datatype (ಬೂಲೀನ್ ಡಾಟಾಟೈಪ್)

2 byte datatype (ಬೈಟ್ ಡಾಟಾಟೈಪ್)

3 char datatype (ಕ್ಯಾರ್ ಡಾಟಾಟೈಪ್)

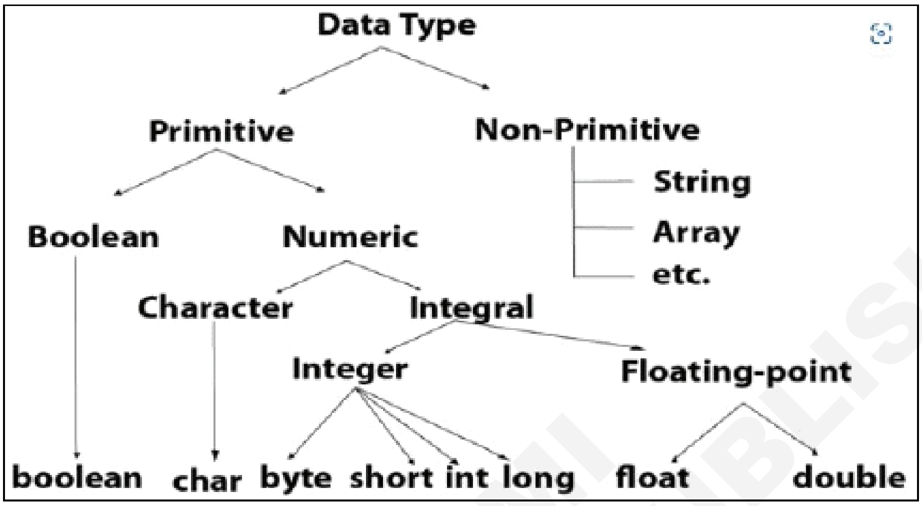
4 short datatype (ಶಾರ್ಟ್ ಡಾಟಾಟೈಪ್)

5 int datatype (ಇಂಟ್ ಡಾಟಾಟೈಪ್)

6 longdatatype (ಲಾಂಗ್ ಡಾಟಾಟೈಪ್)

7 float datatype (ಫ್ಲೋಟ್ ಡಾಟಾಟೈಪ್)

8 double datatype (ಡಬಲ್ ಡಾಟಾಟೈಪ್)



ಡಾಟಾಟೈಪ್	ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯ	ಡೀಫಾಲ್ಟ್ ಗಾತ್ರ
boolean	false	1bit
char	'\u0000'	2byte
byte	0	1byte
short	0	2byte
int	0	4byte
long	0L	8byte
float	0.0f	4byte
double	0.0d	8byte

ಜಾವಾ ಡಾಟಾಟೈಪ್‌ಗಳು (Java DataTypes)

ಬೂಲೀನ್ ಡಾಟಾಟೈಪ್ (BooleanDataType)

ಬೂಲೀನ್ ಡಾಟಾಟೈಪ್‌ನ್ನು ಕೇವಲ ಎರಡು ಮೌಲ್ಯಗಳನ್ನು ಮಾತ್ರ ಸಂಗ್ರಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ: True ಮತ್ತು False. ಈ ಡಾಟಾಟೈಪನ್ನು true / false ಕಂಡೀಷನ್‌ಗಳನ್ನು ಸೂಚಿಸುವ ಸರಳ ಫ್ಲ್ಯಾಗ್‌ಗಳಿಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

boolean ಡಾಟಾಟೈಪ್ ಒಂದು ಬಿಟ್ ಮಾಹಿತಿಯನ್ನು ಸೂಚಿಸುತ್ತದೆ, ಆದರೆ, ಅದರ "ಗಾತ್ರ"ವನ್ನು ನಿಖರವಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ.

ಉದಾಹರಣೆ :

Booleanone =false

ಬೈಟ್ ಡಾಟಾಟೈಪ್ (Byte DataType)

ಬೈಟ್ ಡಾಟಾಟೈಪ್, ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ನ ಒಂದು ಉದಾಹರಣೆಯಾಗಿದೆ. ಇದು 8-bit signed two's complement ಪೂರ್ಣಾಂಕವಾಗಿದೆ. ಇದರ ಮೌಲ್ಯ- ವ್ಯಾಪ್ತಿಯು -128 ರಿಂದ 127 ರವರೆಗೆ (ಒಳಗೊಂಡಿರುತ್ತದೆ). ಇದರ ಕನಿಷ್ಟ ಮೌಲ್ಯ -128 ಆಗಿರುತ್ತದೆ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯ 127 ಆಗಿರುತ್ತದೆ. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯ 0 ಆಗಿರುತ್ತದೆ.

ದೊಡ್ಡ ಅರೇಗಳಲ್ಲಿ ಮೆಮೊರಿಯನ್ನು ಉಳಿತಾಯ ಮಾಡುವುದು ಹೆಚ್ಚು ಅವಶ್ಯಕವಿರುತ್ತದೆ. ಇಲ್ಲಿ ಮೆಮೊರಿಯನ್ನು ಉಳಿತಾಯ ಮಾಡಲು ಬೈಟ್ ಡಾಟಾಟೈಪ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಜಾಗವನ್ನು ಉಳಿಸುತ್ತದೆ ಏಕೆಂದರೆ ಒಂದು ಬೈಟ್ ಒಂದು ಪೂರ್ಣಾಂಕಕ್ಕಿಂತ 4 ಪಟ್ಟು ಚಿಕ್ಕದಾಗಿರುತ್ತದೆ.ಇದನ್ನು "int" ಡಾಟಾಟೈಪ್‌ನ ಬದಲಿಗೆ ಸಹ ಉಪಯೋಗಿಸಬಹುದು.

ಉದಾಹರಣೆ :

byte a = 10,byteb= -20

© NIMI NOT TO BE REPUBLISHED

ಶಾರ್ಟ್ ಡಾಟಾಟೈಪ್ (Short DataType)

ಶಾರ್ಟ್ ಡಾಟಾಟೈಪ್ 16-bit signed two's complement integer (ಪೂರ್ಣಾಂಕ) ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯವು -32,768 ದಿಂದ 32,767 (ಒಳಗೊಂಡಂತೆ) ರವರೆಗೆ ಇರುತ್ತದೆ. ಇದರ ಕನಿಷ್ಟ ಮೌಲ್ಯ-ವ್ಯಾಪ್ತಿಯು -32,768 ಆಗಿರುತ್ತದೆ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯವು 32,767 ಆಗಿರುತ್ತದೆ. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯವು 0 ಆಗಿರುತ್ತದೆ.

ಬೈಟ್ ಡಾಟಾಟೈಪ್‌ನಂತೆಯೆ ಶಾರ್ಟ್ ಡಾಟಾಟೈಪನ್ನೂ ಸಹ ಮೆಮೊರಿಯನ್ನು ಉಳಿಸಲು ಉಪಯೋಗಿಸಬಹುದು. ಶಾರ್ಟ್ ಡಾಟಾಟೈಪ್ ಇಂಟಿಜರ್ ಡಾಟಾಟೈಪ್‌ಗಿಂತ 2 ಪಟ್ಟು ಚಿಕ್ಕದಾಗಿರುತ್ತದೆ

ಉದಾಹರಣೆ :

short s =10000,short r=-5000

int ಡಾಟಾಟೈಪ್ (ntDataType)

int ಡಾಟಾಟೈಪ್, 32-bit signed two's complement integer (ಪೂರ್ಣಾಂಕ) ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯ-ವ್ಯಾಪ್ತಿಯು -2,147,483,648 (-2^31) ದಿಂದ 2,147,483,647 (2^31 -1) (ಒಳಗೊಂಡಂತೆ). ರವರೆಗೆ ಇರುತ್ತದೆ. ಇದರ ಕನಿಷ್ಟ ಮೌಲ್ಯವು - 2,147,483,648 ಇರುತ್ತದೆ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯವು 2,147,483,647 ಇರುತ್ತದೆ. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯವು 0 ಆಗಿರುತ್ತದೆ.

ಮೆಮೊರಿಯ ಬಗ್ಗೆ ಯಾವುದೇ ಸಮಸ್ಯೆ ಇಲ್ಲದಿದ್ದರೆ, int ಡಾಟಾಟೈಪ್‌ನ್ನು ಸಾಮಾನ್ಯವಾಗಿ ಅವಿಭಾಜ್ಯ ಮೌಲ್ಯಗಳಿಗೆ ಡೀಫಾಲ್ಟ್ ಡಾಟಾಟೈಪ್ ಆಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ :

int a= 100000,int b =-200000

Long ಡಾಟಾಟೈಪ್(LongDataType)

Long ಡಾಟಾಟೈಪ್, 64-bit two's complement integer (ಪೂರ್ಣಾಂಕ) ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯ- ವ್ಯಾಪ್ತಿಯು -9,223,372,036,854,775,808 (-2^63) ದಿಂದ 9,223,372,036,854,775,807(2^63-1) (ಒಳಗೊಂಡಂತೆ) ರವರೆಗೆ ಇರುತ್ತದೆ. .ಇದರ ಕನಿಷ್ಟ ಮೌಲ್ಯವು -9,223,372,036,854,775,808 ಆಗಿರುತ್ತದೆ ಮತ್ತು ಗರಿಷ್ಟ ಮೌಲ್ಯವು 9,223,372,036,854,775,807 ಆಗಿರುತ್ತದೆ. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯವು 0 ಆಗಿರುತ್ತದೆ. int ಡಾಟಾಟೈಪ್ ಒದಗಿಸುವ ಮೌಲ್ಯದ ವ್ಯಾಪ್ತಿಗಿಂತಲೂ ಹೆಚ್ಚು ವ್ಯಾಪ್ತಿಯ ಅಗತ್ಯವಿದ್ದಾಗ long ಡಾಟಾಟೈಪ್‌ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ :

long a=100000L,long b=-200000L

Float ಡಾಟಾಟೈಪ್ (Float DataType)

Float ಡಾಟಾಟೈಪ್, single-precision 32-bit IEEE 754 floating point ಸಂಖ್ಯೆ ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯದ ವ್ಯಾಪ್ತಿಯು ಅಪರಿಮಿತವಾಗಿದೆ. floating point ಸಂಖ್ಯೆಗಳ ದೊಡ್ಡ ಅರೇಗಳಲ್ಲಿ ನೀವು ಮೆಮೊರಿಯನ್ನು ಉಳಿತಾಯ ಮಾಡಲು ಬಯಸಿದರೆ, float (double ಬದಲಾಗಿ) ಡಾಟಾಟೈಪನ್ನು ಉಪಯೋಗಿಸಲು ಶಿಫಾರಸು ಮಾಡಲಾಗುತ್ತದೆ. float ಡಾಟಾಟೈಪನ್ನು ಕರೆನ್ಸಿಯಂತ 'ಹ ನಿಖರವಾದ ಮೌಲ್ಯಗಳಿಗೆ ಎಂದಿಗೂ ಬಳಸಬಾರದು. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯ 0.0F ಆಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ :

float f1 =234.5f

Double ಡಾಟಾಟೈಪ್ (Double DataType)

Double ಡಾಟಾಟೈಪ್, double-precision 64-bit IEEE 754 floating point ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯದ ವ್ಯಾಪ್ತಿಯು ಅಪರಿಮಿತವಾಗಿದೆ. Double ಡಾಟಾಟೈಪ್‌ನ್ನು ಸಾಮಾನ್ಯವಾಗಿ float ನಂತಹ ದಶಮಾಂಶ ಮೌಲ್ಯಗಳಿಗೆ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. Double ಡಾಟಾಟೈಪನ್ನು ಸಹ ಕರೆನ್ಸಿಯಂತಹ ನಿಖರವಾದ ಮೌಲ್ಯಗಳಿಗೆ ಎಂದಿಗೂ ಬಳಸಬಾರದು. ಇದರ ಡೀಫಾಲ್ಟ್ ಮೌಲ್ಯ 0.0d. ಆಗಿರುತ್ತದೆ.

ಉದಾಹರಣೆ :

double d1=12.3

Char ಡಾಟಾಟೈಪ್ (CharDataType)

Char ಡಾಟಾಟೈಪ್, single 16-bit Unicode character ಆಗಿದೆ. ಇದರ ಮೌಲ್ಯದ ವ್ಯಾಪ್ತಿಯು '\u0000' (ಅಥವಾ 0) ದಿಂದ '\uffff' (ಅಥವಾ 65,535 ನ್ನೂ ಒಳಗೊಂಡಂತೆ) ವರೆಗೆ ಇರುತ್ತದೆ. Char ಡಾಟಾಟೈಪ್‌ನ್ನು ಅಕ್ಷರಗಳನ್ನು(characters.) ಸಂಗ್ರಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ :

charletterA='A'

ಜಾವಾದಲ್ಲಿ char 2 ಬೈಟ್‌ಗಳನ್ನು ಏಕೆ ಬಳಸುತ್ತದೆ ಮತ್ತು \u0000 ಎಂದರೇನು?

ಏಕೆಂದರೆ ಜಾವಾ ಯುನಿಕೋಡ್ ಸಿಸ್ಟಮ್‌ನ್ನು ಬಳಸುತ್ತದೆ ಮತ್ತು ASCII ಕೊಡ್ ಸಿಸ್ಟಮ್ ಬಳಸುವುದಿಲ್ಲ. \u0000 ಇದು ಯುನಿಕೋಡ್ ಸಿಸ್ಟಮ್‌ನ ಅತ್ಯಂತ ಕನಿಷ್ಟ ರೇಂಜ್ ಆಗಿದೆ. ಯುನಿಕೋಡ್ ಬಗ್ಗೆ ಹೆಚ್ಚಿನ ವಿಸ್ತಾರವಾದ ವಿವರವನ್ನು ಪಡೆಯಲು ಮುಂದಿನ ಪುಟವನ್ನು ನೋಡಿರಿ.

ಜಾವಾದಲ್ಲಿ ಆಪರೇಟರ್‌ಗಳು (Operators in Java)

ಜಾವಾದಲ್ಲಿ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸುವ ಚಿನ್ಹೆಗಳಿಗೆ ಆಪರೇಟರ್‌ಗಳು ಎನ್ನುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ : +, -, *, / ಇತ್ಯಾದಿ. ಜಾವಾದಲ್ಲಿ ಅನೇಕ ರೀತಿಯ ಆಪರೇಟರ್‌ಗಳಿವೆ. ಅವುಗಳನ್ನು ಕೆಳಗೆ ನೀಡಲಾಗಿದೆ.

- Unary ಆಪರೇಟರ್
- Arithmetic ಆಪರೇಟರ್,
- Shift ಆಪರೇಟರ್,
- Relational ಆಪರೇಟರ್,
- Bitwise ಆಪರೇಟರ್,
- Logical ಆಪರೇಟರ್,
- Ternary ಆಪರೇಟರ್ ಮತ್ತು
- Assignment ಆಪರೇಟರ್

ಜಾವಾ ಆಪರೇಟರ್‌ಗಳ ಆದ್ಯತೆ (Java Operator Precedence)

ಆಪರೇಟರ್ ವಿಧ (OperatorType)	ವರ್ಗ (Category)	ಆದ್ಯತೆ (Precedence)
Unary	postfix	expr++expr--
prefix	++expr--expr+expr-expr ~!	
Arithmetic	multiplicative	* /%
additive	+-	
Shift	shift	<<>>>>>
Relational	comparison	<><=>= instanceof
equality	== !=	
Bitwise	bitwiseAND	&
bitwiseexclusiveOR	^	
bitwiseinclusiveOR	|	
Logical	logicalAND	&&
logicalOR	||	
Ternary	ternary	? :
Assignment	assignment	=+=-= *=/=%=&=^=|= <<= >>=>>>=

ಜಾವಾ ಯುನರಿ ಆಪರೇಟರ್ (Java Unary Operator)

ಜಾವಾ ಯುನರಿ ಆಪರೇಟರ್‌ಗಳಿಗೆ ಕೇವಲ ಒಂದು ಆಪರಂಡ್‌ನ ಅವಶ್ಯಕತೆ ಇರುತ್ತದೆ. ಯುನರಿ ಆಪರೇಟರ್‌ಗಳನ್ನು ವಿವಿಧ ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಅವು :

- ಒಂದು ಮೌಲ್ಯವನ್ನು ಹೆಚ್ಚಿಸುವುದು / ಕಡಿಮೆಗೊಳಿಸುವುದು (incrementing/decrementing)
- ಒಂದು ಎಕ್ಸ್‌ಪ್ರೆಷನ್‌ನ್ನು ಋಣಾತ್ಮಕವಾಗಿಸುವುದು(negating an expression)
- ಬೂಲಿನ್ ಮೌಲ್ಯವನ್ನು ತಿರುಗಿಸುವುದು(inverting the value of a boolean)

ಜಾವಾ ಯುನರಿ ಆಪರೇಟರ್ ಉದಾಹರಣೆ :++ ಮತ್ತು --

```
public class OperatorExample{
public static void main(String args[]){
int x=10;
System.out.println(x++);//10 (11)
System.out.println(++x);//12
System.out.println(x--);//12 (11)
System.out.println(--x);//10
}}
```

© NIMI NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ :

10

12

12

10

ಜಾವಾ ಯುನರಿ ಆಪರೇಟರ್ ಉದಾಹರಣೆ 2 : ++ ಮತ್ತು --

```
publicclassOperatorExample{
public static void main(String args[]){inta=10;
intb=10;
System.out.println(a++ + ++a);//10+12=22
System.out.println(b+++b++);//10+11=21
}}
```

ಫಲಿತಾಂಶ :

22

21

ಜಾವಾ ಯುನರಿ ಆಪರೇಟರ್ ಉದಾಹರಣೆ :~ ಮತ್ತು !

```
publicclassOperatorExample{
public static void main(String args[]){inta=10;
int b=-10;
boolean c=true;
booleand=false;
System.out.println(~a);//-11 (minus of total positive value which starts from 0)
System.out.println(~b);//9 (positive of total minus, positive starts from 0)
System.out.println(!c);//false (opposite of boolean value)
System.out.println(!d);//true
}}
```

ಫಲಿತಾಂಶ :

-11

9

false

true

ಜಾವಾ ಅಂಕಗಣಿತ ಆಪರೇಟರ್‌ಗಳು (Java Arithmetic Operators)

ಜಾವಾ arithmetic ಆಪರೇಟರ್‌ಗಳನ್ನು ಸಂಕಲನ(addition), ವ್ಯವಕಲನ (subtraction), ಗುಣಾಕಾರ(multiplication), ಮತ್ತು ಭಾಗಾಕಾರ (division) ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇವು ಸಾಮಾನ್ಯ ಗಣಿತದ ಲೆಕ್ಕಾಚಾರಗಳನ್ನು ಮಾಡುತ್ತವೆ.

ಜಾವಾ Arithmetic ಆಪರೇಟರ್ ಉದಾಹರಣೆ

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=5;
System.out.println(a+b);//15
System.out.println(a-b);//5
System.out.println(a*b);//50
System.out.println(a/b);//2
System.out.println(a%b);//0
}}
```

ಫಲಿತಾಂಶ :

15

5

50

2

0

Java Arithmetic Operator ಉದಾಹರಣೆ : Expression

```
public class OperatorExample{
public static void main(Stringargs[]){
System.out.println(10*10/5+3-1*4/2);
}}
```

ಫಲಿತಾಂಶ :

21

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾ Left Shift ಆಪರೇಟರ್ (Java Left Shift Operator)

ಜಾವಾ left shift ಆಪರೇಟರ್ << ಅನ್ನು ಒಂದು ಸಂಖ್ಯೆಯಲ್ಲಿರುವ ಎಲ್ಲ ಬಿಟ್‌ಗಳನ್ನು ಸೂಚಿಸಿದ ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಅನುಸಾರ ಎಡಭಾಗಕ್ಕೆ ವರ್ಗಾಯಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾ Left Shift ಆಪರೇಟರ್ ಉದಾಹರಣೆ :

```
public class OperatorExample{
public static void main(Stringargs[]){
System.out.println(10<<2);//10*2^2=10*4=40
System.out.println(10<<3);//10*2^3=10*8=80
System.out.println(20<<2);//20*2^2=20*4=80
System.out.println(15<<4);//15*2^4=15*16=240
}}
```

ಫಲಿತಾಂಶ :

```
40
80
80
240
```

ಜಾವಾ Right Shift ಆಪರೇಟರ್ (Java Right Shift Operator)

ಜಾವಾ right shift ಆಪರೇಟರ್‌ನ್ನು >> ಎಡ ಆಪರಂಡ್‌ನ ಸಂಖ್ಯೆಯನ್ನು ಬಲ ಆಪರಂಡ್‌ನಿಂದ ಸೂಚಿಸಲಾದ ನಿರ್ದಿಷ್ಟ ಸಂಖ್ಯೆಯ ಬಿಟ್‌ಗಳನ್ನು ಬಲ ಭಾಗಕ್ಕೆ ವರ್ಗಾಯಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾ Right Shift ಆಪರೇಟರ್ ಉದಾಹರಣೆ :

```
public OperatorExample{
public static void main(String args[]){
System.out.println(10>>2);//10/2^2=10/4=2
System.out.println(20>>2);//20/2^2=20/4=5
System.out.println(20>>3);//20/2^3=20/8=2
}}
```

ಫಲಿತಾಂಶ :

```
2
5
2
```

ಜಾವಾ Shift ಆಪರೇಟರ್ ಉದಾಹರಣೆ (Java Shift Operator Example): >> vs >>>

```
publicclass OperatorExample{
public static void main(Stringargs[]){
//For positive number, >> and >>> works same
System.out.println(20>>2);
System.out.println(20>>>2);
//For negative number, >>> changes parity bit (MSB) to 0
System.out.println(-20>>2);
System.out.println(-20>>>2);
}}
```

ಫಲಿತಾಂಶ :

```
5
5
-5
1073741819
```

ಜಾವಾ AND ಆಪರೇಟರ್ ಉದಾಹರಣೆ (Java AND Operator Example): Logical && ಮತ್ತು Bitwise &

ಮೊದಲನೇ ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ ಲಾಜಿಕಲ್ && ಆಪರೇಟರ್ ಎರಡನೇ ಕಂಡಿಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸುವುದಿಲ್ಲ. ಇದು ಮೊದಲನೇ ಕಂಡಿಷನ್ true ಇದ್ದರೆ ಮಾತ್ರ ಎರಡನೇ ಕಂಡಿಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ.

ಮೊದಲನೇ ಕಂಡಿಷನ್ true ಇರಲಿ ಅಥವಾ false ಇರಲಿ, ಬಿಟ್‌ವೈಸ್ & ಆಪರೇಟರ್ ಯಾವಾಗಲೂ ಎರಡೂ ಕಂಡೀಷನ್‌ಗಳನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ.

```
public class Operator Example{
public static void main(Stringargs[]){
int a=10;int b=5;int c=20;
System.out.println(a<b&&a<c);//false && true = false
System.out.println(a<b&a<c);//false&true=false
}}
```

ಫಲಿತಾಂಶ :

```
false
false
```


© NIMI NOT TO BE REPUBLISHED

Java AND Operator ಉದಾಹರಣೆ : Logical && vs Bitwise&

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=5;
int c=20;
System.out.println(a<b&&a++<c);//false && true = false
System.out.println(a);//10 because second condition is not checked
System.out.println(a<b&a++<c);//false && true = false
System.out.println(a);//11 because second condition is checked
}}
```

ಫಲಿತಾಂಶ :

```
false10
false11
```

ಜಾವಾ OR ಆಪರೇಟರ್ ಉದಾಹರಣೆ (Java OR Operator Example): Logical || ಮತ್ತು Bitwise |

ಮೊದಲನೇ ಕಂಡಿಷನ್ true ಇದ್ದರೆ, ಜಾವಾ logical || ಆಪರೇಟರ್ ಎರಡನೇ ಕಂಡೀಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸುವುದಿಲ್ಲ. ಮೊದಲನೆ ಕಂಡಿಷನ್ false ಇದ್ದರೆ ಮಾತ್ರ ಇದು ಎರಡನೇ ಕಂಡಿಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ.

ಮೊದಲನೇ ಕಂಡಿಷನ್ true ಇರಲಿ ಅಥವಾ false ಇರಲಿ, bitwise | ಆಪರೇಟರ್ ಯಾವಾಗಲೂ ಎರಡೂ ಕಂಡಿಷನ್‌ಗಳನ್ನು ಪರೀಕ್ಷಿಸುತ್ತದೆ.

```
public class Operator Example{
public static void main(String args[])
{
int a=10;
int b=5;
int c=20;
System.out.println(a>b||a<c);//true || true = true
System.out.println(a>b|a<c);//true|true=true
//|| vs|
System.out.println(a>b||a++<c);//true || true = true
System.out.println(a);//10 because second condition is not checked
System.out.println(a>b|a++<c);//true | true = true
System.out.println(a);//11because second condition is checked
}}
```

ಫಲಿತಾಂಶ :

```
true
true
true
10
true
11
```

ಜಾವಾ Ternary ಆಪರೇಟರ್ (Java Ternary Operator)

ಜಾವಾ Ternary ಆಪರೇಟರ್‌ನ್ನು if - then - else ಸ್ಟೇಟಮೆಂಟ್‌ನ ಒಂದು ಸಾಲಿನ ಬದಲಾವಣೆಯ ಸ್ಟೇಟಮೆಂಟ್ ಆಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ ಬಹಳಷ್ಟು ಬಾರಿ ಉಪಯೋಗಿಸಲಾಗಿದೆ. ಇದು ಮೂರು ಆಪರಂಡ್‌ಗಳನ್ನು ತೆಗೆದುಕೊಳ್ಳುವ ಏಕೈಕ ಕಂಡೀಷನಲ್ ಆಪರೇಟರ್ ಆಗಿದೆ.

ಜಾವಾ Ternary ಆಪರೇಟರ್ ಉದಾಹರಣೆ

```
public class Operator Example {
public static void main (String args[]){inta=2;
int b=5;
int min=(a<b)?a:b;
System.out.println(min);
}}
```

ಫಲಿತಾಂಶ :

```
2
```

ಇನ್ನೊಂದು ಉದಾಹರಣೆ :

```
public class Operator Example{
public static void main(String args[]){
int a=10;
int b=5;
```

NOT TO BE REPUBLISHED © NIMI

```
int min=(a<b)?a:b;System.out.println(min);
}}
```

ಫಲಿತಾಂಶ :

5

ಜಾವಾ Assignment ಆಪರೇಟರ್ (Java Assignment Operator)

ಜಾವಾ Assignment ಆಪರೇಟರ್, ಅತ್ಯಂತ ಸಾಮಾನ್ಯವಾಗಿ ಉಪಯೋಗಿಸಲ್ಪಡುವ ಆಪರೇಟರ್‌ಗಳಲ್ಲಿ ಒಂದಾಗಿದೆ. ಇದನ್ನು ಇದರ ಬಲಭಾಗದಲ್ಲಿರುವ ಮೌಲ್ಯವನ್ನು ಇದರ ಎಡಭಾಗದಲ್ಲಿರುವ ಆಪರಂಡ್‌ಗೆ ನಿಯೋಜಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾ Assignment ಆಪರೇಟರ್ ಉದಾಹರಣೆ

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=20;
a+=4;//a=a+4 (a=10+4)
b-=4;//b=b-4 (b=20-4)
System.out.println(a);
System.out.println(b);
}}
```

ಫಲಿತಾಂಶ :

14

16

ಜಾವಾ Assignment ಆಪರೇಟರ್ ಉದಾಹರಣೆ

```
public class OperatorExample{
public static void main(String[] args){
int  a=10;
a+=3;//10+3
System.out.println(a);
a-=4;//13-4
System.out.println(a);
a*=2;//9*2
System.out.println(a);
a/=2;//18/2
System.out.println(a);
}}
```

ಫಲಿತಾಂಶ :

13

9

18

9

ಜಾವಾ Assignment ಆಪರೇಟರ್ ಉದಾಹರಣೆ :Adding short

```
public class OperatorExample{
public static void main(String args[]){
short a=10;
short b=10;
//a+=b;//a=a+b internally so fine
a=a+b;//Compile time error because 10+10=20 now int
System.out.println(a);
}}
```

ಫಲಿತಾಂಶ :

Compile timeerror

After typecast:

```
public class OperatorExample
{
public static void main(String args[])
{
short a=10;
short b=10;
a=(short)(a+b);//20 which is int now converted to short
System.out.println(a);
}
}
```

ಫಲಿತಾಂಶ :

20

ಜಾವಾ I/O ಸ್ಟ್ರೀಮ್‌ಗಳು (Java I/O Streams)

ಈ ಟ್ಯುಟೋರಿಯಲ್‌ನಲ್ಲಿ, ನಾವು ಜಾವಾ input/output ಸ್ಟ್ರೀಮ್‌ಗಳು ಮತ್ತು ಅವುಗಳ ವಿಧಗಳ ಬಗ್ಗೆ ಕಲಿಯುತ್ತೇವೆ.

ಜಾವಾದಲ್ಲಿ, ಸ್ಟ್ರೀಮ್‌ಗಳು ಎಂದರೆ ಡಾಟಾದ ಅನುಕ್ರಮವಾಗಿದೆ. ಇವುಗಳನ್ನು ಸೋರ್ಸ್‌ನಿಂದ ಓದಲಾಗುತ್ತದೆ ಮತ್ತು ಡೆಸ್ಟಿನೇಷನ್‌ನಲ್ಲಿ ಬರೆಯಲಾಗುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಡಾಟಾವನ್ನು ಸೋರ್ಸ್‌ನಿಂದ ಓದಲು input ಸ್ಟ್ರೀಮನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಮತ್ತು ಡಾಟಾವನ್ನು ಡೆಸ್ಟಿನೇಷನ್‌ನಲ್ಲಿ ಬರೆಯಲು output ಸ್ಟ್ರೀಮನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ,

```
classHelloWorld{
publicstaticvoidmain(String[] args){
        System.out.println("Hello, World!");
    }
}
Run Code
```

ಉದಾಹರಣೆಗೆ, ನಮ್ಮ ಮೊದಲ Hello World ಉದಾಹರಣೆಯಲ್ಲಿ, ಒಂದು ಸ್ಟ್ರಿಂಗ್‌ನ್ನು ಪ್ರಿಂಟ್ ಮಾಡಲು System.out ಅನ್ನು ಉಪಯೋಗಿಸಿದ್ದೇವೆ. ಇಲ್ಲಿ System.out ಇದು output ಸ್ಟ್ರೀಮ್‌ನ ಒಂದು ವಿಧವಾಗಿದೆ.

ಅದೇ ರೀತಿ ಇನ್ಪುಟ್ ಅನ್ನು ತೆಗೆದುಕೊಳ್ಳಲು input ಸ್ಟ್ರೀಮ್‌ಗಳಿವೆ.

ಮುಂದಿನ ಟ್ಯುಟೋರಿಯಲ್‌ಗಳಲ್ಲಿ ನಾವು ಇನ್ಪುಟ್ ಸ್ಟ್ರಿಮ್‌ಗಳು ಮತ್ತು ಔಟ್‌ಪುಟ್ ಸ್ಟ್ರೀಮ್‌ಗಳ ಬಗ್ಗೆ ಕಲಿಯೋಣ.

ಸ್ಟ್ರೀಮ್‌ಗಳ ವಿಧಗಳು (Types of Streams)

ಒಂದು ಸ್ಟ್ರೀಮ್ ಸಂಗ್ರಹಿಸಬಲ್ಲ ಡಾಟಾದ ಆಧಾರದ ಮೇಲೆ ಅವುಗಳನ್ನು ಈ ರೀತಿಯಾಗಿ ವರ್ಗೀಕರಿಸಲಾಗಿದೆ :

- Byte Stream
- Character Stream

ಬೈಟ್ ಸ್ಟ್ರೀಮ್ (Byte Stream)

ಡಾಟಾದಲ್ಲಿರುವ ಒಂದೇ ಬೈಟನ್ನು (8 bits) ಓದಲು ಅಥವಾ ಬರೆಯಲು ಬೈಟ್ ಸ್ಟ್ರೀಮನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

InputStream ಮತ್ತು OutputStream ಎನ್ನುವ ಬೇಸ್ ಅಬ್‌ಸ್ಟ್ರಾಕ್ಟ್ ಕ್ಲಾಸ್ (base abstract classes) ಗಳಿಂದ ಎಲ್ಲಾ ಬೈಟ್ ಸ್ಟ್ರೀಮ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲಾಗುತ್ತದೆ.

ಕ್ಯಾರೆಕ್ಟರ್ ಸ್ಟ್ರೀಮ್ (Character Stream)

ಡಾಟಾದ ಒಂದೇ ಅಕ್ಷರವನ್ನು ಓದಲು ಅಥವಾ ಬರೆಯಲು Character Stream ಅನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. Reader ಮತ್ತು Writer ಎನ್ನುವ ಬೇಸ್ ಅಬ್‌ಸ್ಟ್ರಾಕ್ಟ್ ಕ್ಲಾಸ್ (base abstract classes) ಗಳಿಂದ ಎಲ್ಲಾ Character ಸ್ಟ್ರೀಮ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಲಾಗುತ್ತದೆ.

ಜಾವಾ ಯೂಸರ ಇನ್ಪುಟ್ (Java User Input) (Scanner)

ಜಾವಾ ಯೂಸರ ಇನ್ಪುಟ್

ಬಳಕೆದಾರರಿಂದ ಇನ್ಪುಟ್ ಪಡೆದುಕೊಳ್ಳಲು Scanner class ನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಇದು java.util ಪ್ಯಾಕೇಜ್‌ನಲ್ಲಿ ಕಂಡುಬರುತ್ತದೆ.

Scanner class ಅನ್ನು ಬಳಸಲು, ಕ್ಲಾಸ್‌ನ ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸಬೇಕಾಗುತ್ತದೆ ಮತ್ತು Scanner class ಡಾಕ್ಯುಮೆಂಟೇಷನ್‌ನಲ್ಲಿ ಕಂಡುಬರುವ ಯಾವುದೆ ಲಭ್ಯವಿರುವ ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿ. ನಮ್ಮ ಉದಾಹರಣೆಯಲ್ಲಿ , ನಾವು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಓದಲು ಬಳಸಲಾಗುವ NextLine() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸೋಣ.

ಉದಾಹರಣೆ

```
importjava.util.Scanner;// Import the Scanner class


classMain{

publicstaticvoidmain(String[] args){

Scanner myObj =newScanner(System.in);// Create a Scanner object

System.out.println("Enter username");


String userName = myObj.nextLine();// Read user input

System.out.println("Username is: "+ userName);// Output user input

}

}
```

© NIMI NOT TO BE REPUBLISHED

ರನ್ ಮಾಡಿ (Run Example)

ಪ್ಯಾಕೇಜ್ ಎಂದರೇನು ಎಂದು ನಿಮಗೆ ತಿಳಿದಿಲ್ಲವೆಂದರೆ, ನಮ್ಮ ಜಾವಾ Packages ಟ್ಯುಟೋರಿಯಲ್ ಅನ್ನು ಓದಿರಿ.

ಇನ್ಪುಟ್‌ನ ವಿಧಗಳು (Input Types)

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾವು NextLine() ಮೆಥೆಡ್‌ನ್ನು ಉಪಯೋಗಿಸಿದ್ದೇವೆ. ಇದನ್ನು ಸ್ಟ್ರಿಂಗ್‌ಗಳನ್ನು ಓದಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇತರ ವಿಧಗಳನ್ನು ತಿಳಿದುಕೊಳ್ಳಲು ಕೆಳಗಿನ ಟೇಬಲ್ ನೋಡಿರಿ :

ಮೆಥೆಡ್	ವಿವರಣೆ
nextBoolean()	ಬಳಕೆದಾರರಿಂದ boolean ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ.
nextByte()	ಬಳಕೆದಾರರಿಂದ byte ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextDouble()	ಬಳಕೆದಾರರಿಂದ double ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextFloat()	ಬಳಕೆದಾರರಿಂದ float ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextInt()	ಬಳಕೆದಾರರಿಂದ int ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextLine()	ಬಳಕೆದಾರರಿಂದ string ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextLong()	ಬಳಕೆದಾರರಿಂದ long ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ
nextShort()	ಬಳಕೆದಾರರಿಂದ short ಮೌಲ್ಯವನ್ನು ಓದುತ್ತದೆ

ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾವು ವಿವಿಧ ರೀತಿಯ ಡಾಟಾವನ್ನು ಓದಲು ವಿಭಿನ್ನ ಮೆಥೆಡ್‌ಗಳನ್ನು ಉಪಯೊಗಿಸುತ್ತೇವೆ.

ಉದಾಹರಣೆ

```
import java.util.Scanner;
classMain{
public static void main(String[] args){
Scanner myObj =newScanner(System.in);
System.out.println("Enter name, age and salary:");
// String input
String name = myObj.nextLine();
// Numerical input
int age = myObj.nextInt();
double salary = myObj.nextDouble();
// Output input by user
System.out.println("Name: "+ name);
System.out.println("Age: "+ age);
System.out.println("Salary: "+ salary);
}
}
```

ಉದಾಹರಣೆಯನ್ನು ರನ್ ಮಾಡಿರಿ (Run Example)

ಗಮನಿಸಿ : ನೀವು ತಪ್ಪು ಇನ್ಪುಟ್ ಅನ್ನು ನಮೂದಿಸಿದರೆ (ಉದಾಹರಣೆಗೆ ಸಂಖ್ಯೆಯ ಇನ್ಪುಟ್‌ನಲ್ಲಿ ಟೆಕ್ಸ್ಟ್ ಅನ್ನು ನಮೂದಿಸಿದರೆ), ನೀವು exception/error ಸಂದೇಶವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತೀರಿ. (ಉದಾಹರಣೆಗೆ "Input Mismatch Exception")

Exceptions ಪಾಠದಲ್ಲಿ ನೀವು Exceptions ಗಳ ಬಗ್ಗೆ ಇನ್ನೂ ಹೆಚ್ಚು ತಿಳಿದುಕೊಳ್ಳಬಹುದು ಮತ್ತು error ಗಳನ್ನು ಹೇಗೆ ನಿಭಾಯಿಸುವುದು ಎನ್ನುವುದನ್ನು ತಿಳಿದುಕೊಳ್ಳಬಹುದು.

ಜಾವಾ ಕನ್ಸೋಲ್ ಕ್ಲಾಸ್ (Java Console Class)

ಕನ್ಸೋಲ್‌ನಿಂದ ಇನ್ಪುಟ್ ಅನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಜಾವಾ Console Class ಅನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದು ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಮತ್ತು ಪಾಸ್‌ವರ್ಡ್‌ಗಳನ್ನು ಓದಲು ಮೆಥೆಡ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

Console ಕ್ಲಾಸ್ ಅನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಓದಿದರೆ, ಅದು ಬಳಕೆದಾರರಿಗೆ ಪ್ರದರ್ಶಿತವಾಗುವದಿಲ್ಲ.

ಸಿಸ್ಟಮ್ ಕನ್ಸೋಲ್‌ನೊಂದಿಗೆ java.io.Console class ಅನ್ನು ಆಂತರಿಕವಾಗಿ ಲಗತ್ತಿಸಲಾಗಿರುತ್ತದೆ. Console ಕ್ಲಾಸ್‌ನ್ನು 1.5 ರಿಂದ ಪರಿಚಯಿಸಲಾಗಿದೆ.

ಕನ್ಸೋಲ್‌ನಿಂದ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಓದಲು ಒಂದು ಸರಳವಾದ ಉದಾಹರಣೆಯನ್ನು ನೋಡೋಣ:

```
1  String text=System.console().readLine();
2  System.out.println("Text is: "+text);
```

ಜಾವಾ ಕನ್ಸೋಲ್ ಕ್ಲಾಸ್ ಡಿಕ್ಲೇರೇಷನ್ (Java Console class declaration)

Java.io.Console class ನ ಡಿಕ್ಲೇರೇಷನ್ ನೊಡೋಣ :

```
1  Public final class Console extends Object implements Flushable
```

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾ ಕನ್ಸೋಲ್ ಕ್ಲಾಸ್ ಮೆಥೆಡ್‌ಗಳು (Java Console class methods)

ಮೆಥೆಡ್	ವಿವರಣೆ
Reader reader()	ಇದನ್ನು ಕನ್ಸೋಲ್‌ಗೆ ಸಂಬಂಧಿಸಿದ ರಿಡರ್ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಪುನಃ ಪಡೆದುಕೊಳ್ಳಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
String readLine()	ಇದನ್ನು ಕನ್ಸೋಲ್‌ನಿಂದ ಒಂದೇ ಸಾಲಿನ ಟೆಕ್ಸ್ಟ್‌ನ್ನು ಓದಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
String readLine(String fmt, Object... args)	ಇದು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿದ ಪ್ರಾಂಪ್ಟ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಕನ್ಸೋಲ್‌ನಿಂದ ಒಂದೇ ಸಾಲಿನ ಟೆಕ್ಸ್ಟ್ ಅನ್ನು ಓದುತ್ತದೆ.
char[] readPassword()	ಇದನ್ನು ಕನ್ಸೋಲ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡದ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಓದಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
char[] readPassword(String fmt, Object... args)	ಇದು ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿದ ಪ್ರಾಂಪ್ಟನ್ನು ಒದಗಿಸುತ್ತದೆ ಮತ್ತು ಆನಂತರ ಕನ್ಸೋಲ್‌ನಲ್ಲಿ ಪ್ರದರ್ಶಿಸಲ್ಪಡದ ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಓದುತ್ತದೆ
Console format(String fmt, Object... args)	ಇದನ್ನು ಕನ್ಸೋಲ್‌ನ ಔಟ್‌ಪುಟ್ ಸ್ಟ್ರೀಮ್‌ನಲ್ಲಿ ಫಾರ್ಮ್ಯಾಟ್ ಮಾಡಿದ ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು ಬರೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
Console printf(String format, Object... args)	ಇದನ್ನು ಕನ್ಸೋಲ್‌ನ ಔಟ್‌ಪುಟ್ ಸ್ಟ್ರೀಮ್‌ನಲ್ಲಿ ಸ್ಟ್ರಿಂಗ್ ಅನ್ನು ಬರೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
PrintWriter writer()	ಇದನ್ನು ಕನ್ಸೋಲ್‌ಗೆ ಸಂಬಂಧಿಸಿದ PrintWriter ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಪುನಃ ಪಡೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
void flush()	ಇದನ್ನು ಕನ್ಸೋಲ್ ಅನ್ನು ಫ್ಲಶ್ (ಚದುರಿಹೋಗು) ಮಾಡಲು ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.

ಕನ್ಸೋಲ್‌ನ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಹೇಗೆ ಪಡೆದುಕೊಳ್ಳುವುದು (How to get the object of Console)

System class, console() ಎನ್ನುವ ಒಂದು static ಮೆಥೆಡ್‌ನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು Console class ನ singleton instance ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

```
public static Console console(){}
```

Console class ನ instance ಅನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಕೋಡ್‌ನ್ನು ನೋಡೋಣ :

```
Console c=System.console();
```

ಜಾವಾ Console ಉದಾಹರಣೆ :

```
import java.io.Console;
class ReadStringTest{
public static void main(String args[]){
Console c=System.console();
System.out.println(“Enter your name: “);
String n=c.readLine();
System.out.println(“Welcome “+n);
}
}
```

ಫಲಿತಾಂಶ :

```
Enter your name: Nakul Jain
Welcome Nakul Jain
```

ಪಾಸ್‌ವರ್ಡ್‌ನ್ನು ಓದಲು ಜಾವಾ Console ಉದಾಹರಣೆ :

```
import java.io.Console;
class ReadPasswordTest{
public static void main(String args[]){
Console c=System.console();
System.out.println(“Enter password: “);
char[] ch=c.readPassword();
String pass=String.valueOf(ch);//converting char array into string
System.out.println(“Password is: “+pass);
}
}
```

ಫಲಿತಾಂಶ :

```
Enter password:
Password is: 123
```

NOT TO BE REPUBLISHED © NIMI

ಜಾವಾ ಪ್ರೊಗ್ರಾಂ ಪ್ರವಹಿಸುವುದನ್ನು ನಿಯಂತ್ರಿಸುವುದು (JAVA Program Flow Control)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಕಂಡೀಷನಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳು
- ಲೂಪ್‌ಗಳು
- ಅರೇಗಳು

ಜಾವಾ ಕಂಡೀಷನ್‌ಗಳು ಮತ್ತು if ಸ್ಟೇಟಮೆಂಟ್ (Java Conditions and If Statements)

ಜಾವಾ ಗಣಿತದಲ್ಲಿನ ಸಾಮಾನ್ಯ ತಾರ್ಕಿಕ ಕಂಡೀಷನ್‌ಗಳನ್ನು ಬೆಂಬಲಿಸುತ್ತದೆ ಎಂದು ನಿಮಗೆ ಈಗಾಗಲೇ ತಿಳಿದಿದೆ.

- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b
- Equal to a == b

ನೀವು ಈ ಕಂಡೀಷನ್‌ಗಳನ್ನು ವಿಭಿನ್ನ ನಿರ್ಧಾರಗಳಿಗಾಗಿ ವಿಭಿನ್ನ ಕಾರ್ಯಗಳನ್ನು ಮಾಡಲು ಉಪಯೋಗಿಸಬಹುದು.

ಜಾವಾ ಕೆಳಗಿನ ಕಂಡಿಷನಲ್ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿದೆ :

- ಒಂದು ನಿರ್ದಿಷ್ಟವಾದ ಕಂಡೀಷನ್ true ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೊಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಸೂಚಿಸಲು if ಅನ್ನು ಬಳಸಿರಿ.
- ಅದೇ ನಿರ್ದಿಷ್ಟವಾದ ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೊಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಸೂಚಿಸಲು else ಅನ್ನು ಬಳಸಿರಿ.
- ಮೊದಲನೇ ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ, ಹೊಸ ಕಂಡೀಷನ್ ಅನ್ನು ಪರೀಕ್ಷಿಸಲು else if ಅನ್ನು ಬಳಸಿರಿ.
- ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಅನೇಕ ಪರ್ಯಾಯ ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ಗಳನ್ನು ಸೂಚಿಸಲು switch ಅನ್ನು ಬಳಸಿರಿ.

if ಸ್ಟೇಟಮೆಂಟ್ (The if Statement)

ಒಂದು ಕಂಡೀಷನ್ true ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೊಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಸೂಚಿಸಲು if ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಬಳಸಿರಿ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
if(condition){
// block of code to be executed if the condition is true
}
```

if ಇದು ಸಣ್ಣ ಅಕ್ಷರಗಳಲ್ಲಿದೆ (lowercase letters) ಎಂಬುದನ್ನು ಗಮನಿಸಿ. ದೊಡ್ಡ ಅಕ್ಷರಗಳು(uppercase letters) (If ಅಥವಾ IF) ಎರರ್‌ನ್ನು ಉಂಟು ಮಾಡುತ್ತವೆ.

ಕೆಳಗಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, 20 ಕ್ಕಿಂತ 18 ಹೆಚ್ಚು ಎಂಬುದನ್ನು ಕಂಡುಹಿಡಿಯಲು ನಾವು ಎರಡು ಸಂಖ್ಯೆಗಳನ್ನು ಪರೀಕ್ಷಿಸೋಣ.

ಉದಾಹರಣೆ

```
if(20>18){
System.out.println("20 is greater than 18");
}
```

Try it Yourself

We can also test variables:

ಉದಾಹರಣೆ

```
int x =20;
int y =18;
if(x > y){
System.out.println("x is greater than y");
}
```

Try it Yourself "

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, x , y ಗಿಂತ ದೊಡ್ಡದಾಗಿದೆಯೆ

ಎಂದು ಪರೀಕ್ಷಿಸಲು (> ಆಪರೇಟರ್ ಉಪಯೋಗಿಸಿ) ನಾವು ಎರಡು ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿದ್ದೇವೆ : x ಮತ್ತು y. x ನ ಮೌಲ್ಯ 20 ಆಗಿದೆ ಮತ್ತು y ನ ಮೌಲ್ಯ 18 ಆಗಿದೆ ಮತ್ತು 20 ಸಂಖ್ಯೆ 18 ಕ್ಕಿಂತ ದೊಡ್ಡದು ಎಂದು ನಮಗೆ ಗೊತ್ತಿದೆ. ನಾವು ಸ್ಕ್ರೀನ್ ಮೇಲೆ "x is greater than y". ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡೋಣ.

else ಸ್ಟೇಟಮೆಂಟ್ (The else Statement)

ಒಂದು ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ, ಕಾರ್ಯಗತಗೊಳಿಸಬೇಕಾದ ಕೊಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಸೂಚಿಸಲು else ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಬಳಸಿರಿ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
if(condition){
// block of code to be executed if the condition is true
}else{
// block of code to be executed if the condition is false
}
```

ಉದಾಹರಣೆ

```
int time =20;
if(time <18){
System.out.println("Good day.");
}else{
System.out.println("Good evening.");
}
// Outputs "Good evening."
```

Try it Yourself

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, time (20) , 18 ಕ್ಕಿಂತ ದೊಡ್ಡದಾಗಿದೆ. ಅದ್ದರಿಂದ ಕಂಡೀಷನ್ false ಆಗುತ್ತದೆ. ಈ ಕಾರಣಕ್ಕಾಗಿ ನಾವು else ಕಂಡೀಷನ್‌ಗೆ ಹೋಗುತ್ತೇವೆ ಮತ್ತು ಸ್ಕ್ರೀನ್‌ನಲ್ಲಿ "Good evening." ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತೇವೆ. ಒಂದು ವೇಳೆ time, 18 ಕ್ಕಿಂತ ಕಡಿಮೆ ಆಗಿದ್ದರೆ, ಪ್ರೊಗ್ರಾಂ "Good day." ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತಿತ್ತು.

else if ಸ್ಟೇಟಮೆಂಟ್ (The else if Statement)

ಮೊದಲನೇ ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ, ಹೊಸ ಕಂಡೀಷನ್ ಅನ್ನು ಸೂಚಿಸಲು else if ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸಿ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
if(condition1){
// condition1, true ಆಗಿದ್ದರೆ ಈ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.
}elseif(condition2){
// condition1 , false ಆಗಿದ್ದು ಮತ್ತು condition2 , true ಆಗಿದ್ದರೆ ಈ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.
}else{
// condition1, false ಆಗಿದ್ದು ಮತ್ತು condition2 ಸಹ false ಆಗಿದ್ದರೆ ಈ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.
}
```

ಉದಾಹರಣೆ

```
int time =22;
if(time <10){
System.out.println("Good morning.");
}elseif(time <18){
System.out.println("Good day.");
}else{
System.out.println("Good evening.");
}
// Outputs "Good evening."
```

Try it Yourself

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ.

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, time(22) ಇದು 10 ಕ್ಕಿಂತ ದೊಡ್ಡದಾಗಿದೆ ಅದ್ದರಿಂದ ಮೊದಲನೇ ಕಂಡೀಷನ್ false ಆಗುತ್ತದೆ. else if ನಲ್ಲಿರುವ ಮುಂದಿನ ಕಂಡೀಷನ್, ಸಹ false ಆಗಿದೆ, ಅದ್ದರಿಂದ ನಾವು else ಕಂಡೀಷನ್‌ಗೆ ಹೋಗುತ್ತೇವೆ. ಹೀಗಾಗಿ condition1 and condition2 ಎರಡೂ ಸಹ false ಆಗಿರುವುದರಿಂದ - ಸ್ಕ್ರೀನ್ ಮೇಲೆ "Good evening." ಎಂದು ಪ್ರಿಂಟ್ ಆಗುತ್ತದೆ.

time 14 ಆಗಿದ್ದರೆ, ನಮ್ಮ ಪ್ರೊಗ್ರಾಂ "Good day." ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತಿತ್ತು.

ಜಾವಾ switch (Java Switch)

ಜಾವಾ switch ಸ್ಟೇಟಮೆಂಟ್ (Java Switch Statements)

ಅನೇಕ if..else ಸ್ಟೇಟಮೆಂಟ್‌ಗಳನ್ನು ಬರೆಯುವ ಬದಲು ನೀವು ಒಂದು switch ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

switch ಸ್ಟೇಟಮೆಂಟ್, ಹಲವಾರು ಕೋಡ್ ಬ್ಲಾಕ್‌ಗಳಲ್ಲಿ ಒಂದನ್ನು ಆಯ್ಕೆ ಮಾಡಿಕೊಂಡು ಅದನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
switch(expression){
case x:
// code block
break;
case y:
// code block
break;
default:
// code block
}
```

ಇದು ಹೀಗೆ ಕಾರ್ಯ ನಿರ್ವಹಿಸುತ್ತದೆ:

- expression ಒಮ್ಮೆ ಮಾತ್ರ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.
- expression ನ ಮೌಲ್ಯವನ್ನು ಪ್ರತಿ case ನ ಮೌಲ್ಯದೊಂದಿಗೆ ಹೋಲಿಸಲಾಗುತ್ತದೆ.
- ಇಲ್ಲಿ ಹೊಂದಾಣಿಕೆ ಕಂಡುಬಂದರೆ, ಸಂಬಂಧಪಟ್ಟ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.
- break ಮತ್ತು default ಕೀವರ್ಡ್‌ಗಳು ಐಚ್ಛಿಕವಾಗಿವೆ, ಮತ್ತು ಈ ಪಾಠದಲ್ಲಿ ನಂತರ ವಿವರಿಸಲಾಗುತ್ತದೆ.

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು ವಾರದ ದಿನಗಳ ಹೆಸರುಗಳನ್ನು ಲೆಕ್ಕ ಮಾಡಲು ವಾರದ ದಿನದ ಸಂಖ್ಯೆಯನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ :

```
int day =4;
switch(day){
case1:
System.out.println("Monday");
break;
case2:
System.out.println("Tuesday");
break;
case3:
System.out.println("Wednesday");
break;
case4:
System.out.println("Thursday");
break;
case5:
System.out.println("Friday");
break;
case6:
System.out.println("Saturday");
break;
case7:
System.out.println("Sunday");
break;
}
// Outputs "Thursday" (day 4)
```

Try it Yourself"

break ಕೀವರ್ಡ್ (The break Keyword)

ಜಾವಾ break ಕೀವರ್ಡ್‌ನ್ನು ತಲುಪಿದಾಗ, ಅದು switch ಬ್ಲಾಕ್‌ನಿಂದ ಹೊರಗೆ ಬರುತ್ತದೆ.

ಇದು ಮುಂದಿನ ಕೋಡ್‌ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸುವುದನ್ನು ತಡೆಯುತ್ತದೆ ಮತ್ತು ಬ್ಲಾಕ್‌ನಲ್ಲಿ case ಅನ್ನು ಪರೀಕ್ಷಿಸುವುದನ್ನು ನಿಲ್ಲಿಸುತ್ತದೆ.

case ಗೆ ಹೊಂದಾಣಿಕೆ ಕಂಡುಬಂದಾಗ ಮತ್ತು ಕೆಲಸವು ಪೂರ್ಣಗೊಂಡಾಗ, break ತೆಗೆದುಕೊಳ್ಳುವಸಮಯವಾಗಿರುತ್ತದೆ.ಇನ್ನು ಹೆಚ್ಚಿನ ಪರೀಕ್ಷೆಯ ಅಗತ್ಯವಿರುವುದಿಲ್ಲ.

ಒಂದು break ಎಕ್ಸಿಕ್ಯೂಷನ್‌ನ ಬಹಳಷ್ಟು ಸಮಯವನ್ನು ಉಳಿಸುತ್ತದೆ. ಏಕೆಂದರೆ ಇದು switch ಬ್ಲಾಕ್‌ನಲ್ಲಿರುವ ಉಳಿದ ಎಲ್ಲಾ ಕೋಡ್‌ಗಳು ಕಾರ್ಯನಿರ್ವಹಿಸುವುದನ್ನು "ನಿರ್ಲಕ್ಷಿಸುತ್ತದೆ".

default ಕೀವರ್ಡ್ (The default Keyword)

case ಗೆ ಯಾವುದೆ ಹೊಂದಾಣಿಕೆ ಕಂಡುಬರದಿದ್ದರೆ, ರನ್ ಮಾಡಬೇಕಾದ ಕೋಡನ್ನು default ಕೀವರ್ಡ್ ಸೂಚಿಸುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

ಉದಾಹರಣೆ

```
int day =4;
switch(day){
case6:
System.out.println("Today is Saturday");
break;
case7:
System.out.println("Today is Sunday");
break;
default:
System.out.println("Looking forward to the Weekend");
}
// Outputs "Looking forward to the Weekend"
```

Try it Yourself"

ಒಂದು ವೇಳೆ default ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು switch ಬ್ಲಾಕ್‌ನ ಕೊನೆಯ ಸ್ಟೇಟಮೆಂಟ್ ಆಗಿ ಉಪಯೋಗಿಸಿದರೆ, ಇದಕ್ಕೆ break ನ ಅವಶ್ಯಕತೆ ಇರುವುದಿಲ್ಲ ಎಂಬುದನ್ನು ಗಮನಿಸಿ.

ಲೂಪ್‌ಗಳು (Loops)

ಸೂಚಿಸಲಾದ ಕಂಡೀಷನ್‌ನ್ನು ತಲುಪುವವರೆಗೆ ಲೂಪ್‌ಗಳು ಕೋಡ್‌ನ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತಿರುತ್ತವೆ.

ಲೂಪ್‌ಗಳು ಸೂಕ್ತವಾಗಿವೆ ಏಕೆಂದರೆ ಅವು ಸಮಯವನ್ನು ಉಳಿಸುತ್ತವೆ, ಎರರ್‌ಗಳನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತವೆ, ಮತ್ತು ಕೋಡ್‌ನ್ನು ಓದಲು ಹೆಚ್ಚು ಅನುಕೂಲವನ್ನು ಮಾಡಿಕೊಡುತ್ತವೆ.

ಜಾವಾ While ಲೂಪ್ (Java While Loop)

ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಂಡೀಷನ್ true ಇರುವವರೆಗೆ While ಲೂಪ್ ಅದರ ಬ್ಲಾಕ್‌ನಲ್ಲಿರುವ ಕೋಡನ್ನು ಪುನರಾವರ್ತಿಸುತ್ತಿರುತ್ತದೆ (loop) :

ಸಿಂಟ್ಯಾಕ್ಸ್

```
while(condition){
// code block to be executed
}
```

ಕೆಳಗೆ ನೀಡಲಾದ ಉದಾಹರಣೆಯಲ್ಲಿ, ಎಲ್ಲಿಯವರೆಗೆ ವೇರಿಯೇಬಲ್ (i) 5 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇರುತ್ತದೆಯೊ ಅಲ್ಲಿಯವರೆಗೆ ಲೂಪ್‌ನಲ್ಲಿರುವ ಕೋಡ್, ಮೇಲಿಂದ ಮೇಲೆ ರನ್ ಆಗುತ್ತಿರುತ್ತದೆ :

ಉದಾಹರಣೆ

```
int i =0;
while(i <5){
System.out.println(i);
i++;
}
```

Try it Yourself"

ಸೂಚನೆ : ಕಂಡೀಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲಾದ ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಚ್ಚಿಸಲು (increment) ಮರೆಯಬೇಡಿರಿ, ಇಲ್ಲದಿದ್ದರೆ ಲೂಪ್ ಎಂದಿಗೂ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ!

Do/While ಲೂಪ್ (The Do/While Loop)

do/while ಲೂಪ್, while ಲೂಪ್‌ನ ರೂಪಾಂತರವಾಗಿದೆ. ಇದು ಕಂಡೀಷನ್ true ಆಗಿದೆಯೆ ಎಂದು ಪರೀಕ್ಷಿಸುವ ಮೊದಲೆ ಕೋಡ್ ಬ್ಲಾಕ್‌ನ್ನು ಒಂದು ಬಾರಿ ಕಾರ್ಯಗತಗೊಳಿಸುತ್ತದೆ. ನಂತರ ಇದು ಕಂಡೀಷನ್ true ಇರುವವರೆಗೆ ಲೂಪನ್ನು ಪುನರಾವರ್ತಿಸುತ್ತಿರುತ್ತದೆ.

ಸಿಂಟ್ಯಾಕ್ಸ್

```
do{
// code block to be executed
}
while(condition);
```

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು do/while ಲೂಪನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಯಾವಾಗಲೂ ಕಂಡೀಷನ್ false ಇದ್ದರೂ ಸಹ ಈ ಲೂಪ್ ಒಂದು ಬಾರಿಯಾದರೂ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ ಏಕೆಂದರೆ ಕಂಡೀಷನ್‌ನ್ನು ಪರೀಕ್ಷಿಸುವ ಮೊದಲೇ ಕೋಡ್ ಬ್ಲಾಕ್ ಕಾರ್ಯನಿರ್ವಹಿಸುತ್ತದೆ.

ಉದಾಹರಣೆ

```
int i =0;
do{
System.out.println(i);
i++;
}
while(i <5);
```

© NIMI NOT TO BE REPUBLISHED

ಸೂಚನೆ : ಕಂಡೀಷನ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲಾದ ವೇರಿಯೇಬಲ್‌ನ್ನು ಹೆಚ್ಚಿಸಲು ಮರೆಯಬೇಡಿರಿ, ಇಲ್ಲದಿದ್ದರೆ ಲೂಪ್ ಎಂದಿಗೂ ಕೊನೆಗೊಳ್ಳುವುದಿಲ್ಲ!

ಜಾವಾ for ಲೂಪ್ (Java For Loop)

ಒಂದು ಕೋಡ್ ಬ್ಲಾಕನ್ನು ಎಷ್ಟು ಬಾರಿ ಪುನರಾವರ್ತಿಸಬೇಕು ಎಂಬುದು ನಿಮಗೆ ಖಚಿತವಾಗಿ ತಿಳಿದಿದ್ದರೆ, while ಲೂಪ್ ಬದಲಾಗಿ for ಲೂಪನ್ನು ಉಪಯೋಗಿಸಿರಿ :

ಸಿಂಟ್ಯಾಕ್ಸ್

```
for(statement 1; statement 2; statement 3){
// code block to be executed
}
```

ಕೋಡ್ ಬ್ಲಾಕ್ ಕಾರ್ಯಗತಗೊಳ್ಳುವ ಮೊದಲು, Statement 1 ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತದೆ (ಒಂದು ಬಾರಿ).

ಕೋಡ್ ಬ್ಲಾಕ್‌ನ್ನು ಕಾರ್ಯಗತಗೊಳಿಸಲು ಬೇಕಾದ ಕಂಡೀಷನ್ ಅನ್ನು Statement 2 ವ್ಯಾಖ್ಯಾನಿಸುತ್ತದೆ.

ಕೋಡ್ ಬ್ಲಾಕ್ ಕಾರ್ಯಗತಗೊಂಡ ನಂತರ, Statement 3 ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ (ಪ್ರತಿ ಬಾರಿ).

ಕೆಳಗಿನ ಉದಾಹರಣೆಯು 0 ದಿಂದ 4 ರವರೆಗೆ ಸಂಖ್ಯೆಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ:

```
for(int i =0; i <5; i++){
System.out.println(i);
}
```

ನೀವು ಪ್ರಯತ್ನಿಸಿ

ಉದಾಹರಣೆಯನ್ನು ವಿವರಿಸಲಾಗಿದೆ

ಲೂಪ್ ಪ್ರಾರಂಭವಾಗುವ ಮೊದಲು, Statement 1 ವೇರಿಯೇಬಲ್‌ನ್ನು ಸೆಟ್ ಮಾಡುತ್ತದೆ. (int i=0)

ಲೂಪ್ ರನ್ ಆಗಲು, Statement 2 ಕಂಡೀಷನ್ ಅನ್ನು ಸೆಟ್ ಮಾಡುತ್ತದೆ.(i ನ ಮೌಲ್ಯವು 5 ಕ್ಕಿಂತ ಕಡಿಮೆ ಇರಬೇಕು). ಒಂದು ವೇಳೆ ಕಂಡಿಷನ್ true ಆಗಿದ್ದರೆ, ಲೂಪ್ ಪುನಃ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಕಂಡೀಷನ್ false ಆಗಿದ್ದರೆ, ಲೂಪ್ ಅಲ್ಲಿಯೇ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ.

ಲೂಪ್‌ನಲ್ಲಿರುವ ಕೋಡ್ ಬ್ಲಾಕ್ ಕಾರ್ಯಗತಗೊಂಡಾಗ Statement 3 ಪ್ರತಿ ಬಾರಿಯೂ ಮೌಲ್ಯವನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ (i++)

ಇನ್ನೊಂದು ಉದಾಹರಣೆ

ಈ ಉದಾಹರಣೆಯು 0 ದಿಂದ 10 ರವರೆಗಿನ ಸಮ ಸಂಖ್ಯೆಗಳನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ.

ಉದಾಹರಣೆ

```
for(int i =0; i <=10; i = i +2){
System.out.println(i);
}
```

ನೀವು ಪ್ರಯತ್ನಿಸಿ

ನೆಸ್ಟೆಡ್ ಲೂಪ್‌ಗಳು (Nested Loops)

ಒಂದು ಲೂಪ್ ಅನ್ನು ಇನ್ನೊಂದು ಲೂಪ್‌ನ ಒಳಗಡೆ ಇರಿಸಲು ಸಾಧ್ಯವಿದೆ. ಇದನ್ನು ನೆಸ್ಟೆಡ್ ಲೂಪ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

"outer loop" ನ ಪ್ರತಿ ಪುನರಾವರ್ತನೆಗೆ "inner loop" ನ್ನು ಒಂದು ಬಾರಿ ಕಾರ್ಯಗತಗೊಳಿಸಲಾಗುತ್ತದೆ.:

ಉದಾಹರಣೆ

```
// Outer loop
for(int i =1; i <=2; i++){
System.out.println("Outer: "+ i);// Executes 2 times
// Inner loop
for(int j =1; j <=3; j++){
System.out.println("Inner: "+ j);// Executes 6 times (2 * 3)
}
}
```

ಜಾವಾ break (Java Break)

ಈ ಟ್ಯುಟೋರಿಯಲ್‌ನ ಹಿಂದಿನ ಅಧ್ಯಯದಲ್ಲಿ ನೀವು ಈಗಾಗಲೇ break ಸ್ಟೇಟಮೆಂಟ್ ಬಗ್ಗೆ ತಿಳಿದುಕೊಂಡಿದ್ದೀರಿ. ಇದನ್ನು switch ಸ್ಟೇಟಮೆಂಟ್‌ನಿಂದ ಹೊರಗೆ ಬರಲು ಬಳಸಲಾಗುತ್ತಿತ್ತು.

break ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಒಂದು ಲೂಪ್‌ನಿಂದ ಹೊರಗೆ ಬರಲು ಸಹ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ ಯಾವಾಗ i ನ ಮೌಲ್ಯವು 4 ಕ್ಕೆ ಸಮನಾಗುತ್ತದೆಯೊ ಆಗ ಲೂಪ್ ಕೊನೆಗೊಳ್ಳುತ್ತದೆ.:

ಉದಾಹರಣೆ

```
for(int i =0; i <10; i++){
if(i ==4){
```

```
break;
}
System.out.println(i);
}
```

ನೀವು ಪ್ರಯತ್ನಿಸಿ

ಜಾವಾ Continue (Java Continue)

ಒಂದು ನಿರ್ದಿಷ್ಟಪಡಿಸಿದ ಕಂಡೀಷನ್ ಸಂಭವಿಸಿದಾಗ, Continue ಸ್ಟೇಟಮೆಂಟ್ ಲೂಪ್‌ನಲ್ಲಿರುವ ಒಂದು ಪುನರಾವರ್ತನೆಯನ್ನು break ಮಾಡುತ್ತದೆ ಮತ್ತು ಲೂಪ್‌ನಲ್ಲಿರುವ ಮುಂದಿನ ಪುನರಾವರ್ತನೆಯನ್ನು ಮುಂದುವರೆಸುತ್ತದೆ.

ಈ ಉದಾಹರಣೆಯು 4 ರ ಮೌಲ್ಯವನ್ನು ಬಿಟ್ಟುಬಿಡುತ್ತದೆ:

ಉದಾಹರಣೆ

```
for(int i =0; i <10; i++){
if(i ==4){
continue;
}
System.out.println(i);
}
```

ನೀವು ಪ್ರಯತ್ನಿಸಿ

While ಲೂಪ್‌ನಲ್ಲಿ Break ಮತ್ತು Continue (Break and Continue in While Loop)

While ಲೂಪ್‌ನಲ್ಲಿ ನೀವು Break ಮತ್ತು Continue ಗಳನ್ನು ಸಹ ಉಪಯೋಗಿಸಬಹುದು :

Break ಉದಾಹರಣೆ

```
int i =0;
while(i <10){
System.out.println(i);
i++;
if(i ==4){
break;
}
}
```

ನೀವು ಪ್ರಯತ್ನಿಸಿ

Continue ಉದಾಹರಣೆ

```
int i =0;
while(i <10){
if(i ==4){
i++;
continue;
}
System.out.println(i);
i++;
}
```

ಜಾವಾ ಅರೇಗಳು (Java Arrays)

ಒಂದೇ ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಅನೇಕ ಮೌಲ್ಯಗಳನ್ನು ಸಂಗ್ರಹಿಸಲು ಅರೇಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಪ್ರತಿಯೊಂದು ಮೌಲ್ಯಕ್ಕಾಗಿ ಪ್ರತ್ಯೇಕ ವೇರಿಯೇಬಲ್‌ಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುವ ಬದಲು ನೀವು ಅರೇಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು.

ಅರೇಯನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಲು ವೇರಿಯೇಬಲ್ ಟೈಪ್‌ನ್ನು ಚೌಕಾಕಾರದ ಆವರಣದೊಟ್ಟಿಗೆ ವ್ಯಾಖ್ಯಾನಿಸಿ:

```
String[ ] cars;
```

ನಾವು ಈಗ ಸ್ಟ್ರಿಂಗ್‌ಗಳ ಅರೇಯನ್ನು ಸಂಗ್ರಹಿಸುವ ಒಂದು ವೇರಿಯೇಬಲ್ ಅನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡಿದ್ದೇವೆ. ಅದಕ್ಕೆ ಮೌಲ್ಯಗಳನ್ನು ಸೇರಿಸಲು , ಕರ್ಲೀ ಬ್ರೇಸ್‌ಗಳಲ್ಲಿ ನೀವು ಮೌಲ್ಯಗಳನ್ನು ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಲಾದ ಪಟ್ಟಿಯಲ್ಲಿ ಇರಿಸಬಹುದು :

```
String[ ] cars ={"Volvo","BMW","Ford","Mazda"};
```

ಪೂರ್ಣಾಂಕಗಳ(integers) ಅರೇಯನ್ನು ರಚಿಸಲು, ಈ ರೀತಿಯಾಗಿ ಬರೆಯಿರಿ:

```
int[ ] myNum ={10,20,30,40};
```

ಅರೇಯ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವುದು (Access the Elements of an Array)

ಇಂಡೆಕ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಉಪಯೋಗಿಸಿ ನೀವು ಅರೇಯ ಎಲಿಮೆಂಟ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು.

ಈ ಸ್ಟೇಟಮೆಂಟ್ cars ನಲ್ಲಿರುವ ಮೊದಲನೇ ಎಲಿಮೆಂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆ:

ಉದಾಹರಣೆ

```
String[ ] cars ={"Volvo","BMW","Ford","Mazda"};
```

© NIMI NOT TO BE REPUBLISHED

```
System.out.println(cars[O]);

// Outputs Volvo
```

ನೀವೇ ಪ್ರಯತ್ನಿಸಿ

ಸೂಚನೆ : ಅರೇಯ ಇಂಡೆಕ್ಸ್ 0 ದಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. [0] ಮೊದಲನೇ ಎಲೆಮೆಂಟ್ ಆಗಿದೆ, [1] ಎರಡನೆ ಎಲೆಮೆಂಟ್ ಆಗಿದೆ... ಇತ್ಯಾದಿ.

ಅರೇ ಎಲೆಮೆಂಟ್‌ನ್ನು ಬದಲಾಯಿಸುವುದು (Change an Array Element)

ಒಂದು ನಿರ್ದಿಷ್ಟ ಎಲೆಮೆಂಟ್‌ನ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸಲು ಇಂಡೆಕ್ಸ್ ಸಂಖ್ಯೆಯನ್ನು ಉಪಯೋಗಿಸಿರಿ.

ಉದಾಹರಣೆ

```
cars[O]="Opel";
```

ಉದಾಹರಣೆ

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};

cars[O]="Opel";

System.out.println(cars[O]);

// Now outputs Opel instead of Volvo
```

ನೀವೇ ಪ್ರಯತ್ನಿಸಿ

ಅರೇಯ ಲೆಂಥ್ (Array Length)

ಒಂದು ಅರೇ ಎಷ್ಟು ಎಲೆಮೆಂಟ್‌ಗಳನ್ನು ಹೊಂದಿದೆ ಎಂದು ಕಂಡುಹಿಡಿಯಲು length property ಯನ್ನು ಬಳಸಿರಿ:

ಉದಾಹರಣೆ

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};

System.out.println(cars.length);

// Outputs 4
```

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾ ಕ್ಲಾಸ್‌ಗಳು, ಓವರ್‌ಲೋಡಿಂಗ್ ಮತ್ತು ಇನ್ಹೆರಿಟನ್ಸ್ (JAVA Classes, Overloading and Inheritance)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಜಾವಾ OOPs ಪರಿಕಲ್ಪನೆಗಳು
- ಆಬ್ಜೆಕ್ಟ್ ಕ್ಲೋನಿಂಗ್
- ಮೆಥೆಡ್‌ನ ವಿಧಗಳು.

ಜಾವಾ ಆಬ್ಜೆಕ್ಟ್ ಕ್ಲಾಸ್ (Java Object Class)

ಜಾವಾ OOPs ಪರಿಕಲ್ಪನೆಗಳು (Java OOPs Concept)

ನಾವು OOPs ನ ಮೂಲಭೂತ ಅಂಶಗಳ ಬಗ್ಗೆ ತಿಳಿದುಕೊಳ್ಳೋಣ. ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಒಂದು ಮಾದರಿಯಾಗಿದೆ (ಥಿಯರಿ). ಇದು ಹಲವಾರು ಪರಿಕಲ್ಪನೆಗಳನ್ನು (ವಿಚಾರಗಳು) ಒದಗಿಸುತ್ತದೆ. ಉದಾಹರಣೆಗೆ - ಇನ್‌ಹೆರಿಟನ್ಸ್(inheritance) , ಡಾಟಾ ಬೈಂಡಿಂಗ್ (data binding,) , ಪಾಲಿಮಾರ್ಫಿಸಮ್ (polymorphism) ಇತ್ಯಾದಿಗಳು.

ಸಿಮ್ಯುಲಾವನ್ನು (simula) ಮೊದಲ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆ ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ. ಯಾವ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಮಾದರಿಯಲ್ಲಿ ಎಲ್ಲವನ್ನೂ ಆಬ್ಜೆಕ್ಟ್ ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆಯೊ ಅದನ್ನು ನಿಜವಾದ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಸ್ಮಾಲ್‌ಟಾಕನ್ನು(Smalltalk) ಮೊದಲ ನಿಜವಾದ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆ ಎಂದು ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ. ಇತರ ಜನಪ್ರಿಯ ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಗಳು ಯಾವವೆಂದರೆ : Java, C#, PHP, Python, C++, ಇತ್ಯಾದಿಗಳು.

ನೈಜ ಪ್ರಪಂಚದ ಘಟಕಗಳನ್ನು ಜಾರಿಗೊಳಿಸುವುದು, ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯ ಮುಖ್ಯವಾದ ಗುರಿಯಾಗಿದೆ, ಉದಾಹರಣೆಗೆ, object, classes, abstraction, inheritance, polymorphism, abstraction, encapsulation, ಇತ್ಯಾದಿಗಳು.

OOPs (Object-Oriented Programming System)

ಆಬ್ಜೆಕ್ಟ್ ಎಂದರೆ ನೈಜ ಜಗತ್ತಿನ ಒಂದು ಘಟಕವಾಗಿದೆ ಉದಾಹರಣೆಗೆ ಪೆನ್, ಕುರ್ಚಿ, ಟೇಬಲ್, ಕಂಪ್ಯೂಟರ್, ಗಡಿಯಾರ ಇತ್ಯಾದಿಗಳು. ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಎಂದರೆ ಕ್ಲಾಸ್‌ಗಳು ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನು ಉಪಯೋಗಿಸಿಕೊಂಡು ಪ್ರೊಗ್ರಾಂನ್ನು ವಿನ್ಯಾಸಗೊಳಿಸುವ ವಿಧಾನ ಅಥವಾ ಮಾದರಿಯಾಗಿದೆ. ಇದು ಕೆಳಗಿನ ಪರಿಕಲ್ಪನೆಗಳನ್ನು ಒದಗಿಸುವ ಮೂಲಕ ಸಾಫ್ಟ್‌ವೇರ್ ಅಭಿವೃದ್ಧಿಯನ್ನು ಮತ್ತು ನಿರ್ವಹಣೆಯನ್ನು ಸರಳಗೊಳಿಸುತ್ತದೆ :

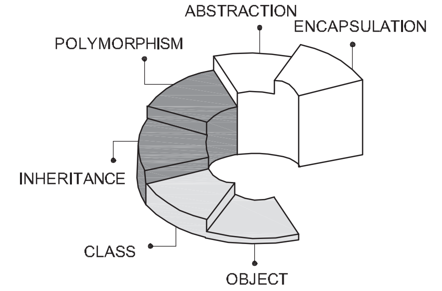
- ಆಬ್ಜೆಕ್ಟ್ (Object)
- ಕ್ಲಾಸ್ (Class)
- ಇನ್‌ಹೆರಿಟನ್ಸ್ (Inheritance)
- ಪಾಲಿಮಾರ್ಫಿಸಮ್ (Polymorphism)
- ಅಬ್‌ಸ್ಟ್ರಾಕ್ಷನ್ (Abstraction)
- ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್ (Encapsulation)

ಈ ಪರಿಕಲ್ಪನೆಗಳ ಹೊರತಾಗಿ, ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ವಿನ್ಯಾಸದಲ್ಲಿ ಇತರ ಕೆಲವು ಪದಗಳನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ :

- ಕಪಲಿಂಗ್ (Coupling)
- ಕೊಹೆಷನ್ (Cohesion)
- ಅಸೋಸಿಯೇಷನ್ Association)
- ಅಗ್ರಿಗೇಷನ್ Aggregation)
- ಕಂಪೋಸಿಷನ್ (Composition)

ಜಾವಾ OOPs ಪರಿಕಲ್ಪನೆಗಳು (Java OOPs Concepts)

Fig 1



© NIMI NOT TO BE REPUBLISHED

ಆಬ್ಜೆಕ್ಟ್ (Object)

ಸ್ಥಿತಿ (state) ಮತ್ತು ನಡುವಳಿಕೆಯನ್ನು (behavior) ಹೊಂದಿರುವ ಯಾವುದೇ ಘಟಕ (entity) ವನ್ನು ಆಬ್ಜೆಕ್ಟ್ ಎನ್ನುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ ಕುರ್ಚಿ, ಪೆನ್, ಟೇಬಲ್, ಕೀಬೋರ್ಡ್, ಬೈಕ್ ಇತ್ಯಾದಿಗಳು.ಇದು ಭೌತಿಕವಾಗಿರಬಹುದು ಅಥವಾ ತಾರ್ಕಿಕವಾಗಿರಬಹುದು.

ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಒಂದು ಕ್ಲಾಸ್‌ನ ನಿದರ್ಶನ ಎಂದು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ಆಬ್ಜೆಕ್ಟ್ ಒಂದು ಅಡ್ರೆಸ್‌ನ್ನು ಹೊಂದಿರುತ್ತದೆ ಮತ್ತು ಮೆಮೊರಿಯಲ್ಲಿ ಸ್ವಲ್ಪ ಸ್ಥಳವನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ. ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಪರಸ್ಪರರ ಡಾಟಾ ಅಥವಾ ಕೋಡ್‌ನ ವಿವರಗಳನ್ನು ತಿಳಿಯದೆಯೆ ಸಂವಹನ ಮಾಡಬಹುದು. ಆದರೆ ಅತ್ಯಂತ ಅಗತ್ಯವಾದ ವಿಷಯಗಳೆಂದರೆ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಸ್ವೀಕರಿಸಿದ ಸಂದೇಶದ ವಿಧ ಮತ್ತು ಅವುಗಳಿಂದ ಹಿಂತಿರುಗಿಸಲ್ಪಟ್ಟ ಪ್ರತಿಕ್ರಿಯೆಯ ವಿಧ.

ಉದಾಹರಣೆ : ನಾಯಿ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಆಗಿದೆ ಏಕೆಂದರೆ ಅದು ಬಣ್ಣ, ಹೆಸರು, ತಳಿ ಮುಂತಾದ ಸ್ಥಿತಿಗಳನ್ನು ಹೊಂದಿದೆ ಮತ್ತು ಬಾಲವನ್ನು ಅಲ್ಲಾಡಿಸುವುದು , ಬೊಗಳುವುದು , ತಿನ್ನುವುದು ಮುಂತಾದ ನಡುವಳಿಕೆಯನ್ನು ಹೊಂದಿದೆ.

ಕ್ಲಾಸ್ (Class)

ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಸಂಗ್ರಹವನ್ನು ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಒಂದು ತಾರ್ಕಿಕ ಘಟಕವಾಗಿದೆ.

ಕ್ಲಾಸ್ ಅನ್ನು ಒಂದು ಸ್ವತಂತ್ರ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸಬಲ್ಲ ಬ್ಲೂಪ್ರಿಂಟ್ ಎಂದೂ ಸಹ ನೀವು ವ್ಯಾಖ್ಯಾನಿಸಬಹುದು. ಕ್ಲಾಸ್ ಯಾವುದೇ ಸ್ಥಳವನ್ನು ಬಳಸುವುದಿಲ್ಲ.

ಇನ್‌ಹೆರಿಟನ್ಸ್ (Inheritance)

ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಪೇರೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್‌ನ ಎಲ್ಲಾ ಗುಣಲಕ್ಷಣಗಳು ಮತ್ತು ನಡುವಳಿಕೆಗಳನ್ನು ಪಡೆದುಕೊಂಡಾಗ, ಅದನ್ನು ಅನುವಂಶೀಯತೆ (inheritance) ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಕೋಡ್‌ನ ಮರುಬಳಕೆಯನ್ನು ಅನುಮತಿಸುತ್ತದೆ. ಇದನ್ನು ರನ್ ಟೈಮ್ ಪಾಲಿ ಮಾರ್ಫಿಸಮ್ ಅನ್ನು ಸಾಧಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಪಾಲಿಮಾರ್ಫಿಸಮ್ (Polymorphism)

ಒಂದು ಕೆಲಸವನ್ನು ವಿವಿಧ ರೀತಿಯಲ್ಲಿ ನಿರ್ವಹಿಸಿದರೆ, ಅದನ್ನು ಪಾಲಿಮಾರ್ಫಿಸಮ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ: ಏನನ್ನಾದರೂ ಚಿತ್ರಬಿಡಿಸಲು ಗ್ರಾಹಕರನ್ನು ವಿಭಿನ್ನವಾಗಿ ಮನವೊಲಿಸುವುದು ಉದಾಹರಣೆಗೆ : ಆಕಾರ, ತ್ರಿಕೋನ, ಆಯತ ಇತ್ಯಾದಿ.

ಜಾವಾದಲ್ಲಿ ಪಾಲಿಮಾರ್ಫಿಸಮ್ ಅನ್ನು ಸಾಧಿಸಲು ನಾವು ಮೆಥೆಡ್ ಓವರ್‌ಲೋಡಿಂಗ್ ಮತ್ತು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್ ಅನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.

ಇನ್ನೊಂದು ಉದಾಹರಣೆ, ಏನಾದರೂ ಮಾತನಾಡುವುದು ಆಗಿರಬಹುದು; ಉದಾಹರಣೆಗೆ ಒಂದು ಬೆಕ್ಕು ಮಿಯಾವ್ ಎನ್ನುತ್ತದೆ , ನಾಯಿ ಬೊಗಳುತ್ತದೆ ಇತ್ಯಾದಿ.

ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಷನ್ (Abstraction)

ಆಂತರಿಕ ವಿವರಗಳನ್ನು ಮರೆಮಾಡುವುದು ಮತ್ತು ಕ್ರಿಯಾತ್ಮಕತೆಯನ್ನು ತೋರಿಸುವುದನ್ನು ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ , ಫೋನ್ ಕರೆ - ನಮಗೆ ಇದರ ಆಂತರಿಕ ಪ್ರಕ್ರಿಯೆ ಗೊತ್ತಿಲ್ಲ.

ಜಾವಾದಲ್ಲಿ, ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಷನ್‌ನ್ನು ಸಾಧಿಸಲು ನಾವು ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಕ್ಲಾಸ್ ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.

ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್ (Encapsulation)

ಕೋಡ್ ಮತ್ತು ಡಾಟಾವನ್ನು ಒಟ್ಟಿಗೆ ಒಂದೇ ಘಟಕವನ್ನಾಗಿ ಬಂಧಿಸುವುದಕ್ಕೆ (ಅಥವಾ ಸುತ್ತುವುದು) ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ ಕ್ಯಾಪ್ಸೂಲ್ - ಇದನ್ನು ವಿವಿಧ ಔಷಧಿಗಳೊಂದಿಗೆ ಸುತ್ತುವರೆಯಲಾಗಿರುತ್ತದೆ.

ಜಾವಾ ಕ್ಲಾಸ್, ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್‌ನ ಉದಾಹರಣೆಯಾಗಿದೆ. ಜಾವಾ ಬೀನ್ (Java bean) ಸಂಪೂರ್ಣ ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಟ್ ಆದ ಕ್ಲಾಸ್ ಆಗಿದೆ. ಏಕೆಂದರೆ ಇದರಲ್ಲಿ ಎಲ್ಲಾ ಡಾಟಾ ಮೆಂಬರ್‌ಗಳು private ಆಗಿರುತ್ತವೆ.

ಕಪಲಿಂಗ್(Coupling)

ಕಪಲಿಂಗ್ ಎಂದರೆ ಜ್ಞಾನ (knowledge), ಅಥವಾ ಮಾಹಿತಿ (information) ಅಥವಾ ಮತ್ತೊಂದು ಕ್ಲಾಸ್‌ನ ಅವಲಂಬನೆ. ಇದು ಕ್ಲಾಸ್‌ಗಳು ಪರಸ್ಪರ ಜಾಗೃತವಾಗಿದ್ದರೆ ಮಾತ್ರ ಸಾಧ್ಯವಾಗುತ್ತದೆ. ಒಂದು ಕ್ಲಾಸ್ ಇನ್ನೊಂದು ಕ್ಲಾಸ್‌ನ ವಿವರವಾದ ಮಾಹಿತಿಯನ್ನು ಹೊಂದಿದ್ದರೆ, ಆಗ ಬಲವಾದ ಕಪಲಿಂಗ್ ಸಾಧ್ಯವಾಗುತ್ತದೆ. ಜಾವಾದಲ್ಲಿ ನಾವು private, protected, ಮತ್ತು public ಮಾಡಿಫಾಯರ್‌ಗಳನ್ನು ಉಪಯೊಗಿಸುತ್ತೇವೆ. ಇವು ಕ್ಲಾಸ್, ಮೆಥೆಡ್ ಮತ್ತು ಫೀಲ್ಡ್‌ನ ಗೋಚರತೆಯ ಮಟ್ಟವನ್ನು ಪ್ರದರ್ಶಿಸುತ್ತವೆ. ದುರ್ಬಲ ಕಪಲಿಂಗ್‌ಗಳಿಗಾಗಿ ನೀವು ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ಏಕೆಂದರೆ ಇಲ್ಲಿ ವಾಸ್ತವಿಕವಾದ ಅನುಷ್ಠಾನ ಇರುವುದಿಲ್ಲ.

ಕೊಹೆಷನ್ (Cohesion)

ಒಂದೇ ಉತ್ತಮವಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುವ ಘಟಕದ ಮಟ್ಟವನ್ನು ಸೂಚಿಸುವುದನ್ನು ಕೊಹೆಶನ್ ಎನ್ನುತ್ತಾರೆ. ಒಂದು ಉನ್ನತಮಟ್ಟದ ಕೊಹೆಸಿವ ಮೆಥೆಡ್, ಉತ್ತಮವಾಗಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಒಂದೇ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುತ್ತದೆ. ದುರ್ಬಲ ಕೊಹೆಸಿವ ಮೆಥೆಡ್ ಒಂದು ಕಾರ್ಯವನ್ನು ಎರಡು ಪ್ರತ್ಯೇಕ ಭಾಗಗಳಲ್ಲಿ ವಿಭಜಿಸುತ್ತದೆ. java.io ಪ್ಯಾಕೇಜ್ ಅತ್ಯಂತ ಪ್ರಬಲ

© NIMI NOT TO BE REPUBLISHED

ಕೊಹೆಸಿವ್ ಪ್ಯಾಕೆಜ್ ಆಗಿದೆ ಏಕೆಂದರೆ ಇದು I/O ಗೆ ಸಂಬಂಧಿಸಿದ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. ಅದಾಗ್ಯೂ java.util ಪ್ಯಾಕೆಜ್ ಒಂದು ದುರ್ಬಲ ಪ್ಯಾಕೆಜ್ ಆಗಿದೆ ಏಕೆಂದರೆ ಇದು ಸಂಬಂಧವಿಲ್ಲದ ಕ್ಲಾಸ್‌ಗಳನ್ನು ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ಗಳನ್ನು ಹೊಂದಿದೆ.

ಅಸೋಸಿಯೇಷನ್ (Association)

ಅಸೋಸಿಯೇಷನ್, ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ನಡುವಿನ ಸಂಬಂಧವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇಲ್ಲಿ ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು, ಇನ್ನೊಂದು ಆಬ್ಜೆಕ್ಟ್ ಅಥವಾ ಒಂದಕ್ಕಿಂತ ಹೆಚ್ಚು ಆಬ್ಜೆಕ್ಟ್‌ಗಳೊಂದಿಗೆ ಸಂಯೋಜಿಸಬಹುದು. ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ನಡುವೆ ನಾಲ್ಕು ರೀತಿಯ ಅಸೋಸಿಯೇಷನ್‌ಗಳು ಇರಬಹುದು.

- One to One
- One to Many
- Many to One, ಮತ್ತು
- Many to Many

ಸಂಬಂಧಗಳನ್ನು ನೈಜ ಸಮಯದ ಉದಾಹರಣೆಗಳೊಂದಿಗೆ ತಿಳಿದುಕೊಳ್ಳೋಣ.

ಉದಾಹರಣೆಗೆ, ಒಂದು ದೇಶವು ಒಬ್ಬ ಪ್ರಧಾನಮಂತ್ರಿಯನ್ನು ಹೊಂದಬಹುದು (one to one) , ಒಬ್ಬ ಪ್ರಧಾನಮಂತ್ರಿಯು ಅನೇಕ ಮಂತ್ರಿಗಳನ್ನು ಹೊಂದಬಹುದು (one to many). ಅಲ್ಲದೇ ಅನೇಕ ಸಂಸದರು (MP ಗಳು) ಒಬ್ಬ ಪ್ರಧಾನಮಂತ್ರಿಯನ್ನು ಹೊಂದಬಹುದು (Many to one) , ಮತ್ತು ಅನೇಕ ಮಂತ್ರಿಗಳು ಅನೇಕ ಇಲಾಖೆಗಳನ್ನು ಹೊಂದಬಹುದು(many to many).

ಅಗ್ರಿಗೇಷನ್ (Aggregation)

ಅಗ್ರಿಗೇಷನ್, ಅಸೋಸಿಯೇಷನ್‌ನ್ನು ಸಾಧಿಸುವ ಮಾರ್ಗವಾಗಿದೆ. ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಇತರ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಅದರ ಸ್ಥಿತಿಯ ಒಂದು ಭಾಗವಾಗಿ ಒಳಗೊಂಡಿರುವ ಸಂಬಂಧವನ್ನು ಅಗ್ರಿಗೇಷನ್ ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇದು ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ನಡುವಿನ ದುರ್ಬಲ ಸಂಬಂಧವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಜಾವಾದಲ್ಲಿ ಇದನ್ನು HAS - A ರಿಲೇಷನ್‌ಶಿಪ್ ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ. ಹಾಗೆ ಇನ್‌ಹೆರಿಟನ್ಸ್ IS - A ರಿಲೇಷನ್‌ಶಿಪ್ ಪ್ರತಿನಿಧಿಸುತ್ತದೆ.

ಇದು ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ಮರುಬಳಕೆ ಮಾಡುವ ಇನ್ನೊಂದು ಮಾರ್ಗವಾಗಿದೆ.

ಕಾಂಪೋಸಿಷನ್(Composition)

ಕಾಂಪೊಸಿಷನ್ ಸಹ , ಅಸೋಸಿಯೇಷನ್ ಅನ್ನು ಸಾಧಿಸುವ ಮಾರ್ಗವಾಗಿದೆ. ಕಾಂಪೋಸಿಷನ್ ಸಂಬಂಧವನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇದರಲ್ಲಿ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ಇತರ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ತನ್ನ ಸ್ಥಿತಿಯ ಒಂದು ಭಾಗವನ್ನಾಗಿ ಹೊಂದಿರುತ್ತದೆ. ಇಲ್ಲಿ ಒಳಗೊಂಡಿರುವ ಆಬ್ಜೆಕ್ಟ್ ಮತ್ತು ಅವಲಂಬಿತ ಆಬ್ಜೆಕ್ಟ್ ಎರಡರ ನಡುವೆ ಬಲವಾದ ಸಂಬಂಧ ಇರುತ್ತದೆ. ಇದು, ಒಳಗೊಂಡಿರುವ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಸ್ವತಂತ್ರ್ಯವಾದ ಅಸ್ತಿತ್ವವನ್ನು ಹೊಂದದೇ ಇರುವ ಒಂದು ಸ್ಥಿತಿಯಾಗಿದೆ. ನೀವು ಪೇರೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಡಿಲೀಟ್ ಮಾಡಿದರೆ, ಎಲ್ಲಾ ಚೈಲ್ಡ್ ಆಬ್ಜೆಕ್ಟ್‌ಗಳು ಸ್ವಯಂ ತಾನೇ ಡಿಲೀಟ್ ಆಗುತ್ತವೆ.

ಪ್ರೊಸಿಜರ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ಗೆ ಹೋಲಿಸಿದರೆ OOPs ನ ಅನುಕೂಲಗಳು

1 OOPs ಅಭಿವೃದ್ಧಿಯನ್ನು ಮತ್ತು ನಿರ್ವಹಣೆಯನ್ನು ಸುಲಭಗೊಳಿಸುತ್ತವೆ. ಆದರೆ ಪ್ರೊಸಿಜರ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಲ್ಲಿ ಕೋಡ್ ಹೆಚ್ಚಾದರೆ ಪ್ರೊಜೆಕ್ಟ್‌ನ ಗಾತ್ರ ಹೆಚ್ಚಾಗುತ್ತದೆ ಮತ್ತು ಅದನ್ನು ನಿರ್ವಹಿಸುವುದು ಸುಲಭವಾಗಿರುವುದಿಲ್ಲ.

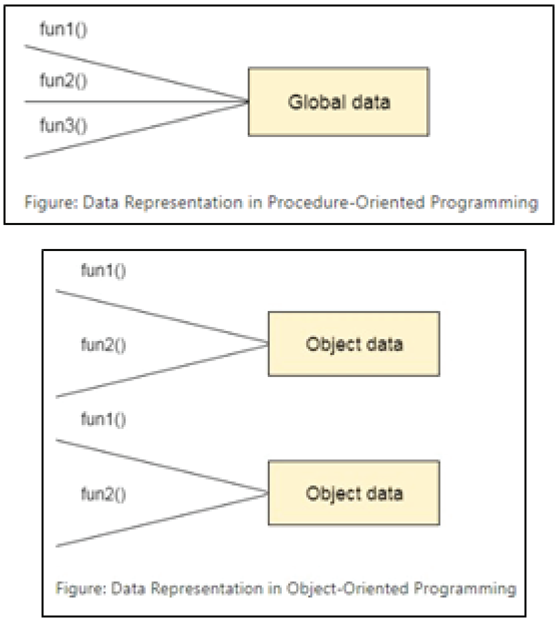
2 OOPs ಡಾಟಾ ಮರೆಮಾಡುವುದನ್ನು(ಡಾಟಾ ಹೈಡಿಂಗ್) ಒದಗಿಸುತ್ತದೆ, ಆದರೆ ಪ್ರೊಸಿಜರ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯಲ್ಲಿ ಗ್ಲೋಬಲ್ ಡಾಟಾವನ್ನು ಎಲ್ಲಿಂದಾದರೂ ಪಡೆದುಕೊಳ್ಳಬಹುದು.

3 OOPs ನೈಜ ಜಗತ್ತಿನ ಘಟನೆಗಳನ್ನು ಹೆಚ್ಚು ಪರಿಣಾಮಕಾರಿಯಾಗಿ ಅನುಕರಿಸುವ ಸಾಮರ್ಥ್ಯವನ್ನು ಒದಗಿಸುತ್ತದೆ. ನಾವು ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯನ್ನು ಬಳಸುತ್ತಿದ್ದರೆ, ನೈಜ ಜಗತ್ತಿನ ಸಮಸ್ಯೆಗಳಿಗೆ ಪರಿಹಾರವನ್ನು ಒದಗಿಸಬಹುದು.

ಗ್ಲೋಬಲ್ ಡಾಟಾ (Global Data)

ಚಿತ್ರ : ಪ್ರೊಸೀಜರ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ (Fig 2)

Fig 2



© NIMI NOT TO BE REPUBLISHED

ಆಬ್ಜೆಕ್ಟ್ ಡಾಟಾ (Object Data)

ಆಬ್ಜೆಕ್ಟ್ ಓರಿಯಂಟೆಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆ ಮತ್ತು ಆಬ್ಜೆಕ್ಟ್ - ಬೇಸ್ಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆ ಎರಡರ ನಡುವಿನ ವ್ಯತ್ಯಾಸವೇನು?

ಆಬ್ಜೆಕ್ಟ್ - ಬೇಸ್ಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯು OOPs ನಲ್ಲಿರುವ ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನ್ನು ಹೊರತುಪಡಿಸಿ ಎಲ್ಲಾ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಅನುಸರಿಸುತ್ತದೆ. ಜಾವಾಸ್ಕ್ರಿಪ್ಟ್ ಮತ್ತು VB ಸ್ಕ್ರಿಪ್ಟ್ ಇವು ಆಬ್ಜೆಕ್ಟ್ - ಬೇಸ್ಡ್ ಪ್ರೊಗ್ರಾಮಿಂಗ್ ಭಾಷೆಯ ಉದಾಹರಣೆಗಳಾಗಿವೆ.

ನಿಮಗೆ ಗೊತ್ತೇ?

- main ಮೆಥೆಡ್‌ನ್ನು ಓವರ್‌ಲೋಡ್ ಮಾಡಬಹುದೇ?
- ಜಾವಾ ಕನ್‌ಸ್ಟ್ರಕ್ಟರ್ ಒಂದು ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ ಆದರೆ ಏನು?
- main ಮೆಥೆಡ್ ಇಲ್ಲದೇ ನಾವು ಪ್ರೊಗ್ರಾಂ ಅನ್ನು ರಚಿಸಬಹುದೇ?
- this ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸುವ ಆರು ಮಾರ್ಗಗಳು ಯಾವವು .
- ಜಾವಾದಲ್ಲಿ ಮಲ್ಟಿಪಲ್ ಇನ್‌ಹೆರಿಟನ್ಸ್ ನ್ನು ಏಕೆ ಬೆಂಬಲಿಸಲಾಗುವುದಿಲ್ಲ ?
- ಅಗ್ರಿಗೇಷನ್‌ನ್ನು ಏಕೆ ಉಪಯೋಗಿಸಬೇಕು?
- static ಮೆಥೆಡ್‌ನ್ನು ನಾವು ಓವರ್‌ರೈಡ್ ಮಾಡಬಹುದೇ?
- covarient return ಟೈಪ್ ಎಂದರೇನು?
- ಜಾವಾ super ಕೀವರ್ಡ್‌ನ ಮೂರು ಉಪಯೋಗಗಳು ಯಾವವು?
- instance initializer ಬ್ಲಾಕ್‌ನ್ನು ಏಕೆ ಉಪಯೋಗಿಸಬೇಕು?
- blank final ವೇರಿಯೇಬಲ್‌ನ ಉಪಯೋಗವೇನು?
- marker ಅಥವಾ tagged interface ಎಂದರೇನು?
- ರನ್ ಟೈಮ್ ಪಾಲಿಮಾರ್ಫಿಸಮ್ ಅಥವಾ ಡೈನಾಮಿಕ್ ಮೆಥೆಡ್ ಎಂದರೇನು?
- static ಮತ್ತು dynamic ಬೈಂಡಿಂಗ್ ನಡುವಿನ ವ್ಯತ್ಯಾಸವೇನು?
- ಜಾವಾದಲ್ಲಿ down casting ಹೇಗೆ ಸಾಧ್ಯವಿದೆ?
- private constructor ನ ಉದ್ದೇಶವೇನು?
- ಆಬ್ಜೆಕ್ಟ್ ಕ್ಲೋನಿಂಗ್ ಎಂದರೇನು?

OOPs ಪರಿಕಲ್ಪನೆಗಳಲ್ಲಿ ನಾವು ಏನು ತಿಳಿದುಕೊಳ್ಳುತ್ತೇವೆ ?

- OOPs ನ ಅನುಕೂಲಗಳು
- Naming Convention
- ಆಬ್ಜೆಕ್ಟ್ ಮತ್ತು ಕ್ಲಾಸ್
- ಮೆಥೆಡ್ ಓವರ್‌ಲೋಡಿಂಗ್
- ಕನ್‌ಸ್ಟ್ರಕ್ಟರ್
- static ಕೀವರ್ಡ್
- this ಕೀವರ್ಡ್ ಮತ್ತು ಆರು ಉಪಯೋಗಗಳು
- ಇನ್‌ಹೆರಿಟನ್ಸ್
- ಅಗ್ರಿಗೇಷನ್
- ಮೆಥೆಡ್ ಓವರ್‌ಲೋಡಿಂಗ್
- Covariant Return ಟೈಪ್
- super ಕೀವರ್ಡ್
- Instance Initializer ಬ್ಲಾಕ್
- final ಕೀವರ್ಡ್
- Abstract ಕ್ಲಾಸ್
- ಇಂಟರ್‌ಫೇಸ್
- Runtime Polymorphism
- Static ಮತ್ತು Dynamic ಬೈಂಡಿಂಗ್
- Downcasting with instance of operator
- ಪ್ಯಾಕೆಜ್
- Access ಮಾಡಿಫಾಯರ್
- ಎನ್‌ಕ್ಯಾಪ್ಸುಲೇಷನ್

ಆಬ್ಜೆಕ್ಟ್ ಕ್ಲೋನಿಂಗ್ (Object Cloning)

ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್‌ಗಳು (Method in Java)

ಸಾಮಾನ್ಯವಾಗಿ ಕೆಲವು ಕಾರ್ಯಗಳನ್ನು ನಿರ್ವಹಿಸಲು ಮೆಥೆಡ್ ಒಂದು ಮಾರ್ಗವಾಗಿದೆ. ಅದೇ ರೀತಿ ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್ ಎಂದರೆ ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯವನ್ನು ನಿರ್ವಹಿಸುವ ಇನ್‌ಸ್ಟ್ರಕ್ಷನ್‌ಗಳ (ಸೂಚನೆಗಳು) ಸಂಗ್ರಹವಾಗಿದೆ. ಇದು ಕೋಡ್‌ನ ಮರುಬಳಕೆಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಮೆಥೆಡ್‌ಗಳನ್ನು ಬಳಸಿ ನಾವು ಕೋಡ್‌ನ್ನು ಸುಲಭವಾಗಿ ಬದಲಾಯಿಸಬಹುದು. ಈ ವಿಭಾಗದಲ್ಲಿ ನಾವು ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್ ಎಂದರೇನು, ಅದರ ವಿಧಗಳು , ಮೆಥೆಡ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು ಮತ್ತು ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್‌ನ್ನು ಹೇಗೆ ಕರೆಯುವುದು ಎಂಬುದನ್ನು

ತಿಳಿದುಕೊಳ್ಳೋಣ.

ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್ ಎಂದರೇನು?(What is a method in Java?)

ಮೆಥೆಡ್ ಎಂದರೆ ನಿರ್ದಿಷ್ಟ ಕಾರ್ಯವನ್ನು ಅಥವಾ

© NIMI NOT TO BE REPUBLISHED

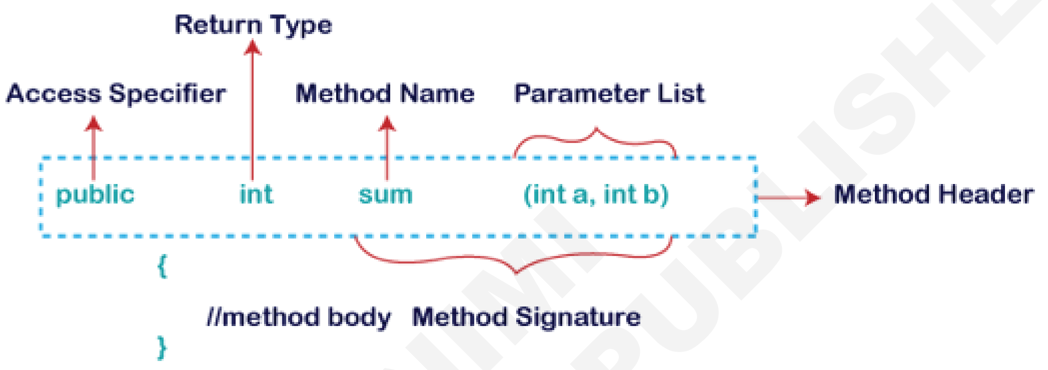
ಕಾರ್ಯಾಚರಣೆಯನ್ನು ಮಾಡುವ ಕೋಡ್‌ಗಳ ಬ್ಲಾಕ್ ಅಥವಾ ಸ್ಟೇಟಮೆಂಟ್‌ಗಳ ಸಂಗ್ರಹ ಅಥವಾ ಒಟ್ಟಿಗೆ ಗುಂಪು ಮಾಡಲಾದ ಕೋಡ್‌ಗಳ ಸೆಟ್ ಆಗಿದೆ. ಇದನ್ನು ಕೋಡ್‌ಗಳ ಮರುಬಳಕೆಯನ್ನು ಸಾಧಿಸಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಎಂದರೆ ಒಂದು ಬಾರಿ ಕೋಡ್‌ನ್ನು ಬರೆಯುವುದು ಮತ್ತು ಹಲವಾರು ಬಾರಿ ಅದನ್ನು ಉಪಯೋಗಿಸಬಹುದು. ನಾವು ಒಂದು ಕೋಡ್ ಅನ್ನು ಮತ್ತೆ ಮತ್ತೆ ಬರೆಯುವ ಅಗತ್ಯವಿಲ್ಲ. ಕೊಡ್‌ನ ಒಂದು ತುಂಡನ್ನು ಸೇರಿಸುವುದರಿಂದ ಅಥವಾ ತೆಗೆದುಹಾಕುವುದರಿಂದ ಕೋಡ್‌ನ್ನು ಮಾರ್ಪಾಡು ಮಾಡುವುದನ್ನು ಮತ್ತು ಓದುವುದನ್ನು ಸಹ ಸುಲಭವಾಗಿಸುತ್ತದೆ. ಮೆಥೆಡ್‌ನ್ನು ಕಾಲ್ ಮಾಡಿದಾಗ ಅಥವಾ ಇನ್ವೋಕ್ ಮಾಡಿದಾಗ ಮಾತ್ರ ಅದು ಕಾರ್ಯಗತಗೊಳ್ಳುತ್ತದೆ.

ಜಾವಾದಲ್ಲಿ ಅತ್ಯಂತ ಮುಖ್ಯವಾದ ಮೆಥೆಡ್ main() ಮೆಥೆಡ್ ಆಗಿದೆ. main() ಮೆಥೆಡ್ ಬಗ್ಗೆ ಇನ್ನೂ ಹೆಚ್ಚು ತಿಳಿದುಕೊಳ್ಳಲು https://www.javatpoint.com/java-mainmethod ಈ ಲಿಂಕ್‌ನ್ನು ಉಪಯೋಗಿಸಿರಿ.

ಮೆಥೆಡ್‌ನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡುವುದು (Method Declaration) (Fig 3)

ಮೆಥೆಡ್ ಡಿಕ್ಲೇರೇಷನ್ ಇದು ಮೆಥೆಡ್ ಆಟ್ರಿಬ್ಯೂಟ್‌ಗಳಾದ visibility, return-type, name, ಮತ್ತು arguments ಇವುಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿಯನ್ನು ಒದಗಿಸುತ್ತದೆ. ಇದು ಮೆಥೆಡ್ ಹೆಡರ್ ಎಂದು ಕರೆಯಲ್ಪಡುವ ಆರು ಘಟಕಗಳನ್ನು ಹೊಂದಿದೆ.

Fig 3



© NIMI NOT TO BE REPUBLISHED

ಮೆಥೆಡ್ ಸಿಗ್ನೆಚರ್ (Method Signature) : ಪ್ರತಿಯೊಂದು ಮೆಥೆಡ್, ಮೆಥೆಡ್ ಸಿಗ್ನೆಚರ್‌ನ್ನು ಹೊಂದಿದೆ. ಇದು ಮೆಥೆಡ್ ಡಿಕ್ಲೇರೇಷನ್‌ನ ಒಂದು ಭಾಗವಾಗಿದೆ. ಇದು ಮೆಥೆಡ್ ಹೆಸರನ್ನು ಮತ್ತು ಪ್ಯಾರಾಮೀಟರ್ ಪಟ್ಟಿಯನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ.

ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಯರ್ (Access Specifier): ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್ ಅಥವಾ ಮಾಡಿಫಾಯರ್, ಮೆಥೆಡ್‌ನ ಒಂದು ಎಕ್ಸೆಸ್ ವಿಧವಾಗಿದೆ. ಇದು ಮೆಥೆಡ್‌ನ ವಿಸಿಬಿಲಿಟಿ (ಗೋಚರತೆ)ಯನ್ನು ಸೂಚಿಸುತ್ತದೆ. ಜಾವಾ ನಾಲ್ಕು ರೀತಿಯ ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್‌ಗಳನ್ನು ಒದಗಿಸುತ್ತದೆ.

- Public : ನಾವು ನಮ್ಮ ಅಪ್ಲಿಕೇಷನ್‌ನಲ್ಲಿ public ಸ್ಪೆಸಿಫಾಯರ್‌ನ್ನು ಉಪಯೋಗಿಸಿದಾಗ, ಮೆಥೆಡ್ ಎಲ್ಲಾ ಕ್ಲಾಸ್‌ಗಳಿಂದ ಉಪಯೊಗಿಸಲ್ಪಡುತ್ತದೆ.
- Private : ನಾವು Private ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್ ಅನ್ನು ಉಪಯೋಗಿಸಿದಾಗ, ಮೆಥೆಡ್‌ನ್ನು ಯಾವ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆಯೋ ಅಲ್ಲಿ ಮಾತ್ರ ಅದನ್ನು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.
- Protected : ನಾವು Protected ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್ ಅನ್ನು ಉಪಯೋಗಿಸಿದಾಗ ಅದನ್ನು ಅದೇ ಪ್ಯಾಕೇಜ್‌ನಲ್ಲಿ ಮಾತ್ರ ಉಪಯೋಗಿಸಬಹುದು ಅಥವಾ ವಿಭಿನ ಪ್ಯಾಕೇಜ್‌ನ ಸಬ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಬಹುದು.
- Default : ನಾವು ಮೆಥೆಡ್ ಡಿಕ್ಲೇರೇಷನ್‌ನಲ್ಲಿ ಯಾವುದೇ ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್‌ನ್ನು ಉಪಯೋಗಿಸದಿದ್ದರೆ, ಡೀಫಾಲ್ಟ್ ಆಗಿ ಜಾವಾ ಡಿಫಾಲ್ಟ್ ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್ ಅನ್ನು ಉಪಯೋಗಿಸುತ್ತದೆ. ಇದು ಅದೇ ಪ್ಯಾಕೇಜ್‌ನಿಂದ ಮಾತ್ರ ಗೋಚರವಾಗುತ್ತದೆ.
- Return Type : ರಿಟರ್ನ್ ಟೈಪ್ ಇದು ಮೆಥೆಡ್ ಹಿಂತಿರುಗಿಸುವ ಒಂದು ಡಾಟಾಟೈಪ್ ಆಗಿದೆ. ಇದು ಪ್ರಿಮಿಟಿವ್ ಡಾಟಾಟೈಪ್‌ಗಳಾದ object, collection,void ಇತ್ಯಾದಿಗಳು ಆಗಿರಬಹುದು. ಒಂದು ವೇಳೆ ಮೆಥೆಡ್ ಏನನ್ನೂ ಹಿಂತಿರುಗಿಸದಿದ್ದರೆ, ನಾವು void ಅನ್ನು ಉಪಯೋಗಿಸುತ್ತೇವೆ.
- Method Name : ಇದು ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲು ಬಳಸಲಾಗುವ ಒಂದು ವಿಶಿಷ್ಟ ಹೆಸರು ಆಗಿದೆ. ಇದು ಮೆಥೆಡ್‌ನ ಕ್ರಿಯಾತ್ಮಕತೆಗೆ ಅನುರೂಪವಾಗಿರಬೇಕು. ನಾವು ಎರಡು ಸಂಖ್ಯೆಗಳ ವ್ಯವಕಲನಕ್ಕಾಗಿ ಒಂದು ಮೆಥೆಡ್‌ನ್ನು ರಚಿಸುತ್ತಿದ್ದೇವೆ ಎಂದು ಉಹಿಸಿಕೊಳ್ಳಿ, ಆಗ ಮೆಥೆಡ್‌ನ ಹೆಸರು Subtraction() ಎಂದು ಆಗಿರಬೇಕು. ಮೆಥೆಡ್‌ನ್ನು ಅದರ ಹೆಸರಿನಿಂದ ಕರೆಯಲಾಗುತ್ತದೆ.(invoke)

- Parameter List: ಇದು ಅಲ್ಪವಿರಾಮದಿಂದ ಬೇರ್ಪಡಿಸಲ್ಪಟ್ಟ ಪ್ಯಾರಾಮಿಟರ್‌ಗಳ ಪಟ್ಟಿಯಾಗಿದೆ ಮತ್ತು ಇದನ್ನು ಆವರಣ(parentheses) ದಲ್ಲಿ ಇರಿಸಲಾಗುತ್ತದೆ. ಇದು ಡಾಟಾಟೈಪ್ ಮತ್ತು ವೇರಿಯೇಬಲ್ ಹೆಸರನ್ನು ಹೊಂದಿರುತ್ತದೆ. ಮೆಥೆಡ್ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಹೊಂದಿರದಿದ್ದರೆ, ಆವರಣವನ್ನು ಖಾಲಿ ಬಿಡಲಾಗುತ್ತದೆ.

- Method Body : ಇದು ಮೆಥೆಡ್ ಡಿಕ್ಲೇರೆಷನ್‌ನ ಒಂದು ಭಾಗವಾಗಿದೆ. ಇದು ನಿರ್ವಹಿಸಬೇಕಾದ ಎಲ್ಲಾ ಕ್ರಿಯೆಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ. ಇದು ಕರ್ಲೀ ಬ್ರೇಸ್‌ಗಳಿಂದ ಸುತ್ತುವರೆಯಲ್ಪಟ್ಟಿರುತ್ತದೆ.

ಮೆಥೆಡ್‌ನ್ನು ಹೆಸರಿಸುವುದು (Naming a Metheod)

ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವಾಗ, ಮೆಥೆಡ್‌ನ ಹೆಸರು ಕ್ರಿಯಾಪದವಾಗಿರಬೇಕು ಮತ್ತು ಸಣ್ಣ ಅಕ್ಷರ (lowercase) ದಿಂದ ಪ್ರಾರಂಭವಾಗಬೇಕು. ಮೆಥೆಡ್‌ನ ಹೆಸರು ಎರಡು ಶಬ್ದಗಳಿಗಿಂತಲೂ ಹೆಚ್ಚಾಗಿದ್ದರೆ ಮೊದಲನೇ ಹೆಸರು ಕ್ರಿಯಾಪದವಾಗಿರಬೇಕು ನಂತರ ವಿಶೇಷಣ ಅಥವಾ ನಾಮಪದವಾಗಿರಬೇಕು. ಮೆಥೆಡ್ ಹೆಸರಿನಲ್ಲಿ ಹೆಚ್ಚು ಶಬ್ದಗಳಿದ್ದರೆ, ಮೊದಲನೇ ಶಬ್ದವನ್ನು ಹೊರತುಪಡಿಸಿ , ಉಳಿದ ಪ್ರತಿ ಶಬ್ದದ ಮೊದಲ ಅಕ್ಷರ ದೊಡ್ಡದಾಗಿರಬೇಕು (uppercase).

ಒಂದೇ ಶಬ್ದದ ಮೆಥೆಡ್ ಹೆಸರುಗಳು :sum(), area()

ಬಹು ಶಬ್ದಗಳ ಮೆಥೆಡ್ ಹೆಸರುಗಳು:areaOfCircle(), stringComparision()

ಒಂದೇ ಕ್ಲಾಸ್‌ನಲ್ಲಿ, ಒಂದು ಮೆಥೆಡ್ ಇನ್ನೊಂದು ಮೆಥೆಡ್‌ನ ಅದೇ ಹೆಸರನ್ನು ಹೊಂದಿರುವ ಸಾಧ್ಯತೆ ಇರುತ್ತದೆ. ಇದನ್ನು ಮೆಥೆಡ್ ಓವರ್‌ಲೋಡಿಂಗ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಮೆಥೆಡ್‌ನ ವಿಧಗಳು (Types of Method)

ಜಾವಾದಲ್ಲಿ ಎರಡು ವಿಧದ ಮೆಥೆಡ್‌ಗಳಿವೆ :

- Predefined Method (ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್)
- User-defined Method (ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್)

Predefined Method (ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್)

ಜಾವಾದಲ್ಲಿ ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ಗಳು ಎಂದರೆ, ಜಾವಾ ಕ್ಲಾಸ್ ಲೈಬ್ರರಿಗಳಲ್ಲಿ ಈಗಾಗಲೇ ಅವುಗಳನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿರುತ್ತದೆ. ಇವುಗಳನ್ನು ಸ್ಟ್ಯಾಂಡರ್ಡ್ ಲೈಬ್ರರಿ ಮೆಥೆಡ್‌ಗಳು ಅಥವಾ ಬಿಲ್ಟ್- ಇನ್ ಮೆಥೆಡ್‌ಗಳು ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ. ನಾವು ಪ್ರೊಗ್ರಾಂನ ಯಾವುದೇ ಭಾಗದಲ್ಲಿ ಇವುಗಳನ್ನು ಕರೆಯುವುದರ ಮೂಲಕ ಅವುಗಳನ್ನು ನೇರವಾಗಿ ಉಪಯೋಗಿಸಬಹುದು. ಕೆಲವು ಪೂರ್ವನಿರ್ಧಾರಿತ ಮೆಥೆಡ್‌ಗಳು ಹೀಗಿವೆ : length(), equals(), compareTo(), sqrt() ಇತ್ಯಾದಿಗಳು. ನಾವು ನಮ್ಮ ಪ್ರೊಗ್ರಾಂನಲ್ಲಿ ಯಾವುದಾದರೂ ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ನ್ನು ಕರೆದಾಗ, ಈಗಾಗಲೇ ಲೈಬ್ರರಿಯಲ್ಲಿ ಸಂಗ್ರಹಿಸಲಾದ ಮೆಥೆಡ್‌ಗೆ ಅನುಗುಣವಾಗಿ ಸಂಬಂಧಿಸಿದ ಕೋಡ್‌ಗಳ ಸರಣಿ ಹಿನ್ನೆಲೆಯಲ್ಲಿ ರನ್ ಆಗುತ್ತದೆ.

ಪ್ರತಿಯೊಂದು ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ನ್ನು ಕ್ಲಾಸ್‌ನಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ. ಉದಾಹರಣೆಗೆ print() ಮೆಥೆಡ್‌ನ್ನು, java. io.PrintStream ಕ್ಲಾಸ್‌ನಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾಗಿದೆ. ಇದು ನಾವು ಮೆಥೆಡ್‌ನಲ್ಲಿ ಬರೆದ ಸ್ಟೇಟಮೆಂಟ್‌ನ್ನು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ. ಉದಾಹರಣೆಗೆ print("Java"), ಇದು Java ಅನ್ನು ಕನ್ಸೋಲ್‌ನಲ್ಲಿ ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ.

ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ನ ಉದಾಹರಣೆಯನ್ನು ನೋಡೋಣ.

Demo.java

```
public class Demo
{
public static void main(String[]args)
{
//using the max() method of Math class
System.out.print ("The maximum number is: " + Math.max(9,7));
}
}
```

ಫಲಿತಾಂಶ :

The maximum number is:9

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ನಾವು ಮೂರು ಪೂರ್ವ ನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ಗಳನ್ನು ಉಪಯೋಗಿಸಿದ್ದೇವೆ - main(), print(), ಮತ್ತು max(). ನಾವು ಈ ಮೆಥೆಡ್‌ಗಳನ್ನು ಡಿಕ್ಲೇರ್ ಮಾಡದೆಯೆ ನೇರವಾಗಿ ಉಪಯೋಗಿಸಿದ್ದೇವೆ ಏಕೆಂದರೆ ಅವು ಪೂರ್ವನಿರ್ಧರಿತವಾಗಿವೆ (predefined). print() ಮೆಥೆಡ್, PrintStream ಕ್ಲಾಸ್‌ನ ಮೆಥೆಡ್ ಆಗಿದೆ. ಇದು ಉತ್ತರವನ್ನು ಕನ್ಸೋಲ್ ಮೇಲೆ ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ. max() ಮೆಥೆಡ್ ಇದು Math ಕ್ಲಾಸ್‌ನ ಮೆಥೆಡ್ ಆಗಿದೆ. ಇದು ಎರಡು ಸಂಖ್ಯೆಗಳಲ್ಲಿ ಗರಿಷ್ಟ ಸಂಖ್ಯೆಯನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

ನಾವು ಯಾವುದೇ ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ನ ಮೆಥೆಡ್ ಸಿಗ್ನೇಚರ್‌ನ್ನು https://docs.oracle.com/ ಈ ಲಿಂಕ್ ಮೇಲೆ ಕ್ಲಿಕ್ ಮಾಡುವುದರ ಮೂಲಕ ನೋಡಬಹುದು. ನಾವು ಲಿಂಕ್‌ನ್ನು ತೆರೆದಾಗ ಮತ್ತು max() ಮೆಥೆಡ್ ಸಿಗ್ನೇಚರ್ ನೋಡಿದಾಗ ಕೆಳಗಿನ ಮಾಹಿತಿಯನ್ನು ತಿಳಿದುಕೊಳ್ಳಬಹುದು:

NOT TO BE REPUBLISHED © NIMI

ಮೇಲಿನ ಮೆಥೆಡ್ ಸಿಗ್ನೇಚರ್‌ನಲ್ಲಿ - ಮೆಥೆಡ್ ಸಿಗ್ನೇಚರ್, ಎಕ್ಸೆಸ್ ಸ್ಪೆಸಿಫಾಯರ್ public ಎಂದು, ನಾನ್ ಎಕ್ಸೆಸ್ ಮಾಡಿಫಾಯರ್ static ಎಂದು, ಮೆಥೆಡ್ ಹೆಸರು max() ಎಂದು , ಪ್ಯಾರಾಮೀಟರ್ ಲಿಸ್ಟ್ (int a , int b) ಇವುಗಳನ್ನು ಹೊಂದಿದೆ ಎಂಬುದನ್ನು ನೋಡುತ್ತೇವೆ. ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸುವ ಬದಲು ನಾವು ಮೆಥೆಡ್‌ನ್ನು invoke(ಕರೆಯುವುದು) ಮಾಡಿದ್ದೇವೆ. ಇದು ಪೂರ್ವನಿರ್ಧರಿತ ಮೆಥೆಡ್‌ನ ಪ್ರಯೋಜವಾನವಾಗಿದೆ. ಇದು ಪ್ರೊಗ್ರಾಂನ ಸಂಕೀರ್ಣತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡುತ್ತದೆ. ಇದೇ ರೀತಿ print() ಮೆಥೆಡ್‌ನ, ಮೆಥೆಡ್ ಸಿಗ್ನೆಚರ್‌ನ್ನು ಸಹ ನಾವು ನೋಡಬಹುದು.

max

```
public static int max(int a,
                      int b)
```

Returns the greater of two int values.
same value.

Parameters:

a - an argument.

b - another argument.

Returns:

the larger of a and b.

ಬಳಕೆದಾರರು - ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್ (User - defined Method)

ಬಳಕೆದಾರರು ಅಥವಾ ಪ್ರೊಗ್ರಾಮರ್ ಬರೆದ ಮೆಥೆಡ್‌ನ್ನು ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಈ ಮೆಥೆಡ್‌ಗಳನ್ನು ಅವಶ್ಯಕತೆಗೆ ತಕ್ಕಂತೆ ಮಾರ್ಪಾಡು ಮಾಡಬಹುದು.

ಬಳಕೆದಾರರು - ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್‌ಗಳನ್ನು ಹೇಗೆ ರಚಿಸುವುದು?(How to CreateaUser-defined Method)

ಒಂದು ಸಂಖ್ಯೆ, ಸಮ ಸಂಖ್ಯೆಯೊ ಅಥವಾ ಬೆಸ ಸಂಖ್ಯೆಯೊ ಎಂದು ಪರಿಶೀಲಿಸಲು ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್‌ನ್ನು ರಚಿಸೋಣ. ಮೊದಲು ನಾವು ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸೋಣ.

```
//user defined method
public static void findEvenOdd(int num)
{
//method bodyif(num%2==0)
System.out.println(num + " is even");else
System.out.println(num+" is odd" );
}
```

ಮೇಲಿನ ಮೆಥೆಡ್‌ನ್ನು findevenodd() ಎನ್ನುವ ಅಹೆಸರಿನಿಂದ ವ್ಯಾಖ್ಯಾನಿಸಿದ್ದೇವೆ. ಇದು int ಡಾಟಾಟೈಪ್ ಇರುವ num ಎನ್ನುವ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಹೊಂದಿದೆ. ಈ ಮೆಥೆಡ್ ಯಾವುದೇ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುವುದಿಲ್ಲ. ಅದ್ದರಿಂದ ನಾವು void ಅನ್ನು ಉಪಯೋಗಿಸಿದ್ದೇವೆ. ಮೆಥೆಡ್‌ನ ಮುಖ್ಯಭಾಗವು ಒಂದು ಸಂಖ್ಯೆ ಸಮಸಂಖ್ಯೆಯಾಗಿದೆಯೆ ಅಥವಾ ಬೆಸಸಂಖ್ಯೆಯಾಗಿದೆಯೆ ಎಂದು ಪರಿಶೀಲಿಸಲು ಸೂಚನೆಗಳನ್ನು ಒಳಗೊಂಡಿದೆ. ಸಂಖ್ಯೆ ಸಮಸಂಖ್ಯೆಯಾಗಿದ್ದರೆ, ಇದು number is even ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ, ಇಲ್ಲದಿದ್ದರೆ number is odd ಎಂದು ಪ್ರಿಂಟ್ ಮಾಡುತ್ತದೆ.

ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್‌ನ್ನು ಹೇಗೆ ಕರೆಯುವುದು ? (How to Call or Invoke a User-defined Method)

ಒಮ್ಮೆ ನಾವು ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೇಲೆ, ಅದನ್ನು ಕರೆಯಬೇಕಾಗುತ್ತದೆ. ಪ್ರೊಗ್ರಾಂನಲ್ಲಿ ಮೆಥೆಡ್‌ನ್ನು ಕರೆಯುವುದು ಸರಳವಾಗಿದೆ. ನಾವು ಬಳಕೆದಾರರು ವ್ಯಾಖ್ಯಾನಿಸಿದ ಮೆಥೆಡ್‌ನ್ನು ಕರೆದಾಗ ಪ್ರೊಗ್ರಾಂನ ನಿಯಂತ್ರಣವು ಕರೆಯಲಾದ ಮೆಥೆಡ್‌ಗೆ ವರ್ಗಾಯಿಸಲ್ಪಡುತ್ತದೆ.

```
import
java.util.Scanner;public
class EvenOdd
{
public static void main(String args[])
{
//creating Scanner class object Scannersc
an=new Scanner(System.in);
System.out.print("Enter the number:");
//reading value from the user
int num=scan.nextInt();
//method calling
find EvenOdd (num);
}
```

ಮೇಲಿನ ಕೋಡ್‌ನಲ್ಲಿ ಯಾವಾಗ ಕಂಪೈಲರ್, find EvenOdd (num) ಸಾಲನ್ನು ತಲುಪುತ್ತದೆಯೊ ಆಗ ನಿಯಂತ್ರಣವು ಮೆಥೆಡ್‌ಗೆ ವರ್ಗಾವಣೆಯಾಗುತ್ತದೆ ಮತ್ತು ಅದಕ್ಕೆ ಅನುಗುಣವಾಗಿ ಉತ್ತರವನ್ನು ನೀಡುತ್ತದೆ.

© NIMI NOT TO BE REPUBLISHED

EvenOdd.java

```
import java.util.Scanner;public
classEvenOdd
{
public static void main(String args[])
{
//creating Scanner class object Scannersc
an=new Scanner(System.in);
System.out.print("Enter the number:");
//reading value from user
int num=scan.nextInt();
//method calling
find EvenOdd (num);
}
//user defined method
public static void find EvenOdd(int num)
{
//method body
if (num%2==0)
System.out.println(num+" is even");
else
System.out.println(num+"is odd");
}
}
```

ಫಲಿತಾಂಶ 1:

```
Enter the number:1212
1212 is even
```

ಫಲಿತಾಂಶ 2:

```
Enter the number:99
99 is odd
```

ಕರೆಯುವ ಮೆಥೆಡ್‌ಗೆ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುವ ಇನ್ನೊಂದು ಪ್ರೊಗ್ರಾಂನ್ನು ನೋಡೋಣ.

ಕೆಳಗಿನ ಪ್ರೊಗ್ರಾಂನಲ್ಲಿ, ನಾವು add() ಎನ್ನುವ ಹೆಸರಿನ ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿದ್ದೇವೆ. ಇದು ಎರಡು ಸಂಖ್ಯೆಗಳನ್ನು ಕೂಡಿಸುತ್ತದೆ. ಇದು n1 ಮತ್ತು n2 ಎನ್ನುವ integer ಟೈಪ್‌ನ ಎರಡು ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ಹೊಂದಿದೆ. n1 ಮತ್ತು n2 ಗಳ ಮೌಲ್ಯವು, a ಮತ್ತು b ಗಳ ಮೌಲ್ಯಗಳಿಗೆ ಕ್ರಮವಾಗಿ ಅನುಗುಣವಾಗಿರುತ್ತವೆ. ಅದ್ದರಿಂದ ಮೆಥೆಡ್ a ಮತ್ತು b ಗಳ ಮೌಲ್ಯಗಳನ್ನು ಸೇರಿಸುತ್ತದೆ ಮತ್ತು ಅದನ್ನು ವೇರಿಯೇಬಲ್‌ನಲ್ಲಿ ಸಂಗ್ರಹಿಸುತ್ತದೆ ಮತ್ತು ಮೊತ್ತ(sum) ಅನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ.

Addition.java

```
public class Addition
{
public static void main(String[]args)
{
int a =19;
intb =5;
//method calling
int c = add(a, b);//a and b are actual parameters
System.out.println("The sum of a and b is="+c);
}
//user defined method
public static int add(int n1,int n2)//n1and n2 are formal
parameters
{
int s;
s=n1+n2;
return s; //returning the sum
}
}
```

ಫಲಿತಾಂಶ :

```
The sum of a and b is = 24
```

ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್ (Static Method)

static ಕೀವರ್ಡ್‌ನ್ನು ಹೊಂದಿರುವ ಮೆಥೆಡ್‌ನ್ನು ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ,

© NIMI NOT TO BE REPUBLISHED

ಕ್ಲಾಸ್‌ನ ನಿದರ್ಶನದ ಬದಲಾಗಿ ಒಂದು ಕ್ಲಾಸ್‌ಗೆ ಸೇರಿದ ಮೆಥೆಡ್‌ನ್ನು ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್ ಎನ್ನುತ್ತಾರೆ. ಮೆಥೆಡ್‌ನ ಹೆಸರಿನ ಮೊದಲು static ಕೀವರ್ಡ್ ಬಳಸುವುದರ ಮೂಲಕ ನಾವು ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್‌ನ್ನು ರಚಿಸಬಹುದು.

ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್‌ನ ಮುಖ್ಯ ಪ್ರಯೋಜನವೆಂದರೆ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸದೆಯೆ ನಾವು ಇದನ್ನು ಕರೆಯಬಹುದು.ಇದು ಸ್ಟ್ಯಾಟಿಕ್ ಡಾಟಾ ಮೆಂಬರ್‌ಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು ಮತ್ತು ಅವುಗಳ ಮೌಲ್ಯವನ್ನು ಬದಲಾಯಿಸಬಹುದು. ಇದನ್ನು instance ಮೆಥೆಡ್‌ನ್ನು ರಚಿಸಲು ಉಪಯೋಗಿಸಬಹುದು. ಇದನ್ನು ಕ್ಲಾಸ್ ಹೆಸರನ್ನು ಬಳಸಿ ಕರೆಯಬಹುದು. main() ಮೆಥೆಡ್, ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್‌ನ ಅತ್ಯಂತ ಉತ್ತಮ ಉದಾಹರಣೆಯಾಗಿದೆ.

static ಮೆಥೆಡ್‌ನ

ಉದಾಹರಣೆ Display.java

```
public class Display
{
public static void main(String[]args)
{
show();
}
static void show()
{
System.out.println("It is an example of static method.");
}
}
```

ಫಲಿತಾಂಶ :

It is an example of a static method.

Instance ಮೆಥೆಡ್ (Instance Method)

ಒಂದು ಕ್ಲಾಸ್‌ನ ಮೆಥೆಡ್‌ನ್ನು instance ಮೆಥೆಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಕ್ಲಾಸ್‌ನಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ non - static ಮೆಥೆಡ್ ಆಗಿದೆ. instance ಮೆಥೆಡ್‌ನ್ನು ಕರೆಯುವ ಅಥವಾ invoke ಮಾಡುವ ಮೊದಲು, ಅದರ ಕ್ಲಾಸ್‌ನ ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ರಚಿಸುವುದು ಅವಶ್ಯಕವಾಗಿರುತ್ತದೆ. instance ಮೆಥೆಡ್‌ನ ಉದಾಹರಣೆಯನ್ನು ನೋಡೋಣ.

Instance ಮೆಥೆಡ್ Example.java

```
public class Instance Method Example
{
public static void main(String[]args)
{
//Creating an object of the class
InstanceMethod Example obj= new Instance Method
Example();
//invoking instancemethod
System.out.println("The sum is:"+ obj.add(12,13));
}
int s;
//user-defined method because we have not used static key-
word
public
int add(int a, intb)
{
s= a+b;
//returning the sum
return s;
}
}
```

ಫಲಿತಾಂಶ :

The sum is:25

instance ಮೆಥೆಡ್‌ನಲ್ಲಿ ಎರಡು ವಿಧಗಳಿವೆ:

- Accessor ಮೆಥೆಡ್
- Mutator ಮೆಥೆಡ್

ಎಕ್ಸೆಸರ್ ಮೆಥೆಡ್ (Accessor Method): instance ವೇರಿಯೇಬಲ್‌ನ್ನು (ಗಳು) ಓದುವ ಮೆಥೆಡ್(ಗಳು) ನ್ನು accessor ಮೆಥೆಡ್ ಎನ್ನುತ್ತಾರೆ. ನಾವು ಸುಲಭವಾಗಿ ಇದನ್ನು ಗುರುತಿಸಬಹುದು. ಏಕೆಂದರೆ ಈ ಮೆಥೆಡ್‌ನ್ನು get ಎನ್ನುವ ಪೂರ್ವಪದ ಸೇರಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು getters ಎಂದೂ ಸಹ ಕರೆಯಲಾಗುತ್ತದೆ. ಇದು private ಫೀಲ್ಡ್‌ನ ಮೌಲ್ಯವನ್ನು ಹಿಂತಿರುಗಿಸುತ್ತದೆ. ಇದನ್ನು private ಫೀಲ್ಡ್‌ನ ಮೌಲ್ಯವನ್ನು ಪಡೆಯಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

```
public int getId()
```


© NIMI NOT TO BE REPUBLISHED

```
{
return Id;
}
```

ಮ್ಯುಟೇಟರ್ ಮೆಥೆಡ್ (Mutator Method) : ಈ ಮೆಥೆಡ್(ಗಳು), instance ವೇರಿಯೇಬಲ್‌ನ್ನು (ಗಳನ್ನು) ಓದುತ್ತದೆ ಮತ್ತು ಮೌಲ್ಯಗಳನ್ನು ಮಾರ್ಪಾಡು ಸಹ ಮಾಡುತ್ತದೆ. ನಾವು ಈ ಮೆಥೆಡ್‌ನ್ನು ಸುಲಭವಾಗಿ ಗುರುತಿಸಬಹುದು ಏಕೆಂದರೆ ಮೆಥೆಡ್‌ಗೆ set ಎನ್ನುವ ಪೂರ್ವಪದ ಸೇರಿಸಲಾಗಿದೆ. ಇದನ್ನು setters ಅಥವಾ modifiers ಎಂದೂ ಸಹ ಕರೆಯಲಾಗುತ್ತದೆ. ಇದು ಏನನ್ನೂ ಹಿಂತಿರುಗಿಸುವುದಿಲ್ಲ. ಇದು ಫೀಲ್ಡ್‌ನ್ನು ಅವಲಂಬಿಸಿರುವ ಒಂದೇ ರೀತಿಯ ಡಾಟಾಟೈಪ್ ಹೊಂದಿರುವ ಪ್ಯಾರಾಮೀಟರ್‌ಗಳನ್ನು ಸ್ವೀಕರಿಸುತ್ತದೆ. ಇದನ್ನು private ಫೀಲ್ಡ್‌ನ ಮೌಲ್ಯಗಳನ್ನು ಸೆಟ್ ಮಾಡಲು ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಉದಾಹರಣೆ

```
public void setRoll(int roll)
{
this.roll=roll;
}
```

accessor ಮತ್ತು mutator ಮೆಥೆಡ್‌ನ ಉದಾಹರಣೆಗಳು

Student.java

```
public class Student
{
private int roll;
private String name;
public int getRoll()//accessor method
{
return roll;
}
public void setRoll(introll)//mutator method
{
this.roll=roll;
}
public String getName()
{
return name;
}
public void setName(Stringname)
{
this.name=name;
}
public void display()
{
System.out.println("Roll no.: "+ roll);
System.out.println("Studentname:"+ name);
}
}
```

© NIMI NOT TO BE REPUBLISHED

ಜಾವಾದಲ್ಲಿ ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಕ್ಲಾಸ್‌ಗಳು ಮತ್ತು ಇಂಟರ್‌ಫೇಸ್‌ಗಳು (Abstract Classes and Interfaces in JAVA)

ಉದ್ದೇಶಗಳು : ಈ ಪಾಠದ ಕೊನೆಯಲ್ಲಿ ನೀವು ತಿಳಿದಿರಬೇಕಾದ ಅಂಶಗಳು

- ಮೆಥೆಡ್
- ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನ ವಿಧಗಳು
- ಓವರ್‌ರೈಡಿಂಗ್‌ನ ನಿಯಮಗಳು.

ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮೆಥೆಡ್ (Abstract Method)

ಮುಖ್ಯಭಾಗವನ್ನು ಹೊಂದದೇ ಇರುವ ಮೆಥೆಡ್‌ನ್ನು ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮೆಥೆಡ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ ಅನುಷ್ಠಾನವಿಲ್ಲದೇ ಇರುವ ಮೆಥೆಡ್‌ನ್ನು ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮೆಥೆಡ್ ಎನ್ನುತ್ತಾರೆ. ಇದನ್ನು ಯಾವಾಗಲೂ ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಕ್ಲಾಸ್‌ನಲಿ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾಗುತ್ತದೆ. ಇದರ ಅರ್ಥ ಒಂದು ಕ್ಲಾಸ್ ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮೆಥೆಡ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ ಅದು ಸ್ವತಃ ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಕ್ಲಾಸ್ ಆಗಿರಬೇಕು. ಅಬ್‌ಸ್ಟ್ರ್ಯಾಕ್ಟ್ ಮೆಥೆಡ್‌ನ್ನು ರಚಿಸಲು ನಾವು abstract ಕೀವರ್ಡ್‌ನ್ನು ಉಪಯೋಗಿಸಬೇಕು.

ಸಿಂಟ್ಯಾಕ್ಸ್

abstract void method_name();

abstract method ನ

ಉದಾಹರಣೆ Demo.java

```
abstract class Demo//abstract class
{
//abstract method declaration abstract void display();
}
public class MyClass extends Demo
{
//method impelmentation void display()
{
System.out.println("Abstract method?");
}
public static void main(Stringargs[])
{
//creating object of abstract
class Demo obj= new MyClass();
//invoking abstract method obj.display();
}
}
```

ಫಲಿತಾಂಶ

Abstract method...

ಫ್ಯಾಕ್ಟರಿ ಮೆಥೆಡ್ (factory Method)

ಇದು ಒಂದು ಆಬ್ಜೆಕ್ಟ್‌ನ್ನು ಅದಕ್ಕೆ ಸೇರಿದ ಕ್ಲಾಸ್‌ಗೆ ಹಿಂತಿರುಗಿಸುವ ಮೆಥೆಡ್ ಆಗಿದೆ. ಎಲ್ಲ ಸ್ಟ್ಯಾಟಿಕ್ ಮೆಥೆಡ್‌ಗಳು ಫ್ಯಾಕ್ಟರಿ ಮೆಥೆಡ್‌ಗಳಾಗಿರುತ್ತವೆ. ಉದಾಹರಣೆಗೆ Number Format obj = Number Format.getNumber Instance();

ಇನ್‌ಹೆರಿಟನ್ಸ್ (Inheritance)

ಜಾವಾದಲ್ಲಿ ಒಂದು ಆಬ್ಜೆಕ್ಟ್ ತನ್ನ ಪೇರೆಂಟ್ ಆಬ್ಜೆಕ್ಟ್‌ನ ಎಲ್ಲಾ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಮತ್ತು ನಡುವಳಿಕೆಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುವ ಒಂದು ಕಾರ್ಯವಿಧಾನವನ್ನು ಇನ್‌ಹೆರಿಟನ್ಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು OOPs ನ (Object Oriented programming system) ಪ್ರಮುಖ ಭಾಗವಾಗಿದೆ.

ಜಾವಾದಲ್ಲಿ ಇನ್‌ಹೆರಿಟನ್ಸ್ ನ ಹಿಂದಿನ ವಿಚಾರವೇನೆಂದರೆ ನೀವು ಈಗಾಗಲೆ ಇರುವ ಕ್ಲಾಸ್‌ಗಳ ಆಧಾರದ ಮೇಲೆ ಹೊಸ ಕ್ಲಾಸ್‌ಗಳನ್ನು ರಚಿಸುವುದು. ನೀವು ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಕ್ಲಾಸ್‌ನಿಂದ ಪರಂಪರೆಯಾಗಿ ಪಡೆದುಕೊಂಡಾಗ (inherit), ನೀವು ಪೇರೆಂಟ್ ಕ್ಲಾಸ್‌ನ ಮೆಥೆಡ್‌ಗಳನ್ನು ಮತ್ತು ಫೀಲ್ಡ್‌ಗಳನ್ನು ಮರುಬಳಕೆ ಮಾಡಬಹುದು. ಇದಲ್ಲದೇ ನೀವು ನಿಮ್ಮ ಹೊಸ ಕ್ಲಾಸ್‌ನಲ್ಲಿಯೂ ಸಹ ಹೊಸ ಮೆಥೆಡ್‌ಗಳನ್ನು ಮತ್ತು ಫೀಲ್ಡ್‌ಗಳನ್ನು ಸೇರಿಸಬಹುದು.

© NIMI NOT TO BE REPUBLISHED

ಇನ್‌ಹೆರಿಟನ್ಸ್ , IS- A ರಿಲೇಷನ್‌ಶಿಪ್ ಅನ್ನು ಪ್ರತಿನಿಧಿಸುತ್ತದೆ. ಇದನ್ನು ಪೆರೆಂಟ್ - ಚೈಲ್ಡ್ ರಿಲೇಷನ್‌ಶಿಪ್ ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ.

ಜಾವಾದಲ್ಲಿ ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನ್ನು ಏಕೆ ಉಪಯೋಗಿಸಬೇಕು ?

- ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ಗಾಗಿ (ರನ್ ಟೈಮ್ ಪಾಲಿ ಮಾರ್ಫಿಸಮ್‌ನ್ನು ಸಾಧಿಸಬಹುದು)
- ಕೋಡ್‌ನ್ನು ಮರುಬಳಕೆ ,ಮಾಡುವ ಸಲುವಾಗಿ (code reusability)

ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನಲ್ಲಿ ಬಳಸಲಾಗುವ ಪದಗಳು

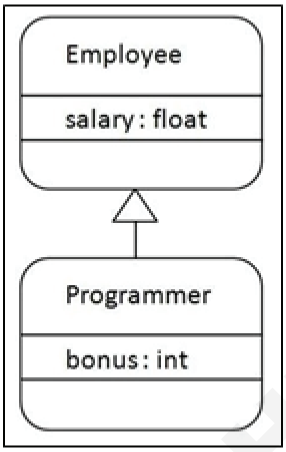
- **(ಕ್ಲಾಸ್) Class** : ಒಂದೇ ಗುಣಲಕ್ಷಣಗಳನ್ನು ಹೊಂದಿರುವ ಆಬ್ಜೆಕ್ಟ್‌ಗಳ ಗುಂಪನ್ನು ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದು ಟೆಂಪ್ಲೇಟ್ ಅಥವಾ ಬ್ಲೂಪ್ರಿಂಟ್ ಆಗಿದೆ. ಇದರಿಂದ ಆಬ್ಜೆಕ್ಟ್‌ಗಳನ್ನು ರಚಿಸಬಹುದು.
- **ಸಬ್ ಕ್ಲಾಸ್ / ಚೈಲ್ಡ್ ಕ್ಲಾಸ್ (Sub Class/Child Class)** : ಇತರ ಕ್ಲಾಸ್‌ನ್ನು inherit ಮಾಡುವ ಕ್ಲಾಸ್‌ನ್ನು ಸಬ್‌ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು ಡಿರೈವ್ಡ್ ಕ್ಲಾಸ್, ಎಕ್ಸ್‌ಟೆಂಡೆಡ್ ಕ್ಲಾಸ್, ಅಥವಾ ಚೈಲ್ಡ್ ಕ್ಲಾಸ್ ಎಂತಲೂ ಕರೆಯುತ್ತಾರೆ.
- **ಸೂಪರ್ ಕ್ಲಾಸ್ / ಪೆರೆಂಟ್ ಕ್ಲಾಸ್ (Super Class/Parent Class)** : ಯಾವ ಕ್ಲಾಸ್‌ನಿಂದ ಸಬ್ ಕ್ಲಾಸ್ ವೈಶಿಷ್ಟ್ಯಗಳನ್ನು ಪಡೆದುಕೊಳ್ಳುತ್ತದೆಯೊ ಆ ಕ್ಲಾಸ್‌ನ್ನು ಸೂಪರ್‌ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಇದನ್ನು ಬೇಸ್ ಕ್ಲಾಸ್ ಅಥವಾ ಪೆರೆಂಟ್ ಕ್ಲಾಸ್ ಎಂದೂ ಸಹ ಕರೆಯುತ್ತಾರೆ.
- **ಮರುಬಳಕೆ (Reusability)** : ಇದರ ಹೆಸರೆ ಹೇಳುವಂತೆ, ನೀವು ಹೊಸ ಕ್ಲಾಸ್‌ನ್ನು ರಚಿಸಿದಾಗ ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಕ್ಲಾಸ್‌ನ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಮೆಥೆಡ್‌ಗಳನ್ನು ಮರುಬಳಕೆ ಮಾಡಲು ಅನುಕೂಲ ಮಾಡಿಕೊಡುವ ವ್ಯವಸ್ಥೆಯನ್ನು ರಿಯೂಸೆಬಲಿಟಿ (ಮರುಬಳಕೆ) ಎನ್ನುತ್ತಾರೆ. ಹಿಂದಿನ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಈಗಾಗಲೇ ವ್ಯಾಖ್ಯಾನಿಸಲಾದ ಅದೇ ಫೀಲ್ಡ್‌ಗಳನ್ನು ಮತ್ತು ಮೆಥೆಡ್‌ಗಳನ್ನು ನೀವು ಉಪಯೋಗಿಸಬಹುದು.

ಜಾವಾ Inheritance ನ ಸಿಂಟ್ಯಾಕ್ಸ್

```
class Subclass-name extends Superclass-name
{
//methods and fields
}
```

ಈಗಾಗಲೇ ಅಸ್ತಿತ್ವದಲ್ಲಿರುವ ಕ್ಲಾಸ್‌ನಿಂದ ಹೊಸ ಕ್ಲಾಸ್‌ನ್ನು ಪಡೆದುಕೊಳ್ಳಬಹುದು ಎಂದು extends ಕೀವರ್ಡ್ ಸೂಚಿಸುತ್ತದೆ. "extends" ಪದದ ಅರ್ಥ ಕ್ರಿಯಾತ್ಮಕತೆಯನ್ನು ಹೆಚ್ಚಿಸುವುದು ಎಂದಾಗಿದೆ.

ಜಾವಾದ ಪರಿಭಾಷೆಯಲ್ಲಿ, ಯಾವ ಕ್ಲಾಸ್‌ನಿಂದ ಪಡೆಯಲಾಗಿದೆಯೊ ಅದನ್ನು ಪೇರೆಂಟ್ ಅಥವಾ ಸೂಪರ್ ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ ಮತ್ತು ಹೊಸ ಕ್ಲಾಸ್‌ನ್ನು ಚೈಲ್ಡ್ ಅಥವಾ ಸಬ್ ಕ್ಲಾಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.



ಜಾವಾ ಇನ್‌ಹೆರಿಟನ್ಸ್ (Java Inheritance)

ಉದಾಹರಣೆ

As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationshipbetweenthetwoclassesisProgrammerI-SAEmployee.

Itmeans thatProgrammer isa type of

Employee.

```
class Employee
{
float salary=40000;
}
class Programmer extends
Employee{int bonus=10000;
public static void main(String args[])
{
Programmer p=new
Programmer();System.out.println("Programmer salary is:"+
p.salary);System.out.print ln ("Bonus of Programmeris:"
+ p.bonus);
}
}
```

NIMI © NOT TO BE REPUBLISHED

ಫಲಿತಾಂಶ

Programmer salary

is: 40000.0 Bonus of programme

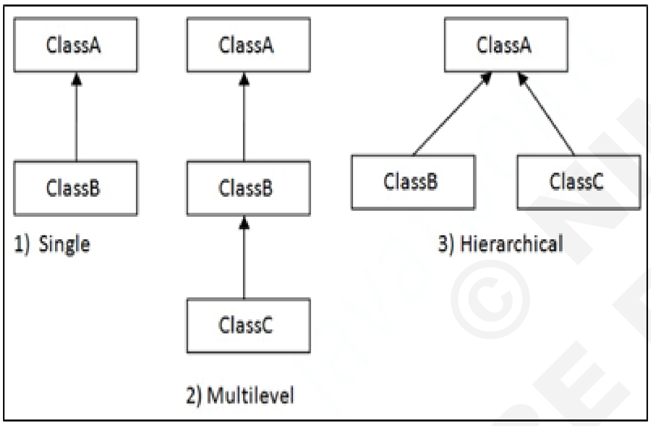
ris:10000

ಮೇಲಿನ ಉದಾಹರಣೆಯಲ್ಲಿ, programmer ಆಬ್ಜೆಕ್ಟ್ ತನ್ನ ಸ್ವಂತ ಕ್ಲಾಸ್‌ನ ಮತ್ತು Employee ಕ್ಲಾಸ್‌ನ ಫೀಲ್ಡ್‌ನ್ನು ಪ್ರವೇಶಿಸಬಹುದು. ಎಂದರೆ ಕೊಡ್‌ನ ಮರುಬಳಕೆ (i.e.code reusability.)

ಜಾವಾದಲ್ಲಿ ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನ ವಿಧಗಳು (Types of inheritance in java)

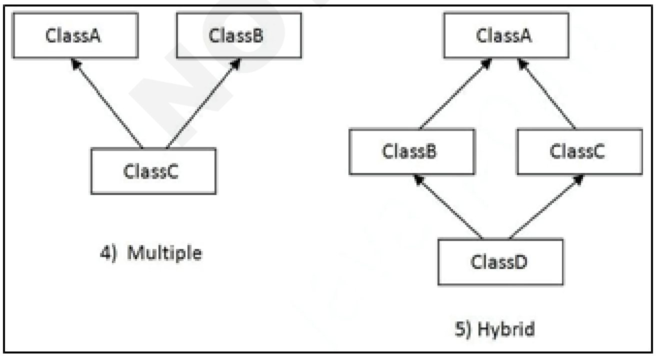
ಜಾವಾದಲ್ಲಿ ಕ್ಲಾಸ್‌ನ ಆಧಾರದ ಮೇಲೆ ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನಲ್ಲಿ ಮೂರು ವಿಧಗಳಿವೆ. : single, multilevel ಮತ್ತು hierarchical.

ಜಾವಾ ಪ್ರೊಗ್ರಾಮಿಂಗ್‌ನಲ್ಲಿ, multiple ಮತ್ತು hybrid ಇನ್‌ಹೆರಿಟನ್ಸ್‌ಗಳು ಇಂಟರ್‌ಫೇಸ್‌ನಿಂದ ಮಾತ್ರ ಬೆಂಬಲಿಸಲ್ಪಡುತ್ತವೆ. ನಾವು ಇಂಟರ್‌ಫೇಸ್ ಬಗ್ಗೆ ನಂತರ ಕಲಿಯೋಣ.



ಸೂಚನೆ : ಜಾವಾದಲ್ಲಿ multiple ಇನ್‌ಹೆರಿಟನ್ಸ್ , ಕ್ಲಾಸ್ ಮೂಲಕ ಬೆಂಬಲಿಸಲ್ಪಡುವುದಿಲ್ಲ.

ಯಾವಾಗ ಒಂದು ಕ್ಲಾಸ್ ಹಲವಾರು ಕ್ಲಾಸ್‌ಗಳನ್ನು inherit ಮಾಡುತ್ತದೆಯೊ ಅದನ್ನು multiple ಇನ್‌ಹೆರಿಟನ್ಸ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಉದಾಹರಣೆಗೆ



Single Inheritance **ಉದಾಹರಣೆ**

ಒಂದು ಕ್ಲಾಸ್ ಇನ್ನೊಂದು ಕ್ಲಾಸ್‌ನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡಿದಾಗ, ಇದನ್ನು Single Inheritance ಎನ್ನುತ್ತಾರೆ. ಕೆಳಗೆ ನೀಡಲಾದ ಉದಾಹರಣೆಯಲ್ಲಿ Dog ಕ್ಲಾಸ್ Animal ಕ್ಲಾಸನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡುತ್ತದೆ. ಅದ್ದರಿಂದ ಇದು Single Inheritance ಆಗುತ್ತದೆ.

File:TestInheritance.java

```
class Animal{
void eat(){System.out.println("eating...");
}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class TestInheritance{
public static void main(String
args[]){ Dog d=new Dog();
d.bark();
d.eat();
}}
```

ಫಲಿತಾಂಶ :

barking...

eating...

Multilevel Inheritance **ಉದಾಹರಣೆ**

ಇನ್‌ಹೆರಿಟನ್ಸ್‌ನ ಸರಪಳಿ ಇದ್ದಾಗ ಅದನ್ನು Multilevel Inheritance ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಕೆಳಗೆ ನೀಡಲಾದ ಉದಾಹರಣೆಯಲ್ಲಿ ನೀವು ನೋಡಬಹುದು BabyDog ಕ್ಲಾಸ್ Dog ಕ್ಲಾಸ್‌ನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡುತ್ತದೆ ಇದು ಮತ್ತೆ Animal ಕ್ಲಾಸ್‌ನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡುತ್ತದೆ. ಅದ್ದರಿಂದ ಇದು multilevel ಇನ್‌ಹೆರಿಟನ್ಸ್ ಆಗಿದೆ.

File:TestInheritance2.java

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
```

NIMI © NOT TO BE REPUBLISHED

```
class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
public static void main(String
args[]){BabyDog d=new
BabyDog();d.weep();
d.bark();
d.eat();
}}
```

ಫಲಿತಾಂಶ: weeping...barking...eating...

Hierarchical Inheritance ಉದಾಹರಣೆ

ಎರಡು ಮತ್ತು ಎರಡಕ್ಕಿಂತಲೂ ಹೆಚ್ಚು ಕ್ಲಾಸ್‌ಗಳು ಒಂದು ಕ್ಲಾಸ್‌ನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡಿದಾಗ ಅದನ್ನು Hierarchical Inheritance ಎಂದು ಕರೆಯುತ್ತಾರೆ. ಕೆಳಗೆ ನೀಡಲಾದ ಉದಾಹರಣೆಯಲ್ಲಿ Dog ಮತ್ತು Cat ಕ್ಲಾಸ್‌ಗಳು Animal ಕ್ಲಾಸನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡುತ್ತವೆ. ಅದರಿಂದ ಇದು hierarchical ಇನ್‌ಹೆರಿಟನ್ಸ್ ಆಗಿದೆ.

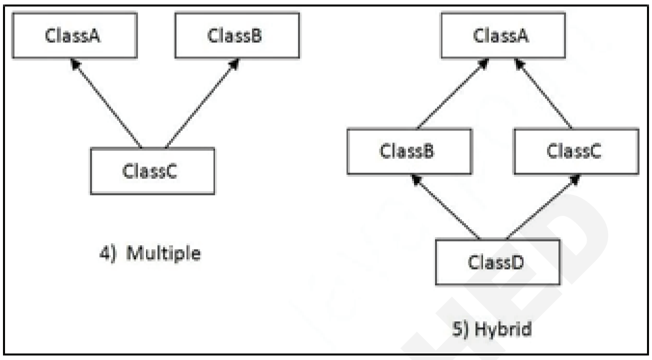
File:TestInheritance3.java

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String
args[]){Catc=new Cat();
c.meow();
c.eat();
//c.bark();//C.T.Error
}}
```

ಫಲಿತಾಂಶ :

meowing...

eating...



ಜಾವಾದಲ್ಲಿ multiple ಇನ್‌ಹೆರಿಟನ್ಸ್ ಬೆಂಬಲಿಸಲ್ಪಡುವುದಿಲ್ಲ ಏಕೆ ?

ಸಂಕೀರ್ಣತೆಯನ್ನು ಕಡಿಮೆ ಮಾಡಲು ಮತ್ತು ಭಾಷೆಯನ್ನು ಸರಳಗೊಳಿಸಲು, ಜಾವಾದಲ್ಲಿ multiple ಇನ್‌ಹೆರಿಟನ್ಸ್ ಬೆಂಬಲಿ ಸಲ್ಪಡುವುದಿಲ್ಲ.

A, B ಮತ್ತು C ಇವು ಮೂರು ಕ್ಲಾಸ್‌ಗಳಿರುವ ಒಂದು ಸನ್ನಿವೇಶವನ್ನು ಪರಿಗಣಿಸಿ. C ಕ್ಲಾಸ್ A ಮತ್ತು B ಕ್ಲಾಸ್‌ಗಳನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡುತ್ತದೆ. A ಮತ್ತು B ಕ್ಲಾಸ್‌ಗಳು ಒಂದೇ ಮೆಥೆಡ್‌ನ್ನು ಹೊಂದಿದ್ದರೆ ಮತ್ತು ನೀವು ಅದನ್ನು ಚೈಲ್ಡ್ ಕ್ಲಾಸ್ ಆಬ್ಜೆಕ್ಟ್‌ನಿಂದ ಕರೆದರೆ, A ಮತ್ತು B ಕ್ಲಾಸ್‌ಗಳಿಂದ ಮೆಥೆಡ್‌ನ್ನು ಕರೆಯುವುದರಲ್ಲಿ ಅಸ್ಪಷ್ಟತೆ ಇರುತ್ತದೆ.

ರನ್‌ಟೈಮ್ ಎರರ್‌ಗಳಿಗಿಂತ ಕಂಪೈಲ್‌ಟೈಮ್ ಎರರ್‌ಗಳು ಉತ್ತಮವಾಗಿರುವುದರಿಂದ, ನೀವು 2 ಕ್ಲಾಸ್‌ಗಳನ್ನು ಇನ್‌ಹೆರಿಟ್ ಮಾಡಿದಾಗ ಜಾವಾ ಕಂಪೈಲ್‌ಟೈಮ್ ಎರರ್‌ನ್ನು ನೀಡುತ್ತದೆ. ಅದ್ದರಿಂದ ನೀವು ಒಂದೇ ಮೆಥೆಡ್‌ನ್ನು ಹೊಂದಿದ್ದರೂ ಅಥವಾ ವಿಭಿನ್ನ ಮೆಥೆಡ್‌ನ್ನು ಹೊಂದಿದ್ದರೂ ಅದು ಕಂಪೈಲ್‌ಟೈಮ್ ಎರರ್‌ನ್ನು ನೀಡುತ್ತದೆ.

```
class A{
void msg(){ System.out.println("Hello");}
}
class B{
void msg(){ System.out.println("Welcome");}
}
class C extends A, B{ //suppose if it were
```


© NIMI NOT TO BE REPUBLISHED

```
public static void main(String args[]){C obj=new C();
obj.msg();//Now which msg() method would be invoked?
}
}
```

CompileTime Error

ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್ (Method Over riding in Java)

ಸಬ್‌ಕ್ಲಾಸ್ ಸಹ ಪೇರೆಂಟ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಡಿಕ್ಲೇರ್ ಮಾಡಿದ ಮೆಥೆಡ್‌ನ್ನೇ ಹೊಂದಿದ್ದರೆ, ಜಾವಾದಲ್ಲಿ ಅದನ್ನು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್ ಎಂದು ಕರೆಯುತ್ತಾರೆ.

ಬೇರೆ ರೀತಿಯಲ್ಲಿ ಹೇಳುವುದಾದರೆ, ಒಂದು ಸಬ್‌ಕ್ಲಾಸ್ ತನ್ನ ಪೇರೆಂಟ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಡಿಕ್ಲೇರ್ ಮಾಡಲಾದ ಮೆಥೆಡ್‌ನ್ನು ಸರಿಯಾಗಿ ಅನುಷ್ಠಾನ ಮಾಡಿದರೆ ಅದನ್ನು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್ ಎಂದು ಕರೆಯಲಾಗುತ್ತದೆ.

ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ಬಳಕೆ (Usage of Java Method Overriding)

- ತನ್ನ ಸೂಪರ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ಈಗಾಗಲೇ ಒದಗಿಸಲಾದ ಮೆಥೆಡ್‌ನ್ನು ಒಂದು ಮೆಥೆಡ್ ನಿರ್ದಿಷ್ಟವಾಗಿ ಅನುಷ್ಠಾನಗೊಳಿಸಲು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ್ನು ಉಪಯೊಗಿಸಲಾಗುತ್ತದೆ.
- ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ್ನು ರನ್‌ಟೈಮ್ ಪಾಲಿ ಮಾರ್ಫಿಸಮ್‌ಗಾಗಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ.

ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ನಿಯಮಗಳು (Rules for Java Method Overriding)

1 ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿರುವ ಅದೇ ಹೆಸರನ್ನೆ ಮೆಥೆಡ್ ಸಹ ಹೊಂದಿರಬೇಕು.

2 ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿರುವಂತೆಯೆ ಅದೇ ಪ್ಯಾರಾಮೀಟರ್‌ನ್ನು ಮೆಥೆಡ್ ಹೊಂದಿರಬೇಕು.

3 ಅಲ್ಲಿ IS - A ರಿಲೇಷನ್‌ಶಿಪ್ (ಇನ್‌ಹೆರಿಟನ್ಸ್) ಇರಬೇಕು.

ನಾವು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ್ನು ಉಪಯೋಗಿಸದಿದ್ದರೆ ನಾವು ಪ್ರೊಗ್ರಾಂನಲ್ಲಿ ಎದುರಿಸಬಹುದಾದ ಸಮಸ್ಯೆಯನ್ನು ತಿಳಿದುಕೊಳ್ಳೋಣ.

//ನಮಗೆ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ಅವಶ್ಯಕತೆ ಏಕೆ ಇದೆ ಎಂಬುದನ್ನು ತಿಳಿಸುವ ಜಾವಾ ಪ್ರೊಗ್ರಾಂ

//ಇಲ್ಲಿ ನಾವು ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನ್ನು ಅದರ ಚೈಲ್ಡ್‌ನೊಂದಿಗೆ ಕಾಲ್ ಮಾಡುತ್ತಿದ್ದೇವೆ.

```
//ಕ್ಲಾಸ್ ಆಬ್ಜೆಕ್ಟ್
//ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನ್ನು ರಚಿಸುವುದು class Vehicle{
void run(){System.out.println("Vehicle is running");}
}
//ಚೈಲ್ಡ್ ಕ್ಲಾಸ್ ರಚಿಸುವುದು class Bike extends Vehicle{
publicstaticvoidmain(String args[]){
//ಚೈಲ್ಡ್‌ನ instance ರಚಿಸುವುದು class Bike obj =new Bike();
//calling the method with child class instance obj.run();
}
}
```

ಫಲಿತಾಂಶ :

Vehicle is running

ಇಲ್ಲಿ ಸಮಸ್ಯೆ ಏನೆಂದರೆ, ನಾವು ಸಬ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ run() ಮೆಥೆಡ್‌ನ ನಿರ್ದಿಷ್ಟ ಅನುಷ್ಠಾನವನ್ನು ಒದಗಿಸಬೇಕಾಗುತ್ತದೆ. ಅದ್ದರಿಂದ ನಾವು ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ್ನು ಉಪಯೋಗಿಸಬೇಕಾಗುತ್ತದೆ.

ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ಉದಾಹರಣೆ

ಈ ಉದಾಹರಣೆಯಲ್ಲಿ ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನಲ್ಲಿ ವ್ಯಾಖ್ಯಾನಿಸಿದಂತೆಯೆ ಸಬ್‌ಕ್ಲಾಸ್‌ನಲ್ಲಿ ನಾವು run ಮೆಥೆಡ್‌ನ್ನು ವ್ಯಾಖ್ಯಾನಿಸಿದ್ದೇವೆ. ಆದರೆ ಇದು ನಿರ್ದಿಷ್ಟ ಅನುಷ್ಠಾನವನ್ನು ಹೊಂದಿದೆ. ಮೆಥೆಡ್‌ನ ಹೆಸರು ಮತ್ತು ಪ್ಯಾರಾಮೀಟರ್ ಒಂದೇ ಆಗಿವೆ, ಮತ್ತು ಕ್ಲಾಸ್‌ಗಳ ನಡುವೆ IS - A ರಿಲೇಷನ್‌ಶಿಪ್ ಇದೆ. ಅದ್ದರಿಂದ ಇಲ್ಲಿ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್ ಆಗಿದೆ.

// ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ಬಳಕೆಯನ್ನು ವಿವರಿಸುವ ಜಾವಾ ಪ್ರೊಗ್ರಾಂ

ಜಾವಾದಲ್ಲಿ ಮೆಥೆಡ್ ಓವರ್ ರೈಡಿಂಗ್

```
//Creating a parent class.
class Vehicle{
//defining a method
void run(){System.out.println("Vehicle is running");}
}
//Creating a child class classBike2 extends Vehicle{
//defining the same method as in the parent class
void run(){System.out.println("Bike is running safely");}
```

```
public static void main(String
args[]){Bike2 obj = new
Bike2();//creating
object obj.run();//calling method
}
}
```

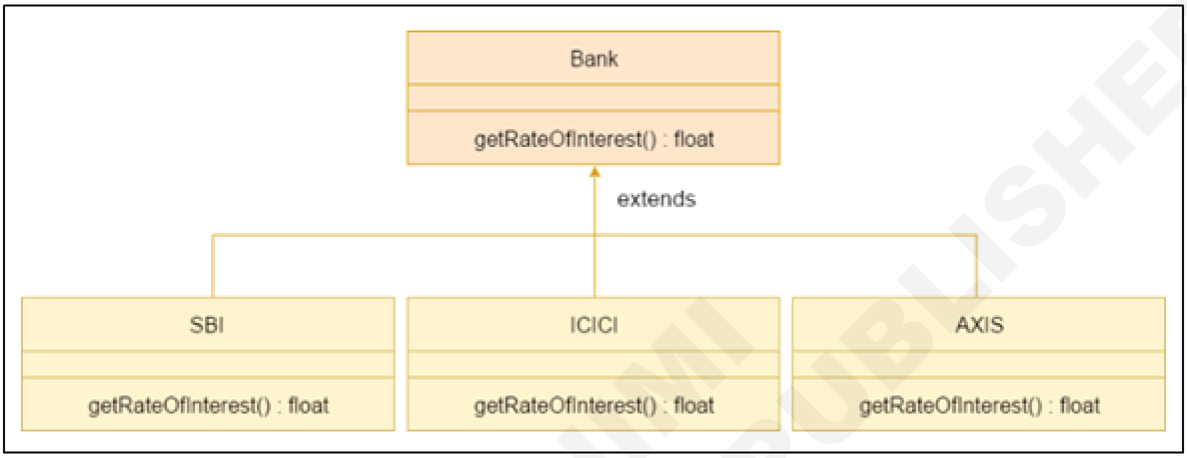
ಫಲಿತಾಂಶ :

Bike is running safely

ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ ನೈಜ ಉದಾಹರಣೆ

Bank, - ಬಡ್ಡಿ ದರವನ್ನು ಪಡೆಯಲು ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಒಂದು ಕ್ಲಾಸ್ ಆಗಿದೆ ಎಂದು ಒಂದು ಸನ್ನಿವೇಶವನ್ನು ಪರಿಗಣಿಸಿ. ಅದಾಗ್ಯೂ ಬ್ಯಾಂಕ್‌ಗಳಿಗೆ ಅನುಗುಣವಾಗಿ ಅವುಗಳ ಬಡ್ಡಿ ದರ ಬದಲಾಗುತ್ತದೆ. ಉದಾಹರಣೆಗೆ SBI, ICICI ಮತ್ತು AXIS ಬ್ಯಾಂಕ್‌ಗಳು 8%, 7%, ಮತ್ತು 9% ಬಡ್ಡಿ ದರವನ್ನು ಒದಗಿಸುತ್ತವೆ.

ಗಮನಿಸಿ : ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್‌ರೈಡಿಂಗ್‌ನ್ನು ಹೆಚ್ಚಾಗಿ ರನ್‌ಟೈಮ್ ಪಾಲಿಮಾರ್ಫಿಸಮ್‌ನಲ್ಲಿ ಉಪಯೋಗಿಸಲಾಗುತ್ತದೆ. ಇದನ್ನು ನಾವು ಮುಂದಿನ ಪುಟಗಳಲ್ಲಿ ಕಲಿಯೋಣ.



© NIMI NOT TO BE REPUBLISHED

// ಜಾವಾ ಮೆಥೆಡ್ ಓವರ್ ರೈಡಿಂಗ್‌ನ ನೈಜ ಸನ್ನಿವೇಶವನ್ನು ವಿವರಿಸುವ ಜಾವಾ ಪ್ರೊಗ್ರಾಂ

// ಒಂದು ಪೆರೆಂಟ್ ಕ್ಲಾಸ್‌ನ ಮೆಥೆಡ್‌ನ್ನು ಮೂರು ಕ್ಲಾಸ್‌ಗಳು ಓವರ್‌ರೈಡ್ ಮಾಡುತ್ತಿವೆ.

// ಪೆರೆಂಟನ್ನು ರಚಿಸುವುದು

```
class.class Bank{
int getRateOfInterest(){return 0;}
}
//Creating child classes.
class SBI extends
Bank{
int getRateOfInterest(){return 8;}
}
class ICICI extends Bank{
int getRateOfInterest(){return 7;}
}
class AXIS extends Bank{
int getRateOfInterest(){return 9;}
}
//Test class to create objects and call the methods
class Test2{
public static void main(String args[]){
SBI s = new SBI();
ICICI i = new
ICICI(); AXIS a=new AXIS();
System.out.println(“SBI Rate of Interest: ”
+ s.getRateOfInterest()); System.out.println(“ICICI Rate
of Interest:
” + i.get RateOfInterest()); System.out. println(“AXISRate
of Interest:” + a.getRateOfInterest());
}
}
```

ಫಲಿತಾಂಶ :

SBI Rate of Interest: 8 ICICI Rate of Interest: 7 AXIS Rate of Interest: 9

ನಾವು static ಮೆಥೆಡ್‌ನ್ನು ಓವರ್ ರೈಡ್ ಮಾಡಬಹುದೇ?

ಇಲ್ಲ, static ಮೆಥೆಡ್‌ನ್ನು ಓವರ್ ರೈಡ್ ಮಾಡಲು ಬರುವುದಿಲ್ಲ. ಇದನ್ನು ರನ್‌ಟೈಮ್ ಪಾಲಿಮಾರ್ಫಿಸಮ್ ಮೂಲಕ ಸಾಬೀತು ಪಡಿಸಬಹುದು. ಇದನ್ನು ನಾವು ನಂತರ ತಿಳಿದುಕೊಳ್ಳೋಣ.

ನಾವು ಏಕೆ static ಮೆಥೆಡ್‌ನ್ನು ಓವರ್ ರೈಡ್ ಮಾಡಲು ಸಾಧ್ಯವಿಲ್ಲ.?

ಏಕೆಂದರೆ static ಮೆಥೆಡ್ ಕ್ಲಾಸ್‌ನೊಂದಿಗೆ ಬಂಧಿಸಲ್ಪಟ್ಟಿರುತ್ತದೆ ಅದಲ್ಲದೆ instance ಮೆಥೆಡ್ ಆಬ್ಜೆಕ್ಟ್‌ನೊಂದಿಗೆ ಬಂಧಿಸಲ್ಪಟ್ಟಿರುತ್ತದೆ. static ಕ್ಲಾಸ್ ಏರಿಯಾಗೆ ಸೇರಿದೆ ಮತ್ತು instance ಹೀಪ್ ಏರಿಯಾಗೆ ಸೇರಿದೆ.

ಜಾವಾ main ಮೆಥೆಡ್‌ನ್ನು ಓವರ್ ರೈಡ್ ಎ333ಉಾಡಬಹುದೆ?

ಇಲ್ಲ, ಏಕೆಂದರೆ main ಮೆಥೆಡ್, static ಮೆಥೆಡ್ ಆಗಿದೆ.

Difference between method Overloading and Method Overriding in java

Click me for the difference between method overloading andoverridingto be taken

MoretopicsonMethodOverriding (NotForBeginners)

MethodOverriding withAccess Modifiertobetaken

Let's see the concept of method overriding with access modifier.ExceptionHandlingwithMethodOverriding

Let's see the conceptofmethodoverriding with exception handling.

© NIMI NOT TO BE REPUBLISHED