

કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ COMPUTER OPERATOR AND PROGRAMMING ASSISTANT

NSQF સ્તર - 3

વેપાર સિદ્ધાંત
(TRADE THEORY)

સેક્ટર : IT & ITES

Sector : IT & ITES

(જુલાઈ 2022 - 1200 કલાકના સંશોધિત અભ્યાસક્રમ મુજબ)

(As per revised syllabus July 2022 - 1200 hrs)



ડાયરેક્ટોરેટ જનરલ ઓફ ટ્રેનિંગ
કૌશલ્ય વિકાસ અને ઉદ્યોગ સાહસિકતા મંત્રાલય
ભારત સરકાર



નેશનલ ઇન્સ્ટ્ર કશનલ
મીડિયા ઇન્સ્ટિટ્યુટ, ચેન્નઈ

પોસ્ટ બોક્સ નં. 3142, સીટીઆઈ કેમ્પસ, ગિન્ડી, ચેન્નઈ - 600 032

સેક્ટર : IT & ITES

અવધિ : 1 - વર્ષ

વ્યવસાય : કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ - વેપાર સિદ્ધાંત - NSQF સ્તર - 3 (સંશોધિત 2022)

Developed & Published by



નેશનલ ઇન્સ્ટ્રક્શનલ મીડિયા ઇન્સ્ટિટ્યુટ,
પોસ્ટ બોક્સ નં. 3142 ગિન્ડી, ચેન્નાઈ - 600 032 ભારત
ઈમેલ : chennai-nimi@nic.in
વેબસાઈટ : www.nimi.gov.in

કોપીરાઈટ © 2024 નેશનલ ઇન્સ્ટ્રક્શનલ મીડિયા ઇન્સ્ટિટ્યુટ, ચેન્નાઈ

પ્રથમ આવૃત્તિ : ફેબ્રુઆરી 2024

નકલો: 1000

₹.410/-

બધા હકો અમારી પાસે રાખેલા છે.

આ પ્રકાશનના કોઈપણ ભાગને નેશનલ ઇન્સ્ટ્રક્શનલ મીડિયા ઇન્સ્ટિટ્યુટ, ચેન્નાઈની લેખિતમાં પરવાનગી વિના કોઈપણ સ્વરૂપમાં અથવા કોઈપણ માધ્યમથી, ફોટોકોપી, રેકોર્ડિંગ અથવા કોઈપણ માહિતી સંગ્રહ અને પુનઃપ્રાપ્તિ પ્રણાલી સહિત ઇલેક્ટ્રોનિક અથવા મિકેનિકલ દ્વારા પુનઃઉત્પાદિત અથવા પ્રસારિત કરી શકાશે નહીં.

આમુખ

ભારત સરકારે રાષ્ટ્રીય કૌશલ્ય વિકાસ નીતિના ભાગ રૂપે નોકરીઓ સુરક્ષિત કરવામાં મદદ કરવા માટે 2020 સુધીમાં 30 કરોડ લોકોને, દર ચારમાંથી એક ભારતીયને કૌશલ્ય પ્રદાન કરવાનો મહત્વાકાંક્ષી લક્ષ્યાંક નક્કી કર્યો છે. ઔદ્યોગિક તાલીમ સંસ્થાઓ (ITIs) આ પ્રક્રિયામાં ખાસ કરીને કુશળ માનવશક્તિ પૂરી પાડવાના સંદર્ભમાં મહત્વપૂર્ણ ભૂમિકા ભજવે છે. આને ધ્યાનમાં રાખીને, અને પ્રશિક્ષણાર્થીઓને વર્તમાન ઉદ્યોગ સંબંધિત કૌશલ્ય પ્રશિક્ષણ આપવા માટે, ITI અભ્યાસક્રમ તાજેતરમાં વિવિધ હિતધારકોની જેમ કે મેન્ટર કાઉન્સિલની મદદથી અપડેટ કરવામાં આવ્યો છે. ઉદ્યોગો, સાહસિકો, શિક્ષણવિદો અને ITIs ના પ્રતિનિધિઓ.

નેશનલ ઇન્સ્ટ્રક્શનલ મીડિયા ઇન્સ્ટિટ્યૂટ (NIMI), ચેન્નાઈ, કૌશલ્ય વિકાસ અને સાહસિકતા મંત્રાલય હેઠળની એક સ્વાયત્ત સંસ્થાને ITIs અને અન્ય સંબંધિત સંસ્થાઓ માટે જરૂરી સૂચનાત્મક મીડિયા પેકેજો (IMPs) વિકસાવવા અને પ્રસારિત કરવાની જવાબદારી સોંપવામાં આવી છે.

સંસ્થા હવે સુધારેલા અભ્યાસક્રમને અનુરૂપ સૂચનાત્મક સામગ્રી લઈને આવી **કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ - વાર્ષિક પેટર્ન** હેઠળ IT & ITES સેક્ટરમાં - **ટ્રેવેપાર સિદ્ધાંત NSQF સ્તર - 3 (સંશોધિત 2022)**. NSQF લેવલ - 3 (સુધારેલ 2022) ટ્રેડ પ્રેક્ટિકલ તાલીમાર્થીઓને અંતરરાષ્ટ્રીય સમકક્ષતા ધોરણ મેળવવામાં મદદ કરશે જ્યાં તેમની કૌશલ્ય પ્રાવીણ્ય અને યોગ્યતાને વિશ્વભરમાં યોગ્ય રીતે માન્યતા આપવામાં આવશે અને આ અગાઉના શિક્ષણની માન્યતાના અવકાશને પણ વધારશે. NSQF સ્તર - 3 (સુધારેલ 2022) તાલીમાર્થીઓને આજીવન શિક્ષણ અને કૌશલ્ય વિકાસને પ્રોત્સાહન આપવાની તકો પણ મળશે. મને કોઈ શંકા નથી કે NSQF લેવલ - 3 (સુધારેલ 2022) સાથે ITI ના પ્રશિક્ષકો અને તાલીમાર્થીઓ અને તમામ હિતધારકો આ IMP નો મહત્તમ લાભ મેળવશે અને NIMI નો પ્રયાસ દેશમાં વ્યાવસાયિક તાલીમની ગુણવત્તા સુધારવામાં ઘણો આગળ વધશે..

NIMI ના એક્ઝિક્યુટિવ ડિરેક્ટર અને સ્ટાફ અને મીડિયા ડેવલપમેન્ટ કમિટીના સભ્યો આ પ્રકાશન બહાર લાવવામાં તેમના યોગદાન માટે પ્રશંસાને પાત્ર છે.

જય હિન્દ

અતુલ કુમાર તિવારી I.A.S

સચિવ

કૌશલ્ય વિકાસ અને ઉદ્યોગ સાહસિકતા મંત્રાલય,
ભારત સરકાર.

ફેબ્રુઆરી ૨૦૨૪

નવી દિલ્હી - 110 001

પ્રસ્તાવના

નેશનલ ઈન્સ્ટ્રક્શનલ મીડિયા ઈન્સ્ટિટ્યૂટ (NIMI) ની સ્થાપના 1986 માં ચેન્નઈ ખાતે તત્કાલીન રોજગાર અને તાલીમ મહાનિર્દેશાલય (D.G.E & T), શ્રમ અને રોજગાર મંત્રાલય, (હવે કૌશલ્ય વિકાસ અને ઉદ્યોગ સાહસિકતા મંત્રાલય હેઠળ) ભારત સરકાર દ્વારા કરવામાં આવી હતી, જેમાં ટેકનિકલ છે. સરકાર તરફથી સહાય ફેડરલ રિપબ્લિક ઓફ જર્મની. આ સંસ્થાનો મુખ્ય ઉદ્દેશ્ય કારીગરો અને એપ્રેન્ટિસશીપ તાલીમ યોજનાઓ હેઠળ નિયત અભ્યાસક્રમ (NSQF) મુજબ વિવિધ વેપારો માટે સૂચનાત્મક સામગ્રી વિકસાવવા અને પ્રદાન કરવાનો છે.

ભારતમાં NCVT/NAC હેઠળ વ્યાવસાયિક તાલીમનો મુખ્ય ઉદ્દેશ્યને ધ્યાનમાં રાખીને સૂચનાત્મક સામગ્રી બનાવવામાં આવી છે, જે વ્યક્તિને નોકરી કરવા માટે કૌશલ્યમાં નિપુણતા પ્રાપ્ત કરવામાં મદદ કરવાનો છે. સૂચનાત્મક સામગ્રી સૂચનાત્મક મીડિયા પેકેજો (IMPs) ના સ્વરૂપમાં બનાવવામાં આવે છે. IMP માં થિયરી બુક, પ્રેક્ટિકલ બુક, ટેસ્ટ અને એસાઈનમેન્ટ બુક, ઈન્સ્ટ્રક્ટર ગાઈડ, ઓડિયો વિડિયો અઈડ (વોલ ચાર્ટ અને ટ્રાન્સપરન્સી) અને અન્ય સપોર્ટ મટિરિયલનો સમાવેશ થાય છે.

વર્કશોપમાં તાલીમાર્થીઓ દ્વારા પૂર્ણ કરવામાં આવનારી વ્યાયામની શ્રેણીબદ્ધ વ્યાવહારિક પુસ્તકમાં સમાવેશ થાય છે. આ કસરતો એ સુનિશ્ચિત કરવા માટે બનાવવામાં આવી છે કે નિયત અભ્યાસક્રમમાં તમામ કૌશલ્યો આવરી લેવામાં આવે છે. વેપાર સિદ્ધાંત પુસ્તક તાલીમાર્થીને નોકરી કરવા સક્ષમ બનાવવા માટે જરૂરી સંબંધિત સૈદ્ધાંતિક જ્ઞાન પ્રદાન કરે છે. પરીક્ષણ અને સોંપણીઓ પ્રશિક્ષકને તાલીમાર્થીની કામગીરીના મૂલ્યાંકન માટે સોંપણીઓ આપવા સક્ષમ બનાવશે. વોલ ચાર્ટ અને પારદર્શિતા અનન્ય છે, કારણ કે તે માત્ર પ્રશિક્ષકને વિષયને અસરકારક રીતે રજૂ કરવામાં મદદ કરે છે પરંતુ તેને તાલીમાર્થીની સમજનું મૂલ્યાંકન કરવામાં પણ મદદ કરે છે. પ્રશિક્ષક માર્ગદર્શિકા પ્રશિક્ષકને તેના સૂચનાના સમયપત્રકની યોજના બનાવવા, કાચા માલની જરૂરિયાતો, રોજિંદા પાઠ અને પ્રદર્શનોની યોજના બનાવવા માટે સક્ષમ બનાવે છે.

કૌશલ્યોને ઉત્પાદક રીતે કરવા માટે આ સૂચનાત્મક સામગ્રીમાં કવાયતના QR કોડમાં સૂચનાત્મક વિડિયોઝ એમ્બેડ કરવામાં આવ્યા છે જેથી કૌશલ્ય શિક્ષણને કવાયતમાં આપવામાં આવેલા પ્રક્રિયાગત વ્યવહાર પગલાં સાથે સાંકળી શકાય. સૂચનાત્મક વિડિયો પ્રાયોગિક તાલીમના ધોરણની ગુણવત્તામાં સુધારો કરશે અને તાલીમાર્થીઓને ધ્યાન કેન્દ્રિત કરવા અને કુશળતાને એકીકૃત કરવા માટે પ્રેરિત કરશે.

IMP અસરકારક ટીમ વર્ક માટે વિકસાવવા માટે જરૂરી જટિલ કૌશલ્યો સાથે પણ વ્યવહાર કરે છે. અભ્યાસક્રમમાં સૂચવ્યા મુજબ સંલગ્ન વેપારના મહત્વના કૌશલ્ય વિસ્તારોને સમાવવા માટે પણ જરૂરી કાળજી લેવામાં આવી છે.

સંસ્થામાં સંપૂર્ણ સૂચનાત્મક મીડિયા પેકેજની ઉપલબ્ધતા ટ્રેનર અને મેનેજમેન્ટ બંનેને અસરકારક તાલીમ આપવામાં મદદ કરે છે.

IMP એ NIMI ના સ્ટાફ મેમ્બરો અને મીડિયા ડેવલપમેન્ટ કમિટીના સભ્યોના સામૂહિક પ્રયાસોનું પરિણામ છે જે ખાસ કરીને જાહેર અને ખાનગી ક્ષેત્રના ઉદ્યોગો, ડિરેક્ટોરેટ જનરલ ઓફ ટ્રેનિંગ (DGT), સરકારી અને ખાનગી ITIs હેઠળની વિવિધ તાલીમ સંસ્થાઓમાંથી લેવામાં આવે છે.

NIMI વિવિધ રાજ્ય સરકારોના રોજગાર અને તાલીમ નિયામક, જાહેર અને ખાનગી ક્ષેત્રના ઉદ્યોગોના પ્રશિક્ષણ વિભાગો, DGT અને DGT ક્ષેત્ર સંસ્થાઓના અધિકારીઓ, પ્રૂફ રીડર્સ, વ્યક્તિગત મીડિયા વિકાસકર્તાઓ અને તમામનો નિષ્ઠાપૂર્વક આભાર વ્યક્ત કરવા આ તક લેવા માંગે છે. સંયોજકો, પરંતુ જેમના સક્રિય સમર્થન માટે NIMI આ સામગ્રીને બહાર લાવવામાં સક્ષમ ન હોત.

આભાર

નેશનલ ઇન્સ્ટ્રક્શનલ મીડિયા ઇન્સ્ટિટ્યૂટ (NIMI) આ સૂચનાત્મક સામગ્રીને બહાર લાવવા માટે નીચેના મીડિયા ડેવલપર્સ અને તેમની પ્રાયોજક સંસ્થાઓ દ્વારા આપવામાં આવેલા સહકાર અને યોગદાન માટે નિષ્ઠાપૂર્વક આભાર સાથે સ્વીકારે છે. **વેપાર સિદ્ધાંત** ના વેપાર માટે **કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ NSQF સ્તર - 3 (સંશોધિત 2022)** IT & ITES ITIs માટે ક્ષેત્ર.

મીડિયા ડેવલપમેન્ટ કમિટીના સભ્યો

શ્રીમતી. એમ. ભાનુમથી	- મદદનીશ તાલીમ અધિકારી, સરકારી ITI, અંબતુર.
શ્રીમતી. વી. રેવતી	- મદદનીશ તાલીમ અધિકારી, સરકારી ITI, કોઈમ્બતુર.

નિમી કો-ઓર્ડિનેટર્સ

શ્રી નિર્માલ્યનાથ	- નાયબ નિયામક, NIMI, ચેન્નાઈ - 32.
શ્રી જી. માઈકલ જોની	- મેનેજર, NMI, ચેન્નાઈ - 32
શ્રી એસ. ગોપાલકૃષ્ણન	- આસિસ્ટન્ટ મેનેજર NIMI, ચેન્નાઈ - 32.

NIMI ડેટા એન્ટ્રી, CAD, DTP ઓપરેટરો માટે આ સૂચનાત્મક સામગ્રીના વિકાસની પ્રક્રિયામાં તેમની ઉત્તમ અને સમર્પિત સેવાઓ માટે તેની પ્રશંસા નોંધે છે.

આ સૂચનાત્મક સામગ્રીના વિકાસમાં યોગદાન આપનાર અન્ય તમામ NIMI સ્ટાફ દ્વારા કરવામાં આવેલા અમૂલ્ય પ્રયત્નોને પણ NIMI આભાર સાથે સ્વીકારે છે.

NIMI એ દરેક વ્યક્તિનો પણ આભારી છે જેમણે આ સૂચનાત્મક સામગ્રી વિકસાવવામાં પ્રત્યક્ષ કે પરોક્ષ રીતે મદદ કરી છે.

પરિચય

વેપાર પ્રેક્ટિકલ માટેની આ માર્ગદર્શિકા IT વર્કશોપમાં ઉપયોગ માટે બનાવાયેલ છે. તેમાં પ્રાયોગિક કસરતોની શ્રેણીનો સમાવેશ થાય છે જે કોર્સના પ્રથમ વર્ષ દરમિયાન તાલીમાર્થીઓ દ્વારા પૂર્ણ કરવાની હોય છે. IT & ITES સેક્ટર હેઠળ કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ વેપાર. તે છે રાષ્ટ્રીય કૌશલ્ય લાયકાત ફ્રેમવર્ક NSQF સ્તર - 3 (સંશોધિત 2022), તાલીમાર્થીઓને કસરત કરવામાં મદદ કરવા સૂચના/માહિતી દ્વારા પૂરક અને સમર્થિત. કસરતો ડિઝાઇન કરવામાં આવી છે. એ સુનિશ્ચિત કરવા માટે કે અભ્યાસક્રમમાં નિર્ધારિત તમામ કૌશલ્યો સંલગ્ન વેપારો સહિત આવરી લેવામાં આવ્યા છે. કોમ્પ્યુટર ઓપરેટર અને પ્રોગ્રામિંગ આસિસ્ટન્ટ હેઠળ વેપાર IT & ITES સેક્ટર ટ્રેડ પ્રેક્ટિકલ ને બાર મોડ્યુલમાં વિભાજિત કરવામાં આવે છે. વિવિધ મોડ્યુલો માટે સમયની ફાળવણી નીચે આપેલ છે:

મોડ્યુલ 1	- સલામત કામ કરવાની પદ્ધતિઓ	મોડ્યુલ 23	- ગ્રાફિક તત્વોનું સંચાલન કરો
મોડ્યુલ 2	- કમ્પ્યુટર ઘટકોનો પરિચય	મોડ્યુલ 24	- ઓડિઓ અને વિડિયો ઘટકોનું સંચાલન કર
મોડ્યુલ 3	- વિન્ડોઝ ઓપરેટિંગ સિસ્ટમનો ઉપયોગ	મોડ્યુલ 25	- સંક્રમણો અને એનિમેશન મેનેજ કરો
મોડ્યુલ 4	- કમ્પ્યુટર બેઝિક્સ અને સોફ્ટવેર ઇન્સ્ટોલેશન	મોડ્યુલ 26	- સહયોગ મેનેજ કરો
મોડ્યુલ 5	- DOS કમાન્ડ લાઇન ઇન્ટરફેસ	મોડ્યુલ 27	- MySQL ઇન્સ્ટોલ કરોગ
મોડ્યુલ 6	- ઉબુન્ટુ લિનક્સ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરો અને મૂળભૂત Linux આદેશો ચલાવો	મોડ્યુલ 28	- પપ્રશ્નો પર પ્રદર્શન
મોડ્યુલ 7	- વર્ડ પ્રોસેસિંગ સોફ્ટવેરનો ઉપયોગ	મોડ્યુલ 29	- પ્કાર્યો પર પ્રદર્શન કર
મોડ્યુલ 8	- ફોર્મેટ દસ્તાવેજો	મોડ્યુલ 30	- કમ્પ્યુટર નેટવર્ક સેટ કરો અને ગોઠવો
મોડ્યુલ 9	- કોષ્ટકો અને સૂચિઓનું સંચાલન કરો	મોડ્યુલ 31	- HTML ટેગ્સનો ઉપયોગ કરીને સરળ સ્થિર વેબ પૃષ્ઠો બનાવો
મોડ્યુલ 10	- સંદર્ભો બનાવો અને મેનેજ કરો	મોડ્યુલ 32	- JavaScript HTML પૃષ્ઠોમાં JavaScript એમ્બેડ કરો
મોડ્યુલ 11	- ગ્રાફિક એલિમેન્ટ્સ	મોડ્યુલ 33	- એક્સેલનો ઉપયોગ કરીને ડેટા વિઝ્યુલાઇઝેશન અથવા વિશ્લેષણ
મોડ્યુલ 12	- દસ્તાવેજ સહયોગ મેનેજ કરો	મોડ્યુલ 34	- ઉત્પાદનો અને સેવાઓને ઓળખવા માટે ઈ-કોમર્સ સાઇટ્સ બ્રાઉઝ કરો
મોડ્યુલ 13	- મેઇલિંગ મેનેજ કરો	મોડ્યુલ 35	- વાયરસ, સ્પાયવેર અને અન્ય દૂષિત કોડથી માર્હિતી, કમ્પ્યુટર અને નેટવર્કને સુરક્ષિત કર
મોડ્યુલ 14	- શીટ એપ્લિકેશન વર્કશીટ્સ અને વર્કબુકનું સંચાલન કરો	મોડ્યુલ 36	- કલાઉડ કમ્પ્યુટિંગ
મોડ્યુલ 15	- ડેટા કોષો અને શ્રેણીઓનું સંચાલન કરો	મોડ્યુલ 37	- એપ્લિકેશન ડેવલપ કરો અને એપ્લીકેશન ડેવલપમેન્ટ લાઇફ સાયકલ કર
મોડ્યુલ 16	- કોષ્ટકો અને કોષ્ટક ડેટા મેનેજ કરો	મોડ્યુલ 38-42	- ઈલેક્ટ્રિક મોડ્યુલ - I - Python માં પ્રોગ્રામિંગ
મોડ્યુલ 17	- સૂત્રો અને કાર્યોનો ઉપયોગ કરીને કામગીરી કરો	મોડ્યુલ 38-42	- ઈલેક્ટ્રિક મોડ્યુલ - II - જાવામાં પ્રોગ્રામિંગ
મોડ્યુલ 18	- મેઇલિંગ મેનેજ		
મોડ્યુલ 19	- પીવટ કોષ્ટકોનું સંચાલન કરો		
મોડ્યુલ 20	- પાવર પોઇન્ટ પ્રેઝન્ટેશન		
મોડ્યુલ 21	- રસ્તુતિઓ ફોર્મેટ કરો		

મોડ્યુલોમાં અભ્યાસક્રમ અને સામગ્રી એકબીજા સાથે જોડાયેલા છે. વિદ્યુત વિભાગમાં ઉપલબ્ધ વર્કસ્ટેશનોની સંખ્યા મશીનરી અને સાધનસામગ્રી દ્વારા મર્યાદિત હોવાથી, યોગ્ય શિક્ષણ અને શીખવાની ક્રમ રચવા માટે મોડ્યુલોમાં કસરતોને આંતરવી જરૂરી છે. સૂચનાનો ક્રમ સૂચનાના સમયપત્રકમાં આપવામાં આવ્યો છે જે પ્રશિક્ષકની માર્ગદર્શિકામાં સમાવિષ્ટ છે. અઠવાડિયાના 5 કામકાજના દિવસોના 25 વ્યવહારુ કલાકો સાથે દર મહિને 100 કલાક પ્રેક્ટિકલ ઉપલબ્ધ છે.

વેપાર વ્યવહારુ વિષયવસ્તુ

1 માટે 106 કસરતો દ્વારા કામ કરવાની પ્રક્રિયા યોક્કસ ઉદ્દેશ્યો સાથેનું વર્ષ કારણ કે દરેક કવાયતના અંતે શીખવાનું બહાર આવે છે તે આ પુસ્તક છે.

કૌશલ્યના ઉદ્દેશ્યો અને વ્યાયામ કરવા માટે જરૂરી સાધનો/યંત્રો, સાધનો/મશીનો અને સામગ્રી દરેક કવાયતની શરૂઆતમાં આપવામાં આવે છે. શોપફ્લોરમાં કૌશલ્ય તાલીમનું આયોજન સંબંધિત સિદ્ધાંતને સમર્થન આપવા માટે પ્રાયોગિક કસરતો/પ્રયોગોની શ્રેણી દ્વારા કરવામાં આવે છે. તાલીમાર્થીઓ સ્તર માટે યોગ્ય સંબંધિત જ્ઞાનાત્મક કૌશલ્યો સાથે ઈલેક્ટ્રીશિયન વેપારમાં તાલીમ મેળવે છે. તાલીમને વધુ અસરકારક બનાવવા અને ટીમમાં કામ કરવાની વૃત્તિ વિકસાવવા માટે ઓછામાં ઓછા પ્રોજેક્ટનો સમાવેશ કરવામાં આવ્યો છે. તાલીમાર્થીઓને તેમના મંતવ્યો વિસ્તૃત કરવામાં મદદ કરવા માટે જ્યાં જરૂરી હોય ત્યાં ચિત્રાત્મક, યોજનાકીય, વાયરિંગ અને સર્કિટ ડાયાગ્રામનો સમાવેશ કસરતોમાં કરવામાં આવ્યો છે. ડાયાગ્રામમાં વપરાતા પ્રતીકો બ્યુરો ઓફ ઈન્ડિયન સ્ટાન્ડર્ડ્સ (BIS) સ્પષ્ટીકરણોનું પાલન કરે છે.

આ માર્ગદર્શિકામાંના ચિત્રો, વિચારો અને વિભાવનાઓના વિઝ્યુઅલ પરિપ્રેક્ષ્યને તાલીમ આપવામાં મદદ કરે છે. કસરતો પૂરા કરવા માટે અનુસરવાની પ્રક્રિયાઓ પણ આપવામાં આવી છે. તાલીમાર્થીથી તાલીમાર્થી અને તાલીમાર્થીથી પ્રશિક્ષકની ક્રિયાપ્રતિક્રિયાને વધારવા માટે, મધ્યવર્તી કસોટીના પ્રશ્નોના વિવિધ સ્વરૂપોનો અભ્યાસમાં સમાવેશ કરવામાં આવ્યો છે.

કૌશલ્ય માહિતી

કૌશલ્ય ક્ષેત્રો જે પ્રકૃતિમાં પુનરાવર્તિત છે તે અલગ કૌશલ્ય માહિતી પત્રકો તરીકે આપવામાં આવે છે. કૌશલ્યો કે જે ચોક્કસ ક્ષેત્રોમાં વિકસાવવાની હોય છે તેનો સમાવેશ કસરતોમાં જ કરવામાં આવે છે. અભ્યાસક્રમને ધ્યાનમાં રાખીને વ્યાયામના ક્રમને પરિપૂર્ણ કરવા માટે કેટલીક પેટા કસરતો વિકસાવવામાં આવી છે.

આ મેન્યુઅલ ઓન ટ્રેડ પ્રેક્ટિકલ લેખિત સૂચનાત્મક સામગ્રી (WIM) નો ભાગ છે. જેમાં મેન્યુઅલ ઓન ટ્રેડ થિયરી અને અસાઈનમેન્ટ/ટેસ્ટનો સમાવેશ થાય છે.

© NIMI
NOT TO BE REPUBLISHED

સમાવિષ્ટી

અભ્યાસ સં.	અભ્યાસનું શીર્ષક	શિક્ષણ પરિણામો	પૃષ્ઠ સં.
	મોડ્યુલ 1 : સલામત કામ કરવાની પદ્ધતિઓ (Safe Working Practices)		
1.1.01 & 02	વિદ્યુત સલામતી (Electrical safety)	1	1
1.1.03 & 04	સલામતી પ્રેક્ટિસ - અગ્નિશામક (Safety practice - fire extinguishers)		2
	મોડ્યુલ 2 : કમ્પ્યુટર ઘટકોનો પરિચય (Assemble a Desktop PC)		
1.2.05 & 06	કમ્પ્યુટર્સનો પરિચય (Introduction to computers)	1	7
	મોડ્યુલ 3: વિન્ડોઝ ઓપરેટિંગ સિસ્ટમનો ઉપયોગ (Using Windows Operating System)		
1.3.07 - 13	CPU આર્કિટેક્ચર અને મધરબોર્ડનો પરિચય (Introduction to CPU architecture and mother board)	1	16
	મોડ્યુલ 4: કમ્પ્યુટર બેઝિક્સ અને સોફ્ટવેર ઇન્સ્ટોલેશન (Computer Basics & Software Installation)		
1.4.14	BIOS સેટિંગ્સ અને તેમના ફેરફારો જુઓ (View the BIOS settings and their modifications)		21
1.4.15	CMOS સેટઅપ અને વિન્ડોઝ ઇન્સ્ટોલેશન (Install Windows operating system)		29
1.4.16	કમ્પ્યુટર બેઝિક્સ અને સોફ્ટવેર ઇન્સ્ટોલેશન હાર્ડ ડિસ્કને ફોર્મેટ કરો અને પાર્ટીશન બનાવો (Format hard disk and create partition)	1	31
1.4.17	OS ઇન્સ્ટોલેશન દરમિયાન સામાન્ય હાર્ડવેર અને સોફ્ટવેર સમસ્યાઓને ઓળખો અને સુધારો (Identify and rectify common hardware and software issues during OS installation)		35
1.4.18	Windows માટે જરૂરી એપ્લિકેશન સોફ્ટવેર ઇન્સ્ટોલ કરો એટલે કે ઓફિસ પેકેજ, PDF રીડર, મીડિયા પ્લેયર વગેરે (Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc)		39
1.4.19	બ્લૂટૂથ અને Wi-Fi સેટિંગ્સને ગોઠવો (Configure Bluetooth and Wi-Fi settings)		40
1.4.20 & 21	ડીવીડી, સીડી અને બર્નિંગ ડીવીડી (DVDs, CDs and burning DVDs)		42
	મોડ્યુલ 5 : DOS કમાન્ડ લાઇન ઇન્ટરફેસ (DOS Command Line Interface)		
1.5.22	ડિરેક્ટરી સૂચિ માટે મૂળભૂત DOS આદેશોનો ઉપયોગ કરો (Use basic DOS commands for directory listing)	1	44
1.5.23	DOS આદેશોનો ઉપયોગ કરીને ફાઇલો અને ફોલ્ડર્સનું સંચાલન કરો (Manage files and folders using DOS commands)		47
	મોડ્યુલ 6: ઉબુન્ટુ લિનક્સ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરો અને મૂળભૂત Linux આદેશો ચલાવો (Install Ubuntu Linux operating system and execute basic Linux commands)		
1.6.24 & 25	Linux ઓપરેટિંગ સિસ્ટમનો પરિચય (Introduction to Linux operating system)	1	51
1.6.26 - 29	આદેશો અને વિવિધ સંપાદકોનું સંચાલન (Handling commands and various editors)		55
	મોડ્યુલ 7: વર્ડ પ્રોસેસિંગ સોફ્ટવેરનો ઉપયોગ (Using Word Processing Software)		
1.7.30 - 33	એમએસ વર્ડ 2010 થીયરી (MS WORD 2010 THEORY)	2	80
	મોડ્યુલ 8 :ફોર્મેટ દસ્તાવેજો (Format documents)		
1.8.34 -36	દાખલ કરો, ટેક્સ્ટ અને ફકરાઓને ફોર્મેટ કરો, દસ્તાવેજ વિભાગો બનાવો અને ગોઠવો (Insert, format text and paragraphs, Create and configure document sections)	2	89

અભ્યાસ સં.	અભ્યાસનું શીર્ષક	શિક્ષણ પરિણામો	પૃષ્ઠ સં.
1.9.37 - 39	મોડ્યુલ 9 : કોષ્ટકો અને સૂચિઓનું સંચાલન કરો (Manage Tables and Lists) કોષ્ટકો બનાવો, સંશોધિત કરો (Create, modify tables)	2	91
1.10.40&41	મોડ્યુલ 10 : સંદર્ભો બનાવો અને મેનેજ કરો (Create and Manage References) સંદર્ભ તત્વો અને કોષ્ટકો બનાવો અને મેનેજ કરો (Create and manage reference elements and tables)	2	93
1.11.42-45	મોડ્યુલ 11 : ગ્રાફિક એલિમેન્ટ્સ (Manage Graphic Elements) ઇન્સર્ટ, ફોર્મેટ ચિત્રો અને ટેક્સ્ટ બોક્સનું સંચાલન કરો (Insert, format illustrations and text boxes)	2	94
1.12.46&47	મોડ્યુલ 12 : દસ્તાવેજ સહયોગ મેનેજ કરો (Manage Document Collaboration) ટિપ્પણીઓમાં ફેરફાર ટ્રેકિંગ અને મેઇલિંગનું સંચાલન (Manage comments change tracking and mailings)	2	96
1.13.48	મોડ્યુલ 13: મેઇલિંગ મેનેજ કરો (Manage Mailings) મેઇલ મર્જ કરો (Perform mail merge)	2	100
1.14.49-54	મોડ્યુલ 14 : સ્પ્રેડ શીટ એપ્લિકેશન વર્કશીટ્સ અને વર્કબુક્સનું સંચાલન કરો (Spread Sheet Application, Manage Worksheets and Workbooks) MS Excel માં ફાઇલો ખોલો (Open files in MS Excel)	3	103
1.15.55-57	મોડ્યુલ 15 : ડેટા કોષો અને શ્રેણીઓનું સંચાલન કરો (Manage Data Cells and Ranges) ડેટાની હેરફેર કરો (Manipulate data)	3	104
1.16.58-60	મોડ્યુલ 16 : કોષ્ટકો અને કોષ્ટક ડેટા મેનેજ કરો (Manage Tables and Table Data) કોષ્ટકો બનાવો અને ફોર્મેટ કરો (Create and format tables)	3	106
1.17.61-63	મોડ્યુલ 17 : સૂત્રો અને કાર્યોનો ઉપયોગ કરીને કામગીરી કરો (Perform Operations using Formulas and Functions) MS-Excel 2010 માં કાર્યો અને સૂત્રો (Functions and formulas in MS-Excel 2010)	3	107
1.18.64-66	મોડ્યુલ 18 : મેઇલિંગ મેનેજ (Manage Mailings) ચાર્ટ્સ મેનેજ કરો (Manage Charts)	3	112
1.19.67	મોડ્યુલ 19 : પીવટ કોષ્ટકોનું સંચાલન કરો (Manage Pivot Tables) પિવટ કોષ્ટકો બનાવો (Create Pivot Tables)	3	114
1.20.68-74	મોડ્યુલ 20 : પાવર પોઇન્ટ પ્રેઝન્ટેશન (Power Point Presentations) પિવટ કોષ્ટકો બનાવો (Open files in MS PowerPoint Presentations)	4	115
1.21.75-77	મોડ્યુલ 21 : પ્રસ્તુતિઓ ફોર્મેટ કરો (Format Presentations) પિવટ કોષ્ટકો બનાવો (Insert, Format text and paragraphs)	4	116
1.22.78-80	મોડ્યુલ 22 : કોષ્ટકો અને બુલેટેડ ટેક્સ્ટ મેનેજ કરો (Manage Tables and Bulleted Text) પિવટ કોષ્ટકો બનાવો (Create tables, modify tables, modify bulleted text)	4	117
1.23.81-83	મોડ્યુલ 23 : ગ્રાફિક તત્વોનું સંચાલન કરો (Manage Graphic Elements) પિવટ કોષ્ટકો બનાવો (Insert illustrations, Format illustrations and text boxes)	4	118
1.24.84&85	મોડ્યુલ 24 : ઑડિઓ અને વિડિયો ઘટકોનું સંચાલન કરો (Manage Audio & Video Elements) ઑડિઓ અને વિડિયો તત્વો (Audio & Video Elements)	4	120
1.25.86&87	મોડ્યુલ 25 : સંક્રમણો અને એનિમેશન મેનેજ કરો (Manage Transitions and Animations) ઑડિઓ અને વિડિયો તત્વો (Add slide transitions & animations)	4	121

અભ્યાસ સં.	અભ્યાસનું શીર્ષક	શિક્ષણ પરિણામો	પૃષ્ઠ સં.
1.26.88	મોડ્યુલ 26 : સહયોગ મેનેજ કરો (Manage Collaboration) ઓડિઓ અને વિડિયો તત્વો (Add and manage comments)	4	122
1.27.89-91 1.27.92-96	મોડ્યુલ 27 : MySQL ઈન્સ્ટોલ કરો (Demonstrate on MySQL) MySQL માં ડેટાબેઝ ઈન્સ્ટોલ કરો, મુશ્કેલીનિવારણ કરો, બનાવો અને ઉપયોગ કરો (Install, Troubleshoot, Create and Use of database in MySQL) ર્મલાઈઝેશન નિયમો, વિવિધ ડેટાટાઈપ્સ, ડેટા ઇન્ટિગ્રિટી, કીકીએલ, કીએમએલ અને કીસીએલ સ્ટેટમેન્ટ્સનો ઉપયોગ કરીને ડેટાબેઝ ડિઝાઇન કરવી પ્રાથમિક કી અને વિદેશી કી (Designing database using normalization rules,various datatypes,data integrity,DDL,DML&DCL Statements Enforcing Primary key and Foreign key)	5	123 125
1.28.97&98	મોડ્યુલ 28 : પ્રશ્નો પર પ્રદર્શન (Demonstrate on Queries) પ્રશ્નો દાખલ કરો અને કાઢી નાખો ક્વેરીઝ અપડેટ કરો (Insert and delete queries Update queries)	5	127
1.29.99&100	મોડ્યુલ 29 : ફાંચો પર પ્રદર્શન કરો (Demonstrate on Functions) નંબર, તારીખ અને અક્ષર ફંક્શનનો ઉપયોગ કરીને ફાંચો પર નિદર્શન કરો, પેટા ક્વેરી કરીને જૂથ બનાવો (Using the Number, Date and Character functions, group by having, sub query)	5	128
1.30.101-112	મોડ્યુલ 30 : કમ્પ્યુટર નેટવર્ક સેટ કરો અને ગોઠવો (Set-up & Configure a Computer Network) કમ્પ્યુટરને નેટવર્કથી કનેક્ટ કરો અને ઉપકરણો એટલે કે પ્રિન્ટર્સ, ફાઈલો, ફોલ્ડર્સ અને ડ્રાઈવ્સ શેર કરો (Connect a computer to a network and share Devices i.e. Printers, files, folders and drives)	6	130
1.31.113 1.31.114	મોડ્યુલ 31 : HTML ટૅગ્સનો ઉપયોગ કરીને સરળ સ્થિર વેબ પૃષ્ઠો બનાવો (Create Simple Static Web Pages using HTML Tags) વેબ ડિઝાઇનિંગ (Web designing) CMS અને વેબ ઓથરિંગ ટૂલ્સનો પરિચય (Introduction to CMS and web authoring tools)	7	139 159
1.32.115 1.32.116&117 1.32.118 1.32.119 1.32.120 1.32.121	મોડ્યુલ 32 : JavaScript HTML પૃષ્ઠોમાં JavaScript એમ્બેડ કરો (JavaScript Embed JavaScript in HTML Pages) JavaScript ને સમજવું (Understanding JavaScript) JavaScript વેરીએબલ અને ડેટા પ્રકારોનો ઉપયોગ કરવો (Using JavaScript Variable and data types) જાવાસ્ક્રિપ્ટમાં કંટ્રોલ સ્ટેટમેન્ટ્સ, લૂપ્સ અને પોપઅપ બોક્સ (Control statements, Loops and Popup boxes in JavaScript) JavaScript માં અરે (Arrays in JavaScript) JavaScript નો ઉપયોગ કરીને ગતિશીલ HTML પૃષ્ઠો વિકસાવો (Develop dynamic HTML pages using JavaScript) IIS નો ઉપયોગ કરીને વેબ પ્રોજેક્ટનો ઉપયોગ કરો (Deploy web project using IIS)	8	161 165 172 178 186 203
1.33.122-129	મોડ્યુલ 33 : એક્સેલનો ઉપયોગ કરીને ડેટા વિઝ્યુલાઈઝેશન અથવા વિશ્લેષણ (Data Visualization or Analysis using Excel) અદ્યતન ફોર્મ્યુલા અને મેક્રો બનાવો (Create advanced formulas and macros)	9	211

અભ્યાસ સં.	અભ્યાસનું શીર્ષક	શિક્ષણ પરિણામો	પૃષ્ઠ સં.
1.34.130-132	મોડ્યુલ 34 :ઉત્પાદનો અને સેવાઓને ઓળખવા માટે ઈ-કોમર્સ સાઈટ્સ બ્રાઉઝ કરો (Browse E-Commerce Sites to Identify Products & Services) અદ્યતન ફોર્મ્યુલા અને મેક્રો બનાવો (E-Commerce scope and benefits)	10	218
1.34.133	ઈ-કોમર્સ સાઈટ પર વ્યવહારો કરો (Undertake transactions on an e-commerce site)		220
1.34.134-137	ઈ કોમર્સ સુરક્ષા સમસ્યાઓ અને પેમેન્ટ ગેટવે (E-Commerce Security issues and Payment Gateways)		221
1.35.138	મોડ્યુલ 35 :વાયરસ, સ્પાયવેર અને અન્ય દૂષિત કોડથી માર્હિતી, કમ્પ્યુટર અને નેટવર્કને સુરક્ષિત કરો (Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code) માહિતી સુરક્ષા અને ધમકીઓનું વિહંગાવલોકન (Overview of information security and threats)	11	223
1.35.139	નેટવર્ક્સને સુરક્ષિત કરો (Privacy Protection and IT Act)		251
1.36.140-143	મોડ્યુલ 36 :ક્લાઉડ કમ્પ્યુટિંગ (Cloud Computing) ક્લાઉડ સેવાઓ સાથે કામ કરે છે (Working with Cloud Services)	12	256
1.37.144-145	મોડ્યુલ 37 :એપ્લિકેશન ડેવલપ કરો અને એપ્લીકેશન ડેવલપમેન્ટ લાઈફ સાયકલ કરો (Develop an application and perform the Application Development Life Cycle) એપ્લિકેશન ડેવલપમેન્ટ જીવન ચક્રના તબક્કાઓ ઓળખો (Identify Phases of the Application Development Life Cycle)	12	260
1.38.01 & 02	મોડ્યુલ 38 - 42 : ઇલેક્ટિવ મોડ્યુલ - I - Python માં પ્રોગ્રામિંગ (Elective Module I - Programming in Python) પ્રોગ્રામિંગ ભાષા (પાયથોન) કમાન્ડ લાઈનમાંથી પાયથોનનો ઉપયોગ કરો (Programming language (Python) Use Python from command line)	13	264
1.39.03 - 05	ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Perform Operations using Data Types and Operators)		266
1.40.06&07	ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Control Flow with Decisions and Loops)		270
1.41.08&09	ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Document and Structure Code)		274
1.42.10&11	મોડ્યુલો અને ટૂલ્સનો ઉપયોગ કરીને કામગીરી કરો (Perform Operations Using Modules and Tools)		292
1.38.01-03	મોડ્યુલ 38 - 42 : ઇલેક્ટિવ મોડ્યુલ - II - જાવામાં પ્રોગ્રામિંગ (Elective Module II Programming in JAVA) ઑબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ અને જાવા લેંગ્વેજ (Object Oriented Programming and JAVA Language)	14	296
1.39.04-07	JAVA પ્રોગ્રામ લખવાનું નિદર્શન કરો (Demonstrate writing JAVA programs)		300
1.40.08-16	જાવા પ્રોગ્રામ ફ્લો કંટ્રોલ (JAVA Program Flow Control)		313
1.41.17-23	જાવા વર્ગો, ઓવરલોડિંગ અને વારસો (JAVA Classes, Overloading and Inheritance)		318
1.42.24-30	JAVA માં અમૂર્ત વર્ગો અને ઇન્ટરફેસો (Abstract Classes and Interfaces in JAVA)		325

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

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.

		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.

		• 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, a n d 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 a n d endnotes • Modify footnote and endnote properties • Create and m o d i f y bibliography citation sources • Insert citations f o r 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 SmartArt graphics • Insert screenshots and screen clippings • Insert text boxes 43.Format illustrations and text boxes (3 Hrs) • Apply artistic effects • Apply picture effects a n d 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	

		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

		54. Configure content of 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	

		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 Power Point 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

		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	

		• 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	

		• 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.

		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.

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 JavaScript.
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

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 other payment 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.

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 (18Hrs) 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.

		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.			

વિદ્યુત સલામતી (Electrical safety)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- જીવંત વાયરના સંપર્કમાં રહેલી વ્યક્તિને કેવી રીતે બચાવવી તે સમજાવો.

વિદ્યુત આંચકાની તીવ્રતા શરીરમાંથી પસાર થતા પ્રવાહના સ્તર અને સંપર્કના સમય પર આધારિત રહેશે. વિલંબ કરશો નહીં, તાત્કાલિક કાર્ય કરો. ખાતરી કરો કે ઇલેક્ટ્રિક કરંટ ડિસ્કનેક્ટ થઈ ગયો છે.

જો અકસ્માત હજુ પણ પુરવઠા સાથે સંપર્કમાં છે - તોડી નાખો ક્યાં તો પાવર બંધ કરીને, પ્લગને દૂર કરીને સંપર્ક કરો અથવા કેબલને મુક્ત કરો. જો નહિં, તો કેટલાક ઇન્સ્યુલેટિંગ પર ઊભા રહો સામગ્રી જેમ કે શુષ્ક લાકડું, રબર અથવા પ્લાસ્ટિક, અથવા ઉપયોગ તમારી જાતને ઇન્સ્યુલેટ કરવા અને તોડવા માટે જે પણ હાથમાં છે વ્યક્તિને દબાવ કરીને અથવા ખેંચીને સંપર્ક કરો. (અંજીર 1 અને 2) ડબલ પોલ સ્વિચ-ફ્યુઝમાં ન્યુટ્રલ લિંક્સ આપવામાં આવે છે જેનો ઉપયોગ લાઈટિંગ કંટ્રોલ માટે થાય છે અને ન્યુટ્રલમાં કોઈ ફ્યુઝ આપવામાં આવતું નથી.

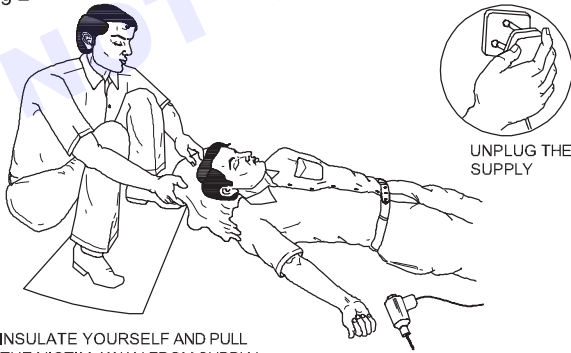
જો તમે અન-ઇન્સ્યુલેટેડ રહેશો, તો પીડિતને તમારી સાથે સ્પર્શ કરશો નહીં જ્યાં સુધી સર્કિટ મૃત અથવા વ્યક્તિ ન બને ત્યાં સુધી ખુલ્લા હાથ સાધનોથી દૂર ખસેડવામાં આવ્યા હતા.

Fig 1



CP20N110111

Fig 2



INSULATE YOURSELF AND PULL THE VICTIM AWAY FROM SUPPLY

CP20N110112

જો પીડિત ઊંચો હોય, તો તેને રોકવા માટે પગલાં લેવા જોઈએ પડવાથી અથવા ઓછામાં ઓછું તેને સુરક્ષિત પતન કરો.

પીડિત પર ઇલેક્ટ્રીક બળે મોટા વિસ્તારને આવરી શકતા નથી પરંતુ ઊંડા બેઠેલા હોઈ શકે છે. તમે જે કરી શકો તે વિસ્તારને આવરી લેવાનું છે સ્વચ્છ, જંતુરહિત ડ્રેસિંગ સાથે અને આંચકા માટે સારવાર. મેળવો શક્ય તેટલી ઝડપથી નિષ્ણાતની મદદ.

જો ઇજાગ્રસ્ત વ્યક્તિ બેભાન હોય પરંતુ શ્વાસ લઈ રહ્યો હોય, તો ગરદન, છાતી અને કમરની આસપાસના કપડાં ઢીલા કરો અને તેને મૂકો. પુનઃપ્રાપ્તિ સ્થિતિમાં અકસ્માત. (ફિગ 3)

Fig 3



KEEP THE VICTIM AT EASE

CP20N110113

શ્વાસ અને પલ્સ રેટ પર સતત તપાસ કરો.

અકસ્માતને ગરમ અને આરામદાયક રાખો. (ફિગ 4)

બેભાન વ્યક્તિને મોં દ્વારા કંઈપણ ન આપો.

બેભાન વ્યક્તિને અડધા વિના છોડશો નહીં.

જો જાનહાનિ શ્વાસ ન લઈ રહી હોય - તરત જ કાર્ય કરો - બગાડો નહીં

Fig 4



CP20N110114

સલામતી પ્રેક્ટિસ - અગ્નિશામક (Safety practice - fire extinguishers)

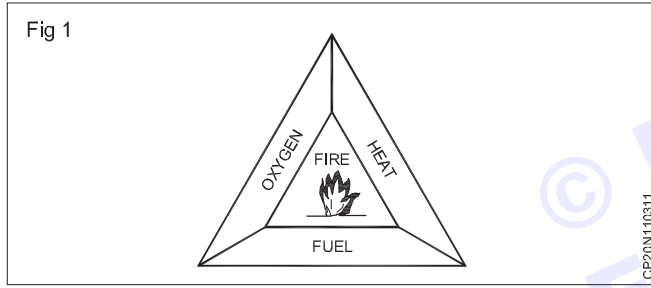
ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- જીવંત વાયરના સંપર્કમાં રહેલી વ્યક્તિને કેવી રીતે બચાવવી તે સમજાવો.
- વિવિધ પ્રકારના અગ્નિશામક ઉપકરણોને અલગ પાડો.
- આગના વર્ગના આધારે ઉપયોગમાં લેવાતા અગ્નિશામકનો યોગ્ય પ્રકાર નક્કી કરો.
- આગ લાગવાની ઘટનામાં અપનાવવામાં આવતી સામાન્ય પ્રક્રિયાનું વર્ણન કર.

આગ એ જ્વલનશીલ સામગ્રીને બાળી નાખવું છે. એક માં આગ અનિચ્છનીય સ્થળ અને અનિચ્છનીય પ્રસંગે અને એક અનિયંત્રિત જથ્થો નુકસાન અથવા નાશ કરી શકે છે મિલકત અને સામગ્રી. તે લોકોને ઈજા પહોંચાડી શકે છે, અને ક્યારેક જાનહાનિ પણ થાય છે. તેથી, દરેક પ્રયાસ આગ અટકાવવા માટે બનાવવી જોઈએ. જ્યારે આગ ફાટી નીકળી છે શોધ્યું, તે દ્વારા નિયંત્રિત અને બુઝાઈ ગયેલ હોવું જ જોઈએ તાત્કાલિક સુધારાત્મક પગલાં

શું આગને અટકાવવી શક્ય છે? હા, આગને અટકાવી શકાય છે આગનું કારણ બને તેવા ત્રણ પરિબલોમાંથી કોઈપણને દૂર કરવું.

નીચેના ત્રણ પરિબલો છે જે હાજર હોવા જોઈએ આગ ચાલુ રાખવા માટેનું મિશ્રણ. (ફિગ 1)



બળતણ: કોઈપણ પદાર્થ, પ્રવાહી, ઘન અથવા વાયુ બળી જશે, જો ત્યાં હશે ઓક્સિજન અને પર્યાપ્ત ઊંચા તાપમાન છે.

ગરમી: દરેક બળતણ ચોક્કસ તાપમાને બળવાનું શરૂ કરશે. તે બદલાય છે અને બળતણ પર આધાર રાખે છે. ઘન અને પ્રવાહી બંધ કરે છે વરાળ જ્યારે ગરમ થાય છે, અને તે આ વરાળ છે જે સળગે છે. કેટલાક પ્રવાહીને ગરમ કરવાની જરૂર નથી કારણ કે તેઓ બંધ કરે છે સામાન્ય ઓરડાના તાપમાને વરાળ 150 કહે છે, દા.ત. પેટ્રોલ

ઓક્સિજન: સામાન્ય રીતે રાખવા માટે હવામાં પૂરતી માત્રામાં હોય છે આગ બળી રહી છે.

આગ ઓલવવી: આમાંથી કોઈપણને અલગ કરવું અથવા દૂર કરવું સંયોજનના પરિબલો આગને ઓલવી નાખશે. ત્યાં આ હાંસલ કરવાની ત્રણ મૂળભૂત રીતો છે.

- બળતણની** આગને ભૂખ્યા રહેવાથી આ તત્વ દૂર થઈ જાય છે.
- સ્પોધરિંગ** - એટલે કે ના પુરવઠામાંથી આગને અલગ કરો ઓક્સિજનને ફીણ, રેતી વગેરેથી બ્લેન્કેટ કરીને.

- ઠંડક** - તાપમાન ઘટાડવા માટે પાણીનો ઉપયોગ કરો. દૂર કરી રહ્યા છીએ આમાંથી કોઈપણ એક પરિબલ આગને ઓલવી નાખશે.

આગ અટકાવવી: મોટાભાગની આગ નાની સાથે શરૂ થાય છે ફાટી નીકળે છે કે જ્યાં સુધી તેમની પાસે સુરક્ષિત ન હોય ત્યાં સુધી કોઈનું ધ્યાન ન જાય પકડી રાખવું. મોટાભાગની આગને વધુ કાળજીથી અને દ્વારા અટકાવી શકાય છે કેટલાક સરળ સામાન્ય જ્ઞાન નિયમોનું પાલન કરો.

જ્વલનશીલ કચરો (પલાળેલા કપાસનો કચરો તેલ, સ્કેપ લાકડું, કાગળ, વગેરે સાથે) વિચિત્ર ખૂણામાં આગ છે જોખમ. સંગ્રહ બિંદુઓ પર ઈનકાર દૂર કરવો જોઈએ.

વિદ્યુત સાધનોમાં આગ લાગવાનું કારણ દુરુપયોગ અથવા છે ઉપેક્ષા છૂટક જોડાણો, ખોટી રીતે રેટ કરેલ ફ્યુઝ, ઓવરલોડ સર્કિટ ઓવરહિટીંગનું કારણ બને છે જે બદલામાં આગ તરફ દોરી શકે છે. કેબલ્સમાં કંડકતર વચ્ચેના ઈન્સ્યુલેશનને નુકસાન થાય છે આગ

કપડાં અને આગ લાગી શકે તેવી અન્ય કોઈપણ વસ્તુ હોવી જોઈએ દ્વેષીઓથી સારી રીતે દૂર રાખવામાં આવે છે. ખાતરી કરો કે હીટર છે કામકાજના દિવસના અંતે બંધ.

અત્યંત જ્વલનશીલ પ્રવાહી અને પેટ્રોલિયમ મિશ્રણ (પાતળા, એડહેસિવ સોલ્યુશન્સ, સોલવન્ટ્સ, કેરોસીન, સ્પિરિટ, એલપીજી ગેસ વગેરે) જ્વલનશીલ સામગ્રીના સંગ્રહમાં સંગ્રહિત થવો જોઈએ વિસ્તાર.

બ્લો લેમ્પ્સ અને ટોર્ચ જ્યારે તે સળગતા છોડવા જોઈએ નહીં ઉપયોગમાં નથી.

આગ ઓલવવી: આગને ચાર પ્રકારમાં વર્ગીકૃત કરવામાં આવી છે બળતણની પ્રકૃતિની શરતો

આગના વિવિધ પ્રકારો (ફિગ 2, ફિગ 3 ફિગ 4 અને ફિગ 5) અલગ અલગ રીતે અને અલગ-અલગ બુઝાવવાના એજન્ટો સાથે વ્યવહાર કરવો.

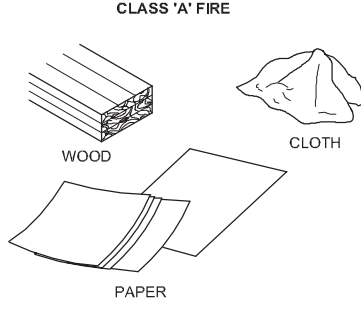
બુઝાવવાનું એજન્ટ એ વપરાયેલી સામગ્રી અથવા પદાર્થ છે આગ બુઝાવવા માટે, અને સામાન્ય રીતે (પરંતુ હંમેશા નહીં) સમાયેલ છે આગમાં છંટકાવ માટે રીલીઝ મિકેનિઝમ સાથે અગ્નિશામકમાં.

બુઝાવવા માટે યોગ્ય પ્રકારના એજન્ટને જાણવું મહત્વપૂર્ણ છે આગનો ચોક્કસ પ્રકાર; ખોટા એજન્ટનો ઉપયોગ કરી શકે છે વસ્તુઓ ખરાબ. 'ઇલેક્ટ્રિકલ ફાયર' માટે કોઈ વર્ગીકરણ નથી જેમ કે, કારણ કે આ માત્ર સામગ્રીમાં આગ છે જ્યાં વીજળી હાજર છે.

બળતણ

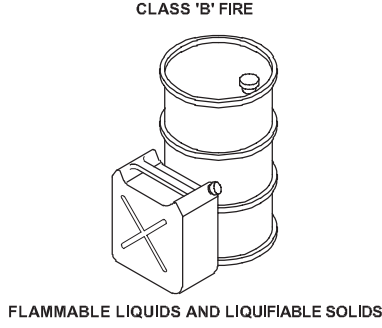
બુઝાવવાનું

Fig 2



સૌથી વધુ અસરકારક એટલે કે. પાણી સાથે ઠંડક. આગના પાયા પર પાણીના જેટનો છંટકાવ કરવો જોઈએ અને પછી ધીમે ધીમે ઉપરની તરફ

Fig 3

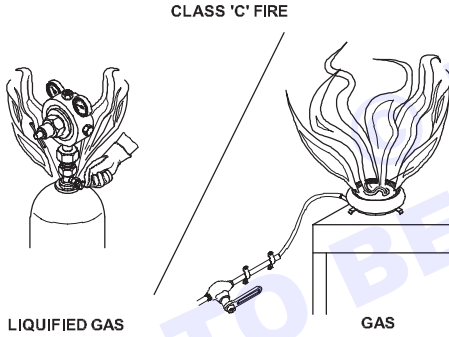


smothered જોઈએ. સમગ્રને આવરી લેવાનો હેતુ છે બર્નિંગ પ્રવાહીની સપાટી. આની અસર છે આગને ઓક્સિજનનો પુરવઠો બંધ કરવો.

સળગતા પ્રવાહી પર પાણીનો ઉપયોગ ક્યારેય ન કરવો જોઈએ.

આ પ્રકારની આગ પર ફોમ, ડ્રાય પાવડર અથવા CO2 નો ઉપયોગ થઈ શકે છે.

Fig 4



લિક્વિફાઈડ સાથે વ્યવહાર કરવામાં અત્યંત સાવધાની જરૂરી છે વાયુઓ વિસ્ફોટ અને અચાનક થવાનું જોખમ છે સમગ્ર આસપાસના વિસ્તારમાં આગ ફાટી નીકળી. જો એક સાધન સિલિન્ડરમાંથી ખવડાવવાથી આગ લાગે છે - પુરવઠો બંધ કરો ગેસનું. સૌથી સલામત અભ્યાસક્રમ એલાર્મ વગાડવાનો છે અને પ્રશિક્ષિત કર્મચારીઓ દ્વારા આગનો સામનો કરવા માટે છોડી દો.

આ પ્રકારની આગ પર સુકા પાવડરનો ઉપયોગ કરવામાં આવે છે.

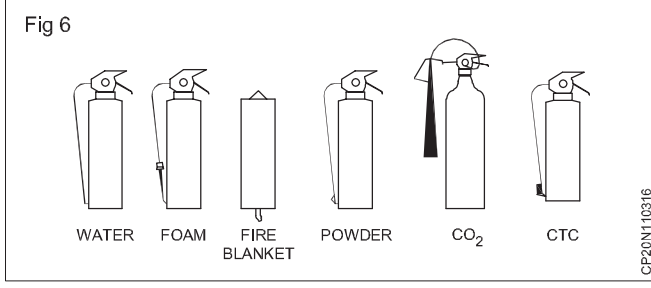
Fig 5



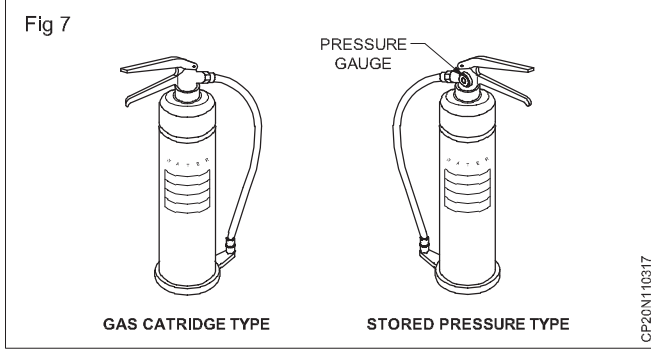
લિક્વિફાઈડ સાથે વ્યવહાર કરવામાં અત્યંત સાવધાની જરૂરી છે વાયુઓ વિસ્ફોટ અને અચાનક થવાનું જોખમ છે સમગ્ર આસપાસના વિસ્તારમાં આગ ફાટી નીકળી. જો એક સાધન સિલિન્ડરમાંથી ખવડાવવાથી આગ લાગે છે - પુરવઠો બંધ કરો ગેસનું. સૌથી સલામત અભ્યાસક્રમ એલાર્મ વગાડવાનો છે અને પ્રશિક્ષિત કર્મચારીઓ દ્વારા આગનો સામનો કરવા માટે છોડી દો.

આ પ્રકારની આગ પર સુકા પાવડરનો ઉપયોગ કરવામાં આવે છે.

અગ્નિશામકના ઘણા પ્રકારો વિવિધ સાથે ઉપલબ્ધ છે આગના વિવિધ વર્ગોનો સામનો કરવા માટે ‘એજન્ટો’ ઓલવવા. (ફિગ 6)



પાણીથી ભરેલા અગ્નિશામક: ની બે પદ્ધતિઓ છે કામગીરી (ફિગ 7)



- ગેસ કારતૂસ પ્રકાર
- સંગ્રહિત દબાણ પ્રકાર

ઓપરેશનની બંને પદ્ધતિઓ સાથે સાવ થઈ શકે છે જરૂરીયાત મુજબ વિક્ષેપિત, સમાવિષ્ટોનું સંરક્ષણ અને બિનજરૂરી પાણીના નુકસાનને અટકાવવું.

ફોમ અગ્નિશામક (ફિગ 8): આ સંગ્રહિત હોઈ શકે છે દબાણ અથવા ગેસ કારતૂસ પ્રકારો. હંમેશા ઓપરેટિંગ તપાસો ઉપયોગ કરતા પહેલા અગ્નિશામક પરની સૂચનાઓ.

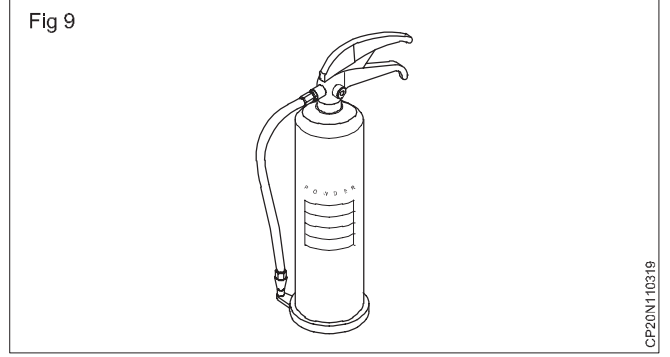


સૌથી યોગ્ય માટે

- જ્વલનશીલ પ્રવાહી આગ
- પ્રવાહી આગ ચાલી રહી છે.

જ્યાં વિદ્યુત ઉપકરણો સામેલ હોય ત્યાં આગ પર ઉપયોગ કરવો જોઈએ નહીં.

ડ્રાય પાઉડર એક્સટિંગ્વિશર્સ (ફિગ 9): એક્સટિંગ્વિશર્સ ફીટ કરવામાં આવ્યા છે સૂકા પાવડર સાથે ગેસ કારતૂસ અથવા સંગ્રહિત હોઈ શકે છે દબાણ પ્રકાર. દેખાવ અને કામગીરીની પદ્ધતિ છે પાણીથી ભરેલાની જેમ જ. મુખ્ય વિશિષ્ટ લક્ષણ કાંટો આકારની નોઝલ છે. વર્ગ ડી આગનો સામનો કરવા માટે પાવડર વિકસાવવામાં આવ્યા છે.



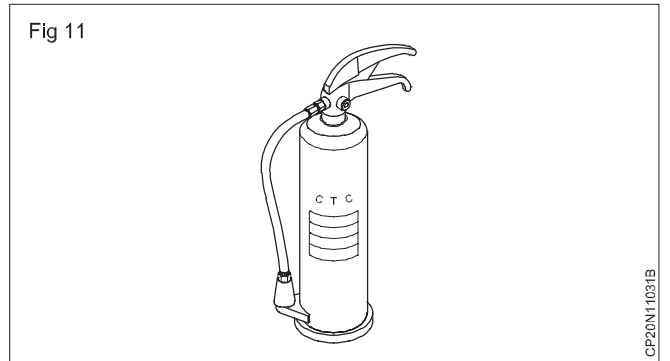
કાર્બન ડાયોક્સાઇડ (CO₂): આ પ્રકાર સરળતાથી ઓળખાય છે વિશિષ્ટ આકારના ડિસ્ચાર્જ હોર્ન દ્વારા. (ફિગ 10).



વર્ગ B આગ માટે યોગ્ય. જ્યાં દૂષણ હોય ત્યાં શ્રેષ્ઠ અનુકૂળ થાપણો દ્વારા ટાળવું જોઈએ. માં સામાન્ય રીતે અસરકારક નથી ખુલ્લી હવા.

કન્ટેનર પરની ઓપરેટિંગ સૂચનાઓ હંમેશા તપાસો ઉપયોગ કરતા પહેલા. ઓપરેશનના વિવિધ ગેજેટ્સ સાથે ઉપલબ્ધ જેમ કે - ફૂદકા મારનાર, લીવર, ટ્રિગર વગેરે.

હેલોન અગ્નિશામક (ફિગ 11): આ અગ્નિશામકો હોઈ શકે છે કાર્બન-ટેટ્રાક્લોરાઇડ અને બ્રોમોકલોરોડીફ્લોરોમેથેન (BCF) થી ભરેલું છે. તે ક્યાં તો ગેસ કારતૂસ અથવા સંગ્રહિત દબાણ પ્રકાર હોઈ શકે છે.



તેઓ નાની આગને ઓલવવામાં વધુ અસરકારક છે પ્રવાહી રેડવું. આ અગ્નિશામકો ખાસ કરીને યોગ્ય છે અને રસાયણો તરીકે ઇલેક્ટ્રિકલ સાધનો પર વાપરવા માટે સલામત ઇલેક્ટ્રિકલ બિન-વાહક છે.

આ અગ્નિશામકો દ્વારા આપવામાં આવેલ ધૂમાડો છે ખતરનાક, ખાસ કરીને મર્યાદિત જગ્યામાં.

આગની ઘટનામાં સામાન્ય પ્રક્રિયા:

- એલાર્મ વગાડો.
- તમામ મશીનરી અને પાવર (ગેસ અને વીજળી) બંધ કરો.

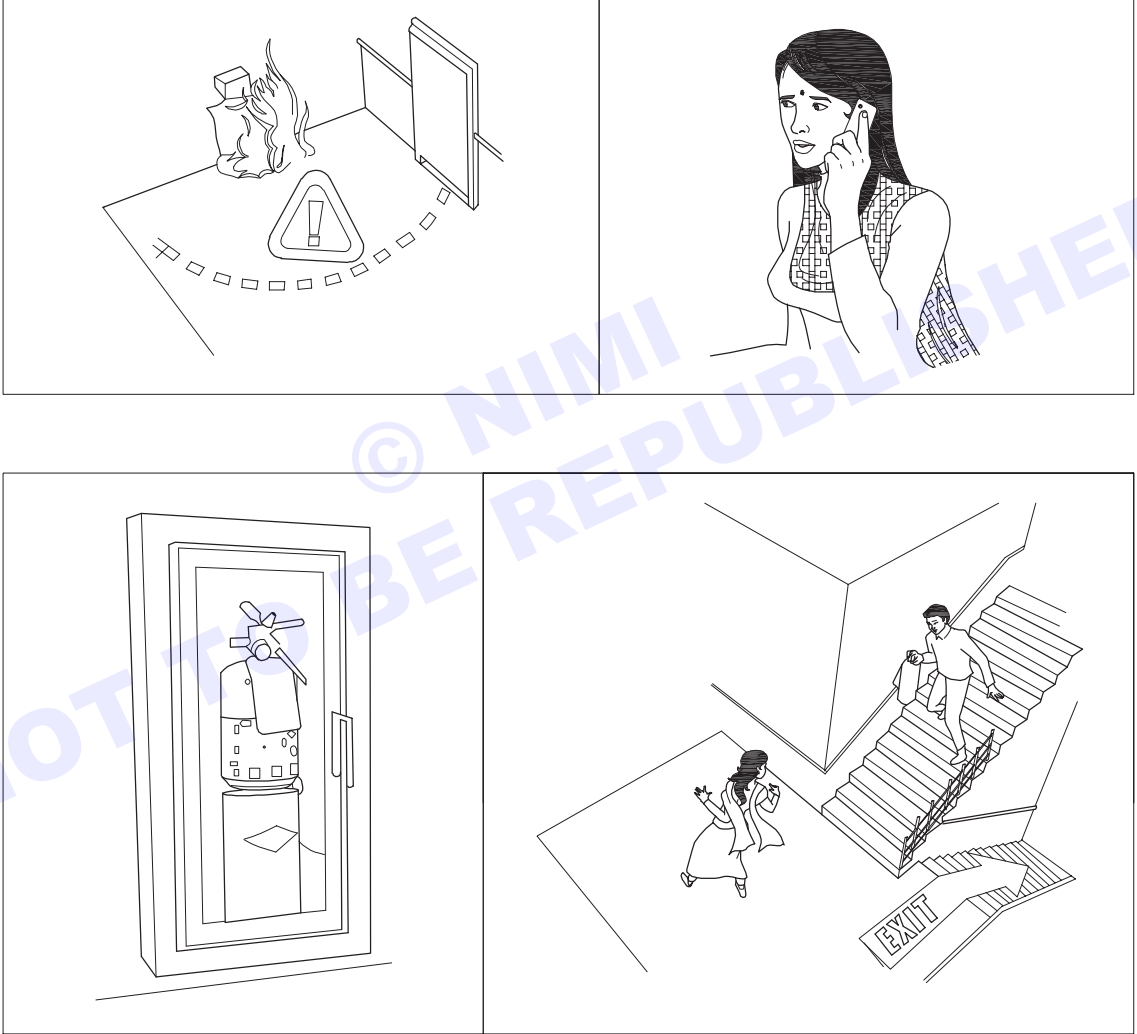
- દરવાજા અને બારીઓ બંધ કરો, પરંતુ લોક કે બોલ્ટ ન લગાવો તેમને આ આગને આપવામાં આવતા ઓક્સિજનને મર્યાદિત કરશે અને અટકાવશે તેનો ફેલાવો.
- જો તમે સુરક્ષિત રીતે કરી શકો તો આગનો સામનો કરવાનો પ્રયાસ કરો. ફસાવવાનું જોખમ ન લો.
- આગ સામે લડવામાં સામેલ ન હોય તેવા કોઈપણ વ્યક્તિએ છોડી દેવું જોઈએ શાંતિથી ઈમરજન્સી એક્ઝિટનો ઉપયોગ કરીને અને પર જાઓ નિયુક્ત એસેમ્બલી બિંદુ. આ કરવામાં નિષ્ફળતા મળી શકે છે મતલબ કે અમુક વ્યક્તિ અને માટે બિનહિસાબી છે અન્ય લોકોને મુશ્કેલીમાં મૂકવી પડી શકે છે તેને અથવા તેણીને પોતાને માટે જોખમમાં શોધવું.

અગ્નિશામક સાધનો પર પ્રેક્ટિસ કરો (Practice on fire extinguishers)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- આગના પ્રકાર અનુસાર અગ્નિશામકની પસંદગી વિશે જણાવો
- અગ્નિશામકની કામગીરીની પદ્ધતિ જણાવો
- આગ કેવી રીતે ઓલવવી તે સમજાવો.

Fig 1



CP20N110321

પ્રક્રિયા (PROCEDURE) (Fig 1)

- આગ, અગ્નિ, અગ્નિની ભૂમો પાડીને ઘેરાયેલા લોકોને ચેતવણી આપો આગનું અવલોકન કરો.
- ફાયર સર્વિસને જાણ કરો અથવા તરત જ જાણ કરવાની વ્યવસ્થા કરો.
- ખુલ્લી કટોકટી અસ્તિત્વમાં છે અને તેમને દૂર જવા માટે કહો.
- વીજ વીજ પુરવઠો “બંધ” કરો.

લોકોને આગની નજીક જવા દો નહીં

- પૃથ્થકરણ કરો અને આગના પ્રકારને ઓળખો. કોષ્ટક 1 નો સંદર્ભ લો.

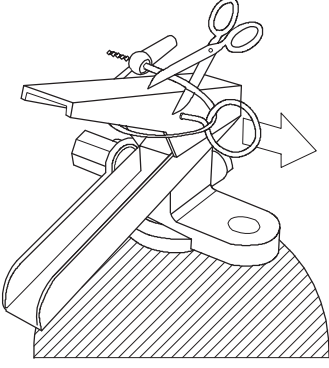
કોષ્ટક-1

Class 'A'	Wood, paper, cloth, solid material
Class 'B'	Oil based fire (grease, gasoline, oil) liquefiable gases
Class 'C'	Gas and liquefiable gases
Class 'D'	Metals and electrical equipment

ઘારો કે આગ 'B' છે; પ્રકાર (જ્વલનશીલ પ્રવાહી પ્રવાહી ઘન)

- CO2 (કાર્બન ડાયોક્સાઇડ) અગ્નિશામક પસંદ કરો.
- સ્થાન અને પિકઅપ, CO2 અગ્નિશામક. તેના માટે ક્લિક કરો અંતિમ તારીખ.
- સીલ તોડો (ફિગ 2)

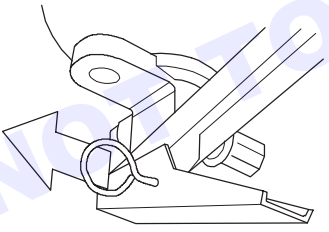
Fig 2



CP20N110322

- હેન્ડલમાંથી સેફ્ટી પિન ખેંચો (પિન પર સ્થિત છે અગ્નિશામકની ટોચ) (ફિગ 3)

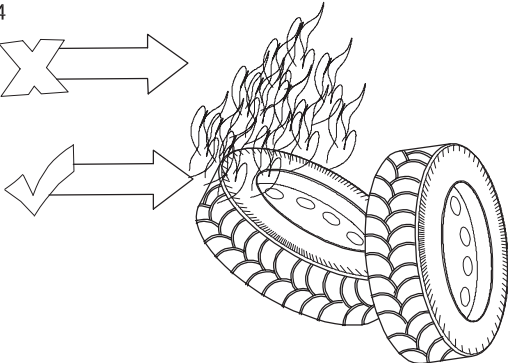
Fig 3



CP20N110323

- અગ્નિશામક નોઝલ અથવા નળીના પાયા પર લક્ષ્ય રાખો આગ (આ બળતણ આગના સ્ત્રોતને દૂર કરશે) (ફિગ 4)

Fig 4

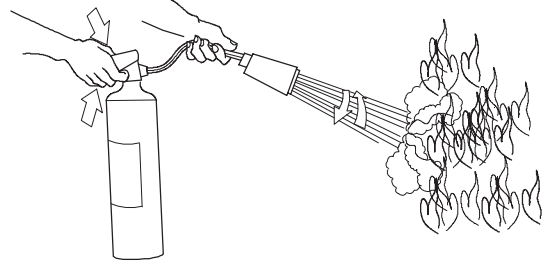


CP20N110324

તમારી જાતને નીચી રાખો

- એજન્ટને ડિસ્ચાર્જ કરવા માટે હેન્ડલ લિવરને ધીમેથી સ્ક્રિવઝ કરો (ફિગ 5)
- બળતણ ઉપર આશરે 15 સેમી ની બાજુથી બાજુ સ્વીપ કરો આગ ઓલવી ન જાય ત્યાં સુધી આગ લગાડો (ફિગ 5)

Fig 5



CP20N110325

અગ્નિશામક ઉપકરણો દૂરથી ઉપયોગ માટે બનાવવામાં આવે છે.

સાવધાન

- આગ ઓલવતી વખતે, આગ ભભૂકી શકે છે
- ગભરાશો નહીં કારણ કે તે તરત જ બંધ થઈ જાય છે.
- જો તમે અગ્નિશામકનો ઉપયોગ કરી લો તે પછી આગ સારી પ્રતિક્રિયા ન આપે તો તમારી જાતને અગ્નિશામક બિંદુથી દૂર ખસેડો.
- જ્યાં આગનો ઝેરી ધુમાડો નીકળતો હોય ત્યાં આગ ઓલવવાનો પ્રયાસ કરશો નહીં તેને વ્યાવસાયિકોમાટે છોડી દો
- યાદ રાખો કે તમારું જીવન મિલકત કરતાં વધુ મહત્વનું છે. તેથી તમારી જાતને અથવા અન્યને જોખમમાં ન મૂકશો. અગ્નિશામકની સરળ કામગીરીને યાદ રાખવા માટે. યાદ રાખો P.A.S.S. આ તમને અગ્નિશામકનો ઉપયોગ કરવામાં મદદ કરશે

P for Pull

A for Aim

S for Squeeze

S for Sweep

કમ્પ્યુટર્સનો પરિચય (Introduction to computers)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- કમ્પ્યુટરને વ્યાખ્યાયિત કરો અને વર્ગીકૃત કરો
- કોમ્પ્યુટરના ફાયદા અને મર્યાદાઓની યાદી બનાવો
- કોમ્પ્યુટરની એપ્લિકેશનોની યાદી બનાવો
- કોમ્પ્યુટરમાં વોલ્ટેજ અને કરંટનું વર્ણન કરો.

કમ્પ્યુટર – વ્યાખ્યાઓ

કોમ્પ્યુટર એ એક ઇલેક્ટ્રોનિક મશીન છે, જે તેની પોતાની મેમરીમાં સંગ્રહિત સૂચનાઓના નિયંત્રણ હેઠળ કાર્ય કરે છે ડેટા (ઇનપુટ) સ્વીકારી શકે છે, તે મુજબ ડેટાની હેરફેર કરી શકે છે ઉલ્લેખિત નિયમો (પ્રક્રિયા), પરિણામો (આઉટપુટ) ઉત્પન્ન કરવા અને ભવિષ્યના ઉપયોગ માટે પરિણામો સંગ્રહિત કરો.

તકનીકી રીતે, કમ્પ્યુટર એ પ્રોગ્રામેબલ મશીન છે. આનો અર્થ એ છે કે તે પ્રોગ્રામ કરેલી સૂચિને એક્ઝિક્યુટ કરી શકે છે સૂચનાઓ અને તે આપવામાં આવેલ નવી સૂચનાઓનો પ્રતિસાદ આપો.

કમ્પ્યુટરનો ઇતિહાસ

ચાર્લ્સ બેબેજનું મશીન

આજના કોમ્પ્યુટરના કાર્યકારી સિદ્ધાંતો અંગ્રેજી ગણિતશાસ્ત્રી ચાર્લ્સ બેબેજ દ્વારા પ્રદાન કરવામાં આવ્યા હતા. 1833 ની આસપાસ “એનાલિટીકલ એન્જિન” તરીકે ઓળખાતા એન્જિન શોધ કરી.” મશીન જે મર્યાદિત તકનીકોનો ઉપયોગ કરીને કાર્યોના કોષ્ટકોની ગણતરી અને પ્રિન્ટ કરી શકે છે.

વિશ્લેષણાત્મક એન્જિનના ચાર ભાગો હતા. એક મિલ, જે હતી વિભાગ કે જેણે ગણતરીઓ કરી હતી, આવશ્યકપણે સી.પી. યુ. સ્ટોર, શું માહિતી રેકૉર્ડ રાખવામાં આવી હતી, અનિવાર્યપણે મેમરી; રીડર, જે પરવાનગી આપશે પંચ કાર્ડનો ઉપયોગ કરીને દાખલ કરવાનો ડેટા, આવશ્યકપણે કીબોર્ડ અને પ્રિન્ટર.

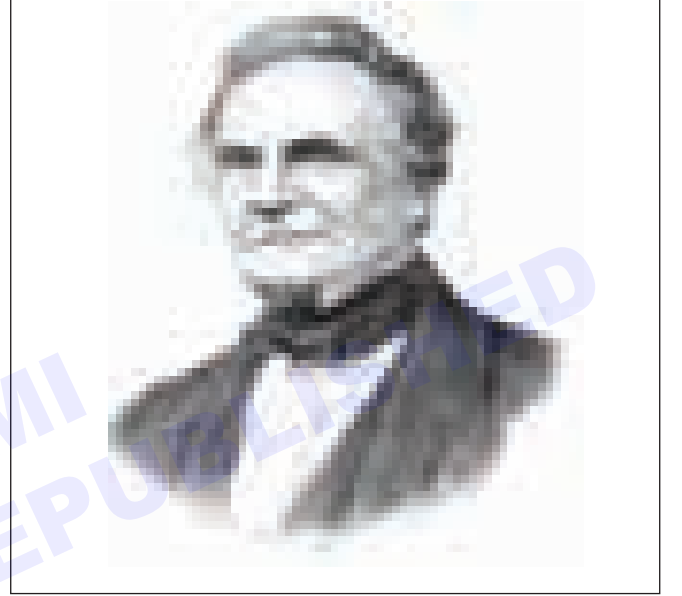
આથી, ચાર્લ્સ બેબેજને ફિગ-1ની જેમ “કોમ્પ્યુટરના પિતા” તરીકે ગણવામાં આવે છે.

કોમ્પ્યુટરની પેઢીઓ એ દ્વારા વર્ગીકૃત થયેલ છે મુખ્ય તકનીકી વિકાસ કે જે મૂળભૂત રીતે કોમ્પ્યુટર ચલાવવાની રીત બદલી, પરિણામે વધુને વધુ નાનું, સસ્તું, વધુ શક્તિશાળી અને વધુ કાર્યક્ષમ અને વિશ્વસનીય ઉપકરણો. વિવિધ પેઢીઓ કોમ્પ્યુટરો નીચે સૂચિબદ્ધ છે:

પ્રથમ પેઢી (1946-1954): 1946 માં ડિજિટલ ઇલેક્ટ્રોનિક વાલ્વ (વેક્યુમ ટ્યુબ) નો ઉપયોગ કરીને કમ્પ્યુટર છે પ્રથમ પેઢીના કમ્પ્યુટર તરીકે ઓળખાય છે. પ્રથમ ‘કોમ્પ્યુટર’ ઇલેક્ટ્રોનિક વાલ્વ એટલે કે વેક્યુમ ટ્યુબનો ઉપયોગ કરવા માટે. ઉચ્ચ શૂન્યાવકાશ ટ્યુબનો ખર્ચ મુખ્ય માટે તેનો ઉપયોગ અટકાવે છે મેમરી તેઓ ફોર્મમાં માહિતી સંગ્રહિત કરે છે ઘનિત તરંગોનો પ્રચાર.

વેક્યુમ ટ્યુબ ખૂબ પાવર વાપરે છે. આ કમ્પ્યુટર્સ કદમાં મોટા હતા અને તેના પર પ્રોગ્રામ લખતા હતા તેઓ મુશ્કેલ હતા. આ પેઢીના કેટલાક કમ્પ્યુટર્સ હતા:

Fig 1



માર્ક I: IBM ઓટોમેટિક સિકવન્સ નિયંત્રિત કેલ્ક્યુલેટર (ASCC), હાર્વર્ડ દ્વારા માર્ક I. કહેવાય છે યુનિવર્સિટી, એક ઇલેક્ટ્રો-મેકેનિકલ કમ્પ્યુટર હતું. ચિહ્ન સફળતાપૂર્વક લાંબી કામગીરી કરનાર હું પ્રથમ મશીન હું અંકગણિત અને તાર્કિક કામગીરીની સેવાઓ. માર્ક I. છે પ્રથમ પેઢીનું કમ્પ્યુટર.

ENIAC: તે 1946 માં બનેલું પ્રથમ ઇલેક્ટ્રોનિક કમ્પ્યુટર હતું યુનિવર્સિટી ઓફ પેન્સિલવેનિયા, યુએસએ ખાતે જોન એકર્ટ દ્વારા અને જહોન મૌરી. તેને ઇલેક્ટ્રોનિક ન્યુમેરિકલ નામ આપવામાં આવ્યું હતું ઇન્ટિગ્રેટર અને કેલ્ક્યુલેટર (ENIAC). ENIAC 30 વર્ષનો હતો- 50 ફૂટ લાંબુ, 30 ટન વજન ધરાવતું, જેમાં 18,000 વેક્યુમ છે ટ્યુબ, 70,000 રેઝિસ્ટર, 10,000 કેપેસિટર્સ અને જરૂરી 150,000 વોટ વીજળી. આજે કોમ્પ્યુટર ઘણા છે ENIAC જેટલો શક્તિશાળી ગણો, હજુ પણ કદ ખૂબ નાનું છે.

EDVAC: તે ઇલેક્ટ્રોનિક ડિસ્ક્રીટ વેરીએબલ માટે વપરાય છે ઓટોમેટિક કોમ્પ્યુટર અને 1950 માં વિકસાવવામાં આવ્યું હતું. આ ની અંદર ડેટા અને સૂચનાઓ સંગ્રહિત કરવાનો ખ્યાલ કમ્પ્યુટર અહીં રજૂ કરવામાં આવ્યું હતું. આ ખૂબ ઝડપી મંજૂરી આપી ઓપરેશન કારણ કે કોમ્પ્યુટર બંનેની ઝડપી એક્સેસ ધરાવે છે ડેટા અને સૂચનાઓ. સંગ્રહ કરવાનો બીજો ફાયદો સૂચના એ હતી કે કમ્પ્યુટર તાર્કિક નિર્ણય લઈ શકે છે આંતરિક રીતે EDVAC એ બાઈનરી સીરીયલ કોમ્પ્યુટર હતું આપોઆપ સરવાળો, બાદબાકી, ગુણાકાર, પ્રોગ્રામ્સ ડિવિઝન અને ઓટોમેટિક ચેકિંગ સાથે અલ્ટ્રાસોનિક સીરીયલ મેમરી.

EDSAC: તે ઇલેક્ટ્રોનિક વિલંબ સંગ્રહ માટે વપરાય છે સ્વચાલિત કમ્પ્યુટર અને M.V દ્વારા વિકસાવવામાં આવ્યું હતું. 1949માં કેમ્બ્રિજ યુનિવર્સિટી ખાતે વિલ્કેસ. ધ EDSAC પ્રથમ સંગ્રહિત-પ્રોગ્રામ કમ્પ્યુટર છે. ઇડીએસએસી ત્રણ મિલીસેકન્ડની શ્રેણીમાં ગણતરીઓ કરી. તે વિના અંકગણિત અને તાર્કિક કામગીરી કરી માનવ હસ્તક્ષેપ. સફળતાની ચાવી આમાં હતી સંગ્રહિત સૂચનાઓ કે જેના પર તે ફક્ત તેની કામગીરી માટે આધાર રાખે છે.

આ મશીને કમ્પ્યુટર યુગની શરૂઆત કરી

UNIVAC-1: તે યુનિવર્સલ ઓટોમેટિક માટે વપરાય છે કમ્પ્યુટર અને તે પ્રથમ વ્યાપારી કમ્પ્યુટર હતું યુનાઇટેડ સ્ટેટ્સ દ્વારા 1951 માં વિકસાવવામાં આવ્યું હતું. મશીન હતું 25 ફૂટ બાય 50 ફૂટ લંબાઈ, જેમાં 5,600 ટ્યુબ, 18,000 છે ક્રિસ્ટલ ડાયોડ અને 300 રિલે. તે સીરીયલ સર્કિટરીનો ઉપયોગ કરે છે, 2.25 MHz બીટ રેટ, અને આંતરિક સ્ટોરેજ ક્ષમતા 1,000 શબ્દો અથવા 12,000 અક્ષરો ધરાવે છે.

UNIVAC નો ઉપયોગ મોટા પ્રમાણમાં ઇનપુટ અને આઉટપુટ સાથે સામાન્ય હેતુની કમ્પ્યુટિંગ માટે કરવામાં આવ્યો હતો. UNIVAC એ સાથે સજ્જ આપનાર પ્રથમ કોમ્પ્યુટર પણ હતું મેગ્નેટિક ટેપ યુનિટ અને બફર મેમરીનો ઉપયોગ કરનાર પ્રથમ કોમ્પ્યુટર હતું.

પ્રથમ પેઢીના કમ્પ્યુટરની મર્યાદાઓ

પ્રથમ પેઢીના કોમ્પ્યુટરની મુખ્ય ખામીઓ નીચે મુજબ છે.

- તેઓ તેમના મુખ્ય ઇલેક્ટ્રોનિક ઘટક તરીકે વાલ્વ અથવા વેક્યુમ ટ્યુબનો ઉપયોગ કરતા હતા.
- તેઓ કદમાં મોટા હતા, પ્રક્રિયામાં ધીમા હતા અને સ્ટોરેજ ક્ષમતા વિનાના હતા.
- તેઓએ ઘણી વીજળીનો વપરાશ કર્યો અને ઘણી ગરમી ઉત્પન્ન કરી.
- તેમની કમ્પ્યુટિંગ ક્ષમતાઓ મર્યાદિત હતી.
- તેઓ એટલા સચોટ અને ભરોસાપાત્ર ન હતા.
- તેઓએ પ્રોગ્રામિંગ માટે મશીન લેવલની ભાષાનો ઉપયોગ કર્યો.
- તેઓ ખૂબ ખર્ચાળ હતા.

સેકન્ડ જનરેશન (1955-1964): સેકન્ડ જનરેશન કોમ્પ્યુટર CPU ઘટકો માટે ટ્રાન્ઝિસ્ટરનો ઉપયોગ કરે છે. અને મુખ્ય મેમરી અને મેગ્નેટિક ડિસ્ક માટે ફેરાઈટ કોરો ગૌણ મેમરી માટે. તેઓ ઉચ્ચ કક્ષાની ભાષાઓનો ઉપયોગ કરતા હતા જેમ કે FORTRAN (1956), ALGOL (1960) અને COBOL (1960 - 1961). ઇનપુટ આઉટપુટ(I/O) પ્રોસેસર I/O કામગીરીને નિયંત્રિત કરવા માટે સામેલ કરવામાં આવ્યું હતું.

ઉત્પાદન ખર્ચ પણ ઘણો ઓછો હતો. આમ કોમ્પ્યુટરનું કદ ઘણું ઓછું થઈ જાય છે.

તે બીજી પેઢીમાં છે કે સેન્ટ્રલનો ખ્યાલ પ્રોસેસિંગ યુનિટ (CPU), મેમરી, પ્રોગ્રામિંગ લેંગ્વેજ અને ઇનપુટ અને આઉટપુટ એકમો વિકસાવવામાં આવ્યા હતા. આમાંના કેટલાક સેકન્ડ જનરેશન કમ્પ્યુટર્સ IBM1620, IBM1401, CDC 3600 છે.

- વેક્યુમ ટ્યુબને બદલે ટ્રાન્ઝિસ્ટરનો ઉપયોગ કરવામાં આવ્યો હતો.
- પ્રોસેસિંગની ઝડપ ફર્સ્ટ જનરેશન કરતાં વધુ ઝડપી છે કમ્પ્યુટર્સ (માઈક્રો સેકન્ડ)

- કદમાં નાનું (51 ચોરસ ફૂટ)

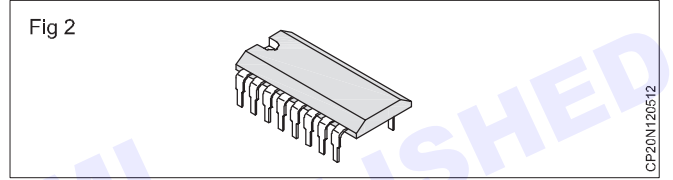
- ઇનપુટ અને આઉટપુટ ઉપકરણો વધુ ઝડપી હતા.

ત્રીજી પેઢી (1964-1977): એ.ના વિકાસ દ્વારા નાની ચિપ જેમાં 300 ટ્રાંઝિસ્ટરની ક્ષમતા હોય છે. આ એકીકૃત સર્કિટ (IC's ચિપ્સ તરીકે પ્રખ્યાત છે.

એક જ ICમાં ઘણા ટ્રાંઝિસ્ટર, રજિસ્ટર અને કેપેસિટર હોય છે સિલિકોનની એક પાતળી સ્લાઈસ પર બનેલ. તેથી તે તદ્દન સ્પષ્ટ છે કે કોમ્પ્યુટરનું કદ વધુ ઘટી ગયું છે. કેટલાક આ સમયગાળા દરમિયાન વિકસિત કમ્પ્યુટર્સમાં IBM 360, ICL-1900, IBM-370, અને VAX-750 હતા. ઉચ્ચ સ્તર BA-SIC જેવી ભાષા (Beginners All purpose સિમ્બોલિક સૂચના કોડ) આ સમયગાળા દરમિયાન વિકસાવવામાં આવી હતી.

આ પેઢીના કોમ્પ્યુટર કદમાં નાના હતા, ઓછી કિંમતના હતા. મોટી મેમરી અને પ્રોસેસિંગ સ્પીડ ખૂબ ઊંચી છે. ખૂબ ટૂંક સમયમાં IC ને LSI (લાઈ સ્કેલ ઇન્ટિગ્રેશન) દ્વારા બદલવામાં આવ્યું, જેમાં લગભગ 100 ઘટકોનો સમાવેશ થતો હતો.

લગભગ 100 ઘટકો ધરાવતા ICને (ફિગ 2) માં LSI કહેવામાં આવે છે. વિશેષતા



- તેઓએ ટ્રાન્ઝિસ્ટરની જગ્યાએ ઇન્ટીગ્રેટેડ સર્કિટ (IC) ચિપ્સનો ઉપયોગ કર્યો.
- સેમી કંડક્ટર મેમરી ઉપકરણોનો ઉપયોગ કરવામાં આવ્યો હતો.
- કદ મોટા પ્રમાણમાં ઘટાડવામાં આવ્યું હતું, પ્રક્રિયાની ઝડપ ઊંચી હતી, અને તેઓ વધુ સચોટ અને વિશ્વસનીય હતા.
- લાઈ સ્કેલ ઇન્ટીગ્રેશન (LSI) અને વેરી લાઈ સ્કેલ એકીકરણ (VLSI) પણ વિકસાવવામાં આવ્યું હતું.
- મિની કોમ્પ્યુટર આ પેઢીમાં રજૂ કરવામાં આવ્યા હતા.
- તેઓએ પ્રોગ્રામિંગ માટે ઉચ્ચ સ્તરીય ભાષાનો ઉપયોગ કર્યો.

ફોર્થ જનરેશન (1978 - વર્તમાન): એક IC ધરાવતું લગભગ 100 ઘટકોને LSI (લાઈ સ્કેલ ઇન્ટિગ્રેશન) કહેવામાં આવે છે અને એક, જેમાં આવા 1000 થી વધુ ઘટકોને VLSI (વેરી લાઈ સ્કેલ ઇન્ટીગ્રેશન) તરીકે ઓળખવામાં આવે છે.

તે સિંગલ પર બનેલા મોટા પાયે ઇન્ટીગ્રેટેડ સર્કિટ (LSIC) નો ઉપયોગ કરે છે સિલિકોન ચિપ જેને માઈક્રોપ્રોસેસર્સ કહેવાય છે. કારણે માઈક્રોપ્રોસેસરનો વિકાસ શક્ય છે કમ્પ્યુટરનું સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (સીપીયુ) સિંગલચીપ પર. આ કમ્પ્યુટર્સને માઈક્રો કોમ્પ્યુટર કહેવામાં આવે છે.

બાદમાં ખૂબ મોટા પાયે ઇન્ટીગ્રેટેડ સર્કિટ (VLSIC) બદલાઈ ગયા LSICs. આમ કોમ્પ્યુટર જે ખૂબ જ કબજે કરી રહ્યું હતું પહેલાના દિવસોમાં મોટો ઓરડો હવે ટેબલ પર મૂકી શકાય છે. વ્યક્તિગત કમ્પ્યુટર (PC) જે તમે તમારી શાળામાં જુઓ છો ફોર્થ જનરેશન કોમ્પ્યુટર મેઈન મેમરી છે જેનો ઝડપી ઉપયોગ થાય છે સેમિકન્ડક્ટર ચિપ્સ 4 Mb સુધી તેની સાઈઝ. કેટલાક ચોથી પેઢીના કમ્પ્યુટર્સ IBM PC, Apple Macintosh, વગેરે છે.

હાર્ડ ડિસ્કનો ઉપયોગ ગૌણ મેમરી તરીકે થતો હતો. કીબોર્ડ, ડોટ મેટ્રિક્સ

પ્રિન્ટર વગેરે વિકસાવવામાં આવ્યા હતા. ઓપરેટિંગ સિસ્ટમ (OS)-જેમ કે MS-DOS, UNIX, Appleના Macintosh ઉપલબ્ધ હતા. ઓબ્જેક્ટ ઓરિએન્ટેડ લેંગ્વેજ, C++ વગેરેનો વિકાસ થયો.

વિશેષતા

- તેઓએ માઈક્રોપ્રોસેસર (VLSI) નો ઉપયોગ તેમના મુખ્ય સ્વિચિંગ તત્વ તરીકે કર્યો.
- તેમને માઈક્રો કમ્પ્યુટર અથવા પર્સનલ કોમ્પ્યુટર તરીકે પણ ઓળખવામાં આવે છે.
- તેમનું કદ ડેસ્કટોપથી લેપટોપ અથવા પામટોપ સુધી બદલાય છે.
- તેમની પાસે પ્રક્રિયા કરવાની ખૂબ જ ઊંચી ઝડપ છે; તેઓ છે 100% સચોટ, ભરોસાપાત્ર, મહેનતું અને બહુમુખી.
- તેમની પાસે ખૂબ મોટી સંગ્રહ ક્ષમતા છે.

પાંચમી પેઢી (વર્તમાન અને ભવિષ્ય): 5મી જનરેશન કમ્પ્યુટર્સ યુએલએસઆઈ (અલ્ટ્રા-લાઈ સ્કેલ ઇન્ટિગ્રેશન) ચિપનો ઉપયોગ કરે છે. VLSI ચિપ્સમાં એક જ ICમાં લાખો ટ્રાન્ઝિસ્ટર મૂકવામાં આવે છે.

આ સમયગાળા દરમિયાન 64 બીટ માઈક્રોપ્રોસેસર વિકસાવવામાં આવ્યા છે. મેમરી ચિપ્સ અને 1 જીબી સુધીની ફ્લેશ મેમરી, 600 જીબી સુધીની હાર્ડ ડિસ્ક અને 50 જીબી સુધીની ઓપ્ટિકલ ડિસ્ક વિકસાવવામાં આવી છે (ફિગ. 3)

Fig 3



આર્ટિફિશિયલ ઇન્ટેલિજન્સ પર આધારિત પાંચમી પેઢીના કમ્પ્યુટિંગ ઉપકરણો હજુ પણ વિકાસમાં છે, જોકે તેમાં કેટલાક એપ્લીકેશનો, જેમ કે અવાજ ઓળખ, જે આજે ઉપયોગમાં લેવાય છે.

આર્ટિફિશિયલ ઇન્ટેલિજન્સ એ કોમ્પ્યુટર વિજ્ઞાનની શાખા છે કોમ્પ્યુટરને માણસોની જેમ વર્તે છે. આ શબ્દ 1956 માં જહોન મેકકાર્થી દ્વારા બનાવવામાં આવ્યો હતો મેશાચ્યુરેટ તકનીકી સંસ્થાન. કૃત્રિમ બુદ્ધિમાં શામેલ છે:

- રમતો રમવી:** રમવા માટે પ્રોગ્રામિંગ કમ્પ્યુટર્સ ચેસ અને ચેકર્સ જેવી રમતો
- નિષ્ણાત સિસ્ટમ્સ:** પ્રોગ્રામિંગ કમ્પ્યુટર્સ બનાવવા માટે વાસ્તવિક જીવનની પરિસ્થિતિઓમાં નિર્ણયો (ઉદાહરણ તરીકે, કેટલાક નિષ્ણાત સિસ્ટમો ડૉક્ટરોને રોગોના આધારે નિદાન કરવામાં મદદ કરે છે લક્ષણો પર)
- નેચરલ લેંગ્વેજ:** પ્રોગ્રામિંગ કોમ્પ્યુટર માટે કુદરતી માનવ ભાષાઓ સમજો
- ન્યુરલ નેટવર્ક્સ:** સિસ્ટમ્સ કે જે બુદ્ધિનું અનુકરણ કરે છે ભૌતિક પ્રકારોનું પુનઃઉત્પાદન કરવાનો પ્રયાસ કરીને જોડાણો જે પ્રાણીઓના મગજમાં થાય છે
- રોબોટિક્સ:** જોવા અને સાંભળવા માટે પ્રોગ્રામિંગ કમ્પ્યુટર્સ અને અન્ય સંવેદનાત્મક ઉત્તેજના પર પ્રતિક્રિયા આપે છે

જનરેશન	ઇલેક્ટ્રોનિક ઘટક	ફાયદા	ગેરફાયદા
પ્રથમ	વેક્યુમ ટ્યુબ અને કોમ્પ્યુટેશનલ વર્ક	ગણતરીમાં મદદ કરી	1 મોટું કદ 2 ખૂબ ખર્ચાળ 3 ધીમી ગતિ 4 ઓછી ચોકસાઈ 5 ઓછો સંગ્રહ 6 ઉચ્ચ પાવર જરૂરિયાતો 7 ઉચ્ચ ગરમીનું ઉત્પાદન 8 ઉચ્ચ નિષ્ફળતા દર 9 વપરાયેલ મશીન ભાષા 10 કોઈ ઓપરેટિંગ સિસ્ટમ નથી
દ્વિતીય	ટ્રાન્ઝિસ્ટર	1 નાનું કદ 2 ઓછી કિંમત 3 સારી ઝડપ 4 ઓછો વીજ વપરાશ અને ઓછી ગરમીનું ઉત્પાદન 5 સારી સંગ્રહ ક્ષમતા 6 વધુ સારી ચોકસાઈ અને વધુ વિશ્વસનીયતા	1 એર કન્ડીશનીંગની જરૂર છે 2 સતત જાળવણી 3 કોઈ ઓપરેટિંગ સિસ્ટમ નથી 4 લેયર સ્ટેજ કમ્પ્યુટર્સ એસેમ્બલી લેંગ્વેજનો ઉપયોગ કરે છે

તૃતીય	સંકલિત સર્કિટ (IC) નાના અને મધ્યમ પાયે	1 I અને II ની તુલનામાં તમામ પાસાઓમાં વધુ સારું 2 વપરાયેલ ઓપરેટિંગ સિસ્ટમ અને ઉચ્ચ સ્તરીય ભાષા	1 ઉત્પાદકો સાથે પ્રારંભિક સમસ્યા. 2 આંતરિક કાર્યમાં કોઈ આંતરદૃષ્ટિ પ્રાપ્ત થઈ નથી
ચોથું	VLSI અથવા માઈક્રોપ્રોસેસર	1 ઓછી કિંમત 2 ઉત્તમ ઝડપ અને વિશ્વસનીયતા 3 માણસની નજીકના કમ્પ્યુટર	1 મેઈનફ્રેમ કોમ્પ્યુટર કરતા ઓછા શક્તિશાળી
પાંચમું (જ્ઞાન, માહિતી, પ્રક્રિયા, સિસ્ટમ્સ)	ULSI અથવા બાયોચિપ્સ	1 ખૂબ સસ્તામાં 2 સુપર સ્પીડ. 3 ખૂબ ઊંચી સંગ્રહ ક્ષમતા 4 અત્યંત આધુનિક ઓએસ 5 પ્રક્રિયા બુદ્ધિ અને નિર્ણય લેવાની ક્ષમતા	1 નવી નીચા સ્તરની ભાષાની જરૂર છે

કમ્પ્યુટરનું વર્ગીકરણ

કમ્પ્યુટરને નીચેના માપદંડો અનુસાર વર્ગીકૃત કરવામાં આવે છે:

- ઓપરેશનનો સિદ્ધાંત
- કોમ્પ્યુટિંગ પાવર, મેમરી ક્ષમતા અને કિંમત
- ટેકનોલોજીકલ વિકાસ
- કામગીરીનો સિદ્ધાંત
- એનાલોગ કોમ્પ્યુટર
- ડિજિટલ કોમ્પ્યુટર
- હાઈબ્રિડ કોમ્પ્યુટર

એનાલોગ કોમ્પ્યુટર

તે એક કમ્પ્યુટર છે જે સતત બદલાતા માપન કરે છે ભૌતિક જથ્થો જેમ કે વર્તમાન, તાપમાન, દબાણ વગેરે અને તેમને જથ્થામાં રૂપાંતરિત કરે છે જે ગણતરી માટે ડેટા તરીકે ઉપયોગ કરી શકાય છે. આ કમ્પ્યુટર્સ તરીકે

તેઓ જે આપશે તે સતત બદલાતી જથ્થા સાથે વ્યવહાર કરો માત્ર અંદાજિત પરિણામો. તેનું આઉટપુટ સામાન્ય રીતે પ્રદર્શિત થાય છે મીટર અથવા સ્કેલ પર. એનાલોગ કમ્પ્યુટરમાં ઓછી મેમરી હોય છે અને ઓછા કાર્યો. આનો ઉપયોગ ઈજનેરી અને વૈજ્ઞાનિક કાર્યક્રમો માટે થાય છે.

- થર્મોમીટર
- સ્પીડોમીટર
- એનાલોગ ઘડિયાળ

ડિજિટલ કમ્પ્યુટર

ડિજિટલ કમ્પ્યુટર ડિજિટલ ડેટા સાથે કામ કરે છે. ડિજિટલ કમ્પ્યુટર બાઈનરી નંબર સિસ્ટમનો ઉપયોગ કરે છે. બાઈનરી નંબર સિસ્ટમ માત્ર બે અંકો '0' અને '1' ધરાવે છે. ડિજિટલ કમ્પ્યુટર ડિજિટલ સિગ્નલમાં ડેટા રજૂ કરે છે. A '0' OFF દર્શાવે છે અને '1' ON દર્શાવે છે. ડિજિટલ કમ્પ્યુટર કરે છે અંકગણિત અને તાર્કિક કામગીરી ડેટા. તે ડિજિટલ સ્વરૂપમાં આઉટપુટ આપે છે.

ડિજિટલ કોમ્પ્યુટર ખુબ જ ઝડપી છે. આ કમ્પ્યુટર્સ કરી શકે છે.

એનાલોગ કોમ્પ્યુટર	ડિજિટલ કમ્પ્યુટર
1 એનાલોગ કમ્પ્યુટર્સ સતત મૂલ્યો પર કામ કરે છે.	1 ડિજિટલ કોમ્પ્યુટર અલગ મૂલ્યો પર કામ કરે છે.
2 એનાલોગ કોમ્પ્યુટરમાં ઓછી મેમરી હોય છે.	2 ડિજિટલ કોમ્પ્યુટરની મેમરી ઘણી મોટી હોય છે
3 એનાલોગ કોમ્પ્યુટર ધીમી ગતિ ધરાવે છે.	3 ડિજિટલ કોમ્પ્યુટર ઝડપી ગતિ ધરાવે છે.
4 એનાલોગ કમ્પ્યુટર્સ ઓછા વિશ્વસનીય છે.	4 ડિજિટલ કમ્પ્યુટર્સ વધુ વિશ્વસનીય છે.
5 એન્જીનીયરીંગમાં વપરાતા એનાલોગ	5 જીવનના તમામ ક્ષેત્રોમાં ડિજિટલ કમ્પ્યુટરનો ઉપયોગ થાય છે. વિજ્ઞાન અને તબીબી ક્ષેત્રો.
6 એનાલોગ કમ્પ્યુટર્સનો ઉપયોગ ગતિ અને તાપમાન જેવા એનાલોગ જથ્થાઓની ગણતરી/માપવા માટે થાય છે.	6 ડિજિટલ કમ્પ્યુટર્સનો ઉપયોગ ગાણિતિક અને તાર્કિક કામગીરીની ગણતરી કરવા માટે થાય છે. તે સરવાળા, બાદબાકી, ભાગાકાર, ગુણાકાર અને અન્ય ગાણિતિક અને આંકડાકીય ક્રિયાઓ ઉકેલી શકે છે.
7 એનાલોગ કમ્પ્યુટર ઓછા સચોટ પરિણામો આપે છે.	7 ડિજિટલ કમ્પ્યુટર્સ 100% સચોટ પરિણામો પ્રદાન કરે છે.
8 સામાન્ય રીતે એનાલોગ કોમ્પ્યુટર ચોક્કસ હેતુ હોય છે	8 ડિજિટલ કોમ્પ્યુટર સામાન્ય હેતુ છે.
9 સામાન્ય રીતે એનાલોગ કોમ્પ્યુટર ચોક્કસ હેતુ હોય છે.	9 ડિજિટલ કોમ્પ્યુટર સામાન્ય હેતુ છે.
10 એનાલોગ કમ્પ્યુટરનાં ઉદાહરણો છે: થર્મોમીટર, એનાલોગ ઘડિયાળ, સ્પીડોમીટર વગેરે.	10 ડિજિટલ કોમ્પ્યુટરના ઉદાહરણો છે: પર્સનલ કોમ્પ્યુટર, લેપટોપ, સ્માર્ટફોન વગેરે.

સ્પેશિયલ પર્યાજ કોમ્પ્યુટર્સ

તે ચોક્કસ પ્રકારની સમસ્યાને ઉકેલવા માટે રચાયેલ કમ્પ્યુટર છે. જહાજો અને એરક્રાફ્ટ વગેરેમાં વપરાતા કમ્પ્યુટર.

સામાન્ય હેતુ કમ્પ્યુટર્સ

તે એક કોમ્પ્યુટર છે જે વિવિધ પ્રકારની સમસ્યાઓ ઉકેલવા માટે રચાયેલ છે, એક સામાન્ય હેતુ કોમ્પ્યુટર સ્ટોર કરી શકે છે વિવિધ કાર્યક્રમો અને તેમની પ્રક્રિયા.

એનાલોગ અને ડિજિટલ કમ્પ્યુટર્સ વચ્ચેનો તફાવત કોષ્ટક 2 માં સૂચિબદ્ધ છે.

હાઇબ્રિડ કોમ્પ્યુટર (ફિગ. 4)

હાઇબ્રિડ કમ્પ્યુટર એ એનાલોગ અને ડિજિટલ કમ્પ્યુટર બંનેનું સંયોજન છે. હાઇબ્રિડ કોમ્પ્યુટર એનાલોગ અને ડીજીટલ ડેટા બંનેને હેન્ડલ કરી શકે છે. એક હાઇબ્રિડ કોમ્પ્યુટર આને જોડે છે એનાલોગ અને ડિજિટલ બંનેની શ્રેષ્ઠ લાક્ષણિકતાઓ કમ્પ્યુટર તે એનાલોગ અને ડિજિટલ સ્વરૂપ બંનેમાં ડેટા સ્વીકારી શકે છે.

Fig 4



પરિભાષા

હાઇબ્રિડ કોમ્પ્યુટર ઉપકરણોનો ઉપયોગ હોસ્પિટલોમાં કરવામાં આવે છે જે થઈ શકે છે દર્દીના હૃદયના કાર્ય, તાપમાન અને લોહીની ગણતરી કરો દબાણ વગેરે. આ ગણતરીમાં રૂપાંતરિત થઈ શકે છે સ્ટોર પરિણામો. તેમની પાસે મોટી મેમરી છે (જે ડેટા સ્ટોરિંગ છે ક્ષમતા). આજે ઓફિસો અને ઘરોમાં વપરાતા મોટાભાગના કોમ્પ્યુટર ડીજીટલ કોમ્પ્યુટર છે.

ડિજીટલ કોમ્પ્યુટરને આગળ વિભાજિત કરવામાં આવે છે નીચેના બે જૂથો: કમ્પ્યુટર્સ

- સામાન્ય હેતુના કમ્પ્યુટર્સ નંબરો અને ડિજિટલ સ્વરૂપમાં દર્શાવેલ છે. ઉદાહરણ તરીકે, ઘ વાઈટલ સાઈન્સ મોનિટરિંગ યુનિટને ટૂંકમાં (VSM) પણ કહેવાય છે. તે છે બ્લડ પ્રેશર મોનિટર, ઇસીજી મોનિટર, શ્વસન મોનિટર, અને એનેસ્થેસિયાની દેખરેખ માટે પણ વપરાય છે.
- હાઇબ્રિડ કોમ્પ્યુટરનો ઉપયોગ સ્પેસશીપ અને મિસાઈલ સિસ્ટમમાં પણ થાય છે.
- હાઇબ્રિડ કોમ્પ્યુટર મશીનોનો ઉપયોગ સામાન્ય રીતે વૈજ્ઞાનિક કાર્યક્રમોમાં થાય છે
- હાઇબ્રિડ કોમ્પ્યુટરનો ઉપયોગ ઔદ્યોગિક પ્રક્રિયાઓને નિયંત્રિત કરવા માટે થાય છે.

કોમ્પ્યુટરને કોમ્પ્યુટીંગ પાવર, મેમરી ક્ષમતા અને ખર્ચના આધારે વર્ગીકૃત કરવામાં આવે છે.

- માઈક્રોકોમ્પ્યુટર અથવા પર્સનલ કોમ્પ્યુટર.
- મીની કોમ્પ્યુટર.
- મેઈનફ્રેમ કોમ્પ્યુટર.
- સુપર કોમ્પ્યુટર.

માઈક્રો કમ્પ્યુટર્સ

માઈક્રો કોમ્પ્યુટરને પર્સનલ કોમ્પ્યુટર પણ કહેવાય છે. તે 1970 માં રજૂ કરવામાં આવ્યું હતું. વ્યક્તિગત કમ્પ્યુટરનાં ઉદાહરણો છે.

પીસી અને એપલ મેકિન્ટોશ. વ્યક્તિગત મુખ્ય પ્રકારો કમ્પ્યુટર્સ ડેસ્કટોપ કમ્પ્યુટર અને પોર્ટેબલ કમ્પ્યુટર છે.

ડેસ્કટોપ કોમ્પ્યુટર

આ કોમ્પ્યુટરો એટેબલ અથવા ડેસ્કટોપ પર સરળતાથી ફિટ થઈ શકે છે, તેથી તેનું નામ. આ કમ્પ્યુટર્સ બે મોડલમાં આવે છે અથવા casings. ડેસ્કટોપ મોડેલમાં, સિસ્ટમ યુનિટ ડેસ્કટોપ અથવા ટેબલ પર મૂકવામાં આવે છે. સિસ્ટમ પર મોનિટર મૂકવામાં આવે છે એકમ ટાવર મોડેલમાં, મોનિટર અને સિસ્ટમ એકમ બંને છે (ફિગ 5) ની જેમ ટેબલ પર મૂકવામાં આવે છે.

Fig 5



TOWER MODEL COMPUTER

CP20N120515

પોર્ટેબલ કમ્પ્યુટર

પોર્ટેબલ એક વ્યક્તિગત કમ્પ્યુટર છે જે સરળતાથી એક જગ્યાએથી બીજી જગ્યાએ લઈ જઈ શકાય છે. નોટબુક કોમ્પ્યુટર અને હેન્ડહેલ્ડ કોમ્પ્યુટર (સ્માર્ટ ફોન) એ પોર્ટેબલ કોમ્પ્યુટરના ઉદાહરણો છે. નોટબુક કોમ્પ્યુટર પણ કહેવાય છે લેપટોપ કમ્પ્યુટર. લેપટોપ કોમ્પ્યુટર ખૂબ નાના છે કદ અને ખોળામાં સરળતાથી મૂકી શકાય છે.

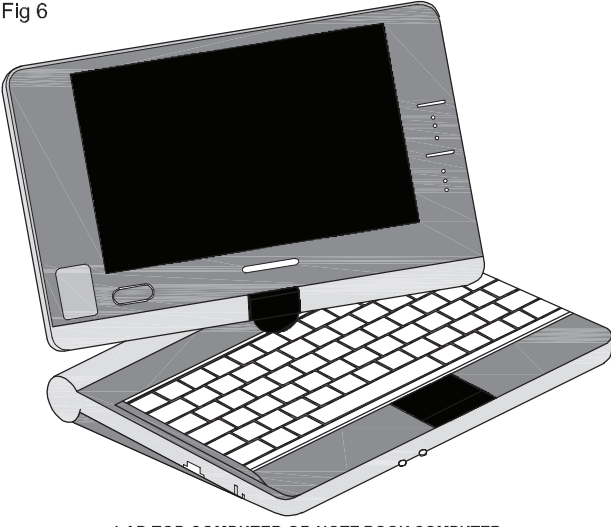
લેપટોપ કમ્પ્યુટર અથવા નોટબુક કમ્પ્યુટર

લેપટોપ કોમ્પ્યુટર કે નોટબુક કોમ્પ્યુટર જેવું હશે ફિગ.6 a અને ફિગ 6b માં બતાવેલ છે. તેને ટાવર મોડલ કોમ્પ્યુટર તરીકે પણ ઓળખવામાં આવે છે.

પામટોપ કોમ્પ્યુટર/ડિજિટલ ડાયરી/નોટબુક/PDAs:

હેન્ડહેલ્ડ કમ્પ્યુટર (જેમ કે સ્માર્ટ ફોન) પણ પોર્ટેબલ છે. હેન્ડ હેલ્ડ કોમ્પ્યુટરને પામટોપ કોમ્પ્યુટર તરીકે ઓળખવામાં આવે છે.

Fig 6



LAP TOP COMPUTER OR NOTE BOOK COMPUTER

CP20N120516

Fig 7



પામટોપમાં કોઈ કીબોર્ડ નથી પરંતુ સ્ક્રીન ઈનપુટ અને આઉટપુટ બંને ઉપકરણ તરીકે સેવા આપે છે. તે વપરાશકર્તાના હાથમાં સરળતાથી ફિટ થઈ જાય છે.

માઈક્રો કોમ્પ્યુટરનો ઉપયોગ

પીસી એ સૌથી સામાન્ય પ્રકારનું કમ્પ્યુટર છે જેનો ઉપયોગ થાય છે ઓફિસ હવે તે ઘણા ઘરોમાં વ્યાપકપણે ઉપયોગમાં લેવાય છે. આ બિઝનેસ અને એન્જિનિયરિંગ એપ્લિકેશન માટે પણ વપરાય છે.

મીની કોમ્પ્યુટર

મિની કોમ્પ્યુટર 1960 ના દાયકામાં રજૂ કરવામાં આવ્યા હતા. મિનીકોમ્પ્યુટર પર્સનલ કોમ્પ્યુટર કરતા મોટું અને વધુ શક્તિશાળી કોમ્પ્યુટર છે. તે સેકન્ડ હીઠ 50 લાખ સૂચનાઓ એક્ઝિક્યુટ કરી શકે છે. તે સામાન્ય રીતે બે અથવા વધુ પ્રોસેસર્સ ધરાવે છે.

મીની કોમ્પ્યુટરનો ઉપયોગ

મિની કોમ્પ્યુટરનો ઉપયોગ મોટાભાગે નાના અને મધ્યમ કદના લોકો દ્વારા કરવામાં આવે છે કંપનીઓ માહિતીનો કેન્દ્રિય સંગ્રહ પ્રદાન કરે છે.

મેઈનફ્રેમ કોમ્પ્યુટર

મેઈનફ્રેમ કોમ્પ્યુટર 1975 માં રજૂ કરવામાં આવ્યા હતા. એ મેઈનફ્રેમ કોમ્પ્યુટર એ કદમાં ખૂબ મોટું કમ્પ્યુટર છે. તે છે પ્રોસેસર્સ તે માટે બહુવિધ

કાર્યો કરવા માટે રચાયેલ છે એક જ સમયે બહુવિધ વપરાશકર્તાઓ. મેઈનફ્રેમ કમ્પ્યુટર્સ એક જ સમયે 50,000 વપરાશકર્તાઓ સુધી સેવા આપી શકે છે.

વપરાશકર્તાઓ મેઈનફ્રેમ કમ્પ્યુટર દ્વારા એક્સેસ કરે છે ટર્મિનલ અથવા વ્યક્તિગત કમ્પ્યુટર. એક લાક્ષણિક મેઈનફ્રેમ કમ્પ્યુટર પ્રતિ સેકન્ડ 16 મિલિયન સૂચનાઓ એક્ઝિક્યુટ કરી શકે છે. કેટલાક મુખ્ય કમ્પ્યુટર મોડેલો છે.

- NEC 610
- DEC 10

મેઈનફ્રેમ કોમ્પ્યુટરનો ઉપયોગ

મેઈનફ્રેમ કોમ્પ્યુટરનો ઉપયોગ મુખ્યત્વે કોર્પોરેટ દ્વારા કરવામાં આવે છે અને જટિલ એપ્લિકેશનો માટે સરકારી સંસ્થાઓ, બલ્ક ડેટા પ્રોસેસિંગ જેમ કે ઇન્દ્રિયો, ઉદ્યોગ અને ઉપભોક્તા આંકડા, અને સોદા પ્રક્રિયા.

સુપર કોમ્પ્યુટર

સુપર કોમ્પ્યુટર 1980 માં રજૂ કરવામાં આવ્યા હતા. સુપર કોમ્પ્યુટર એ સૌથી ઝડપી કોમ્પ્યુટર છે. સુપર કોમ્પ્યુટર કદમાં સૌથી મોટું અને અન્ય કોમ્પ્યુટર કરતાં કિંમતમાં સૌથી મોંઘું છે.

તે સૌથી અત્યાધુનિક, જટિલ અને અદ્યતન છે કમ્પ્યુટર તે ખૂબ મોટી સંગ્રહ ક્ષમતા ધરાવે છે. તે કરી શકે છે એક સેકન્ડમાં ટ્રિલિયન સૂચનાઓ પર પ્રક્રિયા કરો. સુપર કમ્પ્યુટર એ સૌથી ઝડપી અને સૌથી શક્તિશાળી કમ્પ્યુટર છે એક સમય. સુપર કોમ્પ્યુટર ખૂબ ખર્ચાળ છે. સુપર કોમ્પ્યુટરનો ઉપયોગ ઉચ્ચ ગણતરી-સઘન કાર્યો માટે થાય છે. સુપર કોમ્પ્યુટરનો ઉપયોગ વિશેષતા માટે પણ થાય છે એપ્લિકેશનો કે જેને ગાણિતિક ગણતરીઓની પુષ્કળ માત્રાની જરૂર હોય છે.

સુપર કોમ્પ્યુટરની એપ્લિકેશન

- હવામાનની આગાહી,
- એનિમેટેડ ગ્રાફિક્સ જેમ કે હોલીવુડ મૂવીઝમાં,
- પ્રવાહી ગતિશીલ ગણતરીઓ
- પરમાણુ ઊર્જા સંશોધન
- અવકાશ વિજ્ઞાન
- હથિયાર અને મિસાઈલ ડિઝાઈન
- પેટ્રોલિયમ સંશોધન, અને વગેરે.

આજે, સુપર કોમ્પ્યુટર પરંપરાગત દ્વારા બનાવવામાં આવે છે કે, આઈબીએમ અને હેવલેટ પેકાર્ડ જેવી કંપનીઓ. ઓક્ટોબર 2010 થી, Tianhe-1A સુપર કોમ્પ્યુટર છે વિશ્વમાં સૌથી ઝડપી છે; તે ચીનમાં સ્થિત છે.

સુપર કોમ્પ્યુટર અને એ વચ્ચેનો મુખ્ય તફાવત મેઈનફ્રેમ એ છે કે સુપર કોમ્પ્યુટર તેની તમામ શક્તિને ચેનલ કરે છે શક્ય તેટલી ઝડપથી એક જ પ્રોગ્રામનો અમલ કરવા માટે, જ્યારે મેઈનફ્રેમ તેની શક્તિનો ઉપયોગ ઘણાને ચલાવવા માટે કરે છે એક સાથે કાર્યક્રમો. આધુનિક સુપર કોમ્પ્યુટર હજારો માઈક્રોપ્રોસેસરનો સમાવેશ થાય છે. સુપર કમ્પ્યુટર ઓનલાઈન પ્રોસેસિંગ માટે સેટેલાઈટ જેવી હાર્ડ-સ્પીડ સુવિધાઓનો ઉપયોગ કરે છે.

સુપર કમ્પ્યુટર મોડલ્સનો સરવાળો CRAY-XP, ETA- છે 10, પરમ અને ડીપ બ્લુ.

કમ્પ્યુટરના ફાયદા

- કોમ્પ્યુટરની પ્રક્રિયાની ઝડપ ખૂબ ઊંચી હોય છે વિશ્વસનીયતા
- માહિતીનો મોટો જથ્થો સંગ્રહિત કરી શકાય છે મેમરી કોઈપણ ચોક્કસ તરત ડેટા/પ્રોગ્રામ પુનઃપ્રાપ્ત કરી શકાય છે.
- જટિલ સમસ્યાનો ઉકેલ ખૂબ જ વધુ ઝડપે શક્ય છે.
- મોટા જથ્થાના ડેટાની પ્રક્રિયા કરવાથી કારકુનીની ઘણી બચત થાય છે

કાર્ય જે પ્રક્રિયા ખર્ચ ઘટાડે છે.

- કોમ્પ્યુટર અસરકારક રીતે કામગીરી કરે છે પર્યાવરણ જ્યાં મનુષ્યની હાજરી નથી શક્ય જેમ કે ભૂમિ, ઝેરી વાતાવરણ, શૂન્યાવકાશ, માનવરહિત ઉપગ્રહ, વગેરે.

કમ્પ્યુટર્સની મર્યાદા

- ઉચ્ચ પ્રારંભિક ખર્ચ.
- ઇનપુટ માહિતી ફોર્મમાં તૈયાર કરવાની રહેશે વિદ્યાર્થીને પ્રોગ્રામ કહેવામાં આવે છે જેમાં નોંધપાત્ર પ્રમાણમાં જ્ઞાનની જરૂર હોય છે.
- કોમ્પ્યુટરનો ઉપયોગ ત્યારે જ આર્થિક રહેશે જ્યારે ત્યાં હશે મોટી માત્રામાં ડેટા માટે કારકુની ડેટા પ્રોસેસિંગ છે અને પ્રકૃતિમાં પુનરાવર્તિત છે
- તે માત્ર એક મશીન છે જે પોતાની જાતે ભૂલો સુધારી શકતું નથી.

કોમ્પ્યુટરના કાર્યો

બધા કોમ્પ્યુટર નીચેના મૂળભૂત એકમોથી બનેલા છે ફિગ (7) માં બતાવેલ છે. તેઓ નીચે મુજબ છે:-

- 1 ઇનપુટ યુનિટ
- 2 સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (CPU)

a અંકગણિત તર્ક એકમ (ALU)

b કંટ્રોલ યુનિટ (CU)

3 મેમરી

4 આઉટપુટ યુનિટ

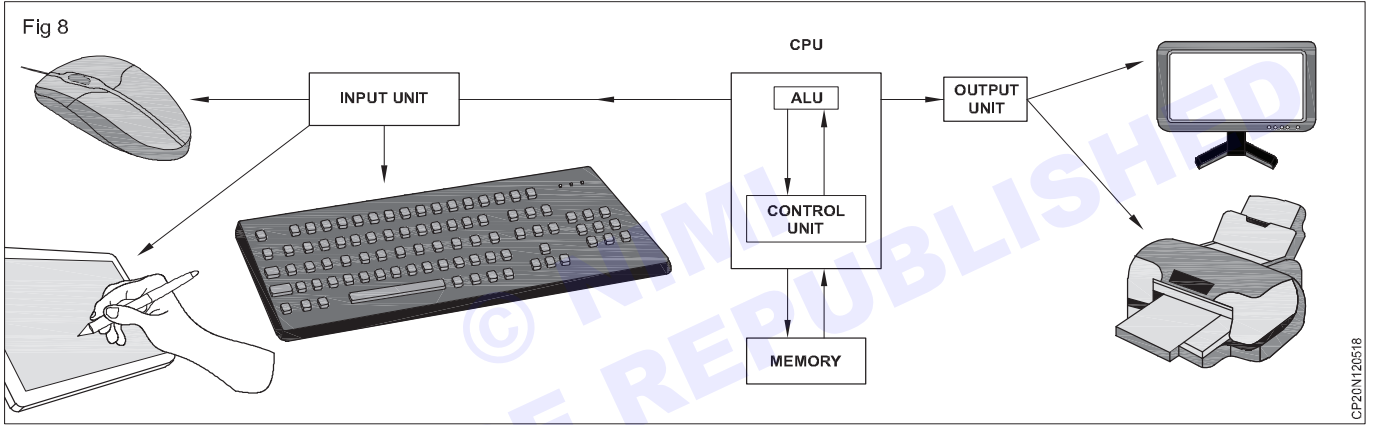
ઇનપુટ યુનિટ

કોઈપણ સમસ્યાને ઉકેલવા માટે કમ્પ્યુટરને ડેટા અને સૂચના પ્રાપ્ત કરવાની જરૂર છે. તેથી આપણે કોમ્પ્યુટરમાં ડેટા અને સૂચનાઓ દાખલ કરવાની જરૂર છે. ઇનપુટ યુનિટ એક અથવા વધુ ઇનપુટ ઉપકરણોનો સમાવેશ થાય છે. કીબોર્ડ એ સૌથી વધુ ઉપયોગમાં લેવાતા ઇનપુટ ઉપકરણમાંથી એક છે. કેટલાક ઇનપુટ ઉપકરણો કોષ્ટક 1 માં સૂચિબદ્ધ છે.

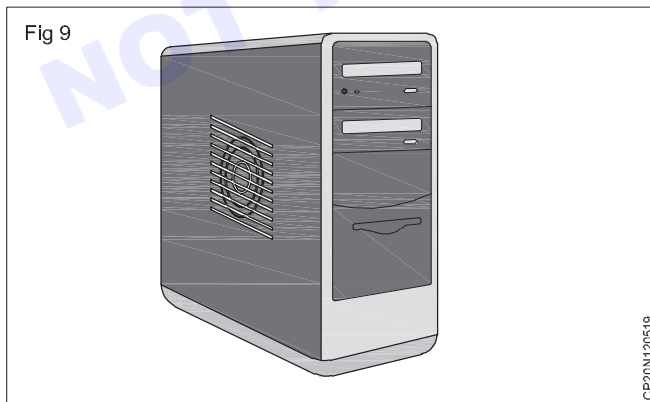
ઇનપુટ ઉપકરણો નીચેના કાર્યો કરે છે.

- બહારની દુનિયામાંથી ડેટા અને સૂચનાઓ સ્વીકારો.
- તેને કોમ્પ્યુટર સમજી શકે તેવા ફોર્મમાં કન્વર્ટ કરો.
- વધુ પ્રક્રિયા માટે રૂપાંતરિત ડેટાને કોમ્પ્યુટર સિસ્ટમમાં સપ્લાય કરો.

સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (CPU) (ફિગ. 8)



સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (CPU) એ ફિગ-8ની જેમ કમ્પ્યુટરનું ઇલેક્ટ્રોનિક મગજ છે. વ્યક્તિગત કમ્પ્યુટરમાં સી.પી.યુ સામાન્ય રીતે એક ચિપ હોય છે.



તે વ્યવસ્થિત કરે છે અને તે સૂચનાઓનું વહન કરે છે જે વપરાશકર્તા અથવા માંથી ક્યાં તો તરફથી આવે છે સોફ્ટવેર પ્રોસેસર ઘણા ઘટકોથી બનેલું છે. CPU નીચેના કાર્યો કરે છે:

- તે તમામ ગણતરીઓ કરે છે.
- તે તમામ નિર્ણયો લે છે.
- તે કોમ્પ્યુટરના તમામ એકમોને નિયંત્રિત કરે છે.

CPU ના બે લાક્ષણિક ઘટકો નીચે મુજબ છે:

અંકગણિત તર્ક એકમ (ALU), જે અંકગણિત અને તાર્કિક કામગીરી કરે છે. કંટ્રોલ યુનિટ (CU), જે મેમરીમાંથી સૂચનાઓ કાઢે છે અને કીકોડ કરે છે અને તેને એક્ઝિક્યુટ કરે છે, જ્યારે જરૂરી હોય ત્યારે ALU પર કોલ કરે છે.

મેમરી

મેમરી સ્ટોર કરવા માટે વપરાતા ભૌતિક ઉપકરણનો સંદર્ભ આપે છે માટે અસ્થાયી અથવા કાયમી ધોરણે પ્રોગ્રામ અથવા ડેટા કમ્પ્યુટર અથવા અન્ય ડિજિટલ ઇલેક્ટ્રોનિક ઉપકરણમાં ઉપયોગ કરો. કમ્પ્યુટરમાં બે પ્રકારની મેમરી હોય છે.

- પ્રાથમિક મેમરી
- ગૌણ મેમરી

આઉટપુટ યુનિટ

આઉટપુટ યુનિટ પ્રોસેસિંગમાંથી માહિતી મેળવે છે એકમ અને માનવ વાંચી શકાય તેવા સ્વરૂપમાં પરિણામો પ્રદાન કરો.

આઉટપુટ ઉપકરણો

કેટલાક આઉટપુટ ઉપકરણો છે

- મોનીટર
- પ્રિન્ટર

- કાવતરાખોર
- સ્પીકર

કમ્પ્યુટર્સની એપ્લિકેશનો

વિજ્ઞાન: વૈજ્ઞાનિકો સિદ્ધાંતો વિકસાવવા અને ડેટાનું પૃથ્થકરણ અને પરીક્ષણ કરવા માટે કમ્પ્યુટરનો ઉપયોગ કરી રહ્યાં છે. આ કમ્પ્યુટરની ઉચ્ચ ગતિ અને ચોકસાઈ વિવિધ વૈજ્ઞાનિક વિશ્લેષણો હાથ ધરવા દે છે. ભૂકંપની કેવી અસર થાય છે તેના વિગતવાર અભ્યાસ માટે તેનો ઉપયોગ કરી શકાય છે. ઇમારતો અથવા પ્રદૂષણ હવામાનની પેટર્નને અસર કરે છે. ઉપગ્રહ-આધારિત એપ્લિકેશનો વિના શક્ય નથી કમ્પ્યુટર્સનો ઉપયોગ. તદ્દુપરાંત, તે શક્ય બનશે નહીં કમ્પ્યુટર વિના સૌરમંડળ અને બ્રહ્માંડની માહિતી મેળવો.

શિક્ષણ: કમ્પ્યુટરોએ પણ શિક્ષણની સમગ્ર પ્રક્રિયામાં ક્રાંતિ કરી છે. હાલમાં, વર્ગખંડો, પુસ્તકાલયો અને સંગ્રહાલયો અસરકારક રીતે કમ્પ્યુટરનો ઉપયોગ કરી રહ્યાં છે. શિક્ષણને વધુ રસપ્રદ બનાવવા માટે. વિપરીત રેકૉર્ડ ટેલિવિઝન શો, કમ્પ્યુટર-સહાયિત શિક્ષણ (CAE) અને કમ્પ્યુટર આધારિત તાલીમ (CBT) પેકેજો છે. શીખવાનું વધુ ઇન્ટરેક્ટિવ બનાવે છે.

દવા અને આરોગ્ય સંભાળ: ત્યાં એક છે દવાના ક્ષેત્રમાં કમ્પ્યુટરનો વધતો ઉપયોગ. હવે, ડોક્ટરો નિદાનથી માંકીને કમ્પ્યુટરનો ઉપયોગ કરી રહ્યા છે જટિલ દરમિયાન દર્દીની સ્થિતિનું નિરીક્ષણ કરવા માટે બીમારી શસ્ત્રક્રિયા સ્વયંસંચાલિત ઇમેજિંગ તકનીકોનો ઉપયોગ કરીને, ડોક્ટરો વ્યક્તિના શરીરની અંદર જોવા માટે સક્ષમ છે અને 15 અભ્યાસ કરી શકે છે દરેક અંગ વિગતવાર (દા.ત. સીટી સ્કેન અથવા એમઆરઆઈ સ્કેન), જે થોડા વર્ષો પહેલા શક્ય ન હતું. ત્યાં ઘણા છે ઓપરેટ કરી શકે તેવા ખાસ હેતુવાળા કમ્પ્યુટરના ઉદાહરણો માનવ શરીરની અંદર જેમ કે કોકલીયર ઇમ્પ્લાન્ટ, એ ખાસ પ્રકારની શ્રવણ સહાય જે તેને શક્ય બનાવે છે બહેરા લોકો સાંભળવા માટે.

એન્જિનિયરિંગ/આર્કિટેક્ચર/મેન્યુફેક્ચરિંગ: ઘ આર્કિટેક્ટ અને એન્જિનિયરો ડિઝાઇનિંગ અને ડ્રોઇંગમાં કમ્પ્યુટરનો વ્યાપક ઉપયોગ કરી રહ્યા છે. કમ્પ્યુટર એવી વસ્તુઓ બનાવી શકે છે જે ત્રણેય પરિમાણમાંથી જોઈ શકાય છે. ઉપયોગ કરીને વર્ચ્યુઅલ રિયાલિટી જેવી તકનીકો, આર્કિટેક્ટ ઘરોની શોધખોળ કરી શકે છે જે ડિઝાઇન કરવામાં આવી છે પરંતુ બાંધવામાં આવી નથી. ઉત્પાદન ફેક્ટરીઓ જોખમી કામ કરવા માટે કમ્પ્યુટરાઈઝડ રોબોટિક્સમાં ઉપયોગ કરી રહી છે. ઉપરાંત, કમ્પ્યુટર સહાયિત ઉત્પાદન (CAM) નો ઉપયોગ ઉત્પાદનની ડિઝાઇન, ઓર્ડર આપવા માટે કરી શકાય છે ભાગો અને આયોજન ઉત્પાદન. આમ, કમ્પ્યુટર્સ સમગ્ર ઉત્પાદન પ્રક્રિયામાં સમન્વય કરવામાં મદદ કરે છે.

મનોરંજન: એન્ટરટેઇનમેન્ટ ઇન્ડસ્ટ્રીમાં કમ્પ્યુટરનો વધુ ઉપયોગ થઈ રહ્યો છે. તેનો ઉપયોગ છબીઓ અને અવાજોને નિયંત્રિત કરવા માટે થાય છે. ખાસ અસરો, જે પ્રેક્ષકોને મંત્રમુગ્ધ કરો, કમ્પ્યુટર વિના શક્ય ન હોત. આ ઉપરાંત કમ્પ્યુટરાઈઝડ એનિમેશન અને રંગીન ગ્રાફિક્સ ફિલ્મ ઉદ્યોગને આધુનિક બનાવ્યો છે.

કોમ્યુનિકેશન: ઈ-મેઇલ અથવા ઇલેક્ટ્રોનિક મેઇલ તેમાંથી એક છે કોમ્યુનિકેશન મીડિયા જેમાં કમ્પ્યુટરનો ઉપયોગ થાય છે. દ્વારા કમ્પ્યુટર અને ટેલિફોન લાઇનની સહાયથી ઈ-મેલ, સંદેશા અને અહેવાલો એક વ્યક્તિ પાસેથી એક અથવા વધુ વ્યક્તિઓને પસાર કરવામાં આવે છે. આ સેવાનો ફાયદો એ છે કે જ્યારે સંદેશાઓને સ્થાનાંતરિત કરવાથી તે સમય બચાવે છે, બગાડ ટાળે છે કાગળ અને તેથી વધુ. વધુમાં, જે વ્યક્તિ પ્રાપ્ત કરી રહી છે સંદેશાઓ જ્યારે પણ હોય ત્યારે સંદેશા વાંચી શકે છે મફત અને તેને સાચવી શકો છો, તેને જવાબ આપી શકો છો, તેને ફોરવર્ડ કરી શકો છો

અથવા તેને કમ્પ્યુટરમાંથી કાઢી શકો છો.

બિઝનેસ એપ્લિકેશન: આ એક મહત્વપૂર્ણ ઉપયોગ છે કમ્પ્યુટરનું. શરૂઆતમાં, બેચ માટે કોમ્પ્યુટરનો ઉપયોગ થતો હતો- પ્રક્રિયા કરવાની નોકરીઓ, જ્યાં કોઈને જરૂર નથી કમ્પ્યુટર તરફથી તાત્કાલિક પ્રતિસાદ. હાલમાં, કમ્પ્યુટર્સનો ઉપયોગ મુખ્યત્વે રીઅલ-ટાઇમ એપ્લિકેશન્સ માટે થાય છે (જેમ કે વેચાણ કાઉન્ટર પર) કે જેને તાત્કાલિક પ્રતિસાદની જરૂર છે કમ્પ્યુટરમાંથી. જ્યાં વિવિધ ચિંતાઓ છે કમ્પ્યુટર્સનો ઉપયોગ થાય છે જેમ કે બિઝનેસ ફોરકાસ્ટિંગમાં બેંકિંગમાં પગાર બિલ અને વ્યક્તિગત રેકૉર્ડ તૈયાર કરો વિવિધ પ્રકારના જીવનમાં કામગીરી અને ડેટા સ્ટોરેજ વીમા વ્યવસાય અને વ્યવસ્થાપન માટે સહાય તરીકે. વ્યવસાયો પણ કમ્પ્યુટરના નેટવર્કિંગનો ઉપયોગ કરી રહ્યા છે, જ્યાં સંખ્યાબંધ કોમ્પ્યુટર એકસાથે જોડાયેલા છે ડેટા અને માહિતી શેર કરવા માટે. ઈ-મેલનો ઉપયોગ અને ઈન્ટરનેટ વેપાર કરવાની રીતો બદલી નાખી છે.

પ્રકાશન: કોમ્પ્યુટરે એક ક્ષેત્ર બનાવ્યું છે જે તરીકે ઓળખાય છે ડેસ્કટોપ પબ્લિશિંગ (DTP). DTP માં, ની મદદ સાથે કોમ્પ્યુટર અને લેસર પ્રિન્ટર પ્રકાશનનું કામ જાતે કરી શકે છે. જરૂરી કાર્યો ઘણા લાંબા મેન્યુઅલ કલાકો જેમ કે સામગ્રીઓનું કોષ્ટક બનાવવું અને કોમ્પ્યુટર અને ડીટીપી સોફ્ટવેરનો ઉપયોગ કરીને ઈન્ડેક્સ આપમેળે કરી શકાય છે.

બેંકિંગ: બેંકિંગ અને ફાઇનાન્સના ક્ષેત્રમાં કોમ્પ્યુટરનો વ્યાપકપણે ઉપયોગ થાય છે. લોકો એટીએમ (ઓટોમેટેડ ટેલર મશીન) રોકડ જમા કરવા અને ઉપાડવા માટે 24 કલાક સેવા આપે છે. જ્યારે બેંકની વિવિધ શાખાઓ કોમ્પ્યુટર નેટવર્ક દ્વારા જોડાયેલ હોય છે, ત્યારે આંતર શાખા ચેક અને ડ્રાફ્ટ જેવા વ્યવહારો કોઈપણ વિલંબ વિના કરી શકાય છે.

રેલ્વે રિઝર્વેશન સિસ્ટમ

આ સિસ્ટમનો ઉપયોગ કરીને, વપરાશકર્તા ઓનલાઇન દ્વારા નીચેની કામગીરી કરી શકે છે. (વેબ સાઈટ: www.irtctc.co.in)

- ટ્રેન અને તેનો સમય શોધો
- સીટો અને જન્મની ઉપલબ્ધતા તપાસો
- ટિકિટો બુક કરવી અને રદ કરવી
- PNR ની સ્થિતિ (પેસેન્જર નેમ રેકૉર્ડ)

ટેલિફોન / વીજળી બોર્ડ બિલિંગ:

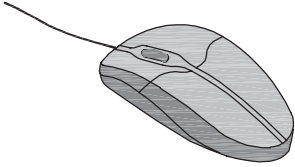
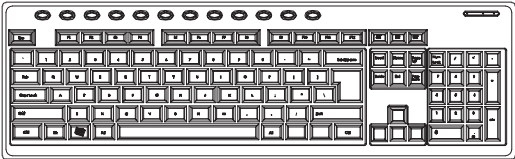
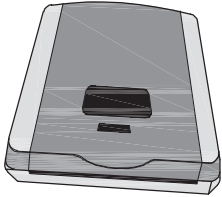
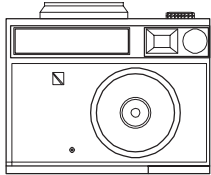
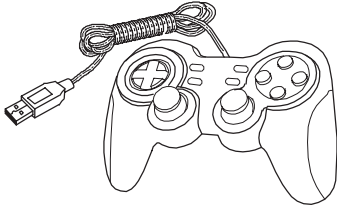
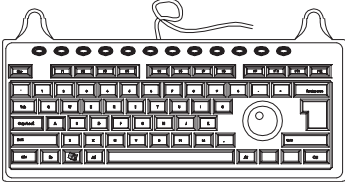
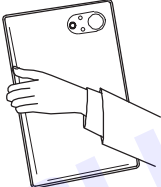
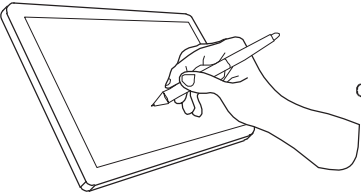
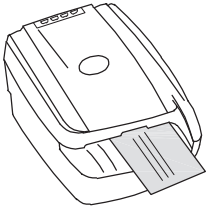
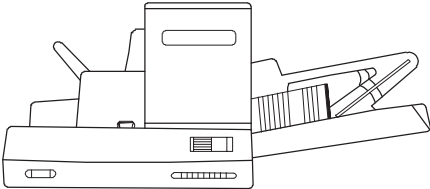
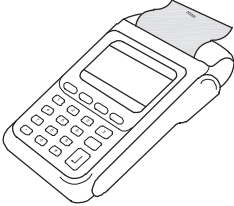
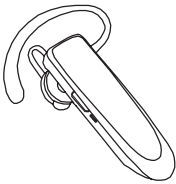
વપરાશકર્તાઓ આ સિસ્ટમનો ઉપયોગ કરીને ઓનલાઇન દ્વારા નીચેની કામગીરી કરી શકે છે. (વેબ સાઈટ: portal.bsnl.in - BSNL)

- ટેલિફોન/વીજળી બોર્ડ નંબર રજીસ્ટર કરો
- ચેક કરો અને બિલની રકમ ચૂકવો
- ફરિયાદો નોંધો

ઈ-ગવર્નન્સ

ઈ-ગવર્નન્સનો અર્થ ટેકનોલોજી આધારિત શાસન છે. ઈ-ગવર્નન્સ એ માહિતીની એપ્લિકેશન છે અને પહોંચાડવા માટે કોમ્યુનિકેશન ટેકનોલોજી (ICT) સરકારી સેવાઓ, માહિતીનું વિનિમય સંચાર વ્યવહારો, વિવિધ સ્ટેન્ડનું એકીકરણ- સરકાર-થી- વચ્ચે એકલી સિસ્ટમ અને સેવાઓ નાગરિકો (G2C), સરકાર-થી- વ્યવસાય(G2B),સરકાર-થી-સરકાર(G2G) તરીકે તેમજ બેક ઓફિસ પ્રક્રિયાઓ અને અંદરની ક્રિયાપ્રતિક્રિયાઓ સમગ્ર સરકારી ફ્રેમ વર્ક.

ઈ-ગવર્નન્સ તમામ ક્ષેત્રોને ધ્યાનમાં રાખીને આવરી લે છે મુશ્કેલી મુક્ત, પારદર્શક અને કાર્યક્ષમ સેવા પૂરી પાડવી સામાન્ય માણસ માટે (શહેરી અને ગ્રામીણ બંને વિસ્તારોમાં).

Fig 1		MOUSE		KEYBOARD
		SCANNER		CAMERA
		WEB CAMERA		JOY STICK
		TRACK BALL		TOUCH PAD
		LIGHT PEN		BAR CODE READER
		MICROPHONE		GRAPHIC TABLET
		MAGNETIC INK CHARACTER READER		OPTICAL MARK READER
		MAGNETIC CARD READER		BLUETOOTH

CPU આર્કિટેક્ચર અને મધરબોર્ડનો પરિચય (Introduction to CPU architecture and motherboard)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- હાર્ડવેર શું છે તે જણાવો અને આંતરિક અને બાહ્ય હાર્ડવેરનું પરીક્ષણ કરો
- સૂચિબદ્ધ હાર્ડવેરને સંક્ષિપ્ત કરો
- પાર્ટીશનો શું છે અને તેમના પ્રકારો જણાવો
- બુટીંગ અને તેની પ્રક્રિયાઓ સમજાવો.

કમ્પ્યુટર હાર્ડવેર

કમ્પ્યુટરના ભૌતિક એકમોને કમ્પ્યુટરનું હાર્ડવેર કહેવામાં આવે છે.

આંતરિક હાર્ડવેર ઉદાહરણો

- બ્લુ-રે, CD-ROM, અને DVD
- સી.પી.યુ
- હાર્ડ ડ્રાઇવ
- મધરબોર્ડ
- રામ
- સાઉન્ડ કાર્ડ
- વીડિઓ કાર્ડ
- SMPS

બાહ્ય હાર્ડવેર ઉદાહરણો

- ફ્લેટ-પેનલ, મોનિટર અને LCD
- કીબોર્ડ
- માઉસ
- પ્રિન્ટર
- સ્કેનર

CPU & ALU

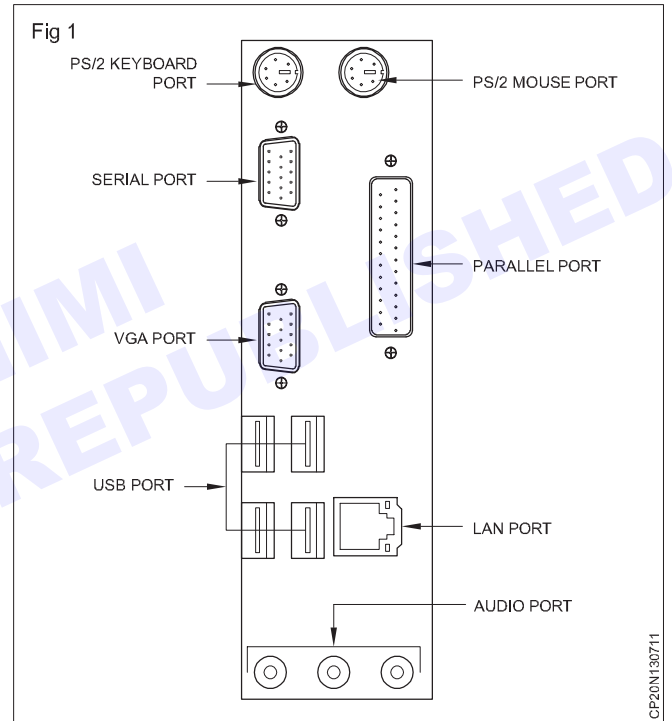
સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (CPU) એ કોમ્પ્યુટરનું હૃદય છે. તે હાર્ડવેર છે, જે બહાર પાડે છે સિસ્ટમના મૂળભૂત અંકગણિત, તાર્કિક, અને ઈનપુટ/આઉટપુટ કામગીરી કરીને કમ્પ્યુટર પ્રોગ્રામની સૂચનાઓ.

CPU અને ALU સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (CPU) એ કોમ્પ્યુટરનું હૃદય છે. તે હાર્ડવેર છે, જે બહાર પાડે છે સિસ્ટમના મૂળભૂત અંકગણિત, તાર્કિક, અને ઈનપુટ/આઉટપુટ કામગીરી કરીને કમ્પ્યુટર પ્રોગ્રામની સૂચનાઓ.

CPU પોર્ટ્સ અને કનેક્ટર્સ

પોર્ટ એ કમ્પ્યુટર કેબિનેટની પાછળનું કનેક્ટર છે જ્યાં તમે પ્રિન્ટર જેવા બાહ્ય ઉપકરણને પ્લગ ઈન કરો છો, કીબોર્ડ, સ્કેનર, મોડેમ વગેરે. આ સૂચનાઓને મંજૂરી આપે છે અને કોમ્પ્યુટર અને ઉપકરણ વચ્ચેના પ્રવાહ માટે ડેટા. કોમ્પ્યુટર પોર્ટને સામાન્ય રીતે તરીકે પણ ઓળખવામાં આવે છે ઈનપુટ/આઉટપુટ પોર્ટ્સ (I/O પોર્ટ). આ બંદરો હોઈ શકે છે સીરીયલ અથવા સમાંતર. ફિગ 1 વ્યક્તિગત કમ્પ્યુટર પર સામાન્ય રીતે ઉપલબ્ધ પોર્ટ્સ બતાવે છે.

મોટાભાગના કનેક્ટર્સને અલગ કરવામાં આવે છે, કેબલને પરવાનગી આપે છે ફક્ત સાચી દિશામાં જ પ્લગ ઈન કરો. કીબોર્ડ અને માઉસ “PS2” (પર્સનલ સિસ્ટમ 2) કનેક્ટર્સનો ઉપયોગ કરે છે. PS2 કનેક્ટર્સ કલર-કોડેડ છે. જાંબલી કનેક્ટર કીબોર્ડ માટે છે. ગ્રીન કનેક્ટર છે માઉસ માટે.



- PS/2 પોર્ટ્સ: પ્રમાણભૂત કીબોર્ડ અને માઉસ વારંવાર PS/2 પોર્ટ દ્વારા કમ્પ્યુટર સાથે કનેક્ટ કરો. પ્લગ કરવા માટે કીબોર્ડ અથવા માઉસ કેબલમાં, પ્રથમ કેબલ સાથે મેચ કરો કનેક્ટર માટે. પછી કેબલને માં દબાવો કરો કનેક્ટર કારણ કે કનેક્ટરને દબાવો ન કરવાની ખાતરી કરો તમે પિનને વળાંક આપશો
- સીરીયલ અને સમાંતર પોર્ટ્સ: સીરીયલ પોર્ટ અને સમાંતર પોર્ટ પ્રિન્ટરો અને અન્ય બાહ્ય સાથે જોડાણોને મંજૂરી આપે છે ઉપકરણો સીરીયલ પોર્ટ દ્વારા બાઈટ ટ્રાન્સફર કરવા માટે, આઠ બિટ્સ કતારબદ્ધ છે અને થોડીવારે મોકલવામાં આવે છે. જો કે, માં એ સમાંતર પોર્ટ, તમામ આઠ બિટ્સ એકસાથે ટ્રાન્સફર કરવામાં આવે છે.
- સમાંતર પોર્ટ, સીરીયલ પોર્ટ અને વિડીયો પોર્ટ બધા “D” પ્રકારના કનેક્ટર્સનો ઉપયોગ કરે છે (DB-25M, DB-9M અને DB-15F અનુક્રમે). આને તેમના આકારને કારણે D કનેક્ટર્સ કહેવામાં આવે છે, જે કેબલ્સને ફક્ત એક જ રીતે પ્લગ કરવાની પરવાનગી આપે છે.

યુએસબી (યુનિવર્સલ સીરીયલ બસ) પોર્ટ્સ

ડિજીટલ કેમેરા, સ્કેનર્સ અને પ્રિન્ટર જેવા ઉપકરણો વારંવાર USB પોર્ટ દ્વારા મધરબોર્ડ સાથે કનેક્ટ કરો. એક યુએસબી કનેક્ટરનો વિશિષ્ટ લંબચોરસ આકાર તેને સરળતાથી ઓળખી શકાય તેવું બનાવે છે.

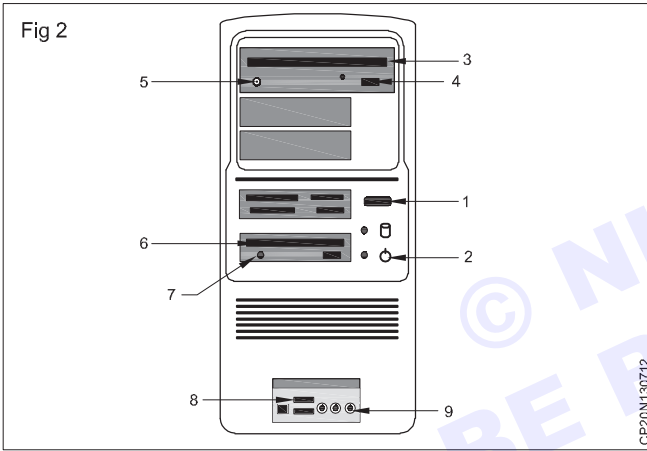
યુએસબીમાં સંખ્યાબંધ સુવિધાઓ છે જે તેને ખાસ કરીને પીસી પર લોકપ્રિય બનાવે છે. પ્રથમ, યુએસબી ઉપકરણો ગરમ સ્વેપ કરી શકાય તેવા છે. તમે તમારી સિસ્ટમને પુનઃપ્રારંભ કર્યા વિના તેમને દાખલ અથવા દૂર કરી શકો છો.

LAN (લોકલ એરિયા નેટવર્ક) પોર્ટ: LAN પોર્ટનો ઉપયોગ PC ને સ્થાનિક નેટવર્ક અથવા હાઈ સ્પીડ ઇન્ટરનેટ સેવાઓ સાથે જોડવા માટે થાય છે.

VGA (વિડિયો ગ્રાફિક્સ એર) પોર્ટ્સ: VGA પોર્ટ સંકલિત વિડિયોની એક્સેસ પ્રદાન કરે છે.

ઓડિયો પોર્ટ્સ: તે સંકલિત ઓડિયોની એક્સેસ પ્રદાન કરે છે. ઓડિયો જેક સૌથી વધુ ગૂંચવણમાં મૂકનારા કનેક્ટર્સ છે પાછળની પેનલ. જોકે જેક ક્યારેક રંગીન હોય છે- કોડેડ, ઉપકરણો કે જે ભાગ્યે જ તેમાં પ્લગ થાય છે.

CPU ફ્રન્ટ પેનલ ફિગ 2 બતાવેલ છે.



તે નીચેના ભાગો સમાવી શકે છે.

- પાવર ચાલુ/બંધ સ્વિચ
- પાવર સૂચક
- CD/DVD ડ્રાઈવ
- CD/DVD ડ્રાઈવ ઓપન/ક્લોઝ બટન
- CD/DVD ડ્રાઈવ સૂચક
- ફ્લોપી ડિસ્ક ડ્રાઈવ
- ફ્લોપી ડિસ્ક ડ્રાઈવ સૂચક
- યુએસબી પોર્ટ્સ
- ઓડિયો અને માઈક કનેક્ટર્સ

સીપીયુ કેબિનેટની ડિઝાઇન તેના આધારે બદલાઈ શકે છે ઉત્પાદક

સિસ્ટમ યુનિટ અને તેના ઘટકો

સિસ્ટમ યુનિટ એ બોક્સ જેવું એકમ છે જે સંખ્યાબંધ સંખ્યાથી ભરેલું છે ઉપયોગી ઘટકો, દરેક એક અલગ કાર્ય કરે છે. આ ઘટકો મુખ્ય પરિપૂર્ણ કરવા માટે એકસાથે કામ કરે છે કમ્પ્યુટરનું કાર્ય, જેમ કે. ઇનપુટ સ્વીકારો અને પ્રક્રિયા કરો અને આઉટપુટ પહોંચાડો. આ વિભાગ આ વિશે વિગતવાર જણાવશે એક પછી એક ઘટકો. ફિગ 3 સિસ્ટમ યુનિટના વિવિધ ઘટકો બતાવે છે.



વીજ પુરવઠો

પાવર પ્રદાન કરવા માટે પીસીમાં લગભગ દરેક ઉપકરણ સાથે પાવર સપ્લાય જોડાયેલ છે. તે કેસના પાછળના ભાગમાં સ્થિત છે. સિસ્ટમ યુનિટ પાવર પ્રોટેક્શન ડિવાઈસ દ્વારા એસી મેઈન્સમાંથી પાવર ખેંચે છે.

આ શક્તિ સીધી આંતરિકને પૂરી પાડવામાં આવતી નથી ઘટકો તેના બદલે, ઘટકોમાંથી એક, જેને કહેવાય છે આંતરિક વીજ પુરવઠો, એસી ઇનપુટને ડીસીમાં રૂપાંતરિત કરે છે 5 અને 12 વોલ્ટનું આઉટપુટ. સામાન્ય રીતે, આંતરિક શક્તિ પુરવઠાને સ્વિચ્ઝ મોડ પાવર સપ્લાય તરીકે ઓળખવામાં આવે છે (SMPS).

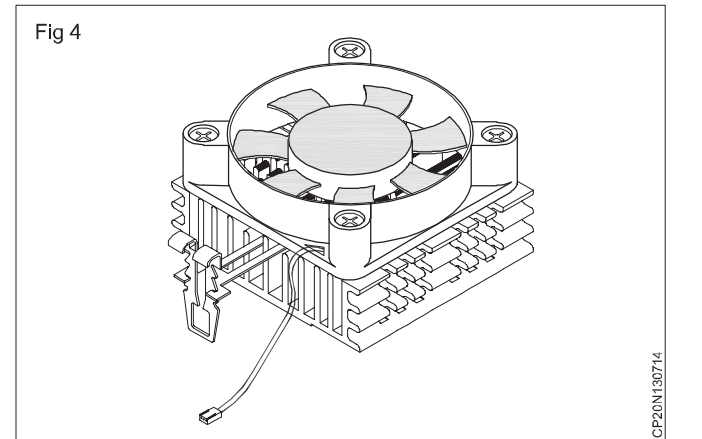
SMPS સપ્લાય કરવા માટે કેબલ કનેક્ટર્સ પ્રદાન કરે છે જેવા અન્ય આંતરિક ઘટકો માટે જરૂરી વોલ્ટેજ ફ્લોપી ડ્રાઈવ્સ, હાર્ડ ડિસ્ક ડ્રાઈવ, મધરબોર્ડ અને બાહ્ય ઉપકરણ જેમ કે કીબોર્ડ. ચાલુ/બંધ સિસ્ટમ યુનિટની સ્વીચ ખરેખર SMPS નો એક ભાગ છે.

પંખો

SMPS પાસે એક નાનો પંખો છે, જેને એકઝોસ્ટ ફેન કહેવાય છે, SMPS સાથે જોડાયેલ (ફિગ. 4). જેટલો લાંબો પંખો ફરે છે કમ્પ્યુટર ચાલુ છે. તેનું કાર્ય SMPS યુનિટને ઠંડુ કરવાનું છે.

ડ્રાઈવ બેઝ: 5.25" અને 3.5" ડ્રાઈવ બેઝ ધરાવે છે કમ્પ્યુટરમાં ઘણા પ્રકારના સ્ટોરેજ ઉપકરણો હોઈ શકે છે.

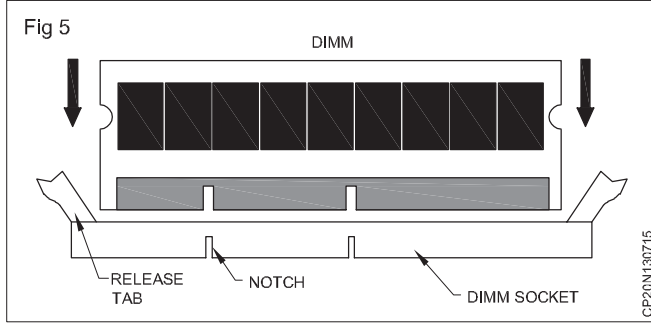
વિસ્તરણ સ્લોટ: અને વિસ્તરણ સ્લોટ સ્થિત થયેલ સ્લોટ છે કમ્પ્યુટર મધર બોર્ડની અંદર જે વધારાના પેરિફેરલ્સને તેની સાથે કનેક્ટ કરવાની મંજૂરી આપે છે.



મેમરી સ્લોટ

મેમરી સ્લોટનો ઉપયોગ રેન્ડમ એક્સેસ દાખલ કરવા માટે થાય છે

મેમરી(RAM) ફિગ 5 માં બતાવેલ છે



સ્ટોરેજ ડ્રાઈવરો

સ્ટોરેજ ડ્રાઈવરો જેમ કે હાર્ડ ડ્રાઈવો, ઓપ્ટિકલ ડ્રાઈવો અને ફ્લોપી ડ્રાઈવ તમામ મધરબોર્ડ સાથે કેબલ દ્વારા કનેક્ટ થાય છે અને કમ્પ્યુટરની અંદર માઉન્ટ થયેલ છે.

IDE અને SATA કેબલ્સ: ફિગ 6 બે હાર્ડ ડિસ્ક ડ્રાઈવ બતાવે છે જે મધરબોર્ડ સાથે અલગ અલગ રીતે જોડાય છે. એક જૂના IDE કેબલ કનેક્શનનો ઉપયોગ કરે છે જ્યારે અન્ય ઉપયોગ કરે છે SATA(સીરીયલ એડવાન્સ ટેકનોલોજી એટેચમેન્ટ) કેબલ જે ઝડપી હાર્ડ ડ્રાઈવ એક્સેસ માટે પ્રદાન કરે છે.



પાવર પોર્ટ: કેબલ દ્વારા વાહન ચલાવવા માટે પાવર પહોંચાડવામાં આવે છે ડ્રાઈવ પર પાવર પોર્ટમાં પ્લગ કરો.

પેરિફેરલ કાર્ડ સ્લોટ

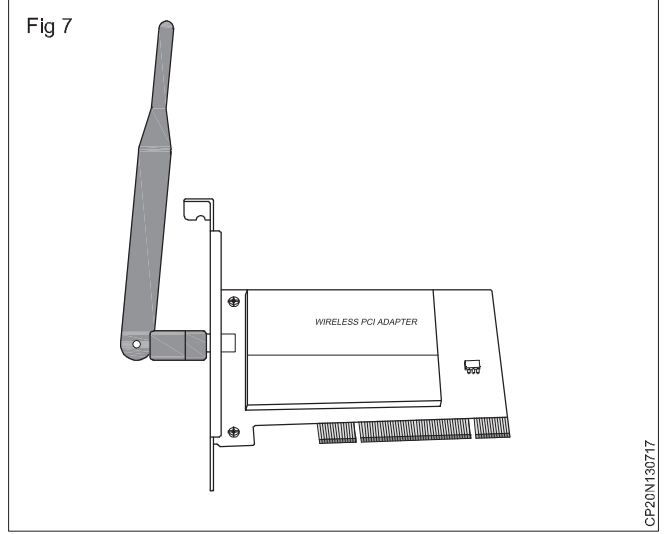
પેરિફેરલ કાર્ડ્સ ફાજલ વિસ્તરણ સ્લોટ છે જે પેરિફેરલ કાર્ડ્સ પર મધર બોર્ડ પર ઉપલબ્ધ છે દાખલ કરી શકાય છે.

નીચેના પેરિફેરલ કાર્ડ્સ છે

- સાઉન્ડ કાર્ડ
- વીડિઓ કાર્ડ
- મોડમ
- તાર વગર નુ તંત્ર

ફિગ. 7 પેરિફેરલ કાર્ડ બતાવે છે, જે PCI(પેરિફેરલ કમ્પોનન્ટ ઇન્ટરફેસ) કનેક્ટર સાથે ડિઝાઇન કરેલું છે.

Fig 7



મધરબોર્ડ

મધરબોર્ડ એ પ્રિન્ટેડ સર્કિટ છે જે પાયો છે કમ્પ્યુટરનું અને CPU, RAM અને અન્ય તમામ કમ્પ્યુટર હાર્ડવેર ઘટકોને ફિગ 8 મુજબ એકબીજા સાથે કામ કરવાની મંજૂરી આપે છે.



મધરબોર્ડ એ સમગ્રનો પ્રાથમિક ઘટક છે સિસ્ટમ મધર બોર્ડ એ એક વિશાળ બોર્ડ છે જેમાં a નાના ઇલેક્ટ્રોનિક સર્કિટ અને અન્ય ઘટકોની સંખ્યા દૃશ્યમાન છે. બધા પેરિફેરલ ઉપકરણો મધરબોર્ડ સાથે જોડાયેલા છે. મધરબોર્ડના ઘટકો છે:

- કીબોર્ડ / માઉસ પોર્ટ
- સમાંતર અને સીરીયલ પોર્ટ
- પ્રોસેસર સોકેટ
- એજીપી સ્લોટ
- PCI સ્લોટ્સ
- ISA સ્લોટ
- CMOS બેટરી
- ડેટા કાર્ડ કનેક્ટર
- મેમરી સ્લોટ્સ
- ફ્લોપી પોર્ટ
- ફેન હેડર
- મુખ્ય પાવર કનેક્ટર

ફ્લોપી પોર્ટ : ફ્લોપી ડ્રાઈવ કોમ્પ્યુટર સાથે જોડાય છે 34-પિન રિબન કેબલ દ્વારા, જે બદલામાં મધરબોર્ડ સાથે જોડાય છે. ફ્લોપી કંટ્રોલર એ છે જેનો ઉપયોગ ફ્લોપી ડ્રાઈવને નિયંત્રિત કરવા માટે થાય છે.

RAM સ્લોટ્સ: રેન્ડમ-એક્સેસ મેમરી (RAM) સ્ટોર્સ પ્રોગ્રામ્સ અને ડેટા હાલમાં CPU દ્વારા ઉપયોગમાં લેવાય છે.

RAM ને બાઈટ તરીકે ઓળખાતા એકમોમાં માપવામાં આવે છે. RAM ને ઘણી અલગ અલગ રીતે પેક કરવામાં આવી છે

- SIMM-સિંગલ ઇનલાઈન મેમરી મોડ્યુલ -32 અથવા 72 પિન
- DIMM- ડ્યુઅલ ઇનલાઈન મેમરી મોડ્યુલ -168 પિન.

DIMM મોડ્યુલના મોટાભાગના PC ના ઉપયોગોમાં

ROM BIOS ચિપ: આનો અર્થ રીડ ઓન્લી મેમરી બેઝિક ઇનપુટ-આઉટપુટ સિસ્ટમ છે.

બિલ્ટ-ઇન સોફ્ટવેર જે નક્કી કરે છે કે કોમ્પ્યુટર શું છે ડિસ્કમાંથી પ્રોગ્રામ્સ એક્સેસ કર્યા વિના કરી શકે છે. પીસી પર, BIOS માં કીબોર્ડ, ડિસ્ક્લેસ્ક્રીન, ડિસ્કડ્રાઈવ્સ, સીરીયલ કોમ્યુનિકેશન્સ અને વિવિધ કાર્યોની સંખ્યાને નિયંત્રિત કરવા માટે જરૂરી તમામ કોડ હોય છે.

BIOS સામાન્ય રીતે રોમ ચિપમાં મૂકવામાં આવે છે જે આવે છે કમ્પ્યુટર સાથે (તેને ઘણીવાર **ROM BIOS** કહેવામાં આવે છે). આ ખાતરી કરે છે કે BIOS હંમેશા ઉપલબ્ધ રહેશે અને રહેશે ડિસ્ક નિષ્ફળતા દ્વારા નુકસાન ન થાય.

તે કમ્પ્યુટર માટે પોતે બુટ કરવાનું પણ શક્ય બનાવે છે. કારણ કે RAM એ ROM કરતાં ઝડપી છે, જોકે, ઘણી કમ્પ્યુટર ઉત્પાદકો સિસ્ટમ ડિઝાઇન કરે છે જેથી દરેક વખતે BIOS ને ROM થી RAM પર કોપિ કરવામાં આવે છે કમ્પ્યુટર બુટ થયેલ છે. આ એશોરોઈંગ તરીકે ઓળખાય છે.

ઘણા આધુનિક પીસીમાં ફ્લેશ BIOS છે, જેનો અર્થ છે BIOS ને ફ્લેશ મેમરી ચિપ પર રેકોર્ડ કરવામાં આવ્યું છે, જે જરૂર પડ્યે અપડેટ કરી શકાય છે. પીસી BIOS એકદમ છે પ્રમાણિત છે, તેથી બધા પીસી આ સ્તરે સમાન છે (જોકે ત્યાં વિવિધ BIOS સંસ્કરણો છે). વધારાના DOS કાર્યો સામાન્ય રીતે સોફ્ટવેર મોડ્યુલો દ્વારા ઉમેરવામાં આવે છે.

આનો અર્થ એ છે કે તમે DOS ના નવા સંસ્કરણ પર અપગ્રેડ કરી શકો છો BIOS બદલ્યા વિના. PC BIOS જે હેન્ડલ કરી શકે છે પ્લગ-એન્ડ-પ્લે (PnP) ઉપકરણોને PnPBIOS તરીકે ઓળખવામાં આવે છે, આ BIOS હંમેશા રોમને બદલે ફ્લેશ મેમરી સાથે લાગુ કરવામાં આવે છે.

CMOS બેટરી

CMOS (પૂરક મેટલ-ઓક્સાઇડ-સેમિકન્ડક્ટર) છે સામાન્ય રીતે નાની રકમનું વર્ણન કરવા માટે વપરાતો શબ્દ કમ્પ્યુટર મધરબોર્ડ પર મેમરી કે જે BIOS સેટિંગ્સ સંગ્રહિત કરે છે.

મોટાભાગની CMOS બેટરી એ જીવનપર્યંત ચાલશે મધરબોર્ડ (મોટા ભાગના કિસ્સાઓમાં 10 વર્ષ સુધી) પરંતુ કરશે ક્યારેક બદલવાની જરૂર છે. ખોટી અથવા ધીમી સિસ્ટમ તારીખ અને સમય અને BIOS સેટિંગ્સની ખોટ એ મૃત અથવા મૃત્યુ પામેલી CMOS બેટરીના મુખ્ય સંકેતો છે.

ISA સ્લોટ: (ઇન્સ્ટ્રી સ્ટાન્ડર્ડ આર્કિટેક્ચર) તે છે વિસ્તરણ બસનું પ્રમાણભૂત આર્કિટેક્ચર. મધરબોર્ડ ISA સુસંગત કાર્ડ્સને કનેક્ટ કરવા માટે કેટલાક સ્લોટ્સ સમાવી શકે છે.

PCI સ્લોટ : ઇન્ટેલે પેરિફેરલ કમ્પોનન્ટ રજૂ કર્યું ઇન્ટરફેસ બસ પ્રોટોકોલ. પીસીઆઈ બસનો ઉપયોગ થાય છે I/O ઉપકરણોને કમ્પ્યુટરના મુખ્ય તર્ક સાથે જોડો. PCI બસે ISA બસનું સ્થાન લીધું છે. PC મધરબોર્ડમાં એક PCI સ્લોટ છે પરંતુ સામાન્ય રીતે એક કરતાં વધુ.

PCI બસ આર્કિટેક્ચર એ પ્રોસેસર-સ્વતંત્ર બસ છે સ્પષ્ટીકરણ કે જે પેરિફેરલ્સને સીપીયુનો ઉપયોગ કર્યા વિના સીધું સિસ્ટમ મેમરીને એક્સેસ કરવાની પરવાનગી આપે છે.

એજીપી સ્લોટ: એક્સિલરેટેડ ગ્રાફિક્સ પોર્ટ (એજીપી) એ કમ્પ્યુટરના મધરબોર્ડ સાથે વિડિયો કાર્ડ જોડવા માટે હાઈ-સ્પીડ પોઈન્ટ-ટુ-પોઈન્ટ ચેનલ છે.

પાવર સપ્લાય પ્લગ ઇન

પાવર સપ્લાય, તેના નામ પ્રમાણે, પૂરી પાડે છે પીસી બનાવવા માટે જરૂરી વિદ્યુત શક્તિ (વ્યક્તિગત કમ્પ્યુટર) ચલાવે છે. વીજ પુરવઠો પ્રમાણભૂત લે છે 110-V AC પાવર અને +/-12-વોલ્ટ, +/-5-વોલ્ટમાં રૂપાંતરિત થાય છે, અને 3.3-વોલ્ટ ડીસી પાવર. પાવર સપ્લાય કનેક્ટર પાસે છે 20-પિન, અને કનેક્ટર ફક્ત એક જ દિશામાં જઈ શકે છે.

હાર્ડ ડિસ્ક અને પાર્ટીશનો

પાર્ટીશન એ હાર્ડ ડિસ્કને વિભાજિત કરવાની પ્રક્રિયા છે ઘણા હિસ્સા, અને કોઈપણ એક ભાગનો ઉપયોગ કરે છે અથવા OS ઇન્સ્ટોલ કરવા અથવા બે અથવા વધુ પાર્ટીશનોનો ઉપયોગ કરવા માટે પાર્ટીશન બહુવિધ OS ઇન્સ્ટોલ કરો..

પરંતુ તે હંમેશા એક પાર્ટીશન ધરાવી શકે છે, અને તેનો ઉપયોગ કરી શકે છે એક જ OS ઇન્સ્ટોલ કરવા માટે સમગ્ર હાર્ડ ડિસ્ક જગ્યા, પરંતુ આ કરશે મોટી હાર્ડ ડિસ્કના ઉપયોગકર્તાઓ માટે ડેટા મેનેજમેન્ટ દુઃસ્વપ્ન બની જાય છે.

હવે, માસ્ટર બૂટ રેકોર્ડની રચનાને કારણે (MBR), માત્ર ચાર પાર્ટીશનો ધરાવે છે, અને આ ચાર પાર્ટીશનો પ્રાથમિક પાર્ટીશનો કહેવાય છે.

વિસ્તૃત પાર્ટીશન એ પોતે જ ઉપયોગી પાર્ટીશન નથી, પરંતુ તે “કન્ટેનર” જેવું છે અને તેનો ઉપયોગ પકડી રાખવા માટે થાય છે લોજિકલ ડ્રાઈવ્સ! એટલે કે આ વિસ્તૃત પાર્ટીશનને બહુવિધ લોજિકલ પાર્ટીશનોમાં પેટાવિભાજિત કરી શકાય છે.

પાર્ટીશનમાં બુટ કરવા માટે, તે નિયુક્ત થયેલ હોવું જ જોઈએ બૂટેબલ પાર્ટીશન અથવા એક્ટિવ પાર્ટીશન તરીકે. સક્રિય પાર્ટીશન તે પાર્ટીશન છે જે બુટ કરી શકાય તેવી અથવા જે તરીકે ફ્લેગ થયેલ છે OS સમાવે છે, આ સામાન્ય રીતે પ્રાથમિક પાર્ટીશન છે.

પાર્ટીશનોના પ્રકાર:

- માસ્ટર
- પાર્ટીશન
- વિસ્તૃત અને
- લોજિકલ વિસ્તૃત

માસ્ટર બૂટ રેકોર્ડ (MBR): MBR એ નાની 512 બાઈટ્સ છે પાર્ટીશન જે હાર્ડના પ્રથમ ભૌતિક ક્ષેત્રમાં છે ડિસ્ક સ્થાનને CHS 0,0,1 એટલે Oth તરીકે દર્શાવવામાં આવ્યું છે સિલિન્ડર, 0 મું હેડ અને 1 મું સેક્ટર.

MBR એક નાનો પ્રોગ્રામ ધરાવે છે જે બુટસ્ટ્રેપ તરીકે ઓળખાય છે પ્રોગ્રામ જે કોઈપણ OS માં બુટ કરવા માટે જવાબદાર છે. MBR માં પાર્ટીશન ટેબલ તરીકે ઓળખાતું કોષ્ટક પણ છે.

આ પાર્ટીશન કોષ્ટક એ કોષ્ટક છે જે ઉપલબ્ધની યાદી આપે છે હાર્ડ ડિસ્કમાં પ્રાથમિક પાર્ટીશનો. પાર્ટીશન ટેબલ સમગ્ર વિસ્તૃત પાર્ટીશનને એક પ્રાથમિક તરીકે ગણે છે પાર્ટીશન કરો અને તેને કોષ્ટકમાં સૂચિબદ્ધ કરો!

તેથી પાર્ટીશન કોષ્ટકમાં બે સંભવિત પ્રવેશો હોઈ શકે છે:-

- 4 પ્રાથમિક પાર્ટીશનો સુધી.
- 3 પ્રાથમિક પાર્ટીશનો સુધી અને 1 વિસ્તૃત પાર્ટીશન.(કુલ 4 થી વધુ નહીં).

પાર્ટીશન બુટ સેક્ટર (PBR): આ લોજિકલ ફર્સ્ટ છે સેક્ટર, કે જે પ્રાથમિક પાર્ટીશનની શરૂઆતમાં સેક્ટર છે. આ પણ 512 બાઈટ વિસ્તાર છે, જેમાં કેટલાક પ્રોગ્રામ્સ છે OS ફાઈલો શરૂ કરવા અથવા ચલાવવા માટે. તમામ પ્રાથમિક પાર્ટીશન પોતાની પીબીઆર ધરાવે છે.

વિસ્તૃત બુટ સેક્ટર (EBR): આ પ્રથમ તાર્કિક છે સેક્ટર, કે જે એક્સટેન્ડેડની શરૂઆતમાં સેક્ટર છે પાર્ટીશન. આ EBR પાર્ટીશન કોષ્ટક ધરાવે છે, જે યાદી આપે છે વિસ્તૃત પાર્ટીશનની અંદર ઉપલબ્ધ લોજિકલ પાર્ટીશનો. એટલે કે તે દરેક લોજિકલ પાર્ટીશનોના પ્રારંભિક સરનામાં સમાવે છે.

લોજિકલ એક્સટેન્ડેડ બુટ સેક્ટર (લેબર): આ છે લોજિકલ પ્રથમ સેક્ટર દરેક લોજિકલની શરૂઆતમાં રહે છે પાર્ટીશન. આ પ્રાથમિક પાર્ટીશનો માટે PBR જેવું જ છે.

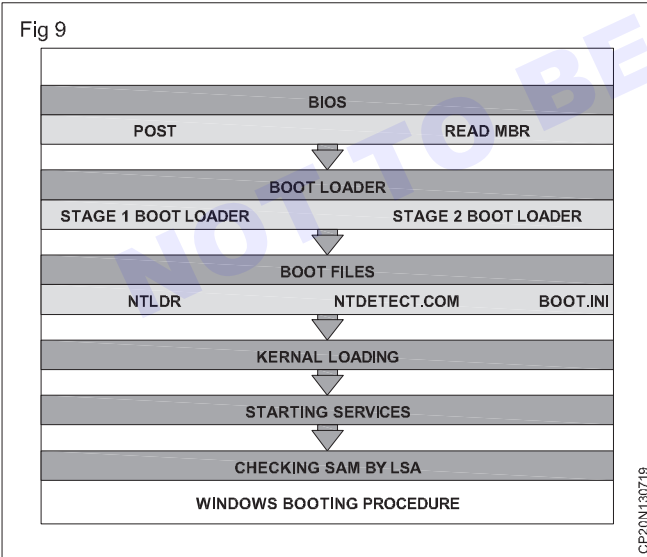
બુટીંગ

બુટીંગ એ ઓપરેટિંગ સિસ્ટમ લોડ કરવાની પ્રક્રિયા છે (OS) અને તમામ સિસ્ટમ સોફ્ટવેર અને હાર્ડવેર તપાસી રહ્યા છે જે કોમ્પ્યુટરમાં ઈન્સ્ટોલ કરેલ છે.

વિન્ડોઝ ઓપરેટિંગ સિસ્ટમની બુટીંગ પ્રક્રિયા

BIOS ના કાર્યો

પ્રથમ પ્રક્રિયા શરૂ થાય છે, જ્યારે કમ્પ્યુટર ચાલુ થાય છે બેઝિક ઈનપુટ આઉટપુટ સિસ્ટમ (BIOS) પોસ્ટ કરવા અને MBR વાંચવા માટે બે કાર્યો કરે છે (ફિગ 9).



POST - POST નો અર્થ પાવર ઓનસેલ્ફ ટેસ્ટ છે. પોસ્ટ કમ્પ્યુટર સાથે જોડાયેલા તમામ હાર્ડવેર ઉપકરણોને તપાસે છે જેમ કે RAM, હાર્ડ ડિસ્ક વગેરે અને ખાતરી કરો કે સિસ્ટમ તે હાર્ડવેર ઉપકરણો સાથે સરળતાથી ચાલી શકે છે. જો POST એ નિષ્ફળતા છે જે સિસ્ટમ બીપ અવાજ સાથે અટકે છે.

નવું BIOS બુટ પ્રાધાન્યતા તપાસે છે. અમે સીડી ડ્રાઈવ, હાર્ડ ડિસ્ક અથવા ફ્લોપી ડ્રાઈવ તરીકે બુટ પ્રાધાન્યતા સેટ કરી શકીએ છીએ.

MBR - BIOS ની આગળની ફરજ MBR વાંચવાની છે. MBR માસ્ટર બુટ રેકોર્ડ માટે વપરાય છે અને તે હાર્ડ ડિસ્ક પરનું પ્રથમ સેક્ટર છે. MBR પાર્ટીશન ટેબલ અને બુટલોડર સમાવે છે.

બુટ લોડરનું કાર્ય

હવે BIOS એ નિયંત્રણને બુટ લોડર પર પસાર કર્યું છે અને બુટ લોડર એ એક નાનો પ્રોગ્રામ છે જે કર્નલને કોમ્પ્યુટર મેમરીમાં લોડ કરે છે. વાસ્તવમાં બૂટના બે તબક્કા છે લોડરો, સ્ટેજ 1 બુટ લોડર અને સ્ટેજ 2 બુટ લોડર.

સ્ટેજ 1 બુટ લોડર એ સ્ટેજ 2 બુટ લોડરની લિંક છે. સ્ટેજ 2 બુટ લોડર બુટ પાર્ટીશનમાં રહે છે અને તે કર્નલને મેમરીમાં લોડ કરે છે.

બુટ ફાઈલો અને કાર્યો

Windows ઓપરેટિંગ સિસ્ટમમાં ત્રણ બુટ ફાઈલો છે અને તેઓ NTLDR, NTDETECT.COM અને Boot.ini છે. બુટ ફાઈલો હાર્ડડિસ્કના સક્રિય પાર્ટીશનમાં અને તેની સામાન્ય રીતે સી ડ્રાઈવમાં વિન્ડોઝ મશીનમાં જોવા મળે છે.

NTLDR - NTLDR નો અર્થ NT લોડર છે અને તે છે બીજા તબક્કાનું બુટ લોડર. NTLDR નો માર્ગ છે C:\Windows\i386\NTLDR.

Boot.ini - Boot.ini ની રૂપરેખાંકન ફાઈલો ધરાવે છે NTLDR. જ્યારે ઓપરેટિંગ સિસ્ટમ લોડ થાય છે ત્યારે અમે કરી શકતા નથી કર્નલ પર કોઈપણ દલીલો પસાર કરો, તેથી તે દલીલો છે boot.ini દ્વારા પસાર થયું. તમે boot.ini ખોલીને એડિટ કરી શકો છો નોટપેડ દ્વારા. Boot.ini નો પાથ C:\boot.ini છે.

NTDETECT.COM

આ ફાઈલ હાર્ડવેરને શોધી કાઢે છે અને માહિતી પસાર કરે છે NTLDR. એકત્રિત કરેલી માહિતીનો ઉપયોગ કરીને NTLDR હાર્ડવેર કી બનાવે છે અને આ કીનો ઉપયોગ હાર્ડવેરને શોધવા માટે થાય છે.

દરેક રીબુટ પછી નવી હાર્ડવેર કી જનરેટ થાય છે ઓપરેટિંગ સિસ્ટમ અને તેથી જ સિસ્ટમ રીબુટ કરવા માટે પૂછે છે નવા હાર્ડવેરની સ્થાપના પછી. હાર્ડવેર કીઓ NTLDR દ્વારા બનાવેલ વિન્ડોઝ રજિસ્ટ્રીમાં અહીં મળી શકે છે HKEY_LOCAL_MACHINE ' હાર્ડવેર.

BIOS સેટિંગ્સ અને તેમના ફેરફારો જુઓ (View the BIOS settings and their modifications)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

• વિન્ડોઝ વર્ઝન અને સુવિધાઓની યાદી બનાવો

વિન્ડોઝ વર્ઝન અને ફીચર્સ

વિન્ડોઝ વર્ઝન અને તેના ફીચર્સ માઈક્રોસોફ્ટ વિન્ડોઝ એ 1985 માં તેની પ્રથમ રજૂઆત પછી નવ મુખ્ય સંસ્કરણો જોયા છે. 29 વર્ષો પછી, વિન્ડોઝ ખૂબ જ અલગ દેખાય છે પરંતુ કોઈક રીતે તે તત્વોથી પરિચિત છે જે સમયની કસોટીમાં બચી ગયા છે, કમ્પ્યુટિંગ પાવરમાં વધારો થયો છે અને - તાજેતરમાં - કીબોર્ડ અને માઉસમાંથી ટચસ્ક્રીન પર પાળી.

અહીં વિન્ડોઝના ઇતિહાસ પર સંક્ષિપ્ત નજર છે, વિન્ડોઝ 1 સાથે બિલ ગેટ્સના હાથે તેના જન્મથી લઈને નવા માઈક્રોસોફ્ટના ચીફ એક્ઝિક્યુટિવ સત્ય નહેલાના નેતૃત્વમાં નવીનતમ આગમન સુધી.

વિન્ડોઝ 1**વિન્ડોઝનું પ્રથમ સંસ્કરણ આ તે**

છે જ્યાં તે બધું વિન્ડોઝ માટે શરૂ થયું હતું. મૂળ વિન્ડોઝ 1 નવેમ્બર 1985માં બહાર પાડવામાં આવ્યું હતું અને તે 16-બીટમાં ગ્રાફિકલ યુઝર ઇન્ટરફેસ પર માઈક્રોસોફ્ટનો પ્રથમ સાથો પ્રયાસ હતો.

વિકાસની આગેવાની માઈક્રોસોફ્ટના સ્થાપક બિલ ગેટ્સ દ્વારા કરવામાં આવી હતી અને MS-DOSની ટોચ પર ચાલે છે, જે કમાન્ડ-લાઈન ઇનપુટ પર આધાર રાખે છે.

તે નોંધપાત્ર હતું કારણ કે માઉસ સામાન્ય કમ્પ્યુટર ઇનપુટ ઉપકરણ હતું તે પહેલાં તે માઉસના ઉપયોગ પર ખૂબ આધાર રાખતું હતું. વપરાશકર્તાઓને આ વિચિત્ર ઇનપુટ સિસ્ટમથી પરિચિત થવામાં મદદ કરવા માટે, માઈક્રોસોફ્ટે એક ગેમ, રિવર્સી (સ્ક્રીનશૉટમાં દૃશ્યમાન)નો સમાવેશ કર્યો છે જે માઉસ નિયંત્રણ પર આધાર રાખે છે, કીબોર્ડ પર નહીં, લોકોને માઉસને આસપાસ ખસેડવાની અને સ્ક્રીન તત્વો પર ક્લિક કરવાની ટેવ પાડવા માટે.

વિન્ડોઝ 2**વિન્ડોઝ 2 ઓવરલેપિંગ વિન્ડો સાથે.**

વિન્ડોઝ 1 લીઝ પર આધ્યાના બે વર્ષ પછી, માઈક્રોસોફ્ટના વિન્ડોઝ 2 એ ડિસેમ્બર 1987માં તેને બદલી નાખ્યું. વિન્ડોઝ 2 માટે મોટી નવીનતા એ હતી કે વિન્ડો એકબીજાને ઓવરલેપ કરી શકે છે, અને તેણે “આઈકોનાઈઝિંગ” અથવા “” ને બદલે વિન્ડોઝને ન્યૂનતમ અથવા મહત્તમ કરવાની ક્ષમતા પણ રજૂ કરી. ઝૂમિંગ”

કંટ્રોલ પેનલ, જ્યાં વિવિધ સિસ્ટમ સેટિંગ્સ અને રૂપરેખાંકન વિકલ્પો એક જગ્યાએ એકસાથે એકત્રિત કરવામાં આવ્યા હતા, તે વિન્ડોઝ 2 માં રજૂ કરવામાં આવ્યું હતું અને તે આજ સુધી અસ્તિત્વમાં છે.

માઈક્રોસોફ્ટ વર્ડ અને એક્સેલ પણ વિન્ડોઝ 2 પર ચાલતા પ્રથમ વખત દેખાયા.

વિન્ડોઝ 3**વિન્ડોઝ 3.0 રંગીન બન્યું.**

પ્રથમ વિન્ડોઝ કે જેને 1990 માં શરૂ કરવામાં આવેલી હાર્ડ ડ્રાઈવની આવશ્યકતા હતી. વિન્ડોઝ 3 એ વધુ વ્યાપક સફળતા જોવાનું પ્રથમ

સંસ્કરણ હતું અને એપલના મેકિન્ટોશ અને કોમોડોર અમીગા ગ્રાફિકલ યુઝર ઇન્ટરફેસ માટે પડકારરૂપ માનવામાં આવે છે, જે પીસી-સુસંગત ઉત્પાદકોના કમ્પ્યુટર્સ પર પહેલાથી ઇન્સ્ટોલ કરેલું છે. ઝેનિથ ડેટા સિસ્ટમ્સ વિન્ડોઝ 3 એ વિન્ડોઝમાં MS DOS પ્રોગ્રામ્સ ચલાવવાની ક્ષમતા રજૂ કરી, જેણે લેગસી પ્રોગ્રામ્સમાં મલ્ટિટાસ્કિંગ લાવ્યા, અને ઇન્ટરફેસમાં વધુ આધુનિક, રંગીન દેખાવ લાવતા 256 રંગોને સપોર્ટ કર્યો. વધુ મહત્વપૂર્ણ - ઓછામાં ઓછા કુલ માનવ સમયનો વ્યય થયો - તેણે કાર્ડ-મૂવિંગ ટાઈમસિંક (અને માઉસ ઉપયોગ ટ્રેનર) સોલિડેર રજૂ કર્યું.

વિન્ડોઝ 3.1**Minesweeper સાથે Windows 3.1.**

વિન્ડોઝ 1 અને 2 બંનેમાં પોઈન્ટ રીલીઝ અપડેટ્સ હતા, પરંતુ 1992 માં રીલીઝ થયેલ વિન્ડોઝ 3.1 નોંધપાત્ર છે કારણ કે તેણે ટુટાઈપ ફોન્ટ્સ રજૂ કર્યા હતા જે વિન્ડોઝને પ્રથમ વખત એક સક્ષમ પ્રકાશન પ્લેટફોર્મ બનાવે છે.

માઈનસ્વીપરે પણ તેની પ્રથમ રજૂઆત કરી હતી. વિન્ડોઝ 3.1 ને ચલાવવા માટે 1MB ની RAM ની જરૂર છે અને સપોર્ટેડ MS-DOS પ્રોગ્રામ્સને પ્રથમ વખત માઉસ વડે નિયંત્રિત કરવાની મંજૂરી આપી હતી. વિન્ડોઝ 3.1 એ સીડી-રોમ પર વિતરિત થનારી પ્રથમ વિન્ડોઝ પણ હતી, જો કે એકવાર હાર્ડ ડ્રાઈવ પર ઇન્સ્ટોલ થઈ જાય તો તે માત્ર 10 થી 15 એમબી લેતી હતી (સીડી સામાન્ય રીતે 700 એમબી સુધી સ્ટોર કરી શકે છે)

વિન્ડોઝ 95**વિન્ડોઝ 95: ઓહ હેલો સ્ટાર્ટ મેનૂ.**

નામ પ્રમાણે, વિન્ડોઝ 95 ઓગસ્ટ 1995માં આવ્યું અને તેની સાથે પ્રથમ સ્ટાર્ટ બટન અને સ્ટાર્ટ લાવ્યા.

તેણે “પ્લગ એન્ડ પ્લે” નો ખ્યાલ પણ રજૂ કર્યો - પેરિફેરલને કનેક્ટ કરો અને ઓપરેટિંગ સિસ્ટમ તેના માટે યોગ્ય ડ્રાઈવરો શોધે છે અને તેને કાર્ય કરે છે. એવો વિચાર હતો; તે હંમેશા વ્યવહારમાં કામ કરતું નથી

વિન્ડોઝ 95 એ 32 બીટ પર્યાવરણ, ટાસ્કબાર પણ રજૂ કર્યું અને મલ્ટિટાસ્કિંગ પર ધ્યાન કેન્દ્રિત કર્યું. MS DOS એ હજુ પણ Windows 95 માટે મહત્વની ભૂમિકા ભજવી હતી, જેના માટે તેને કેટલાક પ્રોગ્રામ્સ અને તત્વો ચલાવવાની જરૂર હતી.

ઈન્ટરનેટ એક્સપ્લોરરે પણ વિન્ડોઝ 95 પર તેની શરૂઆત કરી હતી, પરંતુ વિન્ડોઝ 95 પ્લસની આવશ્યકતા માટે ડિફોલ્ટ રૂપે ઇન્સ્ટોલ કરવામાં આવ્યું ન હતું! પેક વિન્ડોઝ 95 ના પછીના સંશોધનોમાં ડિફોલ્ટ રૂપે IE નો સમાવેશ થાય છે, કારણ કે તે સમયે નેટસ્કેપ નેવિગેટર અને NCSCA મોઝેક લોકપ્રિય હતા.

વિન્ડોઝ 98**વિન્ડોઝ 98, છેલ્લું મહાન ડોસ આધારિત વિન્ડોઝ.**

જૂન 1998 માં રિલીઝ થયેલ, વિન્ડોઝ 98 વિન્ડોઝ 95 પર બનેલ અને તેની સાથે IE 4, આઉટલુક એક્સપ્રેસ, વિન્ડોઝ એડ્રેસ બુક, માઈક્રોસોફ્ટ ચેટ

અને નેટશો પ્લેયર લાવ્યું, જે 1999 માં વિન્ડોઝ 98 સેકન્ડ એડિશનમાં વિન્ડોઝ મીડિયા પ્લેયર 6.2 દ્વારા બદલવામાં આવ્યું.

વિન્ડોઝ 98 એ અન્ય વસ્તુઓની સાથે વિન્ડોઝ એક્સપ્લોરરમાં બેક અને ફોરવર્ડ નેવિગેશન બટનો અને એડ્રેસ બાર રજૂ કર્યા છે. સૌથી મોટા ફેરફારો પૈકી એક કમ્પ્યુટર ઘટકો અને એસેસરીઝ માટે વિન્ડોઝ ડ્રાઈવર મોડલની રજૂઆત હતી - વિન્ડોઝના તમામ ભાવિ સંસ્કરણોને સપોર્ટ કરવા માટે એક ડ્રાઈવર.

વિન્ડોઝ 98 માં યુએસબી સપોર્ટમાં ઘણો સુધારો થયો હતો અને યુએસબી હબ અને યુએસબી ઉંદર સહિત તેના વ્યાપક દત્તક લેવા તરફ દોરી ગયો

વિન્ડોઝ ME

વિન્ડોઝ ME એક છોડવા માટે હતું.

ઘણા લોકો દ્વારા વિન્ડોઝ શ્રેણીમાં નીચું બિંદુ માનવામાં આવે છે - ઓછામાં ઓછું, જ્યાં સુધી તેઓ વિન્ડોઝ વિસ્ટા જોતા ન હોય ત્યાં સુધી - વિન્ડોઝ મિલેનિયમ એડિશન MS-DOS પર આધારિત છેલ્લી વિન્ડોઝ હતી, અને Windows 9x લાઈનમાં છેલ્લી

સપ્ટેમ્બર 2000માં બહાર પાડવામાં આવેલ, તે એન્ટરપ્રાઇઝ માર્કેટને ધ્યાનમાં રાખીને વિન્ડોઝ 2000 સાથે જોડાયેલી ઉપભોક્તાલક્ષી ઓપરેટિંગ સિસ્ટમ હતી. તેણે વધુ સ્વચાલિત સિસ્ટમ પુનઃપ્રાપ્તિ સાધનો સહિત કેટલાક મહત્વપૂર્ણ ખ્યાલો ગ્રાહકોને રજૂ કર્યા.

IE 5.5, વિન્ડોઝ મીડિયા પ્લેયર 7 અને વિન્ડોઝ મૂવી મેકર બધાએ પ્રથમ વખત તેમનો દેખાવ કર્યો. વિન્ડોઝ એક્સપ્લોરરમાં પણ ઓટોકમ્પલીટ દેખાણું, પરંતુ ઓપરેટિંગ સિસ્ટમ બગડેલ હોવા, યોગ્ય રીતે ઈન્સ્ટોલ કરવામાં નિષ્ફળ અને સામાન્ય રીતે નબળી હોવા માટે કુખ્યાત હતી.

વિન્ડોઝ 2000

વિન્ડોઝ 2000 એ MEનું એન્ટરપ્રાઇઝ દ્વીન હતું.

ME નું એન્ટરપ્રાઇઝ દ્વીન, Windows 2000 ફેબ્રુઆરી 2000માં બહાર પાડવામાં આવ્યું હતું અને તે માઈક્રોસોફ્ટની બિઝનેસ ઓરિએન્ટેડ સિસ્ટમ Windows NT પર આધારિત હતું અને બાદમાં Windows XP માટે આધાર બન્યું હતું. વિન્ડોઝ 2000માં માઈક્રોસોફ્ટના સ્વચાલિત અપડેટ મહત્વની ભૂમિકા ભજવી હતી અને હાઈબરનેશનને સમર્થન આપનારી પ્રથમ વિન્ડોઝ બની હતી.

વિન્ડોઝ XP

વિન્ડોઝ XP આજે પણ ટકી છે.

વિન્ડોઝના શ્રેષ્ઠ સંસ્કરણોમાંનું એક, વિન્ડોઝ XP ઓક્ટોબર 2001માં બહાર પાડવામાં આવ્યું હતું અને માઈક્રોસોફ્ટની એન્ટરપ્રાઇઝ લાઈન અને ઓપરેટિંગ સિસ્ટમ્સની કન્ઝ્યુમર લાઈનને એક છત નીચે લાવી હતી.

જાહેરાત

તે વિન્ડોઝ 2000 જેવા વિન્ડોઝ એનટી પર આધારિત હતી, પરંતુ વિન્ડોઝ ME માંથી ગ્રાહક-મૈત્રીપૂર્ણ તત્વો લાવ્યા હતા. સ્ટાર્ટ મેનુ અને ટાસ્ક બારને વિઝ્યુઅલ ઓવરહોલ મળ્યું છે, જેમાં પરિચિત લીલા સ્ટાર્ટ બટન, બ્લુ ટાસ્ક બાર અને વિસ્ટા વૉલપેપર સાથે વિવિધ શેડો અને અન્ય વિઝ્યુઅલ ઈફેક્ટ્સ આવ્યા છે.

ClearType, જે LCD સ્ક્રીન પર લખાણને વાંચવા માટે સરળ બનાવવા માટે ડિઝાઇન કરવામાં આવ્યું હતું, તેને રજૂ કરવામાં આવ્યું હતું, જેમ કે બિલ્ટ-ઇન સીડી બર્નિંગ, સીડી અને અન્ય મીડિયામાંથી ઑટોપ્લે, ઉપરાંત વિવિધ સ્વચાલિત અપડેટ અને પુનઃપ્રાપ્તિ સાધનો, જે Windows MEથી વિપરીત ખરેખર કામ કરે છે.

રોમ સીડી અને અન્ય મીડિયા, ઉપરાંત વિવિધ સ્વચાલિત અપડેટ અને પુનઃપ્રાપ્તિ સાધનો, જે વિન્ડોઝ MEથી વિપરીત ખરેખર કામ કરે છે. વિન્ડોઝ XP એ સૌથી લાંબો સમય ચાલતી માઈક્રોસોફ્ટ ઓપરેટિંગ સિસ્ટમ હતી, જેમાં એપ્રિલ 2014 સુધી ત્રણ મોટા અપડેટ્સ અને સપોર્ટ જોવા મળ્યા હતા - તેની મૂળ રિલીઝ તારીખથી 13 વર્ષ. વિન્ડોઝ XP હજુ પણ અંદાજિત 430m PC પર ઉપયોગમાં લેવાતું હતું જ્યારે તેને બંધ કરવામાં આવ્યું હતું

તેની સૌથી મોટી સમસ્યા સુરક્ષા હતી: તેમાં ફાયરવોલ બિલ્ટ ઇન હોવા છતાં, તે ડિફોલ્ટ રૂપે બંધ હતી. વિન્ડોઝ XP ની વિશાળ લોકપ્રિયતા હેકરો અને ગુનેગારો માટે વરદાન રૂપ બની, જેમણે તેની ખામીઓનો ઉપયોગ કર્યો, ખાસ કરીને ઈન્ટરનેટ એક્સપ્લોરરમાં, નિર્દયતાથી બિલ ગેટ્સને “વિશ્વાસપાત્ર કમ્પ્યુટિંગ” પહેલ શરૂ કરવા અને ત્યારબાદ સર્વિસ પેક અપડેટ્સ જારી કરવા તરફ દોરી ગયા જેણે XP સામે કઠણ નોંધપાત્ર હુમલો કર્યું.

વિન્ડોઝ વિસ્ટા

વિન્ડોઝ વિસ્ટા, વિન્ડોઝ ME કરતાં દલીલપૂર્વક ખરાબ.

જાન્યુઆરી 2007માં વિન્ડોઝ વિસ્ટા દ્વારા બદલવામાં આવે તે પહેલા વિન્ડોઝ XP લગભગ છ વર્ષ સુધી અભ્યાસક્રમ રહ્યો. વિસ્ટાએ પારદર્શક તત્વો, શોધ અને સુરક્ષા પર વધુ ધ્યાન કેન્દ્રિત કરીને વિન્ડોઝના દેખાવને અપડેટ કર્યો. “લોંગહોર્ન” કોડનેમ હેઠળ તેનો વિકાસ મુશ્કેલીમાં મુકાયો હતો, તેને ઉત્પાદનમાં લાવવા માટે મહત્વાકાંક્ષી તત્વોને છોડી દેવામાં આવ્યા હતા.

જાહેરાત

તે બગડેલ હતી, “વપરાશકર્તા એકાઉન્ટ કંટ્રોલ” હેઠળ એપ્લિકેશન પરવાનગીઓ માટે સેંકડો વિનંતીઓ સાથે વપરાશકર્તા પર ભાર મૂક્યો - વિશ્વાસપાત્ર કમ્પ્યુટિંગ પહેલનું પરિણામ જેનો અર્થ એ થયો કે વપરાશકર્તાઓએ વિવિધ ફેરફારો કરવા માટેના કાર્યક્રમો દ્વારા પ્રયાસોને મંજૂરી અથવા નામંજૂર કરવી પડી હતી.

UAC ની સમસ્યા એ હતી કે તે આત્મસંતુષ્ટતા તરફ દોરી જાય છે, લોકો લગભગ કોઈપણ વસ્તુ માટે “હા” પર ક્લિક કરતા હતા - સુરક્ષાને UAC પહેલાની સ્થિતિમાં લઈ જતી હતી. તે “વિસ્ટા રેડી” તરીકે માનવામાં આવતું હોવા છતાં જૂના કમ્પ્યુટર્સ પર પણ તે ધીમેથી ચાલતું હતું - એક લેબલિંગ જેણે તેના પર દાવો કર્યો હતો કારણ કે વિસ્ટાના તમામ સંસ્કરણો તે લેબલવાળા પીસી પર ચાલી શકતા નથી.

વિસ્ટા દ્વારા માઈક્રોસોફ્ટની ડાયરેક્ટએક્સ 10 ટેકનોલોજીના સમાવેશથી P C ગેમર્સને પ્રોત્સાહન મળ્યું

વિન્ડોઝ મીડિયા \ પ્લેયર 11 અને IE 7 એ વિન્ડોઝ ડિફેન્ડર સાથે એન્ટી સ્પાયવેર પ્રોગ્રામની શરૂઆત કરી. વિસ્ટામાં સ્પીચ રેકગ્નિશન, વિન્ડોઝ ડીવીડી મેકર અને ફોટો ગેલેરીનો પણ સમાવેશ થાય છે, તેમજ ડીવીડી પર વિતરિત થનારી પ્રથમ વિન્ડોઝ છે. બાદમાં વિન્ડોઝ મીડિયા પ્લેયર વગરનું વિન્ડોઝ વિસ્ટાનું વર્ઝન એન્ટી-ટ્રૂસ્ટ તપાસના જવાબમાં બનાવવામાં આવ્યું હતું.

વિન્ડોઝ 7

વિન્ડોઝ 7 એ બધું હતું જે વિન્ડોઝ વિસ્ટા હોવું જોઈએ.

વિન્ડોઝ વિસ્ટા કેવું હોવું જોઈએ તે અંગે ઘણા લોકો દ્વારા વિચારણા કરવામાં આવે છે, વિન્ડોઝ 7 પ્રથમ ઓક્ટોબર 2009માં બહાર પાડવામાં આવ્યું હતું. તેનો હેતુ વિસ્ટા દ્વારા સામનો કરવામાં આવતી તમામ સમસ્યાઓ અને ટીકાઓને ઠીક કરવાનો હતો, તેના દેખાવમાં થોડો ફેરફાર અને વપરાશકર્તા-

મેત્રીપૂર્ણ સુવિધાઓ પર એકાગ્રતા અને ઓછા “સંવાદ બોક્સ ઓવરલોડ” તે ઝડપી, વધુ સ્થિર અને ઉપયોગમાં સરળ હતું, જે ઓપરેટિંગ સિસ્ટમ બની હતી જે મોટાભાગના વપરાશકર્તાઓ અને વ્યવસાય વિસ્તારને સંપૂર્ણપણે છોડીને Windows XP થી અપગ્રેડ કરશે.

હસ્તાક્ષર ઓળખ 7 માં ડેબ્યૂ કરવામાં આવી હતી, જેમ કે વિન્ડોઝને સ્ક્રીનની ટોચ અથવા બાજુઓ પર “સ્નેપ” કરવાની ક્ષમતા, ઝડપી વધુ સ્વચાલિત વિન્ડો માપ બદલવાની મંજૂરી આપે છે.

વિન્ડોઝ 7 એ યુરોપમાં IE ના પ્રી-ઇન્સ્ટોલિંગ પર અવિશ્વાસની તપાસ સાથે માઈક્રોસોફ્ટને ફટકો માર્યો હતો, જેના કારણે નવા વપરાશકર્તાઓને બ્રાઉઝર બેલેટ સ્ક્રીન બતાવવામાં આવી હતી જે તેમને પ્રથમ બુટ પર કયા બ્રાઉઝરને ઇન્સ્ટોલ કરવું તે પસંદ કરવાની મંજૂરી આપે છે.

વિન્ડોઝ 8

વિન્ડોઝ 8 કીબોર્ડ અને માઉસ કરતાં ટચ પર વધુ ધ્યાન કેન્દ્રિત કરે છે.

ઓક્ટોબર 2012 માં રિલીઝ થયેલ, Windows 8 એ વધુ ટચ ફ્રેન્ડલી સ્ટાર્ટ સ્ક્રીનની તરફેણમાં સ્ટાર્ટ બટન અને સ્ટાર્ટ મેનૂને છોડીને વિન્ડોઝ ઇન્ટરફેસનું માઈક્રોસોફ્ટનું સૌથી આમૂલ પરિવર્તન હતું.

જાહેરાત

નવા ટાઈલ્ડ ઇન્ટરફેસમાં પ્રોગ્રામ આઈકોન્સ અને લાઈવ ટાઈલ્સ જોવા મળ્યા, જે સામાન્ય રીતે “વિજેટ્સ” સાથે સંકળાયેલી એક નજરમાં માહિતી પ્રદર્શિત કરે છે, જે પ્રોગ્રામ્સ અને આઈકોનની યાદીને બદલે છે. ડેસ્કટોપ હજુ પણ સમાવિષ્ટ હતું, જે વિન્ડોઝ 7 જેવું હતું. વિન્ડોઝ 8 વિન્ડોઝના અગાઉના વર્ઝન કરતાં ઝડપી હતું અને તેમાં નવા, વધુ ઝડપી USB 3.0 ઉપકરણો માટે સપોર્ટનો સમાવેશ થતો હતો.

વિન્ડોઝ સ્ટોર, જે યુનિવર્સલ વિન્ડોઝ એપ્સ ઓફર કરે છે જે ફક્ત પૂર્ણ-સ્ક્રીન મોડમાં ચાલે છે, તે રજૂ કરવામાં આવ્યું હતું. વિન્ડોઝના અન્ય પુનરાવૃત્તિઓની જેમ તૃતીય-પક્ષી તરફથી પ્રોગ્રામ્સ હજી પણ ઇન્સ્ટોલ થઈ શકે છે, પરંતુ તેઓ ફક્ત વિન્ડોઝના પરંપરાગત ડેસ્કટોપ ઇન્ટરફેસને એક્સેસ કરી શકે છે.

આમૂલ ઓવરઓલને ઘણા લોકો દ્વારા આવકારવામાં આવ્યો ન હતો. માઈક્રોસોફ્ટે ટચસ્ક્રીન સપોર્ટ અને ડેસ્કટોપ યુઝર્સ વચ્ચે સરસ લાઈન પર ચાલવાનો પ્રયાસ કર્યો, પરંતુ આખરે ડેસ્કટોપ યુઝર્સ જેઓ વિન્ડોઝને પરંપરાગત માઉસ અને કીબોર્ડ વડે નિયંત્રિત કરવા માગે છે અને ટચસ્ક્રીનને નહીં માને છે કે Windows 8 એક પગલું પાછળ છે.

તેના ટચ ઓરિએન્ટેડ ઇન્ટરફેસને ઉપયોગી અથવા તો જરૂરી બનાવવા માટે ઘણી ઓછી ટચસ્ક્રીનનો ઉપયોગ કરવામાં આવ્યો હતો અથવા ઓફર કરવામાં આવી હતી - આઈપેડ અને સ્માર્ટફોન જેવા ટેબ્લેટ્સનો સમાવેશ વધારો થયો હોવા છતાં, જેણે 2010 ના અંત સુધીમાં પીસીનું વેચાણ શરૂ કર્યું હતું.

વિન્ડોઝ આરટી, જે પરંપરાગત રીતે સ્માર્ટફોન અને નોન-પીસી ટેબ્લેટમાં જોવા મળતા એઆરએમ-આધારિત પ્રોસેસર્સ પર ચાલે છે, તે જ સમયે માઈક્રોસોફ્ટ સરફેસ ટેબ્લેટ સાથે વિન્ડોઝ 8 તરીકે રજૂ કરવામાં આવ્યું હતું.

તે જોયું અને લાગ્યું વિન્ડોઝ 8 ની જેમ, પરંતુ પરંપરાગત વિન્ડોઝ એપ્લિકેશન ચલાવી શકતી નથી, તેના બદલે તૃતીય-પક્ષ એપ્લિકેશનો માટે ફક્ત વિન્ડોઝ સ્ટોર પર આધાર રાખે છે.

વિન્ડોઝ 8.1

વિન્ડોઝ 8.1 અને સ્ટાર્ટ બટનનું પુનઃપ્રાપ્તિ: ઓક્ટોબર 2013માં વિન્ડોઝ 8 માટે ફ્રી પોઈન્ટ રીલીઝ, વિન્ડોઝ 8.1 એ Microsoft તરફથી વાર્ષિક સોફ્ટવેર અપડેટ્સ તરફ પાળીને ચિહ્નિત કર્યું અને તેના નવા વિઝ્યુઅલ ઇન્ટરફેસની આસપાસ માઈક્રોસોફ્ટના યુ-ટર્નમાં પ્રથમ પગલું શામેલ કર્યું.

વિન્ડોઝ 8.1 એ ફરીથી સ્ટાર્ટ બટન રજૂ કર્યું, જે વિન્ડોઝ 8.1 ના ડેસ્કટોપ વ્યુમાંથી સ્ટાર્ટ સ્ક્રીનને લાવ્યું. વપરાશકર્તાઓ વિન્ડોઝ 8.1 ના ડેસ્કટોપમાં સીધું જ બુટ કરવાનું પણ પસંદ કરી શકે છે, જે ટચ-ફ્રીક્સ સ્ટાર્ટ સ્ક્રીન કરતાં માઉસ અને કીબોર્ડ સાથે ડેસ્કટોપ કમ્યુટરનો ઉપયોગ કરતા લોકો માટે વધુ યોગ્ય હતું.

વિન્ડોઝ 10

વિન્ડોઝ 10 સાથે, માઈક્રોસોફ્ટ વિન્ડોઝ 8 માટે બનાવેલી કેટલીક ટચ અને ટેબ્લેટ સુવિધાઓ રાખવાનો પ્રયાસ કરી રહ્યું છે, તેને પરિચિત સ્ટાર્ટ મેનૂ અને ડેસ્કટોપ સાથે જોડે છે, અને તે બધું વધુ સુરક્ષા સાથે સુધારેલ ઓપરેટિંગ સિસ્ટમની ટોચ પર ચલાવે છે, એક નવું બ્રાઉઝર , Cortana સહાયક, સફરમાં સંપાદન માટે ઓફિસનું તેનું પોતાનું સંસ્કરણ અને જીવનને સરળ બનાવવાના હેતુથી પુષ્કળ નવી સુવિધાઓ.

અલબત્ત, તેનો અર્થ એ પણ છે કે તે વાપરવા માટે ખૂબ જ અલગ છે, પછી ભલે તમે Windows 7, Windows 8 અથવા Windows XPમાંથી આવો. તમારે તમારા પીસીને બંધ કરવા માટે પણ નવી જગ્યાએ જોવું પડશે.

તેના ઉપર, Windows 10 એ માત્ર એક PC ઓપરેટિંગ સિસ્ટમ કરતાં વધુ છે; તે વિન્ડોઝ ફોન્સ પર પણ ચાલશે - અને નાના ટેબ્લેટ પર પણ, કારણ કે 6-ઇંચનો ફોન અને 7- ઇંચ ટેબ્લેટ આવા બહુ અલગ ઉપકરણો નથી.

Microsoft અપેક્ષા રાખે છે કે લોકો એક અબજ ઉપકરણો પર Windows 10 મૂકે (જે વધુ એપ્લિકેશન વિકાસકર્તાઓને ઓછામાં ઓછા Windows ફોન અને ટેબ્લેટ તેમજ Xbox One અને HoloLens માટે તેમની એપ્લિકેશનો બનાવવા માટે પ્રોત્સાહિત કરવા જોઈએ).

સ્ટાર્ટ મેનૂ વિકસિત

થાય છે વિન્ડોઝ 8 ની પૂર્ણ સ્ક્રીન સ્ટાર્ટ સ્ક્રીન વિન્ડોઝ 10 માં સ્ટાર્ટ મેનૂ તરીકે પાછી આવી છે જે બંને વિકલ્પોમાંથી શ્રેષ્ઠને જોડવાનો પ્રયાસ કરે છે. એક સ્ક્રોલિંગ સ્ટાર્ટ મેનૂ જે સિંગલ કોલમ સુધી મર્યાદિત છે, વધારાના વિકલ્પો માટે જમ્પ લિસ્ટ અને ફ્લાયઆઉટ મેનૂ સાથે, વારંવાર ઉપયોગમાં લેવાતા અને તાજેતરમાં ઇન્સ્ટોલ કરેલા પ્રોગ્રામમાં વિભાજિત, તમામ એપ્લિકેશનોના સ્ક્રોલિંગ વ્યૂ પર સ્વિચ કરવાના વિકલ્પ સાથે, મૂળાક્ષરો પ્રમાણે સોર્ટ કરેલ છે.

પણ એક વધારાની તકતી પણ મેળવો, જ્યાં વિન્ડોઝ 8-શૈલીની ટાઈલ્સ પિન કરી શકાય, લાઈવ ટાઈલ્સના ‘રોટેટિંગ 3D ક્યુબ’ એનિમેશન સાથે પૂર્ણ થાય. સ્ટાર્ટ મેનૂને મોટા કદમાં ખેંચો અથવા તેને પૂર્ણ સ્ક્રીન પર સેટ કરો.

ડેસ્કટોપ બેકગ્રાઉન્ડ

ડેસ્કટોપનો બીજો ઘટક બેકગ્રાઉન્ડ છે. આ ફક્ત એક છબી છે જે સ્ક્રીનની પાછળ દેખાય છે. મોટાભાગના કમ્યુટર્સ પહેલાથી પસંદ કરેલ પૃષ્ઠભૂમિ સાથે આવે છે, પરંતુ તેને કોઈપણ છબી પર બદલો.

Fig 1



પૃષ્ઠભૂમિ બદલવા માટે, આ પગલાં અનુસરો:

- 1 પૃષ્ઠભૂમિ પર જમણું-ક્લિક કરો અને “વ્યક્તિગત કરો” પસંદ કરો
- 2 વૈયક્તિકરણ વિંડોમાંથી, પૂર્વ-પસંદ કરેલ ચિત્રોની શ્રેણીમાંથી પસંદ કરો અથવા તમારા પોતાના માટે બ્રાઉઝ કરો.

ચિત્ર પસંદ કર્યા પછી, પૃષ્ઠભૂમિ આપમેળે બદલાઈ જશે.

સ્ટાર્ટ

મેનૂ જો કોઈ ચોક્કસ એપ્લિકેશન શોધી રહ્યાં હોય, તો સ્ટાર્ટ મેનૂ ખોલો અને “બધી એપ્લિકેશન” પર ક્લિક કરો. આ કમ્પ્યુટર પર ઇન્સ્ટોલ કરેલી બધી એપ્લિકેશનોની મૂળાક્ષરોની સૂચિ ખોલશે.

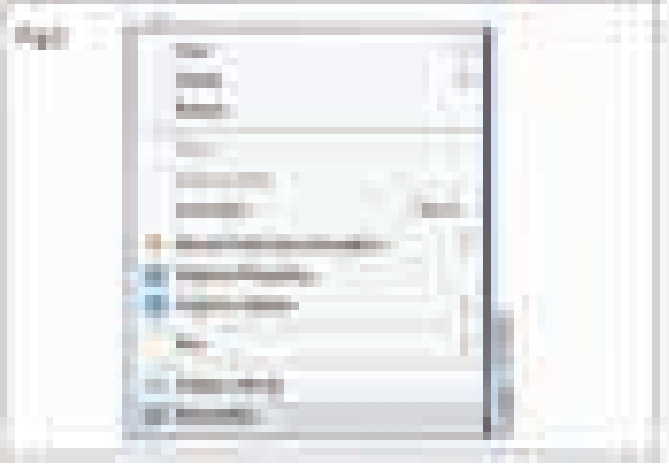
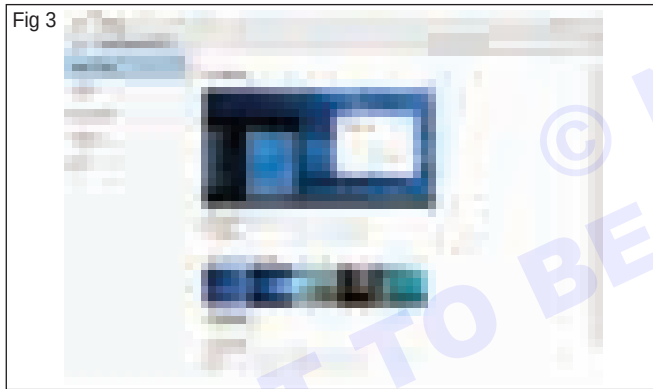


Fig 3



ફાઈલ એક્સપ્લોરર

જો તમે કોઈ ચોક્કસ દસ્તાવેજ જોઈ રહ્યા હો, તો બીજો વિકલ્પ ટાસ્કબાર પરના ફોલ્ડર આઈકોન પર ક્લિક કરીને ફાઈલ એક્સપ્લોરરનો ઉપયોગ કરવાનો છે.

ફાઈલ એક્સપ્લોરર વિંડોમાં, બધા ફોલ્ડર્સ અને દસ્તાવેજો બ્રાઉઝ કરો.

વર્ચ્યુઅલ ડેસ્કટોપ્સ

વિન્ડોઝ 10 ની નવી વિશેષતાઓમાંની એક વર્ચ્યુઅલ ડેસ્કટોપ્સનો ઉમેરો છે. આ તમને બહુવિધ ડેસ્કટોપ સ્કીનો રાખવા માટે પરવાનગી આપે છે જ્યાં ખુલ્લી વિન્ડો ગોઠવવી.

વર્ચ્યુઅલ ડેસ્કટોપ ઉમેરવા માટે, આ પગલાં અનુસરો:

- 1 ટાસ્કબાર પર ટાસ્ક વ્યૂ પર ક્લિક કરો
- 2 નીચેના જમણા ખૂણે “નવું ડેસ્કટોપ” વિકલ્પ પર ક્લિક કરો.

તમે ટાસ્ક વ્યૂ પર ફરીથી ક્લિક કરીને નવા ડેસ્કટોપને એક્સેસ કરી શકો છો અથવા કાઢી નાખી શકો છો.

Fig 4

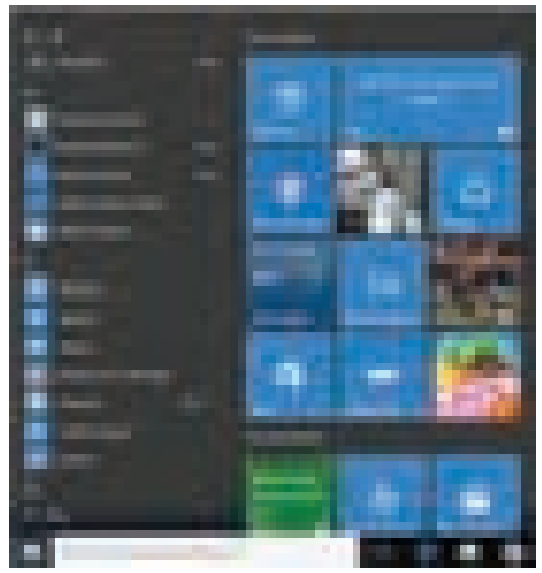


Fig 5

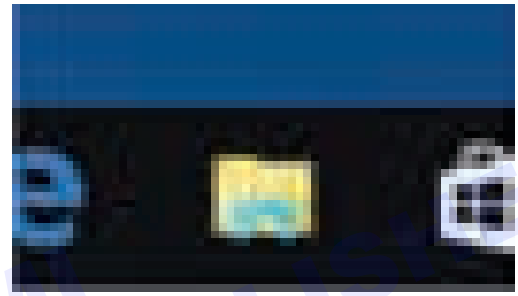
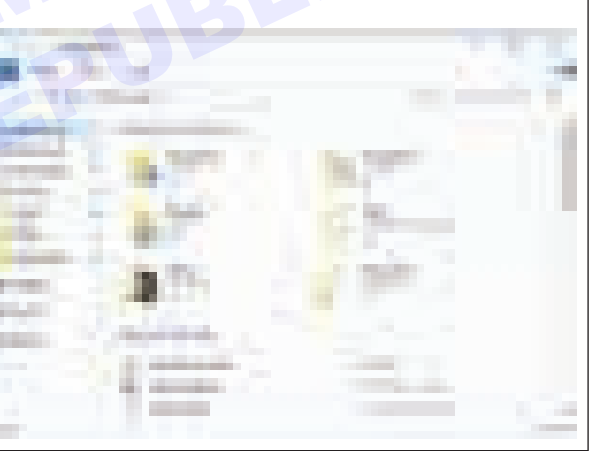


Fig 6



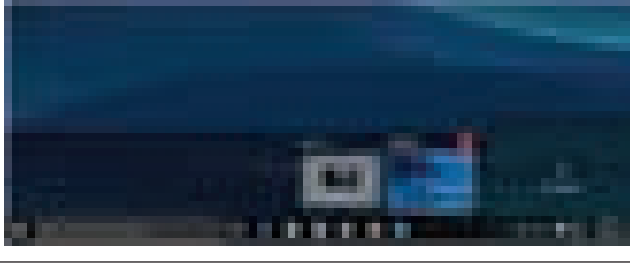
Cortana શોધ અને નિયંત્રણમાં મદદ કરે છે Cortana, Windows Phone Assistant, Windows 10 માં ટાસ્કબાર પર સર્ચ પેન તરીકે દેખાય છે, જે ‘Hey Cortana’ કહીને પણ ટ્રિગર થઈ શકે છે - અને જ્યારે સ્ટાર્ટ મેનૂ શોધવાનું શરૂ કરો.

Fig 7



તેનાથી એપ્લિકેશન્સ ઈન્સ્ટોલ થઈ ગઈ છે, દસ્તાવેજોની ઍક્સેસ, સ્ટોરમાંથી એપ્લિકેશન્સ ઈન્સ્ટોલ થઈ શકે છે, વેબ પરથી શોધ પરિણામો અને Cortana સાથે સંકલિત એપ્લિકેશન્સ અને સેવાઓ સહિત અન્ય માહિતીની શ્રેણી મળે છે.

Fig 8



Cortana સક્રિય કરી રહ્યું છે

જો Cortana સક્રિય ન હોય, તો Cortana સેટિંગ્સને ઍક્સેસ કરવા માટે ટાસ્કબારમાં “Cortana” લખીને અથવા ફક્ત મેનુની ડાબી બાજુએ “Gear” આયકન પર ક્લિક કરીને તેને ચાલુ કરી શકો છો.

Fig 9



Cortana સક્રિય કર્યા પછી, તે અનુભવને વ્યક્તિગત કરવા વિશે માહિતી એકત્ર કરવાનું શરૂ કરશે.

કાર્ય સ્વિચર

મોટાભાગના વિન્ડોઝ વપરાશકર્તાઓને બધી ચાલી રહેલ એપ્લિકેશનો જોવા અને તેમની વચ્ચે સ્વિચ કરવા માટે Alt-Tab કીબોર્ડ સંયોજન ખબર નથી, તેથી મોટા થંબનેલ્સ સાથે ફરીથી ડિઝાઇન કરેલ ટાસ્ક સ્વિચર હોવા ઉપરાંત, Windows 10 તેમને શોધવામાં મદદ કરવા માટે ટાસ્કબારમાં ટાસ્ક વ્યૂ આઈકોન પણ મૂકે છે. તે

Fig 10

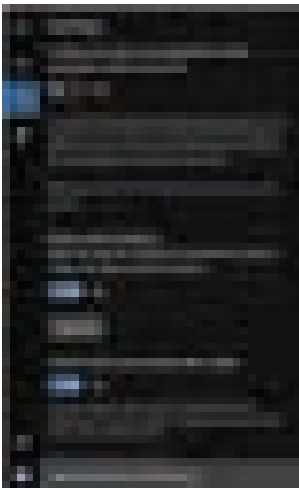
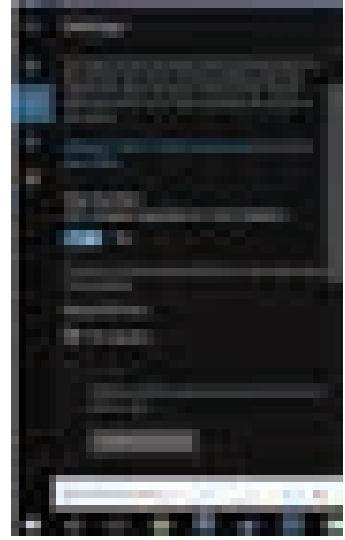


Fig 11



ટાસ્કબાર: વિન્ડોઝ 10 ટાસ્કબાર સ્ક્રીનના તળિયે બેસે છે જે વપરાશકર્તાને સ્ટાર્ટ મેનુની ઍક્સેસ આપે છે, તેમજ વારંવાર ઉપયોગમાં લેવાતી એપ્લિકેશનના ચિહ્નો. જમણી બાજુએ, ટાસ્કબારમાં નોટિફિકેશન એરિયા છે જે યુઝરને ઈન્ટરનેટ કનેક્શનની સ્થિતિ અથવા લેપટોપ બેટરીના ચાર્જ જેવી વિવિધ બાબતોની જાણ કરે છે.

ટાસ્કબારની મધ્યમાં આવેલા ચિહ્નો “પિન કરેલ” એપ્લિકેશનો છે, જે તમે વારંવાર ઉપયોગ કરો છો તે એપ્લિકેશન્સની ઝડપી ઍક્સેસ મેળવવાનો એક માર્ગ છે. જ્યાં સુધી તમે તેમને “અનપિન” નહીં કરો ત્યાં સુધી “પિન કરેલ” એપ્લિકેશનો ટાસ્કબારમાં રહેશે.

Fig 12



પગલું 1: તમે સ્ટાર્ટ મેનુમાં જે એપ્લિકેશનને પિન કરવા માંગો છો તે શોધો.

પગલું 2: એપ્લિકેશન પર જમણું-ક્લિક કરો.

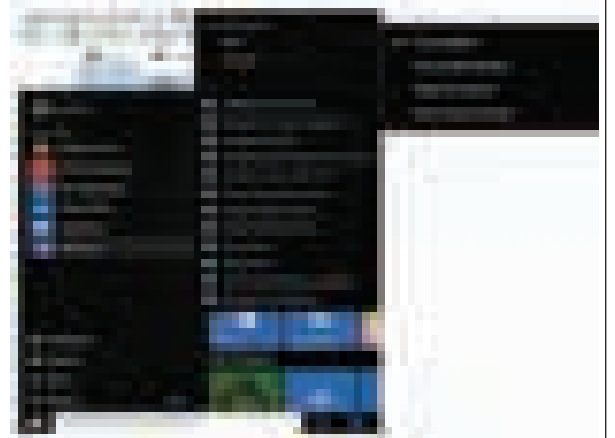
પગલું 3: મેનુની ટોચ પર “વધુ” વિકલ્પ પસંદ કરો.

પગલું 4: “ટાસ્કબારમાં પિન કરો” વિકલ્પ પસંદ કરો.

ટાસ્કબારમાંથી એપ્લિકેશનને અનપિન કરો

તેને “અનપિન” કરવા માટે, ફક્ત ટાસ્કબારમાંના ચિહ્ન પર જમણું-ક્લિક કરો અને “ટાસ્કબારમાંથી અનપિન કરો” પસંદ કરો. તમે ઈચ્છો ત્યારે તેને ફરીથી “પિન” કરી શકો છો.

Fig 13

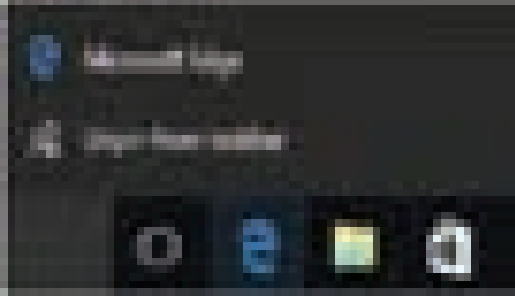


નોટિફિકેશન એરિયા

નોટિફિકેશન એરિયા ટાસ્કબારની જમણી બાજુએ આવેલું છે. તે તમારા કમ્પ્યુટરથી વિવિધ પ્રકારની સૂચનાઓ બતાવે છે જેમ કે તમારું ઇન્ટરનેટ કનેક્શન અથવા વોલ્યુમ સ્તર

શરૂઆતમાં, સૂચના ક્ષેત્ર મર્યાદિત સંખ્યામાં ચિહ્નો દર્શાવે છે. પરંતુ તમે અન્ય ચિહ્નો જોવા માટે તેની ડાબી બાજુએ ઉપર તરફના તીરને ક્લિક કરી શકો છો

Fig 14



સ્નેપ આસિસ્ટ

કારણ કે તમામ એપ્સ અને પ્રોગ્રામ્સ ડેસ્કટોપ પરની વિન્ડોમાં ચાલે છે, સ્ટોરમાંથી આધુનિક એપ્સ તેમની પોતાની જગ્યામાં હોવાને બદલે, અને બીજી એપ્સને સ્ક્રીન પર લાવવા અને સ્વિલટ વ્યૂ મેળવવા માટે સ્ક્રીનની ડાબી કિનારે ખેંચી શકાતી નથી. તેના બદલે, પરિચિત સ્નેપ દૃશ્ય મેળવવા માટે વિન્ડોને સ્ક્રીનના ખૂણામાં ખેંચો.

Fig 15



હવે સ્ક્રીનના ચારેય ખૂણાઓનો ઉપયોગ કરો જો દરેક વિન્ડો અડધાને બદલે સ્ક્રીનનો ચોથા ભાગ લે, અને જે જગ્યા તમે હમણાં ખેંચી છે તે વિન્ડો દ્વારા ભરવામાં આવતી નથી તે તમારી અન્ય વિન્ડોની થંબનેલ્સને સ્નેપ કરવાનું સરળ બનાવવા માટે બતાવે છે. આગામી એક જગ્યાએ.

એક્શન સેન્ટર

જો આપણે વિન્ડોઝ ફોન 8.1 (અથવા એન્ડ્રોઇડ અને/અથવા iOS) નો ઉપયોગ કર્યો હોય, તો અમે નોટિફિકેશન સેન્ટરનો ઉપયોગ કરતા હોઈએ છીએ તે સ્ક્રીનની ટોચ પરથી નીચે ખેંચી શકીએ છીએ.

વિન્ડોઝ 10 તેને સ્ક્રીનની જમણી બાજુએ મૂકે છે, જ્યાં વિન્ડોઝ 8 માં ચાર્મ્સ બાર હતું, જેમાં ટોચ પર વિવિધ એપ્સની સૂચનાઓ અને ઝડપી એક્સેસ માટે તળિયે વિવિધ સેટિંગ્સ બટનોની પસંદગી છે.

આદેશ પ્રોમ્ટ

આપણામાંના જેઓ કમાન્ડ પ્રોમ્ટનો ઉપયોગ કરે છે તેઓ 1990 ના દાયકાથી લગભગ સમાન અનુભવ સાથે અટકી ગયા છે, પરંતુ Windows 10 અંતે કમાન્ડ પ્રોમ્ટ વિન્ડોનું કદ બદલી શકે છે અને કમાન્ડ પ્રોમ્ટ પર કૉપી અને પેસ્ટ કરવા માટે પરિચિત કીબોર્ડ શોર્ટકટ્સનો ઉપયોગ કરી શકે છે.

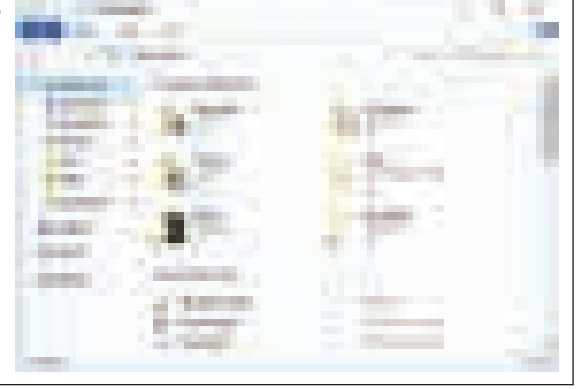
તે ગ્રાઉન્ડ બ્રેકિંગથી દૂર છે પરંતુ વર્ષોની હતાશા પછી તે ખૂબ જ આવકારદાયક સુધારો છે.

ફાઇલ એક્સપ્લોરર

ફાઇલ એક્સપ્લોરર એ વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ દ્વારા ફોલ્ડર્સ અને ફાઇલોને બ્રાઉઝ કરવા માટે ઉપયોગમાં લેવાતી ફાઇલ મેનેજમેન્ટ એપ્લિકેશન છે. તે વપરાશકર્તાને કોમ્પ્યુટરમાં સંગ્રહિત ફાઇલોને નેવિગેટ કરવા અને એક્સેસ કરવા માટે ગ્રાફિકલ ઇન્ટરફેસ પ્રદાન કરે છે.

ફાઇલ એક્સપ્લોરરને એક્સેસ કરવાની મુખ્ય રીત ટાસ્કબારમાં ફોલ્ડર આઇકોન પર ક્લિક કરીને છે. આઇકોન પર ક્લિક કર્યા પછી, ફાઇલ એક્સપ્લોરર વિન્ડો ખુલશે.

Fig 16



પ્રારંભિક ફાઇલ એક્સપ્લોરર વિન્ડોમાં નીચેના વિભાગોનો સમાવેશ થાય છે:

- 1 ફાઇલ એક્સપ્લોરર રિબન, જે Microsoft Office માં દર્શાવવામાં આવેલ રિબન જેવું લાગે છે. રિબન તમારી ફાઇલો અને ફોલ્ડર્સ સાથે કરવા માટેના સામાન્ય કાર્યો માટેના બટનો ધરાવે છે.
- 2 નેવિગેશન પેન તમને તમારી દસ્તાવેજો અને ચિત્રોની લાઇબ્રેરીઓ તેમજ તમારા સ્ટોરેજ ઉપકરણોની એક્સેસ આપે છે. તે વારંવાર ઉપયોગમાં લેવાતા ફોલ્ડર્સ અને નેટવર્ક ઉપકરણોને પણ દર્શાવે છે.
- 3 જમણી બાજુના ફિક્ચર્ડ ફોલ્ડર્સ વિભાગમાં તમે તાજેતરમાં જેની સાથે કામ કર્યું છે તે ફોલ્ડર્સને ઝડપથી એક્સેસ કરવાની મંજૂરી આપે છે.
- 4 વિન્ડોની નીચેના ભાગમાં તાજેતરના ફાઇલો વિભાગમાં તમે તાજેતરમાં ખોલેલ ફાઇલો અને દસ્તાવેજો દર્શાવ્યા છે

નવું એજ બ્રાઉઝર

ક્રોમ અને ફાયરફોક્સ જેવા ફાસ્ટ-મૂવિંગ બ્રાઉઝર્સને પકડવા માટે, માઇક્રોસોફ્ટે તેના બ્રાઉઝરને પાછું બેઝિક્સ પર લઈ લીધું, વેબ ધોરણો સાથે બંધબેસતા ન હોય તેવા વર્ષોના કોડને ફાડી નાખ્યા અને એક દુર્બળ, ઝડપી બ્રાઉઝર બનાવ્યું.

તે પ્રગતિમાં છે - વિન્ડોઝ 10 લૉન્ચ થયા પછી થોડા સમય સુધી તે એડ-બ્લોકિંગ એક્સ્ટેન્શન જેવી વસ્તુઓ માટે સમર્થન મેળવશે નહીં - પરંતુ અહીં ઘણી બધી સુધારા વસ્તુઓ કરી શકે છે. ઉદાહરણ તરીકે, અને મિત્રને મોકલવા માટે વેબ પેજ પર નોંધો લખી શકે છે અને એજ પાસે રેસ્ટોરન્ટનો ફોન નંબર અથવા શરૂઆતના કલાકો જેવી ઉપયોગી માહિતી વેબ પૃષ્ઠોમાંથી બહાર કાઢવા માટે Cortana બિલ્ટ ઇન છે.

માધ્યમ જેવી સાઈટ્સ કે જે IE સાથે યોગ્ય રીતે કામ કરતી નથી તે વધુ સારી દેખાવી જોઈએ અને એજમાં વધુ સુવિધાઓ હોવી જોઈએ

બહુવિધ ડેસ્કટોપ

વપરાશકર્તાને ઘણી બધી વિન્ડો ગોઠવવાની જરૂર છે અને તેની પાસે બહુવિધ મોનિટર નથી, વપરાશકર્તા તેને બહુવિધ વર્ચ્યુઅલ ડેસ્કટોપ પર મૂકી શકે છે. અને એપ વચ્ચે હંમેશાની જેમ ખસેડવા માટે Alt-Tab નો ઉપયોગ કરી શકો છો અને પછી Windows-Ctrl અને ડેસ્કટોપ વચ્ચે ખસેડવા માટે ડાબી અને જમણી એરો કીનો ઉપયોગ કરી શકો છો.

શેડ્યૂલ પુનઃપ્રારંભ થાય છે

વિન્ડોઝની જાહેરાત કર્યા પછી હવે તમારી પાસે અપડેટ લાગુ કરવા માટે પુનઃપ્રારંભ થાય તે પહેલાં બધું પૂર્ણ કરવા માટે પંદર મિનિટ છે. તે ક્યારે કરવું તે નક્કી કરવા Windows 10 છોડવાને બદલે, જો ત્યાં કોઈ અપડેટ હોય જેને પુનઃપ્રારંભની જરૂર પડશે અને જ્યારે તમે તેના માટે શેડ્યૂલ કરવા માંગો છો ત્યારે Windows પૂછી શકે છે. અપડેટ ડાઉનલોડ થઈ જાય તે પછી જ વપરાશકર્તા તે કરી શકે છે. જો વપરાશકર્તા પુનઃપ્રારંભ કરવા માટે અમુક ચોક્કસ સમયની છૂટ મર્યાદા મેળવવા માંગતા હોય, તો તેમને વિન્ડોઝ અપડેટ ફોર બિઝનેસ (Windows 10 Pro અને Enterprise માટે) માં સુવિધાઓની જરૂર પડશે જે પુનઃપ્રારંભને અવરોધિત કરે છે જેથી તે કામના કલાકોમાં અથવા ચોક્કસ તારીખો પર ન થાય.

યુનિવર્સલ એપ્સ - ઓફિસ

વિન્ડોઝ 10 સહિત એક નવું વિન્ડોઝ સ્ટોર મેળવે છે, જ્યાં ડેસ્કટોપ પ્રોગ્રામ્સ તેમજ આધુનિક વિન્ડોઝ એપ્સ ડાઉનલોડ થાય છે. તેમાંથી ઘણી બધી એપ્સ યુનિવર્સલ એપ્સ હશે જે પીસી, વિન્ડોઝ ફોન, એક્સબોક્સ વન અને હોલોલેન્સ પર પણ સમાન કોડ છે, જેમાં વિવિધ સ્ક્રીન માપોને અનુરૂપ

ઈન્ટરફેસ બદલાશે. આઉટલુક મેઈલ અને કેલેન્ડર એપ્સની જેમ વર્ક અને એક્સેલ જેવી વિન્ડોઝ એપ્લિકેશન્સ માટે ઓફિસ સાર્વત્રિક એપ્લિકેશન્સ છે.

સેટિંગ્સ અને કંટ્રોલ પેનલ

વિન્ડોઝ 8 સેટિંગ્સ એપ એ ઘણી વધુ સેટિંગ્સને કબજે કરી લીધી છે જે કંટ્રોલ પેનલમાં હતી, અને તેમાં નેવિગેટ કરવા માટે આઈકોન સાથેનું કંટ્રોલ પેનલ-શૈલી ઈન્ટરફેસ છે. પરંતુ જૂની કંટ્રોલ પેનલ ઈન્ટરફેસ હજી પણ ત્યાં છે, તે સેટિંગ્સ માટે કે જે નવી સેટિંગ્સ એપ્લિકેશનમાં નથી (અથવા જો તમે ત્યાં વસ્તુઓ શોધવા માટે ટેવાયેલા છો).

વિન્ડોઝ 10 - કીબોર્ડ શોર્ટકટ્સ

યુટીએસ મોટાભાગની વિન્ડોઝ એપ્લિકેશનની જેમ, ત્યાં કેટલાક કીબોર્ડ શોર્ટકટ્સ છે જેનો ઉપયોગ તમે અમુક કાર્યોને સરળ અથવા ઝડપી બનાવવા માટે કરી શકો છો.

મોટાભાગના નવા વિન્ડોઝ શોર્ટકટ્સ ઘણી ક્રિયાઓ કરવા માટે અન્ય કી સાથે વિન્ડોઝ કી () નો ઉપયોગ કરો. Windows 10 માં ઉપયોગમાં લેવાતા કેટલાક સૌથી સામાન્ય અથવા ઉપયોગી શોર્ટકટ્સ નીચે મુજબ છે.

વિન્ડોઝ 10 નેવિગેટ કરવા માટે કીબોર્ડ શોર્ટકટ્સ

Press This	To Do This
Windows Logo	સ્ટાર્ટ મેનૂને ટોંગલ કરો
Windows Logo+A	નોટિફિકેશન તકતી ખોલો
Windows Logo+B	સૂચના ક્ષેત્રને સક્રિય કરો છુપાયેલા ચિહ્નો તીર બતાવો (છુપાયેલા ચિહ્નો દર્શાવવા માટે એન્ટર દબાવો)
Windows Logo+C	વૉઇસ આદેશો માટે Cortana ખોલો
Windows Logo+D	ડેસ્કટોપ પ્રદર્શિત કરવા માટે બધી ખુલ્લી વિન્ડોને નાની કરો
Windows Logo+E	ફાઈલ એક્સપ્લોરર ચલાવો
Windows Logo+F	સ્ટાર્ટ મેનૂ પ્રદર્શિત કરો અને શોધ બોક્સને સક્રિય કરો
Windows Logo+H	શેર પેન દર્શાવો
Windows Logo+I	સેટિંગ્સ એપ્લિકેશન ચલાવો
Windows Logo+K	ઉપકરણો પેનલ પ્રદર્શિત કરો
Windows Logo+L	તમારા કમ્પ્યુટરને લોક કરો
Windows Logo+M	બધી વિન્ડો નાની કરો
Windows Logo+O	ટેબલ ઓરિએન્ટેશન લૉક ચાલુ અને બંધ કરો
Windows Logo+P	બીજા ડિસ્પ્લેને ગોઠવવા માટે પ્રોજેક્ટ ફલક પ્રદર્શિત કરો
Windows Logo+Q	વૉઇસ આદેશો માટે Cortana ખોલો
Windows Logo+R	રન ડાયલોગ બોક્સ ખોલો
Windows Logo+S	કીબોર્ડ આદેશો માટે Cortana ખોલો
Windows Logo+T	ટાસ્કબાર ચિહ્નો સક્રિય કરો (ચિહ્નો નેવિગેટ કરવા માટે એરો કીનો ઉપયોગ કરો)
Windows Logo+U	Ease of Access Center ખોલો
Windows Logo+W	શોધ બોક્સ સક્રિય કરો
Windows Logo+X	વિન્ડોઝ ટૂલ્સ અને ઉપયોગિતાઓનું મેનૂ પ્રદર્શિત કરો
Windows Logo+Z	એપ્લિકેશનના આદેશો પ્રદર્શિત કરો (જોકે આ ફક્ત કેટલીક આધુનિક એપ્લિકેશન્સમાં જ કાર્ય કરે છે)
Windows Logo+=	મેગ્નિફાયર ખોલો અને ઝૂમ ઈન કરો

Press This	To Do This
Windows Logo+-	ઝૂમ આઉટ કરો (જો મેગ્નિફાયરનો ઉપયોગ કરીને પહેલાથી જ ઝૂમ ઇન કરેલ હોય તો)
Windows Logo+,	અસ્થાયી રૂપે ડેસ્કટોપ પ્રદર્શિત કરો
Windows Logo+Enter	ઓપન નેરેટર
Windows Logo+Left	વર્તમાન એપ્લિકેશનને સ્ક્રીનની ડાબી બાજુએ સ્નેપ કરો
Windows Logo+Right	વર્તમાન એપ્લિકેશનને સ્ક્રીનની જમણી બાજુએ સ્નેપ કરો
Windows Logo+Up	ન્યૂનતમ એપ્લિકેશન પુનઃસ્થાપિત કરો; પુનઃસ્થાપિત એપ્લિકેશનને મહત્તમ કરો
Windows Logo+Down	મહત્તમ એપ્લિકેશન પુનઃસ્થાપિત કરો; પુનઃસ્થાપિત એપ્લિકેશનને નાનું કરો
Windows Logo+PgUp	વર્તમાન એપ્લિકેશનને ડાબી મોનિટર પર ખસેડો
Windows Logo+PgDn	વર્તમાન એપ્લિકેશનને જમણા મોનિટર પર ખસેડો
Windows Logo+PrtSc	વર્તમાન સ્ક્રીનને કેપ્ચર કરો અને તેને પિકચર્સ ફોલ્ડરમાં સાચવો
Windows Logo+Ctrl+D	વર્ચ્યુઅલ ડેસ્કટોપ બનાવો
Windows Logo+Ctrl + Right	આગલા વર્ચ્યુઅલ ડેસ્કટોપ પર સ્વિચ કરો
Windows Logo+Ctrl+ Left	અગાઉના વર્ચ્યુઅલ ડેસ્કટોપ પર સ્વિચ કરો
Windows Logo+Ctrl + F4	વર્તમાન વર્ચ્યુઅલ ડેસ્કટોપ બંધ કરો
Windows Logo+Tab	ટાસ્ક વ્યૂ ખોલો, જે દરેક ચાલી રહેલ એપ્લિકેશન તેમજ ઉપલબ્ધ વર્ચ્યુઅલ ડેસ્કટોપ માટે થંબનેલ્સ દર્શાવે છે

CMOS સેટઅપ અને વિન્ડોઝ ઇન્સ્ટોલેશન (Install Windows operating system)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- BIOS અને CMOS ના કાર્યોનું વર્ણન કરો
- BIOS સેટિંગ્સ જોવાની અને બદલવાની પદ્ધતિનું વર્ણન કરો
- પાર્ટીશન અને ફોર્મેટિંગના અર્થનું વર્ણન કરો
- વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરવાની પ્રક્રિયાનું વર્ણન કરો.

મૂળભૂત ઇનપુટ/આઉટપુટ સિસ્ટમ (BIOS), તરીકે પણ ઓળખાય છે સિસ્ટમ BIOS, ROM BIOS અથવા PC BIOS એ ફર્મવેર ઇન્ટરફેસને વ્યાખ્યાયિત કરતા સામાન્ય રીતે સ્વીકૃત ધોરણો છે.

BIOS નો મૂળભૂત હેતુ આરંભ કરવાનો છે અને સિસ્ટમ હાર્ડવેર ઘટકોનું પરીક્ષણ કરો અને લોડ કરો માસ મેમરી ઉપકરણમાંથી ઓપરેટિંગ સિસ્ટમ. BIOS ખાસ સૉફ્ટવેર છે જે મુખ્ય હાર્ડવેરને ઇન્ટરફેસ કરે છે ઓપરેટિંગ સિસ્ટમ સાથે કમ્પ્યુટરના ઘટકો. તે સામાન્ય રીતે ફ્લેશ મેમરી ચિપ પર સંગ્રહિત થાય છે મધરબોર્ડ, પરંતુ કેટલીકવાર ચિપ અન્ય પ્રકારનો હોય છે રોમ. BIOS એ ફર્મવેર છે (સૉફ્ટવેર સૂચનાઓ તમારા પર સ્થિત ચિપ પર કાયમી ધોરણે રેકૉર્ડ કરવામાં આવે છે મધરબોર્ડ). (ફિગ.1 નો સંદર્ભ લો).

Fig 1



BIOS ના કાર્યો

BIOS સૉફ્ટવેરમાં સંખ્યાબંધ વિવિધ ભૂમિકાઓ છે, પરંતુ તેની સૌથી મહત્વની ભૂમિકા ઓપરેટિંગ સિસ્ટમ લોડ કરવાની છે. BIOS દરેક પીસી હાર્ડવેરને તપાસે છે અને પ્રારંભ કરે છે હેન્ડિંગ પહેલાં સિસ્ટમ પાવર અપ અથવા પુનઃપ્રારંભ થવાનો સમય ઓપરેટિંગ સિસ્ટમ પર નિયંત્રણ. અન્ય કેટલાક સામાન્ય કાર્યો કે જે BIOS કરે છે તેમાં નીચેનાનો સમાવેશ થાય છે:

- એક પાવર-ઓન સેલ્ફ-ટેસ્ટ (POST) બધા માટે અલગ-અલગ ખાતરી કરવા માટે સિસ્ટમમાં હાર્ડવેર ઘટકો બધું બરાબર કામ કરી રહ્યું છે.
- ઇન્સ્ટોલ કરેલ વિવિધ કાડર્સ પર અન્ય BIOS ચિપ્સને સક્રિય કરવી કમ્પ્યુટરમાં - ઉદાહરણ તરીકે ગ્રાફિક્સ કાડર્સ વારંવાર તેમની પોતાની BIOS ચિપ્સ છે.
- નિમ્ન-સ્તરની દિનચર્યાઓનો સમૂહ પૂરો પાડવો જે ઓપરેટિંગ કરે છે સિસ્ટમ વિવિધ હાર્ડવેરને ઇન્ટરફેસ કરવા માટે વાપરે છે ઉપકરણો તેઓ કીબોર્ડ જેવી વસ્તુઓનું સંચાલન કરે છે સ્ક્રીન, અને બંદરો, ખાસ કરીને જ્યારે કમ્પ્યુટર બુટ કરી રહ્યું છે.
- હાર્ડ ડિસ્ક માટે સેટિંગ્સના સંગ્રહનું સંચાલન કરવું, ઘડિયાળ, વગેરે

CMOS સેટઅપ

BIOS જે કરશે તે પ્રથમ વસ્તુ માહિતીની તપાસ છે પર સ્થિત RAM ના નાના (64 બાઈટ્સ) જથ્થામાં સંગ્રહિત પૂરક મેટલ ઓક્સાઈડ સેમિકન્ડક્ટર (CMOS) ચિપ CMOS સેટઅપ વિગતવાર માહિતી પ્રદાન કરે છે તમારી સિસ્ટમ માટે ખાસ અને તમારા તરીકે બદલી શકાય છે સિસ્ટમ ફેરફારો. BIOS આ માહિતીનો ઉપયોગ કરે છે સંશોધિત કરો અથવા તેના ડિફોલ્ટ પ્રોગ્રામને જરૂરી પૂરક બનાવો.

BIOS ની ગોઠવણી

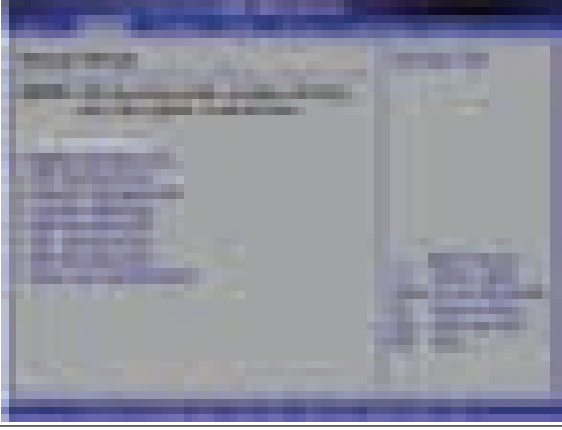
BIOS કસ્ટમ સેટિંગ્સ માટે CMOS સેટઅપને તપાસે છે. CMOS સેટિંગ્સ બદલવા માટે અમારે દાખલ કરવાની જરૂર છે CMOS સેટઅપ. CMOS સેટઅપ દાખલ કરવા માટે, ચોક્કસ કી અથવા કીઓનું સંયોજન પ્રારંભિક દરમિયાન દબાવવું આવશ્યક છે સ્ટાર્ટઅપ ક્રમ. મોટાભાગની સિસ્ટમો “Esc,” “Del,” “F1,” નો ઉપયોગ કરે છે. સેટઅપ દાખલ કરવા માટે “F2,” “Ctrl-Esc” અથવા “Ctrl-Alt-Esc”. ત્યાં છે સામાન્ય રીતે ડિસ્પ્લેના તળિયે ટેક્સ્ટની એક લીટી કે જે કહે છે “સેટઅપ દાખલ કરવા માટે ___ દબાવો.”

BIOS સેટઅપ એ સાથે ટેક્સ્ટ સ્ક્રીનનો સમૂહ બતાવે છે વિકલ્પોની સંખ્યા. આમાંના કેટલાક પ્રમાણભૂત છે, જ્યારે અન્ય BIOS ઉત્પાદક અનુસાર બદલાય છે. સામાન્ય વિકલ્પોમાં શામેલ છે:

- સિસ્ટમ સમય/તારીખ - સિસ્ટમ સમય અને તારીખ સેટ કરો
- બુટ સિક્વન્સ - ઓર્ડર કે જે BIOS લોડ કરવાનો પ્રયાસ કરશે ઓપરેટિંગ સિસ્ટમ
- પ્લગ એન્ડ પ્લે - કનેક્ટેડ સ્વતઃ-શોધ માટેનું માનક ઉપકરણો; “હા” પર સેટ કરવું જોઈએ જો તમારું કમ્પ્યુટર અને ઓપરેટિંગ સિસ્ટમ બંને તેને સપોર્ટ કરે છે
- માઉસ/કીબોર્ડ - “નમ લોક સક્ષમ કરો,” “સક્ષમ કરો કીબોર્ડ,” “માઉસ સ્વતઃ શોધો”...
- ડ્રાઈવ રૂપરેખાંકન - હાર્ડ ડ્રાઈવો, CD-ROM ને ગોઠવો અને ફ્લોપી ડ્રાઈવો
- મેમરી - BIOS ને ચોક્કસ પર પડછાયો કરવા માટે દિશામાન કરો મેમરી સરનામું
- સુરક્ષા - કમ્પ્યુટરને એક્સેસ કરવા માટે પાસવર્ડ સેટ કરો
- પાવર મેનેજમેન્ટ - પાવરનો ઉપયોગ કરવો કે નહીં તે પસંદ કરો મેનેજમેન્ટ, તેમજ “સ્ટેન્ડબાય” અને “સલ્પેન્ડ” માટે સમયની રકમ સેટ કરો
- બહાર નીકળો - તમારા ફેરફારો સાચવો, તમારા ફેરફારો કાઢી નાખો અથવા ડિફોલ્ટ સેટિંગ્સ પુનઃસ્થાપિત કરો.

BIOS કોઈપણ ફેરફારોને સાચવવા માટે CMOS ટેકનોલોજીનો ઉપયોગ કરે છે કમ્પ્યુટરની સેટિંગ્સમાં બનાવેલ છે. આ ટેકનોલોજી સાથે, એ નાની લિથિયમ અથવા Ni-Cad બેટરી પૂરતી પાવર સપ્લાય કરી શકે છે વર્ષો સુધી ડેટા રાખવા. મુખ્ય BIOS ઉત્પાદકો અમેરિકન Megatrends Inc. (AMI), ફોનિક્સનો સમાવેશ થાય છે ટેકનોલોજી, વિનબોન્ડ વગેરે. એક લાક્ષણિક BIOS સ્ક્રીનશૉટ અંજીરમાં બતાવેલ છે. નીચે આપેલ છે. (ફિગ.2 નો સંદર્ભ લો)

Fig 2



વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરી રહ્યું છે

હાર્ડ ડિસ્કને પાર્ટીશન કરવાની જરૂર છે (જોકે નહીં ફરજિયાત) અને તમે તેના પર ડેટા સ્ટોર કરી શકો તે પહેલાં ફોર્મેટ કરેલ છે.

પાર્ટીશન

પાર્ટીશન, જેને ક્યારેક વોલ્યુમ પણ કહેવાય છે, તે વિસ્તાર છે હાર્ડ ડિસ્ક પર કે જે ફાઇલ સિસ્ટમ સાથે ફોર્મેટ કરી શકાય છે અને મૂળાક્ષરના અક્ષરથી ઓળખાય છે. દાખલા તરીકે, મોટાભાગના વિન્ડોઝ કોમ્પ્યુટર પર ડ્રાઇવ સી એ પાર્ટીશન છે. આ તમે બનાવેલ પ્રથમ ત્રણ પાર્ટીશનો પ્રાથમિક પાર્ટીશનો છે. આનો ઉપયોગ ઓપરેટિંગ સિસ્ટમ શરૂ કરવા માટે થઈ શકે છે. જો તમે ત્રણ કરતા વધુ પાર્ટીશનો બનાવવા માંગો છો, ચોથું પાર્ટીશન વિસ્તૃત પાર્ટીશન તરીકે બનાવવામાં આવે છે.

વિસ્તૃત પાર્ટીશન એ એક કન્ટેનર છે જે એક અથવા પકડી શકે છે વધુ લોજિકલ ડ્રાઇવો. લોજિકલ ડ્રાઇવ પ્રાથમિકની જેમ કાર્ય કરે છે પાર્ટીશનો સિવાય કે તેઓ ઓપરેટિંગ સિસ્ટમ શરૂ કરવા માટે વાપરી શકાતા નથી.

ઘણા કમ્પ્યુટર્સ એક પાર્ટીશન તરીકે વિભાજિત થાય છે જે હાર્ડ ડિસ્કના કદની બરાબર છે. હાર્ડ ડિસ્કનું પાર્ટીશન કરવું ઘણા નાના પાર્ટીશનોમાં જરૂરી નથી, પરંતુ તે કરી શકે છે તમારી હાર્ડ ડિસ્ક પર ડેટા ગોઠવવા માટે ઉપયોગી થશે.

એક કરતા વધુ પાર્ટીશન બનાવવાના નીચેના ફાયદા છે:

- ઓપરેટિંગ સિસ્ટમ (OS) અને પ્રોગ્રામનું વિભાજન વપરાશકર્તા ફાઇલોમાંથી ફાઇલો.
- ઓપરેટિંગ સિસ્ટમ વર્ચ્યુઅલમેમરી સ્વેપિંગ/પેજિંગ માટે અલગ વિસ્તાર હોવો.
- વારંવાર વપરાતા પ્રોગ્રામ્સ અને ડેટાને એકબીજાની નજીક રાખવા.

- મલ્ટિ-બૂટ સેટઅપનો ઉપયોગ, જે વપરાશકર્તાઓને પરવાનગી આપે છે એક કમ્પ્યુટર પર એક કરતાં વધુ ઓપરેટિંગ સિસ્ટમ. ઉદાહરણ તરીકે, તમે Linux અને Microsoft ઇન્સ્ટોલ કરી શકો છો વિન્ડોઝ અથવા અન્ય ઓપરેટિંગ સિસ્ટમો પર વિવિધ સમાન HDD ના પાર્ટીશનો અને તેની પસંદગી છે પાવર-અપ પર કોઈપણ ઓપરેટિંગ સિસ્ટમમાં બુટ કરવું.
- તેને પુનઃપ્રાપ્ત કરવાનું સરળ બનાવવા માટે ફાઇલોને સુરક્ષિત કરવી અથવા તેને અલગ કરવી દૂષિત ફાઇલ સિસ્ટમ અથવા ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલેશન. જો એક પાર્ટીશન દૂષિત હોય, તો અન્ય ફાઇલ સિસ્ટમો કદાચ નહીં કરે અસર પામે છે.
- સિસ્ટમો પર એકંદર કોમ્પ્યુટર પ્રદર્શનને વધારવું જ્યાં નાની ફાઇલ સિસ્ટમો વધુ કાર્યક્ષમ હોય છે.
- પૂર્ણ કદ કરતાં નોંધપાત્ર રીતે ઓછા માટે પાર્ટીશન ઉપલબ્ધ ચેકડિસ્ક જેવા ડાયગ્નોસ્ટિક સાધનોનો સમય ઘટાડી શકે છે.

ફોર્મેટિંગ

ડિસ્ક ફોર્મેટિંગ એ ડેટા સ્ટોરેજ તૈયાર કરવાની પ્રક્રિયા છે ઉપકરણ જેમ કે હાર્ડ ડિસ્ક ડ્રાઇવ, સોલિડ-સ્ટેટ ડ્રાઇવ અથવા પ્રારંભિક ઉપયોગ માટે USB ફ્લેશ ડ્રાઇવ. તે બનાવવાની ક્રિયા છે વોલ્યુમ પર ફાઇલ સિસ્ટમ, જેથી ઓપરેટિંગ સિસ્ટમ તે વોલ્યુમ પર ડેટા સંગ્રહિત અને પુનઃપ્રાપ્ત કરી શકે છે.

ડિસ્કનું ફોર્મેટિંગ બે શ્રેણીઓનું છે:

- 1 લો-લેવલ ફોર્મેટિંગ (એટલે કે, હાર્ડવેરની સૌથી નજીક) માર્કર્સ સાથે ડિસ્કની સપાટીને ચિહ્નિત કરે છે રેકોર્ડિંગ બ્લોકની શરૂઆત સૂચવે છે. તે પણ માટે કરવામાં આવેલ બ્લોક તપાસો વિશે માહિતી પ્રદાન કરે છે ડેટા વાંચવા અથવા લખવા માટે ડિસ્ક નિયંત્રક દ્વારા ભાવિ ઉપયોગ. આનો કાયમી પાયો બનાવવાનો હેતુ છે ડિસ્ક, અને ઘણીવાર ફેક્ટરીમાં પૂર્ણ થાય છે. કદાચ ડિસ્કને તમારા પહેલાં પાર્ટીશન અને ફોર્મેટ કરવાની જરૂર છે તેના પર ડેટા સ્ટોર કરી શકે છે.
- 2 ઉચ્ચ-સ્તરનું ફોર્મેટિંગ ફાઇલ સિસ્ટમ ફોર્મેટ બનાવે છે ડિસ્ક પાર્ટીશન અથવા લોજિકલ વોલ્યુમની અંદર. આ ફોર્મેટિંગમાં દ્વારા ઉપયોગમાં લેવાતા ડેટા સ્ટ્રક્ચર્સનો સમાવેશ થાય છે લોજિકલ ડ્રાઇવ અથવા પાર્ટીશનના સમાવિષ્ટોને ઓળખવા માટે OS. આ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલેશન દરમિયાન અથવા નવી ડિસ્ક ઉમેરતી વખતે થઈ શકે છે.

વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરી રહ્યું છે

વિન્ડોઝ ઇન્સ્ટોલેશન પ્રક્રિયાના ત્રણ મૂળભૂત પ્રકારો નીચે મુજબ છે:

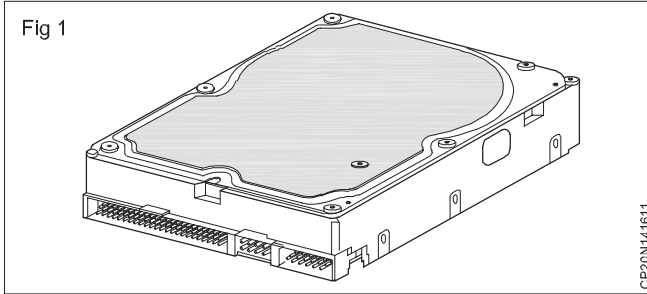
- તદ્દન નવી ડિસ્ક અથવા કમ્પ્યુટર સિસ્ટમ પર ઇન્સ્ટોલ કરો
- ડિસ્કને ભૂંસી નાખો, તેને ફોર્મેટ કરો અને ઇન્સ્ટોલ કરો.
- ડ્યુઅલ-બૂટિંગ માટે નવી ડિસ્ક્ટરીમાં ઇન્સ્ટોલ કરો

પ્રથમ બે પદ્ધતિઓ માટે, તે સુનિશ્ચિત કરવું આવશ્યક છે કે કમ્પ્યુટર ડીવીડી અથવા કોઈપણ અન્ય દૂર કરી શકાય તેવા માંથી બુટ કરી શકે છે ડ્રાઇવ આ કરવા માટે ડ્રાઇવ બૂટ ઓર્ડર બદલવાની જરૂર છે BIOS માં. નવીનતમ વિન્ડોઝ ડીવીડી બુટ કરી શકાય તેવી છે અને સેટઅપ પ્રોગ્રામ આપોઆપ ચલાવો. પછી ઇ-ન્ટ્રેડ પ્રેક્ટિકલ તરીકે અનુગામી સ્ક્રીનો પર સૂચવવામાં આવેલ પ્રક્રિયાને અનુસરીને ઇન્સ્ટોલેશન કરી શકાય છે.

COPA - કમ્પ્યુટર બેઝિક્સ અને સૉફ્ટવેર ઇન્સ્ટોલેશન**કમ્પ્યુટર બેઝિક્સ અને સૉફ્ટવેર ઇન્સ્ટોલેશન હાર્ડ ડિસ્કને ફોર્મેટ કરો અને પાર્ટીશન બનાવો (Format hard disk and create partition)**

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- BIOS અને CMOS ના કાર્યોનું વર્ણન કરો
- BIOS સેટિંગ્સ જોવાની અને બદલવાની પદ્ધતિનું વર્ણન કરો
- પાર્ટીશન અને ફોર્મેટિંગના અર્થનું વર્ણન કરો
- વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરવાની પ્રક્રિયાનું વર્ણન કરો.

હાર્ડ ડ્રાઈવને ફોર્મેટ કરો

હાર્ડ ડ્રાઈવને ફોર્મેટ કરવા માટે શા માટે જરૂરી છે તેના ઘણા કારણો છે, જેમ કે વિન્ડોઝ ફેશ ઇન્સ્ટોલ કરવા, વાયરસ અથવા માલવેરથી છુટકારો મેળવવા અથવા ફક્ત પીસીને રિફ્રેશ કરવામાં આવે છે અથવા વેચાણ પર સાફ કરવામાં આવે છે.

તે માત્ર હાર્ડ ડ્રાઈવ છે કે કેમ અને ફાજલ પીસી છે કે નહીં તેના આધારે પ્રક્રિયા અલગ હોઈ શકે છે.

તે ફોર્મેટ કરી શકાતું નથી, હાર્ડ ડ્રાઈવ કે જેના પર વિન્ડોઝ ચાલી રહ્યું છે. આ કરવા માટે, તેને Windows ઇન્સ્ટોલેશન ડિસ્ક, USB ફ્લેશ ડ્રાઈવ અથવા અન્ય બુટ કરી શકાય તેવી ડિસ્કમાંથી પીસીને બુટ કરવાની જરૂર છે.

ડિસ્કને ફોર્મેટ કરીએ?

ફોર્મેટિંગ એ હાર્ડ ડ્રાઈવ પરના તમામ ડેટાને કાઢી નાખવાની પ્રક્રિયા છે, પરંતુ 'ક્વિક ફોર્મેટ' થી સાવચેત રહો જે તમામ ડેટાને સ્થાને છોડી દે છે અને ડ્રાઈવ ખાલી હોવાનું જણાય છે. જો તદ્દન નવી હાર્ડ ડ્રાઈવ હોય, અથવા વિન્ડોઝને પુનઃસ્થાપિત કરવાની જરૂર હોય તો ઝડપી ફોર્મેટ બરાબર છે, પરંતુ જો તે ડિસ્કનો નિકાલ કરી રહી હોય અથવા કોઈ અન્યને આપી રહી હોય તો નહીં.

ચેતવણીનો શબ્દ: ડ્રાઈવમાંથી કોઈપણ ફોટા, વિડિયો, સંગીત અને અન્ય દસ્તાવેજોને ફોર્મેટ કરતા પહેલા તેનું સફળતાપૂર્વક બેકઅપ લેવાની ખાતરી કરો. જોકે કાઢી નાખેલી ફાઈલો કેટલીક પરિસ્થિતિઓમાં પુનઃપ્રાપ્ત કરી શકાય છે, નિવારણ હંમેશા ઉપચાર કરતાં વધુ સારું છે.

હાર્ડ ડ્રાઈવ પાર્ટીશન ફોર્મેટ

કરો શરૂ કરતા પહેલા પાર્ટીશનો વિશે સમજવું મહત્વપૂર્ણ છે. હાર્ડ ડ્રાઈવને નાના ભાગોમાં વિભાજિત કરી શકાય છે, જેને પાર્ટીશનો કહેવાય છે. એક પાર્ટીશનને ફોર્મેટ કરવું શક્ય છે જ્યારે અન્યને અસ્પૃશ્ય છોડીને.

જો સમગ્ર હાર્ડ ડ્રાઈવને ફોર્મેટ કરવા અને એક બ્લોકમાં સમગ્ર ક્ષમતાનો ઉપયોગ કરવાની જરૂર હોય, તો પાર્ટીશન માહિતી કાઢી નાખો.

BIOS માંથી હાર્ડ ડ્રાઈવને ફોર્મેટ કરીએ?

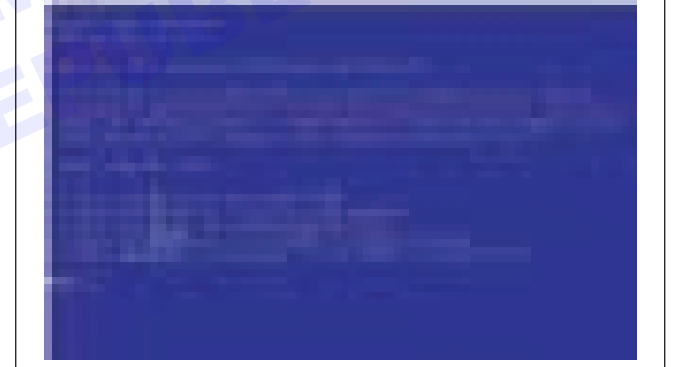
ઘણા લોકો પૂછે છે કે BIOS માંથી હાર્ડ ડિસ્ક કેવી રીતે ફોર્મેટ કરવી.

ટૂંકો જવાબ ના છે.

જો ડિસ્કને ફોર્મેટ કરવાની જરૂર હોય અને તમે તેને Windows ની અંદરથી કરી શકતા નથી, તો બુટ કરી શકાય તેવી CD, DVD અથવા USB ફ્લેશ ડ્રાઈવ બનાવો અને મફત તૃતીય-પક્ષ ફોર્મેટિંગ ટૂલ ચલાવો.

એક વિકલ્પ છે Darik's Boot and Nuke (DBAN), જે વ્યક્તિગત ઉપયોગ માટે મફત છે. આ પ્રોગ્રામ તમારી હાર્ડ ડિસ્કને સંપૂર્ણપણે ભૂંસી નાખશે અને ફોર્મેટ કરશે, નવી OS ના કલીન ઇન્સ્ટોલ કરવાની મંજૂરી આપશે, પરંતુ પ્રક્રિયા પૂર્વવત્ કરી શકાશે નહીં.

Fig 2



DBAN માત્ર બુટ કરી શકાય તેવી CD/DVD-R બનાવવા માટે સક્ષમ હોવાનું માનવામાં આવે છે, પરંતુ જો કોઈ ખાલી ડિસ્ક અથવા બર્નર ન હોય, તો એક અલગ તૃતીય પક્ષ પ્રોગ્રામના રૂપમાં વર્કઅરાઉન્ડ ઉપલબ્ધ છે.

યુનિવર્સલ યુએસબી ઇન્સ્ટોલર બૂટેબલ યુએસબીમાંથી ચલાવવા માટે ડાઉનલોડ કરેલી DBAN ISO ઇમેજને ઝડપથી અને સરળતાથી કન્વર્ટ કરશે.

ખાલી USB ફ્લેશ ડ્રાઈવ દાખલ કરો, યુનિવર્સલ USB ઇન્સ્ટોલર સેટઅપ પ્રોગ્રામ ચલાવો અને પ્રોમ્પ્ટ્સને અનુસરો.

તે સ્કોલ કરવાનું કહેશે અને લિનક્સ ડિસ્ટ્રિબ્યુશન જે USB પર ઇન્સ્ટોલ કરવા માંગે છે તેને પસંદ કરશે (આ કિસ્સામાં, DBAN નું નવીનતમ સંસ્કરણ), ત્યારબાદ કમ્પ્યુટર પર તેનું સ્થાન અને USB ડ્રાઈવનો અક્ષર તેને ઇન્સ્ટોલ કરવા માટે પસંદ કરશે.

એકવાર તે માહિતી પૂર્ણ થઈ જાય, પછી બનાવો ક્લિક કરો.

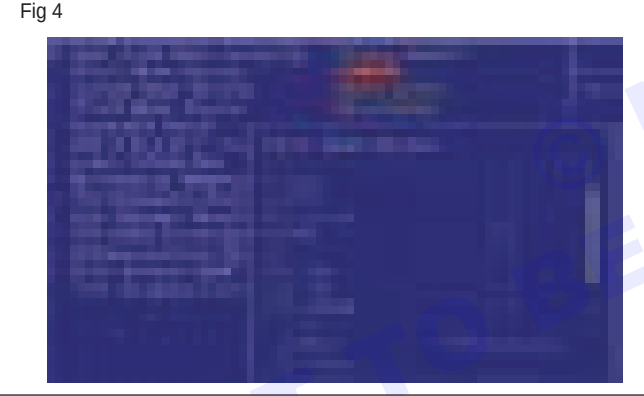
સામાન્ય બુટ ઉપકરણને બદલે આ USB ડ્રાઈવમાંથી બુટ કરવા માટે (મોટા ભાગના કિસ્સાઓમાં, આ હાર્ડ ડ્રાઈવ હશે), તેને BIOS માં કેટલીક સેટિંગ્સ બદલવી પડશે.

BIOS માં, બુટ ઓર્ડર સેટિંગ્સ પર નેવિગેટ કરો અને પ્રાથમિક બુટ ઉપકરણને USB ડ્રાઈવમાં બદલો (આ કરવા માટે તેને પ્લગ ઇન કરવાની જરૂર નથી. પસંદગી). સેટિંગ્સ સાચવ્યા પછી અને BIOS માંથી બહાર નીકળ્યા પછી, બૂટ કરી શકાય તેવી USB દાખલ કરો, કમ્પ્યુટરને ફરીથી પ્રારંભ કરો

Fig 3



Fig 4



પીસીએ DBAN સોફ્ટવેરને આપમેળે બુટ કરવું જોઈએ, જે હાર્ડ ડ્રાઈવને ભૂંસી નાખવાની પ્રક્રિયામાં માર્ગદર્શન આપશે, જેમાં ડેટા-વાઈપિંગના વિવિધ સ્તરોના વિકલ્પો છે.

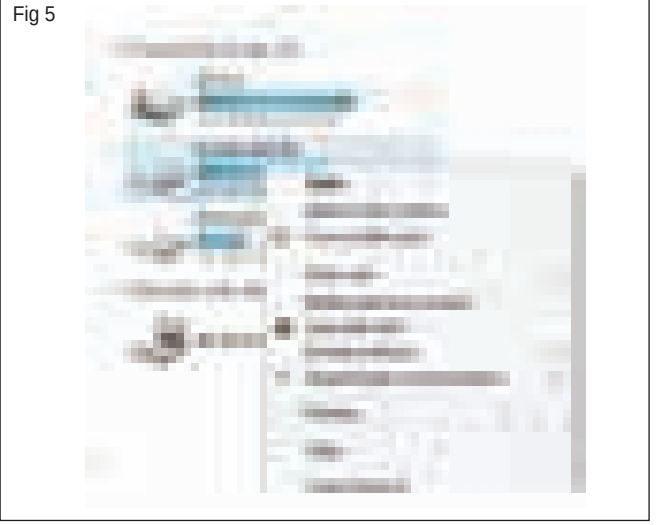
તે USB ને બીજી ડ્રાઈવ તરીકે ગણાશે જેથી અજાણતાં તેને સાફ ન થાય તે માટે, DBAN માં બુટ થયા પછી તેને દૂર કરો.

ઝડપી ફોર્મેટ હાર્ડ ડ્રાઈવ?

હા, પરંતુ જો ડેટા કાયમ માટે ભૂંસી નાખવા માંગતા હોય તો આ પદ્ધતિનો ઉપયોગ કરશો નહીં. ઝડપી ફોર્મેટ ડેટાને ડિલીટ કરતું નથી પરંતુ તેના બદલે ફાઈલોના માત્ર નિર્દેશકોને જ ભૂંસી નાખે છે.

વિન્ડોઝ વિસ્ટા, 7, 8 અને 10 માં બિલ્ટ-ઇન ડિસ્ક મેનેજમેન્ટ ટૂલ છે (નીચે જુઓ), પરંતુ હાર્ડ ડ્રાઈવને ફોર્મેટ કરવાની સૌથી ઝડપી રીત એ છે કે સ્ટાર્ટ બટનને ક્લિક કરો, પછી કમ્પ્યુટર અને વાઈપ કરવા માટે હાર્ડ ડ્રાઈવ પર જમણું-ક્લિક કરો. તે ડ્રાઈવને ફોર્મેટ કરી શકાતી નથી કે જેના પર વિન્ડોઝ સ્પષ્ટ કારણોસર ઈન્સ્ટોલ કરેલું છે.

Fig 5



મૂળભૂત રીતે ઝડપી ફોર્મેટ ચકાસાયેલ છે, અને ફાઈલ સિસ્ટમ અને ફાળવણી એકમનું કદ તેમજ વોલ્યુમ લેબલ (ડ્રાઈવનું નામ) બદલવાની પસંદગી કરો. સામાન્ય રીતે, બધી સેટિંગ્સને યથાવત રાખો અને પ્રારંભ બટનને ક્લિક કરો. એક મિનિટમાં હાર્ડ ડ્રાઈવ ફોર્મેટ થઈ જશે.

Fig 6



જો તે Windows Vista, 7, 8 અથવા 10 માટે પહેલાથી પસંદ કરેલ ન હોય તો ફાઈલ સિસ્ટમ તરીકે NTFS પસંદ કરો અને ખાતરી કરો કે ફાળવણી એકમનું કદ 'ડિફોલ્ટ ફાળવણી કદ' પર સેટ છે.

ડિસ્ક મેનેજમેન્ટ ટૂલનો

ઉપયોગ કરીને Vista, 7, 8 અથવા 10 માં શોધ બોક્સમાં diskmgmt.msc અથવા ડિસ્ક મેનેજમેન્ટ લખો અને પછી તે જ નામ સાથે, ઉપરના મેનુમાં દેખાતા માત્ર પરિણામ પર ક્લિક કરો.

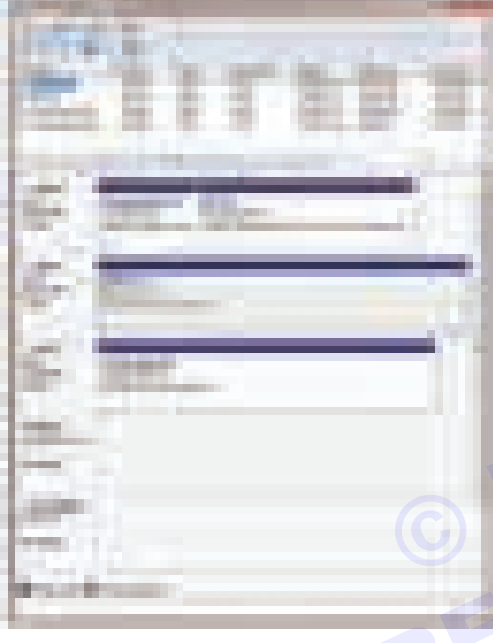
ડિસ્ક મેનેજમેન્ટ શરૂ કરવાની આ સૌથી સહેલી રીત છે, પણ જો 'ડિસ્ક' માટે શોધ કરો અને 'હાર્ડ ડિસ્ક પાર્ટીશનો બનાવો અને ફોર્મેટ કરો' પસંદ કરો તો તેને કંટ્રોલ પેનલમાં પણ શોધો. ડિસ્ક મેનેજમેન્ટ સ્ટેન્ડઅલોન પાર્ટીશન મેનેજમેન્ટ ટૂલ જેટલું શક્તિશાળી નથી, પરંતુ તે હજુ પણ ડેટાને ફોર્મેટ કરવા સક્ષમ છે.

જો તે Windows Vista, 7, 8 અથવા 10 માટે પહેલાથી પસંદ કરેલ ન હોય તો ફાઈલ સિસ્ટમ તરીકે NTFS પસંદ કરો અને ખાતરી કરો કે ફાઈલવણી એકમનું કદ 'ડિફોલ્ટ ફાઈલવણી કદ' પર સેટ છે.

ડિસ્ક મેનેજમેન્ટ ટૂલનો ઉપયોગ કરીને Vista, 7, 8 અથવા 10 માં શોધ બોક્સમાં diskmgmt.msc અથવા ડિસ્ક મેનેજમેન્ટ લખો અને પછી તે જ નામ સાથે, ઉપરના મેનુમાં દેખાતા માત્ર પરિણામ પર ક્લિક કરો.

ડિસ્ક મેનેજમેન્ટ શરૂ કરવાની આ સૌથી સહેલી રીત છે, પણ જો 'ડિસ્ક' માટે શોધ કરો અને 'હાર્ડ ડિસ્ક પાર્ટીશનો બનાવો અને ફોર્મેટ કરો' પસંદ કરો તો તેને કંટ્રોલ પેનલમાં પણ શોધો. ડિસ્ક મેનેજમેન્ટ સ્ટેન્ડઅલોન પાર્ટીશન મેનેજમેન્ટ ટૂલ જેટલું શક્તિશાળી નથી, પરંતુ તે હજુ પણ ડેટાને ફોર્મેટ કરવા સક્ષમ છે.

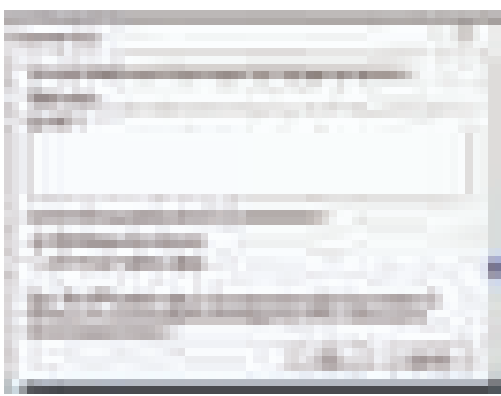
Fig 7



જો તે પીસીમાં નવી (વધારાની) હાર્ડ ડ્રાઈવ ઇન્સ્ટોલ કરવાની હોય, તો તે વિન્ડોઝ એક્સપ્લોરરમાં શા માટે દેખાતું નથી તે આશ્ચર્યજનક હોઈ શકે છે. કારણ એ છે કે તેને આરંભ અને ફોર્મેટ કરવાની જરૂર છે

જે ડિસ્ક મેનેજમેન્ટમાં કરી શકાય છે. જ્યારે ટૂલ લોડ થાય છે, ત્યારે તે કોમ્પ્યુટરની તમામ ડ્રાઈવોનું વિશ્લેષણ કરશે અને તેને મળેલી નવી ડિસ્કને પ્રારંભ કરવા માટે સંકેત આપશે.

Fig 8



જો ડિસ્ક 2TB કરતા મોટી હોય, તો GPT (GUID પાર્ટીશન ટેબલ) પસંદ કરો. આ સેટિંગ ચાર કરતા વધુ પાર્ટીશનો બનાવવા માટે પણ પરવાનગી આપે છે.

જો પ્રોમ્પ્ટ દેખાતો નથી, તો ડ્રાઈવ્સની સૂચિમાં જુઓ અને એક જુઓ જે કહે છે કે 'પ્રારંભ નથી'. તેના પર જમણું-ક્લિક કરો અને ડિસ્કને પ્રારંભ કરો પસંદ કરો.

એકવાર તે થઈ જાય, પછી હેચ કરેલ અનએલોકેટેડ સ્પેસ પર જમણું-ક્લિક કરો અને નવું સરળ વોલ્યુમ પસંદ કરો...

સૂચનાઓને અનુસરો, પાર્ટીશન કેટલું મોટું હોવું જોઈએ (MB - 1024MB = 1GB માં), અને કયો ડ્રાઈવ લેટર સોંપવો તે પસંદ કરીને (એક પસંદ કરવામાં આવશે, પરંતુ જો ઇચ્છિત હોય તો તેને બદલવાનું પસંદ કરી શકો છો).

જો ડ્રાઈવની કુલ ક્ષમતા કરતાં નાની પાર્ટીશન માટે કદ પસંદ કરો, તો 1TB ડ્રાઈવ પર 500GB કહો, ડ્રાઈવ પર અમુક બિન-ફાઈલવેલ જગ્યા સાથે સમાપ્ત થાય છે જે હમણાં જ પૂર્ણ થયેલ પ્રક્રિયાને પુનરાવર્તિત કરીને ફોર્મેટ કરી શકે છે.

પાર્ટીશન માપ બદલો

ડિસ્ક મેનેજમેન્ટ પાર્ટીશનને વિસ્તૃત અથવા સંકોચવા માટે વાપરી શકાય છે. ફક્ત એક પર જમણું-ક્લિક કરો અને દેખાતા મેનુમાંથી યોગ્ય વિકલ્પ પસંદ કરો. જો સંકોચાય છે, તો પાર્ટીશન કેટલું છે તે જાણવા માટે તપાસવામાં આવશે તે ખાલી જગ્યા સમાવે છે

Fig 9



તે થોડું ગૂંચવણભર્યું છે કારણ કે નંબરો GB ને બદલે MB માં પ્રદર્શિત થાય છે, પરંતુ તેને સંકોચવા માટે જગ્યાના જથ્થાને સમાયોજિત કરી શકાય છે અને 'સંકોચન પછી કુલ કદ' અપડેટ કરવામાં આવશે, તે બિંદુથી આગળ પાર્ટીશનને સંકોચાઈ શકતું નથી. ફાઈલો સ્થિત છે - તે પહેલા ડ્રાઈવને ડિફેગમેન્ટ કરીને જગ્યા ખાલી કરવામાં સક્ષમ હશે.

તેનાથી વિપરિત તે માત્ર ત્યારે જ પાર્ટીશનને વિસ્તૃત કરી શકે છે જો ડ્રાઈવ પર ન વપરાયેલ જગ્યા હોય. જો નહીં, તો વિકલ્પ ગ્રે થઈ જશે.

વિન્ડોઝ રીસેટ

કરવું જો પીસીને સમસ્યા આવી રહી છે અને વપરાશકર્તા હાર્ડ ડ્રાઈવને સાફ કરીને વ્યક્તિગત ફાઈલો ગુમાવવા માંગતા નથી, તો તે પીસીને તાજું કરવા અથવા રીસેટ કરવા માંગે છે જે Windows 8 અને 10 માં કરી શકાય છે.

જો વપરાશકર્તા તમારી અંગત ફાઈલો અને સેટિંગ્સ રાખવા માંગે છે, પરંતુ નવી વિન્ડોઝ ઇન્સ્ટોલ કરવા માંગે છે, તો તે બધા પીસીને તાજું કરવા માંગે છે.

નોંધ કરો કે રિફ્રેશ મશીન પર ઇન્સ્ટોલ કરેલા તમામ પ્રોગ્રામ્સ અને એપ્લિકેશન્સને દૂર કરશે, પરંતુ Windows-ડિફોલ્ટ પ્રોગ્રામ્સને અકબંધ રાખશે.

રીસેટ વિન્ડોઝને ફરીથી ઇન્સ્ટોલ કરે છે અને બધી ફાઈલો, સેટિંગ્સ અને પ્રોગ્રામ્સને કાઢી નાખે છે, તે સૂચવવામાં આવે છે જો તમે અગાઉ

બધી ફાઈલોનું બેકઅપ લીધું હોય અને વપરાશકર્તાની અંગત ફાઈલોને સ્થાનાંતરિત કરવામાં કોઈ વાંધો ન હોય તો આ કરવું.

વિન્ડોઝ 10 એ થોડો અલગ અભિગમ ધરાવે છે અને તે વિન્ડોઝ 8 માંથી આવતા લોકોને મૂંઝવણમાં મૂકી શકે છે. માઈક્રોસોફ્ટે રિફ્રેશ વિકલ્પ દૂર કર્યો છે અને તેના બદલે રિફ્રેશ અને રીસેટ વિકલ્પોને એક સેટિંગમાં જોડ્યા છે.

વિકલ્પ શોધવા માટે, સ્ટાર્ટ મેનૂ ખોલો, સેટિંગ્સ > અપડેટ અને સુરક્ષા > પુનઃપ્રાપ્તિ > પ્રારંભ કરો (રીસેટ આ પીસી વિકલ્પ હેઠળ) પર ક્લિક કરો.

વિકલ્પ પસંદ કરવા પર, તે ત્રણ અલગ-અલગ વિકલ્પો સાથે રજૂ કરશે:

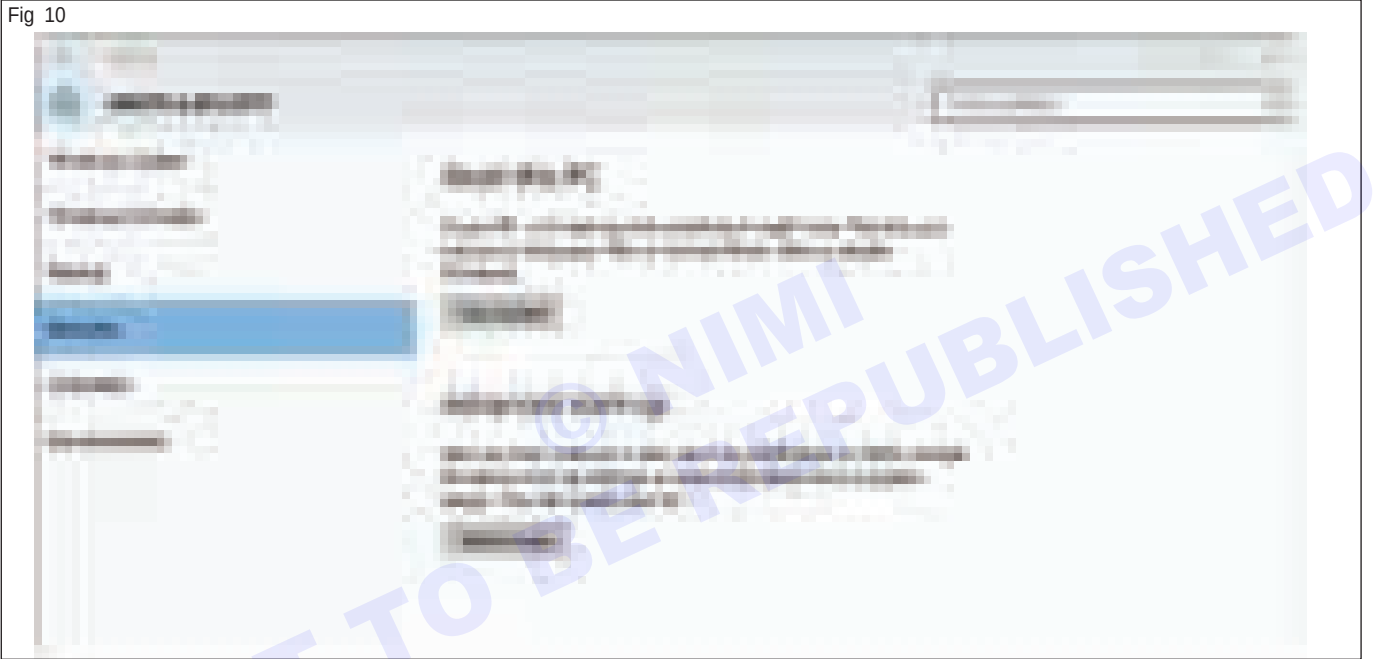
- મારી ફાઈલો રાખો
- બધું દૂર કરો
- ફેક્ટરી સેટિંગ્સ પુનઃસ્થાપિત કરો (બધા પીસી પર ઉપલબ્ધ નથી)

‘મારી ફાઈલો રાખો’ વ્યક્તિગત ફાઈલોને સાચવે છે પરંતુ વિન્ડોઝ 10 પુનઃસ્થાપિત કરતી વખતે વપરાશકર્તા સેટિંગ્સ, ડ્રાઈવરો અને પ્રોગ્રામ્સ કાઢી નાખે છે.

‘બધું દૂર કરો’ વ્યક્તિગત ફાઈલોથી પણ છુટકારો મેળવે છે અને ‘ફેક્ટરી સેટિંગ્સ પુનઃસ્થાપિત કરો’ એ ‘બધું દૂર કરો’ જેવી જ ક્રિયાઓ કરે છે, પરંતુ પીસીને વિન્ડોઝના સંસ્કરણ પર ફરીથી સેટ કરે છે જે મશીન સાથે આવે છે.

આ વિકલ્પ ફક્ત તે લોકો માટે જ ઉપલબ્ધ હશે જેમણે એ ખરીદ્યું છે પ્રી-બિલ્ટ પીસી અથવા લેપટોપ જે વિન્ડોઝ સાથે આવે છે તે પહેલાથી જ ઇન્સ્ટોલ કરેલું છે.

Fig 10



OS ઇન્સ્ટોલેશન દરમિયાન સામાન્ય હાર્ડવેર અને સોફ્ટવેર સમસ્યાઓને ઓળખો અને સુધારો (Identify and rectify common hardware and software issues during OS installation)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- પીસીના મુશ્કેલીનિવારણમાં સામેલ મૂળભૂત પગલાં જણાવો
- સમસ્યા હલ કરવા માટે મૂળભૂત અભિગમ સમજાવો
- ખામીયુક્ત કમ્પ્યુટરમાં સંભવિત ખામીઓ અને લક્ષણોની યાદી બનાવો
- ફરિયાદોનાં કારણોનું વિશ્લેષણ કરો
- ખામીઓને દૂર કરવા માટેનો સૌથી ટૂંકો રસ્તો જણાવો.
- ખામીયુક્ત કમ્પ્યુટરમાં સંભવિત ખામી અને લક્ષણોની યાદી બનાવો
- ફરિયાદના કારણોનું પૃથ્થકરણ કરો “જ્યારે વિન્ડોઝ શરૂ થાય છે, ત્યારે સિસ્ટમ સપાટીની તપાસ કરે છે અને જાય છે પ્રોબ્લેમ ટ્રી અને ટીએસસીની મદદથી સલામત મોડ”

મૂળભૂત મુશ્કેલીનિવારણ

પર સમસ્યાઓનું નિવારણ કરતી વખતે મુશ્કેલીઓમાંથી એક પીસી એ છે કે મોટાભાગના કિસ્સાઓમાં તેઓ જે દેખાય છે તે નથી. પીસીના વારંવાર હેંગ થવા પાછળનું કારણ હોઈ શકે છે છ અથવા વધુ સારી રીતે વ્યાખ્યાયિત વિસ્તારોમાંથી એક અથવા એક ડાઉન સુધી અજાણી સમસ્યાઓ. સમસ્યાને કારણે હોઈ શકે છે સોફ્ટવેર અથવા હાર્ડવેર. વર્ષોના અનુભવ સાથે પણ અને તાલીમ, પીસી ટેકનિશિયન મુશ્કેલીનિવારણ સાથે બહાર આવે છે પ્રક્રિયાઓ જે વાસ્તવિક સમસ્યાને હલ કરતી નથી. માટે ઉદાહરણ તરીકે, વિન્ડોઝ ઓપરેટ ચલાવતા વ્યક્તિગત કમ્પ્યુટર- કનેક્ટેડ ઘણા i/o કાર્ડ સાથે ing સિસ્ટમ સ્થિર થઈ શકે છે સ્ક્રીન, માઉસ અને કીબોર્ડ અને 3 જેટલો લાંબો સમય લે છે જવાબ આપતા પહેલા મિનિટો. બધા સાથે પ્રયાસ કર્યા પછી મેમરી મોડ્યુલ બદલવા, નવું ઇન્સ્ટોલ કરવા જેવા વિકલ્પો સમાંતર પોર્ટ અને NIC ડ્રાઇવરો, ટેકનિશિયન છેલ્લે તપાસે છે તે મહત્વપૂર્ણ ઓપરેટિંગ સિસ્ટમ શોધવા માટે સિસ્ટમ લોગ કરે છે લાઇબ્રેરી દૂષિત હતી અને તેને ફરીથી ઇન્સ્ટોલ કરવાની જરૂર હતી. આ અહીં ઉલ્લેખ કરવાનો મુદ્દો એ છે કે જો ટેકનિશિયનનો ઉપયોગ થાય છે મુશ્કેલીનિવારણ માટે વ્યવસ્થિત અભિગમ, સમસ્યા કરશે સંભવતઃ ખૂબ પહેલા ઉકેલાઈ ગયા છે. જાળવણી સારી મુશ્કેલીનિવારણ યોજના ચોક્કસપણે અમને અવકાશ આપે છે સમસ્યાનો વધુ વ્યવસ્થિત રીતે સંપર્ક કરવો અને વૈજ્ઞાનિક રીત. મુશ્કેલીનિવારણ યોજના સિવાય બીજું કંઈ નથી એક લેખિત ચેક લિસ્ટ જેનો ઉપયોગ અમે કોઈપણ સમસ્યા માટે કરીએ છીએ. આ તત્વો કે જે કોઈપણ મુશ્કેલીનિવારણમાં શામેલ હોવા જોઈએ યોજના નીચે મુજબ છે.

- જાળવણી રેકૉર્ડ
- સંભવિત કારણોની ઓળખ
- સંભવિત ઉકેલોની ઓળખ
- ઉકેલોની અરજી અને પરીક્ષણ
- ફોલો-અપ

જાળવણી રેકૉર્ડમાં, ઇન્સ્ટોલ કરેલ હાર્ડવેર રેકૉર્ડ કરો પીસીમાં જ્યારે તે ઇન્સ્ટોલ થાય છે, ત્યારે તમામ નિવારક જાળવણી પ્રવૃત્તિઓ, તમામ સોફ્ટવેર અપડેટ્સ અથવા ઉમેરાઓ, અને તમામ હાર્ડવેર ઇન્સ્ટોલેશન અને અપગ્રેડ. આગળ કોઈપણ સમસ્યા- Iems કે જે થાય છે અને તમે તેને ઉકેલવા માટે શું પગલાં લો છો નોંધવું જોઈએ. જ્યારે મુશ્કેલીનિવારણની વાત આવે છે પીસી, જાળવણી રેકૉર્ડ સાથે એક પિન ડાઉન કરી શકે છે

સમસ્યા અને તેનું નિરાકરણ. આવી પ્રથમ એન્ટ્રી રેકૉર્ડ એ પીસીની પ્રોફાઇલ હોવી જોઈએ, જેમાં તેનો સમાવેશ થાય છે રૂપરેખાંકન, ઓપરેટિંગ સિસ્ટમ અને દરેક તારીખ ઘટક સ્થાપિત કરવામાં આવ્યું હતું.

આવા કોષ્ટકો માહિતીના પ્રકારનો ખ્યાલ આપે છે સમરાઈઝડ તરીકે, સિસ્ટમ રૂપરેખાંકનમાં શામેલ હોવું જોઈએ નીચે:

- પ્રોસેસરની મેક, મોડલ અને સ્પીડ.
- સિસ્ટમ મેમરી (RAM) અને મેમરીની માત્રા મોડ્યુલ પ્રકાર, કદ અને મેમરી રૂપરેખાંકન
- હાર્ડ ડિસ્કનું કદ અને ઇન્ટરફેસનો પ્રકાર - કોમ્પેક્ટ ડિસ્ક (સીડી)નું બનાવો, મોડેલ અને ઝડપ અથવા ડિજિટલ વર્સટાઇલ ડિસ્ક (ડીવીડી) ડ્રાઇવ
- વિડિઓ અથવા ગ્રાફિક્સ એડેપ્ટરનું મેમરી કદ
- મોડેમનું મેક, મોડલ, પ્રકાર અને ગતિ
- ઓપરેટિંગ સિસ્ટમનો સંસ્કરણ નંબર
- પીસી પર ઇન્સ્ટોલ કરેલ સોફ્ટવેર એપ્લિકેશન્સની સૂચિ
- પીસી સાથે જોડાયેલ પેરિફરલ સાધનોની યાદી, તેઓ જે પોર્ટ સાથે જોડાયેલા છે તે દર્શાવે છે

જાળવણી રેકૉર્ડ દરેક વખતે અપડેટ થવો જોઈએ પીસી પર જાળવણી કાર્ય હાથ ધરવામાં આવે છે. કોઈપણ સમયે નવું અથવા રિપ્લેસમેન્ટ હાર્ડવેર ઇન્સ્ટોલ કરેલું છે, પ્રવૃત્તિ રેકૉર્ડ કરો અને સિસ્ટમ રૂપરેખાંકન અપડેટ કરો. પ્રવૃત્તિ એન્ટ્રીઓ સમાવેશ કરવો જોઈએ

- પ્રવૃત્તિની તારીખ અથવા સિસ્ટમમાં કરેલા ફેરફારો
- કોઈપણ હાર્ડવેરનો મેક, મોડલ અને સીરીયલ નંબર પીસીમાં દૂર અથવા ઉમેરવામાં આવે છે
- કોઈપણ સોફ્ટવેરનું નામ, સંસ્કરણ અને પ્રકાશક ઉમેરવામાં આવે છે પીસી માટે
- કોઈપણ રૂપરેખાંકન ફેરફારો પર વિગતવાર માહિતી મૂળભૂત ઇનપુટ/ આઉટપુટ સિસ્ટમ(BIOS) અથવા નવા ઉપકરણ અથવા સોફ્ટવેર માટે અન્ય રૂપરેખાંકન માટે બનાવેલ છે.

મુશ્કેલીનિવારણ અભિગમ

પીસી સાથે સંકળાયેલી કોઈપણ સમસ્યાઓના નિરાકરણ માટે, પ્રથમ જાઓ તે પીસીના જાળવણી રેકૉર્ડ દ્વારા અને અનુસરો a સમસ્યાને અલગ કરવા માટેની પદ્ધતિસરની પ્રક્રિયા. આ પ્રમાણભૂત સમસ્યા હલ કરવાની પ્રક્રિયામાં નીચેના પગલાંઓ શામેલ છે.

સમસ્યાને ઓળખો: આ સૌથી મુશ્કેલ ભાગ છે પ્રક્રિયા આ પગલું સફળતાપૂર્વક કરવા માટે બધા એકત્રિત કરો સમસ્યા વિશેનો ડેટા

સંભવિત કારણોને ઓળખો: ના તમામ લક્ષણોનું વિશ્લેષણ કરો સમસ્યા અને તમામ કારણોને ક્રમમાં સૂચિબદ્ધ કરવાનો પ્રયાસ કરો સૌથી વધુ શક્યતા ઓછામાં ઓછી શક્યતા

સંભવિત ઉકેલો ઓળખો: દરેક માટે ઉકેલો ઓળખો જે કારણો ઓળખવામાં આવે છે. સંભવિત કારણ એક કરતાં વધુ સંભવિત ઉકેલ હોઈ શકે છે.

સંભવિત ઉકેલોનું પૃથક્કરણ કરો: જો બે ઉકેલો સમાન પરિણામ આપે છે, તો એક વધુ છે તે ધ્યાનમાં લો આર્થિક અને તે જ લાગુ કરો.

ઉપરોક્ત પગલાંને અનુસર્યા પછી અને ઉકેલ પર સમસ્યા જાળવણી રેકૉર્ડ અપડેટ કરો અને બનાવો તેમાં જરૂરી પ્રવેશો. કેટલાક કિસ્સાઓમાં સમસ્યાઓ ખૂબ જ સ્પષ્ટ હોઈ શકે છે અને ઉકેલ ખૂબ જ પારદર્શક છે અને આવા કિસ્સાઓમાં પણ ઉપર જણાવેલ પગલાંને અનુસરવાનો પ્રયાસ કરો વ્યવસ્થિત અભિગમને અનુસરવાની પ્રેક્ટિસ બનાવવા માટે. જ્યારે પણ પીસીમાં કોઈ સમસ્યા આવે છે, ત્યારે આને અનુસરો દ્વારા સિસ્ટમ વિશેની માહિતી એકત્રિત કરવાનો પ્રયાસ કરો જેવા પ્રશ્નોના જવાબ આપે છે

- કયા સંજોગોમાં આ સમસ્યા ઉભી થઈ?
- શું બીપ કોડના સ્વરૂપમાં કોઈ સંકેતો હતા/ ભૂલ સંદેશાઓ અથવા કોઈ સ્પષ્ટ લક્ષણો?
- જ્યારે સમસ્યા આવી ત્યારે કયા સોફ્ટવેર સક્રિય હતા- પેન ક્યું?
- શું તે પ્રથમ વખત બન્યું છે અથવા ભૂતકાળમાં બન્યું છે પણ?
- શું તે દરમિયાન કોઈ રૂપરેખાંકન ફેરફારો કરવામાં આવ્યા હતા સત્ર કે જેને પુનઃપ્રારંભની જરૂર હતી જે ન હતી ક્યું?

પીસીને ઓપ્ટિમાઇઝ કરવું:

એક પીસી જે એકદમ સારી રીતે કામ કરી રહ્યું હતું અને વિકસિત હતું ધીમું થવાનું લક્ષણ અથવા જો તે ગતિ જાળવી રાખવામાં અસમર્થ છે નવા સોફ્ટવેરની માંગ સાથે, શક્ય પૈકી એક ઉકેલો એ છે કે પીસીને અપડેટ અથવા ઓપ્ટિમાઇઝ કરવાનું વિચારવું તેની કામગીરીમાં વધારો. ઓપ્ટિમાઇઝેશન પગલાં ખર્ચ થઈ શકે છે પૈસા , પરંતુ ઘણા સોફ્ટવેરનો સમાવેશ કરે છે જે તમે પહેલાથી જ ધરાવો છો અથવા સોફ્ટવેર વેબ પર સરળતાથી ઉપલબ્ધ છે.

BIOS અને બુટ પ્રક્રિયાને ઓપ્ટિમાઇઝ કરી રહ્યા છીએ:

BIOS સેટઅપ ગોઠવણીમાં ઘણી સેટિંગ્સ શામેલ છે CMOS. સિસ્ટમ કેટલી ઝડપથી બુટ થાય છે અને કાર્ય કરે છે આ સેટિંગ્સ પર આધાર રાખે છે. મૂલ્યવાનને સક્ષમ કરવું સિસ્ટમ કેશીંગ અથવા ઝડપી ઉપયોગ જેવી સુવિધાઓ શ્રેષ્ઠ કામગીરી માટે પોસ્ટ પ્રક્રિયા ખૂબ જ મહત્વપૂર્ણ છે.

હાર્ડ ડિસ્કને ઓપ્ટિમાઇઝ કરવું: વિન્ડોઝ સ્કેનડિસ્ક અને ડિસ્ક ડિફ્રેગમેન્ટર ઉપયોગિતાઓ માટે ઉપલબ્ધ શ્રેષ્ઠ સાધનો છે ઉપયોગ અને એક્સેસની દ્રષ્ટિએ હાર્ડ ડિસ્કને ઓપ્ટિમાઇઝ કરવું ઝડપ સ્કેનડિસ્કનો

ઉપયોગ ભૂલો માટે ડિસ્ક તપાસવા માટે થાય છે તેમને રિપેર કરો અથવા ડિસ્કના પુનઃપ્રાપ્ત ન થઈ શકે તેવા વિસ્તારોને દૂર કરો ભાવિ ભૂલોને રોકવા માટે વપરાશ કોષ્ટકોમાંથી. તેવી જ રીતે ડિસ્ક ડિફ્રેગમેન્ટર ડેટા ફાઈલના ટુકડાને એમાં ગોઠવે છે વધુ ઓપ્ટિમાઇઝ અને લોજિકલ ફોર્મેટ જે ઝડપી એક્સેસ ટાઈમ અને ઓછી હેડ મૂવમેન્ટ પ્રદાન કરે છે.

વિસ્તરણ કાર્ડને ઓપ્ટિમાઇઝ કરવું: શ્રેષ્ઠ માર્ગ ઓપ્ટિમાઇઝ I/O નિયંત્રકો અને અન્ય વિસ્તરણ કાર્ડ માટે છે તેમને યોગ્ય ક્રમમાં સ્થાપિત કરો. જો તેઓ ઓર્ડરની બહાર ઈન્સ્ટોલ કરવામાં આવ્યા હોય તો પણ કોઈ નુકસાન થતું નથી, પરંતુ તેનો થોડો ફાયદો છે તેમને યોગ્ય ક્રમમાં મૂકવાથી મેળવી શકાય છે. એના પર પેન્ટિયમ સિસ્ટમ, PCI કાર્ડનો ઉપયોગ કરો અને ISA કાર્ડ ટાળો, જો શક્ય. વિડિયો કાર્ડ્સ, સાઉન્ડ સહિત તમામ I/O એડેપ્ટર કાર્ડ્સ, NICs, મોડેમ અને SCDS એડેપ્ટરો માટે ઉપલબ્ધ છે PCI વિસ્તરણ બસ. મધરબોર્ડ દસ્તાવેજની સલાહ લો- પ્રથમ પીસીઆઈ સ્લોટમાં વિડિયો કાર્ડનો ઉલ્લેખ કરો અને ઈન્સ્ટોલ કરો, તે ક્રમમાં NIC, મોડેમ અને સાઉન્ડ કાર્ડ દ્વારા અનુસરવામાં આવે છે.

પ્રક્રિયાને ઓપ્ટિમાઇઝ કરવું: તમે નીચેની રીતે પ્રોસેસરને ઝડપી બનાવી શકો છો

- તેને ઝડપી ગતિ અથવા ઉચ્ચ સ્તરના પ્રોસેસરથી બદલો
- પ્રોસેસરના તર્કમાં પેચો અથવા ફિક્સેસ લાગુ કરવા માટે પ્રોસેસર ઉત્પાદકની ઉપયોગિતાનો ઉપયોગ કરો
- પ્રોસેસરને ઓવરક્લોક કરો

પ્રોસેસરને ઉચ્ચ સાથે બદલવાની આવશ્યકતા લેવલ અથવા ઝડપી પ્રોસેસર એ છે કે તમારું મધરબોર્ડ અને ચિપસેટ નવા પ્રોસેસરને તાર્કિક અને બંને રીતે સપોર્ટ કરશે શારીરિક રીતે તાર્કિક રીતે ચિપસેટ અને મધરબોર્ડ આવશ્યક છે પ્રોસેસરની બસ સ્પીડને સપોર્ટ કરો અને હોય તેને જરૂરી સહાયક સર્કિટ. તે ઘણી વખત વધુ સારું છે સંપૂર્ણ મધરબોર્ડ બદલો. કેટલીક વખત મનુ- પ્રોસેસરની ફેક્ટરર કેટલીક ઉપયોગિતાઓને રિલીઝ કરી શકે છે જે પ્રોસેસરની ક્ષમતાઓના કેટલાક પાસાઓને સુધારશે જેમ કે વિડિઓ પ્રોસેસિંગ, બફર હેન્ડલિંગ, કેશીંગ અને અન્ય તરીકે પ્રોસેસર આધારિત કાર્યો. પ્રોસેસર ઓવરક્લોકિંગ મતલબ કે પ્રોસેસરને તે હતી તેના કરતા વધુ ઝડપે ચલાવવું ટેકો આપવા માટે પ્રકાશિત. મોટાભાગના પ્રોસેસરો સક્ષમ છે તેમની નજીવી ઝડપ કરતાં વધુ ઝડપે દોડે છે. આ પ્રોસેસરની નજીવી ઝડપ એ ઝડપ છે જે તેની પાસે છે ચોક્કસ ચિપસેટ, મધરબોર્ડ સાથે ચલાવવા માટે ટ્યુન કરવામાં આવ્યું છે, ફૂલિંગ સિસ્ટમ અને પીસીના અન્ય ઘટકો. ઉછેર પ્રોસેસરની ઝડપ હીટિંગ બનાવી શકે છે પર સમસ્યાઓ પ્રોસેસર અને ફ્રીક્વન્સીસિસ્ટમ લોકઅપ્સ, મેમરી સમસ્યાઓ અને અન્ય સમસ્યાઓ તરફ દોરી જાય છે.

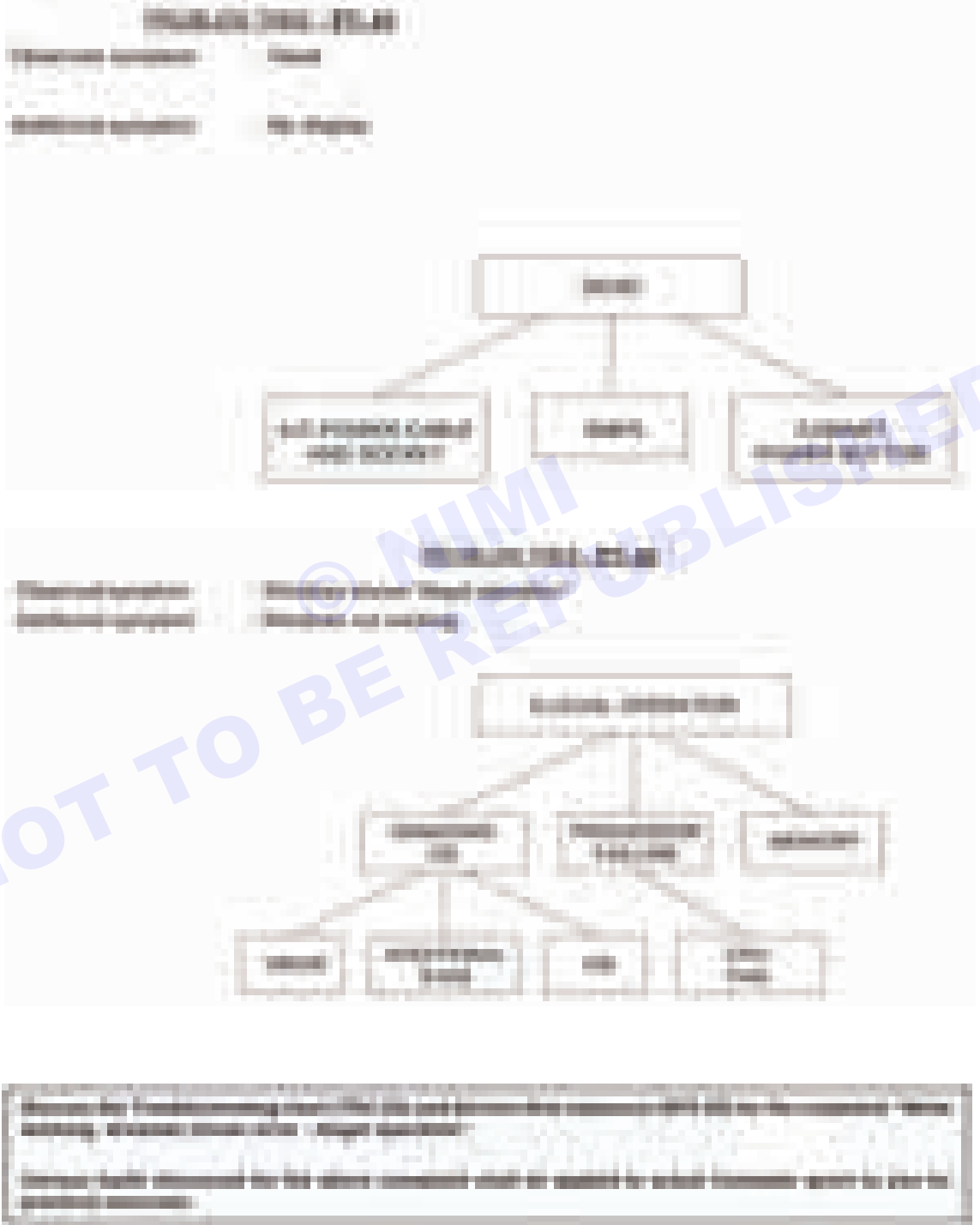
બિન-સોફ્ટવેર સમસ્યાઓના મુશ્કેલીનિવારણ સ્ત્રોતો:

કોઈપણ સમયે કોઈ દેખીતા સોફ્ટવેર કારણ વગર પીસી નિષ્ફળ જાય, નીચેના વિસ્તારો તપાસો

- યોગ્ય AC પાવરની ખાતરી કરો
- કમ્પ્યુટર વાયરસ માટે પીસી સ્કેન કરો
- ખાતરી કરો કે CPU ફેન સ્પિનિંગ કરી રહ્યો છે
- બાહ્ય I/O કનેક્ટર્સના યોગ્ય જોડાણોની ખાતરી કરો
- વિસ્તરણ કાર્ડને ફરીથી સેટ કરો અને પાવર તપાસો અને આંતરિક ઉપકરણોના ડેટા કેબલ્સ

- મોટાભાગની બુટ સમસ્યાઓ તાજેતરનું પરિણામ છે બદલો, BIOS સેટઅપ રૂપરેખાંકન ડેટા તપાસો
- કોઈપણ નવા હાર્ડવેર અથવા સોફ્ટવેરને ઇન્સ્ટોલ કરવા માટે, મેન્યુફેક્ચરની મુલાકાત લો
- કોઈપણ જાણીતા સંઘર્ષ અથવા અસંગતતા માટે ટ્યુરરની વેબ સાઈટ
- જો કોઈ નવું હાર્ડવેર અથવા કોઈપણ સંસાધન સંઘર્ષ હોય તો તપાસો સોફ્ટવેર ઇન્સ્ટોલ કરેલ છે.

મુશ્કેલીનિવારણ PC પરના આગામી પાઠ છે વિવિધ પ્રકારની સમસ્યાઓ માટે પ્રોબ્લેમ ટ્રી આપવામાં આવે છે જેનો વપરાશકર્તા સામાન્ય રીતે સામનો કરે છે. ચોક્કસ સમસ્યા સાથેનું દરેક વૃક્ષ શંકાસ્પદ વિસ્તારોનું વિશ્લેષણ કરવાનો અવકાશ આપે છે અથવા સમસ્યાને ઠીક કરવા માટે તપાસ કરી. આ પાઠનો સમાવેશ થાય છે પીસી માટે પ્રોબ્લેમ ટ્રી કે જે મોનિટર પર ડિસ્પ્લે વગર મૃત છે.





COPA - કમ્પ્યુટર બેઝિક્સ અને સોફ્ટવેર ઇન્સ્ટોલેશન

Windows માટે જરૂરી એપ્લિકેશન સોફ્ટવેર ઇન્સ્ટોલ કરો એટલે કે ઓફિસ પેકેજ, PDF રીડર, મીડિયા પ્લેયર વગેરે (Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- વિવિધ સોફ્ટવેર પ્રકારોનું વર્ણન કરો.

એપ્લિકેશન સોફ્ટવેર

એપ્લિકેશન સોફ્ટવેર એ એક શબ્દ છે જેનો ઉપયોગ સોફ્ટવેર માટે થાય છે ચોક્કસ હેતુ માટે બનાવેલ છે. તે સામાન્ય રીતે એક પ્રોગ્રામ છે અથવા અંતિમ વપરાશકર્તાઓ દ્વારા ઉપયોગમાં લેવાતા કાર્યક્રમોનો સંગ્રહ. તે હોઈ શકે છે એપ્લિકેશન અથવા ફક્ત એપ્લિકેશન કહેવાય છે.

હકીકતમાં સિસ્ટમ સોફ્ટવેર સિવાયના તમામ સોફ્ટવેર અને પ્રોગ્રામિંગ સોફ્ટવેર એપ્લિકેશન સોફ્ટવેર છે.

એપ્લિકેશન સોફ્ટવેર વ્યાખ્યા

એક સોફ્ટવેર જે વપરાશકર્તાને કાર્ય કરવામાં મદદ કરવા માટે વિકસાવવામાં આવ્યું છે ચોક્કસ કાર્યોને એપ્લિકેશન સોફ્ટવેર કહેવામાં આવે છે.

વિવિધ પ્રકારના એપ્લિકેશન સોફ્ટવેરમાં નીચેનાનો સમાવેશ થાય છે:

એપ્લિકેશન સોફ્ટવેર પ્રકાર	ઉદાહરણો
વર્ડ પ્રોસેસિંગ સોફ્ટવેર	એમએસ વર્ડ, વર્ડપેડ અને નોટપેડ
ડેટાબેઝ સોફ્ટવેર	ઓરેકલ, એમએસ એક્સેસ વગેરે
સ્પ્રેડશીટ સોફ્ટવેર	એપલ નંબર્સ, માઈક્રોસોફ્ટ એક્સેલ
મલ્ટીમીડિયા સોફ્ટવેર	મલ્ટીમીડિયા સોફ્ટવેર
પ્રસ્તુતિ સોફ્ટવેર	માઈક્રોસોફ્ટ પાવરપોઈન્ટ, કીનોટ્સ
એન્ટરપ્રાઇઝ સોફ્ટવેર	ગ્રાહક સંબંધ વ્યવસ્થાપન સિસ્ટમ
માહિતી કાર્યકર સોફ્ટવેર	દસ્તાવેજીકરણ સાધનો, સંસાધન સંચાલન સાધનો
શૈક્ષણિક સોફ્ટવેર	શબ્દકોશો: એન્કાર્ટા, બ્રિટાનીકા મેથેમેટિકલ: MATLAB Others: Google Earth, NASA World Wind
સિમ્યુલેશન સોફ્ટવેર	ફ્લાઈટ અને વૈજ્ઞાનિક સિમ્યુલેટર
સામગ્રી એક્સેસ સોફ્ટવેર બ્રાઉઝર્સ	મીડિયા પ્લેયર, વેબ દ્વારા સામગ્રીને એક્સેસ કરવી
એપ્લિકેશન સ્યુટ્સ	ઓપનઓફિસ, માઈક્રોસોફ્ટ ઓફિસ
એન્જિનિયરિંગ અને ઉત્પાદન વિકાસ માટે સોફ્ટવેર	સંકલિત વિકાસ પર્યાવરણ માટે IDE

લાયસન્સ, વેચાણ, ફ્રીવેર, શેરવેર અને ઓપન સોર્સ જેવા વિવિધ પ્રકારના એપ્લિકેશન સોફ્ટવેર છે.

એપ્લિકેશન સોફ્ટવેરને કાં તો ઇન્સ્ટોલ કરવાની જરૂર છે અથવા કરી શકો છો ઓનલાઈન ચલાવો. એપ્લિકેશન સોફ્ટવેરને પણ ઓળખી શકાય છે નીચેનામાં ઉપયોગના આધારે:

- યુટીલીટી પ્રોગ્રામ્સ
- ગેનેરિક પ્રોગ્રામ્સ
- ઈટેગ્રાટેડ પ્રોગ્રામ્સ
- સ્પેસિફિક સોફ્ટવેર
- બેસપોકે સોફ્ટવેર
- વર્ડ પ્રોસેસિંગ સોફ્ટવેર

- ડેસ્કટોપ પ્રકાશન સોફ્ટવેર
- સ્પ્રેડશીટ સોફ્ટવેર
- ડેટાબેઝ સોફ્ટવેર
- પ્રસ્તુતિ સોફ્ટવેર
- ઈન્ટરનેટ બ્રાઉઝર્સ
- ઈમેલ પ્રોગ્રામ્સ
- ગ્રાફિક પ્રોગ્રામ્સ (પિક્સેલ આધારિત)
- ગ્રાફિક પ્રોગ્રામ્સ (વેક્ટર આધારિત)
- કોમ્યુનિકેશન સોફ્ટવેર: દ્વારા કોમ્યુનિકેશન ઓડિયો, વિડિયો અથવા ચેટ આધારિત માધ્યમ.

બ્લૂટૂથ અને Wi-Fi સેટિંગ્સને ગોઠવો (Configure Bluetooth and Wi-Fi settings)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- બ્લૂટૂથના અર્થનું વર્ણન કરો
- બ્લૂટૂથનો ઉપયોગ કરવાની પદ્ધતિનું વર્ણન કરો
- બ્લૂટૂથની મુખ્ય એપ્લિકેશનોની યાદી બનાવો.

પરિચય:

બ્લૂટૂથ માટે વાયરલેસ ટેકનોલોજી સ્ટાન્ડર્ડ છે ટૂંકા અંતર પર ડેટાની આપલે કરવી (ટૂંકાનો ઉપયોગ કરીને 2.4 થી 2.485 રેન્જમાં તરંગલંબાઈ UHF રેડિયો તરંગો GHz) ફ્રિક્વન્સી અને મોબાઈલ ડિવાઈસ અને બિલ્ડિંગમાંથી પર્સનલ એરિયા નેટવર્ક (PAN) તે પ્રમાણભૂત વાયર છે- રિપ્લેસમેન્ટ કમ્યુનિકેશન પ્રોટોકોલ મુખ્યત્વે ઓછા-પાવર વપરાશ માટે રચાયેલ છે, જેમાં દરેક ઉપકરણમાં ઓછી કિંમતના ટ્રાન્સમિટ્ટર માઈક્રોચિપ્સ પર આધારિત ટૂંકી શ્રેણી છે.

તે આઠ જેટલા ઉપકરણોને કનેક્ટ કરી શકે છે (ઇલેક્ટ્રોનિક વસ્તુઓ સાધનો) તે જ સમયે. ચિપ પ્લગ કરી શકાય છે કમ્પ્યુટર, ડિજિટલ કેમેરા, મોબાઈલ જેવી વસ્તુઓમાં ફોન અને ફેક્સ. બ્લૂટૂથ ખાસ કરીને અનુકૂળ છે અમુક પરિસ્થિતિઓમાં - ઉદાહરણ તરીકે, ફાઈલો સ્થાનાંતરિત કરતી વખતે એક મોબાઈલ ફોનથી બીજા મોબાઈલ ફોનમાં કેબલ વગર. પીસી અને મોબાઈલ વચ્ચે સંગીત અને ફોટા મોકલવા ફોન અન્ય ઉપયોગી એપ્લિકેશન છે.

કારણ કે ઉપકરણો રેડિયોનો ઉપયોગ કરે છે (પ્રસારણ) સંદેશાવ્યવહાર સિસ્ટમ, તેઓ દ્રશ્યમાં હોવા જરૂરી નથી એકબીજાની દૃષ્ટિની રેખા, જોકે અર્ધ ઓપ્ટિકલ વાયરલેસ પાથ સધ્ધર હોવો જોઈએ.

શ્રેણી શક્તિ-વર્ગ આધારિત છે, પરંતુ અસરકારક શ્રેણી છે વ્યવહારમાં 10 થી 100 મીટર સુધી બદલાય છે.

‘બ્લૂટૂથ’ નામ સ્કેન્ડિનેવિયન મૂળને પ્રતિબિંબિત કરે છે ટેકનોલોજીની. તેનું નામ 10મી સદીના ડેનિશના નામ પરથી રાખવામાં આવ્યું છે વાઈકિંગ, કિંગ હેરાલ્ડ બ્લાટેન્ડ (‘બ્લૂટૂથ’ તરીકે ભાષાંતર અંગ્રેજી). તેમણે ડેનમાર્ક અને નોર્વેને એક કરીને નિયંત્રિત કર્યું, તેથી બ્લૂટૂથ દ્વારા ઉપકરણોને એકીકૃત કરવાનું જોડાણ.

બ્લૂટૂથનો ઉપયોગ

બ્લૂટૂથનો ઉપયોગ કરવા માટે, ઉપકરણ બ્લૂટૂથ સક્ષમ હોવું આવશ્યક છે. આ હેતુ માટે “ડોંગલ” નામના ઉપકરણનો ઉપયોગ કરી શકાય છે. ડોંગલ એ એક ઉપકરણ છે જે કમ્પ્યુટરમાં પ્લગ કરે છે તેને બ્લૂટૂથ વાપરવા માટે સક્ષમ કરો. ના દરેક ઉત્પાદક સુસંગત ઉપકરણો માટે તેમની પોતાની સૂચનાઓ હશે બ્લૂટૂથ એક્સેસ કરી રહ્યું છે. વિગતવાર સૂચનાઓ માટે તમે મેન્યુઅલ જોવાની જરૂર છે, પરંતુ સામાન્ય માર્ગદર્શિકા તરીકે:

બ્લૂટૂથ સેટ કરવા માટે:

(ઉપકરણો પર બ્લૂટૂથ આયકન ઓળખો.)

- 1 Bluetooth ચાલુ કરો અથવા સક્ષમ કરો. ખાતરી કરો કે તમારું ઉપકરણ છે ‘દૃશ્યમાન’ અને ‘છુપાયેલ’ નથી, જેથી અન્ય નજીકના ઉપકરણો કરી શકે સિગ્નલ ઉપાડો.

- 2 તમારા ઉપકરણને અન્ય સુસંગત સાધનો સાથે કનેક્ટ કરતી વખતે તેને ઓળખવા માટે એક નામ આપો.

જ્યારે મોબાઈલ ફોન, લેપટોપ, ટેબલેટ વગેરે જેવા ઉપકરણો. Bluetooth સક્ષમ કરો બ્લૂટૂથ આઈકન ચાલુ છે.

બ્લૂટૂથ કનેક્શન સ્થાપિત કરવા માટે:

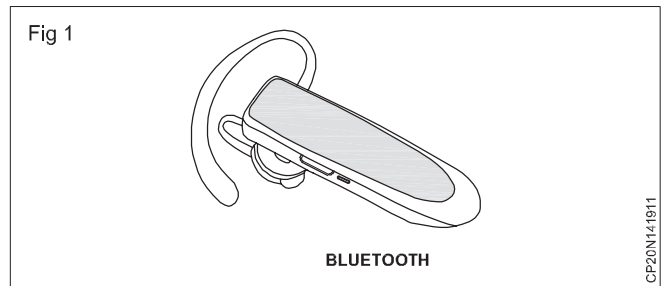
- 1 તમે મોકલવા માંગો છો તે ફાઈલ શોધો.
- 2 બ્લૂટૂથ દ્વારા મોકલવાનો વિકલ્પ પસંદ કરો - તમારું ઉપકરણ શ્રેણીમાં અન્ય ઉપકરણોની શોધ કરશે અને તેમને પ્રદર્શિત કરશે.
- 3 તમે જેની સાથે જોડાવા માંગો છો તે ઉપકરણ પર સ્ક્રોલ કરો અને તેને પસંદ કરો.
- 4 જો અન્ય ઉપકરણને ‘પેયરિંગ’ની જરૂર હોય, તો તમારે કરવાની જરૂર પડશે પાસવોર્ડ દાખલ કરો - થોડો પિન નંબર જેવો - અને બનાવો ખાતરી કરો કે તે અન્ય ઉપકરણ પર દાખલ થયેલ છે.

જ્યારે કનેક્શન સ્થાપિત થાય છે, ત્યારે ડેટા શરૂ થશે મોકલો. તમારે ઉપકરણો વચ્ચેની સ્પષ્ટ રેખા વિશે ચિંતા કરવાની જરૂર નથી.

બ્લૂટૂથ એપ્લિકેશન્સની સૂચિ

કેટલીક બ્લૂટૂથ એપ્લિકેશનો નીચે મુજબ છે:

મોબાઈલ વચ્ચે વાયરલેસ નિયંત્રણ અને સંચાર ફોન અને હેન્ડસફ્રી હેડસેટ. (ફિગ. 1 નો સંદર્ભ લો)

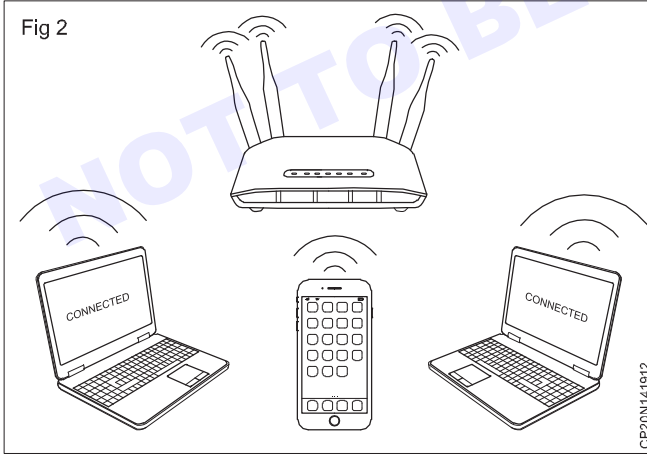


- વાયરલેસ નિયંત્રણ અને એ વચ્ચે સંચાર મોબાઈલ ફોન અને બ્લૂટૂથ સુસંગત કાર સ્ટીરિયો સિસ્ટમ
- ટેબલેટ સાથે વાયરલેસ નિયંત્રણ અને સંચાર અને સ્પીકર્સ જેમ કે iPad અને Android ઉપકરણો.
- મર્યાદિત જગ્યામાં PC વચ્ચે વાયરલેસ નેટવર્કિંગ અને જ્યાં થોડી બેન્ડવિડ્થ જરૂરી છે.
- પીસી ઈનપુટ અને આઉટપુટ સાથે વાયરલેસ સંચાર ઉપકરણો, સૌથી સામાન્ય છે માઉસ, કીબોર્ડ અને પ્રિન્ટર.

- ફાઇલો, સંપર્ક વિગતો, કેલેન્ડરનું ટ્રાન્સફર સાથેના ઉપકરણો વચ્ચે એપોઇન્ટમેન્ટ અને રીમાઇન્ડર્સ OBEX(ઓબ્જેક્ટ એક્સચેન્જ).
- અગાઉના વાયર્ડ RS-232 સીરીયલનું ફેરબદલ પરીક્ષણ સાધનોમાં સંચાર, જીપીએસ રીસીવર, તબીબી સાધનો, બાર કોડ સ્કેનર્સ અને ટ્રાફિક નિયંત્રણ ઉપકરણો.
- બે ઔદ્યોગિક ઈથરનેટ વચ્ચે વાયરલેસ બ્રિજ નેટવર્ક્સ • ગેમિંગ કન્સોલમાં વાયરલેસ નિયંત્રકો.
- માટે મોબાઇલ ફોન પર વ્યક્તિગત સુરક્ષા એપ્લિકેશન ચોરી અથવા વસ્તુઓની ખોટ અટકાવવી. રક્ષિત આઇટમમાં બ્લૂટૂથ માર્કર (દા.ત., ટેગ) છે જે અંદર છે ફોન સાથે સતત વાતચીત. જો કનેક્શન તૂટી ગયું છે (માર્કર ની શ્રેણીની બહાર છે ફોન) પછી એલાર્મ ઊભો થાય છે.

Wi-Fi:

Wi-Fi એ એક લોકપ્રિય વાયરલેસ નેટવર્કિંગ ટેકનોલોજી છે. Wi-Fi નો અર્થ “વાયરલેસ ફ્રીડેલિટી” છે. Wi-Fi ની શોધ થઈ 1991માં નેધરલેન્ડમાં એનસીઆર કોર્પોરેશન/એટી એન્ડ ટી દ્વારા. દ્વારા આ ટેકનોલોજીનો ઉપયોગ કરીને આપણે માહિતીની આપ-લે કરી શકીએ છીએ બે અથવા વધુ ઉપકરણો વચ્ચે. Wi-Fi વિકસાવવામાં આવ્યું છે મોબાઇલ કમ્યુટિંગ ઉપકરણો માટે, જેમ કે લેપટોપ છે, પરંતુ તે છે હવે મોબાઇલ એપ્લિકેશન માટે વ્યાપકપણે ઉપયોગ થાય છે અને કન્ઝ્યુમર ઇલેક્ટ્રોનિક્સ જેમ કે ટેલિવિઝન, ડીવીડી પ્લેયર અને ડિજિટલ કેમેરા. માં બે શક્યતાઓ હોવી જોઈએ Wi-Fi કનેક્શન સાથે સંચાર કે જે હોઈ શકે છે ક્લાયન્ટ કનેક્શન અથવા ક્લાયન્ટને એક્સેસ પોઇન્ટ દ્વારા ક્લાયન્ટ કનેક્શન. Wi-Fi એ એક પ્રકારનો વાયરલેસ છે ટેકનોલોજી તેને સામાન્ય રીતે વાયરલેસ LAN (સ્થાનિક વિસ્તાર નેટવર્ક). Wi-Fi લોકલ એરિયા નેટવર્કને પરવાનગી આપે છે કેબલ અને વાયરિંગ વગર ચલાવો. તે લોકપ્રિય બની રહ્યું છે ઘર અને વ્યવસાય નેટવર્ક માટે પસંદગી. કમ્પ્યુટરનું વાયરલેસ એડેપ્ટર કેટાને રેડિયો સિગ્નલમાં સ્થાનાંતરિત કરે છે અને વપરાશકર્તાઓ માટે કેટાને એન્ટેનામાં સ્થાનાંતરિત કરે છે.



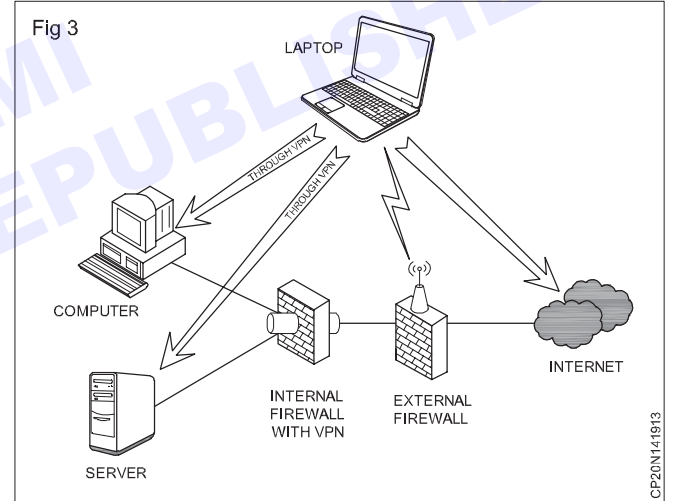
કાર્ય સિદ્ધાંત:

Wi-Fi એ હાઇ સ્પીડ ઇન્ટરનેટ કનેક્શન અને નેટવર્ક છે કોઈપણ કેબલ અથવા વાયરના ઉપયોગ વિના જોડાણ. આ વાયરલેસ નેટવર્ક ત્રણ આવશ્યક તત્વોનું સંચાલન કરે છે જે રેડિયો સિગ્નલ, એન્ટેના અને રાઉટર છે. રેડિયો તરંગો એ કી છે જે Wi-Fi નેટવર્કિંગ બનાવે છે શક્ય. કમ્પ્યુટર અને સેલ ફોન સાથે તૈયાર છે Wi-Fi કાર્ડ્સ. Wi-Fi સુસંગતતા

એક નવો ઉપયોગ કરી રહી છે સાથે જોડાયેલ જમીનની અંદરના ઘટકની રચના સમુદાય નેટવર્ક. વાસ્તવિક પ્રસારણ જોડાયેલ છે ક્રમમાં હકીકતમાં તે સ્ટીરિયો દ્વારા પૂર્ણ થાય છે સિસ્ટમ સર્ફ તેમજ મોનિટર સાથે વાયરની કિંમત વર્ગીકરણની સંભાવના. Wi-Fi વ્યક્તિને ક્રમમાં પરવાનગી આપે છે વાસ્તવિક પ્રદાન કરેલ વિસ્તારમાં કોઈપણ જગ્યાએ વેબની એક્સેસ મેળવો.

અને હવે રિસોર્સ, લાઇબ્રેરીમાં સિસ્ટમ જનરેટ કરી શકે છે. શાળાઓ, કોલેજો, કેમ્પસ, વ્યક્તિગત સંસ્થાઓ, તેમજ એસ્પ્રેસો સ્ટોર્સ તેમજ ખુલ્લા જાહેર સ્થળ પર કંપનીને વધુ નફાકારક બનાવવામાં પણ મદદ કરે છે જ્યારે પણ તેમના પોતાના ગ્રાહક સાથે સંપર્ક કરો. Wi-Fi સુસંગતતાનો ઉપયોગ કરીને કંપનીને તાકીને સર્ફ કરી શકે છે તેમના પ્રેરણાદાયક કેબલ ટેલિવિઝન ઘણી ઓછી માત્રામાં છે.

રેડિયો સિગ્નલો એન્ટેનાથી પ્રસારિત થાય છે અને Wi-Fi રીસીવરો દ્વારા સિગ્નલ લેવામાં આવતા રાઉટર્સ, જેમ કે કોમ્પ્યુટર અને સેલ ફોન સાથે તૈયાર છે Wi-Fi કાર્ડ્સ. જ્યારે પણ કોમ્પ્યુટર સિગ્નલ મેળવે છે રાઉટર માટે 100-150 ફીટની રેન્જમાં તે કનેક્ટ કરે છે તરત જ ઉપકરણ. Wi-Fi ની શ્રેણી આધાર રાખે છે પર્યાવરણ પર, ઇન્ડોર અથવા આઉટડોર રેન્જ પર. આ Wi-Fi કાર્ડ સિગ્નલો વાંચશે અને ઇન્ટરનેટ બનાવશે વપરાશકર્તા અને નેટવર્ક વચ્ચે જોડાણ. ની ઝડપ Wi-Fi કનેક્શનનો ઉપયોગ કરતા ઉપકરણ કમ્પ્યુટર તરીકે વધે છે મુખ્ય સ્ત્રોતની નજીક જાય છે અને ઝડપ ઘટે છે કમ્પ્યુટર વધુ દૂર જાય છે.



સુરક્ષા:

વાઇ-ફાઇ ટેકનોલોજીમાં સુરક્ષા એ એક ઘટક છે. સુરક્ષા એ અમારો વ્યક્તિગત નિર્ણય છે પરંતુ વાયરલેસ હોવો જોઈએ જોડાણ અમે અમારા ખાનગી રક્ષણ માટે ધ્યાન આપવું જોઈએ વિગતો અમે અસુરક્ષિત વાયરલેસ સાથે સરળતાથી કનેક્ટ થઈ શકીએ છીએ રાઉટર્સ સમસ્યા એ છે કે કોઈપણ તેની સાથે જોડાયેલ છે ડાઉનલોડ ગેમ્સ જેવા કેટાનો ઉપયોગ કરીને વાયરલેસ રાઉટર, એપ્લિકેશન્સ ડાઉનલોડ કરો અને આતંકવાદી પ્રવૃત્તિઓનું આયોજન કરો, શિરિંગ કરો ગેરકાયદેસર સંગીત અને મૂવી ફાઇલો વગેરે. તેથી તે જરૂરી છે વાયરલેસ ટેકનોલોજી આધારિત ઉપકરણોને સુરક્ષા પ્રદાન કરે છે.

ડીવીડી, સીડી અને બર્નિંગ ડીવીડી (DVDs, CDs and burning DVDs)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સીડી અને ડીવીડીની વિશેષતાઓનું વર્ણન કરો
- ડીવીડીના મુખ્ય ફોર્મેટનું વર્ણન કરો ડીવીડી બર્નિંગનું વર્ણન કરો
- સીડીની વિશેષતાઓનું વર્ણન કરો.

પરિચય:

ડીવીડી (ક્યારેક “ડિજિટલ વિડિયો ડિસ્ક” અથવા “ડિજિટલ તરીકે ઓળખાય છે બહુમુખી ડિસ્ક”) એ ડિજિટલ ઓપ્ટિકલ ડિસ્ક સ્ટોરેજ ફોર્મેટ છે. ડીવીડીનો ઉપયોગ પીસી અને સ્ટેન્ડઅલોન પ્લેયર્સ સહિત ઘણા પ્રકારના પ્લેયર્સ સાથે કરી શકાય છે.

આ ડિસ્કને DVD-ROM તરીકે ઓળખવામાં આવે છે, કારણ કે ડેટા કરી શકે છે માત્ર વાંચી શકાય અને લખવામાં કે ભૂંસી ન શકાય. ખાલી રેકોર્ડ કરવા યોગ્ય DVD ડિસ્ક (DVD-R અને DVD+R) એકવાર રેકોર્ડ કરી શકાય છે ડીવીડી રેકોર્ડરનો ઉપયોગ કરો અને પછી ડીવીડી-રોમ તરીકે કાર્ય કરો. ફરીથી લખી શકાય તેવી DVD (DVD-RW, DVD+RW, અને DVD-RAM) ઘણી વખત રેકોર્ડ અને ભૂંસી શકાય છે.

ડીવીડી સુવિધાઓ અને બંધારણો

DVD-Video કન્ઝ્યુમર ડિજિટલ વિડિયોમાં DVD નો ઉપયોગ થાય છે ફોર્મેટ અને ડીવીડી-ઓડિયો કન્ઝ્યુમર ડિજિટલ ઓડિયો ફોર્મેટમાં. તેનો ઉપયોગ ખાસ AVCHD ફોર્મેટમાં પણ થઈ શકે છે (અદ્યતન વિડિયો કોડિંગ હાઈ ડેફિનેશન) સાથે વારંવાર ઉપયોગ થાય છે AVCHD ફોર્મેટ કેમકોર્ડર્સ. અન્ય પ્રકારો ધરાવતી ડીવીડી માહિતીની ડીવીડી ડેટા ડિસ્ક તરીકે ઉલ્લેખ કરી શકાય છે.

સામાન્ય રેકોર્ડેબલ ડીવીડી લગભગ 4.7 ગીગાબાઈટ્સ ધરાવી શકે છે (GB). જો કે, ડિસ્ક જગ્યાનો કુલ જથ્થો કે જે તમે ડિસ્ક પર ફાઈલો બર્ન કરવા માટે ઉપયોગ કરી શકો છો રકમ કરતાં ઓછી છે જે ઘણીવાર ડિસ્ક પર જ સૂચિબદ્ધ હોય છે. આ એટલા માટે છે કારણ કે ડિસ્કની ક્ષમતા અલગ રીતે ગણવામાં આવે છે જ્યારે તેનો a માં ઉપયોગ થાય છે કમ્પ્યુટર ઉદાહરણ તરીકે, એક લાક્ષણિક ડીવીડી-આર જેમાં સૂચિબદ્ધ છે 4.7 GB ની ડિસ્ક ક્ષમતા ફક્ત 4.37 GB નો જ સંગ્રહ કરી શકે છે ડિસ્ક પરનો ડેટા. ડીવીડી ની સ્ટોરેજ ક્ષમતા પ્રદાન કરે છે આશરે 4.7 જીબી. ડીવીડી ડિસ્ક વધુ બગડતી નથી સમય અને ચુંબકીય ક્ષેત્રોથી પ્રભાવિત નથી.

ઉપયોગમાં લેવાતી રેકોર્ડેબલ ડિસ્કનો પ્રકાર કેટલાક વિવિધ પરિબલો પર આધાર રાખે છે, જેમ કે:

- રેકોર્ડ કરી શકાય તેવી ડિસ્કના પ્રકારો જે ડિસ્કબર્નર સાથે કામ કરે છે.
- કોમ્પ્યુટર અથવા ઉપકરણ પરની ડિસ્ક ડ્રાઈવ ડિસ્કને બર્ન કર્યા પછી વાંચશે.
- બધી ફાઈલોનું કુલ કદ જે ડિસ્ક પર બર્ન થશે.

ડીવીડી બર્નિંગ : ઓપ્ટિકલ પર સ્ત્રોત સામગ્રીને રેકોર્ડ કરવાની પ્રક્રિયા ડિસ્ક (સીડી અથવા ડીવીડી)ને બર્નિંગ/રાઈટિંગ અથવા ઓપ્ટિકલ કહેવામાં આવે છે ડિસ્ક ઓથરીંગ. ઓપ્ટિકલ ડિસ્ક બનાવવા માટે સામાન્ય રીતે સમાવેશ થાય છે પ્રથમ ડિઝાઈન કરેલી સંપૂર્ણ ફાઈલ સિસ્ટમ સાથે

ડિસ્ક ઇમેજ બનાવો ઓપ્ટિકલ ડિસ્ક માટે, અને પછી ખરેખર ઇમેજ બર્નિંગ ડિસ્ક માટે. ડિસ્ક ઇમેજ બનાવવા અને ફાઈલો બર્ન કરવા માટે ઘણા પ્રોગ્રામ બંડલ એપ્લિકેશન તરીકે ઉપલબ્ધ છે.

ડીવીડી લખી શકાય તે ઝડપે વ્યક્ત કરવામાં આવે છે ગુણક તરીકે: 16X એટલે માત્ર કરતાં 16 ગણી ઝડપી તેને વગાડવું. 5 2X સુધીની ઝડપ પણ ખૂબ સામાન્ય છે.

સીડી અથવા ડીવીડી ફોર્મેટ

ડીવીડી બર્ન કરવા માટે, ઉપયોગમાં લેવાતા બે મુખ્ય ડિસ્ક ફોર્મેટ છે:

- લાઈવ ફાઈલ સિસ્ટમ અને
- માસ્ટર્ડ ડિસ્ક ફોર્મેટ.

લાઈવ ફાઈલ સિસ્ટમ ફોર્મેટ:

ડિસ્ક કે જે લાઈવ ફાઈલ સિસ્ટમ ફોર્મેટનો ઉપયોગ કરે છે તે ઘણી વખત હોય છે વધુ અનુકૂળ કારણ કે તમે પસંદ કરેલી ફાઈલોની નકલ કરી શકો છો તરત જ અને જેટલી વાર તમે ઇચ્છો, જેમ તમે કરી શકો યુએસબી ફ્લેશ ડ્રાઈવ સાથે. જો તમે ઇચ્છો તો આ અનુકૂળ છે તમારા સીડી, ડીવીડી અથવા બ્લુ રે ડિસ્ક બર્નરમાં ડિસ્ક રાખો અને જ્યારે પણ જરૂર પડે ત્યારે ફાઈલોની નકલ કરો. આ ફોર્મેટમાં તમે ફરીથી અને ફરીથી ફાઈલોની નકલ અને ભૂંસી શકે છે. જો કે, લાઈવ ફાઈલ સિસ્ટમ ઓપ્ટિકલ ડિસ્ક ફોર્મેટ ફક્ત વાંચી શકાય તેવું છે Windows 7, Windows Vista અને Windows XP દ્વારા સિસ્ટમો આ ડિસ્ક ફરીથી ફોર્મેટ કર્યા પછી ખાલી નથી.

આ ફોર્મેટમાં ફોર્મેટ કરાયેલ ડિસ્કમાં ડિસ્ક બર્ન અ ડિસ્ક સંવાદ બોક્સમાં વિકલ્પ નામ છે: “USB ફ્લેશ ડ્રાઈવની જેમ.”

માસ્ટર્ડ ડિસ્ક ફોર્મેટ્સ:

જો આપણે એક ઓપ્ટિકલ ડિસ્ક બનાવવા માંગીએ છીએ જેનો ઉપયોગ કરી શકાય વિન્ડોઝના જૂના સંસ્કરણોમાં ડેટા ફાઈલોને સ્થાનાંતરિત કરો ઓપરેટિંગ સિસ્ટમ અથવા તો બીજી ઓપરેટિંગ સિસ્ટમ માટે, માસ્ટર્ડ ઓપ્ટિકલ ડિસ્ક ફોર્મેટનો ઉપયોગ કરવાની જરૂર છે. પણ જો અમે સંગીત અથવા ચિત્રો બર્ન કરવા અને ડિસ્કનો ઉપયોગ કરવા માંગીએ છીએ નિયમિત સીડી, ડીવીડી અથવા બ્લુ રે ડિસ્ક પ્લેયર્સ જે પ્લે કરી શકે છે MP3 ફાઈલો અને ડિજિટલ ચિત્રો બતાવવા માટે, આપણે માસ્ટર્ડ ઓપ્ટિકલ ડિસ્ક ફોર્મેટનો ઉપયોગ કરવો જોઈએ.

માસ્ટર્ડ ફોર્મેટ સીડી બર્ન કરવા જેવું જ કામ કરે છે વિન્ડોઝ XP. બીજા શબ્દોમાં કહીએ તો, જ્યારે આપણે ડિસ્ક લખીએ છીએ, ત્યારે આપણે ઓપ્ટિકલ ડિસ્ક પર ફાઈલ અથવા ફાઈલોના જૂથની નકલ કરો એકવાર એકવાર આ થઈ જાય, ડિસ્ક બંધ થઈ જાય છે અને આપણે કરી શકતા નથી ડિસ્કમાં વધુ ફાઈલોની નકલ કરી શકીએ છીએ અને અમે હાલની ફાઈલોને કાઢી શકતા નથી ફાઈલો. તેથી ફાઈલોની નકલ ન કરવાની ભલામણ

કરવામાં આવે છે તરત; સમગ્રને એસેમ્બલ કરવાની સારી પ્રથા છે ડિસ્ક પર કોપી કરવાની જરૂર હોય તેવી ફાઇલોનો સંગ્રહ અને પછી તે બધાને એક જ સમયે બાળી નાખો.

માસ્ટર્સ વિકલ્પ સાથે ફોર્મેટ કરેલી ડિસ્કમાં બર્ન અ ડિસ્ક ડાયલોગ બોક્સમાં વિકલ્પ હોય છે: “સીડી/ડીવીડી પ્લેયર સાથે.”

ડેટા બનાવવા માટે ઘણા પ્રકારના સાધનો ઉપલબ્ધ છે, સંગીત, વિડિયો અને ઓડિયો ડિસ્ક. અમે બેકઅપ પણ બનાવી શકીએ છીએ જે બહુવિધ ડિસ્ક, રીપ મ્યુઝિક ટ્રેક પર ફેલાયેલ છે ઓડિયો સીડીમાંથી અને ડિસ્ક ઈમેજીસ બનાવો અથવા બર્ન કરો વિવિધ બંધારણો. તેઓ જેવી સુવિધાઓ પણ પ્રદાન કરી શકે છે WAV, MP3, FLAC થી ઓટોમેટિક ઓડિયો કન્વર્ઝન, ડબલ્યુએમએ ફાઇલો, ડિસ્ક કોપી, સંકુચિત ફાઇલ બેકઅપ અને પુનઃસ્થાપિત કરો, ડિસ્ક ભૂંસી નાખો, VCD/SVCD સપોર્ટ, પ્રોજેક્ટ બર્ન કરો વગેરે.

બ્લુ - રે ડિસ્ક

બ્લુ-રે, જેને બ્લુ-રે ડિસ્ક (BD) તરીકે પણ ઓળખવામાં આવે છે તેનું નામ છે એક નવું ઓપ્ટિકલ ડિસ્ક ફોર્મેટ જે ઝડપથી DVD ને બદલી રહ્યું છે. રેકોર્ડિંગ, પુનઃલેખનને સક્ષમ કરવા માટે ફોર્મેટ વિકસાવવામાં આવ્યું હતું અને હાઇ-ડેફિનેશન વિડિયો (HD)નું પ્લેબેક, તેમજ મોટી માત્રામાં ડેટાનો

સંગ્રહ. ફોર્મેટ વધુ તક આપે છે પરંપરાગત ડીવીડીની સંગ્રહ ક્ષમતા કરતાં પાંચ ગણી અને સિંગલ-લેયર ડિસ્ક પર 25GB અને ડ્યુઅલ-લેયર ડિસ્ક પર 50GB સુધી પકડી શકે છે.

બ્લુ-રે નામ અંતર્ગત પરથી ઉતરી આવ્યું છે ટેકનોલોજી, જે વાંચવા માટે બ્લુ-વાયોલેટ લેસરનો ઉપયોગ કરે છે અને ડેટા લખો. નામ “વાદળી” (વાદળી-વાયોલેટ લેસર) અને “રે” (ઓપ્ટિકલ રે). તેમને “બ્લુ-રે” ડિસ્ક અથવા BDs તરીકે ઓળખવામાં આવે છે.

નીચેના ફોર્મેટ આજકાલના બ્લુ-નો ભાગ છે રે ડિસ્ક સ્પષ્ટીકરણ:

- 1 BD-ROM - HD ના વિતરણ માટે ફક્ત વાંચવા માટેનું ફોર્મેટ ફિલ્મો, રમતો, સોફ્ટવેર, વગેરે.
- 2 BD-R - HD વિડિયો રેકોર્ડિંગ માટે રેકોર્ડ કરવા યોગ્ય ફોર્મેટ અને પીસી ડેટા સ્ટોરેજ.
- 3 BD-RE - HD વિડિયો રેકોર્ડિંગ માટે ફરીથી લખી શકાય તેવું ફોર્મેટ અને પીસી ડેટા સ્ટોરેજ.

હાલમાં, સિંગલ-લેયર ડિસ્ક 25GB અને ડ્યુઅલ-લેયર ડિસ્ક 50GB પકડી શકે છે. હાઇ-ડેફિનેશનના 9 કલાકથી વધુ 50GB ડિસ્ક પર (HD) વિડિયો. 50GB ડિસ્ક પર લગભગ 23 કલાકનો સ્ટાન્ડર્ડ-ડેફિનેશન (SD) વિડિયો.

COPA - DOS કમાન્ડ લાઈન ઈન્ટરફેસ

ડિરેક્ટરી સૂચિ માટે મૂળભૂત DOS આદેશોનો ઉપયોગ કરો (Use basic DOS commands for directory listing)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- DOS માં હાયરાર્કિકલ ડિરેક્ટરી સિસ્ટમનું વર્ણન કરો
- ડિરેક્ટરીઓ અને સબડિરેક્ટરીઝ બનાવવા માટે dos આદેશોનો ઉપયોગ કરો
- ડિરેક્ટરી બદલવા અને સૂચિબદ્ધ કરવા માટે dos આદેશોનો ઉપયોગ કરો
- ચોક્કસ ફાઈલોને એક્સેસ કરવા માટે dos આદેશોનો ઉપયોગ કરો.

હાયરાર્કિકલ ડિરેક્ટરી સિસ્ટમ: સરળમાં વંશવેલો શરતો, છે, સંસ્થા અથવા સંસ્થાઓની વ્યવસ્થા. એન્ટિટી કંઈપણ હોઈ શકે છે જેમ કે વસ્તુઓ, ફાઈલો, લોકો, વિચારો, અથવા અન્ય કોઈપણ વસ્તુ.

ગોઠવણીનો સંદર્ભ આપે છે, ઉદાહરણ તરીકે, ચલણ હોઈ શકે છે સંપ્રદાય દ્વારા ગોઠવાયેલ. કાંકરા તેમના કદ દ્વારા ગોઠવી શકાય છે.

આ ઉપરાંત સંસ્થાઓને ગોઠવવાની બીજી ઘણી રીતો છે વંશવેલો. પરંતુ, વંશવેલો સંગઠન વિશેષ છે કારણ કે આ વ્યવસ્થા દ્વારા તમે દરેક એન્ટિટીનું નામ તેના સંબંધથી અલગ અલગ સંસ્થાઓ દ્વારા આપી શકો છો.

DOS માં, એકમો એ ડિરેક્ટરી સિસ્ટમમાં ડિરેક્ટરીઓ છે. અહીં, વંશવેલો આવશ્યક કોર અથવા મૂળથી શરૂ થાય છે એન્ટિટી દાખલા તરીકે, કુટુંબના વૃક્ષમાં, આપણે વિચારી શકીએ મહાન-પ્રપૌત્ર પિતા જે અમારા મૂળ કારણ હતા મુખ્ય એન્ટિટી તરીકે અસ્તિત્વ. DOS માં, આ મુખ્ય એન્ટિટી છે રૂટ ડિરેક્ટરી તરીકે ઓળખવામાં આવે છે.

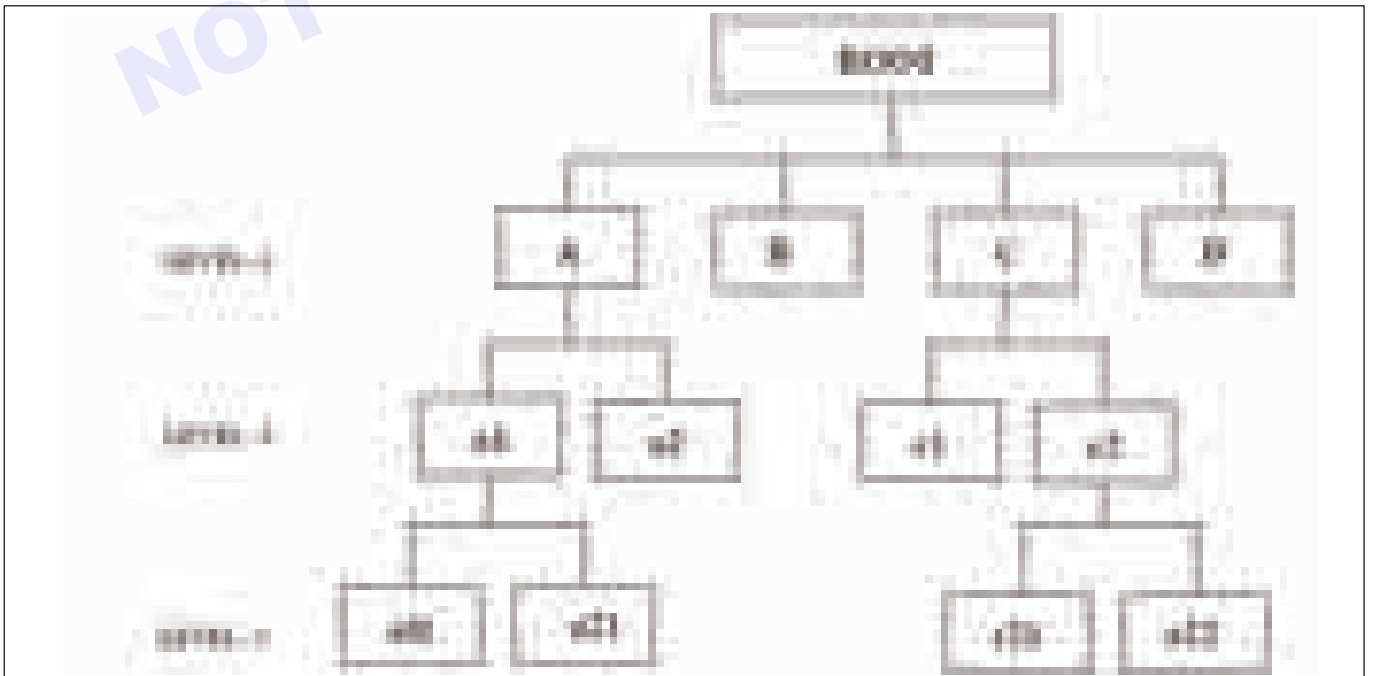
ઉપરના ઉદાહરણની જેમ, જો આપણે ધ્યાનમાં લઈએ રૂટ ડિરેક્ટરી તરીકે મહાન-મહાન-દાદા પિતા, પછી, મહાન- દાદા પિતા, દાદા પિતા, પિતાને પેટા તરીકે ઓળખવામાં આવે છે ડિરેક્ટરીઓ તેથી રૂટ ડિરેક્ટરી હેઠળની ડિરેક્ટરીઓ છે DOS માં સબડિરેક્ટરીઝ કહેવાય છે. આ સબડાયરેક્ટરીઝ કરી શકે છે તેમના પાથને રૂટ ડિરેક્ટરીમાં પાછા ખેંચો.

DOS અધિકૃત ફાઈલ સિસ્ટમને ટ્રી-સ્ટ્રક્ચર્ડ કહેવામાં આવે છે ફાઈલ સિસ્ટમ. આ વૃક્ષની રચનાના પાયામાં રૂટ ડિરેક્ટરી છે.

કુટુંબના વૃક્ષમાં, કહો, ગોવિંદા રાજપ્પાનો પુત્ર છે, જે રામપ્પાના પુત્ર છે જે વેંકપ્પાના પુત્ર છે. વેંકપ્પા ગોવિંદા માટે કુટુંબ વૃક્ષનું માથું અથવા મૂળ છે.

તમે રૂટમાંથી ઘણી ડિરેક્ટરીઓ બનાવી શકો છો. મૂળ પછી આ દરેક ડિરેક્ટરીઓના પિતૃ હશે. તમે પેટા નિર્દેશિકાઓ પણ બનાવી શકે છે જે અન્યમાંથી પેદા થાય છે સબડિરેક્ટરીઝ કે જે અન્ય સબડિરેક્ટરીઝમાંથી પેદા થાય છે અને તેથી પર આ નવી સબડિરેક્ટરીઓમાં સબડિરેક્ટરી છે તેમની પિતૃ નિર્દેશિકા. પેટા નિર્દેશિકાઓ કેવી રીતે ગોઠવાય છે મૂળમાંથી વંશવેલો ફિગ 1 માં સચિત્ર છે. આ DOS ડિરેક્ટરી સિસ્ટમને ઘણીવાર ટ્રી-સ્ટ્રક્ચર્ડ ડિરેક્ટરી સિસ્ટમ કહેવામાં આવે છે

પેટા નિર્દેશિકાઓના ત્રણ સ્તર ફિગ 1 માં દર્શાવવામાં આવ્યા છે. સ્તરોની સંખ્યાને ધ્યાનમાં લીધા વિના, ના સંબંધ પેટા નિર્દેશિકાઓ મહત્વપૂર્ણ છે. દરેક સબડિરેક્ટરી, જેમ તેમજ રૂટ ડિરેક્ટરીમાં, વપરાશકર્તા ફાઈલો સમાવી શકે છે. બે ફાઈલો સુધી સમાન ફાઈલ નામ અને એક્સટેન્શન હોઈ શકે છે ફાઈલો વિવિધ ડિરેક્ટરીઓમાં રહે છે. આ કારણ છે કે, DOS તમારી બે સમાન નામવાળી ફાઈલોમાંથી કઈ છે તે જાણવાની જરૂર છે આદેશ સ્પષ્ટ કરે છે. આ માટે DOS ને નામની જરૂર છે ડિરેક્ટરીઓ, મૂળથી શરૂ કરીને, જે ઈચ્છિત તરફ દોરી જાય છે ફાઈલ ફાઈલ તરફ દોરી જતી ડિરેક્ટરી નામોનો આ ક્રમ છે પાથ કહેવાય છે.



પાથ એ ડિરેક્ટરી નામોની સાંકળ છે જે DOS ને કેવી રીતે કરવું તે જણાવે છે તમને જોઈતી ફાઈલ શોધો. દરેક ડિરેક્ટરી અલગ છે અન્ય '।' અક્ષર દ્વારા. આ '।' તરીકે ઓળખવામાં આવે છે DOS ડિરેક્ટરી ડિલિમિટર. ફાઈલોના સંપૂર્ણ પાથનું નામ સહિત ડ્રાઈવ સ્પષ્ટકર્તા (C: અથવા D: વગેરે,) એ સંપૂર્ણ સૂચક છે જ્યાં ફાઈલ સ્થિત છે. લાક્ષણિક પાથ નોટેશન છે નીચે આપેલ છે;

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

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

પાથ અને ડિરેક્ટરી માળખાની વધુ વિગતોની ચર્ચા યોગ્ય પાઠોમાં કરવામાં આવશે.

DOS આદેશો

1 MKDIR Makes or Creates a new Directory.

or

MD

Syntax

MKDIR C:path\dirname

Or

MD d:path\dirname

જ્યાં,

C: સબ ડિરેક્ટરી માટે ડિસ્ક ડ્રાઈવ છે

path\ એ નિર્દેશિકાનો પાથ સૂચવે છે જે કરશે જે સબડિરેક્ટરી બનાવવામાં આવી રહી છે તેને પકડી રાખો.

dirname સર્જનાત્મક હોવાના સબડિરેક્ટરીનું નામ છે.

સ્વિચ

(None)

મહત્વપૂર્ણ નોંધો

- MKDIR અથવા તેનું ટૂંકું સ્વરૂપ અને પસંદગીની રૂટ ડાયરેક્ટરી હેઠળ નવી પેટા નિર્દેશિકાઓ બનાવો.
- તેટલી બધી સબડિરેક્ટરીઝ બનાવવાનું શક્ય છે તમે ઇચ્છો છો, પરંતુ યાદ રાખો: DOS હવે સ્વીકારતું નથી પાથનામ માટે બેકલેશ સહિત 63 કરતાં વધુ અક્ષરો.
- પેટા નિર્દેશિકાઓના ઘણા બધા સ્તરો અને લાંબા નામો સાથે બનાવશો નહીં.
- તમે ડાયરેક્ટરીનું નામ બનાવી શકતા નથી જે વર્તમાન ડિરેક્ટરીમાં ફાઈલ નામ જેવું જ હોય.

ઉદાહરણ તરીકે, જો તમારી પાસે વર્તમાન ડિરેક્ટરીમાં FLIES નામની ફાઈલ છે, તો તમે આ ડિરેક્ટરીમાં FLIES નામથી સબડિરેક્ટરી બનાવી શકતા નથી. જો કે, જો ફાઈલ FLIES ને એક્સ્ટેન્શન આપવામાં આવે તો FLIES.DOC , પછી નામો વચ્ચે વિરોધાભાસ નહીં આવે અને તમે FLIES નામથી સબડિરેક્ટરી બનાવી શકો છો.

ઉદાહરણો

વર્તમાન ડ્રાઈવ હેઠળ ડ્રાઈવરો નામથી સબડિરેક્ટરી બનાવવા માટે, સૂચના હશે,

MKDIR\ડ્રાઈવર્સ અથવા

MD\Drivers

C:\Devices>MD \Printers

આ સૂચના વર્તમાન ડ્રાઈવ C: હેઠળ પ્રિન્ટર્સ નામથી સબડિરેક્ટરી બનાવે છે. નોંધ કરો કે આદેશ અન્ય ઉપડિરેક્ટરી નામના ઉપકરણોમાંથી જારી કરવામાં આવ્યો હોવા છતાં, નવા બનાવેલ સબડિરેક્ટરી પ્રિન્ટર્સ ડિરેક્ટરી ઉપકરણો હેઠળ નહીં પરંતુ સીધા રૂટ C: હેઠળ બનાવવામાં આવે છે. આને C:\ હેઠળ અને C:\Devices હેઠળ DIR આદેશ જારી કરીને ચકાસી શકાય છે

નિર્દેશિકા ઉપકરણો હેઠળ સબડિરેક્ટરી બનાવવા માટે સૂચના હશે,

C:\Devices>MD Printers ડિરેક્ટરીઓ બનાવવાની નીચેની વિવિધ જાતોની ચર્ચા કરો:

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

2 CHDIR અથવા **CD**

વર્તમાન ડિરેક્ટરીનો પાથ બદલે છે અથવા બતાવે છે.

Syntax

CHDIR d: path

અથવા, ટૂંકા સ્વરૂપનો ઉપયોગ કરીને:

C D d : path

D : પાથ માન્ય ડિસ્ક ડ્રાઈવ અને ડિરેક્ટરી નામો છે.

SWITCH

(none)

તમારી પાસે CD સાથે અધિકૃત નિર્દેશિકાઓ દ્વારા દાવપેચ કરવા માટે બે પદ્ધતિઓ છે: (1) રુટ , અથવા ટોચ પર, ડિસ્કની ડિરેક્ટરીથી શરૂ કરીને અને નીચે ખસેડો, અથવા (2) વર્તમાન ડિરેક્ટરીથી શરૂ કરીને અને કોઈપણ દિશામાં આગળ વધો.

ડિસ્કની રૂટ ડાયરેક્ટરીથી શરૂ કરવા માટે, તમારે પાથ અક્ષર (\), જેમ કે \ અથવા B:\ માં પાથની શરૂઆત કરવી જોઈએ. જ્યારે DOS પાથમાં પ્રથમ અક્ષર તરીકે \ જુએ છે, ત્યારે સિસ્ટમ રુટ ડિરેક્ટરીથી શરૂ થાય છે. નહિંતર, DOS વર્તમાન ડિરેક્ટરી સાથે શરૂ થાય છે. ડ્રાઈવ બદલવી: કોમ્પ્યુટરમાં મેમરી, હાર્ડ ડિસ્ક બિલ્ટ ઇન હશે અને તેમાં ફ્લોપી ડિસ્ક, કોમ્પેક્ટ ડિસ્ક વગેરેમાંથી ડેટા સ્ટોર/રીડ કરવાની જોગવાઈ પણ હશે. દરેક ડિસ્કને નામથી ઓળખવામાં આવે છે જેમ કે સી ડ્રાઈવ, એ ડ્રાઈવ, બી ડ્રાઈવ વગેરે..

C ડ્રાઈવને C દ્વારા રજૂ કરવામાં આવે છે: અને A ડ્રાઈવને A: અને તેથી વધુ દ્વારા રજૂ કરવામાં આવે છે. DOS વર્તમાન અથવા ડિફોલ્ટ ડ્રાઈવમાંથી ઇચ્છિત ડિસ્ક ડ્રાઈવનું લેટર આઈડેન્ટિફિકેશન ટાઈપ કરીને અને નીચેના ઉદાહરણમાં બતાવ્યા પ્રમાણે તરત જ કોલોન દ્વારા બદલવાની મંજૂરી આપે છે:

C\> a:

આ આદેશ C ડ્રાઈવથી D ડ્રાઈવમાં નિયંત્રણ બદલવાની સૂચના આપે છે.

જો ફ્લોપીની ઉપલબ્ધતા કે અન્ય કોઈ કારણસર ડિસ્ક ડ્રાઈવ એક્સેસ ન થઈ હોય, તો DOS એક એરર મેસેજ પ્રદર્શિત કરશે

ભૂલ વાંચવા માટે તૈયાર નથી ડ્રાઈવ A

Abort, Retry, Fail?

ભૂલ માટે જવાબદાર આદેશને રદ કરવા માટે A,R અથવા F કી દબાવવી જરૂરી છે, જે

A Directs DOS ની નીચે વ્યાખ્યાયિત છે. જો આ વિકલ્પ પસંદ કરવામાં આવે તો DOS આદેશને સમાપ્ત કરશે અને પ્રોમ્પ્ટ ફરીથી પ્રદર્શિત કરશે.

R DOS ને આદેશને ફરીથી પ્રયાસ કરવા માટે નિર્દેશિત કરે છે જે ભૂલનું કારણ બને છે. મોટા ભાગના કિસ્સાઓમાં આ વિકલ્પ એ સમસ્યાને સુધારવા માટે પસંદ કરવામાં આવે છે જે ભૂલનું કારણ બની રહી હતી. (ફ્લોપી ડિસ્ક કદાચ દાખલ કરી શકાશે નહીં).

F એ ભૂલને અવગણવા અને પ્રક્રિયા ચાલુ રાખવાનો પ્રયાસ કરવા DOS ને નિર્દેશ આપે છે. કેટલાક કિસ્સાઓમાં DOS જ્યારે ડિસ્કનો એક ભાગ વાંચે છે ત્યારે તેમાં ભૂલ હશે

DOS કમાન્ડ

DIR ડિરેક્ટરીમાં ફાઈલો અને સબડિરેક્ટરીઓની સૂચિ દર્શાવે છે.

syntax

DIR C:path/file name [/P] [/W] [/A[:attribs]] [/O[:sortord]]

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

Where,

- C: ફાઈલો અને સબડિરેક્ટરીઓની સૂચિ પ્રદર્શિત કરવા માટે ડિરેક્ટરી ધરાવતી ડિસ્ક ડ્રાઈવ છે
- પાથ/ નિર્દેશિકા અને/અથવા ફાઈલોને સૂચિબદ્ધ કરવા માટે સ્પષ્ટ કરે છે.
- ફાઈલનામ પ્રદર્શિત કરવા માટેની ફાઈલ અથવા ફાઈલોની સૂચિનો ઉલ્લેખ કરે છે, જો ફાઈલનું નામ ઉલ્લેખિત ન હોય તો ડિરેક્ટરીમાંની બધી ફાઈલો સૂચિબદ્ધ કરવામાં આવશે.
- [/P] [/W] આઉટપુટ ફોર્મેટ કરવા માટે સ્વીચોનો ઉલ્લેખ કરે છે. દરેક સ્ક્રીનભર માહિતી પછી

સ્વિચ

/P થોભાવે છે અને કોઈપણ કી દબાવવાની રાહ જુએ છે. કોઈપણ કી દબાવવા પર બીજી સ્ક્રીનફુલ અથવા બાકીની માહિતી પ્રદર્શિત થશે. આદેશ DIR/P

/W છે ફક્ત ફાઈલના નામો દર્શાવવા માટે 80 કોલમના વિશાળ ફોર્મેટનો ઉપયોગ કરે છે અને ફાઈલના કદ, તારીખ અને સમય વિશેની માહિતી પ્રદર્શિત થતી નથી. આદેશ DIR/W

/A છે જે દર્શાવેલ વિશેષતાઓ સાથે ફાઈલો દર્શાવે છે.

લક્ષણો

D DIRECTORIES

R Read only files

H હિડન ફાઈલો

S સિસ્ટમ ફાઈલો

આ કાર્બવ કરવા માટે તૈયાર ફાઈલો - ઉપસર્ગ જેનો અર્થ થાય છે “નહી”

/O સૂચિબદ્ધ ક્રમમાં ફાઈલો.

નામ દ્વારા N ક્રમાંકિત (આલ્ફાબેટિક)

S કદ દ્વારા (સૌથી નાનું પ્રથમ)

E એક્સટેન્શન દ્વારા (આલ્ફાબેટિક)

D તારીખ અને સમય દ્વારા (સૌથી વહેલું પ્રથમ) પ્રથમ

G ગ્રુપ ડિરેક્ટરીઓ

- કમ્પ્રેશન રેશિયો દ્વારા

C રિવર્સ ઓર્ડર માટે ઉપસર્ગ (સૌથી નાનો પ્રથમ)

/S ઉલ્લેખિત ડિરેક્ટરીમાં અને બધી સબડિરેક્ટરીઓમાં ફાઈલો દર્શાવે છે.

/B એકદમ ફોર્મેટનો ઉપયોગ કરે છે (કોઈ હેડિંગ માહિતી અથવા સારાંશ નથી).

/L લોઅરકેસનો ઉપયોગ કરે છે.

/C[H] ફાઈલ કમ્પ્રેશન રેશિયો દર્શાવે છે; /CH હોસ્ટ ફાળવણી એકમ કદનો ઉપયોગ કરે છે.

મહત્વપૂર્ણ નોંધો:

- સમાન ફાઈલોની યાદી આપતી ડિરેક્ટરીમાં વાઈલ્ડકાર્ડ્સ (* અને ?) નો ઉપયોગ કરીને સૂચિબદ્ધ કરી શકાય છે, જ્યાં (*) સ્ટાર અને (?) પ્રશ્ન ચિહ્નને વાઈલ્ડ કેરેક્ટર અથવા વાઈલ્ડ કાર્ડ કહેવામાં આવે છે. * બાકીના અક્ષરોને બદલી શકે છે અને? કોઈપણ એક અક્ષર બદલી શકે છે.
- જ્યારે ડીઆઈઆરનો ઉપયોગ પરિમાણો અથવા સ્વીચો વિના કરવામાં આવે છે, ત્યારે તે ડિસ્ક વોલ્યુમ લેબલ અને સીરીયલ નંબર દર્શાવે છે; એક ડાયરેક્ટરી અથવા ફાઈલનામ પ્રતિ લીટી, બાઈટ્સમાં ફાઈલનું કદ અને તારીખ અને સમય સહિત ફાઈલ સંશોધિત કરવામાં આવી હતી; અને સૂચિબદ્ધ ફાઈલોની કુલ સંખ્યા, તેમનું સંચિત કદ અને ડિસ્ક પર બાકી રહેલી ખાલી જગ્યા (બાઈટ્સમાં)

ઉદાહરણો

DIR *.txt *

txt સૂચના સ્પષ્ટ નિર્દેશિકામાં txt એક્સટેન્શન ધરાવતી બધી ફાઈલોને સૂચિબદ્ધ કરશે.

DIR ???T.*

???T સૂચના ચાર અક્ષરો ધરાવતી ફાઈલોને શોધશે જે T સાથે સમાપ્ત થાય છે જેમ કે TEST, REST વગેરે. અને * સૂચના આપે છે કે આ ફાઈલોમાં .txt, .dat વગેરે જેવા કોઈપણ એક્સટેન્શન હોઈ શકે છે.

COPA - DOS કમાન્ડ લાઈન ઇન્ટરફેસ

DOS આદેશોનો ઉપયોગ કરીને ફાઈલો અને ફોલ્ડર્સનું સંચાલન કરો (Manage files and folders using DOS commands)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ટેક્સ્ટ ફાઈલની સામગ્રી પ્રદર્શિત કરવા માટે DOS આદેશોનો ઉપયોગ કરો
- ફાઈલોની નકલ કરવા, નામ બદલવા, કાઢી નાખવા અને કાઢી નાખવા માટે DOS આદેશોનો ઉપયોગ કરો.

DOS આદેશો

TYPE ટેક્સ્ટ ફાઈલની સામગ્રી દર્શાવે છે.

Syntax

Type C: path/file name

જ્યાં

- C: એ ડિસ્ક ડ્રાઈવ છે જે પ્રદર્શિત કરવા માટે ફાઈલને પકડી રાખે છે.
- path/ પ્રદર્શિત કરવા માટે ફાઈલનું સ્થાન સ્પષ્ટ કરે છે.
- file name પ્રદર્શિત કરવા માટે ફાઈલને સ્પષ્ટ કરે છે.

Switch

(none)

મહત્વપૂર્ણ નોંધો:

- TYPE કમાન્ડ ASCII ફાઈલની સામગ્રીને અન્ય પ્રોગ્રામનો ઉપયોગ કર્યા વિના પ્રદર્શિત કરવાની ઝડપી રીત પ્રદાન કરે છે. ફાઈલ ડિસ્ક પર ASCII (અમેરિકન સ્ટાન્ડર્ડ કોડ ફોર ઇન્ફોર્મેશન ઇન્ટરચેન્જ) ટેક્સ્ટ તરીકે સંગ્રહિત થાય છે. જે પ્રમાણભૂત રીતે કોમ્પ્યુટર બાઈનરી (એક અને શૂન્ય) ને અક્ષરો, સંખ્યાઓ અને પ્રતીકોમાં અનુવાદિત કરે છે. જો માહિતી ASCII ફોર્મેટમાં સંગ્રહિત ન હોય, તો TYPE આદેશનો ઉપયોગ કરવાથી માહિતી અસ્પષ્ટ જેવી દેખાશે.
- આદેશ જારી કરવા પર DOS ઉલ્લેખિત ડ્રાઈવમાં દેખાશે, ઉલ્લેખિત ફાઈલનામ સુધી પહોંચવા માટે પાથમાં જશે. પછી તે ASCII ફોર્મેટને અક્ષરો, સંખ્યાઓ અને પ્રતીકોમાં ભાષાંતર કરે છે અને મોનિટર પર પ્રદર્શિત કરે છે. વિડિયો મોનિટર માત્ર 24 લાઈનની માહિતી બતાવી શકે છે. જો ફાઈલમાં 24 થી વધુ લીટીઓ હોય તો શરુઆતની લીટીઓ જોઈ શકાતી નથી કારણ કે ટાઈપ કમાન્ડ ફક્ત સ્ક્રીન પર બધી માહિતીને સ્ક્રોલ કરે છે. કંટ્રોલ + એસ કીને એકસાથે દબાવીને સ્ક્રોલીંગને નિયંત્રિત કરી શકાય છે (કંટ્રોલ કીને પકડી રાખવા પર S કી દબાવો અને બંને કી રીલીઝ કરો કંટ્રોલ + એસ તરીકે ઓળખાય છે) માહિતીનું સ્ક્રોલીંગ મોનિટર પર બંધ થઈ જશે. સ્ક્રીન પર સમાવિષ્ટો જોયા પછી બાકીના સમાવિષ્ટોને સ્ક્રોલ કરવા માટે કોઈપણ કી દબાવી શકાય છે. સ્ક્રીન પેજ દ્વારા ફાઈલ સ્ક્રીન પૃષ્ઠની સામગ્રીઓ જોવા માટે, MORE આદેશનો પણ ઉપયોગ કરી શકાય છે. જે સ્ક્રીન પરની માહિતીના સ્ક્રોલીંગને સ્ક્રીન પેજ પછી બરાબર બંધ કરશે અને સ્ક્રીન પેજમાં 24 લીટી પર પ્રોમ્પ્ટ મેસેજ — Mor — પ્રદર્શિત થાય છે. કોઈપણ કી દબાવ્યા પછી બીજું સ્ક્રીન પેજ પ્રદર્શિત થશે. MORE એ ફિલ્ટર છે દા.ત. itis એક પ્રોગ્રામ છે જે સ્ટાન્ડર્ડ આઉટપુટ (મોનિટર) સ્ક્રીન પેજ પર સ્ક્રીન પેજ દ્વારા ફાઈલમાં પ્રમાણભૂત અક્ષરોના પ્રવાહને મેનિયુલેટ કરે છે.

ઉદાહરણ

- 1 C:\COPA\DOS\PR CT 3>TYPE TEST1.txt:\COPA\ OS\ PRACT_3 એ TEST.txt ફાઈલનો પાથ છે અને TYPE એ DOS દ્વારા એક્ઝિક્યુટ કરવાનો આદેશ છે.
- 2 C:\>TYPE C:\COPA\DOS\PR CT_3\>TEST1.txt
આ ઉદાહરણ 1 ની જેમ જ આઉટપુટમાં પરિણમે છે. C: (C ડ્રાઈવ) થી કામ કરતી વખતે આ આદેશ ડિરેક્ટરીઓ બદલ્યા વિના જારી કરી શકાય છે.
- 3 C:\COPA\DOS\PR CT_3>TYPE TEST1.txt | MORE
આ સમાન આઉટપુટમાં પણ પરિણમશે પરંતુ સ્ક્રીન પૃષ્ઠ દ્વારા સ્ક્રીન પૃષ્ઠ પ્રદર્શિત થશે. કોઈપણ કી દબાવવા પર સ્ક્રીન પૃષ્ઠ બદલી શકાય છે. MORE ની સાથે અન્ય એક અક્ષરનો ઉપસર્ગ 'J' છે જેને પાઈપિંગ કમાન્ડ કહે છે, જે TYPE કમાન્ડના આઉટપુટને અન્ય આદેશમાં રૂટ કરશે અને વધુ ફિલ્ટર માહિતીને આઉટપુટ કરશે.

Renaming of File(s)

RENAME આ આદેશ ફાઈલનું નામ

OR

અથવા નામ બદલવાની મંજૂરી આપે છે.

REN

Syntax

REN C: PATH\filename1.e 1 filename2.ex2

Where,

- C: ડિસ્ક માટે ફાઈલને પકડી રાખેલી ડિસ્ક ડ્રાઈવ છે.
- PATH/ પ્રદર્શિત કરવા માટે ફાઈલનું સ્થાન સ્પષ્ટ કરે છે.
- filename1.ex1 એ ફાઈલ છે જેનું નામ બદલી શકાય છે
- filename2.ex2 એ નવું ફાઈલનામ છે

મહત્વપૂર્ણ નોંધો:

- જો ડ્રાઈવ સ્પષ્ટ કરેલ નથી, તો વર્તમાન ડિસ્ક ડ્રાઈવનો ઉપયોગ કરવામાં આવશે.
- જો પાથ ઉલ્લેખિત ન હોય તો વર્તમાન ડિરેક્ટરીનો ઉપયોગ કરવામાં આવશે
- ફાઈલનું નામ બદલવા માટે એક્સ્ટેન્શન સાથે ચોક્કસ ફાઈલનું નામ આપવું પડશે.

- નવા ફાઇલનામ માટે યોગ્ય એક્સ્ટેન્શન સાથેનું માન્ય ફાઇલ નામ આપવાનું રહેશે.
- ફાઇલના નામોમાં જંગલી અક્ષરોની પરવાનગી છે જેના દ્વારા ફાઇલોના જરૂરી જૂથનું નામ બદલી શકાય છે.
ફક્ત ફાઇલના નામો જ બદલાશે અને સમાવિષ્ટો સમાન રહેશે.
- જો ડિરેક્ટરીમાં પહેલાથી જ અસ્તિત્વમાં હોય તેવા નામમાં ફાઇલનું નામ બદલવાનો પ્રયાસ કરવામાં આવે તો.
DOS ડુપ્લિકેટ ફાઇલ નામ અથવા ફાઇલ મળી નથી એક ભૂલ સંદેશો પૂછે છે
- જો અમાન્ય ફાઇલનું નામ અથવા નવું નામ આપવામાં આવ્યું નથી, તો DOS પણ એક ભૂલ સંદેશો સૂચવે છે

ફાઇલ નામો માટે નિયમો

- ફાઇલના નામમાં 1 થી 8 અક્ષરો હોવા આવશ્યક છે.
- 1 થી 3 અક્ષરોનું વૈકલ્પિક વિસ્તરણ
- નામ અને એક્સ્ટેન્શન નામ વચ્ચેનો સમયગાળો (.), જો એક્સ્ટેન્શનનો ઉપયોગ કરવામાં આવે છે
- A થી Z સુધીના બધા અક્ષરો (લોઅરકેસ અક્ષરો આપોઆપ અપરકેસમાં ટ્રાન્સફર થાય છે), 0 થી 9 નંબરો અને વિશિષ્ટ અક્ષરો અને પ્રતીકો \$ # & @ ! ^ () _ { } ' ~ ફાઇલના નામમાં પરવાનગી છે.
- Esc, Del, અથવા space bar જેવા નિયંત્રણ અક્ષરોનો ફાઇલના નામમાં ઉપયોગ કરી શકાતો નથી.
- અક્ષરો + = / [] ; : ? * < > : પરવાનગી નથી.
- ડિરેક્ટરીમાં દરેક ફાઇલનું નામ અનન્ય હોવું જોઈએ

ઉદાહરણો:

- 1 C:\COPA\DOS\PR CT_3\>REN TEST2.txt CHECKED.txt
C:\COPA\DOS\PR CT_3\ એ TEST2.txt ફાઇલની ડ્રાઇવ અને પાથ છે
TEST2.txt એ ફાઇલનું નામ છે નામ બદલવું
CHECKED.txt એ નવું ફાઇલનામ છે.
- 2 C:\COPA\DOS\PR CT_3\>REN *.pic *.jpg ચિત્ર એક્સ્ટેન્શનને jpg એક્સ્ટેન્શન ફાઇલોમાં બદલવામાં આવશે.

Copying Files :

Copy એક અથવા વધુ ફાઇલોને બીજા સ્થાન પર કૉપી કરે છે.

Syntax

Copy [/A | /B] સ્ત્રોત [/A | /B] [+ સ્ત્રોત [/A | /B] [+ ...]] [ગંતવ્ય[/A | /B]] [/V] [/Y | /-Y] સ્રોત નકલ કરવા માટેની ફાઇલ અથવા ફાઇલોને સ્પષ્ટ કરે છે. ગંતવ્ય નવી ફાઇલ(ઓ) માટે ડિરેક્ટરી અને/અથવા ફાઇલનામનો ઉલ્લેખ કરે છે.

Switches

/A એ ASCII ટેક્સ્ટ ફાઇલ સૂચવે છે.

/B બાઈનરી ફાઇલ સૂચવે છે.

/V ચકાસે છે કે નવી ફાઇલો યોગ્ય રીતે લખાઈ છે.

/Y તમે હાલની ગંતવ્ય ફાઇલ પર ફરીથી લખવા માંગો છો તેની પુષ્ટિ કરવા માટે સંકેતને દબાવી દે છે.

/-Y તમે હાલની ગંતવ્ય ફાઇલ પર ફરીથી લખવા માંગો છો તેની પુષ્ટિ કરવા માટે સંકેત આપે છે

પ્રશિક્ષક દરેક કેસમાં ઓછામાં ઓછા બે ઉદાહરણો સાથે સરળ સ્વીચોની ચર્ચા કરશે. COPY કમાન્ડ સ્વીચો પર વધુ વિગતો માટે DOS પર કોઈપણ ટ્યુટોરિયલ અથવા હેન્ડ બુકનો સંદર્ભ લો

મહત્વપૂર્ણ નોંધો:

- DOS કમાન્ડ કોપી એક અથવા વધુ ડુપ્લિકેટ કરી શકે છે ફાઇલો. એક જ ડિરેક્ટરીમાં અલગ-અલગ નામો સાથે અથવા એક ડિરેક્ટરીમાંથી બીજી ડિરેક્ટરીમાં તે જ નામથી અથવા અલગ નામથી.
- જો ડ્રાઇવ સ્પષ્ટ કરેલ નથી, તો વર્તમાન ડિસ્ક ડ્રાઇવનો ઉપયોગ કરવામાં આવશે.
- જો પાથ ઉલ્લેખિત ન હોય તો વર્તમાન ડિરેક્ટરીનો ઉપયોગ કરવામાં આવશે.
- ફાઇલ કોપી કરવા માટે એક્સ્ટેન્શન સાથેનું ચોક્કસ નામ આપવું જોઈએ
- નવી કોપી કરેલી ફાઇલનામ માટે યોગ્ય એક્સ્ટેન્શન સાથેનું માન્ય ફાઇલ નામ આપવાનું રહેશે
- ફાઇલના નામોમાં જંગલી અક્ષરોની પરવાનગી છે જેના દ્વારા ફાઇલોના જરૂરી જૂથની નકલ કરી શકાય છે
- કૉપી કરવા પર, સ્રોત અને લક્ષ્ય ફાઇલો બંનેમાં સમાન સામગ્રી હશે.
- કૉપી એ જ નામ સાથે લક્ષ્ય ફાઇલ પર ફરીથી લખે છે
- કૉપી એ ફાઇલને તેની જાતે કૉપી કરવાની મંજૂરી આપશે નહીં જે સ્રોત છે અને લક્ષ્ય ફાઇલો સમાન હોવી જોઈએ નહીં
- જો ગંતવ્ય ફાઇલનું નામ નિર્દિષ્ટ ન હોય તો જોડાણ કરતી વખતે પ્રથમ ફાઇલનું નામ ગંતવ્ય નામ બની જશે. પ્રથમ ફાઇલના નામ પછી, વધારાની સ્રોત ફાઇલોની આગળ a વર્તા (+) ચિહ્ન.

ઉદાહરણ

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

ઉપરના આદેશ સાથે C:\COPA\DOS\PR CT_3 ડિરેક્ટરી TEST2.txt ફાઇલ એ જ ડિરેક્ટરીમાં TRIAL.txt ફાઇલ તરીકે કૉપી કરવામાં આવશે. . ડાયરેક્ટરી લિસ્ટ કરવા પર બંને ફાઇલોમાં સમાન વિગતો હશે અને બંને ફાઇલની સામગ્રી જોવા પર સમાન હશે. DOS કોપી કર્યા પછી સંદેશો 1 ફાઇલ કોપી કરે છે સમાન ડિરેક્ટરીમાં pic એક્સ્ટેન્શન ફાઇલો તરીકે કૉપી કરેલ. DOS કૉપી કરતી વખતે તેણે કૉપી કરેલી ફાઇલનું નામ પૂછે છે અને કૉપી કરવાનું પૂરું થયા પછી કૉપી કરેલી ફાઇલોની સંખ્યા પૂછે છે.

- 2 C:\COPA\DOS\PRACT_3\>Copy *bmp *.pic

ઉપરોક્ત આદેશ C:\COPA\DOS\PRACT_3 ડાયરેક્ટરી સાથે bmp એક્સ્ટેન્શન ફાઇલ ધરાવતી તમામ ફાઇલો એ જ ડિરેક્ટરીમાં pic એક્સ્ટેન્શન ફાઇલો તરીકે કૉપી કરવામાં આવશે. કૉપી કરતી વખતે

DOS એ કૉપિ કરેલી ફાઇલનું નામ પૂછે છે અને કૉપિ કરવાનું પૂરું થયા પછી કૉપિ કરેલી ફાઇલોની સંખ્યા પૂછે છે.

3 C:\COPA\DOS\PRCT_3\>COPY *.pic C:\COPA\DOS\PRCT_4\

C:\COPA\DOS\PART 3 માં ચિત્ર એક્સ્ટેન્શન સાથેની બધી ફાઇલો ડિરેક્ટરી C:\COPA\DOS\PART_4 પર કૉપિ કરવામાં આવશે સમાન નામ અને એક્સ્ટેન્શન સાથેની ડિરેક્ટરી.

કૉપિ કમાન્ડ પછી *.* નો ઉપયોગ કરવાથી તમામ એક્સ્ટેન્શનવાળી બધી ફાઇલોને ડેસ્ટિનેશન પર કૉપિ કરવામાં આવશે

કૉપિ કૉનકેટનેટિંગ: સ્ત્રોત ફાઇલો વચ્ચે + નો ઉપયોગ કરીને બહુવિધ ફાઇલને એકીકૃત કરવા માટે એક ફાઇલ બનાવી શકાય છે અને તેને કૉનકેટનેશન કહેવામાં આવે છે.

ઉદાહરણ 4 C:\COPA\DOS\PRCT_3\ COPY TEST2.txt + TRIAL.txt CONCAT1.txt

ઉપરોક્ત આદેશ સાથે TEST2.txt અને TRIAL.txt ને જોડવામાં આવશે અને CONCAT1.txt ફાઇલ બનાવવામાં આવશે જેમાં સમાવિષ્ટો હશે. પ્રથમ બે સ્રોત ફાઇલોમાંથી.

Deleting file

DEL ઉલ્લેખિત ફાઇલો કાઢી નાખે છે.

or

ERASE

સિન્ટેક્સ

DEL

C:path/filename [/P]

ERASE C:path/filename [/P]

Where,

- C: કાઢી નાખવાની ફાઇલને પકડી રાખતી ડિસ્ક ડ્રાઇવ છે.
- path/ કાઢી નાખવાની ફાઇલનું સ્થાન સ્પષ્ટ કરે છે.
- ફાઇલનામ એ ફાઇલ છે જેને કાઢી નાખવામાં આવે છે.

Switch

/P ઉલ્લેખિત કાઢી નાખતા પહેલા પુષ્ટિ માટે પૂછે છે ફાઇલ /P સ્વીચનો ઉપયોગ કરીને

જો /P સ્વીચનો ઉપયોગ થાય, તો DEL ફાઇલનું નામ દર્શાવે છે અને નીચેના ફોર્મેટમાં સંદેશ સાથે સંકેત આપે છે:

ફાઇલનામ, કાઢી નાખો (Y/N)?

કાઢી નાખવાની પુષ્ટિ કરવા માટે Y દબાવો, કાઢી નાખવાને રદ કરવા માટે N દબાવો અને આગલી ફાઇલનામ (જો ફાઇલોનો સમૂહ ઉલ્લેખિત હોય) અથવા DEL આદેશને રોકવા માટે CTRL+C દબાવો.

મહત્વપૂર્ણ નોંધો

- જો ડ્રાઇવ સ્પષ્ટ કરેલ નથી, તો વર્તમાન ડિસ્ક ડ્રાઇવનો ઉપયોગ કરવામાં આવશે
- જો પાથ ઉલ્લેખિત ન હોય તો વર્તમાન ડિરેક્ટરીનો ઉપયોગ કરવામાં આવશે

- ફાઇલ કાઢી નાખવા માટે એક્સ્ટેન્શન સાથે ચોક્કસ ફાઇલનું નામ આપવાનું છે
 - ફાઇલના નામોમાં જંગલી અક્ષરોની પરવાનગી છે જેના દ્વારા ફાઇલોના જરૂરી જૂથને કાઢી શકાય છે
 - કાઢી નાખવા પર, ફાઇલોના નામ(ઓ) ડિરેક્ટરીમાંથી દૂર કરવામાં આવશે.
 - ડિરેક્ટરીમાંની બધી ફાઇલો ડીઈએલ કમાન્ડ અને ત્યારબાદ [ડ્રાઇવ:] પાથ ટાઇપ કરીને ડિલીટ કરી શકાય છે. વાઈલ્ડકાર્ડનો ઉપયોગ (* અને?) એક સમયે એક કરતાં વધુ ફાઇલો કાઢી નાખવા માટે પણ થઈ શકે છે. જો કે, અજાણતા ફાઇલો કાઢી નાખવાનું ટાળવા માટે DEL આદેશ સાથે વાઈલ્ડકાર્ડનો ઉપયોગ સાવધાનીપૂર્વક કરવો જોઈએ.
- બધી ફાઇલો કાઢી નાખવા માટે નીચેનો આદેશ આપવામાં આવ્યો છે.

del *.*

DEL નીચેનો પ્રોમ્પ્ટ દર્શાવે છે:

ડિરેક્ટરીમાંની બધી ફાઇલો કાઢી નાખવામાં આવશે! શું તમને ખાતરી છે (Y/N)?

વર્તમાન ડિરેક્ટરીમાંની બધી ફાઇલોને કાઢી નાખવા માટે Y અને પછી ENTER દબાવો, અથવા

કાઢી નાખવાનું રદ કરવા માટે N દબાવો અને પછી ENTER દબાવો.

- ડિરેક્ટરીઓ DEL આદેશ સાથે દૂર કરી શકાતી નથી ડિરેક્ટરીને દૂર કરવા માટે એક અલગ આદેશ ઉપલબ્ધ છે.
- એકવાર ફાઇલ કાઢી નાખવામાં આવે તો તે પુનઃપ્રાપ્ત કરી શકાતી નથી જો મેમરી સ્પેસ નવી ફાઇલ દ્વારા કબજે કરવામાં આવી હોય. જો આકસ્મિક રીતે ફાઇલ (ઓ) તરત જ કાઢી નાખવામાં આવે છે, તો તે ઉપયોગિતા આદેશનો ઉપયોગ કરીને પુનઃપ્રાપ્ત કરી શકાય છે.
- ડિસ્ક અથવા ડિસ્કેટ પર કાઢી નાખેલી ફાઇલ દ્વારા કબજે કરેલી જગ્યા ખાલી કરવામાં આવે છે.
- ડિલીટ કમાન્ડને સક્રિય કરવા માટે એન્ટર કી દબાવતા પહેલા ફાઇલના નામોમાં ટાઇપોગ્રાફિક ભૂલો માટે તપાસો.

ઉદાહરણ

1 C:\COPA\DOS\PRCT_3\>DEL TEST2.txt

ઉપરના આદેશ સાથે TEST2.txt ફાઇલ C:\COPA\DOS\PRCT_3 ડિરેક્ટરીમાંથી કાઢી નાખવામાં આવશે. નિર્દેશિકા TEST2 સૂચિબદ્ધ કરવા પર, txt ઉપલબ્ધ રહેશે નહીં.

2 C:\COPA\DOS\PRCT_4\>DEL *.txt

C:\COPA\DOS\PRCT_4 ડિરેક્ટરીમાં ઉપરના આદેશ સાથે txt એક્સ્ટેન્શન ધરાવતી બધી ફાઇલો કાઢી નાખવામાં આવશે.

3 C:\COPA\DOS\PRCT_3\TEMP\>DEL *.*

C:\COPA\DOS\PRCT_3\TEMP ડિરેક્ટરીમાં કોઈપણ એક્સ્ટેન્શન ધરાવતી બધી ફાઇલો કાઢી નાખવામાં આવશે.

Recovering deleted files :

UNDELET delet protectaction facility

Syntax

UNDELET C:path/filename [/DT | /DS | /DOS]
UNDELET [/LIST | /ALL | /PURGE[DRIVE] | /
STATUS] /LOAD | /UNLOAD
/UNLOAD | /S[DRIVE] | /T[DRIVE]-entrys]]

Where,

- **C:** એ ડિસ્ક ડ્રાઇવ છે જે કાઢી નાખવાની ફાઇલોને પકડી રાખે છે.
- **પાથ/** કાઢી નાખવાની ફાઇલનું સ્થાન સ્પષ્ટ કરે છે.
- **ફાઇલનામ** એ કાઢી નાખવાની ફાઇલ છે

switches

/LIST પુનઃપ્રાપ્ત કરવા માટે ઉપલબ્ધ કાઢી નાખેલી ફાઇલોની યાદી આપે છે.

/ALL પુષ્ટિ માટે સંકેત આપ્યા વિના ફાઇલોને પુનઃપ્રાપ્ત કરે છે.

/DOS MS DOS દ્વારા કાઢી નાખવામાં આવેલી ફાઇલોને પુનઃપ્રાપ્ત કરે છે.

/DT કાઢી નાખી ટ્રેકર દ્વારા સુરક્ષિત ફાઇલોને પુનઃપ્રાપ્ત કરે છે.

/DS Delete Sentry દ્વારા સુરક્ષિત ફાઇલોને પુનઃપ્રાપ્ત કરે છે.

/LOAD ડિલીટ પ્રોટેક્શન માટે મેમરીમાં લોડ અનડિલીટ કરો.

/UNLOAD અનલોડ મેમરીમાંથી અનડિલીટ કરો.

/PURGE[DRIVE] ડિલીટ સેન્ટ્રી ડિરેક્ટરીમાંની બધી ફાઇલોને પર્જ કરે છે.

/STATUS દરેક ડ્રાઇવ માટે પ્રભાવમાં સુરક્ષા પદ્ધતિ દર્શાવો.

/S[DRIVE] સંરક્ષણની ડિલીટ સેન્ટ્રી પદ્ધતિને સક્ષમ કરે છે.

/T[Sdrive][-entrys] સંરક્ષણની ટ્રેકિંગ પદ્ધતિ કાઢી નાખો સક્ષમ કરે છે

મહત્વપૂર્ણ નોંધો:

એકવાર ડિસ્કમાંથી ફાઇલ કાઢી નાખવામાં આવે, તો તેને પુનઃપ્રાપ્ત કરવું શક્ય નથી. જો કે UNDELETE કમાન્ડ કાઢી નાખેલી ફાઇલોને પુનઃપ્રાપ્ત કરી શકે છે, તે નિશ્ચિતતા સાથે માત્ર ત્યારે જ કરી શકે છે જો ડિસ્ક પર કોઈ અન્ય ફાઇલો બનાવવામાં આવી ન હોય અથવા બદલાઈ ન હોય. જો કોઈ ફાઇલ આકસ્મિક રીતે ડિલીટ થઈ જાય અને તેને રાખવાની જરૂર હોય, તો કોમ્પ્યુટર પરની અન્ય તમામ પ્રવૃત્તિઓ બંધ કરો અને ફાઇલ પુનઃપ્રાપ્ત કરવા માટે તરત જ UNDELETE આદેશનો ઉપયોગ કરો.

ઉદાહરણ

- 1 C:\COPA\DOS\PR CT_3\>UnDELETE TEST2.txt
ઉપરોક્ત આદેશ સાથે TEST2.txt ફાઇલ પુનઃપ્રાપ્ત થશે. લિસ્ટિંગ પર TEST2.txt ફાઇલ C:\COPA\DOS\PR CT_3 ડિરેક્ટરીમાં ઉપલબ્ધ થશે.
- 2 C:\COPA\DOS\PR CT_4\TEMP\>UNDELETE
ઉપરોક્ત આદેશ સાથે બહુવિધ ફાઇલો પુનઃપ્રાપ્ત કરી શકાય છે. DOS દરેક ફાઇલને અનડિલીશનની પુષ્ટિ માટે પૂછશે અને ફાઇલનો પહેલો અક્ષર ટાઈપ કરવાનું કહેશે. ના કાઢી નાખવા અને સૂચિબદ્ધ કર્યા પછી C:\COPA\DOS\PR CT_4 ડાયરેક્ટરી, અનડિલીટ કરેલ ફાઇલના નામ જોઈ શકાય છે.
- 3 C:\COPA\DOS\PR CT_4\TEMP\>UNDELETE /ALL
ઉપરના આદેશથી બહુવિધ ફાઇલો પુનઃપ્રાપ્ત કરી શકાય છે. ડોસ દરેક ફાઇલને કાઢી નાખવાનું રદ કરવાની પુષ્ટિ માટે પૂછશે નહીં. ના કાઢી નાખવા અને સૂચિબદ્ધ કર્યા પછી C:\COPA\DOS\PR CT_4 ડાયરેક્ટરી, અનડિલીટ કરેલ ફાઇલના નામ જોઈ શકાય છે.

Linux ઓપરેટિંગ સિસ્ટમનો પરિચય (Introduction to Linux operating system)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- લિનક્સની ઝાંખી
- લિનક્સના ફ્યુચર્સને વ્યાખ્યાયિત કરો
- લિનક્સના એપ્લિકેશન વિસ્તારને સમજાવો
- કર્નલ વિશે વર્ણન કરો.

Linux ની ઝાંખી

ઓપરેટિંગ સિસ્ટમ

વિકાસકર્તાઓને ખાસ સાધનોની જરૂર હોય છે (જેમ કે કમ્પાઇલર્સ અને GNU માં મળેલી કમાન્ડ લાઇન) એપ્લિકેશન લખવા માટે કે જે કર્નલ સાથે વાત કરી શકે છે. તેઓને ટૂલ્સ અને એપ્લિકેશનની પણ જરૂર છે જેથી એપ્લિકેશન લખાઈ અને ઇન્સ્ટોલ કર્યા પછી બહારની એપ્લિકેશનો માટે કર્નલ એક્સેસ કરવાનું સરળ બને.

ટૂલ્સનો આ સામૂહિક સમૂહ, કર્નલ સાથે જોડાયેલો છે ઓપરેટિંગ સિસ્ટમ તરીકે ઓળખાય છે. તે સામાન્ય રીતે સૌથી નીચો છે કોમ્પ્યુટરના સોફ્ટવેરનું સ્તર જે દ્વારા સુલભ છે સરેરાશ વપરાશકર્તા. સામાન્ય વપરાશકર્તાઓ ઓપરેટિંગ સિસ્ટમ પર પહોંચે છે જ્યારે તેઓ આદેશ વાક્યને એક્સેસ કરે છે.

Linux શક્તિશાળી ટૂલ્સ પૂરા પાડે છે જેની સાથે તેમનું લખવું એપ્લિકેશન્સ: વિકાસકર્તા વાતાવરણ, સંપાદકો અને કમ્પાઇલર્સ ડેવલપરનો કોડ લેવા માટે રચાયેલ છે અને તેને એવી વસ્તુમાં રૂપાંતરિત કરો કે જે કર્નલને એક્સેસ કરી શકે અને કાર્યો પૂર્ણ કરો.

કર્નલની જેમ, Linux ઓપરેટિંગ સિસ્ટમ પણ છે મોડ્યુલર વિકાસકર્તાઓ ઓપરેટિંગ પસંદ કરી શકે છે અને પસંદ કરી શકે છે ના નવા સ્વાદ સાથે વપરાશકર્તાઓ અને વિકાસકર્તાઓને પ્રદાન કરવા માટેના સાધનો Linux ચોક્કસ કાર્યોને પહોંચી વળવા માટે રચાયેલ છે.

Linux નો પરિચય

Linux (ઉચ્ચારણ Lih-nucks) એ UNIX જેવું ઓપરેટિંગ છે સિસ્ટમ કે જે ઘણાં વિવિધ કમ્પ્યુટર્સ પર ચાલે છે. જોકે ઘણા લોકો ઓપરેટિંગ તરીકે Linux નો ઉલ્લેખ કરી શકે છે સિસ્ટમ અને સમાયેલ સોફ્ટવેર, કડક રીતે કહીએ તો, Linux છે ઓપરેટિંગ સિસ્ટમ કર્નલ, જે સોફ્ટવેરના વિતરણ સાથે આવે છે.

Linux સૌપ્રથમ 1991 માં તેના લેખક લિનસ ટોરવાલ્ડ્સ દ્વારા રજૂ કરવામાં આવ્યું હતું હેલ્સિંકી યુનિવર્સિટી ખાતે. ત્યારથી તે વધ્યો છે આસપાસના પ્રોગ્રામરો તરીકે જબરદસ્ત લોકપ્રિયતામાં વિશ્વએ તેમના મફત નિર્માણના પ્રોજેક્ટને સ્વીકાર્યું ઓપરેટિંગ સિસ્ટમ, વિશેષતાઓ ઉમેરવા અને સમસ્યાઓનું નિરાકરણ.

Linux આજની પેઢીના કોમ્પ્યુટરમાં લોકપ્રિય છે યુનિક્સનાં પ્રારંભિક સંસ્કરણો સમાન કારણોસર વપરાશકર્તાઓ ઓપરેટિંગ સિસ્ટમે 20 વર્ષ પહેલાં ચાહકોને આકર્ષ્યા હતા. Linux પોર્ટેબલ છે, જેનો અર્થ છે કે તમને આવૃત્તિઓ મળશે નામ-બ્રાન્ડ અથવા ક્લોન પીસી, Apple Macintoshes પર ચાલી રહ્યું છે, સન વર્કસ્ટેશન, અથવા ડિજિટલ ઇન્વિપેમેન્ટ કોર્પોરેશન આલ્ફા-આધારિત કમ્પ્યુટર્સ. Linux પણ સ્ત્રોત સાથે આવે છે કોડ, જેથી

તમે સોફ્ટવેરને બદલી અથવા કસ્ટમાઇઝ કરી શકો તમારી જરૂરિયાતોને અનુકૂળિત કરો. છેલ્લે, લિનક્સ એક ઉત્તમ ઓપરેટિંગ છે સિસ્ટમ, UNIX ના અન્ય સંસ્કરણોમાંથી અપનાવવામાં આવેલી સુવિધાઓથી સમૃદ્ધ.

Linux ક્યાં છે?

Linux ની સૌથી વધુ જાણીતી મિલકતોમાંની એક તે છે જ્યાં તે કરી શકે છે વાપરેલું. Windows અને OS X મુખ્યત્વે પર જોવા મળે છે વ્યક્તિગત કમ્પ્યુટિંગ ઉપકરણો જેમ કે ડેસ્કટોપ અને લેપટોપ કમ્પ્યુટર્સ અન્ય ઓપરેટિંગ સિસ્ટમ્સ, જેમ કે સિમ્બિયન, ફોન અને પીડીએ જેવા નાના ઉપકરણો પર જોવા મળે છે, જ્યારે મેઈનફ્રેમ અને સુપર કોમ્પ્યુટર્સ મેજરમાં જોવા મળે છે શૈક્ષણિક અને કોર્પોરેટ લેબ્સ વિશિષ્ટ સંચાલનનો ઉપયોગ કરે છે સિસ્ટમો જેમ કે AS/400 અને Cray OS.

Linux, જેણે સર્વર OS અને Has તરીકે તેનું અસ્તિત્વ શરૂ કર્યું ડેસ્કટોપ ઓએસ તરીકે ઉપયોગી બને છે, બધા પર પણ વાપરી શકાય છે આ ઉપકરણોમાંથી. કાંડા ઘડિયાળથી લઈને સુપરકોમ્પ્યુટર્સ, Linuxની ક્ષમતાઓનું લોકપ્રિય વર્ણન છે.

Linux નું ભવિષ્ય

Linux પહેલાથી જ ઘણા વિવિધ પ્રકારો પર સફળ છે ઉપકરણો, પરંતુ ઘણા તકનીકી ક્ષેત્રો પણ છે જ્યાં Linux તરફ આગળ વધી રહ્યું છે, ડેસ્કટોપ અને સર્વર તરીકે પણ વિકાસ અન્ય કોઈપણ કરતા વધુ ઝડપથી વધતો રહે છે ઓપરેટિંગ સિસ્ટમ આજે.

Linux ને લેપટોપની BIOS સિસ્ટમ પર ઇન્સ્ટોલ કરવામાં આવી રહ્યું છે અને નોટબુક કોમ્પ્યુટર, જે વપરાશકર્તાઓને તેમના ચાલુ કરવા માટે સક્ષમ કરશે સેકન્ડની બાબતમાં ઉપકરણો ચાલુ, a લાવવા સુવ્યવસ્થિત Linux પર્યાવરણ. આ વાતાવરણ હશે ઇન્ટરનેટ કનેક્ટિવિટી સાધનો જેમ કે વેબ બ્રાઉઝર અને એન ઇ-મેલ ક્લાયંટ, વપરાશકર્તાઓને વગર ઇન્ટરનેટ પર કામ કરવાની મંજૂરી આપે છે તેમના ઉપકરણમાં તમામ રીતે બુટ કર્યા પ્રાથમિક ઓપરેટિંગ સિસ્ટમ - ભલે તે ઓપરેટિંગ સિસ્ટમ હોય વિન્ડોઝ.

તે જ સમયે, લિનક્સ મોબાઇલ ઇન્ટરનેટ પર દેખાઈ રહ્યું છે ઉપકરણો (MIDS). આમાં એમ્બેડેડ ઉપકરણોનો સમાવેશ થાય છે જેમ કે સ્માર્ટ ફોન અને પીડીએ, તેમજ નેટબુક ઉપકરણો- નાના લેપટોપ-પ્રકારના મશીનો કે જે મુખ્ય ધરાવે છે નાના, વધુમાં તેમના મોટા સમકક્ષોની કાર્યક્ષમતા ઊર્જા-કાર્યક્ષમ પેકેજ.

ક્લાઉડ કમ્પ્યુટિંગની વૃદ્ધિ એ Linux માટે કુદરતી રીતે યોગ્ય છે, જે પહેલાથી જ ઇન્ટરનેટના ઘણા વેબ સર્વર ચલાવે છે. Linux એ એમેઝોનની A3 જેવી ક્લાઉડ સેવાઓને વપરાશકર્તાઓને ઓનલાઈન એપ્લિકેશન અને માહિતી પહોંચાડવા માટે શ્રેષ્ઠ ક્ષમતા સાથે કામ કરવા સક્ષમ બનાવે છે.

ક્લાઉડ કમ્પ્યુટિંગમાં Linux ની વૃદ્ધિ સાથે સંબંધિત છે સુપર કોમ્પ્યુટર્સ પર Linux ની જાણીતી સફળતા, બંનેમાં ઉચ્ચ પ્રદર્શન કમ્પ્યુટિંગ (HPC) અને ઉચ્ચ ઉપલબ્ધતા (HA) વિસ્તારો, જ્યાં ભૌતિકશાસ્ત્રમાં શૈક્ષણિકસંશોધન

અને બાયોએન્જિનિયરિંગ, અને નાણાકીય અને ઉર્જા ઉદ્યોગોમાં કંપનીઓને વિશ્વસનીય અને માપી શકાય તેવી કમ્પ્યુટિંગ શક્તિની જરૂર છે તેમના ઘ્યેયો સિદ્ધ કરો.

ઈન્ટરનેટ પર ઘણી લોકપ્રિય વેબ 2.0 સેવાઓ, જેમ કે Twitter, Linked In, YouTube અને Google બધા આધાર રાખે છે Linux પર તેમની ઓપરેટિંગ સિસ્ટમ તરીકે. નવી વેબ સેવાઓ તરીકે ભવિષ્યમાં આવો, Linux વધુને વધુ પ્લેટફોર્મ બનશે જે આ નવી તકનીકોને ચલાવે છે.

Linux ઓપરેટિંગ સિસ્ટમની વર્તમાન એપ્લિકેશન

આજે Linux ડેસ્કટોપ માર્કેટમાં જોડાઈ ગયું છે. Linux વિકાસકર્તાઓએ નેટવર્કિંગ અને સેવાઓ પર ધ્યાન કેન્દ્રિત કર્યું શરૂઆત, અને ઓફિસ અરજીઓ દૂર કરવા માટે છેલ્લી અવરોધ છે. તેઓને તે સ્વીકારવું ગમતું નથી માઈક્રોસોફ્ટ આ બજાર પર શાસન કરી રહ્યું છે, તેથી પુષ્કળ વિકલ્પો બનાવવા માટે છેલ્લા બે વર્ષથી શરૂ કરવામાં આવી છે Linux ને વર્કસ્ટેશન તરીકે સ્વીકાર્ય પસંદગી, પૂરી પાડે છે સરળ વપરાશકર્તા ઈન્ટરફેસ અને MS સુસંગત ઓફિસ એપ્લિકેશન જેમ કે વર્ડ પ્રોસેસર, સ્પ્રેડશીટ્સ, પ્રેઝન્ટેશન અને તેના જેવા. સર્વર બાજુએ, લિનક્સ એક સ્થિર અને વિશ્વસનીય પ્લેટફોર્મ તરીકે જાણીતું છે, જે એમેઝોન, જાણીતી ઓનલાઈન બુકશોપ, યુએસ પોસ્ટ ઓફિસ, જર્મન આર્મી અને અન્ય ઘણી કંપનીઓ માટે ડેટાબેઝ અને ટ્રેડિંગ સેવાઓ પ્રદાન કરે છે. ખાસ કરીને ઈન્ટરનેટ પ્રદાતાઓ અને ઈન્ટરનેટ સેવા પ્રદાતાઓ ફાયરવોલ, પ્રોક્સી- અને લિનક્સના શોખીન બન્યા છે વેબ સર્વર, અને તમને તેની પહોંચની અંદર એક Linux બોક્સ મળશે દરેક UNIX સિસ્ટમ એડમિનિસ્ટ્રેટર જે પ્રશંસા કરે છે a આરામદાયક મેનેજમેન્ટ સ્ટેશન. પોસ્ટ ઓફિસોમાં, તેઓ છે ચેતા કેન્દ્રો જે મેઈલને રૂટ કરે છે અને મોટા સર્ચ એન્જિનમાં, ક્લસ્ટરોનો ઉપયોગ ઈન્ટરનેટ શોધ કરવા માટે થાય છે. આ છે હજારો હેવી-ડ્યુટી નોકરીઓમાંથી માત્ર થોડીક જ Linux છે સમગ્ર વિશ્વમાં દરરોજ પ્રદર્શન કરી રહ્યું છે. તે પણ વર્થ છે નોંધ કરો કે આધુનિક Linux માત્ર વર્કસ્ટેશનો પર જ ચાલતું નથી, મિડ- અને હાઈ-એન્ડ સર્વિસ, પણ પીડીએ જેવા “ગેજેટ્સ” પર પણ, મોબાઈલ, એમ્બેડેડ એપ્લિકેશન્સનો શિપલોડ અને તે પણ પ્રાયોગિક કાંડા ઘડિયાળો પર. આ ફક્ત Linux ને બનાવે છે વિશ્વની ઓપરેટિંગ સિસ્ટમ હાર્ડવેરની વિશાળ શ્રેણીને આવરી લે છે.

કોડ

લિનક્સ એ અન્ય ઓપરેટિંગ સિસ્ટમોથી પણ અનન્ય છે તેનો કોઈ એક માલિક નથી. Torvalds હજુ પણ વ્યવસ્થા કરે છે Linux કર્નલનો વિકાસ, પરંતુ વ્યાવસાયિક અને ખાનગી વિકાસકર્તાઓ સમગ્ર Linux ઓપરેટિંગ સિસ્ટમ બનાવવા માટે અન્ય સોફ્ટવેરનું યોગદાન આપે છે.

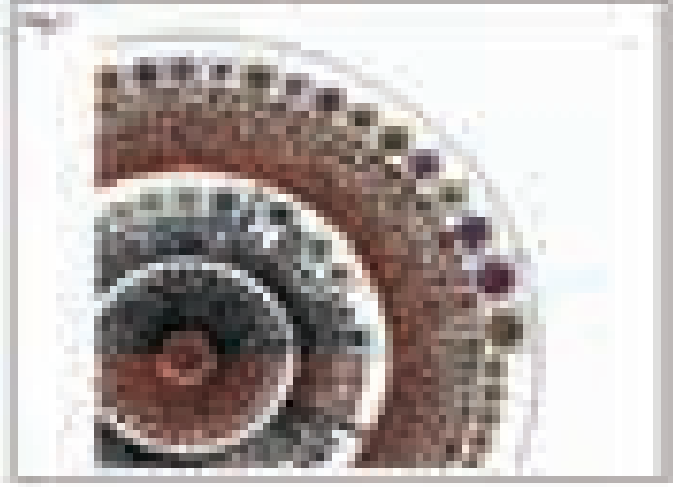
કર્નલ

બધી ઓપરેટિંગ સિસ્ટમમાં કર્નલ હોય છે, જે આજુબાજુ બનેલા હોય છે આર્કિટેક્ચરલ રૂપક કે જેનો કેન્દ્રિય સમૂહ હોવો જોઈએ દ્વારા ઘેરાયેલા ઉપકરણ હાર્ડવેરને નિર્દેશિત કરવા માટેની સૂચનાઓ કાર્યક્ષમતાના વિવિધ મોડ્યુલર સ્તરો. Linux કર્નલ છે અનન્ય અને લવચીક કારણ કે તે પ્રકૃતિમાં મોડ્યુલર પણ છે.

મોડ્યુલારિટી ઇચ્છનીય છે કારણ કે તે વિકાસકર્તાઓને પરવાનગી આપે છે કર્નલના શેડ ભાગો તેઓને વાપરવાની જરૂર નથી. લાક્ષણિક રીતે નાની કર્નલ એ ઝડપી કર્નલ છે, કારણ કે તે ચાલતું નથી પ્રક્રિયાઓ તેની જરૂર નથી.

જો કોઈ ઉપકરણ વિકાસકર્તા ઈચ્છે છે કે Linux નું વર્ઝન એ પર ચાલે સેલ ફોન, તેણીને કર્નલ કાર્યક્ષમતાની જરૂર નથી ડિસ્ક ડ્રાઈવો, ઈથરનેટ ઉપકરણો અથવા મોટા મોનિટર સાથે ડીલ કરે છે સ્ક્રીન તે છોડીને તે ટુકડાઓ (અને અન્ય) ખેંચી શકે છે ફોન માટે ફક્ત ઑપ્ટિમાઇઝ્ડ કર્નલનો ઉપયોગ કરો.

વિન્ડોઝ ઓપરેટિંગ સિસ્ટમનું કર્નલ (જે થોડા માઈક્રોસોફ્ટની બહારના લોકોને વિશેષાધિકાર માટે ચૂકવણી કર્યા વિના જોવાની મંજૂરી છે) એક મજબૂત રીતે જોડાયેલ ભાગ છે કોડ, સરળતાથી ટુકડાઓમાં વિભાજિત કરવામાં અસમર્થ. તે મુશ્કેલ છે (જો અશક્ય ન હોય તો) ફોનને ફિટ કરવા માટે વિન્ડોઝ કર્નલને ડાઉન કરો.



આ મોડ્યુલારિટી Linux ની સફળતા માટે નોંધપાત્ર છે. આ ચોક્કસ જરૂરિયાતોને પહોંચી વળવા માટે સ્કેલ ડાઉન (અથવા ઉપર) કરવાની ક્ષમતા પ્લેટફોર્મ એ અન્ય ઓપરેટિંગ સિસ્ટમ્સ પર એક મોટો ફાયદો છે જે માત્ર થોડા સંભવિત પ્લેટફોર્મ્સ સુધી મર્યાદિત છે.

મોડ્યુલારિટી સ્થિરતા અને સુરક્ષાને પણ અસર કરે છે. જો કર્નલ કોડનો એક ભાગ નિષ્ફળ જાય છે, બાકીનો કર્નલ કેશ થશે નહીં. એ જ રીતે, એક પર ગેરકાયદે હુમલો કર્નલનો ભાગ (અથવા ઓપરેટિંગ સિસ્ટમનો બાકીનો ભાગ) કોડના તે ભાગને અવરોધી શકે છે, પરંતુ ન જોઈએ સમગ્ર ઉપકરણની સુરક્ષા સાથે ચેડા કરો.

પર્યાવરણો

વિન્ડો, મેનુ અને સંવાદ બોક્સ મોટાભાગના લોકો વિચારે છે ઓપરેટિંગ સિસ્ટમના ભાગરૂપે ખરેખર અલગ છે સ્તરો, જે વિન્ડોઈંગ સિસ્ટમ અને ડેસ્કટોપ પર્યાવરણ તરીકે ઓળખાય છે.

આ સ્તરો માનવ-લક્ષી ગ્રાફિકલ વપરાશકર્તા પ્રદાન કરે છે ઈન્ટરફેસ (GUI) જે વપરાશકર્તાઓને ઓપરેટિંગ સિસ્ટમમાંની એપ્લિકેશનો અને ઓપરેટિંગ સિસ્ટમ પર ઈન્સ્ટોલ થનારી તૃતીય-પક્ષ એપ્લિકેશનો સાથે સરળતાથી કામ કરવા સક્ષમ બનાવે છે.

Linux માં, વિન્ડો માટે ઘણી બધી પસંદગીઓ છે સિસ્ટમ અને ડેસ્કટોપ પર્યાવરણનો ઉપયોગ કરી શકાય છે, કંઈક કે જે Linux વપરાશકર્તાઓને નક્કી કરવા દે છે. આ ન કરી શકે Windows માં કરી શકાય છે અને OS X માં કરવું મુશ્કેલ છે.

ઓપરેટિંગ સિસ્ટમ અને કર્નલની જેમ, ત્યાં સાધનો અને છે કોડ લાઈબ્રેરીઓ ઉપલબ્ધ છે જે એપ્લિકેશન વિકાસકર્તાઓને કરવા દે છે આ વાતાવરણ સાથે વધુ સરળતાથી કામ કરો (દા.ત., gtk+ for GNOME, KDE માટે Qt).

અરજી

ઓપરેટિંગ સિસ્ટમમાં બે પ્રકારની એપ્લિકેશનો હોય છે: તે જે ઓપરેટિંગ સિસ્ટમના આવશ્યક ઘટકો છે પોતે, અને જે વપરાશકર્તાઓ પછીથી ઈન્સ્ટોલ કરશે. બંધ વિન્ડોઝ અને ઓએસ એક્સ જેવી ઓપરેટિંગ સિસ્ટમો નહીં થવા દે વપરાશકર્તાઓ (અથવા વિકાસકર્તાઓ) જરૂરી પસંદ

કરે છે અને પસંદ કરે છે ઘટક કાર્યક્રમો તેઓ ઉપયોગ કરી શકે છે. વિન્ડોઝ વિકાસકર્તાઓએ માઈક્રોસોફ્ટના કમ્પાઈલર, વિન્ડોઈંગસિસ્ટમ વગેરેનો ઉપયોગ કરવો જ જોઈએ.

Linux એપ્લિકેશન વિકાસકર્તાઓ પાસે પસંદગીઓનો મોટો સમૂહ છે તેમની એપ્લિકેશન વિકસાવવા માટે. આ વધુ સુગમતા માટે પરવાનગી આપે છે એપ્લિકેશન બનાવો, પરંતુ તેનો અર્થ એ છે કે વિકાસકર્તા કરશે કયા Linux ઘટકોનો ઉપયોગ કરવો તે નક્કી કરવાની જરૂર છે.

વિતરણો

Linux વિતરણ એ (સામાન્ય રીતે ઓપન સોર્સ) નો સંગ્રહ છે Linux કર્નલની ટોચ પર સોફ્ટવેર. વિતરણ (અથવા ટૂંકું, ડિસ્ટ્રો) સર્વર સોફ્ટવેર, સિસ્ટમ મેનેજમેન્ટને બંડલ કરી શકે છે સાધનો, દસ્તાવેજીકરણ અને ઘણી ડેસ્કટોપ એપ્લિકેશનો એ કેન્દ્રીય સુરક્ષિત સોફ્ટવેર રીપોઝીટરી. ડિસ્ટ્રોનો હેતુ છે એક સામાન્ય દેખાવ અને અનુભવ પ્રદાન કરો, સુરક્ષિત અને સરળ સોફ્ટવેર મેનેજમેન્ટ અને ઘણી વખત વિશિષ્ટ ઓપરેશનલ હેતુ તરીકે.

ચાલો કેટલાક લોકપ્રિય વિતરણો પર એક નજર કરીએ.

લાલ ટોપી

Red Hat એક અબજ ડોલરની કોમર્શિયલ લિનક્સ કંપની છે જે Linux ને વિકસાવવા માટે ઘણા પ્રયત્નો કરે છે. તેમની પાસે સેંકડો છે Linux નિષ્ણાતો અને તેમના શ્રેષ્ઠ માટે જાણીતા છે આધાર તેઓ તેમના ઉત્પાદનો આપે છે (રેડ હેટ એન્ટરપ્રાઇઝ લિનક્સ અને ફેડોરા) દૂર મફતમાં. જ્યારે Red Hat Enterprise Linux (RHEL) એ પ્રકાશન પહેલાં સારી રીતે ચકાસાયેલ છે અને રીલીઝ પછી સાત વર્ષ સુધી આધારભૂત, Fedora એ ઝડપી અપડેટ્સ સાથે ડિસ્ટ્રો છે પરંતુ આધાર વિના.

ઓપન સોર્સ સોફ્ટવેરનો પરિચય (Introduction to Open source software)

ઉદ્દેશ્ય : આ પાઠના અંતે તમે સક્ષમ થશો

- OSS ની વ્યાખ્યા
- OSS નો ઇતિહાસ
- ઓપન સોર્સ પહેલ
- OSS ના ઉદાહરણો.

વ્યાખ્યા

ઓપન સોર્સ સોફ્ટવેર (OSS) એ સોફ્ટવેર છે જે તેના સ્રોત કોડ સાથે વિતરિત કરવામાં આવે છે, જે તેને તેના મૂળ અધિકારો સાથે ઉપયોગ, ફેરફાર અને વિતરણ માટે ઉપલબ્ધ બનાવે છે. સોર્સ કોડ એ સોફ્ટવેરનો ભાગ છે જે મોટાભાગના કમ્પ્યુટર વપરાશકર્તાઓ ક્યારેય જોતા નથી; તે કોડ છે જે કોમ્પ્યુટર પ્રોગ્રામરો પ્રોગ્રામ અથવા એપ્લિકેશન કેવી રીતે વર્તે છે તે નિયંત્રિત કરવા માટે હેરફેર કરે છે.

પ્રોગ્રામર્સ કે જેમની પાસે સ્રોત કોડની ઍક્સેસ છે તેઓ પ્રોગ્રામને તેમાં ઉમેરીને, તેને બદલીને અથવા તેના ભાગોને ઠીક કરીને બદલી શકે છે જે યોગ્ય રીતે કામ કરી રહ્યાં નથી. OSS માં સામાન્ય રીતે લાયસન્સ શામેલ હોય છે જે પ્રોગ્રામરોને તેમની જરૂરિયાતોને શ્રેષ્ઠ રીતે ફિટ કરવા અને સોફ્ટવેરનું વિતરણ કેવી રીતે કરી શકાય તે નિયંત્રિત કરવા માટે સોફ્ટવેરમાં ફેરફાર કરવાની મંજૂરી આપે છે.

OSS નો ઇતિહાસ શું છે?

સ્ત્રોત કોડ મુક્તપણે ઉપલબ્ધ કરાવવાનો વિચાર 1983માં એમઆઈટીના પ્રોગ્રામર રિચાર્ડ સ્ટોલમેન દ્વારા અનૌપચારિક રીતે સ્થાપિત વૈચારિક ચળવળમાંથી ઉદ્ભવ્યો હતો. સ્ટોલમેન માનતા હતા કે સોફ્ટવેર પ્રોગ્રામરો માટે સુલભ હોવું જોઈએ જેથી તેઓ તેને સમજવા, તેના વિશે શીખવાના

ઉબુન્ટુ

કેનોનિકલ સાથે મફત કોમ્પેક્ટ ડિસ્ક મોકલવાનું શરૂ કર્યું ઉબુન્ટુ લિનક્સ 2004 માં અને ઝડપથી ઘર માટે લોકપ્રિય બન્યું વપરાશકર્તાઓ (માઈક્રોસોફ્ટ વિન્ડોઝમાંથી ઘણા સ્વિચ કરે છે). કેનોનિકલ ઉબુન્ટુ ગ્રાફિકલ Linux ડેસ્કટોપનો ઉપયોગ કરવા માટે સરળ બનવા માંગે છે ક્યારેય આદેશ વાક્ય જોવાની જરૂર વગર. અલબત્ત તેઓ ઉબુન્ટુ માટે સપોર્ટ વેચીને પણ નફો કરવા માંગે છે.

ડેબિયન

ડેબિયનની પાછળ કોઈ કંપની નથી. તેના બદલે ત્યાં છે હજારો સુવ્યવસ્થિત વિકાસકર્તાઓ જે ડેબિયનને પસંદ કરે છે દર બે વર્ષે પ્રોજેક્ટ લીડર. ડેબિયનને એક તરીકે જોવામાં આવે છે સૌથી સ્થિર Linux વિતરણો. તેનો આધાર પણ છે ઉબુન્ટુની દરેક રીલીઝ. ડેબિયન ત્રણ સંસ્કરણોમાં આવે છે: સ્થિર, પરીક્ષણ અને અસ્થિર. દરેક ડેબિયન રિલીઝ છે ટોચ સ્ટોરી ફિલ્મના એક પાત્રના નામ પરથી નામ આપવામાં આવ્યું છે.

અન્ય

સેન્ટ ઓએસ, ઓરેકલ એન્ટરપ્રાઇઝ લિનક્સ અને જેવા વિતરણો વૈજ્ઞાનિક Linux Red Hat Enterprise Linux પર આધારિત છે અને સમાન સિદ્ધાંતો, ડિરેક્ટરીઓ અને શેર કરો સિસ્ટમ એડમિનિસ્ટ્રેશન તકનીકો. લિનક્સ મિન્ટ, એડુબન્ટુ અને અન્ય ઘણા ઉબુન્ટુ નામના વિતરણો પર આધારિત છે ઉબુન્ટુ અને આમ ડેબિયન સાથે ઘણું શેર કરો. અન્ય સેંકડો લિનક્સ વિતરણો છે.

ધ્યેય સાથે તેની ઈચ્છા મુજબ તેમાં ફેરફાર કરી શકે. સ્ટોલમેને તેના પોતાના લાયસન્સ હેઠળ મફત કોડ બહાર પાડવાનું શરૂ કર્યું, જેને GNU પબ્લિક લાયસન્સ કહેવાય છે. સોફ્ટવેર બનાવટની આસપાસના આ નવા અભિગમ અને વિચારધારાને પકડી લીધો અને છેવટે 1998 માં ઓપન સોર્સ પહેલની રચના તરફ દોરી.

ઓપન સોર્સ પહેલ શું છે?

ઓપન સોર્સ ઇનિશિયેટિવ (OSI) ની રચના ઓપન સોર્સ સોફ્ટવેર અને સમુદાયોને પ્રોત્સાહન અને રક્ષણ કરવા માટે કરવામાં આવી હતી. ii ટૂંકમાં, OSI એ ઓપન સોર્સ સોફ્ટવેરના કેન્દ્રીય માહિતીપ્રદ અને સંચાલક ભંડાર તરીકે કામ કરે છે. તે OSS નો ઉપયોગ કેવી રીતે કરવો અને તેની સાથે કેવી રીતે ક્રિયાપ્રતિક્રિયા કરવી તે માટેના નિયમો અને માર્ગદર્શિકા પ્રદાન કરે છે, તેમજ ઓપન સોર્સના ઉપયોગ અને સારવારને સમજી શકાય તેવું અને નૈતિક બનાવવામાં મદદ કરવા માટે કોડ લાઈસન્સિંગ માહિતી, સમર્થન, વ્યાખ્યાઓ અને સામાન્ય સમુદાય સહયોગ પ્રદાન કરે છે.

OSS કેવી રીતે કામ કરે છે?

ઓપન સોર્સ કોડ સામાન્ય રીતે સાર્વજનિક રિપોઝીટરીમાં સંગ્રહિત થાય છે અને સાર્વજનિક રીતે શેર કરવામાં આવે છે. કોઈપણ વ્યક્તિ સ્વતંત્ર

રીતે કોડનો ઉપયોગ કરવા માટે રીપોઝીટરીને ઍક્સેસ કરી શકે છે અથવા ઍકંદર પ્રોજેક્ટની ડિઝાઇન અને કાર્યક્ષમતામાં સુધારાઓનું યોગદાન આપી શકે છે.

OSS સામાન્ય રીતે વિતરણ લાઇસન્સ સાથે આવે છે. આ લાયસન્સમાં ઁવી શરતોનો સમાવેશ થાય છે જે વ્યાખ્યાયિત કરે છે કે વિકાસકર્તાઓ કેવી રીતે સોફ્ટવેરનો ઉપયોગ કરી શકે છે, અભ્યાસ કરી શકે છે, સંશોધિત કરી શકે છે અને સૌથી અગત્યનું, વિતરિત કરી શકે છે.iii સિનોપ્સિસ બ્લેક ડક® નોલેજ બેઝ અનુસાર, સૌથી વધુ પાંચ લોકપ્રિય લાઇસન્સ છે:

- MIT લાઇસન્સ
- GNU જનરલ પબ્લિક લાઇસન્સ (GPL) 2.0-આ વધુ પ્રતિબંધિત છે અને જરૂરી છે કે સંશોધિત કોડની નકલો જાહેર ઉપયોગ માટે ઉપલબ્ધ કરાવવામાં આવે.
- અપાચે લાઇસન્સ 2.0
- GNU જનરલ પબ્લિક લાઇસન્સ (GPL) 3.0
- BSD લાઇસન્સ 2.0 (3-કલમ, નવું અથવા સુધારેલું)-આ ઓછું પ્રતિબંધિત છે

જ્યારે સોર્સ કોડ બદલવામાં આવે છે, ત્યારે OSS માં શું ફેરફાર કરવામાં આવ્યા હતા તેમજ સામેલ પદ્ધતિઓનો સમાવેશ થવો જોઈએ. લાયસન્સની શરતો પર આધાર રાખીને, આ ફેરફારોના પરિણામે સોફ્ટવેરને મફતમાં ઉપલબ્ધ કરાવવાની જરૂર પડી શકે છે અથવા ન પણ હોઈ શકે.

OSS ના કેટલાક ઉદાહરણો શું છે?

- GNU/Linux
- મોઝીલા ફાયરફોક્સ
- VLC મીડિયા પ્લેયર
- સુગરસીઆરએમ
- GIMP
- VNC
- અપાચે વેબ સર્વર
- લીબરઓફીસ
- jQuery

© NIMI
NOT TO BE REPUBLISHED

આદેશો અને વિવિધ સંપાદકોનું સંચાલન (Handling commands and various editors)

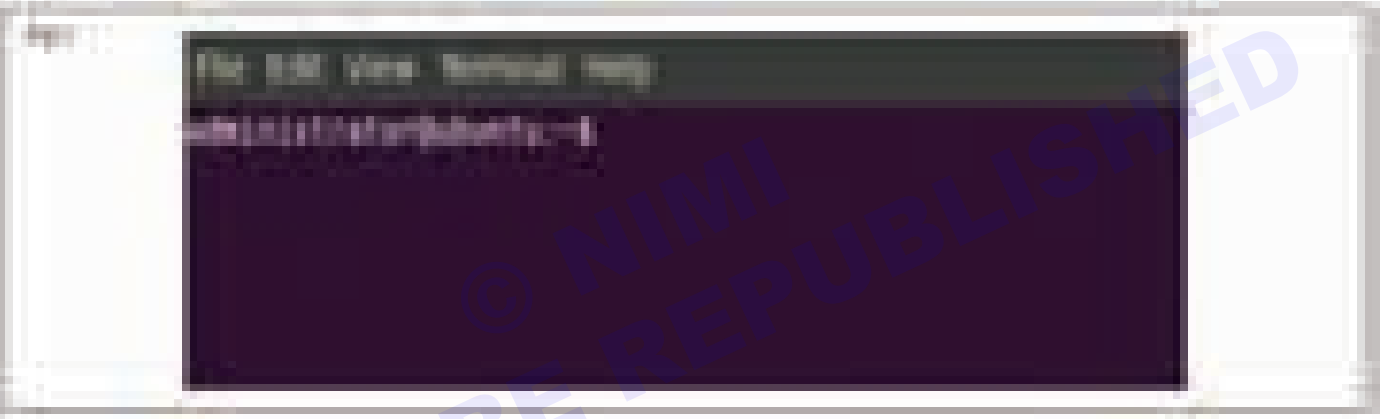
ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ટર્મિનલ વિશે જાણો
- command શેલ સમજાવો
- લિનક્સના ડિસ્કટરી લેઆઉટની યાદી બનાવો
- લિનક્સ આદેશો વ્યાખ્યાયિત કરો
- linux OS ના વિશિષ્ટ અક્ષરોની યાદી બનાવો
- linux OS માં વિવિધ સંપાદકો સમજાવો

ટર્મિનલ શરૂ કરી રહ્યા છીએ

શેલને એક્સેસ કરવા માટે આપણે શેલ જેવી એપ્લિકેશનનો ઉપયોગ કરીશું, જેને ટર્મિનલ ઇમ્યુલેટર પણ કહેવાય છે. GNOME અથવા KDE, અથવા Yakuake, Guake, rxvt વગેરેમાં ડિફોલ્ટ એપ્લિકેશનો સહિત, ત્યાં ઘણી સારી ટર્મિનલ એપ્લિકેશનો છે. હમણાં માટે ચાલો ફક્ત તમારી સિસ્ટમ સાથે આવતા ડિફોલ્ટને વળગી રહીએ. જો તમે GNOME નો

- **sh** - બોર્ન શેલ, એક જૂનો શેલ જે હવે એટલો બહોળો ઉપયોગ થતો નથી.
- **csh** - 'C' શેલ, જે એક સિન્ટેક્સ સ્વીકારે છે જે 'C' પ્રોગ્રામિંગ લેંગ્વેજ જેવું લાગે છે.
- **tcsh** - 'C' શેલનું સુધારેલું સંસ્કરણ.



ઉપયોગ કરી રહ્યા હોવ તો તમે Applications -> Accessories -> Terminal પર જઈને અથવા Alt+F2 દબાવીને અને દેખાતા રન બોક્સમાં gnome-terminal ટાઈપ કરીને એન્ટર દ્વારા અનુસરીને ટર્મિનલને એક્સેસ કરી શકો છો. જો તમે KDE ચલાવી રહ્યા હોવ તો તમે Alt+F2 દબાવીને બદલે કન્સોલ લખી શકો છો.

વિતરણ પર આધાર રાખીને, પ્રોમ્પ્ટ user@host\$ જેવો દેખાઈ શકે છે. એમ્પરસેન્ડ agoનો પહેલો ભાગ લોગિન યુઝરનેમ છે અને બીજો કોમ્પ્યુટરનું હોસ્ટનામ છે.

કમાન્ડ શેલ

શેલ એ કમાન્ડ ઇન્ટરપ્રીટર છે જે તમને કોમ્પ્યુટર સાથે ક્રિયાપ્રતિક્રિયા કરવાની પરવાનગી આપે છે. વસ્તુઓ જે રીતે કાર્ય કરે છે તે ખૂબ જ સરળ છે: તમે આદેશો લખો છો, શેલ તેનું અર્થઘટન કરે છે, તેને જે કરવાનું કહેવામાં આવ્યું હતું તે કાર્યો કરે છે, અને અંતે તે પરિણામોને પ્રમાણભૂત આઉટપુટ પર મોકલે છે, જે સામાન્ય રીતે સ્ક્રીન છે.

આ રૂટ ડિસ્કટરીની અંદરની ફાઈલોની યાદી છે. રુટ ડાયરેક્ટરી એ ફાઈલ સિસ્ટમ ટ્રી હાયરાર્કીમાં પ્રથમ સ્થાન છે, અને તે સ્લેશ અક્ષર દ્વારા રજૂ થાય છે : /.

કેટલાક સૌથી લોકપ્રિય શેલો છે:

- **bash** - બોર્ન-અગેઈન શેલ, મોટાભાગની Linux સિસ્ટમો પર ડિફોલ્ટ શેલ.

- **ksh** - કોર્ન શેલ, શરૂઆતમાં 1980 ના દાયકાની શરૂઆતમાં વિકસિત.
- **dash** - ડેબિયન આલ્મક્રિવેસ્ટ શેલ, ડેબિયન વિતરણ દ્વારા બનાવેલ શેલ

સિસ્ટમમાં ઉપલબ્ધ શેલોની સૂચિ

\$ cat /etc/shells/

ઉપરોક્ત command ફિગ 2 ની જેમ નીચેનું આઉટપુટ પ્રદર્શિત કરશે.

આ ટ્યુટોરીયલમાં આપણે Bash પર ધ્યાન કેન્દ્રિત કરીશું, કારણ કે તે સૌથી વધુ ઉપયોગમાં લેવાય છે અને તે સૌથી શક્તિશાળી શેલ્સમાંથી એક છે. Bash એ GNU પ્રોજેક્ટ દ્વારા વિકસિત જૂના બોર્ન શેલ (sh) નું આધુનિક અમલીકરણ છે, જે વિશાળ માત્રામાં ટૂલ્સ પૂરા પાડે છે અને જે Linux કર્નલ સાથે મળીને, GNOME અથવા KDE જેવા ડેસ્કટોપ એન્વાયર્નમેન્ટ્સ અને એપ્લિકેશન્સ જે તેમની ઉપર ચાલે છે., સમગ્ર Linux પ્લેટફોર્મનો સમાવેશ કરે છે. ડેબિયન અથવા ઉબુન્ટુ વિતરણ પર, સિસ્ટમ દ્વારા ઉપયોગમાં લેવાતા ડિફોલ્ટ શેલ /etc/passwd fileમાં ઉલ્લેખિત છે (ડિફોલ્ટ બેશ છે)

સિસ્ટમમાં ડિફોલ્ટ શેલ કેવી રીતે પ્રદર્શિત કરવું

ટર્મિનલમાં નીચેનો command ટાઈપ કરો અને

\$ echo \$SHELL

Enter કી દબાવો. ડિફોલ્ટ શેલ ફિગ 3 માં દર્શાવવામાં આવશે.

[illegible]

Linux આદેશો શું છે?

Linux આદેશો એક્ઝિક્યુટેબલ બાઈનરી ફાઈલો છે જે ચોક્કસ કાર્યો કરવા માટે ચલાવી શકાય છે, જેમ કે સમગ્ર ગ્રાફિકલ એપ્લિકેશન ચલાવતી ડિસ્કેટરીમાં ફાઈલોની યાદી કરવી. વારંવાર વપરાતા આદેશોના ઉદાહરણો ls, cd, pwd, date અથવા cat છે. એક્ઝિક્યુટેબલ ફાઈલોના અપવાદ સાથે, શેલ બિલ્ટ-ઇન્સ તરીકે ઓળખાતી શ્રેણી પણ છે, જે શેલ દ્વારા જ આપવામાં આવેલ આદેશો છે (અમારા કિસ્સામાં બેશ). અમે તેની સાથે પછીથી વ્યવહાર કરીશું.

Linux command નું સામાન્ય સ્વરૂપ છે:

command વિકલ્પો(ઓ) fileનામ(ઓ) જે command નો ઉલ્લેખ કરે છે, એક અથવા વધુ પરિમાણો દ્વારા અનુસરવામાં આવે છે, અને વૈકલ્પિક રીતે એક અથવા વધુ ફાઇલો તેના પર લાગુ થાય છે. દાખલા તરીકે:

```
$ echo -e 'Hello, world!\n'
```

ટેક્સ્ટને આઉટપુટ કરશે ‘હેલો, વર્લ્ડ!’ એક નવી લાઇન પાત્ર દ્વારા અનુસરવામાં આવે છે. -e પરિમાણ (જેને દલીલ અથવા આ કિસ્સામાં સ્વિચ પણ કહેવાય છે) એસ્કેપ કરેલા અક્ષરોનું અર્થઘટન કરવા ઈકો કમાન્ડને કહે છે, જેમ કે પાછળનું \n, જે એક અવતરણની અંદરના ટેક્સ્ટ પછી નવી લાઇન ઉમેરશે. અગ્રણી ડોલર ચિહ્નને અવગણો, તે ફક્ત શેલ પ્રોમ્પ્ટ દર્શાવે છે. command માં દલીલો હોઈ શકે અથવા ન પણ હોય. દલીલ એક વિકલ્પ અથવા fileનામ હોઈ શકે છે.

લિનક્સ ઓપરેટિંગ સિસ્ટમમાં વિશિષ્ટ અક્ષરો

એ જાણવું અગત્યનું છે કે ત્યાં ઘણા બધા પ્રતીકો અને અક્ષરો છે જેનું શેલ વિશેષ રીતે અર્થઘટન કરે છે. આનો અર્થ એ છે કે અમુક ટાઈપ કરેલા અક્ષરો: a) અમુક પરિસ્થિતિઓમાં ઉપયોગ કરી શકાતો નથી, b) વિશિષ્ટ કામગીરી કરવા માટે ઉપયોગમાં લઈ શકાય છે, અથવા, c) જો તમે તેનો સામાન્ય રીતે ઉપયોગ કરવા માંગતા હોવ તો “એસ્કેપ્ડ” હોવું આવશ્યક છે.

Character	Description
1	Common directory: If you enter in reference a common directory, you must "enter" it with a backslash (\). Example: \\server\share\folder\file
2	Personal directory: used to reference a group of directory names. Example: \\server\share\folder
3	Common directory: that uses "share" that refers to the file system on a server.
4	Parent directory
5	Current directory
6	Represented by two back slashes (\\) or a tilde (~) or by itself, all that is a directory. Example: \\server\share\folder\file, ~, or .
7	Representing a single character in a directory. Example: \\server\share\folder\file, folder\file, or file
8	Can be used to represent a range of values, e.g. [A-Z], [a-z], etc. Example: \\server\share\folder\file, folder\file, or file
9	"Share" refers to the right of using commoned data another computer. Example: \\server\share
10	Refers to the right of a commoned data from the, if the file already exists, overwriting it. Example: \\server\share\file
11	Refers to the right of a commoned data from the, if the file already exists, not overwriting it. Example: \\server\share\file, or \\server\share\file, or \\server\share\file
12	Refers to the right of a commoned data from the, if the file already exists, not overwriting it. Example: \\server\share\file, or \\server\share\file, or \\server\share\file
13	Refers to the right of a commoned data from the, if the file already exists, not overwriting it. Example: \\server\share\file, or \\server\share\file, or \\server\share\file
14	Commoned directory: refers to the commoned directory on a server. Example: \\server\share\file, or \\server\share\file, or \\server\share\file

The cd command

CD કમાન્ડનો ઉપયોગ લિનક્સ અને અન્ય યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમમાં વર્તમાન ડિરેક્ટરી (એટલે કે, ડિરેક્ટરી જેમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે) બદલવા માટે થાય છે. તે MS DOS માં CD અને CHDIR આદેશો જેવું જ છે.

cd's syntax is

cd [option] [directory]

ચોરસ કૌંસમાંની વસ્તુઓ વૈકલ્પિક છે. જ્યારે કોઈપણ ડિરેક્ટરી નામનો ઉલ્લેખ કર્યા વિના ઉપયોગ થાય છે, ત્યારે cd વપરાશકર્તાને પાછલી વર્તમાન ડિરેક્ટરીમાં પરત કરે છે. આ બે ડિરેક્ટરીઓ વચ્ચે ટોંગલ કરવાનું અનુકૂળ માધ્યમ પૂરું પાડે છે.

જ્યારે ડિરેક્ટરીનું નામ આપવામાં આવે છે, ત્યારે cd વર્તમાન ડિરેક્ટરીમાં ફેરફાર કરે છે. નામને સંપૂર્ણ પાથનામ (એટલે કે, રૂટ ડિરેક્ટરીને સંબંધિત સ્થાન) અથવા સ્થાનિક પાથનામ (એટલે કે, વર્તમાન ડિરેક્ટરીને સંબંધિત સ્થાન) તરીકે વ્યક્ત કરી શકાય છે. વર્તમાન ડિરેક્ટરીની સબડિરેક્ટરીમાં બદલાતી વખતે સ્થાનિક પાથનામનો ઉપયોગ કરવો સામાન્ય રીતે વધુ અનુકૂળ છે.

ઉદાહરણ તરીકે, નીચે આપેલ વર્તમાન નિર્દેશિકાને બદલી નાખશે, તે સિસ્ટમ પર ક્યાં છે તે ધ્યાનમાં લીધા વિના (કારણ કે તે એક સંપૂર્ણ પાથ છે), રૂટમાં ડિરેક્ટરી (જે ફોરવર્ડ સ્લેશ દ્વારા દર્શાવવામાં આવે છે):

CD /

તેવી જ રીતે, નીચેના વર્તમાન નિર્દેશિકાને, તેના સ્થાનને ધ્યાનમાં લીધા વિના, /usr/sbin ડિરેક્ટરીમાં બદલશે (જેમાં બિન-મહત્વપૂર્ણ સિસ્ટમ ઉપયોગિતાઓ છે જેનો ઉપયોગ સિસ્ટમ સંચાલક દ્વારા કરવામાં આવે છે):

cd/usr/sbin

જો હાલમાં ડિરેક્ટરી /usr/local/share/ an/ માં વપરાશકર્તા /usr/local/share/man/man2 ડિરેક્ટરીમાં બદલવા ઈચ્છે છે, જે વર્તમાન ડિરેક્ટરીની સબડિરેક્ટરી છે, તો તેનો ઉપયોગ કરીને તેને બદલવાનું શક્ય બનશે. સંપૂર્ણ પાથનામ, એટલે કે,

cd/usr/local/share man/man2

જો કે, તે સ્પષ્ટપણે સંબંધિત પાથનામનો ઉપયોગ કરવા માટે ઘણું ઓછું કંટાળાજનક હશે, એટલે કે,

cd man2

યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ્સ પર વર્તમાન ડિરેક્ટરી સિંગલ ડોટ દ્વારા દર્શાવવામાં આવે છે અને તેની પેરેન્ટ ડિરેક્ટરી (એટલે કે, ડિરેક્ટરી કે જેમાં તે શામેલ છે) સતત બે બિંદુઓ દ્વારા રજૂ થાય છે. આમ, નીચેનાનો ઉપયોગ કરીને વર્તમાન ડિરેક્ટરીના પેરેન્ટમાં ફેરફાર કરવાનું શક્ય (અને ઘણી વાર અનુકૂળ) છે:

CD -

cd ની બીજી અનુકૂળ વિશેષતા એ છે કે કોઈ પણ વપરાશકર્તા માત્ર દલીલ તરીકે ટિલ્ડનો ઉપયોગ કરીને તેની હોમ ડિરેક્ટરીમાં સીધા જ પાછા ફરવાની ક્ષમતા છે. હોમ ડિરેક્ટરી, જેને લૉગિન ડિરેક્ટરી પણ કહેવાય છે, તે યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ પરની ડિરેક્ટરી છે જે વપરાશકર્તાની વ્યક્તિગત ફાઈલો, ડિરેક્ટરીઓ અને પ્રોગ્રામ્સ માટે રિપોઝીટરી તરીકે કામ કરે છે. તે ડિરેક્ટરી પણ છે કે જેમાં વપરાશકર્તા સિસ્ટમમાં લૉગિન થયા પછી પ્રથમ વખત આવે છે. ટિલ્ડ એ ટૂંકું, લહેરાતું, આડી રેખાનું પાત્ર

છે જે વર્તમાન વપરાશકર્તાની હોમ ડિરેક્ટરીનું પ્રતિનિધિત્વ કરે છે. એટલે કે, કોઈપણ વપરાશકર્તા નીચે આપેલ લખીને અને પછી એન્ટર કી દબાવીને તરત જ તેની હોમ ડિરેક્ટરીમાં પાછા આવી શકે છે:

CD ~

વપરાશકર્તાની હોમ ડિરેક્ટરીનું પૂરું નામ લખવા કરતાં આ સરળ છે, ઉદાહરણ તરીકે, જોસેફાઈન નામના વપરાશકર્તાના કિસ્સામાં /home/josephine. (અને તે અસંખ્ય શોર્ટકટ્સમાંથી એક છે જે યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ્સ પર કમાન્ડ લાઈનને વાપરવા માટે ખૂબ જ સરળ બનાવવામાં મદદ કરે છે.) જ્યારે સ્પેસ અને પછી હાઈફન આવે છે, ત્યારે cd બંને વપરાશકર્તાને પાછલી વર્તમાન ડિરેક્ટરીમાં પરત કરે છે અને રિપોર્ટ કરે છે. નવી લાઈન પર તે ડિરેક્ટરીનું સંપૂર્ણ પાથનામ. આ સીડીની agoથી જ અનુકૂળ ટોંગલ કરવાની ક્ષમતાને વધુ વધારી શકે છે. ટોંગલીંગ ખાસ કરીને અનુકૂળ છે જ્યારે બે ડિરેક્ટરીઓમાંથી ઓછામાં ઓછી એક લાંબી સંપૂર્ણ પાથનામ ધરાવે છે, જેમ કે /usr/local/share/an/man2. cd પાસે ફક્ત બે વિકલ્પો છે, અને તેમાંથી કોઈનો સામાન્ય રીતે ઉપયોગ થતો નથી. -P વિકલ્પ સીડીને સાંકેતિક લિંક્સને અનુસરવાને બદલે ભૌતિક નિર્દેશિકા માળખાનો ઉપયોગ કરવાની સૂચના આપે છે. -L વિકલ્પ સાંકેતિક લિંક્સને અનુસરવાની ફરજ પાડે છે.

The pwd command

pwd command વર્તમાન નિર્દેશિકાના સંપૂર્ણ પાથનો અહેવાલ આપે છે. વર્તમાન ડિરેક્ટરી એ ડિરેક્ટરી છે જેમાં વપરાશકર્તા હાલમાં command વાક્ય ઈન્ટરફેસનો ઉપયોગ કરતી વખતે કાર્ય કરે છે. કમાન્ડ લાઈન ઈન્ટરફેસ એ ઓલ ટેક્સ્ટ ડિસ્પ્લે મોડ છે અને તે કન્સોલ (એટલે કે, ડિસ્પ્લે મોડ કે જેમાં આખી સ્ક્રીન ફક્ત ટેક્સ્ટ હોય છે) અથવા ટર્મિનલ વિન્ડો (એટલે કે, GUI માં ફક્ત ટેક્સ્ટ-વિન્ડો) દ્વારા પ્રદાન કરવામાં આવે છે. ડિરેક્ટરી અથવા fileનો સંપૂર્ણ પાથ, જેને સંપૂર્ણ પાથ પણ કહેવાય છે, તે રૂટ ડિરેક્ટરીથી તે ડિરેક્ટરી અથવા file સહિતની ડિરેક્ટરીઓનો સંપૂર્ણ વંશવેલો છે. રૂટ ડાયરેક્ટરી, જે ફોરવર્ડ સ્લેશ (/) દ્વારા નિયુક્ત કરવામાં આવે છે, તે fileસિસ્ટમ (એટલે કે, ડિરેક્ટરીઓનું વંશવેલો) પરની બેઝ ડિરેક્ટરી છે અને તે સિસ્ટમ પરની અન્ય તમામ ડિરેક્ટરીઓ, સબડિરેક્ટરીઝ અને ફાઈલો ધરાવે છે. આમ, કોઈપણ ડિરેક્ટરી અથવા file માટેનો સંપૂર્ણ માર્ગ હંમેશા ફોરવર્ડ સ્લેશથી શરૂ થાય છે. pwd એ Linux અને અન્ય યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમમાં ls સાથે સૌથી મૂળભૂત આદેશોમાંનો એક છે, જેનો ઉપયોગ વર્તમાન ડાયરેક્ટરી અને સીડીના સમાવિષ્ટોની સૂચિ બનાવો, જેનો ઉપયોગ વર્તમાન ડિરેક્ટરીને બદલવા માટે થાય છે

pwd's syntax is

pwd [option]

મોટા ભાગના આદેશોથી વિપરીત, pwd નો ઉપયોગ હંમેશા પોતાના દ્વારા જ થાય છે, એટલે કે,

Pwd

એટલે કે, તેનો ઉપયોગ તેના વિકલ્પો સાથે ભાગ્યે જ થાય છે અને ક્યારેય દલીલો સાથે ઉપયોગ થતો નથી (એટલે કે, fileના નામો અથવા ઈનપુટ તરીકે પ્રદાન કરેલી અન્ય માહિતી). pwd પછી સમાન લાઈન પર જે કંઈપણ ટાઈપ કરવામાં આવે છે, વિકલ્પના અપવાદ સિવાય, તેને અવગણવામાં આવે છે, અને કોઈ ભૂલ સંદેશાઓ પરત કરવામાં આવતા નથી.

ઉદાહરણ તરીકે, જો વપરાશકર્તાનામ janis સાથેનો વપરાશકર્તા તેની હોમ ડિરેક્ટરીમાં હોય, તો ઉપરોક્ત command સામાન્ય રીતે/home/janis/ પરત કરશે (કારણ કે, મૂળભૂત રીતે, બધી હોમ ડિરેક્ટરીઓ /home

ડિરેક્ટરીમાં સ્થિત છે). તેવી જ રીતે, જો વપરાશકર્તા હાલમાં ડિરેક્ટરી/usr/share/config (જેમાં સંખ્યાબંધ પ્રોગ્રામ રૂપરેખાંકન ફાઇલો હોય છે) માં કામ કરતા હોય, તો તે જ command /usr/share/ config પરત કરશે.

pwd એ પુષ્ટિ કરવા માટે ઉપયોગી છે કે વર્તમાન ડાયરેક્ટરી વાસ્તવમાં સીડીનો ઉપયોગ કર્યા પછી વપરાશકર્તાના હેતુમાં બદલાઈ ગઈ છે. ઉદાહરણ તરીકે, વર્તમાન ડિરેક્ટરીને /home/janis માંથી /usr/share/config માં બદલવા માટે cd command જારી કર્યા પછી, pwd નો ઉપયોગ પુષ્ટિ માટે થઈ શકે છે; એટલે કે, આદેશોનો નીચેનો ક્રમ જારી કરવામાં આવશે:

```
cd/usr/share/config/
```

```
pwd
```

pwd ના પ્રમાણભૂત સંસ્કરણમાં માત્ર બે વિકલ્પો છે, જે બંનેનો ઉપયોગ ભાગ્યે જ થાય છે. આ-help વિકલ્પનો ઉપયોગ નીચે પ્રમાણે થાય છે:

```
pwd --help
```

આ વિકલ્પ pwd વિશેની માહિતી દર્શાવે છે, જેમાંથી બહુ ઓછી છે કારણ કે તે આટલો સરળ command છે (એટલે કે, તેમાં ફક્ત બે વિકલ્પો છે અને કોઈ દલીલો સ્વીકારતી નથી). બીજો વિકલ્પ છે - વર્ઝન, જે વર્ઝન નંબર દર્શાવે છે, એટલે કે,

```
pwd --version
```

જો કે તે ઘણી વખત વર્તમાન વર્કિંગ ડિરેક્ટરી માટે સ્ટેન્ડિંગ તરીકે માનવામાં આવે છે, pwd વાસ્તવમાં પ્રિન્ટ વર્કિંગ ડિરેક્ટરીનું ટૂંકું નામ છે. પ્રિન્ટ શબ્દ એ લખવા અથવા ડિસ્પ્લે માટે પરંપરાગત UNIX પરિભાષા છે, અને તેનો ઉદ્ભવ ત્યારે થયો જ્યારે કમ્પ્યુટર આઉટપુટ સામાન્ય રીતે ડિફોલ્ટ રૂપે કાગળ પર છાપવામાં આવતું હતું કારણ કે CRT (કેથોડ રે ટ્યુબ) ડિસ્પ્લે મોનિટર હજુ સુધી વ્યાપકપણે ઉપલબ્ધ નહોતા.

The echo command

ઇકો એ બેશ અને સી શેલ્સમાં બિલ્ટ-ઇન કમાન્ડ છે જે તેની દલીલોને પ્રમાણભૂત આઉટપુટ પર લખે છે.

શેલ એ એક પ્રોગ્રામ છે જે Linux અને અન્ય યુનિક્સ પર કમાન્ડ લાઇન (એટલે કે ઓલ-ટેક્સ્ટ ડિસ્પ્લે યુઝર ઇન્ટરફેસ) પ્રદાન કરે છે. ઓપરેટિંગ સિસ્ટમ્સની જેમ. તે આદેશો ચલાવે છે (એટલે કે, ચાલે છે) જે તેમાં ટાઇપ કરવામાં આવે છે અને પરિણામો દર્શાવે છે. bash એ Linux પર મૂળભૂત શેલ છે. command એ કમ્પ્યુટરને કંઈક કરવા માટે કહેતી સૂચના છે. દલીલ એ command માટે ઇનપુટ ડેટા છે. માનક આઉટપુટ એ ડિફોલ્ટ રૂપે ડિસ્પ્લે સ્ક્રીન છે, પરંતુ તેને file, પ્રિન્ટર વગેરે પર રીડાયરેક્ટ કરી શકાય છે. ઇકો માટે સિન્ટેક્સ

```
$ echo $USER
```

```
$ echo "Hello world"
```

ચોરસ કૌંસમાંની વસ્તુઓ વૈકલ્પિક છે. શબ્દમાળા એ અક્ષરોનો કોઈપણ મર્યાદિત ક્રમ છે (એટલે કે, અક્ષરો, અંકો, પ્રતીકો અને વિરામચિહ્નો).

જ્યારે કોઈપણ વિકલ્પો અથવા શબ્દમાળાઓ વિના ઉપયોગ કરવામાં આવે છે, ત્યારે ઇકો ડિસ્પ્લે સ્ક્રીન પર એક ખાલી લાઇન પરત કરે છે અને ત્યારપછીની લાઇન પર કમાન્ડ પ્રોમ્પ્ટ આવે છે. આનું કારણ એ છે કે ENTER કી દબાવવી એ સિસ્ટમને નવી લાઇન શરૂ કરવા માટેનો સંકેત છે, અને આમ ઇકો આ સિગ્નલને પુનરાવર્તિત કરે છે.

જ્યારે એક અથવા વધુ શબ્દમાળાઓ દલીલો તરીકે પ્રદાન કરવામાં આવે છે, ત્યારે ઇકો મૂળભૂત રીતે સ્ક્રીન પર તે ડંખને પુનરાવર્તિત કરે છે. આમ, ઉદાહરણ તરીકે, નીચેનામાં ટાઇપ કરવાથી અને ENTER કી દબાવવાથી આ એક પેન વાક્યનું પુનરાવર્તન થશે. સ્ક્રીન પર:

```
echo This is a pen
```

શબ્દમાળાઓને અવતરણ સાથે ઘેરી લેવું જરૂરી નથી, કારણ કે તે સ્ક્રીન પર જે લખેલું છે તેને અસર કરતું નથી. જો અવતરણ (સિંગલ અથવા ડબલ) નો ઉપયોગ કરવામાં આવે છે, તો તે સ્ક્રીન પર પુનરાવર્તિત થતા નથી. સદભાગ્યે, ઇકો તેને અનુસરે છે તે શબ્દશ: પુનરાવર્તન કરતાં વધુ કરી શકે છે. એટલે કે, તે ચોક્કસ ચલનું મૂલ્ય પણ બતાવી શકે છે જો ચલનું નામ ડોલર અક્ષર (\$) દ્વારા સીધું (એટલે કે, કોઈ મધ્યસ્થી જગ્યા વિના) આગળ આવે છે, જે શેલને ચલના મૂલ્યને બદલવા માટે કહે છે. એનું નામ.

ઉદાહરણ તરીકે, x નામનું ચલ બનાવી શકાય છે અને તેની કિંમત નીચેના command સાથે 5 પર સેટ કરી શકાય છે:

```
x = 5
```

x ની કિંમત પછીથી નીચેના દ્વારા યાદ કરી શકાય છે:

```
echo The number is $x.
```

ઇકો ખાસ કરીને પર્યાવરણીય ચલોના મૂલ્યો બતાવવા માટે ઉપયોગી છે, જે શેલને જણાવે છે કે વપરાશકર્તા કમાન્ડ લાઇન પર અથવા સ્ક્રિપ્ટ્સ (ટૂંકા પ્રોગ્રામ્સ)માં કેવી રીતે કામ કરે છે તે રીતે કેવી રીતે વર્તવું.

ઉદાહરણ તરીકે, HOME નું મૂલ્ય જોવા માટે, પર્યાવરણીય મૂલ્ય જે વર્તમાન દર્શાવે છે વપરાશકર્તાની હોમ ડિરેક્ટરીમાં, નીચેનાનો ઉપયોગ કરવામાં આવશે:

```
echo $HOME
```

તેવી જ રીતે, echoનો ઉપયોગ વપરાશકર્તાના PATH પર્યાવરણીય ચલને બતાવવા માટે થઈ શકે છે, જેમાં ડિરેક્ટરીઓની કોલોનથી વિભાજિત સૂચિ હોય છે જેને સિસ્ટમ જારી command ને અનુરૂપ એક્ઝિક્યુટેબલ પ્રોગ્રામ શોધવા માટે શોધે છે. વપરાશકર્તા દ્વારા:

```
echo $PATH
```

echo, મૂળભૂત રીતે, નવી લાઇન અક્ષર સાથે કોઈપણ આઉટપુટને અનુસરે છે. આ એક બિન-પ્રિન્ટિંગ (એટલે કે, અદ્રશ્ય) અક્ષર છે જે ટેક્સ્ટની એક લીટીનો અંત અને પછીની શરૂઆતનું પ્રતિનિધિત્વ કરે છે. તે યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમમાં \n દ્વારા રજૂ થાય છે. પરિણામ એ છે કે અનુગામી કમાન્ડ પ્રોમ્પ્ટ ઇકો દ્વારા પરત આઉટપુટની સમાન લાઇનને બદલે નવી લાઇન પર શરૂ થાય છે. -e વિકલ્પનો ઉપયોગ નવી લાઇન કેરેક્ટરના વધારાના દાખલાઓના ઇકોના અર્થઘટન તેમજ અન્ય વિશિષ્ટ અક્ષરોના અર્થઘટનને સક્ષમ કરવા માટે થાય છે, જેમ કે આડી ટેબ, જે \t દ્વારા રજૂ થાય છે. આમ, ઉદાહરણ તરીકે, નીચેના ફોર્મેટ કરેલ આઉટપુટ ઉત્પન્ન કરશે:

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

(ઉપરનો command એક લીટી પર લખવો જોઈએ, જો કે તે નાની ડિસ્પ્લે સ્ક્રીન પર બે લીટીઓ તરીકે રેન્ડર થઈ શકે છે.) -n વિકલ્પનો ઉપયોગ ઇકોને આઉટપુટમાં નવી લાઇન ઉમેરવાથી રોકવા માટે કરી શકાય છે.

આઉટપુટ રીડાયરેક્શનનો ઉપયોગ કરીને, ઇકો એક નવી file બનાવવાની ખૂબ જ સરળ રીત પ્રદાન કરે છે જેમાં ટેક્સ્ટ હોય છે. ઇકો લખીને ઈચ્છિત ટેક્સ્ટ, આઉટપુટ રીડાયરેક્શન ઓપરેટર (જે રાઈટવર્ડ પોઈન્ટિંગ ઍંગલ બ્રેકેટ છે) અને છેલ્લે નવી fileનું નામ લખીને આ પરિપૂર્ણ થાય છે. ફાઇલ

પણ ખાસ અક્ષરોનો ઉપયોગ કરીને ફોર્મેટ કરી શકાય છે. આમ, ઉદાહરણ તરીકે, ઉપરના ઉદાહરણમાંથી ફોર્મેટ કરેલ આઉટપુટનો ઉપયોગ પ્રોજેક્ટ1 નામની નવી file બનાવવા માટે થઈ શકે છે:

```
echo -e "\n Project1: \n\n\tp\n \n\twrite \n\ttest\n" &gt; project1
```

નવી fileની સામગ્રી, કોઈપણ ફોર્મેટિંગ સહિત, કેટ અથવા ઓછા જેવા command નો ઉપયોગ કરીને ચકાસી શકાય છે, એટલે કે,

less project1

ઈકો એ એપેન્ડ ઓપરેટર સાથે તેનો ઉપયોગ કરીને fileના અંતમાં ટેક્સ્ટને જોડવાનો એક અનુકૂળ માર્ગ બની શકે છે, જે બે સળંગ જમણેરી પોઈન્ટિંગ એંગલ કૌંસ દ્વારા રજૂ થાય છે. જો કે, ત્યાં હંમેશા આકસ્મિક રીતે બેને બદલે એક જ કૌંસનો ઉપયોગ કરવાનું જોખમ રહેલું છે, જેનાથી ફાઈલની તમામ સામગ્રીઓ પર ફરીથી લખાઈ જાય છે, અને આમ, આ સુવિધા સ્ક્રિપ્ટમાં ઉપયોગ માટે શ્રેષ્ઠ રીતે આરક્ષિત છે. ઈકોનો ઉપયોગ પેટર્ન મેચિંગ સાથે પણ કરી શકાય છે, જેમ કે વાઈલ્ડકાર્ડ કેરેક્ટર, જે સ્ટાર કેરેક્ટર દ્વારા દર્શાવવામાં આવે છે. ઉદાહરણ તરીકે, નીચે આપેલ વાક્ય પરત કરશે gif ફાઈલો વર્તમાન નિર્દેશિકામાં તમામ .gif ઈમેજ ફાઈલોના નામ દ્વારા અનુસરવામાં આવે છે:

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

The cal command

વર્તમાન મહિનાનું કેલેન્ડર દર્શાવે છે.

જુલાઈ 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’ ઉલ્લેખિત મહિના અને વર્ષ માટે કેલેન્ડર પ્રદર્શિત કરશે.

```
$ cal 08 1991
```

ઓગસ્ટ 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 command

વર્તમાન સમય અને date દર્શાવો.

```
$ date
```

```
Fri Jul 6 01:07:09 IST 2012
```

જો તમને માત્ર સમય માં જ રસ હોય, તો તમે ‘date +%T’ (hh:mm:ss માં):

```
$ date +%T
```

```
01:13:14
```

tty command નો ઉપયોગ કરી શકો છો

વર્તમાન ટર્મિનલ દર્શાવે છે.

```
$ tty
```

```
/dev/pts/0
```

whoami command

આ command વર્તમાન લૉગ ઈન થયેલ વપરાશકર્તાને દર્શાવે છે.

```
$ whoami
```

```
raghu
```

id command

આ command વર્તમાન વપરાશકર્તાના વપરાશકર્તા અને જૂથો (UID અને GID) છાપે છે.

```
$ id
```

```
uid=1000(raghu) gid=1000(raghu)
```

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

વર્તમાન વપરાશકર્તા વિશે મૂળભૂત માહિતી પ્રદર્શિત થાય છે. જો અન્ય વપરાશકર્તાનામ દલીલ તરીકે પ્રદાન કરવામાં આવે, તો તે વપરાશકર્તા વિશેની માહિતી છાપવામાં આવશે:

```
$ id root
```

```
uid=0(root) gid=0(root) group=0(root)
```

Clear command

આ command સ્ક્રીનને સાફ કરે છે.

Getting help command

તેના તમામ ફાયદાઓ માટે, કમાન્ડ લાઈનનો એક મોટો ગેરલાભ એ છે કે ત્યાં ઘણા બધા આદેશો છે અને તેનાથી પણ વધુ તેમના વિકલ્પો અને ઉપયોગ છે. પરંતુ કોઈ પણ બધા આદેશો યાદ રાખી શકતું નથી. command વાક્યનો ઉપયોગ કરવાની કેટલીક સ્માર્ટ રીતો છે. Linux અમને અહીં ચર્ચા કરેલ આવા ઘણા સંસાધનો પૂરા પાડે છે.

: --help option

લગભગ દરેક command સાથે, ‘--help’ વિકલ્પ તે command માટે વપરાશ સારાંશ દર્શાવે છે

```
$ date --help
```

```
Usage: date [OPTION]... [+FORMAT]
```

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

વર્તમાન સમય દર્શાવો આપેલ ફોર્મેટ, અથવા સિસ્ટમ date સેટ કરો.

The whatis command

whatis command કમાન્ડ લાઈન પ્રોગ્રામ્સ (એટલે કે, તમામ ટેક્સ્ટ મોડ પ્રોગ્રામ્સ) અને Linux અને અન્ય યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ્સ સંબંધિત અન્ય વિષયોનું ખૂબ જ સંક્ષિપ્ત વર્ણન પ્રદાન કરે છે.

તે દલીલ તરીકે તેને આપવામાં આવેલ દરેક કીવર્ડ માટે ડેટાબેઝ શું છે તેમાં ટૂંકા વર્ણનો શોધીને આ પરિપૂર્ણ કરે છે (એટલે કે, ઇનપુટ ડેટા). આ ડેટાબેઝમાં મેન્યુઅલમાં દરેક પૃષ્ઠના NAME વિભાગમાંથી ફક્ત શીર્ષક, વિભાગ નંબર અને વર્ણન છે જે મોટાભાગની યુનિક્સ જેવી સિસ્ટમમાં બનેલ છે.

The syntax for whatis is:

whatis keyword(s)

ઉદાહરણ તરીકે, નીચે આપેલ હેડ કમાન્ડનો સિંગલ લાઇન સારાંશ પૂરો પાડે છે (જે ડિફોલ્ટ રૂપે દરેક fileની પ્રથમ દસ લીટીઓ દર્શાવે છે જે તેને દલીલ તરીકે પૂરી પાડવામાં આવે છે):

whatis head

શું છે તેનો ઉપયોગ એકસાથે બહુવિધ વિષયો વિશેની માહિતી શોધવા માટે થઈ શકે છે. ઉદાહરણ તરીકે, નીચેના માથા અને પૂંછડી બંને વિશે માહિતી પ્રદાન કરશે (જે મૂળભૂત રીતે ફાઇલોની અંતિમ દસ લાઇન વાંચે છે):

whatis head tail

તેનું આઉટપુટ શું છે તે હકીકત દ્વારા મર્યાદિત છે કે તે ડેટાબેઝમાં જોવા મળતા દરેક કીવર્ડ માટે માત્ર એક લીટી પ્રદાન કરે છે; આમ તે સાધારણ જટિલ આદેશો વિશે અધૂરી માહિતી પૂરી પાડે છે. ઉદાહરણ તરીકે, બિલાડી કમાન્ડ વિશેની માહિતી મેળવવા માટે શું છે તેનો નીચેનો ઉપયોગ આઉટપુટ “કંકેટનેટ ફાઇલો અને સ્ટાન્ડર્ડ આઉટપુટ પર પ્રિન્ટ” જનરેટ કરે છે:

whatis cat

જો કે, આ બિલાડી વિશેની કેટલીક મહત્વપૂર્ણ માહિતીને છોડી દે છે, ખાસ કરીને હકીકતો કે તે ખૂબ જ છે. ફાઇલો વાંચવા માટે વાપરવા માટે અનુકૂળ છે અને તેનો ઉપયોગ ફાઇલો બનાવવા અને લખવા માટે પણ થાય છે.

જે apropos command જેવું જ છે. જો કે, એપ્રોપોસ વધુ શક્તિશાળી છે કારણ કે તેની દલીલો સંપૂર્ણ શબ્દો સુધી મર્યાદિત નથી પરંતુ તે શબ્દમાળાઓ (એટલે કે, અક્ષરોના કોઈપણ મર્યાદિત ક્રમ) પણ હોઈ શકે છે જેમાં શબ્દોના ભાગોનો સમાવેશ થાય છે. બંને આદેશો અસામાન્ય છે કે તેમની પાસે કોઈ વિકલ્પો નથી. મેન કમાન્ડ (જેનો ઉપયોગ બિલ્ટ-ઇન

મેન્યુઅલ પેજીસને વાંચવા માટે થાય છે), જ્યારે તેના -f વિકલ્પ સાથે ઉપયોગ કરવામાં આવે છે, ત્યારે whatis જેવું જ આઉટપુટ ઉત્પન્ન કરે છે. આમ, ઉદાહરણ તરીકે,

man -f cat

એ

whatis cat

Info pages

માહિતી દસ્તાવેજો કેટલીકવાર મેન પૃષ્ઠો કરતાં વધુ વિસ્તૃત હોય છે. પરંતુ કેટલાક આદેશો માટે, માહિતી પૃષ્ઠો મેન પૃષ્ઠો જેવા જ છે. આ વેબ પેજીસ જેવા છે. આંતરિક લિંક્સ માહિતી પૃષ્ઠોમાં હાજર છે. આ લિંક્સને નોંડ સ કહેવામાં આવે છે. આ નોંડ દ્વારા માહિતી પૃષ્ઠોને એક પૃષ્ઠથી બીજા પૃષ્ઠ પર નેવિગેટ કરી શકાય છે.

\$ info date

Word processors in the Linux environment

ટેક્સ્ટ એડિટર્સનો ઉપયોગ વિવિધ પ્રકારના લોકો દ્વારા કરવામાં આવે છે. કેઝ્યુઅલ યુઝર્સ, લેખકો, પ્રોગ્રામરો અને સિસ્ટમ એડમિનિસ્ટ્રેટર્સ બધા અહીં ટેક્સ્ટ એડિટરનો ઉપયોગ કરશે Linux માં એક અથવા બીજી વાર.

Use of text editor

ટેક્સ્ટ એડિટર ઘણી બધી વિશેષતાઓ વિના વર્ડ પ્રોસેસરની જેમ છે. બધી ઓપરેટિંગ સિસ્ટમ મૂળભૂત ટેક્સ્ટ એડિટર સાથે આવે છે. Linux અનેક સાથે આવે છે. ટેક્સ્ટ એડિટરનો મુખ્ય ઉપયોગ કોઈ ફોર્મેટિંગ વિના સાદા ટેક્સ્ટમાં કંઈક લખવા માટે છે જેથી અન્ય પ્રોગ્રામ તેને વાંચી શકે. તે fileમાંથી મેળવેલી માહિતીના આધારે, પ્રોગ્રામ એક અથવા બીજી રીતે ચાલશે.

Movement commands:

તમારી સિસ્ટમ પર આધાર રાખીને, એરો કી અથવા બેકસ્પેસ કી કામ કરી શકશે નહીં. તેના બદલે, તમે સમાન કાર્યો કરવા માટે આ આદેશોનો ઉપયોગ કરી શકો છો.

પ્રતિ	Ctrl કી દબાવી રાખો અને દબાવો	ની બદલે
એક અક્ષર કાઢી નાખો	Back space	Back space
એક લીટી ઉપર ખસેડો	P	Up arrow
એક લીટી નીચે ખસેડો	n	Down arrow
એક જગ્યા ડાબી બાજુ ખસેડો	b	Left arrow
એક જગ્યા જમણી બાજુ ખસેડો	f	Right arrow
લીટીના અંત સુધી જાઓ	e	end

ફાઇલો સાથે કામ કરવું : આ પ્રકરણમાં આપણે શીખીએ છીએ કે file, touch, આરએમ, સીપી, એમવી અને નામ બદલવું વગેરે જેવા આદેશોનો ઉપયોગ કરીને ફાઇલોને કેવી રીતે ઓળખવી, બનાવવી, દૂર કરવી, કોપી કરવી અને ખસેડવી...

બધી ફાઇલો કેસ સેન્સિટિવ છે : લિનક્સ પરની ફાઇલો (અથવા કોઈપણ યુનિક્સ) કેસ સેન્સિટિવ છે.

આનો અર્થ એ છે કે FILE1 એ file1 થી અલગ છે, અને /etc/hosts / etc/Hosts થી અલગ છે (બાદનું એક લાક્ષણિક Linux કમ્પ્યુટર પર અસ્તિત્વમાં નથી).

file command : file command દરેક file સિસ્ટમ ઓબ્જેક્ટ (એટલે કે, file, ડિરેક્ટરી અથવા લિંક) ને વર્ગીકૃત કરવાનો પ્રયાસ કરે છે જે તેને દલીલ (એટલે કે ઇનપુટ) તરીકે પ્રદાન કરવામાં આવે છે. આમ, તે સામાન્ય રીતે તાત્કાલિક માહિતી પ્રદાન કરી શકે છે કે શું અમુક ચોક્કસ ઓબ્જેક્ટ છે, ઉદાહરણ તરીકે, GIF89a ઇમેજ file, ડિરેક્ટરી, એક GNU ટાર આર્કાઇવ, ASCII અંગ્રેજી ટેક્સ્ટ, એક સાંકેતિક લિંક, એક HTML દસ્તાવેજ, એક ખાલી file, bzip2 સંકુચિત ડેટા, એક ELF 32-bit LSB એકિઝક્યુટેબલ, વગેરે.

એક સફળ ન થાય ત્યાં સુધી file ત્રણ પ્રકારના પરીક્ષણો સાથે દરેક ઓબ્જેક્ટની તપાસ કરીને આ પરિપૂર્ણ કરે છે. પ્રથમ file સિસ્ટમ પરીક્ષણ છે, જે ઓબ્જેક્ટના આઈનોડ (જે file વિશેની માહિતી ધરાવે છે) માંથી માહિતી મેળવવા માટે સ્ટેટ સિસ્ટમ કૉલનો ઉપયોગ કરે છે. સિસ્ટમ કૉલ એ યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમમાં કર્નલ (એટલે કે ઓપરેટિંગ સિસ્ટમનો મુખ્ય ભાગ) દ્વારા કરવામાં આવતી સેવા માટેની વિનંતી છે.

બીજી કસોટી એ જોવા માટે તપાસે છે કે શું કોઈ જાદુઈ નંબર છે, જે ઘણા પ્રકારની ફાઇલોની શરૂઆતમાં અથવા તેની નજીકમાં જડાયેલો નંબર છે જે ફાઇલ ફોર્મેટ (એટલે કે, ફાઇલનો પ્રકાર) સૂચવે છે.

પ્રથમ બે પરીક્ષણો fileના પ્રકારને નિર્ધારિત કરવામાં નિષ્ફળ જાય તેવી સ્થિતિમાં, તે સાદો લખાણ છે કે કેમ તે નિર્ધારિત કરવા માટે ભાષા પરીક્ષણોનો ઉપયોગ કરવામાં આવે છે (એટલે કે, સંપૂર્ણ રીતે માનવ વાંચી શકાય તેવા અક્ષરોથી બનેલું), અને, જો એમ હોય તો, કેવા પ્રકારનો સાદો લખાણ, જેમ કે HTML (હાયપરટેક્સ્ટ માર્કઅપ ભાષા) અથવા સ્ત્રોત કોડ (એટલે કે, માનવ દ્વારા લખાયેલ પ્રોગ્રામનું મૂળ સંસ્કરણ). આ સ્થિતિમાં, file કુદરતી ભાષા (દા.ત., અંગ્રેજી, તુર્કી અથવા જાપાનીઝ) નક્કી કરવાનો પણ પ્રયાસ કરે છે જેનો ઉપયોગ fileમાં થાય છે.

fileના વાક્યરચનાનું સરળ સંસ્કરણ

file [વિકલ્પ(ઓ)] object_name(s)

fileમાં ઘણા બધા વિકલ્પો છે, પરંતુ તે સામાન્ય રીતે તેમાંના કોઈપણ વિના ઉપયોગમાં લેવાય છે. ઉદાહરણ તરીકે, file1 નામની file વિશેની માહિતી જે વર્તમાન ડિરેક્ટરીમાં સ્થિત છે (એટલે કે, જે ડિરેક્ટરીમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે) ફક્ત નીચે લખીને અને રીટર્ન કી દબાવીને મેળવી શકાય છે:

file file1

વર્તમાન નિર્દેશિકામાંની તમામ ફાઇલોના પ્રકારો વિશેની માહિતી તે નિર્દેશિકામાંના દરેક ઓબ્જેક્ટને નીચે પ્રમાણે રજૂ કરવા માટે સ્ટાર વાઈલ્ડકાર્ડનો ઉપયોગ કરીને મેળવી શકાય છે:

file *

તેવી જ રીતે, અન્ય ડિરેક્ટરીમાંની તમામ ફાઇલો વિશેની માહિતી તે ડિરેક્ટરીનો દલીલ તરીકે ઉપયોગ કરીને અને તેને ફોરવર્ડ સ્લેશ અને સ્ટાર વાઈલ્ડકાર્ડ દ્વારા તરત જ અનુસરીને મેળવી શકાય છે. ઉદાહરણ તરીકે, નીચેના /boot ડિરેક્ટરીમાં તમામ ઓબ્જેક્ટનું વર્ગીકરણ કરે છે:

file/boot/*

ચોરસ કૉસ વાઈલ્ડકાર્ડનો ઉપયોગ સ્ટાર વાઈલ્ડકાર્ડ સાથે મળીને માત્ર એવા ઓબ્જેક્ટ્સ માટે file પ્રકારો બતાવવા માટે કરી શકાય છે જેમના નામ સ્પષ્ટ અક્ષરોથી અથવા અક્ષરોની ચોક્કસ શ્રેણી સાથે શરૂ થાય છે. ઉદાહરણ તરીકે, નીચે આપેલ વર્તમાન ડિરેક્ટરીમાં ફક્ત તે જ ઓબ્જેક્ટ્સ બતાવશે જેમના નામ a થી g:

file [a g]*

અક્ષરોથી શરૂ થાય છે.-k વિકલ્પ fileને પ્રથમ સફળ કસોટી પર રોકવા માટે નહીં, પરંતુ ચાલુ રાખવા માટે કહે છે; આ અમુક file સિસ્ટમ ઓબ્જેક્ટ્સ વિશે વધારાની માહિતીના અહેવાલમાં પરિણમી શકે છે. -b (એટલે કે, સંક્ષિપ્ત) વિકલ્પ fileને આઉટપુટ લાઈનમાં fileનામોને આગળ ન રાખવા માટે કહે છે, જે file પ્રકારો વિશે આંકડાઓનું સંકલન કરતી વખતે ઉપયોગી થઈ શકે છે. -v વિકલ્પ ના સંસ્કરણ વિશે માહિતી આપે છે file કે જે ઇન્સ્ટોલ કરેલ છે.

ફાઇલો અને ડિરેક્ટરીઓ બનાવવી

mkdir command

mkdir command નો ઉપયોગ નવી ડિરેક્ટરીઓ બનાવવા માટે થાય છે.

અમુક ઓપરેટિંગ સિસ્ટમમાં ફોલ્ડર તરીકે ઓળખાતી ડિરેક્ટરી, અન્ય ડિરેક્ટરીઓ અને ફાઇલો માટે કન્ટેનર તરીકે વપરાશકર્તાને દેખાય છે. જો કે, યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ ડાયરેક્ટરીઝને માત્ર એક ખાસ પ્રકારની file તરીકે માને છે જેમાં fileના નામ અને તેના અનુરૂપ ઇનોડ નંબરોની સૂચિ હોય છે. દરેક ઇનોડ નંબર એ આઈનોડનો સંદર્ભ આપે છે, જે આઈનોડ કોષ્ટકોમાં સ્થિત છે (જે file સિસ્ટમની આસપાસ વ્યૂહાત્મક સ્થાનો પર રાખવામાં આવે છે) અને જેમાં file વિશેની તમામ માહિતી (દા.ત., કદ, પુરવાનગીઓ અને બનાવટની date) સિવાય તેના નામ અને વાસ્તવિક ડેટા કે જે file સમાવે છે

mkdir પાસે નીચેનું ઉદાહરણ છે

\$ mkdir ઉદાહરણ

\$ls -l

total 4

drwxr-xr-x 2 raghu raghu 4096 2012 07-06 14:09 ઉદાહરણ

Directory_name એ કોઈપણ ડિરેક્ટરીનું નામ છે જે વપરાશકર્તા mk-
dir ને બનાવવા માટે કહે છે. કોઈપણ સંખ્યાની ડિરેક્ટરીઓ એકસાથે બનાવી શકાય છે.

આમ, ઉદાહરણ તરીકે, નીચેનો command વર્તમાન ડિરેક્ટરીમાં ત્રણ ડિરેક્ટરીઓ બનાવશે (એટલે કે, ડિરેક્ટરી જેમાં વપરાશકર્તા હાલમાં કામ કરે છે) dir_1, dir_2 અને dir_3:

mkdir dir_1 dir_2 dir_3

જો mkdir ને દલીલ (એટલે કે ઇનપુટ) તરીકે આપવામાં આવેલ ડિરેક્ટરીનું નામ એ જ ડિરેક્ટરીમાં અસ્તિત્વમાંની ડિરેક્ટરી અથવા file જેવું જ છે જેમાં વપરાશકર્તા mkdir ને નવી ડિરેક્ટરી બનાવવા માટે પૂછે છે, તો mkdir એક ચેતવણી સંદેશ આપશે જેમ કે mkdir તરીકે: ડિરેક્ટરી 'dir_1' બનાવી શકતા નથી:

file અસ્તિત્વમાં છે અને તે નામની file બનાવશે નહીં. જો કે, તે પછી દલીલો તરીકે પૂરા પાડવામાં આવેલ કોઈપણ અન્ય નામો માટે ડિરેક્ટરીઓ બનાવવાનું ચાલુ રાખશે. વપરાશકર્તાને સક્ષમ થવા માટે પેરેન્ટ ડાયરેક્ટરી

(એટલે કે જે ડિરેક્ટરીમાં નવી ડિરેક્ટરી બનાવવાની છે) માં લખવાની પરવાનગી (એટલે કે, file અથવા ડિરેક્ટરી બનાવવા અથવા બદલવા માટે સિસ્ટમની પરવાનગી) હોવી જરૂરી છે. નવી ડિરેક્ટરી બનાવવા માટે.

mkdir દ્વારા બનાવવામાં આવેલી ડિરેક્ટરીઓમાં આપમેળે બે છુપાયેલી ડિરેક્ટરીઓનો સમાવેશ થાય છે, એક હમણાં જ બનાવેલી ડિરેક્ટરીનું પ્રતિનિધિત્વ કરે છે (અને એક ટપકાં દ્વારા રજૂ થાય છે) અને બીજી તેની પેરેન્ટ ડિરેક્ટરીનું પ્રતિનિધિત્વ કરે છે (અને સળંગ બે બિંદુઓ દ્વારા રજૂ થાય છે). આને તેના -a વિકલ્પ સાથે ls (એટલે કે, સૂચિ) command નો ઉપયોગ કરીને જોઈ શકાય છે, જે તેને દલીલ તરીકે પૂરી પાડવામાં આવેલ કોઈપણ ડિરેક્ટરીમાં અથવા વર્તમાન ડિરેક્ટરીમાં બધી ડિરેક્ટરીઓ અને ફાઇલો (છુપાયેલા સહિત) બતાવવાનું કહે છે. જો ત્યાં કોઈ દલીલો ન હોય, એટલે કે,

```
ls -a
```

mkdir's -m વિકલ્પનો ઉપયોગ નવી ડિરેક્ટરીઓની પરવાનગીઓને નિયંત્રિત કરવા માટે થાય છે. નવી ડિરેક્ટરીઓ મૂળભૂત રીતે rીડ, રાઈટ અને એક્ઝિક્યુટ (એટલે કે, પ્રોગ્રામ તરીકે ચલાવવામાં આવે તો) માલિક (એટલે કે, ડિફોલ્ટ રૂપે ડિરેક્ટરીના નિર્માતા) અને જૂથ માટે સક્ષમ પરવાનગીઓ અને અન્ય માટે સક્ષમ કરેલી વાંચન અને અમલ પરવાનગીઓ સાથે બનાવવામાં આવે છે. વપરાશકર્તાઓ આમ, ઉદાહરણ તરીકે, dir_4 નામની ડિરેક્ટરી બનાવવા માટે જેના માટે તમામ વપરાશકર્તાઓ માટે ત્રણેય પ્રકારની પરવાનગીઓ સક્ષમ કરવામાં આવી હતી, ક્રમ 777 નો ઉપયોગ -m પછી કરવામાં આવશે, ઉદાહરણ તરીકે:

```
mkdir -m 777 dir_4
```

પ્રથમ અંક માલિકનું પ્રતિનિધિત્વ કરે છે, બીજો રજૂ કરે છે જૂથ અને ત્રીજું અન્ય વપરાશકર્તાઓને રજૂ કરે છે. સંખ્યા 7 એ ત્રણેય પ્રકારની પરવાનગી રજૂ કરે છે (એટલે કે, વાંચો, લખો અને એક્ઝિક્યુટ), 6 એટલે ફક્ત વાંચવા અને લખવા માટે, 5 સ્ટેન્ડવાંચવા અને ચલાવવા માટે, 4 ફક્ત વાંચવા માટે છે, 3 લખવા અને ચલાવવા માટે છે, 2 એ ફક્ત લખવાનું છે, 1 માત્ર એક્ઝિક્યુટ છે અને 0 કોઈ પરવાનગી નથી.

આમ, ઉદાહરણ તરીકે, dir_5 નામની નવી ડિરેક્ટરી બનાવવા માટે જેના માટે માલિકે વાંચવા અને લખવાની પરવાનગીઓ છે, જૂથ પાસે વાંચવાની પરવાનગી છે અને અન્ય વપરાશકર્તાઓ પાસે નથીરવાનગીઓ, ધ નીચેનાનો ઉપયોગ કરવામાં આવશે:

```
mkdir -m 640 dir_5
```

-p (એટલે કે, માતાપિતા) વિકલ્પ ઉલ્લેખિત બનાવે છે નવી ડિરેક્ટરી માટે મધ્યવર્તી ડિરેક્ટરીઓ જો તે ન હોય તો પહેલેથી જ અસ્તિત્વમાં. ઉદાહરણ તરીકે, તે બનાવવા માટે વાપરી શકાય છે નીચેની ડિરેક્ટરી માળખું:

```
mkdir -p food/fruit/citrus/oranges
```

દલીલ તરીકે પ્રથમ ડિરેક્ટરીના નામ સાથે du (એટલે કે, ડિસ્ક વપરાશ) command નો ઉપયોગ કરીને ડિરેક્ટરીઓની આ શ્રેણી બનાવવામાં આવી છે તેની પુષ્ટિ કરવી ખૂબ જ સરળ છે. ઉપરોક્ત ઉદાહરણના કિસ્સામાં આ

```
du food
```

હશે અન્ય વિકલ્પોમાં -v (એટલે કે વર્બોઝ) નો સમાવેશ થાય છે, જે દરેક બનાવેલ નિર્દેશિકા માટે સંદેશ આપે છે, --help, જે mkdir વિશે સંક્ષિપ્ત માહિતી આપે છે, અને - સંસ્કરણ, જે પરત કરે છે. હાલમાં ઇન્સ્ટોલ કરેલ mkdir પ્રોગ્રામ touch કમાન્ડનો વર્ઝન નંબર:

Touch command: દરેક FILE ના એક્સેસ અને ફેરફારના સમયને વર્તમાન સિસ્ટમ સમય પર અપડેટ કરે છે.

જો તમે એવી FILE નો ઉલ્લેખ કરો કે જે agooથી અસ્તિત્વમાં નથી, તો touch એ નામની ખાલી file બનાવે છે.

જો FILE દલીલ ડેશ હોય તો ("") વિશેષ રીતે હેન્ડલ કરવામાં આવે છે અને પ્રમાણભૂત આઉટપુટ સાથે સંકળાયેલ fileના સમયને બદલવા માટે touchનું કારણ બને છે.

```
$ touch file1 file2 file3
```

```
$ ls -l
```

```
total 4
```

```
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 ઉદાહરણ
```

```
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1
```

```
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2
```

```
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3
```

touchની એક વિશેષતા એ છે કે, cp (જે ફાઇલો અને ડિરેક્ટરીઓની નકલ કરવા માટે વપરાય છે) અને mv (જે ફાઇલો અને ડિરેક્ટરીઓને ખસેડવા અથવા તેનું નામ બદલવા માટે વપરાય છે) જેવા કેટલાક આદેશોથી વિપરીત, તે આપમેળે ઓવરરાઈટ થતું નથી (એટલે કે, ભૂંસી નાખે છે. ની સામગ્રીઓ) સમાન નામ સાથેની હાલની ફાઇલો. તેના બદલે, તે આવી ફાઇલો માટેના છેલ્લા એક્સેસ સમયને વર્તમાન સમયમાં બદલે છે.

touchના કેટલાક વિકલ્પો ખાસ કરીને વપરાશકર્તાને ફાઇલો માટે ટાઈમસ્ટેમ્પ બદલવાની મંજૂરી આપવા માટે ડિઝાઈન કરવામાં આવ્યા છે. ઉદાહરણ તરીકે, -a વિકલ્પ માત્ર બદલાય છે એક્સેસ સમય, જ્યારે -m વિકલ્પ માત્ર ફેરફાર સમય બદલે છે. આ બંને વિકલ્પોનો એકસાથે ઉપયોગ કરવાથી એક્સેસ અને ફેરફારના સમય બંને વર્તમાન સમયમાં બદલાય છે, ઉદાહરણ તરીકે:

```
Touch -am file 3
```

-r (એટલે કે, સંદર્ભ) વિકલ્પ સીધા જ સ્પેસ દ્વારા અનુસરવામાં આવે છે અને પછી ફાઇલ નામ દ્વારા ટચને વર્તમાન સમયને બદલે તે ફાઇલના ટાઈમ સ્ટેમ્પનો ઉપયોગ કરવાનું કહે છે. ઉદાહરણ તરીકે, નીચેના તેને file5 માટે file4 ના સમયનો ઉપયોગ કરવાનું કહેશે:

```
touch -r file4 file5
```

-B વિકલ્પ ચોક્કસ સેકન્ડની સંખ્યા પાછળ જઈને ટાઈમસ્ટેમ્પમાં ફેરફાર કરે છે, અને -F વિકલ્પ સેકન્ડની નિર્દિષ્ટ સંખ્યામાં આગળ જઈને સમયને સુધારે છે. ઉદાહરણ તરીકે, નીચેનો આદેશ file7 ને file6 કરતા 30 secondsજુનો બનાવશે.

```
touch -r file6 -B 30 file7
```

-d અને -t વિકલ્પો વપરાશકર્તાને ચોક્કસ છેલ્લા એક્સેસ સમય ઉમેરવા માટે પરવાનગી આપે છે. agooની date, મહિનો, વર્ષ, મિનિટ: secondsફોર્મેટમાં સ્ટ્રિંગ (એટલે કે અક્ષરોનો ક્રમ) આવે છે અને બાદમાં [[CC] YY]MMDDhhmm[.ss] ફોર્મેટનો ઉપયોગ કરે છે. ઉદાહરણ તરીકે, 1 મે, 2005, 1 મે 2005 10:22 એ ફાઇલ 8 ના છેલ્લા એક્સેસ ટાઈમને સવારે 10:22 થી બદલવા માટે 10:22 એક અવતરણ ચિહ્નોમાં બંધ કરવામાં આવશે અને તેનો ઉપયોગ નીચે પ્રમાણે કરવામાં આવશે, એટલે કે:

```
touch -d '1 May 2005 10:22' file8
```

આંશિક date-સમય શબ્દમાળાઓનો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે, નીચેની ફાઇલ9 માટે બતાવ્યા પ્રમાણે માત્ર date જ પ્રદાન કરવાની જરૂર છે (જે કિસ્સામાં સમય આપોઆપ 0:00 પર સેટ થાય છે):

touch -d '14 May' file9

ફક્ત સમય પૂરો પાડવો, નીચે બતાવ્યા પ્રમાણે, આપોઆપdateને વર્તમાન dateમાં બદલો:

touch -d '14:24' file9

ફાઇલો માટે છેલ્લી ફેરફાર date જોવા માટે સૌથી સામાન્ય રીતે ઉપયોગમાં લેવાતી રીત એ છે કે તેના નામની સાથે ls command નો ઉપયોગ કરવો. ઉદાહરણ તરીકે, file10 નામની file ના કિસ્સામાં આ હશે

ls -l file10

સ્ટેટ કમાન્ડનો ઉપયોગ કરીને કોઈપણ file અથવા ડિરેક્ટરી માટે સંપૂર્ણ ટાઇમસ્ટેમ્પ જોઈ શકાય છે. ઉદાહરણ તરીકે, નીચેના file11 નામની file માટે ટાઇમસ્ટેમ્પ બતાવશે:

stat file11

- હેલ્પ વિકલ્પ વિકલ્પોની મૂળભૂત સૂચિ દર્શાવે છે, અને --version વિકલ્પ હાલમાં ઈન્સ્ટોલ કરેલ touch પ્રોગ્રામનું સંસ્કરણ પરત કરે છે.

Copy, move, and remove commands

Copy command : cp command નો ઉપયોગ ફાઇલો અને ડિરેક્ટરીઓ કૉપી કરવા માટે થાય છે. નકલો મૂળથી સ્વતંત્ર બને છે (એટલે કે, એકમાં અનુગામી ફેરફાર બીજાને અસર કરશે નહીં).

Cp's basic syntax is

\$cp source destination

સલામતીની સાવચેતી તરીકે, મૂળભૂત રીતે cp ફક્ત ફાઇલોની નકલ કરે છે અને ડિરેક્ટરીઓની નહીં. જો file ની નકલને સૉપેલ સમાન નામવાળી file (અથવા ડિરેક્ટરીની નકલને સૉપેલ સમાન નામની ડિરેક્ટરી) અગ્રોથી જ અસ્તિત્વમાં છે, તો તે ઓવરરાઇટ થઈ જશે (એટલે કે, તેની સામગ્રી ખોવાઈ જશે). જો કે, નકલ માટે માલિક, જૂથ અને પરવાનગીઓ તે જ નામની file ની જેમ જ બની જાય છે જેને તેણે બદલ્યું હતું. સ્રોત file નો છેલ્લો એક્સેસ સમય અને નવી file નો છેલ્લો ફેરફાર સમય કૉપી કરવામાં આવ્યો તે સમય પર સેટ છે.

જ્યારે કૉપી file અથવા ડિરેક્ટરીની બનેલી હોય, ત્યારે કૉપિનું મૂળ નામ કરતાં અલગ નામ હોવું આવશ્યક છે જો તે મૂળ જેવી જ ડિરેક્ટરીમાં મૂકવાની હોય. જો કે, કૉપીનું નામ સમાન હોઈ શકે છે જો તે કોઈ અલગ ડિરેક્ટરીમાં બનાવવામાં આવી હોય. આમ, ઉદાહરણ તરીકે, વર્તમાન ડાયરેક્ટરી (એટલે કે, જે ડિરેક્ટરીમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે) file1 નામની file તેની સાથે કૉપી કરી શકાય છે. અન્ય ડિરેક્ટરીમાં નામ, જેમ કે in/home/john/, નીચે પ્રમાણે:

cp file1 /home/john/file1

કોઈપણ ફાઇલોની એકસાથે અન્ય ડિરેક્ટરીમાં તેમના નામની યાદી બનાવીને ડિરેક્ટરીના નામની સાથે કૉપી કરી શકાય છે. cp એ એક બુદ્ધિશાળી command છે અને જ્યારે માત્ર અંતિમ દલીલ (એટલે કે, ઈનપુટ ડેટાનો ભાગ) ડિરેક્ટરી હોય ત્યારે આ કરવાનું જાણે છે. ડાયરેક્ટરીમાં કૉપી કરેલી ફાઇલોના મૂળ નામના જ નામ હશે. આમ, ઉદાહરણ તરીકે, નીચે આપેલ file2, file3 અને file4 નામની ફાઇલોને dir1 નામની ડિરેક્ટરીમાં કૉપી કરશે:

cp file2 file3 file4 dir1

The -r (એટલે કે, પુનરાવર્તિત) વિકલ્પ, જેને મોટા કેસ R સાથે પણ લખી શકાય છે, પરવાનગી આપે છે. કૉપી કરવા માટેની તેમની તમામ સામગ્રી સહિત ડિરેક્ટરીઓ. (ડિરેક્ટરીઝ ડિફોલ્ટ રૂપે કૉપી કરવામાં આવતી નથી

જેથી વપરાશકર્તાઓ માટે આકસ્મિક રીતે અસ્તિત્વમાંની ડિરેક્ટરીઓ પર ફરીથી લખવાનું વધુ મુશ્કેલ બને કે જેનું નામ કૉપી કરવા માટે અસાઈન કરેલા સમાન હોય અને જેમાં નિર્ણાયક ડિરેક્ટરી સ્ટ્રક્ચર્સ અથવા મહત્વપૂર્ણ ડેટા હોઈ શકે.) આમ, ઉદાહરણ તરીકે, નીચેનો command વર્તમાનની નકલ બનાવશે dir2 નામની ડાયરેક્ટરી, તેમાં તમામ સમાવિષ્ટો (એટલે કે, ફાઇલો, સબડિરેક્ટરીઝ, તેમની સબડિરેક્ટરીઝ, વગેરે), જેને dir3 કહેવાય છે:

cp -r dir2 dir3

-i (એટલે કે, ઈન્ટરેક્ટિવ) વિકલ્પ વપરાશકર્તાને એવી ઘટનામાં પૂછે છે કે નકલને સૉપેલ કોઈપણ નામ પહેલેથી જ બીજી file દ્વારા ઉપયોગમાં લેવાય છે અને તે file આમ હશે. ઓવરરાઇટ પ્રોમ્પ્ટના જવાબમાં લેટર (લોઅર કેસ અથવા અપરકેસ) દાખલ કરવાથી command ચાલુ રહે છે; અન્ય કોઈપણ જવાબ command ને ફાઇલ પર ફરીથી લખતા અટકાવે છે. આમ, ઉદાહરણ તરીકે, જો તેને dir4 નામની ડિરેક્ટરીની નકલ બનાવવાની અને તેને dir5 કહેવાની ઈચ્છા હોય અને જો dir4 નામની ડિરેક્ટરી પહેલેથી અસ્તિત્વમાં હોય, તો પછીની ડિરેક્ટરીમાં સમાન નામો ધરાવતી કોઈપણ ફાઇલોને બદલતા agn નીચેના વપરાશકર્તાને પૂછશે:

cp -ri dir4 dir5

-a વિકલ્પ મૂળ ડિરેક્ટરી અને તેના સમાવિષ્ટોની શક્ય તેટલી રચના અને વિશેષતાઓને નવી ડિરેક્ટરીમાં સાચવે છે અને આમ આર્કાઇવ્સ બનાવવા માટે ઉપયોગી છે. તે r વિકલ્પ જેવું જ છે જેમાં તે ડિરેક્ટરીઓની પુનરાવર્તિત નકલ કરે છે; જો કે, તે ક્યારેય સાંકેતિક લિંકને અનુસરતું નથી. તે વિકલ્પોના rdp સંયોજનની સમકક્ષ છે.

સ્ટાર વાઇલ્ડકાર્ડનો ઉપયોગ કરીને ડિરેક્ટરીમાંની બધી ફાઇલોને બીજી ડિરેક્ટરીમાં કૉપી કરી શકાય છે. સ્ટાર કેરેક્ટર કોઈપણ એક પાત્ર અથવા અક્ષરોના કોઈપણ સંયોજનને રજૂ કરે છે. આમ, ઉદાહરણ તરીકે, ધનીચે પ્રમાણે dir6 નામની ડિરેક્ટરીમાંની તમામ ફાઇલોને dir7 નામની અન્ય હાલની ડિરેક્ટરીમાં કૉપી કરશે:

cp dir6/* dir7

cp ફાઇલો અને ડિરેક્ટરીઓની પસંદગીપૂર્વક કૉપી કરવા માટે સ્ટાર વાઇલ્ડકાર્ડ અથવા અન્ય પેટર્ન મેચિંગ અક્ષરો સાથે પણ વાપરી શકાય છે. ઉદાહરણ તરીકે, હાલની ડિરેક્ટરીમાં જે file નામ એક્સ્ટેન્શન .html ધરાવે છે તે તમામ ફાઇલોને dir8 નામની અન્ય હાલની ડિરેક્ટરીમાં કૉપી કરવા માટે, નીચેનાનો ઉપયોગ કરવામાં આવશે:

cp *.html dir8

આ કિસ્સામાં, સ્ટાર વાઇલ્ડકાર્ડ એવી કોઈપણ વસ્તુનું પ્રતિનિધિત્વ કરે છે જેનું નામ .html એક્સ્ટેન્શન સાથે સમાપ્ત થાય છે.

cp માટેના અન્ય વિકલ્પોમાં -b છે, જે દરેક ગંતવ્ય file ની બેકઅપ નકલો બનાવે છે, -f (એટલે કે, બળ), જે ડેસ્ટિનેશન ફાઇલોને દૂર કરે છે જે ખોલી શકાતી નથી અને ફરીથી પ્રયાસ કરે છે, -s, જે નકલ કરવાને બદલે સાંકેતિક લિંક્સ બનાવે છે, -u (એટલે કે, અપડેટ), જે માત્ર ત્યારે જ નકલ કરે છે જો સ્રોત file ગંતવ્ય file કરતાં નવી હોય અથવા જો ગંતવ્ય file ખૂટે છે, -v (એટલે કે, વર્બોઝ), જે શું થઈ રહ્યું છે તેના વિશે ટૂંકી ટિપ્પણી કરે છે, અને -x, જે cp ને સમાન file સિસ્ટમ પર રહેવા માટે કહે છે.

Move command : mv command નો ઉપયોગ ફાઇલો અને ડિરેક્ટરીઓનું નામ બદલવા અને ખસેડવા માટે થાય છે. તેનું સામાન્ય વાક્યરચના છે:

\$ mv source destination

આ દલીલો ફાઇલો અને ડિરેક્ટરીઓના નામ છે. જો બે file નામ દલીલો તરીકે પ્રદાન કરવામાં આવ્યા હોય, તો mv પ્રથમનું નામ બીજા તરીકે બદલી નાખે છે. જો દલીલોની સૂચિ પ્રદાન કરવામાં આવે છે અને અનુક્રમમાં અંતિમ દલીલ એ હાલની ડિરેક્ટરીનું નામ છે, તો mv અન્ય તમામ વસ્તુઓને તેમાં ખસેડે છે. તે ડિરેક્ટરી. જો અંતિમ દલીલ હાલની ડિરેક્ટરી નથી અને બે કરતાં વધુ દલીલો પૂરી પાડવામાં આવે છે, તો એક ભૂલ સંદેશ પરત કરવામાં આવે છે.

જો ગંતવ્ય file સ્ત્રોત file ની સમાન ડિરેક્ટરીમાં સ્થિત છે, તો સ્ત્રોત file નું નામ બદલી શકાય છે. જો બંને અલગ-અલગ ડિરેક્ટરીઓમાં હોય, તો સ્ત્રોત file ને ગંતવ્ય દલીલમાં નામ આપવામાં આવેલી ડિરેક્ટરીમાં ખસેડવામાં આવે છે, જેમાં તે તેનું મૂળ નામ રાખી શકે છે અથવા તેને નવું નામ સોંપવામાં આવે છે. જો લક્ષ્ય નિર્દેશિકા છે, તો સ્ત્રોત file અથવા ડિરેક્ટરી તે ડિરેક્ટરીમાં ખસેડવામાં આવે છે અને તેનું મૂળ નામ જાળવી રાખે છે. આમ, ઉદાહરણ તરીકે, નીચે આપેલ file1 નામની file નું નામ બદલીને file2 કરશે, જ્યારે તેને વર્તમાન ડિરેક્ટરીમાં રાખશે (એટલે કે, ડિરેક્ટરી જેમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે):

mv file1 file2

નીચેની file3 નામની file ને ખસેડશે, વગર તેનું નામ બદલીને, વર્તમાન ડિરેક્ટરીમાંથી dir1 નામની વર્તમાન ડિરેક્ટરીની હાલની સબડિરેક્ટરીમાં:

mv file3 dir1/file3

mv નો ઉપયોગ કોઈપણ સંખ્યામાં ફાઇલો અને ડિરેક્ટરીઓ એકસાથે ખસેડવા માટે કરી શકાય છે. ઉદાહરણ તરીકે, નીચેનો command બધી ફાઇલો અને ડિરેક્ટરીઓને ખસેડે છે, જેમાં તે ડિરેક્ટરીઓના તમામ સમાવિષ્ટોનો સમાવેશ થાય છે, વર્તમાન ડિરેક્ટરીમાંથી ડિરેક્ટરી /home/alice/new/ પર:

mv */home/alice/new/

ફૂદડી એ વાઈલ્ડકાર્ડ અક્ષર છે જે કોઈપણ શબ્દમાળા (એટલે કે, અક્ષરોનો ક્રમ) રજૂ કરે છે. આમ, ઉપરના ઉદાહરણમાં તે વર્તમાન ડિરેક્ટરીમાંની દરેક ફાઇલ અને ડિરેક્ટરીનું નામ દર્શાવે છે.

mv file અથવા ડિરેક્ટરીને ડાયરેક્ટરીના વંશવેલો ઉપર ખસેડવાનું સરળ બનાવે છે (એટલે કે, રૂટ ડિરેક્ટરીની નજીક) જેટલું નીચે. ઉદાહરણ તરીકે, નીચે આપેલ file4 નામની file ને ખસેડશે, જે હાલમાં વપરાશકર્તાની હોમ ડિરેક્ટરીની સબ-ડિરેક્ટરી dir/dir/ માં સ્થિત છે, વપરાશકર્તાની હોમ ડિરેક્ટરીમાં ટોચના સ્તર પર છે:

mv dir/dir/file4 ~

રૂટ ડિરેક્ટરી એ ડિરેક્ટરી છે જેમાં યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ પરની અન્ય તમામ ડિરેક્ટરીઓ શામેલ છે અને જે ડિરેક્ટરીઓના વંશવેલોની ટોચ પર છે. વપરાશકર્તાની હોમ ડિરેક્ટરી એ ડિરેક્ટરી છે જેમાં વપરાશકર્તા સિસ્ટમમાં લૉગિન થયા પછી ડિફોલ્ટ રૂપે પોતાને શોધે છે અને જે ટિલ્ડ (વેવી હોરીઝોન્ટલ લાઈન કેરેક્ટર) દ્વારા રજૂ કરી શકાય છે.

મૂળભૂત રીતે, mv કરતું નથી ડિસ્કલે સ્કીન પર કોઈપણ કન્ફર્મેશન પ્રદાન કરો જો તેની ક્રિયા સમસ્યા વિના પૂર્ણ થાય. આ યુનિક્સ ફિલસૂફીના મૌન સિદ્ધાંત સાથે સુસંગત છે.

આ રીતે યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમમાં નવા વપરાશકર્તાઓ માટે હંમેશાં i વિકલ્પનો ઉપયોગ કરવો તે શાણપણની વાત છે, જે એવી પરિસ્થિતિમાં mv ને ઇન્ટરેક્ટિવ બનાવે છે જેમાં ગંતવ્ય નિર્દેશિકામાં

સમાન નામની ફાઇલો અને/અથવા ડિરેક્ટરીઓ પહેલેથી જ અસ્તિત્વમાં છે. ઉદાહરણ તરીકે, ઉપરોક્ત command નીચે પ્રમાણે ઇન્ટરેક્ટિવ બનાવવામાં આવશે:

mv -i */home/alice/new/

એમવીના કેટલાક અન્ય વિકલ્પોમાં b છે, જે તેને દરેક file ની બેકઅપ કોપી બનાવવાનું કહે છે જે અન્યથા ઓવરરાઈટ અથવા દૂર કરવામાં આવશે, અને -v, જે તેને બેવરબોઝ કરવાનું કહે છે અને તેને ખસેડતા ago દરેક file નું નામ પ્રદર્શિત કરે છે. mv વિશે વિગતવાર માહિતી (બધા વિકલ્પો સહિત) તેના --help વિકલ્પનો ઉપયોગ કરીને મેળવી શકાય છે, અને વર્તમાન સંસ્કરણ વિશેની માહિતી તેના - સંસ્કરણ વિકલ્પનો ઉપયોગ કરીને મેળવી શકાય છે.

Remove or Delete

\$rmmdir

'rmdir' command કોઈપણ ખાલી ડિરેક્ટરીઓ દૂર કરે છે, પરંતુ જો તેમાં કોઈ file હાજર હોય તો ડિરેક્ટરીને કાઢી શકાતી નથી. 'rmdir' command નો ઉપયોગ કરવા માટે, તમારે ago બધી ફાઇલો દૂર કરવી પડશે જે તમે દૂર કરવા માંગો છો તે ડિરેક્ટરી હાજર છે (અને કદાચ ડિરેક્ટરીઓ જો કોઈ હોય તો).

Remove files and directories : rm (એટલે કે, દૂર કરો) command નો ઉપયોગ Linux અને અન્ય યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ પર ફાઇલો અને ડિરેક્ટરીઓ કાઢી નાખવા માટે થાય છે.

The general syntax for rm is :

rm [option] [-r directories] filenames

ચોરસ કૌંસમાંની વસ્તુઓ વૈકલ્પિક છે. જ્યારે ફક્ત એકના નામ સાથે વપરાય છે અથવા વધુ ફાઇલો, rm તે બધી ફાઇલોને વપરાશકર્તા દ્વારા પુષ્ટિની જરૂર વગર કાઢી નાખે છે. આમ, નીચેના ઉદાહરણમાં, rm તરત જ file1, file2 અને file3 નામની ફાઇલોને કાઢી નાખશે, એમ ધારીને કે ત્રણેય વર્તમાન ડિરેક્ટરીમાં સ્થિત છે (એટલે કે, ડિરેક્ટરી જેમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે):

rm file1 file2 file3

ભૂલ સંદેશો જો file અસ્તિત્વમાં ન હોય અથવા વપરાશકર્તાને તેને કાઢી નાખવાની યોગ્ય પરવાનગી ન હોય તો પરત કરવામાં આવે છે. સંરક્ષિત ફાઇલો લખો, દૂર કરતા ago વપરાશકર્તાને પુષ્ટિ માટે પ્રોમ્પ્ટ કરો (હા માટે y અને ના માટે n સાથે).

રાઈટ-પ્રોટેક્ટેડ ડિરેક્ટરીઓમાં આવેલી ફાઇલોને ક્યારેય દૂર કરી શકાતી નથી, ભલે તે ફાઇલો રાઈટ પ્રોટેક્ટેડ ન હોય. -f (એટલે કે, બળ) વિકલ્પ rm ને બધી સ્પષ્ટ કરેલી ફાઇલોને દૂર કરવા કહે છે, પછી ભલે તે લખે સુરક્ષિત હોય કે ન હોય, વપરાશકર્તાને પૂછ્યા વગર.

જો કોઈ ઉલ્લેખિત file અસ્તિત્વમાં ન હોય તો તે ભૂલ સંદેશો અથવા રીટર્ન એરર સ્ટેટસ દર્શાવતું નથી. જો કે, જો લખાણ-સંરક્ષિત ડિરેક્ટરીમાં ફાઇલોને દૂર કરવાનો પ્રયાસ કરવામાં આવે, તો આ વિકલ્પ નહીં મળે ભૂલ સંદેશને દબાવો.

-i (એટલે કે, ઇન્ટરેક્ટિવ) વિકલ્પ rm ને દરેક file અને ડિરેક્ટરીને દૂર કરતા ago પુષ્ટિ માટે વપરાશકર્તાને પૂછવા માટે કહે છે. જો -f અને -i બંને વિકલ્પો ઉલ્લેખિત હોય, તો છેલ્લો ઉલ્લેખિત વિકલ્પ અસર કરે છે.

સુરક્ષા માપદંડ તરીકે, rm એ ડિફોલ્ટ રૂપે ડિરેક્ટરીઓ કાઢી નાખતું નથી. ડિરેક્ટરીઓ કાઢી નાખવા માટે, -r વિકલ્પનો ઉપયોગ કરવો જરૂરી છે, જે -R વિકલ્પ જેવો જ છે. આ વિકલ્પ દલીલ સૂચિમાંથી ડિરેક્ટરીઓ અને

તેમના સમાવિષ્ટોને વારંવાર દૂર કરે છે; એટલે કે, ઉલ્લેખિત ડિરેક્ટરીઓ પ્રથમ કોઈપણ સબડિરેક્ટરીઝ (તેમની સબડિરેક્ટરીઝ અને ફાઈલો વગેરે સહિત) અને ફાઈલોમાંથી ખાલી કરવામાં આવશે અને પછી દૂર કરવામાં આવશે. વપરાશકર્તાને સામાન્ય રીતે ડિરેક્ટરીઓમાં કોઈપણ રાઈટ પ્રોટેક્ટેડ ફાઈલોને દૂર કરવા માટે પૂછવામાં આવે છે સિવાય કે -f વિકલ્પનો ઉપયોગ કરવામાં આવે.

જો rm દ્વારા મળેલી file સાંકેતિક લિંક હોય, તો લિંકને દૂર કરવામાં આવે છે, પરંતુ તે લિંકને સંદર્ભિત કરતી file અથવા ડિરેક્ટરીને અસર થશે નહીં. વપરાશકર્તાને સાંકેતિક લિંકને કાઢી નાખવા માટે લખવાની પરવાનગીની જરૂર નથી, જ્યાં સુધી વપરાશકર્તા પાસે તે લિંક રહેતી ડિરેક્ટરી માટે લખવાની પરવાનગી હોય.

rm command -- (સળંગ બે ડેશ) પરિમાણને સીમાંક તરીકે આધાર આપે છે જે વિકલ્પોનો અંત સૂચવે છે. જ્યારે file અથવા ડિરેક્ટરીનું નામ ડેશ અથવા હાઈફનથી શરૂ થાય ત્યારે આ ઉપયોગી છે. ઉદાહરણ તરીકે, નીચેના -dir1 નામની ડિરેક્ટરીને દૂર કરે છે:

rm -r --- -dir1

અન્ય વિકલ્પોમાં -v (એટલે કે વર્બોઝ), જે શું થઈ રહ્યું છે તે વિશે વધારાની માહિતી પ્રદાન કરે છે, --help, જે rm વિશે મૂળભૂત દસ્તાવેજીકરણ પ્રદાન કરે છે, અને --સંસ્કરણ, જે હાલમાં ઉપયોગમાં છે તે rm નું સંસ્કરણ જણાવે છે. rm ના વિવિધ સંસ્કરણો વચ્ચે કેટલાક તફાવતો અસ્તિત્વમાં છે, તેથી ચોક્કસ સિસ્ટમ માટે દસ્તાવેજીકરણ વાંચવું હંમેશા સમજદાર છે.

The rmdir command એ rm થી અલગ છે કે તે ફક્ત ઉપયોગમાં લેવાય છે ખાલી ડિરેક્ટરીઓ દૂર કરો

The rmdir command

Linux માં ખાલી ડિરેક્ટરીઓ દૂર કરવા માટે rmdir command નો ઉપયોગ થાય છે

rmdir માટે વાક્યરચના એ

rmdir [option] directory_names

જ્યારે કોઈપણ વિકલ્પો વિના ઉપયોગ થાય છે, ત્યારે rm એવી કોઈપણ ખાલી ડિરેક્ટરીઓ કાઢી નાખશે કે જેના નામ દલીલો (એટલે કે, ઈનપુટ્સ) તરીકે પૂરા પાડવામાં આવે છે, પછી ભલે તે આવી ડિરેક્ટરીઓને લખવાની પરવાનગી હોય કે ન હોય. આમ, ઉદાહરણ તરીકે, નીચેનો command dir1 અને dir2 નામની બે ખાલી ડિરેક્ટરીઓ દૂર કરશે જે વર્તમાન ડિરેક્ટરીમાં સ્થિત છે (એટલે કે, ડિરેક્ટરી જેમાં વપરાશકર્તા હાલમાં કામ કરી રહ્યો છે):

rmdir dir1 dir2

માત્ર ખાલી ડિરેક્ટરીઓ દૂર કરવાની ક્ષમતા છે. બિલ્ટ-ઇન સેફગાર્ડ જે ડેટાના આકસ્મિક નુકશાનને રોકવામાં મદદ કરે છે. આ અગત્યનું છે કારણ કે એકવાર કાઢી નાખ્યા પછી, યુનિક્સ જેવી ઓપરેટિંગ સિસ્ટમ્સ પર કાઢી નાખેલ ડેટા પુનઃપ્રાપ્ત કરવો અત્યંત મુશ્કેલ અથવા અશક્ય છે.

-p (એટલે કે, માતાપિતા) વિકલ્પ rmdir ને ઉલ્લેખિત ડિરેક્ટરીની પેરેન્ટ ડિરેક્ટરીઓ દૂર કરવા કહે છે જો દરેક ક્રમિક પેરેન્ટ ડિરેક્ટરી, બદલામાં, ખાલી થઈ જશે અને જો દરેક પિતૃ નિર્દેશિકાને લખવાની પરવાનગી છે. આમ, ઉદાહરણ તરીકે, જો dir5 ખાલી હોય તો નીચે આપેલ dir5, dir4 અને dir3 ને દૂર કરશે, dir4 માં માત્ર dir5 અને dir3 માં માત્ર dir4 (જે બદલામાં, dir5 ધરાવે છે):

rmdir -p dir3/dir4/dir5

આ mkdir command ના -p વિકલ્પ સાથે સમપ્રમાણતા પૂરી પાડે છે, જેનો ઉપયોગ ડિરેક્ટરીઓ બનાવવા માટે થાય છે. આમ, નેસ્ટેડ ડિરેક્ટરીઓનો ઉપરોક્ત સેટ નીચેની સાથે સરળતાથી બનાવી શકાય છે:

mkdir -p dir3/dir4/dir5

rm command થી વિપરીત, જેનો ઉપયોગ ફાઈલો અને ડિરેક્ટરીઓ બંનેને કાઢી નાખવા માટે થાય છે, ત્યાં rmdir માટે કોઈ -r વિકલ્પ નથી. ઓછામાં ઓછા GNU સંસ્કરણ પર કે જે Linux પર પ્રમાણભૂત છે. તે વિકલ્પ rm ને તેના તમામ સમાવિષ્ટો કાઢી નાખીને પુનરાવર્તિત રીતે ડાયરેક્ટરી કાઢી નાખવા માટે પરવાનગી આપે છે, જે સબડિરેક્ટરીઝના સૌથી નીચા સ્તરોથી શરૂ થાય છે. આમ, જો વપરાશકર્તા સંપૂર્ણ ડિરેક્ટરી સ્ટ્રક્ચરને દૂર કરવા માંગે છે, તો સામાન્ય રીતે દરેક ડિરેક્ટરી, તેની સબડિરેક્ટરીઝ વગેરેની સામગ્રીને agoo દૂર કરવાનો પ્રયાસ કરવાને બદલે તેના -r વિકલ્પ સાથે rm નો ઉપયોગ કરવો સૌથી વધુ કાર્યક્ષમ છે.

ત્રણ વિકલ્પો કે જે rmdir rm સાથે શેર કરે છે તે છે -v (એટલે કે વર્બોઝ), જે શું થઈ રહ્યું છે તે વિશે વધારાની માહિતી પૂરી પાડે છે, --help, જે rmdir વિશે મૂળભૂત દસ્તાવેજીકરણ પૂરા પાડે છે, અને આવૃત્તિ, જે હાલમાં ઉપયોગમાં છે તે rmdir ની આવૃત્તિને કહે છે. rmdir ની વિવિધ આવૃત્તિઓ વચ્ચે કેટલાક તફાવતો અસ્તિત્વમાં છે, તેથી ચોક્કસ સિસ્ટમ માટે દસ્તાવેજીકરણ વાંચવું હંમેશા સમજદાર છે.

Listing and combining files with the cat command

બિલાડી (કોનકેટેનેટ file) command નો ઉપયોગ તમારી સ્ક્રીન પર ફાઈલોની સામગ્રી મોકલવા માટે થાય છે. આ

command નો ઉપયોગ ફાઈલોની સામગ્રીને અન્ય ફાઈલોમાં મોકલવા માટે પણ થઈ શકે છે. અવર 6 સ્ટાન્ડર્ડ ઈનપુટ, સ્ટાન્ડર્ડ આઉટપુટ અને રીડાયરેક્શન જેવા શબ્દોને આવરી લે છે અને આ વિભાગ તમને આ command

માટે કેટલાક મૂળભૂત ઉપયોગો બતાવે છે. જોકે બિલાડી ટૂંકી ફાઈલો વાંચવા માટે ઉપયોગી હોઈ શકે છે, તેનો ઉપયોગ સામાન્ય રીતે ફાઈલોને જોડવા, બનાવવા, ફરીથી લખવા અથવા જોડવા માટે થાય છે. ટૂંકી file જોવા માટે બિલાડીનો ઉપયોગ કરવા માટે, તમે

\$ cat test.txt

દાખલ કરી શકો છો આ ટેક્સ્ટ file cat command

દ્વારા બનાવવામાં આવી હતી. બિલાડી હોઈ શકે છે વિશ્વનો સૌથી સરળ ટેક્સ્ટ એડિટર.

જો તમે આ પુસ્તક વાંચશો, તો તમે બિલાડીનો ઉપયોગ કેવી રીતે કરવો તે શીખી શકશો.

આ fileમાં લખાણની આ છેલ્લી લાઈન છે. કેટ કમાન્ડમાં પણ ઘણા બધા વિકલ્પો છે.

જો તમે તમારી fileને લીટી નંબરો સાથે જોવા માંગતા હો, તો કદાચ ચોક્કસ શબ્દસમૂહની નોંધ લેવા માટે, તમે -n વિકલ્પનો ઉપયોગ કરી શકો છો:

\$ cat -n test.txt

- આ ટેક્સ્ટ file cat command દ્વારા બનાવવામાં આવી હતી.
- બિલાડી વિશ્વની સૌથી સરળ ટેક્સ્ટ એડિટર હોઈ શકે છે.
- જો તમે આ પુસ્તક વાંચશો, તો તમે બિલાડીનો ઉપયોગ કેવી રીતે કરવો તે શીખી શકશો.
- આ આ ફાઈલમાં લખાણની છેલ્લી લીટી છે. અને એક સાથે અનેક ફાઈલો જોવા માટે પણ બિલાડીનો ઉપયોગ કરો, કારણ કે બિલાડી વાઈલ્ડકાર્ડ સ્વીકારે છે, ઉદાહરણ તરીકે:

\$ cat -n test*

- આ ટેક્સ્ટ file cat command દ્વારા બનાવવામાં આવી હતી.
- બિલાડી વિશ્વની સૌથી સરળ ટેક્સ્ટ એડિટર હોઈ શકે છે.
- જો તમે આ પુસ્તક વાંચશો, તો તમે બિલાડીનો ઉપયોગ કેવી રીતે કરવો તે શીખી શકશો.
- આ આ ફાઈલમાં લખાણની છેલ્લી લીટી છે.
- આ test2.txt ની પ્રથમ લાઈન છે.
- આ file પણ બિલાડી દ્વારા બનાવવામાં આવી હતી.
- આ test2.txt ની છેલ્લી લાઈન છે.

જેમ તમે જોઈ શકો છો, બિલાડીએ તેના આઉટપુટમાં બીજી fileનો પણ સમાવેશ કર્યો છે, અને આઉટપુટની દરેક લાઈનને નંબર આપ્યો છે, દરેક fileને નહીં. નોંધ કરો કે

\$ cat test.txt test2.txt

સાથેની બંને ફાઈલો પણ જુઓ આઉટપુટ બરાબર એ જ હશે જેમ કે વાઈલ્ડકાર્ડનો ઉપયોગ કર્યો હોય. પરંતુ ઘણી ફાઈલો જોવી એ બિલાડીનો ઉપયોગ કરવાનો એક જ રસ્તો છે. તમે ફાઈલોને જોડવા માટે પુનઃદિશામાન ઓપરેટર સાથે cat command નો પણ ઉપયોગ કરી શકો છો. ઉદાહરણ તરીકે, જો તમે test.txt અને test2.txt ને test3.txt નામની ત્રીજી fileમાં જોડવા માંગતા હો, તો તમે

\$ cat test* > test3.txt

નો ઉપયોગ કરી શકો છો

ls -l test*

વડે પરિણામ તપાસો. આ કિસ્સામાં, વપરાશકર્તા ago નક્કી કરો કે જુ તમે test.txt ની સામગ્રી test2.txt માં જવા માંગો છો અથવા test2.txt ની સામગ્રી test.txt માં જવા માંગો છો. પછી, >> રીડાયરેક્શન ઓપરેટર સાથે બિલાડીનો ઉપયોગ કરીને, તમે

\$ cat test.txt >> test2.txt લખી શકો છો.

આ test2.txt ના અંતમાં test.txt ની સામગ્રીને જોડે છે. પરિણામો તપાસવા માટે, ફરીથી બિલાડીનો ઉપયોગ કરો:

\$ cat test2.txt

આ test2.txt ની પ્રથમ લાઈન છે. આ file પણ બિલાડી દ્વારા બનાવવામાં આવી હતી. આ test2.txt ની છેલ્લી લાઈન છે.

આ ટેક્સ્ટ file cat command દ્વારા બનાવવામાં આવી હતી.

બિલાડી વિશ્વની સૌથી સરળ ટેક્સ્ટ એડિટર હોઈ શકે છે.

જો તમે આ પુસ્તક વાંચશો, તો તમે બિલાડીનો ઉપયોગ કેવી રીતે કરવો તે શીખી શકશો.

આ fileમાં લખાણની આ છેલ્લી લાઈન છે. નોંધ કરો કે જો તમે command દાખલ કર્યો હોય.

\$ cat -n test.txt && test2.txt

test2.txt file

\$ cat test2.txt જેવી દેખાશે

આ test2.txt ની પ્રથમ લાઈન છે.

આ file પણ બિલાડી દ્વારા બનાવવામાં આવી હતી.

આ test2.txt ની છેલ્લી લાઈન છે.

- આ ટેક્સ્ટ file cat command દ્વારા બનાવવામાં આવી હતી.
- બિલાડી વિશ્વની સૌથી સરળ ટેક્સ્ટ એડિટર હોઈ શકે છે.
- જો તમે આ પુસ્તક વાંચશો, તો તમે બિલાડીનો ઉપયોગ કેવી રીતે કરવો તે શીખી શકશો.
- આ આ ફાઈલમાં લખાણની છેલ્લી લીટી છે.

અંતે, જો તમે વર્ડ પ્રોસેસર અથવા ટેક્સ્ટ એડિટર ચલાવ્યા વિના ટૂંકી ટેક્સ્ટ file બનાવવા માંગતા હોવ તો તમે ઉપયોગ કરી શકો છો તે અહીં એક યુક્તિ છે. કારણ કે cat command પ્રમાણભૂત ઈનપુટ વાંચી શકે છે (આ વિશે વધુ કલાક 6 માં), તમે cat command ને file બનાવવા અને તેને તમારા કીસ્ટ્રોકથી ભરી શકો છો.

અહીં કેવી રીતે છે:

\$ cat && myfile.txt

હવે, અમુક ટેક્સ્ટ દાખલ કરો:

\$ cat > myfile.txt

આ કેટ વર્ડ પ્રોસેસર છે.

આ ફાઈલનો અંત છે.

પછી, જ્યારે તમે ટાઈપિંગ પૂર્ણ કરી લો, ત્યારે fileને બંધ કરવા માટે Ctrl+D દબાવો. આ કામ કરે છે કે કેમ તે જોવા માટે,

\$ ls -l myfile.txt -

rw-rw-r-- 1 bball bball 61 નવે 12 18:26 myfile.txt અજમાવી જુઓ

\$ cat myfile.txt

આ કેટ વર્ડ પ્રોસેસર છે.

આ ફાઈલનો અંત છે. વપરાશકર્તાએ એ પણ જાણવું જોઈએ કે cat કમાન્ડ કોઈપણ fileની સામગ્રીને છાપશે, અને માત્ર ટેક્સ્ટ ફાઈલો નહીં. જો કે બિલાડી એક અથવા ઘણી ટૂંકી ફાઈલો જોવા માટે ઉપયોગી હોઈ શકે છે,

date command ના ઉદાહરણો પ્રદર્શિત કરવા અને સિસ્ટમ date સમય સેટ કરવા માટે

date command ઘણાબધા ફોર્મેટમાં date દર્શાવવા માટે મદદરૂપ છે.

Date command examples to display and set system date time

date સાથે command નો ઉપયોગ કેવી રીતે કરવો તેના થોડા ઉદાહરણો છે. જ્યારે કોઈપણ વિકલ્પ વિના date command ચલાવો, ત્યારે તે નીચે બતાવ્યા પ્રમાણે વર્તમાન date અને સમય દર્શાવશે.

\$ date

mon મે 20 22:02:24 PDT 2013

1 Display Date from a String Value using -date Option

જો તમારી પાસે શબ્દમાળામાં સ્થિર date અથવા સમય મૂલ્ય હોય, તમે ઈનપુટ સ્ટ્રિંગને નીચે બતાવ્યા પ્રમાણે ડેટ ફોર્મેટમાં કન્વર્ટ કરવા માટે -d અથવા -date વિકલ્પનો ઉપયોગ કરી શકો છો.

કૃપા કરીને નોંધો કે આ વર્તમાન date અને સમય મૂલ્યનો ઉપયોગ કરતું નથી. તેના બદલે તમે શબ્દમાળા તરીકે પસાર કરો છો તે date અને સમય મૂલ્યનો ઉપયોગ કરે છે.

નીચેના ઉદાહરણો ઇનપુટ date માત્ર શબ્દમાળા લે છે, અને date ફોર્મેટમાં આઉટપુટ દર્શાવે છે. જો તમે સમયનો ઉલ્લેખ કરતા નથી, તો તે સમય માટે 00:00:00 નો ઉપયોગ કરે છે.

```
$ date --date="12/2/2014"
```

```
મંગળ 2 Dec00:00:00 PST 2014
```

```
$ date --date="2 Feb2014"
```

```
Sun Feb2 00:00:00 PST 2014
```

```
$ date --date="Feb2 2014"
```

```
Sun Feb2 00:00:00 PST 2014
```

નીચેનું ઉદાહરણ ઇનપુટ date અને સમય શબ્દમાળા લે છે, અને date ફોર્મેટમાં આઉટપુટ દર્શાવે છે.

```
$ date --date="Feb2 2014 13:12:10"
```

```
Sun Feb2 13:12:10 PST 2014
```

2 Read Date Patterns from a file using -file option

આ -d અથવા -date વિકલ્પ જેવું જ છે જેની આપણે ઉપર ચર્ચા કરી છે. પરંતુ, તમે બહુવિધ date શબ્દમાળાઓ માટે તે કરી શકો છો. જો તમારી પાસે એવી file છે જેમાં વિવિધ સ્ટેટિક ડેટ સ્ટ્રીંગ્સ છે, તો તમે નીચે બતાવ્યા પ્રમાણે -f અથવા -file વિકલ્પનો ઉપયોગ કરી શકો છો.

આ ઉદાહરણમાં, આપણે જોઈ શકીએ છીએ કે ડેટફાઈલમાં 2 dateના શબ્દમાળાઓ છે. ડેટફાઈલની દરેક લાઈનને date command દ્વારા વિશ્લેષિત કરવામાં આવે છે અને દરેક લાઈન માટે date આઉટપુટ કરવામાં આવે છે.

```
$ cat datefile
```

```
9 સપ્ટેમ્બર 1986
```

```
23 ઓગસ્ટ 1987
```

```
$ date --file=datefile
```

```
મંગળવાર 9 સપ્ટેમ્બર 00:00:00 PDT 1986
```

```
Sun 23 ઓગસ્ટ 00:00:00 PDT 1987
```

3 Get Relative Date Using -date option

તમે સંબંધિત મૂલ્યોનો ઉપયોગ કરીને ભાવિ date મેળવવા માટે date command નો ઉપયોગ પણ કરી શકો છો.

ઉદાહરણ તરીકે, નીચેના ઉદાહરણો date મેળવે છે monવાર.

```
$ date -- date = "next mon"
```

```
mon મે 27 00:00:00 PDT 2013
```

જો શબ્દમાળા=@ date command ને આપવામાં આવે છે, તો date command યુગ (1970-01-01 UTC) થી સેકન્ડને a માં રૂપાંતરિત કરો- date

તે date દર્શાવે છે જેમાં 1970-01-01 UTC ના યુગથી 5 seconds વીતી ગયા છે:

```
$ date -- date=@5
```

```
બુધ Dec31 16:00:05 PST 1969
```

તે date દર્શાવે છે કે જેમાં 1970-01-01 UTC યુગથી 10 seconds વીતી ગઈ છે:

```
$ date -- date=@10
```

```
બુધ Dec31 16:00:10 PST 1969
```

તે date દર્શાવે છે જેમાં 1 મિનિટ (એટલે કે 60 સેકન્ડ) યુગ 1970-01-01 UTC થી વીતી ગયો છે:

```
$ date -- date=@60
```

```
બુધ Dec31 16:01:00 PST 1969
```

4 Display past date

તમે -date command નો ઉપયોગ કરીને ભૂતકાળની date દર્શાવી શકો છો. થોડી શક્યતાઓ નીચે દર્શાવેલ છે.

```
$ date --date='3 seconds ago'
```

```
mon મે 20 21:59:20 PDT 2013
```

```
$ date --date="1 દિવસ ago"
```

```
Sun મે 19 21:59:36 PDT 2013
```

```
$ date --date="yesterday"
```

```
Sun મે 19 22:00:26 PDT 2013
```

```
$ date --date="1 month ago"
```

```
શનિ એપ્રિલ 20 21:59:58 PDT 2013
```

```
$ date --date="1 year ago"
```

```
Sun મે 20 22:00:09 PDT 2012
```

5 Set Date and Time using -set option

તમે નીચે બતાવ્યા પ્રમાણે -s અથવા -set વિકલ્પનો ઉપયોગ કરીને તમારી સિસ્ટમની date અને સમય સેટ કરી શકો છો..

આ ઉદાહરણમાં, શરૂઆતમાં તે 20:09:31 તરીકે સમય દર્શાવે છે.

અમે પછી તેને 21:00:00 માં બદલવા માટે date command નો ઉપયોગ કર્યો.

```
$ date
```

```
Sun મે 20 20:09:31 PDT 2013
```

```
$ date -s "Sun may 20 21:00:00 PDT 2013"
```

```
Sun મે 20 21:00:00 PDT 2013
```

```
$ date
```

```
Sun મે 20 21:00:05 PDT 2013
```

6 Display Universal Time using -u option

તમે નીચે બતાવ્યા પ્રમાણે -u, અથવા -utc, અથવા -યુનિવર્સલ વિકલ્પનો ઉપયોગ કરીને UTC ફોર્મેટમાં date પ્રદર્શિત કરી શકો છો.

```
$ date
```

```
mon મે 20 22:07:53 PDT 2013
```


\$ date -u

મંગળ 21 મે 05:07:55 UTC 2013

7 Display Last Modification Time using -r option

આ ઉદાહરણમાં, વર્તમાન સમય 20:25:48 છે

\$ date

Sun મે 20 20:25:48 PDT 2013

touch કમાન્ડનો ઉપયોગ કરીને ડેટાફાઇલનો ટાઇમસ્ટેમ્પ બદલવામાં આવે છે. આ ઉપરોક્ત date command ના આઉટપુટ પછી થોડી સેકન્ડ પછી કરવામાં આવ્યું હતું.

\$ touch datefile

આ પછીનો વર્તમાન સમય ઉપર touch command છે 20:26:12

\$ date

Sun મે 20 20:26:12 PDT 2013

છેલ્લે, નીચે બતાવ્યા પ્રમાણે fileની છેલ્લી સંશોધિત ટાઇમસ્ટેમ્પ પ્રદર્શિત કરવા માટે date command -r વિકલ્પનો ઉપયોગ કરો. આ ઉદાહરણમાં, તે ડેટાફાઇલનો છેલ્લો સંશોધિત સમય 20:25:57 તરીકે દર્શાવે છે. તે ક્યાંક 20:25:48 અને 20:26:12 ની વચ્ચે છે (જે ત્યારે થાય છે જ્યારે આપણે ટાઇમસ્ટેમ્પને સંશોધિત કરવા માટે ઉપરોક્ત touch command ચલાવીએ છીએ).

\$ date -r datefile

Sun મે 20 20:25:57 PDT 2013

8 Various Date command formats

તમે નીચેના વાક્યચરણનાનો ઉપયોગ કરીને date command ને વિવિધ ફોર્મેટમાં દર્શાવવા માટે ફોર્મેટિંગ વિકલ્પનો ઉપયોગ કરી શકો છો:

\$ date +%<format-option>

© NIMI
NOT TO BE REPUBLISHED

એમએસ વર્ડ 2010 થીયરી (MS WORD 2010 THEORY)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- MS Office શું છે તે જણાવો
- સંક્ષિપ્તમાં MS વર્ડ શું છે અને શરૂઆતના પગલાં
- એમએસ વર્ડના વિવિધ સ્કીન બ્લોકસ સમજાવો
- દસ્તાવેજ બનાવવા, સાચવવા, પ્રિન્ટ કરવાની પ્રક્રિયા સમજાવો.

માઈક્રોસોફ્ટ ઓફિસ એ એપ્લિકેશન સોફ્ટવેર પેકેજ છે. માઈક્રોસોફ્ટ કોર્પોરેશન દ્વારા રજૂ કરવામાં આવ્યું છે. એમએસ ઓફિસ સમાવે છે નીચેના લોકપ્રિય પેકેજોમાંથી:

- માઈક્રોસોફ્ટ વર્ડ
- માઈક્રોસોફ્ટ એક્સેલ
- માઈક્રોસોફ્ટ પાવરપોઈન્ટ
- માઈક્રોસોફ્ટ એક્સેસ
- Microsoft Outlook

Word 2010

તે એક વર્ડ પ્રોસેસર પેકેજ છે જે બનાવવામાં મદદ કરે છે અને દસ્તાવેજ સંપાદિત કરો. તે સૌથી જાણીતું વર્ડ પ્રોસેસર છે આજકાલ તે દ્વારા પ્રોફેશનલ દેખાતા દસ્તાવેજો બનાવે છે બનાવવા માટે સાધનોનો વ્યાપક સમૂહ પ્રદાન કરે છે અને દસ્તાવેજ, મેમો, પત્રો, અહેવાલો, બ્રોશર, બિઝનેસ દસ્તાવેજો અને ઈન્ટરનેટ વેબપેજ પણ.

Starting word

Start > (windows Logo) All programs > Microsoft office > Microsoft Word પર ક્લિક કરો. MS Word ડિફોલ્ટ સેટિંગ્સ પૃષ્ઠ લેઆઉટ સાથે ડિફોલ્ટ ખાલી દસ્તાવેજ સાથે ખુલે છે.

Default page layout (Fig 1)



અગાઉના સંસ્કરણથી વિપરીત, MS Office 2010 માં એક સામાન્ય છે બધા ઘટકો માટે સુવિધાઓનો સમૂહ. તે તેને સરળ બનાવે છે સમજવા માટે અને સામાન્ય ઉપયોગિતાઓ ઉપલબ્ધ રહેવા માટે વર્ડ, એક્સેલ સહિત એમએસ ઓફિસના તમામ પેકેજો પર પાવરપોઈન્ટ, એક્સેસ, વગેરે ઘણી સુવિધાઓ ફરીથી ડિઝાઇન કરવામાં આવી છે જેથી અન્ય પેકેજ્ઝ કેનનાં ડાયવર્ટ કરાયેલા વપરાશકર્તાઓને સક્ષમ કરી શકાય સાથે પરવડી શકે

છે. વર્ડમાં આ ઉન્નતિકરણો અને ઉપયોગિતાઓ 2010 નીચે પ્રમાણે વર્ણવેલ છે.

વર્ડની રિબન ટેબમાં આઠ મુખ્ય ભાગો છે જેમ કે. ફાઇલ, હોમ, ઇન્સર્ટ, પેજ લેઆઉટ, રેફરન્સ, મેઈલિંગ્સ, સમીક્ષા કરો અને જુઓ.

નવી ફાઇલ બનાવવા માટે રિબનની ફાઇલ ટેબ ઉપયોગી છે, એ ખાલી અથવા ટેમ્પલેટ પૃષ્ઠ જરૂરી છે. Ctrl + N હંમેશા ઓફિસ પેકેજમાં નવો ખાલી દસ્તાવેજ બનાવો. આ આ રીતે બનાવેલ દસ્તાવેજો મૂળભૂત શબ્દ તરીકે સાચવી શકાય છે દસ્તાવેજ અથવા આપેલ સુસંગત ફોર્મેટ. વધુમાં, દસ્તાવેજો એક્સેસ કરવાની પરવાનગી નિશ્ચિત કરી શકાય છે, નેટવર્ક પર દસ્તાવેજ શેર કરો, એનું સંચાલન પણ કરો દસ્તાવેજ અગાઉના અને બાહ્ય સાથે સુસંગત છે આવૃત્તિઓ. પ્રિન્ટ વિકલ્પ દસ્તાવેજને સખત બનાવે છે ઈન્સ્ટોલ કરેલ પ્રિન્ટર અનુસાર નકલ અથવા પીડીએફ ફોર્મેટ વિશેષતા. તાજેતરમાં ખોલેલી ફાઇલો પણ જોઈ શકાય છે ફરીથી કામ કરવું સરળ શોધો. વર્ડ ઓન હેલ્પ એ જ ટેબમાં છે.

Home Tab (Fig 2)



હોમ ટેબમાં કટ, કોપી, પેસ્ટ સહિતના ક્લિપબોર્ડ વિકલ્પો છે અને ખાસ પેસ્ટ કરો. ફોન્ટ બ્લોકમાં તમામ વિકલ્પો છે ફોન્ટ્સ, કદ, સુપરસ્ક્રિપ્ટ, સબસ્ક્રિપ્ટ, બોલ્ડ, ઇટાલિક, સેટિંગ અન્ડરલાઈન, સ્ટ્રાઈકથ્રુ, ફોન્ટ રંગો, વગેરે.

Paragraph Group (Fig 3)



ફકરા જૂથ સાથે ફોર્મેટિંગ ફકરા બનાવે છે સંરેખણ ડાબે, જમણે, મધ્યમાં, ન્યાયી અને ઈન્ડેન્ટેશન, પેરા અને રેખા અંતર. રેખા અને પૃષ્ઠ વિરામ બનાવવા માટે પરવાનગી આપે છે પૃષ્ઠ ક્રમાંકન વિકલ્પો અને ફોર્મેટિંગના અપવાદો.

Styles and Editing Group (Fig 4)

Fig 4



શૈલીઓ જૂથ પૂર્વ ફોર્મેટ કરેલ ટેક્સ્ટ શૈલીઓને મંજૂરી આપે છે જેમ કે મથાળાની શૈલીઓ, ફકરા શૈલીઓ, ઉપશીર્ષકો, વગેરે. કસ્ટમ શૈલીઓ પણ બદલાયેલ વિશિષ્ટતાઓ સાથે સંગ્રહિત કરી શકાય છે. આ સંપાદન જૂથનો ઉપયોગ ચોક્કસ વિસ્તાર પસંદ કરવા, શોધવા અને/અથવા માટે થાય છે ટેક્સ્ટમાં ઉલ્લેખિત ક્ષેત્રમાં વિકલ્પ બદલો.

ઈન્સર્ટ ટેબમાં 7 જૂથો છે. ફિગ 5 - પૃષ્ઠો, કોષ્ટકો અને ચિત્ર બ્લોક્સ (ફિગ 5)

Fig 5



પૃષ્ઠ જૂથ કવર પૃષ્ઠો, ખાલી પૃષ્ઠો દાખલ કરે છે અને પેજ બ્રેક્સ. કોષ્ટકોનું જૂથ a માં કોષ્ટક દાખલ કરવામાં મદદ કરે છે તૈયાર ઉલ્લેખિત પંક્તિ-કોલમ સેટ સાથેનો ટેક્સ્ટ દસ્તાવેજ અથવા એ કસ્ટમાઈઝ ટેબલ ફોર્મેટ. ચિત્ર જૂથ ચિત્રો દાખલ કરે છે બાહ્ય સ્ત્રોતો, ક્લિપર્ટ્સ, આકાર, સ્માર્ટ આર્ટ, ડેટાના ચાર્ટ્સ, ટેક્સ્ટ ડોક્યુમેન્ટમાં સ્ક્રીનશોટ પણ.

Links and Header/Footer Groups (Fig 6)

Fig 6



લિંક્સ જૂથ દસ્તાવેજમાં ટેક્સ્ટ, બુકમાર્ક અને ક્લોસ સંદર્ભો પર હાઈપરલિંક બનાવે છે.

હેડર અને ફૂટર ગ્રૂપ હેડર, ફૂટર ઈન્સર્ટ કરે છે પ્લેસમેન્ટ એરિયામાં પ્રદર્શિત કરવા માટે દરેક પૃષ્ઠ અને પૃષ્ઠ નંબરો પર દેખાય છે.

Text and Symbols block (Fig 7)

Fig 7



ટેક્સ્ટ ગ્રૂપ ટેક્સ્ટ બોક્સ બનાવવાની મંજૂરી આપે છે, સંભવતઃ ઝડપી ભાગો બ્રોશરો બનાવો, વર્કઆર્ટ તરીકે ડિઝાઇન કરેલ ટેક્સ્ટ, ફકરા શૈલીઓ, સહી રેખા, તારીખ અને ઓબ્જેક્ટ દાખલ કરવાનો વિકલ્પ.

પ્રતીકોનું જૂથ ગણિત જેવા સમીકરણોના પ્રતીકો દાખલ કરે છે ચલણ, ગણિતના પ્રતીકો, વગેરે જેવા સમીકરણો અથવા પ્રતીકો.

Themes and Page Setup Group (Fig 8)

Fig 8



Page Layout ટેબમાં પાંચ મુખ્ય જૂથો છે. થીમ્સ બ્લોક થીમ્સનો ઉપયોગ કરીને પૂર્વવ્યાખ્યાયિત ટેમ્પલેટ સેટઅપ બનાવે છે દસ્તાવેજો. કસ્ટમાઈઝ કરેલી નવી થીમ પણ ભવિષ્યના ઉપયોગ માટે બનાવી અને સાચવી શકાય છે.

Page setup જૂથમાં માર્જિન આસપાસ, પૃષ્ઠ પર સુવિધાઓ છે ઓરિએન્ટેશન એટલે કે ઊભી અથવા આડી, કાગળનું કદ, કોલમ પ્રદર્શિત કરવા, વિરામ, રેખા નંબરો અને હાઈફ્રેનેશન.

Page Background and Paragraph Group (Fig 9)

Fig 9



પૃષ્ઠ પૃષ્ઠભૂમિ જૂથ વોટરમાર્ક, પૃષ્ઠ બનાવે છે પૃષ્ઠભૂમિ રંગ અને પૃષ્ઠ સરહદો. ફકરો બ્લોક હોમ ->ફકરો જૂથમાં પહેલેથી જ ચર્ચા કરેલ વિકલ્પોનો ઉલ્લેખ કર્યો છે.

Arrange Group (Fig 10)

Fig 10



ગોઠવણી જૂથ વસ્તુઓ, ટેક્સ્ટની સ્થિતિ બનાવે છે એલિમેન્ટ્સ, ટેક્સ્ટ રેપિંગ, ઓબ્જેક્ટ ગોઠવણી, સંરેખિત કરવું વસ્તુઓ, જૂથ અને પરિવર્તન. સંદર્ભ ટેબમાં છ બ્લોક્સ છે.

TOC and Footnotes Group (Fig 11)

Fig 11



વિષયવસ્તુનું કોષ્ટક ચોક્કસ માટે TOC બનાવે છે પ્રકાશન દસ્તાવેજ આપોઆપ અને પર સુધારાઓ અપડેટ કરવા યોગ્ય. ફૂટનોટ્સ ટેબ ફૂટનોટ્સ બનાવે છે પૃષ્ઠનો, બ્લોક વિસ્તાર જેમાં સ્પષ્ટીકરણાત્મક શબ્દસમૂહો શામેલ છે અથવા સંદર્ભો. એન્ડનોટ દસ્તાવેજના અંતે નોંધ બનાવે છે.

Citations, Bibilography and Captions Group (Fig 12)



સંદર્ભ અને ગ્રંથસૂચિ બાહ્ય સ્ત્રોત તરીકે બનાવે છે સંદર્ભ, સંદર્ભ કોષ્ટકો અને સંબંધિત લેખકોની ક્રેડિટ પ્રકાશન માટે. કેપ્શન્સ ટેબ કેપ્શન, કોષ્ટક દાખલ કરે છે આંકડાઓની, ટેકસ્ટની અંદરના કોસ સંદર્ભો.

Index and Table of Authorities Group (Fig 13)



ઈન્ડેક્સ મુખ્ય વિષય અથવા પેટા વિષય પર મૂલ્ય દાખલ કરે છે અને તે મુજબ ઈન્ડેક્સ બનાવો. સત્તાવાળાઓનું ટેબલ છે અવતરણ ચિહ્નમાં કેસો, કાયદાઓની સૂચિ શામેલ છે અને અન્ય સત્તાવાળાઓ દસ્તાવેજમાં ટાંકવામાં આવ્યા છે.

Create and Mailmerge Groups (Fig 14)



મેઈલીંગ ટેબનો ઉપયોગ પત્ર અથવા ઓર્ડરની મેઈલીંગ બનાવવા માટે થાય છે સામાન્ય મેઈલીંગ સુવિધાઓનો ઉપયોગ કરીને. બ્લોક બનાવે છે બનાવો પરબિડીયાઓમાં પૂર્વવ્યાખ્યાયિત ફોર્મેટનો ઉપયોગ કરીને તેમજ પ્રિન્ટ કરે છે કસ્ટમ ફોર્મેટ્સ. લેબલ્સ પણ પ્રિન્ટ કરી શકાય છે જેના પર વિતરણ સરનામું ઉલ્લેખિત છે.

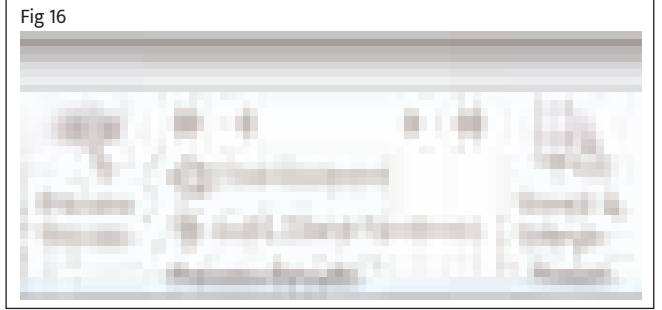
મેઈલ મર્જ બ્લોકનો ઉપયોગ પત્રના મેઈલ મર્જ બનાવવા માટે થાય છે, ઇમેઈલ જો સામાન્ય પત્ર બનાવવામાં આવે અને તેને મોકલવામાં આવે ઘણા પ્રાપ્તકર્તાઓ, આ વિકલ્પ સરનામું બનાવવા માટે પરવાનગી આપે છે બ્લોક કરો જ્યાં પ્રાપ્તકર્તાની માહિતી દેખાવી જોઈએ. સમ પસંદગીના પ્રાપ્તકર્તાઓને તેમજ નવા તરીકે સૂચિબદ્ધ કરી શકાય છે પ્રાપ્તકર્તા યાદી બનાવી શકાય છે.

Write and Insert fields Group (Fig 15)



પત્ર લખતી વખતે, તેના ઘણા ભાગો હોય છે, જે આ મેઈલ કરે છે મર્જ એડ્રેસ બ્લોક, ગ્રીટીંગ લાઈનના ફીલ્ડ્સ બનાવે છે, સામગ્રી, લેબલ્સ વગેરેનું ક્ષેત્ર મર્જ કરો.

Preview Results and Finish Groups (Fig 16)



મેઈલ મર્જના સફળ સમાપ્તિ પર, તે હોઈ શકે છે સમીક્ષા અને દરેક પ્રાપ્તકર્તાઓ માટે ભૂલો માટે ચકાસાયેલ અને તે મુજબ ફેરફાર કરો. મેઈલ મર્જને માન્ય કર્યા પછી પૂર્ણ થાય છે.

Proofing and Language Groups (Fig 17)



અહીં સમીક્ષા ટેબનો ઉપયોગ વિવિધ દસ્તાવેજ વિસ્તારો માટે થાય છે સમીક્ષા કરવાની છે. પ્રૂફિંગ જોડણી તપાસવાનું સરળ બનાવે છે અને દસ્તાવેજનું વ્યાકરણ. આકસ્મિક ભૂલો થઈ શકે છે અહીં સુધારેલ. સંશોધન માત્ર શોધને શુદ્ધ કરે છે દસ્તાવેજની અંદર, પણ સંદર્ભિત દસ્તાવેજો. થિસોરસ દસ્તાવેજની સંપૂર્ણતાને ઓળખે છે પ્રદેશની પ્રમાણભૂત ભાષા સુસંગતતાનો ઉપયોગ કરીને જેમ કે અંગ્રેજી યુકે અથવા અંગ્રેજી યુએસએ વગેરે. શબ્દોની સંખ્યા દર્શાવે છે કુલ શબ્દો, કુલ તરીકે દસ્તાવેજનું વિશ્લેષણ અક્ષરો, જગ્યાઓ સાથે અને વગર, ના. ફકરાઓનું, લીટીઓ વગેરે. અહીં આવા વિશ્લેષણ માટે ટેક્સ્ટ બ્લોક્સ પણ સામેલ કરી શકાય છે.

ભાષા બ્લોક પૃષ્ઠને માં અનુવાદિત કરવામાં મદદ કરે છે અન્ય ભાષાઓ ઈન્સ્ટોલ કરી અને પ્રૂફિંગ બદલવા માટે ભાષા તેને માઈક્રોસોફ્ટ ઓનલાઈનથી સીધી અનુવાદક સેવાની જરૂર છે.

Comments & Tracking Group (Fig 18)

ટિપ્પણીઓ જૂથ ચોક્કસ ફકરા અથવા ટેક્સ્ટ બ્લોક માટે ટિપ્પણી દાખલ કરે છે.

ટ્રેકિંગ ગ્રૂપ ડોક્યુમેન્ટમાં થયેલા ફેરફારો શોધે છે અન્ય લેખકો દ્વારા સુરક્ષિત સ્થિતિમાં. ઉદાહરણ તરીકે, જો એ દસ્તાવેજ વપરાશકર્તા X દ્વારા બનાવવામાં આવ્યો છે અને વપરાશકર્તા Y દ્વારા સંપાદિત કરવામાં આવ્યો છે ટ્રેક માર્ક્સ સાથે અલગથી ટ્રેક કરવામાં આવે છે.

Fig 18



Changes, Compare and Protect Groups (Fig 19)

આથી દસ્તાવેજો પર કરવામાં આવેલ ફેરફારો કાં તો પછી થઈ શકે છે સ્વીકાર્યું કે નકાર્યું. દસ્તાવેજ કરતાં વધુ હોઈ શકે છે સરખામણીનો ઉપયોગ કરીને સમાનતા માટે સરખામણી કરો. તેમજ ઘ દસ્તાવેજને અન્ય લેખકો દ્વારા સંપાદનથી સુરક્ષિત કરી શકાય છે.

Fig 19



Document views and show Groups (Fig 20)

વ્યુ ટેબ વર્ડ ડોક્યુમેન્ટ પ્રદર્શિત કરવાની રીત બતાવે છે. દસ્તાવેજ દૃશ્યોમાં પ્રિન્ટ લેઆઉટ હોય છે, એક સામાન્ય દૃશ્ય શબ્દનું, પૂર્ણ સ્ક્રીન વાંચન, ટેબને ઓછું કરે છે અને ઘોડાની લગામ અદૃશ્ય અને વાંચવા માટે સરળ, વેબ

Fig 20



લેઆઉટ, જે HTML સુસંગત દૃશ્યનું પૂર્વાવલોકન કરે છે, રૂપરેખા, મૂળભૂતને જુએ છે માત્ર ટેક્સ્ટ મોડમાં દસ્તાવેજનું સંસ્કરણ અને ટેક્સ્ટ એડિટિંગ મોડ માટે ડ્રાફ્ટ મોડ.

શાસકો, ગ્રીડલાઈન્સ અને નેવિગેશન પેન વપરાશકર્તાની પસંદગીઓ અનુસાર બતાવી અથવા છુપાવી શકાય છે.

Zoom / Window / Macro Groups (Fig 21)

Fig 21



તેમજ પેજના વ્યુને કુલ પેજ પર ઝૂમ કરી શકાય છે, બે પૃષ્ઠ દૃશ્ય, 100% દસ્તાવેજ અને કસ્ટમ દૃશ્ય. પ્રતિ વિન્ડોને વિભાજિત કરી શકાય તેવા દસ્તાવેજને સંપાદિત કરવાનું સરળ બનાવો બેમાં, દસ્તાવેજને કાપવા માટે નવી વિન્ડો, વગેરે. બનાવી શકાય છે અને તમામ ઓપન વર્ડ ડોક્યુમેન્ટ્સ બની શકે છે જોવા માટે વ્યવસ્થા કરી છે. દસ્તાવેજો ફોર્મેટ કરતી વખતે, માટે આદેશોનું પુનરાવર્તન મેક્રો વિકલ્પનો ઉપયોગ કરી શકાય છે. તે આદેશોના સમૂહને સંગ્રહિત કરવા માટે રેકૉર્ડ વિકલ્પનો ઉપયોગ કરે છે અને તેને દસ્તાવેજના અન્ય ભાગ પર અથવા અન્ય દસ્તાવેજ પર એક જ ક્લિક પર ફરીથી પુનરાવર્તન કરો.

ફાઈલ બનાવવી, સાચવો અને અન્ય વિકલ્પો (Creating a file, save and other options)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- નવો દસ્તાવેજ કેવી રીતે બનાવવો, સાચવો અને છાપો તે સમજાવો
- લખાણ અને દસ્તાવેજ શૈલીને કેવી રીતે સંપાદિત કરવી, ફોર્મેટ કરવું તે જણાવો
- ડેટા સાથે શબ્દ દસ્તાવેજની અંદર કોષ્ટકોનો સંક્ષિપ્ત ઉપયોગ
- દસ્તાવેજમાં શૈલીઓ કેવી રીતે બનાવવી અને ભવિષ્યના ઉપયોગ માટે કેવી રીતે સાચવવી તે સમજાવો
- વર્ડ 2010 ની કેટલીક વિશિષ્ટ વિશેષતાઓ સમજાવો
- મેઈલ મર્જ પ્રક્રિયાને સંક્ષિપ્ત કરો.

File Info view (Fig 1): વર્ડ 2010 નો નવો દસ્તાવેજ હંમેશાની જેમ સાથે બનાવી શકાય છે Ctrl + N અથવા ફાઈલ મેનૂ દ્વારા નવો વિકલ્પ. મૂળભૂત રીતે ઘ બનાવેલ ફાઈલ વર્ડ ફાઈલમાં વર્ડ ડોક્યુમેન્ટ એક્સટેન્શન ફોર્મેટસ docx તરીકે સાચવવામાં આવે છે.

Save options in Word (Fig 2):

તેને કોઈપણ સુસંગતમાં સેવ એઝ વિકલ્પનો ઉપયોગ કરીને સાચવી શકાય છે ફોર્મેટ અથવા ઓફિસના જૂના સંસ્કરણો, જેમ કે 2003 અથવા પહેલાનાં સંસ્કરણો. શબ્દ સોફ્ટવેરની મુખ્ય ઉપયોગિતા શબ્દની રચના છે પ્રક્રિયા દસ્તાવેજો. તે કોઈપણ ફોર્મેટ જેવા હોઈ શકે છે પ્રકાશન, પત્ર, પુસ્તિકા, વગેરે. વર્ડ તમામ પ્રકારના આધાર આપે છે ટેક્સ્ટ આધારિત

પ્રસ્તુતિ ડિઝાઇન કરવા માટે ફોર્મેટિંગનું. તે પણ ઉદ્યોગના ધોરણો અનુસાર મુખ્ય સ્વીકૃત ફોર્મેટમાં આઉટપુટ ફાઈલોને સપોર્ટ કરે છે.

Print options in word (Fig 3):

સાચવેલા દસ્તાવેજો ફાઈલ મેનૂનો ઉપયોગ કરીને પ્રિન્ટ કરી શકાય છે -> પ્રિન્ટ વિકલ્પ અને ઈન્સ્ટોલ કરેલ પ્રિન્ટર સપોર્ટ તેને બનાવે છે દસ્તાવેજ છાપવામાં સરળ છે. જો એડોબ એક્રોબેટ વ્યવસાયિક ઈન્સ્ટોલ કરેલું છે, તે જ દસ્તાવેજ હોઈ શકે છે શેરિંગ હેતુઓ માટે પીડીએફ ફાઈલ તરીકે સંગ્રહિત. સિવાય પ્રિન્ટિંગ અને સ્ટોરેજ, બનાવેલ ફાઈલમાં પ્રકાશિત કરી શકાય છે વેબ પેજ અથવા ટેમ્પલેટ તરીકે વેબ કે જેનો ઉપયોગ કરી શકાય છે ભવિષ્યના પ્રકાશનો સમાન ફોર્મેટમાં.

Fig 1

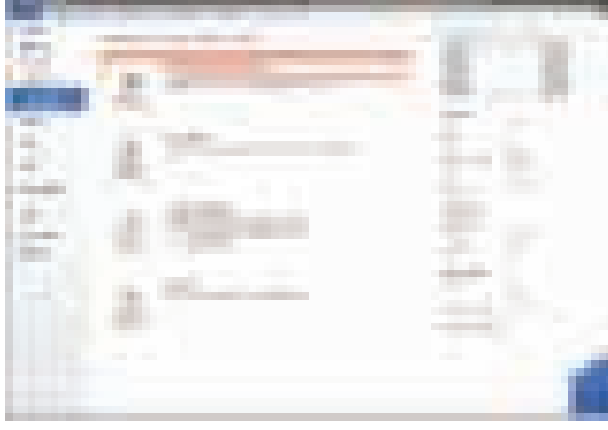


Fig 2

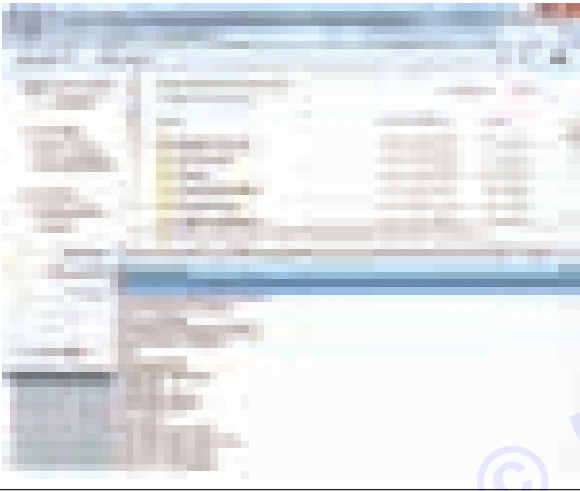


Fig 3



વર્ડનો ઉપયોગ કરતી વખતે ધ્યાનમાં લેવાના મૂળભૂત ખ્યાલો ટેક્સ્ટ છે ગુણધર્મો તેમાં ફોન્ટનો પ્રકાર, ફોન્ટનું કદ, લખાણનો રંગ, અને ટેક્સ્ટની સામાન્ય સજાવટ. ફકરાઓનું પણ નિર્માણ શૈલીઓ ધ્યાનમાં રાખવાની છે. ફકરા સંરેખણ ધરાવે છે ડાબે, જમણે, કેન્દ્ર અને ન્યાયી સેટિંગ્સ. ટેક્સ્ટ તત્વો ઓર્ડર કરેલ સૂચિ, અક્રમાંકિત સૂચિ, ઉપ- વિભાગ યાદીઓ. તેઓ ત્યાં ફકરા ફોર્મેટિંગ જોવા મળે છે હોમ ટેબનો બ્લોક. અવતરણ બનાવવા

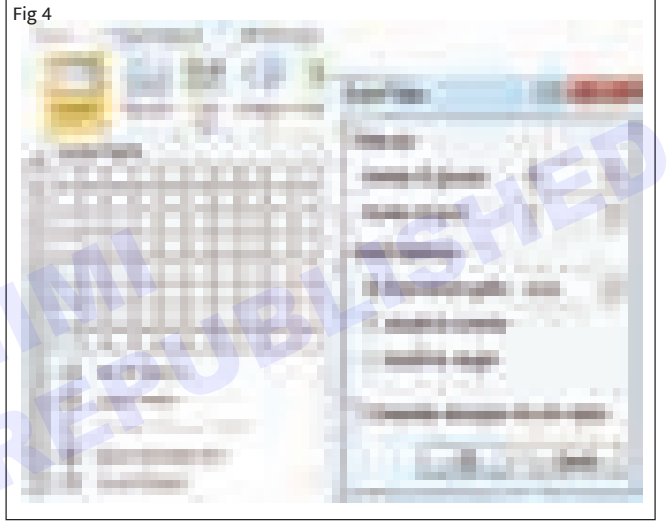
માટે ટેક્સ્ટનું ઇન્ડેન્ટિંગ તે જ ટેબની અંદર પણ છે. પૃષ્ઠ મુજબ કદ, રેખા અંતર અને ફકરા અંતર ગોઠવી શકાય છે, જેમ કે પહેલા અને પછી ફકરાની જગ્યાઓ, રેખાની ઊંચાઈઓ વગેરે. આ શૈલીઓ પર મૂકવા માટે તૈયાર તરીકે ઉપયોગ કરવા માટે પૂર્વવ્યાખ્યાયિત કરી શકાય છે સ્થાનો જ્યાં તે જરૂરી છે. માનક નમૂનાઓ છે ઉપલબ્ધ છે પરંતુ તે કસ્ટમ શૈલીઓ પણ બનાવવાની મંજૂરી આપે છે.

વિવિધ ઉપયોગિતાઓ માટે કોષ્ટકો દાખલ કરી શકાય છે. સામાન્ય ટેબલ એડજસ્ટેબલ પહોળાઈ અને કોલમ ધરાવે છે જેથી તે સીધા જ Insert > Table વિકલ્પ સાથે દાખલ કરી શકાય.

Table insert wizard options (Fig 4)

જો કસ્ટમ માપના ટેબલની આવશ્યકતા હોય, તો તે બનાવી શકાય છે દાખલ ટેબલ / ડ્રો ટેબલ વિકલ્પોનો ઉપયોગ કરીને. દાખલ ટેબલ પરવાનગી આપે છે કોલમ, પહોળાઈ અને અન્ય ગુણધર્મો માટે કસ્ટમ મૂલ્યો. ડ્રો ટેબલ પેન ડ્રોઈંગ ટૂલનો ઉપયોગ કરીને કસ્ટમ ટેબલ બનાવે છે જેના દ્વારા નવા ટેબલ મુજબ ડ્રો કરી શકાય છે ઉપલબ્ધ ટેક્સ્ટ સામગ્રી.

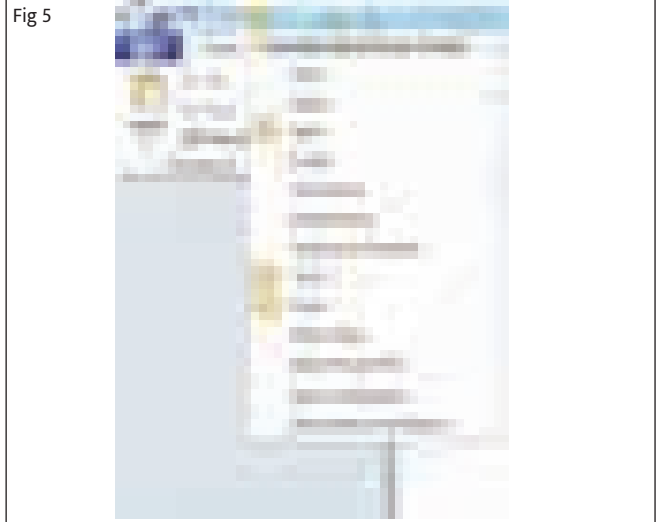
Fig 4



કસ્ટમાઈઝ કરવા યોગ્ય ક્વિક એક્સેસ ટૂલ બાર

વર્ડ 2010 નું ક્વિક એક્સેસ ટૂલબાર તમામ દર્શાવે છે સામાન્ય રીતે ઉપયોગમાં લેવાતા વિકલ્પો. તે ઉપરની ડાબી બાજુએ સ્થિત છે ઓફિસ બટનની નજીક, એપ્લિકેશન વિંડોનો ખૂણો. મૂળભૂત રીતે તે નીચેના ત્રણ વિકલ્પો દર્શાવે છે, સાચવો, પૂર્વવત્ કરો અને ફરીથી કરો, પરંતુ વૈવિધ્યપૂર્ણ છે અને તમે સરળતાથી કરી શકો છો તેમાં વધુ વિકલ્પો ઉમેરો. (ફિગ 5)

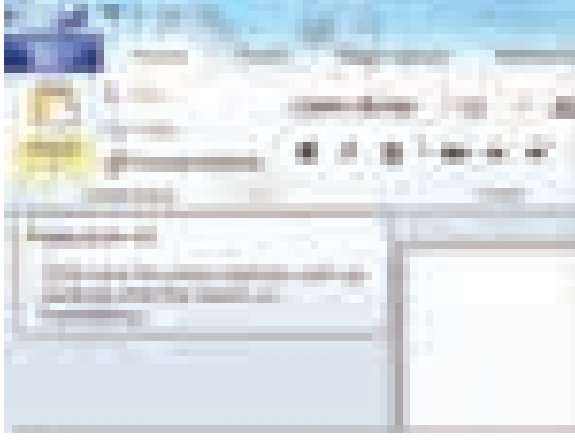
Fig 5



Paste Preview

તે મોટાભાગના વપરાશકર્તાઓ સાથે થાય છે કે કોપી અને પેસ્ટ કર્યા પછી તેમના દસ્તાવેજમાં કંઈક, તેઓને પૂર્વવત્ કરવાની જરૂર છે કેટલાક ફેરફારો. વર્ડ 2010 એ વપરાશકર્તાઓ માટે સરળ બનાવ્યું છે, હવે તમે ઉપયોગ કરીને આ બિનજરૂરી પગલાને દૂર કરી શકો છો પેસ્ટ પૂર્વાવલોકન વિકલ્પ. તે વપરાશકર્તાઓને ફક્ત પેસ્ટ કરવાની મંજૂરી આપે છે મૂલ્યો અથવા ફોર્મેટિંગ. (ફિગ 6)

Fig 6



Navigation Pane

માઈક્રોસોફ્ટ ઓફિસના પાછલા વર્ઝનમાં, વ્યક્તિએ કરવું પડશે કોઈપણ શબ્દ અથવા શબ્દસમૂહ શોધવા માટે Ctrl+F હોટકીનો ઉપયોગ કરો દસ્તાવેજની અંદર. વર્ડ 2010 એ એક નવો જાદુ ઉમેર્યો છે આ વિકલ્પ, Ctrl+F હવે નેવિગેશન પેનને બોલાવે છે દસ્તાવેજની ડાબી બાજુએ દેખાય છે. તમે જોશો ત્રણ દૃશ્યો તેમના સંબંધિત ટેબ પર ક્લિક કરીને ઉપલબ્ધ છે, મથાળું દૃશ્ય, થંબનેલ પૃષ્ઠ દૃશ્ય અને શોધ પરિણામ જુઓ. (ફિગ 7)

Customizable Ribbon Button

દેખીતી રીતે વર્ડ 2010 માં રિબન બટન જેવો દેખાય છે વર્ડ 2007 માં એક. પરંતુ તેમાં એક મોટો ઉમેરો છે, તમે કરી શકો છો શબ્દ 2010 ના રિબન બટનને કસ્ટમાઈઝ કરો. ના અનુસાર રિબન બટનને કસ્ટમાઈઝ કરો નીચેના વિકલ્પ પર નેવિગેટ કરો ઓફિસ બટન > શબ્દ વિકલ્પ > કસ્ટમાઈઝ રિબન. (ફિગ 8)

Fig 7



Fig 8



Improved Collaborations

માઈક્રોસોફ્ટ વર્ડ 2010 માં કો- લેબન તે એક કરતાં વધુ લેખકોને સંપાદિત કરવાની મંજૂરી આપે છે તે જ સમયે દસ્તાવેજ. વર્ડ 2010 તમને કેવી રીતે કહે છે ઘણા લેખકો દસ્તાવેજ અને તેમના ફેરફારોને સંપાદિત કરી રહ્યાં છે પણ જોઈ શકાય છે. (ફિગ 9)

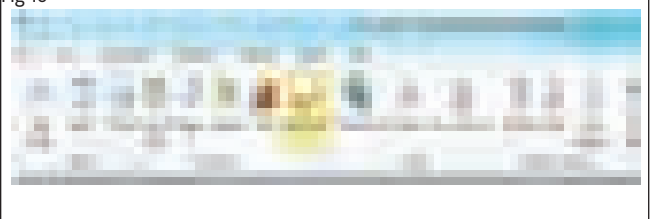
Fig 9



Screen Capture Tool

વર્ડ 2010 માં સ્ક્રીન કેપ્ચરિંગ નામની સુવિધા શામેલ છે, હવે તૃતીય પક્ષ અથવા વધારાના સાધનનો ઉપયોગ કરવાની જરૂર નથી વર્ડમાં તેનો ઉપયોગ કરવા માટે સ્ક્રીનશોટ મેળવવા માટે, બસ કોઈપણ વિસ્તારને મેળવવા માટે ફક્ત Word 2010 ના બિલ્ટ-ઇન ટૂલનો ઉપયોગ કરો સ્ક્રીન પર નેવિગેટ કરીને સ્ક્રીનશોટ લેવામાં આવી શકે છે નીચેનો વિકલ્પ Insert > Screen-shot. (ફિગ 10)

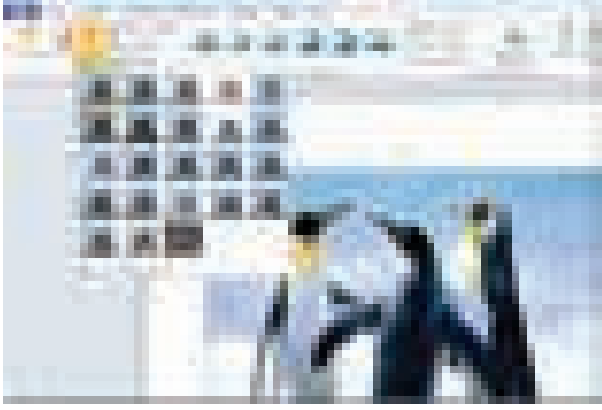
Fig 10



Artistic Effects

વર્ડ 2010માં વપરાશકર્તાઓ હવે સંખ્યાબંધ સ્નેઝી અરજી કરી શકે છે ચિત્રો પર કલાત્મક અસરો. કલાત્મક ઉમેરવા માટે તમારા દસ્તાવેજ પર અસરો, નીચેના વિકલ્પ પર નેવિગેટ કરો દાખલ કરો > ચિત્રો > ચિત્ર. પછી બ્રાઉઝ કરો અને પસંદ કરો તમે ઇચ્છો છો તે ચિત્ર, એકવાર ચિત્ર તમારામાં ઉમેરવામાં આવે છે દસ્તાવેજ, પછી ચિત્ર સાધનો સંદર્ભ ટેબ છે પ્રદર્શિત થાય છે અને તમે નવી આર્ટિસ્ટિક જોવા માટે સમર્થ હશો ઇફેક્ટ્સ ડાઉન બટન અહીંથી નીચે આવે છે. (ફિગ 11)

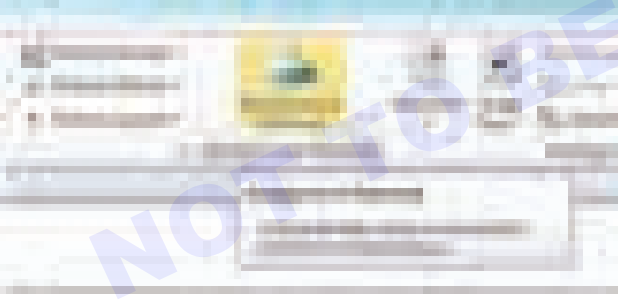
Fig 11



The Background Removal Option

ઓફિસ 2010 ના નામથી એક અદ્ભુત વિકલ્પ ધરાવે છે પૃષ્ઠભૂમિ દૂર, જે ખાલી દૂર કરે છે કોઈપણ છબીની પૃષ્ઠભૂમિ. હા, તમારે ફોટોશોપની જરૂર નથી હવે પૃષ્ઠભૂમિ દૂર કરવા માટે. પ્રથમ દાખલ કરો Insert > Picture માંથી તમારા વર્ડ ડોક્યુમેન્ટમાં ચિત્ર વિકલ્પ. પછી પૃષ્ઠભૂમિ દૂર સાધન સ્થિત કરો અને મેળવો પૃષ્ઠભૂમિ છુટકારો. (ફિગ 12)

Fig 12



Office Backstage

ઓફિસ બેકસ્ટેજ એક નવો કોન્સેપ્ટ છે, તે ઉન્નત છે સાદા જૂના ઓફિસ બટનનું સ્વરૂપ અને ઘણું બધું પ્રદાન કરે છે વપરાશકર્તા મૈત્રીપૂર્ણ મેનુ. તે વપરાશકર્તાઓને દસ્તાવેજોનું સંચાલન કરવામાં મદદ કરે છે, પ્રસ્તુતિઓ, અથવા સ્પ્રેડશીટ્સ વધુ સ્તરે. (ફિગ 13)

રસપ્રદ લક્ષણ, અધિકાર? તેના વિશે અહીં વધુ વાંચો.

New Art Effects in WordArt

અન્ય સુવિધાઓની જેમ, વર્ડઆર્ટ સાથે અપડેટ કરવામાં આવ્યું છે નવી રંગીન કલા અસરો. ટેક્સ્ટ પસંદ કરો, પછી વર્ડ પર ક્લિક કરો કલા અને તમામ ઉપલબ્ધ વિકલ્પોની સૂચિ પ્રદર્શિત કરવામાં આવશે. (ફિગ 14)

Fig 13

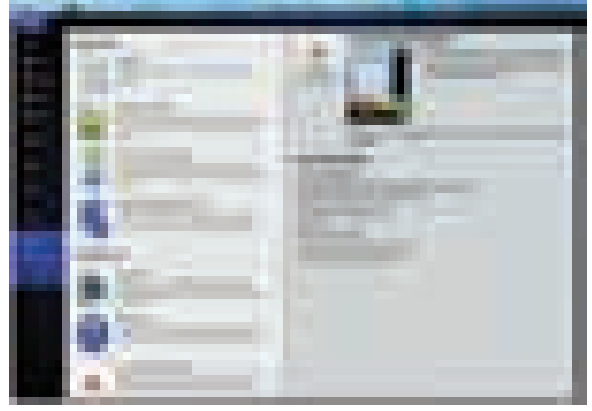


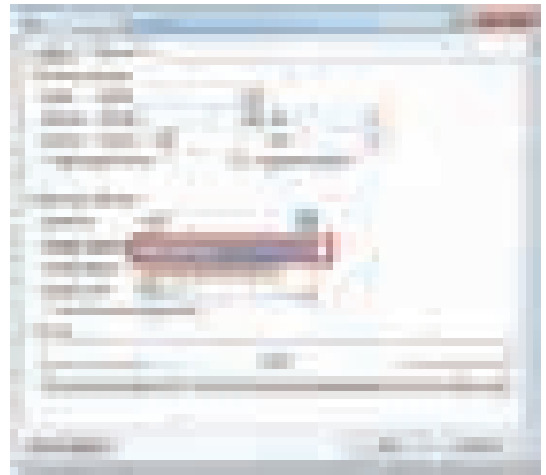
Fig 14



Ligatures

તમે Ligatures વિશે સાંભળ્યું હશે. તેઓ બનાવે છે ફોન્ટ્સ ફ્રેન્સી લાગે છે અને તેનો ઉપયોગ અક્ષરો રાખવા માટે પણ થાય છે અલગથી તમે ટેક્સ્ટને ફોન્ટની જેમ શોધી શકો છો નિયમિત હતા. તે સાચું છે કે બધા ફોન્ટ્સ લિગેચર્સને સપોર્ટ કરતા નથી, પરંતુ ફોન્ટ્સની વિશાળ વિવિધતા તેમને સપોર્ટ કરે છે. તમે કરી શકો છો તેમને ફોન્ટ પસંદગીઓ > અદ્યતન માંથી સક્ષમ કરો, પછી પસંદ કરો અસ્થિબંધન ડ્રોપ ડાઉન બોક્સમાં પ્રમાણભૂત એકમાત્ર વિકલ્પ. (ફિગ 15)

Fig 15



Improved Spell Checks

વર્ડ 2010 એ તેની જોડણીમાં કેટલીક નવી વિશેષતાઓ ઉમેરી છે તપાસનાર, હવે તે ભૂલ શોધીને સૂચન કરશે વાક્ય બદલવું.

Fig 16



વર્ડ 2010 માં શોર્ટકટ કી (Shortcut keys in Word 2010)

ઉદ્દેશ્ય: પાઠના અંતે તમે સક્ષમ થશો

- MS Word માં શોર્ટકટ કી શીખો.

CTRL+SHIFT+A	converts the selected text to capital letters or vice versa
CTRL+SHIFT+F	Displays the Font dialog box.
CTRL+SHIFT+G	Displays the Word Count dialog box.
CTRL+SHIFT+S	Displays the Apply Styles task pane.
ALT+R	Displays the Review tab
Displays the Review tab	Apply Heading 1, Similarly ALT + CTRL + 2 will apply heading 2
CTRL+SHIFT+L	Applies Bullets
CTRL+SHIFT+F5	Bookmark
CTRL + B	Bold Text
CTRL + I	Italic Text
CTRL + U	Underline Text
CTRL+PAGE DOWN	Browse Next
CTRL+E	Navigate to the center Paragraph
CTRL+SHIFT+ENTER	Column Break
CTRL+SHIFT+C	Copy Format
ALT+SHIFT+F7	Dictionary
ALT+CTRL+S	Splits the Document
CTRL+SHIFT+D	Double Underline
CTRL+END	End of Document
END	End of line
CTRL+SHIFT+P	Font size select
SHIFT+F5 or ALT+CTRL+Z	Go Back to previous state

CTRL+SHIFT+.	Grow Font
CTRL+]	Grow Font one point
ALT+SHIFT+R	Header Footer Link
CTRL+K	Hyperlink
CTRL+M	Indentation
CTRL+J	Justifies Paragraph
ALT+F8	Inserts Macros
ALT+SHIFT+K	Mail Merge Check
F10	Menu Mode
ALT+F7	Moves to the Next Misspelling
CTRL+H	Replace
CTRL+P	Print
CTRL+SHIFT+F12	Also launches Print
ALT+SHIFT+BACKSPACE	Redo
F12	Save As
CTRL+SHIFT+K	Small Caps
CTRL+SHIFT+S	Style
SHIFT+F7	Thesaurus
ALT+SHIFT+T	Time Field
CTRL+SHIFT+M	Unindent

દાખલ કરો, ટેક્સ્ટ અને ફકરાઓને ફોર્મેટ કરો, દસ્તાવેજ વિભાગો બનાવો અને ગોઠવો (Insert, format text and paragraphs, Create and configure document sections)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- એમએસ વર્ડમાં ટેક્સ્ટ દાખલ કરો
- ફોર્મેટ ટેક્સ્ટ
- એક વિભાગ વિરામ ઉમેરો.

Insert Text in MS Word

તમે રિબનની નીચે ટેક્સ્ટ એરિયામાં ઝબકતું કર્સર અથવા નિવેશ બિંદુ જોશો. હવે, જેમ તમે ટાઇપ કરવાનું શરૂ કરશો, ટેક્સ્ટ એરિયામાં શબ્દો સ્ક્રીન પર દેખાશે. નિવેશ બિંદુનું સ્થાન બદલવા માટે સ્પેસબાર, એન્ટર અથવા ટેબ કી દબાવો.

વર્ડમાં ટેક્સ્ટ દાખલ કરવા અથવા નવો દસ્તાવેજ બનાવવા માટેના મૂળભૂત પગલાં નીચે સૂચિબદ્ધ છે;

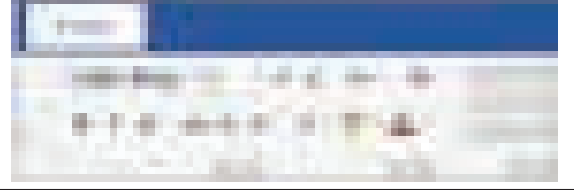
- સ્ટાર્ટ મેનૂ પર જાઓ અને Microsoft Word ચિહ્ન માટે જુઓ
- Microsoft Word ખોલવા માટે આયકન પર ક્લિક કરો
- તમે રિબનની નીચે ટેક્સ્ટ એરિયામાં ઝબકતું કર્સર અથવા નિવેશ બિંદુ જોશો
- હવે, જેમ તમે ટાઇપ કરવાનું શરૂ કરશો, શબ્દો ટેક્સ્ટ એરિયામાં સ્ક્રીન પર દેખાશે
- નિવેશ પોઇન્ટનું સ્થાન બદલવા માટે સ્પેસબાર, એન્ટર અથવા ટેબ કી દબાવો

Fig 1



Format text

Fig 2



- 1 તમે ફોર્મેટ કરવા માંગો છો તે ટેક્સ્ટ પસંદ કરો.
એક શબ્દ પસંદ કરવા માટે, તેના પર ડબલ-ક્લિક કરો. ટેક્સ્ટની લાઈન પસંદ કરવા માટે, તેની ડાબી બાજુએ ક્લિક કરો.
- 2 ફોન્ટ, ફોન્ટનું કદ, ફોન્ટ રંગ બદલવા અથવા ટેક્સ્ટને બોલ્ડ, ઇટાલિક અથવા અન્ડરલાઈન બનાવવા માટે વિકલ્પ પસંદ કરો.

Copy formatting

- 1 તમે નકલ કરવા માંગો છો તે ફોર્મેટિંગ સાથેનો ટેક્સ્ટ પસંદ કરો.
- 2 ફોર્મેટ પેઇન્ટર પર ક્લિક કરો અને પછી તમે ફોર્મેટિંગની નકલ કરવા માંગો છો તે ટેક્સ્ટ પસંદ કરો.

Tip : જો તમે એક કરતાં વધુ જગ્યાએ ફોર્મેટિંગની નકલ કરવા માંગતા હોવ તો ફોર્મેટ પેઇન્ટર પર ડબલ-ક્લિક કરો.

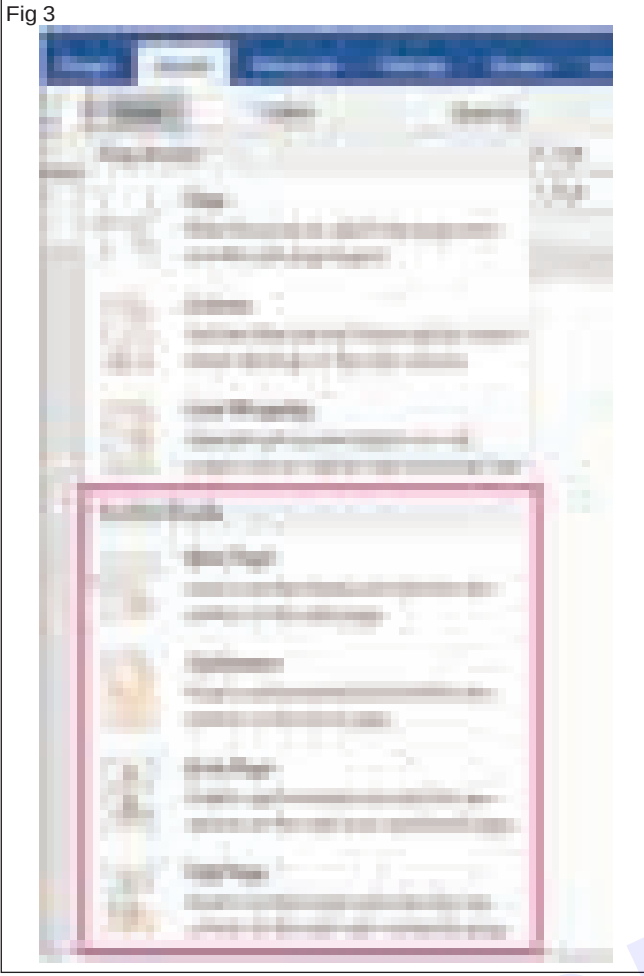
Add a section break

- 1 તમે જ્યાંથી નવો વિભાગ શરૂ કરવા માંગો છો તે પસંદ કરો.
- 2 Go to **Layout > Breaks**
- 3 તમને જોઈતા વિભાગ વિરામનો પ્રકાર પસંદ કરો: આગાવું પૃષ્ઠ વિભાગ વિરામ નીચેના પૃષ્ઠ પર નવો વિભાગ શરૂ કરે છે. સતત વિભાગ વિરામ એ જ પૃષ્ઠ પર નવો વિભાગ શરૂ કરે છે

Add a section break

- 1 તમે જ્યાંથી નવો વિભાગ શરૂ કરવા માંગો છો તે પસંદ કરો.
- 2 Go to **Layout > Breaks**

Fig 3



3 તમને જોઈતા વિભાગ વિરામનો પ્રકાર પસંદ કરો:

- **Next Page** - વિભાગ વિરામ નીચેના પૃષ્ઠ પર નવો વિભાગ શરૂ કરે છે



- **Continuous** - વિભાગ વિરામ એ જ પૃષ્ઠ પર નવો વિભાગ શરૂ કરે છે. આ પ્રકારના વિભાગ વિરામનો ઉપયોગ ઘણીવાર નવું પૃષ્ઠ શરૂ કર્યા વિના કોલમની સંખ્યા બદલવા માટે થાય છે.



- **Even પેજ** - સેક્શન બ્રેક આગલા સમ-ક્રમાંકિત પૃષ્ઠ પર એક નવો વિભાગ શરૂ કરે છે.



- **Odd પૃષ્ઠ** - વિભાગ વિરામ આગલા બેકી-ક્રમાંકિત પૃષ્ઠ પર એક નવો વિભાગ શરૂ કરે છે



કોષ્ટકો બનાવો, સંશોધિત કરો (Create, modify tables)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- કોષ્ટકો બનાવો
- ટેક્સ્ટને ટેબલમાં કન્વર્ટ કરો
- ટેબલ વિભાજિત કરો.

મોટા ટેબલ માટે અથવા ટેબલને કસ્ટમાઈઝ કરવા માટે, Insert > Table > Insert Table પસંદ કરો. ટિપ્સ: જો તમારી પાસે પહેલાથી જ ટેબલ દ્વારા અલગ કરેલ ટેક્સ્ટ હોય, તો તમે તેને ઝડપથી ટેબલમાં કન્વર્ટ કરી શકો છો. ઈન્સર્ટ > ટેબલ પસંદ કરો અને પછી ટેક્સ્ટને ટેબલમાં કન્વર્ટ કરો પસંદ કરો.

Create tables

ડેટા સેટ્સ, ખાસ કરીને આંકડાકીય માહિતીના, ઘણીવાર ટેક્સ્ટના ફક્ત કરતાં કોષ્ટકમાં વધુ સ્પષ્ટ અને અસરકારક રીતે રજૂ કરી શકાય છે. કોષ્ટકો મોટી માત્રામાં ડેટા, અથવા જટિલ ડેટા, એક ફોર્મેટમાં રજૂ કરે છે જે તેને પંક્તિઓ અને કોલમમાં સંરચિત કરીને વાંચવા અને સમજવામાં સરળ હોય છે, જેમાં ઘણીવાર ડેટાના હેતુ અથવા અર્થને સમજાવવા માટે હેડરનો સમાવેશ થાય છે.

ટેક્સ્ટને ટેબલમાં અથવા ટેબલને ટેક્સ્ટમાં કન્વર્ટ કરવા માટે, હોમ ટેબ પર ફક્ત બતાવો/છુપાવો પર ક્લિક કરીને પ્રારંભ કરો જેથી તમે જોઈ શકો કે તમારા દસ્તાવેજમાં ટેક્સ્ટ કેવી રીતે અલગ કરવામાં આવે છે.



Convert text to a table

- 1 વિભાજક અક્ષરો દાખલ કરો-જેમ કે અલ્પવિરામ અથવા ટેબ્સ લખાણને કોષ્ટક કોલમમાં ક્યાં વિભાજિત કરવું તે સૂચવવા માટે.

નોંધ: જો તમારી પાસે તમારા ટેક્સ્ટમાં અલ્પવિરામ છે, તો તમારા વિભાજક અક્ષરો માટે ટેબ્સનો ઉપયોગ કરો.

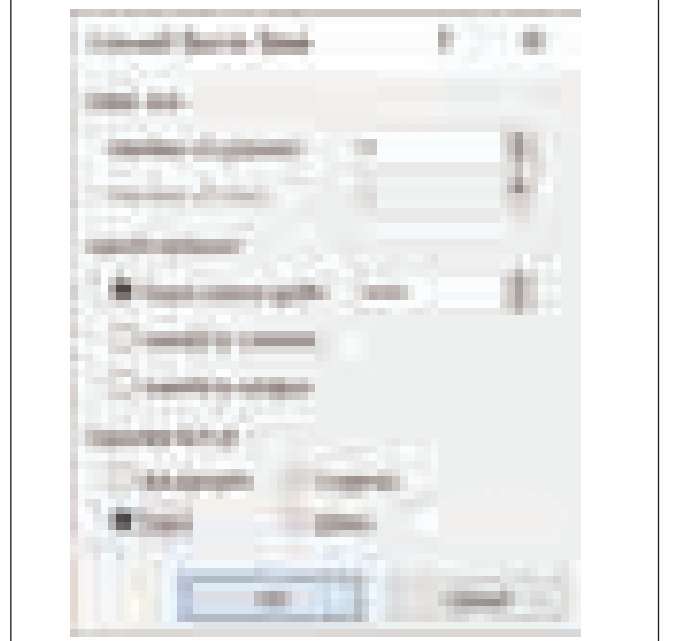
- 2 તમે નવી કોષ્ટક પંક્તિ ક્યાંથી શરૂ કરવા માંગો છો તે દર્શાવવા માટે ફ્લેચના ચિહ્નોનો ઉપયોગ કરો.

આ ઉદાહરણમાં, ટેબ્સ અને ફ્લેચના ચિહ્નો 3 કોલમ અને 2 પંક્તિઓ સાથે કોષ્ટક બનાવશે:

RED, YELLOW ---> BLUE, GREEN ---> ORANGE
PURPLE

RED, YELLOW ---> BLUE, GREEN ---> ORANGE,
PURPLE

- 3 તમે કન્વર્ટ કરવા માંગો છો તે ટેક્સ્ટ પસંદ કરો, અને પછી ક્લિક કરો દાખલ કરો > ટેબલ > ટેક્સ્ટને ટેબલમાં કન્વર્ટ કરો.
- 4 કન્વર્ટ ટેક્સ્ટ ટુ ટેબલ બોક્સમાં, તમને જોઈતા વિકલ્પો પસંદ કરો. કોષ્ટકના કદ હેઠળ, ખાતરી કરો કે નંબરો તમને જોઈતી કોલમ અને પંક્તિઓની સંખ્યા સાથે મેળ ખાય છે.



- 1 ઓટોફિટ વર્તન હેઠળ, તમે તમારું ટેબલ કેવું દેખાવા માંગો છો તે પસંદ કરો. શબ્દ આપમેળે ટેબલ કોલમ માટે પહોળાઈ પસંદ કરે છે. જો તમને અલગ કોલમની પહોળાઈ જોઈતી હોય, તો આ વિકલ્પોમાંથી એક પસંદ કરો:

To do this	Choose this option
Specify a width for all the columns	In the Fixed column width box , type or select a value.
Resize the columns to fit the width of the text in each column	AutoFit to contents
Resize the table automatically in case the width of the available space changes (for example, web layout or landscape orientation)	AutoFit to window

- 2 પર અલગ ટેક્સ્ટ હેઠળ, તમે ટેક્સ્ટમાં ઉપયોગમાં લીધેલ વિભાજક અક્ષર પસંદ કરો.
- 3 બરાબર ક્લિક કરો. પછી કોષ્ટકમાં રૂપાંતરિત ટેક્સ્ટ આના જેવું કંઈક દેખાવું જોઈએ:

Red, yellow	blue, green	orange, purple
Red, yellow	blue, green	orange, purple

Split a table

- 1 તમારા કર્સરને તે પંક્તિ પર મૂકો જે તમે તમારા બીજા કોષ્ટકની પ્રથમ પંક્તિ તરીકે ઇચ્છો છો. ઉદાહરણ કોષ્ટકમાં, તે ત્રીજી પંક્તિ પર છે....
- 2 લેઆઉટ ટેબ પર, મર્જ જૂથમાં, વિભાજિત કોષ્ટક પર ક્લિક કરો. ટેબલ બે કોષ્ટકોમાં વિભાજિત થાય છે.

એકવાર તમારી પાસે વર્ડમાં કોષ્ટક હોય, તો તમે તે કોષ્ટકને બે અથવા વધુ કોષ્ટકોમાં વિભાજિત કરવાનું નક્કી કરી શકો છો. આ રીતે, તમે નાના કોષ્ટકો બનાવી શકો છો, અથવા બે કોષ્ટકો વચ્ચે ટેક્સ્ટ ઉમેરી શકો છો.

- 1 તમારા કર્સરને તે પંક્તિ પર મૂકો જે તમે તમારા બીજા કોષ્ટકની પ્રથમ પંક્તિ તરીકે ઇચ્છો છો. ઉદાહરણ કોષ્ટકમાં, તે ત્રીજી પંક્તિ પર છે.

1	A
2	B
3	C
4	D

જ્યારે તમે કોષ્ટકની અંદર ક્લિક કરો છો, ત્યારે રિબન પર બે નવા ટેબલ ટૂલ્સ ટેબ્સ દેખાય છે: ડિઝાઇન અને લેઆઉટ. આ સાધનો ફક્ત ત્યારે જ દેખાય છે જ્યારે તમે કોષ્ટકમાં હોવ, ઉદાહરણ તરીકે, જ્યારે 1 તમે કોષોમાં સામગ્રી ઉમેરો છો.

- 2 લેઆઉટ ટેબ પર, મર્જ જૂથમાં, વિભાજિત કોષ્ટક પર ક્લિક કરો.



ટેબલ બે કોષ્ટકોમાં વિભાજિત થાય છે.

જ્યાં સુધી બહુવિધ પંક્તિઓ હોય ત્યાં સુધી તમે કોષ્ટકને વધુ વિભાજિત કરી શકો છો

1	A
2	B
3	C
4	D

COPA - સંદર્ભો બનાવો અને મેનેજ કરો

સંદર્ભ તત્વો અને કોષ્ટકો બનાવો અને મેનેજ કરો (Create and manage reference elements and tables)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સંદર્ભ તત્વોનું સંચાલન કરો
- વિષયવસ્તુના કોષ્ટકોને કસ્ટમાઈઝ કરો.

શબ્દની પોતાની, મૂળભૂત સંદર્ભ સિસ્ટમ છે જે રિબન પરના 'સંદર્ભ' ટેબ દ્વારા ઉપલબ્ધ છે. આ તમને દસ્તાવેજમાં સંદર્ભો ઉમેરવા દે છે અને પછી ટેક્સ્ટના અંતે એક ગ્રંથસૂચિ બનાવી શકે છે. સંદર્ભો મુખ્ય સૂચિમાં સંગ્રહિત થાય છે, જેનો ઉપયોગ આગળના દસ્તાવેજોમાં સંદર્ભ ઉમેરવા માટે થઈ શકે છે.

નોંધ: આ સિસ્ટમ EndNote, Mendeley અથવા Zotero જેટલી શક્તિશાળી નથી

પ્રથમ પગલાં

- 1 તમારો દસ્તાવેજ બનાવો.
- 2 ટેક્સ્ટમાં અવતરણ દાખલ કરવા માટે રિબન પરના 'સંદર્ભ' ટેબ પર જાઓ અને 'સંદર્ભ દાખલ કરો' અને 'નવા સ્ત્રોત ઉમેરો' પર ક્લિક કરો.
- 3 'સ્ત્રોતનો પ્રકાર' પસંદ કરો અને બોક્સ ભરો.
- 4 તમારા દસ્તાવેજમાં તમામ અવતરણો ઉમેરો.
- 5 જ્યારે તમે પૂર્ણ કરી લો, ત્યારે તમારા દસ્તાવેજના અંતમાં જાઓ અને 'બિબ્લિયોગ્રાફી' વિકલ્પ પર ક્લિક કરો. પ્રીફોર્મેટ કરેલા વિકલ્પોમાંથી એક પસંદ કરો અથવા ફક્ત પોતાને ફોર્મેટ કરવા માટે ગ્રંથસૂચિ દાખલ કરો.
- 6 ડિફોલ્ટ 'APA' શૈલીમાંથી તમારા સંદર્ભોની શૈલી બદલવા માટે 'શૈલી' સૂચિ પર ક્લિક કરો અને બીજી પસંદ કરો.

નોંધ કરો કે વર્ડમાં ઉપલબ્ધ શૈલીઓ તમારા અસાઈનમેન્ટ માટે જરૂરી હોય તે સાથે મેળ ખાતી નથી. તેમને કાળજીપૂર્વક તપાસો અને જો તેઓ મેળ ખાતા ન હોય તો તમે ગ્રંથસૂચિને સંપાદનયોગ્ય ટેક્સ્ટમાં કન્વર્ટ કરી શકો છો. આ કરવા માટેની સૂચના માટે નીચેની માર્ગદર્શિકા જુઓ.

વર્ડમાંના સંદર્ભો તમારા કમ્પ્યુટરની હાર્ડ ડિસ્ક પર સંગ્રહિત થાય છે. જો તમે બીજા કમ્પ્યુટર પર તેમના પર કામ કરવા માંગતા હો, તો ફાઈલ શોધવા માટે આ પગલાં અનુસરો:

- સ્ટાર્ટ મેનૂ સર્ચ બોક્સમાં % APPDATA% લખો
- Roaming >; Microsoft > Bibliography ફોલ્ડર પર ક્લિક કરો
- સંદર્ભો XML ફાઈલમાં સાચવવામાં આવે છે

References in Word

માઈક્રોસોફ્ટ વર્ડમાં સંદર્ભો ટેબમાં એક સરળ બિલ્ટ-ઇન સંદર્ભ કાર્ય છે. અહીં તમે મેન્યુઅલી સંદર્ભો દાખલ કરી શકો છો અને ટેક્સ્ટમાં તેનો સંદર્ભ લઈ શકો છો અને સૌથી સામાન્ય શૈલીઓ (APA, હાર્વર્ડ, MLA, વગેરે)માં ગ્રંથસૂચિ બનાવી શકો છો. આ બિલ્ટ-ઇન ફંક્શન સારું છે જો તમારી પાસે મર્યાદિત સંખ્યામાં સંદર્ભો હોય અને તમારે વિવિધ કમ્પ્યુટર્સ પર તમારા સંદર્ભોની એક્સેસની જરૂર નથી. રિબન પર, સંદર્ભો પર જાઓ. ડાબી બાજુના વિષયવસ્તુ જૂથ હેઠળ, વિષયવસ્તુના કોષ્ટક બટનને ક્લિક કરો. તમે ડિફોલ્ટ વિકલ્પ દાખલ કરવાનું પસંદ કરી શકો છો અથવા મેનૂના તળિયે કસ્ટમ ટેબલ ઓફ કન્ટેન્ટ પર ક્લિક કરી શકો છો. વિષયવસ્તુનું કસ્ટમ કોષ્ટક દાખલ કરવા માટે, મેનુમાંથી વિકલ્પ પસંદ કરો.

તમે વર્ડમાં વિષયવસ્તુનું કોષ્ટક બનાવી લો તે પછી, તમે તે દેખાય તે રીતે કસ્ટમાઈઝ કરી શકો છો. તમારા વિષયવસ્તુના કોષ્ટકને કસ્ટમાઈઝ કરવાથી તમારી પસંદગીઓ લાગુ થાય છે તમારું હાલનું ટેબલ. તમે જે જુઓ છો તે પસંદ કરો, બરાબર પસંદ કરો. જો તમે દેખાવથી ખુશ નથી, તો ફક્ત રદ કરો પસંદ કરો અને બધા ફેરફારો ભૂલી ગયા. તમારી સામગ્રીઓનું કોષ્ટક હજી પણ ત્યાં હશે.

તમારી હાલની સામગ્રીઓનું કોષ્ટક કસ્ટમાઈઝ કરવા માટે:

- 1 સંદર્ભો >; વિષયવસ્તુના કોષ્ટક પર જાઓ.
- 2 વિષયવસ્તુનું વૈવિધ્યપૂર્ણ કોષ્ટક પસંદ કરો.
- 3 પૃષ્ઠ નંબરો બતાવવા, છુપાવવા અને સંરેખિત કરવા માટે સેટિંગ્સનો ઉપયોગ કરો, ટેબ લીડર ઉમેરો અથવા બદલો, ફોર્મેટ્સ સેટ કરો અને મથાળાના કેટલા સ્તરો દર્શાવવા તે સ્પષ્ટ કરો. વધુ માહિતી માટે, સામગ્રીઓનું કસ્ટમ ટેબલ જુઓ.



ઇન્સર્ટ, ફોર્મેટ ચિત્રો અને ટેક્સ્ટ બોક્સનું સંચાલન કરો (Insert, format illustrations and text boxes)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ટેક્સ્ટ બોક્સ
- ગ્રાફિક ઘટકોમાં ટેક્સ્ટ ઉમેરો.

ટેક્સ્ટ બોક્સ એ એક ઓબ્જેક્ટ છે જેને તમે તમારા દસ્તાવેજમાં ઉમેરી શકો છો જે તમને તમારી ફાઇલમાં ગમે ત્યાં ટેક્સ્ટ મૂકવા અને લખવા દે છે. ટેક્સ્ટ બોક્સ ચોક્કસ ટેક્સ્ટ પર ધ્યાન દોરવા માટે ઉપયોગી થઈ શકે છે અને જ્યારે તમારે તમારા દસ્તાવેજમાં ટેક્સ્ટને ફરતે ખસેડવાની જરૂર હોય ત્યારે પણ તે મદદરૂપ થઈ શકે છે.

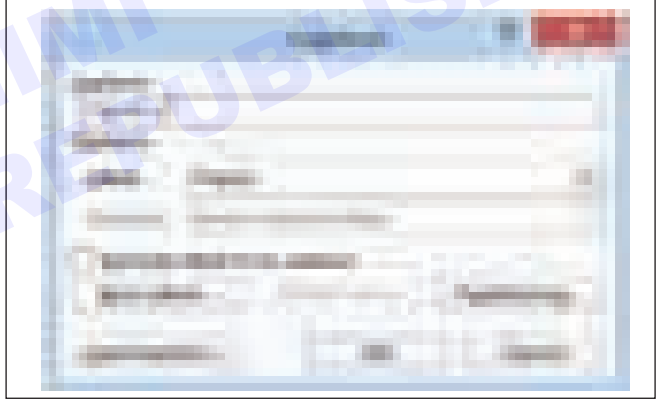
Insert > Text Box પર જાઓ અને પછી યાદીમાંથી પૂર્વ-ફોર્મેટ કરેલ ટેક્સ્ટ બોક્સમાંથી એક પસંદ કરો, Office.com માંથી વધુ ટેક્સ્ટ બોક્સ પસંદ કરો અથવા ફ્રી ટેક્સ્ટ બોક્સ પસંદ કરો. જો તમે ટેક્સ્ટ બોક્સ દોરો પસંદ કરો છો, તો દસ્તાવેજમાં ક્લિક કરો, અને પછી તમે ઇચ્છો તે કદના ટેક્સ્ટ બોક્સને દોરવા માટે ખેંચો.

- ટેક્સ્ટ બોક્સમાં ટેક્સ્ટને ફોર્મેટ કરવા માટે, ટેક્સ્ટ પસંદ કરો, અને પછી હોમ ટેબ પર ફોન્ટ જૂથમાં ફોર્મેટિંગ વિકલ્પોનો ઉપયોગ કરો.
- ટેક્સ્ટ બોક્સને જ ફોર્મેટ કરવા માટે, ફોર્મેટ સંદર્ભ ટેબ પરના આદેશોનો ઉપયોગ કરો, જે તમે ટેક્સ્ટ બોક્સ પસંદ કરો ત્યારે ડ્રોઇંગ ટૂલ્સ હેઠળ દેખાય છે.
- ટેક્સ્ટ બોક્સને સ્થાન આપવા માટે, તેને ક્લિક કરો, અને પછી જ્યારે પોઇન્ટર a બને, ત્યારે ટેક્સ્ટ બોક્સને નવા સ્થાન પર ખેંચો.
- તમે ટેક્સ્ટ બોક્સ અથવા આકારમાંથી બોર્ડર બદલી અથવા દૂર પણ કરી શકો છો.
- જો તમારી પાસે બહુવિધ ટેક્સ્ટ બોક્સ હોય, તમે તેમને એકસાથે લિંક કરી શકો છો જેથી ટેક્સ્ટ એક બોક્સમાંથી બીજા બોક્સમાં વહેશે. ટેક્સ્ટ બોક્સમાંથી એક પર ક્લિક કરો અને ફોર્મેટ ટેબ પર, ડ્રોઇંગ ટૂલ્સ હેઠળ, ટેક્સ્ટ જૂથમાં, લિંક બનાવો ક્લિક કરો.

- 1 તમે કેપ્શન ઉમેરવા માંગો છો તે ચિત્રને ક્લિક કરો.
- 2 સંદર્ભો > કેપ્શન દાખલ કરો પર ક્લિક કરો.



- 3 ડિફોલ્ટ લેબલ (આકૃતિ) નો ઉપયોગ કરવા માટે, કેપ્શન બોક્સમાં તમારું કેપ્શન લખો



ટીપ: તમે નવું લેબલ પર ક્લિક કરીને અને પછી લેબલ બોક્સમાં તમારું કેપ્શન ઉમેરીને તમારું પોતાનું ડિફોલ્ટ કેપ્શન લેબલ પણ બનાવી શકો છો. શબ્દ તમારા માટે નવા લેબલ્સને આપમેળે નંબર આપે છે.

અન્ય તમામ ઓફિસ એપ્લિકેશન્સ

- 1 ચિત્રની નજીક બોક્સ દોરવા માટે Insert > Text Box નો ઉપયોગ કરો. (વધારાની વિગતો માટે ટેક્સ્ટ બોક્સ ઉમેરો જુઓ.)
- 2 ટેક્સ્ટ બોક્સની અંદર ક્લિક કરો અને તમે કેપ્શન માટે ઉપયોગ કરવા માંગો છો તે ટેક્સ્ટ ટાઇપ કરો.
- 3 ટેક્સ્ટ પસંદ કરો. હોમ ટેબ પર, તમે ઇચ્છો તે પ્રમાણે કેપ્શનને સ્ટાઇલ કરવા માટે ફોન્ટ વિકલ્પોનો ઉપયોગ કરો.
- 4 ચિત્ર અને ટેક્સ્ટ બોક્સ પસંદ કરવા માટે Ctrl+ક્લિકનો ઉપયોગ કરો અને પછી ચિત્ર ફોર્મેટ ટેબ પર, જૂથ ગોઠવો, જૂથ > જૂથ પસંદ કરો



ચિત્ર માટે કેપ્શન દાખલ કરો

વર્ડમાં ઇન્સર્ટ કેપ્શન ફીચર દસ્તાવેજમાં ચિત્રોમાં વ્યવસ્થિત રીતે કેપ્શન ઉમેરવાનું સરળ બનાવે છે.

અન્ય Office એપ્લિકેશન્સમાં, જેમ કે પાવરપોઇન્ટ, તમે મેન્યુઅલી ઈમેજની નજીક એક ટેક્સ્ટ બોક્સ ઉમેરો અને પછી ટેક્સ્ટ બોક્સ અને ઈમેજનું જૂથ બનાવો. નીચેની સૂચનાઓ જુઓ. જો તમારી પાસે શ્રેણીમાં બહુવિધ ચિત્રો હોય, તો તમારે તેને મેન્યુઅલી નંબર આપવો પડશે.



© NIMI
NOT TO BE REPUBLISHED

COPA - દસ્તાવેજ સહયોગ મેનેજ કરો

ટિપ્પણીઓમાં ફેરફાર ટ્રેકિંગ અને મેઇલિંગનું સંચાલન (Manage comments change tracking and mailings)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- વર્કમાં સહયોગ કરો
- દસ્તાવેજ સહ-સંપાદિત કરો
- ચાલુ અને બંધ ફેરફારોને ટ્રેક કરો.

વર્કમાં સહયોગ કરો

- 1 શેર પસંદ કરો. રિબન પર. અથવા, ફાઇલ > શેર પસંદ કરો. નોંધ: જો તમારી ફાઇલ પહેલેથી OneDrive પર સાચવેલ નથી, તો તમને તમારી ફાઇલને શેર કરવા માટે OneDrive પર અપલોડ કરવા માટે સંકેત આપવામાં આવશે.
- 2 ડ્રોપ ડાઉનમાંથી તમે કોની સાથે શેર કરવા માંગો છો તે પસંદ કરો અથવા નામ અથવા ઈમેલ એડ્રેસ દાખલ કરો.
- 3 સંદેશ ઉમેરો (વૈકલ્પિક) અને મોકલો પસંદ કરો.

તમારા દસ્તાવેજને હરાવો

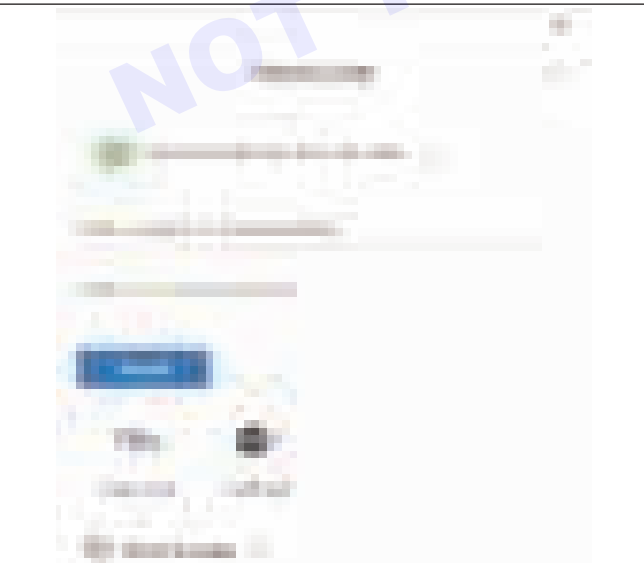
વર્કની અંદરથી ફાઇલ શેર કરવા માટે:

- 1 રિબન પર શેર પસંદ કરો.

અથવા, ફાઇલ > શેર પસંદ કરો

નોંધ: જો તમારી ફાઇલ પહેલેથી OneDrive પર સાચવેલ નથી, તો તમને તમારી ફાઇલને શેર કરવા માટે OneDrive પર અપલોડ કરવા માટે સંકેત આપવામાં આવશે.

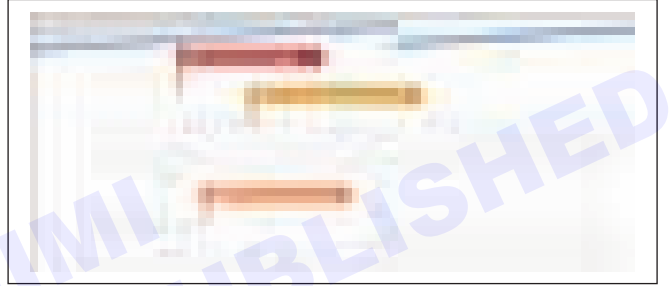
- 2 ડ્રોપ ડાઉનમાંથી તમે કોની સાથે શેર કરવા માંગો છો તે પસંદ કરો અથવા નામ અથવા ઈમેલ એડ્રેસ દાખલ કરો.
- 3 સંદેશ ઉમેરો (વૈકલ્પિક) અને મોકલો પસંદ કરો.



દસ્તાવેજ સહ-સંપાદિત કરો

તમે તમારો દસ્તાવેજ શેર કર્યા પછી, તમે તે જ સમયે અન્ય લોકો સાથે તે ફાઇલ પર કામ કરી શકો છો.

- શ્રેષ્ઠ અનુભવ માટે, વર્ક ફોર વેબમાં સાથે કામ કરો અને રીઅલ-ટાઇમ ફેરફારો જુઓ.
- શેર હેઠળ, તમે અન્ય કોણ પણ ફાઇલને સંપાદિત કરી રહ્યું છે તેના નામ જોશો.
- રંગીન ધ્વજ તમને બતાવે છે કે દરેક વ્યક્તિ ક્યાં છે દસ્તાવેજમાં કામ કરે છે.



ટ્રેક કરો અને ફેરફારોની સમીક્ષા કરો

- 1 ફેરફારો ટ્રેક કરવા માટે, રિવ્યૂ > ટ્રેક ફેરફારો પસંદ કરો.
- 2 ફેરફારોની સમીક્ષા કરવા માટે, ફેરફાર પહેલાં કર્સર મૂકો અને પસંદ કરો:
 - ફેરફાર રાખવા માટે સ્વીકારો, અથવા
 - તેને દૂર કરવાનો અસ્વીકાર કરો.

ટ્રેક ફેરફારો ચાલુ અને બંધ કરો

સમીક્ષા ટેબ પર, ટ્રેક ફેરફારો પસંદ કરો.

- જ્યારે ટ્રેક ફેરફારો ચાલુ હોય, ત્યારે વિભાગ પ્રકાશિત થાય છે. કાઢી નાખવાને સ્ટ્રાઇકથ્રુ સાથે ચિહ્નિત કરવામાં આવે છે, અને ઉમેરાઓને રેખાંકિત સાથે ચિહ્નિત કરવામાં આવે છે. વિવિધ લેખકોના ફેરફારો વિવિધ રંગો સાથે સૂચવવામાં આવે છે.
- જ્યારે ટ્રેક ફેરફારો બંધ હોય, ત્યારે વિભાગ પ્રકાશિત થતો નથી. શબ્દ ફેરફારોને ચિહ્નિત કરવાનું બંધ કરે છે, પરંતુ રંગીન રેખાંકનો અને સ્ટ્રાઇકથ્રુ હજી પણ દસ્તાવેજમાં છે.

ટીપ: તમે સ્ટેટસ બારમાં ટ્રેક ફેરફારો સૂચક પણ ઉમેરી શકો છો. સ્ટેટસ બાર પર જમણું-ક્લિક કરો અને ટ્રેક ફેરફારો પસંદ કરો.

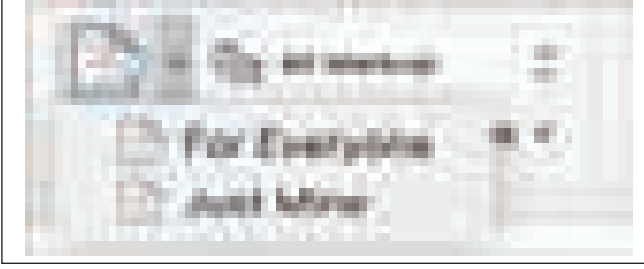
ટ્રેક કરેલા ફેરફારો જુઓ

નોંધ: જ્યારે દસ્તાવેજ સંપાદન અથવા સમીક્ષા મોડમાં હોય, ત્યારે તમે પ્રદર્શિત થતા કાર્ડમાં સહયોગીનો સંપૂર્ણ, સૂચવેલ ફેરફાર જોવા માટે ટ્રેક કરેલ ફેરફાર પસંદ કરી શકો છો. તે ક્રિયા તમારા અંતિમ દસ્તાવેજ સાથે શું કરશે તેનું પૂર્વાવલોકન જોવા માટે સ્વીકારો અથવા નકારો બટન પર હોવર કરો

કોના ફેરફારોને ટ્રેક કરવા તે પસંદ કરો

તમે ફક્ત તમારા પોતાના ફેરફારો અથવા દરેકના ફેરફારોને ટ્રેક કરવાનું પસંદ કરી શકો છો.

- ફક્ત તમારા પોતાના ફેરફારોને ટ્રેક કરવા માટે - સમીક્ષા ટેબ પર, ટ્રેક ફેરફારો > ફક્ત મારું પસંદ કરો.
- દરેકના ફેરફારોને ટ્રેક કરવા - સમીક્ષા ટેબ પર, ટ્રેક ફેરફારો > દરેક માટે પસંદ કરો.



ટીપ: અન્ય લોકોને ટ્રેક ફેરફારો બંધ કરવાથી રોકવા માટે પાસવર્ડનો ઉપયોગ કરવા - સમીક્ષા ટેબ પર, ટ્રેક ફેરફારો > લોક ટ્રેકિંગ પસંદ કરો.

તમે દસ્તાવેજમાં ફેરફારો કેવી રીતે જોવા માંગો છો તે પસંદ કરો તમે

જે માર્કઅપ જોવા માંગો છો તે તમે પસંદ કરી શકો છો.

- સમીક્ષા ટેબ પર, ટ્રેકિંગ પસંદ કરો.
- પ્રદર્શિત કરવા માટે ડ્રોપ-ડાઉન સૂચિ માટે તમામ માર્કઅપ પસંદ કરો.
- નીચેના વિકલ્પોમાંથી એક પસંદ કરો.
 - સરળ માર્કઅપ માર્જિનમાં લાલ લાઇન સાથે ટ્રેક કરેલા ફેરફારો દર્શાવે છે.
 - બધા માર્કઅપ દરેક સમીક્ષક માટે ટેક્સ્ટના વિવિધ રંગો અને રેખાઓ સાથે ટ્રેક કરેલા ફેરફારો દર્શાવે છે
 - સમાવિષ્ટ ફેરફારો સાથે દસ્તાવેજ બતાવવા માટે કોઈ માર્કઅપ માર્કઅપને છુપાવતું નથી
 - ઓરિજિનલ ઓરિજિનલ દસ્તાવેજને ટ્રેક કરેલા ફેરફારો અને ટિપ્પણીઓ દર્શાવ્યા વિના પ્રદર્શિત કરે છે. જો કે, દસ્તાવેજમાં કોઈપણ ટ્રેક કરેલા ફેરફારો અથવા ટિપ્પણીઓ કે જે સ્વીકારવામાં આવ્યાં નથી, નકારી કાઢવામાં આવ્યાં નથી અથવા કાઢી નાખવામાં આવ્યાં નથી તે દસ્તાવેજમાં રહે છે.

જે રીતે ટ્રેક બદલાય છે તે પસંદ કરો

તમે રિવિઝનના પ્રકારો પસંદ કરી શકો છો જે પ્રદર્શિત કરે છે અને તે કેવી રીતે પ્રદર્શિત કરે છે. જો તમે પુનરાવર્તનોને કુગ્ગા તરીકે બતાવવાનું પસંદ કરો છો, તો તેઓ દસ્તાવેજના હાંસિયામાં દર્શાવેલો. જો તમે તેમને સીધા જ દસ્તાવેજમાં લાઇનમાં દર્શાવવાનું પસંદ કરો છો. ઇન લાઇન રિવિઝન અંદરના કુગ્ગાઓને બદલે સ્ટ્રાઈકથ્રુ સાથે તમામ ડિલીટ દર્શાવે છે.

- સમીક્ષા ટેબ પર, ટ્રેક > માર્કઅપ બતાવો પસંદ કરો.
- કુગ્ગા પસંદ કરો અને પછી તમને જોઈતા ડિસ્પ્લેનો પ્રકાર પસંદ કરો.
 - કુગ્ગાઓમાં પુનરાવર્તનો બતાવો
 - બધા પુનરાવર્તનો ઇનલાઇન બતાવો
 - કુગ્ગાઓમાં માત્ર ફોર્મેટિંગ બતાવો

નોંધ: જો તમે કુગ્ગાઓમાં ટ્રેક કરેલા ફેરફારો જોવા માંગતા હો, તો તમારે પ્રિન્ટ લેઆઉટ દૃશ્ય અથવા વેબ લેઆઉટ દૃશ્યમાં હોવું આવશ્યક છે.

ફેરફારના પ્રકાર દ્વારા ફેરફારો દર્શાવો

- સમીક્ષા ટેબ પર, ટ્રેક ફેરફારો > માર્કઅપ બતાવો પસંદ કરો.
- સંપાદનનો પ્રકાર પસંદ કરો.
 - નિવેશ અને કાઢી નાખવું
 - ફોર્મેટિંગ

આઈટમની બાજુનું ચેક માર્ક સૂચવે છે કે તે પસંદ કરેલ છે

નોંધ: જો તમે માર્કઅપ બતાવો મેનૂ પર તેને સાફ કરીને માર્કઅપના પ્રકારને છુપાવો તો પણ, જ્યારે પણ તમે અથવા સમીક્ષક દ્વારા દસ્તાવેજ ખોલવામાં આવે ત્યારે માર્કઅપ આપમેળે પ્રદર્શિત થાય છે.

સમીક્ષકો દ્વારા ફેરફારો દર્શાવો

- સમીક્ષા > ટ્રેકિંગમાં, માર્કઅપ બતાવો પસંદ કરો.
- ચોક્કસ લોકો પસંદ કરો.
- સમીક્ષકોના નામની બાજુના ચેક બોક્સ સિવાયના તમામ ચેક બોક્સને સાફ કરવા માટે પસંદ કરો કે જેમના ફેરફારો તમે બતાવવા માંગો છો

નોંધ: સૂચિમાંના તમામ સમીક્ષકો માટે તમામ ચેક બોક્સ પસંદ કરવા અથવા સાફ કરવા માટે, બધા સમીક્ષકો પસંદ કરો.

ફેરફારો મેનૂ વિભાગમાં ટ્રેક કરેલા ફેરફારો નેવિગેટ કરો

તમે એક ટ્રેક કરેલા ફેરફારમાંથી બીજામાં નેવિગેટ કરી શકો છો.

- સમીક્ષા > ફેરફારોમાં, આગામી ટ્રેક કરેલ ફેરફાર જોવા માટે આગળ પસંદ કરો.
- સમીક્ષા > ફેરફારોમાં, પહેલાનો ટ્રેક કરેલ ફેરફાર જોવા માટે પહેલાનું પસંદ કરો



ટ્રેક કરેલા ફેરફારોને સ્વીકારો અથવા નકારો ટ્રેક

કરેલા ફેરફારોને સંબોધવાની બહુવિધ રીતો છે. જ્યારે તમે ચિહ્નિત ફેરફાર પર ક્લિક કરો છો ત્યારે કાર્ડ ડિસ્પ્લે દ્વારા ટિપ્પણી કરનારને ઓળખવાની અને ટ્રેક કરેલા ફેરફારનો પ્રતિસાદ આપવાની સૌથી વધુ નિપુણ પદ્ધતિ છે. કાર્ડ ટિપ્પણીકર્તાનું નામ અને સૂચન સ્વીકારવા કે નકારવાનો વિકલ્પ દર્શાવે છે.

મુખ્ય મેનૂનો ઉપયોગ કરીને અનુક્રમમાં ટ્રેક કરેલા ફેરફારોને સ્વીકારો અથવા નકારી કાઢો તમે દસ્તાવેજની શરૂઆતથી દસ્તાવેજના અંત સુધી ક્રમમાં ટ્રેક કરેલા ફેરફારને ઉકેલી શકો છો.

- સમીક્ષા ફેરફારો > આગળ પસંદ કરો.

- સ્વીકારો અથવા નામંજૂર કરો પસંદ કરો. આગામી ક્રમિક ટ્રેક કરેલ ફેરફાર તમારી સમીક્ષા અને ક્રિયા માટે હાઇલાઇટ કરવામાં આવે છે

નોંધ: તમે મુખ્ય મેનૂના ડ્રોપ ડાઉન મેનૂનો ઉપયોગ કરીને વ્યક્તિગત ટ્રેક કરેલા ફેરફારોને સ્વીકારી અથવા નકારી પણ શકો છો. આ પદ્ધતિનો ઉપયોગ વારંવાર ટ્રેક કરેલા ફેરફારોની સમીક્ષા કરવા માટે પદ્ધતિસરના અભિગમ તરીકે થાય છે

મુખ્ય મેનૂનો ઉપયોગ કરીને ફેરફારો સ્વીકારો

- સમીક્ષા > ફેરફારોમાં, સ્વીકારો પસંદ કરો.
- વિકલ્પોમાંથી એક પસંદ કરો.
 - સ્વીકારો અને આગળ ખસેડો
 - આ ફેરફાર સ્વીકારો
 - બધા ફેરફારો સ્વીકારો
 - બધા ફેરફારો સ્વીકારો અને ટ્રેકિંગ બંધ કરો

મુખ્ય મેનૂનો ઉપયોગ કરીને ફેરફારોને નકારો

- સમીક્ષા > ફેરફારોમાં, > નામંજૂર કરો પસંદ કરો.
- વિકલ્પોમાંથી એક પસંદ કરો.
 - નકારો અને આગળ ખસેડો
 - આ ફેરફારને નકારી કાઢો
 - બધા ફેરફારોને નકારી કાઢો
 - બધા ફેરફારોને નકારી કાઢો અને ટ્રેકિંગ બંધ કરો

Accept or reject changes within a document with a right click

Working within a document, you can right-click on a tracked change to accept or reject the change. દસ્તાવેજમાં જમણું ક્લિક કરીને ફેરફારો સ્વીકારો અથવા નકારો રાઇટ-ક્લિક પદ્ધતિનો ઉપયોગ કરીને બેનર મેનૂ વિકલ્પ અને ડ્રોપ-ડાઉન મેનૂ વિકલ્પ દર્શાવે છે.

રાઇટ-ક્લિક બેનર વિકલ્પ સાથે વાક્યમાં ફેરફારો સ્વીકારો

- એક સૂચનની સમીક્ષા કરવા માટે દસ્તાવેજમાં ટ્રેક કરેલા ફેરફાર પર જમણું-ક્લિક કરો.
- બેનર પર, નીચેનામાંથી એક પસંદ કરો.
 - સ્વીકારો > સ્વીકારો અને આગળ ખસેડો
 - સ્વીકારો > આ ફેરફાર સ્વીકારો
 - સ્વીકારો > બધા ફેરફારો સ્વીકારો
 - સ્વીકારો > બધા ફેરફારો સ્વીકારો અને ટ્રેકિંગ બંધ કરો
 - સ્વીકારો અને આગળ ખસેડો

રાઇટ-ક્લિક ડ્રોપ-ડાઉન મેનૂ વિકલ્પ સાથે વાક્યમાં ફેરફારો સ્વીકારો

- એક સૂચનની સમીક્ષા કરવા માટે દસ્તાવેજમાં ટ્રેક કરેલા ફેરફાર પર જમણું-ક્લિક કરો.
- ડ્રોપ ડાઉન મેનૂમાં, નીચેનામાંથી એક પસંદ કરો.
 - નિવેશ સ્વીકારો (અથવા કાઢી નાખવું સ્વીકારો)
 - નિવેશને નકારો (અથવા કાઢી નાખવાનો અસ્વીકાર કરો)

મુખ્ય મેનૂ પર એકસાથે બધા ટ્રેક કરેલા ફેરફારો સ્વીકારો અથવા નકારી કાઢો

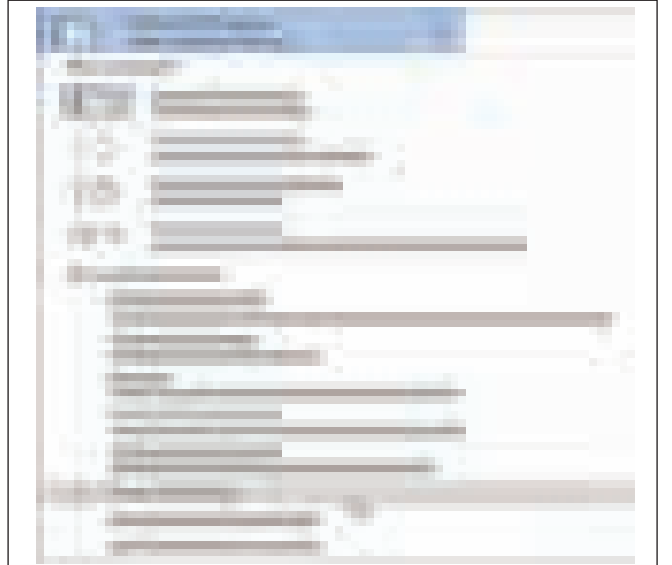
ટ્રેક કરેલા તમામ ફેરફારો સ્વીકારો

- સમીક્ષા > ફેરફારોમાં, સ્વીકારો પસંદ કરો.
- વિકલ્પોમાંથી એક પસંદ કરો
 - દર્શાવેલ તમામ ફેરફારો સ્વીકારો
 - બધા ફેરફારો સ્વીકારો
 - બધા ફેરફારો સ્વીકારો અને ટ્રેક કરવાનું રોકો બધા ટ્રેક કરેલા ફેરફારોને નકારી કાઢો
- સમીક્ષા > ફેરફારોમાં, નામંજૂર કરો પસંદ કરો.
- વિકલ્પોમાંથી એક પસંદ કરો.
 - દર્શાવેલ તમામ ફેરફારોને નકારી કાઢો
 - બધા ફેરફારોને નકારી કાઢો
 - બધા ફેરફારોને નકારી કાઢો અને ટ્રેકિંગ બંધ કરો

નોંધ: જ્યારે સમીક્ષા > ટ્રેકિંગ શો માર્કઅપ > વિશિષ્ટ લોકો > બધા સમીક્ષકો પસંદ કરવામાં આવે છે, ત્યારે બતાવેલ તમામ ફેરફારો સ્વીકારો અને બધા ફેરફારોને નકારો બતાવેલ વિકલ્પો પ્રદર્શિત થતા નથી

ટ્રેક કરેલા ફેરફારો અને ટિપ્પણીઓને છુપાવો જ્યારે પ્રિન્ટિંગ ફેરફારો છુપાવવાથી તે દસ્તાવેજમાંથી દૂર થતા નથી. તમારા દસ્તાવેજમાંથી માર્કઅપને દૂર કરવા માટે તમારે ચેન્જેસ ગ્રૂપમાં સ્વીકારો અને નકારો આદેશોનો ઉપયોગ કરવો આવશ્યક છે.

- ફાઇલ > પ્રિન્ટ > સેટિંગ્સ > બધા પૃષ્ઠો છાપો પર જાઓ.
- દસ્તાવેજ માહિતી હેઠળ, ચેક માર્ક સાફ કરવા માટે પ્રિન્ટ માર્કઅપ પસંદ કરો.



સમીક્ષા ફલક પર સારાંશ સૂચિમાં તમારા બધા ફેરફારો જુઓ

- 1 સમીક્ષા > ટ્રેકિંગમાં, સમીક્ષા ફલક પસંદ કરો.
- 2 તમે તમારા દસ્તાવેજની બાજુમાં અથવા નીચે રિવ્યુઇંગ પેન જોવા માંગતા હોવ તો પસંદ કરો.
 - તમારા દસ્તાવેજની બાજુમાં બધા ફેરફારોની સૂચિ જોવા માટે રિવ્યુઇંગ પેન વર્ટિકલ પસંદ કરો.
 - તમારા દસ્તાવેજની નીચે બધા ફેરફારોની સૂચિ જોવા માટે આડી ફલકની સમીક્ષા કરવાનું પસંદ કરો.

તમારા દસ્તાવેજમાંથી ટ્રેક કરાયેલા બધા ફેરફારો દૂર કરવામાં આવ્યા છે કે નહીં તે તપાસવા માટે તમે એક સરળ સાધન તરીકે સમીક્ષા ફલકનો ઉપયોગ કરી શકો છો જેથી કરીને તેઓ તમારા દસ્તાવેજને જોઈ શકે તેવા

અન્ય લોકોને દેખાય. સમીક્ષા ફલકની ટોચ પરનો સારાંશ વિભાગ તમારા દસ્તાવેજમાં રહેલ દૃશ્યમાન ટ્રેક કરેલા ફેરફારો અને ટિપ્પણીઓની ચોક્કસ સંખ્યા દર્શાવે છે.

સમીક્ષા ફલક તમને લાંબી ટિપ્પણીઓની સમીક્ષા કરવાની પણ મંજૂરી આપે છે જે ટિપ્પણીના બબલમાં બંધબેસતી નથી.

નોંધ: દસ્તાવેજ અથવા ટિપ્પણી બલૂનસથી વિપરીત સમીક્ષા ફલક, તમારા દસ્તાવેજમાં ફેરફાર કરવા માટે શ્રેષ્ઠ સાધન નથી. ટેક્સ્ટ અથવા ટિપ્પણીઓ કાઢી નાખવા અથવા સમીક્ષા ફલકમાં અન્ય ફેરફારો કરવાને બદલે, દસ્તાવેજમાં તમામ સંપાદકીય ફેરફારો કરો. ફેરફારો પછી સમીક્ષા ફલકમાં દેખાશે.

મહત્વપૂર્ણ: ટિપ્પણીઓ હવે ટ્રેક ચેન્જ ફંક્શનનો ભાગ નથી

© NIMI
NOT TO BE REPUBLISHED

મેઈલ મર્જ કરો (Perform mail merge)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- વસ્તુઓ દાખલ કરો
- મેલ મર્જ
- નમૂનાઓ.

ઓબ્જેક્ટ, મેક્રો, મેલ મર્જ, ટેમ્પલેટ્સ અને અન્ય ટૂલ્સ સાથે કામ કરવું

ઓબ્જેક્ટ:

વર્ડ અથવા આઉટલુકના જે વર્ઝનનો તમે ઉપયોગ કરી રહ્યાં છો તેના આધારે, તમે વર્ડ ડોક્યુમેન્ટમાં વિવિધ ઓબ્જેક્ટ્સ (જેમ કે પીડીએફ ફાઈલો, એક્સેલ ચાર્ટ્સ અથવા વર્કશીટ્સ અથવા પાવરપોઈન્ટ પ્રેઝન્ટેશન)ને લિંક કરીને અથવા એમ્બેડ કરીને ઈમેઈલ મેસેજ દાખલ કરી શકો છો. ઓબ્જેક્ટ દાખલ કરવા માટે, Insert ટેબ પર ઓબ્જેક્ટ પર ક્લિક કરો.

તમારા વર્ડ ડોક્યુમેન્ટ અથવા ઈમેલ મેસેજમાં દાખલ કરેલી નવી ફાઈલ બનાવવા માટે:

- 1 ઓબ્જેક્ટ સંવાદ બોક્સમાં, નવું બનાવો ટેબ પર ક્લિક કરો, અને પછી ઓબ્જેક્ટ પ્રકાર સૂચિમાંથી એક વિકલ્પ પસંદ કરો.

તે ફાઈલ પ્રકાર માટે ડિફોલ્ટ પ્રોગ્રામ ખુલે છે, જ્યાં તમે ઈચ્છો તે કોઈપણ ટેકસ્ટ અથવા ડેટા દાખલ કરી શકો છો. જ્યારે તમે પ્રોગ્રામ બંધ કરો છો, ત્યારે કોઈપણ ઉમેરાયેલ સામગ્રી અથવા ફેરફારો તમારા વર્ડ દસ્તાવેજ અથવા ઈમેઈલ સંદેશમાં દેખાય છે.

- 2 જો તમે ઈચ્છો છો કે નવી ફાઈલ તમારી નવી ફાઈલના પ્રથમ પૃષ્ઠને બદલે ક્લિક કરી શકાય તેવા આયકન તરીકે દેખાય, તો ચિહ્ન તરીકે દર્શાવો પસંદ કરો. જો આ ચેક બોક્સ પસંદ કરેલ હોય, તો તમે બદલો આયકન પર ક્લિક કરીને અલગ ચિહ્ન પસંદ કરી શકો છો.

હાલની ફાઈલને લિંક કરો અથવા એમ્બેડ કરો

પહેલેથી જ બનાવેલ ઓબ્જેક્ટને લિંક કરવા અથવા એમ્બેડ કરવા માટે:

- 1 ઓબ્જેક્ટ સંવાદ બોક્સમાં, ફાઈલમાંથી બનાવો ટેબ પસંદ કરો, અને પછી તમે દાખલ કરવા માંગો છો તે ફાઈલ શોધવા માટે બ્રાઉઝ કરો ક્લિક કરો.
- 2 સ્ત્રોત ફાઈલ સાથે લિંક કરવા માટે, તેને તમારા વર્ડ ડોક્યુમેન્ટ અથવા ઈમેલ મેસેજમાં એમ્બેડ કરવાને બદલે, ફાઈલની લિંક પસંદ કરો.
- 3 જો તમે ફાઈલના પ્રથમ પૃષ્ઠને બદલે, દાખલ કરેલ ફાઈલને ક્લિક કરી શકાય તેવા આયકન તરીકે દેખાવા માંગતા હો, તો ચિહ્ન તરીકે દર્શાવો પસંદ કરો. જો આ ચેક બોક્સ પસંદ કરેલ હોય, તો તમે બદલો આયકન પર ક્લિક કરીને અલગ ચિહ્ન પસંદ કરી શકો છો.

મેઈલ મર્જ:

શું તમારી પાસે નામો અને સરનામાંઓની લાંબી સૂચિ છે કે જેના પર તમારે પત્ર મોકલવાની જરૂર છે? મેઈલ મર્જ પ્રક્રિયા વ્યક્તિગત લાગે તેવા અક્ષરો ઝડપથી બનાવવા માટે ડેટા સ્ત્રોત સાથે વર્ડ દસ્તાવેજને જોડે છે.

મેઈલ મર્જ વ્યાખ્યાઓ

પ્રારંભિક દસ્તાવેજ: (મુખ્ય દસ્તાવેજ) એક દસ્તાવેજ જેમાં માહિતી હોય છે જે દરેક મર્જ કરેલા દસ્તાવેજ માટે સમાન હોય છે. પ્રારંભિક દસ્તાવેજમાં

વેરિએબલ માહિતી માટે ફીલ્ડ નામો શામેલ છે, જેમ કે નામ અને સરનામાં જે દાખલ કરવામાં આવશે.

ડેટા સ્ત્રોત અથવા પ્રાપ્તકર્તાઓની સૂચિ: એક ફાઈલ જેમાં મેઈલ મર્જ દરમિયાન મુખ્ય દસ્તાવેજમાં દાખલ કરવાની માહિતી શામેલ છે. ઉદાહરણ તરીકે, તેમાં એવા લોકોના નામ અને સરનામાં ધરાવતા રેકૉર્ડ્સ છે કે જેને મેઈલ મર્જ લેટર મોકલવામાં આવે છે. એક્સેલ સ્પ્રેડશીટ્સ, એક્સેસ ડેટાબેસેસ અથવા વર્ડ ડોક્યુમેન્ટ કોષ્ટકો ડેટા સ્ત્રોતોના સારા ઉદાહરણો છે.

ફીલ્ડ: માહિતી કેટેગરી કે જે ચોક્કસ માહિતીનો સંગ્રહ કરે છે. ઉદાહરણ તરીકે, “છેલ્લું નામ” ફીલ્ડમાં ફક્ત લોકોના છેલ્લા નામ હશે.

રેકૉર્ડ: રેકૉર્ડ એ ડેટા ફીલ્ડનો સંપૂર્ણ સેટ છે જે એક વસ્તુ અથવા વ્યક્તિ સાથે સંબંધિત છે. ઉદાહરણ તરીકે, એક રેકૉર્ડમાં વ્યક્તિનું નામ અને છેલ્લું નામ, સરનામું, ફોન નંબર અને જન્મ તારીખનો સમાવેશ થાય છે.

મર્જ ફીલ્ડ: એક મર્જ ફીલ્ડ એ છે જ્યાં તમે ડેટા સ્ત્રોતમાંથી માહિતીને મુખ્ય દસ્તાવેજમાં દાખલ કરવા માંગો છો. મર્જ ફીલ્ડ્સ તેમની આસપાસ શેવરોન (“”) સાથે દેખાય છે. એક ઉદાહરણ હશે: પ્રિય “પ્રથમ નામ”.

સરનામું બ્લોક: મર્જ ફીલ્ડ્સનું જૂથ જે મેઈલ મર્જ દસ્તાવેજમાં સરનામું બનાવે છે. ઉદાહરણ તરીકે, સિંગલ સરનામું નામથી બનેલું છે, શેરી સરનામું, શહેર, રાજ્ય અને પિન કોડ. વર્ડ આપોઆપ તમામ યોગ્ય એડ્રેસ ફીલ્ડ એકસાથે દાખલ કરી શકે છે, તેથી તમારે પાંચ કે છ મર્જ ફીલ્ડ જાતે દાખલ કરવાની જરૂર નથી.

ગ્રીટિંગ લાઈન: મર્જ ફીલ્ડ્સનું એક જૂથ જે મેઈલ મર્જ ડોક્યુમેન્ટની શુભેચ્છા લાઈન બનાવે છે, જેમ કે “ડિયર મિ. મેકડોનાલ્ડ”. વર્ડ આપમેળે તમામ યોગ્ય ગ્રીટિંગ ટેકસ્ટ, શીર્ષક અને નામ ફીલ્ડ્સ એકસાથે દાખલ કરી શકે છે, તેથી તમારે ટેકસ્ટ અને જરૂરી મર્જ ફીલ્ડ્સ જાતે દાખલ કરવાની જરૂર નથી.

હેડર રો: ડેટા સ્ત્રોત માહિતી કોષ્ટકમાં સંગ્રહિત થાય છે. કોષ્ટકની પ્રથમ પંક્તિ હેડર પંક્તિ છે અને તેમાં ડેટા સ્ત્રોત માટે ફીલ્ડ નામો છે. ઉદાહરણ તરીકે, FirstName, LastName અને Address એ હેડર પંક્તિઓ છે.

સેટ કરો અને દસ્તાવેજનો પ્રકાર પસંદ કરો

મેઈલ મર્જ પ્રક્રિયા શરૂ કરવા માટે, તમારે પહેલાં તમે કયા પ્રકારના દસ્તાવેજ બનાવવા માંગો છો તે પસંદ કરવાની જરૂર છે.

- 1 મેઈલિંગ્સ ટેબ પર ક્લિક કરો.
- 2 સ્ટાર્ટ મેઈલ મર્જ બટનને ક્લિક કરો.
- 3 સ્ટેપ-બાય સ્ટેપ મેઈલ મર્જ વિઝાર્ડ પસંદ કરો.

મેઈલ મર્જ પેન જમણી બાજુએ દેખાય છે, જે તમને મેઈલ મર્જ દ્વારા લઈ જવા માટે તૈયાર છે.

4 બનાવવા માટે દસ્તાવેજનો પ્રકાર પસંદ કરો.

5 આગળ ક્લિક કરો: દસ્તાવેજ શરૂ કરો

મેઇલ મર્જ વિઝાર્ડ આગલા પગલા પર આગળ વધે છે.

Select a Document

એક દસ્તાવેજ પસંદ કરો આ આગલું પગલું પ્રારંભિક દસ્તાવેજ પસંદ કરવાનું છે.

1 પ્રારંભિક દસ્તાવેજ પસંદ કરો. તમે વર્તમાન દસ્તાવેજનો ઉપયોગ મેઇલ મર્જના આધાર તરીકે કરી શકો છો, અથવા તમે તેના બદલે ટેમ્પલેટ અથવા અસ્તિત્વમાં છે તે દસ્તાવેજ પસંદ કરી શકો છો.

2 આગળ ક્લિક કરો: પ્રાપ્તકર્તાઓને પસંદ કરો.

મેઇલ મર્જ વિઝાર્ડ પગલું 3 પર આગળ વધે છે.

પ્રાપ્તકર્તાઓ પસંદ કરો,

હવે તમારે તમારા સરનામાંઓની સૂચિ ક્યાંથી મળશે તે પસંદ કરવાની જરૂર પડશે. આ ઉદાહરણ ડેટાબેઝમાંથી હાલની સૂચિનો ઉપયોગ કરે છે, પરંતુ તમે Outlook સંપર્કો પણ પસંદ કરી શકો છો અથવા તમારી પોતાની સૂચિ જાતે બનાવી શકો છો.

1 હાલની યાદીનો ઉપયોગ કરો પસંદ કરો.

2 બ્રાઉઝ પર ક્લિક કરો.

3 તમારી ડેટા ફાઇલ પસંદ કરો.

મેઇલ મર્જ પ્રાપ્તકર્તાઓની ડેટા ફાઇલ ડેટાબેઝ ફાઇલ, એક્સેલ સ્પ્રેડશીટ, અન્ય વર્ડ દસ્તાવેજ અથવા અન્ય પ્રકારની ડેટા ફાઇલોમાં હોઈ શકે છે.

4 ખોલો ક્લિક કરો.

મેઇલ મર્જ પ્રાપ્તકર્તા સંવાદ બોક્સ એ સરનામાંઓ દર્શાવે છે જેનો ઉપયોગ કરવામાં આવશે. જો તમે ડેટા સ્ત્રોત તરીકે એક્સેલ સ્પ્રેડશીટનો ઉપયોગ કરી રહ્યાં છો, તો તમને એક વર્કશીટ પસંદ કરવા માટે પણ સંકેત આપવામાં આવશે જેમાં સરનામાં

જો ત્યાં કોઈ સરનામું છે જેનો તમે ઉપયોગ કરવા માંગતા નથી, તો તમે તેને અનચેક કરી શકો છો.

5 ખાતરી કરો કે યોગ્ય પ્રાપ્તકર્તાઓ પસંદ કરેલ છે અને બરાબર ક્લિક કરો.

6 આગળ ક્લિક કરો: તમારો પત્ર લખો.

મેઇલ મર્જ વિઝાર્ડ પગલું 4 પર આગળ વધે છે.

તમારો પત્ર લખો

મુખ્ય દસ્તાવેજ સેટ થયા પછી અને પ્રાપ્તકર્તાની સૂચિ જોડાયેલ અને સંપાદિત થયા પછી, તમે દસ્તાવેજમાં મર્જ ફીલ્ડ્સ દાખલ કરવા માટે તૈયાર છો. મર્જ ક્ષેત્રો પ્રાપ્તકર્તાઓની સૂચિમાંથી અનન્ય માહિતી માટે દસ્તાવેજમાં પ્લેસહોલ્ડર્સ છે.

જ્યારે તમે મુખ્ય દસ્તાવેજમાં મર્જ ફીલ્ડ મૂકો છો, ત્યારે તે ફીલ્ડમાંથી માહિતી તે દસ્તાવેજ માટે દેખાશે જે તે પ્રાપ્તકર્તા માટે અનન્ય છે.

1 તમને જ્યાં માહિતી જોઈતી હોય ત્યાં ક્લિક કરો.

2 પ્લેસહોલ્ડર વિકલ્પોમાંથી એક પસંદ કરો.

તમે વિઝાર્ડમાંથી અથવા રિબન પર ફીલ્ડ્સ લખો અને દાખલ કરો જૂથમાંથી મર્જ ફીલ્ડ્સ ઉમેરી શકો છો:

• **સરનામું બ્લોક:** પ્રાપ્તકર્તાઓના નામ અને સરનામાં દાખલ કરવા માટે આ ક્ષેત્રોનું સંયોજન છે.

• **ગ્રીટિંગ લાઇન:** આ ગ્રીટિંગ લાઇનમાં પ્રાપ્તકર્તાનું નામ દાખલ કરવા માટે ફીલ્ડ્સનું સંયોજન છે

• **ઇન્સર્ટ મર્જ ફીલ્ડ:** જ્યારે તમે આ બટન પર ક્લિક કરો છો, ત્યારે વધારાના મર્જ ફીલ્ડ્સની યાદી દેખાય છે જે તમે દાખલ કરી શકો છો.

પ્લેસહોલ્ડરને કસ્ટમાઇઝ કરો.

OK પર ક્લિક કરો.

(વૈકલ્પિક) તમે શામેલ કરવા માંગો છો તે દરેક મર્જ ફીલ્ડ ઉમેરવા માટે પગલાંઓનું પુનરાવર્તન કરો.

આગળ ક્લિક કરો: તમારા અક્ષરોનું પૂર્વાવલોકન કરો.

મેઇલ મર્જ વિઝાર્ડ પગલું 5 પર આગળ વધે છે.

તમારા મેઇલ મર્જનું પૂર્વાવલોકન કરો

કેટલીકવાર, ફક્ત મર્જ ફીલ્ડના નામો જોવાને બદલે, એકવાર દસ્તાવેજમાં દાખલ કરવામાં આવે તે પછી ડેટા કેવો દેખાશે તે જોવામાં મદદરૂપ થાય છે.

મેઇલ મર્જ પૂર્ણ કરતા પહેલા મેઇલ મર્જ કેવી રીતે દેખાશે તેનું તમે સરળતાથી પૂર્વાવલોકન કરી શકો છો. તમે ઇચ્છો તે પ્રમાણે પરિણામો દેખાચ તેની ખાતરી કરવા માટે આ પ્રોત્સાહિત કરવામાં આવે છે.

1 દરેક મર્જ કરેલા દસ્તાવેજનું પૂર્વાવલોકન કરવા માટે મેઇલ મર્જ ફલકમાં એરો બટનોનો ઉપયોગ કરો.

તમે રિબન પરના પૂર્વાવલોકન પરિણામો જૂથમાં તીર બટનોનો પણ ઉપયોગ કરી શકો છો.

ચોક્કસ પ્રાપ્તકર્તાને શોધવા માટે પૂર્વાવલોકન પરિણામો જૂથમાં અથવા મેઇલ મર્જ ફલકમાં પ્રાપ્તકર્તા શોધો બટનને ક્લિક કરો.

2 આગળ ક્લિક કરો: મર્જ પૂર્ણ કરો.

મેઇલ મર્જ વિઝાર્ડ અંતિમ પગલા પર આગળ વધે છે.

પૂર્ણ મર્જ

એકવાર તમે પ્રાપ્તકર્તાઓની સૂચિ ઉમેરી લો અને મર્જ ફીલ્ડ્સ સાથે દસ્તાવેજ ભરી લો, પછી દરેક પ્રાપ્તકર્તા માટે દસ્તાવેજનું અલગ સંસ્કરણ બનાવીને મર્જ કરવાનું છેલ્લું પગલું છે.

તમે મેઇલ મર્જને સમાપ્ત કરી શકો છો તે કેટલીક અલગ અલગ રીતો છે:

• વ્યક્તિગત દસ્તાવેજો સંપાદિત કરો: મેઇલ મર્જના પરિણામોને નવા દસ્તાવેજમાં મૂકો. તમે મેઇલ મર્જના પરિણામોને સંપાદિત કરવા અને અન્ય દસ્તાવેજોની જેમ સાચવવા અને છાપવા માટે મુક્ત છો.

• દસ્તાવેજો છાપો: રેકૉર્ડને મર્જ કરે છે અને તેમને સીધા જ પ્રિન્ટરને મોકલે છે.

1 મેઇલ મર્જ સમાપ્ત કરવા માટે તમે ઉપયોગ કરવા માંગો છો તે વિકલ્પ પસંદ કરો.

તમે રિબન પરના ફિનિશ અને મર્જ બટનને પણ ક્લિક કરી શકો છો અને ત્યાં મર્જ વિકલ્પ પસંદ કરી શકો છો.

તમને કયા રેકૉર્ડ્સ મર્જ કરવા તે પસંદ કરવાનો વિકલ્પ પણ આપવામાં આવ્યો છે. તમે સૂચિમાંના તમામ રેકૉર્ડ્સને મર્જ કરી શકો છો, ફક્ત હાલમાં પ્રદર્શિત રેકૉર્ડ, અથવા શ્રેણીનો ઉલ્લેખ કરી શકો છો.

2 તમે મર્જ કરવા માંગો છો તે રેકોર્ડ્સ પસંદ કરો.

3 બરાબર ક્લિક કરો.

વર્ડ મુખ્ય દસ્તાવેજ અને ડેટા સ્ત્રોતમાંથી માહિતીને નવા વર્ડ દસ્તાવેજમાં મર્જ કરે છે, અથવા તમે પસંદ કરેલ વિકલ્પના આધારે તેને મર્જ કરે છે અને પ્રિન્ટરને મોકલે છે.

Templates

વર્ડ ડોક્યુમેન્ટમાં ટેમ્પલેટ કેવી રીતે બનાવવું

ટેમ્પલેટ એ પહેલાથી બનાવેલ ડોક્યુમેન્ટ છે જેમાં મેક્રો, હેડર અને ફૂટર્સ, ફસ્ટમ ડિક્શનરી, લેઆઉટ, ઈમેજીસ અને ઓટોટેક્સ્ટ એન્ટ્રીઝ જેવા પહેલાથી બનાવેલા ચોક્કસ ફોર્મેટિંગ વિકલ્પોનો સમાવેશ થાય છે.

સમાન સામગ્રી અને બંધારણ સાથે દસ્તાવેજ બનાવતી વખતે ટેમ્પલેટ તમને સમય બચાવવામાં મદદ કરે છે.

વર્ડ ડોક્યુમેન્ટમાં ટેમ્પલેટ બનાવો

વર્ડ ડોક્યુમેન્ટમાં ટેમ્પલેટ બનાવવા માટે નીચે જણાવેલ સૌથી સરળ સ્ટેપ્સને અનુસરો -

પગલું 1: વર્ડ દસ્તાવેજ ખોલો.

પગલું 2: રિબન પર ફાઇલ ટેબ પર જાઓ અને નવા વિકલ્પ પર ક્લિક કરો.

પગલું 3: ખાલી દસ્તાવેજ પર ક્લિક કરો.

પગલું 4: તમે ટેમ્પલેટ તરીકે આગળ ઉપયોગ કરો છો તે ઈચ્છનીય સામગ્રી ઉમેરો.

માઈક્રોસોફ્ટ વર્ડ ટેમ્પલેટ સાચવો

એકવાર તમે Microsoft Word માં ટેમ્પલેટ બનાવી લો, પછી તમે તેને તમારા આગળના ઉપયોગ માટે સાચવી શકો છો.

માઈક્રોસોફ્ટ વર્ડમાં, તમે માઈક્રોસોફ્ટ વર્ડ ટેમ્પલેટને .dotx, .dot અથવા .dotm ફાઇલ ફોર્મેટ તરીકે સાચવી શકો છો.

માઈક્રોસોફ્ટ વર્ડમાં ટેમ્પલેટ સાચવવા માટે, નીચેની સૂચનાને અનુસરો -

1 વર્ડ ડોક્યુમેન્ટ ખોલો જેમાં તમે બનાવો છો નમૂનો

2 રિબન પર ફાઇલ ટેબ પર જાઓ. સ્ક્રીન પર વિકલ્પોની યાદી દેખાશે જેમાં Save As પર ક્લિક કરો.

3 જ્યાં તમે ફાઇલ સાચવવા માંગો છો તે સ્થાન બ્રાઉઝ કરો. Save as type સાથે ડ્રોપ ડાઉન એસોસિયેટ પર ક્લિક કરો. વર્ડ ટેમ્પલેટ (*.dotx) પસંદ કરો અને સ્ક્રીનના તળિયે સેવ બટન પર ક્લિક કરો.

હવે, તમે જોઈ શકો છો કે તમારું વર્ડ ટેમ્પલેટ વર્ડ ડોક્યુમેન્ટમાં સેવ થયેલ છે.

વર્ડ ટેમ્પલેટને એડિટ કરો

એકવાર તમે Microsoft Word માં ટેમ્પલેટ બનાવી લો, પછી તમે તેને તમારી જરૂરિયાતના આધારે એડિટ પણ કરી શકો છો અને પછી તેને આગામી, સમાન વર્ડ ડોક્યુમેન્ટ્સ માટે સાચવી શકો છો.

વર્ડ ટેમ્પલેટને સંપાદિત કરવા માટે નીચેના સૌથી સરળ પગલાં છે -

સ્ટેપ 1: વર્ડ ડોક્યુમેન્ટના ઉપરના ડાબા ખૂણે ફાઇલ ટેબ પર જાઓ અને ઓપન બટન પર ક્લિક કરો.

સ્ટેપ 2: સ્ક્રીન પર ઓપન ડાયલોગ બોક્સ દેખાશે. તે સ્થાન બ્રાઉઝ કરો જ્યાં તમે અગાઉ બનાવેલ ટેમ્પલેટ સાચવો છો અને સ્ક્રીનના તળિયે ઓપન બટન પર ક્લિક કરો.

પગલું 3: તમે તમારા નમૂનામાં ઉમેરવા માંગો છો તે ફેરફારો કરો.

પગલું 4: એકવાર તમે તમારા નમૂનામાં તમારા બધા ઈચ્છિત ફેરફારો કર્યા પછી, તમે તમારા નમૂનાને સાચવી શકો છો અને તેને બંધ કરી શકો છો.

માઈક્રોસોફ્ટ વર્ડ ટેમ્પલેટનો ઉપયોગ કરો

માઈક્રોસોફ્ટ વર્ડ બિલ્ટ વર્ડ ટેમ્પલેટ્સમાં પણ પ્રદાન કરે છે જેનો ઉપયોગ તમે તમારા વર્ડ ડોક્યુમેન્ટ પર કરી શકો છો.

માઈક્રોસોફ્ટ વર્ડ ટેમ્પલેટનો ઉપયોગ કરવા માટે નીચેના પગલાંઓ છે -

પગલું 1: નવો વર્ડ દસ્તાવેજ ખોલો.

પગલું 2: ફાઇલ પર જાઓ રિબન પર ટેબ. સ્ક્રીન પર ફાઇલ મેનુ વિકલ્પો દેખાશે. નવા પર ક્લિક કરો.

પગલું 3: એક ઉપલબ્ધ નમૂનાઓ વિન્ડો દેખાય છે. પર્સનલ પર ક્લિક કરો.

પગલું 4: વર્ડ ટેમ્પલેટ્સની સૂચિ સ્ક્રીન પર દેખાય છે.

તમે જે નમૂનાને લાગુ કરવા માંગો છો તેના પર પસંદ કરો અને ડબલ ક્લિક કરો.

હવે, તમે જોઈ શકો છો કે પસંદ કરેલ ટેમ્પલેટ વર્ડ ડોક્યુમેન્ટમાં ઉમેરવામાં આવ્યું છે.

MS Excel માં ફાઈલો ખોલો (Open files in MS Excel)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- એક્સેલનો પરિચય
- સેલ સંદર્ભ અને લિંકિંગ શીટ
- વર્કશીટ પર પ્રિન્ટ એરિયા સેટ કરો અથવા સાફ કરો.

તે માઈક્રોસોફ્ટ દ્વારા વિકસિત સ્પ્રેડશીટ પ્રોગ્રામ છે. એક્સેલ ડેટાને કોલમ અને પંક્તિઓમાં ગોઠવે છે અને તમને ગાણિતિક કાર્યો કરવા દે છે. તે Windows, macOS, Android અને iOS પર ચાલે છે.

પ્રથમ સંસ્કરણ 1985 માં બહાર પાડવામાં આવ્યું હતું અને તે વર્ષોથી ઘણા ફેરફારોમાંથી પસાર થયું છે. જો કે, મુખ્ય કાર્યક્ષમતા મોટે ભાગે સમાન રહે છે.

એક્સેલનો ઉપયોગ સામાન્ય રીતે આ માટે થાય છે:

- વિશ્લેષણ
- ડેટા એન્ટ્રી
- ડેટા મેનેજમેન્ટ
- એકાઉન્ટિંગ
- બજેટિંગ
- ડેટા વિશ્લેષણ
- વિઝ્યુઅલ અને આલેખ
- પ્રોગ્રામિંગ
- નાણાકીય મોડેલિંગ
- અને ઘણું બધું!

એક્સેલ શા માટે વાપરો?

- તે વિશ્વનો સૌથી લોકપ્રિય સ્પ્રેડશીટ પ્રોગ્રામ છે
- તે શીખવું અને પ્રારંભ કરવું સરળ છે.
- કૌશલ્યની ટોચમર્યાદા ઊંચી છે, જેનો અર્થ છે કે તમે જેમ જેમ વધુ સારા બનશો તેમ તમે વધુ અદ્યતન વસ્તુઓ કરી શકો છો
- તેનો ઉપયોગ કામ અને રોજિંદા જીવનમાં બંનેમાં થઈ શકે છે, જેમ કે કુટુંબનું બજેટ બનાવવા માટે
- તેને વિશાળ સમુદાય સમર્થન છે
- તે સતત Microsoft દ્વારા આધારભૂત છે
- ટેમ્પલેટ્સ અને ફ્રેમવર્કનો તમારા અને અન્ય લોકો દ્વારા પુનઃઉપયોગ કરી શકાય છે, સર્જન ખર્ચ ઘટાડીને બનાવો

બીજી વર્કશીટનો કોષ સંદર્ભ: કોષ પર ક્લિક કરો જેમાં તમે ફોર્મ્યુલા દાખલ કરવા માંગો છો. , પ્રકાર = (સમાન ચિહ્ન) અને તમે ઉપયોગ કરવા માંગો છો તે સૂત્ર. સંદર્ભ લેવા માટે વર્કશીટ માટે ટેબ પર ક્લિક કરો. સંદર્ભ આપવા માટે કોષ અથવા કોષોની શ્રેણી પસંદ કરો.

- 1 કોષ પર ક્લિક કરો જેમાં તમે ફોર્મ્યુલા દાખલ કરવા માંગો છો.
- 2 ફોર્મ્યુલા બારમાં, ટાઈપ કરો = (સમાન ચિહ્ન).
- 3 નીચેનામાંથી એક કરો:
 - એક અથવા વધુ કોષોનો સંદર્ભ આપો: સંદર્ભ બનાવવા માટે, સમાન કાર્યપત્રક પર કોષ અથવા કોષોની શ્રેણી પસંદ કરો.
 - તમે પસંદગીને ખસેડવા માટે કોષ પસંદગીની સરહદને ખેંચી શકો છો અથવા પસંદગીને વિસ્તૃત કરવા માટે સરહદના ખૂણાને ખેંચી શકો છો.
 - નિર્ધારિત નામનો સંદર્ભ આપો: નિર્ધારિત નામનો સંદર્ભ બનાવવા માટે, નીચેનામાંથી એક કરો:
 - નામ લખો

- F3 દબાવો, પેસ્ટ નામ બોક્સમાં નામ પસંદ કરો અને પછી બરાબર ક્લિક કરો.

નોંધ: જો રંગ કોડેડ બોર્ડર પર કોઈ ચોરસ ખૂણો નથી, તો સંદર્ભ નામવાળી શ્રેણીનો છે

4 નીચેનામાંથી એક કરો:

- જો તમે એક કોષમાં સંદર્ભ બનાવી રહ્યા છો, તો Enter દબાવો.
- જો તમે અરે ફોર્મ્યુલા (જેમ કે A1:G4) માં સંદર્ભ બનાવી રહ્યા હોવ, તો Ctrl+Shift+Enter દબાવો.

સંદર્ભ એક કોષ અથવા કોષોની શ્રેણી હોઈ શકે છે, અને અરે ફોર્મ્યુલા એક હોઈ શકે છે જે એક અથવા બહુવિધ પરિણામોની ગણતરી કરે છે.

વર્કશીટ પર પ્રિન્ટ એરિયા સેટ કરો અથવા સાફ કરો: જો તમે વર્કશીટ પર ચોક્કસ પસંદગીને વારંવાર પ્રિન્ટ કરો છો, તો તમે પ્રિન્ટ એરિયાને વ્યાખ્યાયિત કરી શકો છો જેમાં માત્ર તે જ પસંદગીનો સમાવેશ થાય છે. પ્રિન્ટ એરિયા એ કોષોની એક અથવા વધુ શ્રેણીઓ છે જે તમે છાપવા માટે નિયુક્ત કરો છો જ્યારે તમે આખી વર્કશીટ છાપવા માંગતા નથી. જ્યારે તમે પ્રિન્ટ વિસ્તાર વ્યાખ્યાયિત કર્યા પછી વર્કશીટ છાપો છો, ત્યારે માત્ર પ્રિન્ટ વિસ્તાર જ પ્રિન્ટ થાય છે. તમે જરૂર મુજબ પ્રિન્ટ વિસ્તારને વિસ્તૃત કરવા માટે કોષો ઉમેરી શકો છો, અને તમે સમગ્ર કાર્યપત્રકને છાપવા માટે પ્રિન્ટ વિસ્તાર સાફ કરી શકો છો.

વર્કશીટમાં બહુવિધ પ્રિન્ટ વિસ્તારો હોઈ શકે છે. દરેક પ્રિન્ટ એરિયા અલગ પેજ તરીકે પ્રિન્ટ કરશે.

એક અથવા વધુ પ્રિન્ટ વિસ્તારો સેટ કરો

- 1 વર્કશીટ પર, કોષો પસંદ કરો કે જેને તમે તરીકે વ્યાખ્યાયિત કરવા માંગો છો પ્રિન્ટ વિસ્તાર.

ટીપ: બહુવિધ પ્રિન્ટ વિસ્તારો સેટ કરવા માટે, Ctrl કી દબાવી રાખો અને તમે છાપવા માંગો છો તે વિસ્તારોને ક્લિક કરો. દરેક પ્રિન્ટ એરિયા તેના પોતાના પેજ પર પ્રિન્ટ કરે છે.

- 2 પેજ લેઆઉટ ટેબ પર, પેજ સેટઅપ ગ્રુપમાં, પ્રિન્ટ એરિયા પર ક્લિક કરો અને પછી પ્રિન્ટ એરિયા સેટ કરો ક્લિક કરો

નોંધ: જ્યારે તમે વર્કબુક સાચવો છો ત્યારે તમે સેટ કરેલ પ્રિન્ટ એરિયા સાચવવામાં આવે છે.



ડેટાની હેરફેર કરો (Manipulate data)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ડેટાની હેરફેર કરો
- કોષ શ્રેણી.

ડેટા વિશ્લેષણ એ એક પડકારજનક કાર્ય છે, ખાસ કરીને જો તમારી પાસે ડેટા મેનીપ્યુલેશન કુશળતા ન હોય. આ લેખમાં, અમે કેટલીક સૌથી સામાન્ય ડેટા મેનીપ્યુલેશન તકનીકોની ચર્ચા કરીશું જેનો ઉપયોગ ડેટા વિશ્લેષણ માટે Excel માં થઈ શકે છે.

આ તકનીકોની શક્તિ કેટલાક વાસ્તવિક જીવન ઉદાહરણોનો ઉપયોગ કરીને દર્શાવવામાં આવશે. ચર્ચા કરેલ 9 સામાન્ય ડેટા મેનીપ્યુલેશન તકનીકો છે:

- 1 ફિલ્ટરિંગ
- 2 વર્ગીકરણ
- 3 જૂથીકરણ
- 4 પિવોટિંગ
- 5 ટ્રાન્સપોઝિંગ
- 6 ડેટા પ્રકારો બદલવો
- 7 કોલમ અને પંક્તિઓ ઉમેરવા
- 8 કોલમ અથવા પંક્તિઓનું નામકરણ
- 9 કોલમ અથવા પંક્તિઓ દાખલ કરવી.

આમાંની દરેક તકનીક તમને તમારા ડેટા અને તે કેવી રીતે કાર્ય કરે છે તેની વધુ સારી સમજ પ્રદાન કરશે - વિવિધ પ્રકારના વિઝ્યુઅલાઇઝેશનની આસપાસ તમારું માથું મેળવવાથી માંડીને બહારના લોકોની શોધખોળ સુધી. આ સરળ યુક્તિઓ માત્ર તમારી કાર્યક્ષમતામાં જ સુધારો કરશે નહીં, પરંતુ જે લોકો એક્સેલને જાણતા નથી તેમના માટે તમે શું કરી રહ્યા છો તે સમજવાનું પણ સરળ બનાવશે.

Changing Data Types

એકવાર તમે આ આદેશ પસંદ કરી લો, પછી “ડેટા પ્રકાર પસંદ કરો” વિન્ડો દેખાશે. ત્રણ જનરલ છે ડેટાની શ્રેણીઓ: ફક્ત ટેક્સ્ટ ડેટા;

માત્ર આંકડાકીય માહિતી; અને મિશ્રિત ડેટા (ટેક્સ્ટ અને ન્યુમેરિક).

તમે આ દરેક શ્રેણીઓમાંથી એક અથવા વધુ વિકલ્પો પસંદ કરીને તમારી પસંદગીને વધુ શુદ્ધ કરી શકો છો: ટેક્સ્ટ; સંખ્યાત્મક; તારીખ સમય; તાર્કિક; અને ઑબ્જેક્ટ લિફ્ટિંગ અને એમ્બેડિંગ

કોલમ અને પંક્તિઓ ઉમેરવી: તમારા ડેટામાં કોલમ અથવા પંક્તિઓ ઉમેરવા એ તમારા કાર્યને વધુ કાર્યક્ષમ બનાવવાનો એક શ્રેષ્ઠ માર્ગ છે. દાખલા તરીકે, જો તમે અલગ-અલગ વિષયો પરના ડેટાના ટેબલ સાથે કામ કરતા હો અને તેમના જવાબો એકબીજાના સંબંધમાં જોવા માંગતા હો, તો તે તમારા માટે વધુ અનુકૂળ રહેશે (અને તમે જેની સાથે ડેટા શેર કરી રહ્યા છો તે લોકો) જો તમારી પાસે હોય બંને જવાબો એક કોલમમાં.

કોલમ્સ અને પંક્તિઓનું નામકરણ: એક્સેલમાં કોલમ અને પંક્તિઓને નામ આપવા માટે, સૌપ્રથમ નામકરણની જરૂર હોય તેવા કોષોને પસંદ કરો. પછી Data > Data Tools > Name Columns and Rows પર જાઓ અને પ્રથમ સંવાદ બોક્સમાં પ્રથમ સેલનું નામ ટાઈપ કરો. જ્યાં સુધી તમારા બધા કોષોને નામ આપવામાં ન આવે ત્યાં સુધી ટાઈપ કરવાનું અથવા ક્લિક કરવાનું ચાલુ રાખો.

કોલમ અથવા પંક્તિઓનું નામકરણ

સ્પ્રેડશીટમાં દરેક કોલમ અને પંક્તિનું ડિફોલ્ટ નામ હોય છે, પરંતુ આ નામો બદલી શકાય છે. જ્યારે તમે ડેટાનો સારાંશ આપતા હોવ અને સમાન કોલમ અથવા પંક્તિ શીર્ષક સતત લાગુ કરવા માંગતા હો ત્યારે આ મદદરૂપ થાય છે.

કોલમનું નામ બદલવા માટે, તે કોલમના કોઈપણ સેલ પર જમણું-ક્લિક કરો અને “કોલમનું નામ” પસંદ કરો. નવું નામ ટાઈપ કરો અને એન્ટર દબાવો. પંક્તિનું નામ બદલવા માટે, તે પંક્તિના કોઈપણ કોષ પર જમણું-ક્લિક કરો અને “રો લેબલ્સ” પસંદ કરો. નવું નામ ટાઈપ કરો અને એન્ટર દબાવો.

કોલમ અથવા પંક્તિઓ દાખલ કરવી Excel માં સૌથી સરળ ડેટા મેનીપ્યુલેશન તકનીકોમાંની એક કોલમ અથવા પંક્તિઓ દાખલ કરવી છે.

આ તકનીક તમને તમારી સ્પ્રેડશીટમાં વધુ કોલમ અથવા પંક્તિઓ ઉમેરીને વધુ સ્પષ્ટતા અને ચોકસાઈ સાથે તમારા ડેટાનું વિશ્લેષણ કરવા દે છે. તેનો ઉપયોગ કરી શકાય છે તમારા ડેટાના વિવિધ પાસાઓ બતાવો, જેમ કે વિવિધ વર્ષ, પ્રદેશો, ઉત્પાદનો વગેરે.

આ તકનીકનો ઉપયોગ કરી શકાય તેવા બે ઉદાહરણો છે:

- 1 તમે ચોક્કસ પ્રદેશ માટે જુદા જુદા વર્ષોની સરખામણી કરવા માંગો છો.
- 2 તમે વિવિધ પ્રદેશોમાં ઉત્પાદન કેવું પ્રદર્શન કરે છે તે જોવા માંગો છો.

નિષ્કર્ષ

ડેટા મેનીપ્યુલેશન એ સમય માંગી લેતું અને જટિલ કાર્ય હોઈ શકે છે. પરંતુ યોગ્ય તકનીક તમને ઘણો સમય બચાવવા અને ભૂલો કરવાનું ટાળવામાં મદદ કરી શકે છે. આ લેખમાંની 9 મૂળભૂત તકનીકો તમને Excel માં ડેટા મેનીપ્યુલેશન નેવિગેટ કરવામાં મદદ કરી શકે છે.

સેલ રેન્જ શું છે

એક્સેલ ફાઈલમાં સેલ રેન્જ એ પસંદ કરેલા કોષોનો સંગ્રહ છે. આ શ્રેણી સામાન્ય રીતે સપ્રમાણ (ચોરસ) હોય છે, પરંતુ અલગ-અલગ કોષો એકસરખા અસ્તિત્વમાં હોઈ શકે છે. કોષ શ્રેણીનો પણ સૂત્રમાં ઉલ્લેખ કરી શકાય છે.

સ્પ્રેડશીટમાં, કોષ શ્રેણીને શ્રેણીના ઉપલા ડાબા કોષ (લઘુત્તમ મૂલ્ય) અને શ્રેણીના નીચલા જમણા કોષ (મહત્તમ મૂલ્ય)ના સંદર્ભ દ્વારા વ્યાખ્યાયિત કરવામાં આવે છે. આખરે આ પસંદગીમાં અલગ કોષો ઉમેરી શકાય છે,

પછી શ્રેણીને અનિયમિત કોષ શ્રેણી કહેવામાં આવે છે. Excel માં, લઘુત્તમ અને મહત્તમ મૂલ્ય શામેલ છે. તે ગાણિતિક શ્રેણીથી અલગ છે, જેમાં તે મહત્તમ અને લઘુત્તમ મૂલ્ય વચ્ચેના મૂલ્યોનો સંગ્રહ છે.

એક સપ્રમાણ કોષ શ્રેણી નીચે પ્રમાણે દેખાઈ શકે છે. આ શ્રેણી માટે સંકેત છે (A1:C6); ઉપલા ડાબા કોષ A1 થી નીચે જમણા કોષ C6 સુધી

કોષ શ્રેણીનો ઉપયોગ ફોર્મ્યુલાની અંદર થઈ શકે છે, ઉદાહરણ તરીકે પસંદ કરેલ કોષોની અંદરના મૂલ્યોના સરવાળાની ગણતરી કરવા માટે. કોષ શ્રેણી (A1:C6) માં તમામ મૂલ્યોના સરવાળા માટે સંકેત =SUM(A1:C6) છે.



© NIMI
NOT TO BE REPUBLISHED

COPA - કોષ્ટકો અને કોષ્ટક ડેટા મેનેજ કરો

કોષ્ટકો બનાવો અને ફોર્મેટ કરો (Create and format tables)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ટેબલ બનાવો અને ફોર્મેટ કરો
- સોર્ટ કરો અને ફિલ્ટર કરો
- કોષ્ટકને ટેક્સ્ટમાં કન્વર્ટ કરો.

ટેબલ બનાવો અને ફોર્મેટ કરો

તમારી શ્રેણીને કોષ્ટક તરીકે ફોર્મેટ કરવું એ એક્સેલને કહે છે કે તે પંક્તિઓ અને કૉલમ્સ બધા સંબંધિત છે, અને તે પ્રથમ હરોળમાં હેડર છે. અને આ કરવાથી, તમારી શ્રેણી હવે અર્થ ધરાવે છે. એક્સેલ તેને વધુ સારી રીતે સમજે છે. અને તેની સાથે, ઘણા બધા વધારાના લાભો જન્મે છે.

ટેબલ ફોર્મેટ કરો

ટેબલ ફોર્મેટિંગ ઓબ્જેક્ટ્સ. કોષ્ટકમાં કેપ્શન, પંક્તિ જૂથો અને કૉલમ જૂથો હોઈ શકે છે. પંક્તિ જૂથમાં પંક્તિઓ હોય છે, જ્યારે કૉલમ જૂથમાં કૉલમ હોય છે. પંક્તિઓ અને કૉલમ કોષો ધરાવે છે. કોષ્ટકો નીચેથી ઉપરના ચોક્કસ ક્રમમાં સ્તરો તરીકે પ્રસ્તુત થાય છે: કોષ્ટક, કૉલમ જૂથો, કૉલમ્સ, પંક્તિ જૂથો, પંક્તિઓ અને કોષો.

સોર્ટિંગ

સોર્ટિંગ એ ડેટા વિશ્લેષણની બીજી તકનીક છે અને તેનો ઉપયોગ તમારા ડેટાના ક્રમને ફરીથી ગોઠવવા માટે થાય છે. તમારા ડેટાની શોધખોળ અને સમજવાની આ એક સરળ રીત છે.

ઉદાહરણ તરીકે, ધારો કે તમારી પાસે 5 અલગ-અલગ નંબરોની સૂચિ છે:

1, 10, 2, 3, 4

જો આપણે આ સૂચિને ચડતા ક્રમમાં (સૌથી નીચાથી ઉચ્ચતમ સુધી) સોર્ટ કરવા માંગતા હોઈએ, તો અમે આ સૂચિ માટેના કૉલમ મથાળા પર ક્લિક કરીશું અને પછી પસંદ કરીશું. “ચડતા સોર્ટ કરો”. આ સૂચિને આ રીતે ગોઠવશે:

1, 2, 3, 4, 10. સંખ્યાઓને ચડતા ક્રમમાં સોર્ટ કરીને આપણે જોઈ શકીએ છીએ કે તેઓ કદમાં વધી રહ્યા છે. જો આપણે સોર્ટ ક્રમને ઉતરતા (ઉચ્ચથી નીચો) માં બદલવા માંગતા હોઈએ તો અમે કૉલમ A પર ક્લિક કરીશું અને “સોર્ટ ડિસેન્ડિંગ” પસંદ કરીશું.

4, 3, 2, 10, 1. ફરીથી સૌથી વધુ સંખ્યાથી સૌથી ઓછી સંખ્યા સુધી ઉતરતા ક્રમમાં સોર્ટ કરીને આપણે જોઈ શકીએ છીએ કે તેઓ કદમાં ઘટાડો કરી રહ્યાં છે

ફિલ્ટરિંગ

ફિલ્ટરિંગ એ ચોક્કસ માપદંડો દ્વારા ડેટાને સોર્ટ કરવાની પ્રક્રિયા છે. મોટા ડેટાસેટમાંથી ડેટાના સબસેટ્સને ઓળખવાની તે એક અસરકારક રીત છે.

નીચેના ઉદાહરણમાં, અમારી પાસે 2012-2015ના માસિક વેચાણ ડેટા સાથેનો ડેટાસેટ છે. ફિલ્ટરિંગ ઉપયોગી છે જો તમે માત્ર વર્ષ માટે કુલ વેચાણ જોવા માંગતા હોવ અથવા જો તમે જાણવા માંગતા હોવ કે કેટલા મહિનામાં હકારાત્મક વૃદ્ધિ થઈ છે.

પ્રથમ, ઓળખો કે કઈ કૉલમમાં ફિલ્ટર માપદંડ છે જેનો ઉપયોગ તમારા ડેટાને ફિલ્ટર કરવા માટે કરવામાં આવશે. આ કિસ્સામાં, મેં “સેલ્સ ગ્રોથ” નામની નવી કૉલમ બનાવી છે. આગળ, તમારી “સેલ્સ ગ્રોથ” કૉલમ હાઈલાઈટ કરો અને ટૂલબાર પરના ડેટા મેનૂમાંથી “ફિલ્ટર” પસંદ કરો. આ એક સંવાદ બોક્સ ખોલશે જ્યાં તમે તમારા ફિલ્ટરિંગ માપદંડને ઇનપુટ કરી શકો છો. આ ઉદાહરણમાં, હું મારા ફિલ્ટર માપદંડ તરીકે “>0%” નો ઉપયોગ કરી રહ્યો છું જેથી સકારાત્મક આવક વૃદ્ધિ > 0% સાથે વર્ષો સુધી કુલ વેચાણની ગણતરી કરવામાં આવે.

કોષ્ટકને ટેક્સ્ટમાં કન્વર્ટ કરો

- 1 તમે ટેક્સ્ટમાં કન્વર્ટ કરવા માંગો છો તે પંક્તિઓ અથવા કોષ્ટક પસંદ કરો.
- 2 લેઆઉટ ટેબ પર, ડેટા વિભાગમાં, ટેક્સ્ટમાં કન્વર્ટ કરો ક્લિક કરો.
- 3 ટેક્સ્ટમાં કન્વર્ટ કરો બોક્સ, સાથે અલગ લખાણ હેઠળ, તમે કૉલમની સીમાઓની જગ્યાએ ઉપયોગ કરવા માંગો છો તે વિભાજક અક્ષરને ક્લિક કરો. ...
- 4 બરાબર ક્લિક કરો

MS-Excel 2010 માં કાર્યો અને સૂત્રો (Functions and formulas in MS-Excel 2010)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- માઈક્રોસોફ્ટ એક્સેલના લક્ષણો અને કાર્યો
- સૂત્રો અને કાર્યો
- એક્સેલ 2010 માં ફરો
- શરતી ફોર્મેટિંગ
- એક્સેલ સ્પ્રેડશીટ ડેટાને લિંક કરો.

એમએસ એક્સેલમાં ફોર્મ્યુલા

ફોર્મ્યુલા, વર્કશીટ માત્ર સરળ ટેબ્યુલર હશે ડેટાનું પ્રતિનિધિત્વ. સૂત્રમાં વિશેષ કોડ હોય છે, જે કોષમાં દાખલ થાય છે. તે કેટલીક ગણતરીઓ કરે છે અને પરિણામ આપે છે, જે સેલમાં પ્રદર્શિત થાય છે.

સૂત્રો વિવિધ ઓપરેટરો અને વર્કશીટનો ઉપયોગ કરે છે મૂલ્યો અને ટેક્સ્ટ સાથે કામ કરવા માટેના કાર્યો. મૂલ્યો અને સૂત્રોમાં વપરાયેલ ટેક્સ્ટ અન્ય કોષોમાં સ્થિત હોઈ શકે છે, જે ડેટા બદલવાનું સરળ બનાવે છે અને વર્કશીટ્સ આપે છે ગતિશીલ પ્રકૃતિ. ઉદાહરણ તરીકે, તે ઝડપથી બદલી શકે છે વર્કશીટમાંનો ડેટા અને ફોર્મ્યુલા કામ કરે છે.

Elements of Formulas

સૂત્રમાં આમાંના કોઈપણ ઘટકો હોઈ શકે છે?

- ગાણિતિક ઓપરેટર્સ, જેમ કે +(ઉમેરવા માટે) અને *(ગુણાકાર માટે)

ઉદાહરણ –

=A1+A2 સેલ A1 અને A2 માં મૂલ્યો ઉમેરે છે.

- મૂલ્યો અથવા ટેક્સ્ટ

ઉદાહરણ –

=200*0.5 200 ગુણ્યા 0.5. આ ફોર્મ્યુલા ફક્ત ઉપયોગ કરે છે મૂલ્યો, અને તે હંમેશા 100 જેટલું જ પરિણામ આપે છે.

- કોષ સંદર્ભો (નામિત કોષો અને શ્રેણીઓ)

ઉદાહરણ –

=A1=C12 સેલ A1 ને સેલ C12 સાથે સરખાવે છે. જો કોષો સમાન હોય છે, સૂત્ર સાચું આપે છે; અન્યથા, તે FALSE પરત કરે છે.

- વર્કશીટ ફંક્શન્સ (જેમ કે SUM અથવા AVERAGE)

ઉદાહરણ –

=SUM(A1:A12) A1:A12 શ્રેણીમાં મૂલ્યો ઉમેરે છે.

ફોર્મ્યુલા બનાવી રહ્યા છીએ

ફોર્મ્યુલા બનાવવા માટે ફોર્મ્યુલા બારમાં ટાઈપ કરવું જરૂરી છે. ફોર્મ્યુલા ‘=’ ચિહ્નથી શરૂ થાય છે. સૂત્રો બનાવતી વખતે મેન્યુઅલી, અને ક્યાં તો સેલ એડ્રેસ ટાઈપ કરી શકે છે અથવા વર્કશીટમાં તેમને નિર્દેશ કરી શકે છે. પોઈન્ટિંગનો ઉપયોગ કરીને ફોર્મ્યુલા માટે સેલ એડ્રેસ સપ્લાય કરવાની પદ્ધતિ ઘણી વાર છે ફોર્મ્યુલા બનાવવાની સરળ અને વધુ શક્તિશાળી પદ્ધતિ. બિલ્ટ-ઇન ફંક્શનનો ઉપયોગ કરતી વખતે, સેલ પર ક્લિક કરવા અથવા ખેંચો કોષ શ્રેણી દ્વારા જે વ્યાખ્યાયિત કરતી વખતે ઉપયોગ કરવા માંગે છે ફંક્શન આર્ગ્યુમેન્ટ્સમાં ફંક્શનની દલીલો ફિગ 1 માં બતાવ્યા પ્રમાણે ડાયલોગ બોક્સ.



જલદી ફોર્મ્યુલા એન્ટ્રી પૂર્ણ કરો, એક્સેલ ગણતરી કરે છે પરિણામ, જે પછી કોષની અંદર અંદર પ્રદર્શિત થાય છે કાર્યપત્રક (સૂત્રની સામગ્રી, જે કે, ફોર્મ્યુલા બાર પર કોઈપણ સમયે દૃશ્યમાન થવાનું ચાલુ રાખો સેલ સક્રિય છે). જો તમે સૂત્રમાં ભૂલ કરો છો એક્સેલને ફોર્મ્યુલાની ગણતરી કરવામાં સક્ષમ થવાથી અટકાવે છે બિલકુલ, એક્સેલ એક ચેતવણી સંવાદ બોક્સ દર્શાવે છે જે સૂચવે છે કે સમસ્યાને કેવી રીતે ઠીક કરવી.

ફોર્મ્યુલામાં કાર્યો

ઘણા ફોર્મ્યુલા ઉપલબ્ધ વર્કશીટનો ઉપયોગ કરીને બનાવવામાં આવે છે કાર્યો આ કાર્યો મોટા પ્રમાણમાં વધારવા માટે સક્ષમ કરે છે સૂત્રોની શક્તિ અને ગણતરીઓ કરે છે જે છે માત્ર ઓપરેટરોનો ઉપયોગ કરવો મુશ્કેલ છે. ઉદાહરણ તરીકે, અને કરી શકો છો લોગરીધમની ગણતરી કરવા માટે LOG અથવા SIN ફંક્શનનો ઉપયોગ કરો અથવા સિન રેશિયો. અને એકલા ગાણિતિક ઓપરેટરોનો ઉપયોગ કરીને આ જટિલ ગણતરી કરી શકતા નથી.

કાર્યોનો ઉપયોગ કરવો

જ્યારે ટાઇપ = સાઈન કરો અને પછી કોઈપણ મૂળાક્ષર ટાઇપ કરો શોધાયેલ કાર્યો આકૃતિ 2 બતાવશે.

Fig 2



ધારો કે શ્રેણીમાં સૌથી મોટું મૂલ્ય નક્કી કરવાની જરૂર છે. ફંક્શનનો ઉપયોગ કર્યા વિના ફોર્મ્યુલા જવાબ કહી શકતી નથી. અમે ફોર્મ્યુલાનો ઉપયોગ કરીશું જે પરત કરવા માટે MAX ફંક્શનનો ઉપયોગ કરે છે B3:B8 શ્રેણીમાં સૌથી મોટું મૂલ્ય =MAX(A1:D100) ફિગ 3 માં બતાવ્યા પ્રમાણે.

Fig 3



કાર્યોનું બીજું ઉદાહરણ. જો કોષ શોધવા માટે ધારો મહિનો 1900 કરતા વધારે હોય તો અમે તેને બોનસ આપી શકીએ છીએ વેચાણ પ્રતિનિધિ. આપણે તેને લેખન દ્વારા પ્રાપ્ત કરી શકીએ છીએ =IF(B9>1900,"હા","ના") તરીકે IF વિધેયો સાથેનું સૂત્ર ફિગ 4 માં બતાવ્યા પ્રમાણે.

Fig 4



કાર્ય દલીલો

ઉપરોક્ત ઉદાહરણોમાં, નોંધ લો કે તમામ કાર્યો વપરાય છે કૌંસ કૌંસની અંદરની માહિતી દલીલોની સૂચિ છે.

તેઓ દલીલોનો ઉપયોગ કેવી રીતે કરે છે તેના આધારે કાર્યો બદલાય છે. આધાર રાખીને તેને શું કરવું છે તેના પર, ફંક્શનનો ઉપયોગ કરી શકે છે.

- કોઈ દલીલો નથી - ઉદાહરણો? Now(), તારીખ(), વગેરે.
- એક દલીલ - UPPER(), LOWER(), વગેરે.
- દલીલોની નિશ્ચિત સંખ્યા - IF(), MAX(), MIN(), AVERAGE(), વગેરે.
- અનંત સંખ્યામાં દલીલો
- વૈકલ્પિક દલીલો

બિલ્ટ ઇન ફંક્શન્સ

એમએસ એક્સેલમાં ઘણા બિલ્ટ-ઇન ફંક્શન છે, જેનો આપણે ઉપયોગ કરી શકીએ છીએ અમારા સૂત્રમાં. શ્રેણી દ્વારા તમામ કાર્યો જોવા માટે, ફોર્મ્યુલા ટેબ પસંદ કરો “ ફિગ-5 માં બતાવ્યા પ્રમાણે ફંક્શન દાખલ કરો. પછી ઇન્સર્ટ ફંક્શન ડાયલોગ દેખાય છે જેમાંથી આપણે કરી શકીએ છીએ કાર્ય પસંદ કરો.

Fig 5



શ્રેણીઓ દ્વારા કાર્યો

ચાલો MS Excel માં બિલ્ટ-ઇન ફંક્શન્સ જોઈએ.

ટેક્સ્ટ કાર્યો

LOWER : પૂરા પાડવામાં આવેલ ટેક્સ્ટ સ્ટ્રિંગમાં બધા અક્ષરોને રૂપાંતરિત કરે છે નાના કેસ માટે

UPPER : પૂરા પાડવામાં આવેલ ટેક્સ્ટ સ્ટ્રિંગમાં બધા અક્ષરોને રૂપાંતરિત કરે છે અપર કેસ સુધી

TRIM : ડુપ્લિકેટ જગ્યાઓ અને સ્પેસને દૂર કરે છે ટેક્સ્ટ સ્ટ્રિંગની શરૂઆત અને અંત

CONCATENATE : બે અથવા વધુ ટેક્સ્ટ સાથે જોડાય છે તાર

LEFT : માંથી અક્ષરોની ચોક્કસ સંખ્યા પરત કરે છે પૂરા પાડવામાં આવેલ ટેક્સ્ટ સ્ટ્રિંગની શરૂઆત.

MID : માંથી અક્ષરોની ચોક્કસ સંખ્યા પરત કરે છે પૂરી પાડવામાં આવેલ ટેક્સ્ટ સ્ટ્રિંગની મધ્યમાં

RIGHT : માંથી અક્ષરોની ચોક્કસ સંખ્યા પરત કરે છે પૂરી પાડવામાં આવેલ ટેક્સ્ટ સ્ટ્રિંગનો અંત.

LEN: આપેલ ટેક્સ્ટ સ્ટ્રિંગની લંબાઈ પરત કરે છે

FIND : પૂરા પાડવામાં આવેલ અક્ષર અથવા ટેક્સ્ટની સ્થિતિ પરત કરે છે સપ્લાય કરેલ ટેક્સ્ટ સ્ટ્રિંગની અંદરથી સ્ટ્રિંગ (કેસ-સેન્સિટિવ).

તારીખ સમય

DATE: વપરાશકર્તા દ્વારા પૂરા પાડવામાં આવેલ વર્ષ, મહિનાની તારીખ પરત કરે છે અને દિવસ.

TIME: વપરાશકર્તા દ્વારા પૂરા પાડવામાં આવેલ કલાક, મિનિટમાંથી, સમય પરત કરે છે અને બીજું.

DATEVALUE: તારીખ દર્શાવતી ટેક્સ્ટ સ્ટ્રિંગને માં રૂપાંતરિત કરે છે પૂર્ણાંક જે એક્સેલના તારીખ-સમયમાં તારીખ દર્શાવે છે કોડ

TIMEVALUE: સમય દર્શાવતી ટેક્સ્ટ સ્ટ્રિંગને માં રૂપાંતરિત કરે છે દર્શાવેલ જે એક્સેલમાં સમય દર્શાવે છે.

NOW: વર્તમાન તારીખ અને સમય પરત કરે છે.

TODAY: આજની તારીખ પરત કરે છે.

Statistical

MAX: પૂરા પાડવામાં આવેલ સૂચિમાંથી સૌથી મોટું મૂલ્ય પરત કરે છે સંખ્યાઓ

MIN: પૂરી પાડવામાં આવેલ સૂચિમાંથી સૌથી નાનું મૂલ્ય પરત કરે છે સંખ્યાઓ

AVERAGE : પૂરા પાડવામાં આવેલ સૂચિની સરેરાશ પરત કરે છે સંખ્યાઓ

COUNT: a માં સંખ્યાત્મક મૂલ્યોની સંખ્યા પરત કરે છે કોષો અથવા મૂલ્યોનો પૂરો પાડવામાં આવેલ સમૂહ.

COUNTIF: કોષોની સંખ્યા પરત કરે છે (સપ્લાય કરેલ શ્રેણી), જે આપેલ માપદંડને સંતોષે છે.

SUM: પૂરી પાડવામાં આવેલ સંખ્યાઓની સૂચિનો સરવાળો આપે છે

Logical

AND: સંખ્યાબંધ વપરાશકર્તા-નિર્ધારિત શરતોનું પરીક્ષણ કરે છે અને જો બધી શરતોનું મૂલ્યાંકન TRUE થાય તો TRUE પરત કરે છે, અથવા અન્યથા FALSE

OR: સંખ્યાબંધ વપરાશકર્તા-નિર્ધારિત શરતોનું પરીક્ષણ કરે છે અને જો કોઈપણ શરતોનું મૂલ્યાંકન TRUE થાય તો TRUE પરત કરે છે, અથવા અન્યથા FALSE.

NOT: તાર્કિક મૂલ્ય આપે છે જે a ની વિરુદ્ધ છે વપરાશકર્તાએ તાર્કિક મૂલ્ય અથવા અભિવ્યક્તિ પૂરી પાડી એટલે કે વળતર FALSE જો આપેલ દલીલ સાચી હોય અને પરત કરે જો આપેલ દલીલ FAL હોય તો TRUE

Math & Trig

ABS: a નું સંપૂર્ણ મૂલ્ય (એટલે કે મોડ્યુલસ) પરત કરે છે પુરવઠો નંબર.

SIGN: પૂરા પાડવામાં આવેલ નંબરનું ચિહ્ન (+1, -1 અથવા 0) પરત કરે છે.

SQRT: આપેલનું ધન વર્ગમૂળ પરત કરે છે સંખ્યા

MOD: બે પૂરા પાડવામાં આવેલ નંબરો વચ્ચેના વિભાજનમાંથી બાકીની રકમ પરત કરે છે.

MS Excel માં ફિલ્ટર્સ

એમએસ એક્સેલમાં ડેટા ફિલ્ટર કરવાનો અર્થ ફક્ત પ્રદર્શિત કરવાનો છે પંક્તિઓ જે ચોક્કસ શરતોને પૂર્ણ કરે છે. (અન્ય પંક્તિઓ મળે છે છુપાયેલ.)

સ્ટોર ડેટાનો ઉપયોગ કરીને, જો વપરાશકર્તાને ડેટા જોવામાં રસ હોય જ્યાં જૂતાનું કદ 36 છે, પછી આ કરવા માટે ફિલ્ટર સેટ કરો. અનુસરો આ કરવા માટે નીચે જણાવેલ પગલાંઓ.

- હેડર રો પર કર્સર મૂકો.
- **Data Tab -> Filter** માં બતાવ્યા પ્રમાણે ફિલ્ટર સેટ કરવા માટે ડેટા ટેબ ફિલ્ટર પસંદ કરો ફિગ 6.



- એરિયા રો હેડરમાં ડ્રોપ-ડાઉન એરો પર ક્લિક કરો અને સિલેક્ટ ઓલમાંથી ચેક માર્ક દૂર કરો, જે બધું પસંદ કરે છે.
- પછી માપ 36 માટે ચેક માર્ક પસંદ કરો જે ફિલ્ટર કરશે બતાવ્યા પ્રમાણે શૂ સાઈઝ 36 નો ડેટા અને ડિસ્પ્લે ડેટા ફિગ-23 માં.
- કેટલાક પંક્તિ નંબરો ખૂટે છે; આ પંક્તિઓ ફિલ્ટર કરેલ (છુપાયેલ) ડેટા ધરાવે છે.
- એરિયા કોલમમાં ડ્રોપ-ડાઉન એરો છે જે હવે બતાવે છે એક અલગ ગ્રાફિક - એક ચિહ્ન જે કોલમ સૂચવે છે ફિલ્ટર કરવામાં આવે છે. (ફિગ 7)

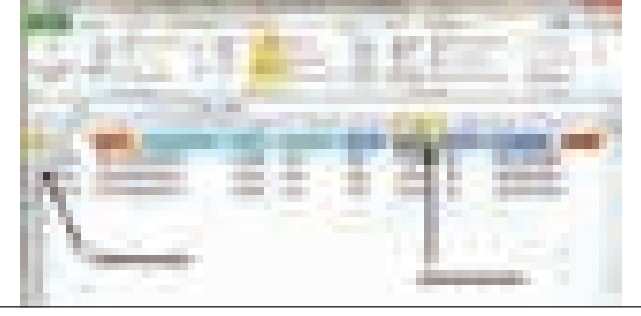


બહુવિધ ફિલ્ટર્સનો ઉપયોગ (ફિગ-24)

બહુવિધ શરતો દ્વારા રેકૉર્ડનું ફિલ્ટરિંગ એટલે કે બહુવિધ દ્વારા કોલમ મૂલ્યો. ધારો કે માપ 36 ફિલ્ટર કર્યા પછી, કરવાની જરૂર છે ફિલ્ટર રાખો

જ્યાં કોફીનો રંગ બરાબર હોય. સેટ કર્યા પછી શૂ સાઈઝ માટે ફિલ્ટર કરો, કલર કોલમ પસંદ કરો અને પછી રંગ માટે ફિલ્ટર સેટ કરો. (ફિગ 8)

Fig 8



એમએસ એક્સેલમાં વર્ગીકરણ

એમએસ એક્સેલમાં ડેટાનું સોર્ટિંગ પંક્તિઓને આધારે ફરીથી ગોઠવે છે ચોક્કસ સ્તંભની સામગ્રી. મૂકવા માટે ટેબલ સોર્ટ કરો મૂળાક્ષરોના ક્રમમાં નામો અથવા રકમ દ્વારા ડેટાને સોર્ટ કરો સૌથી નાનાથી મોટા અથવા સૌથી નાનાથી નાના.

ડેટાને સોર્ટ કરવા માટે નીચે જણાવેલ સ્ટેપ્સને અનુસરો.

- ફિગ 9 મુજબ ડેટાને સોર્ટ કરવા માટે કૉલમ પસંદ કરો.

Fig 9



- ડેટા ટેબ પસંદ કરો “ નીચે સોર્ટ કરો સંવાદ દેખાય છે.
- જો વપરાશકર્તા પસંદ કરેલ કૉલમના આધારે ડેટાને સોર્ટ કરે છે, પસંદગી સાથે ચાલુ રાખો અથવા જો ડેટા પસંદ કરો અન્ય કૉલમના આધારે વર્ગીકરણ, વિસ્તૃત કરો પસંદ કરો પસંદગી.
- નીચેની શરતોના આધારે ડેટાને સોર્ટ કરો ફિગ 10.

Fig 10



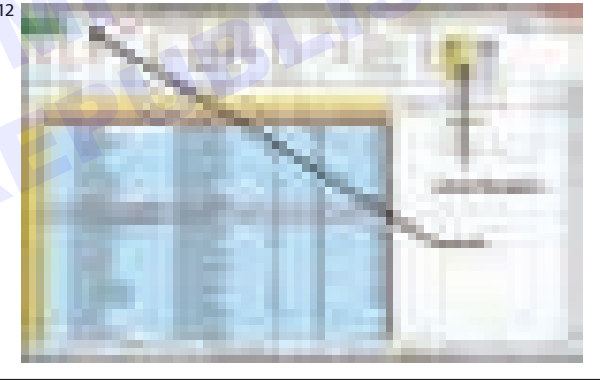
- મૂલ્યો - મૂળાક્ષરો અથવા આંકડાકીય રીતે.
- સેલનો રંગ - સેલના રંગ પર આધારિત.
- ફોન્ટ કલર - ફોન્ટ કલર પર આધારિત.
- સેલ આઈકોન - સેલ આઈકોન પર આધારિત.
- Ok પર ક્લિક કરવાથી ફિગ11 મુજબ ડેટા સોર્ટ થશે.

Fig 11



હોમ ટેબમાંથી સોર્ટિંગ વિકલ્પ પણ ઉપલબ્ધ છે. પસંદ કરો હોમ ટેબ “ સોર્ટ કરો અને ફિલ્ટર કરો. તમે સમાન સંવાદ જોઈ શકો છો ફિગ12 મુજબ રેકૉર્ડને સોર્ટ કરો.

Fig 12



MS Excel માં શ્રેણીઓ:

કોષ એ વર્કશીટમાં એક જ તત્વ છે જે a ને પકડી શકે છે મૂલ્ય, અમુક ટેક્સ્ટ અથવા સૂત્ર. કોષની ઓળખ તેના દ્વારા થાય છે સરનામું, જેમાં તેના કૉલમ અક્ષર અને પંક્તિનો સમાવેશ થાય છે સંખ્યા ઉદાહરણ તરીકે, સેલ B1 એ સેકન્ડનો કોષ છે કૉલમ અને પ્રથમ પંક્તિ.

કોષોના જૂથને શ્રેણી કહેવામાં આવે છે. તમે શ્રેણી નિયુક્ત કરો. તેના ઉપલા-ડાબે સેલ સરનામું સ્પષ્ટ કરીને સરનામું અને તેના નીચલા-જમણા કોષનું સરનામું, કોલોન દ્વારા અલગ.

શ્રેણીઓનું ઉદાહરણ:

- C24** - એક શ્રેણી જેમાં એક કોષનો સમાવેશ થાય છે.
- A1:B1** - બે કોષો જે એક પંક્તિ અને બે પર કબજો કરે છે કૉલમ.
- A1:A100** - કૉલમ Aમાં 100 કોષો.
- A1:D4** - 16 કોષો (ચાર કૉલમ દ્વારા ચાર પંક્તિઓ).

શ્રેણીઓ પસંદ કરવી

વિવિધ રીતે શ્રેણી પસંદ કરી રહ્યા છીએ?

- ડાબું માઉસ બટન દબાવો અને હાઈલાઈટ કરીને ખેંચો શ્રેણી. પછી માઉસ બટન છોડો. જો ખેંચો સ્ક્રીનના અંતમાં, વર્કશીટ સ્કોલ થશે.
- જ્યારે નેવિગેશન કીનો ઉપયોગ કરે ત્યારે Shift કી દબાવો શ્રેણી પસંદ કરો.
- F8 દબાવો અને પછી સેલ પોઈન્ટરને સાથે ખસેડો શ્રેણીને પ્રકાશિત કરવા માટે નેવિગેશન કીઓ. નેવિગેશન કીને સામાન્ય હિલચાલમાં પરત કરવા માટે ફરીથી F8 દબાવો.
- નામ બોક્સમાં કોષ અથવા શ્રેણી સરનામું લખો અને Enter દબાવો. એક્સેલ સ્પષ્ટ કરેલ કોષ અથવા શ્રેણી પસંદ કરે છે. (ફિગ 13)

સંપૂર્ણ પંક્તિઓ અને કોલમ પસંદ કરી રહ્યા છીએ (ફિગ-30)

જ્યારે વપરાશકર્તાને સમગ્ર પંક્તિ અથવા કોલમ પસંદ કરવાની જરૂર હોય સમગ્ર પંક્તિઓ અને સ્તંભોને સમાન રીતે પસંદ કરી શકો છો પસંદગીની શ્રેણીઓ તરીકેની રીત:

Fig 13



Fig 1

- એક પંક્તિ પસંદ કરવા માટે પંક્તિ અથવા કોલમ બોર્ડર પર ક્લિક કરો અથવા કોલમ.
- બહુવિધ સંલગ્ન પંક્તિઓ અથવા કોલમ પસંદ કરવા માટે, a પર ક્લિક કરો પંક્તિ અથવા કોલમની સરહદ અને વધારાને પ્રકાશિત કરવા માટે ખેંચો પંક્તિઓ અથવા કોલમ.
- બહુવિધ (બિન અડીને) પંક્તિઓ અથવા કોલમ પસંદ કરવા માટે, પંક્તિ અથવા કોલમ બોર્ડર્સ પર ક્લિક કરતી વખતે Ctrl દબાવો. (ફિગ 14)

Fig 14



ચાર્ટ્સ મેનેજ કરો (Manage Charts)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ચાર્ટ બનાવો
- ચાર્ટમાં ફેરફાર કરો
- ફોર્મેટ ચાર્ટ.

ચાર્ટ

ડેટાને દર્શાવવા અને તેની સરખામણી કરવા માટે ચાર્ટ દાખલ કરો. બાર, પાઈ, લાઈન, વિસ્તાર અને સપાટી કેટલાક ઉપલબ્ધ પ્રકારો છે.

એક ચાર્ટ (ગ્રાફ) બનાવો જે તમારા ડેટા માટે ભલામણ કરવામાં આવે છે, લગભગ તેટલી જ ઝડપી ચાર્ટ વિઝાર્ડનો ઉપયોગ કરો જે હવે ઉપલબ્ધ નથી.

એક ચાર્ટ બનાવો

- 1 તે ડેટા પસંદ કરો જેના માટે તમે ચાર્ટ બનાવવા માંગો છો.
- 2 INSERT > ભલામણ કરેલ ચાર્ટ પર ક્લિક કરો.
- 3 ભલામણ કરેલ ચાર્ટ્સ ટેબ પર, એક્સેલ તમારા ડેટા માટે ભલામણ કરે છે તે ચાર્ટની સૂચિમાંથી સ્કોલ કરો અને તમારો ડેટા કેવો દેખાશે તે જોવા માટે કોઈપણ ચાર્ટ પર ક્લિક કરો.
જો તમને તમને ગમતો ચાર્ટ દેખાતો નથી, તો બધા ઉપલબ્ધ ચાર્ટ પ્રકારો જોવા માટે બધા ચાર્ટ પર ક્લિક કરો.
- 4 જ્યારે તમને ગમતો ચાર્ટ મળે, ત્યારે તેના પર ક્લિક કરો > બરાબર.
- 5 ચાર્ટ તત્વો, ચાર્ટ શૈલીઓ અને ચાર્ટ ફિલ્ટર બટનોનો ઉપયોગ કરો, ચાર્ટના ઉપરના જમણા ખૂણે અક્ષના શીર્ષકો અથવા ડેટા લેબલ્સ જેવા ચાર્ટ ઘટકો ઉમેરવા, તમારા ચાર્ટના દેખાવને કસ્ટમાઈઝ કરવા અથવા આમાં દર્શાવેલ ડેટાને બદલવા માટે. ચાર્ટ
- 6 વધારાની ડિઝાઈન અને ફોર્મેટિંગ સુવિધાઓને એક્સેસ કરવા માટે, ચાર્ટ ઉમેરવા માટે ચાર્ટમાં ગમે ત્યાં ક્લિક કરો રિબન પર ટૂલ્સ, અને પછી ડિઝાઈન અને ફોર્મેટ ટેબ પર તમને જોઈતા વિકલ્પો પર ક્લિક કરો

એક્સેલ ચાર્ટમાં ડેટા સિરીઝ શું છે?

Excel માં ડેટા સિરીઝ શું છે? ડેટા શ્રેણી એ સ્પ્રેડશીટ પરની કોલમમાં સતત મૂલ્યો અથવા સંખ્યાઓનો સંગ્રહ છે. અમે ડેટાની પ્રારંભિક શ્રેણીમાંથી ગ્રાફ બનાવીને ડેટા શ્રેણી બનાવી શકીએ છીએ, પછી, અમે ડેટાની બીજી શ્રેણી બનાવી શકીએ છીએ અને તેને પહેલાથી અસ્તિત્વમાં રહેલી ડેટા શ્રેણી સાથે પ્રકાશિત કરી શકીએ છીએ.

એક અલગ ચાર્ટ શીટ પર ચાર્ટમાં ડેટા સિરીઝ ઉમેરો

જો તમારો ચાર્ટ અલગ વર્કશીટ પર છે, તો નવી ડેટા સિરીઝ ઉમેરવા માટે કદાચ ડ્રેગિંગ શ્રેષ્ઠ માર્ગ નથી. તે કિસ્સામાં, તમે ડેટા સ્ત્રોત પસંદ કરો સંવાદ બોક્સમાં ચાર્ટ માટે નવો ડેટા દાખલ કરી શકો છો.

- 1 વર્કશીટ પર કે જેમાં તમારો ચાર્ટ ડેટા છે, ચાર્ટ માટે તમારા હાલના સ્રોત ડેટાની બાજુમાં અથવા નીચે આવેલા કોષોમાં, તમે ઉમેરવા માંગો છો તે નવી ડેટા શ્રેણી દાખલ કરો.
- 2 વર્કશીટ પર ક્લિક કરો જેમાં તમારો ચાર્ટ છે.
- 3 ચાર્ટ પર જમણું-ક્લિક કરો, અને પછી ડેટા પસંદ કરો પસંદ કરો

Fig 1



વર્કશીટ પર ડેટા સ્ત્રોત પસંદ કરો સંવાદ બોક્સ દેખાય છે જેમાં ચાર્ટ માટેનો સ્રોત ડેટા હોય છે.

- 4 સંવાદ બોક્સને ખુલ્લું છોડીને, વર્કશીટમાં ક્લિક કરો, અને પછી નવી ડેટા શ્રેણી સહિત તમે ચાર્ટ માટે ઉપયોગમાં લેવા માંગતા હો તે તમામ ડેટાને પસંદ કરવા માટે ક્લિક કરો અને ખેંચો.

નવી ડેટા સિરીઝ સિલેક્ટ ડેટા સોર્સ ડાયલોગ બોક્સમાં લિજેન્ડ એન્ટ્રીઝ (સિરીઝ) હેઠળ દેખાય છે.

- 5 ડાયલોગ બોક્સ બંધ કરવા અને ચાર્ટ શીટ પર પાછા જવા માટે OK પર ક્લિક કરો.



© NIMI
NOT TO BE REPUBLISHED

પિવટ કોષ્ટકો બનાવો (Create Pivot Tables)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ચાર્ટ બનાવો
- ચાર્ટમાં ફેરફાર કરો
- ફોર્મેટ ચાર્ટ.

પીવટ ટેબલ

PivotTable એ મોટી માત્રામાં ડેટાને ઝડપથી સારાંશ આપવા માટે એક ઇન્ટરેક્ટિવ રીત છે. આંકડાકીય માહિતીનું વિગતવાર વિશ્લેષણ કરવા અને તમારા ડેટા વિશે અણધાર્યા પ્રશ્નોના જવાબ આપવા માટે તમે PivotTable નો ઉપયોગ કરી શકો છો. PivotTable ખાસ કરીને આ માટે રચાયેલ છે: ઘણી વપરાશકર્તા-મૈત્રીપૂર્ણ રીતે મોટી માત્રામાં ડેટાની ક્વેરી કરવી.

પીવટ ટેબલ વર્ક

પીવટ ટેબલનો ઉપયોગ કોષ્ટકમાં સંગ્રહિત ડેટાનો સારાંશ, સોર્ટ, પુનઃગઠન, જૂથ, ગણતરી, કુલ અથવા સરેરાશ ડેટા માટે થાય છે. તે અમને કોલમને હરોળમાં અને પંક્તિઓને કોલમમાં રૂપાંતરિત કરવાની મંજૂરી આપે છે. તે કોઈપણ ક્ષેત્ર (કોલમ) દ્વારા જૂથબદ્ધ કરવાની અને તેના પર અદ્યતન ગણતરીઓનો ઉપયોગ કરવાની મંજૂરી આપે છે.

જૂથ ડેટા

PivotTable માં ડેટાને જૂથબદ્ધ કરવાથી તમને પૃથ્થકરણ કરવા માટે ડેટાનો સબસેટ બતાવવામાં મદદ મળી શકે છે. ઉદાહરણ તરીકે, તમે પિવટ ટેબલમાં અનિચ્છનીય સૂચિ તારીખ અને સમય ફીલ્ડ્સને ક્વાર્ટર અને મહિનાઓમાં જૂથબદ્ધ કરવા માગી શકો છો

COPA - પીવટ કોષ્ટકોનું સંચાલન કરો

પિવટ કોષ્ટકો બનાવો (Open files in MS PowerPoint Presentations)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- MS પાવર પોઈન્ટ પ્રેઝન્ટેશનનો પરિચય
- પિક્ચર ઈફેક્ટ લાગુ કરો અને સંપાદિત કરો.

માઈક્રોસોફ્ટ પાવરપોઈન્ટ એ એક સરળ છતાં શક્તિશાળી સોફ્ટવેર પ્રોગ્રામ છે જે Microsoft દ્વારા વધુ અસરકારક પ્રસ્તુતિઓ બનાવવા માટે વિકસાવવામાં આવ્યો છે. તે મુખ્યત્વે Microsoft Office સ્યુટનો એક ભાગ છે. પ્રોગ્રામમાં સ્લાઈડ્સ અને વર્ડ પ્રોસેસિંગ, ડ્રોઈંગ, ગ્રાફિંગ અને આઉટલાઈનિંગ જેવા વિવિધ ઉપયોગી સાધનોનો સમાવેશ થાય છે. આમ તેનો ઉપયોગ સ્લાઈડ્સમાં ટેક્સ્ટ, કોષ્ટકો, ચાર્ટ્સ, ગ્રાફિક્સ અને મીડિયાને સરળતાથી પ્રદર્શિત કરવા માટે થઈ શકે છે. માઈક્રોસોફ્ટ પાવરપોઈન્ટ વિન્ડોની મુખ્યત્વે ત્રણ નોંધપાત્ર સુવિધાઓ છે જેના પર તમારે પાવરપોઈન્ટ વિશે વધુ શીખતી વખતે ધ્યાન કેન્દ્રિત કરવાની જરૂર છે. આ સુવિધાઓને Microsoft Office બટન, ક્લિક એક્સેસ ટૂલબાર અને રિબન કહેવામાં આવે છે. પાવરપોઈન્ટ સ્લાઈડશો (PPT) એ Microsoft ના સોફ્ટવેર પર બનાવેલ પ્રેઝન્ટેશન છે જે વપરાશકર્તાઓને પ્રેઝન્ટેશનમાં ઓડિયો, વિઝ્યુઅલ અને ઓડિયો/વિઝ્યુઅલ સુવિધાઓ ઉમેરવાની મંજૂરી આપે છે. તેને મલ્ટીમીડિયા ટેકનોલોજી માનવામાં આવે છે અને તે સહયોગ અને સામગ્રી શેરિંગ માટેના સાધન તરીકે પણ કાર્ય કરે છે. પાવરપોઈન્ટને Microsoft Office માં સમાવવામાં આવેલ છે, જે તેને સૌથી વધુ જાણીતા અને વ્યાપક રૂપે એક બનાવે છે પ્રેઝન્ટેશન સોફ્ટવેરની બ્રાન્ડનો ઉપયોગ કર્યો.

પાવરપોઈન્ટ સ્લાઈડશોને પાવરપોઈન્ટ પ્રેઝન્ટેશન તરીકે પણ ઓળખવામાં આવે છે.

ચિત્રો સંપાદિત કરો

- 1 બ્રાઈટનેસ, કોન્ટ્રાસ્ટ અથવા શાર્પનેસ એડજસ્ટ કરો. ચિત્ર પસંદ કરો. પિક્ચર ફોર્મેટ પસંદ કરો અને સુધારાઓ પસંદ કરો....
- 2 કલાત્મક અસરો લાગુ કરો. ચિત્ર પસંદ કરો....
- 3 રંગ બદલો. ચિત્ર પસંદ કરો....
- 4 ચિત્ર અસરો લાગુ કરો. ચિત્ર પસંદ કરો....
- 5 સરહદ ઉમેરો. ચિત્ર પસંદ કરો....
- 6 ચિત્રને સંકુચિત કરો. ચિત્ર પસંદ કરો.
 - બ્રાઈટનેસ, કોન્ટ્રાસ્ટ અથવા શાર્પનેસ એડજસ્ટ કરો
 - ચિત્ર પસંદ કરો.
 - પિક્ચર ફોર્મેટ પસંદ કરો અને સુધારાઓ પસંદ કરો.
 - વિકલ્પોનું પૂર્વાવલોકન કરવા માટે તેના પર હોવર કરો અને તમને જોઈતો એક પસંદ કરો.

વધુ માહિતી માટે, ચિત્રની બ્રાઈટનેસ, કોન્ટ્રાસ્ટ અથવા શાર્પનેસ બદલો જુઓ.

કલાત્મક અસરો લાગુ કરો

- 1 ચિત્ર પસંદ કરો.
- 2 ચિત્રનું ફોર્મેટ પસંદ કરો અને કલાત્મક અસરો પસંદ કરો.
- 3 વિકલ્પોનું પૂર્વાવલોકન કરવા માટે તેના પર હોવર કરો, પછી તમને જોઈતો એક પસંદ કરો.

નોંધ: તમે ચિત્ર પર એક સમયે માત્ર એક જ કલાત્મક અસર લાગુ કરી શકો છો, તેથી અલગ કલાત્મક અસર લાગુ કરવાથી અગાઉ લાગુ કરાયેલ કલાત્મક અસર દૂર થઈ જશે.

4 રંગ બદલો

5 ચિત્ર પસંદ કરો.

6 ચિત્ર ફોર્મેટ પસંદ કરો અને રંગ પસંદ કરો.

- 7 વિકલ્પોનું પૂર્વાવલોકન કરવા માટે તેના પર હોવર કરો, પછી તમને જોઈતો એક પસંદ કરો.

ચિત્ર અસરો લાગુ કરો

- 1 ચિત્ર પસંદ કરો.
- 2 પિક્ચર ફોર્મેટ પસંદ કરો અને પિક્ચર ઈફેક્ટ્સ પસંદ કરો.
- 3 તમને જોઈતું હોય તે પસંદ કરો: શેડો, રિફ્લેક્શન, ગ્લો, સોફ્ટ એજ, બેવલ અથવા 3-D રોટેશન.
વધુ માહિતી માટે, ચિત્ર માટે અસર ઉમેરો અથવા બદલો જુઓ.
- 4 સરહદ ઉમેરો
- 5 ચિત્ર પસંદ કરો.
- 6 પિક્ચર ફોર્મેટ > પિક્ચર બોર્ડર પસંદ કરો, પછી બોર્ડર પસંદ કરો.
વધુ માહિતી માટે, ચિત્ર બોર્ડર્સ ઉમેરો અને દૂર કરો જુઓ.
- 7 પૃષ્ઠભૂમિ દૂર કરો
માહિતી માટે, જુઓ ચિત્રની પૃષ્ઠભૂમિ દૂર કરો.
- 8 ચિત્રને સંકુચિત કરો
- 9 ચિત્ર પસંદ કરો.
- 10 ચિત્રનું ફોર્મેટ પસંદ કરો અને ચિત્રો સંકુચિત કરો પસંદ કરો.
- 11 તમને જોઈતા વિકલ્પો પસંદ કરો, પછી બરાબર પસંદ કરો.

COPA -પ્રસ્તુતિઓ ફોર્મેટ કરો

પિવટ કોષ્ટકો બનાવો (Insert, Format text and paragraphs)

ઉદ્દેશ્યો:આ પાઠના અંતે તમે સમર્થ હશો

- ટેક્સ્ટ સંરેખણ લાગુ કરો.

ટેક્સ્ટ ગોઠવણી કેવી રીતે બદલવી

- જરૂરી લખાણ પસંદ કરો
- હવે, હોમ ટેબમાં ફરો જૂથ શોધો
- ફરો જૂથના તળિયે ચાર ગોઠવણી વિકલ્પો ઉપલબ્ધ છે
- ડાબું ક્લિક કરીને તમારા ઇચ્છિત સંરેખણ વિકલ્પને પસંદ કરો.

ચાર સંરેખણ વિકલ્પો ઉપલબ્ધ છે:

- ટેક્સ્ટને ડાબે સંરેખિત કરો: તે તમારા ટેક્સ્ટને ડાબા હાંસિયા તરફ સંરેખિત કરે છે
- કેન્દ્ર: ટેક્સ્ટને કેન્દ્ર તરફ લાવે છે

- ટેક્સ્ટને જમણે સંરેખિત કરો: તમારા ટેક્સ્ટને જમણા માર્જિન તરફ સંરેખિત કરે છે
- જસ્ટિફાઈ કરો: ટેક્સ્ટને ડાબે અને જમણા માર્જિન બંને તરફ સંરેખિત કરે છે



COPA - કોષ્ટકો અને બુલેટેડ ટેક્સ્ટ મેનેજ કરો**પિવટ કોષ્ટકો બનાવો (Create tables, modify tables, modify bulleted text)**

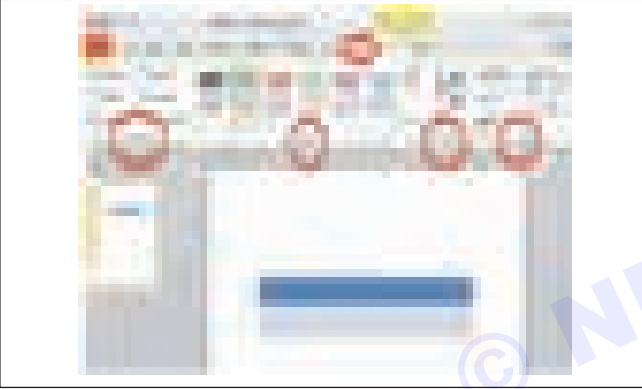
ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- કોષ્ટકને સંશોધિત કરો અથવા ફોર્મેટ કરો
- બુલેટેડ અને નંબરવાળી યાદી લાગુ કરો.

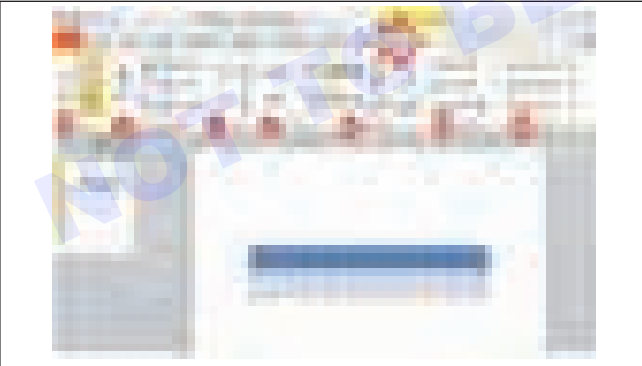
કોષ્ટકને કેવી રીતે સંશોધિત કરવું અથવા ફોર્મેટ કરવું

તે ચોક્કસ કોષ્ટક પસંદ કરો કે જેને તમે સુધારવા માંગો છો. ડિઝાઇન અને લેઆઉટ નામની બે નવી ટેબલ હવે રિબનમાં દેખાશે. આ ટેબલ તમારા કોષ્ટકોને ફોર્મેટ કરવા માટે આદેશોના વિવિધ નવા જૂથો ઓફર કરે છે.

જ્યારે તમે ડિઝાઇન ટેબ પર ક્લિક કરો છો ત્યારે તે તમને આદેશોના ચાર નવા જૂથો દર્શાવે છે; ટેબલ સ્ટાઇલ વિકલ્પો, ટેબલ સ્ટાઇલ, વર્ક આર્ટ સ્ટાઇલ અને ડ્રો બોર્ડર્સ



જ્યારે તમે લેઆઉટ ટેબ પર ક્લિક કરો છો ત્યારે તે આદેશોના છ જૂથો પ્રદર્શિત કરશે; કોષ્ટક, પંક્તિઓ અને કોલમ, મર્જ, કોષ્ટક કદ અને સંરેખણ.

**બુલેટેડ અથવા નંબરવાળી યાદીઓ કેવી રીતે બનાવવી**

- પ્રથમ, ટેક્સ્ટ અથવા સૂચિ પસંદ કરો કે જેમાં તમે તમારા બુલેટ અથવા નંબરિંગ ઉમેરવા માંગો છો
- હવે, હોમ ટેબ પસંદ કરો અને ફક્ત જૂથને શોધો
- બુલેટ અથવા નંબરિંગ બટન પર ક્લિક કરો અથવા વધુ બુલેટ શૈલીઓ અને નંબરિંગ ફોર્મેટ્સ જોવા માટે આ બટનોની બાજુમાં ડ્રોપ-ડાઉન એરો પર ક્લિક કરો
- તમારી શૈલી પસંદ કરો જે તમે ટેક્સ્ટ પર લાગુ કરવા માંગો છો



પિવટ કોષ્ટકો બનાવો (Insert illustrations, Format illustrations and text boxes)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સ્લાઈડમાં ટેક્સ્ટ ઉમેરો
- લેઆઉટ રંગ અથવા શૈલી બદલો
- સ્માર્ટ આર્ટને ફોર્મેટ કરો.

સ્લાઈડમાં ટેક્સ્ટ ઉમેરો તમે ટેક્સ્ટ બોક્સ દાખલ કરીને અને તે બોક્સની અંદર ટાઈપ કરીને પાવરપોઈન્ટ સ્લાઈડ અથવા સ્લાઈડ માસ્ટરમાં ટેક્સ્ટ ઉમેરી શકો છો. પછી તમે ટેક્સ્ટ અથવા સંપૂર્ણ બોક્સ પસંદ કરીને તે ટેક્સ્ટને ફોર્મેટ કરી શકો છો. તમે પ્લેસહોલ્ડર્સ અને આકારોમાં ટેક્સ્ટ પણ ઉમેરી શકો છો.

તેને ખોલવા માટે નીચેનું મથાળું પસંદ કરો અને વિગતવાર સૂચનાઓ જુઓ.

ટેક્સ્ટ બોક્સમાં ટેક્સ્ટ ઉમેરો

- કોઈ પણ વ્યક્તિ સંપાદિત કરી શકે તેવા ટેક્સ્ટ બોક્સમાં ટેક્સ્ટ ઉમેરવા માટે, સામાન્ય દૃશ્યમાં, ટેક્સ્ટ બોક્સની અંદર ક્લિક કરો, અને પછી ટેક્સ્ટને ટાઈપ કરો અથવા પેસ્ટ કરો.
- ટેક્સ્ટ બોક્સમાં ટેક્સ્ટ ઉમેરવા માટે કે જેમાં કાયમી અને સંપાદન ન કરી શકાય તેવું હોય, સ્લાઈડ માસ્ટર વ્યૂમાં, ટેક્સ્ટ બોક્સની અંદર ક્લિક કરો, અને પછી ટેક્સ્ટ ટાઈપ કરો અથવા પેસ્ટ કરો.

સ્લાઈડ પર ગમે ત્યાં ટેક્સ્ટ મૂકવા માટે ટેક્સ્ટ બોક્સનો ઉપયોગ કરો, જેમ કે ટેક્સ્ટ પ્લેસહોલ્ડરની બહાર. ઉદાહરણ તરીકે, ચિત્રમાં કેપ્શન ઉમેરવા માટે, ટેક્સ્ટ બોક્સ બનાવો અને તેને ચિત્રની નજીક મૂકો.

ટેક્સ્ટ ઉમેરો કે જે આકારનો ભાગ છે

આકારો જેમ કે ચોરસ, વર્તુળો, કોલઆઉટ બલૂન અને બ્લોક એરોમાં ટેક્સ્ટ હોઈ શકે છે. જ્યારે તમે ટેક્સ્ટને આકારમાં ટાઈપ કરો છો, ત્યારે ટેક્સ્ટ આકાર સાથે જોડાય છે અને આગળ વધે છે તેની સાથે ફરે છે.

- ટેક્સ્ટ ઉમેરવા માટે કે જે આકારનો ભાગ બને છે, આકાર પસંદ કરો, અને પછી લખાણ લખો અથવા પેસ્ટ કરો.

ટેક્સ્ટ ઉમેરો જે આકારથી સ્વતંત્ર હોય

જો તમે આકારમાં ટેક્સ્ટ ઉમેરવા માંગતા હોવ તો ટેક્સ્ટ બોક્સ સરળ છે, પરંતુ તમે ટેક્સ્ટને આકાર સાથે જોડવા માંગતા નથી. તમે ટેક્સ્ટ બોક્સમાં ટેક્સ્ટમાં બોર્ડર, ફિલ, શેડો અથવા ત્રિ-પરિમાણીય (3-D) અસર ઉમેરી શકો છો.

- ટેક્સ્ટ ઉમેરવા માટે કે જે આકારથી સ્વતંત્ર રીતે આગળ વધે છે, ટેક્સ્ટ બોક્સ ઉમેરો, અને પછી લખાણ લખો અથવા પેસ્ટ કરો.

ટેક્સ્ટ બોક્સ ઉમેરો

a Insert ટેબ પર, ટેક્સ્ટ જૂથમાં, ટેક્સ્ટ બોક્સ પર ક્લિક કરો.

નોંધ: જો તમે પૂર્વ એશિયાઈ ભાષાનો ઉપયોગ કરી રહ્યાં છો, તો ટેક્સ્ટ બોક્સ ડ્રોપ ડાઉન મેનૂમાંથી, ક્યાં તો આડી અથવા ઊભી ગોઠવણી પર ક્લિક કરો.

b સ્લાઈડ પર ક્લિક કરો, અને પછી ટેક્સ્ટ બોક્સ દોરવા માટે પોઈન્ટરને ખેંચો.

લેઆઉટ, રંગ અથવા શૈલી બદલો

SmartArt પસંદ કરો. સ્માર્ટઆર્ટ ટૂલ્સ > ડિઝાઈન ટેબ પસંદ કરો. લેઆઉટનું પૂર્વાવલોકન કરવા માટે તેના પર હોવર કરો અને તમને જોઈતું હોય તે પસંદ કરો. રંગો બદલો પસંદ કરો, તેનું પૂર્વાવલોકન કરવા માટે વિકલ્પો પર હોવર કરો અને તમને જોઈતો એક પસંદ કરો.

તમે સ્માર્ટ આર્ટને કેવી રીતે ફોર્મેટ કરો છો?



Smart Art ફોર્મેટ કરો

તમે ફોર્મેટ કરવા માંગો છો તે SmartArt આકાર પર ક્લિક કરો. સ્માર્ટઆર્ટ ટૂલ્સ હેઠળ ફોર્મેટ ટેબ પર ક્લિક કરો. દરેક આકારને ફોર્મેટ કરવા માટે આકાર શૈલી જૂથમાં વિકલ્પોનો ઉપયોગ કરો.

તમે સ્માર્ટઆર્ટ ગ્રાફિક કેવી રીતે દાખલ અને ફોર્મેટ કરશો?



સ્માર્ટઆર્ટ ગ્રાફિક દાખલ કરો અને તેમાં ટેક્સ્ટ ઉમેરો

Insert મેનુ પર, SmartArt Graphic પસંદ કરો. રિબનના SmartArt ટેબ પર, Insert SmartArt Graphic જૂથમાં, તમને જોઈતા ગ્રાફિકના પ્રકાર પર પસંદ કરો (સૂચિ, પ્રક્રિયા, વગેરે), અને પછી લેઆઉટ પસંદ કરો.

હું પાવરપોઈન્ટમાં સ્માર્ટ આર્ટનો આકાર કેવી રીતે બદલી શકું?



સ્માર્ટઆર્ટ ટૂલ્સ હેઠળ, ડિઝાઇન ટેબ પર, રીસેટ જૂથમાં, કન્વર્ટ પર ક્લિક કરો અને પછી કન્વર્ટ ટુ શેપ્સ પર ક્લિક કરો. જો તમને સ્માર્ટઆર્ટ ટૂલ્સ અથવા ડિઝાઇન ટેબ્સ દેખાતા નથી, તો ખાતરી કરો કે તમે સ્માર્ટઆર્ટ ગ્રાફિક પસંદ કર્યું છે. ડિઝાઇન ટેબ ખોલવા માટે તમારે આકાર પર ડબલ-ક્લિક કરવું પડશે.

પાવરપોઇન્ટમાં સ્માર્ટ આકારનો ઉપયોગ શું છે ઉદાહરણ સાથે સમજાવો?



સ્માર્ટઆર્ટ એ એક ગતિશીલ પ્રકારની ઇમેજ છે જે તમે પાવરપોઇન્ટ સ્લાઇડ્સમાં વારંવાર જોશો. SmartArt નો ઉપયોગ માહિતીની યાદીઓને એકસાથે જૂથ કરવા માટે, ઉદાહરણ તરીકે બુલેટ લિસ્ટ દ્વારા અથવા ચક્ર દ્વારા પ્રક્રિયા બતાવવા માટે થઈ શકે છે.

સ્માર્ટઆર્ટ ગ્રાફિકના કયા પ્રકારો છે?

લેઆઉટ પ્રકારો. સ્માર્ટઆર્ટ ગ્રાફિક પસંદ કરો ગેલેરી અગિયાર વિવિધ પ્રકારોમાં વિભાજિત તમામ ઉપલબ્ધ લેઆઉટ પ્રદર્શિત કરે છે - બધા, સૂચિ, પ્રક્રિયા, ચક્ર, વંશવેલો, સંબંધ, મેટ્રિક્સ, પિરામિડ, ચિત્ર, Office.com અને અન્ય



પાવરપોઇન્ટમાં સ્માર્ટઆર્ટ શા માટે મહત્વપૂર્ણ છે?

સ્માર્ટ આર્ટ સંસ્થા, વંશવેલો, તબક્કાઓ અને પ્રક્રિયાઓ બનાવવામાં મદદ કરે છે. તે દૃષ્ટિથી આનંદદાયક રંગો અને આકૃતિઓનો પરિચય આપે છે અને વક્તા જે કહે છે તે મૂલ્ય ઉમેરે છે.

જરૂરિયાતો અને પ્રકારની માહિતીના આધારે, સ્માર્ટઆર્ટના ઘણા પ્રકારો છે જેનો ઉપયોગ વ્યક્તિ કરી શકે છે.

ઓડિઓ અને વિડિયો તત્વો (Audio & Video Elements)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- PPP માં ઓડિઓ ઉમેરો અથવા કાઢી નાખો
- ઓડિઓ રેકોર્ડ કરો
- પ્લેબેક વિકલ્પો બદલો.

તમારી PowerPoint પ્રસ્તુતિમાં ઓડિઓ ઉમેરો અથવા કાઢી

નાખો તમે તમારી PowerPoint પ્રસ્તુતિમાં ઓડિઓ ઉમેરી શકો છો, જેમ કે સંગીત, વર્ણન અથવા સાઉન્ડ બાઈટ્સ. કોઈપણ ઓડિઓ રેકોર્ડ કરવા અને સાંભળવા માટે, તમારું કમ્પ્યુટર સાઉન્ડ કાર્ડ, માઈક્રોફોન અને સ્પીકર્સથી સજ્જ હોવું આવશ્યક છે.

તમારા PC માંથી ઓડિઓ ઉમેરો

- 1 દાખલ કરો > ઓડિઓ પસંદ કરો.
- 2 મારા PC પર ઓડિઓ પસંદ કરો.
- 3 ઓડિઓ દાખલ કરો સંવાદ બોક્સમાં, તમે ઉમેરવા માંગો છો તે ઓડિઓ ફાઈલ પસંદ કરો.
- 4 દાખલ કરો પસંદ કરો.

ઓડિઓ રેકોર્ડ કરો

- 1 દાખલ કરો > ઓડિઓ પસંદ કરો.
- 2 ઓડિઓ રેકોર્ડ કરો પસંદ કરો.
- 3 તમારી ઓડિઓ ફાઈલ માટે નામ લખો, રેકોર્ડ કરો પસંદ કરો અને પછી બોલો.

નોંધ: ઓડિઓ રેકોર્ડ કરવા માટે તમારા ઉપકરણમાં માઈક્રોફોન સક્ષમ હોવું આવશ્યક છે.

- 4 તમારા રેકોર્ડિંગની સમીક્ષા કરવા માટે, રોકો પસંદ કરો અને પછી પ્લે પસંદ કરો.
- 5 તમારી ક્લિપને ફરીથી રેકોર્ડ કરવા માટે રેકોર્ડ પસંદ કરો અથવા જો તમે સંતુષ્ટ હોવ તો બરાબર પસંદ કરો.
- 6 તમારી ક્લિપને ખસેડવા માટે, ઓડિઓ આઈકોનને પસંદ કરો અને તેને સ્લાઈડ પર જ્યાં ઇચ્છો ત્યાં ખેંચો.

જો તમે સ્લાઈડ દીઠ એક કરતાં વધુ ઓડિઓ ફાઈલનો ઉપયોગ કરી રહ્યાં છો, તો અમે તેને સરળતાથી શોધવા માટે ઓડિઓ આઈકનને સ્લાઈડ પર સમાન સ્થાને મૂકવાની ભલામણ કરીએ છીએ.

- 7 પ્લે પસંદ કરો

પ્લેબેક વિકલ્પો બદલો ઓડિઓ આયકન પસંદ કરો અને પછી ઓડિઓ ટૂલ્સ પ્લેબેક ટેબ પસંદ કરો. પછી તમે કયા વિકલ્પોનો ઉપયોગ કરવા માંગો છો તે પસંદ કરો:

- ઓડિઓને ટ્રિમ કરવા માટે, ટ્રિમ પસંદ કરો અને પછી ઓડિઓ ફાઈલને તે મુજબ ટ્રિમ કરવા માટે લાલ અને લીલા સ્લાઈડર્સનો ઉપયોગ કરો.
- ઓડિઓને ફેડ ઇન અથવા ફેડ આઉટ કરવા માટે, ફેડ ડ્યુરેશન બોક્સમાં નંબર બદલો.
- વોલ્યુમ સમાયોજિત કરવા માટે, વોલ્યુમ પસંદ કરો અને તમને પસંદ હોય તે સેટિંગ પસંદ કરો.
- ઓડિઓ ફાઈલ કેવી રીતે શરૂ થાય તે પસંદ કરવા માટે, ડ્રોપડાઉન એરો પસંદ કરો અને વિકલ્પ પસંદ કરો:
- ક્લિક સિક્વન્સમાં: એક ક્લિક સાથે ઓડિઓ ફાઈલ ઓટોમેટિક રીતે ચલાવે છે.
- ઓટોમેટિક રીતે: ઓડિઓ ફાઈલ ચાલુ હોય તે સ્લાઈડ પર તમે આગળ વધો એટલે ઓટોમેટિક રીતે વગાડે છે.
- જ્યારે ક્લિક કરવામાં આવે ત્યારે: જ્યારે આયકન પર ક્લિક કરવામાં આવે ત્યારે જ ઓડિઓ વગાડે છે.
- તમારી પ્રસ્તુતિમાં ઓડિઓ કેવી રીતે ચાલે તે પસંદ કરવા માટે, એક વિકલ્પ પસંદ કરો:
- સ્લાઈડ્સ પર ચલાવો: બધી સ્લાઈડ્સમાં એક ઓડિઓ ફાઈલ ચલાવે છે.
- બંધ ન થાય ત્યાં સુધી લૂપ: પ્લે/પોઝ બટનને ક્લિક કરીને મેન્યુઅલી બંધ ન થાય ત્યાં સુધી લૂપ પર ઓડિઓ ફાઈલ ચલાવે છે.
- ઓડિઓ સતત ચાલુ રાખવા માટે પૃષ્ઠભૂમિમાં બધી સ્લાઈડ્સ, પૃષ્ઠભૂમિમાં રમો પસંદ કરો.

ઓડિઓ ડિલીટ

કરો ઓડિઓ ક્લિપ ડિલીટ કરવા માટે, સ્લાઈડ પર ઓડિઓ આઈકન પસંદ કરો અને ડિલીટ દબાવો.

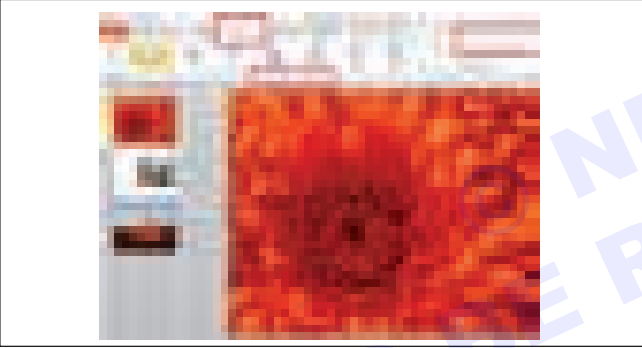
ઓડિઓ અને વિડિઓ તત્વો (Add slide transitions & animations)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સ્લાઇડ સંક્રમણ અવાજ સેટ કરો
- સ્લાઇડ સંક્રમણ અસરો લાગુ કરો
- કસ્ટમ એનિમેશન અસર લાગુ કરો.

સ્લાઇડ ટ્રાન્ઝિશન સાઉન્ડ કેવી રીતે સેટ કરવો

- પ્રથમ, તમે સંક્રમણ અવાજ ઉમેરવા માંગો છો તે સ્લાઇડ પસંદ કરો
- સંક્રમણ ટેબ પસંદ કરો
- આ સ્લાઇડ જૂથમાં સંક્રમણમાં ટ્રાન્ઝિશન સાઉન્ડ વિકલ્પની બાજુમાં ડ્રોપડાઉન ઓરો પર ક્લિક કરો
- વિવિધ સંક્રમણ અવાજોની સૂચિ દેખાય છે
- તમારો ઇચ્છિત સંક્રમણ અવાજ પસંદ કરો
- તમામ સ્લાઇડ્સ પર અસર લાગુ કરવા માટે બધાને લાગુ કરો પર ક્લિક કરો

**સ્લાઇડ ટ્રાન્ઝિશન ઇફેક્ટ્સ કેવી રીતે લાગુ કરવી**

જ્યારે સ્લાઇડ શોમાં એક સ્લાઇડ આગલી સ્લાઇડમાં બદલાય છે ત્યારે સંક્રમણ અસરો સામાન્ય રીતે દેખાય છે.

- પ્રથમ, તમે જે સ્લાઇડ પર અસર લાગુ કરવા માંગો છો તેને પસંદ કરો
- એનિમેશન ટેબ પસંદ કરો
- હવે, આ સ્લાઇડ જૂથમાં સંક્રમણમાં તમે સંક્રમણ અસરો જોશો સંક્રમણ અસરોનું
- મેનૂ જોવા માટે ડ્રોપ-ડાઉન ઓરો પર ક્લિક કરો

- તમારી ઇચ્છિત સંક્રમણ અસર પસંદ કરો
- તમામ સ્લાઇડ્સ પર અસર લાગુ કરવા માટે બધાને લાગુ કરો પર ક્લિક કરો

**કસ્ટમ એનિમેશન ઇફેક્ટ કેવી રીતે લાગુ કરવી**

- પ્રથમ, જરૂરી ટેક્સ્ટ અથવા ઓબ્જેક્ટ પસંદ કરો
- પછી, એનિમેશન ટેબ પસંદ કરો
- એનિમેશન ગ્રુપમાં કસ્ટમ એનિમેશન પર ક્લિક કરો
- કસ્ટમ એનિમેશન ટાસ્ક પેન હવે જમણી બાજુએ દેખાય છે
- અસર ઉમેરો પર ક્લિક કરો, તે વધુ ચાર વિકલ્પો પ્રદર્શિત કરશે જે આગળ વિવિધ વિકલ્પોમાં વિભાજિત છે
- તમારી ઇચ્છિત અસર પસંદ કરો.



ઓડિઓ અને વિડિયો તત્વો (Add and manage comments)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- આદેશ ઉમેરો
- આદેશ બતાવો અથવા છુપાવો
- આદેશ સંપાદિત કરો અને કાઢી નાખો.

પ્રેઝન્ટેશનમાં ટિપ્પણીઓ ઉમેરો, બદલો, છુપાવો અથવા કાઢી નાખો

જ્યારે તમે ઈચ્છો કે લોકો તમે બનાવેલ પ્રેઝન્ટેશનની સમીક્ષા કરે અને પ્રતિસાદ આપે અથવા જ્યારે સહકર્મીઓ પ્રેઝન્ટેશન પર તમારો પ્રતિસાદ પૂછે ત્યારે ટિપ્પણીઓનો ઉપયોગ કરો. ટિપ્પણી એ એક નોંધ છે જેને તમે સ્લાઈડ પરના અક્ષર અથવા શબ્દ સાથે અથવા આખી સ્લાઈડ સાથે જોડી શકો છો.

ટિપ્પણી ઉમેરો

- 1 તમે જે ઓબ્જેક્ટ અથવા સ્લાઈડ પર ટિપ્પણી કરવા માંગો છો તે પસંદ કરો. સમીક્ષા > નવી ટિપ્પણી પસંદ કરો. અથવા જો ટિપ્પણીઓ પેન ખુલ્લું હોય તો નવું પસંદ કરો. તમે Insert > Comment પસંદ કરીને પણ ટિપ્પણી ઉમેરી શકો છો.
- 2 ટિપ્પણીઓ ફલકમાં, બોક્સમાં તમારો સંદેશ લખો અને પોસ્ટ પસંદ કરો અથવા Ctrl+Enter દબાવો.

ટિપ્પણીઓ બતાવો અથવા છુપાવો

- 1 વ્યુ ટેબ પર, સામાન્ય પર ક્લિક કરો.
- 2 સમીક્ષા ટેબ પર, ટિપ્પણીઓ બતાવો પર

ક્લિક કરો અને ટિપ્પણીઓનો જવાબ આપો

- 1 ટિપ્પણીઓ ફલકમાં ટિપ્પણી પસંદ કરો.
- 2 ટિપ્પણીનો જવાબ આપવા માટે જવાબ પસંદ કરો.

ટિપ્પણીઓ સંપાદિત કરો

- 1 નેવિગેશન ફલકમાં, સામાન્ય દૃશ્યમાં, તમે જે સ્લાઈડ પર ટિપ્પણી કરવા માંગો છો તેને ક્લિક કરો.
- 2 ટિપ્પણીઓ ફલકમાં, તમે જે ટિપ્પણીને સંપાદિત કરવા માંગો છો તેને ક્લિક કરો અને પછી પેન્સિલ આયકન પર ક્લિક કરો ફેરફારો કરો.

નોંધ: ધ્યાનમાં રાખો કે અન્ય લોકો માટે તમારી ટિપ્પણીઓ સંપાદિત કરવી શક્ય છે. ઓફિસ દસ્તાવેજમાંની ટિપ્પણીઓ ફાઈલમાં સંગ્રહિત થાય છે, તેથી તમારી ફાઈલમાં ફેરફાર કરવાની એક્સેસ ધરાવનાર કોઈપણ તમારી ટિપ્પણીને સંપાદિત કરી શકે છે.

ટિપ્પણી કાઢી નાખો

ટિપ્પણીઓ ફલકમાં, તમે કાઢી નાખવા માંગો છો તે ટિપ્પણી પસંદ કરો, વધુ શ્રેડ ક્રિયાઓ (...) પસંદ કરો અને શ્રેડ કાઢી નાખો પસંદ કરો.

MySQL માં ડેટાબેઝ ઇન્સ્ટોલ કરો, મુશ્કેલીનિવારણ કરો, બનાવો અને ઉપયોગ કરો (Install, Troubleshoot, Create and Use of database in MySQL)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- DBMS અને RDBMS નો ખ્યાલ
- DBMS અને RDBMS નો તફાવત
- ડેટા મોડલ્સ વ્યાખ્યાયિત કરો
- DBA નો ખ્યાલ
- ડેટાબેઝ વપરાશકર્તાઓને વ્યાખ્યાયિત કરો
- ડેટાબેઝ સ્કીમા વ્યાખ્યાયિત કરો

DBMS અને RDBMS ની વિભાવનાઓ રિલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (RDBMS) એ પ્રોગ્રામ્સ અને ક્ષમતાઓનો સંગ્રહ છે જે IT ટીમો અને અન્યને રિલેશનલ ડેટાબેઝ બનાવવા, અપડેટ કરવા, વહીવટ કરવા અને અન્યથા ક્રિયાપ્રતિક્રિયા કરવા સક્ષમ બનાવે છે. RDBMSes ડેટાબેઝને એક્સેસ કરવા માટે સ્ટ્રક્ચર્ડ ક્વેરી લેંગ્વેજ (SQL) નો ઉપયોગ કરીને મોટાભાગની કોમર્શિયલ રિલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ્સ સાથે કોષ્ટકોના રૂપમાં ડેટા સ્ટોર કરે છે. જો કે, રિલેશનલ મોડલના પ્રારંભિક વિકાસ પછી SQL ની શોધ કરવામાં આવી હોવાથી, તે RDBMS ના ઉપયોગ માટે જરૂરી નથી.

RDBMS એ વિશ્વભરની સંસ્થાઓમાં સૌથી લોકપ્રિય ડેટાબેઝ સિસ્ટમ છે. તે મોટી માત્રામાં ડેટા સંગ્રહિત અને પુનઃપ્રાપ્ત કરવાની વિશ્વસનીય પદ્ધતિ પ્રદાન કરે છે જ્યારે સિસ્ટમ પ્રદર્શન અને અમલીકરણની સરળતાના સંયોજનની ઓફર કરે છે.

RDBMS વિ. DBMS

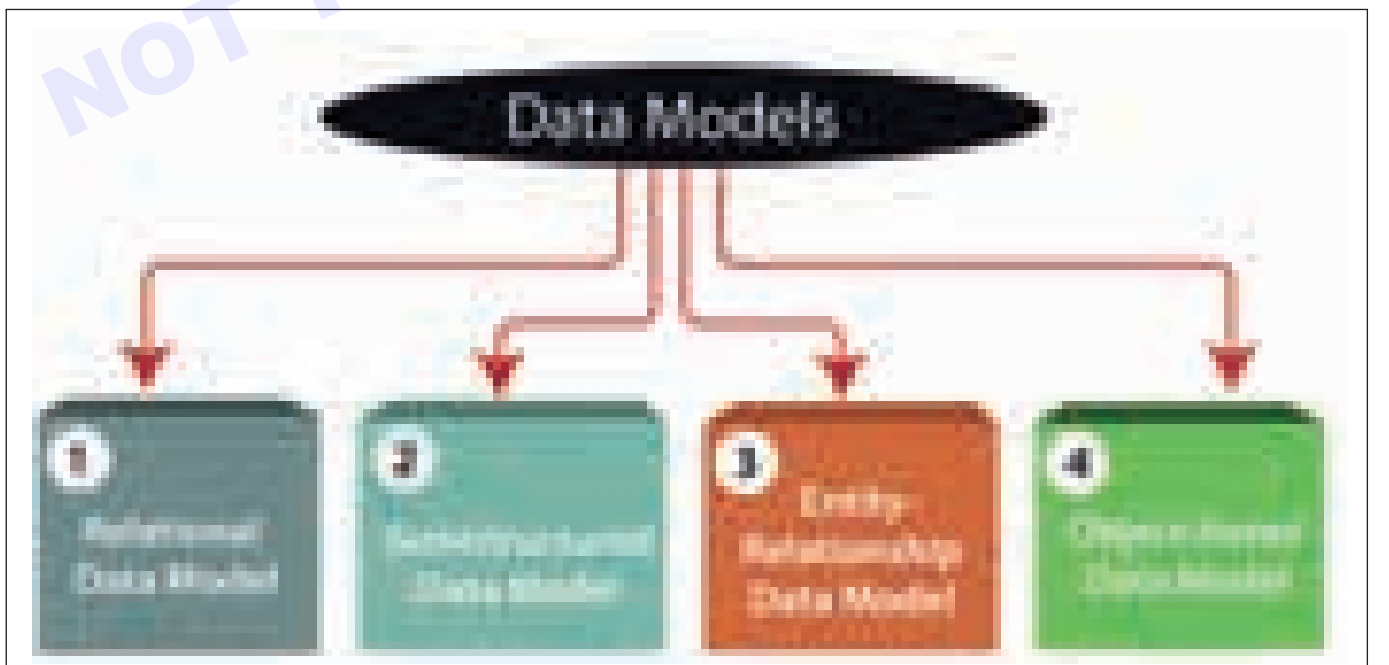
સામાન્ય રીતે, ડેટાબેસેસ ડેટાના સેટ્સનો સંગ્રહ કરે છે જેને અન્ય એપ્લિકેશન્સમાં ઉપયોગ માટે પૂછી શકાય છે. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ ડેટાબેઝ પ્લેટફોર્મના વિકાસ, વહીવટ અને ઉપયોગને સમર્થન આપે છે. આરડીબીએમએસ એ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (ડીબીએમએસ)નો

એક પ્રકાર છે જે ડેટાને પંક્તિ-આધારિત કોષ્ટક માળખામાં સંગ્રહિત કરે છે જે સંબંધિત ડેટા ઘટકોને જોડે છે. આરડીબીએમએસમાં એવા કાર્યોનો સમાવેશ થાય છે જે ડેટાની સુરક્ષા, ચોકસાઈ, અખંડિતતા અને સુસંગતતા જાળવી રાખે છે. આ DBMS માં ઉપયોગમાં લેવાતા ફાઈલ સ્ટોરેજ કરતા અલગ છે.

ડેટા મોડલ્સ

ડેટા મોડલ એ ડેટા વર્ણન, ડેટા સિમેન્ટિક્સ અને ડેટાની સુસંગતતાની મર્યાદાઓનું મોડેલિંગ છે. તે ડેટા એબ્સ્ટ્રેક્શનના દરેક સ્તરે ડેટાબેઝની ડિઝાઈનનું વર્ણન કરવા માટે વૈચારિક સાધનો પૂરા પાડે છે. તેથી, ડેટાબેઝની રચનાને સમજવા માટે નીચેના ચાર ડેટા મોડલનો ઉપયોગ થાય છે:

- 1 **રિલેશનલ ડેટા મોડલ** : આ પ્રકારનું મોડલ કોષ્ટકની અંદર પંક્તિઓ અને કૉલમના રૂપમાં ડેટાને ડિઝાઈન કરે છે. આમ, રિલેશનલ મોડલ ડેટાનું પ્રતિનિધિત્વ કરવા અને સંબંધો વચ્ચે કોષ્ટકોનો ઉપયોગ કરે છે. કોષ્ટકોને સંબંધો પણ કહેવામાં આવે છે. આ મોડલને શરૂઆતમાં એડગર એફ. કોડ દ્વારા 1969માં વર્ણવવામાં આવ્યું હતું. રિલેશનલ ડેટા મોડલ વ્યાપકપણે ઉપયોગમાં લેવાતું મોડેલ છે જે મુખ્યત્વે વાણિજ્યિક ડેટા પ્રોસેસિંગ એપ્લિકેશન્સ દ્વારા ઉપયોગમાં લેવાય છે.



2 **એન્ટિટી રિલેશનશિપ ડેટા મોડલ**: ER મોડલ એ ઑબ્જેક્ટ્સ અને તેમની વચ્ચેના સંબંધો તરીકે ડેટાની તાર્કિક રજૂઆત છે. આ વસ્તુઓને એન્ટિટી તરીકે ઓળખવામાં આવે છે, અને સંબંધ એ આ સંસ્થાઓ વચ્ચેનું જોડાણ છે. આ મોડેલ પીટર ચેન દ્વારા ડિઝાઇન કરવામાં આવ્યું હતું અને 1976 પેપર્સમાં પ્રકાશિત થયું હતું. ડેટાબેઝ ડિઝાઇનિંગમાં તેનો વ્યાપક ઉપયોગ થતો હતો. લક્ષણોનો સમૂહ એકમોનું વર્ણન કરે છે. ઉદાહરણ તરીકે, student_name, student_id ‘વિદ્યાર્થી’ એન્ટિટીનું વર્ણન કરે છે. સમાન પ્રકારની એન્ટિટીનો સમૂહ ‘એન્ટિટી સેટ’ તરીકે ઓળખાય છે, અને સમાન પ્રકારના સંબંધોના સમૂહને ‘રિલેશનશિપ સેટ’ તરીકે ઓળખવામાં આવે છે.

3 **ઑબ્જેક્ટ-આધારિત ડેટા મોડલ** : ફંક્શનલ, એન્ટ્રેપ્રાયુલેશન અને ઑબ્જેક્ટ ઓળખની કલ્પના સાથે ER મોડેલનું વિસ્તરણ. આ મોડેલ સમૃદ્ધ પ્રકારની સિસ્ટમને સમર્થન આપે છે જેમાં માળખાગત અને સંગ્રહ પ્રકારોનો સમાવેશ થાય છે. આમ, 1980 ના દાયકામાં, ઑબ્જેક્ટ-ઓરિએન્ટેડ અભિગમને અનુસરીને વિવિધ ડેટાબેઝ સિસ્ટમ્સ વિકસાવવામાં આવી હતી. અહીં, ઑબ્જેક્ટ્સ તેના ગુણધર્મો વહન કરતા ડેટા સિવાય બીજું કંઈ નથી.

4 **સેમીસ્ટ્રક્ચર્ડ ડેટા મોડલ** : આ ડેટા મોડેલનો પ્રકાર અન્ય ત્રણ ડેટા મોડલથી અલગ છે (ઉપર સમજાવેલ). અર્ધસંરચિત ડેટા મોડેલ એવા સ્થાનો પર ડેટા વિશિષ્ટતાઓને મંજૂરી આપે છે જ્યાં એક જ પ્રકારની વ્યક્તિગત ડેટા આઈટમ્સમાં વિવિધ વિશેષતાઓ સેટ હોઈ શકે છે. એક્સટેન્સિબલ માર્કઅપ લેંગ્વેજ, જેને XML તરીકે પણ ઓળખવામાં આવે છે, તેનો ઉપયોગ અર્ધસંરચિત ડેટાને રજૂ કરવા માટે વ્યાપકપણે થાય છે. XML શરૂઆતમાં ટેક્સ્ટ ડોક્યુમેન્ટમાં માર્કઅપ માહિતીનો સમાવેશ કરવા માટે ડિઝાઇન કરવામાં આવ્યું હતું, તેમ છતાં, ડેટાના વિનિમયમાં તેના ઉપયોગને કારણે તે મહત્વ મેળવે છે.

DBA ની વિભાવના

ઘણા વિવિધ પ્રકારના ડેટાબેઝ છે જે એક બીજાથી સહેજ અલગ છે. જ્યારે DBA એ સામાન્ય ડેટાબેઝ એડમિનિસ્ટ્રેટર્સ છે જેઓ મોટાભાગની વસ્તુઓને કેવી રીતે હેન્ડલ કરે છે તે જાણે છે, ત્યાં DBAs છે જેઓ ચોક્કસ ડેટાબેઝનું સંચાલન કરવામાં અનુભવી છે. MySQL DBA એ ડેટાબેઝ એડમિનિસ્ટ્રેટર છે જે MySQL ડેટાબેસેસનું સંચાલન કરવામાં નિષ્ણાત છે.

MySQL એ આજના મોટા ડેટા ઇકોસિસ્ટમમાં સૌથી વધુ જાણીતી, વ્યાપકપણે ઉપયોગમાં લેવાતી અને કાર્યક્ષમ તકનીકોમાંની એક છે. એન્ટરપ્રાઇઝ ડેટા અથવા સામાન્ય IT સાથે સંકળાયેલા કોઈપણ માટે MySQL નું મૂળભૂત જ્ઞાન આવશ્યક છે, અને જેઓ રિલેશનલ સિસ્ટમ્સથી અજાણ છે તેઓ પણ MySQL નો ઉપયોગ કરીને ડેટા સ્ટોરેજ સિસ્ટમને ઝડપથી કિકસ્ટાર્ટ કરી શકે છે. આ ડેટાબેઝનો ઓપન સોર્સ અને રિલેશનલ નેચર તેના સૌથી મોટા ફાયદા છે, જે મોટા અને નાના બંને બિઝનેસમાં તેના ઉપયોગનું કારણ છે.

એવું કહેવામાં આવે છે, અને તેમ છતાં MySQL ડેટાબેઝ સારી રીતે દસ્તાવેજીકૃત માનવામાં આવે છે અને તેને સમર્થન આપવા માટે એક વિશાળ સમુદાય ધરાવે છે, તેમ છતાં ઉપયોગ કરતી વખતે કેટલીક બાબતો ધ્યાનમાં લેવાની જરૂર છે. MySQL ડેટાબેઝ.

ડેટાબેઝ વપરાશકર્તાઓ : કેટલીકવાર તમે MySQL માં ડેટાબેઝનું સંચાલન કરવા માંગો છો. તે કિસ્સામાં, આપણે ડેટાબેઝમાં બધા વપરાશકર્તાના એકાઉન્ટ્સની સૂચિ જોવાની જરૂર છે. મોટાભાગે, અમે ધારીએ છીએ કે ડેટાબેઝ સર્વરમાં ઉપલબ્ધ તમામ વપરાશકર્તાઓની સૂચિ પ્રદર્શિત કરવા માટે ડેટાબેઝ બતાવો, કોષ્ટકો બતાવો, વગેરે જેવા SHOW USERS આદેશ છે. કમનસીબે, MySQL ડેટાબેઝ પાસે MySQL સર્વરમાં બધા વપરાશકર્તાઓની સૂચિ પ્રદર્શિત કરવા માટે SHOW USERS આદેશ નથી. ડેટાબેઝ સર્વરમાં તમામ યુઝરની યાદી જોવા માટે અમે નીચેની ક્વેરીનો

ઉપયોગ કરી શકીએ છીએ: mysql>mysql.userમાંથી વપરાશકર્તા પસંદ કરો; ઉપરોક્ત નિવેદનના સફળ અમલ પછી, અમે MySQL ડેટાબેઝ સર્વરના વપરાશકર્તા કોષ્ટકમાંથી વપરાશકર્તા ડેટા મેળવીશું.

ડેટાબેઝ સ્કીમા

mysql સ્કીમા એ સિસ્ટમ સ્કીમા છે. તેમાં કોષ્ટકો છે જે MySQL સર્વર દ્વારા ચાલતી વખતે જરૂરી માહિતીનો સંગ્રહ કરે છે. એક વ્યાપક વર્ગીકરણ એ છે કે mysql સ્કીમામાં ડેટા ડિક્શનરી કોષ્ટકો છે જે ડેટાબેઝ ઑબ્જેક્ટ મેટાડેટાને સંગ્રહિત કરે છે અને અન્ય ઓપરેશનલ હેતુઓ માટે ઉપયોગમાં લેવાતા સિસ્ટમ કોષ્ટકો ધરાવે છે. આ નીચેની યર્થા સિસ્ટમ કોષ્ટકોના સમૂહને નાની શ્રેણીઓમાં પેટાવિભાજિત કરે છે.

- ડેટા ડિક્શનરી કોષ્ટકો
- સિસ્ટમ કોષ્ટકો આપો
- ઑબ્જેક્ટ માહિતી સિસ્ટમ કોષ્ટકો
- લોગ સિસ્ટમ કોષ્ટકો
- સર્વર-સાઈડ હેલ્પ સિસ્ટમ કોષ્ટકો
- ટાઈમ ઝોન સિસ્ટમ કોષ્ટકો
- પ્રતિકૃતિ સિસ્ટમ કોષ્ટકો
- ઓપ્ટિમાઇઝર સિસ્ટમ કોષ્ટકો
- પરચુરણ સિસ્ટમ કોષ્ટકો

આ વિભાગનો બાકીનો ભાગ વધારાની માહિતી માટે કોસ સંદર્ભો સાથે દરેક શ્રેણીમાં કોષ્ટકોની ગણતરી કરે છે. ડેટા ડિક્શનરી કોષ્ટકો અને સિસ્ટમ કોષ્ટકો InnoDB સ્ટોરેજ એન્જિનનો ઉપયોગ કરે છે સિવાય કે અન્યથા સૂચવવામાં આવે.

mysql સિસ્ટમ કોષ્ટકો અને ડેટા શબ્દકોશ કોષ્ટકો MySQL ડેટા ડિરેક્ટરીમાં mysql.libd નામની સિંગલ InnoDB ટેબલસ્પેસ ફાઈલમાં રહે છે. પહેલાં, આ કોષ્ટકો mysql ડેટાબેઝ ડિરેક્ટરીમાં વ્યક્તિગત ટેબલસ્પેસ ફાઈલોમાં બનાવવામાં આવ્યાં હતાં.

ડેટા-એટ-રેસ્ટ એન્જિનને mysql સિસ્ટમ સ્કીમા ટેબલસ્પેસ માટે સક્ષમ કરી શકાય છે. વધુ માહિતી માટે, વિભાગ 15.13, “InnoDB ડેટા-એટ રેસ્ટ એન્જિન” જુઓ.

COPA - MySQL પર પ્રદર્શન કરો

નોર્મલાઇઝેશન નિયમો, વિવિધ ડેટાટાઇપ્સ, ડેટા ઇન્ટિગ્રિટી, કીકીએલ, કીએમએલ અને કીસીએલ સ્ટેટમેન્ટ્સનો ઉપયોગ કરીને ડેટાબેઝ ડિઝાઇન કરવી પ્રાથમિક કી અને વિદેશી કી (Designing database using normalization rules, various datatypes, data integrity, DDL, DML & DCL Statements Enforcing Primary key and Foreign key)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- MySQL માં ડેટાબેઝ નોર્મલાઇઝેશન
- વિવિધ ડેટાટાઇપ્સ
- માહિતી સંકલિતતા
- MySQL માં DDL, DML અને DCL સ્ટેટમેન્ટ્સ ડેટાબેઝ નોર્મલાઇઝેશન
- પ્રાથમિક કી અને વિદેશી કી લાગુ કરવી
- સૂચકાંકો ઉમેરી રહ્યા છે.

MySQL માં ડેટાબેઝ નોર્મલાઇઝેશન

ઝેશન એ ડેટાબેઝ ડિઝાઇન અથવા ડેટા મોડેલિંગમાં સૌથી મહત્વપૂર્ણ પરિબળ છે. ડેટાબેઝ નોર્મલાઇઝેશન એ ડેટા રિડન્ડન્સીને દૂર કરવાની અને ડેટા મેનેજમેન્ટને સરળ બનાવવા માટે ડેટાને તાર્કિક રીતે સ્ટોર કરવાની પ્રક્રિયા છે. ડેટાબેઝ નોર્મલાઇઝેશન પ્રક્રિયામાં ડેટાબેઝ સંબંધો અને કીઓ ઉપયોગી છે. ડેટાબેઝ નોર્મલાઇઝેશન E.F.Codd દ્વારા વિકસાવવામાં આવ્યું હતું. ડેટાબેઝ નોર્મલાઇઝેશન પ્રક્રિયામાં, સામાન્ય સ્વરૂપો તરીકે ઓળખાતા નિયમોની શ્રેણી છે.

સામાન્ય સ્વરૂપના મુખ્યત્વે 6 પ્રકાર છે: પ્રથમ સામાન્ય સ્વરૂપ (1NF), બીજું સામાન્ય સ્વરૂપ (2NF), ત્રીજું સામાન્ય સ્વરૂપ (3NF), ચોથું સામાન્ય સ્વરૂપ (4NF), પાંચમું સામાન્ય સ્વરૂપ (5NF), અને બોયસ કોડ સામાન્ય સ્વરૂપ (BCNF). પરંતુ મોટાભાગે અમે અમારા ડેટાબેઝ ડિઝાઇનમાં ત્રીજા સામાન્ય સ્વરૂપનો ઉપયોગ કર્યો છે.

- 1 પ્રથમ સામાન્ય ફોર્મ (1NF): પ્રથમ સામાન્ય સ્વરૂપમાં, દરેક કોલમમાં માત્ર એક જ મૂલ્ય હોવું જોઈએ અને કોઈ કોષ્ટકમાં સંબંધિત ડેટાના પુનરાવર્તિત જૂથોને સંગ્રહિત કરવા જોઈએ નહીં.
- 2 સેકન્ડ નોર્મલ ફોર્મ (2NF): બીજા નોર્મલ ફોર્મમાં, પ્રથમ ડેટાબેઝ પ્રથમમાં હોવો જોઈએ સામાન્ય સ્વરૂપ, તે સમાન કોષ્ટકમાં ડુપ્લિકેટ પંક્તિઓ સંગ્રહિત ન કરવી જોઈએ. અને જો પંક્તિમાં ડુપ્લિકેટ મૂલ્યો હોય, તો તે તેમના પોતાના અલગ કોષ્ટકોમાં સંગ્રહિત થવી જોઈએ અને વિદેશી કીનો ઉપયોગ કરીને કોષ્ટક સાથે લિંક કરવી જોઈએ. બીજા સામાન્ય સ્વરૂપમાં ડેટાબેઝની આદર્શ રીત એ છે કે એકથી અનેક સંબંધ કોષ્ટકો બનાવવી.
- 3 ત્રીજું સામાન્ય ફોર્મ (3NF): ત્રીજા સામાન્ય સ્વરૂપમાં, ડેટાબેઝ પહેલાથી જ ત્રીજા સ્વરૂપમાં હોય છે, જો તે બીજા સામાન્ય સ્વરૂપમાં હોય અને દરેક બિન-કી કોલમ પરસ્પર સ્વતંત્ર હોય. કોષ્ટકમાંના કોઈપણ કોલમ્સને ઓળખો કે જે એકબીજા પર આધારિત છે અને તે કોલમ્સને તેમના પોતાના અલગ કોષ્ટકોમાં તોડો.
- 4 બોયસ કોડ નોર્મલ ફોર્મ (BCNF): તે ત્રીજા સામાન્ય સ્વરૂપનું સર્વોચ્ચ સ્વરૂપ છે જે 3NF દ્વારા નિયંત્રિત થતી નથી તેવા વિવિધ પ્રકારની વિસંગતતાઓ સાથે વ્યવહાર કરે છે.

વિવિધ ડેટા પ્રકારો**ડેટા અખંડિતતા**

ડેટા અખંડિતતા કોઈપણ ડેટાબેઝ અથવા વેરહાઉસમાં સંગ્રહિત ડેટાની સુસંગતતા, ચોકસાઈ અને વિશ્વસનીયતાનો સંદર્ભ આપે છે. તમામ

લાક્ષણિકતાઓ સચોટ ધરાવતું સંપૂર્ણ માળખું ધરાવતો ડેટા કહેવાય છે અખંડિતતા સાથે ડેટા

ડેટાની અખંડિતતાના ઘણા પાસાઓ છે જેમ કે ભૌતિક અખંડિતતા, જેમાં ડેટાને અધિકૃત રીતે સંગ્રહિત કરવા અને એકત્રિત કરવાનો સમાવેશ થાય છે, પછી તાર્કિક અખંડિતતા આવે છે, જેમાં માહિતી ચોક્કસ સંદર્ભમાં સુસંગત અને સચોટ છે કે કેમ તે તપાસવાનો સમાવેશ થાય છે કે એકંદરે તમામ નિયમો જે ડેટાની ગુણવત્તા જાળવવા માટે જરૂરી છે જે ડેટા અખંડિતતા હેઠળ આવે છે. અખંડિતતા અને ચોકસાઈ વિના, તમારો તમામ એકત્રિત ડેટા કંપની માટે નકામો છે, તેથી તેની કામગીરી અને સ્થિરતા વધારવા માટે ડેટા સુરક્ષા અને તેની ચોકસાઈની ખાતરી કરવી જરૂરી છે. દૂષિત ડેટા ક્યારેક તમારા વ્યવસાયને પણ નુકસાન પહોંચાડી શકે છે. ડેટામાં ફેરફાર કરવો સહેલો નથી કારણ કે ડેટા સ્થિર નથી. તમે જે માહિતી મેળવો છો તે વિવિધ સ્થળોએથી આવે છે, અને ઘણી વસ્તુઓ તે બનાવેલ સાઈટ પરથી તમારો ડેટા બદલી શકે છે.

વધુમાં, તેને અન્ય ઉપકરણો પર સ્થાનાંતરિત કરી શકાય છે, કોઈપણ દ્વારા અને જ્યારે પણ જરૂરી હોય ત્યારે તેને બદલી શકાય છે અને અપડેટ કરી શકાય છે. અને ડેટાની અખંડિતતાને સુનિશ્ચિત કરવી એ માત્ર એક-પગલાની પ્રક્રિયા નથી.

તે તપાસવું પડશે અને મોડેલની ડિઝાઇનથી શરૂ કરીને તેના અંતિમ આઉટપુટ સુધી દરેક પગલા પર પુષ્ટિ થાય છે.

DDL, DML અને DCL સ્ટેટમેન્ટ્સ: સ્ટ્રક્ચર્ડ ક્વેરી લેંગ્વેજ (SQL) આપણે બધા જાણીએ છીએ કે ડેટાબેઝ લેંગ્વેજ છે જેના ઉપયોગથી આપણે વર્તમાન ડેટાબેઝ પર ચોક્કસ કામગીરી કરી શકીએ છીએ અને ડેટાબેઝ બનાવવા માટે પણ આ ભાષાનો ઉપયોગ કરી શકીએ છીએ. જરૂરી કાર્યો કરવા માટે એસક્ચુએલ અમુક આદેશોનો ઉપયોગ કરે છે જેમ કે બનાવો, છોડો, દાખલ કરો વગેરે.

આ SQL આદેશો મુખ્યત્વે ચાર શ્રેણીઓમાં વર્ગીકૃત કરવામાં આવે છે:

- 1 DDL - ડેટા વ્યાખ્યા ભાષા
- 2 DML - ડેટા મેનીપ્યુલેશન ભાષા
- 3 DCL - ડેટા કંટ્રોલ લેંગ્વેજ

DDL (ડેટા ડેફિનેશન લેંગ્વેજ)

DDL અથવા ડેટા ડેફિનેશન લેંગ્વેજ વાસ્તવમાં SQL આદેશોનો સમાવેશ કરે છે જેનો ઉપયોગ ડેટાબેઝ સ્કીમાને વ્યાખ્યાયિત કરવા માટે થઈ શકે

છે. તે ફક્ત ડેટાબેઝ સ્કીમાના વર્ણન સાથે વ્યવહાર કરે છે અને ડેટાબેઝમાં ડેટાબેઝ ઓબ્જેક્ટના બંધારણને બનાવવા અને સંશોધિત કરવા માટે વપરાય છે. DDL એ SQL આદેશોનો સમૂહ છે જેનો ઉપયોગ ડેટાબેઝ સ્ટ્રક્ચર્સ બનાવવા, સંશોધિત કરવા અને કાઢી નાખવા માટે થાય છે પરંતુ ડેટા નથી. આ આદેશો સામાન્ય રીતે દ્વારા ઉપયોગમાં લેવાતા નથી એક સામાન્ય વપરાશકર્તા, જે એપ્લિકેશન દ્વારા ડેટાબેઝને એક્સેસ કરતો હોવો જોઈએ.

DDL આદેશોની યાદી

- **CREATE** : આ આદેશનો ઉપયોગ ડેટાબેઝ અથવા તેના ઓબ્જેક્ટ્સ (જેમ કે કોષ્ટક, અનુક્રમણિકા, કાર્ય, દૃશ્યો, સ્ટોર પ્રક્રિયા અને ટ્રિગર્સ) બનાવવા માટે થાય છે.
- **DROP** : આ આદેશનો ઉપયોગ ડેટાબેઝમાંથી વસ્તુઓને કાઢી નાખવા માટે થાય છે.
- **ALTER** : આનો ઉપયોગ ડેટાબેઝના બંધારણને બદલવા માટે થાય છે.
- **ટ્રંકેટ** : આનો ઉપયોગ કોષ્ટકમાંથી તમામ રેકૉર્ડ્સને દૂર કરવા માટે થાય છે, જેમાં રેકૉર્ડ્સ માટે ફાળવવામાં આવેલી તમામ જગ્યાઓ દૂર કરવામાં આવે છે.
- **ટિપ્પણી** : આનો ઉપયોગ ડેટા શબ્દકોશમાં ટિપ્પણીઓ ઉમેરવા માટે થાય છે.
- **RENAME** : આનો ઉપયોગ ડેટાબેઝમાં હાજર ઓબ્જેક્ટનું નામ બદલવા માટે થાય છે.

DML(ડેટા મેનીપ્યુલેશન લેંગ્વેજ)

એસક્યુએલ આદેશો કે જે ડેટાબેઝમાં હાજર ડેટાના મેનીપ્યુલેશન સાથે કામ કરે છે તે ડીએમએલ અથવા ડેટા મેનીપ્યુલેશન લેંગ્વેજના છે અને આમાં મોટાભાગના એસક્યુએલ સ્ટેટમેન્ટનો સમાવેશ થાય છે. તે SQL સ્ટેટમેન્ટનો ઘટક છે જે ડેટા અને ડેટાબેઝની એક્સેસને નિયંત્રિત કરે છે. મૂળભૂત રીતે, DCL નિવેદનો DML નિવેદનો સાથે જૂથબદ્ધ છે.

DML આદેશોની સૂચિ

- **INSERT** : તેનો ઉપયોગ કોષ્ટકમાં ડેટા દાખલ કરવા માટે થાય છે.
- **અપડેટ** : તેનો ઉપયોગ અપડેટ કરવા માટે થાય છે કોષ્ટકની અંદર હાલનો ડેટા.
- **DELETE** : તેનો ઉપયોગ ડેટાબેઝ ટેબલમાંથી રેકૉર્ડ કાઢી નાખવા માટે થાય છે.
- **લોક** : ટેબલ કન્ટ્રોલ કન્કરન્સી.
- **કોલ કરો** : PL/SQL અથવા JAVA સબપ્રોગ્રામ પર કોલ કરો.

- **યોજના સમજાવો** : તે ડેટાના એક્સેસ પાથનું વર્ણન કરે છે.

ડીસીએલ (ડેટા કન્ટ્રોલ લેંગ્વેજ)

ડીસીએલમાં ગ્રાન્ટ અને રિવોક જેવા આદેશોનો સમાવેશ થાય છે જે મુખ્યત્વે ડેટાબેઝ સિસ્ટમના અધિકારો, પરવાનગીઓ અને અન્ય નિયંત્રણો સાથે વ્યવહાર કરે છે.

DCL આદેશોની યાદી

- **ગ્રાન્ટ** : આ આદેશ વપરાશકર્તાઓને ડેટાબેઝની એક્સેસ વિશેષાધિકારો આપે છે.
- **રદ કરો** : આ આદેશ ગ્રાન્ટ આદેશનો ઉપયોગ કરીને આપવામાં આવેલા વપરાશકર્તાના એક્સેસ વિશેષાધિકારોને પાછો ખેંચી લે છે.

પ્રાથમિક કી પ્રાથમિક કી અવરોધ ટેબલ પર પ્રાથમિક કી બનાવીને વ્યાખ્યાયિત કરવામાં આવે છે. પ્રાથમિક કીમાંનું મૂલ્ય કોષ્ટકમાં દરેક પંક્તિને વિશિષ્ટ રીતે ઓળખતું હોવું જોઈએ. પ્રાથમિક કી અવરોધ પંક્તિ અખંડિતતાને લાગુ કરે છે.

પ્રાથમિક કી વ્યાખ્યાયિત કરતી વખતે નીચેના તથ્યો ધ્યાનમાં લો:

- જ્યારે તમે ટેબલ પર પ્રાથમિક કી બનાવો છો, ત્યારે એક અનન્ય અનુક્રમણિકા (અને તેથી અનન્ય અવરોધ) આપમેળે અને સ્પષ્ટ રીતે બનાવવામાં આવે છે જે આ કોલમમાં ડેટાની વિશિષ્ટતાને લાગુ કરે છે.
- પ્રાથમિક કી કોલમમાં NULL મૂલ્યો હોઈ શકતા નથી. ડેટાબેઝ એન્જીન સ્પષ્ટપણે પ્રાથમિક કી કોલમ પર NOT NULL અવરોધ બનાવે છે.
- ટેબલ પર માત્ર એક પ્રાથમિક કી વ્યાખ્યાયિત કરી શકે છે.
- પ્રાથમિક કી એક કોલમ અથવા બહુવિધ કોલમ (એટલે કે સંયુક્ત પ્રાથમિક કી) ની રચના દ્વારા બનાવી શકાય છે.
- જો તમે પ્રાથમિક કીમાં ડુપ્લિકેટ રેકૉર્ડ દાખલ કરો છો, તો તમને એક મળશે ભૂલ

પ્રાથમિક કી બે પ્રકારની છે: કૃત્રિમ પ્રાથમિક કી અને કુદરતી પ્રાથમિક કી.

ફોરેન કી

MySQL ફોરેન કીઝને સપોર્ટ કરે છે, જે કોષ્ટકોમાં ક્રોસ રેફરન્સિંગ સંબંધિત ડેટાને પરવાનગી આપે છે અને વિદેશી કી અવરોધો, જે સંબંધિત ડેટાને સુસંગત રાખવામાં મદદ કરે છે.

વિદેશી કી સંબંધમાં પિતૃ કોષ્ટકનો સમાવેશ થાય છે જે પ્રારંભિક કોલમ મૂલ્યો ધરાવે છે, અને કોલમ મૂલ્યો સાથે બાળ કોષ્ટક જે પિતૃ કોલમ મૂલ્યોનો સંદર્ભ આપે છે. ચાઈલ્ડ ટેબલ પર વિદેશી કી અવરોધ વ્યાખ્યાયિત થયેલ છે

પ્રશ્નો દાખલ કરો અને કાઢી નાખો ક્વેરીઝ અપડેટ કરો (Insert and delete queries Update queries)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સરળ પસંદગીના પ્રશ્નો.

ટ્રાન્ઝેક્શનના**સૂચકાંકો ક્વેરીઝનો ખ્યાલ ઉમેરવો**

ટ્રાન્ઝેક્શન એ ડેટાબેઝ મેનીપ્યુલેશન ઓપરેશન્સનું ક્રમિક જૂથ છે, જે એક જ કાર્ય એકમ હોય તેમ કરવામાં આવે છે. બીજા શબ્દોમાં કહીએ તો, જ્યાં સુધી જૂથમાં દરેક વ્યક્તિગત કામગીરી સફળ ન થાય ત્યાં સુધી વ્યવહાર ક્યારેય પૂર્ણ થશે નહીં. જો ટ્રાન્ઝેક્શનની અંદર કોઈપણ કામગીરી નિષ્ફળ જાય, તો સમગ્ર વ્યવહાર નિષ્ફળ જશે. વ્યવહારિક રીતે, તમે એક જૂથમાં ઘણી SQL ક્વેરીઝને કલબ કરશો અને તમે વ્યવહારના ભાગ રૂપે તે બધાને એકસાથે એક્ઝિક્યુટ કરશો.

વ્યવહારોની મિલકતો

વ્યવહારોમાં નીચેના ચાર પ્રમાણભૂત ગુણધર્મો હોય છે, જેને સામાન્ય રીતે ટ્રેકાક્ષર ACID દ્વારા ઓળખવામાં આવે છે?

- **એટોમિસિટી** - આ સુનિશ્ચિત કરે છે કે વર્ક યુનિટની અંદરની તમામ કામગીરી સફળતાપૂર્વક પૂર્ણ થાય છે; અન્યથા, ટ્રાન્ઝેક્શન નિષ્ફળતાના બિંદુએ બંધ કરવામાં આવે છે અને અગાઉની કામગીરી તેમની ભૂતપૂર્વ સ્થિતિમાં પાછી ફેરવવામાં આવે છે.
- **સુસંગતતા** - આ ખાતરી કરે છે કે ડેટાબેઝ સફળતાપૂર્વક પ્રતિબદ્ધ ટ્રાન્ઝેક્શન પર રાજ્યોને યોગ્ય રીતે બદલે છે.
- **અલગતા** - આ સક્ષમ કરે છે વ્યવહારો સ્વતંત્ર રીતે ચલાવવા માટે અને એકબીજા સાથે પારદર્શક છે.
- **ટકાઉપણું** - આ સુનિશ્ચિત કરે છે કે સિસ્ટમની નિષ્ફળતાના કિસ્સામાં પ્રતિબદ્ધ વ્યવહારનું પરિણામ અથવા અસર ચાલુ રહે છે

© NIMI
NOT TO BE REPUBLISHED

નંબર, તારીખ અને અક્ષર ફંક્શનનો ઉપયોગ કરીને કાર્યો પર નિદર્શન કરો, પેટા ક્વેરી કરીને જૂથ બનાવો (Using the Number, Date and Character functions, group by having, sub query)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- પેટા ક્વેરી જણાવો.

MySQL જોઈનિંગ કોષ્ટકો

તેમની વચ્ચે સંબંધિત કોલમના આધારે, બે અથવા વધુ કોષ્ટકોમાંથી પંક્તિઓને જોડવા માટે JOIN કલમનો ઉપયોગ કરવામાં આવે છે.

ચાલો “ઓર્ડર્સ” કોષ્ટકમાંથી પસંદગી જોઈએ:

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

પછી, “ગ્રાહકો” કોષ્ટકમાંથી પસંદગી જુઓ:

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

નોંધ લો કે “ઓર્ડર્સ” કોષ્ટકમાં “CustomerID” કોલમ “ગ્રાહકો” કોષ્ટકમાં “CustomerID” નો સંદર્ભ આપે છે. ઉપરોક્ત બે કોષ્ટકો વચ્ચેનો સંબંધ “CustomerID” કોલમ છે.

તે પછી, અમે નીચેનું SQL સ્ટેટમેન્ટ બનાવી શકીએ છીએ (જેમાં એક INNER JOIN હોય છે), જે બંને કોષ્ટકોમાં મેળ ખાતા મૂલ્યો ધરાવતા રેકૉર્ડ પસંદ કરે છે:

ઉદાહરણ

SELECT Orders.OrderID, Customers.CustomerName, Orders.
OrderDate

ઓર્ડર્સથી અંદરના ગ્રાહકો સાથે જોડાઓ ઓર્ડર પર. CustomerID=Customers.CustomerID;

તેને જાતે અજમાવી જુઓ “અને તે આના જેવું કંઈક ઉત્પન્ન કરશે:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

MySQL માં જોડાવાના સપોર્ટેડ પ્રકારો

- આંતરિક જોડો: બંને કોષ્ટકોમાં મેળ ખાતા મૂલ્યો ધરાવતા રેકૉર્ડ્સ પરત કરે છે
- ડાબે જોડાઓ: ડાબા કોષ્ટકમાંથી તમામ રેકૉર્ડ્સ અને જમણા કોષ્ટકમાંથી મેળ ખાતા રેકૉર્ડ્સ પરત કરે છે
- રાઈટ જોઈન: જમણા ટેબલમાંથી તમામ રેકૉર્ડ્સ અને ડાબા ટેબલમાંથી મેળ ખાતા રેકૉર્ડ્સ પરત કરે છે
- ક્રોસ જોઈન: બંને કોષ્ટકોમાંથી તમામ રેકૉર્ડ્સ પરત કરે છે

સબ ક્વેરી

MySQL માં સબક્વેરી એ એક ક્વેરી છે, જે અન્ય SQL ક્વેરી માં નેસ્ટેડ છે અને વિવિધ ઓપરેટરો સાથે SELECT, INSERT, UPDATE અથવા DELETE સ્ટેટમેન્ટ સાથે એમ્બેડેડ છે. આપણે બીજી સબક્વેરી સાથે સબક્વેરી નેસ્ટ પણ કરી શકીએ છીએ. સબક્વેરી આંતરિક ક્વેરી તરીકે ઓળખાય છે અને સબક્વેરી ધરાવતી ક્વેરી બાહ્ય ક્વેરી તરીકે ઓળખાય છે. એક્ઝિક્યુટ કરવામાં આવેલ આંતરિક ક્વેરી પ્રથમ બાહ્ય ક્વેરીનું પરિણામ આપે છે, અને પછી મુખ્ય/ બાહ્ય ક્વેરી કરવામાં આવશે. MySQL અમને ગમે ત્યાં સબક્વેરીનો ઉપયોગ કરવાની પરવાનગી આપે છે, પરંતુ તે કૌંસમાં બંધ હોવી જોઈએ. SQL સ્ટાન્ડર્ડ દ્વારા સમર્થિત તમામ સબક્વેરી ફોર્મ્સ અને ઓપરેશન્સ MySQL માં પણ સપોર્ટેડ હશે.

Fig 1

The figure displays four Venn diagrams, each consisting of two overlapping circles labeled 'TABLE 1' and 'TABLE 2'.
1. **INNER JOIN**: Only the intersection of the two circles is shaded.
2. **LEFT JOIN**: The entire area of 'TABLE 1' (both the non-overlapping part and the intersection) is shaded.
3. **RIGHT JOIN**: The entire area of 'TABLE 2' (both the non-overlapping part and the intersection) is shaded.
4. **CROSS JOIN**: The entire area of both 'TABLE 1' and 'TABLE 2' is shaded, including the intersection.

INNER JOIN

LEFT JOIN

RIGHT JOIN

CROSS JOIN

CP20N1299811

- સબક્વેરીઝનો હમેશા કૌંસમાં ઉપયોગ કરવો જોઈએ.
- જો મુખ્ય ક્વેરી સબક્વેરી માટે બહુવિધ કૉલમ ધરાવતી નથી, તો સબક્વેરી SELECT આદેશમાં માત્ર એક કૉલમ ધરાવી શકે છે.
- અમે સબક્વેરી સાથે વિવિધ સરખામણી ઓપરેટર્સનો ઉપયોગ કરી શકીએ છીએ, જેમ કે >,,<,,=,IN,ANY,SOME,અને ALL. જ્યારે સબક્વેરી એક કરતાં વધુ પંક્તિ પરત કરે છે ત્યારે બહુવિધ-પંક્તિ ઓપરેટર ખૂબ જ ઉપયોગી છે.
- અમે સબક્વેરી માં ORDER BY કલમ નો ઉપયોગ કરી શકતા નથી, જો કે તેનો ઉપયોગ મુખ્ય ક્વેરી માં થઈ શકે છે.
- જો આપણે સેટ ફંક્શનમાં સબક્વેરીનો ઉપયોગ કરીએ, તો તેને સેટ ફંક્શનમાં તરત જ બંધ કરી શકાતું નથી.

- સબકવેરીઝ કવેરીઝને માળખાગત સ્વરૂપમાં બનાવે છે જે અમને સ્ટેટમેન્ટના દરેક ભાગને અલગ કરવાની મંજૂરી આપે છે.
- સબકવેરીઝ ટેબલમાંથી ડેટાની ક્વેરી કરવાની વૈકલ્પિક રીતો પ્રદાન કરે છે; અન્યથા, આપણે જટિલ જોડાણો અને યુનિયનોનો ઉપયોગ કરવાની જરૂર છે.
- સબકવેરીઝ જટિલ જોડાવા અથવા યુનિયન સ્ટેટમેન્ટ કરતાં વધુ વાંચવા યોગ્ય છે.

નીચે મુજબ છે MySQL માં સબક્વેરીનો ઉપયોગ કરવા માટે મૂળભૂત વાક્યરચના છે:

પસંદ કરેલ ક્વેરી સિન્ટેક્સ દ્વારા પરત કરાયેલા રેકૉર્ડ્સની સંખ્યા
પરત કરે છે: COUNT(અભિવ્યક્તિ)

COPA - કમ્પ્યુટર નેટવર્ક સેટ કરો અને ગોઠવો

કમ્પ્યુટરને નેટવર્કથી કનેક્ટ કરો અને ઉપકરણો એટલે કે પ્રિન્ટર્સ, ફાઇલો, ફોલ્ડર્સ અને ડ્રાઇવ્સ શેર કરો (Connect a computer to a network and share Devices i.e. Printers, files, folders and drives)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સ્થાનિક નેટવર્ક્સ.
- સ્થાનિક નેટવર્ક પર વાતચીત કરવી:
- કોમ્યુનિકેશનના સિદ્ધાંતો.

કનેક્ટેડ વિશ્વમાં સંચાર**સ્થાનિક નેટવર્ક્સ.**

લોકલ એરિયા નેટવર્ક (LAN) એ કમ્પ્યુટર્સ અને પેરિફેરલ ઉપકરણોનું જૂથ છે જે એક અલગ ભૌગોલિક વિસ્તારની અંદર સર્વર સાથે સામાન્ય સંચાર લાઈન અથવા વાયરલેસ લિંક શેર કરે છે. લોકલ એરિયા નેટવર્ક હોમ ઓફિસમાં બે કે ત્રણ જેટલા વપરાશકર્તાઓ અથવા કોર્પોરેશનની સેન્ટ્રલ ઓફિસમાં હજારો વપરાશકર્તાઓને સેવા આપી શકે છે.

સ્થાનિક નેટવર્ક પર વાતચીત :

પીઅર-ટુ-પીઅર LAN બે ઉપકરણોને સીધું જ જોડે છે -- સામાન્ય રીતે, વર્કસ્ટેશન અથવા પર્સનલ કમ્પ્યુટર્સ -- એકસાથે ઈથરનેટ કેબલનો ઉપયોગ કરીને. ક્લાયન્ટ-સર્વર LAN માં બહુવિધ એન્ડપોઇન્ટ્સ અને સર્વર્સનો સમાવેશ થાય છે જે LAN સ્વીચ સાથે જોડાયેલા હોય છે. સ્વીચ બહુવિધ કનેક્ટેડ ઉપકરણો વચ્ચે સંચાર સ્ટ્રીમ્સનું નિર્દેશન કરે છે

કોમ્યુનિકેશનના સિદ્ધાંતો

કોમ્યુનિકેશન પ્રોફેશનલ્સ ક્યારેક કોમ્યુનિકેશનના 7 સિદ્ધાંતોનો સંદર્ભ લેશે; દુર્ભાગ્યે, આ પ્રમાણભૂત નથી (તેથી કદાચ સ્તર 200!), પરંતુ કેટલીક સામાન્ય થીમ્સ છે જેનો ઉપયોગ માઈક્રોસોફ્ટની અંદર કોમ્યુનિકેશન કમ્પિટન્સીની જાણ કરવા માટે કરવામાં આવ્યો હતો. 365 પરિપક્વતા મોડલ. જો તમે કોમ્યુનિકેશન પ્રોફેશનલ ન હોવ તો તમને સંચાર વ્યૂહરચના, પરિપક્વતા મૂલ્યાંકન અથવા તકનીકી પસંદગીના ભાગરૂપે સિદ્ધાંતોનું આ અર્થઘટન મદદરૂપ થઈ શકે છે.

- 1 અસરકારક
- 2 વ્યાપક
- 3 સ્પષ્ટતા
- 4 ધ્યાન અને શૈલી
- 5 સુસંગતતા
- 6 સમયસૂચકતા અને તાકીદ
- 7 પ્રતિસાદનું મહત્વ

ઈથરનેટ નેટવર્ક્સ (Ethernet networks)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- ઈથરનેટ વ્યાખ્યાયિત કરો
- ઈથરનેટ કેવી રીતે કામ કરવું
- બિલ્ટ નેટવર્ક વ્યાખ્યાયિત કરો

ઈથરનેટ એ મૂળ નેટવર્કિંગ તકનીકોમાંની એક છે, જેની શોધ લગભગ 50 વર્ષ પહેલાં કરવામાં આવી હતી. અને તેમ છતાં, કોમ્યુનિકેશન પ્રોટોકોલને જમાવટ કરી શકાય તેવી સરળતાને કારણે અને તેની પાછળની સુસંગતતા ગુમાવ્યા વિના આધુનિક પ્રગતિને સમાવિષ્ટ કરવાની ક્ષમતાને કારણે, ઈથરનેટ કમ્પ્યુટર નેટવર્કિંગ માટે વાસ્તવિક ધોરણ તરીકે શાસન કરવાનું ચાલુ રાખે છે.

તેના મૂળમાં, ઈથરનેટ એ એક પ્રોટોકોલ છે જે કમ્પ્યુટર્સને (સર્વરથી લેપટોપ સુધી) વાયર્ડ નેટવર્ક્સ પર એકબીજા સાથે વાત કરવાની મંજૂરી આપે છે જે ટ્રાફિકને ડાયરેક્ટ કરવા માટે રાઉટર્સ, સ્વિચ અને હબ જેવા ઉપકરણોનો ઉપયોગ કરે છે. ઈથરનેટ વાયરલેસ પ્રોટોકોલ સાથે પણ એકીકૃત રીતે કામ કરે છે.

લગભગ કોઈપણ વાતાવરણમાં કામ કરવાની તેની ક્ષમતાને કારણે વિશ્વભરમાં તેને સાર્વત્રિક અપનાવવામાં આવ્યું છે. આ ખાસ કરીને સાચું છે કારણ કે તે સંસ્થાઓને તેમના લોકલ એરિયા નેટવર્ક (LAN) અને તેમના

વાઈડ-એરિયા નેટવર્ક (WAN) માં સમાન ઈથરનેટ પ્રોટોકોલનો ઉપયોગ કરવાની મંજૂરી આપે છે. તેનો અર્થ એ કે તે ડેટા સેન્ટર્સમાં, ખાનગી અથવા આંતરિક કંપની નેટવર્ક્સમાં, ઈન્ટરનેટ એપ્લિકેશન્સ માટે અને લગભગ સારી રીતે કાર્ય કરે છે વચ્ચે કંઈપણ. તે નેટવર્કિંગના સૌથી જટિલ સ્વરૂપોને પણ સમર્થન આપી શકે છે, જેમ કે વર્ચ્યુઅલ પ્રાઇવેટ નેટવર્ક્સ (VPN) અને સોફ્ટવેર નિર્ધારિત નેટવર્કિંગ જમાવટ.

ઈથરનેટને બેન્ડવિડ્થ ઈન્ટેન્સિવ એપ્લિકેશન જેમ કે વિડિયો સ્ટ્રીમિંગ અથવા વોઈસ ઓવર આઈપી એપ્લિકેશનને હેન્ડલ કરવામાં કોઈ સમસ્યા નથી. અને બીજી બાજુએ, તેની સરળતા તેને કોઈપણ વિશિષ્ટ રૂપરેખાકનની આવશ્યકતા વિના, ઈન્ટરનેટ ઓફ થિંગ્સ (IoT) જેવા અત્યંત નાના, પ્રમાણમાં અસંસ્કારી ઉપકરણો સાથે કામ કરવા સક્ષમ બનાવે છે.

ઈથરનેટ કેવી રીતે કામ કરે છે?

ઈથરનેટ પર્સનલ કોમ્પ્યુટર જેવા ઉપકરણો પર અથવા તેમાંથી મોડલવામાં આવતી માહિતીને ફ્રેમ નામની માહિતીના વિવિધ કદના બિટ્સના ટૂંકા

ટુકડાઓમાં વિભાજિત કરીને કામ કરે છે. તે ફ્રેમ્સમાં પ્રમાણિત માહિતી હોય છે જેમ કે સ્રોત અને ગંતવ્ય સરનામું જે ફ્રેમને નેટવર્ક દ્વારા તેના માર્ગમાં મદદ કરે છે.

અને કારણ કે LAN પરના કોમ્પ્યુટરો સામાન્ય રીતે એક જ કનેક્શન શેર કરે છે, ઈથરનેટ CSMA/CD ના પ્રિન્સિપલની આસપાસ બનાવવામાં આવ્યું હતું, અથવા અથડામણ શોધ સાથે વાહક-સેન્સ મલ્ટિપલ એક્સેસ. મૂળભૂત રીતે, પ્રોટોકોલ ખાતરી કરે છે કે કોઈપણ ફ્રેમ મોકલતા પહેલા લાઈન 129 ઉપયોગમાં નથી. આજે, તે નેટવર્કિંગના શરૂઆતના દિવસો કરતાં ઘણું ઓછું મહત્વનું છે કારણ કે ઉપકરણો સામાન્ય રીતે સ્વીચ અથવા નોડ દ્વારા નેટવર્ક સાથે તેમનું પોતાનું ખાનગી જોડાણ ધરાવે છે. અને કારણ કે ઈથરનેટ હવે સંપૂર્ણ ડુપ્લેક્સનો ઉપયોગ કરીને કાર્ય કરે છે, મોકલવા અને પ્રાપ્ત કરવાની ચેનલો પણ સંપૂર્ણપણે અલગ છે, તેથી તેમની મુસાફરીના તે પગ પર ખરેખર અથડામણ થઈ શકતી નથી.

જ્યારે અથડામણની પરિસ્થિતિનો સામનો કરવો પડે તે સિવાય, ઈથરનેટમાં કોઈ ભૂલ સુધારણા નથી, તેથી સંચારને એ સુનિશ્ચિત કરવા માટે અદ્યતન પ્રોટોકોલ પર આધાર રાખવો જરૂરી છે કે બધું સંપૂર્ણ રીતે પ્રસારિત થઈ રહ્યું છે. જો કે, ઈથરનેટ હજુ પણ મોટાભાગના ઈન્ટરનેટ અને ડિજિટલ કોમ્યુનિકેશન્સ માટે આધાર પૂરો પાડે છે, અને મોટાભાગના ઉચ્ચ-સ્તરના પ્રોટોકોલ્સ સાથે સરળતાથી સંકલિત પણ થાય છે, તેથી તે આ દિવસોમાં લગભગ ક્યારેય સમસ્યા નથી.

નેટવર્ક કેવી રીતે બનાવવામાં આવે છે?

નોડ્સ અને કનેક્શન્સ (લિંક્સ) નો ઉપયોગ કરીને કમ્પ્યુટર નેટવર્કની રચના કરવામાં આવે છે જે ટીમોને એકબીજા સાથે વાતચીત કરવામાં સક્ષમ કરે છે. દરેક નોડ નેટવર્ક ઈન્ટરફેસ કાર્ડનો ઉપયોગ કરે છે જે વ્યાખ્યાયિત કરે છે કે તે નેટવર્ક સાથે કેવી રીતે ક્રિયાપ્રતિક્રિયા કરે છે. દરેક નેટવર્ક કાર્ડનું પોતાનું સંખ્યાત્મક સરનામું હોય છે.

અંતિમ-વપરાશકર્તા ઉપકરણો અને સ્થાનિક નેટવર્ક (End-User Devices and local networks)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સમર્થ હશો

- કોમ્યુનિકેટ નેટવર્ક વ્યાખ્યાયિત કરો
- કમ્પ્યુટર નેટવર્કમાં નોડ ઉપકરણોના પ્રકાર
- અંતિમ ઉપકરણોના કાર્યો.

નેટવર્ક ઉપકરણો કે જેનાથી લોકો સૌથી વધુ પરિચિત છે તેને અંતિમ ઉપકરણો કહેવામાં આવે છે. નેટવર્ક સાથે જોડાયેલા તમામ કોમ્પ્યુટર કે જેઓ નેટવર્ક કોમ્યુનિકેશનમાં સીધા ભાગ લે છે તેને હોસ્ટ તરીકે વર્ગીકૃત કરવામાં આવે છે. આ ઉપકરણો વપરાશકર્તાઓ અને અંતર્ગત સંચાર નેટવર્ક વચ્ચે ઈન્ટરફેસ બનાવે છે.

એન્ડ યુઝર

નોડ ડેસ્કટોપ કોમ્પ્યુટર, લેપટોપ, પ્રિન્ટર, ટેબ્લેટ અને સર્વર સહિતના અંતિમ વપરાશકર્તા ઉપકરણોના ઘણા ઉદાહરણો છે. આ અંતિમ વપરાશકર્તા ઉપકરણો માટે સૌથી સામાન્ય શબ્દો છે હોસ્ટ, અને અંતિમ વપરાશકર્તા નોડ; અમે આ સમગ્ર પ્રકરણમાં અંતિમ વપરાશકર્તા નોડનો ઉપયોગ કરીશું.

કમ્પ્યુટર નેટવર્કમાં નોડ ઉપકરણોના પ્રકાર:

અંતિમ ઉપકરણો અને મધ્યસ્થી ઉપકરણો આપણે જાણીએ છીએ કે કોમ્પ્યુટર નેટવર્ક એ એક શબ્દ છે જેનો ઉપયોગ સામાન્ય રીતે ટ્રાન્સમિશન મીડિયા (અથવા ચેનલો) તરીકે ઓળખાતી કોમ્યુનિકેશન લિંક્સ દ્વારા કનેક્ટેડ એકબીજા સાથે જોડાયેલા નોડ્સ (કમ્પ્યુટર, પ્રિન્ટર્સ અથવા અન્ય કોઈપણ ઉપકરણો) ના કોઈપણ જૂથ (અથવા સિસ્ટમ) નો સંદર્ભ આપવા માટે થાય છે. માહિતીના વિનિમય અને સંસાધનોની વહેંચણી માટે. કમ્પ્યુટર નેટવર્કમાં, નોડ એ કોઈપણ ઉપકરણ છે જે ડેટા મોકલવા અથવા પ્રાપ્ત કરવામાં સક્ષમ છે, ચોક્કસ અને ઈચ્છિત પ્રવાહ દરે સુરક્ષિત અને વિશ્વસનીય રીતે અન્ય નોડ્સ પર અને ત્યાંથી.

કાર્યક્ષમતા અને વપરાશના આધારે, નોડ ઉપકરણોને વ્યાપક રીતે નીચેના પ્રકારોમાં વર્ગીકૃત કરી શકાય છે -

- 1 અંતિમ ઉપકરણો
- 2 મધ્યસ્થી ઉપકરણો

અંતિમ ઉપકરણો

એન્ડ ડીવાઈસ એ નોડ ડીવાઈસ છે જે કોમ્પ્યુટર નેટવર્ક પર થતા કોમ્યુનિકેશનમાં સોર્સ પોઈન્ટ અથવા ડેસ્ટિનેશન પોઈન્ટ તરીકે સેવા આપે છે. કોમ્પ્યુટર નેટવર્કિંગ સિસ્ટમમાં આવનારી પ્રગતિ સાથે, અમારી પાસે

નોડ્સ છે જે ક્લાયન્ટ, સર્વર અથવા બંને તરીકે કામ કરી શકે છે. બાકીનું નેટવર્ક આ અંતિમ ઉપકરણોની આસપાસ તેમની વચ્ચે સંચાર લિંક્સ સ્થાપિત કરવા માટે બનાવવામાં આવ્યું છે. નોડ ઉપકરણો પર ઈન્સ્ટોલ કરેલ સૉફ્ટવેર કમ્પ્યુટર નેટવર્કમાં તેઓ જે ભૂમિકા ભજવે છે તે નક્કી કરે છે. તેમના ઉપયોગ અનુસાર, અંતિમ ઉપકરણોને વ્યાપક રીતે નીચેના શ્રેણીઓમાં જૂથબદ્ધ કરી શકાય છે - (ફિગ 1)

Fig 1



- ક્લાઈન્ટ તરીકે અંતિમ ઉપકરણો - ડેટાની વિનંતી કરવા, પ્રાપ્ત ડેટા પ્રદર્શિત કરવા વગેરેના કાર્યો કરો જે સામાન્ય રીતે અંતિમ ગ્રાહકો દ્વારા ઉપયોગ માટે હોય છે.
- સર્વર તરીકે અંતિમ ઉપકરણો - એવા પ્રોગ્રામ્સથી સજ્જ છે જે નેટવર્ક પરના અન્ય નોડ્સ (અથવા હોસ્ટ) ને વેબપેજ અથવા ઈ-મેઈલ જેવી માહિતી અને સેવાઓ પ્રદાન કરે છે.

અંતિમ ઉપકરણોના કાર્યો

- 1 તેઓ નેટવર્ક દ્વારા વહેતા ડેટા અથવા માહિતીના પ્રવર્તક તરીકે સેવા આપે છે.
- 2 ઘણા નોડ ઉપકરણો ધરાવતા અંતિમ-વપરાશકર્તાઓ (માણસો) અને સંચાર નેટવર્ક વચ્ચેના ઈન્ટરફેસ તરીકે કાર્ય કરો.

અંતિમ ઉપકરણોના ઉદાહરણો

વર્ક સ્ટેશન, લેપટોપ, ડેસ્કટોપ કોમ્પ્યુટર, પ્રિન્ટર્સ અને સ્કેનર્સ, સર્વર્સ (ફાઇલ સર્વર્સ, વેબ સર્વર્સ), મોબાઇલ ફોન, ટેબ્લેટ્સ, સ્માર્ટ ફોન વગેરે

મધ્યસ્થી ઉપકરણો

મધ્યસ્થી ઉપકરણો એ નોડ ઉપકરણો છે જે કમ્પ્યુટર નેટવર્કમાં ડેટાને એક બાજુથી બીજી બાજુ ફોરવર્ડ કરવા માટે રચાયેલ છે. આ મધ્યસ્થી ઉપકરણો અન્ય નોડ્સ માટે કનેક્ટિંગ માધ્યમ તરીકે (અન્ય સેવાઓ ઓફર કરવામાં આવી રહી છે) તરીકે કામ કરે છે અને સમગ્ર કમ્પ્યુટર નેટવર્ક પર ઈચ્છિત પ્રવાહ દરે ડેટા અસરકારક રીતે વહે છે તેની ખાતરી કરીને પૃષ્ઠભૂમિમાં કાર્યોને હેન્ડલ કરે છે.

તેમના દ્વારા વહેતા ડેટાના સંચાલન માટે મધ્યસ્થી ઉપકરણો નેટવર્ક ઇન્ટરકનેક્શન્સ વિશેની માહિતી સાથે વિવિધ એડ્રેસિંગ સિસ્ટમ્સ જેમ કે IP એડ્રેસ, MAC એડ્રેસ અને પોર્ટ નંબર્સ (અથવા પોર્ટ એડ્રેસ) નો ઉપયોગ કરે છે. કમ્પ્યુટર નેટવર્ક્સમાં વધુ વિવિધ પ્રકારના સ્વિચિંગ સંદેશાવ્યવહાર દરમિયાન નેટવર્ક દ્વારા જે માર્ગ લે છે તે નક્કી કરે છે.

મધ્યસ્થી ઉપકરણોના કાર્યો

- સિગ્નલ ડેમ્પિંગ એ એક સામાન્ય ઘટના છે જે આ ઉપકરણો દ્વારા કરવામાં આવતા ડેટા સિગ્નલોના પુનર્જીવન અને પુનઃપ્રસારણ દ્વારા દૂર કરવામાં આવે છે.
- સફળ ટ્રાન્સફરની ખાતરી કરવા માટે સ્વિચિંગ ટેકનિકનો ઉપયોગ કરવામાં આવી રહી છે તેના આધારે ઉપકરણો નેટવર્ક દ્વારા અસ્તિત્વમાં રહેલા સ્રોત સરનામાં, ગંતવ્ય સરનામાં અથવા વિવિધ માર્ગોની માહિતીનો રેકૉર્ડ રાખે છે.
- તેઓ રિડન્ડન્સી બિટ્સ વગેરેનો ઉપયોગ કરીને અસરકારક રીતે ખામીઓ અને ભૂલો પણ શોધી કાઢે છે અને કમ્પ્યુટર નેટવર્કમાં ડેટા ટ્રાન્સફર કરતી વખતે સુધારાઓ કરીને ખામી સહિષ્ણુતાને વધુ સુનિશ્ચિત કરવા માટે ઉપકરણોને સૂચિત કરે છે.
- સંદેશાવ્યવહારના સમગ્ર અભ્યાસક્રમ દરમિયાન ચોક્કસ ઈચ્છિત પ્રવાહ નિયંત્રણ અને પ્રતિભાવ સમય સમાપ્તિની જાળવણી.
- સંદેશાઓ (અથવા પેકેટો)નું સેટઅપ, જૂથ અને દિશા તેમની પ્રાથમિકતાઓ અનુસાર આ ઉપકરણો દ્વારા કરવામાં આવે છે.
- સુરક્ષા સેટિંગ્સના આધારે ડેટાના પ્રવાહને પરવાનગી આપીને અથવા નકારીને નેટવર્ક સુરક્ષા પ્રદાન કરો.

મધ્યસ્થી ઉપકરણોના ઉદાહરણો

હબ, સ્વીચો, વાયરલેસ એક્સેસ પોઇન્ટ અને નેટવર્ક એક્સેસ કરવા માટે વપરાતા અન્ય ઉપકરણો, ફાઇલ સર્વર્સ, વેબ સર્વર્સ, પ્રિન્ટ સર્વર્સ, મોડેમ, ઇન્ટરનેટવર્કિંગ માટે ઉપયોગમાં લેવાતા ઉપકરણો જેમ કે રાઉટર્સ, બ્રિજ, રીપીટર અને સુરક્ષા ફાયરવોલ વગેરે..

સ્થાનિક નેટવર્ક

નેટવર્કને કદ, જટિલતા, સુરક્ષા સ્તર અથવા ભૌગોલિક શ્રેણીના આધારે વર્ગીકૃત કરી શકાય છે. અમે ભૌગોલિક ફેલાવાના આધારે કેટલીક સૌથી લોકપ્રિય ટોપોલોજીની ચર્ચા કરીશું.

PAN (ફિગ 2)

PAN એ પર્સનલ એરિયા નેટવર્કનું ટૂંકું નામ છે. PAN એ વ્યક્તિની ખાનગી જગ્યાની રેન્જમાં, સામાન્ય રીતે 10 મીટરની રેન્જમાંના ઉપકરણો વચ્ચેનું ઇન્ટરકનેક્શન છે. જો તમે બ્લૂટૂથનો ઉપયોગ કરીને તમારા લેપટોપથી

Fig 2



મોબાઇલમાં અથવા મોબાઇલથી તમારા મિત્રના મોબાઇલમાં ઈમેલો કે ગીતો ટ્રાન્સફર કર્યા હોય, તો તમે પર્સનલ એરિયા નેટવર્ક સેટ કર્યું છે અને તેનો ઉપયોગ કર્યો છે.

વ્યક્તિ તેના લેપટોપ, સ્માર્ટ ફોન, પર્સનલ ડિજિટલ આસિસ્ટન્ટ અને પોર્ટેબલ પ્રિન્ટરને ઘરે બેઠા નેટવર્કમાં કનેક્ટ કરી શકે છે. આ નેટવર્ક સંપૂર્ણપણે Wi-Fi અથવા વાયર્ડ અને વાયરલેસનું સંયોજન હોઈ શકે છે.

LAN (ફિગ 3)

LAN અથવા લોકલ એરિયા નેટવર્ક એ ઓફિસ, બિલ્ડિંગ અથવા મેન્યુફેક્ચરિંગ યુનિટ જેવી એક સાઈટ પર ફેલાયેલું વાયર્ડ નેટવર્ક છે. જ્યારે ટીમના સભ્યોને સોફ્ટવેર અને હાર્ડવેર સંસાધનો એકબીજા સાથે શેર કરવાની જરૂર હોય ત્યારે LAN સેટ કરવામાં આવે છે પરંતુ બહારની દુનિયા સાથે નહીં. લાક્ષણિક સોફ્ટવેર સંસાધનોમાં સત્તાવાર દસ્તાવેજો, વપરાશકર્તા માર્ગદર્શિકા, કર્મચારીની હેન્ડબુક વગેરેનો સમાવેશ થાય છે. નેટવર્ક પર સરળતાથી શેર કરી શકાય તેવા હાર્ડવેર સંસાધનોમાં પ્રિન્ટર, ફેક્સ મશીન, મોડેમ, મેમરી સ્પેસ વગેરેનો સમાવેશ થાય છે. આ સંસ્થા માટે માળખાકીય ખર્ચમાં ભારે ઘટાડો કરે છે

Fig 3



વાયર્ડ અથવા વાયરલેસ કનેક્શનનો ઉપયોગ કરીને LAN સેટ કરી શકાય છે. LAN જે સંપૂર્ણપણે વાયરલેસ હોય તેને વાયરલેસ LAN અથવા WLAN કહેવાય છે.

માણસ (ફિગ 4)

MAN એ મેટ્રોપોલિટન એરિયા નેટવર્કનું ટૂંકું નામ છે. તે એક શહેર, કોલેજ કેમ્પસ અથવા નાના પ્રદેશમાં ફેલાયેલું નેટવર્ક છે. MAN એ LAN કરતા મોટો છે અને સામાન્ય રીતે કેટલાક કિલોમીટરમાં ફેલાયેલો છે. MAN નો ઉદ્દેશ્ય હાર્ડવેર અને સોફ્ટવેર સંસાધનોને વહેંચવાનો છે, જેનાથી ઇન્ફ્રાસ્ટ્રક્ચર ખર્ચમાં ઘટાડો થાય છે. MAN અનેક LAN ને જોડીને બનાવી શકાય છે.

MAN નું સૌથી સામાન્ય ઉદાહરણ કેબલ ટીવી નેટવર્ક છે.

Fig 4



WAN (ફિગ 5)

WAN અથવા વાઈડ એરિયા નેટવર્ક એક દેશ અથવા ઘણા દેશોમાં ફેલાયેલું છે. WAN એ સામાન્ય રીતે ઘણા LAN, MANs અને WAN નું નેટવર્ક છે. ઉપલબ્ધતા અને વિશ્વસનીયતા પર આધાર રાખીને, વાયર્ડ અથવા વાયરલેસ કનેક્શનનો ઉપયોગ કરીને નેટવર્ક સેટ કરવામાં આવે છે.

WAN નું સૌથી સામાન્ય ઉદાહરણ ઇન્ટરનેટ છે

Fig 5



નેટવર્ક સંચાર

- નેટવર્ક સંચારની મૂળભૂત બાબતો
- હોમ નેટવર્ક વ્યાખ્યાયિત કરો
- Wi-Fi વ્યાખ્યાયિત કરો
- LAN નો પરિચય

નેટવર્ક કમ્યુનિકેશનની

મૂળભૂત બાબતો આ વિભાગ નેટવર્ક સંચારને ખૂબ જ ઉચ્ચ સ્તરેથી જુએ છે અને સમગ્ર માહિતીમાં ઉપયોગમાં લેવાતા કેટલાક શબ્દોને વ્યાખ્યાયિત કરે છે. z/OS® નેટવર્ક કમ્યુનિકેશન અને TCP/IP સોકેટ્સ પર વધુ વિગતવાર માહિતી માટે, z/OS કોમ્યુનિકેશન્સ સર્વર: IP રૂપરેખાંકન માર્ગદર્શિકા અને z/OS કોમ્યુનિકેશન્સ સર્વર: IP પ્રોગ્રામરની માર્ગદર્શિકા અને સંદર્ભ જુઓ. IPv6 નેટવર્ક કોમ્યુનિકેશન અને AF_INET6 સોકેટ્સ પર વધુ વિગતવાર માહિતી માટે, જુઓ z/OS કોમ્યુનિકેશન સર્વર: IPv6 નેટવર્ક અને એપલ ડિઝાઇન માર્ગદર્શિકા.

નેટવર્ક કમ્યુનિકેશન, અથવા ઇન્ટરનેટવર્કિંગ, પ્રોટોકોલનો સમૂહ (એટલે કે નિયમો અને ધોરણો) વ્યાખ્યાયિત કરે છે જે એપ્લિકેશન પ્રોગ્રામ્સને જ્યાં તેઓ ચલાવવામાં આવે છે તે હાર્ડવેર અને ઓપરેટિંગ સિસ્ટમને ધ્યાનમાં લીધા વિના એકબીજા સાથે વાત કરવાની મંજૂરી આપે છે. ઇન્ટરનેટવર્કિંગ એપ્લિકેશન પ્રોગ્રામ્સને તેમના ભૌતિક નેટવર્ક કનેક્શનથી સ્વતંત્ર રીતે વાતચીત કરવાની મંજૂરી આપે છે.

TCP/IP નામની ઇન્ટરનેટવર્કિંગ ટેકનોલોજીનું નામ તેના બે મુખ્ય પ્રોટોકોલ પર રાખવામાં આવ્યું છે: ટ્રાન્સમિશન કંટ્રોલ પ્રોટોકોલ (TCP) અને ઇન્ટરનેટ પ્રોટોકોલ (IP). TCP/IP ને સમજવા માટે, તમારે નીચેની શરતોથી પરિચિત હોવા જોઈએ:

ક્લાયન્ટ

એ પ્રક્રિયા કે જે નેટવર્ક પર સેવાઓની વિનંતી કરે છે.

સર્વર એ પ્ર

ક્રિયા કે જે ક્લાયન્ટની સેવા માટેની વિનંતીનો જવાબ આપે છે.

ડેટાગ્રામ માહિતીનું

મૂળભૂત એકમ, જેમાં એક અથવા વધુ ડેટા પેકેટનો સમાવેશ થાય છે, જે પરિવહન સ્તરે ઇન્ટરનેટ પર પસાર થાય છે.

પેકેટ

કમ્પ્યુટર અને તેના નેટવર્ક વચ્ચેના ડેટા ટ્રાન્ઝેક્શનનું એકમ અથવા બ્લોક. પેકેટમાં સામાન્ય રીતે નેટવર્ક હેડર, ઓછામાં ઓછું એક ઉચ્ચ સ્તરીય પ્રોટોકોલ હેડર અને ડેટા બ્લોક્સ હોય છે. સામાન્ય રીતે, ડેટા બ્લોક્સનું ફોર્મેટ પેકેટોને કેવી રીતે હેન્ડલ કરવામાં આવે છે તેની અસર કરતું નથી. પેકેટ્સ એ નેટવર્ક દ્વારા ડેટા મોકલવા માટે ઇન્ટરનેટવર્ક લેયર પર વપરાતું વિનિમય માધ્યમ છે.

ઇન્ટરનેટ પરના ઉપકરણો એકબીજા સાથે કેવી રીતે વાતચીત કરે છે?

ઇથરનેટ એ એક પ્રમાણભૂત પ્રોટોકોલ છે જે કોઈપણ કમ્પ્યુટરને એક સાથે વાતચીત કરવાની મંજૂરી આપે છે અન્ય આ મશીનોને ઇથરનેટ કેબલ અથવા “કેટેગરી 5” વાયરિંગ દ્વારા એકબીજા સાથે કનેક્ટ કરવાની જરૂર છે.

હોમ નેટવર્કિંગ શું છે?

હોમ નેટવર્ક એ ઉપકરણોનું જૂથ છે - જેમ કે કમ્પ્યુટર્સ, ગેમ સિસ્ટમ્સ, પ્રિન્ટર્સ અને મોબાઇલ ઉપકરણો - જે ઇન્ટરનેટ અને એકબીજા સાથે જોડાય છે. હોમ નેટવર્ક્સ બે રીતે કનેક્ટ થાય છે:

- વાયર્ડ નેટવર્ક, જે પ્રિન્ટર અને સ્કેનર જેવા ઉપકરણોને કેબલ વડે જોડે છે
- વાયરલેસ નેટવર્ક, જે કેબલ વગર ટેબ્લેટ અને ઈ રીડર જેવા ઉપકરણોને જોડે છે

હોમ નેટવર્ક કેમ સેટ કરવું?

હોમ નેટવર્ક સ્થાપિત કરવાના ઘણા કારણો છે. અહીં કેટલીક વસ્તુઓ છે જે હોમ નેટવર્કિંગ તમને કરવાની મંજૂરી આપે છે:

- બહુવિધ કમ્પ્યુટર્સ, ગેમ સિસ્ટમ્સ, મોબાઇલ ઉપકરણો અને વધુથી ઇન્ટરનેટથી કનેક્ટ કરો.
- નેટવર્ક સાથે જોડાયેલા તમામ ઉપકરણો પર ફાઇલો અને ફોલ્ડર્સને એક્સેસ કરો.
- એક જ પ્રિન્ટર પર બહુવિધ કમ્પ્યુટર્સથી પ્રિન્ટ કરો.
- એક જ જગ્યાએ બધા નેટવર્ક ઉપકરણો માટે સુરક્ષા સેટિંગ્સ મેનેજ કરો.

જો તમે હોમ નેટવર્કિંગ અજમાવવા માટે તૈયાર છો, તો તમારે શું મેળવવાની જરૂર પડશે તે જાણવા માટે નીચે વાંચો શરૂ કર્યું અને તમારા નેટવર્ક માટે ઉપલબ્ધ વિકલ્પો જોવા માટે

હોમ નેટવર્ક સેટ કરવા માટે તમારે શું કરવાની જરૂર છે હોમ નેટવર્કિંગ

સેટ કરવા માટે, તમારે નીચેનાની જરૂર પડશે:

- એક્સફિનિટી ઇન્ટરનેટ સેવાનું સબસ્ક્રિપ્શન (અથવા અન્ય ઇન્ટરનેટ પ્રદાતાનું સબસ્ક્રિપ્શન)

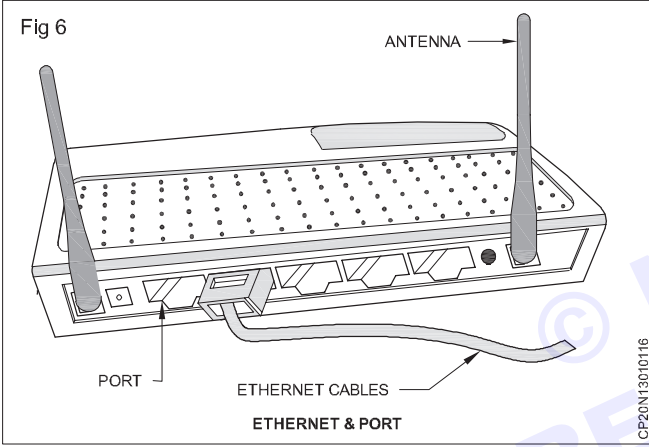
- એક મોડેમ, જે ઈન્ટરનેટ સાથે જોડાય છે, અને રાઉટર, જે તમારા મોડેમ દ્વારા તમારા ઉપકરણોને એકબીજા સાથે અને ઈન્ટરનેટ સાથે જોડે છે (અથવા ગેટવે, જે મોડેમ અને રાઉટર બંને તરીકે કાર્ય કરે છે)
- નેટવર્ક સાથે જોડાવા માટે કમ્પ્યુટર અથવા અન્ય ઉપકરણ

વધુ માહિતી માટે વાયરલેસ ગેટવે સક્રિય કરો જુઓ.

વાયરલેસ ગેટવે 1 (મોડલ નંબર TG852G, TG862G, SMCD3GNV, TC8305C) અને વાયરલેસ ગેટવે 2 (મોડલ નંબર DPC3939) ઓલ-ઇન-વન તરીકે કાર્ય કરે છે મોડેમ, રાઉટર અને ફોન ઉપકરણ. તેઓ આપોઆપ વપરાશકર્તાઓને હોમ નેટવર્ક માટે ઉપલબ્ધ શ્રેષ્ઠ સુરક્ષા સેટિંગ્સ પ્રદાન કરો. જુઓ વાયરલેસ ગેટવે શું છે? વધારે માહિતી માટે.

વાયરલેસ હોમ નેટવર્ક : વાયરલેસ નેટવર્ક, જેને ઘણીવાર Wi-Fi કહેવામાં આવે છે, કેબલનો ઉપયોગ કર્યા વિના ઉપકરણોને એકબીજા સાથે અને ઈન્ટરનેટ સાથે જોડે છે. જુઓ Wi-Fi શું છે અને હું તેને કેવી રીતે મેળવી શકું? વધુ માટે માહિતી

વાયર્ડ હોમ નેટવર્ક (ફિગ 6) : વાયર્ડ હોમ નેટવર્ક ઈથરનેટ કેબલનો ઉપયોગ કરીને ઉપકરણોને એકબીજા સાથે અને ઈન્ટરનેટ સાથે જોડે છે.



વાયર્ડ હોમ નેટવર્ક હોવાના ઘણા ફાયદા છે:

- ઈન્ટરનેટ સાથે ઝડપી અને વધુ વિશ્વસનીય કનેક્શન
- સુરક્ષામાં વધારો, કારણ કે કોઈ બહારના વપરાશકર્તાઓ તમારા નેટવર્કને એક્સેસ કરી શકતા નથી
- વાયરલેસ કનેક્શન મિક્સ્ડ હોમ નેટવર્ક કરતાં વધુ સરળ સેટ-અપ અને

મુશ્કેલીનિવારણ : ઘણા લોકોને લાગે છે કે વાયરલેસ અને વાયર્ડ નેટવર્કિંગનું મિશ્રણ તેમની જરૂરિયાતોને શ્રેષ્ઠ રીતે પૂર્ણ કરે છે. દાખલા તરીકે, ઉપકરણો કે જે મૂવીઝ સ્ટ્રીમ કરે છે તે ઝડપી અને વધુ સ્થિર વાયર્ડ કનેક્શનથી લાભ મેળવે છે. લેપટોપ અથવા ટેબ્લેટ જેવા ઉપકરણો, જોકે, વાયરલેસ કનેક્શન સાથે ઉપલબ્ધ ગતિશીલતાથી લાભ મેળવે છે. વાયરલેસ ગેટવે 1 અને વાયરલેસ ગેટવે 2 બંને વાયરલેસ ક્ષમતા અને ચાર ઈથરનેટ પોર્ટ સાથે આવે છે, જે તમને એક જ સમયે કેબલ સાથે અને વગર ઉપકરણોને કનેક્ટ કરવાની મંજૂરી આપે છે. હોમ નેટવર્કિંગ ગ્લોસરી નીચે સામાન્ય હોમ નેટવર્કિંગ શબ્દોની સૂચિ છે.

આ વ્યાખ્યાઓથી પોતાને પરિચિત કરવા માટે થોડો સમય ફાળવો, કારણ કે તેનો ઉલ્લેખ અન્ય સહાય લેખોમાં કરવામાં આવશે. વાયરલેસ નેટવર્ક - ઉપકરણોનું જૂથ (કમ્પ્યુટર, ગેમ સિસ્ટમ્સ, ઈ-રીડર્સ, વગેરે) એકબીજા સાથે અને કેબલ વિના ઈન્ટરનેટ સાથે જોડાયેલ છે, જેને Wi-Fi પણ કહેવાય છે.

હોમ નેટવર્કિંગ ગ્લોસરી

નીચે સામાન્ય હોમ નેટવર્કિંગ શરતોની સૂચિ છે. આ વ્યાખ્યાઓથી પોતાને પરિચિત કરવા માટે થોડો સમય ફાળવો, કારણ કે તેનો ઉલ્લેખ અન્ય સહાય લેખોમાં કરવામાં આવશે.

મોડેમ - એક ઉપકરણ કે જે રાઉટર અથવા કમ્પ્યુટરને કેબલ દ્વારા ઈન્ટરનેટ સાથે જોડે છે.

રાઉટર - એક ઉપકરણ જે વાયરલેસ સિગ્નલ પ્રસારિત કરે છે અને મોડેમ દ્વારા ઉપકરણોને એકબીજા અને ઈન્ટરનેટ સાથે જોડે છે.

ગેટવે - એક ઉપકરણ જે મોડેમ અને રાઉટર બંને તરીકે કાર્ય કરે છે, જેમ કે વાયરલેસ ગેટવે 1 અથવા વાયરલેસ ગેટવે 2.

સિગ્નલ રેન્જ - તે વિસ્તાર કે જેમાં વાયરલેસ સક્ષમ ઉપકરણ વાયરલેસ સિગ્નલ લઈ શકે છે અને વાયરલેસ નેટવર્ક સાથે કનેક્ટ થઈ શકે છે.

એડમિનિસ્ટ્રેશન સાઈટ - રાઉટર અને વાયરલેસ નેટવર્કની સેટિંગ્સ જોવા અને બદલવા માટે વપરાતી ઓનલાઈન સાઈટ, જેને વાયરલેસ ગેટવે 1 અને વાયરલેસ માટે એડમિન ટૂલ કહેવાય છે. ગેટવે 2.

રાઉટર વપરાશકર્તા નામ અને પાસવર્ડ - એડમિનિસ્ટ્રેશન સાઈટને એક્સેસ કરવા માટે વપરાતી લોગિન માહિતી.

નેટવર્ક નામ (SSID) - વાયરલેસ નેટવર્કનું નામ.

નેટવર્ક પાસવર્ડ (કી) - વાયરલેસ નેટવર્ક સાથે જોડાવા માટે વપરાતો પાસવર્ડ.

Wi-Fi વ્યાખ્યા

Wi-Fi એ એક વાયરલેસ તકનીક છે જેનો ઉપયોગ કમ્પ્યુટર, ટેબ્લેટ, સ્માર્ટફોન અને અન્ય ઉપકરણોને ઈન્ટરનેટથી કનેક્ટ કરવા માટે થાય છે.

Wi-Fi એ વાયરલેસ રાઉટરથી નજીકના ઉપકરણ પર મોકલવામાં આવેલ રેડિયો સિગ્નલ છે, જે સિગ્નલને તમે જોઈ અને ઉપયોગમાં લઈ શકો છો તે ડેટામાં અનુવાદિત કરે છે. ઉપકરણ રાઉટર પર પાછા રેડિયો સિગ્નલ પ્રસારિત કરે છે, જે વાયર અથવા કેબલ દ્વારા ઈન્ટરનેટ સાથે જોડાય છે.

Wi-Fi નેટવર્ક શું છે?

Wi-Fi નેટવર્ક એ ફક્ત એક ઈન્ટરનેટ કનેક્શન છે જે વાયરલેસ રાઉટર દ્વારા ઘર અથવા વ્યવસાયમાં બહુવિધ ઉપકરણો સાથે શેર કરવામાં આવે છે. રાઉટર તમારા ઈન્ટરનેટ મોડેમ સાથે સીધું જ જોડાયેલ છે અને તમારા બધા Wi-Fi સક્ષમ ઉપકરણો પર ઈન્ટરનેટ સિગ્નલને બ્રોડકાસ્ટ કરવા માટે હબ તરીકે કાર્ય કરે છે. જ્યાં સુધી તમે તમારા નેટવર્ક કવરેજ ક્ષેત્રમાં હોવ ત્યાં સુધી આ તમને ઈન્ટરનેટથી કનેક્ટ રહેવા માટે સુગમતા આપે છે.

શું કરે Wi-Fi સ્ટેન્ડ માટે?

આ શબ્દ માર્કેટિંગ ફર્મ દ્વારા બનાવવામાં આવ્યો હતો કારણ કે વાયરલેસ ઉદ્યોગ IEEE 802.11 તરીકે ઓળખાતી કેટલીક વપરાશકર્તા મૈત્રીપૂર્ણ તકનીકનો સંદર્ભ આપવા માટે વપરાશકર્તા-મૈત્રીપૂર્ણ નામ શોધી રહ્યો હતો અને નામ અટકી ગયું હતું. Wi-Fi, જેને ઘણીવાર WiFi, wifi, wi-fi અથવા wiifi તરીકે ઓળખવામાં આવે છે, તે વાયરલેસ ફિડેલિટી માટે ટૂંકું હોવાનું માનવામાં આવે છે અને માર્કેટિંગ પેઢી માટે ચૂકવણી કરતી સંસ્થાને ક્યારેક વાયરલેસ ફિડેલિટી એલાયન્સ ઈન્ક તરીકે ઓળખવામાં આવે છે.

Wi-Fi કેવી રીતે કામ કરે છે?

Wi-Fi રેડિયોનો ઉપયોગ કરે છે તમારા વાયરલેસ રાઉટરથી તમારા ટીવી, સ્માર્ટફોન, ટેબ્લેટ અને કમ્પ્યુટર જેવા તમારા Wi-Fi સક્ષમ ઉપકરણો પર

ડેટા ટ્રાન્સમિટ કરવા માટે તરંગો. કારણ કે તેઓ એરવેવ્સ પર એકબીજા સાથે વાતચીત કરે છે, તમારા ઉપકરણો અને વ્યક્તિગત માહિતી હેક્સ, સાયબર હુમલા અને અન્ય જોખમો માટે સંવેદનશીલ બની શકે છે. જ્યારે તમે કોફી શોપ અથવા એરપોર્ટ જેવા સ્થળોએ સાર્વજનિક Wi-Fi નેટવર્કથી કનેક્ટ કરો છો ત્યારે આ ખાસ કરીને સાચું છે. જ્યારે શક્ય હોય, ત્યારે પાસવર્ડ સુરક્ષિત હોય અથવા વ્યક્તિગત હોટસ્પોટ હોય તેવા વાયરલેસ નેટવર્કથી કનેક્ટ થવું શ્રેષ્ઠ છે.

Wi-Fi કનેક્શનના પ્રકારો

મોબાઇલ નેટવર્ક્સ હોમ ઈન્ટરનેટ ક્ષેત્રમાં વિસ્તરે છે તેમ ઘરે વાયરલેસ રીતે કનેક્ટ થવા માટેના તમારા વિકલ્પો વધી રહ્યા છે. ઈન્ટરનેટ સેવાની જેમ, દરેક પ્રકારના વાયરલેસ કનેક્શનના ફાયદા અને ગેરફાયદા છે, જેમ કે ઝડપ અને સિગ્નલની મજબૂતાઈ. અમે અહીં તેમાંથી કેટલાકની રૂપરેખા આપી છે.

વાયરલાઈન/રાઉટર

આજકાલ મોટાભાગના ઘરો ઈન્ટરનેટ એક્સેસ કરવા માટે વાયરલેસ રાઉટરનો ઉપયોગ કરે છે. ગુણોમાં સેટઅપની સગવડ, અંદરની ગતિશીલતાનો સમાવેશ થાય છે Wi-Fi એક્સેસ પોઈન્ટ (રાઉટર) ની શ્રેણી અને બહુવિધ ઉપકરણોને કનેક્ટ કરવાની ક્ષમતા. ગેરફાયદા: મર્યાદિત બેન્ડવિડ્થ અને ઓછી ઝડપ કારણ કે વધુ ઉપકરણો સમાન Wi-Fi નેટવર્ક સાથે જોડાયેલા છે, તેમજ ઘરના અન્ય ઇલેક્ટ્રોમેગ્નેટિક ઉપકરણો દ્વારા સંભવિત હસ્તક્ષેપ.

મોબાઇલ હોટસ્પોટ અથવા

જેટપેક મોબાઇલ અને સમર્પિત હોટસ્પોટ સફરમાં સલામત રીતે કનેક્ટ થવાની વધુ લોકપ્રિય રીત બની રહ્યા છે. બે સામાન્ય હોટસ્પોટ ઉપકરણો છે તમારો સ્માર્ટફોન અને એક જેટપેક. આજે મોટાભાગના કોઈપણ સ્માર્ટફોન અથવા ટેબ્લેટનો ઉપયોગ અસ્થાયી હોટસ્પોટ તરીકે થઈ શકે છે અને જો તમને પ્રસંગોપાત તેની જરૂર હોય તો તે એક શ્રેષ્ઠ વિકલ્પ છે. તે વાપરવા માટે સરળ છે અને વધારાના ઉપકરણો ખરીદવાની જરૂર નથી, પરંતુ તે તમારી બેટરી જીવન અને ડેટાને ખૂબ જ ઝડપી બનાવી શકે છે. બીજી તરફ, જેટપેક એક સમર્પિત મોબાઇલ હોટસ્પોટ તરીકે કામ કરે છે જે તમારા સ્માર્ટફોનની જેમ જ તમારા વિસ્તારમાં સેલ ટાવરમાંથી સિગ્નલ મેળવે છે. વધુ ઉપકરણો તેની સાથે કનેક્ટ થઈ શકે છે અને તે Wi-Fi ની મોટી શ્રેણી ઓફર કરે છે. અને કારણ કે તે એક અલગ ઉપકરણ છે, તમારા સ્માર્ટફોનની બેટરી પાવરને સ્પર્શતી નથી. આ ડાઉનફોલ માટે જેટપેક અને અલગ પ્લાન ખરીદવો પડે છે.

LTE હોમ ઈન્ટરનેટ

જો તમે એવા ગ્રામીણ વિસ્તારમાં રહો છો જ્યાં તમારા ઈન્ટરનેટ વિકલ્પો મર્યાદિત છે, તો 4G LTE હોમ ઈન્ટરનેટ ધ્યાનમાં લેવા યોગ્ય છે. તે લગભગ 25 Mbpsની સરેરાશ ડાઉનલોડ સ્પીડ સાથે સેલ ફોન ટાવર્સ અને મોબાઇલ નેટવર્ક્સ પર પહોંચાડવામાં આવતી હાઈ સ્પીડ ઈન્ટરનેટ સેવા પ્રદાન કરે છે. સેટેલાઈટ પર LTE ના ફાયદા તમારા કેરિયરના આધારે વધુ સારી ઝડપ અને વિશ્વસનીયતા છે.

5G હોમ ઈન્ટરનેટ : 5G હોમ ઈન્ટરનેટ (ફિક્સ્ડ વાયરલેસ એક્સેસ) હવે દેશભરમાં વધુ સ્થળોએ છે. ફિક્સ્ડ વાયરલેસ એક્સેસમાં એક સરળ પ્લગ એન્ડ પ્લે સેટઅપ છે, જેનો અર્થ છે કે ત્યાં કોઈ અવ્યવસ્થિત વાયર નથી અને સેટઅપ માટે તમારા ઘરે ટેકનિશિયન આવે તેની રાહ જોવાની જરૂર નથી.

ઘરે Wi-Fi મેળવવા

માટે ઉલ્લેખ કર્યા મુજબ, તમારા ઘરે વાયરલેસ સેવા મેળવવાની ઘણી રીતો છે અને મોટાભાગે ભૌગોલિક સ્થાન અને ઉપલબ્ધતા પર આધાર

રાખે છે. (જુઓ કે તમારા વિસ્તારમાં કયું Wi-Fi હોમ ઈન્ટરનેટ ઉપલબ્ધ છે.) મોટાભાગના શહેરી અને ઉપનગરીય વિસ્તારો આમાંની મોટાભાગની સેવાઓ પ્રદાન કરે છે, જેમાં 5G હોમ ઈન્ટરનેટ બરાબર ખૂણે છે. ગ્રામીણ વિસ્તારો મોટે ભાગે સેટેલાઈટ અને 4G LTE હોમ ઈન્ટરનેટ ઓફર કરશે. જો તમારી પાસે વાયર્ડ ઈન્ટરનેટ સેવા છે, તો તમે ઘરે તમારું પોતાનું Wi-Fi નેટવર્ક સેટ કરી શકશો. તમારા મોડેમ સાથે રાઉટરને કનેક્ટ કરીને, તમે રેન્જમાં તમારા બધા Wi-Fi સક્ષમ ઉપકરણો સાથે તમારું ઈન્ટરનેટ કનેક્શન શેર કરી શકો છો. જો તમારા ઘરમાં બે સ્તરો, કોંક્રિટની દિવાલો અથવા રેન્ડમ ડેડ ઝોન હોય, તો આ વિસ્તારોમાં વાયરલેસ સિગ્નલ રિલે કરતા Wi-Fi એક્સ્ટેન્ડર ઉમેરવાથી મોટો ફરક પડી શકે છે.

ધ્યાનમાં રાખો કે જેમ જેમ તમારા મોબાઇલ ઉપકરણોની સંખ્યા વધે છે તેમ તેમ બેન્ડવિડ્થની માંગ પણ વધે છે. તમારા ઉપકરણોને ટોચની ઝડપે ચાલુ રાખવા માટે, તમારે તમારા ઈન્ટરનેટ સ્પીડ પ્લાનને અપગ્રેડ કરવાની જરૂર પડી શકે છે. વેરાઈઝન અનેક ઓફર કરે છે તમારા સ્થાનના આધારે, DSL અને ફાઈબરથી લઈને 5G હોમ ઈન્ટરનેટ સુધીની ઈન્ટરનેટ સેવાઓ. વધુ સારી વિશ્વસનીયતા અને વેરિઝોનની સૌથી ઝડપી ડાઉનલોડ ઝડપ માટે, તમારા સરનામાં પર Fios ઈન્ટરનેટ અથવા 5G હોમ ઈન્ટરનેટ ઉપલબ્ધ છે કે કેમ તે તપાસો.

તમારા વિસ્તારમાં શું Wi-Fi ઈન્ટરનેટ ઉપલબ્ધ છે તે જુઓ.

LAN નો પરિચય

લોકલ એરિયા નેટવર્ક (LAN) માં એક સાથે જોડાયેલા કોમ્પ્યુટરોની શ્રેણીનો સમાવેશ થાય છે જે એક સંકુચિત સ્થાન પર નેટવર્ક બનાવે છે. LAN માંના કમ્પ્યુટર્સ TCP/IP ઈથરનેટ અથવા Wi-Fi દ્વારા એકબીજા સાથે જોડાય છે. LAN સામાન્ય રીતે સંસ્થા માટે વિશિષ્ટ હોય છે, જેમ કે શાળા, કાર્યાલય, સંગઠન અથવા ચર્ચ.

LAN ઉપકરણ શું છે?

લોકલ એરિયા નેટવર્ક (LAN) એ કમ્પ્યુટર્સ અને પેરિફેરલ ઉપકરણોનું જૂથ છે જે એક અલગ ભૌગોલિક વિસ્તારની અંદર સર્વર સાથે સામાન્ય સંચાર લાઈન અથવા વાયરલેસ લિંક શેર કરે છે. લોકલ એરિયા નેટવર્ક હોમ ઓફિસમાં બે કે ત્રણ જેટલા વપરાશકર્તાઓ અથવા કોર્પોરેશનની સેન્ટ્રલ ઓફિસમાં હજારો વપરાશકર્તાઓને સેવા આપી શકે છે.

નેટવર્ક ઉપકરણોના પ્રકાર

- હબ.
- સ્વિચ કરો.
- રાઉટર.
- પુલ.
- ગેટવે.
- મોડેમ.
- રીપીટર.
- એક્સેસ પોઈન્ટ.

લોકલ એરિયા નેટવર્ક (LAN) ના પ્રકાર

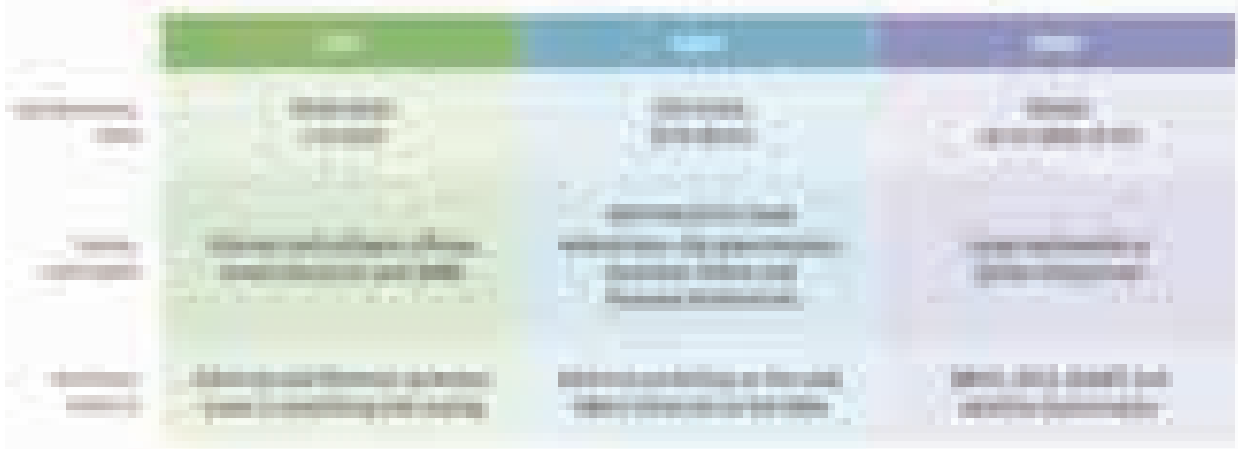
- પીઅર ટુ પીઅર (P2P) LAN....
- ટોકન રિંગ LAN....
- ટોકન બસ LAN....

- વાયર્ડ LAN....
- ક્લાઉડ-મેનેજ્ડ LAN....
- સાર્વજનિક ઇન્ટરનેટ....

- વાયર્ડ એન્ડ-યુઝર ઉપકરણો....
- મોબાઇલ અંતિમ-વપરાશકર્તા ઉપકરણો.

LAN વિ. MAN વિ. WAN (ફિગ 7)

Fig 7



ઇન્ટરનેટ ખ્યાલો

- www વ્યાખ્યાયિત કરો
- WWW આર્કિટેક્ચર
- વપરાશકર્તા ઇન્ટરફેસ અને એપ્લિકેશન્સ

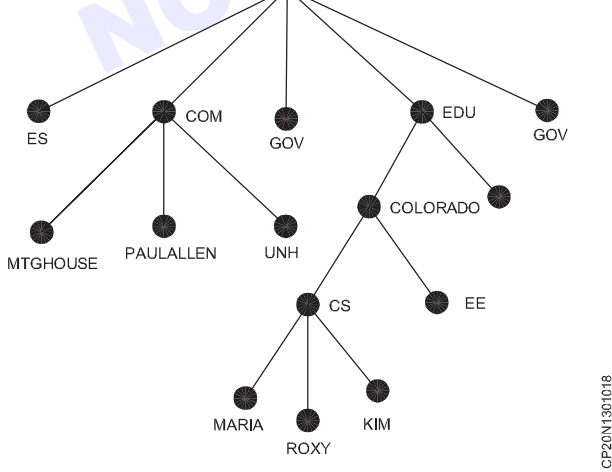
WWW એટલે વર્લ્ડ વાઇડ વેબ. વર્લ્ડ વાઇડ વેબની તકનીકી વ્યાખ્યા છે: ઇન્ટરનેટ પરના તમામ સંસાધનો અને વપરાશકર્તાઓ કે જે હાયપરટેક્સ્ટ ટ્રાન્સફર પ્રોટોકોલ (HTTP) નો ઉપયોગ કરી રહ્યાં છે.

વર્લ્ડ વાઇડ વેબ એ નેટવર્ક-સુલભ માહિતીનું બ્રહ્માંડ છે, જે માનવ જ્ઞાનનું મૂર્ત સ્વરૂપ છે.

સરળ શબ્દોમાં કહીએ તો, વર્લ્ડ વાઇડ વેબ એ ઇન્ટરનેટ પર કમ્પ્યુટર્સ વચ્ચે માહિતીની આપ-લે કરવાનો એક માર્ગ છે, તેને એકસાથે જોડીને ઇન્ટરેક્ટિવ મલ્ટીમીડિયા સંસાધનોના વિશાળ સંગ્રહમાં.

ઇન્ટરનેટ અને વેબ એક જ વસ્તુ નથી: વેબ માહિતીને પસાર કરવા માટે ઇન્ટરનેટનો ઉપયોગ કરે છે. (ફિગ 8)

Fig 8



ઉત્ક્રાંતિ: વર્લ્ડ વાઇડ વેબ 1989 માં જિનીવામાં CERN ખાતે ટીમોથી બર્નર્સ લી દ્વારા બનાવવામાં આવી હતી. વિશ્વવ્યાપીસંશોધકોને CERN ખાતે અસરકારક અને કાર્યક્ષમ રીતે સાથે મળીને કામ કરવાની મંજૂરી આપવા માટે તેમના દ્વારા પ્રસ્તાવ તરીકે વેબ અસ્તિત્વમાં આવ્યું.

આખરે તે વર્લ્ડ વાઇડ વેબ બની ગયું નીચેનો આકૃતિ વર્લ્ડ વાઇડ વેબના ઉત્ક્રાંતિને સંક્ષિપ્તમાં વ્યાખ્યાયિત કરે છે: (ફિગ 9)

Fig 9



WWW આર્કિટેક્ચર

WWW આર્કિટેક્ચર નીચેના રેખાકૃતિમાં બતાવ્યા પ્રમાણે ઘણા સ્તરોમાં વિભાજિત થયેલ છે:

ઓળખકર્તા અને અક્ષર

સમૂહ યુનિફોર્મ રિસોર્સ આઇડેન્ટિફાયર (URI) નો ઉપયોગ વેબ પર સંસાધનોને વિશિષ્ટ રીતે ઓળખવા માટે કરવામાં આવે છે અને UNIFORM-CODE તે વેબ પૃષ્ઠો બનાવવાનું શક્ય બનાવે છે જે માનવમાં વાંચી અને લખી શકાય છે.

ભાષાઓ

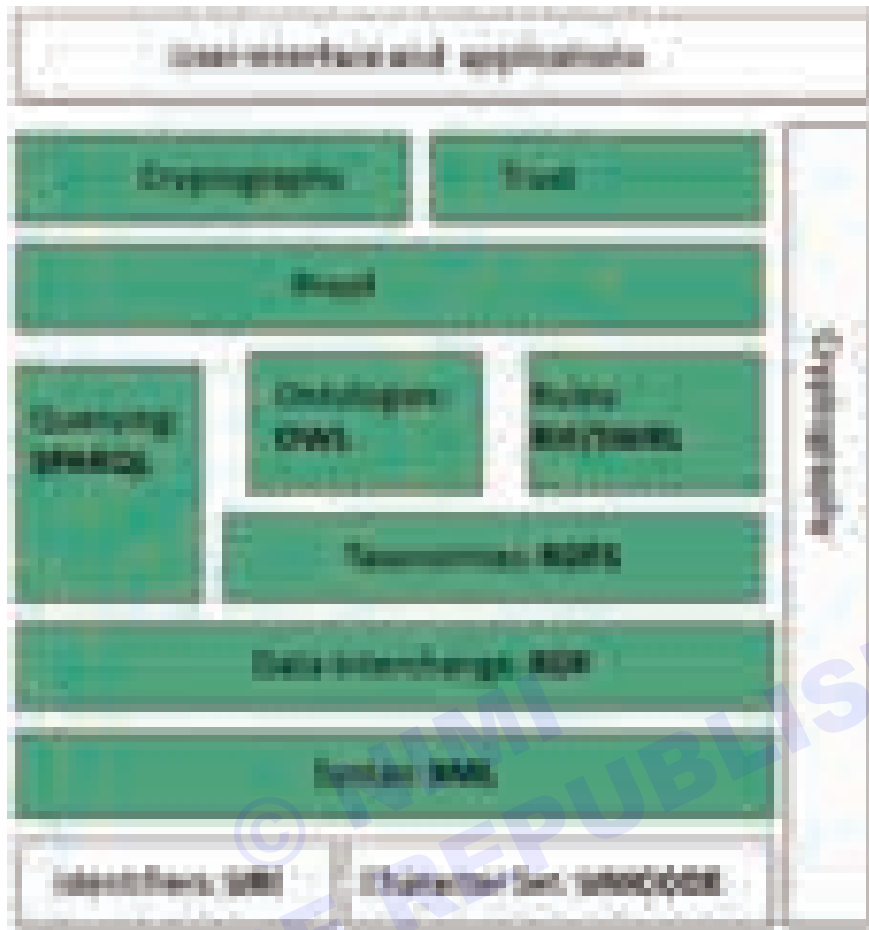
સિન્ટેક્સ XML (એક્સટેન્સિબલ માર્કઅપ લેંગ્વેજ) સિમેન્ટિક વેબમાં સામાન્ય સિન્ટેક્સને વ્યાખ્યાયિત કરવામાં મદદ કરે છે. ડેટા ઇન્ટરચેન્જ રિસોર્સ ડિસ્ક્રિપ્શન ફ્રેમવર્ક (RDF) ફ્રેમવર્ક વેબ માટે ડેટાના મુખ્ય પ્રતિનિધિત્વને વ્યાખ્યાયિત કરવામાં મદદ કરે છે. આરડીએફ ગ્રાફ સ્વરૂપમાં સંસાધન વિશેના ડેટાને રજૂ કરે છે.

આરડીએફ સ્કીમા (આરડીએફએસ) વર્ગીકરણ અને અન્ય ઓન્ટોલોજીકલ રચનાઓના વધુ પ્રમાણિત વર્ણનની મંજૂરી આપે છે.

ઓન્ટોલોજીસ (ફિગ 10)

વેબ ઓન્ટોલોજી લેંગ્વેજ (OWL) RDFS પર વધુ રચનાઓ ઓફર કરે છે. તે નીચેના ત્રણ સંસ્કરણોમાં આવે છે:

Fig 10



- વર્ગીકરણ અને સરળ અવરોધો માટે OWL લાઈટ.
- સંપૂર્ણ વર્ણન તર્ક આધાર માટે OWL DL.
- RDF નિયમોની વધુ વાક્યરચના

ત્રક

સ્વતંત્રતા માટે OWL RIF અને SWRL RDFs અને OWL માંથી ઉપલબ્ધ બાંધકામો ઉપરાંત નિયમો ઓફર કરે છે. સિમ્પલ પ્રોટોકોલ અને RDF ક્વેરી લેંગ્વેજ (SPARQL) એ એસક્ચ્યુઅલ જેવી ભાષા છે જેનો ઉપયોગ RDF ડેટા અને OWL ઑન્ટોલોજીસની ક્વેરી કરવા માટે થાય છે.

५३

તમામ સિમેન્ટીક અને નિયમો કે જે પ્રૂફની નીચેના સ્તરો પર એક્ઝિઝ્યુટ કરવામાં આવે છે અને તેમના પરિણામનો ઉપયોગ કપાત સાબિત કરવા માટે કરવામાં આવશે.

ક્રિપ્ટોગ્રાફી

ફિષ્ટોગ્રાફી એટલે કે સ્ત્રોતની ઉત્પત્તિની ચકાસણી માટે ડિજિટલ હસ્તાક્ષરનો ઉપયોગ થાય છે.

પ્રુઝર ઈન્ટરફેસ

અને એપ્લીકેશન્સ la er ની ટોચ પર ચુઝર ઈન્ટરફેસ અને એપ્લીકેશન લેયર ચુઝર ઈન્ટરફેસ માટે બનાવવામાં આવેલ છે.

WWW ઓપરેશન

WWW ક્લાયન્ટ-સર્વર અભિગમ પર કામ કરે છે. નીચેના પગલાં વેબ કેવી રીતે કાર્ય કરે છે તે સમજાવે છે:

- 1 વપરાશકર્તા સરનામામાં વેબ પૃષ્ઠનું URL (કહો, [http:// www.tutorialspoin .com](http://www.tutorialspoin.com)) દાખલ કરે છે વેબ બ્રાઉઝરનો બાર.
- 2 પછી બ્રાઉઝર [www.tutorialspoin .com](http://www.tutorialspoin.com) ને અનુરૂપ IP સરનામા માટે ડોમેન નેમ સર્વરને વિનંતી કરે છે.
- 3 IP સરનામું પ્રાપ્ત કર્યા પછી, બ્રાઉઝર HTTP પ્રોટોકોલનો ઉપયોગ કરીને વેબ સર્વરને વેબ પેજ માટેની વિનંતી મોકલે છે જે બ્રાઉઝર અને વેબ સર્વર કેવી રીતે વાતચીત કરે છે તેનો ઉલ્લેખ કરે છે.
- 4 પછી વેબ સર્વર HTTP પ્રોટોકોલનો ઉપયોગ કરીને વિનંતી મેળવે છે અને વિનંતી કરેલ વેબ પૃષ્ઠની શોધ તપાસે છે. જો તે મળે તો તેને વેબ બ્રાઉઝર પર પાછું આપે છે અને HTTP કનેક્શન બંધ કરો.
- 5 હવે વેબ બ્રાઉઝર વેબ પૃષ્ઠ મેળવે છે, તે તેનું અર્થઘટન કરે છે અને વેબ બ્રાઉઝરની વિંડોમાં વેબ પૃષ્ઠની સામગ્રી પ્રદર્શિત કરે છે.

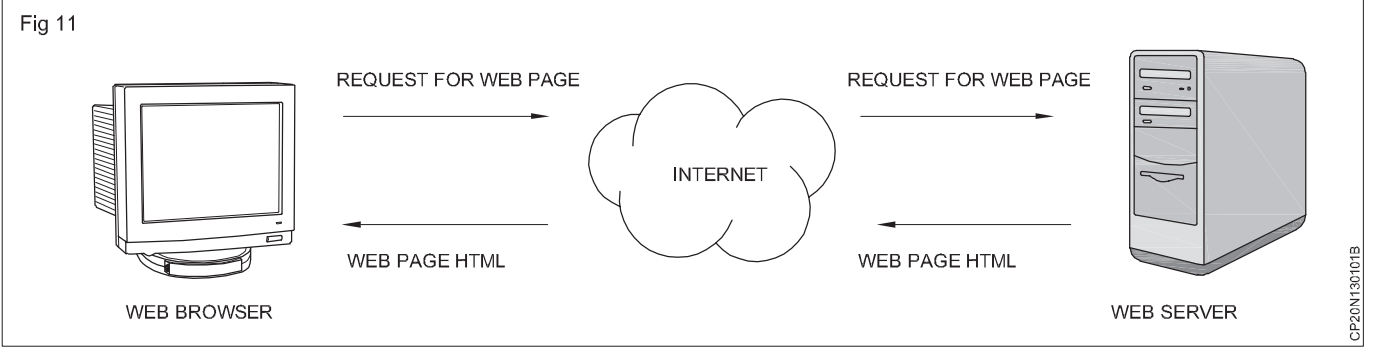
ભાવિ

વેબ ક્ષેત્રે ઝડપી વિકાસ થયો હતો. શિક્ષણ, સંશોધન, ટેકનોલોજી, વાણિજ્ય, માર્કેટિંગ વગેરે જેવા લગભગ દરેક ક્ષેત્રોમાં તેની અસર છે. તેથી વેબનું ભાવિ

લગભગ અણધારી છે. WWW ના ક્ષેત્રમાં વિશાળ વિકાસ ઉપરાંત, કેટલીક તકનીકી સમસ્યાઓ પણ છે જેનો W3 કન્સોર્ટિયમ સમનો કરવો પડે છે.

વપરાશકર્તા ઈન્ટરફેસ (ફિગ 11)

3D માહિતીની ઉચ્ચ ગુણવત્તાની રજૂઆત પર કામ કરો વિકાસ હેઠળ છે. W3 કન્સોર્ટિયમ વૈશ્વિક સમુદાયોની જરૂરિયાતોને પૂર્ણ કરવા માટે વેબને વધારવા માટે પણ આતુર છે જેમાં તમામ પ્રાદેશિક ભાષાઓ અને લેખન પ્રણાલીઓનો સમાવેશ થશે.



ટેકનોલોજી પ્રાઈવસી

અને સિક્યુરિટી પર કામ ચાલી રહ્યું છે. આમાં માહિતી છુપાવવી, એકાઉન્ટિંગ, એક્સેસ કંટ્રોલ, અખંડિતતા અને જોખમ વ્યવસ્થાપનનો સમાવેશ થશે.

આર્કિટેક્ચર

વેબના ક્ષેત્રમાં ભારે વૃદ્ધિ થઈ છે જે ઈન્ટરનેટને ઓવરલોડ કરી શકે છે અને તેનું પ્રદર્શન બગાડી શકે છે. તેથી વધુ સારા પ્રોટોકોલ વિકસાવવાની જરૂર છે.

ઈ મેઈલ (ઈલેક્ટ્રોનિક મેઈલ) શું છે?

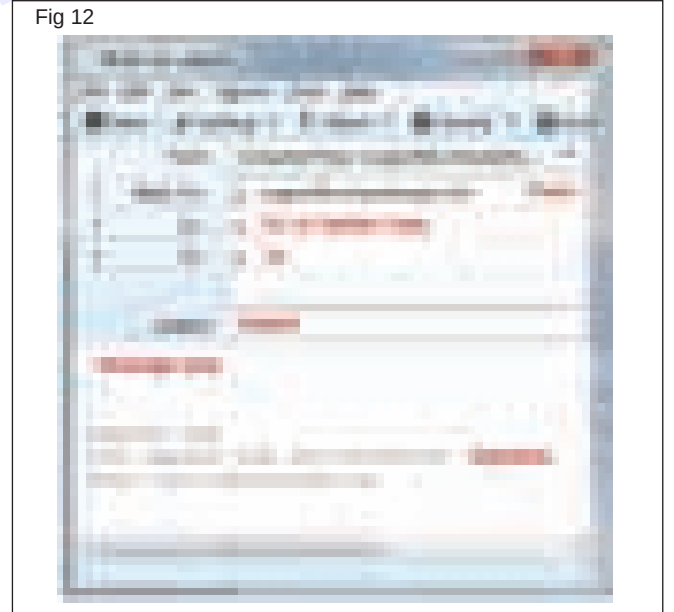
ઈ-મેઈલ (ઈલેક્ટ્રોનિક મેઈલ) એ ઈન્ટરનેટ દ્વારા એક વપરાશકર્તા તરફથી એક અથવા વધુ પ્રાપ્તકર્તાઓને કમ્પ્યુટર - સંગ્રહિત સંદેશોનું વિનિમય છે. ઈ મેલ્સ એ વ્યવસાય અથવા વ્યક્તિગત ઉપયોગ માટે વાતચીત કરવાની ઝડપી, સસ્તી અને સુલભ રીત છે.

ઈ-મેલ લખવું (ફિગ 12)

ઈ-મેલ સંદેશ લખતી વખતે, તે નીચેની ઉદાહરણ વિન્ડોની જેમ કંઈક દેખાવું જોઈએ. જેમ તમે જોઈ શકો છો, ઈ-મેલ મોકલતી વખતે કેટલાક ફીલ્ડ જરૂરી છે:

- ટુ ફીલ્ડ એ છે જ્યાં ઈ-મેલ પ્રાપ્ત કરનાર વ્યક્તિનું ઈ-મેલ સરનામું મૂકવામાં આવે છે.
- ફ્રોમ ફીલ્ડમાં તમારું ઈ-મેલ સરનામું હોવું જોઈએ.
- જો તમે સંદેશનો જવાબ આપી રહ્યા છો, તો પ્રતિ: અને પ્રતિ: ફીલ્ડ્સ આપમેળે થઈ જશે ભરેલ. જો તે નવો સંદેશ છે, તો તમારે પ્રતિ: ફીલ્ડમાં પ્રાપ્તકર્તાઓને તમારી સંપર્ક સૂચિમાંથી પસંદ કરીને અથવા ઈ-મેલ સરનામાં લખીને સ્પષ્ટ કરવાની જરૂર છે. જો તમે એક કરતાં વધુ પ્રાપ્તકર્તા (દા.ત., જૂથ ઈ-મેલ) દાખલ કરો છો, તો સરનામાંને અલ્પવિરામ અને સ્પેસ દ્વારા અથવા ટેબ કી દબાવીને અલગ કરવા જોઈએ.
- વિષયમાં ઈ-મેલની સામગ્રીઓનું વર્ણન કરતા થોડા શબ્દો હોવા જોઈએ. આ વિષય પ્રાપ્તકર્તાને સંપૂર્ણ ઈ-મેલ ખોલ્યા અને વાંચ્યા વિના, ઈ-મેલ શોના વિશે છે તે જોવા દે છે. આ ક્ષેત્ર વૈકલ્પિક છે.

- CC (“કાર્બન કોપી”) ફીલ્ડ તમને પ્રાપ્તકર્તાઓને સ્પષ્ટ કરવા દે છે જેઓ સીધા સરનામાં ન હોય (“ટુ” ફીલ્ડમાં સૂચિબદ્ધ). દાખલા તરીકે, તમે જેફ અને સીસી લિન્ડા અને સ્ટીવનને ઈમેલ મોકલી શકો છો. જો કે ઈમેલ જેફને સંબોધવામાં આવ્યો છે, લિન્ડા અને સ્ટીવનને પણ એક નકલ મળે છે અને દરેક વ્યક્તિ જોઈ શકે છે કે ઈ-મેલ કોને મળ્યો છે. આ ક્ષેત્ર વૈકલ્પિક છે.
- BCC (“બ્લાઈન્ડ કાર્બન કોપી”) ફીલ્ડ CC જેવું જ છે, સિવાય કે પ્રાપ્તકર્તાઓ ગુપ્ત હોય. દરેક BCC પ્રાપ્તકર્તા ઈ-મેલ મેળવે છે, પરંતુ બીજા કોને કોપી મળી છે તે જોતા નથી. આસરનામાંઓ (“પ્રતિ” ફીલ્ડમાં સૂચિબદ્ધ કોઈપણ) બધા પ્રાપ્તકર્તાઓને દૃશ્યક્ષમ રહે છે. આ ક્ષેત્ર વૈકલ્પિક છે.
- છેલ્લે, મેસેજ બોડી એ સ્થાન છે જ્યાં તમે તમારો મુખ્ય સંદેશ લખો છો. તે ઘણીવાર તળિયે તમારી સહી ધરાવે છે; હસ્તાલિખિત પત્ર જેવું જ



વેબ ડિઝાઇનિંગ (Web designing)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- વેબ સાઈટ અને વેબ પેજ્સ સમજાવો
- સ્ટેટિક અને ડાયનેમિક વેબ પેજ્સ સમજાવો
- HTML, DHTML અને XML સમજાવો
- વેબ હોસ્ટિંગ, વેબ સર્વર, એપ્લિકેશન સર્વર અને ડેટાબેઝ સર્વરની વિભાવના સમજાવો.

પરિચય

વર્લ્ડ વાઈડ વેબ (WWW) ની રચના 1990 માં CERN ભૌતિકશાસ્ત્રી ટિમ બર્નર્સ-લી દ્વારા કરવામાં આવી હતી. 30 એપ્રિલ 1993ના રોજ, CERN એ જાહેરાત કરી હતી કે વર્લ્ડ વાઈડ વેબ કોઈપણ વ્યક્તિ માટે ઉપયોગ કરવા માટે મફત હશે. HTML અને HTTP ની રજૂઆત પહેલાં, અન્ય પ્રોટોકોલ જેમ કે ફાઈલ ટ્રાન્સફર પ્રોટોકોલ અને ગોફર પ્રોટોકોલનો ઉપયોગ સર્વરમાંથી વ્યક્તિગત ફાઈલોને પુનઃપ્રાપ્ત કરવા

માટે કરવામાં આવતો હતો. આ પ્રોટોકોલ્સ એક સરળ ડિરેક્ટરી માળખું પ્રદાન કરે છે જે વપરાશકર્તા નેવિગેટ કરે છે અને ડાઉનલોડ કરવા માટે ફાઈલો પસંદ કરે છે. દસ્તાવેજો મોટાભાગે ફોર્મેટિંગ વિના સાદા ટેક્સ્ટ ફાઈલો તરીકે રજૂ કરવામાં આવતા હતા અથવા વર્ડ પ્રોસેસર ફોર્મેટમાં એન્કોડ કરવામાં આવતા હતા.

વેબસાઈટ્સ

વેબસાઈટ, વેબસાઈટ, વેબ સાઈટ અથવા સાદી સાઈટ તરીકે પણ લખાયેલી વેબસાઈટ, લેખિત ભાષા, ઈમેજ, વિડીયો, ધ્વનિ વગેરે જેવી સામગ્રી (મીડિયા) ધરાવતા સંબંધિત વેબ પેજનો સમૂહ છે. વેબસાઈટ ઓછામાં ઓછી એક વેબ પર હોસ્ટ કરવામાં આવે છે. સર્વર, યુનિફોર્મ રિસોર્સ લોકેટર તરીકે ઓળખાતા ઈન્ટરનેટ એક્સ દ્વારા ઈન્ટરનેટ અથવા ખાનગી લોકલ એરિયા નેટવર્ક જેવા નેટવર્ક દ્વારા એક્સેસિબલ. તમામ સાર્વજનિક રીતે સુલભ વેબસાઈટ સામૂહિક રીતે વર્લ્ડ વાઈડ વેબની રચના કરે છે.

વેબપેજ એ એક દસ્તાવેજ છે, જે સામાન્ય રીતે હાયપરટેક્સ્ટ માર્કઅપ લેંગ્વેજ (એચટીએમએલ) એક્સએચટીએમએલની ફોર્મેટિંગ સૂચનાઓ સાથે સાદા લખાણમાં લખાયેલ છે. વેબપેજ યોગ્ય HTML એન્કર સાથે અન્ય વેબસાઈટ્સના ઘટકોને સમાવી શકે છે.

વેબ પૃષ્ઠોને હાઈપરટેક્સ્ટ ટ્રાન્સફર પ્રોટોકોલ (HTTP) સાથે એક્સેસ કરવામાં આવે છે અને પરિવહન કરવામાં આવે છે, જે વેબપેજ સામગ્રીના વપરાશકર્તા માટે સુરક્ષા અને ગોપનીયતા પ્રદાન કરવા માટે વૈકલ્પિક રીતે એન્ક્રિપ્શન સુરક્ષિત, HTTPS નો ઉપયોગ કરી શકે છે. વપરાશકર્તાની એપ્લિકેશન, ઘણીવાર વેબ બ્રાઉઝર, કમ્પ્યુટર મોનિટર પર તેના HTML માર્ક અપ સૂચનાઓ અનુસાર પૃષ્ઠ સામગ્રીને રેન્ડર કરે છે.

વેબસાઈટના પેજ સામાન્ય રીતે વેબ એક્સેસ તરીકે ઓળખાતા સાદા યુનિફોર્મ રિસોર્સ લોકેટર (URL) પરથી એક્સેસ કરી શકાય છે. પૃષ્ઠોના યુઆરએલ તેમને વંશવેલોમાં ગોઠવે છે, જો કે તેમની વચ્ચેની હાઈપરલિંક વાચકના અનુભવેલા સાઈટમેપને જણાવે છે અને વાચકના સાઈટના નેવિગેશનને માર્ગદર્શન આપે છે જેમાં સામાન્ય રીતે સાઈટની વેબ સામગ્રીની મોટાભાગની લિંક્સ સાથેનું હોમ પેજ અને પૃષ્ઠ વિશે પૂરક હોય છે., સંપર્ક પૃષ્ઠ અને લિંક પૃષ્ઠ.

કેટલીક વેબસાઈટ્સને તેમની કેટલીક અથવા બધી સામગ્રીને એક્સેસ કરવા માટે સબસ્ક્રિપ્શનની જરૂર હોય છે. સબસ્ક્રિપ્શન વેબસાઈટ્સના ઉદાહરણોમાં ઘણી બિઝનેસ સાઈટ્સ, ન્યૂઝ વેબસાઈટ્સના ભાગો, એકેડેમિક જર્નલ વેબસાઈટ્સ, ગેમિંગ વેબસાઈટ્સ, ફાઈલ શેરિંગ વેબસાઈટ્સ, ઈન્ટરનેટ ફોરમ, વેબ-આધારિત ઈમેઈલ, સોશિયલ નેટવર્કિંગ વેબસાઈટ્સ, વાસ્તવિક સમયનો સ્ટોક માર્કેટ ડેટા પ્રદાન કરતી વેબસાઈટ્સ અને અન્ય વિવિધ વેબસાઈટ્સનો સમાવેશ થાય છે. સેવાઓ (દા.ત., ઈમેલ્સ, ફાઈલો વગેરેનો સંગ્રહ અને/અથવા શેરિંગ ઓફર કરતી વેબસાઈટ).

વેબસાઈટ હોઈ શકે છે

- વ્યક્તિગત વેબસાઈટ
- વાણિજ્યિક વેબસાઈટ
- ઈ-સરકાર
- બિન-લાભકારી સંસ્થાની વેબસાઈટ.

સ્ટેટિક વેબસાઈટ

સ્ટેટિક વેબસાઈટ એવી છે કે જેમાં વેબ પેજ્સ સર્વર પર સંગ્રહિત ફોર્મેટમાં હોય છે જે ક્લાયંટ વેબ બ્રાઉઝરને મોકલવામાં આવે છે. તે મુખ્યત્વે હાઈપરટેક્સ્ટ માર્કઅપ લેંગ્વેજ (HTML) માં કોડેડ છે.

વેબસાઈટના સરળ સ્વરૂપો અથવા માર્કેટિંગ ઉદાહરણો, જેમ કે ક્લાસિક વેબસાઈટ, પાંચ પાનાની વેબસાઈટ અથવા બ્રોશર વેબસાઈટ ઘણી વખત સ્થિર વેબસાઈટ હોય છે, કારણ કે તે વપરાશકર્તાને પૂર્વ નિર્ધારિત, સ્થિર માહિતી રજૂ કરે છે. આમાં ટેક્સ્ટ, ફોટા, એનિમેશન, ઓડિઓ/વિડિયો અને ઈન્ટરેક્ટિવ મેનૂ અને નેવિગેશન દ્વારા કંપની અને તેના ઉત્પાદનો અને સેવાઓ વિશેની માહિતી શામેલ હોઈ શકે છે.

આ પ્રકારની વેબસાઈટ સામાન્ય રીતે તમામ મુલાકાતીઓને સમાન માહિતી દર્શાવે છે. ગ્રાહકો અથવા ગ્રાહકોને પ્રિન્ટેડ બ્રોશર સોંપવાની જેમ, સ્થિર વેબસાઈટ સામાન્ય રીતે વિસ્તૃત સમય માટે સુસંગત, પ્રમાણભૂત માહિતી પ્રદાન કરશે. જોકે વેબસાઈટ માલિક સમયાંતરે અપડેટ્સ કરી શકે છે, તે ટેક્સ્ટ, ફોટા અને અન્ય સામગ્રીને સંપાદિત કરવાની મેન્યુઅલ પ્રક્રિયા છે અને તેને મૂળભૂત વેબસાઈટ ડિઝાઇન કુશળતા અને સોફ્ટવેરની જરૂર પડી શકે છે.

ફાયદા

- સ્ટેટિક વેબસાઈટ વિકસાવવા માટે સરળ છે
- ઝડપથી વિકસાવી શકાય છે
- તેઓ શોધ એન્જીન દ્વારા સરળતાથી અનુક્રમિત થાય છે કારણ કે તમામ વેબ પૃષ્ઠો ખરેખર સર્વર પર અસ્તિત્વમાં છે, જે ગતિશીલ વેબસાઈટ્સ સાથે કેસ નથી.

ગેરફાયદા

- સ્ટેટિક વેબસાઇટ્સ ઘણી ઓનલાઇન સેવાઓ માટે જરૂરી જટિલ કાર્યો કરી શકતી નથી.
- આખી સાઇટને અપડેટ કરવી બોજાડપ અને સમય માંગી શકે છે.
- સ્ટેટિક વેબસાઇટ્સમાં ડેટા અને ડિઝાઇનનું અલગીકરણ પ્રદાન કરવામાં આવતું નથી.

ડાયનેમિક વેબસાઇટ

ગતિશીલ વેબસાઇટ એવી છે જે અમુક માપદંડોના આધારે વારંવાર અને આપમેળે પોતાને બદલે છે અથવા કસ્ટમાઇઝ કરે છે.

ડાયનેમિક વેબસાઇટ્સમાં બે પ્રકારની ગતિશીલ પ્રવૃત્તિ હોઈ શકે છે: કોડ અને સામગ્રી. ડાયનેમિક કોડ અદ્રશ્ય છે અથવા પડદા પાછળ છે અને ગતિશીલ સામગ્રી દૃશ્યમાન છે અથવા સંપૂર્ણ પ્રદર્શિત છે

ફાયદા

- ઓનલાઇન સેવાઓ માટે જરૂરી વધુ જટિલ કાર્ય કરી શકે છે.
- તેઓ અપડેટ કરવા માટે સરળ છે.
- ડેટા અને ડિઝાઇનને અલગ પાડવાથી પ્રોગ્રામરો અને સામગ્રી લેખકોને સ્વતંત્ર રીતે કામ કરવાની મંજૂરી મળે છે.

ગેરફાયદા

- બિલ્ડ કરવામાં વધુ સમય લાગી શકે છે.
- નિર્માણ કરવું મુશ્કેલ હોઈ શકે છે.
- ડાયનેમિક વેબસાઇટ્સને શોધ એન્જિન દ્વારા સરળતાથી અનુક્રમિત કરવામાં આવતી નથી, કારણ કે તેમની પાસે વેબ સર્વર પર વાસ્તવિક વેબ પૃષ્ઠો હાજર નથી. સર્ચ એન્જિન ટેક્નોલોજીમાં સતત સુધારા સાથે, આ સમસ્યા હવે ઘણી દૂર થઈ ગઈ છે અને તમે શોધી શકો છો કે ઘણી ગતિશીલ વેબસાઇટ્સ આજકાલ સર્ચ એન્જિન દ્વારા ખૂબ જ સારી રીતે અનુક્રમિત છે.

આવા કેટલાક વર્ગીકરણ છે:

- સંલગ્ન: સક્ષમ પોર્ટલ કે જે ફક્ત તેના કસ્ટમ CMS જ નહીં પણ અન્ય સામગ્રી પ્રદાતાઓ તરફથી સંમત ફી માટે સિન્ડિકેટ સામગ્રી પણ રેન્ડર કરે છે. સામાન્ય રીતે સંબંધના ત્રણ સ્તર હોય છે. સંલગ્ન એજન્સીઓ (દા.ત., કમિશન જંકશન), જાહેરાતકર્તાઓ (દા.ત., eBay) અને ઉપભોક્તા (દા.ત., Yahoo!).
- આર્કાઇવ સાઇટ: મૂલ્યવાન ઇલેક્ટ્રોનિક સામગ્રીને સાચવવા માટે વપરાય છે જે લુપ્ત થવાની આરે છે. ઉદાહરણ તરીકે: ઇન્ટરનેટ આર્કાઇવ, જે 1996 થી અબજો જૂના અને નવા વેબ પૃષ્ઠોને સાચવી રાખે છે; અને Google જૂથો, જેણે 2005ની શરૂઆતમાં યુઝનેટ સમાચાર/ચર્ચા જૂથો પર પોસ્ટ કરેલા 845,000,000 સંદેશાઓ સાચવ્યા હતા.
- બ્લોગ સાઇટ: સામાન્ય રીતે ઓનલાઇન ડાયરીઓ, ટિપ્પણીઓ અથવા દૃશ્યો પોસ્ટ કરવા માટે ઉપયોગમાં લેવાતી સાઇટ્સ જેમાં ચર્ચા મંચો (દા.ત., બ્લોગર, Xanga) સામેલ હોઈ શકે છે.
- સામગ્રી સાઇટ: આ સાઇટ્સ અંતિમ વપરાશકર્તાને મૂળ સામગ્રી બનાવે છે અને વેચે છે. (દા.ત., સ્લેટ, About.com).
- કોર્પોરેટ વેબસાઇટ: વ્યવસાય, સંસ્થા અથવા સેવા સંબંધિત માહિતી પ્રદાન કરવા માટે વપરાય છે.

- કોમર્સ સાઇટ (અથવા ઈકોમર્સ સાઇટ): આ સાઇટ્સ Amazon.com, CSN સ્ટોર્સ અને Overstock.com જેવા સામાન્ય બ્રીડી અથવા વેચાણ માટે ડિઝાઇન કરવામાં આવી છે.
- કોમ્યુનિટી સાઇટ: એવી સાઇટ્સ કે જ્યાં સમાન રુચિ ધરાવતી વ્યક્તિઓ ચેટિંગ અને મેસેજિંગ દ્વારા અથવા સામાજિક સંદેશ બોર્ડ, જેમ કે MySpace અથવા Facebook દ્વારા એકબીજા સાથે વાતચીત કરે છે.
- શહેરની સાઇટ: એક સાઇટ કે જે ચોક્કસ શહેર અથવા નગર વિશેની માહિતી અને તે નગરમાં બનતી ઘટનાઓ દર્શાવે છે. સામાન્ય રીતે સિટી કાઉન્સિલ દ્વારા બનાવવામાં આવે છે. ઉદાહરણ તરીકે, Richmond.com એ રિચમોન્ડ, વર્જિનિયા માટે જીઓડોમેન છે.
- માહિતી સાઇટ: તે સામગ્રી સમાવે છે જેનો હેતુ મુલાકાતીઓને જાણ કરવાનો છે, પરંતુ વ્યાપારી હેતુઓ માટે જરૂરી નથી, જેમ કે: RateMyProfessors.com, Free Internet Lexicon અને Encyclopaedia. મોટાભાગની સરકારી, શૈક્ષણિક અને બિન-લાભકારી સંસ્થાઓ પાસે છે માહિતીપ્રદ સાઇટ.
- મિરર સાઇટ: વેબસાઇટનું સંપૂર્ણ પ્રજનન.
- સમાચાર સાઇટ: માહિતીની સાઇટ જેવી જ છે, પરંતુ સમાચાર અને કોમેન્ટરી આપવા માટે સમર્પિત છે.
- વ્યક્તિગત હોમપેજ: વ્યક્તિગત અથવા નાના જૂથ દ્વારા ચલાવવામાં આવે છે જેમ કે કુટુંબ કે જેમાં માહિતી અથવા કોઈપણ સામગ્રી શામેલ હોય જે વ્યક્તિ શામેલ કરવા માંગે છે. આ સામાન્ય રીતે વેબ હોસ્ટિંગ સેવા જેમ કે Geocities નો ઉપયોગ કરીને અપલોડ કરવામાં આવે છે.
- ફિશ સાઇટ: ઇલેક્ટ્રોનિક સંચારમાં વિશ્વાસપાત્ર વ્યક્તિ અથવા વ્યવસાય (જેમ કે સામાજિક સુરક્ષા વહીવટ, પેપાલ) તરીકે છૂપાવીને છેતરપિંડીથી સંવેદનશીલ માહિતી, જેમ કે પાસવર્ડ અને ક્રેડિટ કાર્ડ વિગતો મેળવવા માટે બનાવવામાં આવેલી વેબસાઇટ.
- રાજકીય સાઇટ: એક એવી સાઇટ કે જેના પર લોકો રાજકીય મંતવ્યો રજૂ કરી શકે છે.
- રેટિંગ સાઇટ: એક એવી સાઇટ કે જેના પર લોકો વખાણ કરી શકે છે અથવા જે દર્શાવવામાં આવે છે તેની નિંદા કરી શકે છે.
- રીવ્યુ સાઇટ: એક એવી સાઇટ કે જેના પર લોકો ઉત્પાદનો અથવા સેવાઓ માટે સમીક્ષાઓ પોસ્ટ કરી શકે છે.
- શાળાની સાઇટ: એક એવી સાઇટ કે જેના પર શિક્ષકો, વિદ્યાર્થીઓ અથવા સંચાલકો વર્તમાન ઘટનાઓ વિશે માહિતી પોસ્ટ કરી શકે છે તેમની શાળાને સામેલ કરે છે.
- વિડિયો શેરિંગ: એવી સાઇટ કે જે યુઝરને વીડિયો અપલોડ કરવા સક્ષમ બનાવે છે, જેમ કે YouTube અને Google Video.
- શોધ એન્જિન સાઇટ: એક સાઇટ કે જે સામાન્ય માહિતી પૂરી પાડે છે અને અન્ય સાઇટ પુનઃપ્રાપ્ત કરવા માટેના ગેટવે તરીકે બનાવાયેલ છે. Google, Yahoo અને MSN એ સૌથી વધુ જાણીતા સર્ચ એન્જિન છે.
- શોક સાઇટ: મોટા ભાગના દર્શકો (દા.ત. rotten.com) માટે અપમાનજનક હોવાનો હેતુ ધરાવતી છબીઓ અથવા અન્ય સામગ્રીનો સમાવેશ કરે છે.
- Warez: એક સાઇટ હોસ્ટ કરવા અને વપરાશકર્તાઓને કૉપિરાઇટ કરેલી સામગ્રીને ગેરકાયદેસર રીતે ડાઉનલોડ કરવા દેવા માટે ડિઝાઇન કરવામાં આવી છે.

- વેબ પોર્ટલ: સાઈટ એ વાહન છે જે ઈન્ટરનેટ અથવા ઈન્ટ્રાનેટ પરના અન્ય સંસાધનો માટે ગેટવે પ્રદાન કરે છે.

વેબ પેજીસ

વેબ પેજ અથવા વેબપેજ એ એક દસ્તાવેજ અથવા માહિતી સંસાધન છે જે વિશ્વવ્યાપી વેબ માટે યોગ્ય છે અને વેબ બ્રાઉઝર દ્વારા એક્સેસ કરી શકાય છે અને કમ્પ્યુટર ડિસ્પ્લે અથવા મોબાઇલ ઉપકરણ પર પ્રદર્શિત કરી શકાય છે. આ માહિતી સામાન્ય રીતે HTML અથવા XHTML ફોર્મેટમાં હોય છે, અને હાયપર ટેક્સ્ટ હાયપર લિંક દ્વારા અન્ય વેબ પૃષ્ઠોને નેવિગેશન બાર પ્રદાન કરી શકે છે. વેબ પેજ અવારનવાર અન્ય સંસાધનો જેમ કે કેસ્કેડીંગ સ્ટાઇલ શીટ, ક્લાઇન્ટ-સાઇડ સ્ક્રિપ્ટિંગ અને ઇમેજ્સને તેમની અંતિમ રજૂઆતમાં સમાવે છે.

વેબ પૃષ્ઠો સ્થાનિક કમ્પ્યુટર અથવા દૂરસ્થ વેબ સર્વરમાંથી પુનઃપ્રાપ્ત થઈ શકે છે. વેબ સર્વર ફક્ત ખાનગી નેટવર્કની એક્સેસને પ્રતિબંધિત કરી શકે છે, દા.ત. કોર્પોરેટ ઈન્ટ્રાનેટ અથવા તે વર્લ્ડ વાઇડ વેબ પર પૃષ્ઠો પ્રકાશિત કરી શકે છે. વેબ પૃષ્ઠોની વિનંતી કરવામાં આવે છે અને વેબ પરથી સેવા આપવામાં આવે છે. હાઇપરટેક્સ્ટ ટ્રાન્સફર પ્રોટોકોલ (HTTP) નો ઉપયોગ કરીને વેબ સર્વર.

વેબ પૃષ્ઠોમાં સ્થિર ટેક્સ્ટની ફાઇલો અને વેબ સર્વરની ફાઇલ સિસ્ટમ (સ્થિર વેબ પૃષ્ઠ)માં સંગ્રહિત અન્ય વેબ સામગ્રીનો સમાવેશ થઈ શકે છે, અથવા જ્યારે તેમને વિનંતી કરવામાં આવે ત્યારે સર્વર-સાઇડ સ્ક્રિપ્ટિંગ દ્વારા બનાવવામાં આવી શકે છે (ડાયનેમિક વેબ પૃષ્ઠ). ક્લાયન્ટ-સાઇડ સ્ક્રિપ્ટિંગ વેબ પૃષ્ઠોને ક્લાયન્ટ બ્રાઉઝર પર એકવાર વપરાશકર્તા ઈનપુટ માટે વધુ પ્રતિભાવશીલ બનાવી શકે છે.

વેબ બ્રાઉઝર

વેબ બ્રાઉઝરમાં ઈન્ટરનેટ એક્સપ્લોરર, મોઝિલા ફાયરફોક્સ, ગૂગલ ક્રોમ અને ઓપેરા (વેબ બ્રાઉઝર) જેવા ગ્રાફિકલ યુઝર ઈન્ટરફેસ હોઈ શકે છે અથવા લિંક્સ (વેબ બ્રાઉઝર) (ફિગ 1) અથવા લિંક્સ (વેબ બ્રાઉઝર) જેવા કમાન્ડ લાઇન ઈન્ટરફેસ હોઈ શકે છે. વિકલાંગતા ધરાવતા વેબ વપરાશકર્તાઓ ઘણીવાર વેબ એક્સેસિબિલિટી વેબ પૃષ્ઠો માટે સહાયક તકનીકો અને અનુકૂળનશીલ વ્યૂહરચનાઓનો ઉપયોગ કરે છે. વપરાશકર્તાઓ રંગ અંધ હોઈ શકે છે, કદાચ પુનરાવર્તિત તાણની ઈજા અથવા મોટર-ન્યુરોનની સમસ્યાને લીધે માઉસનો ઉપયોગ કરવા માંગતા નથી અથવા ન પણ ઈચ્છતા હોઈ શકે છે, બહેરા હોઈ શકે છે અને કેન્સન આપવા માટે ઓડિયોની જરૂર પડી શકે છે, અંધ હોઈ શકે છે અને સ્ક્રીન રીડર અથવા ડિસ્પ્લેનો ઉપયોગ કરી શકે છે. સ્ક્રીન મેગ્નિફિકેશન, વગેરે. અક્ષમ અને સક્ષમ શારીરિક વપરાશકર્તાઓ સમય, નેટવર્ક બેન્ડવિડ્થ બચાવવા અથવા ફક્ત તેમના બ્રાઉઝિંગ અનુભવને સરળ બનાવવા માટે, છબીઓ અને અન્ય મીડિયાના ડાઉનલોડ અને જોવાને અક્ષમ કરી શકે છે.

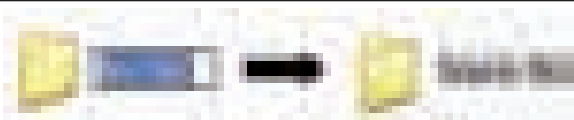
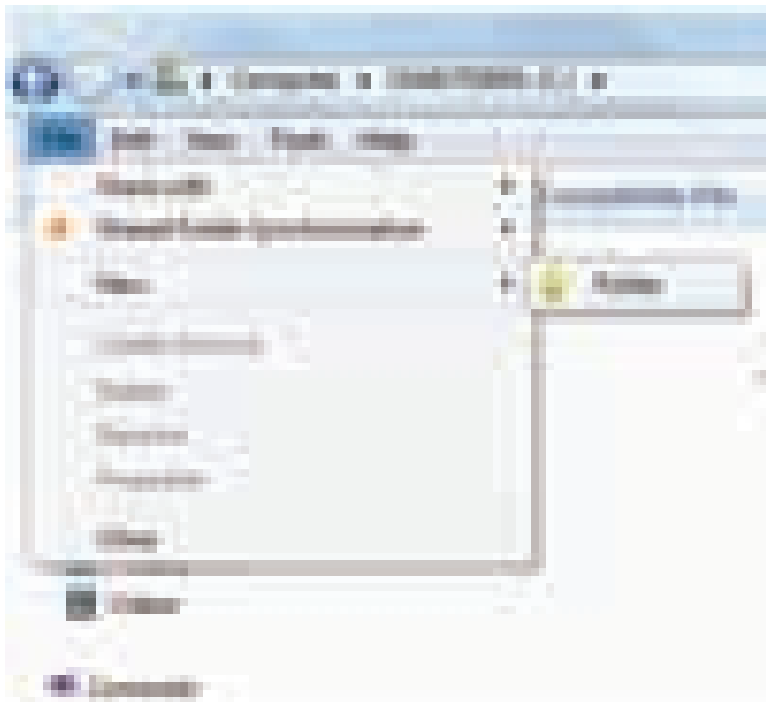
મોબાઇલ ઉપકરણોના વપરાશકર્તાઓ ઘણીવાર પ્રતિબંધિત ડિસ્પ્લે અને બેન્ડવિડ્થ ધરાવે છે. કોઈપણ વ્યક્તિ વેબ પેજ ડિઝાઇનર દ્વારા પસંદ કરાયેલ ફોન્ટ્સ, ફોન્ટ સાઈઝ, સ્ટાઇલ અને રંગ યોજનાઓનો ઉપયોગ ન કરવાનું પસંદ કરી શકે છે અને પેજ પર તેમની પોતાની CSS સ્ટાઇલ લાગુ કરી શકે છે. વર્લ્ડ વાઇડ વેબ કન્સોર્ટિયમ (W3C) અને વેબ એક્સેસિબિલિટી ઈનિશિયેટિવ (WAI) ભલામણ કરે છે કે તમામ વેબ પૃષ્ઠો આ બધા વિકલ્પોને ધ્યાનમાં રાખીને ડિઝાઇન કરવા જોઈએ.



ઈન્ટરનેટ પરથી સોફ્ટવેર ડાઉનલોડ કરી રહ્યા છીએ

- 1 માય કોમ્પ્યુટર ખોલીને એક ટેમ્પરરી ફાઇલ્સ ફોલ્ડર બનાવો, તમારી હાર્ડ ડ્રાઇવ (સામાન્ય રીતે C: ડ્રાઇવ) પર ડબલ ક્લિક કરો, પછી ફિગ 2 મુજબ ફાઇલ/નવું/ફોલ્ડર પસંદ કરો.

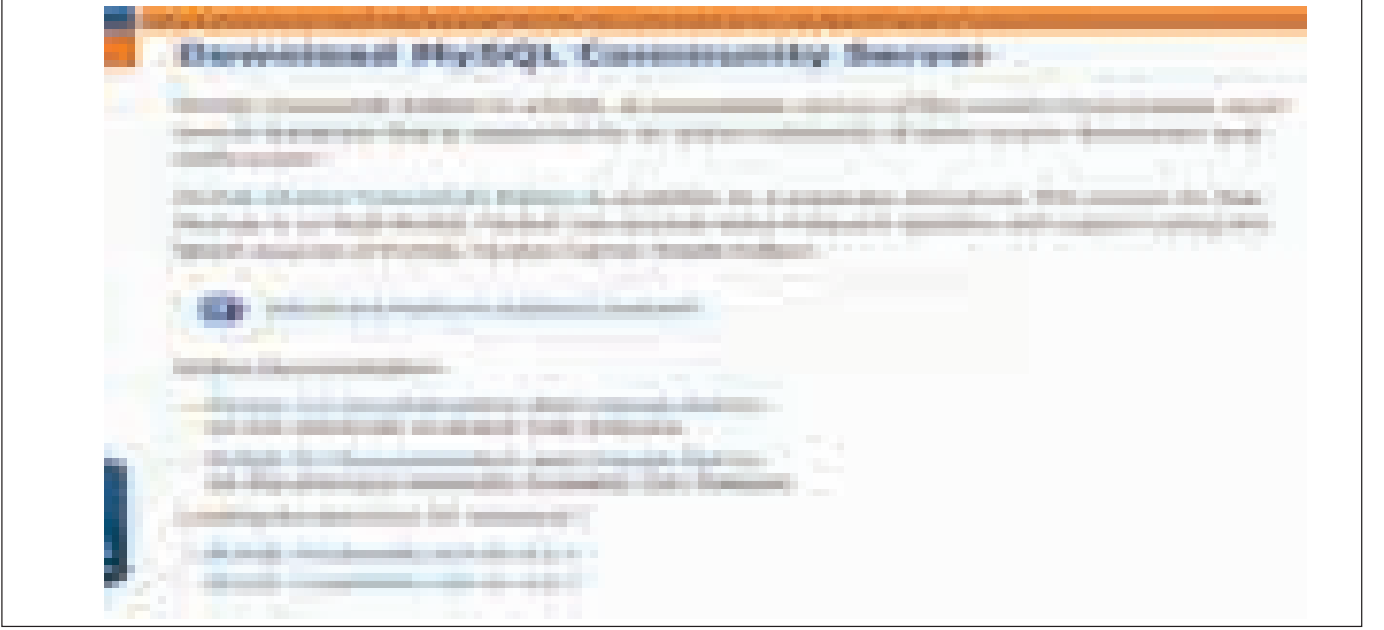
- 2 “ટેમ્પરરી ફાઇલ” ટાઇપ કરો અને તેને ફિગ 3 પ્રમાણે નામ આપો.
- 3 ઈન્ટરનેટ પર ફ્રીવેર સોફ્ટવેર વેબસાઇટ પરથી “માય એસક્યુએલ 5.1 ફ્રી ડાઉનલોડ” ટાઇપ કરો.



નોંધ: ઉદાહરણ તરીકે “[http:// dev.mysql.com/d wnloads/](http://dev.mysql.com/downloads/)” પસંદ કરો, ફિગ 4 મુજબ ખુલેલી સાઈટ પરથી ડાઉનલોડમાં “માય એસક્યુએલ કોમ્યુનિટી સર્વર” પસંદ કરો.



- 4 ફિગ 5 પરના પહેલાના સંસ્કરણ વિકલ્પને શોધીને “MySQL કોમ્યુનિટી સર્વર 5.1” પર ક્લિક કરો



- 5 “Windows (x86,32-bit),MSI ઇન્સ્ટોલર એસેન્શિયલ્સ - ભલામણ કરેલ” પસંદ કરો અને ડાઉનલોડ પર ક્લિક કરો (ફિગ.6).



નોંધ: બનાવેલ ફોલ્ડર નામ “ટેમ્પરરી ફોલ્ડર” માં “My SQL 5.1.63” સાચવો.

- 6 ડાઉનલોડ કરેલ “My SQL 5.1.63” ને CD ROM માં ઇન્સ્ટોલેશન માટે બર્ન કરો.

વેબ ભાષાઓ

વેબ ભાષાઓને માર્કઅપ ભાષાઓ તરીકે ઓળખવામાં આવે છે કારણ કે ટેક્સ્ટની પ્રક્રિયા, વ્યાખ્યા અને પ્રસ્તુતિ માટે ડિઝાઇન કરવામાં આવી છે. ભાષા ટેક્સ્ટ ફાઇલમાં ફોર્મેટિંગ માટે કોડનો ઉલ્લેખ કરે છે, લેઆઉટ અને શૈલી બંને. ફોર્મેટિંગનો ઉલ્લેખ કરવા માટે ઉપયોગમાં લેવાતા કોડને ટેગ્સ કહેવામાં આવે છે

માર્કઅપ ભાષાઓના ચાર પ્રકાર

- 1 BML
- 2 HTML
- 3 DHTML
- 4 XML

BML (બેટર માર્કઅપ લેંગ્વેજ) BML અનિવાર્યપણે એક સરળ મેક્રો ભાષા છે. BML માં મેક્રોને બ્લોક કહેવામાં આવે છે. બ્લોક્સ લુક ફાઇલોમાં વ્યાખ્યાયિત કરવામાં આવે છે, અને BML ફાઇલોમાં બોલાવવામાં આવે છે. બ્લોક પેરામીટર્સ સ્વીકારે છે અને પેરામીટર્સ કેવી રીતે પ્રસારિત થાય છે અને બ્લોકની વ્યાખ્યા તેનો ઉપયોગ કેવી રીતે કરવામાં સક્ષમ છે તેના આધારે તેને ઘણા પ્રકારોમાં વહેંચવામાં આવે છે.

HTML (હાઇપર ટેક્સ્ટ માર્કઅપ લેંગ્વેજ) : HTML અથવા હાઇપરટેક્સ્ટ માર્કઅપ લેંગ્વેજ એ વેબની ભાષા છે. બધા વેબ પૃષ્ઠો HTML માં લખેલા છે. HTML વેબ બ્રાઉઝર્સમાં છબીઓ, મલ્ટીમીડિયા અને ટેક્સ્ટ પ્રદર્શિત થાય તે રીતે વ્યાખ્યાયિત કરે છે. તેમાં દસ્તાવેજો (હાઇપરટેક્સ્ટ) ને જોડવા અને વેબ દસ્તાવેજોને અરસપરસ (જેમ કે ફોર્મ સાથે) બનાવવા માટે ઘટકોનો સમાવેશ થાય છે.

HTML એ વ્યાખ્યાયિત પ્રમાણભૂત માર્કઅપ ભાષા છે. તે ધોરણ વર્લ્ડ વાઇડ વેબ કન્સોર્ટિયમ (W3C) દ્વારા વિકસાવવામાં આવ્યું હતું. તે SGML (સ્ટાન્ડર્ડ જનરલાઇઝ્ડ માર્કઅપ લેંગ્વેજ) પર આધારિત છે. તે એક એવી ભાષા છે જે તમારા ટેક્સ્ટની રચનાને વ્યાખ્યાયિત કરવા માટે ટૅગ્સનો ઉપયોગ કરે છે. તત્વો ટૅગ્સ અને <and> અક્ષરો દ્વારા વ્યાખ્યાયિત કરવામાં આવે છે.

ડીએચટીએમએલ(DHTML) ડાયનેમિક HTML એ ખરેખર HTML નું નવું સ્પષ્ટીકરણ નથી, પરંતુ પ્રમાણભૂત HTML કોડ્સ અને આદેશોને જોવા અને નિયંત્રિત કરવાની એક નવી રીત છે.

ગતિશીલ HTML વિશે વિચારતી વખતે, આપણે પ્રમાણભૂત HTML ના ગુણો યાદ રાખવાની જરૂર છે, ખાસ કરીને કે એકવાર સર્વરમાંથી પૃષ્ઠ લોડ થઈ જાય, તે સર્વર પર બીજી વિનંતી ન આવે ત્યાં સુધી તે બદલાશે નહીં. ડાયનેમિક HTML એ HTML તત્વો પર વધુ નિયંત્રણ આપે છે અને વેબ સર્વર પર પાછા ફર્યા વિના તેમને કોઈપણ સમયે બદલવાની મંજૂરી આપે છે.

DHTML ના ચાર ભાગો છે:

- દસ્તાવેજ ઓબ્જેક્ટ મોડલ (DOM) (વ્યાખ્યા)
- સ્ક્રિપ્ટો
- કેસ્કેડિંગ સ્ટાઇલ શીટ્સ (CSS)
- XHTML

DOM : DOM વેબ પેજના કોઈપણ ભાગને DHTML સાથે બદલવા માટે તેને એક્સેસ કરવાની મંજૂરી આપે છે. વેબ પેજના દરેક ભાગને DOM દ્વારા નિર્દિષ્ટ કરવામાં આવે છે અને તેના સુસંગત નામકરણ સંમેલનોનો ઉપયોગ કરીને તેમને એક્સેસ કરી શકાય છે અને તેમની મિલકતો બદલી શકાય છે.

સ્ક્રિપ્ટો: JavaScript અથવા ActiveX માં લખેલી સ્ક્રિપ્ટો DHTML ને સક્રિય કરવા માટે ઉપયોગમાં લેવાતી બે સૌથી સામાન્ય સ્ક્રિપ્ટિંગ ભાષાઓ છે. તમે DOM માં ઉલ્લેખિત ઓબ્જેક્ટ્સને નિયંત્રિત કરવા માટે સ્ક્રિપ્ટિંગ ભાષાનો ઉપયોગ કરો છો.

કેસ્કેડિંગ સ્ટાઇલ શીટ્સ: વેબ પેજના દેખાવ અને અનુભૂતિને નિયંત્રિત કરવા માટે DHTML માં CSS નો ઉપયોગ થાય છે. સ્ટાઇલ શીટ્સ ટેક્સ્ટના રંગો અને ફોન્ટ્સ, પૃષ્ઠભૂમિના રંગો અને છબીઓ અને પૃષ્ઠ પર ઓબ્જેક્ટ્સની પ્લેસમેન્ટને વ્યાખ્યાયિત કરે છે. સ્ક્રિપ્ટિંગ અને DOM નો ઉપયોગ કરીને, આપણે વિવિધ તત્વોની શૈલી બદલી શકીએ છીએ.

એક્સએચટીએમએલ(XHTML): XHTML અથવા HTML 4.x નો ઉપયોગ પેજ બનાવવા અને CSS અને DOM પર કામ કરવા માટે તત્વો બનાવવા માટે થાય છે. ડીએચટીએમએલ માટે એક્સએચટીએમએલ વિશે કંઈ ખાસ નથી - પરંતુ માન્ય એક્સએચટીએમએલ હોવું એ પણ વધુ મહત્વનું છે, કારણ કે તેમાં ફક્ત બ્રાઉઝર કરતાં વધુ વસ્તુઓ કામ કરે છે.

DHTML ની વિશેષતાઓ

DHTML ના ચાર પ્રાથમિક લક્ષણો છે:

- 1 ટૅગ્સ અને ગુણધર્મો બદલવું

- 2 રીઅલ-ટાઇમ પોઝિશનિંગ

- 3 ડાયનેમિક ફોન્ટ્સ (નેટસ્કેપ કોમ્યુનિકેટર)

- 4 ડેટા બંધનકર્તા (ઇન્ટરનેટ એક્સપ્લોરર)

ટૅગ્સ અને પ્રોપર્ટીઝ બદલવી

આ DHTML નો સૌથી સામાન્ય ઉપયોગ છે. તે બ્રાઉઝરની બહારની ઘટના (જેમ કે માઉસ ક્લિક, સમય અથવા તારીખ વગેરે)ના આધારે HTML ટેગના ગુણોને બદલવાની મંજૂરી આપે છે. અમે આનો ઉપયોગ પૃષ્ઠ પર માહિતીને પહેલાથી લોડ કરવા માટે કરી શકીએ છીએ, અને જ્યાં સુધી વાચક ચોક્કસ લિંક પર ક્લિક ન કરે ત્યાં સુધી તેને પ્રદર્શિત કરી શકતા નથી.

રીઅલ-ટાઇમ પોસ્ટિંગ

ઓબ્જેક્ટ્સ, છબીઓ અને ટેક્સ્ટ વેબ પૃષ્ઠની આસપાસ ફરતા હોય છે. આનાથી અમે વાચકો સાથે ઇન્ટરેક્ટિવ ગેમ્સ રમી શકીએ છીએ અથવા સ્ક્રીનના ભાગોને એનિમેટ કરી શકીએ છીએ.

ડાયનેમિક ફોન્ટ્સ

આ માત્ર નેટસ્કેપ ફીચર છે. નેટસ્કેપે ડિઝાઇનર્સને વાચકની સિસ્ટમ પર કયા ફોન્ટ્સ હશે તે જાણતા ન હોવાથી સમસ્યાનો સામનો કરવા માટે આનો વિકાસ કર્યો હતો. ગતિશીલ ફોન્ટ્સ સાથે, ફોન્ટ્સ એન્કોડ કરવામાં આવે છે અને પૃષ્ઠ સાથે ડાઉનલોડ કરવામાં આવે છે, જેથી પૃષ્ઠ હંમેશા જુએ છે કે ડિઝાઇનર તેને કેવી રીતે ઇચ્છે છે.

ડેટા બંધનકર્તા : આ એક માત્ર IE સુવિધા છે. માર્કેટસોફ્ટે વેબસાઇટમાંથી ડેટાબેસેસની સરળ એક્સેસ આપવા માટે આને વિકસાવ્યું છે. તે ડેટાબેઝને એક્સેસ કરવા માટે CGI નો ઉપયોગ કરવા જેવું જ છે, પરંતુ કાર્ય કરવા માટે ActiveX નિયંત્રણનો ઉપયોગ કરે છે.

XML: એક્સટેન્સિબલ માર્કઅપ લેંગ્વેજ (XML) એ એક માર્કઅપ લેંગ્વેજ છે જે દસ્તાવેજોને એન્કોડ કરવા માટેના નિયમોના સમૂહને એવા ફોર્મેટમાં વ્યાખ્યાયિત કરે છે જે માનવ વાંચી શકાય અને મશીન વાંચી શકાય. તે W3C દ્વારા ઉત્પાદિત XML 1.0 સ્પેસિફિકેશનમાં વ્યાખ્યાયિત થયેલ છે, અને અન્ય કેટલાક સંબંધિત સ્પષ્ટીકરણો, તમામ મફતમાં ખુલ્લા ધોરણો. XML ના ડિઝાઇન લક્ષ્યો ઇન્ટરનેટ પર સરળતા, સામાન્યતા અને ઉપયોગિતા પર ભાર મૂકે છે. તે વિશ્વની ભાષાઓ માટે યુનિકોડ દ્વારા મજબૂત સમર્થન સાથેનું એક ટેક્સ્ટ્યુઅલ ડેટા ફોર્મેટ છે. XML ની ડિઝાઇન દસ્તાવેજો પર ધ્યાન કેન્દ્રિત કરતી હોવા છતાં, તેનો ઉપયોગ મનસ્વી ડેટા સ્ટ્રક્ચર્સની રજૂઆત માટે વ્યાપકપણે થાય છે, ઉદાહરણ તરીકે વેબ સેવાઓમાં

HTML દસ્તાવેજ બનાવી રહ્યા છીએ : વેબ પેજ લખવા માટે કોડ લખવાનું શરૂ કરતા પહેલા, વેબ પેજના દેખાવની યોજના બનાવવી એ સારી પ્રથા છે. HTML દસ્તાવેજમાં બે ઘટકો હોય છે:

- 1 દસ્તાવેજ સામગ્રી

- 2 ટૅગ્સ

દસ્તાવેજ સામગ્રી એ વેબ પૃષ્ઠ પરની માહિતી છે જે વપરાશકર્તા જોશે. તે માહિતી ટેક્સ્ટ અથવા ગ્રાફિક્સ હોઈ શકે છે.

ટૅગ્સ એ HTML કોડ છે જે દસ્તાવેજ સામગ્રી કેવી રીતે દેખાશે તેનું નિયંત્રણ કરે છે. ટૅગ્સ, બીજા શબ્દોમાં, નિર્ધારિત કરશે કે ટેક્સ્ટ બોલ્ડ, કાળો કે વાદળી અથવા ફોન્ટ પ્રકાર ટાઇમ ન્યૂ રોમન અથવા એરલ હશે.

નોટપેડ શરૂ કરો

નોટપેડ શરૂ કરવા માટે આના પર જાઓ:

શરૂઆત

બધા કાર્યક્રમો

એસેસરીઝ

નોટપેડ

નોટપેડ વડે તમારું HTML સંપાદિત કરો (ફિગ 7)

તમારા નોટપેડમાં તમારો HTML કોડ લખો:

તમારું HTML સાચવો

નોટપેડના ફાઇલ મેનુમાં સેવ એઝ પસંદ કરો.

જ્યારે તમે HTML ફાઇલ સાચવો છો, ત્યારે તમે .htm નો ઉપયોગ કરી શકો છો

અથવા .html ફાઇલ એક્સ્ટેન્શન.

ફાઇલને યાદ રાખવામાં સરળ હોય તેવા ફોલ્ડરમાં સાચવો

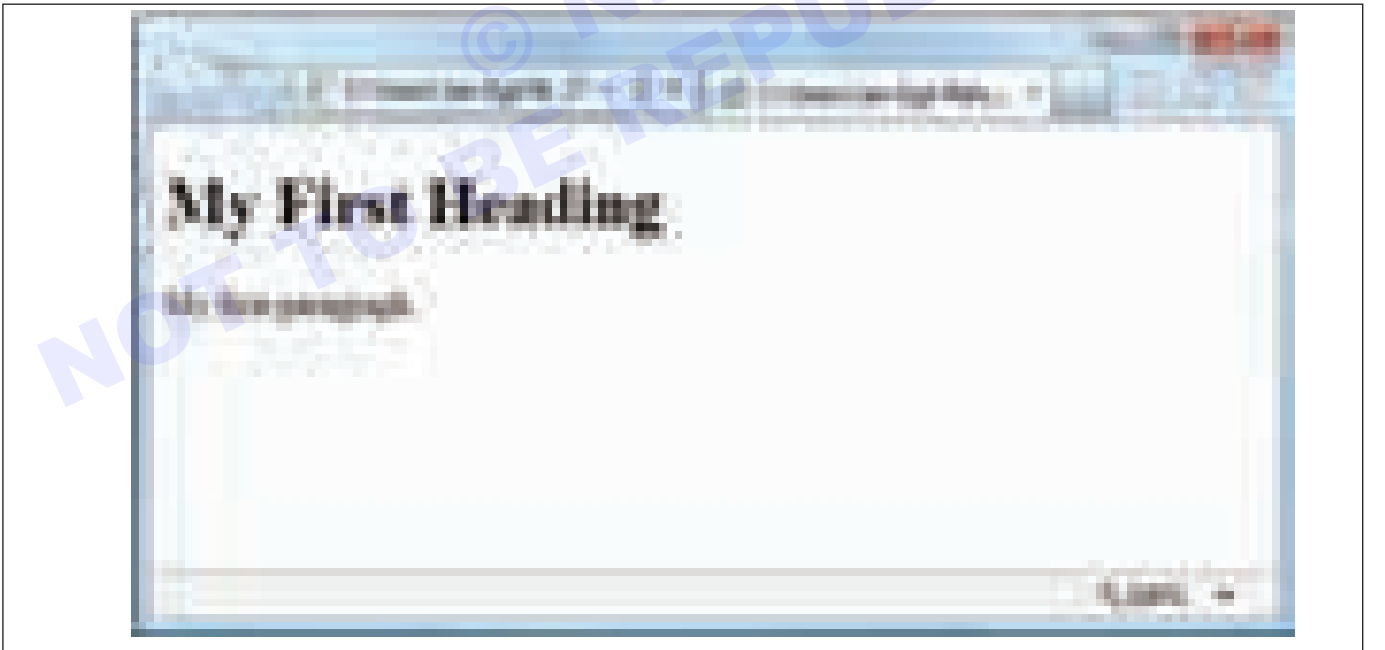
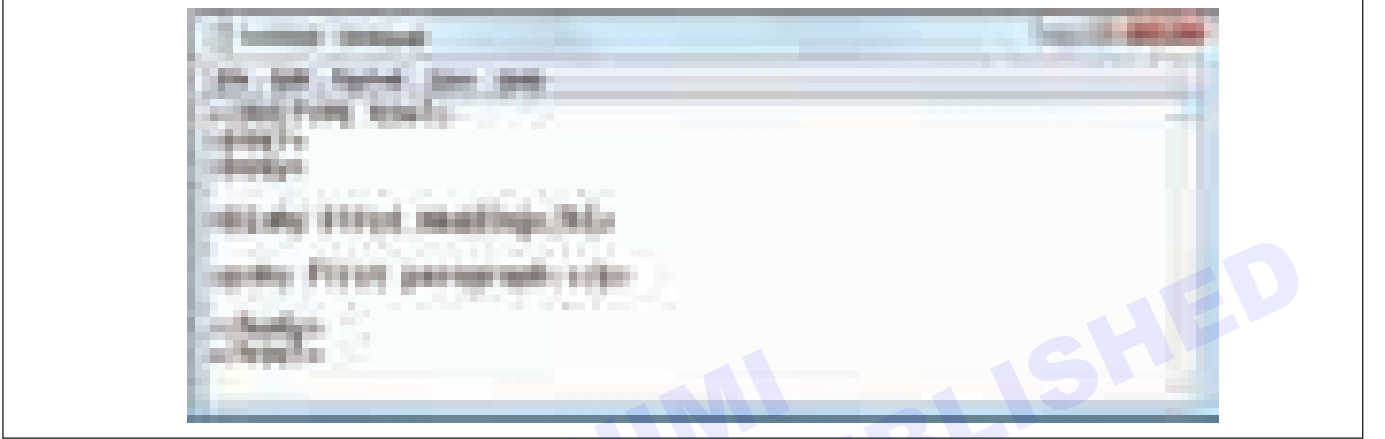
તમારા બ્રાઉઝરમાં HTML ચલાવો

તમારું વેબ બ્રાઉઝર શરૂ કરો અને માંથી તમારી html ફાઇલ ખોલો

ફાઇલ કરો, મેનૂ ખોલો અથવા ફક્ત ફોલ્ડરને બ્રાઉઝ કરો અને ડબલ ક્લિક કરો

તમારી HTML ફાઇલ.

પરિણામ આના જેવું દેખાવું જોઈએ: (ફિગ 8)



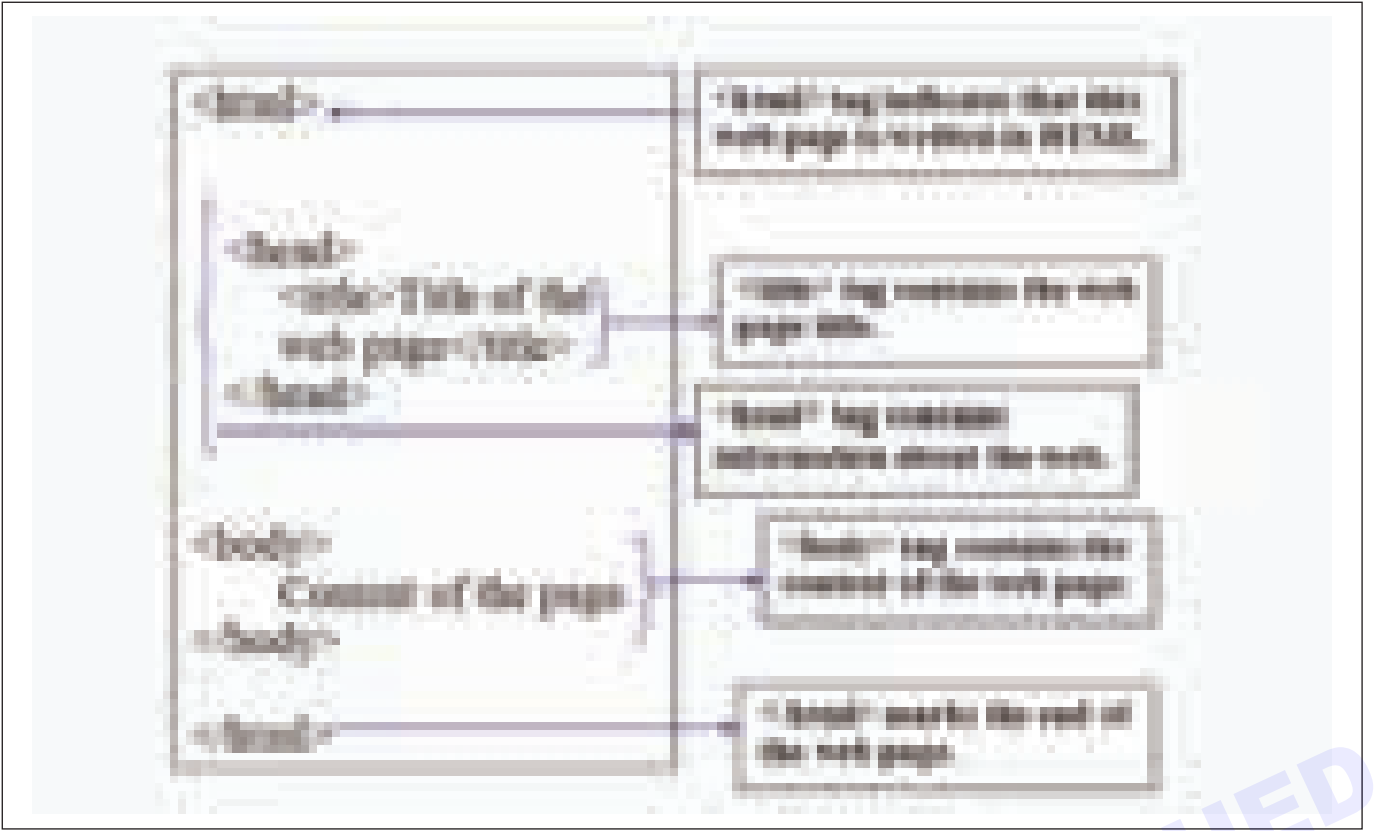
માર્કઅપ લેંગ્વેજનું માળખું

HTML દસ્તાવેજમાં બે* મુખ્ય ભાગો હોય છે:

- 1 માથું. હેડ એલિમેન્ટ વેબ દસ્તાવેજનું શીર્ષક અને મેટા ડેટા ધરાવે છે.
- 2 શરીર. બોડી એલિમેન્ટમાં તે માહિતી હોય છે જે તમે વેબ પેજ પર પ્રદર્શિત કરવા માંગો છો.

તમારા વેબ પૃષ્ઠોને HTML 4 સાથે સુસંગત બનાવવા માટે, તમારે HTML તત્વ પહેલાં દસ્તાવેજ પ્રકાર ઘોષણા (DTD) ઉમેરવાની જરૂર છે. જ્યારે તમે નવું વેબ પેજ બનાવો છો ત્યારે ઘણા વેબ ઓથરીંગ સોફ્ટવેર કીટીકી અને મૂળભૂત ટેગ્સ આપમેળે ઉમેરે છે.

વેબ પૃષ્ઠમાં, પ્રથમ ટેગ (ખાસ કરીને,) દસ્તાવેજ માટે ઉપયોગમાં લેવાતી માર્કઅપ ભાષા સૂચવે છે. આ ટેગ વેબ પૃષ્ઠ વિશેની માહિતી ધરાવે છે. છેલ્લે, સામગ્રી ટેગમાં દેખાય છે. (ફિગ 9)



<!DOCTYPE> ઘોષણા

વેબ પર ઘણાં વિવિધ દસ્તાવેજો છે. બ્રાઉઝર માત્ર દસ્તાવેજને યોગ્ય રીતે પ્રદર્શિત કરી શકે છે, જો તેને ખબર હોય કે તે કયા પ્રકારનો દસ્તાવેજ છે.

HTML ના ઘણાં વિવિધ સંસ્કરણો પણ છે, અને બ્રાઉઝર ફક્ત HTML પૃષ્ઠને 100% યોગ્ય રીતે પ્રદર્શિત કરી શકે છે જો તે પૃષ્ઠમાં વપરાયેલ ચોક્કસ <!DOCTYPE> ને જાણતું હોય. આ તે છે જેનો ઉપયોગ થાય છે.

<!DOCTYPE> ટેગ નથી. HTML કયામાં લખાયેલ છે તે વિશે તે બ્રાઉઝરને માહિતી (ઘોષણા) છે.

The HTML <head> Element

એ તમામ <head> માટેનું કન્ટેનર છે. અંદરના ઘટકોમાં સ્ક્રિપ્ટ્સ શામેલ હોઈ શકે છે, બ્રાઉઝરને સૂચના આપી શકે છે કે સ્ટાઇલ શીટ્સ ક્યાં શોધવી, મેટા માહિતી પ્રદાન કરવી અને વધુ.

નીચેના ટેગ્સ હેડ વિભાગમાં ઉમેરી શકાય છે: <title>, <base>, <link>, <meta>, <script>, and <style>.

The HTML <title> Element

<title> ટેગ દસ્તાવેજના શીર્ષકને વ્યાખ્યાયિત કરે છે.

બધા HTML/XHTML દસ્તાવેજોમાં શીર્ષક જરૂરી છે.

શીર્ષક:

- બ્રાઉઝર ટૂલબારમાં શીર્ષક વ્યાખ્યાયિત કરે છે.
- જ્યારે તે મનપસંદમાં ઉમેરવામાં આવે ત્યારે પૃષ્ઠ માટે શીર્ષક પ્રદાન કરે છે.
- શોધ એન્જિન પરિણામોમાં પૃષ્ઠ માટે શીર્ષક દર્શાવે છે.

HTML એલિમેન્ટ સિન્ટેક્સ

- HTML એલિમેન્ટ સ્ટાર્ટ ટેગ / ઓપનિંગ ટેગથી શરૂ થાય છે
- HTML એલિમેન્ટ એન્ડ ટેગ/ક્લોઝિંગ ટેગ સાથે સમાપ્ત થાય છે
- તત્વની સામગ્રી એ શરૂઆત અને અંતિમ ટેગ વચ્ચેની દરેક વસ્તુ છે
- કેટલાક HTML ઘટકોમાં ખાલી સામગ્રી છે
- સ્ટાર્ટ ટેગમાં ખાલી તત્વો બંધ છે
- મોટાભાગના HTML ઘટકોમાં વિશેષતાઓ હોઈ શકે છે

HTML હેડિંગ

HTML શીર્ષકોને ટેગ્સ સાથે વ્યાખ્યાયિત કરવામાં આવે છે.

ઉદાહરણો

<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>

પરિણામ

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 Paragraphs

HTML paragraphs are defined with the <p> tag.

```
<html>
```

```
<body>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

ઉદાહરણો

This is a paragraph.

This is a paragraph.

This is a paragraph.

HTML Links

HTML links are defined with the <a> tag.

```
<html>
```

```
<body>
```

```
<a href="http://www.facebook.com">
```

```
This is a link</a>
```

```
</body>
```

```
</html>
```

Result

This is a link

By clicking the link it shows the facebook login page

HTML Images

HTML images are defined with the tag.

```
<html>
```

```
<body>
```

```

```

```
</body>
```

```
</html>
```

Result (Fig 10)



HTML લક્ષણો

- HTML તત્વોમાં વિશેષતાઓ હોઈ શકે છે
- વિશેષતાઓ તત્વ વિશે વધારાની માહિતી પ્રદાન કરે છે
- વિશેષતાઓ હંમેશા સ્ટાર્ટ ટેગમાં ઉલ્લેખિત હોય છે
- વિશેષતાઓ નામ/મૂલ્ય જોડીમાં આવે છે જેમ કે: name="value"

વિશેષતા ઉદાહરણ

HTML લિંક્સ ટેગ સાથે વ્યાખ્યાયિત થયેલ છે. લિંક સરનામું href એટ્રિબ્યુટમાં ઉલ્લેખિત છે:

```
<html>
```

```
<body>
```

```
<a href="http://www.yahoo.com">
```

```
This is a link</a>
```

```
</body>
```

```
</html>
```

પરિણામ

આ લિંક છે લિંક પર ક્લિક કરવાથી yahoo હોમ પેજ દેખાય છે (ફિગ 11).

ફોર્મેટિંગ

લાઈન બ્રેક્સ બનાવો - તત્વ:

જ્યારે પણ
, તેને અનુસરતું કંઈપણ આગલી લીટી પર શરૂ થાય છે. આ ટેગ એ ખાલી તત્વનું ઉદાહરણ છે, જ્યાં તમારે ઓપનિંગ અને ક્લોઝિંગ ટેગ્સની જરૂર નથી, કારણ કે તેમની વચ્ચે જવા માટે કંઈ નથી.

ઉદાહરણ:

```
Hello<br/>
```

```
You come most carefully upon your hour.<br/>
```

```
Thanks<br/>
```

```
Mahnaz
```

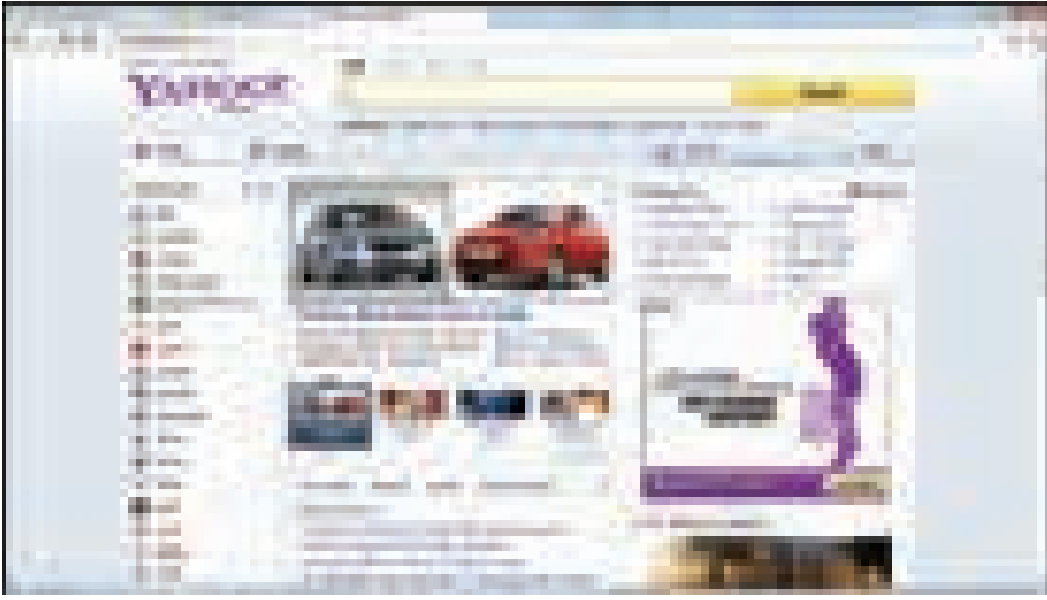
પરિણામ

```
Hello
```

```
You come most carefully upon your hour.
```

```
Thanks
```

```
Mahnaz
```



બનવું

કેન્દ્રીય <center> :

તમે પૃષ્ઠની મધ્યમાં અથવા કોઈપણ કોષ્ટક કોષમાં કોઈપણ સામગ્રી મૂકવા માટે ટેગનો ઉપયોગ કરી શકો છો.

ઉદાહરણ:

```
<p>This is not in the center.</p>
```

```
<center>
```

```
<p>This is in the center.</p>
```

```
</center>
```

This will produce following result:

This is not in the center.

This is in the center.

સોફ્ટ હાઈફન્સ:

પ્રસંગોપાત, તમે ફકરાને વધુ સારી રીતે ન્યાયી ઠેરવવા માટે બ્રાઉઝરને લાંબા શબ્દો હાઈફનેટ કરવાની મંજૂરી આપવા માંગો છો. ઉદાહરણ તરીકે, નીચેના કોડ અને તેના પરિણામી આઉટપુટને ધ્યાનમાં લો.

```
<p style="text-align: justify;"> The morbid fear of  
the number 13, or triskaidekaphobia, has plagued  
some important historic figures like Mahamiya and  
Nanao.</p>
```

આ નીચેના પરિણામ લાવશે:

Example for soft hyphen - The morbid fear of the
number 13, or triskaidekaphobia, has plagued some
important historic figures like Mahamiya and Nanao.

ફોર્મેટિંગ સાથવો - <pre>

કેટલીકવાર તમે ઈચ્છો છો કે તમારો ટેક્સ્ટ HTML દસ્તાવેજમાં કેવી રીતે લખાયેલ છે તેના ચોક્કસ ફોર્મેટને અનુસરે. તે કિસ્સાઓમાં, તમે પ્રીફોર્મેટેડ

ટેગનો ઉપયોગ કરી શકો છો(<pre>) ઓપનિંગ ટેગ અને ક્લોઝિંગ ટેગ વચ્ચેનો કોઈપણ ટેક્સ્ટ સ્રોત દસ્તાવેજના ફોર્મેટિંગને સાચવશે.

```
<pre>
```

```
function testFunction( strText ){
```

```
alert (strText)
```

```
}
```

```
</pre>
```

આ નીચેના પરિણામ લાવશે:

```
function testFunction( strText ){
```

```
alert (strText)
```

```
}
```

આડા નિયમો - <hr/>

આડા નિયમોનો ઉપયોગ દસ્તાવેજના વિભાગોને દૃષ્ટિની રીતે વિભાજિત કરવા માટે થાય છે. ટેગ દસ્તાવેજમાં વર્તમાન સ્થિતિથી જમણા હાંસિયા સુધી એક રેખા બનાવે છે અને તે મુજબ રેખાને તોડે છે.

ઉદાહરણ તરીકે તમે નીચે પ્રમાણે બે ફકરાઓ વચ્ચે એક લીટી આપવા માંગો છો:

```
<p>This is paragraph one and should be on top</p>
```

```
<hr />
```

```
<p>This is paragraph two and should be at bottom</p>
```

```
p>
```

This will produce following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

ફરીથી ટેગ એ ખાલી તત્વનું ઉદાહરણ છે, જ્યાં તમારે ખોલવાની જરૂર નથી

ફરીથી ટેગ એ ખાલી તત્વનું ઉદાહરણ છે, જ્યાં તમારે ઓપનિંગ અને ક્લોઝિંગ ટેગની જરૂર નથી, કારણ કે તેમની વચ્ચે જવા માટે કંઈ નથી.

નોંધ: < hr> અને ફોરવર્ડ સ્લેશ અક્ષરો વચ્ચે જગ્યા ધરાવે છે. જો તમે આ જગ્યા છોડો છો, તો જૂના બ્રાઉઝર્સને લાઈન બ્રેક રેન્ડર કરવામાં મુશ્કેલી પડશે, જ્યારે તમે ફોરવર્ડ સ્લેશ અક્ષર ચૂકી જશો અને માત્ર તેનો ઉપયોગ કરો છો તો તે માન્ય XHTML નથી.

પ્રેઝન્ટેશનલ ટેગ્સ:

જો તમે વર્ડ પ્રોસેસરનો ઉપયોગ કરો છો, તો તમે ટેક્સ્ટને બોલ્ડ, ઇટાલિક અથવા અન્ડરલાઈન કરવાની ક્ષમતાથી પરિચિત છો; HTML અને XHTML માં ટેક્સ્ટ કેવી રીતે દેખાઈ શકે તે દર્શાવવા માટે ઉપલબ્ધ દસ વિકલ્પોમાંથી આ ફક્ત ત્રણ છે.

Bold Text - The Element:

દેખાતી કોઈપણ વસ્તુ બોલ્ડમાં પ્રદર્શિત થાય છે, જેમ કે અહીં બોલ્ડ શબ્દ:

<p>The following word uses a bold typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ બોલ્ડ ટાઈપફેસનો ઉપયોગ કરે છે.

Italic Text - The <i> Element:

જે કંઈપણ દેખાય છે તે ત્રાંસી રંગમાં પ્રદર્શિત થાય છે, જેમ કે અહીં ત્રાંસા કરેલ શબ્દ:

<p>The following word uses a <i>italicized</i> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ ઇટાલિક ટાઈપફેસનો ઉપયોગ કરે છે.

Underlined Text - The <u> Element:

જે કંઈપણ દેખાય છે તે અન્ડરલાઈન સાથે પ્રદર્શિત થાય છે, જેમ કે અહીં રેખાંકિત શબ્દ:

<p>The following word uses a <u>underlined</u> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ રેખાંકિત ટાઈપફેસનો ઉપયોગ કરે છે.

Strike Text - The <strike> Element:

જે કંઈપણ દેખાય છે તે સ્ટ્રાઈકથ્રુ સાથે પ્રદર્શિત થાય છે, જે ટેક્સ્ટ દ્વારા પાતળી રેખા છે:

<p>The following word uses a <strike> strikethrough </strike> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ સ્ટ્રાઈકથ્રુ ટાઈપફેસનો ઉપયોગ કરે છે.

Monospaced font - The <tt> Element:

તત્વની સામગ્રી મોનોસ્પેસ ફોન્ટમાં લખવામાં આવે છે. મોટાભાગના ફોન્ટ્સને ચલ પહોળાઈના ફોન્ટ્સ તરીકે ઓળખવામાં આવે છે કારણ કે વિવિધ અક્ષરો વિવિધ પહોળાઈના હોય છે (ઉદાહરણ તરીકે, અક્ષર m

અક્ષર i કરતાં પહોળો છે). મોનોસ્પેસ ફોન્ટમાં, જોકે, દરેક અક્ષર સમાન પહોળાઈ ધરાવે છે.

<p>The following word uses a <tt>monospaced</tt> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ મોનોસ્પેસ ટાઈપફેસનો ઉપયોગ કરે છે.

Superscript Text - The <sup> Element:

તત્વની સામગ્રી સુપરસ્ક્રિપ્ટમાં લખવામાં આવે છે; વપરાયેલ ફોન્ટનું કદ તેની આસપાસના અક્ષરો જેટલું જ છે પરંતુ અન્ય અક્ષરો કરતાં અડધા અક્ષરની ઊંચાઈ દર્શાવવામાં આવે છે.

<p>The following word uses a^{superscript} typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ સુપરસ્ક્રિપ્ટ ટાઈપફેસનો ઉપયોગ કરે છે.

Subscript Text - The <sub> Element:

સામગ્રી સબસ્ક્રિપ્ટમાં લખાયેલ છે; વપરાયેલ ફોન્ટનું કદ તેની આસપાસના અક્ષરો જેટલું જ છે, પરંતુ અન્ય અક્ષરોની નીચે અડધા અક્ષરની ઊંચાઈ દર્શાવવામાં આવે છે.

Larger Text - The <big> Element:

<big> આસપાસના બાકીના લખાણ કરતાં એક ફોન્ટ સાઈઝ મોટામાં પ્રદર્શિત થાય છે.

<p>The following word uses a <big>big</big> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ એક મોટો ટાઈપફેસ છે.

Smaller Text - The <small> Element:

<small> તેની આસપાસના બાકીના લખાણ કરતાં એક ફોન્ટ સાઈઝ નાની દર્શાવવામાં આવે છે.

<p>The following word uses a <small>small</small> typeface.</p>

આ નીચેના પરિણામ લાવશે:

નીચેનો શબ્દ નાનો ટાઈપફેસ વાપરે છે.

CSS સાથે HTML સ્ટાઈલ

HTML તત્વોને વધુ સારી રીતે શૈલી આપવા માટે, HTML 4 સાથે CSS રજૂ કરવામાં આવ્યું હતું.

CSS ને HTML માં નીચેની રીતે ઉમેરી શકાય છે:

- ઇનલાઈન - HTML ઘટકોમાં શૈલી વિશેષતાનો ઉપયોગ કરીને
- આંતરિક - વિભાગમાં તત્વોનો ઉપયોગ કરીને
- બાહ્ય - બાહ્ય CSS ફાઈલનો ઉપયોગ કરીને

<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>
```

પરિણામ:

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 હાઇપરલિંક્સ (Links)

હાયપરલિંક (અથવા લિંક) એ એક શબ્દ, શબ્દોનું જૂથ અથવા છબી છે જેના પર તમે નવા દસ્તાવેજ અથવા વર્તમાન દસ્તાવેજમાંના નવા વિભાગ પર જવા માટે ક્લિક કરી શકો છો.

લિંક્સ <a> ટેગનો ઉપયોગ કરીને HTML માં ઉલ્લેખિત છે.

<a> ટેગનો ઉપયોગ બે રીતે કરી શકાય છે:

- 1 href એટ્રિબ્યુટનો ઉપયોગ કરીને, અન્ય દસ્તાવેજની લિંક બનાવવા માટે
- 2 નામ વિશેષતાનો ઉપયોગ કરીને દસ્તાવેજની અંદર બુકમાર્ક બનાવવા માટે

The HTML code for a link is simple. It looks like this:

```
<a href="url">Link text</a>
```

The href attribute specifies the destination of a link.

```
<a href="http://www.yahoo.com/">Visit yahoo</a>
```

જે આના જેવું દેખાશે: yahoo.com ની મુલાકાત લો

આ હાઇપરલિંક પર ક્લિક કરવાથી યુઝરને Yahoo હોમપેજ પર મોકલવામાં આવશે.

“લિંક ટેક્સ્ટ” એ ટેક્સ્ટ હોવું જરૂરી નથી. તે છબી અથવા અન્ય કોઈપણ HTML હોઈ શકે છે.

HTML લિંક્સ - લક્ષ્ય વિશેષતા

લક્ષ્ય વિશેષતા સ્પષ્ટ કરે છે કે લિંક કરેલ દસ્તાવેજ ક્યાં ખોલવો.

નીચેનું ઉદાહરણ લિંક કરેલ દસ્તાવેજને નવી બ્રાઉઝર વિન્ડોમાં અથવા નવી ટેબમાં ખોલશે:

ઉદાહરણ

```
<a href="http://www.yahoo.com/" target="_blank">Visit
yahoo !</a>
<html>
```

```
<body>
```

```
<a href="http://www.yahoo.com" target="_blank">Visit
yahoo.com!</a>
```

```
<p>If you set the target attribute to “_blank”, the link will
open in a new browser window/tab.</p>
```

```
</body>
```

```
</html>
```

પરિણામ

Visit yahoo.com!

જો તમે લક્ષ્ય વિશેષતા “_blank” પર સેટ કરો છો, તો લિંક નવી બ્રાઉઝર વિન્ડો/ટેબમાં ખુલશે.

HTML ઇમેજીસ - ઇ ટેગ અને એસઆરસી એટ્રિબ્યુટ

HTML માં, છબીઓને ટેગ સાથે વ્યાખ્યાયિત કરવામાં આવે છે.

ટેગ ખાલી છે, જેનો અર્થ છે કે તેમાં માત્ર વિશેષતાઓ છે અને તેમાં કોઈ કલોઝિંગ ટેગ નથી.

પૃષ્ઠ પર છબી પ્રદર્શિત કરવા માટે, તમારે src વિશેષતાનો ઉપયોગ કરવાની જરૂર છે. Src નો અર્થ “સ્રોત” છે. src લક્ષણનું મૂલ્ય તમે પ્રદર્શિત કરવા માંગો છો તે છબીનું URL છે.

છબીને વ્યાખ્યાયિત કરવા માટે વાક્યરચના:

```

```

URL તે સ્થાન તરફ નિર્દેશ કરે છે જ્યાં છબી સંગ્રહિત છે. “www.w3schools. om” પર “images” ડિરેક્ટરીમાં સ્થિત “bamboo. gif” નામની છબીનું URL છે: http:// www.background abs.com/index. ph ?search=bamboo.

બ્રાઉઝર ઇમેજ પ્રદર્શિત કરે છે જ્યાં દસ્તાવેજમાં ટેગ આવે છે. જો તમે બે ફકરા વચ્ચે ઇમેજ ટેગ મૂકો છો, તો બ્રાઉઝર પ્રથમ ફકરો, પછી ઇમેજ અને પછી બીજો ફકરો બતાવે છે.

HTML છબીઓ - Alt એટ્રિબ્યુટ

જો ઇમેજ પ્રદર્શિત કરી શકાતી નથી, તો જરૂરી alt એટ્રિબ્યુટ ઇમેજ માટે વૈકલ્પિક ટેક્સ્ટનો ઉલ્લેખ કરે છે.

Alt એટ્રિબ્યુટનું મૂલ્ય લેખક દ્વારા નિર્ધારિત ટેક્સ્ટ છે:

```

```

જો કોઈ કારણસર વપરાશકર્તા તેને જોઈ શકતો નથી (ધીમા કનેક્શનને કારણે, src એટ્રિબ્યુટમાં ભૂલ અથવા જો વપરાશકર્તા સ્ક્રીન રીડરનો ઉપયોગ કરે છે) તો Alt એટ્રિબ્યુટ છબી માટે વૈકલ્પિક માહિતી પ્રદાન કરે છે.

HTML છબીઓ - એક છબીની ઊંચાઈ અને પહોળાઈ સેટ કરો

ઊંચાઈ અને પહોળાઈના લક્ષણોનો ઉપયોગ ઇમેજની ઊંચાઈ અને પહોળાઈને સ્પષ્ટ કરવા માટે થાય છે.

વિશેષતા મૂલ્યો ડિફોલ્ટ રૂપે પિક્સેલ્સમાં ઉલ્લેખિત છે:

```

```


નોંધ: ઇમેજ માટે ઉંચાઈ અને પહોળાઈ બંને લક્ષણોનો ઉલ્લેખ કરવો એ સારી પ્રથા છે. જો આ વિશેષતાઓ સેટ કરેલ હોય, તો જ્યારે પેજ લોડ થાય ત્યારે ઇમેજ માટે જરૂરી જગ્યા આરક્ષિત હોય છે. જો કે, આ વિશેષતાઓ વિના, બ્રાઉઝર છબીનું કદ જાણતું નથી. અસર એ થશે કે લોડિંગ દરમિયાન પૃષ્ઠ લેઆઉટ બદલાશે (જ્યારે છબીઓ લોડ થાય છે).

જો HTML ફાઈલમાં દસ છબીઓ હોય તો - પૃષ્ઠને જમણે દર્શાવવા માટે અગિયાર ફાઈલો જરૂરી છે. છબીઓ લોડ કરવામાં સમય લાગે છે, તેથી મારી શ્રેષ્ઠ સલાહ છે: છબીઓનો કાળજીપૂર્વક ઉપયોગ કરો.

જ્યારે વેબ પૃષ્ઠ લોડ થાય છે, ત્યારે તે બ્રાઉઝર છે, તે સમયે, તે ખરેખર વેબ સર્વરમાંથી છબી મેળવે છે અને તેને પૃષ્ઠમાં દાખલ કરે છે. (ફિગ 12) તેથી, ખાતરી કરો કે છબીઓ વેબ પેજના સંબંધમાં વાસ્તવમાં તે જ સ્થાને રહે

છે, અન્યથા તમે મુલાકાતીઓને તૂટેલી લિંક આઈકોન મળશે. જો બ્રાઉઝર છબી શોધી શકતું નથી તો તૂટેલી લિંક આયકન બતાવવામાં આવે છે.

```
<html>
<body>
<h2>Friendship Card</h2>

</body>
</html>
```



HTML કોષ્ટકો

<html>કોષ્ટકો ટેગ સાથે વ્યાખ્યાયિત થયેલ છે.

<td> ટેગ એ “ટેબલ ડેટા” માટે વપરાય છે અને ડેટા સેલની સામગ્રી ધરાવે છે. ટેગમાં ટેક્સ્ટ, લિંક્સ, છબીઓ, સૂચિઓ, ફોર્મ્સ, અન્ય કોષ્ટકો વગેરે શામેલ હોઈ શકે છે.

કોષ્ટક ઉદાહરણ

```
<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>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

પંક્તિ 1, કોષ 1 પંક્તિ 1, કોષ 2

પંક્તિ 2, કોષ 1 પંક્તિ 2, કોષ 2

HTML કોષ્ટકો અને બોર્ડર એટ્રિબ્યુટ

જો તમે બોર્ડર એટ્રિબ્યુટનો ઉલ્લેખ કરતા નથી, તો ટેબલ કિનારીઓ વિના પ્રદર્શિત થશે. કેટલીકવાર આ ઉપયોગી થઈ શકે છે, પરંતુ મોટાભાગે, અમે સરહદો બતાવવા માંગીએ છીએ.

કિનારીઓ સાથે કોષ્ટક પ્રદર્શિત કરવા માટે, સરહદ વિશેષતાનો ઉલ્લેખ કરો:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

HTML કોષ્ટક હેડરો

કોષ્ટકમાં હેડર માહિતી <th> ટેગ સાથે વ્યાખ્યાયિત કરવામાં આવે છે.

બધા મુખ્ય બ્રાઉઝર્સ એલિમેન્ટમાંના ટેક્સ્ટને બોલ્ડ અને કેન્દ્રિત તરીકે <th> પ્રદર્શિત કરે છે.

```
<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>
```

તમારા બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row2, cell2

એચટીએમએલ અવ્યવસ્થિત યાદીઓ

અવ્યવસ્થિત સૂચિ ટેગથી શરૂ થાય છે. દરેક સૂચિ આઈટમ

ટેગ સાથે શરૂ થાય છે.

યાદી વસ્તુઓ બુલેટ (સામાન્ય રીતે નાના કાળા વર્તુળો) સાથે ચિહ્નિત થયેલ છે.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

- Coffee • Milk

HTML ઓર્ડર કરેલ યાદીઓ

ઓર્ડર કરેલ સૂચિ ટેગથી શરૂ થાય છે. દરેક સૂચિ આઈટમ ટેગથી શરૂ થાય છે.

યાદી વસ્તુઓ નંબરો સાથે ચિહ્નિત થયેલ છે.

```
<ol>
```

```
<li>Coffee</li>
```

```
<li>Milk</li>
```

```
</ol>
```

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

- 1 Coffee
- 2 Milk

HTML વ્યાખ્યા યાદીઓ

વ્યાખ્યા સૂચિ એ વસ્તુઓની સૂચિ છે, જેમાં દરેક આઈટમનું વર્ણન હોય છે.

<dl> ટેગ વ્યાખ્યા સૂચિને વ્યાખ્યાયિત કરે છે.

<dt> ટેગનો ઉપયોગ (સૂચિમાંની આઈટમને વ્યાખ્યાયિત કરે છે) અને <dd> (સૂચિમાંની આઈટમનું વર્ણન કરે છે):

```
<dl>
<dt>Coffee</dt>
<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
</dl>
```

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

Coffee - black hot drink

Milk- white cold drink

નોંધ : સૂચિની આઈટમની અંદર તમે ટેક્સ્ટ, લાઈન બ્રેક્સ, છબીઓ, લિંક્સ, અન્ય સૂચિઓ વગેરે મૂકી શકો છો.

HTML યાદી ટેગ્સ

ટેગ	વર્ણન
	ઓર્ડર કરેલ સૂચિ વ્યાખ્યાયિત કરે છે
	અવ્યવસ્થિત સૂચિ વ્યાખ્યાયિત કરે છે
	સૂચિ આઈટમ વ્યાખ્યાયિત કરે છે
<dl>	વ્યાખ્યા સૂચિ વ્યાખ્યાયિત કરે છે
<dt>	વ્યાખ્યા સૂચિમાં આઈટમ વ્યાખ્યાયિત કરે છે
<dd>	વ્યાખ્યા સૂચિમાં આઈટમનું વર્ણન વ્યાખ્યાયિત કરે છે

HTML <div> અને સાથે જૂથબદ્ધ કરી શકાય છે

HTML બ્લોક

મોટાભાગના HTML ઘટકોને બ્લોક સ્તરના ઘટકો અથવા ઈનલાઈન ઘટકો તરીકે વ્યાખ્યાયિત કરવામાં આવે છે.

જ્યારે બ્રાઉઝરમાં પ્રદર્શિત થાય છે ત્યારે બ્લોક લેવલ તત્વો સામાન્ય રીતે નવી લાઈન સાથે શરૂ થાય છે (અને સમાપ્ત થાય છે).

ઉદાહરણો: <h1>, <p>, , <table>

HTML ઈનલાઈન

ઈનલાઈન તત્વો સામાન્ય રીતે નવી લાઈન શરૂ કર્યા વિના પ્રદર્શિત થાય છે.

ઉદાહરણો: , <td>, <a>,

HTML <div> એલિમેન્ટ

HTML <div> એલિમેન્ટ એ બ્લોક લેવલ એલિમેન્ટ છે જેનો ઉપયોગ અન્ય HTML ઘટકોને જૂથબદ્ધ કરવા માટે કન્ટેનર તરીકે કરી શકાય છે.

<div> એલિમેન્ટ વિશેષ અર્થ નથી. તે સિવાય, કારણ કે તે બ્લોક સ્ટરનું તત્વ છે, બ્રાઉઝર તેની પહેલા અને પછી લાઈન બ્રેક પ્રદર્શિત કરશે.

જ્યારે CSS સાથે એકસાથે ઉપયોગ થાય છે, ત્યારે તત્વનો ઉપયોગ સામગ્રીના મોટા બ્લોક્સમાં શૈલીના લક્ષણો સેટ કરવા માટે થઈ શકે છે.

તત્વનો બીજો સામાન્ય ઉપયોગ, દસ્તાવેજ લેઆઉટ માટે છે. તે કોષ્ટકોનો ઉપયોગ કરીને લેઆઉટને વ્યાખ્યાયિત કરવાની “જૂની રીત” ને બદલે છે. કોષ્ટકોનો ઉપયોગ એ તત્વનો સાચો ઉપયોગ નથી. તત્વનો હેતુ ટેબ્યુલર ડેટા પ્રદર્શિત કરવાનો છે.

HTML એલિમેન્ટ

HTML એલિમેન્ટ એ ઈનલાઈન એલિમેન્ટ છે જેનો ઉપયોગ ટેક્સ્ટ માટે કન્ટેનર તરીકે થઈ શકે છે.

HTML નો કોઈ વિશેષ અર્થ નથી

જ્યારે CSS સાથે એકસાથે ઉપયોગ કરવામાં આવે છે, ત્યારે નો ઉપયોગ ટેક્સ્ટના ભાગોમાં શૈલી વિશેષતાઓ સેટ કરવા માટે થઈ શકે છે.

HTML ગ્રુપિંગ ટેગ્સ

ટેગ	વર્ણન
<div>	div વ્યાખ્યાયિત કરે છે
	span વ્યાખ્યાયિત કરે છે

HTML લેઆઉટ

વેબસાઈટ લેઆઉટ

મોટાભાગની વેબસાઈટ્સે તેમની સામગ્રી બહુવિધ કોલમમાં મૂકી છે (મેગેઝિન અથવા અખબારની જેમ ફોર્મેટ કરેલ).

બહુવિધ કોલમ્સ <div> અથવા <table> ઉપયોગ કરીને બનાવવામાં આવે છે. CSS નો ઉપયોગ તત્વોને સ્થાન આપવા અથવા પૃષ્ઠો માટે પૃષ્ઠભૂમિ અથવા રંગીન દેખાવ બનાવવા માટે થાય છે.

HTML લેઆઉટ - <div> ઉપયોગ કરવો

div એલિમેન્ટ એ બ્લોક લેવલ એલિમેન્ટ છે જેનો ઉપયોગ HTML એલિમેન્ટ્સને જૂથબદ્ધ કરવા માટે થાય છે.

નીચેના ઉદાહરણમાં એક બહુવિધ કોલમ લેઆઉટ બનાવવા માટે પાંચ div ઘટકોનો ઉપયોગ કરવામાં આવ્યો છે, જે અગાઉના ઉદાહરણની જેમ જ પરિણામ બનાવે છે:

ઉદાહરણ

```
<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 Web Page</h1></div>
<div id="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;">
Content goes here</div>
<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
</div>
</div>
</body>
</html>
```

Result (Fig 13)



HTML ફોર્મ્સ

HTML ફોર્મનો ઉપયોગ સર્વર પર ડેટા પાસ કરવા માટે થાય છે.

ફોર્મમાં ટેક્સ્ટ ફીલ્ડ, ચેકબોક્સ, રેડિયો બટન, સબમિટ બટન અને વધુ જેવા ઇનપુટ તત્વો હોઈ શકે છે. ફોર્મમાં પસંદગીની યાદીઓ, ટેક્સ્ટરિયા, ફીલ્ડસેટ, દંતકથા અને લેબલ તત્વો પણ હોઈ શકે છે.

<form>ટેગનો ઉપયોગ HTML ફોર્મ બનાવવા માટે થાય છે:

<form>.

input elements.

</form>

HTML <form> - ઇનપુટ એલિમેન્ટ

સૌથી મહત્વપૂર્ણ ફોર્મ એલિમેન્ટ એ ઇનપુટ એલિમેન્ટ છે.

ઇનપુટ તત્વનો ઉપયોગ વપરાશકર્તાની માહિતી પસંદ કરવા માટે થાય છે.

પ્રકાર વિશેષતાના આધારે ઇનપુટ તત્વ ઘણી રીતે બદલાઈ શકે છે. ઇનપુટ તત્વ ટેક્સ્ટ ફીલ્ડ, ચેકબોક્સ, પાસવર્ડ, રેડિયો બટન, સબમિટ બટન અને વધુ પ્રકારનું હોઈ શકે છે.

સૌથી વધુ ઉપયોગમાં લેવાતા ઇનપુટ પ્રકારો નીચે વર્ણવેલ છે.

ટેક્સ્ટ ફીલ્ડ્સ

<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>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

First name:

Last name:

નોંધ: ફોર્મ પોતે દેખાતું નથી. એ પણ નોંધ કરો કે ટેક્સ્ટ ફીલ્ડની ડિફોલ્ટ પહોળાઈ 20 અક્ષરો છે.

પાસવર્ડ ફીલ્ડ

<input type="password" /> defines a password field: <form>

Password: <input type="password" name="pwd" /> </form>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

Password:

નોંધ: પાસવર્ડ ફીલ્ડમાં અક્ષરો મેસ કરેલા છે (ફૂદડી અથવા વર્તુળો તરીકે બતાવવામાં આવે છે)

રેડિયો બટનો

રેડિયો બટન વ્યાખ્યાયિત કરે છે. રેડિયો બટનો વપરાશકર્તાને મર્યાદિત સંખ્યામાં પસંદગીઓમાંથી માત્ર એક પસંદ કરવા દે છે:

<form>

<input type="radio" name="sex" value="male" /> Male

<input type="radio" name="sex" value="female" /> Female

</form>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

☐ Male

☐ Female

ચેકબોક્સ

ચેકબોક્સ વ્યાખ્યાયિત કરે છે. ચેકબોક્સીસ વપરાશકર્તાને મર્યાદિત સંખ્યામાં પસંદગીઓમાંથી એક અથવા વધુ વિકલ્પો પસંદ કરવા દે છે.

<form>

<input type="checkbox" name="vehicle" value="Bike" />

I have a bike

<input type="checkbox" name="vehicle" value="Car" /> I

have a car

</form>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

☐ I have a bike

☐ I have a car

સબમિટ બટન

<input type="submit" /> સબમિટ બટન વ્યાખ્યાયિત કરે છે.

સબમિટ બટનનો ઉપયોગ સર્વર પર ફોર્મ ડેટા મોકલવા માટે થાય છે. ફોર્મની ક્રિયા વિશેષતામાં ઉલ્લેખિત પૃષ્ઠ પર ડેટા મોકલવામાં આવે છે. ક્રિયા વિશેષતામાં વ્યાખ્યાયિત ફાઇલ સામાન્ય રીતે પ્રાપ્ત ઇનપુટ સાથે કંઈક કરે છે:

<form name="input" action="html_form_action.asp" method="get">

Username: <input type="text" name="user" />

<input type="submit" value="Submit" />

</form>

બ્રાઉઝરમાં ઉપરનો HTML કોડ કેવો દેખાય છે:

Username: Submit

જો તમે ઉપરના ટેક્સ્ટ ફીલ્ડમાં કેટલાક અક્ષરો લખો છો, અને “સબમિટ કરો” બટનને ક્લિક કરો છો, તો બ્રાઉઝર તમારું ઇનપુટ “html_form_action.asp” નામના પૃષ્ઠ પર મોકલશે. પૃષ્ઠ તમને પ્રાપ્ત ઇનપુટ બતાવશે.

HTML Iframes

iframe ઉમેરવા માટે સિન્ટેક્સ:

<iframe src="URL"></iframe>

URL અલગ પૃષ્ઠના સ્થાનને નિર્દેશ કરે છે.

Iframe - સેટ ઊંચાઈ અને પહોળાઈ

ઊંચાઈ અને પહોળાઈના લક્ષણોનો ઉપયોગ iframeની ઊંચાઈ અને પહોળાઈનો ઉલ્લેખ કરવા માટે થાય છે.

વિશેષતા મૂલ્યો ડિફોલ્ટ રૂપે પિક્સેલ્સમાં ઉલ્લેખિત છે, પરંતુ તે ટકાવારીમાં પણ હોઈ શકે છે (જેમ કે “80%”).

ઉદાહરણ

```
<!DOCTYPE html>
<html>
<body>
<iframe src="demo_iframe.htm" width="200"
height="200"></iframe>
</body>
</html>
```

તે ફિગ 14 માં બતાવ્યા પ્રમાણે દેખાશે.

Iframe - સરહદ દૂર કરો

ફ્રેમબોર્ડર એટ્રિબ્યુટ સ્પષ્ટ કરે છે કે iframe ની આસપાસ બોર્ડર પ્રદર્શિત કરવી કે નહીં.

સરહદ દૂર કરવા માટે વિશેષતા મૂલ્યને “0” પર સેટ કરો:

ઉદાહરણ

```
<iframe src="demo_iframe.htm" frameborder="0"></
iframe>
```

લિંક માટે લક્ષ્ય તરીકે iframe નો ઉપયોગ કરો

એક iframe એક લિંક માટે લક્ષ્ય ફ્રેમ તરીકે ઉપયોગ કરી શકાય છે.

લિંકના લક્ષ્ય વિશેષતાએ iframe ના નામ વિશેષતાનો સંદર્ભ આપવો જોઈએ:



ઉદાહરણ

```
<iframe src="demo_iframe.htm" name="iframe_a"></
iframe>
<p><a href="http://www.yahoo.com"
target="iframe_a">yahoo.com</a></p>
```

HTML iframe ટેગ

ટેગ	વર્ણન
<iframe>	ઈનલાઈન સબ વિન્ડો વ્યાખ્યાયિત કરે છે (ફ્રેમ)

Color Values (Fig 15)

```
<!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>
```

પરિણામ

Color set by using hex value

Color set by using rgb value

Color set by using color name

બ્રાઉઝર્સ હંમેશા HTML પૃષ્ઠોમાં જગ્યાઓ કાપી નાખશે. જો તમે તમારા લખાણમાં 10 જગ્યાઓ લખો છો, તો બ્રાઉઝર તેમાંથી 9 ને દૂર કરશે, પૃષ્ઠ પ્રદર્શિત કરતા પહેલા. તમારા ટેક્સ્ટમાં જગ્યાઓ ઉમેરવા માટે, તમે અક્ષર એન્ટિટીનો ઉપયોગ કરી શકો છો.

HTML ઉપયોગી પાત્ર સંસ્થાઓ

નોંધ: એન્ટિટીના નામ કેસ સંવેદનશીલ છે!

HTML યુનિકોર્મ રિસોર્સ લોકેટર

URL એ વેબ સરનામા માટેનો બીજો શબ્દ છે.

Result	Description	Entity Name	Entity Number
	non-breaking	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
•	euro	€	€
§	section	§	§
©	copyright	©	©
®	registered trademark	®	®
™	trademark	™	™

URL શબ્દોથી બનેલું હોઈ શકે છે, જેમ કે “w3schools.com”, અથવા ઈન્ટરનેટ પ્રોટોકોલ (IP) સરનામું: 192.68.20.50. મોટાભાગના લોકો સર્ફિંગ કરતી વખતે વેબસાઈટનું નામ દાખલ કરે છે, કારણ કે નંબરો કરતાં નામો યાદ રાખવાનું સરળ હોય છે.

URL - યુનિકોર્મ રિસોર્સ લોકેટર

જ્યારે તમે HTML પૃષ્ઠની લિંક પર ક્લિક કરો છો, ત્યારે અંતર્ગત ટેગ વર્લ્ડ વાઈડ વેબ પરના સરનામાં તરફ નિર્દેશ કરે છે.

યુનિકોર્મ રિસોર્સ લોકેટર (URL) નો ઉપયોગ વર્લ્ડ વાઈડ વેબ પર દસ્તાવેજ (અથવા અન્ય ડેટા) ને સંબોધવા માટે થાય છે.

સમજૂતી:

- **યોજના** - ઈન્ટરનેટ સેવાના પ્રકારને વ્યાખ્યાયિત કરે છે. સૌથી સામાન્ય પ્રકાર http છે
- **હોસ્ટ** - ડોમેન હોસ્ટને વ્યાખ્યાયિત કરે છે (http માટે ડિફોલ્ટ હોસ્ટ www છે)
- **ડોમેન** - ઈન્ટરનેટ ડોમેન નામને વ્યાખ્યાયિત કરે છે, જેમ કે w3schools.com
- **પોર્ટ** - હોસ્ટ પર પોર્ટ નંબર વ્યાખ્યાયિત કરે છે (http માટે ડિફોલ્ટ પોર્ટ નંબર 80 છે)
- **પાથ** - સર્વર પર પાથ વ્યાખ્યાયિત કરે છે (જો અવગણવામાં આવે તો, દસ્તાવેજ વેબ સાઈટની રૂટ ડિરેક્ટરીમાં સંગ્રહિત હોવો જોઈએ)
- **ફાઈલનામ** - દસ્તાવેજ/રિસોર્સનું નામ વ્યાખ્યાયિત કરે છે

સામાન્ય URL યોજનાઓ

નીચેનું કોષ્ટક કેટલીક સામાન્ય યોજનાઓની યાદી આપે છે:

Scheme	Short for....	Which pages will the scheme be used for...
http	HyperText Transfer Protocol	Common web pages starts with http://. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. All information exchanged are encrypted
ftp	File Transfer Protocol	For downloading or uploading files to a website. Useful for domain maintenance
file		A file on your computer

URL એન્કોડિંગ

ASCII કેરેક્ટર-સેટનો ઉપયોગ કરીને યુઆરએલ ફક્ત ઈન્ટરનેટ પર મોકલી શકાય છે.

URL માં ઘણીવાર ASCII સેટની બહારના અક્ષરો હોવાથી, URL ને માન્ય ASCII ફોર્મેટમાં રૂપાંતરિત કરવું પડશે.

URL એન્કોડિંગ બિન ASCII અક્ષરોને “%” સાથે બદલે છે અને ત્યારબાદ બે હેક્સાડેસિમલ અંકો.

URL માં જગ્યાઓ શામેલ હોઈ શકતી નથી. URL એન્કોડિંગ સામાન્ય રીતે + ચિહ્ન સાથે સ્પેસને બદલે છે.

XML માળખું

દસ્તાવેજના ભાગો, પ્રસ્તાવના સહિત XML માળખું અને એક સરળ XML ઉદાહરણ દસ્તાવેજ પ્રદાન કરે છે.

દસ્તાવેજના ભાગો

- પ્રોલોગ
- દસ્તાવેજ તત્વ (રુટ તત્વ)

પ્રસ્તાવના

પ્રસ્તાવના, HTML માં હેડરની સમકક્ષ, નીચેનાનો સમાવેશ કરી શકે છે:

- XML ઘોષણા (વૈકલ્પિક) જેમ કે:

- DTD અથવા એકનો સંદર્ભ (વૈકલ્પિક). બાહ્ય DTD ફાઈલનો ઉદાહરણ સંદર્ભ:

પ્રોસેસિંગ સૂચનાઓ - એક ઉદાહરણ પ્રોસેસિંગ સૂચના કે જે સ્ટાઈલ શીટ દ્વારા શૈલીને નિર્ધારિત કરવાનું કારણ બને છે:

```
<?xml-stylesheet type="text/css"
href="xmlstyle.css"?>
```

એક XML ઉદાહરણ

તેથી સંપૂર્ણ સારી રીતે રચાયેલ XML દસ્તાવેજ આના જેવો દેખાઈ શકે છે:

```
<?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>
```

જમીન તત્વ, ઉપર, મૂળ તત્વ છે.

પરિણામ

ઓક પાઈન મેપલ બ્લુગ્રાસ ફેસ્ક્યુ રાઈ

વેબ તત્વો

વેબ પૃષ્ઠ, માહિતી સમૂહ તરીકે, અસંખ્ય પ્રકારની માહિતી સમાવી શકે છે, જે અંતિમ વપરાશકર્તા દ્વારા જોવા, સાંભળવામાં અથવા ક્રિયાપ્રતિક્રિયા કરવા સક્ષમ છે.

વેબ હોસ્ટિંગ

વેબ હોસ્ટિંગ સેવા એ ઇન્ટરનેટ હોસ્ટિંગ સેવાનો એક પ્રકાર છે જે વ્યક્તિઓ અને સંસ્થાઓને તેમની વેબસાઈટને વર્લ્ડ વાઈડ વેબ દ્વારા ઍક્સેસિબલ બનાવવાની મંજૂરી આપે છે. વેબ હોસ્ટ એ એવી કંપનીઓ છે કે જે સર્વર (કમ્પ્યુટિંગ) માલિકીના અથવા ગ્રાહકો દ્વારા ઉપયોગ માટે ભાડે લીધેલ જગ્યા પૂરી પાડે છે, તેમજ ઇન્ટરનેટ કનેક્ટિવિટી પૂરી પાડે છે, ખાસ કરીને ડેટા સેન્ટરમાં. વેબ હોસ્ટ તેમના ડેટા સેન્ટરમાં સ્થિત અન્ય સર્વિસ માટે ડેટા સેન્ટર સ્પેસ અને ઇન્ટરનેટ સાથે કનેક્ટિવિટી પણ પ્રદાન કરી શકે છે, જેને કોલોકેશન કહેવાય છે.

CMS અને વેબ ઓથરિંગ ટૂલ્સનો પરિચય (Introduction to CMS and web authoring tools)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સામગ્રી મેનેજમેન્ટ સિસ્ટમનો અર્થ સમજાવો.
- કેટલાક લોકપ્રિય CMS ની યાદી બનાવો.
- કોમ્પોઝર વેબ ઓથરિંગ ટૂલના ઉપયોગ અને મુખ્ય લક્ષણો સમજાવો.

CMS ની વ્યાખ્યા:

CMS એટલે “કન્ટેન્ટ મેનેજમેન્ટ સિસ્ટમ.” CMS એ એક સોફ્ટવેર સાધન છે જે તમને સામગ્રી બનાવવા, સંપાદિત કરવા અને પ્રકાશિત કરવાની મંજૂરી આપે છે.

CMS નું વર્ણન:

CMS નો ધ્યેય વેબપેજ સામગ્રી બનાવવા અને સંશોધિત કરવા માટે સાહજિક વપરાશકર્તા ઇન્ટરફેસ પ્રદાન કરવાનો છે. દરેક CMS વેબ પબ્લિશિંગ ટૂલ પણ પ્રદાન કરે છે જે એક અથવા વધુ વપરાશકર્તાઓને વેબ પર લાઇવ અપડેટ્સ પ્રકાશિત કરવાની મંજૂરી આપે છે. સંપાદન ઘટકને કન્ટેન્ટ મેનેજમેન્ટ એપ્લિકેશન (CMA) કહેવામાં આવે છે, જ્યારે પ્રકાશન સાધનને સામગ્રી વિતરણ એપ્લિકેશન (CDA) કહેવામાં આવે છે. વેબ ડેવલપમેન્ટ પ્રક્રિયાને સુવ્યવસ્થિત કરવા માટે આ બે ઘટકોને CMSમાં એકસાથે સંકલિત કરવામાં આવ્યા છે.

વેબ પબ્લિશિંગના સંદર્ભમાં, સામગ્રી સરળ ટેક્સ્ટ, ફોટા, સંગીત, વિડિયો, દસ્તાવેજો અથવા તમે જે વિશે વિચારી શકો તે કંઈપણ હોઈ શકે છે. જ્યારે પ્રારંભિક CMS સોફ્ટવેરનો ઉપયોગ દસ્તાવેજો અને સ્થાનિક કમ્પ્યુટર ફાઇલોને સંચાલિત કરવા માટે કરવામાં આવતો હતો, મોટાભાગની CMS સિસ્ટમો હવે વેબ પર સામગ્રીનું સંચાલન કરવા માટે વિશિષ્ટ રીતે ડિઝાઇન કરવામાં આવી છે. CMS નો ઉપયોગ કરવાનો મુખ્ય ફાયદો એ છે કે તેને સંચાલિત કરવા માટે લગભગ કોઈ તકનીકી કૌશલ્ય અથવા જ્ઞાનની જરૂર નથી.

કન્ટેન્ટ મેનેજમેન્ટ સિસ્ટમ્સ ઇન્સ્ટોલ કરી શકાય તેવી એપ્લિકેશન્સ અને વેબ-આધારિત વપરાશકર્તા ઇન્ટરફેસ તરીકે ઉપલબ્ધ છે. વેબ ઇન્ટરફેસનો ઉપયોગ વેબસાઇટ અપડેટ કરવાની પ્રક્રિયાને સરળ બનાવે છે. વધુમાં, મોટાભાગના વેબ-આધારિત CMSes આપમેળે અપડેટ થાય છે, ખાતરી કરે છે કે બધા વપરાશકર્તાઓ પાસે તેમની સામગ્રીનું સંચાલન કરવા માટે નવીનતમ સાધનો છે.

CMS નો ઉપયોગ કરવાનો મુખ્ય ફાયદો એ છે કે તેને સંચાલિત કરવા માટે લગભગ કોઈ તકનીકી કૌશલ્ય અથવા જ્ઞાનની જરૂર નથી. CMS તમારી બધી સામગ્રીનું સંચાલન કરે છે, તેથી તમારે તે કરવાની જરૂર નથી.

આજે ઘણા વેબ-આધારિત CMS સાધનો ઉપલબ્ધ છે. નીચે આપેલા કેટલાક સૌથી લોકપ્રિય છે:

- વર્ડપ્રેસ - ટેમ્પલેટ આધારિત વેબસાઇટ અથવા બ્લોગ બનાવવા માટે રચાયેલ મફત વેબ સોફ્ટવેર
- બ્લોગર - Google નું બ્લોગિંગ ટૂલ ખાસ કરીને બ્લોગની જાળવણી માટે રચાયેલ છે
- જુમલા - એક લવચીક વેબ પબ્લિશિંગ ટૂલ જે કસ્ટમ ડેટાબેસેસ અને એક્સ્ટેન્શનને સપોર્ટ કરે છે

- Drupal - એક ઓપન સોર્સ પ્લેટફોર્મ જેનો ઉપયોગ સમુદાય-આધારિત સાઇટ્સ વિકસાવવા માટે થાય છે
- Weebly - સરળ વ્યક્તિગત અને વ્યવસાયિક વેબસાઇટ્સ બનાવવા માટે વેબ આધારિત પ્લેટફોર્મ
- Wix - અત્યંત વૈવિધ્યપૂર્ણ બનાવી શકાય તેવી વેબસાઇટ બનાવવા માટે વેબ પ્રકાશન સાધનોનો સંગ્રહ

WYSIWYG વેબ ઓથરિંગ ટૂલ્સનો પરિચય

WYSIWYG નો અર્થ છે “તમે જે જુઓ છો તે તમને મળે છે”. આવા સંપાદકોમાં તમે સીધા જ તમારા દસ્તાવેજોના સ્ત્રોત કોડને સંપાદિત કરશો નહીં, પરંતુ અંતિમ દસ્તાવેજમાં દેખાશે તે રીતે તેની રજૂઆત. તેથી કોડના બ્લોક્સ જાતે લખવાને બદલે (જેમ કે તમે તેને વર્ડ અથવા નોટપેડમાં કરો છો), તમે એડિટર વિંડોનો ઉપયોગ કરીને ડિઝાઇન ઘટકો સાથે ચાલાકી કરો છો. આનો અર્થ એ છે કે જ્યારે દસ્તાવેજ અથવા છબી બનાવવામાં આવી રહી હોય ત્યારે તમે અંતિમ પરિણામ જેવું જ કંઈક જુઓ છો.

એવા ઘણા ઉપયોગમાં સરળ WYSIWYG પ્રોગ્રામ્સ છે જેમાં એક જાટિલ અને સંપૂર્ણ કાર્યકારી વેબસાઇટ્સ બનાવવા માટે જરૂરી તમામ ટૂલ્સ હોય છે, નવા નિશાળીયા દ્વારા પણ. આ સાઇટ્સમાં HTML કોડ સાથે કામ કરવાના વિકલ્પો પણ છે, પછી તે ડિઝાઇન હોય કે સંપાદન. HTML સંપાદક સાથે વેબ સાઇટ બનાવવી સરળ છે, કારણ કે સોફ્ટવેર ડેવલપર્સ એવા સાધનો ઉમેરવાનું યાદુ રાખે છે જે તમને શૈલી સાથે અદ્યતન સુવિધાઓ વિકસાવવા દે છે. ઘણા WYSIWYG વેબ ઓથરિંગ ટૂલ્સ ડાયનેમિક HTML અથવા અન્ય ઘણી સુવિધાઓને એક ભવ્ય અને સુસંગત ડિઝાઇનવાળી સાઇટમાં એકીકૃત કરવા માટે અદ્યતન સુવિધાઓ પ્રદાન કરે છે.

કોમ્પોઝરનો પરિચય

KompoZer એ એક સંપૂર્ણ વેબ ઓથરિંગ સિસ્ટમ છે જે Microsoft FrontPage, Adobe DreamWeaver અને અન્ય હાઇ એન્ડ પ્રોગ્રામ્સમાં જોવા મળતી વેબ ફાઇલ મેનેજમેન્ટ અને ઉપયોગમાં સરળ WYSIWYG- web પૃષ્ઠ સંપાદન ક્ષમતાઓને જોડે છે. કોમ્પોઝર વડે તમે વેબ પેજ્સ બનાવી શકો છો અને એચટીએમએલની કોઈ ટેકનિકલ કુશળતા કે જ્ઞાન વગરની વેબસાઇટનું સંચાલન કરી શકો છો.

કોમ્પોઝરની ઉપયોગી વિશેષતા એ છે કે તમે તેનો ઉપયોગ તમારા વેબ પૃષ્ઠોને વેબ હોસ્ટિંગ સર્વર પર પ્રકાશિત કરવા માટે કરી શકો છો. તમારે ફક્ત તમારું FTP એકાઉન્ટ વપરાશકર્તાનામ અને પાસવર્ડ તેમજ સાઇટ URL જાણવાની જરૂર છે અને કોમ્પોઝર લોગ ઇન કરશે અને તમારા વેબ પૃષ્ઠોને સર્વર પર મૂકશે.

કોમ્પોઝરના ફાઇલ મેનૂમાં તેનો ઉપયોગ કરતી વખતે શક્ય તમામ મુખ્ય કામગીરી માટે મેનુ આઈટમ્સ શામેલ છે. કોમ્પોઝરના ટૂલ બાર યુઝર ઇન્ટરફેસમાં આનો સમાવેશ થાય છે:

- કમ્પોઝિશન ટૂલબાર
- ફોર્મેટ ટૂલબાર
- ટેબ બ્રાઉઝર ટૂલબાર
- સંપાદિત કરો મોડ ટૂલબાર
- સ્ટેટસ ટૂલબાર

રચના ટૂલબાર

કમ્પોઝિશન ટૂલબારમાં સૌથી વધુ ઉપયોગમાં લેવાતી મુખ્ય કામગીરી માટેના બટનોનો સમાવેશ થાય છે. નીચે ડિફોલ્ટ કમ્પોઝિશન ટૂલબારનો સ્નેપશોટ છે જે દરેક બટનની નીચે ટેક્સ્ટ સાથે સંબંધિત બટનના કાર્યને દર્શાવે છે. (ફિગ.1 નો સંદર્ભ લો)



ફોર્મેટ ટૂલબાર(ફિગ.2 નો સંદર્ભ લો)

ફોર્મેટ ટૂલબાર એ KompoZer સાથે વેબ પૃષ્ઠોને સંપાદિત કરતી વખતે ખૂબ જ ઉપયોગી સાધન છે. ફોર્મેટ ટૂલબાર વડે તમે ફકરાનું ફોર્મેટ લાગુ કરી શકો છો, ફોન્ટ પસંદ કરી શકો છો, ફોરગ્રાઉન્ડ અથવા બેકગ્રાઉન્ડ રંગ બદલી શકો છો, કદ વધારી અથવા ઘટાડી શકો છો અને ટેક્સ્ટના બ્લોક પર ભાર મૂકી શકો છો. તમે ક્રમાંકિત અથવા બિનક્રમાંકિત સૂચિઓ પણ બનાવી શકો છો અને ડાબે, જમણે અથવા મધ્યમાં ટેક્સ્ટના બ્લોકને યોગ્ય ઠેરવી શકો છો. કયું બટન શું કરે છે તે જાણવા માટે ફક્ત બટન પર તમારું માઉસ ફેરવો અને એક ટૂલ ટીપ દેખાશે જે બટનનું કાર્ય સૂચવે છે.



ટેબ બ્રાઉઝર ટૂલબાર(ફિગ.3 નો સંદર્ભ લો)

KompoZer તમને દરેક ખોલેલા દસ્તાવેજ માટે વિવિધ ટેબનો ઉપયોગ કરીને એકસાથે બહુવિધ વેબ દસ્તાવેજોને સંપાદિત કરવાની મંજૂરી આપે છે. બહુવિધ ટેબ્સ રાખવાથી તમારા ડેસ્કટોપને વધુ સ્વચ્છ દેખાવ મળે છે કારણ કે જ્યારે દરેક દસ્તાવેજ માટે બહુવિધ વિંડોઝનો ઉપયોગ કરવામાં આવે છે ત્યારે તે અવ્યવસ્થિત નથી. દ્રશ્ય સૂચક તરીકે “રેડ ફ્લોપી આઈકોન” આઈકોન એવા પૃષ્ઠો માટે દેખાય છે જે સંપાદિત કરવામાં આવ્યા છે પરંતુ હજુ સુધી સાચવવામાં આવ્યા નથી.



KompoZer તમને દરેક ખોલેલા દસ્તાવેજ માટે અલગ-અલગ ટેબનો ઉપયોગ કરીને એકસાથે બહુવિધ વેબ દસ્તાવેજોને સંપાદિત કરવાની મંજૂરી આપે છે. બહુવિધ ટેબ રાખવાથી તમારા ડેસ્કટોપને વધુ સ્વચ્છ દેખાવ મળે છે કારણ કે જ્યારે દરેક દસ્તાવેજ માટે બહુવિધ વિંડોઝનો ઉપયોગ કરવામાં આવે છે ત્યારે તે અવ્યવસ્થિત થતો નથી. દ્રશ્ય સૂચક તરીકે “લાલ ફ્લોપી આયકન” ચિહ્ન એવા પૃષ્ઠો માટે દેખાય છે જે સંપાદિત કરવામાં આવ્યા છે પરંતુ હજુ સુધી સાચવવામાં આવ્યા નથી.

સંપાદિત કરો મોડ ટૂલબાર (ફિગ 4)



એડિટ મોડ ટૂલબાર વ્યુઈંગ મોડ સૂચવે છે જે વર્તમાન દસ્તાવેજ માટે હાલમાં સક્રિય છે. ઉપલબ્ધ છેજોવાના મોડ્સ સામાન્ય દૃશ્ય, HTML ટેગ્સ દૃશ્ય, HTML સ્રોત દૃશ્ય અને પૂર્વાવલોકન મોડ છે. તમે માઉસ વડે અન્ય ત્રણમાંથી કોઈપણ પર ક્લિક કરીને સરળતાથી તમારો જોવાનો મોડ બદલી શકો છો.

સ્ટેટસ ટૂલબાર (ફિગ 5)



કોમ્પોઝરનો સ્ટેટસ બાર HTML ટેગ હાયરાર્કીના સંદર્ભમાં કર્સરની સ્થિતિ દર્શાવે છે. તમે સ્ટેટસ બારમાં કોઈપણ ચોક્કસ ટેગના ગુણધર્મોને ફક્ત રાઈટ ક્લિક કરીને અને ઇચ્છિત વિકલ્પ પસંદ કરીને સરળતાથી સંપાદિત/સોંપણી કરી શકો છો. સ્ટેટસ બારમાં ટેગ પર માઉસનું સરળ ડાબું ક્લિક તે ટેગથી ઘેરાયેલું લખાણ પસંદ કરે છે.

JavaScript ને સમજવું (Understanding JavaScript)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- પ્રોગ્રામિંગ અને સ્ક્રિપ્ટિંગ ભાષાઓને વ્યાખ્યાયિત કરો
- જાણો જાવાસ્ક્રિપ્ટ શું છે અને જાવા સ્ક્રિપ્ટનો ઇતિહાસ
- JavaScript કેવી રીતે ચલાવવું તે સમજાવો
- જાવાસ્ક્રિપ્ટ ચલાવવા માટે જરૂરી સાધનોની યાદી બનાવો
- નમૂના JavaScript પ્રોગ્રામ જુઓ
- JavaScript ના લક્ષણો જાણો
- JavaScript ના ફાયદા અને ગેરફાયદાનું વર્ણન કરો
- JavaScript આવૃત્તિઓ સમજાવો.

પ્રોગ્રામિંગ અને સ્ક્રિપ્ટિંગ ભાષાઓનો પરિચય

કમ્પ્યુટર પ્રોગ્રામિંગ એ સૂચનાઓ લખવાની પ્રક્રિયા છે જે કમ્પ્યુટર દ્વારા ચલાવવામાં આવે છે. સૂચનાઓ, જેને કોડ તરીકે પણ ઓળખવામાં આવે છે, તે પ્રોગ્રામિંગ ભાષામાં લખવામાં આવે છે જેને કમ્પ્યુટર સમજી શકે છે અને તેનો ઉપયોગ કાર્ય કરવા અથવા સમસ્યા હલ કરવા માટે કરી શકે છે.

સ્ક્રિપ્ટ અથવા સ્ક્રિપ્ટિંગ લેંગ્વેજ એ ફાઇલની અંદર આદેશોની શ્રેણી સાથેની કમ્પ્યુટર ભાષા છે જે કમ્પાઇલ કર્યા વિના ચલાવવામાં સક્ષમ છે. સર્વર-સાઇડ સ્ક્રિપ્ટિંગ ભાષાઓના સારા ઉદાહરણોમાં પર્લ, PHP અને પાયથોનનો સમાવેશ થાય છે. ક્લાયન્ટ સાઇડ સ્ક્રિપ્ટિંગ ભાષાનું શ્રેષ્ઠ ઉદાહરણ JavaScript છે.

સ્ક્રિપ્ટના ફાયદા

- ઓપન સોર્સ, જે જરૂરી હોય તો વપરાશકર્તાઓને સ્ક્રિપ્ટ જોવા અને સંપાદિત કરવાની મંજૂરી આપે છે.
- ફાઇલને કમ્પાઇલ કરવાની જરૂર નથી, પરંતુ જ્યારે જરૂરી હોય ત્યારે હોઈ શકે છે.
- શીખવા અને લખવામાં સરળ.
- વિવિધ ઓપરેટિંગ સિસ્ટમો વચ્ચે પોર્ટ કરવા માટે સરળ.
- વાસ્તવિક પ્રોગ્રામ કરતાં વધુ ઝડપથી વિકાસ થાય છે - કેટલીક વ્યક્તિઓ અને કંપનીઓ વાસ્તવિક પ્રોગ્રામ્સ માટે પ્રોટોટાઇપ તરીકે સ્ક્રિપ્ટો લખે છે.

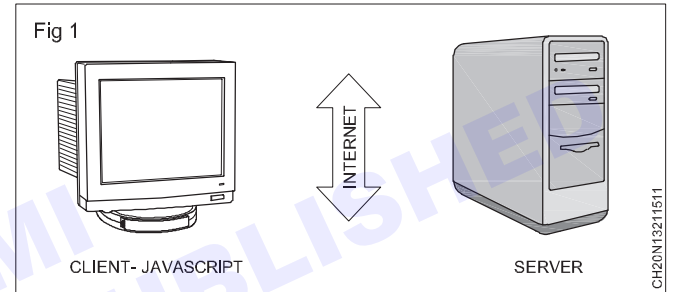
સ્ક્રિપ્ટના ગેરફાયદા

- ઓપન સોર્સ, અન્ય લોકોને સોત કોડ જોવાની મંજૂરી આપે છે, જે કેટલીક કંપનીઓ દ્વારા પ્રતિબંધિત હોઈ શકે છે.
- સ્ક્રિપ્ટ ચલાવી શકાય તે પહેલાં વપરાશકર્તાને દુભાષિયા અથવા અલગ પ્રોગ્રામ ઇન્સ્ટોલ કરવાની જરૂર છે.
- કેટલીક પરિસ્થિતિઓમાં, તેઓ સંકલિત પ્રોગ્રામ કરતાં ધીમા હોઈ શકે છે.

જાવા સ્ક્રિપ્ટ શું છે?

JavaScript એ ખૂબ જ શક્તિશાળી ક્લાયન્ટ સાઇડ સ્ક્રિપ્ટિંગ ભાષા છે. JavaScript નો ઉપયોગ મુખ્યત્વે વેબપેજ (ફિગ 1) સાથે વપરાશકર્તાની ક્રિયાપ્રતિક્રિયા વધારવા માટે થાય છે. બીજા શબ્દોમાં કહીએ તો, તમે

JavaScriptની મદદથી તમારા વેબપેજને વધુ જીવંત અને ઇન્ટરેક્ટિવ બનાવી શકો છો. જાવાસ્ક્રિપ્ટનો ઉપયોગ ગેમ ડેવલપમેન્ટ અને મોબાઇલ એપ્લિકેશન ડેવલપમેન્ટમાં પણ વ્યાપકપણે થઈ રહ્યો છે.



જાવાસ્ક્રિપ્ટ ઇતિહાસ

JavaScript 1995 માં બ્રેન્ડન ઈય દ્વારા વિકસાવવામાં આવી હતી, જે તે સમયના લોકપ્રિય બ્રાઉઝર નેટસ્કેપમાં દેખાય છે. આ ભાષાને શરૂઆતમાં લાઇવ સ્ક્રિપ્ટ કહેવામાં આવતું હતું અને બાદમાં તેનું નામ JavaScript રાખવામાં આવ્યું હતું. એવા ઘણા પ્રોગ્રામરો છે જેઓ વિચારે છે કે JavaScript અને Java એક જ છે. હકીકતમાં, JavaScript અને Java ખૂબ જ અસંબંધિત છે. જાવા એ ખૂબ જ જટિલ પ્રોગ્રામિંગ ભાષા છે જ્યારે JavaScript માત્ર એક સ્ક્રિપ્ટિંગ ભાષા છે. જાવા સ્ક્રિપ્ટનું વાક્યરચના મોટે ભાગે પ્રોગ્રામિંગ ભાષા C દ્વારા પ્રભાવિત છે.

JavaScript કેવી રીતે ચલાવવું?

સ્ક્રિપ્ટિંગ લેંગ્વેજ હોવાને કારણે જાવાસ્ક્રિપ્ટ પોતાની રીતે ચાલી શકતી નથી. વાસ્તવમાં, બ્રાઉઝર JavaScript કોડ ચલાવવા માટે જવાબદાર છે. જ્યારે વપરાશકર્તા JavaScript સાથે HTML પૃષ્ઠની વિનંતી કરે છે, ત્યારે સ્ક્રિપ્ટ બ્રાઉઝરને મોકલવામાં આવે છે અને તેને એક્ઝિક્યુટ કરવાનું બ્રાઉઝર પર છે. JavaScript નો મુખ્ય ફાયદો એ છે કે તમામ આધુનિક વેબ બ્રાઉઝર JavaScript ને સપોર્ટ કરે છે. તેથી, તમારે ચિંતા કરવાની જરૂર નથી કે તમારી સાઇટના મુલાકાતી ઇન્ટરનેટ એક્સપ્લોરર, ગૂગલ ક્રોમ, ફાયરફોક્સ અથવા અન્ય કોઈ બ્રાઉઝરનો ઉપયોગ કરે છે કે કેમ. JavaScript ને સપોર્ટ કરવામાં આવશે. ઉપરાંત, JavaScript Windows, Linux અથવા Mac સહિત કોઈપણ ઓપરેટિંગ સિસ્ટમ પર ચાલે છે.

જાવાસ્ક્રિપ્ટ ચલાવવા માટે તમારે જરૂરી સાધનો

શરૂ કરવા માટે, કોડ લખવા માટે ટેક્સ્ટ એડિટર અને વેબ પૃષ્ઠો પ્રદર્શિત કરવા માટે બ્રાઉઝર. નોટપેડ++, વિઝ્યુઅલ સ્ટુડિયો કોડ, સબલાઇમ ટેક્સ્ટ, એટમ અથવા અન્ય કોઈપણ ટેક્સ્ટ એડિટર સહિત પસંદગીનો

ટેક્સ્ટ એડિટર ઉપયોગ કરે છે. અને તે પણ, Google Chrome, Firefox, Microsoft Edge, Internet Explorer વગેરે સહિત કોઈપણ વેબ બ્રાઉઝરનો ઉપયોગ કરી શકો છો.

નમૂના JavaScript પ્રોગ્રામ

```
<html>

<head>

<title>My First JavaScript code!!!</title>

<script type="text/javascript">

alert("Welcome to JavaScript Program!");

</script>

</head>

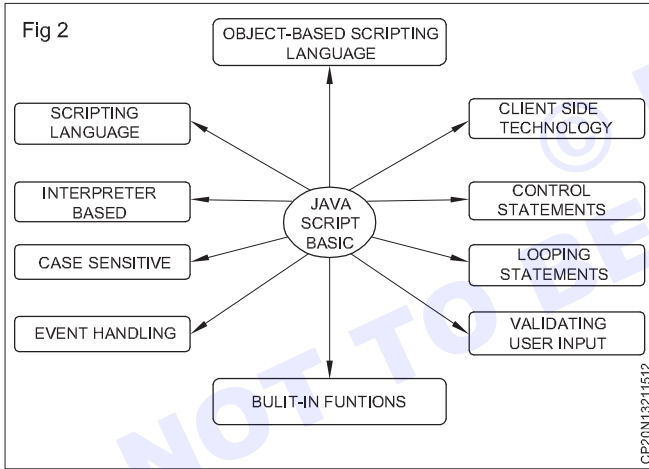
<body>

</body>

</html>
```

JavaScript ની વિશેષતાઓ

JavaScript એ ક્લાયન્ટ સાઈડ ટેકનોલોજી છે, તેનો ઉપયોગ મુખ્યત્વે ક્લાયન્ટ સાઈડ વેલિકેશન માટે થાય છે, પરંતુ તેમાં ઘણી બધી સુવિધાઓ છે જે ફિગ 2 માં દર્શાવેલ છે.



- JavaScript એ ઓબ્જેક્ટ-આધારિત સ્ક્રિપ્ટિંગ ભાષા છે.
- તે વપરાશકર્તાને બ્રાઉઝર પર વધુ નિયંત્રણ આપે છે.
- તે તારીખો અને સમય સંભાળે છે.
- તે વપરાશકર્તાના બ્રાઉઝર અને OSને શોધે છે,
- તે હળવા વજનનું છે.
- તે સ્ક્રિપ્ટિંગ ભાષા છે અને તે જાવા નથી.
- તે દુભાષિયા આધારિત સ્ક્રિપ્ટિંગ ભાષા છે.
- તે કેસ સંવેદનશીલ છે.
- તે ઓબ્જેક્ટ આધારિત ભાષા છે કારણ કે તે પૂર્વવ્યાખ્યાયિત વસ્તુઓ પ્રદાન કરે છે.

- JavaScript માં દરેક નિવેદન અર્ધવિરામ (;) સાથે સમાપ્ત થવું આવશ્યક છે.
- મોટાભાગના JavaScript કંટ્રોલ સ્ટેટમેન્ટ સિન્ટેક્સ C લેંગ્વેજમાં કંટ્રોલ સ્ટેટમેન્ટના સિન્ટેક્સ સમાન છે.
- JavaScript નો મહત્વનો ભાગ એ સ્ક્રિપ્ટમાં નવા કાર્યો બનાવવાની ક્ષમતા છે.

JavaScript ની ફાયદા

- ક્લાયન્ટ બાજુ પર અમલ: ઉદાહરણ તરીકે, સર્વરને વિનંતી મોકલતા પહેલા વપરાશકર્તા કોઈપણ વપરાશકર્તા ઇનપુટને માન્ય કરી શકે છે. આનાથી સર્વર પર ઓછો લોડ થાય છે.
- પ્રમાણમાં સરળ ભાષા: આ શીખવા માટે એકદમ સરળ છે અને વાક્યરચના જે અંગ્રેજીની નજીક છે.
- મુલાકાતીઓને ઇન્ટરફેસ પ્રતિસાદ: કોઈપણ સર્વર ક્રિયાપ્રતિક્રિયા વિના, ઇચ્છા પરિણામ મેળવવા માટે પૃષ્ઠ ફરીથી લોડ થવાની રાહ જોવી પડતી નથી.
- અંતિમ વપરાશકર્તા માટે ઝડપી: વપરાશકર્તાના કમ્પ્યુટર પર સ્ક્રિપ્ટ ચલાવવામાં આવે છે, કાર્ય પર આધાર રાખીને, પરિણામો લગભગ તરત જ પૂર્ણ થાય છે.
- ઇન્ટરએક્ટિવિટી વધી છે: એવા ઇન્ટરફેસ બનાવવા કે જે જ્યારે વપરાશકર્તા તેમના પર હોવર કરે અથવા કીબોર્ડનો ઉપયોગ કરીને તેમને સક્રિય કરે ત્યારે પ્રતિક્રિયા આપી શકે.
- રિચ ઇન્ટરફેસ: ઘટકોને ખેંચો અને છોડો અથવા સ્લાઈડર સાઈટ મુલાકાતીઓને સમૃદ્ધ ઇન્ટરફેસ આપી શકે છે.

JavaScript ના ગેરફાયદા

- સુરક્ષા સમસ્યાઓ: કોઈપણ JavaScript સ્ક્રિપ્ટેડ, જ્યારે ક્લાયન્ટ બાજુ પર વેબ પૃષ્ઠો પર તરત જ જોડવામાં આવે છે, તેનો ઉપયોગ વપરાશકર્તાની સિસ્ટમનું શોષણ કરવા માટે પણ થઈ શકે છે.
- તેમાં કોઈ મલ્ટિપ્રોસેસર અથવા મલ્ટિ થ્રેડિંગ ક્ષમતાઓ નથી.
- કોઈ સપોર્ટ ઉપલબ્ધ ન હોવાથી, જાવાસ્ક્રિપ્ટનો ઉપયોગ કોઈપણ નેટવર્કિંગ એપ્લિકેશન માટે કરી શકાતો નથી.
- JavaScript અમને ફાઈલો વાંચવા કે લખવાની મંજૂરી આપતું નથી.
- JavaScript રેન્ડર બદલાય છે: JavaScript અલગ-અલગ લેઆઉટ એન્જિન દ્વારા અલગ રીતે રેન્ડર કરવામાં આવી શકે છે. પરિણામે, આ ઇન્ટરફેસ અને કાર્યક્ષમતાના સંદર્ભમાં અસંગતતાનું કારણ બને છે.

JavaScript આવૃત્તિઓ

જાવાસ્ક્રિપ્ટની શોધ બ્રેન્ડન ઈય દ્વારા 1995માં કરવામાં આવી હતી અને તે 1997માં ECMA ધોરણ બની હતી. ECMA સ્ક્રિપ્ટ એ ભાષાનું સત્તાવાર નામ છે.

2015 થી ECMA સ્ક્રિપ્ટને વર્ષ દ્વારા નામ આપવામાં આવ્યું છે (ECMA સ્ક્રિપ્ટ 2015).

આવૃત્તિઓ	સત્તાવાર નામ	વર્ણન
1	ECMA Script 1 (1997)	પ્રથમ આવૃત્તિ.
2	ECMA Script 2 (1998)	માત્ર સંપાદકીય ફેરફારો.
3	ECMA Script 3 (1999)	નિયમિત અભિવ્યક્તિઓ ઉમેર્યા. ઉમેરાયેલ પ્રયાસ/કેચ.
4	ECMA Script 4	ક્યારેય છૂટી નથી.
5	ECMA Script 5 (2009)	“કડક મોડ” ઉમેર્યું. JSON સપોર્ટ ઉમેર્યો. String.trim() ઉમેર્યું. Array.isArray() ઉમેર્યું. ઉમેરાયેલ એરે પુનરાવર્તન પદ્ધતિઓ.
5.1	ECMA Script 5.1 (2011)	સંપાદકીય ફેરફારો.
6	ECMA Script 2015	લેટ અને કોન્સ્ટ ઉમેર્યું. મૂળભૂત પરિમાણ મૂલ્યો ઉમેર્યા. Array.find().Array.findIndex() ઉમેર્યું.
7	ECMA Script 2016	ઘાતાંકીય ઓપરેટર (**) ઉમેર્યું. ઉમેરાયેલ Array.prototype.includes.
8	ECMA Script 2017	સ્ટ્રેંગ પેડિંગ ઉમેર્યું. નવા ઓબ્જેક્ટ ગુણધર્મો ઉમેર્યા. Async ફંક્શન ઉમેર્યું. શેર કરેલી મેમરી ઉમેરાઈ
9	ECMA Script 2018	બાકીના/સ્પ્રેડ ગુણધર્મો ઉમેર્યા. અસુમેળ પુનરાવર્તન ઉમેર્યું. Promise.finally().Reg Exp માં ઉમેરાઓ ઉમેર્યા

વેબ સર્વર્સ અને બાહ્ય JavaScript ફાઈલોનો પરિચય (Introduction to Web servers and External JavaScript files)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

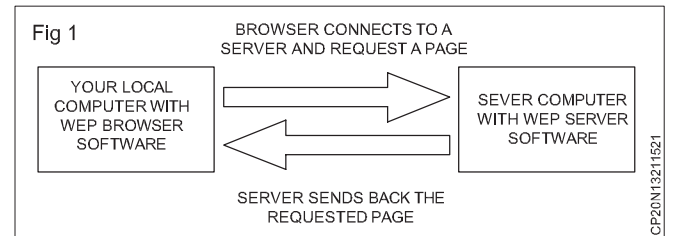
- વેબ સર્વર અને તેમની વિશેષતાઓ સમજાવો
- બાહ્ય JavaScript ફાઈલો સમજાવો.

વેબ સર્વર શું છે અને તે કેવી રીતે કાર્ય કરે છે?

વેબ સર્વર્સ કોઈપણ વેબ હોસ્ટિંગ માટે મુખ્ય છે. (ફિગ 1)

વેબ સર્વર એ એક પ્રોગ્રામ છે જે ફાઈલોને સેવા આપવા માટે HTTP નો ઉપયોગ કરે છે જે વપરાશકર્તાઓને તેમની વિનંતીઓના જવાબમાં વેબ પૃષ્ઠો બનાવે છે, જે તેમના કમ્પ્યુટર્સ HTTP કનેક્શન દ્વારા મોકલવામાં આવે છે. કોઈપણ સર્વર જે XML દસ્તાવેજને અન્ય ઉપકરણ પર પહોંચાડે છે તે વેબ સર્વર હોઈ શકે છે. વધુ સારી વ્યાખ્યા એ હોઈ શકે છે કે વેબ સર્વર એ ઈન્ટરનેટ સર્વર છે જે સામગ્રી અને સેવાઓ પહોંચાડવા માટે HTTP વિનંતીઓનો જવાબ આપે છે. હંમેશા વેબ સર્વર છે ઈન્ટરનેટ સાથે જોડાયેલ છે. દરેક વેબ સર્વર જે ઈન્ટરનેટ સાથે જોડાય છે તે અનન્ય સરનામું ધરાવતું હશે જેમાં 0 અને 255 ની વચ્ચેની ચાર સંખ્યાઓની શ્રેણી હોય છે. એક સમયગાળો (.) આ સંખ્યાઓને અલગ કરે છે. ઉપરાંત, વેબ સર્વર હોસ્ટિંગ પ્રદાતાઓને બહુવિધ ડોમેન્સ (વપરાશકર્તાઓ) ને સંચાલિત કરવા સક્ષમ બનાવે છે. એક જ સર્વર. વેબ હોસ્ટ એ એવી કંપની છે જે લોકોને તેમની

પોતાની સામગ્રી અને વેબપૃષ્ઠો સર્વ કરવા માટે સશક્ત કરવા સર્વર્સના કલસ્ટર પર જગ્યા ભાડે આપે છે.



વિવિધ પ્રકારના વેબ સર્વર

ઓપન માર્કેટમાં વિવિધ પ્રકારના વેબ સર્વર ઉપલબ્ધ છે. ચાલો સૌથી વધુ લોકપ્રિય વેબ સર્વરો વિશે ચર્ચા કરીએ. Apache, IIS, Nginx અને Lite Speed તેમાંથી થોડા છે.

અપાચે વેબ સર્વર

અપાચે સોફ્ટવેર ફાઉન્ડેશન દ્વારા વિકસિત વિશ્વના સૌથી લોકપ્રિય વેબ સર્વરમાંનું એક. Apache એક ઓપન સોર્સ સોફ્ટવેર છે જે Linux, Unix, Windows, FreeBSD, Mac OS X અને વધુ સહિત લગભગ તમામ ઓપરેટિંગ સિસ્ટમ્સને સપોર્ટ કરે છે. લગભગ 60% મશીનો અપાચે વેબ સર્વર પર ચાલે છે. (ફિગ 2)

Fig 2



અપાચે વેબ સર્વરનું કસ્ટમાઈઝેશન સરળ છે કારણ કે તે મોડ્યુલર માળખું ધરાવે છે. તે એક ઓપન સોર્સ પણ છે જેનો અર્થ છે કે જ્યારે જરૂર હોય ત્યારે સર્વરમાં પોતાના મોડ્યુલો ઉમેરી શકે છે અને જરૂરિયાતોને અનુરૂપ ફેરફારો કરી શકે છે. તે અન્ય વેબ સર્વર કરતાં વધુ સ્થિર છે અને વહીવટી સમસ્યાઓ ઉકેલવા માટે સરળ છે. તે બહુવિધ પ્લેટફોર્મ પર સફળતાપૂર્વક ઈન્સ્ટોલ કરી શકાય છે. તાજેતરના અપાચે રીલીઝ તેના પહેલાનાં વર્ઝનની સરખામણીમાં વધુ વિનંતીઓને હેન્ડલ કરવાની શક્યતા પૂરી પાડે છે.

IIS વેબ સર્વર

IIS એ Microsoft ઉત્પાદન છે. આ સર્વરમાં અપાચેની જેમ જ તમામ સુવિધાઓ છે. પરંતુ તે ઓપન સોર્સ નથી અને વધુ પર્સનલ મોડ્યુલ ઉમેરવું સહેલું નથી અને ફેરફાર થોડું મુશ્કેલ કામ બની જાય છે. (ફિગ 3)

Fig 3



માઈક્રોસોફ્ટે આ ઉત્પાદન વિકસાવ્યું છે અને તેઓ જાળવે છે, આમ તે તમામ વિન્ડોઝ ઓપરેટિંગ સિસ્ટમ પ્લેટફોર્મ સાથે કામ કરે છે. ઉપરાંત, જો તેને કોઈ સમસ્યા હોય તો તેઓ સારો ગ્રાહક સપોર્ટ પૂરો પાડે છે.

Nginx વેબ સર્વર

અન્ય મફત ઓપન સોર્સ વેબ સર્વર Nginx છે, તેમાં IMAP/POP3 પ્રોક્સી સર્વર શામેલ છે. Nginx તેના ઉચ્ચ પ્રદર્શન, સ્થિરતા, સરળ રૂપરેખાંકન અને ઓછા સંસાધન વપરાશ માટે જાણીતું છે. (ફિગ 4)

આ વેબ સર્વર વિનંતીઓને હેન્ડલ કરવા માટે થ્રેડોનો ઉપયોગ કરતું નથી તેના બદલે વધુ સ્કેલેબલ ઇવેન્ટ સંચાલિત આર્કિટેક્ચર જે લોડ હેન્ડલ મેમરીની નાની અને અનુમાનિત માત્રાનો ઉપયોગ કરે છે. તે તાજેતરના

Fig 4



સમયમાં લોકપ્રિય થઈ રહ્યું છે અને તે વિશ્વભરના તમામ ડોમેન્સમાંથી લગભગ 7.5% હોસ્ટ કરી રહ્યું છે. મોટાભાગની વેબ હોસ્ટિંગ કંપનીઓ તાજેતરના સમયમાં આનો ઉપયોગ કરી રહી છે.

બાહ્ય JavaScript ફાઈલો : HTML કોડમાં JavaScript લખવાથી કેટલીકવાર મૂંઝવણ ઊભી થાય છે અને HTML ફાઈલો બદલવાથી JavaScript ફાઈલોને પણ અસર થઈ શકે છે. તેથી HTML અને JavaScript ફાઈલોને અલગ કરવા માટે વધુ સારું છે જેથી, એક ફાઈલમાં ફેરફાર અન્ય ફાઈલોને અસર ન કરે.

બાહ્ય JavaScript ફાઈલોને નીચે પ્રમાણે અલગથી લખવી જોઈએ:-

File myjs.js Contents:

```
function popup() {  
    alert("Hello World");  
}
```

હવે આપણે ફાઈલને HTML ફાઈલમાં નીચે પ્રમાણે આયાત કરી શકીએ છીએ:-

બાહ્ય ફાઈલને આયાત કરવી પ્રમાણમાં પીડારહિત છે. પ્રથમ, તમે જે ફાઈલ આયાત કરી રહ્યા છો તે માન્ય JavaScript અને માત્ર JavaScript જ હોવી જોઈએ. બીજું, ફાઈલમાં ફાઈલ એક્સટેન્શન ".js" હોવું આવશ્યક છે. છેલ્લે, તમારે ફાઈલનું સ્થાન જાણવું આવશ્યક છે.

ચાલો ધારીએ કે આપણી પાસે "myjs.js" ફાઈલ છે જેમાં એક લાઈન હેલો વર્લ્ડ એલર્ટ ફંક્શન છે. ઉપરાંત, ચાલો ધારીએ કે ફાઈલ એ HTML ફાઈલ જેવી જ ડિરેક્ટરી છે જેને આપણે કોડ અપ કરવા જઈ રહ્યા છીએ. ફાઈલને આયાત કરવા માટે તમે તમારા HTML દસ્તાવેજમાં નીચે મુજબ કરશો.

JavaScript Code:

```
<html>  
  
<head>  
  
<script src="myjs.js">  
  
</script>  
  
</head>  
  
<body>  
  
<input type="button" onclick="popup()" value="Click Me!">  
  
</body>  
  
</html>
```

હવે આ HTML ફાઈલ myjs.js ફાઈલને આયાત કરે છે અને પરિણામે તે HTML બટન એલિમેન્ટમાંથી પોપઅપ() ફંક્શનને એક્સેસ કરી શકે છે

JavaScript વેરીએબલ અને ડેટા પ્રકારોનો ઉપયોગ કરવો (Using JavaScript Variable and data types)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- JavaScript માં ચલો સમજાવો
- JavaScript માં વિવિધ ડેટા પ્રકારો સમજાવો.

ચલો

JavaScript ચલો એ ડેટા વેલ્યુ સ્ટોર કરવા માટેના કન્ટેનર છે.

ઉદાહરણ તરીકે 1, a, b, અને c, ચલ છે:

ઉદાહરણ 1

```
var a = 12;

var b = 10;

var c = a + b;
```

ઉદાહરણ 1 થી, આપણે તે સમજી શકીએ છીએ

- a stores the value 12
- b stores the value 10
- c stores the value 22

ઉદાહરણ તરીકે 2, માર્ક1, માર્ક2 અને કુલ, ચલ છે:

ઉદાહરણ 2

```
var mark1 = 85;

var mark2 = 66;

var total = mark1 + mark2;
```

પ્રોગ્રામિંગમાં, બીજગણિતની જેમ, આપણે મૂલ્યો રાખવા માટે માર્ક1 અને માર્ક2 વેરીએબલનો ઉપયોગ કરીએ છીએ અને કુલ = માર્ક1 + માર્ક2 જેવા સમીકરણોમાં ચલોનો ઉપયોગ કરીએ છીએ. ઉપરના ઉદાહરણમાંથી, અને કુલ 151 ની ગણતરી કરો.

જાવાસ્ક્રિપ્ટ આઈડેન્ટિફાયર

બધા JavaScript ચલોને અનન્ય નામોથી ઓળખવા જોઈએ. આ અનન્ય નામોને ઓળખકર્તા કહેવામાં આવે છે. ઓળખકર્તા ટૂંકા નામો જેવા કે a અને b અથવા વધુ વર્ણનાત્મક નામો જેમ કે માર્ક1, માર્ક2, કુલ, ઉંમર, રકમ, કુલ વોલ્યુમ હોઈ શકે છે.

ચલો માટે નામો બનાવવા માટેના સામાન્ય નિયમો છે:

- નામોમાં અક્ષરો, અંકો, અન્ડરસ્કોર અને ડોલર ચિહ્નો હોઈ શકે છે.
- નામ એક અક્ષરથી શરૂ થવું જોઈએ
- નામો \$ અને _ થી પણ શરૂ થઈ શકે છે
- નામો કેસ સેન્સિટિવ છે (a અને A અલગ અલગ ચલ છે)
- JavaScript કીવર્ડ્સ જેવા આરક્ષિત શબ્દોનો ઉપયોગ નામ તરીકે કરી શકાતો નથી

નોંધ: JavaScript ઓળખકર્તાઓ કેસ સંવેદનશીલ હોય છે.

સૌપણી ઓપરેટર

JavaScript માં, સમાન ચિહ્ન (=) એ “એસાઈનમેન્ટ” ઓપરેટર છે અથવા “ઈક્વલ ટુ” ઓપરેટર નથી.

```
x = x + 10;
```

તે x + 10 થી x ની કિંમત અસાઈન કરે છે. તે x + 10 ની કિંમતની ગણતરી કરે છે અને પરિણામને x માં મૂકે છે. x નું મૂલ્ય 10 વડે વધ્યું છે.

JavaScript ડેટા પ્રકારો

JavaScript ચલો 100 જેવા નંબરો અને “સંતોષ કુમાર” જેવા ટેક્સ્ટ મૂલ્યો ધરાવી શકે છે.

પ્રોગ્રામિંગમાં, ટેક્સ્ટ મૂલ્યોને ટેક્સ્ટ સ્ટ્રિંગ્સ કહેવામાં આવે છે. જાવા સ્ક્રિપ્ટ ઘણા પ્રકારના ડેટાને હેન્ડલ કરી શકે છે, પરંતુ હમણાં માટે, ફક્ત સંખ્યાઓ અને શબ્દમાળાઓનો વિચાર કરો. સ્ટ્રિંગ્સ ડબલ અથવા સિંગલ અવતરણની અંદર લખવામાં આવે છે. નંબરો અવતરણ વિના લખવામાં આવે છે. જો તમે અવતરણમાં સંખ્યા મૂકો છો, તો તેને ટેક્સ્ટ સ્ટ્રિંગ તરીકે ગણવામાં આવશે.

ઉદાહરણ 3

```
var pi = 3.14;

var person = "santhoshkumar";

var city = "coimbatore";
```

JavaScript વેરીએબલ્સની ઘોષણા

JavaScript માં વેરીએબલ બનાવવાને વેરીએબલ જાહેર કરવું કહેવાય છે. JavaScript ચલ var કીવર્ડ સાથે જાહેર કરવામાં આવે છે.

```
var traineeName;
```

ઘોષણા પછી, ચલનું કોઈ મૂલ્ય નથી. તકનીકી રીતે તે અવ્યાખ્યાયિત મૂલ્ય ધરાવે છે. ચલને મૂલ્ય આપવા માટે, સમાન ચિહ્નોનો ઉપયોગ કરો.

```
traineeName = "Santhosh Kumar";
```

જ્યારે તમે તેને જાહેર કરો ત્યારે તમે ચલને મૂલ્ય પણ અસાઈન કરી શકો છો.

```
var traineeName = "Santhosh Kumar";
```

નીચેના ઉદાહરણમાં, અમે TraineeName નામનું ચલ બનાવીએ છીએ અને તેને “સંતોષ કુમાર” મૂલ્ય અસાઈન કરીએ છીએ.

પછી આપણે id="demo" સાથે HTML ફકરાની અંદરની કિંમત “આઉટપુટ” કરીએ છીએ:

```
<p id="demo"></p>
```

```
<script>
```

```
var traineeName = "santhoshkumar";
```

```
document.getElementById("demo").innerHTML
```

```
HTML = traineeName;
```

```
</script>
```

નોંધ: સ્ક્રિપ્ટની શરૂઆતમાં તમામ ચલોને જાહેર કરવા માટે તે સારી પ્રોગ્રામિંગ પ્રેક્ટિસ છે.

તમે એક સ્ટેટમેન્ટમાં ઘણા વેરિએબલ્સ જાહેર કરી શકો છો. વિધાન var થી શરૂ કરો અને ચલોને અલ્પવિરામ દ્વારા અલગ કરો.

ઉદાહરણ 4

```
var traineeName = "santhoshkumar", city =  
"coimbatore", total = "151";
```

અવ્યાખ્યાયિત મૂલ્ય

કોમ્પ્યુટર પ્રોગ્રામ્સમાં, ચલોને મોટાભાગે મૂલ્ય વગર જાહેર કરવામાં આવે છે. મૂલ્ય કંઈક એવું હોઈ શકે છે જેની ગણતરી કરવાની હોય, અથવા કંઈક કે જે પછીથી પ્રદાન કરવામાં આવશે, જેમ કે વપરાશકર્તા ઇનપુટ.

મૂલ્ય વિના જાહેર કરાયેલ ચલનું મૂલ્ય અવ્યાખ્યાયિત હશે.

આ સ્ટેટમેન્ટના અમલ પછી વેરિએબલ TraineeName ની કિંમત અવ્યાખ્યાયિત હશે.

ઉદાહરણ 5

```
var traineeName;
```

JavaScript ચલોને ફરીથી જાહેર કરી રહ્યાં છીએ

જો તમે JavaScript વેરીએબલને ફરીથી જાહેર કરો છો, તો તે તેનું મૂલ્ય ગુમાવશે નહીં. આ વિધાનોના અમલ પછી પણ વેરિએબલ TraineeName પાસે "santhoshkumar" મૂલ્ય રહેશે.

ઉદાહરણ 6

```
var traineeName = "santhoshkumar";
```

```
var traineeName;
```

જાવાસ્ક્રિપ્ટ અંકગણિત

= અને + જેવા ઓપરેટરોનો ઉપયોગ કરીને JavaScript ચલો સાથે અંકગણિત કરો

ઉદાહરણ 7

```
var x = 8 + 2 + 5;
```

હવે x ની કિંમત 15 છે.

તમે શબ્દમાળાઓ પણ ઉમેરી શકો છો, પરંતુ શબ્દમાળાઓ સંકલિત કરવામાં આવશે:

ઉદાહરણ 8

```
var x = "Dharani" + "" + "Shree"
```

હવે x ની કિંમત ધરણી શ્રી છે

નીચેના ઉદાહરણનું પરિણામ 725 આપે છે.

ઉદાહરણ 9

```
var x = "7" + 2 + 5;
```

નોંધ: જો તમે અવતરણમાં સંખ્યા મૂકો છો, તો બાકીની સંખ્યાઓ શબ્દમાળાઓ તરીકે ગણવામાં આવશે, અને સંકલિત કરવામાં આવશે.

નીચેના ઉદાહરણનું પરિણામ 75 આપે છે.

ઉદાહરણ 10

```
var x = 3 + 4 + "5";
```

ડેટા પ્રકારો

પ્રોગ્રામિંગમાં, ડેટા પ્રકારો એક મહત્વપૂર્ણ ખ્યાલ છે. ચલો પર કામ કરવા માટે સક્ષમ થવા માટે, ડેટા પ્રકાર વિશે જાણવું મહત્વપૂર્ણ છે.

JavaScript ચલો સંખ્યાઓ, શબ્દમાળાઓ, ઓબ્જેક્ટ્સ અને વધુ જેવા ઘણા ડેટા પ્રકારોને પકડી શકે છે.

ઉદાહરણ 11

```
var side = 10; // Number
```

```
var firstName = "Rithika"; // String
```

```
var x = {firstName: "Harini", lastName: "Kumar"};
```

```
// Object
```

ડેટા પ્રકારો વિના, કમ્પ્યુટર તેને સુરક્ષિત રીતે હલ કરી શકતું નથી.

ઉદાહરણ 12

```
var a = 10 + "Apple";
```

JavaScript ઉપરના ઉદાહરણને આ પ્રમાણે ગણાશે,

```
var a = "10" + "Apple";
```

The output is 10 Apple

નોંધ: નંબર અને સ્ટ્રિંગ ઉમેરતી વખતે, JavaScript નંબરને સ્ટ્રિંગ તરીકે ગણાશે.

JavaScript ડાબેથી જમણે અભિવ્યક્તિઓનું મૂલ્યાંકન કરે છે. વિવિધ ક્રમ વિવિધ પરિણામો પેદા કરી શકે છે.

ઉદાહરણ 13

```
var y = 20 + 5 + "Apple";
```

The result is 25 Apple

ઉદાહરણ 14

```
var y = "Apple" + 20 + 5;
```

The result is Apple205.

નોંધ: પ્રથમ ઉદાહરણમાં, JavaScript 20 અને 5 ને સંખ્યાઓ તરીકે ગણે છે, જ્યાં સુધી તે "Apple" સુધી ન પહોંચે. બીજા ઉદાહરણમાં, પ્રથમ ઓપરેન્ડ એક સ્ટ્રિંગ હોવાથી, તમામ ઓપરેન્ડને સ્ટ્રિંગ તરીકે ગણવામાં આવે છે.

ડાયનેમિક ડેટા પ્રકારો

JavaScript ગતિશીલ પ્રકારો ધરાવે છે. આનો અર્થ એ છે કે સમાન ચલનો ઉપયોગ વિવિધ ડેટા પ્રકારો રાખવા માટે થઈ શકે છે:

ઉદાહરણ 15

```
var z;           // Now z is undefined
z = 10;          // Now z is a Number
z = "Sakthi";    // Now z is a String
```

જાવાસ્ક્રિપ્ટ સ્ટ્રિંગ્સ

શબ્દમાળા અથવા ટેક્સ્ટ સ્ટ્રિંગ એ “હરિણી કુમાર” જેવા પાત્રોની શ્રેણી છે. શબ્દમાળાઓ અવતરણ સાથે લખવામાં આવે છે. તમે સિંગલ અથવા ડબલ અવતરણનો ઉપયોગ કરી શકો છો.

ઉદાહરણ 16

```
var bikeName = "Yamaha R15";    // Using double quotes
var bikeName = 'Yamaha R15';    // Using single quotes

તમે સ્ટ્રિંગની અંદરના અવતરણોનો ઉપયોગ કરી શકો છો, જ્યાં સુધી તે સ્ટ્રિંગની આસપાસના અવતરણો સાથે મેળ ખાતા નથી:
```

ઉદાહરણ 17

```
var answer = "It's OK";          // Single quote inside double quotes
var answer = 'Patel is called    // Double quotes inside "Iron Man"; single quotes
```

જાવાસ્ક્રિપ્ટ નંબર્સ

JavaScript માં ફક્ત એક જ પ્રકારનો નંબર છે. સંખ્યાઓ દશાંશ સાથે અથવા વગર લખી શકાય છે.

ઉદાહરણ 18

```
var num1 = 87.0;    // Written with decimals
var num2 = 87;      // Written without decimals
```

વધારાની મોટી અથવા વધારાની નાની સંખ્યાઓ વૈજ્ઞાનિક (ઘાતાંકીય) સંકેત સાથે લખી શકાય છે:

ઉદાહરણ 19

```
var exp1 = 232e5;    // result is 232000000
var z = 123e-5;      // result is 0.00232
```

ઉદાહરણ 20

```
var p = 3;
var q = 3;
var r = 5;
(p == q)    // Returns true
(p == r)    // Returns false
```

નોંધ: બુલિયનનો ઉપયોગ ઘણીવાર શરતી પરીક્ષણમાં થાય છે.

JavaScript એરે

JavaScript એરે ચોરસ કૌંસ સાથે લખવામાં આવે છે. એરે વસ્તુઓ અલ્પવિરામ દ્વારા અલગ કરવામાં આવે છે. નીચેનો કોડ ત્રણ વસ્તુઓ (બાઈકના નામો) ધરાવતી બાઈક તરીકે ઓળખાતી એરેની ઘોષણા કરે છે (બનાવે છે):

ઉદાહરણ 21

```
var bikes = ["Yamaha", "TVS", "Royal Enfield"];
```

નોંધ: એરે ઇન્ડેક્સ શૂન્ય-આધારિત છે, જેનો અર્થ છે કે પ્રથમ આઈટમ [0] છે, બીજી [1] છે, અને તેથી વધુ.

JavaScript ઓબ્જેક્ટ્સ

JavaScript ઓબ્જેક્ટ સર્પાકાર કૌંસ સાથે લખવામાં આવે છે. ઓબ્જેક્ટ પ્રોપર્ટીઝ નામ:મૂલ્ય જોડી, અલ્પવિરામ દ્વારા અલગ કરીને લખવામાં આવે છે.

ઉદાહરણ 22

```
var personName = {firstName: "Harini", lastName:
                  "Kumar", age: 13, height:
                  "155 cms"};
```

ઉપરના 22 ના ઉદાહરણમાં ઓબ્જેક્ટ (વ્યક્તિનું નામ) 4 ગુણધર્મો ધરાવે છે: પ્રથમ નામ, છેલ્લું નામ, ઉંમર અને ઊંચાઈ.

ઓપરેટરનો પ્રકાર

JavaScript typeof operator નો ઉપયોગ JavaScript ચલનો પ્રકાર શોધવા માટે થાય છે.

પ્રકારનો ઓપરેટર ચલ અથવા અભિવ્યક્તિનો પ્રકાર પરત કરે છે.

ઉદાહરણ 23

```
typeof ""           // Returns "string"
typeof "Rithika"    // Returns "string"
typeof "Harini Kumar" // Returns "string"
typeof 0             // Returns "number"
typeof 81            // Returns "number"
typeof 8.14          // Returns "number"
typeof (3+2)         // Returns "number"
```

અવ્યાખ્યાયિત

JavaScript માં, મૂલ્ય વિનાના ચલ, મૂલ્ય અવ્યાખ્યાયિત છે. પ્રકાર પણ અવ્યાખ્યાયિત છે

ઉદાહરણ 24

```
var bike;           // Value is undefined, type is undefined
```

નોંધ: કોઈપણ ચલ ખાલી કરી શકાય છે, મૂલ્યને અવ્યાખ્યાયિત પર સેટ કરીને. પ્રકાર પણ અવ્યાખ્યાયિત હશે.

ખાલી મૂલ્યો

ખાલી મૂલ્યને અવ્યાખ્યાયિત સાથે કોઈ લેવાદેવા નથી. ખાલી સ્ટ્રિંગમાં કાનૂની મૂલ્ય અને પ્રકાર બંને હોય છે.

ઉદાહરણ 25

```
var bike = ""; // The value is "", the type of
                is "string"
```

શૂન્ય

જાવાસ્ક્રિપ્ટમાં નલ “કંઈ નથી” છે. તે એવું માનવામાં આવે છે જે અસ્તિત્વમાં નથી. JavaScript માં, null નો ડેટા પ્રકાર એક ઓબ્જેક્ટ છે. તમે ઓબ્જેક્ટને નલ પર સેટ કરીને ખાલી કરી શકો છો.

ઉદાહરણ 26

```
var personName = {firstName:"Harini",last Name:
                  "Kumar",age:13,height:"155
                  cms"};

personName = null; //Now value in null, but
                  type is still an object
```

તમે ઓબ્જેક્ટને અવ્યાખ્યાયિત પર સેટ કરીને પણ ખાલી કરી શકો છો:

ઉદાહરણ 27

```
var personName = {firstName:"Harini",lastName:
                  "Kumar",age:13,height:"155
                  cms"};

personName = undefined; // Now both value
                        and type is undefined.
```

અવ્યાખ્યાયિત અને નલ વચ્ચેનો તફાવત

અવ્યાખ્યાયિત અને નલ મૂલ્યમાં સમાન છે પરંતુ પ્રકારમાં અલગ છે.

ઉદાહરણ 28

```
typeof undefined // undefined
typeof null       // object
null === undefined // false
null == undefined // true
```

આદિમ ડેટા

આદિમ ડેટા મૂલ્ય એ કોઈ વધારાના ગુણધર્મો અને પદ્ધતિઓ વિનાનું એક સરળ ડેટા મૂલ્ય છે. પ્રકાર ઓપરેટર આ આદિમ પ્રકારોમાંથી એક પરત કરી શકે છે.

જાવાસ્ક્રિપ્ટ કોન્સ્ટન્ટ્સ અને ઓપરેટર્સનો ઉપયોગ કરવો (Using JavaScript Constants and Operators)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- JavaScript માં સ્થિરાંકો સમજાવો
- JavaScript માં ઓપરેટરોને સમજાવો.

સ્થિર

સ્થિરાંકો એક વિશિષ્ટ પ્રકારનું ચલ છે, એક મૂલ્ય સંગ્રહિત કરે છે જે પ્રોગ્રામ દરમિયાન ક્યારેય બદલાતું નથી.

- તાર
- સંખ્યા
- બુલિયન
- અવ્યાખ્યાયિત

ઉદાહરણ 29

```
typeof "Rajesh" // Returns "string"
typeof 1.44      // Returns "number"
typeof true      // Returns "boolean"
typeof false     // Returns "boolean"
typeof a         // if a has no value, it returns "un
                  defined"
```

જટિલ ડેટા

પ્રકારનો ઓપરેટર બે જટિલ પ્રકારોમાંથી એક પરત કરી શકે છે:

- કાર્ય
- પદાર્થ

ઓપરેટરનો પ્રકાર ઓબ્જેક્ટ, એરે અને નલ બંને માટે ઓબ્જેક્ટ પરત કરે છે. તે કાર્યો માટે ઓબ્જેક્ટ પરત કરતું નથી.

ઉદાહરણ 30

```
typeof {name:'Karthik',age:27} // Returns
                                "object"
typeof [10,20,30,40,50]        // Returns
                                "object"(not
                                "array", see
                                note below)
typeof null                    // Returns
                                "object"
typeof function sampleFunc() {} // Returns
                                "function"
```

નોંધ: typeof ઓપરેટર એરે માટે “ઓબ્જેક્ટ” પરત કરે છે કારણ કે JavaScript માં એરે ઓબ્જેક્ટ છે.

ઉપરોક્ત વાક્યરચનામાં, “const” એ વિશિષ્ટ કીવર્ડ છે, જે સ્થિરતાને વ્યાખ્યાયિત કરવા માટે આરક્ષિત છે. જેમ તમે જોઈ શકો છો, આ વાક્યરચના એક ચલ ઘોષણા જેવું લાગે છે પરંતુ var કીવર્ડ સાથે “const” સાથે બદલાયેલ છે. મોટાભાગના પ્રોગ્રામરો ચલથી અલગ પાડવા માટે સ્થિરાંકોના નામ માટે તમામ કેપ્સનો ઉપયોગ કરે છે.

ઉદાહરણ 1

```
const FRIEND      = 'Shanthi';
const BROTHER_AGE = 46;
```

નોંધ: કીવર્ડ કોન્સ્ટ થોડો ભ્રામક છે. તે સ્થિર મૂલ્યને વ્યાખ્યાયિત કરતું નથી. તે મૂલ્યના સતત સંદર્ભને વ્યાખ્યાયિત કરે છે. આના કારણે, આપણે સતત આદિમ મૂલ્યોને બદલી શકતા નથી, પરંતુ આપણે સતત પદાર્થોના ગુણધર્મો બદલી શકીએ છીએ.

આદિમ મૂલ્યો

જો આપણે સ્થિર મૂલ્યને આદિમ મૂલ્ય અસાઈન કરીએ છીએ, તો આપણે આદિમ મૂલ્ય બદલી શકતા નથી.

ઉદાહરણ 2

```
const PI = 3.141592653589793;
PI = 3.14;           // This will give an error
PI = PI + 10;        // This will also give an error
```

સતત પદાર્થો બદલી શકે છે

સ્થિર પદાર્થના ગુણધર્મો બદલો.

ઉદાહરણ 3

```
// You can create a const object:
const bike = {type: "Yamaha", model: "R15", color:
    "blue"};
// You can change a property:
bike.color = "grey";
// You can add a property:
bike.owner = "Sree";
```

પરંતુ તમે સતત ઓબ્જેક્ટને ફરીથી સોંપી શકતા નથી.

ઉદાહરણ 4

```
const bike = {type: "Yamaha", model: "R15", color: "blue"};
bike = {type: "Tvs", model: "Star city", color: "black"};
// ERROR
```

સતત એરે બદલી શકે છે

તમે સતત એરેના તત્વો બદલી શકો છો.

ઉદાહરણ 5

```
// You can create a constant array:
constant bikes = ["TVS", "Yamaha", "Royal
    Enfield"];
```

```
// You can change an element;
```

```
bikes[0] = "suzuki";
```

```
bikes.push("Bajaj");//you can add an element
```

પરંતુ તમે સતત એરેને ફરીથી સોંપી શકતા નથી:

ઉદાહરણ 6

```
const bikes = ["TVS", "Yamaha", "Royal Enfield"];
bikes = ["TVS", "Yamaha", "Bajaj"]; //ERROR
```

ઓપરેટરો

JavaScript માં આઠ પ્રકારના ઓપરેટરો છે.

- એડિટિવ ઓપરેટર્સ
- ગુણાકાર ઓપરેટર્સ
- બીટવાઈઝ ઓપરેટર
- સમાનતા ઓપરેટર
- રિલેશનલ ઓપરેટર
- યુનરી ઓપરેટર્સ
- ટર્નરી ઓપરેટર
- સોંપણી ઓપરેટરો

એડિટિવ ઓપરેટર્સ : એડિટિવ ઓપરેટર્સ શબ્દમાં સરવાળો (+) અને બાદબાકી(-) બંનેનો સમાવેશ થાય છે કારણ કે બાદબાકી પણ નકારાત્મક સંખ્યા સાથેનો સરવાળો છે.

ઉદાહરણ 7

```
32+67;           // this is 99
d+e;             // Adds d with e
3-7;             // return -4
```

કેટલીકવાર JavaScript ઉમેરાથી અણધાર્યા પરિણામો આવી શકે છે.

ઉદાહરણ 8

```
var num1="Runs";
var num2=784;
var res=num1+num2;// result is Runs784 as java concatstwo
strings.
```

ગુણાકાર ઓપરેટર્સ : એડિટિવ ઓપરેટરની જેમ, ગુણાકાર(*), ભાગાકાર(/) અને મોડ્યુલો(%) ગુણાકાર ઓપરેટર્સ છે. મોડ્યુલો ઓપરેટર ડિવિઝનનો બાકીનો ભાગ પરત કરે છે.

ઉદાહરણ 9

```
javascript:alert(4%3);// returns 1
```

ઉમેરણ અને ગુણાકાર ઓપરેટરને એકસાથે અંકગણિત ઓપરેટર કહી શકાય.

બીટવાઈઝ ઓપરેટર:

JavaScript 32 બિટ્સ બિટવાઈઝ ઓપરેટરનો ઉપયોગ કરે છે. JavaScript નંબરોને 64 બિટ્સ ફ્લોટિંગ પોઈન્ટ નંબર્સ તરીકે સ્ટોર કરે છે, પરંતુ તમામ બિટવાઈઝ ઓપરેશન્સ 32 બિટ્સ બાઈનરી નંબર્સ પર કરવામાં

આવે છે. બિટવાઈઝ ઓપરેશન કરવામાં આવે તે પહેલાં, JavaScript નંબરોને 32 બિટ્સ સાઈન કરેલા પૂર્ણાંકોમાં રૂપાંતરિત કરે છે. બિટવાઈઝ ઓપરેશન કર્યા પછી, પરિણામ 64 બિટ્સ JavaScript નંબરમાં રૂપાંતરિત થાય છે.

ઓપરેટર	નામ	વર્ણન
&	AND	જો બંને બિટ્સ 1 હોય તો દરેક બીટને 1 પર સેટ કરે છે
	OR	જો બે બિટ્સમાંથી એક 1 હોય તો દરેક બીટને 1 પર સેટ કરે છે
^	XOR	દરેક બીટને 1 પર સેટ કરે છે જો બેમાંથી માત્ર એક બીટ 1 હોય
~	NOT	તમામ બિટ્સને ઉલટાવે છે
<<	Zero fill left shift	જમણી બાજુથી શૂન્યને અંદર દબાવીને ડાબે શિફ્ટ કરો અને ડાબી બાજુના બિટ્સને નીચે પડવા દો
>>	Signed right shift	ડાબી બાજુથી સૌથી ડાબી બાજુની બિટ્સની નકલોને અંદર દબાવવા કરીને જમણે શિફ્ટ કરો અને સૌથી જમણી બાજુના બિટ્સને પડવા દો
>>>	Zero fill right shift	ડાબી બાજુથી શૂન્યને અંદર દબાવવા કરીને જમણે શિફ્ટ કરો અને જમણી બાજુના બિટ્સને નીચે પડવા દો

ઉદાહરણ 10

ઓપરેશન	પરિણામ	ના જેવું સરખું	પરિણામ
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

નોંધ: ઉપરના ઉદાહરણો 4 બિટ્સ સહી વગરના બાઈનરી નંબરોનો ઉપયોગ કરે છે. આના કારણે ~ 5 10 પરત કરે છે. જાવાસ્ક્રિપ્ટ 32 બિટ્સ સાઈન કરેલ પૂર્ણાંકોનો ઉપયોગ કરે છે, તેથી તે 10 પરત કરશે નહીં. તે -6 પરત કરશે. 0000000000000000 0000000000000000 01 (5) 1111111111111111 1111111111111111 10 (~5 = -6)

એક સહી કરેલ પૂર્ણાંક માર્શનસ ચિહ્ન તરીકે ડાબી બાજુના બીટનો ઉપયોગ કરે છે.

સમાનતા ઓપરેટર

બે સમીકરણો સમાન છે કે કેમ તે ચકાસવા માટે સમાનતા ઓપરેટરનો ઉપયોગ થાય છે.

ઉદાહરણ તરીકે “42” અને 42 == સાથે સમાન છે પરંતુ === સાથે સમાન નથી કારણ કે તે માત્ર મૂલ્ય માટે જ નહીં પણ પ્રકાર માટે પણ તપાસે છે.

રિલેશનલ ઓપરેટર

રિલેશનલ ઓપરેટર તપાસે છે કે એક મૂલ્ય અન્ય મૂલ્ય કરતાં વધુ છે કે ઓછું.

```
if(5<2) {
// do something
}
```

ઓપરેટર	કાર્ય
>	કરતા વધારે
<	કરતાં ઓછી છે
>=	કરતાં વધુ અથવા તેના સમાન
<=	કરતાં ઓછું અથવા બરાબર
in	અભિવ્યક્તિમાં મૂલ્ય જોવા મળે છે કે કેમ તે પરીક્ષણ કરે છે
instanceof	અભિવ્યક્તિ ઓબ્જેક્ટનું ઉદાહરણ છે કે કેમ તે પરીક્ષણ કરે છે

ઓપરેટર કરતાં ઓછું એ તપાસે છે કે પ્રથમ મૂલ્ય બીજા મૂલ્ય કરતાં ઓછું છે કે નહીં અને જો માન્ય હોય, તો તે ખોટું પરત કરે છે. ઉપરના ઉદાહરણમાં, 5 એ 2 કરતા ઓછું નથી, તેથી તે સાચું નથી અને if બ્લોકની અંદરનો કોડ એક્ઝિક્યુટ થશે નહીં.

અન્ય ત્રણ ઓપરેટર એ જ રીતે વધુ (>), તેના કરતા વધારે અથવા સમાન (>=), તેનાથી ઓછા અથવા સમાન (<=) માટે તપાસ કરે છે.

ઓપરેટરમાં આપેલ ઇન્ડેક્સ ઓબ્જેક્ટમાં સમાયેલ છે કે કેમ તે તપાસે છે. દાખલા તરીકે:

```
var MyObj= {star:"Algol",constellation:"Perseus"};
if("star" in MyObj) {
// do something
}
```

સ્ટાર એ ઇન્ડેક્સ હોવાથી કોડ કામ કરશે. પરંતુ ઓપરેટરમાં સંખ્યાત્મક પ્રકારો પર કામ કરતા નથી કારણ કે તે ફક્ત સંખ્યાઓ માટે જ કામ કરે છે.

ઓપરેટર ઓબ્જેક્ટ ઓબ્જેક્ટનો દાખલો અથવા ઓબ્જેક્ટ વેરીએબલ ચોક્કસ ઓબ્જેક્ટનો દાખલો છે કે કેમ તે તપાસે છે.

ઉદાહરણ 11

```
var mydate=new Date();
if(mydate instanceof Date) {
```

//do something

}

અહીં mydate એ બિલ્ટ-ઇન Date ઓબ્જેક્ટનું ઉદાહરણ છે. તેથી કોડ if બ્લોકની અંદર એક્ઝિક્યુટ થશે.

યુનરી ઓપરેટર

કાઢી નાખો, રદબાતલ કરો, typeof, ++, --, +, -, ~, ! Javascript માં unary ઓપરેટરો છે.

ઉદાહરણ 12

a = -10;

p=++a;

q=a++;

s=+p;

પ્રી અને પોસ્ટ ઇન્ક્રીમેન્ટ અને ડીક્રીમેન્ટ ઓપરેટર છે.

p=++a; is equivalent to

a=a+1;

p=a;

and q=a++; is equivalent to

q=a;

a=a+1;

ડિલીટ ઓપરેટર

ડિલીટ ઓપરેટરનો ઉપયોગ ઓબ્જેક્ટમાંથી પ્રોપર્ટીઝ ડિલીટ કરવા માટે થઈ શકે છે.

ઉદાહરણ 13

var person = {firstName:"John",lastName:"Doe",age:50,

eyeColor:"blue"};

delete person.age;

ડિલીટ ઓપરેટર ઓબ્જેક્ટ પ્રોપર્ટીઝ પર વાપરવા માટે રચાયેલ છે. ચલો અથવા કાર્યો પર તેની કોઈ અસર નથી.

ડિલીટ ઓપરેટરનો ઉપયોગ પૂર્વવ્યાખ્યાયિત જાવા સ્ક્રિપ્ટ ઓબ્જેક્ટ ગુણધર્મો પર થવો જોઈએ નહીં. તે તમારી એપ્લિકેશનને કેશ કરી શકે છે.

યુનરી + ઓપરેટર

ચલને નંબરમાં કન્વર્ટ કરવા માટે યુનરી + ઓપરેટરનો ઉપયોગ કરી શકાય છે.

ઉદાહરણ 14

var y = "5"; // y is a string

var x = + y; // x is a number

જો ચલને કન્વર્ટ કરી શકાતું નથી, તો તે હજુ પણ સંખ્યા બની જશે, પરંતુ મૂલ્ય NaN (નંબર નહીં):

ઉદાહરણ 15

var y = "John"; // y is a string

var x = + y; // x is a number (NaN)

એ જ રીતે યુનરી - ઓપરેટર પણ ઓપરેટ કરે છે.

ટર્નરી અથવા કન્ડીશનલ ઓપરેટર : જો અન્યથા હોય તો તેનો કોમ્પેક્ટ તરીકે ઉપયોગ કરી શકાય છે.

ઉદાહરણ 16

a = (b>5 ? 4:7) ; means

if(b>5)

a=4;

else

a=7;

સોંપણી ઓપરેટર:

અસાઈનમેન્ટ ઓપરેટરનો ઉપયોગ ચલમાં મૂલ્યો સોંપવા માટે થાય છે. આ સિવાય = નીચે પ્રમાણે કમ્પાઉન્ડ અસાઈનમેન્ટ ઓપરેટરો છે

*=	/=	%=
+=	-=	<<=
>>=	>>>=	&=
^=	=	

a = q; એનો અર્થ એ છે કે q ની કિંમત એ ચલમાં અસાઈન કરવામાં આવે છે જે a ની અગાઉની કિંમતને કાઢી નાખે છે.

હવે a* = 3; a = a*3 ની સમકક્ષ છે; અને તેની જેમ અન્ય તમામ કમ્પાઉન્ડ અસાઈનમેન્ટ ઓપરેટર વર્તે છે

જાવાસ્ક્રિપ્ટમાં કંટ્રોલ સ્ટેટમેન્ટ્સ, લૂપ્સ અને પોપઅપ બોક્સ (Control statements, Loops and Popup boxes in JavaScript)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- નિયંત્રણ નિવેદનો સમજાવો
- વિવિધ લૂપ્સ વિશે ચર્ચા કરો
- પોપઅપ બોક્સના ઉપયોગો સમજાવો.

કંટ્રોલ સ્ટેટમેન્ટ્સ : જ્યારે આપણે કોઈ ચોક્કસ પ્રોગ્રામ માટે કોડ લખીએ છીએ, ત્યારે આપણે કેટલીકવાર વિવિધ ક્રિયાઓ કરવા માટે વિવિધ નિર્ણયો લઈએ છીએ. આ શરતી/કંટ્રોલ સ્ટેટમેન્ટ દ્વારા કરી શકાય છે.

JavaScript માં અમારી પાસે નીચેના શરતી નિવેદનો છે:

એક્ઝેક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે if નો ઉપયોગ કરો, જો ઉલ્લેખિત શરત સાચી હોય જો સમાન શરત ખોટી હોય તો એક્ઝેક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે else નો ઉપયોગ કરો

જો પ્રથમ શરત ખોટી હોય તો ચકાસવા માટે નવી શરતનો ઉલ્લેખ કરવા માટે અન્યનો ઉપયોગ કરો

એક્ઝેક્યુટ કરવાના કોડના ઘણા વૈકલ્પિક બ્લોક્સનો ઉલ્લેખ કરવા માટે સ્વીચનો ઉપયોગ કરો.

જો નિવેદન

જો કોઈ શરત સાચી હોય તો એક્ઝેક્યુટ કરવા માટે જાવાસ્ક્રિપ્ટ કોડના બ્લોકનો ઉલ્લેખ કરવા માટે if સ્ટેટમેન્ટનો ઉપયોગ કરો.

વાક્યરચના

```
If (condition) {
    block of code to be executed if the condition is true
}
```

ઉદાહરણ 1

Make a “Good day” greeting if the time is less than 18:00:

```
if (time < 18) {
    greeting = “Good day”;
}
```

શુભેચ્છાનું પરિણામ આ હશે:

Good day

બીજું નિવેદન

જો શરત ખોટી હોય તો એક્ઝેક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે else સ્ટેટમેન્ટનો ઉપયોગ કરો.

```
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
}
```

```
}
```

ઉદાહરણ 2

જો સમય 18:00 થી ઓછો હોય, તો “શુભ દિવસ” ની શુભેચ્છા બનાવો, અન્યથા “શુભ સાંજ”:

```
if (time < 18) {
    greeting = “Good day”;
} else {
    greeting = “Good evening”;
}
```

શુભેચ્છાનું પરિણામ આ હશે:

Good day

બીજું જો નિવેદન

જો પ્રથમ શરત ખોટી હોય તો નવી શરતનો ઉલ્લેખ કરવા માટે else if સ્ટેટમેન્ટનો ઉપયોગ કરો.

વાક્યરચના

```
if (condition1) {
    block of code to be executed if condition1 is true
} else if (condition2) {
    block of code to be executed if the condition1 is false
    and condition2 is true
} else {
    block of code to be executed if the condition1 is false
    and condition2 is false
}
```

ઉદાહરણ 3

જો સમય 10:00 કરતા ઓછો હોય, તો “ગુડ મોર્નિંગ” શુભેચ્છા બનાવો, જો નહીં, પરંતુ સમય 18:00 કરતા ઓછો છે, તો “ગુડ ડે” શુભેચ્છા બનાવો, અન્યથા “શુભ સાંજ”:

```
if (time < 10) {
    greeting = “Good morning”;
}
```

```

} else if (time < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}

```

The result of x will be:

Good day

JavaScript સ્વિચ સ્ટેટમેન્ટ

ના ઘણા બ્લોક્સમાંથી એકને પસંદ કરવા માટે સ્વિચ સ્ટેટમેન્ટનો ઉપયોગ કરોકોડ અમલમાં મૂકવાનો છે.

વાક્યરચના

```

switch(expression) {
    case n1:
        code block
        break;
    case n2:
        code block
        break;
    default:
        default code block
}

```

આ રીતે તે કાર્ય કરે છે:

- સ્વીચ અભિવ્યક્તિનું મૂલ્યાંકન એકવાર કરવામાં આવે છે.
- અભિવ્યક્તિના મૂલ્યની સરખામણી દરેક કેસના મૂલ્યો સાથે કરવામાં આવે છે.
- જો ત્યાં મેચ હોય, તો કોડનો સંકળાયેલ બ્લોક એક્ઝિક્યુટ થાય છે.

ઉદાહરણ 4

અઠવાડિયાના દિવસના નામની ગણતરી કરવા માટે આજના અઠવાડિયાના દિવસની સંખ્યાનો ઉપયોગ કરો: (રવિવાર=0, સોમવાર=1, મંગળવાર=2,...)

```

switch (new Date().getDay()) {
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
        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;
}

```

The result of day will be:

Tuesday

બ્રેક કીવર્ડ

જ્યારે JavaScript કોડ ઈન્ટરપ્રીટર બ્રેક કીવર્ડ સુધી પહોંચે છે, ત્યારે તે સ્વીચ બ્લોકમાંથી તૂટી જાય છે.

આ બ્લોકની અંદર કોડના વધુ એક્ઝેક્યુશન અને/અથવા કેસ પરીક્ષણને અટકાવશે.

મૂળભૂત કીવર્ડ

જો કોઈ કેસ મેચ ન હોય તો ડિફોલ્ટ કીવર્ડ ચલાવવા માટે કોડનો ઉલ્લેખ કરે છે:

ઉદાહરણ 5

જો આજે શનિવાર કે રવિવાર નથી, તો ડિફોલ્ટ લખોસંદેશ:

```

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";
}

```

ટેકસ્ટનું પરિણામ આ હશે:

વીકએન્ડની રાહ જોઈ રહ્યા છીએ

સામાન્ય કોડ અને ફોલ-થ્રુ

કેટલીકવાર, સ્વીચ બ્લોકમાં, તમે એક જ કોડનો ઉપયોગ કરવા માટે જુદા જુદા કેસ ઇચ્છો છો, અથવા સામાન્ય ડિફોલ્ટમાં આવે છે.

આગલા ઉદાહરણમાંથી નોંધ કરો કે કેસ સમાન કોડ બ્લોક શેર કરી શકે છે અને ડિફોલ્ટ કેસ સ્વીચ બ્લોકમાં છેલ્લો કેસ હોવો જરૂરી નથી.

ઉદાહરણ 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:  
        text = "Now in Weekend";  
}
```

JavaScript લૂપ્સ

લૂપ્સ હાથમાં છે, જો તમે એક જ કોડને વારંવાર, દરેક વખતે અલગ મૂલ્ય સાથે ચલાવવા માંગતા હોવ.

અરે સાથે કામ કરતી વખતે આ ઘણી વાર થાય છે:

Instead of writing:

```
text += train[0] + "<br>";  
text += train [1] + "<br>";  
text += train [2] + "<br>";  
text += train [3] + "<br>";  
text += train [4] + "<br>";  
text += train [5] + "<br>";
```

You can write:

```
for (i = 0; i < train.length; i++) {  
    text += train [i] + "<br>";  
}
```

વિવિધ પ્રકારના લૂપ્સ

JavaScript વિવિધ પ્રકારના લૂપ્સને સપોર્ટ કરે છે:

- માટે - કોડના બ્લોકમાંથી ઘણી વખત લૂપ કરે છે

- ફોર/ઇન - ઑબ્જેક્ટના ગુણધર્મો દ્વારા લૂપ કરે છે
- જ્યારે - ઉલ્લેખિત શરત સાચી હોય ત્યારે કોડના બ્લોકમાંથી લૂપ થાય છે
- do/while - જ્યારે ઉલ્લેખિત શરત સાચી હોય ત્યારે કોડના બ્લોકમાંથી પણ લૂપ થાય છે

આ ફોર લૂપ

જ્યારે તમે લૂપ બનાવવા માંગતા હો ત્યારે લૂપ માટે એ ઘણીવાર સાધન છે જેનો તમે ઉપયોગ કરશો. ફોર લૂપમાં નીચેનું સિન્ટેક્સ છે:

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

વિધાન 1 લૂપ (કોડ બ્લોક) પહેલાં એક્ઝિક્યુટ કરવામાં આવે છે. શરૂ થાય છે. તેને પ્રારંભિક ભાગ કહેવામાં આવે છે

વિધાન 2 લૂપ (કોડ બ્લોક) ચલાવવા માટેની શરત વ્યાખ્યાયિત કરે છે. તેને શરત ભાગ કહેવામાં આવે છે.

લૂપ (કોડ બ્લોક) હોય તે પછી દર વખતે સ્ટેટમેન્ટ 3 ચલાવવામાં આવે છે ચલાવવામાં આવ્યો. તેને વધારો/ઘટાડો ભાગ કહેવામાં આવે છે.

ઉદાહરણ 7

```
for (i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

ઉપરના ઉદાહરણમાંથી, તમે વાંચી શકો છો:

વિધાન 1 લૂપ શરૂ થાય તે પહેલાં ચલ સેટ કરે છે (var i = 0).

વિધાન 2 લૂપ ચલાવવા માટેની સ્થિતિને વ્યાખ્યાયિત કરે છે (હું 5 કરતા ઓછો હોવો જોઈએ).

વિધાન 3 દરેક વખતે જ્યારે લૂપમાં કોડ બ્લોક એક્ઝિક્યુટ કરવામાં આવે ત્યારે મૂલ્ય (i++) વધે છે.

પ્રારંભિક ભાગ

સામાન્ય રીતે તમે લૂપ (var i = 0) માં વપરાતા ચલને શરૂ કરવા માટે સ્ટેટમેન્ટ 1 નો ઉપયોગ કરશો.

આ હંમેશા કેસ નથી, JavaScript કાળજી લેતું નથી. વિધાન 1 વૈકલ્પિક છે. તમે વિધાન 1 (અલ્પવિરામ દ્વારા અલગ) માં ઘણા મૂલ્યો શરૂ કરી શકો છો:

ઉદાહરણ 8

```
for (i = 0, len = train.length, text = ""; i < len; i++) {  
    text += train [i] + "<br>";  
}
```

અને તમે સ્ટેટમેન્ટ 1 છોડી શકો છો (જેમ કે જ્યારે લૂપ શરૂ થાય તે પહેલાં તમારા મૂલ્યો સેટ કરવામાં આવે છે):

ઉદાહરણ 9

```
var i = 2;
```



```
var len = train.length;
```

```
var text = "";
```

```
for (; i < len; i++) {  
    text += train[i] + "<br>";  
}
```

શરત ભાગ

ઘણીવાર વિધાન 2 નો ઉપયોગ પ્રારંભિક ચલની સ્થિતિનું મૂલ્યાંકન કરવા માટે થાય છે.

આ હંમેશા કેસ નથી, JavaScript કાળજી લેતું નથી. વિધાન 2 પણ વૈકલ્પિક છે.

જો સ્ટેટમેન્ટ 2 સાચું પાછું આવે છે, તો લૂપ ફરીથી શરૂ થશે, જો તે ખોટું પાછું આવે છે, તો લૂપ સમાપ્ત થઈ જશે.

જો તમે સ્ટેટમેન્ટ 2 છોડી દો છો, તો તમારે લૂપની અંદર વિરામ આપવો આવશ્યક છે. નહિંતર લૂપ ક્યારેય સમાપ્ત થશે નહીં. આ તમારા બ્રાઉઝરને કેશ કરશે. આ ટ્યુટોરીયલના પછીના પ્રકરણમાં વિરામ વિશે વાંચો.

ઇન્ક્રીમેન્ટ/ડિક્રીમ ભાગ

ઘણીવાર સ્ટેટમેન્ટ 3 પ્રારંભિક ચલને વધારે છે.

આ હંમેશા કેસ નથી, JavaScript કાળજી લેતું નથી, અને વિધાન 3 વૈકલ્પિક છે.

વિધાન 3 નકારાત્મક વૃદ્ધિ (i--), અથવા મોટી વૃદ્ધિ (i = i + 15), અથવા બીજું કંઈપણ કરી શકે છે.

વિધાન 3 પણ અવગણી શકાય છે (જેમ કે જ્યારે તમે લૂપની અંદર તમારા મૂલ્યોમાં વધારો કરો છો):

ઉદાહરણ 10

```
var i = 0;  
len = train.length;  
for (; i < len; ) {  
    text += train[i] + "<br>";  
    i++;  
}
```

માટે/ઇન લૂપ

જાવાસ્ક્રિપ્ટ ફોર/ઇન સ્ટેટમેન્ટ ઓબ્જેક્ટના ગુણધર્મો દ્વારા લૂપ થાય છે:

```
var person = {fname:"Raja",lname:"Sen",age:35};  
var text = "";  
var x;  
for (x in person) {  
    text += person[x];  
}
```

જ્યારે લૂપ

જ્યારે ઉલ્લેખિત શરત સાચી હોય ત્યાં સુધી કોડના બ્લોકમાંથી જ્યારે લૂપ લૂપ થાય છે.

વાક્યરચના

```
while (condition) {  
    code block to be executed  
}
```

નીચેના ઉદાહરણમાં, જ્યાં સુધી ચલ (i) 10 કરતા ઓછું હોય ત્યાં સુધી લૂપમાંનો કોડ વારંવાર ચાલશે:

ઉદાહરણ 11

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

જો તમે શરતમાં ઉપયોગમાં લેવાતા ચલને વધારવાનું ભૂલી જાઓ છો, તો લૂપ ક્યારેય સમાપ્ત થશે નહીં. આ તમારા બ્રાઉઝરને કેશ કરશે.

ડુ/વ્હાઈલ લૂપ

do/while લૂપ એ while લૂપનો એક પ્રકાર છે. આ લૂપ કોડ બ્લોકને એકવાર એક્ઝિક્યુટ કરશે, શરત સાચી છે કે કેમ તે તપાસતા પહેલા, પછી જ્યાં સુધી શરત સાચી છે ત્યાં સુધી તે લૂપનું પુનરાવર્તન કરશે.

વાક્યરચના

```
do {  
    code block to be executed  
}  
while (condition);
```

નીચેનું ઉદાહરણ do/while લૂપનો ઉપયોગ કરે છે. લૂપ હંમેશા ઓછામાં ઓછા એક વખત એક્ઝિક્યુટ કરવામાં આવશે, ભલે શરત ખોટી હોય, કારણ કે શરત ચકાસવામાં આવે તે પહેલાં કોડ બ્લોક એક્ઝિક્યુટ કરવામાં આવે છે:

ઉદાહરણ 12

```
do {  
    text += "The number is " + i;  
    i++;  
}
```

```
while (i < 10);
```

શરતમાં ઉપયોગમાં લેવાતા ચલને વધારવાનું ભૂલશો નહીં, અન્યથા લૂપ ક્યારેય સમાપ્ત થશે નહીં!

માટે અને જ્યારે સરખામણી

જો તમે પાછલું પ્રકરણ વાંચ્યું હોય, ફોર લૂપ વિશે, તો તમને ખબર પડશે કે જ્યારે લૂપ એ ફોર લૂપ જેવું જ છે, જેમાં સ્ટેટમેન્ટ 1 અને સ્ટેટમેન્ટ 3 અવગણવામાં આવ્યું છે.

આ ઉદાહરણમાંનો લૂપ ટ્રેન એરેમાંથી કારના નામ એકત્રિત કરવા માટે લૂપનો ઉપયોગ કરે છે:

ઉદાહરણ 13

```
train = ["Duronto","Satabdi","Garib Rath","Rajdhani"];
var i = 0;
var text = "";
for (;train[i];) {
    text += train[i] + "<br>";
    i++;
}
```

આ ઉદાહરણમાંનો લૂપ ટ્રેન એરેમાંથી કારના નામ એકત્રિત કરવા માટે જ્યારે લૂપનો ઉપયોગ કરે છે:

```
train = ["Duronto","Satabdi","Garib Rath","Rajdhani"];
var i = 0;
var text = "";
while (train[i]) {
    text += train[i] + "<br>";
    i++;
}
```

લૂપમાં બ્રેક સ્ટેટમેન્ટ

બ્રેક સ્ટેટમેન્ટનો ઉપયોગ લૂપને તેની પૂર્ણતા પહેલાં સમાપ્ત કરવા માટે થાય છે. તે નકામી રીતે લૂપને પુનરાવર્તિત ન કરવા માટે મશીનનો સમય બચાવે છે.

ઉદાહરણ તરીકે: રેખીય શોધમાં, જો આપણને આઈટમ મળે તો આપણે લૂપને તોડી શકીએ છીએ કારણ કે તે બિનજરૂરી નથી.

ઉદાહરણ 14

```
for(i=0;i<l;i++) {
    if(arr[i]==item) {
        alert("Found at :"+i);
        fl=1;
        break;
    }
}
```

if(fl==0) alert("Not Found");

અહીં, જો વસ્તુ મળી આવે, તો લૂપ તૂટી જાય છે અને CPU સમય ફિગ 2 સાચવવામાં આવે છે.

પોપઅપ બોક્સ

JavaScript ત્રણ પ્રકારના પોપઅપ બોક્સ ધરાવે છે.

- 1 ચેતવણી બોક્સ
- 2 કન્ફર્મ બોક્સ અને
- 3 પ્રોમ્પ્ટ બોક્સ.

ચેતવણી બોક્સ

જો તમે બનાવવા માંગતા હોવ તો વારંવાર ચેતવણી બોક્સનો ઉપયોગ કરવામાં આવે છે ખાતરી કરો કે માહિતી વપરાશકર્તા સુધી પહોંચે છે. જ્યારે ચેતવણી બોક્સ પોપ અપ થાય છે, ત્યારે વપરાશકર્તાએ આગળ વધવા માટે “ઓકે” ક્લિક કરવું પડશે.

વાક્યરચના

```
window.alert("sometext");
```

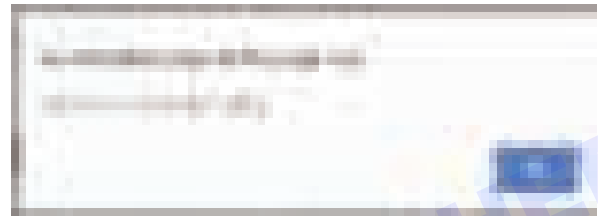
નોંધ: window.alert() પદ્ધતિ વિન્ડો ઉપસર્ગ વગર લખી શકાય છે.

ઉદાહરણ 15

```
alert ("Welcome to Java Script Coding!");
```

પરિણામ ફિગ 1 માં બતાવવામાં આવ્યું છે.

Fig 1



કન્ફર્મ બોક્સ

કન્ફર્મ બોક્સનો ઉપયોગ ઘણીવાર કંઈક ચકાસવા અથવા સ્વીકારવા માટે થાય છે. જ્યારે કન્ફર્મ બોક્સ પોપ અપ થાય છે, ત્યારે વપરાશકર્તાએ આગળ વધવા માટે “ઓકે” અથવા “રદ કરો” પર ક્લિક કરવું પડશે. જો વપરાશકર્તા “ઓકે” પર ક્લિક કરે છે, તો બોક્સ સાચું પાછું આવે છે. જો વપરાશકર્તા “રદ કરો” પર ક્લિક કરે છે, તો બોક્સ ખોટા પરત કરે છે.

વાક્યરચના

```
window.confirm("sometext");
```

નોંધ: window.confirm() પદ્ધતિ વિન્ડો ઉપસર્ગ વગર લખી શકાય છે.

ઉદાહરણ 16

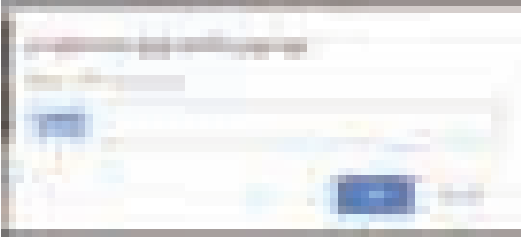
```
if (confirm("Click a button!"))
```

```
{
    txt = "You clicked OK!";
}
else
{
    txt = "You clicked Cancel!";
}
```

પરિણામ ફિગ 2 માં બતાવવામાં આવ્યું છે.

નોંધ: જ્યારે OK બટન પર ક્લિક કરો ત્યારે તે સંદેશ પ્રદર્શિત કરે છે “તમે બરાબર ક્લિક કર્યું!” અને જ્યારે રદ કરો બટન પર ક્લિક કરો ત્યારે તે સંદેશ પ્રદર્શિત કરે છે “તમે રદ કરો ક્લિક કર્યું!”

Fig 2



પ્રોમ્પ્ટ બોક્સ

જો વપરાશકર્તા પૃષ્ઠ દાખલ કરતા પહેલા મૂલ્ય દાખલ કરે તો પ્રોમ્પ્ટ બોક્સનો વારંવાર ઉપયોગ થાય છે. જ્યારે પ્રોમ્પ્ટ બોક્સ પોપ અપ થાય છે, ત્યારે વપરાશકર્તાએ ઇનપુટ મૂલ્ય દાખલ કર્યા પછી આગળ વધવા માટે “ઓકે” અથવા “રદ કરો” પર ક્લિક કરવું પડશે. જો વપરાશકર્તા “ઓકે” પર ક્લિક કરવાથી બોક્સ ઇનપુટ મૂલ્ય પરત કરે છે. જો વપરાશકર્તા “રદ કરો” પર ક્લિક કરે છે, તો બોક્સ નલ પરત કરે છે.

વાક્યરચના

```
window.prompt("sometext","default text");
```

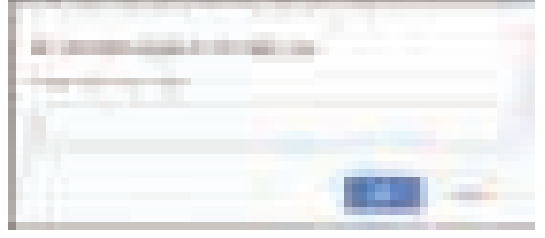
નોંધ: window.prompt() પદ્ધતિ વિન્ડો ઉપસર્ગ વગર લખી શકાય છે.

ઉદાહરણ 17

```
var tname = prompt("Please Enter your Name", "Lakshmi");
if (tname == null || tname == "")
{txt = "User cancelled the prompt.";}
else
{txt = "Hello " + tname + "! Congratulations!!!!!!";}
}
```

પરિણામ ફિગ 3 માં બતાવવામાં આવ્યું છે.

Fig 3



નોંધ: જો OK બટન પર ક્લિક કરો તો તે સંદેશ પ્રદર્શિત કરે છે “હેલો લક્ષ્મી! અભિનંદન!!!” જો આકૃતિ 4 માં બતાવ્યા પ્રમાણે ‘લક્ષ્મી’ નામ રદ કરવામાં આવે તો તે “વપરાશકર્તાએ પ્રોમ્પ્ટને રદ કર્યો” સંદેશ આપે છે. જ્યારે કેન્સલ બટન પર ક્લિક કરો ત્યારે પણ જો બોક્સમાં લખાણ ‘લક્ષ્મી’ હોય તો પણ તે સંદેશ આપે છે “વપરાશકર્તાએ પ્રોમ્પ્ટ રદ કર્યો”.

Fig 4



લાઈન બ્રેક્સ

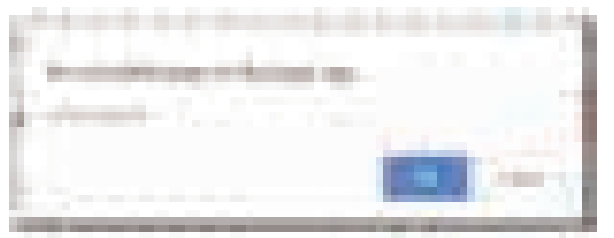
પોપઅપ બોક્સની અંદર લાઈન બ્રેક પ્રદર્શિત કરવા માટે, અક્ષર n દ્વારા અનુસરતા બેકસ્લેશનો ઉપયોગ કરો.

ઉદાહરણ 18

```
alert("Hello\nWelcome!");
```

પરિણામ ફિગ 5 માં બતાવવામાં આવ્યું છે.

Fig 5



JavaScript માં અરે (Arrays in JavaScript)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- અરે વ્યાખ્યાયિત કરો
- અરેના ખ્યાલો સમજાવો
- અરે પદ્ધતિઓનું વર્ણન કરો
- અરેની સોર્ટિંગ જાણો.

અરે શું છે?

અરે એ સ્પેશિયલ વેરિએબલ છે, જે એક સમયે એક કરતાં વધુ મૂલ્યને પકડી શકે છે.

જો તમારી પાસે આઈટમ્સની સૂચિ છે (ઉદાહરણ તરીકે, ટ્રેનના નામોની સૂચિ), તો ટ્રેનોને સિંગલ વેરીએબલ્સમાં સ્ટોર કરવી આના જેવી દેખાઈ શકે છે.

```
var train1 = "Jan Satabdi";
```

```
var train1 = "Garib Rath";
```

```
var train1 = "Duronto";
```

જો કે, જો તમે ટ્રેનમાંથી લૂપ કરવા અને કોઈ ચોક્કસ શોધવા માંગતા હોવ તો શું? અને જો તમારી પાસે 3 નહીં, પણ 300 હોય તો?

ઉકેલ એ અરે છે!

JavaScript અરે

JavaScript અરેનો ઉપયોગ એક વેરીએબલમાં બહુવિધ મૂલ્યોને સંગ્રહિત કરવા માટે થાય છે.

અરેનો ઉપયોગ કરીને

અરે બનાવવું એ JavaScript અરે બનાવવાનો સૌથી સરળ રસ્તો છે.

વાક્યરચના:

```
var array-name = [item1, item2, ...];
```

ઉદાહરણ 1

```
var trains = ["Duronto", "Jan Satabdi", "RAJDHANI"];
```

JavaScript કીવર્ડ નવા નો ઉપયોગ કરવો

નીચેનું ઉદાહરણ પણ અરે બનાવે છે અને તેને મૂલ્યો અસાઈન કરે છે:

ઉદાહરણ 2

```
var trains = new Array("Duronto", "Jan Satabdi", "RAJDHANI");
```

ઉપરોક્ત બે ઉદાહરણો બરાબર એ જ કરે છે. નવા Array() નો ઉપયોગ કરવાની જરૂર નથી.

સરળતા, વાંચનક્ષમતા અને અમલની ઝડપ માટે, પ્રથમ (અરે શાબ્દિક પદ્ધતિ) નો ઉપયોગ કરો.

અરેના તત્વોને ઍક્સેસ કરો

તમે ઇન્ડેક્સ નંબરનો ઉલ્લેખ કરીને અરે એલિમેન્ટનો સંદર્ભ લો છો.

આ નિવેદન માયટ્રેન્સમાં પ્રથમ ઘટકની કિંમતને ઍક્સેસ કરે છે:

```
var name = trains[0];
```

આ નિવેદન ટ્રેનોમાં પ્રથમ તત્વને સંશોધિત કરે છે:

```
trains[0] = "Jan Satabdi";
```

[0] એ અરેમાંનું પ્રથમ તત્વ છે. [1] બીજો છે. અરે ઇન્ડેક્સ 0 થી શરૂ થાય છે.

અરે પ્રદર્શિત કરી રહ્યા છીએ : id="demo" સાથે એલિમેન્ટની અંદર અરે પ્રદર્શિત કરવા માટે અમે સ્ક્રિપ્ટનો ઉપયોગ કરીશું:

ઉદાહરણ 3

```
<p id="demo"></p>
```

```
<script>
```

```
var trains = ["Duronto", "Jan Satabdi", "RAJDHANI"];
```

```
document.getElementById("demo").innerHTML = trains;
```

```
</script>
```

પ્રથમ લીટી (સ્ક્રીપ્ટમાં) ટ્રેનો નામની અરે બનાવે છે.

બીજી લાઈન id="demo" સાથે તત્વ "શોધે છે" અને તેના "innerHTML" માં અરેને "પ્રદર્શિત કરે છે".

જગ્યાઓ અને રેખા વિરામ મહત્વપૂર્ણ નથી. એક ઘોષણા બહુવિધ રેખાઓ સુધી ફેલાવી શકે છે.

ઉદાહરણ 4

```
var trains = [
```

```
"Duronto",
```

```
"Jan Satabdi",
```

```
"RAJDHANI"
```

```
];
```

છેલ્લા તત્વ (જેમ કે "રાજધાની") પછી અલ્પવિરામ ન મૂકશો. તે સમગ્ર બ્રાઉઝર્સમાં અસંગત છે.

એક અરે એક નામ હેઠળ ઘણી કિંમતો ધરાવે છે અને તમે ઇન્ડેક્સ નંબરનો ઉલ્લેખ કરીને મૂલ્યોને ઍક્સેસ કરી શકો છો.

તમારી પાસે એક અરેમાં વિવિધ વસ્તુઓ હોઈ શકે છે

JavaScript ચલો ઑબ્જેક્ટ હોઈ શકે છે. અરે એ ખાસ પ્રકારની વસ્તુઓ છે.

આ કારણે, તમારી પાસે સમાન એરેમાં વિવિધ પ્રકારના ચલ હોઈ શકે છે.

તમારી પાસે એરેમાં વસ્તુઓ હોઈ શકે છે. તમારી પાસે એરેમાં કાર્યો હોઈ શકે છે. તમારી પાસે એરેમાં એરે હોઈ શકે છે:

```
my Array[0] = Date.now;
```

```
my Array [1] = my Function;
```

```
my Array [2] = myTrains;
```

એરે ઓબ્જેક્ટો છે

એરે એ એક ખાસ પ્રકારની વસ્તુઓ છે. JavaScript માં ઓપરેટરનો પ્રકાર એરે માટે “ઓબ્જેક્ટ” પરત કરે છે.

પરંતુ, JavaScript એરેને એરે તરીકે શ્રેષ્ઠ રીતે વર્ણવવામાં આવે છે.

એરે તેના “તત્વો” ને એક્સેસ કરવા માટે સંખ્યાઓનો ઉપયોગ કરે છે. આ ઉદાહરણમાં, વ્યક્તિ[0] રાજાને પરત કરે છે:

એરે

```
var person = ["Raja", "Sen", 46];
```

ઓબ્જેક્ટ્સ તેના “સભ્યો” ને એક્સેસ કરવા માટે નામોનો ઉપયોગ કરે છે. આ ઉદાહરણમાં, person.firstName રાજા પરત કરે છે:

ઓબ્જેક્ટ

```
var person = {firstName:"Raja", lastName:"Sen", age:46};
```

લંબાઈ મિલકત

એરેની લંબાઈ ગુણધર્મ એરેની લંબાઈ પરત કરે છે

(એરે તત્વોની સંખ્યા).

ઉદાહરણ 6

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.length; // the length of fruits is 4
```

લંબાઈ ગુણધર્મ હંમેશા સર્વોચ્ચ એરે ઇન્ડેક્સ કરતાં એક વધુ હોય છે.

એરે તત્વો ઉમેરી રહ્યા છે

એરેમાં નવું તત્વ ઉમેરવાનો સૌથી સહેલો રસ્તો લંબાઈ ગુણધર્મનો ઉપયોગ કરવાનો છે:

ઉદાહરણ 7

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits[fruits.length] = "Lemon"; // adds a new element
```

(Lemon) to fruits

ઉચ્ચ અનુક્રમણિકાઓ સાથે તત્વો ઉમેરવાથી એરેમાં અવ્યાખ્યાયિત “છિદ્રો” બનાવી શકે છે:

ઉદાહરણ 8

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits[10] = "Lemon"; // adds a new element (Lemon) to
```

fruits

લૂપિંગ એરે તત્વો

એરે દ્વારા લૂપ કરવાની શ્રેષ્ઠ રીત લૂપ માટે માનકનો ઉપયોગ કરી રહી છે:

ઉદાહરણ 9

```
var index;
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
for (index = 0; index < fruits.length; index++) {
```

```
text += fruits[index];
```

```
}
```

સહયોગી એરે? નો વે!

ઘણી પ્રોગ્રામિંગ ભાષાઓ નામાંકિત ઇન્ડેક્સ સાથે એરેને સપોર્ટ કરે છે.

નામાંકિત ઇન્ડેક્સ સાથેના એરેને એસોસિએટીવ એરે (અથવા હેશ) કહેવામાં આવે છે.

JavaScript નામાંકિત અનુક્રમણિકાઓ સાથે એરેને સપોર્ટ કરતું નથી.

Wrong:

```
var person = new Array()
```

```
person ["firstName"] = "Raja";
```

```
person ["lastName"] = "Sen";
```

```
person ["age"] = 46;
```

ઉપરનું ઉદાહરણ એવું લાગે છે કે તે કામ કરે છે. પણ એવું થતું નથી.

જો તમે તેનો પ્રયાસ કરો છો, તો વ્યક્તિ ["firstName"] રાજાને પરત કરશે, પરંતુ વ્યક્તિ [0] અવ્યાખ્યાયિત પરત કરશે, અને વ્યક્તિ. લંબાઈ 0 આપશે.

જો તમે સહયોગી એરે બનાવવા માંગો છો, તો તેના બદલે એક ઓબ્જેક્ટ બનાવો.

એરેનો ઉપયોગ ક્યારે કરવો? ઓબ્જેક્ટ્સનો ઉપયોગ ક્યારે કરવો?

JavaScript સહયોગી એરેને સપોર્ટ કરતું નથી.

તમારે ઓબ્જેક્ટનો ઉપયોગ કરવો જોઈએ જ્યારે તમે ઈચ્છો કે તત્વના નામો શબ્દમાળાઓ હોય.

તમારે એરેનો ઉપયોગ કરવો જોઈએ જ્યારે તમે ઈચ્છો કે તત્વના નામ ક્રમિક નંબરો હોય.

નવા એરે () ને ટાળો

JavaScript ના બિલ્ટ-ઇન એરે કન્સ્ટ્રક્ટર નવા Array() નો ઉપયોગ કરવાની જરૂર નથી.

તેના બદલે [] નો ઉપયોગ કરો.

આ બે અલગ અલગ નિવેદનો બંને પોઈન્ટ નામના નવા ખાલી એરે બનાવે છે.

```
var points = new Array(); // Bad
```

```
var points = []; // Good
```

આ બે અલગ-અલગ નિવેદનો બંને 6 નંબરો ધરાવતી નવી એરે બનાવે છે.

```
var points = new Array(40, 100, 1, 5, 25, 10) // Bad
```

```
var points = [40, 100, 1, 5, 25, 10]; // Good
```

નવો કીવર્ડ તમારા કોડને જટિલ બનાવે છે અને ખરાબ આડઅસર પેદા કરે છે.

```
var points = new Array(40,100); // Creates an array with two elements (40 and 100)
```

જો હું ઘટકોમાંથી એકને દૂર કરું તો શું?

```
var points = new Array(40); // Creates an array with 40
```

undefined elements !!!!!

એરેને કેવી રીતે ઓળખવું?

એક સામાન્ય પ્રશ્ન છે: હું કેવી રીતે જાણી શકું કે ચલ એ છે એરે?

The problem is that the JavaScript operator type of returns “object”:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
typeof fruits; // typeof returns object
```

ઓપરેટરનો પ્રકાર ઓબ્જેક્ટ પરત કરે છે કારણ કે JavaScript એરે ઓબ્જેક્ટ છે.

આ સમસ્યાને ઉકેલવા માટે તમે તમારું પોતાનું `isArray()` ફંક્શન બનાવી શકો છો:

```
function isArray(myArray) {  
  return myArray.constructor.toString().indexOf("Array") > 1;  
}
```

જો દલીલ એરે હોય તો ઉપરોક્ત ફંક્શન હંમેશા સાચું પરત કરે છે.

અથવા વધુ સ્પષ્ટ રીતે: જો દલીલનો ઓબ્જેક્ટ પ્રોટો પ્રકાર [ઓબ્જેક્ટ એરે] હોય તો તે સાચું પરત કરે છે”.

JavaScript એરે પદ્ધતિઓ

એરેને સ્ટ્રીંગ્સમાં રૂપાંતરિત કરવું

toString() method

JavaScript પદ્ધતિ `toString()` એરેને (અલ્પવિરામથી અલગ કરેલ) એરે મૂલ્યોની સ્ટ્રીંગમાં રૂપાંતરિત કરે છે.

ઉદાહરણ 10

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];  
document.getElementById("demo").innerHTML = trade.  
toString();
```

પરિણામ

COPA,IT,ICTSM,CHNM,Fitter

join() method

`join()` પદ્ધતિ બધા એરે તત્વોને સ્ટ્રીંગમાં જોડે છે. તે `toString()` ની જેમ જ વર્તે છે, પરંતુ વધુમાં તમે વિભાજકનો ઉલ્લેખ કરી શકો છો.

ઉદાહરણ 11

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];
```

```
document.getElementById("demo").innerHTML = trade.  
join("-");
```

પરિણામ

COPA - IT - ICTSM - CHNM - Fitter

પોપિંગ અને દબાણ

જ્યારે તમે એરે સાથે કામ કરો છો, ત્યારે તત્વોને દૂર કરવા અને નવા ઘટકો ઉમેરવાનું સરળ છે.

પોપિંગ

`pop()` પદ્ધતિ એરેમાંથી છેલ્લું તત્વ દૂર કરે છે.

ઉદાહરણ 12

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];  
trade.pop(); // Removes the last element ("Fitter")  
from trade.
```

પરિણામ

COPA,IT,ICTSM,CHNM

The `pop()` method returns the value that was “popped out”.

ઉદાહરણ 13

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
var x = trade.pop(); // the value of x is "CHNM".
```

દબાણ

`push()` પદ્ધતિ એરેમાં એક નવું તત્વ ઉમેરે છે (અંતમાં).

ઉદાહરણ 14

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.push("DTPO"); // Adds a new element ("DTPO")  
to trade.
```

પરિણામ

COPA,IT,ICTSM,CHNM,DTPO

`push()` પદ્ધતિ નવી એરે લંબાઈ પરત કરે છે.

ઉદાહરણ 15

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
var x = trade.push("DTPO"); // the value of x is 5
```

સ્થળાંતર તત્વો

શિફ્ટિંગ એ પોપિંગ સમાન છે, છેલ્લાને બદલે પ્રથમ તત્વ પર કામ કરવું. `shift()` પદ્ધતિ પ્રથમ એરે ઘટકને દૂર કરે છે અને અન્ય તમામ ઘટકોને નીચલા ઇન્ડેક્સમાં “શિફ્ટ” કરે છે.

ઉદાહરણ 16

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.shift(); // Removes the first element "COPA" from  
trade.
```


પરિણામ

IT,ICTSM,CHNM

shift() પદ્ધતિ એ સ્ટ્રિંગ પરત કરે છે જે “શિફ્ટ આઉટ” હતી:

ઉદાહરણ 17

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.shift(); // Returns "COPA"
```

unshift() પદ્ધતિ એરેમાં એક નવું તત્વ ઉમેરે છે (શરૂઆતમાં), અને જૂના તત્વોને “unshifts” કરે છે.

ઉદાહરણ 18

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
fruits.unshift("ElecMech"); // Adds a new  
element "ElecMech"  
to trade
```

પરિણામ

ElecMech,COPA,IT,ICTSM,CHNM

The unshift() method returns the new array length.

ઉદાહરણ 19

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.unshift("ElecMech"); // Returns 5
```

બદલાતા તત્વો

એરે તત્વો તેમના અનુક્રમણિકા નંબરનો ઉપયોગ કરીને ઍક્સેસ કરવામાં આવે છે: એરે અનુક્રમણિકા 0 થી શરૂ થાય છે. [0] એ પ્રથમ એરે ઘટક છે, [1] બીજું છે, [2] ત્રીજું છે.

ઉદાહરણ 20

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade[2] = "DTPO"; // Changes the third element of  
trade to "DTPO"
```

પરિણામ

COPA,IT,DTPO,CHNM

લંબાઈ ગુણધર્મ એરેમાં નવા ઘટકને જોડવાની સરળ રીત પ્રદાન કરે છે:

ઉદાહરણ 21

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];  
trade[trade.length] = "ICTSM"; // Appends  
"ICTSM" to  
fruits
```

પરિણામ

COPA,IT,DTPO,CHNM,ICTSM

તત્વો કાઢી રહ્યા છીએ

JavaScript એરે ઑબ્જેક્ટ હોવાથી, JavaScript ઑપરેટર ડિલીટનો ઉપયોગ કરીને તત્વોને કાઢી શકાય છે.

ઉદાહરણ 22

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];  
delete trade[0]; // Changes the first element in  
trade to undefined
```

નોંધ: ડિલીટનો ઉપયોગ કરવાથી એરેમાં અવ્યાખ્યાયિત છિદ્રો રહી શકે છે. તેના બદલે pop() અથવા shift() નો ઉપયોગ કરો

એરેને વિભાજિત કરવું

splice() પદ્ધતિનો ઉપયોગ એરેમાં નવી આઈટમ ઉમેરવા માટે કરી શકાય છે:

ઉદાહરણ 23

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];  
trade.splice(2, 0, "Turner", "Machinist");
```

નોંધ: પ્રથમ પરિમાણ (2) તે સ્થાનને વ્યાખ્યાયિત કરે છે જ્યાં નવા તત્વો ઉમેરવા જોઈએ (તેમાં કાપેલા). બીજા પરિમાણ (0) એ વ્યાખ્યાયિત કરે છે કે કેટલા તત્વો દૂર કરવા જોઈએ. બાકીના પરિમાણો (“ટર્નર”, “મશિનિસ્ટ”) ઉમેરવા માટે નવા તત્વો વ્યાખ્યાયિત કરો.

પરિણામ

COPA,IT,Turner,Machinist,DTPO,CHNM

તત્વોને દૂર કરવા માટે splice() નો ઉપયોગ કરવો

ચતુર પરિમાણ સેટિંગ સાથે, તમે એરેમાં “છિદ્રો” છોડ્યા વિના તત્વોને દૂર કરવા માટે splice() નો ઉપયોગ કરી શકો છો.

ઉદાહરણ 24

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];  
trade.splice(0, 1); // Removes the first element of  
trade
```

પરિણામ

IT,DTPO,CHNM

નોંધ : પ્રથમ પરિમાણ (0) તે સ્થાનને વ્યાખ્યાયિત કરે છે જ્યાં નવા તત્વો ઉમેરવા જોઈએ (તેમાં કાપેલા). બીજું પરિમાણ (1) વ્યાખ્યાયિત કરે છે કે કેટલા તત્વો દૂર કરવા જોઈએ. બાકીના પરિમાણો અવગણવામાં આવે છે. કોઈ નવા ઘટકો ઉમેરવામાં આવશે નહીં.

એરેને મર્જ કરવું અથવા જોડવું

concat() પદ્ધતિ હાલના એરેને મર્જ કરીને નવી એરે બનાવે છે.

ઉદાહરણ 25 (બે એરેને મર્જ કરવું)

```
var names1 = ["Devi", "Deepa"];  
var names2 = ["Poorna", "Saranya", "Shalini"];  
var myTrainee = names1.concat(names2)  
//Concatenates (joins) names1 and names2.
```

પરિણામ

નોંધ: concat() પદ્ધતિ વર્તમાન એરેને બદલતી નથી. તે હંમેશા નવી એરે આપે છે.

concat() પદ્ધતિ કોઈપણ સંખ્યામાં એરે દલીલો લઈ શકે છે.

ઉદાહરણ 26 (ત્રણ એરેને મર્જ કરવું)

```
var arr1 = ["Priya", "Mythili"];
var arr2 = ["Sangeetha", "Nancy", "Sahana"];
var arr3 = ["Ramya", "Kavi"];
var myTrainee = arr1.concat(arr2, arr3);
```

// Concatenates arr1 with arr2 and arr3

concat() પદ્ધતિ દલીલો તરીકે મૂલ્યો પણ લઈ શકે છે.

ઉદાહરણ 27 (મૂલ્યો સાથે એરેને મર્જ કરવું)

```
var arr1 = ["Priya", "Mythili"];
var myTrainee = arr1.concat(["Ramya", "Kavi"]);
```

એરે સ્લાઈસિંગ

સ્લાઈસ() પદ્ધતિ એરેના ટુકડાને નવા એરેમાં કાપી નાખે છે. આ ઉદાહરણ એરે એલિમેન્ટ 2 ("DTPO") થી શરૂ થતા એરેના એક ભાગને કાપી નાખે છે. સ્લાઈસ() પદ્ધતિ નવી એરે બનાવે છે. તે સ્ત્રોત એરેમાંથી કોઈપણ ઘટકોને દૂર કરતું નથી.

ઉદાહરણ 28

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(2);
```

સ્લાઈસ() પદ્ધતિ સ્લાઈસ (1, 3) જેવી બે દલીલો લઈ શકે છે. પદ્ધતિ પછી શરૂઆતની દલીલમાંથી ઘટકો પસંદ કરે છે, અને અંતિમ દલીલ સુધી (પરંતુ શામેલ નથી).

ઉદાહરણ 29

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(1, 3);
```

જો અંતિમ દલીલ અવગણવામાં આવે છે, જેમ કે પ્રથમ ઉદાહરણોમાં, સ્લાઈસ() પદ્ધતિ બાકીના એરેને કાપી નાખે છે.

ઉદાહરણ 30

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(2);
```

સ્થાપિત toString()

જ્યારે આદિમ મૂલ્ય અપેક્ષિત હોય ત્યારે JavaScript આપમેળે એરેને અલ્પવિરામથી અલગ કરેલ સ્ટ્રિંગમાં રૂપાંતરિત કરે છે. જ્યારે તમે એરે આઉટપુટ કરવાનો પ્રયાસ કરો છો ત્યારે આ હંમેશા કેસ છે.

આ બે ઉદાહરણો સમાન પરિણામ આપશે:

ઉદાહરણ 31

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById("demo").innerHTML =
trade.toString();
```

ઉદાહરણ 32

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById("demo").innerHTML = trade;
```

નોંધ: બધા JavaScript ઓબ્જેક્ટમાં toString() પદ્ધતિ હોય છે.

એરેને સોર્ટ કરી રહ્યાં છીએ

sort() પદ્ધતિ એરેને મૂળાક્ષરો પ્રમાણે સોર્ટ કરે છે.

ઉદાહરણ 33

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); // Sorts the elements of trade
```

પરિણામ

CHNM, COPA, ICTSM, IT

એરેને ઉલટાવી રહ્યું છે

રિવર્સ() પદ્ધતિ એરેમાંના તત્વોને ઉલટાવે છે. તમે તેનો ઉપયોગ ઉતરતા ક્રમમાં એરેને સોર્ટ કરવા માટે કરી શકો છો.

ઉદાહરણ 34

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); // Sorts the elements of
trade;
trade.reverse(); // Reverse the order of the
elements
```

જાવાસ્ક્રિપ્ટમાં એનિમેશન અને મલ્ટીમીડિયા ફાઈલોની વિભાવનાઓ (Concepts of Animation and Multimedia files in JavaScript)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- JavaScript માં એનિમેશન સેટિંગ્સ જાણો
- JavaScript માં મલ્ટીમીડિયા સમજાવો.

એનિમેશન

તત્વોની શૈલી

એનિમેશન શક્ય બનાવવા માટે, એનિમેટેડ તત્વ “પેરેન્ટ કન્ટેનર” ની તુલનામાં એનિમેટેડ હોવું આવશ્યક છે.

કન્ટેનર તત્વ શૈલી = “સ્થિતિ: સંબંધિત” સાથે બનાવવું જોઈએ.

એનિમેશન તત્વ શૈલી = “પોઝિશન: સંપૂર્ણ” સાથે બનાવવું જોઈએ.

ઉદાહરણ

<!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>
?

```

એનિમેશન કોડ

JavaScript એનિમેશન એ એલિમેન્ટની શૈલીમાં ક્રમિક ફેરફારો પ્રોગ્રામિંગ દ્વારા કરવામાં આવે છે. ફેરફારોને ટાઈમર દ્વારા બોલાવવામાં આવે છે. જ્યારે ટાઈમર અંતરાલ નાનો હોય છે, ત્યારે એનિમેશન સતત દેખાય છે. મૂળભૂત કોડ છે:

ઉદાહરણ

```

var id = setInterval(frame, 5);
function frame() {
if (/* test for finished */) {
clearInterval(id);
} else {
/* code to change the element style */
}
}

```

JavaScript નો ઉપયોગ કરીને એનિમેશન બનાવો

ઉદાહરણ

```

<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>
</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>

મલ્ટીમીડિયા ફાઇલો

મલ્ટીમીડિયા શું છે?

મલ્ટીમીડિયા ઘણા વિવિધ ફોર્મેટમાં આવે છે. તે તમે સાંભળી અથવા જોઈ શકો તે લગભગ કંઈપણ હોઈ શકે છે. વેબ પૃષ્ઠોમાં ઘણીવાર વિવિધ પ્રકારો અને ફોર્મેટના મલ્ટીમીડિયા ઘટકો હોય છે.

ઉદાહરણો: છબીઓ, સંગીત, ધ્વનિ, વિડિઓઝ, રેકોર્ડ્સ, ફિલ્મો, એનિમેશન અને વધુ.

મલ્ટીમીડિયા ફોર્મેટ્સ

મલ્ટીમીડિયા તત્વો (જેમ કે ઓડિયો અથવા વિડિયો) મીડિયા ફાઇલોમાં સંગ્રહિત થાય છે. ફાઇલનો પ્રકાર શોધવાની સૌથી સામાન્ય રીત, ફાઇલ એક્સટેન્શનને જોવાનું છે. મલ્ટીમીડિયા ફાઇલોમાં ફોર્મેટ અને વિવિધ એક્સટેન્શન હોય છે જેમ કે: .swf, .wav, .mp3, .mp4, .mpg, .wmv અને .avi.

HTML માં વિડિઓઝ ચલાવી રહ્યા છીએ

HTML માં વિડિઓ બતાવવા માટે, તત્વનો ઉપયોગ કરો:

ઉદાહરણ

```
<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>
```

તે કેવી રીતે કામ કરે છે

કંટ્રોલ્સ એટ્રિબ્યુટ વિડિઓ નિયંત્રણો ઉમેરે છે, જેમ કે પ્લે, પોઝ અને વોલ્યુમ. હંમેશા પહોળાઈ અને ઊંચાઈના લક્ષણોનો સમાવેશ કરવો એ સારો વિચાર છે. જો ઊંચાઈ અને પહોળાઈ સેટ કરેલી ન હોય, તો વિડિયો લોડ થાય ત્યારે પેજ ફિલ્લકર થઈ શકે છે. તત્વ તમને વૈકલ્પિક વિડિયો ફાઇલોનો ઉલ્લેખ કરવાની મંજૂરી આપે છે જેમાંથી બ્રાઉઝર પસંદ કરી શકે છે. બ્રાઉઝર પ્રથમ માન્ય ફોર્મેટનો ઉપયોગ કરશે. અને ટેગ્સ વચ્ચેનો ટેક્સ્ટ ફક્ત એવા બ્રાઉઝર્સમાં જ પ્રદર્શિત થશે જે એલિમેન્ટને સપોર્ટ કરતા નથી.

HTML <video> Autoplay

વિડિયો શરૂ કરવા માટે ઓટોપ્લે એટ્રિબ્યુટનો ઓટોમેટિક રીતે ઉપયોગ કરો:

ઉદાહરણ

```
<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>
```

નોંધ: આઈપેડ અને આઈફોન જેવા મોબાઇલ ઉપકરણોમાં ઓટોપ્લે એટ્રિબ્યુટ કામ કરતું નથી

HTML વિડિઓ - મીડિયા પ્રકારો

ફાઇલ ફોર્મેટ	મીડિયા પ્રકાર
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML વિડિઓ - પદ્ધતિઓ, ગુણધર્મો અને ઘટનાઓ

HTML5 એ તત્વ માટે DOM પદ્ધતિઓ, ગુણધર્મો અને ઇવેન્ટ્સને વ્યાખ્યાયિત કરે છે. આ તમને વિડિયો લોડ કરવા, ચલાવવા અને થોભાવવાની તેમજ સમયગાળો અને વોલ્યુમ સેટ કરવાની પરવાનગી આપે છે. ત્યાં DOM ઇવેન્ટ્સ પણ છે જે તમને સૂચિત કરી શકે છે જ્યારે વિડિઓ ચલાવવાનું શરૂ થાય છે, થોભાવવામાં આવે છે, વગેરે.

HTML5 વિડિઓ ટેગ્સ

ટેગ	વર્ણન
<video>	વિડિઓ અથવા મૂવી વ્યાખ્યાયિત કરે છે
<source>	મીડિયા ઘટકો માટે બહુવિધ મીડિયા સંસાધનોને વ્યાખ્યાયિત કરે છે
<track>	મીડિયા પ્લેયર્સમાં ટેક્સ્ટ ટ્રેક વ્યાખ્યાયિત કરે છે

વેબ પર ઓડિયો

HTML5 તત્વ વેબ પેજમાં ઓડિયોને એમ્બેડ કરવાની માનક રીતનો ઉલ્લેખ કરે છે.

HTML એલિમેન્ટ

HTML માં ઓડિઓ ફાઇલ ચલાવવા માટે, તત્વનો ઉપયોગ કરો:

ઉદાહરણ

```
<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 ઓડિઓ - તે કેવી રીતે કાર્ય કરે છે

નિયંત્રણ વિશેષતા ઓડિઓ નિયંત્રણો ઉમેરે છે, જેમ કે પ્લે, પોઝ અને વોલ્યુમ. તત્વ તમને વૈકલ્પિક ઓડિઓ ફાઇલોનો ઉલ્લેખ કરવાની મંજૂરી આપે છે જેમાંથી બ્રાઉઝર પસંદ કરી શકે છે. બ્રાઉઝર પ્રથમ માન્ય ફોર્મેટનો ઉપયોગ કરશે. અને ટેગ્સ વચ્ચેનો ટેક્સ્ટ ફક્ત એવા બ્રાઉઝર્સમાં જ પ્રદર્શિત થશે જે એલિમેન્ટને સપોર્ટ કરતા નથી.

HTML ઓડિઓ - મીડિયા પ્રકારો

ફાઇલ ફોર્મેટ	મીડિયા પ્રકાર
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML ઑડિઓ - પદ્ધતિઓ, ગુણધર્મો અને ઘટનાઓ

HTML5 એ તત્વ માટે DOM પદ્ધતિઓ, ગુણધર્મો અને ઇવેન્ટ્સને વ્યાખ્યાયિત કરે છે. આ તમને ઑડિયોને લોડ કરવા, ચલાવવા અને થોભાવવાની તેમજ સમયગાળો અને વોલ્યુમ સેટ કરવાની મંજૂરી આપે છે. ત્યાં DOM ઇવેન્ટ્સ પણ છે જે તમને સૂચિત કરી શકે છે જ્યારે ઑડિઓ ચલાવવાનું શરૂ થાય છે, થોભાવવામાં આવે છે, વગેરે.

HTML5 ઑડિઓ ટેગ્સ

ટેગ	વર્ણન
<audio>	ધ્વનિ સામગ્રી વ્યાખ્યાયિત કરે છે
<source>	મીડિયા ઘટકો માટે બહુવિધ મીડિયા સંસાધનોને વ્યાખ્યાયિત કરે છે, જેમ કે અને

© NIMI
NOT TO BE REPUBLISHED

JavaScript નો ઉપયોગ કરીને ગતિશીલ HTML પૃષ્ઠો વિકસાવો (Develop dynamic HTML pages using JavaScript)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- કાર્ય વ્યાખ્યાયિત કરો
- કાર્યની કામગીરી સમજાવો
- ફંક્શનનો ઉપયોગ કરવાના ફાયદા સમજાવો
- ચલોનો અવકાશ સમજાવો.

JavaScript ફંક્શન્સ: JavaScript ફંક્શન એ બ્લોક છે ચોક્કસ કાર્ય કરવા માટે રચાયેલ કોડનો.

JavaScript ફંક્શન એક્ટ્રિક્યુટ થાય છે જ્યારે “કંઈક” તેને બોલાવે છે (તેને બોલાવે છે).

ઉદાહરણ 1

```
function myFunction(p1, p2) {
  return p1 * p2; // the function returns the product of p1
  and p2
}
```

JavaScript ફંક્શન સિન્ટેક્સ

JavaScript ફંક્શનને ફંક્શન કીર્ડ સાથે વ્યાખ્યાયિત કરવામાં આવે છે, ત્યારબાદ નામ અને ત્યારબાદ કૌંસ ().

ફંક્શન નામોમાં અક્ષરો, અંકો, અન્ડરસ્કોર અને ડોલર ચિહ્નો હોઈ શકે છે (ચલો જેવા જ નિયમો).

કૌંસમાં અલ્પવિરામ દ્વારા વિભાજિત પેરામીટર નામો શામેલ હોઈ શકે છે: (પેરામીટર1, પેરામીટર2, ...)

કાર્ય દ્વારા એક્ટ્રિક્યુટ કરવા માટેનો કોડ સર્પાકાર કૌંસની અંદર મૂકવામાં આવે છે: {}

```
functionName(parameter1, parameter2, parameter3) { કોડ
અમલમાં મૂકવાનો છે
}
```

ફંક્શન પેરામીટર્સ ફંક્શન વ્યાખ્યામાં સૂચિબદ્ધ નામો છે. ફંક્શન દલીલો એ ફંક્શન દ્વારા પ્રાપ્ત વાસ્તવિક મૂલ્યો છે જ્યારે તેને બોલાવવામાં આવે છે. ફંક્શનની અંદર, દલીલોનો ઉપયોગ સ્થાનિક ચલો તરીકે થાય છે. કાર્ય પ્રક્રિયા અથવા સબરોટિન જેવું જ છે, અન્યમાં પ્રોગ્રામિંગ ભાષાઓ.

ફંક્શન ઈન્વોકેશન

ફંક્શનની અંદરનો કોડ જ્યારે “કંઈક” હશે ત્યારે એક્ટ્રિક્યુટ થશે ફંક્શનને બોલાવે છે (કોલ્સ).

- જ્યારે કોઈ ઇવેન્ટ થાય છે (જ્યારે વપરાશકર્તા બટન પર ક્લિક કરે છે)
- જ્યારે તેને JavaScript કોડમાંથી બોલાવવામાં આવે છે (કહેવાય છે).
- આપમેળે (સ્વયં બોલાવેલ)

ફંક્શન રીટર્ન

જ્યારે JavaScript રીટર્ન સ્ટેટમેન્ટ પર પહોંચે છે, ત્યારે ફંક્શન એક્ટ્રિક્યુટ કરવાનું બંધ કરશે. જો ફંક્શનને સ્ટેટમેન્ટમાંથી બોલાવવામાં આવ્યું હોય,

તો જાવાસ્ક્રિપ્ટ ઈન્વોકિંગ સ્ટેટમેન્ટ પછી કોડને એક્ટ્રિક્યુટ કરવા માટે “પાછું” આવશે. કાર્યો વારંવાર વળતર મૂલ્યની ગણતરી કરે છે. વળતર મૂલ્ય “કોલર” પર પાછું “પાછું” છે:

ઉદાહરણ 2

Calculate the product of two numbers, and return the result:

```
var x = myFunction(4, 3); // Function is called, return value
will end up in x
```

```
function myFunction(a, b) {
  return a * b; // Function returns the product of a and b
}
```

The result in x will be: 12

શા માટે કાર્યો?

તમે કોડનો ફરીથી ઉપયોગ કરી શકો છો: કોડને એકવાર વ્યાખ્યાયિત કરો અને ઘણી વખત તેનો ઉપયોગ કરો. તમે એક જ કોડનો ઉપયોગ વિવિધ દલીલો સાથે કરી શકો છો, વિવિધ પરિણામો ઉત્પન્ન કરવા માટે.

ઉદાહરણ 3

Convert Fahrenheit to Celsius:

```
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
```

JavaScript ફંક્શન ઓબ્જેક્ટ છે

JavaScript માં, ફંક્શન ઓબ્જેક્ટ છે. JavaScript ફંક્શનમાં ગુણધર્મો અને પદ્ધતિઓ હોય છે. તમે તમારા પોતાના ઉમેરી શકો છો ગુણધર્મો અને કાર્યોની પદ્ધતિઓ.

JavaScript ફંક્શન્સ પણ વેરિયેબલ છે

JavaScript માં, વિધેયોનો ચલ તરીકે ઉપયોગ કરી શકાય છે:

ઉદાહરણ 4

Instead of:

```
temp = toCelsius(32);
```

```
text = “The temperature is “ + temp + “ Centigrade”;
```

You can use:

```
text = “The temperature is “ + toCelsius(32) + “ Centigrade”;
```


JavaScript કાર્યોને સામાન્યની જેમ પુનઃવ્યાખ્યાયિત કરી શકાય છે ચલો તે અન્ય કાર્યોમાં મૂલ્યો તરીકે પણ પસાર કરી શકાય છે.

જાવાસ્ક્રિપ્ટ સ્કોપ

સ્કોપ એ ચલોનો સમૂહ છે જેની તમને ઍક્સેસ છે.

In JavaScript, ઑબ્જેક્ટ્સ અને ફંક્શન્સ પણ ચલો છે. માં JavaScript, સ્કોપ એ ચલો, ઑબ્જેક્ટ્સ અને તમને ઍક્સેસ હોય તેવા કાર્યો. JavaScript પાસે ફંક્શનનો અવકાશ છે: કાર્યની અંદર અવકાશ બદલાય છે.

સ્થાનિક JavaScript ચલો

જાવાસ્ક્રિપ્ટ ફંક્શનમાં જાહેર કરાયેલ ચલો, બની જાય છે કાર્ય માટે LOCAL. સ્થાનિક ચલોમાં સ્થાનિક અવકાશ હોય છે. તેઓ ફંક્શનમાં જ ઍક્સેસ કરી શકાય છે.

ઉદાહરણ 5

```
// code here can not use train
```

```
function myFunction() {
```

```
var train = "Jan Satabdi";
```

```
// code here can use train
```

```
}
```

કારણ કે સ્થાનિક ચલો ફક્ત તેમની અંદર જ ઑળખાય છે ફંક્શન્સ, સમાન નામ સાથેના ચલોનો ઉપયોગ કરી શકાય છે વિવિધ કાર્યો. સ્થાનિક ચલો બનાવવામાં આવે છે જ્યારે a ફંક્શન શરૂ થાય છે, અને જ્યારે ફંક્શન હોય ત્યારે કાઢી નાખવામાં આવે છે પૂર્ણ

વૈશ્વિક જાવાસ્ક્રિપ્ટ ચલો

ફંક્શનની બહાર જાહેર કરેલ ચલ, વૈશ્વિક બને છે. વૈશ્વિક ચલનો વૈશ્વિક અવકાશ છે: વેબ પૃષ્ઠ પરની બધી સ્ક્રિપ્ટો અને કાર્યો તેને ઍક્સેસ કરી શકે છે.

ઉદાહરણ 6

```
var train = "Jan Satabdi";
```

```
// code here can use train
```

```
function myFunction() {
```

```
// code here can use train
```

JavaScript માં સ્ટ્રિંગ અને નંબર પદ્ધતિઓ ()

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- શબ્દમાળા સમજાવો
- વિવિધ શબ્દમાળા પદ્ધતિઓ સમજાવો.

જાવાસ્ક્રિપ્ટ સ્ટ્રીંગ્સ

JavaScript શબ્દમાળાઓનો ઉપયોગ સંગ્રહ અને હેરફેર માટે થાય છે ટેક્સ્ટ JavaScript શબ્દમાળા શૂન્ય અથવા વધુ અક્ષરો લખેલી હોય છે અંદર અવતરણ. તમે સિંગલ અથવા ડબલ અવતરણનો ઉપયોગ કરી શકો છો.

ઉદાહરણ 1

```
var cityname = "Chennai"; // Double quotes
```

```
}
```

આપોઆપ વૈશ્વિક

જો તમે એવા ચલને મૂલ્ય અસાઈન કરો છો જે ન હોય જાહેર કર્યું, તે આપોઆપ વૈશ્વિક ચલ બની જશે.

આ કોડ ઉદાહરણ ટ્રેનને વૈશ્વિક ચલ તરીકે જાહેર કરશે, ભલે તે ફંક્શનની અંદર ઍક્ઝેક્યુટ કરવામાં આવે.

ઉદાહરણ 7

```
// code here can use train
```

```
function myFunction() {
```

```
train = "Jan Satabdi";
```

```
// code here can use train
```

```
}
```

JavaScript ચલોનો જીવનકાળ

JavaScript વેરીએબલનું જીવનકાળ શરૂ થાય છે જ્યારે તે જાહેર થાય છે. જ્યારે કાર્ય પૂર્ણ થાય છે ત્યારે સ્થાનિક ચલો કાઢી નાખવામાં આવે છે. જ્યારે તમે પૃષ્ઠ બંધ કરો છો ત્યારે વૈશ્વિક ચલો કાઢી નાખવામાં આવે છે.

કાર્ય દલીલો

ફંક્શન આર્ગ્યુમેન્ટ્સ (પેરામીટર્સ) ફંક્શનની અંદર સ્થાનિક ચલ તરીકે કામ કરે છે.

HTML માં વૈશ્વિક ચલો

JavaScript સાથે, વૈશ્વિક અવકાશ એ સંપૂર્ણ JavaScript પર્યાવરણ છે. HTML માં, વૈશ્વિક અવકાશ એ વિન્ડો ઑબ્જેક્ટ છે: બધા વૈશ્વિક ચલો વિન્ડો ઑબ્જેક્ટના છે.

ઉદાહરણ 8

```
// code here can use window.train
```

```
function myFunction() {
```

```
train = "Jan Satabdi";
```

```
}
```

```
var cityname = 'Chennai'; // Single quotes
```

You can use quotes inside a string, as long as they don't match the quotes surrounding the string.

ઉદાહરણ 2

```
var notes = "You're Welcome";
```

```
var ans = "Coimbatore is called 'Cotton City'";
```

var ans = 'Coimbatore is called "Cotton City";

String Lengthproperty

The length of a string is found in the built in property length.

ઉદાહરણ 3

var s = "Computer Operator and Programming Assistant";

var slen = s.length;

Backslash escape character.

The backslash (\) escape character turns special characters into string characters.

કોડ	પરિણામ	વર્ણન
\'	'	એક અવતરણ
\"	"	ડબલ ક્વોટ
\\	\	બેકસ્લેશ

ક્રમ \ " શબ્દમાળામાં ડબલ ક્વોટ દાખલ કરે છે.

ઉદાહરણ 4

var x = "Bhubaneswar is known as the \"Temple City of India\"";

The sequence \' inserts a single quote in a string:

ઉદાહરણ 5

var x = 'you\'re Welcome.';

The sequence \\ inserts a backslash in a string:

ઉદાહરણ 6

var x = "The character \\ is called backslash.";

શબ્દમાળાઓ વસ્તુઓ હોઈ શકે છે

સામાન્ય રીતે, JavaScript શબ્દમાળાઓ આદિમ મૂલ્યો છે, જે શાબ્દિક શબ્દોમાંથી બનાવવામાં આવે છે.

var tName = "વેની";

પરંતુ શબ્દમાળાઓને નવા કીવર્ડ સાથે ઓબ્જેક્ટ તરીકે પણ વ્યાખ્યાયિત કરી શકાય છે. var tName = new String("Veni");

JavaScript શબ્દમાળા પદ્ધતિઓ

શબ્દમાળા પદ્ધતિઓ તમને શબ્દમાળાઓ સાથે કામ કરવામાં મદદ કરે છે.

શબ્દમાળામાં શબ્દમાળા શોધવી

indexOf() પદ્ધતિ

indexOf() પદ્ધતિ સ્ટ્રિંગમાં ઉલ્લેખિત ટેક્સ્ટની પ્રથમ ઘટનાની અનુક્રમણિકા(સ્થિતિ) પરત કરે છે:

ઉદાહરણ 7

var str = "When I do good I feel good";

var num = str.indexOf("good");

પરિણામ:

ચલ સંખ્યા પોઝિશન મૂલ્ય 10 ધરાવે છે.

નોંધ: JavaScript શૂન્યમાંથી પોઝિશનની ગણતરી કરે છે. 0 એ સ્ટ્રિંગમાં પ્રથમ સ્થાન છે, 1 એ બીજું છે, 2 એ ત્રીજું છે ...

lastIndexOf() પદ્ધતિ

LastIndexOf() પદ્ધતિ સ્ટ્રિંગમાં ઉલ્લેખિત ટેક્સ્ટની છેલ્લી ઘટનાની અનુક્રમણિકા(સ્થિતિ) પરત કરે છે.

ઉદાહરણ 8

var str = "When I do good I feel good";

var num = str.lastIndexOf("good");

પરિણામ:

ચલ સંખ્યા 22 ની સ્થિતિ મૂલ્ય ધરાવે છે.

જો ટેક્સ્ટ ન મળે તો બંને indexOf(), અને.lastIndexOf() -1 પરત કરે છે.

ઉદાહરણ 9

var str = "When I do good I feel good";

var num = str.lastIndexOf("better");

પરિણામ:

ચલ સંખ્યા પોઝિશન મૂલ્ય -1 ધરાવે છે.

બંને પદ્ધતિઓ શોધ માટે પ્રારંભિક સ્થિતિ તરીકે બીજા પરિમાણને સ્વીકારે છે.

ઉદાહરણ 10

var str = "When I do good I feel good";

var num = str.indexOf("good",15);

પરિણામ:

ચલ સંખ્યા 22 ની સ્થિતિ મૂલ્ય ધરાવે છે.

શબ્દમાળામાં શબ્દમાળા શોધી રહ્યાં છીએ

શોધ() પદ્ધતિ ચોક્કસ મૂલ્ય માટે સ્ટ્રિંગ શોધે છે અને મેચની સ્થિતિ પરત કરે છે.

ઉદાહરણ 11

var str = "When I do good I feel good";

var num = str.search("good");

પરિણામ:

ચલ સંખ્યા પોઝિશન મૂલ્ય 10 ધરાવે છે.

નોંધ: શોધ() પદ્ધતિ એક સેકંડ લઈ શકતી નથી સ્થિતિ દલીલ શરૂ કરો.

શબ્દમાળા ભાગો કાઢવા

slice() પદ્ધતિ

slice() શબ્દમાળાનો એક ભાગ કાઢે છે અને કાઢેલ પરત કરે છે નવી સ્ટ્રિંગમાં ભાગ. પદ્ધતિ શરૂઆતની સ્થિતિ અને અંતની સ્થિતિ લે છે.

આ ઉદાહરણ પોઝિશન 14 થી પોઝિશન 21 સુધી સ્ટ્રિંગના એક ભાગને કાપી નાખે છે.

ઉદાહરણ 12

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.slice(14, 21);
```

પરિણામ:

Resનું પરિણામ ક્રિકેટમાં આવશે

જો પરિમાણ નકારાત્મક હોય, તો પોઝિશન સ્ટ્રિંગના અંતથી ગણવામાં આવે છે. આ ઉદાહરણ પોઝિશન -14 થી પોઝિશન -8 સુધી સ્ટ્રિંગના એક ભાગને કાપી નાખે છે.

ઉદાહરણ 13

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.slice(-14, -8);
```

પરિણામ:

રિઝલ્ટ કબાડી હશે

જો તમે બીજા પરિમાણને છોડી દો છો, તો પદ્ધતિ બાકીના સ્ટ્રિંગને કાપી નાખશે.

ઉદાહરણ 14

```
var res = str.slice(7);
```

પરિણામ:

રિઝલ્ટ કબાડી, ક્રિકેટમાં આવશે

સબસ્ટ્રિંગ() પદ્ધતિ

substring() સ્લાઈસ() જેવું જ છે. તફાવત એ છે કે substring() નકારાત્મક અનુક્રમણિકા સ્વીકારી શકતું નથી.

ઉદાહરણ 15

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.substring(7, 13);
```

પરિણામ:

રિઝલ્ટ કબાડી હશે

સબસ્ટ્રિંગ() પદ્ધતિ

જો તમે બીજા પેરામીટરને છોડી દો છો, તો substring() બાકીના સ્ટ્રિંગને કાપી નાખશે. substr() પદ્ધતિ substr() slice() જેવી જ છે. તફાવત એ છે કે બીજા પેરામીટર કાઢવામાં આવેલ ભાગની લંબાઈ સ્પષ્ટ કરે છે.

ઉદાહરણ 16

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.substr(7, 6);
```

પરિણામ:

રિઝલ્ટ કબાડી હશે

જો તમે બીજા પેરામીટરને છોડી દો છો, તો substr() સ્ટ્રિંગના બાકીના ભાગને કાપી નાખશે.

ઉદાહરણ 17

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.substr(7);
```

પરિણામ:

રેસનું પરિણામ કબાડી, ક્રિકેટ હશે

જો પ્રથમ પરિમાણ નકારાત્મક હોય, તો સ્થિતિ શબ્દમાળાના અંતથી ગણાય છે.

ઉદાહરણ 18

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.substr(-7);
```

પરિણામ:

રેસનું પરિણામ ક્રિકેટ હશે

શબ્દમાળા સામગ્રી બદલી રહ્યા છીએ

રિપ્લેસ() પદ્ધતિ સ્પષ્ટ કરેલ મૂલ્યને સાથે બદલે છે શબ્દમાળામાં અન્ય મૂલ્ય.

ઉદાહરણ 19

```
str = "I like custard apple";
```

```
var newstr = str.replace("custard apple", "mango");
```

પરિણામ:

Newsstr નું પરિણામ આવશે મને કેરી ગમે છે

નોંધ: રિપ્લેસ() પદ્ધતિ તે સ્ટ્રિંગને બદલી શકતી નથી જેના પર તેને કોલ કરવામાં આવે છે. તે નવી સ્ટ્રિંગ પરત કરે છે.

મૂળભૂત રીતે, રિપ્લેસ() ફંક્શન ફક્ત પ્રથમ મેચને બદલે છે.

મૂળભૂત રીતે, રિપ્લેસ() ફંક્શન કેસ સેન્સિટિવ છે. CUSTARD APPLE (અપર-કેસ સાથે) લખવું કામ કરશે નહીં.

અસંવેદનશીલ કેસને બદલવા માટે, /i ફ્લેગ (અસંવેદનશીલ) સાથે નિયમિત અભિવ્યક્તિનો ઉપયોગ કરો.

ઉદાહરણ 20

```
str = "I like custard apple";
```

```
var newstr = str.replace(/CUSTARD APPLE/i, "Mango");
```

પરિણામ:

Newsstr નું પરિણામ એ આવશે કે મને કેરી ગમે છે

નોંધ કરો કે નિયમિત અભિવ્યક્તિઓ અવતરણ વિના લખવામાં આવે છે.

તમામ મેચોને બદલવા માટે, a/g ફ્લેગ (ગ્લોબલ મેચ) સાથે રેગ્યુલર એક્સપ્રેશનનો ઉપયોગ કરો.

ઉદાહરણ 21

```
str = "I like custard apple and apple";
```

```
var newstr = str.replace(/apple/g, "Mango");
```

પરિણામ:

Newsstr નું પરિણામ એ આવશે કે મને કસ્ટર્ડ કેરી અને કેરી ગમે છે

અપર અને લોઅર કેસમાં કન્વર્ટ કરવું

સ્ટ્રિંગને toUpperCase() સાથે અપરકેસમાં રૂપાંતરિત કરવામાં આવે છે.

ઉદાહરણ 23

```
var str1 = "Information Technology";
```

```
var str2 = str1.toUpperCase();
```

પરિણામ:

str2 નું પરિણામ માહિતી ટેકનોલોજી હશે

concat() પદ્ધતિ

concat() બે અથવા વધુ શબ્દમાળાઓ જોડે છે.

ઉદાહરણ 24

```
var txt1 = "Mr";
```

```
var txt2 = "Selvaraj";
```

```
var txt3 = txt1.concat(".", txt2);
```

પરિણામ:

txt3 નું પરિણામ શ્રી સેલ્વરાજ હશે

પ્લસઓપરેટરને બદલે concat() પદ્ધતિનો ઉપયોગ કરી શકાય છે. આ બે લીટીઓ એ જ કરે છે.

ઉદાહરણ 25

```
var txt = "Mr" + "." + "Selvaraj";
```

```
var txt = "Mr".concat(".", "Selvaraj");
```

```
String.trim()
```

String.trim() સ્ટ્રિંગની બંને બાજુઓમાંથી વ્હાઈટસ્પેસ દૂર કરે છે.

ઉદાહરણ 26

```
var str = "India Gate";
```

```
alert(str.trim());
```

પરિણામ:

ચેતવણી બોક્સમાં આગળ અને પાછળની ખાલી જગ્યાઓ વગર ઇન્ડિયા ગેટ પ્રદર્શિત થાય છે.

શબ્દમાળા અક્ષરો કાઢવા

charAt() પદ્ધતિ સ્ટ્રિંગમાં ઉલ્લેખિત ઇન્ડેક્સ પર અક્ષર પરત કરે છે.

ઉદાહરણ 27

```
var str = "Thirumalai Nayakkar Mahal";
```

```
str.charAt(0); // returns T
```

charCodeAt() પદ્ધતિનો યુનિકોડ પરત કરે છે શબ્દમાળામાં ઉલ્લેખિત અનુક્રમણિકા પર અક્ષર. પદ્ધતિ 0 અને 65535 ની વચ્ચે UTF-16 શંકુ પૂર્ણાંક પરત કરે છે.

ઉદાહરણ 28

```
var str = "Hill Station";
```

```
str.charCodeAt(0); // returns 72
```

સ્ટ્રિંગને એરેમાં રૂપાંતરિત કરવું

split() પદ્ધતિ વડે સ્ટ્રિંગને એરેમાં રૂપાંતરિત કરી શકાય છે.

ઉદાહરણ 29

```
var txt1 = "Kovai,Nellai,Madurai"; // String
```

```
var txt2 = txt1.split(","); // Split on commas
```

પરિણામ:

txt2 નું પરિણામ Kovai હશે

જો વિભાજક અવગણવામાં આવે છે, તો પરત કરેલ એરે કરશે અનુક્રમણિકા [0] માં સંપૂર્ણ સ્ટ્રિંગ સમાવે છે. જો વિભાજક "" હોય, તો પરત કરેલ એરે એક અક્ષરોની એરે હશે.

ઉદાહરણ 30

```
var txt = "Anaimai"; // String
```

```
txt.split(""); // Split in characters
```

ઉદાહરણ 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>
```

પરિણામ:

T

e

m

p

l
e
?

JavaScript નંબર પદ્ધતિઓ

સંખ્યા પદ્ધતિઓ તમને સંખ્યાઓ સાથે કામ કરવામાં મદદ કરે છે.

સંખ્યા પદ્ધતિઓ અને ગુણધર્મો

2018 અથવા 1.44 જેવા આદિમ મૂલ્યોમાં ગુણધર્મો અને પદ્ધતિઓ હોઈ શકતી નથી કારણ કે તે ઓબ્જેક્ટ નથી.

પરંતુ JavaScript સાથે, પદ્ધતિઓ અને ગુણધર્મો પણ છે

આદિમ મૂલ્યો માટે ઉપલબ્ધ છે, કારણ કે JavaScript ટ્રીટ કરે છે

પદ્ધતિઓ અને ગુણધર્મો ચલાવતી વખતે ઓબ્જેક્ટ તરીકે આદિમ મૂલ્યો.

toString() પદ્ધતિ

toString() સંખ્યાને સ્ટ્રિંગ તરીકે પરત કરે છે. બધી સંખ્યા પદ્ધતિઓનો ઉપયોગ કોઈપણ પ્રકારની સંખ્યાઓ (શાબ્દિક, ચલ અથવા સમીકરણ) પર થઈ શકે છે.

ઉદાહરણ 32

```
var n = 2018;  
n.toString(); // returns 2018 from variable x  
(2018).toString(); // returns 2018 from literal 2018  
(2000+18).toString(); // returns 2018 from expression  
2000 + 18
```

ઘાતાંકીય() પદ્ધતિ

ઘાતાંકીય() ઘાતાંકીય સંકેતનો ઉપયોગ કરીને ગોળાકાર અને લખેલી સંખ્યા સાથે સ્ટ્રિંગ પરત કરે છે. એક પરિમાણ દશાંશ બિંદુની પાછળના અક્ષરોની સંખ્યાને વ્યાખ્યાયિત કરે છે.

ઉદાહરણ 33

```
var x = 3.869;  
x.toExponential(2); // returns 3.87e+0  
x.toExponential(4); // returns 3.8690e+0  
x.toExponential(6); // returns 3.869000e+0
```

પરિમાણ વૈકલ્પિક છે. જો તમે તેનો ઉલ્લેખ ન કરો, તો JavaScript નંબરને રાઉન્ડ કરશે નહીં.

toFixed() પદ્ધતિ

Fixed() માં ચોક્કસ દશાંશ સંખ્યા સાથે લખાયેલ સંખ્યા સાથે સ્ટ્રિંગ પરત કરે છે.

ઉદાહરણ 34

```
var x = 3.869;  
x.toFixed(0); // returns 4  
x.toFixed(2); // returns 3.87  
x.toFixed(4); // returns 3.8690
```

```
x.toFixed(6); // returns 3.869000
```

નોંધ: toFixed(2) પૈસા સાથે કામ કરવા માટે યોગ્ય છે.

to Precision() પદ્ધતિ

Precision() માં ચોક્કસ લંબાઈ સાથે લખાયેલ સંખ્યા સાથે સ્ટ્રિંગ પરત કરે છે.

ઉદાહરણ 35

```
var x = 3.869;  
x.toPrecision(); // returns 3.869  
x.toPrecision(2); // returns 3.9  
x.toPrecision(4); // returns 3.869  
x.toPrecision(6); // returns 3.86900
```

મૂલ્ય O f() પદ્ધતિ

મૂલ્ય O f() સંખ્યાને સંખ્યા તરીકે પરત કરે છે.

ઉદાહરણ 36

```
var x = 451;  
x.valueOf(); // returns 451 from variable x  
(451).valueOf(); // returns 451 from literal 451  
(400 + 51).valueOf(); // returns 451 from expression  
+ 51 400
```

ચલોને સંખ્યાઓમાં રૂપાંતરિત કરવું

ત્યાં 3 JavaScript પદ્ધતિઓ છે જેનો ઉપયોગ ચલોને નંબરોમાં કન્વર્ટ કરવા માટે કરી શકાય છે:

- 1 નંબર() પદ્ધતિ
- 2 parseInt() પદ્ધતિ
- 3 parseFloat() પદ્ધતિ

આ પદ્ધતિઓ સંખ્યા પદ્ધતિઓ નથી, પરંતુ વૈશ્વિક છે જાવાસ્ક્રિપ્ટ પદ્ધતિઓ.

વૈશ્વિક જાવાસ્ક્રિપ્ટ પદ્ધતિઓ

JavaScript વૈશ્વિક પદ્ધતિઓનો ઉપયોગ તમામ JavaScript ડેટા પ્રકારો પર થઈ શકે છે. સંખ્યાઓ સાથે કામ કરતી વખતે આ સૌથી સુસંગત પદ્ધતિઓ છે.

પદ્ધતિ	વર્ણન
Number()	તેની દલીલમાંથી રૂપાંતરિત, સંખ્યા પરત કરે છે.
parseInt ()	તેની દલીલનું વિશ્લેષણ કરે છે અને પૂર્ણાંક પરત કરે છે
parseFloat()	તેની દલીલનું વિશ્લેષણ કરે છે અને ફ્લોટિંગ પોઈન્ટ નંબર આપે છે

નંબર() પદ્ધતિ

Number() નો ઉપયોગ JavaScript ચલોને નંબરોમાં કન્વર્ટ કરવા માટે કરી શકાય છે.

ઉદાહરણ 37

```
Number(true);           // returns 1
Number(false);          // returns 0
Number("25");           // returns 25
Number(" 25");          // returns 25
Number("25 ");          // returns 25
Number("25.77");        // returns 25.77
Number("25,77");        // returns NaN
Number("2577");         // returns NaN
Number("ITI");          // returns NaN
```

નોંધ : જો નંબર કન્વર્ટ કરી શકાતો નથી, તો NaN (નંબર નથી) પરત કરવામાં આવે છે.

તારીખો પર વપરાયેલ નંબર() પદ્ધતિ. નંબર() તારીખને નંબરમાં પણ કન્વર્ટ કરી શકે છે:

ઉદાહરણ 38

```
નંબર(નવી તારીખ("2018-09-15")); // 1536969600000 પરત કરે છે
```

નોંધ: ઉપરની Number() પદ્ધતિ 1.1.1970 થી મિલિસેકન્ડની સંખ્યા પરત કરે છે.

parseFloat() પદ્ધતિ

parseFloat() શબ્દમાળાનું વિશ્લેષણ કરે છે અને સંપૂર્ણ સંખ્યા આપે છે. જગ્યાઓ માન્ય છે. માત્ર પ્રથમ નંબર જ પરત કરવામાં આવે છે..

ઉદાહરણ 39

```
parseFloat("25");           // returns 25
parseFloat("25.33");        // returns 25
parseFloat("25 20 30");     // returns 25
parseFloat("25 years");     // returns 25
parseFloat("years 25");     // returns NaN
```

જાતે અજમાવી જુઓ

જો નંબર કન્વર્ટ કરી શકાતો નથી, તો NaN (નંબર નથી) પરત કરવામાં આવે છે.

parseFloat() પદ્ધતિ

parseFloat() સ્ટ્રિંગને પાર્સ કરે છે અને નંબર પરત કરે છે. જગ્યાઓ માન્ય છે. ફક્ત પ્રથમ નંબર પરત કરવામાં આવે છે:

ઉદાહરણ 40

```
parseFloat("25");           // returns 25
parseFloat("25.77");        // returns 25.77
parseFloat("255075");       // returns 25
parseFloat("25 years");     // returns 25
```

```
parseFloat("years 25");     // returns NaN
```

નોંધ : જો નંબર કન્વર્ટ કરી શકાતો નથી, તો NaN (નંબર નથી) પરત કરવામાં આવે છે.

?

સંખ્યા ગુણધર્મો

મિલકત

વર્ણન

MIN_VALUE	JavaScript માં શક્ય તેટલી નાની સંખ્યા પરત કરે છે
MAX_VALUE	JavaScript માં શક્ય તેટલી સૌથી મોટી સંખ્યા પરત કરે છે
POSITIVE_INFINITY	અનંતનું પ્રતિનિધિત્વ કરે છે (ઓવરફ્લો પર પરત)
NEGATIVE_INFINITY	નકારાત્મક અનંતનું પ્રતિનિધિત્વ કરે છે (ઓવરફ્લો પર પરત)
NaN	"Not-a-Number" મૂલ્યનું પ્રતિનિધિત્વ કરે છે

JavaScript MIN_VALUE અને MAX_VALUE

ઉદાહરણ 41

```
var n = Number.MAX_VALUE;
```

પરિણામ:

MAX_VALUE એ સૌથી મોટી સંભવિત સંખ્યા પરત કરે છે જાવા સ્ક્રિપ્ટ.
1.7976931348623157e 308

ઉદાહરણ 42

```
var n = Number.MIN_VALUE;
```

પરિણામ:

MIN_VALUE શક્ય તેટલી નાની સંખ્યા પરત કરે છે જાવાસ્ક્રિપ્ટ.
5e-324

JavaScript POSITIVE_INFINITY

ઉદાહરણ 43

```
var n = Number.POSITIVE_INFINITY;
```

POSITIVE_INFINITY is returned on overflow.

ઉદાહરણ 44

```
var n = 2 / 0;
```

JavaScript NEGATIVE_INFINITY

ઉદાહરણ 45

```
var n = Number.NEGATIVE_INFINITY;
```

NEGATIVE_INFINITY is returned on overflow:

ઉદાહરણ 46

```
var x = -1 / 0;
```


JavaScript NaN - Not a Number

ઉદાહરણ 47

```
var x = Number.NaN;
```

NaN એ JavaScript આરક્ષિત શબ્દ છે જે દર્શાવે છે કે સંખ્યા એ કાયદેસરની સંખ્યા નથી. અંકગણિત નો અંકગણિત કરવાનો પ્રયાસ કરવો શબ્દમાળા NaN (નંબર નહીં) માં પરિણમશે.

ઉદાહરણ 48

```
var n = 500 / "Price"; // n will be NaN (Not a Number)
```

વેરિએબલ્સ પર નંબર પ્રોપર્ટીઝનો ઉપયોગ કરી શકાતો નથી

નંબર પ્રોપર્ટીઝ JavaScript ના નંબરની છે ઓબ્જેક્ટ રેપર જેને Number કહેવાય છે. આ ગુણધર્મો માત્ર Number.MAX_VALUE તરીકે જ એક્સેસ કરી શકાય છે.

newNumber.MAX_VALUE નો ઉપયોગ કરવો, જ્યાં 'newNumber' એ ચલ, અભિવ્યક્તિ અથવા મૂલ્ય છે, તે અવ્યાખ્યાયિત પરત આવશે.

ઉદાહરણ 49

```
var a = 10;
```

```
var b = a.MAX_VALUE; // b becomes undefined
```

જાવાસ્ક્રિપ્ટમાં ગણિતની વસ્તુઓ (Math objects in JavaScript)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- જાવા સ્ક્રિપ્ટમાં ગણિતની વસ્તુઓ સમજાવો.

Java Script ગણિત ઓબ્જેક્ટ

JavaScript Math ઓબ્જેક્ટ તમને કરવા માટે પરવાનગી આપે છે

સંખ્યાઓ પર ગાણિતિક કાર્યો.

```
Math.PI; // returns 3.141592653589793
```

Math.round(x) તેના નજીકના પૂર્ણાંકમાં x ગોળાકારની કિંમત પરત કરે છે:

ઉદાહરણ 1

```
Math.round(5.8); // returns 6
```

```
Math.round(5.4); // returns 5
```

Math.pow(x,y) returns the value of x to the power of y:

ઉદાહરણ 2

```
Math.pow(5,2); // returns 25
```

Math.sqrt(x) returns the square root of x:

ઉદાહરણ 3

```
Math.sqrt(25); // returns 5
```

Math.abs(x) returns the absolute (positive) value of x:

ઉદાહરણ 4

```
Math.abs(-3.5); // returns 3.5
```

Math.ceil(x) returns the value of x rounded up to its nearest integer:

ઉદાહરણ 5

```
Math.ceil(5.4); // returns 6
```

Math.floor(x) returns the value of x rounded down to its nearest integer:

ઉદાહરણ 6

```
Math.floor(5.8); // returns 5
```

Math.sin(x) returns the sine (a value between -1 and 1) of the angle x (given in radians).

If you want to use degrees instead of radians, you have to convert degrees to radians.

Angle in radians = Angle in degrees x PI / 180.

ઉદાહરણ 7

```
Math.sin(90 * Math.PI / 180); // returns 1 (the sine of 90 degrees)
```

```
Math.sin(0 * Math.PI / 180); // returns 0 (the sine of 0 degrees)
```

Math.cos(x) returns the cosine (a value between -1 and 1) of the angle x (given in radians).

If you want to use degrees instead of radians, you have to convert degrees to radians:

Angle in radians = Angle in degrees x PI / 180.

ઉદાહરણ 8

```
Math.cos(0 * Math.PI / 180); // returns 1 (the cos of 0 degrees)
```

Math.min() and Math.max() can be used to find the lowest or highest value in a list of arguments:

ઉદાહરણ 9

```
Math.min(20,40,6,0,-10); // returns -10
```

ઉદાહરણ 10

```
Math.max(20,40,6,0,-10); // returns 40
```

Math.random() returns a random number between 0 (inclusive), and 1 (exclusive):

ઉદાહરણ 11

```
Math.random(); // returns a random number
```

ગણિત ગુણધર્મો (અચલ)

JavaScript 8 ગાણિતિક સ્થિરાંકો પ્રદાન કરે છે જે ગણિત ઓબ્જેક્ટ સાથે એક્સેસ કરી શકાય છે:

ઉદાહરણ 12

Math.E // returns Euler's number

Math.PI // returns PI

Math.SQRT2 // returns the square root of 2

Math.SQRT1_2 // returns the square root of 1/2

Math.LN2 // returns the natural logarithm of 2

Math.LN10 // returns the natural logarithm of 10

Math.LOG2E // returns base 2 logarithm of E

Math.LOG10E // returns base 10 logarithm of E

ગણિત કન્સ્ટ્રક્ટર

અન્ય વૈશ્વિક ઓબ્જેક્ટ્સથી વિપરીત, ગણિત ઓબ્જેક્ટ પાસે નં કન્સ્ટ્રક્ટર પદ્ધતિઓ અને ગુણધર્મો સ્થિર છે. તમામ પદ્ધતિઓ અને ગુણધર્મો (અચલ) નો ઉપયોગ પ્રથમ ગણિત ઓબ્જેક્ટ બનાવ્યા વિના કરી શકાય છે.

ગણિત ઓબ્જેક્ટ પદ્ધતિઓ

પદ્ધતિ	વર્ણન
abs(x)	x નું સંપૂર્ણ મૂલ્ય પરત કરે છે
acos(x)	x નું આર્કોસિન, રેડિયનમાં પરત કરે છે
asin(x)	x નું આર્કસાઈન રેડિયનમાં પરત કરે છે
atan(x)	PI/2 અને PI/2 રેડિયન વચ્ચેના આંકડાકીય મૂલ્ય તરીકે x ના આર્કટેન્જન્ટને પરત કરે છે
atan2(y, x)	તેની દલીલોના ભાગલાકારનો આર્કટેન્જન્ટ પરત કરે છે
ceil(x)	x નું મૂલ્ય તેના નજીકના પૂર્ણાંક સુધી ગોળાકાર પરત કરે છે
cos(x)	x નું કોસાઈન પરત કરે છે (x રેડિયનમાં છે)
exp(x)	Ex નું મૂલ્ય પરત કરે છે
floor(x)	x નું મૂલ્ય તેના નજીકના પૂર્ણાંક પર ગોળાકાર નીચે આપે છે
log(x)	x નો કુદરતી લઘુગણક (બેઝ E) પરત કરે છે
max(x, y, z, ..., n)	ઉચ્ચતમ મૂલ્ય સાથેની સંખ્યા પરત કરે છે
min(x, y, z, ..., n)	સૌથી ઓછા મૂલ્ય સાથેની સંખ્યા પરત કરે છે
pow(x, y)	x ની કિંમત y ની ઘાત પર આપે છે
random()	0 અને 1 વચ્ચેની રેન્ડમ સંખ્યા પરત કરે છે
round(x)	x નું મૂલ્ય તેના નજીકના પૂર્ણાંક પર ગોળાકાર પરત કરે છે
sin(x)	x ની સાઈન પરત કરે છે (x રેડિયનમાં છે)
sqrt(x)	x નું વર્ગમૂળ પરત કરે છે
tan(x)	ખૂણાની સ્પર્શક પરત કરે છે

JavaScript તારીખો (JavaScript Dates)

ઉદ્દેશ્ય : આ પાઠના અંતે તમે સક્ષમ થશો

- JavaScript તારીખ ઓબ્જેક્ટ્સ સમજાવો
- JavaScript તારીખ ફોર્મેટ સમજાવો
- JavaScript તારીખ મેળવો પદ્ધતિઓ સમજાવો
- JavaScript તારીખ સેટ પદ્ધતિઓ સમજાવો.

JavaScript તારીખ ઓબ્જેક્ટ્સ

ડિફોલ્ટ રૂપે, JavaScript બ્રાઉઝરના ટાઈમ ઝોનનો ઉપયોગ કરશે અને સંપૂર્ણ ટેક્સ્ટ સ્ટ્રિંગ તરીકે તારીખ પ્રદર્શિત કરશે.

ગુરુ 27 સપ્ટેમ્બર 2018 09:09:39 GMT 0530 (ભારત માનક સમય)

તારીખ ઓબ્જેક્ટ્સ બનાવી રહ્યા છીએ

તારીખ ઓબ્જેક્ટ નવા Date() કન્સ્ટ્રક્ટર સાથે બનાવવામાં આવે છે.

નવી તારીખ ઓબ્જેક્ટ બનાવવાની 4 રીતો છે. તેઓ છે

- 1 નવી તારીખ()
- 2 નવી તારીખ (વર્ષ, મહિનો, દિવસ, કલાક, મિનિટ, સેકન્ડ, મિલિસેકન્ડ)
- 3 નવી તારીખ (તારીખ શબ્દમાળા)
- 4 નવી તારીખ (મિલિસેકન્ડ)

1 નવી તારીખ()

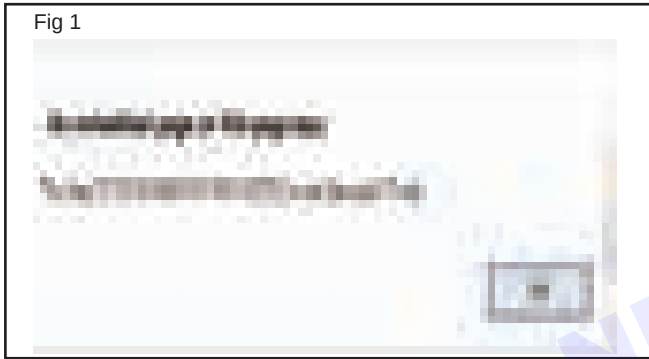
નવી તારીખ() વર્તમાન તારીખ અને સમય સાથે નવી તારીખ ઓબ્જેક્ટ બનાવે છે.

ઉદાહરણ 1:

```
var d = new Date();
```

```
alert(d);
```

પરિણામ: (ફિગ 1)



નોંધ તારીખ ઓબ્જેક્ટ સ્થિર છે. કમ્પ્યુટરનો સમય ઘબકતો હોય છે, પરંતુ તારીખની વસ્તુઓ નથી.

2 નવી તારીખ (વર્ષ, મહિનો, ...)

નવી તારીખ (વર્ષ, મહિનો, ...) ચોક્કસ તારીખ અને સમય સાથે નવી તારીખ ઓબ્જેક્ટ બનાવે છે.

ઉદાહરણ 2

```
var d = new Date(2018, 06, 30, 09, 10, 25, 0);
```

7 numbers specify year, month, day, hour, minute, second, and millisecond (in that order).

પરિણામ: (ફિગ 2)



નોંધ: JavaScript 0 થી 11 મહિનાની ગણતરી કરે છે. જાન્યુઆરી 0 છે. ડિસેમ્બર 11 છે.

6 નંબરો વર્ષ, મહિનો, દિવસ, કલાક, મિનિટ, સેકન્ડનો ઉલ્લેખ કરે છે.

ઉદાહરણ 3

```
var d = new Date(2018, 06, 30, 09, 10, 25);
```

5 numbers specify year, month, day, hour, and minute.

ઉદાહરણ 4

```
var d = new Date(2018, 06, 30, 09, 10);
```

ઉદાહરણ 5

```
var d = new Date(2018, 06, 30, 09);
```

3 numbers specify year, month, and day.

ઉદાહરણ 6

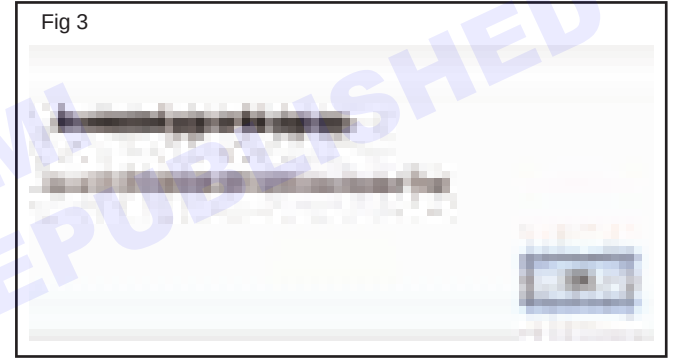
```
var d = new Date(2018, 06, 30);
```

2 numbers specify year and month.

ઉદાહરણ 7

```
var d = new Date(2018, 06);
```

પરિણામ: (ફિગ 3)



તમે મહિનો છોડી શકતા નથી. જો તમે માત્ર એક પરિમાણ સપ્લાય કરો છો તેને મિલિસેકન્ડ તરીકે ગણવામાં આવશે.

ઉદાહરણ 8

```
var d = new Date(2018);
```

Previous Century

One and two digit years will be interpreted as 19xx.

ઉદાહરણ 9

```
var d = new Date(96, 04, 12);
```

3 નવી તારીખ(dateString)

new Date(dateString) તારીખ સ્ટ્રિંગમાંથી નવી તારીખ ઓબ્જેક્ટ બનાવે છે.

ઉદાહરણ 10

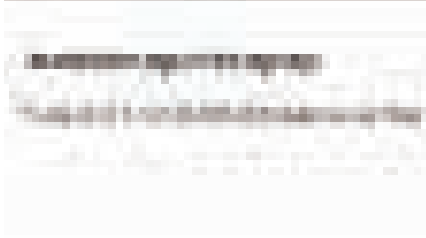
```
var d = new Date("December 20, 2018 10:15:00");
```

પરિણામ: (ફિગ 4)

નોંધ: JavaScript જાન્યુઆરી 01, 1970, 00:00:00UTC (યુનિવર્સલ ટાઈમ કોઓર્ડિનેટેડ) થી તારીખોને મિલિસેકન્ડની

સંખ્યા તરીકે સંગ્રહિત કરે છે. શૂન્ય સમય જાન્યુઆરી 01, 1970 00:00:00 UTC છે. હવે સમય છે: 1537962903199 મિલિસેકન્ડ ભૂતકાળ જાન્યુઆરી 01, 1970

Fig 4



4 નવી તારીખ (મિલિસેકન્ડ)

નવી તારીખ(મિલિસેકન્ડ) ઝીરોટાઈમ વત્તા મિલિસેકન્ડ તરીકે નવી તારીખ ઓબ્જેક્ટ બનાવે છે.

ઉદાહરણ 11

```
var d = new Date(0);
```

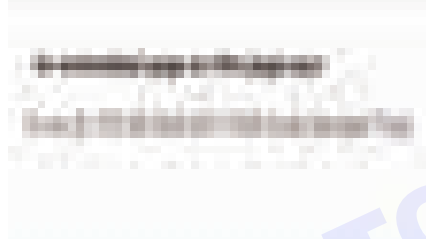
01 January 1970 plus 100 000 000 milliseconds is approximately 02 January 1970.

ઉદાહરણ 12

```
var d = new Date(1000000000);
```

પરિણામ: (ફિગ 5)

Fig 5



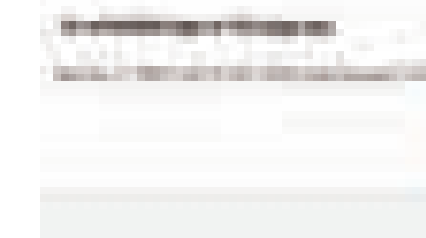
જાન્યુઆરી 01 1970 ઓછા 100 000 000 મિલિસેકન્ડ છે લગભગ 31 ડિસેમ્બર 1969.

ઉદાહરણ 13

```
var d = new Date(-1000000000);
```

પરિણામ: (ફિગ 6)

Fig 6



ઉદાહરણ 14

```
var d = new Date(864000000);
```

પરિણામ:

શુક્ર જાન્યુઆરી 02 1970 05:30:00 GMT 0530 (ભારત માનક સમય)

નોંધ: નવી તારીખ (મિલિસેકન્ડ) નો ઉપયોગ કરીને, નવી તારીખ બનાવે છેઑબ્જેક્ટ 1 જાન્યુઆરી, 1970, 00:00:00 યુનિવર્સલ ટાઈમ (UTC) વત્તા મિલિસેકન્ડ. એક દિવસ (24 કલાક) 86 400 000 મિલિસેકન્ડ છે.

તારીખ પદ્ધતિઓ

જ્યારે તારીખ ઑબ્જેક્ટ બનાવવામાં આવે છે, ત્યારે સંખ્યાબંધ પદ્ધતિઓ તમને તેના પર કાર્ય કરવાની મંજૂરી આપે છે. તારીખ પદ્ધતિઓ તમને સ્થાનિક સમય અથવા UTC (યુનિવર્સલ અથવા GMT) સમયનો ઉપયોગકરીનેવર્ષ,મહિનો,દિવસ,કલાક,મિનિટ,સેકન્ડઅનેમિલિસેકન્ડતારીખના ઑબ્જેક્ટ મેળવવા અને સેટ કરવાની મંજૂરી આપે છે.

તારીખો દર્શાવે છે

ડિફોલ્ટ રૂપે JavaScript સંપૂર્ણ ટેક્સ્ટ સ્ટ્રિંગમાં તારીખો આઉટપુટ કરશેફોર્મેટ જ્યારે તમે HTML માં તારીખ ઑબ્જેક્ટ પ્રદર્શિત કરો છો, ત્યારે તે છેસ્ટ્રિંગ() પદ્ધતિ સાથે આપોઆપ સ્ટ્રિંગમાં રૂપાંતરિત થાય છે.

ઉદાહરણ 15

```
d = new Date();
```

```
alert(d);
```

Same as:

```
d = new Date();
```

```
alert(d.toString());
```

The toUTCString() method converts a date to a UTC string.

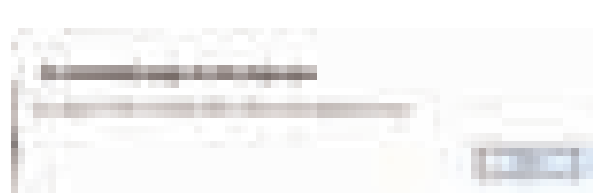
ઉદાહરણ 16

```
var d = new Date();
```

```
alert(d);
```

પરિણામ: (ફિગ 7)

Fig 7



toDateString() પદ્ધતિ તારીખને વધુ વાંચી શકાય તેવા ફોર્મેટમાં રૂપાંતરિત કરે છે.

ઉદાહરણ 17

```
var d = new Date();
```

```
alert(d.toDateString());
```

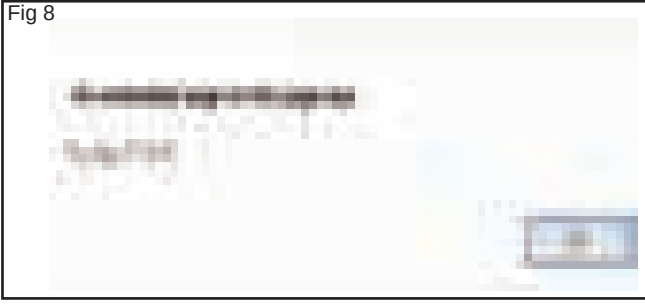
પરિણામ: (ફિગ 8)

JavaScript તારીખ ફોર્મેટ્સ

JavaScript તારીખ ઈનપુટ

જાવાસ્ક્રિપ્ટ તારીખ ઈનપુટ સામાન્ય રીતે 3 પ્રકારના હોય છેબધારણો

Fig 8



ઉદાહરણ	પ્રકાર
ISO તારીખ	“2002-06-30” (ઇન્ટરનેશનલ સ્ટેન ઈઈ)
ટૂંકી તારીખ	“06/30/2002”
લાંબી તારીખ	“જૂન 30 2002” અથવા “30 જૂન 2002”

JavaScript તારીખ આઉટપુટ

ઇનપુટ ફોર્મેટથી સ્વતંત્ર, JavaScript પૂર્ણ ટેક્સ્ટ સ્ટ્રિંગ ફોર્મેટમાં તારીખો આઉટપુટ કરશે.

JavaScript ISO તારીખો

ISO 8601 માટે આંતરરાષ્ટ્રીય ધોરણ છે તારીખો અને સમયનું પ્રતિનિધિત્વ. ISO 8601 સિન્ટેક્સ (YYYY-MM-DD) પણ પસંદગીનું JavaScript તારીખ ફોર્મેટ છે.

ઉદાહરણ 18 (સંપૂર્ણ તારીખ)

```
var d = new Date("2002-06-30");
```

નોંધ: ગણતરી કરેલ તારીખ તમારા સમય ઝોન સાથે સંબંધિત હશે. તમારા સમય ઝોનના આધારે, ઉપરોક્ત પરિણામ 29 જૂન અને 30 જૂનની વચ્ચે બદલાશે.

ISO તારીખો (વર્ષ અને મહિનો)

ISO તારીખો દિવસ (YYYYMM) નો ઉલ્લેખ કર્યા વિના લખી શકાય છે.

ઉદાહરણ 19

```
var d = new Date("2002-06");
```

પરિણામ:

શનિ જૂન 01 2002 05:30:00 GMT 0530 (ભારત માનક સમય)

ISO તારીખો (માત્ર વર્ષ)

ISO તારીખો મહિનો અને દિવસ (YYYY) વગર લખી શકાય છે.

ઉદાહરણ 20

```
var d = new Date("2011");
```

પરિણામ:

શનિ 01 જાન્યુઆરી 2011 05:30:00 GMT 0530 (ભારત માનક સમય)

ISO તારીખો (તારીખ-સમય)

ISO તારીખો ઉમેરવામાં આવેલા કલાકો, મિનિટો અને સેકન્ડ સાથે લખી શકાય છે (YYYY-MM-DDTHH:MM:SSZ)

ઉદાહરણ 21

```
var d = new Date("2011-12-20T12:00:00Z");
```

પરિણામ:

મંગળ 20 ડિસેમ્બર 2011 17:30:00 GMT 0530 (ભારત માનક સમય)

નોંધ: તારીખ અને સમયને કેપિટલ T સાથે અલગ કરવામાં આવ્યો છે. UTC સમયને મોટા અક્ષર Z સાથે વ્યાખ્યાયિત કરવામાં આવ્યો છે.

JavaScript ટૂંકી તારીખો.

ટૂંકી તારીખો આ રીતે “MM/DD/YYYY” સિન્ટેક્સ સાથે લખવામાં આવે છે.

ઉદાહરણ 22

```
var d = new Date("06/30/2002");
```

પરિણામ:

રવિ જૂન 30 2002 00:00:00 GMT 0530 (ભારત માનક સમય)

નોંધ: કેટલાક બ્રાઉઝર્સમાં, કોઈ અગ્રણી શૂન્ય વગરના મહિનાઓ અથવા દિવસો ભૂલ પેદા કરી શકે છે. “YYYY/MM/DD” નું વર્તન અવ્યાખ્યાયિત છે. કેટલાક બ્રાઉઝર્સ ફોર્મેટનું અનુમાન લગાવવાનો પ્રયાસ કરશે. કેટલાક NaN પરત કરશે. “DD-MM-YYYY” નું વર્તન પણ અવ્યાખ્યાયિત છે. કેટલાક બ્રાઉઝર્સ ફોર્મેટનું અનુમાન લગાવવાનો પ્રયાસ કરશે. કેટલાક NaN પરત કરશે.

JavaScript લાંબી તારીખો

લાંબી તારીખો મોટે ભાગે આના જેવા “MMM DD YYYY” સિન્ટેક્સ સાથે લખવામાં આવે છે.

ઉદાહરણ 23

```
var d = new Date("Aug 31 2012");
```

Month and day can be in any order.

ઉદાહરણ 24

```
var d = new Date("31 Aug 2012");
```

And, month can be written in full (January), or abbreviated (Jan).

ઉદાહરણ 25

```
var d = new Date("August 31 2012");
```

```
var d = new Date("AUGUST 31 2012");
```

નોંધ: અલ્પવિરામ અવગણવામાં આવે છે. નામો કેસ અસંવેદનશીલ છે.

તારીખ ઇનપુટ - તારીખોનું પદચ્છેદન

જો તમારી પાસે માન્ય તારીખ શબ્દમાળા હોય, તો તમે Date.parse() પદ્ધતિનો ઉપયોગ તેને મિલિસેકન્ડ્સમાં રૂપાંતરિત કરવા માટે કરી શકો છો. Date.parse() તારીખ અને જાન્યુઆરી 1, 1970 વચ્ચેની મિલિસેકન્ડની સંખ્યા પરત કરે છે.

ઉદાહરણ 26

```
var msec = Date.parse("Sep 15, 1996");  
document.getElementById("demo").innerHTML = msec;
```

પરિણામ:

842725800000

પછી તમે તેને તારીખ ઓબ્જેક્ટમાં કન્વર્ટ કરવા માટે મિલિસેકન્ડ્સની સંખ્યાનો ઉપયોગ કરી શકો છો.

ઉદાહરણ 27

```
var msec = Date.parse("Sep 15, 1996");  
var d = new Date(msec);  
document.getElementById("demo").innerHTML = d;
```

પરિણામ:

રવિ સપ્ટે 15 1996 00:00:00 GMT 0530 (ભારત માનક સમય)

JavaScript તારીખ પદ્ધતિઓ મેળવો

તારીખ ઓબ્જેક્ટમાંથી માહિતી મેળવવા માટે આ પદ્ધતિઓનો ઉપયોગ કરી શકાય છે.

પદ્ધતિ	વર્ણન
સંપૂર્ણ વર્ષ મેળવો()	ચાર અંકની સંખ્યા તરીકે વર્ષ મેળવો (yyyy)
મહિનો મેળવો()	નંબર તરીકે મહિનો મેળવો (0-11)
તારીખ મેળવો()	દિવસને નંબર તરીકે મેળવો (1-31)
ગ્રે કલાક()	કલાક મેળવો (0-23)
મિનિટ મેળવો()	મિનિટ મેળવો (0-59)
સેકન્ડ મેળવો()	બીજો મેળવો (0-59)
મિલિસેકન્ડ્સ મેળવો()	મિલિસેકન્ડ મેળવો (0-999)
સમય મેળવો()	સમય મેળવો (1 જાન્યુઆરી, 1970 થી મિલિસેકન્ડ)
દિવસ મેળવો()	અઠવાડિયાનો દિવસ નંબર તરીકે મેળવો (0-6)
તારીખ. હવે()	સમય મેળવો. ECMAScript 5.

getTime() પદ્ધતિ

getTime() પદ્ધતિ જાન્યુઆરી 1, 1970 થી મિલિસેકન્ડની સંખ્યા પરત કરે છે.

ઉદાહરણ 28

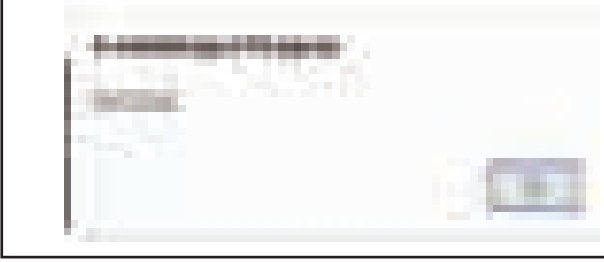
```
var d = new Date();  
alert(d.getTime());
```

પરિણામ: (ફિગ 9)

getFullYear() પદ્ધતિ

getFullYear() પદ્ધતિ ચાર અંકની સંખ્યા તરીકે તારીખનું વર્ષ પરત કરે છે.

Fig 9

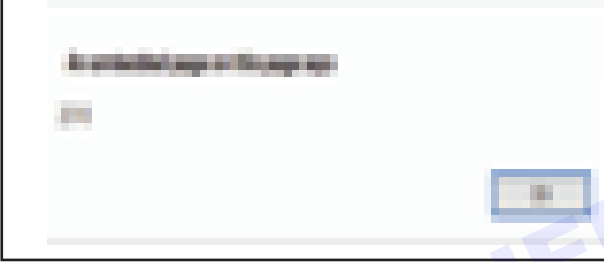


ઉદાહરણ 29

```
var d = new Date();  
alert(d.getFullYear());
```

પરિણામ: (ફિગ 10)

Fig 10



getMonth() પદ્ધતિ

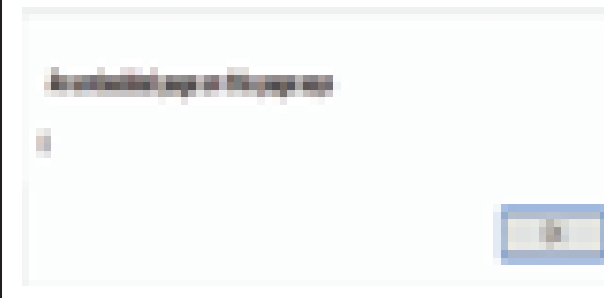
getMonth() પદ્ધતિ નંબર (0-11) તરીકે તારીખનો મહિનો પરત કરે છે.

ઉદાહરણ 30

```
var d = new Date();  
document.getElementById("demo").innerHTML =  
d.getMonth();
```

પરિણામ: (ફિગ 11)

Fig 11



JavaScript માં, પ્રથમ મહિનો (જાન્યુઆરી) મહિનો નંબર 0 છે, તેથી ડિસેમ્બર મહિનો નંબર 11 પરત કરે છે.

તમે નામની શ્રેણીનો ઉપયોગ કરી શકો છો, અને નામ તરીકે મહિનો પરત કરવા getMonth() નો ઉપયોગ કરી શકો છો.

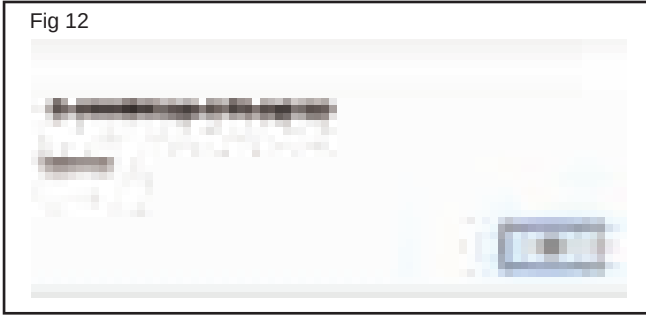
ઉદાહરણ 31

```
var d = new Date();  
var months = ["January", "February", "March", "April",  
"May", "June", "July", "August", "September", "October",  
"November", "December"];
```



```
document.getElementById("demo").innerHTML  
months[d.getMonth()];
```

પરિણામ: (ફિગ 12)



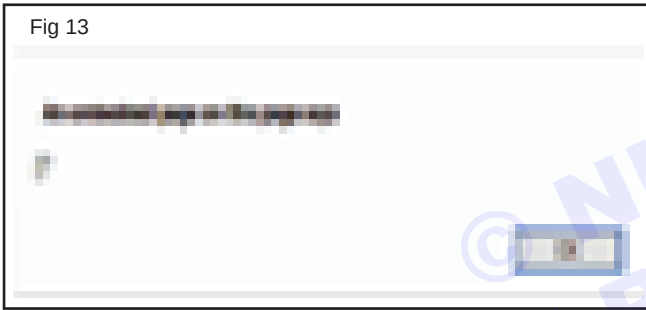
getDate() પદ્ધતિ

getDate() પદ્ધતિ નંબર (1-31) તરીકે તારીખનો દિવસ પરત કરે છે.

ઉદાહરણ 32

```
var d = new Date();  
alert(d.getDate());
```

પરિણામ: (ફિગ 13)



getHours() પદ્ધતિ

getHours() પદ્ધતિ નંબર (0-23) તરીકે તારીખના કલાકો પરત કરે છે.

ઉદાહરણ 33

```
var d = new Date();  
alert(d.getHours());
```

પરિણામ: (ફિગ 14)



getMinutes() પદ્ધતિ

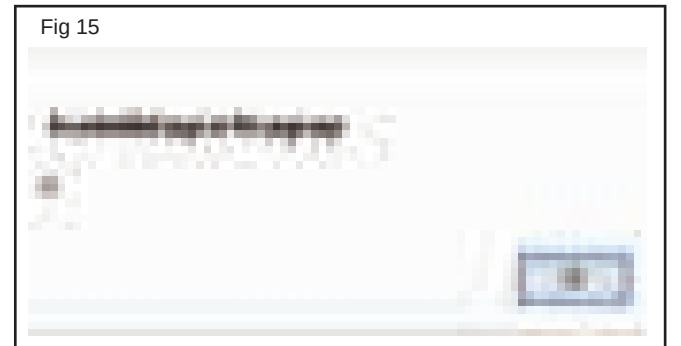
getMinutes() પદ્ધતિ નંબર (0-59) તરીકે તારીખની મિનિટ પરત કરે છે.

ઉદાહરણ 34

```
var d = new Date();
```

```
alert(d.getMinutes());
```

પરિણામ: (ફિગ 15)



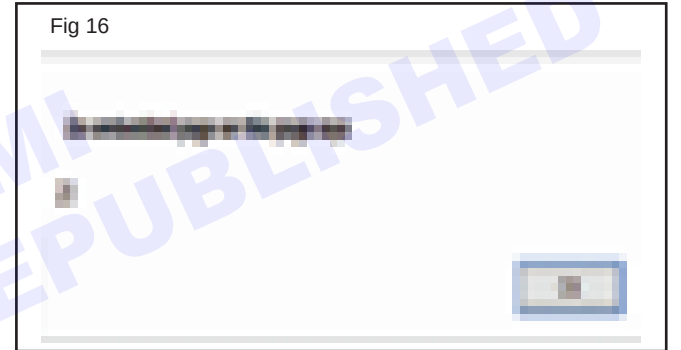
getSeconds() પદ્ધતિ

getSeconds() પદ્ધતિ નંબર (0-59) તરીકે તારીખની સેકન્ડ પરત કરે છે.

ઉદાહરણ 35

```
var d = new Date();  
alert(d.getSeconds());
```

પરિણામ: (ફિગ 16)



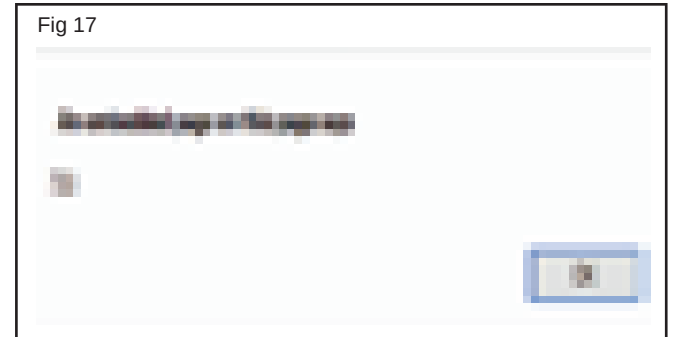
getMilliseconds() પદ્ધતિ

getMilliseconds() પદ્ધતિ નંબર (0-999) તરીકે તારીખના મિલિસેકન્ડ પરત કરે છે.

ઉદાહરણ 36

```
var d = new Date();  
alert(d.getMilliseconds());
```

પરિણામ: (ફિગ 17)



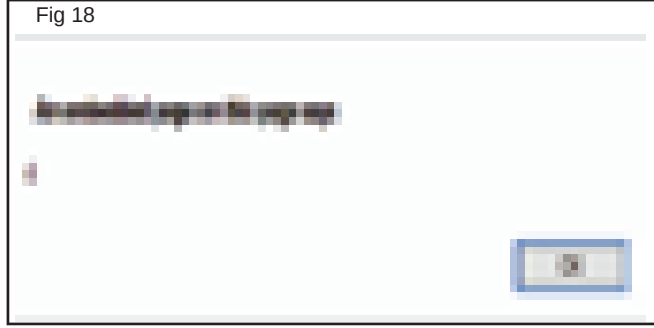
getDay() પદ્ધતિ

getDay() પદ્ધતિ નંબર (0-6) તરીકે તારીખનો સપ્તાહનો દિવસ પરત કરે છે.

ઉદાહરણ 37

```
var d = new Date();
alert(d.getDay());
```

પરિણામ: (ફિગ 18)



જાવાસ્ક્રિપ્ટમાં, અઠવાડિયાના પ્રથમ દિવસ (0) નો અર્થ “રવિવાર” થાય છે, ભલે વિશ્વના કેટલાક દેશો અઠવાડિયાના પ્રથમ દિવસને “સોમવાર” ગણતા હોય.

તમે નામોની શ્રેણીનો ઉપયોગ કરી શકો છો, અને અઠવાડિયાના દિવસને નામ તરીકે પરત કરવા માટે `getDay()`નો ઉપયોગ કરી શકો છો.

ઉદાહરણ 38

```
var d = new Date();
var days = ["Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"];
alert(days[d.getDay()]);
```

પરિણામ: (ફિગ 19)



UTC તારીખ પદ્ધતિઓ

UTC તારીખ પદ્ધતિઓનો ઉપયોગ UTC તારીખો સાથે કામ કરવા માટે થાય છે(યુનિવર્સલ ટાઈમ ઝોન તારીખો).

પદ્ધતિ	વર્ણન
UTCDate() મેળવો	get Date() જેવું જ છે, પરંતુ UTC તારીખ પરત કરે છે
UTC દિવસ મેળવો ()	Get Day() ની જેમ જ, પરંતુ UTC દિવસ પરત કરે છે
સંપૂર્ણ વર્ષ () UTC મેળવો	પૂર્ણ વર્ષ () મેળવો સમાન છે, પરંતુ UTC વર્ષ પરત કરે છે
UTC કલાક મેળવો()	get Hours() જેવું જ છે, પરંતુ UTC કલાક પરત કરે છે

UTC મિલિસેકન્ડ્સ મેળવો()	getMilliseconds(), પરંતુ UTC મિલિસેકન્ડ્સ પરત કરે છે
UTC મિનિટ() મેળવો	get Minutes() ની જેમ જ, પરંતુ UTC મિનિટ ફરી વળે છે
UTC મહિનો મેળવો()	get Month(), પરંતુ વળતરની જેમ જ UTC મહિનો
UTC સેકન્ડ્સ મેળવો()	get Seconds() ની જેમ જ, પરંતુ UTC સેકન્ડ ફરી વળે છે

JavaScript સેટ તારીખ પદ્ધતિઓ

તારીખ સેટ કરવાની પદ્ધતિઓ તમને તારીખ ઓબ્જેક્ટ માટે તારીખ મૂલ્યો (વર્ષ, મહિના, દિવસો, કલાકો, મિનિટ, સેકન્ડ, મિલિસેકન્ડ) સેટ કરવા દે છે.

તારીખ પદ્ધતિઓ સેટ કરો

તારીખનો ભાગ સેટ કરવા માટે સેટ ડેટ પદ્ધતિઓનો ઉપયોગ કરવામાં આવે છે.

પદ્ધતિ	વર્ણન
તારીખ સેટ કરો()	દિવસને નંબર તરીકે સેટ કરો (1-31)
સંપૂર્ણ વર્ષ સેટ કરો()	વર્ષ સેટ કરો (વૈકલ્પિક રીતે મહિનો અને દિવસ)
સમય સેટ કરો()	કલાક સેટ કરો (0-23)
મિલી સેકન્ડ() સેટ કરો	મિલિસેકન્ડ સેટ કરો (0-999)
મિનિટ સેટ કરો()	મિનિટ સેટ કરો (0-59)
મહિનો સેટ કરો()	મહિનો સેટ કરો (0-11)
સેકન્ડ સેટ કરો()	સેકન્ડ સેટ કરો (0-59)
સમય ગોઠવવો()	સમય સેટ કરો (1 જાન્યુઆરી, 1970 થી મિલિસેકન્ડ)

The setFullYear() Method

The setFullYear() method sets the year of a date object. In this example to 2020.

ઉદાહરણ 39

```
<script>
var d = new Date();
d.setFullYear(2020);
alert(d);
</script>
```

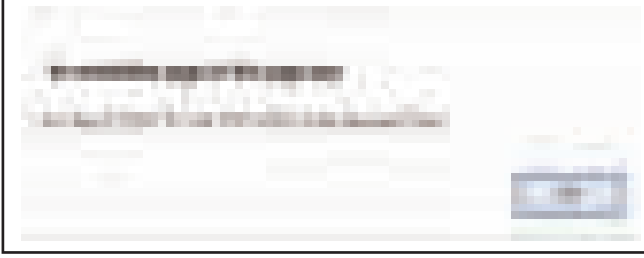
પરિણામ:

નોંધ: સેટ ફુલ યર() પદ્ધતિ વૈકલ્પિક રીતે મહિનો અને દિવસ સેટ કરી શકે છે.

ઉદાહરણ 40

```
<script>
var d = new Date();
```

Fig 20



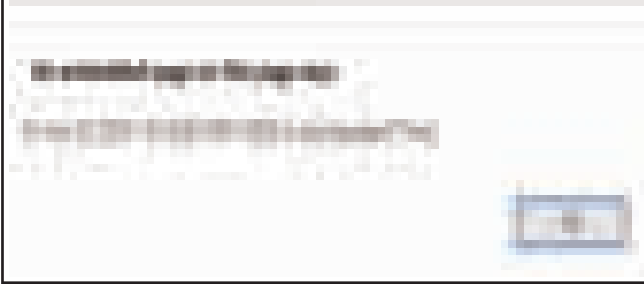
```
d.setFullYear(2018, 10, 2);
```

```
alert(d);
```

```
<script>
```

પરિણામ: (ફિગ 21)

Fig 21



નોંધ: 0 થી મહિનાની ગણતરી થાય છે. ડિસેમ્બર મહિનો 11 છે

સેટ મહિનો() પદ્ધતિ

સેટ Month() પદ્ધતિ તારીખ ઓબ્જેક્ટ (0-11)નો મહિનો સેટ કરે છે.

ઉદાહરણ 41

```
<script>
```

```
var d = new Date();
```

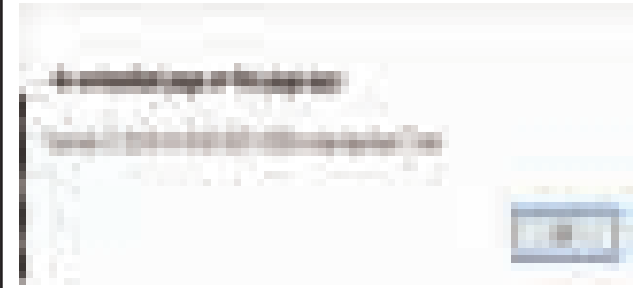
```
d.setMonth(2);
```

```
alert(d);
```

```
</script>
```

પરિણામ: (ફિગ 22)

Fig 22



સેટ તારીખ () પદ્ધતિ

સેટ તારીખ () પદ્ધતિ તારીખ ઓબ્જેક્ટનો દિવસ સેટ કરે છે (1-31).

ઉદાહરણ 42

```
<script>
```

```
var d = new Date();
```

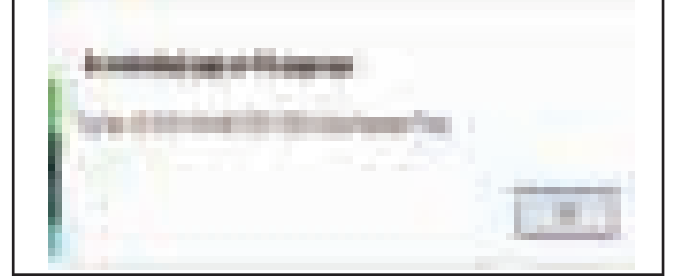
```
d.setDate(18);
```

```
alert(d);
```

```
</script>
```

પરિણામ: (ફિગ 23)

Fig 23



સેટ તારીખ() પદ્ધતિનો ઉપયોગ તારીખમાં દિવસો ઉમેરવા માટે પણ થઈ શકે છે.

ઉદાહરણ 43

```
<script>
```

```
var d = new Date();
```

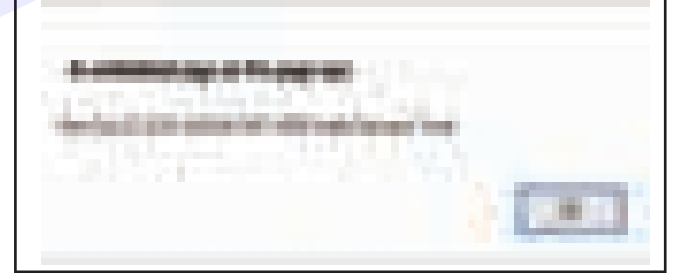
```
d.setDate(d.getDate() + 25);
```

```
alert(d);
```

```
</script>
```

પરિણામ: (ફિગ 24)

Fig 24



જો દિવસો ઉમેરવાથી, મહિનો અથવા વર્ષ બદલાય છે, તો ફેરફારો તારીખ ઓબ્જેક્ટ દ્વારા આપમેળે નિયંત્રિત થાય છે.

સેટ Hours() પદ્ધતિ

સેટ Hours() પદ્ધતિ તારીખ ઓબ્જેક્ટના કલાકો સેટ કરે છે (0-23).

ઉદાહરણ 44

```
<script>
```

```
var d = new Date();
```

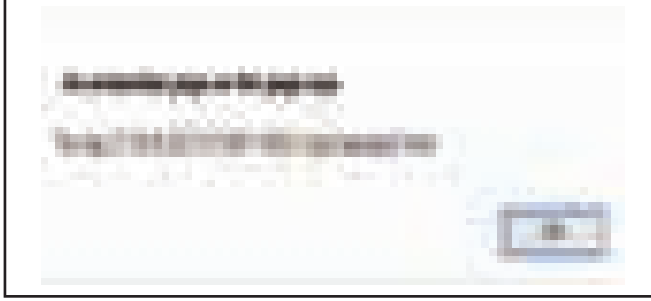
```
d.setHours(20);
```

```
alert(d);
```

```
</script>
```

પરિણામ: (ફિગ 25)

Fig 25



સેટ મિનિટ() પદ્ધતિ

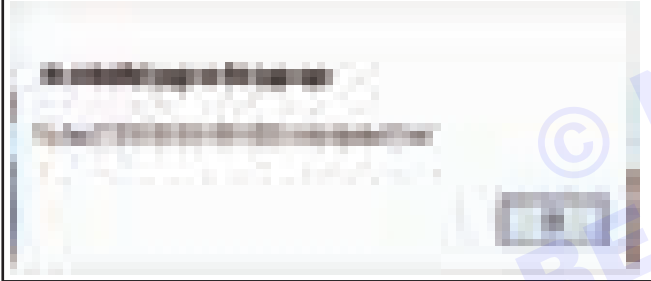
સેટ મિનિટ્સ() પદ્ધતિ તારીખ ઓબ્જેક્ટ (0-59)ની મિનિટ સેટ કરે છે.

ઉદાહરણ 45

```
<script>
var d = new Date();
d.setMinutes(15);
alert(d);
</script>
```

પરિણામ: (ફિગ 26)

Fig 26



સેટ સેકન્ડ્સ() પદ્ધતિ

સેટ સેકન્ડ્સ() પદ્ધતિ તારીખ ઓબ્જેક્ટ (0-59) ની સેકન્ડ સેટ કરે છે.

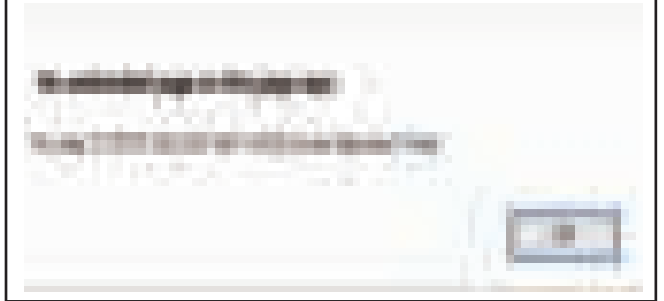
ઉદાહરણ 46

```
<script>
```

```
var d = new Date();
d.setSeconds(20);
alert(d);
</script>
```

પરિણામ: (ફિગ 27)

Fig 27



તારીખોની સરખામણી કરો

તારીખો સરખાતાથી સરખાવી શકાય છે.

નીચેનું ઉદાહરણ આજની તારીખની સરખામણી જાન્યુઆરી 14, 2100 સાથે કરે છે.

ઉદાહરણ 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.");
}
```

IIS નો ઉપયોગ કરીને વેબ પ્રોજેક્ટનો ઉપયોગ કરો (Deploy web project using IIS)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- DOM ને વ્યાખ્યાયિત કરો
- DOM પદ્ધતિઓ સમજાવો
- DOM દસ્તાવેજો સમજાવો
- HTML DOM તત્વોનું વર્ણન કરો
- HTML DOM ઇવેન્ટ્સ સમજાવો
- ઓપન સોર્સ સોફ્ટવેર જાણો

JavaScript HTML DOM

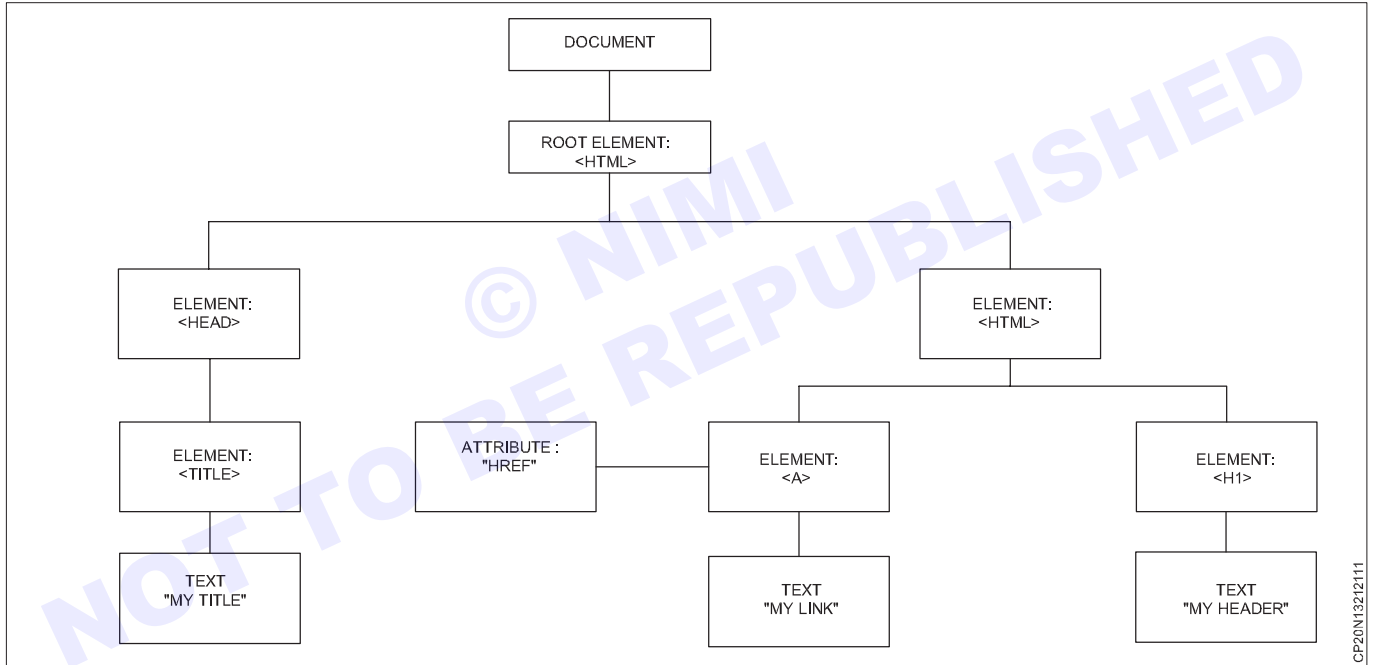
HTML DOM સાથે, JavaScript HTML દસ્તાવેજના તમામ ઘટકોને ઍક્સેસ કરી શકે છે અને તેમાં ફેરફાર કરી શકે છે.

HTML DOM (ડોક્યુમેન્ટ ઓબ્જેક્ટ મોડલ) : જ્યારે વેબ પેજ લોડ થાય છે, ત્યારે બ્રાઉઝર પેજનું ડોક્યુમેન્ટ ઓબ્જેક્ટ મોડલ બનાવે છે.

HTML DOM મોડલ ઓબ્જેક્ટના વૃક્ષ તરીકે બનાવવામાં આવ્યું છે:

ઓબ્જેક્ટ્સનું HTML DOM ટ્રી

ઓબ્જેક્ટ મોડલ સાથે, JavaScript ને ગતિશીલ HTML બનાવવા માટે જરૂરી તમામ શક્તિ મળે છે:



- JavaScript પૃષ્ઠના તમામ HTML ઘટકોને બદલી શકે છે.
- JavaScript પૃષ્ઠની તમામ HTML વિશેષતાઓને બદલી શકે છે
- JavaScript પૃષ્ઠના તમામ CSS શૈલીઓને બદલી શકે છે
- JavaScript હાલના HTML તત્વો અને વિશેષતાઓને દૂર કરી શકે છે
- JavaScript નવા HTML તત્વો અને વિશેષતાઓ ઉમેરી શકે છે
- JavaScript પૃષ્ઠમાં અસ્તિત્વમાં છે તે તમામ HTML ઇવેન્ટ્સ પર પ્રતિક્રિયા આપી શકે છે

DOM

DOM એ W3C (વર્લ્ડ વાઇડ વેબ કન્સોર્ટિયમ) સ્ટાન્ડર્ડ છે.

DOM દસ્તાવેજોને ઍક્સેસ કરવા માટેના ધોરણને વ્યાખ્યાયિત કરે છે:

“W3C ડોક્યુમેન્ટ ઓબ્જેક્ટ મોડલ (DOM) એ એક પ્લેટફોર્મ અને ભાષા-તટસ્થ ઇન્ટરફેસ છે જે પ્રોગ્રામ્સ અને સ્ક્રિપ્ટ્સને દસ્તાવેજની સામગ્રી, માળખું અને શૈલીને ગતિશીલ રીતે ઍક્સેસ અને અપડેટ કરવાની મંજૂરી આપે છે.”

W3C DOM સ્ટાન્ડર્ડ 3 જુદા જુદા ભાગોમાં વિભાજિત થયેલ છે:

- કોર DOM - તમામ દસ્તાવેજ પ્રકારો માટે પ્રમાણભૂત મોડલ
- XML DOM - XML દસ્તાવેજો માટે પ્રમાણભૂત મોડલ
- HTML DOM - HTML દસ્તાવેજો માટે પ્રમાણભૂત મોડલ

HTML DOM શું છે?

HTML DOM એ HTML માટે પ્રમાણભૂત ઓબ્જેક્ટ મોડલ અને પ્રોગ્રામિંગ ઇન્ટરફેસ છે. તે વ્યાખ્યાયિત કરે છે:

- ઓબ્જેક્ટ તરીકે HTML તત્વો

- બધા HTML ઘટકોના ગુણધર્મો
- તમામ HTML ઘટકોને ઍક્સેસ કરવાની પદ્ધતિઓ
- તમામ HTML ઘટકો માટેની ઘટનાઓ

બીજા શબ્દોમાં કહીએ તો: HTML DOM એ HTML ઘટકોને કેવી રીતે મેળવવું, બદલવું, ઉમેરવું અથવા કાઢી નાખવાનું પ્રમાણભૂત છે.

HTML DOM પદ્ધતિઓ

HTML DOM પદ્ધતિઓ એ ક્રિયાઓ છે જે તમે કરી શકો છો (HTML તત્વો પર)

HTML DOM પ્રોપર્ટીઝ એ મૂલ્યો (HTML તત્વોના) છે જે તમે સેટ અથવા બદલી શકો છો.

DOM પ્રોગ્રામિંગ ઇન્ટરફેસ

HTML DOM ને JavaScript (અને અન્ય પ્રોગ્રામિંગ ભાષાઓ સાથે) ઍક્સેસ કરી શકાય છે.

DOM માં, બધા HTML ઘટકોને ઑબ્જેક્ટ તરીકે વ્યાખ્યાયિત કરવામાં આવે છે. પ્રોગ્રામિંગ ઇન્ટરફેસ એ દરેક ઑબ્જેક્ટના ગુણધર્મો અને પદ્ધતિઓ છે. મિલકત એ એક મૂલ્ય છે જે તમે મેળવી શકો છો અથવા સેટ કરી શકો છો (જેમ કે HTML ઘટકની સામગ્રી બદલવી). પદ્ધતિ એ એક ક્રિયા છે જે તમે કરી શકો છો (જેમ કે HTML ઘટક ઉમેરવું અથવા કાઢી નાખવું).

ઉદાહરણ 1

The following example changes the content (the innerHTML) of the <p> element with id="demo":

of the <p> element with id="demo":

```
<html>
<body>
<p id="demo"></p>
<script>
```

HTML તત્વો શોધવી

પદ્ધતિ	વર્ણન
document.getElementById()	તત્વ id દ્વારા એક તત્વ શોધો
document.getElementsByTagName()	ટેગ નામ દ્વારા તત્વો શોધો
document.getElementsByClassName()	વર્ગના નામ દ્વારા તત્વો શોધો

HTML તત્વો બદલવાનું

પદ્ધતિ	વર્ણન
element.innerHTML=	એલિમેન્ટનું આંતરિક HTML બદલો
element.attribute=	HTML એલિમેન્ટની વિશેષતા બદલો
element.setAttribute(attribute,value)	HTML એલિમેન્ટની વિશેષતા બદલો
element.style.property=	HTML ઘટકની શૈલી બદલો

```
document.getElementById("demo").innerHTML =
```

```
"Welcome to JavaScript!";
```

```
</script>
```

```
</body>
```

```
</html>
```

ઉપરના ઉદાહરણમાં, getElementById એક પદ્ધતિ છે, જ્યારે innerHTML એ પ્રોપર્ટી છે.

getElementById પદ્ધતિ

HTML ઘટકને ઍક્સેસ કરવાની સૌથી સામાન્ય રીત એ એલિમેન્ટની id નો ઉપયોગ કરવાનો છે. ઉપરના ઉદાહરણમાં getElementById પદ્ધતિએ તત્વ શોધવા માટે id="demo" નો ઉપયોગ કર્યો છે.

ઇનરએચટીએમએલ પ્રોપર્ટી

તત્વની સામગ્રી મેળવવાનો સૌથી સહેલો રસ્તો છે innerHTML ગુણધર્મનો ઉપયોગ કરીને. ઇનરએચટીએમએલ ગુણધર્મ HTML ઘટકોની સામગ્રી મેળવવા અથવા બદલવા માટે ઉપયોગી છે.

નોંધ: innerHTML ગુણધર્મનો ઉપયોગ કોઈપણ HTML તત્વ મેળવવા અથવા બદલવા માટે થઈ શકે છે, જેમાં અને .

HTML DOM દસ્તાવેજ

HTML DOM દસ્તાવેજ ઑબ્જેક્ટ

દસ્તાવેજ પદાર્થ છે.તમારા વેબ પૃષ્ઠમાં અન્ય તમામ ઑબ્જેક્ટના માલિક. HTML DOM ઑબ્જેક્ટ મોડેલમાં, દસ્તાવેજ ઑબ્જેક્ટ તમારા વેબ પૃષ્ઠને રજૂ કરે છે. જો તમે HTML પૃષ્ઠમાં ઑબ્જેક્ટને ઍક્સેસ કરવા માંગો છો, તો તમે હંમેશા દસ્તાવેજ ઑબ્જેક્ટને ઍક્સેસ કરવા સાથે પ્રારંભ કરો છો.

નીચે કેટલાક ઉદાહરણો છે કે તમે કેવી રીતે HTML ને ઍક્સેસ કરવા અને તેને ચાલાકી કરવા માટે દસ્તાવેજ ઑબ્જેક્ટનો ઉપયોગ કરી શકો છો.

HTML તત્વો ઉમેરવા અને કાઢી નાખવા

પદ્ધતિ	વર્ણન
document.createElement()	એક HTML ઘટક બનાવો
document.removeChild()	HTML ઘટક દૂર કરો
document.appendChild()	એક HTML ઘટક ઉમેરો
document.replaceChild()	HTML ઘટક બદલો
document.write(text)	HTML આઉટપુટ સ્ટ્રીમમાં લખો

ઇવેન્ટ હેન્ડલર્સ ઉમેરવાનું

પદ્ધતિ	વર્ણન
document.getElementById(id).onclick=function(){code	ઓનક્લિક ઇવેન્ટમાં ઇવેન્ટ હેન્ડલર કોડ ઉમેરવાનું

JavaScript HTML DOM તત્વો

HTML તત્વો શોધવી

ઘણી વાર, JavaScript સાથે, તમે HTML ઘટકોને ચાલાકી કરવા માંગો છો.

આ કરવા માટે, તમારે પહેલા તત્વો શોધવા પડશે. આ કરવાની કેટલીક રીતો છે:

- id દ્વારા HTML તત્વો શોધવી
- ટેગ નામ દ્વારા HTML તત્વો શોધવી
- વર્ગના નામ દ્વારા HTML તત્વો શોધવી
- HTML શોધવી HTML ઓબ્જેક્ટ સંગ્રહ દ્વારા તત્વો

ID દ્વારા HTML તત્વો શોધવી

DOM માં HTML તત્વો શોધવાનો સૌથી સહેલો રસ્તો એલિમેન્ટ આઈડીનો ઉપયોગ કરીને છે.

આ ઉદાહરણ id="demo" સાથે તત્વ શોધે છે:

ઉદાહરણ 2

```
var x = document.getElementById("demo");
```

If the element is found, the method will return the element as an object (in x).

If the element is not found, x will contain null.

ટેગ નામ દ્વારા HTML તત્વો શોધવી

આ ઉદાહરણ id="main" સાથે તત્વ શોધે છે અને પછી "main" ની અંદરના તમામ ઘટકો શોધે છે:

ઉદાહરણ 3

```
var x = document.getElementById("main");  
var y = x.getElementsByTagName("p");
```

વર્ગના નામ દ્વારા HTML તત્વો શોધવી

જો તમે સમાન વર્ગના નામ સાથેના બધા HTML ઘટકો શોધવા માંગતા હો, તો આ પદ્ધતિનો ઉપયોગ કરો ElementsByClassName() મેળવો

ઉદાહરણ 4

```
document.getElementsByClassName("intro");
```

The example above returns a list of all elements with class="intro".

નોંધ: વર્ગના નામ દ્વારા તત્વો શોધવાનું ઇન્ટરનેટ એક્સપ્લોરર 5,6,7 અને 8 માં કામ કરતું નથી.

HTML ઓબ્જેક્ટ કલેક્શન દ્વારા HTML તત્વો શોધવી

આ ઉદાહરણ ફોર્મ કલેક્શનમાં id="frm1" સાથે ફોર્મ એલિમેન્ટ શોધે છે, અને તમામ ઘટક મૂલ્યો દર્શાવે છે:

પદ્ધતિ	વર્ણન	DOM
document.anchors	નામ વિશેષતામાં મૂલ્ય સાથે બધું પરત કરે છે	1
document.applets	બધા ઘટકો પરત કરે છે (HTML5 માં નાપસંદ)	1
document.baseURI	દસ્તાવેજનો સંપૂર્ણ આધાર URI પરત કરે છે	3
document.body	તત્વ પરત કરે છે	1
document.cookie	દસ્તાવેજની કૂકી પરત કરે છે	1
document.doctype	દસ્તાવેજનો doctype પરત કરે છે	3

document.documentElement	<html> તત્વ પરત કરે છે	3
document.documentMode	બ્રાઉઝર દ્વારા ઉપયોગમાં લેવાતો મોડ પરત કરે છે	3
document.documentURI	દસ્તાવેજનો URI પરત કરે છે	3
document.domain	દસ્તાવેજ સર્વરનું ડોમેન નામ પરત કરે છે	1
document.domConfig	DOM રૂપરેખાંકન પરત કરે છે	3
document.embeds	બધા <embed> ઘટકો પરત કરે છે	3
document.forms	બધા <form> ઘટકો પરત કરે છે	1
document.head	<head> તત્વ પરત કરે છે	3
document.images	બધા <image> ઘટકો પરત કરે છે	1
document.implementation	DOM અમલીકરણ પરત કરે છે	3
document.inputEncoding	દસ્તાવેજનું એન્કોડિંગ પરત કરે છે (અક્ષર સમૂહ)	3
document.lastModified	દસ્તાવેજ અપડેટ કરવામાં આવ્યો હતો તે તારીખ અને સમય પરત કરે છે	3
document.links	href માં તમામ અને તત્વોની કિંમત પરત કરે છે	1
document.readyState	દસ્તાવેજની સ્થિતિ (લોડિંગ) પરત કરે છે	3
document.referrer	રેફરરનો URI પરત કરે છે (લિંકિંગ દસ્તાવેજ)	1
document.scripts	બધા <script> ઘટકો પરત કરે છે	3
document.strictErrorChecking	જો ભૂલ ચકાસણી લાગુ કરવામાં આવી હોય તો પરત કરે છે	3
document.title	<title> તત્વ પરત કરે છે	1
document.URL	દસ્તાવેજનું સંપૂર્ણ URL પરત કરે છે	1

ઉદાહરણ 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("ડેમો").માં HTML = ટેક્સ્ટ;
નીચેના HTML ઓબ્જેક્ટ્સ (અને ઓબ્જેક્ટ સંગ્રહો) પણ સુલભ છે:

- document.anchors
- document.body
- document.documentElement
- document.embeds
- document.forms
- document.head
- document.images
- document.links
- document.scripts

- document.title

HTML DOM JavaScript ને HTML ઘટકોની સામગ્રી બદલવાની મંજૂરી આપે છે.

HTML આઉટપુટ સ્ટ્રીમ બદલવું

JavaScript ડાયનેમિક HTML સામગ્રી બનાવી શકે છે. JavaScript માં, document.write() નો ઉપયોગ સીધા HTML આઉટપુટ સ્ટ્રીમ પર લખવા માટે થઈ શકે છે.

ઉદાહરણ 6

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(Date());
</script>
</body>
</html>
```

નોંધ: દસ્તાવેજ લોડ થયા પછી ક્યારેય document.write() નો ઉપયોગ કરશો નહીં. તે દસ્તાવેજને ઓવરરાઈટ કરશે.

HTML સામગ્રી બદલવી

HTML તત્વની સામગ્રીને સંશોધિત કરવાની સૌથી સરળ રીત છે inner-HTML ગુણધર્મનો ઉપયોગ કરીને. HTML તત્વની સામગ્રીને બદલવા માટે, આ વાક્યરચનાનો ઉપયોગ કરો.

```
document.getElementById(id).innerHTML = new HTML
```

This example changes the content of a <p> element:

ઉદાહરણ 7

```
<html>
<body>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
</body>
</html>
```

ઉદાહરણ સમજાવ્યું:

- ઉપરના HTML દસ્તાવેજમાં id="p1" સાથેનું એક ઘટક છે
- અમે id="p1" સાથે તત્વ મેળવવા માટે HTML DOM નો ઉપયોગ કરીએ છીએ
- JavaScript તે ઘટકની સામગ્રી (innerHTML) ને “નવા ટેક્સ્ટ!” માં બદલી દે છે.

આ ઉદાહરણ તત્વની સામગ્રીને બદલે છે:

ઉદાહરણ 8

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Old Heading</h1>
<script>
var element = document.getElementById("id1");
element.innerHTML = "New Heading";
</script>
</body>
</html>
```

ઉદાહરણ સમજાવ્યું:

- ઉપરના HTML દસ્તાવેજમાં id="id1" સાથેનું એક ઘટક છે
- અમે id="id1" સાથે તત્વ મેળવવા માટે HTML DOM નો ઉપયોગ કરીએ છીએ
- JavaScript તે ઘટકની સામગ્રી (innerHTML) ને “નવું મથાળું” માં બદલી નાખે છે

એટ્રિબ્યુટનું મૂલ્ય બદલવું HTML

એટ્રિબ્યુટની કિંમત બદલવા માટે, આ સિન્ટેક્સનો ઉપયોગ કરો.

```
document.getElementById(id).attribute = new value
```

આ ઉદાહરણ એલિમેન્ટના src એટ્રિબ્યુટના મૂલ્યને બદલે છે.

ઉદાહરણ 9

```
<!DOCTYPE html>
<html>
<body>

<script>
document.getElementById("Image1").src =
"newflower.jpg";
</script>
</body>
</html>
```

ઉદાહરણ સમજાવ્યું:

- ઉપરના HTML દસ્તાવેજમાં id="myImage" સાથેનું એક ઘટક છે
- અમે id="myImage" સાથે તત્વ મેળવવા માટે HTML DOM નો ઉપયોગ કરીએ છીએ
- JavaScript તે તત્વના src લક્ષણને “smiley.gif” થી “landscape.jpg” માં બદલી દે છે

HTML શૈલી બદલવી

HTML ઘટકની શૈલી બદલવા માટે, આ વાક્યરચનાનો ઉપયોગ કરો.

```
document.getElementById(id).style.property = new style
```

The following example changes the style of a <p> element:

ઉદાહરણ 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>
```

ઇવેન્ટ્સનો ઉપયોગ કરવો : HTML DOM તમને જ્યારે ઇવેન્ટ થાય ત્યારે કોડ એક્ઝિક્યુટ કરવાની મંજૂરી આપે છે. જ્યારે HTML ઘટકોમાં “વસ્તુઓ થાય છે” ત્યારે બ્રાઉઝર દ્વારા ઇવેન્ટ્સ જનરેટ થાય છે.

- એક તત્વ પર ક્લિક કરવામાં આવે છે
- પૃષ્ઠ લોડ થઈ ગયું છે
- ઈનપુટ ક્ષેત્રો બદલાઈ ગયા છે

જ્યારે વપરાશકર્તા બટન પર ક્લિક કરે છે ત્યારે આ ઉદાહરણ id="id1" સાથે HTML ઘટકની શૈલીને બદલે છે.

ઉદાહરણ 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>
```

JavaScript HTML DOM ઇવેન્ટ્સ

HTML DOM JavaScript ને HTML ઇવેન્ટ્સ પર પ્રતિક્રિયા કરવાની મંજૂરી આપે છે.

ઘટનાઓ પર પ્રતિક્રિયા

JavaScript ને એક્ઝિક્યુટ કરી શકાય છે જ્યારે કોઈ ઇવેન્ટ થાય છે, જેમ કે જ્યારે વપરાશકર્તા HTML એલિમેન્ટ પર ક્લિક કરે છે. જ્યારે વપરાશકર્તા એલિમેન્ટ પર ક્લિક કરે ત્યારે કોડને એક્ઝિક્યુટ કરવા માટે, HTML ઇવેન્ટ એટ્રિબ્યુટમાં JavaScript કોડ ઉમેરો

onclick=JavaScript

HTML ઇવેન્ટ્સના ઉદાહરણો:

- જ્યારે વપરાશકર્તા માઉસ પર ક્લિક કરે છે
- જ્યારે વેબ પેજ લોડ થાય છે
- જ્યારે ઈમેજ લોડ કરવામાં આવી હોય
- જ્યારે માઉસ કોઈ તત્વ પર ફરે છે
- જ્યારે ઈનપુટ ફીલ્ડ બદલાય છે
- જ્યારે HTML ફોર્મ સબમિટ કરવામાં આવે છે
- જ્યારે વપરાશકર્તા કી સ્ટ્રોક કરે છે

આ ઉદાહરણમાં, જ્યારે વપરાશકર્તા તેના પર ક્લિક કરે છે ત્યારે તત્વની સામગ્રી બદલાઈ જાય છે.

ઉદાહરણ 12

```
<!DOCTYPE html>

<html>

<body>
```

```
<h1 onclick="this.innerHTML = 'Oops!'">Click on this

text!</h1>

</body>

</html>
```

HTML ઇવેન્ટ લક્ષણો

HTML ઘટકોને ઇવેન્ટ્સ સોંપવા માટે તમે ઇવેન્ટ લક્ષણોનો ઉપયોગ કરી શકો છો.

ઉદાહરણ 13

બટન એલિમેન્ટને ઓનક્લિક ઇવેન્ટ સોંપો:

```
<button onclick="displayDate()">Try it</button>
```

ઉપરના ઉદાહરણમાં, એક કાર્યજ્યારે બટન ક્લિક કરવામાં આવશે ત્યારે ડિસ્પ્લે ડેટ નામ આપવામાં આવશે.

HTML DOM નો ઉપયોગ કરીને ઇવેન્ટ્સ સોંપો

HTML DOM તમને JavaScript નો ઉપયોગ કરીને HTML ઘટકોને ઇવેન્ટ્સ સોંપવાની મંજૂરી આપે છે.

ઉદાહરણ 14

Assign an onclick event to a button element:

```
<script>

document.getElementById("myBtn").onclick =

displayDate;

</script>
```

ઉપરના ઉદાહરણમાં, displayDate નામનું ફંક્શન id="myBtn" સાથેના HTML ઘટકને અસાઈન કરવામાં આવ્યું છે.

જ્યારે બટન ક્લિક કરવામાં આવશે ત્યારે ફંક્શન એક્ઝિક્યુટ થશે.

ઓનલોડ અને ઓનલોડ ઇવેન્ટ્સ

જ્યારે વપરાશકર્તા પેજમાં પ્રવેશે છે અથવા છોડે છે ત્યારે ઓનલોડ અને ઓનલોડ ઇવેન્ટ્સ ટ્રિગર થાય છે. ઓનલોડ ઇવેન્ટનો ઉપયોગ મુલાકાતીઓના બ્રાઉઝર પ્રકાર અને બ્રાઉઝર સંસ્કરણને તપાસવા અને માહિતીના આધારે વેબ પૃષ્ઠનું યોગ્ય સંસ્કરણ લોડ કરવા માટે થઈ શકે છે. ઓનલોડ અને ઓનલોડ ઇવેન્ટ્સનો ઉપયોગ ફૂકીઝ સાથે વ્યવહાર કરવા માટે થઈ શકે છે.

ઉદાહરણ 15

```
<body onload="checkCookies()">
```

The onchange Event

ઓનચેન્જ ઇવેન્ટનો ઉપયોગ ઈનપુટ ફીલ્ડની માન્યતા સાથે સંયોજનમાં થાય છે. નીચે ઓનચેન્જનો ઉપયોગ કેવી રીતે કરવો તેનું ઉદાહરણ છે. જ્યારે વપરાશકર્તા ઈનપુટ ફીલ્ડની સામગ્રીને બદલે છે ત્યારે અપરકેસ() ફંક્શનને કોલ કરવામાં આવશે.

ઉદાહરણ 16

```
<input type="text" id="fname" onchange="upperCase()">
```

ઓનમાઉસઓવર અને ઓનમાઉસઆઉટ ઇવેન્ટ્સ

જ્યારે વપરાશકર્તા HTML એલિમેન્ટ ઉપર અથવા બહાર માઉસ કરે છે ત્યારે ઓન-માઉસઓવર અને માઉસઆઉટ ઇવેન્ટનો ઉપયોગ ફંક્શનને ટ્રિગર કરવા માટે થઈ શકે છે.

onmousedown, onmouseup અને onclick ઇવેન્ટ્સ

onmousedown, onmouseup અને onclick ઇવેન્ટ્સ એ માઉસ-ક્લિકના તમામ ભાગો છે. પ્રથમ જ્યારે માઉસ બટન ક્લિક કરવામાં આવે છે, ત્યારે ઓન-માઉસડાઉન ઇવેન્ટ ટ્રિગર થાય છે, પછી, જ્યારે માઉસ-બટન રિલીઝ થાય છે, ત્યારે ઓન-માઉસઅપ ઇવેન્ટ ટ્રિગર થાય છે, છેલ્લે, જ્યારે માઉસ ક્લિક પૂર્ણ થાય છે, ત્યારે ઓનક્લિક ઇવેન્ટ ટ્રિગર થાય છે.

onmousedown અને onmouseup

જ્યારે વપરાશકર્તા માઉસ બટન દબાવી રાખે છે ત્યારે છબી બદલો.

ઓનલોડ

જ્યારે પૃષ્ઠ લોડ કરવાનું સમાપ્ત કરે ત્યારે ચેતવણી બોક્સ દર્શાવો.

ફોકસ

જ્યારે ઇનપુટ ફીલ્ડ પર ફોકસ થાય ત્યારે તેનો બેકગ્રાઉન્ડ-કલર બદલો.

માઉસ ઇવેન્ટ્સ

જ્યારે કર્સર તેની ઉપર ફરે ત્યારે તેનો રંગ બદલો.

DOM ઇવેન્ટ લિસનર

addEventListener() પદ્ધતિ

એક ઇવેન્ટ લિસનર ઉમેરો જે જ્યારે વપરાશકર્તા બટન પર ક્લિક કરે છે ત્યારે ફાયર થાય છે.

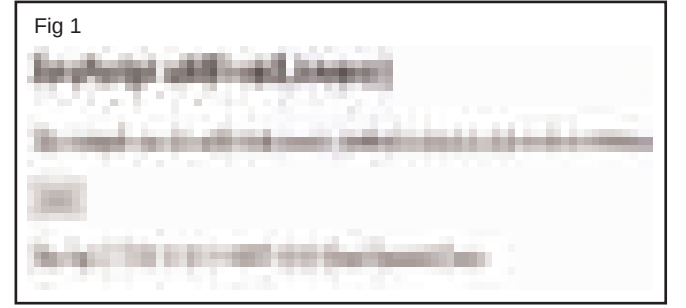
ઉદાહરણ 17

```
<!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>

પરિણામ: (ફિગ 1)



- addEventListener() પદ્ધતિ ઇવેન્ટ હેન્ડલરને ઉલ્લેખિત તત્વ સાથે જોડે છે.
- addEventListener() પદ્ધતિ હાલના ઇવેન્ટ હેન્ડલર્સને ઓવરરાઈટ કર્યા વિના એલિમેન્ટ સાથે ઇવેન્ટ હેન્ડલરને જોડે છે.
- તમે એકમાં ઘણા ઇવેન્ટ હેન્ડલર તત્વ ઉમેરી શકો છો
- તમે એક જ પ્રકારના ઘણા ઇવેન્ટ હેન્ડલર્સને એક ઘટકમાં ઉમેરી શકો છો, એટલે કે બે "ક્લિક કરો" ઇવેન્ટ.
- તમે કોઈપણ DOM ઓબ્જેક્ટમાં માત્ર HTML ઘટકોમાં ઇવેન્ટ શ્રોતાઓને ઉમેરી શકો છો. એટલે કે વિન્ડો ઓબ્જેક્ટ.
- addEventListener() પદ્ધતિ ઇવેન્ટ બબલિંગ પર કેવી પ્રતિક્રિયા આપે છે તે નિયંત્રિત કરવાનું સરળ બનાવે છે.
- addEventListener() પદ્ધતિનો ઉપયોગ કરતી વખતે, JavaScript વધુ સારી રીતે વાંચી શકાય તે માટે,
- HTML માર્કઅપથી અલગ કરવામાં આવે છે અને જ્યારે તમે HTML માર્કઅપને નિયંત્રિત ન કરતા હો ત્યારે પણ તમને ઇવેન્ટ શ્રોતાઓને ઉમેરવાની મંજૂરી આપે છે.
- તમે removeEventListener() પદ્ધતિનો ઉપયોગ કરીને ઇવેન્ટ સાંભળનારને સરળતાથી દૂર કરી શકો છો.

ઓપન સોર્સ સોફ્ટવેર

તમે આ લોકપ્રિય ફ્રી અને ઓપન સોર્સ વેબસાઈટ બિલ્ડિંગ ટૂલ્સનો ઉપયોગ કરીને વેબસાઈટ બનાવી શકો છો. આજકાલ, ભલે તમે વ્યક્તિગત ઉદ્યોગસાહસિક હો કે વ્યવસાયિક સંસ્થાનું પ્રતિનિધિત્વ કરતા હોવ, વ્યક્તિગત અને વ્યાવસાયિક વિકાસ માટે વેબસાઈટ આવશ્યક છે. સંસ્થાઓ આકર્ષક વેબસાઈટ બનાવવા માટે ઘણા પૈસા ખર્ચી રહી છે. નીચે આપેલા કેટલાક ઓપન સોર્સ વેબસાઈટ બિલ્ડિંગ ટૂલ્સ છે જેનો ઉપયોગ તમે પ્રોગ્રામિંગ અથવા ઇન્ટરનેટ વિશે વધુ જાણકારી વગર તમારી જાતે તમારી વેબસાઈટ બનાવવા માટે કરી શકો છો.

1 વર્ડપ્રેસ

વર્ડપ્રેસ માટે સત્તાવાર વેબસાઈટ્સ છે
<https://wordpress.com> અને
<https://wordpress.org/>.

2 કોમ્પોઝર

કોમ્પોઝર માટે સત્તાવાર વેબસાઈટ છે

<https://www.kompozer.net>.

3 જુમલા

જુમલા માટેની સત્તાવાર વેબસાઈટ છે

<https://www.joomla.org/>.

4 Drupal

Drupal માટે સત્તાવાર વેબસાઈટ છે

<https://www.drupal.org/>.

5 OpenCms

OpenCms માટેની સત્તાવાર વેબસાઈટ છે

<http://www.opencms.org/en/>

© NIMI
NOT TO BE REPUBLISHED

અદ્યતન ફોર્મ્યુલા અને મેક્રો બનાવો (Create advanced formulas and macros)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- સંદર્ભોના પ્રકાર
- એક્સેલ લિકેજ કસ્ટમ ફોર્મેટ & એક્સેલ પ્રોટેક્શન
- પીવટ ટેબલ & ચાર્ટ વ્યાખ્યાયિત કરો
- પાવર ક્વેરી વ્યાખ્યાયિત કરો.

એડવાન્સ એક્સેલ

સંદર્ભોના પ્રકાર

બે પ્રકારના કોષ સંદર્ભો છે: સંબંધિત અને સંપૂર્ણ. જ્યારે અન્ય કોષોમાં નકલ અને ભરવામાં આવે ત્યારે સંબંધિત અને સંપૂર્ણ સંદર્ભો અલગ રીતે વર્તે છે. જ્યારે અન્ય કોષમાં ફોર્મ્યુલાની નકલ કરવામાં આવે ત્યારે સંબંધિત સંદર્ભો બદલાય છે. સંપૂર્ણ સંદર્ભો, બીજી તરફ, તેઓની નકલ ક્યાંય પણ કરવામાં આવી હોય તો પણ તે સ્થિર રહે છે.

=સંબંધિત સંદર્ભો

મૂળભૂત રીતે, બધા કોષ સંદર્ભો સંબંધિત સંદર્ભો છે. જ્યારે બહુવિધ કોષોમાં કોપી કરવામાં આવે છે, ત્યારે તેઓ પંક્તિઓ અને કોલમ્સની સંબંધિત સ્થિતિના આધારે બદલાય છે. ઉદાહરણ તરીકે, જો તમે પંક્તિ 1 થી પંક્તિ 2 માં સૂત્ર $=A1+B1$ ની નકલ કરો છો, તો સૂત્ર $=A2+B2$ બનશે. જ્યારે પણ તમારે બહુવિધ પંક્તિઓ અથવા કોલમમાં સમાન ગણતરીનું પુનરાવર્તન કરવાની જરૂર હોય ત્યારે સંબંધિત સંદર્ભો ખાસ કરીને અનુકૂળ હોય છે.

સંપૂર્ણ સંદર્ભો

એવા સમયે હોઈ શકે છે જ્યારે તમે કોષો ભરતી વખતે સેલ સંદર્ભ બદલવા માંગતા નથી. સંબંધિત સંદર્ભોથી વિપરીત, નિરપેક્ષ સંદર્ભો જ્યારે નકલ અથવા ભરવામાં આવે ત્યારે બદલાતા નથી. તમે પંક્તિ અને/અથવા કોલમ સતત રાખવા માટે ચોક્કસ સંદર્ભનો ઉપયોગ કરી શકો છો.

કોલમ અને પંક્તિ પહેલાં ડોલર ચિહ્ન (\$) ઉમેરીને સૂત્રમાં સંપૂર્ણ સંદર્ભ નિયુક્ત કરવામાં આવે છે. જો તે કોલમ અથવા પંક્તિ (પરંતુ બંને નહીં) ની આગળ હોય, તો તેને મિશ્ર સંદર્ભ તરીકે ઓળખવામાં આવે છે.

\$A\$2	જ્યારે કોપી કરવામાં આવે ત્યારે કોલમ અને મી પંક્તિ બદલાતી નથી
A\$2	જ્યારે કોપી કરવામાં આવે ત્યારે પંક્તિ બદલાતી નથી
\$A2	કોપી કરવામાં આવે ત્યારે કોલમ બદલાતી નથી

તમે મોટાભાગના સૂત્રોમાં સંબંધિત (A2) અને સંપૂર્ણ (\$A\$2) ફોર્મેટનો ઉપયોગ કરશો. મિશ્ર સંદર્ભોનો ઉપયોગ ઓછો વારંવાર થાય છે.

માઈક્રોસોફ્ટ એક્સેલમાં ફોર્મ્યુલા લખતી વખતે, તમે નીચેની વિડિયોમાં બતાવ્યા પ્રમાણે, સંબંધિત, સંપૂર્ણ અને મિશ્ર સેલ સંદર્ભો વચ્ચે સ્વિચ કરવા માટે તમારા કીબોર્ડ પરની F4 કી દબાવી શકો છો. સંપૂર્ણ સંદર્ભ ઝડપથી દાખલ કરવાની આ એક સરળ રીત છે.

એક્સેલ લિકેજ કસ્ટમ ફોર્મેટ અને એક્સેલ પ્રોટેક્શન

જો બિલ્ટ-ઇન નંબર ફોર્મેટ તમારી જરૂરિયાતોને પૂર્ણ કરતું નથી, તો તમે એક નવું નંબર ફોર્મેટ બનાવી શકો છો જે હાલના નંબર ફોર્મેટ પર આધારિત છે અને તેને કસ્ટમ નંબર ફોર્મેટની સૂચિમાં ઉમેરી શકો છો. ઉદાહરણ તરીકે, જો

તમે ગ્રાહકની માહિતી ધરાવતી સ્પ્રેડશીટ બનાવી રહ્યાં છો, તો તમે ટેલિફોન નંબરો માટે નંબર ફોર્મેટ બનાવી શકો છો. તમે ટેલિફોન નંબર તરીકે ફોર્મેટ કરવા માટે સેલમાં નંબરોની સ્ટ્રિંગ પર કસ્ટમ નંબર ફોર્મેટ લાગુ કરી શકો છો.

મહત્વપૂર્ણ : કસ્ટમ નંબર ફોર્મેટ્સ માત્ર નંબરને પ્રદર્શિત કરવાની રીતને અસર કરે છે અને સંખ્યાના અંતર્ગત મૂલ્યને અસર કરતા નથી. કસ્ટમ નંબર ફોર્મેટ્સ સક્રિય વર્કબુક્માં સંગ્રહિત થાય છે અને તમે ખોલો છો તે નવી વર્કબુક માટે ઉપલબ્ધ નથી.

કસ્ટમ નંબર ફોર્મેટ બનાવો

- 1 હોમ ટેબ પર, નંબર જૂથમાં, નંબર ફોર્મેટ સૂચિની નીચે વધુ નંબર ફોર્મેટ્સ પર ક્લિક કરો.
- 2 ફોર્મેટ સેલ ડાયલોગ બોક્સમાં, કેટેગરી હેઠળ, કસ્ટમ પર ક્લિક કરો.
- 3 પ્રકાર સૂચિમાં, બિલ્ટ-ઇન ફોર્મેટ પસંદ કરો જે તમે બનાવવા માંગો છો તેના જેવું લાગે છે. ઉદાહરણ તરીકે, 0.00. તમે જે નંબર ફોર્મેટ પસંદ કરો છો તે પ્રકાર બોક્સમાં દેખાય છે.
- 4 ટાઇપ બોક્સમાં, તમે ઈચ્છો તે ચોક્કસ ફોર્મેટ બનાવવા માટે નંબર ફોર્મેટ કોડ્સમાં ફેરફાર કરો. ઉદાહરણ તરીકે, 000-000 0000. તમારા ફેરફારો બિલ્ટ-ઇન ફોર્મેટને બદલશે નહીં. તેના બદલે, તમારા ફેરફારો નવા કસ્ટમ નંબર ફોર્મેટ બનાવે છે.
- 5 જ્યારે તમે પૂર્ણ કરી લો, ઓકે ક્લિક કરો.

કસ્ટમ નંબર ફોર્મેટ લાગુ કરો

- 1 તમે ફોર્મેટ કરવા માંગો છો તે કોષ અથવા કોષોની શ્રેણી પસંદ કરો.
- 2 હોમ ટેબ પર, નંબર જૂથમાં, નંબર ફોર્મેટ સૂચિની નીચે વધુ નંબર ફોર્મેટ્સ પર ક્લિક કરો.
- 3 ફોર્મેટ સેલ ડાયલોગ બોક્સમાં, શ્રેણી હેઠળ, કસ્ટમ પર ક્લિક કરો.
- 4 પ્રકાર સૂચિના તળિયે, તમે હમણાં જ બનાવેલ બિલ્ટ-ઇન ફોર્મેટ પસંદ કરો. ઉદાહરણ તરીકે, 000-000 0000. તમે જે નંબર ફોર્મેટ પસંદ કરો છો તે પ્રકાર બોક્સમાં દેખાય છે.
- 5 બરાબર ક્લિક કરો.

કસ્ટમ નંબર ફોર્મેટ કાઢી નાખો

- 1 હોમ ટેબ પર, નંબર જૂથમાં, નંબર ફોર્મેટ સૂચિની નીચે વધુ નંબર ફોર્મેટ્સ પર ક્લિક કરો.
- 2 ફોર્મેટ સેલ ડાયલોગ બોક્સમાં, કેટેગરી હેઠળ, કસ્ટમ પર ક્લિક કરો.
- 3 પ્રકાર સૂચિમાં, કસ્ટમ નંબર ફોર્મેટ પસંદ કરો, અને પછી કાઢી નાખો પર ક્લિક કરો.

નોંધો

- બિલ્ટ-ઇન નંબર ફોર્મેટ્સ ડિલીટ કરી શકાતા નથી.
- વર્કબુકમાંના કોઈપણ કોષો કે જે કાઢી નાખેલ કસ્ટમ ફોર્મેટ સાથે ફોર્મેટ કરવામાં આવ્યા હતા તે ડિફોલ્ટ સામાન્ય ફોર્મેટમાં પ્રદર્શિત થશે.

એક્સેલ પ્રોટેક્શન

તમારી એક્સેલ ફાઇલોમાંના ડેટાને એક્સેસ કરવાથી અન્ય લોકોને રોકવા માટે, તમારી એક્સેલ ફાઇલને પાસવર્ડ વડે સુરક્ષિત કરો.

નોંધ: આ વિષય ફક્ત ફાઇલ-સ્તર સુરક્ષાને આવરી લે છે, અને વર્કબુક અથવા વર્કશીટ સુરક્ષાને નહીં. તમારી એક્સેલ ફાઇલ, વર્કબુક અથવા વર્કશીટને સુરક્ષિત કરવા વચ્ચેનો તફાવત જાણવા માટે, Excel માં સુરક્ષા અને સુરક્ષા જુઓ.

- ફાઇલ >select માહિતી પસંદ કરો.
- પ્રોટેક્ટ વર્કબુક બોક્સ પસંદ કરો અને પાસવર્ડ સાથે એન્ક્રિપ્ટ પસંદ કરો.
- પાસવર્ડ બોક્સમાં પાસવર્ડ દાખલ કરો, અને પછી બરાબર પસંદ કરો.
- પાસવર્ડ ફરીથી દાખલ કરો બોક્સમાં પાસવર્ડની પુષ્ટિ કરો, અને પછી બરાબર પસંદ કરો.

ચેતવણી

- માઇક્રોસોફ્ટ ભૂલી ગયેલા પાસવર્ડ્સ પુનઃપ્રાપ્ત કરી શકતું નથી, તેથી ખાતરી કરો કે તમારો પાસવર્ડ ખાસ કરીને યાદગાર છે.
- લંબાઈ, અક્ષરો અથવા સંખ્યાઓના સંદર્ભમાં તમે જે પાસવર્ડ્સનો ઉપયોગ કરો છો તેના પર કોઈ નિયંત્રણો થી, પરંતુ પાસવર્ડ્સ કેસ-સંવેદનશીલ હોય છે.
- ક્રેડિટ કાર્ડ નંબર જેવી સંવેદનશીલ માહિતી ધરાવતી પાસવર્ડ્સથી સુરક્ષિત ફાઇલોનું વિતરણ કરવું હંમેશા સુરક્ષિત નથી.
- અન્ય વપરાશકર્તાઓ સાથે ફાઇલો અથવા પાસવર્ડ શેર કરતી વખતે સાવચેત રહો. તમે જુદા પાસવર્ડ્સ અણધાર્યા વપરાશકર્તાઓના હાથમાં આવી જવાનું જોખમ ચલાવો છો. યાદ રાખો કે પાસવર્ડ વડે ફાઇલને લોક કરવાથી તમારી ફાઇલને દૂષિત ઉદ્દેશ્યથી સુરક્ષિત કરવી જરૂરી નથી.

એક્સેલ ટિપ્સ

- ડેટાને ઓળખવા અને સમજવા માટે પીવટ કોષ્ટકોનો ઉપયોગ કરો.
- એક કરતાં વધુ પંક્તિ અથવા કૉલમ ઉમેરો.
- તમારા ડેટાને સરળ બનાવવા માટે ફિલ્ટર્સનો ઉપયોગ કરો.
- ડુપ્લિકેટ ડેટા પોઈન્ટ અથવા સેટ્સ દૂર કરો.
- પંક્તિઓને કૉલમમાં સ્થાનાંતરિત કરો.
- કૉલમ વચ્ચે ટેક્સ્ટ માહિતીને વિભાજિત કરો.
- સરળ ગણતરીઓ માટે આ સૂત્રોનો ઉપયોગ કરો.
- તમારા કોષોમાં સંખ્યાઓની સરેરાશ મેળવો.
- ડેટાના આધારે કોષોને આપમેળે રંગ બદલવા માટે શરતી ફોર્મેટિંગનો ઉપયોગ કરો.

- થોકકસ એક્સેલ કાર્યોને સ્વચાલિત કરવા માટે IF Excel સૂત્રનો ઉપયોગ કરો.
- એક કોષનું સૂત્ર એકસરખું રાખવા માટે ડોલરના ચિહ્નોનો ઉપયોગ કરો, તે ગમે ત્યાં ફરે છે.
- શીટના એક વિસ્તારમાંથી બીજા વિસ્તારમાં ડેટા ખેંચવા માટે VLOOK-UP ફંક્શનનો ઉપયોગ કરો.
- આડી સ્તંભોમાંથી ડેટા ખેંચવા માટે INDEX અને MATCH ફોર્મ્યુલાનો ઉપયોગ કરો.
- એક્સેલને કોષોની કોઈપણ શ્રેણીમાં શબ્દો અથવા સંખ્યાઓની ગણતરી કરવા માટે COUNTIF ફંક્શનનો ઉપયોગ કરો.
- એમ્પરસેન્ડનો ઉપયોગ કરીને કોષોને જોડો.
- ચેકબોક્સ ઉમેરો
- વેબસાઈટ પર સેલને હાઈપરલિંક કરો.
- ડ્રોપ ડાઉન મેનુ ઉમેરો.
- ચિત્રકાર ફોર્મેટનો ઉપયોગ કરો

પીવટ ટેબલ

સારાંશ ડેટાનો સારાંશ, વિશ્લેષણ, અન્વેષણ અને પ્રસ્તુત કરવા માટે તમે PivotTable નો ઉપયોગ કરી શકો છો. PivotCharts PivotTable માં સારાંશ ડેટામાં વિઝ્યુલાઇઝેશન ઉમેરીને PivotTables ને પૂરક બનાવે છે અને તમને સરખામણીઓ, પેટર્ન અને વલણો સરળતાથી જોવાની મંજૂરી આપે છે. PivotTables અને PivotCharts બંને તમને તમારા એન્ટરપ્રાઇઝમાં નિર્ણાયક ડેટા વિશે જાણકાર નિર્ણયો લેવા સક્ષમ કરે છે. તમે બાહ્ય ડેટા સ્ત્રોતો જેમ કે SQL સર્વર કોષ્ટકો, SQL સર્વર વિશ્લેષણ સેવાઓ ક્યુબ્સ, એઝ્યુર માર્કેટપ્લેસ, ઓફિસ ડેટા કનેક્શન (.odc) ફાઇલો, XML ફાઇલો, એક્સેસ ડેટાબેસેસ અને ટેક્સ્ટ ફાઇલો સાથે પણ કનેક્ટ કરી શકો છો. નવા કોષ્ટકો બનાવો.

PivotTable એ મોટી માત્રામાં ડેટાને ઝડપથી સારાંશ આપવા માટે એક ઇન્ટરેક્ટિવ રીત છે. આંકડાકીય માહિતીનું વિગતવાર વિશ્લેષણ કરવા અને તમારા ડેટા વિશે અણધાર્યા પ્રશ્નોના જવાબ આપવા માટે તમે PivotTable નો ઉપયોગ કરી શકો છો. પીવટ ટેબલ ખાસ કરીને આ માટે રચાયેલ છે:

- ઘણી વપરાશકર્તા-મૈત્રીપૂર્ણ રીતે મોટી માત્રામાં ડેટાની ક્વેરી કરવી.
- સબટોટલીંગ અને આંકડાકીય ડેટાને એકત્ર કરવા, શ્રેણીઓ અને ઉપકેટેગરીઝ દ્વારા ડેટાનો સારાંશ, અને કસ્ટમ ગણતરીઓ અને સૂત્રો બનાવવા.
- તમારા પરિણામો પર ધ્યાન કેન્દ્રિત કરવા માટે ડેટાના સ્તરને વિસ્તરણ અને સંકુચિત કરવું, અને તમને રુચિના ક્ષેત્રો માટે સારાંશ ડેટામાંથી વિગતો માટે ડ્રિલિંગ.
- સ્ત્રોત ડેટાના વિવિધ સારાંશ જોવા માટે પંક્તિઓને કૉલમમાં અથવા કૉલમ્સને પંક્તિઓમાં ખસેડવી (અથવા “પીવટિંગ”).
- ફિલ્ટરિંગ, સોર્ટિંગ, ગ્રૂપિંગ અને શરતી રીતે ડેટાના સૌથી ઉપયોગી અને રસપ્રદ સબસેટનું ફોર્મેટ કરવું તમને ફક્ત તમને જોઈતી માહિતી પર ધ્યાન કેન્દ્રિત કરવા સક્ષમ બનાવે છે.
- સંક્ષિપ્ત, આકર્ષક અને ટીકાવાળા ઓનલાઈન અથવા મુદ્રિત અહેવાલો પ્રસ્તુત કરવા.

ઉદાહરણ તરીકે, અહીં ડાબી બાજુએ ઘરગથ્થુ ખર્ચાઓની એક સરળ સૂચિ છે, અને જમણી બાજુની સૂચિ પર આધારિત પીવટ ટેબલ છે:

પિવટ ટેબલ સાથે કામ કરવાની રીતો

તમે PivotTable તેના ડેટા સ્ત્રોતને પસંદ કરીને, PivotTable ફીલ્ડ લિસ્ટમાં ફીલ્ડ્સ ગોઠવીને અને પ્રારંભિક લેઆઉટ પસંદ કરીને બનાવ્યા પછી, તમે PivotTable સાથે કામ કરતા હોવ ત્યારે તમે નીચેના કાર્યો કરી શકો છો:

નીચે મુજબ કરીને ડેટાનું અન્વેષણ કરો

- ડેટાને વિસ્તૃત અને સંકુચિત કરો, અને મૂલ્યોથી સંબંધિત અંતર્ગત વિગતો દર્શાવો.
- સોર્ટ કરો, ફિલ્ટર કરો અને ગ્રૂપ ફીલ્ડ્સ અને વસ્તુઓ.
- સારાંશ કાર્યો બદલો, અને કસ્ટમ ગણતરીઓ અને સૂત્રો ઉમેરો.

નીચે પ્રમાણે કરીને ફોર્મ લેઆઉટ અને ફીલ્ડની ગોઠવણી બદલો

- PivotTable ફોર્મ બદલો: કોમ્પેક્ટ, આઉટલાઈન અથવા ટેબ્યુલર.
- ફીલ્ડ ઉમેરો, ફરીથી ગોઠવો અને દૂર કરો.
- ક્ષેત્રો અથવા વસ્તુઓનો ક્રમ બદલો.

નીચે પ્રમાણે કરીને કૉલમ, પંક્તિઓ અને સબટોટલનું લેઆઉટ બદલો

- કૉલમ અને પંક્તિ ફીલ્ડ હેડરને ચાલુ અથવા બંધ કરો અથવા ખાલી રેખાઓ દર્શાવો અથવા છુપાવો.
- તેમની પંક્તિઓ ઉપર અથવા નીચે સબટોટલ દર્શાવો.
- તાજું કરવા પર કૉલમની પહોળાઈને સમાયોજિત કરો.
- કૉલમ ફીલ્ડને પંક્તિ ક્ષેત્રમાં અથવા પંક્તિ ક્ષેત્રને કૉલમ વિસ્તારમાં ખસેડો.
- બાહ્ય પંક્તિ અને કૉલમ વસ્તુઓ માટે કોષોને મજબૂત અથવા અનમજબૂત કરો.

નીચે પ્રમાણે કરીને ખાલી જગ્યાઓ અને ભૂલોના પ્રદર્શનને બદલો

- ભૂલો અને ખાલી કોષો કેવી રીતે પ્રદર્શિત થાય છે તે બદલો.
- ડેટા વગરની વસ્તુઓ અને લેબલ્સ કેવી રીતે બતાવવામાં આવે છે તે બદલો.
- ખાલી પંક્તિઓ દર્શાવો અથવા છુપાવો

નીચે પ્રમાણે કરીને ફોર્મેટ બદલો

- મેન્યુઅલી અને શરતી રીતે કોષો અને રેન્જને ફોર્મેટ કરો.
- સમગ્ર PivotTable ફોર્મેટ શૈલી બદલો.

- ક્ષેત્રો માટે નંબર ફોર્મેટ બદલો.
- OLAP સર્વર ફોર્મેટિંગ શામેલ કરો.

પીવટ ટેબલ અને પીવટ ચાર્ટ બનાવો

તમે PivotTable અથવા PivotChart માટે આધાર તરીકે Excel વર્કશીટમાંથી ડેટાનો ઉપયોગ કરી શકો છો. ડેટા યાદી ફોર્મેટમાં હોવો જોઈએ, જેમાં પ્રથમ પંક્તિમાં કૉલમ લેબલ હોવા જોઈએ, જેનો એક્સેલ ક્ષેત્રના નામો માટે ઉપયોગ કરશે. અનુગામી પંક્તિઓના દરેક કોષમાં તેના કૉલમ મથાળાને અનુરૂપ ડેટા હોવો જોઈએ અને તમારે સમાન કૉલમમાં ડેટા પ્રકારોને મિશ્રિત ન કરવા જોઈએ. દાખલા તરીકે, તમારે એક જ કૉલમમાં ચલણ મૂલ્યો અને તારીખોને મિશ્રિત કરવી જોઈએ નહીં. વધુમાં, ડેટા રેન્જમાં કોઈપણ ખાલી પંક્તિઓ અથવા કૉલમ્સ ન હોવા જોઈએ.

એક્સેલ કોષ્ટકો પહેલેથી જ સૂચિ ફોર્મેટમાં છે અને PivotTable સ્રોત ડેટા માટે સારા ઉમેદવારો છે. જ્યારે તમે PivotTable ને તાજું કરો છો, ત્યારે Excel કોષ્ટકમાંથી નવો અને અપડેટ થયેલ ડેટા રિફ્રેશ ઓપરેશનમાં આપમેળે સમાવવામાં આવે છે.

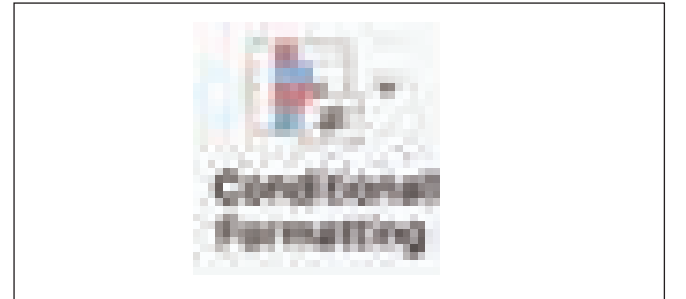
ગતિશીલ નામવાળી શ્રેણીનો ઉપયોગ કરવો : PivotTable ને અપડેટ કરવાનું સરળ બનાવવા માટે, તમે ગતિશીલ નામવાળી શ્રેણી બનાવી શકો છો અને તે નામનો ઉપયોગ PivotTable ના ડેટા સ્રોત તરીકે કરી શકો છો. જો નામની શ્રેણી વધુ ડેટાનો સમાવેશ કરવા માટે વિસ્તરે છે, તો PivotTable ને તાજું કરવાથી નવો ડેટા શામેલ થશે.

સરવાળો સહિત : એક્સેલ પીવટ ટેબલમાં આપમેળે પેટાટોટલ અને ગ્રાન્ડ ટોટલ બનાવે છે. જો સ્રોત ડેટામાં ઓટોમેટિક પેટાટોટલ અને ગ્રાન્ડ ટોટલનો સમાવેશ થાય છે જે તમે ડેટા ટેબલ પર આઉટલાઈન ગ્રૂપમાં સબટોટલ આદેશનો ઉપયોગ કરીને બનાવેલ છે, તો તમે PivotTable બનાવતા પહેલા પેટાટોટલ અને ગ્રાન્ડ ટોટલ્સને દૂર કરવા માટે તે જ આદેશનો ઉપયોગ કરો.

શરતી ફોર્મેટિંગ

શરતી ફોર્મેટિંગ લાગુ કરો

- 1 કોષોની શ્રેણી, કોષ્ટક અથવા આખી શીટ પસંદ કરો કે જેના પર તમે શરતી ફોર્મેટિંગ લાગુ કરવા માંગો છો.
- 2 હોમ ટેબ પર, કન્ડીશનલ ફોર્મેટિંગ પર ક્લિક કરો.
- 3 નીચેનામાંથી એક કરો:



પ્રકાશિત કરવા માટે	આ કરવું
ચોક્કસ કોષોમાં મૂલ્યો. ઉદાહરણો આ અઠવાડિયા પછીની તારીખો અથવા 50 અને 100 ની વચ્ચેની સંખ્યાઓ અથવા સ્કોરના નીચેના 10% છે.	કોષોના નિયમો અથવા ટોપ/બોટમ રૂલ્સને હાઈલાઈટ કરવા માટે પોઈન્ટ કરો અને પછી યોગ્ય વિકલ્પ પર ક્લિક કરો.
કોષ શ્રેણીમાં મૂલ્યોનો સંબંધ. સમગ્ર કોષમાં રંગના બેન્ડને વિસ્તૃત કરે છે. ઉદાહરણો સૌથી મોટા શહેરોમાં કિંમતો અથવા વસ્તીની તુલના છે.	ડેટા બાર તરફ નિર્દેશ કરો, અને પછી તમે ઇચ્છો તે ભરણ પર ક્લિક કરો.
કોષ શ્રેણીમાં મૂલ્યોનો સંબંધ. રંગ સ્કેલ લાગુ કરે છે જ્યાં કોષના રંગની તીવ્રતા શ્રેણીની ટોચ અથવા નીચે તરફના મૂલ્યના પ્લેસમેન્ટને પ્રતિબિંબિત કરે છે. એક ઉદાહરણ સમગ્ર પ્રદેશોમાં વેચાણ વિતરણ છે.	રંગ ભીંગડા તરફ નિર્દેશ કરો, અને પછી તમને જોઈતા સ્કેલ પર ક્લિક કરો.
એક કોષ શ્રેણી જેમાં મૂલ્યોના ત્રણથી પાંચ જૂથો હોય છે, જ્યાં દરેક જૂથની પોતાની શ્રેણી હોય છે. ઉદાહરણ તરીકે, તમે \$80,000 થી નીચે, \$60,000 થી નીચે અને \$40,000 થી ઓછા વેચાણને પ્રતિબિંબિત કરતા કોષોને પ્રકાશિત કરવા માટે ત્રણ ચિહ્નોનો સમૂહ અસાઈન કરી શકો છો. અથવા તમે ઓટોમોબાઈલ માટે 5-પોઈન્ટ રેટિંગ સિસ્ટમ અસાઈન કરી શકો છો અને પાંચ આઈકોન્સનો સેટ લાગુ કરી શકો છો.	આઈકન સેટ્સ તરફ નિર્દેશ કરો અને પછી સેટ પર ક્લિક કરો.

પાવર ક્વેરીઝ

પાવર ક્વેરી (જે એક્સેલમાં ગેટ એન્ડ ટ્રાન્સફોર્મ તરીકે ઓળખાય છે) સાથે, તમે બાહ્ય ડેટાને આયાત કરી શકો છો અથવા તેને કનેક્ટ કરી શકો છો, અને પછી તે ડેટાને આકાર આપી શકો છો, ઉદાહરણ તરીકે કોલમ દૂર કરો, ડેટા પ્રકાર બદલો અથવા કોષ્ટકોને મર્જ કરો, જે તમારી જરૂરિયાતોને પૂર્ણ કરે છે. પછી, તમે ચાર્ટ અને રિપોર્ટ્સ બનાવવા માટે તમારી ક્વેરી એક્સેલમાં લોડ કરી શકો છો. સમયાંતરે, તમે ડેટાને અદ્યતન બનાવવા માટે તેને તાજું કરી શકો છો. પાવર ક્વેરી ત્રણ એક્સેલ એપ્લિકેશન પર ઉપલબ્ધ છે, વિન્ડોઝ માટે એક્સેલ, મેક માટે એક્સેલ અને વેબ માટે એક્સેલ. પાવર ક્વેરી હેલ્પ વિષયોના સારાંશ માટે, એક્સેલ હેલ્પ માટે પાવર ક્વેરી જુઓ.

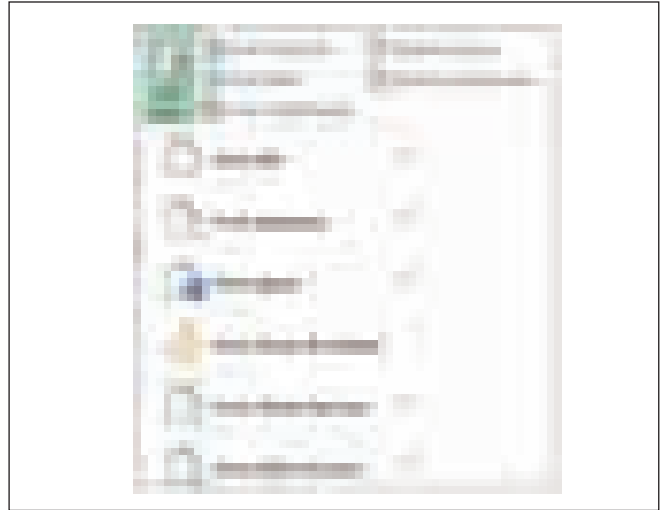
પાવર ક્વેરીનાં ચાર તબક્કા

પાવર ક્વેરીનો ઉપયોગ કરવાના ચાર તબક્કા છે.



- 1 કનેક્ટ કરો: ક્લાઉડમાં, સેવા પર અથવા સ્થાનિક રીતે ડેટા સાથે જોડાણો બનાવો
 - 2 ટ્રાન્સફોર્મ: તમારી જરૂરિયાતોને પૂર્ણ કરવા માટે ડેટાને આકાર આપો, જ્યારે મૂળ સ્ત્રોત યથાવત રહે છે
 - 3 ભેગું કરો: ડેટામાં અનન્ય દૃશ્ય મેળવવા માટે બહુવિધ સ્ત્રોતોમાંથી ડેટાને એકીકૃત કરો
 - 4 લોડ કરો: તમારી ક્વેરી પૂર્ણ કરો અને તેને વર્કશીટ અથવા ડેટા મોડલમાં લોડ કરો અને સમયાંતરે તેને રિફ્રેશ કરો.
- નીચેના વિભાગો દરેક તબક્કાને વધુ વિગતવાર શોધે છે.

કનેક્ટ કરો : તમે એકલ ડેટા સ્ત્રોત, જેમ કે એક્સેલ વર્કબુક, અથવા ક્લાઉડમાં ફેલાયેલા બહુવિધ ડેટાબેસેસ, ફીડ્સ અથવા સેવાઓ પર આયાત કરવા માટે પાવર ક્વેરીનો ઉપયોગ કરી શકો છો. ડેટા સ્ત્રોતોમાં વર્તમાન વર્કબુકમાં વેબ, ફાઈલો, ડેટાબેસેસ, Azure અથવા તો Excel કોષ્ટકોનો સમાવેશ થાય છે. પાવર ક્વેરી સાથે, પછી તમે તમારા પોતાના અનન્ય રૂપાંતરણો અને સંયોજનોનો ઉપયોગ કરીને તે તમામ ડેટા સ્ત્રોતોને એકસાથે લાવી શકો છો જેથી તમે અન્યથા જોઈ ન હોત તેવી આંતરદૃષ્ટિને ઉજાગર કરી શકો.



એકવાર આયાત કર્યા પછી, તમે બાહ્ય ડેટા સ્ત્રોતમાંથી ઉમેરાઓ, ફેરફારો અને કાઢી નાખવા માટે ડેટાને તાજું કરી શકો છો. વધુ માહિતી માટે, જુઓ એક્સેલમાં એક્સટર્નલ ડેટા કનેક્શન રિફ્રેશ કરો.

પરિવર્તન

ડેટાને રૂપાંતરિત કરવાનો અર્થ છે કે તમારી ડેટા વિશ્લેષણની જરૂરિયાતોને પૂર્ણ કરવા માટે તેને અમુક રીતે સંશોધિત કરવું. ઉદાહરણ તરીકે, તમે કોલમ

દૂર કરી શકો છો, ડેટા પ્રકાર બદલી શકો છો અથવા પંક્તિઓ ફિલ્ટર કરી શકો છો. આમાંની દરેક કામગીરી ડેટા ટ્રાન્સફોર્મેશન છે. ડેટાના એક અથવા વધુ સેટમાં રૂપાંતરણો (અને સંયોજન) લાગુ કરવાની આ પ્રક્રિયાને ડેટા શેપિંગ પણ કહેવામાં આવે છે.

આ રીતે વિચારો. ફૂલદાની માટીના ગઢા તરીકે શરૂ થાય છે જે વ્યવહારિક અને સુંદર કંઈક બનાવે છે. ડેટા સમાન છે. તેને તમારી જરૂરિયાતો માટે યોગ્ય એવા ટેબલમાં આકાર આપવાની જરૂર છે અને જે આકર્ષક રિપોર્ટ્સ અને ડેશબોર્ડ્સને સક્ષમ કરે છે.

પાવર ક્વેરી ડેટા ટ્રાન્સફોર્મેશનને સરળ બનાવવા અને પ્રદર્શિત કરવા માટે પાવર ક્વેરી એડિટર નામની સમર્પિત વિન્ડોનો ઉપયોગ કરે છે. તમે Get & Transform Data ગ્રુપમાં Get Data કમાન્ડમાંથી લોન્ચ ક્વેરી એડિટર પસંદ કરીને પાવર ક્વેરી એડિટર ખોલી શકો છો, પરંતુ જ્યારે તમે ડેટા સ્ત્રોત સાથે કનેક્ટ કરો છો, નવી ક્વેરી બનાવો છો અથવા ક્વેરી લોડ કરો છો ત્યારે પણ તે ખુલે છે.

પાવર ક્વેરી એડિટર તમે ડેટા પર લાગુ કરો છો તે દરેક ટ્રાન્સફોર્મેશન અથવા સ્ટેપને રેકૉર્ડ કરીને અને લેબલ કરીને તમે ડેટા સાથે કરો છો તે દરેક



વસ્તુનો ટ્રેક રાખે છે. ભલે ટ્રાન્સફોર્મેશન ડેટા કનેક્શન હોય, કૉલમ રિમૂવલ, મર્જ અથવા ડેટા પ્રકાર ફેરફાર હોય, તમે ક્વેરી સેટિંગ્સ ફલકના લાગુ પગલાં વિભાગમાં દરેક ટ્રાન્સફોર્મેશનને જોઈ અને સંશોધિત કરી શકો છો.

યુઝર ઈન્ટરફેસમાંથી તમે ઘણા પરિવર્તનો કરી શકો છો. દરેક પરિવર્તન પૃષ્ઠભૂમિમાં એક પગલા તરીકે રેકૉર્ડ કરવામાં આવે છે. તમે એડવાન્સ્ડ એડિટરમાં પાવર ક્વેરી એમ લેંગ્વેજનો ઉપયોગ કરીને તમારા પોતાના પગલાંને સંશોધિત અને લખી શકો છો.

તમે તમારા ડેટા કનેક્શન્સ પર લાગુ કરો છો તે તમામ પરિવર્તનો સામૂહિક રીતે ક્વેરી બનાવે છે, જે મૂળ (અને અપરિવર્તિત) ડેટા સ્ત્રોતનું નવું પ્રતિનિધિત્વ છે. જ્યારે તમે ક્વેરી રિફ્રેશ કરો છો, ત્યારે દરેક પગલું આપમેળે ચાલે છે. એક્સેલમાં ડેટાને મેન્યુઅલી કનેક્ટ કરવાની અને તેને આકાર આપવાની જરૂરિયાતને ક્વેરીઝ બદલે છે.

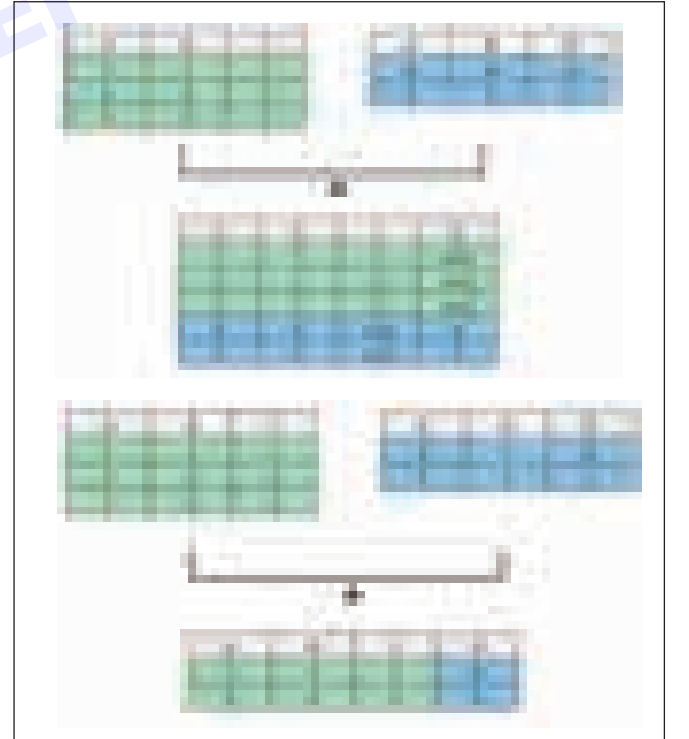
ભેગું કરો : તમે તમારી એક્સેલ વર્કબુકમાં બહુવિધ પ્રશ્નોને જોડીને અથવા મર્જ કરીને જોડી શકો છો. એપેન્ડ અને મર્જ ઓપરેશન્સ કોઈપણ ક્વેરી પર ટેબ્યુલર આકાર સાથે કરવામાં આવે છે અને તે ડેટા સ્ત્રોતોથી સ્વતંત્ર છે જેમાંથી ડેટા આવે છે.

એપેન્ડ : એપેન્ડ ઓપરેશન નવી ક્વેરી બનાવે છે જેમાં પ્રથમ ક્વેરીમાંથી બધી પંક્તિઓ અને બીજી ક્વેરીમાંથી બધી પંક્તિઓ હોય છે. તમે બે પ્રકારની એપેન્ડ કામગીરી કરી શકો છો:

- **ઈન્ટરમીડિયેટ એપેન્ડ :** દરેક એપેન્ડ ઓપરેશન માટે નવી ક્વેરી બનાવે છે.
- **ઈનલાઈન એપેન્ડ :** જ્યાં સુધી તમે અંતિમ પરિણામ પર ન પહોંચો ત્યાં સુધી તમારી હાલની ક્વેરી સાથે ડેટા જોડે છે.

મર્જ કરો : મર્જ ઓપરેશન હાલની બે ક્વેરીમાંથી નવી ક્વેરી બનાવે છે. આ એક ક્વેરી પ્રાથમિક કોષ્ટકની તમામ કૉલમ ધરાવે છે, જેમાં એક કૉલમ સંબંધિત કોષ્ટકની નેવિગેશન લિંક તરીકે સેવા આપે છે. સંબંધિત કોષ્ટકમાં

પ્રાથમિક કોષ્ટકમાં સામાન્ય કૉલમ મૂલ્યમાંથી દરેક પંક્તિ સાથે મેળ ખાતી બધી પંક્તિઓ શામેલ છે. વધુમાં, તમે પ્રાથમિક કોષ્ટકમાં સંબંધિત કોષ્ટકમાંથી કૉલમને વિસ્તૃત અથવા ઉમેરી શકો છો.



લોડ

તમારી વર્કબુકમાં પ્રશ્નો લોડ કરવાની બે મુખ્ય રીતો છે:

- પાવર ક્વેરી એડિટરમાંથી, તમે હોમ ટેબ પર ક્લોઝ ગ્રુપમાં ક્લોઝ અને લોડ આદેશોનો ઉપયોગ કરી શકો છો.

- એક્સેલ વર્કબુક ક્વેરીઝ પેનમાંથી (ક્વેરીઝ અને કનેક્શન પસંદ કરો), તમે ક્વેરી પર રાઈટ-ક્લિક કરી શકો છો અને લોડ ટુ પસંદ કરી શકો છો.

તમે તમારા ડેટાને કેવી રીતે જોવા માંગો છો અને તમે ડેટા ક્યાં લોડ કરવા માંગો છો તે પસંદ કરવા માટે ક્વેરી ઓપ્શન્સ ડાયલોગ બોક્સ (ફાઈલ પસંદ કરો > વિકલ્પો અને સેટિંગ્સ > ક્વેરી ઓપ્શન્સ) નો ઉપયોગ કરીને તમે તમારા લોડ વિકલ્પોને ફાઈન ટ્યુન કરી શકો છો, ક્યાં તો વર્કશીટમાં અથવા ડેટા મોડલ (જે વર્કબુકમાં રહેલ બહુવિધ કોષ્ટકોનો રિલેશનલ ડેટા સ્ત્રોત છે).

એક્સેલમાં પાવર ક્વેરી માટે એલિકેશન સપોર્ટ

દસ વર્ષથી, પાવર ક્વેરી વિન્ડોઝ માટે એક્સેલ પર સપોર્ટેડ છે. હવે, એક્સેલ મેક માટે એક્સેલ પર પાવર ક્વેરી સપોર્ટને વિસ્તૃત કરી રહ્યું છે અને વેબ માટે એક્સેલમાં સપોર્ટ ઉમેરી રહ્યું છે. આનો અર્થ એ છે કે અમે પાવર

ક્વેરી ત્રણ મુખ્ય પ્લેટફોર્મ પર ઉપલબ્ધ કરાવી રહ્યા છીએ અને એક્સેલ ગ્રાહકોમાં પાવર ક્વેરી ની લોકપ્રિયતા અને કાર્યક્ષમતા દર્શાવે છે. Microsoft 365 રોડમેપ પર ભાવિ ઘોષણાઓ માટે જુઓ અને Microsoft 365 માટે Excel માં નવું શું છે.

એક્સેલમાં પાવર ક્વેરીનો ઇતિહાસ

ગેટ એન્ડ ટ્રાન્સફોર્મ ડેટા (હવે પાવર ક્વેરી તરીકે ઓળખાય છે) નું એક્સેલમાં એકીકરણ વર્ષોથી ઘણા ફેરફારોમાંથી પસાર થયું છે.

Windows માટે એક્સેલ 2010 અને 2013

Windows માટે એક્સેલ 2010 માં, અમે સૌપ્રથમ પાવર ક્વેરી રજૂ કરી હતી અને તે મફત એડ-ઇન તરીકે ઉપલબ્ધ હતી જે અહીંથી ડાઉનલોડ કરી શકાય છે: પાવર ક્વેરી એડિન ડાઉનલોડ કરો. એકવાર સક્ષમ થઈ ગયા પછી, પાવર ક્વેરી કાર્યક્ષમતા રિબન પરના પાવર ક્વેરી ટેબમાંથી ઉપલબ્ધ હતી.



Microsoft 365 : અમે ડેટાને આયાત કરવા અને સાફ કરવા માટે એક્સેલમાં પ્રાથમિક અનુભવ તરીકે પાવર ક્વેરી અપડેટ કરી છે. તમે એક્સેલ રિબનના ડેટા ટેબ પર ડેટા ગેટ એન્ડ ટ્રાન્સફોર્મ ગ્રૂપમાંથી પાવર ક્વેરી ડેટા આયાત વિઝાર્ડ્સ અને ટૂલ્સને એક્સેસ કરી શકો છો.

આ અનુભવમાં ઉત્તર ડેટા આયાત કાર્યક્ષમતા, ડેટા ટેબ પર ફરીથી ગોઠવાયેલા આદેશો, નવી ક્વેરીઝ અને કનેક્શન બાજુની તકતી, અને સોર્ટ કરીને, ડેટા પ્રકારો બદલીને, કોલમને વિભાજિત કરવા, ડેટાને એકત્રિત કરીને શક્તિશાળી રીતે ડેટાને આકાર આપવાની સતત ક્ષમતાનો સમાવેશ થાય છે.



આ નવા અનુભવે ગેટ એક્સર્ટર્નલ ડેટા ગ્રૂપમાં ડેટા કમાન્ડ હેઠળ જૂના, લેગસી ડેટા આયાત વિઝાર્ડ્સને પણ બદલ્યા છે. જો કે, તેઓ હજુ પણ એક્સેલ ઓપ્શન્સ ડાયલોગ બોક્સમાંથી એક્સેસ કરી શકાય છે (ફાઈલ પસંદ કરો > વિકલ્પો > ડેટા > લેગસી ડેટા આયાત વિઝાર્ડ બતાવો).

Windows માટે એક્સેલ 2016 અને 2019

અમે Microsoft 365ની જેમ પાવર ક્વેરી ટેકનોલોજીના આધારે ડેટા મેળવો અને ટ્રાન્સફોર્મ કરવાનો અનુભવ ઉમેર્યો છે.

Mac માટે Microsoft 365 માટે એક્સેલ

2019 માં અમે Mac માટે Excel માં પાવર ક્વેરીનું સમર્થન કરવાની યાત્રા શરૂ કરી. ત્યારથી, અમે TXT, CSV, XLSX, JSON અને XML ફાઈલોમાંથી પાવર ક્વેરી ક્વેરી રિફ્રેશ કરવાની ક્ષમતા ઉમેરી છે. અમે વર્તમાન વર્કબુકમાં SQL સર્વર અને કોષ્ટકો અને શ્રેણીઓમાંથી ડેટા રિફ્રેશ કરવાની ક્ષમતા પણ ઉમેરી છે.

ઑક્ટોબર 2019માં, અમે વર્તમાન પાવર ક્વેરી ક્વેરી રિફ્રેશ કરવાની અને નવી ક્વેરી બનાવવા અને સંપાદિત કરવા માટે VBA નો ઉપયોગ કરવાની ક્ષમતા ઉમેરી છે.

2021ના જાન્યુઆરીમાં, અમે OData અને SharePoint સ્ત્રોતોમાંથી પાવર ક્વેરી ક્વેરી રિફ્રેશ કરવા માટે સમર્થન ઉમેર્યું.

વધુ માહિતી માટે, Mac માટે Excel માં પાવર ક્વેરીનો ઉપયોગ કરો જુઓ.

નોંધ: Mac માટે Excel 2016 અને Excel 2019 પર પાવર ક્વેરી માટે કોઈ સમર્થન નથી.

ડેટા કેટલોગ અવમૂલ્યન

ડેટા કેટલોગ સાથે, તમે તમારી શેર કરેલી ક્વેરીઝ જોઈ શકો છો, અને પછી વર્તમાન વર્કબુકમાં લોડ કરવા, સંપાદિત કરવા અથવા અન્યથા ઉપયોગ કરવા માટે તેમને પસંદ કરી શકો છો. આ સુવિધા ધીમે ધીમે નાપસંદ કરવામાં આવી હતી:

- 1લી ઓગસ્ટ, 2018 ના રોજ, અમે નવા ગ્રાહકોને ડેટા કેટલોગમાં ઓનબોર્ડ કરવાનું બંધ કર્યું.
- 3જી ડિસેમ્બર, 2018ના રોજ, વપરાશકર્તાઓ ડેટા કેટલોગમાં નવી અથવા અપડેટ કરેલી ક્વેરી શેર કરી શક્યા નથી.

- 4થી માર્ચ, 2019ના રોજ, ડેટા કેટલોગ કામ કરવાનું બંધ કરી દીધું. આ તારીખ પછી, અમે તમારી શેર કરેલી ક્વેરીઝ ડાઉનલોડ કરવાની ભલામણ કરી છે જેથી કરીને તમે માય ડેટા કેટલોગ ક્વેરીઝ ટાસ્ક પેનમાંથી ઓપન વિકલ્પનો ઉપયોગ કરીને ડેટા કેટલોગની બહાર તેનો ઉપયોગ કરવાનું ચાલુ રાખી શકો.

પાવર ક્વેરી નાપસંદમાં ઉમેરો

2019 ના ઉનાળાની શરૂઆતમાં, અમે પાવર ક્વેરી એડને સત્તાવાર રીતે નાપસંદ કરી દીધું હતું જેમાં Windows માટે Excel 2010 અને 2013 માટે જરૂરી છે. સૌજન્ય તરીકે, તમે હજી પણ એડ-ઇનનો ઉપયોગ કરી શકો છો, પરંતુ આ પછીની તારીખે બદલાઈ શકે છે.

© NIMI
NOT TO BE REPUBLISHED

અદ્યતન ફોર્મ્યુલા અને મેક્રો બનાવો (E-Commerce scope and benefits)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ઈ કોમર્સ વ્યાખ્યાયિત કરો
- પરંપરાગત વાણિજ્ય અને ઈ કોમર્સ વચ્ચેનો તફાવત સમજાવો
- ઈ-કોમર્સનો પ્રકાર, અવકાશ અને લાભ સમજાવો.

ઈ કોમર્સ

ઈ કોમર્સ એ ઉત્પાદનો અથવા સેવાની ખરીદી અને વેચાણ અને ઇલેક્ટ્રોનિક રીતે ભંડોળના ટ્રાન્સફરની પ્રક્રિયા છે.

ઇલેક્ટ્રોનિક કોમર્સ મોબાઇલ કોમર્સ, ઇલેક્ટ્રોનિક ફંડ ટ્રાન્સફર, સપ્લાય ચેઇન મેનેજમેન્ટ, ઇન્ટરનેટ માર્કેટિંગ, ઓનલાઇન ટ્રાન્ઝેક્શન પ્રોસેસિંગ, ઇલેક્ટ્રોનિક ડેટા ઇન્ટરચેન્જ (EDI), ઇન્વેન્ટરી મેનેજમેન્ટ સિસ્ટમ્સ અને ઓટોમેટેડ ડેટા કલેક્શન સિસ્ટમ્સ જેવી ટેકનોલોજીઓ પર ડ્રો કરે છે. આધુનિક ઇલેક્ટ્રોનિક વાણિજ્ય સામાન્ય રીતે ટ્રાન્ઝેક્શનના જીવન ચક્રના ઓછામાં ઓછા એક ભાગ માટે વર્લ્ડ વાઇડ વેબનો ઉપયોગ કરે છે, જોકે તે ઈ-મેલ જેવી અન્ય તકનીકોનો પણ ઉપયોગ કરી શકે છે.

ઈ-કોમર્સ વ્યવસાયો સામાન્ય રીતે નીચેની કેટલીક અથવા બધી પદ્ધતિઓનો ઉપયોગ કરે છે:

ઓનલાઇન કેટલોગ સાથેની વેબસાઇટો પર Etail અથવા “વર્ચ્યુઅલ સ્ટોરફ્રન્ટ્સ” પ્રદાન કરો, કેટલીકવાર “વર્ચ્યુઅલ મોલ” માં ભેગા થાય છે.

વેબસાઇટ્સ અથવા ઓનલાઇન બજાર સ્થાનો પર ખરીદો અથવા વેચો.

વેબ સંપર્કો અને સોશિયલ મીડિયા દ્વારા વસ્તી વિષયક ડેટા એકત્રિત કરો અને તેનો ઉપયોગ કરો.

ઇલેક્ટ્રોનિક ડેટા ઇન્ટરચેન્જનો ઉપયોગ કરો, ડેટાનું બિઝનેસ-ટુ-બિઝનેસ એક્સચેન્જ.

ઈ-મેલ અથવા ફેક્સ દ્વારા સંભવિત અને સ્થાપિત ગ્રાહકો સુધી પહોંચો (ઉદાહરણ તરીકે, ન્યૂઝલેટર્સ સાથે).

વેપાર-ધંધાના ખરીદ-વેચાણનો ઉપયોગ કરો.

સુરક્ષિત વ્યવસાયિક વ્યવહારો પ્રદાન કરો.

ઑક્ટોબર 6, 2014 ના રોજ ફ્લિપકાર્ટના બિગ બિલિયન ડે સેલને તમામ ઓનલાઇન વેચાણની માતા તરીકે ઓળખવામાં આવ્યું હતું, પરંતુ મોટા દિવસે, વસ્તુઓ અવ્યવસ્થિત થઈ ગઈ હતી અને ગ્રાહકોને વધુ અને શુષ્ક લાગણી થઈ હતી. જ્યારે સ્થાપકોએ એ હકીકત વિશે તાત્કાલિક માફી માંગી કે તેઓ પૂરતા પ્રમાણમાં તૈયાર ન હતા, હકીકત એ છે કે આપણું એક રાષ્ટ્ર છે જે મોટા પાયે ઈકોમર્સ તરફ આગળ વધી રહ્યું છે.

આના જેવી એક પણ ઘટના વિસ્ફોટક વૃદ્ધિ માટે તૈયાર \$3 બિલિયન સેક્ટરમાં રસ ઓછો કરવા માટે બહુ કામ કરશે નહીં. ફ્લિપકાર્ટના સ્પર્ધકો જેમ કે એમેઝોન અને સ્નેપડીલ પહેલેથી જ આ અનુભવમાંથી શીખ્યા છે અને કદાચ ભવિષ્યમાં આવી ઘટનાઓનું પુનરાવર્તન ન કરવાનો પ્રયાસ કરશે.

એસોસિયેટ ચેમ્બર્સ ઓફ કોમર્સ એન્ડ ઇન્ડસ્ટ્રી ઈન ઇન્ડિયા (એસોચેમ) મુજબ, ઓનલાઇન શોપિંગનું શાસન ચાલુ છે અને તે 3.10,000 કરોડના આંકડાને પાર કરશે જે ગયા વર્ષના વોલ્યુમ કરતાં 350 ટકા વધુ છે.

પરંપરાગત વાણિજ્ય વિ. ઈ કોમર્સ:

પરંપરાગત વાણિજ્યમાં ખરીદનારને ઉત્પાદન અથવા સેવા ખરીદવા માટે દુકાનમાં જવું પડે છે, પરંતુ ઈ કોમર્સમાં, ઉત્પાદન અથવા સેવાઓ વેબ સાઇટ પર સૂચિબદ્ધ હોય છે અને ખરીદનાર તેના ફોટા અને વેબમાં આપેલ અન્ય સ્પષ્ટીકરણો ચકાસીને ઉત્પાદન પસંદ કરે છે. સાઈટ્સ અને પછી ડેબિટ/ક્રેડિટ કાર્ડ અથવા ઇન્ટરનેટ બેંકિંગ પદ્ધતિ દ્વારા પ્રાધાન્ય ઇલેક્ટ્રોનિક રીતે કિંમત ચૂકવવી. પછી કંપની દ્વારા ગ્રાહકના સરનામા પર ઉત્પાદન પહોંચાડવામાં આવે છે. ડિલિવરી ચાર્જ મફત હોઈ શકે છે અથવા તે ઉત્પાદન કિંમતની ચૂકવણી સાથે ચૂકવવામાં આવી શકે છે.

પરંપરાગત વાણિજ્યમાં ગ્રાહક ભૌતિક રીતે ઉત્પાદનની ચકાસણી કરી શકે છે ઉદાહરણ તરીકે જો કોઈ એલઈડી ટીવી ખરીદતું હોય તો તે દુકાનમાં જઈને વ્યક્તિગત રીતે ચિત્રની ગુણવત્તા જાતે ચકાસી શકે છે,

પરંતુ ઈ કોમર્સ વેબસાઇટમાં તે શક્ય નથી. માત્ર ખરીદનાર જ ઉત્પાદનના ફોટા જોઈ શકે છે પરંતુ તે પોતાની જાતે ચિત્રની ગુણવત્તા કે અવાજની ગુણવત્તા જોઈ શકતો નથી.

ઈ-કોમર્સમાં ગ્રાહક શારીરિક રીતે દુકાનમાં ન જઈને પોતાનો સમય અને પૈસા બચાવી શકે છે. પરંતુ ગ્રાહકને ઉત્પાદનો ખરીદવા માટે ઇન્ટરનેટ કનેક્શન અને ઉપકરણની જરૂર છે.

પરંપરાગત વાણિજ્યમાં ખરીદનાર વિક્રેતાને શારીરિક રીતે મળી શકે છે તેથી જો વેચાણ પછીની સેવાની આવશ્યકતા હોય, તો ખરીદનાર વેચનારનો સંપર્ક કરી શકે છે પરંતુ ઈ-કોમર્સમાં ખરીદનારને વેચાણકર્તાને શારીરિક રીતે મળવાની તક મળતી નથી. તેથી ગ્રાહક દ્વારા સંબંધિત ઉત્પાદનના સેવા કેન્દ્રોનો સંપર્ક કરીને વેચાણ પછીની સેવાની વ્યવસ્થા કરવાની રહેશે.

પરંપરાગત વાણિજ્યમાં ઉત્પાદનોની કિંમતો પણ એક પરિબળ છે કારણ કે ઉત્પાદનોને દુકાનોમાં સંગ્રહિત કરવા જોઈએ અને દુકાન ચલાવવા માટે કર્મચારીઓની પણ નિમણૂક કરવામાં આવે છે, દુકાન માટે ભાડું હોવાથી અથવા તે વેચનારની માલિકીની હોવી જોઈએ તેથી કિંમત ઊંચી જાય છે. ઈ કોમર્સમાં ઉત્પાદનોનું પરિવહન થતું નથી દુકાન કરો જેથી દુકાન જાળવવાની ઓવરહેડ ઉત્પાદન કિંમતમાં ઉમેરવામાં આવતી નથી. તેથી ઈ-કોમર્સ સાઈટ્સમાં ઉત્પાદનો સસ્તી છે.

જોકે એવા ઓછા લોકો છે જેઓ ઈ-કોમર્સ સાઈટ્સનો લાભ મેળવી શકે છે. મોટાભાગે શહેરી વિસ્તારોમાં રહેતા લોકોને લાભ મળે છે. કારણ કે આ સેવા સામાન્ય રીતે માત્ર શહેરી વિસ્તારો સુધી જ મર્યાદિત છે કારણ કે ગ્રામીણ લોકો ઇન્ટરનેટ સક્ષમ ઉપકરણોનો ઉપયોગ કરવા અને ઓનલાઇન વ્યવહારો કરવા માટે સુસજ્જ નથી. તેથી જ ઈ-કોમર્સ સાઈટ્સ સૌપ્રથમ ખરીદદારોનો પિન કોડ નંબર ચકાસવા માટે તપાસે છે કે ઉત્પાદન ખરીદદારને મોકલી શકાય છે કે નહીં.

ભવિષ્યમાં ટેકનોલોજીની પ્રગતિ સાથે આ સેવાનો લાભ દેશભરમાં મળી શકશે.

ફિલ્પકાર્ટ, એમેઝોન વગેરે જેવી ઇ-કોમર્સ સાઈટ્સના આગમન સાથે પરંપરાગત દુકાનોમાં વેચાણમાં ઘટાડો જોવા મળ્યો છે કારણ કે મોટાભાગના શહેરી ખરીદદારો હવે ઉત્પાદનો ખરીદવા માટે ઇ-કોમર્સ સાઈટ્સને પસંદ કરે છે કારણ કે તેઓ તેમના ઘરમાં આરામથી બેસીને ઉત્પાદન ખરીદી શકે છે.

ગ્રાહકને દુકાન કરતાં ઇ-કોમર્સ સાઈટ્સમાં વધુ ઉત્પાદનો જોવાની તક મળી શકે છે. તેથી વિકલ્પ વધુ છે અને ઇ-કોમર્સ સાઈટ્સ પણ ખરીદદારોના ઘરે ઉત્પાદન સપ્લાય કરે છે જે સામાન્ય રીતે દુકાનો કરતી નથી.

ઇ કોમર્સનો પ્રકાર અને તેનો વ્યાપ:

ઇ-કોમર્સને ચાર કેટેગરીમાં વિભાજિત કરી શકાય છે, જે બિઝનેસ ટુ બિઝનેસ બી2બી, બિઝનેસ ટુ કસ્ટમર બી2સી, ગવર્નમેન્ટ ટુ બિઝનેસ જી2બી અને ગવર્નમેન્ટ ટુ સિટીઝન જી2સી છે.

ત્યાં ચાર મુખ્ય ક્ષેત્રો છે જેમાં કંપનીઓ ઇકોમર્સનું સંચાલન કરે છે આ વિસ્તારો છે:

- પ્રત્યક્ષ માર્કેટિંગ, વેચાણ અને સેવા.
- ઓનલાઈન બેંકિંગ અને બિલિંગ.
- માહિતીનું સુરક્ષિત વિતરણ.
- મૂલ્ય સાંકળ વેપાર અને કોર્પોરેટ ખરીદી.
- સરકારને ટેક્સ રિટર્ન ભરવું.

ઇ-કોમર્સનું ક્ષેત્ર ઘણું વિશાળ છે. ઇ-કોમર્સની ઘણી એપ્લિકેશનો છે જેમ કે હોમ બેંકિંગ, ઇલેક્ટ્રોનિક મોલમાં ખરીદી કરવી, સ્ટોક ખરીદવો, નોકરી શોધવી, હરાજી કરવી, વિશ્વભરના વ્યવસાયિક ભાગીદારો સાથે ઇલેક્ટ્રોનિક રીતે સહયોગ કરવો અને ગ્રાહક સેવા પ્રદાન કરવી. વિવિધ ઇ-કોમર્સ એપ્લિકેશન્સનો અમલ ચાર મુખ્ય સપોર્ટ કેટેગરીઝ જેમ કે લોકો, જાહેર નીતિ અને માર્કેટિંગ/જાહેરાત અને સપ્લાય ચેઇન લોજિસ્ટિક્સ પર આધારિત છે. આ ઉપરાંત ઇન્ફ્રાસ્ટ્રક્ચર સપોર્ટ હોવો જોઈએ. દરેક સંસ્થામાં ઇ-કોમર્સ મેનેજમેન્ટ એપ્લિકેશન અને ઇન્ફ્રાસ્ટ્રક્ચરનું સંકલન કરે છે. સંબંધોને સમજાવવા માટે મેં B2C ઇકોમર્સના કિસ્સામાં એપ્લિકેશનની નીચે સમજાવ્યું છે.

ઇ કોમર્સના ફાયદા:

- ઇ કોમર્સ લોકોને સમય કે અંતરના અવરોધો વિના વ્યવસાય કરવા દે છે. વ્યક્તિ કોઈપણ સમયે ઇન્ટરનેટ પર લોગઈન થઈ શકે છે, પછી તે દિવસ હોય કે રાત હોય અને માઉસની એક ક્લિક પર કોઈ પણ વસ્તુની ખરીદી કે વેચાણ કરી શકે છે.
- વેબ સાઈટ પરથી લીધેલા ઓર્ડર માટે વેચાણની સીધી કિંમત પરંપરાગત માધ્યમો (રિટેલ, કાગળ આધારિત) કરતાં ઓછી છે, કારણ કે ઓન-લાઈન ઇલેક્ટ્રોનિક ખરીદી ઓર્ડર પ્રક્રિયા દરમિયાન કોઈ માનવીય ક્રિયાપ્રતિક્રિયા થતી નથી. ઉપરાંત, ઇલેક્ટ્રોનિક વેચાણ વર્ચ્યુઅલ રીતે પ્રક્રિયાની ભૂલોને દૂર કરે છે, તેમજ મુલાકાતીઓ માટે ઝડપી અને વધુ અનુકૂળ છે.

- ઇ કોમર્સ વિશિષ્ટ ઉત્પાદનો માટે આદર્શ છે. આવા ઉત્પાદનો માટે ગ્રાહકો સામાન્ય રીતે ઓછા હોય છે. પરંતુ વિશાળ માર્કેટ પ્લેસ એટલે કે ઇન્ટરનેટમાં, વિશિષ્ટ ઉત્પાદનો પણ સક્ષમ વોલ્યુમ પેદા કરી શકે છે.
- ઇ કોમર્સનો બીજો મહત્વનો ફાયદો એ છે કે તે બિઝનેસ કરવાનું સૌથી સસ્તું માધ્યમ છે.
- માર્કેટ પ્લેસના રોજબરોજના દબાણે કંપનીઓને તેમની સ્પર્ધાત્મક સ્થિતિ સુધારવા માટે રોકાણ કરવાની તકો ઘટાડવામાં તેમનો ભાગ ભજવ્યો છે. પરિપક્વ બજાર, વધેલી સ્પર્ધાઓએ રોકાણ માટે ઉપલબ્ધ નાણાંની માત્રામાં ઘટાડો કર્યો છે. જો વેચાણ કિંમત વધારી શકાતી નથી અને ઉત્પાદિત ખર્ચ ઘટાડી શકાતો નથી તો તફાવત જે રીતે ઘંઘો કરવામાં આવે છે તેમાં હોઈ શકે છે. ઇકોમર્સ ખર્ચ ઘટાડીને ઉકેલ પૂરો પાડ્યો છે, જે ખર્ચ થાય છે.
- ખરીદનારના પરિપ્રેક્ષ્યમાં પણ ઇકોમર્સ ઘણા મૂર્ત ફાયદાઓ આપે છે.
 - 1 ખરીદનારના સોર્ટ આઉટ સમયમાં ઘટાડો.
 - 2 બહેતર ખરીદદાર નિર્ણયો.
 - 3 ઇન્વોઇસ અને ઓર્ડરની વિસંગતતાઓને ઉકેલવામાં ઓછો સમય પસાર થાય છે.
 - 4 વૈકલ્પિક ઉત્પાદનો ખરીદવા માટેની તકોમાં વધારો.
- વ્યવસાયને ‘ઇ કોમર્સ સક્ષમ’ બનાવવાનો વ્યૂહાત્મક ફાયદો એ છે કે તે નીચેના ક્ષેત્રોમાં ડિલિવરી સમય, શ્રમ ખર્ચ અને ખર્ચ ઘટાડવામાં મદદ કરે છે:
 - 1 દસ્તાવેજની તૈયારી
 - 2 ભૂલ શોધ અને સુધારણા
 - 3 સમાધાન
 - 4 મેઈલની તૈયારી
 - 5 ટેલિફોન કોલિંગ
 - 6 ક્રેડિટ કાર્ડ મશીનો
 - 7 ડેટા એન્ટ્રી
 - 8 ઓવરટાઈમ
 - 9 દેખરેખ ખર્ચ
- ઇ-કોમર્સનાં કાર્યકારી લાભોમાં વ્યવસાય પ્રક્રિયાઓ પૂર્ણ કરવા માટે જરૂરી સમય અને કર્મચારીઓ બંનેને ઘટાડવાનો અને અન્ય સંસાધનો પરનો તાણ ઘટાડવાનો સમાવેશ થાય છે. આ બધા ફાયદાઓને કારણે જ કોઈ વ્યક્તિ ઇ-કોમર્સ શક્તિનો ઉપયોગ કરી શકે છે અને ઇ બિઝનેસ સોલ્યુશન પ્રદાતાઓ દ્વારા ઉપલબ્ધ શક્તિશાળી ઇ કોમર્સ સોલ્યુશન્સનો ઉપયોગ કરીને બિઝનેસને ઇ બિઝનેસમાં કન્વર્ટ કરી શકે છે.

ઈ-કોમર્સ સાઈટ પર વ્યવહારો કરો (Undertake transactions on an e-commerce site)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- વિવિધ ઈ-કોમર્સ સાઈટ્સ સમજાવો
- ઓનલાઈન કેટલોગ, શોપિંગ કાર્ટ અને પેજ તપાસો સમજાવો
- ચુકવણી, ઓર્ડર પ્રોસેસિંગ અને અધિકૃતતા સમજાવો, ચાર્જ પાછા આપો
- અન્ય ચુકવણી વિકલ્પો સમજાવો.

વિવિધ ઈ કોમર્સ સાઈટ્સ

વિશ્વની કેટલીક સૌથી લોકપ્રિય ઈ કોમર્સ સાઈટ્સ છે: કેટ એન્ડ બેરલ, સિમેન્ટેક, એમવે, માઈક્રોસોફ્ટ, એમેઝોન, એચપી વગેરે.

ભારતમાં, કેટલીક પ્રારંભિક અડચણો પછી, ઈ કોમર્સ ધીમે ધીમે ગતિ પકડી રહ્યું છે. કેટલીક લોકપ્રિય વેબ સાઈટ્સ છે:

Amazon, FlipKart, Jabong, Naaptol વગેરે.

ઓનલાઈન કેટલોગ, શોપિંગ કાર્ટ અને ચેકઆઉટ પેજ

ઓનલાઈન કેટલોગ એ વેબ સાઈટ પર વેચાણ માટે આપવામાં આવેલ ઉત્પાદનોની યાદી છે. ખરીદનાર ઉત્પાદનોને બ્રાઉઝ કરીને અને તેને અનુકૂળ હોય તે ઉત્પાદન પસંદ કરીને ઉત્પાદન પસંદ કરે છે.

શોપિંગ કાર્ટ એ આખરે કિંમત ચૂકવતા પહેલા ખરીદનાર દ્વારા પસંદ કરાયેલ ઉત્પાદનોથી ભરેલી બકેટ છે. તેનો ઉપયોગ છૂટક સ્ટોર્સમાં થાય છે પરંતુ તે જ ખ્યાલ વેબ સાઈટમાં વર્ચ્યુઅલ બકેટ બનાવીને અમલમાં મૂકવામાં આવ્યો છે જે ખરીદનાર દ્વારા પસંદ કરાયેલ ઉત્પાદન દર્શાવે છે.

ઈચ્છિત ઉત્પાદન પસંદ કર્યા પછી ખરીદનાર ઉત્પાદનોની કિંમત ચૂકવવા માટે અંતે ચેકઆઉટ પર ક્લિક કરે છે.

ચુકવણી અને ઓર્ડર પ્રક્રિયા

ચેકઆઉટ પર ક્લિક કર્યા પછી એક ચુકવણી વિકલ્પ દેખાય છે જેમાં સામાન્ય રીતે COD (કેશ ઓન ડિલિવરી), ઈન્ટરનેટ બેંકિંગ, ડેબિટ કાર્ડ, ક્રેડિટ કાર્ડ અને અન્ય વિવિધ વિકલ્પો જેવા વિવિધ વિકલ્પો હોય છે.

ખરીદનાર ચુકવણી માટે યોગ્ય વિકલ્પ પસંદ કરે છે અને સફળ ચુકવણી પછી, તે ખરીદનારને સૂચિત કરવામાં આવે છે અને અંતે ઉત્પાદનો ખરીદનારના સરનામા પર પહોંચાડવામાં આવે છે.

અધિકૃતતા અને ચાર્જબેક

અધિકૃતતા અથવા અધિકૃતતા એ સામાન્ય રીતે માહિતી સુરક્ષા અને કોમ્પ્યુટર સુરક્ષા અને ખાસ કરીને એક્સેસ નિયંત્રણ સંબંધિત સંસાધનોના એક્સેસ અધિકારોને સ્પષ્ટ કરવાનું કાર્ય છે.

ચાર્જબેક એ ખરીદનાર પાસેથી કિંમત બાદ કર્યા પછી તેને પૈસા પાછા આપવાનો ઉલ્લેખ કરે છે. તે વિવિધ પરિસ્થિતિઓમાં થાય છે. ઉદાહરણ તરીકે, ધારો કે કોઈએ irctc વેબસાઈટ પરથી ટિકિટ ખરીદી અને ટિકિટની કિંમત સફળતાપૂર્વક ચૂકવી દીધી, પરંતુ પછીથી ટિકિટ બુક થઈ ન હતી. પછી ચાર્જ બેક થશે અને પૈસા ગ્રાહકને પરત કરવામાં આવશે.

અન્ય ચુકવણી વિકલ્પો

ઉપરોક્ત ચર્ચા કરેલ પેમેન્ટ વિકલ્પ સિવાય પેમેન્ટ માટે મોબાઈલ પેમેન્ટ જેવા અન્ય કેટલાક રસ્તાઓ પણ છે. તાજેતરમાં એરટેલ મની અથવા વોડાફોન એમપેસા વગેરે મોબાઈલ ચુકવણી પદ્ધતિઓ વિકસિત થઈ છે જેથી વ્યક્તિઓ તેમના મોબાઈલ દ્વારા પણ ચૂકવણી કરી શકે.

વિવિધ ઈ કોમર્સ વ્યવહારો જેમ કે યુટિલિટી બિલની ચુકવણી, વેબ સાઈટ્સથી ખરીદી, રિચાર્જિંગ વગેરે મોબાઈલ દ્વારા કરી શકાય છે.

ઈ કોમર્સ સુરક્ષા સમસ્યાઓ અને પેમેન્ટ ગેટવે (E-Commerce Security issues and Payment Gateways)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ઈ કોમર્સ સુરક્ષા સમસ્યા સમજાવો
- પેમેન્ટ ગેટવે સમજાવો.

ઈ કોમર્સ સુરક્ષા સમસ્યા

ઈ-કોમર્સ સિક્યોરિટી એ ઈ-કોમર્સ સંપત્તિઓનું અનધિકૃત ઍક્સેસ, ઉપયોગ, ફેરફાર અથવા વિનાશથી રક્ષણ છે. જ્યારે સુરક્ષા સુવિધાઓ સુરક્ષિત સિસ્ટમની બાંધધરી આપતી નથી, તે સુરક્ષિત સિસ્ટમ બનાવવા માટે જરૂરી છે.

ઈકોમર્સનાં વપરાશમાં આ જંગી વધારાને કારણે નવી પેઢી સાથે સંકળાયેલા સુરક્ષા જોખમો ઉભા થયા છે, પરંતુ કોઈપણ ઈકોમર્સ સિસ્ટમે ચાર અભિન્ન આવશ્યકતાઓને પૂર્ણ કરવી આવશ્યક છે:

- એક ગોપનીયતા - વિનિમય માહિતી અનધિકૃત પક્ષો પાસેથી રાખવી આવશ્યક છે.
- અખંડિતતા - વિનિમય કરેલ માહિતીમાં ફેરફાર અથવા ચેડાં ન કરવા જોઈએ.
- પ્રમાણીકરણ - પ્રેષક અને પ્રાપ્તકર્તા બંનેએ એકબીજાને તેમની ઓળખ સાબિત કરવી આવશ્યક છે
- અસ્વીકાર - પુરાવા જરૂરી છે કે વિનિમય માહિતી ખરેખર પ્રાપ્ત થઈ હતી.

સુરક્ષા ધમકીઓ

ટેકનિકલ હુમલાઓ એ સૌથી પડકારજનક પ્રકારનાં સુરક્ષા સમાધાનોમાંનું એક છે જે ઈ-કોમર્સ પ્રદાતાએ સામનો કરવો જોઈએ. ટેક્નિકલ હુમલાના ગુનેગારો, અને ખાસ કરીને ડિનાયલ ઑફ-સર્વિસ હુમલાઓ, સામાન્ય રીતે બેંકો, ક્રેડિટ કાર્ડ પેમેન્ટ ગેટવે, મોટા ઓનલાઈન રિટેલર્સ અને લોકપ્રિય સોશિયલ નેટવર્કિંગ સાઈટ જેવા હાઈ પ્રોફાઈલ વેબ સર્વિસ પર હોસ્ટ કરેલી સાઈટ્સ અથવા સેવાઓને લક્ષ્ય બનાવે છે.

સેવા હુમલાનો ઈનકાર

સેવાનો ઈનકાર (DoS) હુમલામાં સર્વર, નેટવર્ક અથવા વેબસાઈટને તેની સામાન્ય પ્રવૃત્તિને લકવો કરવા માટે જબરજસ્ત બનાવવું શામેલ છે. DoS હુમલાઓ સામે બચાવ એ આજે ઈન્ટરનેટ પર સૌથી પડકારજનક સુરક્ષા સમસ્યાઓ પૈકીની એક છે. આ હુમલાઓને નિષ્ફળ કરવામાં મોટી મુશ્કેલી એ હુમલાના સ્ત્રોતને શોધી કાઢવાની છે, કારણ કે તેઓ હુમલાના સાચા મૂળને છૂપાવવા માટે વારંવાર ખોટા અથવા બનાવટી IP સ્ત્રોત સરનામાંઓનો ઉપયોગ કરે છે.

યુનાઈટેડ સ્ટેટ્સની કોમ્પ્યુટર ઈમરજન્સી રેડીનેસ ટીમ સેવાના અસ્વીકાર હુમલાના લક્ષણોને વ્યાખ્યાયિત કરે છે જેમાં નીચેનાનો સમાવેશ થાય છે:

- અસામાન્ય રીતે ધીમું નેટવર્ક પ્રદર્શન
- કોઈ ચોક્કસ વેબ સાઈટની અનુપલબ્ધતા
- કોઈપણ વેબ સાઈટ ઍક્સેસ કરવામાં અસમર્થતા

- પ્રાપ્ત સ્થાન ઈમેઈલની સંખ્યામાં નાટ્યાત્મક વધારો.

DoS હુમલાઓ વિવિધ રીતે ચલાવી શકાય છે જેમાં નીચેનાનો સમાવેશ થાય છે:

ICMP ફ્લડ (સ્મર્ફ એટેક)

જ્યાં અપરાધીઓ પીડિતનું સરનામું હોવાનું દેખાડવા માટે બનાવટી સ્ત્રોત સરનામા સાથે મોટી સંખ્યામાં IP પેકેટ મોકલશે. નેટવર્કની બેન્ડવિડ્થનો ઝડપથી ઉપયોગ થાય છે, જે કાયદેસર પેકેટોને તેમના ગંતવ્ય સુધી પહોંચતા અટકાવે છે.

ટિયરડ્રોપ એટેક

ટિયરડ્રોપ એટેકમાં ટાર્ગેટ મશીનને ઓવર લેપિંગ, ઓવર સાઈઝ, પેલોડ્સ સાથે વ્યંગિત IP ટુકડાઓ મોકલવાનો સમાવેશ થાય છે. વિવિધ ઓપરેટિંગ સિસ્ટમના TCP/IP ફ્રેગમેન્ટેશન રી એસેમ્બલી કોડમાં બગને કારણે ટુકડાઓને અયોગ્ય રીતે હેન્ડલ કરવામાં આવે છે, જેના પરિણામે તે ક્રેશ થાય છે.

ફ્લેશિંગ

પરમેનન્ટ ડિનાયલ ઑફ-સર્વિસ (DDoS) તરીકે પણ ઓળખાય છે તે એટેક છે જે સિસ્ટમને એટલી ખરાબ રીતે નુકસાન પહોંચાડે છે કે તેને હાર્ડવેરને બદલવા અથવા પુનઃસ્થાપિત કરવાની જરૂર પડે છે. ગુનેગારો પીડિતના હાર્ડવેરના રિમોટ મેનેજમેન્ટ ઈન્ટરફેસમાં સુરક્ષા ખામીઓનો ઉપયોગ કરે છે, પછી તે રાઉટર્સ, પ્રિન્ટર્સ અથવા અન્ય નેટવર્કિંગ હાર્ડવેર હોય. આ ખામીઓ હુમલાખોર માટે ઉપકરણ ફર્મવેરને સંશોધિત, ભ્રષ્ટ અથવા ખામીયુક્ત ફર્મવેર ઈમેજમાં દૂરસ્થ રીતે 'અપડેટ' કરવા માટેનો દરવાજો ખુલ્લો રાખે છે, તેથી ઉપકરણને બ્રિક કરે છે અને તેને તેના મૂળ હેતુ માટે કાયમ માટે બિનઉપયોગી બનાવે છે.

ડિસ્ટ્રિબ્યુટેડ ડિનાયલ ઑફ-સર્વિસ હુમલા

ડિસ્ટ્રિબ્યુટેડ ડિનાયલ ઑફ સર્વિસ (DDoS) હુમલા IT મેનેજરો માટે સૌથી મોટો સુરક્ષા ભય છે. મિનિટોની બાબતમાં, હજારો નબળા કમ્પ્યુટર્સ કાયદેસર ટ્રાફિકને ગૂંગળાવીને પીડિત વેબસાઈટને પૂર કરી શકે છે. ડિસ્ટ્રિબ્યુટેડ ડિનાયલ ઑફ સર્વિસ એટેક (DDoS) ત્યારે થાય છે જ્યારે બહુવિધ ચેડાવાળી સિસ્ટમ્સ લક્ષિત સિસ્ટમની બેન્ડ પહોળાઈ અથવા સંસાધનોને પૂર કરે છે, સામાન્ય રીતે એક અથવા વધુ વેબ સર્વિસ. સૌથી પ્રસિદ્ધ DDoS હુમલા ફેબ્રુઆરી 2000માં થયા હતા જ્યાં Yahoo, Buy.com, eBay, Amazon અને CNN સહિતની વેબસાઈટો પર હુમલો કરવામાં આવ્યો હતો અને દરેકને કેટલાક કલાકો સુધી પહોંચી શકાય તેમ ન હતો.

બ્રુટ ફોર્સ એટેક્સ

બ્રુટ ફોર્સ એટેક એ મોટી સંખ્યામાં શક્યતાઓ અજમાવીને ક્રિપ્ટોગ્રાફિક સ્કીમને હરાવવાની એક પદ્ધતિ છે; ઉદાહરણ તરીકે, a માં સંભવિત કીઓની મોટી સંખ્યાસંદેશને ડિક્રિપ્ટ કરવા માટે કી જગ્યા. બ્રુટ ફોર્સ એટેક, જોકે પ્રકૃતિમાં લો-ટેક હોવાનું માનવામાં આવે છે તે ભૂતકાળની વાત નથી.

બિન-તકનીકી હુમલાઓ

ફિશિંગ એ ઇલેક્ટ્રોનિક કોમ્યુનિકેશનમાં વિશ્વાસપાત્ર એન્ટિટી તરીકે માસ્કરેડ કરીને વપરાશકર્તાનામ, પાસવર્ડ અને ક્રેડિટ કાર્ડની વિગતો જેવી સંવેદનશીલ માહિતી મેળવવાનો પ્રયાસ કરવાની ગુનાહિત રીતે કપટી પ્રક્રિયા છે. ફિશિંગ કૌભાંડો સામાન્ય રીતે પીડિતને ‘કપટપૂર્ણ’ ઈમેલ સાથે ઈમેલ કરીને કરવામાં આવે છે જેમાંથી સંવેદનશીલ માહિતીની વિનંતી કરતી કાયદેસર સંસ્થા હોવાનો હેતુ છે. જ્યારે પીડિત ઈમેલમાં એમ્બેડ કરેલી લિંકને અનુસરે છે. વિસ્તૃત અને સુસંસ્કૃત લાવવામાં આવે છે. કાયદેસર સંસ્થાઓની વેબસાઈટનું ડુપ્લિકેટ. ફિશિંગ હુમલાઓ સામાન્ય રીતે બેંકના ખરીદદારો, ઓનલાઈન ઓફશન સાઈટ્સને લક્ષ્ય બનાવે છે. (જેમ કે ઈબે), ઓનલાઈન રિટેલર્સ (જેમ કે એમેઝોન) અને સેવા પ્રદાતાઓ (જેમ કે પેપાલ). અનુસાર કોમ્યુનિટી બેંકર, તાજેતરના સમયમાં સાયબર ગુનેગારો તેમના હુમલાના સમયમાં વધુ આધુનિક બન્યા છે. તેમની સાથે ફુદરતી આપત્તિના સમયે સખાવતી સંસ્થાઓ તરીકે પોઝ આપે છે.

સામાજિક ઈજનેરી-સામાજિક ઈજનેરી એ લોકોને ક્રિયાઓ કરવા અથવા ગોપનીય માહિતી જાહેર કરવા માટે ચાલાકી કરવાની કળા છે. સામાજિક ઈજનેરી તકનીકોમાં બહાનું બનાવવું (જ્યાં છેતરપિંડી કરનાર પીડિતને માહિતી જાહેર કરવા માટે શોધાયેલ દૃશ્ય બનાવે છે), ઈન્ટરેક્ટિવ વૉઇસ રેકૉર્ડિંગ (IVR) અથવા ફોન ફિશિંગ (જ્યાં છેતરપિંડી કરનાર પીડિતને ફોન પર સંવેદનશીલ માહિતી જાહેર કરવા માટે મેળવે છે) અને ટ્રોજન સાથે પ્રલોભનનો સમાવેશ થાય છે. ઘોડા (જ્યાં છેતરપિંડી કરનાર પીડિતને સિસ્ટમમાં માલવેર લોડ કરવા માટે ‘બાઈટ’ કરે છે). સોશિયલ એન્જિનિયરિંગ ઈ-કોમર્સ સુરક્ષા માટે એક ગંભીર ખતરો બની ગયું છે કારણ કે તેને શોધી કાઢવું અને તેનો સામનો કરવો મુશ્કેલ છે કારણ કે તેમાં ‘માનવ’ પરિબલોનો સમાવેશ થાય છે જેને હાર્ડવેર અથવા સૉફ્ટવેરની જેમ પેચ કરી શકાતા નથી, જોકે સ્ટાફની તાલીમ અને શિક્ષણ કંઈક અંશે હુમલાને નિષ્ફળ કરી શકે છે.

કેવી રીતે સુરક્ષિત રહેવું

સુરક્ષિત વેબ સાઈટ્સ પર ખરીદી કરો

સુરક્ષિત સાઈટ્સ તમારા કમ્પ્યુટરથી ઓનલાઈન વેપારીના કમ્પ્યુટર પર માહિતી સ્થાનાંતરિત કરવા માટે એન્ક્રિપ્શન તકનીકનો ઉપયોગ કરે છે. એન્ક્રિપ્શન તમે મોકલો છો તે માહિતી, જેમ કે તમારો ક્રેડિટ કાર્ડ નંબર, કોમ્પ્યુટર હેક્સને રસ્તામાં તેને મેળવવાથી અટકાવવા માટે સ્ક્રેમ્બલ કરે છે. કાયદેસર એક્સેસ વિશેષાધિકારો ધરાવતા લોકો જ કોડને અનસ્ક્રેમ્બલ કરી શકે છે. જ્યારે તમે સુરક્ષિત સાઈટ સાથે કામ કરી રહ્યાં હોવ ત્યારે તમે કેવી રીતે કહી શકો તે અહીં છે:

જો તમે તમારી સ્ક્રીનની ટોચ પર જુઓ જ્યાં વેબ સાઈટનું સરનામું પ્રદર્શિત થાય છે (“એડ્રેસ બાર”), તમારે <https://> જોવું જોઈએ. “http” પછી પ્રદર્શિત થયેલ “s” સૂચવે છે કે વેબ સાઈટ સુરક્ષિત છે. ઘણીવાર, જ્યાં સુધી તમે ખરેખર વેબ સાઈટ પર ઓર્ડર પેજ પર ન જાઓ ત્યાં સુધી તમને “s” દેખાતું નથી.

વેબ સાઈટ સુરક્ષિત છે કે કેમ તે નિર્ધારિત કરવાની બીજી રીત એ છે કે તમારી સ્ક્રીનના એડ્રેસ બાર પર પ્રદર્શિત બંધ પેડલોક જોવાનું છે.

જો તે લોક ખુલ્લું હોય, તો તમારે માની લેવું જોઈએ કે તે સુરક્ષિત સાઈટ નથી. અલબત્ત, તમારા ડેટાને સુરક્ષિત ચેનલો પર ટ્રાન્સમિટ કરવાનું તમારા માટે ઓછું મૂલ્યવાન છે જો વેપારી ડેટાને અનસ્ક્રેમ્બલ સ્ટોર કરે છે. તમારે એ જાણવાનો પ્રયાસ કરવો જોઈએ કે શું વેપારી ડેટાને એન્ક્રિપ્ટેડ સ્વરૂપમાં સ્ટોર કરે છે. જો હેકર ઘુસણખોરી કરવામાં સક્ષમ હોય, તો તે તમારો ક્રેડિટ ડેટા અને અન્ય વ્યક્તિગત માહિતી મેળવી શકશે નહીં. વેપારીની ગોપનીયતા અને સુરક્ષા નીતિઓ તે તેના કમ્પ્યુટર્સ પર તમારા વ્યક્તિગત ડેટાને કેવી રીતે સુરક્ષિત કરે છે તે જાણવા માટે તેની ખાતરી કરો.

તમે ઓર્ડર કરો તે પહેલાં વેબ સાઈટ પર સંશોધન કરો

તમે પહેલાથી જ જાણો છો તે કંપનીઓ સાથે વેપાર કરો. જો કંપની અજાણી હોય, તો તેમના ઉત્પાદનો ખરીદતા પહેલા તમારું હોમવર્ક કરો. જો તમે અજાણી કંપની પાસેથી કંઈક ખરીદવાનું નક્કી કરો છો, તો કંપની વિશ્વાસપાત્ર છે કે નહીં તે જાણવા માટે સસ્તા ઓર્ડરથી શરૂઆત કરો.

વિશ્વસનીય કંપનીઓએ તેમના ભૌતિક વ્યવસાયના સરનામા અને ઓછામાં ઓછા એક ફોન નંબરની જાહેરાત કરવી જોઈએ, ક્યાં તો ખરીદદાર સેવા અથવા ઓર્ડર લાઈન. ફોન નંબર પર કૉલ કરો અને વ્યવસાય કાયદેસર છે કે કેમ તે નક્કી કરવા માટે પ્રશ્નો પૂછો. જો તમે કલાકો પછી કૉલ કરો છો, તો પણ ઘણી કંપનીઓ પાસે “લાઈવ” જવાબ આપવાની સેવા છે, ખાસ કરીને જો તેઓ ઓર્ડર ચૂકી જવા માંગતા ન હોય. પૂછો કે વેપારી પરત કરેલ માલસામાન અને ફરિયાદોને કેવી રીતે હેન્ડલ કરે છે. તે સંપૂર્ણ રિફંડ ઓફર કરે છે અથવા માત્ર સ્ટોર ક્રેડિટ્સ આપે છે તે શોધો.

તમે બેટર બિઝનેસ બ્યુરો, અથવા ડિસ્ટ્રિક્ટ એટર્ની ઓફિસ અથવા એટર્ની જનરલ જેવી સરકારી ગ્રાહક સુરક્ષા એજન્સી દ્વારા પણ કંપનીનું સંશોધન કરી શકો છો. કદાચ લિસ્ટેડ શહેરમાં રહેતા મિત્રો અથવા પરિવારના સભ્યો કંપનીની માન્યતા ચકાસી શકે છે. યાદ રાખો, કોઈપણ વ્યક્તિ વેબ સાઈટ બનાવી શકે છે.

પેમેન્ટ ગેટવે

પેમેન્ટ ગેટવે એ ઈ-કોમર્સ એપ્લિકેશન સેવા પ્રદાતા સેવા છે જે ઈ-વ્યવસાયો, ઓનલાઈન રિટેલર્સ, ઈંટો અને ક્લિક્સ અથવા પરંપરાગત ઈંટ અને મોર્ટાર માટે ક્રેડિટ કાર્ડ ચુકવણીને અધિકૃત કરે છે.

તે મોટાભાગના રિટેલ આઉટલેટ્સમાં સ્થિત ફિઝિકલ પોઈન્ટ ઓફ સેલ ટર્મિનલની સમક્ષ છે. પેમેન્ટ ગેટવે ખરીદનાર અને વેપારી વચ્ચે અને વેપારી અને પેમેન્ટ પ્રોસેસર વચ્ચે પણ માહિતી સુરક્ષિત રીતે પસાર થાય છે તેની ખાતરી કરવા માટે ક્રેડિટ કાર્ડ નંબર જેવી સંવેદનશીલ માહિતીને એન્ક્રિપ્ટ કરીને ક્રેડિટ કાર્ડની વિગતોનું રક્ષણ કરે છે.

પેમેન્ટ ગેટવે પેમેન્ટ પોર્ટલ (જેમ કે વેબસાઈટ, મોબાઈલ ફોન અથવા ઈન્ટરેક્ટિવ વૉઇસ રિસ્પોન્સ સર્વિસ) અને ફ્રન્ટ એન્ડ પ્રોસેસર અથવા હસ્તગત બેંક વચ્ચે માહિતીના ટ્રાન્સફરની સુવિધા આપે છે.

વ્યવહાર પ્રક્રિયા

- જ્યારે ખરીદદાર પેમેન્ટ ગેટવે-સક્ષમ વેપારી પાસેથી પ્રોડક્ટનો ઓર્ડર આપે છે, ત્યારે પેમેન્ટ ગેટવે વ્યવહાર પર પ્રક્રિયા કરવા માટે વિવિધ કાર્યો કરે છે.

માહિતી સુરક્ષા અને ધમકીઓનું વિહંગાવલોકન (Overview of information security and threats)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- માહિતી સુરક્ષા અને તેના મૂળભૂત સિદ્ધાંતોનું વર્ણન કરો
- માહિતી સુરક્ષા અને સાયબર સુરક્ષા વચ્ચેના સંબંધનું વર્ણન કરો
- માહિતી સુરક્ષામાં મુખ્ય પડકારોનું વર્ણન કરો
- માહિતી સુરક્ષાના લાભોનું વર્ણન કરો
- માહિતી સુરક્ષાના અમલીકરણની પદ્ધતિઓ સમજાવો.

પરિચય

માહિતી સુરક્ષા એ ગોપનીયતા, અખંડિતતા અને ઉપલબ્ધતા પ્રદાન કરવા માટે અનધિકૃત ઍક્સેસ, ઉપયોગ, જાહેરાત, વિક્ષેપ, ફેરફાર અથવા વિનાશથી માહિતી અને માહિતી સિસ્ટમોનું રક્ષણ છે. તે એક સામાન્ય શબ્દ છે જેનો ઉપયોગ ડેટા ગમે તે સ્વરૂપે લઈ શકે (ઇલેક્ટ્રોનિક, ભૌતિક, વગેરે)

સરકારો, સૈન્ય, કોર્પોરેશનો, નાણાકીય સંસ્થાઓ, હોસ્પિટલો અને ખાનગી વ્યવસાયો તેમના કર્મચારીઓ, ગ્રાહકો વિશે ઘણી બધી ગોપનીય માહિતીનો સંગ્રહ કરે છે. , ઉત્પાદનો, સંશોધન અને નાણાકીય સ્થિતિ. આમાંની મોટાભાગની માહિતી હવે ઇલેક્ટ્રોનિક કમ્પ્યુટર્સ પર એકત્રિત, પ્રક્રિયા અને સંગ્રહિત કરવામાં આવે છે અને નેટવર્કમાંથી અન્ય કમ્પ્યુટર્સ પર પ્રસારિત થાય છે.

જો કોઈ વ્યવસાયના ગ્રાહકો અથવા નાણાકીય અથવા નવી પ્રોડક્ટ લાઈન વિશેની ગોપનીય માહિતી કોઈ હરીફ અથવા હેકરના હાથમાં આવી જાય, તો વ્યવસાય અને તેના ગ્રાહકોને વ્યાપક ભરપાઈ ન થઈ શકે તેવું નાણાકીય નુકસાન થઈ શકે છે, કંપનીની પ્રતિષ્ઠાને નુકસાનનો ઉલ્લેખ ન કરવો. ગોપનીય રક્ષણ માહિતી એ વ્યવસાયની જરૂરિયાત છે અને ઘણા કિસ્સાઓમાં નૈતિક અને કાનૂની જરૂરિયાત પણ છે.

માહિતી ખાતરી: માહિતી ખાતરી એ સુનિશ્ચિત કરવાનું કાર્ય છે કે જ્યારે ગંભીર સમસ્યાઓ ઊભી થાય ત્યારે ડેટા ખોવાઈ ન જાય. આ મુદ્દાઓમાં સમાવેશ થાય છે પરંતુ તે આટલા સુધી મર્યાદિત નથી: કુદરતી આફતો, કમ્પ્યુટર/સર્વરની ખામી, ભૌતિક ચોરી અથવા અન્ય કોઈપણ ઉદાહરણ જ્યાં ડેટા ખોવાઈ જવાની સંભાવના હોય. આપણા આધુનિક યુગમાં મોટાભાગની માહિતી કમ્પ્યુટર પર સંગ્રહિત હોવાથી, માહિતીની ખાતરી સામાન્ય રીતે IT સુરક્ષા નિષ્ણાતો દ્વારા કરવામાં આવે છે. માહિતીની ખાતરી પૂરી પાડવાની સૌથી સામાન્ય પદ્ધતિઓમાંની એક એ છે કે ઉલ્લેખિત સમસ્યાઓમાંથી કોઈ એક ઉદ્ભવે તો ડેટાનો ઓફ-સાઈટ બેકઅપ લેવો.

માહિતી સુરક્ષાના મૂળભૂત સિદ્ધાંતો: સીઆઈએ ટ્રાયડ સુરક્ષા નીતિના વિકાસ માટે જાણીતું મોડેલ છે, જેનો ઉપયોગ સમસ્યા વિસ્તારો અને માહિતી સુરક્ષા માટે જરૂરી ઉકેલો ઓળખવા માટે થાય છે. ગોપનીયતા, અખંડિતતા અને પ્રાપ્યતા (CIA) એ એક મોડેલ છે જે માહિતી સુરક્ષા માટે નીતિઓનું માર્ગદર્શન આપવા માટે રચાયેલ છે. સંસ્થા ગોપનીયતા, અખંડિતતા અને ઉપલબ્ધતાની CIA ત્રિપુટી માહિતી સુરક્ષાના કેન્દ્રમાં છે. ક્લાસિક માહિતી સુરક્ષા ત્રિપુટીના સભ્યો

ગોપનીયતા

ગોપનીયતા, અખંડિતતા અને ઉપલબ્ધતા - ને સુરક્ષા વિશેષતાઓ, ગુણધર્મો, સુરક્ષા ધ્યેયો, મૂળભૂત પાસાઓ, માહિતી માપદંડો, મહત્વપૂર્ણ માહિતી લાક્ષણિકતાઓ અને મૂળભૂત બિલ્ડિંગ બ્લોક્સ તરીકે પણ ઓળખવામાં આવે છે.

ગોપનીયતા

ગોપનીયતા એ નિયમોનો સમૂહ છે જે માહિતીની ઍક્સેસને મર્યાદિત કરે છે. ગોપનીયતા સંવેદનશીલ માહિતીને ખોટા લોકો સુધી પહોંચતા અટકાવે છે, જ્યારે તે સુનિશ્ચિત કરે છે કે યોગ્ય લોકો તે મેળવી શકે છે. ગોપનીયતાનું રક્ષણ કરવું એ માહિતી માટે યોગ્ય ઍક્સેસ સ્તરોને વ્યાખ્યાયિત કરવા અને લાગુ કરવા પર આધાર રાખે છે. કોને તેની ઍક્સેસ હોવી જોઈએ અને તે કેટલી સંવેદનશીલ છે તેના દ્વારા આયોજિત અલગ-અલગ ઍક્સેસોમાં માહિતીને અલગ કરીને આ કરી શકાય છે.

અખંડિતતા

અખંડિતતા એ ખાતરી છે કે માહિતી તેના સમગ્ર જીવન ચક્રમાં વિશ્વાસપાત્ર, સુસંગત અને સચોટ છે. આનો અર્થ એ છે કે ડેટાને અનધિકૃત અથવા વણશોધાયેલ રીતે સુધારી શકાતો નથી. ટ્રાન્ઝિએટમાં ડેટા બદલવો જોઈએ નહીં અને અનધિકૃત લોકો (ઉદાહરણ તરીકે, ગોપનીયતાના ભંગમાં) દ્વારા ડેટાને બદલી શકાય નહીં તેની ખાતરી કરવા માટે પગલાં લેવા જોઈએ. આ ઉપરાંત, ઇલેક્ટ્રોમેગ્નેટિક પલ્સ જેવી બિન-માનવીય ઘટનાઓના પરિણામે થતા ડેટામાં થતા કોઈપણ ફેરફારોને શોધવા માટે કેટલાક માધ્યમો હોવા જોઈએ. (EMP) અથવા સર્વર કેશ. જો કોઈ અનપેક્ષિત ફેરફાર થાય છે, તો અસરગ્રસ્ત ડેટાને તેની સાચી સ્થિતિમાં પુનઃસ્થાપિત કરવા માટે બેકઅપ કોપી ઉપલબ્ધ હોવી આવશ્યક છે.

ઉપલબ્ધતા

પ્રાપ્યતા એ અધિકૃત લોકો દ્વારા માહિતીની તૈયાર ઍક્સેસની બાંધધરી છે. કોઈપણ માહિતી પ્રણાલી તેના હેતુને પૂર્ણ કરવા માટે, માહિતી જ્યારે જરૂરી હોય ત્યારે ઉપલબ્ધ હોવી જોઈએ. આનો અર્થ એ છે કે માહિતીને સંગ્રહિત કરવા અને પ્રક્રિયા કરવા માટે ઉપયોગમાં લેવાતી કમ્પ્યુટિંગ સિસ્ટમ્સ, તેને સુરક્ષિત કરવા માટે ઉપયોગમાં લેવાતા સુરક્ષા નિયંત્રણો અને તેને ઍક્સેસ કરવા માટે ઉપયોગમાં લેવાતી સંચાર ચેનલો યોગ્ય રીતે કાર્ય કરતી હોવી જોઈએ.

પાવર આઉટેજ, હાર્ડવેર નિષ્ફળતા અને સિસ્ટમ અપગ્રેડને કારણે સેવામાં આવતા વિક્ષેપોને અટકાવીને, ઉચ્ચ ઉપલબ્ધતા સિસ્ટમો હંમેશા ઉપલબ્ધ રહેવાનું લક્ષ્ય રાખે છે. તમામ હાર્ડવેરની સખત રીતે જાળવણી કરીને, જરૂર પડ્યે તરત જ હાર્ડવેર રિપેર કરીને, રિડન્ડન્સી અને ફેલઓવરનું ચોક્કસ માપ પ્રદાન કરીને, પર્યાપ્ત કોમ્પ્યુનિકેશન બેન્ડવિડ્થ પ્રદાન કરીને અને બોટલ નેકની ઘટનાને અટકાવવા, કટોકટી બેકઅપનો અમલ કરીને ઉપલબ્ધતા શ્રેષ્ઠ રીતે સુનિશ્ચિત કરવામાં આવે છે. પાવર સિસ્ટમ્સ, તમામ જરૂરી સિસ્ટમ અપગ્રેડ સાથે વર્તમાન જાળવી રાખે છે અને ડિનાયલ-ઓફ-સર્વિસ (DoS) હુમલા જેવી દૂષિત ક્રિયાઓ સામે રક્ષણ આપે છે.

ઉપરોક્ત ત્રણ સભ્યો ઉપરાંત, અધિકૃતતા અને બિન-અસ્વીકાર પણ CIA મોડેલના સભ્યો તરીકે ગણવામાં આવે છે.

અધિકૃતતા

અધિકૃતતા એ ખાતરી કરવાની પ્રક્રિયા છે કે ડેટા, વ્યવહારો, સંદેશાવ્યવહાર અથવા દસ્તાવેજો અસલી છે. અધિકૃતતા માટે તે માન્ય કરવું પણ મહત્વપૂર્ણ છે કે સામેલ પક્ષો અસલી છે. કેટલીક માહિતી સુરક્ષા પ્રણાલીઓમાં “ડિજિટલ હસ્તાક્ષર” જેવી પ્રમાણીકરણ સુવિધાઓનો સમાવેશ થાય છે, જે પુરાવા આપે છે કે સંદેશ મોકલવામાં આવેલ ડેટા અસલી છે અને યોગ્ય સાઈનિંગ કી ધરાવતા કોઈ વ્યક્તિ દ્વારા મોકલવામાં આવ્યો હતો.

નોન-રિપ્યુએશન

નોન-રિપ્યુએશનનો અર્થ એ છે કે વ્યક્તિનો કરાર પ્રત્યેની તેની જવાબદારીઓ પૂરી કરવાનો ઈરાદો. તેનો અર્થ એ પણ થાય છે કે ટ્રાન્ઝેક્શનનો એક પક્ષ એમ કહી શકતો નથી કે તેમને કોઈ ટ્રાન્ઝેક્શન મળ્યું નથી અને ન તો બીજી પાર્ટી ટ્રાન્ઝેક્શન મોકલ્યાનો ઇનકાર કરી શકે છે.

એ નોંધવું અગત્યનું છે કે જ્યારે ક્રિપ્ટોગ્રાફિક સિસ્ટમ્સ જેવી ટેકનોલોજી બિન-અસ્વીકારના પ્રયત્નોમાં મદદ કરી શકે છે, ત્યારે ખ્યાલ મૂળભૂત રીતે કાનૂની ખ્યાલ છે. દાખલા તરીકે, તે બતાવવા માટે પૂરતું નથી કે સંદેશ પ્રેષકની ખાનગી કી સાથે હસ્તાક્ષર કરેલ ડિજિટલ હસ્તાક્ષર સાથે મેળ ખાય છે, અને આ રીતે ફક્ત મોકલનાર સંદેશ મોકલ્યો છે અને અન્ય કોઈ તેને ટ્રાન્ઝિટમાં બદલી શક્યું નથી. કથિત પ્રેષક બદલામાં દર્શાવી શકે છે કે ડિજિટલ હસ્તાક્ષર અલ્ગોરિધમ સંવેદનશીલ અથવા ખામીયુક્ત છે, અથવા આરોપ અથવા સાબિત કરી શકે છે કે તેની સાઈનિંગ કી સાથે ચેડા કરવામાં આવ્યો છે. આ ઉલ્લંઘનો માટે દોષ પોતે પ્રેષકની સાથે હોઈ શકે છે અથવા ન પણ હોઈ શકે, અને આવા દાવાઓ પ્રેષકને જવાબદારીમાંથી મુક્ત કરી શકે છે અથવા ન પણ કરી શકે છે, પરંતુ નિવેદન એ દાવાને અમાન્ય કરશે કે હસ્તાક્ષર આવશ્યકપણે અધિકૃતતા અને અખંડિતતા સાબિત કરે છે અને આમ અસ્વીકાર અટકાવે છે.

અમને લગભગ અમર્યાદિત સ્વતંત્રતા આપતી તમામ પ્રવૃત્તિઓ સાથે, જોખમો છે. કારણ કે ઈન્ટરનેટ કોઈપણ માટે સરળતાથી સુલભ છે, તે જોખમી સ્થળ બની શકે છે. તમે કોની સાથે વ્યવહાર કરી રહ્યાં છો અથવા તમે શું કરી રહ્યાં છો તે જાણો. શિકારી, સાયબર ગુનેગારો, ગુંડાઓ અને ભ્રષ્ટ વ્યવસાયો અવિચારી મુલાકાતીનો લાભ લેવાનો પ્રયાસ કરશે.

ઈન્ટરનેટ, ઈન્ટ્રાનેટ અને સુરક્ષા

જો કે ઈન્ટ્રાનેટ અને ઈન્ટરનેટ વચ્ચેનો તફાવત ટેકનોલોજીની દ્રષ્ટિએ મોટો નથી, માહિતીનું પ્રસારણ સંસ્થાકીય દૃષ્ટિકોણથી સંપૂર્ણપણે અલગ છે.

ઈન્ટ્રાનેટ્સ અને અન્ય નેટવર્ક્સ અને માહિતી પ્રણાલીઓ વચ્ચે માહિતી સુરક્ષા જોખમો તેના બદલે સમાન છે. આંતરિક માહિતી ચેનલો તરીકે ઈન્ટ્રાનેટનો ઉપયોગ માહિતી સુરક્ષાના મહત્વ પર ભાર મૂકે છે. આંતરિક ઈન્ટ્રાનેટ પર રાખવામાં આવેલી સંપત્તિ સંભવિત દુરુપયોગકર્તાઓના હિતમાં વધારો કરી શકે છે. આથી, ગુપ્તતા, અખંડિતતા અને માહિતીની ઉપલબ્ધતાને જોખમમાં મૂકતા વિવિધ જોખમો સામે ઈન્ટ્રાનેટ્સ અને તેમના દ્વારા પ્રસારિત થતી માહિતી અને માહિતીનું રક્ષણ કરવું એ અત્યંત મહત્વપૂર્ણ વિચારણા છે.

ઈન્ટ્રાનેટના ભાગ રૂપે ઈન્ટરનેટનો ઉપયોગ કરવો એ ગંભીર ખતરો છે, કારણ કે ઈન્ટરનેટ સ્વાભાવિક રીતે જ અસુરક્ષિત છે. પરિણામે, વપરાશકર્તાઓએ ખાસ કરીને તેમના સંદેશાવ્યવહારને એન્ક્રિપ્ટ કરવામાં ખૂબ કાળજી રાખવી જોઈએ. ઈમિટેશન (સ્પૂફિંગ), રિપ્લાય (રેપિડ ફાયર), મેસેજની સામગ્રીમાં ફેરફાર (સુપરઝેપિંગ), સેવાની ઉપલબ્ધતાનું નિવારણ અને સક્રિય અને નિષ્ક્રિય વાયરટેપિંગ સૌથી વધુ દૂષિત છે. ધમકીઓ ઉદાહરણ તરીકે, વાયરટેપિંગ એવી પરિસ્થિતિ તરફ દોરી શકે છે કે જ્યાં

કોઈ સંસ્થા સંબંધિત વ્યૂહાત્મક જ્ઞાન બહારના લોકોના હાથમાં આવે, જો સંચાર એન્ક્રિપ્શન મજબૂત એન્ક્રિપ્શન પદ્ધતિઓ દ્વારા અમલમાં ન આવે તો.

હેકર ટૂલ્સ, જો કે ઈન્ટરનેટ માટે વિકસાવવામાં આવ્યા છે, તે ઈન્ટ્રાનેટ પર પણ વાપરી શકાય તેવા છે. તે સોફ્ટવેર અથવા હાર્ડવેર આધારિત અથવા બંનેનું મિશ્રણ હોઈ શકે છે. તેમના અધિકૃત ઉપયોગમાં ઈન્ટ્રાનેટ્સ પર માહિતી સુરક્ષા નબળાઈઓ શોધવા અને સુધારવાનો સમાવેશ થાય છે. જો કે, તેઓ આંતરિક લોકોને આવી સંચાર પ્રણાલીને હેક કરવા અને માહિતી મેળવવા માટે સક્ષમ કરે છે જેને એક્સેસ કરવા માટે તેઓ અધિકૃત નથી.

માહિતી સુરક્ષા અને સાયબર સુરક્ષા વચ્ચેનો

સંબંધ માહિતી સુરક્ષા, જે અગાઉના વિભાગોમાં ઉલ્લેખિત છે તે માહિતી અને માહિતી પ્રણાલીઓને ગુપ્તતા, અખંડિતતા અને ઉપલબ્ધતા પ્રદાન કરવા માટે અનધિકૃત એક્સેસ, ઉપયોગ, જાહેરાત, વિક્ષેપ, ફેરફાર અથવા વિનાશથી રક્ષણ આપે છે. બીજી તરફ સાયબર સુરક્ષાને સાયબર હુમલાઓથી સાયબર સ્પેસના ઉપયોગને બચાવવા અથવા બચાવવાની ક્ષમતા તરીકે વ્યાખ્યાયિત કરી શકાય છે. સાયબરસ્પેસ એ “એવું વાતાવરણ છે જેમાં કમ્પ્યુટર નેટવર્ક્સ પર સંચાર થાય છે.” સાયબર સુરક્ષામાં સાયબર ડોમેન અથવા ક્ષેત્રમાં (અથવા સાયબરસ્પેસ). માહિતી સુરક્ષામાં માહિતી અથવા માહિતી પ્રણાલીઓની સુરક્ષાનો સમાવેશ થાય છે, તે ગમે તે ક્ષેત્રમાં થાય છે (દા.ત., ભૌતિક વિશ્વમાં સંપર્કમાં આવવાનું જોખમ). સાયબર ક્ષેત્રમાં જે કંઈપણ થાય છે તેમાં કોઈક રીતે માહિતી અને માહિતી પ્રણાલીના રક્ષણનો સમાવેશ થતો હોવાથી, તમે નિષ્કર્ષ પર આવી શકો છો કે માહિતી સુરક્ષા એ સાયબર સુરક્ષાનો એક સુપર સમૂહ છે. (ફિગ 1) અમુક સમયે બે શબ્દો એકબીજાના બદલે વાપરવામાં આવે છે.

માહિતી સુરક્ષામાં મુખ્ય પડકારો

1 IT સુરક્ષાને ઓછી પ્રાથમિકતા સોંપવામાં આવી છે

- સંસ્થાએ IT સુરક્ષા પ્રથાઓ લાગુ કરવા પર યોગ્ય ધ્યાન કેન્દ્રિત કર્યું નથી.

2 તદર્થ સુરક્ષા શાસન

- ઈન્ફોર્મેશન સિક્યુરિટી મેનેજમેન્ટ સિસ્ટમ (ISMS) અથવા સ્ટ્રક્ચર્ડ ગવર્નન્સ મિકેનિઝમની ગેરહાજરી. (ફિગ 1)

3 ભૂમિકાઓ અને જવાબદારીઓમાં અસ્પષ્ટતા

- SSO માં વિવિધ ખેલાડીઓ (વ્યવસાય, SSO માં ટીમો, વગેરે) ની ભૂમિકાઓ અને જવાબદારીઓ પર અસ્પષ્ટતાઓ અસ્તિત્વમાં છે. સિંગલ સાઈન-ઓન (SSO) એ બહુવિધ સંબંધિત એક્સેસ નિયંત્રણની મિલકત છે, પરંતુ સ્વતંત્ર સોફ્ટવેર સિસ્ટમો. આ ગુણધર્મ સાથે વપરાશકર્તા એકવાર લોગ ઈન કરે છે અને દરેક સિસ્ટમમાં ફરીથી લોગ ઈન કરવા માટે સંકેત આપ્યા વિના તમામ સિસ્ટમમાં પ્રવેશ મેળવે છે.

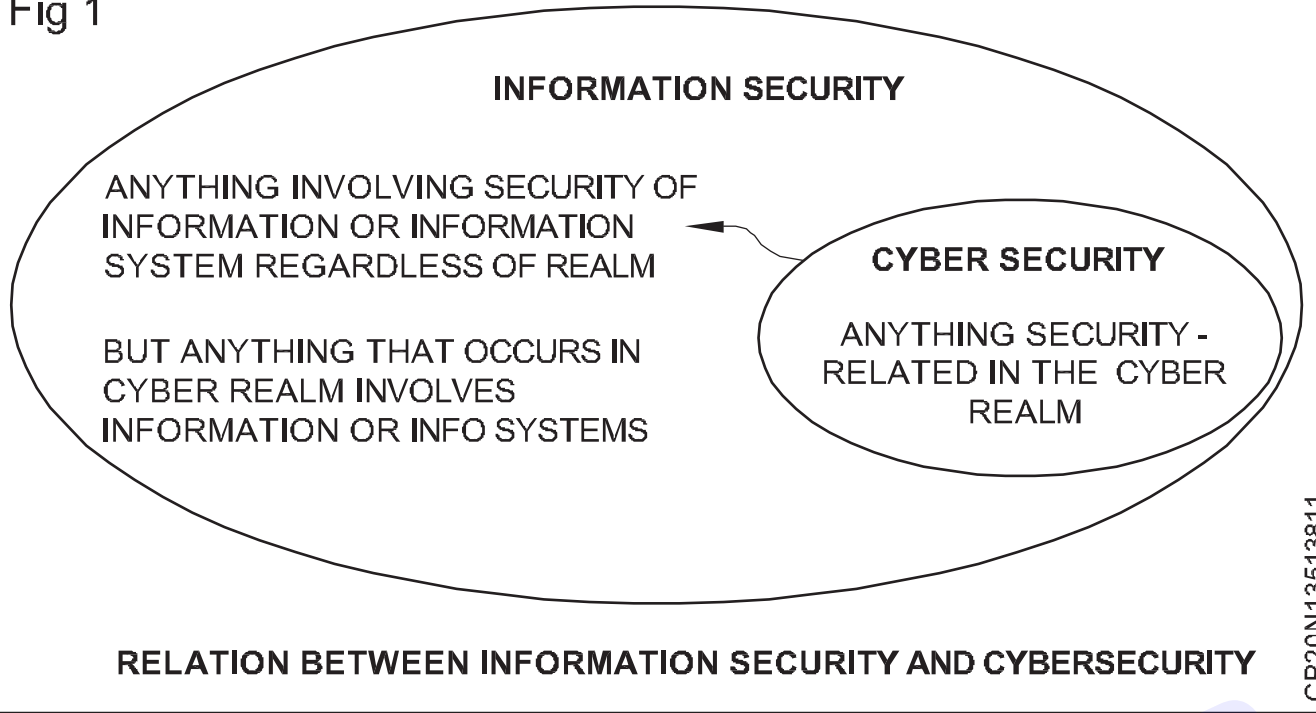
4 ફરજોનું અપૂરતું વિભાજન

- SSO માં ઓવરલેપિંગ અને વહેંચાયેલ જવાબદારીઓ ફરજોમાં અલગતાના યોગ્ય સ્તરને લાગુ કરવાનું મુશ્કેલ બનાવે છે.

5 સુરક્ષા જરૂરિયાતોના વિવિધ અર્થઘટન

- પ્રમાણભૂત અર્થઘટનોની ગેરહાજરીમાં, વિવિધ વ્યક્તિઓ અને ટીમોના પોતાના અર્થઘટન હોય છે.

Fig 1



6 જોખમ સ્તર ઘટાડવાનું વલણ

- ટીમો સંચાલક પ્રક્રિયાઓની કઠોરતાને બાયપાસ કરવા માટે 'રિસ્ક લેવલ' ઘટાડવાનું વલણ દર્શાવે છે.

7 બહુવિધ વિકેતાઓ

- અવિરત સ્પર્ધા અને અસુરક્ષાની ભાવનાને લીધે વિકેતાઓ વચ્ચે જવાબદારી વહેંચવામાં અનિચ્છા અને ઓછા અથવા કોઈ સહયોગ નથી.

8 વ્યાપાર/ કામગીરી બહુવિધ ભૌગોલિક ક્ષેત્રોમાં ફેલાયેલી છે

- સંસ્થા સમગ્ર વિશ્વમાં ફેલાયેલા બહુવિધ સ્થાનો પર આધારિત છે અને કાર્ય કરે છે.

9 તાલીમ/જાગૃતિનો અભાવ

- સુરક્ષા પ્રથાઓ પર અપૂરતી તાલીમ અને જાગૃતિ.

માહિતી સુરક્ષાના ફાયદા:

- વિવિધ સાયબર ક્ષાઈમના બાહ્ય જોખમોથી પ્રભાવને ઘટાડવા માટે અનધિકૃત ઍક્સેસથી નેટવર્ક્સ, કમ્પ્યુટર્સ અને ડેટાને સુરક્ષિત કરો
- તકનીકી, સંચાલન, વહીવટી અમલીકરણ માટે સુધારેલ માહિતી સુરક્ષા અને વ્યવસાય સાતત્ય સંચાલન અને ઓપરેશનલ કંટ્રોલ, જે જોખમ ઘટાડવાની સૌથી સસ્તી અસરકારક રીત છે.
- માહિતી સુરક્ષા વ્યવસ્થામાં હિસ્સેદારોનો વિશ્વાસ સુધર્યો.
- સંસ્થામાં યોગ્ય સુરક્ષા નિયંત્રણો સાથે સુધારેલ કંપની ઓળખપત્ર આંતરિક હિસ્સેદારો અને બાહ્ય વિકેતાઓ વચ્ચે વિશ્વસનીયતા અને વિશ્વાસમાં સુધારો કરશે. વિશ્વસનીયતા અને વિશ્વાસ એ વ્યવસાય જીતવા માટેના મુખ્ય પરિબલો છે.
- વિક્ષેપની સ્થિતિમાં ઝડપી પુનઃપ્રાપ્તિ સમય.

સંસ્થામાં IS લાગુ કરવા માટેની તકનીકી માહિતી સુરક્ષાને લાગુ કરવા માટેના સાધનોની ઓળખ એક સફળ માહિતી સુરક્ષા

નીતિ કોર્પોરેશનોને ઘણા લાભો પ્રદાન કરે છે. અમલ કરવા યોગ્ય નીતિઓ સુનિશ્ચિત કરે છે કે નબળાઈઓને ઓળખવામાં આવે છે અને તેને સંબોધવામાં આવે છે. આના પરિણામે વ્યાપાર સાતત્યનું રક્ષણ થાય છે અને આઈટી ઇન્ફ્રાસ્ટ્રક્ચર મજબૂત બને છે. જ્યારે સમગ્ર સંસ્થામાં કર્મચારીઓ સુરક્ષા નીતિનું પાલન કરે છે, તેની ખાતરી કરીને કે માહિતી સંસ્થામાં તેમજ ગ્રાહકો, ભાગીદારો અને વિકેતાઓ સાથે સુરક્ષિત રીતે વહેંચવામાં આવે છે, ત્યારે જોખમ ઓછું થાય છે.

1 અસરકારક માહિતી સુરક્ષા નીતિ બનાવવાનું પ્રથમ પગલું માહિતી સંપત્તિનું મૂલ્યાંકન કરવું અને તે સંપત્તિઓ માટેના જોખમોને ઓળખવું.

સંસ્થાની અંદરની કેટલીક સંપત્તિ અન્ય કરતાં વધુ મૂલ્યવાન હશે, પરંતુ નાણાકીય મૂલ્ય એકમાત્ર પરિબલ હોવું જોઈએ નહીં. નાણાકીય મૂલ્ય અને સંપત્તિનું આંતરિક મૂલ્ય બંને નક્કી કરવું તેની કિંમતનું ચોક્કસ મૂલ્યાંકન કરવા માટે જરૂરી છે. ગણતરી કરવા માટે એસેટના નાણાકીય મૂલ્યમાં, સંસ્થાએ અસરને ધ્યાનમાં લેવી જોઈએ જો તે સંપત્તિના ડેટા, નેટવર્ક્સ અથવા સિસ્ટમ્સ સાથે કોઈપણ રીતે ચેડા કરવામાં આવે. આંતરિક મૂલ્યની ગણતરી કરવા માટે, સંસ્થાએ વિશ્વસનીયતા, પ્રતિષ્ઠા અને મુખ્ય હિસ્સેદારો સાથેના સંબંધો પર સુરક્ષા ઘટનાની અસરને ધ્યાનમાં લેવી જોઈએ.

2 એક નીતિ બનાવવી એ સંસ્થાઓ માટે જોખમ મૂલ્યાંકન કરવા માટે છે

અસ્કયામતો અને ધમકીઓની ઓળખ પછી, નીતિ બનાવવાનું આગલું પગલું સંસ્થાઓ માટે જોખમ મૂલ્યાંકન કરવાનું છે. આ મૂલ્યાંકન સંસ્થાને તે નક્કી કરવાની મંજૂરી આપે છે કે શું માહિતી સુરક્ષિત, અતિસંરક્ષિત અથવા પર્યાપ્ત રીતે સુરક્ષિત છે. આ જોખમ મૂલ્યાંકનનો ઘ્યેય સંસ્થાને બિનજરૂરી જોખમમાં મૂક્યા વિના ખર્ચ ઘટાડવાનો હોવો

જોઈએ. એકવાર સુરક્ષા નીતિ અસરકારક રીતે અમલમાં આવી જાય પછી આ મૂલ્યાંકન સંસાધનોની યોગ્ય ફાળવણી નક્કી કરવામાં મદદ કરશે.

માહિતી સુરક્ષા ફ્રેમવર્ક માહિતી સુરક્ષા ફ્રેમવર્ક

સંસ્થા માટે સુરક્ષા નીતિ અને પ્રથાઓ સ્થાપિત કરે છે. નીતિઓ સુરક્ષાને અસર કરતી બાબતો પર માર્ગદર્શન પૂરું પાડે છે જેને સંસ્થાના સભ્યો અનુસરે તેવી અપેક્ષા છે. સુરક્ષા નીતિ તમામ હાર્ડવેર, સોફ્ટવેર, ડેટા, માહિતી, નેટવર્ક, વ્યક્તિગત કમ્પ્યુટિંગ ઉપકરણો, સહાયક કર્મચારીઓ અને સંસ્થામાંના વપરાશકર્તાઓને લાગુ પડે છે.

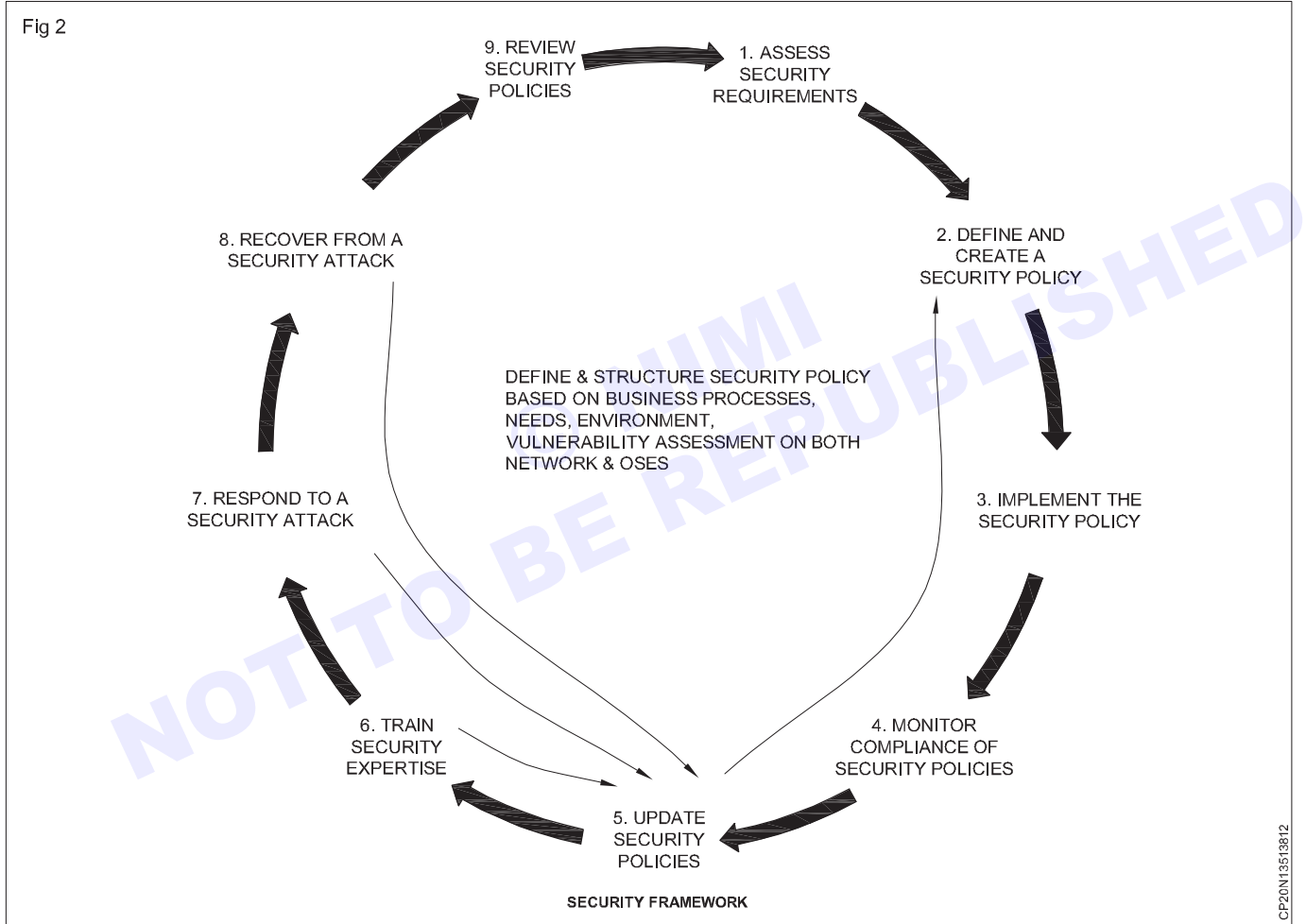
IT સુરક્ષા પ્રણાલી કામ કરે તે માટે, બધા હિતધારકોને સમાવિષ્ટ કરીને સારી રીતે વ્યાખ્યાયિત ફ્રેમવર્ક વિકસાવવાની જરૂર છે, અને ઉપયોગી થવા માટે તેને સમયાંતરે અપડેટ કરવાની જરૂર છે. માહિતી સુરક્ષા ફ્રેમવર્ક અસ્કયામતોને ઓળખવા, સુરક્ષા જરૂરિયાતો નક્કી કરવા, જોખમનું

મૂલ્યાંકન, નિયંત્રણ મૂલ્યાંકન, નિયંત્રણ અમલીકરણ, પ્રક્રિયા મોનિટરિંગ અને અપડેટના સ્વરૂપમાં ડેટાના સંચાલન અને સિસ્ટમ/પ્રક્રિયાના અમલીકરણને ધ્યાનમાં રાખીને વ્યવસ્થાપન પ્રક્રિયાને સરળ બનાવે છે.

માહિતી સુરક્ષા ફ્રેમવર્કમાં સામાન્ય રીતે ઉપયોગમાં લેવાતા શબ્દો છે:

- નીતિ: સામાન્ય સંચાલન નિવેદનો
- ધોરણો: ચોક્કસ ફરજિયાત નિયંત્રણો
- માર્ગદર્શિકા: ભલામણો / શ્રેષ્ઠ વ્યવહારો
- કાર્યપદ્ધતિઓ: પગલું-દર-પગલાની

સૂચનાઓ સુરક્ષા ફ્રેમવર્કને વાસ્તવમાં અમલમાં લાવવામાં સામેલ વિગતવાર પ્રવૃત્તિઓ, અનુસરવા માટેની પ્રેક્ટિસનો ક્રમ, જે સુધારાત્મક પગલાં લેવાના છે તે નીચે દર્શાવેલ આકૃતિમાં દર્શાવવામાં આવ્યા છે. (ફિગ 2)



ફ્રેમવર્કમાં દરેક હેડ હેઠળ મુખ્ય હેડ અને પ્રેક્ટિસ નીચે બતાવ્યા પ્રમાણે છે.

આમ સુરક્ષા માળખું એ તકનીક બની જાય છે જે સુરક્ષા નીતિઓને વ્યવહારમાં ફેરવે છે. તે યોજના, કાર્ય અને તપાસ ચક્રના ચાર પગલા ચક્ર દ્વારા તેને પ્રાપ્ત કરે છે. આ હાંસલ કરવા માટે પીપીટી ટ્રાયડ એટલે કે લોકો, પ્રક્રિયા અને ટેકનોલોજીને સમાન મહત્વ આપવાની જરૂર છે.

નવી ટેકનોલોજી અને નવા નેટવર્ક્સ સુરક્ષા માળખામાં પ્લગ થઈ શકે છે અને સુરક્ષા ખર્ચ વધુ અનુમાનિત અને વ્યવસ્થિત બને છે.

Identify	Protect	Detect	Respond	Recover
• Asset Management	• Access Control Events	• Anomalies and	• Response Planning	• Recovery Planning
• Business Environment	• Awareness and Training • Data Security	• Security Continuous Monitoring	• Communications	• Improvement
• Governance	• Information Protection and Procedures	• Detection Process	• Analysis	• Communications
• Risk	• Maintenance Assessment		• Mitigation	
• Risk Management Strategy	• Protective technology		• Improvements	

સુરક્ષા જોખમોની ઝાંખી (Overview of security threats)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- સુરક્ષા જોખમ અને તેના પ્રકારોનું વર્ણન કરો
- ધમકીઓને ઓળખવાની પદ્ધતિઓનું વર્ણન કરો
- ધમકીઓ સિસ્ટમને કેવી રીતે અસર કરે છે તે સમજાવો
- ધમકીઓના સ્ત્રોતોનું વર્ણન કરો
- ધમકીઓને ઓળખવા અને તેને ઘટાડવા માટેની શ્રેષ્ઠ પદ્ધતિઓનું વર્ણન કરો.

પરિચય

ધમકી એ કોઈપણ સંજોગો અથવા ઘટના છે જેમાં સિસ્ટમ અથવા પ્રવૃત્તિને વિનાશ, ડિસ્ક્લોઝર અને ડેટામાં ફેરફાર અથવા સેવાના ઇનકારના સ્વરૂપમાં નુકસાન પહોંચાડવાની સંભાવના છે. ધમકી એ નુકસાનની સંભાવના છે. ધમકીની હાજરીનો અર્થ એ નથી કે તે આવશ્યકપણે વાસ્તવિક નુકસાન પહોંચાડશે.

ધમકીઓ અને તેમના વર્ણન સાથે સંકળાયેલા કેટલાક સામાન્ય શબ્દો નીચે મુજબ છે:

- 1 અનધિકૃત ઍક્સેસ માહિતી અથવા સિસ્ટમની પરવાનગી અથવા અધિકારો વિના પ્રયાસ કરેલ અથવા સફળ ઍક્સેસ.
- 2 સાયબર જાસૂસી કોમ્પ્યુટરના ઉપયોગ દ્વારા જાસૂસીનું કાર્ય, જેમાં સંવેદનશીલ માહિતી મેળવવા માટે કંપની અથવા સરકારી નેટવર્કની અપ્રગટ ઍક્સેસ અથવા ‘હેકિંગ’ સામેલ છે.
- 3 માલવેર દૂષિત સોફ્ટવેર માટે એક સામૂહિક શબ્દ છે, જેમ કે વાયરસ, વોર્મ્સ અને ટ્રોજન; વ્યાપારી અથવા વિનાશક હેતુઓ માટે સિસ્ટમો અને માહિતીમાં ઘૂસણખોરી કરવા માટે રચાયેલ છે.
- 4 ડેટા લિકેજ ઇરાદાપૂર્વક અથવા આકસ્મિક નુકસાન, ચોરી અથવા સંવેદનશીલ કંપની અથવા વ્યક્તિગત ઍક્સપોઝર માહિતી
- 5 મોબાઇલ ડિવાઇસ એટેક મોબાઇલ ડિવાઇસ અને તેમના દ્વારા સંગ્રહિત અથવા પ્રોસેસ કરેલી માહિતી પર દૂષિત હુમલો અથવા તેની અનધિકૃત ઍક્સેસ; વાયરલેસ રીતે અથવા ભૌતિક કબજા દ્વારા કરવામાં આવે છે.
- 6 સોશિયલ એન્જીનિયરિંગ કંપનીની માહિતી અથવા સિસ્ટમ્સ સુધી પહોંચવા માટે, ફોન, ઈમેલ, ઓનલાઇન અથવા રૂબરૂમાં, સંવેદનશીલ માહિતી જાહેર કરવા માટે અન્ય લોકો સાથે છેતરપિંડી અને ચાલાકી કરવી.

- 7 ઇન્સાઇડર્સ સંવેદનશીલ કંપનીની માહિતીની ચોરી કરવા, છેતરપિંડી કરવા અથવા કંપનીની સિસ્ટમ્સ અથવા માહિતીને નુકસાન પહોંચાડવાના દૂષિત ઇરાદા સાથે કર્મચારી અથવા કાર્યકર.
- 8 ફિશિંગ એ સામાજિક ઇજનેરીનું એક સ્વરૂપ છે, જેમાં સામાન્ય રીતે સિસ્ટમની ઍક્સેસ મેળવવા અથવા ઓળખની ચોરી માટે, પ્રાપ્તકર્તાઓ પાસેથી છેતરપિંડીથી સંવેદનશીલ માહિતી કાઢવાના હેતુથી કાયદેસર દેખાતા ઇમેઇલ્સ મોકલવાનો સમાવેશ થાય છે.
- 9 સિસ્ટમ કોમ્પ્રોમાઇઝ એક સિસ્ટમ કે જેના પર દૂષિત વ્યક્તિઓ અથવા ‘હેક્સ’ દ્વારા હુમલો કરવામાં આવ્યો છે અને તેને કબજે કરવામાં આવ્યો છે, સામાન્ય રીતે એક અથવા વધુ નબળાઈઓના શોષણ દ્વારા, અને પછી ઘણીવાર અન્ય સિસ્ટમ પર હુમલો કરવા માટે તેનો ઉપયોગ કરવામાં આવે છે.
- 9 સ્પામ વણમાગી ઇમેઇલ ઘણી વ્યક્તિઓને જથ્થામાં મોકલવામાં આવે છે, સામાન્ય રીતે વ્યાપારી લાભ માટે, પરંતુ વધુને વધુ માલવેર ફેલાવવા માટે.
- 10 સેવાનો ઇનકાર સિસ્ટમ પર ઇરાદાપૂર્વક અથવા અજાણતા હુમલો તેના પર સંગ્રહિત માહિતી, સિસ્ટમને અનુપલબ્ધ અને અધિકૃત વપરાશકર્તાઓ માટે અપ્રાપ્ય રેન્ડર કરે છે.
- 11 ઓળખની ચોરી અજાણી વ્યક્તિની અંગત માહિતીની ચોરી, છેતરપિંડીથી માની લેવા માટે કે વ્યક્તિની ઓળખ ગુનો કરવા માટે, સામાન્ય રીતે નાણાકીય લાભ માટે

ધમકીઓની શ્રેણીઓ સુરક્ષા ધમકીઓને ઘણી રીતે વર્ગીકૃત કરી શકાય છે. કેટલાક લોકપ્રિય વર્ગીકરણ નીચે મુજબ છે:

1 અભિજાત્યપણના આધારે, સુરક્ષા જોખમોને ત્રણ શ્રેણીઓમાં વર્ગીકૃત કરી શકાય છે.

- પ્રથમ પેઢીના સાદા જોખમો એ સામાન્ય વાયરસ પ્રકારના હુમલા છે જે વપરાશકર્તાઓ દ્વારા ચેપગ્રસ્ત ઈ-મેલ અને અસ્પષ્ટ ફાઇલ જોડાણો ખોલીને ફેલાય છે.
- બીજી પેઢીની ધમકીઓ વધુ સુસંસ્કૃત છે અને મોટી સમસ્યાઓ ઊભી કરે છે. સ્વયંસંચાલિત સાધનો વડે બનાવેલ, આ કીડા માનવીય ક્રિયાપ્રતિક્રિયા વિના નબળાઈઓ પર હુમલો કરે છે. નવા પીડિતોની નકલ, ઓળખ અને લક્ષ્યાંક આપોઆપ છે.
- ત્રીજી પેઢીની ધમકીઓ મિશ્રિત ધમકીઓ છે, સામાન્ય છે અને વાયરસ, ટ્રોજન અને ઓટોમેશનનો સમાવેશ કરે છે. આ વોર્મ્સ હાયપર પ્રચાર માટે લક્ષ્યોને સંકલિત કરે છે, જાણીતી નબળાઈઓનો ઉપયોગ કરે છે અને છુપાયેલા નબળાઈઓનો લક્ષિત ઉપયોગ સક્ષમ કરે છે. તેઓ મલ્ટિપલ એટેક વાયરલેસ લિંક્સ, વર્ચ્યુઅલ પ્રાઈવેટ નેટવર્ક અને ફાયરવોલ અને ઈન્ટ્રુઝન ડિટેક્શન જેવા પેરીમીટર ડિફેન્સની અંદરના હુમલાને પણ નિશાન બનાવે છે. સિસ્ટમો

2 OWASP (ઓપન વેબ એપ્લિકેશન સિક્યુરિટી પ્રોજેક્ટ) અનુસાર ટોચની ધમકીઓ નીચે મુજબ છે:

- ઈન્જેક્શન
- ક્રોસ સાઈટ સ્ક્રીપ્ટિંગ (CSS)
- ટૂટેલી પ્રમાણીકરણ અને સત્ર વ્યવસ્થાપન
- અસુરક્ષિત ડાયરેક્ટ ઓબ્જેક્ટ સંદર્ભો
- ક્રોસ - સાઈટ રિક્વેસ્ટ ફોર્જરી (CSRF)
- સુરક્ષા ખોટી ગોઠવણી
- અસુરક્ષિત ક્રિપ્ટોગ્રાફિક સ્ટોરેજ
- URL એક્સેસને પ્રતિબંધિત કરવામાં નિષ્ફળતા
- અપર્યાપ્ત ટ્રાન્સપોર્ટ લેયર પ્રોટેક્શન
- અમાન્ય રીડાયરેક્ટ અને ફોરવર્ડ

3 ઉપયોગ થાય છે (અથવા હુમલાખોરની પ્રેરણા) એટલે કે શોષણના પ્રકારો અનુસાર માર્કોસોફ્ટ દ્વારા વર્ગીકરણ. STRIDE સિસ્ટમ. STRIDE ટૂંકાક્ષર નીચેની દરેક શ્રેણીના પ્રથમ અક્ષરથી બનેલ છે.

- સ્પૂફિંગ ઓળખ. ઓળખની સ્પૂફિંગનું ઉદાહરણ ગેરકાયદેસર રીતે એક્સેસ કરવું અને પછી અન્ય વપરાશકર્તાની પ્રમાણીકરણ માહિતીનો ઉપયોગ કરવો, જેમ કે વપરાશકર્તા નામ અને પાસવર્ડ.
- ડેટા સાથે છેડછાડ. ડેટા ટેમ્પરિંગમાં ડેટાના દૂષિત ફેરફારનો સમાવેશ થાય છે.
- અસ્વીકાર. અસ્વીકારની ધમકીઓ એવા વપરાશકર્તાઓ સાથે સંકળાયેલી છે જેઓ ક્રિયા કરવાનો ઈનકાર કરે છે અન્ય પક્ષો પાસે અન્યથા સાબિત કરવાની કોઈ રીત ન હોય, ઉદાહરણ તરીકે, વપરાશકર્તા એવી સિસ્ટમમાં ગેરકાયદેસર કામગીરી કરે છે જેમાં પ્રતિબંધિત કામગીરીને ટ્રેસ કરવાની ક્ષમતા નથી.

- માહિતીની જાહેરાત. માહિતી જાહેર કરવાની ધમકીઓમાં એવી વ્યક્તિઓ માટે માહિતીના સંપર્કનો સમાવેશ થાય છે જેમને તેની એક્સેસ ન હોય.
- સેવાનો ઈનકાર. સેવાનો ઈનકાર (DoS) હુમલાઓ માન્ય વપરાશકર્તાઓને સેવા આપવાનો ઈનકાર કરે છે.
- વિશેષાધિકારની ઉન્નતિ. આ પ્રકારની ધમકીમાં, એક બિનપ્રાપ્તિહીત વપરાશકર્તા વિશેષાધિકૃત એક્સેસ મેળવે છે અને ત્યાંથી સમગ્ર સિસ્ટમ સાથે સમાધાન અથવા નાશ કરવા માટે પૂરતી એક્સેસ હોય છે.

ટેકનોલોજી પર આધારિત ધમકીઓ

- 1 WWW ટેકનોલોજી પર આધારિત ધમકીઓ
- 2 બ્રાઉઝર સોફ્ટવેરમાં નવી સુવિધાઓ
- 3 બ્રાઉઝર સોફ્ટવેર ટેસ્ટ વર્ઝન
- 4 સર્વર સોફ્ટવેર
- 5 CGI સ્ક્રિપ્ટો
- 6 ફૂકીઝ
- 7 યુનિક્સ અને TCP/IP ટૂલ્સ પર આધારિત ધમકીઓ
- 8 ફાયરવોલ મેનેજમેન્ટમાં મુશ્કેલીઓ
- 9 ક્રિપ્ટોગ્રાફિક સોફ્ટવેરનો ઉપયોગ
- 10 હેકર સાધનો
- 11 અન્ય સોફ્ટવેર આધારિત ધમકીઓ
- 12 ઈન્ટ્રાનેટ એપ્લિકેશન સોફ્ટવેર
- 13 જાવા ભાષા
- 14 ActiveX
- 15 સંદેશાવ્યવહાર પર આધારિત ધમકીઓ
- 16 વાયરસ પર આધારિત ધમકીઓ
- 17 માનવ પ્રવૃત્તિઓ પર આધારિત ધમકીઓ

માહિતી સુરક્ષા જોખમોની ઓળખ માહિતી સુરક્ષા વ્યવસ્થાપનની સફળતા કાર્યક્રમ સંસ્થાની માહિતી પ્રણાલીઓ માટેના જોખમોની સચોટ ઓળખ પર આધારિત છે.

સુરક્ષા આયોજકો માટે માહિતી સુરક્ષા ધમકીઓની ઓળખ એ આવશ્યક પ્રથમ પગલું છે. યોગ્ય ખતરો અને નબળાઈની ઓળખમાં સુરક્ષા પરીક્ષણો અને તપાસનો સમાવેશ થવો જોઈએ, જે સાધનસામગ્રી યોગ્ય રીતે કાર્યરત છે, જરૂર પડ્યો સહેલાઈથી ઉપલબ્ધ છે અને કર્મચારીઓ સાધનસામગ્રીના ઉપયોગમાં નિપુણ છે તે સુનિશ્ચિત કરવા માટે તૈયાર છે.

આ પરિપૂર્ણ કરવા માટે, સિસ્ટમોએ એક પરીક્ષણ પ્રોગ્રામ ડિઝાઇન કરવો જોઈએ જે માત્ર સુરક્ષાની વર્તમાન સ્થિતિનું જ મૂલ્યાંકન કરતું નથી, પરંતુ તેનો ઉપયોગ તાલીમ દ્વારા સ્ટાફની અસરકારકતાને અપગ્રેડ કરવા માટે પણ થઈ શકે છે.

સુરક્ષા જોખમોને ઓળખવાની બે મુખ્ય પદ્ધતિઓ પ્રોબિંગ અને સ્કેનિંગ છે.

પ્રોબિંગ એ કમ્પ્યુટર સિસ્ટમમાં જાણીતા અથવા સંભવિત નબળા બિંદુ દ્વારા કમ્પ્યુટર અને તેની ફાઇલોની એક્સેસ મેળવવાનો પ્રયાસ છે. તે

નેટવર્કની સ્થિતિ વિશે કંઈક શીખવાના હેતુથી લેવામાં આવેલી ક્રિયા છે. સ્કેનિંગ એ બધી ફાઇલો અથવા નેટવર્ક ઘટકોમાંથી કંઈક અસામાન્ય શોધવાના ઈરાદાથી પસાર થવાની પદ્ધતિ છે.

ફાઇલ સ્કેનિંગ એ ફાઇલોનું નિરીક્ષણ કરે છે જેને વપરાશકર્તાઓ વાયરસ અને અન્ય દૂષિત સામગ્રી માટે દૂરથી ડાઉનલોડ અથવા ખોલવાનો પ્રયાસ કરે છે. ફાઇલ સ્કેનિંગ નીતિના અમલીકરણ માટે કેટલીક માહિતી પરત કરે છે.

ફાઇલ સ્કેનિંગના 2 પ્રકાર છે. તેઓ એકસાથે વાપરી શકાય છે.

- અદ્યતન શોધ વાઈરસ, ટ્રોજન હોર્સ, વોર્મ્સ અને અન્ય સહિત જાણીતા અને ઉભરતા જોખમોને શોધવા માટેની તકનીકો લાગુ કરે છે.
- એન્ટિ-વાયરસ સ્કેનિંગ વાયરસથી ચેપગ્રસ્ત ફાઇલોને ઓળખવા માટે એન્ટિ-વાયરસ વ્યાખ્યા ફાઇલોનો ઉપયોગ કરે છે.

નેટવર્ક સ્કેનિંગ એ નેટવર્ક સુરક્ષા મૂલ્યાંકનના હેતુ માટે નેટવર્ક પર સક્રિય હોસ્ટને ઓળખવા માટેની પ્રક્રિયા છે. સ્કેનિંગ પ્રક્રિયાઓ, જેમ કે પિંગ સ્વીપ્સ અને પોર્ટ સ્કેન, ઈન્ટરનેટ પર સક્રિય રહેલા જીવંત યજમાનો અને કોઈપણ શંકાસ્પદ પ્રવૃત્તિ વગેરેને કયા IP સરનામાઓ મેપ કરે છે તે વિશેની માહિતી પરત કરે છે.

સિસ્ટમ પર હુમલો કરવાની ધમકીઓની મોડસ ઓપરેન્ડી સામાન્ય રીતે ઉપકરણ પર હુમલો 3 તબક્કામાં થાય છે:

- આ યજમાનનો ચેપ,
- તેના ધ્યેયની સિદ્ધિ, અને
- અન્ય સિસ્ટમોમાં માલવેરનો ફેલાવો.

ધમકીઓ વારંવાર ચેપગ્રસ્ત ઉપકરણો દ્વારા ઓફર કરાયેલા સંસાધનોનો ઉપયોગ કરે છે. તેઓ બ્લૂટૂથ અથવા ઈન્ફ્રારેડ જેવા આઉટપુટ ઉપકરણોનો

ઉપયોગ કરે છે, પરંતુ વપરાશકર્તાના પરિચિતોને સંક્રમિત કરવા માટે વ્યક્તિની સરનામા પુસ્તિકા અથવા ઈમેઇલ સરનામાનો પણ ઉપયોગ કરી શકે છે.

તેઓ વિશ્વાસનું શોષણ કરે છે જે કોઈ પરિચિત દ્વારા મોકલવામાં આવેલા ડેટાને આપવામાં આવે છે.

ચેપ ચેપ એ ઉપકરણમાં પ્રવેશવા માટે ધમકી દ્વારા ઉપયોગમાં લેવાતું માધ્યમ છે. તે કાં તો અગાઉ પ્રસ્તુત કરેલા ખામીઓમાંથી એકનો ઉપયોગ કરી શકે છે અથવા વપરાશકર્તાની ભૂલનો ઉપયોગ કરી શકે છે. ચેપ પરવાનગી માંગી શકે છે, અથવા અસ્પષ્ટ ઉપયોગકર્તા સાથે ક્રિયાપ્રતિક્રિયા કરી શકે છે અથવા આ બેમાંથી કોઈ પણ કરી શકશે નહીં અને સિસ્ટમ પર સીધો હુમલો કરી શકે છે.

ધ્યેયની પરિપૂર્ણતા એકવાર ધમકીએ ઉપકરણને સંક્રમિત કર્યા પછી તે તેના ધ્યેયને પૂર્ણ કરવાનો પ્રયાસ કરશે, જે સામાન્ય રીતે નીચેનામાંથી એક છે:

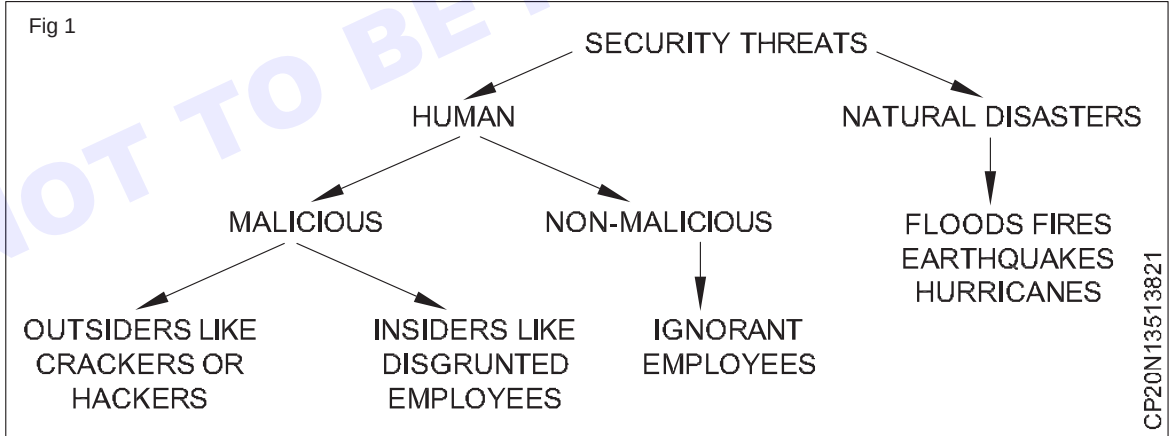
હાર્ડવેર નુકસાન, સેવાનો ઈનકાર(DoS), નાણાકીય નુકસાન, નુકસાન ડેટા અને/અથવા , ઉપકરણ અને છુપાયેલ નુકસાન વગેરે.

અન્ય પ્રણાલીઓમાં ફેલાવો એકવાર ધમકીએ ઉપકરણને ચેપ લગાવી દીધા પછી, તે હંમેશા એક અથવા બીજી રીતે ફેલાવવાનું લક્ષ્ય રાખે છે.

તે નેટવર્ક, વાયર્ડ અથવા વાયરલેસ, ઈન્ટરનેટ દ્વારા, વાઈફાઈ, બ્લૂટૂથ અને ઈન્ફ્રારેડ લાઈટનો ઉપયોગ કરીને નિકટવર્તી ઉપકરણો અને શેર કરેલ ઉપકરણો વગેરે દ્વારા ફેલાય છે.

ધમકીઓના સ્ત્રોત (ફિગ 1)

ધમકીઓના પ્રાથમિક સ્ત્રોત કર્મચારીઓ/આંતરિક છે, દૂષિત હેક્સર્સ, કુદરતી આફતો, વિદેશી વિરોધીઓ અને પ્રતિકૂળ હુમલાઓ. કેટલાક કિસ્સાઓમાં, ધમકીઓના સ્ત્રોત માટેના વિસ્તારો ઓવરલેપ થઈ શકે છે. ઉદાહરણ તરીકે, વિદેશી વિરોધીઓ અથવા અસંતુષ્ટ કર્મચારી દ્વારા પ્રતિકૂળ હુમલાઓ કરવામાં આવી શકે છે.



કુદરતી આફતો: ઘરતીકંપ, વાવાઝોડું, પૂર, વીજળી અને આગ કમ્પ્યુટર સિસ્ટમને ભારે નુકસાન પહોંચાડી શકે છે. માહિતી ગુમ થઈ શકે છે, સમય ઓછો થઈ શકે છે અથવા ઉત્પાદકતામાં ઘટાડો થઈ શકે છે, અને હાર્ડવેરને નુકસાન અન્ય આવશ્યક સેવાઓને વિક્ષેપિત કરી શકે છે. કુદરતી આફતો સામે થોડાં રક્ષકોનો અમલ કરી શકાય છે. શ્રેષ્ઠ અભિગમ એ આપત્તિ પુનઃપ્રાપ્તિ યોજનાઓ અને આકસ્મિક યોજનાઓ છે.

કર્મચારીઓ/અંદરથી માનવીય ધમકીઓ: કર્મચારીઓ અને આંતરિક લોકો દ્વારા ઈરાદાપૂર્વકની અને આકસ્મિક ભૂલો અને દૂષિત કૃત્યો ટેલિકોમ્યુનિકેશન ઉદ્યોગમાં નોંધપાત્ર પ્રમાણમાં નુકસાન અને નુકસાન પહોંચાડે છે, અસંતુષ્ટ કર્મચારીઓ કમ્પ્યુટર સિસ્ટમ પર તોફાન અને તોડફોડ

બંને કરી શકે છે. જાહેર અને ખાનગી બંને ક્ષેત્રોમાં તેમની નોકરીમાંથી દૂર કરાયેલા કર્મચારીઓએ મહત્વપૂર્ણ સંસ્થાકીય જ્ઞાન ધરાવતા વ્યક્તિઓનું એક જૂથ બનાવ્યું છે જે સંભવિત સિસ્ટમ એક્સેસ જાળવી શકે છે. સિસ્ટમ મેનેજર્સ પાસવર્ડને અમાન્ય કરીને અને સમયસર સિસ્ટમ એકાઉન્ટ્સ કાઢી નાખીને આ જોખમને મર્યાદિત કરી શકે છે.

દૂષિત હેક્સર્સ દૂષિત ધમકીઓમાં અસંતુષ્ટ અથવા દૂષિત કર્મચારીઓ દ્વારા આંતરિક હુમલાઓ અને બિન કર્મચારીઓ દ્વારા ફક્ત સંસ્થાને નુકસાન પહોંચાડવા અને વિક્ષેપિત કરવા માટેના બહારના હુમલાઓનો સમાવેશ થાય છે. દૂષિત હુમલાખોરો સામાન્ય રીતે સિસ્ટમ પરના હુમલા માટે ચોક્કસ ધ્યેય, ઉદ્દેશ્ય અથવા હેતુ ધરાવતા હોય છે

આ ધ્યેયો ડિનાયલ-ઓફ સર્વિસ (DoS) એટેક ટૂલ્સનો ઉપયોગ કરીને સેવાઓ અને વ્યવસાયિક કામગીરીની સાતત્યને વિદ્વેષિત કરવાના હોઈ શકે છે. તેઓ માહિતીની ચોરી કરવા અથવા લેપટોપ કોમ્પ્યુટર જેવા હાર્ડવેરની પણ ચોરી કરવા માંગે છે. હેકર્સ એવી માહિતી વેચી શકે છે જે સ્પર્ધકો માટે ઉપયોગી થઈ શકે.

સૌથી ખતરનાક હુમલાખોરો સામાન્ય રીતે આંતરિક (અથવા ભૂતપૂર્વ આંતરિક) હોય છે, કારણ કે તેઓ ઘણા બધા કોડ્સ અને સુરક્ષા પગલાં જાણે છે જે પહેલાથી જ છે. અંદરના લોકો પાસે ચોક્કસ ધ્યેયો અને ઉદ્દેશ્યો હોવાની અને સિસ્ટમમાં કાયદેસરની ઍક્સેસ હોવાની શક્યતા છે. કર્મચારીઓ એ સંસ્થાના કમ્પ્યુટર્સ અને એપ્લિકેશનોથી સૌથી વધુ પરિચિત લોકો છે, અને તેઓ જાણતા હોય છે કે કઈ ક્રિયાઓ સૌથી વધુ નુકસાન પહોંચાડી શકે છે. અંદરના લોકો વાયરસ, ટ્રોજન હોર્સ અથવા વોર્મ્સ રોપી શકે છે અને તેઓ ફાઈલ સિસ્ટમ દ્વારા બ્રાઉઝ કરી શકે છે.

જો કે, અસંતુષ્ટ વર્તમાન કર્મચારીઓ વાસ્તવમાં ભૂતપૂર્વ કર્મચારીઓ કરતાં વધુ નુકસાન પહોંચાડે છે. કમ્પ્યુટર-સંબંધિત કર્મચારીની તોડફોડના સામાન્ય ઉદાહરણોમાં નીચેનાનો સમાવેશ થાય છે:

- ડેટા બદલવો
- ડેટા કાઢી રહ્યા છીએ
- લોજિક બોમ્બ વડે ડેટા અથવા પ્રોગ્રામ્સનો નાશ કરવો
- કેશિંગ સિસ્ટમ્સ
- ડેટાને બાનમાં રાખવો
- હાર્ડવેર સુવિધાઓનો નાશ કરવો
- ખોટી રીતે ડેટા દાખલ કરવો

વિદેશી વિરોધીઓ

કોમ્પ્યુટર ઘુસણખોર પ્રવૃત્તિઓ આંતરરાષ્ટ્રીય સ્તરે થઈ છે, આંતરરાષ્ટ્રીય ગેટવે દ્વારા ઘુસણખોરીના પ્રયાસોની સંખ્યા ચિંતાજનક દરે વધી રહી છે. એવા થોડા સંકેતો મળ્યા છે કે વિદેશી દેશોની અંદર કોમ્પ્યુટર રાજકીય એજન્ડાઓનું સંચાલન કરે છે. કેટલીકવાર એક દેશની ગુપ્તચર સેવાઓ અન્ય દેશોની સંદેશાવ્યવહાર પ્રણાલીઓને સીધી રીતે લક્ષ્ય બનાવે છે, તેમાં પ્રવેશ કરે છે અથવા સમાધાન કરે છે.

બહારના હુમલાખોરો અથવા ‘ફટાકડા’

લોકો ઘણીવાર “ફટાકડાને” “હેકર્સ” તરીકે ઓળખે છે. હેકર શબ્દ એવા લોકોનો ઉલ્લેખ કરે છે કે જેઓ કાં તો એવી સિસ્ટમમાં પ્રવેશ કરે છે કે જેના માટે તેમની પાસે કોઈ અધિકૃતતા નથી અથવા ઈરાદાપૂર્વક એવી સિસ્ટમ્સ પર તેમની સીમાઓ વટાવે છે કે જેના માટે તેમની પાસે કાયદેસરની ઍક્સેસ નથી. સિસ્ટમમાં પ્રવેશ મેળવવા માટેની સામાન્ય પદ્ધતિઓમાં પાસવર્ડ કેકીંગ, જાણીતી સુરક્ષા નબળાઈઓનું શોષણ, નેટવર્ક સ્પૂફિંગ અને સામાજિક ઈજનેરી.

પ્રતિકૂળ હુમલાઓ

પ્રતિકૂળ હુમલાઓ દ્વારા, નેટવર્કની ઉપલબ્ધતાને અસર કરવી શક્ય છે. પ્રતિકૂળ હુમલાઓની પ્રાથમિક અસર જેમ કે સંકલિત પરમાણુ હુમલા, મર્યાદિત/અસંકલિત પરમાણુ હુમલા, પરમાણુ અકસ્માતો, આતંકવાદ, ઈલેક્ટ્રોનિક યુદ્ધ, તોડફોડ અને રાષ્ટ્રના નેટવર્ક પર નાગરિક અવ્યવસ્થા વિદ્વેષ અને સેવાનો ઈનકાર છે. આવી આપત્તિઓ વિતરિત સેવાઓની સમયસરતા અને ગુણવત્તાને અસર કરે છે

B બિન-દૂષિત કર્મચારીઓ

હુમલાખોરો જ એવા નથી કે જેઓ સંસ્થાને નુકસાન પહોંચાડી શકે. ડેટા અખંડિતતા માટે પ્રાથમિક ખતરો અજ્ઞાન વપરાશકર્તાઓ તરફથી આવે છે. આ અધિકૃત વપરાશકર્તાઓ છે જેઓ તેઓ જે ક્રિયાઓ કરી રહ્યા છે તેનાથી વાકેફ નથી. ભૂલો અને ભૂલો મૂલ્યવાન ડેટા ગુમાવી શકે છે, નુકસાન પહોંચાડી શકે છે અથવા બદલી શકે છે.

વપરાશકર્તાઓ, ડેટા એન્ટ્રી ક્લાર્ક, સિસ્ટમ ઓપરેટરો અને પ્રોગ્રામરો વારંવાર અજાણતાં ભૂલો કરે છે જે પ્રત્યક્ષ અને પરોક્ષ રીતે સુરક્ષા સમસ્યાઓમાં ફાળો આપે છે. કેટલીકવાર ભૂલ એ ધમકી છે, જેમ કે ડેટા એન્ટ્રી ભૂલ અથવા પ્રોગ્રામિંગ ભૂલ જે સિસ્ટમને કેશ કરે છે. અન્ય કિસ્સાઓમાં, ભૂલો નબળાઈઓ બનાવે છે.

ધમકીઓને ઓળખવા માટે વપરાતી શ્રેષ્ઠ પદ્ધતિઓ અથવા માર્ગદર્શિકાઓ

વિવિધ પદ્ધતિઓ અને પદ્ધતિઓનો ઉપયોગ તેમના પ્રકાર પર આધાર રાખીને ધમકીઓ/ હુમલાઓને સફળતાપૂર્વક ઓળખવા માટે કરી શકાય છે. બીજા શબ્દોમાં કહીએ તો, ધમકીના આધારે, તમે તેને ઓળખવા અને તે મુજબ વર્ગીકૃત કરવા માટે ચોક્કસ તકનીકોનો ઉપયોગ કરી શકો છો. નીચેની સૌથી સામાન્ય પદ્ધતિઓ છે:

- વિસંગતતા શોધ સાધનોનો ઉપયોગ
- નેટવર્ક ફ્લો-આધારિત વિશ્લેષણનો ઉપયોગ કરીને ટેલિમેટ્રી
- ઘૂસણખોરી શોધ અને ઘૂસણખોરી નિવારણ સિસ્ટમ્સ (IDS/IPS) નો ઉપયોગ
- નેટવર્ક કમ્પોનન્ટ લોગ્સનું પૃથ્થકરણ કરવું (એટલે કે, વિવિધ નેટવર્ક ઉપકરણોમાંથી SYSLOG, એકાઉન્ટિંગ રેકોર્ડ્સ, એપ્લિકેશન લોગ્સ, સિમ્પલ નેટવર્ક મેનેજમેન્ટ પ્રોટોકોલ (SNMP) વગેરે.

ધમકીઓ ઘટાડવા માટે વપરાતી શ્રેષ્ઠ પ્રથાઓ અથવા માર્ગદર્શિકા નીચેની કેટલીક પ્રથાઓ છે જેનો અમલ કરવામાં આવે ત્યારે સંસ્થાની માહિતી, ડેટા અને વિશ્વસનીયતાની સુરક્ષા માટેના જોખમો ઘટે છે.

સુરક્ષા જાગૃતિ તાલીમ

- મોટાભાગની સુરક્ષા ભંગ વાસ્તવમાં કંપનીઓની અંદર અસંતુષ્ટ અથવા બેદરકાર કર્મચારીઓ દ્વારા ઉદ્ભવે છે. તેથી તમારી કંપનીમાં દરેકને શિક્ષિત કરો જેથી તેઓ વિવિધ સુરક્ષા જોખમોને ઓળખવામાં મદદ કરી શકે.
- કર્મચારીઓ ઈમેલ ફિશિંગ અને સ્પૂફિંગ હુમલાઓને શોધી અને ઓળખવામાં સક્ષમ હોવા જોઈએ. તેમને એનક્રિપ્ટેડ ન હોય તેવી સંવેદનશીલ માહિતીનો સંગ્રહ, મોકલવા અથવા નકલ ન કરવા માટે પણ તાલીમ આપવી જોઈએ.
- તેઓએ જાણવું જોઈએ કે તેઓ ફોન પર સંવેદનશીલ માહિતી શેર ન કરે સિવાય કે તેઓ 100% પ્રેક્ષકોની ખાતરી.
- કર્મચારીઓને સુરક્ષા નીતિઓ અને વ્યવહારો પર તાલીમ આપો. અને તેમને અપડેટ કરવાની ખાતરી કરો અને વારંવાર ફરીથી તાલીમ આપો.
- તાલીમ સામગ્રીઓએ કોર્પોરેટ નીતિઓની પણ સમીક્ષા કરવી જોઈએ અને કર્મચારીઓ વચ્ચેના કોઈપણ શંકાસ્પદ અથવા દૂષિત વર્તણૂક માટેના પરિણામોની સ્પષ્ટ વિગતો આપવી જોઈએ. તેમને વિવિધ સુરક્ષા નીતિઓ પર તાલીમ આપવી જોઈએ, જેમાં નીચેનાનો સમાવેશ થાય છે: - સ્વીકાર્ય ઉપયોગ અને અસ્વીકાર્ય ઉપયોગ

- સામાન્ય ઉપયોગ અને માલિકી
- સુરક્ષા અને માલિકીની માહિતી
- બ્લોગિંગ અને સોશિયલ મીડિયા
- અમલીકરણ એટલે કે પાલન ન કરવા બદલ શિસ્તની કાર્યવાહી
- સામાજિક મીડિયા
- મારી પોતાની ઉપકરણ નીતિઓ લાવો
- ડેટા નીતિ
- મોબાઇલ ડિવાઇસ મેનેજમેન્ટ (MDM) પોલિસી
- મોબાઇલ ડિવાઇસ સપોર્ટ પોલિસી
- કંપની દ્વારા જારી કરાયેલા ઉપકરણોને લગતી નીતિઓ
- નુકશાન અને ચોરી
- કર્મચારી સમાપ્તિ નીતિ
- સુરક્ષા ઘટના વ્યવસ્થાપન

એન્ટિ-વાયરસ અને એન્ટિ-માલવેર પ્રોટેક્શન

- માલવેરના સામાન્ય સ્વરૂપોમાં નીચેનાનો સમાવેશ થાય છે: વોર્મ્સ, કી લોગર્સ, વિડીયો ફ્રેમ ગ્રેબર્સ, રુટકીટ અને ટ્રોજન હોર્સ.
- સારો એન્ટિવાયરસ ઇન્સ્ટોલ કરો, અપડેટ કરો, શેડ્યૂલ કરો અને ચલાવો કાર્યક્રમો
- લડવા માટે “અંતિમ બિંદુ સુરક્ષા” વ્યૂહરચના અપનાવો માલવેર ધમકીઓ. એન્ડપોઇન્ટ સિક્યોરિટી એ માહિતી સુરક્ષા ખ્યાલ છે જેનો અર્થ છે કે નેટવર્ક પરના દરેક ઉપકરણ (અથવા એન્ડપોઇન્ટ) તેની પોતાની સુરક્ષા માટે જવાબદાર અને સક્ષમ હોવા જોઈએ.
- તમારું એન્ટિ-માલવેર સોલ્યુશન ગમે તે હોય, તેણે જોડાયેલ વાઈરસ માટે ઈમેલ સ્કેન કરવું જોઈએ, ચેપ માટે રીઅલ ટાઈમમાં ફાઈલોનું મોનિટર કરવું જોઈએ અને દરેક ફાઈલનું સંપૂર્ણ સ્કેન કરવું જોઈએ. ડેટા એન્ક્રિપ્શન
- ડેટા એન્ક્રિપ્શન એ માહિતી સુરક્ષાનો એક શક્તિશાળી ભાગ છે. એન્ક્રિપ્શન તમારા ડેટાને એક્સેસ કર્યા પછી પણ તેને સુરક્ષિત કરે છે. પેચીંગ
- તમારી કોમ્પ્યુટર સિસ્ટમો પરના જોખમને ઘટાડવા માટે પેચીંગ આવશ્યક છે. સિસ્ટમો અને એપ્લિકેશનમાં સુરક્ષા છિદ્રોને ઠીક કરવા માટે ઘણીવાર પેચ છોડવામાં આવે છે. ખાતરી કરો કે તમે ચલાવો છો તે બધી ઓપરેટિંગ સિસ્ટમ્સ અને એપ્લિકેશનોને પેચ કરેલી રાખો છો. બધા નેટવર્ક ઉપકરણો પર નવીનતમ ફર્મવેર અપડેટ્સ ઇન્સ્ટોલ કરો.

એક્સેસ નિયંત્રણો

- વધેલી સુરક્ષા માટે, કર્મચારીઓને (અને ભાગીદારોને) તેઓને જોઈતા ડેટાની એક્સેસ આપો. આમાં ભૌતિક અને તાર્કિક એક્સેસ બંનેનો સમાવેશ થાય છે.
- ઓછામાં ઓછો વિશેષાધિકાર આપીને શરૂઆત કરો. પછી તમે અનધિકૃત ડેટાની એક્સેસની મંજૂરી આપવા માટે વિશેષાધિકારો વધારી શકો છો.

મોબાઇલ ઉપકરણો

- લેપટોપ, સ્માર્ટફોન અને ટેબ્લેટ્સે આજના કર્મચારીઓની ઉત્પાદકતા અને ગતિશીલતામાં વધારો કર્યો છે. પરંતુ તેની સાથે ઉત્પાદકતા પણ આવે છે. ખોવાયેલા અથવા ચોરાયેલા લેપટોપ અને અન્ય મોબાઇલ ઉપકરણો ડેટા ભંગનું મુખ્ય કારણ છે.
- ઓટો-લોક સક્ષમ કરો અથવા બધા ઉપકરણોને એક્સેસ કરવા માટે પાસવર્ડની જરૂર છે.

મોનીટરીંગ

- ખાતરી કરો કે તમારો વ્યવસાય કોઈપણ અસાધારણતા માટે સિસ્ટમો અને નેટવર્ક ઉપકરણોને મોનિટર કરવા માટે સેટઅપ થયેલ છે.
- તમામ સ્થાનો અથવા ઈન્ફ્રાસ્ટ્રક્ચર - નેટવર્ક, સિસ્ટમ્સ અને વપરાશકર્તા પ્રવૃત્તિમાંથી માહિતી એકત્રિત કરો અને સહસંબંધિત કરો.
- માત્ર ફાયરવોલ અથવા IPS પર પ્રવૃત્તિને અવરોધિત કરશો નહીં. તેને લોગ કરો, તેની સમીક્ષા કરો અને તેમાંથી શીખો.
- તમારા વ્યવસાયની અંદરથી વપરાશકર્તાની પ્રવૃત્તિને મોનિટર કરવા માટે સામગ્રી ફિલ્ટરિંગ ઇન્સ્ટોલ કરો. ઇન્ટરનેટના કર્મચારીઓના દુરુપયોગનું સૌથી સામાન્ય સ્વરૂપ અનિચ્છનીય સાઈટ્સ ફાયરવોલ પર સર્ફ કરવાનું છે ફાયરવોલ નિયમો અને નીતિઓ ગોઠવો કારણ કે ફાયરવોલ કોઈપણ હુમલા (નેટવર્ક અથવા હોસ્ટ) સામે સંરક્ષણની પ્રથમ લાઈન છે. તે જાહેર નેટવર્ક અને ખાનગી નેટવર્ક વચ્ચે અવરોધનું કામ કરે છે.

દૂરસ્થ બેકઅપ

- તમારા ડેટાનો નિયમિતપણે દૂરસ્થ સ્થાન પર બેકઅપ લો. બેકઅપ એ કમ્પ્યુટિંગના સૌથી ઉપેક્ષિત ક્ષેત્રોમાંનું એક છે અને તેથી સામાન્ય રીતે તમારા વ્યવસાયમાં જોખમ ઘટાડવા માટેની સૌથી મોટી તકોમાંથી એક છે.
- ઘણીવાર, વ્યવસાયો હેકર્સ અથવા માલવેરથી ડેટા સુરક્ષિત કરવા માટે રોકાણ કરે છે, પરંતુ પછી માહિતી કુદરતી કારણો દ્વારા ભૌતિક રીતે નાશ પામે છે. જો ડેટા અસ્તિત્વમાં નથી, તો તેને બહારના જોખમોથી સુરક્ષિત કરવાનો કોઈ અર્થ નથી.

સુરક્ષા મૂલ્યાંકન અને ઘૂસપેંઠ પરીક્ષણ

- તમારા વ્યવસાયને સુરક્ષિત કરવા માટે તમારે જાગ્રત રહેવું જોઈએ. ખોટા ઇરાદાવાળા લોકો હંમેશા તમારા વ્યવસાયની માહિતી સાથે ચેડા કરવાની આગલી રીત શોધતા હોય છે
- નવા જોખમોને ઓળખવા માટે વાર્ષિક અથવા વધુ સારી, ત્રિમાસિક નબળાઈ આકારણીઓ કરો. સતત બદલાતું સુરક્ષા વાતાવરણ હંમેશા નવા જોખમો સર્જે છે.
- તમારા વ્યવસાય પર લાગુ થતા નવા જોખમોને ઓળખો અને અન્ય કોઈ તેને શોધે તે પહેલાં તેને ઠીક કરો.
- દર ત્રણ વર્ષે ઔપચારિક માહિતી સુરક્ષા જોખમ મૂલ્યાંકન મેળવો, જે આ મોટાભાગના ઉત્પાદનોનું જીવન ચક્ર દિવસ છે..

માહિતી સુરક્ષા નબળાઈઓ અને જોખમ વ્યવસ્થાપન (Information security vulnerabilities and Risk Management)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- સુરક્ષા નબળાઈનું વર્ણન કરો
- નબળાઈઓના પ્રકારોનું વર્ણન કરો
- ધમકીઓ સિસ્ટમને કેવી રીતે અસર કરે છે તે સમજાવો
- નબળાઈ આકારણીના સાધનો અને તકનીકોનું વર્ણન કરો
- સુરક્ષા નબળાઈઓને ઘટાડવા માટે શ્રેષ્ઠ પ્રયાસોનું વર્ણન કરો
- ધમકી, નબળાઈ અને જોખમ વચ્ચેના સંબંધનું વર્ણન કરો
- વિવિધ જોખમી એજન્ટોનું વર્ણન કરો
- જોખમ સંચાલનના હેતુનું વર્ણન કરો
- જોખમ સંચાલનના પ્રકારોનું વર્ણન કરો
- ICT સપ્લાય ચેઇન મેનેજમેન્ટના જોખમો સમજાવો.

પરિચય

કમ્પ્યુટર સુરક્ષામાં, નબળાઈ એ સોફ્ટવેર, હાર્ડવેર, પ્રક્રિયાગત અથવા માનવીય નબળાઈ છે જે હુમલાખોરને સિસ્ટમની માહિતી ખાતરીને ઘટાડવાની મંજૂરી આપે છે.

નબળાઈ એ એક નબળાઈ છે જે હુમલાખોરને કમ્પ્યુટર અથવા નેટવર્કમાં પ્રવેશવા માટે અને પર્યાવરણની અંદરના સંસાધનોની અનધિકૃત ઍક્સેસ મેળવવા માટે તે શોધી રહેલા ખુલ્લા દરવાજાને પ્રદાન કરી શકે છે. તે સલામતી માપદંડની ગેરહાજરી અથવા નબળાઈનું પ્રતિનિધિત્વ કરે છે જેનો ઉપયોગ કરી શકાય છે.

ધમકી, ધમકી એજન્ટ, શોષણ, જોખમ અને નબળાઈ.

- ખતરો એ માહિતી અથવા સિસ્ટમો માટે કોઈપણ સંભવિત જોખમ છે. ધમકી એ છે કે કોઈ વ્યક્તિ, અથવા કંઈક, ચોક્કસ નબળાઈને ઓળખશે અને તેનો ઉપયોગ કંપની અથવા વ્યક્તિ સામે કરશે.
- જે એન્ટિટી નબળાઈનો લાભ લે છે તેને ધમકી એજન્ટ તરીકે ઓળખવામાં આવે છે.
- શોષણ એ નબળાઈનો લાભ લેવાનું અને સિસ્ટમ અથવા નેટવર્કનો લાભ લેવા માટે તેનો ઉપયોગ કરવાનું માધ્યમ છે.
- જોખમ એ જોખમ એજન્ટની નબળાઈનો લાભ લેવાની સંભાવના છે અને તેને અનુરૂપ વ્યવસાય અસર.

બીજા શબ્દોમાં કહીએ તો, ખતરો એ છે જેની સામે આપણે રક્ષણ કરવાનો પ્રયાસ કરી રહ્યા છીએ, નબળાઈ એ આપણા સંરક્ષણ પ્રયાસોમાં નબળાઈ અથવા અંતર છે.

માહિતી સુરક્ષા નબળાઈઓ શા માટે અસ્તિત્વમાં છે?

કોડ અથવા નેટવર્કિંગ પ્રોટોકોલ્સના શોષણને કારણે નબળાઈઓ અસ્તિત્વમાં છે. ઓપરેટિંગ સિસ્ટમ બનાવવા માટે કોડની લાખો રેખાઓ જરૂરી છે, અને કેટલીકવાર નબળાઈઓ અંદર મળી શકે છે. જો ફાયરવોલમાં ઘણાબધા પોર્ટ ખુલ્લા હોય, તો ઘુસણખોર અનધિકૃત પદ્ધતિથી નેટવર્કને ઍક્સેસ કરવા માટે એકનો ઉપયોગ કરશે તેવી સંભાવના વધારે છે. નબળાઈ “હાર્ડવેર, સોફ્ટવેર અથવા પ્રક્રિયામાં ખામી અથવા નબળાઈનું પરિણામ હોઈ શકે છે જે સિસ્ટમને સમાધાન માટે ખુલ્લી પાડે છે”. તે નબળાઈ, ડિઝાઇન અથવા અમલીકરણની ભૂલનું અસ્તિત્વ છે જે અણધારી તરફ દોરી શકે છે, ની સુરક્ષા સાથે ચેડા કરતી અનિચ્છનીય ઘટના કમ્પ્યુટર સિસ્ટમ, નેટવર્ક, ઍપ્લિકેશન અથવા પ્રોટોકોલ સામેલ છે.

તકનીકી નબળાઈઓના પ્રકાર

મોટાભાગની સોફ્ટવેર સુરક્ષા નબળાઈઓ એક નાનામાં આવે છે શ્રેણીઓનો સમૂહ:

- બફર ઓવરફ્લો
- અમાન્ય ઈનપુટ
- રેસ શરતો
- ઍક્સેસ-નિયંત્રણ સમસ્યાઓ
- પ્રમાણીકરણ, અધિકૃતતા, અથવા માં નબળાઈઓ

ક્રિપ્ટોગ્રાફિક પ્રેક્ટિસ બફર ઓવરફ્લો બફર ઓવરફ્લો, જેનો મુખ્ય સ્ત્રોત માનવામાં આવે છે નબળાઈઓ, ત્યારે થાય છે જ્યારે ઍપ્લિકેશન લખવાનો પ્રયાસ કરે છે બફરના અંતની બહારનો ડેટા (અથવા ક્યારેક ક્યારેક, શરૂઆત પહેલાં). બફર ઓવરફ્લો ઍપ્લિકેશનને કેશ થવાનું કારણ બની શકે છે, ડેટા સાથે ચેડા કરી શકે છે અને જે સિસ્ટમ પર ઍપ્લિકેશન ચાલી રહી છે તેની સાથે સમાધાન કરવા માટે વધુ વિશેષાધિકાર વધારવા માટે એટેક વેક્ટર પ્રદાન કરી શકે છે. બફર ઓવરફ્લો હુમલા સામાન્ય રીતે સ્ટેક, ધ સાથે સમાધાન કરીને થાય છે ઢગલો, અથવા બંને.

અમાન્ય ઈનપુટ

સામાન્ય નિયમ તરીકે, ડેટા વાજબી અથવા માન્ય છે તેની ખાતરી કરવા માટે તમારે તમારા પ્રોગ્રામ દ્વારા પ્રાપ્ત થયેલ તમામ ઈનપુટ તપાસવા જોઈએ. એવા કિસ્સામાં જ્યાં અમાન્ય ડેટાને મંજૂરી આપવામાં આવી છે અથવા સ્વીકારવામાં આવી છે, આવી ફાઈલને વાંચવાનો પ્રયાસ કરતો સામાન્ય પ્રોગ્રામ ખોટા કદના બફરને ફાળવવાનો પ્રયાસ કરશે, જેનાથી ઢગલો ઓવરફ્લો થવાની સંભાવના છે. હુમલો અથવા અન્ય સમસ્યા. આ કારણોસર, તમારે તમારા ઈનપુટ ડેટાને કાળજીપૂર્વક તપાસવું આવશ્યક છે. આ પ્રક્રિયાને સામાન્ય રીતે ઈનપુટ વેલિડેશન અથવા સેનિટી ચેકિંગ તરીકે ઓળખવામાં આવે છે. તમારા પ્રોગ્રામ દ્વારા અવિશ્વસનીય પાસેથી પ્રાપ્ત થયેલ કોઈપણ ઈનપુટ સ્ત્રોત હુમલા માટે સંભવિત લક્ષ્ય છે. હેકર્સ પ્રોગ્રામના ઈનપુટના દરેક સ્ત્રોતને જુએ છે અને પસાર કરવાનો પ્રયાસ કરે છે તેઓ કલ્પના કરી શકે તેવા દરેક પ્રકારના દૂષિત ડેટામાં. જો પ્રોગ્રામ કેશ થાય છે અથવા અન્યથા ગેરવર્તણૂક કરે છે, હેકર પછી સમસ્યાનું શોષણ કરવાનો માર્ગ શોધવાનો પ્રયાસ કરે છે.

રેસ શરતો

રેસની સ્થિતિ ત્યારે થાય છે જ્યારે એપ્લિકેશનમાં નિયમિત પ્રોગ્રામિંગ કોલ્સની જોડી ક્રમિક રીતે પરફોર્મ કરતી નથી

જે રીતે નિયમો મુજબ હેતુ હતો. તે એક સમય છે

સોફ્ટવેરની અંદરની ઘટના જે સુરક્ષા બની શકે છે

જો કોલ્સ યોગ્ય ક્રમમાં કરવામાં ન આવે તો નબળાઈ. જો અમલનો સાચો ક્રમ જરૂરી છે

પ્રોગ્રામની યોગ્ય કામગીરી, આ એક ભૂલ છે. જો એન

હુમલાખોર દાખલ કરવા માટે પરિસ્થિતિનો લાભ લઈ શકે છે

દૂષિત કોડ, ફાઈલનામ બદલો અથવા અન્યથા પ્રોગ્રામની સામાન્ય કામગીરી, રેસની સ્થિતિ સાથે દખલ કરે છે

સુરક્ષા નબળાઈ છે. હુમલાખોરો ક્યારેક લઈ શકે છે

કોડની પ્રક્રિયામાં નાના સમયના અંતરનો લાભ ઓપરેશનના ક્રમમાં દખલ કરવા માટે, જેનો તેઓ પછી શોષણ કરે છે.

સોફ્ટવેર ડેવલપમેન્ટમાં, ઉપયોગના સમયની તપાસનો સમય (TOC-TOU અથવા TOCTOU, “TOCK too” ઉચ્ચારવામાં આવે છે) એ શરતની ચકાસણી (જેમ કે સુરક્ષા ઓળખપત્ર) અને ઉપયોગ વચ્ચેની સિસ્ટમમાં ફેરફારને કારણે સોફ્ટવેર બગનો વર્ગ છે. પરિણામોની. આ જાતિની સ્થિતિનું એક ઉદાહરણ છે.

ઈન્ટરપ્રોસેસ કમ્યુનિકેશન (IPC) ઈન્ટરપ્રોસેસ કમ્યુનિકેશન (IPC) એ પ્રોગ્રામિંગનો સમૂહ છે

ઈન્ટરફેસ કે જે પ્રોગ્રામરને વિવિધ પ્રોગ્રામ પ્રક્રિયાઓ વચ્ચે પ્રવૃત્તિઓનું સંકલન કરવાની મંજૂરી આપે છે જે ઓપરેટિંગ સિસ્ટમમાં એકસાથે ચાલી શકે છે. આ પ્રોગ્રામને એક જ સમયે ઘણી વપરાશકર્તા વિનંતીઓને હેન્ડલ કરવાની મંજૂરી આપે છે. આ

ઈન્ટરપ્રોસેસ કમ્યુનિકેશન માટે ઉપયોગમાં લેવાતા મેસેજિંગ પ્રોટોકોલ ઘણીવાર હુમલા માટે સંવેદનશીલ હોય છે.

રીમોટ પ્રોસીજર કોલ (RPC) એક ઈન્ટરપ્રોસેસ છે

કોમ્યુનિકેશન મિકેનિઝમ જે એક હોસ્ટ પર ચાલતા પ્રોગ્રામને રિમોટ હોસ્ટ પર કોડ ચલાવવાની મંજૂરી આપે છે.

અસુરક્ષિત ફાઈલ ઓપરેશન્સ સમય-તપાસ-સમય-ઉપયોગની સમસ્યાઓ ઉપરાંત, અન્ય ઘણી ફાઈલ ઓપરેશન્સ અસુરક્ષિત છે. પ્રોગ્રામરો ઘણીવાર બનાવે છે

વિશે ધારણાઓ માલિકી, સ્થાન અથવા ફાઈલની વિશેષતાઓ જે કદાચ સાચી ન હોય. ઉદાહરણ તરીકે, તમે ધારી શકો છો કે તમે હંમેશા તમારા પ્રોગ્રામ દ્વારા બનાવેલી ફાઈલ પર લખી શકો છો. જો કે, જો કોઈ હુમલાખોર તમે તેને બનાવ્યા પછી તે ફાઈલ પરની પરવાનગીઓ અથવા ફ્લેગ બદલી શકે છે, અને જો તમે લખવાની કામગીરી પછી પરિણામ કોડ તપાસવામાં નિષ્ફળ થશો, તો તમે એ હકીકત શોધી શકશો નહીં કે ફાઈલ સાથે ચેડા કરવામાં આવ્યા છે.

B મૂળ નબળાઈઓના પ્રકાર

મૂળ નબળાઈઓના ઉદાહરણો છે:

- સેન્ડબોક્સિંગ મિકેનિઝમમાં નબળાઈઓ જે અવિશ્વસનીય બાઈટકોડને પ્રતિબંધોને ટાળવા માટે પરવાનગી આપે છે સુરક્ષા મેનેજર દ્વારા લાદવામાં આવે છે

જાવા ક્લાસ લાઈબ્રેરીમાં નબળાઈઓ જેના પર એન એપ્લિકેશન તેની સુરક્ષા પર આધાર રાખે છે

સુરક્ષા નબળાઈઓને સમજવી

સોફ્ટવેર અથવા પ્રોટોકોલ ડિઝાઈનમાં ખામીઓ સોફ્ટવેર ડિઝાઈનમાં મૂળભૂત ભૂલો અને દેખરેખ

ડિઝાઈન નબળાઈઓનું કારણ છે. ડિઝાઈન ખામીઓ પરિણામ

સોફ્ટવેરમાં સુરક્ષિત ન હોવાને કારણે તે ઉચ્ચ સ્તરની નબળાઈનો કેસ બને છે.

કમ્પ્યુટર નેટવર્ક્સ પર આધાર રાખે છે પ્રોટોકોલ્સ કે જે રનટાઈમ પર વિનિમય કરવામાં આવતા સંદેશાઓ, તેમનું ફોર્મેટ અને માળખું સ્પષ્ટ કરે છે. પ્રોટોકોલ અલગ-અલગ પ્રોટોકોલ સ્ટેક્સ સાથે જોડાયેલા હોય છે, દા.ત., TCP/IP, અથવા અલગ-અલગ મોડલ, દા.ત., OSI, અને અન્ડરસ્પેસિફાઈડ સુરક્ષા સાથેના ઘણા પ્રોટોકોલ હજુ પણ વ્યવહારમાં હાજર છે. ખામીયુક્ત પ્રોટોકોલને કારણે ઊભી થતી કેટલીક નબળાઈઓ નીચે વર્ણવેલ છે:

સોફ્ટવેર ડેવલપમેન્ટમાં, ઉપયોગના સમયની તપાસનો સમય (TOC-TOU અથવા TOCTOU, “TOCK too” ઉચ્ચારવામાં આવે છે) એ શરતની ચકાસણી (જેમ કે સુરક્ષા ઓળખપત્ર) અને ઉપયોગ વચ્ચેની સિસ્ટમમાં ફેરફારને કારણે સોફ્ટવેર બગનો વર્ગ છે. પરિણામોની. આ જાતિની સ્થિતિનું એક ઉદાહરણ છે.

ઈન્ટરપ્રોસેસ કમ્યુનિકેશન (IPC) ઈન્ટરપ્રોસેસ કમ્યુનિકેશન (IPC) એ પ્રોગ્રામિંગનો સમૂહ છે

ઈન્ટરફેસ કે જે પ્રોગ્રામરને વિવિધ પ્રોગ્રામ પ્રક્રિયાઓ વચ્ચે પ્રવૃત્તિઓનું સંકલન કરવાની મંજૂરી આપે છે જે ઓપરેટિંગ સિસ્ટમમાં એકસાથે ચાલી શકે છે. આ પ્રોગ્રામને એક જ સમયે ઘણી વપરાશકર્તા વિનંતીઓને હેન્ડલ કરવાની મંજૂરી આપે છે. આ

ઈન્ટરપ્રોસેસ કમ્યુનિકેશન માટે ઉપયોગમાં લેવાતા મેસેજિંગ પ્રોટોકોલ ઘણીવાર હુમલા માટે સંવેદનશીલ હોય છે.

રીમોટ પ્રોસીજર કોલ (RPC) એક ઈન્ટરપ્રોસેસ છે

કોમ્યુનિકેશન મિકેનિઝમ જે એક હોસ્ટ પર ચાલતા પ્રોગ્રામને રિમોટ હોસ્ટ પર કોડ ચલાવવાની મંજૂરી આપે છે.

અસુરક્ષિત ફાઈલ ઓપરેશન્સ સમય-તપાસ-સમય-ઉપયોગની સમસ્યાઓ ઉપરાંત, અન્ય ઘણી ફાઈલ ઓપરેશન્સ અસુરક્ષિત છે. પ્રોગ્રામરો ઘણીવાર બનાવે છે

વિશે ધારણાઓ માલિકી, સ્થાન અથવા ફાઈલની વિશેષતાઓ જે કદાચ સાચી ન હોય. ઉદાહરણ તરીકે, તમે ધારી શકો છો કે તમે હંમેશા તમારા પ્રોગ્રામ દ્વારા બનાવેલી ફાઈલ પર લખી શકો છો. જો કે, જો કોઈ હુમલાખોર તમે તેને બનાવ્યા પછી તે ફાઈલ પરની પરવાનગીઓ અથવા ફ્લેગ બદલી શકે છે, અને જો તમે લખવાની કામગીરી પછી પરિણામ કોડ તપાસવામાં નિષ્ફળ થશો, તો તમે એ હકીકત શોધી શકશો નહીં કે ફાઈલ સાથે ચેડા કરવામાં આવ્યા છે.

B મૂળ નબળાઈઓના પ્રકાર

મૂળ નબળાઈઓના ઉદાહરણો છે:

સેન્ડબોક્સિંગ મિકેનિઝમમાં નબળાઈઓ જે

અવિશ્વસનીય બાઈટકોડને પ્રતિબંધોને ટાળવા માટે પરવાનગી આપે છે

સુરક્ષા મેનેજર દ્વારા લાદવામાં આવે છે

જાવા ક્લાસ લાઇબ્રેરીમાં નબળાઈઓ જેના પર એન

એપ્લિકેશન તેની સુરક્ષા પર આધાર રાખે છે

સુરક્ષા નબળાઈઓને સમજવી

સોફ્ટવેર અથવા પ્રોટોકોલ ડિઝાઇનમાં ખામીઓ સોફ્ટવેર ડિઝાઇનમાં મૂળભૂત ભૂલો અને દેખરેખ

ડિઝાઇન નબળાઈઓનું કારણ છે. ડિઝાઇન ખામીઓ પરિણામ

સોફ્ટવેરમાં સુરક્ષિત ન હોવાને કારણે તે ઉચ્ચ સ્તરની નબળાઈનો કેસ બને છે.

કમ્પ્યુટર નેટવર્ક્સ પર આધાર રાખે છે પ્રોટોકોલ્સ કે જે રનટાઈમ પર વિનિમય કરવામાં આવતા સંદેશાઓ, તેમનું ફોર્મેટ અને માળખું સ્પષ્ટ કરે છે. પ્રોટોકોલ અલગ-અલગ પ્રોટોકોલ સ્ટેક્સ સાથે જોડાયેલા હોય છે, દા.ત., TCP/IP, અથવા અલગ-અલગ મોડલ, દા.ત., OSI, અને અન્ડરસ્પેસિફાઇડ સુરક્ષા સાથેના ઘણા પ્રોટોકોલ હજુ પણ વ્યવહારમાં હાજર છે. ખામીયુક્ત પ્રોટોકોલને કારણે ઊભી થતી કેટલીક નબળાઈઓ નીચે વર્ણવેલ છે:

A. TCP/IP. TCP/IP પ્રોટોકોલ સ્ટેક્માં કેટલાક નબળા મુદ્દાઓ છે જે પરવાનગી આપે છે:

સ્પૂફિંગ : સ્પૂફિંગ એટેક એ છે જ્યારે કોઈ દૂષિત પક્ષ નેટવર્ક હોસ્ટ્સ સામે હુમલા કરવા, ડેટા ચોરી કરવા, માલવેર ફેલાવવા અથવા ઍક્સેસ નિયંત્રણોને બાયપાસ કરવા માટે નેટવર્ક પર અન્ય ઉપકરણ અથવા વપરાશકર્તાનો ઢોંગ કરે છે. દૂષિત પક્ષો આને પરિપૂર્ણ કરવા માટે ઉપયોગ કરી શકે તેવા વિવિધ પ્રકારના સ્પૂફિંગ હુમલાઓ છે. તે IP એડ્રેસ સ્પૂફિંગ એટેક, ARP (એડ્રેસ રિઝોલ્યુશન પ્રોટોકોલ) સ્પૂફિંગ એટેક, DNS સર્વર સ્પૂફિંગ એટેક વગેરે છે.

ટેલનેટ પ્રોટોકોલ : ટેલનેટનો ઉપયોગ Microsoft Windows 2000 અને Unix પર ચાલતી સિસ્ટમને સંચાલિત કરવા માટે થઈ શકે છે. જ્યારે Microsoft સિસ્ટમથી UNIX સિસ્ટમ સાથે જોડાવા માટે ટેલનેટ ક્લાયન્ટનો ઉપયોગ કરવામાં આવે છે અને તેનાથી વિપરિત, વપરાશકર્તાનામ અને પાસવર્ડ સ્પષ્ટ લખાણમાં પ્રસારિત થાય છે જેથી સુરક્ષા નબળાઈ સર્જાય છે.

ફાઈલ ટ્રાન્સફર પ્રોટોકોલ (FTP) : ફાઈલ ટ્રાન્સફર પ્રોટોકોલ વપરાશકર્તાઓને રિમોટ સિસ્ટમ સાથે જોડાવા અને ફાઈલોને આગળ અને પાછળ ટ્રાન્સફર કરવાની મંજૂરી આપે છે. દૂરસ્થ કમ્પ્યુટર સાથે જોડાણ સ્થાપિત કરવાના ભાગરૂપે, FTP વપરાશકર્તા નામ પર આધાર રાખે છે અને

પ્રમાણીકરણ માટે પાસવર્ડ સંયોજન. FTP નો ઉપયોગ ટેલનેટ પ્રોટોકોલના ઉપયોગ જેવી જ સુરક્ષા સમસ્યા ઊભી કરે છે કારણ કે FTP પર ટાઇપ કરેલા પાસવર્ડ્સ

નેટવર્ક પર સાદા ટેક્સ્ટમાં પ્રસારિત થાય છે, પેકેટ દીઠ એક અક્ષર. આ પેકેટોને અટકાવી શકાય છે.

પ્રોટોકોલ અને સોફ્ટવેર કેવી રીતે છે તેમાં નબળાઈઓ

અમલમાં મૂક્યો

જ્યારે પ્રોટોકોલ સારી રીતે ડિઝાઇન કરવામાં આવે ત્યારે પણ, તે જે રીતે અમલમાં આવે છે તેના કારણે તે સંવેદનશીલ બની શકે છે. ઉદાહરણ તરીકે, ઇલેક્ટ્રોનિક મેઈલ માટેનો પ્રોટોકોલ એવી રીતે લાગુ થઈ શકે છે કે જે ઘૂસણખોરોને પીડિતના મશીનના મેઈલ પોર્ટ સાથે જોડાવા અને મશીનને સેવા દ્વારા હેતુ ન હોય તેવું કાર્ય કરવા માટે મૂર્ખ બનાવે છે. આ પ્રકારની

નબળાઈ ઘૂસણખોરોને પીડિતની સિસ્ટમ પરના એકાઉન્ટની ઍક્સેસ વિના દૂરસ્થ સાઈટ્સથી પીડિતના મશીન પર હુમલો કરવા સક્ષમ બનાવે છે.

જે ખામીઓ હતી તેના કારણે સોફ્ટવેર સંવેદનશીલ હોઈ શકે છે

સોફ્ટવેર રીલીઝ થાય તે પહેલા ઓળખાઈ ન હતી. આ પ્રકારની નબળાઈ પેટા વર્ગોની વિશાળ શ્રેણી ધરાવે છે, જે ઘૂસણખોરો વારંવાર શોષણ કરે છે તેમના પોતાના હુમલાના સાધનોનો ઉપયોગ કરીને જેમ કે ફાઈલ ઍક્સેસમાં રેસની સ્થિતિ, ડેટા કન્ટેન્ટ અને સાઈઝની બિન-અસ્તિત્વ તપાસ વગેરે.

સિસ્ટમ અને નેટવર્ક રૂપરેખાંકનોમાં નબળાઈઓ

સિસ્ટમ એડમિનિસ્ટ્રેટર્સ અને વપરાશકર્તાઓ નેટવર્ક ગોઠવણીમાં ડિફોલ્ટ સેટિંગ્સ બદલવાની અવગણના કરી શકે છે,

અથવા તેઓ ફક્ત તેમની સિસ્ટમને એમાં ચલાવવા માટે સેટ કરી શકે છે

જે નેટવર્કને સંવેદનશીલ બનાવે છે.

અસિંક્રોનસ ટ્રાન્સફર મોડ (ATM). જેને “મેનહોલ” તરીકે ઓળખવામાં આવે છે તેનાથી સુરક્ષા સાથે ચેડા થઈ શકે છે

મેનીપ્યુલેશન”- અંડરગ્રાઉન્ડ પાર્કિંગ ગેરેજ અને એલિવેટર શાફ્ટમાં નેટવર્ક કેબલ અને કનેક્શનની સીધી ઍક્સેસ.

ફ્રેમ રિલે. એટીએમની સમસ્યા જેવી જ છે

ઉપકરણ વહીવટ. સ્વીચો અને રાઉટર સરળતાથી HTTP ઈન્ટરફેસ દ્વારા અથવા કમાન્ડ લાઈન ઈન્ટરફેસ દ્વારા સંચાલિત થાય છે. નબળા પાસવર્ડ્સ (ઉદાહરણ તરીકે, સાર્વજનિક પાસવર્ડ્સ) ના ઉપયોગ સાથે જોડીને, તે કોઈપણ તકનીકી જ્ઞાન ધરાવતા કોઈપણને ઉપકરણને નિયંત્રિત કરવાની મંજૂરી આપે છે.

મોડેમ. મોડેમ “ફાયરવોલ” ને બાયપાસ કરે છે જે નેટવર્કને બહારના ઘૂસણખોરોથી સુરક્ષિત કરે છે. હેકર મોડેમ ટેલિફોન નંબર ઓળખવા માટે “વોર ડાયલર” ટૂલનો ઉપયોગ કરે છે અને નબળા પાસવર્ડને તોડવા માટે “પાસવર્ડ ક્રેકર” ટૂલ સિસ્ટમમાં પ્રવેશ મેળવી શકે છે.

વેબ અથવા ક્લાઉડ એપ્લિકેશન્સમાં નબળાઈઓ

ત્યાં ઘણી નોંધપાત્ર નબળાઈઓ છે જે હોવી જોઈએ

જ્યારે કોઈ સંસ્થા તેમની મહત્વપૂર્ણ એપ્લિકેશનો અને ડેટાને ક્લાઉડ કમ્પ્યુટિંગમાં ખસેડવા માટે તૈયાર હોય ત્યારે ધ્યાનમાં લેવામાં આવે છે

પર્યાવરણ, આ નબળાઈઓ નીચે વર્ણવેલ છે:

સેશન રાઈડિંગ અને હાઈજેકિંગ સેશન રાઈડિંગ એ હેક્સનો સંદર્ભ આપે છે જે લક્ષિત વપરાશકર્તા વતી વેબ એપ્લિકેશનને આદેશો મોકલે છે અને ફક્ત તે વપરાશકર્તાને ઈમેઈલ મોકલીને અથવા વપરાશકર્તાને ખાસ રચિત વેબસાઈટની મુલાકાત લેવા માટે છેતરે છે. સેશન રાઈડિંગ યુઝર ડેટાને ડિલીટ કરે છે, બિડ અથવા ઓર્ડર જેવા ઓનલાઈન વ્યવહારો કરે છે, ઈન્ટરનેટ દ્વારા ઈન્ટ્રાનેટ સિસ્ટમમાં સ્પામ મોકલે છે અને સિસ્ટમ તેમજ નેટવર્ક કન્ફિગરેશનમાં ફેરફાર કરે છે અથવા તો ફાયરવોલ ખોલે છે.

B વર્ચ્યુઅલ મશીન એસ્કેપ VM એસ્કેપ એ નબળાઈ છે જે ગેસ્ટ-લેવલ VM ને તેના હોસ્ટ પર હુમલો કરવા સક્ષમ કરે છે. આ નબળાઈ હેઠળ હુમલાખોર VM પર કોડ ચલાવે છે જે તેની અંદર ચાલતા OSને તોડીને હાયપરવાઈઝર સાથે સીધો સંપર્ક કરવા દે છે.

C સેવાની વિશ્વસનીયતા અને ઉપલબ્ધતા ક્લાઉડ સ્ટોરેજ ઈન્ફ્રાસ્ટ્રક્ચર નોંધપાત્ર સમય માટે નીચે જઈ શકે છે, જેના કારણે ડેટા ખોવાઈ શકે છે અને

વેબ સેવાઓ સાથે એક્સેસ સમસ્યાઓ થઈ શકે છે.

ડી અસુરક્ષિત ક્ષિપ્રગ્રાફી

હુમલાખોરો કોઈપણ ક્ષિપ્રગ્રાફિક મિકેનિઝમ અથવા અલ્ગોરિધમને ડીકોડ કરી શકે છે કારણ કે તેમને હેક કરવાની મુખ્ય પદ્ધતિઓ શોધાઈ છે.

ઈ ડેટા પ્રોટેક્શન અને પોર્ટબિલિટી જોકે ક્લાઉડ સેવાઓ ક્લાઈન્ટ અને પ્રદાતા વચ્ચેના કારારના આધારે ઓફર કરવામાં આવે છે પરંતુ જ્યારે કોન્ટ્રાક્ટ સમાપ્ત થશે અને ક્લાઈન્ટ હવે ચાલુ રાખવા માંગતો નથી ત્યારે શું થશે.

F વેન્ડર લોક-ઇન આ નબળાઈ અપરિપક્વ પ્રદાતાઓ અને નવા બિઝનેસ મોડલ્સને કારણે થાય છે જે નિષ્ફળતા અને બિઝનેસમાંથી બહાર જવાનું જોખમ વધારે છે.

જી ઈન્ટરનેટ ડિપેન્ડન્સી ક્લાઉડ કમ્પ્યુટિંગ એ ઈન્ટરનેટ આધારિત ટેકનોલોજી છે જ્યાં વપરાશકર્તાઓ વેબ બ્રાઉઝર દ્વારા સેવાઓનો ઉપયોગ કરી રહ્યાં છે. જો ઈન્ટરનેટ ઉપલબ્ધ ન હોય અથવા સેવા બંધ હોય તો શું થશે, વપરાશકર્તાઓની સિસ્ટમો અને ઓપરેશન્સનું શું થશે જે ખૂબ જ જટિલ છે અને જેને 24 કલાક ચલાવવાની જરૂર છે જેમ કે હેલ્થકેર અને બેંકિંગ સિસ્ટમ્સ

ઓનલાઈન ઈ ટ્રાન્ઝેક્શન્સમાં નબળાઈઓ ઓનલાઈન લેવડદેવડમાં જબરદસ્ત વધારો ઓનલાઈન પેમેન્ટ સિસ્ટમ્સની સુરક્ષા સામેના હુમલાની સંખ્યામાં અને પ્રકારોમાં સમાન વધારો સાથે છે. આમાંના કેટલાક હુમલાઓએ નબળાઈઓનો ઉપયોગ કર્યો છે જે વેબસાઈટ્સ દ્વારા ઉપયોગમાં લેવાતા ફરીથી વાપરી શકાય તેવા તૃતીય પક્ષ ઘટકોમાં પ્રકાશિત કરવામાં આવી છે, જેમ કે શોપિંગ કાર્ટ સોફ્ટવેર. અન્ય હુમલાઓએ નબળાઈઓનો ઉપયોગ કર્યો છે જે કોઈપણ વેબ એપ્લિકેશનમાં સામાન્ય છે, જેમ કે SQL ઈન્જેક્શન અથવા કોસ-સાઈટ સ્ક્રિપ્ટિંગ. ઓનલાઈન ઈ ટ્રાન્ઝેક્શન્સમાં નબળાઈઓના સામાન્ય પ્રકારો SQL ઈન્જેક્શન, કોસ-સાઈટ સ્ક્રિપ્ટિંગ, માહિતી જાહેર, પાથ ડિસ્ક્લોઝર, કિંમતમાં મેનીપ્યુલેશન અને બફર ઓવરફ્લો છે.

આ નબળાઈઓનું સફળ શોષણ પરિણામોની વિશાળ શ્રેણી તરફ દોરી શકે છે. માહિતી અને પાથની જાહેરાત

નબળાઈઓ સામાન્ય રીતે પ્રારંભિક તબક્કા તરીકે કાર્ય કરશે જે વધુ શોષણ તરફ દોરી જશે. એસક્યુએલ ઈન્જેક્શન અથવા કિંમતની હેરફેર

હુમલાઓ વેબસાઈટને અપંગ કરી શકે છે, સમાધાન કરી શકે છે

ગોપનીયતા, અને સૌથી ખરાબ કેસોને કારણે ઈ-કોમર્સ બિઝનેસ સંપૂર્ણપણે બંધ થઈ જાય છે. આવી નબળાઈઓ માટેનું એક મુખ્ય કારણ એ હકીકત છે કે વેબ એપ્લિકેશન ડેવલપર્સ ઘણીવાર સુરક્ષિત પ્રોગ્રામિંગ તકનીકોથી સારી રીતે વાકેફ હોતા નથી. બ્રાઉઝર સુરક્ષા અને કૂકીઝ અને પોપ અપની ભૂમિકા સુરક્ષા નબળાઈઓ હેકર દ્વારા કૂકીના ડેટાને વાંચવાની મંજૂરી આપી શકે છે, વપરાશકર્તાના ડેટાની એક્સેસ મેળવવા માટે ઉપયોગમાં લેવાય છે અથવા (વપરાશકર્તાના ઓળખપત્રો સાથે) એક્સેસ મેળવવા માટે ઉપયોગમાં લેવાય છે.

વેબસાઈટ કે જેની કૂકી છે. પોપ-અપ જાહેરાતો અથવા પોપ-અપ્સ

પોપઅપ વિન્ડોમાંથી “સુરક્ષા” જોખમો ફિશિંગ છે, અનિચ્છનીય વેબ સાઈટ્સ પર ફસાયા વગેરે. બ્રાઉઝર, વેબ એપ્લિકેશન, ઓએસ અને સ્માર્ટફોનમાં સુરક્ષા છિદ્રો: સુરક્ષા પરિભાષામાં, છિદ્ર એ સોફ્ટવેર અથવા ઓપરેટિંગ સિસ્ટમની નબળાઈનો સંદર્ભ આપે છે જેનો ઉપયોગ કરી શકાય છે. કોમ્પ્યુટર સિસ્ટમ અથવા નેટવર્ક કે જેના પર હોલ રહે છે તેની

એકંદર સુરક્ષા સાથે સમાધાન કરે છે. ત્રણ વિવિધ પ્રકારની નબળાઈઓ છે:

ઓપરેટિંગ સિસ્ટમની નબળાઈઓ તે છે જે પર અસર કરે છે

લિનક્સ કર્નલ; અથવા ઘટકો જે Microsoft, Apple અથવા માલિકી દ્વારા ઉત્પાદિત ઓપરેટિંગ સિસ્ટમ સાથે મોકલે છે

યુનિક્સ વિકેતા, અને ઓપરેટિંગના ભાગ તરીકે વ્યાખ્યાયિત

વિકેતા દ્વારા સિસ્ટમ.

બ્રાઉઝરની નબળાઈઓ તે ઘટકોને અસર કરે છે

pવેબ બ્રાઉઝરના ભાગ તરીકે વ્યાખ્યાયિત. આમાં વેબનો સમાવેશ થાય છે બ્રાઉઝર્સ કે જે ઓપરેટિંગ સિસ્ટમ્સ સાથે મોકલે છે, જેમ કે

વિન્ડોઝ ઈન્ટરનેટ એક્સપ્લોરર અને એપલની સફારી, તૃતીય-પક્ષ બ્રાઉઝર્સ સાથે, જેમ કે મોઝિલા ફાયરફોક્સ અને ગૂગલ ક્રોમ.

એપ્લિકેશન નબળાઈઓ તે છે જે ઓપરેટિંગ સિસ્ટમ વિકેતાઓ અને અન્ય વિકેતાઓ દ્વારા પ્રકાશિત ઘટકો સહિત અન્ય તમામ ઘટકોને અસર કરે છે.

ઓપન સોર્સ ઘટકોમાં નબળાઈઓ કે જે લિનક્સ ડિસ્ટ્રિબ્યુશન (જેમ કે X વિન્ડો સિસ્ટમ, જીનોમ ડેસ્કટોપ એન્વાયર્નમેન્ટ, જીઆઈએમપી અને અન્ય) સાથે મોકલી શકે છે તેને એપ્લિકેશન નબળાઈઓ ગણવામાં આવે છે.

વેબ એપ્લિકેશન્સમાં સુરક્ષા છિદ્રો નીચે OWASP (ઓપન વેબ એપ્લિકેશન સિક્યુરિટી પ્રોજેક્ટ) શ્રેણીમાં ટોચના 10 જોખમોની સૂચિ છે. ઈન્જેક્શન (SQLi -> SQL ઈન્જેક્શન), ટૂટેલા પ્રમાણીકરણ અને સત્ર વ્યવસ્થાપન, XSS (કોસ સાઈટ સ્ક્રિપ્ટિંગ), અસુરક્ષિત

ડાયરેક્ટ ઓબ્જેક્ટ રેફરન્સ, સુરક્ષા ખોટી ગોઠવણી,

સંવેદનશીલ ડેટા એક્સપોઝર, ખૂટે ફંક્શન લેવલ એક્સેસ કંટ્રોલ, કોસ સાઈટ રિકવેસ્ટ ફોર્જરી (CSRF અથવા XSRF), જાણીતી નબળાઈઓ અને અમાન્ય રીડાયરેક્ટ અને ફોર્વર્ડ સાથે ઘટકોનો ઉપયોગ. OS માં સુરક્ષા છિદ્રો UNIX OS માં કેટલીક નબળાઈઓ છે Setuid સમસ્યાઓ, ટ્રોજન હોર્સિસ અને ટર્મિનલ મુશ્કેલીઓ Windows OS માં કેટલીક નબળાઈઓ પાસવર્ડ્સ, પીઅર ટુ પીઅર ફાઇલ શેરિંગ, Microsoft Outlook અને Outlook Express માં એમ્બેડેડ ઓટોમેશન સુવિધાઓમાં નબળાઈઓ છે જે મંજૂરી આપી શકે છે. બદમાશ કોડનો અમલ. LINUX OS માં કેટલીક અસમર્થતાઓમાં પરવાનગી તપાસો, અપ્રારંભિક ડેટા અને મેમરી ખૂટે છે

ગેરવહીવટ

સ્માર્ટફોનમાં કેટલીક નબળાઈઓ ડેટા લીકેજ છે ઉપકરણની ખોટ અથવા ચોરીના પરિણામે, ડેટાની અજાણતા જાહેરાત, ડિકમિશન પરના હુમલા

સ્માર્ટફોન્સ, ફિશિંગ હુમલાઓ, સ્પાયવેર હુમલાઓ, નેટવર્ક સ્પૂફિંગ હુમલાઓ, સર્વેલન્સ (વપરાશકર્તા દેખરેખ હેઠળ)

હુમલાઓ, ડાયલરવેર હુમલાઓ (પૈસા ચોરી), નાણાકીય

મોલવેર હુમલાઓ (શ્રેણીપત્રોની ચોરી) અને નેટવર્ક ભીડ.

નબળાઈ આકારણી સાધનો અને તકનીકો

નબળાઈનું મૂલ્યાંકન એ એક સુરક્ષા કવાયત છે જે નબળાઈઓને ઓળખે છે, ઓળખે છે અને તેની ગણતરી કરે છે અને નેટવર્ક્, એપ્લિકેશન્સ અને સિસ્ટમ્સમાં સુરક્ષા જવાબદારીઓ વિશેની શોધ પર અહેવાલ આપે છે. નબળાઈ આકારણી આ દ્વારા નબળાઈઓ શોધે છે:

સુરક્ષા તકનીકો

પેચ મેનેજમેન્ટ સિસ્ટમ્સ (WSUS, SCCM, LanDesk, VMWare અપડેટ મેનેજર) નબળાઈ આકારણીમાં મુખ્યત્વે નીચેના ત્રણ પગલાં શામેલ છે:

માહિતી એકત્રીકરણ અને શોધ જેમાં નેટવર્ક સ્કેનિંગ, પોર્ટ્સ સ્કેનિંગ, ડિરેક્ટરી સેવાઓ અને DNS ઝોનનો સમાવેશ થાય છે અને રજીસ્ટર.

ગણતરી જેમાં યજમાનો અને OS, પોર્ટ્સ, સેવાઓ અને તેમના સંસ્કરણો, માહિતી અને SNMP સમુદાયોનો સમાવેશ થાય છે

તપાસ જેમાં નબળાઈઓની ઓળખ, નબળાઈઓની ઓળખ, રિપોર્ટ જનરેશન અને ઉપચાર સાધનોનો ઉપયોગ સામેલ છે

નબળાઈ મૂલ્યાંકન સાધનો નબળાઈ આકારણી સાધનો વિવિધ સ્તરો પર જોવા મળતી નબળાઈઓની અસરને શોધી કાઢે છે, ઓળખે છે, માપે છે. મોટાભાગના નબળાઈ આકારણી સાધનો નેટવર્કિંગ અને નેટવર્ક ઉપકરણો (સ્વીચો, ફાયરવોલ, પ્રિન્ટર, વગેરે) તેમજ સર્વર, ડેસ્કટોપ અને પોર્ટેબલ કોમ્પ્યુટર સહિત સંખ્યાબંધ નેટવર્ક નોડ્સને સ્કેન કરવામાં સક્ષમ છે. સામાન્ય નબળાઈ આકારણી સાધનો નેટવર્ક સ્કેનર્સ, હોસ્ટ સ્કેનર્સ, ડેટાબેઝ સ્કેનર્સ, વેબ એપ્લિકેશન સ્કેનર્સ, મલ્ટિલેવલ સ્કેનર્સ, ઓટોમેટેડ પેનિટ્રેશન ટેસ્ટ ટૂલ્સ અને નબળાઈ સ્કેન કન્સોલિડેટર્સ છે. નબળાઈઓનું શોષણ કરવાની તકનીકો ભૂતપૂર્વ માટે નબળાઈઓનો ઉપયોગ કરી શકાય છે. પેકેટ સ્નિફર્સનો ઉપયોગ કરીને. અન્ય સાધનોનો ઉપયોગ બનાવટી સરનામાં સાથે પેકેટો બનાવવા માટે થાય છે; આ ટૂલ્સનો એક ઉપયોગ એ છે કે અસ્વીકાર-ઓફ-સર્વિસ હુમલાને એવી રીતે માઉન્ટ કરવો કે જે હુમલાના સ્ત્રોતને છુપાવે. ઘૂસણખોરો કોમ્પ્યુટર એડ્રેસને “સ્પૂફ” પણ કરે છે, તેમની વાસ્તવિક ઓળખને ઢાંકી દે છે અને સફળતાપૂર્વક જોડાણો બનાવે છે જે અન્યથા ન થાય. પરવાનગી છે. આ રીતે, તેઓ કમ્પ્યુટર્સ વચ્ચેના વિશ્વાસ સંબંધોનું શોષણ કરે છે. ટ્રોજન, વાઈરસ, વોર્મ્સ, લોજિક બોમ્બ, ફિશીંગ, ફોરવોર્ડિંગ અને શહેરી દંતકથાઓ શેર કરવા, નાઈજીરીયન કૌભાંડોને પ્રતિસાદ આપવા વગેરેના ઉપયોગ દ્વારા સૌથી સામાન્ય શોષણ થાય છે. નબળાઈઓને ઠીક કરવાની તકનીકો અસરકારક ઉપાયો સતત પ્રક્રિયાઓની માંગ કરે છે જેને એકસાથે નબળાઈ વ્યવસ્થાપન કહેવામાં આવે છે. નબળાઈ વ્યવસ્થાપન દ્વારા વ્યાખ્યાયિત પ્રક્રિયાઓ અને સંબંધિત તકનીક સંસ્થાઓને નેટવર્ક સુરક્ષા નબળાઈઓને અસરકારક રીતે શોધવા અને ઠીક કરવામાં મદદ કરે છે. આ પ્રક્રિયાઓનો વ્યવસ્થિત ઉપયોગ વ્યાપારી પ્રણાલીઓને વધુ વારંવાર થતા વાઈરસ, વોર્મ્સ અને અન્ય નેટવર્ક જન્ય હુમલાઓથી સુરક્ષિત કરે છે. નબળાઈ વ્યવસ્થાપનની સતત પ્રક્રિયાઓમાં સુરક્ષા નીતિઓ અને નિયંત્રણો, ટ્રેકિંગ ઈન્વેન્ટરી / અસ્કયામતોનું વર્ગીકરણ, નબળાઈઓ માટે સ્કેનિંગ સિસ્ટમ્સ, ઈન્વેન્ટરી સામે નબળાઈઓની તુલના, જોખમોનું વર્ગીકરણ, પેચોનું પૂર્વ-પરીક્ષણ, પેચો લાગુ કરવું અને ફરીથી સ્કેનિંગ અને પુષ્ટિકરણનો સમાવેશ થાય છે. તમે કરી શકો છો તેમાંના મોટા ભાગનાને હવે સુરક્ષા એપ્લિકેશનો અને વેબ આધારિત સેવાઓ સાથે સ્વચાલિત કરો. સુરક્ષાની નબળાઈઓને ઘટાડવા માટે શ્રેષ્ઠ પ્રયાસો અને માર્ગદર્શિકા

- 1 ઉપયોગ કરતા પહેલા તમામ ચલોને પ્રારંભ કરો
- 2 ઉપયોગ કરતા પહેલા બધા વપરાશકર્તા ઈનપુટને માન્ય કરો
- 3 સર્વર પર વહીવટી પરવાનગીઓને પ્રતિબંધિત કરો અને ડેટાબેઝ
- 4 ભૂલો સંભાળો અને સિસ્ટમ ભૂલ સંદેશાઓ પ્રદર્શિત કરશો નહીં અંતિમ વપરાશકર્તાઓ માટે

- 5 ઓછામાં ઓછી પરવાનગીઓ સાથે એકાઉન્ટ્સ પ્રદાન કરો અને વિશેષાધિકારો જરૂરી છે
- 6 તમારા કોડમાં રહસ્યો (દા.ત. પાસવર્ડ્સ, કી) સ્ટોર કરશો નહીં
- 7 સામાન્ય કાર્યો (દા.ત. પ્રમાણીકરણ, એન્ક્રિપ્શન, સત્ર ટ્રેકિંગ) માટે પરીક્ષણ કરેલ, વિશ્વસનીય પુસ્તકાલયો અથવા મોક્યુલોનો ઉપયોગ કરો.
- 8 સુરક્ષિત લોગિન પૃષ્ઠો અને પૃષ્ઠો HTTPS સાથે પ્રમાણીકરણ દ્વારા સુરક્ષિત
- 9 ખાતરી કરો કે સર્વર ઘટકો (OS, સોફ્ટવેર/એપ્સ) અદ્યતન છે
- 10 પ્રોડક્શન સર્વર્સ પર બિનજરૂરી એપ્લિકેશન ઈન્સ્ટોલ કરવાનું ટાળો
- 11 વેબ સર્વરમાંથી બિનઉપયોગી અને બેકઅપ પૃષ્ઠો દૂર કરો
- 12 જો શક્ય હોય તો, કોડ લાઈબ્રેરીઓ અને રૂપરેખાંકન ફાઈલો બનાવો વેબ પરથી અગમ્ય
- 13 ડિરેક્ટરી બ્રાઉઝિંગને અક્ષમ કરો
- 14 વપરાશકર્તા પર આધારિત ઓપરેટિંગ સિસ્ટમ કૉલ કરવાનું ટાળો ઈનપુટ
- 15 તમારા ડેવલપમેન્ટ ફ્રેમવર્કમાં બનેલ સત્ર ટ્રેકિંગ મિકેનિઝમનો ઉપયોગ કરો. જોખમ એ ઘટનાની સંભાવનાનું સંયોજન છે અને

તેના પરિણામો. તે હોવાની સંભાવનાનો ઉલ્લેખ કરે છે

આપેલ હુમલા દ્વારા લક્ષ્યાંકિત.

ખતરો, નબળાઈ અને જોખમ વચ્ચેનો સંબંધ ધમકી, નબળાઈ અને જોખમ વચ્ચેના સંબંધને વ્યાખ્યાયિત કરતા પહેલા ચાલો આપણે નીચેની શરતોની સમીક્ષા કરીએ: સંપત્તિ: માહિતી સુરક્ષામાં, સંપત્તિ એ છે જેનો આપણે પ્રયાસ કરી રહ્યા છીએ. રક્ષણ તે લોકો, મિલકત અથવા માહિતી હોઈ શકે છે.

ધમકી: કોઈ પણ વસ્તુ જે કોઈ નબળાઈનો ઉપયોગ કરી શકે છે, ઈરાદાપૂર્વક અથવા આકસ્મિક રીતે અને સંપત્તિ પ્રાપ્ત કરી શકે છે, નુકસાન પહોંચાડી શકે છે અથવા તેનો નાશ કરી શકે છે.

નબળાઈ: તે અમારા સંરક્ષણ પ્રયાસોમાં નબળાઈ અથવા ગાબડાને દર્શાવે છે.

જોખમ: જ્યારે કોઈ ધમકી નબળાઈનું શોષણ કરે છે, ત્યારે તે સંપત્તિનું નુકસાન, નુકસાન અથવા વિનાશનું કારણ બની શકે છે. આને જોખમ કહેવાય છે.

તેથી જોખમ એ અસ્કયામતો, ધમકીઓ અને નબળાઈઓનું આંતરછેદ છે. એટલે કે. અસ્કયામતો x ધમકીઓ x નબળાઈઓ = સંપત્તિનું જોખમ મૂલ્ય માહિતી પર મૂકવામાં આવેલ મૂલ્ય સામેલ પક્ષોને સંબંધિત છે, તેને વિકસાવવા માટે કયા કાર્યની જરૂર હતી, તેની જાળવણી માટે કેટલો ખર્ચ થાય છે, જો તે ખોવાઈ જાય અથવા નાશ પામે તો શું નુકસાન થશે, શું દુશ્મનો તેના માટે ચૂકવણી કરશે, અને કઈ જવાબદારી દંડ સહન કરી શકાય છે. જો કોઈ કંપની માહિતી અને અન્ય સંપત્તિઓનું મૂલ્ય જાણતી નથી જે તે સુરક્ષિત કરવાનો પ્રયાસ કરી રહી છે, તો તે જાણતી નથી કે તેણે તેમને બચાવવા માટે કેટલા પૈસા અને સમય ખર્ચવા જોઈએ. સંપત્તિઓને મૂલ્યો સોંપતી વખતે, એક અમુક મુદ્દાઓને ધ્યાનમાં લેવાની જરૂર છે જે નીચે મુજબ છે.

સંપત્તિ હસ્તગત કરવા અથવા વિકસાવવા માટેનો ખર્ચ

- સંપત્તિની જાળવણી અને રક્ષણ માટેનો ખર્ચ
- માલિકો અને વપરાશકર્તાઓ માટે સંપત્તિનું મૂલ્ય
- પ્રતિસ્પર્ધીઓ માટે સંપત્તિનું મૂલ્ય
- બૌદ્ધિક સંપદાનું મૂલ્ય જે માહિતીને વિકસાવવામાં આવ્યું હતું
- કિંમત જે અન્ય લોકો સંપત્તિ માટે ચૂકવવા તૈયાર છે
- જો સંપત્તિ ખોવાઈ ગઈ હોય અથવા નુકસાન થઈ હોય તો તેને બદલવાની કિંમત
- કાર્યકારી અને ઉત્પાદન પ્રવૃત્તિઓ કે જે જો સંપત્તિ અનુપલબ્ધ હોય તો પ્રભાવિત થાય છે
- જો સંપત્તિ સાથે ચેડા કરવામાં આવે તો જવાબદારીની સમસ્યાઓ
- સંસ્થામાં સંપત્તિની ઉપયોગિતા અને ભૂમિકા

સંપત્તિના મૂલ્યને સમજવું એ પ્રથમ પગલું છે

કઈ સુરક્ષા પદ્ધતિઓનો ઉપયોગ કરવો જોઈએ તે સમજવું

અને તેના રક્ષણ માટે કયા ભંડોળ જવું જોઈએ.

ધમકી સ્ત્રોત/એજન્ટ શું છે?

ધમકીનો સ્ત્રોત અથવા ધમકી એજન્ટ એ એક એવી એન્ટિટી છે જેની પાસે અસર લાવવાનો હેતુ અને ક્ષમતા છે. શ્રેટ એજન્ટો નીચેનામાંથી એક અથવા વધુ પગલાં લઈ શકે છે

સંપત્તિ સામે:

- એક્સેસ - સરળ અનધિકૃત એક્સેસ
- દુરુપયોગ - સંપત્તિનો અનધિકૃત ઉપયોગ (દા.ત., ઓળખની ચોરી,

સમાધાન પર પોર્ન વિતરણ સેવાની સ્થાપના

સર્વર, વગેરે)

- જાહેર કરો - ધમકી એજન્ટ ગેરકાયદેસર રીતે જાહેર કરે છે સંવેદનશીલ માહિતી
- સંશોધિત કરો - સંપત્તિમાં અનધિકૃત ફેરફારો
- એક્સેસ નકારવું - વિનાશ, બિન-ડેટા એસેટની ચોરી વગેરેનો સમાવેશ થાય છે. ધમકી એજન્ટો નીચેનામાંથી કોઈપણ હોઈ શકે છે: આ વ્યક્તિઓ અને જૂથોને નીચે પ્રમાણે વર્ગીકૃત કરી શકાય છે:
- નોન-ટાર્ગેટ સ્પેસિફિક: નોન-ટાર્ગેટ સ્પેસિફિક શ્રેટ એજન્ટ્સ કમ્પ્યુટર વાયરસ, વોર્મ્સ, ટ્રોજન અને લોજિક બોમ્બ છે.
- કર્મચારીઓ: સ્ટાફ, કોન્ટ્રાક્ટરો, ઓપરેશનલ/મેન્ટેનન્સ કર્મચારીઓ અથવા સુરક્ષા રક્ષકો કે જેઓ કંપનીથી નારાજ છે.
- સંગઠિત અપરાધ અને ગુનેગારો: ગુનેગારો તેમના માટે મૂલ્યવાન માહિતીને લક્ષ્ય બનાવે છે, જેમ કે બેંક ખાતા, ક્રેડિટ કાર્ડ અથવા બૌદ્ધિક સંપદા કે જેને નાણાંમાં રૂપાંતરિત કરી શકાય છે. ગુનેગારો ઘણીવાર તેમની મદદ કરવા માટે અંદરના લોકોનો ઉપયોગ કરશે.
- કોર્પોરેશનો: કોર્પોરેશનો અપમાનજનક માહિતી યુદ્ધ અથવા સ્પર્ધાત્મક બુદ્ધિમાં રોકાયેલા છે. ભાગીદારો અને સ્પર્ધકો આ શ્રેણી હેઠળ આવે છે.
- માનવ, અજાણતા: અકસ્માતો, બેદરકારી.
- માનવ, ઈરાદાપૂર્વક: અંદરનો, બહારનો વ્યક્તિ.

- કુદરતી: પૂર, આગ, વીજળી, ઉલ્કા, ધરતીકંપ

જોખમ નિયંત્રણો જો જોખમનું શમન એ માહિતી સુરક્ષાનું કેન્દ્રિય કેન્દ્ર છે, તો નિયંત્રણો આ લક્ષ્યને હાંસલ કરવા માટેના પ્રાથમિક સાધનો છે. નિયંત્રણ એ કોઈપણ ઉપકરણ અથવા પ્રક્રિયા છે જેનો ઉપયોગ જોખમ ઘટાડવા માટે થાય છે

મૂળભૂત રીતે ત્રણ પ્રકારના નિયંત્રણો છે:

- 1 વહીવટી : વહીવટી નિયંત્રણો એ લોકો કરે છે તે ક્રિયાઓ છે. વહીવટી નિયંત્રણો એ નીતિ અને કાર્યવાહીના વિકાસ અને પાલનની ખાતરી કરવાની પ્રક્રિયા છે
- 2 તકનીકી અથવા તાર્કિક : આ વર્ચ્યુઅલ, એપ્લિકેશન અને તકનીકી નિયંત્રણો (સિસ્ટમ અને સૉફ્ટવેર) છે, જેમ કે ફાયરવોલ, એન્ટી વાયરસ સૉફ્ટવેર, એન્ક્રિપ્શન અને મેકર/ચેકર એપ્લિકેશન રૂટિન. તકનીકી નિયંત્રણો કમ્પ્યુટર સિસ્ટમ્સ દ્વારા હાથ ધરવામાં આવે છે અથવા સંચાલિત થાય છે.
- 3 પ્રવૃત્તિ તબક્કાના નિયંત્રણો તકનીકી અથવા વહીવટી હોઈ શકે છે અને જોખમ ઘટાડવાના સ્તરના આધારે નીચે પ્રમાણે વર્ગીકૃત કરવામાં આવે છે:
 - નબળાઈના સંપર્કમાં આવતા જોખમને રોકવા માટે નિવારક નિયંત્રણો અસ્તિત્વમાં છે. આ એવા નિયંત્રણો છે જે નુકસાન અથવા નુકસાન થવાથી અટકાવે છે. ઉદાહરણ તરીકે, એક નિયંત્રણ કે જે જવાબદારીઓના વિભાજનને લાગુ કરે છે (એક વ્યક્તિ ચૂકવણીની વિનંતી સબમિટ કરી શકે છે, પરંતુ બીજી વ્યક્તિએ તેને અધિકૃત કરવી જોઈએ), કર્મચારી છેતરપિંડી કરી શકે તેવી શક્યતાને ઘટાડે છે. ચૂકવણી
 - અમારી સિસ્ટમમાં ખતરો આવી ગયો છે તે ઓળખવા માટે ડિટેક્ટીવ નિયંત્રણો અસ્તિત્વમાં છે. આ નિયંત્રણો એવા દાખલાઓને ઓળખવા માટે પ્રવૃત્તિનું નિરીક્ષણ કરે છે કે જ્યાં પ્રથમ અથવા પ્રક્રિયાઓ અનુસરવામાં આવી ન હતી. ઉદાહરણ તરીકે, છેતરપિંડીયુક્ત ચૂકવણીઓને ઓળખવા માટે વ્યવસાય સામાન્ય ખાતાવહી અથવા સમીક્ષા ચૂકવણી વિનંતી ઓડિટ લોગ સાથે સમાધાન કરી શકે છે.
 - દેખાતી ધમકીની અસરોને ઘટાડવા અથવા ઘટાડવા માટે સુધારાત્મક નિયંત્રણો અસ્તિત્વમાં છે. સુધારાત્મક નિયંત્રણો સિસ્ટમને પુનઃસ્થાપિત કરે છે અથવા હાનિકારક ઘટના પહેલા પ્રક્રિયાને રાજ્યમાં પાછી લાવે છે. ઉદાહરણ તરીકે, કોઈ વ્યક્તિએ પેમેન્ટ ડેટાને અયોગ્ય રીતે બદલ્યો હોવાના પુરાવા મળ્યા પછી કોઈ વ્યવસાય બેકઅપ ટેપમાંથી સિસ્ટમની સંપૂર્ણ પુનઃસ્થાપનનો અમલ કરી શકે છે.
 - વળતર આપતા નિયંત્રણો એ વૈકલ્પિક નિયંત્રણો છે જે મૂળ નિયંત્રણોના ઉદ્દેશ્યને શક્ય તેટલી નજીકથી પરિપૂર્ણ કરવા માટે રચાયેલ છે, જ્યારે પર્યાવરણની મર્યાદાઓને કારણે મૂળ રીતે ડિઝાઇન કરાયેલ નિયંત્રણોનો ઉપયોગ કરી શકાતો નથી. જોખમની સંભાવના જોખમની સંભાવના એ આ ચોક્કસ નબળાઈની કેટલી સંભાવના છે તેનું એક રફ માપ છે હુમલાખોર દ્વારા ખુલ્લું અને શોષણ કરવામાં આવશે. આ અંદાજમાં વધુ પડતું ચોક્કસ હોવું જરૂરી નથી. સામાન્ય રીતે, સંભાવના ઓછી, મધ્યમ અથવા ઊંચી છે કે કેમ તે ઓળખવું પૂરતું છે. ત્યાં સંખ્યાબંધ પરિબલો છે જે સંભાવના નક્કી કરવામાં મદદ કરી શકે છે. પરિબલોનો પ્રથમ સમૂહ સંકળાયેલા ધમકી એજન્ટ સાથે સંબંધિત છે. ઘ્યેય સંભવિત હુમલાખોરોના જૂથમાંથી સફળ હુમલાની સંભાવનાનો અંદાજ કાઢવાનો છે. નોંધ કરો કે ત્યાં બહુવિધ ધમકી એજન્ટો હોઈ શકે છે જે ચોક્કસ નબળાઈનો ઉપયોગ કરી શકે છે, તેથી સામાન્ય રીતે

સૌથી ખરાબ પરિસ્થિતિનો ઉપયોગ કરવો શ્રેષ્ઠ છે. ઉદાહરણ તરીકે, એક આંતરિક ઘણું હોઈ શકે છે

અનામી બહારના વ્યક્તિ કરતાં વધુ સંભવિત હુમલાખોર, પરંતુ તે સંખ્યાબંધ પરિબલો પર આધાર રાખે છે.

પરિબળોનો પ્રથમ સમૂહ સંકળાયેલા ધમકી એજન્ટ સાથે સંબંધિત છે. અહીં ધ્યેય ધમકી એજન્ટોના આ જૂથ દ્વારા સફળ હુમલાની સંભાવનાનો અંદાજ કાઢવાનો છે. ધ્યાનમાં લેવાનું બીજું પરિબલ એ હુમલા પાછળનો હેતુ છે. એક્સેસ અથવા સંસાધનોની આવશ્યકતા અને જોખમી એજન્ટોના જૂથનું કદ ધ્યાનમાં લેવાના અન્ય પરિબલો છે. પરિબલોનો આગળનો સમૂહ સામેલ નબળાઈ સાથે સંબંધિત છે. અહીંનો ધ્યેય શોધવામાં અને શોષણમાં સામેલ ચોક્કસ નબળાઈની સંભાવનાનો અંદાજ કાઢવાનો છે. આ શોધની સરળતા, શોષણની સરળતા, જોખમી એજન્ટોના આ જૂથ પ્રત્યેની જાગરૂકતા, આ શોષણની શોધની સંભાવનાને ધ્યાનમાં લે છે. પ્રભાવનો અંદાજ લગાવવા માટેના પરિબલો જ્યારે સફળ હુમલાની અસરને ધ્યાનમાં લેતા, ત્યારે તે સમજવું મહત્વપૂર્ણ છે કે ત્યાં બે પ્રકારની અસરો છે. પ્રથમ એપ્લિકેશન પરની “તકનીકી અસર” છે, તે જે ડેટાનો ઉપયોગ કરે છે અને તે જે કાર્યો પ્રદાન કરે છે. બીજું છે ધંધા પર “વ્યવસાયની અસર”. અને એપ્લિકેશનનું સંચાલન કરતી કંપની. છેવટે, વ્યવસાયની અસર વધુ મહત્વપૂર્ણ છે. જો કે, સફળ શોષણના વ્યવસાયિક પરિણામોને શોધવા માટે જરૂરી બધી માહિતીની તમારી પાસે એક્સેસ હોઈ શકતી નથી. આ કિસ્સામાં, તકનીકી જોખમ વિશે વધુ વિગતો પ્રદાન કરવાથી વ્યવસાયના જોખમ વિશે નિર્ણય લેવા માટે યોગ્ય વ્યવસાય પ્રતિનિધિને સક્ષમ બનાવશે. તકનીકી અસરને ચિંતાના પરંપરાગત સુરક્ષા ક્ષેત્રો સાથે સંરેખિત પરિબલોમાં વિભાજિત કરી શકાય છે: ગોપનીયતા, અખંડિતતા, ઉપલબ્ધતા અને જવાબદારી. જો નબળાઈનો ઉપયોગ કરવામાં આવે તો સિસ્ટમ પરની અસરની તીવ્રતાનો અંદાજ કાઢવાનો ધ્યેય છે. ગોપનીયતાની ખોટ, અખંડિતતાની ખોટ, ઉપલબ્ધતાની ખોટ અને જવાબદારીની ખોટ જેવા મુદ્દાઓ ધ્યાનમાં લેવાના છે. વ્યવસાયિક અસરનું મૂલ્યાંકન કરવા માટે ધ્યાનમાં લેવાના પરિબલોમાં નાણાકીય નુકસાન, પ્રતિષ્ઠાને નુકસાન, નીતિઓનું પાલન ન કરવું અને ગોપનીયતાનું ઉલ્લંઘન છે.

જોખમ નિયંત્રણ અસરકારકતા: જોખમ નિયંત્રણ અસરકારકતા

ઓળખાયેલ પ્રણાલીગત જોખમોની સંખ્યા પર આધાર રાખે છે, જોખમ મૂલ્યાંકનમાં સામેલ પ્રક્રિયા વિસ્તારોની ટકાવારી, ઘટાડાવામાં આવેલા મુખ્ય જોખમોની ટકાવારી અને સામેલ ઘણા પરિબલોમાં મોનિટર કરાયેલા મુખ્ય જોખમોની ટકાવારી. રિસ્ક મેનેજમેન્ટ રિસ્ક મેનેજમેન્ટ અને રિસ્ક એસેસમેન્ટ ઈન્ફોર્મેશન સિક્યુરિટી મેનેજમેન્ટ (ISM) ના મુખ્ય ઘટકો છે.

જોખમ વ્યવસ્થાપન એ સંસ્થા દ્વારા ઉપયોગમાં લેવાતા માહિતી સંસાધનોની નબળાઈઓ અને જોખમોને ઓળખવાની અને સંપત્તિના મૂલ્યના આધારે ઘટાડવા માટે કયા પ્રતિકૂળ પગલાં લેવા તે નક્કી કરવાની પ્રક્રિયા છે. જોખમ વ્યવસ્થાપનની પ્રક્રિયા એક ચાલુ, પુનરાવર્તિત પ્રક્રિયા છે. તે અનિશ્ચિતપણે પુનરાવર્તિત થવું જોઈએ. વ્યવસાયનું વાતાવરણ સતત બદલાતું રહે છે અને દરરોજ નવા જોખમો અને નબળાઈઓ બહાર આવે છે. બીજું, જોખમોનું સંચાલન કરવા માટે વપરાતા કાઉન્ટરમેઝર્સ (નિયંત્રણો) ની પસંદગીએ ઉત્પાદકતા, ખર્ચ, કાઉન્ટરમેઝર્સની અસરકારકતા અને માહિતીના મૂલ્ય વચ્ચે સંતુલન જાળવવું જોઈએ. સંપત્તિ સુરક્ષિત છે.

તેનાથી વિપરિત, જોખમનું મૂલ્યાંકન અલગ સમયના બિંદુઓ પર કરવામાં આવે છે (દા.ત. વર્ષમાં એકવાર, માંગ પર, વગેરે) અને આગામી મૂલ્યાંકનનું પ્રદર્શન થાય ત્યાં સુધી - મૂલ્યાંકિત જોખમોનો અસ્થાયી દૃષ્ટિકોણ પ્રદાન કરે છે. જોખમ વ્યવસ્થાપનનો હેતુ સંસ્થામાં જોખમનું સંચાલન કરવા માટેનું

મુખ્ય કારણ સંસ્થાના મિશન અને સંપત્તિનું રક્ષણ કરવાનું છે.

તેથી, જોખમ વ્યવસ્થાપન મેનેજમેન્ટ હોવું જોઈએ

તકનીકી કાર્યને બદલે કાર્ય. જોખમને સમજવું, અને ખાસ કરીને, સિસ્ટમ માટેના ચોક્કસ જોખમોને સમજવાથી સિસ્ટમના માલિકને સંસ્થાના મૂલ્યને અનુરૂપ માહિતી સિસ્ટમનું રક્ષણ કરવાની મંજૂરી મળે છે. હકીકત એ છે કે તમામ સંસ્થાઓ પાસે મર્યાદિત સંસાધનો છે અને જોખમ ક્યારેય શૂન્ય સુધી ઘટાડી શકાતું નથી. તેથી, જોખમને સમજવું, ખાસ કરીને જોખમની તીવ્રતા, સંસ્થાઓને દુર્લભ સંસાધનોને પ્રાધાન્ય આપવા માટે પરવાનગી આપે છે. જોખમનું મૂલ્યાંકન (તબક્કાઓ) જોખમનું મૂલ્યાંકન કરવાનો હેતુ સંસાધનોને ક્યાં નિર્દેશિત કરવા તે નક્કી કરવામાં મેનેજમેન્ટને મદદ કરવાનો છે. માટે ચાર મૂળભૂત વ્યૂહરચના છે જોખમનું સંચાલન: શમન, સ્થાનાંતરણ, સ્વીકૃતિ અને અવગણના.

મિટિગેશન મિટિગેશન એ સૌથી સામાન્ય રીતે માનવામાં આવતી જોખમ વ્યવસ્થાપન વ્યૂહરચના છે. શમનમાં ખામીને ઠીક કરવી અથવા ખામી સાથે સંકળાયેલી સંભાવના અથવા અસરને ઘટાડવા માટે અમુક પ્રકારના વળતરકારી નિયંત્રણનો સમાવેશ થાય છે. ટેકનિકલ સુરક્ષા ખામી માટેનું સામાન્ય નિવારણ એ છે કે વિક્રેતા દ્વારા પ્રદાન કરવામાં આવેલ પેચ ઈન્સ્ટોલ કરવું. કેટલીકવાર શમન વ્યૂહરચના નક્કી કરવાની પ્રક્રિયાને નિયંત્રણ વિશ્લેષણ કહેવામાં આવે છે. ટ્રાન્સફર ટ્રાન્સફર એ બીજા પક્ષને તમારા વતી જોખમ સ્વીકારવાની મંજૂરી આપવાની પ્રક્રિયા છે. IT સિસ્ટમ્સ માટે આ વ્યાપકપણે કરવામાં આવતું નથી, પરંતુ દરેક વ્યક્તિ તેના અંગત જીવનમાં આખો સમય કરે છે. કાર, આરોગ્ય અને જીવન વીમો એ જોખમને સ્થાનાંતરિત કરવાના તમામ માર્ગો છે. આ કિસ્સાઓમાં, જોખમ વ્યક્તિ પાસેથી વીમા કંપની સહિત વીમા ધારકોના પૂલમાં સ્થાનાંતરિત થાય છે. નોંધ કરો કે આ સંભાવનાને ઘટાડતું નથી અથવા કોઈપણ ખામીઓને સુધારતું નથી, પરંતુ તે એકંદરે અસર (મુખ્યત્વે નાણાકીય) ઘટાડે છે. સંસ્થા સ્વીકૃતિ સ્વીકૃતિ એ સિસ્ટમને જાણીતા જોખમ સાથે કામ કરવાની મંજૂરી આપવાની પ્રથા છે. ઘણા ઓછા જોખમો સરળ રીતે સ્વીકારવામાં આવે છે. જે જોખમોને ઘટાડવા માટે અત્યંત ઊંચી કિંમત હોય છે તે પણ ઘણીવાર સ્વીકારવામાં આવે છે. મેનેજમેન્ટ દ્વારા સ્વીકારવામાં આવતા ઉચ્ચ જોખમોથી સાવચેત રહો. ખાતરી કરો કે આ વ્યૂહરચના લેખિતમાં છે અને નિર્ણય લેતા મેનેજર દ્વારા સ્વીકારવામાં આવે છે. ઘણીવાર એવા જોખમો સ્વીકારવામાં આવે છે જે સ્વીકારવા જોઈએ નહીં, અને પછી જ્યારે ઘૂસપેઠ થાય છે, ત્યારે આઈટી સુરક્ષા કર્મચારીઓને જવાબદાર ગણવામાં આવે છે. સામાન્ય રીતે, વ્યવસાય સંચાલકો, આઈટી સુરક્ષા કર્મચારીઓ નહીં, સંસ્થા વતી જોખમ સ્વીકારવા માટે અધિકૃત છે.

ત્યાગ

અવગણના એ સિસ્ટમ અથવા તો સિસ્ટમના નબળા પાસાને દૂર કરવાની પ્રથા છે. દાખલા તરીકે, જોખમ મૂલ્યાંકન દરમિયાન, એક વેબસાઈટ ખુલ્લી પડી હતી જે વિક્રેતાઓને તેમના ઈન્વોઈસ જોવા દે છે, ઓળખ તરીકે HTML ફાઈલ નામમાં એમ્બેડ કરેલ વિક્રેતા ID નો ઉપયોગ કરીને અને વિક્રેતા ઈડ કોઈ પ્રમાણીકરણ અથવા અધિકૃતતા નથી. અંગે જાણ કરવામાં આવી ત્યારે વેબ પેજીસ અને સંસ્થા માટેનું જોખમ, મેનેજમેન્ટે વેબ પેજીસને દૂર કરવાનો અને અન્ય મિકેનિઝમ દ્વારા વેન્ડર ઈન્વોઈસ આપવાનું નક્કી કર્યું. આ કિસ્સામાં, સંવેદનશીલ વેબ પૃષ્ઠોને દૂર કરીને જોખમ ટાળવામાં આવ્યું હતું

જોખમ આકારણીના પ્રકાર ક્વોન્ટિટેટિવ રિસ્ક એસેસમેન્ટ ક્વોન્ટિટેટિવ રિસ્ક એસેસમેન્ટ નાણાકીય સંસ્થાઓ અને વીમા કંપનીઓ દ્વારા ઉપયોગમાં લેવાતી પદ્ધતિઓ પર આધારિત છે. માહિતી, સિસ્ટમો, વ્યવસાય

પ્રક્રિયાઓ, પુનઃપ્રાપ્તિ ખર્ચ વગેરેને મૂલ્યો સોંપીને, અસર અને તેથી જોખમને પ્રત્યક્ષ અને પરોક્ષ ખર્ચના સંદર્ભમાં માપી શકાય છે. પરંતુ તેનો ઉપયોગ સામાન્ય રીતે માહિતી પ્રણાલીઓમાં જોખમ માપવા માટે થતો નથી કારણ કે

- 1 અસ્કયામતોને મૂલ્ય ઓળખવા અને સોંપવામાં મુશ્કેલીઓ, અને
- 2 આંકડાકીય માહિતીનો અભાવ જે આવર્તન નક્કી કરવાનું શક્ય બનાવશે. આમ, મોટાભાગના જોખમ મૂલ્યાંકન સાધનો કે જે આજે માહિતી પ્રણાલીઓ માટે ઉપયોગમાં લેવાય છે તે ગુણાત્મક જોખમના માપન છે. ગુણાત્મક જોખમ મૂલ્યાંકન ગુણાત્મક જોખમ મૂલ્યાંકન ધારે છે કે સંભાવના અને અસર મૂલ્યોમાં પહેલેથી જ મોટી માત્રામાં અનિશ્ચિતતા છે અને તેને વ્યાખ્યાયિત કરે છે, અને તેથી જોખમ, કંઈક અંશે વ્યક્તિલક્ષી અથવા ગુણાત્મક દ્રષ્ટિએ. જથ્થાત્મક જોખમ આકારણીના મુદ્દાઓની જેમ, મોટી મુશ્કેલી ગુણાત્મક જોખમ મૂલ્યાંકનમાં સંભાવના અને અસર મૂલ્યોને વ્યાખ્યાયિત કરવામાં આવે છે. તદ્દુપરાંત, આ મૂલ્યોને એવી રીતે વ્યાખ્યાયિત કરવાની જરૂર છે જે બહુવિધ જોખમ મૂલ્યાંકનોમાં સમાન ભીંગડાઓનો સતત ઉપયોગ કરવાની મંજૂરી આપે છે. ગુણાત્મક જોખમ મૂલ્યાંકન સામાન્ય રીતે “ઉચ્ચ”, “મધ્યમ” અને “નીચા” ના જોખમ પરિણામો આપે છે. જો કે, અસર અને સંભાવના વ્યાખ્યા કોષ્ટકો અને અસરનું વર્ણન પ્રદાન કરીને, સંસ્થાના મેનેજમેન્ટને આકારણીની પર્યાપ્ત રીતે સંચાર કરી શકાય છે. જોખમ મૂલ્યાંકનમાં નીચેની ક્રિયાઓ અને પ્રવૃત્તિઓનો સમાવેશ થાય છે:

- અસ્કયામતોની ઓળખ;
- કાનૂની અને વ્યવસાયિક જરૂરિયાતોની ઓળખ કે જે ઓળખાયેલ અસ્કયામતો માટે સંબંધિત છે
- ઓળખવામાં આવેલી સંપત્તિનું મૂલ્યાંકન, ધ્યાનમાં લેતા ઓળખાયેલ કાનૂની અને
- વ્યાપાર જરૂરિયાતો અને ગોપનીયતા, અખંડિતતા અને ઉપલબ્ધતાના નુકશાનની અસરો

ડિરેક્ટરી સેવાઓ (Directory services)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- ડિરેક્ટરી અને ડિરેક્ટરી સેવાનું વર્ણન કરો
- ડિરેક્ટરી સેવાઓના લાભોનું વર્ણન કરો
- ડિરેક્ટરી સેવાઓના વિવિધ અમલીકરણોનો ઉલ્લેખ કરો
- સક્રિય નિર્દેશિકાની તાર્કિક અને ભૌતિક રચનાનું વર્ણન કરો
- વૈશ્વિક સૂચિ અને જૂથ નીતિનું વર્ણન કરો.

- 1 એક સરળ વૃક્ષની રચનામાં પ્રદર્શિત થાય છે, તેથી તેઓને શોધવા અને સંચાલિત કરવા માટે સરળ છે. આ બંધારણની અંદર, દરેક સંસાધનને એક ઓબ્જેક્ટ તરીકે ઓળખાતા ચિહ્ન દ્વારા પ્રદર્શિત કરવામાં આવે છે. ઓબ્જેક્ટ પસંદ કરીને, તેની સેટિંગ્સ ઉપલબ્ધ છે, અને નેટવર્ક એડમિનિસ્ટ્રેટર તેને યોગ્ય લાગે તેમ તેમાં ફેરફાર કરી શકે છે.
- 2 વપરાશકર્તાઓ: તમે જે સંસાધનોને અધિકારો આપવામાં આવ્યા છે તેને એક્સેસ કરવા માટે તમે સિંગલ લોગ ઓનનો ઉપયોગ કરી શકો છો. દરેક સર્વર પર લોગ ઈન કરવા અથવા દરેક પ્રિન્ટર અથવા અન્ય ઉપકરણને પ્રમાણિત કરવાને બદલે, ડિરેક્ટરીમાં આ માહિતી શામેલ છે. તમે એકવાર લોગ ઓન કરો અને નેટવર્ક એડમિનિસ્ટ્રેટરે તમને

- ઓળખાયેલ અસ્કયામતો માટે નોંધપાત્ર જોખમો અને નબળાઈઓની ઓળખ;
- ની સંભાવનાનું મૂલ્યાંકન ધમકીઓ અને થવાની નબળાઈઓ
- જોખમની ગણતરી;
- પૂર્વવ્યાખ્યાયિત જોખમ સ્કેલ સામે જોખમોનું મૂલ્યાંકન.

ICT સપ્લાય ચેઇન મેનેજમેન્ટ માટેના જોખમો ICT સપ્લાય ચેઇન માટેના જોખમો ગુપ્તતા, અખંડિતતા, અથવા માહિતી અથવા માહિતી પ્રણાલીની ઉપલબ્ધતાના નુકસાનથી ઉદ્ભવે છે અને સંસ્થાકીય કામગીરી (મિશન, કાર્યો, છબી અથવા પ્રતિષ્ઠા સહિત) પર સંભવિત પ્રતિકૂળ અસરોને પ્રતિબિંબિત કરે છે. સંસ્થાકીય અસ્કયામતો, વ્યક્તિઓ અને અન્ય સંસ્થાઓ. આવા જોખમો ઢોળાવવાળા સોફ્ટવેર ડેવલપમેન્ટનું સ્વરૂપ લઈ શકે છે, ઉત્પાદનો મોકલવામાં આવે તે પહેલાં તેનું અપૂરતું પરીક્ષણ, અને ઓછામાં ઓછા ખર્ચાળ ભાગોનો ઉપયોગ કરી શકે છે, ભલે તેનો અર્થ એ થાય કે તે અધિકૃત ન હોય. સપ્લાય ચેઇન રિસ્ક મેનેજમેન્ટ (SCRM) એ આ જોખમોને સમજવાની પ્રક્રિયા છે, તેમની વ્યાપાર અસરો અને કેવી રીતે તેનું સંચાલન સિસ્ટમ જીવનચક્ર દરમિયાન સપ્લાય ચેઇનની નબળાઈઓ અને શોષણોને ઘટાડી શકાય છે.

અસરકારક ICT SCRM માટે પ્રક્રિયાઓ, પ્રક્રિયાઓ અને જરૂરી છે

સાધનો કે જે સંસ્થાઓને SCRM સિદ્ધાંતો લાગુ કરવાની મંજૂરી આપે છે સમગ્ર ICTમાં સતત સિસ્ટમો એવો એક સિદ્ધાંત

નકલી ભાગોના જોખમને ઘટાડવાનું છે કારણ કે તે અણધારી વર્તન, પ્રારંભિક નિષ્ફળતા અથવા વધુ ખરાબ તરફ દોરી શકે છે. આથી નકલી ભાગોને અધિકૃત ભાગોથી અલગ પાડવું જરૂરી બને છે. આ લાક્ષણિકતાઓને વ્યક્ત કરવા માટે એક સંરચિત ભાષાની જરૂર છે, જેમ કે સપ્લાય ચેઇનના તમામ સભ્યો તેમના વિશે વાતચીત કરી શકે, અને જેનો ઉપયોગ નકલી વસ્તુઓ વિશે અન્ય લોકોને ચેતવણી આપવા અથવા તેના માપદંડો વ્યક્ત કરવા માટે થઈ શકે. કાયદેસર વસ્તુઓ. કાયદેસર અને ગેરકાયદેસર ઘટકોના આ અવલોકનક્ષમ લક્ષણોનું વર્ણન કરવા માટે એક સંરચિત ભાષા એ સપ્લાય ચેઇન જોખમ ઘટાડવાનું એક સાધન છે.

આપવામાં આવેલા અધિકારો દ્વારા જે કંઈપણ મંજૂરી આપી હોય તે તમે કરી શકો છો.

- 3 સુરક્ષા : માત્ર એક જ ડોમેન હોવાનો અર્થ એ છે કે સિંગલ સિક્યોરિટી પોલિસી અને એડમિનિસ્ટ્રેટર્સના એક સેટ દ્વારા વધુ સારી સુરક્ષા. જો તમારી પાસે બહુવિધ ડોમેન્સ અને ફોરેસ્ટ હોય, તો દરેકના પોતાના એડમિનિસ્ટ્રેટર હોય છે. એક નબળું પરંતુ વિશ્વસનીય ડોમેન અન્ય તમામ જંગલો અને ડોમેન્સનો પર્દાફાશ કરે છે. માત્ર એક જ ડોમેન સાથે, સંસ્થા-વ્યાપી સુરક્ષા નીતિ લાગુ કરવી પણ વધુ સરળ છે
- 4 સિંગલ પ્લેટફોર્મ: એ સિંગલ ડાયરેક્ટરી સર્વિસ અથવા ગ્લોબલ કેટેલોગ (GC) નો અર્થ છે મોનિટરિંગ અને મેસેજિંગ સહિત અન્ય તમામ ડિરેક્ટરી વેર સેવાઓ માટે એક જ પ્લેટફોર્મ.

- 5 ઝડપી જમાવટ: ફક્ત એક જ ડોમેન સાથેની સંસ્થામાં શરૂ થાય છે અને શેર કરેલ એકાઉન્ટ ડેટાબેઝ સોલ્યુશન્સ માત્ર એક જ વાર જમાવવામાં આવે છે, જેનો અર્થ છે કે જો સંસ્થા પાસે બહુવિધ અને અલગ ડોમેન્સ હોય તો તેના કરતાં કંપની-વ્યાપી ડિપ્લોયમેન્ટ વધુ ઝડપી હોય છે.
 - 6 સિંગલ મેનેજમેન્ટ: ઈન્ફ્રાસ્ટ્રક્ચર સિંગલ મેનેજમેન્ટ ઈન્ફ્રાસ્ટ્રક્ચર હોવાનો અર્થ એ છે કે અન્ય તમામ ડિરેક્ટરી સેવાઓના કાર્યો માટે માત્ર એક ઈન્ફ્રાસ્ટ્રક્ચર છે, જેમ કે સોફ્ટવેર ડિપ્લોયમેન્ટ, ઈન્વેન્ટરી અને ઓબ્જેક્ટ મેનેજમેન્ટ શેરિંગ અને ડેલિગેશન (જેમ કે વપરાશકર્તા ખાતાઓ માટે).
 - 7 સિંગલ ગ્રૂપ પોલિસી કન્ટેનર (GPC): એક જ GPC સાથે, મેનેજમેન્ટ પોલિસીને માત્ર એક જ વાર વ્યાખ્યાયિત કરવાની જરૂર છે, અને મેન્યુઅલી ગ્રૂપ પોલિસી ઓબ્જેક્ટ્સ (GPOs) ની નિકાસ અને આયાત કરવાની જરૂર વગર સમગ્ર એન્ટરપ્રાઇઝમાં ઉપયોગ કરી શકાય છે.
 - 8 બેકઅપ અને પુનઃપ્રાપ્તિ: માત્ર એક જ ડોમેન હોવાનો અર્થ વધુ સારી સ્થિતિસ્થાપકતા છે કારણ કે દરેક સ્થાનમાં સંપૂર્ણ ડોમેન બેકઅપ હોય છે.
 - 9 ઓછા હાર્ડવેર: બહુવિધ ડોમેન્સ ધરાવતી સંસ્થામાં, દરેક સ્થાનને બે ડોમેન નિયંત્રકો (DCs)ની જરૂર હોય છે. એક જ ડોમેન સાથે, દરેક સ્થાનને માત્ર એક જ ડીસીની જરૂર છે કારણ કે જો સ્થાનિક ડીસી નિષ્ફળ જાય, તો સ્થાનો હબ ડીસીનો ઉપયોગ કરી શકે છે. ઘટાડેલા હાર્ડવેરનો અર્થ એ પણ છે કે ઓછા લાઇસન્સ, ઓછા મેનેજમેન્ટ સોફ્ટવેર અને સર્વર મેનેજમેન્ટ માટે ઓછા ઓવરહેડ. પાછળ પડવાની પણ જરૂર નથી રીમોટ ડીસી ઉપર કારણ કે રીમોટ ડીસી માત્ર એ જ માહિતી ધરાવે છે જે કેન્દ્રીય ડીસી ધારે છે કે ડીસી માત્ર ડિરેક્ટરી સેવાઓ કરે છે. ડિરેક્ટરી સેવાઓના અમલીકરણો ડિરેક્ટરી સેવાઓ એ ઓપન સિસ્ટમ્સ ઈન્ટરકનેક્શન (OSI) પહેલનો એક ભાગ હતો જેથી ઉદ્યોગમાં દરેક વ્યક્તિ મલ્ટિ વેન્ડર ઈન્ટરઓપરેબિલિટી પ્રદાન કરવા માટે સામાન્ય નેટવર્ક ધોરણો સાથે સંમત થાય. 1980ના દાયકામાં, ઈન્ટરનેશનલ ટેલિકોમ્યુનિકેશન યુનિયન (ITU) અને ઈન્ટરનેશનલ ઓર્ગનાઇઝેશન ફોર સેન્ડર્ડાઇઝેશન J(ISO) એ ડિરેક્ટરી સેવાઓ માટે ધોરણોના સમૂહ - X.500 સાથે આવ્યા હતા. લાઇટ વેઇટ ડિરેક્ટરી એક્સેસ પ્રોટોકોલ, LDAP, જે X.500 ની ડિરેક્ટરી માહિતી સેવાઓ પર આધારિત છે, પરંતુ તે TCP/IP સ્ટેક અને X.500 પ્રોટોકોલ DAP ની સ્ટ્રિંગ એન્કોડિંગ યોજનાનો ઉપયોગ કરે છે. LDAP/X.500 આધારિત અમલીકરણોમાં આ છે:
 - એક્ટિવ ડિરેક્ટરી: વિન્ડોઝ માટે માઇક્રોસોફ્ટની આધુનિક ડિરેક્ટરી સેવા, X.500 ડિરેક્ટરીમાંથી ઉદ્ભવતી, એક્સચેન્જ સર્વરમાં ઉપયોગ માટે બનાવવામાં આવી છે, જે પ્રથમ Windows સાથે મોકલવામાં આવી છે. 2000 સર્વર અને વિન્ડોઝના કમિક વર્ઝન દ્વારા સપોર્ટેડ છે.
 - અપાચે ડિરેક્ટરી સર્વર: Java માં લખેલી ડિરેક્ટરી સેવા, LDAP, Kerberos 5 અને ચેન્જને સપોર્ટ કરતી
- પાસવર્ડ પ્રોટોકોલ. LDAPv3 પ્રમાણિત. અપાચે
- ડિરેક્ટરી સર્વર એ અપાચે સોફ્ટવેર ફાઉન્ડેશનનો ટોચનો સ્તરનો પ્રોજેક્ટ પણ છે.
 - eDirectory: આ NetIQ ની ડિરેક્ટરી સેવાઓનો અમલ છે. તે વિન્ડોઝ, નેટવેર, લિનક્સ અને યુનિક્સના કેટલાક ફ્લેવર સહિત બહુવિધ આર્કિટેક્ચરને સપોર્ટ કરે છે અને લાંબા સમયથી વપરાશકર્તા વહીવટ, ગોઠવણી વ્યવસ્થાપન અને સોફ્ટવેર મેનેજમેન્ટ માટે ઉપયોગમાં લેવાય છે. eDirectory ઓળખ વ્યવસ્થાપન ઉત્પાદનોની વ્યાપક શ્રેણીમાં કેન્દ્રિય ઘટક તરીકે વિકસિત થઈ છે. તે પહેલા નોવેલ ડિરેક્ટરી સર્વિસીસ તરીકે જાણીતી હતી.

- Red Hat ડિરેક્ટરી સર્વર: Red Hat એ એક ડિરેક્ટરી સેવા બહાર પાડી, જે તેણે AOL ના નેટસ્કેપ સિક્યોરિટી સોલ્યુશન્સ યુનિટ પાસેથી પ્રાપ્ત કરી, એક કોમર્શિયલ પ્રોડક્ટ તરીકે જે Red Hat Enterprise Linux પર Red Hat ડિરેક્ટરી સર્વર કહેવાય છે અને સમુદાયે 389 ડિરેક્ટરી સર્વર પ્રોજેક્ટને સપોર્ટ કર્યો છે.
- ઓરેકલ ઈન્ટરનેટ ડિરેક્ટરી: (OID) ઓરેકલ કોર્પોરેશનની ડિરેક્ટરી સેવા છે, જે તેની સાથે સુસંગત છે

LDAP સંસ્કરણ 3.

- સન જાવા સિસ્ટમ ડિરેક્ટરી સર્વર: સન માઇક્રોસિસ્ટમ્સની વર્તમાન ડિરેક્ટરી સેવા ઓફર કરે છે.
- OpenDS: ઓપન સોર્સ ડિરેક્ટરી સેવા

જાવામાં શરૂઆતથી અમલીકરણ, સન માઇક્રોસિસ્ટમ્સ દ્વારા સમર્થિત.

- IBM Tivoli ડિરેક્ટરી સર્વર: તે ઓપન LDAP ના જૂના પ્રકાશનનું કસ્ટમાઇઝ્ડ બિલ્ડ છે.
- Windows NT ડિરેક્ટરી સેવાઓ (NTDS), પછીથી

સક્રિય ડિરેક્ટરીનું નામ બદલીને, ભૂતપૂર્વ NT ને બદલે છે

ડોમેન સિસ્ટમ.

ઓપનએલડીએપી : તે તમામ વર્તમાન કોમ્પ્યુટરને સપોર્ટ કરે છે

આર્કિટેક્ચર, જેમાં યુનિક્સ અને યુનિક્સ ડેરિવેટિવ્ઝ, લિનક્સ, વિન્ડોઝ, z/OS, અને વિવિધ એમ્બેડેડ/રીઅલ ટાઇમ સિસ્ટમનો સમાવેશ થાય છે.

ત્યાં પણ પુષ્કળ ખુલ્લા છે ઓપન LDAP અને કર્બોસ સહિત ડિરેક્ટરી સેવાઓ બનાવવા માટે સ્ત્રોત સાધનો

પ્રોટોકોલ અને સામ્બા સોફ્ટવેર.

સક્રિય માર્ગદર્શન

- એક્ટિવ ડિરેક્ટરી (AD) એ ડિરેક્ટરી સેવા છે જે Microsoft એ Windows ડોમેન નેટવર્ક્સ માટે વિકસાવી છે અને પ્રક્રિયાઓ અને સેવાઓના સમૂહ તરીકે મોટાભાગની Windows Server ઓપરેટિંગ સિસ્ટમ્સમાં શામેલ છે.
- એક AD ડોમેન નિયંત્રક વિન્ડોઝ ડોમેન પ્રકારના નેટવર્કમાં તમામ વપરાશકર્તાઓ અને કમ્પ્યુટર્સને પ્રમાણિત કરે છે અને અધિકૃત કરે છે-તમામ કમ્પ્યુટર્સ માટે સુરક્ષા નીતિઓ સોંપવા અને લાગુ કરવા અને સોફ્ટવેર ઈન્સ્ટોલ અથવા અપડેટ કરવા. ઉદાહરણ તરીકે, જ્યારે વપરાશકર્તા વિન્ડોઝ ડોમેનનો ભાગ હોય તેવા કમ્પ્યુટરમાં લૉગ ઈન કરે છે, ત્યારે એક્ટિવ ડિરેક્ટરી સબમિટ કરેલા પાસવર્ડની તપાસ કરે છે અને તે નક્કી કરે છે કે વપરાશકર્તા સિસ્ટમ એડમિનિસ્ટ્રેટર છે કે સામાન્ય વપરાશકર્તા.[3]
- એક્ટિવ ડિરેક્ટરી લાઇટવેઇટ ડિરેક્ટરી એક્સેસ પ્રોટોકોલ (LDAP) વર્ઝન 2 અને 3, માઇક્રોસોફ્ટનું કર્બોસ વર્ઝન અને DNS નો ઉપયોગ કરે છે.
- સક્રિય ડિરેક્ટરી યુઝર મેનેજમેન્ટને પણ સરળ બનાવે છે કારણ કે તે આ બધા યુઝર અને કોમ્પ્યુટર સંબંધિત માટે એક રીપોઝીટરી તરીકે કામ કરે છે માહિતી
- AD તેના એક્સેસ પ્રોટોકોલ તરીકે LDAP નો ઉપયોગ કરે છે.
- AD તેની લોકેટર સેવા તરીકે DNS પર આધાર રાખે છે, જે DNS ક્વેરીઝ દ્વારા ક્લાયન્ટ્સને ડોમેન કંટ્રોલર શોધવા માટે સક્ષમ બનાવે છે.

એક્ટિવ ડિરેક્ટરીનું લોજિકલ માળખું એક્ટિવ ડિરેક્ટરી એવિતરિટ ડેટાબેઝ છે જે નેટવર્ક સંસાધનો વિશેની માહિતી તેમજ ડિરેક્ટરી સક્ષમ એપ્લિકેશનમાંથી એપ્લિકેશન વિશિષ્ટ ડેટાને સંગ્રહિત અને સંચાલિત કરે છે.

સક્રિય નિર્દેશિકા સંચાલકોને તત્વોને ગોઠવવાની મંજૂરી આપે છે

નેટવર્ક (જેમ કે વપરાશકર્તાઓ, કમ્પ્યુટર્સ, ઉપકરણો અને તેથી વધુ) ની શ્રેણીબદ્ધ નિયંત્રણ માળખામાં. સક્રિય નિર્દેશિકામાં, સંસાધનો તાર્કિક માળખામાં ગોઠવવામાં આવે છે, અને સંસાધનોનું આ જૂથ તાર્કિક રીતે સંસાધનને તેના ભૌતિક સ્થાનને બદલે તેના નામ દ્વારા શોધવા માટે સક્ષમ બનાવે છે. એડી લોજિકલ સ્ટ્રક્ચરના ફાયદા

- તાર્કિક માળખું માત્ર નિર્દિષ્ટ જૂથો (OU) ને સંસાધનોની એક્સેસ પ્રદાન કરીને વધુ નેટવર્ક સુરક્ષા પ્રદાન કરે છે.
- લોજિકલ માળખું નેટવર્કના વહીવટ, ગોઠવણી અને નિયંત્રણ દ્વારા નેટવર્ક મેનેજમેન્ટને સરળ બનાવે છે.
- ની તાર્કિક રચના વચ્ચેનો સંબંધ

ડોમેન્સ અને ફોરેસ્ટ સમગ્ર સંસ્થામાં સંસાધનોની વહેંચણીને સરળ બનાવે છે.

- લોજિકલ માળખું સરળ નેટવર્ક પૂરું પાડે છે

સંચાલન, તે નેટવર્ક સંસાધનો પરનો ભાર ઘટાડે છે અને માલિકીની કુલ કિંમત ઘટાડે છે.

એડી લોજિકલ સ્ટ્રક્ચરના ઘટકો

તાર્કિક માળખાના ઘટકોનો એકબીજા સાથે સંબંધ છે તે સંગ્રહિત ડેટાની એક્સેસને નિયંત્રિત કરવા માટેનું સંચાલન કરે છે અને શોધે છે કે જંગલમાં વિવિધ ડોમેન્સ વચ્ચે ડેટાનું સંચાલન કેવી રીતે થશે.

- ઑબ્જેક્ટ્સ: જેમ કે વપરાશકર્તા, કમ્પ્યુટર, જૂથ, પ્રિન્ટર વગેરે...
- સંસ્થાકીય એકમો - કોઈપણ ફોલ્ડરની જેમ પરંતુ સક્રિય નિર્દેશિકાના નિયંત્રણમાં
- ડોમેન્સ - વસ્તુઓ માટે તાર્કિક સીમાઓ

- વૃક્ષો - બહુવિધ ડોમેન્સ માટે તાર્કિક સીમા

- જંગલો - બહુવિધ વૃક્ષો માટે તાર્કિક સીમા

એકંદરે, માઈક્રોસોફ્ટ ડોમેન નિયંત્રક તરીકે ચાલતું એક ભૌતિક મશીન ચોક્કસ કાર્યો કરવા માટે સમર્પિત 'એ ઓપરેશન માસ્ટર' ની મદદથી આ તમામ તાર્કિક વિભાગોને નિયંત્રિત કરી શકે છે.

ટોપ લેવલ કન્ટેનર જંગલ છે. વન એ વૃક્ષોનો સંગ્રહ છે જે સામાન્ય વૈશ્વિક સૂચિ, નિર્દેશિકા યોજના, તાર્કિક માળખું અને નિર્દેશિકા ગોઠવણીને શેર કરે છે. જંગલ એ સુરક્ષા સીમાનું પ્રતિનિધિત્વ કરે છે જેની અંદર વપરાશકર્તાઓ, કમ્પ્યુટર્સ, જૂથો અને અન્ય વસ્તુઓ સુલભ છે.

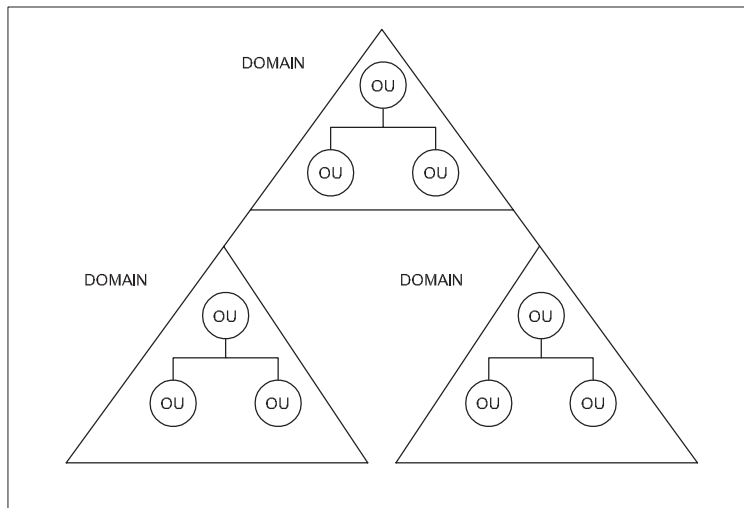
જંગલોની અંદર ડોમેન છે. ડોમેન એ નેટવર્ક ઑબ્જેક્ટ્સ (કમ્પ્યુટર, વપરાશકર્તાઓ, ઉપકરણો) ના લોજિકલ જૂથ તરીકે વ્યાખ્યાયિત કરવામાં આવે છે જે સમાન સક્રિય ડિરેક્ટરી ડેટાબેઝને શેર કરે છે. ડોમેન્સની અંદર સંસ્થાકીય એકમો છે. OU ડોમેનને વંશવેલો પ્રદાન કરી શકે છે, તેના વહીવટને સરળ બનાવી શકે છે અને સંચાલકીય અથવા ભૌગોલિક દ્રષ્ટિએ સંસ્થાના માળખાને મળતા આવે છે. OUs માં અન્ય OUs શામેલ હોઈ શકે છે ડોમેન્સ આ અર્થમાં કન્ટેનર છે. OU એ સ્તર છે કે જેના પર સામાન્ય રીતે વહીવટી સત્તાઓ સોંપવામાં આવે છે, પરંતુ પ્રતિનિધિત્વ કરી શકાય છે વ્યક્તિગત વસ્તુઓ અથવા વિશેષતાઓ પર પણ. આને લોજિકલ મોડલ કહેવામાં આવે છે કારણ કે તે જમાવટના ભૌતિક પાસાઓથી સ્વતંત્ર છે, જેમ કે દરેક ડોમેન અને નેટવર્ક ટોપોલોજીમાં જરૂરી ડોમેન નિયંત્રકોની સંખ્યા. આકૃતિ 1 જંગલો, ડોમેન્સ અને સંસ્થાકીય એકમો વચ્ચેનો સંબંધ દર્શાવે છે. આકૃતિ 1 એક્ટિવ ડિરેક્ટરી ફોરેસ્ટ્સ, ડોમેન્સ અને ઓર્ગનાઈઝેશનલ યુનિટ્સ (OUs) વચ્ચેનો સંબંધ એક્ટિવ ડિરેક્ટરીનું ભૌતિક માળખું સક્રિય ડિરેક્ટરી ભૌતિક માળખું ક્યારે અને ક્યાં લોગોન અને પ્રતિકૃતિ ટ્રાફિક થાય છે તેની તપાસ કરે છે. એક્ટિવ ડાયરેક્ટરીનું ભૌતિક માળખું નેટવર્કમાં હાજર તમામ ભૌતિક સબનેટ્સ ધરાવે છે જેમ કે ડોમેન નિયંત્રકો અને ડોમેન કંટ્રોલર્સ વચ્ચેની નકલ

સક્રિય નિર્દેશિકાનું ભૌતિક માળખું:

- ડોમેન કંટ્રોલર્સ: આ કમ્પ્યુટર્સ Microsoft Windows સર્વર 2003/2000 અને એક્ટિવ ડિરેક્ટરી ચલાવે છે. દરેક ડોમેન કંટ્રોલર પ્રતિકૃતિ, સંગ્રહ અને પ્રમાણીકરણ જેવા વિશિષ્ટ કાર્યો કરે છે. તે

Fig 1

FOREST



RELATIONSHIP BETWEEN ACTIVE DIRECTORY FORESTS, DOMAINS, AND ORGANIZATIONAL UNITS (OUS)

FOREST
CONTAINS DOMAINS.
USED TO DEFINE THE
SCOPE OF AUTHORITY
OF THE ADMINISTRATORS.

DOMAIN
CONTAINS OUS.
USED TO PARTITION THE
DIRECTORY DATA AND
CONTROL REPLICATION.

ORGANIZATIONAL UNIT
CONTAINS USER AND
COMPUTER ACCOUNTS.
USED TO DELEGATE CONTROL
AND APPLY POLICIES.

CP20N13212131

મહત્તમ એક ડોમેનને સપોર્ટ કરી શકે છે. દરેક ડોમેનમાં એક કરતા વધુ ડોમેન કંટ્રોલર રાખવાની હંમેશા સલાહ આપવામાં આવે છે.

• સક્રિય ડિરેક્ટરી સાઈટ્સ: આ સાઈટ્સનો સંગ્રહ છે

સારી રીતે જોડાયેલા કમ્પ્યુટર્સ. અમે શા માટે સાઈટ બનાવીએ છીએ તેનું કારણ એ છે કે ડોમેન નિયંત્રકો સાઈટની અંદર વારંવાર વાતચીત કરી શકે છે. આ રીતે તે સાઈટની અંદર લેટન્સીને ઘટાડે છે કહે છે કે એક ડોમેન નિયંત્રક પર કરવામાં આવેલા ફેરફારોને અન્ય ડોમેન નિયંત્રકો પર નકલ કરવા માટે. સાઈટ બનાવવા પાછળનું બીજું કારણ ડોમેન નિયંત્રકો વચ્ચે બેન્ડવિડ્થને ઓપ્ટિમાઇઝ કરવાનું છે જે વિવિધ સ્થળોએ સ્થિત છે.

કોમ્પ્યુટરના વાસ્તવિક ભૌતિક સ્થાનને જાણ્યા વિના સામાન્ય લોકલ એરિયા નેટવર્ક (LAN) કનેક્ટિવિટી શેર કરતા તમામ IP સબનેટને સાઈટ કહેવામાં આવે છે. વૈશ્વિક સૂચિ એ ડેટા સ્ટોરેજ સ્ત્રોત છે મલ્ટિડોમેન એક્ટિવ ડિરેક્ટરી ડોમેન સર્વિસીસ (એડીડીએસ) ફોરેસ્ટમાં જોવા મળતી વસ્તુઓની આંશિક રજૂઆતો ધરાવે છે. વૈશ્વિક કેટલોગ ખાસ કરીને વૈશ્વિક કેટલોગ સર્વર તરીકે સોંપેલ ડોમેન નિયંત્રકો પર સંગ્રહિત થાય છે. તે વાસ્તવિક ડોમેન નામ જાણ્યા વિના કોઈપણ ડોમેનમાં ઓબ્જેક્ટ શોધી શકે છે.

વૈશ્વિક કેટલોગ પર નિર્દેશિત કરાયેલી શોધો ઝડપી છે કારણ કે તેમાં વિવિધ ડોમેન નિયંત્રકોના રેફરલ્સનો સમાવેશ થતો નથી.

OU OU માં સંસાધનોનું આયોજન કરવું એ વપરાશકર્તા, કમ્પ્યુટર અને અન્ય ઓબ્જેક્ટ માહિતીને વધુ સરળતાથી સમજી શકાય તેવા લેઆઉટમાં ગોઠવવાની પ્રાથમિક પદ્ધતિ છે. સંસ્થા પાસે રુટ સંસ્થાકીય એકમ છે જ્યાં ત્રણ નેસ્ટેડ સંસ્થાકીય એકમો મૂકવામાં આવે છે. આ નેસ્ટિંગ સંસ્થાને નેટવર્ક સંસાધનોના સરળ જોવા અને વહીવટ માટે બહુવિધ કન્ટેનરમાં વપરાશકર્તાઓને વિતરિત કરવા સક્ષમ બનાવે છે.

વહીવટના સરળ સંગઠન અને પ્રતિનિધિમંડળ માટે OU ને વધુ સંસાધન OU માં વિભાજિત કરી શકાય છે. દૂર-દૂરના કાર્યાલયોમાં સ્થાનિક વહીવટ માટે તેમના પોતાના OU હોઈ શકે છે તેમજ. જો કે, એ સમજવું અગત્યનું છે કે જો સંસ્થાને અન્ય સમૂહને વહીવટ સોંપવાની ચોક્કસ જરૂરિયાત હોય તો જ OU બનાવવું જોઈએ.

સંચાલકો જો તે જ વ્યક્તિ અથવા લોકોનું જૂથ

સમગ્ર ડોમેનનું સંચાલન કરો, OU ઉમેરીને પર્યાવરણની જટિલતા વધારવાની જરૂર નથી. હકીકતમાં, ઘણા બધા OU જૂથ નીતિઓ, લોગોન્સ અને અન્ય પરિબળોને અસર કરી શકે છે.

અલગ વિભાગોને તેમના પોતાના વપરાશકર્તાઓ પર વહીવટી નિયંત્રણના વિવિધ સ્તરોની મંજૂરી આપવા માટે OU ની રચના કરી શકાય છે. ઉદાહરણ તરીકે, એન્જિનિયરિંગ વિભાગના સેક્ટરીને તેના પોતાના OUમાં વપરાશકર્તાઓ માટે પાસવર્ડ રીસેટ કરવાનું નિયંત્રણ સોંપવામાં આવી શકે છે. આ પરિસ્થિતિઓમાં OU ના ઉપયોગનો બીજો ફાયદો એ છે કે વપરાશકર્તાઓને સરળતાથી ખેંચી શકાય છે અને એક OU થી બીજામાં ડ્રોપ કરી શકાય છે. ઉદાહરણ તરીકે, જો વપરાશકર્તાઓને એક વિભાગમાંથી બીજા વિભાગમાં ખસેડવામાં આવે છે, તો તેમને તેમના નવા વિભાગના OUમાં ખસેડવું અત્યંત સરળ છે. તે ધ્યાનમાં રાખવું અગત્યનું છે કે જ્યારે પણ વ્યવસ્થાપક માળખાકીય ફેરફારો કરવા માટે યોગ્ય લાગે ત્યારે OU માળખું ફલાય પર સુધારી શકાય છે. આ એક્ટિવ ડિરેક્ટરીને કોઈપણ સમયે ફેરફાર કરવાનો વધારાનો ફાયદો આપે છે.

ગ્રુપ પોલિસી એ Microsoft Windows NT પરિવારની ઓપરેટિંગ સિસ્ટમની વિશેષતા છે જે વપરાશકર્તા ખાતાઓ અને કમ્પ્યુટર એકાઉન્ટ્સના કાર્યકારી વાતાવરણને નિયંત્રિત કરે છે. ગ્રુપ પોલિસી ઓપરેટિંગ સિસ્ટમ્સ, એપ્લિકેશન્સ અને વપરાશકર્તાઓનું કેન્દ્રિય સંચાલન અને ગોઠવણી પૂરી પાડે છે. સક્રિય ડિરેક્ટરી પર્યાવરણમાં સેટિંગ્સ. સ્થાનિક જૂથ નીતિ (“LGPO” અથવા “LocalGPO”) તરીકે ઓળખાતી જૂથ નીતિનું સંસ્કરણ, એકલ અને નોન-ડોમેન કમ્પ્યુટર્સ પર જૂથ નીતિ ઓબ્જેક્ટ મેનેજમેન્ટને પણ મંજૂરી આપે છે. ગ્રુપ પોલિસી એ એક્ટિવ ડિરેક્ટરી ડિપ્લોય કરવાના ટોચના કારણોમાંનું એક છે કારણ કે તે તમને યુઝર અને કોમ્પ્યુટર ઓબ્જેક્ટ્સનું સંચાલન કરવાની મંજૂરી આપે છે. જૂથ નીતિ, આંશિક રીતે, વપરાશકર્તાઓ કમ્પ્યુટર સિસ્ટમ પર શું કરી શકે છે અને શું કરી શકતા નથી તેનું નિયંત્રણ કરે છે, ઉદાહરણ તરીકે: પાસવર્ડ જટિલતા નીતિ લાગુ કરવા કે જે વપરાશકર્તાઓને વધુ પડતા સરળ પાસવર્ડને પસંદ કરતા અટકાવે છે, દૂરસ્થ કમ્પ્યુટરથી અજાણ્યા વપરાશકર્તાઓને કનેક્ટ થવા માટે પરવાનગી આપે છે અથવા અટકાવે છે. નેટવર્ક શેર, વિન્ડોઝ ટાસ્ક મેનેજરની ઍક્સેસને અવરોધિત કરવા અથવા ચોક્કસ ફોલ્ડર્સની ઍક્સેસને પ્રતિબંધિત કરવા. આવા રૂપરેખાંકનોના સમૂહને ગ્રુપ પોલિસી ઓબ્જેક્ટ (GPO) કહેવામાં આવે છે. એક્ટિવ ડિરેક્ટરી બેકઅપ અને રિસ્ટોર એક્ટિવ ડિરેક્ટરી એ તમારા ઈન્ફ્રાસ્ટ્રક્ચરના સૌથી મહત્વપૂર્ણ ઘટકોમાંનું એક છે. જો તે નીચે જાય, તો તમારું નેટવર્ક નકામું રેન્ડર થશે. તેથી, વ્યવસાયની સાતત્યતા અને અનુપાલનની ખાતરી કરવા માટે, તમે સક્રિય ડિરેક્ટરી માટે નક્કર બેકઅપ અને પુનઃપ્રાપ્તિ યોજના હોવી જરૂરી છે. જ્યાં ગ્રુપ પોલિસી પ્રેફરન્સ સેટિંગ ગોઠવેલ હોય અને સમકક્ષ ગ્રુપ પોલિસી સેટિંગ પણ ગોઠવેલ હોય, તો ગ્રુપ પોલિસી સેટિંગનું મૂલ્ય અગ્રતા મેળવશે. જૂથ નીતિ ડોમેનની સુરક્ષા છે. જ્યારે તમે સિસ્ટમ સ્ટેટો બેકઅપ લો છો ત્યારે બેકઅપ યુટિલિટી આપોઆપ શોધી કાઢશે અને તેમાં સમાવેશ કરશે.

એક્સેસ કંટ્રોલ, ઓડિટ અને ટેસ્ટિંગ (Access Control, Audit and testing)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- એક્સેસ કંટ્રોલમાં હેતુ અને પગલાંનું વર્ણન કરો
- એક્સેસ કંટ્રોલમાં વિવિધ સ્તરોનું વર્ણન કરો
- એક્સેસ કંટ્રોલ મિકેનિઝમનું વર્ણન કરો
- સુરક્ષા માટે માહિતીનું વર્ગીકરણ કરો
- એક્સેસ કંટ્રોલ પ્રોટોકોલ્સની યાદી બનાવો
- સુરક્ષા ઓડિટનું વર્ણન કરો
- સુરક્ષા ઓડિટના મહત્વનું વર્ણન કરો
- ઓડિટ સુરક્ષા કરવાની પ્રક્રિયાનું વર્ણન કરો
- નબળાઈ આકારણી અને પ્રવેશ પરીક્ષણનું વર્ણન કરો
- વિવિધ પ્રકારના સુરક્ષા ઓડિટ સાધનોનો ઉલ્લેખ કરો.

પરિચય: સંરક્ષિત માહિતીની એક્સેસ એવા લોકો માટે પ્રતિબંધિત હોવી જોઈએ કે જેઓ માહિતીને એક્સેસ કરવા માટે અધિકૃત છે. એક્સેસ કંટ્રોલનો હેતુ કમ્પ્યુટર સિસ્ટમનો કાયદેસર વપરાશકર્તા કરી શકે તેવી ક્રિયાઓ અથવા કામગીરીને મર્યાદિત કરવાનો છે, તેમજ વપરાશકર્તાઓ વતી એક્ઝિક્યુટ કરતા પ્રોગ્રામ્સને શું કરવાની મંજૂરી છે. આ રીતે એક્સેસ કંટ્રોલ એવી પ્રવૃત્તિને રોકવાનો પ્રયાસ કરે છે જે સુરક્ષાના ભંગ તરફ દોરી શકે છે.

એક્સેસ કંટ્રોલ મિકેનિઝમ્સની સુસંસ્કૃતતા સુરક્ષિત કરવામાં આવી રહેલી માહિતીના મૂલ્યના પ્રમાણસર હોવી જોઈએ - માહિતી જેટલી વધુ સંવેદનશીલ અથવા મૂલ્યવાન હશે તેટલી વધુ મજબૂત નિયંત્રણ પદ્ધતિઓ હોવી જરૂરી છે. જે પાયા પર એક્સેસ કંટ્રોલ મિકેનિઝમ બનાવવામાં આવે છે તે ઓળખ અને પ્રમાણીકરણથી શરૂ થાય છે.

એક્સેસ નિયંત્રણ સામાન્ય રીતે ત્રણ પગલામાં પ્રાપ્ત થાય છે: ઓળખ, પ્રમાણીકરણ અને અધિકૃતતા.

- ઓળખ એ કોઈ વ્યક્તિ કોણ છે અથવા કંઈક શું છે તેનું નિવેદન છે. વપરાશકર્તા તે વપરાશકર્તા નામ દાખલ કરીને દાખલ કરીને પોતાને ઓળખે છે.
- પ્રમાણીકરણ એ વપરાશકર્તાની ઓળખ ચકાસવાની પ્રક્રિયા છે. પ્રમાણીકરણ એ માહિતી ખાતરી (IA) ના પાંચ સ્તંભોમાંથી એક છે. અન્ય ચાર અખંડિતતા, ઉપલબ્ધતા, ગોપનીયતા અને અસ્વીકાર છે.

પાસવર્ડ દાખલ કરવો એ પ્રમાણીકરણ માટેની સામાન્ય પદ્ધતિ છે. પરંતુ પાસવર્ડ આધારિત પ્રમાણીકરણ કમ્પ્યુટર નેટવર્ક પર ઉપયોગ માટે યોગ્ય નથી. સમગ્ર નેટવર્ક પર મોકલવામાં આવેલા પાસવર્ડને અટકાવી શકાય છે અને ત્યારબાદ હેકર્સ દ્વારા તેનો દુરુપયોગ કરી શકાય છે. સુરક્ષાની ચિંતા ઉપરાંત, પાસવર્ડ આધારિત ઓથેન્ટિકેશન અસુવિધાજનક છે કારણ કે જ્યારે પણ નેટવર્ક સર્વિસ એક્સેસ કરવાની હોય ત્યારે વપરાશકર્તાએ પાસવર્ડ દાખલ કરવો પડે છે. વ્યવહારો માટે આ સિસ્ટમની નબળાઈ (જેમ કે પૈસાની આપ-લે) એ છે કે પાસવર્ડ્સ ઘણીવાર ચોરાઈ શકે છે, આકસ્મિક રીતે જાહેર થઈ શકે છે અથવા ભૂલી જઈ શકે છે. અધિકૃતતા : કોઈ વ્યક્તિ, પ્રોગ્રામ અથવા કોમ્પ્યુટરને સફળતાપૂર્વક ઓળખવામાં આવે અને પ્રમાણિત કરવામાં આવે તે પછી તે નક્કી કરવું આવશ્યક છે કે તેમને કયા માહિતી સંસાધનોને એક્સેસ કરવાની મંજૂરી છે અને કઈ ક્રિયાઓ તેઓને (ચલાવો, જુઓ, બનાવો, કાઢી નાખો અથવા બદલો) કરવાની મંજૂરી આપવામાં આવશે. આને અધિકૃતતા કહેવામાં આવે છે. માહિતી અને અન્ય કમ્પ્યુટિંગ સેવાઓને એક્સેસ કરવાની અધિકૃતતા વહીવટી નીતિઓ અને

પ્રક્રિયાઓથી શરૂ થાય છે. નીતિઓ સૂચવે છે કે કઈ માહિતી અને કમ્પ્યુટિંગ સેવાઓને એક્સેસ કરી શકાય છે, કોના દ્વારા અને કઈ શરતો હેઠળ..

પ્રમાણીકરણ પદ્ધતિઓ

- ઓથેન્ટિકેશન માટે ડાયજેસ્ટનો ઉપયોગ કરીને ઈવસ્ક્રોપિંગના જોખમને નિયંત્રિત કરી શકાય છે. કનેક્ટિંગ પાર્ટી મૂલ્ય મોકલે છે, સામાન્ય રીતે ક્લાયન્ટનું IP એડ્રેસ, ટાઇમ સ્ટેમ્પ અને વધારાની ગુપ્ત માહિતીનો હેશ. કારણ કે આ હેશ દરેક એક્સેસ કરેલ યુઝરઆઈ માટે અનન્ય છે, અન્ય કોઈ દસ્તાવેજો એક્સેસ કરી શકાતા નથી અને શોધ્યા વિના અન્ય આઈપી એડ્રેસમાંથી તેનો ઉપયોગ કરી શકાતો નથી. હેશિંગને કારણે પાસવર્ડ પણ છીનવી લેવા માટે સંવેદનશીલ નથી. જોકે, સિસ્ટમ -મેન-ઈન-ધ મિડલ એટેક જેવા સક્રિય હુમલાઓ માટે સંવેદનશીલ છે.
- વન-ટાઇમ પાસવર્ડ્સ: પાસવર્ડ પુનઃઉપયોગ સાથે સંકળાયેલી સમસ્યાઓને ટાળવા માટે, વન-ટાઇમ પાસવર્ડ્સ વિકસાવવામાં આવ્યા હતા. વન-ટાઇમ પાસવર્ડ બે પ્રકારના હોય છે, ચેલેન્જ રિસ્પોન્સ પાસવર્ડ અને પાસવર્ડ લિસ્ટ.
- પબ્લિક-કી ક્રિપ્ટોગ્રાફી: પબ્લિક કી ક્રિપ્ટોગ્રાફી ખૂબ જ જટિલ ગાણિતિક સમસ્યાઓ પર આધારિત છે જેને ખૂબ જ વિશિષ્ટ જ્ઞાનની જરૂર હોય છે. પબ્લિક કી ક્રિપ્ટોગ્રાફી બે કીનો ઉપયોગ કરે છે, એક ખાનગી અને બીજી સાર્વજનિક. બે ચાવીઓ જોડાયેલ છે એકસાથે અત્યંત જટિલ ગાણિતિક સમીકરણ દ્વારા. પ્રાઈવેટ કીનો ઉપયોગ કોમ્યુનિકેટિંગ મશીનો વચ્ચેના મેસેજને ડિક્રિપ્ટ કરવા અને એન્ક્રિપ્ટ કરવા માટે થાય છે. એન્ક્રિપ્શન અને સહીની ચકાસણી બંને સાર્વજનિક કી વડે પૂર્ણ થાય છે.
- શૂન્ય-જ્ઞાન પુરાવાઓ: શૂન્ય જ્ઞાન પુરાવાઓ યજમાનને કોઈપણ “ગુપ્ત માહિતી” જાહેર કર્યા વિના એક્સેસની મંજૂરી આપવા માટે અન્ય યજમાનને સહમત કરવાનું શક્ય બનાવે છે. પ્રમાણીકરણના આ સ્વરૂપમાં સામેલ હોસ્ટ સામાન્ય રીતે પ્રમાણીકરણને અંતિમ સ્વરૂપ આપવા માટે ઘણી વખત વાતચીત કરે છે.
- જાહેર કી ઈન્ફ્રાસ્ટ્રક્ચરના ભાગ રૂપે પ્રમાણપત્ર સત્તાધિકારી (CA) દ્વારા જારી કરાયેલ અને ચકાસાયેલ ડિજિટલ પ્રમાણપત્રોનો ઉપયોગ ઈન્ટરનેટ પર પ્રમાણીકરણ કરવા માટેનો બીજો પ્રમાણભૂત માર્ગ માનવામાં આવે છે.
- પ્રમાણીકરણની બીજી પદ્ધતિ, બાયોમેટ્રિક્સ, વપરાશકર્તાની હાજરી અને જૈવિક મેકઅપ (એટલે કે, રેટિના અથવા ફિંગરપ્રિન્ટ્સ) પર આધારિત છે. આ ટેકનોલોજી હેકર્સ માટે કોમ્પ્યુટર સિસ્ટમમાં ઘૂસવાનું

વધુ મુશ્કેલ બનાવે છે. સિક્યોરિટી સોલ્યુશન જેવા પ્રમાણીકરણ માટે વિશેષ પ્રોટોકોલનો ઉપયોગ કરવો લેયર (SSL), IP SEC, સિક્યોર શેલ (SSH), કર્બોરોસ ઓથેન્ટિકેશન અને એક્સ્ટેન્સિબલ ઓથેન્ટિકેશન પ્રોટોકોલ (EAP) વગેરે.

સુરક્ષા વ્યવસ્થાપક સંસ્થાની સુરક્ષા નીતિના આધારે અધિકૃતતાઓનો ડેટાબેઝ જાળવી રાખે છે. સંદર્ભ મોનિટર એ નિર્ધારિત કરવા માટે અધિકૃત ડેટાબેઝની સલાહ લે છે કે શું વપરાશકર્તા ઓપરેશન કરવાનો પ્રયાસ કરે છે તે ઓપરેશન કરવા માટે ખરેખર અધિકૃત છે. ઓડિટીંગ મોનિટર કરે છે અને સિસ્ટમમાં સંબંધિત પ્રવૃત્તિનો રેકૉર્ડ રાખે છે.

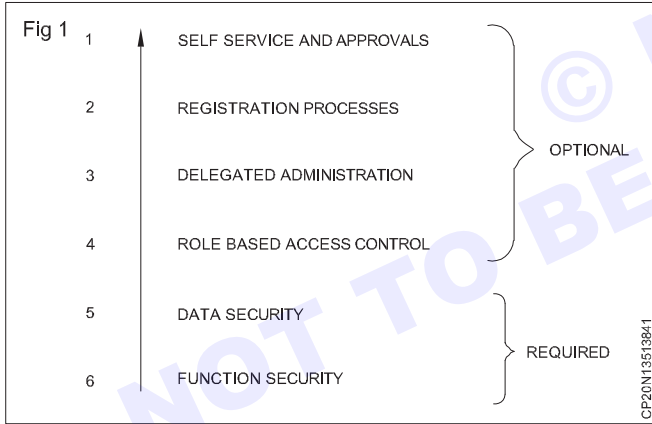
પ્રમાણીકરણ અને એક્સેસ નિયંત્રણ વચ્ચે સ્પષ્ટ તફાવત કરવો મહત્વપૂર્ણ છે. વપરાશકર્તાની ઓળખને યોગ્ય રીતે સ્થાપિત કરવી એ પ્રમાણીકરણ સેવાની જવાબદારી છે. એક્સેસ કંટ્રોલ ધારે છે કે રેફરન્સ મોનિટર દ્વારા એક્સેસ કંટ્રોલ લાગુ કરતા પહેલા યુઝરનું પ્રમાણીકરણ સફળતાપૂર્વક ચકાસવામાં આવ્યું છે.

એક્સેસ કંટ્રોલના અનુગામી સ્તરો એક્સેસ કંટ્રોલ ક્રમિક સ્તરોમાં લાગુ કરવામાં આવે છે અને દરેક સ્તર તેની આગળના સ્તર પર બને છે.

સંસ્થાઓ વૈકલ્પિક રીતે, વિવિધ સ્તરોને ઉપાડી શકે છે

ઓટોમેશન અને માપનીયતાની ડિઝાઇનના આધારે તેઓ હાલના કાર્ય પર બિલ્ડ કરવા માંગે છે અને ડેટા

સુરક્ષા મોડલ. આને અમલમાં મૂકવા માટે વિવિધ મોડેલો હોઈ શકે છે. ઉદાહરણ તરીકે, ઓરેકલ યુઝર મેનેજમેન્ટ પાસે એક્સેસ કંટ્રોલના છ સ્તરો છે. (ફિગ 1 નો સંદર્ભ લો) મુખ્ય સુરક્ષા સ્તરોમાં શામેલ છે:



• કાર્ય સુરક્ષા

• ડેટા સુરક્ષા

આગળના ચાર સ્તરો ઓરેકલ યુઝર મેનેજમેન્ટનો ભાગ છે:

- ભૂમિકા-આધારિત એક્સેસ નિયંત્રણ
- નિયુક્ત વહીવટ
- નોંધણી પ્રક્રિયાઓ
- સ્વ સેવા અને મંજૂરીઓ

સામાન્ય રીતે, એક્સેસ કંટ્રોલ સાથેની શરૂઆત મૂળભૂત સિસ્ટમ એડમિનિસ્ટ્રેશન કાર્યોથી થાય છે, વધુ વિતરિત, વહીવટની સ્થાનિક રીતો તરફ આગળ વધે છે અને છેવટે વપરાશકર્તાઓને તેમના પોતાના પર કેટલાક મૂળભૂત, પૂર્વવ્યાખ્યાયિત નોંધણી કાર્યો કરવા સક્ષમ બનાવે છે. કોષ્ટક 1. સ્તરો એકબીજા પર કેવી રીતે બને છે તે સમજાવે છે.

સ્તર	વિવરણ
1	સ્વ સેવા અને મંજૂરીઓ
2	નોંધણી પ્રક્રિયાઓ
3	ભૂમિકા-આધારિત એક્સેસ નિયંત્રણ
4	નિયુક્ત વહીવટ
5	ડેટા સુરક્ષા
6	ફંક્શન સુરક્ષા

સિક્યોરિટી અને ડેટા સિક્યુરિટી મિકેનિઝમ્સ સિક્યુરિટી સિસ્ટમના બેઝ લેયરની રચના કરે છે અને તેમાં પરંપરાગત સિસ્ટમ એડમિનિસ્ટ્રેટિવ ક્ષમતાઓ શામેલ છે. તેઓ ચોક્કસ મેનૂની એક્સેસ આપીને વપરાશકર્તા વ્યવસ્થાપનના અવકાશને મૂળભૂત સિસ્ટમ વહીવટ સુધી મર્યાદિત કરે છે.

સ્થાનિક સંચાલકો: જ્યારે ડેટા સુરક્ષા અને કાર્ય સુરક્ષા સ્તરોમાં ભૂમિકા-આધારિત એક્સેસ કંટ્રોલ અને ડેલિગેટેડ એડમિનિસ્ટ્રેશન ઉમેરવામાં આવે છે, ત્યારે સિસ્ટમ એડમિનિસ્ટ્રેટર્સને સ્થાનિક વહીવટકર્તાઓને વિતરિત કરી શકાય છે જેઓ સંસ્થાના વપરાશકર્તાઓના સબસેટનું સંચાલન કરે છે.

અંતિમ વપરાશકર્તાઓ: નોંધણી પ્રક્રિયાઓ અને સ્વ સેવા અને મંજૂરીઓ કેટલાક નોંધણી કાર્યોને સ્વચાલિત કરીને સિસ્ટમ વહીવટને વધુ વિતરિત કરે છે જેથી અંતિમ વપરાશકર્તાઓ તેમને કરી શકે. અંતિમ વપરાશકર્તાઓ નવા વપરાશકર્તા એકાઉન્ટ્સ મેળવવા, સિસ્ટમમાં વધારાની એક્સેસની વિનંતી કરવા અને પાસવર્ડ્સ રીસેટ કરવાના કાર્યો કરી શકે છે.

સ્વ-સેવા અને મંજૂરીઓ: નોંધણી પ્રક્રિયાઓ જરૂરિયાતો અનુસાર ગોઠવાઈ ગયા પછી, વ્યક્તિઓ સ્વ-સેવા નોંધણી કાર્યો કરી શકે છે, જેમ કે નવા વપરાશકર્તા ખાતા મેળવવા અથવા સિસ્ટમમાં વધારાની એક્સેસની વિનંતી કરવી. વધુમાં, સંસ્થાઓ આ વિનંતીઓ માટે કસ્ટમાઈઝ્ડ એપ્રુવલ ફ્લો બનાવવા માટે ઓરેકલ એપ્રુવલ્સ મેનેજમેન્ટ એન્જિનનો ઉપયોગ કરી શકે છે.

પ્રિવેન્ટિવ ત્રણ પ્રકારના હોય છે નિયંત્રણો: વહીવટી નિયંત્રણો

- નીતિઓ/પ્રક્રિયાઓ: પ્રક્રિયાઓ કઈ રીતે કરવી જોઈએ તે ઓળખવા. આ તાલીમ, ડિટેક્ટીવ કંટ્રોલ અને ઓડિટ સાથે સાથે જ જવું જોઈએ.

ભૌતિક નિયંત્રણો

- બાયોમેટ્રિક સેન્સર્સ, સ્માર્ટ કાર્ડ વગેરેનો ઉપયોગ કરીને. ટેકનિકલ (લોજિકલ) નિયંત્રણો
- એન્ક્રિપ્શન
- પાસવર્ડ્સ અને ટોકન્સ

સ્તર	વિવરણ
1	સ્વ સેવા અને મંજૂરીઓ
2	નોંધણી પ્રક્રિયાઓ
3	ભૂમિકા-આધારિત એક્સેસ નિયંત્રણ
4	નિયુક્ત વહીવટ
5	ડેટા સુરક્ષા
6	ફંક્શન સુરક્ષા

- બાયોમેટ્રિક્સ
- ઓ.એસ. અને એપ્લિકેશન નિયંત્રણો
- ઓળખ અને અધિકૃતતા ટેકનોલોજી એક્સેસ કંટ્રોલ મિકેનિઝમ્સ
એક્સેસ કંટ્રોલ માટે નીચેના મોડલ/મિકેનિઝમ્સ છે. ઉપરોક્ત દરેક એક્સેસ મોડલ્સના પોતાના ફાયદા અને ગેરફાયદા છે. સંસ્થા દ્વારા યોગ્ય એક્સેસ મોડલની પસંદગી વિવિધ પરિબલો જેમ કે વ્યવસાયનો પ્રકાર, વપરાશકર્તાઓની સંખ્યા, સંસ્થાની સુરક્ષા નીતિ વગેરેને ધ્યાનમાં લઈને થવી જોઈએ.
- ભૂમિકા આધારિત એક્સેસ કંટ્રોલ (RBAC)
એક્સેસ નિર્ણયો સંસ્થા અથવા વપરાશકર્તા આધારમાં વ્યક્તિની ભૂમિકાઓ અને જવાબદારીઓ પર આધારિત છે. RBAC ને બિન-વિવેકાધીન એક્સેસ નિયંત્રણ તરીકે પણ ઓળખવામાં આવે છે કારણ કે વપરાશકર્તાને વિશેષાધિકારો વારસામાં મળે છે જે તેની ભૂમિકા સાથે જોડાયેલા હોય છે. વપરાશકર્તાને જે ભૂમિકા સોંપવામાં આવશે તેના પર તેનું નિયંત્રણ નથી.
- વિવેકાધીન એક્સેસ નિયંત્રણ (DAC)
નામ સૂચવે છે તેમ, આ એક્સેસ કંટ્રોલ મોડલ વપરાશકર્તાની વિવેકબુદ્ધિ પર આધારિત છે. એટલે કે, સંસાધનનો માલિક કરી શકે છે
આધારિત અન્ય વપરાશકર્તાઓને તે સંસાધન પર એક્સેસ અધિકારો આપો
તેના વિવેકબુદ્ધિ પર. એક્સેસ કંટ્રોલ લિસ્ટ (ACLs) એ DAC નું વિશિષ્ટ ઉદાહરણ છે. a પર “rwx” પરવાનગીઓનો ઉલ્લેખ કરવો
તમારી માલિકીની યુનિક્સ ફાઇલ એ DAC નું બીજું ઉદાહરણ છે વિન્ડોઝ સહિતની મોટાભાગની ઓપરેટિંગ સિસ્ટમ્સ, યુનિક્સની ફ્લેવર્સ DAC મોડલ પર આધારિત છે.
- ફરજિયાત એક્સેસ કંટ્રોલ (MAC)
આ મોડેલમાં, વપરાશકર્તાઓ/માલિકો તેમની ફાઇલોને કોણ એક્સેસ કરી શકે છે તે નક્કી કરવાના વિશેષાધિકારનો આનંદ માણતા નથી. અહીં ઓપરેટિંગ સિસ્ટમ એ નિર્ણય લેનાર છે જે વપરાશકર્તાના નિર્ણયને ઓવરરાઇડ કરે છે ઈચ્છાઓ આ મોડેલમાં દરેક વિષય (વપરાશકર્તાઓ) અને ઓબ્જેક્ટ (સંસાધનો) વર્ગીકૃત કરવામાં આવે છે અને સુરક્ષા લેબલ સાથે સોંપવામાં આવે છે. સુરક્ષા નીતિ સાથે વિષય અને ઓબ્જેક્ટના સુરક્ષા લેબલ્સ નક્કી કરે છે કે વિષય ઓબ્જેક્ટને એક્સેસ કરી શકે છે કે નહીં. સિક્યોરિટી ઓફિસર દ્વારા ઓબ્જેક્ટને કેવી રીતે એક્સેસ કરવામાં આવે છે તેના નિયમો, એડમિનિસ્ટ્રેટર દ્વારા ગોઠવવામાં આવે છે, ઓપરેટિંગ સિસ્ટમ દ્વારા લાગુ કરવામાં આવે છે અને સુરક્ષા તકનીકો દ્વારા સપોર્ટેડ છે.
DAC ની તુલનામાં આ એક કડક અને તેના બદલે સ્થિર એક્સેસ કંટ્રોલ મોડલ છે અને મોટાભાગે લશ્કરી સંસ્થાઓ માટે અનુકૂળ છે જ્યાં ડેટા વર્ગીકરણ અને ગોપનીયતા મુખ્ય મહત્વ ધરાવે છે. ખાસ પ્રકારની યુનિક્સ ઓપરેટિંગ સિસ્ટમ MAC મોડલ પર આધારિત છે.
- એટ્રિબ્યુટ આધારિત એક્સેસ કંટ્રોલ (ABAC)
આ વપરાશકર્તાની મનસ્વી વિશેષતાઓ, ઓબ્જેક્ટની મનસ્વી વિશેષતાઓ અને પર્યાવરણની પરિસ્થિતિઓના આધારે વપરાશકર્તા વિનંતીઓ મંજૂર અથવા નકારવા માટે છે જે હાથ પરની નીતિઓ સાથે વધુ સુસંગત હોઈ શકે છે.
પાસવર્ડ કેક્રીંગની પદ્ધતિઓ અને તેના વિરોધી પગલાં

હેક્સ દ્વારા એકાઉન્ટ હેક કરવા અથવા વ્યક્તિગત માહિતી ચોરી કરવા માટે ઘણી પદ્ધતિઓનો ઉપયોગ કરવામાં આવે છે. પાસવર્ડ કેક કરવા માટે સૌથી વધુ ઉપયોગમાં લેવાતી કેટલીક પદ્ધતિઓ અને તેના પ્રતિરોધક પગલાં નીચે મુજબ છે:

1 બ્રુટ - ફોર્સ એટેક

બ્રુટ-ફોર્સ એટેકનો ઉપયોગ કરીને કોઈપણ પાસવર્ડ કેક કરી શકાય છે.

બ્રુટ-ફોર્સ એટેક જ્યાં સુધી સાચો પાસવર્ડ મેચ ન થાય ત્યાં સુધી સંખ્યાઓ, અક્ષરો અને વિશિષ્ટ અક્ષરોના દરેક સંભવિત સંયોજનોનો પ્રયાસ કરે છે. પાસવર્ડની જટિલતાને આધારે બ્રુટ ફોર્સ હુમલામાં ઘણો સમય લાગી શકે છે. કમ્પ્યુટરની ઝડપ અને પાસવર્ડની જટિલતાને આધારે.

કાઉન્ટર માપ : લાંબા અને જટિલ પાસવર્ડનો ઉપયોગ કરો. સંખ્યાઓ સાથે અપર અને લોઅરકેસ અક્ષરોના સંયોજનોનો ઉપયોગ કરવાનો પ્રયાસ કરો. આવા જટિલ અને લાંબા પાસવર્ડને કેક કરવામાં બ્રુટ ફોર્સ એટેક ખૂબ લાંબો સમય લેશે. તમે વારંવાર પાસવર્ડ બદલતા પણ રહી શકો છો.

2 સામાજિક ઈજનેરી

સોશિયલ એન્જિનિયરિંગ એ કોઈને તમારા પર વિશ્વાસ કરવા અને તેમની પાસેથી માહિતી મેળવવા માટે ચાલાકી કરવાની પ્રક્રિયા છે. ઉદાહરણ તરીકે, જો હેકર કોઈ સહકાર્યકરો અથવા મિત્રોના કમ્પ્યુટરનો પાસવર્ડ મેળવવાનો પ્રયાસ કરી રહ્યો હોય, તો તે તેને આઈટી વિભાગ અથવા બેંકનો હોવાનો ડોળ કરીને કોલ કરી શકે છે અને ફક્ત તેના લોગિન અથવા ક્રેડિટ કાર્ડની વિગતો માટે પૂછી શકે છે.

કાઉન્ટર મેઝર : ક્યારેય પણ ફોન પર ક્રેડિટ કાર્ડની વિગતો જેવી તમારી સંવેદનશીલ માહિતી આપશો નહીં.

3 ઉંદરો અને કી લોગર્સ કી લોગીંગ અથવા રેટિંગમાં હેકર પીડિતને કી લોગર અથવા ઉંદર મોકલે છે. આનાથી હેકર તેના કમ્પ્યુટર પર પીડિત જે કંઈ કરે છે તેનું નિરીક્ષણ કરી શકે છે. દરેક કીસ્ટ્રોક પાસવર્ડ સહિત લોગ થયેલ છે. આ ઉપરાંત હેકર પીડિતના કોમ્પ્યુટરને પણ કંટ્રોલ કરી શકે છે.

કાઉન્ટરમેઝર : સાયબર કાફે અથવા અન્ય કોઈના કમ્પ્યુટરથી તમારા બેંક એકાઉન્ટમાં ક્યારેય લોગિન ન કરો. જો તે ખૂબ જ મહત્વપૂર્ણ છે, તો લોગિન બાંધતી વખતે સ્ક્રીન અથવા વર્ચ્યુઅલ કીબોર્ડનો ઉપયોગ કરો. નવીનતમ એન્ટી વાયરસ સોફ્ટવેરનો ઉપયોગ કરો અને વ્યાખ્યાઓને અપડેટ રાખો.

4 ફિશીંગ

ફિશીંગ એ સૌથી સરળ અને લોકપ્રિય હેકિંગ પદ્ધતિ છે જેનો ઉપયોગ હેક્સ દ્વારા કોઈના એકાઉન્ટની વિગતો મેળવવા માટે કરવામાં આવે છે. ફિશીંગ હુમલામાં હેકર વાસ્તવિક વેબસાઈટનું નકલી પેજ મોકલે છે પીડિતને ફેસબુક, જીમેલની જેમ. જ્યારે કોઈ વ્યક્તિ તે નકલી પેજ દ્વારા લોગ ઇન કરે છે ત્યારે તેની વિગતો હેકરને મોકલવામાં આવે છે. આ નકલી પૃષ્ઠો સરળતાથી મફત વેબ હોસ્ટિંગ સાઈટ્સ પર બનાવી અને હોસ્ટ કરી શકાય છે.

કાઉન્ટર મેઝર : ફિશીંગ હુમલાઓ ટાળવા માટે ખૂબ જ સરળ છે.

આ ફિશીંગ પૃષ્ઠોના url વાસ્તવિક પૃષ્ઠથી અલગ છે. ઉદાહરણ તરીકે facebook ના ફિશીંગ પેજનું URL facbbbook.com જેવું દેખાઈ શકે છે (જેમ તમે જોઈ શકો છો કે બે “b” છે). હંમેશા ખાતરી કરો કે વેબસાઈટનું url સાચું છે.

5 રેઈન્બો ટેબલ

રેઈન્બો ટેબલ એ અક્ષરોના દરેક સંભવિત સંયોજન માટે હેશની વિશાળ પ્રી-કમ્પ્યુટેડ સૂચિ છે. પાસવર્ડ હેશ એ પાસવર્ડ છે જે ગાણિતિક અલ્ગોરિધમ જેમ કે md5 માંથી પસાર થયો છે અને ઓળખી ન શકાય તેવી વસ્તુમાં રૂપાંતરિત થાય છે. હેશ એ વન-વે એન્ક્રિપ્શન છે તેથી એકવાર પાસવર્ડ હેશ

થઈ જાય પછી હેશ કરેલી સ્ટ્રિંગમાંથી મૂળ સ્ટ્રિંગ મેળવવાનો કોઈ રસ્તો નથી.

કાઉન્ટર માપ : ખાતરી કરો કે તમે પાસવર્ડ પસંદ કર્યો છે જે લાંબો અને જટિલ છે. લાંબા અને જટિલ પાસવર્ડ માટે કોષ્ટકો બનાવવામાં ઘણો લાંબો સમય અને ઘણાં સંસાધનો લે છે.

6 અનુમાન લગાવવું આ એક સરળ પદ્ધતિ છે જે તમને સેકન્ડોમાં કોઈનો પાસવર્ડ મેળવવામાં મદદ કરે છે. જો હેકર તમને ઓળખે છે, તો તે તમારા પાસવર્ડનું અનુમાન લગાવવા માટે તે તમારા વિશે જાણે છે તે માહિતીનો ઉપયોગ કરી શકે છે. હેકર તમારો પાસવર્ડ મેળવવા માટે સોશિયલ એન્જિનિયરિંગ અને અનુમાનના સંયોજનનો પણ ઉપયોગ કરી શકે છે.

કાઉન્ટર મેઝર: તમારા પાસવર્ડ તરીકે તમારું નામ, અટક, ફોન નંબર અથવા જન્મ તારીખનો ઉપયોગ કરશો નહીં. તમારાથી સંબંધિત પાસવર્ડ બનાવવાનું ટાળવાનો પ્રયાસ કરો. જટિલ અને લાંબી બનાવો અક્ષરો અને સંખ્યાઓના સંયોજન સાથેનો પાસવર્ડ.

માહિતી માટે સુરક્ષા વર્ગીકરણ બધી માહિતી સમાન હોતી નથી અને તેથી બધી માહિતીને સમાન સુરક્ષાની જરૂર હોતી નથી. આ માટે સુરક્ષા વર્ગીકરણ અસાઈન કરવાની માહિતી જરૂરી છે. માહિતી વર્ગીકરણમાં પ્રથમ પગલું એ છે કે વરિષ્ઠ વ્યવસ્થાપનના સભ્યને વર્ગીકૃત કરવાની ચોક્કસ માહિતીના માલિક તરીકે ઓળખાવવી. આગળ, વર્ગીકરણ નીતિ વિકસાવો. નીતિ આ માટે સક્ષમ હોવી જોઈએ:

- માહિતીનો સંગ્રહ કરવો
- માહિતીનું પ્રસારણ
- વિવિધ વર્ગીકરણ લેબલોનું વર્ણન કરો,
- ચોક્કસ લેબલ અસાઈન કરવા માટેની માહિતી માટેના માપદંડોને વ્યાખ્યાયિત કરો, અને

દરેક વર્ગીકરણ માટે જરૂરી સુરક્ષા નિયંત્રણોની યાદી બનાવો.

- બિનજરૂરી માહિતીનો નિકાલ
- માહિતીની અખંડિતતાનું રક્ષણ કરવું
- યોગ્ય એક્સેસ અને જાહેરાતની મંજૂરી આપવી
- જવાબદારીની સ્થાપના.
- કેટલાક પરિબલો જે પ્રભાવિત કરે છે કે કઈ વર્ગીકરણ માહિતી સોંપવી જોઈએ તેમાં નીચેનાનો સમાવેશ થાય છે:
- તે માહિતી માટે કેટલું મૂલ્ય છે સંસ્થા
- માહિતી કેટલી જૂની છે અને
- માહિતી અપ્રચલિત થઈ ગઈ છે કે નહીં.
- કાયદાઓ અને અન્ય નિયમનકારી આવશ્યકતાઓ માહિતી સુરક્ષા માટેનું બિઝનેસ મોડલ સુરક્ષા વ્યાવસાયિકોને સિસ્ટમના પરિપ્રેક્ષ્યમાં સુરક્ષાની તપાસ કરવા સક્ષમ બનાવે છે, એવું વાતાવરણ ઊભું કરે છે કે જ્યાં સુરક્ષાને સર્વગ્રાહી રીતે સંચાલિત કરી શકાય, વાસ્તવિક જોખમોને સંબોધિત કરી શકાય.

માહિતી સુરક્ષા વર્ગીકરણ લેબલ્સનો પ્રકાર પસંદ કરેલ અને વપરાયેલ સંસ્થાની પ્રકૃતિ પર આધાર રાખે છે, ઉદાહરણ તરીકે:

- વ્યવસાય ક્ષેત્ર, લેબલો જેમ કે: સાર્વજનિક, સંવેદનશીલ, ખાનગી અને ગોપનીય.

- સરકારી ક્ષેત્રમાં, લેબલો જેમ કે: અવર્ગીકૃત, સંવેદનશીલ પરંતુ અવર્ગીકૃત, પ્રતિબંધિત, ગોપનીય, ગુપ્ત, ટોપ સિક્રેટ અને તેમના બિન-અંગ્રેજી સમકક્ષ.
- કોસ-સેક્ટરલ રચનાઓમાં, ટ્રાફિક લાઈટ પ્રોટોકોલ, જેમાં નીચેનાનો સમાવેશ થાય છે: સફેદ, લીલો, એમ્બર અને લાલ.

સંસ્થાના તમામ કર્મચારીઓ, તેમજ વ્યવસાયિક ભાગીદારોએ, વર્ગીકરણ યોજના પર પ્રશિક્ષિત હોવા જોઈએ અને દરેક વર્ગીકરણ માટે જરૂરી સુરક્ષા નિયંત્રણો અને હેન્ડલિંગ પ્રક્રિયાઓને સમજવી જોઈએ. ચોક્કસ માહિતી સંપત્તિનું વર્ગીકરણ કે જે સોંપવામાં આવ્યું છે તેની સમયાંતરે સમીક્ષા થવી જોઈએ જેથી વર્ગીકરણ સ્થિર છે.

માહિતી માટે અને સુરક્ષા સુનિશ્ચિત કરવા માટે યોગ્ય

વર્ગીકરણ દ્વારા જરૂરી નિયંત્રણો સ્થાને છે.

વર્ગીકરણ અને ડાઉનગ્રેડિંગ માહિતીને માત્ર તે સમય માટે જ વર્ગીકૃત અથવા નિયુક્ત કરવી જોઈએ જ્યારે તેને રક્ષણની જરૂર હોય, ત્યારબાદ તેને અવર્ગીકૃત અથવા ડાઉનગ્રેડ કરવાની હોય. આ એટલા માટે છે કારણ કે વર્ગીકૃત અથવા નિયુક્ત માહિતી સમય પસાર થવા અથવા ચોક્કસ ઘટનાઓની ઘટના સાથે તેની સંવેદનશીલતા ગુમાવશે. આ પ્રક્રિયા સુરક્ષા પ્રણાલીની એકંદર અખંડિતતામાં ફાળો આપે છે અને સુનિશ્ચિત કરે છે કે જાહેર જનતાના રસ ધરાવતા સભ્યોને માહિતી ઝડપથી અને અનૌપચારિક રીતે ઉપલબ્ધ કરવામાં આવે.

એક્સેસ કંટ્રોલ એડમિનિસ્ટ્રેશન એક્સેસ કંટ્રોલ એડમિનિસ્ટ્રેશન બે રીતે કરી શકાય છે.

- કેન્દ્રીકૃત : અહીં એક એન્ટિટી (વિભાગ અથવા વ્યક્તિગત) તમામ કોર્પોરેટ સંસાધનોની એક્સેસની દેખરેખ માટે જવાબદાર છે. આ પ્રકારનો વહીવટ વપરાશકર્તાઓના એક્સેસ અધિકારોને નિયંત્રિત કરવાની સુસંગત અને સમાન પદ્ધતિ પ્રદાન કરે છે. ઉદાહરણ: RADIUS, TACACS અને વ્યાસ

- વિકેન્દ્રિત

એક્સેસ કંટ્રોલ / ડેટા કલેક્શન પ્રોટોકોલ્સ: AAA (RADIUS, Diameter, and TACACS+)

RADIUS, Diameter, અને TACACS+ એ ત્રણ પ્રોટોકોલ છે

પ્રમાણીકરણ વહન, નેટવર્ક એક્સેસ સર્વર (NAS) વચ્ચેની અધિકૃતતા અને એકાઉન્ટિંગ (AAA) માહિતી જે તેની લિંક્સ અથવા અંતિમ વપરાશકર્તાઓને પ્રમાણિત કરવા માંગે છે. RADIUS TheRemote Authentication Dial-In User Service (RADIUS) એ લ્યુસેન્ટ ઈન્ટર નેટવર્કિંગ સિસ્ટમ્સ દ્વારા બનાવવામાં આવેલ ક્લાયન્ટ/સર્વર સુરક્ષા પ્રોટોકોલ છે. RADIUS એ ઈન્ટરનેટ ડ્રાફ્ટ છે

પ્રમાણભૂત પ્રોટોકોલ. વપરાશકર્તા પ્રોફાઈલ્સ કેન્દ્રિય સ્થાનમાં સંગ્રહિત થાય છે, જે RADIUS સર્વર તરીકે ઓળખાય છે. RADIUS ક્લાયન્ટ્સ (જેમ કે પોર્ટ માસ્ટર કોમ્યુનિકેશન સર્વર) વપરાશકર્તાઓને પ્રમાણિત કરવા માટે RADIUS સર્વર સાથે વાતચીત કરે છે. સર્વર ક્લાયન્ટને પાછા સ્પષ્ટ કરે છે કે પ્રમાણિત વપરાશકર્તા શું કરવા માટે અધિકૃત છે. જો કે RADIUS શબ્દ નેટવર્ક પ્રોટોકોલનો સંદર્ભ આપે છે જેનો ઉપયોગ ક્લાયન્ટ અને સર્વર વાતચીત કરવા માટે કરે છે, તે ઘણીવાર સમગ્ર ક્લાયન્ટ/સર્વર સિસ્ટમનો સંદર્ભ આપવા માટે વપરાય છે.

વ્યાસ: વ્યાસ એ પ્રમાણીકરણ, અધિકૃતતા છે,

અને કમ્પ્યુટર નેટવર્ક્સ માટે એકાઉન્ટિંગ પ્રોટોકોલ. તે તેના પહેલાના ઘણા ઓછા સક્ષમ RADIUS પ્રોટોકોલમાંથી વિકસિત અને તેને બદલે છે. વ્યાસ એપ્લિકેશન નવા આદેશો અને/અથવા વિશેષતાઓ ઉમેરીને બેઝ પ્રોટોકોલને વિસ્તૃત કરે છે. RADIUS ની સરખામણીમાં તે બહેતર, બહેતર પરિવહન, બહેતર સુરક્ષા, બહેતર પ્રોક્સીંગ, બહેતર સત્ર નિયંત્રણ અને બહેતર આંતરકાર્યક્ષમતા પ્રદાન કરે છે.

TACACS: ટર્મિનલ એક્સેસ કંટ્રોલર એક્સેસ-કંટ્રોલ

સિસ્ટમ (TACACS, સામાન્ય રીતે ટેક-એક્સની જેમ ઉચ્ચારવામાં આવે છે) નો સંદર્ભ આપે છે

રિમોટ હેન્ડલ કરતા સંબંધિત પ્રોટોકોલના પરિવાર માટે

નેટવર્ક એક્સેસ માટે પ્રમાણીકરણ અને સંબંધિત સેવાઓ

કેન્દ્રિય સર્વર દ્વારા નિયંત્રણ. મૂળ TACACS પ્રોટોકોલ, જે 1984નો છે, તેનો ઉપયોગ પ્રમાણીકરણ સર્વર સાથે વાતચીત કરવા માટે થતો હતો, જે સામાન્ય રીતે

જૂના UNIX નેટવર્ક્સ. વિસ્તૃત TACACS (XTACACS) એ છે

મૂળ પ્રોટોકોલ સાથે પાછળની સુસંગતતા વિના 1990 માં સિસ્કો સિસ્ટમ્સ દ્વારા રજૂ કરાયેલ TACACS માટે માલિકીનું વિસ્તરણ. ટર્મિનલ એક્સેસ કંટ્રોલર એક્સેસ કંટ્રોલ સિસ્ટમ પ્લસ (TACACS+) એ સિસ્કો દ્વારા વિકસિત પ્રોટોકોલ છે અને 1993માં ઓપન સ્ટાન્ડર્ડ તરીકે બહાર પાડવામાં આવ્યું છે.

વિકેન્દ્રિત એક્સેસ નિયંત્રણ

- વિકેન્દ્રિત એક્સેસ કંટ્રોલ એડમિનિસ્ટ્રેશન પદ્ધતિ સંસાધનોની નજીકના લોકો સુધી પહોંચનું નિયંત્રણ આપે છે
- આ અભિગમમાં, તે ઘણીવાર કાર્યકારી મેનેજર હોય છે જે કર્મચારીઓને એક્સેસ નિયંત્રણ અધિકારો સોંપે છે.
- આ પ્રકારના વહીવટ દ્વારા ફેરફારો ઝડપથી થઈ શકે છે કારણ કે માત્ર એક જ સંસ્થા સમગ્ર સંસ્થા માટે ફેરફારો કરી રહી નથી.
- ત્યાં તકરાર થવાની સંભાવના છે જે સંસ્થાને લાભ ન આપી શકે કારણ કે વિવિધ સંચાલકો અને વિભાગો સુરક્ષા અને એક્સેસ નિયંત્રણની પ્રેક્ટિસ અલગ અલગ રીતે કરી શકે છે.
- અમુક નિયંત્રણો ઓવરલેપ થવાની સંભાવના છે, જે કિસ્સામાં ક્રિયાઓ યોગ્ય રીતે પ્રતિબંધિત ન હોઈ શકે અથવા

પ્રતિબંધિત

- આ પ્રકારનું વહીવટ સતત નિયંત્રણ માટેની પદ્ધતિઓ પ્રદાન કરતું નથી, જેમ કે કેન્દ્રીયકૃત પદ્ધતિ હશે.

સુરક્ષા ઓડિટ એ અનિવાર્યપણે મૂલ્યાંકન છે કેવી રીતે

અસરકારક રીતે સંસ્થાની સુરક્ષા નીતિનો અમલ કરવામાં આવી રહ્યો છે. તે IT સિસ્ટમની નીતિ, રેકૉર્ડ અને પ્રવૃત્તિઓની સ્વતંત્ર સમીક્ષા અને પરીક્ષા છે.

ઈન્ફર્મેશન સિસ્ટમ્સ ઓડિટ મહત્વપૂર્ણ છે કારણ કે તે ખાતરી આપે છે કે IT સિસ્ટમ્સ પર્યાપ્ત રીતે સુરક્ષિત છે, વપરાશકર્તાઓને વિશ્વસનીય માહિતી પ્રદાન કરે છે, અને તેમના હેતુ લાભો પ્રાપ્ત કરવા માટે યોગ્ય રીતે સંચાલિત છે. તે ડેટા સાથે ચેડાં, ડેટા લોસ અથવા લીકેજ, સેવામાં વિક્ષેપ અને આઈટી સિસ્ટમના નબળા સંચાલનના જોખમને પણ ઘટાડે છે.

સિક્યોરિટી ઓડિટીંગ એન્ડ ટેસ્ટિંગ (SAT) સંસ્થાને આંતરિક કારણોસર સુરક્ષાની સ્થિતિ સમજવામાં મદદ કરે છે અને બાહ્ય પક્ષોને ખાતરી પૂરી પાડે છે, તેના પર ઉચ્ચતમ સ્તરનું ધ્યાન જરૂરી છે. તે હાલના સંરક્ષણોમાંના અંતરને ઓળખવામાં મદદ કરે છે.

ઓડિટ હેતુઓ સ્થાપિત કરવા ઓડિટનું આયોજન કર્યા પછી અને ઓડિટ કરવા માટે આગળ વધતા પહેલા, વ્યક્તિએ ઓડિટ હેતુઓ સ્થાપિત કરવા જોઈએ.

ઓડિટરે સમીક્ષા કરવી જોઈએ તે હેતુઓની સૂચિ નીચે મુજબ છે:

- કર્મચારીઓની પ્રક્રિયાઓ અને જવાબદારીઓ જેમાં સિસ્ટમ્સ અને કોસ ફંક્શનલ ટ્રેનિંગનો સમાવેશ થાય છે
- પરિવર્તન વ્યવસ્થાપન પ્રક્રિયાઓ ચાલુ છે અને IT અને મેનેજમેન્ટ કર્મચારીઓ દ્વારા અનુસરવામાં આવે છે. ચેન્જ મેનેજમેન્ટ એ સેવા પર કોઈપણ સંબંધિત ઘટનાઓની સંખ્યા અને અસરને ઘટાડવા માટે, IT ઈન્ફ્રાસ્ટ્રક્ચરને નિયંત્રિત કરવા માટેના તમામ ફેરફારોના કાર્યક્ષમ અને તાત્કાલિક સંચાલનનો સંદર્ભ આપે છે.
- ડાઉનટાઈમ ઘટાડવા અને મહત્વપૂર્ણ ડેટાના નુકશાનને રોકવા માટે યોગ્ય બેકઅપ પ્રક્રિયાઓ છે
- ડેટા સેન્ટરમાં ડેટા સેન્ટરની અનધિકૃત એક્સેસને રોકવા માટે પર્યાપ્ત ભૌતિક સુરક્ષા નિયંત્રણો છે
- સાધનો આગ અને પૂરથી સુરક્ષિત છે તેની ખાતરી કરવા માટે પર્યાપ્ત પર્યાવરણીય નિયંત્રણો છે.
- ઓડિટ આયોજન અને તૈયારી ઓડિટર કંપની અને તેના નિર્ણાયક વ્યવસાય વિશે પૂરતા પ્રમાણમાં શિક્ષિત હોવા જોઈએ ડેટા સેન્ટર સમીક્ષા હાથ ધરતા પહેલા પ્રવૃત્તિઓ. ઓડિટરે સમીક્ષા હાથ ધરતા પહેલા નીચેની બાબતો કરવી જોઈએ:
- ચિંતાના સંભવિત ક્ષેત્રો નક્કી કરવા માટે IT મેનેજમેન્ટ સાથે મળો
- વર્તમાન IT સંસ્થાના ચાર્ટની સમીક્ષા કરો
- ડેટા સેન્ટરના કર્મચારીઓના જોબ વર્ણનની સમીક્ષા કરો
- ડેટા સેન્ટરની અંદર કાર્યરત તમામ ઓપરેટિંગ સિસ્ટમ્સ, સોફ્ટવેર એપ્લિકેશન્સ અને ડેટા સેન્ટર સાધનોનું સંશોધન કરો
- કંપનીની IT નીતિઓ અને પ્રક્રિયાઓની સમીક્ષા કરો
- કંપનીના IT બજેટ અને સિસ્ટમ પ્લાનિંગ દસ્તાવેજીકરણનું મૂલ્યાંકન કરો
- ડેટા સેન્ટરના ડિઝાસ્ટર રિકવરી પ્લાનની સમીક્ષા કરો. ઓડિટ કરવું ત્યાં કોઈ પ્રમાણભૂત સુરક્ષા ઓડિટ પ્રક્રિયા નથી, પરંતુ ઓડિટર્સ સામાન્ય રીતે વ્યક્તિગત ઈન્ટરવ્યુ, નબળાઈ સ્કેન, OSની તપાસ અને સુરક્ષા એપ્લિકેશન સેટિંગ્સ અને નેટવર્ક વિશ્લેષણ, તેમજ ઈવેન્ટ લોગ્સ જેવા ઐતિહાસિક ડેટાનો અભ્યાસ કરીને તેમનું કાર્ય પૂર્ણ કરે છે. તેઓ શું આવરી લે છે, તેનો ઉપયોગ કેવી રીતે થાય છે અને તેઓ અસરકારક છે કે કેમ તે નક્કી કરવા માટે ઓડિટર્સ વ્યવસાયની સુરક્ષા નીતિઓ પર પણ ધ્યાન કેન્દ્રિત કરે છે. ચાલુ અને ભાવિ જોખમોનો સામનો કરવો. સામાન્ય રીતે, કમ્પ્યુટર સુરક્ષા ઓડિટ આના દ્વારા કરવામાં આવે છે:

- 1 ફેડરલ અથવા રાજ્ય નિયમનકારો.
- 2 કોર્પોરેટ આંતરિક ઓડિટર.

3 બાહ્ય ઓડિટર્સ - ટેકનોલોજી ઓડિટીંગ સાથે સંબંધિત ક્ષેત્રોમાં વિશિષ્ટ.

4 કન્સલ્ટન્ટ્સ - જ્યાં સંસ્થા પાસે વિશિષ્ટ કૌશલ્ય સમૂહનો અભાવ હોય ત્યાં ટેકનોલોજી ઓડિટનું આઉટસોર્સિંગ.

પ્રથમ, ઓડિટનો અવકાશ નક્કી થવો જોઈએ અને તેમાં કોમ્પ્યુટર સાધનો, ફોન, નેટવર્ક, ઈમેલ, ડેટા અને કાર્ડ્સ, ટોકન્સ અને પાસવર્ડ્સ જેવી કોઈપણ એક્સેસ-સંબંધિત વસ્તુઓ સહિત માહિતી સુરક્ષા સંબંધિત તમામ કંપનીની સંપત્તિઓનો સમાવેશ કરવો જોઈએ. પછી, ભૂતકાળ અને સંભવિત ભાવિ સંપત્તિના જોખમોની સમીક્ષા કરવી આવશ્યક છે. માહિતી સુરક્ષા ક્ષેત્રમાં કોઈપણ વ્યક્તિએ નવા વલણો તેમજ અન્ય કંપનીઓ દ્વારા લેવામાં આવતા સુરક્ષા પગલાંથી માહિતગાર રહેવું જોઈએ. આગળ, ઓડિટીંગ ટીમે વિનાશની માત્રાનો અંદાજ કાઢવો જોઈએ જે ભયજનક પરિસ્થિતિઓમાં થઈ શકે છે. ખતરો આવી ગયા પછી વ્યવસાયિક કામગીરી જાળવવા માટે એક સ્થાપિત યોજના અને નિયંત્રણો હોવા જોઈએ, જેને ઘૂસણખોરી નિવારણ સિસ્ટમ કહેવામાં આવે છે.

સમીક્ષા કરવી આગળનું પગલું ડેટા સેન્ટર ઓડિટ ઉદ્દેશ્યોને સંતોષવા પુરાવા એકત્રિત કરવાનું છે. આમાં ડેટા સેન્ટરના સ્થાન પર મુસાફરી કરવી અને ડેટા સેન્ટરમાં કરવામાં આવતી પ્રક્રિયાઓ અને પ્રક્રિયાઓનું નિરીક્ષણ કરવું શામેલ છે. સંતોષવા માટે નીચેની સમીક્ષા પ્રક્રિયાઓ હાથ ધરવી જોઈએ પૂર્વનિર્ધારિત ઓડિટ ઉદ્દેશ્યો:

- ઓડિટરે તેમના ઉદ્દેશ્યોને સંતોષવા માટે ડેટા સેન્ટરના કર્મચારીઓનું અવલોકન કરવું જોઈએ અને તેમની મુલાકાત લેવી જોઈએ.
- ઓડિટર એ ચકાસવું જોઈએ કે તમામ ડેટા સેન્ટર સાધનો યોગ્ય રીતે અને અસરકારક રીતે કામ કરી રહ્યા છે.
- તમામ ડેટા સેન્ટર નીતિઓ અને પ્રક્રિયાઓ દસ્તાવેજીકૃત અને ડેટા સેન્ટર પર સ્થિત હોવી જોઈએ. મહત્વપૂર્ણ દસ્તાવેજી પ્રક્રિયાઓમાં શામેલ છે: ડેટા સેન્ટર કર્મચારીઓની નોકરીની જવાબદારીઓ, બેકઅપ નીતિઓ, સુરક્ષા નીતિઓ, કર્મચારીની સમાપ્તિ નીતિઓ, સિસ્ટમ ઓપરેટિંગ પ્રક્રિયાઓ અને ઓપરેટિંગ સિસ્ટમ્સની ઝાંખી.
- ઓડિટરે ભૌતિક સુરક્ષા નિયંત્રણોના સંદર્ભમાં ક્લાયન્ટના ડેટા સેન્ટરની સુરક્ષાનું મૂલ્યાંકન કરવું જોઈએ અને ડેટા સેન્ટર સાધનોની સુરક્ષાની ખાતરી કરવા માટે પર્યાવરણીય નિયંત્રણો અમલમાં હોવા જોઈએ. આમાં શામેલ છે:
- એર કન્ડીશનીંગ એકમો, ઊંચા માળ, હ્યુમિડિફાયર અને અવિરત વીજ પુરવઠો.
- બેકઅપ પ્રક્રિયાઓ - ઓડિટરે ચકાસવું જોઈએ કે સિસ્ટમની નિષ્ફળતાના કિસ્સામાં ક્લાયન્ટ પાસે બેકઅપ પ્રક્રિયાઓ છે. ગ્રાહકો બેકઅપ જાળવી શકે છે એક અલગ સ્થાન પર ડેટા સેન્ટર કે જે તેમને સિસ્ટમની નિષ્ફળતાના કિસ્સામાં તાત્કાલિક કામગીરી ચાલુ રાખવા માટે પરવાનગી આપે છે.

પેનિટ્રેશન ટેસ્ટ, નબળાઈનું મૂલ્યાંકન અને સુરક્ષા ઓડિટ બાહ્ય હુમલાખોરો ઘણીવાર સિસ્ટમ અને નેટવર્કમાં ઘૂસણખોરી કરવા માટે પહેલેથી જ જાણીતી નબળાઈઓ અને શોષણનો ઉપયોગ કરે છે. યોગ્ય રક્ષણાત્મક પગલાં લેવાથી અને પર્યાપ્ત સુરક્ષા ડિઝાઇન આ સમસ્યાને ઓછી કરી શકે છે. આ પહેલેથી જ અસ્તિત્વમાં રહેલી ખુલ્લી સિસ્ટમો અને નિયમિત અંતરાલ પર તેમના જોખમોને ઓળખીને પણ સારી હદ સુધી પ્રાપ્ત કરી શકાય છે. અગાઉના તબક્કામાં જોખમો શોધી શકાય છે અને અગાઉના તબક્કે યોગ્ય પગલાં લઈ શકાય છે.

નબળાઈ આકારણી એ તમામ સંભવિત નબળાઈઓને ઓળખવા માટે વપરાતી પ્રથા છે જેનો પર્યાવરણમાં શોષણ થઈ શકે છે. આકારણીનો ઉપયોગ ભૌતિક સુરક્ષા, કર્મચારીઓ અથવા સિસ્ટમ અને નેટવર્ક સુરક્ષાનું મૂલ્યાંકન કરવા માટે થઈ શકે છે. તેમને ઓળખવા માટે નબળાઈ ઓળખ સાધનોનો ઉપયોગ થઈ શકે છે. IP સરનામા દ્વારા દરેક કમ્પ્યુટર સિસ્ટમની સૂચિ અને તેમની સંલગ્ન નબળાઈઓ અને નબળાઈઓને કેવી રીતે “ફિક્સ” કરવી તે અંગેના પગલાં પછી જનરેટ કરવા જોઈએ.

પેનિટ્રેશન ટેસ્ટ જાણીતી નબળાઈઓનો ઉપયોગ કરીને “નેટવર્કમાં પ્રવેશ કરવા”ના હેતુ સાથે હાથ ધરવામાં આવે છે. અહીંથી, ઉદ્દેશ્ય નેટવર્કની સૌથી મહત્વપૂર્ણ સિસ્ટમ પર એડમિનિસ્ટ્રેટર અથવા રૂટ એક્સેસ મેળવવાનો છે. આ સિસ્ટમો અને સિસ્ટમો પરના ડેટા સાથે ચેડાં કરવા અથવા સંશોધિત કરવા માટે નેટવર્કને સંપૂર્ણ એક્સેસ આપે છે. વાસ્તવિક હેકર શું કરશે તેનું અનુકરણ કરવા માટે પેનિટ્રેશન ટેસ્ટ હાથ ધરવામાં આવે છે અને તે કંપનીને સાબિત કરે છે કે સંસ્થામાં ખરેખર ઘૂસી શકાય છે. પેનિટ્રેશન ટેસ્ટિંગને એથિકલ હેકિંગ તરીકે પણ ઓળખવામાં આવે છે. મોટાભાગના કિસ્સાઓમાં, સુરક્ષા વ્યાવસાયિક નબળાઈ સ્કેનરના અહેવાલો જોઈ શકે છે અને કંપની જે જોખમનો સામનો કરી રહી છે તેના સ્તરને સમજી શકે છે. ઓડિટ કંટ્રોલ્સ એપ્લિકેશન કંટ્રોલ્સની સમીક્ષા - તે તૈનાત ટેકનોલોજીના જોખમોની ઓળખ અને કંપનીના આવા જોખમોના સંપર્કમાં ઘટાડો, તેની ખાતરી કરીને જરૂરી નિયંત્રણો અને સુરક્ષા સ્થાને છે. સામાન્ય કમ્પ્યુટર નિયંત્રણોની સમીક્ષા - તે કંપનીની અંદર વિવિધ પ્લેટફોર્મ્સ પર ચાલતી એપ્લિકેશન સિસ્ટમ્સ માટે સુરક્ષિત અને સ્થિર વાતાવરણ પ્રદાન કરવા માટે કરવામાં આવે છે. જો ઓપરેશનલ સિસ્ટમના ઉદ્દેશ્યો નિયંત્રણોમાં ભૂલો જોવા મળે તો સિસ્ટમ ખોવાઈ શકે છે

- કોમ્પ્યુટરમાં દાખલ થતો ડેટા સાચો છે તેની ખાતરી કરવા
- ડેટાને કોમ્પ્યુટરમાં ઈનપુટ કરતા પહેલા તેની કારકુની હેન્ડલિંગ તપાસો
- ખરાબ ડેટા અથવા ખરાબ પ્રોગ્રામને કારણે થતી ભૂલોને શોધવા અને ટ્રેસ કરવાના માધ્યમો પ્રદાન કરો
- ખાતરી કરો કે કાનૂની જરૂરિયાતો પૂરી થઈ છે
- છેતરપિંડી સામે રક્ષણ મેળવવા એક્સેસ કંટ્રોલ અને ઓડિટીંગ ભૌતિક અને તાર્કિક સુરક્ષા ભૌતિક અને તાર્કિક સુરક્ષાના અનેક પ્રકારો છે. ભૌતિક સુરક્ષામાં તાળાઓ અથવા બાયોમેટ્રિક્સ જેવી વસ્તુઓનો સમાવેશ થાય છે. તાર્કિક સુરક્ષા ઉદાહરણોમાં એક્સેસ કંટ્રોલ અને ઓડિટીંગ, યુઝર એકાઉન્ટ મેનેજમેન્ટ, ઉલ્લંઘન અને સુરક્ષા પ્રવૃત્તિ અહેવાલો અને

ફાયરવોલ એક્સેસ કંટ્રોલ એ એક સિસ્ટમ છે જે અધિકારીઓને વિસ્તારો અને સંસાધનોની એક્સેસને નિયંત્રિત કરવા સક્ષમ કરે છે આપેલ ભૌતિક સુવિધા અથવા કમ્પ્યુટર-આધારિત માહિતી સિસ્ટમમાં. કમ્પ્યુટર માટેનો પાસવર્ડ અથવા ATM સિસ્ટમનો PIN એ એક્સેસ કંટ્રોલના સ્વરૂપ છે. જ્યારે વ્યક્તિઓ ગોપનીય, મહત્વપૂર્ણ અથવા સંવેદનશીલ માહિતી અને સાધનસામગ્રીને સુરક્ષિત કરવા માંગતા હોય ત્યારે એક્સેસ કંટ્રોલ મિકેનિઝમનો ઉપયોગ કરવો મહત્વપૂર્ણ છે.

એક્સેસ કંટ્રોલ સિસ્ટમનું ઓડિટ કરવું એ સિસ્ટમમાં પ્રવેશ અથવા પ્રયાસ કરેલ પ્રવેશની ઘટનાને ટ્રેક કરવાનો એક માર્ગ છે. આ અગત્યનું છે કારણ કે તે દર્શાવે છે કે એક્સેસ કંટ્રોલ સિસ્ટમ કેટલી સફળ છે, તેમજ કોને પ્રવેશ નકારવામાં આવ્યો હતો અને જો તેઓ એક કરતા વધુ વખત પ્રવેશ કરવાનો પ્રયાસ કરે છે, તો તેમનો ઈરાદો શું છે?

લોજિકલ ઓડિટ નીચેની તપાસ કરવા માટે કરવામાં આવે છે

- એક્સેસ ઉલ્લંઘનની શક્તિ અને નબળાઈઓ
- સુરક્ષા પ્રવૃત્તિ અહેવાલો
- લોગિંગ પ્રવૃત્તિ અહેવાલો
- ફાયરવોલની કાર્યક્ષમતા
- સુરક્ષાના ઉલ્લંઘન અંગેના અહેવાલો
- અનધિકૃત વ્યક્તિઓ દ્વારા સિસ્ટમને હેક કરવાના પ્રયાસોના અહેવાલો વગેરે.

ઓડિટર માટે વ્યવસાયિક નીતિશાસ્ત્ર: ઉદ્દેશ્ય ઓડિટમાં વિશ્વાસ મેળવવા માટે, વ્યવસાયિક નીતિશાસ્ત્રના સમૂહને જાળવી રાખવું જરૂરી છે. વ્યવસાયિક નૈતિકતા વ્યક્તિગત વ્યક્તિઓ દ્વારા તેમજ માહિતી સુરક્ષા ઓડિટીંગના ક્ષેત્રમાં સેવાઓ પ્રદાન કરતી કંપનીઓ દ્વારા જાળવવી આવશ્યક છે. વ્યવસાયિક નીતિશાસ્ત્રમાં નીચેના સિદ્ધાંતો સામેલ છે:

પ્રમાણિકતા અને ગોપનીયતા: પ્રમાણિકતા વિશ્વાસનો પાયો છે અને તે બનાવે છે આકારણીની વિશ્વસનીયતા માટેનો આધાર. સંવેદનશીલ વ્યવસાયિક પ્રક્રિયાઓ અને માહિતી મોટાભાગે માહિતી સુરક્ષા પર આધારિત હોવાનું જાણવા મળે છે, ઓડિટ દરમિયાન પ્રાપ્ત માહિતીની ગોપનીયતા અને IS ઓડિટના પરિણામો અને તારણોનું સમજદારીપૂર્વક સંચાલન એ આવા કાર્ય માટે મહત્વપૂર્ણ આધાર છે. IS ઓડિટર્સ તેઓ મેળવેલી માહિતીના મૂલ્યથી વાકેફ છે અને તેની માલિકી કોની છે, અને તેઓ આ માહિતીને અનુરૂપ પરવાનગી વિના જાહેર કરશે નહીં સિવાય કે તેઓ કાયદેસર અથવા વ્યવસાયિક રીતે આવું કરવા માટે જરૂરી હોય. નિષ્ણાત જ્ઞાન: IS ઓડિટર્સ ફક્ત તે જ નોકરીઓ સ્વીકારે છે જેના માટે તેમની પાસે જરૂરી જ્ઞાન અને કૌશલ્યો તેમજ અનુરૂપ અનુભવ હોય છે અને તેઓ તેમના કાર્યને પૂર્ણ કરતી વખતે તેનો ઉપયોગ કરે છે. તેઓ સતત તેમના જ્ઞાન તેમજ તેમના કાર્યની અસરકારકતા અને ગુણવત્તામાં સુધારો કરે છે.

ઉદ્દેશ્ય અને સંપૂર્ણતા: એક IS ઓડિટરને એકત્રિત કરતી વખતે, મૂલ્યાંકન કરતી વખતે, અને ઓડિટ કરાયેલ પ્રવૃત્તિઓ અથવા વ્યવસાય પ્રક્રિયાઓ વિશેની માહિતી પસાર કરવી. તમામ સંબંધિત સંજોગોનું મૂલ્યાંકન નિષ્પક્ષપણે થવું જોઈએ અને તે ઓડિટરના પોતાના હિત અથવા અન્યના હિતથી પ્રભાવિત ન હોઈ શકે. ઉદ્દેશ્ય પ્રસ્તુતિ: IS ઓડિટરની ફરજ છે કે તે પરીક્ષાના પરિણામોની ચોકસાઈ અને સત્યતાપૂર્વક તેના કલાયન્ટને જાણ કરે. આમાં IS ઓડિટ અહેવાલોમાં તથ્યોની નિષ્પક્ષ અને સમજી શકાય તેવી રજૂઆત, નિર્ધારિત તથ્યોનું રચનાત્મક મૂલ્યાંકન અને સલામતી અને પ્રક્રિયાઓને સુધારવા માટેની વિશિષ્ટ ભલામણોનો સમાવેશ થાય છે.

ચકાસણી અને પુનઃઉત્પાદનક્ષમતા: વિશ્વસનીય અને સમજી શકાય તેવા તારણો અને પરિણામો માટેનો તર્કસંગત આધાર એ વાસ્તવિક હકીકતોનું સ્પષ્ટ અને સુસંગત દસ્તાવેજીકરણ છે. આમાં એ પણ સામેલ છે કે IS ઓડિટ ટીમ તેના નિષ્કર્ષ પર આવવા માટે દસ્તાવેજીકૃત અને પુનઃઉત્પાદનક્ષમ પદ્ધતિને અનુસરે છે.

અનુપાલન ઓડિટ: અનુપાલન ઓડિટ એ સંસ્થાના નિયમનકારી પાલનની વ્યાપક સમીક્ષા છે માર્ગદર્શિકા સ્વતંત્ર એકાઉન્ટિંગ, સુરક્ષા અથવા IT સલાહકારો પાલન તૈયારીઓની મજબૂતાઈ અને સંપૂર્ણતાનું મૂલ્યાંકન કરે છે. ઓડિટર્સ પાલન ઓડિટ દરમિયાન સુરક્ષા નીતિઓ, વપરાશકર્તા એક્સેસ નિયંત્રણો અને જોખમ સંચાલન પ્રક્રિયાઓની સમીક્ષા કરે છે.

અનુપાલન ઓડિટમાં શું તપાસવામાં આવે છે તે સંસ્થા સાર્વજનિક છે કે ખાનગી કંપની છે, તે કેવા પ્રકારનો ડેટા હેન્ડલ કરે છે અને જો તે સંવેદનશીલ નાણાકીય ડેટા ટ્રાન્સમિટ કરે છે અથવા સ્ટોર કરે છે તેના આધારે બદલાશે.

માહિતી સુરક્ષા નીતિઓ: સંસ્થાઓ માહિતી સુરક્ષા નીતિઓના વિકાસને વધુ પ્રાધાન્ય આપી રહી છે, કારણ કે તેમની સંપત્તિનું રક્ષણ એ અગ્રણી બાબતોમાંની એક છે જેને ધ્યાનમાં લેવાની જરૂર છે. માહિતી સુરક્ષા નીતિઓમાં સ્પષ્ટતાનો અભાવ ગંભીર નુકસાન તરફ દોરી શકે છે જે પુનઃપ્રાપ્ત કરી શકાતો નથી. તેથી એક સંસ્થા સુરક્ષા નીતિનો સફળતાપૂર્વક અમલ કરવા માટે વિવિધ વ્યૂહરચના બનાવે છે. માહિતી સુરક્ષા નીતિ સમગ્ર સંસ્થામાં માહિતી સુરક્ષા માટે વ્યવસ્થાપન દિશા અને સમર્થન પ્રદાન કરે છે. માહિતી સંસ્થામાં ગોપનીયતા, અખંડિતતા અને ઉપલબ્ધતાના ભંગના પરિણામો સામે યોગ્ય રીતે સુરક્ષિત હોવું જરૂરી છે. અનધિકૃત ફેરફારો, કાઢી નાખવા અને જાહેરાતોથી માહિતીને નિયંત્રિત અને સુરક્ષિત કરવા માટે યોગ્ય સુરક્ષા પગલાં અમલમાં મૂકવાની જરૂર છે. સુરક્ષા પગલાંનું સ્તર શોધવા માટે કે જેને લાગુ કરવાની જરૂર છે, જોખમનું મૂલ્યાંકન ફરજિયાત છે.

સુરક્ષા નીતિઓનો હેતુ માહિતી પ્રણાલીઓના સંદર્ભમાં સંસ્થામાં કર્મચારીઓ પાસેથી શું અપેક્ષા રાખવામાં આવે છે તે વ્યાખ્યાયિત કરવાનો છે. ઉદ્દેશ્ય માહિતી સંપત્તિઓ માટેના જોખમને ઘટાડવા માટે સિસ્ટમોના ઉપયોગને માર્ગદર્શન આપવા અથવા નિયંત્રિત કરવાનો છે.

તે માહિતી પ્રણાલીઓ સાથે કામ કરતા કર્મચારીઓને સ્વીકાર્ય ઉપયોગની નીતિ પણ આપે છે, જે સમજાવે છે કે શું માન્ય છે અને શું નથી. તમામ કંપનીઓની સુરક્ષા નીતિઓ એકસરખી હોતી નથી, પરંતુ તેમની પાછળનો મુખ્ય ઉદ્દેશ એસેટ્સનું રક્ષણ કરવાનો છે. સુરક્ષા નીતિઓ ચોક્કસ ધ્યેયો સાથે ડિઝાઇન કરવામાં આવી છે. માહિતી સુરક્ષા ઓડિટ સાધનો: માહિતી સુરક્ષા ઓડિટ સાધનોમાં ઉપયોગિતાઓ અને પાવર ટૂલ્સનો સમાવેશ થાય છે, બંને ખુલ્લા છે સ્ત્રોત અને વ્યાપારી.

યુટિલિટી ટૂલ્સ : આ એકલ-હેતુના સાધનો છે જે ઓપરેટિંગ સિસ્ટમના મૂળ હોઈ શકે છે અથવા મુક્તપણે ઉપલબ્ધ હોઈ શકે છે.

યુટિલિટી ટૂલ્સને મેન્યુઅલ અભિગમની જરૂર હોય છે, જો કે તે ઘણીવાર કસ્ટમાઈઝ્ડ સ્ક્રિપ્ટ્સમાં સમાવિષ્ટ હોય છે--અથવા વ્યવસાયિક ઉત્પાદનો પણ. તમે સ્થાનિક ઉપયોગિતાઓનો પણ સમાવેશ કરી શકો છો, જેમ કે મોટાભાગના પ્લેટફોર્મ્સ પર ઉપલબ્ધ પિંગ, જેનો ઉપયોગ નેટવર્ક લક્ષ્ય ICMP પેકેટોને પ્રતિસાદ આપે છે કે કેમ તે નિર્ધારિત કરવા માટે થાય છે.

ગુણ: ઉપયોગિતા સાધનો મુક્તપણે ઉપલબ્ધ છે અને ચોક્કસ કાર્ય માટે ચુસ્તપણે કેન્દ્રિત છે, તેમને વધુ કાર્યક્ષમ બનાવે છે. તેઓ મેન્યુઅલી જોવા મળતી નબળાઈઓ કરતાં ઘણી ઝડપથી નબળાઈઓ શોધવામાં મદદ કરે છે.

વિપક્ષ: તેનો ઉપયોગ કરવા માટે કૌશલ્યની જરૂર છે. મોટા ઓડિટ માટે, મેન્યુઅલ પરીક્ષણ સમય માંગી લેતું હોય છે અને ઓડિટરની કુશળતાના આધારે અસંગત પરિણામો લાવી શકે છે.

ટ્રેસ રૂટ : નેટવર્ક ટ્રેસિંગ યુટિલિટીનો ઉપયોગ યજમાનને નેટવર્ક નક્કી કરવા માટે થાય છે.

nslookup: ડોમેન માલિકી નક્કી કરવા માટે વપરાય છે.

અને ઓપન સોર્સ સ્ક્રિપ્ટો, જેમાં નીચેનાનો સમાવેશ થાય છે:

Nmap: ફ્રી પોર્ટ સ્કેનિંગ યુટિલિટી.

ક્રેક: પાસવર્ડને તોડવાનો પ્રયાસ કરીને નબળા છે કે કેમ તે નિર્ધારિત કરવા માટે લોકપ્રિય પાસવર્ડ-ક્રેકિંગ ટૂલનો ઉપયોગ થાય છે.

જહોન ઘ રિપર: પાસવર્ડ ક્રેકિંગ ટૂલનો ઉપયોગ મુખ્યત્વે યુનિક્સ પાસવર્ડ્સ શોધવા માટે થાય છે.

binfo.c : એક BIND સંસ્કરણ તપાસનાર, binfo એ રિમોટ નેમ સર્વર પર ચાલતા નામના સંસ્કરણને પાછું ખેંચવા માટે એક ઝડપી નાની સ્ક્રિપ્ટ છે.

ghbba.c: આપેલ વર્ગ B અથવા C સબનેટના તમામ મશીન નામો અને IP સરનામાઓ કાઢવા માટે એક સરળ સાધન.

પાવર ટૂલ્સ પાવર ટૂલ્સ એ ઓડિટ પ્રક્રિયાના ભાગોને સુવ્યવસ્થિત અને સ્વચાલિત કરવા માટે મલ્ટી ફંક્શન બંડલ યુટિલિટીઝ છે. જ્યારે કેટલાક ઓપન-સોર્સ પેકેજો છે, ઘણા કસ્ટમ નબળાઈ ડેટાબેસેસ સાથેના વ્યાવસાયિક ઉત્પાદનો છે.

ગુણ: ઓટોમેટેડ ટૂલ્સ ડેટાબેઝ સામે નબળાઈઓ માટે સ્કેન કરે છે. ચેતવણીઓને હેલ્પ ડેસ્ક મોનિટરિંગ ટૂલ્સ સાથે જોડી શકાય છે. કેટલાક કિસ્સાઓમાં, સ્કેનિંગ સાધનને ફાયરવોલ અથવા ઇન્ટ્રુઝન ડિટેક્શન મેનેજમેન્ટ સ્ટેશન સાથે સંકલિત કરી શકાય છે.

કેટલાક વ્યાપારી સ્કેનર્સ એક્સપોઝર અને સંકળાયેલા જોખમોની વિગતો આપતા ઉત્તમ અહેવાલો ઉત્પન્ન કરે છે.

વિપક્ષ: સ્કેનર્સ માત્ર તેમના ડેટાબેઝમાં નબળાઈઓ માટે તપાસ કરે છે, જે વર્તમાન હોવી જોઈએ. ઘણા સ્કેનર્સનું વેચાણ કરવામાં આવતી નબળાઈ તપાસની સંખ્યા પર કરવામાં આવે છે.

આ હંમેશા સાધનની અસરકારકતાનો સારો સંકેત નથી. ઘણીવાર, નબળાઈઓનું ખોટું નિદાન થાય છે. સ્કેનર જોખમનું ચોક્કસ મૂલ્યાંકન કરી શકતું નથી.

કેટલાક ઓપન-સોર્સ પાવર ટૂલ્સ છે નેસસ, SARA(ધ સિક્યુરિટી ઓડિટરનું રિસર્ચ આસિસ્ટન્ટ), વ્હિસ્કર, વગેરે.

આજે ઉપલબ્ધ કેટલાક વ્યવસાયિક સ્કેનર્સમાં ઇન્ટરનેટ સિક્યુરિટી સિસ્ટમનું ઇન્ટરનેટ સ્કેનર, eEye ડિજિટલ સિક્યુરિટીનું રેટિના, BindViewનું BV કંટ્રોલ, CORE સિક્યુરિટી ટેકનોલોજીનું ઓડિટિંગ ટૂલ્સ સ્યુટ અને ફાઉન્ડસ્ટોનનું ફાઉન્ડ સ્કેન છે.

Google

એક વાસ્તવિક હેકર બોક્સની બહાર વિચારે છે અને સાધનોનો એ રીતે ઉપયોગ કરવાનું શીખે છે જે કદાચ તેનો હેતુ ન હોય. જ્યારે ગૂગલ સર્ચ એન્જિન નથી, કડક શબ્દોમાં કહીએ તો, ઓડિટિંગ ટૂલ, તે સાઈટ વિશેની માહિતી ભેગી કરવા માટે ઉત્તમ છે. ઉદાહરણ તરીકે, “@DGET.com” દાખલ કરવાનો પ્રયાસ કરો (જ્યાં “DGET” તમારું ડોમેન છે). કેટલીકવાર, આનાથી કેટલાક સારા ડેટા મળી શકે છે, જેમ કે સિસ્ટમ એડમિનિસ્ટ્રેટર તેની સાઈટ વિશે તકનીકી વિગતો પોસ્ટ કરે છે, જેમાં તેના એકાઉન્ટનું નામ સહેલાઈથી હોય છે. ગૂગલ સ્ટેરોઈડ્સ પર યુનિક્સ “ગ્રેપ” આદેશ જેવું છે.

સંદેશાવ્યવહાર પરિણામો જે વ્યક્તિઓની સંપત્તિઓનું ઓડિટ કરવામાં આવી રહ્યું છે.

ઓડિટને સતત ઓડિટ પ્રવૃત્તિના પરિણામોની જાણ કરતી વખતે, એક્સચેન્જ માટે સ્વતંત્ર અને સુસંગત હોવું મહત્વપૂર્ણ છે.

વ્યાખ્યાયિત માપદંડો પર વરિષ્ઠ મેનેજમેન્ટને જાણ કરવી મેનેજમેન્ટને એક લાક્ષણિક ઓડિટ રિપોર્ટ અને મેનેજમેન્ટનો પ્રતિસાદ નીચે કોષ્ટક 1 માં દર્શાવેલ એક જેવો હોઈ શકે છે, પરંતુ અહેવાલો અને પ્રતિસાદના અન્ય ઘણા ફોર્મેટ ઉપયોગમાં છે.

સંદર્ભ	વિષય	વિષય	વિષય	વિષય	વિષય
1	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
2	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
3	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
4	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
5	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
6	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
7	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
8	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
9	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા
10	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા	સામાન્ય સુધારા

નેટવર્ક્સને સુરક્ષિત કરો (Privacy Protection and IT Act)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- માહિતીની ગોપનીયતાનું વર્ણન કરો
- માહિતી પ્રણાલીઓમાં ગોપનીયતાનું રક્ષણ કરવાની પદ્ધતિનું વર્ણન કરો
- શ્રેષ્ઠ ઓનલાઈન ગોપનીયતા પ્રથાઓનું વર્ણન કરો
- આઈટી એક્ટ શું છે તે વિશે સમજાવો
- સાયબર ગુનાઓ વિશે વર્ણન કરો.

પરિચય ગોપનીયતા એ વ્યક્તિ અથવા જૂથની તેમની સિસ્ટમો, ડેટા અથવા માહિતીને અલગ કરવાની અને તેને પસંદગીપૂર્વક શેર કરવાની ક્ષમતા છે. જ્યારે કોઈ વ્યક્તિ માટે કંઈક ખાનગી હોય છે, ત્યારે તેનો સામાન્ય રીતે અર્થ થાય છે કે તે કંઈક વિશેષ અથવા સંવેદનશીલ છે.

માહિતી અથવા ડેટા ગોપનીયતા એ ટેકનોલોજી અને પોતાના વિશેના ડેટાના સંગ્રહ અને શેરિંગમાં ગોપનીયતાના કાયદાકીય અધિકાર અથવા જાહેર અપેક્ષા વચ્ચેના વિકસતા સંબંધોનો સંદર્ભ આપે છે. જ્યાં પણ વ્યક્તિ અથવા વ્યક્તિઓ સંબંધિત અનન્ય રીતે ઓળખી શકાય તેવા ડેટાને ડિજિટલ સ્વરૂપમાં અથવા અન્યથા એકત્ર અને સંગ્રહિત કરવામાં આવે છે ત્યાં ગોપનીયતાની ચિંતાઓ અસ્તિત્વમાં છે. કેટલાક કિસ્સાઓમાં આ ચિંતાઓ ડેટા કેવી રીતે એકત્રિત, સંગ્રહિત અને સંકળાયેલ છે તેનો સંદર્ભ આપે છે. અન્ય કિસ્સાઓમાં મુદ્દો એ છે કે માહિતીની ઍક્સેસ કોને આપવામાં આવે છે. અન્ય સમસ્યાઓનો સમાવેશ થાય છે કે શું વ્યક્તિ પાસે તેમના વિશેના ડેટા પર કોઈ માલિકી હકો છે કે નહીં, અને/અથવા તે માહિતીને જોવાનો, ચકાસવાનો અને પડકારવાનો અધિકાર છે.

માહિતી સુરક્ષા અનધિકૃત વ્યક્તિઓ અથવા સિસ્ટમોને પ્રતિબંધિત માહિતીની ઍક્સેસ મેળવવાથી રોકે છે.

ગોપનીયતા છે નિયમો અને જવાબદારીઓનો સંગ્રહ જે નક્કી કરે છે કે કોઈપણ માધ્યમમાં ઍક્સેસ કેવી રીતે અને ક્યારે અધિકૃત કરવામાં આવશે. તે અનુસરે છે કે સારી સુરક્ષા અને ગોપનીયતા પ્રથાઓ એકબીજા પર આધાર રાખે છે. ગોપનીયતાનું ડોમેન આંશિક રીતે સુરક્ષાને ઓવરલેપ કરે છે, ઉદાહરણ તરીકે યોગ્ય ઉપયોગની વિભાવનાઓ, તેમજ માહિતીની સુરક્ષા.

કોમ્પ્યુટર સુરક્ષા અને ગોપનીયતા વચ્ચેનો સંબંધ એ હકીકતમાં રહેલો છે કે પર્યાપ્ત કોમ્પ્યુટર સુરક્ષા, અથવા તેનો અભાવ, તે ગોપનીયતાના સ્તરનું નિર્ણાયક છે જેની કોમ્પ્યુટર વપરાશકર્તા અપેક્ષા રાખી શકે છે. લોકો ઘણા કાર્યો કરવા માટે કમ્પ્યુટરનો ઉપયોગ કરે છે, જેમાં વ્યવસાય, બેંકિંગ, સામાજિકકરણ અને ખાનગી માહિતીનો સંગ્રહ શામેલ છે. જો કોમ્પ્યુટર સુરક્ષાનો ભંગ થાય છે, તો આ પ્રકારનાં કાર્યો હાથ ધરવામાં આવે છે તેના પર તેની નકારાત્મક અસર પડશે. ઈ-કોમર્સના ક્ષેત્રમાં, કોમ્પ્યુટર સુરક્ષા અને ગોપનીયતાનો મુદ્દો વેપારી પક્ષો વચ્ચે વિશ્વાસનું સ્તર નક્કી કરશે. જો બંને બાજુએ સુરક્ષાના ભંગની કોઈ શંકા હોય, તો આ તરફ દોરી જશે વિશ્વાસનો વિનાશ અને વ્યવસાયિક સંબંધોનો અંત. આમાં તૃતીય પક્ષોના જોખમો અને ધમકીઓનો સમાવેશ થાય છે જે વ્યવસાય ભાગીદારો સાથે પણ સંબંધિત નથી. અયોગ્ય અથવા અવિદ્યમાન ડિસ્ક્લોઝર નિયંત્રણ ગોપનીયતા સમસ્યાઓનું મૂળ કારણ હોઈ શકે છે. ડેટાની ગોપનીયતા સમસ્યાઓ સ્ત્રોતોની વિશાળ શ્રેણીમાંથી માહિતીના પ્રતિભાવમાં ઊભી થઈ શકે છે, જેમ કે:

- આરોગ્ય સંભાળ રેકૉર્ડ્સ
- તપાસ અને કાર્યવાહી
- નાણાકીય સંસ્થાઓ અને વ્યવહારો
- રહેઠાણ અને ભૌગોલિક રેકૉર્ડ
- સંરક્ષણ ડેટા
- ગોપનીયતા ભંગ
- સ્થાન-આધારિત સેવા અને ભૌગોલિક સ્થાન
- વૈજ્ઞાનિક સંશોધન વગેરે.

ડેટા ગોપનીયતામાં પડકાર એ છે કે વ્યક્તિગત રીતે ઓળખી શકાય તેવી માહિતીનું રક્ષણ કરતી વખતે ડેટા શેર કરવો. ડેટા સુરક્ષા અને માહિતી સુરક્ષાના ક્ષેત્રો આ મુદ્દાને ઉકેલવા માટે સોફ્ટવેર, હાર્ડવેર અને માનવ સંસાધનોની ડિઝાઈન અને ઉપયોગ કરે છે. ડેટા પ્રોટેક્શનને લગતા કાયદા અને નિયમો સતત બદલાતા રહેતા હોવાથી, કાયદામાં થતા કોઈપણ ફેરફારોથી વાકેફ રહેવું અને ડેટા ગોપનીયતા અને સુરક્ષા નિયમો સાથેના તમારા પાલનનું સતત પુનઃમૂલ્યાંકન કરવું મહત્વપૂર્ણ છે.

માહિતી પ્રણાલીઓમાં ગોપનીયતાનું રક્ષણ: વિવિધ ગોપનીયતા નિયમો સાથેની વિવિધ માહિતી પ્રણાલીઓ એકબીજા સાથે જોડાયેલી હોવાથી અને માહિતી શેર કરવામાં આવતી હોવાથી, નીતિ ઉપકરણોને ગોપનીયતા નીતિ નિયમો (અને કાયદા) ની વધતી જતી માત્રામાં સમાધાન, અમલ અને દેખરેખ રાખવાની જરૂર પડશે. બે શ્રેણીઓ છે. ના કોમર્શિયલ આઈટી સિસ્ટમ્સમાં ગોપનીયતા સુરક્ષાને સંબોધવા માટેની તકનીક: સંચાર અને અમલીકરણ. પોલિસી કોમ્પ્યુનિકેશન P3P : આ ગોપનીયતા પસંદગીઓ માટેનું પ્લેટફોર્મ છે. P3P એ ગોપનીયતા પ્રથાઓને સંચાર કરવા અને વ્યક્તિઓની પસંદગીઓ સાથે તેમની સરખામણી કરવા માટેનું એક માનક છે.

નીતિ અમલીકરણ

- XACML - તેની ગોપનીયતા પ્રોફાઈલ સાથે એક્સટેન્સિબલ એક્સેસ કંટ્રોલ માર્કઅપ લેંગ્વેજ એ મશીન-વાંચી શકાય તેવી ભાષામાં ગોપનીયતા નીતિઓ વ્યક્ત કરવા માટેનું એક માનક છે જેનો ઉપયોગ સોફ્ટવેર સિસ્ટમ એન્ટરપ્રાઇઝ IT સિસ્ટમ્સમાં નીતિને લાગુ કરવા માટે કરી શકે છે.
- EPAL - એન્ટરપ્રાઇઝ પ્રાઈવસી ઓથોરાઈઝેશન લેંગ્વેજ XACML જેવી જ છે, પરંતુ હજુ સુધી તે પ્રમાણભૂત નથી.
- WS-ગોપનીયતા - “વેબ સેવા ગોપનીયતા” વેબ સેવાઓમાં ગોપનીયતા નીતિના સંચાર માટે સ્પષ્ટીકરણ હશે. ઉદાહરણ તરીકે, તે સ્પષ્ટ કરી

શકે છે કે કેવી રીતે ગોપનીયતા નીતિની માહિતી વેબ સેવા સંદેશના SOAP એન્વલપમાં એમેસડ કરી શકાય છે.

ઈન્ટરનેટ પર ગોપનીયતાનું રક્ષણ ઈન્ટરનેટ પર તમે લગભગ હંમેશા તમારા વિશે ઘણી બધી માહિતી આપો છો. બિન-એન્ક્રિપ્ટેડ ઈ-મેઈલ ઈ-મેલ સર્વરના સંચાલકો દ્વારા વાંચી શકાય છે જ્યાં કનેક્શન એનક્રિપ્ટેડ નથી (કોઈ https નથી). તેમજ ઈન્ટરનેટ સેવા પ્રદાતા અને અન્ય પક્ષો જે તે કનેક્શનના ટ્રાફિકને સુંઘે છે તે વિષયવસ્તુ જાણવા માટે સક્ષમ છે. વધુમાં, આ જ ઈન્ટરનેટ પર જનરેટ થતા કોઈપણ પ્રકારના ટ્રાફિકને લાગુ પડે છે (વેબ બ્રાઉઝિંગ, ઈ-સ્ટ્રન્ટ મેસેજિંગ, ...) વધુ પડતી વ્યક્તિગત માહિતી ન આપવા માટે, ઈ-મેઈલને એન્ક્રિપ્ટ કરી શકાય છે અને વેબપેજ તેમજ અન્ય ઓનલાઈન બ્રાઉઝ કરી શકાય છે. પ્રવૃત્તિઓ અનામીઓ દ્વારા ટ્રેલેસ કરી શકાય છે, અથવા, એવા કિસ્સાઓમાં કે જેઓ ભરોસાપાત્ર નથી, ઓપન સોર્સ ડિસ્ટ્રિબ્યુટેડ અનામીઝર દ્વારા, કહેવાતા મિક્સ નેટ્સ દ્વારા. જાણીતા ઓપન સોર્સ મિક્સ નેટ્સ I2P - ધ અનામિક નેટવર્ક અથવા ટોર છે. તમારી ગોપનીયતાને સુરક્ષિત કરો તમારી ગોપનીયતા, વ્યક્તિગત માહિતીને ઓનલાઈન સુરક્ષિત રાખવા અને છેતરપિંડી અને દુરુપયોગને રોકવા માટે નીચે આપેલ ટિપ્સ અને માર્ગદર્શિકાઓની સૂચિ છે. ઈન્ટરનેટ.

- નવા પાસવર્ડ મેળવો: તમારા દરેક ઓનલાઈન એકાઉન્ટ માટે અલગ-અલગ, મજબૂત પાસવર્ડનો ઉપયોગ કરો જેથી જો એક સાથે ચેડા કરવામાં આવે તો બાકીના સુરક્ષિત રહે. મજબૂત પાસવર્ડમાં અક્ષરો, સંખ્યાઓ, વિવિધ કેસો અને પ્રતીકો હોય છે. તમારા પાસવર્ડની તાકાત અહીં તપાસો.
- જૂના ઓનલાઈન એકાઉન્ટ્સ બંધ કરો: નહિ વપરાયેલ ઓનલાઈન એકાઉન્ટ્સ જવાબદારી છે. હેક્સ તેનો ઉપયોગ તમારા વધુ મહત્વપૂર્ણ એકાઉન્ટ્સમાં ઘૂસણખોરી કરવા માટે કરી શકે છે. તેમાંથી છુટકારો મેળવો.
- તમારા મિત્રોની યાદી ઓછી કરો.
- પેપરલેસ જાઓ: સંવેદનશીલ ડેટા ઓનલાઈન અથવા તમારા મેઈલ એકાઉન્ટમાં રાખશો નહીં.
- સંવેદનશીલ દસ્તાવેજો કટકા કરો: સંવેદનશીલ ડેટા ધરાવતા અનિચ્છનીય દસ્તાવેજોથી છુટકારો મેળવો. કટકા કરનારનો ઉપયોગ કરીને તેમને સુરક્ષિત રીતે નિકાલ કરો.

બ્રાઉઝર ગોપનીયતા આધુનિક બ્રાઉઝર્સમાં ગોપનીયતા વધારવાની ક્ષમતાઓ અને વિકલ્પોની પ્રભાવશાળી શ્રેણી છે. તેઓ કરી શકે છે, ઉદાહરણ તરીકે,

તમે શંકાસ્પદ અથવા કપટપૂર્ણ વેબસાઈટ્સની મુલાકાત લો તે પહેલાં તમને ચેતવણી આપો અને તમારા કમ્પ્યુટર પર ફૂકીઝ જેવી ટ્રેકિંગ ફાઇલો ડાઉનલોડ કર્યા વિના તમને વેબ સર્ફ કરવાની પણ મંજૂરી આપી શકે છે. ઉપરાંત, જ્યારે કોઈ વેબસાઈટ SSL, સુરક્ષાનો ઉપયોગ કરે છે ત્યારે મોટાભાગના બ્રાઉઝર તમને જાણ કરી શકે છે માપો કે જે તમારા ડેટાને એન્ક્રિપ્ટ કરે છે. જ્યારે કોઈ વેબસાઈટ SSL નો ઉપયોગ કરે છે ત્યારે બ્રાઉઝર તમને પેડલોક આઈકન (સામાન્ય રીતે તમારા બ્રાઉઝરના નીચેના બાર પર સ્થિત) દર્શાવીને અથવા એફેસ બારમાં વેબસાઈટના નામને લીલા રંગમાં હાઈલાઈટ કરીને આનો સંકેત આપી શકે છે.

ફક્ત વિશ્વસનીય વેબસાઈટ્સની મુલાકાત લો. તમે જે સાઈટ ખોલી રહ્યા છો તેના વિશે જાણવા માટે સાઈટ એડવાઈઝર વગેરે જેવી એપ્લિકેશનનો ઉપયોગ કરો. જ્યારે વેબસાઈટ્સ આજે વધુ માહિતી શેર કરે છે, ત્યારે તેઓ તેમના વપરાશકર્તાઓને આ શેરિંગ પ્રવૃત્તિઓ પર મહાન વિશિષ્ટતા અને

નિયંત્રણ પણ પ્રદાન કરે છે. ઘણી વેબસાઈટ્સ પર તમે જોશો કે જ્યારે તમે વ્યક્તિગત માહિતી અથવા સામગ્રી શેર કરો છો ત્યારે તમે તમારા પ્રેક્ષકોને વ્યાખ્યાયિત કરી શકો છો, પછી ભલે તે એક અથવા સમગ્ર જનતાના પ્રેક્ષકો હોય

છેલ્લા દાયકામાં ઈમેલ મોટાભાગે યથાવત છે. જોકે, ઈમેલનો ઉપયોગ કરવાની પદ્ધતિઓ નોંધપાત્ર રીતે વિકસિત થઈ છે અને ઈમેલ વાતાવરણમાં વ્યક્તિગત માહિતીનું રક્ષણ કરવું વધુ પડકારરૂપ બની ગયું છે. પાછલા દાયકામાં હેકિંગ વધુ અસરકારક અને ફિશિંગ તકનીકો બની છે, વધુ વિસ્તૃત. ઈમેઈલનો ઉપયોગ કરતી વખતે તમારી ગોપનીયતાને સુરક્ષિત રાખવા માટે અહીં કેટલીક વ્યૂહરચનાઓ છે:

- 1 ગૌણ, “સ્પામ” ઈમેઈલ સરનામાંનો ઉપયોગ કરો
- 2 મજબૂત સુરક્ષા અને સ્પામ ફિલ્ટર્સ સાથે ઈમેઈલ સેવા પ્રદાતાઓનો ઉપયોગ કરો
- 3 ઈમેલ ખોલતી વખતે સાવધાની રાખો
- 4 ઓળખો કે ઈમેઈલ નિખાલસતા અને ઈન્ટરનેટક્રિપ્ટિવિટી તરફ વિકસિત થઈ રહ્યો છે
- 5 મજબૂત પાસવર્ડ્સનો ઉપયોગ કરો અને સાઈન આઉટ કરવાનું યાદ રાખો

શ્રેષ્ઠ ઓનલાઈન ગોપનીયતા વ્યવહાર

- 1 વ્યક્તિગત માહિતી શેરિંગ ન્યૂનતમ કરો
- 2 વેબસાઈટ્સ પર વિશ્વાસના ચિહ્નો માટે જુઓ અને તેમની અધિકૃતતા ચકાસો
- 3 ઓનલાઈન ખરીદી કરતી વખતે અસ્થાયી કેડિટ કાર્ડ નંબરો ધ્યાનમાં લો
- 4 મજબૂત પાસવર્ડ્સનો ઉપયોગ કરો અને સાઈન આઉટ કરવાનું યાદ રાખો
- 5 તમારા પાસવર્ડ વારંવાર બદલો.
- 6 એન્ટી વાઈરસ અને એન્ટી સ્પાયવેર પ્રોટેક્શનનો ઉપયોગ કરો
- 7 બ્રાઉઝરનો લાભ લો ગોપનીયતા વધારવાની ક્ષમતાઓ અને વિકલ્પો
- 8 તમારા બ્રાઉઝર અને અન્ય સાધનોને અપડેટ કરો.

મોબાઈલ ગોપનીયતા

- 1 મોબાઈલ ઉપકરણો પર તમારી વ્યક્તિગત માહિતી સાથે ઉપકરણની ચોરી અથવા ખોટ દ્વારા ચેડા થવાની શક્યતા વધુ છે - યોગ્ય સાવચેતી રાખો
- 2 તમારું મોબાઈલ ઉપકરણ તમારા સ્થાનથી વાકેફ હોઈ શકે છે અને તે ડેટાને એપ્લિકેશનો અને જાહેરાતકર્તાઓ સાથે શેર કરી શકે છે સાયબર ક્રાઈમ એક્ટ સાયબર વિશ્વના યુગમાં કમ્પ્યુટરનો ઉપયોગ વધુ લોકપ્રિય બન્યો, ટેકનોલોજીના વિકાસમાં પણ વિસ્તરણ થયું, અને ‘સાયબર’ શબ્દ લોકો માટે વધુ પરિચિત બન્યો. ઈન્ફર્મેશન ટેકનોલોજી (IT) ના વિકાસે સાયબર સ્પેસને જન્મ આપ્યો જેમાં ઈન્ટરનેટ તમામ લોકોને ઉચ્ચ ટેકનોલોજીના ઉપયોગ સાથે કોઈપણ માહિતી, ડેટા સ્ટોરેજ, પૃથ્થકરણ વગેરેને એક્સેસ કરવા માટે સમાન તકો પૂરી પાડે છે. નેટીઝન્સની સંખ્યામાં વધારો થવાને કારણે, સાયબર સ્પેસમાં ટેકનોલોજીનો દુરુપયોગ વધી રહ્યો હતો જેણે સ્થાનિક અને

આંતરરાષ્ટ્રીય સ્તરે પણ સાયબર ગુનાઓને જન્મ આપ્યો હતો. જોકે કાઈમ શબ્દ તેનો સામાન્ય વહન કરે છે જેનો અર્થ થાય છે “કાયદેસરની ખોટી જે ફોજદારી કાર્યવાહી દ્વારા અનુસરવામાં આવી શકે છે જે સજામાં પરિણમી શકે છે” જ્યારે સાયબર કાઈમ “ગેરકાનૂની કૃત્યો હોઈ શકે છે જેમાં કમ્પ્યુટર કાં તો સાધન અથવા લક્ષ્ય અથવા બંને છે”.

સાયબર ટેકનોલોજીની પ્રગતિ સાથે ‘ડેટા પ્રોટેક્શન એક્ટ, 1970’ ના રૂપમાં જર્મન રાજ્ય હેસી દ્વારા વર્ષ 1970માં વિશ્વનો પ્રથમ કમ્પ્યુટર વિશિષ્ટ કાયદો ઘડવામાં આવ્યો હતો. ટેકનોલોજીના ઉદભવ સાથે ટેકનોલોજીનો દુરુપયોગ પણ તેના મહત્તમ સ્તરે વિસ્તર્યો છે અને પછી સાયબર વિશ્વમાં ગુનાહિત પ્રવૃત્તિઓને નિયંત્રિત કરવા અને તકનીકી પ્રગતિની સિસ્ટમને સુરક્ષિત કરવા માટે કડક કાયદાકીય કાયદાઓની જરૂરિયાત ઊભી થઈ છે. આ સંજોગોમાં ભારતીય સંસદે 17મી ઓક્ટોબરે ઈ-કોમર્સ, ઈ-ગવર્નન્સ, ઈ-બેંકિંગ તેમજ આ ક્ષેત્રમાં દંડ અને સજાના ક્ષેત્રમાં ટેકનોલોજી સાથે વ્યવહાર કરવા માટેનો સંપૂર્ણ કાયદો ધરાવતો તેનો “ઈન્ફોર્મેશન ટેકનોલોજી એક્ટ, 2000” પસાર કર્યો હતો. સાયબર ગુનાઓ.

સાયબર કાઈમ્સ વાસ્તવમાં તે હેકર્સ તમારી સાઈટની તોડફોડ કરી શકે છે, ગોપનીય માહિતી જોતા હોય છે, ઈન્ટરનેટના ઉપયોગથી વેપારના રહસ્યો અથવા બૌદ્ધિક સંપત્તિની ચોરી કરી શકે છે. તેમાં ‘સેવાઓનો ઈનકાર’ અને નિયમિત ટ્રાફિકને તમારા સુધી પહોંચતા અટકાવતા વાયરસ હુમલાનો પણ સમાવેશ થઈ શકે છે સાઈટ સાયબર ગુનાઓ વાયરસના કેસ સિવાય અને સુરક્ષા સંબંધિત સાયબર અપરાધોના સંદર્ભમાં બહારના લોકો સુધી મર્યાદિત નથી કે જે સામાન્ય રીતે ચોક્કસ કંપનીના કર્મચારીઓ દ્વારા કરવામાં આવે છે જેઓ તેમના લાભ માટે કંપનીના પાસવર્ડ અને ડેટા સ્ટોરેજને સરળતાથી એક્સેસ કરી શકે છે. સાયબર અપરાધોમાં કોમ્પ્યુટરના ઉપયોગથી કરવામાં આવતી ગુનાહિત પ્રવૃત્તિઓનો પણ સમાવેશ થાય છે જે ગુનાઓને વધુ ટકાવી રાખે છે જેમ કે નાણાકીય ગુનાઓ, ગેરકાયદેસર લેખોનું વેચાણ, પોર્નોગ્રાફી, ઓનલાઈન જુગાર, બૌદ્ધિક સંપદા અપરાધ, ઈમેલ, સ્પૂફિંગ, બનાવટી, સાયબર બદનકી, સાયબર સ્ટેકિંગ, અનધિકૃત એક્સેસ. કમ્પ્યુટર સિસ્ટમ, ઇલેક્ટ્રોનિક સ્વરૂપમાં રહેલી માહિતીની ચોરી, ઈ-મેલ બોમ્બિંગ, કમ્પ્યુટર સિસ્ટમને શારીરિક રીતે નુકસાન પહોંચાડવું વગેરે.

સાયબર કાઈમ્સનું વર્ગીકરણ: સાયબર કાઈમ્સ જે દિવસેને દિવસે વધી રહ્યા છે, ખરેખર સાયબર કાઈમ શું છે અને પરંપરાગત ગુનો શું છે તે શોધવું ખૂબ જ મુશ્કેલ છે તેથી આ મૂંઝવણમાંથી બહાર આવવા માટે, સાયબર ગુનાઓને વિવિધ કેટેગરીમાં વર્ગીકૃત કરી શકાય છે જે નીચે મુજબ છે: સાયબર વ્યક્તિઓ સામેના ગુનાઓ: વ્યક્તિઓના વ્યક્તિત્વને અસર કરતા અમુક ગુનાઓ છે જેને આ રીતે વ્યાખ્યાયિત કરી શકાય છે: ઈ-મેઈલ દ્વારા પજવણી: પત્રો, ફાઈલો અને ફોલ્ડર્સના જોડાણો એટલે કે ઈ-મેઈલ દ્વારા સત્તામણી એ ખૂબ જ સામાન્ય પ્રકારનો છે. હાલમાં સોશિયલ સાઈટ્સ એટલે કે ફેસબુક, ટ્વિટર વગેરેના ઉપયોગથી હેરાનગતિ સામાન્ય છે.

દિવસેને દિવસે વધી રહી છે.

સાયબર-સ્ટોફિંગનો અર્થ એવો થાય છે કે વ્યક્ત અથવા ગર્ભિત ભૌતિક ધમકી કે જે ઈન્ટરનેટ, ઈ-મેલ, ફોન, ટેક્સ્ટ સંદેશાઓ, વેબકેમ, વેબસાઈટ્સ અથવા વિડિયો જેવી કમ્પ્યુટર ટેકનોલોજીના ઉપયોગ દ્વારા ભય પેદા કરે છે.

અશલીલ સામગ્રીનો પ્રસાર તેમાં અશલીલનો સમાવેશ થાય છે

એક્સપોઝર/ પોર્નોગ્રાફી (મૂળભૂત રીતે બાળ પોર્નોગ્રાફી), આ પ્રતિબંધિત સામગ્રી ધરાવતી વેબ સાઈટનું હોસ્ટિંગ. આ અશલીલ બાબતો કિશોરોના મનને નુકસાન પહોંચાડી શકે છે અને તેમના મનને ભ્રષ્ટ્ર અથવા ભ્રષ્ટ્ર કરી શકે છે.

બદનકી

કોઈ પણ વ્યક્તિનું મેઈલ એકાઉન્ટ હેક કરીને અને અજાણ્યા વ્યક્તિના મેઈલ એકાઉન્ટમાં અભદ્ર ભાષાનો ઉપયોગ કરીને કેટલાક મેઈલ મોકલીને વ્યક્તિની ગરિમાને નીચું લાવવાના ઈરાદા સાથે તેને દોષિત ઠેરવવાનું કાર્ય છે.

હેકિંગ તેનો અર્થ છે કોમ્પ્યુટર સિસ્ટમ પર અનધિકૃત નિયંત્રણ/એક્સેસ અને હેકિંગની ક્રિયા સમગ્ર ડેટા તેમજ કોમ્પ્યુટર પ્રોગ્રામને સંપૂર્ણપણે નાશ કરે છે. હેકર્સ સામાન્ય રીતે ટેલિકોમ્યુનિકેશન અને મોબાઈલ નેટવર્કને હેક કરે છે.

કેકીંગ તે અત્યાર સુધીના સૌથી ગંભીર સાયબર ગુનાઓમાંનો એક છે. કોઈ અજાણી વ્યક્તિ તમારી જાણ અને સંમતિ વિના તમારી કોમ્પ્યુટર સિસ્ટમમાં ઘૂસી ગઈ છે અને કિંમતી ગોપનીય ડેટા અને માહિતી સાથે છેડછાડ કરી છે તે જાણવું એક ભયાનક લાગણી છે. ઈ-મેઈલ સ્પૂફિંગ એક સ્પુફ ઈ-મેઈલ એક કહેવાય છે, જે તેના મૂળને ખોટી રીતે રજૂ કરે છે. તે દર્શાવે છે કે તે મૂળ જેમાંથી વાસ્તવમાં ઉદ્ભવે છે તેનાથી અલગ છે. એસએમએસ સ્પૂફિંગ: સ્પૂફિંગ એ સ્પામ દ્વારા અવરોધિત છે જેનો અર્થ છે અનિચ્છનીય બિનઆમંત્રિત સંદેશાઓ. અહીં એક અપરાધી મોબાઈલના રૂપમાં બીજાની ઓળખ ચોરી લે છે ફોન નંબર અને ઈન્ટરનેટ અને રીસીવર દ્વારા એસએમએસ મોકલવાથી પીડિતના મોબાઈલ ફોન નંબર પરથી એસએમએસ મળે છે. તે કોઈપણ વ્યક્તિ સામે ખૂબ જ ગંભીર સાયબર અપરાધ છે.

કાર્ડિંગનો અર્થ થાય છે ખોટા એટીએમ કાર્ડ્સ એટલે કે ડેબિટ અને ક્રેડિટ કાર્ડનો ઉપયોગ ગુનેગારો દ્વારા તેમના નાણાકીય લાભ માટે પીડિતના બેંક ખાતામાંથી નિષ્ઠાપૂર્વક પૈસા ઉપાડીને. આ પ્રકારના સાયબર ગુનાઓમાં હંમેશા એટીએમ કાર્ડનો અનધિકૃત ઉપયોગ થતો હોય છે.

છેતરપિંડી અને છેતરપિંડી

તેનો અર્થ એ છે કે જે વ્યક્તિ સાયબર કાઈમનું કૃત્ય કરી રહી છે એટલે કે પાસવર્ડ અને ડેટા સ્ટોરેજની ચોરી કરી રહી છે તેણે તે દોષિત મન સાથે કર્યું છે જે છેતરપિંડી અને છેતરપિંડી તરફ દોરી જાય છે.

બાળ પોર્નોગ્રાફી

તેમાં સગીર વયના બાળકોનું જાતીય શોષણ કરતી સામગ્રી બનાવવા, વિતરિત કરવા અથવા એક્સેસ કરવા માટે કમ્પ્યુટર નેટવર્કનો ઉપયોગ સામેલ છે.

ધમકી દ્વારા હુમલો એ કોમ્પ્યુટર નેટવર્ક એટલે કે ઈ-મેઈલ, વિડીયો અથવા ફોનના ઉપયોગ દ્વારા કોઈ વ્યક્તિને તેમના જીવન અથવા તેમના પરિવારના જીવન માટે ભય સાથે ધમકી આપવાનો ઉલ્લેખ કરે છે. વ્યક્તિની સંપત્તિ સામેના ગુનાઓ:

માં ઝડપી વૃદ્ધિ થઈ રહી હોવાથી આંતરરાષ્ટ્રીય વેપાર જ્યાં વ્યવસાયો અને ઉપભોક્તાઓ પરંપરાગત કાગળના દસ્તાવેજોને બદલે ઇલેક્ટ્રોનિક સ્વરૂપમાં માહિતી બનાવવા, પ્રસારિત કરવા અને સંગ્રહિત કરવા માટે કમ્પ્યુટરનો વધુને વધુ ઉપયોગ કરી રહ્યા છે.

અમુક ગુનાઓ છે જે વ્યક્તિની મિલકતને અસર કરે છે જે નીચે મુજબ છે: બૌદ્ધિક સંપદા ગુનાઓ બૌદ્ધિક સંપદામાં અધિકારોના બંડલનો સમાવેશ થાય છે. કોઈપણ ગેરકાનૂની કૃત્ય કે જેના દ્વારા માલિકને તેના અધિકારોથી સંપૂર્ણ અથવા આંશિક રીતે વંચિત કરવામાં આવે છે તે ગુનો છે. IPR ઉલ્લંઘનનું સામાન્ય સ્વરૂપ સોફ્ટવેર પાયરસી, કૉપિરાઈટનું ઉલ્લંઘન, ટ્રેડમાર્ક, પેટન્ટ, ડિઝાઈન અને સર્વિસ માર્કનું ઉલ્લંઘન, કમ્પ્યુટર સ્રોત

કોડની ચોરી વગેરે કહી શકાય.

સાયબર સ્કવોર્ડિંગનો અર્થ એ થાય છે કે જ્યાં બે વ્યક્તિઓ એક જ ડોમેન નામ માટે દાવો કરે છે અથવા તો દાવો કરે છે કે તેઓએ બીજા પહેલાં તેનો ઉપયોગ કરવાના અધિકાર દ્વારા અથવા અગાઉ તેના જેવું જ કંઈક ઉપયોગ કરીને નામ નોંધ્યું છે.

સાયબર તોડફોડ તોડફોડનો અર્થ છે જાણીજોઈને બીજાની મિલકતનો નાશ કરવો અથવા નુકસાન પહોંચાડવું. આમ સાયબર તોડફોડનો અર્થ થાય છે જ્યારે નેટવર્ક સેવા બંધ અથવા વિક્ષેપિત થાય ત્યારે ડેટાનો નાશ કરવો અથવા તેને નુકસાન પહોંચાડવું. તે તેના કાર્યક્ષેત્રમાં કોઈપણ વ્યક્તિના કમ્પ્યુટરને કરવામાં આવતી કોઈપણ પ્રકારની શારીરિક હાનિનો સમાવેશ કરી શકે છે. આ કૃત્યો આનું સ્વરૂપ લઈ શકે છે કમ્પ્યુટરની ચોરી, કમ્પ્યુટરનો અમુક ભાગ અથવા કમ્પ્યુટર સાથે જોડાયેલ પેરિફેરલ.

હેકિંગ કોમ્પ્યુટર સિસ્ટમ: હેકટીવિઝમ કોમ્પ્યુટર પર અનધિકૃત એક્સેસ/ નિયંત્રણ દ્વારા પ્રખ્યાત ટ્વિટર, બ્લોગિંગ પ્લેટફોર્મ પર હુમલો કરે છે. હેકિંગ પ્રવૃત્તિને કારણે ડેટાની સાથે કોમ્પ્યુટરનું પણ નુકશાન થશે. સંશોધન પણ ખાસ કરીને સૂચવે છે કે તે હુમલાઓ મુખ્યત્વે નાણાકીય લાભ માટે અને કોઈ ચોક્કસ વ્યક્તિ અથવા કંપનીની પ્રતિષ્ઠા ઘટાડવા માટે ન હતા.

વાઈરસનું પ્રસારણ: વાઈરસ એ એવા પ્રોગ્રામ્સ છે જે પોતાને કોમ્પ્યુટર અથવા ફાઈલ સાથે જોડે છે અને પછી પોતાની જાતને અન્ય ફાઈલો અને નેટવર્ક પરના અન્ય કોમ્પ્યુટરમાં પરિભ્રમણ કરે છે. તેઓ સામાન્ય રીતે કમ્પ્યુટર પરના ડેટાને બદલીને અથવા કાઢી નાખીને અસર કરે છે. કૃમિના હુમલા વ્યક્તિઓની કોમ્પ્યુટરાઈઝ સિસ્ટમને અસર કરવામાં મુખ્ય ભૂમિકા ભજવે છે.

સાયબર ટ્રેસેસ: તેનો અર્થ એ છે કે માલિકની યોગ્ય અધિકૃતતા વિના કોઈના કમ્પ્યુટરને એક્સેસ કરવું અને વાયરલેસ ઈન્ટરનેટનો ઉપયોગ કરીને ડેટા અથવા સિસ્ટમને ખલેલ પહોંચાડવા, બદલાવ, દુરુપયોગ અથવા નુકસાન ન કરે. જોડાણ

ઈન્ટરનેટ સમયની ચોરી: મૂળભૂત રીતે, ઈન્ટરનેટ સમયની ચોરી હેકિંગ હેઠળ આવે છે. તે અનધિકૃત વ્યક્તિ દ્વારા અન્ય વ્યક્તિ દ્વારા ચૂકવવામાં આવેલા ઈન્ટરનેટ કલાકોનો ઉપયોગ છે. જે વ્યક્તિ અન્ય કોઈના ISP યુઝર આઈડી અને પાસવર્ડની એક્સેસ મેળવે છે, કાં તો હેકિંગ દ્વારા અથવા ગેરકાયદેસર રીતે તેની એક્સેસ મેળવીને, અન્ય વ્યક્તિની જાણ વગર ઈન્ટરનેટનો ઉપયોગ કરવા માટે તેનો ઉપયોગ કરે છે. જો ઈન્ટરનેટ સમયનો અવારનવાર ઉપયોગ કરવા છતાં વારંવાર રિચાર્જ કરાવવો પડે તો તમે સમયની ચોરીને ઓળખી શકો છો.

સરકાર સામે સાયબર ક્રાઈમ:

ઈન્ટરનેટ સવલતોનો ઉપયોગ કરીને આંતરરાષ્ટ્રીય સરકારોને ધમકી આપવાના ઈરાદા ધરાવતા વ્યક્તિઓના જૂથ દ્વારા અમુક ગુનાઓ કરવામાં આવે છે. તે પણ સમાવેશ થાય:

સાયબર ટેરરિઝમ સાયબર ટેરરિઝમ એ સ્થાનિક તેમજ વૈશ્વિક ચિંતાનો મુખ્ય સળગતો મુદ્દો છે. ઈન્ટરનેટ પરના આ આતંકવાદી હુમલાઓનું સામાન્ય સ્વરૂપ સેવા હુમલા, નફરતની વેબસાઈટ અને અપ્રિય ઈ-મેઈલ, સંવેદનશીલ કોમ્પ્યુટર નેટવર્ક પરના હુમલા વગેરેનો વિતરિત અસ્વીકાર છે. સાયબર આતંકવાદ પ્રવૃત્તિઓ રાષ્ટ્રની સાર્વભૌમત્વ અને અખંડિતતાને જોખમમાં મૂકે છે.

સાયબર વોરફેર

તે તોડફોડ અને જાસૂસી કરવા માટે રાજકીય રીતે પ્રેરિત હેકિંગનો ઉલ્લેખ કરે છે. તે માહિતી યુદ્ધનું એક સ્વરૂપ છે જેને કેટલીકવાર પરંપરાગત યુદ્ધના સમાન તરીકે જોવામાં આવે છે, જોકે આ સામ્યતા તેની ચોક્કસાઈ અને તેના રાજકીય પ્રેરણા બંને માટે વિવાદાસ્પદ છે.

પાઈરેટેડ સોફ્ટવેરનું વિતરણ તેનો અર્થ એ છે કે સરકારના ડેટા અને સત્તાવાર રેકૉર્ડનો નાશ કરવાના હેતુથી એક કમ્પ્યુટરથી બીજા કમ્પ્યુટરમાં પાઈરેટેડ સોફ્ટવેરનું વિતરણ કરવું.

અનધિકૃત માહિતીનો કબજો:

ઈન્ટરનેટની મદદથી આતંકવાદીઓ દ્વારા કોઈપણ માહિતી મેળવવાનું અને તે માહિતી રાજકીય, ધાર્મિક, સામાજિક, વૈચારિક ઉદ્દેશ્યો.

મોટા પ્રમાણમાં સમાજ સામે સાયબર ક્રાઈમ્સ:

સાયબર સ્પેસને નુકસાન પહોંચાડવાના ઈરાદાથી કરવામાં આવેલ ગેરકાનૂની કૃત્ય મોટી સંખ્યામાં વ્યક્તિઓને અસર કરશે. આ ગુનાઓમાં શામેલ છે:

ચાઈલ્ડ પોર્નોગ્રાફી: તેમાં સગીર વયના બાળકોનું જાતીય શોષણ કરતી સામગ્રી બનાવવા, વિતરિત કરવા અથવા એક્સેસ કરવા માટે કમ્પ્યુટર નેટવર્કનો ઉપયોગ સામેલ છે. તેમાં અશિષ્ટ એક્સપોઝર અને અશલીલતા સંબંધિત પ્રવૃત્તિઓનો પણ સમાવેશ થાય છે.

સાયબર ટ્રાફિકિંગ: તે માદક દ્રવ્યો, મનુષ્યો, શસ્ત્રો વગેરેની હેરફેર હોઈ શકે છે જે મોટી સંખ્યામાં વ્યક્તિઓને અસર કરે છે. સાયબર સ્પેસમાં ટ્રાફિકિંગ એ પણ સૌથી મોટો ગુનો છે.

ઓનલાઈન જુગાર ઓનલાઈન છેતરપિંડી અને છેતરપિંડી એ સૌથી વધુ નફાકારક વ્યવસાયો પૈકી એક છે જે આજે સાયબર સ્પેસમાં વધી રહ્યો છે. એવા ઘણા કિસ્સાઓ પ્રકાશમાં આવ્યા છે જે કેડિટ કાર્ડના ગુનાઓ, કરારના ગુનાઓ, જોબ ઓફર કરવા વગેરેને લગતા છે. નાણાકીય ગુનાઓ આ પ્રકારનો ગુનો સામાન્ય છે કારણ કે નેટવર્કિંગ સાઈટ્સ અને ફોન નેટવર્કિંગના વપરાશકર્તાઓમાં ઝડપી વૃદ્ધિ થઈ રહી છે જ્યાં ગુનેગાર પ્રયત્ન કરો ઈન્ટરનેટ દ્વારા બોગસ મેઈલ અથવા મેસેજ મોકલીને હુમલો. ઉદા: ગેરકાયદેસર રીતે પાસવર્ડ મેળવીને કેડિટ કાર્ડનો ઉપયોગ કરવો. બનાવટનો અર્થ છે ધમકીભર્યા મેઈલ મોકલીને મોટી સંખ્યામાં લોકોને છેતરવા કારણ કે ઓનલાઈન બિઝનેસ ટ્રાન્ઝેક્શન એ આજની જીવનશૈલીની આદતની જરૂરિયાત બની ગઈ છે.

કોને અસર કરે છે સાયબર ક્રાઈમ હંમેશા કોઈપણ કદની કંપનીઓને અસર કરે છે કારણ કે લગભગ તમામ કંપનીઓ ઓનલાઈન હાજરી મેળવે છે અને ટેકનોલોજીમાં ઝડપી લાભનો લાભ લે છે પરંતુ તેના સુરક્ષા જોખમો પર વધુ ધ્યાન આપવામાં આવે છે. આધુનિક સાયબર વિશ્વમાં સાયબર ગુનાઓ એ એક મુખ્ય મુદ્દો છે જે વ્યક્તિની સાથે સાથે સમાજને પણ વ્યાપકપણે અસર કરી રહ્યો છે.

સાયબર લો ઈન્ફોર્મેશન ટેકનોલોજીની જરૂરિયાત સમગ્ર વિશ્વમાં ફેલાઈ ગઈ છે.

કમ્પ્યુટરનો ઉપયોગ દરેક ક્ષેત્રમાં થાય છે જ્યાં સાયબર સ્પેસ આર્થિક વિકાસ અને માનવ વિકાસ માટે બધાને સમાન તકો પૂરી પાડે છે. જેમ જેમ સાયબર સ્પેસનો ઉપયોગકર્તા વધુને વધુ વૈવિધ્યસભર બની રહ્યો છે અને ઓનલાઈન ક્રિયાપ્રતિક્રિયાની શ્રેણી વિસ્તરી રહી છે તેમ તેમ સાયબર ગુનાઓમાં વિસ્તરણ થઈ રહ્યું છે એટલે કે ઓનલાઈન કોન્ટ્રાક્ટનો ભંગ,

ઓનલાઈન ટોર્ટ અને ગુનાઓ વગેરે. આ પરિણામોને કારણે કડક કાયદો અપનાવવાની જરૂર હતી. સાયબર સ્પેસ ઓથોરિટી સાયબર સંબંધિત ગુનાહિત પ્રવૃત્તિઓનું નિયમન કરવા અને પીડિતને ન્યાયનો બહેતર વહીવટ પ્રદાન કરવા માટે સાયબર અપરાધ.

આધુનિક સાયબર ટેકનોલોજીની દુનિયામાં સાયબર ગુનાઓનું નિયમન કરવું ખૂબ જ જરૂરી છે અને સૌથી અગત્યનું સાયબર આતંકવાદ અને હેકર્સના કેસમાં સાયબર કાયદાને વધુ કડક બનાવવો જોઈએ. 2000' જે કોઈ કોમ્પ્યુટરના માલિકની પરવાનગી વગર કોમ્પ્યુટર સિસ્ટમના સમગ્ર ડેટાને નુકસાન પહોંચાડવાના ઈરાદાથી કોઈપણ કોમ્પ્યુટરને નષ્ટ કરવા, કાઢી નાખવા, ફેરફાર કરવા અને વિક્ષેપિત કરવા અથવા વિક્ષેપનું કારણ બને તેવું કોઈપણ કૃત્ય કરે છે, તે દંડ ભરવા માટે જવાબદાર રહેશે.

ઉપાયના માર્ગ દ્વારા અસરગ્રસ્ત વ્યક્તિને 1 કરોડ. કલમ: 43A મુજબ જે 'ઇન્ફોર્મેશન ટેકનોલોજી (સુધારા) અધિનિયમ, 2008' દ્વારા દાખલ કરવામાં આવી છે જ્યાં કોઈ સંસ્થા કોર્પોરેટ કેન્દ્ર સરકાર દ્વારા પ્રદાન કરવામાં આવેલ વ્યક્તિઓના ડેટાની જાળવણી અને સુરક્ષા કરી રહી છે, જો કોઈ બેદરકારીભર્યું કૃત્ય અથવા રક્ષણ કરવામાં નિષ્ફળતા હોય તો ડેટા/માહિતી પછી બોડી કોર્પોરેટ અસરગ્રસ્ત વ્યક્તિને વળતર ચૂકવવા માટે જવાબદાર રહેશે. અને વિભાગ 66 'કમ્પ્યુટર સિસ્ટમ સાથે હેકિંગ' સાથે વ્યવહાર કરે છે અને 3 વર્ષ સુધીની કેદ અથવા દંડની જોગવાઈ છે, જે 2 વર્ષ અથવા બંને સુધી લંબાવી શકે છે.

© NIMI
NOT TO BE REPUBLISHED

ક્લાઉડ સેવાઓ સાથે કામ કરે છે (Working with Cloud Services)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ક્લાઉડ કમ્પ્યુટિંગનો પરિચય
- ક્લાઉડ કમ્પ્યુટિંગના પ્રકાર
- ક્લાઉડ કમ્પ્યુટિંગના ગેરફાયદા અને ફાયદા.

ક્લાઉડ કમ્પ્યુટિંગનો પરિચય

ક્લાઉડ કમ્પ્યુટિંગ: ક્લાઉડ કમ્પ્યુટિંગ સેવાઓના લાભો, ગેરફાયદા અને પ્રકારો ક્લાઉડ કમ્પ્યુટિંગ એ આજે બિઝનેસમાં સૌથી વધુ લોકપ્રિય કેચફ્રેઝ છે. તે સંસ્થાઓ દ્વારા માહિતી સંગ્રહિત કરવાની, એક્સેસ કરવાની અને શેર કરવાની, કોમ્પ્યુટિંગ સંસાધનોને સહયોગ અને સંચાલિત કરવાની રીતને બદલી નાખી છે. ઈન્ટરનેટના આગમન સાથે, ક્લાઉડ કોમ્પ્યુટિંગ કંપનીઓને પ્રિમાઈસીસ આઈટી ઈન્ફ્રાસ્ટ્રક્ચર પર પરંપરાગત કરતાં ઉપર આવવાની મંજૂરી આપીને વ્યવસાય ચલાવવાની નવી રીતો પ્રદાન કરી છે.

ક્લાઉડ કમ્પ્યુટિંગ આધુનિક વ્યવસાયોને સુગમતા, કાર્યક્ષમતા, માપનીયતા, સુરક્ષા, સહયોગમાં વધારો અને ઘટાડેલા ખર્ચની ઓફર કરે છે. જ્યારે કોવિડ 19 રોગચાળાએ ક્લાઉડને અપનાવવાની પ્રક્રિયાને વેગ આપ્યો છે, ત્યારે ક્લાઉડ ટેકનોલોજી પરની નિર્ભરતા 2022માં ચાલુ રહેવાની છે, ખાસ કરીને હાઈબ્રિડ વર્ક કેન્દ્રના સ્ટેજને લઈને. તેથી, ભલે સંસ્થા પહેલેથી જ ક્લાઉડ સેવાઓનો ઉપયોગ કરતી હોય અથવા આગામી વર્ષમાં આયોજન કરી રહી હોય, ક્લાઉડ સંચાલિત ઉકેલોનો સંપૂર્ણ લાભ લેવા માટે ક્લાઉડ કમ્પ્યુટિંગની મૂળભૂત બાબતોને સમજવી હિતાવહ છે.

આ બ્લોગમાં, અમે અન્વેષણ કરીશું ક્લાઉડ કમ્પ્યુટિંગ બરાબર શું છે, તે કેવી રીતે કાર્ય કરે છે, તેના ફાયદા અને ગેરફાયદા અને કંપનીઓ તેમના SaaS ડેટાને વધુ સારી રીતે કેવી રીતે સુરક્ષિત કરી શકે છે



ક્લાઉડ કમ્પ્યુટિંગ શું છે?

ZDNet અનુસાર, “ક્લાઉડ કમ્પ્યુટિંગ એ માંગ પરની કમ્પ્યુટિંગ સેવાઓની ડિલિવરી છે - એપ્લિકેશન્સથી સ્ટોરેજ અને પ્રોસેસિંગ પાવર સુધી - સામાન્ય રીતે ઈન્ટરનેટ પર અને તમે જાઓ તે પ્રમાણે ચૂકવણી કરો.”

સરળ શબ્દોમાં, ક્લાઉડ ઈન્ટરનેટનો સંદર્ભ આપે છે. જ્યારે સંસ્થાઓ વર્ચ્યુઅલ ડેટા સેન્ટર્સમાં ડેટા સ્ટોર કરે છે અથવા તેમના ઉપકરણની હાર્ડ ડ્રાઈવ અથવા ઓન-પ્રિમાઈસ આઈટી ઈન્ફ્રાસ્ટ્રક્ચર પર આધાર રાખવાને બદલે ઈન્ટરનેટ કનેક્શનનો ઉપયોગ કરીને પ્રોગ્રામ્સ એક્સેસ કરે છે, તો તેનો અર્થ એ છે કે તેઓ ક્લાઉડમાં કાર્ય કરી રહ્યાં છે.

ક્લાઉડ કમ્પ્યુટિંગ “તૃતીય પક્ષ ડેટા સેન્ટરમાં સર્વિસ” અથવા સંપૂર્ણ સર્વરલેસ વર્કલોડ જેટલું સરળ હોઈ શકે છે જે અનંત રીતે સ્કેલેબલ અને જિયો રીડન્ડન્ટ છે. ક્લાઉડ સર્વિસ અને સેવાઓ સ્કેલેબલ અને સ્થિતિસ્થાપક છે.

ક્લાઉડ કમ્પ્યુટિંગ કેવી રીતે કાર્ય કરે છે?

ક્લાઉડ કમ્પ્યુટિંગ એ કમ્પ્યુટિંગ સંસાધનોની ડિલિવરી છે, જેમ કે IT ઈન્ફ્રાસ્ટ્રક્ચર અથવા ઈન્ટરનેટ પર ડેટા સેન્ટર.

આ મોડેલ વ્યવસાયોને તેમના પોતાના IT ઈન્ફ્રાસ્ટ્રક્ચર અથવા ડેટા સેન્ટર બનાવવા અને જાળવવાને બદલે સ્ટોરેજ સ્પેસ ભાડે અથવા ક્લાઉડ સેવા પ્રદાતા પાસેથી સોફ્ટવેર પ્રોગ્રામ્સ એક્સેસ કરવાની મંજૂરી આપે છે. ક્લાઉડ કમ્પ્યુટિંગ સેવાઓનો ઉપયોગ કરવાનો એક મોટો ફાયદો એ છે કે કંપનીઓ માત્ર તેઓ જે સંસાધનોનો ઉપયોગ કરે છે તેના માટે ચૂકવણી કરે છે.

તેના તકનીકી પાસાઓને વધુ સારી રીતે સમજવા માટે, ક્લાઉડ કમ્પ્યુટિંગ પ્રક્રિયાઓને ફ્રન્ટ એન્ડ અને બેકએન્ડમાં વિભાજિત કરી શકાય છે. ફ્રન્ટએન્ડ ઘટક વપરાશકર્તાઓને ઈન્ટરનેટ બ્રાઉઝર દ્વારા અથવા ક્લાઉડ કમ્પ્યુટિંગ એપ્લિકેશનનો ઉપયોગ કરીને ક્લાઉડમાં સંગ્રહિત ડેટા અને પ્રોગ્રામ્સને એક્સેસ કરવાની મંજૂરી આપે છે. બેકએન્ડમાં સર્વિસ, કમ્પ્યુટર્સ અને ડેટાબેસેસનો સમાવેશ થાય છે જે ડેટા સ્ટોર કરે છે.

ક્લાઉડ કમ્પ્યુટિંગનો

ઈતિહાસ ટેકનોલોજી રિવ્યુ અનુસાર, “ક્લાઉડ કમ્પ્યુટિંગ” શબ્દનો સૌપ્રથમ ઉલ્લેખ કોમ્પેક આંતરિક દસ્તાવેજમાં 1996માં કરવામાં આવ્યો હતો.

વર્ષ 1999 ક્લાઉડ કમ્પ્યુટિંગ માટે એક સીમાચિહ્નરૂપ હતું જ્યારે સેલ્સફોર્સ ઈન્ટરનેટ પર એન્ટરપ્રાઇઝ એપ્લિકેશન પર્ફોર્માન્સનાર પ્રથમ કંપની બની. આ સોફ્ટવેર-એઝ-એ સર્વિસ (સાસ)ની શરૂઆત પણ હતી.

2002 માં, એમેઝોને એમેઝોન વેબ સર્વિસીસ (AWS) શરૂ કરી, જે ક્લાઉડ કમ્પ્યુટિંગમાં અન્ય નોંધપાત્ર વિકાસ હતો. ક્લાઉડ આધારિત સેવાઓના તેના સમૂહમાં સંગ્રહ, ગણતરી અને માનવ બુદ્ધિનો પણ સમાવેશ થાય છે. 2006માં, એમેઝોને ઈલાસ્ટીક કોમ્પ્યુટ ક્લાઉડ (EC2) લોન્ચ કર્યું, જે વ્યવસાયો તેમજ વ્યક્તિઓને વર્ચ્યુઅલ કોમ્પ્યુટર ભાડે આપવા અને તેમની પોતાની કોમ્પ્યુટર એપ્લિકેશન ચલાવવાની મંજૂરી આપે છે.

વર્ષ 2009 માં ક્લાઉડ કમ્પ્યુટિંગમાં એક અન્ય વિશાળ સીમાચિહ્નરૂપ જોવા મળ્યું કારણ કે Google Workspace (હવે Google Workspace) એ બ્રાઉઝર-આધારિત એન્ટરપ્રાઇઝ એપ્લિકેશન્સ પ્રદાન કરવાનું શરૂ કર્યું. તે જ વર્ષે, માઈક્રોસોફ્ટે માઈક્રોસોફ્ટ એઝ્યુર સાથે ક્લાઉડ કમ્પ્યુટિંગ ક્ષેત્રે પ્રવેશ કર્યો અને ટૂંક સમયમાં ઓરેકલ અને એચપી જેવી કંપનીઓએ તેને અનુસર્યું.

ક્લાઉડ કમ્પ્યુટિંગના ઉદાહરણો શું છે?

ક્લાઉડ કમ્પ્યુટિંગમાં વર્ચ્યુઅલ મશીનોથી લઈને ડેટાબેસેસ સુધીની સર્વરલેસ એપ્લિકેશન્સ સુધીની દરેક વસ્તુનો સમાવેશ થાય છે. વાદળના કેટલાક ઉદાહરણો કમ્પ્યુટિંગમાં શામેલ છે:

Salesforce: Salesforce.com એ SaaS પ્રદાતા છે જે ગ્રાહક સંબંધ સંચાલન (CRM) માં નિષ્ણાત છે. કંપની માર્કેટિંગ, વેચાણ, ગ્રાહક સેવાઓ વગેરેને સંરેખિત કરવામાં મદદ કરવા માટે એન્ટરપ્રાઇઝ એપ્લિકેશન પ્રદાન કરે છે અને વપરાશકર્તાઓને ગમે ત્યાંથી કામ કરવાની મંજૂરી આપે છે.

ડિજિટલ મહાસાગર: આ કંપની ન્યૂયોર્ક સ્થિત છે સોફ્ટવેર માટે ઈન્ફ્રાસ્ટ્રક્ચર-એ-એ-સર્વિસ (IaaS) પ્રદાતાવિકાસકર્તાઓ એકસાથે બહુવિધ ક્લાઉડ સર્વિસ પર ચાલતી એપ્લિકેશનને જમાવવા અને સ્કેલ કરવા માટે વ્યવસાયો DigitalOcean નો ઉપયોગ કરે છે.

માઈક્રોસોફ્ટ એઝ્યુર: માઈક્રોસોફ્ટ એઝ્યુર એ એનું ઉત્તમ ઉદાહરણ છે પ્લેટફોર્મ-એ-એ-સર્વિસ (PaaS) જે સમગ્રને સપોર્ટ કરે છે એપ્લિકેશન ડેવલપમેન્ટ લાઈફસાઈકલ, વિકાસથી જમાવટ સુધી અને તેનાથી આગળ. Azure વિકસાવવા માટે ઘણા બધા સાધનો, ભાષાઓ અને ફ્રેમવર્ક પ્રદાન કરે છે

ડ્રોપબોક્સ: ડ્રોપબોક્સ એ ક્લાઉડ-આધારિત ફાઈલ હોસ્ટિંગ સેવા છે જે વપરાશકર્તાઓને તેમના ઉપકરણોમાં ફાઈલોને સંગ્રહિત અને સમન્વયિત કરવાની મંજૂરી આપે છે જેથી તેઓ તેમને ગમે ત્યાંથી એક્સેસ કરી શકે. તે વપરાશકર્તાઓને અસરકારક સહયોગની સુવિધા આપતા, ઈન્ટરનેટ દ્વારા છબીઓ અને વિડિઓઝ સહિત મોટી ફાઈલોને શેર કરવાની પણ મંજૂરી આપે છે.

ક્લાઉડ કમ્પ્યુટિંગનું મહત્વ શું છે?

ક્લાઉડ કમ્પ્યુટિંગ અસ્તિત્વમાં આવ્યું તે પહેલાં, કંપનીઓએ તેનો ઉપયોગ કરી શકે તે માટે તેમના ભૌતિક પીસી અથવા ઓન-પ્રિમિસીસ સર્વર પર એપ્લિકેશન્સ અથવા પ્રોગ્રામ્સ ડાઉનલોડ કરવાની જરૂર હતી. કોઈપણ સંસ્થા માટે, પોતાનું આઈટી ઈન્ફ્રાસ્ટ્રક્ચર અથવા ડેટા સેન્ટર બનાવવું અને તેનું સંચાલન કરવું એ એક મોટો પડકાર છે. જેઓ પોતાના ડેટા સેન્ટર ધરાવે છે તેમના માટે પણ મોટી સંખ્યામાં IT એડમિનિસ્ટ્રેટર્સ અને સંસાધનોની ફાળવણી એ એક સંઘર્ષ છે.

ક્લાઉડ કમ્પ્યુટિંગ અને વર્ચ્યુઅલાઈઝેશનનો પરિચય એ ટેકનોલોજીના ઇતિહાસમાં એક નમૂનો બદલાવ હતો. ઉદ્યોગ. પોતાની IT બનાવવા અને મેનેજ કરવાને બદલે ઈન્ફ્રાસ્ટ્રક્ચર અને સર્વિસ, પાવર અને વાસ્તવિક માટે ચૂકવણીએસ્ટેટ વગેરે, ક્લાઉડ કમ્પ્યુટિંગ વ્યવસાયોને ભાડે આપવા દે છે ક્લાઉડ સેવા પ્રદાતાઓ પાસેથી કમ્પ્યુટિંગ સંસાધનો. આવ્યવસાયોને ભારે અપફ્રન્ટ બર્થ અને તેમના પોતાના ડેટા સેન્ટરનું સંચાલન કરવાની જટિલતાને ટાળવામાં મદદ કરે છે. ક્લાઉડ સેવાઓ ભાડે આપીને, કંપનીઓ તેઓ જે વાપરે છે તેના માટે જ ચૂકવણી કરે છે જેમ કે કમ્પ્યુટિંગ સંસાધનો અને ડિસ્ક સ્પેસ. આ કંપનીઓને વધુ સચોટતા સાથે બર્થની અપેક્ષા રાખવાની મંજૂરી આપે છે.

ક્લાઉડ સેવા પ્રદાતાઓ ભારે પ્રશિક્ષણ કરે છે

આઈટી ઈન્ફ્રાસ્ટ્રક્ચરનું સંચાલન અને જાળવણી, તે વ્યવસાયો માટે ઘણો સમય, પ્રયત્ન અને નાણાં બચાવે છે. ક્લાઉડ સંસ્થાઓને તેમના કમ્પ્યુટિંગ ઈન્ફ્રાસ્ટ્રક્ચરને જ્યારે જરૂર હોય ત્યારે સીમલેસ રીતે અપસ્કેલ અથવા ડાઉનસ્કેલ કરવાની ક્ષમતા પણ આપે છે. પરંપરાગત ઓન-પ્રિમિસીસ ડેટાની સરખામણીમાં સેન્ટર મોડલ, ક્લાઉડ ગમે ત્યાંથી અને ઈન્ટરનેટ કનેક્ટિવિટીવાળા કોઈપણ ઉપકરણ પર ડેટાની સરળ એક્સેસ પ્રદાન કરે છે, જેનાથી અસરકારક સહયોગ અને ઉન્નત ઉત્પાદકતા સક્ષમ બને છે.

ક્લાઉડ કમ્પ્યુટિંગના સૌથી સામાન્ય ઉપયોગો શું છે?

સ્ટાર્ટઅપ્સથી લઈને મોટા કોર્પોરેશનો અને સરકાર સુધીએજન્સીઓ, દરેક સંસ્થા એક્સેસ કરવા માટે ક્લાઉડનો ઉપયોગ કરે છે વર્કફોર્સને

સુવ્યવસ્થિત કરવા, સુધારવા માટે તકનીકી સેવાઓસંચાર, ઉત્પાદકતા, સેવા વિતરણ અને વધુ. કમોટેથી કમ્પ્યુટિંગના કેટલાક સૌથી સામાન્ય ઉપયોગો નીચે સૂચિબદ્ધ છે.

- **સંગ્રહ :** ક્લાઉડના સૌથી સામાન્ય ઉપયોગોમાંનો એક કમ્પ્યુટિંગ એ ફાઈલ સ્ટોરેજ છે. જ્યારે પીસી પર હાર્ડ ડ્રાઈવો, એક્સ્ટર્નલ હાર્ડ ડ્રાઈવો, યુએસબી ડ્રાઈવો વગેરે જેવા ડેટા સ્ટોર કરવા અને એક્સેસ કરવાના ઘણા વિકલ્પો છે, ત્યારે ક્લાઉડ સ્ટોરેજ વ્યવસાયોને કોઈપણ જગ્યાએથી અને કોઈપણ ઉપકરણ પર ઈન્ટરનેટ કનેક્શન સાથે ડેટાને એકીકૃત રીતે એક્સેસ કરવા સક્ષમ બનાવે છે. મેઘ સંગ્રહ Amazon S3, DropBox અથવા OneDrive જેવી સેવાઓ ડેટાની સુરક્ષિત એક્સેસ પ્રદાન કરે છે અને વ્યવસાયોને તેમની જરૂરિયાતોના આધારે સ્ટોરેજ સ્પેસને અપસ્કેલ અને ડાઉનસ્કેલ કરવાની પણ મંજૂરી આપે છે.
- **ડેટાબેઝ:** ક્લાઉડ ડેટાબેઝ એ અન્ય લોકપ્રિય વ્યવસાય ઉપયોગ કેસ છે. IBM ક્લાઉડ ડેટાબેઝને “ક્લાઉડ પ્લેટફોર્મ દ્વારા બનાવવામાં અને એક્સેસ કરાયેલ ડેટાબેઝ સેવા” તરીકે વ્યાખ્યાયિત કરે છે. ક્લાઉડ ડેટાબેઝ પરંપરાગત ડેટાબેઝની જેમ મોટાભાગની સમાન કાર્યક્ષમતા પ્રદાન કરે છે, પરંતુ વધારાના લાભો જેમ કે લવચીકતા, બર્થ બચત, ફેલઓવર સપોર્ટ, વિશિષ્ટ કુશળતા અને વધુ સાથે.
- **વેબ એપ્લિકેશન્સ:** વેબ એપ્લિકેશન આજે વ્યવસાયો માટે આવશ્યક સાધન છે. ક્લાઉડ ટેકનોલોજી દ્વારા સંચાલિત, કોઈપણ વેબ બ્રાઉઝરનો ઉપયોગ કરીને વેબ-આધારિત એપ્લિકેશન્સને એક્સેસ કરી શકે છે, માહિતીની ત્વરિત રીમોટ એક્સેસ પ્રદાન કરે છે. આનાથી બિઝનેસ પ્રોફેશનલ્સ ગ્રાહકો સાથે વાતચીત કરી શકે છે અને તેઓ સફરમાં હોય ત્યારે તેમને જરૂરી માહિતી પૂરી પાડે છે અને તેમને ગમે ત્યાંથી સાથીદારો સાથે સહયોગ કરવામાં મદદ કરે છે.
- **સહયોગ:** તેની સરળ એક્સેસિબિલિટી, એકીકરણ, લવચીકતા, સુરક્ષા અને ઉપયોગમાં સરળતાને લીધે, ક્લાઉડ આધારિત ટૂલ્સ, જેમ કે Microsoft 365 અને Google Workspace, તમામ વિભાગોમાં આંતરિક રીતે અને ક્લાયન્ટ સાથે બાહ્ય રીતે સહયોગ કરવા માંગતા વ્યવસાયો માટે સ્પષ્ટ પસંદગી બની ગયા છે. જુમેલ, ગૂગલ ડોક્સ, માઈક્રોસોફ્ટ આઉટલુક, માઈક્રોસોફ્ટ વર્ડ, ટીમ્સ વગેરે, સહયોગ અને ઉત્પાદકતા વધારવા માટે રચાયેલ શક્તિશાળી બિઝનેસ ટૂલ્સ છે.
- **SaaS એપ્લિકેશન્સ:** સોફ્ટવેર-એ-એ-સર્વિસ (SaaS) એપ્લિકેશન્સ, જેમ કે સેલ્સફોર્સ, વ્યવસાયોને ડેટા સ્ટોર કરવા, ગોઠવવા અને જાળવવા તેમજ માર્કેટિંગને સ્વચાલિત કરવા અને ક્લાયન્ટ્સને અસરકારક રીતે સંચાલિત કરવાની મંજૂરી આપે છે. SaaS સોલ્યુશન્સ અત્યંત કાર્યાત્મક છે અને તેને સોફ્ટવેર અને/અથવા હાર્ડવેર મેનેજમેન્ટની જરૂર નથી.

ક્લાઉડ કમ્પ્યુટિંગના વિવિધ પ્રકારો શું છે?

ક્લાઉડ કમ્પ્યુટિંગના ચાર મુખ્ય પ્રકાર છે: જાહેર, ખાનગી, હાઈબ્રિડ અને મલ્ટીક્લાઉડ.

જાહેર વાદળ

VMware જાહેર ક્લાઉડને “એક IT મોડેલ જ્યાં ઓન-ડમાન્ડ કમ્પ્યુટિંગ સેવાઓ અને ઈન્ફ્રાસ્ટ્રક્ચર છેતૃતીય-પક્ષ પ્રદાતા દ્વારા સંચાલિત અને તેની સાથે શેર કરવામાં આવે છે જાહેર ઈન્ટરનેટનો ઉપયોગ કરતી બહુવિધ સંસ્થાઓ.” ક્લાઉડ સેવા પ્રદાતાઓ ઈન્ફ્રાસ્ટ્રક્ચર-એ-એ-સેવા (IaaS), પ્લેટફોર્મ-એ-એ-સર્વિસ જેવી વિવિધ સેવાઓ પ્રદાન કરે છે (PaaS) અને SaaS વ્યક્તિઓ અને વ્યવસાયો કે જેઓ આ સેવાઓ માસિક અથવા ઉપયોગ દીઠ ચૂકવણીના આધારે ભાડે આપે છે. Amazon Web Ser-

vices (AWS), Microsoft Azure, Google Cloud, Alibaba Cloud અને IBM Cloud ટોચના પાંચ ક્લાઉડ પ્રદાતાઓ છે.

ખાનગી વાદળ

ખાનગી વાદળ અથવા આંતરિક વાદળ એ છે જ્યાં આઈ.ટી.ઈન્ફ્રાસ્ટ્રક્ચર (હાર્ડવેર અને સોફ્ટવેર સંસાધનો) છેમાત્ર એક સંસ્થાને સમર્પિત, જાહેર ક્લાઉડથી વિપરીત જ્યાં કમ્પ્યુટિંગ સંસાધનો બહુવિધ ભાડૂતો વચ્ચે વહેંચવામાં આવે છે. ખાનગી ક્લાઉડ વાતાવરણ એવા વ્યવસાયો માટે આદર્શ છે કે જેમના માટે નિયમનકારી જરૂરિયાતો પૂરી થાય છે, સુરક્ષા અને નિયંત્રણ પ્રાથમિકતા છે. પરંપરાગત રીતે, ખાનગી ક્લાઉડ કંપનીના ડેટા સેન્ટરમાં હોસ્ટ કરવામાં આવે છે અને તેના પોતાના હાર્ડવેરનો ઉપયોગ કરે છે. જો કે, કોઈ સંસ્થા તૃતીય પક્ષ પ્રદાતાને હોસ્ટિંગ આઉટસોર્સ કરી શકે છે જે રિમોટલી કમ્પ્યુટિંગ સંસાધનોનું સંચાલન કરે છે.

વર્ણસંકર વાદળ

હાઈબ્રિડ ક્લાઉડ એ જાહેર ક્લાઉડ અને પ્રાઈવેટ ક્લાઉડ એન્વાયર્નમેન્ટ્સ એમ બંનેનું મિશ્રણ છે. વ્યવસાયો તેમની ગણતરી ક્ષમતાને પૂરક બનાવવા માટે આ મોડેલનો ઉપયોગ કરે છે. જ્યારે ખાનગી ક્લાઉડની ક્ષમતા તેની ટોચ પર પહોંચે છે, ત્યારે વ્યવસાયો ખાનગી ક્લાઉડની ક્ષમતાઓને વધારવા માટે જાહેર ક્લાઉડનો લાભ લઈ શકે છે. હાઈબ્રિડ ક્લાઉડ વ્યવસાયોને ટ્રાફિક અથવા સેવાની માંગના આધારે ગણતરી ક્ષમતાને ઉપર અથવા નીચે માપવામાં સક્ષમ બનાવે છે. આ નવા સર્વિસ ખરીદવા અને જાળવવાની જરૂરિયાતને દૂર કરે છે, જે વ્યવસાયોને ખર્ચ, સમય અને પ્રયત્ન બચાવવા માટે પરવાનગી આપે છે.

મલ્ટીક્લાઉડ

મલ્ટીક્લાઉડ એ સંયોજનનો ઉપયોગ કરવાની પ્રથા છેવાદળો - બે અથવા વધુ જાહેર અથવા ખાનગી વાદળો, અથવા એકલા ક્લાઉડ પ્રદાતાઓ તરફથી બંનેનું સંયોજન. એમલ્ટીક્લાઉડ અભિગમ વ્યવસાયોને તેમના આધારે વિવિધ ક્લાઉડ વિકેતાઓમાંથી શ્રેષ્ઠ સેવાઓ પસંદ કરવાની મંજૂરી આપે છેબજેટ, તકનીકી જરૂરિયાતો, ભૌગોલિક સ્થાનો અને તેથી વધુ. આ મોડેલ વ્યવસાયોને વિવિધ હેતુઓ માટે વિવિધ વાદળોનો ઉપયોગ કરવા સક્ષમ બનાવે છે. દાખલા તરીકે, સંસ્થા એક ક્લાઉડનો ઉપયોગ સોફ્ટવેર ડેવલપમેન્ટ અને ટેસ્ટિંગ માટે કરી શકે છે, બીજા ક્લાઉડનો ડેટા બેકઅપ અને ડિઝાસ્ટર રિકવરી માટે અને અન્ય ડેટા એનાલિટિક્સ માટે કરી શકે છે.

ક્લાઉડ કમ્પ્યુટિંગના ત્રણ અલગ-અલગ પ્રકાર શું છેસેવાઓ?

ક્લાઉડ કમ્પ્યુટિંગ સેવાઓ ત્રણ પ્રકારની છેઈન્ફ્રાસ્ટ્રક્ચર-એ-એ-સર્વિસ (IaaS), પ્લેટફોર્મ-એ-એ-સર્વિસ (PaaS) અને સોફ્ટવેર-એ-એ-સર્વિસ (SaaS).

ઈન્ફ્રાસ્ટ્રક્ચર-એ-એ-સર્વિસ (IaaS)

IaaS એ ક્લાઉડ કમ્પ્યુટિંગ સેવા છે જ્યાં ક્લાઉડ પ્રદાતાઓ ઈન્ટરનેટ પર વર્ચ્યુઅલાઈઝ્ડ કમ્પ્યુટિંગ ઈન્ફ્રાસ્ટ્રક્ચર પહોંચાડી અને તેનું સંચાલન કરો. ઈન-હાઉસ આઈટી ઈન્ફ્રાસ્ટ્રક્ચર બનાવવાને બદલે, વ્યવસાયો આવશ્યક સંસાધનોને ઍક્સેસ કરી શકે છે, જેમ કે ઓપરેટિંગ સિસ્ટમ્સ, નેટવર્કિંગ, સ્ટોરેજ સ્પેસ, ડેવલપમેન્ટ ટૂલ્સ વગેરે, માંગ પર. આનાથી હાર્ડવેર અને સોફ્ટવેરનો ખર્ચ બચે છે તેમજ IT સ્ટાફનો બોજ ઓછો થાય છે.

પ્લેટફોર્મ-એ-એ-સર્વિસ (PaaS)

PaaS વ્યવસાયોને અંતર્ગત ઈન્ફ્રાસ્ટ્રક્ચર વિશે ચિંતા કર્યા વિના સોફ્ટવેર એપ્લિકેશન્સ અને સેવાઓના વિકાસ, જમાવટ અને સંચાલન પર ધ્યાન

કેન્દ્રિત કરવાની મંજૂરી આપે છે કારણ કે ક્લાઉડ પ્રદાતાઓ ભારે લિફ્ટિંગ કરે છે. PaaS સાથે, ડેવલપર્સ અને પ્રોગ્રામર્સ માત્ર IT ઈન્ફ્રાસ્ટ્રક્ચર જ નહીં પણ એપ્લિકેશન/સોફ્ટવેર પ્લેટફોર્મ અને સોલ્યુશન સ્ટેકની પણ ઍક્સેસ મેળવે છે. PaaS ના કેટલાક ઉદાહરણોમાં AWS Elastic Beanstalk, Google App Engine અને Microsoft Azure નો સમાવેશ થાય છે.

સોફ્ટવેર-એ-એ-સર્વિસ (SaaS)

SaaS વ્યવસાયોને ઉપયોગ માટે તૈયાર સોફ્ટવેર પ્રદાન કરે છે જે વપરાશકર્તાઓને ઈન્ટરનેટ પર પહોંચાડવામાં આવે છે. હાર્ડવેર, સોફ્ટવેર, ડેટા સ્ટોરેજ, પેય મેનેજમેન્ટ અને હાર્ડવેર/સોફ્ટવેર અપડેટ્સ સહિત તમામ અંતર્ગત ઈન્ફ્રાસ્ટ્રક્ચર SaaS પ્રદાતાઓ દ્વારા સંચાલિત થાય છે. SaaS એ સબસ્ક્રિપ્શન-આધારિત મોડલ છે, જેમાં વ્યવસાયોને તેઓ જે સેવાઓનો ઉપયોગ કરવા માગે છે તેનું સબસ્ક્રાઇબ કરવું જરૂરી છે. વપરાશકર્તાઓ વેબ બ્રાઉઝર દ્વારા સીધા જ SaaS એપ્લિકેશનને ઍક્સેસ કરી શકે છે, જે તેમને ડાઉનલોડ અથવા ઈન્સ્ટોલ કરવાની જરૂરિયાતને દૂર કરે છે. SaaS વપરાશકર્તાઓને સક્રિય ઈન્ટરનેટ કનેક્શન સાથે ગમે ત્યાંથી અને કોઈપણ સમયે વેબ-આધારિત સોલ્યુશન્સ ઍક્સેસ કરવાની મંજૂરી આપે છે. કેટલાક લોકપ્રિય SaaS ઉકેલોમાં Microsoft 365, Google Workspace અને Salesforceનો સમાવેશ થાય છે.

ક્લાઉડ કમ્પ્યુટિંગના ફાયદા શું છે?

ક્લાઉડ કમ્પ્યુટિંગ વ્યવસાયોને વર્ચ્યુઅલ રીતે ગમે ત્યાંથી અને વધુ કાર્યક્ષમતા સાથે કામ કરવા સક્ષમ બનાવે છે. કેટલાકક્લાઉડ કમ્પ્યુટિંગના ફાયદાઓમાં નીચેનાનો સમાવેશ થાય છે:

- **ખર્ચ બચત :** ક્લાઉડનો સૌથી મોટો ફાયદોકમ્પ્યુટિંગ ખર્ચમાં ઘટાડો થાય છે. વ્યવસાયોને પોતાનું IT ઈન્ફ્રાસ્ટ્રક્ચર બનાવવાની અથવા હાર્ડવેર અથવા સાધનો ખરીદવાની જરૂર નથી, તેથી તે કંપનીઓને મૂડી ખર્ચમાં નોંધપાત્ર ઘટાડો કરવામાં મદદ કરે છે.
- **સુગમતા/સ્કેલેબિલિટી:** ક્લાઉડ કમ્પ્યુટિંગ વધુ તક આપે છે તમામ કદના વ્યવસાયો માટે સુગમતા. ભલે તેઓને વધારાની બેન્ડવિડ્થ, કમ્પ્યુટિંગ પાવર અથવા સ્ટોરેજ સ્પેસની જરૂર હોય, તેઓ તેમની જરૂરિયાતો અને બજેટના આધારે કમ્પ્યુટિંગ સંસાધનોને એકીકૃત રીતે અપ અથવા ડાઉન કરી શકે છે.
- **સુરક્ષા :** આજે વ્યવસાયો માટે ડેટા સુરક્ષા એ મુખ્ય ચિંતા છે. ક્લાઉડમાં સંવેદનશીલ ડેટા સુરક્ષિત રીતે છે તેની ખાતરી કરવા માટે ક્લાઉડ વિકેતાઓ પ્રમાણીકરણ, ઍક્સેસ મેનેજમેન્ટ, ડેટા એન્ક્રિપ્શન વગેરે જેવી અદ્યતન સુરક્ષા સુવિધાઓ પ્રદાન કરે છે. નિયંત્રિત અને સંગ્રહિત.
- **ગતિશીલતા:** ક્લાઉડ કમ્પ્યુટિંગ વપરાશકર્તાઓને ઍક્સેસ કરવાની મંજૂરી આપે છે કોઈપણ ઉપકરણમાંથી, ગમે ત્યાં અને કોઈપણ સમયે, ઈન્ટરનેટનો ઉપયોગ કરીને કોર્પોરેટ ડેટા. સગવડતાથી ઉપલબ્ધ માહિતી સાથે, કર્મચારીઓ સફરમાં પણ ઉત્પાદક રહી શકે છે.
- **વધેલો સહયોગ :** ક્લાઉડ એપ્લિકેશનને મંજૂરી આપે છેવ્યવસાયો એકીકૃત રીતે વાતચીત કરવા અને માહિતીને સુરક્ષિત રીતે ઍક્સેસ કરવા અને શેર કરવા, સહયોગને સરળ અને મુશ્કેલી-મુક્ત બનાવે છે. ક્લાઉડ કમ્પ્યુટિંગ 1 બહુવિધ વપરાશકર્તાઓને દસ્તાવેજોને સંપાદિત કરવા અથવા ફાઇલો પર એકસાથે અને પારદર્શક રીતે કામ કરવાની શક્તિ આપે છે.
- **આપત્તિ પુનઃપ્રાપ્તિ:** ડેટાની ખોટ અને ડાઉનટાઇમ કારણ બની શકે છેકોઈપણ કદના વ્યવસાયોને ન ભરી શકાય તેવું નુકસાન. મુખ્ય ક્લાઉડ વિકેતાઓ ઉચ્ચ એપ્લિકેશન ઉપલબ્ધતા અને વ્યવસાય

સાતત્યની ખાતરી કરવા માટે હાર્ડવેર/સોફ્ટવેર નિષ્ફળતા, કુદરતી આફતો અને પાવર આઉટેજ જેવી અણધાર્યા વિક્ષેપકારક ઘટનાઓનો સામનો કરવા માટે સારી રીતે સજ્જ છે.

- **ઓટોમેટિક અપડેટ્સ:** મેન્યુઅલ ઓર્ગનાઇઝેશન વાઈડ સોફ્ટવેર અપડેટ્સ કરવા માટે IT સ્ટાફનો ઘણો મૂલ્યવાન સમય લાગી શકે છે. જો કે, ક્લાઉડ કમ્પ્યુટિંગ સાથે, સેવા પ્રદાતાઓ વ્યવસાયોને અદ્યતન સોફ્ટવેર સંસ્કરણો, નવીનતમ સર્વિસ અને અપગ્રેડેડ પ્રદાન કરવા માટે નવીનતમ તકનીક સાથે સિસ્ટમને નિયમિતપણે તાજું અને અપડેટ કરે છે. પ્રક્રિયા શક્તિ.

ક્લાઉડ કમ્પ્યુટિંગના ગેરફાયદા શું છે?

ક્લાઉડમાં ઓપરેટ કરવાના ફાયદા અપાર છે. જો કે, ક્લાઉડ પર સંક્રમણ કરવાનું નક્કી કરતા પહેલા કંપનીઓએ જાણવું જોઈએ કે કેટલાક ગેરફાયદા છે. ક્લાઉડ કમ્પ્યુટિંગના ટોચના પાંચ ગેરફાયદા નીચે સૂચિબદ્ધ છે.

- 1 **ડાઉનટાઇમ:** ક્લાઉડ કમ્પ્યુટિંગ સિસ્ટમ્સ હોવાથી સક્રિય વિના, ઈન્ટરનેટ પર સંપૂર્ણપણે નિર્ભર ઈન્ટરનેટ કનેક્શન, વ્યવસાયો ક્લાઉડમાં હોસ્ટ કરેલા ડેટા અથવા એપ્લિકેશનને ઍક્સેસ કરી શકતા નથી. ગૂગલે ત્રણનો ભોગ લીધો 2020 માં ગંભીર આઉટેજ કે જેણે સમગ્ર વિશ્વમાં તેની મોટાભાગની સેવાઓ અને વપરાશકર્તાઓને અસર કરી.
- 2 **વેન્ડર લોક-ઇન:** કંપનીના વર્કલોડને સ્થાનાંતરિત કરવું અને એક ક્લાઉડ પ્રદાતાથી બીજાને સેવાઓ એ મુખ્ય છે ક્લાઉડ કમ્પ્યુટિંગમાં પડકાર. મેઘ વાતાવરણ વચ્ચેનો તફાવત સુસંગતતા અથવા એકીકરણનું કારણ બની શકે છે મુદ્દાઓ જો સંક્રમણ યોગ્ય રીતે નિયંત્રિત ન થાય, તો તે થઈ શકે છે સંસ્થાના ડેટાને બિનજરૂરી સુરક્ષા માટે ખુલ્લા પાડો નબળાઈઓ
- 3 **મર્યાદિત નિયંત્રણ :** ક્લાઉડ ઈન્ફ્રાસ્ટ્રક્ચર સંપૂર્ણ માલિકીનું અને ક્લાઉડ વેન્ડર દ્વારા સંચાલિત હોવાથી, ક્લાઉડ કમ્પ્યુટિંગ સેવાઓનો ઉપયોગ કરતા વ્યવસાયો તેમના ડેટા, એપ્લિકેશન્સ અને સેવાઓ પર મર્યાદિત નિયંત્રણ ધરાવે છે. તેથી, ક્લાઉડ ઈન્ફ્રાસ્ટ્રક્ચરમાં વ્યવસાય શું કરી શકે છે અને શું કરી શકતો નથી તે સમજવા માટે યોગ્ય એન્ડ-યુઝર લાઇસન્સ એગ્રીમેન્ટ (EULA) હોવું મહત્વપૂર્ણ છે.
- 4 **સુરક્ષા:** સંગ્રહ કરવાની મુખ્ય ચિંતાઓમાંની એક ક્લાઉડમાં કંપનીનો સંવેદનશીલ ડેટા સુરક્ષા છે. જોકે ક્લાઉડ સેવા પ્રદાતાઓ અદ્યતન સુરક્ષા લાગુ કરે છે પગલાં, રિમોટ સર્વિસ પર ગોપનીય ફાઇલોને સંગ્રહિત કરવી જે સંપૂર્ણપણે માલિકીની છે અને તૃતીય પક્ષ દ્વારા સંચાલિત છે તે હંમેશા સુરક્ષા જોખમો ખોલે છે. જ્યારે કોઈ સંસ્થા ક્લાઉડ કમ્પ્યુટિંગ મોડલ અપનાવે છે, ત્યારે IT સુરક્ષા જવાબદારી ક્લાઉડ વેન્ડર અને વપરાશકર્તા વચ્ચે વહેંચવામાં આવે છે. જેમ કે, દરેક પક્ષ મિલકતો, પ્રક્રિયાઓ માટે જવાબદાર છે અને કાર્યો તેઓ નિયંત્રિત કરે છે.
- 5 **ડેટા ખોટ કે ચોરી:** વર્ચ્યુઅલ ડેટામાં નિર્ણાયક ડેટાનો સંગ્રહ કરવો કેન્દ્રો વિવિધ પ્રકારના જોખમો માટે દરવાજા ખોલી શકે છે જે ડેટાની ખોટ તરફ દોરી શકે છે, જેમ કે ક્લાઉડ ખોટી ગોઠવણી, માહિતીની ચોરી, સુરક્ષા ભંગ, ચોરાયેલી ઓળખપત્રો, વગેરે. વધુમાં, ક્લાઉડ સેવા પ્રદાતાઓ, જેમ કે માઈક્રોસોફ્ટ અને ગૂગલ, એક વહેંચાયેલ જવાબદારી મોડલને અનુસરે છે, જ્યાં વિક્રેતા એપ્લિકેશનની ઉપલબ્ધતા અને તેમાં જરૂરી દરેક વસ્તુની જવાબદારી સ્વીકારે છે, જ્યારે ગ્રાહક જાળવી રાખે છે. એપ્લિકેશન ડેટા માટેની જવાબદારી, વહીવટ અને વપરાશકર્તા સંચાલન.

સ્પાનિંગ બેકઅપ સાથે SaaS ડેટા સુરક્ષાને બહેતર બનાવો

સ્ટેટિસ્ટા અનુસાર, 2021 સુધીમાં, લગભગ 50% કોર્પોરેટ ડેટા ક્લાઉડમાં સંગ્રહિત થાય છે. ડેટા સૂચવે છે કે વ્યવસાયો વૈશ્વિક સ્તરે તેમના ક્લાઉડ સેવા પ્રદાતાઓને તેમના સંવેદનશીલ ડેટા સાથે વિશ્વાસ કરે છે. કંપની જે ક્લાઉડ વિક્રેતા પસંદ કરે છે તેને ધ્યાનમાં લીધા વિના, તેણે સાવચેત રહેવું જોઈએ કે ક્લાઉડના લાભો અંતર્ગત સુરક્ષા જોખમો કરતાં વધી ન જાય.

માઈક્રોસોફ્ટ 365 જેવા SaaS સોલ્યુશન્સનો ઉપયોગ કરતા વ્યવસાયો, Google Workspace અને Salesforce દરોજ ડેટા ગુમાવે છે. ઘણી કંપનીઓ માને છે કે SaaS વિક્રેતાઓ છે તેમના ડેટાના રક્ષણ માટે જવાબદાર છે. જો કે, તે કેસ નથી. જ્યારે SaaS પ્રદાતાઓ એપ્લિકેશન અપડેટ્સ અને ઉપલબ્ધતાને સુનિશ્ચિત કરે છે, ત્યારે ડેટા સુરક્ષા ગ્રાહકોની છે જવાબદારી

જેમ કે, વ્યવસાયોને વિશ્વસનીય SaaS બેકઅપ સોલ્યુશનની જરૂર છે જે તેમના મૂલ્યવાન ડેટાને સૌથી વધુ સામે રક્ષણ આપી શકે. ફિશીંગ, રેન્સમવેર અને માલવેર હુમલા, માનવીય ભૂલ, દૂષિત વર્તન અને ગોઠવણી અને સમન્વયન ભૂલો જેવા ડેટા નુકશાનના સામાન્ય કારણો.

માઈક્રોસોફ્ટ 365, Google Workspace અને Salesforce માટે બેકઅપ અને એન્ડ-ટુ-એન્ડ પ્રોટેક્શન સોલ્યુશન્સનું વિસ્તરણ, આ જોખમોને કારણે ગંભીર ડેટાને થતા નુકસાનથી બચાવવા માટે મૂળ કાર્યક્ષમતામાં અંતર ભરે છે, સમાધાનનું જોખમ ઘટાડે છે અને અંતિમ વપરાશકર્તાઓ અને વ્યવસ્થાપકોને ઝડપથી શોધવામાં સક્ષમ બનાવે છે. માત્ર થોડા ક્લિક્સમાં ડેટાને તેની મૂળ સ્થિતિમાં પુનઃસ્થાપિત કરો.

ક્લાઉડ કમ્પ્યુટિંગ શીખવા માટે શ્રેષ્ઠ સંસાધનો

આજે, ઘણા વ્યાવસાયિકો તેમની કારકિર્દી તરીકે ક્લાઉડ કમ્પ્યુટિંગને અનુસરી રહ્યા છે. આ કારકિર્દીમાં સફળ થવા માટે, તેઓએ તેમની કુશળતા સુધારવા અને ક્લાઉડ કમ્પ્યુટિંગ શીખવા માટે નવીનતમ સમાચારો અને વલણો સાથે અદ્યતન રહેવાની જરૂર છે. ક્લાઉડ કમ્પ્યુટિંગ શીખવા અને ઉદ્યોગમાં નવીનતમ વલણો સાથે સંપર્કમાં રહેવા માટે નીચેના કેટલાક ટોચના સંસાધનો છે:

ઓનલાઈન ક્લાઉડ કોમ્પ્યુટિંગ અભ્યાસક્રમો

ઓનલાઈન અભ્યાસક્રમો ક્લાઉડ શીખવાની શ્રેષ્ઠ રીતોમાંની એક છે કમ્પ્યુટિંગ ઘણા ઓનલાઈન અભ્યાસક્રમો વિડિયો, લેખો, ઇબુક્સ, રિપોર્ટ્સ, હેન્ડ-ઓન લેબ્સ અને શીખનારાઓને મદદ કરવા માટે મૂલ્યાંકન. ઓનલાઈન કેટલાક શ્રેષ્ઠ ક્લાઉડ કમ્પ્યુટિંગ અભ્યાસક્રમો નીચે સૂચિબદ્ધ છે:

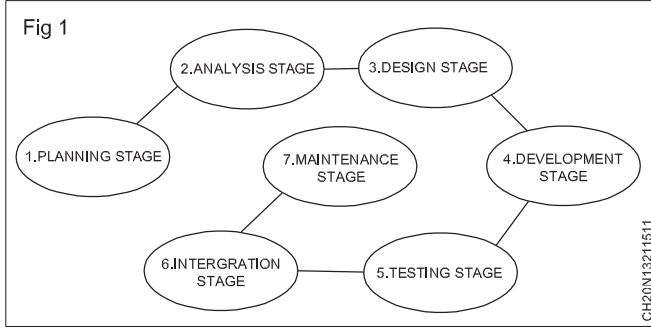
- માટે એમેઝોન AWS પર ક્લાઉડ કમ્પ્યુટિંગનો પરિચય Udemey પર પ્રારંભિક
- કોર્સેરા પર ક્લાઉડ કોમ્પ્યુટિંગનો પરિચય
- ઈન્ટર્નશાલા પર AWS તાલીમ સાથે ક્લાઉડ કોમ્પ્યુટિંગ
- ક્લાઉડ કોમ્પ્યુટિંગ બેઝિક્સ: Udemey પર ક્લાઉડ એન્જિનિયર તરીકે તમારી કારકિર્દીને બહેતર બનાવો
- edX પર Microsoft દ્વારા ક્લાઉડ કોમ્પ્યુટિંગનો પરિચય
- NPTEL દ્વારા ક્લાઉડ કમ્પ્યુટિંગ
- કોર્સેરા પર ક્લાઉડ કોમ્પ્યુટિંગ બેઝિક્સ (ક્લાઉડ 101).
- ફુલ: ક્લાઉડ કમ્પ્યુટિંગ / કોમ્પીઆ ક્લાઉડ પ્રમાણપત્ર. (CVO-002) Udemey પર

એપ્લિકેશન ડેવલપમેન્ટ જીવન ચક્રના તબક્કાઓ ઓળખો (Identify Phases of the Application Development Life Cycle)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- એપ્લિકેશન વિકાસ જીવન ચક્રના તબક્કાઓ ઓળખો
- SDLC ના લાભો.

એપ્લિકેશન વિકાસ જીવન ચક્રના તબક્કાઓ ઓળખો (ફિગ 1)



સિસ્ટમ

સોફ્ટવેર ડેવલપમેન્ટ પ્રક્રિયા સામાન્ય રીતે લાંબી અને કંટાળાજનક હોય છે. પરંતુ પ્રોજેક્ટ મેનેજરો અને સિસ્ટમ વિશ્લેષકો વધુ નિયમિતતા, કાર્યક્ષમતા અને એકંદર ગુણવત્તા સાથે માહિતી પ્રણાલીઓ અથવા સોફ્ટવેર ઉત્પાદનોની રૂપરેખા, ડિઝાઇન, વિકાસ, પરીક્ષણ અને અંતે જમાવટ કરવા માટે સોફ્ટવેર વિકાસ જીવન ચક્રનો લાભ લઈ શકે છે.

સિસ્ટમ વિકાસ જીવન ચક્ર શું છે?

સિસ્ટમ ડેવલપમેન્ટ લાઈફ સાઈકલ અથવા SDLC એ આવશ્યકપણે પ્રોજેક્ટ મેનેજમેન્ટ મોડલ છે. તે વિવિધ તબક્કાઓને વ્યાખ્યાયિત કરે છે જે પ્રોજેક્ટને તેના પ્રારંભિક વિચાર અથવા વિભાવનાથી જમાવટ અને પછીની જાળવણી સુધી લાવવા માટે જરૂરી છે.

સિસ્ટમ વિકાસ જીવન ચક્ર યુએસ માર્ગદર્શિકા (ફિગ 2)

આ માર્ગદર્શિકામાં, અમે તેના તમામ તબક્કાઓ સહિત, સિસ્ટમ વિકાસ જીવન ચક્ર વિશે તમારે જે જાણવાની જરૂર છે તે બધું તોડી પાડીશું. અમે સિસ્ટમ વિશ્લેષકોની ભૂમિકાઓ અને તમારા પ્રોજેક્ટને SDLC અપનાવવાથી જોઈ શકે તેવા લાભો પર પણ જઈશું.

સિસ્ટમ વિકાસ જીવન ચક્રના 7 તબક્કા



આધુનિક સિસ્ટમ વિકાસ જીવન ચક્રના સાત પ્રાથમિક તબક્કાઓ છે. અહીં સંક્ષિપ્ત વિરામ છે:

- આયોજન સ્ટેજ
- પૃથ્થકરણ તબક્કાની શક્યતા અથવા જરૂરિયાતો

- ડિઝાઇન અને પ્રોટોટાઇપિંગ સ્ટેજ
- સોફ્ટવેર ડેવલપમેન્ટ સ્ટેજ
- સોફ્ટવેર પરીક્ષણ સ્ટેજ
- અમલીકરણ અને એકીકરણ
- ઓપરેશન્સ અને મેન્ટેનન્સ સ્ટેજ

હવે ચાલો દરેક તબક્કાને વ્યક્તિગત રીતે નજીકથી જોઈએ.

આયોજન સ્ટેજ

અમે પ્લાનિંગ સ્ટેજથી પણ શરૂઆત કરીએ તે પહેલાં, અમે તમને સમય કાઢીને એપ ડેવલપમેન્ટ લાઈફ સાઈકલની યોગ્ય સમજ મેળવવાની શ્રેષ્ઠ ટીપ આપી શકીએ છીએ.

આયોજનનો તબક્કો (જેને સંભવિતતાનો તબક્કો પણ કહેવાય છે) તે જેવો લાગે છે તે બરાબર છે: તે તબક્કો જેમાં વિકાસકર્તાઓ આગામી પ્રોજેક્ટ માટે આયોજન કરશે.

તે કોઈપણ હાલની સિસ્ટમોની સમસ્યા અને અવકાશને વ્યાખ્યાયિત કરવામાં મદદ કરે છે, તેમજ તેમની નવી સિસ્ટમો માટેના હેતુઓ નક્કી કરવામાં મદદ કરે છે.

આગામી વિકાસ ચક્ર માટે અસરકારક રૂપરેખા વિકસાવીને, તેઓ વિકાસને અસર કરે તે પહેલાં સૈદ્ધાંતિક રીતે સમસ્યાઓને પકડી લેશે.

અને તેમની યોજનાને સાકાર કરવા માટે જરૂરી ભંડોળ અને સંસાધનોને સુરક્ષિત કરવામાં મદદ કરે છે.

કદાચ સૌથી અગત્યનું, આયોજન તબક્કો પ્રોજેક્ટ શેડ્યૂલ સેટ કરે છે, જે મુખ્ય મહત્વનો હોઈ શકે છે જો વિકાસ કોઈ વ્યાવસાયિક ઉત્પાદન માટે હોય કે જે ચોક્કસ સમય સુધીમાં બજારમાં મોકલવામાં આવે.

વિશ્લેષણ સ્ટેજ

વિશ્લેષણના તબક્કામાં નવી સિસ્ટમ માટે જરૂરી તમામ ચોક્કસ વિગતો ભેગી કરવી તેમજ પ્રોટોટાઇપ માટેના પ્રથમ વિચારો નક્કી કરવાનો સમાવેશ થાય છે.

વિકાસકર્તાઓ કરી શકે છે:

- કોઈપણ પ્રોટોટાઇપ સિસ્ટમ આવશ્યકતાઓને વ્યાખ્યાયિત કરો
- હાલના પ્રોટોટાઇપ્સના વિકલ્પોનું મૂલ્યાંકન કરો
- અંતિમ વપરાશકારોની જરૂરિયાતો નક્કી કરવા માટે સંશોધન અને વિશ્લેષણ કરો

વધુમાં, વિકાસકર્તાઓ ઘણીવાર સોફ્ટવેર જરૂરિયાત સ્પષ્ટીકરણ અથવા SRS દસ્તાવેજ બનાવશે.

આમાં તેઓ જે સિસ્ટમ બનાવવાની યોજના ધરાવે છે તેના માટે સૉફ્ટવેર, હાર્ડવેર અને નેટવર્ક આવશ્યકતાઓ માટેની તમામ વિશિષ્ટતાઓનો સમાવેશ કરે છે. આ તેમને અન્ય વિકાસ ટીમોની જેમ તે જ સ્થળે કામ કરતી વખતે ભંડોળ અથવા સંસાધનોને ઓવરડ્રો કરવાથી અટકાવશે.

ડિઝાઇન સ્ટેજ

ડિઝાઇન સ્ટેજ મુખ્ય ડેવલપર સ્ટેજ માટે જરૂરી પુરોગામી છે.

વિકાસકર્તાઓ પ્રથમ એકંદર એપ્લિકેશન માટેની વિગતોની રૂપરેખા કરશે, ચોક્કસ પાસાઓ સાથે, જેમ કે તેના:

- વપરાશકર્તા ઈન્ટરફેસ
- સિસ્ટમ ઈન્ટરફેસ
- નેટવર્ક અને નેટવર્ક જરૂરિયાતો
- ડેટાબેસેસ

તેઓ સામાન્ય રીતે તેમણે બનાવેલા SRS દસ્તાવેજને વધુ તાર્કિક માળખામાં ફેરવશે જે પછીથી પ્રોગ્રામિંગ ભાષામાં અમલમાં મૂકી શકાય છે. ઓપરેશન, તાલીમ અને જાળવણી યોજનાઓ તમામ તૈયાર કરવામાં આવશે જેથી વિકાસકર્તાઓને ખબર પડે કે ચક્રના દરેક તબક્કામાં તેમને શું કરવાની જરૂર છે.

એકવાર પૂર્ણ થઈ ગયા પછી, વિકાસ સંચાલકો SDLC ના આગળના તબક્કામાં સંદર્ભિત કરવા માટે ડિઝાઇન દસ્તાવેજ તૈયાર કરશે.

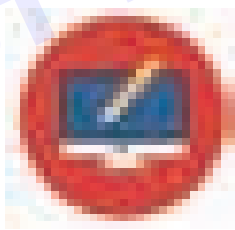
વિકાસ તબક્કો (ફિગ 3)

ડેવલપમેન્ટ સ્ટેજ એ ભાગ છે જ્યાં વિકાસકર્તાઓ વાસ્તવમાં કોડ લખે છે અને અગાઉના ડિઝાઇન દસ્તાવેજો અને દર્શાવેલ સ્પષ્ટીકરણો અનુસાર એપ્લિકેશન બનાવે છે.

આ તે છે જ્યાં સ્ટેટિક એપ્લિકેશન સિક્યુરિટી ટેસ્ટિંગ અથવા SAST ટૂલ્સ અમલમાં આવે છે.

પ્રોડક્ટ પ્રોગ્રામ કોડ ડિઝાઇન દસ્તાવેજના સ્પષ્ટીકરણો અનુસાર બનાવવામાં આવે છે. સૈદ્ધાંતિક રીતે, અગાઉના આયોજન અને દર્શાવેલ તમામ વાસ્તવિક વિકાસના તબક્કાને પ્રમાણમાં સીધો બનાવવો જોઈએ.

Fig 3



વિકાસકર્તાઓ સંસ્થા દ્વારા નિર્ધારિત કોઈપણ કોડિંગ માર્ગદર્શિકાનું પાલન કરશે અને કમ્પાઈલર્સ, ડીબગર્સ અને દુભાષિયા જેવા વિવિધ સાધનોનો ઉપયોગ કરશે.

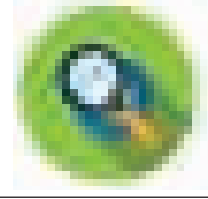
પ્રોગ્રામિંગ ભાષાઓમાં C++, PHP અને વધુ જેવા સ્ટેપલ્સનો સમાવેશ થઈ શકે છે. વિકાસકર્તાઓ પ્રોજેક્ટ સ્પષ્ટીકરણો અને જરૂરિયાતોને આધારે ઉપયોગ કરવા માટે યોગ્ય પ્રોગ્રામિંગ કોડ પસંદ કરશે.

પરીક્ષણ સ્ટેજ (ફિગ 4)

બિલ્ડિંગ સૉફ્ટવેર અંત નથી.

હવે તે ખાતરી કરવા માટે પરીક્ષણ કરવું આવશ્યક છે કે તેમાં કોઈ ભૂલો નથી અને અંતિમ-વપરાશકર્તાના અનુભવને કોઈપણ સમયે નકારાત્મક અસર થશે નહીં.

Fig 4



પરીક્ષણના તબક્કા દરમિયાન, વિકાસકર્તાઓ તેમના સૉફ્ટવેરને ઝીણા દાંતાવાળા કાંસકો સાથે જોશે, જેમાં કોઈપણ ભૂલો અથવા ખામીઓ નોંધવામાં આવશે જેને ટ્રેક કરવાની, ઠીક કરવાની અને પછીથી ફરીથી પરીક્ષણ કરવાની જરૂર છે.

તે મહત્વનું છે કે સૉફ્ટવેર એકંદરે ગુણવત્તાના ધોરણોને પૂર્ણ કરે છે જે અગાઉ SRS દસ્તાવેજમાં વ્યાખ્યાયિત કરવામાં આવ્યા હતા.

વિકાસકર્તાઓની કૌશલ્ય, સૉફ્ટવેરની જટિલતા અને અંતિમ-વપરાશકર્તા માટેની આવશ્યકતાઓના આધારે, પરીક્ષણ કાં તો અત્યંત નાનો તબક્કો હોઈ શકે છે અથવા ઘણો લાંબો સમય લઈ શકે છે. વધુ માહિતી માટે સૉફ્ટવેર પરીક્ષણ પ્રોજેક્ટ્સ માટેની અમારી ટોચની 10 શ્રેષ્ઠ પદ્ધતિઓ પર એક નજર નાખો.

અમલીકરણ અને એકીકરણ સ્ટેજ

પરીક્ષણ કર્યા પછી, સૉફ્ટવેર માટે એકંદર ડિઝાઇન એકસાથે આવશે. સામાન્ય રીતે વધુ ભૂલો અથવા ખામીઓ શોધવા માટે તાલીમ વાતાવરણનો લાભ લઈને વિકાસકર્તાના પ્રયત્નો દ્વારા વિવિધ મોડ્યુલો અથવા ડિઝાઇનને પ્રાથમિક સ્ત્રોત કોડમાં એકીકૃત કરવામાં આવશે.

માહિતી સિસ્ટમ તેના પર્યાવરણમાં સંકલિત કરવામાં આવશે અને આખરે ઈન્સ્ટોલ કરવામાં આવશે. આ તબક્કો પસાર કર્યા પછી, સૉફ્ટવેર સૈદ્ધાંતિક રીતે બજાર માટે તૈયાર છે અને કોઈપણ અંતિમ વપરાશકર્તાઓને પ્રદાન કરી શકાય છે.

જાળવણી સ્ટેજ (ફિગ 5)

જ્યારે સૉફ્ટવેર બજારમાં પહોંચે છે ત્યારે SDLC સમાપ્ત થવું નથી. ડેવલપર્સ હવે મેન્ટેનન્સ મોડમાં જવું જોઈએ અને અંતિમ વપરાશકર્તાઓ દ્વારા જાણ કરવામાં આવેલી સમસ્યાઓને હેન્ડલ કરવા માટે જરૂરી કોઈપણ પ્રવૃત્તિઓની પ્રેક્ટિસ કરવાનું શરૂ કરવું જોઈએ.

વધુમાં, વિકાસકર્તાઓ કોઈપણ ફેરફારોને અમલમાં મૂકવા માટે જવાબદાર છે જે સૉફ્ટવેરને જમાવટ પછી જરૂર પડી શકે છે.

આમાં અવશેષ બગસને હેન્ડલ કરવાનો સમાવેશ થઈ શકે છે જે લૉન્ચ પહેલાં પેચ કરવામાં સક્ષમ ન હતા અથવા વપરાશકર્તાના અહેવાલોને કારણે ઉદભવતી નવી સમસ્યાઓનું નિરાકરણ શામેલ હોઈ શકે છે. નાની સિસ્ટમોની સરખામણીમાં મોટી સિસ્ટમોને લાંબા સમય સુધી જાળવણીના તબક્કાની જરૂર પડી શકે છે.

સિસ્ટમ એનાલિસ્ટની ભૂમિકા

SDLC ના સિસ્ટમ વિશ્લેષક, અમુક રીતે, સમગ્ર સિસ્ટમ માટે નિરીક્ષક છે. તેઓ સિસ્ટમ અને તેના તમામ ગતિશીલ ભાગોથી સંપૂર્ણપણે વાકેફ હોવા જોઈએ અને યોગ્ય દિશાઓ આપીને પ્રોજેક્ટને માર્ગદર્શન આપવામાં મદદ કરી શકે છે.

સિસ્ટમ વિશ્લેષક હોવું જોઈએ:

- પ્રોજેક્ટ માટે જરૂરી કોઈપણ તકનીકી કૌશલ્યમાં નિષ્ણાત
- પોતાની ટીમને સફળતા માટે કમાન્ડ કરવામાં મદદ કરવા માટે સારો કોમ્યુનિકેટર

Fig 5



- એક સારો આયોજક જેથી વિકાસ ચક્રના દરેક તબક્કામાં વિકાસ કાર્યો સમયસર થઈ શકે

આમ, સિસ્ટમ વિશ્લેષકો પાસે આંતરવ્યક્તિત્વ, તકનીકી, સંચાલન અને વિશ્લેષણાત્મક કૌશલ્યોનું એકસાથે મિશ્રણ હોવું જોઈએ. તેઓ બહુમુખી વ્યાવસાયિકો છે જે SDLC બનાવી અથવા તોડી શકે છે.

આપેલ પ્રોજેક્ટની અંતિમ સફળતા માટે તેમની જવાબદારીઓ તદ્દન વૈવિધ્યસભર અને મહત્વપૂર્ણ છે. સિસ્ટમ વિશ્લેષકો પાસે ઘણી વખત અપેક્ષા રાખવામાં આવશે:

- તથ્યો અને માહિતી એકત્ર કરો
- કઈ ભૂલોને પ્રાથમિકતા આપવી અથવા કઈ સુવિધાઓ કાપવી તે અંગે આદેશ નિર્ણયો લો
- વૈકલ્પિક ઉકેલો સૂચવો
- સ્પષ્ટીકરણો દોરો જે વપરાશકર્તાઓ અને પ્રોગ્રામરો બંને દ્વારા સરળતાથી સમજી શકાય
- પછીના એકીકરણ માટે મોડ્યુલરિટી જાળવી રાખતી વખતે લોજિકલ સિસ્ટમ્સ લાગુ કરો
- પ્રોજેક્ટ ધ્યેયો દ્વારા આવશ્યકતા મુજબ પરિણામી સિસ્ટમનું મૂલ્યાંકન અને ફેરફાર કરવામાં સક્ષમ બનો
- વપરાશકર્તાની જરૂરિયાતોને વ્યાખ્યાયિત કરીને અને સમજીને પ્રોજેક્ટની જરૂરિયાતો અને લક્ષ્યોની યોજના બનાવવામાં મદદ કરો

6 મૂળભૂત SDLC પદ્ધતિઓ

જો કે સિસ્ટમ ડેવલપમેન્ટ લાઈફ સાઈકલ વ્યાપક અર્થમાં પ્રોજેક્ટ મેનેજમેન્ટ મોડલ છે, છ વધુ ચોક્કસ પદ્ધતિઓનો ચોક્કસ પરિણામો પ્રાપ્ત કરવા અથવા વિવિધ વિશેષતાઓ સાથે વધુ SDLC પ્રદાન કરવા માટે લાભ લઈ શકાય છે.

વોટરફોલ મોડલ (ફિગ 6)

તમામ SDLC પદ્ધતિમાં વોટરફોલ મોડલ સૌથી જૂનું છે. તે રેખીય અને સીધું છે અને વિકાસ ટીમોને પ્રોજેક્ટનો એક તબક્કો બીજા તબક્કામાં જતા પહેલા પૂર્ણપણે સમાપ્ત કરવાની જરૂર છે.

દરેક તબક્કામાં એક અલગ પ્રોજેક્ટ પ્લાન હોય છે અને સમાન સમસ્યાઓને ટાળવા માટે અગાઉના તબક્કામાંથી માહિતી લે છે (જો આવી હોય તો). જો કે, તે પ્રારંભિક વિલંબ માટે સંવેદનશીલ છે અને વિકાસ ટીમો માટે પાછળથી રસ્તા પર ઊભી થતી મોટી સમસ્યાઓ તરફ દોરી શકે છે.

પુનરાવર્તિત મોડલ (ફિગ 7)

પુનરાવર્તિત મોડેલ પુનરાવર્તન અને પુનરાવર્તન પરીક્ષણ પર ધ્યાન કેન્દ્રિત કરે છે. સોફ્ટવેર પ્રોજેક્ટના નવા સંસ્કરણો સંભવિત ભૂલોને પકડવા માટે દરેક તબક્કાના અંતે ઉત્પન્ન થાય છે અને વિકાસકર્તાઓને બજાર માટે તૈયાર થાય ત્યાં સુધી અંતિમ ઉત્પાદનમાં સતત સુધારો કરવાની મંજૂરી આપે છે.

Fig 6



Fig 7



આ મોડલનો એક ફાયદો એ છે કે વિકાસકર્તાઓ તેમના વિકાસના જીવન ચક્રમાં પ્રમાણમાં વહેલી તકે પ્રોજેક્ટનું વર્કિંગ વર્ઝન બનાવી શકે છે, તેથી ફેરફારોને અમલમાં મૂકવો ઘણીવાર ઓછા ખર્ચાળ હોય છે.

સર્પાકાર મોડલ

સર્પાકાર મોડેલો અન્ય પદ્ધતિઓની તુલનામાં લવચીક છે. રૂપકાત્મક રીતે સર્પાકાર ગતિમાં પ્રોજેક્ટ્સ ચાર મુખ્ય તબક્કાઓમાંથી વારંવાર પસાર થાય છે.

તે મોટા પ્રોજેક્ટ્સ માટે ફાયદાકારક છે કારણ કે વિકાસ ટીમો ખૂબ જ કસ્ટમાઈઝ્ડ પ્રોડક્ટ્સ બનાવી શકે છે અને જીવન ચક્રની શરૂઆતમાં કોઈપણ પ્રાપ્ત પ્રતિસાદનો સમાવેશ કરી શકે છે.

વી-મોડલ

વી-મોડલ (જે ચકાસણી અને માન્યતા માટે ટૂંકું છે) વોટરફોલ મોડલ જેવું જ છે. સંભવિત ભૂલો અને ખામીઓને પકડવા માટે દરેક વિકાસના તબક્કામાં પરીક્ષણ તબક્કાનો સમાવેશ કરવામાં આવે છે.

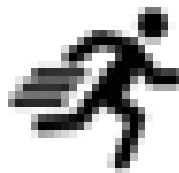
તે અતિ શિસ્તબદ્ધ છે અને સખત સમયરેખાની જરૂર છે. પરંતુ સૈદ્ધાંતિક રીતે, તે મોટી ભૂલોને નિયંત્રણની બહાર જતા અટકાવીને મુખ્ય વોટરફોલ મોડેલની ખામીઓને પ્રકાશિત કરે છે.

બિગ બેંગ મોડલ

બિગ બેંગ મોડલ અવિશ્વસનીય રીતે લવચીક છે અને સખત પ્રક્રિયા અથવા પ્રક્રિયાને અનુસરતું નથી. તે વિગતવાર આયોજન પાછળ પણ છોડી દે છે. તેનો ઉપયોગ મોટાભાગે વ્યાપક વિચારો વિકસાવવા માટે થાય છે જ્યારે ગ્રાહક અથવા ક્લાયન્ટને ખાતરી હોતી નથી કે તેઓ શું ઇચ્છે છે. વિકાસકર્તાઓ ફક્ત પૈસા અને સંસાધનો સાથે પ્રોજેક્ટ શરૂ કરે છે.

તેમનું આઉટપુટ ક્લાયન્ટને આખરે તેઓ ઇચ્છે છે તેની અનુભૂતિ કરતાં નજીક અથવા દૂર હોઈ શકે છે. તે મોટાભાગે નાના પ્રોજેક્ટ્સ અને પ્રાયોગિક જીવન ચક્ર માટે વપરાય છે જે સમાન કંપનીમાં અન્ય પ્રોજેક્ટ્સને જાણ કરવા માટે રચાયેલ છે. (ફિગ 8)

Fig 8



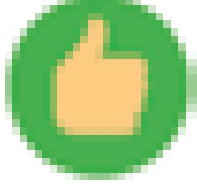
ચપળ મોડલ

ચપળ મોડેલ પ્રમાણમાં જાણીતું છે, ખાસ કરીને સોફ્ટવેર વિકાસ ઉદ્યોગમાં.

ચપળ પદ્ધતિ પ્રકાશન વચ્ચે નાના પરંતુ વધારાના ફેરફારોનો ઉપયોગ કરીને, ઝડપી અને ચાલુ પ્રકાશન ચક્રને પ્રાથમિકતા આપે છે. આના પરિણામે અન્ય મોડલની તુલનામાં વધુ પુનરાવર્તનો અને ઘણા વધુ પરીક્ષણો થાય છે.

સૈદ્ધાંતિક રીતે, આ મોડેલ ટીમોને પ્રોજેક્ટના વધુ જટિલ તબક્કાઓ સુધી, તેમને ગુમ કરવાને બદલે ઉદભવતી નાની સમસ્યાઓને ઉકેલવામાં મદદ કરે છે. (ફિગ 9)

Fig 9



SDLC ના લાભો : SDLC વિકાસ ટીમોને ઘણા ફાયદા પ્રદાન કરે છે જે તેને યોગ્ય રીતે અમલમાં મૂકે છે.

ધ્યેયના વર્ણનો સાફ કરો : વિકાસકર્તાઓ સ્પષ્ટપણે જાણે છે કે તેઓને જે ધ્યેયો પૂરા કરવાની જરૂર છે અને ડિલિવેરેબલ્સ તેઓ એક સેટ સમયરેખા દ્વારા હાંસલ કરવા જોઈએ, સમય અને સંસાધનોનો વ્યય થવાનું જોખમ ઘટાડે છે.

ઈન્સ્ટોલેશન પહેલાં યોગ્ય પરીક્ષણ : SDLC મોડલ્સ વધુ સોર્સ કોડમાં ઈન્સ્ટોલ થાય તે પહેલાં તમામ સોફ્ટવેરનું પરીક્ષણ કરવામાં આવે તેની ખાતરી કરવા માટે ચેક અને બેલેન્સ લાગુ કરે છે.

ક્લિયર સ્ટેજ પ્રોગ્રેસન : ડેવલપર્સ આગલી ઉંમરમાં આગળ વધી શકતા નથી જ્યાં સુધી પહેલાની પ્રક્રિયા પૂર્ણ ન થાય અને મેનેજર દ્વારા સાઈન ઓફ ન થાય.

સભ્ય સુગમતા : SDLC પાસે પ્રોજેક્ટના લક્ષ્યો અને પદ્ધતિઓ માટે સારી રીતે સંરચિત દસ્તાવેજો હોવાથી, ટીમના સભ્યો પ્રમાણમાં પીડારહિત રીતે છોડી શકે છે અને નવા સભ્યો દ્વારા બદલી શકાય છે.

સંપૂર્ણતા પ્રાપ્ત કરી શકાય છે : બધા SDLC તબક્કાઓ એકબીજામાં પાછા ફીડ કરવા માટે છે. તેથી SDLC મોડલ્સ આવશ્યકપણે સંપૂર્ણ ન થાય ત્યાં સુધી પ્રોજેક્ટ્સને પુનરાવર્તિત કરવામાં અને પોતાને સુધારવામાં મદદ કરી શકે છે.

કોઈ એક સભ્ય પ્રોજેક્ટ બનાવતો નથી અથવા તોડતો નથી

ફરીથી, કારણ કે SDLC વ્યાપક કાગળ અને માર્ગદર્શિકા દસ્તાવેજોનો ઉપયોગ કરે છે, તે એક ટીમ પ્રયાસ છે અને એક પણ મોટા સભ્યને ગુમાવવાથી પ્રોજેક્ટની સમયરેખા જોખમમાં આવશે નહીં.

સિસ્ટમ વિકાસ જીવન ચક્ર વિશે તમારું જુઓ જાણવાની જરૂર છે

SDLC ક્યાં વપરાય છે?

આઈટી પ્રોજેક્ટ્સ વિકસાવતી વખતે સિસ્ટમ ડેવલપમેન્ટ લાઈફ સાઈકલનો ઉપયોગ સામાન્ય રીતે થાય છે.

સોફ્ટવેર ડેવલપમેન્ટ મેનેજર વિવિધ વિકાસ તબક્કાઓની રૂપરેખા બનાવવા માટે SDLC નો ઉપયોગ કરશે, ખાતરી કરો કે દરેક જણ સમયસર અને યોગ્ય ક્રમમાં તબક્કાઓ પૂર્ણ કરે છે અને પ્રોજેક્ટ શક્ય તેટલી ઝડપથી અને બગ-ફ્રી વિતરિત થાય છે.

SDLC નો ઉપયોગ સિસ્ટમ વિશ્લેષકો દ્વારા પણ વધુ વિશિષ્ટ રીતે કરી શકાય છે કારણ કે તેઓ નવી માહિતી સિસ્ટમ વિકસાવે છે અને પછીથી અમલમાં મૂકે છે.

કયું SDLC મોડલ શ્રેષ્ઠ છે?

તે મોટાભાગે તમારી ટીમના ધ્યેયો અને સંસાધનની જરૂરિયાતો પર આધાર રાખે છે.

મોટાભાગની IT વિકાસ ટીમો તેમના SDLC માટે ચપળ પદ્ધતિનો ઉપયોગ કરે છે. જો કે, અન્યો પુનરાવર્તિત અથવા સર્પાકાર પદ્ધતિઓ પસંદ કરી શકે છે.

આ ત્રણેય પદ્ધતિઓ લોકપ્રિય છે કારણ કે તે ઉત્પાદનને વધુ સ્ત્રોત કોડ સાથે સંકલિત કરવામાં આવે અથવા બજારમાં પહોંચાડવામાં આવે તે પહેલાં તે વ્યાપક પુનરાવર્તન અને બગ પરીક્ષણ માટે પરવાનગી આપે છે.

DevOps પદ્ધતિઓ પણ લોકપ્રિય પસંદગીઓ છે. અને જો તમને ક્યારેય DevOps શું છે તેના પર રિફ્રેશર કોર્સની જરૂર હોય, તો તમારે ચિંતા કરવાની જરૂર નથી કારણ કે CloudDefense ખાતેની અમારી ટીમે તમને આવરી લીધા છે!

SDLC શું વિકાસ કરે છે?

SDLC નો ઉપયોગ સોફ્ટવેર, સિસ્ટમ્સ અને માહિતી પ્રણાલીઓને વિકસાવવા અથવા એન્જિનિયર કરવા માટે થઈ શકે છે. તેનો ઉપયોગ હાર્ડવેર વિકસાવવા માટે અથવા એક જ સમયે સોફ્ટવેર અને હાર્ડવેર બંનેના સંયોજન માટે પણ થઈ શકે છે.

FAQs

સિસ્ટમ વિકાસ જીવન ચક્રના 5 મૂળ તબક્કાઓ શું હતા?

સિસ્ટમ્સ ડેવલપમેન્ટ લાઈફ સાઈકલમાં મૂળ રીતે સાતને બદલે પાંચ તબક્કાનો સમાવેશ થતો હતો. તેમાં આયોજન, નિર્માણ, વિકાસ, પરીક્ષણ અને જમાવટનો સમાવેશ થાય છે. નોંધ કરો કે તે વિશ્લેષણ અને જાળવણીના મુખ્ય તબક્કાઓને છોડી દે છે.

SDLC ના 7 તબક્કાઓ શું છે?

SDLCના નવા સાત તબક્કાઓમાં આયોજન, વિશ્લેષણ, ડિઝાઈન, વિકાસ, પરીક્ષણ, અમલીકરણ અને જાળવણીનો સમાવેશ થાય છે.

MIS માં સિસ્ટમ વિકાસ જીવન ચક્ર શું છે?

મેનેજમેન્ટ ઈન્ફર્મેશન સિસ્ટમ્સ અથવા MISના વધુ સંદર્ભમાં, SDLC મેનેજરને લક્ષ્ય લક્ષ્યોને પૂરા કરવા માટે માહિતી સિસ્ટમોની રચના, વિકાસ, પરીક્ષણ અને જમાવટ કરવામાં મદદ કરે છે.

નિષ્કર્ષ

આખરે, IT અને અન્ય ઉદ્યોગો બંનેમાં કોઈપણ વિકાસ ટીમ તેમના પ્રોજેક્ટ્સમાં સિસ્ટમ ડેવલપમેન્ટ લાઈફ સાઈકલ લાગુ કરવાથી લાભ મેળવી શકે છે. શ્રેષ્ઠ પરિણામો માટે તમે તમારા SDLC સાથે જોડાણમાં કઈ પદ્ધતિનો ઉપયોગ કરવા માંગો છો તે ઓળખવા માટે ઉપરોક્ત માર્ગદર્શિકાનો ઉપયોગ કરો.

પ્રોગ્રામિંગ ભાષા (પાયથોન) કમાન્ડ લાઇનમાંથી પાયથોનનો ઉપયોગ કરો (Programming language (Python) Use Python from command line)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- પાયથોનનો પરિચય
- Python ના લક્ષણો
- પાયથોનની ટિપ્પણીઓ
- ચલ.

પાયથોન એ 1980 ના દાયકાના અંતમાં નેધરલેન્ડ્સમાં ગણિત અને કોમ્પ્યુટર સાયન્સ માટે નેશનલ રિસર્ચ ઇન્સ્ટિટ્યૂટ ખાતે ગાઈડો વાન રોસમ દ્વારા વિકસિત સામાન્ય હેતુ, ઉચ્ચ સ્તરીય, અર્થઘટન કરાયેલ પ્રોગ્રામિંગ ભાષા છે.

પાયથોન એ કન્સોલ આધારિત, GUI આધારિત, વેબ પ્રોગ્રામિંગ અને ડેટા વિશ્લેષણ સહિતના કાર્યોના સમૂહ માટે ઉપયોગમાં લેવાતી સૌથી લોકપ્રિય અને વ્યાપકપણે ઉપયોગમાં લેવાતી પ્રોગ્રામિંગ ભાષા છે.

પાયથોન એ શીખવામાં સરળ અને સરળ પ્રોગ્રામિંગ ભાષા છે તેથી જો તમે પ્રોગ્રામિંગ માટે નવા હોવ તો પણ, તમે કોઈપણ સમસ્યાનો સામનો કર્યા વિના અજગર શીખી શકો છો.

હકીકત : પાયથોનનું નામ કોમેડી ટેલિવિઝન શો મોન્ટી પાયથોન્સ ફ્લાઈંગ સર્કસ પરથી રાખવામાં આવ્યું છે.

- પાયથોન એપ્લિકેશન્સ
- પાયથોનનું સ્થાપન

પાયથોનની વિશેષતાઓ

પાયથોન ઘણી બધી સુવિધાઓ પ્રદાન કરે છે જે નીચે સૂચિબદ્ધ છે.

- અન્ય પ્રોગ્રામિંગ ભાષાઓની

સરખામણીમાં પાયથોન શીખવા અને વાપરવા માટે સરળ છે. તે વિકાસકર્તા-મૈત્રીપૂર્ણ અને ઉચ્ચ સ્તરીય પ્રોગ્રામિંગ ભાષા છે.

- ઈન્ટરપ્રેટર લેંગ્વેજ

પાયથોન એ ઈન્ટરપ્રેટર લેંગ્વેજ છે કારણ કે સંકલનની જરૂર નથી. આ ડિબગીંગને સરળ બનાવે છે અને તેથી નવા નિશાળીયા માટે યોગ્ય છે.

- ક્રોસ-પ્લેટફોર્મ લેંગ્વેજ

પાયથોન વિન્ડોઝ, લિનક્સ, યુનિક્સ અને મેકિન્ટોશ વગેરે જેવા વિવિધ પ્લેટફોર્મ પર સમાન રીતે ચાલી શકે છે. તેથી, આપણે કદી શકીએ કે પાયથોન એક પોર્ટેબલ ભાષા છે.

- ફ્રી અને ઓપન

સોર્સ ધ પાયથોન ઈન્ટરપ્રીટર ઓપન સોર્સ લાયસન્સ હેઠળ વિકસાવવામાં આવ્યું છે, જે તેને ઈન્સ્ટોલ, ઉપયોગ અને વિતરણ માટે મફત બનાવે છે.

- ઓબ્જેક્ટ-ઓરિએ

ન્ટેડ લેંગ્વેજ પાયથોન ઓબ્જેક્ટ ઓરિએન્ટેડ લેંગ્વેજને સપોર્ટ કરે છે અને વર્ગો અને ઓબ્જેક્ટના ખ્યાલો અસ્તિત્વમાં આવે છે.

- GUI પ્રોગ્રામિંગ સપોર્ટ

પાયથોનનો ઉપયોગ કરીને ગ્રાફિકલ યુઝર ઈન્ટરફેસ વિકસાવી શકાય છે.

- એકીકૃત તેને

C, C++ અને JAVA વગેરે જેવી ભાષાઓ સાથે સરળતાથી સંકલિત કરી શકાય છે.

પાયથોન માં ટિપ્પણીઓ

સામાન્ય રીતે, પ્રોગ્રામનું વર્ણન કરવા અથવા કોડના અમુક ભાગને દુભાષિયાથી છુપાવવા માટે પ્રોગ્રામિંગ ભાષામાં ટિપ્પણીઓનો ઉપયોગ કરવામાં આવે છે.

પાયથોનમાં ટિપ્પણીઓનો ઉપયોગ કોઈપણ પ્રોગ્રામ કોડને સમજાવવા માટે થઈ શકે છે. તેનો ઉપયોગ કોડને છુપાવવા માટે પણ થઈ શકે છે.

ટિપ્પણી એ પ્રોગ્રામનો એક ભાગ નથી, પરંતુ તે પ્રોગ્રામની ક્રિયાપ્રતિક્રિયાને વધારે છે અને પ્રોગ્રામને વાંચવા યોગ્ય બનાવે છે.

પાયથોન બે પ્રકારની ટિપ્પણીઓને સમર્થન આપે છે:

- સિંગલ લાઇન ટિપ્પણી
- મલ્ટી લાઇન ટિપ્પણી

1 સિંગલ લાઇન ટિપ્પણી

જો યુઝર એક લીટીની કોમેન્ટનો ઉલ્લેખ કરવા માંગે છે, તો કોમેન્ટ ‘#’ થી શરૂ થવી જોઈએ.

ફોર્મેટ:

```
# This is single line comment
```

ઉદાહરણ: "scomment.py"

```
# This is single line comment.
print("Hello Python")
```

આઉટપુટ: \$python3 scomment.py

```
Hello Python
```

2 મલ્ટી લાઇન ટિપ્પણી

ટ્રિપલ અવતરણની અંદર બહુ-રેખાવાળી ટિપ્પણી આપી શકાય છે. લાઇનની શરૂઆતમાં શરૂ થવી જોઈએ.

ફોર્મેટ:

```
""" This
```


Is

Multiline comment”

ઉદાહરણ:”mcomment.py”

““ This

is

Multi line comment”

print(“Hello Python”)

આઉટપુટ:\$python3 mcomment.py

Hello Python

ઉદાહરણ:”comments.py”

આ ઉદાહરણ ટિપ્પણીઓનો ઉપયોગ દર્શાવે છે

...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)

આઉટપુટ:\$python3 comments.py

Welcome to Python

Sum is

50

પાયથોન માં ચલો

વેરીએબલ એ નામનું મેમરી લોકેશન છે જેમાં આપણે ચોક્કસ પ્રોગ્રામ માટે વેલ્યુ સ્ટોર કરી શકીએ છીએ. બીજા શબ્દોમાં કહીએ તો, વેરીએબલ એ નામ છે જેનો ઉપયોગ મેમરી લોકેશનનો સંદર્ભ આપવા માટે થાય છે. ચલ ઓળખકર્તા તરીકે પણ ઓળખાય છે અને મૂલ્ય રાખવા માટે વપરાય છે.

ચલો બનાવી રહ્યા છે

પાયથોનમાં, અમારે પાયથોનમાં સ્પષ્ટ રીતે વેરીએબલ જાહેર કરવાની જરૂર નથી. જ્યારે આપણે વેરીએબલને કોઈપણ મૂલ્ય અસાઈન કરીએ છીએ ત્યારે તે વેરીએબલ આપમેળે જાહેર થાય છે.

પાયથોનમાં, અમારે વેરીએબલનો પ્રકાર સ્પષ્ટ કરવાની જરૂર નથી કારણ કે પાયથોન એ ઢીલી રીતે લખેલી ભાષા છે. એટલે કે ઢીલી રીતે ટાઈપ કરેલી ભાષામાં વેરીએબલના પ્રકારને સ્પષ્ટ કરવાની જરૂર નથી કારણ કે વેરીએબલ આપમેળે સોપેલ મૂલ્યના આધારે તેના ડેટાટાઈપમાં ફેરફાર કરે છે.

દાખલા તરીકે:

?? a=10???જ્યાં ચલ a એ પૂર્ણાંક ડેટાટાઈપ છે.

?? b=”ગ્લાન્સ”???જ્યાં ચલ b એ સ્ટ્રિંગ ડેટાટાઈપ છે.

ચલના નામકરણ માટેના નિયમો :

- વેરિએબલ નામો બંને અક્ષરો અને અંકોનું જૂથ હોઈ શકે છે, પરંતુ તે અક્ષર અથવા અન્ડરસ્કોરથી શરૂ થવું જોઈએ.
- ચલ નામ માટે લોઅરકેસ અક્ષરોનો ઉપયોગ કરવાની ભલામણ કરવામાં આવે છે. ‘SUM’ અને ‘sum’ બંને બે અલગ અલગ ચલ છે.

ઉદાહરણ:”vardemo.py”

a=10 #integer

b=”StudyGlance” #string

c=12.5 #float

print(a)

print(b)

print(c)

આઉટપુટ:\$python3 vardemo.py

10

StudyGlance

12.5

બહુવિધ ચલોને મૂલ્ય સોંપો

પાયથોન આપણને એક જ સ્ટેટમેન્ટમાં બહુવિધ વેરીએબલ્સને વેલ્યુ અને બહુવિધ વેરીએબલ્સને બહુવિધ વેલ્યુ અસાઈન કરવાની પરવાનગી આપે છે જેને બહુવિધ અસાઈનમેન્ટ તરીકે પણ ઓળખવામાં આવે છે.

બહુવિધ ચલોને એક મૂલ્ય સોંપો

ઉદાહરણ:”vardemo1.py”

x=y=z=50

print x

print y

print z

આઉટપુટ: \$python3 vardemo1.py

50

50

50

બહુવિધ ચલોને બહુવિધ મૂલ્યો સોંપો

ઉદાહરણ:”vardemo2.py”

a,b,c=5,10,15

print a

print b

print c

આઉટપુટ: \$python3 vardemo2.py

5

10

15

ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Perform Operations using Data Types and Operators)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- ડેટા પ્રકારો, પાયથોનમાં ઓપરેટર્સ.

સામાન્ય રીતે, ડેટા પ્રકારો સ્પષ્ટ કરે છે કે ચલોમાં કયા પ્રકારનો ડેટા સંગ્રહિત કરવામાં આવશે. ચલો વિવિધ ડેટા પ્રકારોના મૂલ્યોને પકડી શકે છે. પાયથોન એ ગતિશીલ રીતે ટાઈપ કરેલી અથવા ઢીલી રીતે ટાઈપ કરેલી ભાષા છે, તેથી આપણે તેને જાહેર કરતી વખતે ચલનો પ્રકાર વ્યાખ્યાયિત કરવાની જરૂર નથી. દુભાષિયા અસ્પષ્ટપણે મૂલ્યને તેના પ્રકાર સાથે જોડે છે.

એટલે કે ઢીલી રીતે ટાઈપ કરેલી ભાષામાં વેરીએબલના પ્રકારને સ્પષ્ટ કરવાની જરૂર નથી કારણ કે વેરીએબલ આપમેળે સોંપેલ મૂલ્યના આધારે તેના ડેટાટાઈપમાં ફેરફાર કરે છે.

પાયથોન પ્રકાર() ફંક્શન પ્રદાન કરે છે જે આપણને વેરીએબલનો પ્રકાર તપાસવામાં સક્ષમ બનાવે છે.

પાયથોન નીચેના પ્રમાણભૂત ડેટા પ્રકારો પ્રદાન કરે છે, તે છે

1 નંબરો

2 શબ્દમાળા

1 નંબરો

સંખ્યા આંકડાકીય મૂલ્યોનો સંગ્રહ કરે છે. જ્યારે વેરીએબલને નંબર અસાઈન કરવામાં આવે ત્યારે પાયથોન નંબર ટાઈપ વેરીએબલ બનાવે છે.

પાયથોનમાં ત્રણ આંકડાકીય પ્રકારો છે

a પૂર્ણાંક

b ફ્લોટ

c જટિલ

a પૂર્ણાંક

પૂર્ણાંક, અથવા પૂર્ણાંક, અમર્યાદિત લંબાઈની, દશાંશ વિના, સકારાત્મક અથવા નકારાત્મક, પૂર્ણ સંખ્યા છે.

ફોર્મેટ:

a =10

b =-12

c =123456789

b ફ્લોટ

ફ્લોટ અથવા “ફ્લોટિંગ પોઈન્ટ નંબર” એ એક સંખ્યા છે, હકારાત્મક અથવા નકારાત્મક, જેમાં એક અથવા વધુ દશાંશ હોય છે.

ફોર્મેટ:

X =1.0

Y =12.3

Z =-13.4

c જટિલ

જટિલ સંખ્યાઓ કાલ્પનિક ભાગ તરીકે “j” વડે લખવામાં આવે છે.

ફોર્મેટ:

A =2+5j

B =-3+4j

C =-6j

ઉદાહરણ: “datatypedemo1.py”

a =10

b =10.5

c =2.14j

print(“Datatype of Variable a :”,type(a))

print(“Datatype of Variable b :”,type(b))

print(“Datatype of Variable c :”,type(c))

આઉટપુટ: \$python3 datatypedemo1.py

Datatype of Variable a :<class ‘int’>

Datatype of Variable b :<class ‘float’>

Datatype of Variable c :<class ‘complex’>

2 શબ્દમાળા

શબ્દમાળાને અવતરણ ચિહ્નોમાં દર્શાવવામાં આવેલા અક્ષરોના ક્રમ તરીકે વ્યાખ્યાયિત કરી શકાય છે. પાયથોનમાં, સ્ટ્રિંગને વ્યાખ્યાયિત કરવા માટે આપણે સિંગલ, ડબલ અથવા ટ્રિપલ અવતરણનો ઉપયોગ કરી શકીએ છીએ.

સ્ટ્રિંગ હેન્ડલિંગના કિસ્સામાં, ઓપરેટર + નો ઉપયોગ બે સ્ટ્રિંગને જોડવા માટે થાય છે કારણ કે ઓપરેશન “hello”+”python” “hello python” પરત કરે છે.

ફોર્મેટ:

S1=’Welcome’

S2=”to Python”

S3=’”world”’

ઉદાહરણ:

a =”Welcome” #using double quotes

```
b='Python' #using single quotes
c="'''World''' #using triple quotes
print("Datatype of Variable a :",type(a))
print(a+b)#to concatenate two strings
```

આઉટપુટ:

```
Datatype of Variable a :<class 'str'>
Welcome Python
```

પાયથોનમાં ઓપરેટરો

ઓપરેટરને એક પ્રતીક તરીકે વ્યાખ્યાયિત કરી શકાય છે જે બે ઓપરેન્ડ વચ્ચેના ચોક્કસ ઓપરેશન માટે જવાબદાર છે.

પાયથોન નીચે પ્રમાણે વર્ણવેલ વિવિધ ઓપરેટરો પ્રદાન કરે છે.

- અંકગણિત ઓપરેટરો
- સોંપણી ઓપરેટરો
- સરખામણી ઓપરેટરો
- બીટવાઈઝ ઓપરેટર્સ
- ઓળખ ઓપરેટર્સ
- લોજિકલ ઓપરેટર્સ
- સભ્યપદ ઓપરેટરો

1 અંકગણિત ઓપરેટરો

અંકગણિત ઓપરેટર્સનો ઉપયોગ બે ઓપરેન્ડ વચ્ચે અંકગણિત કામગીરી કરવા માટે થાય છે.

ઓપરેટર	વર્ણન
+(addition)	બે ઓપરેન્ડ ઉમેરો
-(subtraction)	ડાબી બાજુથી જમણો ઓપરેન્ડ બાદ કરો
*(multiplication)	બે ઓપરેન્ડનો ગુણાકાર
/(divide)	કરો ડાબા ઓપરેન્ડને જમણા ઓપરેન્ડથી વિભાજિત કરો (હંમેશા ફ્લોટમાં પરિણમે છે)

%(remainder)	મોડ્યુલસ - જમણી બાજુએ ડાબા ઓપરેન્ડના વિભાજનનો બાકીનો ભાગ
// (floor division) results	ફ્લોર ડિવિઝન - નંબર લાઈનમાં ડાબી બાજુએ સમાયોજિત થયેલ સંપૂર્ણ સંખ્યામાં વિભાજન
** (exponent)	ઘાતાંક - ડાબું ઓપરેન્ડ જમણી ઘાત સુધી ઉભું કરેલું

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
x + y =30
x - y =10
x * y =200
x / y =2
x // y =0
x ** y =8
```

2 અસાઈનમેન્ટ ઓપરેટર્સ

અસાઈનમેન્ટ ઓપરેટર્સનો ઉપયોગ ડાબી ઓપરેન્ડને જમણી અભિવ્યક્તિની કિંમત સોંપવા માટે થાય છે.

ઓપરેટર	વર્ણન
= (Assigns to)	જમણી બાજુના ઓપરેન્ડથી ડાબી બાજુના ઓપરેન્ડને મૂલ્યો અસાઈન કરે છે.
+= (Assignment after Addition)	તે ડાબા ઓપરેન્ડમાં જમણું ઓપરેન્ડ ઉમેરે છે અને પરિણામ ડાબા ઓપરેન્ડને સોંપે છે.
*= (Assignment after Multiplication)	તે ડાબા ઓપરેન્ડ સાથે જમણા ઓપરેન્ડનો ગુણાકાર કરે છે અને પરિણામ ડાબા ઓપરેન્ડને સોંપે છે.
/= (Assignment after Division)	તે ડાબા ઓપરેન્ડને જમણા ઓપરેન્ડ સાથે વિભાજિત કરે છે અને પરિણામ ડાબા ઓપરેન્ડને સોંપે છે.
%= (Assignment after Modulus)	તે બે ઓપરેન્ડનો ઉપયોગ કરીને મોડ્યુલસ લે છે અને પરિણામને ડાબા ઓપરેન્ડને સોંપે છે.
**= (Assignment after Exponent)	ઓપરેટરો પર ઘાતાંકીય (પાવર) ગણતરી કરે છે અને ડાબા ઓપરેન્ડને મૂલ્ય અસાઈન કરે છે.
//= (Assignment after floor division)	તે ઓપરેટરો પર ફ્લોર ડિવિઝન કરે છે અને ડાબા ઓપરેન્ડને મૂલ્ય અસાઈન કરે છે.

ઉદાહરણ:

આઉટપુટ:

3 સરખામણી ઓપરેટરો

બે ઓપરેન્ડ અને તે મુજબ બુલિયન સાચા કે ખોટા પરત કરે છે.

ઓપરેટર	વર્ણન
== (Equal to)	સમાન - જો બંને ઓપરેન્ડ સમાન હોય તો સાચું.
!= (Not equal to)	ના બરાબર - જો ઓપરેન્ડ્સ સમાન ન હોય તો સાચું.
<= (Less than or equal)	તેનાથી ઓછું અથવા બરાબર - સાચું જો ડાબું ઓપરેન્ડ જમણા કરતા ઓછું અથવા બરાબર હોય
>= (Greater than or equal)	તેનાથી મોટો અથવા બરાબર - જો ડાબું ઓપરેન્ડ જમણી કરતા વધારે અથવા બરાબર હોય તો સાચું
< (Less than)	તે ઓછું - જો ડાબું ઓપરેન્ડ જમણા કરતા ઓછું હોય તો સાચું
> (Greater than)	ગ્રેટર થેન - જો ડાબો ઓપરેન્ડ જમણા કરતા મોટો હોય તો સાચું

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
x > y is True
x < y is False
x == y is False
x != y is True
x >= y is True
x <= y is False
```

4 બીટવાઈઝ ઓપરેટર્સ

ઓપરેટર	વર્ણન
& (binary and)	જો બંને બિટ્સ 1 હોય તો દરેક બીટને 1 પર સેટ કરે છે
(binary or)	જો બે બિટ્સમાંથી એક 1 હોય તો દરેક બીટને 1 પર સેટ કરે છે
^ (binary xor)	દરેક બીટને 1 પર સેટ કરે છે જો બેમાંથી માત્ર એક બીટ 1 હોય

~ (negation)	તમામ બિટ્સને ઉલટાવે છે
<< (left shift)	જમણી બાજુથી શૂન્યને અંદર દબાવીને ડાબે શિફ્ટ કરો અને ડાબી બાજુના બિટ્સ ને નીચે પડવા દો
>> (right shift)	ડાબી બાજુની સૌથી ડાબી બાજુની બિટ્સની નકલોને અંદર દબાવ કરીને જમણે શિફ્ટ કરો અને સૌથી જમણી બાજુના બિટ્સને પડવા દો

બીટવાઈઝ ઓપરેટરો બે ઓપરેન્ડના મૂલ્યો પર બીટ બાય બીટ ઓપરેશન કરે છે.

ઉદાહરણ:

```
a = 9 # equal to 1001
b = 12 # equal to 1100
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)
```

આઉટપુટ:

```
a & b = 8
a | b = 13
a ^ b = 5
~a = -10
a << 2 = 36
```

a >> 2 = 2

5 ઓળખ ઓપરેટર્સ

ઓળખ ઓપરેટર્સનો ઉપયોગ ઓબ્જેક્ટ્સની સરખામણી કરવા માટે થાય છે, જો તેઓ સમાન હોય તો નહીં, પરંતુ જો તેઓ વાસ્તવમાં એક જ ઓબ્જેક્ટ હોય, સમાન મેમરી સ્થાન સાથે.

ઓપરેટર	વર્ણન
is	જો બંને ચલ સમાન પદાર્થ હોય તો સાચું પરત કરે છે (જો id(x) બરાબર id(y) હોય તો)
is not	જો બંને ચલ સમાન પદાર્થ ન હોય તો સાચું પરત કરે છે

ઉદાહરણ:

```
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)))
```

આઉટપુટ:

```
True
id(x)=10105376id(y)=10105376
True
id(x)=10105376id(y)=10105696
```

6 લોજિકલ ઓપરેટર્સ

લોજિકલ ઓપરેટર્સનો ઉપયોગ મુખ્યત્વે અભિવ્યક્તિ મૂલ્યાંકનમાં નિર્ણય લેવા માટે થાય છે.

ઓપરેટર	વર્ણન
and (logical and)	જો બંને વિધાન સાચા હોય તો True પરત કરે છે
or (logical or)	જો નિવેદનોમાંથી એક સાચું હોય તો સાચું પરત કરે છે
not (logical not)	પરિણામને ઉલટાવી દો, જો પરિણામ સાચું હોય તો False પરત કરે છે

ઉદાહરણ :

```
x=True
y=False
print('x and y is',x and y)
print('x or y is',x or y)
print('not x is',not x)
```

આઉટપુટ:

```
x and y isFalse
x or y isTrue
not x isFalse
```

7 સભ્યપદ ઓપરેટર્સ

લોજિકલ ઓપરેટર્સનો ઉપયોગ મુખ્યત્વે અભિવ્યક્તિ મૂલ્યાંકનમાં નિર્ણય લેવા માટે થાય છે.

ઓપરેટર	વર્ણન
in	જો ઓબ્જેક્ટમાં ઉલ્લેખિત મૂલ્ય સાથેનો ક્રમ હાજર હોય તો તે સાચું પરત કરે છે (સૂચિ, ટ્યુપલ અથવા શબ્દકોશ)
not in	જો ઓબ્જેક્ટમાં ઉલ્લેખિત મૂલ્ય સાથેનો ક્રમ હાજર ન હોય તો તે સાચું પરત કરે છે (સૂચિ, ટ્યુપલ અથવા શબ્દકોશ)

ઉદાહરણ:

```
x='Hello world'
print('H'in x)
print('hello'notin x)
```

આઉટપુટ:

```
True
True
```


ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Control Flow with Decisions and Loops)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

• જો, જો-બીજું, જો-એલિફ-ઇઝ.

પાયથોનમાં શરતી નિવેદનો શરતોના આધારે વિવિધ ગણતરીઓ અથવા ક્રિયાઓ કરે છે.

પાયથોનમાં, નીચેના શરતી નિવેદનો છે.

- if
- if - else
- if - elif -else

ઇન્ડેન્ટેશન

પ્રોગ્રામિંગની સરળતા માટે અને સરળતા હાંસલ કરવા માટે, પાયથોન બ્લોક લેવલ કોડ માટે કૌંસનો ઉપયોગ કરવાની મંજૂરી આપતું નથી. પાયથોનમાં, ઇન્ડેન્ટેશનનો ઉપયોગ બ્લોક જાહેર કરવા માટે થાય છે. જો બે નિવેદનો સમાન ઇન્ડેન્ટેશન સ્તર પર હોય, તો તે એક જ બ્લોકનો ભાગ છે.

સામાન્ય રીતે, સ્ટેટમેન્ટને ઇન્ડેન્ટ કરવા માટે ચાર સ્પેસ આપવામાં આવે છે જે પાયથોનમાં ઇન્ડેન્ટેશનની લાક્ષણિક માત્રા હોય છે.

ઇન્ડેન્ટેશન એ પાયથોન ભાષાનો સૌથી વધુ ઉપયોગમાં લેવાતો ભાગ છે કારણ કે તે કોડના બ્લોકને જાહેર કરે છે. એક બ્લોકના તમામ નિવેદનો સમાન સ્તરના ઇન્ડેન્ટેશન માટે બનાવાયેલ છે.

જો નિવેદન

if સ્ટેટમેન્ટનો ઉપયોગ ચોક્કસ સ્થિતિને ચકાસવા માટે થાય છે. જો શરત સાચી હોય, તો કોડનો એક બ્લોક (જો બ્લોક) એક્ઝિક્યુટ કરવામાં આવશે.

વાક્યરચના:

if expression:

statement

ઉદાહરણ:

a = 33

b = 200

if b > a:

print("b is greater than a")

આઉટપુટ:

b is greater than a

જો - બીજું નિવેદન

if-else સ્ટેટમેન્ટ શરતના ખોટા કેસમાં એક્ઝિક્યુટ કરવામાં આવેલ if સ્ટેટમેન્ટ સાથે મળીને અન્ય બ્લોક પ્રદાન કરે છે.

જો શરત સાચી હોય, તો if બ્લોક ચલાવવામાં આવે છે. નહિતર, અન્ય બ્લોક ચલાવવામાં આવે છે.

વાક્યરચના:

if expression:

#block of statements

else:

#another block of statements

ઉદાહરણ:

age = int(input("Enter your age :"))

if age >= 18:

print("You are eligible to vote !!")

else:

print("Sorry! you have to wait !!")

આઉટપુટ:

Enter your age:19

You are eligible to vote!!

if - elif - else નિવેદન

એલિફ સ્ટેટમેન્ટ અમને બહુવિધ શરતો તપાસવા અને તેમની વચ્ચેની સાચી સ્થિતિના આધારે સ્ટેટમેન્ટના ચોક્કસ બ્લોકને ચલાવવા માટે સક્ષમ બનાવે છે.

અમારી જરૂરિયાતને આધારે અમે અમારા પ્રોગ્રામમાં કોઈપણ સંખ્યાબંધ elif સ્ટેટમેન્ટ ધરાવી શકીએ છીએ. જો કે, elif નો ઉપયોગ વૈકલ્પિક છે.

વાક્યરચના:

if expression 1:

block of statements

elif expression 2:

block of statements

elif expression 3:

block of statements

else:

block of statements

ઉદાહરણ:

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 ?")

આઉટપુટ:

Enter the marks:70

You scored grade B +..

ઉદાહરણ કાર્યક્રમ:

ધ્વ્યેય: ત્રણ નંબરો વચ્ચે મહત્તમ શોધવા માટે પાયથોન પ્રોગ્રામ

ઉદાહરણ:

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)

આઉટપુટ:

Enter a value:10

Enter b value:14

Enter c value:9

Maximum value is:14

પાયથોનમાં લૂપ સ્ટેટમેન્ટ (Loop Statements in Python)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

• જ્યારે લૂપ, લૂપ માટે.

કેટલીકવાર અમને પ્રોગ્રામના પ્રવાહમાં ફેરફાર કરવાની જરૂર પડી શકે છે. જો ચોક્કસ કોડના અમલને ઘણી વખત પુનરાવર્તિત કરવાની જરૂર પડી શકે છે તો આપણે લૂપ સ્ટેટમેન્ટ્સ માટે જઈ શકીએ છીએ.

આ હેતુ માટે, અજગર વિવિધ પ્રકારના લૂપ્સ પ્રદાન કરે છે જે અમુક ચોક્કસ કોડને ઘણી વખત પુનરાવર્તિત કરવામાં સક્ષમ છે. તે છે,

- જ્યારે લૂપ
- લૂપ માટે

જ્યારે લૂપ

જ્યાં સુધી શરત સાચી હોય ત્યાં સુધી જ્યારે લૂપ વડે આપણે સ્ટેટમેન્ટના સેટને એક્ઝિક્યુટ કરી શકીએ છીએ. જ્યારે લૂપ મોટે ભાગે એવા કિસ્સામાં વપરાય છે જ્યાં પુનરાવૃત્તિઓની સંખ્યા અગાઉથી જાણીતી નથી.

વાક્યરચના:

while expression:

Statement(s)

ઉદાહરણ:

i=1;

while i<=3:

print(i);

i=i+1;

આઉટપુટ:

1

2

3

જ્યારે લૂપ સાથે અન્યનો ઉપયોગ કરવો

Python અમને else બ્લોક સાથે જ્યારે લૂપનો ઉપયોગ કરવા સક્ષમ બનાવે છે. જ્યારે જ્યારે સ્ટેટમેન્ટમાં આપેલી શરત ખોટી બને છે ત્યારે અન્ય બ્લોક એક્ઝિક્યુટ થાય છે.

વાક્યરચના:

while expression:

Statements

else:

Statements

ઉદાહરણ:

i=1;

while i<=3:

print(i)

i=i+1;

else:print("The while loop terminated");

આઉટપુટ:

1
2
3

The while loop terminated

લૂપ માટે

પાયથોનમાં ફોર લૂપનો ઉપયોગ ઘણી વખત નિવેદનો અથવા પ્રોગ્રામના ભાગને પુનરાવર્તિત કરવા માટે થાય છે. તેનો વારંવાર ઉપયોગ યાદી, ટ્યૂપલ અથવા શબ્દકોશ જેવા ડેટા સ્ટ્રક્ચરને પાર કરવા માટે થાય છે.

વાક્યરચના:

```
for iterating_var in sequence:  
    statement(s)
```

ઉદાહરણ:

```
i=1  
n=int(input("Enter n value :"))  
for i in range(i,n+1):  
    print(i,end =')
```

આઉટપુટ:

Enter n value:5
1 2 3 4 5

લૂપ માટે અન્યનો ઉપયોગ કરવો

Python અમને લૂપ સાથે else સ્ટેટમેન્ટનો ઉપયોગ કરવાની પરવાનગી આપે છે જે માત્ર ત્યારે જ એક્ઝિક્યુટ થઈ શકે છે જ્યારે તમામ પુનરાવર્તનો ખતમ થઈ જાય. અહીં, આપણે નોંધવું જોઈએ કે જો લૂપમાં બ્રેક સ્ટેટમેન્ટ હોય તો અન્ય સ્ટેટમેન્ટ એક્ઝિક્યુટ કરવામાં આવશે નહીં.

વાક્યરચના:

```
for iterating_var in sequence:  
    statements  
else:  
    statements
```

ઉદાહરણ:

```
for i in range(0,5):  
    print(i)  
else:  
    print("for loop completely exhausted");
```

Output:

0
1

2

3

4

for loop completely exhausted

નોંધ

શ્રેણી() ફંક્શન ડિફોલ્ટ રૂપે 0 થી શરૂ કરીને, અને 1 (ડિફોલ્ટ રૂપે) દ્વારા વધારો કરીને સંખ્યાઓનો ક્રમ પરત કરે છે અને નિર્દિષ્ટ સંખ્યા પર સમાપ્ત થાય છે.

ઉદાહરણ:

range(6) means the values from 0 to 5.

range(2,6) means the values from 2 to 5.

ઉદાહરણ કાર્યક્રમ:

ધ્યેય : આપેલ નંબરના કોષ્ટકને છાપવાનો કાર્યક્રમ

ઉદાહરણ:

```
i=1;  
num =int(input("Enter a number:"));  
for i in range(1,11):  
    print("%d X %d = %d"%(num,i,num*i))
```

આઉટપુટ:

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

python માં લૂપ માટે નેસ્ટેડ

પાયથોન આપણને લૂપ માટે લૂપની અંદર ગમે તેટલા લૂપ્સને નેસ્ટ કરવા દે છે. બાહ્ય લૂપના દરેક પુનરાવર્તિત માટે આંતરિક લૂપ n વખત ચલાવવામાં આવે છે.

વાક્યરચના:

```
for iterating_var1 in sequence:  
    for iterating_var2 in sequence:  
        #block of statements
```

ઉદાહરણ કાર્યક્રમ:

ઉદ્દેશ્ય: આપેલ સંખ્યાના તારાઓની પંક્તિઓ છાપવાનો કાર્યક્રમ

ઉદાહરણ:

```
n=int(input("Enter the no.of rows you want to print :"))
for i in range(0,n):
    for j in range(0,i+1):
        print("*",end="")
```

print()

આઉટપુટ:

Enter the no.of rows you want to print:4

```
*
**
***
****
```

© NIMI
NOT TO BE REPUBLISHED

ડેટા પ્રકારો અને ઓપરેટર્સનો ઉપયોગ કરીને કામગીરી કરો (Document and Structure Code)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- list sequence in python
- function of list.

Python માં સૂચિ ક્રમ

પાયથોનમાં, સૂચિને મૂલ્યોના સંગ્રહ અથવા વિવિધ પ્રકારની વસ્તુઓ તરીકે વ્યાખ્યાયિત કરી શકાય છે. સૂચિમાંની વસ્તુઓ અલ્પવિરામ (,) વડે અલગ કરવામાં આવી છે અને ચોરસ કૌંસ [] સાથે બંધ કરવામાં આવી છે.

વાક્યરચના:

```
List = [value1, value2, value3, ...,]
```

ઉદાહરણ:

```
LO = [] #creates empty list
L1 = [123, "python", 3.7]
L2 = [1, 2, 3, 4, 5, 6]
L3 = ["C Programming", "Java", "Python"]
print(LO)
print(L1)
print(L2)
print(L3)
```

આઉટપુટ:

```
[]
[123, 'python', 3.7]
[1, 2, 3, 4, 5, 6].
['C Programming', 'Java', 'Python']
```

સૂચિ અનુક્રમણિકા

અનુક્રમણિકા એ જ રીતે પ્રક્રિયા કરવામાં આવે છે જે રીતે તે શબ્દમાળાઓ સાથે થાય છે. સ્લાઈસ ઓપરેટર [] નો ઉપયોગ કરીને સૂચિના ઘટકોને ઍક્સેસ કરી શકાય છે.

અનુક્રમણિકા 0 થી શરૂ થાય છે, સૂચિનું પ્રથમ તત્વ 0મી અનુક્રમણિકા પર સંગ્રહિત થાય છે, સૂચિનું બીજું તત્વ 1 લી અનુક્રમણિકા પર સંગ્રહિત થાય છે, વગેરે.

પાયથોન આપણને નેગેટિવ ઇન્ડેક્સિંગનો પણ ઉપયોગ કરવાની સુગમતા પૂરી પાડે છે. નકારાત્મક સૂચકાંકો જમણી બાજુથી ગણવામાં આવે છે. સૂચિના છેલ્લા તત્વ (સૌથી વધુ જમણો) અનુક્રમણિકા -1 ધરાવે છે, તેની નજીકનું ડાબું તત્વ અનુક્રમણિકા -2 પર હાજર છે અને તેથી જ જ્યાં સુધી ડાબી બાજુનું સૌથી તત્વ સામે ન આવે ત્યાં સુધી.

યાદી ઓપરેટરો

ઓપરેટર	વર્ણન
+	તે બે યાદીઓને જોડવા માટે વપરાતા જોડાણ ઓપરેટર તરીકે ઓળખાય છે
*	તે પુનરાવર્તન ઓપરેટર તરીકે ઓળખાય છે. તે સમાન સૂચિની બહુવિધ નકલોને જોડે છે.
[]	તે સ્લાઈસ ઓપરેટર તરીકે ઓળખાય છે. તેનો ઉપયોગ સૂચિમાંથી સૂચિ આઈટમને ઍક્સેસ કરવા માટે થાય છે.
[:]	તે રેન્જ સ્લાઈસ ઓપરેટર તરીકે ઓળખાય છે. તેનો ઉપયોગ સૂચિમાંથી સૂચિ વસ્તુઓની શ્રેણીને ઍક્સેસ કરવા માટે થાય છે.
in	તે સભ્યપદ ઓપરેટર તરીકે ઓળખાય છે. જો કોઈ ચોક્કસ આઈટમ ઉલ્લેખિત સૂચિમાં હાજર હોય તો તે પરત કરે છે.
not in	તે સભ્યપદ ઓપરેટર પણ છે અને જો સૂચિમાં કોઈ ચોક્કસ આઈટમ હાજર ન હોય તો તે સાચું પરત કરે છે.

ઉદાહરણ:

```
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 not in num)
```

આઉટપુટ:

```
[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

સૂચિમાં ઘટકોને કેવી રીતે ઉમેરવું અથવા બદલવું?

પાયથોન અમને સ્લાઈસ અને અસાઈનમેન્ટ ઓપરેટરનો ઉપયોગ કરીને સૂચિની વસ્તુઓમાં ફેરફાર કરવાની મંજૂરી આપે છે.

અમે આઈટમ અથવા વસ્તુઓની શ્રેણી બદલવા માટે અસાઈનમેન્ટ ઓપરેટર (=) નો ઉપયોગ કરી શકીએ છીએ.

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
[1,2,3,4,5]
[1,2,30,4,5]
[1,25,36,4,5]
[1,25,36,4,'Python']
```

સૂચિમાંથી ઘટકોને કેવી રીતે કાઢી નાખવું અથવા દૂર કરવું?

પાયથોન અમને ડેલ કીવર્ડનો ઉપયોગ કરીને સૂચિમાંથી એક અથવા વધુ વસ્તુઓને કાઢી નાખવાની મંજૂરી આપે છે.

ઉદાહરણ:

```
num=[1,2,3,4,5]
print(num)
delnum[1]
print(num)
delnum[1:3]
print(num)
```

આઉટપુટ:

```
[1,2,3,4,5]
[1,3,4,5]
[1,5]
```

સૂચિ કાર્યો

પાયથોન નીચેના બિલ્ટ-ઇન કાર્યો પૂરા પાડે છે જેનો ઉપયોગ સૂચિ સાથે કરી શકાય છે.

- લેન()
- મહત્તમ()
- મિનિટ()
- ચાદી()
- સરવાળો()
- સોર્ટ કરેલ()
- જોડો()
- દૂર કરો()
- સોર્ટ()
- વિપરીત()
- ગણતરી()
- અનુક્રમણિકા()
- insert()
- પોપ()
- ચોખ્ખું()

લેન()

Python len() માં સૂચિની લંબાઈ શોધવા માટે વપરાય છે, એટલે કે તે સૂચિમાંની વસ્તુઓની સંખ્યા પરત કરે છે.

વાક્યરચના:

```
len(list)
```

ઉદાહરણ:

```
num=[1,2,3,4,5,6]
print("length of list:",len(num))
```

આઉટપુટ:

```
length of list:6
?max()
```

In Python max() is used to find maximum value in the list

વાક્યરચના:

```
max(list)
```

ઉદાહરણ:

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp']
print("Max of list :",max(num))
print("Max of list :",max(lang))
```

આઉટપુટ:

```
Max of list:6
```

Max of list: python

min()

Python માં min() નો ઉપયોગ યાદીમાં ન્યૂનતમ મૂલ્ય શોધવા માટે થાય છે

વાક્યરચના:

```
min(list)
```

ઉદાહરણ:

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp']
print("Min of list :",min(num))
print("Min of list :",min(lang))
```

આઉટપુટ:

```
Min of list:1
Min of list: c
```

sum()

python માં, sum(list) ફંક્શન સૂચિમાંના તમામ મૂલ્યોનો સરવાળો આપે છે. સૂચિ મૂલ્યો સંખ્યા પ્રકારમાં હોવા જોઈએ.

વાક્યરચના:

```
sum(list)
```

ઉદાહરણ:

```
num=[1,2,3,4,5,6]
print("sum of list items :",sum(num))
```

આઉટપુટ:

```
sum of list items:21
```

sorted()

python માં, sorted(list) ફંક્શનનો ઉપયોગ યાદીની તમામ વસ્તુઓને ચડતા ક્રમમાં ગોઠવવા માટે થાય છે.

વાક્યરચના:

```
sorted(list)
```

ઉદાહરણ:

```
num=[1,3,2,4,6,5]
lang=['java','c','python','cpp']
print(sorted(num))
print(sorted(lang))
```

આઉટપુટ:

```
[1,2,3,4,5,6]
['c','cpp','java','python']
```

list ()

યાદી() પદ્ધતિ ક્રમ પ્રકારો લે છે અને તેમને યાદીઓમાં રૂપાંતરિત કરે છે. આનો ઉપયોગ આપેલ સ્ટ્રિંગ અથવા ટપલને સૂચિમાં કન્વર્ટ કરવા માટે થાય છે.

વાક્યરચના:

```
list(sequence)
```

ઉદાહરણ:

```
str="python"
list1=list(str)
print(list1)
```

આઉટપુટ:

```
['p','y','t','h','o','n']
```

append ()

python append() પદ્ધતિમાં યાદીના અંતમાં એક આઈટમ ઉમેરે છે.

વાક્યરચના:

```
list.append(item)
item may be number,string,list and etc
```

ઉદાહરણ:

```
num=[1,2,3,4,5]
lang=['python','c','java','php']
num.append(6)
print(num)
lang.append("cpp")
print(lang)
```

આઉટપુટ:

```
[1,2,3,4,5,6]
['python','c','java','php','cpp']
```

remove()

પાયથોન રિમૂવ() પદ્ધતિમાં સૂચિમાંથી પ્રથમ આઈટમ દૂર થાય છે જે પાસ કરેલ મૂલ્યની બરાબર છે. જો આઈટમ સૂચિમાં હાજર ન હોય તો તે ભૂલ ફેંકે છે.

વાક્યરચના:

```
list.remove(item)
item may be number,string,list and etc
```

ઉદાહરણ:

```
num=[1,2,3,4,5]
lang=['python','c','java','php','c']
num.remove(2)
```

```
print(num)
lang.remove("c")# first occurrence will remove
print(lang)
lang.remove("cpp")
print(lang)
```

આઉટપુટ:

```
[1,3,4,5]
['python','java','php','c']
ValueError:list.remove(x):x not in list
```

sort()

python sort() પદ્ધતિમાં યાદી તત્વોને સોર્ટ કરે છે. તે વસ્તુઓને ઉતરતા અને ચડતા ક્રમમાં પણ ગોઠવે છે. તે વૈકલ્પિક પરિમાણ 'રિવર્સ' લે છે જે સૂચિને ઉતરતા ક્રમમાં સોર્ટ કરે છે. મૂળભૂત રીતે, યાદી તત્વોને ચડતા ક્રમમાં સોર્ટ કરે છે.

વાક્યરચના:

```
list.sort([reverse=true])
reverse, which displays items in descending
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
['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()

પાયથોન રિવર્સ() પદ્ધતિમાં યાદીના ઘટકોને ઉલટાવે છે. જો સૂચિ ખાલી છે, તો તે ખાલી સૂચિ પરત કરે છે. યાદીની છેલ્લી ઇન્ડેક્સ વેલ્યુ ઉલટાવ્યા પછી 0 ઇન્ડેક્સ પર હાજર રહેશે.

વાક્યરચના:

```
list.reverse ()
```

ઉદાહરણ:

```
lang = ['p','y','t','h','o','n']
lang = ['p','y','t','h','o','n']
print("After reverse")
lang.reverse()
print(lang)
```

આઉટપુટ:

```
After reverse
['n','o','h','t','y','p']
```

count()

પાયથોન કાઉન્ટ() પદ્ધતિ યાદીમાં તત્વ દેખાય તેટલી વખત આપે છે. જો તત્વ સૂચિમાં હાજર ન હોય, તો તે 0 પરત કરે છે.

વાક્યરચના:

```
list.count(item)
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
Count of 2is:3
Count of 10is:0
```

index()

પાયથોન ઇન્ડેક્સ () પદ્ધતિમાં પાસ કરેલ તત્વનો ઇન્ડેક્સ પરત કરે છે. આ પદ્ધતિ દલીલ લે છે અને તેનો ઇન્ડેક્સ આપે છે. જો તત્વ હાજર ન હોય, તો તે મૂલ્યની ભૂલ ઊભી કરે છે.

જો સૂચિમાં ડુપ્લિકેટ તત્વો હોય, તો તે પ્રથમ આવેલા તત્વની અનુક્રમણિકા પરત કરે છે.

આ પદ્ધતિ બે વધુ વૈકલ્પિક પરિમાણો લે છે જે એક મર્યાદામાં અનુક્રમણિકા શોધવા માટે વપરાય છે.

વાક્યરચના:

```
list.index(x[, start[, end]])
```

ઉદાહરણ:

```
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'))
```

આઉટપુટ:

```
index of t is:2
index of p is:0
index of p is:6
ValueError:'z'isnotinlist
```

insert()

python insert() પદ્ધતિ સૂચિમાં ઉલ્લેખિત ઇન્ડેક્સ પર તત્વ દાખલ કરે છે. પ્રથમ દલીલ એ તત્વની અનુક્રમણિકા છે કે જેમાં તત્વ દાખલ કરવું છે.

વાક્યરચના:

```
list.insert(i,x)
i : index at which element would be inserted.
x : element to be inserted.
```

ઉદાહરણ:

```
num=[10,20,30,40,50]
num.insert(4,60)
print("updated list is :",num)
num.insert(7,70)
print("updated list is :",num)
```

આઉટપુટ:

```
updated listis:[10,20,30,40,60,50]
updated listis:[10,20,30,40,60,50,70]
```

pop()

python pop() એલિમેન્ટમાં સૂચિમાંથી ઉલ્લેખિત ઇન્ડેક્સ પર હાજર એક ઘટકને દૂર કરે છે. તે પોપ કરેલ તત્વ પરત કરે છે.

વાક્યરચના:

```
list.pop([i])
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
updated listis:[10,20,30,40]
updated listis:[10,20,40]
IndexError: pop index out of range
```

clear()

python clear() પદ્ધતિમાં યાદીમાંથી તમામ તત્વો દૂર થાય છે. તે સૂચિને સંપૂર્ણપણે સાફ કરે છે અને કંઈપણ પરત કરતું નથી.

વાક્યરચના:

```
list.clear()
```

ઉદાહરણ:

```
num=[10,20,30,40,50]
num.clear()
print("After clearing ",num)
```

આઉટપુટ:

```
After clearing [ ]
```

Python માં Tuple ક્રમ (Tuple Sequence in Python)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- ટપલ
- અનુક્રમણિકા:
- ટપલ ઓપરેટરો
- યાદીમાં ઘટકોને કેવી રીતે ઉમેરવું કે દૂર કરવું?
- Tuple કાર્યો.

પાયથોનમાં ટ્યુપલનો ઉપયોગ અપરિવર્તનશીલ અજગર પદાર્થોના ક્રમને સંગ્રહિત કરવા માટે થાય છે. ટ્યુપલ એ સૂચિઓ જેવું જ છે કારણ કે સૂચિમાં સંગ્રહિત વસ્તુઓની કિંમત બદલી શકાય છે જ્યારે ટ્યુપલ અપરિવર્તનશીલ છે અને ટ્યુપલમાં સંગ્રહિત વસ્તુઓની કિંમત બદલી શકાતી નથી.

ટ્યુપલને નાના કૌંસ સાથે બંધાયેલ અલ્પવિરામથી વિભાજિત મૂલ્યોના સંગ્રહ તરીકે લખી શકાય છે.

વાક્યરચના:

```
Tuple=(value1,value2...)
```

ઉદાહરણ:

```
T1=()
T2=(10,30,20,40,60)
T3=("C","Java","Python")
```

T4 =(501,"abc",19.5)

(501,'abc',19.5)

T5 =(90,)

(90,)

print(T1)

print(T2)

print(T3)

print(T4)

print(T5)

આઉટપુટ:

()

(10,30,20,40,60)

('C','java','Python')

ટપલ ઇન્ડેક્સિંગ:

ટ્યુપલમાં ઇન્ડેક્સિંગ અને સ્લાઈસિંગ યાદીઓ સમાન છે. ટ્યુપલમાં ઇન્ડેક્સિંગ 0 થી શરૂ થાય છે અને લંબાઈ (ટ્યુપલ) - 1 સુધી જાય છે.

સ્લાઈસ ઓપરેટરનો ઉપયોગ કરીને ટપલમાંની વસ્તુઓને એક્સેસ કરી શકાય છે. પાચથોન અમને ટ્યુપલમાં બહુવિધ વસ્તુઓને એક્સેસ કરવા માટે કોલોન ઓપરેટરનો ઉપયોગ કરવાની પણ પરવાનગી આપે છે.

અન્ય ભાષાઓથી વિપરીત, પાચથોન અમને નકારાત્મક અનુક્રમણિકાનો પણ ઉપયોગ કરવાની સુગમતા પ્રદાન કરે છે. નકારાત્મક સૂચકાંકો જમણી બાજુથી ગણવામાં આવે છે. ટ્યુપલના છેલ્લા તત્વ (સૌથી વધુ જમણો) અનુક્રમણિકા -1 ધરાવે છે, તેની નજીકનું ડાબું તત્વ અનુક્રમણિકા -2 પર હાજર છે અને તેથી જ્યાં સુધી ડાબે મોટાભાગના તત્વોનો સામનો ન થાય ત્યાં સુધી.

Tuple ઓપરેટરો

ઓપરેટર	વર્ણન
+	તે બે ટ્યુપલ્સને જોડવા માટે વપરાતા જોડાણ ઓપરેટર તરીકે ઓળખાય છે
*	તે પુનરાવર્તન ઓપરેટર તરીકે ઓળખાય છે. તે એક જ ટ્યુપલની બહુવિધ નકલોને જોડે છે.
[]	તે સ્લાઈસ ઓપરેટર તરીકે ઓળખાય છે. તેનો ઉપયોગ ટપલમાંથી આઈટમને એક્સેસ કરવા માટે થાય છે.
[:]	તે રેન્જ સ્લાઈસ ઓપરેટર તરીકે ઓળખાય છે. તેનો ઉપયોગ ટપલમાંથી વસ્તુઓની શ્રેણીને એક્સેસ કરવા માટે થાય છે.
in	તે સભ્યપદ ઓપરેટર તરીકે ઓળખાય છે. જો કોઈ ચોક્કસ વસ્તુ ઉલ્લેખિત ટ્યુપલમાં હાજર હોય તો તે પરત કરે છે.
not in	તે સભ્યપદ ઓપરેટર પણ છે અને જો કોઈ ચોક્કસ આઈટમ ટ્યુપલમાં હાજર ન હોય તો તે સાચું પરત કરે છે.

ઉદાહરણ:

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)

આઉટપુટ:

(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

સૂચિમાં ઘટકોને કેવી રીતે ઉમેરવું અથવા દૂર કરવું?

યાદીઓથી વિપરીત, ટ્યુપલ આઈટમ અપડેટ કરી શકાતી નથી અથવા કાઢી શકાતી નથી કારણ કે ટ્યુપલ્સ અપરિવર્તનશીલ છે. આખું ટપલ કાઢી નાખવા માટે, આપણે ટ્યુપલ નામ સાથે ડેલ કીવર્ડનો ઉપયોગ કરી શકીએ છીએ.

ઉદાહરણ:

mytuple=('python','c','java','php')

mytuple[3]="html"

#'tuple' object does not support item assignment

print(mytuple)

delmytuple[3]

#'tuple' object doesn't support item deletion

print(mytuple)

del mytuple

#deletes entire tuple

આઉટપુટ:

'tuple' object does not support item assignment

'tuple' object doesn't support item deletion

Tuple કાર્યો

પાયથોન નીચેના બિલ્ટ-ઇન કાર્યો પૂરા પાડે છે જેનો ઉપયોગ ટ્યુપલ્સ સાથે કરી શકાય છે.

- len()
- max()
- min()
- tuple()
- sum()
- sorted()
- index()
- count()

len()

Python len() માં ટ્યુપલની લંબાઈ શોધવા માટે વપરાય છે, એટલે કે તે ટ્યુપલમાં વસ્તુઓની સંખ્યા પરત કરે છે.

વાક્યરચના:

```
len(tuple)
```

ઉદાહરણ:

```
num=(1,2,3,4,5,6)
print("length of tuple:",len(num))
```

આઉટપુટ:

```
length of tuple:6
```

max()

Python max() માં ટ્યુપલમાં મહત્તમ મૂલ્ય શોધવા માટે વપરાય છે.

વાક્યરચના:

```
max(tuple)
```

ઉદાહરણ:

```
num=(1,2,3,4,5,6)
lang=('java','c','python','cpp')
print("Max of tuple :",max(num))
print("Max of tuple :",max(lang))
```

આઉટપુટ:

```
Max of tuple:6
Max of tuple: python
```

min()

Python માં min() નો ઉપયોગ ટ્યુપલમાં ન્યૂનતમ મૂલ્ય શોધવા માટે થાય છે.

વાક્યરચના:

```
min(tuple)
```

ઉદાહરણ:

```
num=(1,2,3,4,5,6)
lang=('java','c','python','cpp')
print("Min of tuple :",min(num))
print("Min of tuple :",min(lang))
```

આઉટપુટ:

```
Min of tuple:1
Min of tuple: c
```

sum()

python માં, sum(tuple) ફંક્શન ટ્યુપલમાં તમામ મૂલ્યોનો સરવાળો આપે છે. ટ્યુપલ મૂલ્યો સંખ્યાના પ્રકારમાં હોવા જોઈએ.

વાક્યરચના:

```
sum(tuple)
```

ઉદાહરણ:

```
num=(1,2,3,4,5,6)
print("sum of tuple items :",sum(num))
```

આઉટપુટ:

```
sum of tuple items:21
```

sorted()

પાયથોનમાં, સોર્ટેડ (ટ્યુપલ) ફંક્શનનો ઉપયોગ ટપલની બધી વસ્તુઓને ચડતા ક્રમમાં ગોઠવવા માટે થાય છે. તે વસ્તુઓને ઉતરતા અને ચડતા ક્રમમાં પણ ગોઠવે છે. તે વૈકલ્પિક પરિમાણ 'રિવર્સ' લે છે જે ટપલને ઉતરતા ક્રમમાં સોર્ટ કરે છે.

વાક્યરચના:

```
sorted(tuple[,reverse=True])
```

ઉદાહરણ:

```
num=(1,3,2,4,6,5)
lang=('java','c','python','cpp')
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

આઉટપુટ:

```
(1,2,3,4,5,6)
('c','cpp','java','python')
(6,5,4,3,2,1)
```

tuple (sequence)

ટ્યુપલ() પદ્ધતિ ક્રમ પ્રકારો લે છે અને તેમને ટ્યુપલ્સમાં રૂપાંતરિત કરે છે. આનો ઉપયોગ આપેલ સ્ટ્રિંગ અથવા સૂચિને ટપલમાં કન્વર્ટ કરવા માટે થાય છે.

વાક્યરચના:

```
tuple(sequence)
```

ઉદાહરણ:

```
str="python"
tuple1=tuple(str)
print(tuple1)
num=[1,2,3,4,5,6]
tuple2=tuple(num)
print(tuple2)
```

આઉટપુટ:

```
('p','y','t','h','o','n')
(1,2,3,4,5,6)
```

count()

પાયથોન કાઉન્ટ() પદ્ધતિમાં ટ્યુપલમાં તત્વ દેખાય તેટલી વખત આપે છે. જો તત્વ ટ્યુપલમાં હાજર ન હોય, તો તે 0 પરત કરે છે.

વાક્યરચના:

```
tuple.count(item)
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
Count of 2is:3
Count of 10is:0
```

Index()

પાયથોન ઇન્ડેક્સ () પદ્ધતિમાં પાસ કરેલ તત્વનો ઇન્ડેક્સ પરત કરે છે. આ પદ્ધતિ દલીલ લે છે અને તેનો ઇન્ડેક્સ આપે છે. જો તત્વ હાજર ન હોય, તો તે મૂલ્યની ભૂલ ઊભી કરે છે.

જો ટ્યુપલમાં ડુપ્લિકેટ તત્વો હોય, તો તે પ્રથમ આવેલા તત્વની અનુક્રમણિકા પરત કરે છે.

આ પદ્ધતિ બે વધુ વૈકલ્પિક પરિમાણો લે છે જે એક મર્યાદામાં અનુક્રમણિકા શોધવા માટે વપરાય છે.

વાક્યરચના:

```
tuple.index(x[, start[, end]])
```

ઉદાહરણ:

```
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'))
```

આઉટપુટ:

```
index of t is:2
index of p is:0
index of p is:6
ValueError:'z'isnotin tuple.
```

પાયથોનમાં સિક્વન્સ સેટ કરો (Set Sequence in Python)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- સમૂહ બનાવવો
- સેટ ઓપરેટરો
- સેટ કાર્યો.

પાયથોનમાં, સમૂહને સર્પાકાર કૌંસની અંદર બંધ કરાયેલી વિવિધ વસ્તુઓના અવ્યવસ્થિત સંગ્રહ તરીકે વ્યાખ્યાયિત કરી શકાય છે. સમૂહના ઘટકો ડુપ્લિકેટ હોઈ શકતા નથી. પાયથોન સમૂહના તત્વો અપરિવર્તનશીલ હોવા જોઈએ.

પાયથોનમાં અન્ય સંગ્રહોથી વિપરીત, સમૂહના ઘટકો સાથે કોઈ અનુક્રમણિકા જોડાયેલ નથી, એટલે કે, અમે અનુક્રમણિકા દ્વારા સેટના કોઈપણ ઘટકને સીધી રીતે ઍક્સેસ કરી શકતા નથી. જો કે, અમે તે બધાને એકસાથે છાપી શકીએ છીએ અથવા અમે સમૂહમાંથી લૂપ કરીને તત્વોની સૂચિ મેળવી શકીએ છીએ.

Creating a set

સર્પાકાર કૌંસ સાથે અલ્પવિરામથી અલગ કરેલી વસ્તુઓને બંધ કરીને સેટ બનાવી શકાય છે.

વાક્યરચના:

```
Set={value1,value2,...}
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
{'Wednesday','Tuesday','Sunday','Friday','Thursday',  
'Saturday','Monday'}
```

Looping through the set elements...

Wednesday

Tuesday

Sunday

Friday

Thursday

Saturday

Monday

Set Operators

પાયથોનમાં, અમે પાયથોન સેટ પર વિવિધ ગાણિતિક ક્રિયાઓ કરી શકીએ છીએ જેમ કે યુનિયન, આંતરછેદ, તફાવત વગેરે.

ઓપરેટર	વર્ણન
	યુનિયન ઓપરેટર
&	આંતરછેદ ઓપરેટર
-	તફાવત ઓપરેટર

Union (|) Operator

અથવા (|) ઓપરેટરનો ઉપયોગ કરીને બે સમૂહોના જોડાણની ગણતરી કરવામાં આવે છે. બે સેટના યુનિયનમાં તે બધી વસ્તુઓ શામેલ છે જે બંને સેટમાં હાજર છે.

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}  
Days2={"Thr","Fri","Sat","Sun","Mon"}  
print(Days1 | Days2)
```

આઉટપુટ:

```
{'Thr','Fri','Sun','Tue','Wed','Mon','Sat'}
```

Intersection (&) Operator

(છેદન) ઓપરેટરનો ઉપયોગ પાયથોનમાં બે સેટના આંતરછેદની ગણતરી કરવા માટે થાય છે. બે સમૂહોના આંતરછેદ એ તત્વોના સમૂહ તરીકે આપવામાં આવે છે જે બંને સમૂહોમાં સમાન હોય છે.

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}  
Days2={"Thr","Fri","Sat","Sun","Mon"}  
print(Days1 & Days2)
```

આઉટપુટ:

```
{'Mon','Sat'}
```

Difference (-) Operator

બાદબાકી (-) ઓપરેટરનો ઉપયોગ કરીને બે સેટના તફાવતની ગણતરી કરી શકાય છે. પરિણામી સમૂહ સમૂહ 1 માંથી તમામ ઘટકોને દૂર કરીને મેળવવામાં આવશે જે સમૂહ 2 માં હાજર છે.

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}  
Days2={"Thr","Fri","Sat","Sun","Mon"}  
print(Days1 - Days2)
```

આઉટપુટ:

```
{'Tue','Wed'}
```

Set Functions

પાયથોનમાં સેટ સાથે વાપરવા માટેની નીચેની પદ્ધતિઓનો સમાવેશ થાય છે.

- len(set)
- max(set)
- min(set)
- sum(set)
- sorted(set)
- set()
- add()
- update()
- discard()
- remove()
- pop()
- clear()
- union()
- intersection()
- difference()
- issubset()
- issuperset()

len()

Python len() માં સેટની લંબાઈ શોધવા માટે વપરાય છે, એટલે કે તે સમૂહમાંની વસ્તુઓની સંખ્યા પરત કરે છે.

વાક્યરચના:

```
len(set)
```

ઉદાહરણ:

```
num={1,2,3,4,5,6}  
print("length of set:",len(num))
```

આઉટપુટ:

length of set:6

max()

Python max() માં સેટમાં મહત્તમ મૂલ્ય શોધવા માટે વપરાય છે.

વાક્યરચના:

max(set)

ઉદાહરણ:

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Max of set :",max(num))
print("Max of set :",max(lang))
```

આઉટપુટ:

Max of set:6

Max of set: python

min()

Python માં min() નો ઉપયોગ સેટમાં ન્યૂનતમ મૂલ્ય શોધવા માટે થાય છે.

વાક્યરચના:

min(set)

ઉદાહરણ:

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Min of set :",min(num))
print("Min of set :",min(lang))
```

આઉટપુટ:

Min of set:1

Min of set: c

sum()

python માં, sum(set) ફંક્શન સેટમાંના તમામ મૂલ્યોનો સરવાળો આપે છે. સંખ્યાના પ્રકારમાં મૂલ્યો સેટ કરો.

વાક્યરચના:

sum(set)

ઉદાહરણ:

```
num={1,2,3,4,5,6}
print("sum of set items :",sum(num))
```

આઉટપુટ:

sum of set items:21

sorted()

પાયથોનમાં, સોર્ટેડ (સેટ) ફંક્શનનો ઉપયોગ સેટની બધી વસ્તુઓને ચડતા ક્રમમાં ગોઠવવા માટે થાય છે. તે વસ્તુઓને ઉતરતા અને ચડતા ક્રમમાં પણ ગોઠવે છે. તે વૈકલ્પિક પરિમાણ 'રિવર્સ' લે છે જે સેટને ઉતરતા ક્રમમાં સોર્ટ કરે છે.

વાક્યરચના:

sorted(set[,reverse=True])

ઉદાહરણ:

```
num={1,3,2,4,6,5}
lang={'java','c','python','cpp'}
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

આઉટપુટ:

```
{1,2,3,4,5,6}
{'c','cpp','java','python'}
{6,5,4,3,2,1}
```

set()

સેટ() પદ્ધતિ ક્રમ પ્રકારો લે છે અને તેમને સેટમાં રૂપાંતરિત કરે છે. આનો ઉપયોગ આપેલ સ્ટ્રિંગ અથવા લિસ્ટ અથવા ટપલને સેટમાં કન્વર્ટ કરવા માટે થાય છે.

વાક્યરચના:

set(sequence)

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
{'N','O','T','H','P','Y'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
```

add()

પાયથોનમાં, એડ() પદ્ધતિનો ઉપયોગ સેટમાં અમુક ચોક્કસ વસ્તુ ઉમેરવા માટે થાય છે.

વાક્યરચના:

set.add (item)

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
printing the original set...
{'Wednesday','Friday','Thursday','Tuesday','Monday'}
Printing the modified set...
{'Wednesday','Sunday','Friday','Thursday','Tuesday',
'Saturday','Monday'}
```

update()

પાયથોન સેટમાં એક કરતાં વધુ વસ્તુઓ ઉમેરવા માટે અપડેટ () પદ્ધતિ પ્રદાન કરે છે.

વાક્યરચના:

```
set.update([item1,item2...])
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
Printing the original set...
{'Mar','Apr','Jan','Feb'}
Printing the modified set...
{'Mar','Apr','Jan','Jun','May','Jul','Feb'}
```

update()

પાયથોન સેટમાં એક કરતાં વધુ વસ્તુઓ ઉમેરવા માટે અપડેટ () પદ્ધતિ પ્રદાન કરે છે.

વાક્યરચના:

```
set.update([item1,item2...])
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
Printing the original set...
{'Mar','Apr','Jan','Feb'}
Printing the modified set...
{'Mar','Apr','Jan','Jun','May','Jul','Feb'}
```

discard()

પાયથોન discard () પદ્ધતિ પ્રદાન કરે છે જેનો ઉપયોગ સમૂહમાંથી વસ્તુઓને દૂર કરવા માટે કરી શકાય છે. જો સેટમાં આઈટમ અસ્તિત્વમાં નથી, તો અજગર ભૂલ આપશે નહીં. પ્રોગ્રામ તેના નિયંત્રણ પ્રવાહને જાળવી રાખે છે.

વાક્યરચના:

```
set.discard(item)
```

ઉદાહરણ:

```
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") #doesn't give error
print("\n Printing the modified set...");
print(Months)
```

આઉટપુટ:

```
printing the original set...
{'Jan','Apr','Mar','Feb'}
Printing the modified set...
{'Jan','Mar','Feb'}
Printing the modified set...
{'Jan','Mar','Feb'}
```

remove()

પાયથોન દૂર () પદ્ધતિ પ્રદાન કરે છે જેનો ઉપયોગ સમૂહમાંથી વસ્તુઓને દૂર કરવા માટે કરી શકાય છે. જો સેટમાં આઈટમ અસ્તિત્વમાં નથી, તો અજગર ભૂલ આપશે.

વાક્યરચના:

```
set.remove(item)
```

ઉદાહરણ:

```
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)
```

આઉટપુટ:

```
printing the original set...
{'Feb','Jan','Apr','Mar'}
Printing the modified set...
{'Feb','Jan','Mar'}
KeyError:'May' doesn't exist.
```

pop()

Python માં, pop() પદ્ધતિનો ઉપયોગ વસ્તુને દૂર કરવા માટે થાય છે. જો કે, આ પદ્ધતિ હંમેશા છેલ્લી વસ્તુને દૂર કરશે.

વાક્યરચના:

```
set.pop ()
```

ઉદાહરણ:

```
Days={"Monday","Tuesday","Wednesday",
"Thursday","Friday"}
print("\n printing the original set ...")
print(Days)
Days.pop()
print("\n Printing the modified set...");
print(Days)
```

આઉટપુટ:

```
Printing the original set...
{'Monday','Wednesday','Friday','Tues
day','Thursday'}
Printing the modified set...
{'Wednesday','Friday','Tuesday','Thursday'}
```

clear()

પાયથોનમાં, ક્લિયર () પદ્ધતિનો ઉપયોગ સેટમાંની બધી વસ્તુઓને દૂર કરવા માટે થાય છે.

વાક્યરચના:

```
set.clear()
```

ઉદાહરણ:

```
Days = {"Monday","Tuesday","Wednesday",
"Thursday","Friday"}
print("\n printing the original set ...")
print(Days)
Days.clear()
print("\n Printing the modified set...");
print(Days)
```

આઉટપુટ:

```
printing the original set...
{'Monday','Wednesday','Friday','Tuesday','Thurs
day'}
Printing the modified set...
set()
```

union ()

પાયથોનમાં, યુનિયન () પદ્ધતિનો ઉપયોગ બે સમૂહોનું જોડાણ કરવા માટે થાય છે. બે સેટના યુનિયનમાં તે બધી વસ્તુઓ શામેલ છે જે બંને સેટમાં હાજર છે.

વાક્યરચના:

```
set1.union (set2)
```

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1.union(Days2))
```

આઉટપુટ:

```
{'Thr','Fri','Sun','Tue','Wed','Mon','Sat'}
```

intersection ()

પાયથોનમાં, આંતરછેદ () નો ઉપયોગ પાયથોનમાં બે સેટના આંતરછેદની ગણતરી કરવા માટે થાય છે. બે સમૂહોના આંતરછેદ એ તત્વોના સમૂહ તરીકે આપવામાં આવે છે જે બંને સમૂહોમાં સમાન હોય છે.

વાક્યરચના:

```
set1.intersection (set2)
```

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1.intersection(Days2))
```

આઉટપુટ:

```
{'Mon','Sat'}
```

difference ()

તફાવત () પદ્ધતિનો ઉપયોગ કરીને બે સેટના તફાવતની ગણતરી કરી શકાય છે. પરિણામી સમૂહ સમૂહ 1 માંથી તમામ ઘટકોને દૂર કરીને મેળવવામાં આવશે જે સમૂહ 2 માં હાજર છે.

વાક્યરચના:

```
set1.difference (set2)
```

ઉદાહરણ:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1.difference(Days2))
```

આઉટપુટ:

```
{'Tue','Wed'}
```

issubset()

જો સમૂહના તમામ ઘટકો બીજા સમૂહમાં હાજર હોય (દલીલ તરીકે પસાર કરવામાં આવે તો) issubset() પદ્ધતિ ટૂંકાપે છે. જો નહિં, તો તે False પરત કરે છે.

વાક્યરચના:

```
set1.issubset (set2)
```

ઉદાહરણ:

```
set1={1,2,3,4}
```

```
set2={1,2,3,4,5,6,7,8,9}
```

```
print(set1.issubset(set2))
```

```
print(set2.issubset(set1))
```

આઉટપુટ:

```
True
```

```
False
```

issuperset()

જો સેટમાં બીજા સેટના દરેક ઘટકો હોય (દલીલ તરીકે પસાર કરવામાં આવે તો) ઇશ્યુપરસેટ () પદ્ધતિ ટૂંકાપે છે. જો નહિં, તો તે False પરત કરે છે.

વાક્યરચના:

```
set1.issuperset (set2)
```

ઉદાહરણ:

```
set1={1,2,3,4}
set2={1,2,3,4,5,6,7,8,9}
print(set1.issuperset(set2))
print(set2.issuperset(set1))
```

આઉટપુટ:

```
False
```

```
True
```

પાયથોનમાં ડિક્શનરી સિક્વન્સ (Dictionary Sequence in Python)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- શબ્દકોષના મૂલ્યો સુધી પહોંચવું
- શબ્દકોષ મૂલ્યો અપડેટ કરી રહ્યા છીએ
- શબ્દકોષ મૂલ્યો કાઢી નાખવું
- લૂપ્સનો ઉપયોગ કરીને શબ્દકોષની કિંમતો ઍક્સેસ કરવી
- શબ્દકોષ કાર્યો.

પાયથોનમાં શબ્દકોષ એ કી વેલ્યુ જોડીઓનો સંગ્રહ છે જ્યાં મૂલ્ય કોઈપણ અજગર ઓબ્જેક્ટ હોઈ શકે છે જ્યારે કી એ અપરિવર્તનશીલ પાયથોન ઓબ્જેક્ટ છે, એટલે કે, નંબર્સ, સ્ટ્રિંગ અથવા ટ્યુપલ.

બહુવિધ કી-વેલ્યુ જોડીનો ઉપયોગ કરીને શબ્દકોષ બનાવી શકાય છે જે અલ્પવિરામ(,) દ્વારા અલગ કરવામાં આવે છે અને સર્પાકાર કૌંસ {} ની અંદર બંધ હોય છે.

વાક્યરચના:

```
Dict={key1:value1,key2:value2,.....}
```

ઉદાહરણ:

```
student ={"Name": "Kiran", "Age": 22, "Regno": 562,
"Branch": "CSE"}
```

```
print(student)
```

આઉટપુટ:

```
{'Name': 'Kiran', 'Age': 22, 'Regno': 562, 'Branch': 'CSE'}
```

શબ્દકોષ મૂલ્યો ઍક્સેસ કરી રહ્યા છીએ

સૂચીમાં ડેટા ઍક્સેસ કરી શકાય છે અને ઇન્ડેક્સીંગનો ઉપયોગ કરીને ટપલ કરી શકાય છે.

જો કે, કીનો ઉપયોગ કરીને શબ્દકોષમાં મૂલ્યો ઍક્સેસ કરી શકાય છે કારણ કે શબ્દકોષમાં કી અનન્ય છે.

ઉદાહરણ:

```
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”])

```

આઉટપુટ:

```

Name : Kiran
Age :22
RegNo :562
Branch : CSE

```

શબ્દકોશ મૂલ્યો અપડેટ કરી રહ્યાં છીએ

શબ્દકોષ એક પરિવર્તનશીલ ડેટા પ્રકાર છે, અને તેની કિંમતો ચોક્કસ કીનો ઉપયોગ કરીને અપડેટ કરી શકાય છે.

ઉદાહરણ:

```

student ={"Name":”Kiran”,”Age”:22,”Regno”: 562,
“Branch”:”CSE”}
print(“printing student data ....”)
print(student)
student[“Name”]=”Kishore”
print(“printing updated data ....”)
print(student)

```

આઉટપુટ:

```

printing student data ....
{‘Name’:’Kiran’,’Age’:22,’Regno’:562,’Branch’:’CSE’}
printing updated data ....
{‘Name’:’Kishore’,’Age’:22,’Regno’:562,’Branch’:’CSE’}

```

શબ્દકોશ મૂલ્યો કાઢી નાખી રહ્યાં છીએ

del કીવર્ડનો ઉપયોગ કરીને શબ્દકોશની વસ્તુઓને કાઢી શકાય છે.

ઉદાહરણ:

```

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)

```

આઉટપુટ:

```

printing student data ....
{‘Name’:’Kiran’,’Age’:22,’Regno’:562,’Branch’:’CSE’}
printing the modified information
{‘Name’:’Kiran’,’Age’:22,’Regno’:562}

```

લૂપ્સનો ઉપયોગ કરીને શબ્દકોશની કિંમતો ઍક્સેસ કરવી

ફોર લૂપનો ઉપયોગ કરીને શબ્દકોશને પુનરાવર્તિત કરી શકાય છે. અમે ફક્ત કી, માત્ર મૂલ્યો અને બંને કી અને મૂલ્યો ઍક્સેસ કરી શકીએ છીએ.

શબ્દકોશની બધી ચાવીઓ છાપો

ઉદાહરણ:

```

student ={"Name":”Kiran”,”Age”:22,”Regno”: 562,
“Branch”:”CSE”}
print(“Keys are :”)
for x in student:
print(x)

```

આઉટપુટ:

```

Keys are :
Name
Age
Regno
Branch
શબ્દકોશના તમામ મૂલ્યો છાપો

```

ઉદાહરણ:

```

print(“values are :”)
for x in student:
print(student[x])

```

આઉટપુટ:

```

values are :
Kiran
22
562
CSE

```

શબ્દકોશની બધી કી અને મૂલ્યો છાપો

ઉદાહરણ:

```

print(“Key and values are :”)
forx,yin student.items():
print(x,y)

```

આઉટપુટ:

Key and values are :

Name Kiran

Age 22

Regno 562

Branch CSE

શબ્દકોશ કાર્યો

પાયથોન નીચેના ઈન-બુલિટ ફંક્શનને સપોર્ટ કરે છે, તે છે

- len()
- copy()
- get()
- keys()
- items()
- values()
- update()
- pop()
- clear()

len()

python માં, len() ફંક્શનનો ઉપયોગ આપેલ શબ્દકોશની લંબાઈ શોધવા માટે થાય છે.

વાક્યરચના:

len(dictionary)

ઉદાહરણ:

```
student={"Name": "Kiran", "Age": 22, "Regno": 562,
"Branch": "CSE"}
print("Length of Dictionary is:", len(student))
```

આઉટપુટ:

Length of Dictionary is:4

copy()

તે આપેલ શબ્દકોશની બીજી નકલ પરત કરે છે.

વાક્યરચના:

dictionary.copy()

ઉદાહરણ:

```
student={"Name": "Kiran", "Age": 22, "Regno": 562,
"Branch": "CSE"}
student2=student.copy()
print(student2)
```

આઉટપુટ:

{'Name': 'Kiran', 'Age': 22, 'Regno': 562, 'Branch': 'CSE'}

get()

python માં, get() નો ઉપયોગ ઉલ્લેખિત કી ફોર્મ શબ્દકોશની કિંમત મેળવવા માટે થાય છે.

વાક્યરચના:

dictionary.get()

ઉદાહરણ:

```
student={"Name": "Kiran", "Age": 22, "Regno": 562,
"Branch": "CSE"}
print("Name is :", student.get("Name"))
print("RegNo is :", student.get("Regno"))
```

આઉટપુટ:

Name is: Kiran

RegNo is:562

keys()

python keys() માં ડિક્શનરીમાંથી બધી કી મેળવવા માટે પદ્ધતિનો ઉપયોગ થાય છે

વાક્યરચના:

dictionary.keys()

ઉદાહરણ:

```
student={"Name": "Kiran", "Age": 22, "Regno": 562,
"Branch": "CSE"}
for x in student.keys():
print(x)
```

આઉટપુટ:

Name

Age

Regno

Branch

items()

પાયથોન આઈટમ્સ() પદ્ધતિમાં શબ્દકોશનું નવું દૃશ્ય પરત કરે છે. આ દૃશ્ય કી મૂલ્યના ટ્યુપલનો સંગ્રહ છે.

વાક્યરચના:

dictionary.items()

ઉદાહરણ:

```
student={"Name": "Kiran", "Age": 22, "Regno": 562,
```

```
“Branch”:"CSE"]}
```

```
for x in student.items():
```

```
print(x)
```

આઉટપુટ:

```
('Name','Kiran')
```

```
('Age',22)
```

```
('Regno',562)
```

```
('Branch','CSE')
```

values()

પાયથોન મૂલ્યોમાં() પદ્ધતિનો ઉપયોગ શબ્દકોશમાંથી તમામ મૂલ્યો એકત્રિત કરવા માટે થાય છે.

વાક્યરચના:

```
Dictionary.values()
```

ઉદાહરણ:

```
student = {"Name": "Kiran", "Age": 22, "Regno": 562, "Branch": "CSE"}  
for x in student.values():  
print(x)
```

આઉટપુટ:

```
Kiran
```

```
22
```

```
562
```

```
CSE
```

update()

python update() પદ્ધતિમાં કી અને વેલ્યુ જોડીઓ સાથે શબ્દકોશને અપડેટ કરે છે. જો તે હાજર ન હોય તો તે કી/મૂલ્ય દાખલ કરે છે. જો તે પહેલાથી જ શબ્દકોશમાં હાજર હોય તો તે કી/મૂલ્ય અપડેટ કરે છે.

વાક્યરચના:

```
Dictionary.update({key:value,...})
```

ઉદાહરણ:

```
student = {"Name": "Kiran", "Age": 22, "Regno": 562, "Branch": "CSE"}  
student.update({"Regno": 590})  
student.update({"phno": 56895})  
print(student)
```

આઉટપુટ:

```
{'Name': 'Kiran', 'Age': 22, 'Regno': 590, 'Branch': 'CSE', 'phno': 56895}
```

pop()

python pop() પદ્ધતિમાં શબ્દકોશમાંથી એક તત્વ દૂર થાય છે. તે નિર્દિષ્ટ કી સાથે સંકળાયેલા તત્વને દૂર કરે છે.

જો ઉલ્લેખિત કી શબ્દકોશમાં હાજર હોય, તો તે તેની કિંમત દૂર કરે છે અને પરત કરે છે.

જો ઉલ્લેખિત કી હાજર ન હોય, તો તે એક ભૂલ KeyError ફેંકે છે.

વાક્યરચના:

```
Dictionary.remove(key)
```

ઉદાહરણ:

```
student = {"Name": "Kiran", "Age": 22, "Regno": 562, "Branch": "CSE"}  
student.pop('Age')  
print(student)  
student.pop('hallno')  
print(student)
```

આઉટપુટ:

```
{'Name': 'Kiran', 'Regno': 562, 'Branch': 'CSE'}  
KeyError: 'hallno'
```

clear()

python માં, clear() નો ઉપયોગ શબ્દકોશની બધી વસ્તુઓને કાઢી નાખવા માટે થાય છે.

વાક્યરચના:

```
Dictionary.clear()
```

ઉદાહરણ:

```
student = {"Name": "Kiran", "Age": 22, "Regno": 562, "Branch": "CSE"}  
print(student)  
student.clear()  
print(student)
```

આઉટપુટ:

```
{'Name': 'Kiran', 'Age': 22, 'Regno': 562, 'Branch': 'CSE'}  
{}
```


પાયથોન એરે (Python Arrays)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- એરેના તત્વો.

નોંધ: પાયથોન પાસે એરે માટે બિલ્ટ-ઇન સપોર્ટ નથી, પરંતુ તેના બદલે પાયથોન યાદીઓનો ઉપયોગ કરી શકાય છે.

એરે

નોંધ: આ પૃષ્ઠ તમને બતાવે છે કે કેવી રીતે લિસ્ટનો ઉપયોગ એરે તરીકે કરવો, જો કે, પાયથોનમાં એરે સાથે કામ કરવા માટે તમારે લાઇબ્રેરી આયાત કરવી પડશે, જેમ કે NumPy લાઇબ્રેરી.

એરેનો ઉપયોગ એક જ ચલમાં બહુવિધ મૂલ્યોને સંગ્રહિત કરવા માટે થાય છે:

ઉદાહરણ

Create an array containing car names:

```
cars = ["Ford", "Volvo", "BMW"]
```

Try it Yourself“

એરે શું છે?

એરે એ એક વિશિષ્ટ ચલ છે, જે એક સમયે એક કરતાં વધુ મૂલ્યોને પકડી શકે છે.

જો તમારી પાસે આઈટમ્સની સૂચિ છે (ઉદાહરણ તરીકે, કારના નામોની સૂચિ), તો કારને સિંગલ વેરિયેબલ્સમાં સ્ટોર કરવી આના જેવી દેખાઈ શકે છે:

```
car1 = "Ford"
```

```
car2 = "Volvo"
```

```
car3 = "BMW"
```

જો કે, જો તમે કારમાંથી લૂપ કરવા અને કોઈ ચોક્કસ શોધવા માંગતા હોવ તો શું? અને જો તમારી પાસે 3 કાર નહીં, પરંતુ 300 હોત તો?

ઉકેલ એ એરે છે!

એક એરે એક નામ હેઠળ ઘણા મૂલ્યો ધરાવે છે, અને તમે ઇન્ડેક્સ નંબરનો ઉલ્લેખ કરીને મૂલ્યોને ઍક્સેસ કરી શકો છો.

એરેના તત્વોને ઍક્સેસ કરો

તમે ઇન્ડેક્સ નંબરનો ઉલ્લેખ કરીને એરે એલિમેન્ટનો સંદર્ભ લો છો.

ઉદાહરણ

Get the value of the first array item:

```
x = cars[0]
```

Try it Yourself“

ઉદાહરણ

Modify the value of the first array item:

```
cars[0] = "Toyota"
```

તેને જાતે અજમાવી જુઓ

એરેની લંબાઈ

એરેની લંબાઈ (એરેમાં તત્વોની સંખ્યા) પરત કરવા માટે len() પદ્ધતિનો ઉપયોગ કરો.

ઉદાહરણ

Return the number of elements in the cars array:

```
x = len(cars)
```

Try it Yourself“

નોંધ: એરેની લંબાઈ હંમેશા સર્વોચ્ચ એરે ઇન્ડેક્સ કરતાં એક વધુ હોય છે.

લૂપિંગ એરે તત્વો

તમે એરેના તમામ ઘટકોને લૂપ કરવા માટે ઇન લૂપનો ઉપયોગ કરી શકો છો.

ઉદાહરણ

Print each item in the cars array:

```
for x in cars:
```

```
print(x)
```

Try it Yourself“

એરે તત્વો ઉમેરી રહ્યા છે

તમે એરેમાં તત્વ ઉમેરવા માટે append() પદ્ધતિનો ઉપયોગ કરી શકો છો.

ઉદાહરણ

Add one more element to the cars array:

```
cars.append("Honda")
```

Try it Yourself“

એરે તત્વો દૂર કરી રહ્યા છીએ

તમે એરેમાંથી તત્વ દૂર કરવા માટે pop() પદ્ધતિનો ઉપયોગ કરી શકો છો.

ઉદાહરણ

Delete the second element of the cars array:

```
cars.pop(1)
```

તેને જાતે અજમાવી જુઓ

તમે એરેમાંથી તત્વને દૂર કરવા માટે remove() પદ્ધતિનો પણ ઉપયોગ કરી શકો છો.

ઉદાહરણ

Delete the element that has the value "Volvo":

```
cars.remove("Volvo")
```

નોંધ: સૂચિની દૂર() પદ્ધતિ માત્ર ઉલ્લેખિત મૂલ્યની પ્રથમ ઘટનાને દૂર કરે છે.

પાયથોનમાં બિલ્ટ-ઇન પદ્ધતિઓનો સમૂહ છે જેનો ઉપયોગ તમે સૂચિઓ/ એરે પર કરી શકો છો.

પદ્ધતિ	વર્ણન
append()	સૂચિના અંતે એક ઘટક ઉમેરે છે
clear()	સૂચિમાંથી તમામ ઘટકોને દૂર કરે છે
copy()	સૂચિની નકલ પરત કરે છે
count()	ઉલ્લેખિત મૂલ્ય સાથે ઘટકોની સંખ્યા પરત કરે છે
extend()	વર્તમાન સૂચિના અંતમાં સૂચિના ઘટકો (અથવા કોઈપણ પુનરાવર્તિત) ઉમેરો
index()	ઉલ્લેખિત મૂલ્ય સાથે પ્રથમ ઘટકની અનુક્રમણિકા પરત કરે છે
insert()	ઉલ્લેખિત સ્થાન પર એક તત્વ ઉમેરે છે
pop()	નિર્દિષ્ટ સ્થાન પર તત્વ દૂર કરે છે
remove()	ઉલ્લેખિત મૂલ્ય સાથે પ્રથમ આઈટમ દૂર કરે છે
reverse()	સૂચિના ક્રમને ઉલટાવે છે
sort()	સૂચિને સોર્ટ કરે છે

નોંધ: પાયથોન પાસે એરે માટે બિલ્ટ-ઇન સપોર્ટ નથી, પરંતુ તેના બદલે પાયથોન યાદીઓનો ઉપયોગ કરી શકાય છે.

મોડ્યુલો અને ટૂલ્સનો ઉપયોગ કરીને કામગીરી કરો (Perform Operations Using Modules and Tools)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- મોડ્યુલ બનાવવું
- મોડ્યુલ લોડ કરી રહ્યું છે
- મોડ્યુલનું નામ બદલો.

પાયથોનમાં મોડ્યુલો

પાયથોન મોડ્યુલમાં પાયથોન પ્રોગ્રામ ફાઈલ તરીકે વ્યાખ્યાયિત કરી શકાય છે જેમાં પાયથોન ફંક્શન્સ, ક્લાસ અથવા વેરીએબલ્સ સહિત પાયથોન કોડ હોય છે. બીજા શબ્દોમાં કહીએ તો, આપણે કહી શકીએ કે એક્સ્ટેન્શન (.py) સાથે સાચવેલ અમારી પાયથોન કોડ ફાઈલને મોડ્યુલ તરીકે ગણવામાં આવે છે.

Python માં મોડ્યુલો અમને કોડને તાર્કિક રીતે ગોઠવવા માટે સુગમતા પ્રદાન કરે છે. એક મોડ્યુલની કાર્યક્ષમતાને બીજામાં વાપરવા માટે, અમારે ચોક્કસ મોડ્યુલ આયાત કરવું પડશે.

મોડ્યુલ બનાવી રહ્યા છીએ

ઉદાહરણ: 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
```

ઉપરના ઉદાહરણમાં આપણે ડેમો નામના મોડ્યુલની અંદર 4 ફંક્શન્સ sum(), sub(), mul() અને div() વ્યાખ્યાયિત કર્યા છે.

અમારા પાયથોન કોડમાં મોડ્યુલ લોડ કરી રહ્યા છીએ

તેની કાર્યક્ષમતાનો ઉપયોગ કરવા માટે અમારે અમારા પાયથોન કોડમાં મોડ્યુલ લોડ કરવાની જરૂર છે. પાયથોન નીચે વ્યાખ્યાયિત કર્યા મુજબ બે પ્રકારના નિવેદનો પૂરા પાડે છે.

- 1 આયાત નિવેદન
- 2 માંથી-આયાત નિવેદન
- 1 આયાત નિવેદન

આયાત નિવેદનનો ઉપયોગ એક મોડ્યુલની તમામ કાર્યક્ષમતાને બીજામાં આયાત કરવા માટે થાય છે. અહીં, આપણે નોંધવું જોઈએ કે આપણે કોઈપણ પાયથોન સ્રોત ફાઈલની કાર્યક્ષમતાનો ઉપયોગ તે ફાઈલને અન્ય પાયથોન સ્રોત ફાઈલમાં મોડ્યુલ તરીકે આયાત કરીને કરી શકીએ છીએ.

અમે એક આયાત નિવેદન સાથે બહુવિધ મોડ્યુલો આયાત કરી શકીએ છીએ.

વાક્યરચના:

```
import module1,module2..
```

ઉદાહરણ:

```
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))
```

આઉટપુટ:

```
Enter a :12

Enter b :6

Sum is:18

Sub is:6

Mul is:72

Div is:2.0
```

2 માંથી-આયાત નિવેદન

નેમસ્પેસમાં આખા મોડ્યુલને આયાત કરવાને બદલે, પાયથોન માત્ર મોડ્યુલના ચોક્કસ લક્ષણોને આયાત કરવા માટે સુગમતા પ્રદાન કરે છે. આમાંથી - આયાત નિવેદનનો ઉપયોગ કરીને કરી શકાય છે. આવા કિસ્સામાં આપણે ડોટ ઓપરેટરનો ઉપયોગ કરતા નથી.

વાક્યરચના:

```
from module-name import*
```

ઉદાહરણ:

```
from demo import*

a=int(input("Enter a :"))

b=int(input("Enter b :"))
```

```
print("Sum is :",sum(a,b))
print("Sub is :",sub(a,b))
print("Mul is :",mul(a,b))
print("Div is :",div(a,b))
```

આઉટપુટ:

```
Enter a :12
Enter b :6
Sum is:18
Sub is:6
Mul is:72
Div is:2.0
```

અમે સમગ્ર મોડ્યુલને આયાત કર્યા વિના મોડ્યુલમાંથી ચોક્કસ ફંક્શનને આયાત કરી શકીએ છીએ. અહીં એક ઉદાહરણ છે.

વાક્યરચના:

```
from module-name import function1,function2...
```

ઉદાહરણ:

```
from demo import sub,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))
```

આઉટપુટ:

```
Enter a :12
Enter b :6
Sub is:6
Mul is:72
```

મોડ્યુલનું નામ બદલીને

પાયથોન અમને ચોક્કસ નામ સાથે કેટલાક મોડ્યુલને આયાત કરવા માટે સુગમતા પ્રદાન કરે છે જેથી કરીને અમે અમારી પાયથોન સ્ત્રોત ફાઇલમાં તે મોડ્યુલનો ઉપયોગ કરવા માટે આ નામનો ઉપયોગ કરી શકીએ.

વાક્યરચના:

```
import module-name as specific-name
```

ઉદાહરણ:

```
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))
```

આઉટપુટ:

```
Enter a :25
Enter b :12
Sum is:37
Sub is:13
```

પાયથોન ઇટરેટર્સ (Python Iterators)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સક્ષમ થશો

- ઇટરેટર દ્વારા લૂપિંગ
- ઇટરેટર બનાવો
- પુનરાવર્તન બંધ કરો.

પાયથોન ઇટરેટર્સ

ઇટરેટર એ એક ઓબ્જેક્ટ છે જેમાં ગણતરી કરી શકાય તેવી સંખ્યા હોય છે.

ઇટરેટર એ એક ઓબ્જેક્ટ છે જેના પર પુનરાવર્તિત થઈ શકે છે, જેનો અર્થ છે કે તમે તમામ મૂલ્યોમાંથી પસાર થઈ શકો છો.

તકનીકી રીતે, પાયથોનમાં, ઇટરેટર એ એક ઓબ્જેક્ટ છે જે ઇટરેટર પ્રોટોકોલનો અમલ કરે છે, જેમાં `__iter__()` અને `__next__()` પદ્ધતિઓનો સમાવેશ થાય છે.

ઇટરેટર vs ઇટરેબલ

યાદીઓ, ટ્યુપલ્સ, શબ્દકોશો અને સમૂહો બધા પુનરાવર્તિત પદાર્થો છે. તે પુનરાવર્તિત કન્ટેનર છે જેમાંથી તમે પુનરાવર્તિત મેળવી શકો છો.

આ તમામ ઓબ્જેક્ટ્સમાં `iter()` પદ્ધતિ છે જેનો ઉપયોગ ઇટરેટર મેળવવા માટે થાય છે:

ઉદાહરણ:

Return an iterator from a tuple, and print each value:

```
mytuple = ("apple", "banana", "cherry")
myit = iter(mytuple)
print(next(myit))
print(next(myit))
print(next(myit))
```

જાતે અજમાવી જુઓ

શબ્દમાળાઓ પણ પુનરાવર્તિત પદાર્થો છે, અને પુનરાવર્તક પરત કરી શકે છે:

ઉદાહરણ:

શબ્દમાળાઓ પણ પુનરાવર્તિત પદાર્થો છે, જેમાં અક્ષરોનો ક્રમ છે:

```

mystr = "banana"
myit = iter(mystr)
print(next(myit))
print(next(myit))
print(next(myit))
print(next(myit))
print(next(myit))
print(next(myit))

```

જાતે અજમાવી જુઓ”

ઇટરરેટર દ્વારા લૂપિંગ

અમે પુનરાવર્તિત ઑબ્જેક્ટ દ્વારા પુનરાવર્તન કરવા માટે લૂપનો ઉપયોગ પણ કરી શકીએ છીએ:

ઉદાહરણ:

```

Iterate the values of a tuple:
mytuple = ("apple", "banana", "cherry")
for x in mytuple:
    print(x)
જાતે અજમાવી જુઓ”

```

ઉદાહરણ:

```

Iterate the characters of a string:
mystr = "banana"
for x in mystr:
    print(x)

```

જાતે અજમાવી જુઓ”

ફોર લૂપ વાસ્તવમાં ઇટરરેટર ઑબ્જેક્ટ બનાવે છે અને દરેક લૂપ માટે નેક્સ્ટ() મેથડનો અમલ કરે છે.

ઇટરરેટર બનાવો

ઇટરરેટર તરીકે ઑબ્જેક્ટ/ક્લાસ બનાવવા માટે તમારે તમારા ઑબ્જેક્ટમાં __iter__() અને __next__() પદ્ધતિઓનો અમલ કરવો પડશે.

જેમ તમે પાયથોન ક્લાસ/ઑબ્જેક્ટ્સ પ્રકરણમાં શીખ્યા છો તેમ, બધા વર્ગોમાં __init__() નામનું ફંક્શન હોય છે, જે તમને ઑબ્જેક્ટ બનાવતી વખતે થોડી શરૂઆત કરવાની પરવાનગી આપે છે.

__iter__() પદ્ધતિ સમાન રીતે કાર્ય કરે છે, તમે ઑપરેશન્સ કરી શકો છો (ઇન્શિયલાઇઝિંગ વગેરે), પરંતુ હંમેશા ઇટરરેટર ઑબ્જેક્ટ પોતે જ પરત કરવું આવશ્યક છે.

__next__() પદ્ધતિ તમને કામગીરી કરવા માટે પણ પરવાનગી આપે છે, અને અનુક્રમમાં આગલી આઈટમ પરત કરવી આવશ્યક છે.

ઉદાહરણ:

એક પુનરાવર્તક બનાવો જે નંબરો પરત કરે છે, 1 થી શરૂ થાય છે, અને દરેક ક્રમ એકથી વધશે (1,2,3,4,5 વગેરે પરત કરે છે):

```

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))

```

જાતે અજમાવી જુઓ”

સ્ટોપઇટરેશન

જો તમારી પાસે પર્યાપ્ત આગામી() સ્ટેટમેન્ટ હોય, અથવા જો તેનો ઉપયોગ લૂપ માટે કરવામાં આવ્યો હોય તો ઉપરનું ઉદાહરણ કાયમ ચાલુ રહેશે.

પુનરાવર્તનને હંમેશ માટે ચાલુ રાખવા માટે, અમે StopIteration સ્ટેટમેન્ટનો ઉપયોગ કરી શકીએ છીએ.

__next__() પદ્ધતિમાં, જો પુનરાવૃત્તિ નિર્દિષ્ટ સંખ્યામાં કરવામાં આવે તો ભૂલ વધારવા માટે અમે સમાપ્તિ સ્થિતિ ઉમેરી શકીએ છીએ:

ઉદાહરણ:

Stop after 20 iterations:

```

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)
```

પાયથોન મેથ્સ

પાયથોનમાં બિલ્ટ-ઇન ગણિત કાર્યોનો સમૂહ છે, જેમાં વ્યાપક ગણિત મોડ્યુલનો સમાવેશ થાય છે, જે તમને સંખ્યાઓ પર ગાણિતિક કાર્યો કરવા દે છે.

બિલ્ટ-ઇન ગણિત કાર્યો

min() અને max() ફંક્શનનો ઉપયોગ પુનરાવર્તિતમાં સૌથી નીચો અથવા ઉચ્ચતમ મૂલ્ય શોધવા માટે કરી શકાય છે:

ઉદાહરણ:

```
x = min(5, 10, 25)
```

```
y = max(5, 10, 25)
```

```
print(x)
```

```
print(y)
```

જાતે અજમાવી જુઓ”

pow(x, y) ફંક્શન x ની કિંમત y (xy) ની ઘાત પર આપે છે.

ઉદાહરણ:

Return the value of 4 to the power of 3 (same as 4 * 4 * 4):

```
x = pow(4, 3)
```

```
print(x)
```

Try it Yourself“

ગણિત મોડ્યુલ

પાયથોનમાં ગણિત નામનું બિલ્ટ-ઇન મોડ્યુલ પણ છે, જે ગાણિતિક કાર્યોની સૂચિને વિસ્તૃત કરે છે.

તેનો ઉપયોગ કરવા માટે, તમારે ગણિત મોડ્યુલ આયાત કરવું આવશ્યક છે:

આયાત ગણિત

જ્યારે તમે ગણિત મોડ્યુલ આયાત કરો છો, ત્યારે તમે મોડ્યુલની પદ્ધતિઓ અને સ્થિરાંકોનો ઉપયોગ કરવાનું શરૂ કરી શકો છો.

ઉદાહરણ તરીકે math.sqrt() પદ્ધતિ, સંખ્યાનું વર્ગમૂળ પરત કરે છે:

ઉદાહરણ:

```
import math
```

```
x = math.sqrt(64)
```

```
print(x)
```

જાતે અજમાવી જુઓ”

math.ceil() પદ્ધતિ સંખ્યાને તેના નજીકના પૂર્ણાંકની ઉપરની તરફ રાઉન્ડ કરે છે, અને math.floor() પદ્ધતિ કોઈ સંખ્યાને તેના નજીકના પૂર્ણાંક સુધી નીચેની તરફ રાઉન્ડ કરે છે અને પરિણામ આપે છે:

ઉદાહરણ:

```
import math
```

```
x = math.ceil(1.4)
```

```
y = math.floor(1.4)
```

```
print(x) # returns 2
```

```
print(y) # returns 1
```

જાતે અજમાવી જુઓ”

math.pi સ્થિરાંક, PI (3.14...) નું મૂલ્ય પરત કરે છે:

ઉદાહરણ:

```
import math
```

```
x = math.pi
```

```
print(x)
```

પૂર્ણ ગણિત મોડ્યુલ સંદર્ભ

અમારા ગણિત મોડ્યુલ સંદર્ભમાં તમને ગણિત મોડ્યુલની તમામ પદ્ધતિઓ અને સ્થિરાંકોનો સંપૂર્ણ સંદર્ભ મળશે.

Python વપરાશકર્તા ઇનપુટ

વપરાશકર્તા ઇનપુટ

Python વપરાશકર્તા ઇનપુટ માટે પરવાનગી આપે છે.

તેનો અર્થ એ કે અમે વપરાશકર્તાને ઇનપુટ માટે પૂછવા સક્ષમ છીએ.

પાયથોન 2.7 કરતાં પાયથોન 3.6 માં પદ્ધતિ થોડી અલગ છે.

Python 3.6 input() પદ્ધતિનો ઉપયોગ કરે છે.

પાયથોન 2.7 raw_input() પદ્ધતિનો ઉપયોગ કરે છે.

નીચેનું ઉદાહરણ વપરાશકર્તાના માટે પૂછે છે, અને જ્યારે તમે વપરાશકર્તા નામ દાખલ કરો છો, ત્યારે તે સ્ક્રીન પર છાપવામાં આવે છે:

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 “

જ્યારે ઇનપુટ() ફંક્શનની વાત આવે ત્યારે પાયથોન એક્ઝેક્યુટ કરવાનું બંધ કરે છે, અને જ્યારે વપરાશકર્તાએ અમુક ઇનપુટ આપ્યા હોય ત્યારે ચાલુ રહે છે.

ઑબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ અને જાવા લેંગ્વેજ (Object Oriented Programming and JAVA Language)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- JAVA શું છે
- OOPs ની વિભાવનાઓ.

જાવા ટ્યુટોરીયલ

જાવા શું છે

અમારું મુખ્ય જાવા પ્રોગ્રામિંગ ટ્યુટોરીયલ વિદ્યાર્થીઓ અને કાર્યકારી વ્યાવસાયિકો માટે રચાયેલ છે. જાવા એ ઑબ્જેક્ટ ઓરિએન્ટેડ, વર્ગ આધારિત, સમવર્તી, સુરક્ષિત અને સામાન્ય હેતુવાળી કમ્પ્યુટર પ્રોગ્રામિંગ ભાષા છે. તે વ્યાપકપણે ઉપયોગમાં લેવાતી મજબૂત તકનીક છે.

જાવા એક પ્રોગ્રામિંગ ભાષા અને પ્લેટફોર્મ છે. જાવા એ ઉચ્ચ સ્તરીય, મજબૂત, ઑબ્જેક્ટ-ઓરિએન્ટેડ અને સુરક્ષિત પ્રોગ્રામી ભાષા છે.

જાવાને વર્ષ 1995માં સન માઈક્રોસિસ્ટમ્સ (જે હવે ઓરેકલની પેટાકંપની છે) દ્વારા વિકસાવવામાં આવી હતી. જેમ્સગોસલિંગને જાવાના પિતા તરીકે ઓળખવામાં આવે છે. જાવા પહેલા તેનું નામ ઓક હતું. ઓક પહેલેથી જ રજિસ્ટર્ડ કોમ્પા y હોવાથી, જેમ્સ ગોસલિંગાંધીની ટીમે ઓકથી જાવા નામ બદલી નાખ્યું.

પ્લેટફોર્મ: કોઈપણ હાર્ડવેર અથવા સોફ્ટવેર પર્યાવરણ જેમાં પ્રોગ્રામ ચાલે છે, તેને પ્લેટફોર્મ તરીકે ઓળખવામાં આવે છે. Java has runtime environment (JRE) અને API હોવાથી, તેને પ્લેટ ફોર્મ કહે છે.

Java ઉદાહરણ

ચાલો જાવા પ્રોગ્રામિંગ ઉદાહરણ પર એક ઝડપી નજર કરીએ. હેલો જાવા ઉદાહરણનું વિગતવાર વર્ણન આગલા પૃષ્ઠ પર ઉપલબ્ધ છે.

Simple.java

```
class Simple{
    public static void main
    (String args[])
    {System.out.println("HelloJava");
    }
}
```

અરજી

સન મુજબ, 3 અબજ ઉપકરણો જાવા ચલાવે છે. એવા ઘણા ઉપકરણો છે જ્યાં જાવા હાલમાં ઉપયોગમાં લેવાય છે. તેમાંના કેટલાક વિસ્તારો નીચે મુજબ છે:

- 1 ડેસ્કટોપ એપ્લિકેશન્સ જેમ કે એક્રોબેટ રીડર, મીડિયાપ્લેયર, એન્ટીવાયરસ, વગેરે.
- 2 વેબ એપ્લિકેશન્સ જેમ કે irctc.co.in, javatpoint.com, વગેરે
- 3 એન્ટરપ્રાઇઝ એપ્લિકેશન્સ જેમ કે બેંકિંગ એપ્લિકેશન.

4 મોબાઇલ

5 એમ્બેડેડ સિસ્ટમ

6 સ્માર્ટકાર્ડ

7 રોબોટિક્સ

8 રમતો, વગેરે.

જાવા એપ્લિકેશનના પ્રકારો

જાવા પ્રોગ્રામિંગનો ઉપયોગ કરીને 4 પ્રકારની એપ્લિકેશનો બનાવી શકાય છે:

1 એકલ એપ્લિકેશન

એકલ એપ્લિકેશનને ડેસ્કટોપ એપ્લિકેશન અથવા વિન્ડો-આધારિત એપ્લિકેશન તરીકે પણ ઓળખવામાં આવે છે. આ પરંપરાગત સોફ્ટવેર છે જેને આપણે દરેક મશીન પર ઇન્સ્ટોલ કરવાની જરૂર છે. સ્ટેન્ડઅલોન એપ્લિકેશનના ઉદાહરણો છે મીડિયાપ્લેયર, એન્ટિવ રસ, વગેરે. AWT અને Swingware નો ઉપયોગ એકલા એપ્લિકેશન બનાવવા માટે જાવા માં થાય છે.

2 વેબ એપ્લિકેશન

એક એપ્લિકેશન જે સર્વર બાજુ પર ચાલે છે અને ગતિશીલ પૃષ્ઠ બનાવે છે તેને વેબ એપ્લિકેશન કહેવામાં આવે છે. હાલમાં, સર્વલેટ, જેએસપી, સ્ટ્રટ્સ, સ્પ્રિંગ, હિબ નેટ, જેએસએફ, વગેરે. જાવામાં વેબ એપ્લિકેશન બનાવવા માટે ટેકનોલોજીનો ઉપયોગ થાય છે.

3 એન્ટરપ્રાઇઝ એપ્લિકેશન

એક એપ્લિકેશન કે જે પ્રકૃતિમાં વિતરિત કરવામાં આવે છે, જેમ કે બેંકિંગ એપ્લિકેશન્સ, વગેરેને એન્ટરપ્રાઇઝ એપ્લિકેશન કહેવામાં આવે છે. તેમાં ઉચ્ચ-સ્તરીય સુરક્ષા, લોડ બેલેન્સિંગ અને કલસ્ટરિંગ જેવા ફાયદા છે. જાવામાં, EJB નો ઉપયોગ એન્ટરપ્રાઇઝ એપ્લિકેશન બનાવવા માટે થાય છે.

4 મોબાઇલ એપ્લિકેશન

મોબાઇલ ઉપકરણો માટે બનાવેલ એપ્લિકેશનને મોબાઇલ એપ્લિકેશન કહેવામાં આવે છે. હાલમાં, Android અને Java ME નો ઉપયોગ મોબાઇલ બનાવવા માટે થાય છે એપ્લિકેશન્સ

જાવા પ્લેટફોર્મ/એડીશન

જાવાના 4 પ્લેટફોર્મ સોર્ટિશન છે:

1 JavaSE (જાવા સ્ટાન્ડર્ડ એડીશન)

તે જાવા પ્રોગ્રામિંગ પ્લેટફોર્મ છે. તેમાં Java પ્રોગ્રામિંગ APIનો સમાવેશ થાય છે જેમ કે java.lang, java.io, java.net, java.util, java.sql, java.math વગેરે. તેમાં OOPs, સ્ટ્રિંગ, રેજેક્સ, અપવાદ, આંતરિક વર્ગો, મલ્ટી થ્રેડીંગ જેવા મુખ્ય વિષયોનો સમાવેશ થાય છે. J/OSTre m, નેટવર્કિંગ, AWT, સ્વિંગ, રિફ્લેક્શન, કલેક્શન, વગેરે.

2 JavaEE (જાવા એન્ટરપ્રાઇઝ એડિશન)

તે એક એન્ટરપ્રાઇઝ પ્લેટફોર્મ છે જેનો ઉપયોગ મુખ્યત્વે વેબ અને એન્ટરપ્રાઇઝ એપ્લિકેશન વિકસાવવા માટે થાય છે. તે જાવા SE platform. It includes એસ્ટોપિક્સ જેમ કે Servlet, JSP, વેબ સેવાઓ, EJB, JPA, વગેરેની ટોચ પર બનેલ છે.

3 JavaME (જાવા માઇક્રો એડિશન)

આઈટીઆઈ એક માઇક્રો પ્લેટફોર્મ છે જે મોબાઇલ એપ્લિકેશનને સમર્પિત છે.

4 JavaFX

તેનો ઉપયોગ નેટ એપ્લિકેશન વિકસાવવા માટે થાય છે. તે હળવા વજનના વપરાશકર્તા ઇન્ટરફેસ API નો ઉપયોગ કરે છે.

પૂર્વશરત

જાવા શીખવા માટે, તમારે C C++ પ્રોગ્રામિંગ ભાષાની મૂળભૂત જાણકારી હોવી જોઈએ.

પ્રેક્ષકો

અમારું જાવા પ્રોગ્રામિંગ ટ્યુટોરીયલ નવા નિશાળીયા અને વ્યાવસાયિકોને મદદ કરવા માટે રચાયેલ છે.

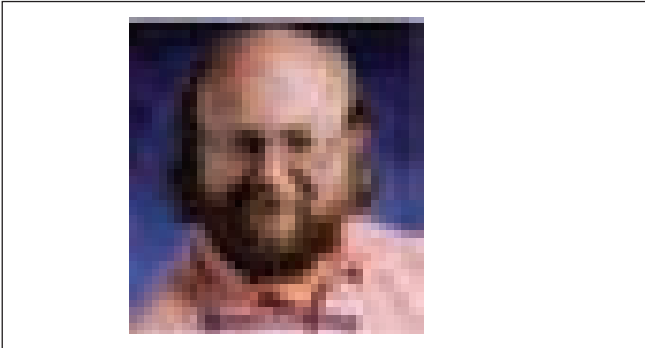
જાવાનો ઇતિહાસ

જાવાનો ઇતિહાસ ખૂબ જ રસપ્રદ છે. જાવા મૂળ રીતે ઇન્ટરેક્ટિવ ટેલિવિઝન માટે ડિઝાઇન કરવામાં આવી હતી, પરંતુ તે સમયે તે ડિજિટલ કેબલ ટેલિવિઝન ઉદ્યોગ માટે ખૂબ જ અદ્યતન તકનીક હતી. જાવાનો ઇતિહાસ ગ્રીન ટીમથી શરૂ થાય છે. જાવા ટીમના સભ્યો (જેને ગ્રીન ટીમ તરીકે પણ ઓળખવામાં આવે છે), એ સેટ-ટોપ બોક્સ, ટેલિવિઝન વગેરે જેવા ડિજિટલ ઉપકરણો માટે ભાષા વિકસાવવા માટે આ પ્રોજેક્ટ શરૂ કર્યો હતો. જો કે, તે ઇન્ટરનેટ પ્રોગ્રામિંગ માટે સૌથી યોગ્ય હતું. નેટસ્કેપ.

જાવા પ્રોગ્રામિંગ બનાવવા માટેના સિદ્ધાંતો “સરળ, મજબૂત, પોર્ટેબલ, પ્લેટફોર્મ સ્વતંત્ર, સુરક્ષિત, ઉચ્ચ પ્રદર્શન, મલ્ટિથ્રેડેડ, આર્કિટેક્ચર ન્યુટ્રલ, ઓબ્જેક્ટ-ઓરિએન્ટેડ, ઇન્ટરપ્રિટેડ અને ડાયનેમિક” હતા. જાવા 1995 માં જેમ્સ ગોસલિંગ દ્વારા વિકસાવવામાં આવી હતી, જે જાવાના પિતા તરીકે ઓળખાય છે. જેમ્સ ગોસલિંગ અને તેમની ટીમના સભ્યોએ 90 ના દાયકાની શરૂઆતમાં આ પ્રોજેક્ટ શરૂ કર્યો હતો.

હાલમાં, જાવાનો ઉપયોગ ઇન્ટરનેટ પ્રોગ્રામિંગ, મોબાઇલ ડિવાઇસ, ગેમ્સ, ઇ-બિઝનેસ સોલ્યુશન્સ વગેરેમાં થાય છે. નીચેના મહત્વના મુદ્દા આપવામાં આવ્યા છે જે જાવાના ઇતિહાસનું વર્ણન કરે છે.

જાવાના સ્થાપક જેમ્સ ગોસલિંગ



1 જેમ્સ ગોસલિંગ, માઈક શેરિડન અને પેટ્રિક નૌટનએ જૂન 1991માં જાવા ભાષા પ્રોજેક્ટની શરૂઆત કરી. સનએન્જિનિયર્સની સ્મોલટીમ ગ્રીનટીમ કહેવાય છે.

- 2 શરૂઆતમાં તે સેટ-ટોપ બોક્સ જેવા ઇલેક્ટ્રોનિક ઉપકરણોમાં નાની, એમ્બેડેડ સિસ્ટમ્સ માટે ડિઝાઇન કરવામાં આવી હતી.
- 3 સૌપ્રથમ, તેને જેમ્સ ગોસલિંગ દ્વારા “Greentalk” કહેવામાં આવતું હતું, અને ફાઇલ એક્સ્ટેન્શન હતું .gt.
- 4 તે પછી, તેને ઓક કહેવામાં આવતું હતું અને તેને ગ્રીન પ્રોજેક્ટના ભાગ રૂપે વિકસાવવામાં આવ્યું હતું.

શા માટે જાવાને “ઓક” નામ આપવામાં આવ્યું? ઓકથી જાવા સુધીનો જાવા ઇતિહાસ

- 5 શા માટે ઓક? ઓક શક્તિનું પ્રતીક છે અને યુએસએ, ફ્રાન્સ, જર્મની, રોમાનિયા વગેરે જેવા ઘણા દેશોના રાષ્ટ્રીય વૃક્ષ તરીકે પસંદ કરવામાં આવે છે.
- 6 માં 1995, ઓકનું નામ બદલીને “જાવા” રાખવામાં આવ્યું કારણ કે તે પહેલેથી જ ઓક ટેકનોલોજીસ દ્વારા ટ્રેડ માર્ક હતું.

જાવા પ્રોગ્રામિંગને “જાવા” શા માટે નામ આપવામાં આવ્યું?

- 7 શા માટે તેઓએ જાવા ભાષા માટે જાવા નામ પસંદ કર્યું? ટીમ નવું નામ પસંદ કરવા એકત્ર થઈ. સૂચિત શબ્દો “ગતિશીલ”, “ક્રાંતિકારી”, “સિલ્ડ”, “આંચકો”, “ડીએનએ”, વગેરે હતા જે ટેકનોલોજીના સારને પ્રતિબિંબિત કરે છે: ક્રાંતિકારી, ગતિશીલ, જીવંત, ઠંડી, અનન્ય અને સરળ જોડણી, અને funtosay.

જેમ્સ ગોસલિંગ અનુસાર, “જાવા સિલ્ડની સાથે ટોચની પસંદગીઓમાંની એક હતી”. જાવા ખૂબ જ અનોખું હોવાથી, મોટાભાગના ટીમના સભ્યોએ અન્ય નામો કરતાં જાવાને પ્રાધાન્ય આપ્યું હતું.

- 8 જાવા ઇન્ડોનેશિયામાં એક ટાપુ છે જ્યાં પ્રથમ કોફીનું ઉત્પાદન થયું હતું (જેને જાવા કોફી કહેવાય છે). તે એક પ્રકારનું એસ્પ્રેસોબીન છે. જાવા નામ ચોસેનબ જેમ્સ ગોસલિંગ જ્યારે તેની ઓફિસ પાસે કોફીનો કપ પીતા હતા.

- 9 નોંધ લો કે જાવા માત્ર એક નામ છે, ટૂંકાક્ષર નથી.

- 10 શરૂઆતમાં જેમ્સ ગોસલિંગ દ્વારા સન માઈક્રો સિસ્ટમ્સ (જે હવે ઓરેકલ કોર્પોરેશનની પેટાકંપની છે) અને 1995માં લીઝડિન દ્વારા વિકસાવવામાં આવી હતી.

- 11 માં 1995, ટાઈમેમાગાએ 1995ના દસ શ્રેષ્ઠ ઉત્પાદનોમાંથી જાવોને અસ્વીકાર કર્યો.

- 12 JDK 1.0 23 જાન્યુઆરી, 1996 ના રોજ રિલીઝ કરવામાં આવ્યું હતું. જાવાની પ્રથમ રજૂઆત પછી, ભાષામાં ઘણી વધારાની સુવિધાઓ ઉમેરવામાં આવી છે. હવે Java નો ઉપયોગ વિન્ડોઝ એપ્લિકેશન, વેબ એપ્લિકેશન, એન્ટરપ્રાઇઝ એપ્લિકેશન, મોબી ઇ એપ્લિકેશન, કાર્ડ્સ વગેરેમાં થઈ રહ્યો છે. દરેક નવી આવૃત્તિ જાવામાં નવી સુવિધાઓ ઉમેરે છે.

જાવા સંસ્કરણ ઇતિહાસ

ઘણા જાવા વર્ઝન અત્યાર સુધી બહાર પાડવામાં આવ્યા છે. Java નું વર્તમાન સ્થિર પ્રકાશન JavaSE18 છે.

- 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 (tobe releasedbyMarch2022)

Java SE 8 રિલીઝ થઈ ત્યારથી, Oracle કોર્પોરેશન એક પેટર્નને અનુસરે છે જેમાં દરેક સમ વર્ગન માર્ચ મહિનામાં રિલીઝ થાય છે અને સપ્ટેમ્બર મહિનામાં એક વિષમ વર્ગન રિલીઝ થાય છે.

જાવાની વિશેષતાઓ

જાવા પ્રોગ્રામિંગ લેંગ્વેજ બનાવવાનો પ્રાથમિક ઉદ્દેશ્ય તેને પોર્ટેબલ, સરળ અને સુરક્ષિત પ્રોગ્રામિંગ લેંગ્વેજ બનાવવાનો હતો. આ ઉપરાંત, કેટલીક ઉત્તમ સુવિધાઓ પણ છે જે આ ભાષાની લોકપ્રિયતામાં મહત્વપૂર્ણ ભૂમિકા ભજવે છે. જાવાની વિશેષતાઓને Java buzzwords તરીકે પણ ઓળખવામાં આવે છે.

જાવા ભાષાના સૌથી મહત્વપૂર્ણ લક્ષણોની સૂચિ નીચે આપેલ છે.

- 1 Simple
- 2 Object-Oriented
- 3 Portable
- 4 Platform independent
- 5 Secured
- 6 Robust
- 7 Architecture neutral
- 8 Interpreted
- 9 High Performance
- 10 Multithreaded
- 11 Distributed
- 12 Dynamic

સરળ

Java શીખવા માટે ખૂબ જ સરળ છે, અને તેની વાક્યરચના સરળ, સ્વચ્છ અને સમજાવામાં સરળ છે. સન માઈક્રોસિસ્ટમ મુજબ, જાવા ભાષા એક સરળ પ્રોગ્રામિંગ ભાષા છે કારણ કે:



- જાવા સિન્ટેક્સ C++ પર આધારિત છે (પ્રોગ્રામર્સ માટે C++ પછી તેને શીખવું સરળ છે).
- Java એ ઘણી જટિલ અને ભાગ્યે જ ઉપયોગમાં લેવાતી વિશેષતાઓને દૂર કરી છે, ઉદાહરણ તરીકે, સ્પષ્ટ નિર્દેશકો, ઓપરેટર ઓવરલોડિંગ વગેરે.
- બિનસંદર્ભિત વસ્તુઓને દૂર કરવાની જરૂર નથી કારણ કે જાવામાં ઓટોમેટિક ગાર્બેજ કલેક્શન છે.

પદાર્થ કેન્દ્રિત

જાવા એક ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ લેંગ્વેજ છે. જાવામાં દરેક વસ્તુ એક વસ્તુ છે. ઓબ્જેક્ટ-ઓરિએન્ટેડ એટલે અમે અમારા સોફ્ટવેરને વિવિધ પ્રકારના ઓબ્જેક્ટના સંયોજન તરીકે ગોઠવીએ છીએ જેમાં ડેટા અને વર્તન બંનેનો સમાવેશ થાય છે.

ઓબ્જેક્ટ-ઓરિએન્ટેડ પ્રોગ્રામિંગ (OOPs) એ એક પદ્ધતિ છે જે કેટલાક નિયમો આપીને સોફ્ટવેર ડેવલપમેન્ટ અને જાળવણીને સરળ બનાવે છે.

OOPs ની મૂળભૂત વિભાવનાઓ છે :

- 1 Object
- 2 Class
- 3 Inheritance
- 4 Polymorphism
- 5 Abstraction
- 6 Encapsulation



પ્લેટફોર્મ સ્વતંત્ર

જાવા પ્લેટફોર્મ સ્વતંત્ર છે કારણ કે તે C, C++, વગેરે જેવી અન્ય ભાષાઓથી અલગ છે જે પ્લેટફોર્મ વિશિષ્ટ મશીનોમાં કમ્પાઈલ કરવામાં આવે છે જ્યારે જાવા એ એક વાર લખવા માટે, ગમે ત્યાંની ભાષા ચલાવો. પ્લેટફોર્મ એ હાર્ડવેર અથવા સોફ્ટવેર વાતાવરણ છે જેમાં પ્રોગ્રામ ચાલે છે.

સોફ્ટવેર આધારિત અને હાર્ડવેર આધારિત બે પ્રકારના પ્લેટફોર્મ છે. જાવા સોફ્ટવેર આધારિત પ્લેટફોર્મ પૂરું પાડે છે.

જાવા પ્લેટફોર્મ એ અર્થમાં મોટાભાગના અન્ય પ્લેટફોર્મ્સથી અલગ છે કે તે એક સોફ્ટવેર-આધારિત પ્લેટફોર્મ છે જે અન્ય હાર્ડવેર-આધારિત પ્લેટફોર્મની ટોચ પર ચાલે છે. તેના બે ઘટકો છે.

1 Runtime Environment

2 API (Application Programming Interface)

જાવા કોડ બહુવિધ પ્લેટફોર્મ્સ પર એક્ઝિક્યુટ કરી શકાય છે, ઉદાહરણ તરીકે, વિન્ડોઝ, લિનક્સ, સન સોલારિસ, મેક/ઓએસ, વગેરે. જાવા કોડ કમ્પાઈલર દ્વારા કમ્પાઈલ કરવામાં આવે છે અને બાયટેકોડમાં રૂપાંતરિત થાય છે. આ બાઈટકોડ એક પ્લેટફોર્મ સ્વતંત્ર કોડ છે કારણ કે તે બહુવિધ પ્લેટફોર્મ પર ચલાવી શકાય છે, એટલે કે, એકવાર લખો અને ગમે ત્યાં ચલાવો (WORA).

સુરક્ષિત

જાવા તેની સુરક્ષા માટે જાણીતું છે. જાવા સાથે, અમે વાયરસ-મુક્ત સિસ્ટમો વિકસાવી શકીએ છીએ. જાવા સુરક્ષિત છે કારણ કે:

- No explicit pointer
- Java Programs run inside a virtual machine sandbox



- **ક્લાસલોડર** : જાવામાં ક્લાસલોડર એ Java રનટાઈમ એન્વાયર્નમેન્ટ (JRE) નો એક ભાગ છે જેનો ઉપયોગ જાવા વર્ચ્યુઅલ મશીનમાં ગતિશીલ રીતે જાવા વર્ગોને લોડ કરવા માટે થાય છે. તે સ્થાનિક ફાઈલ સિસ્ટમના વર્ગો માટેના પેકેજને નેટવર્ક સ્ત્રોતોમાંથી આયાત કરેલા પેકેજમાંથી અલગ કરીને સુરક્ષા ઉમેરે છે.
- **બાઈટકોડ વેરિફાયર** : તે ગેરકાયદેસર કોડ માટે કોડ ટુકડાઓ તપાસે છે જે ઑબ્જેક્ટના એક્સેસ અધિકારોનું ઉલ્લંઘન કરી શકે છે.
- **સુરક્ષા વ્યવસ્થાપક** : તે નક્કી કરે છે કે વર્ગ કયા સંસાધનોને એક્સેસ કરી શકે છે જેમ કે સ્થાનિક ડિસ્ક પર વાંચન અને લેખન.

જાવા ભાષા મૂળભૂત રીતે આ સિક્યોરિટીઝ પ્રદાન કરે છે. એપ્લિકેશન ડેવલપર દ્વારા SSL, JAAS, ક્રિપ્ટોગ્રાફી વગેરે દ્વારા સ્પષ્ટપણે કેટલીક સુરક્ષા પણ પૂરી પાડવામાં આવી શકે છે.

મજબૂત

રોબસ્ટનું અંગ્રેજી માઈનિંગ મજબૂત છે. જાવા મજબૂત છે કારણ કે:

- તે મજબૂત મેમરી મેનેજમેન્ટનો ઉપયોગ કરે છે.

- ત્યાં પોઈન્ટર્સનો અભાવ છે જે સુરક્ષા સમસ્યાઓને ટાળે છે.
- જાવા આપોઆપ કચરો સંગ્રહ પૂરો પાડે છે જે જાવા વર્ચ્યુઅલ મશીન પર ચાલે છે તે વસ્તુઓથી છુટકારો મેળવવા માટે જે હવે જાવા એપ્લિકેશન દ્વારા ઉપયોગમાં લેવાતી નથી.
- જાવામાં અપવાદ હેન્ડલિંગ અને પ્રકાર તપાસવાની પદ્ધતિ છે. આ તમામ બિંદુઓ જાવાને મજબૂત બનાવે છે.

આર્કિટેક્ચર તટસ્થ

જાવા આર્કિટેક્ચર તટસ્થ છે કારણ કે ત્યાં કોઈ અમલીકરણ આધારિત સુવિધાઓ નથી, ઉદાહરણ તરીકે, આદિમ પ્રકારોનું કદ નિશ્ચિત છે.

C પ્રોગ્રામિંગમાં, int ડેટા પ્રકાર 32-બીટ આર્કિટેક્ચર માટે 2 બાઈટ્સ મેમરી અને 64-બીટ આર્કિટેક્ચર માટે 4 બાઈટ્સ મેમરી ધરાવે છે. જો કે, તે જાવામાં 32 અને 64-બીટ આર્કિટેક્ચર બંને માટે 4 બાઈટ્સ મેમરી ધરાવે છે.

પોર્ટેબલ

જાવા પોર્ટેબલ છે કારણ કે તે તમને જાવા બાયટેકોડને કોઈપણ પ્લેટફોર્મ પર લઈ જવાની સુવિધા આપે છે. તેને કોઈ અમલીકરણની જરૂર નથી.

સારો પ્રદર્શન

જાવા અન્ય પરંપરાગત અર્થઘટન કરાયેલ પ્રોગ્રામિંગ ભાષાઓ કરતાં ઝડપી છે કારણ કે જાવા બાયટેકોડ મૂળ કોડની “નજીક” છે. તે હજુ પણ સંકલિત ભાષા (દા.ત., C++) કરતાં થોડી ધીમી છે. જાવા એક અર્થઘટન ભાષા છે તેથી જ તે સંકલિત ભાષાઓ કરતાં ધીમી છે, દા.ત., C, C++, વગેરે.

વિતરણ કર્ચુ

જાવા વિતરિત કરવામાં આવે છે કારણ કે તે વપરાશકર્તાઓને જાવામાં વિતરિત એપ્લિકેશનો બનાવવા માટે સુવિધા આપે છે. RMI અને EJB નો ઉપયોગ વિતરિત એપ્લિકેશન બનાવવા માટે થાય છે. જાવાની આ વિશેષતા અમને ઈન્ટરનેટ પર કોઈપણ મશીનમાંથી પદ્ધતિઓ કોલ કરીને ફાઈલોને એક્સેસ કરવા સક્ષમ બનાવે છે.

મલ્ટી-થ્રેડેડ

થ્રેડ એક અલગ પ્રોગ્રામ જેવો છે, જે એકસાથે એક્ઝેક્યુટ કરે છે. અમે જાવા પ્રોગ્રામ્સ લખી શકીએ છીએ જે એક સાથે અનેક થ્રેડો વ્યાખ્યાયિત કરીને ઘણા કાર્યો સાથે વ્યવહાર કરે છે. મલ્ટી-થ્રેડિંગનો મુખ્ય ફાયદો એ છે કે તે દરેક થ્રેડ માટે મેમરીને રોકતું નથી. તે સામાન્ય મેમરી વિસ્તાર વહેંચે છે. મલ્ટી-મીડિયા, વેબ એપ્લિકેશન્સ વગેરે માટે થ્રેડો મહત્વપૂર્ણ છે.

ગતિશીલ

જાવા એક ગતિશીલ ભાષા છે. તે ડાયનેમિકને સપોર્ટ કરે છે વર્ગોનું લોડિંગ. તેનો અર્થ એ છે કે વર્ગો માંગ પર લોડ થાય છે. તે તેની મૂળ ભાષાઓના કાર્યોને પણ સપોર્ટ કરે છે, એટલે કે, C અને C++.

જાવા ડાયનેમિક કમ્પાઈલેશન અને ઓટોમેટિક મેમરી મેનેજમેન્ટ (કચરો સંગ્રહ) ને સપોર્ટ કરે છે.

JAVA પ્રોગ્રામ લખવાનું નિર્દેશન કરો (Demonstrate writing JAVA programs)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- java વર્ચ્યુઅલ મશીન
- JVM નું આંતરિક આર્કિટેક્ચર.

JVM (જાવા વર્ચ્યુઅલ મશીન) આર્કિટેક્ચર

JVM (જાવા વર્ચ્યુઅલ મશીન) એ એબ્સ્ટ્રેક્ટ મશીન છે. તે એક સ્પષ્ટીકરણ છે જે રનટાઇમ વાતાવરણ પૂરું પાડે છે જેમાં જાવા બાઈટકોડ એક્ઝિક્યુટ થઈ શકે છે.

JVM ઘણા હાર્ડવેર અને સોફ્ટવેર પ્લેટફોર્મ માટે ઉપલબ્ધ છે (એટલે કે JVM પ્લેટફોર્મ આધારિત છે).

JVM શું છે

તે છે:

- 1 સ્પષ્ટીકરણ જ્યાં જાવા વર્ચ્યુઅલ મશીનનું કાર્ય નિર્દિષ્ટ કરેલ છે. પરંતુ અમલીકરણ પ્રદાતા એલ્ગોરિધમ પસંદ કરવા માટે સ્વતંત્ર છે. તેનો અમલ ઓરેકલ અને અન્ય કંપનીઓ દ્વારા આપવામાં આવ્યો છે.
- 2 એક અમલીકરણ તેના અમલીકરણને JRE (જાવા રનટાઇમ એન્વાયર્નમેન્ટ) તરીકે ઓળખવામાં આવે છે.
- 3 રનટાઇમ ઇન્સ્ટન્સ જ્યારે પણ તમે java ક્લાસ ચલાવવા માટે કમાન્ડ પ્રોમ્પ્ટ પર java કમાન્ડ લખો છો, ત્યારે JVM નું ઇન્સ્ટન્સ બનાવવામાં આવે છે.

તે શું કરે છે

JVM નીચેની કામગીરી કરે છે:

- કોડ લોડ કરે છે
- કોડની ચકાસણી કરે છે
- કોડ ચલાવે છે
- રનટાઇમ વાતાવરણ પૂરું પાડે છે

JVM આ માટે વ્યાખ્યાઓ પ્રદાન કરે છે:

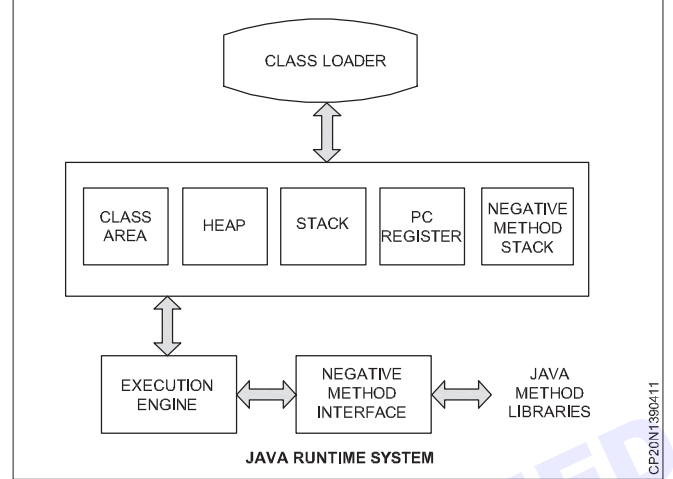
- મેમરી વિસ્તાર
- વર્ગ ફાઇલ ફોર્મેટ
- રજીસ્ટર સેટ
- કચરાનો ઢગલો
- ઘાતક ભૂલની જાણ કરવી વગેરે.

JVM આર્કિટેક્ચર

ચાલો JVM ના આંતરિક આર્કિટેક્ચરને સમજીએ. તેમાં ક્લાસલોડર, મેમરી એરિયા, એક્ઝિક્યુશન એન્જિન વગેરે છે.

1 ક્લાસલોડર

ક્લાસલોડર એ JVM નું સબસિસ્ટમ છે જેનો ઉપયોગ ક્લાસ ફાઇલો લોડ કરવા માટે થાય છે. જ્યારે પણ આપણે જાવા પ્રોગ્રામ ચલાવીએ છીએ, તે ક્લાસલોડર દ્વારા પ્રથમ લોડ થાય છે. જાવામાં ત્રણ બિલ્ટ-ઇન ક્લાસલોડર છે.



- 1 બુટસ્ટ્રેપ ક્લાસલોડર : આ પહેલું ક્લાસલોડર છે જે એક્સટેન્શન ક્લાસલોડરનો સુપર ક્લાસ છે. તે rt.jar ફાઇલને લોડ કરે છે જેમાં java.lang પેકેજ વર્ગો, java.net પેકેજ વર્ગો, java.util પેકેજ વર્ગો, java.io પેકેજ વર્ગો, java.sql પેકેજ વર્ગો વગેરે જેવી જાવા સ્ટાન્ડર્ડ એડિશનની તમામ ક્લાસ ફાઇલોનો સમાવેશ થાય છે.
- 2 એક્સ્ટેન્શન ક્લાસલોડર : આ બુટસ્ટ્રેપનું ચાઇલ્ડ ક્લાસલોડર અને સિસ્ટમ ક્લાસલોડરનું પેરેન્ટ ક્લાસલોડર છે. તે \$JAVA_HOME/jre/lib/ext ડિરેક્ટરીમાં સ્થિત જાર ફાઇલોને લોડ કરે છે.
- 3 સિસ્ટમ/એપ્લીકેશન એન ક્લાસલોડર : આ એક્સ્ટેન્શન ક્લાસલોડરનું ચાઇલ્ડ ક્લાસલોડર છે. તે ક્લાસપાથમાંથી ક્લાસફાઇલો લોડ કરે છે. મૂળભૂત રીતે, વર્ગપાથ વર્તમાન નિર્દેશિકા પર સેટ છે. તમે “cp” અથવા “-classpath” સ્વિચનો ઉપયોગ કરીને ક્લાસપાથ બદલી શકો છો. તેને એપ્લિકેશન ક્લાસલોડર તરીકે પણ ઓળખવામાં આવે છે.

- 1 //Let's see an example to print the classloader name
- 2 public class ClassLoaderExample
- 3 {
- 4 public static void main(String[] args)
- 5 {
- 6 // Let's print the classloader name of the current class.
- 7 //Application/System classloader will load this class
- 8 Class c=ClassLoaderExample.class;
- 9 System.out.println(c.getClassLoader());
- 10 //If we print the classloader name of String, it will printnull because it is an

```

11 //in-built class which is found in rt.jar, so it is loaded by-
    Bootstrap classloader
12 System.out.println (String.class.getClassLoader());
13 }
14 }

```

આઉટપુટ :

આ જાવા દ્વારા પૂરા પાડવામાં આવેલ આંતરિક ક્લાસલોડર્સ છે. જો તમે તમારું પોતાનું ક્લાસલોડર બનાવવા માંગો છો, તો તમારે ClassLoader ક્લાસને વિસ્તારવાની જરૂર છે.

વર્ગ (પદ્ધતિ) વિસ્તાર

વર્ગ(પદ્ધતિ) એરિયા સ્ટોર્સ પ્રતિ ક્લાસ સ્ટ્રક્ચર્સ જેમ કે રનટાઈમ કોન્સ્ટન્ટ પૂલ, ફીલ્ડ અને મેથડ ડેટા, પદ્ધતિઓ માટેનો કોડ.

ઢગલો

તે રનટાઈમ ડેટા વિસ્તાર છે જેમાં ઓબ્જેક્ટ્સ ફાળવવામાં આવે છે.

સ્ટેક

જાવા સ્ટેક ફ્રેમ સ્ટોર કરે છે. તે સ્થાનિક ચલો અને આંશિક પરિણામો ધરાવે છે, અને પદ્ધતિની વિનંતી અને વળતરમાં ભાગ ભજવે છે.

દરેક થ્રેડમાં એક ખાનગી JVM સ્ટેક હોય છે, જે થ્રેડની જેમ જ બનાવવામાં આવે છે.

જ્યારે પણ કોઈ પદ્ધતિનો ઉપયોગ કરવામાં આવે ત્યારે એક નવી ફ્રેમ બનાવવામાં આવે છે. જ્યારે તેની પદ્ધતિની વિનંતી પૂર્ણ થાય છે ત્યારે ફ્રેમનો નાશ થાય છે.

પ્રોગ્રામ કાઉન્ટર રજીસ્ટર

PC (પ્રોગ્રામ કાઉન્ટર) રજીસ્ટરમાં જાવા વર્ચ્યુઅલ મશીન સૂચનાનું સરનામું છે જે હાલમાં અમલમાં છે.

નેટિવ મેથડ સ્ટેક

તેમાં એપ્લીકેશનમાં વપરાતી તમામ મૂળ પદ્ધતિઓનો સમાવેશ થાય છે.

એક્ઝેક્યુશન એન્જિન

તે સમાવે છે :

- 1 વર્ચ્યુઅલ પ્રોસેસર
- 2 દુભાષિયા : બાયટેકોડ સ્ટ્રીમ વાંચો પછી સૂચનાઓ ચલાવો.
- 3 જસ્ટ-ઇન-ટાઈમ (JIT) કમ્પાઈલર : તેનો ઉપયોગ પ્રદર્શનને સુધારવા માટે થાય છે. JIT બાઈટ કોડના ભાગોનું સંકલન કરે છે જે એક જ સમયે સમાન કાર્યક્ષમતા ધરાવે છે, અને તેથી સંકલન માટે જરૂરી સમય ઘટાડે છે. અહીં, “કમ્પાઈલર” શબ્દ જાવા વર્ચ્યુઅલ મશીન (JVM) ના સૂચના સેટમાંથી ચોક્કસ CPU ના સૂચના સમૂહ સુધીના અનુવાદકનો સંદર્ભ આપે છે.

જાવા નેટિવ ઈન્ટરફેસ : જાવા નેટિવ ઈન્ટરફેસ (JNI) એ એક ફ્રેમવર્ક છે જે C, C++, એસેમ્બલી વગેરે જેવી બીજી ભાષામાં લખેલી અન્ય એપ્લિકેશન સાથે વાતચીત કરવા માટે ઈન્ટરફેસ પૂરું પાડે છે. Java કન્સોલને આઉટપુટ મોકલવા અથવા OS લાઈબ્રેરીઓ સાથે ક્રિયાપ્રતિક્રિયા કરવા માટે JNI ફ્રેમવર્કનો ઉપયોગ કરે છે.

જાવામાં બાઈટકોડ શું છે

ઝાંખી

જાવામાં બાઈટકોડ એ Java વર્ચ્યુઅલ મશીન માટેની સૂચનાઓનો સમૂહ છે જે તેમના અર્થઘટન માટે જવાબદાર છે. જાવા પ્રોગ્રામ કમ્પાઈલ કરવા પર, જાવા બાઈટકોડ જનરેટ થાય છે જે JVM નો ઉપયોગ કરીને કોઈપણ પ્લેટફોર્મ (પ્લેટફોર્મમાં સ્વતંત્ર) પર એક્ઝિક્યુટ કરી શકાય છે. તે .class ફાઈલમાં હાજર છે.

અવકાશ

આ લેખનો હેતુ છે:

- જાવામાં બાઈટકોડનો ખ્યાલ સમજાવો.
- Java માં બાઈટકોડના ફાયદા અને ગેરફાયદાને સમજાવવામાં મદદ કરો.

જાવામાં બાઈટકોડ શું છે?

શું તમે એવી દુનિયાની કલ્પના કરી શકો છો કે જ્યાં આપણે ઓપરેટિંગ સિસ્ટમ, પ્રોસેસર આર્કિટેક્ચર વગેરે જેવા વિવિધ વિશિષ્ટતાઓ સાથે દરેક ઉપકરણ માટે ઘણી વખત કોડ લખવો પડે? તે કંટાળાજનક પ્રક્રિયા હશે. આ બાબતોને ધ્યાનમાં રાખીને, જાવા ડેવલપર્સ રાઈટ વન્સ રીડ એનીલેરનો કોન્સેપ્ટ લઈને આવ્યા, જે જાવા ભાષામાં એક વિશેષતા તરીકે સામેલ થઈ. આ હાંસલ કરવા માટે, અમારા જાવા સોર્સ કોડને કમ્પાઈલ કર્યા પછી બાઈટકોડ તરીકે ઓળખાતો કોડ જનરેટ થાય છે. હવે આપણે તેની જ ચર્ચા કરીશું.

જાવામાં બાઈટકોડ એ Java વર્ચ્યુઅલ મશીન માટેની સૂચનાઓનો સમૂહ છે. જાવા વર્ચ્યુઅલ મશીન, સંક્ષિપ્તમાં JVM તરીકે, કમ્પ્યુટરને જાવામાં લખેલા કોડને ચલાવવા માટે સક્ષમ કરે છે. જ્યારે જાવા પ્રોગ્રામ કમ્પાઈલ કરવામાં આવે છે, ત્યારે બાઈટકોડ જનરેટ થાય છે. તે C++ માં એસેમ્બલરની સમકક્ષ છે.

બાઈટકોડ એ એક પ્લેટફોર્મ સ્વતંત્ર સૂચનાઓનો સમૂહ છે કારણ કે તેનું JVM દ્વારા અર્થઘટન અને અમલ કરવામાં આવે છે. કોડ કે જે બહુવિધ કોમ્પ્યુટર આર્કિટેક્ચર પર કોઈપણ ફેરફાર કર્યા વગર ચાલે છે તેને મશીન અથવા પ્લેટફોર્મ ઈન્ડિપેન્ડન્ટ કોડ કહેવામાં આવે છે.

બાઈટકોડ સૂચનોના નીચા-સ્તર અને ઉચ્ચ-સ્તરના સેટ વચ્ચે આવેલું છે. અમે અમારા કોડને ઉચ્ચ સ્તરની ભાષામાં લખતા હોવાથી, તે બાઈટકોડમાં સંકલિત થાય છે, અને પછીથી, JVM તેને મશીન કોડમાં અર્થઘટન કરે છે, જે એક્ઝિક્યુટ કરવા માટે તૈયાર સૂચનાઓનો નીચા-સ્તરનો સમૂહ છે. તેથી, અમે બાઈટકોડને નિમ્ન-સ્તર અને ઉચ્ચ-સ્તરની ભાષા વચ્ચેનો કોડ કહીએ છીએ.

હકીકત :

જાવા એ સંકલિત અને અર્થઘટન કરેલી ભાષા છે, અન્ય ભાષાઓથી વિપરીત, જે કાં તો સંકલિત અથવા અર્થઘટન કરવામાં આવે છે.

તે કેવી રીતે કામ કરે છે?

ચાલો ઉપરોક્ત પ્રક્રિયાને સમજૂતી સાથે સમજીએ:

- અમે જાવામાં જે કોડ લખીએ છીએ તેને સોર્સ કોડ કહેવામાં આવે છે, જે ઉચ્ચ સ્તરની ભાષામાં લખવામાં આવે છે. ઉચ્ચ-સ્તરની ભાષા એ પ્રોગ્રામર મૈત્રીપૂર્ણ ભાષા છે જેમાં અંગ્રેજીમાં લખાયેલા નિવેદનો છે અને તે માનવ ભાષાઓની નજીક છે. Java ફાઈલનું એક્સ્ટેન્શન “java” છે.
- જ્યારે આપણે પ્રોગ્રામ કમ્પાઈલ કરીએ છીએ, ત્યારે કમ્પાઈલર “java” ફાઈલને કમ્પાઈલ કરે છે અને “.class” ફાઈલ જનરેટ કરે છે. તેમાં બાઈટકોડ છે.

- બાઈટકોડ અમને કોઈપણ અન્ય પ્લેટફોર્મ પર “.class” ફાઈલ ચલાવવા માટે પરવાનગી આપે છે.
- પરંતુ આ બાયટેકોડને તેને ચલાવવા માટે દુભાષિયાની જરૂર છે. અહીં JVM ચિત્રમાં આવે છે. JVM પાસે દુભાષિયા છે. તે કોડ ટુકડે ટુકડે એક્ઝિક્યુટ કરે છે, એટલે કે, એક સમયે એક સ્ટેટમેન્ટ, જ્યાં સુધી તેને કોઈ ભૂલ ન મળે અથવા કોડના અંતને એક્ઝિક્યુટ કરવામાં આવે ત્યાં સુધી.

આથી, બાઈટકોડ નિમ્ન-સ્તરની ભાષા અને ઉચ્ચ-સ્તરની ભાષા વચ્ચે હોવાનું કહેવાય છે.

ચાલો એક ઉદાહરણ જોઈએ -

ધારો કે તમે કેલ્ક્યુલેટર એપ માટે જાવામાં કોડ લખ્યો છે. આ રીતે આપણે તેની પ્રક્રિયા કેવી રીતે થાય છે તે સરળતાથી સમજી શકીએ છીએ.

ફાયદા

પ્લેટફોર્મ સ્વતંત્રતા

જાવા પ્રોજેક્ટ કોઈપણ પ્રકારના ફેરફાર વિના કોઈપણ ઉપકરણ પર સમાન કોડના ટુકડાને ચલાવવાના હેતુથી શરૂ કરવામાં આવ્યો હતો. જાવામાં બાઈટકોડનો આ ખ્યાલ આ લક્ષ્યને હાંસલ કરવામાં મદદ કરે છે. બાઈટકોડ જુદી જુદી સિસ્ટમો માટે અલગ અલગ હોઈ શકે છે, પરંતુ તે JVM દ્વારા તમામ સિસ્ટમો પર એકસરખું અર્થઘટન અને એક્ઝિક્યુટ કરવામાં આવે છે.

બાઈટકોડ એ આવશ્યકપણે મશીન-સ્તરનો કોડ છે જે જાવા વર્ચ્યુઅલ મશીન પર ચાલે છે. જ્યારે પણ વર્ગ લોડ થાય છે, ત્યારે તેને વર્ગ પદ્ધતિ દીઠ બાઈટકોડનો પ્રવાહ મળે છે. પ્રોગ્રામના અમલ દરમિયાન જ્યારે પણ તે પદ્ધતિને બોલાવવામાં આવે છે, ત્યારે તે પદ્ધતિ માટે બાયટેકોડને બોલાવવામાં આવે છે. Javac માત્ર પ્રોગ્રામને કમ્પાઈલ કરતું નથી પણ પ્રોગ્રામ માટે બાયટેકોડ પણ જનરેટ કરે છે. આમ, અમને સમજાયું છે કે બાયટેકોડ અમલીકરણ જાવાને પ્લેટફોર્મ સ્વતંત્ર ભાષા બનાવે છે.

પોર્ટબિલિટી

તે સુવાહ્યતા ઉમેરવામાં મદદ કરે છે જે સુનિશ્ચિત કરે છે કે કોડ વિવિધ ઉપકરણો પર ચાલી શકે છે. આ કહેવત સાથે સારી રીતે પડઘો પાડે છે “એકવાર લખો, ગમે ત્યાં દોડો” સિદ્ધાંત જાવા દ્વારા અમલમાં મૂકવામાં આવે છે. અમારે કોઈપણ અન્ય ઉપકરણ પર ફરીથી કોડ લખવાની જરૂર નથી.

ગેરફાયદા

- બાઈટકોડ દુભાષિયા અથવા JVM વગર ચાલી શકતું નથી. જો કોઈપણ ઉપકરણમાં JVM નથી, તો તે ઉપકરણ પર બાયટેકોડ ચાલશે નહીં.
- બાઈટકોડનું પૃથ્થકરણ કરવું મુશ્કેલ છે કારણ કે તે દ્વિસંગી સ્વરૂપમાં છે અને મનુષ્યો દ્વારા સમજી શકાતું નથી.

નિષ્કર્ષ

- જાવામાં બાઈટકોડ એ Java વર્ચ્યુઅલ મશીન માટેની સૂચનાઓનો સમૂહ છે.
- બાઈટકોડ એ પ્લેટફોર્મ સ્વતંત્ર કોડ છે.
- બાઈટકોડ એ એક કોડ છે જે નિમ્ન-સ્તરની ભાષા અને ઉચ્ચ-સ્તરની ભાષા વચ્ચે આવેલો છે.
- જાવા કોડ કમ્પાઈલ થયા પછી, બાઈટકોડ જનરેટ થાય છે, જે JVM નો ઉપયોગ કરીને કોઈપણ મશીન પર એક્ઝિક્યુટ કરી શકાય છે.

- જ્યારે આપણે પ્રોગ્રામ કમ્પાઈલ કરીએ છીએ, ત્યારે કમ્પાઈલર “.java” ફાઈલને કમ્પાઈલ કરે છે અને “.class” ફાઈલ જનરેટ કરે છે.
- બાઈટકોડને ચલાવવા માટે દુભાષિયાની જરૂર છે; તેથી, JVM દુભાષિયા તરીકે કામ કરે છે.
- જાવા સંકલિત અને અર્થઘટન બંને છે.

Java માં Classpath કેવી રીતે સેટ કરવી?

- Read
- Discuss
- Practice
- Video
- Courses

CLASSPATH એ સ્થાનનું વર્ણન કરે છે જ્યાં તમામ જરૂરી ફાઈલો ઉપલબ્ધ છે જેનો ઉપયોગ એપ્લિકેશનમાં થાય છે. Java કમ્પાઈલર અને JVM (જાવા વર્ચ્યુઅલ મશીન) જરૂરી ફાઈલો શોધવા માટે CLASSPATH નો ઉપયોગ કરે છે. જો CLASSPATH સેટ ન હોય, તો Java કમ્પાઈલર જરૂરી ફાઈલો શોધી શકશે નહીં અને તેથી નીચેની ભૂલ ફેંકશે.

(ભૂલ: મુખ્ય વર્ગ શોધી અથવા લોડ કરી શકાયો નથી (દા.ત. GFG))

CLASSPATH સેટ થવા પર ઉપરોક્ત ભૂલ ઉકેલાઈ જાય છે.

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

```
import java.io.*;
class GFG {
    public static void main(String[] args)
    {
        // prints GeeksForGeeks to the console
        System.out.println("GeekForGeeks!");
    }
}
```

આઉટપુટ

GeekForGeeks!

Windows માં JAVA માં CLASSPATH સેટ કરો

Command Prompt:

set PATH=.;C:\Program Files\Java\JDK1.6.20\bin

નોંધ : સેમી-કોલોન (;) નો ઉપયોગ વિભાજક તરીકે થાય છે અને ઉપરના આદેશમાં ડોટ (.) એ CLASSPATH ની મૂળભૂત કિંમત છે.

GUI :

- 1 Select Start
- 2 Go to the Control Panel

PICTURE

3 Select System and Security

PICTURE

4 Select Advanced System settings

PICTURE

5 Click on Environment Variables

PICTURE

6 Click on New under System Variables

PICTURE

7 Add CLASSPATH as variable name and path of files as a variable value.

PICTURE

Select OK.

Linux પર CLASSPATH સેટ કરો

Command Line:

Find out where you have installed Java, basically, it's in /usr/lib/jvm path. Set the CLASSPATH in /etc/environment using

sudo <editor name> /etc/environment

Add the following lines,

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

નોંધ : કોલોન (:) નો ઉપયોગ અલગ ડિરેક્ટરી તરીકે થાય છે અને ઉપરના આદેશમાં ડોટ (.) એ

CLASSPATH ની ડિફોલ્ટ કિંમત છે.

To check the current CLASSPATH, run

echo \${CLASSPATH}

કીવર્ડ્સ

જાવા કીવર્ડ્સ

જાવા કીવર્ડ્સને અનામત શબ્દો તરીકે પણ ઓળખવામાં આવે છે. કીવર્ડ્સ એ ચોક્કસ શબ્દો છે જે કોડની ચાવી તરીકે કાર્ય કરે છે. જાવાસોથ દ્વારા આ પહેલાથી નિર્ધારિત શબ્દોનો ઉપયોગ બદલાય તેવા અથવા ઓબ્જેક્ટનામ અથવા વર્ગના નામ તરીકે કરી શકાતો નથી.

જાવા કીવર્ડ્સની યાદી

નીચે આપેલા જાવાકીવર્ડ્સ અથવા આરક્ષિત શબ્દોની સૂચિ:

- 1 **એબ્સ્ટ્રેક્ટ :** જાવા એબ્સ્ટ્રેક્ટ કીવર્ડનો ઉપયોગ એબ્સ્ટ્રેક્ટ ક્લાસ જાહેર કરવા માટે થાય છે. એક અમૂર્ત વર્ગ ઇન્ટરફેસમાં અમલીકરણ પ્રદાન કરી શકે છે. તે અમૂર્ત અને બિન-અમૂર્ત પદ્ધતિઓનો ઉપયોગ કરી શકે છે.
- 2 **બુલિયન :** જાવા બુલિયન કીવર્ડનો ઉપયોગ વેરીએબલને બુલિયન પ્રકાર તરીકે જાહેર કરવા માટે થાય છે. તે સાચા અને ખોટા મૂલ્યોને y પકડી શકે છે.
- 3 **બ્રેક :** જાવા બ્રેક કીવર્ડનો ઉપયોગ લૂપ અથવા સ્વિચ સ્ટેટમેન્ટને તોડવા માટે થાય છે. તે નિર્દિષ્ટ પરિસ્થિતિઓમાં પ્રોગ્રામના વર્તમાન પ્રવાહને તોડે છે.
- 4 **બાઈટ :** Java byte કીવર્ડનો ઉપયોગ વેરીએબલ કે જે કી8-બીટડેટા મૂલ્યો જાહેર કરવા માટે થાય છે.
- 5 **કેસ :** જાવા કેસ કીવર્ડનો ઉપયોગ ટેક્સ્ટના બ્લોક્સને માર્ક કરવા માટે સ્વિચ સ્ટેટમેન્ટ સાથે થાય છે.
- 6 **કેચ :** જાવા કેચ કીવર્ડનો ઉપયોગ ટ્રાય સ્ટેટમેન્ટ દ્વારા જનરેટ થયેલા અપવાદોને પકડવા માટે થાય છે. તેનો ઉપયોગ ફક્ત પ્રયાસ કર્યા પછી જ કરવો જોઈએ.
- 7 **char :** Java char કીવર્ડનો ઉપયોગ એવેરીએબલ જાહેર કરવા માટે થાય છે જે સહી ન કરેલા 16 બીટયુનિકોડ અક્ષરો ધરાવે છે.
- 8 **વર્ગ :** જાવાક્લાસ શબ્દનો ઉપયોગ કરીને ફરીથી વર્ગનો ઉપયોગ કરો.
- 9 **ચાલુ રાખો :** જાવા ચાલુ રાખો કીવર્ડનો ઉપયોગ લૂપ ચાલુ રાખવા માટે થાય છે. તે પ્રોગ્રામના વર્તમાન પ્રવાહને ચાલુ રાખે છે અને ઉલ્લેખિત સ્થિતિમાં બાકીના કોડને છોડી દે છે.
- 10 **ડિફોલ્ટ :** જાવડેફોલ કીવર્ડનો ઉપયોગ સ્વીચ સ્ટેટમેન્ટમાં કોડના ડિફોલ્ટ બ્લોકને સ્પષ્ટ કરવા માટે થાય છે.
- 11 **do :** Java do કીવર્ડનો ઉપયોગ લૂપ જાહેર કરવા માટે કંટ્રોલ સ્ટેટમેન્ટમાં થાય છે. તે પ્રોગ્રામના એક ભાગને ઘણી વખત પુનરાવર્તિત કરી શકે છે.
- 12 **ડબલ :** જાવા ડબલ કીવર્ડનો ઉપયોગ ચલ જાહેર કરવા માટે થાય છે જે 64 બીટફ્લોટિંગ પોઈન્ટ નંબર ધરાવે છે.
- 13 **અન્ય :** Java else keyword વૈકલ્પિક શાખાઓના નિષ્કર્ષનો ઉલ્લેખ કરવા માટે ઉપયોગ કરે છે.
- 14 **enum :** Java enum કીવર્ડનો ઉપયોગ સ્થિરાંકોના નિશ્ચિત સમૂહને વ્યાખ્યાયિત કરવા માટે થાય છે. એનમ કન્સ્ટ્રક્ટર હંમેશા ખાનગી અથવા ડિફોલ્ટ હોય છે.
- 15 **એક્સ્ટેન્ડ્સ :** Java extensions કીવર્ડનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે ક્લાસ બીજા ક્લાસ અથવા ઇન્ટરફેસમાંથી આવ્યો છે.
- 16 **ફાઈનલ :** જાવા ફાઈનલ કીવર્ડનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે વેરીએબલ સતત મૂલ્ય ધરાવે છે. તેનો ઉપયોગ avariable સાથે થાય છે. તેનો ઉપયોગ વપરાશકર્તાને પ્રતિબંધિત કરવા માટે થાય છે ચલના મૂલ્યને અપડેટ કરી રહ્યું છે.
- 17 **છેલ્લે :** જાવા છેલ્લે કીવર્ડ ટ્રાય કેચ સ્ટ્રક્ચરમાં કોડના બ્લોકને સૂચવે છે. આ બ્લોક હંમેશા એક્ઝિક્યુટ કરવામાં આવે છે પછી ભલેને કોઈ અપવાદ હેન્ડલ ન હોય.
- 18 **ફ્લોટ :** જાવા ફ્લોટ કી શબ્દનો ઉપયોગ એવેરિએબલ જાહેર કરવા માટે થાય છે જે 32 બીટફ્લોટિંગ પોઈન્ટ નંબર ધરાવે છે.

- 19 **માટે** : કીવર્ડ માટે Java નો ઉપયોગ ફોર લૂપ શરૂ કરવા માટે થાય છે. જ્યારે કેટલીક શરત સાચી બને છે ત્યારે તેનો ઉપયોગ વારંવાર સૂચનાઓ/ફંક્શન ઓબ્જેક્ટના સમૂહને ચલાવવા માટે થાય છે. જો પુનરાવૃત્તિની સંખ્યા નિશ્ચિત છે, તો લૂપ માટે તેને ભલામણ કરવામાં આવે છે.
- 20 **જો** : Java If Keyword શરતનું પરીક્ષણ કરો. It executes the if block if the condition is true.
- 21 **ઈમ્પ્લીમેન્ટેડ** : જાવા ઈમ્પ્લીમેન્ટેડ કીવર્ડનો ઉપયોગ ઈન્ટરફેસને અમલમાં કરવા માટે થાય છે.
- 22 **ઈમ્પોર્ટ** : જાવા ઈમ્પોર્ટ કીવર્ડ વર્ગો અને ઈન્ટરફેસને ઉપલબ્ધ બનાવે છે અને વર્તમાન સ્ત્રોત કોડ માટે સુલભ બનાવે છે.
- 23 **નો દાખલો** : કીવર્ડના Java instance નો ઉપયોગ એ ચકાસવા માટે થાય છે કે શું ઓબ્જેક્ટ સ્પષ્ટ થયેલ વર્ગનો એક દાખલો છે કે જે ઈન્ટરફેસનો અમલ કરે છે.
- 24 **int** : જાવેન્ટ કીવર્ડનો ઉપયોગ વેરીએબલને જાહેર કરવા માટે કરવામાં આવે છે જે 32 બીટસાઈઝ કરી શકે છે પૂર્ણાંક
- 25 **ઈન્ટરફેસ** : Javainter face કીવર્ડનો ઉપયોગ ઈન્ટરફેસ જાહેર કરવા માટે થાય છે. તેમાં માત્ર અમૂર્ત પદ્ધતિઓ હોઈ શકે છે.
- 26 **લાંબી** : જાવલોન્ગકી અથવા 64 બિટિનેજર ધરાવી શકે તેવા ટોડેકલા ઈએ વેરીએબલનો ઉપયોગ કરે છે.
- 27 **મૂળ** : જાવા મૂળ કીવર્ડનો ઉપયોગ JNI(Java Native Interface) નો ઉપયોગ કરીને નેટિવ કોડમાં પદ્ધતિ લાગુ કરવામાં આવે છે તે સ્પષ્ટ કરવા માટે થાય છે.
- 28 **નવા** : Javaneu કીવર્ડનો ઉપયોગ નવા ઓબ્જેક્ટ ખાવા માટે થાય છે.
- 29 **નલ** : જાવા નલ કીવર્ડનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે સંદર્ભ કોઈ પણ વસ્તુનો સંદર્ભ આપતો નથી. તે કચરાના મૂલ્યને દૂર કરે છે.
- 30 **પેકેજ** : જાવા પેકેજ કીવર્ડનો ઉપયોગ જાવા પેકેજ જાહેર કરવા માટે થાય છે જેમાં વર્ગોનો સમાવેશ થાય છે.
- 31 **ખાનગી** : જાવા ખાનગી કીવર્ડ એ એક્સેસ મોડિફાયર છે. તેનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે પદ્ધતિ અથવા ચલ ફક્ત તે વર્ગમાં જ એક્સેસ કરી શકાય છે જેમાં જાહેર કરવામાં આવેલ છે.
- 32 **સુરક્ષિત** : જાવા સંરક્ષિત કીવર્ડ એ એક્સેસ મોડિફાયર છે. તે પેકેજની અંદર અને પેકેજની બહાર એક્સેસિબલ હોઈ શકે છે પરંતુ માત્ર હેરિટેન્સ દ્વારા. તે વર્ગ સાથે લાગુ કરી શકાતું નથી.
- 33 **સાર્વજનિક** : જાવા જાહેર કીવર્ડ એ એક્સેસ મોડિફાયર છે. તેનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે આઈટમ ગમે ત્યાંથી સુલભ છે. અન્ય તમામ સંશોધકોમાં સૌથી વધુ વ્યાપક અવકાશ છે.

- 34 **return** : જાવા રીટર્ન કીવર્ડનો ઉપયોગ મેથડમાંથી પરત આવવા માટે થાય છે જ્યારે તેનું સેક્સેક્યુશન પૂર્ણ થાય છે.
- 35 **શોર્ટ** : જાવા શોર્ટકીવર્ડનો ઉપયોગ વેરીએબલને જાહેર કરવા માટે થાય છે જે 16 બિટિનેજર ધરાવે છે.
- 36 **સ્ટેટિક** : જાવા સ્ટેટિક કીવર્ડનો ઉપયોગ થાય છે સૂચવે છે કે ચલ અથવા પદ્ધતિ એ વર્ગ પદ્ધતિ છે. જાવામાં સ્ટેટિક કીવર્ડ મુખ્યત્વે મેમરીમેનેજમેન્ટ માટે વપરાય છે.
- 37 **સ્ટ્રિક્ટએફપી** : જાવાસ્ટ્રિક્ટફનો ઉપયોગ પોર્ટેબિલિટીની ખાતરી કરવા માટે પ્રતિબંધિત ફ્લોટિંગ ઈન્ટની ગણતરી માટે થાય છે.
- 38 **સુપર** : જાવા સુપર કીવર્ડ એ રેફરન્સ વેરીએબલ છે જેનો ઉપયોગ પેરેન્ટ ક્લાસ ઓબ્જેક્ટનો સંદર્ભ આપવા માટે થાય છે. તેનો ઉપયોગ તાત્કાલિક પારેન વર્ગ પદ્ધતિને બોલાવવા માટે થઈ શકે છે.
- 39 **સ્વિચ** : જાવા સ્વિચ કીવર્ડમાં એક સ્વીચ સ્ટેટમેન્ટ છે જે ટેસ્ટ મૂલ્યના આધારે કોડને એક્ઝેક્યુટ કરે છે. સ્વીચ સ્ટેટમેન્ટ બહુવિધ મૂલ્યો સામે ચલની સમાનતાનું પરીક્ષણ કરે છે.
- 40 **સિંકનાઈઝ** : જાવા સિંકનાઈઝ કીવર્ડનો ઉપયોગ મલ્ટી થ્રેડ કોડમાં જટિલ વિભાગો અથવા પદ્ધતિઓનો ઉલ્લેખ કરવા માટે થાય છે.
- 41 **આ** : જાવા આ કીવર્ડનો ઉપયોગ મેથડ અથવા કન્સ્ટ્રક્ટરમાં વર્તમાન ઓબ્જેક્ટનો સંદર્ભ આપવા માટે કરી શકાય છે.
- 42 **થ્રો** : જાવા થ્રો કીવર્ડનો ઉપયોગ અપવાદને સ્પષ્ટ રીતે ફેંકવા માટે થાય છે. થ્રો કીવર્ડનો ઉપયોગ મુખ્યત્વે કસ્ટમ અપવાદો ફેંકવા માટે થાય છે.
- 43 **થ્રો** : જાવા થ્રો કીવર્ડનો ઉપયોગ અપવાદ જાહેર કરવા માટે થાય છે. ચકાસાયેલ અપવાદો થ્રો સાથે પ્રચાર કરી શકાય છે.
- 44 **ક્ષણિક** : જાવા ક્ષણિક કીવર્ડ સીરીયલાઈઝેશનમાં વપરાય છે. જો તમે કોઈપણ ડેટા સભ્યને ક્ષણિક તરીકે વ્યાખ્યાયિત કરો છો, તો તે ક્રમાંકિત કરવામાં આવશે નહીં.
- 45 **પ્રયાસ કરો** : જાવા ટ્રાય કીવર્ડનો ઉપયોગ કોડના બ્લોકને શરૂ કરવા માટે થાય છે જે અપવાદો માટે પરીક્ષણ કરવામાં આવશે. ટ્રાય બ્લોક પછી કેચ અથવા છેલ્લે બ્લોક દ્વારા અનુસરવું આવશ્યક છે.
- 46 **void** : Java void keyword તે પદ્ધતિમાં વળાંક મૂલ્ય નથી તે સ્પષ્ટ કરવા માટે વપરાય છે.
- 47 **વોલેટાઈલ** : જાવા વોલેટાઈલ કીવર્ડનો ઉપયોગ એ દર્શાવવા માટે થાય છે કે એવેરીએબલ સરળ રીતે બદલાઈ શકે છે.
- 48 **while** : Java while કીવર્ડનો ઉપયોગ જ્યારે લૂપ શરૂ કરવા માટે થાય છે. આ લૂપ પ્રોગ્રામના એક ભાગને ઘણી વખત પુનરાવર્તિત કરે છે. જો પુનરાવૃત્તિની સંખ્યા નિશ્ચિત ન હોય, તો તે જ્યારે લૂપનો ઉપયોગ કરવાની ભલામણ કરે છે.

જાવામાં ડેટાટાઈપ્સ (Datatypes in Java)

ઉદ્દેશ્યો : આ પાઠના અંતે તમે સક્ષમ થશો

- ડેટાટાઈપ્સ પ્રકાર
- java આદિમ ડેટા પ્રકારો
- બિન-આદિમ ડેટા પ્રકારો

જાવામાં ડેટા પ્રકારો

ડેટા પ્રકારો વિવિધ કદ અને મૂલ્યોનો ઉલ્લેખ કરે છે જે ચલમાં સંગ્રહિત કરી શકાય છે. જાવામાં બે પ્રકારના ડેટાટાઈપ્સ છે:

આદિમ ડેટા પ્રકારો : આદિમ ડેટા પ્રકારોમાં બુલિયન, ચાર, બાઈટ, શોર્ટ, ઈન્ટ, લોંગ, ફ્લોટ અને ડબલનો સમાવેશ થાય છે.

બિન-આદિમ ડેટા પ્રકારો તેનન-આદિમ ડેટા પ્રકારોમાં વર્ગો, ઈન્ટરફેસ અને એરેનો સમાવેશ થાય છે.

જાવા આદિમ ડેટા પ્રકારો

જાવા ભાષામાં, આદિમ ડેટા પ્રકારો ડેટા મેનીપ્યુલેશનના બિલ્ડિંગ બ્લોક્સ છે. જાવા ભાષામાં આ સૌથી મૂળભૂત ડેટાટાઈપ લેબલ છે.

નોંધ : જાવા એ સ્ટેટિકલી-ટાઈપ કરેલી પ્રોગ્રામિંગ ભાષા છે. તેનો અર્થ એ છે કે, બધા વેરીએબલ્સ તેના ઉપયોગ પહેલા જાહેર કરવા જોઈએ. એટલા માટે અમે `edtoDeclarevari ble` નું ટાઈપ અને નામ.

8 પ્રકારના આદિમ ડેટા પ્રકારો છે:

1 booleandatatype

2 bytedatatype

3 chardatatype

4 shortdatatype

5 intdatatype

6 longdatatype

7 floatdatatype

8 doubledatatype



જાવા ડેટા પ્રકારો

ડેટા પ્રકાર	ડિફોલ્ટ મૂલ્ય	ડિફોલ્ટસાઈઝ
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

બુલિયન ડેટા પ્રકાર

બુલિયન ડેટા પ્રકારનો ઉપયોગ ફક્ત બે સંભવિત મૂલ્યોને સંગ્રહિત કરવા માટે થાય છે: સાચું અને ખોટું. આ ડેટા પ્રકારનો ઉપયોગ સાધારણ ફ્લેગસ્થિત્ર `ktrue/false` શરતો માટે થાય છે.

બુલિયન ડેટા પ્રકાર થોડી માહિતીનો ઉલ્લેખ કરે છે, પરંતુ તેનું “કદ” ચોક્કસ રીતે વ્યાખ્યાયિત કરી શકાતું નથી.

ઉદાહરણ :

Booleanone =false

બાઈટ ડેટા પ્રકાર

બાઈટ ડેટા પ્રકાર એ આદિમ ડેટા પ્રકારનું ઉદાહરણ છે. તે 8-બીટ સહી કરેલ બેના પૂરક પૂર્ણાંક છે. તેની મૂલ્ય શ્રેણી -128 થી 127 (સમાવિષ્ટ) ની વચ્ચે છે. તેનું ન્યૂનતમ મૂલ્ય -128 છે અને મહત્તમ મૂલ્ય 127 છે. તેનું ડિફોલ્ટ મૂલ્ય 0 છે.

બાઈટ ડેટા પ્રકારનો ઉપયોગ મોટી એરેમાં મેમરીને સાચવવા માટે થાય છે જ્યાં મેમરી બચત સૌથી વધુ જરૂરી હોય છે. તે જગ્યા બચાવે છે કારણ કે `abyte` પૂર્ણાંક કરતા 4 ગણું નાનું છે. તે “int” ડેટાટાઈપની જગ્યાએ પણ વાપરી શકાય છે.

ઉદાહરણ :

byte a = 10,byteb = -20

ટૂંકા ડેટા પ્રકાર

ટૂંકા ડેટા પ્રકાર એ 16-બીટ હસ્તાક્ષરિત બેના પૂરક પૂર્ણાંક છે. તેની મૂલ્ય શ્રેણી -32,768 થી 32,767 (સમાવિષ્ટ) ની વચ્ચે છે. લઘુત્તમ મૂલ્ય -32,768 અને મહત્તમ મૂલ્ય 32,767 છે.

બાઈટ ડેટા ટાઈપની જેમ મેમરીને બચાવવા માટે ટૂંકા ડેટા પ્રકારનો પણ ઉપયોગ કરી શકાય છે. ટૂંકા ડેટા પ્રકાર પૂર્ણાંક કરતા 2 ગણો નાનો છે.

ઉદાહરણ:

shorts =10000,shortr=-5000

Int ડેટા પ્રકાર

પૂર્ણાંક ડેટા પ્રકાર એ 32-બીટ સહી કરેલ બેના પૂરક પૂર્ણાંક છે. તેની મૂલ્ય શ્રેણી - 2,147,483,648 (- 2³¹) થી 2,147,483,647 (2³¹ -1) (સમાવિષ્ટ) ની વચ્ચે છે. તેનું ન્યૂનતમ મૂલ્ય છે - 2,147,483,648 અને મહત્તમ મૂલ્ય 2,147,483,647 છે. તેનું ડિફોલ્ટ મૂલ્ય 0 છે.

int ડેટા પ્રકાર સામાન્ય રીતે ઇન્ટિગ્રલ વેલ્યુ માટે ડિફોલ્ટ ડેટા પ્રકાર તરીકે ઉપયોગમાં લેવાય છે સિવાય કે મેમરી વિશે કોઈ સમસ્યા ન હોય.

ઉદાહરણ :

inta= 100000,intb =-200000

લાંબો ડેટા પ્રકાર

લાંબો ડેટા પ્રકાર એ 64-બીટ બેના પૂરક પૂર્ણાંક છે. તેની મૂલ્ય શ્રેણી -9,223,372,036,85 ,775,808 (-2⁶³) થી 9,223,3 2,036,854,775,807 2⁶³-1) (સમાવિષ્ટ) ની વચ્ચે છે. તેનું ન્યૂનતમ મૂલ્ય

9,223,372,036,85 છે અને મહત્તમ મૂલ્ય 9,223,737,367,375,375,367,365,375,36,375 છે. તેનું ડિફોલ્ટ મૂલ્ય 0 છે. લાંબા ડેટાટાઈપિસસ જ્યારે તમને જરૂર હોય મૂલ્યોની શ્રેણી જે int દ્વારા પ્રદાન કરવામાં આવે છે તેના કરતાં વધુ.

ઉદાહરણ :

```
long a=1000000L,long b=-2000000L
```

ફ્લોટ ડેટા પ્રકાર

ફ્લોટ ડેટા પ્રકાર એ સિંગલ પ્રિસિઝન 32-બીટ IEEE 754 ફ્લોટિંગ પોઈન્ટ છે. તેની કિંમત શ્રેણી અમર્યાદિત છે. જો તમારે ફ્લોટિંગ પોઈન્ટ નંબર્સના મોટા એરેમાં મેમરીને સાચવવાની જરૂર હોય તો ફ્લોટ (ડબલને બદલે) નો ઉપયોગ કરવાની ભલામણ કરવામાં આવે છે. ફ્લોટ ડેટા પ્રકારનો ઉપયોગ ચોક્કસ એલ્યુઝ માટે ક્યારેય થવો જોઈએ નહીં, જેમ કે ચલણ. તેનું મૂળભૂત મૂલ્ય 0.0F છે.

ઉદાહરણ :

```
float f1=234.5f
```

ડબલ ડેટા પ્રકાર

ડબલ ડેટા પ્રકાર એ ડબલ પ્રિસિઝન 64-બીટ IEEE 754 ફ્લોટિંગ પોઈન્ટ છે. તેની કિંમત શ્રેણી અમર્યાદિત છે. ડબલ ડેટા પ્રકારનો ઉપયોગ સામાન્ય રીતે ફ્લોટની જેમ દશાંશ મૂલ્યો માટે થાય છે. ડબલ ડેટા પ્રકારનો પણ ચોક્કસ મૂલ્યો માટે ક્યારેય ઉપયોગ થવો જોઈએ નહીં, જેમ કે ચલણ. તેનું ડિફોલ્ટ મૂલ્ય 0.0d છે.

ઉદાહરણ :

```
doubled1=12.3
```

જાવા ઓપરેટર અગ્રતા

ઓપરેટર પ્રકાર	શ્રેણી	અગ્રતા
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	= += -= *= /= %= &= ^= = <<= >>= >>>=

ચાર ડેટા પ્રકાર

ચાર ડેટા પ્રકાર એ એક 16-બીટ યુનિકોડ અક્ષર છે. તેની મૂલ્ય-શ્રેણી 'u0000' (અથવા 0) થી '\uffff' (અથવા 65,535સહિત). ચાર્ક્ટા પ્રકારનો ઉપયોગ સ્ટોર અક્ષરો માટે થાય છે.

ઉદાહરણ :

```
charletterA='A'
```

Whycharuses2byte in java and what is\u0000?

કારણ કે જાવા એએસસીઆઈઆઈ કોડ સિસ્ટમ નહીં પણ યુનિકોડ સિસ્ટમનો ઉપયોગ કરે છે. \u0000 યુનિકોડ સિસ્ટમની સૌથી ઓછી શ્રેણી છે. યુનિકોડ વિશે વિગતવાર સમજૂતી મેળવવા માટે આગલા પૃષ્ઠની મુલાકાત લો.

જાવામાં ઓપરેટરો

જાવામાં ઓપરેટર એ એક પ્રતીક છે જેનો ઉપયોગ કામગીરી કરવા માટે થાય છે. ઉદાહરણ તરીકે: +, -, *, / વગેરે. જાવામાં ઘણા પ્રકારના ઓપરેટરો છે જે નીચે આપેલા છે:

- UnaryOperator,
- ArithmeticOperator,
- ShiftOperator,
- RelationalOperator,
- BitwiseOperator,
- LogicalOperator,
- TernaryOperator and
- AssignmentOperator.

જાવા યુનરી ઓપરેટર

જાવા યુનરી ઓપરેટરોને માત્ર એક ઓપરેન્ડની જરૂર છે. યુનરી ઓપરેટર્સનો ઉપયોગ વિવિધ કામગીરી કરવા માટે થાય છે.

- મૂલ્યમાં વધારો/ઘટાડો
- નેગેટીવ એક્સપ્રેસન
- અબુલિયનનું ઉલટાનું મૂલ્ય

JavaUnaryOperatorExample:++and --

```
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  
    }  
}
```

આઉટપુટ :

10
12
12
10

JavaUnaryOperatorExample2:++and--

```
public class OperatorExample{  
    public static void main(String args[]){  
        int a=10;  
        int b=10;  
        System.out.println(a++ + ++a);//10+12=22  
        System.out.println(b+++b++);//10+11=21  
    }  
}
```

આઉટપુટ :

22
21

જાવા યુનરી ઓપરેટર ઉદાહરણ: ~અને!

```
public class OperatorExample{  
    public static void main(String args[]){  
        int a=10;  
        int b=-10;  
        boolean c=true;  
        boolean d=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

}}

આઉટપુટ :

-11

9

false

true

જાવા અંકગણિત ઓપરેટર્સ

જાવા અંકગણિત ઓપરેટર્સનો ઉપયોગ સરવાળા, બાદબાકી, ગુણાકાર અને ભાગાકાર કરવા માટે થાય છે. તેઓ મૂળભૂત ગાણિતિક કામગીરી તરીકે કાર્ય કરે છે.

ત ઓપરેટર ઉદાહરણ

```
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  
    }  
}
```

આઉટપુટ :

15

5

50

2

0

જાવા અંકગણિત ઓપરેટર ઉદાહરણ: અભિવ્યક્તિ

```
public class OperatorExample{  
    public static void main(String args[]){  
        System.out.println(10*10/5+3-1*4/2);  
    }  
}
```

આઉટપુટ :

21

જાવા લેફ્ટશિફ્ટ ઓપરેટર

Java લેફ્ટ શિફ્ટ ઓપરેટર << નો ઉપયોગ ઘણી વખત નિર્દિષ્ટ સંખ્યાની ડાબી બાજુએ મૂલ્યમાંના તમામ બિટ્સને શિફ્ટ કરવા માટે થાય છે.

જાવા લેફ્ટ શિફ્ટ ઓપરેટર

```
ExamplepublicclassOperatorExample{
    publicstaticvoidmain(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
    }
}
```

આઉટપુટ :

```
40
80
80
240
```

જાવા રાઈટશિફ્ટ ઓપરેટર

જાવા રાઈટ શિફ્ટ ઓપરેટર >> નો ઉપયોગ જમણા ઓપરેન્ડ દ્વારા ઉલ્લેખિત બિટ્સની સંખ્યા દ્વારા ડાબા ઓપરેન્ડના મૂલ્યને જમણે ખસેડવા માટે થાય છે.

જાવા રાઈટ શિફ્ટ ઓપરેટરનું ઉદાહરણ

```
publicOperatorExample{
    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
    }
}
```

આઉટપુટ :

```
2
5
2
```

Java Shift ઓપરેટર ઉદાહરણ:>>>vs>>>

```
publicclassOperatorExample{
    publicstaticvoidmain(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);
```

```
}}
```

આઉટપુટ :

```
5
5
-5
1073741819
```

JavaANDOperatorExample:Logical&&andBitwise&

જો પ્રથમ શરત ખોટી હોય તો લોજિકલ && ઓપરેટર બીજી શરત તપાસતા નથી. જો પ્રથમ સાચી હોય તો જ તે બીજી શરત તપાસે છે.

બીટવાઈઝ અને ઓપરેટર હંમેશા બંને શરતો તપાસે છે કે પ્રથમ શરત સાચી છે કે ખોટી.

```
publicclass OperatorExample{
    publicstaticvoidmain(Stringargs[]){
        int a=10;int b=5;intc=20;
        System.out.println(a<b&&a<c);//false && true = false
        System.out.println(a<b&a<c);//false&true=false
    }
}
```

આઉટપુટ :

```
false
false
```

JavaANDOperatorExample:Logical&&vsBitwise&

```
publicclassOperatorExample{
    public static void main(String args[]){
        inta=10;
        int b=5;
        intc=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 conditionis
        checked
    }
}
```

આઉટપુટ :

```
false10
false11
```

JavaOOperatorExample:Logical||andBitwise |

તાર્કિક || જો પ્રથમ શરત સાચી હોય તો ઓપરેટર બીજી શરત તપાસતા નથી. જો પ્રથમ ખોટી હોય તો જ તે બીજી શરત તપાસે છે.

બીટવાઈઝ | ઓપરેટર હંમેશા બંને શરતો તપાસે છે કે પ્રથમ શરત સાચી છે કે ખોટી.

```
publicclass OperatorExample{
    public static void main(String args[])
    {
        inta=10;
        intb=5;
        intc=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 conditionis
        not checked
        System.out.println(a>b|a++<c);//true | true = true
        System.out.println(a);//11because second condition
        ischecked
    }
}
```

આઉટપુટ :

```
true
true
true
10
true
11
```

જાવા ટર્નરી ઓપરેટર

જાવા ટર્નરી ઓપરેટરનો ઉપયોગ if-then-else સ્ટેટમેન્ટ અને માટે એક લાઈન રિપ્લેસમેન્ટ તરીકે થાય છેજાવાપ્રોગ્રામિંગમાં ઘણો ઉપયોગ થાય છે તે એકમાત્ર કંડીશનલોપેરા છે અથવા જે ત્રણ ઓપરેન્ડ લે છે.

જાવા ટર્નરી ઓપરેટરનું ઉદાહરણ

```
publicclass Operator Example {
    public static void main (String args[]){inta=2;
    intb=5;
    int min=(a<b)?a:b;
    System.out.println(min);
}
```

આઉટપુટ :

2

બીજું ઉદાહરણ :

```
publicclassOperatorExample{
    public static void main(String args[]){
        inta=10;
        intb=5;
        int min=(a<b)?a:b;System.out.println(min);
    }
}
```

આઉટપુટ :

5

જાવા અસાઈનમેન્ટ ઓપરેટર

જાવા અસાઈનમેન્ટ ઓપરેટર સૌથી સામાન્ય ઓપરેટરોમાંનું એક છે. તેનો ઉપયોગ તેની ડાબી બાજુના ઓપરેન્ડને તેની જમણી બાજુની કિંમત સોંપવા માટે થાય છે.

જાવા અસાઈનમેન્ટ ઓપરેટરનું ઉદાહરણ

```
publicclassOperatorExample{
    public static void main(String args[]){
        inta=10;
        intb=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);
    }
}
```

આઉટપુટ :

14

16

જાવા અસાઈનમેન્ટ ઓપરેટરનું ઉદાહરણ

```
publicclassOperatorExample{
    public static void main(String[] args){
        inta=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);
```

```
}}
```

આઉટપુટ :

```
13
```

```
9
```

```
18
```

```
9
```

જાવા અસાઈનમેન્ટ ઓપરેટર ઉદાહરણ: એડિંગશોર્ટ

```
publicclassOperatorExample{  
    public static void main(String args[]){  
        shorta=10;  
        shortb=10;  
        //a+=b;//a=a+binternallysofine  
        a=a+b;//Compile time error because 10+10=20 nowint  
        System.out.println(a);  
    }  
}
```

આઉટપુટ :

```
ટાઈમ એરર કમ્પાઈલ કરો
```

ટાઈપકાસ્ટ પછી :

```
publicclassOperatorExample  
{  
    public static void main(String args[])  
{  
        shorta=10;  
        shortb=10;  
        a=(short)(a+b);//20 which is int now converted to  
        shortSystem.out.println(a);  
    }  
}
```

આઉટપુટ :

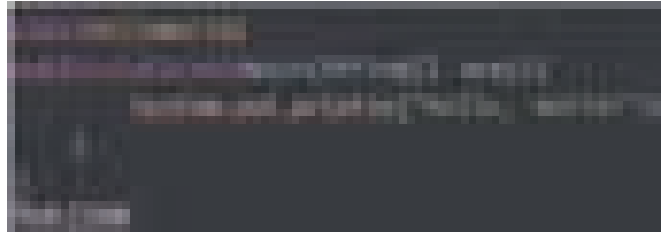
```
20
```

જાવા I/O સ્ટ્રીમ્સ

આ ટ્યુટોરીયલમાં, આપણે જાવા ઇનપુટ/આઉટપુટ સ્ટ્રીમ્સ અને તેમના વિશે શીખીશું પ્રકારો.

જાવામાં, સ્ટ્રીમ્સ એ ડેટાનો ક્રમ છે જે સ્ત્રોતમાંથી વાંચવામાં આવે છે અને ગંતવ્ય સ્થાન પર લખવામાં આવે છે.

સ્ત્રોતમાંથી ડેટા વાંચવા માટે ઇનપુટ સ્ટ્રીમનો ઉપયોગ થાય છે. અને, ગંતવ્ય પર ડેટા લખવા માટે આઉટપુટ સ્ટ્રીમનો ઉપયોગ થાય છે.



ઉદાહરણ તરીકે, અમારા પ્રથમ હેલો વર્લ્ડ ઉદાહરણમાં, અમે સ્ટ્રીંગ પ્રિન્ટ કરવા માટે System.out નો ઉપયોગ કર્યો છે. અહીં, System.out એ આઉટપુટ સ્ટ્રીમનો એક પ્રકાર છે.

તેવી જ રીતે, ઇનપુટ લેવા માટે ઇનપુટ સ્ટ્રીમ્સ છે.

આપણે પછીના ટ્યુટોરિયલ્સમાં ઇનપુટ સ્ટ્રીમ્સ અને આઉટપુટ સ્ટ્રીમ્સ વિશે વિગતવાર શીખીશું.

પ્રવાહોના પ્રકાર

સ્ટ્રીમ જે ડેટા ધરાવે છે તેના આધારે તેને આમાં વર્ગીકૃત કરી શકાય છે:

- બાઈટ સ્ટ્રીમ
- અક્ષર પ્રવાહ

બાઈટ સ્ટ્રીમ

બાઈટ સ્ટ્રીમનો ઉપયોગ ડેટાનો એક બાઈટ (8 બિટ્સ) વાંચવા અને લખવા માટે થાય છે.

બધા બાઈટ સ્ટ્રીમ વર્ગો બેઝ એબ્સ્ટ્રેક્ટ વર્ગોમાંથી ઉતરી આવ્યા છે જેને ઇનપુટસ્ટ્રીમ અને આઉટપુટસ્ટ્રીમ કહેવાય છે.

પાત્ર પ્રવાહ

કેરેક્ટર સ્ટ્રીમનો ઉપયોગ ડેટાના એક અક્ષરને વાંચવા અને લખવા માટે થાય છે.

બધા પાત્ર પ્રવાહ વર્ગો મૂળ અમૂર્ત વર્ગો રીડર અને લેખકમાંથી ઉતરી આવ્યા છે.

જાવા યુઝર ઇનપુટ (સ્કેનર)

જાવા યુઝર ઇનપુટ

સ્કેનર ક્લાસનો ઉપયોગ યુઝર ઇનપુટ મેળવવા માટે થાય છે અને તે java.util પેકેજમાં જોવા મળે છે.

સ્કેનર વર્ગનો ઉપયોગ કરવા માટે, વર્ગનો એક ઓબ્જેક્ટ બનાવો અને કોઈપણ ઉપલબ્ધ પદ્ધતિઓનો ઉપયોગ કરો સ્કેનર વર્ગ દસ્તાવેજીકરણમાં જોવા મળે છે. અમારા ઉદાહરણમાં, અમે NextLine() પદ્ધતિનો ઉપયોગ કરીશું, જેનો ઉપયોગ સ્ટ્રીંગ્સ વાંચવા માટે થાય છે:

ઉદાહરણ

ઉદાહરણ ચલાવો

જો તમને ખબર નથી કે પેકેજ શું છે, તો અમારું Java Packages ટ્યુટોરીયલ વાંચો.

ઇનપુટ પ્રકારો

ઉપરના ઉદાહરણમાં, અમે NextLine() પદ્ધતિનો ઉપયોગ કર્યો છે, જેનો ઉપયોગ સ્ટ્રીંગ્સ વાંચવા માટે થાય છે. અન્ય પ્રકારો વાંચવા માટે, નીચેનું કોષ્ટક જુઓ:



પદ્ધતિ	વર્ણન
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

નીચેના ઉદાહરણમાં, અમે વિવિધ પ્રકારના ડેટા વાંચવા માટે વિવિધ પદ્ધતિઓનો ઉપયોગ કરીએ છીએ:

ઉદાહરણ

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(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);
}
}
```

ઉદાહરણ ચલાવો

નોંધ : જો તમે ખોટો ઈનપુટ દાખલ કરો છો (દા.ત. સંખ્યાત્મક ઈનપુટમાં ટેક્સ્ટ), તો તમને એક અપવાદ/ભૂલ સંદેશ મળશે (જેમ કે "InputMismatchException").

તમે અપવાદો પ્રકરણમાં અપવાદો અને ભૂલોને કેવી રીતે હેન્ડલ કરવી તે વિશે વધુ વાંચી શકો છો.

જાવા કન્સોલ વર્ગ

જાવા કન્સોલ વર્ગનો ઉપયોગ કન્સોલમાંથી ઈનપુટ મેળવવા માટે થાય છે. તે ટેક્સ્ટ્સ અને પાસવર્ડ્સ વાંચવાની પદ્ધતિઓ પ્રદાન કરે છે.

જો તમે કન્સોલ ક્લાસનો ઉપયોગ કરીને પાસવર્ડ વાંચો છો, તો તે વપરાશકર્તાને પ્રદર્શિત કરવામાં આવશે નહીં.

java.io.Console વર્ગ આંતરિક રીતે સિસ્ટમ કન્સોલ સાથે જોડાયેલ છે. કન્સોલ વર્ગ 1.5 થી રજૂ કરવામાં આવ્યો છે.

ચાલો કન્સોલમાંથી ટેક્સ્ટ વાંચવા માટે એક સરળ ઉદાહરણ જોઈએ.

1 સ્ટ્રિંગ ટેક્સ્ટ=System.console().readLine();

2 System.out.println “ટેક્સ્ટ છે: “+ટેક્સ્ટ);

જાવા કન્સોલ વર્ગ ઘોષણા

ચાલો Java.io.Console વર્ગ માટેની ઘોષણા જોઈએ:

1 પબ્લિક ફાઈનલ ક્લાસ કન્સોલ ફ્લશેબલ ઓબ્જેક્ટને લાગુ કરે છે

જાવા કન્સોલ વર્ગ પદ્ધતિઓ

પદ્ધતિ	વર્ણન
Reader reader()	તેનો ઉપયોગ કન્સોલ સાથે સંકળાયેલ રીડર ઓબ્જેક્ટને પુનઃપ્રાપ્ત કરવા માટે થાય છે
String readLine()	તેનો ઉપયોગ કન્સોલમાંથી ટેક્સ્ટની એક લીટી વાંચવા માટે થાય છે.
String readLine(String fmt,Object... args)	તે ફોર્મેટ કરેલ પ્રોમ્પ્ટ પ્રદાન કરે છે પછી કન્સોલમાંથી ટેક્સ્ટની એક લીટી વાંચે છે.
char[] readPassword()	તેનો ઉપયોગ પાસવર્ડ વાંચવા માટે થાય છે જે કન્સોલ પર પ્રદર્શિત થતો નથી.
char[] readPassword (String fmt,Object... args)	તે ફોર્મેટ કરેલ પ્રોમ્પ્ટ પૂરો પાડે છે પછી પાસવર્ડ વાંચે છે જે કન્સોલ પર પ્રદર્શિત થતો નથી.
Console format(String fmt,Object... args)	તેનો ઉપયોગ કન્સોલ આઉટપુટ સ્ટ્રીમ પર ફોર્મેટ કરેલ સ્ટ્રિંગ લખવા માટે થાય છે.
Console printf(String format,Object... args)	તેનો ઉપયોગ કન્સોલ આઉટપુટ સ્ટ્રીમ પર સ્ટ્રિંગ લખવા માટે થાય છે.
PrintWriter writer()	તેનો ઉપયોગ કન્સોલ સાથે સંકળાયેલ PrintWriter ઓબ્જેક્ટને પુનઃપ્રાપ્ત કરવા માટે થાય છે.
void flush()	તેનો ઉપયોગ કન્સોલને ફ્લશ કરવા માટે થાય છે.

કન્સોલનો ઓબ્જેક્ટ કેવી રીતે મેળવવો

સિસ્ટમ ક્લાસ સ્ટેટિક મેથડ કન્સોલ() પ્રદાન કરે છે જે કન્સોલ ક્લાસના સિંગલટન ઇન્સ્ટન્સ પરત કરે છે.

1 સાર્વજનિક સ્થિર કન્સોલ કન્સોલ({}

ચાલો કન્સોલ ક્લાસનો દાખલો મેળવવા માટે કોડ જોઈએ.

1 કન્સોલ c=System.console();

જાવા કન્સોલનું ઉદાહરણ

1 import java.io.Console;

2 class ReadStringTest{

3 public static void main(String args[]){

4 Console c=System.console();

5 System.out.println(“Enter your name:”);

6 String n=c.readLine();

7 System.out.println(“Welcome “+n);

8 }

9 }

આઉટપુટ

```
Enter your name: Nakul Jain
Welcome Nakul Jain
```

પાસવર્ડ વાંચવા માટે જાવા કન્સોલનું ઉદાહરણ

1 import java.io.Console;

2 class ReadPasswordTest{

3 public static void main(String args[]){

4 Console c=System.console();

5 System.out.println(“Enter password:”);

6 char[] ch=c.readPassword();

7 String pass=String.valueOf(ch);//converting char array-into string

8 System.out.println(“Password is: “+pass);

9 }

10 }

આઉટપુટ

```
Enter password:
Password is: 123
```

જાવા પ્રોગ્રામ ફ્લો કંટ્રોલ (JAVA Program Flow Control)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- શરતી નિવેદનો
- આંટીઓ
- એરે.

જાવા શરતો અને જો નિવેદનો

તમે પહેલેથી જ જાણો છો કે જાવા ગણિતની સામાન્ય તાર્કિક પરિસ્થિતિઓને સમર્થન આપે છે:

- Less than: $a < b$
- Less than or equal to: $a \leq b$
- Greater than: $a > b$
- Greater than or equal to: $a \geq b$
- Equal to: $a == b$
- Not Equal to: $a != b$

તમે આ શરતોનો ઉપયોગ વિવિધ નિર્ણયો માટે વિવિધ ક્રિયાઓ કરવા માટે કરી શકો છો.

જાવા પાસે નીચેના શરતી નિવેદનો છે :

- જો કોઈ ચોક્કસ શરત સાચી હોય તો, એક્ઝિક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે જો ઉપયોગ કરો
- જો તે જ શરત ખોટી હોય તો એક્ઝિક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે અન્યનો ઉપયોગ કરો
- જો પ્રથમ શરત ખોટી હોય તો ચકાસવા માટે નવી શરતનો ઉલ્લેખ કરવા માટે અન્યનો ઉપયોગ કરો
- કોડના ઘણા વૈકલ્પિક બ્લોકને અમલમાં મૂકવા માટે સ્વીચનો ઉપયોગ કરો

જો નિવેદન

જો કોઈ શરત સાચી હોય તો એક્ઝિક્યુટ કરવા માટે જાવા કોડના બ્લોકનો ઉલ્લેખ કરવા માટે if સ્ટેટમેન્ટનો ઉપયોગ કરો.

વાક્યરચના

```
if(condition){
    // block of code to be executed if the condition is true
}
```

નોંધ લો કે જો લોઅરકેસ અક્ષરોમાં હોય. અપરકેસ અક્ષરો (If અથવા IF) ભૂલ જનરેટ કરશે.

નીચેના ઉદાહરણમાં, 20 18 કરતા વધારે છે કે કેમ તે જાણવા માટે અમે બે મૂલ્યોનું પરીક્ષણ કરીએ છીએ. જો શરત સાચી હોય, તો અમુક ટેકસ્ટ છાપો:

ઉદાહરણ

```
if(20>18){
```

```
System.out.println("20 is greater than 18");
}
```

તેને જાતે અજમાવી જુઓ

અમે ચલો પણ ચકાસી શકીએ છીએ:

ઉદાહરણ

```
int x =20;
int y =18;
if(x > y){
    System.out.println("x is greater than y");
}
```

તેને જાતે અજમાવી જુઓ

ઉદાહરણ સમજાવ્યું

ઉપરના ઉદાહરણમાં x એ y કરતા મોટો છે કે કેમ તે ચકાસવા માટે આપણે બે ચલોનો ઉપયોગ કરીએ છીએ, x અને y (> ઓપરેટરનો ઉપયોગ કરીને). જેમ x 20 છે, અને y 18 છે, અને આપણે જાણીએ છીએ કે 20 18 કરતા મોટો છે, અમે સ્ક્રીન પર છાપીએ છીએ કે "x y કરતાં મોટો છે".

બીજું નિવેદન

જો શરત ખોટી હોય તો એક્ઝિક્યુટ કરવાના કોડના બ્લોકનો ઉલ્લેખ કરવા માટે else સ્ટેટમેન્ટનો ઉપયોગ કરો.

વાક્યરચના

```
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
}
```

ઉદાહરણ

```
int time =20;
if(time <18){
    System.out.println("Good day.");
}else{
    System.out.println("Good evening.");
}
```

// Outputs “Good evening.”

તેને જાતે અજમાવી જુઓ

ઉદાહરણ સમજાવ્યું

ઉપરના ઉદાહરણમાં, સમય (20) 18 કરતા વધારે છે, તેથી શરત ખોટી છે. આના કારણે, અમે બીજી સ્થિતિ પર આગળ વધીએ છીએ અને સ્ક્રીન પર “શુભ સાંજ” છાપીએ છીએ. જો સમય 18 થી ઓછો હોય, તો પ્રોગ્રામ “અચ્છા દિવસ” છાપશે.

બીજું જો નિવેદન

જો પ્રથમ શરત ખોટી હોય તો નવી શરતનો ઉલ્લેખ કરવા માટે else if સ્ટેટમેન્ટનો ઉપયોગ કરો.

વાક્યરચના

```
if(condition1){
// block of code to be executed if condition1 is true
}elseif(condition2){
// block of code to be executed if the condition1 is
false and condition2 is true
}else{
// block of code to be executed if the condition1 is
false and condition2 is false
}
```

ઉદાહરણ

```
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.”
```

તેને જાતે અજમાવી જુઓ

ઉદાહરણ સમજાવ્યું

ઉપરના ઉદાહરણમાં, સમય (22) 10 કરતા વધારે છે, તેથી પ્રથમ શરત ખોટી છે. આગળની શરત, else if સ્ટેટમેન્ટમાં, પણ ખોટી છે, તેથી અમે else શરત પર આગળ વધીએ છીએ કારણ કે શરત1 અને કંડીશન2 બંને ખોટા છે - અને સ્ક્રીન પર પ્રિન્ટ કરો “શુભ સાંજ”.

જો કે, જો સમય 14 હતો, તો અમારો પ્રોગ્રામ “શુભ દિવસ” છાપશે.

જાવા સ્વિચ

જાવા સ્વિચ સ્ટેટમેન્ટ

ઘણા if..else સ્ટેટમેન્ટ લખવાને બદલે, તમે સ્વિચ સ્ટેટમેન્ટનો ઉપયોગ કરી શકો છો.

સ્વીચ સ્ટેટમેન્ટ એક્ઝેક્યુટ કરવા માટે ઘણા કોડ બ્લોકમાંથી એક પસંદ કરે છે:

વાક્યરચના

```
switch(expression){
case x:
// code block
break;
case y:
// code block
break;
default:
// code block
}
```

આ રીતે તે કાર્ય કરે છે:

- સ્વીચ અભિવ્યક્તિનું મૂલ્યાંકન એકવાર કરવામાં આવે છે.
- અભિવ્યક્તિના મૂલ્યની સરખામણી દરેક કેસના મૂલ્યો સાથે કરવામાં આવે છે.
- જો ત્યાં મેચ હોય, તો કોડનો સંકળાયેલ બ્લોક એક્ઝેક્યુટ થાય છે.
- બ્રેક અને ડિફોલ્ટ કીવર્ડ્સ વૈકલ્પિક છે, અને આ પ્રકરણમાં પછીથી વર્ણવવામાં આવશે,

નીચેનું ઉદાહરણ અઠવાડિયાના દિવસના નામની ગણતરી કરવા માટે અઠવાડિયાના દિવસની સંખ્યાનો ઉપયોગ કરે છે:

ઉદાહરણ

```
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)
```

જાતે અજમાવી જુઓ

બ્રેક કીવર્ડ

જ્યારે જાવા બ્રેક કીવર્ડ સુધી પહોંચે છે, ત્યારે તે સ્વીચ બ્લોકમાંથી બહાર નીકળી જાય છે.

આ બ્લોકની અંદર વધુ કોડ અને કેસ ટેસ્ટિંગના અમલને અટકાવશે.

જ્યારે મેચ મળી જાય, અને કામ પૂર્ણ થઈ જાય, ત્યારે વિરામ લેવાનો સમય છે. વધુ પરીક્ષણની જરૂર નથી.

વિરામ અમલીકરણનો ઘણો સમય બચાવી શકે છે કારણ કે તે સ્વીચ બ્લોકમાં બાકીના તમામ કોડના અમલને "અવગણના" કરે છે.

મૂળભૂત કીવર્ડ

જો કોઈ કેસ મેચ ન હોય તો ડિફોલ્ટ કીવર્ડ ચલાવવા માટે અમુક કોડનો ઉલ્લેખ કરે છે:

ઉદાહરણ

```
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"
```

જાતે અજમાવી જુઓ

નોંધ કરો કે જો ડિફોલ્ટ સ્ટેટમેન્ટનો ઉપયોગ સ્વિચ બ્લોકમાં છેલ્લા સ્ટેટમેન્ટ તરીકે થાય છે, તો તેને વિરામની જરૂર નથી.

લૂપ્સ

જ્યાં સુધી ચોક્કસ શરત પહોંચી જાય ત્યાં સુધી લૂપ્સ કોડના બ્લોકને એક્ઝિક્યુટ કરી શકે છે.

લૂપ્સ સરળ છે કારણ કે તેઓ સમય બચાવે છે, ભૂલો ઘટાડે છે અને કોડને વધુ વાંચવા યોગ્ય બનાવે છે.

જાવા જ્યારે લૂપ

જ્યારે ઉલ્લેખિત શરત સાચી હોય ત્યાં સુધી કોડના બ્લોકમાંથી જ્યારે લૂપ લૂપ થાય છે:

વાક્યરચના

```
while(condition){  
    // code block to be executed  
}
```

નીચેના ઉદાહરણમાં, લૂપમાંનો કોડ, વેરીએબલ (i) 5 કરતા ઓછો હોય ત્યાં સુધી, વારંવાર ચાલશે:

ઉદાહરણ

```
int i =0;  
while(i <5){  
    System.out.println(i);  
    i++;  
}
```

જાતે અજમાવી જુઓ

નોંધ : શરતમાં ઉપયોગમાં લેવાતા ચલને વધારવાનું ભૂલશો નહીં, અન્યથા લૂપ ક્યારેય સમાપ્ત થશે નહીં!

ડુ/વ્હાઈલ લૂપ

do/while લૂપ એ while લૂપનો એક પ્રકાર છે. આ લૂપ કોડ બ્લોક એકવાર એક્ઝિક્યુટ કરશે, શરત સાચી છે કે કેમ તે તપાસતા પહેલા, પછી જ્યાં સુધી શરત સાચી છે ત્યાં સુધી તે લૂપનું પુનરાવર્તન કરશે.

વાક્યરચના

```
do{  
    // code block to be executed  
}  
while(condition);
```

નીચેનું ઉદાહરણ do/while લૂપનો ઉપયોગ કરે છે. લૂપ હંમેશા ઓછામાં ઓછા એક વખત એક્ઝિક્યુટ કરવામાં આવશે, ભલે શરત ખોટી હોય, કારણ કે શરત ચકાસવામાં આવે તે પહેલાં કોડ બ્લોક એક્ઝિક્યુટ કરવામાં આવે છે:

ઉદાહરણ

```
int i =0;  
do{  
    System.out.println(i);
```

```
i++;
```

```
}
```

```
while(i < 5);
```

જાતે અજમાવી જુઓ”

શરતમાં ઉપયોગમાં લેવાતા ચલને વધારવાનું ભૂલશો નહીં, અન્યથા લૂપ ક્યારેય સમાપ્ત થશે નહીં!

લૂપ માટે જાવા

જ્યારે તમે જાણો છો કે તમે કોડના બ્લોકમાંથી કેટલી વાર લૂપ કરવા માંગો છો, ત્યારે લૂપને બદલે for લૂપનો ઉપયોગ કરો:

વાક્યરચના

```
for(statement 1; statement 2; statement 3){
```

```
// code block to be executed
```

```
}
```

વિધાન 1 કોડ બ્લોકના અમલ પહેલા (એક વખત) ચલાવવામાં આવે છે.

વિધાન 2 કોડ બ્લોક ચલાવવા માટેની સ્થિતિને વ્યાખ્યાયિત કરે છે.

કોડ બ્લોક એક્ઝિક્યુટ થયા પછી સ્ટેટમેન્ટ 3 એક્ઝિક્યુટ કરવામાં આવે છે (દર વખતે).

નીચેનું ઉદાહરણ 0 થી 4 નંબરો છાપશે:

ઉદાહરણ

```
for(int i = 0; i < 5; i++){
```

```
System.out.println(i);
```

```
}
```

જાતે અજમાવી જુઓ”

ઉદાહરણ સમજાવ્યું

વિધાન 1 લૂપ શરૂ થાય તે પહેલા ચલ સેટ કરે છે (int i = 0).

વિધાન 2 લૂપ ચલાવવા માટેની સ્થિતિને વ્યાખ્યાયિત કરે છે (કુલ 5 કરતા ઓછો હોવો જોઈએ). જો શરત સાચી છે, તો લૂપ ફરીથી શરૂ થશે, જો તે ખોટું છે, તો લૂપ સમાપ્ત થશે.

વિધાન 3 દરેક વખતે જ્યારે લૂપમાં કોડ બ્લોક એક્ઝિક્યુટ કરવામાં આવે ત્યારે મૂલ્ય (i++) વધે છે.

બીજું ઉદાહરણ

આ ઉદાહરણ માત્ર 0 અને 10 ની વચ્ચેના સમાન મૂલ્યોને છાપશે:

ઉદાહરણ

```
for(int i = 0; i <= 10; i = i + 2){
```

```
System.out.println(i);
```

```
}
```

તેને જાતે અજમાવી જુઓ

નેસ્ટેડ લૂપ્સ

બીજા લૂપની અંદર લૂપ મૂકવું પણ શક્ય છે. તેને નેસ્ટેડ લૂપ કહેવામાં આવે છે.

“આંતરિક લૂપ” ને “બાહ્ય લૂપ” ના દરેક પુનરાવર્તન માટે એક વખત ચલાવવામાં આવશે:

ઉદાહરણ

```
// 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)
```

```
}
```

```
}
```

જાવા બ્રેક

તમે આ ટ્યુટોરીયલના અગાઉના પ્રકરણમાં વપરાયેલ વિરામ નિવેદન જોઈ લીધું છે. તેનો ઉપયોગ સ્વિચ સ્ટેટમેન્ટમાંથી “જમ્પ આઉટ” કરવા માટે થતો હતો.

બ્રેક સ્ટેટમેન્ટનો ઉપયોગ લૂપમાંથી બહાર જવા માટે પણ થઈ શકે છે.

આ ઉદાહરણ લૂપને રોકે છે જ્યારે કુલ 4 ની બરાબર હોય છે:

ઉદાહરણ

```
for(int i = 0; i < 10; i++){
```

```
if(i == 4){
```

```
break;
```

```
}
```

```
System.out.println(i);
```

```
}
```

તેને જાતે અજમાવી જુઓ

જાવા ચાલુ રાખો

ચાલુ નિવેદન એક પુનરાવૃત્તિ (લૂપમાં) તોડે છે, જો કોઈ સ્પષ્ટ સ્થિતિ થાય છે, અને લૂપમાં આગામી પુનરાવર્તન સાથે ચાલુ રહે છે.

આ ઉદાહરણ 4 નું મૂલ્ય છોડે છે:

ઉદાહરણ

```
for(int i = 0; i < 10; i++){
```

```
if(i == 4){
```

```
continue;
```

```
}
```

```
System.out.println(i);
```

```
}
```

તેને જાતે અજમાવી જુઓ

જ્યારે લૂપમાં બ્રેક કરો અને ચાલુ રાખો

તમે વિરામનો ઉપયોગ પણ કરી શકો છો અને જ્યારે લૂપ્સ ચાલુ રાખો છો:

બ્રેક ઉદાહરણ

```
int i =0;
while(i <10){
System.out.println(i);
i++;
if(i ==4){
break;
}
}
```

તેને જાતે અજમાવી જુઓ

ઉદાહરણ ચાલુ રાખો

```
int i =0;
while(i <10){
if(i ==4){
i++;
continue;
}
System.out.println(i);
i++;
}
```

જાવા એરે

એરેનો ઉપયોગ દરેક મૂલ્ય માટે અલગ ચલો જાહેર કરવાને બદલે એક જ ચલમાં બહુવિધ મૂલ્યોને સંગ્રહિત કરવા માટે થાય છે.

એરે જાહેર કરવા માટે, ચલ પ્રકારને ચોરસ કૌંસ સાથે વ્યાખ્યાયિત કરો:

સ્ટ્રિંગ [] કાર;

અમે હવે વેરીએબલ જાહેર કર્યું છે જે સ્ટ્રિંગ્સની શ્રેણી ધરાવે છે. તેમાં મૂલ્યો દાખલ કરવા માટે, તમે સર્પાકાર કૌંસની અંદર, અલ્પવિરામથી વિભાજિત સૂચિમાં મૂલ્યો મૂકી શકો છો:

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};
```

પૂર્ણાંકોની શ્રેણી બનાવવા માટે, તમે લખી શકો છો:

```
int[] myNum ={10,20,30,40};
```

એરેના તત્વોને ઍક્સેસ કરો

તમે ઇન્ડેક્સ નંબરનો ઉલ્લેખ કરીને એરે એલિમેન્ટને ઍક્સેસ કરી શકો છો.

આ નિવેદન કારમાંના પ્રથમ તત્વના મૂલ્યને ઍક્સેસ કરે છે:

ઉદાહરણ

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};
System.out.println(cars[0]);
// Outputs Volvo
```

તેને જાતે અજમાવી જુઓ

નોંધ : એરે ઇન્ડેક્સ 0 થી શરૂ થાય છે: [0] એ પ્રથમ તત્વ છે. [1] બીજું તત્વ છે, વગેરે.

એરે એલિમેન્ટ બદલો

ચોક્કસ તત્વનું મૂલ્ય બદલવા માટે, અનુક્રમણિકા નંબરનો સંદર્ભ લો:

ઉદાહરણ

```
cars[0]="Opel";
```

ઉદાહરણ

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};
cars[0]="Opel";
System.out.println(cars[0]);
// Now outputs Opel instead of Volvo
```

તેને જાતે અજમાવી જુઓ

એરે લંબાઈ

એરેમાં કેટલા તત્વો છે તે શોધવા માટે, લંબાઈ ગુણધર્મનો ઉપયોગ કરો:

ઉદાહરણ

```
String[] cars ={"Volvo","BMW","Ford","Mazda"};
System.out.println(cars.length);
// Outputs 4
```

જાવા વર્ગો, ઓવરલોડિંગ અને વારસો (JAVA Classes, Overloading and Inheritance)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- java OOPs કન્સેપ્ટ
- ઓબ્જેક્ટ ક્લોનિંગ
- પદ્ધતિના પ્રકાર.

જાવા ઓબ્જેક્ટ વર્ગ

જાવા OOPs કન્સેપ્ટ

આપણે OOPs ની મૂળભૂત બાબતો વિશે શીખીશું. ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ એ એક દાખલો (સિદ્ધાંત) છે જે ઘણા ખ્યાલો (વિચારો) પ્રદાન કરે છે, જેમ કે વારસામાં, ડેટા બાઈન્ડિંગ, પોલીમોર્ફિઝમ, વગેરે.

સિમુલેને પ્રથમ ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ ભાષા ગણવામાં આવે છે. પ્રોગ્રામિંગ પેરાડાઈમ જ્યાં દરેક વસ્તુને એટ્રિબ્યુટોરિયન્ટ કીપ્રોગ્રામિંગ લેંગ્વેજ તરીકે આનબેક્ટિસ્ક્રોવ તરીકે રજૂ કરવામાં આવે છે.

Smalltalk ને સાચી રીતે પ્રથમ ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ ભાષા ગણવામાં આવે છે. લોકપ્રિય ઓબ્જેક્ટ ઓરિએન્ટેડ ભાષાઓ જાવા, C#, PHP, Python, C++, વગેરે છે.

ઓબ્જેક્ટ-ઓરિએન્ટેડ પ્રોગ્રામિંગનો મુખ્ય ઉદ્દેશ વાસ્તવિક વિશ્વની સંસ્થાઓને અમલમાં મૂકવાનો છે, ઉદાહરણ તરીકે, ઓબ્જેક્ટ, વર્ગો, અમૂર્તતા, વારસો, પોલીમોર્ફિઝમ, એબ્સ્ટ્રેક્શન, એન્કેપ્સ્યુલેશન વગેરે.

OOPs (ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ સિસ્ટમ)

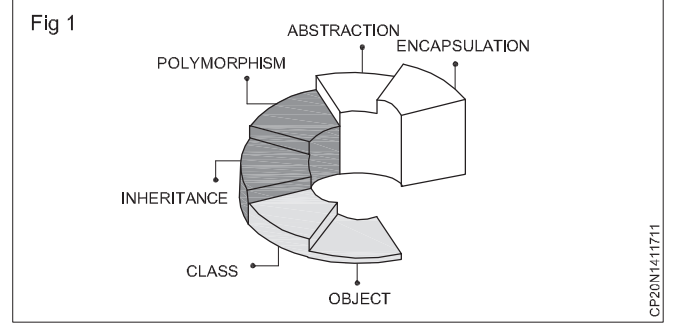
ઓબ્જેક્ટનો અર્થ થાય છે વાસ્તવિક-દુનિયાની એન્ટિટી જેમ કે પેન, ખુરશી, ટેબલ, કમ્પ્યુટર, ઘડિયાળ વગેરે. ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ એ વર્ગો અને ઓબ્જેક્ટ્સનો ઉપયોગ કરીને પ્રોગ્રામ ડિઝાઇન કરવા માટેની પદ્ધતિ અથવા દાખલો છે. તે કેટલાક ખ્યાલો પ્રદાન કરીને સોફ્ટવેર વિકાસ અને જાળવણીને સરળ બનાવે છે:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

આ વિભાવનાઓ સિવાય, કેટલાક અન્ય શબ્દો છે જે ડિઝાઇન ઓબ્જેક્ટ-ઓરિએન્ટેડ ડિઝાઇનનો ઉપયોગ કરે છે:

- Coupling
- Cohesion
- Association
- Aggregation
- Composition

જાવા OOPs ખ્યાલો



ઓબ્જેક્ટ

રાજ્ય અને વર્તન ધરાવતી કોઈપણ એન્ટિટીને ઓબ્જેક્ટ તરીકે ઓળખવામાં આવે છે. ઉદાહરણ તરીકે, ખુરશી, પેન, ટેબલ, કીબોર્ડ, બાઈક, વગેરે. તે ભૌતિક અથવા તાર્કિક હોઈ શકે છે.

ઓબ્જેક્ટને વર્ગના ઉદાહરણ તરીકે વ્યાખ્યાયિત કરી શકાય છે. ઓબ્જેક્ટ એક સરનામું ધરાવે છે અને કેટલીક જગ્યા મેમરી લે છે. ઓબ્જેક્ટ એકબીજાના ડેટા અથવા કોડની વિગતો જાણ્યા વિના વાતચીત કરી શકે છે. એકમાત્ર જરૂરી વસ્તુ એ છે કે સ્વીકારવામાં આવેલ સંદેશનો પ્રકાર અને ઓબ્જેક્ટ દ્વારા પ્રતિસાદનો પ્રકાર.

ઉદાહરણ : કૂતરો એક પદાર્થ છે કારણ કે તેમાં રંગ, નામ, જાતિ વગેરે જેવી સ્થિતિઓ તેમજ પૂંછડી હલાવવા, ભસવું, ખાવું વગેરે જેવી વર્તણૂકો છે.

વર્ગ

કલેક્શન ઓફ ઓબ્જેક્ટ સિસ્ટેમ્સ કલાસ. ઇટીસાલોજિકલેન્ટિટી.

વર્ગને બ્લુપ્રિન્ટ તરીકે પણ વ્યાખ્યાયિત કરી શકાય છે જેમાંથી તમે વ્યક્તિગત ઓબ્જેક્ટ બનાવી શકો છો. વર્ગ કોઈપણ જગ્યાનો ઉપયોગ કરતું નથી.

વારસો

જ્યારે એક ઓબ્જેક્ટ પિતૃ ઓબ્જેક્ટના તમામ ગુણધર્મો અને વર્તણૂકો મેળવે છે, ત્યારે તેને વારસા તરીકે ઓળખવામાં આવે છે. તે કોડ પુનઃઉપયોગિતા પ્રદાન કરે છે. Itis નો ઉપયોગ ઇમેપોલિમોર્ફિઝમને પ્રાપ્ત કરવા માટે થાય છે.

પોલીમોર્ફિઝમ

જો એક કાર્ય જુદી જુદી રીતે કરવામાં આવે છે, તો તે પોલીમોર્ફિઝમ તરીકે ઓળખાય છે. ઉદાહરણ તરીકે: ગ્રાહકને અલગ રીતે સમજાવવા, કંઈક દોરવા, ઉદાહરણ તરીકે, આકાર, ત્રિકોણ, લંબચોરસ, વગેરે.

જાવામાં, અમે પોલીમોર્ફિઝમ પ્રાપ્ત કરવા માટે મેથોડોવરલોડીંગ અને મેથોડોવરરીનો ઉપયોગ કરીએ છીએ.

બીજું ઉદાહરણ કંઈક બોલવાનું હોઈ શકે; ઉદાહરણ તરીકે, એક બિલાડી મ્યાઉ બોલે છે, કૂતરો ભસશે, વગેરે.

એબ્સ્ટ્રેક્શન

આંતરિક વિગતો છુપાવવી અને કાર્યક્ષમતા દર્શાવવી એ એબ્સ્ટ્રેક્શન તરીકે ઓળખાય છે. ઉદાહરણ તરીકે, ફોન કોલ, અમને આંતરિક પ્રક્રિયાની ખબર નથી.

જાવામાં, અમે અમૂર્તતા પ્રાપ્ત કરવા માટે અમૂર્ત વર્ગ અને ઇન્ટરફેસનો ઉપયોગ કરીએ છીએ.

એન્કેપ્સ્યુલેશન

બાઈન્ડિંગ (અથવા રેપિંગ) કોડ અને ડેટાને એકસાથે એક એકમમાં એન્કેપ્સ્યુલેશન તરીકે ઓળખવામાં આવે છે. ઉદાહરણ તરીકે, acapsule, itis વિવિધ દવાઓ સાથે આવરિત.

જાવા વર્ગ એ એન્કેપ્સ્યુલેશનનું ઉદાહરણ છે. જાવા બીન એ સંપૂર્ણ રીતે સમાવિષ્ટ વર્ગ છે કારણ કે તમામ ડેટા સભ્યો અહીં ખાનગી છે.

કપલિંગ

જોડાણ એ અન્ય વર્ગના જ્ઞાન અથવા માહિતી અથવા અવલંબનનો સંદર્ભ આપે છે. જ્યારે વર્ગો એકબીજાથી પરિચિત હોય ત્યારે તે ઉદ્ભવે છે. જો કોઈ વર્ગમાં બીજા વર્ગની વિગતોની માહિતી હોય, તો મજબૂત જોડાણ છે. Java માં, અમે વર્ગ, પદ્ધતિ અને ક્ષેત્રના દૃશ્યતા સ્તરને પ્રદર્શિત કરવા માટે ખાનગી, સુરક્ષિત અને જાહેર સંશોધકોનો ઉપયોગ કરીએ છીએ. તમે નબળા જોડાણ માટે ઇન્ટરફેસનો ઉપયોગ કરી શકો છો કારણ કે ત્યાં કોઈ નક્કર અમલીકરણ નથી.

સંયોગ

સુસંગતતા એ એક ઘટકના સ્તરનો ઉલ્લેખ કરે છે જે એક સારી રીતે વ્યાખ્યાયિત કાર્ય કરે છે. એક જ સારી રીતે વ્યાખ્યાયિત કાર્ય અત્યંત સુસંગત પદ્ધતિ દ્વારા કરવામાં આવે છે. નબળી સંયોજક પદ્ધતિ કાર્યને અલગ-અલગ ભાગોમાં વિભાજિત કરશે. java.io પેકેજ એ અત્યંત સુસંગત પેકેજ છે કારણ કે તેમાં I/O સંબંધિત વર્ગો અને ઇન્ટરફેસ છે. જો કે, the java.util package eisa નબળી રીતે કોહેસિવ પેકેજ કારણ કે તે અસંબંધિત વર્ગો અને ઇન્ટરફેસ છે.

એસોસિએશન

એસોસિએશન વસ્તુઓ વચ્ચેના સંબંધનું પ્રતિનિધિત્વ કરે છે. અહીં, એક ઓબ્જેક્ટ એક ઓબ્જેક્ટ અથવા ઘણા ઓબ્જેક્ટ સાથે સંકળાયેલ હોઈ શકે છે. વસ્તુઓ વચ્ચે ચાર પ્રકારના જોડાણ હોઈ શકે છે:

- OnetoOne
- OnetoMany
- ManytoOne, and
- ManytoMany

ચાલો વાસ્તવિક સમયના ઉદાહરણો સાથેના સંબંધને સમજીએ. ઉદાહરણ તરીકે, એક દેશમાં એક વડા પ્રધાન (એકથી એક) હોઈ શકે છે, અને વડા પ્રધાન પાસે ઘણા પ્રધાનો (એકથી ઘણા) હોઈ શકે છે. ઉપરાંત, ઘણા સાંસદો પાસે એક વડા પ્રધાન (ઘણાથી એક) હોઈ શકે છે, અને ઘણા પ્રધાનો પાસે ઘણા વિભાગો હોઈ શકે છે (ઘણાથી ઘણા).

એસોસિએશન દિશાહીન અથવા દ્વિપક્ષીય હોઈ શકે છે.

એકત્રીકરણ

એકત્રીકરણ એ એસોસિએશન હાંસલ કરવાનો એક માર્ગ છે. એકત્રીકરણ એ સંબંધનું પ્રતિનિધિત્વ કરે છે જ્યાં એક ઓબ્જેક્ટ તેના રાજ્યના ભાગ તરીકે અન્ય ઓબ્જેક્ટ ધરાવે છે. તે પદાર્થો વચ્ચેના નબળા સંબંધને દર્શાવે

છે. તેને અહસારેલેશનશિપ nJavaLike તરીકે પણ ઓળખવામાં આવે છે, વારસો એ સંબંધનું પ્રતિનિધિત્વ કરે છે. તે અન્ય રીતે ઓબ્જેક્ટનો ઉપયોગ કરે છે.

રચના

રચના એ એસોસિએશન હાંસલ કરવાનો પણ એક માર્ગ છે. રચના એ સંબંધનું પ્રતિનિધિત્વ કરે છે જ્યાં એક પદાર્થ તેની સ્થિતિના ભાગ રૂપે અન્ય વસ્તુઓ ધરાવે છે. સમાવિષ્ટ પદાર્થ અને આશ્રિત પદાર્થ વચ્ચે મજબૂત સંબંધ છે. તે એવી સ્થિતિ છે કે જ્યાં પદાર્થો સમાવિષ્ટ હોય તેનું સ્વતંત્ર અસ્તિત્વ હોતું નથી. જો તમે ઓબ્જેક્ટમાંથી કોઈને કાઢી નાખો છો, તો તમામ ચાઈલ્ડ ઓબ્જેક્ટ્સ આપમેળે કાઢી નાખવામાં આવશે.

OOP પ્રક્રિયા લક્ષી પ્રોગ્રામિંગ ભાષાનો ફાયદો

- 1 OOPs વિકાસ અને જાળવણીને સરળ બનાવે છે, જ્યારે, પ્રક્રિયા લક્ષી પ્રોગ્રામિંગ ભાષામાં, જો પ્રોજેક્ટનું કદ વધે તેમ કોડ વધે તો તેનું સંચાલન કરવું સરળ નથી.
- 2 OOPs ડેટા છુપાવવાનું પ્રદાન કરે છે, જ્યારે, પ્રક્રિયાલક્ષી પ્રોગ્રામિંગ ભાષામાં, વૈશ્વિક ડેટા ગમે ત્યાંથી એક્સેસ કરી શકાય છે.
- 3 OOPs વાસ્તવિક વિશ્વની ઘટનાને વધુ અસરકારક રીતે અનુકરણ કરવાની ક્ષમતા પ્રદાન કરે છે. જો આપણે ઓબ્જેક્ટ-ઓરિએન્ટેડ પ્રોગ્રામિંગ લેંગ્વેજનો ઉપયોગ કરતા હોઈએ તો અમે વાસ્તવિક દુનિયાની સમસ્યાનો ઉકેલ આપી શકીએ છીએ.

ગ્લોબલડેટા

આકૃતિ : પ્રોસિજર ઓરિએન્ટેડ પ્રોગ્રામિંગમાં ડેટા રિપ્રેઝન્ટેશન (ફિગ 2)

Fig 2



ઓબ્જેક્ટડેટા

ઓબ્જેક્ટ-ઓરિએન્ટેડ પ્રોગ્રામિંગ ભાષા અને ઓબ્જેક્ટ-આધારિત પ્રોગ્રામિંગ ભાષા વચ્ચે શું તફાવત છે?

ઓબ્જેક્ટ-આધારિત પ્રોગ્રામિંગ લેંગ્વેજ ઇન્ટરિપ્રેટર્સ સિવાય OOPs ની તમામ સુવિધાઓને અનુસરે છે. JavaScript અને VB સ્ક્રિપ્ટ ઓબ્જેક્ટ-આધારિત પ્રોગ્રામિંગ ભાષાઓના ઉદાહરણો છે.

શું તમે જાણો છો?

- મુખ્ય પદ્ધતિને ઓવરલોડ કરી શકો છો?

- A Java Constructor એ મૂલ્યનું વળતર આપે છે પરંતુ, શું?
- મુખ્ય પદ્ધતિથી પ્રોગ્રામ બનાવી શકીએ?
- આ કીવર્ડનો ઉપયોગ શું છે?
- શા માટે બહુવિધ વારસાગત નોત્સુ પોટેન્ડિનજાવા?
- Aggregati n નો ઉપયોગ શા માટે કરવો?
- અમે સ્ટેટિક પદ્ધતિને ઓવરરાઈડ કરી શકીએ?
- સહવર્તી વળતરનો પ્રકાર શું છે?
- Jav સુપર કીવર્ડના ત્રણ ઉપયોગો શું છે?
- ઈન્સ્ટન્સ ઈનિશિયલાઈઝર બ્લોકનો ઉપયોગ શા માટે?
- ખાલી અંતિમ ચલનો શું ઉપયોગ છે?
- અમાર્કર અથવા ટેગ કરેલ ઈન્ટરફેસ શું છે?
- રન ટાઈમ પોલીમોર્ફિસ્મર ડાયનેમિક મેથડ પેચ શું છે?
- સ્ટેટિક અને ડાયનેમિક બાઈન્ડિંગ વચ્ચે શું તફાવત છે?
- જાવામાં ડાઉનકાસ્ટિંગ કેવી રીતે શક્ય છે?
- ખાનગી કન્સ્ટ્રક્ટરનો હેતુ શું છે?
- ઓબ્જેક્ટકલોનિંગ શું છે?

OOPsના ખ્યાલો શું હશે?

- OOPs નો ફાયદો
- નામકરણ સંમેલન
- ઓબ્જેક્ટ અને ક્લાસ
- પદ્ધતિ ઓવરલોડિંગ
- કન્સ્ટ્રક્ટર
- સ્ટેટિક કીવર્ડ
- આ કીવર્ડ છ ઉપયોગ સાથે
- વારસો
- એકત્રીકરણ
- પદ્ધતિ ઓવરરાઈડિંગ
- CovariantReturn Type
- સુપરકીવર્ડ
- InstanceInitializerblock
- અંતિમ કીવર્ડ
- એબ્સ્ટ્રેક્ટક્લાસ
- ઈન્ટરફેસ
- રનટાઈમ પોલીમોર્ફિઝમ
- સ્ટેટિક અને ડાયનેમિક બાઈન્ડિંગ
- ઓપરેટરની સાથે ડાઉનકાસ્ટિંગ
- પેકેજ

- એક્સેસ મોડિફાયર
- એન્કેપ્સ્યુલેશન

ઓબ્જેક્ટકલોનિંગ

મેથોડિનજાવા

સામાન્ય રીતે, પદ્ધતિ એ અમુક કાર્ય કરવા માટેની એક રીત છે. તેવી જ રીતે, Java માં પદ્ધતિ એ સૂચનાઓનો સંગ્રહ છે જે ચોક્કસ કાર્ય કરે છે. તે કોડની પુનઃઉપયોગીતા પૂરી પાડે છે. અમે પદ્ધતિઓનો ઉપયોગ કરીને કોડને સરળતાથી સુધારી શકીએ છીએ. આ વિભાગમાં, આપણે શીખીશું કે જાવામાં પદ્ધતિ શું છે, પદ્ધતિઓના પ્રકારો, પદ્ધતિની ઘોષણા અને એમેથોડિન જાવાને કેવી રીતે કોલ કરવો.

એમેથોડિન જાવા શું છે?

પદ્ધતિ એ કોડનો બ્લોક અથવા નિવેદનોનો સંગ્રહ અથવા ચોક્કસ કાર્ય અથવા કામગીરી કરવા માટે એકસાથે જૂથ થયેલ કોડનો સમૂહ છે. તેનો ઉપયોગ કોડની પુનઃઉપયોગીતા હાંસલ કરવા માટે થાય છે. અમે એક મેથડ એકવાર લખીએ છીએ અને ઘણી વખત તેનો ઉપયોગ કરીએ છીએ. અમને વારંવાર કોડ લખવાની જરૂર નથી. તે કોડમાં સરળ ફેરફાર અને વાંચનક્ષમતા પણ પ્રદાન કરે છે, જસ્ટબાયડિન અથવા કોડનો એક ભાગ દૂર કરવો. જ્યારે આપણે વોકીટમાં કોલ કરીએ છીએ ત્યારે જ પદ્ધતિનો અમલ થાય છે.

જાવામાં સૌથી મહત્વની પદ્ધતિ મુખ્ય() પદ્ધતિ છે. જો તમે મુખ્ય() પદ્ધતિ વિશે વધુ વાંચવા માંગતા હો, તો <https://www.javatoint.com/java/mainmethod> લિંક પર જાઓ.

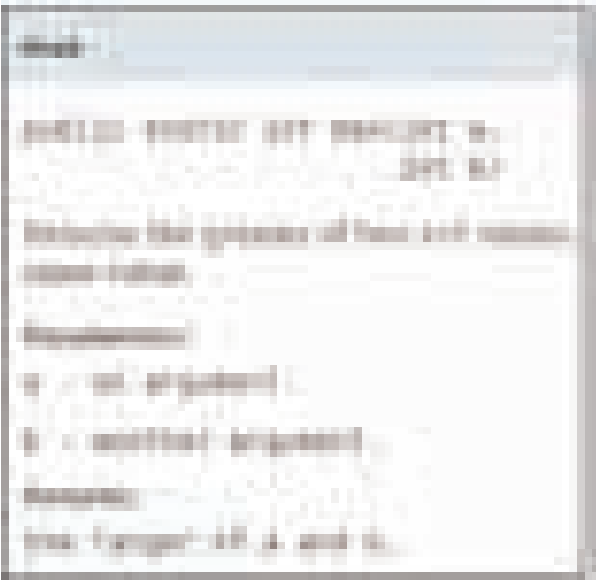
પદ્ધતિની ઘોષણા (ફિગ 3)

પદ્ધતિની ઘોષણા પદ્ધતિ વિશેષતાઓ વિશે માહિતી પ્રદાન કરે છે, જેમ કે દૃશ્યતા, વળતર-પ્રકાર, નામ અને દલીલો. તેમાં છ ઘટકો છે જે મેથડ હેડર તરીકે ઓળખાય છે, જેમ કે આપણે નીચેની આકૃતિમાં બતાવ્યું છે.

પદ્ધતિ હસ્તાક્ષર : દરેક પદ્ધતિમાં એક પદ્ધતિ હસ્તાક્ષર હોય છે. તે પદ્ધતિની ઘોષણાનો એક ભાગ છે. તે પણ સમાવેશ થાય પદ્ધતિ નામ અને પરિમાણ યાદી.

એક્સેસ સ્પેસિફાયર : એક્સેસ સ્પેસિફાયર અથવા મોડિફાયર એ પદ્ધતિનો એક્સેસ પ્રકાર છે. તે થીમ પદ્ધતિની દૃશ્યતા સ્પષ્ટ કરે છે. જાવા ચાર પ્રકારના એક્સેસ સ્પષ્ટકર્તા પ્રદાન કરે છે:

- **સાર્વજનિક :** જ્યારે અમે અમારી એપ્લિકેશનમાં સાર્વજનિક સ્પષ્ટીકરણોનો ઉપયોગ કરીએ છીએ ત્યારે પદ્ધતિ તમામ વર્ગો દ્વારા સુલભ છે.
- **ખાનગી :** જ્યારે આપણે પ્રાઈવેટ એક્સેસ સ્પેસિફાયરનો ઉપયોગ કરીએ છીએ, ત્યારે પદ્ધતિ ફક્ત તે વર્ગોમાં જ સુલભ છે જેમાં તે વ્યાખ્યાયિત કરવામાં આવી છે.
- **સંરક્ષિત :** જ્યારે આપણે સુરક્ષિત એક્સેસ સ્પેસિફાયરનો ઉપયોગ કરીએ છીએ, ત્યારે પદ્ધતિ એક જ પેકેજમાં અથવા અલગ પેકેજમાં પેટા વર્ગોમાં એક્સેસિબલ હોય છે.
- **ડિફોલ્ટ :** જ્યારે અમે પદ્ધતિની ઘોષણામાં કોઈપણ એક્સેસ સ્પષ્ટીકરણનો ઉપયોગ કરતા નથી, ત્યારે Java મૂળભૂત રીતે ડિફોલ્ટ એક્સેસ સ્પષ્ટીકરણનો ઉપયોગ કરે છે. તે ફક્ત સમાન પેકેજમાંથી જ દૃશ્યમાન છે.



ઉપયોગ કર્યો છે. આ પૂર્વવ્યાખ્યાયિત પદ્ધતિનો ફાયદો છે. તે પ્રોગ્રામિંગને ઓછું જટિલ બનાવે છે. એ જ રીતે, આપણે print() મેથડની મેથડ સિગ્નેચર પણ જોઈ શકીએ છીએ.

વપરાશકર્તા-નિર્ધારિત પદ્ધતિ

વપરાશકર્તા અથવા પ્રોગ્રામર દ્વારા લખાયેલ પદ્ધતિને વપરાશકર્તા વ્યાખ્યાયિત પદ્ધતિ તરીકે ઓળખવામાં આવે છે. આ પદ્ધતિઓમાં જરૂરિયાત મુજબ ફેરફાર કરવામાં આવે છે.

યુઝર ડિફાઈન્ડ મેથડ

કેવી રીતે બનાવવી ચાલો વપરાશકર્તા વ્યાખ્યાયિત પદ્ધતિ જે તપાસે છે નંબર બેકી કે બેકી છે તે બનાવીએ. પ્રથમ, આપણે પદ્ધતિને વ્યાખ્યાયિત કરીશું.

```
//userdefinedmethod
publicstaticvoidfindEvenOdd(intnum)
{
//method bodyif(num%2==0)
System.out.println(num+" is even");else
System.out.println(num+"isodd");
}
```

અમે findevenodd() નામની ઉપરની પદ્ધતિને વ્યાખ્યાયિત કરી છે. તે પ્રકાર int ની પરિમાણ સંખ્યા ધરાવે છે. પદ્ધતિ કોઈપણ મૂલ્ય પરત કરતી નથી તેથી જ અમે void નો ઉપયોગ કર્યો છે. મેથડ બોડીમાં સંખ્યા બેકી છે કે બેકી છે તે ચકાસવા માટેનાં પગલાંઓ શામેલ છે. જો સંખ્યા બેકી હોય, તો તે સંખ્યાને બેકી છાપે છે, અન્યથા તે સંખ્યાને બેકી છાપે છે.

Invoke a User defined Method કેવી રીતે કોલર કરવી

એકવાર આપણે પદ્ધતિ વ્યાખ્યાયિત કરી લીધા પછી, તેને બોલાવવી જોઈએ. પ્રોગ્રામમાં મેથડનું કોલિંગ સરળ છે. જ્યારે wecallorinvoke a user વ્યાખ્યાયિત પદ્ધતિ, કાર્યક્રમ કંટ્રોલ કહેવાય પદ્ધતિમાં ટ્રાન્સફર.

```
import
java.util.Scanner;public
classEvenOdd
```

```
{
publicstaticvoidmain(String args[])
{
//creatingScannerclassobjectScannersc
an=newScanner(System.in);
System.out.print("Enterthenumber:");
//reading value from the
userintnum=scan.nextInt();
//method
callingfindEvenOdd
(num);
}
```

ઉપરોક્ત કોડ સ્પિટમાં, કમ્પાઈલર લાઈન પર પહોંચે કે તરત જ EvenOdd (num) શોધો, કંટ્રોલ પદ્ધતિમાં ટ્રાન્સફર થાય છે અને તે મુજબ આઉટપુટ આપે છે.

ચાલો એક જ પ્રોગ્રામમાં કોડના પેટને જોડીએ અને તેને એક્ઝિક્યુટ કરીએ.

EvenOdd.java

```
import
java.util.Scanner;public
classEvenOdd
{
publicstaticvoidmain(String args[])
{
//creatingScannerclassobjectScannersc
an=newScanner(System.in);
System.out.print("Enterthenumber:");
//reading value from
userintnum=scan.nextInt();
//method
callingfindEvenOdd
(num);
}
//userdefinedmethod
publicstaticvoidfindEvenOdd(intnum)
{
//method
bodyif(num%2
```

```

==0)
System.out.println(num+" is
even");else
System.out.println(num+"isodd");
}
}

```

આઉટપુટ 1 :

Enter the number:

1212 is even

આઉટપુટ 2 :

Enter the number:99

99 is odd

ચાલો બીજો પ્રોગ્રામ જોઈએ જે કૉલિંગ પદ્ધતિમાં મૂલ્ય આપે છે.

નીચેના પ્રોગ્રામમાં, અમે add() નામની પદ્ધતિ વ્યાખ્યાયિત કરી છે જે બે સંખ્યાઓનો સરવાળો કરે છે. તેમાં પૂર્ણાંક પ્રકારના બે પરિમાણ n1 અને n2 છે. n1 અને n2 ની કિંમતો અનુક્રમે a અને b ના મૂલ્યને અનુરૂપ છે. તેથી, પદ્ધતિ a અને b ની કિંમત ઉમેરે છે અને તેને ચલોમાં સંગ્રહિત કરે છે અને સરવાળો આપે છે.

```

Addition.java
publicclassAddition
{
publicstaticvoidmain(String[]args)
{
int a =
19;intb
=5;
//methodcalling
int c = add(a,b);//a and b are actual
parametersSystem.out.println("Thesum of
aandbis="+c);
}
//userdefinedmethod
publicstaticintadd(intn1,intn2)//n1andn2are formal
parameters
{
int
s;s=n1+n
2;
returns;//returningthesum

```

```
}
```

```
}
```

આઉટપુટ :

The sum of anubis=24

સ્થિર પદ્ધતિ

સ્ટેટિક કીવર્ડ ધરાવતી પદ્ધતિને સ્ટેટિક મેથડ તરીકે ઓળખવામાં આવે છે. બીજા શબ્દોમાં કહીએ તો, એક પદ્ધતિ કે જે વર્ગના દાખલાને બદલે વર્ગની છે તેને સ્થિર પદ્ધતિ તરીકે ઓળખવામાં આવે છે. આપણે મેથડના નામ પહેલા સ્ટેટિક કીવર્ડનો ઉપયોગ કરીને સ્ટેટિક મેથડ પણ બનાવી શકીએ છીએ.

સ્થિર પદ્ધતિનો મુખ્ય ફાયદો એ છે કે આપણે તેને ઓબ્જેક્ટ બનાવ્યા વિના કૉલ કરી શકીએ છીએ. તે સ્ટેટિક ડેટા મેમ્બર્સને એક્સેસ કરી શકે છે અને તેની કિંમત પણ બદલી શકે છે. તેનો ઉપયોગ દાખલાની પદ્ધતિ બનાવવા માટે થાય છે. તેને વર્ગના નામનો ઉપયોગ કરીને બોલાવવામાં આવે છે. સ્થિર પદ્ધતિનું શ્રેષ્ઠ ઉદાહરણ themain () પદ્ધતિ છે.

સ્ટેટિકનું ઉદાહરણ

methodDisplay.java

```

public classDisplay
{
publicstaticvoidmain(String[]args)
{
show();
}
staticvoidshow()
{
System.out.println("Itisanexampleofstaticmethod.");
}
}

```

આઉટપુટ :

તે સ્થિર પદ્ધતિનું ઉદાહરણ છે.

ઈન્સ્ટન્સ પદ્ધતિ

વર્ગની પદ્ધતિને ઉદાહરણ પદ્ધતિ તરીકે ઓળખવામાં આવે છે. તે વર્ગમાં વ્યાખ્યાયિત બિન-સ્થિર પદ્ધતિ છે. દાખલા પદ્ધતિને બોલાવતા અથવા બોલાવતા પહેલા, તેના વર્ગનું ઓબ્જેક્ટ બનાવવું જરૂરી છે. ચાલો એક ઉદાહરણ પદ્ધતિનું ઉદાહરણ જોઈએ.

ઈન્સ્ટન્સ મેથડ Example.java

```

publicclassInstanceMethodExample
{
publicstaticvoidmain(String[]args)
{
//Creatinganobjectoftheclass

```

```

InstanceMethod Example obj= new Instance Metho-
dExample();

//invoking

instancemethodSystem.out.println("Thesumis:"+obj.
add(12,13));

}

ints;

//user-defined method because we have not used stat-
ickeywordpublicintadd(inta,intb)

{

s= a+b;

//returning the sumreturns;

}

}

```

આઉટપુટ :

The Sum is:25

દાખલાની પદ્ધતિના બે પ્રકાર છે:

- Accessor Method
- Mutator Method

એક્સેસર મેથડ : જે પદ્ધતિ(ઓ) ઇન્સ્ટન્સ વેરીએબલ વાંચે છે તેને એક્સેસર મેથડ તરીકે ઓળખવામાં આવે છે. અમે તેને સરળતાથી ઓળખી શકીએ છીએ કારણ કે પદ્ધતિ શબ્દ સાથે ઉપસર્ગ છે. તેને ગેટર તરીકે પણ ઓળખવામાં આવે છે. તે ખાનગી ક્ષેત્રનું મૂલ્ય પરત કરે છે. તે ખાનગી ક્ષેત્રના મૂલ્યને મેળવવા માટે વપરાય છે.

ઉદાહરણ

```

publicintgetId()

{

returnId;

}

```

મ્યુટેટર મેથડ : પદ્ધતિ(ઓ) ઇન્સ્ટન્સ વેરીએબલ વાંચે છે અને મૂલ્યોમાં ફેરફાર પણ કરે છે. અમે તેને સરળતાથી ઓળખી શકીએ છીએ કારણ કે પદ્ધતિ શબ્દ સમૂહ સાથે ઉપસર્ગ છે. તેને સેટર્સ અથવા મોડિફાયર તરીકે પણ ઓળખવામાં આવે છે. તે કંઈપણ પાછું આપતું નથી. તે ક્ષેત્ર પર આધાર રાખે છે તે જ ડેટા પ્રકારનું પરિમાણ સ્વીકારે છે. તેનો ઉપયોગ ખાનગી ક્ષેત્રનું મૂલ્ય નક્કી કરવા માટે થાય છે.

ઉદાહરણ

```

publicvoidsetRoll(introll)

{

this.roll=roll;

}

```

એક્સેસર અને મ્યુટેટર પદ્ધતિનું ઉદાહરણ વિદ્યાર્થી.Student.java

```

publicclassStudent

{

private int roll;privateStringname;

publicintgetRoll();//accessormethod

{

returnroll;

}

publicvoidsetRoll(introll)//mutatormethod

{

this.roll=roll;

}

publicStringgetName()

{

returnname;

}

publicvoidsetName(Stringname)

{

this.name=name;

}

publicvoiddisplay()

{

System.out.println("Roll no.:

"+roll);System.out.println("Studentname:"+name);

}

}

```


JAVA માં અમૂર્ત વર્ગો અને ઇન્ટરફેસો (Abstract Classes and Interfaces in JAVA)

ઉદ્દેશ્યો: આ પાઠના અંતે તમે સમર્થ હશો

- java OOPs કન્સેપ્ટ
- ઓબ્જેક્ટ ક્લોનિંગ
- પદ્ધતિના પ્રકાર.

એબ્સ્ટ્રેક્ટ પદ્ધતિ

જે પદ્ધતિમાં મેથડ બોડી નથી તે અમૂર્ત પદ્ધતિ તરીકે ઓળખાય છે. બીજા શબ્દોમાં કહીએ તો, અમલીકરણ વિના અમૂર્ત પદ્ધતિ તરીકે ઓળખાય છે. તે હંમેશા અમૂર્ત વર્ગમાં જાહેર કરે છે. તેનો અર્થ એ છે કે જો તેમાં અમૂર્ત પદ્ધતિ હોય તો વર્ગ પોતે જ અમૂર્ત હોવો જોઈએ. અમૂર્ત પદ્ધતિ બનાવવા માટે, અમે અમૂર્ત શબ્દનો ઉપયોગ કરીએ છીએ.

વાક્યરચના

```
abstract void method_name();
```

Example of abstract

method Demo.java

```
abstract class Demo // abstract class
```

```
{
// abstract method declaration
abstract void display();
}
```

```
public class MyClass extends Demo
```

```
{
// method implementation
void display()
{
System.out.println("Abstract method?");
}
```

```
public static void main(String args[])
```

```
{
// creating object of abstract
class Demo obj = new MyClass();
// invoking abstract method
obj.display();
}
```

આઉટપુટ :

Abstract method...

ફેક્ટરી પદ્ધતિ

તે એક પદ્ધતિ છે જે ઓબ્જેક્ટને તે વર્ગમાં પરત કરે છે જેનો તે સંબંધ છે. બધી સ્થિર પદ્ધતિઓ ફેક્ટરી પદ્ધતિઓ છે. ઉદાહરણ તરીકે, Number-Format obj = Number-Format.get-Number-Instance();

ઇનહેરિટન્સ

જાવામાં વારસો એ એક પદ્ધતિ છે જેમાં એક ઓબ્જેક્ટ પેરેન્ટ ઓબ્જેક્ટના તમામ ગુણધર્મો અને વર્તણૂકો મેળવે છે. તે OOPs (ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામિંગ સિસ્ટમ) નો મહત્વપૂર્ણ ભાગ છે.

જાવામાં વારસા પાછળનો વિચાર એ છે કે તમે નવા વર્ગો બનાવી શકો છો જે હાલના વર્ગો પર બનેલ છે. જ્યારે તમે અસ્તિત્વમાંના વર્ગમાંથી વારસામાં મેળવો છો, ત્યારે તમે પેરેન્ટ ક્લાસની પદ્ધતિઓ અને ક્ષેત્રોનો ફરીથી ઉપયોગ કરી શકો છો. વધુમાં, તમે તમારા વર્તમાન વર્ગમાં પણ નવી પદ્ધતિઓ અને ક્ષેત્રો ઉમેરી શકો છો.

ઇનહેરિટન્સ IS-A relation જહાજનું પ્રતિનિધિત્વ કરે છે જેને માતા-પિતા-બાળક સંબંધ તરીકે પણ ઓળખવામાં આવે છે.

શા માટે જાવામાં વારસાનો ઉપયોગ કરો

- મેથડ ઓવરરાઈડિંગ માટે (તેથી રન ટાઈમ પોલીમોર્ફિઝમ પ્રાપ્ત કરી શકાય છે).
- કોડ પુનઃઉપયોગીતા માટે.

ઇનહેરિટન્સ વપરાતી શરતો

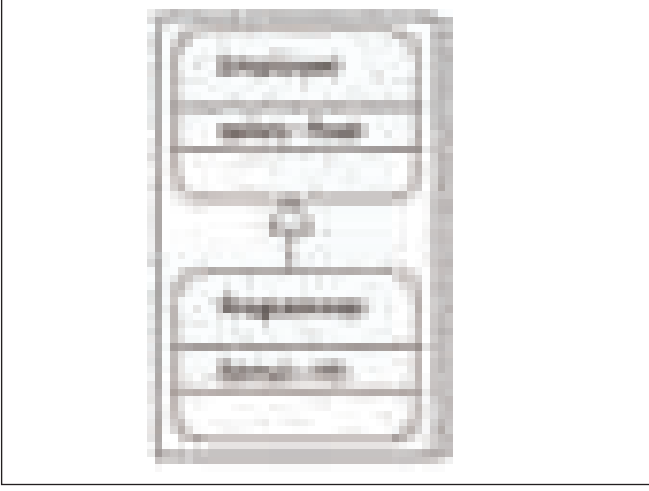
- **વર્ગ :** વર્ગ એ વસ્તુઓનો સમૂહ છે જેમાં સામાન્ય ગુણધર્મો હોય છે. તે એક ટેમ્પલેટ અથવા બ્લુપ્રિન્ટ છે જેમાંથી વસ્તુઓ બનાવવામાં આવે છે.
- **પેટા વર્ગ/બાળ વર્ગ :** પેટા વર્ગ એ એક વર્ગ છે જે અન્ય વર્ગને વારસામાં મેળવે છે. તેને વ્યુત્પન્ન વર્ગ, વિસ્તૃત વર્ગ અથવા બાળ વર્ગ પણ કહેવામાં આવે છે.
- **સુપર ક્લાસ/પેરેન્ટ ક્લાસ :** સુપરક્લાસ એ ક્લાસ છે જ્યાંથી પેટાક્લાસ સુવિધાઓને વારસામાં મેળવે છે. તેને બેઝક્લાસ અને પેરેન્ટક્લાસ પણ કહેવામાં આવે છે.
- **પુનઃઉપયોગીતા :** નામ સ્પષ્ટ કરે છે તેમ, પુનઃઉપયોગીતા એ એક પદ્ધતિ છે જે તમને નવો વર્ગ બનાવતી વખતે વર્તમાન વર્ગના ક્ષેત્રો અને પદ્ધતિઓનો પુનઃઉપયોગ કરવાની સુવિધા આપે છે. તમે અગાઉના વર્ગમાં પહેલાથી જ વ્યાખ્યાયિત સમાન ક્ષેત્રો અને પદ્ધતિઓનો ઉપયોગ કરી શકો છો.

જાવા ઇનહેરિટન્સનું સિનેક્સ

```
class Subclass-name extends Superclass-name
{
// methods and fields
}
```

વિસ્તૃત કીવર્ડ સૂચવે છે કે તમે એક નવો વર્ગ બનાવી રહ્યા છો જે હાલના વર્ગમાંથી ઉતરી આવ્યો છે. “વિસ્તરણ” નો અર્થ કાર્યક્ષમતા વધારવાનો છે.

જાવાની પરિભાષામાં, વારસામાં મળેલ વર્ગને પિતૃ અથવા સુપરક્લાસ કહેવામાં આવે છે, અને નવા વર્ગને ચિલ્ડર્સબક્લાસ કહેવામાં આવે છે.



જાવા ઇનહેરિટન્સ

ઉદાહરણ

ઉપરની આકૃતિમાં દર્શાવ્યા મુજબ, પ્રોગ્રામર એ સબક્લાસ છે અને કર્મચારી એ સુપરક્લાસ છે. એન્થેટવોકલાસીસી પ્રોગ્રામર SAE કર્મચારી વચ્ચેનો સંબંધ. તેનો અર્થ એ છે કે પ્રોગ્રામર એક પ્રકારનો કર્મચારી છે.

```
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 In ("Bonus of Program-
        meris:"+p.bonus);
    }
}
```

આઉટપુટ

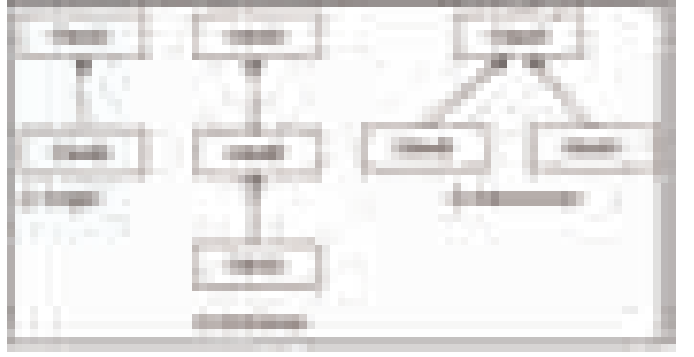
```
Programmer salary
is:40000.0Bonus of programme
ris:10000
```

ઉપરોક્ત ઉદાહરણમાં, પ્રોગ્રામર ઓબ્જેક્ટ પોતાના વર્ગના ક્ષેત્ર તેમજ કર્મચારી વર્ગ એટલે કે કોડ પુનઃઉપયોગિતાને એક્સેસ કરી શકે છે.

જાવામાં વારસાના પ્રકાર

વર્ગના આધારે, વારસાગત ઇન્જાવાના ત્રણ પ્રકારો છે: સિંગલ, બહુવિધ એલેન્ડેરાકિકલ.

જાવા પ્રોગ્રામિંગમાં, મલ્ટીપલ અને હાઇબ્રિડ વારસાને ફક્ત ઇન્ટરફેસ દ્વારા જ સપોર્ટ કરવામાં આવે છે. આપણે ઇન્ટરફેસલેટર વિશે શીખીશું.



નોંધ : વર્ગ દ્વારા જાવામાં બહુવિધ વારસાને સમર્થન નથી.

જ્યારે એક વર્ગમાં બહુવિધ વર્ગ હોય છે, ત્યારે તેને બહુવિધ વારસા તરીકે ઓળખવામાં આવે છે. દાખલા તરીકે:



સિંગલ વારસાનું ઉદાહરણ

જ્યારે એક વર્ગ બીજા વર્ગને વારસામાં મેળવે છે, ત્યારે તેને એક વારસો તરીકે ઓળખવામાં આવે છે. નીચે આપેલા ઉદાહરણમાં, ફૂતરો એક માલકલાસનો વર્ગ ધરાવે છે, તેથી એક વારસો છે.

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();
    }}
}
```

આઉટપુટ :

```
barking...
eating...
```

મલ્ટિલેવલ વારસાનું ઉદાહરણ

જ્યારે વારસાની સાંકળ હોય છે, ત્યારે તેને બહુસ્તરીય વારસા તરીકે ઓળખવામાં આવે છે. જેમ તમે નીચે આપેલ ઉદાહરણમાં જોઈ શકો છો, બેબીડોગ વર્ગ ડોગ વર્ગને વારસામાં મેળવે છે જે ફરીથી એનિમલ વર્ગને વારસામાં મેળવે છે, તેથી બહુસ્તરીય વારસો છે.

File:TestInheritance2.java

```
classAnimal{
    voideat(){System.out.println("eating...");}
}
class DogextendsAnimal{
    voidbark(){System.out.println("barking...");}
}
classBabyDog extendsDog{
    voidweep(){System.out.println("weeping...");}
}
classTestInheritance2{
    public static void main(String
    args[]){BabyDog d=new
    BabyDog();d.weep();
    d.bark();
    d.eat();
    }}
```

આઉટપુટ:weeping...barking...eating...

હાયરાર્કિકલ વારસાનું ઉદાહરણ

જ્યારે બે અથવા વધુ વર્ગો એક વર્ગને વારસામાં મેળવે છે, ત્યારે તેને હાયરાર્કિકલ વારસા તરીકે ઓળખવામાં આવે છે. નીચે આપેલા ઉદાહરણમાં, ડોગાન્ડ કેટ વર્ગો એનિમલ વર્ગને વારસામાં મેળવે છે, તેથી વંશવેલો વારસો છે.

File:TestInheritance3.java

```
classAnimal{
    voideat(){System.out.println("eating...");}
}
class DogextendsAnimal{
    voidbark(){System.out.println("barking...");}
}
classCatextendsAnimal{
    voidmeow(){System.out.println("meowing...");}
}
classTestInheritance3{
    public static void main(String
    args[]){Catc=newCat();
    c.meow();
    c.eat();
    //c.bark();//C.T.Error
}
```

}}

આઉટપુટ :

meowing...

eating...



શા માટે જાવામાં બહુવિધ વારસાને સમર્થન નથી?

જટિલતાને દૂર કરો અને ભાષાને વિસ્તૃત કરો, જાવામાં બહુવિધ વારસો સપોર્ટેડ નથી.

એક દૃશ્યનો વિચાર કરો જ્યાં A, B અને C ત્રણ વર્ગો છે. C વર્ગને A અને B વર્ગો વારસામાં મળે છે. જો A અને B વર્ગ સમાન પદ્ધતિ ધરાવે છે અને તમે તેને બાળ વર્ગના ઓબ્જેક્ટમાંથી કોલ કરો છો, તો A or B વર્ગની પદ્ધતિને કોલ કરવા માટે અસ્પષ્ટતા રહેશે.

કમ્પાઈલ સમયની ભૂલો રનટાઈમ ભૂલો કરતાં વધુ સારી હોવાથી, જાવા રેન્ડર કમ્પાઈલ ટાઈમ એરર જો તમને 2 વર્ગો વારસામાં મળે છે. જો તમારી પાસે સમાન પદ્ધતિ અલગ હશે, તો કમ્પાઈલ સમયની ભૂલ હશે.

```
classA{
    voidmsg(){System.out.println("Hello");}
}
class B{
    voidmsg(){System.out.println("Welcome");}
}
classCextendsA,B{//suppose ifitwere
    public static void main(String args[]){Cobj=newC();
    obj.msg();//Nowwhichmsg()methodwouldbeinvoked?
}
}
```

કમ્પાઈલ ટાઈમ ભૂલ

જાવામાં ઓવર રાઈડિંગની પદ્ધતિ

જો સબક્લાસ (ચાઈલ્ડ ક્લાસ)માં પેરેન્ટ ક્લાસમાં જાહેર કરવામાં આવેલી પદ્ધતિ સમાન હોય, તો તે જાવામાં મેથડ ઓવરરાઈડિંગ તરીકે ઓળખાય છે.

બીજા શબ્દોમાં કહીએ તો, જો પેટા વર્ગ પદ્ધતિનો ચોક્કસ અમલીકરણ પ્રદાન કરે છે જે તેના મૂળ વર્ગમાંથી એક દ્વારા જાહેર કરવામાં આવી છે, તો તે પદ્ધતિ ઓવરરાઈડિંગ તરીકે ઓળખાય છે.

જાવા મેથડ ઓવરરાઈડિંગનો ઉપયોગ

- મેથડ ઓવરરાઈડિંગનો ઉપયોગ પદ્ધતિના ચોક્કસ અમલીકરણ માટે થાય છે જે તેના સુપર ક્લાસ દ્વારા પહેલેથી જ પ્રદાન કરવામાં આવે છે.
- રનટાઈમ પોલીમોર્ફિઝમ માટે ઓવર રાઈડિંગ પદ્ધતિનો ઉપયોગ થાય છે.

જાવા મેથડ ઓવરરાઈડિંગ માટેના નિયમો

- થીમથોડમુસ્ટ વેથે સમાન નામ સિંથેપાર ટીક્લાસ
- થીમેથોડ એ પેરેન્ટ ક્લાસની જેમ જ પેરામીટર હોવું જોઈએ.
- ત્યાં અનિવાર્ય સંબંધ છે(તેમાં).

મેથ ડોવરરાઈડિંગ વિનાની સમસ્યાને સમજવી

ચાલો સમજીએ કે જો આપણે ઓવરરાઈડિંગ ન કરીએ તો પ્રોગ્રામનો સામનો કરી શકે છે.

```
//JavaProgramtodemonstratwhyweneedmethodoverriding
//Here,weare calling themethodof parentclasswithchild
//classobject.
//Creating a parent classclassVehicle{
voidrun(){System.out.println("Vehicleisrunning");}
}
//Creating a child classclassBikeextendsVehicle{
publicstaticvoidmain(Stringargs[]){
//creating an instance of child classBike obj =newBike();
//calling the method with child class instanceobj.run();
}
}
```

આઉટપુટ:

Vehicle is running

સમસ્યા એ છે કે મારે સબક્લાસમાં રન() પદ્ધતિનું ચોક્કસ અમલીકરણ પ્રદાન કરવું પડશે તેથી જ આપણે મેથડ ઓવરરાઈડિંગનો ઉપયોગ કરીએ છીએ.

પદ્ધતિ ઓવરરાઈડિંગનું ઉદાહરણ

આ ઉદાહરણમાં, અમે પેરેન્ટ ક્લાસમાં વ્યાખ્યાયિત કર્યા મુજબ પેટાક્લાસમાં રન મેથડ વ્યાખ્યાયિત કરી છે પરંતુ તેનો અમુક ચોક્કસ અમલીકરણ છે. પદ્ધતિનું નામ અને પરિમાણ સમાન છે, અને વર્ગો વચ્ચે IS-A સંબંધ છે, તેથી પદ્ધતિ ઓવરરાઈડિંગ છે.

//JavaProgram to illustrate the use ofJavaMethodOverriding

જાવા પર સવારી કરવાની પદ્ધતિ

```
//Creating a parent class.
classVehicle{
//definingamethod
voidrun(){System.out.println("Vehicleisrunning");}
}
//CreatingachildclassclassBike2extendsVehicle{
//definingthe samemethodas inthe parentclass
voidrun(){System.out.println("Bikeisrunningsafely");}
public static void main(String
args[]){Bike2 obj = new
Bike2();//creating
objectobj.run();//callingmethod
}
}
```

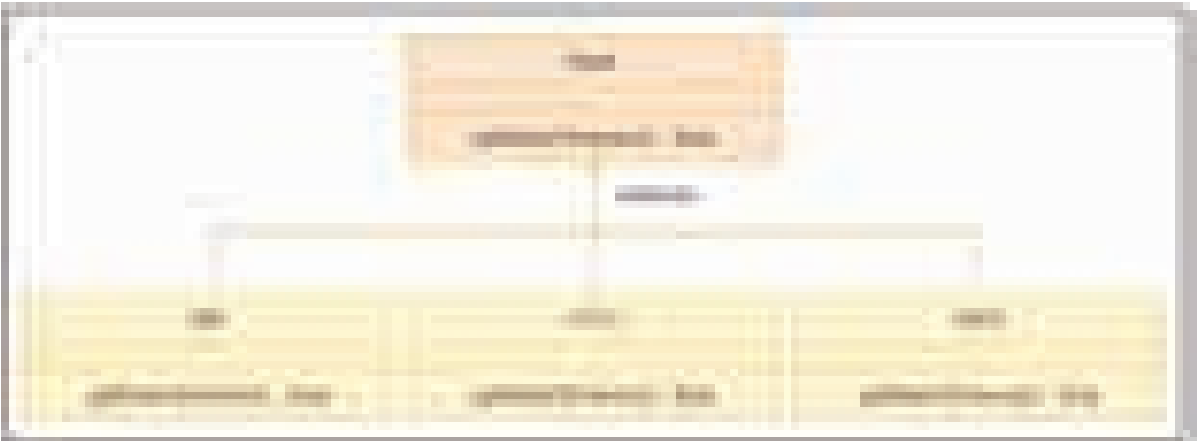
આઉટપુટ:

Bike is running safely

જાવા મેથડ ઓવરરાઈડિંગનું વાસ્તવિક ઉદાહરણ

એક દૃશ્યનો વિચાર કરો જ્યાં બેંક એવર્ગ કે જે વ્યાજ દર મેળવવા માટે કાર્યક્ષમતા પ્રદાન કરે છે. જો કે, વ્યાજનો દર બેંકો અનુસાર બદલાય છે. ઉદાહરણ તરીકે, SBI, ICICI અને AXIS બેંકો 8%, 7% અને 9% વ્યાજ આપી શકે છે.

નોંધ :જાવમેથોડઓવરરાઈડિંગનો સૌથી વધુ ઉપયોગ રનટાઈમ પોલીમોર્ફિઝમમાં થાય છે જે આપણે આગામી પૃષ્ઠોમાં શીખીશું.



//Java Program to demonstrate the real scenario of Java
Method Overriding

//where three classes are overriding the method of a
parent class.

//Creating a parent

```
class Bank{  
    int getRateOfInterest(){return 0;}  
}
```

//Creating child

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.getRateOfInterest());System.out.println
```

```
("AXIS Rate of Interest:"+a.getRateOfInterest());
```

```
}
```

```
}
```

આઉટપુટ :

SBI Rate of Interest: 8 ICICI Rate of Interest: 7
AXIS Rate of Interest: 9

શું આપણે સ્થિર પદ્ધતિને ઓવરરાઈડ કરી શકીએ?

ના, અસ્ટેટિક મેથડને ઓવરરાઈડ કરી શકાશે નહીં. તે રનટાઈમ પોલીમોર્ફિઝમ દ્વારા સાબિત થઈ શકે છે, પછીથી વાવરશે.

શા માટે આપણે સ્થિર પદ્ધતિને ઓવરરાઈડ કરી શકતા નથી?

તે એટલા માટે છે કારણ કે સ્થિર પદ્ધતિ વર્ગ સાથે બંધાયેલ છે જ્યારે ઉદાહરણ પદ્ધતિ ઑબ્જેક્ટ સાથે બંધાયેલ છે. સ્ટેટિક એ ક્લાસ એરિયાનું છે, અને ઉદાહરણ એ હીપ એરિયાનું છે.

શું આપણે જાવા મુખ્ય પદ્ધતિને ઓવરરાઈડ કરી શકીએ?

ના, કારણ કે મુખ્ય એક સ્થિર પદ્ધતિ છે. જાવામાં મેથડ ઓવરલોડિંગ અને મેથડ ઓવરરાઈડિંગ વચ્ચેનો તફાવત

મેથડ ઓવરલોડિંગ અને ઓવરરાઈડિંગ લેવાના વચ્ચેના તફાવત માટે મને ક્લિક કરો

મેથડ ઓવરરાઈડિંગ પર વધુ વિષયો (પ્રારંભિકો માટે નથી)

એક્સેસ મોડિફાયર સાથે મેથડ ઓવરરાઈડિંગ લેવાની છે

ચાલો એક્સેસ મોડિફાયર સાથે મેથડ ઓવરરાઈડિંગનો ખ્યાલ જોઈએ.

મેથડ ઓવરરાઈડિંગ સાથે અપવાદ અને લિંગ

ચાલો અપવાદ હેન્ડલિંગ સાથે ઓવરરાઈડિંગ પદ્ધતિનો ખ્યાલ જોઈએ.