કેસ્કેડિંગ સ્ટાઇલશીટ અને જાવાસ્ક્રિપ્ટ 2

કેસ્કેડિંગ સ્ટાઇલશીટ (Cascading Style- Sheet) અથવા CSS વેબસાઇટમાં આવેલા દર્શનીય ઘટકોની શૈલી (Style) સ્પષ્ટ કરવા માટેની સુવિધા પૂરી પાડે છે. દસ્તાવેજની માહિતીને તેને દર્શાવવાની વિગતોથી અલગ રાખવા તે મદદરૂપ બને છે. દસ્તાવેજને કેવી રીતે દર્શાવવો તેની વિગતોને સ્ટાઇલ (Style) તરીકે ઓળખવામાં આવે છે. સ્ટાઇલ જે-તે ઘટકનો દેખાવ સ્ક્રીન પર નક્કી કરે છે. સ્ટાઇલને અન્ય માહિતીથી અલગ રાખવાથી નીચે જણાવેલ લાભ મેળવી શકાય :

- કોડિંગમાં નકલ (duplication) ટાળી શકાય.
- એક જ વિગત માટે જુદા-જુદા ઉદ્દેશોને અનુલક્ષીને જુદી-જુદી સ્ટાઇલનો ઉપયોગ કરી શકાય.
- કોડની સરળ જાળવણી કરી શકાય.

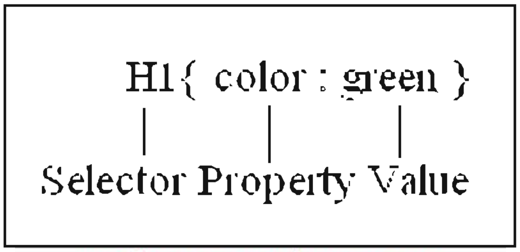
ઉદાહરણ તરીકે, HTMLમાં આપણે ઇચ્છતા હોઈએ કે <p> ટેગથી ઉમેરેલા કેટલાક ફકરાનું લખાણ ઘાટા (bold) અક્ષરોમાં આવવું જોઈએ, તો તે માટે સૉર્સકોડમાં આવેલ તમામ ફકરાના <p> ટેગ માટે અક્ષરો ઘાટા કરવા માટેનો <b> ટેગ ઉમેરવો પડે. જો વેબસાઇટ વિશાળ હોય તો ટેગનું આ પુનરાવર્તન કંટાળાજનક અને સમયનો વ્યય કરનારું બની રહે. પરંતુ CSSનો ઉપયોગ કરી આપણે તમામ ઘટકોનો પ્રકાર આપણી પસંદગી પ્રમાણેની સ્ટાઇલ સાથે દર્શાવી શકીએ. માટે, વેબસાઇટમાં વધુ સંખ્યામાં <p> ટેગ આવેલા હોય તોપણ, CSSનો ઉપયોગ કરી, તમામ <p> ટેગ માટે એકસાથે સ્ટાઇલ ગોઠવી શકીએ. આમ, એ કહી શકાય કે, HTMLનો ઉપયોગ માહિતીના વર્ણન માટે કરવામાં આવે છે, શૈલી માટે નહીં, જ્યારે CSS દસ્તાવેજમાં ઘટકોની શૈલીનું વર્ણન કરે છે, તેની વિગતોનું નહીં. વેબસાઇટમાં CSSનો ઉપયોગ કરી ફોન્ટના પ્રકાર, ફોન્ટ અને ઘટકોના રંગ, આજુબાજુની જગ્યા (Pad Space), હાંસિયા અને ઘટકોની સ્થિતિનું નિયંત્રણ કરી શકાય છે.

CSSની વાક્યરચના (Syntax of CSS)

CSSની વાક્યરચનામાં rules નામે ઓળખાતાં વિશિષ્ટ ચિહ્નોનો ઉપયોગ કરવામાં આવે છે. CSS rulesના બે મુખ્ય વિભાગ છે : પસંદગીકાર (selector) અને એક કે વધુ ઘોષણાઓ (declarations). સિલેક્ટર એ HTML ઘટક છે, જેના પર આપણે સ્ટાઇલને લાગુ કરવા માંગીએ છીએ. સિલેક્ટર તરીકે ઉપયોગમાં લેવામાં આવેલ HTML ઘટક સાથે સંકળાયેલ ગુણધર્મ અને તેને અનુરૂપ કિંમતોનો સમાવેશ ડિક્લેરેશન વિભાગમાં કરવામાં આવે છે. CSSની સામાન્ય વાક્યરચના નીચે મુજબ વ્યાખ્યાયિત કરી શકાય :

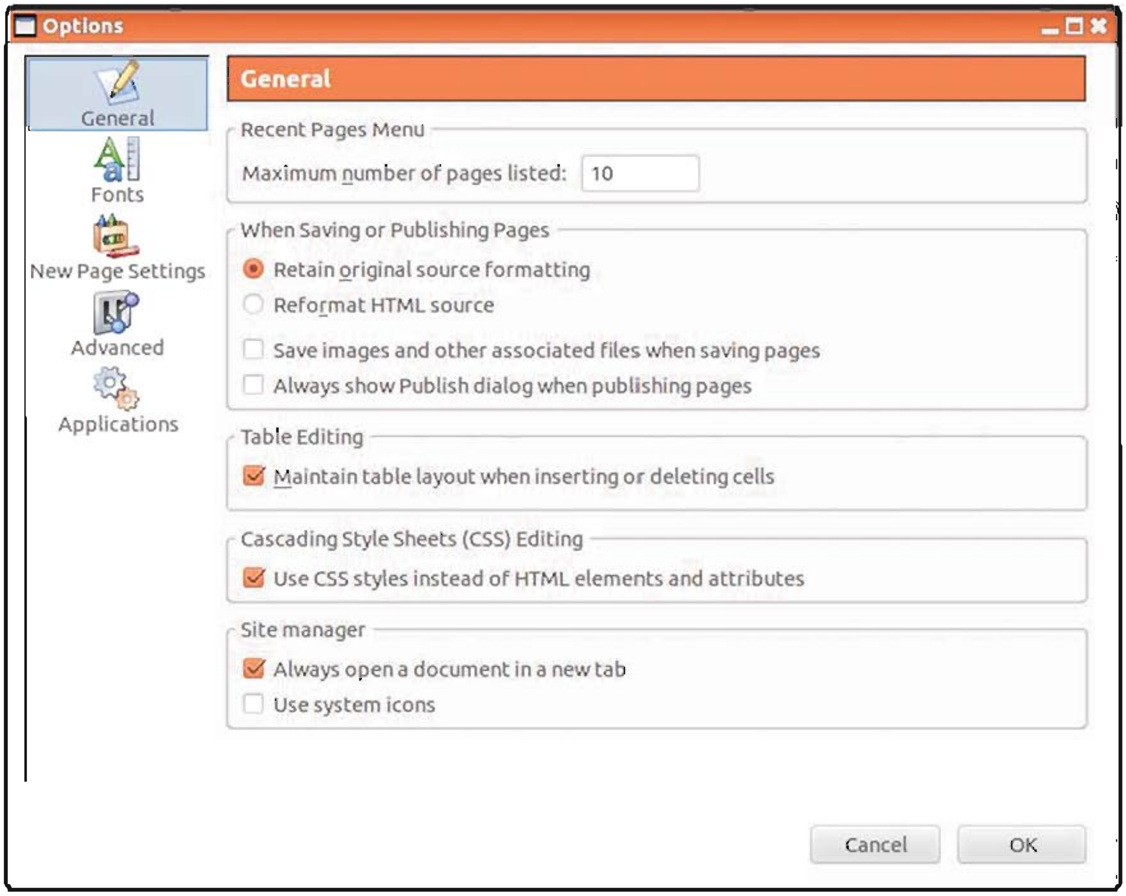
selector {property : value}

આ વાક્યરચનાનું એક ઉદાહરણ આકૃતિ 2.1માં આપ્યું છે.



આકૃતિ 2.1 CSSની વાક્યરચના

કમ્પોઝરનો ઉપયોગ કરી CSS સરળતાથી ઉમેરી શકાય છે. હવે, કમ્પોઝરમાં વેબપેજ સાથે CSS કેવી રીતે ઉમેરી શકાય, તેનો અભ્યાસ કરીએ. કમ્પોઝર પૂર્વનિર્ધારિત રીતે CSSનો ઉપયોગ કરે છે, તેની ખાતરી કરવા **Tools → Preferences** વિકલ્પ પસંદ કરો. આમ કરવાથી આકૃતિ 2.2માં દર્શાવ્યા મુજબ Option ડાયલોગબૉક્સ ખૂલશે. વિન્ડોની ડાબી બાજુ આવેલ General વર્ગ પસંદ કરો. તેમાં "Use CSS Style Instead of HTML Elements and Attributes" પસંદ કરેલ ન હોય, તો તેને પસંદ કરો. હવે, કમ્પોઝરમાં HTMLને બદલે લખાણની ગોઠવણી માટે CSS સ્ટાઇલનો ઉપયોગ કરવામાં આવશે.



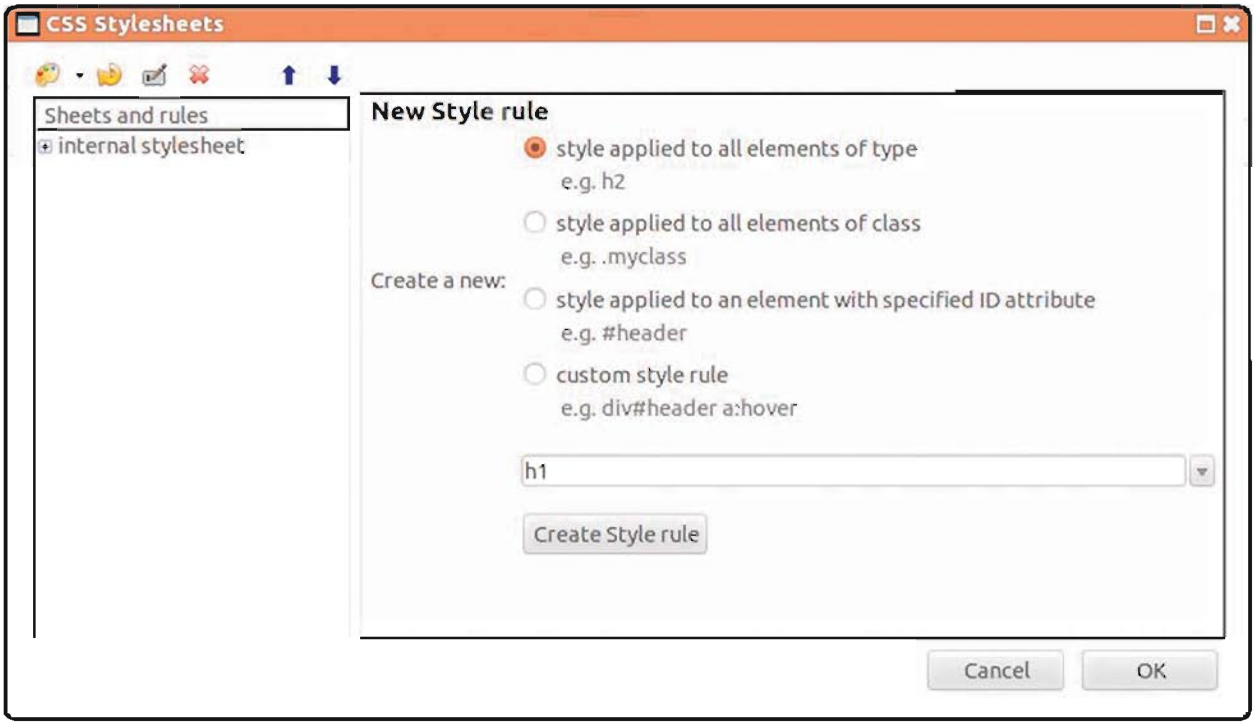
આકૃતિ 2.2 Options ડાયલોગબૉક્સ

ધારો કે, આપણે એક એવી વેબસાઇટની રચના કરવી છે, જેનાં તમામ વેબપેજનાં શીર્ષકો એક નિશ્ચિત શૈલીને અનુસરે. ઉદાહરણ તરીકે, વેબપેજમાં ઉમેરવામાં આવેલાં તમામ Heading1 (h1) નીચે આપેલ શૈલીને અનુસરવાં જોઈએ :

- Font : Times New Roman
- Case : Uppercase
- Alignment : Centre aligned
- Background color : Light Blue
- Border : Dotted Border

ઉપરનાં શીર્ષક માટે CSSની રચના કરવા માટે નીચે આપેલ પગલાંને અનુસરો.

- નવી ફાઇલ ખોલો. ફાઇલનો શીર્ષક આપી સંગ્રહ કરો.
- કમ્પોઝિશન ટૂલબાર પર Cascade બટન પર ક્લિક કરો. (નોંધ : જો ફાઇલનો સંગ્રહ કરવામાં ન આવ્યો હોય તો, Page title ડાયલોગબૉક્સ ખૂલશે. પાનાંને શીર્ષક આપ્યા પછી ફાઇલનો સંગ્રહ કરો.) આમ કરવાથી આકૃતિ 2.3માં દર્શાવ્યા મુજબનું ડાયલોગબૉક્સ ખૂલશે. આ ડાયલોગબૉક્સનો ઉપયોગ કરી દરેક ઘટક માટે સ્ટાઇલ વ્યાખ્યાયિત કરી શકાય છે. પ્રથમ રેડિયો-બટન "style applied to all the elements of type" પર ક્લિક કરો.



આકૃતિ 2.3 : CSS Stylesheets ડાયલોગબૉક્સ

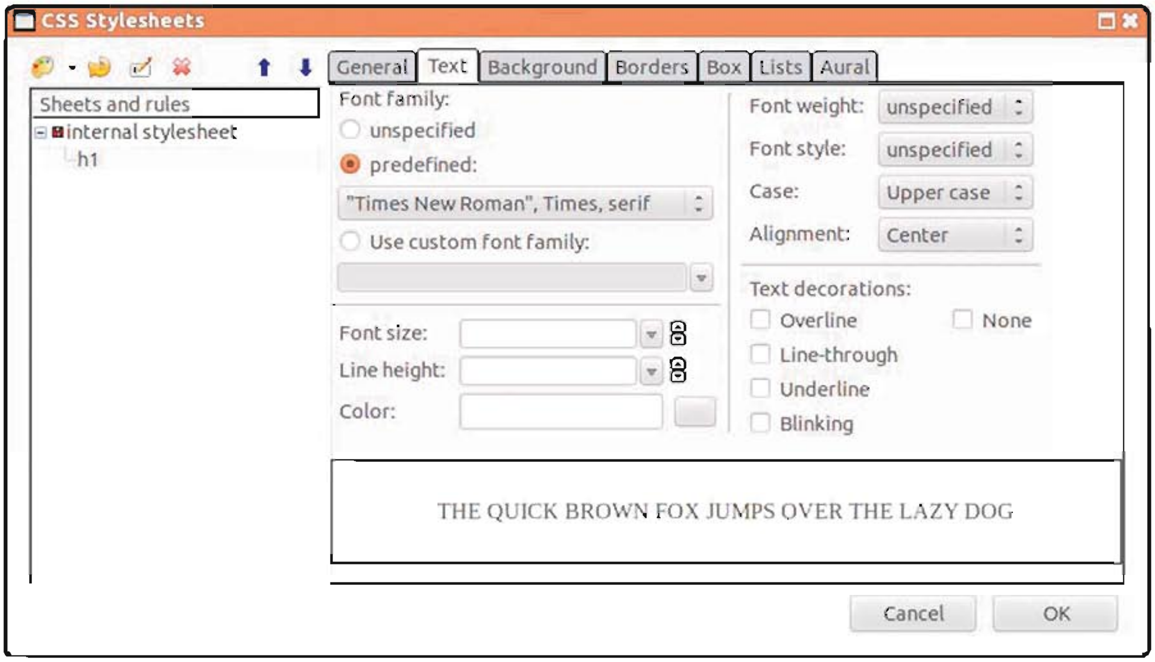
- જેના માટે શૈલીના નિયમો (Style rule) વ્યાખ્યાયિત કરવાના હોય તે ઘટકને ડ્રૉપડાઉન મેનુમાંથી પસંદ કરો. આપણે Heading1 માટે શૈલીની રચના કરવાની હોવાથી આકૃતિ 2.3માં ડ્રૉપડાઉન મેનુમાંથી h1 (Heading1) પસંદ કરવામાં આવ્યું છે.

- Create Style rule બટન પર ક્લિક કરો. જોઈ શકાશે કે CSS Stylesheets ડાયલોગબૉક્સ ખૂલ્લું રહે છે, પરંતુ તેમાંનાં વિકલ્પો બદલાઈ ગયા છે.

- આકૃતિ 2.4માં ડાબી બાજુના વિભાગમાં internal stylesheet શીર્ષકની તરત નીચે h1 ઘટક દર્શાવ્યો છે. જે ઘટક માટે શૈલીના નિયમ વ્યાખ્યાયિત કરવામાં આવે તે દરેકને internal stylesheet શીર્ષક હેઠળ આ યાદીમાં દર્શાવવામાં આવે છે. ડાયલોગબૉક્સમાં જમણી બાજુ General, Text, Background, Borders, Box, Lists અને Aural જેવા વિવિધ વિભાગો (tabs) જોઈ શકાય છે. ઘટકને નિશ્ચિત શૈલી આપવા માટે આ દરેક વિભાગનો ઉપયોગ કરી શકાય છે. પરંતુ જે-તે ઘટક માટે શૈલીના નિયમની રચના કરવામાં આવે, ત્યારે આ બધા જ વિભાગો તેને લાગુ ન પડે એમ પણ બને. હવે, h1 સિલેક્ટર માટે શૈલીના નિયમોની રચના કરીએ.

આકૃતિ 2.4 : h1ના વિકલ્પો સાથે CSS stylesheets ડાયલોગબૉક્સ

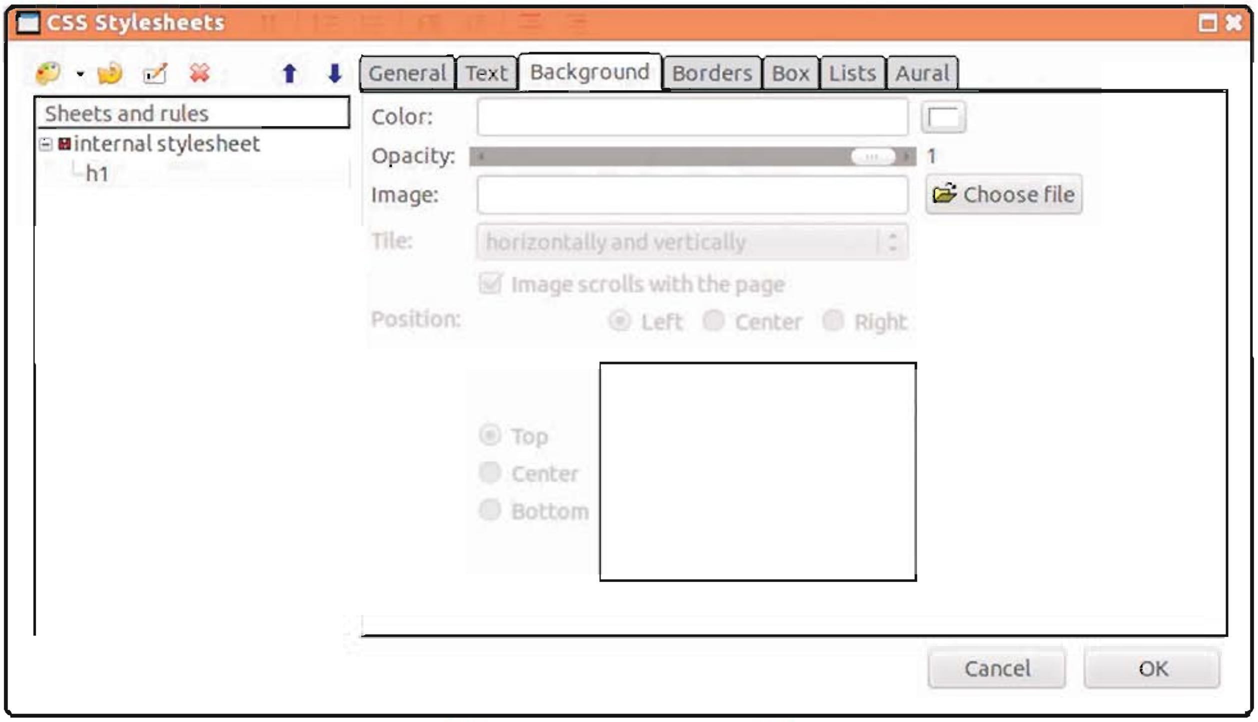
- ડાયલોગબૉક્સની ડાબી બાજુના વિભાગમાં આપેલ h1ને પસંદ કરો. આકૃતિ 2.5માં દર્શાવ્યા મુજબ Text વિભાગ પસંદ કરો. આમ કરવાથી Font family, Font size, Line height, Color, Case, Alignment જેવા વિવિધ વિકલ્પો દર્શાવવામાં આવશે.



આકૃતિ 2.5 : Text વિભાગ

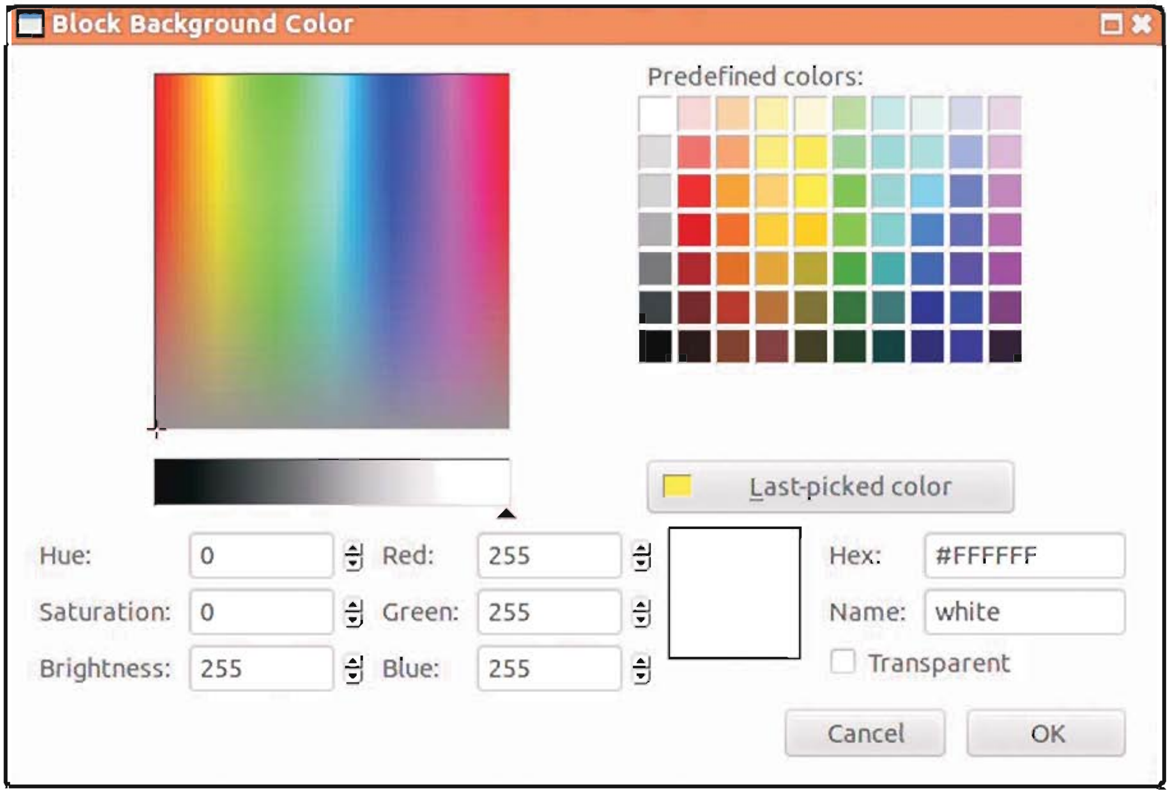
- h1 માટે જરૂરી શૈલી માટેના વિકલ્પોની પસંદગી કરીએ. Font family વિભાગમાં predefined રેડિયો-બટન પસંદ કરી ડ્રૉપડાઉન મેનુમાંથી "Times New Roman" વિકલ્પ પસંદ કરો. Case મેનુમાંથી "Upper Case" અને Alignment મેનુમાંથી "Center" વિકલ્પ પસંદ કરો. પસંદ કરેલા વિકલ્પો આકૃતિ 2.5માં જોઈ શકાય છે. વિકલ્પો બદલવાથી વિન્ડોમાં આવેલું પૂર્વનિર્ધારિત લખાણ પણ શૈલીને અનુરૂપ આપોઆપ બદલાઈ જાય છે તે જોઈ શકાશે.

- બૅકગ્રાઉન્ડ રંગ પસંદ કરવા માટે Background વિભાગ (tab) પસંદ કરો. આમ કરવાથી આકૃતિ 2.6માં દર્શાવ્યા મુજબ બૅકગ્રાઉન્ડ માટેના વિકલ્પો ખૂલશે.



આકૃતિ 2.6 : Background વિભાગ

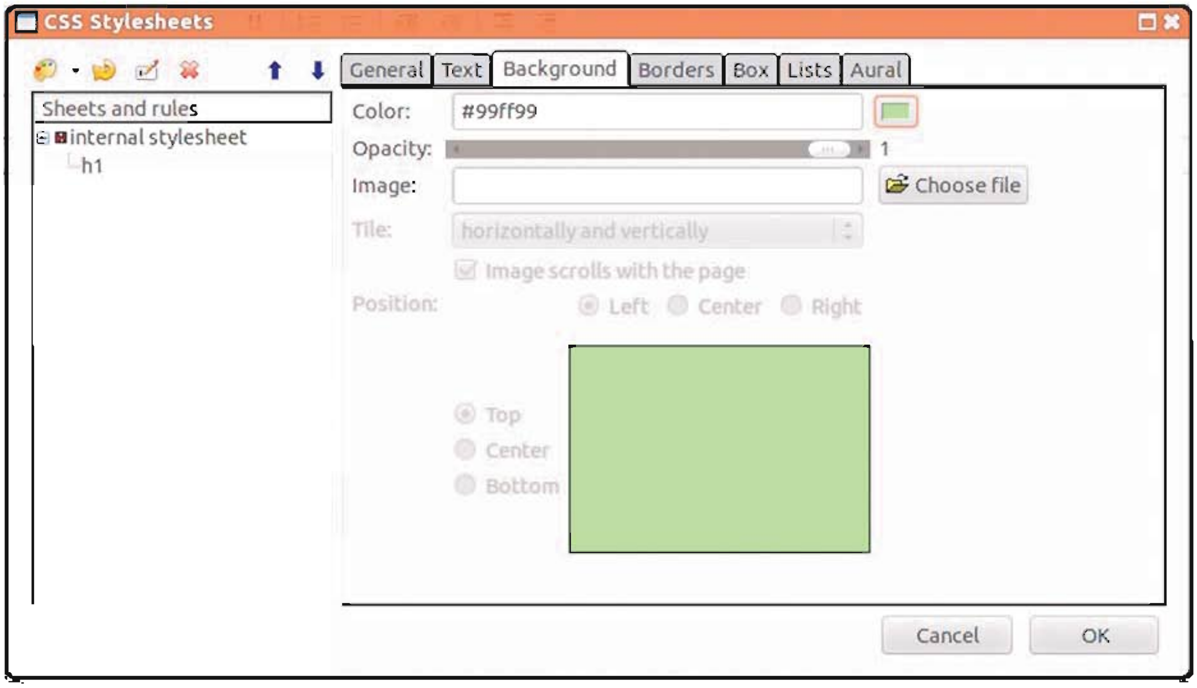
- Color વિકલ્પમાં આવેલ Color Palette બટન પર ક્લિક કરો. આકૃતિ 2.7માં દર્શાવ્યા મુજબ Block Background Color ડાયલોગબૉક્સ રજૂ કરવામાં આવશે. તમારો અપેક્ષિત રંગ પસંદ કરી OK બટન પર ક્લિક કરો.



આકૃતિ 2.7 : Block Background Color ડાયલોગબૉક્સ

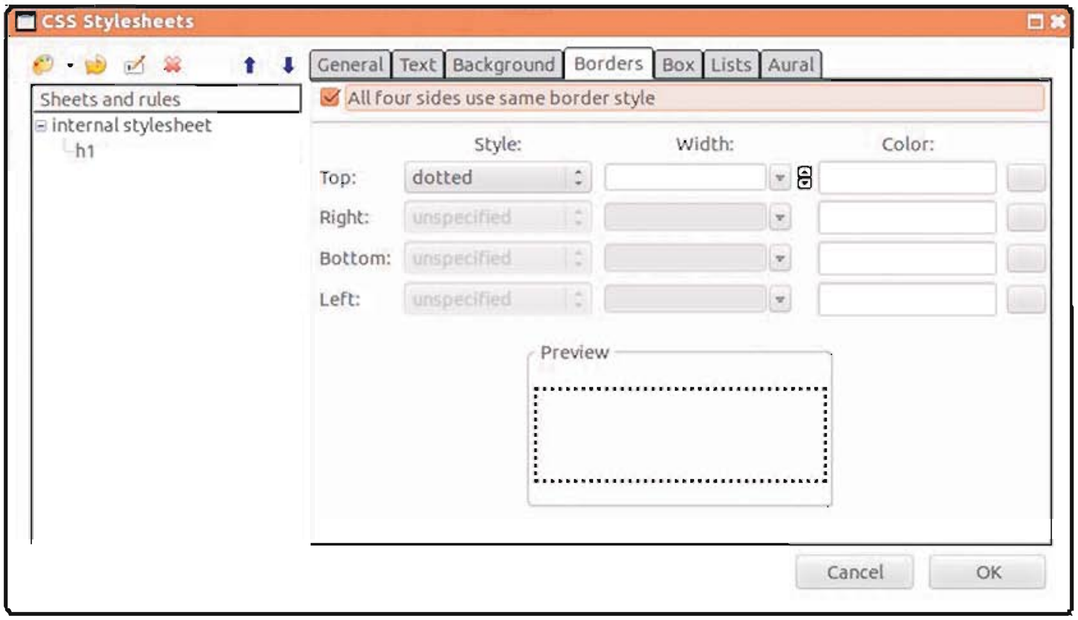
- રંગ પસંદ કર્યા પછી બૅકગ્રાઉન્ડ વિભાગનો દેખાવ આકૃતિ 2.8માં દર્શાવ્યો છે.

  નોંધ : જો બૅકગ્રાઉન્ડ તરીકે ચિત્રનો ઉપયોગ કરવા માંગતા હોઈએ, તો Image વિકલ્પમાં "Choose file" પર ક્લિક કરી બૅકગ્રાઉન્ડ માટેની ફાઇલ પસંદ કરવી.



આકૃતિ 2.8 : h1 માટે Background વિભાગમાં પસંદ કરવામાં આવેલા વિકલ્પો

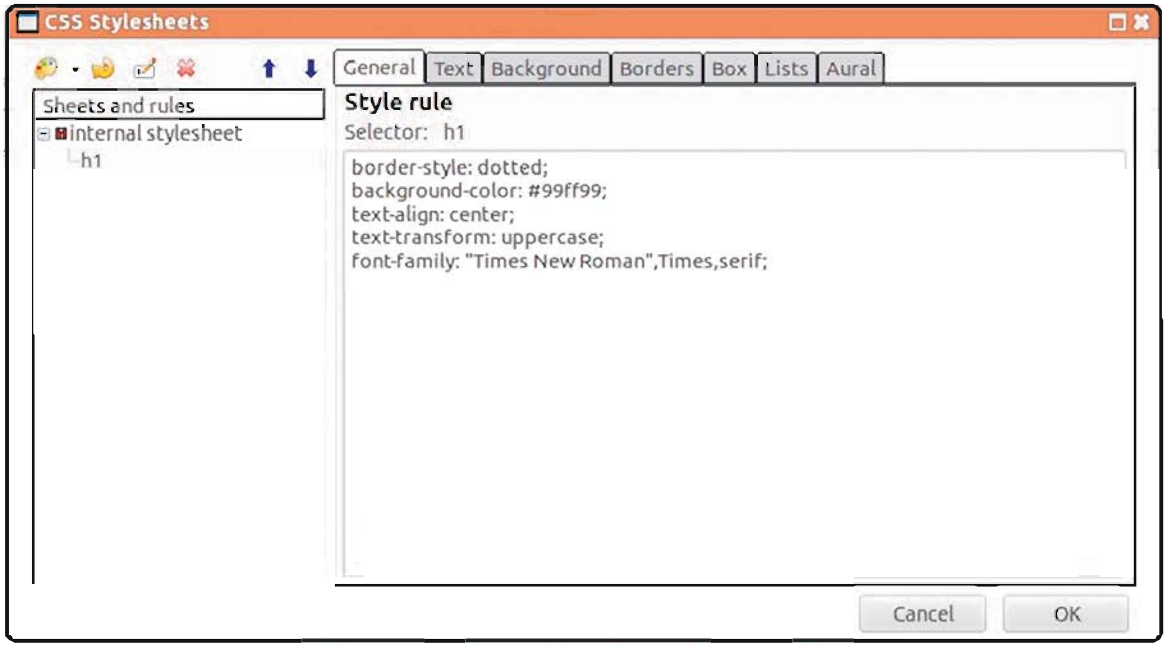
- સીમારેખા ઉમેરવા માટે Border વિભાગ પસંદ કરો. આકૃતિ 2.9માં દર્શાવ્યા મુજબ સીમારેખા માટેના વિકલ્પો દર્શાવવામાં આવશે. જો ચારેબાજુ પર એકસમાન સીમારેખા દર્શાવવી હોય, તો "All four sides use same border style" લખાણની આગળ આવેલું ચેકબૉક્સ પસંદ કરો. હવે, માત્ર "Top" વિકલ્પ ઉપલબ્ધ રહેશે, અન્ય તમામ વિકલ્પો અનુપલબ્ધ બનશે. આમ થવાનું કારણ એ છે કે, એક જ બાજુને આપવામાં આવેલી અસર તમામ ચાર બાજુઓ પર લાગુ પાડવામાં આવશે. Style ડ્રૉપડાઉન મેનુમાંથી Dotted વિકલ્પ પસંદ કરો. અહીં સીમારેખા માટે અપેક્ષિત જાડાઈ (width) અને રંગ (color) પણ પસંદ કરી શકાય છે. સીમારેખા માટે પસંદ કરેલ શૈલીનું પૂર્વદર્શન (preview) પણ કરી શકાય છે. OK બટન પર ક્લિક કરો.



આકૃતિ 2.9 : h1 માટે Border વિભાગમાં પસંદ કરવામાં આવેલા વિકલ્પ

CSS Stylesheet ડાયલોગબૉક્સમાં અન્ય વિભાગ પસંદ કરવામાં આવે, તો બીજા વિકલ્પ પણ ઉપલબ્ધ છે. જોકે, આપણા ઉદાહરણમાં h1ની શૈલી લાગુ પાડવા માટે માત્ર પસંદગીના વિકલ્પોની જરૂર છે.

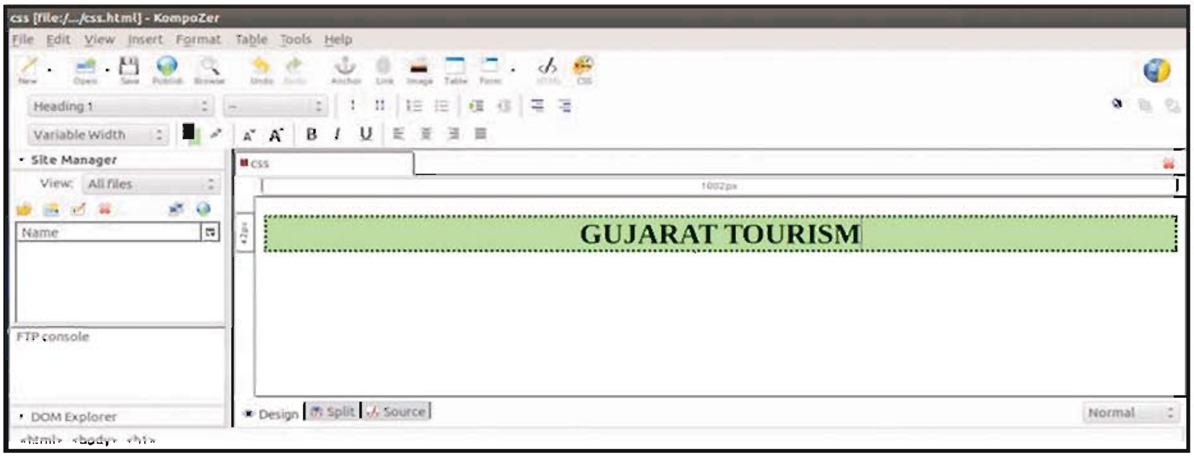
- General વિભાગ પર ક્લિક કરો. આકૃતિ 2.10માં દર્શાવ્યા મુજબ નિર્માણ કરવામાં આવેલ CSS કોડ દર્શાવવામાં આવશે. જો HTMLની જાણકારી હોય, તો આ કોડ સુધારી પણ શકાય છે.



આકૃતિ 2.10 : CSS કોડ દર્શાવતો General વિભાગ

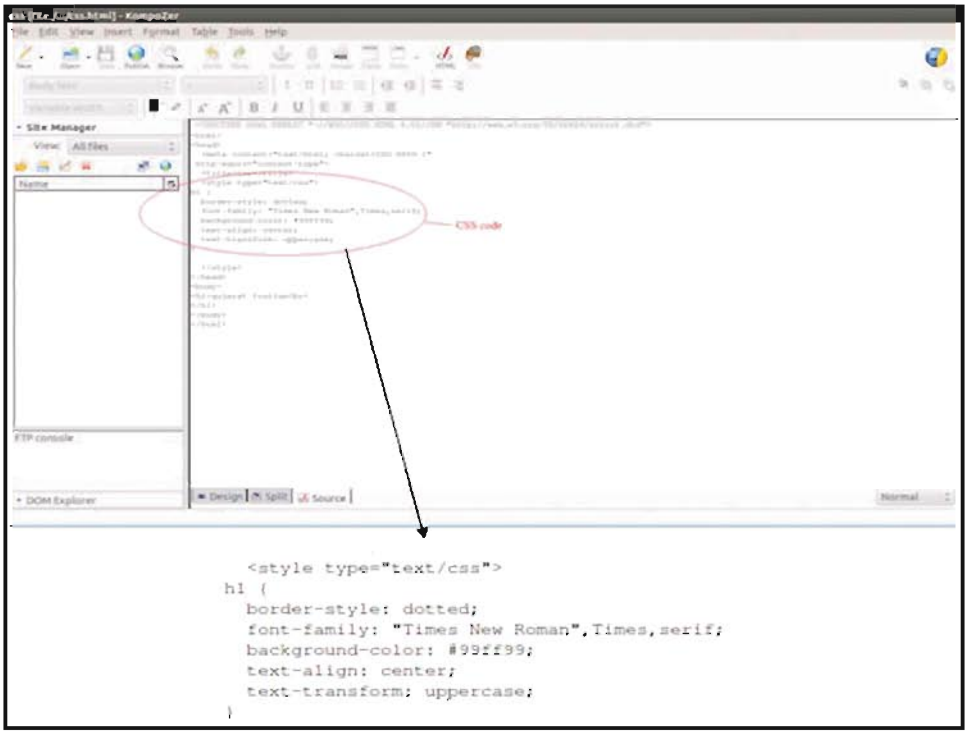
આમ, આપણે h1ને લાગુ પાડી શકાય, તેવી શૈલીની રચના કરી. વેબપેજના અન્ય કોઈ પણ ઘટકની શૈલીની રચના કરવા માટે આ જ પદ્ધતિનો ઉપયોગ કરી શકાય. અન્ય ઘટક ઉમેરવા માટે CSS Stylesheet ડાયલોગબૉક્સની ઉપર ડાબી બાજુના ખૂણા પર આવેલા CSS બટન ની તીરની નિશાની પર ક્લિક કરો. ડ્રૉપડાઉન મેનુમાંથી "Style-rule" વિકલ્પ પસંદ કરો. શૈલી માટેના નિયમ ઉમેર્યા બાદ OK બટન પર ક્લિક કરો.

હવે, h1 ટેગ માટે રચવામાં આવેલ CSSનો ઉપયોગ વેબપેજમાં કરીએ. પાનાંમાં શીર્ષક ઉમેરવા માટે ફૉર્મેટ ટૂલબાર-1માંથી Heading1 વિકલ્પ પસંદ કરો. Heading1 વિકલ્પ પસંદ કરવામાં આવે, ત્યારે કર્સર આપોઆપ વેબપેજની મધ્યમાં ગોઠવાઈ જાય છે. (કારણકે આપણે CSSમાં Center ગોઠવણ લાગુ પાડી છે.) થોડું લખાણ ઉમેરો. લખાણ અપરકેસમાં અને Times New Roman ફૉન્ટ સાથે ઉમેરાશે. આકૃતિ 2.11 CSS લાગુ પાડવામાં આવી હોય તેવું Heading1 લખાણ દર્શાવે છે. માટે, હવે પછી કોઈ પણ પાનામાં Heading1 ઉમેરવામાં આવશે, તો તેના માટેની શૈલી દરેક વખતે સમાન રહેશે.



આકૃતિ 2.11 : વેબપેજમાં લાગુ પાડવામાં આવેલ CSS

CSS stylesheet કોડ જોવા માટે વિન્ડોની નીચેના ભાગમાં આવેલ Source વિભાગ પસંદ કરો. આકૃતિ 2.12 પાનાંના સૉર્સકોડમાં આવેલ શીર્ષક વિભાગનો CSS કોડ દર્શાવે છે.



આકૃતિ 2.12 : Source વ્યૂમાં CSS કોડ

CSSના ફાયદા (Advantages of CSS)

ઉપર્યુક્ત ચર્ચા બાદ કહી શકાય કે, CSS તેની નીચે આવેલા સ્તરમાં ફેરફાર કર્યા વગર વેબસાઇટની સંરચના કરવાની સુવિધા પૂરી પાડે છે. વેબસાઇટની ગોઠવણ અને સંરચના માટેના ગુણધર્મોને તેની નીચે આવેલા તેના તાર્કિક બંધારણથી અલગ રાખવામાં આવે, તો વેબસાઇટમાં ફેરફાર કરતી વખતે તેમાં આવેલી વિગતોમાં અકસ્માતે ફેરફાર થવાનો ભય રહેતો નથી.

ઇચ્છિત ઘટકની શૈલીમાં સુધારો કરવા માટે પ્રોગ્રામરે માત્ર CSS ફાઇલમાં ફેરફાર કરવાનો રહે છે, જે કાર્યને સરળ બનાવે છે. દરેક ઘટક માટેની શૈલી માત્ર એક વાર જ ઉમેરવાની હોવાથી, HTMLની સરખામણીએ CSSમાં ઓછો કોડ ઉમેરવો પડે છે. અને તેનાથી વેબપેજ ઝડપથી દર્શાવી શકાય છે. CSSનો ઉપયોગ વેબસાઇટની રચનાને ઝડપી અને કાર્યક્ષમ બનાવે છે.

CSSના ગેરફાયદા (Disadvantages of CSS)

જુદાં-જુદાં બ્રાઉઝર માટે CSSની સુસંગતતા બદલાતી રહે છે. આનો અર્થ એ થયો કે, સ્ટાઇલશીટની કેટલીક સુવિધાઓને દરેક બ્રાઉઝર સમર્થન આપતું નથી અને ઉપયોગકર્તાની અપેક્ષા અનુસાર તે શૈલીને દર્શાવી શકતું નથી. પરંતુ હવે, બ્રાઉઝરની અદ્યતન આવૃત્તિઓ વધુ પ્રમાણભૂત બની છે અને સુસંગતતાનો મુદ્દો ક્ષીણ થયેલો છે.

જાવાસ્ક્રિપ્ટ (JavaScript)

મૂળભૂત રીતે વેબપેજમાં દેખાવ (appearance)નું નિયંત્રણ કરવા માટે HTMLનો ઉપયોગ કરવામાં આવે છે. HTMLમાં બનાવવામાં આવેલા વેબપેજ સ્થિત (Static) છે અને બ્રાઉઝર રેન્ડર કરે પછી તેને બદલી શકાતાં નથી. બીજી તરફ, ઇન્ટરનેટના વિકાસ સાથે લોકો વધુ સારું સંવાદન અને દર્શનીય આલેખનની અપેક્ષા રાખતા થયા. પરંતુ HTML માત્ર અપરિવર્તનશીલ વેબપેજ આપી શકે છે. આમ, વેબ માટેની નવી પ્રોગ્રામિંગ ભાષામાં વધુ સંવાદનની જરૂર ઊભી થઈ. તેથી નેટસ્કેપે જાવાસ્ક્રિપ્ટનો વિકાસ કર્યો. જાવાસ્ક્રિપ્ટ એ વેબપેજમાં પ્રોગ્રામિંગનું પાસું ઉમેરવાની સુવિધા આપતી સ્ક્રિપ્ટિંગ ભાષા છે.

સ્ક્રિપ્ટિંગ ભાષા સરળ અને હળવી પ્રોગ્રામિંગ ભાષા છે, જેમાં 'સી' કે 'જાવા' ભાષા જેવાં અદ્યતન કાર્યોનો સમાવેશ કરવામાં આવ્યો હોતો નથી. વેબસાઇટના આલેખનમાં સુધારો કરવા તથા ફૉર્મની યથાર્થતા ચકાસવા માટે વેબપેજમાં જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. તે HTML પાનામાં સંવાદન ઉમેરે છે. અને તેને HTML કોડમાં પ્રત્યક્ષ રૂપે ઉમેરવામાં આવે છે. હાલમાં મોઝિલા ફાયરફોક્સ, ક્રોમ, સફારી અને ઇન્ટરનેટ એક્સપ્લોરર જેવાં તમામ વેબબ્રાઉઝર જાવાસ્ક્રિપ્ટને સમર્થન આપે છે. જાવાસ્ક્રિપ્ટના ઉપયોગ પછી વેબપેજ અપરિવર્તનશીલ રહેતું નથી, પરંતુ ઉપયોગકર્તા સાથેના સંવાદન, બ્રાઉઝરનું નિયંત્રણ અને HTML વિગતોની ગતિશીલ (dynamic) રજૂઆત માટેના કોડનો સમાવેશ કરી શકે છે.

કમ્પોઝરની મદદથી ફૉર્મ કેવી રીતે બનાવી શકાય તે આપણે જોયું. જ્યારે ઉપયોગકર્તા ફૉર્મમાં વિગતો ઉમેરે ત્યારે, તે કેટલાંક મહત્ત્વનાં ફિલ્ડને ખાલી છોડી દે અથવા ફિલ્ડમાં વિગતો ખોટા સ્વરૂપમાં ઉમેરે તેવું પણ બને. આવા કિસ્સામાં કેટલીક યથાર્થતા પૂરી પાડવી જરૂરી બને છે. યથાર્થતા ઉપયોગકર્તાને ભૂલ કરતા અટકાવે છે. જ્યારે ઉપયોગકર્તા ખોટી વિગત ઉમેરે અથવા ફિલ્ડને ખાલી છોડી દે ત્યારે ભૂલનો સંદેશ દર્શાવવો જોઈએ અને ફૉર્મને સબમિટ થતાં રોકવું જોઈએ.

હાલમાં, જાવાસ્ક્રિપ્ટના સૌથી સામાન્ય વિનિયોગ ક્લાયન્ટ બાજુની સ્ક્રિપ્ટ છે, જે વેબબ્રાઉઝરમાં ચલાવવામાં આવે છે. HTML ફૉર્મમાં ઉમેરેલી વિગતોને સર્વર તરફ મોકલવામાં આવે તે પહેલાં ક્લાયન્ટ બાજુ તેની યથાર્થતા ચકાસવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. જાવાસ્ક્રિપ્ટનો ઉપયોગ કરી સામાન્ય રીતે ફૉર્મમાં નીચેની બાબતો તપાસવામાં આવે છે :

- ઉપયોગકર્તાએ કોઈ જરૂરી ફિલ્ડ ખાલી છોડી દીધું છે ?
- ઉપયોગકર્તાએ યોગ્ય ઇ-મેઇલ સરનામું દાખલ કર્યું છે ?

- બે ફિલ્ડની વિગતો સમાન છે કે નહીં ?
- ઉપયોગકર્તાએ યોગ્ય તારીખ દાખલ કરી છે ?
- ઉપયોગકર્તાએ આંકડાકીય ફિલ્ડમાં લખાણ ઉમેર્યું છે ? ઉદાહરણ તરીકે, જથ્થા (quantity) માટેના ફિલ્ડમાં આંકડાકીય વિગતને બદલે ઉપયોગકર્તા શાબ્દિક વિગત દાખલ કરે.

ફૉર્મની યથાર્થતા કેવી રીતે ચકાસવી, તેનો અભ્યાસ કરતાં પહેલાં જાવાસ્ક્રિપ્ટ કોડ કેવી રીતે લખવો તે શીખીએ.

જાવાસ્ક્રિપ્ટ કોડનો અમલ HTML વેબપેજમાં જ કરવામાં આવે છે. આમ, જાવાસ્ક્રિપ્ટ કોડને વેબપેજમાં સીધા જ એક સ્વતંત્ર વિભાગ સ્વરૂપે મૂકી શકાય છે. HTML પાનામાં <script>...</script> ટેગનો ઉપયોગ કરી જાવાસ્ક્રિપ્ટનો કોડ ઉમેરવામાં આવે છે. <script> અને </script> ટેગની વચ્ચે આવેલી લીટીઓ જાવાસ્ક્રિપ્ટનો કોડ ધરાવે છે. યાદ રાખો કે, જાવાસ્ક્રિપ્ટ એ 'કેસ-સેન્સિટિવ' ભાષા છે.

જાવાસ્ક્રિપ્ટને HTMLના <body> અથવા <head> વિભાગમાં સમાવવામાં આવે છે, પરંતુ કોડને <head> ટેગમાં ઉમેરવામાં આવે તે વધુ ઇચ્છનીય છે. <script> ટેગબ્રાઉઝરને તેની વચ્ચે રહેલાં તમામ લખાણોને એક સ્ક્રિપ્ટ તરીકે અર્થઘટન કરવાનો નિર્દેશ કરે છે. કમ્પોઝરના સૉર્સવ્યૂનો ઉપયોગ કરી એક સરળ જાવાસ્ક્રિપ્ટ કોડ લખીએ. આકૃતિ 2.13 જાવાસ્ક્રિપ્ટ સાથેનો સૉર્સવ્યૂ દર્શાવે છે. ફાઇલનો સંગ્રહ કરી તેને વેબબ્રાઉઝરમાં ખોલો.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
 http-equiv="content-type">
  <title>js1</title>
  <script>
document.write("HELLO STUDENTS");
  </script>
</head>
<body>
<br>
</body>
</html>
```

આકૃતિ 2.13 : સૉર્સવ્યૂમાં જાવાસ્ક્રિપ્ટ

અહીં જોઈ શકાય છે કે, <script> ટેગમાં માત્ર એક વિધાન "document.write("HELLO STUDENTS")" લખવામાં આવ્યું છે. અહીં write એક પ્રક્રિયા (method) છે, જે document ઑબ્જેક્ટનો એક ભાગ છે. જે લખાણ દર્શાવવાનું હોય, તેને document.write() પ્રક્રિયાના ચલ તરીકે આપવામાં આવે છે.

આકૃતિ 2.13માં વિધાનના અંતમાં આપવામાં આવેલું અર્ધવિરામ ચિહ્ન (semicolon) જાવાસ્ક્રિપ્ટમાં વૈકલ્પિક છે. અર્ધવિરામ ચિહ્નનો ઉપયોગ સમાપ્તિ (termination) ને બદલે અલગતા (separation) માટે થાય છે. તેથી જો દરેક વિધાન નવી લીટીમાં લખવામાં આવે, તો અર્ધવિરામ મૂકવાની જરૂર નથી. આપણા ઉદાહરણમાં document.write વિધાન પછીનું અર્ધવિરામ કાઢી નાખવામાં આવશે તોપણ પરિણામ સમાન રહેશે. આમ છતાં, અર્ધવિરામનો ઉપયોગ એક સારી પ્રોગ્રામિંગ પ્રથા છે. આકૃતિ 2.14માં કોડના પરિણામને બ્રાઉઝરમાં દર્શાવ્યું છે.

આકૃતિ 2.14 : બ્રાઉઝરમાં પરિણામ

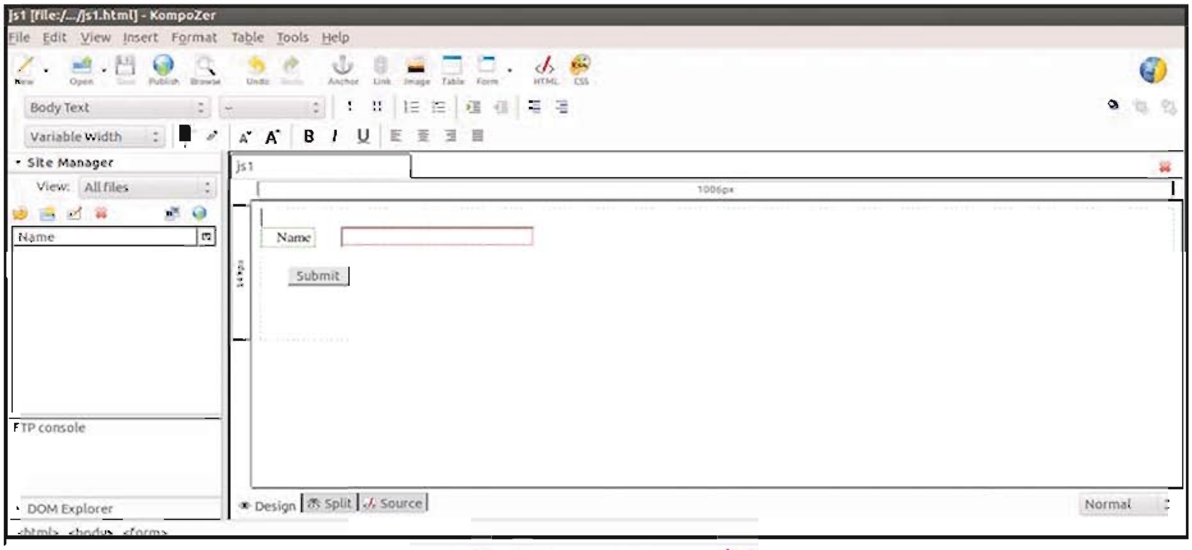
નોંધ : જાવાસ્ક્રિપ્ટ વિધાનોને બ્લૉક સ્વરૂપે એકત્રિત કરી શકાય છે. બ્લૉકની શરૂઆત અને અંત છગડિયા કૌંસ ({ }) દ્વારા કરવામાં આવે છે.

હવે, ફૉર્મની યથાર્થતા માટેની સરળ જાવાસ્ક્રિપ્ટ લખવાનું શીખીએ. પ્રથમ, આપણે બે ફિલ્ડ સાથેનું એક સરળ ફૉર્મ બનાવીએ : name ફિલ્ડ અને submit બટન.

ફૉર્મની યથાર્થતા માટે, જો ઉપયોગકર્તા name ફિલ્ડને ખાલી રાખી submit બટન પર ક્લિક કરે, તો ઉપયોગકર્તા સમક્ષ નામ ઉમેરવા માટેનો સંદેશ પ્રદર્શિત થવો જોઈએ. આ ક્રિયા માટે નીચે જણાવેલ પગલાંને અનુસરો :

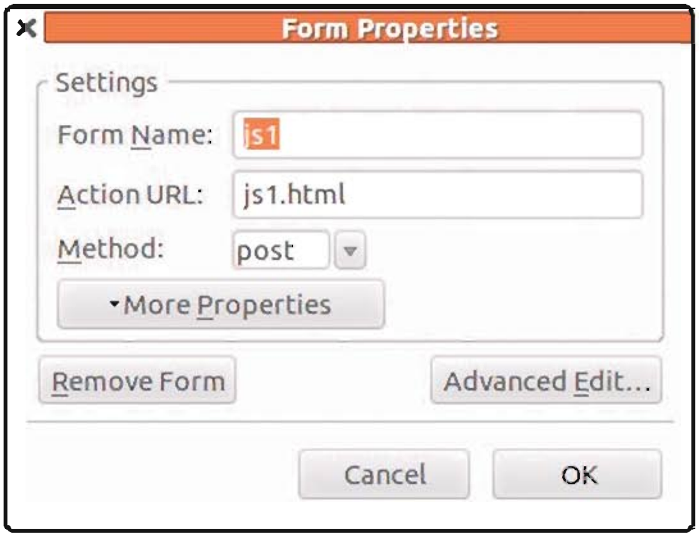
- નવી ફાઇલ બનાવી તેનો સંગ્રહ કરો. આ ઉદાહરણમાં આપણે ફાઇલને js1.html નામ આપી સંગ્રહ કર્યો છે.
- **Form → Define Form** વિકલ્પનો ઉપયોગ કરી નવા ફૉર્મની રચના કરો. ફૉર્મને નામ આપો. આપણા ઉદાહરણમાં ફૉર્મને js1 નામ આપ્યું છે. આકૃતિ 2.15માં દર્શાવ્યા મુજબ નામ માટેનું એક લેબલ અને એક ઇનપુટ ટૅક્સ્ટફિલ્ડ ઉમેરો. એક સબમિટ બટન પણ ઉમેરો.

  નોંધ : ફૉર્મ બનાવતી વખતે આપવામાં આવેલ ફૉર્મ, ઇનપુટ ફિલ્ડ અને કોઈ પણ અન્ય ફિલ્ડનાં નામ સરળ અને યાદ રહે તેવાં હોવાં જોઈએ, કારણકે સ્ક્રિપ્ટમાં વારંવાર તેનો સંદર્ભ આપવાની જરૂર પડશે.



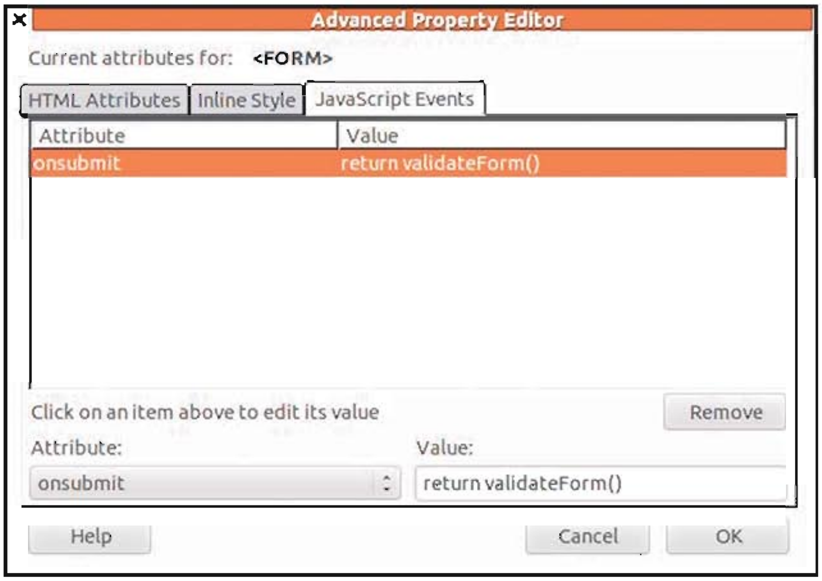
આકૃતિ 2.15 : નમૂનારૂપ ફૉર્મ

- **Form → Define Form** પસંદ કરો. તેથી આકૃતિ 2.16માં દર્શાવવામાં આવેલ Form Properties ડાયલોગબૉક્સ ખૂલશે.



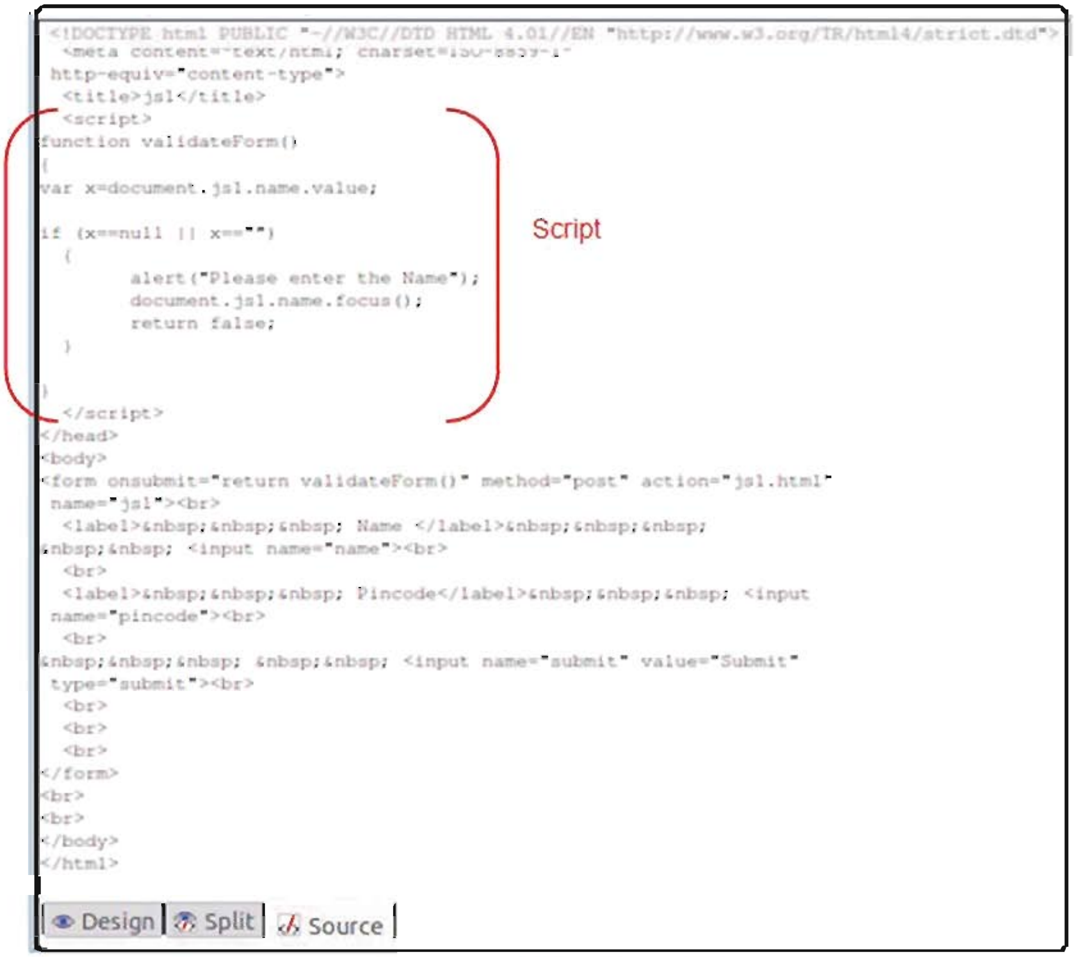
આકૃતિ 2.16 : Form Properties ડાયલોગબૉક્સ

- Advanced Edit બટન પર ક્લિક કરો. તેનાથી આકૃતિ 2.17માં દર્શાવેલ Advanced Property Editor ડાયલોગબૉક્સ ખૂલશે. JavaScript Events વિભાગ પસંદ કરો. આકૃતિ 2.17માં દર્શાવ્યા પ્રમાણે Attribute વિકલ્પમાં "onsubmit" વિકલ્પ પસંદ કરો અને Valueમાં "return validateForm()" લખો. OK બટન પર ક્લિક કરો.



આકૃતિ 2.17 : Advanced Property Editor ડાયલોગબૉક્સ

- હવે, સૉર્સવ્યૂ ખોલો. આકૃતિ 2.18માં દર્શાવ્યા મુજબ <head> ટૅગની અંદર સ્ક્રિપ્ટ ઉમેરો.



આકૃતિ 2.18 : સૉર્સવ્યૂમાં જાવાસ્ક્રિપ્ટ

આકૃતિ 2.8માં દર્શાવેલ સ્ક્રિપ્ટમાં નીચેના ઘટકો વ્યાખ્યાયિત કરવામાં આવ્યા છે :

- validateForm() વિધેય
- *x* ચલ
- if શરતી વિધાન
- alert() આંતરપ્રસ્થાપિત વિધેય
- document ઑબ્જેક્ટ
- focus પદ્ધતિ

હવે, દરેક ઘટકની સંક્ષેપમાં ચર્ચા કરીએ.

ValidateForm વિધેય (Function ValidateForm())

નિશ્ચિત કાર્યને પાર પાડતા પુનઃઉપયોગમાં લઈ શકાય તેવા કોડના બ્લૉકને વિધેય (અથવા રૂટિન) કહે છે. વિધેયને function કી-વર્ડ વડે વ્યાખ્યાયિત કરવામાં આવે છે અને કોડના બ્લૉકને છગડિયા કૌંસ "{ }"માં લખવામાં આવે છે. અહીં validateForm() એક વિધેય છે.

જાવાસ્ક્રિપ્ટ alert() જેવાં કેટલાંક આંતરપ્રસ્થાપિત વિધેય પૂરાં પાડે છે. alert() વિધેય લખાણ સ્વીકારે છે અને તેને એલર્ટ બૉક્સમાં દર્શાવે છે.

ઉદાહરણ તરીકે,

```
function hello()
{
  alert("Hello Students");
}
```

જ્યારે "hello()" વિધેય બોલાવવામાં આવે (called), ત્યારે Hello Students સંદેશ સાથેનું એલર્ટ બૉક્સ દર્શાવવામાં આવશે.

વિધેયનો અમલ કોઈ ઘટના (event) દ્વારા અથવા સૉર્સકોડમાંથી બોલાવવામાં આવે (called) ત્યારે કરવામાં આવે છે. વિધેયને સૉર્સકોડમાં કોઈ પણ સ્થાનેથી બોલાવી શકાય છે. HTML કોડના head કે body વિભાગમાં વિધેયને વ્યાખ્યાયિત કરી શકાય છે. વિધેયને તેનાં નામથી બોલાવવામાં આવે છે. તે કૌંસમાં આપેલી કિંમતો સ્વીકારી શકે છે. આ કિંમતો વિધેયને પ્રાચલ (parameter) તરીકે મોકલવામાં આવે છે. વિધેય કિંમત પરત કરે, તેવું પણ આપણે ક્યારેક ઇચ્છતા હોઈએ. return વિધાનની મદદથી આ શક્ય બને છે. return વિધાનનો ઉપયોગ વિધેયનો અમલ અટકાવી કોઈ નિશ્ચિત કિંમત પરત કરે છે.

આપણા ઉદાહરણમાં validateForm( ) વિધેયને સબમિટ બટન પર ક્લિક આપવાથી ઉદ્ભવતી ઘટના દ્વારા બોલાવવામાં (call) આવે છે. આમ, જ્યારે ઉપયોગકર્તા સબમિટ બટન પર ક્લિક કરશે, જેને ઘટના (event) તરીકે ઓળખવામાં આવે છે, ત્યારે વિધેયને બોલાવી તેમાં આવેલાં વિધાનોનો અમલ કરવામાં આવશે. વિધેય ખોટી (false) કિંમત પરત કરશે.

ઘટના (Events)

ઉપયોગકર્તા દ્વારા કરવામાં આવતી પ્રક્રિયાનો પ્રતિભાવ આપે તેવા સંવાદિત વેબપેજની રચના કરવા માટે જાવાસ્ક્રિપ્ટ ઉપયોગી છે. ઉપયોગકર્તા અને વેબપેજ વચ્ચેના સંવાદન દ્વારા ઘટના (Event) ઉત્પન્ન થાય છે. બીજા શબ્દોમાં કહીએ તો, જ્યારે ઉપયોગકર્તા કંઈ પણ કાર્ય કરે, ત્યારે ઘટના ઉદ્ભવે છે. કોષ્ટક 2.1માં જાવાસ્ક્રિપ્ટ ઘટનાઓની યાદી આપવામાં આવી છે.

ઘટના	સમજૂતી
abort	ચિત્ર દર્શાવવાનું રદ કરવામાં આવે, ત્યારે
blur	રેડિયો-બટન જેવું કોઈ ઘટક નિષ્ક્રિય બને, ત્યારે
click	ઉપયોગકર્તા ફૉર્મના કોઈ ઘટક પર ક્લિક કરે, ત્યારે
change	ઉપયોગકર્તા દ્વારા ફૉર્મના ફિલ્ડની કિંમત બદલવામાં આવે, ત્યારે
error	દસ્તાવેજ કે ચિત્ર દર્શાવતી વખતે ક્ષતિ ઉદ્ભવે, ત્યારે
focus	બટન જેવો કોઈ ઘટક સક્રિય બને, ત્યારે
load	દસ્તાવેજ કે ચિત્ર દર્શાવવામાં આવે, ત્યારે
mouseout	માઉસનું પૉઈન્ટર ઘટક પરથી દૂર કરવામાં આવે, ત્યારે
mouseover	માઉસનું પૉઈન્ટર ઘટક પર લાવવામાં આવે, ત્યારે
reset	ફૉર્મના ફિલ્ડને પૂર્વનિર્ધારિત કિંમતો સાથે રીસેટ કરવામાં આવે, ત્યારે
select	ઉપયોગકર્તા ફૉર્મનું ફિલ્ડ પસંદ કરે, ત્યારે
submit	ઉપયોગકર્તા ફૉર્મ સબમિટ કરે, ત્યારે
unload	ઉપયોગકર્તા પાનું છોડી દે, ત્યારે

કોષ્ટક 2.1 : જાવાસ્ક્રિપ્ટ ઘટનાઓ

જ્યારે કોઈ ઘટના ઉદ્ભવે છે ત્યારે આપેલ સ્થિતિ અનુસાર એક નિશ્ચિત જાવાસ્ક્રિપ્ટ કોડનો અમલ કરવામાં આવે છે. આ જાવાસ્ક્રિપ્ટ કોડને ઘટના-સંચાલક (event handler) કહે છે. ઈવેન્ટ હૅન્ડલરનું નામ ઘટનાનાં નામ જેવું જ રાખવામાં આવે છે. ઉદાહરણ તરીકે, 'ક્લિક' ઘટના માટે ઈવેન્ટ હૅન્ડલરનું નામ onclick() છે. એ જ રીતે 'સબમિટ' ઘટના માટે onsubmit() ઈવેન્ટ હૅન્ડલરનો ઉપયોગ કરવામાં આવશે.

આપણા ઉદાહરણમાં ઉપયોગકર્તા સબમિટ બટન પર ક્લિક કરે ત્યારે, એટલે કે સબમિટ ઘટના વખતે ફૉર્મની યથાર્થતા ચકાસવાની જરૂર છે. આ માટે આપણે onsubmit ઈવેન્ટ હૅન્ડલરનો ઉપયોગ કરવાની જરૂર છે. આકૃતિ 2.18માં onsumbit() ઈવેન્ટ હૅન્ડલર ઉમેરવામાં આવ્યું છે. આ ઈવેન્ટ હૅન્ડલર validateForm વિધેયને બોલાવશે. 'name' ફિલ્ડમાં કોઈ કિંમત છે કે તે ખાલી (null) છે તેના આધારે True કે False પરત કરશે.

ચલ (Variable)

આપણા ઉદાહરણમાં, validateForm() વિધેયમાં નીચેનું વિધાન લખ્યું છે :

```
var x=document.js1.name.value
```

અહીં આપણે x નામનો ચલ ઘોષિત કર્યો છે. ચલ એ વિગતોનો સંગ્રહ કરતો સંગ્રાહક છે. યાદ રાખો કે, જાવાસ્ક્રિપ્ટમાં ચલ પણ કેસ-સૅન્સિટિવ છે. માટે ચલ "x" અને ચલ "X" સમાન નથી. ચલમાં અંકો, સ્ટ્રિંગ કે લખાણનો સંગ્રહ કરી શકાય છે. જાવાસ્ક્રિપ્ટમાં var કી-વર્ડનો ઉપયોગ કરી ચલની ઘોષણા કરી શકાય છે. ઉદાહરણ તરીકે,

```
var x=3;
var y="Hello";
var z="Hello Students";
```

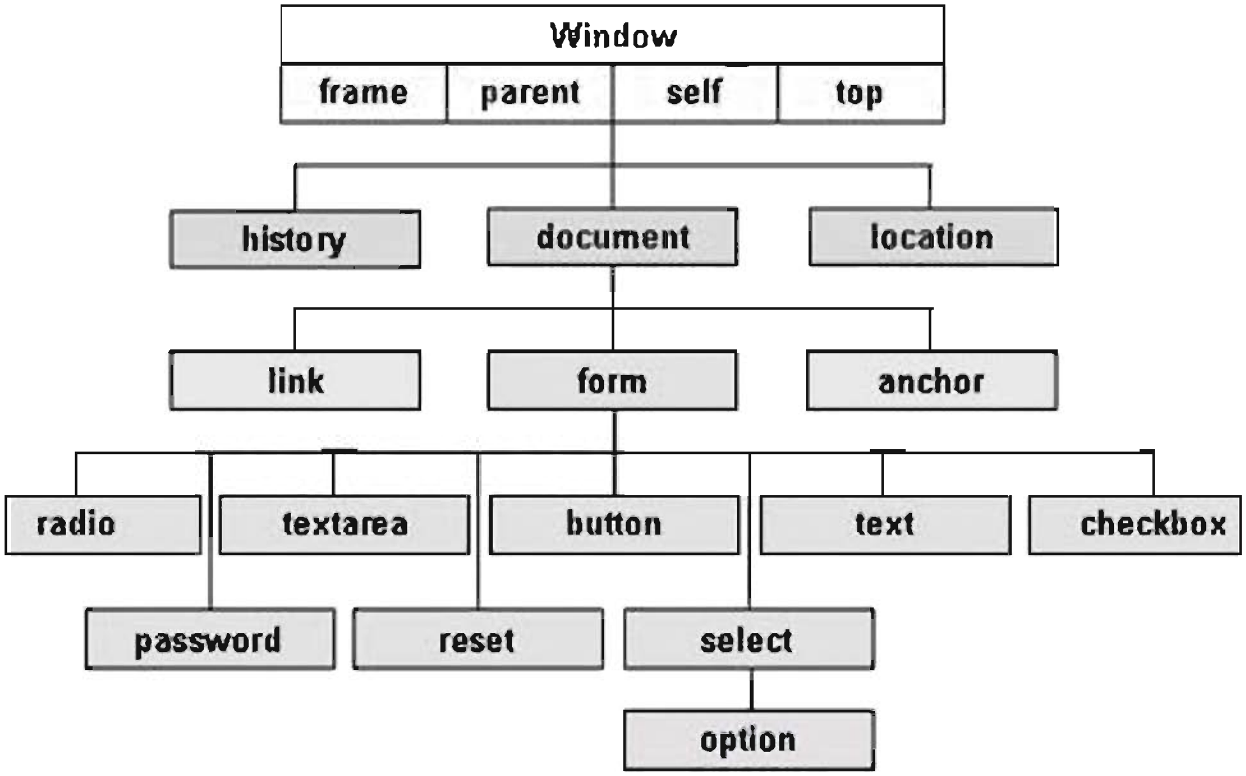
If વિધાન (If Statement)

તમે શરતી વિધાન ifથી સુપરિચિત છો જ. પ્રોગ્રામનો પ્રવાહ બદલવા માટે શરતી વિધાન ifનો ઉપયોગ કરવામાં આવે છે. કોઈ નિશ્ચિત શરતની યથાર્થતા તપાસવા માટે if વિધાન પદાવલીનું મૂલ્યાંકન કરે છે. જો શરત સાચી હોય, તો પ્રોગ્રામ if વિધાનના બ્લૉકમાં પ્રવેશી તેમાં આવેલાં વિધાનોનો અમલ કરે છે. આપણા ઉદાહરણમાં x ચલની કિંમત ખાલી (null) છે કે નહીં તે ચકાસવા માટે if વિધાનનો ઉપયોગ કર્યો છે. જો ચલ xની કિંમત ખાલી હશે, તો if બ્લૉકનો અમલ કરવામાં આવશે.

ડૉક્યુમેન્ટ ઑબ્જેક્ટ મૉડેલ (Document Object Model) - document.js1.name.value

કેટલીક વાર, વેબબ્રાઉઝરના નિયંત્રણ માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, ધારો કે તમારે વેબપેજને બદલવાની કે તેમાં આવેલા ઘટકોનું નિયંત્રણ કરવાની જરૂર છે. વેબ બ્રાઉઝરની વિન્ડો કે વેબપેજના નિયંત્રણ માટે બ્રાઉઝર ઑબ્જેક્ટ મૉડેલ (Browser Object Model - BOM)નો ઉપયોગ કરવામાં આવે છે.

તમામ બ્રાઉઝરને જુદા જુદા વિભાગોમાં અથવા ઘટકોમાં વિભાજિત કરવામાં આવે છે, જેને જાવાસ્ક્રિપ્ટ દ્વારા ઉપયોગમાં લઈ શકાય છે. આ વિભાગોને બ્રાઉઝર ઑબ્જેક્ટ મૉડેલ અથવા BOM તરીકે ઓળખવામાં આવે છે. આકૃતિ 2.19માં દર્શાવ્યા મુજબ આ બ્રાઉઝરના પદાનુક્રમમાં સૌથી ઉપર વિન્ડો ઑબ્જેક્ટ હોય છે.



આકૃતિ 2.19 : બ્રાઉઝર ઑબ્જેક્ટનો પદાનુક્રમ

અહીં ટૂલબાર, મેનુ, સ્ટેટસબાર, વેબપેજ અને અન્ય ઘણા ઘટકો સાથેનો સમગ્ર બ્રાઉઝરનો પદાનુક્રમ રજૂ કરેલો છે. બ્રાઉઝરમાં દેખાતી વિન્ડો કે તેમાં આવેલા ઘટકોને બદલવા માટે બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં આવેલા ગુણધર્મો (properties) અને પદ્ધતિઓ (methods)નો ઉપયોગ કરવામાં આવે છે. બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં ઑબ્જેક્ટ બનાવવાની જરૂર પડતી નથી; પરંતુ જ્યારે બ્રાઉઝર વેબપેજ દર્શાવે, ત્યારે તેની રચના આપોઆપ થતી હોય છે.

બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી ઉપરના સ્તરનો ઑબ્જેક્ટ Window છે. Window ઑબ્જેક્ટ બ્રાઉઝરની વિન્ડો અથવા તેમાં આવેલી સ્વતંત્ર ફ્રેમને રજૂ કરે છે. તેની રચના બ્રાઉઝર દ્વારા આપોઆપ કરવામાં આવે છે. Window ઑબ્જેક્ટ વૈશ્વિક (global) ઑબ્જેક્ટ ગણાય છે. કારણકે, મૉડેલના અન્ય તમામ બ્રાઉઝર ઑબ્જેક્ટનો તેની અંદર સમાવેશ કરવામાં આવે છે. ઉદાહરણ તરીકે, Window ઑબ્જેક્ટમાં document ઑબ્જેક્ટનો સમાવેશ થાય છે. Window ઑબ્જેક્ટની પદ્ધતિઓ (methods) અને ગુણધર્મો (properties) દ્વારા વેબબ્રાઉઝરની વિન્ડોને નિયંત્રિત કરવામાં આવે છે, જ્યારે document ઑબ્જેક્ટની પદ્ધતિઓ (methods) અને ગુણધર્મો (properties) દ્વારા વેબપેજને નિયંત્રિત કરવામાં આવે છે.

બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી અગત્યનો ઑબ્જેક્ટ document ઑબ્જેક્ટ છે. બ્રાઉઝરમાં વેબપેજ રજૂ કરવા માટે તેનો ઉપયોગ કરવામાં આવે છે. વેબપેજના અન્ય તમામ ઘટકો જેવા કે, ફૉર્મ, ચિત્રો, લિંક અને અન્ય ઘટકો document ઑબ્જેક્ટની અંદર સમાવવામાં આવ્યા છે. ઉદાહરણ તરીકે, <form> ઘટક દ્વારા રચવામાં આવેલો form ઑબ્જેક્ટ જાવાસ્ક્રિપ્ટ દ્વારા document ઑબ્જેક્ટની અંદર રજૂ કરવામાં આવે છે. document ઑબ્જેક્ટમાં form ઑબ્જેક્ટનો સમાવેશ કરવામાં આવ્યો છે. તેવી જ રીતે form ઑબ્જેક્ટમાં element ઑબ્જેક્ટને સમાવવામાં આવ્યો છે. ફૉર્મના દરેક ઘટકનો સંદર્ભ મેળવવા element ઑબ્જેક્ટનો ઉપયોગ કરવામાં આવે છે. element ઑબ્જેક્ટ તરીકે રેડિયો-બટન, ટેક્સ્ટબૉક્સ, ચેકબૉક્સ કે અન્ય કોઈ પણ ઘટક હોઈ શકે છે.

focus પદ્ધતિ (focus Method)

સ્ક્રિપ્ટમાં focus નામની પદ્ધતિનો પણ ઉપયોગ કરવામાં આવે છે. ફૉર્મના નિશ્ચિત ઘટક પર નિયંત્રણ લઈ જવા તેને લાગુ પાડવામાં આવે છે. આમ, ઉપયોગકર્તા એલર્ટ સંદેશ મેળવે પછી, focus પદ્ધતિની મદદથી ઘટકની અંદર કર્સર ગોઠવવામાં આવે છે અને તે નિશ્ચિત ઘટક પ્રકાશિત (highlighted) બને છે. અહીં નિયંત્રણને name ઘટક પર લઈ જવામાં આવ્યું છે.

હવે, ફાઇલને બ્રાઉઝરમાં ખોલીએ. આકૃતિ 2.20માં પરિણામ દર્શાવ્યું છે.

આકૃતિ 2.20 : બ્રાઉઝરમાં પરિણામ

name ફિલ્ડને ખાલી રાખી submit બટન પર ક્લિક કરો. આકૃતિ 2.21માં દર્શાવ્યા પ્રમાણે એક એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

આકૃતિ 2.21 : ખાલી name ફિલ્ડ માટેનો એલર્ટ સંદેશ

alertનો સંદેશ એ સૉર્સવ્યૂમાં લખવામાં આવેલ સ્ક્રિપ્ટનું પરિણામ છે. આ એવી યથાર્થતા છે, જે ફૉર્મને જાવાસ્ક્રિપ્ટ દ્વારા

પૂરી પાડવામાં આવી છે, આમ, અનેક ફિલ્ડ ધરાવતા ફૉર્મમાં કોઈ પણ ફિલ્ડ ખાલી રાખવામાં આવે, તો તેને ચકાસવા માટે યથાર્થતા ઉમેરી શકાય છે.

ઇન્ટરનેટ પર ફૉર્મ ભરતી વખતે તમે કેટલાંક ફિલ્ડને લાલ રંગની ફૂદડી "*" સાથે દર્શાવેલાં જરૂર જોયાં હશે. તે ઉપયોગકર્તાને નિર્દેશ કરે છે કે આ ફિલ્ડ ફરજિયાત છે અને તેને ખાલી છોડવાં જોઈએ નહીં. જાવાસ્ક્રિપ્ટનો ઉપયોગ કરી આ પ્રકારનાં ફિલ્ડની વિગતો તપાસી શકાય છે તથા ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવી શકાય છે.

હવે, ફૉર્મમાં Pincode નામનું અન્ય ફિલ્ડ ઉમેરીએ. આ ફિલ્ડ પર નીચે જણાવેલ યથાર્થતા લાગુ પાડવામાં આવશે :

- ઉપયોગકર્તા આ ફિલ્ડને ખાલી છોડી શકશે નહીં.
- માત્ર અંક માન્ય હશે. (અક્ષર માન્ય રાખવામાં આવશે નહીં).
- પિનકોડ છ અંકોનો હોવો જોઈએ.

જો કોઈ પણ યથાર્થતાનો ભંગ કરવામાં આવશે તો ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે. આ યથાર્થતા ઉમેરવા નીચે જણાવેલ પગલાંને અનુસરો :

- આ જ ફૉર્મમાં પિનકોડ માટેનું એક લેબલ અને ઇનપુટ ફિલ્ડ ઉમેરો.
- આકૃતિ 2.22માં દર્શાવેલ સ્ક્રિપ્ટ સૉર્સવ્યૂમાં ઉમેરો.

```
<script>
function validateForm()
{
 var x=document.js1.name.value;
 var y=document.js1.pincode.value;
 if (x==null || x=="")
 {
  alert("Please enter the Name");
  document.js1.name.focus();
  return false;
 }

 if (y=="" || isNaN(y) || y.length>6 || y.length<6)
 {
  alert("Please enter the Pincode properly");
  document.js1.pincode.focus();
  return false;
 }
}
</script>
```

આકૃતિ 2.22 : નામ અને પિનકોડની યથાર્થતા ચકાસવા માટેની જાવાસ્ક્રિપ્ટ

સ્ક્રિપ્ટમાં જોઈ શકાય છે તે મુજબ, y નામનો અન્ય એક ચલ ઘોષિત કરવામાં આવે છે. y ચલમાં પિનકોડ ફિલ્ડની કિંમત સમાવવામાં આવી છે. if શરતમાં, isNaN() નામના વિધેયનો ઉપયોગ કર્યો છે. આ વિધેયને સમજીએ.

isNaN()

આંકડાકીય કિંમતો સાથે કાર્ય કરવા માટે જાવાસ્ક્રિપ્ટ આંતરપ્રસ્થાપિત વિધેયોનો ઉપયોગ કરે છે. isNaN() એ સામાન્ય રીતે સૌથી વધુ ઉપયોગમાં લેવાતું આંકડાકીય વિધેય છે. NaN એટલે "Not a Number" આપેલ કિંમત આંકડાકીય પ્રકારની ન હોય, તો તે true પરત કરે છે તથા કિંમત આંકડાકીય હોય, તો તે false પરત કરે છે. ઉદાહરણ તરીકે,

isNaN(123) વિધેય false પરત કરશે, કારણકે "123" એ સંખ્યા છે.

isNaN("hello") વિધેય true પરત કરશે, કારણકે "hello" એ સંખ્યા નથી.

આમ, if વિધાનનો ઉપયોગ કરી તપાસી શકાયું છે કે y ખાલી, અંક ન હોય તેવી અથવા છ અંકોની ન હોય તેવી કિંમત હોવી જોઈએ. જો આમાંથી એક પણ શરતનો ભંગ થશે, તો ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે.

બ્રાઉઝરમાં ફાઇલ ખોલો. આકૃતિ 2.23માં દર્શાવ્યા મુજબનું પરિણામ જોવા મળશે. name ફિલ્ડમાં નામ ઉમેરો. (name ફિલ્ડને ખાલી છોડશો નહીં, નહીં તો તેના માટેનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.)

આકૃતિ 2.23 : બ્રાઉઝરમાં પરિણામ

Pincode ફિલ્ડને ખાલી રાખી Submit બટન પર ક્લિક કરો. આકૃતિ 2.24માં દર્શાવ્યા મુજબનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.

આકૃતિ 2.24 : ખાલી ફિલ્ડ માટેનો એલર્ટ સંદેશ

હવે, Pincode ફિલ્ડમાં થોડા અક્ષર કે છ અંકોથી વધુ કોઈ આંકડાકીય સંખ્યા ઉમેરો. આમ કરવાથી આકૃતિ 2.24માં દર્શાવેલ આ જ પ્રકારનો એલર્ટ સંદેશ દર્શાવવામાં આવશે.

અહીં, Pincode ફિલ્ડમાં ત્રણે યથાર્થતા માટે આપણે એકસમાન એલર્ટ સંદેશ દર્શાવીએ છીએ. જો યથાર્થતા પ્રમાણે યોગ્ય એલર્ટ સંદેશ દર્શાવવા ઇચ્છતા હોઈએ, તો સ્ક્રિપ્ટમાં સુધારા કરવાની જરૂર પડશે. ઉદાહરણમાં ત્રણે યથાર્થતા માટે એક જ if શરતનો ઉપયોગ કરવામાં આવ્યો છે, જે એલર્ટ સંદેશ દર્શાવે છે. યથાર્થતા મુજબ એલર્ટ સંદેશ દર્શાવવા માટે, ત્રણ if શરતો ઉમેરવી પડે. ત્રણ if શરતો સાથેની સ્ક્રિપ્ટ આકૃતિ 2.25માં દર્શાવી છે.

```
<script>
function validateForm()
{
 var x=document.js1.name.value;
 var y=document.js1.pincode.value;
 if (x==null || x=="")
 {
  alert("Please enter the Name");
  document.js1.name.focus();
  return false;
 }
if (y=="")
 {
  alert("please enter the pincode properly");
  document.js1.pincode.focus();
  return false;
 }
if(isNaN(y))
{
 alert("please enter a number");
 document.js1.pincode.focus();
 return false;
}
if(y.length>6 || y.length<6)
{
 alert("please enter a six digits");
 document.js1.pincode.focus();
 return false;
}
}
</script>
```

આકૃતિ 2.25 : યોગ્ય સંદેશ દર્શાવવા માટેનો જાવાસ્ક્રિપ્ટ કોડ

ફાઇલને બ્રાઉઝરમાં ખોલો. Pincode ફિલ્ડમાં કેટલાક અક્ષરો ઉમેરીને સબમિટ બટન પર ક્લિક કરો. આકૃતિ 2.26માં દર્શાવ્યા મુજબ એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

આકૃતિ 2.26 : આંકડાકીય લખાણ તપાસવા માટેનો એલર્ટ સંદેશ

Pincode ફિલ્ડમાં 6 અંકોથી નાની સંખ્યા ઉમેરો અને સબમિટ બટન પર ક્લિક કરો. આકૃતિ 2.27માં દર્શાવ્યા મુજબનો એલર્ટ સંદેશ રજૂ કરવામાં આવશે.

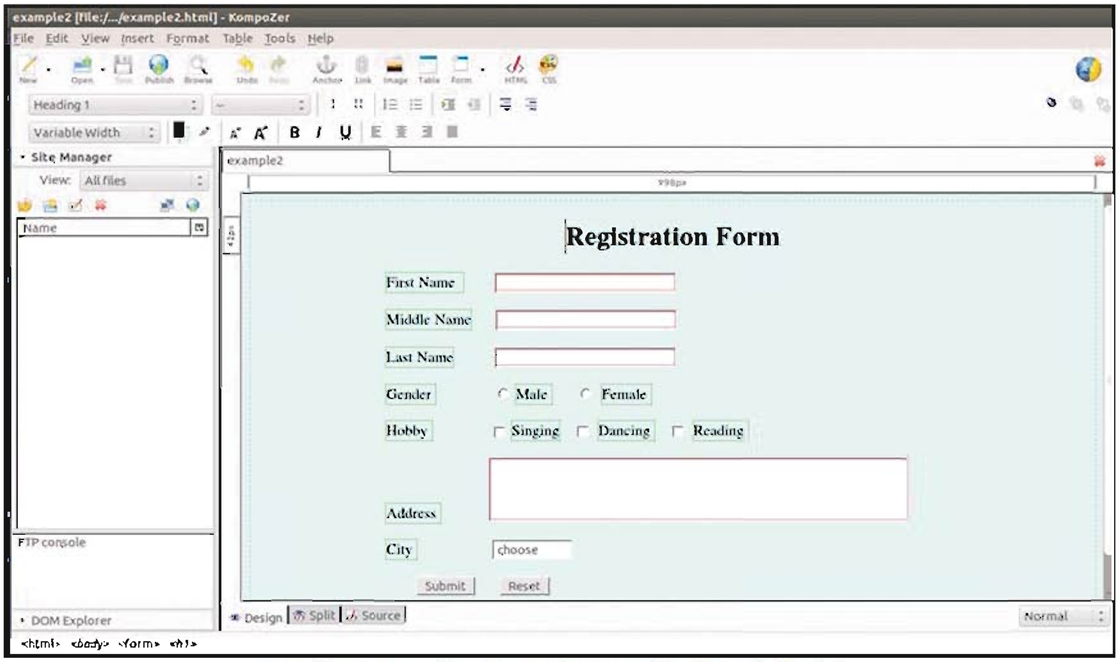
આકૃતિ 2.27 : લંબાઈ તપાસવા માટેનો એલર્ટ સંદેશ

વેબપેજમાં સંવાદન ઉમેરવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ આપણે શીખ્યા. હવે, પ્રકરણ-1માં બનાવવામાં આવેલા નોંધણી માટેના ફોર્મ (Registration Form)માં જાવાસ્ક્રિપ્ટ ઉમેરીએ.

યથાર્થતા લાગુ પાડવા માટે ફોર્મમાં નીચેની યાદીમાં જણાવેલ સરળ ફેરફાર કરીશું :

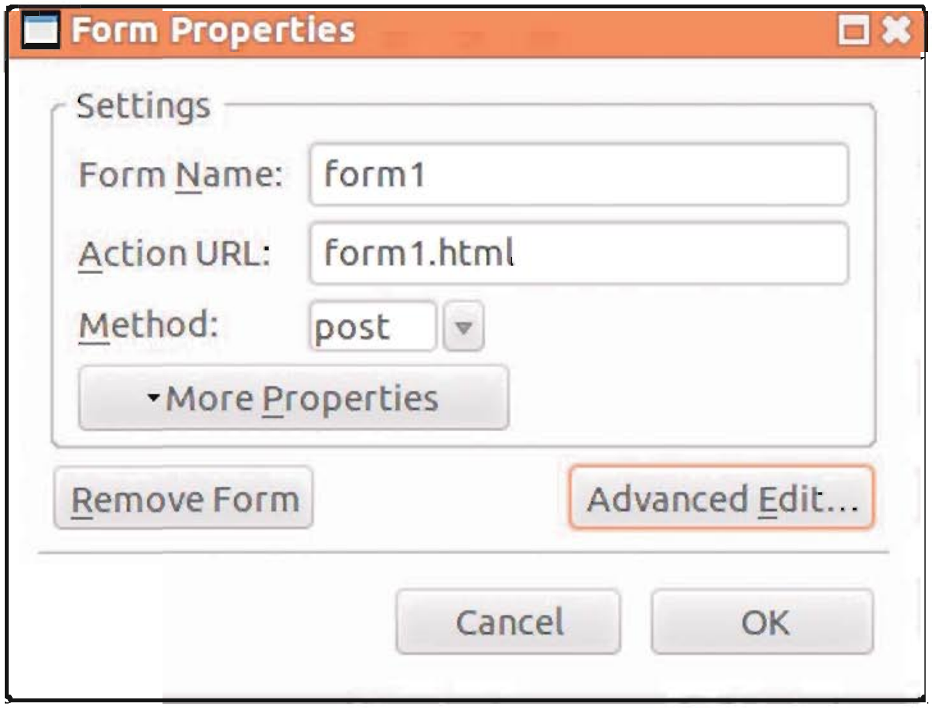
- Gender અને Hobby ફિલ્ડ શરૂઆતમાં પસંદ થયેલાં નહીં હોય.
- Address ફિલ્ડમાં શરૂઆતની કોઈ કિંમત નહીં હોય.
- City ફિલ્ડમાં "choose the city" વિકલ્પ હશે, જેની કિંમત -1 છે. અન્ય તમામ શહેરો, જેવાં કે Ahmedabad, Rajkot અને Suratને અનુક્રમે 1, 2 અને 3 કિંમતો હશે.

ઉપર્યુક્ત ફેરફારો કરવા માટે, ફોર્મ ખોલી તેમાં સંબંધિત ફિલ્ડ પસંદ કરો. **Form → Form Field** વિકલ્પ પસંદ કરો. આમ કરવાથી જે-તે ફિલ્ડમાં ફેરફાર કરવા માટે તેના ગુણધર્મ દર્શાવવામાં આવશે. ફેરફાર કર્યા બાદ ફોર્મ આકૃતિ 2.28માં આપેલ ફોર્મ જેવું દેખાશે.



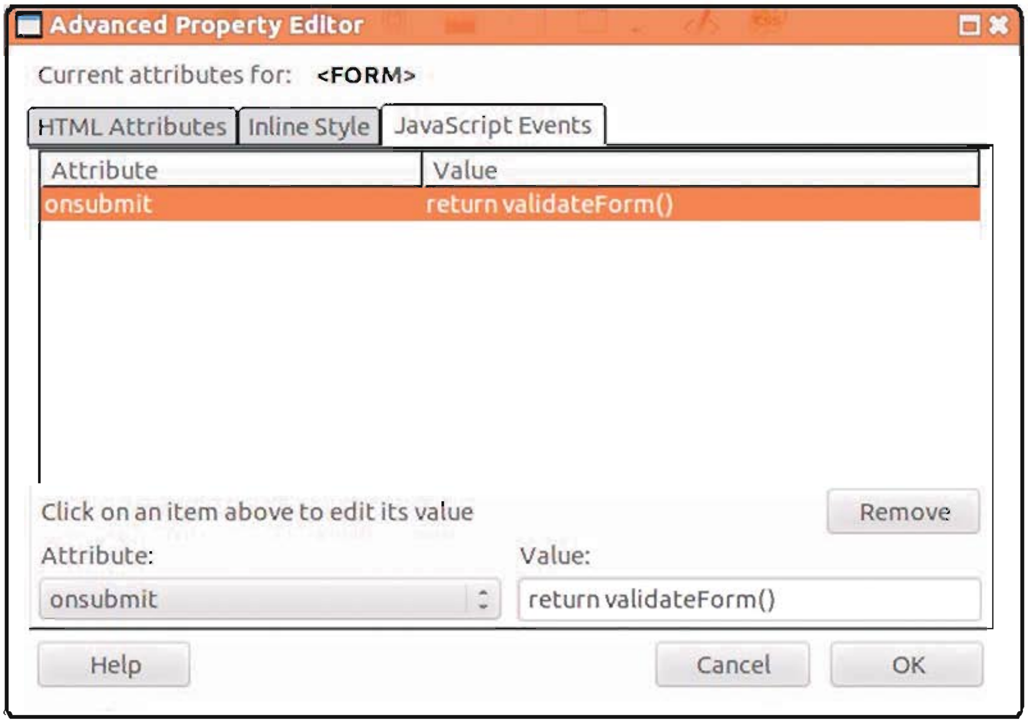
આકૃતિ 2.28 : ફેરફાર કર્યા બાદ નોંધણી માટેનું ફોર્મ

- **Form → Define Form** વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે, ફોર્મમાં રાઇટ ક્લિક કરીને Properties વિકલ્પ પસંદ કરી શકાય. તે આકૃતિ 2.29માં દર્શાવેલ Form Properties ડાયલોગબૉક્સ રજૂ કરશે.



આકૃતિ 2.29 : Form Properties ડાયલોગબૉક્સ

- Advanced Edit બટન પર ક્લિક કરો. આમ કરવાથી આકૃતિ 2.30માં આપેલ Advanced Property Editor ડાયલોગબૉક્સ ખૂલશે. JavaScript Events વિભાગ પસંદ કરો. Attribute વિકલ્પમાં "onsubmit" અને Value વિકલ્પમાં "return validateForm() ઉમેરો. OK બટન પર ક્લિક કરો.



આકૃતિ 2.30 : Advanced Property Editor ડાયલોગબૉક્સ

- આકૃતિ 2.31માં દર્શાવેલ સ્ક્રિપ્ટ HTMLના <head> વિભાગમાં ઉમેરો.

```
<script>
function validateForm()
{
var x=document.form1.firstname.value;
var y=document.form1.middlename.value;
var z=document.form1.lname.value;
var r=document.form1.address.value;
if (x==null || x=="")
{
alert("Please enter the first name properly");
document.form1.firstname.focus();
return false;
}

if (y==""||y==null)
{
alert("Please enter the middle name properly");
document.form1.middlename.focus();
return false;
}
if(z==""||z==null)
{
alert("Please enter a last name properly");
document.form1.lname.focus();
return false;
}
if(r=="")
{
alert("Please enter address");
document.form1.address.focus();
return false;
}
if (( document.form1.gender[0].checked == false ) && ( document.form1.gender[1].checked ==
false ) )
{
alert ( "Please choose your Gender: Male or Female" );
document.form1.gender[0].focus();
return false;
}
if ((document.form1.hobby[0].checked == false) && (document.form1.hobby[1].checked ==
false ) && (document.form1.hobby[2].checked == false))
{
alert ( "Please choose a hobby" );
document.form1.hobby[0].focus();
return false;
}
if( document.form1.city.value == "-1" )
{
alert( "Please provide your city!" );
document.form1.city.focus();
return false;
}
}
</script>
```

આકૃતિ 2.31 : જાવાસ્ક્રિપ્ટની યથાર્થતા

સ્ક્રિપ્ટમાં દર્શાવ્યા મુજબ x, y, z અને r એમ ચાર ચલ ઘોષિત કરવામાં આવ્યા છે.

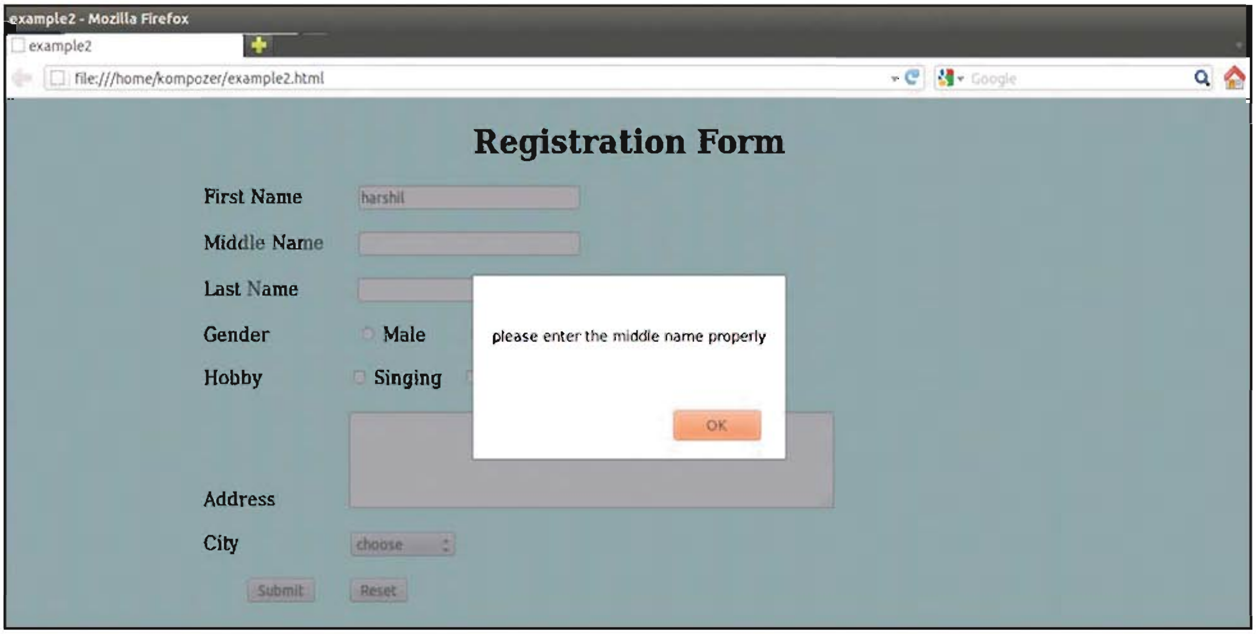
var x=document.form1.firstname.value વિધાનમાં form1 એ ફૉર્મના નામનો સંદર્ભ આપે છે, firstname પદ ઘટકને આપેલ નામનો સંદર્ભ આપે છે. ("first name" ઇનપુટ ફિલ્ડને આપવામાં આવેલ નામ). આમ,

- x ચલમાં first nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- y ચલમાં middle nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- z ચલમાં last nameની કિંમતનો સંગ્રહ કરવામાં આવશે.
- r ચલમાં addressની કિંમતનો સંગ્રહ કરવામાં આવશે.

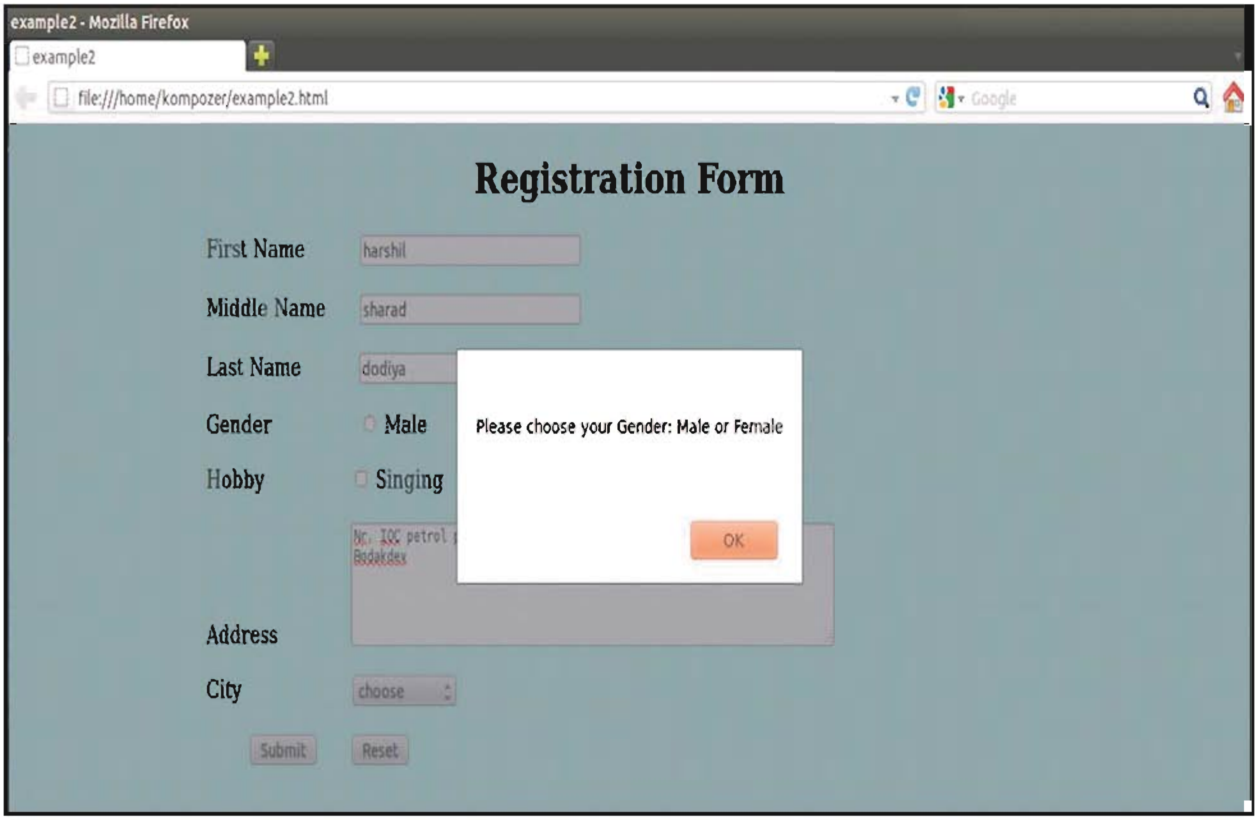
સ્ક્રિપ્ટમાં, if શરતનો ઉપયોગ કરી,

- first name ખાલી છે કે નહીં તે ચલ x દ્વારા ચકાસવામાં આવશે.
- middle name ખાલી છે કે નહીં તે ચલ y દ્વારા ચકાસવામાં આવશે.
- last name ખાલી છે કે નહીં તે ચલ z દ્વારા ચકાસવામાં આવશે.
- address ખાલી છે કે નહીં તે ચલ r દ્વારા ચકાસવામાં આવશે.

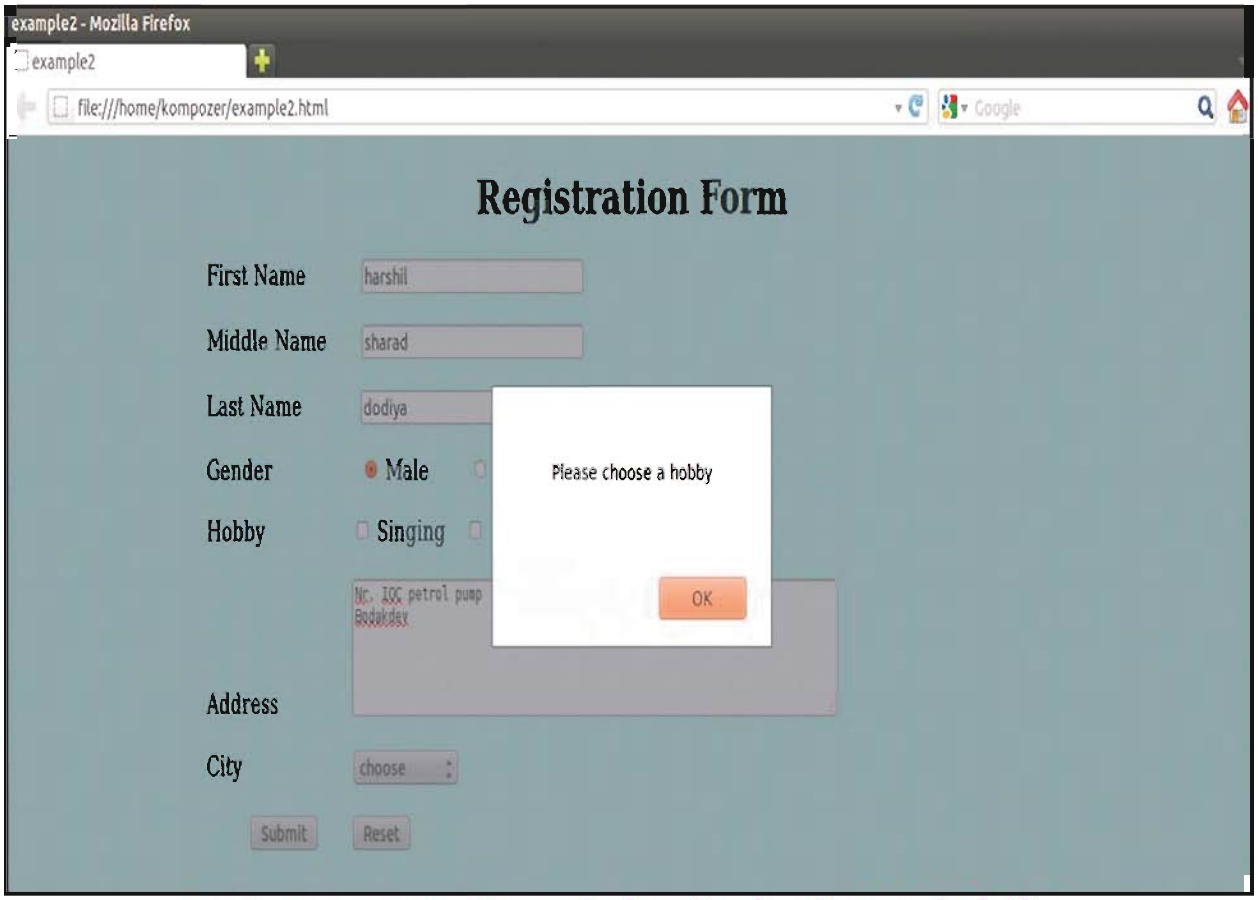
જાતિ (gender) ફિલ્ડ માટેનાં રેડિયો-બટનને "gender" નામ દ્વારા જૂથમાં મૂકવામાં આવ્યાં છે. gender બે ઘટક ધરાવતો ઍરે છે. document.form1.gender[0].checkedનો ઉપયોગ કરી gender ઍરેના પ્રથમ ઘટકની ચકાસણી if વિધાન દ્વારા કરવામાં આવે છે તથા document.form1.gender[1].checkedનો ઉપયોગ કરી if વિધાન દ્વારા ઍરેના બીજા ઘટકની ચકાસણી કરવામાં આવે છે. જો આ બન્નેનું પરિણામ false મળે તો ઉપયોગકર્તાએ કોઈ વિકલ્પ પસંદ કરેલ નથી. આથી એલર્ટ સંદેશ દર્શાવવામાં આવશે અને રેડિયો-બટન પર નિયંત્રણ (focus) લઈ જવામાં આવશે.
આવી જ રીતે, શોખ ફિલ્ડ માટેનાં ચેકબૉક્સને hobby નામ દ્વારા જૂથમાં મૂકવામાં આવ્યાં છે. hobby ત્રણ ઘટક ધરાવતો ઍરે છે. જેવી રીતે if શરત દ્વારા gender ફિલ્ડની કિંમત તપાસવામાં આવી, તેવી જ રીતે hobby ફિલ્ડની કિંમત ચકાસવામાં આવશે. અને જો તમામ ત્રણ ઘટક false દર્શાવે, તો ઉપયોગકર્તાએ કોઈ વિકલ્પ પસંદ કર્યો નથી. તેથી, ઉપયોગકર્તા સમક્ષ શોખ પસંદ કરવા અંગેનો એલર્ટ સંદેશ દર્શાવવામાં આવશે અને ચેકબૉક્સ પર નિયંત્રણ (focus) લઈ જવામાં આવશે.
શહેર માટેના ફિલ્ડ માટે -1 કિંમત તપાસવામાં આવશે. (**નોંધ :** -1 કિંમત "choose the city" માટે રાખવામાં આવી છે.), તે દર્શાવે છે કે ઉપયોગકર્તા દ્વારા એક પણ કિંમત પસંદ કરવામાં આવી નથી. ત્યારે ઉપયોગકર્તા સમક્ષ એલર્ટ સંદેશ દર્શાવવામાં આવશે. ફિલ્ડ ખાલી છોડવાથી મળતા કેટલાક એલર્ટ સંદેશાઓ આકૃતિ 2.32થી 2.35માં દર્શાવ્યા છે.



આકૃતિ 2.32 : middle name ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ

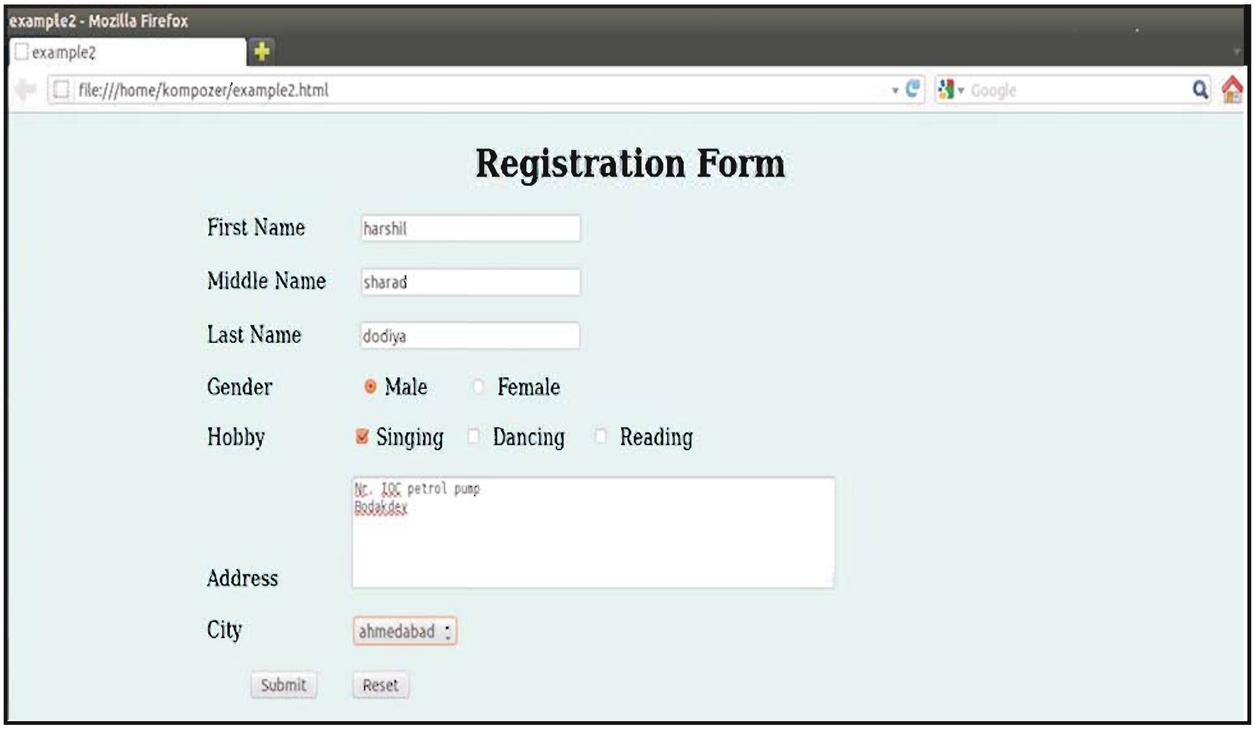


આકૃતિ 2.33 : gender ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ



આકૃતિ 2.34 : hobby ફિલ્ડ ખાલી છોડવામાં આવે ત્યારે મળતા એલર્ટ સંદેશ

તમામ ફિલ્ડની કિંમતો ઉમેર્યા પછી ફૉર્મનો અંતિમ દેખાવ આકૃતિ 2.35માં દર્શાવ્યો છે.



આકૃતિ 2.35 : અંતિમ ફૉર્મ

ફૉર્મની યથાર્થતા માટે તથા વેબપેજને વધુ સંવાદિત બનાવવા માટે આપણે જાવાસ્ક્રિપ્ટનો અભ્યાસ કર્યો.

સારાંશ

આ પ્રકરણમાં આપણે કેસ્કેડિંગ સ્ટાઇલશીટ અને જાવાસ્ક્રિપ્ટ વિશે અભ્યાસ કર્યો. વેબસાઇટના દર્શનીય ઘટકો માટેની શૈલી સ્પષ્ટ કરવા CSSનો ઉપયોગ કરવામાં આવે છે. તેની મદદથી દસ્તાવેજની માહિતીને તેના દેખાવ સંબંધી વિગતોથી અલગ રાખવામાં આવે છે. આપણે CSSના ફાયદા અને ગેરફાયદા વિશે ચર્ચા કરી. વેબપેજમાં તાર્કિક પાસું ઉમેરવા માટે જાવાસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. સ્ક્રિપ્ટિંગ ભાષા એક સરળ, હળવી પ્રોગ્રામિંગ ભાષા છે, જેમાં અદ્યતન પ્રોગ્રામિંગ કાર્યનો સમાવેશ કરવામાં આવતો નથી. વેબપેજની રૂપરેખા સુધારવા તથા ફૉર્મની યથાર્થતા ચકાસવા માટે તેનો ઉપયોગ કરવામાં આવે છે. તે HTML પાનાંમાં સંવાદન ઉમેરે છે તથા તેને પ્રત્યક્ષ રીતે HTML કોડમાં ઉમેરવામાં આવે છે. HTML પાનામાં જાવાસ્ક્રિપ્ટ ઉમેરવા માટે <script>... </script> ટૅગનો ઉપયોગ કરવામાં આવે છે.

સ્વાધ્યાય

1. કેસ્કેડિંગ સ્ટાઇલશીટનો ઉદ્દેશ જણાવો.
2. વેબપેજમાં શૈલી અને તેમાં આવેલ વિગતોને શા માટે અલગ રાખવી જોઈએ ?
3. CSSની વાક્યરચના સમજાવો.
4. CSSના ફાયદાઓ જણાવો.
5. CSSના ગેરફાયદાઓ જણાવો.
6. HTML પાનામાં જાવાસ્ક્રિપ્ટનો ઉપયોગ શા માટે કરવામાં આવે છે ?

7. HTML પાનામાં જાવાસ્ક્રિપ્ટને કેવી રીતે ઓળખી શકાય ? ઉદાહરણ આપો.

8. આપેલ વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરો :

(1) નીચેનામાંથી કોનો ઉપયોગ વેબસાઇટમાં દર્શનીય ઘટકો માટેની શૈલી સ્પષ્ટ કરવા માટે થાય છે ?

(a) Cascading Style Sheets (b) Webpage

(c) Form (d) Animation

(2) નીચેનામાંથી કોને CSSની વાક્યરચનામાં એક વિશિષ્ટ નિશાની તરીકે ઓળખવામાં આવે છે ?

(a) નિયમ (Rules) (b) પસંદગીકાર (Selector)

(c) ઘોષણા (Declaration) (d) નિવેશ (Input)

(3) નીચેનામાંથી CSS નિયમના મુખ્ય બે વિભાગ ક્યા છે ?

(a) Selector, declaration (b) Select, declaration

(c) Selector, declare (d) Selection, declaration

(4) જેના પર શૈલી લાગુ પાડવાની છે, તેવા HTMLના ઘટકને શું કહે છે ?

(a) declaration (b) selector

(c) select (d) declare

(5) નીચેનામાંથી CSSની વાક્યરચના કઈ છે ?

(a) select {property : value} (b) selector {value : property}

(c) selector {property : value} (d) selection {property : value}

(6) નીચેનામાંથી કઈ સંસ્થાએ જાવાસ્ક્રિપ્ટનો વિકાસ કર્યો ?

(a) Yahoo (b) Google

(c) Wikipedia (d) Netscape

(7) નીચેનામાંથી કઈ સ્ક્રિપ્ટિંગ ભાષાનો ઉપયોગ વેબપેજમાં પ્રોગ્રામિંગ ઉમેરવા માટે કરવામાં આવે છે ?

(a) Action script (b) JavaScript

(c) HTML (d) CSS

(8) નીચેનામાંથી કઈ સરળ, હળવી પ્રોગ્રામિંગ ભાષાને સ્ક્રિપ્ટિંગ ભાષા તરીકે ઓળખવામાં આવે છે કે જેમાં અદ્યતન પ્રોગ્રામિંગ-કાર્યોનો સમાવેશ કરવામાં આવ્યો ન હોય ?

(a) JavaScript (b) HTML

(c) C (d) Java

(9) નીચેનામાંથી ક્યા ટેગનો ઉપયોગ HTML પાનાંમાં જાવાસ્ક્રિપ્ટનો કોડ ઉમેરવા માટે કરવામાં આવે છે ?

(a) <script>... <script> (b) <script>... </script>

(c) <script>... <//script> (d) </script>... </script>

(10) નીચેનામાંથી કઈ નિશાની દ્વારા જાવાસ્ક્રિપ્ટ બ્લૉકની શરૂઆત અને અંત દર્શાવવામાં આવે છે ?

(a) અર્ધવિરામ (semicolon) (b) ચોરસ કૌંસ (square bracket)

(c) છગડિયા કૌંસ (curly bracket) (d) સાદા કૌંસ (round bracket)

(11) નિશ્ચિત કાર્ય કરવા માટે પુનઃઉપયોગમાં લઈ શકાય તે પ્રકારના ઘટકને શું કહે છે ?

(a) Array (b) Code

(c) Program (d) Function

(12) નીચેનામાંથી ક્યા વિધાનનો ઉપયોગ વિધેયમાં કિંમત પરત કરવા માટે થાય છે ?

(a) return (b) function

(c) select (d) send

(13) ઉપયોગકર્તા અને વેબપેજ વચ્ચેના સંવાદનથી શું બને છે ?

(a) વિધેય (Function) (b) પ્રતિભાવ (Response)

(c) ઘટના (Event) (d) કિંમત (Value)

(14) નીચેનામાંથી કોને ઘટના કહી શકાય નહીં ?

(a) Abort (b) Mouseover

(c) Set (d) Load

(15) વિગતોનો સંગ્રહ કરવા માટેના સંગ્રાહકને શું કહે છે ?

(a) ચલ (Variable) (b) પૂર્ણાંક (Integer)

(c) ઘટના (Event) (d) ઘટના-સંચાલક (Event handler)

(16) BOM એટલે શું ?

(a) Browser of Model (b) Browser Object Model

(c) Browser Object Modelling (d) Browse Object Model

(17) બ્રાઉઝર ઑબ્જેક્ટ મૉડેલમાં સૌથી ઉપરના સ્તરે રહેલ ઑબ્જેક્ટ કયો છે ?

(a) Window (b) Document

(c) Page (d) Location

(18) NaN એટલે શું ?

(a) Not a Numeric (b) Not a Number

(c) Not a Noun (d) Not an Numeric

પ્રાયોગિક સ્વાધ્યાય

1. નીચે આપેલ નિયમોના ઉપયોગ દ્વારા H1 ટેગ માટે CSSની રચના કરી તેને લખાણ પર લાગુ કરો :

- Font : Times New Roman
- Color : Red
- Case : Lowercase
- Fontstyle : Italic
- Alignment : Centre
- Text decoration : Underline
- Background color : Light Grey
- Border : Dotted

2. આપેલ ઘટકો સાથેનું ફૉર્મ બનાવો : name, email address, phone number અને submit button. તેમાં નીચે દર્શાવ્યા મુજબ યથાર્થતા ઉમેરો :

- ફિલ્ડ ખાલી ન હોવાં જોઈએ.
- ફોન માટેના ફિલ્ડમાં માત્ર અંક માન્ય છે.
- ફોનનંબર દસ અંકોનો હોવો જોઈએ.

3. વિદ્યાર્થીની અંગત વિગતો ઉમેરવા માટેના પ્રકરણ-1માં બનાવવામાં આવેલ ફૉર્મમાં યથાર્થતા ઉમેરો.

4. પ્રતિસાદ (feedback) માટેના પ્રકરણ-1માં બનાવવામાં આવેલ ફૉર્મમાં યથાર્થતા ઉમેરો.