CSS - somo la 1: Maana ya CSS, kazi zake na historia yake

Katika somo hili utakwenda kujifunza maana ya CSS, pia nitakujulisha kazi zake. Mwisho utatambuwa historia ya CSS toka kuanzishwa.

CSS ni kifupi cha "Cascading Style Sheets" (Inatafsiriwa kwa Kiswahili kama "Vitambulisho vya Mtindo wa Kupangilia"). Ni lugha ya programu inayotumiwa kwa kawaida katika maendeleo ya wavuti ili kuunda na kuweka mtindo na muundo wa kurasa za wavuti. CSS inaruhusu wavuti kuonekana vizuri zaidi na kuwa na muundo unaofaa kwa kutumia mali za mtindo kama vile rangi, fonti, ukubwa wa maandishi, upana wa vifungo, nafasi, na zaidi.

Kwa kutumia CSS, unaweza:

1. Kupanga vitu vya kurasa za wavuti: Unaweza kudhibiti jinsi vitu kama vile maandishi, picha, vichwa vya habari, viungo, na sehemu zingine za kurasa zinavyoonekana na zinavyopangwa kwenye ukurasa.

2. Kurekebisha rangi na fonti: Unaweza kuchagua rangi za maandishi, historia, na fonti za maandishi kwa kurasa zako za wavuti.

3. Kuboresha utumiaji wa mtumiaji: CSS inaweza kutumika kutoa athari za hali ya juu kama vile michoro, mabadiliko ya rangi, na hoja za vitu wakati mtumiaji anavyoingiliana na kurasa za wavuti.

4. Kufanya kurasa za wavuti zipatikane zaidi: Kwa kutumia CSS kwa njia sahihi, unaweza kuboresha upatikanaji wa wavuti kwa watu wenye ulemavu na vifaa vya kutumia.

CSS inatumika kwa kutumia sheria za mtindo zinazoelezwa kwenye faili ya CSS na kisha kuunganishwa na kurasa za wavuti za HTML. Hii inaruhusu kuwa na udhibiti mkubwa juu ya jinsi maudhui yako yanavyoonekana na kuhisi kwa watumiaji wa wavuti.

HISTORIA YA CSS

CSS (Cascading Style Sheets) ilianzishwa na Håkon Wium Lie na Bert Bos. Håkon Wium Lie alikuwa mtaalamu wa teknolojia kutoka Norway, na Bert Bos alikuwa mtaalamu wa kompyuta kutoka Uholanzi. Wote walikuwa wakifanya kazi kwa pamoja kwenye Shirika la World Wide Web Consortium (W3C) wakati CSS ilipoanza kuundwa mnamo mwaka 1994.

Håkon Wium Lie alitoa wazo la kuunda lugha mpya ya mtindo ambayo ingeweza kutumika kudhibiti muundo na mtindo wa kurasa za wavuti bila kutegemea sana vitambulisho vya mtindo vilivyokuwa vikitumiwa wakati huo. Bert Bos alikuwa msanifu wa lugha ya mtindo ambayo ilifanikisha wazo hilo na kuanzisha CSS kama tunavyoijua leo.

Kwa pamoja, Håkon Wium Lie na Bert Bos walikuwa sehemu ya timu iliyofanya kazi kwenye CSS kwa niaba ya W3C, na matokeo yake yalikuwa viwango vya CSS Level 1 na baadaye CSS Level 2. Wao ndio waliochangia sana katika maendeleo ya CSS na kusaidia kuleta mabadiliko makubwa katika jinsi kurasa za wavuti zinavyoweza kuundwa na kubuniwa.

Historia ya CSS (Cascading Style Sheets) ina mizizi yake katika maendeleo ya wavuti na jitihada za kuunda njia bora ya kuweka mtindo kwa kurasa za wavuti. Hapa kuna muhtasari wa historia ya CSS:

1. Mapema miaka ya 1990:
   - Katika miaka ya 1990, wavuti zilikuwa zikitumia HTML (Hypertext Markup Language) kwa kuweka muundo na mtindo wa kurasa za wavuti. Walakini, HTML ilikuwa na mapungufu katika kudhibiti mtindo na muundo.

2. 1994:

- Hadi mwaka 1994, matumizi ya HTML yalikuwa yakisambaa kwa kasi, na tovuti zilikuwa zikiwa na muundo uliojaa sana katika HTML. Hii ilifanya kubadilisha muundo wa wavuti kuwa kazi ngumu na isiyo na utaratibu.

3. 1994-1996:
   - Hafla muhimu zilizotokea katika kipindi hiki ni pamoja na maendeleo ya HTML+ (HTML Plus) na jukwaa la Netscape Navigator. Netscape ilianzisha "HTML Tags" (sasa inajulikana kama "Netscape HTML") na ilijaribu kuboresha udhibiti wa mtindo kwa kuongeza vitambulisho maalum vya kufafanua mtindo katika HTML.

4. 1996:
   - Shirika la World Wide Web Consortium (W3C) lilianzisha Tim Berners-Lee, mwanzilishi wa World Wide Web, lilianzisha mradi wa CSS ili kutatua matatizo ya udhibiti wa mtindo wa HTML.

5. 1996-1997:
   - CSS Level 1 ilichapishwa na W3C mwaka 1996 na ilitoa msingi wa kwanza wa kuweka mtindo kwenye wavuti. Hata hivyo, utekelezaji wake na msaada wa browser ulikuwa mdogo mwanzoni.

6. 1998:
   - CSS Level 2 ilichapishwa mwaka 1998, na ilileta maboresho makubwa katika kudhibiti mtindo na muundo wa kurasa za wavuti. CSS Level 2 iliruhusu kwa udhibiti wa kina wa kurasa za wavuti na ilikuwa na msaada bora kwa browsers.

7. 2000 na kuendelea:
   - CSS ilikua kwa kasi baada ya CSS Level 2, na viwango vya CSS vimeendelea kuongezeka. CSS3 ilileta sifa nyingi za hali ya juu kama michoro, mabadiliko, na vipengele vingine vya kuongeza ubunifu kwenye wavuti.

Kwa miaka mingi iliyofuata, CSS imeendelea kuwa sehemu muhimu ya maendeleo ya wavuti na imesaidia kuboresha utumiaji na muundo wa kurasa za wavuti. Leo, CSS inaendelea kuwa muhimu kwa kubuni wavuti zenye ubora na za kuvutia.

CSS ni kifupi cha "Cascading Style Sheets" (Inatafsiriwa kwa Kiswahili kama "Vitambulisho vya Mtindo wa Kupangilia"). Ni lugha ya programu inayotumiwa kwa kawaida katika maendeleo ya wavuti ili kuunda na kuweka mtindo na muundo wa kurasa za wavuti. CSS inaruhusu wavuti kuonekana vizuri zaidi na kuwa na muundo unaofaa kwa kutumia mali za mtindo kama vile rangi, fonti, ukubwa wa maandishi, upana wa vifungo, nafasi, na zaidi.

Kwa kutumia CSS, unaweza:

1. Kupanga vitu vya kurasa za wavuti: Unaweza kudhibiti jinsi vitu kama vile maandishi, picha, vichwa vya habari, viungo, na sehemu zingine za kurasa zinavyoonekana na zinavyopangwa kwenye ukurasa.

2. Kurekebisha rangi na fonti: Unaweza kuchagua rangi za maandishi, historia, na fonti za maandishi kwa kurasa zako za wavuti.

3. Kuboresha utumiaji wa mtumiaji: CSS inaweza kutumika kutoa athari za hali ya juu kama vile michoro, mabadiliko ya rangi, na hoja za vitu wakati mtumiaji anavyoingiliana na kurasa za wavuti.

4. Kufanya kurasa za wavuti zipatikane zaidi: Kwa kutumia CSS kwa njia sahihi, unaweza kuboresha upatikanaji wa wavuti kwa watu wenye ulemavu na vifaa vya kutumia.

CSS inatumika kwa kutumia sheria za mtindo zinazoelezwa kwenye faili ya CSS na kisha kuunganishwa na kurasa za wavuti za HTML. Hii inaruhusu kuwa na udhibiti mkubwa juu ya jinsi maudhui yako yanavyoonekana na kuhisi kwa watumiaji wa wavuti.

HISTORIA YA CSS

CSS (Cascading Style Sheets) ilianzishwa na Håkon Wium Lie na Bert Bos. Håkon Wium Lie alikuwa mtaalamu wa teknolojia kutoka Norway, na Bert Bos alikuwa mtaalamu wa kompyuta kutoka Uholanzi. Wote walikuwa wakifanya

kazi kwa pamoja kwenye Shirika la World Wide Web Consortium (W3C) wakati CSS ilipoanza kuundwa mnamo mwaka 1994.

Håkon Wium Lie alitoa wazo la kuunda lugha mpya ya mtindo ambayo ingeweza kutumika kudhibiti muundo na mtindo wa kurasa za wavuti bila kutegemea sana vitambulisho vya mtindo vilivyokuwa vikitumiwa wakati huo. Bert Bos alikuwa msanifu wa lugha ya mtindo ambayo ilifanikisha wazo hilo na kuanzisha CSS kama tunavyoijua leo.

Kwa pamoja, Håkon Wium Lie na Bert Bos walikuwa sehemu ya timu iliyofanya kazi kwenye CSS kwa niaba ya W3C, na matokeo yake yalikuwa viwango vya CSS Level 1 na baadaye CSS Level 2. Wao ndio waliochangia sana katika maendeleo ya CSS na kusaidia kuleta mabadiliko makubwa katika jinsi kurasa za wavuti zinavyoweza kuundwa na kubuniwa.

Historia ya CSS (Cascading Style Sheets) ina mizizi yake katika maendeleo ya wavuti na jitihada za kuunda njia bora ya kuweka mtindo kwa kurasa za wavuti. Hapa kuna muhtasari wa historia ya CSS:

1. Mapema miaka ya 1990:
   - Katika miaka ya 1990, wavuti zilikuwa zikitumia HTML (Hypertext Markup Language) kwa kuweka muundo na mtindo wa kurasa za wavuti. Walakini, HTML ilikuwa na mapungufu katika kudhibiti mtindo na muundo.

2. 1994:
   - Hadi mwaka 1994, matumizi ya HTML yalikuwa yakisambaa kwa kasi, na tovuti zilikuwa zikiwa na muundo uliojaa sana katika HTML. Hii ilifanya kubadilisha muundo wa wavuti kuwa kazi ngumu na isiyo na utaratibu.

3. 1994-1996:
   - Hafla muhimu zilizotokea katika kipindi hiki ni pamoja na maendeleo ya HTML+ (HTML Plus) na jukwaa la Netscape Navigator. Netscape ilianzisha "HTML Tags" (sasa inajulikana kama "Netscape HTML") na ilijaribu kuboresha udhibiti wa mtindo kwa kuongeza vitambulisho maalum vya kufafanua mtindo katika HTML.

4. 1996:
   - Shirika la World Wide Web Consortium (W3C) lilianzisha Tim Berners-Lee, mwanzilishi wa World Wide Web, lilianzisha mradi wa CSS ili kutatua matatizo ya udhibiti wa mtindo wa HTML.

5. 1996-1997:
   - CSS Level 1 ilichapishwa na W3C mwaka 1996 na ilitoa msingi wa kwanza wa kuweka mtindo kwenye wavuti. Hata hivyo, utekelezaji wake na msaada wa browser ulikuwa mdogo mwanzoni.

6. 1998:
   - CSS Level 2 ilichapishwa mwaka 1998, na ilileta maboresho makubwa katika kudhibiti mtindo na muundo wa kurasa za wavuti. CSS Level 2 iliruhusu kwa udhibiti wa kina wa kurasa za wavuti na ilikuwa na msaada bora kwa browsers.

7. 2000 na kuendelea:
   - CSS ilikua kwa kasi baada ya CSS Level 2, na viwango vya CSS vimeendelea kuongezeka. CSS3 ilileta sifa nyingi za hali ya juu kama michoro, mabadiliko, na vipengele vingine vya kuongeza ubunifu kwenye wavuti.

Kwa miaka mingi iliyofuata, CSS imeendelea kuwa sehemu muhimu ya maendeleo ya wavuti na imesaidia kuboresha utumiaji na muundo wa kurasa za wavuti. Leo, CSS inaendelea kuwa muhimu kwa kubuni wavuti zenye ubora na za kuvutia.

CSS - somo la 2: Jinsi ya ku weka code za css kwenye HTML

katika somo hili utajifunza jinsi ya ku install css kwenye ukurasa wa html

Utangulizi

Css ni kifupisho cha maneno cascading style sheet. Hii ni moja katika lugha za kompyuta ambazo hutumika katika

kuboresha mtindo kwenye ukurasa wa wavuti. Hutumika kwenye HTML na XML. css ni style sheet angauage, hata hivyo haipo pekee kwenye kundi hili. Lugha nyingine za style ni kama:

- JavaScript Style Sheets (JSSS)
- Formatting Output Specification Instance (FOSI)
- Syntactically Awesome Style Sheets (Sass)
- Less (Less)
- Stylus
- SMIL Timesheets
- Document Style Semantics and Specification Language (DSSSL)
- Extensible Stylesheet Language (XSL)

Css ilianzishwa miaka ya 1990 na kuendelea. Historia a css imeanza punde tu baada ya tovuti ya kwnza kuchapishwa na watu wakaanza kutengeneza tovuti mbalimbali. Hivyo ikaonekana kunahitajika kuwepo na ubireshwaji wa muonekano.

Mwaka 1994 ndipo css ilianzishwa rasmi na Håkon Wium Lie na Bert Bos. Kutokea hapo css iliweza kukuw kwa kasi. Na kuanza kupata u maarufu. Mpaka sasa tupo toleo la 4 la css.

Kama nilivyotangulua kusema kuwa css hutumiaka kwenye HTML na xml katika course hii tutakwenda kujifunza kutumia css kwenye html. Tutajifunza kutumia style sheet kwneye xml kwenye course ya xml.

JINSI YA KU INSTALL CSS

Hakuna uhitaji wa ku install css ili uweze kuitumia kama zilivyo lugha nyingne. Unachotakiwa kufanya ni kuitumia moja kwa moja kwenye ukurasa wako wa wavuti kwenye html. Kuna njia kadhaa ambazo hutumika katika kutumia css kwenye faili la HTML. Njia hizo ni kama:-

1. **In line css**: Hapa tuna andika css ndani ya mstari katika faili la html.
2. **Internal css** Hapa tutatumia css ndani ya faili la html.
3. **External css** Hapa tunaandika css kwenye faili lingine kisha tuna iunganisha moja kwa moja
4. **Kwa kutumia javascript** Katika njia hii tunanaitumia css kenye code za javascript.

Katika course hii tutakwenda kutumia njia hizo tatu za kwanza. Njia ya nne tutatjifunza kwenye course ya javascript level 1 na level 2.

In line css

Njia hii utakuwa na uzoefu nayo kwani tumesha itumia mara kadhaa kwenye course yetu ya html level 1 na level 2. Njia hii ni ile ya kutumia tag za css kwenye fali la html. Tag yenye we ni ile ya **<style>.** Kwa kutumia tag hii ndani ya html itakuwezesha kuweza kuandika code za css na zikaweza kufanya kazi.

Katika inline css tutaandika css ndani ya tag ya html kw akuanza na neno style likifatiwa na attribute zake. Tutakwenda kutumia code hizi kwenye mfano huu:-

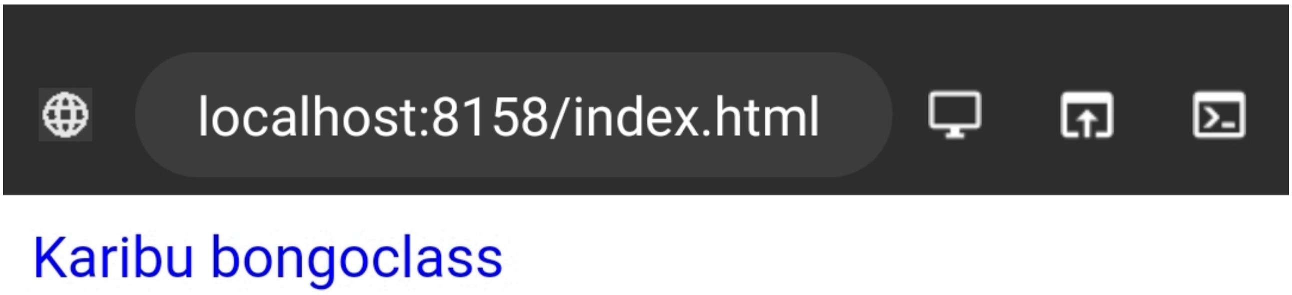
```
<html>

<head>

   <title>Bongoclass</title>

</head>

<body>

<p>Karibu bongoclass</p>

</body>

</html>
```

Mfano 1:

Tunataka kubadili nano karibu bongoclass kuwa na rangi ya buluu. Tutafanya hivi kwa kutumia in line css. Unachotakiwa kufanya ni kuweka style ndani ya tag ya <p>, ikifuatiwa na alama ya = kisha utaweka hizo attribute za css kwenye alama za kunukuu (**“**).

Uandishi wa attribute utakuwa hivi, kwanza utaweka jina la property likifuatiliwa na nukta pacha (**:**) ama colon, Kisha utaandika value. Mfano ikitaka rangi iwe ya buluu utaweka style="color:blue" katika mfano huo hapo neno color ndio property na neno blue ndio value.

Hivyo tag ya html <p>itasomeka hivi <p style ="color:blue">Karibu bongoclass</p>



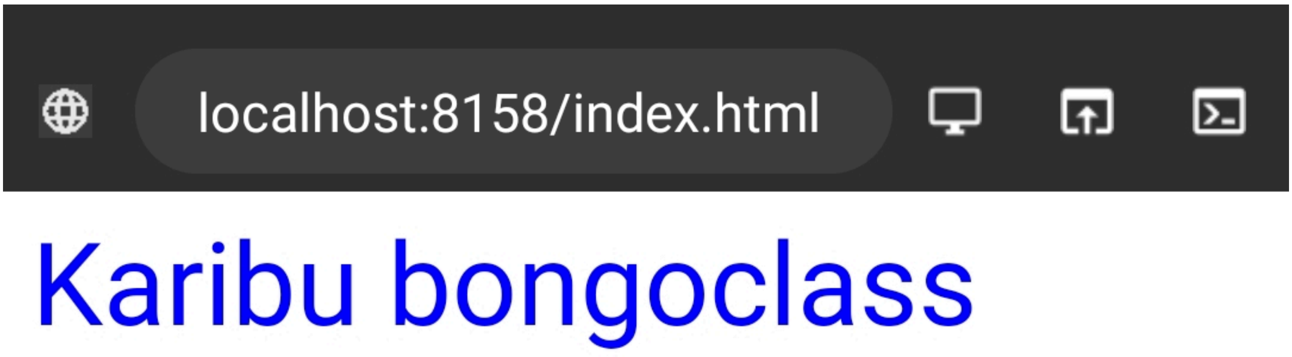
Pia unaweza kuweka style zaidi ya Moja kwenye inline css. Kufanya hivyo utatenganisha style Moja na nyingine kwa kutumia alama ya semicolon yaani **(;).**

Mfano 2:

Katika mfano wetu huu tunataka kufanya maadhishi yawe makubwa kwa asilimia 200%. Kufanya hivi tutatumia

property ya **font-size** kwa kuwa tunayo property ya color basi tutatenganisha kwa semicolon

```
<p style ="color:blue; font-size:200%">Karibu bongoclass</p>
```



Internal css

Tunapotumia internal css hapa tunaweka code zote za style sehemu Moja. Mara nyingi code hizo huwekwa sehemu **head** kwenye faili la html. Na code hizo za css huwa kwenye tag ya <style>

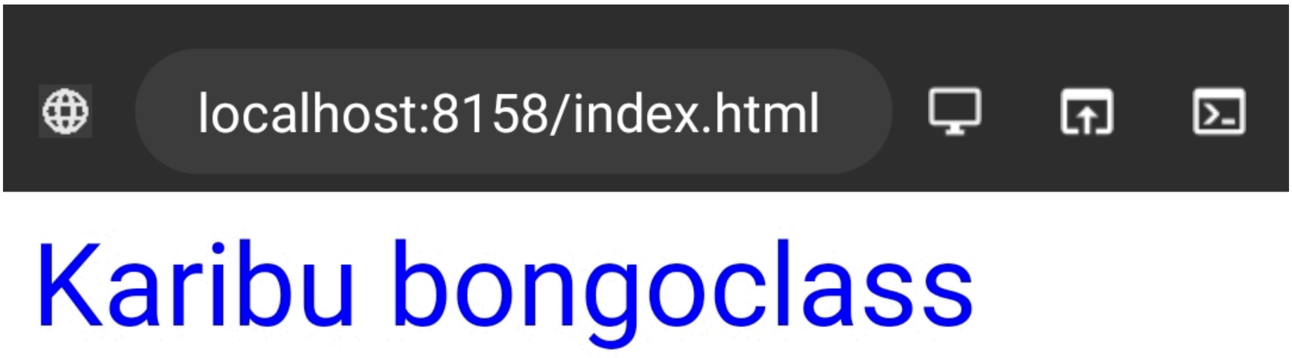
Katika internal css code za css zitakuwa ndani ya mabano {}. Tutakuja kujifunza Sheria zaidi za css kwenye somo linalofuata.

Katika mfano wetu wa hapo juu code hizo kwenye internal css zitakuwa hivi:-

```
<style>
     p{
       color:blue;
       font-size:200%
     }
</style>
```

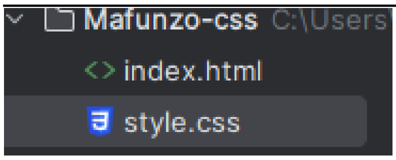
Code zote zitakuwa hivi:-

```
<html>
  <head>
    <title>bongoclass</title>
    <style>
      p{
        color:blue;
        font-size:200%
      }
    </style>
  </head>
  <body>
    <p>Karibu bongoclass</p>
  </body>
</html>
```



External css

Katika mtindo huu tutaweka code za css mbali na faili la html. Kwanza tutatengeneza faili la css, kisha tutakuja kuliunganisha na html. Kwenye kifaa unachotumia tengeneza faili ipya liite style.css. Kisha weka code hizo tulizoweka kwenye internal css ila kusiwepo na hizo tag za style.



Utaona hapo kwenye folda langu kuna mafaili mawili ambayo ni **index.html** na faili la **style.css.** Hili la index ndlo ambalo litakaa code za html na hili la style ndilo ambalo litakaa code za css. Kama nilivyokwambai hapo pest code zetu tulizotumia hapo juu

```
 p{
       color:blue;
       font-size:200%
    }
```

Kwenye faili la css code zetu zitaonekana hivi:-

Sasa tunakwenda kuunganisha mafaili yetu haya mawili ili code za css ziweze kutumika kwenye faili la index.html. Ili kufanya hivi tutatumia tag ya <**link**> ambayo itakuwa na attribute ya **rel** hii kazi yake ni kuonyesha uhusiano uliopo kati ya faili na faili lingine. Neno **rel** ni kifupisho cha neno relation. Katika tag ya kink kuna attribute nyingine kama **type** na **href**

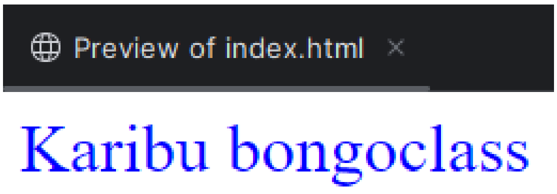
Katika rel tutaweka property ya stylesheet kuonesha aina ya hilo faili tutalotaka kuliunganisha kwenye faili letu la html. Mfano **rel=”stylesheet”**. Katika type tutaweka MIME type ama format ambayo ni text/css mfano **type=”text/css”** na katika href tutaweka link ama location ama path ya hilo faili la css. Mfano **href=”style.css”** kwa kuwa mafaili yote yapo kwenye folder moja.

Hivyo basi tag nzima ya <**link** itasomeka hiv

<link rel="stylesheet" type="text/css" href="style.css">

Code nzima za faili la html ztasomeka hivi

```
<html lang="sw">

<head>

    <title>Bongoclass</title>

    <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<p>Karibu bongoclass</p>

</body>

</html>
```



Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu sheria za uandishi wa css.

Utangulizi

Css ni kifupisho cha maneno cascading style sheet. Hii ni moja katika lugha za kompyuta ambazo hutumika katika kuboresha mtindo kwenye ukurasa wa wavuti. Hutumika kwenye HTML na XML. css ni style sheet angauage, hata hivyo haipo pekee kwenye kundi hili. Lugha nyingine za style ni kama:

- JavaScript Style Sheets (JSSS)
- Formatting Output Specification Instance (FOSI)
- Syntactically Awesome Style Sheets (Sass)
- Less (Less)
- Stylus
- SMIL Timesheets
- Document Style Semantics and Specification Language (DSSSL)
- Extensible Stylesheet Language (XSL)

Css ilianzishwa miaka ya 1990 na kuendelea. Historia a css imeanza punde tu baada ya tovuti ya kwnza kuchapishwa na watu wakaanza kutengeneza tovuti mbalimbali. Hivyo ikaonekana kunahitajika kuwepo na ubireshwaji wa muonekano.

Mwaka 1994 ndipo css ilianzishwa rasmi na Håkon Wium Lie na Bert Bos. Kutokea hapo css iliweza kukuw kwa

kasi. Na kuanza kupata u maarufu. Mpaka sasa tupo toleo la 4 la css.

Kama nilivyotangulua kusema kuwa css hutumiaka kwenye HTML na xml katika course hii tutakwenda kujifunza kutumia css kwenye html. Tutajifunza kutumia style sheet kwneye xml kwenye course ya xml.

JINSI YA KU INSTALL CSS

Hakuna uhitaji wa ku install css ili uweze kuitumia kama zilivyo lugha nyingne. Unachotakiwa kufanya ni kuitumia moja kwa moja kwenye ukurasa wako wa wavuti kwenye html. Kuna njia kadhaa ambazo hutumika katika kutumia css kwenye faili la HTML. Njia hizo ni kama:-

1. **In line css**: Hapa tuna andika css ndani ya mstari katika faili la html.
2. **Internal css** Hapa tutatumia css ndani ya faili la html.
3. **External css** Hapa tunaandika css kwenye faili lingine kisha tuna iunganisha moja kwa moja
4. **Kwa kutumia javascript** Katika njia hii tunanaitumia css kenye code za javascript.

Katika course hii tutakwenda kutumia njia hizo tatu za kwanza. Njia ya nne tutatjifunza kwenye course ya javascript level 1 na level 2.

In line css

Njia hii utakuwa na uzoefu nayo kwani tumesha itumia mara kadhaa kwenye course yetu ya html level 1 na level 2. Njia hii ni ile ya kutumia tag za css kwenye fali la html. Tag yenye we ni ile ya **<style>.** Kwa kutumia tag hii ndani ya html itakuwezesha kuweza kuandika code za css na zikaweza kufanya kazi.

Katika inline css tutaandika css ndani ya tag ya html kw akuanza na neno style likifatiwa na attribute zake. Tutakwenda kutumia code hizi kwenye mfano huu:-

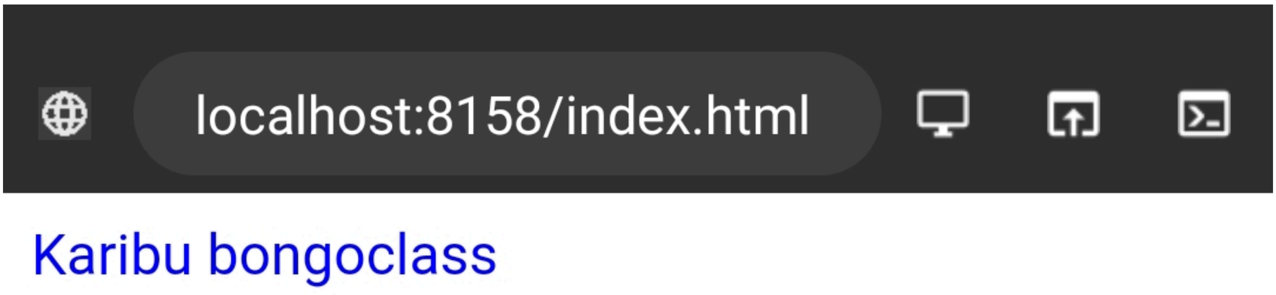
```
<html>

<head>

   <title>Bongoclass</title>

</head>

<body>

<p>Karibu bongoclass</p>

</body>

</html>
```

Mfano 1:

Tunataka kubadili nano karibu bongoclass kuwa na rangi ya buluu. Tutafanya hivi kwa kutumia in line css. Unachotakiwa kufanya ni kuweka style ndani ya tag ya <p>, ikifuatiwa na alama ya **=** kisha utaweka hizo attribute za css kwenye alama za kunukuu (**“**).

Uandishi wa attribute utakuwa hivi, kwanza utaweka jina la property likifuatiliwa na nukta pacha (**:**) ama colon, Kisha utaandika value. Mfano ikitaka rangi iwe ya buluu utaweka style="color:blue" katika mfano huo hapo neno color ndio property na neno blue ndio value.

Hivyo tag ya html <p>itasomeka hivi <p style ="color:blue">Karibu bongoclass</p>

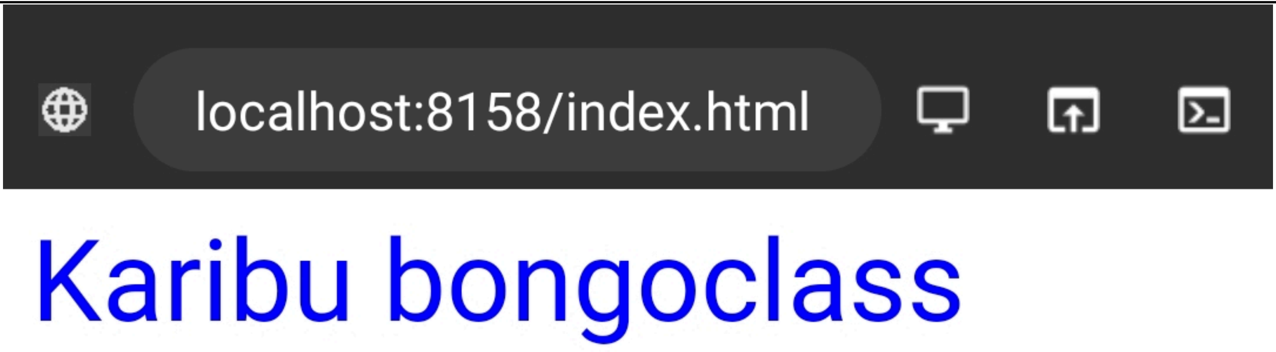


Pia unaweza kuweka style zaidi ya Moja kwenye inline css. Kufanya hivyo utatenganisha style Moja na nyingine kwa kutumia alama ya semicolon yaani **(;).**

Mfano 2:

Katika mfano wetu huu tunataka kufanya maadhishi yawe makubwa kwa asilimia 200%. Kufanya hivi tutatumia property ya **font-size** kwa kuwa tunayo property ya color basi tutatenganisha kwa semicolon

<p style ="color:blue; font-size:200%">Karibu bongoclass</p>



Internal css

Tunapotumia internal css hapa tunaweka code zote za style sehemu Moja. Mara nyingi code hizo huwekwa sehemu **head** kwenye faili la html. Na code hizo za css huwa kwenye tag ya <style>

Katika internal css code za css zitakuwa ndani ya mabano {}. Tutakuja kujifunza Sheria zaidi za css kwenye somo linalofuata.

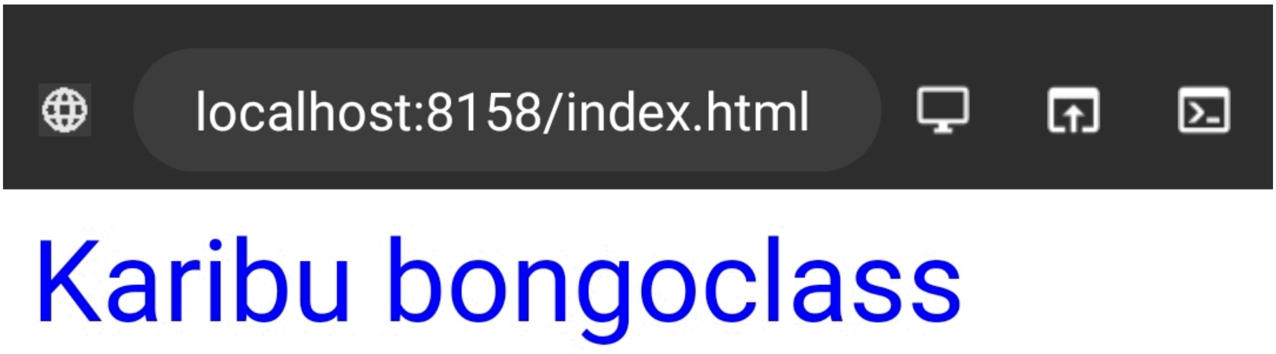
Katika mfano wetu wa hapo juu code hizo kwenye internal css zitakuwa hivi:-

```
<style>
     p{
       color:blue;
       font-size:200%
     }
</style>
```

Code zote zitakuwa hivi:-

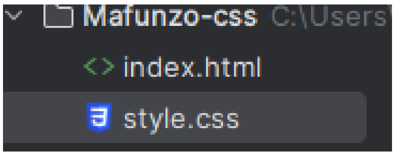
```
<html>
  <head>
    <title>bongoclass</title>
```

```
    <style>
      p{
        color:blue;
        font-size:200%
      }
    </style>
  </head>
  <body>
    <p>Karibu bongoclass</p>
  </body>
</html>
```



External css

Katika mtindo huu tutaweka code za css mbali na faili la html. Kwanza tutatengeneza faili la css, kisha tutakuja kuliunganisha na html. Kwenye kifaa unachotumia tengeneza faili ipya liite style.css. Kisha weka code hizo tulizoweka kwenye internal css ila kusiwepo na hizo tag za style.



Utaona hapo kwenye folda langu kuna mafaili mawili ambayo ni **index.html** na faili la **style.css.** Hili la index ndlo

ambalo litakaa code za html na hili la style ndilo ambalo litakaa code za css. Kama nilivyokwambai hapo pest code zetu tulizotumia hapo juu

```
 p{

      color:blue;

      font-size:200%

   }
```

Kwenye faili la css code zetu zitaonekana hivi:-

Sasa tunakwenda kuunganisha mafaili yetu haya mawili ili code za css ziweze kutumika kwenye faili la index.html. Ili kufanya hivi tutatumia tag ya **<link>** ambayo itakuwa na attribute ya **rel** hii kazi yake ni kuonyesha uhusiano uliopo kati ya faili na faili lingine. Neno **rel** ni kifupisho cha neno relation. Katika tag ya kink kuna attribute nyingine kama **type** na **href**

Katika rel tutaweka property ya stylesheet kuonesha aina ya hilo faili tutalotaka kuliunganisha kwenye faili letu la html. Mfano **rel=”stylesheet”**. Katika type tutaweka MIME type ama format ambayo ni text/css mfano **type=”text/css”** na katika href tutaweka link ama location ama path ya hilo faili la css.
Mfano **href=”style.css”** kwa kuwa mafaili yote yapo kwenye folder moja.

Hivyo basi tag nzima ya **<link** itasomeka hiv

<link rel="stylesheet" type="text/css" href="style.css">

Code nzima za faili la html ztasomeka hivi

```
<html lang="sw">

<head>

    <title>Bongoclass</title>

    <link rel="stylesheet" type="text/css" href="style.css">
```

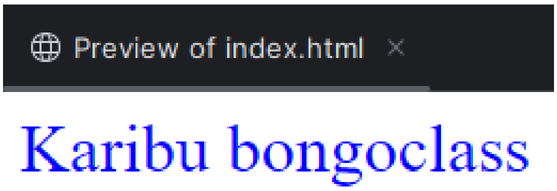
</head>

<body>

<p>Karibu bongoclass</p>

</body>

</html>



Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu sheria za uandishi wa css.

CSS - somo la 3: syntax za css yaani sheria za uandishi wa css

Katika somo hili utakwenda kujifunza sheria za uandish wa css yaani syntax za css

Sheria za uandishi wa css yaani syntax za css.

Css ni moja katika lugha za ki kompyuta (computer language) ambayo ina sheria zilizo rahisi sana na ni chache. Ukishazijuwa hizo basi mabo yote yatakuwa sawa.

Sheria za kuandka css zimezama kwenye mambo makuu mawili tu ambayo ni **selector** na **declaration**. Yaani code zote za css unaweza kuzigawa kwenye sehemu kuu hizi mbili ambazo ni selector na declaration.

Selectors: hii ni sehemu ambayo yenyewe inahusisha element za html. Yaani hapa ndipo tunakwenda kujuwa je hizo code za css zinahud element gani. Mfano ukisema

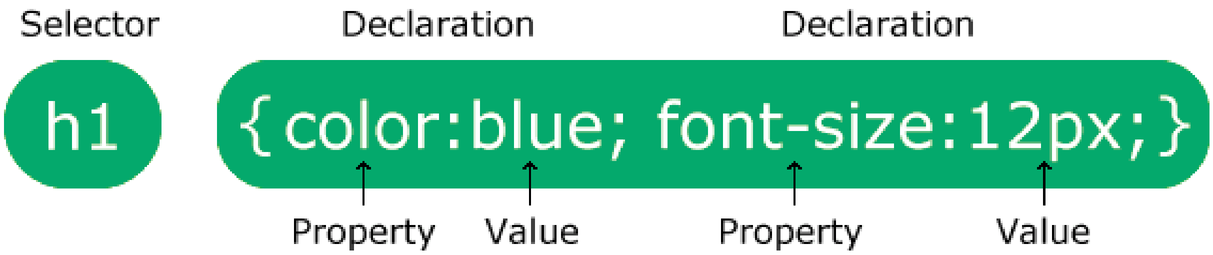
haloo

katika code hizi selector ni

ambapo inaonesha kuwa tuna target paragraph.

Declaration: hii ni sehemu ambayo inakwenda kuweka code za css. Declaration yenyewe imegawanyika katika sehemu kuu mbili ambazo ni **property** na **value**. Kwa mfano katika mfano niliotoa hapo juu declaration ni **color:yellow** ambapo tunapata color ni property na yellow ni value.

Kwa ufupi selector ina kuwa na tag husika ya html na declaration inakuwa na protery pamoja na value. Declaration inakuwa kwenye mabano ya **{}** kama utatumia internal css ama external css.



[credit: w3schools]

Selector inaweza kuwa zaidi ya moja na declaration pia inawezakuwa zaidi ya moja. Jambo la kuzingatia nikiwa unapotumia selector zaidi ya Moja utatakiwa kuzitenganisha kwa alama ya koma (,). Na unapotumia declation zaidi ya Moja utatenganisha kwa kutumia semicolon (;). Angalia mfano hapo chini

```
p, h1, u, i, b{

  color:blue;

  font-size:200%;

  background: red;

}
```

Katika code hizi selectors ni **p, h1, u, i** na **b** hizi zinatarget tag za html. Tag hizo ni <p>, <h1>, <u>, <i> na <b>. Kwa pamoja element hizo zote zitafuata style hiyo popote zitakapotumiwa.

Pia hapo kuna declaration tatu ambazo ni color:blue yenye property color na value blue, font-size:100% yenye property font-size na value 100%, na ya mwisho ni background-color:red yenye property background-color na value red.

Baada ya kutambuwa kuhusu sheria hizo basi unaweza kuandika cess kwa msaada wa masomo yaliotangulia.rejea njia za kutumia css kwenye faili la html.

Jinsi ya ku comment: Ni kawaida kwa kila lugha ya kompyuta kuweka namna ya ku comment ili kuwezesha watu kuweka kumbukumbu fulani. Tofauti na html ambayo hutumia <!-- –> katika ku comment css hutumia alama hizi /* */

```
p, h1, u, i, b{

  /*rangi ya buluu */

  color:blue;

  font-size:200%;

  background: red;
```

```
   /*pia unaweza kujifunza

   zaidi kuhusu jinsi

   ya ku comment

    */

}
```

Kama unavyoona hapo nimecomment mstari mmoja, pia nikakoment kwa mistari zaidiya mmoja kwa kutumia alama zile zile.

Inheritance: maana yake ni kurithi. Hii ni sheria muhimu sana kuijuwa.ili uijuwe vyema zingatia haya:-

Katika HTML element kuna ambazo ni parent na kuna children. Mfano element <body> nii ni parent ambapo ndani yake kuna element nyingi zinazounda maudhui ya faili husika. Kama kama <body> utaipa style basi style hiyo ita athiri faili lote. Kwa maana ukisema body{color:red} ina maana ement zote zitakuwa na rangi nyekundu

Pia element child inaweza ikawa na element nyingine ndaniyake. Kwa mfano element <p> ndani yake inaweza kubeba element kama b, i, u ambapo style utakayoiweka kwenye <p> itaweza ku athiri element zote zilizopo humo.

Sasa katika sheria hii ni kuwa code zilizo juu zinaweza athiri vilivyochini hata hivyo endapo utaweka style kwa code zinazofuata zitachukuwa style hiyo mpya. Mfano kwenye <body> unaweza kuset rangi ya maandishi kuwa ya buluu kwamfano, kisha kwenye <p> ukaiset iwe namaandishi ya kijani. Sasa hapa sheria ya inheritance haitafanya kazi kwa sababu imekutana na sheria ndogondogo.

Halikadhalika pia zingatia haya:-

Style za external css zinaweza kuvunjwa na internal css

Style za internal css zinaweza kuvunjwa na inline css

Sheria za inline css zinaweza kuvunjwa kwa kutumia id selector. Tutakuja jifunza aina za selecto

Mwisho:

Katika somo linalofuata utakwenda k

ujifunza aina za selector
Sheria za uandishi wa css yaani syntax za css.

Css ni moja katika lugha za ki kompyuta (computer language) ambayo ina sheria zilizo rahisi sana na ni chache. Ukishazijuwa hizo basi mabo yote yatakuwa sawa.

Sheria za kuandka css zimezama kwenye mambo makuu mawili tu ambayo ni **selector** na **declaration**. Yaani code zote za css unaweza kuzigawa kwenye sehemu kuu hizi mbili ambazo ni selector na declaration.

Selectors: hii ni sehemu ambayo yenyewe inahusisha element za html. Yaani hapa ndipo tunakwenda kujuwa je hizo code za css zinahud element gani. Mfano ukisema

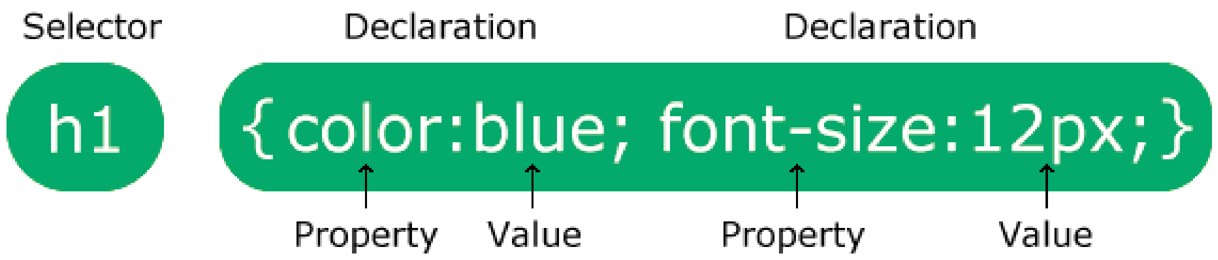
haloo

katika code hizi selector ni

ambapo inaonesha kuwa tuna target paragraph.

Declaration: hii ni sehemu ambayo inakwenda kuweka code za css. Declaration yenyewe imegawanyika katika sehemu kuu mbili ambazo ni **property** na **value**. Kwa mfano katika mfano niliotoa hapo juu declaration ni **color:yellow** ambapo tunapata color ni property na yellow ni value.

Kwa ufupi selector ina kuwa na tag husika ya html na declaration inakuwa na protery pamoja na value. Declaration inakuwa kwenye mabano ya **{}** kama utatumia internal css ama external css.



credit: w3schools

Selector inaweza kuwa zaidi ya moja na declaration pia inawezakuwa zaidi ya moja. Jambo la kuzingatia nikiwa unapotumia selector zaidi ya Moja utatakiwa kuzitenganisha kwa alama ya koma (,). Na unapotumia declation zaidi ya Moja utatenganisha kwa kutumia semicolon (;). Angalia mfano hapo chini

```
p, h1, u, i, b{

  color:blue;

  font-size:200%;

  background: red;

}
```

Katika code hizi selectors ni **p, h1, u, i** na **b** hizi zinatarget tag za html. Tag hizo ni <p>, <h1>, <u>, <i> na <b>. Kwa pamoja element hizo zote zitafuata style hiyo popote zitakapotumiwa.

Pia hapo kuna declaration tatu ambazo ni color:blue yenye property color na value blue, font-size:100% yenye property font-size na value 100%, na ya mwisho ni background-color:red yenye property background-color na value red.

Baada ya kutambuwa kuhusu sheria hizo basi unaweza kuandika cess kwa msaada wa masomo yaliotangulia.rejea njia za kutumia css kwenye faili la html.

Jinsi ya ku comment: Ni kawaida kwa kila lugha ya kompyuta kuweka namna ya ku comment ili kuwezesha watu kuweka kumbukumbu fulani. Tofauti na html ambayo hutumia <!-- –> katika ku comment css hutumia alama hizi /* */

```
p, h1, u, i, b{

  /*rangi ya buluu */

  color:blue;

  font-size:200%;

  background: red;

  /*pia unaweza kujifunza

  zaidi kuhusu jinsi

  ya ku comment

   */

}
```

Kama unavyoona hapo nimecomment mstari mmoja, pia nikakoment kwa mistari zaidiya mmoja kwa kutumia alama zile zile.

Inheritance: maana yake ni kurithi. Hii ni sheria muhimu sana kuijuwa.ili uijuwe vyema zingatia haya:-

Katika HTML element kuna ambazo ni parent na kuna children. Mfano element <body> nii ni parent ambapo ndani yake kuna element nyingi zinazounda maudhui ya faili husika. Kama kama <body> utaipa style basi style hiyo ita athiri faili lote. Kwa maana ukisema body{color:red} ina maana ement zote zitakuwa na rangi nyekundu

Pia element child inaweza ikawa na element nyingine ndaniyake. Kwa mfano element <p> ndani yake inaweza kubeba element kama b, i, u ambapo style utakayoiweka kwenye <p> itaweza ku athiri element zote zilizopo humo.

Sasa katika sheria hii ni kuwa code zilizo juu zinaweza athiri vilivyochini hata hivyo endapo utaweka style kwa code zinazofuata zitachukuwa style hiyo mpya. Mfano kwenye <body> unaweza kuset rangi ya maandishi kuwa ya buluu kwamfano, kisha kwenye <p> ukaiset iwe namaandishi ya kijani. Sasa hapa sheria ya inheritance haitafanya kazi kwa sababu imekutana na sheria ndogondogo.

Halikadhalika pia zingatia haya:-

Style za external css zinaweza kuvunjwa na internal css

Style za internal css zinaweza kuvunjwa na inline css

Sheria za inline css zinaweza kuvunjwa kwa kutumia id selector. Tutakuja jifunza aina za selecto

Mwisho:

Katika somo linalofuata utakwenda k

ujifunza aina za selector

CSS - somo la 4: Aina za css selecto

Katika somo hili uatkwenda kujifunza aina za css selectors

Aina za selector

Katika css selectors unaweza kuzigawa katika makundi ma nne abayo ni kama:-

1. **Element selectors**
2. **Class selectors**
3. **Id selectors**
4. **Group selectos**
5. **Universal selector**

1. Element selectors

Hizi ni zile selector ambazo zinatarget element za html, kama vile heading kwa <h1>, paragraph kwa <p>, bolding kwa <b> na kuendelea.

```
b{color: red;

 font-family: "Times New Roman";

 text-align: center;

}
```

Hii imetagget **b**

2. Class selectors

Hizi hutumika wakati ambapo unataka ku target element zaidi ya moja. Unapotumia class selector internal css na external css hazitaweza kufanya kazi katika element husika tu na sio kwa parent element ya hiyo child element. Hii ni nzuri sana kwa kuwa haiwezi kuathiri element nyinginezo kwani haifanyi kazi mpaka iitwe.

Uandishi wa class

Kwanza utaweka nukta yaani period (**.**) ama doti ilifuatiwa na jina la class likifuatiwa na mabano**{}** ambapo ndani ya mabano hayo ndipo utaweka code za css. Mfano tunatengeneza claaa tutakayoiita **muhimu** hii itakuwa na style hizi: rangi nyekundu, font 200%, font-family arial

```
No.muhimu{

   font-size: 200%;

   font-family: Arial;

   color: red;

}
```

Ili kuweza kuitumia class hiyo kwenye element unayo target utaiitwa kwa kuanza kuandika neno class likifuatiwa na alama ya = likufuatiwa na alama za kunukuu, ndani yake weka jina la class.

```
<p class="muhimu">Karibu bongoclass</p>
```

Class hiyo unaweza kuitumia katika element yeyote ile unayohitaji iwe na style hiyo. Na hakuna kikomo juu ya matumiazi yake. Jambo la kuzingatia ni kuwa class itaathiriwa na parent style. Kwa mfano kwenye <body> tuliweka text background colo iwe red, sasa hiyo background color itaendelea kwenye class yetu lamda na sisi kwenye class tu set background ntingine.

Pia zingatia kuwa jina la class halitakiwi kuanza na namba ama space (empty space)

3. ID selectors

Hizi hutumika pale unapo target element moja tu. Element inaweza kuwa na class zaidi ya Moja, lakini haiwezi kuwa na Id zaidi ya Moja.

Pia id huzingatiwa mwanzo kuliko class wakati ukurasa wa wavuti unapofunguliwa kwenye browser.

Aw

Uandishi wa Id

Uandishi wa ID Haina utofauti sana na class, tifauti ni kuwa wakati class inaanza na nukta yaani **period** huku kwenye **ID** utaanza na alama ya reli yaani **hash (#)**.

Na wakati wa kuuuta kwenye hiyo element ni sawa kama ulivyoiita kwenye class.

#muhimu{

font-size: 200%;

font-Rw: Arial;

color: red;

}

Mfano

```
<html lang="sw">

<head>

   <title>Bongoclass</title>

   <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<p id="muhimu">Karibu bongoclass</p>

</body>

</html>
```

4. Group selectors

hizi ni selector ambazo Zina target element zaidi ya Moja. Tulisha zungunzia hili kwenye somo lililopita. Hapa nitakuletea mifano TU uone.

Kwa mfano unataka ku target element hizi tu (**p, u, b, i)** na unataka zote ziwe na style Moja. Hapa ndipo tutatumia group selectors

p,b,u,i{

font-size: 200%;

font-family: Arial;

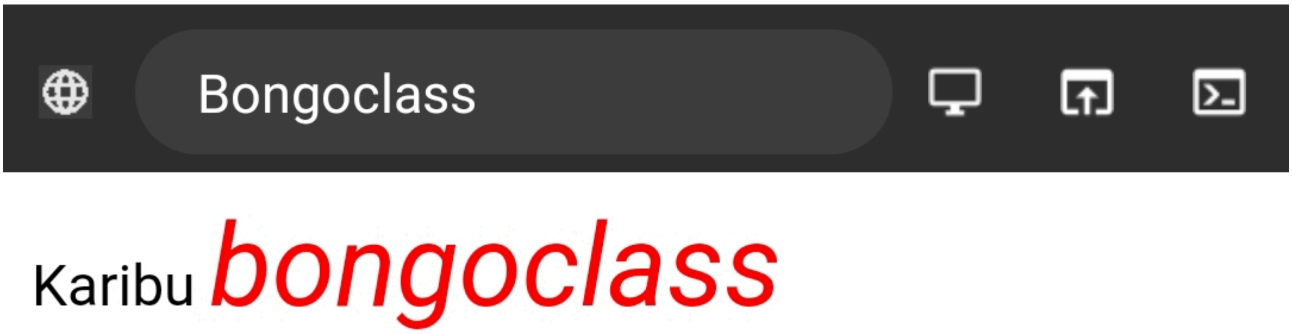
color: red;

}

Kwenye html weka hizi

```
<html lang="sw">
```

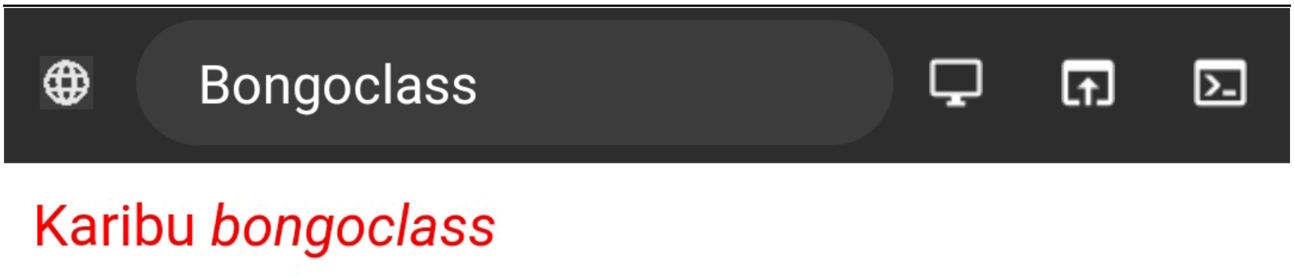
```
<head>

   <title>Bongoclass</title>

   <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<div>Karibu <i>bongoclass</i></div>

</body>

</html>
```



5. Universal selectors

Hii hutumika ku target element zote kwenye faili la html. Hii huandikwa kwa kuanza na alama ya nyota (*****).

```
*{

  font-size: 100%;

  font-family: Arial;

  color: red;

}
```



Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu matumizi ya rangi.
Aina za selector

Katika css selectors unaweza kuzigawa katika makundi ma nne abayo ni kama:-

1. **Element selectors**
2. **Class selectors**
3. **Id selectors**
4. **Group selectos**
5. **Universal selector**

1. Element selectors

Hizi ni zile selector ambazo zinatarget element za html, kama vile heading kwa <h1>, paragraph kwa <p>, bolding kwa <b> na kuendelea.

```
b{color: red;

 font-family: "Times New Roman";

 text-align: center;

}
```

Hii imetagget **b**

2. Class selectors

Hizi hutumika wakati ambapo unataka ku target element zaidi ya moja. Unapotumia class selector internal css na external css hazitaweza kufanya kazi katika element husika tu na sio kwa parent element ya hiyo child element. Hii ni nzuri sana kwa kuwa haiwezi kuathiri element nyinginezo kwani haifanyi kazi mpaka iitwe.

Uandishi wa class

Kwanza utaweka nukta yaani period (**.**) ama doti ilifuatiwa na jina la class likifuatiwa na mabano**{}** ambapo ndani ya mabano hayo ndipo utaweka code za css. Mfano tunatengeneza claaa tutakayoiita **muhimu** hii itakuwa na style hizi: rangi nyekundu, font 200%, font-family arial

```
No.muhimu{

   font-size: 200%;

   font-family: Arial;

   color: red;

}
```

Ili kuweza kuitumia class hiyo kwenye element unayo target utaiitwa kwa kuanza kuandika neno class likifuatiwa na alama ya = likufuatiwa na alama za kunukuu, ndani yake weka jina la class.

```
<p class="muhimu">Karibu bongoclass</p>
```

Class hiyo unaweza kuitumia katika element yeyote ile unayohitaji iwe na style hiyo. Na hakuna kikomo juu ya matumiazi yake. Jambo la kuzingatia ni kuwa class itaathiriwa na parent style. Kwa mfano kwenye <body> tuliweka text background colo iwe red, sasa hiyo background color itaendelea kwenye class yetu lamda na sisi kwenye class tu set background ntingine.

Pia zingatia kuwa jina la class halitakiwi kuanza na namba ama space (empty space)

3. ID selectors

Hizi hutumika pale unapo target element moja tu. Element inaweza kuwa na class zaidi ya Moja, lakini haiwezi kuwa na Id zaidi ya Moja.

Pia id huzingatiwa mwanzo kuliko class wakati ukurasa wa wavuti unapofunguliwa kwenye browser.

Aw

Uandishi wa Id

Uandishi wa ID Haina utofauti sana na class, tifauti ni kuwa wakati class inaanza na nukta yaani **period** huku kwenye **ID** utaanza na alama ya reli yaani **hash (#)**.

Na wakati wa kuuuta kwenye hiyo element ni sawa kama ulivyoiita kwenye class.

```
#muhimu{

   font-size: 200%;
```

```
  font-Rw: Arial;

  color: red;

}
```

Mfano

```
<html lang="sw">

<head>

   <title>Bongoclass</title>

   <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<p id="muhimu">Karibu bongoclass</p>

</body>

</html>
```

4. Group selectors

hizi ni selector ambazo Zina target element zaidi ya Moja. Tulisha zungunzia hili kwenye somo lililopita. Hapa nitakuletea mifano TU uone.

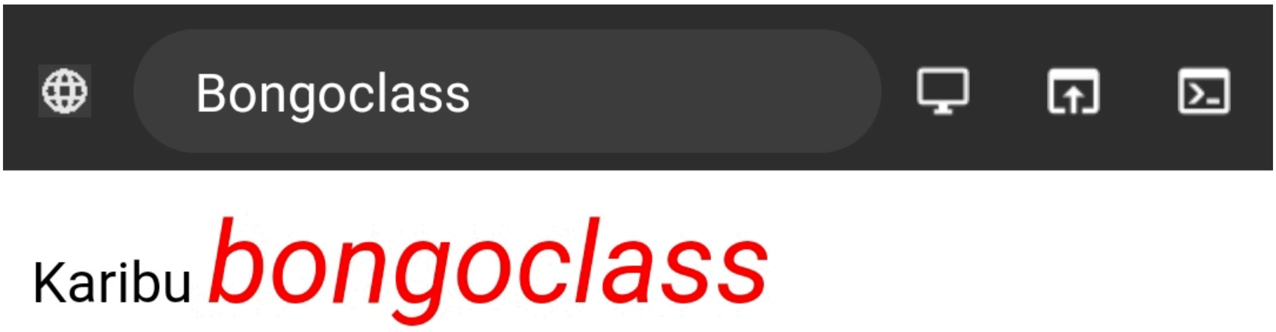
Kwa mfano unataka ku target element hizi tu (**p, u, b, i)** na unataka zote ziwe na style Moja. Hapa ndipo tutatumia group selectors

```
p,b,u,i{

  font-size: 200%;

  font-family: Arial;

  color: red;

}
```

Kwenye html weka hizi

```
<html lang="sw">

<head>
```

```
    <title>Bongoclass</title>

    <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<div>Karibu <i>bongoclass</i></div>

</body>

</html>
```



5. Universal selectors

Hii hutumika ku target element zote kwenye faili la html. Hii huandikwa kwa kuanza na alama ya nyota (*****).

```
*{

   font-size: 100%;

   font-family: Arial;

   color: red;

}
```

Mwisho:

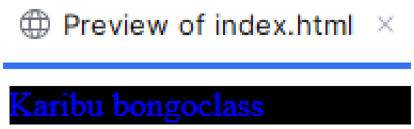
Katika somo linalofuata tutakwenda kujifunza kuhusu matumizi ya rangi.

CSS - somo la 5: Njia tano zinazotumika kuweka rangi kwenye css

Katika somo hili utajifunza aina tano za kuweka rangi kw akutumia css

Tumesha jifunza katika masomo ya nyuma jinsi ya kuweka rangi. Rangi inaweza kuwa ni kwa ajilia backround ama kwa ajii ya text. Kama tulivyojifunza huko awali, property **background** hutumika kwa ajili ya ku set rangi kwa ajili ya background, na propert ya **color** hutumika ku set rangi ya text

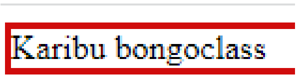
```
p{

  background: black;

  color: blue;

}
```



Tofauti na hizo mbili kuna nyingine ya kuweka rangi kwenye borders. Hii inaitwa **border**

Mfano:

```
p{

  border: solid #d00c0c;

}
```



Namna za uwekaji warangi

Katika **css** kuna aina tano za kuweka rangi kwenye faii la html. Njia hizo ni :-

1. **Kwa kutumia jina la rangi**
2. **Kwa kutumia Hex color**
3. **Kwa kutumia RGB**
4. **Kwa kutumia HSL**
5. **Kwa kutumia HSLA**

1. **Kwa kutumia jina la rangi:**

Kwa kutumia jina la rangi ni kama tulivyofanya hapo juu. Utaitaja rangi hyo kwa jina. Jambo la msingi ni kujuwa tu majina ya hizo rangi kwa ufasaha. Mfano **red, blue, black, yellow, green, purple n.k**. Nikama tulivyofanya hapo juu.

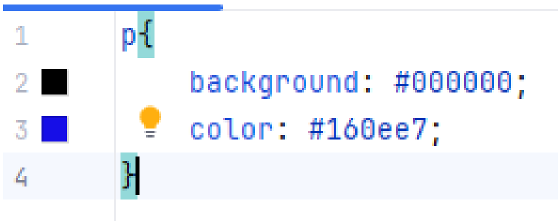
```
p{
  background: black;
  color: blue;
}
```

2. Kwa kutumia Hex color

Kuna baadhi ya rangi majina yake huyajui, saa utatumia thamani zake za hex color. Hex color zipo katika namna mbili ambzo ni kwa kutumia herufi ama kwa kutumia namba. Njia yeyote utakayotumia utaanza na alama ya reli yaani hash **#.** Baada ya kuandika # itafuatiwa na character 6 kama ni herufi ziwe 6 na kama ni namba ziwe 6.

Mfano:

```
p{
  background: #000000;
  color: #160ee7;
}
```



Unaweza kuzi generate mwenyewe kwenye text editor. Ama w3schools wanayo system ya ku generate hex color free.

3. Kwa kutumia RGB

RGB ni kifupisho cha maneno Red, Green, Blue. kwa hiyo ni kitendo cha kuchanganya rangi hizo tatu ili kupata rangi nyingine. Kwa mfano ukicanganya rangi nyekundu, ya kijani na buluu kwa uwiano sawa utapata rangi nyeupe.

```
p{
  background: rgb(255, 255, 255);
  color: rgb(0,0,0);
}
```

Kwenye css editor itaonekana hivyo

```
p{
      background: rgb(255, 255, 255);
      color: rgb(0,0,0);
}
```

Thamani za RGB huwekwa kutoka 0 hadi 255. Yeyote itakayokuwa kubwa kuliko wenzie ndio itaonekana sana. Kwa mfano kama unataka rangi iwe nyekundu tu unaweza kuweka hivi

color: rgb(255,0,0);

Kadiri itakavyopunguwa kutoka 255 kuja chini ndivyo rangi itaendelea kubadilika. Mfano kama itashuka hapo kutoka 255 kuwa 100 utaona itabadilika, kuwa nyekundu iliyofanana na weusi color: rgb(100,0,0);

```
p{
      color: rgb(100,0,0);
}
```

Baadhi ya rangi na thamani zao cheki [w3schools]

4. Kwa kutumia HSL

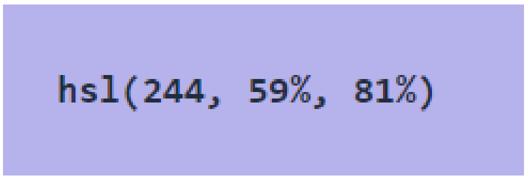
HSL ni kifupisho cha maneno Hue, Saturation, Lightness. Wacha tuone kila moja inamaanisha nini:

1. Hue ni kiwango cha rangi kutoka 0 hadi 360. Nia sawa na mzunguruko wa duara.hili ji jiduara lenye nyuzi 360, na kila nyuzi katika hilo duara linamaanisha rangi fuani. Kwa mfano rangi ya kijani ni nyuzi 120 na rangi ya buluu ni nyuzi 240. Kwaiyo hue hupimwa kwa ktumia nyuzi. Ndogo ni 0 na kubwa zaidi ni 360
2. Saturation humaanisha asilimia ya kiwango cha kuangaza kwa rangi (intensity). Yenyewe inapimwa kwa asilimia kutoka 0 hadi 100. Ikiwa 0 maana yake ni gray ambao hakuna mng’ao wa rangi, ila 100 inakuwa i full. Hii utapima je rangi ulioiweka kwa kutumia Hue unahitaji itokeze yote ama kwa kiasi maalumu. Kwahyo saturation inaeza kubadili rangi iliyowekwa hapo awali.
3. Lightness ni kiwango cha uwepo wa mwangaza kwenye hiyo rangi, nayo hupimwa kwa asilimia. Kwa mfano ikiwa ni 0 hakuna mwanga hivyo rangi itaonekana ni nyeusi, na ikiwa 100 rangi itaonekano yote.

Jinsi ya kutumia HSL:

Ili uweze kuitumia utaanaza na kuandika hsl ikifatiwa na mabano, ndani ya hayo mabano utaweka value zake.

```
p{

  color: hsl(244, 59%, 81%);

}
```



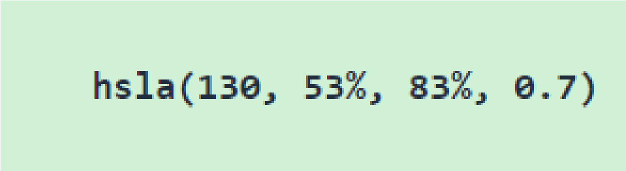
5. **Kwa kutumia HLSA**

Hii ni sawa na HLS ila hapa kumeongezeka A ambayo inamaanisha Alpha. Alpha huwakilisha kiwango cha opacity. Ili uweze kuelewa opacity chukulia mfano karatasi ukaliweka mafuta, utaona linaweza kuonyesha upande wa pili na lupitisha mwanga. Hii ndio opacity.

Alpha huandikwa kwa kiwango kutoka 0.0 hadi 1. Ikiwa ni 1 hapo inakuwa haionyeshi kabisa na ikiwa 0.0 hapo itakuwa inaonyesha kabisa.

Mfano:

```
p{

  color: hsla(130, 53%, 83%, 0.7)

}
```



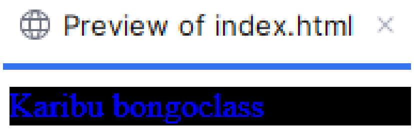
Je unahitaji kufanya mazoezi zaidi ya rangi? Cheki link hii kutoka w3schools

Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu background activities.

Tumesha jifunza katika masomo ya nyuma jinsi ya kuweka rangi. Rangi inaweza kuwa ni kwa ajilia backround ama kwa ajii ya text. Kama tulivyojifunza huko awali, property **background** hutumika kwa ajili ya ku set rangi kwa ajili ya background, na propert ya **color** hutumika ku set rangi ya text

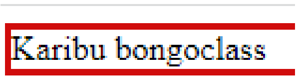
```
p{
   background: black;
   color: blue;
}
```



Tofauti na hizo mbili kuna nyingine ya kuweka rangi kwenye borders. Hii inaitwa **border**

Mfano:

```
p{
   border: solid #d00c0c;
}
```



Namna za uwekaji warangi

Katika **css** kuna aina tano za kuweka rangi kwenye faii la html. Njia hizo ni :-

1. **Kwa kutumia jina la rangi**
2. **Kwa kutumia Hex color**
3. **Kwa kutumia RGB**
4. **Kwa kutumia HSL**
5. **Kwa kutumia HSLA**

1. **Kwa kutumia jina la rangi:**

Kwa kutumia jina la rangi ni kama tulivyofanya hapo juu. Utaitaja rangi hyo kwa jina. Jambo la msingi ni kujuwa tu majina ya hizo rangi kwa ufasaha. Mfano **red, blue, black, yellow, green, purple n.k**. Nikama tulivyofanya hapo juu.

```
p{
   background: black;
   color: blue;
```

}

2. Kwa kutumia Hex color

Kuna baadhi ya rangi majina yake huyajui, saa utatumia thamani zake za hex color. Hex color zipo katika namna mbili ambzo ni kwa kutumia herufi ama kwa kutumia namba. Njia yeyote utakayotumia utaanza na alama ya reli yaani hash **#.** Baada ya kuandika # itafuatiwa na character 6 kama ni herufi ziwe 6 na kama ni namba ziwe 6.

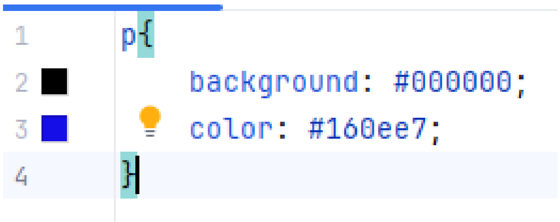
Mfano:

p{

background: #000000;

color: #160ee7;

}



Unaweza kuzi generate mwenyewe kwenye text editor. Ama w3schools wanayo system ya ku generate hex color free.

3. Kwa kutumia RGB

RGB ni kifupisho cha maneno Red, Green, Blue. kwa hiyo ni kitendo cha kuchanganya rangi hizo tatu ili kupata rangi nyingine. Kwa mfano ukicanganya rangi nyekundu, ya kijani na buluu kwa uwiano sawa utapata rangi nyeupe.

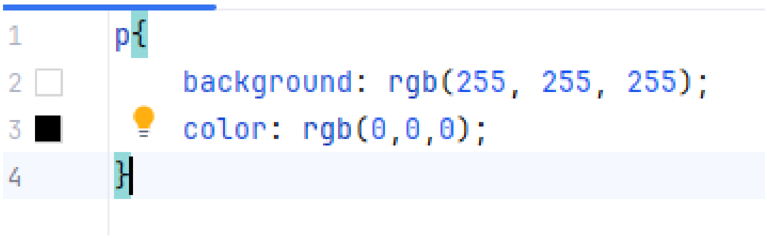
p{

background: rgb(255, 255, 255);

color: rgb(0,0,0);

}

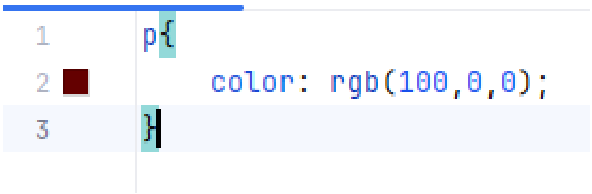
Kwenye css editor itaonekana hivyo



Thamani za RGB huwekwa kutoka 0 hadi 255. Yeyote itakayokuwa kubwa kuliko wenzie ndio itaonekana sana. Kwa mfano kama unataka rangi iwe nyekundu tu unaweza kuweka hivi

color: rgb(255,0,0);

Kadiri itakavyopunguwa kutoka 255 kuja chini ndivyo rangi itaendelea kubadilika. Mfano kama itashuka hapo kutoka 255 kuwa 100 utaona itabadilika, kuwa nyekundu iliyofanana na weusi color: rgb(100,0,0);



Baadhi ya rangi na thamani zao cheki [w3schools]

4. Kwa kutumia HSL

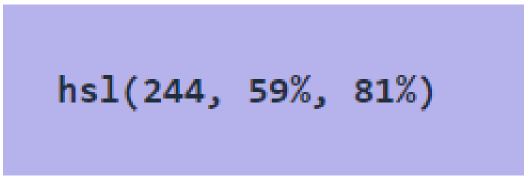
HSL ni kifupisho cha maneno Hue, Saturation, Lightness. Wacha tuone kila moja inamaanisha nini:

1. Hue ni kiwango cha rangi kutoka 0 hadi 360. Nia sawa na mzunguruko wa duara.hili ji jiduara lenye nyuzi 360, na kila nyuzi katika hilo duara linamaanisha rangi fuani. Kwa mfano rangi ya kijani ni nyuzi 120 na rangi ya buluu ni nyuzi 240. Kwaiyo hue hupimwa kwa ktumia nyuzi. Ndogo ni 0 na kubwa zaidi ni 360
2. Saturation humaanisha asilimia ya kiwango cha kuangaza kwa rangi (intensity). Yenyewe inapimwa kwa asilimia kutoka 0 hadi 100. Ikiwa 0 maana yake ni gray ambao hakuna mng'ao wa rangi, ila 100 inakuwa i full. Hii utapima je rangi ulioiweka kwa kutumia Hue unahitaji itokeze yote ama kwa kiasi maalumu. Kwahyo saturation inaeza kubadili rangi iliyowekwa hapo awali.
3. Lightness ni kiwango cha uwepo wa mwangaza kwenye hiyo rangi, nayo hupimwa kwa asilimia. Kwa mfano ikiwa ni 0 hakuna mwanga hivyo rangi itaonekana ni nyeusi, na ikiwa 100 rangi itaonekano yote.

Jinsi ya kutumia HSL:

Ili uweze kuitumia utaanaza na kuandika hsl ikifatiwa na mabano, ndani ya hayo mabano utaweka value zake.

```
p{

  color: hsl(244, 59%, 81%);

}
```



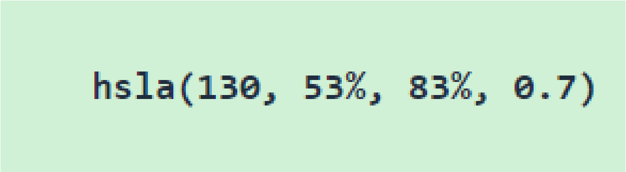
5. **Kwa kutumia HLSA**

Hii ni sawa na HLS ila hapa kumeongezeka A ambayo inamaanisha Alpha. Alpha huwakilisha kiwango cha opacity. Ili uweze kuelewa opacity chukulia mfano karatasi ukaliweka mafuta, utaona linaweza kuonyesha upande wa pili na lupitisha mwanga. Hii ndio opacity.

Alpha huandikwa kwa kiwango kutoka 0.0 hadi 1. Ikiwa ni 1 hapo inakuwa haionyeshi kabisa na ikiwa 0.0 hapo itakuwa inaonyesha kabisa.

Mfano:

```
p{

  color: hsla(130, 53%, 83%, 0.7)

}
```



Je unahitaji kufanya mazoezi zaidi ya rangi? Cheki link hii kutoka w3schools

Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu background activities.