CSS- cascading Style Sheet (style sheet language එකක්)

1994-hakon wium lie තමයි හදලා තියෙන්නේ.

ආකාර 3ට css එකක් add කරන්න පුළුවන්

1. Inline style (element එකක් ඇතුලේ style ලෙස යොදලා add කරන්න පුළුවන්.)

<h1 style=”color: red”>

1. Meta style

<style>

div{ width :100px;

Height:100px;

Background-color: red;

}

</style>

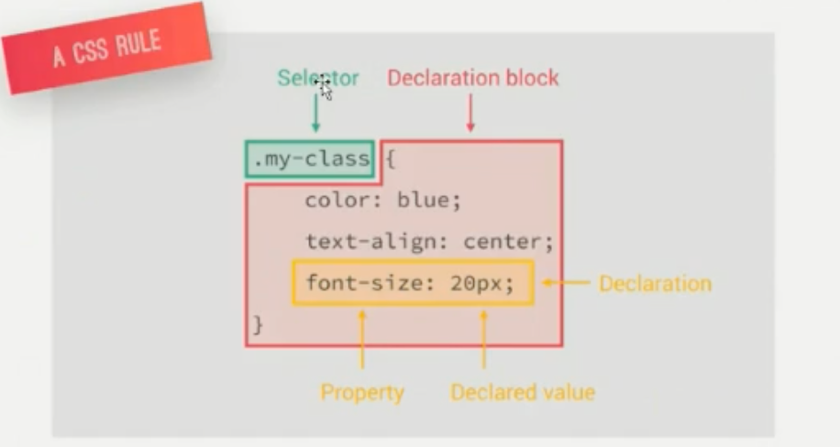
මෙලෙස div කියන element eke width එකයි height එකයි background-color එකයි දලා තියෙන විදියට වෙනස් වෙනවා.

1. Outsource style (External style)

<link rel=”stylesheet” href=”style.css”>

මෙලෙස යඑපක එකේ link කරලා අදාල විදියට code වෙනමම style.css එකේ type කරගෙන යන්න පුළුවන්

CSS rule🡪



1. Type selector 🡪

h1{

color: red;

}

h1 තියෙන හැම එකකටම මෙතනදි red add වෙනවා.

Element එකේ name එකෙන් නම්, select කරල තියෙන්නෙ එ්ක type selector එකක්.

1. Universal selector

\*{

Color: orange;

}

සේරම select කරනවා

1. Id selector

<h1 id=”si”> මේ html code එක

#si{

Color: green;

}

Html code එකේ තියෙන id එක ඇතුලෙ දලා තියෙන name එක, # එකත් සමගම ලියාගෙන select කිරීම

(මෙහිදී ids dublicate කරන්න එපා)

1. class selector

<h1 class=”thi”>

<h2 class=”thi”>

.thi{

color: blue;

}

Html code එකේ තියෙන class එකේ දලා තියෙන name එක , ‍”.” එකත් සමග ලියාගෙ select කිරීමට.

මෙය එන්න හේතුව id selector එකේ duplicate වීමේ පහසුව නැති වීමයි.

මෙහි duplicate පුළුවන්.

1. Attribute selector

[id]{

Color: pink;

}

Attribute එක පිටින්ම select කර ගෑනීමට.

1. Grouping selector

H1,h2,a{

Color: yellow;

}

H1,h2,a මේ 3ම select කිරීමට (element කිහිපයක් select කිරීමට)

1. Psudo Selectors
   1. Psudo class

H1:hover{

Color: red;

}

H1 එකට mouse එක ගියොත් ඒකෙ පාට රතු කරන්න

hover වෙනුවට එක එක ඒවා යොදන්න පුළුවන්

first-child🡪 මුල් එක select කරන්න

last-child🡪අන්තිමයා select කරගන්න.

nth-child(1)🡪 මුල් එක select කරන්න

Cosmetic elements

* h1:before{

content :”@”;

}

මෙය යෙදුවහම h1 එකට කලින් @ යෙදිය හැකිය

* h1:after{

content:”!”;

}

මෙය යෙදුවහම h1 එකට පසුව ! යෙදිය හැකිය

* h1:active{

color: red;

}

h1 එක click කරොත් ඒ h1 එකට අදාල කොටස රතු වෙනවා

(මේක click කරන්න පුළුවන් element එක්ක විතරයි use කරන්න පුළුවන්)

* a:visited{

color: orange;

}

මේක යොදන්න පුළුවන් link තියෙන ඒවට පමනයි.

ඒ කියන්නේ link එක එක පාරක් click කරලා තියෙනවනම් ඒකෙ පාට orange කරන්න කියන එකයි.

* Input:focus{

Background-color: red;

}

Input කියන element එක click කරොත් මේකෙ පාට රතු කරන්න.

(මේක හැම එකකටම යොදන්න බෑ)

b pseudo Elements

h1::first-letter{

color: blue;

}

* Pseudo element එකේ මේවා : දෙකක් තියෙනවා class එකේ එකයි තියෙන්නේ.
* H1 කියන එකේ first letter එක නිල් කරන්න
* First-letter වෙනුවට first-line ,… මේ වගේ ඒවා යොදන්න පුළුවන්.

Input::placeholder{

Color: green;

}

* Input එකේ displace කරල කියෙන words වල color එක කොල පාට කරන්න.
* Placeholder වෙනුවට file-selector-button මේකටත් යොදන්න පුළුවන්.

P::selection{

Color: black;

}

* Paragraph එක select කරනකොට ඒකෙ color එක black කරන්න.

Li::marker{

Content: @;

}

* List එකේ තියෙන එක ඉස්සරහට 1,2.. මේවා වෙනුවට @ එයි
* Getemogi.com එකෙන් ගිහින් ඒකෙ emoji අපිට copy කරගෙන මෙතනට past කරගත හැකිය.

Combinators

1. Child Combinator (grater than)

Section>h1{

Color: red;

}

Section එක ඇතුලෙ ඉන්න මුල්ම h1 ට රතු add කරන්න.

1. Descendent Combinator(space)

body h1{

Color: blue;

}

Body එක ඇතුලෙ කොහේ හරි h1 ඉන්නවනම් ඒකෙ color එක නිල් පාට කරන්න.

1. Adjacent Combinator(plus)

h1+ul {

Color: green;

}

H1 එක ලගම ul තියෙන්න ඕන තියෙනවනම් ඒක කොලපාට කරන්න.

1. General siblings Combinator

h1 ~ ul{

color: red;

}

H1 එකක් තියෙන්න ඕනෙ ඒක යටින් ul තියෙනවනම් ඒ සේරම රතු පාට කරන්න.

Cascade style sheet

එක element එකක් target කරගෙන deferent selectors වලින් same property එක deferent values දාලා ඒක apply කරන්න ගියාට පස්ස එතන ඇති වෙනවා conflict එකක් ඒ conflict එක විසදන mechanism එකට cascade එකක් කියනවා.

**!important** මේක යෙදුවහම මුලින්ම run වෙන්නෙ මේක දාන එකයි.

Ex-: යොදන ආකාරය

H1{

Color: red !important;

}

**Specificity**

1. Inline Selector. (1000)
2. ID Selector (100)
3. Class, Pseudo, Attribute (10)
4. Type Selector, Pseudo Element (1)

**Source Order**

Specificity එකේ 3) ට සහ 4) අදාලව 3ක් සහ 2ක් පවතී මේ selectors සේරම ආවොත් ප්‍රශ්නයක් වන නිසා මෙය එයි.

මෙහිදි යටටම පවතින selector එකේ ඉදන් උඩට යන්න display කරයි.

* තනි තනිව ගත් කල මේවාට direct මෙවාට අනුව display වෙයි නුමුත් combinators ආවාහම ඒවාට values දී ඇත ඒ අනුව මේවා වෙනස් වෙයි.

Ex-:

1+1

P h1{

Color: red;

}

මෙහි value එක 2යි

1 +1 + 10

Section>h1:nth-child(2){

Color: blue;

}

මෙහි value එක 12යි.

**propertys**

1. border: 5px solid red;
2. padding: 5px;
3. marging:5px;
4. width: 300px;
5. height:300px;
6. opacity: 0.3;
7. background-color: blue
8. background: green;
9. transform: rotate(200beg); 🡪 අදාල දේ rotate කර ගන්න පුළුවන්
10. border-radius: 150px; 🡪 circle එකක් හදා ගන්න පරිමිතියයි 150 px
11. font-size

Absolute Measurement Units 🡪 cm, mm , in , pc , pt ,px

Relative Mesurement Units 🡪 vw(viewport width), vh(viewport height)

Ex-: div {

Width: 90vw;

Height: 100vh;

Background-color: red;

}

Vw දැන්මහම width එක තියෙන ගානෙන් % එකක් එන්නේ. (කෙනා මත depend වෙනවා.)

Nearest element 🡪 em , rem

Ex-:

div{

Font-size: 20px;

}

div>h1{

font-size:3em;

}

div>p{

font-size: 1em;

}

මෙතනදි div එකේ main එකට සාපේක්ෂව div>h1 තුන් ගුණයක් විශාලව පෙනේ

Property

1. Inherited properties
   1. Font-size
   2. Color
2. Non-inherited properties
   1. Width
   2. Height
   3. border

Elements

1. Block level Elements 🡪 content එකට සාපේක්ෂව perant ගෙ element එකම අල්ලගෙන ඉන්නවා.
   1. h1, ul
2. Inline Elements 🡪parent ගෙ full width එකම අල්ලන්න යන්නෙ නෑ තමන්ගෙ content එකට අදාල කොටස විතරක් අල්ලගෙන ඉන්නෙ.
   1. Img
   2. Button
3. Inline-block

Block එක inline කරන ආකාරය සහ inline එක block කරන ආකාරය

1. display

Ex-: h1{

display: inline;

}

img{

display: block;

}

h1{

display: Inline-block;

}

1. position

CSS POSSITION

1. Static (Default) 🡪 වෙනස් වෙන්නෙ නෑ
2. Relative. 🡪 මේ අවස්ථාවේදී left, top දේවල් යොදලා අපිට move කරන්න පුළුවන් ඒ element එක නුමුත් අනිත් කිසිම element එකකටවත් මුකුත් වෙන්නෙ නෑ



1. Absolute 🡪 absolute දුනන්හම අදාල දෙන එකට එක වෙනම නියතයක් වෙනවා ( ආදාල element එක තියෙද්දි අනිත් ඒවා පෙලට එනවා අදාල එක ඒවාගෙ උඩට එනවා , word වල wrap text එක through දුන්න වගේ වෙනවා.)
   * + relative වල දාන ඒවත් දාන්න පුළුවන්
     + after applying absolute it is removed from the normal flow
     + then he decides his position according to the parent



1. Fixed 🡪 හැම නිතරම එකම තැන පවතින්න ඕන නම් මෙය යොදයි.

(ඒ කියන්නේ nearanse parent දිහා නෙවෙයි බලන්නෙ  viewport එකට අනුවයි.)

1. Sticky 🡪 මෙය යෙදුවහම ඒක අහුවුන තැන ඉදන්න යටට scroll කරනහොට ඒක ටිකක් වෙලා උඩට display වෙයි.

=-

1. Box-shadow

Ex-: box-shadow: 1px 2px 3px 3px black;

1. Font-family

Ex-: font-family: ‘Rubik Burned’ , cursive;

Google font වලින් අපිට ඕන font එක link කරලයි ඊට පසුව මෙය යොදන්නෙ

Z-index 🡪 යෙදුවහම අදාල එකේ දෙන ගාන අනුව z-index එකේ value එක විශාල එක මුලට එනවා.

Flex box

**Display කියන property එකට flex දාලා තියෙන්න ඔනෙ මේවා යොදන්න. Conterner එකටයි යොදන්නෙ property**

**Ex-; <section><div>**

**මේ විදියට තියෙනවනම් div එකේ වෙනස් කරන්න ඕනෙ නම් section එකටයි දාන්න ඔන.**

align-items: center; 🡪y අක්ෂය දිගේ center එකට ගෙන ඒමට (center වෙනුවට flex-start,flex-end මේවා යෙදිය හැකිය)

justify-content:center🡪 x අක්ෂය දිගේ center එකට ගෙන ඒමට (center වෙනුවට flex-start,flex-end , space-evenly මේවා යෙදිය හැකිය

flex-direction: column හරි row 🡪 row විදියට තියෙන ඒවා column කරන්න පුළුවන් column විදියට තියෙන ඒවා row කරන්න පුළුවන්.

flex-wrap: wrap ; 🡪 මේක යෙදුවහම එක line එකේ තියෙන ඒවා වැඩිනම් line එකේ තියන්න පුළුවන් ප්‍රමාණය තියල වැඩි ඒවා අනිත් line එකට ගෙනියනවා.default තියෙන්නෙ nowrap

flex-shrink:0 🡪 මේක default තියෙන්නෙ 1 මේක යෙදුවහම අදාල property එකේ size එක දෙනවා පිට පැනලා හරි.

Flex-grow: 1 🡪 මේක යෙදුවහම අදාල conterner එකේ width එකම ගන්නවා(මේකෙ default තියෙන්නෙ 0) 1 වෙනුවට 2,3,4 ... මේ වගේ යයොදන්න පුළුවන්.

Flex-basis: 300px; 🡪 width එකේ ප්‍රමානෙම ගන්නෙ නෑ 300px කර ගන්නවා.

Align-self: self-start 🡪 start කරපු තැනට යනවා ( මේ වෙනුවට self-end ද යෙදිය හැකිය)

Order:1 🡪 අදාල flex item වල order වෙනස් කරන්න පුළුවන්

**Grid layout**

display: grid; 🡪 grid layout එකට මෙය යෙදීම අනිවාර්‍යවේ.

Grid-template-rows: 1fr 1fr 1fr; 🡪 rows 3ක් අවශ්‍යනම්

Grid-template-column: 1fr 1fr 🡪 coloumn 2ක් අවශ්‍ය නම්

Column-gap:5px 🡪 column අතර gap තිබීමට

row-gap:5px 🡪 row අතර ගැප් තිබීමට

gap:5px🡪 margin එකෙන් වෙනවා වගේ.

**Css FrameWork**

මෙය එන්න හේතුව අපි CSS වලින් අමාරැවෙන් හදන දේවල්(position වගේ දේවල් දාලා) ලේසි කිරීම සදහා මෙය යොදාගනී.

Framework 🡪 commonly ඇති වෙන problem එකකට දීල තියෙන common solution එක

1. What is CSS framework
   1. Bootstrap
   2. Semantic ui
2. How to solve this problem?