

Cascading Style Sheets

Key Notes

- content HTML = House structure
- CSS content look better more presentable = House design
- color, position, effect, font size

CSS Rules/Rule section Syntax

- selectors are used to target element
- declaration of key value pair

```
div { /* selector*/  
    font-size:20px; /* declaration*/  
    margin:20px; /* declaration*/  
}  
p{ /* selector*/  
    padding:30px /* declaration*/  
}
```

adding external CSS to html

- in head or anywhere

```
<style>
  h1 {
    color:red;
  }
  p {
    color:blue;
  }
</style>
```

- not the best approach how can we fix this?
- link it to html page `<link rel="stylesheet" href="style.css">`

some styles

```
html {  
  padding: 50px;  
  margin: "auto";  
  scrollbar-width: "none";  
  background-color: rgb(252, 241, 224);  
  font-family: 'Roboto', sans-serif;  
  text-align: left;  
  letter-spacing: 0.25px;  
}  
  
h1 {  
  color: saddlebrown;  
  text-decoration: underline;  
  font-family: 'Radio Canada', sans-serif;  
  text-align: center;  
}
```

some styles to add - 2

```
ul, ol {
  color:darkgoldenrod;
  /* border-width:2px;
  border-style: dashed;
  border-color:cadetblue */
  border: 4px solid darkblue;
  /* border-bottom:8px dotted red;
  border-left:3px dashed darksalmon */
}
li{
  /* the disk in li tag */
  list-style-type: none;
  text-shadow: 2px 2px white;
  /* 3 values right bottom colors */
}
p {
  column-count: 2;
  column-gap: 20px;
```

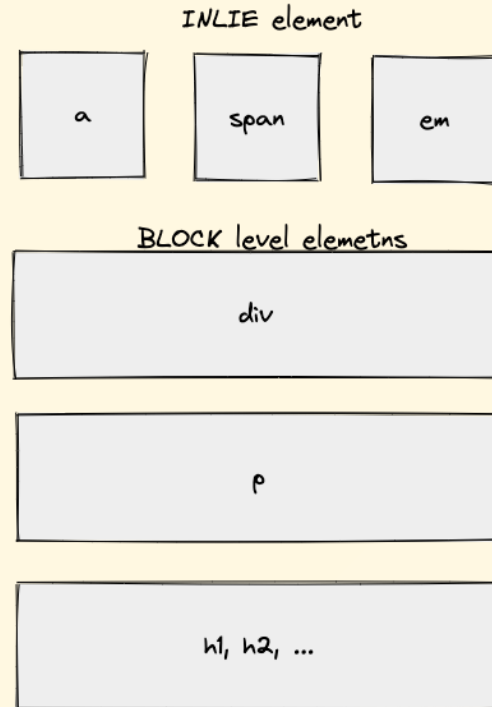
custom colors

RGB-hex code

- red blue green
- 0 darkest f whites
- #440000 would be maroon i.e dark red
- #EE0000 would be red i.e light red
- YOU WILL NAVER MAKE YOUR OWN HEX COLOR
- use vscode :P

inline/block html element

- line up next to each other
- take up whole width
- block-> we want a link without a new line `<p> sdad <a>link </p>`
- change inline to block, block to inline using css :)



Box-Model

- How element represents themselves in terms of space
- we can control this spacing
 - margin
 - padding
 - border



css basic summary

- what is it how we added it
- basic selectors
- hex code and vscode colors
- inline and block element
- margin and padding
- default styles by browses

Classes and Sector

class selection

- select a class with `.` to style

```
<div class="coolClassName">some text</div>
<style>
  .coolClassName {
    color:red;
  }
</style>
```

- only get specific element with the class
- we do using `tag.coolClassName` `p.coolClassName`

multiple classes

- style element with multiple class name using

```
div.coolClassName.coolClassName2
```

```
<div class="coolClassName coolClassName2">
```

id selection

- we can use classes or ids
- more commonly used in JS
- target the content using #
- can be only used ONLY ONCE, HAS TO BE UNIQUE

```
<div id="coolId"> some text </div>
<style>
  #coolId {
    color:red;
  }
</style>
```

- `parent child {declaration}`
- this is specific targeting
- if we want to target a specific class in an element b

```
<div>
  <div class="child">
    <p>some text</p>
  </div>
</div>
<p class="child">
  <span>some text</span>
</p>
<style>
  div .child {
    color:red;
  }
</style>
```

- selecting with and without space

attribute selectors

We want to select links with href Attribute

```
a [href] {color:darksalmon}
```

or with some values

```
a[href="https://www.google.com"] {color:darksalmon}
```

“ ***Tips:***

- You can edit default link underline property with
`text-decoration: none`
- instead of looking up whole link you can search a keyword
`a[href*="google"] {color:darksalmon}`
- or ends in with `a[href$=".pdf"] {color:darksalmon}`

”

The CASCADE

- go down like waterfall
- html children can inherit parent styles
- p inherits div style but not all, we can force inherit them
- last one has precedence over the previous one unless its not specifically Set

```
<div>
  <p>
    <span>some text</span>
  </p>
</div>
<style>
  div {
    color:red;
    border:2px solid red;
    font-style: italic;
    margin:20px;
    font-size:20px;
  }
  p {
    border:inherit;
    margin:inherit;
  }
</style>
```

Layout and position

- Static : default position
- Relative: can position relative to other elements or its original position
 - we do this position child absolute to the parent
- Absolute: positioned within relative loses original space
- Fixed: position relative to the viewport always fixed
- Sticky: position relative to the viewport always fixed and can be fixed to top or bottom based on scroll

```
<section class="banner">
  
  <div class="welcome">
    <h2>Static page to Learn/Teach <br><span>HTML & CSS</span></h2>
  </div>
</section>
<style>
  .banner {
    position: relative;
  }
  .banner .welcome {
    position: absolute;
  }
</style>
```

The nav bar

- `position:fixed` with `top:0px` and `left:0px`
 - it loses its width, set `width:100%`
- the `z-index` everything has a z-index of 0

```
header {  
  background-color:darksalmon;  
  padding: 2px 15px;  
  width: 100%;  
  top:0;  
  left:0;  
  position:fixed;  
  z-index: 1;  
}
```

position sticky

```
nav{  
background-color: rgba(63, 61, 122, 0.891);  
position: sticky;  
padding: 10px;  
width: 100%;  
top:80px  
}
```