Intermediate CSS

16/09/2022.

7.0 => The Cascade - Specificity & Inheritance :
How Competing styles are Resolved ?

CSS → Cascading Style Sheet

Cascading :- if there are 2 CSS targeting single html element then there is a hierarchy

①CSS ②CSS

will be displayed ✓ → h1 ← ✗ will not be shown/displayed.

What actually happens is both CSS are applied to that particular element. but ①CSS is overlapping ②CSS. just like a cascade.

(X) usually external style sheet is applied first and it is overlapped by internal / inline style sheet if it is there.

Categories of Level of Importance :- [CSS rule]

① Position : [lower position - higher hierarchy]

```
li {
   color: red ;  → applied first
   color : blue ; → overlapped next
}
```

② Specificity : [How specific a selector is ?]

```
li { color : blue ; } → element specifying
.first-class { color : red ; } → class specifying
li [draggable] { color: purple ; } → attribute "
# first-id { color: Orange ; } → id specifying
```

③ Type :- [Type of CSS]

```
<link rel="stylesheet" href="./style.CSS"> → using external css
<style> </style>
<h1 style=" " > Hello </h1> → but overlapped by internal CSS.
```

④ Importance :

```
color : red ;
color: green !important ; → keyword is prioritized
```

71. Combining CSS selectors :-

How to target specific elements to style?

Types of Combiners :-

① Grouping Combiners :-

```
selector, selector {
      do something ;
}
```

Example :-

```
h1, h2 {
    color : blue ;
}
```

② Child Combiners :-

```
selector (Parent) > selector (child) {
      do something ;
}
```

Example :-

```
div > p {
     font : 20px ;
}
```

③ Descendant Combiners :-

```
selector (ancestor) selector(descendant) {
      do something ;
}
```

Example :-

```
.box .list {
      color : violet ;
}
```

④ chaining selector Combiners :-

```
selectorselectorselector ... {
   do something ;
}
```

Example :-

```
li.done#id-name {
  font : 10 px ;
}
```

⑤ Combining Combinors :-

```
selector(anchestor)  selector(chain) {
   do something ;
}
```

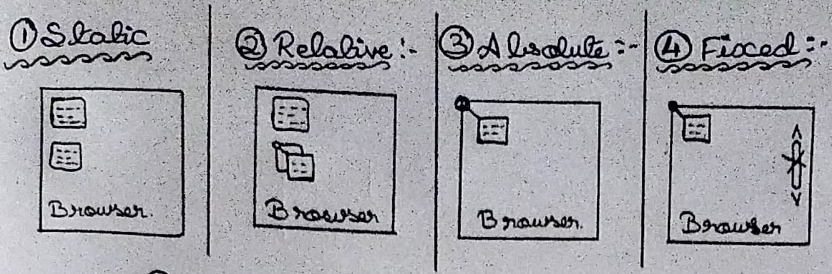
Example :-

```
ul p.done {
   color : red ;
}
```

7.1. CSS Positioning :-

-> Relative, absolute, static and Fixed positioning.

-> used for Positioning elements on screen.



① Static :-

-> HTML Default flow.

Example :-

```
.class.class {
    position : static; [Default one]
    left : 10px;
    top : 10px;
}
```

② Relative :-

-> Position Relative to default position.

Example :-

```
.class.class {
    position : relative;
    left : 10px;
    top : 10px;
}
```

③ Absolute Positioning :-

-> Position relative to nearest positioned ancestor or top left corner of webpage.

Example :-

```
.div1.div2 {
    position : absolute;
    top : 10px;
    left : 10px;
}
```

Note :-

z-index : -1;

makes the element behind all element[z-axis]

④ Fixed :-

-> Position relative to top left corner of browser window. [no change in position even scrolled down]

Example :-

```
.class .class {
    position : ~~absolute~~ fixed;
    top : 10px;
    left : 10px;
}
```