LEARN CSS

I) CSS Basics

1) Anatomy of a CSS Rule :

- This is a basic syntax of a CSS Rule :
  selector { property : value ;}
- A CSS Rule can consist of multiple **declarations** (properties and values).
- The collection of CSS Rules is called **stylesheet.**

2) Element, Class and ID selectors:

a) Element Selector :
selector { property : value ;}

→ Every element in the HTML document, having this **name**, will have the property : value

→ In the HTML document: < selector >

b) Class Selector :
.selector { property : value ;}

→ Every element in the HTML document, having this **class**, will have the property : value

→ In the HTML document: < elt class = "selector" >

→ In the HTML document: Elements could have the same class

c) ID Selector :
#selector { property : value ;}

→ Every element in the HTML document, having this **ID**, will have the property : value

→ In the HTML document: < elt ID = "selector" >

→ In the HTML document: Elements could not have the same ID

d) Grouping Selectors :
Selector1 , **.selector2** , **#selector3** { property : value ;}

3) Combining selectors:

a) Element with class Selector :
elt.selector { property : value ;}

→ Every **elt** element in the HTML document, having this **class**, will have the property : value

b) Child Selector :
elt1 > elt2 { property : value ;}

→ Every **elt2** element in the HTML document, inside the **el1** element (direct child), will have the property : value

c) Descendant Selector :
elt1 elt2 { property : value ;}

→ Every **elt2** element in the HTML document, inside the **el1** element (at any level), will have the property : value

4) Pseudo-Class selectors:

a) Link and Visited :

a:**link** { property : value ;}

→ The **a** element in the HTML document, when not visited, will have the property : value

a:**visited** { property : value ;}

→ The **a** element in the HTML document, when visited, will have the property : value

b) Hover and Active :

elt:**hover** { property : value ;}

→ The element in the HTML document, when hovered over, will have the property : value

elt:**active** { property : value ;}

→ The element in the HTML document, when clicked on and not released, will have the property : value

c) N-th Child:

elt:**n-thchild(**k**)** { property : value ;}

→ The k^{th} element in the HTML document, inside the **elt** element, will have the property : value

→ if **k=odd** : The odd elements in the HTML document, inside the **elt** element, will have the property : value

→ if **k=even** : The even elements in the HTML document, inside the **elt** element, will have the property : value

II) CSS Conflict Resolution :

1) Style Placement

a) Inline CSS:

```
<elt style=" property : value ;">Content.</elt>
```

b) Internal CSS:

```
<style>
        selector { property : value ;}
</style>
```

c) External CSS:

In the HTML document: `<link rel="stylesheet" href="index.css">`

In the CSS document: selector { property : value ;}

2) Conflict Resolution:

a) Origin:

- Conflict: Last declaration wins
- No conflict: Declarations merge

b) Inheritance:

If we specify a property on an element, all the element's childs have that property too

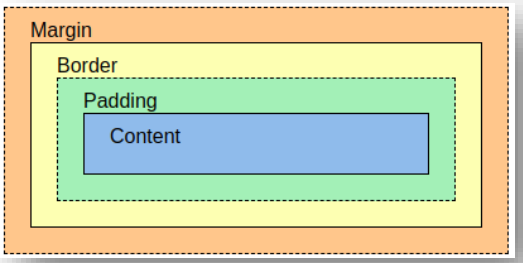
c) Specificity:

The selectors with the most score win:

- Style in HTML: 1000
- ID: 100
- Class: 10
- Element: 1

III) Box Model and Layout :

1) Box Model:



a) Box components:

- Each box consists of: **Content-Padding-Border-Margin**
- The body tag in the html doc has a **margin** of 8px by default

b) Width property:

- The **width** property changes the value of the **content** box → **Real size** : content+2*padding+2*border
- To change the size of the **whole** box, we use **border-box.**

*{ box-sizing : border-box ;} → Apply Box-sizing to all boxes.

- If we specify box width and height with **%** it becomes **flexible**

c) Cumulative and collapsing Margins:

- Cumulative: Margins of boxes on the same lines **add up**
- Collapsing: Biggest Margin of boxes on top of each other **persists**.

d) Content overflow :

Overflow property is used when **content box height** exceeds **box height**

Overflow : Visible → Box by default
Overflow : Hidden → Clip the content where the box ends
Overflow : Auto →Adds scrollbar on the box
Overflow : Scroll →Adds scrollbar both ways on the box

2) Background Property:

a) Color:

Background-color : blue → Fills the box with color

b) Image:

Background-img : url("img.jpg")→ Fills the box with image
Background-repeat : norepeat → Repeats (or doesn't) repeat the image
Background-position : up left → Chooses position of the box in the image

3) Float Positioning:

Float : left → Float the box.
Clear : left → Clears the floating.

4) Relative and Absolute Positioning:

element {
position : **relative** ; → Element moves 50px from **top**(offset) of its original position (still in document flow).
top : 50px;}

element {
position : **absolute** ; → Element moves 50px from **top**(offset) of its original position (out of document flow).
top : 50px;}

- If we have a **relative container**, absolute positioning applied to the **container** not the HTML document.

Ⅳ)Responsive Design & Bootstrap:

1) Media Queries:

a) Syntax:

@media (feature : value) {
    selector { property : value ;} → If **True**, styles within curly braces apply.
}

b) Common features:

max-width(px) **min-width**(px) **orientation**(portrait or landscape)

c) Combination of media features :

@media (feature1 : value1) and (feature2 : value2) {
    selector { property : value ;} → feature1 **AND** feature2
}

@media (feature1 : value1) , (feature2 : value2) {
    selector { property : value ;} → feature1 **OR** feature2
}

2) Bootstrap & Jquery:

a) CSS Declaration:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<link rel="stylesheet" href="css/index.css">
```

b) JS Declaration:

```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
< script src="js/ index.js"></script>
```

c) Bootstrap classes:

< div class="container"> → Creates a Container
< div class="row"> → Creates a Row (Requires container)
< div class="col-sz-nbr"> → Creates a Column (Requires row)

< nav class="navbar navbar-default "> → Creates a Navbar
< div class="navbar-header"> → Creates a Navbar Header (Requires container)
< div class="navbar-toggle"> → Creates a Navbar Toggle (Requires container)

< p class="text-center"> → Centers the text
< div class="pull-left "> → Floats the element to the left
< div class="visible-md "> → Element is **visible** only when screen in medium

https://www.w3schools.com/bootstrap/bootstrap_ref_all_classes.asp