CSS

2.1 Basics of CSS

2.2.1 CSS introduction

- CSS (Cascading Style Sheets) is a stylesheet language used to control the presentation and layout of HTML documents. It plays a crucial role in creating visually appealing and user-friendly websites.

2.2.2 Ways of Adding CSS to HTML

External Style Sheets

- One of the key concepts in CSS is the use of external style sheets. External style sheets are separate CSS files with a .css extension that contain all the styles for a website.
- To use External Stylesheet, we have to use the ‘link’ tag
- Eg:

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

Internal Style Sheets

- We can use internal style sheets within an HTML document.
- An internal style sheet is defined inside the <style> element in the <head> section of the HTML file.

- Eg:

```
<style>
        .container{
         Color : white;
         Background-color : black;
         }
</style>
```

Inline Style

- Inline styles are applied directly to individual HTML elements using the 'style' attribute.
- This method allows us to apply styles to a single element without needing an external or internal style sheet.
- It should not be used for applying long CSS effects to avoid clutter.
- Eg:

```
<h1 style=” color: Red; ”> Hello, This is Raj </h1>
```

2.2.3 CSS Syntax

- CSS Syntax Consists of ‘selector’ followed by a set of CSS properties to apply different kinds of visual effects and formatting.
- Eg:

```
.container{
        Margin : 0 auto;
        Padding : 10px;
}
```

2.2.4 CSS Selector

- Selectors are used to target specific HTML elements that we want to style.
- There are various types of selectors, such as element selectors, class selectors, ID selectors, Child selectors, Pseudo-Class Selectors etc.
- **Element selector**
    - The element selector targets HTML elements based on their tag name.

```
h1{
color: red;
}
```

- **Class Selector**
    - Class selectors target elements with a specific class attribute.

```
.form-control{
        margin: 20px;
        width: 100%;
}
```

- **ID selectors**
    - ID selectors target a single element with a unique ID attribute.

```
#heading{
font-size : 50px;
}
```

- **Child selectors**
    - The child selector targets elements that are direct children of a specific parent element.
    - It applies styles only to immediate children, not to nested elements.
    - Here, all the child elements (li) of parent(ul) will get affected.

```
ul > li {
  font-size : 20px;
}
```

- **Descendant Selector**
    - Descendant selectors target elements that are descendants of a specific parent element.

```
p em {
  font-style: italic;
}
```

Note: If you only want to target or select direct children of an element, you should use Child Combinator, if you want to select all direct and indirect children of HTML element, you should use Descendant Combinator.

- **Pseudo-Class Selectors**
  - Pseudo-classes target elements in a specific state or position.
  - They are preceded by a colon ":".

```
.github-link:hover{
    text-decoration: underline;
}
```

2.2.5 CSS basic properties

- **Color**
  - The color property sets the text color of an element

```
h1 {
  color: red;
}
```

- **margin**
  - The margin property sets the space around an element outside its border.

```
.container {
  margin: 10px;
}
```

- **padding**
  - The padding property sets the space between the element's content and its border.

```
.container {
  padding: 20px;
}
```

- **Font-size**
    - The font-size property sets the size of the text. It can be specified in pixels, em, rem, percentages, etc.

```
p {
  font-size: 15px;
}
```

- **Background-color**
    - The background-color property sets the background color of an element.

```
body{
  background-color: black;
}
```

- **border**
    - The border property sets the border around an element.
    - It includes three sub-properties: border-width, border-style, and border-color.

```
.avatar {
border: 2px solid white;
}
```

- **width**
    - The width property sets the width of an element.
    - It can be specified in px, %, vw(viewport width), etc.

```
.container {
  width: 50%;
}
```

- **height**
    - The height property sets the height of an element.
    - It can be specified in px, %, vw(viewport width), etc.

```
.avatar{
  height: 50px;
}
```

- **text-align**
    - The text-align property aligns the text within an element horizontally.

```
p {
  text-align: center;
}
```

- **font-family**
    - The font-family property sets the font family for text.

```
body {
  font-family: 'Arial', sans-serif;
}
```

- **display**
    - The display property defines how an element should be displayed.
    - **‘display: block’**
        - The element will be displayed as a block-level element, taking up the full width available and starting on a new line.

```
p {
  display: block;
}
```

- **‘display: inline’**
  - The element will be displayed inline, taking up only as much width as necessary and allowing other elements to appear on the same line.

```
span {
  display: inline;
}
```

- **‘display: inline-block’**
  - The element will be displayed as an inline-level element but will allow setting width and height properties like a block-level element.

```
button {
  display: inline-block;
  width: 100px;
  height: 40px;
}
```

- **‘display: none’**
  - The element will be completely hidden and won't take up any space on the page.
  - Note: ‘Visibility: Hidden’ hides the element but takes up the space for it, which is not the case with ‘display:none’

```
.avatar{
display: None;
}
```

- **Float**
  - The float property moves an element to the left or right of its container

```
.github-link {
        float: right;
}
```