

## ASSIGNMENT-2

### Exploring CSS styling

Submitted By:  
Himanshu Uniyal

CSS or Cascading Style Sheets is instrumental in designing visually appealing & responsive web pages.

#### CSS styling techniques

##### 1) Element type selector

- Selects all elements of a specific type
- Syntax: element { property: value; }

e.g.

```
p {  
  color: red;  
}
```

##### 2) Class Selector

- Selects elements with specific class attribute
- Syntax: .classname { property: value; }

e.g.

```
.highlight {  
  background-color: yellow;  
}
```



### 3) ID selector:-

- Selects a single element with a specific ID attribute

• Syntax: `#idname { property: value; }`

e.g.

```
#header {  
    font-size: 24px;  
}
```

### 4) Descendant Selector:

- Select elements that are descendants of specified element

• Syntax: `parent child { property: value; }`

e.g.

```
div p { font-weight: bold;  
}
```

### 5) Adjacent Sibling Selector

- Selects an element that is immediately preceded by a specified sibling element.

• Syntax: `element + adjacent { property: value; }`

e.g. `h2 + p {`

```
    margin-top: 20px;  
}
```



## 6) Attribute Selector:-

- Selects elements based on their attributes.
- Syntax: [attribute = value] { property: value; }

e.g

```
input [ type = "text" ] {  
    border: 1px solid #ccc;  
}
```

## 7) Pseudo-classes:-

- Selects elements based on their state or position.
- Syntax: selector: pseudo-class { property: value; }

e.g,

```
a: hover  
{  
    color: blue;  
}
```

## 8) Box model

- Defines the layout of elements including padding, border, margin & content.
- Syntax: box-model-property: value;



e.g. `box {`

`padding: 10px;`

`border: 1px solid #000;`

`margin: 20px;`

`}`

9) Flexbox:

- Provides a flexible way to layout elements in a container, aligning them horizontally or vertically

- Syntax: `display: flex;`

e.g. `container {`

`display: flex;`

`justify-content: center;`

`align-items: center;`

`}`

10) Grid layout

- Enables creation of complex layouts by dividing page into rows & columns.

- Syntax: `display: grid;` & other properties

e.g. `grid-container`

`{`

`display: grid;`

`grid-template-columns: 1fr 1fr 1fr;`

`grid-gap: 10px;`

`}`