- CSS can use various types of selectors
- The primary selectors used in styles are:
    - o Type Selector
    - o ID Selector
    - o Class Selector

Type Selector

- Type selector refers to HTML element tag name [Image <img>, Bold <b>].
- The given styles will be applied to specified tag where ever it is used in page.
- It will apply effects to every occurrence of the tag in page. You can't disable for any specific.

Ex:

```
<!DOCTYPE html>

<html>

  <head>

    <style>

      h2

      {

        background-color:red;

        color:white;

        text-align:center;

      }
```

```
     </style>

  </head>

  <body>

    <h2>HTML</h2>

    <p>It is a markup language.</p>

    <h2>CSS</h2>

    <p>Defines styles of HTML.</p>

    <h2>JavaScript</h2>

    <p>Handles client side interactions.</p>

  </body>

</html>
```

ID Selector

- Every element can be defined with ID.
- You can use ID to access the element and apply effects.
- You can choose to which element you want the effects.
- Element is defined with ID
  <div id="effects"> </div>
- You can access the ID in styles by using "#" reference
  <style>
  #effects
  {
  }
  </style>
- Every tag can use only one ID reference.

- If you have configured multiple categories of styles with ID selector and want to use for specific tag, then it is not possible to define all effects to one element.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #textEffects {
        text-align: center;
        color:yellow;
      }
      #bgEffects {
        background-color: red;
      }
    </style>
  </head>
  <body>
    <h2 id="textEffects">HTML</h2>
    <p>It is a markup language.</p>
    <h2 id="bgEffects">CSS</h2>
    <p>Defines styles of HTML.</p>
```

```
    <h2>JavaScript</h2>

    <p>Handles client side interactions.</p>

  </body>

</html>
```

Class Selector

- A class selector is defined by using ".".
- Class is accessed and applied to element by using "class" attribute.
- Every tag can implement multiple classes.
- Multiple classes are specified with space.

```
<style>
.cssClassName
{
}
</style>
<div  class="cssClassName1  cssClassName2"> </div>
```

- The CSS selectors are further classified into various groups based on behaviour
    - **Combinators / Rational Selectors**
    - **Attribute Selectors**
    - **Pseudo Selectors**
    - **Structural Pseudo Selectors**

Rational or Combinators

- These selector default with parent and child elements as well as with elements that have relation.
- Relation like adjacent, below, above, before, after, first, last etc..

Selector	Description

Descendent Selector	Targets all tags under specified parent. It includes any level hierarchy.
	
	It defines the parent element and the child element by using space.
	
	Syntax:
	
	 parentElement childElement {
	
	 }
	
	Ex:
	
	<!DOCTYPE html>
	
	<html>
	
	  <head>
	
	    <style>
	
	      ol li {
	
	        color: red;
	
	      }
	
	      div p {

```html
        color:green;
      }
    </style>
  </head>
  <body>
    <h2>Web Technologies</h2>
    <ol>
      <li>HTML
        <ol>
          <li>Void Elements</li>
          <li>Normal Elements</li>
        </ol>
      </li>
      <li>CSS</li>
      <li>JavaScript</li>
    </ol>
    <div>

<blockquote>Blockquote...</blockquote>
      <p>Para-1</p>
      <div>
        <p>Para-2</p>
```

| | ```
      </div>
    </div>
    <p>Para-3</p>
  </body>
</html>
```<br><br>**Output:**<br><br>## Web Technologies<br><br>1. HTML<br>    1. Void Elements<br>    2. Normal Elements<br>2. CSS<br>3. JavaScript<br><br>Blockquote...<br><br>Para-1<br><br>Para-2<br><br>Para-3 |
| Child Selector | It applies effects only to the direct child of parent element.

Syntax:

Parent > child {

} |

	Ex:
| | ```html
<!DOCTYPE html>
<html>
  <head>
    <style>
      div>p {
        color:red;
      }
    </style>
  </head>
  <body>
    <div>
      <p>Para-1</p>
    </div>
    <div>
      <span>
        <p>Para-2</p>
      </span>
    </div>
  </body>
</html>
``` |

	Output: Para-1 Para-2

Adjacent Sibling	It defines effects to an element which is specified immediately after current element.
	
	It is not parent and child, it is one below another.
	
	It will apply only to the first adjacent element.
	
	Syntax:
	FirstElement + adjacentElement
	{
	}
	Ex:
	<!DOCTYPE html>
	<html>
	  <head>
	   <style>
	     h2+p {

| | ```
      color:red;
    }
  </style>
</head>
<body>
  <h2>HTML Elements</h2>
  <p>Para-1</p>
  <p>Para-2</p>
  <p>Para-3</p>
  <p>Para-4</p>
</body>
</html>
```<br><br>**Output:**<br><br>**HTML Elements**<br><br><span style="color:red">Para-1</span><br>Para-2<br>Para-3<br>Para-4 |
| General Sibling | It defines effects to all elements which are specified after the current element. |

Syntax:

FirstElement ~ AdjacentElements

{

}

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h2~p {
        color:red;
      }
    </style>
  </head>
  <body>
   <h2>HTML Elements</h2>
   <p>Para-1</p>
   <p>Para-2</p>
   <p>Para-3</p>
   <p>Para-4</p>
```

<table>
<tr><td></td><td>

    </body>

</html>

Output:

HTML Elements

Para-1

Para-2

Para-3

Para-4

</td></tr>
</table>

Attribute Selectors

- Several elements in HTML are presented by using attribute of tag.

  <input type="button">

  <input type="radio">

- "type" is attribute.

- We have to apply effects based on attribute and value.

  Syntax:

  tagName["attribute"] { }

  tagName["attribute=value"] { }

Ex: Attribute and Value

```html
<!DOCTYPE html>
<html>
  <head>
    <style>
      input[type="button"] {
        background-color: lightgreen;
      }
      input[type="password"] {
        background-color: lightpink;
      }
    </style>
  </head>
  <body>
    <form>
      <dl>
        <dt>Name</dt>
        <dd><input type="text"></dd>
        <dt>Password</dt>
        <dd><input type="password"></dd>
```

```
        </dl>
        <input type="button" value="Register">
      </form>
    </body>
</html>
```

Output:

Name

Password

Register

Ex: Only Attribute

```
<!DOCTYPE html>
<html>
    <head>
      <style>
        p[id] {
            color: red;
        }
      </style>
    </head>
```

```
<body>

    <p>Para-1</p>

    <p id="p2">Para-2</p>

    <p>Para-3</p>

    <p id="p4">Para-4</p>

    </body>

</html>
```

Output:

Para-1
Para-2
Para-3
Para-4

- Attribute selectors can be defined with conditions.
- Effects are applied only to attribute that match the given condition.

Condition	Purpose
[attribute="val"]	Equal specifies that it should be exact match. Ex: <!DOCTYPE html>

```
<html>
  <head>
    <style>
      p[class="Effect"] {
        color:red;
      }
    </style>
  </head>
  <body>
    <p class="paraEffect">Para-1</p>
    <p class="para">Para-2</p>
    <p class="Effectpara">Para-3</p>
    <p class="Effect">Para-4</p>
  </body>
</html>
```

Para-1

	Para-2 Para-3 Para-4

[attribute**^=**"val"]	It refers the value starting with specified term.
	
	Ex:
	
	<!DOCTYPE html>
	
	<html>
	
	   <head>
	
	    <style>
	
	     p[class^="Effect"] {
	
	      color:red;
	
	     }
	
	    </style>
	
	   </head>
	
	   <body>
	
	    <p class="paraEffect">Para-1</p>
	
	    <p class="para">Para-2</p>

	`<p class="Effectpara">Para-3</p>`
	
	`<p class="Effect">Para-4</p>`
	
	`</body>`
	
	`</html>`
	
	Para-1
	Para-2
	Para-3
	Para-4

[attribute$=”val”]	It specifies that the value ending with given term.
	
	Ex:
	
	`<!DOCTYPE html>`
	
	`<html>`
	
	`<head>`
	
	`<style>`
	
	`p[class$="Effect"] {`
	
	`color:red;`
	
	`}`

| | ```html
</style>
</head>
<body>
<p class="paraEffect">Para-1</p>
<p class="para">Para-2</p>
<p class="Effectpara">Para-3</p>
<p class="Effect">Para-4</p>
</body>
</html>
```<br><br><span style="color:red">Para-1</span><br>Para-2<br>Para-3<br><span style="color:red">Para-4</span> |
| [attribute*="val"] | It matches the term at any location.

Ex: |

```html
<!DOCTYPE html>
<html>
  <head>
    <style>
      p[class*="Effect"] {
        color:red;
      }
    </style>
  </head>
  <body>
    <p class="paraEffect">Para-1</p>
    <p class="para">Para-2</p>
    <p class="Effectpara">Para-3</p>
    <p class="Effect">Para-4</p>
  </body>
</html>
```

	Para-1 Para-2 Para-3 Para-4

[attribute|="val"]	Name starts with specified term and separated with "-".
	
	Ex:
	
	\<!DOCTYPE html\>
	
	\<html\>
	
	  \<head\>
	
	   \<style\>
	
	    p[class|="Effect"] {
	
	     color:red;
	
	    }
	
	   \</style\>
	
	  \</head\>
	
	  \<body\>
	
	   \<p class="para-Effect"\>Para-1\</p\>
	
	   \<p class="para"\>Para-2\</p\>

	<p class="Effect-para">Para-3</p> <p class="Effect">Para-4</p> </body> </html> **Output:** Para-1 Para-2 Para-3 Para-4

[attribute~="val"]	Name start with specified term and contain blank space.
	
	 Ex:
	
	 <!DOCTYPE html>
	
	 <html>
	
	   <head>
	
	     <style>
	
	       p[class~="Effect"] {
	
	         color:red;

```html
        }
    </style>
  </head>
  <body>
    <p class="para-Effect">Para-1</p>
    <p class="Effect para">Para-2</p>
    <p class="Effect-para">Para-3</p>
    <p class="Effect">Para-4</p>
  </body>
</html>
```

Para-1

Para-2

Para-3

Para-4