

## CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and feel of a document written in markup language such as HTML and XML.

CSS is used with HTML and javascript in most websites to create user interfaces for web applications and user interface for many mobile applications.

Using CSS we can change the old looks with new looks and can completely change the look of the website with only few changes in CSS code.

### Benefits of CSS

1. Separable content and presentation
2. Supports code reusability
3. Simplifies updates and maintenance
4. Promotes browser compatibility
5. Enhances accessibility for all users

### Types of CSS

CSS is added to HTML pages to format the document according to presentation in style sheet.

There are three ways to include CSS in HTML documents:

#### 1) Inline CSS

It is used to apply CSS on a single element.

```
<h1 style="color: blue; font-size: 24px;">Hello, world</h1>
```

#### 2) Internal CSS

It is used to apply CSS on a single document or web page. All the elements can be affected. It is written inside the `<style>` tag within the `<head>` tag.

`<style>`

`h1 {`

```
    color: blue;  
    font-size: 18px;
```

`}`

`p {`

```
    color: red;  
    font-size: 24px;
```

`}`

`</style>`

`</head>`

```
<html>  
<body>  
  <h1> Hello </h1>  
  <p> Paragraph </p>  
</body>
```

### 3) External CSS

It is stored in separate CSS file with .css extension and link to HTML file using `<link>` tag.  
This approach is used to apply style to multiple HTML files.

```
style.css  
h1 {  
  color: blue;  
  font-size: 24px;  
}  
  
p {  
  color: red;  
  font-size: 16px;  
}
```

```
index.html  
<head>  
  <link rel="stylesheet" href="style.css">  
</head>  
<body>  
  <h1> Hello </h1>  
  <p> Paragraph </p>  
</body>
```

```
<html>  
<head>  
  <title> CSS Example </title>  
  <style>  
    h1 {  
      color: blue;  
      font-size: 24px;  
    }  
    </style>  
  <link rel="stylesheet" href="style.css" />  
</head>  
<body>  
  <h1> Example of Internal CSS </h1>  
  <p> Example of External CSS </p>  
  <p style="color: red; font-size: 16px;">  
    Example of Inline CSS </p>  
</body>  
</html>
```

```
style.css  
h1 {  
  color: green;  
  font-size: 16px;  
}
```

PAGE No.	
DATE	

PAGE No.	
DATE	

### CSS Selector

CSS selector are used to select the content that you want to style.

CSS selector select the HTML element acc. to its id, class, type, attribute etc.

#### Types of selector in CSS are

##### i) Element Selector

This selector targets elements based on their tag name

```
p {  
    color: blue;  
}
```

This will select all `<p>` elements and sets the color to blue

##### ii) Class Selector

This selector targets elements with a specific class attribute.

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

This will select all elements with `class="highlight"` and set their background color to green

##### 3) ID selector

This selector targets elements with a specific ID attribute

```
# logo {  
    width: 200px;  
}
```

This will select the element with `id="logo"` and set its width to 200

##### 4) Universal Selector

It selects all the elements in HTML document

\* \*

```
margin: 0;  
color: black;
```

3

##### 5) Group Selector

It allows to select multiple elements at once by separating them with comma

```
# logo, h1, p, h3 {  
    color: Red;  
}
```

3

## Text formatting Properties

It is used to control the appearance and styling of text content.

- 1) color : sets the color of the text
- 2) font-size : sets the size of the text
- 3) font-weight : specifies the boldness of text
- 4) text-align : aligns the text horizontally within its container
- 5) text-decoration : decorates text such as underline, overline or line-through
- 6) text-transform : changes text to lower or upper case
- 7) letter-spacing : adjusts the spacing b/w the characters
- 8) word-spacing : adjusts the spacing b/w the words
- 9) text-indent : specifies the indentation of the first line of text within a block element
- 10) text-shadow : adds the shadow to the text

There are more properties available, these are just few examples

## Box Model Concept

It describes how elements on web page are rendered and how their content, padding, border and margin are structured within a rectangular box.

The box consists of following components

### 1. Content

It contains the actual content of an element such as text, image or any other nested elements. It can be defined by width and height properties

### 2. Padding

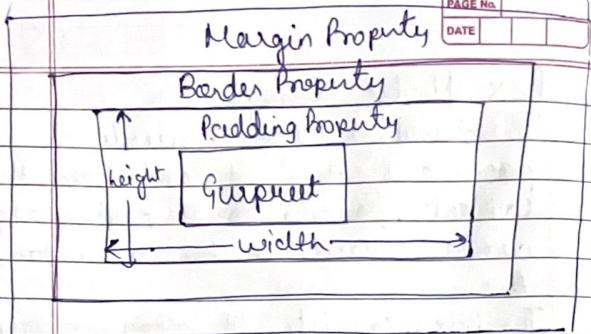
The space b/w the content and element's border. It can be set by padding property, (padding-top, padding-right, padding-bottom, padding-left).

### 3. Border

The border which surrounds the content and padding. It can be customized by using width, color, style of border property

### 4. Margin

Used to create the space around element is around the border area (margin-top, margin-right, margin-left, margin-bottom)



Content area: consists of content like image, text etc.

Padding area: the space around the content area

Border area: area b/w box's padding and margin

Margin area: space b/w the border and the margin, used to separate element from its neighbour.

<html>

<head>

<title> Box Model Example </title>

<style>

.box {

width: 200px;

height: 100px;

padding: 20px;  
 border: 2px solid black;  
 margin: 10px;  
 background-color: gray;

}

</style>

</head>

<body>

<div class="box"> Box Model Example </div>

</body>

</html>

### Position Property

It specifies the type of positioning for an element. It describes how the element is placed or positioned within its parent container or viewport.

There are five different position properties available in CSS.

1. position: static

Elements are positioned in the normal flow of the document. The top, right, left, bottom properties have no effect on statically positioned elements.

## 2. position: relative

elements are positioned relative to their normal position in the flow of the document. Top, right, bottom, left properties can be used to adjust away from its normal position.

## 3. position: fixed

elements are positioned relative to viewport so they do not move when the page is scrolled. Top, bottom, right, left can be used to the exact position of the element within the viewport.

## 4. position: absolute

elements are positioned relative to nearest positioned ancestor (instead of fixed). Top, bottom, right, left can be used to define exact position relative to its containing block.

## 5. position: sticky

elements are positioned based on user scroll position.

It acts like relative and fixed positioning. This allows elements to stick to a specific position as the user scrolls.

<html>

<head>

<title> Position Example </title>

<style>

body {

margin: 0;

padding: 0;

}

.container {

height: 400px;

background-color: lightgray;

position: relative;

}

.box {

width: 100px;

height: 100px;

background-color: red;

position: static;

margin: 10px;

display: inline-block;

}

.box-relative {

position: relative;

top: 20px;

left: 20px;

}

.box-absolute {

position: absolute;

top: 50px;

left: 50px;

,

.box.fixed {  
position: fixed;  
top: 100px;  
left: 20px;  
}

.box.sticky {  
position: sticky;  
top: 150px;  
left: 150px;  
}

.box.sticky-container {  
height: 500px;  
background-color: lightblue;  
}

</Style>  
<Head>  
<Body>

<h1> Position Examples </h1>

```
<div class="container">  
<div class="static box"> Static </div>  
<div class="box relative"> Relative </div>  
<div class="box absolute"> Absolute </div>  
<div class="box fixed"> Fixed </div>  
<div class="box sticky"> Sticky </div>  
</div>
```

<div class="box sticky-container">

<div class="box"> Element within  
Sticky Container </div>  
</div>  
</body>  
</html>

### Pseudo Classes

Pseudo classes are used to select the element based on the states or conditions.  
Pseudo classes starts with colon (:) followed by the name of pseudo class.

Pseudo classes can be combined with CSS selector to add effect to existing elements based on their condition.

Commonly used pseudo classes are

#### 1. :hover

It is used to add some special effect when the mouse pointer is over it.

#### 2. :active

It is used to select an element which activated when the user clicks on it.

#### 3. :focus

PAGE NO. DATE

It is used to select an element that is currently focused by the user.

#### 4. :visited

It is used to select and style links that have been visited by the user. It allows to specify style to visited link to distinguish them from unvisited links.

#### 5. :first-child

Targets the first child element of its parent.

#### 6. :last-child

Targets the last child element of its parent.

<html>

<head>

<title> Pseudo-classes example </title>

<style>

a:hover {

color: red;

}

button:active {

background-color: yellow;

}

PAGE NO. DATE

input:focus {

border: 2px solid blue;

a:visited {

color: purple;

}

ul li:first-child {

font-weight: bold;

}

ul li:last-child {

color: green;

}

</style>

</head>

<body>

<h1> Pseudo classes </h1>

<p>

<a href="#"> Hover over me </a>

<p>

<button> Click me </button>

<p>

<input type="text" placeholder="name">

<p>

<p>

```
<a href="#" class="visited-link">  
    Visited Link </a>  
<a href="#">Unvisited Link </a>  
</p>  
<ul>  
    <li> First Item </li>  
    <li> Second Item </li>  
    <li> Third Item </li>  
</ul>  
</body>  
</html>
```