

CSS

CSS stands for **Cascading Style Sheets**.

It makes an HTML website presentable.

It adds style to various HTML elements.

It helps you to define how the elements should look, where they should be placed, and whether they should be displayed or not.

Types of CSS

1. Inline
 2. Internal
 3. External
-

Inline CSS

- Use `style` attribute of an element to decorate it
- Simplest way to add decoration
- Very difficult to manage because it can target only one element at a time
- It is discouraged to use inline CSS

Example:

```
<p style="color:red;">test</p>
```

Internal CSS

- Use `style` tag in header section
- Can target multiple elements at a time in the given page
- Can target only one page at a time

Example:

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

External CSS

- Use external CSS file to hold all the rules
- Link the external CSS with **link** tag in header section

Example:

```
<link rel="stylesheet" href="mystyles.css" />
```

Terminology

Rule or Ruleset

Pair of CSS selector and declaration block

Declaration Block

Collection of declarations

Declaration

Pair of CSS property and its value separated by colon (:) and terminated by semi-colon (;

Selector

Used to select one or more elements from the page

Units

- **px** – pixel (Picture Element)
 - **deg** – degree
 - **s** – seconds
 - **%** – with respect to its parent
-

CSS Selector

Used to select one or more elements from given page

Types of Selector

1. Element Selector

Also called type selector

Targets only similar type of element

Example:

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

2. Multiple Element Selector (,)

Multiple type selector

Select multiple types of elements

Example:

```
p,  
h2,  
h3 {  
    color: green;  
}
```

3. ID Selector (#)

Select only element matching the given id

Id can appear in a page only once

Example:

```
/* select only paragraph having id para1 */  
p#para1 {  
    color: red;  
}  
  
/* select any element having id para1 */  
#para1 {  
    color: green;  
}
```

4. Class Selector (.)

Select only element matching the given class

Example:

```
/* select only paragraph having class para1 */  
p.para1 {
```

```
color: red;
}

/* select any element having class para1 */
.para1 {
    color: green;
}
```

5. Descendant Selector (white-space)

Used to select child elements at any level

Example:

```
/* paragraph at any level inside div will have color red */
div p {
    color: red;
}
```

6. Child Selector (>)

Used to select child elements which are at first level

Used to select only direct child elements

Example:

```
/* paragraph at first level (direct) inside div will have color red */
div > p {
    color: red;
}
```

7. Attribute Selector ([])

Used to select an element based on the given attribute value

Example:

```
/* input of type submit will have color red */
input[type="submit"] {
    color: red;
}
```

8. Universal Selector (*)

Used to select all elements

Example:

```
/* all elements will have font family as arial */  
* {  
    font-family: arial;  
}
```

9. Pseudo Selector

Used to apply CSS rules in specific condition

The conditions are also known as pseudo classes

Examples: hover, nth-child, active, focus, visited

Example:

```
/* when mouse gets over on div, the color will change to red */  
div:hover {  
    color: red;  
}
```

CSS Box Model

Every element in HTML is rendered as a box (rectangle)

There are **3 properties**:

1. Border
 2. Padding
 3. Margin
-

CSS Display

Used to control the display behavior of an element

Values

1. **block**
 - Considers width and height
 - Displays elements on new line

2. `inline`

- Ignores width and height
- Displays in same line

3. `none`

- Hides the element

4. `inline-block`

- Considers width and height
 - Displays elements on same line
-

CSS Float

The `float` property in CSS is used to position elements to the left or right of a container, allowing text or other elements to wrap around them.

Values

1. `right`
 2. `left`
-

CSS Flex

Flexbox (Flexible Box Layout) is a powerful, one-dimensional layout system in CSS designed for organizing elements in rows or columns.

It simplifies alignment, spacing, and distribution of elements, making it ideal for responsive design.

To use Flexbox, apply:

```
display: flex;
```

This makes all child elements (flex items) automatically adjust according to the rules of Flexbox.

Flex Container and Flex Items

1. Flex Container – The parent element that holds flex items
2. Flex Items – The child elements inside the container

Properties Applied to Flex Container

1. `flex-direction`

1. `row` – Default. Items are placed left to right
2. `row-reverse` – Items are placed right to left

3. **column** – Items are placed top to bottom
4. **column-reverse** – Items are placed bottom to top

2. justify-content

1. **flex-start** – Default. Items align to the start (left)
 2. **flex-end** – Items align to the end (right)
 3. **center** – Items are centered
 4. **space-between** – First item at start, last item at end, space between them
-

CSS Position

Used to control the position

Values

1. **static**
 - Default value
 - Ignores top, bottom, left, and right
 2. **relative**
 - Element is aligned with respect to top, bottom, left, and right
 3. **absolute**
 - Positioned relative to the current displayed window
 - Moves as window scrolls
 4. **fixed**
 - Fixed at the position
 - Does not move when window scrolls
-

CSS3 Advanced Properties

- border-radius
- Shadow
- Text shadow
- Box shadow
- Gradients
 - Linear
 - Radial
- column-counts
 - Used to divide an element into number of columns
- CSS Animations
 - Transition
 - Transform

- scale
- rotate
- translate
- CSS Media Queries