

CSS

- **CSS (Cascading Style Sheet)** is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed (element's presentation).

We can use style sheets in three different ways in our HTML document. (Levels of Style Sheets)

1) Inline Style Sheet

- It is used to apply a unique style to HTML elements individually.
- You can insert inline styles anywhere in the middle of your HTML code.
- To use inline styles, you use the **style attribute** in the relevant tag. The style attribute can contain any CSS property.

2) Internal Style Sheet (Document level)

- It is used for applying styling to the entire body of the document.
- You define internal styles in the head section of an HTML page, by using the `<style>` tag.

3) External Style Sheet

- With an external stylesheet file, you can change the look of an entire website by changing just one file.
- External CSS is a file that contains only CSS code and is saved with a **".css"** file extension.
- It is linked to HTML document using `<link>` element.

[CSS Comments: Comments are introduced with `/*` and terminated with `*/`]

Syntax

- A CSS rule has two main parts: a selector and one or more declarations.
- Selector is normally the HTML element you want to style and each declaration consists of a property and value.
- The property is the style attribute we want to use and each property has a value associated with it.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

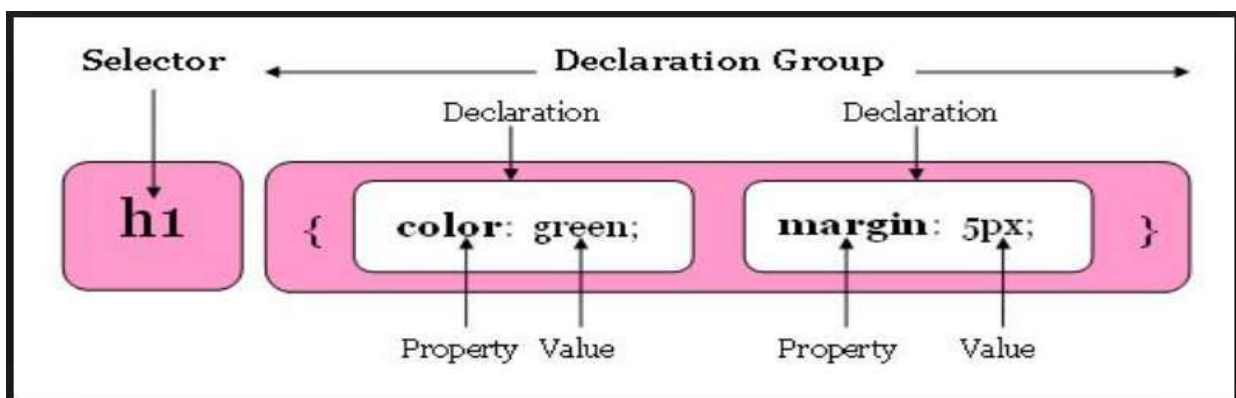


Figure 8: CSS Style Specification Format

CSS Syntax

```
■ Inline    <p style="color:red">Content</p>

■ Internal  <head>
             <style type="text/css">
             p {color:blue;}
             </style>
             </head>

■ External  <head>
             <link href="style.css"
             rel="stylesheet" type="text/css">
             </head>

             p {color:green; font:arial, sans-serif;}
```

- The **<link>** tag is used to specify external style sheets.
- The **'href'** attribute is used to specify the URL of the style sheet document.
- Within **<link>**, the **'rel'** attribute is used to specify the relationship between linked document to the document in which the link appears.

CSS Selectors

- In CSS, selectors are patterns used to select the element(s) you want to style.

Types of Selectors:

1. Element Selector

- The element selector selects elements based on the element name.
- It selects all elements of the given type within a document.

Examples:

```
h1 { color:red; }    →applies red color to all h1 elements
p { font-size: 24pt;} →applies font size 24pt to all paragraph content.
```

- **Group selector:**

- When several selectors share the same declarations, they may be grouped into a comma-separated list.
- Example:

```
h1, h2, p { color: blue; } →blue color applied to all h1,h2 and p elements.
```

- **Descendant selectors:**

- Used to match an element that is the descendant of another element in the document tree.

Example: `<h1>This headline is very important</h1>`

Then css will be → `h1 em { color: blue; }`

2. Class Selector

- The CSS class selector matches elements based on the contents of their class attribute.
- It's declared with a dot preceding a class name.

Syntax:

```
.class_name { style properties }
```

Example:

HTML: <p class="introduction"> </p>

```
CSS: .introduction{
      color:red;
}
```

- **Generic Classes** : multiple tags with same class name.

Example:

```
<h1 class="intro">Introduction heading </h1>
<p class="intro">introduction paragraph.</p>
```

Then CSS can be applied to each element in the following way:

```
p.intro {
  text-align: center;
  color: red;
}
```

3. ID Selector

- The CSS id selector matches elements based on the contents of their id attribute.
- An ID selector is declared using a hash, or pound symbol (#) preceding the value of id attribute of html element.
 - Syntax: #id_value{ style properties}
 - Example: HTML: <p id="summary"> </p>

```
CSS: #summary { color:green; }
```

4. Universal Selector

- The CSS universal selector (*) matches elements of any type.

```
/* Selects all elements */
* {
  color: green;
}
```

Pseudo Classes

- A pseudo-class is used to define a special state of an element.
- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists.
- Syntax:

```
selector:pseudo-class {
  property:value;
}
```

- It can be used to:
 - Style an element when a user mouse over it.
 - Style visited and unvisited links differently.
 - Style an element when it gets focus.

:link Use this class to add special style to an unvisited link.	<pre>a:link { color: #FF0000; }</pre>
:visited Use this class to add special style to a visited link.	<pre>/* visited link */ a:visited { color: #00FF00; }</pre>
:hover Use this class to add special style to an element when you mouse over it.	<pre>/* mouse over link */ a:hover { color: #FF00FF; }</pre>
:active Use this class to add special style to an active element.	<pre>/* selected link */ a:active { color: #0000FF; }</pre>
:focus Use this class to add special style to an element while the element has focus.	<pre>input:focus { background-color: yellow; }</pre>

CSS PROPERTY VALUE FORMS

Colors

Colors in CSS can be specified by the following methods:

Selector { color: red; } /* color keyword */

Selector { color: rgb(0 255 0); } /* RGB range 0-255 */

Selector { color: rgb(0% 100% 0%); } /* RGB range 0%-100% */

Selector { color: #ff6347; } /* hexadecimal code (#rrggbb) All values must be
between 00 and FF. */

- style="background-color:Blue;" ->> background color
- style="color:Tomato;" ->> font color

Background properties

background-color:

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

background-image:

The background-image property specifies an image to use as the background of an element.

```
body {  
  background-image: url("paper.gif");  
}
```

By default, the background-image property repeats an image both horizontally and vertically.

Other properties:

- background-repeat

- body {
 background-image: url("gradient_bg.png");
 background-repeat: repeat-x; (~~repeat-y, no-repeat~~)
}

- background-position

- The position of the image is specified by the background-position property:
 - background-position: right top;

- Background-size

- :width height (in pixel or %)
 - Cover/contain/auto/initial

cover Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges

contain Resize the background image to make sure the image is fully visible

left top
left center
left bottom
right top
right center
right bottom
center top
center center
center bottom

Font Properties

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text
<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

Examples:

font-family: 'Times New Roman', Times, serif;

font-size: 2px / font-size: 50% / font-size: large / font-size: 2em (1 em=16px) / font-size: 24pt etc.

font-style:italic; [Other values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

font-variant:small-caps; [or font-variant:normal;]

font-weight:bold [Other values : normal/lighter/lightest/bolder or from 100-900 in multiple of 100]

Font shorthand:

Syntax: font: font-style font-variant font-weight font-size font-family;

Example → font: bold 14pt 'Times New Roman' ;

Text properties

<u>text-align</u>	Specifies the horizontal alignment of text
<u>text-decoration</u>	Specifies the decoration added to text
<u>text-indent</u>	Specifies the indentation of the first line in a text-block

Examples:

text-align: left; [Other values : right/center/justify]

text-decoration:underline; [Other values: line-through/underline/none]

text-indent:2%; [Other units: em/ px]

Height and width

- The height and width properties are used **to set the height and width of an element**.
- The height and width can be set to **auto** (this is default. Means that the browser calculates the height and width), or be specified in **length values, like px, cm, etc.**, or in percent (%) of the containing block.

```
div {  
    height: 100px;  
    width: 500px;  
    background-color: powderblue;  
}
```

CSS BOX MODEL

- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of: margins, borders, padding, and the actual content.
- It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.



- **Content** - The content of the box, where text and images appear.
- **Padding** - Clears an area around the content. The padding is transparent.
- **Border** - A border that goes around the padding and content.
- **Margin** - Clears an area outside the border. The margin is transparent.

CSS Border

The CSS border properties allow you to specify the style, width, and color of an element's border.

a. Border style

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border

`border-style:none;` defines no border

b. Border width

The `border-width` property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The `border-width` property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p.one {
  border-style: solid;
  border-width: 5px;
}

p.two {
  border-style: solid;
  border-width: medium;
}

p.three {
  border-style: solid;
  border-width: 2px 10px 4px 20px;
}
```

c. Border color

The border-color property is used to set the color of the four borders.

Example: border-color:red;

d. Border shorthand

To specify all the individual border properties together.

```
p {
  border: 5px solid red;
}
```

- ✓ You can also specify all the individual border properties for just one side:
- ✓ Example→ border-left: 6px solid red;

CSS Margin

- The CSS margin properties are used to create space around elements, outside of any defined borders.

CSS has properties for specifying the margin for each side of an element:

- margin-top
- margin-right
- margin-bottom
- margin-left

- Values: **auto** - the browser calculates the margin | | **length** - specifies a margin in px, pt, cm, etc. or

specifies a margin in % of the width of the containing element.

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
 - `padding-top`
 - `padding-right`
 - `padding-bottom`
 - `padding-left`
- Example:
 - `padding: 25px 50px 75px;`
 - top padding is 25px
 - right and left paddings are 50px
 - bottom padding is 75px
 - `padding: 25px 50px;`
 - top and bottom paddings are 25px
 - right and left paddings are 50px

List properties

- The CSS list properties allow you to:
 - Set different list item markers for ordered lists.
 - Set different list item markers for unordered lists.
 - Set an image as the list item marker.
 - Add background colors to lists and list items.
- ❖ For Unordered list
 - `list-style-type: circle/square/disc/none;`
 - `list-style-image: url('image path');`
- ❖ For ordered list
 - `list-style-type: decimal/upper-alpha/lower-alpha/upper-roman/lower-roman`
- ✓ Other properties that can be applied: color, background-color, padding etc.
- ✓ Examples:

```

ol {
    background: #ff9999;
    padding: 20px;
}

ul {
    background: #3399ff;
    padding: 20px;
}

ol li {
    background: #ffe5e5;
    padding: 5px;
    margin-left: 35px;
}

ul li {
    background: #cce5ff;
    margin: 5px;
}

```

Table properties

- With CSS table properties, you can set how tables and table columns should be displayed.

1. Table border

- To specify table borders in CSS, use the **border** property.

Example:

```

table, th, td {
    border: 1px solid black;
}

```

- The **border-collapse** property sets whether the table borders should be collapsed into a single border or not.

Example:

```

table {
    border-collapse: collapse;
}

```

2. Table height & width

- Width and height of a table are defined by the width and height properties.

Example:

```

table {
    width: 100%;
}

```

```

th {

```

```
height: 50px;
```

```
}
```

3. Text align

- For the alignment of text either horizontally or vertically.
- The **text-align** property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.
- The **vertical-align** property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

Examples:

```
th {
    text-align: left;
}
td {
    vertical-align: bottom;
}
```

4. Table padding and spacing

- To control the space between the border and the content in a table, use the **padding** property on <td> and <th> elements
- The **border-spacing** property sets the distance between the borders of adjacent cells.

Example:

```
th, td {
    padding: 15px;
}
#table2 {
    border-spacing: 15px 50px;
}
```

//Using two values (the first sets the horizontal spacing and the second sets the vertical spacing)

//If single value it sets both the horizontal spacing and the vertical spacing

5. Table color

Example:

```
th {
    background-color: #4CAF50;
    color: white;
}
```

6. Table caption side

- Specify the placement of table captions

Example:

```
#table1 {
    caption-side: top;
}
#table2 {
    caption-side: bottom;
}
```

CSS POSITION

- The position property specifies the type of positioning method used for an element.
- There are five different position values:
 - static
 - relative
 - fixed
 - absolute
 - sticky
- ✓ Elements are then positioned using the top, bottom, left, and right properties.

Static position

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way, it is always positioned according to the normal flow of the page.

Relative position

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element.

Fixed position

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.

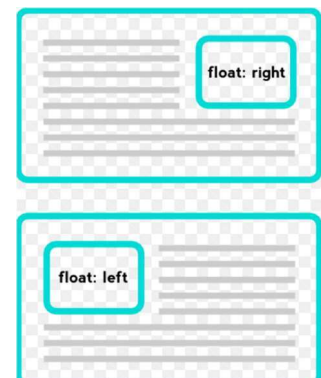
Absolute position

- An element with position: absolute; is positioned relative to the nearest positioned ancestor.

CSS FLOAT

- The float property is used for positioning and formatting html content e.g. let an image float left to the text in a container.
- The float property can have one of the following values:
 - left - The element floats to the left of its container
 - right - The element floats to the right of its container
 - none - The element does not float (will be displayed just where it occurs in the text).

- ✓ the float property can be used to wrap text around images.



- The clear property specifies what elements can float beside the cleared element and on which side.

- The clear property can have one of the following values:
 - none - Allows floating elements on both sides. This is default
 - left - No floating elements allowed on the left side
 - right - No floating elements allowed on the right side
 - both - No floating elements allowed on either the left or the right side