INDEX

CSS Introduction.....2
Include CSS in a webpage.....2
Inline Style.....2
Internal CSS.....2
External CSS.....3
CSS Syntax.....4
CSS Selector :.....5
CSS basic property.....12
Text Properties :.....12
List Properties :.....12
Border Properties :.....12
Font Properties :.....13
Overflow Properties :.....14
Float Property.....14

CSS Introduction

CSS (Cascading Style Sheets) is a styling language used to add style to a webpage.

HTML provides structure and adds content to a webpage, while CSS enhances the visual presentation of that content through various styles.

Why should you learn to use CSS?

1. Customizes and styles a website
2. Responsive Design
3. CSS Animations and Transitions

Include CSS in a webpage

There are three ways to add CSS in HTML

1. **Inline CSS**: Styles added directly to the HTML element.
2. **Internal CSS**: Styles defined at the head section of the document.
3. **External CSS**: Styles defined in a separate file. (that file save as a .css and have to include in head section of that document where we want to apply it.)

Inline Style

Inline style is the approach of adding CSS rules directly to the HTML element using the style attribute.

For example,
Hello CSS

Internal CSS

Internal CSS applies CSS styles to a specific HTML document. Internal CSS is defined inside an HTML document using <style> attribute within the head tag of an HTML.

For example,

```
<html lang="en">
<head>
    <style>
        p {
            color: red;
        }
    </style>

    <title>Internal CSS</title>
</head>

<body>
    <p>Hello CSS.</P>
 </body>

</html>
```

External CSS

External CSS is an approach to applying CSS styles to HTML pages by defining the CSS in a separate file.

we have CSS in a separate file named style.css. The external CSS file should have a .css extension.

The external CSS file can be linked to the HTML document by using the link element in the HTML.

Example:

style.css

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

Index.html

```
<html lang="en">

<head>
   <link href="style.css" rel="stylesheet">
   <title>Browser</title>
</head>

<body>
   <p>This is a sample text.</p>
</body>

</html>
```

Note :

- **We can link multiple CSS file to an HTML file.**
- **Aslo we can give multiple type of CSS at a time. But prtiority order is inline CSS, internal CSS, External CSS. That means if some tag have inline and internal both CSS than it follow inline CSS.**

CSS Syntax

CSS syntax consists of a selector and a declaration block.

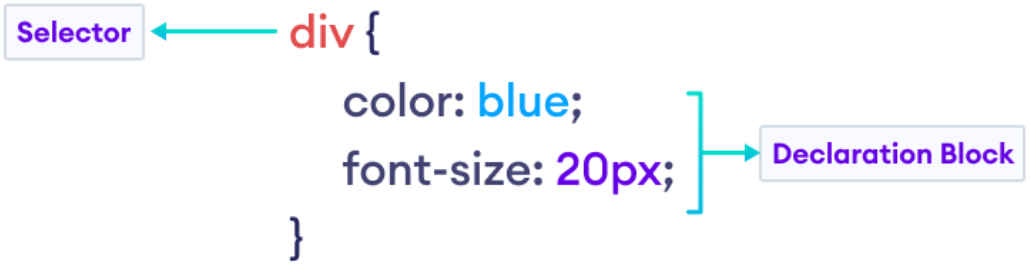
For example,

```
selector {
   property1: value;
   property2: value;
}
```

Selector : specifies the HTML element that we want to apply the styles

Property : specifies the attribute of HTML elements that we want to change

Value : specifies the new value you want to assign to the property



Styling Multiple Elements

We can apply CSS to multiple HTML elements at once.

For example

```
h1, p {
    color: red;
    font-size: 20px;
    background-color: yellow;
}
```

CSS Selector :

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Selector	**Example**	**Example description**
.class	.intro	Selects all elements with class="intro"

.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects the first <p> element that is placed immediately after elements

element1~element2	p ~ ul	Selects every element that is preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute*=*value*]	[target="_blank"]	Selects all elements with target="_blank"
[*attribute*~=*value*]	[title~="flower"]	Selects all elements with a title attribute containing the word "flower"
[*attribute*|=*value*]	[lang|="en"]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
[*attribute*^=*value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[*attribute*$=*value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[*attribute**=*value*]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
:active	a:active	Selects the active link

::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:default	input:default	Selects the default <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element

:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus
:fullscreen	:fullscreen	Selects the element that is in full-screen mode
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects input elements with a value within a specified range
:invalid	input:invalid	Selects all input elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)
:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links

::marker	::marker	Selects the markers of list items
:not(*selector*)	:not(p)	Selects every element that is not a <p> element
:nth-child(*n*)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(*n*)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(*n*)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects input elements with no "required" attribute

:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
::placeholder	input::placeholder	Selects input elements with the "placeholder" attribute specified
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified
:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
:required	input:required	Selects input elements with the "required" attribute specified
:root	:root	Selects the document's root element
::selection	::selection	Selects the portion of an element that is selected by a user
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all input elements with a valid value

:visited	a:visited	Selects all visited links

CSS basic property

Text Properties :

- Color
- Line-height
- Letter-spacing
- Text-align
- Text-decoration
- Text-indent
- text-transform

List Properties :

- List-style
- List-style-image
- List-style-position
- list-style-type

Border Properties :

- Border

- Border-bottom
- Border-bottom-color
- Border-bottom-style
- Border-bottom-width
- Border-color
- Border-left
- Border-left-color
- Border-left-style
- Border-left-width
- Border-right
- Border-right-color
- Border-right-style
- Border-right-width
- Border-style
- Border-top
- Border-top-color
- Border-top-style
- Border-top-width
- Border-width

Font Properties :

- Font
- Font-family
- Font-size
- Font-style
- Font-variant

- Font-weight

Overflow Properties :

- **visible** - Default. The overflow is not clipped. The content renders outside the element's box
- **hidden** - The overflow is clipped, and the rest of the content will be invisible
- **scroll** - The overflow is clipped, and a scrollbar is added to see the rest of the content
- **auto** - Similar to scroll, but it adds scrollbars only when necessary

Float Property

The float property is used for positioning and formatting 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). This is default
- inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.