CSS

### Introduction to CSS

CSS is short for **C**ascading **S**tyle **S**heets, and is the preferred way for setting the look and feel of a website. Cascading Style Sheets (CSS) is a markup language responsible for how your web pages will look like. It controls the colors, fonts, and layouts of your website elements.



This style sheet language also allows you to add effects or animations to your website. You can use it to display some CSS animations like click button effects, spinners or loaders, and animated backgrounds. Without CSS, your website will appear as a plain HTML page.

The cascading means that a style applied to a parent element will also apply to all children elements within the parent. For example, setting the colour of body text will mean all headings and paragraphs within the body will also be the same colour.

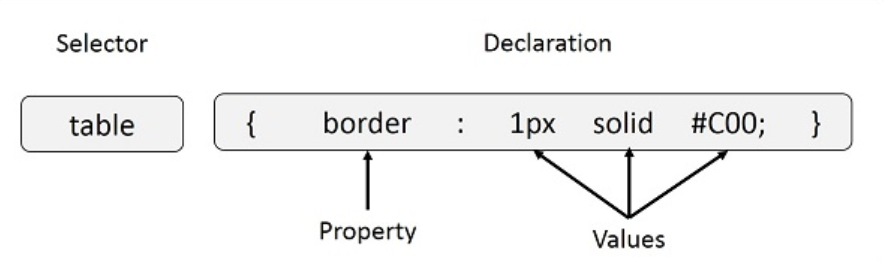
**CSS - Syntax**

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
* **Property**− A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
* **Value**− Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows −

    selector { property: value }



**Example**− You can define a table border as follows −

    table{ border :1px solid #C00; }

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

**The Type Selectors:**This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings −

h1 {

  color: #36cfff;

}

**The Universal Selectors:**Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type −

\* {

  color: #000000;

}

This rule renders the content of every element in our document in black.

**The Descendant Selectors:**Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

ul em {

  color: #000000;

}

**The Class Selectors:** You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

.black {

  color: #000000;

}

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example −

h1.black {

  color: #000000;

}

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

You can apply more than one class selectors to given element. Consider the following example −

<p class="center bold">

  This para will be styled by the classes center and bold.

</p>

**The ID Selectors:**You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

#black {

  color: #000000;

}

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example −

h1#black {

  color: #000000;

}

This rule renders the content in black for only <h1> elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors, For example −

#black h2 {

  color: #000000;

}

In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having *id* attribute set to *black*.

**The Child Selectors:** You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example −

body > p {

  color: #000000;

}

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

**The Attribute Selectors:**You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text* −

input[type='text'] {

  color: #000000;

}

The advantage to this method is that the <input type = "submit" /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

* p[lang] − Selects all paragraph elements with a *lang* attribute.
* p[lang="fr"] − Selects all paragraph elements whose *lang* attribute has a value of exactly "fr".
* p[lang~="fr"] − Selects all paragraph elements whose *lang* attribute contains the word "fr".
* p[lang|="en"] − Selects all paragraph elements whose *lang* attribute contains values that are exactly "en", or begin with "en-".

**Multiple Style Rules:** You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example −

h1 {

  color: #36c;

  font-weight: normal;

  letter-spacing: 0.4em;

  margin-bottom: 1em;

  text-transform: lowercase;

}

Here all the property and value pairs are separated by a semicolon (;). You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find complete detail about properties in CSS References

**Grouping Selectors:** You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example −

h1, h2, h3 {

    color: #36C;

    font-weight: normal;

    letter-spacing: .4em;

    margin-bottom: 1em;

    text-transform: lowercase;

 }

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *id* selectors together as shown below −

#content,

#footer,

#supplement {

  position: absolute;

  left: 510px;

  width: 200px;

}

**Types of CSS**

There is three type (ways to use) of Cascading Style Sheets

* Inline CSS
* Internal or Embedded CSS
* External CSS

## How to use CSS in HTML?

As we know there is 3 type or ways to use CSS in HTML – inline, internal and external. Let’s See one by one how use it with HTML code (tags).

**Inline CSS**

With inline CSS you have to dine a css in the HTML tag, for example, you are using <p> tag than inside open this tag have to write style = “property: value;”>. See the below example a using CSS (color, font-size, font-style, test-align) with HTML.

<!DOCTYPE html>

<html>

    <head>

        <title>Inline CSS</title>

    </head>

    <body>

        <p style = "color:#049CA2;

                    font-size:50px;

                    font-style:bold;

                    text-align:center;">

        Coodefun</p>

    </body>

</html>

**Output and source code screenshot**, a screenshot took with inception view, so you can see left side is output and right side is code.



**Internal or Embedded CSS**

It’s also a single HTML document CSS same as inline CSS. What is Single HTML documented – When a CSS and HTML code inside the same file. Internal CSS (Embedded CSS) and inline CSS both are example of it.

To define a Internal CSS you need to use **<head>** tag and inside it use **<style>…</style>**. here is One concept come a “class”, you have to use the class attribute to give a particular identity of the tag. So it will change only the single tag. In this example using a <div> tag, which is stand for division or a section in an HTML document.

<!DOCTYPE html>

<html>

  <head>

    <title>Internal CSS</title>

    <style>

      .main {

        text-align: center;

      }

      .name {

        color: #049ca2;

        font-size: 40px;

        font-weight: bold;

      }

      .learn {

        font-style: bold;

        font-size: 20px;

      }

    </style>

  </head>

  <body>

    <div class="main">

      <div class="name">Coodefun</div>

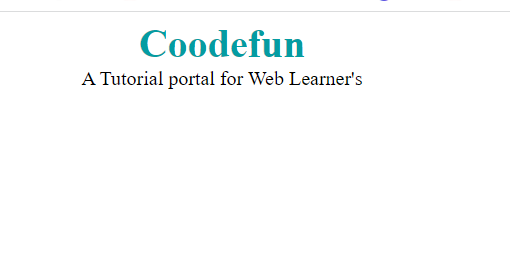
      <div class="learn">A Tutorial portal for Web Learner's</div>

    </div>

  </body>

</html>

**Output and source code screenshot –** You can also use **<p> tag** for this example.



**External CSS**

The last and most important External CSS, it uses a separate class and link to HTML page. The separate CSS class only contain a style property with the help of tag attributes of HTML (For Example class, id, etc).

The CSS file extension is dot CSS and you can give any name of it, for example “**style.css**“. To Link a **Cascading Style Sheets**you have to use **<link>** tag in a**<head>** tag with the file path.

**Cascading Style Sheets –**style.css

body {

  background-color: lightgray;

}

.main {

  text-align: center;

}

.name {

  color: dimgray;

  font-size: 50px;

  font-weight: bold;

}

#learn {

  color: #898989;

  font-style: italic;

  font-size: 20px;

}

**HTML file** – MySite.html

<!DOCTYPE html>

<html>

    <head>

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

    </head>

    <body>

            <div class="main">

                <div class="name">Coodefun</div>

                <div id="learn">

                    Hello Little Ones </p>

                </div>

    </body>

</html>

Output and source code screenshot: Left Side is output and rise is resources then code.



**Save a CSS File**

Just like the html save the file, you have to save CSS file name with dot css extension. For an example can file – “style.css”. You can put any name for cascading style sheets.

**Imported CSS - @import Rule**

@import is used to import an external stylesheet in a manner similar to the <link> element. You can also import css from another external css file. Here is the generic syntax of @import rule.

head>

  @import "URL";

</head>

Here URL is the URL of the style sheet file having style rules. You can import CSSsyntax as well −

<head>

    @import url("URL");

 </head>

Let's look at an example. Here, department\_store.css will be imported into sports\_styles.css which will then be linked to the HTML document.

**department\_store.css**

h1 {  
  text-decoration: underline;  
}  
p {  
  padding-left: 20px;  
  font-size: 18px;

}

**sports\_style.css**

@import url('department\_store.css');

h1 {

  color: #4583c2;

}

**sports\_main\_page.html**

<!DOCTYPE html>

<html>

  <head>

    <title>Sports Department</title>

    <link href="sports\_style.css" rel="stylesheet" />

  </head>

  ...

</html>

**Advantages of css**

* Easy maintenance
* Fast webpage loading
* Search engine friendly
* Absolute positioning
* Printer Friendly

**Disadvantages of css**

* Cross-Browser Issues
* Comes in Different levels – CS, CS1 to CS3
* Fragmentation – what works with one browser may not always work with another.
* Lack of security – Because it is an open text-based system, CSS doesn’t have the built-in security that will protect it from being overridden. Anyone who has a read/write access to a website can change the CSS file, alter the links or disrupt the formatting, whether by accident or design.

**What does CSS stand for or CSS meaning?**

CSS meaning is Cascading Style Sheets and its use for the style a webpage in Web Applications.

**Why CSS used in HTML?**

A CSS is used in with HTML code make your website more attractive. Without using CSS code in HTML webpage look very ugly, everything will be messed. CSS is controlling a tag (property) of HTML.

**How to use a CSS or How to use a CSS file in HTML?**

There are 3 ways, You can use a CSS with HTML first one is inline within a tag (e.g. **<p>, <a>, <h1>**etc). Second ways it used Internal CSS which required to give identity to tag and define is the style in **<head>** section. And last way is to create a separate **CSS file** and link to by using a **<Link…>**tag. Don’t forget to add rel attribute as **rel=”StyleSheet”**.

# Important Note  !

## CSS Rules Overriding

We have discussed four ways to include style sheet rules in a an HTML document. Here is the rule to override any Style Sheet Rule.

* Any inline style sheet takes highest priority. So, it will override any rule defined in <style>...</style> tags or rules defined in any external style sheet file.
* Any rule defined in <style>...</style> tags will override rules defined in any external style sheet file.
* Any rule defined in external style sheet file takes lowest priority, and rules defined in this file will be applied only when above two rules are not applicable.