**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
* **Offline Browsing** − CSS can store web applications locally with the help of an offline catche.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.
* **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

Who Creates and Maintains CSS?

CSS was invited by **Håkon Wium Lie** on October 10, 1994 and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called **specifications**. When a specification has been discussed and officially ratified by W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

**NOTE** − The World Wide Web Consortium, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

CSS Versions

Cascading Style Sheets, level 1 (CSS1) was came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 was became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

CSS3 was became a W3C recommendation in June 1999 and builds on older versions CSS. it has divided into documentations is called as Modules and here each module having new extension features defined in CSS2.

CSS3 Modules

CSS3 Modules are having old CSS specifications as well as extension features.

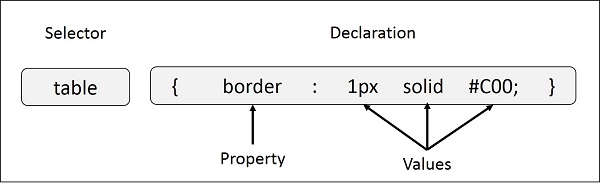
* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

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: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

Here all the property and value pairs are separated by a **semi colon (;)**. You can keep them in a single line or multiple lines. For better readability we keep them into 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](https://www.tutorialspoint.com/css/css_references.htm).

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;

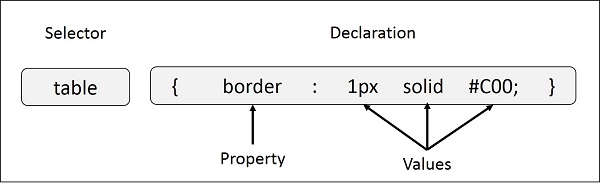
}

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: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

Here all the property and value pairs are separated by a **semi colon (;)**. You can keep them in a single line or multiple lines. For better readability we keep them into 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](https://www.tutorialspoint.com/css/css_references.htm).

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;

}

# CSS Selector

**CSS selectors** are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

## 1) CSS Element Selector

The element selector selects the HTML element by name.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p{
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<p>**This style will be applied on every paragraph.**</p>**
13. **<p** id="para1"**>**Me too!**</p>**
14. **<p>**And me!**</p>**
15. **</body>**
16. **</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssselector1)

Output:

This style will be applied on every paragraph.

Me too!

And me!

## 2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let?s take an example with the id "para1".

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. #para1 {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<p** id="para1"**>**Hello Javatpoint.com**</p>**
13. **<p>**This paragraph will not be affected.**</p>**
14. **</body>**
15. **</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssselector2)

Output:

Hello Javatpoint.com

This paragraph will not be affected.

## 3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

#### Note: A class name should not be started with a number.

Let's take an example with a class "center".

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. .center {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1** class="center"**>**This heading is blue and center-aligned.**</h1>**
13. **<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**
14. **</body>**
15. **</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssselector3)

Output:

## This heading is blue and center-aligned.

This paragraph is blue and center-aligned.

## CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p.center {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1** class="center"**>**This heading is not affected**</h1>**
13. **<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**
14. **</body>**
15. **</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssselector32)

Output:

## This heading is not affected

This paragraph is blue and center-aligned.

## 4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. \* {
6. color: green;
7. font-size: 20px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**This is heading**</h2>**
13. **<p>**This style will be applied on every paragraph.**</p>**
14. **<p** id="para1"**>**Me too!**</p>**
15. **<p>**And me!**</p>**
16. **</body>**
17. **</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssselector4)

Output:

## This is heading

This style will be applied on every paragraph.

Me too!

And me!

## 5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

1. h1 {
2. text-align: center;
3. color: blue;
4. }
5. h2 {
6. text-align: center;
7. color: blue;
8. }
9. p {
10. text-align: center;
11. color: blue;
12. }

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

1. h1,h2,p {
2. text-align: center;
3. color: blue;
4. }

Let's see the full example of CSS group selector.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1, h2, p {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1>**Hello Javatpoint.com**</h1>**
13. **<h2>**Hello Javatpoint.com (In smaller font)**</h2>**
14. **<p>**This is a paragraph.**</p>**
15. **</body>**
16. **</html>**