

CSS is the language we use to style a Web page.

## What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

## Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

## CSS Example

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

## CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

```
<h1>This is a heading</h1>
```

`<p>This is a paragraph.</p>`

When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

## Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

## External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

```
<link rel="stylesheet" href="mystyle.css">
```

An external style sheet can be written in any text editor, and must be saved with a `.css` extension.

The external `.css` file should not contain any HTML tags.

Here is how the "mystyle.css" file looks:

**Mystyle.css**

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

```
}  
  
h1 {  
  color: navy;  
  margin-left: 20px;  
}
```

## Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

### Example

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<head>  
<style>  
body {  
  background-color: linen;  
}  
  
h1 {  
  color: maroon;  
  margin-left: 40px;  
}  
</style>
```

## Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>
```

# Cascading Order

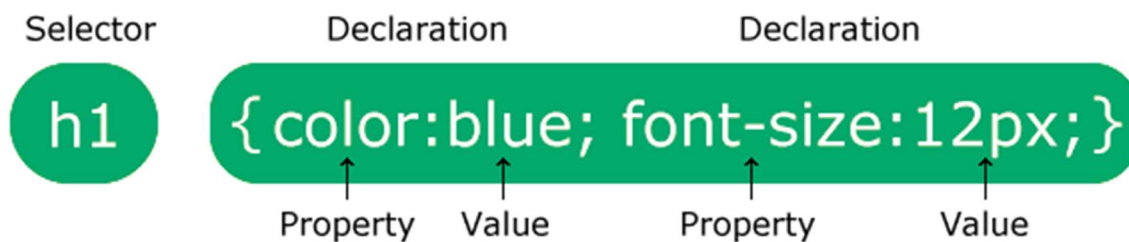
What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

## CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

## Example

In this example all <p> elements will be center-aligned, with a red text color:

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

```
text-align: center;
}
```

## Example Explained

- `p` is a selector in CSS (it points to the HTML element you want to style: `<p>`).
- `color` is a property, and `red` is the property value
- `text-align` is a property, and `center` is the property value

## CSS selectors

The **CSS selectors** module defines the patterns to select elements to which a set of CSS rules are then applied along with their [specificity](#). The CSS selectors module provides us with more than 60 selectors and five combinators. [Other modules](#) provide additional pseudo-class selectors and pseudo-elements.

In CSS, selectors are patterns used to match, or select, the elements you want to style. Selectors are also used in JavaScript to enable selecting the DOM nodes to return as a [NodeList](#).

Selectors, whether used in CSS or JavaScript, enable targeting HTML elements based on their type, attributes, current states, and even position in the DOM. Combinators allow you to be more precise when selecting elements by enabling selecting elements based on their relationship to other elements.

## CSS Selector

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>.class1.class2</u>	.name1.name2	Selects all elements with both <i>name1</i> and <i>name2</i> set within its class attribute
<u>.class1 .class2</u>	.name1 .name2	Selects all elements with <i>name2</i> that is a descendant of an element with <i>name1</i>
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element.class</u>	p.intro	Selects all <p> elements with class="intro"
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element&gt;element</u>	div > p	Selects all <p> elements where the parent is a <div> element

<u>element+element</u>	div + p	Selects the first <p> element that is placed immediately after <div> elements
<u>element1~element2</u>	p ~ ul	Selects every <ul> element that is preceded by a <p> element
<u>[attribute]</u>	[target]	Selects all elements with a target attribute
<u>[attribute=value]</u>	[target="_blank"]	Selects all elements with target="_blank"
<u>[attribute~=value]</u>	[title~="flower"]	Selects all elements with a title attribute containing the word "flower"
<u>[attribute =value]</u>	[lang="en"]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
<u>[attribute^=value]</u>	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
<u>[attribute\$=value]</u>	a[href\$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
<u>[attribute*=value]</u>	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

<a href="#"><u>:active</u></a>	a:active	Selects the active link
<a href="#"><u>::after</u></a>	p::after	Insert something after the content of each <p> element
<a href="#"><u>::before</u></a>	p::before	Insert something before the content of each <p> element
<a href="#"><u>:checked</u></a>	input:checked	Selects every checked <input> element
<a href="#"><u>:default</u></a>	input:default	Selects the default <input> element
<a href="#"><u>:disabled</u></a>	input:disabled	Selects every disabled <input> element
<a href="#"><u>:empty</u></a>	p:empty	Selects every <p> element that has no children (including text nodes)
<a href="#"><u>:enabled</u></a>	input:enabled	Selects every enabled <input> element
<a href="#"><u>:first-child</u></a>	p:first-child	Selects every <p> element that is the first child of its parent
<a href="#"><u>::first-letter</u></a>	p::first-letter	Selects the first letter of every <p> element
<a href="#"><u>::first-line</u></a>	p::first-line	Selects the first line of every <p> element
<a href="#"><u>:first-of-type</u></a>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<a href="#"><u>:focus</u></a>	input:focus	Selects the input element which has focus
<a href="#"><u>:fullscreen</u></a>	:fullscreen	Selects the element that is in full-screen mode
<a href="#"><u>:hover</u></a>	a:hover	Selects links on mouse over
<a href="#"><u>:in-range</u></a>	input:in-range	Selects input elements with a value within a specified range
<a href="#"><u>:indeterminate</u></a>	input:indeterminate	Selects input elements that are in an indeterminate state
<a href="#"><u>:invalid</u></a>	input:invalid	Selects all input elements with an invalid value
<a href="#"><u>:lang(<i>language</i>)</u></a>	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)
<a href="#"><u>:last-child</u></a>	p:last-child	Selects every <p> element that is the last child of its parent
<a href="#"><u>:last-of-type</u></a>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<a href="#"><u>:link</u></a>	a:link	Selects all unvisited links
<a href="#"><u>::marker</u></a>	::marker	Selects the markers of list items
<a href="#"><u>:not(<i>selector</i>)</u></a>	:not(p)	Selects every element that is not a <p> element

<u>:nth-last-child(<i>n</i>)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(<i>n</i>)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(<i>n</i>)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects input elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects input elements with a value outside a specified range
<u>::placeholder</u>	input::placeholder	Selects input elements with the "placeholder" attribute specified
<u>:read-only</u>	input:read-only	Selects input elements with the "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects input elements with the "readonly" attribute NOT specified
<u>:required</u>	input:required	Selects input elements with the "required" attribute specified
<u>:required</u>	input:required	Selects input elements with the "required" attribute specified
<u>:root</u>	:root	Selects the document's root element
<u>::selection</u>	::selection	Selects the portion of an element that is selected by a user
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<u>:valid</u>	input:valid	Selects all input elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links



## CSS Basic Properties

### Text Properties

Property	Description	Values
<b>color</b>	Sets the color of a text	RGB, hex, keyword
<b>line-height</b>	Sets the distance between lines	normal, <i>number</i> , <i>length</i> , %
<b>letter-spacing</b>	Increase or decrease the space between characters	normal, <i>length</i>
<b>text-align</b>	Aligns the text in an element	left, right, center, justify
<b>text-decoration</b>	Adds decoration to text	none, underline, overline, line-through
<b>text-indent</b>	Indents the first line of text in an element	<i>length</i> , %
<b>text-transform</b>	Controls the letters in an element	none, capitalize, uppercase, lowercase

### List Properties

Property	Description	Values
<b>list-style</b>	Sets all the properties for a list in one declaration	<i>list-style-type</i> , <i>list-style-position</i> , <i>list-style-image</i> , inherit
<b>list-style-image</b>	Specifies an image as the list-item marker	URL, none, inherit
<b>list-style-position</b>	Specifies where to place the list-item marker	inside, outside, inherit
<b>list-style-type</b>	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit

## Border Properties

Property	Description	Values
<b>border</b>	Sets all the border properties in one declaration	<i>border-width, border-style, border-color</i>
<b>border-bottom</b>	Sets all the bottom border properties in one declaration	<i>border-bottom-width, border-bottom-style, border-bottom-color</i>
<b>border-bottom-color</b>	Sets the color of the bottom border	<i>border-color</i>
<b>border-bottom-style</b>	Sets the style of the bottom border	<i>border-style</i>
<b>border-bottom-width</b>	Sets the width of the bottom border	<i>border-width</i>
<b>border-color</b>	Sets the color of the four borders	<i>color_name, hex_number, rgb_number, transparent, inherit</i>
<b>border-left</b>	Sets all the left border properties in one declaration	<i>border-left-width, border-left-style, border-left-color</i>
<b>border-left-color</b>	Sets the color of the left border	<i>border-color</i>
<b>border-left-style</b>	Sets the style of the left border	<i>border-style</i>
<b>border-left-width</b>	Sets the width of the left border	<i>border-width</i>
<b>border-right</b>	Sets all the right border properties in one declaration	<i>border-right-width, border-right-style, border-right-color</i>
<b>border-right-color</b>	Sets the color of the right border	<i>border-color</i>
<b>border-right-style</b>	Sets the style of the right border	<i>border-style</i>
<b>border-right-width</b>	Sets the width of the right border	<i>border-width</i>
<b>border-style</b>	Sets the style of the four borders	<i>none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit</i>
<b>border-top</b>	Sets all the top border properties in one declaration	<i>border-top-width, border-top-style, border-top-color</i>
<b>border-top-color</b>	Sets the color of the top border	<i>border-color</i>
<b>border-top-style</b>	Sets the style of the top border	<i>border-style</i>
<b>border-top-width</b>	Sets the width of the top border	<i>border-width</i>
<b>border-width</b>	Sets the width of the four borders	<i>thin, medium, thick, length, inherit</i>

## Font Properties

Property	Description	Values
<b>font</b>	Sets all the font properties in one declaration	<i>font-style, font-variant, font-weight, font-size/line-height, font-family</i> , caption, icon, menu, message-box, small-caption, status-bar, inherit
<b>font-family</b>	Specifies the font family for text	<i>family-name, generic-family</i> , inherit
<b>font-size</b>	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, <i>length</i> , %, inherit
<b>font-style</b>	Specifies the font style for text	normal, italic, oblique, inherit
<b>font-variant</b>	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
<b>font-weight</b>	Specifies the weight of a font	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit <b>Careful, many of these are not supported!</b>