CSS Tutorial

* CSS stands for Cascading Style Sheets.
* It is the language, used to style a document written in a markup language such as HTML or XML.
* 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.
* World Wide Web Consortium (W3C) created CSS.

# 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.

**Types of CSS Selectors:**

1. Simple selectors (select elements based on name, id, class)
2. [Combinator selectors](https://www.w3schools.com/css/css_combinators.asp) (select elements based on a specific relationship between them)
3. [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
4. [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
5. [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)
6. **Simple Selector:**
7. **The tag Selector:**  styles a html tag based on its name.

Example: p { text-align: center; color: red;}

1. **The id Selector:** uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example: CSS -> #id1 { text-align: center; color: red;}

HTML -> <p id=”id1”>test data</p>

1. **The class selector:** selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.

For Example: .center { text-align: center; color: red;}

You can also specify that only specific HTML elements should be affected by a class.

For Example: p.center { text-align: center; color: red;}

HTML elements can also refer to more than one class.

Example: <p class="center large">This paragraph refers to two classes.</p>

1. **The Universal Selector**: (\*) selects all HTML elements on the page.

Example: \* { text-align: center; color: blue;}

1. **The Grouping Selector:** selects all the HTML elements with the same style definitions.

Example: h1, h2, p { text-align: center; color: red;}

1. [**Combinator selectors**](https://www.w3schools.com/css/css_combinators.asp)**:**
2. **Descendant combinator (space):** matches all elements that are inside of a specified element.

For Example: div p { background-color: yellow;} style all <p> elements which are nested in <div> element.

1. **Child combinator (>):** matches all elements that are the **direct** children of a specified element. For Example: div > p {background-color: yellow;}
2. **Adjacent sibling combinator (+):** matches an element which is **immediately** following a specific element.

For Example: h1 + h2 { color: blue;} style the <h2> tag which is immediate after a <h1> tag.

1. **Subsequent-sibling combinator (~):** matches all elements which are following a specific element. Example: h1 ~ h2 { color: blue;}
2. **Pseudo-class selectors**

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user moves the mouse over it
* Style visited and unvisited links differently
* Style an element when it gets focus
* Style valid/invalid/required/optional form elements

**Syntax:**

selector:pseudo-class { property: value;}

Some Common types:

1. **:hover** – style an element when user hover over it using their mouse.

Example – button:hover { color:red;}

1. **:active** – style an element when it being activated just as when user click on it.
2. **:visited** – style a link that has already been clicked by the user.

Example – a:visited { color:red;}

**Note:** a:hover MUST come after a:link (to style unvisited link) and a:visited in the CSS definition in order to be effective!

a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

1. **:focus** – style a element that currently have focus such as input or enter button.
2. **:first-child** – style a specified element that is the first child of another element or root.
3. **:last-child** – style a specified element that is the last child of another element or root.
4. **:first-of-type** – style first occurence of a specified element type within its parent.
5. **:nth-child(n)** – style a specified element that is the nth child of another element or root.

Example – p:nth-child(3) { color:red;} style 3rd <p> tag in its parent.

p:nth-child(even) { color:red;} style all even positioned (0,2,..) <p> tag in its parent.

**Pseudo-classes and HTML Classes:**

Pseudo-classes can be combined with HTML classes.

Example: a.class\_name:hover { color: #ff0000;} style a link whose class=”class\_name”

**Match the first <i> element in all <p> elements:**

p i:first-child { color: blue;}

**Match all <i> elements in all first child <p> elements:**

p:first-child i {color: blue;}

1. **Pseudo-elements selectors**

A CSS pseudo-element is used to style specific parts of an element.

For example, it can be used to:

* Style the first letter or line, of an element
* Insert content before or after an element
* Style the markers of list items
* Style the viewbox behind a dialog box

**Syntax:**

selector::pseudo-element { property: value;}

Some Common types:

1. **::first-line** – used to add a special style to the first line of a text.

Example- p::first-line { color: #ff0000;  font-variant: small-caps;}

1. **::first-letter** – used to add a special style to the first letter of a text.
2. **::before** – used to insert some content before the content of an element.

Example- h1::before { content: url(smiley.gif);}

1. **::after** – used to insert some content after the content of an element.
2. **::marker** – style the markers of list items.

Example- ::marker { color: red; font-size: 23px;}

1. **Attribute selectors**

Used to style HTML elements that have specific attributes or attribute values.

**Syntax:**

Tag\_name[attribute condition] { property: value;}

Some common types:

1. **[attribute] selector:** used to select elements with a specified attribute.

Example: a[target] { background-color: yellow;} selects all <a> elements with a target attribute:

1. **[attribute="value"] selector**: used to select elements with a specified attribute and value.
2. **[attribute~="word"] selector**: used to select elements with an attribute value containing a specified word.
3. **[attribute|="value"] selector**: used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).
4. **[attribute^="value"] selector**: used to select elements with the specified attribute, whose value starts with the specified value.

Eg: a[href^="https"] Selects all <a> elements with a href attribute value starting with "https".

1. **[attribute$="value"] selector**: used to select elements whose attribute value ends with a specified value.
2. **[attribute\*="value"] selector:** used to select elements whose attribute value contains a specified value.

Example: [class\*="te"] {color: yellow;} selects all elements with a class attribute value that contains "te".

**Three Ways to add CSS:**

1. **External CSS:**

With an external style sheet, you can change the look of an entire website by changing just one file. External styles are defined within the <link> element, inside the <head> section of an HTML page.

Example: test.html -> <link rel="stylesheet" href="mystyle.css">

mystyle.css -> body { background-color: lightblue;} h1 {color: navy;}

1. **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 of an HTML page.

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

1. **Inline CSS:**

An inline style may be used to apply a unique style for a single element. Inline styles are defined within the "style" attribute of the relevant element.

Example: <h1 style="color:blue;text-align:center;">This is a heading</h1>

**Cascading Order (Priority Order):**

Inline style > External and internal style sheet (calling order) > Browser default

**CSS Comments:**

A CSS comment is placed inside the <style> element, and starts with /\* and ends with \*/.

Example: /\* This is a single-line comment \*/

**CSS Colors:**

In CSS, a Color is specified using predefined color name, or RGB, HEX, HSL, RGBA, HSLA value.

For example: color name "**Tomato**" can be represented as:

* rgb(255, 99, 71)
* #ff6347
* hsl(9, 100%, 64%)

Same as color name "Tomato", but 50% transparent:

* rgba(255, 99, 71, 0.5)
* hsla(9, 100%, 64%, 0.5)

1. **rgb(*red,* *green*, *blue*):**

* Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
* To display black, set all color parameters to 0, like this: rgb(0, 0, 0).
* To display white, set all color parameters to 255, like this: rgb(255, 255, 255).

1. **rgba(*red,* *green*, *blue, alpha*):**

* RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.
* Alpha values lies between 0.0 (fully transparent) and 1.0 (not transparent at all).

1. **Hex value:**

* A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. Each lies between 00 and ff (same as decimal 0-255).
* To display black, set all values to 00, like this: #000000.
* To display white, set all values to ff, like this: #ffffff.
* The 3-digit hex code (#rgb) is a shorthand for some 6-digit hex codes (#rrggbb).

1. **hsl(*hue*, *saturation*, *lightness*):**

* Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
* Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.
* Lightness is also a percentage. 0% is black, 50% is neither light or dark, 100% is white.

Syntax Example: <p style="color:Tomato;">

**CSS Backgrounds**: used to add background effects for elements.

1. **CSS Background Color:** set the background color for HTML elements.

Example: <p style="background-color:Tomato;">

1. **CSS background-image:** The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

Example: <style> body {background-image: url("paper.gif"); } </style>

1. **CSS background-repeat:** It is used to repeat an image only horizontally (background-repeat: **repeat-x**;), or only vertically (background-repeat: **repeat-y**;), or **no-repeat**.

Example: body {  
  background-image: url("img\_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top; /\* to specify the position of the background image\*/  
 }

1. **CSS background-attachment:** The background-attachment property specifies whether the background image should **scroll** or be **fixed** (will not scroll with the rest of the page):

Example: background-attachment: fixed;

**Note:** To shorten the code, it is also possible to specify all the background properties in one single property. This is called a **shorthand property.**

Syntax: background: background-color background-image background-repeat background-attachment background-position;

Example: background: #ffffff url("img\_tree.png") no-repeat right top;

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

1. **CSS 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
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border.

1. **CSS 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. Default value is medium.

Example: p {  
  border-style: solid;  
  border-width: 5px 20px; /\* 5px top and bottom, 20px on the sides \*/  
 }

1. **CSS Border Color:** The **border-color** property is used to set the color of the four borders. If border-color is not set, it inherits the color of the element.
2. **CSS Border - Shorthand Property:**

The **border** property is a shorthand property for the following individual border properties:

* border-width
* border-style (**required**)
* border-color

Example: p { border: 5px solid red;}

You can also specify all the individual border properties for just one side:

Example: p { border-bottom: 6px solid red;}

1. **CSS Rounded Borders:**

The **border-radius** property is used to add rounded borders to an element.

Example: p {border: 2px solid red;  border-radius: 5px;}

**Note:**

* None of the OTHER CSS border properties will have ANY effect unless the border-style property is set!
* The border-style, border-width and border-color properties can have from one to four values (for the top border, right border, bottom border, and the left border).

**CSS Margins:**

Margins 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

All the margin properties can have the following values:

* auto - to horizontally center the element within its container.
* *length* - specifies a margin in px, pt, cm, etc.
* *%* - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element

**Tip:** Negative values are allowed.

To shorten the code, it is possible to specify all the margin properties in one property using ***margin***.

margin property has different number of values:

* margin: 25px 50px 75px 100px; = top, right, bottom, left;
* margin: 25px 50px 75px; = top, right and left, bottom;
* margin: 25px 50px; = Top and bottom margins are 25px, right and left margins are 50px
* margin: 25px; = all four margins are 25px;

Example: p {margin: 25px 50px;}

**CSS Padding:**

Padding is used to create space around an element's content, inside of any defined borders.

Similar to margin, we have properties for setting the padding for each side of an element (top, right, bottom, and left).

Example: div { padding: 25px 50px 75px 100px;}

**CSS height and width:**

The **height** and **width** properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

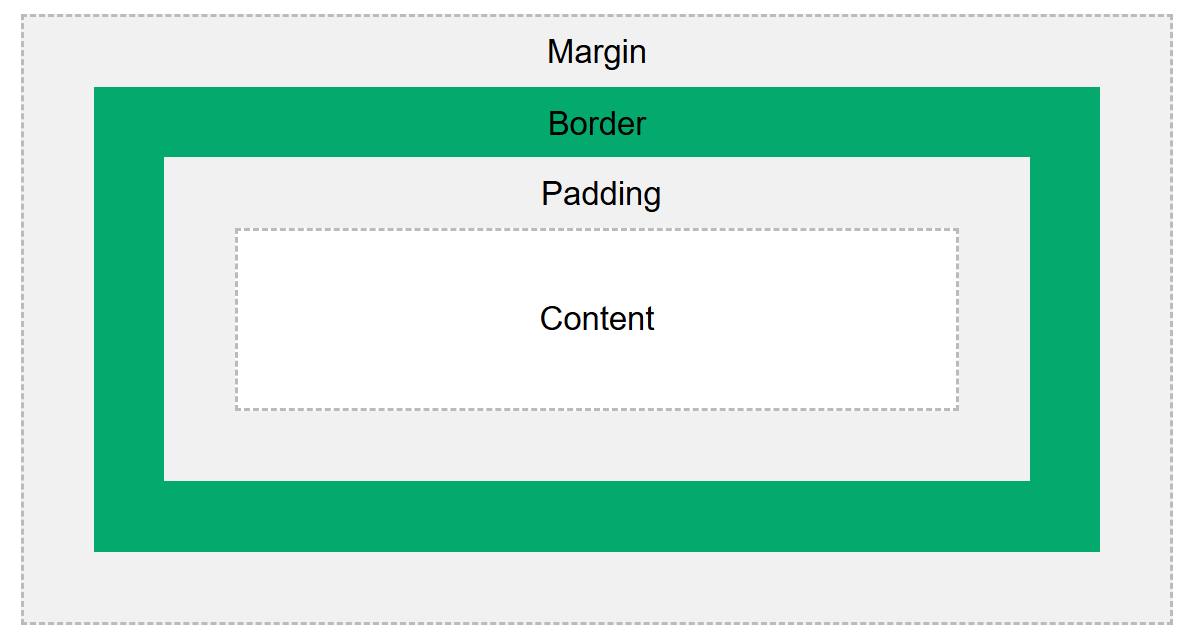
* max-height = Sets the maximum height of an element
* max-width = Sets the maximum width of an element
* min-height = Sets the minimum height of an element
* min-width = Sets the minimum width of an element

Example: div {height: 200px; width: 50%; }

**CSS Box Model:**

It is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins.

**Note:** When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the total width and height of an element, you must also include the padding and borders.



**CSS Box Sizing:**

The box-sizing property allows us to include the padding and border in an element's total width and height.

If you set box-sizing: border-box; on an element, padding and border are included in the width and height:

Example: div {  
  width: 300px;  
   padding: 25px;  
   box-sizing: border-box; }

**CSS Outline:**

An outline is a line drawn outside the element's border.

CSS has the following outline properties (similar to border-properties):

* outline-style
* outline-color
* outline-width
* outline-offset: adds space between an outline and the edge/border of an element.
* outline: outline-width outline-style outline-color;