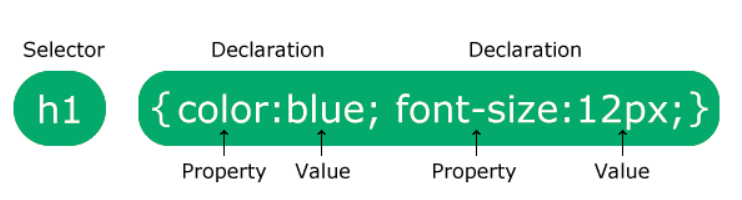
**CSS**

**Css Introduction**

* CSS stands for Cascading Style Sheets
* Used to style HTML document.

**Css Syntax:**

* A CSS rule consists of a selector and a declaration block.
* 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.
* Declaration blocks are surrounded by curly braces.



* Example:

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

**Explanation**

* p is a selector (it points to the HTML element we 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 comment:**

* Comments are used to explain the code.
* Comment not interpreted by browser.
* Syntax:

/\* comment \*/

**CSS Selectors**

* CSS selectors are used to select the HTML element.
* CSS selectors categories:
* **Simple selectors** (element, id, class)
* **Combinator selectors** (select elements based on relationship between them)
* **Pseudo-class selectors** (select elements based on a certain state)
* **Pseudo-elements selectors** (select and style a part of an element)
* **Attribute selectors** (select elements based on an attribute or attribute value)

**Simple selectors**

**Element select:**

* Element selector selects HTML elements based on the element name.
* Example:

p {

text-align: center;

color: red;

}

**Id Selector:**

* Id selector uses the **id attribute** of an HTML element to select a specific element.
* 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.
* Note: An id name cannot start with a number.
* Example:

#para1 {  
  text-align: center;  
  color: red;  
}

**Class Selector:**

* Selects HTML elements with a specific **class attribute** of HTML element.
* To select elements with a specific class, write a period (.) character, followed by the class name.
* In HTML, Multiple element can have same class name.
* And also one element can have multiple class separated by class.
* **Note:** A class name cannot start with a number.
* Example

.center {

text-align: center;

color: red;

}

* We can also specify that only specific HTML elements should be affected by a class.
* Example:

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

/\* Styled applied to only <p> elements with class="center" \*/

**Universal Selector:**

* The universal selector (\*) selects all HTML elements on the page.
* Example:

\* {

text-align: center;

color: blue;

}

/\* This style apply to all element in html document \*/

**Grouping Selector:**

* It is used to select multiple element with same style.
* Example:

h1, h2, p {

text-align: center;

color: red;

}

/\* This style apply to **h1**, **h2** and **p** element in html document \*/

**Combinator selectors**

**Descendant Selector (space):**

* The descendant selector matches all elements that are descendants of a specified element.
* Example:

div p {

background-color: yellow;

}

/\* selects all <p> elements inside <div> elements \*/

**Direct Child Selector (>):**

* The child selector selects all elements that are the direct children of a specified element.
* Example:

div > p {

background-color: yellow;

}

**Adjacent Sibling Selector (+)**

* The adjacent sibling selector is used to select an element that is directly after another specific element.
* Sibling elements must have the same parent element, and "adjacent" means "immediately following".
* Example:

div + p {

background-color: yellow;

}

**General Sibling Selector (~)**

* The general sibling selector selects all elements that are next siblings of a specified element.
* Example:

div ~ p {  
  background-color: yellow;  
}

/\* selects all <p> elements that are next siblings of <div> elements \*/

**Pseudo-class selectors:**

* A pseudo-class is used to select a specific state of an element.
* Syntax of pseudo-classes:

selector:pseudo-class {

property: value;

}

**Anchor Pseudo-classes**

* Example:

/\* unvisited link \*/

a:link {

color: #FF0000;

}

/\* visited link \*/

a:visited {

color: #00FF00;

}

/\* mouse over link \*/

a:hover {

color: #FF00FF;

}

/\* selected link \*/

a:active {

color: #0000FF;

}

**Note:**

a:hover MUST come after a: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!

**:first-child Pseudo-class**

* The :first-child pseudo-class matches a specified element that is the first child of another element.
* Example:

p:first-child {

color: blue;

}

**:lang Pseudo-class**

* The :lang pseudo-class allows you to define special rules for different languages.
* Example:

Html:

<p>

Some text

<q lang="fr">A quote in a paragraph</q>

Some text.

</p>

Style:

q:lang(fr) {

background: red;

}

**Pseudo Classes list**

|  |  |  |
| --- | --- | --- |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> elements that is the first child of its parent |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the <input> element that has focus |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects <input> elements with a value within a specified range |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all <input> elements with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> elements that is the last child of its parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links |
| [:nth-child(n)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects <input> elements with no "required" attribute |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects <input> elements with a "required" attribute specified |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all <input> elements with a valid value |
| [:visited](https://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links |

**Pseudo-element selectors:**

* A CSS pseudo-element is used to style specified parts of an element.
* Syntax:

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

**::first-line Pseudo-element:**

* The ::first-line pseudo-element is used to add a special style to the first line of a text.
* The ::first-line pseudo-element can only be applied to block-level elements.
* Example:

p::first-line {

color: #ff0000;

}

**::first-letter Pseudo-element:**

* The ::first-letter pseudo-element is used to add a special style to the first letter of a text.
* The ::first-letter pseudo-element can only be applied to block-level elements.
* Example:

p::first-letter {

color: #ff0000;

font-size: xx-large;

}

**::before Pseudo-element:**

* The ::before pseudo-element can be used to insert some content before the content of an element.
* Example:

p::before {

content: "Before Psuedo Element";

color: lightblue;

font-size: 2rem;

}

**::after Pseudo-element:**

* The ::after pseudo-element can be used to insert some content after the content of an element.
* Example:

p::after {

content: "After Psuedo Element";

color: lightblue;

font-size: 2rem;

}

**::marker Pseudo-element**

* The ::marker pseudo-element selects the markers of list items.
* Example:

::marker {

color: red;

font-size: 23px;

}

**::selection Pseudo-element**

* The ::selection pseudo-element matches the portion of an element that is selected by a user.
* Example:

::selection {

background-color: lightblue;

color: black;

}

**Attribute selectors:**

* The attribute selector is used to select elements with a specified attribute.

**[attribute]:**

* Used to select elements with a specified attribute.
* Example:

a[target] {

background-color: yellow;

}

**[attribute=value]:**

* Used to select elements with a specified attribute and value.
* Example:

a[target="\_blank"] {

background-color: yellow;

}

**[attribute~=value]**

* Used to select elements with an attribute value containing a specified word.
* Example:

[title~="flower"] {

border: 5px solid yellow;

}

**[attribute|=value]**

* Used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-)
* Example:

[class|="top"] {

background: yellow;

}

**[attribute^=value]**

* Used to select elements with the specified attribute, whose value starts with the specified value.
* Example:

[class^="top"] {

background: yellow;

}

**[attribute$="value"]**

* Used to select elements whose attribute value ends with a specified value.
* Example:

[class$="test"] {

background: yellow;

}

**[attribute\*=value]**

* Used to select elements whose attribute value contains a specified value.
* The value does not have to be a whole word.
* Example:

[class\*="te"] {

background: yellow;

}

**Colors:**

* Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

**RGB:**

* RGB color value represents Red, Green and Blue
* The value of Red, Green and Blue is between 0 to 255.

rgb(red, green, blue)

**RGBA:**

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

rgba(red, green, blue, alpha)

**Hex:**

* A hexadecimal color is specified with: #RRGGBB,

where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.

* RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and ff.

**3 Digit HEX:**

* The 3-digit hex code is a short form for some 6-digit hex codes.
* It specified with: #RGB
* Value of R, G and B is between 0 to f.

**HSL:**

* HSL stands for hue, saturation, and lightness.
* The value of hue is between 0 to 360. 0 is red, 120 is green, and 240 is blue.
* The value of saturation is between 0% to 100%. 0% means a shade of gray, and 100% is the full color.
* The value of lightness is between 0% to 100%. 0% is black and 100% is white.

hsla(hue, saturation, lightness, alpha)

**HSLA:**

* HSLA color values are an extension of HSL color values with an alpha channel
* alpha specifies the opacity for a color.
* The value of alpha is between 0 to 1.

hsla(hue, saturation, lightness, alpha)