

Multimedia,HTML

CSS, Selectors

Bhumika

CSS selectors define the elements to which a set of CSS rules apply.

Note: There is no selectors or combinators to select parent items or siblings of parent or child of sibling of parent.

Simple selectorsSection

Universal selector

Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.

Syntax: `* ns|* *|*`

Example: `*` will match all the elements of the document.

Type selector

Selects all elements that have the given node name.

Syntax: `eltname`

Example: `input` will match any `<input>` element.

Class selector

Selects all elements that have the given `class` attribute.

Syntax: `.classname`

Example: `.index` will match any element that has a class of "index".

ID selector

Selects an element based on the value of its `id` attribute. There should be only one element with a given ID in a document.

Syntax: `#idname`

Example: `#toc` will match the element that has the ID "toc".

Attribute selector

Selects all elements that have the given attribute.

Syntax: `[attr] [attr=value] [attr~=value] [attr|=value] [attr^=value] [attr$=value] [attr*=value]`

Example: `[autoplay]` will match all elements that have the `autoplay` attribute set (to any value).

CombinatorsSection

Comma combinator

The `,` combinator is a grouping method, it selects all the matching nodes.

Syntax: `A, B`

Example: `div, span` will match both `<div>` and `` elements.

Descendant combinator

The (space) combinator selects nodes that are descendants of the first element.

Syntax: `A B`

Example: `div span` will match all `` elements that are inside a `<div>` element.

Multimedia,HTML

CSS, Selectors

Bhumika

Child combinator

The > combinator selects nodes that are direct children of the first element.

Syntax: $A > B$

Example: `ul > li` will match all `` elements that are nested directly inside a `` element.

General sibling combinator

The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.

Syntax: $A \sim B$

Example: `p ~ span` will match all `` elements that follow a `<p>`, immediately or not.

Adjacent sibling combinator

The + combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.

Syntax: $A + B$

Example: `h2 + p` will match all `<p>` elements that directly follow an `<h2>`.

Column combinator

The || combinator selects nodes which belong to a column.

Syntax: $A || B$

Example: `col || td` will match all `<td>` elements that belong to the scope of the `<col>`.

PseudoSection

Pseudo classes

The : pseudo allow the selection of elements based on state information that is not contained in the document tree.

Example: `a:visited` will match all `<a>` elements that have been visited by the user.

Pseudo elements

The :: pseudo represent entities that are not included in HTML.

Example: `p::first-line` will match the first line of all `<p>` elements.

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- [Combinator selectors](#) (select elements based on a specific 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)

This page will explain the CSS simple selectors.

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

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

The CSS id Selector

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.

Multimedia,HTML

CSS, Selectors

Bhumika

Example

The CSS rule below will be applied to the HTML element with id="para1":

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

Note: An id name cannot start with a number!

The CSS class Selector

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.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

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

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

Multimedia,HTML

CSS, Selectors

Bhumika

Example

In this example only <p> elements with class="center" will be center-aligned:

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

HTML elements can also refer to more than one class.

Example

In this example the <p> element will be styled according to class="center" and to class="large":

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

Note: A class name cannot start with a number!

The CSS Universal Selector

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

Example

The CSS rule below will affect every HTML element on the page:

```
* {  
  text-align: center;  
  color: blue;  
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {  
  text-align: center;  
  color: red;  
}
```

```
h2 {  
  text-align: center;  
  color: red;  
}
```

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

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

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

CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector

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

The following example selects all <p> elements inside <div> elements:

Example

```
div p {  
  background-color: yellow;  
}
```

Child Selector

The child selector selects all elements that are the children of a specified element.

The following example selects all <p> elements that are children of a <div> element:

Example

```
div > p {  
  background-color: yellow;  
}
```

Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all <p> elements that are placed immediately after <div> elements:

Example

```
div + p {  
  background-color: yellow;  
}
```

General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all <p> elements that are siblings of <div> elements:

Example

```
div ~ p {  
  background-color: yellow;  
}
```

What are Pseudo-classes?

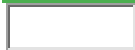
A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus



Mouse Over Me



Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {  
  property:value;  
}
```

Anchor Pseudo-classes

Links can be displayed in different ways:

Multimedia,HTML

CSS, Selectors

Bhumika

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! Pseudo-class names are not case-sensitive.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

When you hover over the link in the example, it will change color:

Example

```
a.highlight:hover {  
  color: #ff0000;  
}
```

Hover on <div>

An example of using the `:hover` pseudo-class on a `<div>` element:

Example

```
div:hover {  
  background-color: blue;  
}
```

Simple Tooltip Hover

Hover over a `<div>` element to show a `<p>` element (like a tooltip):

Hover over me to show the `<p>` element.

Example

```
p {  
  display: none;  
  background-color: yellow;  
  padding: 20px;  
}  
  
div:hover p {  
  display: block;  
}
```

CSS - The :first-child Pseudo-class

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

Match the first <p> element

In the following example, the selector matches any <p> element that is the first child of any element:

Example

```
p:first-child {  
  color: blue;  
}
```

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

In the following example, the selector matches the first <i> element in all <p> elements:

Example

```
p i:first-child {  
  color: blue;  
}
```

Match all `<i>` elements in all first child `<p>` elements

In the following example, the selector matches all `<i>` elements in `<p>` elements that are the first child of another element:

Example

```
p:first-child i {  
  color: blue;  
}
```

CSS - The `:lang` Pseudo-class

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

In the example below, `:lang` defines the quotation marks for `<q>` elements with `lang="no"`:

Example

```
<html>  
<head>  
<style>  
q:lang(no) {  
  quotes: "~" "~";  
}  
</style>  
</head>  
<body>
```

```
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
```

```
</body>  
</html>
```

What are Pseudo-Elements?

A CSS pseudo-element is used to style specified 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, the content of an element

Syntax

The syntax of pseudo-elements:

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

Notice the double colon notation - `::first-line` versus `:first-line`

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between **pseudo-classes** and **pseudo-elements**.

The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1.

For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

The `::first-line` Pseudo-element

Multimedia,HTML

CSS, Selectors

Bhumika

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all `<p>` elements:

Example

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

Note: The `::first-line` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-line` pseudo-element:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

The ::first-letter Pseudo-element

The `::first-letter` pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all `<p>` elements:

Example

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}
```

Note: The `::first-letter` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-letter` pseudo- element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

Pseudo-elements and CSS Classes

Pseudo-elements can be combined with CSS classes:

Example

```
p.intro::first-letter {  
  color: #ff0000;  
  font-size:200%;  
}
```


The example above will display the first letter of paragraphs with class="intro", in red and in a larger size.

Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

Example

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}  
  
p::first-line {  
  color: #0000ff;  
  font-variant: small-caps;  
}
```

CSS - The ::before Pseudo-element

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

The following example inserts an image before the content of each `<h1>` element:

Example

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

CSS - The ::after Pseudo-element

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

The following example inserts an image after the content of each `<h1>` element:

Example

```
h1::after {  
  content: url(smiley.gif);  
}
```

CSS - The ::selection Pseudo-element

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

The following CSS properties can be applied to `::selection`: `color`, `background`, `cursor`, and `outline`.

The following example makes the selected text red on a yellow background:

Example

```
::selection {  
  color: red;
```

```
background: yellow;  
}
```

Style HTML Elements With Specific Attributes

It is possible to style HTML elements that have specific attributes or attribute values.

CSS [attribute] Selector

The `[attribute]` selector is used to select elements with a specified attribute.

The following example selects all `<a>` elements with a target attribute:

Example

```
a[target] {  
  background-color: yellow;  
}
```

CSS [attribute="value"] Selector

The `[attribute="value"]` selector is used to select elements with a specified attribute and value.

The following example selects all `<a>` elements with a `target="_blank"` attribute:

Example

```
a[target="_blank"] {  
  background-color: yellow;  
}
```

CSS [attribute~="value"] Selector

The `[attribute~="value"]` selector is used to select elements with an attribute value containing a specified word.

The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":

Example

```
[title~="flower"] {  
  border: 5px solid yellow;  
}
```

The example above will match elements with title="flower", title="summer flower", and title="flower new", but not title="my-flower" or title="flowers".

CSS [attribute]="value"] Selector

The `[attribute]="value"]` selector is used to select elements with the specified attribute starting with the specified value.

The following example selects all elements with a class attribute value that begins with "top":

Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen(-), like class="top-text"!

Example

```
[class]="top" {  
  background: yellow;  
}
```

CSS [attribute^="value"] Selector

The `[attribute^="value"]` selector is used to select elements whose attribute value begins with a specified value.

The following example selects all elements with a class attribute value that begins with "top":

Note: The value does not have to be a whole word!

Example

```
[class^="top" ] {  
  background: yellow;  
}
```

CSS [attribute\$="value"] Selector

The `[attribute$="value"]` selector is used to select elements whose attribute value ends with a specified value.

The following example selects all elements with a class attribute value that ends with "test":

Note: The value does not have to be a whole word!

Example

```
[class$="test"] {  
  background: yellow;  
}
```

CSS [attribute*="value"] Selector

The `[attribute*="value"]` selector is used to select elements whose attribute value contains a specified value.

The following example selects all elements with a class attribute value that contains "te":

Note: The value does not have to be a whole word!

Example

```
[class*="te"] {  
  background: yellow;  
}
```

Styling Forms

The attribute selectors can be useful for styling forms without class or ID:

Example

```
input[type="text"] {  
  width: 150px;  
  display: block;  
  margin-bottom: 10px;  
  background-color: yellow;
```

Multimedia,HTML

CSS, Selectors

Bhumika

```
}
```

```
input[type="button"] {  
  width: 120px;  
  margin-left: 35px;  
  display: block;  
}
```