**Selectors and Pseudo Classes**

## **1. CSS Selectors:**

In CSS, selectors are patterns used to select the element(s) you want to style.

**1.1 List of selectors used in the CSS3:**

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [.class](https://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" |
| [#id](https://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="firstname" |
| [\*](https://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements |
| [*element*](https://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements |
| [*element,element*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements |
| [*element element*](https://www.w3schools.com/cssref/sel_element_element.asp) | div p | Selects all <p> elements inside <div> elements |
| [*element>element*](https://www.w3schools.com/cssref/sel_element_gt.asp) | div > p | Selects all <p> elements where the parent is a <div> element |
| [*element+element*](https://www.w3schools.com/cssref/sel_element_pluss.asp) | div + p | Selects all <p> elements that are placed immediately after <div> elements |
| [*element1~element2*](https://www.w3schools.com/cssref/sel_gen_sibling.asp) | p ~ ul | Selects every <ul> element that are preceded by a <p> element |
| [[attribute]](https://www.w3schools.com/cssref/sel_attribute.asp) | [target] | Selects all elements with a target attribute |
| [[attribute=value]](https://www.w3schools.com/cssref/sel_attribute_value.asp) | [target=\_blank] | Selects all elements with target="\_blank" |
| [[attribute~=value]](https://www.w3schools.com/cssref/sel_attribute_value_contains.asp) | [title~=flower] | Selects all elements with a title attribute containing the word "flower" |
| [[attribute|=value]](https://www.w3schools.com/cssref/sel_attribute_value_lang.asp) | [lang|=en] | Selects all elements with a lang attribute value starting with "en" |
| [[attribute^=value]](https://www.w3schools.com/cssref/sel_attr_begin.asp) | a[href^="https"] | Selects every <a> element whose href attribute value begins with "https" |
| [[attribute$=value]](https://www.w3schools.com/cssref/sel_attr_end.asp) | a[href$=".pdf"] | Selects every <a> element whose href attribute value ends with ".pdf" |
| [[attribute\*=value]](https://www.w3schools.com/cssref/sel_attr_contain.asp) | a[href\*="w3schools"] | Selects every <a> element whose href attribute value contains the substring "w3schools" |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert something after the content of each <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert something before the content of each <p> element |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element |
| [: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 (including text nodes) |
| [: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> element that is the first child of its parent |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element |
| [: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 which 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 equal to "it" (Italian) |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> element 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 |
| [:not(selector)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element |
| [: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 |
| [:nth-last-child(n)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| [:nth-last-of-type(n)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| [:nth-of-type(n)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects input elements with no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects input elements with a value outside a specified range |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects input elements with the "readonly" attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects input elements with the "readonly" attribute NOT specified |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects input elements with the "required" attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | :root | Selects the document's root element |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | ::selection | Selects the portion of an element that is selected by a user |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| [: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 |

# **1.2 CSS Attribute Selectors**

## 

## **1.2.1 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:**

<style>

a[target] {

background-color: yellow;

}

</style>

<body>

<p>The links with a target attribute gets a yellow background:</p>

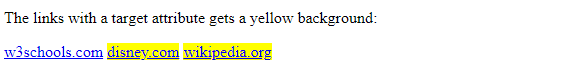
<a href="https://www.w3schools.com">w3schools.com</a>

<a href="http://www.disney.com" target="\_blank">disney.com</a>

<a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>

</body>

**Output:**



## 

## 

## 

## **1.2.2 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:**

<style>

[title~=flower] {

border: 5px solid yellow;

}

</style>

<body>

<p>All images with the title attribute containing the word "flower" get a yellow border.</p>

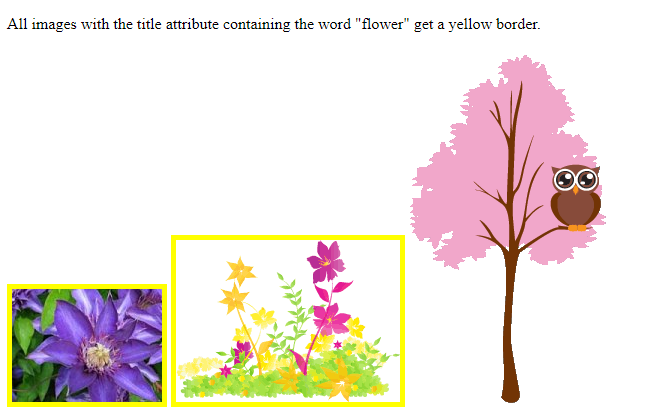
<img src="klematis.jpg" title="klematis flower" width="150" height="113">

<img src="img\_flwr.gif" title="flower" width="224" height="162">

<img src="img\_tree.gif" title="tree" width="200" height="358">

</body>

**Output:**



**1.2.3 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;

}

## **1.2.4 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;

}

## 

## 

## 

## 

## **1.2.5 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;

}

## **1.2.6 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;

}

# **2. CSS Pseudo-classes**

**Introduction**

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

**Example:**

Button in normal state

****

On Mouse Over

****

**2.1 Target Pseudo class**

URLs with an # followed by an anchor name link to a certain element within a document. The element being linked to is the target element.

The :target selector can be used to style the current active target element.

:target {

*css declarations*;

}

**Example:**

<style>

.tab div {

display: none;

}

.tab div:target {

display: block;

}

</style>

<body>

<div class="tab">

<a href="#link1">Link 1</a>

<a href="#link2">Link 2</a>

<a href="#link3">Link 3</a>

<div id="link1">

<h3>Content to Link 1</h3>

<p>Hello World!</p>

</div>

<div id="link2">

<h3>Content to Link 2</h3>

<h4>Great success!</h4>

</div>

<div id="link3">

<h3>Content to Link 3</h3>

<p>Yeah!</p>

</div>

</div>

</body>

**Output:**

On clicking Link 1

****

On clicking Link 2

****

On clicking Link 2

****

## **2.3 All CSS Pseudo Elements**

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert content after every <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert content before every <p> element |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | p::selection | Selects the portion of an element that is selected by a user |

**2.4 Negation Pseudo-Class**

The :not() [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) [pseudo-class](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes) represents elements that do not match a list of selectors. Since it prevents specific items from being selected, it is known as the *negation pseudo-class*.

/\* Selects any element that is NOT a paragraph \*/  
:not(p) {  
 color: blue;  
}

The :not() pseudo-class requires a comma-separated list of one or more selectors as its argument. The list must not contain another negation selector or a [pseudo-element](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements).

The ability to list more than one selector is experimental and not yet widely supported.

:not( <selector>[#](https://developer.mozilla.org/en-US/docs/CSS/Value_definition_syntax#Hash_mark_()) )

**Example:**

### **HTML**

<p>I am a paragraph.</p>  
<p class="fancy">I am so very fancy!</p>  
<div>I am NOT a paragraph.</div>

### **CSS**

.fancy {  
 text-shadow: 2px 2px 3px gold;  
}  
  
/\* <p> elements that are not in the class `.fancy` \*/  
p:not(.fancy) {  
 color: green;  
}  
  
/\* Elements that are not <p> elements \*/   
body :not(p) {  
 text-decoration: underline;  
}  
  
/\* Elements that are not <div> or <span> elements \*/  
body :not(div):not(span) {  
 font-weight: bold;  
}  
  
/\* Elements that are not `.crazy` or `.fancy` \*/  
/\* Note that this syntax is not well supported yet. \*/  
body :not(.crazy, .fancy) {  
 font-family: sans-serif;  
}

**RESULT:**

****

**2.5 Structural Pseudo-Classes**

Selectors introduces the concept of structural pseudo-classes to permit selection based on extra information that lies in the document tree but cannot be represented by other simple selectors or combinators.

Standalone text and other non-element nodes are not counted when calculating the position of an element in the list of children of its parent. When calculating the position of an element in the list of children of its parent, the index numbering starts at 1

## **:root pseudo-class**

The :root pseudo-class represents an element that is the root of the document. In HTML 4, this is always the HTML element.

## **:nth-child() pseudo-class**

The :nth-child(an+b) pseudo-class notation represents an element that has an+b-1 siblings before it in the document tree, for any positive integer or zero value of n, and has a parent element. For values of a and b greater than zero, this effectively divides the element’s children into groups of a elements (the last group taking the remainder), and selecting the bth element of each group. For example, this allows the selectors to address every other row in a table, and could be used to alternate the color of paragraph text in a cycle of four. The a and b values must be integers (positive, negative, or zero). The index of the first child of an element is 1.

In addition to this, :nth-child() can take ‘odd’ and ‘even’ as arguments instead. ‘odd’ has the same signification as 2n+1, and ‘even’ has the same signification as 2n.

The argument to :nth-child() must match the grammar below, where INTEGER matches the token [0-9]+ and the rest of the tokenization is given by the Lexical scanner in section 10.2:

|  |
| --- |
| **nth  : S\* [ ['-'|'+']? INTEGER? {N} [ S\* ['-'|'+'] S\* INTEGER ]? |  ['-'|'+']? INTEGER | {O}{D}{D} | {E}{V}{E}{N} ] S\*  ;** |

### 

**REFERENCES**

**https://developer.mozilla.org/en-US/docs/Web/CSS/:not**

**https://www.w3schools.com/cssref/sel\_target.asp**

**https://www.w3schools.com/css/css\_pseudo\_elements.asp**

**https://webplatform.github.io/docs/css/selectors/pseudo-classes/Structural\_pseudo-classes**