

# CSS Cheat Sheet

Type to filter sheet

## Index

1. [Selectors](#)
2. [Attributes](#)
3. [Pseudo-classes](#)

## Selectors

.class	Selects all elements with .class
#id	Selects all elements with #id
*	Selects all elements
div	Selects all <div> elements

<code>div, p</code>	Selects all <code>&lt;div&gt;</code> and <code>&lt;p&gt;</code> elements
<code>div &gt; p</code>	Selects all <code>&lt;p&gt;</code> elements that are a direct descendant of a <code>&lt;div&gt;</code> element
<code>div + p</code>	Selects all <code>&lt;p&gt;</code> elements that are the next sibling of (i.e. placed directly after) <code>&lt;div&gt;</code> elements
<code>div ~ p</code>	Selects all <code>&lt;p&gt;</code> elements that follow, and are siblings of, <code>&lt;div&gt;</code> elements

## Attributes

<code>[attribute]</code>	Selects all elements with the specified attribute
<code>[attribute=value]</code>	Selects all elements where the specified attribute is equal to 'value'
<code>[attribute~=value]</code>	Selects all elements with an attribute containing the word 'value'
<code>[attribute =value]</code>	Selects all elements with an attribute list starting with 'value'

<code>[attribute^=value]</code>	Selects all elements with an attribute beginning with 'value'
<code>[attribute\$=value]</code>	Selects all elements with an attribute ending with 'value'
<code>[attribute*=value]</code>	Selects all elements with an attribute containing the substring 'value'

## Pseudo-classes

<code>a:link</code>	Selects all unvisited links
<code>a:visited</code>	Selects all visited links
<code>a:hover</code>	Selects links on mouse hover
<code>a:active</code>	Selects the active link element
<code>p::after</code>	Insert content after <p> element
<code>p::before</code>	Insert content before <p> element
<code>input:checked</code>	Selects every checked <input> element

<code>input:disabled</code>	Selects every disabled <code>&lt;input&gt;</code> element
<code>p:empty</code>	Selects every <code>&lt;p&gt;</code> element with no children
<code>input:enabled</code>	Selects every enabled <code>&lt;input&gt;</code> element
<code>p:first-child</code>	Selects every <code>&lt;p&gt;</code> element that is the first child of its parent
<code>p::first-letter</code>	Selects the first letter of every <code>&lt;p&gt;</code> element
<code>p::first-line</code>	Selects the first line of every <code>&lt;p&gt;</code> element
<code>p:first-of-type</code>	Selects every <code>&lt;p&gt;</code> element that is the first <code>&lt;p&gt;</code> element of its parent
<code>input:focus</code>	Selects the <code>&lt;input&gt;</code> element which has focus
<code>input:in-range</code>	Selects <code>&lt;input&gt;</code> elements with a value within a specified range
<code>input:invalid</code>	Selects all <code>&lt;input&gt;</code> elements with an invalid value
<code>p:lang(<i>language</i>)</code>	Selects all <code>&lt;p&gt;</code> elements with a lang attribute equal to 'language'

<code>p:last-child</code>	Selects every <p> element which is the last child of its parent
<code>p:last-of-type</code>	Selects every <p> element which is the last <p> element of its parent
<code>:not(p)</code>	Selects every element that is not a <p>
<code>p:nth-child(2)</code>	Selects every <p> element that is the second child of its parent
<code>p:nth-last-child(2)</code>	Selects every <p> element that is the second child of its parent, counting from the last child
<code>p:nth-last-of-type(2)</code>	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<code>p:nth-of-type(2)</code>	Selects every <p> element that is the second <p> element of its parent
<code>p:only-of-type</code>	Selects every <p> element that is the only <p> element of its parent
<code>p:only-child</code>	Selects every <p> element that is the only child of its parent

<code>input:optional</code>	Selects <input> elements with no 'required' attribute
<code>input:out-of-range</code>	Selects <input> elements with a value outside a specified range
<code>input:read-only</code>	Selects <input> elements with the 'readonly' attribute specified
<code>input:read-write</code>	Selects <input> elements with the 'readonly' attribute not specified
<code>input:required</code>	Selects <input> elements with the 'required' attribute specified
<code>:root</code>	Selects the documents root element
<code>::selection</code>	Selects the portion of an element that is selected by a user
<code>#id:target</code>	Selects the current active #id element
<code>input:valid</code>	Selects all <input> elements with a valid value