

CSS Pseudo cheat sheet

Simple selectors

Selector	Syntax	Example
Element	element	<code>div {</code>
Class	.class	<code>.alpha {</code>
ID	#id	<code>#alpha {</code>
Universal	*	<code>* {</code>

Variations of simple selectors

Elements	Syntax	Example	Description
Two classes	.first-class.second-class	<code>.alpha .beta {</code>	All elements with classes alpha and beta
Element and class	element.class	<code>p.alpha {</code>	All alpha class elements inside <p>
Two elements	element, element	<code>p, div {</code>	All <p> and <div> elements
Two elements	element element	<code>p div {</code>	All <div> elements inside <p>

Descendant selectors/combinators

Selector	Syntax	Example	Description
Descendant	element element	<code>div p {</code>	All <p> descendants of <div>
Child	element>element	<code>div > p {</code>	All <p> direct descendants of <div>
Adjacent Sibling	element+element	<code>div + p {</code>	<p> element directly after <div>
General Sibling	element~element	<code>div ~ p {</code>	All <p> element iterations after <div>

Attribute selectors

Selector	Syntax	Example
[attribute]	[href]	Selects all elements with a href attribute
[attribute=value]	[lang="fr"]	Selects all elements with lang attribute that has a value of "fr"
[attribute~=value]	[input~=hello]	Elements with input attribute containing the whitespace separated substring "hello"
[attribute =value]	[lang =en]	Elements with lang attribute value equal to "en" or "en-(en hyphen)"
[attribute^=value]	a[href^="https"]	Every <a> element with href attribute value begins with "https"
[attribute\$=value]	a[href\$=".docx"]	Every <a> element with href attribute value ends with ".docx"
[attribute*=value]	a[href*="meta"]	Every <a> element with href attribute value has substring "meta"

Pseudo-class	Example	Description of selection
:active	a:active	All active links
:checked	input:checked	All the checked <input> elements
:default	input:default	All default <input> elements
:disabled	input:disabled	All disabled <input> elements

:empty	<code>div:empty {</code>	All the <div> elements with no children
:enabled	<code>input:enabled {</code>	All the enabled <input> elements
:first-child	<code>p:first-child {</code>	All the <p> elements who are the first child of a parent element
:first-of-type	<code>p:first-of-type {</code>	All the <p> element who are the first <p> element of a parent element
:focus	<code>input:focus {</code>	Input element under focus
:fullscreen	<code>:fullscreen {</code>	The element in full-screen mode
:hover	<code>p:hover {</code>	Action effect on mouse hover
:invalid	<code>input:invalid {</code>	Input elements with an invalid value
:last-child	<code>p:last-child {</code>	All the <p> elements who are the last child of a parent element
:last-of-type	<code>p:last-of-type {</code>	All the <p> elements who are the last <p> element of a parent element
:link	<code>a:link {</code>	All unvisited links
:not(selector)	<code>:not(div) {</code>	All the elements that are not a <div> element
:nth-child(n)	<code>div:nth-child(3) {</code>	All the <p> elements that are the third child of a parent element
:nth-last-child(n)	<code>child(3) {</code>	All the <div> elements which are the third child of a parent element, counting from last child element
:nth-last-of-type(n)	<code>p:nth-last-of-type(2) {</code>	The second sibling from the last child of a parent element.
:nth-of-type(n)	<code>p:nth-of-type(2) {</code>	The second sibling of a parent element.
:only-of-type	<code>p:only-of-type {</code>	All the <p> elements which are only <p> elements inside its parent
:only-child	<code>p:only-child {</code>	All the <p> elements which are only child of a parent element
:optional	<code>input:optional {</code>	The input elements with no "required" attribute
:required	<code>input:required {</code>	Selects input elements with the "required" attribute specified
:root	<code>:root {</code>	The Root element of document
::selection	<code>::selection {</code>	The portion of an element that is selected by a user
::valid	<code>input:valid {</code>	All the input elements with a valid value
::visited	<code>a:visited {</code>	Selects all visited links

Pseudo-element selectors

Syntax	Example	Description
::after	<code>p::after { }</code>	Inserts content after content of <p> element
::before	<code>p::before { }</code>	Inserts content before content of <p> element
::first-letter	<code>p::first-letter { }</code>	Selects first letter of every <p> element
::first-line	<code>p::first-line { }</code>	Selects first line of every <p> element
::placeholder	<code>input::placeholder</code>	Selects input elements with "placeholder" attribute specified
::marker	<code>::marker {</code>	Selects markers in a list