

CSS Pseudo cheat sheet

Simple selectors

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

Variations of simple selectors

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

Descendant selectors/combinators

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

Attribute selectors

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

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

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

Pseudo-element selectors

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