CSS Pseudo cheat sheet

Simple selectors

Selector	Syntax	Example

Element	element	`div {`
		`}`

Class	.class	`.alpha {`
		`}`

ID	#id	`#alpha {`
		`}`

Universal	*	`* {`
		`}`

Variations of simple selectors

Elements	Syntax	Example	Description

Two classes	.first-class.second-class	`.alpha.beta`
		`{`
		`}`
			All elements with classes alpha and beta

Element and class	element.class	`p.alpha {`
		`}`
			All alpha class elements inside <p>

Two elements	element, element	`p, div {`
		`}`
			All <p> and <div> elements

Two elements	element element	`p div {`
		`}`
			All <div> elements inside <p>

Descendant selectors/combinators

Selector	Syntax	Example	Description

Descendant	element element	`div p {`
		`}`
			All <p> descendants of <div>

Child	element>element	`div > p {`
		`}`
			All <p> direct descendants of <div>

Adjacent Sibling	element+element	`div + p {`
		`}`
			<p> element directly after <div>

General Sibling	element~element	`div ~ p {`
		`}`
			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"

	`a[href$=".docx"]`	

[attribute$=value]	`{`
	`}`
		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	`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*)	`div:nth-last-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

Mark as completed

👍 Like 👎 Dislike ⚑ Report an issue