

CSS Pseudo Cheat Sheet

(Source: <https://www.coursera.org/learn/html-and-css-in-depth/supplement/zbsTK/css-pseudo-cheat-sheet>)

Simple Selectors

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

Variations of Simple Selectors

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

Pseudo-class Selectors

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	<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

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

Pseudo-class	Example	Description of selection
:visited	<i>a:visited { }</i>	Selects all visited links

Pseudo-class Elements Selectors

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