

Pseudo-class	Examples	Example description
<u>:active</u>	a:active	Selects the active link
<u>:any-link</u>	a:anylink area:anylink	Selects any <a> or <area> element with an href attribute
<u>:auto-fill</u>	input:autofill	Selects any <input> element with its value autofilled by the browser
<u>:checked</u>	input:checked option:checked	Matches any <input> or <option> element that is checked
<u>:default</u>	input:default button:default option:default	Selects form elements that are default in a group of related elements
<u>:defined</u>	:defined	Selects any element that has been defined (standard or custom elements)
<u>:dir()</u>	:dir(ltr) :dir(rtl)	Selects any element with the specified text direction
<u>:disabled</u>	:disabled input:disabled option:disabled	Selects any element that is disabled. Most used for form elements
<u>:empty</u>	div:empty	Selects any element that has no children (including text nodes)
<u>:enabled</u>	:enabled input:enabled	Selects any element that is enabled. Most used for form elements
<u>:first</u>	@page :first	Represents the first page of a printed document (used with the @page rule)
<u>:first-child</u>	p:first-child li:first-child	Selects the element that is the first child of its parent (among a group of sibling elements)
<u>:first-of-type</u>	p:first-of-type li:first-of-type	Selects the first element of its type among a group of sibling elements
<u>:focus</u>	input:focus select:focus	Selects the element that gets focus. Most used for form elements
<u>:focus-visible</u>	button:focus-visible	Selects the element that gets focus (used to apply focus styles only when

		the keyboard is used to focus something, not the mouse)
<u>:focus-within</u>	form:focus-within label:focus-within	Matches an element if the element or any of its descendants gets focus
<u>:fullscreen</u>	:fullscreen	Selects any element that is currently in full-screen mode
<u>:has()</u>	h2:has(+p)	Selects h2 elements that are immediately followed by a p element, and applies the style to h2
<u>:hover</u>	a:hover p:hover	Selects element on mouse over
<u>:in-range</u>	input:in-range	Select any <input> element with a value within the specified range limit
<u>:indeterminate</u>	input:indeterminate	Selects any form element that is in an indeterminate state
<u>:invalid</u>	input:invalid fieldset:invalid	Selects invalid form elements
<u>:is()</u>	:is(ul, ol)	Selects all and elements
<u>:lang()</u>	p:lang(it)	Selects any <p> element with a lang attribute equal to "it" (Italian)
<u>:last-child</u>	li:last-child	Selects any element that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects any <p> element that is the last <p> element of its parent
<u>:left</u>	@page :left	Represents all left-hand pages of a printed document (used with the @page rule)
<u>:link</u>	a:link	Selects any unvisited link
<u>:modal</u>	:modal	Selects the element that is in a modal state
<u>:not()</u>	:not(p)	Selects any element that is not a <p> element

<u>:nth-child()</u>	p:nth-child(2)	Selects any <p> element that is the second child of its parent
<u>:nth-last-child()</u>	p:nth-last-child(2)	Selects any <p> element that is the second child of its parent, counting from the end
<u>:nth-last-of-type()</u>	p:nth-last-of-type(2)	Selects any <p> element that is the second <p> element of its parent, counting from the end
<u>:nth-of-type()</u>	p:nth-of-type(2)	Selects any <p> element that is the second <p> element of its parent
<u>:only-child</u>	p:only-child	Selects any <p> element that is the only child of its parent
<u>:only-of-type</u>	p:only-of-type	Selects any <p> element that is the only <p> element of its parent
<u>:optional</u>	input:optional select:optional textarea:optional	Selects any <input>, <select>, or <textarea> elements without a "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects any <input> element with a value outside the specified range limit
<u>:placeholder-shown</u>	input:placeholder-shown textarea:placeholder-shown	Selects any <input> or <textarea> element that is currently displaying placeholder text
<u>:popover-open</u>	:popover-open	Selects any element that is in a showing popover state
<u>:read-only</u>	input:read-only	Selects input elements with the "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects editable input elements
<u>:required</u>	input:required	Selects input elements with the "required" attribute specified
<u>:right</u>	@page :right	Represents all right-hand pages of a printed document (used with the @page rule)
<u>:root</u>	:root	Selects the document's root element

<code>:scope</code>	<code>:scope</code>	Selects elements that are a reference point, or scope, for selectors to match against
<code>:state()</code>	<code>:state()</code>	Selects custom elements that have the specified custom state
<code>:target</code>	<code>:target</code>	Selects the current active target element
<code>:user-invalid</code>	<code>:user-invalid</code>	Selects any form element with an invalid value (after the user have interacted with it)
<code>:user-valid</code>	<code>:user-valid</code>	Selects any form element with a valid value (after the user have interacted with it)
<code>:valid</code>	<code>input:valid</code>	Selects all input elements with a valid value
<code>:visited</code>	<code>a:visited</code> <code>area:visited</code>	Selects all visited links
<code>:where()</code>	<code>:where(ol, ul)</code>	Selects all <code></code> and <code></code> elements