| **Selector** | **Description** | **CSS / Selector Level** |
| --- | --- | --- |
| \* | Selects all elements. | 2 |
| E | Selects an element of type E | 1 |
| E:not(*s1*, *s2*) | Selects an E element that does not match either *s1* or *s2*. | 3/4 |
| E:matches(*s1*, *s2*) | Selects an E element that matches *s1* and/or *s2*. | 4 |
| E:has(*rs1*, *rs2*) | Selects an E element, if either of the relative selectors *rs1* or *rs2*, when evaluated with E as the :scope elements, match an element. | 4 |
| E.classname | Selects an E element belonging to the class named classname. | 1 |
| E#myid | Selects an E element with an ID of myid. | 1 |
| E[foo] | Selects an E element with a foo attribute. | 2 |
| E[foo="bar"] | Selects an E element whose foo attribute value is exactly equal to bar. | 2 |
| E[foo="bar" i] | Selects an E element whose foo attribute value is exactly equal to bar, regardless of its case. Basically, using i allows you to specify "case-sensitive" when specifying the value.  So for example, Text, text, and TEXT will all be selected if i is specified. | 4 |
| E[foo~="bar"] | Selects an E element whose foo attribute value is a list of whitespace-separated values, one of which is exactly equal to bar. | 2 |
| E[foo^="bar"] | Selects an E element whose foo attribute value begins exactly with the string bar. | 3 |
| E[foo$="bar"] | Selects an E element whose foo attribute value ends exactly with the string bar. | 3 |
| E[foo\*="bar"] | Selects an E element whose foo attribute value contains the substring bar. | 3 |
| E[foo|="fruit"] | Selects an E element whose foo attribute value is a hyphen-separated list of values beginning with en. | 2 |
| E:dir(ltr) | Selects an element of type E in with left-to-right directionality (the document language specifies how directionality is determined). | 4 |
| E:lang(zh, "\*-hant") | Selects an element of type E tagged as being either in Chinese (any dialect or writing system) or othewise written with traditional Chinese characters. Simply replace zh with the applicable language code, and \*-hant with the applicable character code. | 2/4 |
| E:any-link | Selects an E element being the source anchor of a hyperlink.  This is the equivalent of using both the :link and :visited pseudo-classes. | 4 |
| E:link | Selects an E element being the source anchor of a hyperlink of which the target is not yet visited. | 1 |
| E:visited | Selects an E element being the source anchor of a hyperlink of which the target is already visited. | 1 |
| E:target | Selects an E element being the target of the referring URL. | 3 |
| E:scope | Selects an E element being a designated reference element. | 4 |
| E:current | Selects an E element that is currently presented in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the current item as it is being read out. | 4 |
| E:current(*s*) | Selects an E element that is the deepest :current element that matches selector *s*. | 4 |
| E:past | Selects an E element that is in the past in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the items that have already been read out. | 4 |
| E:future | Select an E element that is in the future in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the items that have yet to be read out. | 4 |
| E:active | Selects an E element that is in an activated state. | 1 |
| E:hover | Selects an E element that is under the cursor, or that has a descendant under the cursor. | 2 |
| E:focus | Selects an E element that has user input focus. | 2 |
| E:drop | Selects an E element that can possibly receive a drop.  For example, in HTML you can specify a drop target with the dropzone attribute. This selector can be used style those elements while the user is dragging another element to be dropped. | 4 |
| E:drop(active) | Selects an E element that is the current drop target for the item being dragged.  In other words, if the user were to release the drag, the element would be dropped on to this drop target. | 4 |
| E:drop(valid) | Selects an E element that could receive the item currently being dragged. It only matches if the drop target is valid for the object currently being dragged.  For example, the HTML dropzone attribute can be used to specify which types of files can be dropped onto an element. | 4 |
| E:drop(invalid) | Selects an E element that cannot receive the item currently being dragged, but could receive some other item. | 4 |
| E:enabled | Selects a user interface element E that is enabled. | 3 |
| E:disabled | Selects a user interface element E that is disabled. | 3 |
| E:read-write | Selects a user interface element E that is user alterable.  For example, an HTML [input](https://www.quackit.com/html/tags/html_input_tag.cfm) element that is not disabled and not readonly is :read-write. Also, any HTML element with contenteditable attribute set to the true state is also :read-write. | 3-UI/4 |
| E:read-only | Selects a user interface element E that is not user alterable.  For example, an HTML [input](https://www.quackit.com/html/tags/html_input_tag.cfm) element that is disabled or readonly is :read-only. | 3-UI/4 |
| E:placeholder-shown | Selects an input control currently showing placeholder text.  This allows you to style the placeholder text. For example, change its color, etc. | 3-UI/4 |
| E:default | Selects a user interface element E that is the default item in a group of related choices.  For example, it could select the initially selected [option](https://www.quackit.com/html/tags/html_option_tag.cfm) element in a [select](https://www.quackit.com/html/tags/html_select_tag.cfm) element, or the default button in the form, etc. | 3-UI/4 |
| E:checked | Selects a user interface element E that is checked/selected (for example, a radio-button or checkbox).  Can be used to style the element's label. | 3 |
| E:indeterminate | Selects a user interface element E that is in an indeterminate state (neither checked nor unchecked).  Can be used to style the element's label. | 4 |
| E:valid | Selects a user-input element E that meets its data validity semantics. | 3-UI/4 |
| E:invalid | Selects a user-input element E that doesn't meet its data validity semantics. | 3-UI/4 |
| E:in-range | Selects a user-input element E whose value is in-range. | 3-UI/4 |
| E:out-of-range | Selects a user-input element E whose value is out-of-range. | 3-UI/4 |
| E:required | Selects a user-input element E that requires input.  For example, a form element with the required attribute. | 3-UI/4 |
| E:optional | Selects a user-input element E that does not require input.  For example, a form element without the required attribute. | 3-UI/4 |
| E:user-error | Selects a user-altered user-input element E with incorrect input (invalid, out-of-range, omitted-but-required). | 4 |
| E:root | Selects an E element, root of the document.  In most cases, when using HTML this will match the [html](https://www.quackit.com/html/tags/html_html_tag.cfm) element, however, this may not always be the case if using another language such as SVG or [XML](https://www.quackit.com/xml/tutorial/). | 3 |
| E:empty | Selects an E element that has no children (not even text nodes). | 3 |
| E:blank | Selects an E element that has no content except maybe white space.  Similar to :empty, however, :blank will also select elements that contain white space (which :empty will not). | 4 |
| E:nth-child(*n* [of *S*]?) | Selects an E element, the *n*-th child of its parent matching *S*. | 3 |
| E:nth-last-child(*n* [of *S*]?) | Selects an E element, the *n*-th child of its parent matching *S*, counting from the last one. | 3 |
| E:first-child | Selects an E element, first child of its parent. | 2 |
| E:last-child | Selects an E element, last child of its parent. | 3 |
| E:only-child | Selects an E element, only child of its parent. | 3 |
| E:nth-of-type(*n*) | Selects an E element, the *n*-th sibling of its type. | 3 |
| E:nth-last-of-type(*n*) | Selects an E element, the *n*-th sibling of its type, counting from the last one. | 3 |
| E:first-of-type | Selects an E element, first sibling of its type. | 3 |
| E:last-of-type | Selects an E element, last sibling of its type. | 3 |
| E:only-of-type | Selects an E element, only sibling of its type. | 3 |
| E F | Selects an F element descendant of an E element. | 1 |
| E >> F | Selects an F element descendant of an E element.  Note that this selector does the same thing as the previous selector (E F). This selector was introduced in CSS Selectors Level 4 to give the descendant combinator a visible, non-whitespace form and to bridge the gap between the child combinator (>) and the shadow-piercing descendant combinator (>>>) | 4 |
| E > F | Selects an F element child of an E element. | 2 |
| E + F | Selects an F element immediately preceded by an E element. | 2 |
| E ~ F | Selects an F element preceded by an E element. | 3 |
| F || E | Selects an E element that represents a cell in a grid/table belonging to a column represented by an element F. | 4 |
| E:nth-column(*n*) | Selects an E element that represents a cell belonging to the *n*th column in a grid/table. | 4 |
| E:nth-last-column(*n*) | Selects an E element that represents a cell belonging to the *n*th column in a grid/table, counting from the last one. | 4 |
| :playing | Selects an element representing an audio, video, or similar resource that is capable of being "played" or "paused", when that element is "playing". | 4 |
| :paused | Selects an element representing an audio, video, or similar resource that is capable of being "played" or "paused", when that element is "paused". This includes both an explicit "paused" state, and other non-playing states like "loaded, hasn't been activated yet", etc. | 4 |