

Pseudo Classes

Assist in selecting something that can't be expressed by a simple selector

Order is important when dealing with anchor tags

:hover must be after :link and :visited

:active must be after :hover

try to keep them in the same order

:link, :visited, :hover, :active (**LoVe HAte**)

List of CSS Pseudo Classes - https://www.w3schools.com/css/css_pseudo_classes.asp

| SELECTOR | EXAMPLE | EXAMPLE DESCRIPTION |
|----------------|--------------------|---|
| :root | :root | Selects the document's root element |
| :not(selector) | :not(p) | Selects every element that is not a <p> element |
| :active | a:active | Selects the active link |
| :hover | a:hover | Selects links on mouse over |
| :link | a:link | Selects all unvisited links |
| :visited | a:visited | Selects all visited links |
| :target | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| :focus | input:focus | Selects the <input> element that has focus |
| :checked | input:checked | Selects every checked <input> element |
| :disabled | input:disabled | Selects every disabled <input> element |
| :enabled | input:enabled | Selects every enabled <input> element |
| :in-range | input:in-range | Selects <input> elements with a value within a specified range |
| :invalid | input:invalid | Selects all <input> elements with an invalid value |
| :optional | input:optional | Selects <input> elements with no "required" attribute |
| :out-of-range | input:out-of-range | Selects <input> elements with a value outside a specified range |
| :read-only | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| :read-write | input:read-write | Selects <input> elements with no "readonly" attribute |
| :required | input:required | Selects <input> elements with a "required" attribute specified |
| :valid | input:valid | Selects all <input> elements with a valid value |

List of CSS Pseudo Classes - https://www.w3schools.com/css/css_pseudo_classes.asp

| | | |
|----------------------|-----------------------|--|
| :empty | p:empty | Selects every <p> element that has no children |
| :first-child | p:first-child | Selects every <p> elements that is the first child of its parent |
| :first-of-type | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| :lang(language) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| :last-child | p:last-child | Selects every <p> elements that is the last child of its parent |
| :last-of-type | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| :nth-child(n) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| :nth-last-child(n) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| :nth-last-of-type(n) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| :nth-of-type(n) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| :only-of-type | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| :only-child | p:only-child | Selects every <p> element that is the only child of its parent |