



[Course](#) > [Modul...](#) > [5.5 CSS...](#) > Pseudo...

Pseudo classes and cursor (OPTIONAL)

Pseudo classes and cursor (OPTIONAL)

Note: *This section is optional material included for the curious. It will not appear on any graded question.*

Refined CSS selectors - pseudo classes

If you have a page with some links on it, and you look at them carefully, you may notice that some of the links that you've visited before are a different color than those you haven't (purple versus blue, typically). Plus, if the mouse is brought over a link, it may change color again, or highlight in some other way. In addition, if you click down and don't let the mouse down, then maybe the link may change yet again. Lastly, if you visit some other sites and make these same explorations, you may observe visual differences from site to site. You may conclude that the styling of the elements are being changed, and you will be right. How did the author of the style sheet know which links you've been to already?

"Pseudo Classes" is a fancy term for simply being able to refine our CSS selection to something that isn't just another element. Pseudo Classes allow us to apply styles to the different states of an element. Or to various children of an element based on their index, or to other interactions with the browser. Best of all, pseudo classes are easy to use.

```
a:visited { color: purple; }
```

Above us we see a tag selector (`a`) followed by a pseudo class, which consists of a colon and a word (e.g. `:visited`). This particular CSS rule will be applied to any `<a>` tag that the user has already visited. There can be no spaces on either side of the colon. Pseudo classes can be amended onto *any* CSS selector, not just tag selectors.

A full list of pseudo classes can be found [here](#). Let's look at some of the most common ones.

:visited

```
a:visited { color: purple; }
```

The `:visited` pseudo class is usually put on a selector that resolves to an `<a>` tag. It enables you to define a style for the visited state of the link. For example, if the user has already been to that Web site, the `:visited` style will be applied.

:hover / :active

```
li:hover { background-color: red; }  
li:active { background-color: green; }
```

The `:hover` pseudo class lets you change the style for an element when the mouse is hovering above it. The `:active` pseudo class is applied when the mouse is depressed into its area. Note that the mouse is rarely hovering or clicking over/into "just one" item. At any given moment, the mouse is usually over several elements, because if it is over a child element, it will be over the parent, grandparent, and great grandparent.

Therefore, if you have two different style rules, such as `li:hover` and `ul:hover`, then they will *both* be activated, when the mouse is over one of the list items.

CSS	Result (Try It!)
<pre>li:hover { background-color: lightblue; } li:active { background-color: red; } ul:active { border: 1px solid pink; }</pre>	<div>shark</div> <div>marlin</div> <div>tuna</div> <div>whale</div> <div>koi</div> <div>barracuda</div> <div>octopus</div>

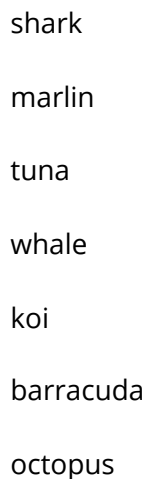
:nth-child

```
tr:nth-child(odd) { background-color: lightgray; }  
tr:nth-child(even) { background-color: white; }
```

The `:nth-child` pseudo class is very handy. Unlike the pseudo classes we've seen so far, it expects a parameter. The pseudo class always ends with a pair of parentheses with an expression inside. This expression is simply the term `odd` or `even`, however, there are other more advanced possibilities with mathematical equations containing the term `n`.

This selector is most commonly used to apply "transaction ledger" styles to tables or lists.

Example: `nth-child(even)`



A vertical list of fish names: shark, marlin, tuna, whale, koi, barracuda, octopus. The items 'marlin', 'whale', and 'octopus' are highlighted with a light gray background, demonstrating the :nth-child(even) selector.

- shark
- marlin
- tuna
- whale
- koi
- barracuda
- octopus

Cursor property

```
li { cursor: pointer; }
```

Since we've broached the topic of mouse-responding pseudo classes, it makes sense to also cover the `cursor` CSS property.

The `cursor` CSS property lets you change the cursor that is displayed when the mouse is over the element in question. This does not have to be relegated to a `:hover` style, it can be applied anywhere. Though, if you want to change the cursor when an element is depressed, set the `cursor` property in the context of an `:active` style.

There are many possible values for the `cursor` property. Their exact representation may vary slightly from browser to browser. Common values include: `default`, `pointer`, `text`, `move` and `grab`. (Hover over each value to see). In addition, some browsers support a custom image as well (`cursor: url("images/my_pointer.png");`).

For more information, please visit the [MDN page on cursor](#).

Knowledge check 5.5.5 (not graded)

0 points possible (ungraded)

You are developing a Web page, and desire that the hyperlinks italicize as the mouse is over them. Which of these rules will achieve that?

☐ This cannot be achieved without JavaScript

☐ `mouseover { font-style: italic; }`

☐ `a:hover { font-style:italic; }`

☐ `a { font-style: italic; }`

Submit

© All Rights Reserved