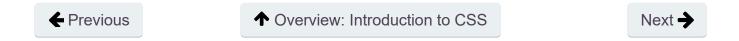


Pseudo-classes and pseudo-elements



In this, the third in our series of articles on selectors, we discuss **pseudo-selectors**. These don't select elements, but rather certain parts of elements, or elements only in certain contexts. They come in two main types: **pseudo-classes** and **pseudo-elements**.

Pseudo-classes &

A CSS pseudo-class is a keyword added to the end of a selector, preceded by a colon (:), which is used to specify that you want to style the selected element but only when it is in a *certain state*. For example, you might want to style a link element only when it is being hovered over by the mouse pointer, or a checkbox when it is disabled or checked, or an element that is the first child of its parent in the DOM tree.

:active	:indeterminate	:only-child
:checked	:in-range	:only-of-type
:default	:invalid	:optional
:dir	:lang	:out-of-range
:disabled	:last-child	:read-only
:empty	:last-of-type	:read-write
:enabled	:left	:required
:first	:link	:right
:first-child	:matches()	:root
:first-of-type	:not	:scope
:fullscreen	:nth-child	:target
:focus	:nth-last-child	:valid
:focus-within	:nth-last-of-type	:visited
:hover	:nth-of-type	

We will not dig into every pseudo-class right now, it is not the objective of the Learning Area to teach you everything in *exhaustive detail*. You'll certainly come across some of these in more detail later, where appropriate.

For now, let's just see a simple example of how to use these. First, an HTML snippet:

```
1 | <a href="https://developer.mozilla.org/" target="_blank">Mozilla Developer Network</a>
```

Then, some CSS rules:

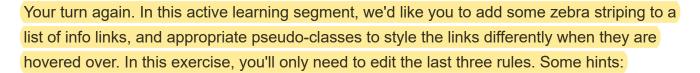
```
/* These styles will style our link
1
       in all states */
2
3
    a {
      color: blue;
4
      font-weight: bold;
5
6
 7
    /* We want visited links to be the same color
8
       as non visited links */
9
    a:visited {
10
      color: blue;
11
12
13
    /* We highlight the link when it is
14
       hovered over (mouse over), activated (mouse down)
15
       or focused (keyboard) */
16
    a:hover,
17
    a:active,
18
    a:focus {
19
      color: darkred;
20
21
```

```
22     text-decoration: none;
}
```

Now let's play with the link we just styled!

Mozilla Developer Network

Active learning: A zebra striped info list



- 1. You have already seen how to write the second pseudo class for the hover styles.
- 2. For the zebra striping, you'll need to use a pseudo class like :nth-of-type(). This gives the two color rules a slightly different version of the pseudo class, to style the even and odd numbered list items. See if you can find how to do this!

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

HTML Input

```
<a href="#">United Kingdom</a>
<a href="#">Germany</a>
<a href="#">Finland</a>
```



CSS Input

```
ul {
  padding: 0;
}
li {
  padding: 3px;
  margin-bottom: 5px;
  list-style-type: none;
}
```

Output

United Kingdom

Germany

Finland

Russia

Spain

Reset

Show solution

Pseudo-elements &

Pseudo-elements are very much like pseudo-classes, but they have differences. They are keywords, this time preceded by two colons (::), that can be added to the end of selectors to select a certain part of an element.

- ::after
- ::before
- ::first-letter
- ::first-line
- ::selection
- ::backdrop

They all have some very specific behaviors and interesting features, but digging into them all in detail is beyond our scope for now.

A pseudo-element example 🔗

Here we'll just show a simple CSS example that selects the space just after all absolute links and adds an arrow in that space:

Let's add this CSS rule:

```
/* All elements with an attribute "href" with values
1
      starting with "http" will have an arrow added after their
       content (to indicate they are external links) */
    [href^=http]::after {
4
     content: '♪';
5
6
```

We get this result:

- CSS defined in the MDN glossary.
- HTML defined in the MDN glossary.

Active learning: A fancy paragraph



Next up for active learning, we have a fancy paragraph to style! All you have to do here is apply the two rulesets, to the paragraph's first line and first letter, using the ::first-line and ::first-letter pseudo elements. This should style the first line of the paragraph in bold, and the first letter as a nice drop cap, giving it an old-style feel.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the Show solution button to see a potential answer.

⊓ I WIL IIIPUL

```
This is my very important paragraph.
I am a distinguished gentleman of such renown that my paragraph needs to be styled in a manner befitting my majesty. Bow before my splendour, dear students, and go forth and learn CSS!
```

CSS Input

```
{
  font-weight: bold;
}

{
  font-size: 3em;
  border: 1px solid black;
  background: red;
  display: block;
  float: left:
```

Output

This is my very important paragraph. I am a distinguished gentleman of such renown that my paragraph needs to be styled in a manner befitting my majesty. Bow before my splendour, dear students, and go forth and learn CSS!

Reset

Show solution

Next up 🔗

We will round off our tour of CSS selectors, by looking at Combinators and multiple selectors.







In this module §

- How CSS works
- CSS syntax
- Selectors
- Simple selectors
- Attribute selectors
- Pseudo-classes and pseudo-elements
- Combinators and multiple selectors
- CSS values and units

- Cascade and inheritance
- The box model
- Debugging CSS
- Fundamental CSS comprehension