

# Pseudo-classes and pseudo-elements

[← Previous](#)[↑ Overview: Introduction to CSS](#)[Next →](#)

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.

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

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.

A pseudo-class example [!\[\]\(99f58673407353e96a019fbca558fd72\_img.jpg\)](#)

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:

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

Your turn again. In this active learning segment, we'd like you to add some zebra striping to a list of info links, and appropriate pseudo-classes to style the links differently when they are hovered over. In this exercise, you'll only need to edit the last three rules. Some hints:



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

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

```
<li><a href="#">Finland</a></li>
<li><a href="#">Russia</a></li>
<li><a href="#">Spain</a></li>
<li><a href="#">Poland</a></li>
</ul>
```

## CSS Input

```
ul {
  padding: 0;
}

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

## Output

United Kingdom  
Germany  
Finland  
Russia  
Spain  
Poland

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:

```
1 <ul>
2   <li><a href="https://developer.mozilla.org/en-US/docs/Glossary/CSS">CSS</a> defined in the MDN glossary
3   <li><a href="https://developer.mozilla.org/en-US/docs/Glossary/HTML">HTML</a> defined in the MDN glossary
4 </ul>
```

Let's add this CSS rule:

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

HTML Input

## HTML Input

```
<p>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!</p>
```

## 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](#).

[← Previous](#)

[↑ Overview: Introduction to CSS](#)

[Next →](#)

---

## 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
-