

# CSS Pseudo-classes

Hello and welcome to the lesson about the CSS pseudo-classes. In the previous lesson, we've looked in detail into creating CSS selectors, and that included using a CSS class. But we also have an additional way to select things by using what is called a pseudo-class. So pseudo-classes in CSS are a type of selector that selects elements that are in a specific state. For example, they are the first element of their type or they are being hovered over by the mouse pointer. And there is over 30 different pseudo-classes.

## Pseudo-class Syntax

But I am going to explain just a few, which I think are the most popular and useful, and the syntax is that they are using a colon. So you have your selector, colon, pseudo-class, and then the rest of the ruleset with declaration and a CSS property and value. So let me show you an example. First, we'll look at anchor pseudo-classes, pseudo-classes used with links. So as you can see here, I've changed our structure a bit. But if we go into Style CSS file, you will see that I've included four different pseudo-classes for "a" elements. So "a" link, link is a pseudo-class that targets links that were not visited. So if in your browser history the browser doesn't recognize the URL that's provided inside the "href" attribute in a link then it will use this color so that's our teal color. And hover, you probably already know that, the hover pseudo-class selects elements when we are hovering over elements with our mouse. Visited, selects links which are in opposed to the link pseudo-class which URL is

inside browser history. When the browser recognizes that you've already visited that URL, you know - by default, they have this purple color. When you, for example, Google something in Google page results you would sometimes see that some of the websites are using that default purple color that's your visited link color.

### **Active Pseudo-class**

And finally, we have an active pseudo-class, and this one may be confusing, and you may think that it's for your active menu items where you are on a certain page that using the active pseudo-class would somehow select that link corresponding to that page, but that's not the case. That active pseudo-class is for when you click when the link is an active state, meaning when you are doing the click, that's the active pseudo-class.

### **Hovering over elements**

And what's interesting about hover or maybe not interesting but what's important to note that the pseudo-class doesn't need to be the last portion of your selector. So for example, if I would like to select, change my H2 when I hover over this selection here, I remember that this footer section had a class content, so if I hover over the content I can change its background, for example. So we can see that it works. You see, when I hover over my content section, the background changes, but I can also target elements inside - use that selector as a parent container and target my H2 inside and also change its color, 3F is for white. And now if I hover over my section, my H2 color changes. It's very useful to know that and in Divi, I use it quite a lot when I want to hover over a column, for example, and I want to change a blurb color inside or a button inside. So hovering over parent container, you can target the sentence element.

## First-child element

Another type of pseudo-class is the first-child element. So if we would want to target our list item and say we want the color to be red. Okay, so that makes all the text inside our list red except for links, because browser do have the default colors for links. So that doesn't apply to links, but the regular text will be affected. But we can specify the first-child and say we want first child to be green. And that includes the first element here. But also if we go to see our page source, that would also be considered a first-child because it's basically first-child inside the parent.

## First-of-type selector

Now, there is a similar selector, first of type, and that one works a bit different. So let's say I want to target "p" paragraph, that's also first child. Let's maybe remove that. And you see, this doesn't affects my "p", because if we look at our page source, the "p" it's not a first child of the parent container. The first child is H1, but we can use first of type, which selects just like it's called first element of its type, first of type, and now it applies to this "p" tag, even though it's not a first child of that div.

## Nth-child

Okay, so we have first child, first of type, but we also have nth-child. So let's go back to our lis and let's do nth-child and then in the brackets, we can specify the number. So let's say I want to select the second list item. You see that will apply to the sub ul as well, and to the main ul. I can do every second using the N, so second, fourth, sixth, and so on. And that as I'm selecting a parent that includes all the child elements as well.

## Nth-last-child

And another similar selector would be `nth-last-child`. And similarly, we can use a number here and that would count this from the `nth`. So `second to last` or `li: last-child` that would select the last one or last of type. That's the same thing because the `lis` are here, but it would work on our `"p"` as its last and also the first, but `last-child` will not work on that `"p"` because `"p"` is not the last-child. The `ul` is the last child of the parent container.

I hope it's clear and helpful. I use these pseudo-classes all the time, so it's important that you kind of understand that you can apply it to any selector, not only at the end not only to the thing you are targeting, but you can use that on the parent element. And when you're targeting something inside that parent.