CSS Selectors

Hello, and welcome to the lesson about the CSS selectors. As we learned in the lesson about CSS syntax selector is the part of the CSS ruleset that specifies our target. That includes elements, CSS classes, IDs. And in this lesson, I would like to explain exactly what are the different, simplest and most popular ways to construct CSS selectors and target website elements, and how that specifically applies to the Divi theme structure.

But before we focus on Divi, let's look at the simple HTML structure first. Heading level one, another heading with a class "subtitle", paragraph with a class "subtitle", a container div with ID "main-content" and article that has a heading and a paragraph inside. Three articles here and the paragraph. Next, a div with ID "footer" with another H2 heading, unordered list, which has two different CSS classes and some list items and a second level, unordered list witha class "sub-menu" with additional list items and that's it. This is a simple structure. And now I would like to show you how we can select all of these different elements with CSS.

I will create a style tag here. And now I can write my CSS and we will see it. Because that's how the browser see all these elements when there's no CSS applied. That's the default look for all that HTML structure. So we can select elements itself. So. If I select h2 and then specify property and value. For example, color orange that will target all h2's throughout the page.

So I have h2 here at the top and then h2 's inside the articles and in the footer div as well. Okay. So this is probably not the very common way to target things, because usually you want to be more specific when targeting elements. That's

when we can use a CSS class. So here that first h2 has a class "subtitle".

Targeting CSS classes

To target CSS class, we are using dot. ".subtitle" means h2 with a class subtitle. And now my CSS ruleset only applies to h2s, which have that CSS class. And I have only one. Now, we do not need to specify element itself with a class. We can only target just the class. Like this, and as you can see: this way, we are targeting all elements with that particular class.

I have h2 with that class. And also the paragraph below also uses that class. That's why we are targeting both of these. So these are classes. Now, we can also target IDs.

Targeting CSS IDs

For example, this article, the first one has ID "featured". To target ID instead of dot we are using a hashtag symbol.

This selector targets element with ID featured and because the color CSS property is inherited - everything inside that element will use that same value. That's why the h2 and p is also orange. So everything inside that ID, will use that CSS property. We have IDs, classes, elements itself, and we can combine all this together.

Targeting multiple attributes

So for example, if I would like to target "content" class. Okay. So that targets all these articles. We have. The first article has "content" and "sticky" class. And these two only have the "content" class. So if I want to target the first one, I could use ".content.sticky". That selector... There's no space here. So that applies to one element: one element with these two classes.

If I would only use a single class ".sticky", we can see that it doesn't just select

that article, but further down our list in the footer also uses that sticky class. That's why we need to be careful when choosing our selectors to make sure that it applies to element that we do want to target. So again, if I would want to target that article with, I could use ".content.sticky", I could even add ID, which is not needed in this case, because I'm already selecting this with, with these two classes.

But if I wasn't sure that maybe I, there are other elements with these two classes, which I do not want to target, then adding a "featured" ID will still select the same thing, but now I can be sure that there's no other elements because ID is a unique attribute, right? It should be only once on a page.

It may be a bit too complex for this simple example, but it's just so you know that you can combine multiple attributes together without any spaces here, it will apply to one element. Because when we are adding spaces inside our selectors, that's completely different. Let me show you an example.

Using space symbol in a selector

If I would want to select these h2s, the h2s within a standard article. The h2s here don't have any attributes. They don't have any CSS class or ID. And if I define just h2 that selects all these different elements, which I don't want, do not want to select so I can use the parent container CSS class to target elements inside that parent. And we are using space symbol to do that.

So the parent is article with a class content I could use just the class so ".content" and then space. So this selector means that we are selecting h2s, but only if the parent container has a class of content. That doesn't need to be the closest parent, any parent container with that class, all the h2s inside, will use our orange color.

And we can specify multiple parents, and it has to work along with the page

structure. So if I define ID main-content as the parent container, and then it has to have a content inside a class content and h2 inside. For example, if I move that article outside of my div ID main-content. This first h2 - this, title is no longer selected because that selector only selects all h2s within all containers with a class content, which are inside main-content container. Okay. So we can specify multiple parents, the full nested structure of our element, which is pretty useful to know of when working with Divi elements.

Selecting multiple targets with commas

Another useful character we are using inside our selectors is a comma. So adding commas inside our selectors allows us to add a second target to the selector. So for example, I want to target all the h2s with the, within these parents and also my, paragraph with a class post-excerpt.

So when using commas inside our selectors, we can specify multiple elements. They will use the same CSS properties, but we can target multiple, multiple targets and another, character I would like you to know of is the right caret.

Selecting direct descendants with rich caret symbol

So let's say, we would have...let me remove that and let's add h2 right here. Okay. So, sorry. Okay. Let's say I want to select that h2 and only that. It doesn't have any specific attribute, so I can look at the parent container. And I know it's main-content. So to target that h2, I could use a selector like this. Right. But it also targets all these other h2s, which are further down the page source.

And I don't want that. I only want to select this one. So there's actually a few different ways we could approach this but the character I want to show you is the right caret. And it means direct descendant. If we are using that, it, it doesn't need spaces here. So you would, you could say it like this, but, that's just easier to read.

So this means that the H2 is the child of main-content, but it's immediate child. If it's somewhere further down the page source, for example, this h2, right here, its direct parent is the article, not the div ID main-content. So with the right caret symbol, we can specify the direct parent and that would only select the immediate descendants. So that's the only h2 that's right here, immediately inside that div. Okay.

Selectors Overview

So a quick recap, we can select elements just by using their HTML tag name. We have the dot symbol for selecting classes, hashtags for IDs. We can combine all these together to select a specific element, a specific tag. We can use space to define the parent containers and we can use right caret to target only direct descendants inside the parent container.

And we can use commas to define multiple targets inside one selector. Now let's go ahead and look how we can target HTML generated by the Divi theme.

Selecting Divi theme elements

Here is a basic Divi layout and to be able to target different elements with CSS, you need to get familiar with the HTML generated by the Divi Builder.

Each section uses the et_pb_section CSS class, but that's not all. Many times you will see multiple CSS classes on a single Divi element. The full-width section has an additional CSS class, depending on some of the settings you set within the builder, you might see more additional CSS classes. There are also these numbered classes starting from zero.

The first section has a CSS class of et_pb_section_0. The second section has a class et_pb_section_1, and so on all the way to the bottom of the page. The raws and modules also use this numbered classes and the numbers for the modules do not reset if the module is a new section. So the Blurbs here, for

example, we see number zero one and two, and then further down, these would be number three and four.

Which CSS Class to target in Divi?

So how do we decide which CSS class to choose when defining our selectors? Choosing the generic, module class, for example, et_pb_blurb for the Blurb module is one of the options. But if I would target that class either as a parent or directly that container it will affect all the blurbs throughout the entire page.

So another option is to use that numbered class, which is unique for each module. And that is how the Builder does it. So for that module, I changed the icon color. So here this span which is inside that et_pb_module, et_pb_blurb, number two container uses that color. So that bit of CSS code is generated by the Divi builder for that, Blurb number two, the third Blurb on that page.

But as soon as I move that, in here. Okay. And refresh the page... now my icon uses a different selector. So Divi builder is smart enough to generate the CSS code that always will work. But if I would want to change something about that blurb with CSS and I would use that numbered class, and then I would move it in a different place.

It would use a different number. So it's not a best choice to use. Luckily, we can use the custom CSS option in the advanced tab of every element, every section or row, column inside the row. And then. In every module, we can add our own CSS class and CSS ID so if you want to change something specific about a single element, you can add your own CSS class. It will be added to the main container.

Let's see. Now that module has additional my-class CSS class, I can use multiple classes, assign multiple classes to a single element. Basically using your own CSS class is the safest way to add your custom code. Sometimes it's better to use ID, adding an ID, gives your selectors more specificity value, which we'll talk more

about soon, but just remember that you can only assign one CSS ID, multiple CSS classes, and you can target all the elements within, your modules using your custom attribute, customs class or ID as the parent container. And this way, for example, you could add more space below the header inside the Blurb module, that might not be inside the module settings. And the only way to target it is with custom CSS.

So using our own selector is the recommended approach. Thank you for watching. I hope it will give you some clarity on how to define your selectors in CSS and how to use it within Divi. And in the next lesson, we will look into using pseudo classes on these selectors.