CSS Selectors

[The text "CSS selectors" is shown.]

Instructor: CSS selectors and specificity. There are many different types of CSS selectors that allow you to target specific elements in HTML.

[Text saying "div { height: 400px; }" and ".logo-box { background-color: blue; }" is shown.]

So far we've learned how to target by using element names like p or H1. We've also seen how we can target class and ID names as well, using, for example, "#content" for an ID or ".icon" for a class. Then it's more specific, and instead of targeting every type of element, like all the divs or all the IMG or images. We can be more specific in which ones to target. This is called specificity. We saw a little bit of specificity when we talked about the order of CSS and how that matters. If the same exact selector had the same rule applied, the one later in the CSS file would be applied and overwrite the earlier one. Think of specificity as a score or rank that will determine which style declaration will be the one applied. For example, if I declare a font family for the whole page, each child element inherits the font throughout the page.

[An image of a website is shown. All the fonts are the same. An image shows some CSS text. The text selects the body element and sets the font family and font size of all the elements.]

But if I later declare my H1 to have a different font family later in the CSS file, it's more specific, and all my H1's ones will have a different font.

[The website image is shown again. The font of the H1 text is now different. The CSS image is shown with text selecting all H1 elements and setting their font family and font size.]

The body rule applies to all the text on the page, but the H1 rule is more specific. So it's the one that's going to be applied. So I might have all my images set to a width of 100%. But I find that some of my images are just too big.

[The website is shown. The instructor scrolls down the website. Some of the images are too big to fit both the height and width comfortably on the screen.]

So I need to have a smaller percentages for those. So I might target just those images that I need smaller with their class name, for example, ".heat" to make just those images have a more specific rule set.

[An image showing HTML is shown. The images that were too big for the screen now have a class name of "heat". CSS text selecting the heat class is shown. The CSS changes the image's width to forty percent. The website is shown again, this time the images are a comfortable size.]

Even though the width of 100% is still applied to all images, including them, the smaller width percentages will win out and will be applied because it's more specific. And here's where it gets interesting. Even if that rule came before the IMG rule, it would still apply because a class selector ranks higher and in specificity than an element name selector. Every selector will have a ranking and the specificity hierarchy, inline styles, as we saw earlier, have a very high ranking. But we're only going to be using an external CSS file. So for us, the next more specific one is the id selector, like "#logo-link". Then it would be classes and pseudo-classes like ".card-img" or ".card-img:hover", which we're going to talk about here in just a minute. Then lastly, the least specific are the element selectors with the element names like p, H1, IMG, et cetera. Keep this in mind when a rule seems to not work, you might have a more specific role targeting that same element and overwriting your rule. You may see some elements and classes used together in CSS. For example, the class dot heat might be placed with the IMG tag using both the element name and the class, like this.

[An image of CSS is shown. The selector is "img.heat".]

“img.heat” no spaces. This is just saying to target all the IMG elements that also have the class of heat. Or I might want all the a tags with the class of highlight.

[An image of CSS is shown. The selector is “a.highlight”.]

With “a.highlight”. This would come in handy if you use the class highlight with different types of elements and want to specify which element with that class name you want to target. We can also list a number of element names, classes, IDs, et cetera, that have the same declaration applying to all of them by separating them with commas. For example, “h1, h4” this would mean that all the H1s and all the H4s would then have that font family, the same one, and wouldn't need a separate rule for each one.

[An image showing CSS is shown. The selector is “h1, h4”. Text above the image says “h1 and h4”.]

We can get even more specific with descendant CSS selectors. For example, maybe we just want all the paragraphs inside of our article element to have a line-height.

[An image showing HTML is shown. Elements are inside an article tag. An image of CSS is shown. The selector is “article p” the text inside sets the line height to 1.5em.]

So we target just paragraphs that are children or descendants of an article element like this “article p”. The descendant would be the last element. So this means that all the p's that are inside of an article element would be affected. Or maybe we just want the images inside of our product gallery like this.

[The text “#product_gallery img” is shown.]

So just the images inside of “#product_gallery. We can also target images like we had before with the width of 40 percent, but without giving them all class names and just selecting them by either all the images in the main section or all the images inside the “img_container img”.

[The images that had the heat class name now do not have a set class name. Two CSS selectors are show, “main img” and “#img_container img”.]

Visit W3 schools for even more specific CSS selectors. Let's also talk about pseudo selectors and pseudo classes. One example of a pseudo-selector is the nth of type selector. If you have a number of the same type of element, it will let you select the first or the second, or the third or the fourth, et cetera, of those elements. For example, if I have a number of list items, I could select just the second one without having to give it a class or an ID by just using the end of type like this.

[An image of CSS is shown. The selector is “li:nth-of-type(2)”. Inside, a line changes the background color to a gray. An image of a list is shown. The second item in the list has a background color of gray.]

If I wanted the fourth one, I'd use a four instead of a two. Pseudo-classes allow you to change the appearance of an element when a user is interacting with it. For example, if I want the a links in my nav to have a different background color or the text a different color when the user hovers over them, I can use the colon hover after the selector and give it the declarations I want to happen when the user hovers over it.

[A CSS image is shown. Two selectors are shown. They are “nav a:hover” and “button:hover”. The website is shown next to the image. When the mouse hovers over a button that is also a link, the background color changes and the text changes color.

There are other pseudo-classes like active, focus, and visited. See W3 schools. For more examples.

[End of video.]