Pseudo-elements

Hello, and welcome to the lesson about CSS pseudo-elements. This is a very important lesson because understanding how the pseudo-elements work will open up so many possibilities for you when editing your websites. This is a CSS property which in short allows us to add new content to our HTML structure. There are quite a few different pseudo-elements in CSS, but I will only cover the ones which are fully supported by all browsers and that is "before", "after" and "selection".

Syntax

But before we start the syntax, should we use double or single colon? Technically, the correct way is to use double colon, single colon was a sort of CSS syntax mistake. And the intention is to separate pseudo-classes like "hover" or "visited" from pseudo-elements. The single colon works just as well and it is fully supported too. So I believe it is a matter of preference really and just sticking to one or another. Just don't do double colon in one place and then you single column the next time. Choose what works well for you and stick with it. There was a time when we had to support Internet Explorer 8, which only respected single colons, and I am kind of used to writing it this way. So in my custom code, you will most likely see single colon but it doesn't necessarily mean it is the best approach. Okay, so the first pseudo-element, the fun one, the "::selection" so you can apply it to any element and it will work within that or use it on its own. So it works for everything. But basically that's the syntax and it accepts only three properties which is: color, let's say white, background, let's say black, and also text shadow but let's not do that. Just to show you, that would be my selection. I can change the background to blue. Okay, I can use different selections on H1 and different for my P tag. Let's say I want H1s to be red and my P to be blue. It is designed to adjust the color for highlighted text, and it's just this nice touch and just shows that attention to detail when the user selects any text on our website and see our custom color or palette. And that's basically it.

Before and After Elements

And now the most important pseudo elements the "before", and "after" they allow you to insert content onto a page without it needing to be in the HTML and the end result appears on the page as if it is in your page source. So, what you need to understand is that the "before", and "after" are still inside the element, they are applied to. The naming sort of feels like these pseudo elements might come, you know, before or after but it's really before or after the other content inside that element. Okay? So just to show you, let's say I will apply a "before" and "after" to my H1. So for the before, and after pseudo-elements, we have to define the content property. And a value for the content can be a string - just some text, an image, so it would be content, URL image. Like that. But the image is inserted at its exact dimensions and cannot be resized. It's better to apply it as a background image. The content property can be empty, like this. And it's useful for inserting images as background images because you can set the width and height and can resize with background size. And you can basically do anything with that element. The content value can be a counter, I will show you that example in a moment and also an element attribute just to show you real quick. If I would have a link here. Okay, that would be my link. I could specify that

I want to display right after my A tag. I want to display that element URL. I could do that with attribute Href, you see you cannot insert any HTML inside for the content property. At least it won't be rendered as HTML. Most of the time I would just use the empty ones but let's go over a few examples you might find helpful to see how these pseudo elements can be used. I have a sample text module with a default looking block quote inside. And now, in the module design settings, I can actually change the appearance a little bit. And what I like to do is set the font to some serif font and font style to italic, and I would go ahead and remove that not beautiful border. And instead will add nice quotes with the "before" and "after" pseudo-element. So, to target block quotes, within a text module. I could use the standard et_pb text CSS class.

Quotes Example

If I want to target every text module but let's be a bit more specific and add our own CSS class "my quote". And I will add a custom CSS to target it. I can't do it here because it will apply to the whole module and I want to target the quotes within a module. So let's save that and let's actually go ahead and preview that in the browser inspector, okay? So, inside our block quote, we had a paragraph and it is inside the module with a class "my quote". So now we can add our CSS. Okay. So we have to target the, "my quote" class as a parent. And then block quote. And we will add the "before" element, pseudo-element. Okay, so our before tag needs a content. And inside the quote, I will paste my double quote symbol And by default this "before" and "after" elements are in line. But as you can see here, it kind of showed as it would be a block element. And that is because you see that the block quote HTML structure and we have our "before" and there is the P paragraph inside which is block by default. So we can add a new rule and also target "my quote" P within the block, quote. So not all Ps, but block quote P. We want to display in line. And that puts the "before" and P next to each other inlined. So it is aligned correctly. And now we can style the before, make it a little bit bigger, font size, 3em and let's change the italics to regular style. So font-style:normal, okay. And we can add the same thing to as an "after" element but with a closing quote, okay? So if we do apply this styles to both "before" and also "after" so "my quote" block quote "after" okay, that also adds this quote here but then we can add a separate rule to "my quote", block quote after and change the content to a closing quote- double quote. Okay, that's correct and now you can see that by increasing the size, we kind of broken the line height here. So let's change the line height to 0 that won't affect the rest of the elements here but it kind of moves that content in incorrect place. Now we can move it to the bottom with the translate property. It has to be displayed inline-block. So display inline-block and now transform and translate Y, which moves the element with the y axis vertically. We can try with 14. Okay, that looks great and now the color can be different. Okay, that's hard to see. We can choose something bluish - light blue. Okay, something like that. And it looks much nicer. Okay, so with a simple CSS, we've added three CSS rulesets. This would be "my quote" "before" and "after" styling.

Heading Example

And I can do a similar thing with headings, okay? So let's say I have a nice big heading here at the top. So let's add a new row, maybe, with a text module. That displays this huge heading at the top. Okay so let's say this is my head H1 and let's make this text centered. Let's say I want to add a decorative lines before and after the heading here, I could try creating divider modules and then use CSS to kind of align them next to each other. But using "before" and "after" should be easier, okay? So this is my class "centered-heading". Okay, I will save that, I will also change this module alignment to center, so it looks better, okay? And now let's save that and preview our page in the inspector again. So I have a text module with a class centered-heading. And inside there's my H1 so H1, which is within centered-heading class element. I want to add the before element and also let's do two at once and also after. Okay, and now I can add my declarations content empty. Okay, this time we are defining the element just by using CSS. So, we want to say display, inline block width let's say 80 pixels. Height 2 pixels, maybe. Background, let's do light blue. And we won't be able to move it with negative margin, but we can use translate and we can add margin to the sides so it's kind of separated from the text. So margin, top and bottom would be 0 and then 20 pixels on both sides and to move it to the top, transform Translate Y. And negative value. Let's try it with minus 1/2 em. That's too much. Something like that and maybe a bit less space here. Okay, that's a pretty nice looking heading, right?

So with a simple few lines of CSS gives us an option to add any type of content that could be a div with a specific background or, you know, an image or anything you like, you can add borders to it, use different shapes, icons, all that. So this is a very powerful tool to add additional content on your website. I hope it's helpful. I would love to see some examples of what you come up with using this "before", and "after" pseudo-elements