VIDEO TRANSCRIPT

CSS Shorthands

Hello, and welcome to the first lesson inside the advanced CSS techniques module. And one of the things that makes you a CSS pro is being able to write your code fast. And that is why in this video I would like to go through a few different CSS properties we can write quicker using CSS shorthands. Shorthand properties allow us to write multiple properties in a single line and in a compact way, which obviously, speeds up our workflow. But also by decreasing the CSS file size it speeds up our websites a little too.

Shorthand Principles and Margin Shorthand

First, let's start with a simple rule that applies to a few different properties. Like, padding and margin, you can specify margin or padding using four different properties margin-top. margin-right, margin-bottom, and margin-left or just by a single property "margin". So we could replace all this by specifying margin 20, pixels, 10 pixels, 30 pixels 10 pixels, and the order of the values matter, the values start at the top and move clockwise. So first value would be top margin. Next would be right then bottom and then left. And if only one value is defined, then it applies to all sides. So if I specify margin 20 The 20 margin will apply to each side. So the margin property can have one, two, three, or four values. If one it applies to all sides, second value, if we use 10 pixels here, then it still works clockwise you can think of it this way because the first value applies to top and bottom and the second to right and left. So it Still works clockwise because this is the first value, top. The second

value is the right. The bottom value. We don't have the third one. So it uses the corresponding opposite value. It uses the first one here. So it would be 20 again. And for the fourth one, we don't have it set. So it also uses corresponding opposite values. That's why the two values is top and bottom and then right and left. What about three values? If we say 30 pixels again, starting from the top 20 pixels here, 10 pixels here, 30 pixels at the bottom and then it doesn't have fourth value so it will use the value from the opposite side, which is 10. So, as you can see, these four roles 20 top, right 10, bottom 30 and left 10 can be specified using this exact shorthand.

Padding and border shorthand

The same rule applies to padding and also border, for example, I could specify border color red border-style:solid, And border-width can use like five pixels, but can also use 5 and then 20 and then 50 And 10 pixels, for example, right? And it also uses the same approach. If I remove these two, we would have five pixels border at the top next. 20 pixels from the right side. We don't have a specified value for the third bottom border width. So it uses the one from the opposite side, 5 pixels, and we don't have the fourth value specified so it uses the one from the right side. So, hopefully, that is helpful. Looking at this as the clock, it goes clockwise starting from the top. And even if there isn't enough values, it is still clockwise, but if it doesn't see the value, it will use the corresponding opposite value. So the next CSS short can't, we can use applies to borders as well. So, the border-width or border-radius can be specified using this method as well. Usually, we only specify one but if for some reason we want different values on different sides, you can use that shorthand as well, but border property also, let's say that would be five. Okay? And we could specify the same exact thing with a single line

border five pixel. So first is border-width then border-style:solid and Border color Let's say... green - no blue. Okay? This could also be more specific; We could only define border-left like that. Or we could be even more specific specifying border-left-width to 5 pixels or let's try 50. But this very specific values are not that common. You would probably only use it if you want to overwrite the main border style from one of the sides and the similar approach applies to outline so I could specify outline-width, let's say 10 pixels, outline-style, let's try with dotted, And outline-color, Oops, let's do blue. Okay? And again, the same exact values can be specified using an outline shorthand. If I remove that. It still applies, right? And there is additional outline property that doesn't go inside that shorthand. It's outline-offset. We can move the outline further away from the element. Okay, if we increase the margin here, let's try with... like that. And let's maybe do a solid one pixel outline. Okay, it's a nice effect you can use that adds a little border but not as normal border directly next to the container but it can be offset away from the element.

Font Property Shorthand

Next, font property. Okay, so maybe let's add some more text to it, lets may be removed this outline. And now for the font property, we can specify many different CSS properties. For styling, fonts: font-weight, let's say bold, font-style, italic. Font-variant, let's do a small caps, not all font families, you have that, but just to show you font size 20 pixels. Line-height, Two ems. Font family, let's say "Georgia", serif Okay, that changed our font styling a little bit and now we can include all these properties in a single font property. So that would be font and we specify the font-weight first bold, next style italic, small caps, 20 pixels/ line-height 2em. So font size/line-height and then font family in quotes.

VIDEO TRANSCRIPT

And then we can use coma to specify a fallback font. It could be a different one. Like Ariel, that's stupid to mix serif with sans serif. But just to show you let's say, serif and that's its font size and font family have to be specified. If not, this shorthand may not work in some browsers. So, just to show you, I can remove that. And it doesn't change the styling this, property actually works but I could only specify font size like 30 pixels and then font family serif. Okay. And that uses the default system serif font and font size. I could specify the line-height as well. I could maybe make it italic Let's try putting italic here. It doesn't work. So the order does matter. But you don't need to specify everything, just the size and font family is important here. Okay?

List and Background Shorthand

So another useful shorthand is for a list, let's add a ul here. Okay, and now, okay, let's put that away. Okay? So we can specify a list style. Sorry, how our ul. Can use list-style-type, and we have a few options... Let's go with the default circle or let's try square so we can see that. Okay. List-style-position It can be inside or outside. So, if I move it outside. We need more text here. Okay, that's better. And now outside or inside. You see, now you can see the difference and list-style-image. We can specify a URL and I have a sample image here. Okay, that's my image is probably too big to be using this way, but we can do that. And we can also list all these three values by specifying a single list style, Circle. Outside. For example, and for them, I could say no, but I don't need to specify that, okay? We can specify different background values, background-color. Background-image. Background-repeat. Background-position. It is position X and position y, so we could do 50 pixels. And 10 pixels, right? Or we could just say top Center. Or bottom center. Bottom right. Okay, or 10%. 10% from the top and from the left. Okay, so there are different

values we can use here but it's not the point of this video. So background-position and background-attachment And we can use fix that makes the image fixed. If we would use it as a background, and scroll, that would stay in the same place. It's the same. As the fixed Parallax background in Divi, when it's not using the true Parallax, it's just using background-attachment fixed. And all these can be specified using the following shorthand. So background, And we have first, we have color, let's change it to gray. Then we have image, we can specify the URL. Background-repeat We can leave it at the default. Repeat then position, let's say, top Center and size, and we need to use forward slash here. So let's say 30 pixels And we can use background-attachment, but we don't need to. Okay, so let's try changing it. Like we only remove the background-position. That doesn't work because specifying the size, is it necessary to specify the background-position as well. So, Okay, if I specify the position and say top left now, everything starts to work, okay? I don't need to do that. I can leave it at the default, but if I want to change the size, I have to specify the position as well.

Animation Shorthand

Now let me show you the animation shorthand. So there is a separate lesson on animations but To show you real quick, how the shorthand functionality works. I could have a CSS that specifies, that kind of animation. We have all these different animation properties, animation-duration let's say five seconds. the animation-name example we have to specify that animation in a moment. So one second, animation-delay 2 seconds. Animation-direction Alternate Animation-fill-mode normal. Animation-iteration-count infinite, for example, and animation-play-state Running. And animation-timing-function ease-out. And now we have to define our keyframes for this animation. So keyframes,

"Example", that's the name of the animation and it will start from and end. So from left is 0 to left 300 pixels. That's how we want to animate this. But as you remember the Left, Right Top and bottom property only works on positioned elements. So we have to define the position here. Okay. And that makes this animation work correctly, but we can change all these properties to use the shorthand to say animation, five seconds, example, 2 seconds, alternate, infinite, ease-out. So, the animation fill mode, and animation play state are at the default, so I don't have to define that here. And as you can see, we can write the same thing using a single line, okay? So let me remove all that.

CSS Transitions Shorthand

Another shorthand we can use is for CSS transitions. So the transition property is a shorthand for the following individual properties: We have transition property You can say, all and it will transition All properties. transition-duration, One second. Transition-timing-function ease-in-out And transition-delay We can use 50 milliseconds. And this is similar to the animation properties, the transition delay specifies when the transition effect begins, transition duration specifies the length of time the transition effect takes to complete and the transition property specifies the name of the CSS property. the transition effect is for. So we could say background color for example, okay? or all to animate, all properties. And now in order to see the transition, its best applied for the hover effect, for example, so if on hover, I will change that box background color, We can see that the background changes using these transition properties, and we can define it in a single line using transition-all, one second, ease in out and we can also add a delay.

And that sums up this overview using CSS shorthands can help you save time and space and make your code look cleaner. And while there are

rules to remember about the order, and a number of values you have to define. You can master them with a little practice, and I know that once you master them it will improve your workflow, and it will speed up your work.