VIDEO TRANSCRIPT

Animations

Hello and welcome to the lesson about CSS animations. Adding animation to your website is a very powerful way to draw users attention to important areas of your page because that is how our brains work, we have this natural reflex to notice movement. And when done well animations can enhance the visitor experience and add personality to your layouts. But as with anything else, the key is not to overdo it, you don't want to add animations just for the sake of adding it. And you certainly do not want to animate every part of the website - that just makes the site look chaotic. So similarly to the transform settings, there is nothing wrong with using animation from within the Divi Builder but it does animate the entire module. And sometimes we only only want to apply the animation to a part of the module or sometimes we might want to animate something when hovering over the parent container. CSS animations are made up of two basic building blocks, we have keyframes which define the stages and styles of the animation and animation properties which assign the keyframes to a specific CSS element and define how it is animated.

Properties

So let's look at each individually. Let's go ahead and define key frames for our first animation. The animation name. Okay. And then opening and closing brackets. Next, if it's a simple animation, we can use from and to value, with opening and closing brace, or we can use percentages. From would be zero and to would be 100%. And that allows

us to add additional stages for animation. Let's do a 60% for example, okay? And inside each of these opening and closing brackets, inside these brackets we define CSS properties for the animation. So let's say that at 0% I want my element to be super small, and with zero opacity. Okay, so transform scale, 0.1 and 0 opacity. And then at 60 percent we would make it visible opacity:1, and make it a bit larger so the transform scale 1.2 larger than its original size. And at the end of the animation, we would set the size to it's regular size, so scale one. We have our keyframes ready. And now using animation properties, we have to apply that animation to an element. So in our box here, 2 animation properties are required for the animation to work animation-name and animation-duration. And now if I refresh my page, Animation properties do two things: they assign keyframes to the elements that you want to animate, and then they also define how it is animated. So right now, we applied the keyframes to the box element and we defined that this animation, these keyframes should take place in duration of two seconds. So these two are required, but there are additional animation properties you might find useful to further customize and create complex animations. Next, animation-timing function. So similarly to the transition property we have ease, ease-in, ease-in-out or we can use a custom cubic bezier with the ease-in-out, it would slow at the beginning and then speed up and then be slow again. Let's leave the default "ease" maybe. Similarly to transitions, we can specify a delay, animation-delay. with milliseconds or seconds and also we have a property called animation iteration count. Which allows us to change the number of times, the animation will play and set the number ourselves like two times or use infinite for always playing animation. There is also an animation direction. Which can be normal, reverse, alternate or alternate reverse and reverse animation basically, place it backwards. On each cycle, the animation resets, and

the end state 100%, and then plays backwards to 0% and the default is normal. We also have animation-fill-mode. And this one can be a little bit confusing because all the CSS we specified, for inside our key frames it isn't applied to the element by default. So, by animation fill mode, we can either use the zero frame or the last frame to kind of apply the styles to an element. So, if at the last frame we would specify that the scale is double the size with the fill mode forwards, the element will stay at the say that size after the animation plays. Okay, and the backwards would work in a similar way, but apply the CSS from the first frame to an element before the animation. And normal is default.

Shorthand

And now we can use a shorthand to use all these properties in one. So at this exact order animation and we could use bounce in two seconds is 0 seconds 2 normal normal, but we can leave that out. Okay. So that should work the same exact way. Only the first two are necessary and important for the animation to work. One last property that doesn't go into the shorthand is animation play state, and this won't work with that example specifically. But if you have an animation and you want to pause on hover, you can specify animation-play-state: pause, and then animation-play-state: playing. But it's a very advanced use I would say. Okay, so that is it with those basic properties the possible animations you can create are endless, you have all the CSS properties you can think of to animate and define at which state of the animation what happens. So, it's very powerful and the best way to learn is to jump in and start animating.

Animista

Just one website I wanted to show you is animista.net and here you can

actually go through some sample animations and you have all these different values and you can play with the settings. And then what's great about this site is that you can actually preview the exact code. You can, you know, copy and paste and use for your own animations or just tweak it and basically, see how these animations are made out. Okay, so a very nice way to kind of learn CSS animations as well.