VIDEO TRANSCRIPT

Transitions and Trasforms

Hello and welcome to the lesson about transition and transform properties of CSS. The most important thing about using these effects, I want you to remember, is that when adding any kind of movement to your projects, you need to keep it simple, subtle and consistent. The movement you create, should convey meaning. It should always be enhancing, not distracting and annoying. So, what are transform and transitions? At the most basic level transforms move or change the appearance of an element, while transitions make the element smoothly and gradually change from one state to another.

Transitions

Let's start with CSS transitions. Without a transition an element being transformed would change abruptly from one state to another. By applying a transition, you can control the change, making it smooth and gradual. In this video, I'll be using transitions in conjunction with transforms. However, transitions can also be used elsewhere, where elements change from one style to another. For example, the most common use would be when a button changes color on hover. There are two properties that are required in order for the transition to take effect: Transition-property and transition-duration, and two optional properties are transition-timing-function and transition-delay. Each transition property can be defined individually but for cleaner and faster code, it's recommended that you use the transition shorthand. Here's the full shorthand sequence, the first two properties are required. So

the transition property specifies the CSS property where the transition will be applied. You may apply a transition to an individual property, for example just the background color or just the transform property or to all properties in the ruleset using all. The transition-duration property specifies the time span of the transition, you can specify it in seconds or milliseconds 0.3 seconds would be the same as 300 milliseconds. The transition-timing-function, this property allows you to define the speed of the transition over the duration, the default timing is ease which starts out slow quickly speeds up, and then slows down at the end. The other timing options are linear ease-in, ease-out and ease-in-out. You can also define an advanced timing function with a cubic-bezier and here's a great generator online you can use. It lets you compare your custom function to one of the default ones, save your functions and all that. So, very useful tool here. The last property is transition-delay, which as the name suggests - delays the transition by a certain amount of time. It can also be specified with seconds or milliseconds and can also use a negative value, which will start the transition immediately, but a kind of partway through the transition process.

Transforms

Now that we reviewed how to make a smooth and gradual transition let's look at CSS transforms. How to make an element change from one state to another with the CSS transform property. You can rotate, move, skew, and scale elements. These are the same exact properties we can find inside the Divi Builder. And transform can also use 3D transformation. But in this video will focus on the 2D. So we can use the Divi settings my main advice would be to be subtle and use it in small doses, and while it is perfectly fine to use the Builder settings, when we want to transform an entire module, I think it is also helpful to get

familiar with this properties and to be able to apply it to different parts of the page. For example, just the icon in the blurb module and for that we need to know how to define it in CSS. Usually, you would use a different transform properties on hover, to change the state of an element and add some interaction to a website. But sometimes you might want to use it to change the default state, for example, to move and align an element with the translate property. The translate value moves an element left, right and up and down, and the movement is based on the parameters, given for the X - the horizontal, and Y - the vertical axis, and the positive X value moves the element to the right, negative X moves to the left, positive Y value moves the element downwards and negative Y value upwards. And in this example, the element will move 20 pixels to the right and 10 pixels down. If we only want to move the element in one direction, instead of the main translate we can also use translate X or translate Y values that are very useful to use when we want to move some element by a few pixels for the perfect alignment or when we want to overlap some elements. Moving an element on hover by a few pixels up for example, also adds a nice touch. So very useful value to know of. And next scale value, the scale allows you to increase or decrease the size of an element. For example, the value 2 would transform the size to be 2 times, its original size. The value of 0.5 would transform the size to be half its original size. And you can scale an element by setting parameters for the width the X-axis or height Y-axis. For example, transform scale X 2 would change the width, make it twice as wide as the original. And just remember to keep things subtle, scaling an element to half its size or twice its size, it's probably not a great idea. And usually, I would just use a value like 1.1 or 1.2 to add that grow effect on hover. Next of - skew, with the skew value, the elements skews or tilts one direction or the other based on the values

given for the X and Y-axis and the positive X value tilts element left, negative right and and so on. And you can also use a shorthand to include both X and Y. I really don't use that skew property, very often. And the last value - rotate. With the rotate value, the element rotates clockwise or counterclockwise by a specified number of degrees. So positive values, such as 90° rotates, the element clockwise whilea negative rotates counterclockwise. Now, there is additional property the transform-origin is separate from the transform property but works together with it. It allows you to specify the location origin of the transform and by default the location at the origin is in the center of the element, but if you are using the transform rotate and you want to rotate not from the center, but from the top left corner, you would use the value 0% 0% or left top for the bottom corner, you'd use, 0 100 or right bottom etc. So make sure to add the transform origin property to the parent element not with the transform property in the hover selector. So, it needs to know from the start: What's the transform origin? And you can combine multiple transforms by using the transform shorthand and the transform shorthand allows you to string the various transfer methods into one property just by separating them with a space symbol. So you can also use the Divi Builder to kind of define your transformation and then inspect that element to copy that CSS property from the page source.

And I hope this explanation of the transition and transform properties will help you add some nice interactions to your layouts or better yet micro-interactions. Try being as subtle as possible. I think it is very important.