VIDEO TRANSCRIPT

Filters

CSS Filter vs Divi Settings

Hello and welcome to the lesson about the CSS filter property. We will go over the Divi filter settings and I will explain a few ways you can use custom CSS to achieve effects not possible using the default Divi settings. So, in the Divi Builder, we can play with the CSS filter property inside each element design settings, we have a filter section. And as you can see, we can rotate the hue, change the saturation, change the brightness, contrast, invert colors, apply sepia effect, I'm not sure if that's how you pronounce it, change the opacity and also blur the image. There is also blend modes here, but this is a different CSS property. So the problem with using the Divi Builder settings to achieve any of these modifications, is that it will apply to an entire element. So while it works well for images, it doesn't necessarily work well if we want to apply it as a background image and then apply our filters to the background. If filters are used on a section settings, it will change the color or blur the entire content of that section. All modules within that section will be affected, it is not a background setting. That is why it's sometimes useful to know the CSS property to be able to use filters in different ways. Especially with conjunction with "before" pseudo-elements, and hover styles. So just to show you if I change the filters, it changes the entire module. You see all the, all the entire section, all the modules will be blurred as well. So that's not necessarily what we would like to achieve.

The Code

VIDEO TRANSCRIPT

So let's look at the syntax. Okay, so this is my sample HTML structure. I have a single div with a class "box" and some content inside that div and I also have an image here which I want to show you at the end. So, first looking at the div, that's the CSS it is using. So my box div uses some padding, it has set width and height, the text styling, and it has position:relative. It's very important because I am using the "before" pseudo-element on that div. And to be able to display it, I have to define the content. It's empty, I'm using background image. And this before element is positioned:absolute. That's why I need the relative position because I want to position this background absolute relative to the parent container. That is why I also defined top right bottom and left values to 0. So my before element is exactly the same size as my main div box, okay? And negative Z-index, basically places that image behind the text. If I remove the Z index, then my before even - even if it's before, it's still on top because they are together in the same container. So we need that z-index to keep it behind the text.

Property Filter

Okay, so now we can look at the filter property. So the syntax is simple. The property name is "filter" and we can define the filter function and value. We can also use "none" to remove any filters applied and we can use multiple filters at once, just by separating each, with a space symbol. So first off the filter, I would use the most is - greyscale. And that's the syntax, we defined the function and then the value for that function. So, grayscale by default, no value, is 100 percent and it accepts values in percentages or decimal values. So, using 35% would work the same as using 0.35, okay? You can use either decimals or percentages. Value of zero - doesn't do anything to an image and 100% completely desaturates the element.

VIDEO TRANSCRIPT

Sepia Filter

Next filter is sepia, it works exactly the same, accepts values from 0 to 100 or from 0 to 1 and no value is 100 percent. Now, filter "invert" works in a similar way accepts values from 0 to 100%. And negative values are not allowed and the default is 100%, it takes the color and kind of goes from the original color to the color on the opposite side of the color wheel at 50% that makes every color gray. Because the middle of the color wheel is gray. And small values make the colors closer to grey and once it passes the gray in the middle, then it shows the opposite color. So that's kind of how it works. Not a very common thing to use, I would say.

Opacity Filter

A more useful function would be "opacity" and works the same from 0 to 100%. A value of zero is completely transparent and this function is similar to the more established opacity property. So, instead of filter opacity I could say, opacity 70%. That does the same thing. But the difference is that with filters some browsers provide hardware acceleration for better performance.

Saturate Filter

Now saturate, value of zero is completely unsaturated, value of 100 leaves the input unchanged, so this doesn't do anything. Values over 100 percent are allowed, providing supersaturated results. Something like that. And if the amount parameter is missing a value of 100% is used and negative values are not allowed.

Brightness and Contrast Filters

And next, we have two functions which accept the same values,

brightness and contrast. And the default is 100% leaving the element color unchanged; a value of 0 will create an image that is completely gray or black and values of an amount over 100 percent are allowed and will provide more contrast or more brightness. So to add brightness we have to go over one hundred percent and that same applies to contrast.

Hue Rotate Filter

Next filter, hue-rotate. And applies to a hue rotation, shift the color on the color wheel, the value of angle defines the number of degrees around the color circle. Each color on the element will be adjusted, so we can use degrees. That would kind of also go to the opposite side of the color wheel, or we can use turn. So 180 degrees, it's the same as 0.5 turn. The maximum value is 360° or one turn.

Blue Filter

The next filter - blur, applies a gaussian blur to the element and the value of radius defines how many pixels on the screen blend into each other. So a larger value will create a more blur. We can use pixels or different CSS units, but we can't use percentages here. So 50 is a lot of blur and five is just a mild blurry image. Now, notice that the blur blurs, the sides of the image. If you only want to blur the actual background element and keep the edges sharp, you have to use the overflow property. And make sure that any pixels, which are outside of the container aren't displayed basically. Overflow: hidden does that, and that basically sums up all the values that are available in Divi.

Using Divi to Generate Values

If you don't want to specify it yourself, you can always use Divi to kind of specify the end result you are trying to achieve. And then once we save

and exit, we can preview the page source and copy the style that is being applied to that element. So my image module, not the image itself, but a container div with a class of et_pb image. It uses filter hue-rotate, saturate, brightness. All the three filters. Rated with a space, we can use multiple ones so I can use Divi to kind of generate the desired effect and then use that property somewhere else in my CSS.

Drop Shadow

But there is one additional function which is not available in Divi as a setting, and that is filter drop- shadow. It accepts the same value as the Box shadow and let me bring up my image so that I can show you. I have just a sample icon here. And now if I say my image - first, let's make it a little bit smaller, so maybe 100 pixels. Okay. So this is my SVG image and I can apply filter drop-shadow. And for the amount we use the same value as the box shadow except for the inset option. We can't use that. But what is different than a standard box shadow? Let's maybe try with all of the box shadow property so you can see how that would work on an image. Let's say that first we have vertical offset, horizontal offset, blur radius, and the color. So we could use rgba for semi-transparent black. And that's the box shadow, we generated, but this SVG has a different shape and by using filter drop shadow, if I copy that code here and place it inside my let's remove that. You can see I have this white actually, let's make a smaller blur and darker shadow, okay? So this doesn't look great, but you can see how the shadow is being applied to the exact size of my icon, which is an awesome effect you might want to achieve with some SVG elements.

Hopefully, you find this summary useful. I often use filters to apply hover effects to part of modules, which need to be edited separately. The block module images, for example, in getting familiar with this CSS property is

important in our work.