

# Styling a Button

## Introduction

### Changing the appearance of link and button elements with CSS

### Changing the :hover state

### Changing the :active state

## Resources

## Action Items

## Introduction

Welcome to the video lesson on styling a button with CSS. This will be a very simple overview of some most common CSS properties you'll be working with.

What we refer to as a button in most cases in Divi is a link element styled to look like a button. But in HTML there is also a button tag.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Website</title>
5   <link rel="stylesheet" href="style.css">
6 </head>
7
8 <body>
9 <div id="main-content">
10
11   <a href="#">My Link</a>
12   <br>
13   <button>My Button</button>
14
15
16 </div>
17
18 </body>
19 </html>
20
21
22
23
```



```
<a href="#">My Link</a>
```

```
<br>
```

```
<button>My Button</button>
```

So in this lesson, I would like to style both of them, a link and a button, because as you can see, each of these HTML elements has different styling used by the browser by default.

As you know, a link is an inline element, and you won't be able to see it, but you can trust me that the button is treated as inline-block by default.

## Changing the appearance of link and button elements with CSS

Now let me add a few CSS properties inside the style sheet to change the appearance of these elements. We are going to target the "a" and "button" elements. Let's add some margin to it.

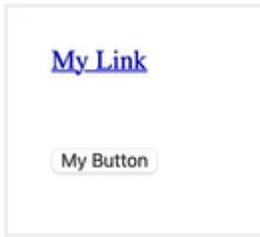
```
a, button {margin: 30px;}
```



And as you can see, the margin-top and bottom was not applied to My Link because, as you remember, inline elements will not respect top and bottom margin.

To fix that, we have to change the display tag to inline-block.

```
display: inline-block;
```



Now, let's change the background color. You can use the "background" property, which is a bit more complex and can include background URL or image. It's a shorthand, but we can also use only the background color inside that property.

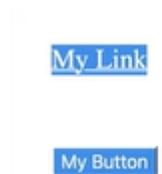
And we can actually define our color using HSL, so I want this background to be blue. I know that blue is around 200 in the hue, so let's try 210. And then for saturation, let's use a nice, vivid color, maybe 80%, and lightness close to 50%, but a bit lighter, so 60%. That gives us a nice blue color.

```
background: hsl (210, 80%, 60%);
```



Now let's change the text color, and to do that we will only specify the color property. Triple f is for white.

```
color: #fff;
```



OK, that starts to look nice, but we really need to add some padding. So let's try with "padding-top" 10px, "padding-bottom" 10px, "padding-right" - let's do more. I think it

always looks good if the buttons have more padding on the side, so let's try with 20px and the same for left.

```
padding-top: 10px;
```

```
padding-bottom: 10px;
```

```
padding-right: 20px;
```

```
padding-left: 20px;
```



And we can actually include all these values in the single property. We will talk more about it in the lesson of CSS Shorthands, but just to show you. The "padding" property would include 4 values or 2 values. If we use 2 values, the first value is for top and bottom, and the second value is for right and left. So by using that single property, we achieve the same effect as with what was here before.

```
padding: 10px 20px;
```

Now the button tag has a border added, so let's try changing it to our own border. Let's define "border-width" as 2px, "border-style" as solid, and "border-color". For color, we can actually use the one we have for background and just change it a little bit. So let's say I want my border a little bit darker. I can decrease the lightness, but I would also decrease the saturation a little bit, so maybe something like that, just a subtle darker border here. Now, a border can have a radius, so rounded corners. Let's try with 8px or maybe 10px. OK, it's not bad.

```
border-width: 2px;
```

```
border-style: solid;
```

```
border-color: hsl(120, 70%, 50%);
```

```
border-radius: 10px;
```



Now the font. So the links have a text decoration, underline, by default, and we can remove that.

```
text-decoration:none;
```

Now the font family. We can specify the exact font family name, for example, Montserrat, and then define the fallback dividing it with the comma, so the fallback would be any sans-serif font.

```
font-family:"Montserrat", sans-serif;
```

But I have Montserrat installed on my computer, so this does display Montserrat.



Now let's make it uppercase using text-transform, and remember from the spacing lesson that when we're using text transformation, it's good to add some letter spacing. Let's try .1em. That will make this size relative to font size.

```
text-transform: uppercase;
```

```
letter-spacing: .1em;
```



MY LINK



MY BUTTON

And we should also define the font size, so let's try with 14px, lovely. I would also use the bold font, so font-weight, you can use a value here. Light is 300, regular is 400, bold is 700, so let's just say bold.

```
font-size: 14px;
```

```
font-weight: bold;
```



MY LINK



MY BUTTON

Actually, I would add some padding here. Let's change our padding values to 20px and 30px. OK, that looks better.

```
padding: 20px 30px;
```

Now we can add text-shadow, and it uses a shorthand for horizontal shadow, vertical shadow, and blur-radius with its horizontal offset. So if we offset that by 1px or actually the negative one pixel and negative for vertical shadow and then zero blur-radius, so we don't need to specify that value and color would be the same as the border color, let's try with that.

```
text-shadow: -1px -1px hsl(120, 70%, 50%);
```

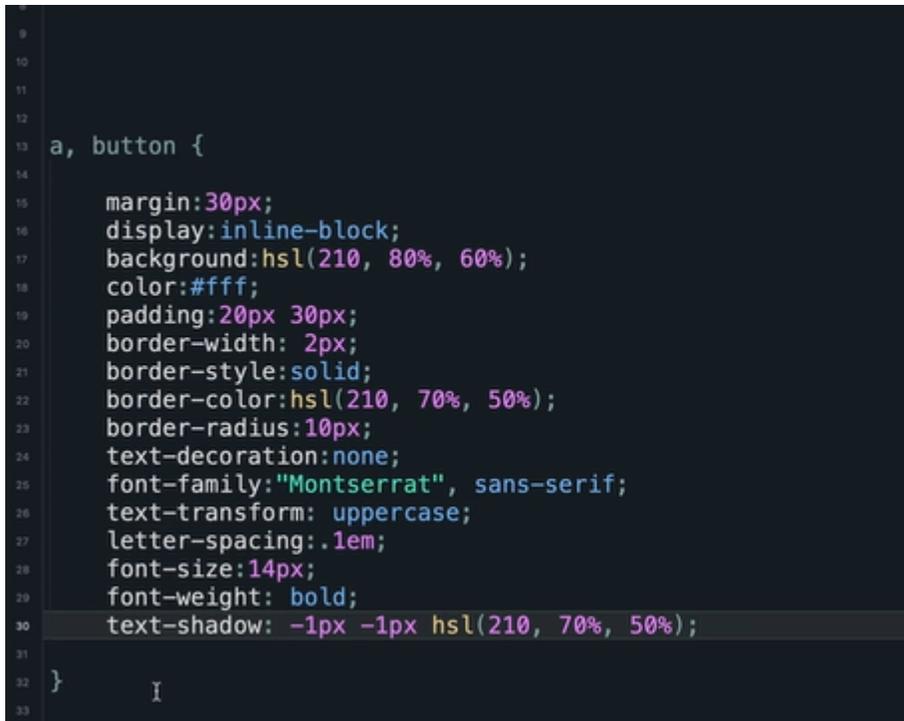
This text shadow kind of makes the text look like it's inserted inside. I will make it bigger, so -2px so you can see it better.



It makes the text look like it's inset inside. It works better with `-1px` but it may be harder for you to see. It's not something I do often, but I just wanted to show you all the different properties you may find helpful, and I'm sure you'll see a lot of these properties when you inspect your website source, so I just want to cover as much as possible.

```
a, button {  
margin: 30px;  
display: inline-block;  
background: hsl(210, 80%, 60%);  
color: #fff;  
padding: 20px 30px;  
border-width: 2px;  
border-style: solid;  
border-color: hsl(120, 70%, 50%);  
border-radius: 10px;  
text-decoration:none;  
font-family:"Montserrat", sans-serif;  
text-transform: uppercase;  
letter-spacing: .1em;
```

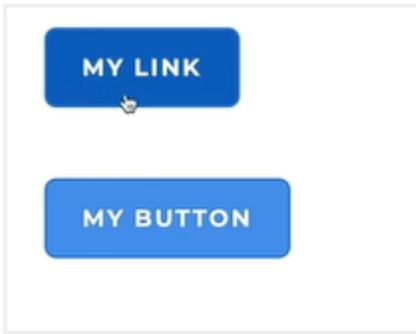
```
font-size: 14px;
font-weight: bold;
text-shadow: -1px -1px hsl(120, 70%, 50%);
}
```



## Changing the :hover state

Now this gives us a nice looking button, but let's try changing the hover state. And let's change the background. We can make it darker. So again, we will decrease the lightness and maybe increase the saturation a little bit. OK, it's nice.

```
a:hover, button:hover {
background: hsl(210, 90%, 40%);
}
```



But the change in color is instant, and if we want a smooth transition, we need to add a transition property to the “a, button” selector. We can specify which property we want to transition, so in our case: the background. Specify the duration, let's say .3 seconds, and the function, so ease-in-out, for example.

```
transition: background .3s ease-in-out;
```

So now let's match the border color with the background color.

```
border-color: hsl (210, 90%, 40%);
```

And the border color will not transition, so instead of “background”, we can say “all” so that every property is transitioned.

```
transition: all .3s ease-in-out;
```

So for the hover state, we have:

```
a:hover, button:hover {
```

```
background: hsl (210, 90%, 40%);
```

```
border-color: hsl (210, 90%, 40%);
```

```
}
```

And the transition property was added to the “a, button” selector.

## Changing the :active state

And now we can also style the active state, which is happening when we click. Instead of :hover, let's try :active and let's make it even darker.

```
a:active, button:active {  
background: hsl (210, 90%, 20%);  
border-color: hsl (210, 90%, 20%);  
}
```



So hopefully getting familiar with some of these properties will help you become more confident when writing your own code. I think the best way to know which property to use is just to inspect your website's CSS using the browser inspector to see which properties are affecting the website in which way and so on. I also included a little cheat sheet that will list all these properties for you. You can print this out and keep it close while you're learning.

## Resources

CSS Cheatsheet (available in the Downloads section of this lesson)

## Action Items

- Try creating a button from scratch using HTML and CSS. You can add it to Divi with the Code module. Share your design in the FB Group.