

Chapter 5

CSS Button Style

In this article, we're going through CSS Buttons. As we all know, buttons are important elements on pages and we use them to visually beautify common actions like **download**, **upload**, **log in/out**, **submit** and so on. A good button design enhances usability.

Before we explain how to style buttons, keep in mind that buttons are not links. The main purpose of a link is to navigate between pages and views, whereas buttons allow you to perform an action (such as submit).

Make sure to follow each step to get the intended results. In the end you will be able to create your own custom styled buttons.

5.1 Prerequisites

Before we start creating and styling buttons please consider downloading **Font Awesome** icon font pack. These icons will be attached to buttons later when we have created the basics.

After downloading, add the folders **css** and **fonts** inside your project folder like this:



Figure 5.1: Folder View after adding the Font Awesome Folders

Also, create your basic HTML file that looks like this:

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Buttons</title>
</head>
<body>

    <!-- STYLE SECTION -->
```

```
<style type="text/css">

</style>

<!-- HTML SECTION -->

</body>
</html>
```

5.2 Basic Styling

In this section, we're going to create a basic button using html and css.

5.2.1 Setting up the HTML

There are 3 main ways you can create a button starting from html.

1. Use the `a` (anchor) tag to create the link and give it a class, which by default if not styled as a button.
2. Use the `button` tag that html5 offers and you have a basic styled button with no css at all.
3. Use the `input` tag and give it a class of button and a type of submit. That will create a pre-styled button.

Your code with these three lines would be:

```
<!-- HTML SECTION -->

<a class="button" href="#">Anchor Button</a>
<button class="button">Button Tag</button>
<input class="button" type="submit">
```

This is what the basic buttons would look like with no css properties applied.



Figure 5.2: HTML - Basic Buttons

But for the sake of creating a button from scratch, we will only use the first method that is using an anchor tag, but add a `div` tag with a class of `btn-wrapper` because it will be useful as some properties cannot be applied over the anchor tag.

```
<a class="button" href="#">Button</a>
```

Until now, it looks like just a link, but we will change that with css.

5.2.2 Setting up the CSS

Lets give this button class some initial attributes to get the view of a button:

`border:0.1em #333336 solid` - gave the border respectively a width, a color and a style.

`border-radius:0.2em` - gave the border a radius of 0.2em (seems normal).

`text-decoration:none` - given this attribute, the text will not be underlined as the default browser link underline attribute.

`color:black` - this will give the text a black color, overriding the default blue color set by the browser

`padding:0.5em 1em` - gave the text inside the button a padding of 0.5em top and bottom and 1em left and right.

`background-color:#f3f3f3` - gave the button a light gray background color.

Lets also place it in a more pleasant position in order to have a better view.

We do that by giving the `btn-wrapper` class (from the div) margins.

The CSS code will look like this:

```
.btn-wrapper {  
    margin-top: 5em;  
    margin-left: 5em;  
}  
  
.button{  
    border: 0.1em #333336 solid;  
    border-radius: 0.2em;  
    text-decoration: none;  
    color: black;  
    padding: 0.5em 1em;  
    background-color: #f3f3f3;  
}
```

Given these attributes, we have created a basic styled button that looks like this in the browser:



Figure 5.3: Basic Styled Button

5.2.3 Button States

In addition to the default state, buttons can have two other states: `hover` and `active`, which respectively mean *mouse over* and *clicked/pressed*. Below we will show the button how to act/change when the cursor is over it and when it is pressed.

1. Hover State

The hover state can be achieved by adding `:hover` pseudo class like below:

```
.button:hover {  
    background-color: #cececc; /* added an intense gray color in active state */  
}
```

This would be the pressed button view:

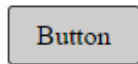


Figure 5.4: Hover State of Button

2. Active State

The active state can be achieved by adding `:active` pseudo class like below:

```
.button:active {  
    background-color: #a2b2bc; /* added a blue color in active state */  
}
```

This would be the active state (clicked) view of the button:

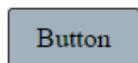


Figure 5.5: Active State of the Button

Note that when entering attributes about the other states rather than the default, you should only write attributes that are going to change when the button will be pressed, so it is not necessary to write again the padding or border-radius, these attributes will remain as the previous state.

As you can see, I gave it only the `background-color` attribute because that was what I needed to change, but you can also change the text color or border color when considering these states.

5.3 It's all about design!

From now on, we are going to use a custom font that is Lato and font icons we downloaded. Let's add the necessary links in the head section:

```
<!-- ADD CUSTOM FONT LATO -->  
<link href='http://fonts.googleapis.com/css?family=Lato&subset=latin,latin-ext' rel='<--  
    stylesheet' type='text/css'>  
  
<!-- ADD FONTAWESOME ICONS -->  
<link rel="stylesheet" href="css/font-awesome.min.css">
```

Also, add the css code to have one font for everything:

```
* {
  /* means apply to everything on the page */
  font-family: "Lato", "sans-serif";
}
/* added custom font lato in css */
```

5.3.1 Start Small

You can have a pretty nice button view with color alternation in aspects of background and text.

One first advice is, you don't always need border for the default state.

Have a look at the following attributes:

```
.button{
  border-radius: 0.5em;
  text-decoration: none;
  color: white;
  padding: 1em 3em;
  button */
  background-color: #329bd8;
  */
  text-transform: uppercase;
  font-weight: bold;
}

.button:hover {
  background-color: transparent;
  border: 0.15em #329bd8 solid;
  color: #329bd8;
  color */
}

.button:active {
  background-color: transparent;
  border: 0.15em #5e8ca5 solid;
  color: #5e8ca5;
  deeper blue */
}

/* increased border-radius */
/* changed text color to white */
/* increased padding for a larger button */
/* changed background color to a nice blue */
/* made the text uppercase */
/* gave the text a bold look */
/* changed the bg-color to transparent */
/* set a border to a blue color */
/* set a text color to the same as the background color */
/* minor text color change in a darker shade of blue */
```

This nice button would look this way in its 3 states:



Figure 5.6: Simple Button Design

5.3.2 Icons on Buttons

Going further into what's called a good user experience, this time with buttons, we will add an icon next to the text which will indicate what the button is for.

Icons are very easy to add, just find the one you want from [here](#) and copy the html code of the icon and paste it before the button text to make it sit right next to the text.

Look how I've added an upload icon to the button:

```
<!-- added <i class="fa fa-upload"></i> next to the button text-->
<<, <i class="fa fa-upload"></i>Upload>>
```

You will also need a few lines of css to make this look fine:

(only the commented lines are new or edited)

```
.button{
    border-radius: 0.5em;
    text-decoration: none;
    color: white;
    padding: 1em;                                /* removed the right and left ←
    padding value */
    padding-right: 3em;                          /* added only padding-right to ←
    align the icon left */
    background-color: #329bd8;
    text-transform: uppercase;
    font-weight: bold;
}

.fa-upload {
    padding-right: 2em;                          /* added padding-right to space it ←
    from the text */
    font-size: 1.2em;                          /* increased font-size to fit the ←
    button */
}
```

The button view in the browser:



Figure 5.7: Button with Icon Attached

Icons can be a good point to consider for buttons. These are just some other buttons with icons:



Figure 5.8: buttons10

You can change icon fonts attributes just like you do with text.

5.3.3 Gradients on Buttons

Just like you apply background colors to buttons, you can also apply *gradient*'s on them.

Gradients make buttons look more like 3D rather than flat like we've seen until now.

Creating gradients yourselves is just time-wasting, so I suggest you generate gradients online in [this website](#)

Feel free to choose any gradient you like and copy the css. To be coherent with what we've started Imma choose a blue gradient.

Look at the code below:

```
/* just removed background-color attribute and added the custom css code */

.button{
  border-radius: 0.5em;
  text-decoration: none;
  color: white;
  padding: 1em;
  padding-right: 3em;
  text-transform: uppercase;
  font-weight: bold;
  text-shadow: -1px -1px 0 rgba(0,0,0,0.3);
  color: #FFFFFF;
  background-color: #49c0f0; background-image:
  -webkit-gradient(linear, left top, left bottom, from(#49c0f0), to(#2CAFE3));
  background-image: -webkit-linear-gradient(top, #49c0f0, #2CAFE3);
  background-image: -moz-linear-gradient(top, #49c0f0, #2CAFE3);
  background-image: -ms-linear-gradient(top, #49c0f0, #2CAFE3);
  background-image: -o-linear-gradient(top, #49c0f0, #2CAFE3);
  background-image: linear-gradient(to bottom, #49c0f0, #2CAFE3);
  filter:progid:DXImageTransform.Microsoft.gradient
    (GradientType=0,startColorstr=#49c0f0, endColorstr=#2CAFE3);
}

.button:hover {
  background-color: #1ab0ec; background-image:
  -webkit-gradient(linear, left top, left bottom, from(#1ab0ec), to(#1a92c2));
```

```
background-image: -webkit-linear-gradient(top, #1ab0ec, #1a92c2);
background-image: -moz-linear-gradient(top, #1ab0ec, #1a92c2);
background-image: -ms-linear-gradient(top, #1ab0ec, #1a92c2);
background-image: -o-linear-gradient(top, #1ab0ec, #1a92c2);
background-image: linear-gradient(to bottom, #1ab0ec, #1a92c2);
filter: progid:DXImageTransform.Microsoft.gradient
        (GradientType=0,startColorstr=#1ab0ec, endColorstr=#1a92c2);
}
```

In the code above I did only consider the normal and hover state of the button.

It's up to you to add the active state if you think you need it.

This is the how the button we'd get:



Figure 5.9: Gradient Applied on Button

That's it on gradients, try more to see how you can get creative on this.

These are just some of them I made for you.

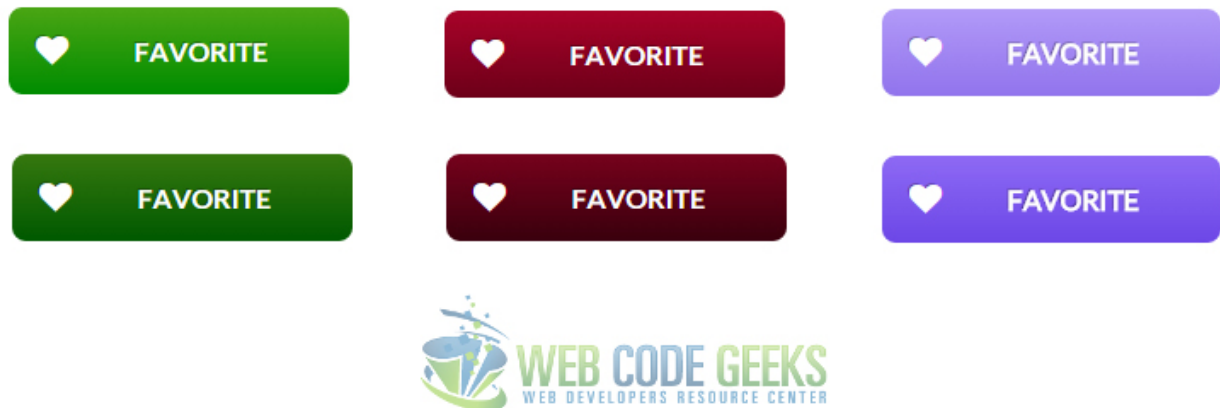


Figure 5.10: Gradients Button Design

5.3.4 Patterns on Buttons

In addition to gradient backgrounds, buttons can have **pattern** backgrounds.

You can find a great gallery of css patterns [here](#), feel free to use any of them.

The way we use them is just like gradients, remove any background color we have and paste the css code for the pattern.

Look at the code below:

```
.button{
  border-radius: 0.5em;
  text-decoration: none;
  color: white;
  padding: 1em;
  padding-right: 3em;
  text-transform: uppercase;
  font-weight: bold;
  background-color: #6d695c;
  background-image:
    repeating-linear-gradient(120deg, rgba(255,255,255,.1), rgba(255,255,255,.1) 1px, ↵
      transparent 1px, transparent 60px),
    repeating-linear-gradient(60deg, rgba(255,255,255,.1), rgba(255,255,255,.1) 1px, ↵
      transparent 1px, transparent 60px),
    linear-gradient(60deg, rgba(0,0,0,.1) 25%, transparent 25%, transparent 75%, rgba ↵
      (0,0,0,.1) 75%, rgba(0,0,0,.1)),
    linear-gradient(120deg, rgba(0,0,0,.1) 25%, transparent 25%, transparent 75%, rgba ↵
      (0,0,0,.1) 75%, rgba(0,0,0,.1));
  background-size: 70px 120px;
}

.button:hover {
  background:
    radial-gradient(black 15%, transparent 16%) 0 0,
    radial-gradient(black 15%, transparent 16%) 8px 8px,
    radial-gradient(rgba(255,255,255,.1) 15%, transparent 20%) 0 1px,
    radial-gradient(rgba(255,255,255,.1) 15%, transparent 20%) 8px 9px;
  background-color:#282828;
  background-size:16px 16px;
}
```

In this code I just added pattern backgrounds for the normal and hover state of the button.

This is how the pattern styled button would look like:



Figure 5.11: Gradients on Buttons

You can play around with these and see a lot of good looking buttons you can create. Here are a few I created for you:



Figure 5.12: Getting Creative with Patterns on Buttons

5.4 Conclusion

To conclude, we can state that there are various ways you can style and fit buttons according to your needs.

You can add png images instead of icons or images as backgrounds into a button, but after some time practicing you will understand that keeping it simple and nice is a combinations you can achieve by trying either flat or gradient design, which will also make your project less cluttered from images on each button.

Below you can find some pre-styled and good designed buttons that you can use for your own projects.

5.5 Download

Download You can download the full source code of this example here: [CSS Buttons](#)