

Chapter 24 – Modifying Bootstrap

Remember what Sam's Sarnies looked like earlier? Much better now, don't you think? There are plenty things left to do, but I hope you are beginning to get a feel as to what is possible by using Bootstrap.

But maybe you don't like how Bootstrap makes some things look? Also, imagine if all websites in the world used Bootstrap out-of-the-box without tweaking anything. The web would look a little tedious.

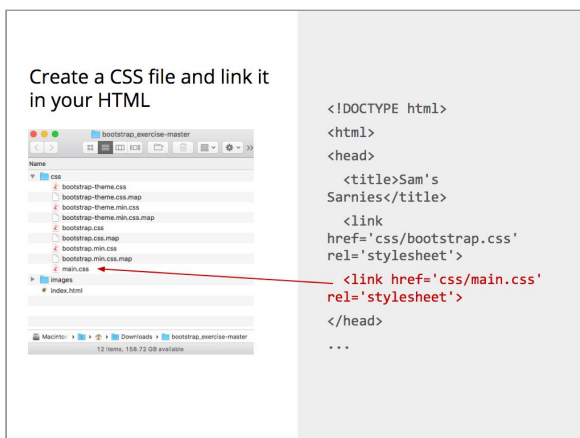
Well, here is the good news. We can override the styling Bootstrap gives us with our own styles. But a word of warning first.

Bootstrap has been designed and heavily tested for good cross-browser compatibility. Unless you know what you are doing, or have a lot of time, it's probably best to stick with the Bootstrap layout and keep your overrides to fonts, colours and general small things that leave the layout structure well alone.

The correct way of overriding Bootstrap styles is by creating a new CSS file of your own and linking to it in your HTML. Never ever make changes in the Bootstrap files themselves.

- Always link your own CSS file after the link to the Bootstrap file
- Use the Bootstrap CSS classes to style elements as you wish

TASK: Create a CSS file for overrides and link it in your HTML



Let's create some CSS overrides, but first we need to create a CSS file.

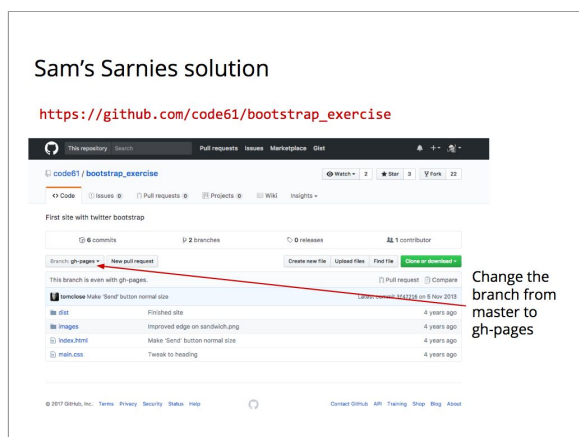
1. Go to your project in Atom and inside the CSS folder create a file called `main.css`
2. In `index.html` link to your new `main.css` file. Make sure the link sits after the link to the Bootstrap CSS file

Maybe you are wondering why the link to your own CSS with overrides needs to come after the Bootstrap CSS. Maybe you remember from the previous class that CSS stands for Cascading Style Sheets. It means that a browser will apply the CSS rules in the order it comes across them. If you add your Bootstrap file first, then a browser will apply the rules in there. If Bootstrap is followed by another file, then a browser will re-apply any rules it finds in there. Because we want to override Bootstrap we need to add our styles last.

Sam's Sarnies solution

So far we have improved Sam's Sarnies web page by adding Bootstrap classes to our HTML. But you can still do a lot more if you wish. If you look in your course notes on page 48 is a link to what Sam's Sarnies should look like. For homework go ahead and finish of the exercise.

If you get stuck you can go back to the GitHub repository for Sam's Sarnies. Change the branch name from master to `gh-pages` and you can see the solution.



NOTE: The following sections (until the JavaScript & jQuery section) are additional tasks you can do if you have extra time.

Even more awesome stuff

The next step for Sam's Sarnies web page is to start making use of that custom CSS file we created and begin to override some of the Bootstrap styling.

Try to do these tasks yourself, without the help of your instructor. Work with a partner or in small groups. If you get stuck your instructor can always help!

TASK: Change the buttons to social buttons

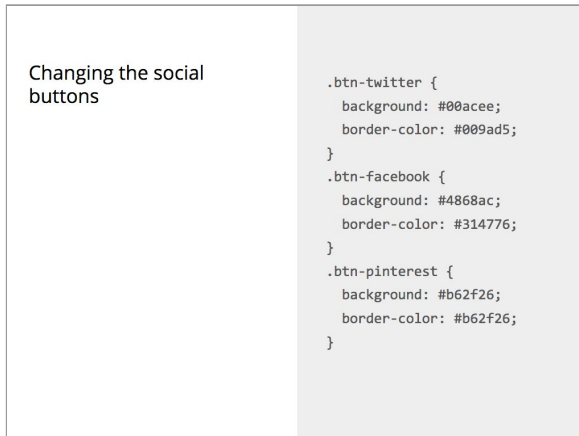
Changing the social buttons

We have classes in our HTML we can use in our CSS to add custom styles

```
<div id="mission">
...
<div id="social-buttons">
  <button class="btn btn-lg btn-twitter">Twitter</button>
  <button class="btn btn-lg btn-facebook">Facebook</button>
  <button class="btn btn-lg btn-pinterest">Pinterest</button>
</div>
</div>
```

Now that we have our own CSS file we can finally start to write some CSS. We'll begin with adding custom colours to the social media buttons at the bottom of our page.

Look at the code for the buttons in index.html. Each button has three different classes. Two are Bootstrap classes: **.btn** and **.btn-lg**. They are responsible for making the buttons bigger and adding rounded corners. For our purposes we are interested in the third class: **btn-twitter**, **btn-facebook** and **btn-pinterest**



1. In your CSS create three rule-sets for each button and add the correct background colour and border colour for each button. They are different for each button.
2. Save the file and look at the effect in Chrome. You can see the buttons are now coloured in the social media brand colours.

But the text is still black and we want it to be white. All three buttons have two other classes in common, we could either of those to fix our problem, but if we do, then any other button using those classes will change colour too.

We need something more specific.



The buttons sit inside a div with an ID of **social-buttons**. An ID is unique on the page, there is no other element using that ID. This is perfect for our needs.

In your CSS add a rule to colour the text of the buttons inside **#social-buttons** white

TASK: Changing the jumbotron background



Next up we are going to get rid of that sad grey background behind the jumbotron and replace it with the fruit and vegetable picture in our images folder.

Add the CSS for the background image to your main.css file. There are a few things to take note of here:

- The file path of the image is again a document-relative link. You are linking to the image from your CSS file. The CSS file sits inside a folder called **/css** and the image sits inside a folder called **/images**. The file path is instructing the browser to go up one folder in which the CSS file is, and then find another folder called **/images** and in there locate **fruit-and-veg.jpg**
- The filename of the image is case sensitive and includes the file extension. A typo or the wrong file extension will result in no image being shown by the browser. Browsers are picky things. If something is not exactly right, it will skip over it and ignore it.
- Background-size and attachment will make sure that the image covers the entire container inside which it is displayed.

Save the file and have a look in Chrome. It will look good, but not yet good enough. We want the background image to stretch from edge to edge of the browser window. If you remember earlier we added a **.container <div>** around all the content of our page. The **.container <div>** centers all the content inside the viewport.

TASK: Adding .container divs inside the three main page sections

<p>Moving the .container divs</p>	<pre>... <body> <div class="jumbotron"> <div class="container"> </div> </div> <div id="buzz"> <div class="container"> </div> </div> <div id="mission"> <div class="container"> </div> </div> </body> </html></pre>
-----------------------------------	--

To fix our problem we will need to move `.jumbotron` to be outside `.container`. This will make the background image bleed from edge to edge. But that's not enough. We also need to create a new `.container <div>` inside `.jumbotron` so that the content in there is centered inside the viewport.

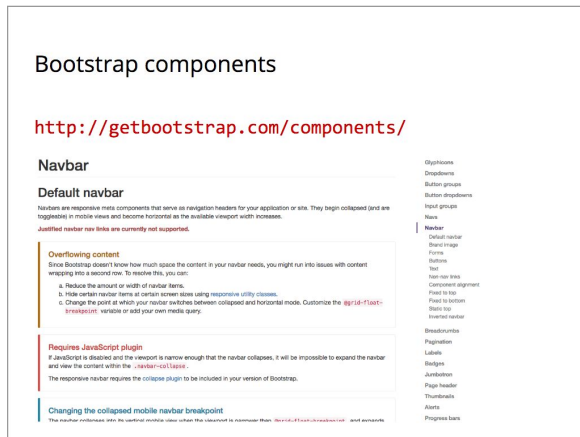
Go ahead and update your HTML. You should do the same for the `#buzz` and `#mission` divs.

TASK: Change the background colour of the mission section

<p>Changing the background colour of #mission</p> <pre>#mission { background: rgba(32, 35, 41, 0.9); color: #ddd; }</pre>

Now each section is separate we can add a full width background colour to the `#mission` section.

In your CSS add a rule-set to change the background colour of the mission section and the font-colour as well.



Navbar

We are nearly there with Sam's Sarnies. One last thing we need to do is add a navigation section at the top. Go to the Bootstrap documentation page and at the top of the page select the Components page: <http://getbootstrap.com/components/> In the sidebar menu on the right click on Navbar.

TASK: Add a navbar to your page

Scroll down to the section called Inverted navbar. That's what we are going to create.

1. Right click on the black nav bar and choose Inspect to bring up the Web Inspector.
2. Highlight the line of HTML beginning with `<nav class="navbar navbar-inverse">`
3. Right click again, with the HTML selected, in the pop-up menu select **Copy** and then **Copy Outer HTML**
4. Go to your index file and paste the code at the top of your HTML `<body>` section, above the `.jumbotron <div>`.
5. It may have copied it all in one block. You can place all code on a separate line to help you understand it.

We also want the navigation to stay at the top of our page. Scroll up in the Bootstrap documentation to the section titled Fixed to top.

To fix a navigation bar at the top of the page we need to add a class to our navigation bar and we need to create a container div inside the navigation

1. Add the `navbar-fixed-top` class to your HTML nav element
2. The code we copied already contains a `.container <div>` inside the `<nav>` element. It has a class of `.container-fluid`, which makes it sit at the far left of our viewport. Change `.container-fluid` into `.container`

A fixed to top navigation also needs some additional CSS in your main.css file. Add the extra CSS at the top of your main.css file. This CSS will add extra padding to the top of the `<body>` element so that the content of the `.jumbotron` sits below the nav bar.

Adding a search box

We want to add a search bar inside our navigation block. Scroll further up the Navbar section of the Bootstrap documentation until you get to a part titled Forms.

1. Copy the code for the form (no need to go via web inspector)
2. Place it inside the **.collapse.navbar-collapse <div>**. That's just above the first of two closing divs at the bottom of the **.navbar <div>**
3. If you save the file and look in Chrome you will see that the search box sits on the left next to the navigation links. We want it to be on the far right.
4. In the Bootstrap documentation go to the CSS page and in the menu on the right hand side click Helper classes and then Quick floats
5. To align something to the right you can use a **.pull-right** class
6. Add the **pull-right** class to your search form element
7. Save the file and look in Chrome to see the effect.

Homework for week 5

Task:

Finish off any unfinished bootstrap_exercise from class
Course competition website

Task:

Continue work on your personal site. Use Bootstrap to make it **responsive!**

Preparation for next week

Task:

[Try jQuery on Code School.](#)