

11.6.3 Customize the Page with CSS

Dana wants to customize her page by adding specific CSS components to the stylesheet, such as applying a background color to the page and adding an image to the jumbotron. Although it's neat and tidy in its current form, it's not very attention-grabbing.

These last few steps are all she needs to wrap up her research and publish it, so she's looking forward to adding that last bit of professional polish.

Now it's time to spruce the page up a bit. Dana wants to make use of some of the built-in styling that Bootstrap offers, but she will also need to code some of the customizations by altering the CSS associated with her webpage. The CSS customizations will take place in the style.css file we created earlier.

CSS syntax, as with all coding languages, is very exact. Each customization entry follows a pattern. First, the area of customization is set, usually identified by its element, class attribute, or id attribute. A colon follows, then the value of the customization, and a semicolon completes the line of code.

It's very similar to KVPs in an object; for example, a line that reads `"background-color: #292b2c;"` `background-color` is the key, and the number following it is the value. This particular combination will generate a dark background with light colored text. Let's use this syntax to apply a light font color to the webpage.

In our style.css sheet, enter the following lines of code:

```
body {  
  
  color: #f7f7f7;  
}
```

This block of code provides our webpage with specific styling instructions: We want a light text (`color: #f7f7f7;`) applied to the body (`body {}`) of the page.

We chose to apply the styling to the body of the page because it is one of the outermost elements—every other visual element lies within the `<body>` of the page. If we were to replace "`body`" with "`h1`," then only the `h1` elements would be affected. Similarly, because we want to apply the same styling to the entire page, we can add a "bg-dark" class to our body element. In our `index.html` file, update the opening `<body>` tag so it looks like the line below:

```
<body class="bg-dark">
```

This will apply a dark background to the entire page. We'll be adding this class to most of our elements so they all have the same styling. This applies a cleaner, more cohesive look to the webpage.

Update the Navbar

Let's also make use of Bootstrap's built-in styling, starting with the navbar. In the `index.html` file, let's add `navbar-dark bg-dark` to the `<navbar>` classes, like so:

```
<nav class="navbar navbar-dark bg-dark navbar-expand-lg">
```

This will apply a dark background and light lettering to the navbar for us, without any extra styling in the CSS stylesheet.

Customize the Jumbotron

Next, update the jumbotron to something more appealing by changing the font and background. This time, we'll add a Bootstrap class called `"display-4"` to the `<h1>` element inside the jumbotron, which will change the font style and size for us.

```
<div class="jumbotron">
  <h1 class="display-4">The Truth is Out There</h1>
</div>
```

Now let's make the background more interesting. Dana has chosen a space image to use as a background instead of a solid color; download it and save it to the "images" folder you created in your repo.

[Download nasa.jpg](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_11/nasa.jpg) [\(https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_11/nasa.jpg\)](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_11/nasa.jpg)

To assign this as our new background, we'll need to use our style.css sheet. Open that file in your code editor. What we'll do next is create an entry that will style only the jumbotron element.

Let's take a look at the CSS code that will style the jumbotron.

```
.jumbotron {
  background-image: url("../images/nasa.jpg");
  background-size: 100% 100%;
  text-align: center;
}
```

This code accomplishes a few things:

1. `.jumbotron` tells the stylesheet to apply the changes only to divs with a class of `"jumbotron."`
2. We specify a background image, then tell the code where the image is stored.
3. The size of the background is set to span the entire width and height of the element, no matter the size of the viewport.
4. We've aligned the text to be in the center of the element.

Customize the Form

The form element in the HTML will also need a small update: The text is now white, but the form's background is also white by default. Identifying what the form is telling us is difficult in its current state, so let's update the font color.

First, let's look at the form element in our HTML to identify which one we'll apply the styling to.

```
<form>
  <p>Filter Search</p>
  <ul>
    <li>
      <label for="date">Enter Date</label>
      <input type="text" placeholder="1/10/2010" id="datetime" />
    </li>
    <li>
      <button id="filter-btn" type="button" class="btn btn-default">
        Filter Table
      </button>
    </li>
  </ul>
</form>
```

We want this element to match the dark theme we've applied to the rest of the page, so we'll add the class "bg-dark" to our form element.

Let's also style the list element and the button to match the rest of the theme. We'll add a couple of different classes here to take advantage of Bootstrap's built-in styling.

SKILL DRILL

Add the class to the indicated elements:

"bg-dark" to the opening `<form>`, ``, and each `` tag.

`]{style="font-family: inherit; font-size: 1rem;"}Update the button element to use a class of "btn-dark" instead of "btn-default".`

By adding these specific classes, we have incorporated the same style and colors we've already selected (bg-dark) and also cleaned up our list elements. Instead of a bullet appearing for each list item, we now have evenly spaced rows. Additionally, the same font has been applied to the

entire form. Changing the button class to "btn-dark" also matches the button's appearance to the rest of the page.

The final result is a dark page with white font – a clean, visually appealing page where each element meshes well with the next.

