Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a peer programming activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are both responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

You will use the same project from the previous Hacktivity. If you need to, you can re-download it from GitHub using the URL, https://github.com/cbourke/cssProject.

1 Using Other Styles

Good graphic design is difficult. Using frameworks and libraries such as Twitter’s Bootstrap or jQuery UI can help us by providing a basic, modern style that we can use and modify to suit our needs.

In this exercise, you’ll experiment with using Bootstrap (http://getbootstrap.com/) to build a more visually appealing web page. In particular, you will attempt to recreate the page in Figure 1. The content of the page is unimportant. In fact, placeholder “lorem ipsum” text is usually use when designing the layout of a page. You can generate such text using the following resource: http://www.lipsum.com/.
Hello, world!


Learn more

New! We have new material for you to review, see here!

Brilliant!

Responsive!

Sleek!

Figure 1: A nice page built with Bootstrap
Bootstrap implements what is known as a **responsive** design. Web pages will be viewed on a variety of devices with a variety of screen sizes from large desktops to small smart phones. A responsive design resizes and rearranges the content of a page to automatically adjust to each screen size to provide a good user interface (UI) and user experience (UX) for users on all devices. This ensures interoperability without having to build a separate version for each potential device.

To make sure your page looks right, make sure you’re displaying it on a large enough screen.

1. Read the *Getting Started: Basic template* ([http://getbootstrap.com/getting-started/#template](http://getbootstrap.com/getting-started/#template)) section of the Bootstrap and create a basic shell to get started.

2. To avoid downloading bootstrap, for now you can use the CDN as described here: [http://getbootstrap.com/getting-started/#download](http://getbootstrap.com/getting-started/#download).

3. To get started, create a *container* as described here: [http://getbootstrap.com/css/#overview-container](http://getbootstrap.com/css/#overview-container).

4. Add a “jumbotron” to provide a nice header ([http://getbootstrap.com/components/#jumbotron](http://getbootstrap.com/components/#jumbotron))

5. Read the documentation on the Grid system ([http://getbootstrap.com/css/#grid](http://getbootstrap.com/css/#grid)) and create rows/columns to match the layout in the page in the figure.

6. Add a “responsive” image ([http://getbootstrap.com/css/#images](http://getbootstrap.com/css/#images)) so that the size of the image adjusts to the rest of the layout as a the screen is resized.


8. Add a table within a panel. Use some of the features described in the table documentation ([http://getbootstrap.com/css/#tables](http://getbootstrap.com/css/#tables)) including striped, bordered, and hovered tables. Place your table within a *panel* ([http://getbootstrap.com/components/#panels](http://getbootstrap.com/components/#panels)) to offset it and give it an outer border.

9. See your page in action as you resize it. Deploy your page to the CSE server. If you have a tablet or a phone, view it to see how the page’s responsive design works.