Media Queries, Offsetting, Forms, Jumbotrons and Navbars

In this exercise, you will create a series of pages with guidance and there will be a small section where you complete a short activity on your own. Please complete both the guided and on your own section as you will be graded on both. Have fun!

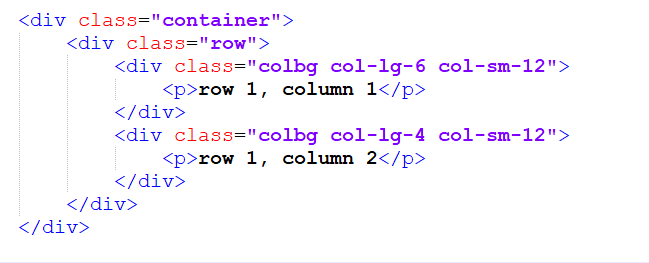
# Offsets and Media Queries

1. Please start by downloading the start files. They come in a zipped folder with several files and a CSS folder/file.

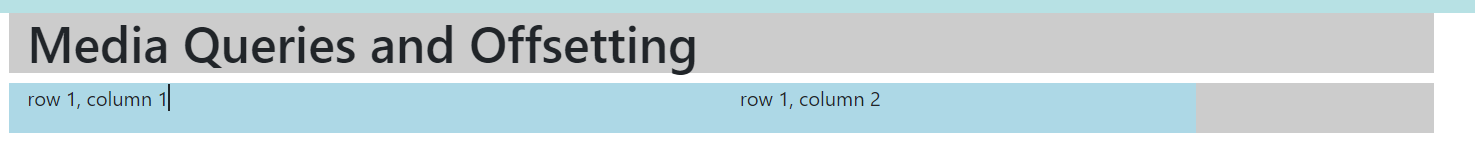
2. You will begin working with the mediaQueryOffset.html file. Edit the file in your favorite text editor.

You will notice there is a row with two columns. Each column has some classes already applied. See code below:

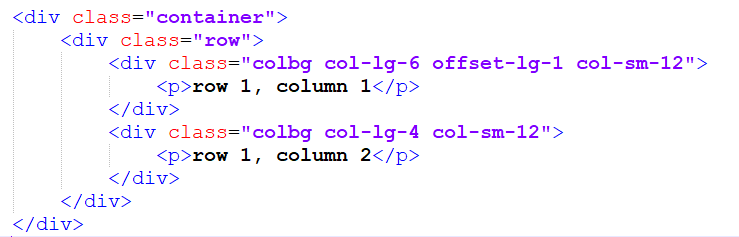
* Each column has a class of colbg which is styled in the external stylesheet
* Additionally, the first column has classes col-lg-6 and col-sm-12
* The second row has classes col-lg-4 and col-sm-12



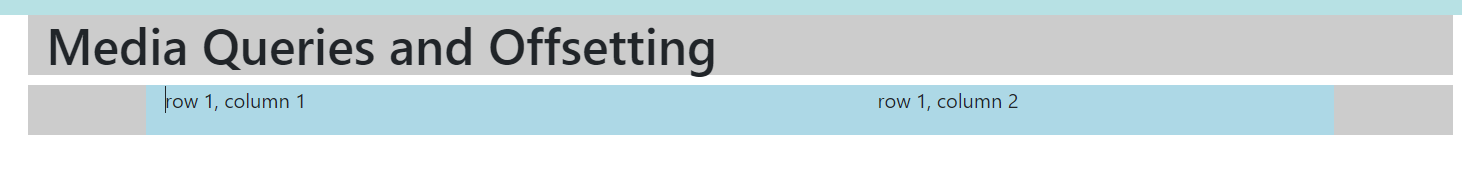
3. Observe the page in the browser.



Notice that the gray column expands across the screen, but the blue columns do not extend to the edge of the gray column. This is because row 1, column 1 takes up 6/12 spaces and row 1, column 2 takes up 4/12 spaces. This totals 10/12 spaces between each column. That is why you see the gray on the far right. Those are the 2 additional available spaces. You can use offset classes to offset these columns so they are more centered.

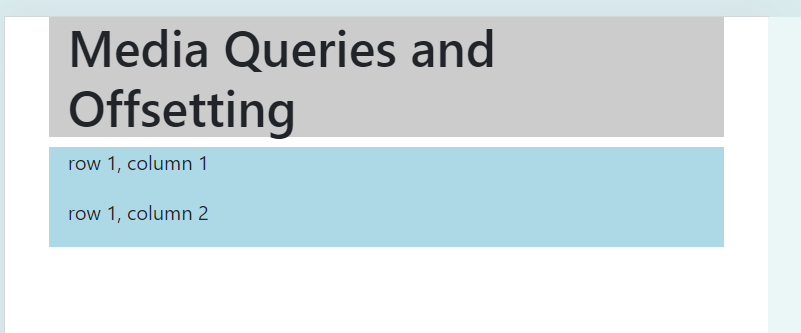
Update the code accordingly. 

4. View the changes in the browser:



Notice that row 1, column 1 is offset by 1 on larger screens. The columns appear more centered in the page. You can experiment and change the value of the offset class to 2 and see what happens.

*Quick note: Because each column has a class of col-sm-12, they break to single column layouts where each column takes 12/12 spaces on small screens.*



Offsets can be applied at larger screens, medium screens, and small screens. To learn more about offsets, check the documentation: <https://getbootstrap.com/docs/4.3/layout/grid/#offsetting-columns>

5. Let’s add some media queries to this page to understand how media queries work

Please understand these media queries are for demonstrative purposes only.

Open the mediaqueryoffset.css and add this media query to the existing code:

@media screen and (max-width: 480px) {
  body {
    background-color: black;
  }
  
  .container {
   background-color: #212226;
   color: #fff;
  }
  
  .colbg {
   background-color: #0e1c57;
   color: #fff;
  }
}


6. View the page in browser in responsive view. Launch the page in the browser and use the key combo ctrl+shift+i

Viewing page in responsive view
Device toolbar is toggled on
Dimension is set to responsive
Width is set to 350

You can adjust the view using the handle bar on the right side of the page to make it bigger or smaller. You can also enter values in the boxes above (in the image above they read 350 and 573 – those are pixel values for the screen’s height and width).

If you scale the page up and down, you will notice these changes only take effect when the screen is 480px or smaller.

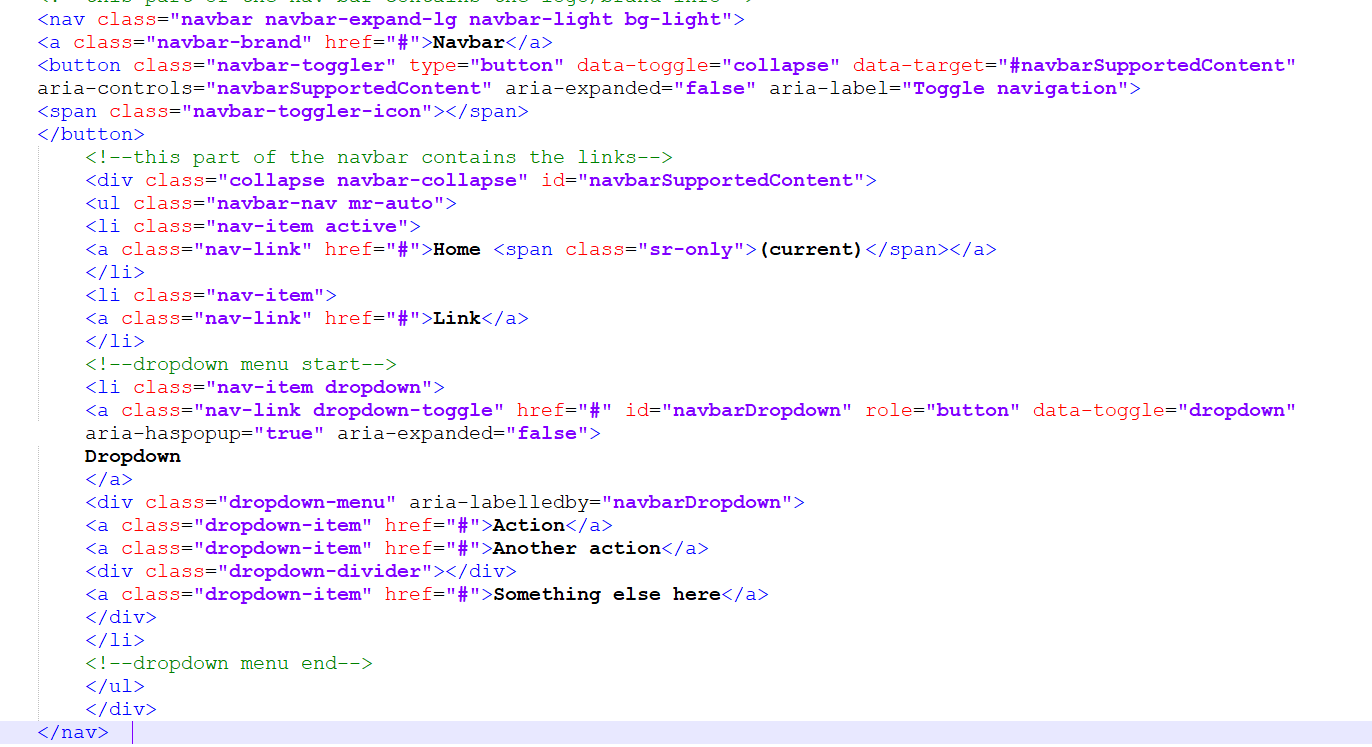
# Forms, Jumbotrons, and navbars

Next, you will work with another file. You will modify a navbar using navbar classes, add a jumbotron, and modify a form using bootstrap classes.

1. Open the starter file called FormNavJumoTron.html

2. Edit the html file in a text editor. You will observe it contains a bootstrap navbar (this navbar was obtained from the bootstrap documentation and modified: <https://getbootstrap.com/docs/4.3/components/navbar/> ). You will also see there is a form that was taken from the documentation page. You will modify it later too. (<https://getbootstrap.com/docs/4.3/components/forms/> ).

3. Observe the navbar code and the comments within it.



Keep in mind this navbar has been modified from the example taken from the documentation page referenced above. Take note of the classes that are applied to the nav element: navbar, navbar-expand-lg, navbar-light, and bg-light

Additionally, note that the navbar-toggler class – this applies functionality when the screen is resized where the navbar converts into a hamburger button. This is mobile friendly navigation.

Furthermore, note that each standard nav item is enclosed in a lit item tag with a class of nav-item and each anchor tag has a class of nav-link.

Lastly, there is a dropdown menu (optional to include) and each dropdown anchor tag has a class of dropdown-item.

3. Observe the form code

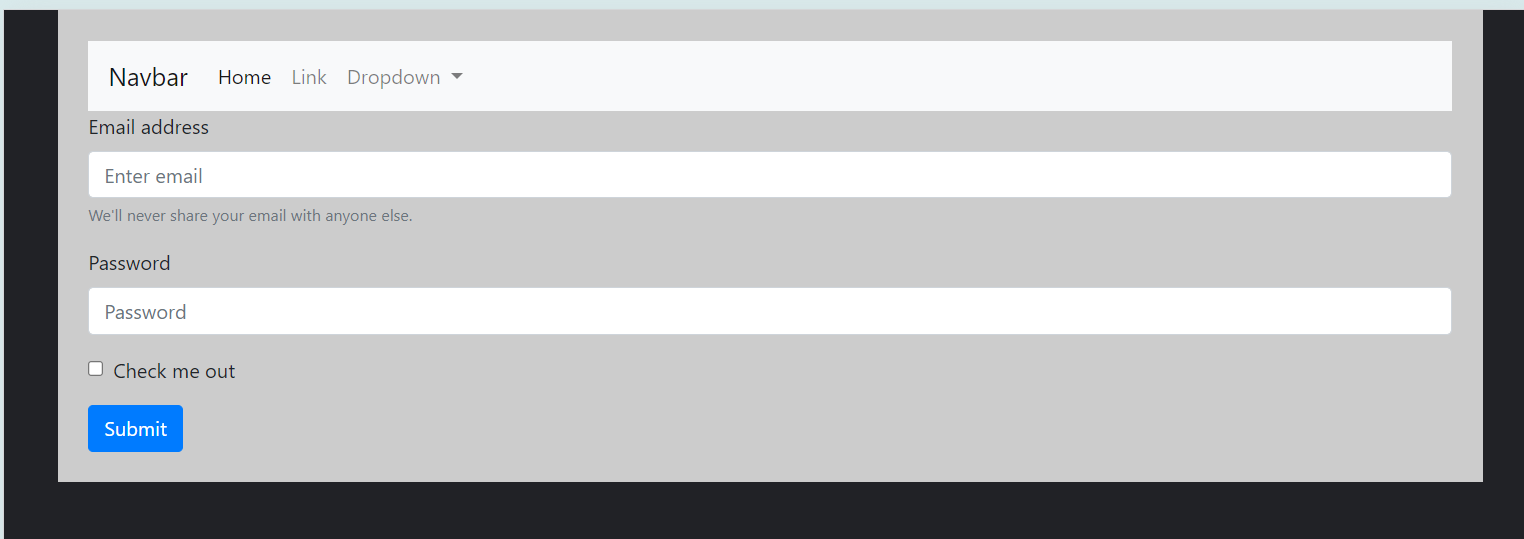


Note that the labels and inputs are organized into divs with a class of form-group each. Also, each input element has a class of form-control except for the checkbox. The checkbox has a class of form-check-input and the div that contains it has an additional class of form-check.

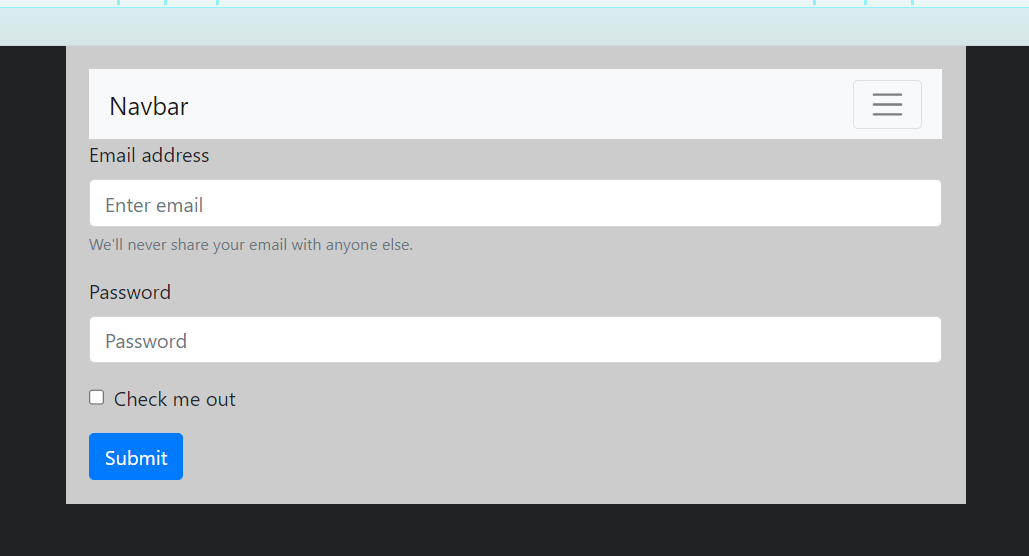
4. Lastly, preview the page in browser. It is linked to the FormNavJumboTron.css stylesheet

View the page on large screens and smaller screens.

Large screen preview



Small screen preview



Note that the navbar converts to a hamburger button on the small screen view. You can click the button to reveal the links. Remember, this hamburger button works because of the JavaScript that is linked in the HTML documents. Without the JavaScript reference, this navbar would not function as we want it.

Ok, now that you’ve observed the code and have a better understanding, time to dive in.

Modify the navbar by changing the classes in the nav element.

5. Change the bg-light and navbar-light class to bg-dark and navbar-dark on the nav element.

Modified code outlined in red in the image

nav element classes bg-light and navbar-light classes were updated to 

bg-dark and navbar-dark

6. Next, change the link text of the anchor tag wit ha class of navbar-brand to read WDD222

Modified code outlined in red in the image

The link text was changed from Navbar to WDD222

7. View the page in browser



Notice that the navbar now has light text and a dark background.

Next, we’ll modify the form.

The form already has all off the layout classes in place, but we have some options for styling.

8. For example, we can add the form-control-sm class to the inputs for email and password

Add the form-control-sm class to the input element for email address and password. Do not remove the existing form-control class.

Modified code outlined in red

form-control-sm was added to the input for email and password classes.

9. Next, modify the button on the form. We can change the color and change the way the button displays on the screen.

Let’s first change the color. Modify the HTML accordingly. Add an ID to the button called “submit.” id=”submit”

add the code outlined in red to the button element. code outlined in red is

id="submit"

v

10. Next, update the stylesheet FormNavJumboTron.css to contain the follow style.

Style the id of button with a background-color of your choice. In the screenshot below, I chose a green color. Feel free to choose whatever color you prefer.

See the CSS below:

added CSS for the #submit selector

#submit {
 background-color: #32a852;
}



11. Now update the HTML for the button again. This time, you will remove the btn-primary class and add the btn-block class. The classes for the button will then be: class="btn btn-block"

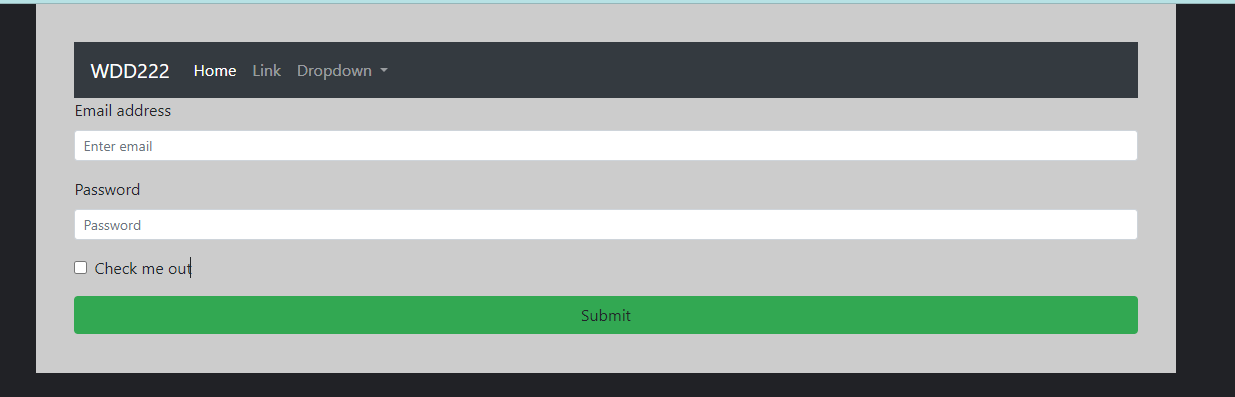
See the updated HTML below (code to modify is outlined in red)

Modified code outlined in red applied to button element. Code outlined in red reads

class="btn btn-block"

v

Observe the changes in browser.



You will see the button is now green and spans across the entire screen.

Lastly, you will explore jumbotrons. Jumbotrons are a great way to add content you want to call out to. You can find and edit pre-built jumbotrons on the bootstrap documentation page: <https://getbootstrap.com/docs/4.1/components/jumbotron>

For the last portion of this exercise, complete the following on your own

* Find a pre-built jumbotron to add to your page from the bootstrap documentation
* Modify the content – be creative while keeping in mind this is just for practice. Add text/an image, etc.
* Add one custom style to the jumbotron that overrides the bootstrap default styling (update jumbotron class on the FormNavJumboTron.css to modify the jumbotron.

Once complete, upload your zipped files to the dropbox.