**Building Your Website**

In this step you’ll make your home page ‘responsive’, which means that it will adapt its layout for various screen sizes.

Start by adding a media query to **main.css**, with the break point set to 800px. The code should look like this:

**@media all and (max-width : 800px) {**

 **/\*We’ll put CSS rule sets here in a minute\*/**

**}**

The rule sets that we put in the media query will take affect when the browser window is 800px wide, or smaller.

Inside the curly braces for the media query, put CSS code that does the following:

1. Set the **height** of the HEADER element to **44px**. We want to make the header a little shorter for small screens, which will leave more room for the main content.
2. For the content DIV (the DIV that has the id of ‘content’), set the **display** property to **block**. In the previous step, we set the display to ‘flex’, which caused the child elements (the MAIN and ASIDE elements) to align horizontally. But for small screens, we don’t have room for that. So, by setting the display to ‘block’ in the media query, the children will align vertically.
3. Set the **width** of the MAIN and ASIDE elements to **100%**. In the previous step we set their widths to 80% and 20%, but since they will align vertically for smaller screens, we want them to stretch out to be as wide as possible.

Make sure to test these changes in the browser by shrinking the window width as small as possible. You should see the changes kick in when the browser window width hits 800px.

If we add many links to our nav bar, it will not look good on small screens (the links will wrap, which looks strange). Instead, we’ll hide the nav bar for small screens, and display a ‘menu’ button that the user can press on to see the nav links.

Start by adding the following DIV element just after the FOOTER element (later we’ll absolute positioning to move it to the top of the page):

**<div id="menu-button">&#9776;</div>**

Note that it has an id of ‘menu-button’ and that we are using an entity character for its content. This entity character appears as a menu icon.

We don’t want this DIV to appear for larger screen sizes, so add a ruleset that targets the DIV by its id, and set its **display** property to **none** (make sure you don’t put this rule set inside the media query – you should put it before the media query)

We do want the DIV to appear for smaller screens, so add a ruleset inside the media query that targets it. And inside the ruleset, add code that does the following:

1. Set the **display** to **block**
2. Set the **position** to **absolute**
3. Set the **top** to **0**
4. Set the **right** to **0**
5. Set the **height** to **44px**
6. Set the **text-align** to **center**
7. Set the **line-height** to **44px**
8. Set the color to a color that you think looks good.

If you load the page in the browser, you should see that the DIV is only visible for screens that are smaller than 800px wide.

Next, we’ll work on the appearance of the nav bar for small screens, it will initially be invisible on small screens, and will appear when the menu button gets clicked (this will require some JavaScript code, but you’ll get that soon).

Add a rule set inside the media query that targets the NAV element, set the **height** to **auto**, and set the **display** to **none**.

Add a ruleset inside the media query that targets LI elements that are inside the NAV element, set the **display** to **block.** We previously set the display to inline so that the LI elements would align horizontally for larger screens. But for small screens, we want them to align vertically.

Now add this rule set inside the media query (it will be used by the JavaScript code that we add soon):

**nav.open{**

 **display: block;**

**}**

Finally, we just need to add some JavaScript code that will toggle the nav bar for small screens. At this point in the course, we haven’t gotten into JavaScript. So, you don’t need to understand the code that I’m about to give you.

First, create a folder named **js** inside the **my-website** folder. This folder will contain all of our JavaScript files.

Inside the **js** folder, create a file named **main.js**

Paste the following code into main.js:

**window.addEventListener("load", function(){**

 **const menuButton = document.getElementById("menu-button");**

 **const navBar = document.getElementById("main-nav");**

 **menuButton.addEventListener("click", function(){**

 **if(navBar.classList.contains("open")){**

 **navBar.classList.remove("open");**

 **}else{**

 **navBar.classList.add("open");**

 **}**

 **});**

**});**

And the last step is to link your homepage to the main.js file. Add this SCRIPT element inside the HEAD element of your homepage:

**<script type="text/javascript" src="js/main.js"></script>**

Now test the page in the browser, and your homepage should work just like it did for the CSS layout activity that you completed recently.