## **Navigation**

Exercise 1 / 5 of Responsive Website: Bootstrap and Plug-Ins

Learning Goals

* Create fixed header navigation using inline lists across all web pages
* Style and position navigation bar by modifying display, positioning, and box model properties
* Style navigation links using nested selectors, pseudo-classes and CSS properties, including color, size, font, text decoration, and hover effect
* Add CSS transitions, taking into account different browser compatibility

Add Navigation Links

http://www.wikihow.com/Create-a-Dropdown-Menu-in-HTML-and-CSS

Ready to begin? Open up index.html in Sublime and Chrome. Make a div with class “nav."

<div class="nav">

A typical way to make navigation bars is an **unordered list.**Create a new unordered list element inside your “nav” div, just like we did in Lesson 3.

<ul></ul>

Inside each list item, make an anchor tag with the link you want it to go to, or a placeholder, for each of the sections: home, about, contact, work, and faq.

<li><a>Home</a></li>

To make "Home" or "Project Name" link to itself (your index.html) page, create a hash tag link, like so:

<a href="#">Home</a>

Navigation Bar with Nested Selectors

What we have so far looks like a list, not a navigation bar! To lay out the list items in a horizontal instead of vertical line, start by setting the **list items** (not the whole list) to the following display property in your CSS file.

display:inline;

To specify list items in the unordered list in the nav div (rather than the whole list) you need to use a **nested selector.** A nested selector is a selector inside of another selector. (Remember that webpage elements are organized like Russian dolls - stacked inside one another. They are boxes that “nested” within larger boxes.)

 Use the following CSS rule:

.nav ul li

See how li (list item) is nested inside of ul (an unordered list)? Great! Now your list is horizontally displayed.

But let’s say you want to create a navigation bar. In your CSS file, set the background-color for my “nav” div to black.

.nav {

background-color: black;

}

Hmm. I can’t really see my navigation items now. I want my list items to appear in white, but when I try to set them, it doesn’t work!

Create a nested selector to set the**color** on the **anchor** tag.

.nav ul li a

Make the navigation links all in uppercase letters. Add the **text-transform** property to the list items (remember that it is nested) and give it a value of **uppercase.**

text-transform: uppercase;

Add**letterspacing**to make your links more readable. Add the letter-spacing property to the list items as well. I’ve set mine to**letter-spacing: 0.05em;**

Navigation Bar: Display and Positioning

We want our navigation bar to be stuck to the top of the page and visible even if we scroll down. This is called a**fixed**navigation bar. It is a common navigation style and one that we will see later in the course to implement scrolling navigation. Creating a fixed navigation bar is easy - just give the “nav” div a position of fixed.

position: fixed;

 Now make the navigation bar stretch to fit the width of the entire page. In your CSS file, adjust the **width** of the nav bar to make it 100%. Use the CSS **width**rule and set the property to 100%.

width: 100%;

Add **padding** to each list item on the top and bottom and a little bit to the sides. (We used 20px 10px 20px.)

padding: 20px 10px 20px;

REMEMBER PADDING?The padding property can have from one to four values.  
padding: 25px 50px 75px 100px;  
top padding is 25px  
right padding is 50px  
bottom padding is 75px  
left padding is 100px

padding: 25px 50px 75px;  
top padding is 25px  
right and left paddings are 50px  
bottom padding is 75px

padding: 25px 50px;  
top and bottom paddings are 25px  
right and left paddings are 50px

padding 25px;

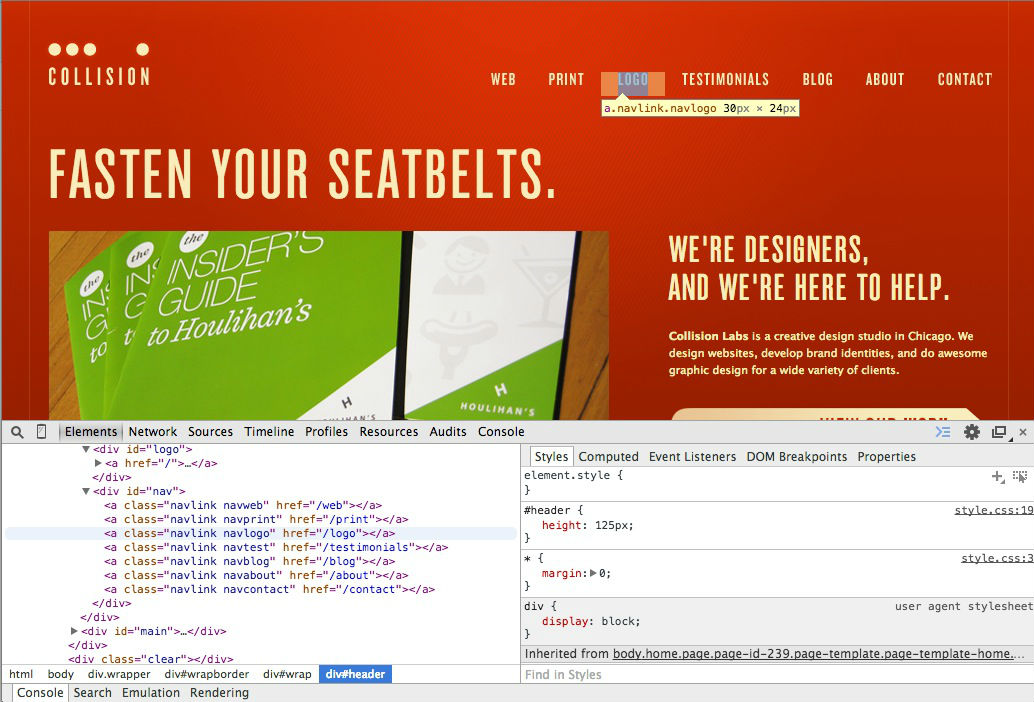
all four paddings are 25px

Now, use **float**property to move your unordered list to the right.

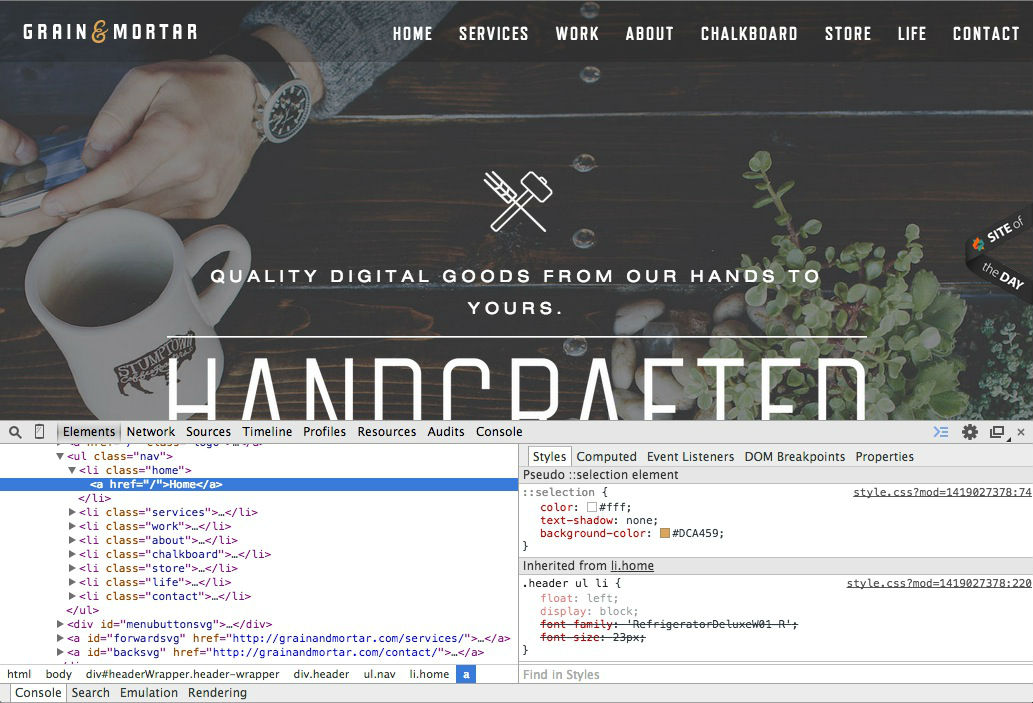
TIP!If you want your “home” link to appear on the left and the rest of your navigation links to appear on the right, take “home” out of the unordered list. You can also style it differently (or replace it with a logo). It’s still part of the div with the class of <nav> but now you can position it using margin and padding that applies only to your home link and not all the rest of your unordered list items. Use the **nested selector** .nav a to select the only link that is NOT part of an unordered list in the div. Because your .nav ul li a selector is much more specific, its styles will not be affected.

**Styling Your Navigation Bar**

Play around with styling using display and positioning properties until you are happy with your navigation. You can look for examples of navigation bars that you like and inspect the code using browser tools. (Mac Users: Just hit **Command + Option + I** on any webpage. That's **Control + Shift + I** on PC.) Click the magnifying glass icon to hover over navigation bar elements to see which display and positioning properties are applied.



[CollisonLabs](http://collisionlabs.com/) has a navigation bar that has the float property is applied, as well as padding to move the navigation bar down from the top of the page.



Another style of a fixed navigation bar is to display a colored box when the user hovers over the link, as seen in[Grain and Morter's](http://grainandmortar.com/) website. Can you use "Styles Tab" of developer tools to discover how this was done?

Your mentor can serve as a great resource to learn different navigation best practices and styles.

Styling with Pseudo-Classes

Wouldn’t it be kind of cool if the links turned a different color when you put your mouse over them? This is called a**hover** effect. With **pseudo-classes,**you can also set a different hover color for the nav items.

A pseudo-class is used to define a special state of an element. It is most often applied to links, since links have different states. For example, a link can be visited, unvisited, or in the process of being clicked (i.e. when a user mouses over or “hovers” over a link).

The syntax looks like this:

selector:pseudo-class {

property: value;

}

To add styling to a pseudo-class, make a new CSS rule with the anchor tag as the selector. Remember, the syntax for the anchor tag in CSS is just an “a.” Add :hover as its pseudo-class, and set the color to #ff0.

.nav a:hover {

color: #ff0;

}

Hmm... I don’t like the fact that my navigation links get underlined when I hover over them — I want to add a custom style instead! I'll add **text-decoration: none;** to my hover pseudo-class, and you can, too.

.nav a:hover {

color: #ff0;

text-decoration: none;

}

Custom Styles and Transitions

To make your hover effect appear more smoothly and naturally, add a CSS transition. This gets added to the rule for the link itself, not its :hover pseudo-class.

transition: color 600ms;

The number is in milliseconds.

Be mindful of how different browsers read CSS. For example, for Safari to understand this property, you need to repeat it with the **vendor prefix** -webkit- like so:

-webkit-transition: color 600ms;

Learn More

Can’t get enough of fixed navigation bars? You can apply what you have learned about nested selectors to create a drop down menu in CSS as well. You nest another unordered list in your main unordered list. Ask your mentor if you are interested in learning more about how to implement these advanced features of navigation bars!

[Here’s](http://css3.bradshawenterprises.com/which-vendor-prefixes-are-needed/) a comprehensive list of more **vendor prefixes** you will need as you play with more advanced effects.

[http://careerfoundry.com/images/questionmark.png](http://careerfoundry.com/steps/421#question)

Task

By the end of this lesson, you should be able to:

* Create fixed header navigation using inline lists across all web pages
* Style and position navigation bar by modifying display, positioning, and box model properties
* Style navigation links using nested selectors, pseudo-classes and CSS properties, including color, size, font, text decoration, and hover effect
* Add CSS transitions, taking into account different browser compatibility

Your assignment: Since these changes you made on your index.html page won't be reflected on your About and Contact pages, you will have to copy the navigation code into those pages as well. Add the navigation bar with links on your index.html, about.html, and your contact.html.

## **Bootstrap Components and Plugins**

Exercise 2 / 5 of Responsive Website: Bootstrap and Plug-Ins

#### Learning Goals

* Link Bootstrap file and describe use of Bootstrap framework, components, and plug-ins
* Apply Bootstrap and customize Bootstrap glyphicon components for Contact page
* Apply and customize Bootstrap collapse plug-in to create Accordion for FAQ page
* Read documentation and Stack Overflow to troubleshoot HTML, CSS, and Bootstrap templates before clarifying with a mentor

#### Introduction

Now that you’ve gotten down the basics of HTML and CSS, we’re going to get fancy by adding components from a front-end framework called Bootstrap. A**framework**is a collection of tools for creating websites and web applications. Frameworks exist for both front-end and back-end web development. For example, in a few weeks, you’ll learn about Rails (as part of Ruby on Rails), a back-end framework for web applications.

In this lesson, you’ll learn not only how to use the Bootstrap framework but customize it as well using Bootstrap’s **documentation**- a user guide for all of Bootstrap’s capabilities and functions.

#### What is Bootstrap?

[**Bootstrap**](http://getboostrap.com/)is a front-end framework that contains HTML and CSS design **templates** for typography, forms, buttons, navigation, and other user interface components, as well as JavaScript extensions or **plug-ins**. Templates make our lives easier because they can add automatic styling and functionality.

Bootstrap is a preferred web development framework for many reasons. Bootstrap is **open source**and available on**GitHub**. Developers are encouraged to participate in the project and make their own contributions to the platform.

Bootstrap also makes it easy to implement responsive design through its built in grid system (Lesson 9) and contains mobile-first styles, ready to use for modern browsers. Bootstrap also contains many free **components,**such as glyphicons, that we will use today.

To use Bootstrap components and plugins, we need to link Bootstrap CSS and JS files to our Contact page. For ease, we are going to link Bootstrap through a **content delivery network**(**CDN**). This basically means that instead of using a Bootstrap file in your local computer, we are going to use Bootstrap components from a remote server, which includes fonts called glyph icons we are going to be using today. Add the following code in the <head> tag of each HTML file.

*<!-- Latest compiled and minified CSS -->*

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">

*<!-- Latest compiled and minified JavaScript -->*

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>

#### Bootstrap Components: Glyphicons

Let's add a glyphicon for each Contact detail on your Contact page, including your Email, Address, and Phone Number. A Bootstrap component, glyphicons are helpful because they can act like visual bullet points and break up text. Because they are part of the **span** class, glyphicons can be inserted easily into the middle of a line or a paragraph.

**Span classes**have **inline styling**, which means they will be part of the existing line of text - and not be broken up into a new one like **block** elements, such as <p> or <div>.

**Glyphicons**are not actually images. They are an **icon font**- a custom font that use glyphs instead of letters. (Remember Wingdings or Webdings? That brings back memories of the Word processor from the 90s... but it’s the same idea!)

Go to the Bootstrap documentation for[components](http://getbootstrap.com/components/)and pick at least 3 glyphicons you want to include to help draw your visitor to key information on your Contact page.

Identify the name of each glyphicon. Then, use the following code to insert it. Put this <span> inside the element containing your text, such as a <p> or <li>

<span class="glyphicon glyphicon-nameofyourglyphicon"></span>

The first “glyphicon” is a generic Bootstrap class that contains styling common to all glyphicons. The second “glyphicon” is the specific glyphicon. Replace “nameofyourglyphicon” with the glyphicon you would like to use.

Check out this example of how icons were used in a professional [Contact](http://infinitcolours.com/en/contacts) page! (Although these icons are not glyphicons, you can create a similar effect).



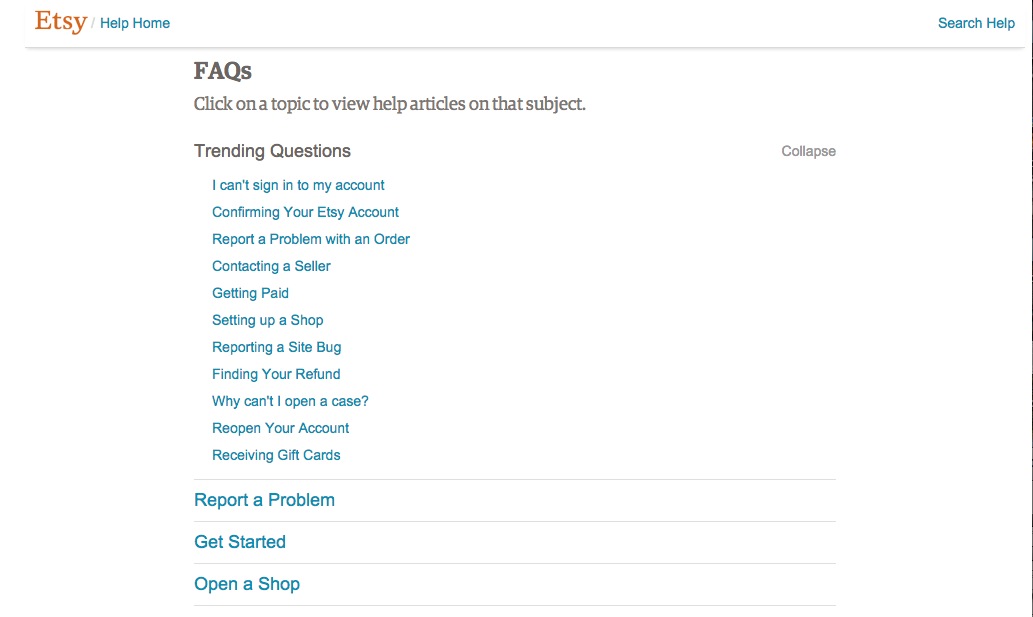
### **Glyphicon Transitions and Effects**

You can add transitions to glyphicons using **pseudo-classes**, just as you did for the hover effect on your links. Refer to **Lesson 5** to add hover properties to the glyphicons. We can add these hover properties because glyphicons are considered to be part of text, rather than images.

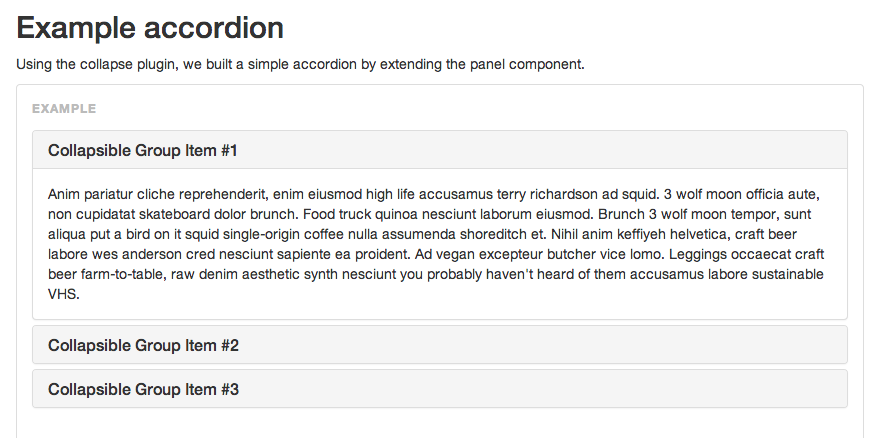
#### Bootstrap Plugins: Accordion

You might have some additional content about your work or services that might not fit on your About page. Create a new file for a FAQ (Frequently Asked Questions) page titled faq.html. Insert required HTML header tags. Add a header and some <div> sections.

There are many ways to organize questions and answers on a FAQ Page.

  
You can see [Etsy’s FAQ page](https://www.etsy.com/help/topics" \t "_blank) as an example below. Etsy has a cool accordion effect that allows you to expand and collapse content.

We will implement a similar effect in our FAQ page. We will create this accordion using a plug-in packaged in Bootstrap called **collapse.** Read the documentation and see a demo of [Collapse](http://getbootstrap.com/javascript/#collapse) here.



Think of 5 questions visitors may ask about you when they come to your page that are not answered on your About page. (Some suggestions: “Why did you decide to become a web developer?” “What is your design process?” “Who are your clients?” “What is your average project turnaround?”) Write down your questions and your responses in a separate word processor document for now. You'll move your text into your accordion shortly.

#### Documentation and Debugging

In the code block for an accordion below, some of the classes are mistyped. Have no fear! More often than not in web development, something has gone wrong, and most of the time, you’re debugging code. Debugging and reading documentation are core web development skills.

DOCUMENTATION

When you encounter problems with Bootstrap plug-ins, the best thing to do are a) read the documentation and b) ask Google and consult [Stack Overflow.](http://www.stackoverflow.com/) Let's go to the Bootstrap documentation to “debug” this code. Replace the missing code with the right data type.

<div class="panel-group" id="accordion">

<div class="panel panel-default">

<div class="panel-heading">

<h4 class="panel-title">

<a data-toggle="collapse" data-parent="#missingcode" href="#collapseOne">

Collapsible Group Item #1

</a>

</h4>

</div>

<div id="collapseOne" class="panel-collapse collapse in">

<div class="panel-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.

</div>

</div>

</div>

<div class="panel panel-default">

<div class="panel-heading">

<h4 class="panel-title">

<a data-toggle="collapse" data-parent="#missingcode" href="#collapseTwo">

Collapsible Group Item #2

</a>

</h4>

</div>

<div id="collapseTwo" class="panel-collapse collapse">

<div class="panel-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.

</div>

</div>

</div>

<div class="panel panel-default">

<div class="panel-heading">

<h4 class="panel-title">

<a data-toggle="collapse" data-parent="#missingcode" href="#collapseThree">

Collapsible Group Item #3

</a>

</h4>

</div>

<div id="collapseThree" class="panel-collapse collapse">

<div class="panel-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.

</div>

</div>

</div>

</div>

Once you have finished using the documentation to solve the programming error, replace the default Lorem Ipsum text in the “panel-body” class with your content and “Collapsible Group Item #1” with your question. Insert this entire code block inside a container div.

DEBUGGING

Another common reason why functionality on your page will not work is because you do not have the right JavaScript or CSS files linked in the header of your HTML file. Your accordion needs a jQuery file, a JavaScript library, in order to work. Add the following code to your header.

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

Your accordion should now work!

Whenever something goes wrong, check whether you have the right libraries linked as the first step, which you can often find through the documentation.

#### Styling your Accordion

Style your accordion with CSS so it doesn't look like the default Bootstrap version.

What kind of font and text will your answers be? Modify the **panel-body**class in your CSS file.

Can you give your headings a different background color? Modify the **background-color** of the panel-heading class.   
What about your heading text? Modify the **panel-title**styles in your CSS file.

You can also add**glyphicons** to make the headers more prominent as well. We added a "+" glyphicon using this code below, but you can return to the Bootstrap documentation to choose your own.

<a data-toggle="collapse" data-parent="#accordion" href="#collapseOne"><span class="glyphicon glyphicon-plus"></span>Collapsible Group Item</a>

#### Learn More

Throughout this lesson and the rest of the front end development unit, you may want to take a look at the Bootstrap documentation in more detail to understand its full capabilities. However, some of this stuff can get pretty technical and feel overwhelming very quickly, so we don’t expect you to read and understand everything overnight. Reading documentation, like debugging, is a skill that you will develop!

**More Documentation**

Looking for an easy documentation task? To experiment further with glyphicons, find the code block in Bootstrap documentation that allows you to create **accessible**icons.

**Ask Stack Overflow**  
[Stack Overflow i](http://stackoverflow.com/)s a platform and library for questions and answers about programming. You can Google any question or error you come across. If you've had an issue before, more likely than not, another web developer has, too!

[http://careerfoundry.com/images/questionmark.png](http://careerfoundry.com/steps/422#question)

#### Task

* Link Bootstrap file and describe use of Bootstrap framework, components, and plug-ins
* Apply Bootstrap and customize Bootstrap glyphicon components for Contact page
* Apply and customize Bootstrap collapse plug-in to create Accordion for a new FAQ page
* Read documentation and Stack Overflow to troubleshoot HTML, CSS, and Bootstrap templates before clarifying with a mentor
* Update your Contact.html page and send your mentor the link to your faq.html page.

## **Advanced Images**

Exercise 3 / 5 of Responsive Website: Bootstrap and Plug-Ins

Learning Goals

By the end of this lesson, you will be able to:

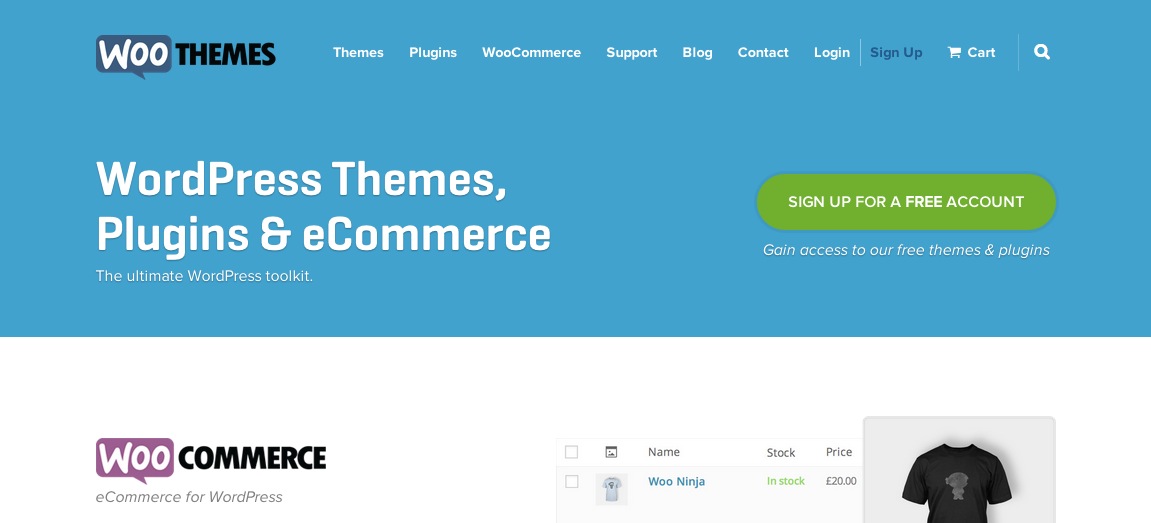
* Add hero images (oversized images for hero headers) to grab viewer’s attention
* Crop and resize images for hero header using basic imaging software, save images appropriately for web, and explain difference between image file types.
* Implement Bootstrap carousel plug-in with carousel captions and adjust layout using container-fluid class.
* Advanced Level: Describe how to measure loading times of various page components, such as images, and optimizes loading times

Introduction

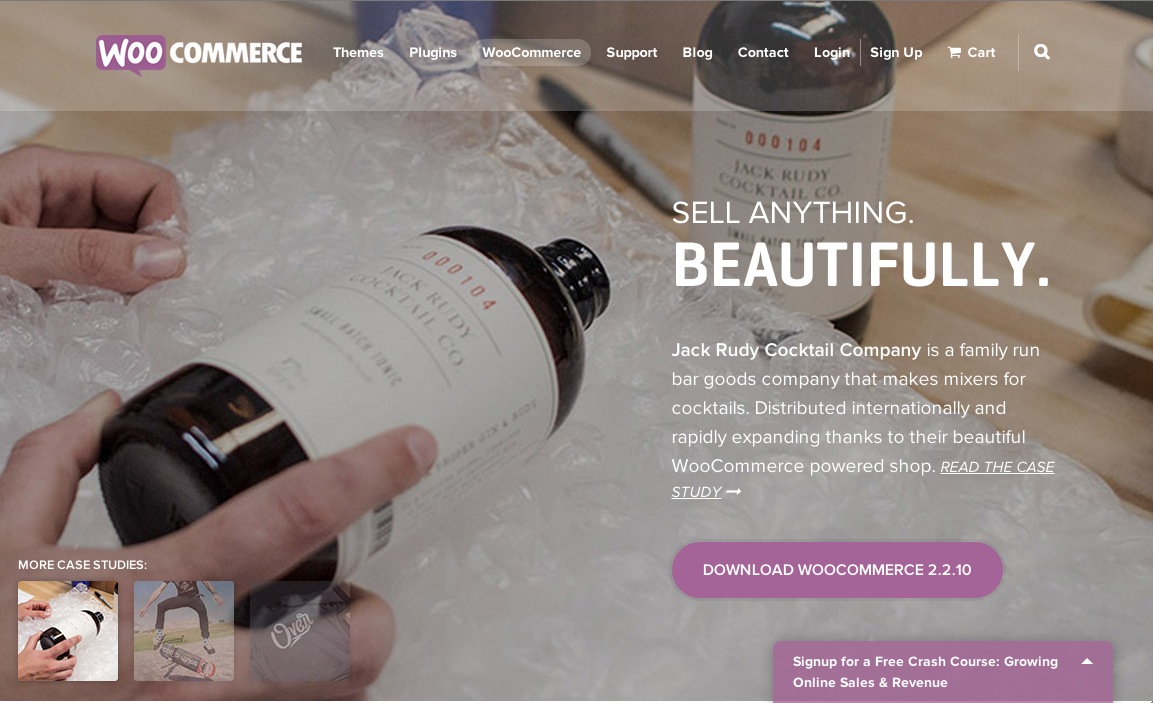
As you have already learned, there are two ways we can add images in HTML and CSS.  
With HTML, you add an <img> tag and alt text. With CSS, you can add it as a background-image in a div. In today’s lesson, you’ll learn how to insert even bigger images, called **hero images,** and create an **image carousel.**

#### Hero Images

Our jumbotron is a Bootstrap class that contains a large call-out section that draws the attention of our users. We want to replace it with a large image that fills the size of the screen. Why?



Check out an example of a jumbotron section from [WooThemes](http://www.woothemes.com/" \t "_blank) versus its sister site, WooCommerce, which uses a **hero image**below.



Which is more effective? We can clearly see that the bigger image grabs our attention more.

One of the biggest trends in web design right now is the use of big images. And not just large, but completely oversized images that seem to come to life on screen. When used at the top of a page, this is often referred to as a “**hero header.**” Hero headers are preferred by many developers and companies because of its simple design that relies on a strong photographic image and its ability to focus the visitor on a main marketing messaging.

Insert a hero image into your website. First, find an image large enough to fit the size of a large browser window. For now, let’s use the default setting of 1920 x 1080, one of the largest browser screens sizes. You can use this free hero image we found for now. Click [here](https://s3.amazonaws.com/coach-courses-us/public/courses/fronend/portfolioheroimage.jpeg) to save the image to your img folder.

Your current image in the background of your jumbotron is probably not big enough to cover the entire browser screen. Never resize an image so that it is larger than the original version. Doing so will stretch your image or cause it to lose quality. Look for a larger image instead.

#### Size and Save Images for the Web

You may need to resize some of your images so they fit 1920 x 1080 (or whatever browser size you end up building for). You can do this by cropping or resizing in an image editing software such as Photoshop or [Pixlr,](http://apps.pixlr.com/editor/" \t "_blank)an online Photoshop editor, and then saving the file for the web. In Photoshop, make sure you click "Save for Web & Devices" to reduce the file size.

Directions for Pixlr:

* Load the image into the workspace.
* Go to Image -> Image Space and constrain the width to “1920.” The height might be taller than 1080px.
* Click on the “crop” icon and Constrain “Output size.” Constrain the width to 1920 and the height to 1080. This will create a fixed box with the dimensions 1920 x 1080.
* Drag a box around the region of your image you want to keep. Hit enter to enact the changes.
* Save your images for the web. You may reduce the quality of your images to about 80% to reduce file size without losing too much image quality.

But wait! What file do you save the images as? Do you keep the default file type? When would you choose a different one?

JPG. vs PNG?

The two most commonly used (and most commonly supported) web image formats are JPG and PNG. The other popular format you have no doubt come across is GIF, which is still used for animated images but not widely used anymore when building websites. While you can interchangeably use JPG and PNG for images, it's good to keep in mind what the strengths and weaknesses of each format are. The JPG format was designed to compress large images into small file sizes, and it does that well enough that it has become the de facto standard for web images. As a result, JPGs are best suited for large photographic images, but smaller images with details may become blurred or pixellated by the compression. This is where the PNG format is useful. PNG files are larger in size but they can reproduce details well. PNGs have another feature, which is transparent backgrounds. These attributes make PNGs well suited for images like logos, UI elements and icons.

#### Adding Background Images

Set the image as the **background-image** of the htmltag, linking to the image correctly inside your img folder. Insert the file path of your image below.

We add CSS properties, such as **background-size: cover;**, which we have used before and **no-repeat**. Our image has a postion of **center** and a display property of **fixed**.

html { background: url(img/yourfilename) no-repeat center center fixed; -webkit-background-size: cover; -moz-background-size: cover; -o-background-size: cover; background-size: cover; }

Image Carousel

Nice job! But what if you wanted more than one image in the background? You can implement an image carousel. An image carousel allows you to add a number of eye-catching large images with accompanying text to give users an interactive overview of your product with links to interior pages.

Check out [grainandmortar’s](http://grainandmortar.com/" \t "_blank) website for an example of a hero image as an image carousel. [Pastini](http://www.pastini.com/" \t "_blank) is another tasty example of how an automatic image carousel which takes up the whole browser screen can be very effective. Notice both website examples have a fixed navigation bar - just like ours!

For your image carousel, you need two additional images to match the size of your hero header. If you do not have any images at the moment, you can use ours. Download them [here](https://s3.amazonaws.com/coach-courses-us/public/courses/fronend/lesson8-portfolioheroimage2.jpg) and [here](https://s3.amazonaws.com/coach-courses-us/public/courses/fronend/portfolioheroimage3.jpeg)

You'll have to crop these images with an image editing software, such as Photoshop or Pixlr again, so they are the exact same size as your first image. Otherwise, your carousel will “jump” or shift places when your user clicks on the next image. After you have cropped your image, go to Save As -> Save images for web and devices, following the directions above.

We are going to use Bootstrap plugin to implement this image carousel. Read [this section](http://getbootstrap.com/javascript/#carousel) of the documentation to see an example of how the feature will look when finished.

Open up your index.html file and copy in the code from the Bootstrap docs into your div with the class of container. Link your images in the <img> tag. Pay attention to the three main sections of code here, denoted by the comments:

* Indicators, which are the dots at the bottom indicating which slide you are currently on
* Wrapper, which contains the images and captions for each slide
* Controls, which are the arrows on the left and right sides of the slide

<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

*<!-- Indicators -->*

<ol class="carousel-indicators">

<li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>

<li data-target="#carousel-example-generic" data-slide-to="1"></li>

<li data-target="#carousel-example-generic" data-slide-to="2"></li>

</ol>

*<!-- Wrapper for slides -->*

<div class="carousel-inner" role="listbox">

<div class="item active">

<img src="..." alt="...">

<div class="carousel-caption">

...

</div>

</div>

<div class="item">

<img src="..." alt="...">

<div class="carousel-caption">

...

</div>

</div>

...

</div>

*<!-- Controls -->*

<a class="left carousel-control" href="#carousel-example-generic" role="button" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="right carousel-control" href="#carousel-example-generic" role="button" data-slide="next">

<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

Add carousel captions.

<div class="carousel-caption">

<h3>...</h3>

<p>...</p>

</div>

</div>

Think about how you want to tell your story or what information you think will be most important to the user when they first visit your site and see your images.

Style your carousel captions**.** Make sure that your text inside carousel-caption is clear on your image carousel. You may need to modify margin and padding of the p class to make sure that your text is positioned at an optimal height and location. You may also change the color or font style so the text is easy to read.

#### Troubleshooting Your Image Carousel

Adding code is the easy part, but we shouldn't be surprised when things don't go exactly as planned. Here are some solutions to common problems you may encounter with your image carousel.

### **Navigation Bar Disappearing**

If you cannot see your navigation bar at this point, it’s probably BEHIND your image carousel. You have two options to solve this problem.

You can add margin or padding the same size as the navigation bar to the top your image carousel to push it down further on the page. For example, let’s give our navigation bar (.nav) a height of 65px. Give the**carousel ID** (#carousel-example-generic) a padding-top of 65px as well.

If you want the image carousel to stay in the same position (because you might want the navigation bar to be transparent in front of your image carousel), you can give your navigation a CSS property called**z-index.**  
The order in which elements stack (meaning, overlap each other) on a website is determined by the **z-index**property. An element’s default z-index is usually 0. To change the stacking order so an element appears in front of another, use the **z-index** property and give it an integer value such as **1.**

### **Image Carousel Too Small**

Your carousel being inside a “container” div may also be causing you problems because the container class includes margins on the sides. To solve this problem, replace **“container”** with **“container-fluid.”**

**“Container”** is a Bootstrap class that creates a responsive, fixed width “container” or box for your HTML elements.

**“Container-fluid”** is a Bootstrap class that will make the container fit the width of the page. If you are still seeing padding on the sides, it’s because Bootstrap sets container-fluid with a default of 15px padding on the left and right. To override Bootstrap’s default settings, go to your CSS file and define your own container-fluid properties. Change the left and right padding from “15px” to “0px.”

### **Did you remove your background image?**

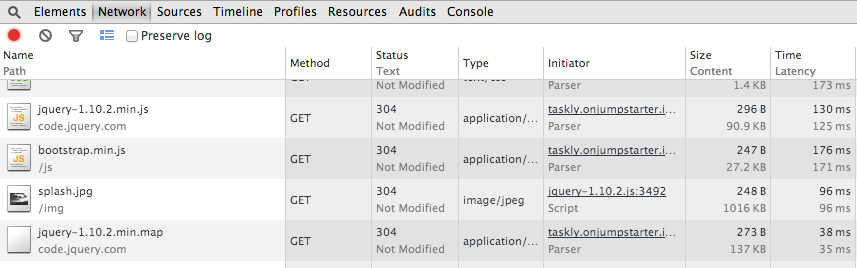
Your image carousel may be overlapping over your background image. Remove your background image property in your CSS on your HTML element, if you haven't done so already.

Still stuck? Try searching [Stack Overflow](http://stackoverflow.com/) or [Google](http://google.com/) with your question and see if other web developers encountered similar issues. Make sure you include any relevant key terms like "bootstrap" and "carousel" to narrow down the context of the problem.

#### Network Tab

One of the downsides of having large, hero images is that they may take a long time to load, slowing down the overall efficiency of your page. You can find out if images are taking too long to load with the developer toolbar. We have mainly used the ‘Elements” and “Styles” tab. Today, we will use the **“Network” tab** to test the loading time of your assets, which include CSS, JavaScript, and images.

Open up your developer tool and click on **“Network.**” Examine the “Size” and “Time” column. Which image is the largest? Which one takes the longest to load? Take a look at the images you're using on your site and make a note of their load time. See how much you can impact their speed by cropping and optimising them.



#### Learn More

HERO IMAGES

While image placement and optimization are important to the overall design of a site, image choice is just as important an ingredient to really sell the story. If you do not want your website to look like every other student’s on CareerFoundry, find, create, and insert your own hero images for your image carousel.

For inspiration on hero images, visit http://designmodo.com/hero-headers/

STOCK IMAGES

You may choose to produce the images yourself or use a stock photo service, depending how important the images are to your site's story. **Stock images**give you the licensing rights to reproduce the images in another context. There are many sites with free stock photography online. [Here is a good review of some of the best resources](http://despreneur.com/15-best-free-stock-photo-websites/).  If you need more specific images, paid sites like [iStock](http://www.istockphoto.com/" \t "_blank) and [Shutterstock](http://www.shutterstock.com/" \t "_blank) offer a massive collection of high-quality images of all categories, even icons and videos.

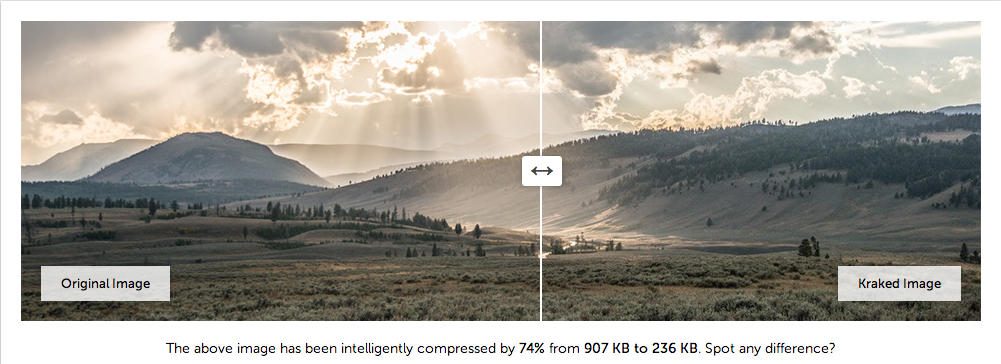
IMAGE COMPRESSION

Because image files can be large in size, they are often the last elements to finish downloading when viewing a website. This can detract from the overall user experience, especially if the images on the site contain information relevant to the content. However there are several tactics for improving image performance. The first of which is to crop large photos to the dimensions of the img element dimensions in which they will display, so the browser doesn't download any more data than necessary, which we have done. While it is possible to resize images using HTML and CSS, it is not recommended because this may stretch or compress the image.

One of the easiest way to decrease the loading time on your website is to **compress** the images to reduce image size without losing quality. For example, you can use [this](http://www.smushit.com/)easy image compressor from Yahoo.

IMAGE OPTIMIZATION

Another common practice is referred to as image **optimisation**, which is essentially another term for image compression but refers to compressing already-saved images. While saving an image as a JPG compresses the file quite a bit, there are some advanced algorithms which can compress the image even further without deteriorating the quality of the image. You can see a good demonstration of this on the homepage of a service called [Kraken](https://kraken.io/).



There are also several free applications available on the web which can automate this process for you.[Imageoptimizer.net](http://www.imageoptimizer.net/Pages/Home.aspx) has a web-based tool, and there is a good desktop app for Mac called [ImageOptim](http://imageoptim.com/" \t "_blank).

SPRITES

Another image performance tactic which we won't cover in depth but you should be aware of is **sprites**. Sprites are a way of combining all of a site's UI elements into one single image file and displaying them separately by using CSS to slice up the image file. This saves page loading time because downloading several small images, even though their file size is seemingly insignificant, requires the overhead of many HTTP requests whereas downloading one larger file only takes one round trip from your computer to the server. You can read more about the history and usage of sprites in [this excellent Smashing Magazine article](http://www.smashingmagazine.com/2009/04/27/the-mystery-of-css-sprites-techniques-tools-and-tutorials/).

Task

By the end of this lesson, you should be able to:

* Add hero images (oversized images for hero headers) to grab viewer’s attention
* Crop and resize images for hero header using basic imaging software, save images appropriately for web, and explain difference between image file types.
* Implement Bootstrap collapse plug-in with carousel captions and adjust layout using container-fluid class.
* Advanced Level: Describe how to measure loading times of various page components, such as images, and optimizes loading times

## **Responsive Web Design**

Exercise 4 / 5 of Responsive Website: Bootstrap and Plug-Ins

Learning Goals

* Use Bootstrap grid system to create columns for small desktop (md) and mobile (xs) screens and edit final layout by modifying padding, margin, and width
* Create a responsive website with at least 2 breakpoints and fluid layout (percentages, relative sizing)
* Add and modify Bootstrap classes to make page elements, such as images, also responsive
* Describe the advantages and disadvantages of designing for mobile-first, and the differences between responsive and adaptive design
* Professional: Create layout for lg and sm screens (up to 4 breakpoints)

Introduction

Responsive websites respond to their viewing environment, displaying an optimal viewing experience no matter what the screen size - small, medium, or large. One of the ways Bootstrap revolutionized web development was by incorporating an easy-to-use grid system for laying out content into rows and columns that respond automatically to different screen sizes. Before the advent of grid systems, developers would have to position elements on a page manually using pixels or percentages. Not only is this process time-consuming but the prevalence of mobile devices with different screen sizes makes this a complex task. Today you will learn how to master the Bootstrap grid system to make your site responsive - ready to go on device!

Media Queries

In a **responsive website,** the layout of the page actually changes depending on the size of the screen. But how does the the responsive website detect different sizes? The answer is **media queries!**

Media queries are simple CSS rules that define behaviors at certain screen sizes, called **breakpoints.** Media queries tell the browser to change layout when the width is equal to a breakpoint, a specific value (320px, 480px, 768px, etc..) that corresponds to a commonly used device.

Here are Bootstrap’s default media queries and break points, which are based on maximum and minimum widths in pixels.

*/\* Large desktop \*/*

@media (min-width: 1200px) { ... }

*/\* Portrait tablet to landscape and desktop \*/*

@media (min-width: 768px) and (max-width: 979px) { ... }

*/\* Landscape phone to portrait tablet \*/*

@media (max-width: 767px) { ... }

*/\* Landscape phones and down \*/*

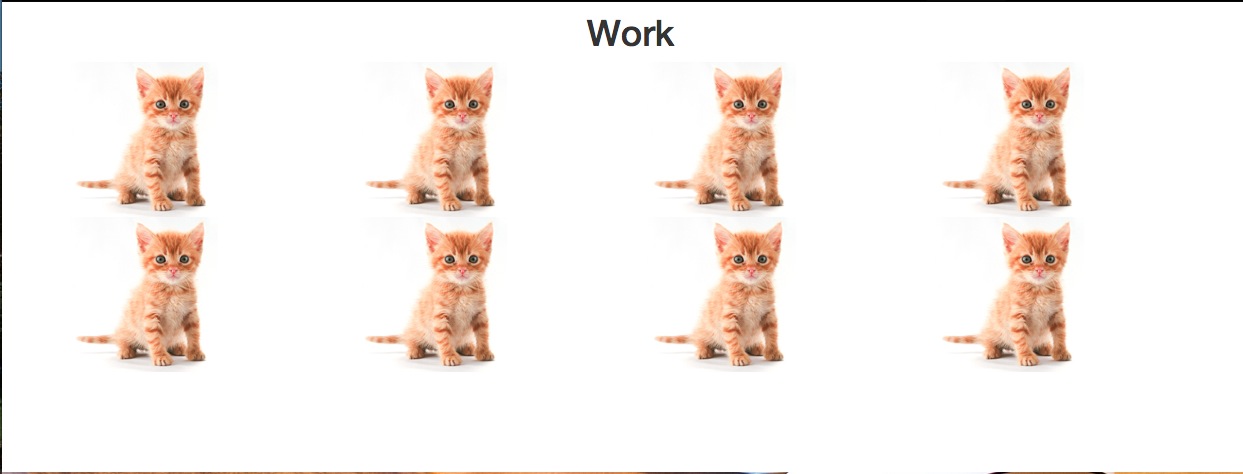
@media (max-width: 480px) { ... }

How many breakpoints does a web designer need to create? In general, every responsive website has a minimum of two breakpoints — one for tablets and one for mobile devices.

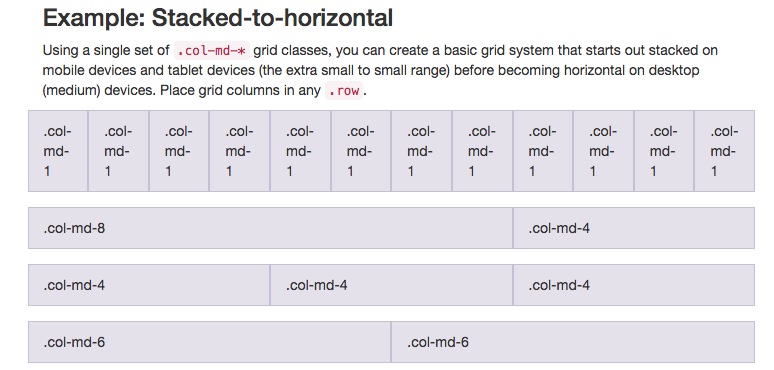
Bootstrap Grid System

Fortunately, Bootstrap does a lot of the work for us in creating a responsive website. Bootstrap divides up the viewport into 12 columns. You can use its grid classes to specify how many of these 12 columns you want a particular piece of content to take up on each size screen on mobile and desktop devices.

We are going to design your “Work” section of your Portfolio page to feature 4 different projects. Each project will have a box containing an image and a box containing text about. For now, we have kitten images as placeholders for boxes. We want it to look something like the example below at large to medium screen sizes.



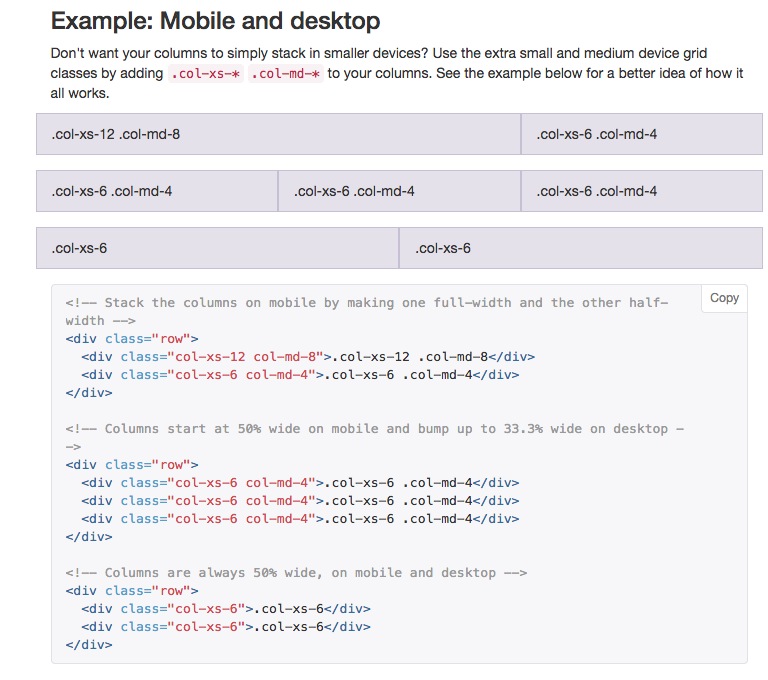
Use the Bootstrap grid system to help guide you. Remember there are 12 columns to a row.



Let's think. On a medium to large screen, we want 4 kittens to a row. We need 4 columns of 2 kittens each. That means that each kitten will take up 3 boxes in the Bootstrap grid system. 3 boxes x 4 columns = 12. 12 is our magic number! Perfect. Let’s set **col-md-3.** Col-md-4 would not work in this case because that would only create 3 columns.

But on a mobile device, we want 2 kittens to a row. We need 2 columns of 4 kittens each. On a smaller device, each column will take up 50% of the screen. 50% of 12 is 6, so we need to set col-xs-6.

See the following [example](http://getbootstrap.com/css/#grid-example-basic) from the Bootstrap documentation. It shows different ways to divide up the viewport for a device with a “medium” screen, such as a small desktop or a large tablet, and an “extra-small” screen, such as a mobile device.



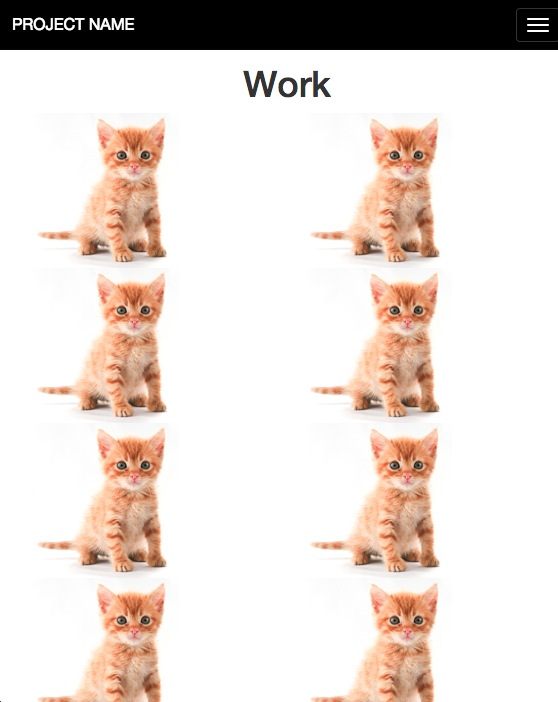
Your code should look like this for each column:

<div class="col-md-3 col-xs-6">

To set up your responsive layout in your HTML file with the grid system:

* Create a new div with a class of container. <div class=”container”></div>
* Create a new div with a class of row. Give your row an ID so it is easier to style later on. <div class=”row” id=”name”></div>
* Create a new column. Grid columns are created by specifying the number of the twelve available columns you wish to span. For example, three equal columns on a medium device would use .col-md-4.
* <div class = “col-size-number”></div>
* Insert your content inside of your new div column. We are using cat placeholder images for now, but you should insert your own examples of work, or boxes with descriptions of your own work. <img src=”img/kitty.jpg”>
* Create space between column content via padding.

Test your site out on your phone. It should look like this (well, without the cats! Please replace these images with screenshots and descriptions of your own work).



If you want to add additional layouts for large (lg) and small (sm) screens, you can insert them with the correct number of column boxes as well.

For example, a div that had the following class would take up all twelve columns -- the full width of the screen -- on an extra-small screen like a mobile one, 6 out of 12 columns, or half the width, on small (tablet) screens, and ⅔ of the screen, or 8 out of 12 columns, on large screens, such as desktops.

<div class="col-xs-12 col-sm-6 col-lg-8">

You've seen responsive column classes in your HTML file before. Now you know what you they mean. Look at your index.html file at the 3 columns in your "About" section. They only have classes for medium (md) screen sizes. Add responsive classes for mobile version as well. What number should go in the blank?

<div class="col-md-4 col-xs-\_\_">

Responsive Images and Text

One of the challenges of mobile phones is that everything is smaller - so how do you make your images and text smaller as your browser size shrinks as well?

It’s an easy fix. Add the Bootstrap responsive image class to your <img> tag.

<img=”yourimage.jpg” class=”img-responsive”>

The following additional CSS properties will ensure images scale down proportionately in smaller devices.

.img-responsive {

display: block;

max-width: 100%;

height: auto;

For text, it's important to continue to use **relative sizing,**percentages instead of pixels, for example. Look at your CSS code and see if you can replace any pixels with percentages.

Responsive vs. Adaptive vs. Mobile-First

RESPONSIVE WEBSITES

Responsive websites have a**fluid layout** AND can detect the size of the user’s screen to actually show them a different view based on the size of their device with **media queries**. Components may get smaller, become stacked (go from horizontal to vertical), or even disappear on smaller devices. For example, you might see a three column layout for desktops, a two column layout for tablets, and a single column layout on smartphones. Buttons and navigation bars may change appearance.

ADAPTIVE WEBSITES

You may have heard about **adaptive design** as well. Adaptive websites aren't necessarily built on a fluid grid. They can have **fixed grids** and introduce **media queries**to target devices.

So why would anyone want to design an adaptive layout over a responsive one? They are generally easier to make and load faster. However, we need to prepare you for the real web development world out there, so in this course, you’ll learn how to design responsive layouts, which allows for the most flexibility.

MOBILE FIRST DESIGNAs versatile responsive design can be, it can often cause problems with mobile phones that may have to load images and styles that appear in the desktop version but not in the mobile version. As a result, professional web developers are now also learning how to design for mobile-first layouts. **Mobile first**means you begin with a simple layout designed your mobile device and then add features as the **viewport,**the size of the browser screen, gets larger, instead of starting with styles for a larger **viewport** and overrriding the styles for mobile. We do this because we don't want someone on a mobile device to have to load a bunch of extra styles that they will not be seeing/using on their device. Bootstrap is designed to be mobile-first.

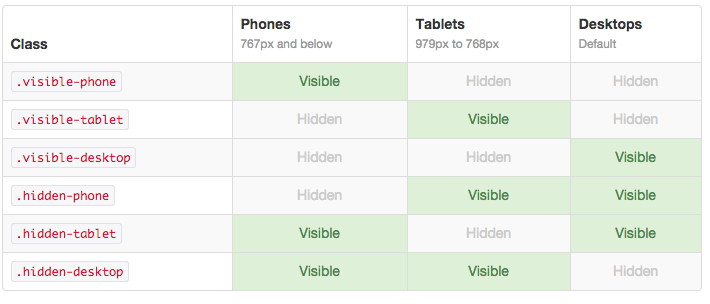
As you dive into the responsive design of your website, consider a mobile first strategy.

1. Determine what is most important content
2. Address mobile devices (size: xs) first with CSS. Then add media queries for tablets, desktops, and other screens.
3. Add elements as screen size increases. Use utility classes to make elements appear or disappear at different screen sizes.

Learn More

UTILITY CLASSES

You can make your assets appear (or disappear) depending on your browser size with **utility classes.** Examine the chart below to determine which elements will appear (or disappear) at various screen sizes. For example, at mobile sizes, navigation bars or large images may be hidden.



[http://careerfoundry.com/images/questionmark.png](http://careerfoundry.com/steps/424#question)

Task

* Use Bootstrap grid system to create columns for small desktop (md) and mobile (xs) screens and edit final layout by modifying padding, margin, and width
* Create a responsive website with at least 2 breakpoints and fluid layout (percentages, relative sizing)
* Add and modify Bootstrap classes to make page elements, such as images, also responsive
* Describe the advantages and disadvantages of designing for mobile-first, and the differences between responsive and adaptive design
* Professional: Create layout for lg and sm screens (up to 4 breakpoints)

## **Single Page Scrolling Navigation**

* Exercise 5 / 5 of Responsive Website: Bootstrap and Plug-Ins

#### Learning Goals

* Transform website navigation to create one long scrolling page with multiple sections (Image Carousel, About, Work1, Work2, Contact, and FAQ).
* Add section IDs to navigation items and padding and margin so that the page lands in the correct place.
* Create scrolling layout by implementing ScrollSpy plug-in with smooth scrolling.
* Add parallax effect with CSS or JQuery (advanced) and read Bootstrap documentation to customize navigation bar, such as adding dropdown menu (advanced)

#### Introduction

A growing trend for many websites is to put all the content on one page. Web designers use scrolling navigation to direct users to website content inside of linking to an external site. To set up single page navigation, you first need to put all your content on a single page. Create a new <div> with the class "container" for each section of your site on your index.html page. Copy and paste the content from your FAQ page and your Contact page into your containers. Adjust margin and padding so that the divs are large enough to fill the viewport.

#### Single Page Navigation

How do you link to a section of the page instead of linking out? Notice that each section has an section ID. These navigation links do not link to an external page (**href**), but to a **target ID**(#thenameofyoursection).

Change the navigation links in the navigation section of your HTML to **target links.** For example, replace about.html with #about. Make sure the div with your about section is named with the ID of “about." Your code snippets should look something like this:

<a href=”#about”>About</a>

<div class=”container” id=”about”>

Continue to do this for your other sections, giving them an ID of “work,” “contact,” “faq,” or whatever you choose to name them, as long as the target ID and the ID of the div are consistent.

#### Scrollspy Navigation

Awesome! Now when you click on a link in your navigation bar, you should see the correct section. But what about the reverse functionality? We want the navigation items automatically highlighted when we reach a different section. This effect is created using Bootstrap’s **Scrollspy**plugin. As you scroll down, Scrollspy shows the section you are on as highlighted in your navigation bar, so you can easily see where you are on a single-page site.

You can see professional examples at W[ndrmnt](http://www.wndrmnt.com/)and [WilliamDady](http://williamdady.com/" \t "_blank). Scroll down on the page without using the navigation links. The links should change states depending on where you are on the website. Use the browser tools to examine how this was done. For another demo, take a look at the Bootstrap documentation on ScrollSpy. You can actually see it in use on all of their documentation pages, where the right hand navigation keeps track of your position as your scroll down the page.

 Add Scrollspy to the top of your navigation bar in the <body> tag.

<body data-spy="scroll" data-target=".navbar">

Make body position relative in your CSS file.

body {

position: relative;

}

#### Responsive Single Scrolling Navigation Bar

What other functionality do these example websites have? Return to<http://williamdady.com/> and <http://www.wndrmnt.com/>. Resize the browser and notice what happens to the horizontal menu bars. At a certain browser size, the menu bar either disappears (williamdady) or collapses (wndrmnt) into a different type of menu. These navigation bars on single scrolling sites are also responsive - a design element we must constantly consider when we implement new features and content to our page.

1) Let’s give our navigation div special Bootstrap nav classes. Underneath the code for your <body> tag, replace the name of your <div> class “nav” with the following classes: **navbar,navbar-inverse,** and **navbar-fixed-top.**For best practices, we are already going to open and close our <divs> for the navigation bar, and comment where our navigation bar will start and end.

<body data-spy="scroll" data-target=".navbar">

*<!--navbar starts here-->*

<div class="navbar navbar-inverse navbar-fixed-top" role="navigation">

…

</div> *<!-- navbar ends here-->*

You already created a fixed navigation bar at the top of the page in a previous lesson. These Bootstrap classes do the same thing, but add some extra functionality.**role=”navigation”** is used to help screen readers navigate the page. Also note the comment. Like your various container divs, this code block is about to get long. Best keep track of which </div> closes it by writing yourself a note.

2) Now, create another div with the class **container-fluid,**indented one level inside your navigation bar to make the navigation bar stretch across the entire page. We are going to comment it by calling it a "nav container." Place your original unordered list items in here.

<div class="navbar navbar-inverse navbar-fixed-top" role="navigation">

<div class="container-fluid">

<ul>

...

</ul>

</div> *<!-- end nav container-->*

3) In order to design a responsive navigation bar, we need to make the original unordered list items collapsible when we resize the browser screen. Use the**“navbar-collapse collapse”**classes for the div. Give the <ul> a class of **“nav navbar-nav.”** Enclose your list items in <ul> and the <div> tags, and close the tags, like in the following example:

*<!-- Collect the nav links, forms, and other content for toggling -->*

<div class="navbar-collapse collapse”>

<ul class="nav navbar-nav">

Your list items here

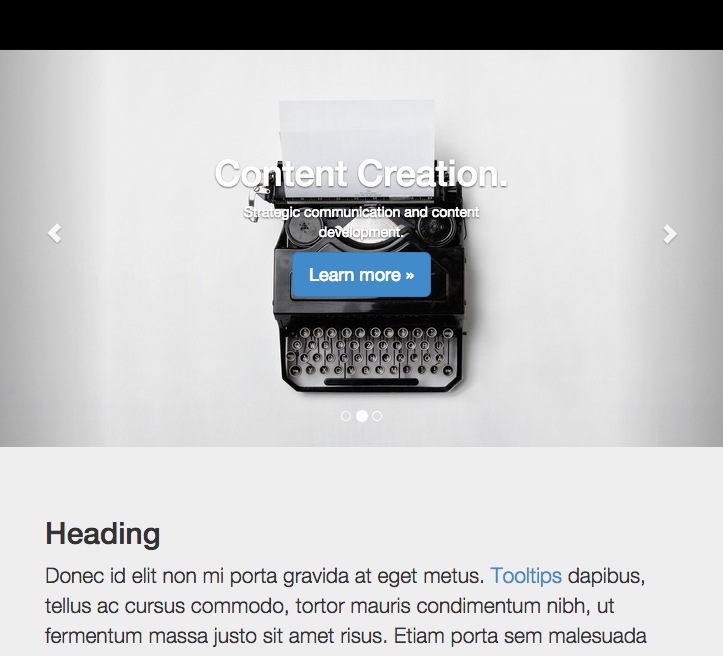
</ul>

</div>

If all goes well, your navigation should look something like this (Note: We've styled our links. You may have styled your links differently).



Hooray! You've just made your navigation bar responsive! But we're not done yet. When you resize your browser window, your list items should disappear when the browser reaches a**breakpoint**defined by a Bootstrap media query. With an image change in the carousel, it might look something like this below.



4) But we don’t want our navigation bar to disappear when we resize the window. We just want another type of navigation bar - a mobile friendly one. To do this, create a navbar with the class of**“navbar header”**ABOVE your div with the navbar class of “collapse.” Indent properly. Each new <div> should be one level INSIDE of the bigger <div>.

<div class="navbar navbar-inverse navbar-fixed-top" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

...

<div class="navbar-collapse collapse">

your unordered list items here

</div> *<!--end navbar collapse-->*

<div> *<!--end navbar container-->*

</div> *<!--end navbar-->*

5) Then, add a button that only appears when the navigation bar is collapsed one level of indentation inside your**navbar-header**. This series of spans bearing the Bootstrap class **icon-bar**creates the common "hamburger" menu -- a stacked-bar icon that expands into a full menu on mobile devices.

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse"

data-target=".navbar-collapse">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

The bootstrap class **sr-only**is for hidden information targeted at screen readers, to allow for accessibility for people with vision problems.

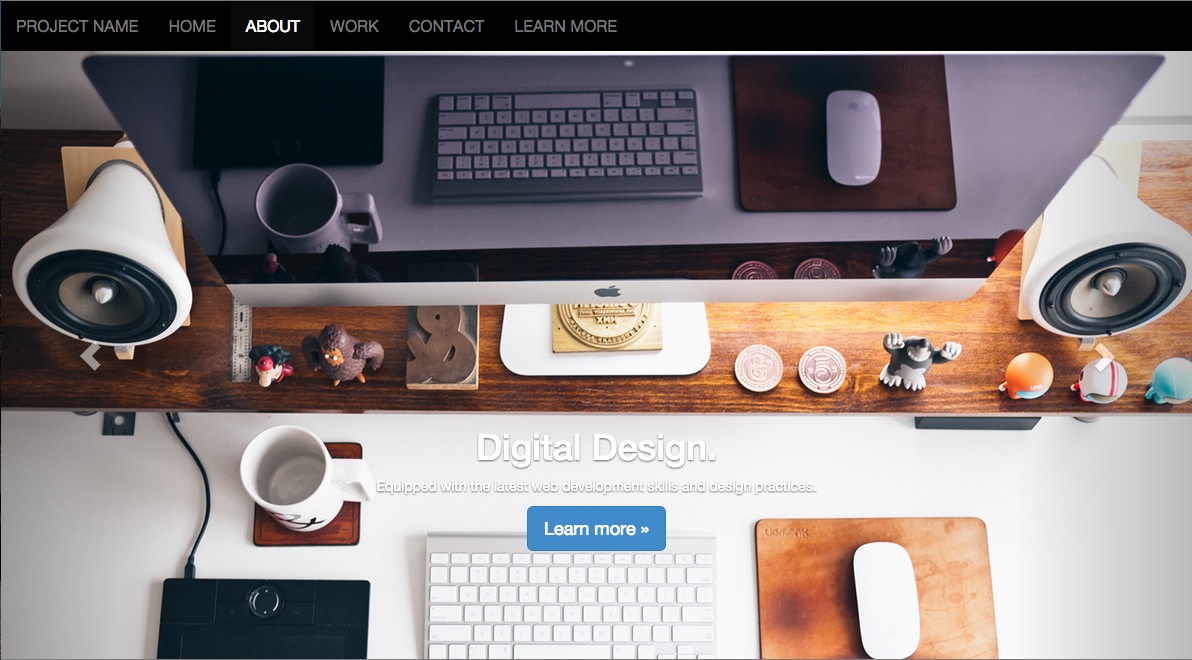
6) If you haven't done also already, you can separate your "Project Name" or "Home" from the rest of your navigation items using Bootstrap and CSS as well. Add another div class for the name of your Project. Replace “projectName” with your portfolio site name, or even your name.

<div class="projectName">

<a href="#">Project name</a>

</div>

</div>

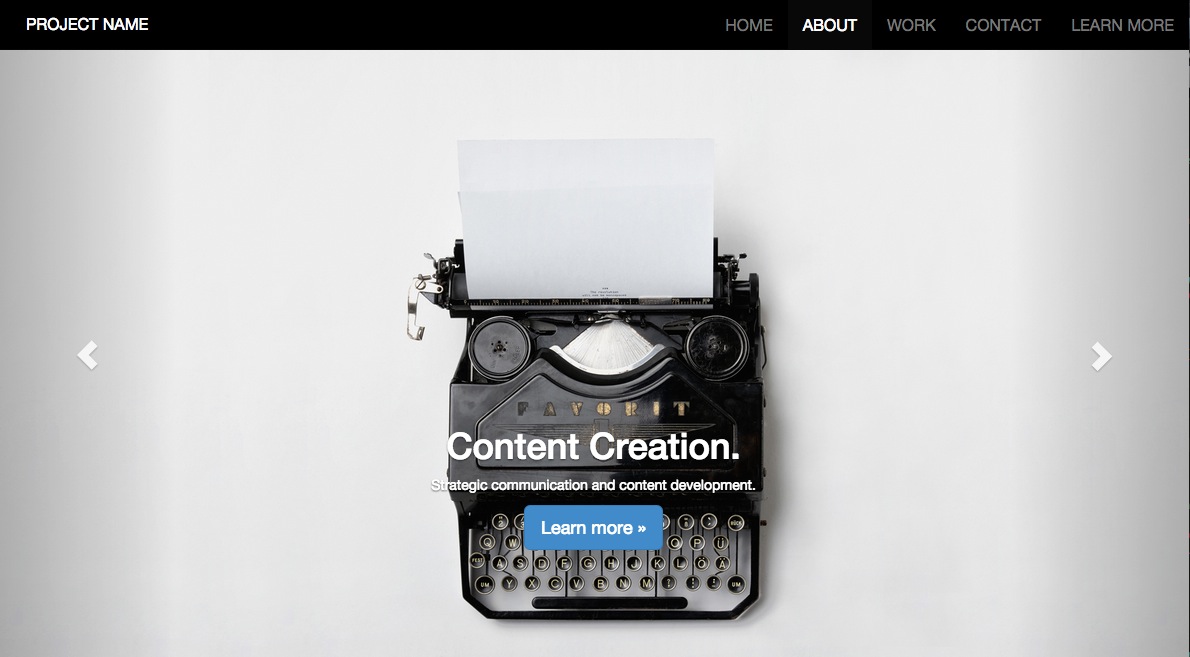


7) Adjust the styles and padding in your CSS file. You may want to float your navigation items to the right, while “Project Name” stays on the left.

.nav {

float: right;

}



8) If you haven't done so already, you can style the div class projectName itself.

.projectName {

text-transform: uppercase;

font-size: 16px;

padding: 13px;

}

We can also modify the color of the link with the nested selector.

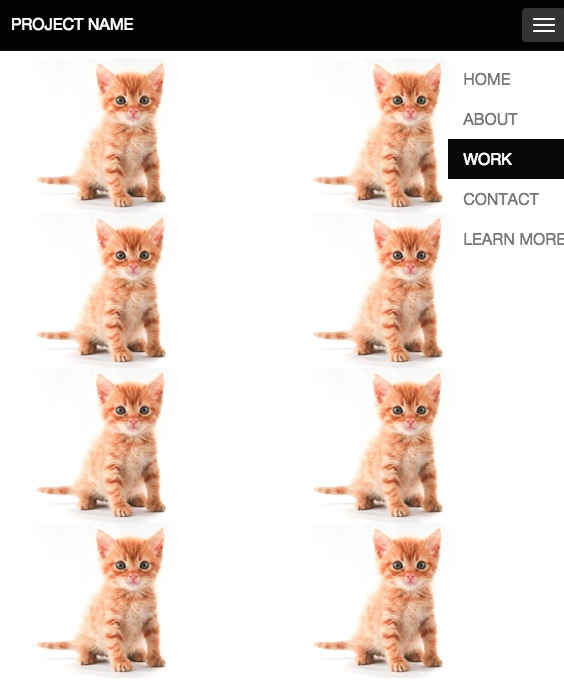
.projectName a {

color: white;

padding: 15px;

}

When you now adjust the size of your browser window, your navigation items will collapse into a Menu button (the three lines in the right hand corner).



You can also replace “Project Name” with a logo or brand image. Bootstrap also has default padding and margins for class “navbar-brand.”

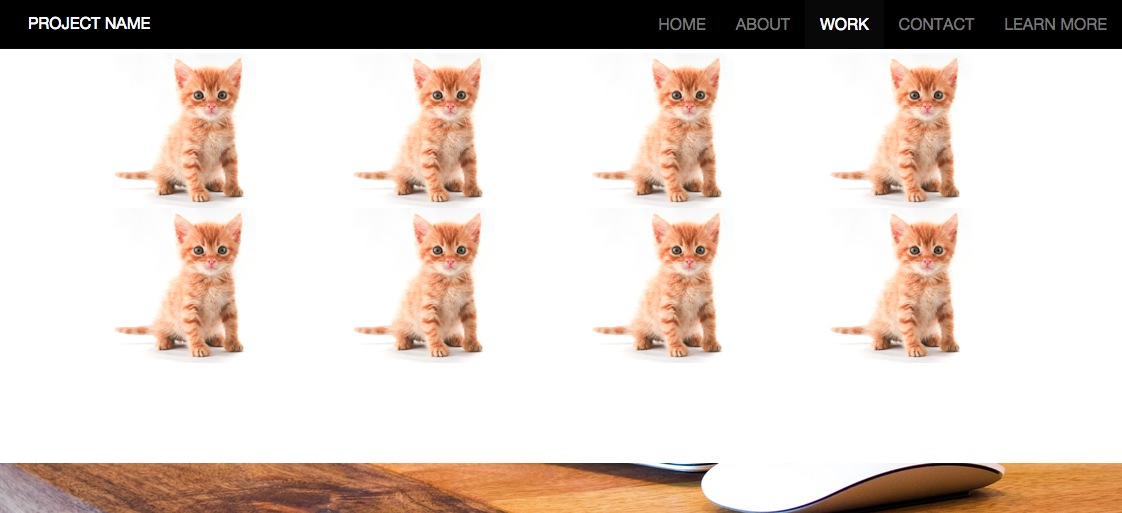
<a class="navbar-brand" href="#">

<img alt="Brand" src="...">

</a>

#### Troubleshooting: Fix the Margin

If you have got the above implementation working, you'll notice that clicking on the anchors will scroll down too far and the fixed navigation will cover up the section headings.



We'll have to figure out a workaround. The most common solution is to add some padding here the size of the navigation bar above the settings. Create a new class called “anchor” and add it to your container classes.

Now, open the Developer Tools on your page. Inspect your navigation bar using the Elements and Styles tab. How many pixels tall is it? That’s how many pixels of padding we need. In my case, I need 50; yours may be different. For example:

<div class="container anchor" id="process">

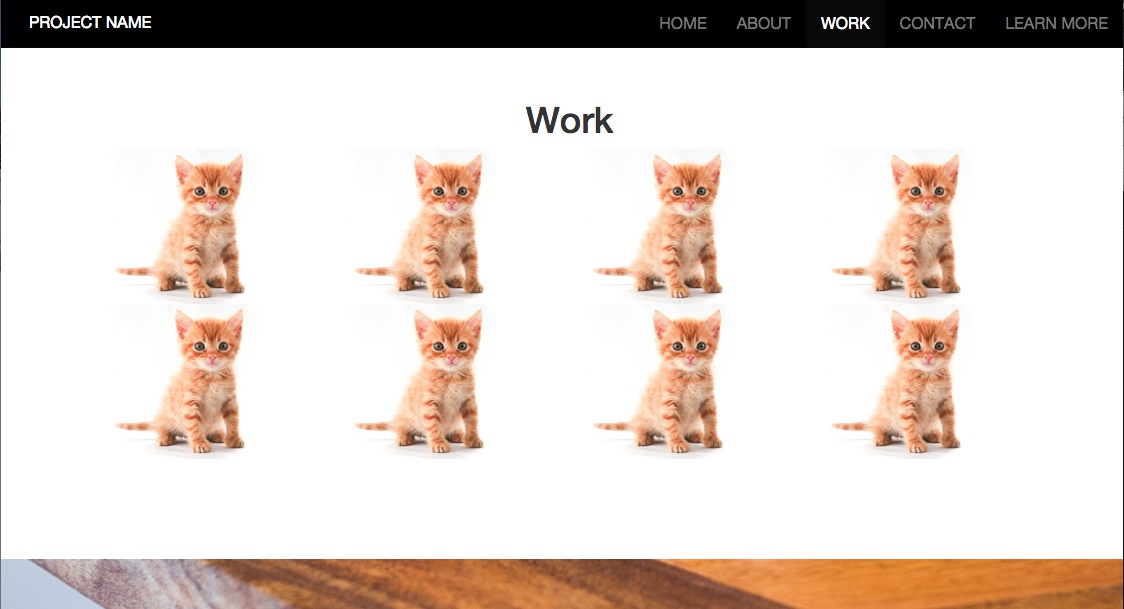
In your CSS, add the necessary padding:

.anchor {

padding-top: 50px;

}

Now your headings should be visible!



#### Smooth Scrolling

Insert the following JavaScript code at the bottom of your page to enable smooth scrolling. We put it there because we don’t want it to run until the whole page has loaded.

<script type="text/javascript">

var $root = $('html, body');

$('.navbar-nav a').click(function() {

var href = $.attr(this, 'href');

$root.animate({

scrollTop: $(href).offset().top

}, 500, function () {

window.location.hash = href;

});

return false;

});

</script>

JavaScript can be inserted into an HTML page inside of the <script></script> tags, or inside of a .js file, and linked within the HTML page (like a CSS file). We won't dive very deep into JavaScript for this week's lessons, but we will cover it more in depth towards the end of the course.

#### Parallax Effect

Websites with **parallax**have background images and website content that scroll at different rates. Check out how it’s done on the [Wndrmnt](http://www.wndrmnt.com/" \t "_blank)site. You can create a simple Parallax effect using only CSS by adding a background image to a div and modifying the properties of the image. The background image will not move when you scroll, giving the effect of layering.

Find another background image by going to [Unsplash](http://www.unsplash.com/" \t "_blank) or a stock image website of your choice, and let’s spruce up our Contact section. If your <div> containing your contact information doesn’t have an ID already, assign it an ID of “contact.” in your HTML page. Then, modify the following in your CSS file:

#contact{

background-image: url("img/yourimagenamehere.jpg");

*/\*Add a background image\*/*

background-attachment: fixed;

*/\*Give the background a fixed position does it not scroll when you scroll\*/*

background-size: cover;

*/\*Have the background cover the entire div section\*/*

color: white;

*/\*Change the color of the text on top so it is readable, and adjust the padding as needed.\*/*

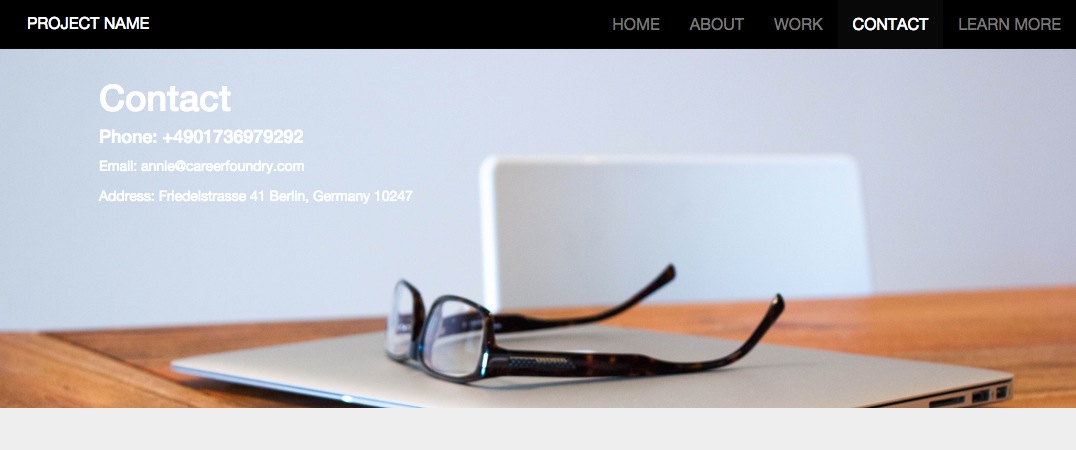
padding:100px;

padding-top: 50px;

height: 400px;

}

Here's what ours looks like with the parallax effect!



Turns out that when you use a background-image, you get a background-attachment property with a default value of 'scroll'! That's why setting it to 'fixed' has this effect. Pretty cool!

#### Learn More

MORE NAVIGATION MENUS

Responsive dropdown menus with Scrollspy effect are also possible in Bootstrap. Visit the Bootstrap documentation for navs at <http://getbootstrap.com/components/#nav> to add a dropdown menu for your “Work” section. Clicking on “Work” should reveal two projects, and clicking on the projects should scroll to each one (i.e. “Work 1” and “Work 2”). Work with your mentor if you have trouble adding these to your page.

 JQUERY PLUGINS

We have created a simple parallax effect with CSS, but if you want more advanced functionality, you can also implement parallax with a **jQuery plugin**. Plugins are pre-packaged bits of code containing JavaScript that are free to use. You install the plugin by downloading a JavaScript file, including it on your page via a **script** tag, and then placing a bit of JavaScript code on your page to run that script on the page or a particular element on the page.

[Stellar.js](http://plugins.jquery.com/stellar/) is a common plugin parallax plugin. The official site - when you click the "[view homepage](http://markdalgleish.com/projects/stellar.js/)" link - has the documentation and a demo. Their [download link](https://github.com/markdalgleish/stellar.js) takes you to the Github repository for the project. Download the **jquery.stellar.min.js**file and put it inside your JS folder, and follow directions on the website to implement the rest.

[http://careerfoundry.com/images/questionmark.png](http://careerfoundry.com/steps/420#question)

#### Task

By the end of this lesson, you should be able to:

* Transform website navigation to create one long scrolling page with multiple sections (Image Carousel, About, Work1, Work2, Contact, and FAQ).
* Add section IDs to navigation items and padding and margin so that the page lands in the correct place.
* Create scrolling layout by implementing ScrollSpy plug-in with smooth scrolling.
* Add parallax effect with CSS or JQuery (advanced) and read Bootstrap documentation to customize navigation bar, such as adding dropdown menu (advanced)

**Achievement Submission - Responsive Website.**

Upload your Gist and send your final single scrolling, responsive website to your mentor. It should demonstrate everything you have learned from Lessons 1-10. See your dashboard for a summary of requirements. Great work!