Bootstrap nav bars: http://getbootstrap.com/components/#navbar

Right, now we're ready to start putting some Bootstrap components into our web page. And the first thing we're gonna do is deal with this question of where can I go? So typically the answer to where can I go is provided with a navigation bar. So you present the user with some options as to where they can go next. And bootstrap has really quite extensive functionality for dealing with navigation bars. And we're gonna cover the basic nav bar in this lesson. So let's go and look at the code to make a basic navbar. I'm gonna begin by just building the framework for a navbar, and then we're gonna add the classes, which turn it into a Bootstrap navbar. Okay, so what am I writing here? Hopefully, you're following what all these tags are. You should have seen all of these tags before except for maybe the nav tag.

```
<!DOCTYPE html>
<html lang="en">
  <head>
   <title></title>

    <link href="css/bootstrap.css" rel="stylesheet">

     <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

  </head>
  <body>

  <nav>
        <div>
                <ul>
                        <li><a href="#"></a></li>
                        <li><a href="#"></a></li>
                </ul>
        </div>
  </nav>

  <div class="container">
       <h1> Hello there </h1>
  </div>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>

  </body>
</html>
```

So, let's step through that and explain it a bit. So I've wrapped everything for my navigation bar inside a nav tag. And nav tags are important because they increase accessibility of your site. When a screen reader which, say someone who doesn't have good vision is visiting your site and they're using some sort of text based way of interacting with the site and they can't see the graphics, then the nav bar tells the reader that they're using, okay, everything inside this tag relates to navigation. So that's why we use a nav bar, for that semantic meaning of our tags. Inside the nav, we have a div tag. So it's quite common in Bootstrap to use div tags to wrap things up, so that you can set some different functionality. Especially responsive type of functionality for different things. So we'll see how we add some special classes to the div to control how it works, later. Inside the div, we've got a ul tag. Can you remember what ul is for? Unordered list, right? And so think about what we're communicating with this structure here. We're saying, okay here's a navigation bar, we can ignore the div for now. So here's a navigation bar, and inside the navigation bar, here's a list of related things. So, uls and ols are used to collect together things that are related, right? So lists of things which are in the same area, or something like that.

So in this case, we've got a list of things, and each of the things on the list, is gonna be a link which the user can click on to go somewhere else. So, we're collecting together all the main lists for the site. And inside the ul, we've got list elements, and each list element then has a list inside it. So that for example might be the home link, and then this might be My photos. Now we're gonna build up a sort of example here, which is a set of photos, which the user wants to sort of share with the world, okay. And we'll continue building that example throughout the specialization. Okay, so we've got my photos there and home. Let's just have a look at how this looks in the browser, right. So we got the basic sort of formatting there. And it kinda looks like a nav bar, but it doesn't look very nice. So how do we turn it into a Bootstrap nav bar?

Well, I'm gonna just grab some boilerplate kind of class definitions here. So this word, boilerplate, really refers to the idea that you've got some ready-made code and you're just kind of changing bits of it. So rather than writing it from scratch, you can just drag bits in.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
   <title></title>

    <link href="css/bootstrap.css" rel="stylesheet">

     <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

  </head>
  <body>

  <nav class="navbar navbar-default">
        <div class="container-fluid">
                <ul class="nav navbar-nav">
                        <li class="active"><a href="#">Home</a></li>
                        <li><a href="#">My photos</a></li>
                </ul>
        </div>
  </nav>

  <div class="container">
       <h1> Hello welcome to my bootstrap site </h1>
  </div>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>

  </body>
</html>
```

1. So I've just grabbed those two classes. So that sets us up in that bar. Okay. So when I reload that, it's now added this gray background. So it's separating the navigation bar from the main content of the site. That's the first indication to the user that oh, this is something different from the rest of the page.

2. Then, the next thing I'm gonna do is set the container to be a **fluid container**. And that's a good type of container for a navigation bar. Again, it's a class provided by bootstrap, the sort of properties of that class are dictate how it flows when you crush the page up or down. And fluid is the best one for navigation bar because it kind of holds the navigation bar in the way that you want. We'll see more about that in a minute.

3. And then, I'm gonna add some classes to the list as well, and classes are here. So, what have we got? We added the class nav bar and nav bar nav. Okay, we save that. And when I reload, you see now I've got something that looks a bit more like an app, I've got sort of horizontal links. So what it's done is it's taken that list and lists are only displayed vertically going down, but it's kind of removed that vertical layout and squashed it out across.

4. And that's really my basic nav bar. i'm just going to add one more thing, which is to tell the user where they are. So, we'll be looking more at where am I later, but let's just do a quick touch on the nav bar to just tell the user where they are. So we simply go into the list and set the class to active for the one. So in this case we're saying right, the user is on the homepage therefore, we highlight the home link, okay. And the idea is as they move around the site, you kind of highlight the bit of the site that they're in to sort of tell them where they are.

And you just saw then, if you're watching the screen, if I squeeze in the nav bar, it kind of behaves responsively. Okay so, if the page is not wide enough, it drops it down into a vertical nav rather than a horizontal one. And that's kind of one of the concepts of responsive where your navigation, you should really make sure that your navigation always works on any device. Cuz there's nothing more annoying than going to a site on your phone and find that you can't navigate around. And that's kind of one of the things that responsive design is trying to solve. The idea that you use the same code but it basically responds to whatever the devices they are using. And in the next lesson, we're gonna see how to make a more advanced version of a responsive navigation bar. But for now, there we have it, a basic Bootstrap navigation bar.