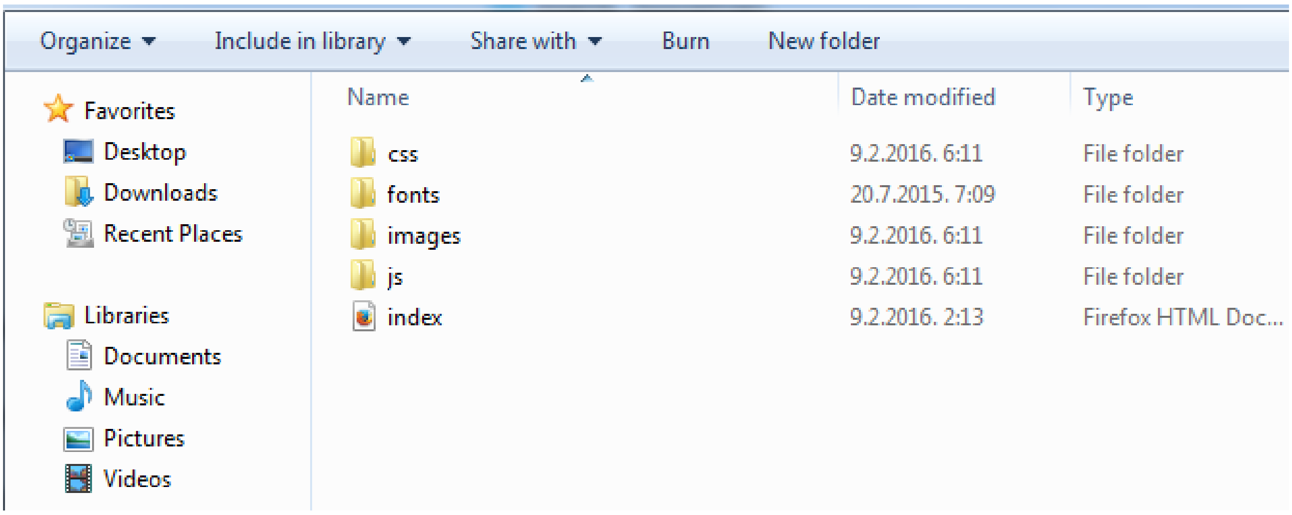
Welcome back. In this lesson, we're going to look at how we can sort of start coding using the bootstrap library, and so we're going to start from right at the beginning with the most basic set of files that we need to build a bootstrap website. So let's have a look at my file system to see what I've got here. So I've downloaded the blank bootstrap example code and unpacked it, and it's given me this set of folders, so we've got CSS, fonts, images, and a JS folder and then an index file. So just to quickly run through those and see what they're for.



Name	Date modified	Type
css	9.2.2016. 6:11	File folder
fonts	20.7.2015. 7:09	File folder
images	9.2.2016. 6:11	File folder
js	9.2.2016. 6:11	File folder
index	9.2.2016. 2:13	Firefox HTML Doc...

The CSS folder contains the most important thing, bootstrap CSS, which is the bootstrap live, that gives us all those components that we're going to learn about as we work through this module. The fonts slide folder contains a set of fonts which provide some of the icons that come with bootstrap. And then images folder's got a whole set of images in there, which allow us to sort of play around and build image-based websites. We've got the index file, which is a simple HTML file showing how to use the basic Bootstrap stuff. Then, finally, we've got this JS folder, which has a bunch of JavaScript libraries, which allow Bootstrap to function.

> So we've got jQuery, which is the basic library that you've been working with in the JavaScript part of this course, and then bootstrap.min and then bootstrap, which kind of sit on top of the jQuery library, and give some of the extra functionality that can't be done with CSS on its own.

So, there's a little bit extra JavaScript which boot strap provides. And the mend version of is, if you look at it, it's just a completely unreadable squashed down version of the file. Where as the bootstrap, plain one, is the same file, but not squashed. So this idea of sort of minification you'll come across with websites. And the basic idea is when someone hits your website, you don't want to have to download too much data before your site displays, because it means that your site will be slow. So minification is a way of taking all the different files, like the CSS and the JavaScript files especially, and squashing them down so that they take up a minimum possible amount of data, so that your page loads as quickly as possible. So we'll be using the un-minified versions, because it's easier to kind of debug and stuff. But then when you, if you launch a website, you have to make sure that everything is minified for maximum performance.

Right, so what we're going to look at now is the index file, which will show us the HTML that we need to just get bootstrap up and running. So let's load it up in the text editor. Right, so that's the whole file there, and so, let's just work through and make sure we understand, because maybe that's a new element. That's a little extra thing that we put into the document to make absolutely clear what type of document it is, that it's an html document. For example it might, other types of document, xhtml, which is a slightly more stricter version of html, if you like.

```
<!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>

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

  </body>
</html>
```

And then we've got an html tag with a language attribute implying, yes, this is an html page written with the English language. Then we've got the head element with a title, and then it kind of gets interesting. So the first thing we notice is we've got this link href, so that is the stylesheet. So we saw that file earlier, bootstrap.css file, and so that's essentially all the CSS that bootstrap provide. And you notice that it's prefixed with .css, because it's not just in the same folder as the index file, if we look at the folder hierarchy again. You'll see that Bootstrap is actually inside that CSS folder, so that prefixing of a css tells the browser, go into the CSS folder, and load this file, okay? So that is pulling in the main Bootstrap CSS library.

Then we got some weird, sort of slightly funky looking stuff right here, and don't worry to much about that, but essentially what it does is it takes an old version of internet explorer, which is another web browser like Chrome or Firefox, and it kind of pimps it up a bit so it actually works like more modern web browsers. So because quite a few people still use internet explore eight. So that is missing certain functionality which you, you would ideally want on your web browser so that, that stuff that you see in the head there is just adding a little bit of extra functionality to Internet Explorer 8 if someone's using it. But you know, it's gradually dying out, but we still have to have backwards compatibility. I mean that's really an interesting thing about the internet is that you never quite know what someone is going to be connecting to your website with. They might be using some ancient version of a web browser that doesn't quite display it properly and all this stuff, and I mentioned earlier the concept of graceful degradation in the previous course, and graceful degradation being this idea of making sure your website doesn't just completely fail if someone has a slightly deficient web browser that doesn't have the full functionality. So this code here is an example of dealing with that by saying, "well, somebody might be using the site with an old web browser". So what we do is add a few extra components that make that web browser good enough to see the site.

Okay, so let's move on. In the body tag, we simply have two script tags, so one is pulling in the jQuery library there, and the other one is pulling in the Bootstrap library. And, as I said earlier, those allow for some extra little bits of nice functionality, which we'll see, especially with navigation bars, and models, and so on, that allow bootstrap to give a better user experience.

Right, so that is the basic starter point. So, let's just see how that looks in the browser. You might want to guess how it's going to look. Probably not that exciting, because we kind of haven't put any content in yet. Right, so we see a completely blank page. Unsurprisingly.

So, normally in a Bootstrap site, you'd start off by just putting everything in a container class

So we go into the body tag and create a div. And set the class to container, because that's the standard class for wrapping your content. And then we might put a heading, and say hello, welcome to my bootstrap site, and you save that, reload, okay, so that's if you like, the simply as possible bootstrap site.

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

And so what we've seen there is just a quick run through of all the bits and bobs that you need to start building these bootstrap sites. And as we work through these lessons, we're going to see how we can add more and more of those interesting components to deliver the use experience that case introduced to you in the previous module.

> Why does the basic Bootstrap site include jQuery before the Bootstrap.js library? The Bootstrap.js library depends on features in jQuery so that it can provide interactive web design elements.