What is Bootstrap and How Do I Use It?

We learned a little about Bootstrap this week in Lectures. According to the official website, **Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.** Now how do I use it?

It would be easy to send you over to their Getting Started page and call it a day. Their setup guide is indeed a host of useful information –but it is not a step by step guide to getting started

When I discovered Bootstrap, responsive design was gaining in popularity. Having only ever made websites from scratch, I needed to learn about the entire concept of a framework. I'd imagine it's even more confusing for beginners who are now expected to learn responsive design concepts and Bootstrap and JavaScript libraries, in addition to HTML, CSS and JS.

This guide is meant as a first look into Bootstrap for beginners, so won't be going into LESS and Sass integration, which are more intermediate/advanced concepts.

Goals

- Learn what a front-end framework is and how it can be useful
- Understand how to properly include Bootstrap's CSS and JavaScript and begin customizing

Prerequities

- Basic knowledge and understanding of HTML and CSS

What is Bootstrap?

Bootstrap can be boiled down to three main files:

- bootstrap.css – a CSS framework
- bootstrap.js – a JavaScript/jQuery framework
- glyphicons – a font (an icon font set)

Additionally, Bootstrap requires jQuery to function. jQuery is an extremely popular and widely used JavaScript library, that both simplifies and adds cross browser compatibility to JavaScript.

Everything else you might happen across while studying the Bootstrap documentation – Grunt, Gulp, Sass, LESS, bower, npm, etc – is not necessary to get started with Bootstrap. These are task runners, preprocessors, installation aids, and package managers, so don't be discouraged if you don't know how to use any of them yet.

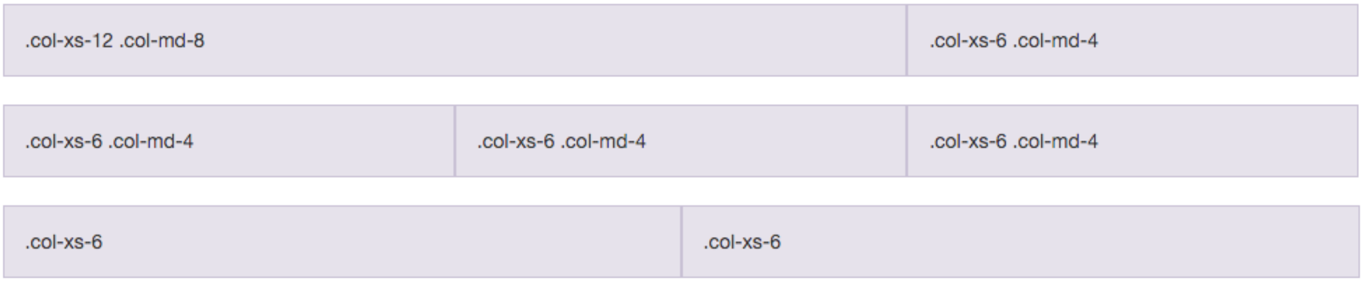
Why is a framework important? Do I need to use one?

Some of the ways that frameworks can help you:

- Prevent repetition between projects
- Utilize responsive design to allow your website to adapt to various screen sizes – mobile, desktop, and everything in between
- Add consistency to design and code between projects and between developers
- Quickly and easily prototype new designs
- Ensure cross-browser compatibility

Generally, every web project you work on will need to be responsive and work properly on all the major browsers, and likely have some fallbacks for older browsers. Bootstrap has a huge open source community that works on covering this so you don't have to. Additionally, when multiple developers all know the same system, they can work in better harmony – and it also makes it easier for newcomers on a project to get up to speed.

The grid is probably one of the most essential aspects of the framework. It's the basis on which the entire layout is created. Beyond that, Bootstrap's core CSS will also add helpful styling to forms, tables, buttons, lists, and images, as well as fully functioning navigation bars, while the core JavaScript will add helpful code for creating modals, carousels, alerts, popups, dropdowns, and accordions.



.col-xs-12 .col-md-8 | .col-xs-6 .col-md-4

.col-xs-6 .col-md-4 | .col-xs-6 .col-md-4 | .col-xs-6 .col-md-4

.col-xs-6 | .col-xs-6

Let's begin!

Building a Basic Template with Bootstrap

Bootstrap comes with a few very simple examples to start from, but it's just as easy to start from "scratch", so that's what we'll do. First, we use only Bootstrap to lay out the foundation, then we'll add our own custom style on top to make something fun and trendy.

Step one is to download Bootstrap direct from their site, or you will also find this on Blackboard – Labs – Week 13. The zip file will come with css, fonts, and js directories. Unzip that and save the files in some directory.

As time goes on you will need to download Bootstrap files direct from the website as they are constantly being updated.

Now create yourself a new folder to organise this week's Lab, copy the 3 folders into here.

css
fonts
js

Bootstrap doesn’t come with any HTML, we will create a “Hello, World!” page to start on the documentation, so we’ll save that as index.html into our new folder.

Use the code below (you might need to turn off the numbered list in Word) or use the code supplied on the Website

Hello, World!

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.    <head>
4.      <meta charset="utf-8">
5.      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.      <meta name="viewport" content="width=device-width, initial-scale=1">
7.      <!--The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
8.      <title>Bootstrap Lab</title>
9.      <link href="css/bootstrap.min.css" rel="stylesheet">
10.     <!--[if lt IE 9]>
11.       <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
12.       <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
13.     <![endif]-->
14.   </head>
15.   <body>
16.     <h1>Hello, world!</h1>
17.
18.
19.      <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
20.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
21.     <script src="js/bootstrap.min.js"></script>
22.   </body>
23. </html>
```

Easy enough to start. We have our basic doctype, html, head and body tags.

The meta name="viewport" tag is particularly important for responsive design – it ensures that your website has a 1:1 ratio with the viewport (screen size). We used this in Chapter 7.

Beyond that, we’re just adding Bootstrap core CSS in the head…see line 9

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

You could have a look at this CSS file in the folder….it looks like only a few lines. If you are in Visual Studio you could select it all, then right click and Format Document. Now you will see code you are more familiar with…..~7000 lines of code! And now that you have worked with CSS you will recognise the code being used here.

Now look for the jQuery via Google CDN (**Content Delivery Network**) before the closing body tag…see line 20

```
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```

and Bootstrap core JavaScript. …see line 21

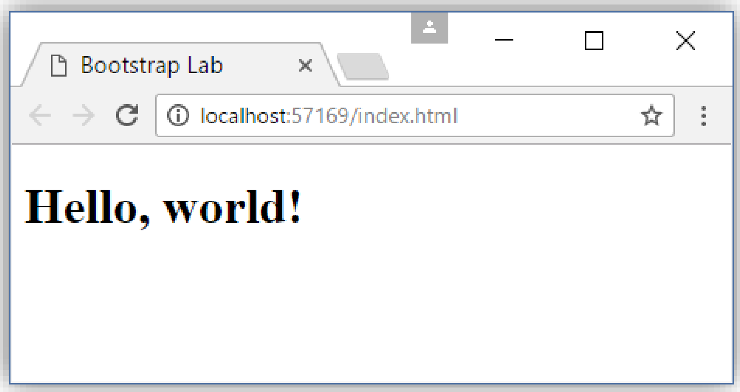
```
<script src="js/bootstrap.min.js"></script>
```

You could take a look at this file as well, it wont make as much sense to you until you have explored more about JavaScript – but that’s ok, this is the beauty of bootstrap – you don’t really have to make sense of this code to make things work!

NB: Bootstrap JavaScript and custom JavaScript *must* go below jQuery to function!

Additionally, we are linking to jQuery via Google's URL because it reduces load on our live server, but you can download it if you want to work locally.

Well, that's all you need to get started with Bootstrap! Let's check out our new site.



Hello, world!

…great!

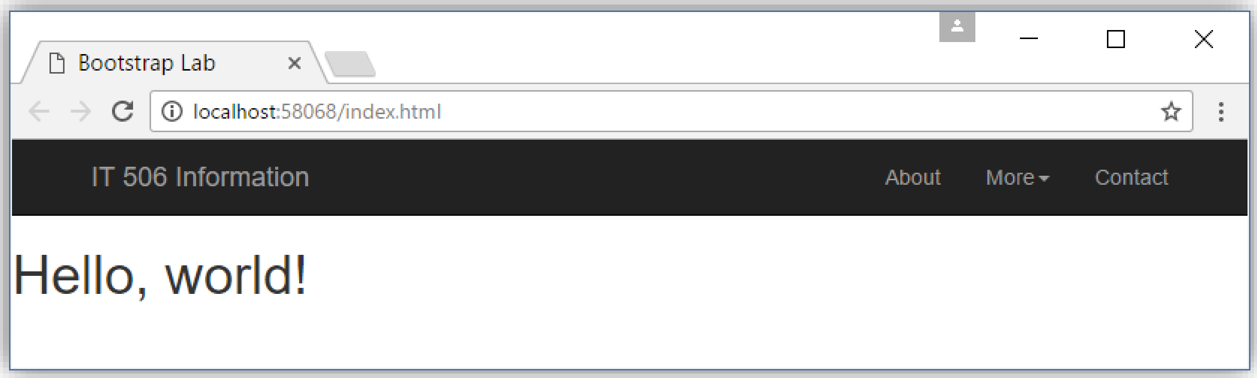
Navigation Bar

Although we have nothing happening yet, in no time at all we can get copying and pasting from the docs and have a nice, functioning website.

First and foremost, we add in the classic Bootstrap version of a top navigation bar. Here is a simplified version of their navbar example. Place this code right below your opening body tag.

Remember to turn off the numbering before you copy this.

```
<nav class="navbar navbar-inverse navbar-static-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">IT 506 Web Design</a>
        </div>
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="#">About</a></li>
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">More<span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="#">Timetable</a></li>
                        <li><a href="#">Topics</a></li>
                        <li><a href="#">Learning Outcomes</a></li>
                        <li><a href="#">Where to Next</a></li>
                    </ul>
                </li>
                <li><a href="#">Contact</a></li>
            </ul>
        </div>
    </div>
</nav>
```



Hello, world!

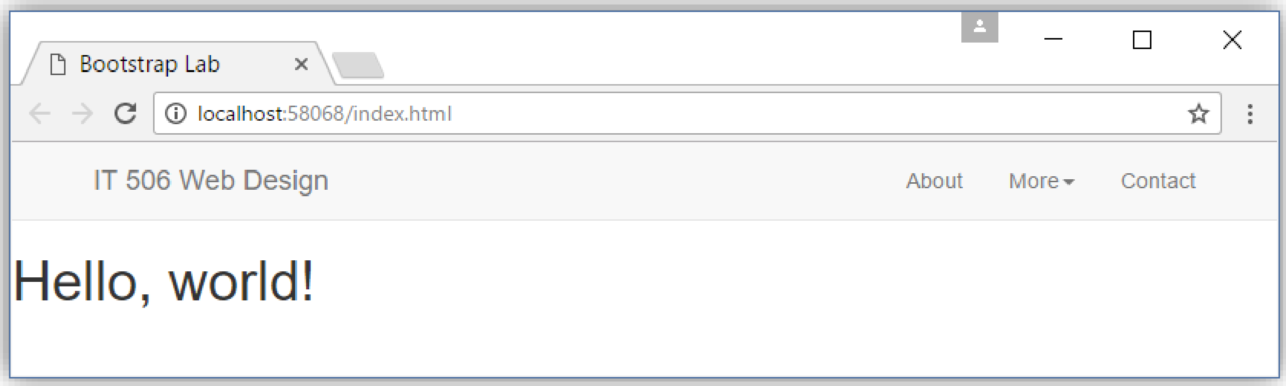
It seems like a lot of code, but it's not so complicated when you pick out the lines.

In the first line, we are using the <nav> element to define the bootstrap classes that will control the entire bar.

There is a class called navbar, then we are choosing a dark color scheme with navbar-inverse, and electing to use navbar-static-top, as opposed to a fixed (sticky) header(more on this later)

```
<nav class="navbar navbar-inverse navbar-static-top">
```

Go ahead and change the colour of the navbar to navbar-default , save and view the change.



Hello, world!

Line 2 is a container that sets a max-width on the content within your full-width navbar. Without this the navbar would stretch right to the edges of your page…just like the Hello World

```
<div class="container">
```

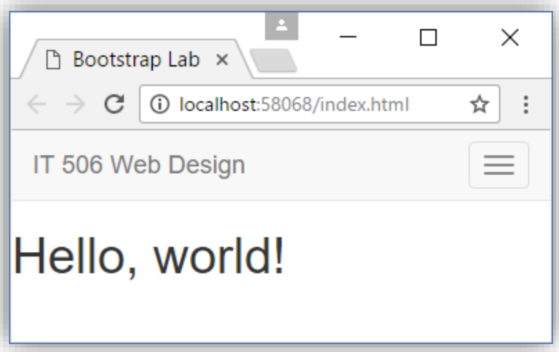
Now looking at lines 3-11

```
<div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">IT 506 Web Design</a>
</div>
```

The navbar-header class contains the “brand” information, where you can put your logo or company name. We’re making a website to advertise this course **IT506 Web Design**.

The button is hidden on desktop, and becomes a dropdown hamburger menu on mobile (each <span class="icon-bar"> is a line in the hamburger).

Like this when the browser is re-sized:-

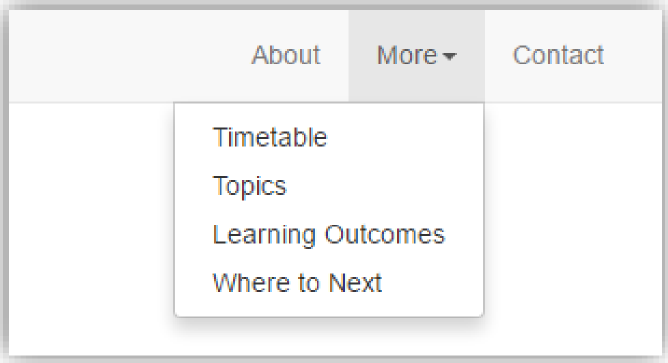


Hello, world!

Try it…

The rest of the code from lines 13-25 is a right-aligned unordered list that serves as both our desktop and mobile menu.

Can you see the nested unordered list within the More list element beginning on Line 16 ? This is how the dropdown is created.



You will see many # characters throughout this nav code, eventually as you create actual pages you will replace these with the name of your newly created page.

Jumbotron Header

Ok, now for one of those big, space-wasting, attention-grabbing headers, which is called a jumbotron in Bootstrap terms. Not much to see here, just a bootstrap jumbotron with a container and some copy.

Delete our Hello World h1 and replace with this code.

```
<div class="jumbotron">
       <div class="container">
           <h1>Ready. Set. Code.</h1>
           <p>Are you ready to learn how to build a website?
              We can set you on the road to learn the basics, so what are you waiting for?
           </p>
           <br>
           <p>
               <a class="btn btn-primary btn-lg" href="#" role="button">Download Our Enrolment Form</a>
               <a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a>
           </p>
       </div>
   </div>
```

There's some extra space we don't want above the jumbotron, but I want to see how far Bootstrap can get us without overriding styles. As you can see, we already have a pretty nice, adaptable layout without having written a single line of CSS.

This is the first time we have used a bootstrap button. Lets try some customisation before we continue.

The button colour is being controlled by a class called btn-primary and its size by btn-lg. Try these options to see the differences.

btn-default

btn-primary

btn-success

btn-info

btn-warning

btn-danger

We use these "colours" with a lot of the standard bootstrap items

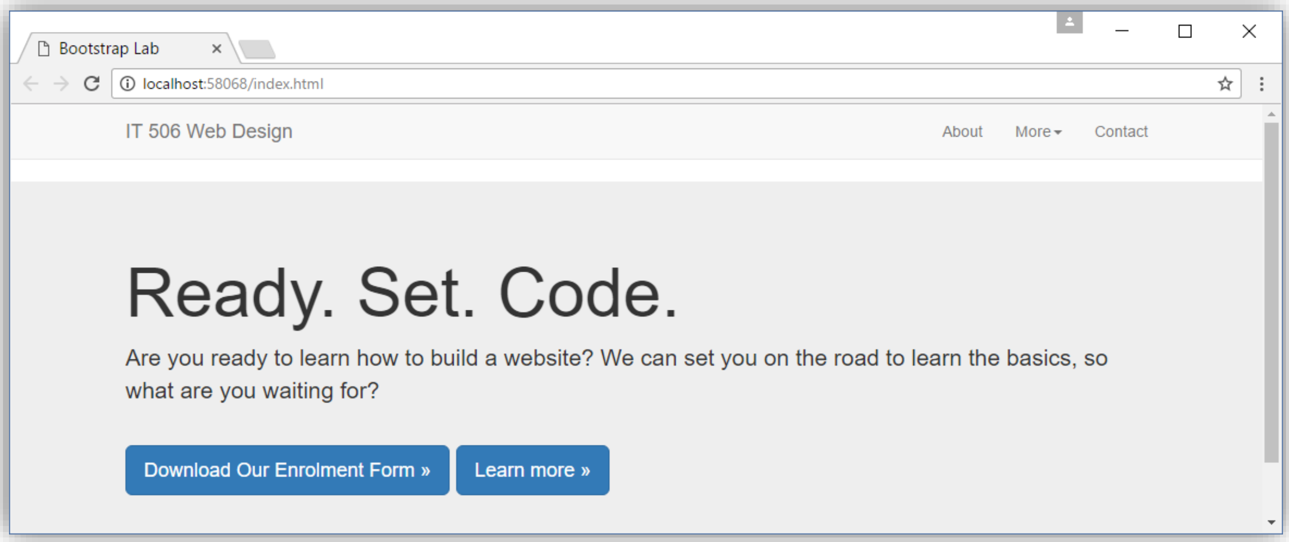
Try these variations on sizes as well.

btn-lg

btn-md

btn-sm

btn-xs



Ready. Set. Code.

Are you ready to learn how to build a website? We can set you on the road to learn the basics, so what are you waiting for?

Grid

The next thing we will do is add in some main content, which will be in the form of a grid.

Copy and paste the code below so that is directly under the jumbotron.

```html
<div class="container">
    <div class="row">
        <div class="col-md-4">
            <span class="glyphicon glyphicon-check" aria-hidden="true"></span>
            <h3>HTML5</h3>
            <p>Donec id elit non mi porta gravida at eget metus.
               Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.
            </p>
        </div>
        <div class="col-md-4">
            <span class="glyphicon glyphicon-edit" aria-hidden="true"></span>
            <h3>CSS3</h3>
            <p>Etiam porta sem malesuada magna mollis euismod.
               Donec sed odio dui. Lorem ipsum dolor.
            </p>
        </div>
        <div class="col-md-4">
            <span class="glyphicon glyphicon-cog" aria-hidden="true"></span>
            <h3>JavaScript</h3>
            <p>Vestibulum id ligula porta felis euismod semper.
               Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.
            </p>
        </div>
    </div>
</div>
```

This code will seem more familiar – you know h3 and p elements. There is some placeholder text used.

Now bootstrap is managing them into 3 columns using bootstrap class="col-md-4"

Think of it like this- Grids have rows…lines 2 and 24

```html
<div> class="row">

</div>
```

that contain columns.

```html
<div class="row">
  <div class="col-md-6"></div>
  <div class="col-md-6"></div>
</div>
```

Bootstrap works on a 12-column system, so as long as you add up to 12, you're on the right track.

The above example will contain two 50% width columns (6/12), which will stack under each other on mobile and become 100% width.

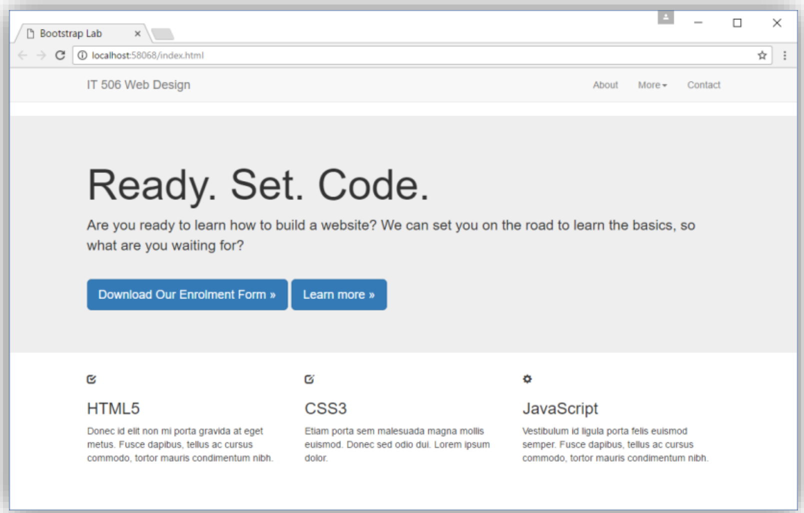
Take a look at the code we pasted – you can see three columns which are 4/12, and as 3 x 4 = 12, everything works out.

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

<table>
<tr><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td></tr>
<tr><td colspan="4">span 4</td><td colspan="4">span 4</td><td colspan="4">span 4</td></tr>
<tr><td colspan="4">span 4</td><td colspan="8">span 8</td></tr>
<tr><td colspan="6">span 6</td><td colspan="6">span 6</td></tr>
<tr><td colspan="12">span 12</td></tr>
</table>

Bootstrap's grid system is responsive, and the columns will re-arrange automatically depending on the screen size.



Ready. Set. Code.

Are you ready to learn how to build a website? We can set you on the road to learn the basics, so what are you waiting for?

HTML5

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

CSS3

Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. Lorem ipsum dolor.

JavaScript

Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

Icons

Take a look at lines 4, 11 and 18 there have been some bootstrap glyphicons added for decoration. **Glyphicons** are the built-in icon set that comes with Bootstrap. If you don’t load your fonts folder, or if you move them to a different directory, these icons won’t work.

```
<span class="glyphicon glyphicon-check" aria-hidden="true"></span>
```

Using a glyphicon will always be that same code, and only the glyphicon-**check** class will change.

Go ahead and change the glyphicon on display to something of your own choice.

Backspace the word **-check** and you should get a selection list.

Adding Custom Styles to Bootstrap

Not bad for getting this far without touching a line of style. It’s professional, responsive, and browser friendly.

It’s definitely not creative or unique, though. After laying down a foundation, you’ll want to add your own personal design.

We happen to be using “vanilla CSS”, which is CSS without a preprocessor. Fortunately, you can just add an additional stylesheet **below** Bootstrap’s core

Don’t modify the Bootstrap core – you’re much better off overriding the existing styles.

Hands off the BOOTSTRAP code!! Otherwise you will run into issues later on.

Find the line `<link href="css/bootstrap.min.css" rel="stylesheet">` and directly under this add these two lines below.

```
<link href="css/custom.css" rel="stylesheet">
<link href='https://fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>
```

Now go ahead and create your new CSS file and save it into the css folder. This is a familiar process for us now.

We have added a link to a Google font called Montserrat. If you’ve never used a Google Font before, all you have to do is add the font stylesheet to your head and change the font-family of your desired element.

You should explore Google Fonts API – lots of great tutorial information for using this. And for a complete list of Google Fonts visit https://fonts.google.com

Now we finally begin styling. In just a few minutes, you will be able to transform the boring, generic Bootstrap layout into a fun, flat layout. First, we are going to get rid of that pesky space between the navbar and the jumbotron.

Add this to your newly created CSS file, check each stage of the following changes to see the display transform.

```
.navbar {
	margin-bottom: 0;
}
```

You are already familiar with the “class needs a dot” rule and bootstrap code requires the “dot” in CSS when referring to these classes.

Add few simple styles to the main tags. We are loading in the Montserrat font on the entire page, making the headers bold, and the background dark and the text light.

```
body {
	background: #3E4649;
	color: #f7f7f7;
	font-family: 'Montserrat', sans-serif;
}
h1,h2 {
	font-weight: bold;
}
p {
	font-size: 16px;
	color: #cdcdcd;

}
```

We are going to make the jumbotron green and centered.

```
.jumbotron {
	background: #27A967;
	color: white;
	text-align: center;
}
.jumbotron p {
	color: white;
	font-size: 26px;

}
```

We are going to turn the buttons into “ghost buttons”, which are buttons that are transparent with a border. Also adding a margin so they’ll stack properly on mobile.

```
.btn-primary {
	color: #fff;
	background-color: transparent;
	border-color: white;
	margin-bottom: 5px;
}
.btn-primary:hover {
	color: #27A967;
	background-color: white;
	border-color: white;

}
```

Next make the navbar a different shade of dark, make the links lighter, and change the background color on hover.

```
.navbar-inverse {
	background: #2E2F31;
	border: 0;
}
.navbar-inverse .navbar-nav li a {
	color: #f7f7f7;
	font-size: 16px;
}
.navbar-inverse .navbar-nav li a:hover {
	background: #27A967;

}
```

The dropdown menu has its own classes, so we are also going to change the background color on these and add a little padding.

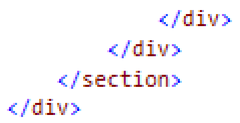
```
.dropdown-menu {
	background: #2E2F31;
	border-radius: 0;
	border: 0;
}
.dropdown-menu li a {
	padding: 10px;
}
.navbar-inverse .navbar-nav .dropdown-menu li a:hover {
	background: #2C463C;
}
```

Back in the HTML, we are going to wrap a section tag around the grid, and call it messages. The section opening tag goes directly below the div class="container"

```
	<section class="messages">
	     <!-- .rows and .columns -->

	</section>
```

Make sure you get the closing section tag in the right place so that the nesting is correct



```
            </div>
        </div>
    </section>
</div>
```

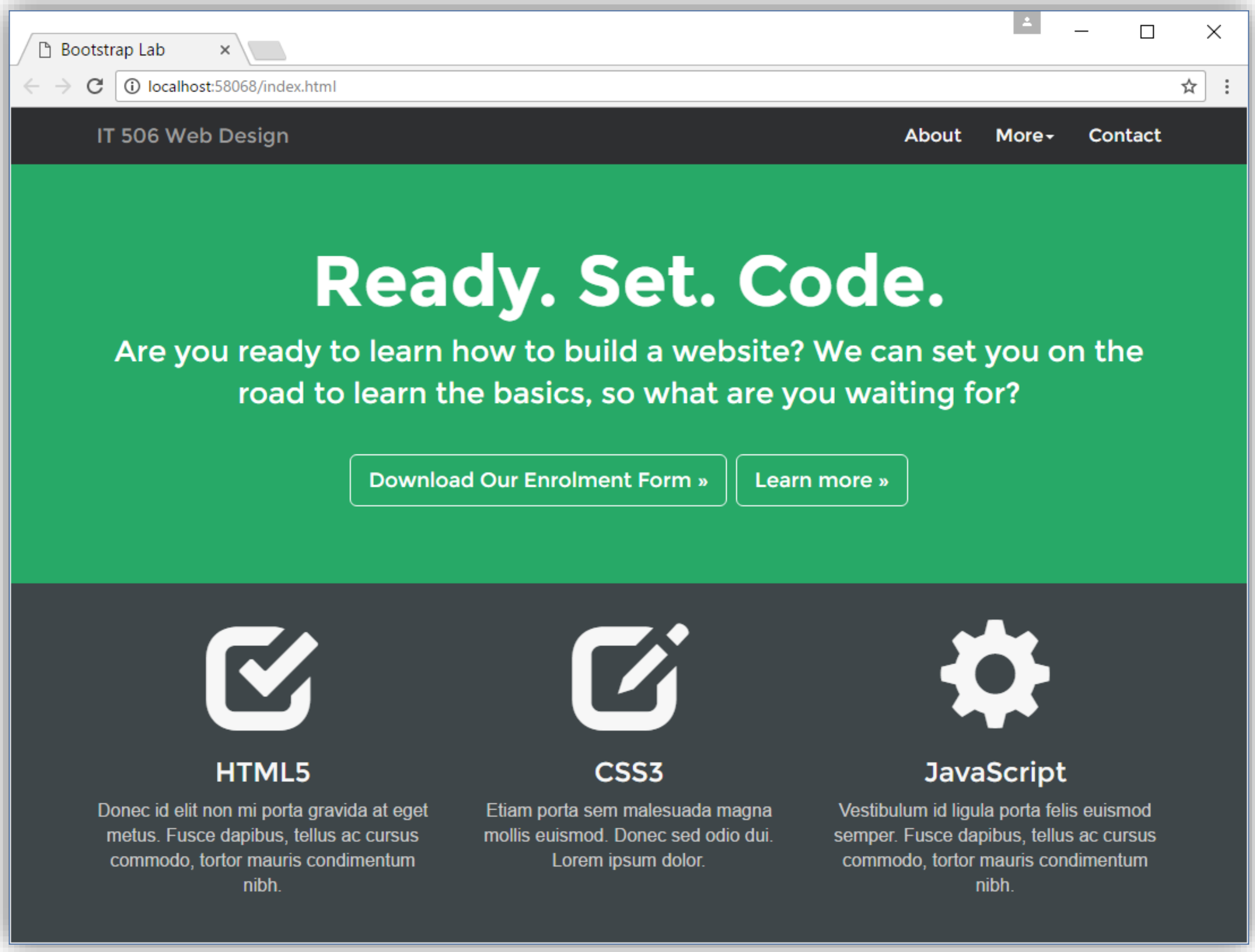
We will also use a glyphicon-large class and add it to each icon span.

```
<span class="glyphicon glyphicon-check glyphicon-large" aria-hidden="true"></span>
```

The final touches to the code are going to be centering the bottom call-out, adding margin-bottom to the p tags so they stack properly on mobile, and making the glyphicons bigger.

```
.messages {
	text-align: center;
}
.messages p {
	margin-bottom: 30px;
	font-family: sans-serif;
}
.glyphicon-large {
	font-size: 100px;

}
```



Ready. Set. Code.

Are you ready to learn how to build a website? We can set you on the road to learn the basics, so what are you waiting for?

HTML5

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

CSS3

Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. Lorem ipsum dolor.

JavaScript

Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

Responsive

With one small page worth of code, we have completely transformed the layout. And even more importantly you can now resize your browser to observe how bootstrap takes care of the responsiveness for smaller media sizes.

Your Chapter 7 code for @media sizes was just a tiny portion of the CSS code that bootstrap now manages for you.

Conclusion

Hopefully you learned a bit about Bootstrap and frameworks from this tutorial.

This is hardly scratching the surface of what Bootstrap is capable of – but you can take it from here. The documentation is enormous, and if ever you can't figure out how to do something, chances are a quick Google search will point you in the right direction.

Spend the rest of your Lab time exploring these websites.

http://getbootstrap.com

http://www.w3schools.com/bootstrap/default.asp

Lab Submission

There is no Lab submission this week BUT there is a Bootstrap one planned next week.

So thinking ahead…..start experimenting with bootstrap features now.

Your job will be to create a bootstrap website, cutting and pasting code directly from http://getbootstrap.com/components and build yourself a website to demonstrate that you have tried out some of the components.

There does not need to be any particular topic or text….just some visual bootstrap elements.

I expect to see a navigation different to the one we have already explored.

In the main section I need to see some responsive grid columns

Maybe some panels, some buttons….

Maybe explore some of the bootstrap input buttons – especially some of the grouped buttons

At first your page will be a heap of experimental components but by next week you will need to shape this up to a more coherent design. All the time you should be thinking about how your website project might be able to use this.

A "vanilla" version of the bootstrap components is ok….but if you really want to excel in using bootstrap then you will want to experiment with customising your display.

I know the instructions are a bit vague….just keep in mind that your job is to show me that you are experimenting with bootstrap features. The main message is :- **Do not upload this tutorial example!**