# Inclass 3

Due December 2, 9AM

Submit all of your answers inside this to the inclass document. Please upload it to the appropriate drop box.

20% deducted each day this is late.

Worth 15% of your final mark.

Contents

[Inclass 3 1](#_Toc467685190)

[Twitter BootStrap 1](#_Toc467685191)

[Responsive Grid System 2](#_Toc467685192)

[NavBar 3](#_Toc467685193)

[Customizing the Styles 6](#_Toc467685194)

[Editing Your Settings 7](#_Toc467685195)

[Creating a Color Theme 8](#_Toc467685196)

[Adobe Color Wheel (Formerly Kuler) 8](#_Toc467685197)

[Color Scheme Designer 8](#_Toc467685198)

[Eyedropper Tool in Firefox Developer 10](#_Toc467685199)

[Pre-made Templates 10](#_Toc467685200)

[Assignment 2 Preparation 14](#_Toc467685201)

# Twitter BootStrap

Twitter Bootstrap is a web styling framework. Bootstrap is one of the most popular styling frameworks so this has positives and negatives. However, the framework really did take the web by storm and it remains a favourite amongst many developers who continue to use it so they can spend more time on functionality. Twitter Bootstrap is easy to use, mobile responsive, powerful and it gives you the ability to customize the look and feel of your projects without too much effort. Since I began using Twitter Bootstrap in the classroom I have noticed that the professional appearance of student projects has increased significantly.

When you add in a reference to the Twitter Bootstrap styling you immediately get a reference to tonnes of styles for things like data grids, buttons, icons (called glyphicons), and much more.

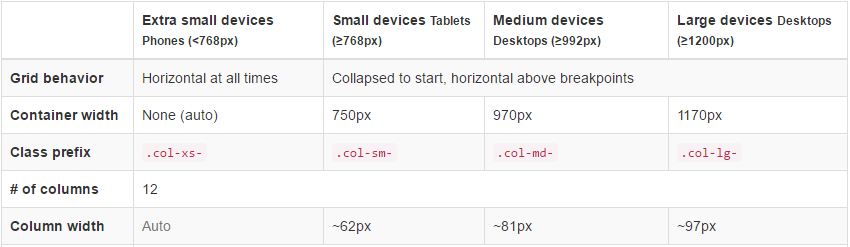
## Responsive Grid System

The responsive grid is one of the most important and exciting items in Twitter Bootstrap. Basically, this grid allows you to place your content in cells which wrap or fold depending on the size of the device being shown. With a bit of planning, you can define how the grid will display on a phone, tablet or desktop.

The default number of columns in the grid is 12. If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line. To help, you can use pre-defined Bootstrap css classes which tell the framework to adjust sizing and display for the viewing device:

* **col-xs-n** specifies the number of columns for a phone.
* **col-sm-n** sets the number of columns for a tablet.
* **col-md-n** or **col-lg-n** sets the number of columns for the desktop.

Here is a listing of default device widths by category which are documented on the Bootstrap site.



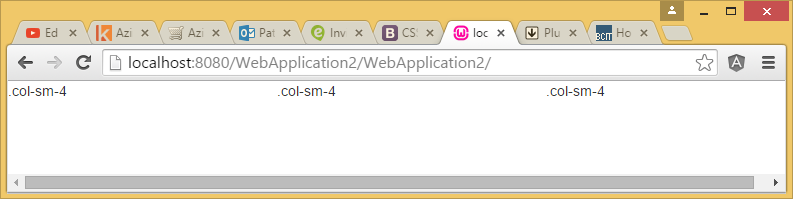
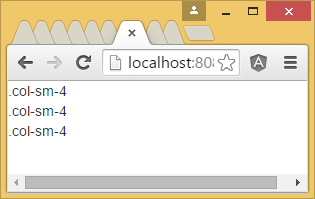
The following Bootstrap css classes also exist to hide and show columns depending on the viewport size:

|  |  |
| --- | --- |
| * hidden-xs * hidden-sm * hidden-md * hidden-lg | * visible-xs * visible-sm * visible-md * visible-lg |

There are many other classes too which allow you to adjust the grid display to different device sizing.

Example 1: BootStrap Grid

🞑 This example shows how to display data in one row in a tablet or desktop and in three separate rows in a phone. You can observe the difference in displays by shrinking and expanding the browser view width.



|  |
| --- |
| <!doctype html>  <html>  <head>  <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.css" rel="stylesheet">  </head>  <body>  <div class="row">  <div class="col-xs-12 col-sm-4">.col-sm-4</div>  <div class="col-xs-12 col-sm-4">.col-sm-4</div>  <div class="col-xs-12 col-sm-4">.col-sm-4</div>  </div>  </body>  </html> |

Exercise 1

🖍 Starting with Example 1, adjust the grid so the third column does not appear on a small device. Test your adjustment by expanding and shrinking the browser.

|  |
| --- |
|  |

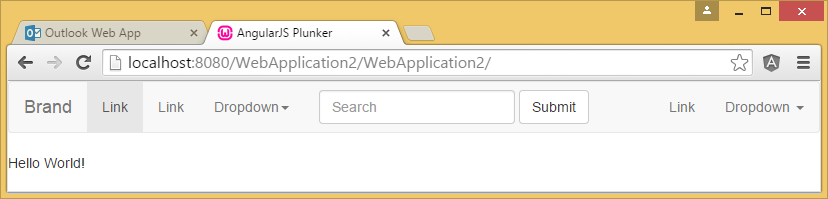
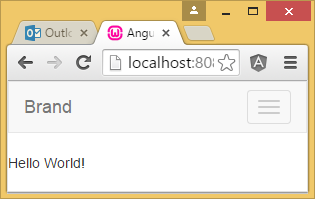
Exercise 2

🖍 Starting with Example 1, create a responsive grid that shows each word from this list - (apples, peaches, pears, plums) in a separate column within one row on the desktop. Also using the same grid, show each word in its own row in a tablet and phone. Show your HTML here:

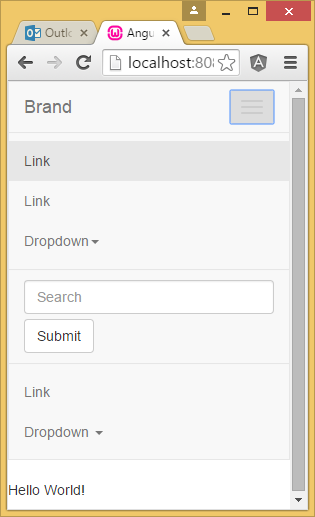
|  |
| --- |
|  |

## NavBar

The navbar is also a popular feature of Twitter Bootstrap. You can usually identify it right away on the web in your phone when you see the 3 layer icon bar known as a hamburger on the right.



On phones, clicking the hamburger opens the menu:



The navbar usually serves as the main menu for a site and it is responsive. You need to include a reference to jQuery to make it work. I really like the jQuery slim:

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.slim.min.js>

Example 2: Bootstrap Navbar

🞑 Here is a simple demo which shows the features of a responsive Bootstrap navbar like the ones displayed in the screenshot above.

**index.html**

|  |
| --- |
| <!doctype html>  <html>  <head>  <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.css" rel="stylesheet">  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.slim.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  </head>  <body>  <nav class="navbar navbar-default">  <div class="container-fluid">  <!-- Brand and toggle get grouped for better mobile display -->  <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="#">Brand</a>  </div>  <!-- Collect the nav links, forms, and other content for toggling -->  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  <ul class="nav navbar-nav">  <li class="active"><a href="#">Link <span class="sr-only">(current)</span></a></li>  <li><a href="#">Link</a></li>  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>  <ul class="dropdown-menu">  <li><a href="#">Action</a></li>  <li><a href="#">Another action</a></li>  <li><a href="#">Something else here</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">Separated link</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">One more separated link</a></li>  </ul>  </li>  </ul>  <form class="navbar-form navbar-left">  <div class="form-group">  <input type="text" class="form-control" placeholder="Search">  </div>  <button type="submit" class="btn btn-default">Submit</button>  </form>  <ul class="nav navbar-nav navbar-right">  <li><a href="#">Link</a></li>  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>  <ul class="dropdown-menu">  <li><a href="#">Action</a></li>  <li><a href="#">Another action</a></li>  <li><a href="#">Something else here</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">Separated link</a></li>  </ul>  </li>  </ul>  </div><!-- /.navbar-collapse -->  </div><!-- /.container-fluid -->  </nav>  </body>  </html> |

Exercise 3

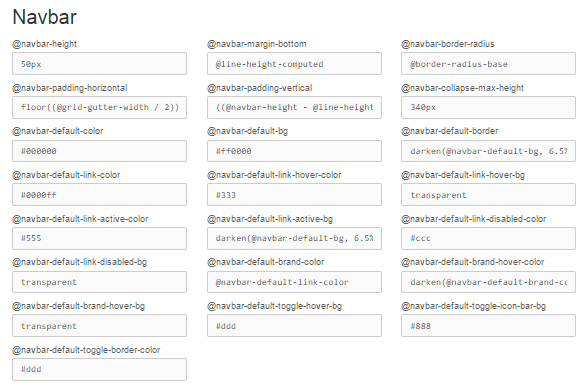
🖍Adjust the navbar in Example 2 by adding a drop down titled ‘Fruit’ on the left but after the brand. Add ‘apples’, ‘peaches’, ‘pears’, and ‘plums’ as choices. As well, remove the search input box and button. Ensure your navbar remains responsive after the changes. Show your revised HTML:

|  |
| --- |
|  |

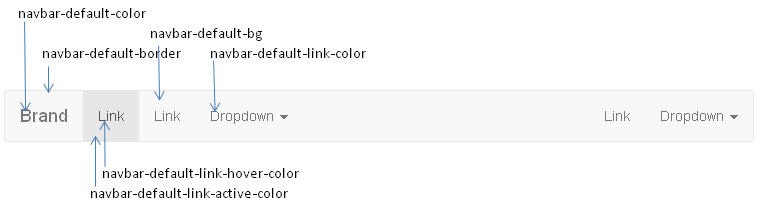
## Customizing the Styles

It is possible, and essential really, to customize the look of the Twitter Boostrap styles. To customize your Bootstrap styles go to <http://getbootstrap.com/customize/>

There are many settings to choose from. As an example, in the screenshot below, you can enter the variable values for the colors and sizes you wish to use in the nav bar.

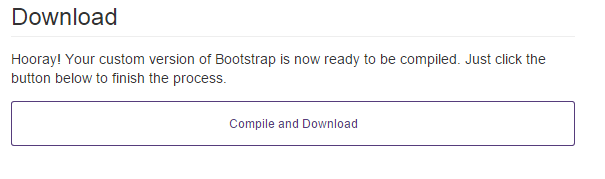


The diagram below highlights a handful of properties that can be adjusted.



Note: If you don’t like the rounded corners on the nav bar just set the navbar-border-radius property to 0 at the top right of the navbar section.

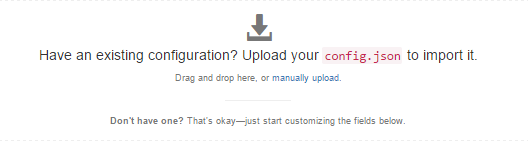
After you finish adjusting properties for the Bootstrap styles, you can download the framework with these styles by clicking the “Compile and Download” button at the bottom of the page.



With Angular UI, you only need to obtain the bootstrap.css file and the config.json file from the downloaded archive. The config.json file stores your settings in LESS variables.

### Editing Your Settings

To update your settings, you can upload your existing config.json file at http://getbootstrap.com/customize/ by selecting the “manually upload” link below.



This pre-populates the form at http://getbootstrap.com/customize/ with your settings. Then you can customize and download the updated bootstrap.css to use in your project. Once again, the config.json file will store your latest custom settings which you can use later to edit your styles if desired.

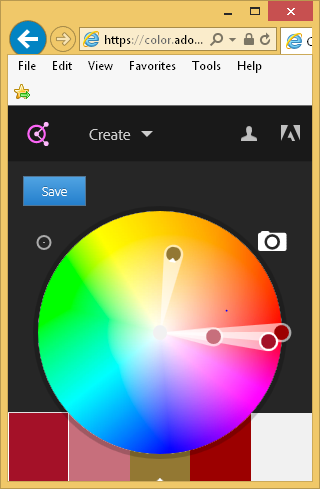
## Creating a Color Theme

This is a side topic but one you may appreciate as a programmer a stereo-typical programmer with no color sense. Many of us are pretty bad with colors so you can use a tool to generate a color palette for you. There are lots of great tools out there to create color themes.

### Adobe Color Wheel (Formerly Kuler)

A tool that I really like is the Adobe Color Wheel which used to be named Kuler. This tool is used for generating a palette of colors that mix well together.

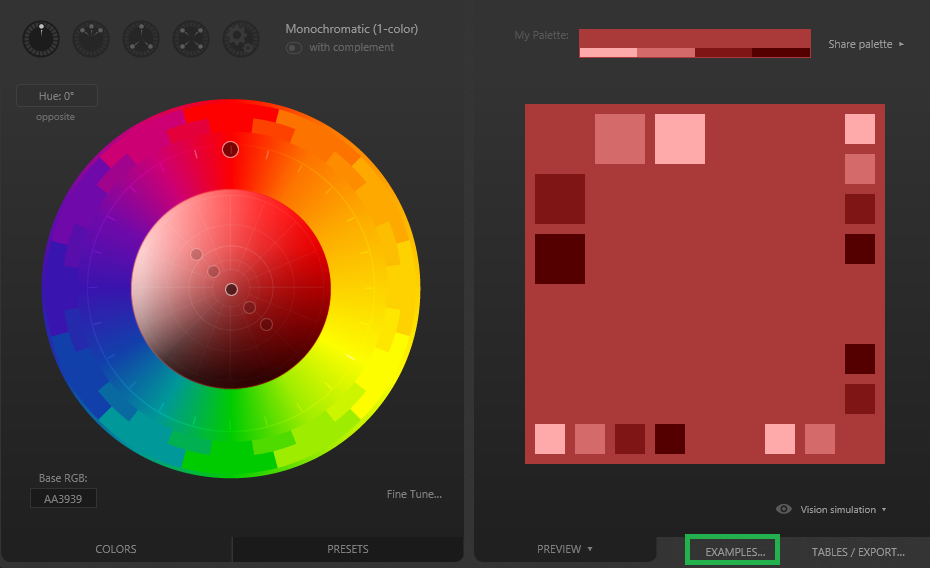
https://color.adobe.com/create/color-wheel/



Your color selection will actually be saved in the URL when you select colors. You can also upload a picture by clicking on the camera. This application will read the picture and generate a palette using the common colors in your image.

### Color Scheme Designer

You can also generate a color palette using color scheme designer at http://paletton.com. This tool is great because it provides samples of pages that use the color pallete. If you click around in the example area you will also find background animations that use these colors.



Your color scheme is also saved in the URL as you select the colors. This way you can return to Color Scheme Designer to adjust your colors or you may refer to it later.

<http://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>

Another cool feature is you can view page templates which give you an idea of how to apply the styles and you can also fine tune the colors:



Exercise 4

You create a really great color theme at <http://paletton.com>. Before you shut off your computer and head out onto the dance floor how can you save your settings?

|  |
| --- |
|  |

### Eyedropper Tool in Firefox Developer

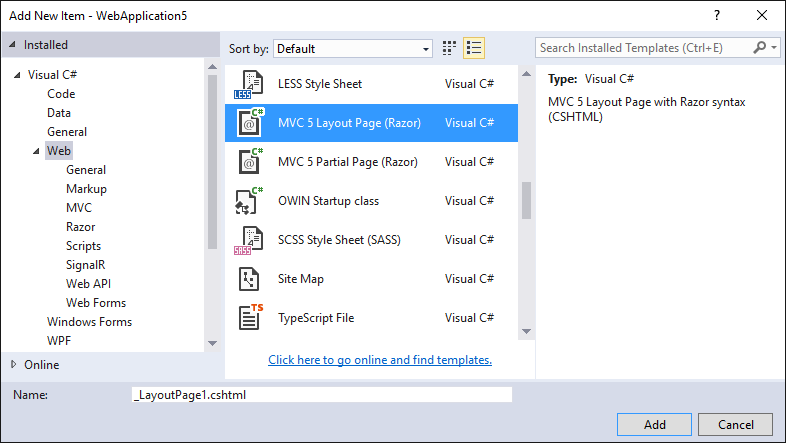
The eye dropper tool that ships with Firefox Developer edition can be accessed from the web developer tools menu. You can use this utility to select a color by the pixel on any web page you are viewing. Once selected, you can then paste the selected value into the text box for a color property on the Bootstrap customization form. The hexadecimal color value is copied directly from the eyedropper to the form without requiring any further adjustments - this is awesome.

### Pre-made Templates

Perhaps you don’t want to use a palette. Instead, there may be a site that implements a color scheme which you really like. In this case you can just use the eyedropper tool to copy the colors you like into the different Bootstrap properties on the customization page.

Exercise 5: Styling the Twitter Bootstrap navbar

🖍 Create an empty MVC project with a Home controller. Add a ‘Shared’ folder to the ‘Views’ folder. Then add a layout page to the shared folder. You can do this by right clicking the ‘Shared’ folder and choosing Add | New Item. In the Add New Item dialog that appears select Web and then MVC 5 Layout Page (Razor). Then click Add.

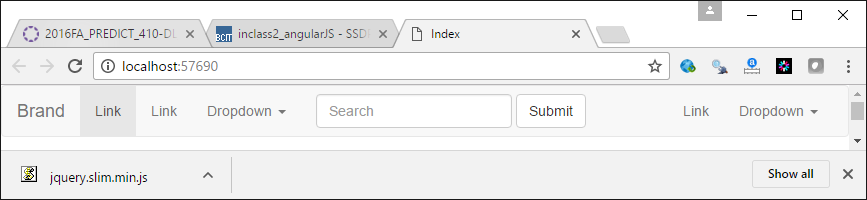


Next, paste the following code in the layout. This code is basically the same as Example 2. However, the items highlighted in yellow allow you to implement ASP.NET MVC features.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width" />  <title>@ViewBag.Title</title>  <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.css" rel="stylesheet">  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.slim.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  </head>  <body>  <nav class="navbar navbar-default">  <div class="container-fluid">  <!-- Brand and toggle get grouped for better mobile display -->  <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="#">Brand</a>  </div>  <!-- Collect the nav links, forms, and other content for toggling -->  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  <ul class="nav navbar-nav">  <li class="active"><a href="#">Link <span class="sr-only">(current)</span></a></li>  <li><a href="#">Link</a></li>  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>  <ul class="dropdown-menu">  <li><a href="#">Action</a></li>  <li><a href="#">Another action</a></li>  <li><a href="#">Something else here</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">Separated link</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">One more separated link</a></li>  </ul>  </li>  </ul>  <form class="navbar-form navbar-left">  <div class="form-group">  <input type="text" class="form-control" placeholder="Search">  </div>  <button type="submit" class="btn btn-default">Submit</button>  </form>  <ul class="nav navbar-nav navbar-right">  <li><a href="#">Link</a></li>  <li class="dropdown">  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>  <ul class="dropdown-menu">  <li><a href="#">Action</a></li>  <li><a href="#">Another action</a></li>  <li><a href="#">Something else here</a></li>  <li role="separator" class="divider"></li>  <li><a href="#">Separated link</a></li>  </ul>  </li>  </ul>  </div><!-- /.navbar-collapse -->  </div><!-- /.container-fluid -->  </nav>  <div>  @RenderBody()  </div>  </body>  </html> |

Next, add a Home controller and from the index action method, add a weakly typed view. Select your new layout page when prompted.

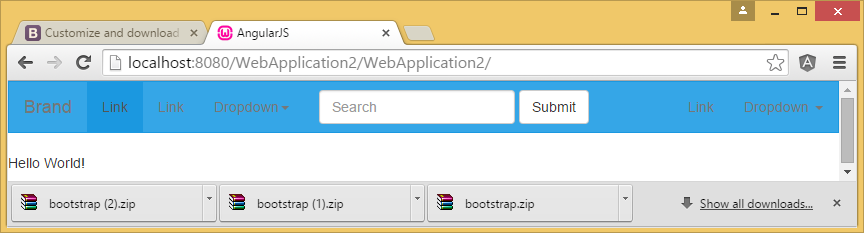
Run your project to ensure it works. You should see the menu appear and when you click on the drop down link you should see the drop down.



Customize the default background color of the navbar using the **@navbar-default-bg** property. Also, customize the background color for a link hover with the **@navbar-default-link-hover-bg**. Finally, eliminate the rounded corners on the nav bar by setting the **@navbar-border-radius** property to 0. When you are finished, your navbar should look differently than the default great navbar. Download your new Bootstrap css and reference it in your project by replacing the reference to the Bootstrap styling with your own local file:

|  |
| --- |
| <head>  <meta charset="utf-8" />  <title>AngularJS</title>  <link href="bootstrap.css" rel="stylesheet" />  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular.js">  </script>  <script src="http://angular-ui.github.io/bootstrap/ui-bootstrap-tpls-0.14.3.js"></script>  <script src="example.js"></script>  </head> |

In the end, your background colors should appear differently than the default:



To complete this exercise just state that you tried the exercise and it worked:

|  |
| --- |
|  |

Exercise 6

🖍 In this exercise, use the solution from Exercise 5. Then open on a new page. Take the config.json file from your last downloaded bootstrap archive and upload it to http://getbootstrap.com/customize/. This will load your customized color settings so you can edit them. Then, customize the default font color for the links of the navbar using the **@navbar-default-color** property. Download the updated boostrap styles and extract them. Try them in your project to ensure the new font style is applied. Indicate here that you tried this exercise and that it worked.

|  |
| --- |
|  |

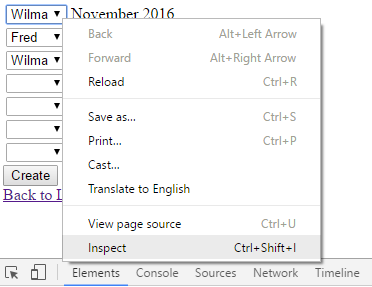
# Assignment 2 Preparation

Assignment two requires you to submit (via post) a list of information rather than just one record. Unfortunately the wizard is not able to generate code to handle such cases so you have to manually add some code to your views to enable this. A helper project has been provided to help you figure this out.

Build the project and add manually insert some volunteers through Microsoft SQL Server’s Management studio.

Exercise 7

After adding a couple of names to the Volunteer table in SQL Server Management Studio, when you run the project, use the Chrome debugger to inspect the HTML that is generated for the first two rows of the table which displays names and months.



Take a screenshot of the first two rows of the table and be sure to expand the cells so you can view all of the html for all columns within each row of the table. Show your screenshot of the HTML that appears in the HTML element tab here.

|  |
| --- |
|  |

Exercise 8

How is the name that is assigned to each control within the table of the view different than the name that appears in the header of the POST enabled Index action method? Is the letter case any different?

|  |
| --- |
|  |

Exercise 9

1. In the view that is rendered in the browser, how are the names for the select and hidden input controls different from one row of the table to the next?

|  |
| --- |
|  |

1. Why are the names set up in this manner?

|  |
| --- |
|  |

Exercise 10

When you post multiple rows of data using the helper project, if you set a break point inside the POST enabled action method to inspect the data that is passed to it you will notice each item within the data appears to have properties. Explain how you think ASP.NET figures out how to assign values to the properties of each TaskAssignmentVM object.

|  |
| --- |
|  |

Exercise 11

In the view, change the name for each control within the form (except the button) so it starts with:

myData[..]

Instead of:

taskAssignmentVM[..]

Next, set a break point in the POST enabled action method and inspect the data that gets passed to it when you submit the form after choosing names for each drop down and submitting the form. Does the data get passed to the POST enabled action method after the change? Explain why you think this is.

|  |
| --- |
|  |