Dingle Coderdojo 3

Week 3

Making It Look Good! – Continued.

Last week we went through some basic styling examples to style a table. Seems like a lot of work, doesn't it? If only there was a shortcut…

We can very easily make the website look good by using a style framework called Bootstrap. Created by some developers in Twitter in 2010, it has exploded into a widely used framework that makes it easy to create good looking web pages that work well on a variety of screen sizes - including desktops, laptops, tablets and mobile phones. All you have to do is add some lines of code to the <HEAD> section of your code, and set up your page to use it.

The idea behind Bootstrap is to represent a webpage in terms of 12 invisible columns on screen. On a full screen monitor, all 12 columns will be shown side by side. On smaller screens, you can set it up so that elements stack on top of each other. That way, the text and content are still nice and big on screen, and you don't have to "pinch and zoom" the screen to read it.

The monitor image on the left is wide enough to put a header panel and 4 content columns underneath. Each of the 4 smaller columns span 3 of the invisible columns, adding up to 12 in total.
The laptop image is a smaller screen, so shows the header panel and 3 content columns underneath. Each of the 3 smaller columns span 4 invisible columns each, adding up to 12 in total.
The tablet shows the header image and two columns. Each of the 2 smaller columns span 6 invisible columns (adding up to 12 in total).
The phone shows the header image and one column. Each of the columns span all 12 of the invisible columns.
Bootstrap automatically decides how many columns to show, based on the screen size. The text should be legible on all screens, and you will never need to scroll horizontally (from left to right).
Before we start bootstrap, there are a couple of other HTML elements and concepts you need to understand.
We learned about **CSS** last week, implementing styles using a **`<STYLE>`** section. To use bootstrap, we need to link to Bootstrap's **CSS file**, which contains lots of styles, ready made for you to use. CSS files are linked from the `<HEAD>` section of the page, using a `<LINK>` tag. One other element you need for pages styled in Bootstrap is a `<DIV>.` A `<DIV>` is an invisible box on screen that can be used as a container and is fundamental to layout. Putting together **Bootstrap, DIVs, IDs** and **CSS Classes**, let's see what we can do.

```
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
        scale=1">

        <title>My First Bootstrap Page</title>
        <link
        href="https://sensorpro.net/coderdojo/assets/css/bootstrap.
        css" rel="stylesheet">
        <link
        href="https://sensorpro.net/coderdojo/assets/css/fontawesom
        e.min.css" rel="stylesheet">
    </head>
    <body>
    <div id="headline" class="jumbotron">
        <div class="container">
            <div class="row">
                <div class="col-md-12 text-center">
                    <h1>Hello, world!</h1>
                    <p>My First Bootstrap project. :)</p>
                </div>
            </div>
        </div>
    </div>
    </body>
</html>
```

Notes:

1. The 3 “meta” tags at the top help the bootstrap page display correctly on all screens/devices.

2. There are two stylesheets linked at the top (see <link> tags). The first is to the bootstrap stylesheet itself, the second is for something called “font awesome” which we will cover later.

3. There are 3 nested DIVs (containers) on the page:

a. The “jumbotron” div is a large header section that you put at the top of a page.
b. The “container” div is just a container inside the jumbotron div and centres everything on screen.
c. The div with the “col-md-12” class is the actual content.

i. “md” is medium size screens
ii. “12” is the number of invisible columns that the div will take up. You will recall that 12 is the full width of the screen or device.

iii.We will experiment with other screen sizes and columns later.

Let’s flesh it out a bit more. Add the following piece after all the DIVs from the last section (in between the last </DIV> and </BODY>)

```
<div id="panels" class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="panel panel-danger">
                <div class="panel-heading">
                    <h3 class="panel-title">Panel Title Style
                    1</h3>
                </div>
                <div class="panel-body">
                    Panel 1 content
                </div>
            </div>
        </div>
        <div class="col-md-6">
            <div class="panel panel-primary">
                <div class="panel-heading">
                    <h3 class="panel-title">Panel Title Style
                    2</h3>
                </div>
                <div class="panel-body">
                    Panel 2 content
                </div>
            </div>
        </div>
    </div>
</div>
```

Notes:

1. This should display two differently styled panels side by side, each taking up 6 columns of our 12 column grid.

2. <h3> is a heading type (you can use <h1> to <h6> for different sized headings). <h1> is the biggest.

3. Try resizing your browser window and see what happens.