

# Exercise: Responsive Design and Bootstrap Grid System

## Part 1

### Objectives and Outcomes

This exercise introduces you to responsive design and Bootstrap support for mobile first responsive design through the use of the grid system. At the end of this exercise, you will be able to:

- Create responsive websites using the Bootstrap grid system
- Reordering content using push, pull and offset classes

**Note: In this exercise we will continue to update the *index.html* file in the *conFusion* folder that we created and edited in the previous lecture.**

### Bootstrap Grid System and Responsive Design

Bootstrap is designed to be mobile first, meaning that the classes are designed such that we can begin by targeting mobile device screens first and then work upwards to larger screen sizes. The starting point for this is first through media queries. We have already added the support for media queries in the last lesson, where we added this line to the head:

1	<code>&lt;meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"&gt;</code>	
	alternative is to use the container-fluid class to make the content automatically to span the full width of the screen. We will discuss further about this when we discuss the Bootstrap grid system in the next lecture. Add the container class to the first div right after the </header> in the file as follows.	
1	<code>&lt;div class="container"&gt; ...</code>	

### Dividing the content into rows

- Let us now add the class *row* to the first-level inner *div* elements inside the container. This organizes the page into rows of content. In the next exercise, we will see how we can add other classes to the rows.

1	<code>&lt;div class="row"&gt; ...</code>	
---	--	--

### Creating a Jumbotron

- Let us add the class *jumbotron* to the header class as shown below. This turns the header element into a Bootstrap component
- Applying column classes within each row
- In the header row, we will display the restaurant name and the description to occupy 6 columns, while we will leave six columns for displaying the restaurant logo in the future. Let us go into the jumbotron and define the classes for the inner divs as follows:

1	<code>&lt;div class="col-12 col-sm-6"&gt; ... &lt;/div&gt;</code>	
2		
3	<code>&lt;div class="col-12 col-sm"&gt; ... &lt;/div&gt;</code>	

- For the remaining three div rows that contain the content, let us define the classes for the inner divs as follows:

1	<code>&lt;div class="col-12 col-sm-4 col-md-3"&gt; ... &lt;/div&gt;</code>	
2		
3	<code>&lt;div class="col col-sm col-md"&gt; ... &lt;/div&gt;</code>	

- For the footer, let us define the classes for the inner divs as follows:

1	<code>... the content &lt;div class="col-12 col-sm-4 col-md-2"&gt; ... &lt;/div&gt;</code>	
	push to alternate so that it gives an interesting look to the web page. For extra small screens, the default stacked layout works best. This can be accomplished by using the <i>.order-sm-last</i> and <i>.order-sm-first</i> for the first and the third rows as follows:	
1	<code>&lt;div class="col-12 col-sm-4 order-sm-last col-md-3"&gt; ... &lt;/div&gt;</code>	
2		
3	<code>&lt;div class="col col-sm order-sm-first col-md"&gt; ... &lt;/div&gt;</code>	

```
4 <div class="col-4 offset-1 col-sm-2">
```

- For the div containing the <ul> with the site links, update the class as follows:

```
1 <div class="col-4 offset-1 col-sm-2">
```

- After saving all the changes, you can do a Git commit with the message "Bootstrap Grid Part 1" and push your changes to the online repository.

Conclusion

In this exercise, we reviewed responsive design and the Bootstrap grid system.