*Bootstrap* is a popular CSS framework with *prewritten* CSS rules designed to help you build webpages faster.

**2.**

**1.**

**2.**

**3.**

**4.**

**1.**

**4.**

**3.**

**4.**

**5.**

**6.**

**4.**

**2.**

**1.**

**2.**

**3.**

**1.**

**5.**

**1.**

**2.**

**3.**

Instructions

Take a look at the web browser. In this lesson, we will build this webpage up from scratch using a combination of Bootstrap and custom CSS.

Also in this lesson, we will encounter three new HTML elements:

1. *header*: Used to organize headers on a webpage.
2. *footer*: Used to organize footers on a webpage.
3. *section*: Defines sections on a webpage.

Before Bootstrap can work for us, we need to link to it.

In earlier lessons, we linked only to **main.css**. Now, in addition to **main.css**, we will link to a URL that hosts Bootstrap.

The HTML link element makes Bootstrap available to us. Remember that the link element is typically contained within HTML head tags.

<head>

...

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>

...

</head>

Above, the href attribute is set to the URLhttps://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css.

Let's explore what Bootstrap can do for us.

Notice the "Skillfair" webpage to the right and locate the diagonal arrows in the top right corner of the web browser (nearhttps://localhost:8000/). Click the arrows to toggle the web browser between narrow and full-width views, and observe the following:

1. Navbar items do not change position as you toggle the webpage width.
2. The "Homemade Goods" text appears on the far left side of the webpage.
3. Photographs increase in size when the web browser is toggled to full-width.
4. Social media icons at the bottom of the webpage appear disorganized.

Click Run after you have observed these points.

Now we will connect to Bootstrap and see changes to the webpage layout.

In **index.html**, locate the head element. Inside head, create a linkelement to link to Bootstrap:

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>

Click Run.

Toggle the web browser's width again to notice the following changes:

1. Navbar items change position based on the webpage's width.
2. The "Homemade Goods" text is now centered.
3. "Kitchen", "woodwork", "antique" and "gifts" photographs are arranged two per row when the webpage is full-width.
4. Social media icons appear organized and change position at full and narrow widths.

Bootstrap's grid is responsible for this screen-width responsiveness. Click Next to learn more!

One of the reasons Bootstrap and frameworks like it are so popular is because they offer *grids*. A *grid* makes it possible to organize HTML elements using preconfigured columns. Using a grid, you can customize responsive page layouts quickly and reliably.

The Bootstrap grid contains *12 equal-sized columns*, as seen in the diagram on the right. HTML elements are arranged to span different numbers of columns in order to create custom page layouts.

Instructions

In the diagram, observe the following:

1. Bootstrap's grid columns are represented by 12 vertical bars. The boxes represent HTML elements.
2. The words "container", "jumbotron", "col-sm-6" and "col-sm-3" refer to Bootstrap classes.
3. The element with class "jumbotron" spans the entire width of the webpage, beyond the borders of the grid.
4. Elements inside the second "container", such as "col-sm-6" and "col-sm-3" are contained within the grid columns.
5. Elements labeled "col-sm-3" take up three grid columns; elements labeled "col-sm-6" take up six grid columns.

Take a moment to familiarize yourself with Bootstrap's grid, then click Next to continue.

Let's use Bootstrap's grid to create a simple header with navbar.

In the example code below, an HTML *header* element with Bootstrap's predefined container class is used:

<header class="container">

...

</header>

Inside the header, a row is created by using a div with Bootstrap's row class:

<header class="container">

<div class="row">

</div>

</header>

Finally, the row is cut into two parts:

<header class="container">

<div class="row">

<h1 class="col-sm-4">Heading</h1>

<nav class="col-sm-8 text-right">

<p>nav item 1</p>

<p>nav item 2</p>

<p>nav item 3</p>

</nav>

</div>

</header>

The first part consists of the h1 with Bootstrap's class col-sm-4. It will take up the first four columns on the grid. The second part consists of the navelement with class col-sm-8. It will take up the remaining eight grid columns.text-right indicates that text will be arranged on the right side of the nav.

In **index.html**, on the line below the opening <body> tag, create a header element with Bootstrap's container class:

<header class="container">

...

</header>

Click Run to continue.

Next, inside the opening <header class="container"> tag, create a div with the Bootstrap class row:

<header class="container">

<div class="row">

</div>

</header>

Click Run to continue.

Now we'll cut the row into two parts.

The first part: inside the <div class="row"> tag, create an h1 with Bootstrap's class col-sm-4. The content for the h1 will be the company name. You can use "Skillfair" or anything you'd like:

<header class="container">

<div class="row">

<h1 class="col-sm-4">Skillfair</h1>

</div>

</header>

Click Run to see the results in the web browser.

Now for the second part of the row. One line below the h1 element, create a nav element with Bootstrap's class col-sm-8 text-right:

<header class="container">

<div class="row">

<h1 class="col-sm-4">Skillfair</h1>

<nav class="col-sm-8 text-right">

</nav>

</div>

</header>

Click Run to continue.

Finally, inside the opening <nav class="col-sm-8">, create three p elements. The content for each p will be a navigation item. You could use item names such as "newest", "catalogue" and "contact":

<header class="container">

<div class="row">

<h1 class="col-sm-4">Skillfair</h1>

<nav class="col-sm-8 text-right">

<p>newest</p>

<p>catalogue</p>

<p>contact</p>

</nav>

</div>

</header>

Click Run to see the compeleted header/navigation in the web browser.

In addition to a header/navigation, many websites have a large showcase area featuring important content. Bootstrap refers to this as a *jumbotron*. Below is an example implementation of a jumbotron.

First, a new *section* element is created and assigned the jumbotron class:

<section class="jumbotron">

</section>

Next a div with the Bootstrap class container is used:

<section class="jumbotron">

<div class="container">

...

</div>

</section>

A div with the Bootstrap class row text-center can center subsequent child elements which will contain text:

<section class="jumbotron">

<div class="container">

<div class="row text-center">

...

</div>

</div>

</section>

The ... is a placeholder for HTML elements containing text, such as h2, p or anchor elements.

Let's explore the jumbotron feature by creating our own below!

In **index.html**, one line below the closing </header> tag, Create a section with the jumbotron class.

<section class="jumbotron">

</section>

Click Run. In the web browser, notice a large photo covering the main part of the webpage.

Inside the opening <section class="jumbotron"> tag, create a div with the container class:

<section class="jumbotron">

<div class="container">

</div>

</section>

Don't forget your </div> closing tag!

Click Run to continue. There won't yet be a visual change in the web browser.

To center text over the jumbotron image, create another div inside thecontainer div. Give the new div a class of row text-center:

<section class="jumbotron">

<div class="container">

<div class="row text-center">

...

</div>

</div>

</section>

The ... is a placeholder for where text elements will go next.

Click Run to continue. There still will not be a visual effect on the web browser.

Finally, add heading and anchor elements to the row.

The anchor element will have Bootstrap's btn btn-primary class, which will transform it into a button.

<section class="jumbotron">

<div class="container">

<div class="row text-center">

<h2>Homemade Goods</h2>

<h3>This Year's Best</h3>

<a class="btn btn-primary" href="#" role="button">See all</a>

</div>

</div>

</section>

Click Run to see the elements centered over the jumbotron image in the web browser.

If you'd like, replace text between h2 , h3 and a tags with text of your choice.

**Note**: The anchor element's href attribute, #, is a placeholder for an actual URL.

Many websites have a supporting content area. Supporting content can be arranged using Bootstrap's grid. Below is an example implementation of a supporting content area.

First, an HTML section element with the container class is used:

<section class="container">

</section>

Next, div elements with the row class are added:

<section class="container">

<div class="row">

</div>

<div class="row">

</div>

</section>

Finally, the rows are divided by using divs with Bootstrap's col-sm-... class.

<section class="container">

<div class="row">

<div class="col-sm-6">

...

</div>

<div class="col-sm-6">

...

</div>

</div>

<div class="row">

<div class="col-sm-6">

...

</div>

<div class="col-sm-6">

...

</div>

</div>

</section>

Above, two rows are divided into two equal parts. Each part takes up 6 of bootstrap's 12 columns. Using the col-sm-6 class ensures that this layout will appear when the user's screen is the width of a tablet device(768 pixels). On narrower screens, such as an iPhone, only one image per row will appear.

Let's create a supporting content area for our webpage!

In **index.html**, one line below the closing </section> tag of the jumbotron, start building a supporting content section. First, create a section element with the class container that will contain supporting content.

<section class="container">

...

</section>

Click Run to continue. There won't yet be a visual difference on the webpage.

Inside the opening <section class="container"> tag, create two divs with the Bootstrap class row:

<section class="container">

<div class="row">

...

</div>

<div class="row">

...

</div>

</section>

Be sure to include </div> closing tags for your rows!

Click Run to continue. There won't yet be a visual difference on the webpage.

Inside each row, create two figure elements with the Bootstrap classcol-sm-6:

<section class="container">

<div class="row">

<figure class="col-sm-6">

...

</figure>

<figure class="col-sm-6">

...

</figure>

</div>

<div class="row">

<figure class="col-sm-6">

...

</figure>

<figure class="col-sm-6">

...

</figure>

</div>

</section>

Make sure you have a total of four figure elements.

Click Run to continue. As in previous steps, no change will be seen the web browser.

Finally, inside each figure, add two additional elements:

1.A p element that contains one of the following catagories:   
  
-kitchen  
-woodwork  
-gifts  
-antiques  
  
For example:

<p>kitchen</p>

2.An image element containing one of the following image URLs:

https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/kitchen.jpg

https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/woodwork.jpg

https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/gifts.jpg

https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/antique.jpg

[Here is a hint](https://www.codecademy.com/en/courses/make-a-website/lessons/site-structure/exercises/add-image) on how to create img elements.

<img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/kitchen.jpg"/>

[Here is how the webpage should look](https://www.codecademy.com/en/courses/make-a-website/lessons/bootstrap/exercises/css-frameworks?action=lesson_resume) like when you've completed this step.

<https://www.codecademy.com/en/courses/make-a-website/lessons/bootstrap/exercises/css-frameworks?action=lesson_resume>

Click Run to view the supporting content section in the web browser. If you toggle between screen widths, you will see how the layout changes

Congratulations! You've made it to the bottom of the webpage: the *footer*!

Footers display copyright information, social media links and sometimes site navigation.

Below is one possible implementation for a footer section using Bootstrap:

First, a footer element with Bootstrap's container class is used:

<footer class="container">

</footer>

Inside the footer, a div with Bootstrap's row class is added to hold footer content:

<footer class="container">

<div class="row">

...

</div>

</footer>

Finally, the row is divided into parts using Bootstrap's grid. Here is one suggestion:

<footer class="container">

<div class="row">

<p class="col-sm-4">...</p>

<ul class="col-sm-8">

<li class="col-sm-1">...</li>

<li class="col-sm-1">...</li>

<li class="col-sm-1">...</li>

</ul>

</div>

</footer>

Above, the row is broken into three parts: a p element that takes up four columns, a ul which takes up 8 columns, and li items which take up 1 column each. The lis could hold navigation menu items or social media icons.

Again, because the col-sm-... class is used, this layout will appear on tablet-width screens and wider. Screen sizes smaller than tablet-width (768 pixels), will not maintain this layout.

Now let's write the code for

Let's create a footer. Below the supporting content's closing</section> tag, create a footer element with the Bootstrap classcontainer:

<footer class="container">

...

</footer>

Click Run to continue.

Beneath the opening <footer class="container"> tag, create a div with the Bootstrap class row:

<footer class="container">

<div class="row">

...

</div>

</footer>

Don't forget your closing </div> tag! Click Run to continue.

Next divide the row using Bootstrap's grid.

Create a p element with the class col-sm-4 and <ul> element with the class col-sm-8:

<footer class="container">

<div class="row">

<p class="col-sm-4">...</p>

<ul class="col-sm-8">

...

</ul>

</div>

</footer>

Click Run to continue.

Between the <p class="col-sm-4"> and closing </p> add the website copyright:

&copy; 2016 Skillfair

&copy is a *character code*, which web browsers interpret as the copyright symbol: ©.

Click Run to view the copyright in the web browser.

Between <ul class="col-sm-8"> and </ul>, create four li items. Each will have the class col-sm-1.

<ul class="col-sm-8">

<li class="col-sm-1">...</li>

<li class="col-sm-1">...</li>

<li class="col-sm-1">...</li>

<li class="col-sm-1">...</li>

</ul>

Click Run to continue.

Finally, inside each <li class="col-sm-1">, add an img element that links to a social media icon.

**Twitter**

<li class="col-sm-1">

<img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/twitter.svg">

</li>

**Facebook**

<li class="col-sm-1">

<img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/facebook.svg">

</li>

**Instagram**

<li class="col-sm-1">

<img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/instagram.svg">

</li>

**Medium**

<li class="col-md-1">

<img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/medium.svg">

</li>

Click Run and view the results in the web browser.

You just built an impressive webpage using the Bootstrap CSS framework. Nice work!

Let's take a moment to review the core concepts learned in this lesson

Instructions

#### WEB CONCEPTS

* *CSS Framework*: Set of prewritten CSS rules designed to help you build webpages faster.
* *Bootstrap Grid*: 12 equal-sized columns which can be utilized via Bootstrap classes to build responsive page layouts quickly and reliably.

#### BOOTSTRAP CLASSES

* *row*: Arranges HTML elements in rows, and can be useful when building headers/navigation menus, main feature sections, supporting content sections and footers.
* *jumbotron*: Used to create large showcase sections featuring important content.
* *col-sm-\**: Used to span a specified number of columns on the Bootstrap grid. The "sm" is short for "small", and maintains desired CSS layouts on small screens (tablet-sized).
* *text-right*: Bootstrap class used to orient text to the right side of the webpage.
* *btn btn-primary*: Bootstrap class used to style page elements as buttons.