Lesson 9: Front-End Frameworks and Bootstrap

[Overview](#h.dji9p6sff2kg)

[HTML Review](#h.4vfhsdd6tnm5)

[Div](#h.n2z7t93esy5b)

[Classes and IDs](#h.bapjcf83ten1)

[What are Frameworks?](#h.iuecsgllewjf)

[What is Bootstrap?](#h.mcqq787s7nn3)

[Alternatives to Bootstrap](#h.ut34vg285oxx)

[Using Bootstrap](#h.ule6ulkymekn)

[The Grid System](#h.wuru548g9gf1)

[Overview](#h.6ozyc8lb6jdz)

[Alternatives](#h.yiukbt5gcwla)

[Using the Grid System in Bootstrap](#h.yisoe2ok01up)

[Responsive Design](#h.a5w6nykpjcjd)

[Responsiveness in Bootstrap](#h.jakolak6kk3k)

[Bootstrap Components](#h.n77o6iekkx1c)

[Jumbotron](#h.ak5h41rbufob)

[Mission - Typography and Google Fonts](#h.6zcnqbw5g1zn)

[Features - Columns and Images](#h.p2ktn8chlqx0)

[Navbar](#h.u8q5e2esbw1l)

[Footer](#h.aoil195oio0e)

[Bonus! Font Awesome Social Media Links](#h.bntcbogjk67y)

[Bonus! Add Button to Jumbotron](#h.4c77tzpt7nty)

[Bonus! Add a Custom Media Query](#h.j64x41fuqsmt)

# Overview

Bootstrap and grid system

make pdf of coffee and talk about how to break it down

Google Fonts, Font-Awesome

Media queries

# HTML Review

## Div

The <div> tag is nothing more than a container that encapsulates (or, contains) other page elements and divides the HTML document into sections. Web developers use <div> elements to group together HTML elements and **apply CSS styles to many elements at once**.

**Bootstrap and other CSS frameworks rely on <div>** elements for page layout and for applying CSS styles using **classes**.

## Classes and IDs

An **id** should be unique within a page, so the id selector is used if you want to select a single, unique element. We call it using a # before the ID name.

A **class** is similar except that by convention, it can be used more than once on a page to help identify several things that should be treated in the same way. We call it using a period before the class name

# What are Frameworks?

# What is Bootstrap?

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. — [getbootstrap.com](http://getbootstrap.com/)

Bootstrap is a front-end framework that was developed by two engineers at Twitter. It now lives on GitHub as one of the most popular projects in the world.

Bootstrap provides a framework for building the front end of your web sites by doing a lot of work for you:

* Flexible grid system
* Responsive design
* Good foundational styles
* Common components like menus and popups
* Themes
* Basic JavaScript stuff for interactivity

## Alternatives to Bootstrap

You don’t have to use Bootstrap. You can create your own framework, or use many that already exist. Here are a few to check out in your free time:

* Foundation by Zurb: <http://foundation.zurb.com/>
* Susy: <http://susy.oddbird.net/>

## Using Bootstrap

You can download Bootstrap and even customize it to include only the components that you need on your site, but for simplicity we are going to use the CDN option.

CDN stands for Content Delivery Network. A CDN is a set of servers designed to serve static content (like images, JavaScript, and CSS files) very quickly to clients all over the world.

Copy these lines into your head to add all Bootstrap functionality (from <http://www.bootstrapcdn.com/> and <https://code.jquery.com/>):

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css" rel="stylesheet">

(add links for overriding stylesheets after bootstrap link)

<script src="https://code.jquery.com/jquery-1.11.3.js"></script>

<script src="<https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js>"></script>

(add links for overriding javascript after bootstrap link)

Bootstrap JavaScript depends on JQuery, so that’s why we also have to load JQuery.

Create a new folder called good-coffee. Inside that folder, create an html file called **index.html** and a css file called **style.css**. Add the correct links to your html header for bootstrap and your custom css file. Go ahead and add an h1 tag with “Welcome to Good Coffee”

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- To ensure proper rendering and touch zooming, add the viewport meta tag to your <head> -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Good Coffee</title>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" type="text/css" href="style.css">

<script src="https://code.jquery.com/jquery-1.11.3.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>

</head>

<body>

<h1>Welcome to Good Coffee</h1>

</body>

</html>

It already looks better than before!

# The Grid System

## Overview

In **graphic design**, a **grid** is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal, and angular) or curved guide lines **used to structure content**. The grid serves as an armature on which a designer can **organize graphic elements** (images, glyphs, paragraphs) **in a rational, easy to absorb manner.**

see udacity content

## Alternatives

For non-grid based (or at least not visually related to grid), checkout this:

<http://designshack.net/articles/layouts/ditching-the-grid-alternate-layout-strategies-and-tips/>

<http://line25.com/articles/25-innovative-websites-that-buck-the-design-trends>

## Using the Grid System in Bootstrap

Bootstrap’s grid system breaks the page down into 12 columns that all sit next to each other. If you want something to take up half the page, then you make it 6 columns wide. Want 3 columns? Make 3 divs that are 4 columns wide.

See container and grid documentation [here](http://getbootstrap.com/css/#overview-container).

Classroom challenge:

1. Break down the Good Coffee printout into its containers, rows, and columns. Also write down which specific classes to use for each. **Google Doc version of file is** [**here**](https://docs.google.com/document/d/1njgBdBTvy5BVJEJbwnXJcfNnQMCKbVP_grc7goBHh2k/edit?usp=sharing).
2. Build the skeleton of all your divs in index.html. Go ahead and add the mission (h3) and features (p) text. Note that navs and jumbotrons do not need to be in the grid scaffold.

<nav class="navbar">

<!-- navbar -->

</nav>

<!-- jumbotron -->

<div class="jumbotron">

<h1>Welcome to Good Coffee</h1>

</div>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<!-- mission -->

</div>

</div>

<div class="row">

<!-- features -->

<div class="col-md-4"></div>

<div class="col-md-4"></div>

<div class="col-md-4"></div>

</div>

<footer class="footer">

</footer>

</div>

# Responsive Design

Responsive web design (RWD) is an approach to web design aimed at crafting sites to provide an optimal viewing experience—**easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices** (from desktop computer monitors to mobile phones).

A site designed with RWD adapts the layout to the viewing environment by using:

* The **fluid, proportion-based grid** concept calls for page element sizing to be in **relative units** like percentages, rather than absolute units like pixels or points.
* **Flexible images** are also sized in **relative units**, so as to prevent them from displaying outside their containing element.
* **Media queries** allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

“Mobile first” is another concept in responsive design where many developers build sites to function first (or optimally) on mobile with the idea that scaling that up to desktop will still result in a beautiful site.

## Responsiveness in Bootstrap

Bootstrap built their framework to be responsive as long as you use the correct tags and classes. The containers, columns, images, media queries, and much more all have embedded responsiveness built in. We will see this in action as we create our Good Coffee page!

# Bootstrap Components

Bootstrap has many, many components built in. See <http://getbootstrap.com/> for all of them, especially the Components, CSS, and JavaScript tabs. We will just cover a few here.

## Jumbotron

Let’s style our Jumbotron!

<http://getbootstrap.com/components/#jumbotron>

Add style and test responsiveness in browser.

.jumbotron {

background-image: url("http://upload.wikimedia.org/wikipedia/commons/6/62/Drip\_Coffee\_Bangkok.jpg");

background-repeat: no-repeat;

background-size: cover;

background-position: center;

margin-top: -70px;

margin-bottom: 0px;

padding: 20%;

}

.jumbotron h1 {

color: #F7FAFE;

position: absolute;

top: 18;

left: 0;

padding-left: 20px;

}

## Mission - Typography and Google Fonts

Now let’s style our Mission statement area. It’s already almost there, but we should look at [typography classes](http://getbootstrap.com/css/#type).

First, find the right class to center the text.

<div class="row mission text-center">

Now, let’s learn about Google fonts. Go to Google fonts and look up both fonts we need. Add each to your “collection”. Then click “use” and copy the stylesheet link in between your Bootstrap and custom stylesheets into your head. Then copy the CSS font-family commands into the class you want to use them.

Add your Google font to the Mission and then complete the rest of its style. Move font color to body.

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css" rel="stylesheet">

<link href='http://fonts.googleapis.com/css?family=Muli|Raleway:500' rel='stylesheet' type='text/css'>

<link rel="stylesheet" type="text/css" href="style.css">

body {

background-color: #A2DED0;

color: #F7FAFE;

}

.mission {

font-family: 'Raleway', sans-serif;

background-color: #323232;

padding: 50px;

}

## Features - Columns and Images

First, let’s start with what we know and change the font and color.

<div class="row features">

.features {

font-family: 'Muli', sans-serif;

color: #323232;

}

Bootstrap will automatically make images responsive as long as you add the img-responsive class to the img tag. Check out the [documentation](http://getbootstrap.com/css/#images).

Add the 3 features images, adjust positioning, then test responsiveness. Yep, the coffee bean picture has a larger resolution so when we reduce, it suddenly pops larger than the others. How could we fix this? Normally with a max-width, but overriding this will break the Bootstrap responsiveness. Any other ideas? For now, I would edit the image to be 640px wide manually.

margin-top: 40px;

## Navbar

Now that we have some practice with Bootstrap, let’s go back to the navbar at the top. Navbars can be tricky. Often, it’s easiest to copy the code in the [documentation](http://getbootstrap.com/components/#navbar) and then just edit it for your project.

Add a navbar with the shown menu options. Copy the full code to see what it looks like then edit from there. (Walk through each of the major elements and classes first).

Hint: navbar-inverse is the class to make it dark with light text.

<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">  
 <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-expanded="false">Dropdown <span class="caret"></span></a>  
 <ul class="dropdown-menu" role="menu">  
 <li><a href="#">Action</a></li>  
 <li><a href="#">Another action</a></li>  
 <li><a href="#">Something else here</a></li>  
 <li class="divider"></li>  
 <li><a href="#">Separated link</a></li>  
 <li class="divider"></li>  
 <li><a href="#">One more separated link</a></li>  
 </ul>  
 </li>  
 </ul>  
 <form class="navbar-form navbar-left" role="search">  
 <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-expanded="false">Dropdown <span class="caret"></span></a>  
 <ul class="dropdown-menu" role="menu">  
 <li><a href="#">Action</a></li>  
 <li><a href="#">Another action</a></li>  
 <li><a href="#">Something else here</a></li>  
 <li class="divider"></li>  
 <li><a href="#">Separated link</a></li>  
 </ul>  
 </li>  
 </ul>  
 </div><!-- /.navbar-collapse -->  
 </div><!-- /.container-fluid -->  
</nav>

<nav class="navbar navbar-default navbar-inverse">

<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">

<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="#">Good Coffee</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="#">Coffee<span class="sr-only">(current)</span></a></li>

<li><a href="#">Locations</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="#">Fair Trade</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false">Coffee Gear<span class="caret"></span></a>

<ul class="dropdown-menu" role="menu">

<li><a href="#">Coffee</a></li>

<li class="divider"></li>

<li><a href="#">Mugs</a></li>

<li><a href="#">T-shirts</a></li>

</ul>

</li>

</ul>

</div><!-- /.navbar-collapse -->

</div><!-- /.container-fluid -->

</nav>

Now add a new transparent class to the navbar and styling to make it transparent:

<nav class="navbar navbar-default navbar-inverse transparent">

nav.transparent {

background: rgba(0,0,0,0.5);

border: none;

}

## Footer

The page is starting to look professional! Finally, we come to our footer. Bootstrap doesn’t have a super-simple copy-paste footer, but now that we are familiar with navbars, we can emulate a little of their behavior.

First, let’s use some semantic html to add a footer with a row and 2 columns. We add special symbols in html using the ampersand (&) and semicolon (;). Note that in any project with a programming language, you should not hard-code the year! Gant will make fun of you if you do. Add style too.

<footer>

<div class="row footer">

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

<p>Good Coffee Co. &copy;2015</p>

</div>

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

</div>

</div>

</footer>

.footer {

background-color: #D64541;

}

Now we can add the links at the bottom right as a list, similar to the navbar, and then grab some of that styling by using the navbar classes for the ul (an alternative is nav-pills). Then we can override style to make it look nice.

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

<ul class="nav navbar-nav navbar-right">

<li><a href="#">About</a></li>

<li><a href="#">Locations</a></li>

<li><a href="#">Coffee</a></li>

</ul>

</div>

.footer p {

padding: 15px;

}

.footer a {

color: white;

}

.footer a:hover {

color: #323232;

}

## Bonus! Font Awesome Social Media Links

It’s not in our design template, but I promised to teach you how to use Font Awesome because it’s so awesome! Add the Font Awesome CDN to the links in your head sometime before your custom stylesheet. <http://fortawesome.github.io/Font-Awesome/icons/>

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">

If it doesn’t load, in Dev Tools, look at the console. This should be your first place to look whenever you have an error (notice the red x and 1). It can’t find the file with that relative path. Go ahead and manually change to https:.

Let’s add a twitter link to the bottom middle. First, adjust the footer to be 3 columns instead, then add Twitter and any other social media links to the middle. Adjust size by adding another [class](http://fortawesome.github.io/Font-Awesome/examples/). You can also use commas in CSS to minimize duplication.

<div class="col-sm-4 text-center">

<a href="https://twitter.com/" target="\_blank"><i class="fa fa-twitter-square fa-2x"></i></a>

</div>

.footer p, .fa {

padding: 15px;

}

## Bonus! Add Button to Jumbotron

Twitter makes awesome [buttons](http://getbootstrap.com/css/#buttons), so let’s jazz up our Jumbotron with a button to get some practice with them.

Add a button. Change the classes to see how the default colors and sizes change.

<p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>

## Bonus! Add a Custom Media Query

Make the Jumbotron button disappear when the screen width drops below 768px.

@media only screen and (max-width: 768px) {

.jumbotron a.btn {

display: none;

}

}