Navigating IDEA, CSS and Bootstrap

Today, we will start by learning how to navigate IntelliJ IDEA. Then we will be going into the basics of CSS and Bootstrap.

IntelliJ IDEA

Step 1: Install the JDK

To develop Java applications in IntelliJ IDEA, you need the Java SDK (JDK). If Java is not installed on your computer, you need to download a JDK package.

Open the jdk.java.net website. There, you'll find the Oracle's OpenJDK binaries for Windows, macOS, and Linux as .tar.gz and .zip archives.

Select and download the necessary JDK version. If you're not sure which version you need, select the latest stable version.

Unzip the archive and place the folder with Java to the Java home directory on your computer.

The default path on Windows: C:\Program Files\Java, on macOS: /Library/Java/JavaVirtualMachines.

On Linux, the default location for Java might differ depending on the distribution that you use.

Step 2: Create a new Java project

In IntelliJ IDEA, a project helps you organize your source code, tests, libraries that you use, build instructions, and your personal settings in a single unit.

Launch IntelliJ IDEA.

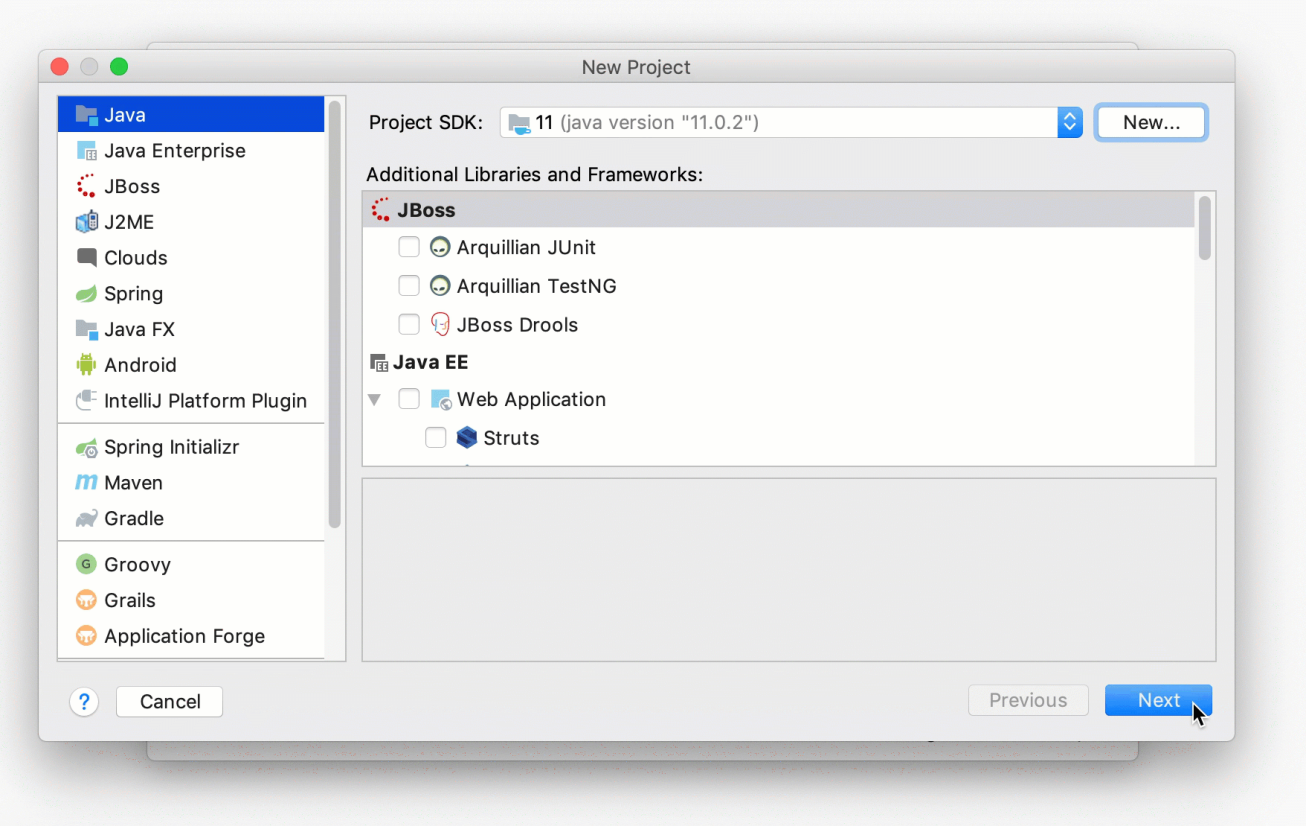
If the Welcome screen opens, click **Create New Project**.

Otherwise, from the main menu, select File | New | Project.

In the New Project wizard, select Java from the list on the left.

From the Project SDK list, select the JDK that you want to use in your project.

If the list is empty, click New and specify the path to the Java home directory (for example, jdk-11.0.0.jdk).



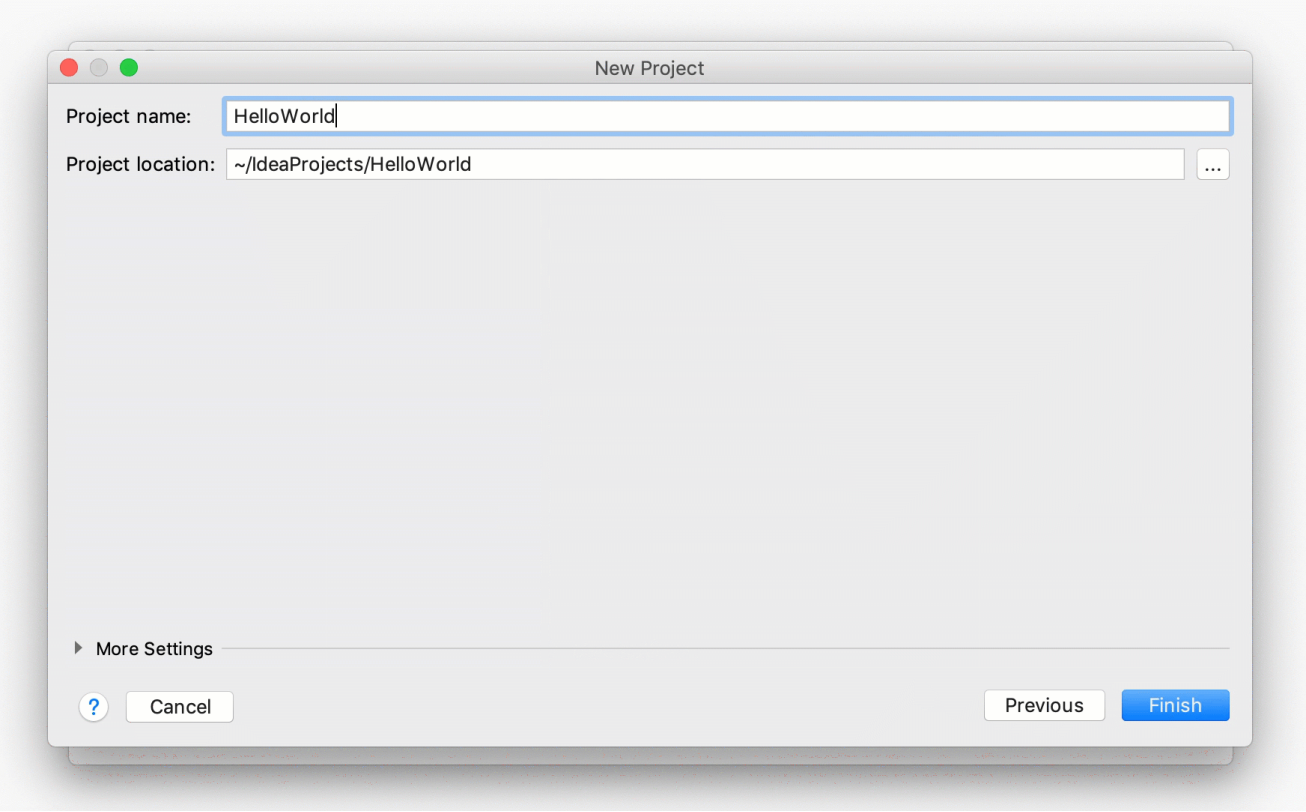
We're not going to use any additional libraries or frameworks for this tutorial, so click Next.

Don't create a project from the template. In this tutorial, we're going to do everything from scratch, so click Next.

Name the project, for example: TTSDemo.

Make the project name similar to my example above. Don't put spaces or numbers. Capitalize the name.

If necessary, change the default project location and click **Finish**.



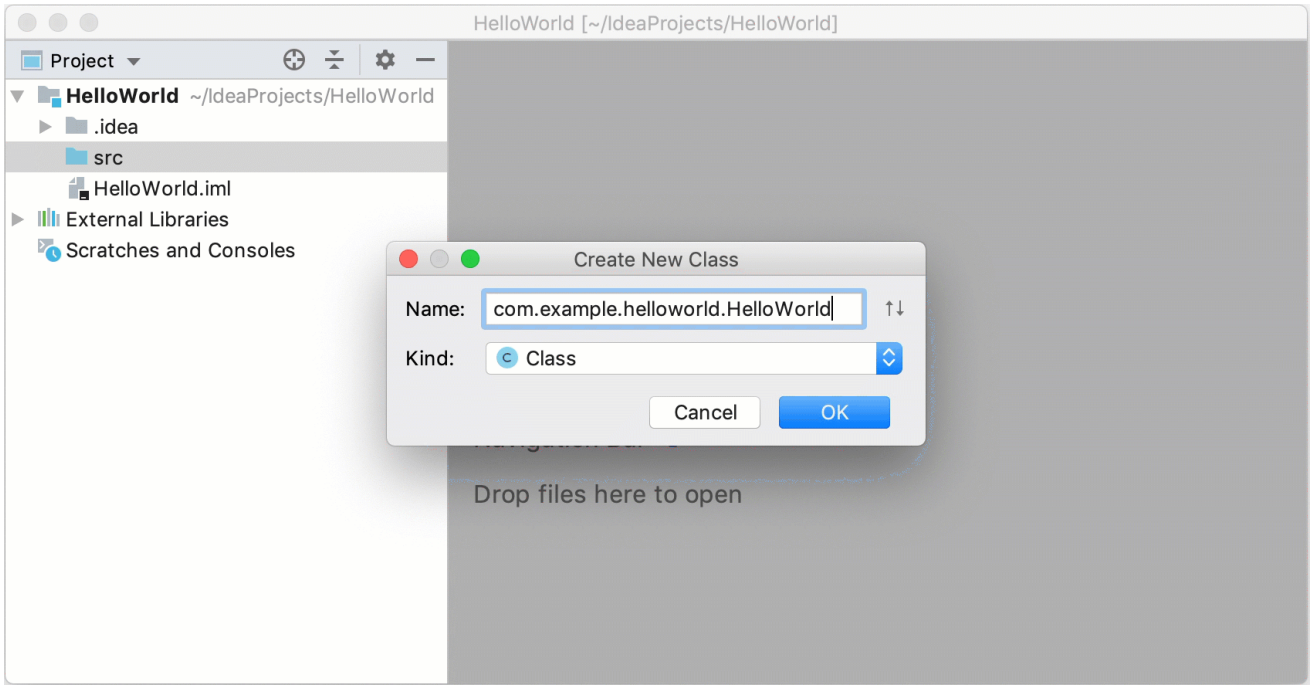
Create a **package** and a **class**

Packages are used for grouping together classes that belong to the same category or provide similar functionality, for structuring and organizing large applications with hundreds of classes.

1. In the Project tool window, select the src folder, press ⌘N, and select Java Class.
2. In the Name field, type com.example.helloworld.HelloWorld and click OK.

IntelliJ IDEA creates the com.example.helloworld package and the HelloWorld class.

Hello World is a special keyword for the developer community.



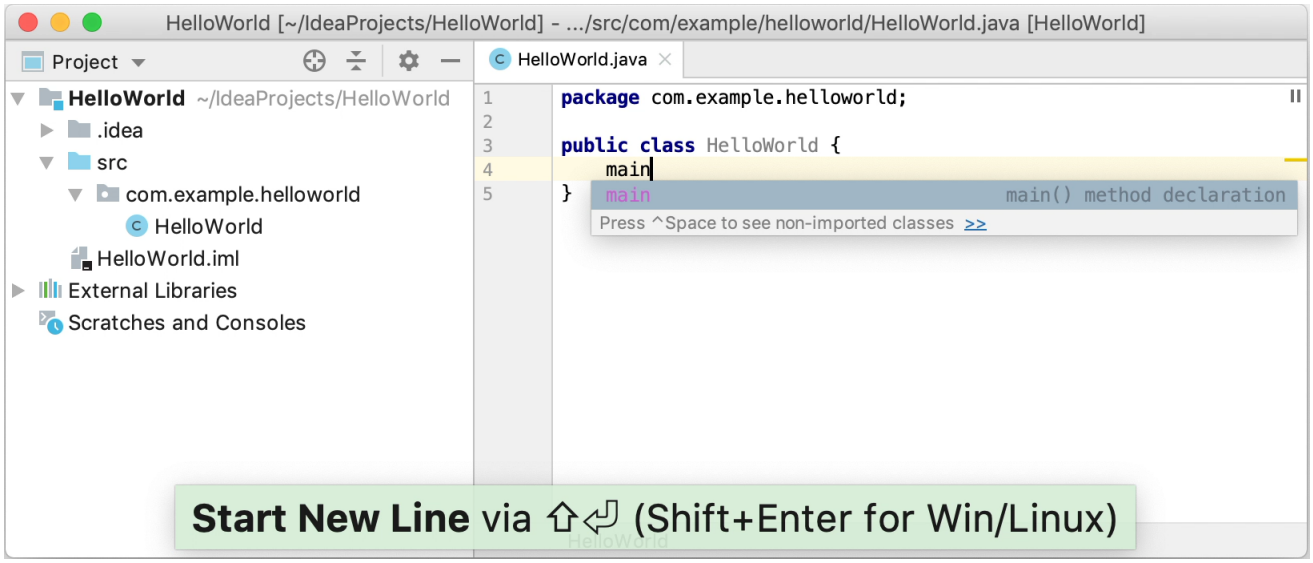
Together with the file, IntelliJ IDEA has automatically generated some contents for your class. In this case, the IDE has inserted the package statement and the class declaration.

Place the caret at the class declaration string after the opening bracket and press ⇧↵.

In contrast to ↵, ⇧↵ starts a new line without breaking the current one.

Type main and select the template that inserts the main() method declaration.

As you type, IntelliJ IDEA suggests various constructs that can be used in the current context. You can see the list of available live templates using ⌘J.

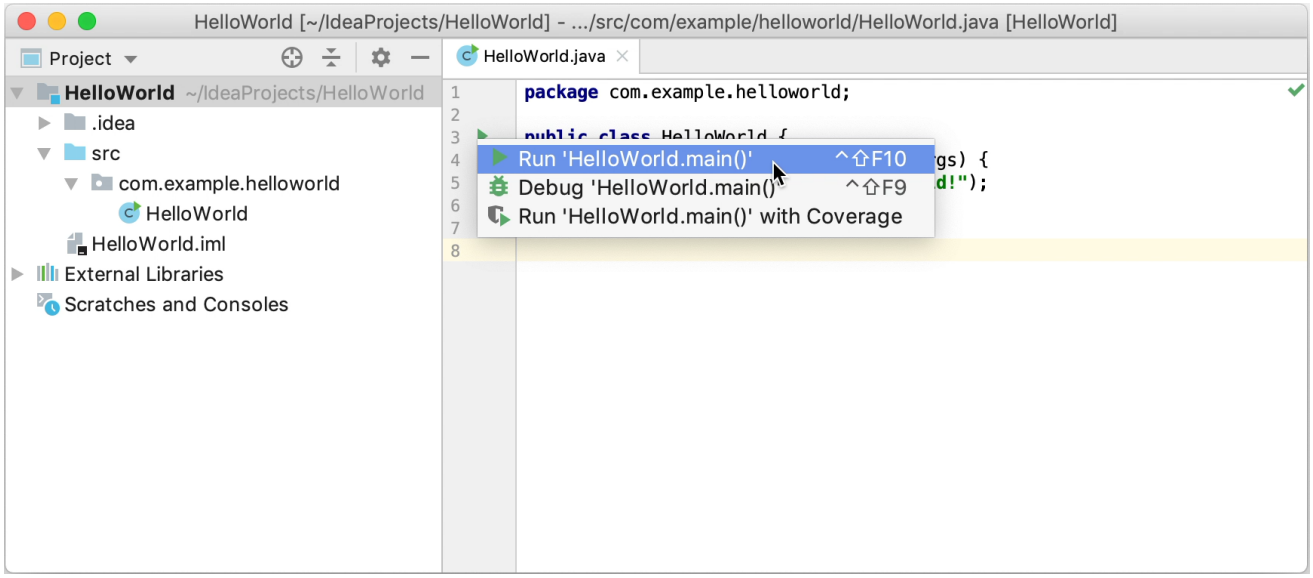


Call the println() method using code completion

After the main() method declaration, IntelliJ IDEA automatically places the caret at the next line.

Build and run the application

Valid Java classes can be compiled into bytecode. You can compile and run classes with the main() method right from the editor using the green arrow icon the Run the Remove button in the gutter.



When you click Run, IntelliJ IDEA creates a special run configuration that performs a series of actions. First, it builds your application. On this stage, javac compiles your source code into JVM bytecode.

Great! Now you should feel comfortable using IntelliJ IDEA

Note: Your development environment needs to work for **you**, but the most important thing is the code that you write in the IDE. So, let's make sure we can use IDEA, and we can become IntelliJ IDEA experts later!

Now that we have been introduced to our IDE, let's jump into CSS and then Bootstrap.

What is CSS?

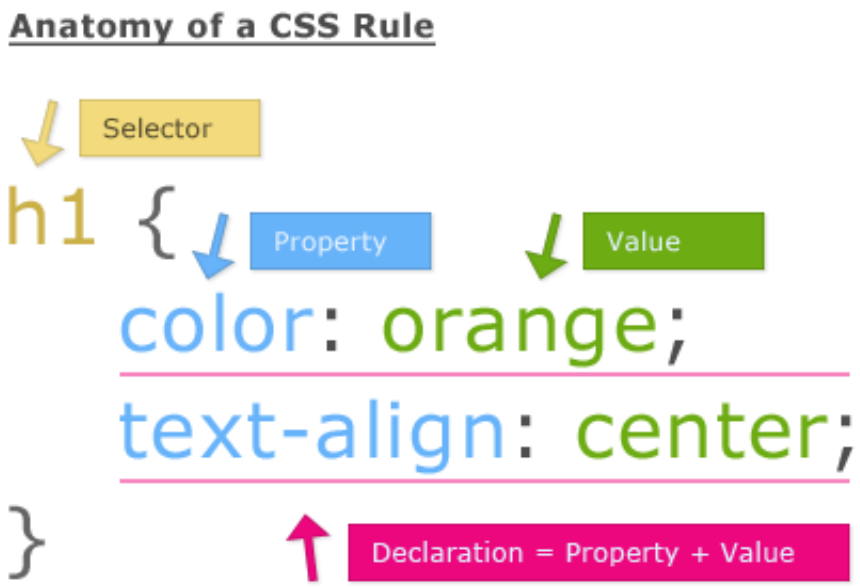
CSS or Cascading Style Sheet is what gives our content style using text color, font styles, font sizes, font families, background color, etc.

CSS can be applied to a document in several ways:

1. inline styling - where we add a style attribute within the element
2. internal style sheets - where we put style tags in the head of our HTML document and apply style rules
3. external style sheet - style is contained in a separate document, then called in to our HTML document via the link tag, where the source is the path to the document. This path can be a file path on our local environment or an HTTP address to a Content Distribution Network (CDN), like Bootstrap.

An inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the tag, or in an external style sheet, or a browser default value.

CSS Syntax



example of css:

```
h1 { color: red; }
```

Inline CSS

While inline styling is convienient for an easy fix, it is not the standard process for adding CSS to our documents. CSS, being it's own language, can crowd an HTML document. Therefore CSS belongs in CSS files

```
<div style="color: red;">
    Red Text
</div>
```

Internal / Embedded CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Test Page</title>
<!-- start of embedded CSS -->
```

```
        <style type="text/css">
    h1 {color:red;}
    p {color:blue;}
  </style>
<!-- end of embedded CSS -->
</head>
<body>
  <h1>Header</h1>
  <p>Paragraph</p>
</body>
</html>
```

External CSS

To minimize crowding the HTML document and to make sure all the code goes into it's appropriate file, external CSS is the standard for styling the HTML document.

- The correct way to include CSS in your website
- CSS is held in a separate file, and linked inside the tags in your HTML

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Test Page</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Header</h1>
  <p>Paragraph</p>
</body>
</html>
```

CSS

```
h1 {
    color:red;
}

p {
    color:blue;
    text-align: center;
}
```

CSS Rules

Element Selector

The element selector targets which HTML elements to apply. Any element can be called out in the selector. All elements on the document that have that selector will be styled by that rule.

- The element selector will style elements based on the element name
- Used for html tags like h1, p, body, etc.
- This example will style all paragraphs in the website

HTML

```
<p>
    Paragraph 1 is blue
</p>
<p>
    Paragraph 2 is also blue
</p>
```

CSS

```
p {
    color: blue;
    text-align: center;
}
```

Class Selector

Classes allow the use of CSS selectors to be more specific by styling elements based on the class they belong to. This allows for using a rule to style a paragraph, but not all paragraphs or only some h1 headings, but not all. A class can be used with many elements, regardless of the type of element they are.

- The class selector will style elements based on the class name
- Used many times in a page
- Assign a class name to an HTML tag to style it
- Most common selector in CSS

HTML

```
<h1 class="red-text">Header is red</h1>

<p class="blue-text">Paragraph is blue and large</p>
```

CSS

```
.red-text {
    color: red;
}

.blue-text {
    color: blue;
    font-size: 24px;
}
```

ID Selector

IDs are similar to classes, in that it allows style rules to be more specific. Unlike classes, a ID can only be used once and can not be called within different elements.

- The ID selector will style elements based on the ID of the tag and NOT the class
- Used when you want to style a specific element
- Assign an ID name to an HTML tag to style it
- Used once per page only
- Uses a hash # instead of a period .

HTML

```
<p id="big-and-green">
    Paragraph is only using ID
</p>

<p id="big-and-green" class="underline-text">
    Paragraph uses class and ID
</p>
```

CSS

```
#big-and-green {
    color: green;
    font-size: 32px;
}



#big-and-green2 {
    color: green;
    font-size: 32px;
}

.underline-text {
    text-decoration: underline;
}
```

CSS Comments

Just as in HTML, comments in CSS are ignored by the browser and are useful for adding explanation in the code.

How to write a comment:

/* this is a comment */

```
/* CSS Comments */

.red-class {
    color: red; /* this will turn the content red */

/* This class is ignored completely
.blue-class {
    color: blue;
}
*/
```

Common CSS Properties

While there are many CSS styling properties, the most used properties are those that style color, element size, and document alignment:

Refer to this page for all the properties!

http://www.w3schools.com/cssref/

Color

```
.red-text-1{
    color: red;
}

.red-text-2{
    color: #ff0000;
}

.red-text-3{
    color: rgb(255,0,0);
}
```

Size

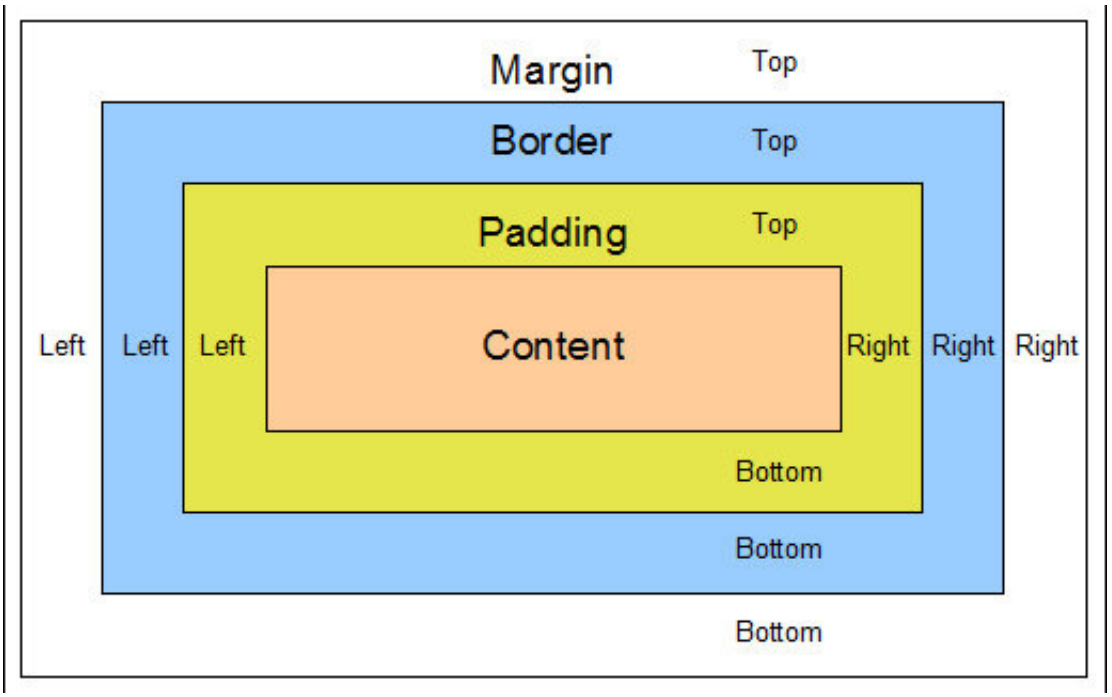
```
.wide{
    width: 100%;
}

.tall{
    height: 1000px;
}

.border-sizing{
    border: 20px; solid #ff0000;
}
```

Alignment

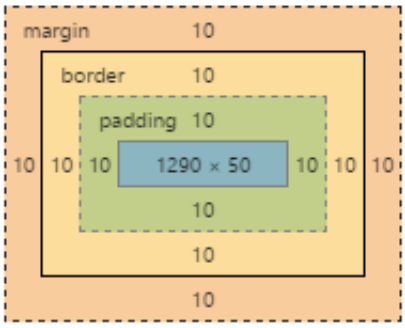
```
.left { text-align: left; }
.center { text-align: center; }
.right { text-align: right; }
```

Margin and Padding

Margins and padding are used to create white space around our content. Margins determine how much space is between out elements, while padding determines how much space is around the content of our element.



```
.margin-padding {
    background-color: #FFCC9A;
    padding: 10px;
    border: 10px solid #99CDFF;
    margin: 10px;
    text-align: center;
}
```



Chrome Inspector

- **Margin:** Add space around an element, outside of its border
- **Padding:** Adds space around an element, inside of the border
- Can specify top, right, bottom, and left separately

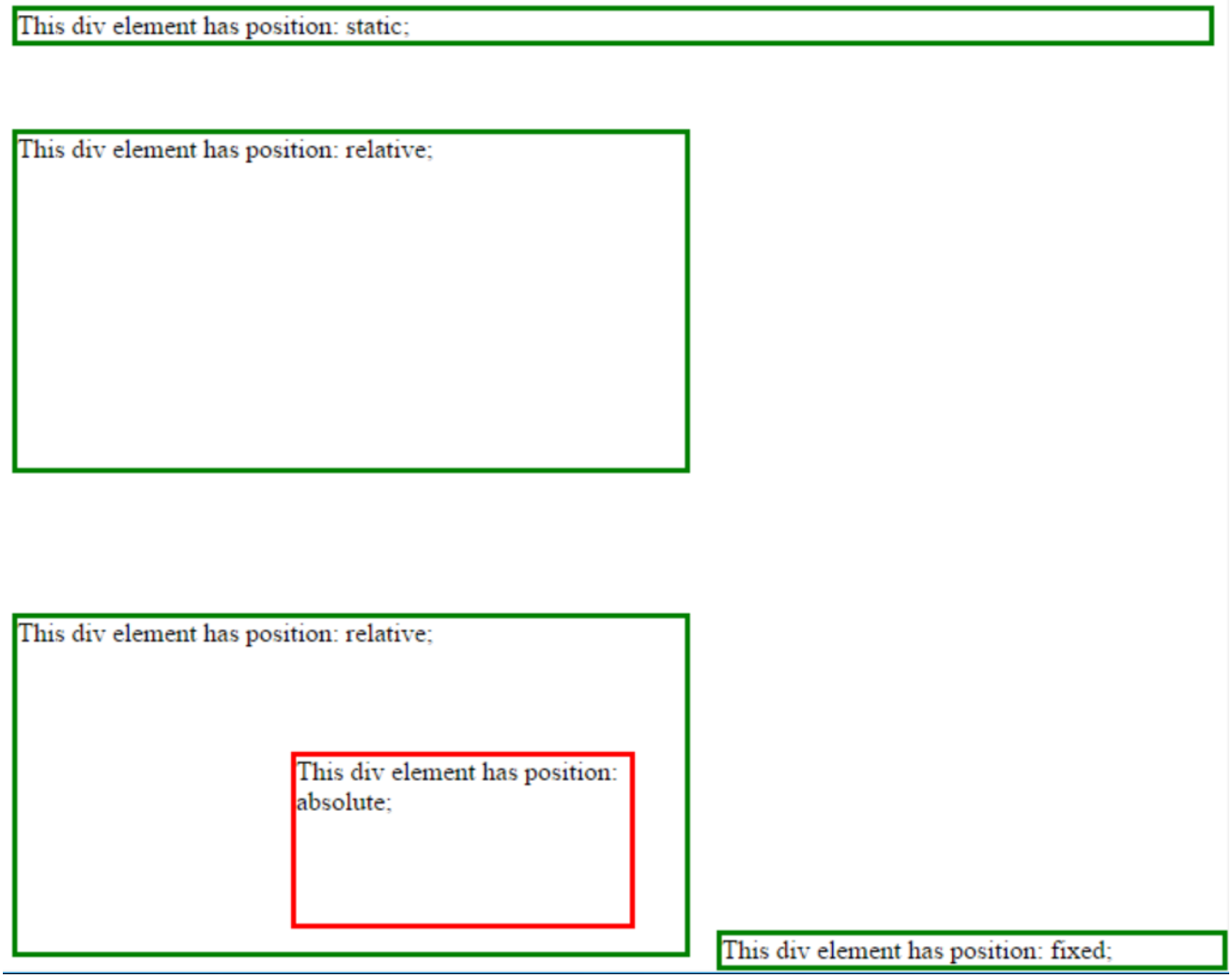
Positioning

Positioning allows us to position elements based on the relationship and positioning of other elements in our document. The default position is **static** which means it is positioned according to the normal flow of the HTML document. Position **relative** means the element will be positioned relative to it's normal position on the HTML document. Position **fixed** always stays in the same place, even if the page is scrolled. Position **absolute** positions an element relative to the nearest positioned ancestors. If an **absolute** positioned element doesn't have any positioned ancestors, it uses the document body, and moves along with page scrolling .

We can position an element 4 different ways:

- Static
- Relative
- Fixed
- Absolute

Here is an example:

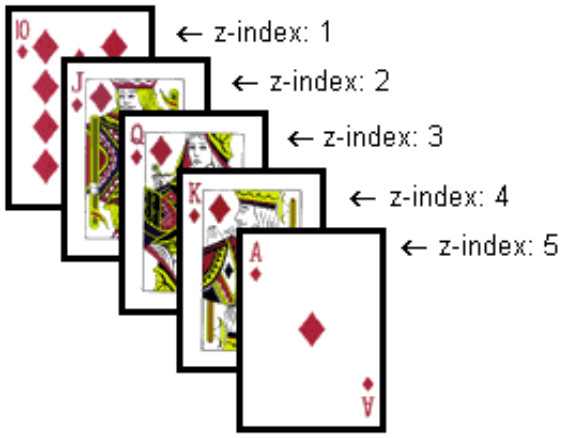



```
.static {
    position: static;
    border: 3px solid green;
}

.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid green;
}

.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid green;
}
```

```
.absolute {
    position: absolute;
    top: 80px;
    right: 30px;
    width: 200px;
    height: 100px;
    border: 3px solid #ff0000;
}
```

Overlapping with z-index

- We can stack elements on top of each other using z-index
- z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).



Bootstrap

Pre-Built CSS & JS Framework

Let's take a look at using Bootstrap's library of CSS and JavaScript classes and id's. Though it's good to have thorough Front End skills under your belt, if you need to get an app out of production quick, Bootstrap can help cut down the time you may spend on UI/UX.

Concept Review

Basic layout of an HTML page:

```
    <!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="assets/stylesheets/styles.css">
  </head>
  <body>
    <h1>My Awesome Site</h1>
    <p>You'll find some really cool stuff here.</p>
  </body>
</html>
```

Review: Divs and Spans

`<div> and <spans>` are HTML tags that are commonly used as a container for other HTML elements in order to style them with CSS or to perform certain tasks with Javascript. `<div>` tag is a type of a “Block Element” (They always start on a new line and take up the full width available). They are also used for large chunks of code.

` tag is a type of an “Inline Element” (They do not start on a new line and only takes up as much width as necessary). They are also used for small bits of code.

```
<div id="content">
    <h1>My Awesome Site</h1>       `
```

`<p>`You'll find some really cool stuff here.

```
#content {
  width: 600px;
  background: #fff;
}

.crazy-color {
  color: fuchsia;
}
```

Each HTML element can be considered as boxes. The box model is used when speaking of designs and layouts thus it is essentially a box that wraps around HTML elements. It consists of: margins, borders, padding, and the actual content.

Below is an explanation of the different parts:

- **Content** - The content of the box, where text and images appear.
- **Padding** - Clears an area around the content. It is transparent.
- **Border** - A border that goes around the padding and content.

- **Margin** - Clears an area outside the border. It is transparent.

Bootstrap

What is Bootstrap?

Bootstrap is a free front-end framework for faster and easier web development. It includes HTML and CSS based design templates as well as JavaScript plugins. It was built by developers at Twitter. It is mobile-ready and built with responsive design in mind.

Basically, it saves you lots of **time**! You can use their pre-built classes and ids to give your site a sleek look without having to write a whole lot of your own CSS and JavaScript.

Built w/ Bootstrap Examples:

Timely

Osmo

Bootstrap CDN

If you don't want to download and host Bootstrap yourself, you can include it from a CDN (Content Delivery Network).

MaxCDN provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery:

Here is an example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js">
</script>
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

```
</body>
</html>
```

How To Download Bootstrap

Method 1

It is very easy to setup and start using Bootstrap. You can download the latest version of bootstrap from here.

- **Download Bootstrap**: Clicking this, you can download the precompiled and minified versions of Bootstrap CSS, JavaScript, and fonts. No documentation or original source code files are included.
- ** Download Source**: Clicking this, you can get the latest Bootstrap LESS and JavaScript source code directly from GitHub.

If you work with Bootstrap's uncompiled source code, you need to compile the LESS files to produce usable CSS files. For compiling LESS files into CSS, Bootstrap officially supports onlyRecess , which is Twitter's CSS hinter based on less.js.

For better understanding and ease of use, we shall use compiled version of Bootstrap throughout. Since the files are complied and minified you don't have to bother every time including separate files for individual functionality. Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure.

- Bootstrap/
- Css/
- Js/
- fonts/

As you can see there are compiled CSS and JS (bootstrap.*), as well as compiled and minified CSS and JS (bootstrap.min.*) inside the folders. Fonts from Glyphicons are included, as is the optional Bootstrap theme.

BOOTSTRAP SOURCE CODE

If you downloaded the Bootstrap source code then the file structure would be as follows:

- The files under less/, js/, and fonts/ are the source code for Bootstrap CSS, JS, and icon fonts (respectively).
- The dist/ folder includes everything listed in the precompiled download section an above.
- docs-assets/, examples/, and all *.html files are Bootstrap documentation.

  Method 2

1. Create a new folder in where you are collecting your Front End projects. Name “Bootstrap Project ”.
2. Open the folder in Sublime.
3. Inside create, a new file called “index.html”.
4. In the browser, go to Bootstrap’s Basic Template. Copy-and-paste (use the Copy button on the top-right!)

into index.html, and get rid of any extra stuff (comments, etc.)
5. Test it out.

Bootstrap isn't coming through yet, but we want to see the difference between normal and Bootstrap. Next, let's bring in the Bootstrap library through CDN (Content Delivery Network) links:

1. Find CDN links at the top of Bootstrap's "Getting Started" page.
2. Copy the CSS link and paste over the dummy tag in index.html
3. Copy the JS link and paste over the dummy.

A Quick Look at Bootstrap Components

Let's add to our index.html page a couple of common Bootstrap components, then we will look at them in more detail.

Jumbotron As the name suggests this component can optionally increase the size of headings and add a lot of margin for landing page content. To use Jumbotron:

- Simply create a container `<div>` with the class of **.jumbotron**
- In addition to a larger `<h1>`, the font-weight is reduced to 200px.

To get a jumbotron full width, and without rounded corners use the **.jumbotron** class outside all **.container** classes and instead add a .container within as shown in the following example:

```
<div class="jumbotron">
<div class="container">
  <h1>My website</h1>
  <p>My name is Aaron Groch.</p>
  <p><a class="btn btn-primary btn-lg" role="button">Click this!</a>
</div>
</div>
```

Bootstrap Grid System

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

- creates a page layout through a series of rows and columns
- columns will rearrange depending on the screen size
- allows up to 12 columns across the page.

Here's an example of a Basic Structure of a Bootstrap 4 Grid:

```
<!DOCTYPE html>
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js">
</script>
```

```
</head>
<body>

      <div class="container-fluid">
  <h1>Basic Grid Structure</h1>
  <p>Resize the browser window to see the effect.</p>
  <p>The first, second and third row will automatically stack on top of each other when
the screen is less than 576px wide.</p>
```

50%

50%

```
    <div class="row">
          <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
<div class="col-sm-4" style="background-color:orange;">33.33%</div>
          <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
    </div>
<br>
<!-- Or let Bootstrap automatically handle the layout -->
    <div class="row">
          <div class="col-sm" style="background-color:yellow;">25%</div>
      <div class="col-sm" style="background-color:orange;">25%</div>
      <div class="col-sm" style="background-color:yellow;">25%</div>
          <div class="col-sm" style="background-color:orange;">25%</div>
</div>
        <br>

        <div class="row">
          <div class="col" style="background-color:yellow;">25%</div>
      <div class="col" style="background-color:orange;">25%</div>
      <div class="col" style="background-color:yellow;">25%</div>
          <div class="col" style="background-color:orange;">25%</div>
```

```
        </div>
```

Working of Bootstrap Grid System Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works:

- Rows must be placed within a .container class for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts. Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.

The Bootstrap Grid Container We create a grid container by declaring display: grid or display: inline-grid on an element. As soon as we do this, all direct children of that element become grid items. In this example, I have a containing div with a class of wrapper and, inside are five child elements.

```
<div class="wrapper">
<div>One</div>
<div>Two</div>
<div>Three</div>
<div>Four</div>
<div>Five</div>
</div>
```

I make the .wrapper a grid container. .wrapper { display: grid; }

All the direct children are now grid items. In a web browser, you won't see any difference to how these items are displayed before turning them into a grid, as grid has created a single column grid for the items.

At this point, you may find it useful to work with the Grid Inspector, available as part of Firefox's Developer Tools. If you view this example in Firefox and inspect the grid, you will see a small icon next to the value grid. Click this and then the grid on this element will be overlaid in the browser window.

As you learn and then work with the CSS Grid Layout this tool will give you a better idea of what is happening with your grids visually.

If you want to start making this more grid-like you need to add column tracks.

Grid Challenge

Give students 5 minutes to complete. Ask to raise hands or slack you when complete and check their work.

- Add a new row with 3 columns
- Hint: use class="col-md-4"

Possible Solution:

```
<div class="row">
  <div class="col-md-4">
<p>Authentic typewriter you probably haven't heard of them normcores paleo.</p
  </div>
  <div class="col-md-4">
        <p>Authentic typewriter you probably haven't heard of them normcore Ugh mi</p>
  </div>
  <div class="col-md-4">
```

<pAuthentic typewriter you probably haven't heard of them normcore

Bootstrap Panels

Panel component are used when you want to put your DOM component in a box. To get a basic panel, just add class .panel to the element. Also add class .panel-default to this element as shown below.

Panels are made up of several classes attributed to nested tags. There is the “panel” class, plus color choice in “panel-”. Then the “panel-body” would hold your main content. You can also add a “panel-heading” (above “panel-body”), and add a “panel-footer” (below “panel-body”).

```
<div class="panel panel-default">
  <div class="panel-heading">
      My Article Title
  </div> <!-- closing of panel heading -->
  <div class="panel-body">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt<super>*</super> ut labore
    et dolore magna aliqua. Ut enim ad minim veniam.
  </div> <!-- closing of panel body -->
  <div class="panel-footer">
    <super>*</super>No, not really.
  </div> <!-- closing of panel footer -->
</div> <!-- closing of panel class -->
```

Panel Challenge Give students 5-10 minutes to complete. Ask to raise hands or slack you when complete and check their work.

- Make a new row with three columns
- Create a panel in each row w/ heading, body & footer
- Make them colorful!
- Let me know when you’re finished
- Share with class

Possible Solution:

```
<div class="row">
  <div class="col-md-4">
    <div class="panel panel-primary">
      <div class="panel-heading panel-primary">Heading 1</div>
      <div class="panel-body">
        `<p>`Authentic typewriter you probably haven't heard of them normcore, sustainable hoodie Thundercats heirloom squid craft beer Schlitz. Ugh biodiesel Pitchfork mixtape High Life. Put a bird on it selvage bicycle rights paleo.`</p>`
      </div>
    </div>
  </div>
  <div class="col-md-4">
    <div class="panel panel-success">
      <div class="panel-heading">Heading 2</div>
      <div class="panel-body">
        <p>Authentic typewriter you probably haven't heard of them normcore, sustainable hoodie Thundercats heirloom squid craft beer Schlitz. Ugh biodiesel Pitchfork mixtape High Life. Put a bird on it selvage bicycle rights paleo.</p>
      </div>
    </div>
  </div>
  <div class="col-md-4">
    <div class="panel panel-warning">
      <div class="panel-heading">Heading 3</div>
      <div class="panel-body">
        <p>Authentic typewriter you probably haven't heard of them normcore, sustainable hoodie Thundercats heirloom squid craft beer Schlitz. Ugh biodiesel Pitchfork mixtape High Life. Put a bird on it selvage bicycle rights paleo.</p>
      </div>
    </div>
  </div>
</div>
```

Bootstrap Wells

A well is a container `<div>` that causes the content to appear sunken or an inset effect on the page. To create a well, simply wrap the content that you would like to appear in the well with a containing the class of .well. (All you get is a grey box with a grey border surrounding your content)

```
<div class="well">
<p>
    Ten years ago a crack commando unit was sent to prison
    by a military court for a crime they didn’t commit.
    These men promptly escaped from a maximum security stockade to the Los Angeles
underground. Today, still wanted by the government, they survive as soldiers of fortune.
  </p>
</div>
```

This also some size variation for wells.

```
<div class="well well-lg">
    <p>
    Kinda looks the same as just the standard well, really.
  </p>
</div>
<div class="well well-sm">
  <p>
    I'm a little well div, short and stout. Here's my grey
    background, here's my grey border.
  </p>
</div>
```

Lab Activities

Well Challenge Give students 5 minutes to complete. Ask to raise hands or slack you when complete and check their work.

- Build a new row, this time with just 2 columns
- Surround content in each column with a “well” class

Possible Solution:

```
<div class="row">
  <div class="col-md-6">
    <div class="well">
        <p>
        Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi
pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber
earthnut pea peanut soko zucchini.
      </p>
    </div>
  </div>
  <div class="col-md-6">
```

```
    <div class="well">
      <p>
        Celery quandong swiss chard chicory earthnut pea potato. Salsify taro catsear
garlic gram celery bitterleaf wattle seed collard greens nori. Grape wattle seed kombu
beetroot horseradish carrot squash brussels sprout chard.
      </p>
    </div>
  </div>
</div>
```

Bootstrap Button Groups

Button groups allow multiple buttons to be stacked together on a single line. This is useful when you want to place items like alignment buttons together. You can add on optional JavaScript radio and checkbox style behavior with Bootstrap Button Plugin.

You can add Bootstrap button (“btn”) classes to nearly any HTML tag (though results may vary). Generally, though, we add to or tags. Add a button to your 3rd column:

```
<a href="#" class="btn btn-success">Sign Up Now!</a>
```

The “btn” classes will not alone create a button. You must also include the style/color of the button.

The choices are: *“btn-default” (white) *“btn-primary” (dark blue) *“btn-info” (light blue) *“btn-warning” (orange) *“btn-danger” (red) *“btn-success” (green)

You can also change the size of the button with: *“btn-lg” (large) *“btn-sm” (small) *“btn-xs” (really small!)

Footer Challenge

Give students 5 minutes to complete. Ask to raise hands or slack you when complete and check their work.

*Add a footer *Use a 3 column layout (again) *Include copyright, navigation, follow me

Possible Solution:

```
<footer>
  <div class="container">
    <div class="row">
      <div class="col-md-4">© 2014</div>
      <div class="col-md-4">
        <ul class="nav nav-pills">
          <li><a href="#">Contact Us </a></li>
          <li><a href="#">Get Support</a></li>
          <a href="#">Privacy Policy</a></li>
        </ul>
      </div>
      <div class="col-md-4">
```

```
          <p class="text-right"><a href="#">Follow Us</a></p>
      </div>
    </div>
  </div>
</footer>
```

Note that copyright symbol could just be copy-n-pasted in, or you can use the proper Number (©) or Entity (©).

W3Schools has a good list of Symbols for HTML.

Bootstrap Glyphicons

Glyphicons are icon fonts which you can use in your web projects. Glyphicons Halflings are not free and require licensing, however their creator has made them available for bootstrap projects free of cost.

Where to find Glyphicons? They can be found within the fonts folder. This contains the following files:

- glyphicons-halflings-regular.eot
- glyphicons-halflings-regular.svg
- glyphicons-halflings-regular.ttf
- glyphicons-halflings-regular.woff

Associated CSS rules are present within bootstrap.css and bootstrap-min.css files within css folder oflist folder.

Glyphicons are just Bootstrap's fancy name for "icons" - those little pictures that often precede or proceed text, or maybe replace text altogether, as a representation of what is meant (e.g., having an envelope icon instead of the phrase "Email us").

Glyphicons comes through as classes. Like the Bootstrap buttons, these are dependent on two classes: "glyphicon" and "glyphicon-", where the hyphen is proceeded by a certain glyphicon name (see the docs).

You can house these classes in either a tag (traditional choice), or an tag (as some developers have recently taken to doing, as italics was moved to the tag). Example, within our Footer:

```
<li>
  <li>
```

Contact Us

Font Awesome is another place to get icons (and with a much greater selection to choose from). You go have to request a CDN link, however - it is no longer available right on their site.

Bootstrap Images Bootstrap has some sweet classes for your images. For this reason, it provides three classes that can be used to apply some simple styles to images:

- .img-rounded: adds border-radius:6px to give the image rounded corners.
- .img-circle: makes the entire image round by adding border-radius:500px.
- .img-thumbnail: adds a bit of padding and a gray border:

The “img-responsive” class makes your images change its size relative to the browser size.

```
<img src="assets/images/llamas.jpg" class="img-responsive" alt="llamas!" />
```

There are also classes that can change the overall shape of your image:

```
<img src="assets/images/llamas.jpg" alt="llama rounded" class="img-rounded">
<img src="assets/images/llamas.jpg" alt="llama circle!" class="img-circle">
<img src="assets/images/llamas.jpg" alt="llamas w/ nice border" class="img-thumbnail">
```

Bootstrap Navbar

The Navbar is a nice feature, and is one of the prominent features of Bootstrap sites. However, it is a navigation header that is placed at the top of the page. Navbars collapse in mobile views and become horizontal as the available viewport width increases.

At its core, the navbar includes styling for site names and basic navigation. All the social media sites have one. Bootstrap makes them easy.

To create a default navbar:

- Add classes .navbar, .navbar-default to the `<nav>` tag.
- Add role="navigation" to the above element, to help with accessibility.
- Add a header class .navbar-header to the `<div>` element. Include an element with classnavbar-brand.

This will give the text a slightly larger size.

- To add links to the navbar, simply add an unordered list with a classes of .nav, .navbar-nav.

Following example demonstrates this:

```
<nav class="navbar navbar-default" role="navigation">
  <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</a></li>
        <li><a href="#">Link</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">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">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>
```

Navbar components: *Brand (usually link to home/index page) *Dropdowns *Active Class (denotes which page you're currently on) *Inline Form (we would need RoR or JS to power) *Placeholder (text within input fields) You can pick and choose from their main example, you probably won't need every aspect of a Bootstrap navbar. But don't delete the navbar-header or remove "navbar-collapse" from its . These power the responsive design of the navbar, which will gradually put items (from the right) into what is called a "hamburger menu", as the browser size decreases. (Try it out!) If Dropdowns are not working, check to make sure jQuery and Bootstrap JS CDN links are written properly.

Media Object

These are abstract object styles for building various types of components (like blog comments, Tweets, etc) that feature a left- or right-aligned image alongside textual content.

The goal of the media object is to make the code for developing these blocks of information drastically shorter.The goal of media objects (light markup, easy extendability) is achieved by applying classes to some simplemarkup. There are two forms to the media object:

- .media: This class allows to float a media object (images, video, audio) to the left or right of a content block.
- .media-list: If you are preparing a list where the items will be part of an unordered list, use class. Useful for comment threads or articles lists.

Add a media object into your second column panel:

```
<div class="media">
  <a class="pull-left" href="#">
    <img class="media-object" src="assets/images/daschund.jpg" alt="Doxie">
  </a>
  <div class="media-body">
    <h4 class="media-heading">Mini Dachshunds are my favorite</h4>
    <p>
        <p>
```

Authentic typewriter you probably haven't heard of them normcore,

sustainable hoodie Thundercats heirloom squid craft beer Schlitz.

Ugh biodiesel Pitchfork mixtape High Life. Put a bird on it selvage

bicycle rights paleo.

Responsive Embed

These classes allow browsers to determine video or slideshow dimensions based on the width of their containing block by creating an intrinsic ratio that will properly scale on any device. Find a video on YouTube and add it to the first column panel:

```
<div class="embed-responsive embed-responsive-16by9">
  <iframe src="http://www.youtube.com/embed/jR4lLJu_-wE" class="embed-responsive-item">
  </iframe>
</div>
```

Div-ception Challenge

Using Bootstrap to development is a shortcut, but that doesn't mean it can't seem incredibly complicated. You tend to have s within s within s to get the right layout.

We saw an example of this when working with Panels.

The challenge is to create a layout similar to the one shown in the notes. See Image here Possible Solution:

```
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <h1>Left Sidebar</h1>
      <p>
        Cut the cheese st. agur blue cheese fromage. Taleggio caerphilly bocconcini
caerphilly cheese and wine feta brie feta. Dolcelatte monterey jack melted cheese
dolcelatte cream cheese croque monsieur brie cheese triangles. Hard cheese the big cheese
squirty cheese everyone loves bavarian bergkase cow caerphilly gouda. Stinking
```

bishop cheese slices bavarian bergkase dolcelatte.

Column 1

Queso rubber cheese melted cheese. Halloumi cheesy grin smelly cheese fondue brie taleggio dolcelatte red leicester.

Column 2

Mascarpone dolcelatte cheese and biscuits camembert de normandie fondue fromage frais cheese on toast pepper jack. Goat.

Column 3

Halloumi melted cheese bocconcini. Manchego danish fontina cream cheese when the cheese comes out everybody's happy melted cheese.

Right Sidebar

Cottage cheese caerphilly jarlsberg. Monterey jack rubber cheese port-salut cheeseburger cut the cheese manchego mascarpone smelly cheese. Chalk and cheese caerphilly queso cheese and biscuits cheese and biscuits cheddar parmesan everyone loves. Port-salut squirty cheese emmental the big cheese mascarpone red leicester melted cheese taleggio. Hard cheese cheese triangles cheese and wine.

Additional Bootstrap Features

- Pagination
- Badges
- Alerts
- Progress Bar
- Carousel

Overriding Bootstrap Styles

- Add an external style sheet called “styles.css” to the “assets/stylesheets” folder.
- Link this stylesheet to index.html - make sure this link is after your Bootstrap CDN link (page is read top-to-bottom).
- Use the Inspector tool to determine the class you’d like to override.
- Write styles that take priority over Bootstrap’s styles.

Let’s say we want to change the color of

s inside the “jumbotron”. Inspecting that element tells us that the color in this Bootstrap style is inherited from a parent element.

To change this style we can define it in our own stylesheet. Inspecting again shows the Bootstrap style crossed out with our own taking priority.

```
.jumbotron h1 {
color: orange;
```

}}

Bootstrap Themes

- Bootswatch
- Wrap Bootstrap
- Start Bootstrap

Lab Activity

- Add Bootstrap to your personal site
- Add a navbar, header and footer
- Flow each page of your content into a 2 or 3 column grid

- Add an external stylesheet and write a few styles
- If there's time: add a theme!

Div-ception Challenge II: Div-ing Deeper

- Try to mimic Twitter's layout (the feed page)
- Just fake the content
- Use custom CSS to size the user pic

References:

1. https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp