Have you ever tried to build a website or webapp from scratch? If you have, you may have found it to be a lengthy and sometimes frustrating process. The end result often doesn't look as good as the initial plan and it can be difficult to achieve the desired layout and responsiveness to different screen sizes. Styling a webpage to a production level of finish can take a long time if doing everything from scratch.

This is where Bootstrap swoops in to save the day! Bootstrap is a framework that makes building a webpage quick and easy through the use of grid structures and pre-formatted widgets. It contains a vast library of styled widgets and frames that allow users to quickly build appealing webpages with very little effort.

By using pre-formatted templates for widgets, Bootstrap users save time and frustration that would be spent on styling each individual element. Bootstrap provides well-designed widget and frame templates that look good and respond appropriately without any customization. Base widgets come in a variety of flavours which can be further customized with CSS to provide unlimited and easy-to-use options. For example, on the left is a button with Bootstrap and on the right is a button without Bootstrap:

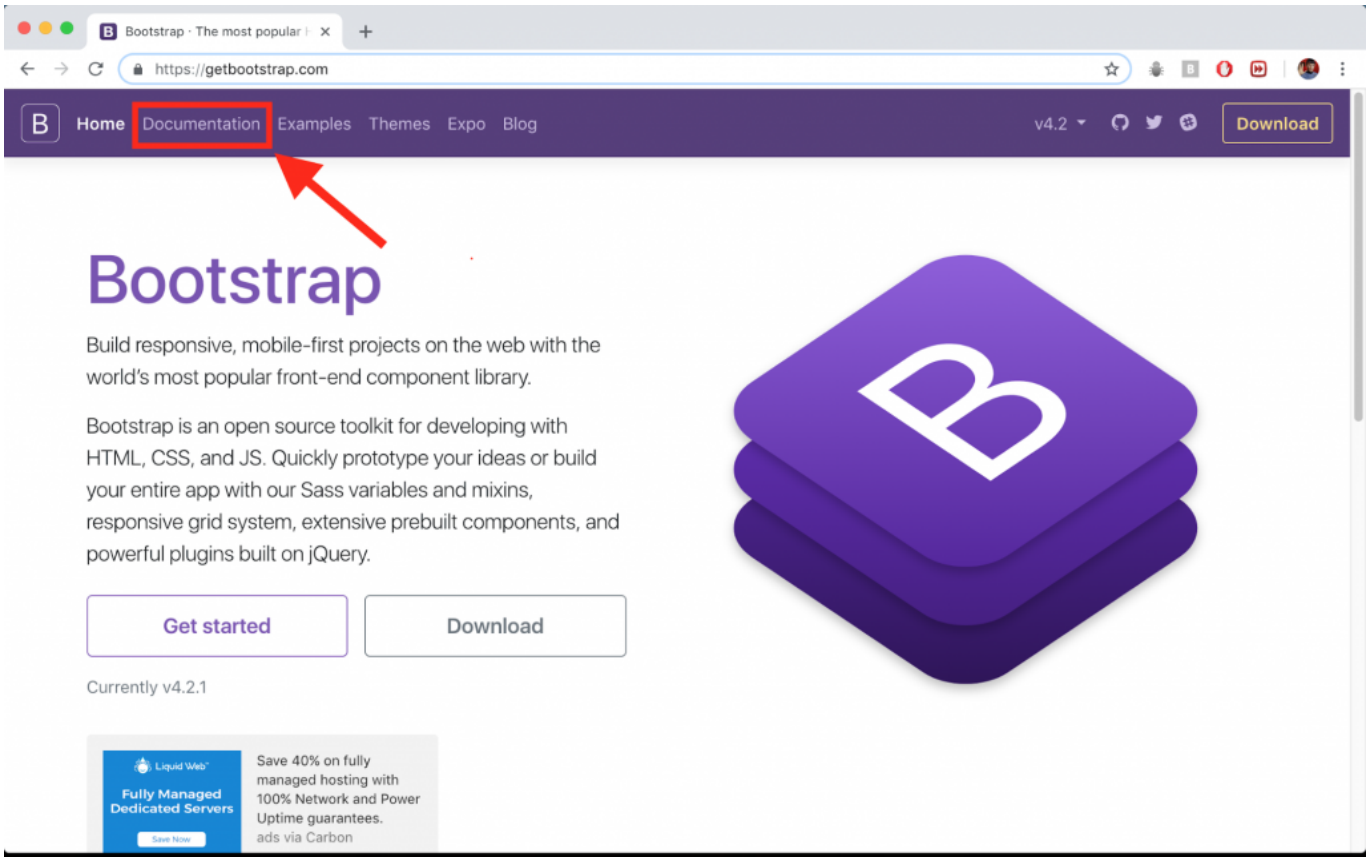
This is a noticeable improvement and requires only that we assign a bootstrap class to the button element.

Bootstrap is very simple to use. There are 5 basic steps:

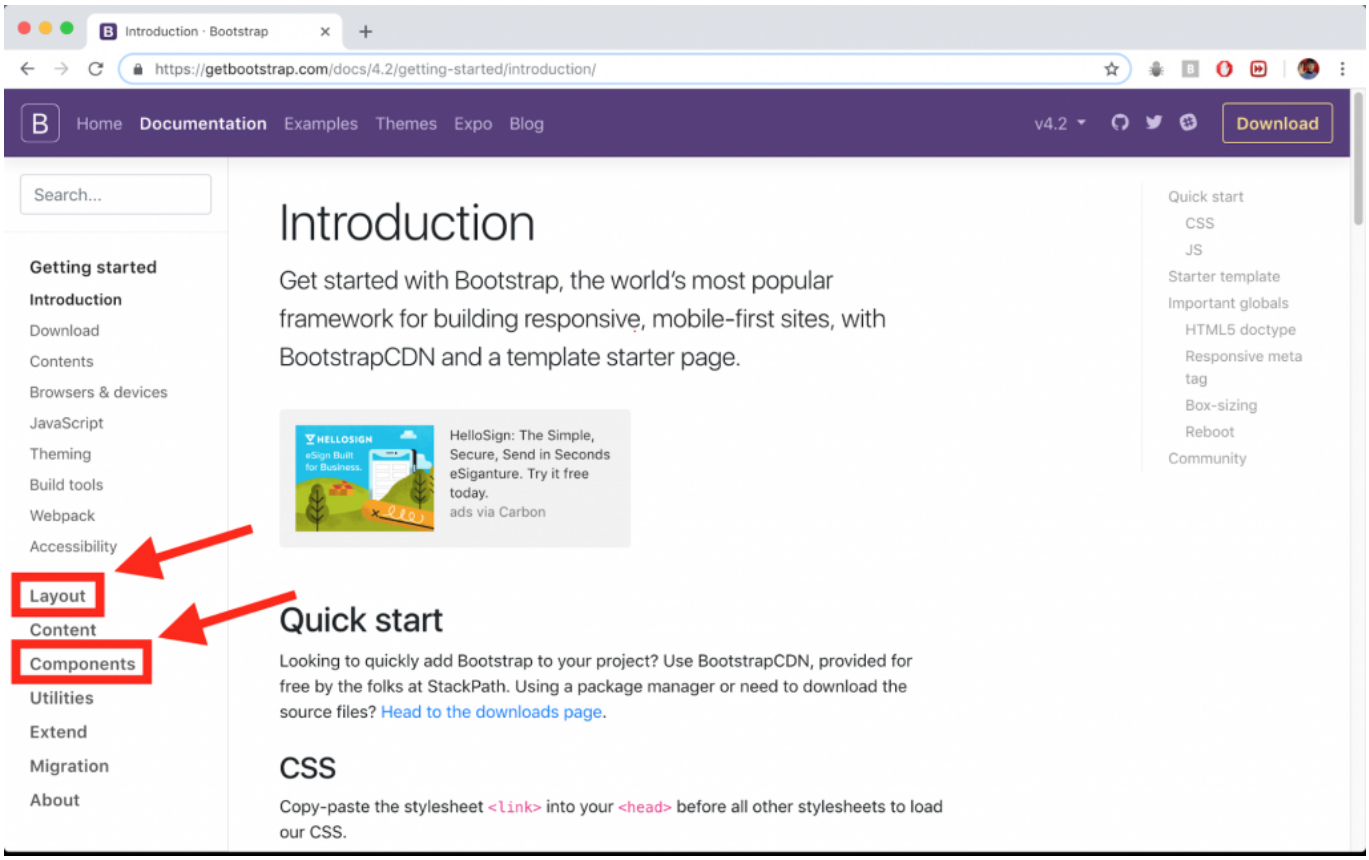
1. Add the bootstrap headers into an HTML file
2. Find the widgets and layout styles that you want on the bootstrap website
3. Copy the code for the element from the website and paste it into your HTML file
4. Add customizations from the bootstrap library to the element
5. Customize the element further with your own CSS

So now that you know what Bootstrap is, let's take a look at the webpage to get a feel for how to use it and what options are available.

An important part of using Bootstrap is getting familiar with the website. It's a copy-and-paste framework so learning how to navigate the website to find the widgets and frames that we want will save us time in the long run. Here we will take a look at the Bootstrap documentation, examine a few elements to see what the source code looks like, and then see how to add the element to our webpages. For now, we will spend most of our time in the documentation:



In the documentation, we can find the layout and frame elements under the tab "Layout" and the widgets and other components under the "Components" tab.



Let's quickly check out a few of the components to get a feel for how Bootstrap code works. Each component page gives an overview of what the element is used for and provides various examples with the source code and the output. We will examine Alerts, Buttons, and Dropdowns.

Alerts

Alerts are used to display some sort of message such as a warning or a notification. Here are a few examples of Bootstrap alerts:

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

And the source code the generate the outputs looks like this:

```
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
```



```
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
```

Note how each of the components is just a div with the class “alert” applied to it and the different alert-options give different outputs

Buttons

Buttons are used to trigger some action when pressed; almost every webpage will use buttons at some point. The basic examples look like this:

And the source code to generate the outputs looks like this:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Note that this time, the element is of type “button” and not “div” like with the alert. This time, the “btn” class is used to provide default style behaviour to the buttons with “btn-option” added to the end to further customize the button appearance.

Not all buttons are of the “button” element type. For example, radio buttons or checkboxes are considered inputs:


```
<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="radio" name="options" id="option1" autocomplete="off" checked> Activ
e
  </label>
  <label class="btn btn-secondary">
    <input type="radio" name="options" id="option2" autocomplete="off"> Radio
  </label>
```



```
  <label class="btn btn-secondary">
    <input type="radio" name="options" id="option3" autocomplete="off"> Radio
  </label>
</div>
```

Dropdowns

Dropdowns are commonly used in menu bars and provide some hidden options to choose from when a user clicks on the initial button. An expanded dropdown might look like this:

And the source code to generate this is:

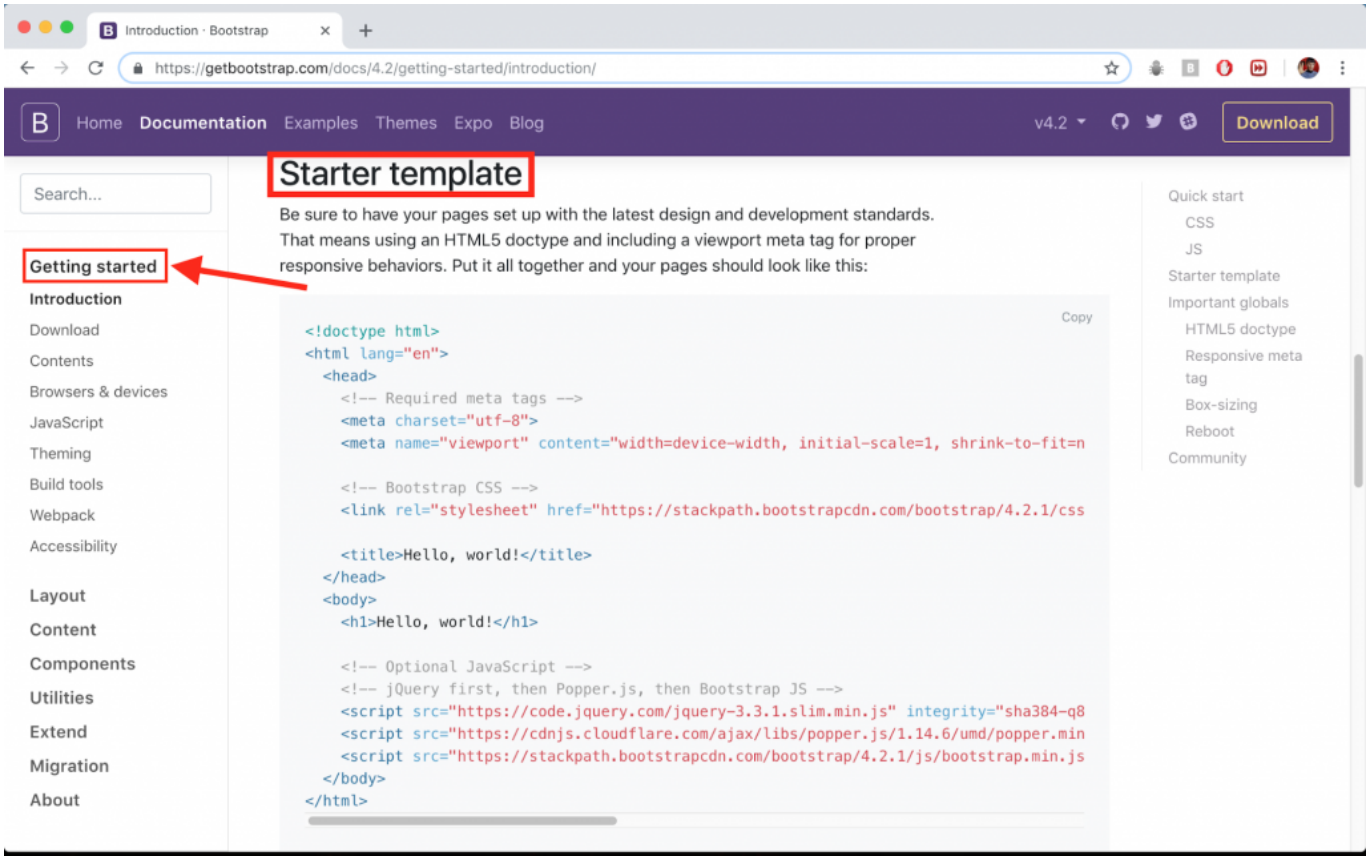
```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuBut
ton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Dropdown button
  </button>
  <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
  </div>
</div>
```

Note how this one is a set of elements enclosed in the “dropdown” class. There is the initial button and the dropdown menu with the menu options as “a” elements

So now you have an idea of how Bootstrap builds and styles the elements, let’s take a look at how to add the Bootstrap library to an HTML file.

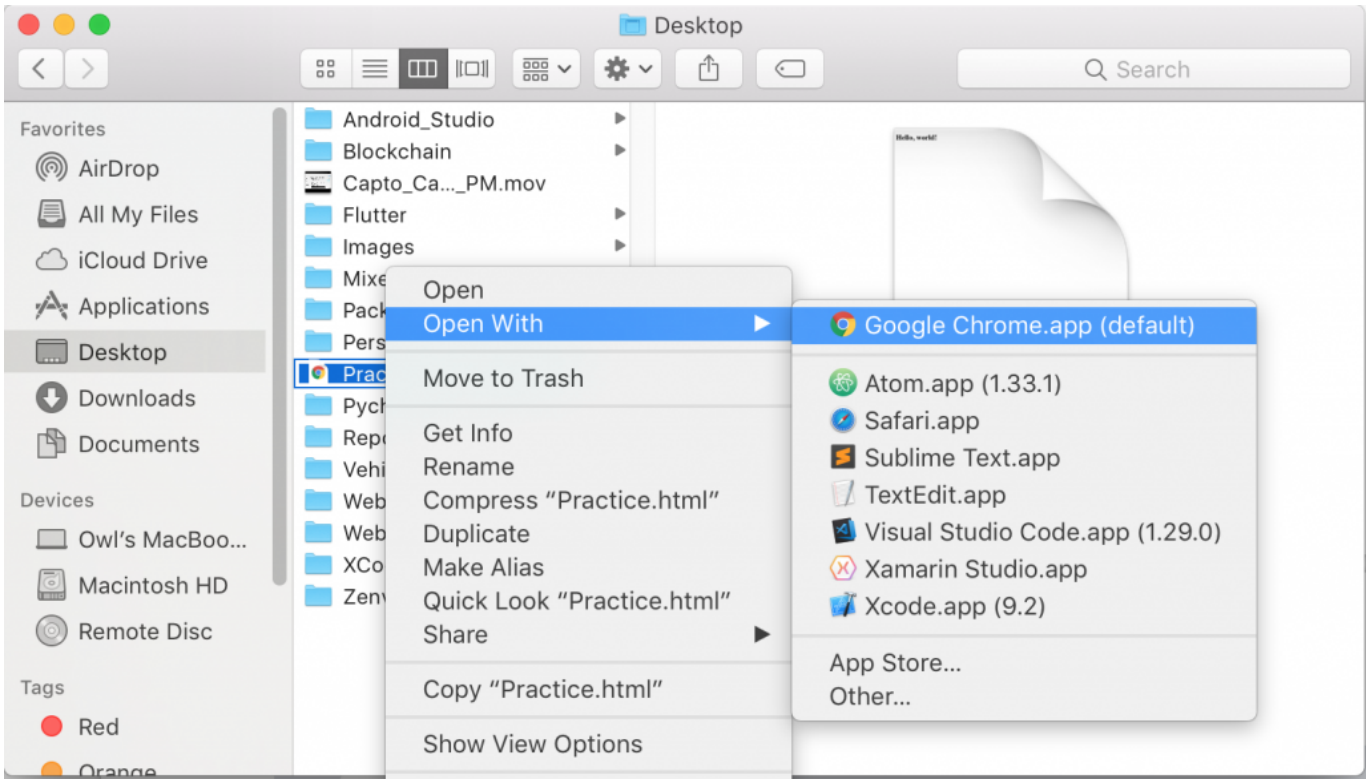
Before we understand how the grid layout works, let's start a new HTML file and add the Bootstrap headers. This will ensure that when we learn about containers, we can incorporate Bootstrap code right in our file and can see the changes happening in real time.

We'll first navigate to the Getting Started section in the Bootstrap website and copy the starter template.



And then we will paste the code in a new HTML file. I like Sublime Text so I will use that as my text editor. Note the stylesheet link in the head tag and the scripts in the body tag:

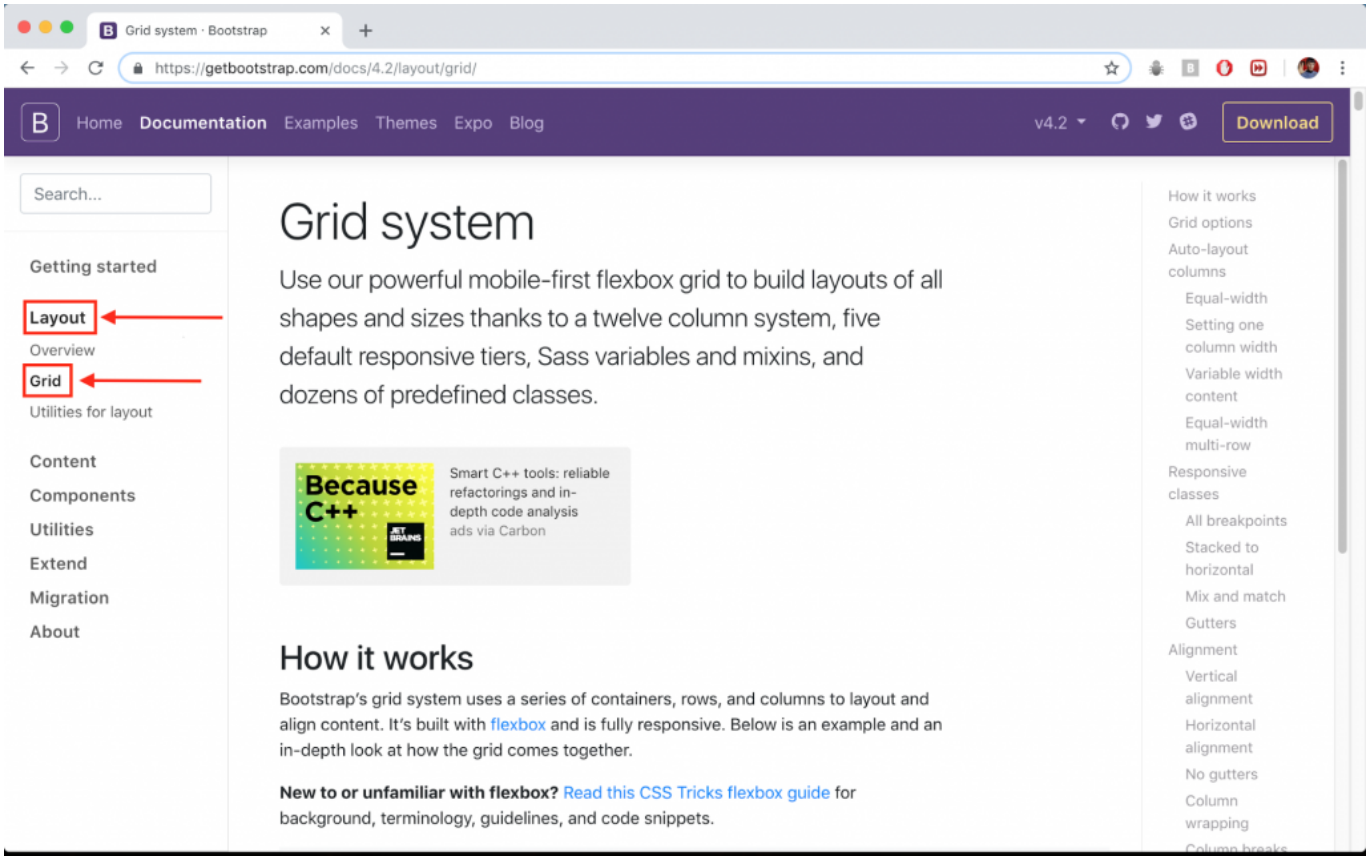
The template contains just a header with the text "Hello, world!" in it and we can run it by opening with a browser of our choice.



After running the page, the result will look like this:

So now we are all set up and ready to incorporate Bootstrap into our new HTML file! Next up, we will be exploring layout techniques with the Bootstrap Grid.

The first step to building any webpage should be deciding how to lay the page out. This can be more challenging than it might first seem, especially when taking into account different screen sizes. Bootstrap has us covered here by providing an easy to use layout system: The Grid. We will explore The Grid on the layout page.



The Grid is basically a series of columns inside of rows inside of a div of class "container" and functions similarly to a CSS flex-box combined with an HTML table. The base of any grid is a container:

```
<div class="container">
  <!-- Content here -->
</div>
```

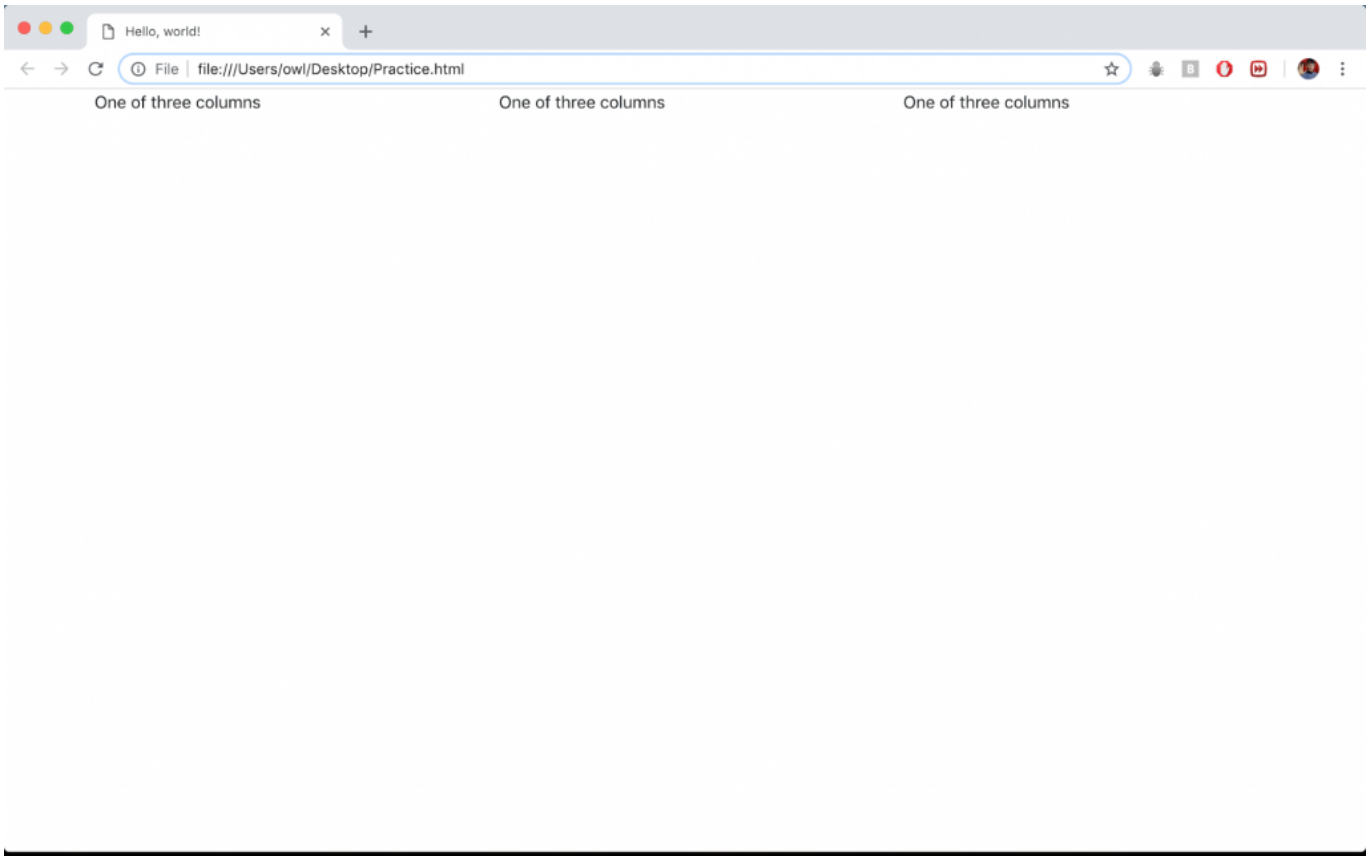
A container will take on the height of the content and the width of the content, unless the container is specified to be a “fluid” container.

```
<div class="container-fluid">
  ...
</div>
```

If we want a single row with, say, 3 columns, we can use the code:

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

To achieve the result:



Columns must be the direct descendants of rows and automatically apply padding to the left and right. They space themselves out evenly unless a specific width is applied. You can get rid of the left and right margins by applying the code to a row:

```
<div class="container">
  <div class="row no-gutters">
    <div class="col align-self-start">
      One of three columns
    </div>
    <div class="col align-self-center">
      One of three columns
    </div>
    <div class="col align-self-end">
      One of three columns
    </div>
  </div>
</div>
```

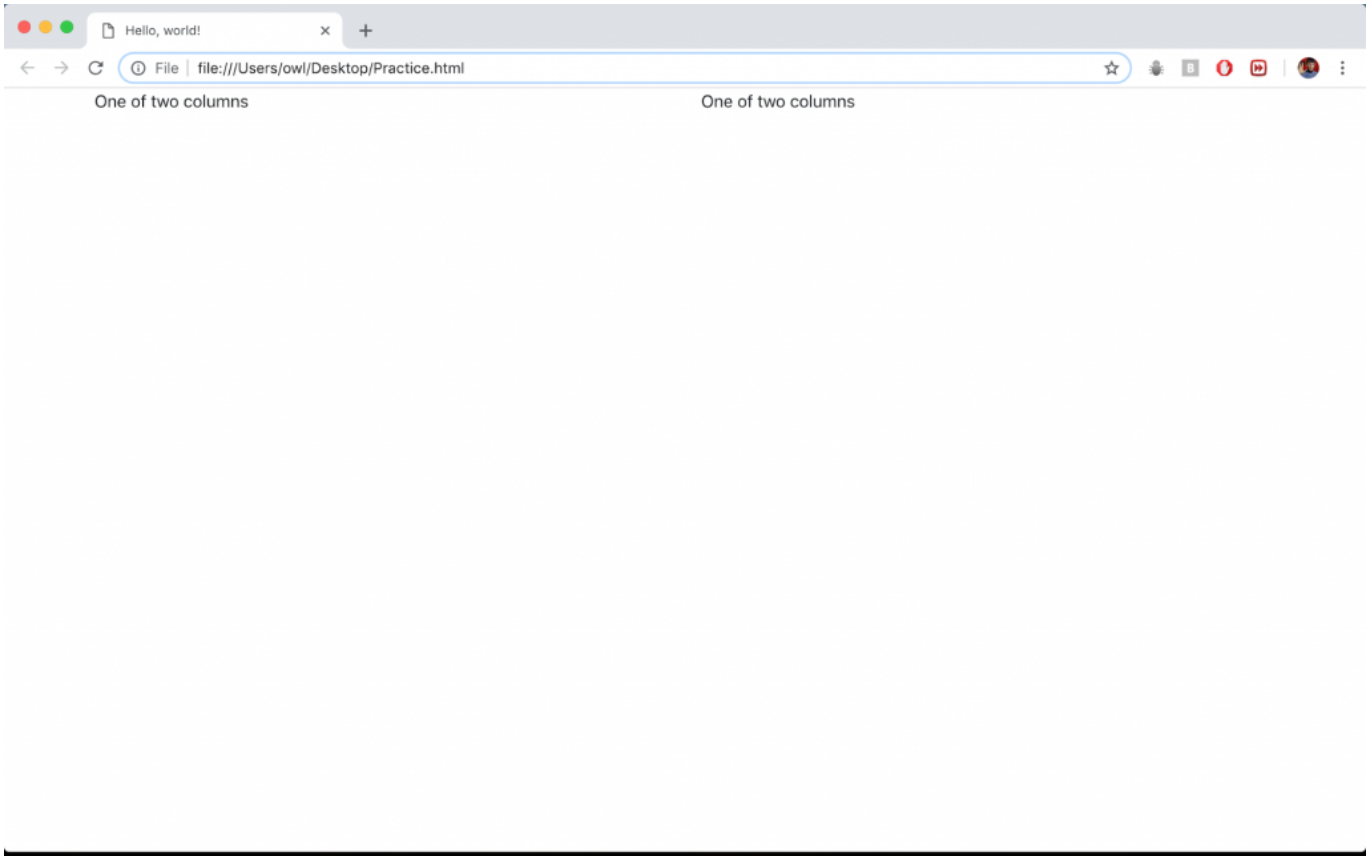
Each row is divided into 12 units and we can use these units to specify proportions of the width to take up. For example, If we wanted a column to take up half the width of a row, we apply the code:

```
<div class="container">
  <div class="row">
    <div class="col-6">
      One of three columns
    </div>
    <div class="col-6">
```



```
      One of three columns
    </div>
  </div>
</div>
```

And get the result:

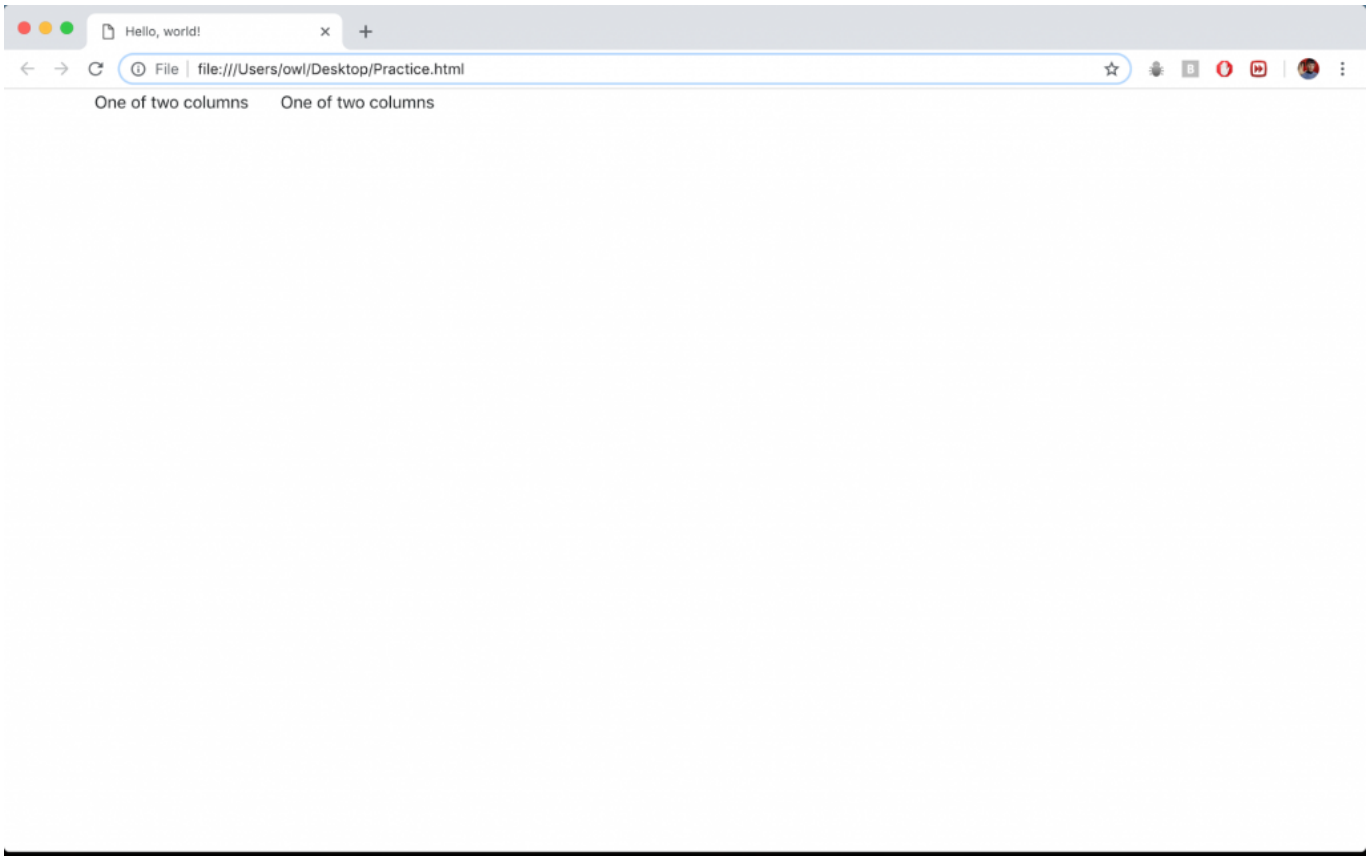


Note, a column doesn't have to contain content to take up a certain width.

We might want a column or columns to just wrap around the width of the content rather than having a predetermined width. We can have a column wrap around the content by applying "auto" to the column width:

```
<div class="container">
  <div class="row">
    <div class="col-auto">
      One of three columns
    </div>
    <div class="col-auto">
      One of three columns
    </div>
  </div>
</div>
```

And the result would be:



By now, we should understand the basics of how grids are built using rows and columns and how to containers out. Before moving to the next section, I recommend you take a break here and try some different techniques out yourself. There is a lot to cover on that page still.

Now that we know how to add bootstrap components to our HTML page, let's put together a small system that allows us to press a button and display some text or select some text to display from a dropdown menu. We will build a 3 column grid that will have a button in the first column, a dropdown menu in the second column, and the text to display in the third column. We'll also add in an image carousel, as it is a popular way to display a cycling series of images. Let's start with the grid:

```
<div class="container">
  <div class="row">
    <div class="col">

    </div>
    <div class="col">

    </div>
    <div class="col">

    </div>
  </div>
</div>
```

Now let's add a button. Let's choose a different style of button and add some more interesting behaviour to it:

```
<div class="container">
  <div class="row">
    <div class="col">
 <button type="button" class="btn btn-outline-success btn-block" onClick="" data-
toggle="button" aria-pressed="false" autocomplete="off">Press me</button>
    </div>
    <div class="col">

    </div>
    <div class="col">

    </div>
  </div>
</div>
```

Now we will add the functionality to the button that will change the text in the 3rd column:

```
<div class="container">
  <div class="row">
    <div class="col">
 <button type="button" class="btn btn-outline-success btn-block" onClick="" data-
toggle="button" aria-pressed="false" autocomplete="off">Press me</button>
    </div>
    <div class="col">

    </div>
    <div class="col" id='display'>
```



```
    </div>
  </div>
</div>

<script type="text/javascript">
  var display = document.getElementById('display');

  function changeText() {
    display.innerHTML = 'Button was pressed';
  }

  function selectText(text) {
    display.innerHTML = text;
  }
</script>
```

The button looks good and works properly as well so we can add the dropdown with some text options to choose from. We will add the functionality as well so that when a user clicks on an option from the dropdown menu, the text in the 3rd column changes:

```
<div class="container">
  <div class="row">
    <div class="col">
      <button type="button" class="btn btn-outline-success btn-block" onClick='change
Text()'  data-toggle="button" aria-
pressed="false" autocomplete="off">Press me</button>
    </div>
    <div class="col">
      <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownM
enuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Dropdown button
        </button>
        <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
          <a class="dropdown-
item" href="#" onClick='selectText("Choice 1")'>Choice 1</a>
          <a class="dropdown-
item" href="#" onClick='selectText("Choice 2")'>Choice 2</a>
          <a class="dropdown-
item" href="#" onClick='selectText("Choice 3")'>Choice 3</a>
        </div>
      </div>
    </div>
    <div class="col" id='display'>

    </div>
  </div>
</div>

<script type="text/javascript">
  var display = document.getElementById('display');

  function changeText() {
```



```
    display.innerHTML = 'Button was pressed';
  }

  function selectText(text) {
    display.innerHTML = text;
  }
</script>
```

Now we will start a new row and add an image carousel. This is basically a small window that displays one of a few images and is very commonly seen on home pages as a way to display some pictures. The next image can be cycled to automatically or when the user presses on left and right buttons. Our carousel will have left and right buttons to go to the next or previous image and will display the Zenva logo in 3 different sizes:

```
<div class='row'>
  <div class='col'>
    <div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          <img src="zenvaLogo.png" class="d-block w-25" alt="...">
        </div>
        <div class="carousel-item">
          <img src="zenvaLogo.png" class="d-block w-50" alt="...">
        </div>
        <div class="carousel-item">
          <img src="zenvaLogo.png" class="d-block w-100" alt="...">
        </div>
      </div>
      <a class="carousel-control-
prev" href="#carouselExampleControls" role="button" data-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="sr-only">Previous</span>
      </a>
      <a class="carousel-control-
next" href="#carouselExampleControls" role="button" data-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="sr-only">Next</span>
      </a>
    </div>
  </div>
</div>
```

And that’s it! We have learned how to incorporate Bootstrap into our webpages. We learned how to navigate the website, how to build our page layouts with the Grid, and how to add and customize elements. We explored alerts, buttons, dropdown menus, and image carousels and now know enough to build our own webpages from scratch using Bootstrap!

