Bootstrap 5

Using Basic Styles in Bootstrap 5

Bootstrap is a free and open-source front-end framework (library) for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.[1]

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub!

Why Use Bootstrap?

Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)[2]

Installing Bootstrap 5

There are different ways to install Bootstrap. Bootstrap can be installed as a local CSS file linked to the website project or an external URL link.

Using a Bootstrap as a local file, you download the pre compiled Bootstrap CSS and JavaScript files and same them in your local project folder. It works really well if you need to install a copy of Bootstrap that will work even without an internet connection.

Another way is to use CDNs. A **CDN, Content Delivery Network,** is a link which needs a place that hosts common libraries like Bootstrap. You can also download the source files. This way, you download not just the CSS and JavaScript, but all the files that the developers use to create Bootstrap.

---

[1] Bootstrap, Wikipedia.com, https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework), retrieved April 2022
[2] Bootstrap, w3school.com, https://www.w3schools.com/bootstrap/bootstrap_get_started.asp, retrieved April 2022

Now to get instructions and to download a copy of Bootstrap manually, first go to getbootstrap.com website and click on this **Download** button.
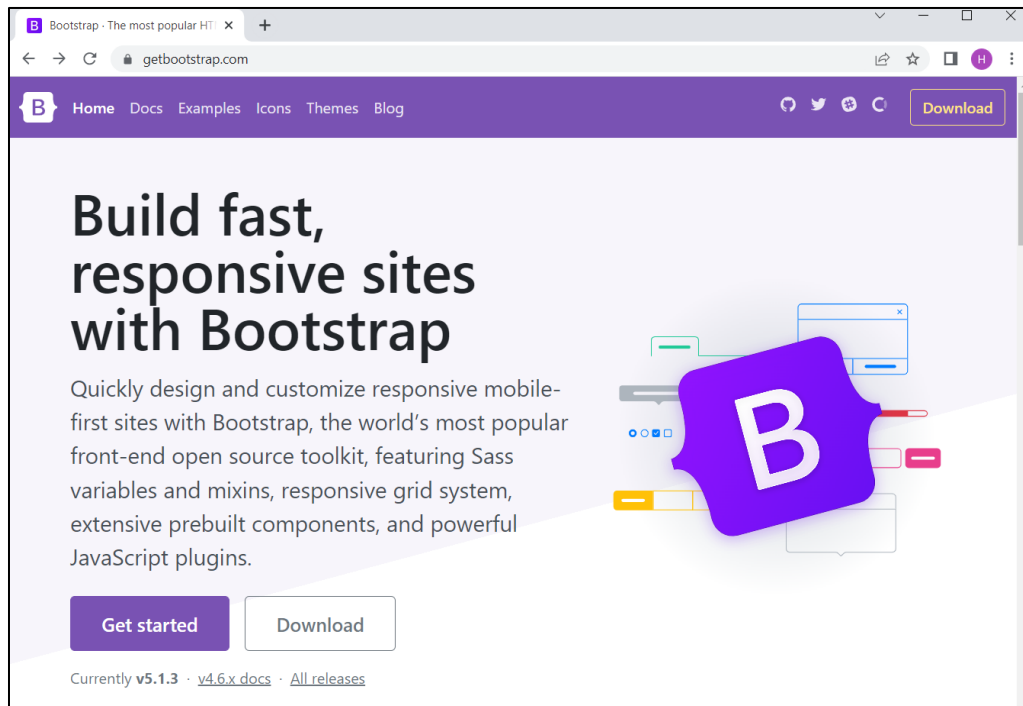

On the Download page, it shows the different ways that you can download Bootstrap, let us explore the compiled CSS and JS, CDN, and Started Template download options.

Compiled CSS and JS

This option allows user to Download ready-to-use compiled code for **Bootstrap v5.1.3** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see CSS files comparison)
- Compiled and minified JavaScript plugins (see JS files comparison)

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper. If you open up the folder, you see a CSS folder and a JavaScript folder:
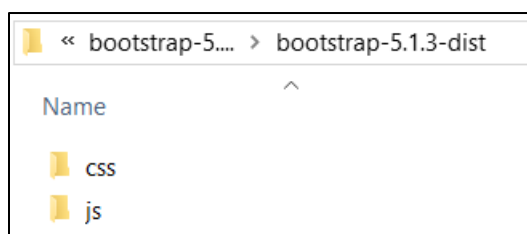
CSS Folder

The CSS folder has a number of different versions of the CSS use for the Bootstrap framework. Now if you look at this folder, you'll notice that there are three types of files: regular css files, minimized version, and map files.

- **Map files** are files that are useful if you're using Bootstrap in development mode so that they could point you not to the CSS, but to the original SAS code that was used to create the CSS. Some people like to use only the grid features of Bootstrap for a layout so you can get just that if you want to.
- There's also a Bootstrap **reboot file**. The reboot files are the special Bootstrap code that resets the CSS and browsers so that they work more consistently in different platforms.
- The **regular version** of Bootstrap contains the grid, the reset, and everything else in Bootstrap. You can choose the regular version if you're going to customize Bootstrap, or just choose **bootstrap.min.css** if you just want to use that file.

Name	Type
bootstrap.rtl	Cascading Style Sheet Document
bootstrap.rtl.css.map	MAP File
bootstrap.rtl.min	Cascading Style Sheet Document
bootstrap.rtl.min.css.map	MAP File
bootstrap-grid	Cascading Style Sheet Document
bootstrap-grid.css.map	MAP File
bootstrap-grid.min	Cascading Style Sheet Document
bootstrap-grid.min.css.map	MAP File
bootstrap-grid.rtl	Cascading Style Sheet Document
bootstrap-grid.rtl.css.map	MAP File
bootstrap-grid.rtl.min	Cascading Style Sheet Document
bootstrap-grid.rtl.min.css.map	MAP File
bootstrap-reboot	Cascading Style Sheet Document
bootstrap-reboot.css.map	MAP File
bootstrap-reboot.min	Cascading Style Sheet Document
bootstrap-reboot.min.css.map	MAP File
bootstrap-reboot.rtl	Cascading Style Sheet Document
bootstrap-reboot.rtl.css.map	MAP File
bootstrap-reboot.rtl.min	Cascading Style Sheet Document
bootstrap-reboot.rtl.min.css.map	MAP File
bootstrap-utilities	Cascading Style Sheet Document
bootstrap-utilities.css.map	MAP File
bootstrap-utilities.min	Cascading Style Sheet Document
bootstrap-utilities.min.css.map	MAP File
bootstrap-utilities.rtl	Cascading Style Sheet Document
bootstrap-utilities.rtl.css.map	MAP File
bootstrap-utilities.rtl.min	Cascading Style Sheet Document
bootstrap-utilities.rtl.min.css.map	MAP File

JavaScript Folder

In the JavaScript folder, you'll also find several files as well.

Name	Type
bootstrap.bundle	JavaScript File
bootstrap.bundle.js.map	MAP File
bootstrap.bundle.min	JavaScript File
bootstrap.bundle.min.js.map	MAP File
bootstrap.esm	JavaScript File
bootstrap.esm.js.map	MAP File
bootstrap.esm.min	JavaScript File
bootstrap.esm.min.js.map	MAP File
bootstrap	JavaScript File
bootstrap.js.map	MAP File
bootstrap.min	JavaScript File
bootstrap.min.js.map	MAP File

JavaScript also has maps. Since most people write in a version of JavaScript, it gets translated so that it works with older browsers.

There are regular and minimized versions of the JavaScript as well. Other than that, there are two types of files: the regular Bootstrap and the Bootstrap bundle. The Bootstrap bundle has all the Bootstrap code plus an additional library called 'popper.js'. **Popper.js** is a positioning engine, its purpose is to calculate the position of an element to make it possible to position it near a given reference element.

Most of time you're going to need just the regular Bootstrap file, however you also use the minimized version since you don't want to edit any of the Bootstrap JavaScript yourself.

CDN

When someone visits a site that uses a CDN link, their browser will check its cache or memory to see if the visitor has been to a similar site that's also using the same link. If that's the case, then the browser will load the cached version of the library. Since it's already stored in memory, that makes the new site load faster since the browser will not have to request the file. You can download the CDN version of bootstrap at https://www.bootstrapcdn.com/

Starter Template: All together

In order to get start a project using bootstrap, you will need the following file links to your project document:

- Bootstrap
- jQuery
- popper.js
- basic HTML document

To get all the templates at once, you can go to: https://getbootstrap.com/docs/5.1/getting-started/introduction/#starter-template and click on the **Copy** link to copy the Starter template at the clipboard

Starter template

Be sure to have your pages set up with the latest design and development standards. That means using an HTML5 doctype and including a viewport meta tag for proper responsive behaviors. Put it all together and your pages should look like this:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="styl

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" int

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integr
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="
    -->
  </body>
</html>
```

When copy the starter template, you will also need to add important global styles and settings to your project files. The global set-ups are:

- **HTML5 doctype**

  Bootstrap requires the use of the HTML5 doctype. Without it, you'll see some funky incomplete styling, but including it shouldn't cause any considerable hiccups.

  ```html
  <!doctype html>
  <html lang="en">
    ...
  </html>
  ```

- **Responsive meta tag**

  Bootstrap is developed *mobile first*, a strategy in which we optimize code for mobile devices first and then scale up components as necessary using CSS media queries. To ensure proper rendering and touch zooming for all devices, **add the responsive viewport meta tag** to your `<head>`.

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

  You can see an example of this in action in the starter template.

- **Box-sizing**

  For more straightforward sizing in CSS, we switch the global box-sizing value from content-box to border-box. This ensures padding does not affect the final computed width of an element, but it can cause problems with some third-party software like Google Maps and Google Custom Search Engine.

  On the rare occasion you need to override it, use something like the following:

  ```css
  .selector-for-some-widget {
    box-sizing: content-box;
  }
  ```

  With the above snippet, nested elements—including generated content via ::before and ::after—will all inherit the specified box-sizing for that .selector-for-some-widget.

  Learn more about box model and sizing at CSS Tricks.

1. Text/Typography

The CSS code that comes with Bootstrap overrides default browser behavior. Besides making things look great, it's also designed to be easy to override with your own CSS.

Bootstrap 5 uses a default **font-size** of 1em as 16px, and its **line-height** is **1.5**.

The default **font-family** is "Helvetica Neue", Helvetica, Arial, sans-serif.

In addition, all <p> elements have **margin-top: 0** and **margin-bottom: 1em** (16px by default).

Heading classes

Bootstrap 5 styles HTML headings (`<h1>` to `<h6>`) with a bolder font-weight and a responsive font-size.

You can also use h1 to h6 classes on other elements to make them behave as headings if you want:

```
<p class="h3">h3 Bootstrap heading</p>
```

`<small>`

In Bootstrap 5 the HTML `<small>` element (and the `small` class) is used to create a smaller, secondary text in any heading:

```
<h2>This is an example of <small>h2 using small tag</small></h2>
<h2>This is an example of h2 without using small tag</h2>
```

Display Headings

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are six classes to choose from: `display-1` to `display-6`:

```
<h1 class="display-1">H1 using display-1 class</h1>
<h1>H1 without using display-1 class</h1>
```

H1 using display-1 class
H1 without using display-1 class

`<mark>`

Bootstrap 5 will style <mark> and .mark with a yellow background color and some padding:

```
<p>Use the mark element to <mark>highlight</mark> text.</p>
```

Use the mark element to highlight(or the .mark class) to text in within the mark text.

You can also check the highlight background color using CSS property background-color within the <mark> open tag:

```
<p>Use the mark element to highlight(or the .mark class) to <mark
style="background-color: lightgreen;">text in within the mark </mark>
text.</p>
```

Use the mark element to highlight(or the .mark class) to text in within the mark text.

`<kbd>`

`<kbd>` element indicates input that is typically entered via the keyboard

```
<p>Use <kbd>ctrl + p</kbd> to open the Print dialog box.</p>
```

Use `ctrl + p` to open the Print dialog box.

You can also change the background color using the background-color CSS property within the open `<kbd>` tag.

```
<p>Use <kbd style="background-color: purple;">ctrl + p</kbd> to open the
Print dialog box.</p>
```

Use `ctrl + p` to open the Print dialog box.

`<blockquote>`

Add the `blockquote` class to a `<blockquote>` when quoting blocks of content from another source. And when naming a source, like "from WWF's website", use the `blockquote-footer` class:

```
<blockquote class="blockquote">
     <p>Birds are a group of warm-blooded vertebrates constituting the class
     Aves.</p>
     <footer class="blockquote-footer">From Wikipedia.org website</footer>
</blockquote>
```

Birds are a group of warm-blooded vertebrates constituting the class Aves.
— From Wikipedia.org website

More Typography Classes

The Bootstrap 5 classes below can be added to style HTML elements further:

Class	Description	Example
.lead	Makes a paragraph stand out	Try it
.text-start	Indicates left-aligned text	Try it
.text-break	Prevents long text from breaking layout	Try it
.text-center	Indicates center-aligned text	Try it
.text-decoration-none	Removes the underline from a link	Try it
.text-end	Indicates right-aligned text	Try it
.text-nowrap	Indicates no wrap text	Try it
.text-lowercase	Indicates lowercased text	Try it
.text-uppercase	Indicates uppercased text	Try it

.text-capitalize	Indicates capitalized text	Try it
.initialism	Displays the text inside an <abbr> element in a slightly smaller font size	Try it
.list-unstyled	Removes the default list-style and left margin on list items (works on both and). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)	Try it
.list-inline	Places all list items on a single line (used together with .list-inline-item on each elements)	

Colors and Image Manipulation using Bootstrap 5

Colors

Bootstrap has a number of colors that you can access through contextual color names. They're used everywhere, including in buttons, background colors, as well as texts, so you'll hear some of these names in different places, but let's take a look at the basic options available in texts, links, and backgrounds. So the way that these work is by using the prefix text, and then using one of the contextual color names. Color is Bootstrap use the following name:

primary secondary success danger
warning info light dark white

Text colors
The classes for text colors are**:.text-muted, .text-primary, .text-success, .text-info, .text-warning, and .text-danger**

```
<h1 class="text-danger">H1 using <b><i>text-danger</i></b> class</h1>

<h2 class="text-success">H2 using <b><i>text-success</i></b> class</h2>

<h3 class="text-muted">H3 using <b><i>text-muted</i></b> class</h3>

<h4 class="text-info">H4 using <b><i>text-success</i></b> class.</h4>

<h5 class="text-primary">H5 using <b><i>text-primary</i></b> class</h5>

<h6 class="text-warning">H6 using <b><i>text-warning</i></b> class</h6>
```

```
<p class="text-secondary">Paragraph using <b><i>text-secondary</i></b>
class</p>
<p class="text-dark">Paragraph using <b><i>text-dark</i></b> class</p>
<p class="text-body">Paragraph using <b><i>text-body</i></b> class</p>
<p class="text-light">Paragraph using <b><i>text-light</i></b> class</p>
<p class="text-white">Paragraph using <b><i>text-white</i></b> class</p>
```


Background color classes
The classes for background colors are: **bg-primary, bg-success, bg-info, bg-warning, bg-danger**
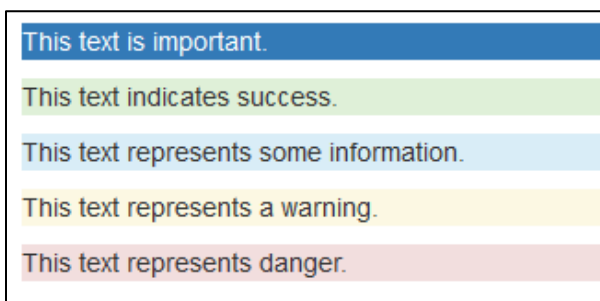
Working with images

Bootstrap also has some very useful classes that help you work with images, as well as with figures.

Border images

The three basic Bootstrap images' classes are rounded, rounded-circle, and img-thumbnail

- The **rounded** class adds rounded corners to an image. You can also apply direction of which edge the image do you want it to be rounded.

  ```
  <img src="paris.jpg" class="rounded" alt="Elfer Tower" width="20%">
  ```


- The **rounded-circle** class shapes the image to a circle. For this, the image should have the same height and width. If the image has different height and width, then the image become an oval shape.

  ```
  <img src="NYC.jpg" class="rounded-circle" alt="image of NYC"
  width="200px" height="200px">
  ```



  ```
  <img src="cinqueterre.jpg" class="rounded-circle" alt="Cinque Terre"
  width="300px" height="200px">
  ```

- The **img-thumbnail** class shapes the image to a thumbnail (bordered).

```
<img src="SF.jpg" class="img-thumbnail" alt="Golden Gate – San
Francisco" width="30%">
```


Aligning images

Float an image to the right with the **float-end** class or to the left with **float-start**

Center an image by adding the utility classes **mx-auto** (margin:auto) and **d-block** (display:block) to the image.

```
<img src="paris.jpg" class="float-start img-thumbnail " alt="Paris" width="20%" >
<img src="paris.jpg" class="float-end rounded-circle" alt="Paris" width="20%">
<img src="paris.jpg" class="mx-auto d-block rounded" alt="Paris" width="20%">
```
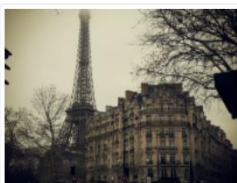

Responsive images

Responsive images automatically adjust to fit the size of the screen.

- **img-fluid:** Create responsive images by adding an **img-fluid** class to the **<img>** tag. The image will then scale nicely to the parent element.

    The **img-fluid** class applies **max-width: 100%;** and **height: auto;**

    ```
    <img class="img-fluid" src="ny.jpg" alt="New York">
    ```

The .img-fluid class makes the image scale nicely to the parent element (resize the browser window to see the effect):

- **figure-img** class makes the text with a smaller font and lighter color. The **<figure>** tag in HTML specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

  While the content of the **<figure>** element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document.

- **figure-caption** class makes the text in **<figcaption>**with a smaller font and lighter color. The HTML **<figcaption>** element represents a caption or legend for the rest of the contents its parent figure element, if any.

```
<figure class="figure-img"> <img src="ny.jpg" alt="New York" width="30%;">
      <p class="figure-caption">Central Park Summer Festival</p>
</figure>
```


Central Park Summer Festival

Basic List Groups

The most basic list group is an unordered list with list items. To create a basic list group, use an **`<ul>`** element with class **`list-group`**, and **`<li>`** elements with class **`list-group-item`**:

Example) Create a simple list
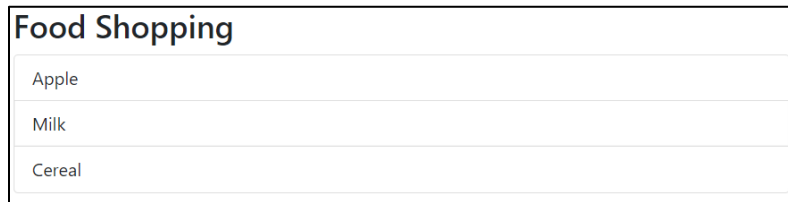

```
<h2>Food Shopping</h2>
  <ul class="list-group">
    <li class="list-group-item">Apple</li>
    <li class="list-group-item">Milk</li>
    <li class="list-group-item">Cereal</li>
  </ul>
```

Active State on a list

Use the **`active`** class to highlight the current item:
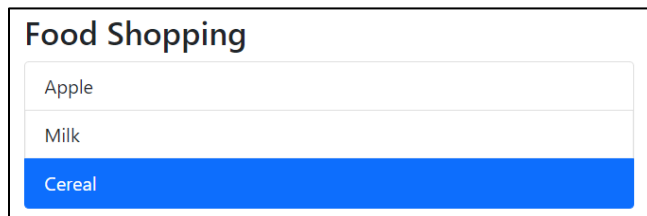

```
<h2>Food Shopping</h2>
  <ul class="list-group">
    <li class="list-group-item">Apple</li>
    <li class="list-group-item">Milk</li>
    <li class="list-group-item active">Cereal</li>
  </ul>
```

Basic Dropdown

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list. Although dropdowns are technically a separate component, they are used very often in menus. You can use dropdown in a bunch of different components including navs, tabs, and buttons.

A dropdown requires a container, just like a lot of other elements to work. Therefore, you will need to create something that has a class of **`dropdown.`** This is usually a **`div`**, or it could be on a

list item if you're using a navigation that uses list items. To do so, we need to apply the **dropdown, dropdown-toggle, data-bs-toggle="dropdown", dropdown-item,** and **dropdown-menu**.

- The **dropdown** class indicates a dropdown menu.
- To open the dropdown menu, use a button or a link with a class of **dropdown-toggle** and the **data-bs-toggle="dropdown"** attribute.
- Add the **dropdown-menu** class to a **<div>** or **<ul>** element to actually build the dropdown menu.
- Then add the **dropdown-item** class to each element (links or buttons) inside the dropdown menu.

Example) Create a dropdown menu with three hyperlinks
First, we create a container for the dropdown menu and use Bootstrap `class="dropdown"`

```
<div class="dropdown">

</div>
```

Within the dropdown container, we can add a button to open and close the dropdown menu. The button should have **class="dropdown-toggle"** and **data-bs-toggle="dropdown"** to open and close the dropdown menu:

```
<div class="dropdown">
    <button type="button" class=" dropdown-toggle btn btn-secondary"
    data-bs-toggle="dropdown">CUNY Community College</button>
 </div>
```

After the button, it is time to create the dropdown menu. The dropdown menu can be created using an unordered list with `class="dropdown-menu"`

```
<div class="dropdown">
    <button type="button" class=" dropdown-toggle btn btn-secondary"
    data-bs-toggle="dropdown">CUNY Community College</button>
    <ul class="dropdown-menu">

    </ul>
 </div>
```

Within the dropdown-menu goes the link or item in the list. For each **`<a>`** element in the list, we add a Bootstrap **`class="dropdown-item"`**

```
<div class="dropdown">
  <button type="button" class=" dropdown-toggle btn btn-secondary"
   data-bs-toggle="dropdown">CUNY Community College</button>
  <ul class="dropdown-menu">
      <li><a href="https://www.qcc.cuny.edu/" target="_blank" class="dropdown-item">
      Queensborough CC</a> </li>
      <li><a href="http://www.bcc.cuny.edu/" class="dropdown-item" target="_blank"> Bronx
      CC</a> </li>
      <li><a href="https://www.kbcc.cuny.edu/" class="dropdown-item" target="_blank">
      Kingsborough BCC</a> </li>
  </ul>
</div>
```

Adding dropdown menu orientation, dropdown menu header, and dropdown menu separator

Dropdown menu can change the dropdown orientation to top, left, or right by the following class in the dropdown container:
- `class="dropdown dropend"` dropdown menu will open to the right
- `class="dropdown dropstart"` dropdown menu will open to the left
- `class="dropdown dropup"` dropdown menu will open on top

To add a dropdown menu separator, we need to add a **`<li><hr></li>`** where the separator goes. For a dropdown menu header, we can add a **`<li>`** with **`class="dropdown-header"`**

Example) Create a dropdown list that opens to the right with menu separator
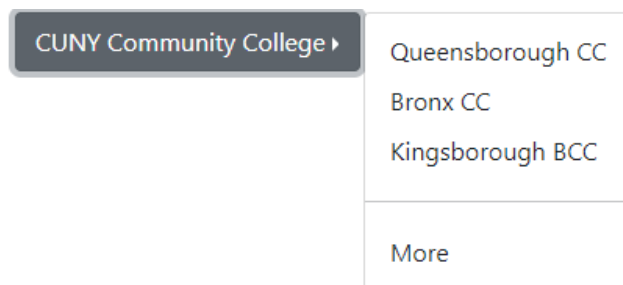
```
<div class="dropdown dropend">
  <button type="button" class=" dropdown-toggle btn btn-secondary"
   data-bs-toggle="dropdown">CUNY Community College</button>
  <ul class="dropdown-menu">
      <li><a href="https://www.qcc.cuny.edu/" target="_blank" class="dropdown-item">
      Queensborough CC</a> </li>
      <li><a href="http://www.bcc.cuny.edu/" class="dropdown-item" target="_blank"> Bronx
      CC</a> </li>
      <li><hr></li>
      <li><a href="https://www.cuny.edu/" target="_blank"
      class="dropdown-item"> More</a> </li>
  </ul>
</div>
```

Example) Create a dropdown menu that opens up with dropdown headers
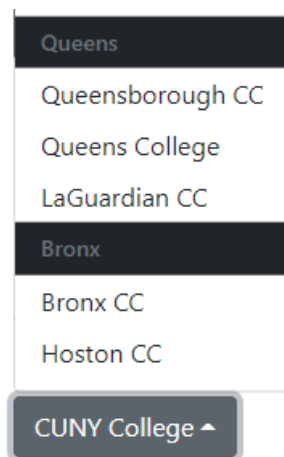

```
<div class="dropdown dropup">
  <button type="button" class=" dropdown-toggle btn btn-secondary"
   data-bs-toggle="dropdown">CUNY Community College</button>
  <ul class="dropdown-menu">
      <li class="bg-dark text-light"><h5 class="dropdown-header"> Queens</h5></li>
      <li><a href="https://www.qcc.cuny.edu/" target="_blank" class="dropdown-item">
      Queensborough CC</a> </li>
      <li><a href="http://www.qc.cuny.edu/" class="dropdown-item"
      target="_blank">Queens College</a> </li>
      <li class="bg-dark text-light"><h5 class="dropdown-header"> Bronx</h5></li>
      <li><a href="http://www.bcc.cuny.edu/" class="dropdown-item" target="_blank"> Bronx
      CC</a> </li>
  </ul>
</div>
```

Grouping buttons and dropdown menu

There are a few classes that help you create button groups. To group simple buttons together you can use the **btn-group** class. There's also a vertical option for button groups that stack on top of one another. Those are traditionally used for mobile devices but you can use them for whatever you want. Finally, there is a toolbar option that let us create groups of button groups. Now when you're creating button groups it's a good idea to use the aria label property to add a little bit of context to your button groups:

- btn-group
- btn-group-vertical
- btn-toolbar

Example) Create a vertical menu with buttons
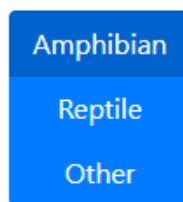
For this, we can create a container using **<div>** and add **class='btn-group-vertical'**

```
<div class='btn-group-vertical'>

</div>
```

Within the container, we can add individual buttons

```
<div class='btn-group-vertical'>
    <button type="button" class="btn btn-primary active">Amphibian</button>
    <button type="button" class="btn btn-primary">Reptile</button>
    <button type="button" class="btn btn-primary">Other</button>
  </div>
```

Example) Create horizontal navigation bars using buttons with one button with dropdown menu
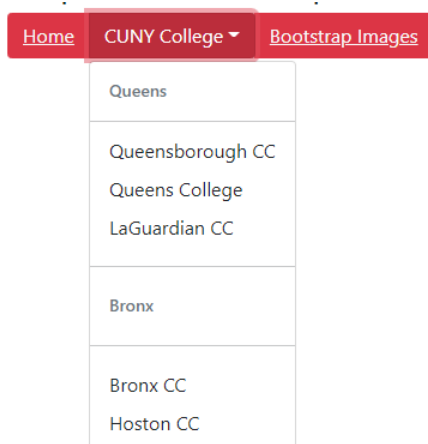
For this, we can create a container using **`<div>`** and add **`class='btn-group'`**

```
<div class='btn-group'>

</div>
```

Within the container, we can add individual buttons

```
<div class='btn-group >
   <button type="button" class="btn btn-danger"> Home </button>
   <button type="button" class="btn btn-danger > CUNY College </button>
   <button type="button" class="btn btn-danger"> Bootstrap Images </button>
</div>
```

Since we are using the second button as downdrop menu, then we can add the classes `dropdown-toggle" data-bs-toggle="dropdown"` to the second button:

```
<div class='btn-group >
   <button type="button" class="btn btn-danger"> Home </button>
   <button type="button" class="btn btn-danger data-bs-toggle="dropdown"
   dropdown-toggle"> CUNY College </button>
   <button type="button" class="btn btn-danger"> Bootstrap Images </button>
</div>
```

Now we can add the dropdown menu and link the menu to the second button:

```
<div class="btn-group">
   <button type="button" class="btn btn-danger"> Home</button>
   <button type="button" class="btn btn-danger data-bs-toggle="dropdown"
   dropdown-toggle" > CUNY College</button>
   <button type="button" class="btn btn-danger"> Bootstrap Images</button>
   <!-- dropdown menu -->
   <ul class="dropdown-menu">
     <li><h5 class="dropdown-header">Queens</h5><hr class="dropdown-divider"></li>
     <li><a href="https://www.qcc.cuny.edu/" target="_blank" class="dropdown-item">
       Queensborough CC</a> </li>
     <li><a href="http://www.qc.cuny.edu/" class="dropdown-item" target="_blank">
       Queens College</a> </li>
     <li><a href="https://www.kbcc.cuny.edu/" class="dropdown-item"
       target="_blank"> LaGuardian CC</a> </li>
     <li><hr><h5 class="dropdown-header">Bronx</h5><hr></li>
     <li><a href="https://www.bcc.cuny.edu/" target="_blank" class="dropdown-item">
         Bronx CC</a> </li>
     <li><a href="https://www.hoston.cuny.edu/" target="_blank"
       class="dropdown-item">Hoston CC</a></li>
   </ul>
</div>
```

After it, we can create a hyperlink to the first and third button:

```
<div class='btn-group >
  <button type="button" class="btn btn-danger"><a href="#" class="text-light">
  Home</a></button>
  <button type="button" class="btn btn-danger data-bs-toggle="dropdown"
  dropdown-toggle"> CUNY College </button>
  <button type="button" class="btn btn-danger"><a href="#BI" class="text-light">
  Bootstrap Images</a></button>
  <!-- dropdown menu -->
  <ul class="dropdown-menu">
    <li><h5 class="dropdown-header">Queens</h5><hr class="dropdown-divider"></li>
    <li><a href="https://www.qcc.cuny.edu/"class="dropdown-item" target="_blank">
      Queensborough CC</a> </li>
    <li><a href="http://www.qc.cuny.edu/" class="dropdown-item" target="_blank">
      Queens College</a> </li>
    <li><a href="https://www.kbcc.cuny.edu/" class="dropdown-item"
      target="_blank"> LaGuardian CC</a> </li>
    <li><hr><h5 class="dropdown-header">Bronx</h5><hr></li>
    <li><a href="https://www.bcc.cuny.edu/" target="_blank"class="dropdown-item">
        Bronx CC</a> </li>
    <li><a href="https://www.hoston.cuny.edu/" target="_blank"
      class="dropdown-item">Hoston CC</a></li>
  </ul>
</div>
```

Collapsible and Accordions

Collapse and accordions are a set of related components that allow you to show or hide HTML content. Collapsible elements are used to hire and show information when an active button is clicked. Accordions are similar but part of a bigger group. When one item shows, another hides, so accordions are essentially group of collapses. To set up a collapse, you create a link or a button that activates the collapse element. To that element, you add a **data-toggle="collapse"** and to target the element, you can use either an **id** if it's an anchor tag or link, or you can use a data-target attribute if it's a button.

Example) create a collapsible element that shows information and a picture of a cat when is clicked

```
<button type="button" class="btn btn-info" data-bs-toggle="collapse" data-bs-
target="#cat">Info about cats</button>
<div id="cat" class="collapse">
   <p style="margin:1em 0em;">The cat is a domestic species of small
   carnivorous mammal. It is the only domesticated species in the family
   Felidae and is often referred to as the domestic cat to distinguish it
   from the wild members of the family.</p>
   <figure>
      <img src="cat.jpg" alt="images of a cat" width="50%;" class="mx-auto
      d-block img-thumbnail">
      <i class="small figure-caption">Photo by <a
      href="https://www.pexels.com/photo/brown-cat-with-green-eyes-617278/"
      target="_blank">Kelvin Valerio</a></i>
   </figure>
</div>
```

Mastering Layout with bootstrap

By far the most useful part of Bootstrap is the grid, so this is the first thing you should master. The Bootstrap grid gives you a framework to create any layout that you can think of. Now, Bootstrap layouts have three main pieces. The first is containers, which can be either responsive fixed width that snap to certain breakpoints, or completely fluid containers which take up 100% of the width of the view port, which is the browser window or the width of your device. Inside containers, you can also place content in rows and columns.

Bootstrap uses a 12-column grid system that has breakpoints for extra small, small, medium, large, and extra large devices. If you're coming from a previous version of Bootstrap, note that there is

an additional breakpoint for smaller devices. The grid is extremely flexible. You can create multiple layouts for different breakpoints, reset, offset, nest, and even customize the order of the columns at different breakpoints. Bootstrap's grid is very powerful.

If you can think of a layout, you can build it easily and responsively with the Bootstrap grid. The more time you spend learning how to use it, the easier it will be to get your work done.

Containers and rows

One of the main reasons people use Bootstrap is to have access to an amazing grid that makes layout easy to do and responsive. This is one of the most important things you should master in Bootstrap, so let's take a look. First of all, the grid is a responsive 12-column system for creating just about any layout you can think of. It uses a technology called Flexbox that makes it easier to create complex layout with minimal code. In order to work with the grid, you need to master three simple components.

The first is containers, which can be used with or without the grid to **align-content** either to the viewport, or center it around a set of breakpoints. Next is rows and columns. They work together to allow you to create the layouts. The rows prepare the columns for layout, and the columns are complex and extremely flexible, so we'll cover them in a separate video in detail. So first, let's go ahead and take a look at containers. There are two different types of containers.

container class is used for regular containers, which center content and snap to certain grid points.

container-fluid are fluid containers, which are always the full width of the viewport, which means the width of the device or the browser window. One of the reasons you use a container is because you get a 15 pixel padding on each side to make sure it works well with backgrounds and other elements. Now here's the breakpoints that the regular container will adjust to. First is anything smaller than 576 pixels, and so the bootstrap grid adjusts to content that is smaller than 576 pixels, and then some of these other breakpoints.

Adjusts to breakpoints

<576px | 576px | 768px | 992px | 1200px
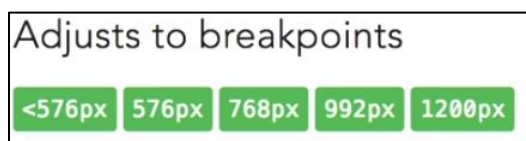
Rows and columns

Bootstrap's grid system allows up to 12 columns across the page. If you do not want to use all 12 column individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1

span 4	span 4	span 4

span 4	span 8

span 6	span 6

span 12

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.

Grid Classes

The Bootstrap 4 grid system has five classes:

> **.col**- (extra small devices - screen width less than 576px)
> **.col-sm**- (small devices - screen width equal to or greater than 576px)
> **.col-md**- (medium devices - screen width equal to or greater than 768px)
> **.col-lg**- (large devices - screen width equal to or greater than 992px)
> **.col-xl**- (xlarge devices - screen width equal to or greater than 1200px)

The classes above can be combined to create more dynamic and flexible layouts.

Tip: Each class scales up, so if you wish to set the same widths for **sm** and **md**, you only need to specify **sm**.

Screen Breakpoints

BP: sm >576px md >768px lg >992px xl >1200px

Grid System Rules

Some Bootstrap 4 grid system rules:

- Rows must be placed within a **.container** (fixed-width) or **.container-fluid** (full-width) 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 classes like **.row** and **.col-sm-4** are available for quickly making grid 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 12 available columns you wish to span. For example, three equal columns would use three **.col-sm-4**
- Column widths are in percentage, so they are always fluid and sized relative to their parent element

Example) Create row with different width:

Computer view

Smartphone view

```html
<div class="container-fluid">
    <!-- Control the column width, and how they should appear on different
devices -->
    <div class="row">
      <div class="col-sm-6" style="background-color:yellow;">50%</div>
      <div class="col-sm-6" style="background-color:orange;">50%</div>
    </div>
    <br>
    <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-sm-6" style="background-color:blue;">50%</div>
    </div>
  </div>
```

To take full advantage of the 12 column grid, you have to master the use of columns.

Multiple columns

The grid in Bootstrap is powerful because of the many ways that we can use it. One thing we can do is specify multiple-column breakpoints for each of our grid elements.

	Extra small < 576px	Small ≥ 576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-

Offsetting columns

Another way to control the grid is to offset columns in order to move their position on the grid. For example, a grid without offset looks as the following:
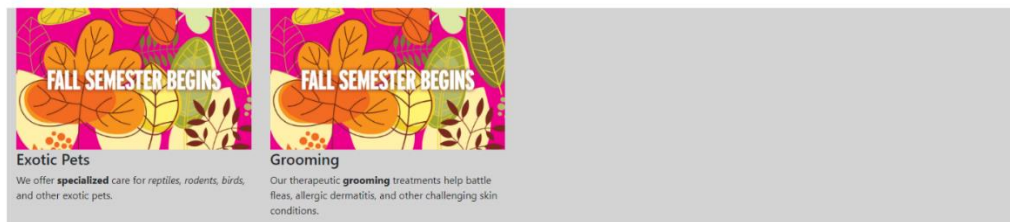

```html
<div class="row" style="background: lightgray;">
    <section class="col-md-3" >
        <img src="images/image.jpg" alt="sample image" style="width:100%;" >
        <h4>Exotic Pets</h4>
        <p>We offer <strong>specialized</strong> care for <em>reptiles,
    rodents, birds,</em> and other exotic pets.</p>
    </section>
    <section class="col-md-3">
        <img src="images/image.jpg" alt="sample image" style="width:100%;" >
        <h4>Grooming</h4>
        <p>Our therapeutic <span class="font-weight-bold">grooming</span>
    treatments help battle fleas, allergic dermatitis, and other challenging
    skin conditions.</p>
    </section>
</div><!-- end of row -->
```

HTML

offset class in columns is used to shift but the number of columns over. From the previous example, if we add offset to **`<section class="col-md-3 offset-sm-1">`** the webpage will look as the following:
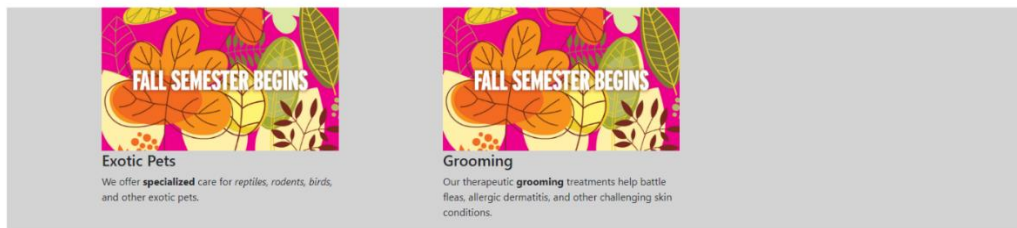
Nested column

You can easily put a whole new grade of columns inside an existing column. Now, that's called nesting. So, let's take a look at how that works. To nest columns, you simply create a new row inside an existing column. Now, this will create a new regular 12-column grid inside that existing column. And inside that column, you can use the same set of classes that you've been using so far. For example, we can create one main image with six sub-images as the following:


```
HTML
<div class="row">
    <section class="col-sm-8">
      <img src="images/image.jpg" alt="sample image">
      <h4>Exotic Pets</h4>
      <p>We offer <strong>specialized</strong> care for <em>reptiles,
      rodents, birds,</em> and other exotic pets.</p>
    </section>
    <section class="col-sm-4">
      <div class="row">
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
      </div> <!-- end of subnets row -->
    </section>
```

We can also set one main image with six sub-images below the main image:

Custom order

One of the more advanced ways that you can control layouts is to customize the order of elements. It helps you to reorder the way that columns appear and you can do this at different screen widths. Class **col order-2** change the order of the element in a row

For example, if we have a session with col

```
<section class="col">
```

Order

1. Exotic Pets
We offer **specialized** care for *reptiles, rodents, birds,* and other exotic pets.

2. Grooming
Our therapeutic **grooming** treatments help battle fleas, allergic dermatitis, and other challenging skin conditions.

3. General Health
Wellness and senior exams, ultrasound, x-ray, and dental cleanings are just a few of our general health services.

If we add order-3 to the first session, that session will be moved to the third position

```
<section class="col order-3">
```

Order

2. Grooming
Our therapeutic **grooming** treatments help battle fleas, allergic dermatitis, and other challenging skin conditions.

3. General Health
Wellness and senior exams, ultrasound, x-ray, and dental cleanings are just a few of our general health services.

1. Exotic Pets
We offer **specialized** care for *reptiles, rodents, birds,* and other exotic pets.

Grid Alignment

Because Bootstrap uses Flexbox to control layouts, there are many new classes to handle how elements align into one another as well as their containers.

Vertical Alignment
- Use in rows
- `align-items-ALN`
  ALN: start center end
- Works on nested cols

Individual Alignn
- Use in cols
- `align-self-ALN`
  ALN: start center end

Horizontal Alignment
- Use in rows
- Need col width
- `justify-content-ALN`
  ALN: start center end around between

Navigation bars

Every web project needs great navigation. Next to the grid, this is the most important component in Bootstrap. The different types of navigation components are **nav**, **tab**, **pill,** and **navbar**. **nav**s are parents to all the other navigation. Tabs and pills are the same type of component and help you create content within a page that changes when clicked. **navbar**s, on the other hand, are typically used for main navigation between pages.

Creating Navigation Bar using nav

There's a group of components called **nav**. Now, this component had a lot of changes in this new version of Bootstrap due to the implementation of Flexbox and, thankfully, became a lot simpler to use. Now to create navs you can use either the unordered lists or use nav and regular divs.

Using unordered list, you will need to call the **nav** class in the **<ul>, nav-item** in each list **<li>**, and **nav-link** in each link.

There's a couple of classes related to that. We can use justify-content-center to center the tabs, or justify-content-end to shift the tabs to the right side.

```
<div class="nav nav-pills justify-content-center">
```



```
<div class="nav nav-pills justify-content-end">
```
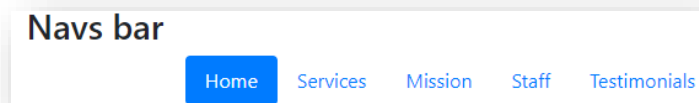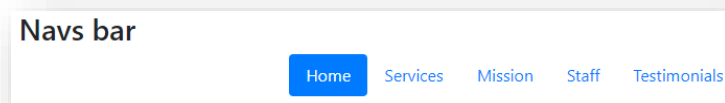

you can also try `nav-justified` to make all the links of the same size.
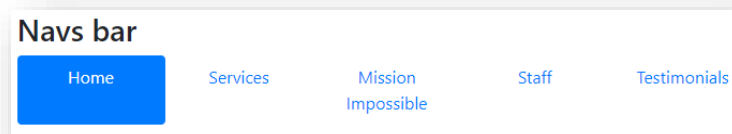
`nav-fills` makes all the links to fill into one line

Flex-column

`flex-column` makes all the links to automatically stack on top of one another.

you can actually specify a breakpoint at which these get converted back to rows. For example, stack all the links in medium size window.

```
<div class="nav nav-pills flex-column flex-md-row">
```

Using navbar classes

The navbar component is related to the nav component, so if you're familiar with navs, then creating navbars will be simple. the navbar class is the class that goes on the main container, and it can have a number of other elements inside it. Now, by default, navbar components will stack, so we need to add a navbar-expand, with optional breakpoints to control when the navbar is going to expand.

```
<nav class="navbar bg-dark navbar-dark">
    <div class='navbar-nav'> <!—div to store all the link →
    <a href="#" class="nav-item nav-link active">Home</a>
    <a href="#" class="nav-item nav-link">Services</a>
    <a href="#" class="nav-item nav-link">Mission Impossible</a>
    <a href="#" class=" nav-item nav-link">Staff</a>
    <a href="#" class="nav-item nav-link">Testimonials</a>
    </div>
    </nav>
```

Navs bar

Home

Services

Mission Impossible

Staff

Testimonials

by default, the navigation will always stack. Bootstrap is designed as a mobile first framework, so the default is to always do what would happen on a mobile device. So, we have an extra class that we need to add here, and it's called `navbar-expand` with a specific breakpoint.

```
<nav class="navbar bg-dark navbar-dark navbar-expand-sm">
```

Navbar brand and text

In addition to the `navbar nav` you can add other components to your navigation. The simplest to use is called the navbar brand and it's used for text or logos and will get an enhanced look. Another simple component is the navbar text component and you can add inline text that works well with the rest of the navigation. So, both of these are going to go right next to the navbar nav component. Navbar-brand is the class that you add when you want to add a logo or some text.

```
<div class='container'>
      <nav class="navbar bg-dark navbar-dark navbar-expand-sm">
        <!-- Create a div for navbar-brand -->
        <a class=navbar-brand href='#'>
            <img src='images/wisdompetlogo.svg' style="width: 40px;"
alt="wisdow pet logo"/>
        </a>
        <!-- nav list-->
        <div class='navbar-nav'>
          <a href="#" class="nav-item nav-link active">Home</a>
          <a href="#" class="nav-item nav-link">Services</a>
          <a href="#" class="nav-item nav-link">Mission Impossible</a>
          <a href="#" class=" nav-item nav-link">Staff</a>
          <a href="#" class="nav-item nav-link">Testimonials</a>
        </div>
      </nav>
    </div>
```
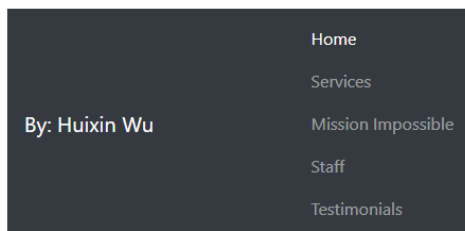
Navs bar

	Home
	Services
By: Huixin Wu	Mission Impossible
	Staff
	Testimonials

Navs bar

The `navbar-brand` can also work with link and/or image

```
<a class=navbar-brand href='#'>
          <img src='images/wisdompetlogo.svg' style="width: 40px;"
alt="wisdow pet logo"/>
 </a>
```

Example) Create a navigation bar with three external links


```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a href="https://www.qcc.cuny.edu/" target="_blank"
         class="nav-link active">Queensborough CC</a>
      </li>
      <li class="nav-item">
        <a href="https://www.qc.cuny.edu/" target="_blank"
         class="nav-link">Queens College</a>
      </li>
      <li class="nav-item">
        <a href="https://www.lgcc.cuny.edu/" target="_blank"
         class="nav-link">LaGuardian CC</a>
      </li>
    </ul>
  </div>
</nav>
```



```
<nav class="navbar navbar-expand-sm bg-warning navbar-dark">
   <div class="container-fluid">
     <a class="navbar-brand" href="https://www.qcc.cuny.edu/">
        <img src="cat.jpg" alt="cat logo" style="width:80px; height:60px;"
         class="rounded-pill">
```

```
        </a>
    </div>
 </nav>
```



```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark" style="margin-
bottom:20em;">
    <div class="container-fluid">
        <a class="navbar-brand" href=https://www.qcc.cuny.edu/ target="_blank">
            <img src="cat.jpg" alt="cat logo" style="width:80px; height:60px;"
            class="rounded-pill">
        </a>
        <!-- Navigation links -->
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#collapsibleNavbar"><span class="navbar-toggler-icon">
        </span>
        </button>
        <div class="collapse navbar-collapse" id="collapsibleNavbar">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="#">To top of window</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#BI">Bootstrap Images</a>
              </li>

              <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href=" https://www.cuny.edu"
              role="button" data-bs-toggle="dropdown">CUNY Quick Links</a>
              <ul class="dropdown-menu">
                  <li><a class="dropdown-item"
href="https://www.qcc.cuny.edu/"> Queensborough CC</a></li>
                  <li><a class="dropdown-item"
href="https://www.qc.cuny.edu/">Queens College</a></li>
                  <li><a class="dropdown-item"
href="https://www.lgcc.cuny.edu/">LaGuardia CC</a></li>
                </ul>
              </li>
            </ul>
        </div><!-- end of nav links-->
      </div><!-- end of nav container-->
    </nav>
```

Create collapsible content

Every so often, we need to create a navigation that collapses when the width of the device or the window is really small. That's because navigation can take up a lot of room vertically on a small device and often, we want to make sure that that room isn't taken over unless the user wants it. So in order to do that, we create these collapsible sections of content and to do that in Bootstrap, you essentially need two parts, the thing that you want to collapse and some element that will control that content.

So we need to work on two different parts. For the collapsible content, you're going to need a couple of classes, the first one is the generic collapse class that goes on the content that you of course want to collapse. And in addition to that, you're also going to use another class called navbar collapse because most of the time when you're using this feature, you're going to be collapsing a navbar. Now there are some other things that you need to do and we need to tie the element that we are collapsing which would be this element with the thing that is collapsing this element which will be a little button that's known as the hamburger menu button.

```
<nav class="navbar bg-dark navbar-dark navbar-expand-sm">
        <!-- Create a div for navbar-brand -->
        <a class=navbar-brand href='#'>
            <img src='images/wisdompetlogo.svg' style="width: 40px;" alt="wisdow pet
logo"/>
        </a>
        <!-- Create a hamburger button to control the collapseTab-->
        <button class='navbar-toggler' type='button' data-bs-toggle='collapse' data-
bs-target='#collapseTab' aria-controls="collapseTab" aria-expanded='false' aria-
label='Toggle navigation'>
            <span class='navbar-toggler-icon'></span>
        </button> <!--end of collapse button -->

        <!-- Create a div to specified the collapse session -->
        <div class='collapse navbar-collapse'  id='collapseTab'>
        <!-- nav list-->
        <div class='navbar-nav'>
          <a href="#" class="nav-item nav-link active">Home</a>
          <a href="#" class="nav-item nav-link">Services</a>
          <a href="#" class="nav-item nav-link">Mission Impossible</a>
          <a href="#" class=" nav-item nav-link">Staff</a>
          <a href="#" class="nav-item nav-link">Testimonials</a>
        </div>
      </div> <!-- end of collapse-->
    </nav>
```
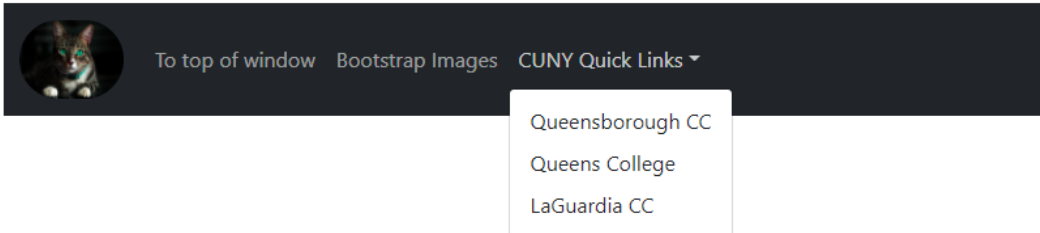

Cards

Cards are a new design element in Bootstrap 4 and it can help you lay out your content into card-like containers that look great and have as a small outline around them. There are a ton of options in terms of the classes that you can use.

You can modify the card content:

- `card-text`

- `card-title`

- `card-subtitle`

- `card-link`

- `card-img`

```
<section class="card mb-5" id="drwinthrop">

    <div class='card-body'> <!-- add a container to card elements -->
      <img class="card-img img-fluid" src="images/doctor-winthrop.jpg"
alt="Doctor Winthrop Photo">
    <h2 class='card-title'>Dr. Stanley Winthrop</h2>
    <h5 class='card-subtitle'>Behaviorist</h5>
      <p class='card-text'>Dr. Winthrop is the guardian of Missy, a three-
      year old Llaso mix, who he adopted at the shelter. Dr. Winthrop is
      passionate about spay and neuter and pet adoption, and works tireless
      hours outside the clinic, performing free spay and neuter surgeries for
      the shelter.</p>
    <a class='card-link' href="#">About Me</a>
    <a class='card-link' href="#">My Pets</a>
    <a class='card-link' href="#">Client Slideshow</a>
  </div>
  </section>
```


You can also add colors

bg-COLOR for backgrounds

primary secondary success danger warning info light dark white

border-COLOR for outlines

primary secondary success danger warning info light dark

text-COLOR for text

primary secondary success danger warning info light dark

Card content classes

The basic container should have a card class, and we should at least have an element that has a card body as a container of some of the content in your cards.

You can create a card header and footer by using classes card-header and card-footer

```
<section class="card mb-5 bg-light" id="drwinthrop">
    <div class='card-header'> <!-- card header-->
      <h2 class='card-title'>Dr. Stanley Winthrop</h2>
      <h5 class='card-subtitle'>Behaviorist</h5>
    </div>
    <div class='card-body'> <!-- add a container to card elements -->
    <img class="card-img img-fluid" src="images/doctor-winthrop.jpg"
alt="Doctor Winthrop Photo">
      <p class='card-text'>Dr. Winthrop is the guardian of Missy, a three-
      year old Llaso mix, who he adopted at the shelter. Dr. Winthrop is
      passionate about spay and neuter and pet adoption, and works tireless
      hours outside the clinic, performing free spay and neuter surgeries for
      the shelter.</p>
    <div class='card-footer'>  <!-- card footer-->
      <a class='card-link' href="#">About Me</a>
      <a class='card-link' href="#">My Pets</a>
      <a class='card-link' href="#">Client Slideshow</a>
    </div>
  </div>
  </section>
```

You can also work with images in a card

- `card-img`

- `card-img-top`

- `card-img-bottom`

- `card-img-overlay`

```
<section class="card mb-5 bg-primary text-white" id="drchase" >
      <img class="card-img-top img-fluid" src="images/doctor-chase.jpg"
alt="Doctor Chase Photo">
      <div class='card-body'>
          <h2 class='card-title'>Dr. Elizabeth Chase</h2>
          <h5 class='card-subtitle'>Dentistry</h5>
            <p class='card-text'>Dr. Chase spends much of her free time
            helping the local bunny rescue organization find homes for
            bunnies, such as Kibbles - a Dalmatian bunny who is part of the
            large Chase household, including two dogs, three cats, and a
            turtle.</p>
          <a class='card-link text-warning' href="#">About Me</a>
          <a class='card-link text-warning' href="#">My Pets</a>
          <a class='card-link text-warning' href="#">Client Slideshow</a>
      </div>
  </section>
```
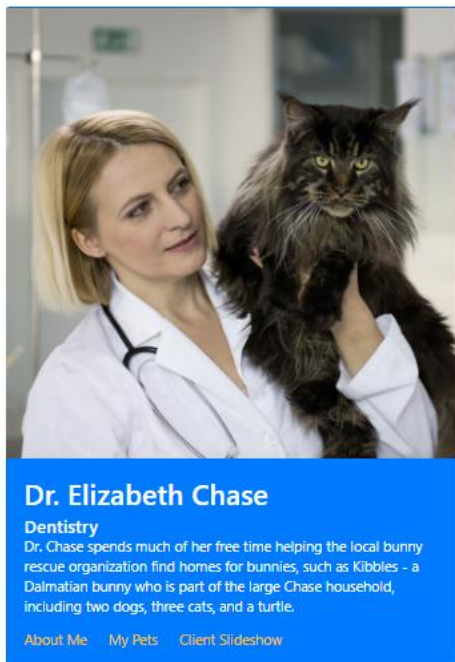
Normal list group container class that will go around the elements, and then inside that, each of the items that is part of the list gets a list-group-item. Finally there is a class called list-group-flush.

That allows you to take the content that is in the list group, and extend it to the edge of the card.

- `list-group` container

- `list-group-item`

- `list-group-flush`

```html
<section class="card mb-5" id="drsanders">
    <div class='card-body'>
    <img class="card-img img-fluid" src="images/doctor-sanders.jpg"
alt="Doctor Sanders Photo">
        <h2 class='card-title'>Dr. Kenneth Sanders</h2>
        <h5 class='card-subtitle'>Nutritionist</h5>
            <p class='card-text'>Leroy walked into Dr. Sanders front door
            when she was moving into a new house. After searching for weeks
            for Leroy's guardians, she decided to make Leroy a part of her
            pet family, and now has three cats.</p>
        <div class='list-group list-group-flush'>
            <a class='list-group-item' href="#">About Me</a>
            <a class='list-group-item' href="#">My Pets</a>
            <a class='list-group-item' href="#">Client Slideshow</a>
        </div>
    </div>
  </section>
```


Dr. Kenneth Sanders
Nutritionist
Leroy walked into Dr. Sanders front door when she was moving into a new house. After searching for weeks for Leroy's guardians, she decided to make Leroy a part of her pet family, and now has three cats.

About Me

My Pets

Client Slideshow

Card layout

So, let's take a look at how we control the layout of different cards. And what you do in Bootstrap is you create a container that is going to have a number of cards in it and that container should get some classes. And the first one I'll talk about is called card group. Card group will place the cards side by side, and they will have a shared order. If you don't want a shared order, you can use card deck. It's pretty much the same thing as card group, puts everything side by side, but it does have a little of a spacing in between the cards.

And finally, there is something called card columns, this makes a really interesting layout where cards align. If you don't like any of these, you can use traditional rows and columns.

```
<div class="card-group">
```



```
<div class="card-deck">
```

```
<div class="card-columns">
```
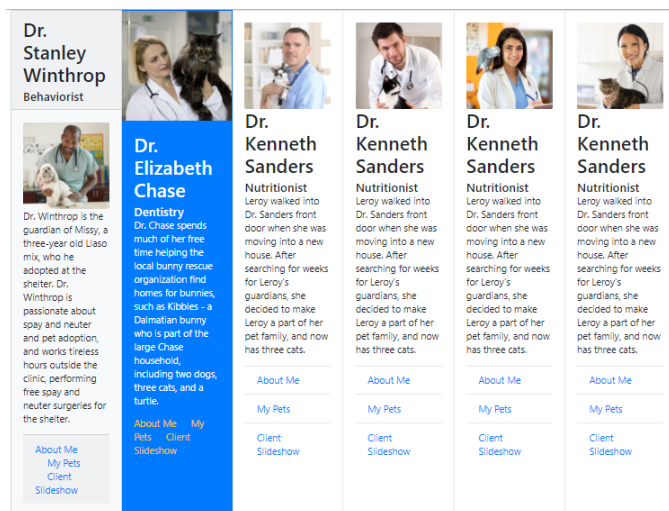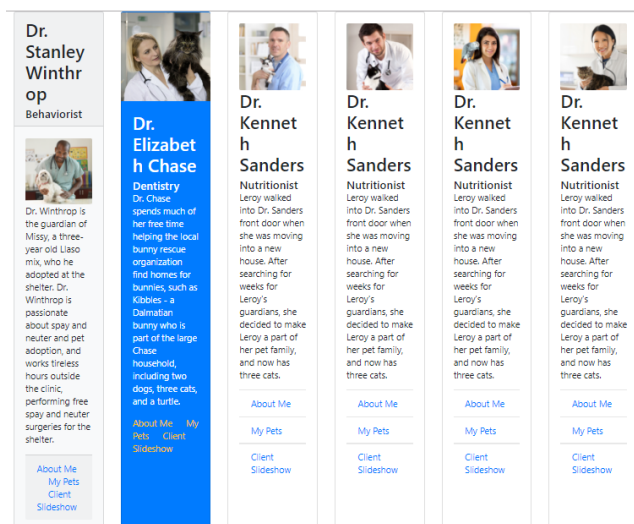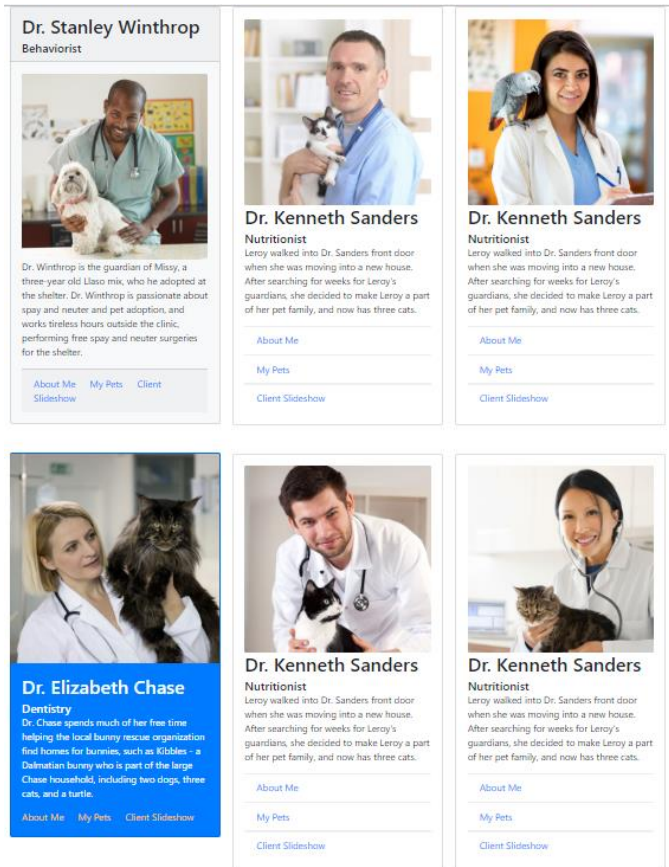

Use can also use traditional row and col

```
<div class='container'>
  <div class='row'>
    <div class='col-lg-6'>
      <section class="card mb-5 bg-light" id="drwinthrop">▭
      </section>
    </div>
    <div class='col-lg-6'>
      <section class="card mb-5 bg-primary text-white" id="drchase" >▭
    </div>
    <div class='col-lg-6'>
      <section class="card mb-5" id="drsanders">▭
    </div>
    <div class='col-lg-6'>
      <section class=" card mb-5" id="drgardner">▭
      </div>
    <div class='col-lg-6'>
      <section class="card mb-5" id="drsanders">▭
    </div>
    </div><!--end of card-row -->
</div><!-- content container -->
```

Working with interactive components

Alerts

Alerts are messages with special contextual styles that are designed to display in great looking boxes. They are optionally dismissed and can have any sort of markup so let us take a look at them. Now setting up alerts is pretty simply. You start out with the basic alert class on a container. There's a few contextual classes that you can use for color. So for example, success, info, warning, and danger. A few classes are available for internal content, like headings as well as links.

Bootstrap provides an easy way to create predefined alert messages using the class **alert**. Alerts are created with the **alert** class, followed by one of the contextual classes **alert-success, alert-info, alert-warning, alert-danger, alert-primary, alert-secondary, alert-light** or **alert-dark**

Success! This alert box could indicate a successful or positive action.

Info! This alert box could indicate a neutral informative change or action.

Warning! This alert box could indicate a warning that might need attention.

Danger! This alert box could indicate a dangerous or potentially negative action.

Primary! Indicates an important action.

Secondary! Indicates a slightly less important action.

```
<p class="alert alert-success"><strong>Success!</strong> This alert box
could indicate a successful or positive action.</p>

<p class="alert alert-info"><strong>Info!</strong> This alert box could
indicate a neutral informative change or action.</p>

<p class="alert alert-warning"><strong>Warning!</strong> This alert box
could indicate a warning that might need attention.</p>
```

```
   <p class="alert alert-danger"><strong>Danger!</strong> This alert box could
indicate a dangerous or potentially negative action.</p>

   <p class="alert alert-primary"><strong>Primary!</strong> Indicates an
important action.</p>

   <p class="alert alert-secondary"><strong>Secondary!</strong> Indicates a
slightly less important action.</p>
```

Alert Links

Add the **alert-link** class to any links inside the alert box to create "matching colored links"

Example) Create an information alert box with an external hypelink

Course Registration Info: Register now for summer courses: **click here to register**.

```
<p class="alert alert-info">
      <strong>Course Registration Info: </strong> Register now for summer
      courses: <a href="https://www.qcc.cuny.edu/" class="alert-link text-
      danger">click here to register</a>.
</p>
```

Closing Alerts

To close the alert message, add an **alert-dismissible** class to the alert container. Then add **class="btn-close"** and **data-bs-dismiss="alert"** to a link or a button element (when you click on this the alert box will disappear).

Example) create a dismissible alert box to display a promotion.

Semi-Annual Sale! Take 50% OFF TODAY with order of more than $19.99 ✕

```
<div class="alert alert-secondary alert-dismissible">
    <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
    <strong>Semi-Annual Sale!</strong> Take <strong class="text-danger
    display-6">50%</strong> OFF TODAY with order of more than $19.99
</div>
```

Animated Alerts

The **fade** and **show** classes adds a fading effect when closing the alert message:

```
<div class="alert alert-secondary alert-dismissible fade show">
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
  <strong>Semi-Annual Sale!</strong> Take <strong class="text-danger display-
  6">50%</strong> OFF TODAY with order of more than $19.99
</div>
```

Carousels

Carousels are the most popular Bootstrap component. It essentially allows you to create a slideshow that auto-advances and it gives you some controls to let you advance through the images. Now there's a ton of classes for the carousel so let's take a look at all of them.

- So the first thing you need to do is have a main component that has a class of **carousel**.
- The main component needs to have at least a data-ride attribute with the word carousel. This will automatically activate the advancing of the carousel when the page loads.
- There will be another container that has a class of **carousel-inner**. This is what's going to have all your photos.
- Now each of the photos should have a **div** wrapping the photos with a class of **carousel-item**.
- In addition to that, we have a few options. The first element that you want to show up when the carousel starts should have a class of **active**. That would go in the **div**, and you do want to crop and size the photos. Otherwise, the carousel will jump up and down.
- We can add captions to a carousel with the **carousel-caption** class. So you would put another **div**, add this class to it, and that will go inside the **div** that has the image. So this is sort of what it looks like.

Example) create a carousel/slideshow with three images and will automatically change the image.

First, we create the carrousel container with class carousel slide and **data-bs-ride="carousel"**

Now, we have a container that will have all the pictures. This container will have **class="carousel-inner">**

```
<!-- Carousel -->
<div  class="carousel slide" data-bs-ride="carousel">
      <div class="carousel-inner">

      </div>
</div>
```

After it, we create a division for each image in the carousel. Each image division will have a class carousel-item

```
<!-- Carousel -->
<div  class="carousel slide" data-bs-ride="carousel">
   <div class="carousel-inner">
           <div class="carousel-item active">
             <img src="cat.jpg" alt="Cat image" style="width:100%">
           </div>
           <div class="carousel-item">
             <img src="dog.jpg" alt="Dog image" style="width:100%">
           </div>
           <div class="carousel-item">
             <img src="fish.jpg" alt="fish image" style="width:100%">
           </div>
      </div>
</div>
```

So in addition to that, there's a number of navigation options so you can add these little arrows that will let you advance to the previous and next slide.

- The way you do that is by tying the carousel to the navigation with a **data-target** attribute. Whenever you use navigation, you need to make sure that you have that, and then you have two buttons that you can add.
- One with a class of **carousel-control-prev** and then you add an icon inside. And then also, you add a **carousel-control-next** with its icon as well.
- These two buttons need to sit within the `carouser` container.
- Within each button, we can have a span to display the next and previous icon. The classed for each icons are **carousel-control-prev-icon** and **carousel-control-next-icon**

Next and previous buttons in a carousel

From the previous code, let us create the previous button first. For it, we can use a button element with class="carousel-control-prev", data-bs-target="#ID_of_carousel_container", and data-bs-slide="prev".

Example) Add the next and previous buttons to the previous slideshow of three images.

```
<!-- Carousel -->
<div  class="carousel slide" data-bs-ride="carousel" id="slideshow">
   <div class="carousel-inner">
           <div class="carousel-item active">
             <img src="cat.jpg" alt="cat image" style="width:100%">
           </div>
           <div class="carousel-item">
             <img src="dog.jpg" alt="dog image" style="width:100%">
           </div>
```

```
                <div class="carousel-item">
                    <img src="fish.jpg" alt="fish image" style="width:100%">
                </div>
        </div>
    <!-- Left and right controls/icons -->
    <button class="carousel-control-prev" type="button" data-bs-slide="prev"
        data-bs-target="#slideshow" data-bs-slide="prev">
    </button>
</div>
```

After it, we can add a span within the button to show the previous icon:

```
<!-- Carousel -->
<div  class="carousel slide" data-bs-ride="carousel" id="slideshow">
    <div class="carousel-inner">
            <div class="carousel-item active">
                <img src="cat.jpg" alt="cat image"style="width:100%">
            </div>
            <div class="carousel-item">
                <img src="dog.jpg" alt="dog image" style="width:100%">
            </div>
            <div class="carousel-item">
                <img src="fish.jpg" alt="fish image" style="width:100%">
            </div>
        </div>
    <!-- Left and right controls/icons -->
    <button class="carousel-control-prev" type="button" data-bs-slide="prev"
        data-bs-target="#slideshow" data-bs-slide="prev">
        <span class="carousel-control-prev-icon"></span>
    </button>
</div>
```

After it, we can create another button to control the next view. To create the next button, we create the same script structure as we create the previous button but instead of **prev**, we type **next**

```
<div  class="carousel slide" data-bs-ride="carousel" id="slideshow">
    <div class="carousel-inner">
            <div class="carousel-item active">
                <img src="cat.jpg" alt="cat image" style="width:100%">
            </div>
            <div class="carousel-item">
                <img src="dog.jpg" alt="dog image" style="width:100%">
            </div>
            <div class="carousel-item">
                <img src="fish.jpg" alt="fish image" style="width:100%">
            </div>
        </div>
    <!-- Left and right controls/icons -->
    <button class="carousel-control-prev" type="button" data-bs-slide="prev"
        data-bs-target="#slideshow" data-bs-slide="prev">
        <span class="carousel-control-prev-icon"></span>
    </button>
```

```
    <button class="carousel-control-next" type="button" data-bs-slide="next"
        data-bs-target="#slideshow" data-bs-slide="next">
        <span class="carousel-control-next-icon"></span>
    </button>
</div>
```

Indicators in a carousel

You can also put indicators underneath each photo so that people can jump to any photo in the slide. Do that:

- Create a section or another **div** with a class of **carousel-indicators**.
- Within **carousel-indicators** should have button elements to work as the indicator. These buttons also get a **data-bs-target** to the carousel container, by using the id name, so that Bootstrap knows which carousel you want to advance.
- Also add a slide number to the **data-bs-slide-to** because, as you'll see, the indicators are sort of at the bottom, so this tells Bootstrap which photo to slide to.

```
<div  class="carousel slide" data-bs-ride="carousel" id="slideshow">
   <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="cat.jpg" alt="cat image" style="width:100%">
    </div>
    <div class="carousel-item">
      <img src="dog.jpg" alt="dog image" style="width:100%">
    </div>
    <div class="carousel-item">
      <img src="fish.jpg" alt="fish image" style="width:100%">
    </div>
     </div>
  <!-- Left and right controls/icons -->
  <button class="carousel-control-prev" type="button" data-bs-slide="prev"
   data-bs-target="#slideshow" data-bs-slide="prev">
      <span class="carousel-control-prev-icon"></span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-slide="next"
   data-bs-target="#slideshow" data-bs-slide="next">
      <span class="carousel-control-next-icon"></span>
  </button>
  <!-- Indicators/dots -->
  <div class="carousel-indicators">
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="0"
      class="active"></button>
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="1">
      </button>
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="2">
      </button>
  </div>
</div>
```

With captions

Add captions to your slides easily with the `carousel-caption` element within any **carousel-item**.

```html
<div  class="carousel slide" data-bs-ride="carousel" id="slideshow">
   <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="cat.jpg" alt="cat image" style="width:100%">
      <div class="carousel-caption">
         <h5>Cat</h5>
         <p>The cat is a domestic species of small carnivorous mammal.</p>
      </div>
    </div>
    <div class="carousel-item">
      <img src="dog.jpg" alt="dog image" style="width:100%">
    </div>
    <div class="carousel-item">
      <img src="fish.jpg" alt="fish image" style="width:100%">
    </div>
    </div>
  <!-- Left and right controls/icons -->
  <button class="carousel-control-prev" type="button" data-bs-slide="prev"
   data-bs-target="#slideshow" data-bs-slide="prev">
      <span class="carousel-control-prev-icon"></span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-slide="next"
   data-bs-target="#slideshow" data-bs-slide="next">
      <span class="carousel-control-next-icon"></span>
  </button>
  <!-- Indicators/dots -->
  <div class="carousel-indicators">
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="0"
      class="active"></button>
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="1">
      </button>
      <button type="button" data-bs-target="#slideshow" data-bs-slide-to="2">
      </button>
  </div>
</div>
```

They can be easily hidden on smaller viewports, as shown below, with optional display utilities. We hide them initially with d-none and bring them back on medium-sized devices with d-md-block.

data-attribute

In order to set it up, you can do that with either a **data- attribute** and we talked about it earlier. It's called a data ride equals carousel. That will activate the carousel and make it start, or you can use JavaScript like this and pass it along some additional options.

Using data-attribute, the **interval** controls the amount of time in between each photo in milliseconds, **pause** will automatically pauses a carousel when somebody moves their mouse on

top of that carousel. So by default, that will be on. You can turn it off by giving it a value of null if you want to.

- `interval` : 5000

- `pause` : hover | null

- `ride` : false

- `wrap` : true

Also, as I mentioned, the data **ride** attribute, you can set that to false if you want to. By default when you say data ride equals carousel, that will be on and that causes the carousel to automatically slide. If you don't want it to automatically advance, you can turn that false. Finally, there is an option to **wrap** the carousel which means that when it reaches the last photo, it'll go to the first photo.

```
$('.carousel').carousel({
  interval: 2000
})
```

Modals

Modals allow you to show and hide content that is in an element showing up as an overlay on top of your page. Like with other components, modals have two parts: the trigger, which can be a button or a link, as well as a piece of content that you want to show in the modal. If you use a link, you'll need to target an ID that you want to use as your modal. If you're using a button, then you use the data-target attribute. The link, or the button, will need a data-toggle="modal" attribute. Now the content you want to show as the modal needs to have the modal class. So this is our main container for the entire element.

- Button or Link

- `#id` or `data-target`

- `data-toggle="modal"`

- `modal`

In addition to the modal class, there's some other classes that set up the structure of how your modal works. Modal-dialog is an extra container that is used for some additional spacing. There's another container class called modal-content that you can use for the main content of the modal. You can also use an optional header for a title in your modal, and you should probably use a header in all your modals.

Now the modal-body is where you put your main content, and there is an optional footer for things like close buttons. There's a couple of other options that you can use if you are using a headline, you can also use modal-title to handle the headline that you're using for that title. You should have at least one element with at a data-dismiss attribute of modal, and that will get rid of the modal and take you back to your page.

- `modal-dialog`

- `modal-content`

- `modal-header`

- `modal-body`

- `modal-footer`

Modal Options

- `modal-title`

- `data-dismiss="modal"`

Popovers

PopOvers are another component that lets you display additional content and are triggered by events like clicks. Now they're almost exactly like tool tips so if you're coming from that video, this is going to be pretty similar but they have a slightly different style. In order to set these up, you simply add a data-toggle attribute of popover to an element, and then you can add a title attribute with some text to it as well as a data-content where you add some additional content.

So the text would become like, the headline of the popover, and then the data content would be the sub-content underneath that, because they do have more of a headline section in them. Now in order to get these started, you do need to activate them, and you have to do that through jQuery so you use the jQuery, sort of start-up function, and then somewhere in there, you can activate all of the popups on the current page by targeting the data-toggle=popover attribute, and then you use the popover method right here, and you can optionally pass along some of the setup options.

```
$(function () {
    $('[data-toggle="popover"]').popover({ OPTIONS })
})
```

Example)

```
<div>
    <p>Queensborough Community College is a community college in
      <a href='#' data-toggle='tooltip' data-placement='top' title='You place
over here what you want to show up'>Bayside, Queens, New York.</a> One of
seven community colleges within the City University of New York system,
Queensborough enrolls more than 15,000 students.</p>
  </div>

  <!-- create a button to work with popover -->
    <button type='button' class='btn btn-info' data-toggle='popover' data-
trigger='hover' title='Check on PopOver' data-content='This is text that will
display when it popover'> PopOver</button>
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="js/jquery.slim.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<!-- script toggle for tooltips-->
<script>
$(function(){
  $('[data-toggle="tooltip"]').tooltip();
  $('[data-toggle="popover"]').popover({placement:"top"});
});
</script>
```
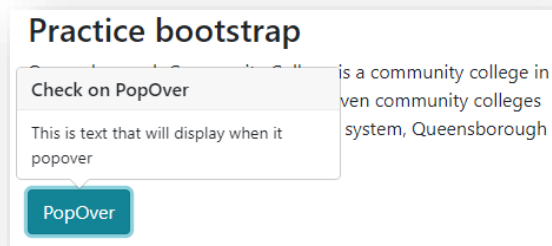

Add tooltips

Tooltips are a great way to show some simple additional information on links and buttons. Now you should only use tooltips on HTML elements that are focusable like links and foreign controls because of problems with usability. In order to set these up, you add a data-toggle of tooltip to an element and then add a title attribute with some text in it.

Tooltips are configured in one of two ways, you can use a data attribute or JavaScript. Data is a lot easier and it's what I'll be showing you here, however you can do anything that you can with the data attributes in JavaScript as well. Tooltips are not turned on by default so if you do want to use them, you will still need to use JavaScript, even if you're using data attributes. Now if you do want to configure them with JavaScript, you would use something like this.

```
$(function () {
  $('[data-toggle="tooltip"]').tooltip({ OPTIONS })
})

<div class="tooltip bs-tooltip-top" role="tooltip">
  <div class="arrow"></div>
  <div class="tooltip-inner">
    Some tooltip text!
  </div>
</div>
```

There are different options that you can add to the tooltips. Some of the basic options that you can add to the tooltips are:

placement `top` `right` `bottom` `left`

trigger `click` `hover` `focus`

Example)

```
<div>
    <p>Queensborough Community College is a community college in
       <a href='#' data-toggle='tooltip' data-placement='top' title='You place
       over here what you want to show up'>Bayside, Queens, New York.</a> One
       of seven community colleges within the City University of New York
       system, Queensborough enrolls more than 15,000 students.</p>
  </div>
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="js/jquery.slim.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<!-- script toggle for tooltips-->
<script>
$(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>
```

Scrollspy

Scrollspy is a really cool component that allows you to keep track of the scroll of the page and modifies classes according to the position of your elements. So the way that you use it is by creating a data attribute called scroll, and you do that on the container of the element that you want to track the scrolling in. Now that element also needs to have a position of relative, and you also need to target that element with a data target attribute, and then an ID. Data target is usually how we put together what we are doing with the object that we're targeting.

- `data-spy="scroll"`
- `position: relative`
- `data-target="ID"`
- `fixed-top`
- `data-offset`

Example)

```
<body data-spy='scroll' data-target='#navbar-site' data-offset='80'>
<nav id="navbar-site" class="fixed-bottom navbar navbar-dark bg-dark navbar-
expand-sm">
  <div class="container">
    <ul class="navbar-nav">
      <li class="nav-item"><a class="nav-link"
href="#mission">Mission</a></li>
      <li class="nav-item"><a class="nav-link"
href="#services">Services</a></li>
      <li class="nav-item"><a class="nav-link" href="#staff">Staff</a></li>
      <li class="nav-item"><a class="nav-link"
href="#testimonials">Testimonials</a></li>
    </div><!-- navbar-nav -->
  </ul><!-- container -->
</nav>
```

CSS variables

Bootstrap has some features that allows you to work with a new feature of CSS called CSS Variables. Now CCS Variables are a newer feature within CSS that allows you to store a set of property values that you can reuse within your CSS. Now this version of Bootstrap provides pre-written CSS variables that can be used in developing your projects and customizing your colors. CSS Variables are so new that browser support is an issue. They're not supported in any version of Internet Explorer, and only on very new versions of the other browsers.

Syntax

```
<style>
  :root{
    --primaryColor:#25FF64;
    }
  </style>
  <h1 style="color:var(--primaryColor);">Our Commitment</h1>
```

Reference

Window size, margin, and padding

Bootstrap has a wide range of responsive margin and padding utility classes. They work for all breakpoints:

xs (<=576px), **sm** (>=576px), **md** (>=768px), **lg** (>=992px) or **xl** (>=1200px))

The classes are used in the format:

{property}{sides}-{size} for xs & **{property}{sides}-{breakpoint}-{size}** for sm, md, lg, and xl.

m - sets margin

p - sets padding

---

t - sets margin-top or padding-top

b - sets margin-bottom or padding-bottom

l - sets margin-left or padding-left

r - sets margin-right or padding-right

x - sets both padding-left and padding-right or margin-left and margin-right

y - sets both padding-top and padding-bottom or margin-top and margin-bottom

blank - sets a margin or padding on all 4 sides of the element

---

0 - sets **margin** or **padding** to 0

1 - sets **margin** or **padding** to .25rem (4px if font-size is 16px)

2 - sets **margin** or **padding** to .5rem (8px if font-size is 16px)

3 - sets **margin** or **padding** to 1rem (16px if font-size is 16px)

4 - sets **margin** or **padding** to 1.5rem (24px if font-size is 16px)

5 - sets **margin** or **padding** to 3rem (48px if font-size is 16px)

auto - sets margin to auto

Bibliography

Duckett, J. (2016). *HTML and CSS Design and build websites.* Indianapolis: John Wiley and Sons Inc.

Duckett, J. (2016). *JavaScript and JQuery: interactive front-end developer.* Indianapolis: John Wiley and Sons Inc.

HTML, CSS, JavaScript, JQuery, and Bootstrap. (2017, June). Retrieved from w3schools: www.w3schools.com

Some material compiled from www.lynda.com (May 2018), www.w3schools.com (2020), www.coursera.org (2020)

IMPORTANT NOTE
The materials used in this manual have the author's rights and are for educational use only.