Bootstrap

Introduction

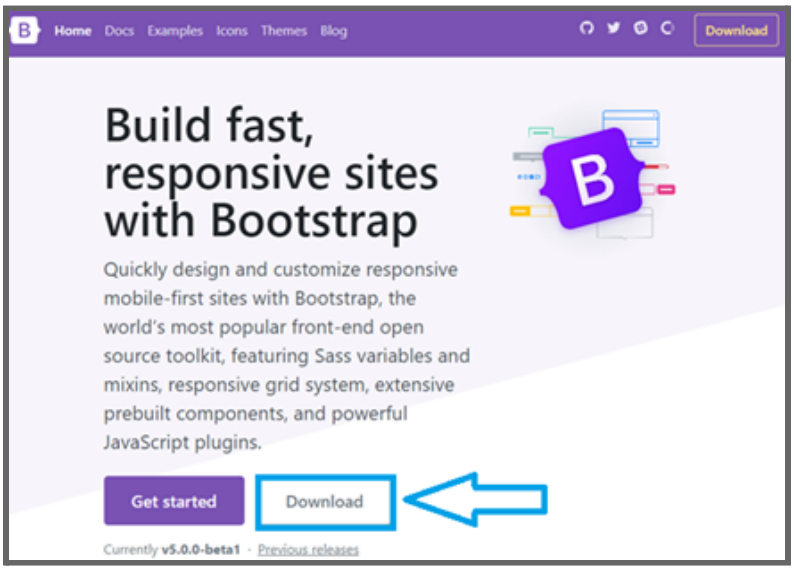
Bootstrap is a big collection of handy, reusable code written in HTML, CSS, and JavaScript. It's a front-end development framework that enables developers and designers to quickly build fully responsive websites. It is a free and open-source project, hosted on GitHub, and originally created by (and for) Twitter.

Bootstrap saves you from writing lots of CSS code, giving you more time to spend on designing web pages. It is flexible and easy to use. Its main advantages are that it is responsive by design, it maintains wide browser compatibility, it offers consistent design by using reusable components, and it is very easy to use and quick to learn.

Installing Bootstrap 4

You can install Bootstrap in two ways:

- The standard way is a very simple method where you can easily download combined and minified JavaScript and CSS bundles that can be later used for your web application project.



- Another way is through **CDNS** or Content Delivery Networks. If you use CDNs, then you don't even have to download Bootstrap to your PC. All you need to do is to add the following code into your existing HTML code:

1. Copy-paste this <link> into your <head> before all other stylesheets to load the Bootstrap CSS:

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/boots
trap.min.css" rel="stylesheet"
integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTf
AAsrekwKmP1" crossorigin="anonymous">
```

2. As many of the bootstrap components require the use of **JavaScript** to function, specifically, they require their own JavaScript plugins. Place the following <script> near the end of your pages, right before the closing </body> tag, to enable it.

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstr
ap.bundle.min.js"
integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf
80OJFdroafW" crossorigin="anonymous"></script>
```

Bootstrap Breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as grid levels, for building responsively. These breakpoints can also be customized, but for that, you'll need to change Bootstrap's CSS files.

In Bootstrap, breakpoints are the building blocks of making a responsive design. You can refer to this table for all the bootstrap breakpoints.

Breakpoint	Class infix	Dimentions
X-Small	None	0-576px
Small	sm	>=576px
Mediium	md	>=768px
Large	lg	>=992px
Extra Large	xl	>=1200px
Extra Extra large	xxl	>=1400px

You can use them to control the layout of your website according to different screen sizes.

You don't need to remember all these dimensions; you just need to remember the class names of respective breakpoints.

Bootstrap Grid System

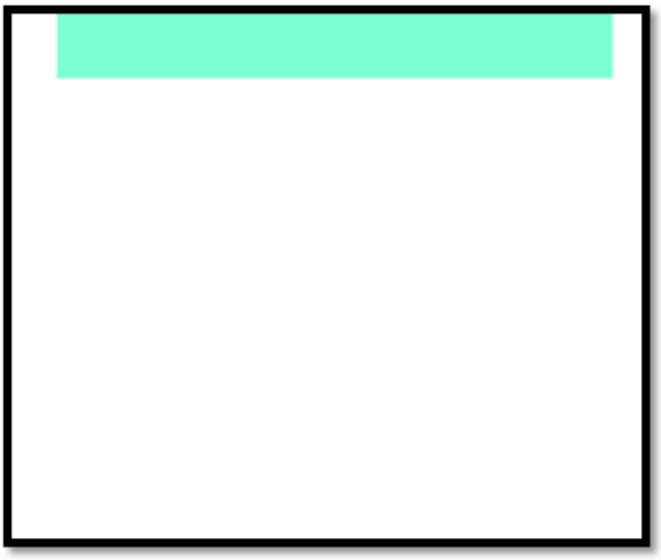
Bootstrap uses a flexbox-grid system which consists of a series of containers, rows, and columns to build its layout and align the content of the page.

- There is a 12-column grid system
- All the items must be in rows
- To prevent things from going full width use the wrapper class
- You can put rows inside of rows

Following is basic structure of Bootstrap grid −

```
<div class="container">
    <div class="row">
        <div class="col" style="background-color: aquamarine; height: 100px">
        </div>
    </div>
</div>
```

We can see in the above example, inside a container there can be rows, inside which there can be multiple elements. Output of the above code will look something like this:

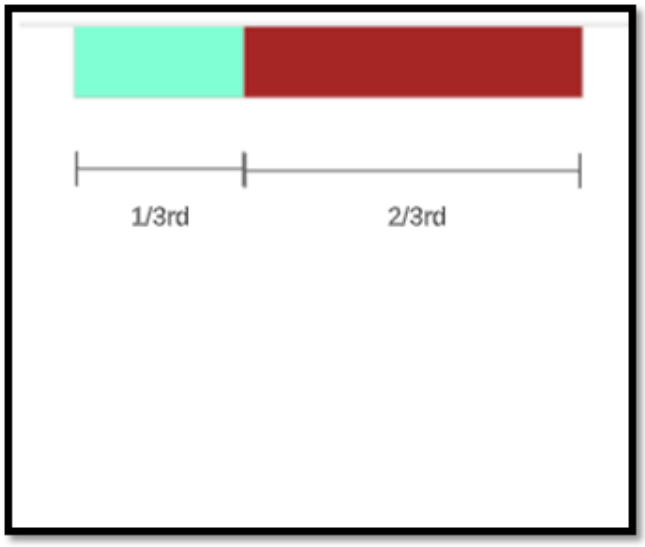


Bootstrap adds some padding by default. As we have not specified how wide this element will be that's why it will take the complete width available.

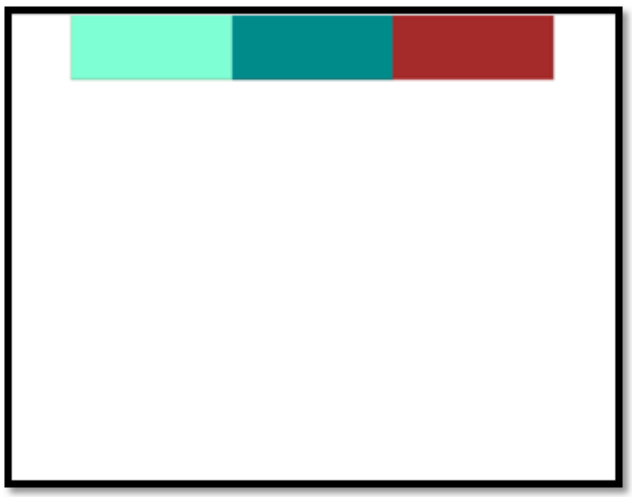
If you want the container to use the complete width of the browser, you can use container-fluid class, which will remove the padding.

If you want to specify the number of columns an element will occupy, you can write in this way:

```
<div class="container">
  <div class="row" style="background-color: brown;">
    <div class="col-4" style="background-color: aquamarine; height: 100px;">
    </div>
  </div>
</div>
```



As we can see, now it's occupying 4-cols or 1/3rd of the 12-cols. Let's add one more element which will occupy 4-cols:

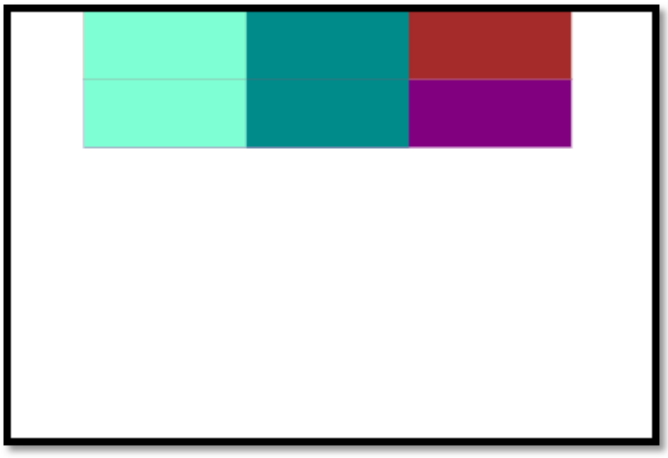


```
<div class="container">
    <div class="row" style="background-color: brown;">
```

```
        <div class="col-4" style="background-color: aquamarine; height:
100px;"></div>
        <div class="col-4" style="background-color: darkcyan; height:
100px;"></div>
    </div>
</div>
```

Now we can see that we have 2 elements occupying 1/3rd of the 12-col width each. Similarly, we can have multiple rows as well:

```
<div class="container">
    <div class="row" style="background-color: brown;">
        <div class="col-4" style="background-color: aquamarine; height: 100px;">
        </div>
        <div class="col-4" style="background-color: darkcyan; height: 100px;">
        </div>
    </div>

    <div class="row" style="background-color: purple;">
        <div class="col-4" style="background-color: aquamarine; height: 100px;">
        </div>
        <div class="col-4" style="background-color: darkcyan; height: 100px;">
        </div>
    </div>
</div>
```



By default, all your rules will be applicable to X-Small devices and to all other breakpoints beyond xs.

If you want to specify the number of columns an element occupies on different breakpoints, you can do that as well:

```
<div class="col-4 col-sm-6 col-lg-2" style="background-color: aquamarine; height: 100px;"> </div>
```

This means that the element will occupy 4-col width on x-small screens, and 6-col width on small screens and 2-col width on large screen and bigger screen devices.

Display Utilities

Now, we'll see different display utilities for layout in Bootstrap using which we can toggle the display values of components. Different values for display property are:

- none
- inline
- inline-block
- block
- table
- table-cell
- table-row
- flex
- inline-flex

You already know many of these properties, like none is used if you want to hide an element. Display utility classes that apply to all breakpoints, from xs to xl. If you want to set/change display properties, you'll need to add classes in this format:

.d-{value} for xs

.d-{breakpoint}-{value} for sm, md, lg, and xl.

where value is one of the properties mentioned above.

For example:

If you want to set the display of an element to block, you can do it this way:

```
<div class="d-block"></div>
```

If you want an element to be hidden on medium screens and visible on extra-small, small, large and extra-large screens:

```
<div class="d-block d-md-none d-lg-block"></div>
```

If you want an element to be hidden only on xs:

```
<div class="d-none d-sm-block"></div>
```

Spacing Utilities

You can use these to add responsive margin and padding utility classes to modify an element's appearance by assigning a responsive-friendly margin or padding values to an element.

Bootstrap also includes support for spacing properties like margin and padding. The classes are named using the format **{property}{sides}-{size}** for **xs** and **{property}{sides}-{breakpoint}-{size}** for **sm, md, lg, and xl** where:

property is one of:

- **m** – for classes that set **margin**
- **p** – for classes that set the **padding**

sides are one of:

- **t** - for classes that set **margin-top or padding-top**
- **b** - for classes that set **margin-bottom or padding-bottom**
- **l** - for classes that set **margin-left or padding-left**
- **r** - for classes that set **margin-right or padding-right**
- **x** - for classes that set both ***-left and *-right**
- **y** - for classes that set both ***-top and *-bottom**
- **blank** - for classes that set a margin or padding on **all 4 sides** of the element

and **size** is one of:

- 0 - for classes that eliminate the margin or padding by setting it to 0
- 1 - (by default) for classes that set the margin or padding to $spacer * .25
- 2 - (by default) for classes that set the margin or padding to $spacer * .5
- 3 - (by default) for classes that set the margin or padding to $spacer
- 4 - (by default) for classes that set the margin or padding to $spacer * 1.5
- 5 - (by default) for classes that set the margin or padding to $spacer * 3
- auto - for classes that set the margin to auto

where **$spacers** is a **Sass** variable, and its value can be customized.

For e.g.:

If you want to set the margin-top to 0:

```
<div class="mt-0"></div>
```

If you want to set the both left and right to 0:

```
<div class="px-0"></div>
```

For more information, you can have a read from the link below:

https://getbootstrap.com/docs/5.0/utilities/spacing/

Other Utilities

Border: You can use border utilities to add or remove an element's borders. The different border classes available are:

Additive: These are used to add borders.

- .border: This class adds a border all around the element.
- .border-top : This class adds a border on the top edge of the element.
- .border-left : This class adds a border on the left edge of the element.

- .border-right : This class adds a border on the right of the element.
- .border-bottom : This class adds a border on the bottom of the element.

Subtractive: These are used to remove borders.

- .border-0 : removes the border from all around the element.
- .border-top-0: removes the border from the top of the element
- .border-left-0: removes the border from the left of the element.
- .border-right-0: removes the border from the right of the element
- .border-bottom-0: removes the border from the bottom of the element

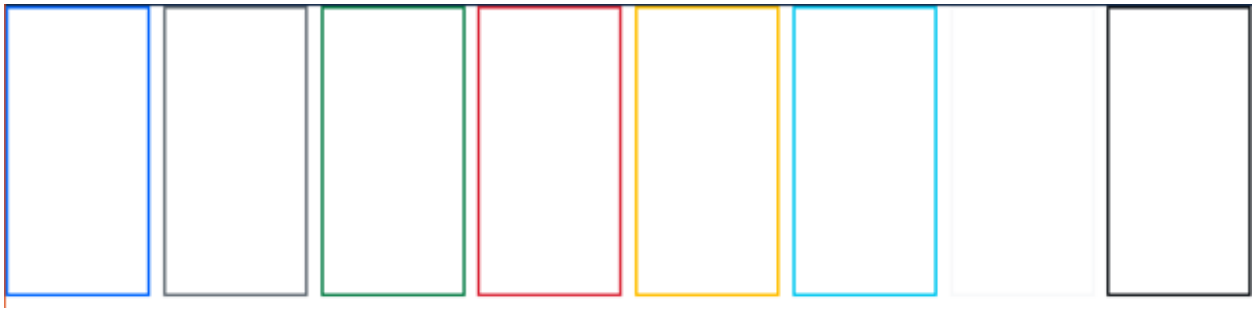
Border Color: Any color can be added to the border by using the following border-color classes:

- border border-primary
- border border-secondary
- border border-success
- border border-danger
- border border-warning
- border border-info
- border border-light
- border border-dark
- border border-white

Look at the code below, where we create spans with colored borders:

```
    <span class="border border-primary"></span>
    <span class="border border-secondary"></span>
    <span class="border border-success"></span>
    <span class="border border-danger"></span>
    <span class="border border-warning"></span>
    <span class="border border-info"></span>
    <span class="border border-light"></span>
    <span class="border border-dark"></span>
    <span class="border border-white"></span>
```

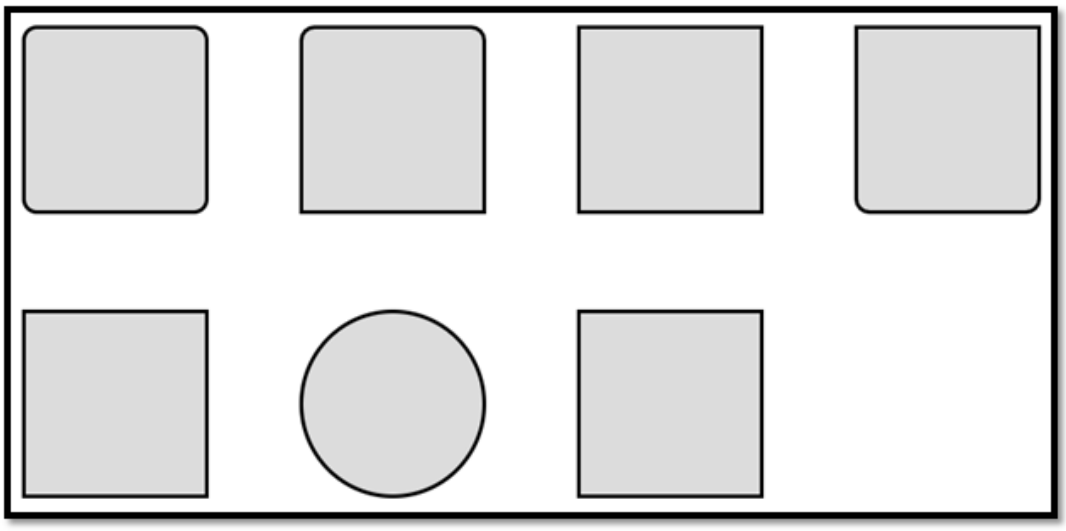
Output of the above code will look something like this:



Border Radius: A border-radius is used to make the corner of the border curved. The more is the radius, the more curved and round it will be. In bootstrap, the following classes as used in the code are used to implement radius at particular corners:

```
    <span class="rounded"></span>
    <span class="rounded-top"></span>
    <span class="rounded-right"></span>
    <span class="rounded-bottom"></span>
    <span class="rounded-left"></span>
    <span class="rounded-circle"></span>
    <span class="rounded-0"></span>
```

Output:



Colors: In Bootstrap, you can convey your meaning through colors, as it has a handful of contextual utility classes. The following classes can be used to color text:

```
<p class="text-primary">.text-primary</p>
    <p class="text-secondary">.text-secondary</p>
    <p class="text-success">.text-success</p>
```

```
    <p class="text-danger">.text-danger</p>
    <p class="text-warning">.text-warning</p>
    <p class="text-info">.text-info</p>
    <p class="text-light bg-dark">.text-light</p>
    <p class="text-dark">.text-dark</p>
    <p class="text-muted">.text-muted</p>
    <p class="text-white bg-dark">.text-white</p>
```

Output:

Background color: Similar to the text color classes, you can easily set the background color and text of elements using these classes:

```
<div class="bg-primary text-white">.bg-primary</div>
    <div class="bg-secondary text-white">.bg-secondary</div>
    <div class="bg-success text-white">.bg-success</div>
    <div class="bg-danger text-white">.bg-danger</div>
    <div class="bg-warning text-dark">.bg-warning</div>
```

```
    <div class="bg-info text-white">.bg-info</div>
    <div class="bg-light text-dark">.bg-light</div>
    <div class="bg-dark text-white">.bg-dark</div>
    <div class="bg-white text-dark">.bg-white</div>
```

Float and Clearfix: You can float an element to the right with the **.float-right** class or to the left with **.float-left,** and clear floats with the .**clearfix** class:

```
<div class="clearfix">
        <span class="float-left">Float left</span>
        <span class="float-right">Float right</span>
</div>
```

Width: You can set the width of an element with **w-*** classes:

```
    <div class="w-25 bg-primary">Width 25%</div>
    <div class="w-50 bg-warning">Width 50%</div>
    <div class="w-75 bg-danger">Width 75%</div>
    <div class="w-100 bg-secondary">Width 100%</div>
    <div class="mw-100 bg-dark">Max Width 100%</div>
```

Height: You can set height of an element using h-* classes:

```
<div>
    <div class="h-25 bg-secondary">Height 25%</div>
    <div class="h-50 bg-dark">Height 50%</div>
    <div class="h-75 bg-danger">Height 75%</div>
    <div class="h-100 bg-warning">Height 100%</div>
    <div class="mh-100 bg-primary">Height 100%</div>
</div>
```

Visibility: You can use classes **visible** and **invisible** to control the visibility of the elements.

```
<div class="visible">I'm here!</div>
<div class="invisible">I'm gone!</div>
```

If you want to know more about utilities, visit the link below:

https://getbootstrap.com/docs/5.0/utilities/api/

Bootstrap Components

Bootstrap has different components like Alerts, Dropdowns, Cards, Buttons, etc., which can be used in your website directly saving a lot of time.

Buttons: Bootstrap provides button classes to achieve button styles: .btn, .btn-default, .btn-primary, .btn-success, .btn-info, .btn-warning, .btn-danger, .btn-link.

Example:

```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
<button type="button" class="btn">Basic</button>
```

In Bootstrap, buttons can be of various sizes: **.btn-lg(large), .btn-sm(small), .btn-xs(extra-small):**

```
<button type="button" class="btn btn-primary btn-lg">Large</button>
<button type="button" class="btn btn-primary">Normal</button>
```

```
<button type="button" class="btn btn-primary btn-sm">Small</button>
<button type="button" class="btn btn-primary btn-xs">XSmall</button>
```

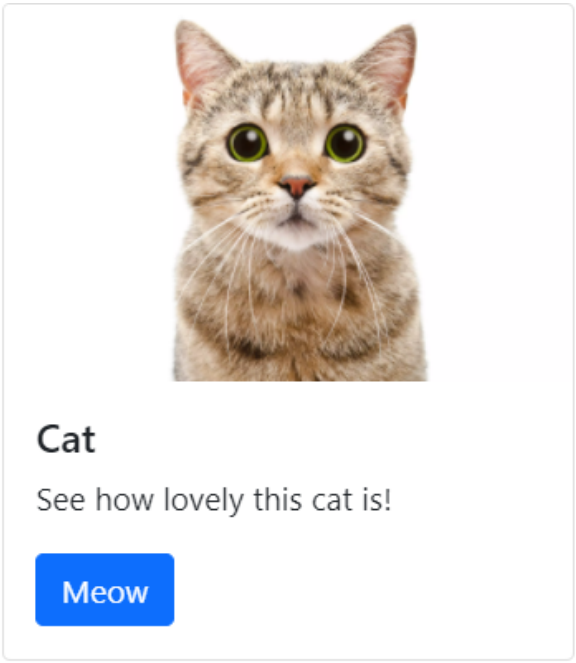
There are block level buttons that occupy the entire width of the parent element. You have to add the class **.btn-block** to create a block level button:

```
<button type="button" class="btn btn-secondary btn-block">Button 1</button>
```

A button can be set to be active or inactive. The classes **.active** and **.disabled** makes a button appear active or inactive respectively:

```
<button type="button" class="btn btn-default active">Active</button>
<button type="button" class="btn disabled">Disabled</button>
```

Cards: Bootstrap 4 has cards that are bordered boxes with some padding around their content. It includes options for headers, footers, content, colors, etc.



A card is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options.

A basic card is created with the **.card** class, and the content inside the card has a **.card-body** class:

```
<div class="card">
    <div class="card-body">Basic card</div>
</div>
```

Card Header: The **.card-header** class adds a heading to the card and the **.card-footer** class adds a footer to the card:

```
<div class="card">
    <div class="card-header">Header</div>
    <div class="card-body">Content</div>
    <div class="card-footer">Footer</div>
</div>
```

You can also add background colors to cards by adding contextual classes **(.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light).**

Card images: The **.card-img-top** places an image to the top of the card. With **.card-text**, text can be added to the card. Text within .card-text can also be styled with the standard HTML tags.

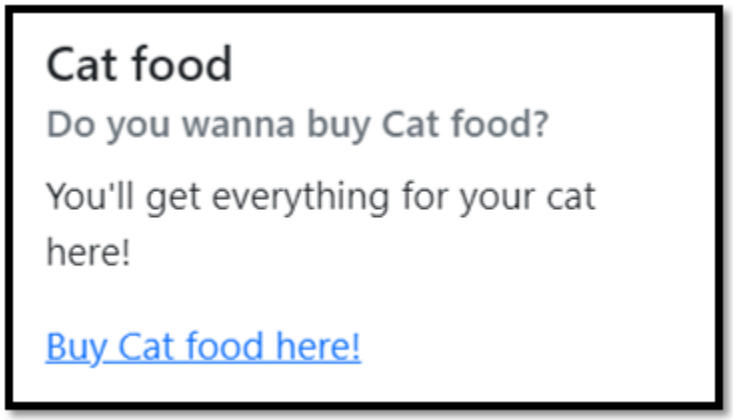
```
<div class="card">
    <img src="cat.png" class="card-img-top" alt="cat_image">
    <div class="card-body">
    <p class="card-text">See, how lovely this cat is!</p>
    </div>
</div>
```

Card Title, Text, and Links: Card titles can be used by adding a **.card-title** class to a <h*> tag.

In the same way, links are added and placed next to each other by adding **.card-link** to a <a> tag.

```
<div class="card" style="width: 300px;">
   <div class="card-body">
     <h5 class="card-title">Cat food</h5>
     <h6 class="card-subtitle mb-2 text-muted">Do you wanna buy Cat food?</h6>
     <p class="card-text">You'll get everything for your cat here!</p>
     <a href="#" class="card-link">Buy Cat food here!</a>
   </div>
</div>
```

Output:



Dropdowns: It's a toggleable menu that allows the user to choose one the predefined options. They're made interactive with the included Bootstrap dropdown JavaScript plugin.

```
<div class="dropdown">
<button   class="btn   btn-secondary   dropdown-toggle"   type="button"
data-toggle="dropdown">My dropdown<span class="caret"></span>
</button>
  <ul class="dropdown-menu">
        <li><a href="#">1</a></li>
        <li><a href="#">2</a></li>
        <li><a href="#">3</a></li>
  </ul></div>
```

The **.dropdown-header** class is used to add headers inside the dropdown menu:

```
<li class="dropdown-header">Dropdown header 1</li>
```

You can read more about components from the link below:

https://getbootstrap.com/docs/5.0/components/accordion/

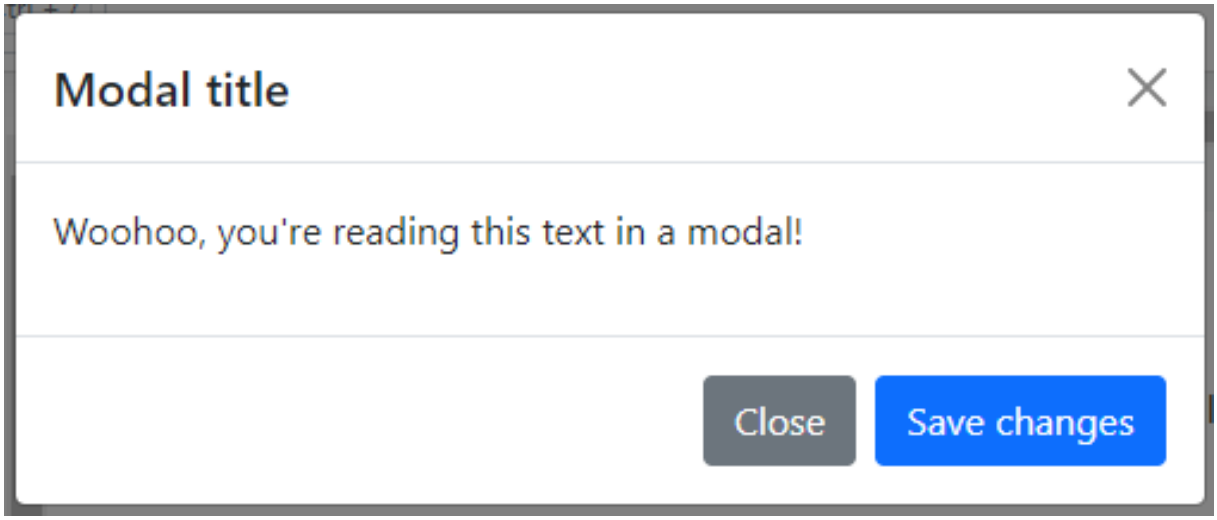
Modals

Modals are a dialog box/popup window that is displayed on top of the current page. They are built with HTML, CSS, and JavaScript. Bootstrap only supports one modal window at a time as nested modals are considered to produce a bad user experience.

Modal Components: These are the modal header, modal body (required for padding), and modal footer (optional).

A modal example:

```
 <!-- Button triggers the modal -->
    <button type="button" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#exampleModal">
    Launch demo modal
  </button>

    <!-- Modal -->
    <div class="modal fade" id="exampleModal">
        <div class="modal-dialog">
      <!-- Modal content-->
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLabel">Modal
title</h5>
                    <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    ...
                </div>
                <div class="modal-footer">
                            <button   type="button"   class="btn   btn-secondary"
data-bs-dismiss="modal">Close</button>
         <button type="button" class="btn btn-primary">Save changes</button>
                </div>
            </div>
       </div></div>
```



Modal explained:

The parent of the modal must have an **ID** that should be the **same** as the value of the data-target attribute used to trigger the modal ("exampleModal").

The **.modal** class identifies the content of as a modal and brings focus to it. The **.fade** class adds a **transition** effect that fades the modal in and out.

You can remove this class if you do not want this effect. The attribute role=" dialog" improves accessibility for people using **screen readers**. The **.modal-dialog** class sets the proper width and margin of the modal.

Modal content explained: The having class="modal-content" will style the modal (border, background-color, etc.). Add the modal's header, body, and footer inside this .

The .modal-header class is used to define the style for the header of the modal. The <button> inside the header has a data-dismiss="modal" attribute which closes the modal if you click on it. The **.close** class styles the close button, and the **.modal-title** class styles the header with a proper line-height.

The **.modal-body** class will be used to define the style for the body of the modal. You can add any HTML here; paragraphs, images, videos, etc. The **.modal-footer** class is used to define the style for the footer of the modal. You should remember that this area is right-aligned by default.

Modal Size: You can change the size of the modal by adding the **.modal-sm** class for small modals or **.modal-lg** class for large modals. You need to add this size class to the element with class **.modal-dialog.**

```
<div class="modal-dialog modal-lg">
```

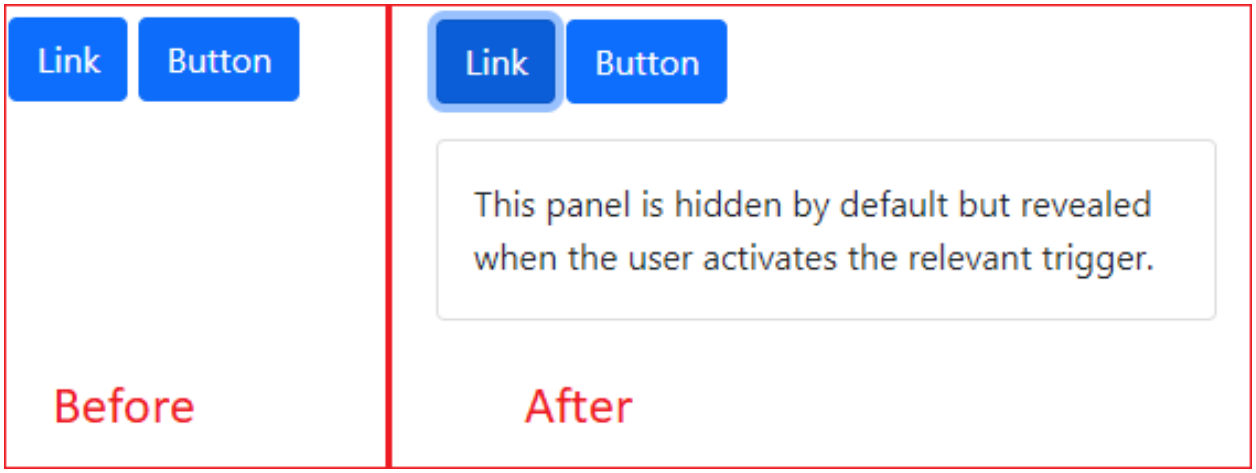
You can read more about modals from here:
https://getbootstrap.com/docs/5.0/components/modal/

Collapse

The collapse JavaScript plugin is used to show and hide content. Buttons or anchor tags are used as triggers that are bound to specific elements you toggle. Collapsing an element will animate the height from its current value to 0. Collapses are useful when you want to hide and show large amount of content:

```
  <p>
                    <a    class="btn    btn-primary"    data-bs-toggle="collapse"
href="#collapseExample"            role="button"            aria-expanded="false"
aria-controls="collapseExample">Link </a>
         <button class="btn btn-primary" type="button" data-bs-toggle="collapse"
data-bs-target="#collapseExample"                          aria-expanded="false"
aria-controls="collapseExample">
          Button
        </button>
    </p>
```

```
<div class="collapse" id="collapseExample">
 <div class="card card-body">
 This panel is hidden by default but revealed when the user activates the relevant trigger.
 </div>
</div>
```



The **.collapse** class indicates an element that can be **collapsed**. The content inside this element will be shown or hidden as per the click of a button. We can control (show/hide) the collapsible content by adding the **data-toggle="collapse"** attribute to an <a> or a <button> element. Then we can add the data-target="#id" attribute to connect the button with the collapsible content .

For <a> elements, you can use the href attribute instead of the data-target attribute:

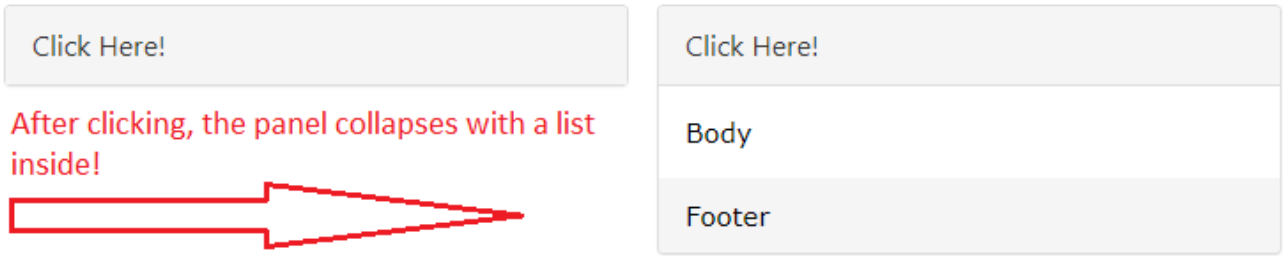
```
<a href="#collapseExample" data-toggle="collapse">Collapsible</a>

    <div id="collapseExample" class="collapse">
        Hello there!
    </div>
```

By default, the collapsible content is hidden. However, you can add the .**show** class to show the content by default:

```
<div id="collapseExample" class="collapse show">
    Random Text
</div>
```

There are collapsible panels also, which you can use:

```
<div class="panel-group">
    <div class="panel panel-default">
       <div class="panel-heading">
           <h4 class="panel-title">
               <a data-toggle="collapse" href="#collapse1">Click Here!</a>
            </h4>
        </div>
        <div id="collapse1" class="panel-collapse collapse">
             <div class="panel-body">Body</div>
             <div class="panel-footer">Footer</div>
        </div>
    </div>
</div>
```



Navs

If you want to create navigation menus, Bootstrap includes classes that can help you. Simple horizontal menu can be created by adding the class **.nav** to a <ul>, and **.nav-item** class to each <li>. If there are links inside <li>, add class **.nav-link** to them:


```
<ul class="nav">
        <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
```

```
        </li>
        <li class="nav-item">
            <a class="nav-link disabled" href="#">Disabled</a>
        </li>
    </ul>
```

If you want to align the nav to the center, add the class **.justify-content-center,** and if you want to align it to the right-end, add the class **.justify-content-end.**

```
    <!-- Centered nav -->
    <ul class="nav justify-content-center">

    <!-- Right-aligned nav -->
    <ul class="nav justify-content-end">
```

You can create a vertical nav bar by adding the class **.flex-column**:

```
<ul class="nav flex-column">
  <li class="nav-item">
       <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
       <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

You can also turn this navigation bar into navigation tabs by adding the .nav-tabs class. You'll need to add the class **.active** to the link you want to be active.


```
<ul class="nav nav-tabs">
   <li class="nav-item">
        <a class="nav-link active" href="#">ActiveLink 1</a>
   </li>
   <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled Link 1</a>
   </li>
</ul>
```

You can also turn your nav menu into **nav pills** by adding the .**nav-pills** class:

```
<ul class="nav nav-pills">
     <li class="nav-item">
         <a class="nav-link active" href="#">Link 1</a>
     </li>
     <li class="nav-item">
         <a class="nav-link disabled" href="#">Link 2</a>
     </li>
</ul>
```

You can read from the below link for more information:

https://getbootstrap.com/docs/4.0/components/navs/

Navbar

You can also use pre-built navbars which can extend or collapse, depending on the screen size. A basic horizontal navbar can be created using the **.navbar** class along with a responsive collapsing class: **.navbar-expand-xl|lg|md|sm** (stacks the navbar vertically on extra-large, large, medium, or small screens).

To add links inside the navbar, use a <ul> element with **class="navbar-nav"**. Then add <li> elements with a **.nav-item** class followed by an <a> element with a **.nav-link class**:


```
<nav class="navbar navbar-expand-sm bg-dark" style="margin-top: 10px;">
        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link" href="#">Link 1</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link 2</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link 3</a>
            </li>
        </ul>
</nav>
```

You can remove the **.navbar-expand-xl|lg|md|sm** class to create a vertical navbar. You can add the justify-content-center class to center the navbar. To create colored navbars, you can use any of the .bg-color classes which will change the background color of the navbar (**.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light**).

The **.navbar-brand** class can be used to highlight the brand/logo/project name of your webpage:

```
<nav class="navbar navbar-expand-sm bg-light" style="margin-top: 10px;">
        <a class="navbar-brand" href="#">Brand Name</a>
        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link" href="#">Link 1</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link 2</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link 3</a>
            </li>
        </ul>
    </nav>
```

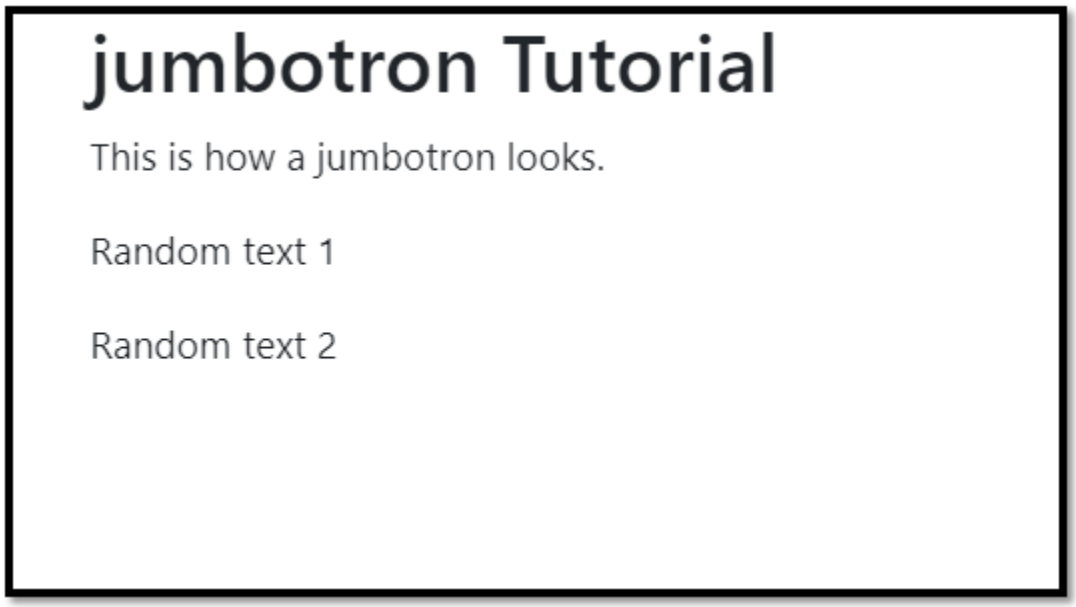
You can read more about navbars from this link: https://getbootstrap.com/docs/4.0/components/navbar/

Jumbotron

A jumbotron is like a big container for giving special attention to some content or information on the page. It's a grey box with rounded corners. The font size of the text inside the jumbotron is enlarged, and you can put all other bootstrap components discussed earlier, inside this jumbotron.

You can create a jumbotron with class **.jumbotron** using a :

```
<div class="container">
        <div class="jumbotron">
          <h1>jumbotron Tutorial</h1>
          <p>This is how a jumbotron looks.</p>
        </div>
        <p>Random text 1</p>
        <p>Random text 2</p>
      </div>
```

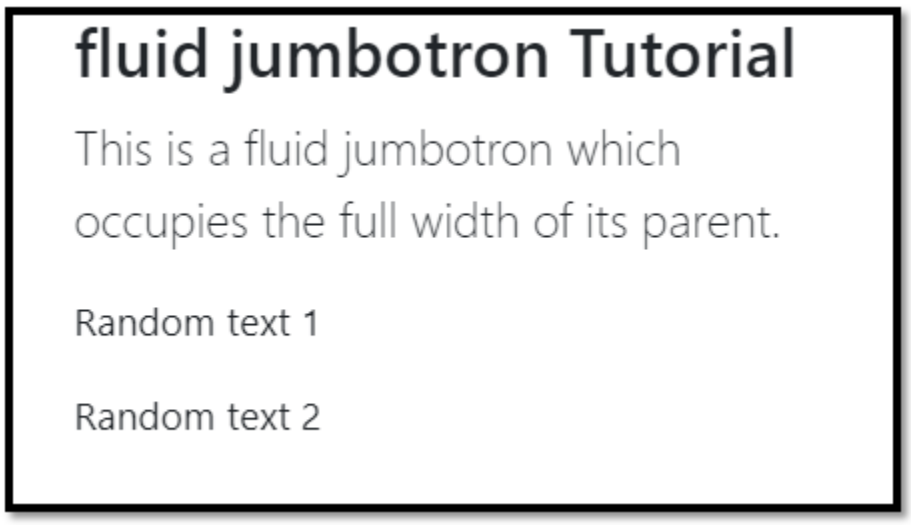


You can place the jumbotron inside a with .container class to prevent the jumbotron to extend the screen edges.

To make the jumbotron full width, and without rounded corners, you can add the **.jumbotron-fluid** modifier class and add a **.container** or **.container-fluid** within.

```
<div class="container-fluid">
        <div class="jumbotron jumbotron-fluid">
          <h1>fluid jumbotron Tutorial</h1>
                  <p class="lead">This is a fluid jumbotron which occupies the
full width of its parent.</p>
```

```
    </div>
    <p>Random text 1</p>
    <p>Random text 2</p>
  </div>
```



For a better understanding, you can look at the Jumbotron template, which uses some part of everything we've learned till now:

https://getbootstrap.com/docs/4.0/examples/jumbotron/#