

Website Design (Manual) BOOTSTRAP

Developed by Alabian Solutions Ltd

Module 1

Chapter One

What is bootstrap?

Bootstrap is a front-end framework that helps developers to jump start the web development process. Developers who are moving to front-end development from hardcore server-side programming languages such as Java or PHP can find it very difficult to come to grips with CSS and JavaScript; however, with Bootstrap they only have to concentrate on writing proper HTML, leaving the tricky CSS and JavaScript to Bootstrap.

Why does it exist?

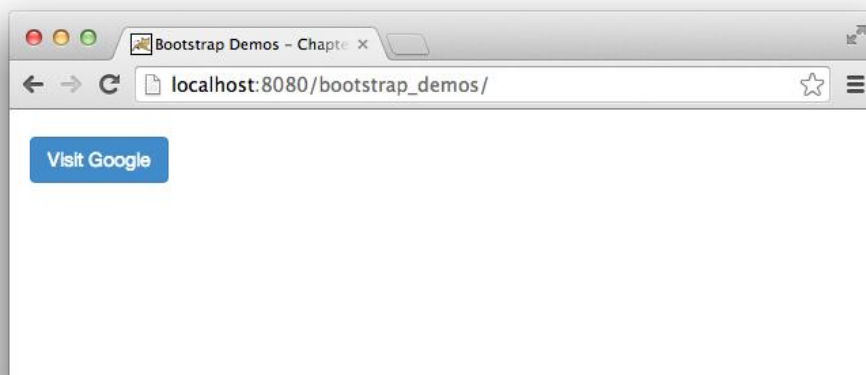
Imagine you have to design a website with an attractive navigation bar, stylish buttons, nice typography, placeholders for texts and images, a big image slider, and more—yet you aren't a front end development expert. But what if these features were already coded for you, and you just had to write a little HTML to use them? This is Bootstrap.

For example, using the class **btn** with link (`<a>`) elements will make them appear like a button as seen in Figure 1.1. Additionally, using the **btn-primary** class with a link will make it a dark blue button:

Code:

```
<a href="http://www.google.com" class="btn btn-primary">
Visit Google</a>
```

Result:



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:** Bootstrap 4 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)

How Can It Help Me?

Bootstrap is a plus for a novice designer, but it's not restricted to novices. Experts can also use Bootstrap's code as a base to create something new. Bootstrap allows you to customize its styles through the use of Less¹ and Sass². Developers acquainted with these technologies can completely modify Bootstrap's default look and feel.

Need for prototyping

The main reason for having a good CSS framework is to ease the development process.

There are many common tasks that every web designer carries out while developing a website. Tasks such as clearing browser resets, creating a proper grid system for website layout, and assigning typography rules can become frustrating and time consuming if done repeatedly for every project. A good CSS framework provides a powerful set of tools that streamline these tasks.

Some of the main highlights of a good CSS framework include:

- Faster development
- Organized and maintainable code
- Allowing you to spend time on innovation rather than reinventing the wheel.

The origins of bootstrap

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

Bootstrap's Competition

There are many other popular frameworks that are competing with Bootstrap in the front-end framework arena. Some of them are:

- Foundation framework by Zurb

- Semantic UI
- Gumbo framework
- Pure by Yahoo

Getting bootstrap ready

Downloading Bootstrap 4

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there.

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

Code: CDN

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/
css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>

<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popp
er.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min
.js"></script>
```

A typical bootstrap page with CDN link

Code:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.cs
s" integrity="sha384-
GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMP0DgiMDm4iYMj70gZWKYbI706tWS"
crossorigin="anonymous">

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

```

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.j
s" integrity="sha384-
wHAIffRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYU1hA6ZOblgut"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I413gm9GU6Hclog6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
  </body>
</html>

```

A typical Bootstrap page when you host it yourself

```

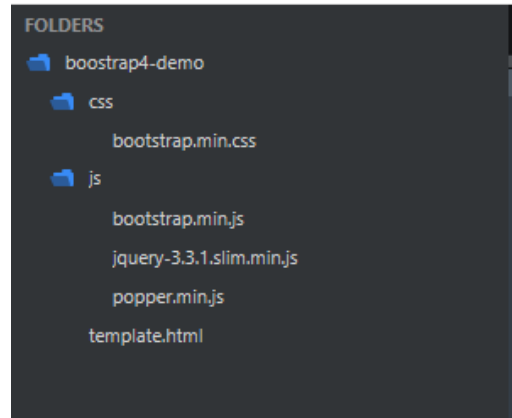
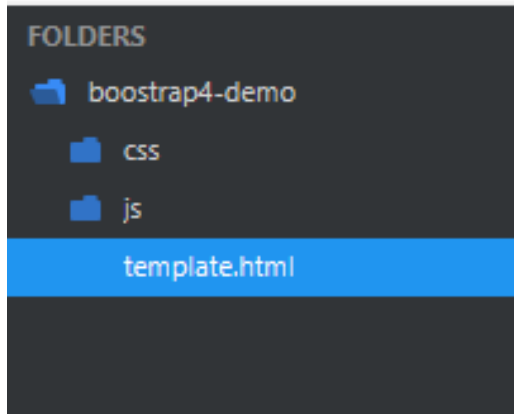
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="js/jquery-3.3.1.slim.min.js"></script>
    <script src="js/popper.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>

```

Our project directory will look like the figure below



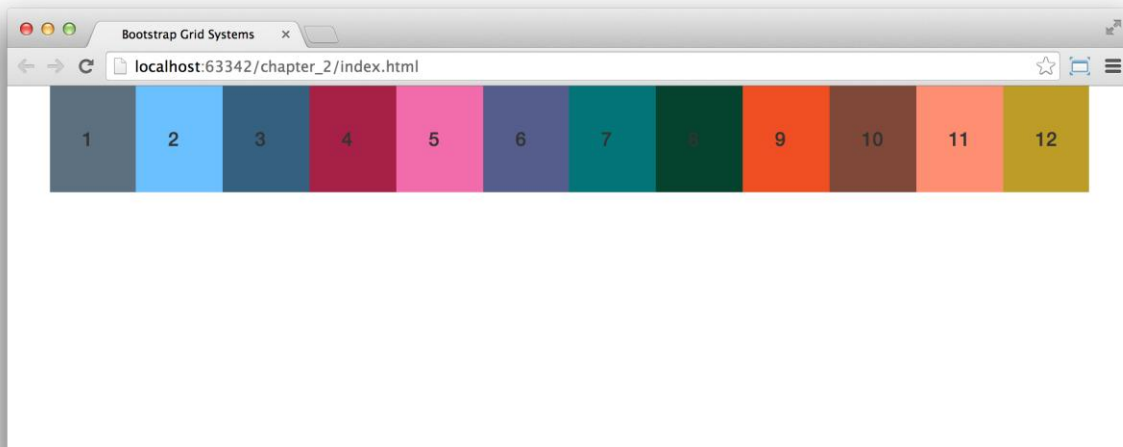
Chapter Two

Bootstrap Grid System

What is a grid system?

A **grid system** allows us to properly house our website's content. It divides the screen into multiple rows and columns that can be used to create various types of layouts.

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



Types of container

There are two types of container classes, they are:

1. Container
2. Container- fluid

Container: The container class is used if we have a responsive fixed-width layout where in max-width changes with the breaking point and it takes 80% of its parent width.

Container-fluid: while a container-fluid occupies 100% of the viewport's width.

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.

Basic Structure of a Bootstrap Grid

The following is a basic structure of a Bootstrap grid:

Example 1

Code:

```
<div class="row">
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  ...
</div>
```

Example 2

`<!-- Or let Bootstrap automatically handle the layout -->`

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

First; create a row (`<div class="row">`). Then, add the desired number of columns (tags with appropriate `.col-*-*` classes). Note that numbers in `.col-*-*` should always add up to 12 for each row.

Below we have collected some examples of basic Bootstrap grid layouts.

Second example: instead of adding a number to each col, let bootstrap handle the layout, to create equal width columns: two "col" elements = 50% width to each col. three cols = 33.33% width to each col. four cols = 25% width, etc. You can also use .col-sm|md|lg|xl to make the columns responsive. This is one of the beauty of bootstrap 4.

Three Equal Columns

Code:

```
<div class="container">
  <div class="row">
    <div class="col-4 bg-danger">
      col-4
    </div>
    <div class="col-4 bg-success">
      col-md-4
    </div>
    <div class="col-4 bg-info">
      col-4
    </div>
  </div>
</div>
```

Result:



Two Unequal Columns

Code:

```
<div class="container">
  <div class="row">
    <div class="col-8 bg-danger">
      col-8
    </div>
    <div class="col-4 bg-success">
      col-4
    </div>
  </div>
</div>
```

Result:



Creating a dynamic layout

Let's see how to put the grid system into practice, creating a dynamic layout that Adjusts to the size of device it is viewed in.

Designing for desktop

The class `.col-md-*` is used for medium size display devices which is usually larger than 992px. Below is the code snippet

Code:

```
<div class="container">
  <div class="row">
    <div class="col-md-4">
      <h3>Post Title 1</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4">
      <h3>Post Title 2</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4">
      <h3>Post Title 3</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4">
      <h3>Post Title 4</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4">
      <h3>Post Title 5</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4">
      <h3>Post Title 6</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
  </div>
</div>
```

Result

Post Title 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Post Title 2 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Post Title 3 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Post Title 4 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Post Title 5 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Post Title 6 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Designing for tablet

Let's now modify our code to achieve the wireframe layout for tablets. Unlike desktop

displays, tablets can be viewed in two formats: **Portrait** and **Landscape**. A tablet's landscape view is considered a medium-sized display (**screen width $\geq 992\text{px}$**), which

we've already taken care of using `col-md-*` classes. We're now left with the portrait view, which is a small-sized display. This can be achieved using `col-sm-*` classes. Since we have to achieve a two-column layout in smaller displays, we have to force each column to span across six Bootstrap columns.

Let's proceed and add an extra class `col-sm-6` to our desktop layout code:

Code:

```
<div class="container">
  <div class="row">
    <div class="col-md-4 col-sm-6 bg-danger">
      <h3>Post Title 1</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4 col-sm-6 bg-success">
      <h3>Post Title 2</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4 col-sm-6 bg-warning">
      <h3>Post Title 3</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4 col-sm-6 bg-info">
      <h3>Post Title 4</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4 col-sm-6 bg-success">
      <h3>Post Title 5</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
    <div class="col-md-4 col-sm-6 bg-warning">
      <h3>Post Title 6</h3>
      <p>Lorem ipsum dolor sit amet ... </p>
    </div>
  </div>
</div>
```

Result

Post Title 1 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!	Post Title 2 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!
Post Title 3 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!	Post Title 4 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!
Post Title 5 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!	Post Title 6 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Rem magni nesciunt, impedit? Facere nesciunt, tempora voluptates quaerat tenetur consequatur quasi sint sapiente itaque deleniti placeat debitis, labore non unde iure!

Designing for Mobile

Like tablets, mobiles can also be viewed in landscape and portrait mode. The landscape view in mobile devices utilize small-sized displays (screen width \geq 768px) that we have already taken care of using `col-sm-*` classes. Portrait view in mobile devices employs extra small-sized displays (**screen width < 768px**).

As per the wireframe for a mobile device, we have to create a single-column layout.

For extra small screens, we have to use classes that have the `col-*` classes.

Here, we want each column to occupy all the 12 Bootstrap columns. Our class will be `col-12`.

Code:

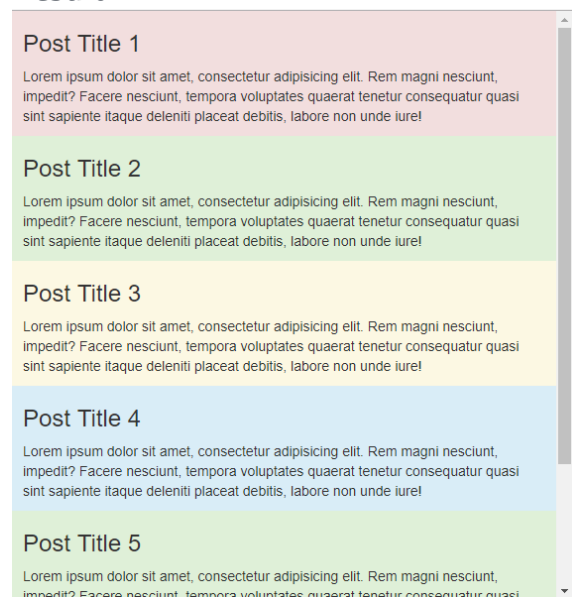
```
<div class="container">
  <div class="row">
    <div class="col-md-4 col-sm-6 col-12 bg-danger">
      <h3>Post Title 1</h3>
      <p>
        Lorem ipsum dolor sit amet.....
      </p>
    </div>
    <div class="col-md-4 col-sm-6 col-12 bg-success">
      <h3>Post Title 2</h3>
      <p>
        Lorem ipsum dolor sit amet...
      </p>
    </div>
    <div class="col-md-4 col-sm-6 col-12 bg-warning">
      <h3>Post Title 3</h3>
      <p>
        Lorem ipsum dolor sit amet...
      </p>
    </div>
    <div class="col-md-4 col-sm-6 col-12 bg-info">
```

```

        <h3>Post Title 4</h3>
        <p>
            Lorem ipsum dolor sit amet...
        </p>
    </div>
    <div class="col-md-4 col-sm-6 col-12 bg-success">
        <h3>Post Title 5</h3>
        <p>
            Lorem ipsum dolor sit amet...
        </p>
    </div>
    <div class="col-md-4 col-sm-6 col-12 bg-warning">
        <h3>Post Title 6</h3>
        <p>
            Lorem ipsum dolor sit amet...
        </p>
    </div>
</div>
</div>

```

Result:



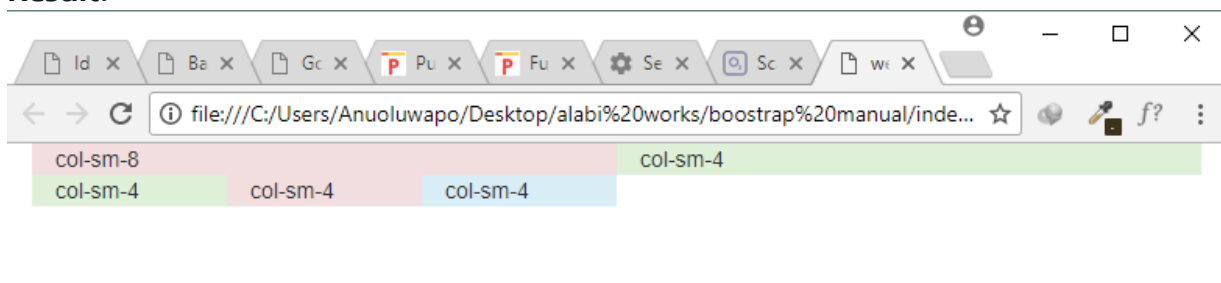
Nesting columns

You can always create a new set of 12 Bootstrap columns within any column in your layout. This can be done by building a new row element within an existing column and then filling this row with custom columns. Since we are starting a new row here, any column within it can also span across 12 Bootstrap columns, but the width of this row will be restricted to its parent's width.

Code:

```
<div class="container">
  <div class="row">
    <div class="col-sm-8 bg-danger">
      col-sm-8
      <div class="row">
        <div class="col-md-4 col-sm-4 col-xm-4 bg-
          success">col-sm-4</div>
        <div class="col-md-4 col-sm-4 col-xm-4 bg-
          danger">col-sm-4</div>
        <div class="col-md-4 col-sm-4 col-xm-4 bg-info">col-
          sm-4</div>
      </div>
    </div>
    <div class="col-sm-4 bg-success">
      col-sm-4
    </div>
  </div>
</div>
```

Result:



The ability to nest columns comes in handy when creating complex layouts. You can also further nest the innermost row and generate a new set of columns within it. This process can continue until you achieve the desired layout.

Offsetting columns

Offsetting is another great feature of Bootstrap's grid system. It is generally used to increase the left margin of a column. For example, if you have a column that should appear after a gap of three Bootstrap columns, you can use the offsetting feature. Classes that are available for offsetting are:

- .offset-sm-*
- .offset-md-*
- .offset-lg-*

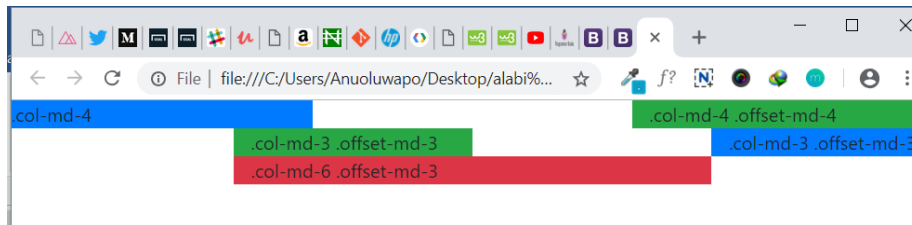
For example:

For example, .offset-md-4 moves .col-md-4 over four columns.

code:

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
</div>
<div class="row">
  <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
</div>
<div class="row">
  <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
</div>
```

Result:



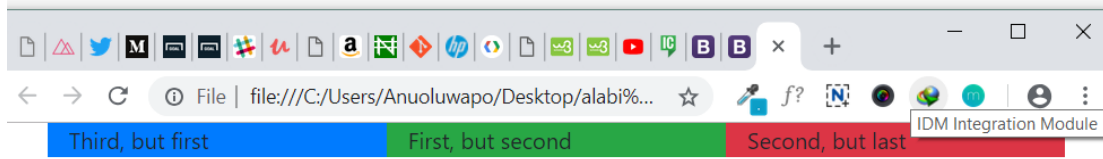
Reordering Grids Manually

Use **.order-** classes for controlling the **visual order** of your content. These classes are responsive, so you can set the **order** by breakpoint (e.g., **.order-1**, **.order-md-2**). Includes support for **1** through **12** across all five grid tiers.

code:

```
<div class="container">
  <div class="row">
    <div class="col bg-success order-2">
      First, but second
    </div>
    <div class="col order-3 bg-danger">
      Second, but last
    </div>
    <div class="col order-1 bg-primary">
      Third, but first
    </div>
  </div>
</div>
```

Result:



In the code, you will notice the first column(div) was reordered to the second column and the second column(div) was reordered to the last column, likewise the the third column(div) was reordered to the first column.

Chapter 3

Buttons

Button Styles

Bootstrap provides seven styles of buttons:

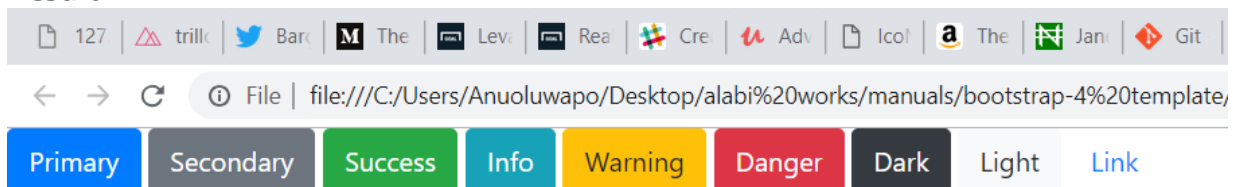
- `.btn-default`
- `.btn-primary`
- `.btn-success`
- `.btn-info`
- `.btn-warning`
- `.btn-danger`
- `.btn-link`
- `.btn-dark`
- `.btn-light`

The following example shows the code for the different button styles:

Code:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-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-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

Result:



Button Outline

Bootstrap 4 provides the following outline/bordered buttons:

- `.btn-outline-primary`
- `.btn-outline-success`
- `.btn-outline-info`
- `.btn-outline-warning`
- `.btn-outline-danger`

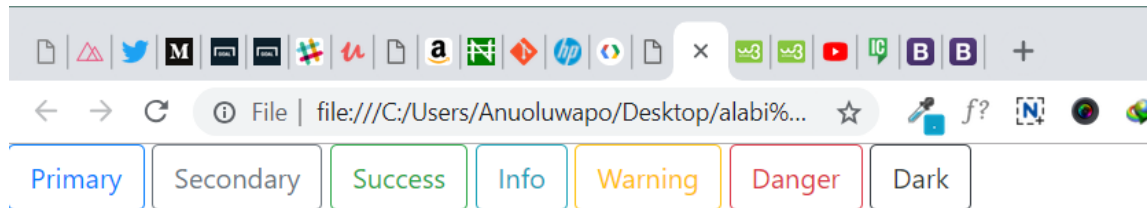
- .btn-outline-dark

The following example shows the code for different button outline:

Code:

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

Result:



Button Sizes

Bootstrap provides four button sizes:

The classes that define the different sizes are:

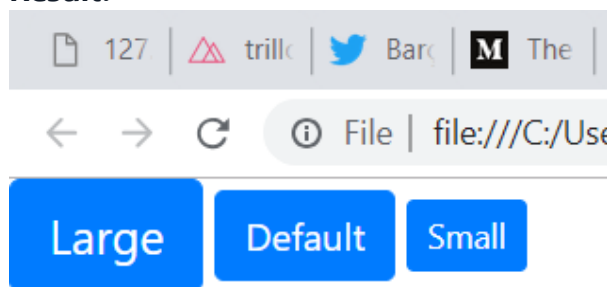
- .btn-lg
- .btn-xs

The following example shows the code for different button sizes:

Code:

```
<button type="button" class="btn btn-primary btn-lg">Large</button>
<button type="button" class="btn btn-primary">Default</button>
<button type="button" class="btn btn-primary btn-sm">Small</button>
```

Result:



Block Level Buttons

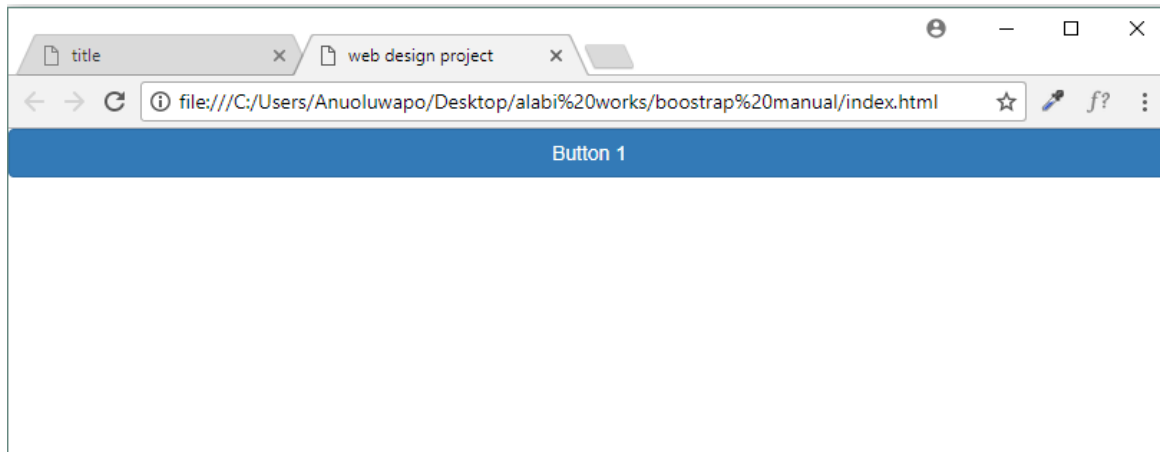
A block level button spans the entire width of the parent element. Add class **.btn-block** to create a block level button:

Example:

Code:

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

Result:



Chapter 4

Utilities

Colors

Contextual Colors

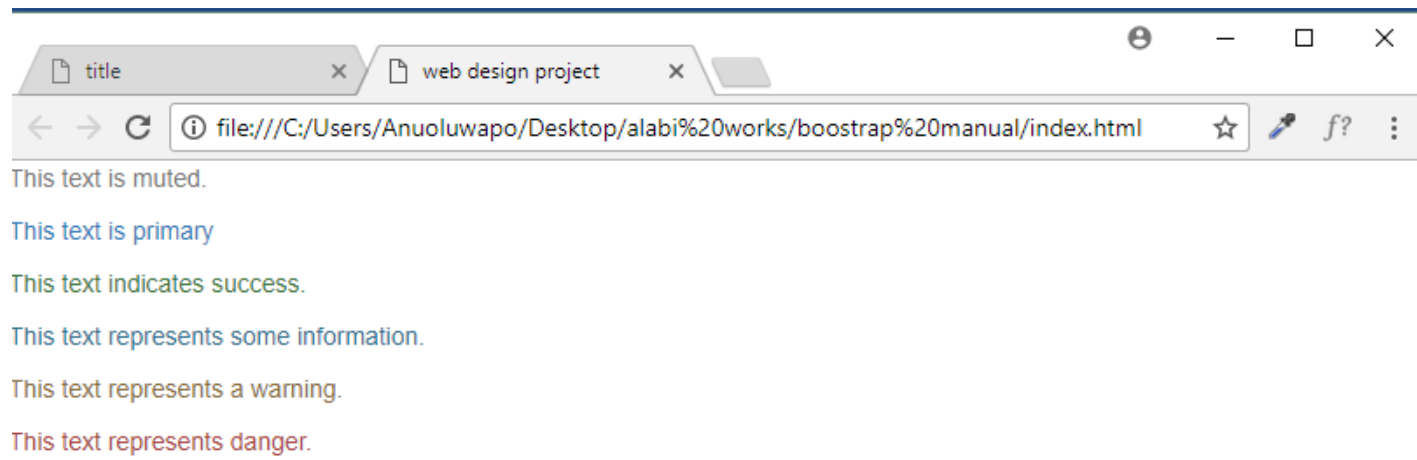
Bootstrap also has some contextual classes that can be used to provide "meaning through colors".

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

Code:

```
<p class="text-muted">This text is muted.</p>
<p class="text-primary">This text is primary</p>
<p class="text-success">This text indicates success.</p>
<p class="text-info">This text represents some information.</p>
<p class="text-warning">This text represents a warning.</p>
<p class="text-danger">This text represents danger.</p>
```

Result:



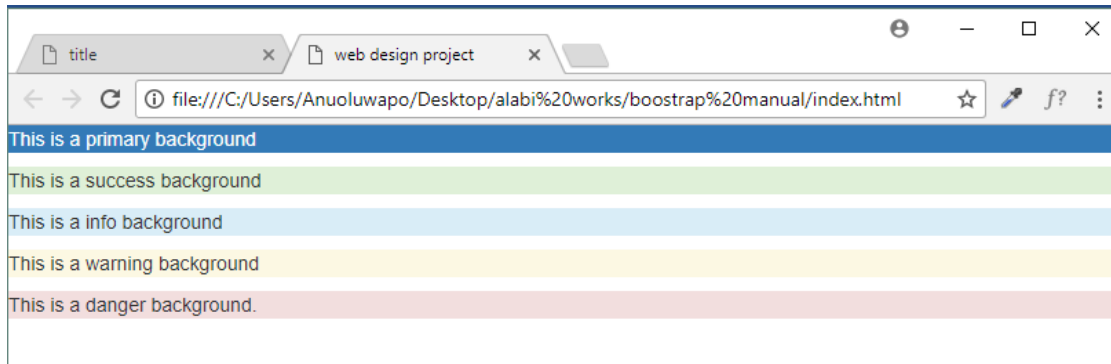
Classes for background color

The classes for background colors are: **.bg-primary**, **.bg-success**, **.bg-info**, **.bg-warning**, **.bg-danger**, **.bg-dark** and **.bg-light**.

Code:

```
<p class="bg-primary">This is a primary background</p>
<p class="bg-success">This is a success background</p>
<p class="bg-info">This is a info background</p>
<p class="bg-warning">This is a warning background</p>
<p class="bg-danger">This is a danger background</p>
```

Result:



Embed

Create responsive video or slideshow embeds based on the width of the parent. Add the **.embed-responsive-item** to any embed elements (like `<iframe>` or `<video>`) in a parent element with **.embed-responsive** and an aspect ratio of your choice:

Example:

Code:

```
<!-- 21:9 aspect ratio -->
<div class="embed-responsive embed-responsive-21by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 16:9 aspect ratio -->
<div class="embed-responsive embed-responsive-16by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 4:3 aspect ratio -->
<div class="embed-responsive embed-responsive-4by3">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 1:1 aspect ratio -->
<div class="embed-responsive embed-responsive-1by1">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

Images

Bootstrap Image Shapes

- i. Rounded Corners: The **.rounded** class adds rounded corners to an image.
- ii. Circle: The **.rounded-circle** class shapes the image to a circle (IE8 does not support rounded corners):
- iii. Thumbnail: The **.img-thumbnail** class shapes the image to a thumbnail:

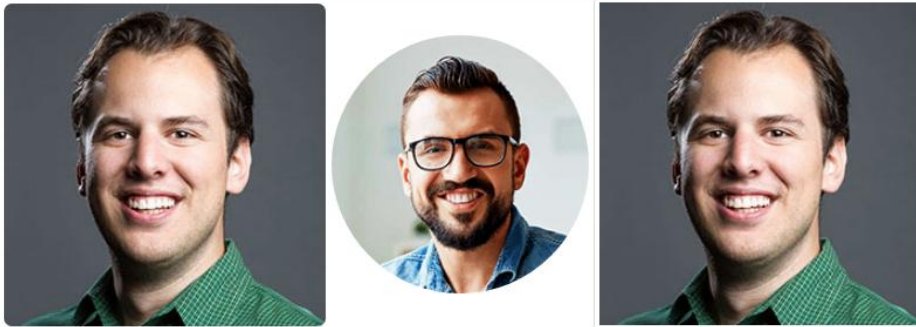
The following example show different image shapes in bootstrap

Code:

```
  
  

```

Result:



Responsive Images

Images comes in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.

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

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

Aligning Images

Float an image to the right with the **.float-right** class or to the left with **.float-left**:

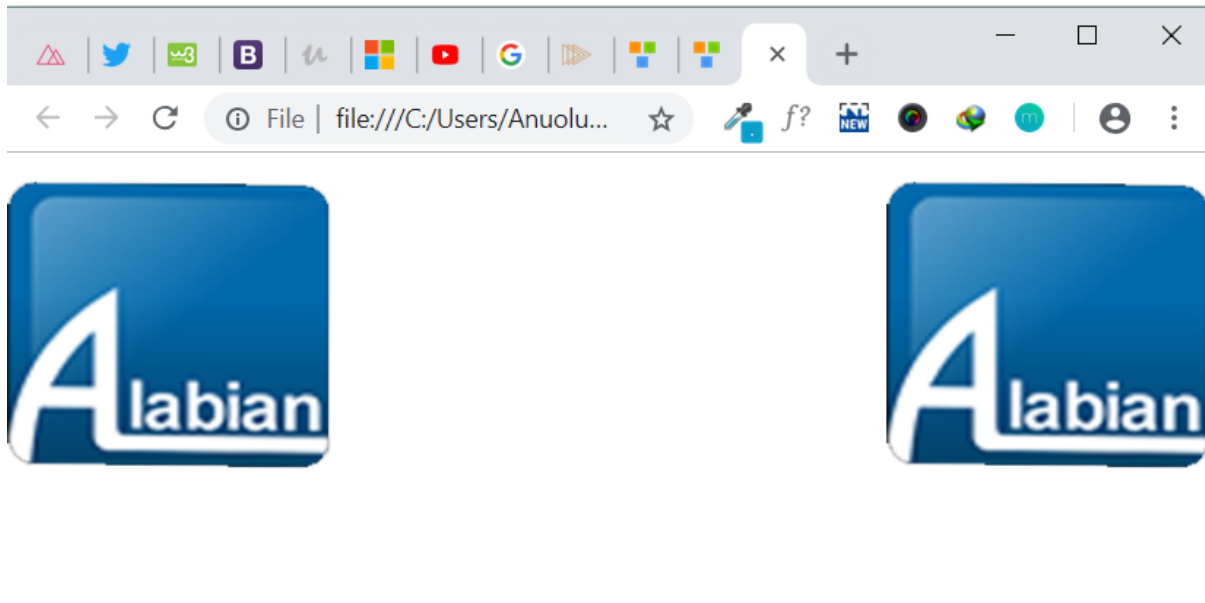
Example:

Code:

```
  

```

Result:



Text alignment classes

- `.text-left` Indicates left-aligned text
- `.text-center` Indicates center-aligned text
- `.text-right` -Indicates right-aligned text
- `.text-justify` Indicates justified text

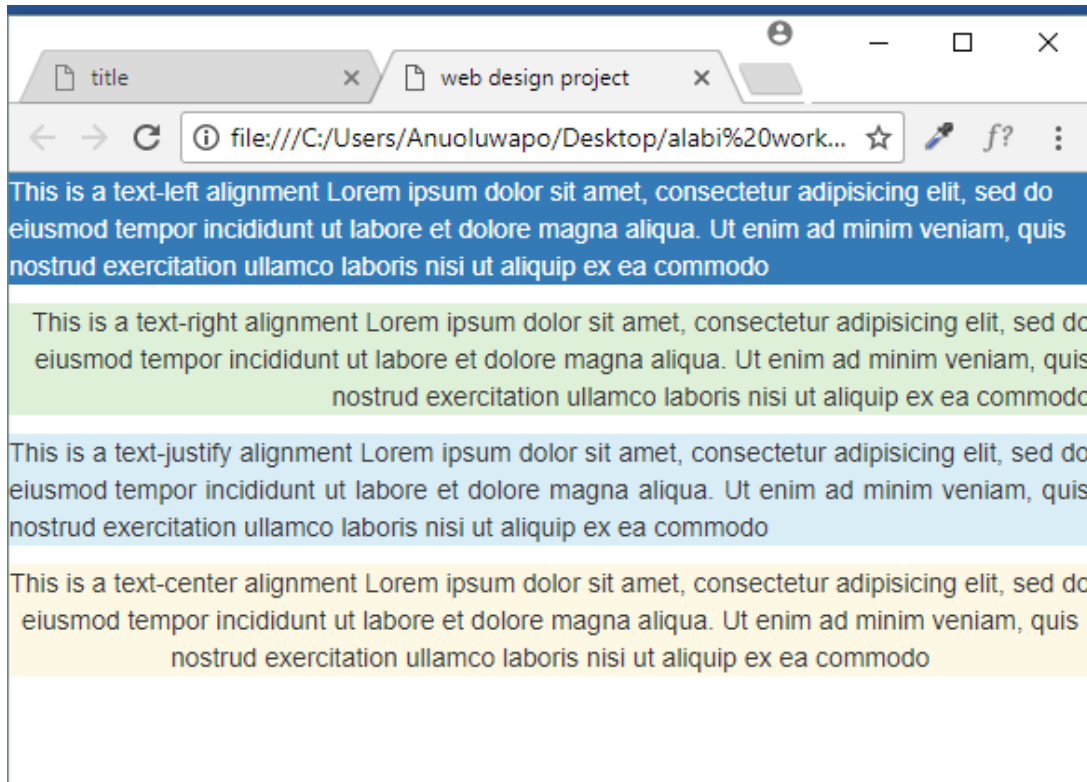
Code:

```
<p class="bg-primary text-left">
  This is a text-left alignment Lorem ipsum dolor sit amet,
  consectetur adipisicing elit, sed do eiusmod
  tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
  minim veniam,
  quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
  commodo
</p>
<p class="bg-success text-right">
  This is a text-right alignment Lorem ipsum dolor sit amet,
  consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
  labore et dolore magna aliqua. Ut enim ad minim veniamquis
  nostrud exercitation ullamco laboris nisi ut aliquip ex ea
  commodo
</p>
<p class="bg-info text-justify">
  This is a text-justify alignment Lorem ipsum dolor sit amet,
  consectetur adipisicing elit, sed do eiusmodo tempor incididunt ut
  labore et dolore magna aliqua. Ut enim ad minim veniam quis
  nostrud exercitation ullamco laboris nisi ut aliquip ex ea
  commodo
</p>
<p class="bg-warning text-center">
```

This is a text-center alignment Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

</p>

Result:



Text transform classes

- `.text-lowercase` Indicates lowercased text
- `.text-uppercase` Indicates uppercased text
- `.text-capitalize` Indicates capitalized text
- `.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)
- `.list-inline` Places all list items on a single line

Spacing

Spacing Notation

Spacing utilities that apply to all breakpoints, from xs to xl, have no breakpoint abbreviation in them. This is because those classes are applied from min-width: 0 and up, and thus are not bound by a media query. The remaining breakpoints, however, do include a breakpoint abbreviation.

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 padding

Where sides is 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

Where size is one of:

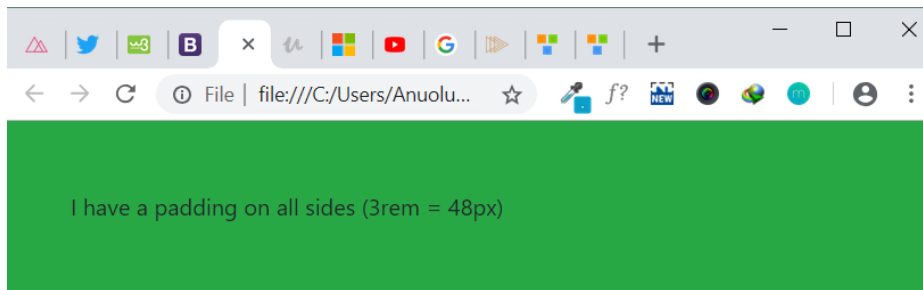
- 0 - for classes that eliminate the margin or padding by setting it to 0
- 1 - (by default) for classes that set the margin or padding to \$spacer * .25
- 2 - (by default) for classes that set the margin or padding to \$spacer * .5
- 3 - (by default) for classes that set the margin or padding to \$spacer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

Margin and Padding applying to all sides of an element

Code:

```
<div class="p-5 bg-success">I have a padding on all sides (3rem = 48px)</div>  
<div class="m-5 bg-info">I have a margin on all sides(3rem = 48px)</div>
```

Result:



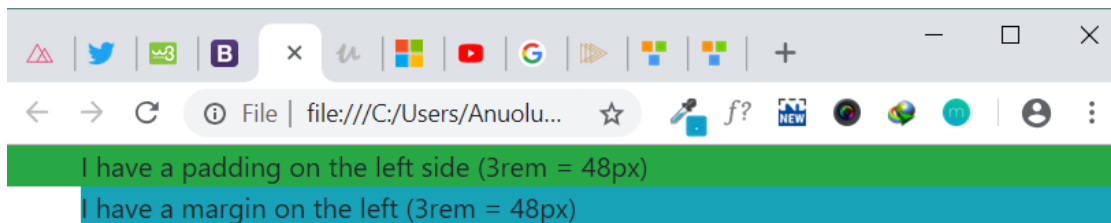
I have a margin on all sides (3rem = 48px) and a bottom padding (3rem = 48px)

Margin and padding applying to one side of an element

Code:

```
<div class="pl-5 bg-success">I have a padding on the left side (3rem = 48px)</div>  
<div class="ml-5 bg-info">I have a margin on the left (3rem = 48px)</div>
```

Result:



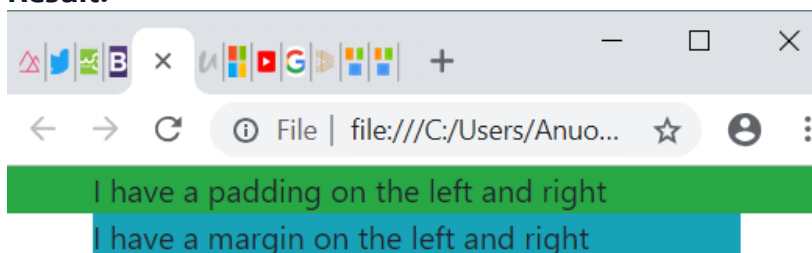
Applying horizontal margin and padding to an element

This is having margin and padding left and right i.e on both sides

Code:

```
<div class="px-5 bg-success">I have a padding on the left and right</div>  
<div class="mx-5 bg-info">I have a margin on the left and right</div>
```

Result:



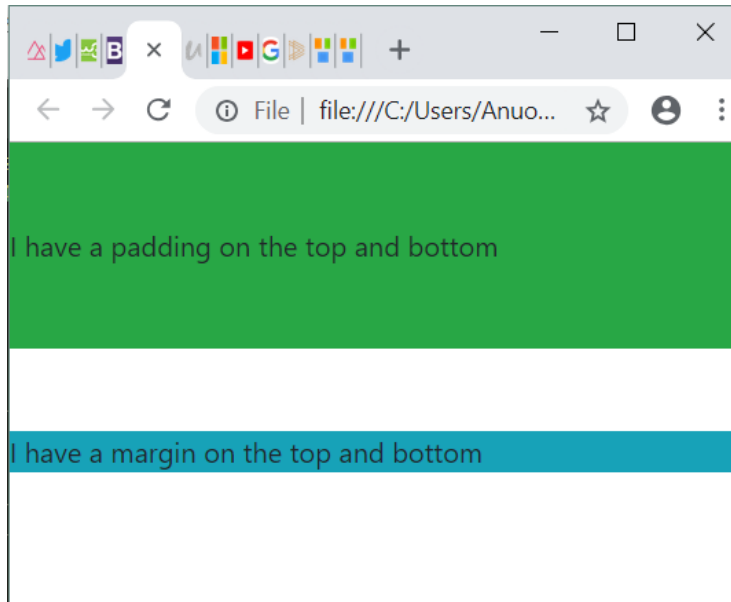
Applying Vertical margin and padding to an element

This is having margin and padding on top and bottom.

Code:

```
<div class="py-5 bg-success">I have a padding on the left and right</div>  
<div class="my-5 bg-info">I have a margin on the left and right</div>
```

Result



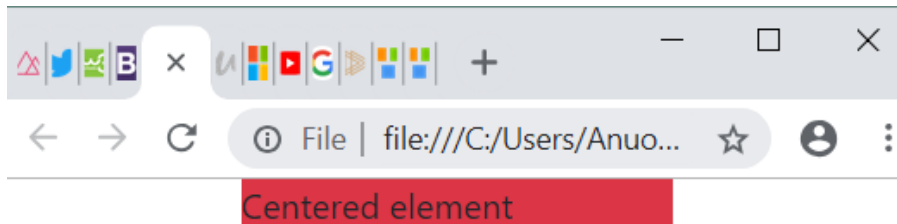
Horizontal Centering

Additionally, Bootstrap also includes an **.mx-auto** class for horizontally centering fixed-width block level content—that is, content that has **display: block** and a **width** set—by setting the horizontal margins to **auto**.

Code:

```
<div class="mx-auto bg-danger" style="width: 200px;">  
  Centered element  
</div>
```

Result:



Sizing

Easily make an element as wide or as tall (relative to its parent) with our width and height utilities.

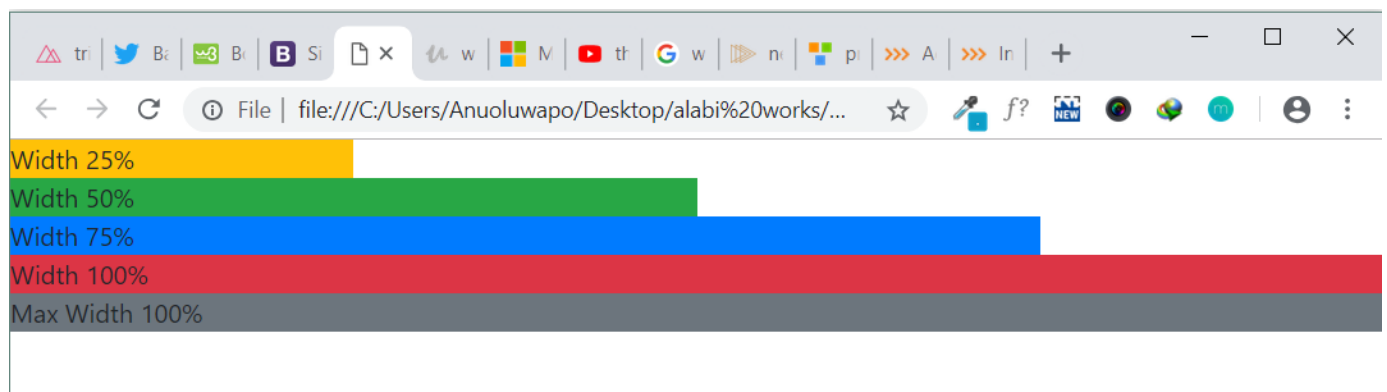
Width

Set the width of an element with the w-* classes (.w-25, .w-50, .w-75, .w-100, .mw-100):

Code:

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

Result:



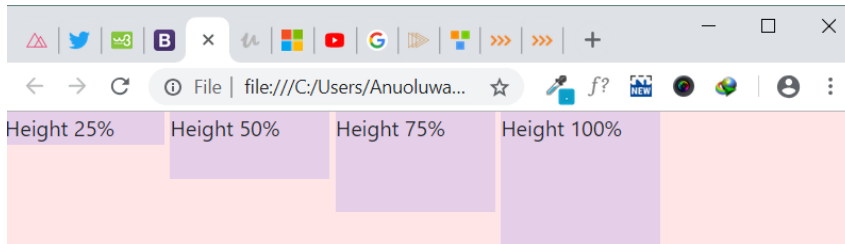
Height

Set the height of an element with the h-* classes (.h-25, .h-50, .h-75, .h-100, .mh-100):

Code:

```
<div style="height: 100px; background-color: rgba(255,0,0,0.1);">
  <div class="h-25 d-inline-block" style="width: 120px; background-color:
  rgba(0,0,255,.1)">Height 25%</div>
  <div class="h-50 d-inline-block" style="width: 120px; background-color:
  rgba(0,0,255,.1)">Height 50%</div>
  <div class="h-75 d-inline-block" style="width: 120px; background-color:
  rgba(0,0,255,.1)">Height 75%</div>
  <div class="h-100 d-inline-block" style="width: 120px; background-color:
  rgba(0,0,255,.1)">Height 100%</div>
</div>
```

Result:



Chapter 5

Tables

Basic table

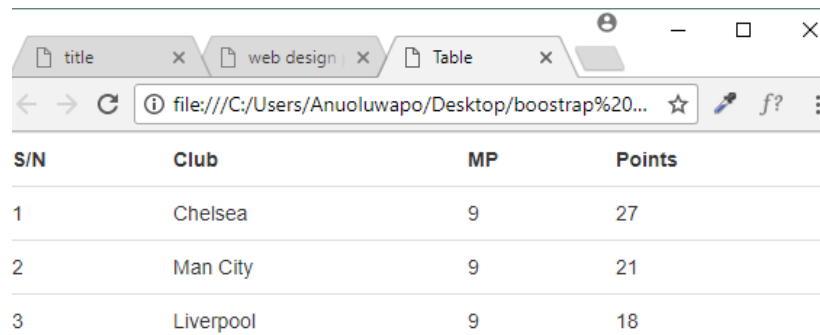
A basic Bootstrap table has a light padding and only horizontal dividers.

The **.table** class adds basic styling to a table:

Code:

```
<table class="table">
  <tr>
    <th>S/N</th>
    <th>Club</th>
    <th>MP</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Chelsea</td>
    <td>9</td>
    <td>27</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Man City</td>
    <td>9</td>
    <td>21</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Liverpool</td>
    <td>9</td>
    <td>18</td>
  </tr>
</table>
```

Result:



S/N	Club	MP	Points
1	Chelsea	9	27
2	Man City	9	21
3	Liverpool	9	18

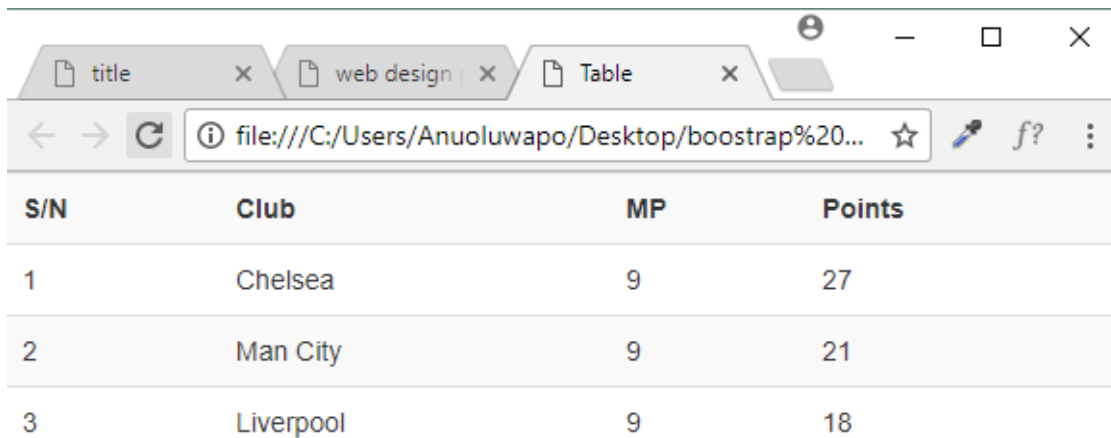
Striped row

The `.table-striped` class adds zebra-stripes to a table:

Code:

```
<table class="table table-striped">
  <tr>
    <th>S/N</th>
    <th>Club</th>
    <th>MP</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Chelsea</td>
    <td>9</td>
    <td>27</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Man City</td>
    <td>9</td>
    <td>21</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Liverpool</td>
    <td>9</td>
    <td>18</td>
  </tr>
</table>
```

Result:

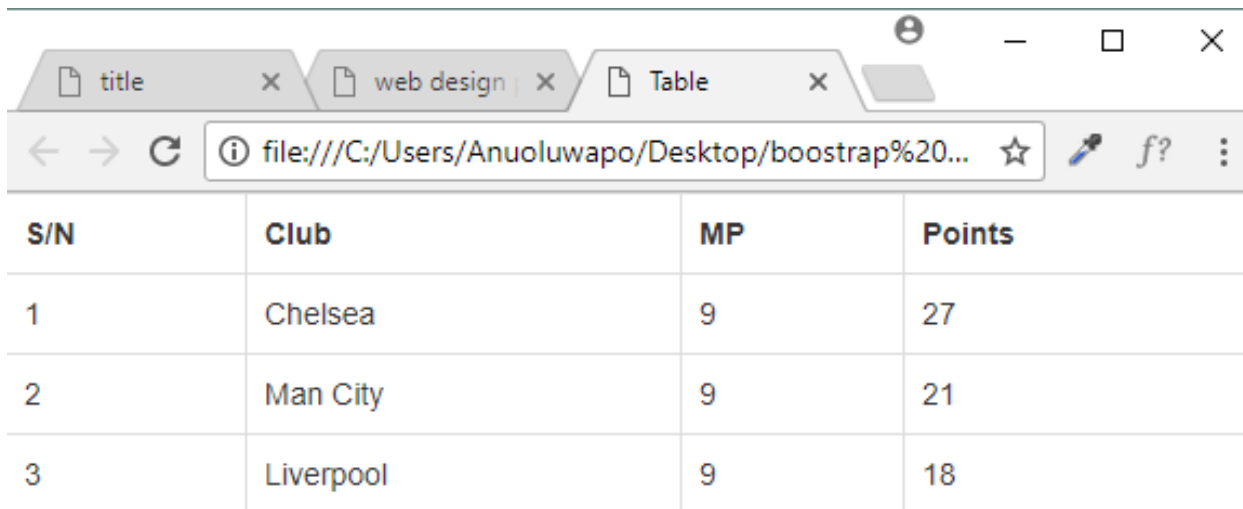


S/N	Club	MP	Points
1	Chelsea	9	27
2	Man City	9	21
3	Liverpool	9	18

Bordered table

The **.table-bordered** class adds borders on all sides of the table and cells:

Result:

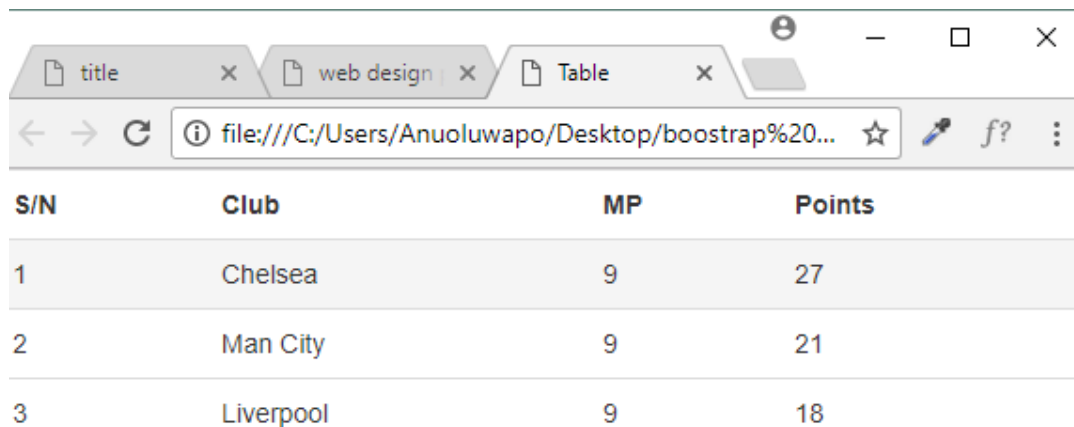


S/N	Club	MP	Points
1	Chelsea	9	27
2	Man City	9	21
3	Liverpool	9	18

Hover rows

The **.table-hover** class enables a hover state on table rows:

Result:



S/N	Club	MP	Points
1	Chelsea	9	27
2	Man City	9	21
3	Liverpool	9	18

Small Table

The **.table-sm** class makes a table more compact by cutting cell padding in half and it is placed in the opening tag of the table element.

Responsive Tables

The **.table-responsive** class creates a responsive table: an horizontal scrollbar is added to the table on screens that **are less than** 992px wide (if needed). When viewing on anything larger than 992px wide, there is no difference:

Code:

```
<div class="table-responsive">
  <table class="table table-striped">
    <tr>
      <th>S/N</th>
      <th>Club</th>
      <th>MP</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
    </tr>
    <tr>
      <td>1</td>
      <td>Chelsea</td>
      <td>9</td>
      <td>27</td>
      <td>27</td>
      <td>27</td>
      <td>27</td>
      <td>27</td>
      <td>27</td>
    </tr>
```

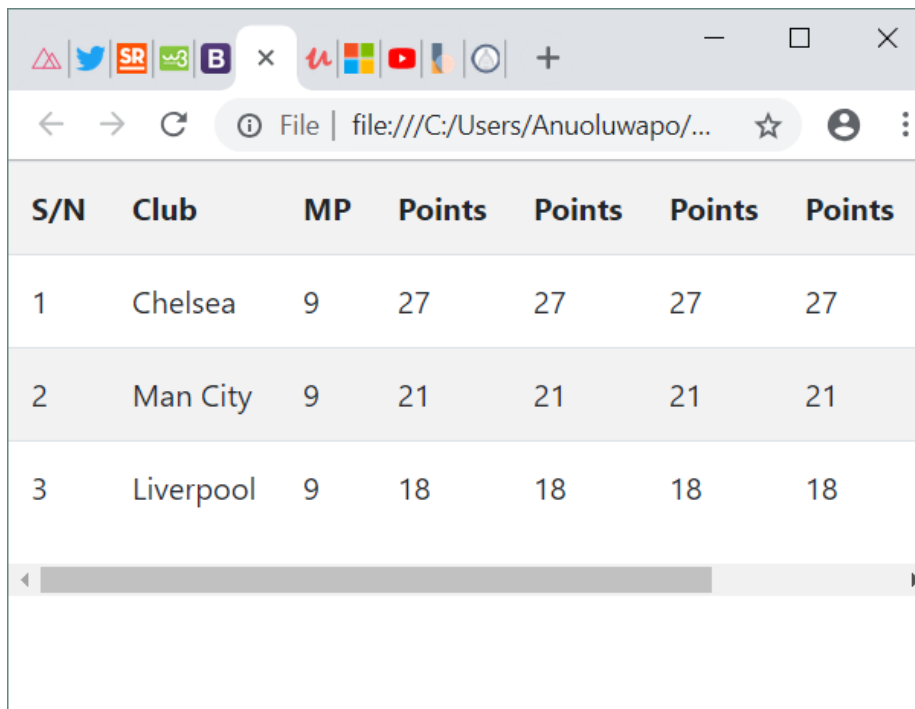


```

<tr>
  <td>2</td>
  <td>Man City</td>
  <td>9</td>
  <td>21</td>
  <td>21</td>
  <td>21</td>
  <td>21</td>
  <td>21</td>
  <td>21</td>
</tr>
<tr>
  <td>3</td>
  <td>Liverpool</td>
  <td>9</td>
  <td>18</td>
  <td>18</td>
  <td>18</td>
  <td>18</td>
  <td>18</td>
  <td>18</td>
</tr>
</table>
</div>

```

Result:



The screenshot shows a web browser window with a single tab. The address bar displays 'file:///C:/Users/Anuoluwapo/...'. The main content area contains a table with the following data:

S/N	Club	MP	Points	Points	Points	Points
1	Chelsea	9	27	27	27	27
2	Man City	9	21	21	21	21
3	Liverpool	9	18	18	18	18

Module 2

Chapter 6

Font Icons, Badges, labels, Alert

Font awesome

To use the Font Awesome icons, add the following line inside the `<head>` section of your HTML page:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Note: No downloading or installation is required!

You place Font Awesome icons by using the prefix `fa` and the icon's name.

Badges

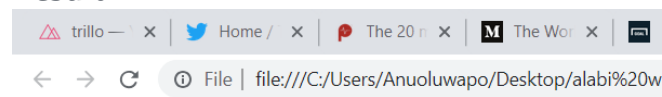
Badges are used to add additional information to any content. Use the `.badge` class together with a contextual class (like `.badge-secondary`) within `` elements to create rectangular badges. Note that badges scale to match the size of the parent element.

Example:

Code:

```
<h1>Example heading <span class="badge badge-secondary">New</span></h1>
<h2>Example heading <span class="badge badge-secondary">New</span></h2>
<h3>Example heading <span class="badge badge-secondary">New</span></h3>
<h4>Example heading <span class="badge badge-secondary">New</span></h4>
<h5>Example heading <span class="badge badge-secondary">New</span></h5>
<h6>Example heading <span class="badge badge-secondary">New</span></h6>
```

Result:



Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Contextual Badges

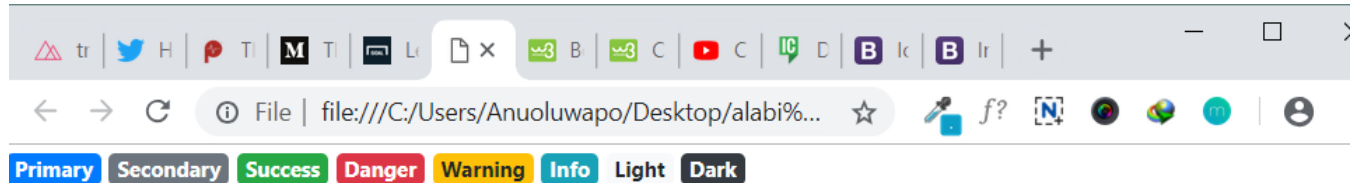
Use any of the contextual classes (`.badge-*`) to change the color of a badge:

Example:

Code:

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

Result:



Alert

Bootstrap provides an easy way to create predefined alert messages:

Success! This alert box indicates a successful or positive action.

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

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

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

Secondary! This alert box indicates a less important action.

Dark! Dark grey alert box.

Alerts are created with the **.alert** class, followed by one of the four contextual classes

.alert-success, .alert-info, .alert-warning or .alert-danger:

Code:

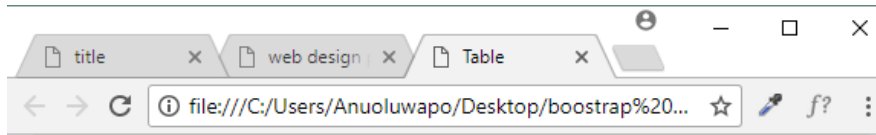
```
<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>

<div class="alert alert-info">
  <strong>Info!</strong> Indicates a neutral informative change or action.
</div>

<div class="alert alert-warning">
  <strong>Warning!</strong> Indicates a warning that might need attention.
</div>

<div class="alert alert-danger">
  <strong>Danger!</strong> Indicates a dangerous or potentially negative
  action.
</div>
```

Result:



Success! Indicates a successful or positive action.

Info! Indicates a neutral informative change or action.

Warning! Indicates a warning that might need attention.

Danger! Indicates a dangerous or potentially negative action.

Chapter 7

Nav

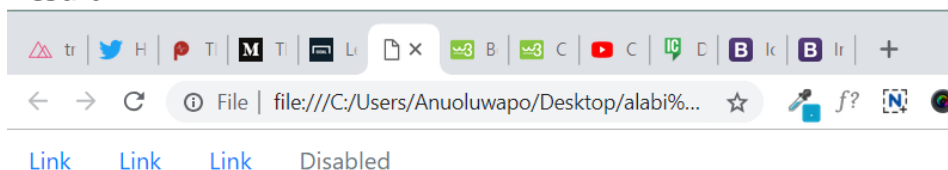
Nav-Menu

If you want to create a simple horizontal menu, add the `.nav` class to a `` element, followed by `.nav-item` for each `` and add the `.nav-link` class to their links:

Code:

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

Result:



Aligned Nav

Add the `.justify-content-center` class to the `.nav` class to center the nav, and the `.justify-content-end` class to right-align the nav.

Vertical Nav

Add the `.flex-column` class to the `.nav` class to create a vertical nav

Nav Tab

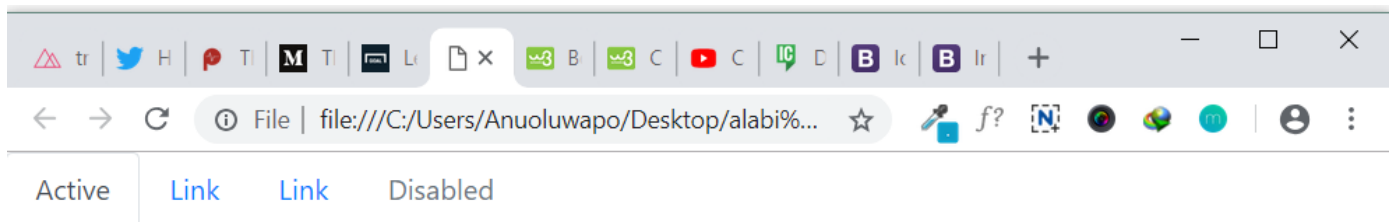
Turn the nav menu into navigation tabs with the **.nav-tabs** class. Add the **.active** class to the active/current link. If you want the tabs to be togglable, see the last example on this page.

Example:

Code:

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

Result:



Nav-pills

Pills are created with `<ul class="nav nav-pills">`. Also mark the current page with `<li class="active">`:

Code:

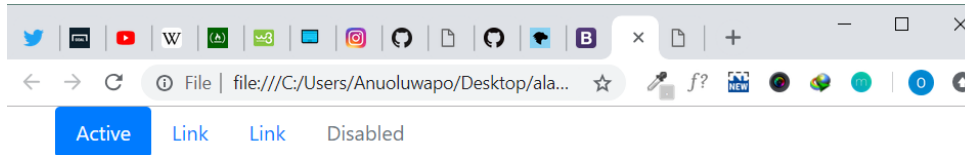
```
<div class="container">
  <ul class="nav nav-pills">
    <li class="nav-item">
      <a class="nav-link active" href="#">Active</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Disabled</a>
    </li>
  </ul>
</div>
```

```

    </li>
  </ul>
</div>

```

Result:



Justified Tabs/pills

Justify the tabs/pills with the .nav-justified class (equal width):

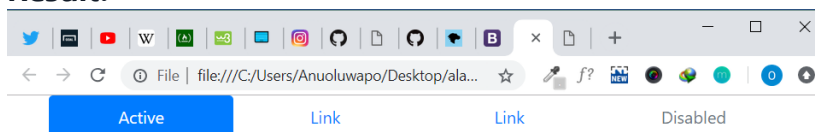
Code:

```

<div class="container">
  <ul class="nav nav-pills nav-justified">
    <li class="nav-item">
      <a class="nav-link active " href="#">Active</a>
    </li>
    <li class="nav-item">
      <a class="nav-link " href="#">Link</a>
    </li>
    <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>
</div>

```

Result:



Toggleable / Dynamic Tabs

To make the tabs toggleable, add the **data-toggle="tab"** attribute to each link. Then add a **.tab-pane** class with a unique ID for every tab and wrap them inside a **<div>** element with class **.tab-content**.

If you want the tabs to fade in and out when clicking on them, add the **.fade** class to **.tab-pane**:

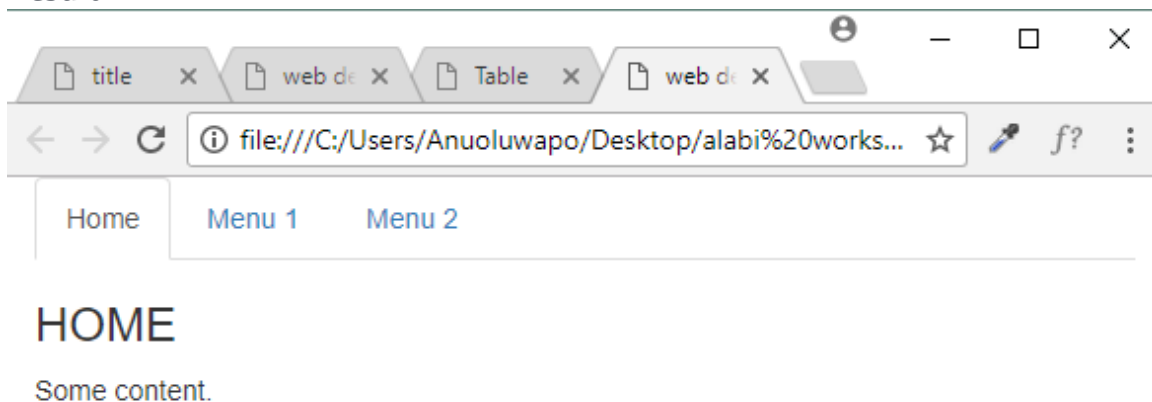
Example:

Code:

```
<div class="container">
  <ul class="nav nav-tabs">
    <li class="nav-item">
      <a class="nav-link active" data-toggle="tab" ref="#home">Active</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" data-toggle="tab" href="#menu1">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" data-toggle="tab" href="#menu2">Link</a>
    </li>
  </ul>

  <div class="tab-content">
    <div id="home" class="tab-pane fade show active">
      <h3>HOME</h3>
      <p>Some content.</p>
    </div>
    <div id="menu1" class="tab-pane fade">
      <h3>Menu 1</h3>
      <p>Some content in menu 1.</p>
    </div>
    <div id="menu2" class="tab-pane fade">
      <h3>Menu 2</h3>
      <p>Some content in menu 2.</p>
    </div>
  </div>
</div>
```

Result:



Chapter 5

Bootstrap navigation bar

Navigation bars

A navigation bar is a navigation header that is placed at the top of the page:

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xl|lg|md|sm` (stacks the navbar vertically on extra large, large, medium or small screens).

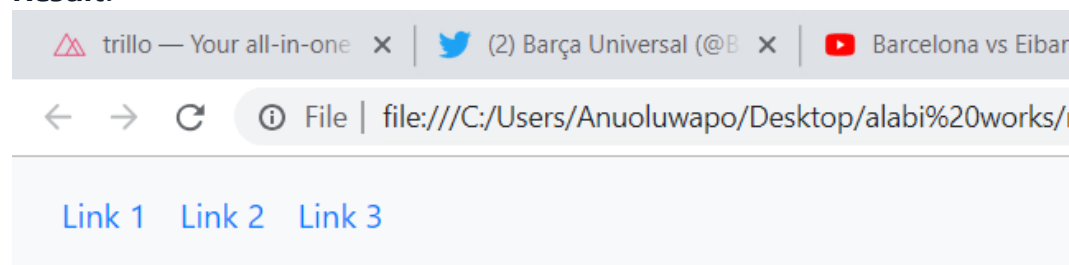
To add links inside the navbar, use a `` element with `class="navbar-nav"`. Then add `` elements with a `.nav-item` class followed by an `<a>` element with a `.nav-link` class:

For example:

Code:

```
<nav class="navbar navbar-expand-sm bg-light">
  <!-- Links -->
  <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>
```

Result:



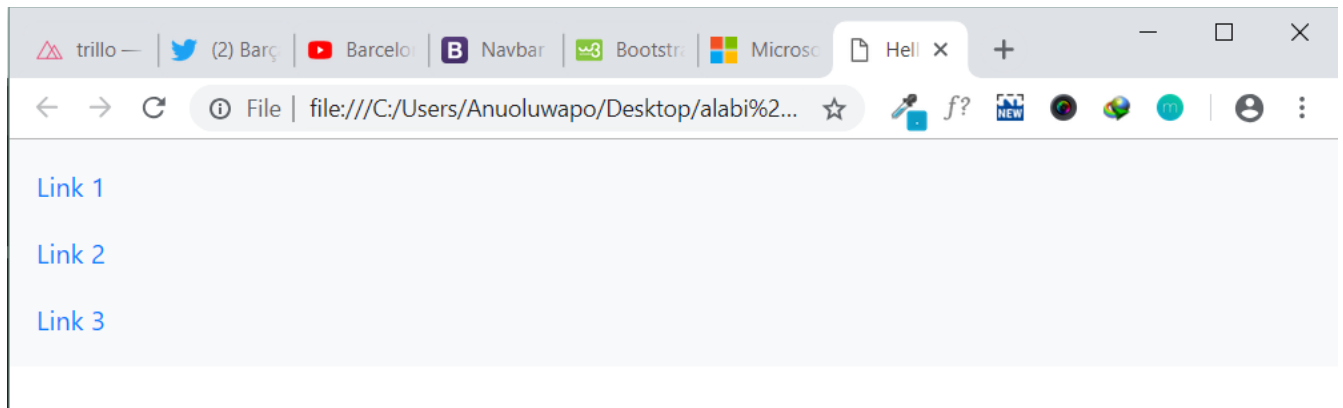
Vertical Navbar

Remove the `.navbar-expand-xl|lg|md|sm` class to create a vertical navigation bar:

Code:

```
<nav class="navbar bg-light">
  <!-- Links -->
  <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>
```

Result:



Centered Navbar

Add the `.justify-content-center` class to center the navigation bar.

The following example will center the navigation bar on medium, large and extra large screens. On small screens it will be displayed vertically and left-aligned (because of the `.navbar-expand-sm` class):

Code:

Fixed navigation bar

The navigation bar can also be fixed at the top or at the bottom of the page.

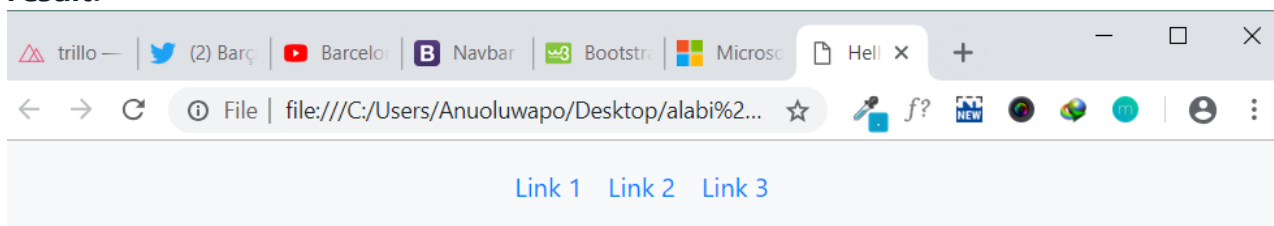
A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

The `.navbar-fixed-top` class makes the navigation bar fixed at the top:

Code:

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">
  <!-- Links -->
  <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>
```

result:



Brand / Logo

The **.navbar-brand** class is used to highlight the brand/logo/project name of your page

When using the **.navbar-brand** class on images, Bootstrap 4 will automatically style the image to fit the navbar vertically.

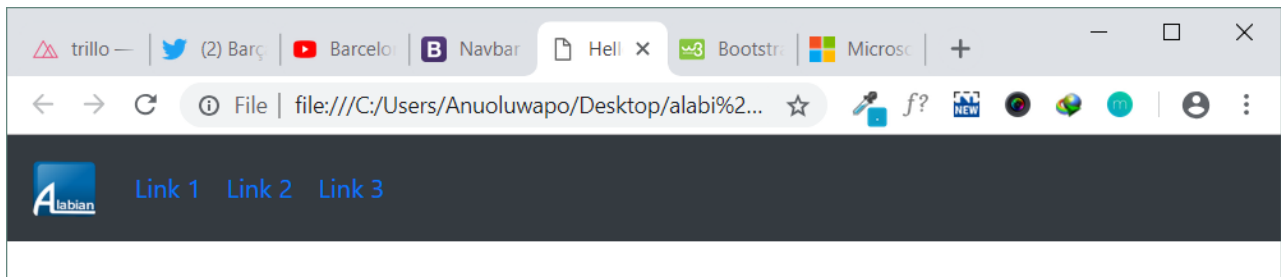
Navigation bar with dropdown

Navigation bars can also hold dropdown menus

Code:

```
<nav class="navbar navbar-expand-sm bg-dark">
  <a class="navbar-brand" href="#">
    
  </a>
  <!-- Links -->
  <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>
```

Result:



Collapsing The Navigation Bar

Very often, especially on small screens, you want to hide the navigation links and replace them with a button that should reveal them when clicked on.

To create a collapsible navigation bar, use a button with **class="navbar-toggler"**, **data-toggle="collapse"** and **data-target="#thetarget"**. Then wrap the navbar content (links, etc) inside a div element with **class="collapse navbar-collapse"**, followed by an id that matches the **data-target** of the button: **"thetarget"**.

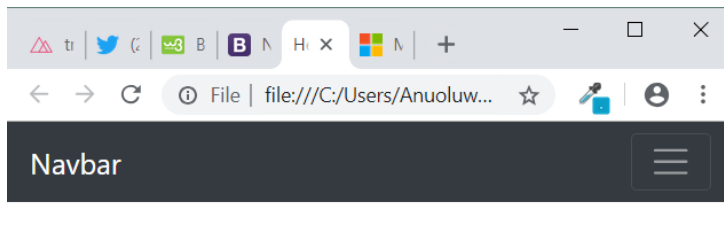
Code:

```
<div class="container">
  <nav class="navbar navbar-expand-md bg-dark navbar-dark">
    <!-- Brand -->
    <a class="navbar-brand" href="#">Navbar</a>

    <!-- Toggler/collapsible Button -->
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>

    <!-- Navbar links -->
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
      </ul>
    </div>
  </nav>
</div>
```

Result:



Navbar With Dropdown

Navbars can also hold dropdown menus:

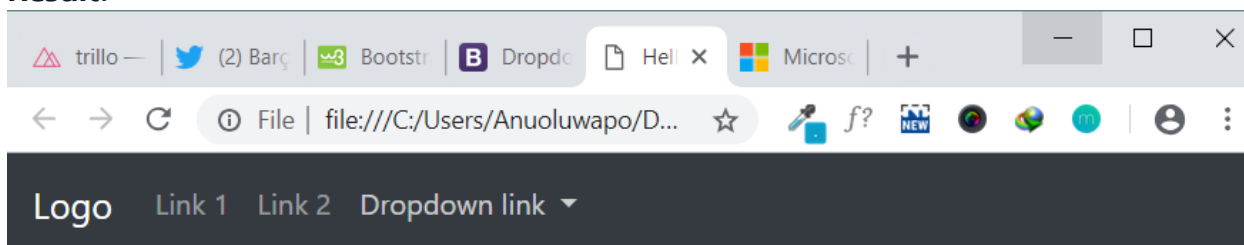
Code:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Logo</a>

  <!-- Links -->
  <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>

    <!-- Dropdown -->
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" data-
toggle="dropdown">
        Dropdown link
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Link 1</a>
        <a class="dropdown-item" href="#">Link 2</a>
        <a class="dropdown-item" href="#">Link 3</a>
      </div>
    </li>
  </ul>
</nav>
```

Result:



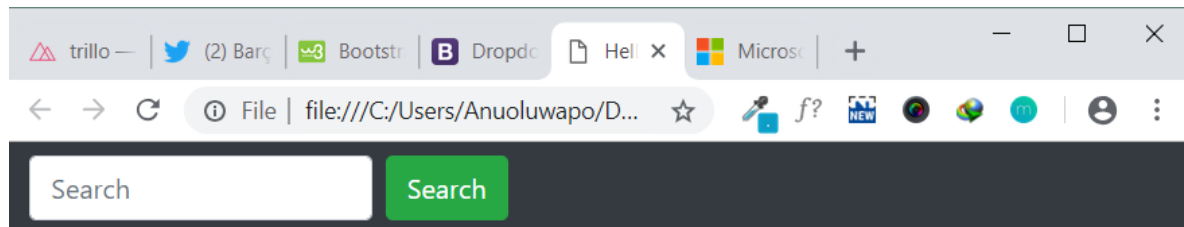
Navbar Forms and Buttons

Add a <form> element with class="form-inline" to group inputs and buttons side-by-side;

Code:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <form class="form-inline" action="/action_page.php">
    <input class="form-control mr-sm-2" type="text" placeholder="Search">
    <button class="btn btn-success" type="submit">Search</button>
  </form>
</nav>
```

Result:



Fixed Navigation Bar

The navigation bar can also be fixed at the top or at the bottom of the page.

A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

- The **.fixed-top** class makes the navigation bar fixed at the top
- The **.fixed-bottom** class makes the navigation bar fixed at the bottom

Chapter 6

Forms

Form controls automatically receive some global styling with Bootstrap: All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Here's a quick example to demonstrate Bootstrap's form styles.

Code:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp" placeholder="Enter email">
    <small id="emailHelp" class="form-text text-muted">We'll never share your
email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Result:

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Checkboxes and radios

Default checkboxes and radios are improved upon with the help of `.form-check`, a single class for both input types that improves the layout and behavior of their HTML elements.

Default (stacked)

By default, any number of checkboxes and radios that are immediate sibling will be vertically stacked and appropriately spaced with `.form-check`.

Code:

```
<div class="form-check">
  <input class="form-check-input" type="checkbox" value=""
id="defaultCheck1">
  <label class="form-check-label" for="defaultCheck1">
    Default checkbox
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="checkbox" value="" id="defaultCheck2"
disabled>
  <label class="form-check-label" for="defaultCheck2">
    Disabled checkbox
  </label>
</div>
```

Result:

- ☐ Default checkbox
- ☐ Disabled checkbox

Inline

Group checkboxes or radios on the same horizontal row by adding `.form-check-inline` to any `.form-check`.

Code:

```
<div class="form-check form-check-inline">
  <input class="form-check-input" type="checkbox" id="inlineCheckbox1"
value="option1">
  <label class="form-check-label" for="inlineCheckbox1">1</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="checkbox" id="inlineCheckbox2"
value="option2">
  <label class="form-check-label" for="inlineCheckbox2">2</label>
</div>
<div class="form-check form-check-inline">
```

```

☐

```

Result:

☐ 1 ☐ 2 ☐ 3 (disabled)

Code:

```

<div class="form-check form-check-inline">
  <input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio1" value="option1">
  <label class="form-check-label" for="inlineRadio1">1</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio2" value="option2">
  <label class="form-check-label" for="inlineRadio2">2</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio3" value="option3" disabled>
  <label class="form-check-label" for="inlineRadio3">3 (disabled)</label>
</div>

```

Result:

☐ 1 ☐ 2 ☐ 3 (disabled)

Layout

Since Bootstrap applies `display: block` and `width: 100%` to almost all our form controls, forms will by default stack vertically. Additional classes can be used to vary this layout on a per-form basis.

Form groups

The `.form-group` class is the easiest way to add some structure to forms. It provides a flexible class that encourages proper grouping of labels, controls, optional help text, and form validation messaging. By default it only applies margin-bottom, but it picks up additional styles in `.form-inline` as needed. Use it with `<fieldset>`s, `<div>`s, or nearly any other element.

Code:

```
<form>
  <div class="form-group">
    <label for="formGroupExampleInput">Example label</label>
    <input type="text" class="form-control" id="formGroupExampleInput"
placeholder="Example input">
  </div>
  <div class="form-group">
    <label for="formGroupExampleInput2">Another label</label>
    <input type="text" class="form-control" id="formGroupExampleInput2"
placeholder="Another input">
  </div>
</form>
```

Result:

Example label

Another label

Form grid

More complex forms can be built using the grid classes. Use these for form layouts that require multiple columns, varied widths, and additional alignment options.

Code:

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="First name">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Last name">
    </div>
  </div>
</form>
```

Result:

Form row

You may also swap `.row` for `.form-row`, a variation of our standard grid row that overrides the default column gutters for tighter and more compact layouts.

Code:

```
<form>
  <div class="form-row">
    <div class="form-group col-md-6">
      <input type="text" class="form-control" placeholder="First
name">
    </div>
    <div class="form-group col-md-6">
      <input type="text" class="form-control" placeholder="Last
name">
    </div>
  </div>
  <div class="form-group">
    <input type="text" class="form-control" placeholder="Address">
  </div>
</form>
```

Result:

Horizontal form

Create horizontal forms with the grid by adding the `.row` class to form groups and using the `.col-*-*` classes to specify the width of your labels and controls. Be sure to add `.col-form-label` to your `<label>`s as well so they're vertically centered with their associated form controls.

At times, you maybe need to use margin or padding utilities to create that perfect alignment you need.

Code:

```
<form>
  <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-
label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3"
placeholder="Email">
    </div>
  </div>
```

```

        </div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-
label">Password</label>
            <div class="col-sm-10">
                <input type="password" class="form-control" id="inputPassword3"
placeholder="Password">
            </div>
        </div>
    </form>

```

Result:

Email	<input type="text" value="Email"/>
Password	<input type="password" value="Password"/>

Inline forms

Use the `.form-inline` class to display a series of labels, form controls, and buttons on a single horizontal row. Form controls within inline forms vary slightly from their default states.

Code:

```

<form class="form-inline">
    <label class="sr-only" for="inlineFormInputName2">Name</label>
    <input type="text" class="form-control mb-2 mr-sm-2"
id="inlineFormInputName2" placeholder="Jane Doe">

    <label class="sr-only"
for="inlineFormInputGroupUsername2">Username</label>
    <div class="input-group mb-2 mr-sm-2">
        <div class="input-group-prepend">
            <div class="input-group-text">@</div>
        </div>
        <input type="text" class="form-control"
id="inlineFormInputGroupUsername2" placeholder="Username">
    </div>

    <div class="form-check mb-2 mr-sm-2">
        <input class="form-check-input" type="checkbox"
id="inlineFormCheck">
        <label class="form-check-label" for="inlineFormCheck">
            Remember me
        </label>
    </div>
    <button type="submit" class="btn btn-primary mb-2">Submit</button>

```

Result

<input type="text" value="Jane Doe"/>	<input type="text" value="@ Username"/>	<input type="checkbox"/> Remember me	<input type="button" value="Submit"/>
---------------------------------------	---	--------------------------------------	---------------------------------------

Chapter 6

Accordion, Modal, Card Deck, Bootstrap Carousel

The `.collapse` class indicates a collapsible element (a `<div>` in our example); this is the content that will be shown or hidden with a click of a button.

To control (show/hide) the collapsible content, add the `data-toggle="collapse"` attribute to an `<a>` or a `<button>` element. Then add the `data-target="#id"` attribute to connect the button with the collapsible content.

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

Use the `data-parent` attribute to make sure that all collapsible elements under the specified parent will be closed when one of the collapsible item is shown.

Code:

```
<div id="accordion">

  <div class="card">
    <div class="card-header">
      <a class="card-link" data-toggle="collapse" href="#collapseOne">
        Collapsible Group Item #1
      </a>
    </div>
    <div id="collapseOne" class="collapse show" data-parent="#accordion">
      <div class="card-body">
        Lorem ipsum..
      </div>
    </div>
  </div>

  <div class="card">
    <div class="card-header">
      <a class="collapsed card-link" data-toggle="collapse"
href="#collapseTwo">
        Collapsible Group Item #2
      </a>
    </div>
    <div id="collapseTwo" class="collapse" data-parent="#accordion">
      <div class="card-body">
        Lorem ipsum..
      </div>
    </div>
  </div>

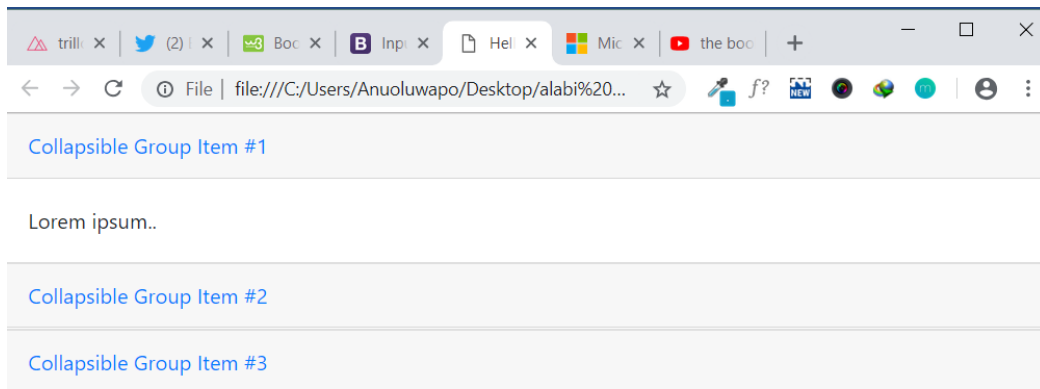
  <div class="card">
    <div class="card-header">
```

```

        <a class="collapsed card-link" data-toggle="collapse"
href="#collapseThree">
            Collapsible Group Item #3
        </a>
    </div>
    <div id="collapseThree" class="collapse" data-parent="#accordion">
        <div class="card-body">
            Lorem ipsum..
        </div>
    </div>
</div>

```

Result:



Modal

How To Create a Modal

The following example shows how to create a basic modal:

Example:

Code:

```

<!-- Trigger the modal with a button -->
<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-
target="#myModal">Open Modal</button>

<!-- Modal -->
<div id="myModal" class="modal fade" role="dialog">
    <div class="modal-dialog">

        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;
            </button>
            <h4 class="modal-title">Modal Header</h4>
        </div>
        <div class="modal-body">
            <p>Some text in the modal.</p>
        </div>
    </div>

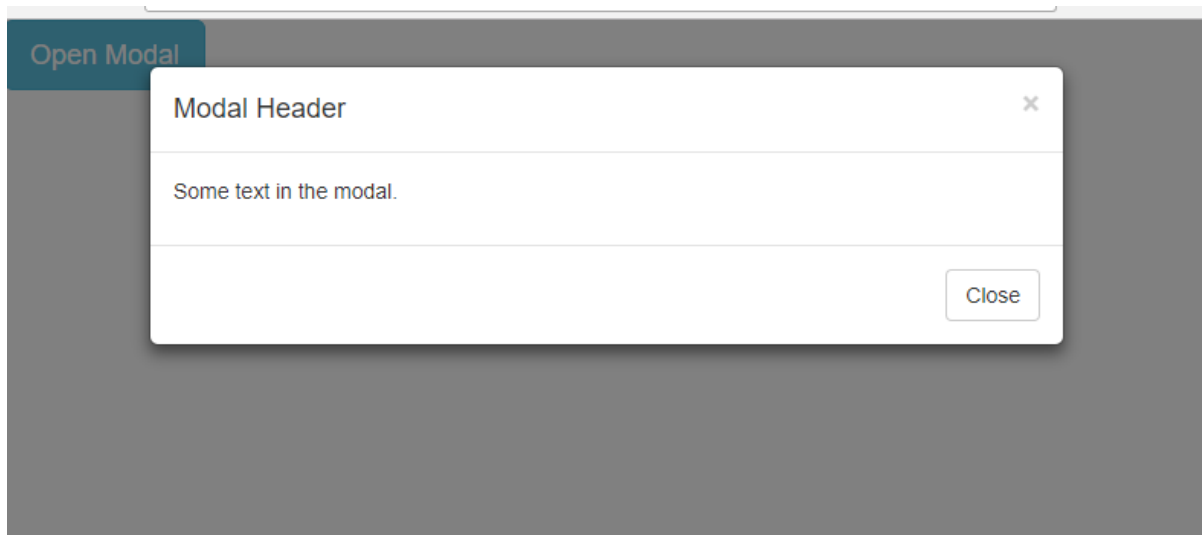
```

```

        <div class="modal-footer">
            <button type="button" class="btn btn-default" data-dismiss="modal">
                Close</button>
        </div>
    </div>
</div>

```

Result:



Example Explained

The "Trigger" part:

To trigger the modal window, you need to use a button or a link.

Then include the two data-* attributes:

data-toggle="modal" opens the modal window **data-target="#myModal"** points to the id of the modal

The "Modal" part:

The parent **<div>** of the modal must have an ID that is the same as the value of the data-target attribute used to trigger the modal ("myModal").

The **.modal** class identifies the content of **<div>** as a modal and brings focus to it.

The **.fade** class adds a transition effect which fades the modal in and out. 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.

The "Modal content" part:

The **<div>** with **class="modal-content"** styles the modal (border, background-color, etc.). Inside this **<div>**, add the modal's header, body, and footer.

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 is used to define the style for the body of the modal. Add any HTML markup here; paragraphs, images, videos, etc.

The **.modal-footer** class is used to define the style for the footer of the modal. Note that this area is right aligned by default.

Bootstrap carousel

The Carousel plugin is a component for cycling through elements, like a carousel (slideshow).

Code:

```
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>Heading 1</h5>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipisicing elit.
          Harum officia illum molestiae ipsam totam, sed eos facere, alias
        </p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>Heading 2</h5>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipisicing elit.
          Harum officia illum molestiae ipsam totam, sed eos facere, alias, similique
          labore voluptate ipsum odit eaque perferendis ducimus esse dolor facilis?
          Excepturi.
        </p>
      </div>
    </div>
    <div class="carousel-item">
      
```

```

        <div class="carousel-caption d-none d-md-block">
            <h5>Heading 3</h5>
            <p>
                Lorem ipsum dolor sit amet, consectetur adipisicing elit.
                Harum officia illum molestiae ipsam totam, sed eos facere, alias, similique
                labore voluptate ipsum odit eaque perferendis ducimus esse dolor facilis?
                Excepturi.
            </p>
        </div>
    </div>
</div>
<a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

```

Result



Example Explained

The outermost <div>:

Carousels require the use of an id (in this case id="myCarousel") for carousel controls to function properly.

The class="carousel" specifies that this <div> contains a carousel.

The .slide class adds a CSS transition and animation effect, which makes the items slide when showing a new item. Omit this class if you do not want this effect.

The data-ride="carousel" attribute tells Bootstrap to begin animating the carousel immediately when the page loads.

The "Indicators" part:

The indicators are the little dots at the bottom of each slide (which indicates how many slides there is in the carousel, and which slide the user are currently viewing).

The indicators are specified in an ordered list with class `.carousel-indicators`.

The **data-target** attribute points to the id of the carousel.

The **data-slide-to** attribute specifies which slide to go to, when clicking on the specific dot.

The "Wrapper for slides" part:

The slides are specified in a `<div>` with class `.carousel-inner`.

The content of each slide is defined in a `<div>` with class **.item**. This can be text or images.

The **.active** class needs to be added to one of the slides. Otherwise, the carousel will not be visible.

The "Left and right controls" part:

This code adds "left" and "right" buttons that allows the user to go back and forth between the slides manually.

The **data-slide** attribute accepts the keywords **"prev"** or **"next"**, which alters the slide position relative to its current position.

Card

A card in Bootstrap 4 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.

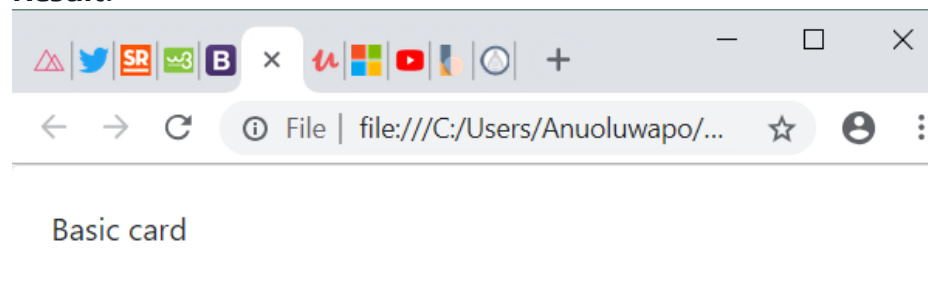
Basic Card

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

Code:

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

Result:





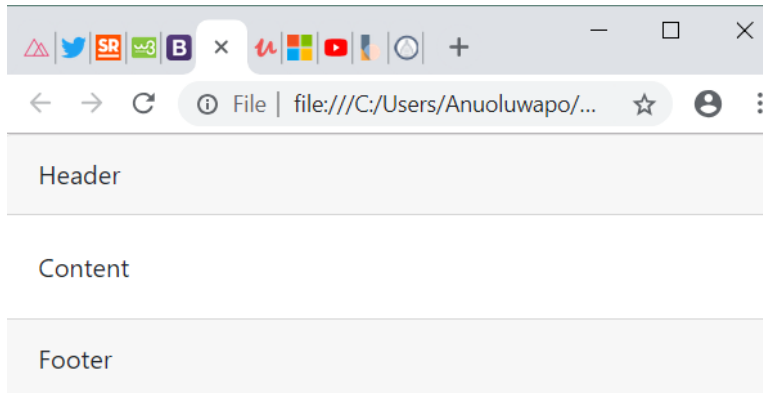
Header and Footer

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

Code:

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

Result:



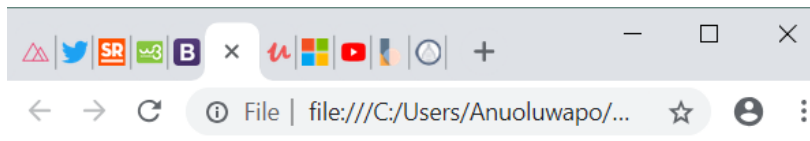
Titles, text, and links

Use **.card-title** to add card titles to any heading element. The **.card-text** class is used to remove bottom margins for a `<p>` element if it is the last child (or the only one) inside **.card-body**. The **.card-link** class adds a blue color to any link, and a hover effect.

Code:

```
<div class="card">
  <div class="card-body">
    <h4 class="card-title">Card title</h4>
    <p class="card-text">Some example text. Some example text.</p>
    <a href="#" class="card-link">Card link</a>
    <a href="#" class="card-link">Another link</a>
  </div>
</div>
```

Result:



Card title

Some example text. Some example text.

[Card link](#) [Another link](#)

Card Images

Add **.card-img-top** or **.card-img-bottom** to an `` to place the image at the top or at the bottom inside the card. Note that we have added the image outside of the **.card-body** to span the entire width:

Code:

```
<div class="card" style="width:400px">
  
  <div class="card-body">
    <h4 class="card-title">John Doe</h4>
    <p class="card-text">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fugiat
      facilis iusto ab quo harum, molestias assumenda enim tempore reprehenderit
      quasi delectus velit distinctio, eius officiis possimus consequatur quae
      error aliquam.
    </p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>
```

Result:



Card Image Overlays

Turn an image into a card background and use **.card-img-overlay** to add text on top of the image:

Code:

```
<div class="card text-white" style="width: 900px">
  
  <div class="card-img-overlay">
    <h4 class="card-title">John Doe</h4>
    <p class="card-text">Some example text.</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>
```

Result:



Card Deck

The **.card-deck** class creates a grid of cards that are of **equal height and width**. The layout will automatically adjust as you insert more cards.

Note: The cards are displayed vertically on small screens (less than 576px):

Code:

```
<div class="card-deck">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Card title</h5>
      <p class="card-text">This is a wider card with supporting text
      below as a natural lead-in to additional content. This content is a little
      bit longer.</p>
    </div>
    <div class="card-footer">
      <small class="text-muted">Last updated 3 mins ago</small>
    </div>
  </div>
  <div class="card">
```



```


<div class="card-body">
  <h5 class="card-title">Card title</h5>
  <p class="card-text">This card has supporting text below as a
natural lead-in to additional content.</p>
</div>
<div class="card-footer">
  <small class="text-muted">Last updated 3 mins ago</small>
</div>
</div>
</div>

```

Result:



Card title

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Last updated 3 mins ago



Card title

This card has supporting text below as a natural lead-in to additional content.

Last updated 3 mins ago

For more card information, you can go through this link
<https://getbootstrap.com/docs/4.2/components/card/>