

What is Bootstrap

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.



History of Bootstrap

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

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

Why use Bootstrap

Following are the main advantage of Bootstrap:

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
- It facilitates users to develop a responsive website.
- It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

What is a responsive website

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

What Bootstrap package contains

Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, and background.

CSS: Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

Components: Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

JavaScript Plugins: Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

Customize: Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

What is Bootstrap 4?

Bootstrap 4 is the newest and latest version of Bootstrap. It is the most popular HTML, CSS, JavaScript framework for developing responsive, mobile first websites.

Bootstrap 3 vs. Bootstrap 4

Bootstrap 4 has some new components, faster stylesheet, more buttons, effects and more responsiveness.

Bootstrap 4 supports some the latest, stable releases of all major browsers and platforms.

How to Use Bootstrap With HTML

Bootstrap is an open-source HTML, CSS, and Javascript framework that makes UI development easier with pre-built responsive classes and other utilities. This open-source framework currently powers more than 18 million websites and the numbers are expected to grow in the future.

The reusable JS and CSS JavaScripts available with Bootstrap can help you achieve the desired results you are after. Using Bootstrap with HTML can be quite confusing for beginners. Here is a comprehensive step-by-step guide on how to do that.

Getting started (Installation)

To include Bootstrap in HTML, you can use one of the **three** following methods:

1. Using Bootstrap CDN

CSS

Simply copy this stylesheet link to the <head> of your HTML file

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

JS

The JavaScript functionalities of some components like dropdowns, tabs, etc depend on jQuery and popper.js.

So, include jQuery and popper.js in the following order right before loading the bootstrap JavaScript file for proper functioning.

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-U02eT0CpHqdsSJQ6hJty5KVphtPhzWj9W0c1HTMga3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEeAaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
```

2. Downloading the files locally

Instead of using CDN, you can download the files locally to your project folder

from <https://getbootstrap.com/docs/4.3/getting-started/download/>

Once you download the file, you can include bootstrap.min.css file in the <head> and bootstrap.min.js in <body>. Even if you are using the downloaded bootstrap file, you have to include jquery.min.js and popper.min.js before loading bootstrap.min.js

3. Using package managers

Bootstrap can be easily pulled into any project using package managers such as 'npm', 'yarn' etc. Since npm is the most popular package manager used by front-end developers, we are going ahead with the npm command for installing bootstrap. Type the following command in your project folder (assuming you have initialized npm in the project)

```
npm install bootstrap
```

This command will download a local copy of bootstrap files inside the 'node_modules' folder in your project. You can then include bootstrap.min.css file in the <head> and bootstrap.min.js in <body>. As mentioned in method no.2, you have to include jquery.min.js and popper.min.js before loading bootstrap.min.js.

Usage

A sample HTML file after including bootstrap CSS and js will look like this (We have used CDN method in the below example. You can edit the HREF attribute for stylesheet and src for javascript with the proper path if you choose any other method):

```
<!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/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0dixCbmQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <title>Bootstrap example</title>
  </head>
  <body>

    <!-- 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+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-U02eT0CpHqdsSJQ6hJty5KVphtPhzWj9W0c1HTMga3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>
</html>
```

Once you have included Bootstrap in your project using any of the above methods, you can use all the components available in bootstrap easily.

For eg, to create a Bootstrap button in Primary color, you have to simply use the markup

```
<button type="button" class="btn btn-primary">Primary
</button>
```

With just these two classes, you will get a button with all relevant styles, thus saving you the time of styling the default HTML button.



Like buttons, Bootstrap has a lot of useful components like navbar, alerts, badges, dropdowns, etc.

First Bootstrap Example

Add the HTML 5 doctype: Bootstrap uses HTML elements and CSS properties, so you have to add the HTML 5 doctype at the beginning of the page with lang attribute and correct character set.

Ex:

```
<!DOCTYPE html>
<html lang="en">
<head><meta http-equiv="Content-Type" content="text/html; charset=windows-1252">

<title>Any title</title>
</head>
<body>
//write code
</body>
</html>
```

Bootstrap is mobile friendly: Bootstrap 3 is designed to be responsive to mobile devices.

Mobile-first styles are part of the core framework of Bootstrap. You have to add the following <meta> tag inside the <head> element for proper rendering and touch zooming:

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

Note: The "**width=device-width**" part is used to set the width of the page to follow the screen-width of the device (vary according to the devices).

The **initial-scale=1** part is used to set the initial zoom level when the page is first loaded by the browser.

Containers: container is used to wrap the site contents. There are two container classes.

- The **.container class** provides a responsive fixed width container.
- The **.container-fluid** class provides a full width container, spanning the entire width of the viewport.

Note: A container cannot be placed inside a container.

First Bootstrap Example (with responsive fixed width container)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This is a Bootstrap example</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css
/bootstrap.min.css">
</head>
<body>
<div class="container">
  <h1> First Bootstrap web page</h1>
  <p>Write your text here..</p>
</div>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">
</script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js
"></script>
</body>
</html>
```

Breakpoints

Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap.

Available breakpoints

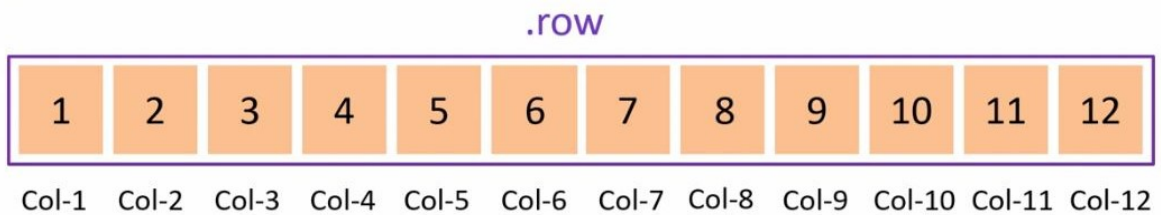
Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Bootstrap Grid System

The Bootstrap Grid System is used for responsive layouts.

A responsive layout represents the way elements align in the page on different resolutions. It is important you understand how to use the grid before learning about any other Bootstrap component, because whatever element you use, you will need to place it somewhere on the screen.



Bootstrap Containers

A Bootstrap container is an element with the class `.container`. The container is the root of the Bootstrap grid system and it is used to control the width of the layout. The Bootstrap container contains all the elements in a page. This means your page should have the following structure: first the body of the HTML page, inside of it you should add the container and all the other elements inside the container.

```
<body>
  <div class="container">
    ...
  </div>
</body>
```

Container Class with different Break Points

Container Width		
Extra Large	$\geq 1200\text{px}$	1140px
Large	$\geq 992\text{px}$	960px
Medium	$\geq 768\text{px}$	720px
Small	$\geq 576\text{px}$	540px
Extra Small	$< 576\text{px}$	Auto

The simple `.container` class sets the width of the layout depending on the width of the screen. It places the content in the middle of the page aligning it horizontally. There is equal space between the Bootstrap container and the left and the right edge of the page.

The `.container` scales down in width as the screen width narrows and becomes full-width on mobile. The width of the container is defined inside the Bootstrap library for every screen size.

A full-width container takes 100% of the screen size regardless of the screen width. To use it you need to add the class `.container-fluid`.

```
<div class="container">
  Hello! I am in a simple container.
</div>
```

```
<div class="container-fluid">
  Hello! I am in a full-width container.
</div>
```

Fluid containers

Use `.container-fluid` for a full width container, spanning the entire width of the viewport.

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

Bootstrap Rows

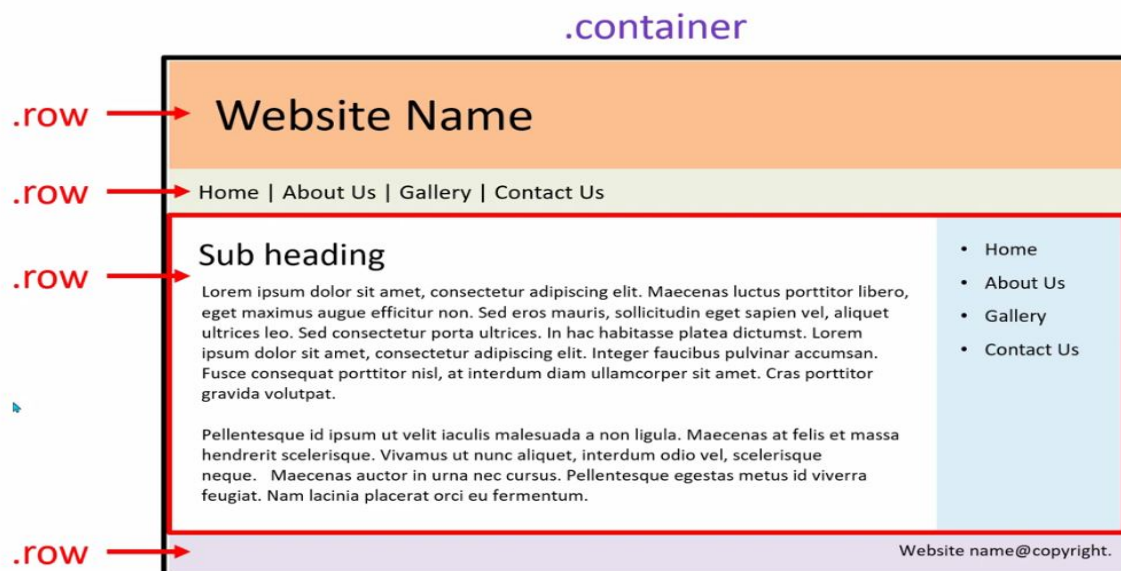
Bootstrap rows are horizontal slices of the screen. They are used only as wrappers for columns. To use them, you need the `.row` class.

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

Here are the most important things you need to remember about Bootstrap rows:

- **They are only used for containing columns.** If you place other elements inside the row along with columns you will not get the expected result.

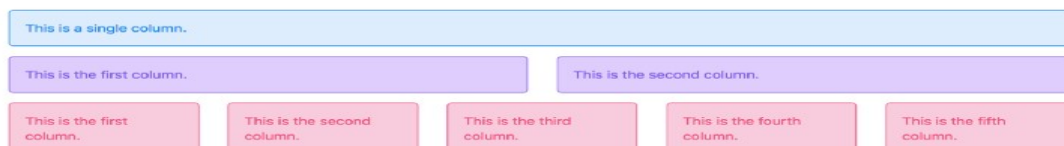
- **They have to be placed in containers.** If you don't do this, you will get a horizontal scroll on your page. This happens because rows have negative left and right margins of 15. The container has 15px paddings so it counteracts the margins.
- **The columns have to be children of the row.** Otherwise they will not align. The rows and columns are created to work together in this strict hierarchy.



Bootstrap Columns

We can now get to the nice part of this tutorial, the Bootstrap columns. Columns are great! They help you divide the screen horizontally.

If you place a single column in your row, it will take up all the width. If you add two columns, they will each take 1/2 from the width. And so it goes for any number of columns.



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



```

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

```

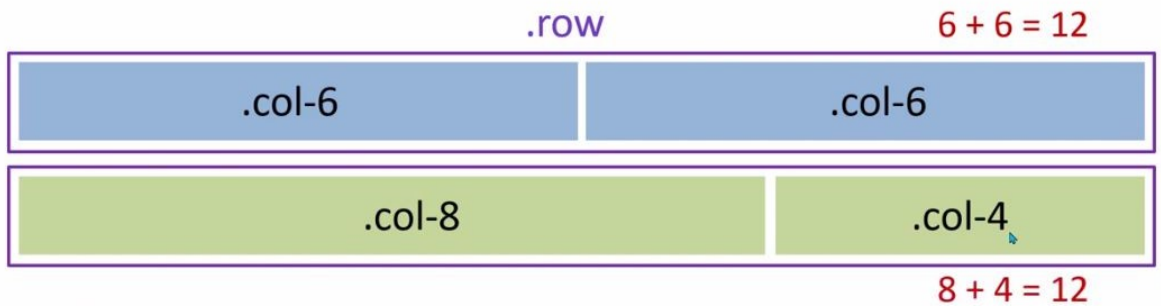
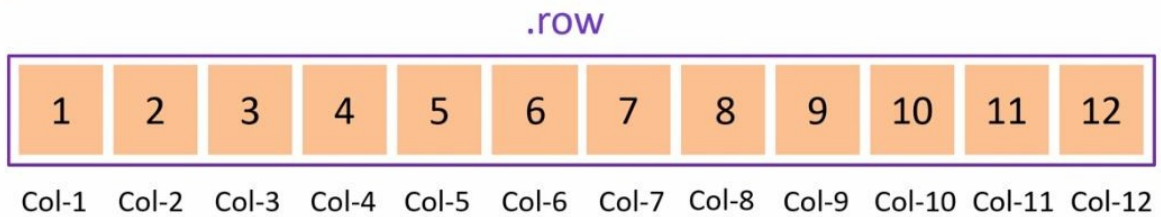
Side note: Columns are not coloured. I just added colours for a more visually compelling description/so they look pretty.

Setting Sizes for Columns

Using the `.col` class sets the width for the column dynamically. That means that depending on the number of columns in a row, the width of a column will be the width of the container divided by the number of columns.

But there is another way to define columns. You can use classes for columns and define their size.

By default, the Bootstrap grid consists of 12 columns. You can select any size from 1 to 12 for your column. If you want 3 equal columns, you can use `.col-4` for each one (because 3×4 cols each = 12). Or you can set different sizes for them. Here are some examples:



```

<div class="row">
  <div class="col-6">
    ...
  </div>
  <div class="col-6">
    ...
  </div>
</div>
<div class="row">
  <div class="col-8">
    ...
  </div>
  <div class="col-4">
    ...
  </div>
</div>
</div>

```

If the sum of the cols in your row doesn't get to 12, then they don't fill the whole row. If the sum of the columns goes beyond 12 then it will move to the next line. The first line will only display the first elements that add up to 12 or lower.

Setting Breakpoints for Columns

If you take the example above and want to display it on mobile, you may run into some problems. Displaying five columns on mobile will make the content unreadable.

This is where one of the most powerful Bootstrap components comes into play. In order to have different layouts on different screens you won't need to write media queries, instead you can use the column breakpoints.

A breakpoint is a Bootstrap variable that stands for a screen resolution. When you are specifying a breakpoint for a class, you are telling the class to be active only for resolutions that are at least as big as the number that the breakpoint holds.

The simplest class that we will learn about is the `.col-[breakpoint]` class. When you use this class, you are defining the behaviour for the columns only when they are displayed on devices that have a resolution of at least the defined breakpoint. Up to the given breakpoint, your columns will align vertically by default. And after your breakpoint, they will align horizontally because of the class.

Bootstrap has 4 breakpoints that you can use:

- `.col-sm` for larger mobile phones (devices with resolutions $\geq 576\text{px}$);
- `.col-md` for tablets ($\geq 768\text{px}$);
- `.col-lg` for laptops ($\geq 992\text{px}$);
- `.col-xl` for desktops ($\geq 1200\text{px}$)


Let's say you want to display two columns one after another vertically on small screens and on the same line on bigger screens. You will need to specify the breakpoint where the columns go on the same line.

In our example we will use the `.col-lg` breakpoint and see how the columns look on different screens. For resolutions that are lower than the given breakpoint ($< 992\text{px}$) the columns will be displayed vertically. This means that on mobile devices and tablets, the columns will look like this:



The display for resolutions $< 992\text{px}$ (mobile devices).

And for devices that have a resolution that is higher or equal to the breakpoint ($\geq 992\text{px}$) the columns will go on the same row. This means that on laptops and desktops you will get this result:



The display for resolutions $\geq 992\text{px}$ (laptops and bigger screens).

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

If you wanted for the 2 columns to go on the same line starting with larger mobile phones you would use `.col-sm`, for tablets `.col-md` and for extra large screens `.col-xl`.

Setting Sizes and Breakpoints for Columns

You can combine the sizes and breakpoints and use a single class with the format `.col-[breakpoint]-[size]`.

For example, if you want three columns of different sizes to align on a row starting with the laptop resolution you need to do this:

```
<div class="row">
  <div class="col-lg-4">
    ...
  </div>
  <div class="col-lg-3">
    ...
  </div>
  <div class="col-lg-5">
    ...
  </div>
</div>
```

You will get this result on resolutions $< 992\text{px}$:

This is a column.

This is a column.

This is a column.

And for screens that are $\geq 992\text{px}$:

This is a column.

This is a column.

This is a column.

But what if you want to display one column per row on small mobile resolutions, two columns per row on tablets and four on laptops or devices with higher resolutions?

Then you add multiple classes for a single column to describe the behaviour for every resolution. Using multiple classes, you specify that the content will take six slots on tablets and three on laptops.

```
<div class="row">
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
</div>
```

The result will show like this on tablets:



And like this on laptops and higher resolutions:



As an exercise, you can try and create rows with different number of columns depending on the screensize and verify the behaviour in your browser console.

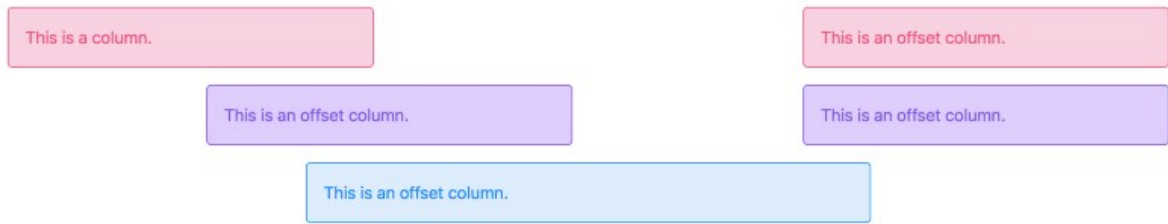
Offsetting Columns

If you don't want your columns to be next to each other, you can use the class `.offset-[breakpoint]-[size]` together with the `.col-[breakpoint]-[size]`. Using this class is the same as adding an empty column before your column. Here is a simple example:



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

You can use the class on any column in the row. Here are some more examples:



```

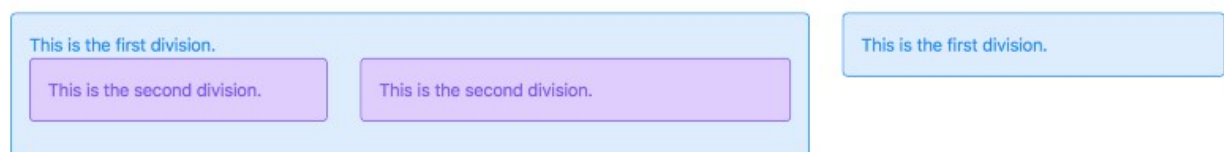
<div class="row">
  <div class="col-md-4">
    ...
  </div>
  <div class="col-md-4 offset-md-4">
    ...
  </div>
</div>
<div class="row">
  <div class="col-md-4 offset-md-2">
    ...
  </div>
  <div class="col-md-4 offset-md-2">
    ...
  </div>
</div>
<div class="row">
  <div class="col-md-6 offset-md-3">
    ...
  </div>
</div>

```

Nesting Columns

This may come as a surprise, but you can add a row inside a column!

The row in question (which will have the width of its parent column) will then be divided into 12 (smaller) columns that you can reference through the `.col-*` classes. Let's take a look at what happens when we insert a new row inside a column:



```

<div class="row">
  <div class="col-md-8">
    ...
    <div class="row">
      <div class="col-md-5">
        ...
      </div>
      <div class="col-md-7">
        ...
      </div>
    </div>
  </div>
</div>
<div class="col-md-4">
  ...
</div>
</div>

```

Knowing this, you can go many levels deep to organise your information. The columns will provide a simple way for you to manage your space.

This wraps up the basic knowledge regarding the Bootstrap responsive grid system. If you have questions, please let me know in the comments, I will be glad to answer.

Bootstrap 4 - Content

Description

Bootstrap 4 uses collection of content methods for displaying the text, blocks of code, responsive images, data in a tabular format etc on the web page.

The following table lists down the content methods which you can use to manipulate the Bootstrap 4 –

S.No.	Methods & Description
1	<u>Typography</u> The typography feature creates headings, paragraphs, lists and other inline elements.

2	<u>Code</u> It is used to display inline and multiline blocks of code in the document.
3	<u>Images</u> Bootstrap 4 provides support for images by using tag.
4	<u>Tables</u> Tables are used for displaying the data in a tabular format.
5	<u>Figures</u> The figure element specifies the content along with related images with an optional caption.

Bootstrap 4 - Tables

Description

Tables are used for displaying the data in a tabular format. If you want a basic table style with just some light padding and horizontal dividers, add the `.table` class to the <table> element. For more information on basic table and its elements, check out this [chapter](#).

Dark Table

You can use the `.table-dark` class to add a black background to the table as seen in the following example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>
```

```

<body>
  <div class = "container">
    <h2>Dark Table</h2>
    <table class = "table table-dark">
      <thead>
        <tr>
          <th>Player</th>
          <th>Country</th>
          <th>Ranking</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Virat Kohli</td>
          <td>India</td>
          <td>1</td>
        </tr>
        <tr>
          <td>Joe Root</td>
          <td>England</td>
          <td>2</td>
        </tr>
        <tr>
          <td>Steven Smith</td>
          <td>Australia</td>
          <td>3</td>
        </tr>
      </tbody>
    </table>
  </div>

  <!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
  </script>

  <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
  </script>

  <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stweULTy"
    crossorigin = "anonymous">
  </script>

</body>
</html>

```

It will produce the following result –

Output

Dark Table

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Table Head Options

You can add a black background or a grey background to table headers by using `.thead-dark` and `.thead-light` classes as seen in the following example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Dark Head</h2>
      <table class = "table">
        <thead class = "thead-dark">
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
          </tr>
          <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
          </tr>
          <tr>
            <td>Steven Smith</td>
```

```

        <td>Australia</td>
        <td>3</td>
    </tr>
</tbody>
</table>

<h2>Light Head</h2>
<table class = "table">
    <thead class = "thead-light">
        <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
        </tr>
        <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
        </tr>
        <tr>
            <td>Steven Smith</td>
            <td>Australia</td>
            <td>3</td>
        </tr>
    </tbody>
</table>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Dark Head

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Light Head

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Striped Rows

By adding the `.table-striped` class, you will get stripes on the rows as seen in the following example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Striped Rows</h2>
      <table class = "table table-striped">
        <thead>
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
          </tr>
```

```

        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
          </tr>
          <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
          </tr>
          <tr>
            <td>Steven Smith</td>
            <td>Australia</td>
            <td>3</td>
          </tr>
        </tbody>
      </table>
    </div>

    <!-- 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+8abtTElPi6jizo"
      crossorigin = "anonymous">
    </script>

    <script src =
    "https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
      integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
      crossorigin = "anonymous">
    </script>

    <script src =
    "https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
      integrity = "sha384-
ChfqquxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
      crossorigin = "anonymous">
    </script>

  </body>
</html>

```

It will produce the following result –

Output

Striped Rows

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Bordered and Borderless Table

Use `.table-bordered` class to make the borders around the table and cells. If you don't want to use border for a table, then use the `.table-borderless` class.

The following example demonstrates usage of above classes in the table –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Bordered Table</h2>
      <table class = "table table-bordered">
        <thead>
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
          </tr>
          <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
          </tr>
          <tr>
            <td>Steven Smith</td>
            <td>Australia</td>
            <td>3</td>
          </tr>
        </tbody>
      </table>

      <h2>Borderless Table</h2>
      <table class = "table table-borderless">
```

```

        <thead>
            <tr>
                <th>Player</th>
                <th>Country</th>
                <th>Ranking</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Virat Kohli</td>
                <td>India</td>
                <td>1</td>
            </tr>
            <tr>
                <td>Joe Root</td>
                <td>England</td>
                <td>2</td>
            </tr>
            <tr>
                <td>Steven Smith</td>
                <td>Australia</td>
                <td>3</td>
            </tr>
        </tbody>
    </table>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Bordered Table

Player	Country	Ranking
Virat Kohli	India	1

Player	Country	Ranking
Joe Root	England	2
Steven Smith	Australia	3

Borderless Table

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Hoverable Rows

By adding the `.table-hover` class, a light gray background color will be added to rows while the cursor hovers over them, as seen in the following example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Hoverable Rows</h2>
      <table class = "table table-hover">
        <thead>
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
```

```

        <td>India</td>
        <td>1</td>
    </tr>
    <tr>
        <td>Joe Root</td>
        <td>England</td>
        <td>2</td>
    </tr>
    <tr>
        <td>Steven Smith</td>
        <td>Australia</td>
        <td>3</td>
    </tr>
</tbody>
</table>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqgqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Hoverable Rows

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Small Table

You can use the `.table-sm` class to make the small table by cutting cell padding in half as shown in the below example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Small Table</h2>
      <table class = "table table-sm">
        <thead>
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
          </tr>
          <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
          </tr>
          <tr>
            <td>Steven Smith</td>
            <td>Australia</td>
            <td>3</td>
          </tr>
        </tbody>
      </table>
    </div>
```

```

<!-- 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+8abtTElPi6jizo"
  crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqxquZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Small Table

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3

Contextual Classes

You can use contextual classes to apply color to table rows or table cells as seen in the following example –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"

```

```
crossorigin = "anonymous">

<title>Bootstrap 4 Example</title>
</head>

<body>
  <div class = "container">
    <h2>Contextual Classes</h2>
    <table class = "table">
      <thead>
        <tr>
          <th>Player</th>
          <th>Country</th>
          <th>Ranking</th>
        </tr>
      </thead>
      <tbody>
        <tr class = "table-active">
          <td>Virat Kohli</td>
          <td>India</td>
          <td>1</td>
        </tr>
        <tr class = "table-success">
          <td>Joe Root</td>
          <td>England</td>
          <td>2</td>
        </tr>
        <tr class = "table-danger">
          <td>Steven Smith</td>
          <td>Australia</td>
          <td>3</td>
        </tr>
        <tr class = "table-info">
          <td>Kane Williamson</td>
          <td>New Zealand</td>
          <td>4</td>
        </tr>
        <tr class = "table-warning">
          <td>Dinesh Chandimal</td>
          <td>Srilanka</td>
          <td>5</td>
        </tr>
        <tr class = "table-secondary">
          <td>Alastair Cook</td>
          <td>England</td>
          <td>6</td>
        </tr>
        <tr class = "table-light">
          <td>Kraigg Brathwaite</td>
          <td>West Indies</td>
          <td>7</td>
        </tr>
        <tr class = "table-primary">
          <td>Cheteshwar Pujara</td>
          <td>India</td>
          <td>9</td>
        </tr>
        <tr class = "table-dark">
          <td>Hashim Amla</td>
          <td>South Africa</td>
          <td>8</td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>
```

```

        </tr>
    </tbody>
</table>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Contextual Classes

Player	Country	Ranking
Virat Kohli	India	1
Joe Root	England	2
Steven Smith	Australia	3
Kane Williamson	New Zealand	4
Dinesh Chandimal	Srilanka	5
Alastair Cook	England	6
Kraigg Brathwaite	West Indies	7
Cheteshwar Pujara	India	9
Hashim Amla	South Africa	8

Responsive Tables

By wrapping any `.table` in `.table-responsive` class, you will make the table scroll horizontally up to small devices (under 992px). When viewing on anything larger than 992px wide, you will not see any difference in these tables.

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Responsive Tables</h2>
      <table class = "table">
        <thead>
          <tr>
            <th>Player</th>
            <th>Country</th>
            <th>Ranking</th>
            <th>Runs</th>
            <th>Highest Score</th>
            <th>Avg</th>
            <th>Strike Rate</th>
            <th>Hundreds</th>
            <th>Fifties</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Virat Kohli</td>
            <td>India</td>
            <td>1</td>
            <td>9774</td>
            <td>183</td>
            <td>53.92</td>
            <td>60.75</td>
            <td>35</td>
            <td>48</td>
          </tr>
          <tr>
            <td>Joe Root</td>
            <td>England</td>
            <td>2</td>
            <td>4800</td>
            <td>133</td>
```

```

        <td>51.61</td>
        <td>55.20</td>
        <td>13</td>
        <td>28</td>
    </tr>
    <tr>
        <td>Steven Smith</td>
        <td>Australia</td>
        <td>3</td>
        <td>3431</td>
        <td>164</td>
        <td>41.84</td>
        <td>86.36</td>
        <td>10</td>
        <td>19</td>
    </tr>
</tbody>
</table>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Responsive Tables

	Player	Country	Ranking	Runs	Highest Score	Avg	Strike Rate	Hundreds	Fifties
	Virat Kohli	India	1	9774	183	53.92	60.75	35	48
	Joe Root	England	2	4800	133	51.61	55.20	13	28
	Steven Smith	Australia	3	3431	164	41.84	86.36	10	19

Bootstrap 4 - Utilities

Description

Bootstrap 4 uses collection of utilities for displaying borders, text color, embedding video etc on the web page.

The following table lists down the utilities types which you can use to manipulate the Bootstrap 4 –

S.No.	Methods & Description
1	<u>Borders</u> Border utility provides style, color and radius of an element's border.
2	<u>Clearfix and Close Icon</u> Clearfix is used to clear the floated content and close icon for dismissing the content.
3	<u>Colors</u> Use the contextual classes to change the color of text, link and background color of an element.
4	<u>Embed</u> It is used to embed the video in a page by using <iframe> element.
5	<u>Float</u> It is used to float an element to left or right side.
6	<u>Shadows and Spacing</u> Shadow utility adds shadow to the elements and spacing utility provides margin or padding values to an element.
7	<u>Sizing</u> You can make the size of an element wide or tall by using width and height utilities.
8	<u>Text</u> Bootstrap provides text utilities to control text alignment, transform, weight and more.

9

Flex

Flex utility can be used to manage the layout, alignment, grid columns, navigation and other components of the page.

Bootstrap 4 - Colors

Description

Use the contextual classes to change the color of text, link (using classes like text-primary, text-secondary etc) and background color (using classes like bg-primary, bg-secondary etc) of an element.

Text Color

The following example shows changing color of the *text* by using contextual classes –

Example

```
<html lang = "en">
  <head>
    <!-- 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://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  >
    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container-fluid">
      <h2>Text Color</h2>
      <p class = "text-primary">Color : text-primary </p>
      <p class = "text-secondary">Color : text-secondary</p>
      <p class = "text-success">Color : text-success</p>
      <p class = "text-danger">Color : text-danger</p>
      <p class = "text-warning">Color : text-warning</p>
      <p class = "text-info">Color : text-info</p>
      <p class = "text-black-50">Color : text-black-50</p>
      <p class = "text-dark">Color : text-dark</p>
      <p class = "text-body">Color : text-body</p>
```

```

        <p class = "text-muted">Color : text-muted</p>
        <p class = "text-white bg-dark">Color : text-white</p>
        <p class = "text-light bg-dark">Color : text-light</p>
        <p class = "text-white-50 bg-dark">Color : text-white-50</p>
    </div>

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src = "https://code.jquery.com/jquery-3.2.1.slim.min.js"
        integrity = "sha384-
KJ3o2DKtIkVYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
        crossorigin = "anonymous">
    </script>

    <!-- Popper -->
    <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
        integrity = "sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin = "anonymous">
    </script>

    <!-- Latest compiled and minified Bootstrap JavaScript -->
    <script src =
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
        integrity = "sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Text Color

Color : text-primary

Color : text-secondary

Color : text-success

Color : text-danger

Color : text-warning

Color : text-info

Color : text-black-50

Color : text-dark

Color : text-body

Color : text-muted

Color : text-white

Color : text-light

Color : text-white-50

Link Color

The following example shows changing color of the *link* by using contextual classes –

Example

```
<html lang = "en">
  <head>
    <!-- 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://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script
>

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container-fluid">
      <h2>Link Color</h2>
      <p><a href = "#" class = "text-primary">Primary link</a></p>
      <p><a href = "#" class = "text-secondary">Secondary link</a></p>
      <p><a href = "#" class = "text-success">Success link</a></p>
      <p><a href = "#" class = "text-danger">Danger link</a></p>
      <p><a href = "#" class = "text-warning">Warning link</a></p>
      <p><a href = "#" class = "text-info">Info link</a></p>
      <p><a href = "#" class = "text-dark">Dark link</a></p>
      <p><a href = "#" class = "text-muted">Muted link</a></p>
      <p><a href = "#" class = "text-light bg-dark">Light link</a></p>
      <p><a href = "#" class = "text-white bg-dark">White link</a></p>
    </div>

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src = "https://code.jquery.com/jquery-3.2.1.slim.min.js"
      integrity = "sha384-
KJ3o2DKtIkvYIK3UENzmM7KcKrr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
      crossorigin = "anonymous">
    </script>

    <!-- Popper -->
```

```
<script src =  
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"  
  integrity = "sha384-  
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"  
  crossorigin = "anonymous">  
</script>  
  
<!-- Latest compiled and minified Bootstrap JavaScript -->  
<script src =  
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"  
  integrity = "sha384-  
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpilMquVdAyjUar5+76PVCmYl"  
  crossorigin = "anonymous">  
</script>  
  
</body>  
</html>
```

It will produce the following result –

Output

Link Color

[Primary link](#)

[Secondary link](#)

[Success link](#)

[Danger link](#)

[Warning link](#)

[Info link](#)

[Dark link](#)

[Muted link](#)

[Light link](#)

[White link](#)

Background Color

The following example shows the changing *background color* of an element by using contextual classes –

Example

```
<html lang = "en">
  <head>
    <!-- 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://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script
>

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container-fluid">
      <h2>Background Color</h2>
      <div class = "p-3 mb-2 bg-primary text-white">class = 'bg-
primary'</div>
      <div class = "p-3 mb-2 bg-secondary text-white">class = 'bg-
secondary'</div>
      <div class = "p-3 mb-2 bg-success text-white">class = 'bg-
success'</div>
      <div class = "p-3 mb-2 bg-danger text-white">class = 'bg-
danger'</div>
      <div class = "p-3 mb-2 bg-warning text-dark">class = 'bg-
warning'</div>
      <div class = "p-3 mb-2 bg-info text-white">class = 'bg-info'</div>
      <div class = "p-3 mb-2 bg-light text-dark">class = 'bg-
light'</div>
      <div class = "p-3 mb-2 bg-dark text-white">class = 'bg-dark'</div>
      <div class = "p-3 mb-2 bg-white text-dark">class = 'bg-
white'</div>
      <div class = "p-3 mb-2 bg-transparent text-dark">class = 'bg-
transparent'</div>
    </div>

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src = "https://code.jquery.com/jquery-3.2.1.slim.min.js"
      integrity = "sha384-
KJ3o2DKtIkVYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
      crossorigin = "anonymous">
    </script>

    <!-- Popper -->
    <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
```

```

        integrity = "sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin = "anonymous">
    </script>

    <!-- Latest compiled and minified Bootstrap JavaScript -->
    <script src =
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
        integrity = "sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpilMquVdAyjUar5+76PVCmYl"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Background Color

```

class = 'bg-primary'
class = 'bg-secondary'
class = 'bg-success'
class = 'bg-danger'
class = 'bg-warning'
class = 'bg-info'
class = 'bg-light'
class = 'bg-dark'
class = 'bg-white'
class = 'bg-transparent'

```

Bootstrap 4 - Alerts

Description

The alert component specifies the predefined message for an user actions. It is used to send the information such as warning, error or confirmation messages to the end users.

You can create an alert box, by adding a class of *.alert* and along with contextual classes such as *.alert-success*, *.alert-info*, *.alert-warning*, *.alert-danger*, *.alert-primary*, *.alert-secondary*, *.alert-light* or *.alert-dark*.

The following example demonstrates usage of above contextual classes –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Alerts</h2>
      <div class = "alert alert-primary" role = "alert">
        primary alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-secondary" role = "alert">
        secondary alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-success" role = "alert">
        success alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-danger" role = "alert">
        danger alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-warning" role = "alert">
        warning alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-info" role = "alert">
        info alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-light" role = "alert">
        light alert - Welcome to Tutorialspoint!!!
      </div>

      <div class = "alert alert-dark" role = "alert">
        dark alert - Welcome to Tutorialspoint!!!
      </div>
    </div>

    <!-- 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+8abtTElPi6jizo"
      crossorigin = "anonymous">
    </script>

    <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
      integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqBjBiSnjAK/l8WvCWPIpM49"

```



```

        crossorigin = "anonymous">
    </script>

    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
        integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Alerts

```

primary alert - Welcome to Tutorialspoint!!!
secondary alert — Welcome to Tutorialspoint!!!
success alert — Welcome to Tutorialspoint!!!
danger alert — Welcome to Tutorialspoint!!!
warning alert — Welcome to Tutorialspoint!!!
info alert — Welcome to Tutorialspoint!!!
light alert — Welcome to Tutorialspoint!!!
dark alert — Welcome to Tutorialspoint!!!

```

Links in Alerts

To get links in alerts, use the `.alert-link` utility class in the `<a>` tag as shown in the below example –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>

```

```

<div class = "container">
  <h2>Links in Alerts</h2>
  <div class = "alert alert-primary" role = "alert">
    primary alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-secondary" role = "alert">
    secondary alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-success" role = "alert">
    success alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-danger" role = "alert">
    danger alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-warning" role = "alert">
    warning alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-info" role = "alert">
    info alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-light" role = "alert">
    light alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>

  <div class = "alert alert-dark" role = "alert">
    dark alert - Welcome to
    <a href = "https://www.tutorialspoint.com/" target = "_blank"
      rel = "nofollow" class = "alert-link">Tutorialspoint!!!</a>
  </div>
</div>

<!-- 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+8abtTElPi6jizo"
  crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"

```

```

        integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
        crossorigin = "anonymous">
    </script>

    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
        integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Links in Alerts

primary alert - Welcome to **Tutorialspoint!!!**

secondary alert — Welcome to **Tutorialspoint!!!**

success alert — Welcome to **Tutorialspoint!!!**

danger alert — Welcome to **Tutorialspoint!!!**

warning alert — Welcome to **Tutorialspoint!!!**

info alert — Welcome to **Tutorialspoint!!!**

light alert — Welcome to **Tutorialspoint!!!**

dark alert — Welcome to **Tutorialspoint!!!**

Dismissal Alerts

To build a dismissal alert, use the `.alert-dismissible` class to alert container. Add the `data-dismiss="alert"` attribute on the button element to close a button, which automatically dismisses the alert message box.

The following example demonstrates dismissal of alert box –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

<title>Bootstrap 4 Example</title>
</head>
<body>
  <div class = "container">
    <h2>Dismissal Alerts</h2>
    <div class = "alert alert-success alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Success!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-primary alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Primary!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-secondary alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Secondary!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-danger alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Danger!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-warning alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Warning!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-info alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Info!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-light alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Light!</strong> Welcome to Tutorialspoint!!!
    </div>

    <div class = "alert alert-dark alert-dismissible">
      <button type = "button" class = "close" data-dismiss =
"alert">×</button>
      <strong>Dark!</strong> Welcome to Tutorialspoint!!!
    </div>
  </div>

```

```

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Dismissal Alerts

```

×Success! Welcome to Tutorialspoint!!!
×Primary! Welcome to Tutorialspoint!!!
×Secondary! Welcome to Tutorialspoint!!!
×Danger! Welcome to Tutorialspoint!!!
×Warning! Welcome to Tutorialspoint!!!
×Info! Welcome to Tutorialspoint!!!
×Light! Welcome to Tutorialspoint!!!
×Dark! Welcome to Tutorialspoint!!!

```

Bootstrap 4 - Forms

Description

The form element is used to collect input from user by using fields such as checkboxes, radio buttons, or text fields etc.

Basic Form

You can wrap labels and controls in a <div> element with class `.form-group` and add a class of `.form-control` to all textual <input>, <textarea>, and <select> elements.

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Basic Form</h2>
      <form>
        <div class = "form-group">
          <label for = "exampleInputEmail1">Email</label>
          <input type = "email" class = "form-control"
            id = "exampleInputEmail1" aria-describedby = "emailHelp"
            placeholder = "Enter your email">
        </div>

        <div class = "form-group">
          <label for = "exampleInputPassword1">Password</label>
          <input type = "password" class = "form-control"
            id = "exampleInputPassword1" placeholder = "Enter your
password">
        </div>

        <div class = "form-group form-check">
          <label class = "form-check-label" for = "exampleCheck1">
            <input type = "checkbox" class = "form-check-input"
              id = "exampleCheck1">Remember me
          </label>
        </div>
        <button type = "submit" class = "btn btn-primary">Sign
In</button>
      </form>
    </div>

    <!-- 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/1.14.3/umd/popper.min.js"
      integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
      crossorigin = "anonymous">
    </script>

```

```

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqgquxZUCnJSK3+MXmPNiyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Basic Form

Email

Password



Remember me

Sign in

Form Controls

Bootstrap natively supports the most common form controls such as *input*, *textarea* and *select*.

The following example demonstrates above supported form controls specified with *.form-control* class –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">

```

```

    <form>
      <h4>Example Input</h4>
      <div class = "form-group">
        <input type = "email" class = "form-control"
          id = "exampleFormControlInput1" placeholder = "Enter your
email">
      </div>

      <h4>Example Select</h4>
      <div class = "form-group">
        <select class = "form-control" id =
"exampleFormControlSelect1">
          <option>Select Option #1</option>
          <option>Select Option #2</option>
          <option>Select Option #3</option>
          <option>Select Option #4</option>
          <option>Select Option #5</option>
        </select>
      </div>

      <h4>Example Multiple Select</h4>
      <div class = "form-group">
        <select multiple class = "form-control" id =
"exampleFormControlSelect2">
          <option>Multiple Select #1</option>
          <option>Multiple Select #2</option>
          <option>Multiple Select #3</option>
          <option>Multiple Select #4</option>
          <option>Multiple Select #5</option>
        </select>
      </div>

      <h4>Example Textarea</h4>
      <div class = "form-group">
        <textarea class = "form-control" id =
"exampleFormControlTextarea1" rows = "3"></textarea>
      </div>
    </form>
  </div>

  <!-- 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/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
  </script>

  <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stweULTy"
    crossorigin = "anonymous">
  </script>

```



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

It will produce the following result –

Output

Example Input

Example Select

Example Multiple Select

Example Textarea

Sizing, Readonly and Range Inputs

The input field can be displayed in large and small sizes by using `.form-control-lg` and `.form-control-sm` classes respectively. The `readonly` attribute is a boolean attribute, which makes input field as readonly and cannot be modified. You can put range for the inputs by using the `.form-control-range` class.

The following example demonstrates above types –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <form>
        <h2>Sizing</h2>
        <input class = "form-control form-control-lg" type = "text"
          placeholder = "Large Input">
```

```

<br>

<input class = "form-control" type = "text"
  placeholder = "Default Input">
<br>

<input class = "form-control form-control-sm" type = "text"
  placeholder = "Small Input">
<br>
<br>

<h2>Readonly</h2>
<input class = "form-control" type = "text"
  placeholder = "This is readonly text" readonly>
<br>
<br>

<h2>Range Inputs</h2>
<div class = "form-group">
  <input type = "range" class = "form-control-range"
    id = "formControlRange">
</div>
</form>
</div>

<!-- 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+8abtTElPi6jizo"
  crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqqqxUzUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Sizing

Readonly

Range Inputs

Form grid using Form Row

You can create complex forms which require multiple columns by using `.form-row` class that specify the compact layout for column. The following example demonstrates this –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <form>
        <h2>Form Row</h2>
        <div class = "form-row">
          <div class = "form-group col-md-6">
            <label for = "inputEmail4">First Name</label>
            <input type = "text" class = "form-control"
              id = "inputEmail4" placeholder = "First Name">
          </div>

          <div class = "form-group col-md-6">
            <label for = "inputPassword4">Last Name</label>
            <input type = "text" class = "form-control"
              id = "inputPassword4" placeholder = "Last Name">
          </div>
        </div>

        <div class = "form-group">
          <label for = "inputAddress">Address</label>
```

```

        <input type = "text" class = "form-control" id =


```

```

        integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Form Row

First Name

Last Name

Address

City

State

Pin Code

☐ I Agree To Terms and Conditions

Horizontal Forms

Create horizontal forms by adding the `.row` class to form groups. The width of labels and controls can be specified by using `.col-*` classes and add the `.col-form-label` class to your `<label>`, so that you can place form controls vertically centered.

The following example demonstrates this –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">

```

```

    <form>
      <h2>Horizontal Forms</h2>
      <div class = "form-group row">
        <label for = "inputEmail3" class = "col-sm-2 col-form-
label">First Name</label>
        <div class = "col-sm-10">
          <input type = "text" class = "form-control" id =
"inputEmail3"
            placeholder = "First Name">
        </div>
      </div>

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

      <fieldset class = "form-group">
        <div class = "row">
          <legend class = "col-form-label col-sm-2 pt-
0">Radios</legend>
          <div class = "col-sm-10">
            <div class = "form-check">
              <input class = "form-check-input" type = "radio"
                name = "gridRadios" id = "gridRadios1" value =
"option1" checked>
              <label class = "form-check-label" for =
"gridRadios1">
                Option 1
              </label>
            </div>

            <div class = "form-check">
              <input class = "form-check-input" type = "radio"
                name = "gridRadios" id = "gridRadios2" value =
"option2">
              <label class = "form-check-label" for =
"gridRadios2">
                Option 2
              </label>
            </div>

            <div class = "form-check disabled">
              <input class = "form-check-input" type = "radio"
                name = "gridRadios" id = "gridRadios3" value =
option3" disabled>
              <label class = "form-check-label" for =
"gridRadios3">
                Option 3 (disabled)
              </label>
            </div>
          </div>
        </div>
      </fieldset>

      <div class = "form-group row">

```

```

        <div class = "col-sm-2">Checkbox</div>
        <div class = "col-sm-10">
            <div class = "form-check">
                <input class = "form-check-input" type = "checkbox"
                    id = "gridCheck1">
                <label class = "form-check-label" for = "gridCheck1">
                    Option 1
                </label>
            </div>

            <div class = "form-check">
                <input class = "form-check-input" type = "checkbox" id
= "gridCheck2">
                <label class = "form-check-label" for = "gridCheck1">
                    Option 2
                </label>
            </div>
        </div>
    </div>

    <div class = "form-group row">
        <div class = "col-sm-10">
            <button type = "submit" class = "btn btn-
primary">Submit</button>
        </div>
    </div>

</form>
</div>

<!-- 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+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqqxUzUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Horizontal Forms

First Name

Last Name

Radios

- ☒ Option 1
- ☐ Option 2
- ☐ Option 3 (disabled)

Checkbox

- ☐ Option 1
- ☐ Option 2

Inline Form

You can create a form where all of the elements are inline, left aligned and labels are alongside, by adding the class `.form-inline` to the `<form>` tag.

The following example demonstrates displaying of form controls on a single horizontal row

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Inline Forms</h2>
      <form class = "form-inline" action = "/action_page.php">
        <label for = "email">Email :
        <input type = "email" class = "form-control" id = "email"
          placeholder = "Enter email" name = "email"></label>

        <label for = "pwd">Password :
        <input type = "password" class = "form-control" id = "pwd"
          placeholder = "Enter password" name = "pswd"></label>
      </form>
    </div>
  </body>
</html>
```



```

        <button type = "submit" class = "btn btn-primary">Sign
In</button>
    </form>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Inline Forms

Email : Password :

Disabled Form and Help Text

Use the *disabled* attribute to disable the form controls (blocks the user interactions on an input). You can enter text to the related fields by using *.form-text* class. The following example demonstrates this –

Example

```

<html lang = "en">
<head>
    <!-- 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/4.1.3/css/bootstrap.min.css"

```

```

        integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
        crossorigin = "anonymous">

<title>Bootstrap 4 Example</title>
</head>

<body>
  <div class = "container">
    <h2>Disabled Form</h2>
    <form>
      <fieldset disabled>
        <div class = "form-group">
          <label for = "disabledTextInput">Name</label>
          <input type = "text" id = "disabledTextInput" class =
"form-control" placeholder = "Name">
        </div>

        <div class = "form-group">
          <label for = "disabledSelect">Select your option</label>
          <select id = "disabledSelect" class = "form-control">
            <option>Select</option>
          </select>
        </div>

        <div class = "form-check">
          <input class = "form-check-input" type = "checkbox"
            id = "disabledFieldsetCheck" disabled>
          <label class = "form-check-label"
for="disabledFieldsetCheck">
            Remember Me
          </label>
        </div>

        <button type = "submit" class = " btn btn-
primary">Submit</button>
        <br>
        <br>

        <h2>Help Text</h2>
        <label for = "inputPassword5">Password</label>
        <input type = "password" id = "inputPassword5" class =
"form-control"
          aria-describedby = "passwordHelpBlock">

        <small id = "passwordHelpBlock" class = "form-text text-
muted">
          Your password must be 6-10 characters long (This is help
text).
        </small>
      </fieldset>
    </form>
  </div>

  <!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
  </script>

```

```

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNiyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Disabled Form

Name

Select your option

☐ Remember Me

Help Text

Password Your password must be 6-10 characters long (This is help text).

Bootstrap 4 - Buttons

Description

Bootstrap provides clickable button to put content such as text and images. You can include the text to button by using the `<input>` tag.

You can create button by using `.btn` class followed by desired style (e.g. `btn-success`). Bootstrap provides some options to style buttons, which are listed below –

- **btn-primary**
- **btn-secondary**
- **btn-success**
- **btn-danger**

- **btn-warning**
- **btn-info**
- **btn-light**
- **btn-dark**
- **btn-link**

The following example demonstrates all the above button classes –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class="container">
      <h2>Button Styles</h2>
      <button type = "button" class = "btn btn-primary">Primary
Button</button>
      <button type = "button" class = "btn btn-secondary">Secondary
Button</button>
      <button type = "button" class = "btn btn-success">Success
Button</button>
      <button type = "button" class = "btn btn-danger">Danger
Button</button>
      <button type = "button" class = "btn btn-warning">Warning
Button</button>
      <button type = "button" class = "btn btn-info">Info
Button</button>
      <button type = "button" class = "btn btn-light">Light
Button</button><br><br>
      <button type = "button" class = "btn btn-dark">Dark
Button</button>
      <button type = "button" class = "btn btn-link">Link
Button</button>
    </div>

    <!-- 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+8abtTElPi6jizo"
      crossorigin = "anonymous">
    </script>
```

```

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin = "anonymous">
</script>

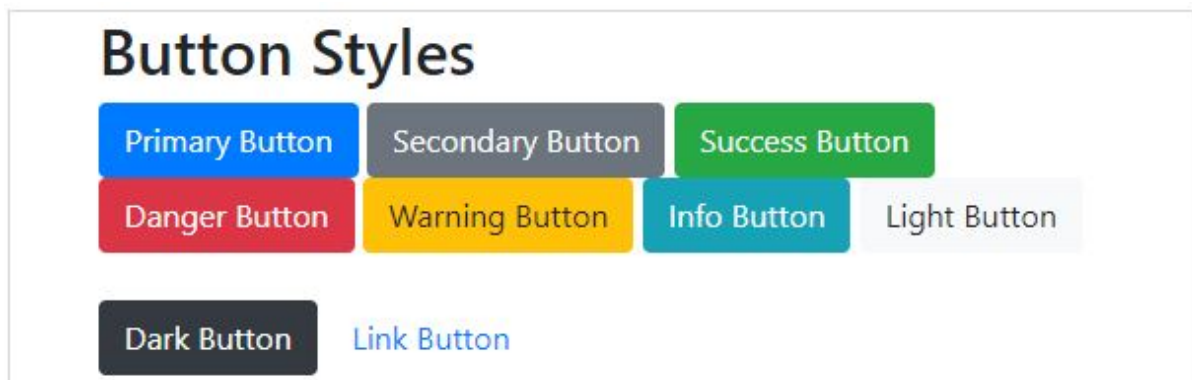
<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output



Outline Buttons

Bootstrap 4 provides semantic color to outline of a button by using `.btn-outline` class followed by desired style (e.g. `btn-outline-success`).

The following example demonstrates displaying of outlined buttons –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"

```

```

        crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
</head>

<body>
    <div class = "container">
        <h2>Button Outlines</h2>
        <button type = "button" class = "btn btn-outline-primary">Primary
Button</button>
        <button type = "button" class = "btn btn-outline-
secondary">Secondary Button</button>
        <button type = "button" class = "btn btn-outline-success">Success
Button</button>
        <button type = "button" class = "btn btn-outline-danger">Danger
Button</button>
        <button type = "button" class = "btn btn-outline-light">Light
Button</button>
        <button type = "button" class = "btn btn-outline-warning">Warning
Button</button>
        <button type = "button" class = "btn btn-outline-info">Info
Button</button>
        <button type = "button" class = "btn btn-outline-dark">Dark
Button</button>
    </div>

    <!-- 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+8abtTElPi6jizo"
        crossorigin = "anonymous">
    </script>

    <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
        integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
        crossorigin = "anonymous">
    </script>

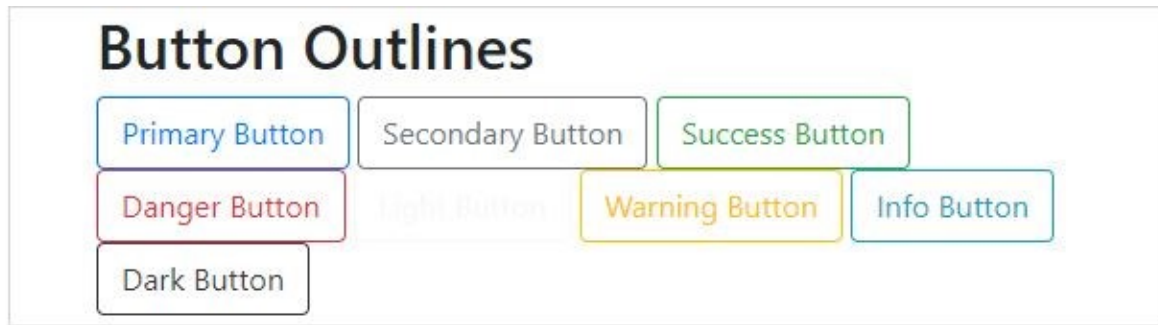
    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
        integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output



Button Sizes and Block Level Buttons

Bootstrap 4 provides smaller and larger buttons by using `.btn-sm` and `.btn-lg` classes and `.btn-block` class creates block level button that span the full width of a parent as shown in the example below –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Button Sizes</h2>
      <button type = "button" class = "btn btn-info btn-sm">Small
button</button>
      <button type = "button" class = "btn btn-info">Default
button</button>
      <button type = "button" class = "btn btn-info btn-lg">Large
button</button>
      </br>
      </br>

      <h2>Block Level Button</h2>
      <button type = "button" class = "btn btn-info btn-lg btn-block">
        Block Level Info Button
      </button>
    </div>
```

```

<!-- 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+8abtTElPi6jizo"
  crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin = "anonymous">
</script>

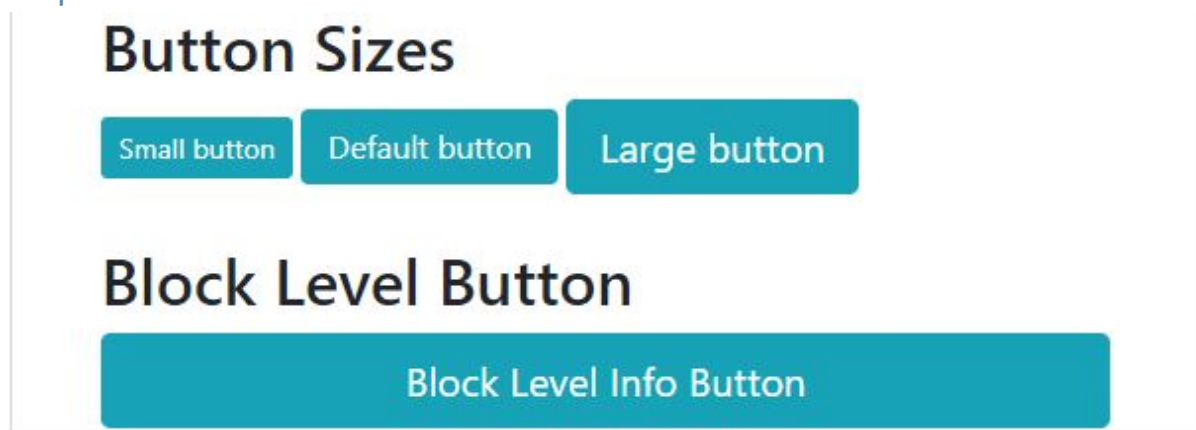
<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE24lrYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output



Active and Disabled State

Bootstrap allows you to change the state of buttons as *active* and *disabled*. Buttons will appear pressed when it's in active state and button will lose the gradient and fade in color by 50%, then we say it's in disabled state.

Use the `.active` class to show activated button and add the `disabled` attribute to make the button inactive. The following example demonstrates states of button –

Example

```

<html lang = "en">
<head>
  <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Active State</h2>
      <button type = "button" class = "btn btn-lg btn-primary" >
        Active button
      </button>
      <br>
      <br>

      <h2>Disabled State</h2>
      <button type = "button" class = "btn btn-primary btn-lg disabled">
        Disabled Button
      </button>
    </div>

    <!-- 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/1.14.3/umd/popper.min.js"
      integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
      crossorigin = "anonymous">
    </script>

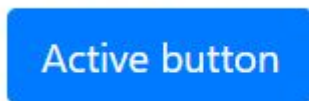
    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
      integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
      crossorigin = "anonymous">
    </script>

  </body>
</html>

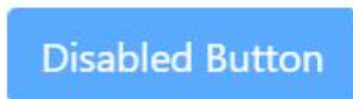
```

It will produce the following result –

Active State



Disabled State



Toggle Checkboxes and Radio Buttons

Bootstrap 4 allows applying button styles to radio buttons and checkboxes to make use of toggle feature. You can provide toggling of checkboxes and radio buttons inside a `<label>` element, by adding `data-toggle="buttons"` inside `.btn-group` element.

The following example demonstrates toggling of checkboxes and radio buttons –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>
  <body>
    <div class = "container">
      <h2>Checkbox</h2>
      <div class = "btn-group btn-group-toggle" data-toggle = "buttons">
        <label class = "btn btn-info active">
          <input type = "checkbox" name = "options" id = "option1"
autocomplete = "off" checked>
HTML-5
        </label>
        <label class = "btn btn-info">
          <input type = "checkbox" name = "options" id = "option2"
autocomplete = "off">
CSS-3
        </label>
      </div>
    </div>
  </body>
</html>
```

```

        <label class = "btn btn-info">
            <input type = "checkbox" name = "options" id = "option3"
autocomplete = "off">
            Bootstrap-4
        </label>
    </div>
    <br>
    <br>

    <h2>Radio Buttons</h2>
    <div class = "btn-group btn-group-toggle" data-toggle = "buttons">
        <label class = "btn btn-primary active">
            <input type = "radio" name = "options" id = "option1"
autocomplete = "off" checked>
            HTML-5
        </label>
        <label class = "btn btn-primary">
            <input type = "radio" name = "options" id = "option2"
autocomplete = "off">
            CSS-3
        </label>
        <label class = "btn btn-primary">
            <input type = "radio" name = "options" id = "option3"
autocomplete = "off">
            Bootstrap-4
        </label>
    </div>
</div>
<!-- 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/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqquxZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output



Bootstrap 4 - Button Group

Description

Button groups allow multiple buttons to be stacked together on a single line.

Basic Button Group

You can create the basic button group by using `.btn-group` class inside `<div>` element as shown in the below example –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Button Group</h2>
      <div class = "btn-group">
        <button type = "button" class = "btn btn-info">Cricket</button>
```

```

        <button type = "button" class = "btn btn-
info">Football</button>
        <button type = "button" class = "btn btn-info">Hockey</button>
    </div>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqqxUzUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Button Group

Cricket Football Hockey

Button Toolbar

You can group sets of button group into a single toolbar by using `.btn-toolbar` class as shown in the example below –

Example

```

<html lang = "en">
<head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
        integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
        crossorigin = "anonymous">

</head>

<body>
  <div class = "container">
    <h2>Button Toolbar</h2>
    <div class = "btn-toolbar" role = "toolbar" aria-label = "Toolbar
with button groups">
      <div class = "btn-group mr-2" role = "group" aria-label =
"First group">
        <button type = "button" class = "btn btn-
info">Cricket</button>
        <button type = "button" class = "btn btn-
info">Football</button>
        <button type = "button" class = "btn btn-
info">Hockey</button>
        <button type = "button" class = "btn btn-
info">Tennis</button>
      </div>

      <div class = "btn-group mr-2" role = "group" aria-label =
"Second group">
        <button type = "button" class = "btn btn-
info">Basketball</button>
        <button type = "button" class = "btn btn-
info">Volleyball</button>
        <button type = "button" class = "btn btn-
info">Boxing</button>
      </div>

      <div class = "btn-group mr-2" role = "group" aria-label =
"Third group">
        <button type = "button" class = "btn btn-
info">Archery</button>
        <button type = "button" class = "btn btn-
info">Badminton</button>
      </div>
    </div>
  </div>

  <!-- 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+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTElPi6jizo"
    crossorigin = "anonymous">
  </script>

  <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/18WvCWPIpM49"
    crossorigin = "anonymous">
  </script>

```

```

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity = "sha384-
ChfqgquxZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output



Button Group Sizes

You can control size of the button group by adding `.btn-group-sm` or `.btn-group-lg` class to the `.button-group` class as shown in the below example –

Example

```

<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Button Group Sizes</h2>
      <div class = "btn-group btn-group-lg" role = "group">
        <button type = "button" class = "btn btn-info">Cricket</button>
        <button type = "button" class = "btn btn-
info">Football</button>
        <button type = "button" class = "btn btn-info">Hockey</button>
      </div>
      <br>
      <br>
    </div>
  </body>
</html>

```

```

        <div class = "btn-group" role = "group">
            <button type = "button" class = "btn btn-info">Cricket</button>
            <button type = "button" class = "btn btn-
info">Football</button>
            <button type = "button" class = "btn btn-info">Hockey</button>
        </div>
        <br>
        <br>

        <div class = "btn-group btn-group-sm" role = "group">
            <button type = "button" class = "btn btn-info">Cricket</button>
            <button type = "button" class = "btn btn-
info">Football</button>
            <button type = "button" class = "btn btn-info">Hockey</button>
        </div>
    </div>

    <!-- 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+8abtTElPi6jizo"
        crossorigin = "anonymous">
    </script>

    <script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
        integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
        crossorigin = "anonymous">
    </script>

    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
        integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
        crossorigin = "anonymous">
    </script>

</body>
</html>

```

It will produce the following result –

Output

Button Group Sizes



Vertical Button Group

You can display the button group vertically by using the class `.btn-group-vertical` as shown in the example below –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Vertical Button Group</h2>
      <div class = "btn-group-vertical" role = "group">
        <button type = "button" class = "btn btn-info">Cricket</button>
        <button type = "button" class = "btn btn-
info">Football</button>
        <button type = "button" class = "btn btn-info">Hockey</button>
      </div>
    </div>

    <!-- 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/1.14.3/umd/popper.min.js"
      integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqBjJiSnjAK/l8WvCWPIpM49"
      crossorigin = "anonymous">
    </script>

    <script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
      integrity = "sha384-
ChfqqqxUzUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
      crossorigin = "anonymous">
    </script>

  </body>
```

```
</html>
```

It will produce the following result –

Output

Vertical Button Group



Nesting Button Group

You can place a `.btn-group` inside another `.btn-group`, when you want to display dropdown menu within a series of buttons.

The following example demonstrates nesting of button groups –

Example

```
<html lang = "en">
  <head>
    <!-- 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/4.1.3/css/bootstrap.min.css"
      integrity = "sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin = "anonymous">

    <title>Bootstrap 4 Example</title>
  </head>

  <body>
    <div class = "container">
      <h2>Nesting Button Group</h2>
      <div class = "btn-group" role = "group">
        <button type = "button" class = "btn btn-info">Cricket</button>
        <button type = "button" class = "btn btn-
info">Football</button>
        <button type = "button" class = "btn btn-info">Hockey</button>
        <div class = "btn-group" role = "group">
          <button id = "nestingbtngroup" type = "button" class = "btn
btn-info
          dropdown-toggle" data-toggle = "dropdown" aria-haspopup =
"true"
          aria-expanded = "false">
```

```

        Sports
    </button>

    <div class = "dropdown-menu" aria-labelledby =
"nestingbtngroup">
        <a class = "dropdown-item" href = "#">Cricket</a>
        <a class = "dropdown-item" href = "#">Football</a>
        <a class = "dropdown-item" href = "#">Hockey</a>
    </div>
</div>
</div>
</div>

<!-- 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+8abtTElPi6jizo"
    crossorigin = "anonymous">
</script>

<script src =
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
    integrity = "sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqBjBjSnjAK/l8WvCWPIpM49"
    crossorigin = "anonymous">
</script>

<script src =
"https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
    integrity = "sha384-
ChfqqxuZUCnJSK3+MXmPNiY6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
    crossorigin = "anonymous">
</script>

</body>
</html>

```

It will produce the following result –

Output

Nesting Button Group

Cricket Football Hockey Sports ▾