

Unit 6

Introduction to Bootstrap

What is Bootstrap?

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers – Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Why use Bootstrap?

By using this framework, we can easily manipulate the styling of any web page, like font style, text color, background color, flex, grid system, etc. Bootstrap Version 4 & Version 5 are the most popular versions. There are lots of other CSS frameworks like Tailwind CSS, Bulma, and Foundation but among them, this framework is the most popular because of below mentioned features:

- It is Faster and Easier way for Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It designs responsive web pages for mobile devices too.
- It is a free and open-source framework available on www.getbootstrap.com

Bootstrap setup

There are two ways to start using Bootstrap on your own web site.

You can:

- Download Bootstrap from getbootstrap.com
- Include Bootstrap from a CDN

Download Bootstrap from getbootstrap.com

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there and save that in your working folder.

<https://getbootstrap.com/docs/4.6/getting-started/download/>

Include Bootstrap from a CDN

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

```
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
```

jsDelivr provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery:

- **jQuery library, JavaScript library and Latest compiled JavaScript library**

```
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script>
```

Jquery and popper

Bootstrap 4 uses jQuery and popper.js for javascript components (like models,tooltip,popovers etc.).

However, if you just use the css part of bootstrap, you don't need them.

JQuery:

- it is a javascript library.
- It greatly simplifies javascript programming.

Components that require jQuery:

- Closable alerts
- Buttons and checkboxes/radio buttons for toggling states
- Carousel for slides, controls, indicators
- Collapse for toggling content
- Dropdown
- Modals
- Navbar
- Tooltip and popovers
- Scrollspy for scroll behavior and navigation updates

Containers: Fixed, Fluid

Containers are used to pad the content inside of them, and there are two container classes available:

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



Fixed Container

Use the **.container** class to create a responsive, fixed-width container. In the below example, the div with class “container” will have a fixed left and right margin and will not take the complete width of its parent or the viewport.

	Extra small (xs) <576px	Small (sm) ≥576px	Medium (md) ≥768px	Large (lg) ≥992px	Extra large (xl) ≥1200px
max-width	100%	540px	720px	960px	1140px

Container Padding/Margin

By default, containers have 15px left and right padding, with no top or bottom padding. Therefore, we often use **spacing utilities**, such as extra padding and margins to make them look even better. For example, **.pt-3** means "add a top padding of 16px":

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

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

The classes are used in the format:

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

1. **m** - sets margin
2. **p** - sets padding
3. **t** - sets margin-top or padding-top

-
4. **b** - sets margin-bottom or padding-bottom
 5. **l** - sets margin-left or padding-left
 6. **r** - sets margin-right or padding-right
 7. **x** - sets both padding-left and padding-right or margin-left and margin-right
 8. **y** - sets both padding-top and padding-bottom or margin-top and margin-bottom
 9. **blank** - sets a margin or padding on all 4 sides of the element
-

- | | |
|--|--|
| | ✓ 0 - sets margin or padding to 0 |
| | ✓ 1 - sets margin or padding to .25rem (4px if font-size is 16px) |
| | ✓ 2 - sets margin or padding to .5rem (8px if font-size is 16px) |
| | ✓ 3 - sets margin or padding to 1rem (16px if font-size is 16px) |
| | ✓ 4 - sets margin or padding to 1.5rem (24px if font-size is 16px) |
| | ✓ 5 - sets margin or padding to 3rem (48px if font-size is 16px) |
| | ✓ auto - sets margin to auto |

Examples:

- | | |
|--|---|
| | ✓ <code>mx-3</code> – 16 px margin from right and left |
| | ✓ <code>pt-2</code> – 8px padding from top |
| | ✓ <code>p-4</code> – 24px padding to all sides. |
| | ✓ <code>mx-auto</code> - <code>.mx-auto</code> class for horizontally centering fixed-width block level content |

Example:

```
<html>
<head>
<title>Bootstrap Example</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container">
<h1>My fixed container Bootstrap Page</h1>
<p>Chapter-6 Introduction to Bootstrap</p>
</div>
</body></html>
```

Output:

My fixed container Bootstrap Page

Chapter-6 Introduction to Bootstrap

Fluid Container

The .container-fluid class provides a full-width container which spans the entire width of the viewport. In the below example, the div with class “container-fluid” will take up the complete width of the viewport and will expand or shrink whenever the viewport is resized.

Example:

```
<html> <head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script> </head>
<body>
<div class="container">
<h1>My fixed container Bootstrap Page</h1>
<p>Chapter-6 Introduction to Bootstrap</p>
</div>
<div class="container-fluid">
<h1>My Fluid container Bootstrap Page</h1>
<p>This part is inside a .container-fluid class.</p>
</div></body> </html>
```

Output:

My fixed container Bootstrap Page

Chapter-6 Introduction to Bootstrap

My Fluid container Bootstrap Page

This part is inside a .container-fluid class.

Bootstrap Typography

Heading tag in Bootstrap 4.0

- <h1>h1 Bootstrap heading (2.5rem = 40px)</h1>
- <h2>h2 Bootstrap heading (2rem = 32px)</h2>
- <h3>h3 Bootstrap heading (1.75rem = 28px)</h3>
- <h4>h4 Bootstrap heading (1.5rem = 24px)</h4>
- <h5>h5 Bootstrap heading (1.25rem = 20px)</h5>
- <h6>h6 Bootstrap heading (1rem = 16px)</h6>

Display Headings

It is an art of specifying text and its formatting. Default font-size:16px. In addition, all <p> elements have margin-top: 0 and margin-bottom: 1rem (16px by default).

1 rem = 16 px

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

.display-1, .display-2, .display-3, .display-4

Example:

```
<html><head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script></head>
<body>
<div class="container">
<h1>Display Headings</h1>
<p>Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight):</p>
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
</div></body></html>
```

Output:**Display Headings**

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight):

Display 1

Display 2

Display 3

Display 4

Abbreviation

<abbr> - This is to denote abbreviation with its full form. Stylized implementation of HTML's <abbr> element for abbreviations and acronyms to show the expanded version on hover. Abbreviations have a default underline and gain a help cursor to provide additional context on hover and to users of assistive technologies.

Use the global `title` attribute to show the description for the abbreviation/acronym when you mouse over the element.

*Ref: `.initialism` class- Displays the text inside an <abbr> element in a slightly smaller font size.

Example

```
<body>
<div class="container">
<h1>Abbreviations</h1>
<p>The abbr element is used to mark up an abbreviation or acronym:</p>
<abbr title="Full Stack Development" class="initialism">FSD</abbr>
</div>
```

Output:**Abbreviations**

The abbr element is used to mark up an abbreviation or acronym:

FSD

Full Stack Development

<blockquote>

Add the .blockquote class to a <blockquote> when quoting blocks of content from another source.

Example

```
<body>
<div>
<blockquote class="blockquote">
<p>YOU WILL HAVE TO BE THE CHANGE, THAT YOU WANT
TO SEE IN THE WORLD....</p>
<footer class="blockquote-footer text-right">
MAHATMA GANDHI
</footer>
</blockquote>
</div>
</body>
```

Output:

YOU WILL HAVE TO BE THE CHANGE, THAT YOU WANT TO SEE IN THE WORLD....

— MAHATMA GANDHI

Mark

Bootstrap will style the HTML <mark> element with a yellow background color and some padding:

```
<div class="container">
<h1>Highlight Text</h1>
<p>Use the mark element to <mark>highlight</mark> text.</p></div>
```

Highlight Text

Use the mark element to **highlight** text.

Some other typography classes

Class	Description
.font-weight-bold	Bold text
.font-weight-bolder	Bolder text
.font-italic	Italic text
.font-weight-light	Light weight text
.font-weight-lighter	Lighter weight text
.font-weight-normal	Normal text
.small	Indicates smaller text (set to 80% of the size of the parent)
.text-left	Indicates left-aligned text
.text-*-left *-xs/sm/md/lg/xl	Indicates left-aligned text on small, medium, large or xlARGE screens
.text-center	Indicates center-aligned text
.text-*center	Indicates center-aligned text on small, medium, large or xlARGE screens
.text-decoration-none	Removes the underline from a link
.text-right	Indicates right-aligned text
.text-*right	Indicates right-aligned text on small, medium, large or xlARGE screens
.text-justify	Indicates justified text
.text-lowercase	Indicates lowercased text
.text-uppercase	Indicates uppcased text
.text-capitalize	Indicates capitalized text
.initialism	Displays the text inside an <abbr> element in a slightly smaller font size
.list-unstyled	Removes the default list-style and left margin on list items (works on both and). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)
.list-inline	Places all list items on a single line (used together with .list-inline-item on each elements)

Example on Typography class

```
<html><head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script></head>
<body>
<div class="container">
<h1>Some other Typography classes</h1>
<p class="text-uppercase">Uppercase</p>
<p class="text-center">text center</p>
<p class="small">small</p>
<p class="font-weight-bold">Bold text</p>
<p class="font-italic">Italic text.</p>
</div>
</body></html>
```

Output:

Some other Typography classes

UPPERCASE

text center

small

Bold text

Italic text.

Bootstrap 4 Colors

Text Colors

The classes for text colors are:

- text-primary:** It gives a sky-blue color to the text.
- text-secondary:** It gives a greyish color to the text.
- text-warning:** It gives a shade of yellow to text.
- text-success:** It gives a color of green to text.
- text-danger:** It gives a color of red to text.
- text-muted:** It gives a grey color to text.
- text-dark:** It gives a dark black color to text.
- text-light:** It gives a very light greyish color to text.
- text-info:** It gives a shade of blue to the text.
- text-white:** It gives a color of white to the text.
- text-body:** It gives default body color. It is often black.

Example:

```
<html><head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">
<p>Use the contextual classes to provide "meaning through colors":</p>
<p class="text-muted">This text is muted.</p>
<p class="text-primary">This text is important.</p>
<p class="text-success">This text indicates success.</p>
<p class="text-info">This text represents some information.</p>
<p class="text-warning">This text represents a warning.</p>
<p class="text-danger">This text represents danger.</p>
<p class="text-secondary">Secondary text.</p>
<p class="text-dark">This text is dark grey.</p>
<p class="text-body">Default body color (often black).</p>
<p class="text-light bg-dark">This text is light grey.</p>
<p class="text-white bg-dark">This text is white.</p>
</div></body></html>
```

Output:

Use the contextual classes to provide "meaning through colors":

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

This text is dark grey.

Default body color (often black).

This text is light grey.

This text is white.

NOTE: The !important rule in CSS is used to add more importance to a property/value than normal.

In fact, if you use the !important rule, it will override ALL previous styling rules for that specific property on that element.

We can also add 50% opacity white and black text colors using "text-white-50" (background must be dark to see changes) and "text-black-50" classes (background must be light to see changes).

50% opacity:

```
<div class="container">
  <h2>Opacity Text Colors</h2>
  <p class="text-success" style="color:black !important">This text indicates success.</p>
  <p>Add 50% opacity for black or white text with the .text-black-50 or .text-white-50
  classes:</p>
  <p class="text-black-50">Black text with 50% opacity on white background</p>
  <p class="text-white-50 bg-dark">White text with 50% opacity on black background</p>
</div>
```

This text indicates success.

This text is black 50% .

This text is white 50%

Background color

The classes for background colors are:

.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark, .bg-light, .bg-transparent, .bg-white

Example:

```
<html><head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">
</script></head>
<body>
<div class="container">
<p>Use the contextual background classes to provide "meaning through colors".</p>
<p class="bg-primary text-white">This text is important.</p>
<p class="bg-success text-white">This text indicates success.</p>
<p class="bg-info text-white">This text represents some information.</p>
<p class="bg-warning text-white">This text represents a warning.</p>
<p class="bg-danger text-white">This text represents danger.</p>
<p class="bg-secondary text-white">Secondary background color.</p>
<p class="bg-dark text-white">Dark background color.</p>
<p class="bg-light text-dark">Light background color.</p>
</div>
</body></html>
```

Output:

Use the contextual background classes to provide "meaning through colors".

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark background color.

Light background color.

Contextual text classes can also be used on links, which will add a darker hover color:

```
<div class="container">
  <h2>Contextual Link Colors</h2>
  <p>Hover over the links.</p>
  <a href="#" class="text-muted">Muted link.</a>
  .
  .
  <a href="#" class="text-secondary">Secondary link.</a>
  <a href="#" class="text-dark">Dark grey link.</a>
  <a href="#" class="text-body">Body/black link.</a>
  <a href="#" class="text-light">Light grey link.</a>
</div>
```

List in Bootstrap

1. **Unstyled** – it is used to remove left margin in list items and also remove style like bullets. This class only applies to immediate children list items.
2. **Inline** – it is used to arrange list in same line. This has no impact or difference between `` or ``.

Example - Inline

```
<div class="container">
  <h2>Typography</h2>
  <p>The class .list-inline places all list items on a single line, when used together with the .list-inline-item:</p>
  <ul class="list-inline text-danger">
    <li class="list-inline-item">Coffee</li>
    <li class="list-inline-item">Tea</li>
    <li class="list-inline-item">Milk</li>
  </ul></div>
```

Typography

The class .list-inline places all list items on a single line, when used together with the .list-inline-item:

Coffee Tea Milk

Example - Unstyled

```
<div class="container">
  <h2>Typography</h2>
  <p>The class .list-unstyled removes the default list-style and left margin on list items (immediate children only):</p>
  <ul class="list-unstyled">
    <li>List 1</li>
    <li>List 2
      <ul>
        <li>List 2.1</li>
        <li>List 2.1</li>
      </ul>
    <li>List 3</li>
  </ul>
```

Typography

The class .list-unstyled removes the default list-style and left margin on list items (immediate children only):

- List 1
- List 2
 - o List 2.1
 - o List 2.1
- List 3

```
</li>
<li>List 3</li>
</ul>
</div>
```

Borders

Use the border classes to add or remove borders from an element:



```
<style>
  .border {display: inline-block; width: 70px; height: 70px; margin: 6px;}
</style>






```

Border Color

Add a color to the border with any of the contextual border color classes:



```
<style>
  .border {display: inline-block; width: 70px; height: 70px; margin: 6px;}
</style>









```

Bootstrap Grid Concept

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

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1			
span 4				span 4				span 4							
span 4				span 8											
span 6						span 6									
span 12															

- Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
- If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:
- The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.
- Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

Grid Classes

The Bootstrap 4 grid system has five classes:

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

create a row (<div class="row">). Then, add the desired number of columns using

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

.col-*-*

The first star (*) represents the responsiveness: sm, md, lg or xl

The second star represents a number, which should always add up to 12 for each row.

The following is a basic structure of a Bootstrap 4 grid:

```
<!-- Control the column width, and how they should appear on different devices -->
<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>

<!-- Or let Bootstrap automatically handle the layout -->
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

- First example: create a row (<div class="row">).
- Then, add the desired number of columns (tags with appropriate .col-*-* classes).
- The first star (*) represents the responsiveness: sm, md, lg or xl, while the second star represents a number, which should add up to 12 for each row.
- Second example: instead of adding a number to each col, let bootstrap handle the layout, to create equal width columns: two "col" elements = 50% width to each col.
- three cols = 33.33% width to each col. four cols = 25% width, etc. You can also use .col-sm|md|lg|xl to make the columns responsive.
- Below we have collected some examples of basic Bootstrap 4 grid layouts.

Three Equal Columns

The following example shows how to create three equal-width columns, on all devices and screen widths:

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

Width Control Only

Control the column width, but not controls how they should appear on different devices.

- For example, if we have created 3 columns using col-4 then It will display 3 columns for all the different devices .
- col-* : The star(*) represents a number, which should add up to 12 for each row.

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

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

Responsive Columns

- ✓ The following example shows how to create four equal-width columns starting at tablets and scaling to extra large desktops.
- ✓ col-*-* : first star (*) represents the responsiveness: sm, md, lg or xl, while the second star represents a number, which should add up to 12 for each row.
- ✓ **On mobile phones or screens that are less than 576px wide, the columns will automatically stack on top of each other:**

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

Two Unequal Responsive Columns

The following example shows how to get two various-width columns starting at tablets and scaling to large extra desktops:

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

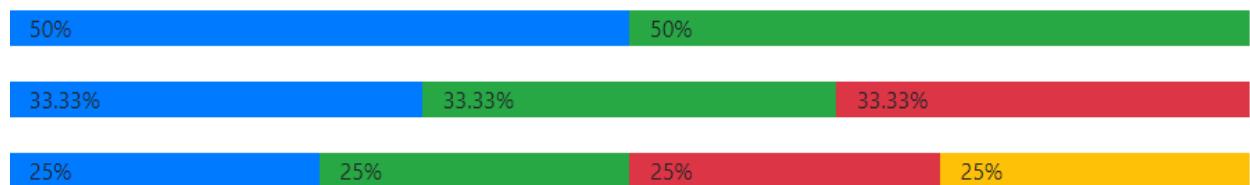
Example

```
<body>
<div class="container-fluid">
<h1>Basic Grid Structure</h1>
<div class="container-fluid">
    <div class="row">
        <div class="col-sm-6 bg-primary">50%</div>
        <div class="col-sm-6 bg-success">50%</div>
    </div> <br>
    <div class="row">
        <div class="col-sm-4 bg-primary">33.33%</div>
        <div class="col-sm-4 bg-success">33.33%</div>
        <div class="col-sm-4 bg-danger">33.33%</div>
    </div> <br>
```

```
<div class="row">
    <div class="col-sm bg-primary">25%</div>
    <div class="col-sm bg-success">25%</div>
    <div class="col-sm bg-danger">25%</div>
    <div class="col-sm bg-warning">25%</div>
</div> <br>
</div></div></body></html>
```

Output:

Basic Grid Structure



Example: To understand the concept of Responsive Columns for different screen sizes

```
<div class="container-fluid">
<div class="row">
    <div class="col-lg-3 col-md-6 bg-info">1</div>
    <div class="col-lg-3 col-md-6 bg-danger">1</div>
    <div class="col-lg-3 col-md-6 bg-warning">1</div>
    <div class="col-lg-3 col-md-6 bg-success">1</div>
</div>
</div>
```

Large screen size $\geq 992\text{px}$: 4columns



Medium screen size $\geq 768\text{px}$ but up to 991



For screen size less than 768 columns will stack on top of each other



- ✓ For screen size $\geq 992\text{px}$ four columns will be displayed.
- ✓ 768px to 991px 2 columns will be displayed.
- ✓ The columns will automatically stack on top of each other when the screen is less than 768px wide.

Bootstrap Tables

We can create different types of Bootstrap tables by using different classes to style them.

Class	Use
table	The basic Bootstrap table has a light padding and only horizontal dividers. The .table class must be used to add basic styling to a table.
table-bordered	This class is used to add borders on all sides of the table and cells.
table-hover	This class is used to enable a hover state on table rows.
table-dark	This class is used to add a black background to the table
table-striped	This class is used to add zebra-stripes to a table
table-borderless	This class removes borders from the table.
thead-dark	This class adds a black background to table headers.
table-sm	This class is used to make the table smaller by cutting cell padding in half.
.table-responsive	adds a scrollbar to the table when needed (when it is too big horizontally): table-responsive-sm/md/lg/xl

Contextual classes

Bootstrap offers a variety of classes combined under the name contextual classes. We can use them to color the whole table, its rows, columns, or even a single cell.

The list of contextual classes is given below.

- table-primary, table-secondary, table-success, table-danger, table-warning
table-info, table-light, table-dark, table-active

Example: (Table head theme)

```
<body>
<table class="table table-bordered table-hover table-sm table-info">
<thead class="thead-dark">
<tr align="center">
    <th>Name</th>
    <th>Value</th>
</tr> </thead>
<tr align="center">
    <td>Fruit</td>
    <td>Mango</td>
</tr>
```

```
<tr align="center">
    <td>Flower</td>
    <td>Rose</td>
</tr>
<tr align="center">
    <td>Animal</td>
    <td>Cat</td>
</tr>
</table></body>
```

Output:

Name	Value
Fruit	Mango
Flower	Rose
Animal	Cat

Example

```
<body>
<table class="table table-borderless table-striped table-hover table-dark">
<thead>
<tr align="center" class="bg-success">
    <th>Name</th>
    <th>Value</th>
</tr>
</thead>
<tr align="center">
    <td>Fruit</td>
    <td>Mango</td>
</tr>
<tr align="center">
    <td>Flower</td>
    <td>Rose</td>
</tr>
<tr align="center">
    <td>Animal</td>
    <td>Cat</td>
</tr></table></body>
```

Output:

Name	Value
Fruit	Mango
Flower	Rose
Animal	Cat

Images

- **Bootstrap** offers different classes for images to make their appearance better and also to make them responsive.
- Making an image responsive means it should scale according to its parent element.
- That is, the size of the image should not overflow its parent element and will grow and shrink according to the change in the size of its parent without losing its aspect ratio.

Rounded

The .rounded class adds rounded corners to an image.

Example:

```
<body>

</body>
```

Output:



Circle

The .rounded-circle class shapes the image to a circle

Example

```
<body>

</body>
```

Output:



Thumbnail

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

Example

```

```

Output:



Responsive Images

The .img-fluid class is used within the tag to create the responsive image. The responsive image is used to adjust the image automatically to the specified box.

Syntax:

```

```

Effect of **class= “img-fluid”** in **** - The responsive image is used to adjust the image automatically to the specified box which consider as responsive image.

Alignment Classes

- **float-left** – Attaches image to leftmost side
- **float-right** – Attaches image to rightmost side
- **mx-auto d-block** – margin auto sets image as centrally aligned element

Example:

```
<body>



</body>
```

Output:



Alerts

Success, Info, Warning, Danger

Alerts are shown on website before and after completing an action. These are highlighted text that are important to take into consideration while performing a process. Alerts are created with the .alert class, followed by one of the contextual classes

.alert-success, .alert-info, .alert-warning, .alert-danger, .alert-primary, .alert-secondary, .alert-light or .alert-dark.

Syntax:

```
<div class="alert alert-success"> Contents... </div>
```

Closing Alerts:

The *.alert-dismissible* class is used within *.container* class to close the alert message. Then use **class="close"** and **data-dismiss="alert"** to link a button element.

Syntax:

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="close" data-dismiss="alert">&times;</button>
  Alert message
</div>
```

Animated Alerts:

The *.fade* and *.show* classes are used to add the fading effect of animation when closing the alert message.

Syntax:

```
<div class="alert alert_type alert-dismissible fade show">
  <button type="button" class="close" data-dismiss="alert">&times;</button>
  Alert message
</div>
```

Alert Links

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

```
<div class="alert alert-success">
  <strong>Success!</strong> You should <a href="#" class="alert-link">read this
  message</a>.
</div>
```

Example

```
<div class="container">
<h2>Alerts</h2>
<p>Alerts are created with the .alert class, followed by a contextual color classes:</p>
<div class="alert alert-success">
    <strong>Success!</strong> This alert box could indicate a successful or positive action.
</div></div>
```

Output:

Alerts

Alerts are created with the .alert class, followed by a contextual color classes:

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

Closing Alert

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

Example

```
<body>
<div class="container">
<div class="alert alert-success alert-dismissible">
    <strong>SUCCESS!!!!!!!!!!</strong>FORM FILLED SUCCESSFULLY
    <button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-info alert-dismissible">
    <strong>SUCCESS!!!!!!!!!!</strong>FORM FILLED SUCCESSFULLY
    <button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-warning">
    <strong>SUCCESS!!!!!!!!!!</strong>FORM FILLED SUCCESSFULLY
    <button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-danger">
    <strong>SUCCESS!!!!!!!!!!</strong>FORM FILLED SUCCESSFULLY
    <button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-primary">
    <strong>SUCCESS!!!!!!!!!!</strong>FORM FILLED SUCCESSFULLY
```

```

<button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-secondary">
  <strong>SUCCESS!!!!!!</strong> FORM FILLED SUCCESSFULLY
  <button class="close" data-dismiss="alert">&times;</button>
</div>
<div class="alert alert-dark">
  <strong>SUCCESS!!!!!!</strong> FORM FILLED SUCCESSFULLY
  <button class="close" data-dismiss="alert">&times;</button>
</div></div></body></html>

```

Output:**Notes:**

- In above code, “alert-dismissible” class is not mandatory.
- For alert and closing to work, we have to link with .js file of bootstrap also, because after clicking x, it won’t be able to close alert if .js is not included. So, bootstrap.min.js file must be imported.
- Animated Alerts: The .fade and .show classes are used to add the fading effect of animation when closing the alert message.

Button style

Bootstrap provides different classes that can be used with different tags, such as <button>, <a>, <input>, and <label> to apply custom button styles. Bootstrap contains many classes to set the style of the button element. The list of button classes is given below:

Variants	Bootstrap includes different styles of buttons, each serving its own semantic purpose, with some extras thrown in for further control.	<ul style="list-style-type: none"> ➤ .btn ➤ .btn-secondary ➤ .btn-primary ➤ .btn-success ➤ .btn-info ➤ .btn-warning ➤ .btn-danger ➤ .btn-link ➤ .btn-dark ➤ .btn-light
Outline button	To get the button without heavy background colors, use .btn-outline-* class which allows you to remove all background images and colors from any button.	<ul style="list-style-type: none"> ➤ .btn-outline-secondary ➤ .btn- outline-primary ➤ .btn- outline-success ➤ .btn- outline-info ➤ .btn- outline-warning ➤ .btn- outline-danger ➤ .btn- outline-dark ➤ .btn- outline-light
Button sizes	To get any larger or smaller buttons add the classes .btn-lg , .btn-sm to the .btn . You can create your own custom size button by using CSS variables.	<ul style="list-style-type: none"> ➤ .btn-lg ➤ .btn-md ➤ .btn-sm
Block Level button	Add class .btn-block to create a block level button that spans the entire width of the parent element.	<ul style="list-style-type: none"> ➤ .btn-block
Button states	<ul style="list-style-type: none"> ➤ A button can be set to an active or disabled state. The class="active" attribute makes a button unclickable. ➤ <a> tag do not support the disabled attribute and must therefore use the class="disabled" to make it visually appear disabled. 	<ul style="list-style-type: none"> ➤ .active ➤ .disabled

Example (variant,outline,size)

```
<body>
<div class="container">
<h2>Button Styles</h2>
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-primary btn-lg">Primary</button>
<button type="button" class="btn btn-primary btn-sm">Primary</button>
```

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

Output:

Button Styles

Basic

Primary

Primary

Primary

Primary

Example (button states)

```
<body><div class="container">
    <button type="button" class="btn btn-primary active">Primary active</button>
    <button type="button" class="btn btn-outline-primary active">Primary outline
active</button>
    <button type="button" class="btn btn-outline-primary disabled">Primary
disabled</button>
</div></body>
```

Output:

Primary active

Primary outline active

Primary disabled

The button classes can be used on `<a>`, `<button>`, or `<input>` elements:

Example:

```
<div class="container">
    <a href="#" class="btn btn-warning" role="button">Link Button</a>
    <button type="button" class="btn btn-danger">Button</button>
    <input type="button" class="btn btn-primary" value="Input Button">
    <input type="submit" class="btn btn-secondary" value="Submit Button">
</div>
```

Link Button

Button

Input Button

Submit Button

Loading/Processing button

Spinners are used to display the loading state of a component or a page. The spinners within buttons are used to represent an action that is currently processing.

- **.spinner-border:** It is used to create a spinner with the border.
- **.spinner-grow:** It is used to create a grow and shrink spinner.

Example:

```
<body>
<div class="container">
<h2>Spinner Buttons</h2>
<button class="btn btn-primary">
    <span class="spinner-border spinner-border-sm"></span>
</button>
<button class="btn btn-primary">
    <span class="spinner-grow spinner-grow-sm"></span>
    Loading..
</button>
<button class="btn btn-primary disabled">
    <span class="spinner-border spinner-border-sm"></span>
    Loading..
</button>
<button class="btn btn-primary disabled">
    <span class="spinner-grow spinner-grow-sm"></span>
    Loading..
</button>
</div></body>
```

Output:

Spinner Buttons



Block Level Buttons

Add class `.btn-block` to create a block level button that spans the entire width of the parent element.

Full-Width Button

Example

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

Spinner in bootstrap

type	class
Spinner - border	.spinner-border
Spinner grow	.spinner-grow
Spinner color	.spinner-border .text-success (bs color class)
Spinner size	.spinner-border-sm .spinner-grow-sm

Example:

```
<body>
<div class="container">
<h2>Spinners</h2>
    <div class="spinner-border"></div>&nbsp;&nbsp;
    <div class="spinner-border text-success"></div>&nbsp;&nbsp;
    <div class="spinner-grow text-info"></div>&nbsp;&nbsp;
    <div class="spinner-border spinner-border-sm text-success"></div>&nbsp;&nbsp;
    <div class="spinner-border text-danger">Loading...</div>
</div></body>
```

Output:



Progress Bars

A progress bar is used to display the progress of a process on a computer. A progress bar displays how much of the process is completed and how much is left. You can add a progress bar on a web page using predefined bootstrap classes. Bootstrap provides many types of progress bars.

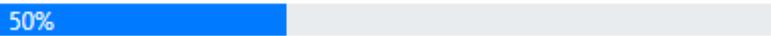
- **.progress** – container of progress bar
- **.progress-bar** – this denotes how much progress bar should be visible with progress

Example:

```
<body>
<div class="container">
<h2>Progress Bar</h2>
<div class="progress">
    <div class="progress-bar" style="width:50%">50%</div>
</div>
</div></body>
```

Output:

Progress Bar



Note:

- In above example, it creates a progress bar occupying entire width and showing 50% progress in default blue color.
- We can color progress bar by adding bg-success or other background color class with “progress-bar” class.
- **To increase height**, provide inline CSS in main Div.
`<div class="progress" style="height:30px">`

Striped Progress Bar

The .progress-bar-striped class adds stripes to the progress bars.

Animated Progress Bar

Add the .progress-bar-animated class to animate the progress bar.

Note: Effect of animated progress bar can be seen using striped progress bar only.

Multiple Progress Bars

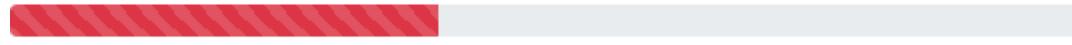
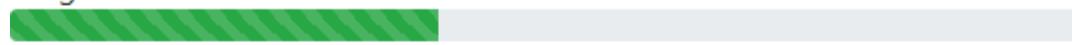
Create a stacked progress bar by placing multiple bars into the same div with `class="progress"`

Example (Different color + stacked bar)

```
<body>
<div class="container">
    Progress bar
    <div class="progress">
        <div class="progress-bar bg-success progress-bar-striped" style="width:40%"></div>
    </div>
    <br>
    <div class="progress">
        <div class="progress-bar bg-danger progress-bar-striped progress-bar-animated" style="width:40%"></div>
    </div>
    <br>
    <div class="progress" style="border:1px solid black">
        <div class="progress-bar bg-success" style="width:40%">
            Free Space 40%
        </div>
        <div class="progress-bar bg-warning" style="width:10%">
            Warning 10%
        </div>
        <div class="progress-bar bg-danger" style="width:20%">
            Danger 20%
        </div>
    </div>
    </div>
</body>
```

Output:

Progress bar



Different Heights of Progress Bars:

- The height of the progress bar is 16px by default.
- However, a user can utilize the CSS height property to modify the height.
- But at the same time, a user has to set the same height for the progress container and the progress bar.

The Example for Different Heights of Progress Bars is given below:

```
<div class="container">
<div class="progress" style="height:10px">
<div class="progress-bar" style="width:40%"></div>
</div>
<br>
<div class="progress" style="height:20px">
<div class="progress-bar" style="width:50%"></div>
</div>
<br>
<div class="progress" style="height:30px">
<div class="progress-bar" style="width:60%"></div>
</div> </div>
```



Example:

**Write HTML code using bootstrap class to design a progressbar as shown below.
Add animation for the progressbar.**



```
<div class="container">
<div class="progress ">
<div class="progress-bar bg-success progress-bar-striped progress-bar-animated border border-dark" style="width:50%">50%
</div>
<div class="progress-bar bg-warning progress-bar-striped progress-bar-animated border border-dark" style="width:20%"> 20%
</div>
<div class="progress-bar bg-danger progress-bar-striped progress-bar-animated border border-dark" style="width:20%"> 20% </div>
</div>
free space <span class="progress bg-white"><span class="progress-bar bg-success rounded" style="width:3%"></span></span>
Used Space <span class="progress bg-white"><span class="progress-bar bg-warning rounded" style="width:3%"></span></span>
Danger<span class="progress bg-white"><span class="progress-bar bg-danger rounded" style="width:3%"></span></span>
</div>
```

Pagination

Pagination is used to sort the web pages of your website in an organized manner. It becomes very necessary if your website has a lot of web pages.

Pagination Class:

.pagination - To create a basic pagination add to an `` element

.page-item – Add to each `` element

.page-link - Add to each link inside ``

Class	Description
.pagination	Add this class to get the pagination on your page.
.disabled, .active	you can customize links by using .disabled for unclickable links and .active to indicate the current page by highlighting.
.justify-content-center/.justify-content-end	Aligns entire control to center or end.

Basic Pagination:

- ✓ In order to create a basic pagination, the user has to include the **.pagination** class to an `` element.
- ✓ After that, the user will have to add the **.page-item** to each `` element and a **.page-link** class to each link`<a>` inside ``.

```
<div class="container">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">Page 1</a></li>
    <li class="page-item"><a class="page-link" href="#">Page 2</a></li>
    <li class="page-item"><a class="page-link" href="#">Page 3</a></li>
    <li class="page-item"><a class="page-link" href="#">Page 4</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</div>
```

Previous Page 1 Page 2 Page 3 Page 4 Next

Active and disabled State Pagination:

- ✓ In order to make the active state pagination, the **.active** class can be used to highlight the current page.
- ✓ The disabled state is used for the purpose of creating an unclickable links. The class used in this case is **.disabled** class.

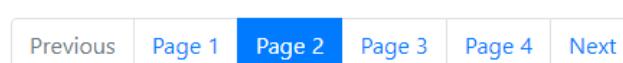
```
<div class="container m-4">
<ul class="pagination">
  <li class="page-item disabled"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 1</a></li>
  <li class="page-item active"><a class="page-link" href="#">Page 2</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 3</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 4</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
</div>
```



Sizing in Pagination:

- ✓ The sizes of pagination can be changed. In order words, pagination blocks can also be sized to a larger or a smaller size.
- ✓ Add class **.pagination-lg** for larger blocks
- ✓ Add class **.pagination-sm** for smaller blocks

```
<!--large size -->
<ul class="pagination pagination-lg" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>
<!--default size -->
<ul class="pagination" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>
<!--Small size -->
<ul class="pagination pagination-sm" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>
```



Alignment in Pagination:

- ✓ The alignment in pagination uses the utility classes to change the alignment of the pagination.
- ✓ Add class **justify-content-center** to align pagination in **center**
- ✓ Add class **justify-content-end** to align pagination in **right** side of the page

```
<!-- Default (left-aligned) -->
<ul class="pagination" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>

<!-- Center-aligned -->
<ul class="pagination justify-content-center" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>

<!-- Right-aligned -->
<ul class="pagination justify-content-end" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>
```



Dropdown

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list

.dropdown - Indicates a dropdown menu.

To open the dropdown menu, use a button or a link with a class of .dropdown-toggle and the data-toggle="dropdown" attribute.

Add the .dropdown-menu class to a <div> element to actually build the dropdown menu. Then add the .dropdown-item class to each element (links or buttons) inside the dropdown menu.

Class	Detail
.dropdown .dropdown-toggle .dropdown-menu .dropdown-item	To create basic dropdown
.dropdown-divider	It used to separate links inside the dropdown menu with a thin horizontal border
.dropright .dropleft	The .dropright and .dropleft classes are used to set the position of dropdown list in left and right side.
.dropdown-header	It is used to add header section inside the dropdown list.
.dropup	The .dropup class is used instead of .dropdown class to expand the menu list in upwards.

Note: To Disable and Active items: The *.active* class is used to add the highlight the list items. The *.disabled* class is used to disable the list of items.

Example:

```
<div class="container m-4">
<div class="dropdown">
<button type="button" class="btn btn-success dropdown-toggle" data-
toggle="dropdown">
    Subjects
</button>
<div class="dropdown-menu">
    <a class="dropdown-item" href="#">FSD-1</a>
    <a class="dropdown-item" href="#">FCSP-1</a>
    <a class="dropdown-item" href="#">DE</a>
    <a class="dropdown-item" href="#">PS</a>
</div>
</div>
</div>
```

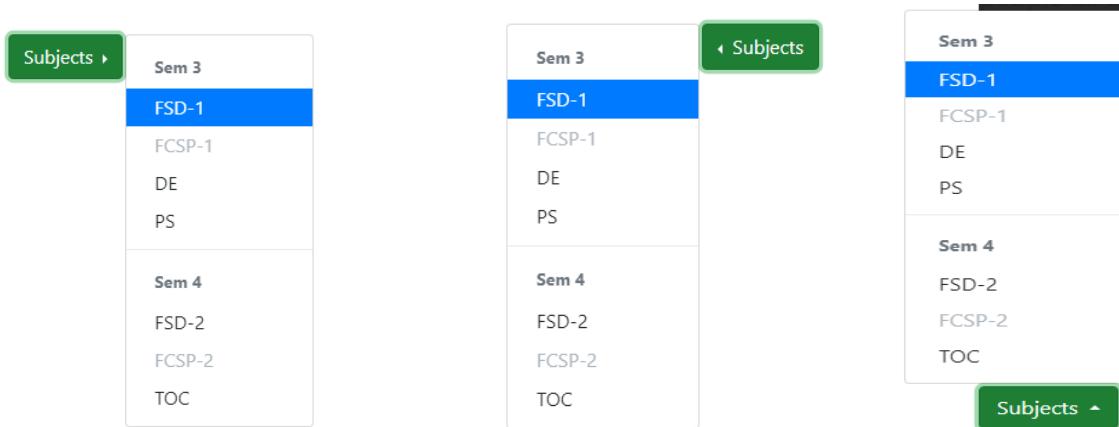


Divider & Header Dropdown

```
<div class="container m-4">
  <div class="dropdown">
    <button type="button" class="btn btn-success dropdown-toggle" data-
      toggle="dropdown">
      Subjects
    </button>
    <div class="dropdown-menu">
      <strong class="dropdown-header">Sem 3</strong>
      <a class="dropdown-item active" href="#">FSD-1</a>
      <a class="dropdown-item disabled" href="#">FCSP-1</a>
      <a class="dropdown-item" href="#">DE</a>
      <a class="dropdown-item" href="#">PS</a>
      <div class="dropdown-divider"></div>
      <strong class="dropdown-header">Sem 4</strong>
      <a class="dropdown-item" href="#">FSD-2</a>
      <a class="dropdown-item disabled" href="#">FCSP-2</a>
      <a class="dropdown-item" href="#">TOC</a>
    </div>
  </div>
</div>
```



Dropdown Position: The `.dropright` and `.dropleft` classes are used to set the position of dropdown list in left and right side. And `.dropdown` will expand upwards.



```
<div class="dropdown dropright">  <div class="dropdown dropleft">  <div class="dropdown">
```

Dropdown Text

The **.dropdown-item-text** class is used to add plain text to a dropdown item, or used on links for default link styling.

```
<div class="dropdown">
  <button type="button" class="btn btn-success dropdown-toggle" data-toggle="dropdown">
    Subjects
  </button>
  <div class="dropdown-menu">
    <strong class="dropdown-header">Sem 3</strong>
    <a class="dropdown-item active" href="#">FSD-1</a>
    <a class="dropdown-item disabled" href="#">FCSP-1</a>
    <p class="dropdown-item-text">DE</p>
    <a class="dropdown-item-text" href="#">PS</a>
  </div>
</div>
```



Dropdown Menu Right



(a) Default



(b)<div class="dropdown-menu dropdown-menu-right">

Collapse

Collapse component is used to show or hide the content by using **.collapse** class.

The content can be collapsed by adding **data-toggle="collapse"** attribute anchor or button element.

The id of these elements references to the id of the content to collapse the data.

Example:

```
<div class="container">
<h2>Simple Collapse</h2>
<button type="button" class="btn btn-primary" data-toggle="collapse"
        data-target ="#demo"> CLICK TO COLLAPSE </button>
<div id="demo" class="collapse">
    Welcome to LJ UNIVERSITY. Full stack development course.
</div>
</div>
```

Example Explained

- ✓ The **.collapse** class indicates a collapsible element (a **<div>** in our example); this is the content that will be shown or hidden with a click of a button.
- ✓ To control (show/hide) the collapsible content, add the **data-toggle="collapse"** attribute to an **<a>** or a **<button>** element. Then add the **data-target="#id"** attribute to connect the button with the collapsible content (**<div id="demo">**).

Output:



Collapse using Link

- ✓ You can collapse the content with **<a>** tag by using **href** value of the ID of the content to collapse.

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

```
<a href="#demo" data-toggle="collapse">Collapsible</a>
<div id="demo" class="collapse">Lorem ipsum dolor text....</div>
```

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

```
<div id="demo" class="collapse show">Lorem ipsum dolor text....</div>
```

Accordion

- Accordion is a graphical element consisting of a vertically stacked list of items.
- You can use collapsible content to make an accordion which is often used for content such as FAQs, overviews, etc. \
- In this digital era, every website contains an accordion at least one of their pages if they have a FAQ section.
- With the use of accordion you can make your content display more interactive.
- The data-parent attribute makes sure that all collapsible elements under the specified parent will be closed when one of the collapsible item is shown.

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

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

Class/Attribute	Detail
data-parent=""	Use the data-parent attribute to make sure that all collapsible elements under the specified parent will be closed when one of the collapsible item is shown.
.card	To provide a flexible and extensible content container with multiple variants and options.
.card-link	To add a hover effect and blue color to any link in a card.
.card-body	It is the building block of a card.
.card-header	It adds a heading to the card

Example

```
<body>
<h3>Accordion</h3>

<div id="accordion">
<div class="card">

    <div class="card-header">
        <a href="#collapseone" class="card-link" data-toggle="collapse">ITEM 1</a>
    </div>
    <div id="collapseone" class="collapse show" data-parent="#accordion">
        <div class="card-body">Demo1</div>
    </div>
</div>

<div class="card">
<div class="card-header">
    <a href="#collapsetwo" class="card-link" data-toggle="collapse">ITEM 2</a>
</div>
<div id="collapsetwo" class="collapse" data-parent="#accordion">
    <div class="card-body">Demo2</div>
</div>
</div>

<div class="card">
<div class="card-header">
    <a href="#collapsethree" class="card-link" data-toggle="collapse">ITEM 3</a>
</div>
<div id="collapsethree" class="collapse" data-parent="#accordion">
    <div class="card-body">Demo3</div>
</div></div>
</div>
```

Output:

Accordion

ITEM 1
Demo1
ITEM 2
ITEM 3