

BOOTSTRAP-5 CURRICULUM

What is Bootstrap?

- Bootstrap Components
- Bootstrap Advantages
- About Bootstrap-5
- Bootstrap 5 Setup
- CSS & JavaScript files

Bootstrap 5 Containers

- Fixed Container
- Fluid Container
- Container Padding
- Container Border & Color
- Responsive Container
- Grid Classes

Bootstrap 5 Grid System

- Basic Structure of Grid
- 3 Equal Columns
- Responsive Columns
- 2 Unequal responsive columns

Bootstrap 5 Typography

- Body Text
- Headings
- Sub-Headings
- Lead Text
- Marked Text, Abbreviation's etc.,,

Bootstrap 5 Tables

- Basic Table
- Dark Table
- Striped Table
- Bordered Table
- Hover Rows
- Table Head Styles
- Condensed Table

Bootstrap 5 Forms

- Basic Form, Inline Form
- Horizontal Form, Hidden Labels
- Form Row
- Form Legends
- Form Control Size
- Form Label Size
- Column Size
- Help Text
- Static Control
- Check Box and radio buttons
- Inline

Bootstrap 5 Custom Forms

- Checkboxes & Radio Buttons
- Inline Controls & Select Controls
- File Upload

Bootstrap 5 Buttons

- Button Elements

- Button Size
- Block Level Buttons
- Disabled Buttons
- Toggle Checkboxes & Radio Buttons

Bootstrap 5 Images

- Responsive Images
- Rounded Corners & Thumbnails
- Image Alignments, Float, Centering Block-level Images, Centering Non-block level Images

Bootstrap 5 Dropdowns

- Creating Dropdown
- About ARIA Attributes
- Drop up Menu
- Align Right
- Menu Headers
- Dividers
- Disabled Menu Items
- Drop end & drop start position

Bootstrap 5 Button Groups

- Horizontal Button Group
- Vertical Button Group
- Button Group Size
- Button Toolbar
- Combining Dropdowns With Button Groups
- Split Dropdowns

Bootstrap 5 Input Groups

- Text on both sides

- Size
- Button Add-ons

Bootstrap 5 Collapse

- Collapse Using a link
- Collapse Using a Button and Accordion

Bootstrap 5 Navs

- Base Nav on and <nav> elements
- Vertical Navs, Tabs
- Pills, Vertical Pills, Disabled Links
- Tabs with Dropdown Menus - Tabs, Pills, Tabbable Panes

Bootstrap 5 Navbars

- Basic Navbars
- Navbars with Dropdowns
- Current Page
- navbars with Form Controls
- navbars Colors - Primary color light, Primary color Dark
- Custom Color
- Fixed Navbars - Fixed Top, Fixed Bottom
- Vertical Navbars
- Brand/Logo in navbars

Bootstrap 5 Breadcrumbs

- Creating Breadcrumbs
- Non-list Breadcrumbs

Bootstrap 5 Pagination

- Default Pagination

- Active State for Current Page
- Pagination Size
- Disabling an Option
- Pagination Alignments
- Stripe Bar
- Animated Stripe Bar
- Multiple Progress Bars

Bootstrap 5 Badges

- Contextual Classes
- Pill Badges
- Badges inside an element

Bootstrap 5 Jumbotron

- Jumbotron
- Full Width Jumbotron

Bootstrap 5 Alerts

- Alerts
- Dismissible Alerts
- Fading Effect
- Alerts with links

Bootstrap 5 Modal

- Modal
- Modal Sizes - Small, Large
- Remove Fade Effect
- Animation Modal
- Full Screen Modal
- Centered Modal
- Scrolling Modal

Bootstrap 5 Progress Bars

- Default Progress Bar
- Contextual Classes

Bootstrap 5 List Groups

- Create a Default List Group
- Badges
- Linked List
- Contextual Classes
- Custom Content
- Flush/Remove Borders
- Numbered Lists Group
- Horizontal list group
- List group with Badges

Bootstrap 5 Cards

- Basic Card
- Header & Footer
- HTML Header,
- Header Navigation - Nav, Pill
- Flush Content(Remove Padding)
- Top & Bottom
- List Groups
- Links, Text Alignments
- Contextual Color
- Border Color
- Image Overlay
- Card Width,
- Card Group, Card Decks, Card Columns

Bootstrap 5 Responsive Embed

- Responsive Embed Videos

- Popovers on Hover

Bootstrap 5 Carousel

- Basic Carousel
- Add Controls
- Add Indicators
- Add Captions

Bootstrap 5 Tooltips

- Tooltip Initialization
- Tooltip on link
- Tooltip on button
- Tooltip Position

Bootstrap 5 Popovers

- Popover Initialization
- Popover Position
- Popover Title
- Closing Popovers

Bootstrap 5 Scrollspy

Scrollspy

Bootstrap 5 off Canvas

Introduction to off canvas

Positioning the off canvas

Bootstrap 5 Spinners

Introduction to spinners

Colored spinners

Growing spinners

Spinner button

Bootstrap 5 Toasts

Introduction to Toasts

Examples of Toasts

Bootstrap 5 Icons

Bootstrap 5 Landing Page Design

FutureVisionComputers

CHAPTER 1: WHAT IS BOOTSTRAP?

Bootstrap is a free and open-source framework for creating websites and web applications. It's the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Bootstrap is promoted as being “**One framework, every device**”. This is because websites built with Bootstrap will automatically scale between devices — whether the device is a mobile phone, tablet, laptop, desktop computer, screen reader, etc.

Bootstrap Components:

Bootstrap includes components such as **buttons, navbars, dropdown menus, alert boxes, and more**. In most cases, you can make use of a component simply by using the appropriate class name.

We'll be looking at these components throughout this tutorial. We've got plenty of interactive examples that enable you to see exactly how these components work

Advantages of Bootstrap:

- One of the main benefits of development frameworks like Bootstrap is that they can help speed up development times, while maintaining quality and consistency across the site. You no longer need to re-design every element. And you don't need to spend hours trying to get everything looking and working right across browsers, platforms, and devices. By using Bootstrap, all (most) of the hard work is one for you.
- Given Bootstrap is the most popular frontend development framework on the web, this skillset could be a useful one to learn. Adding Bootstrap to your bag of tricks could help you in many ways — from building websites faster, to landing your dream job.
- Also, although Bootstrap comes with its own set of styles, these are easy to override. You're not locked into the "Bootstrap design". You are free to use whichever Bootstrap components you choose, while adding your own on top. There are thousands of websites out there that are built on Bootstrap, but with their own design.

About New Bootstrap 5:

- New off canvas component
- New accordion
- New and updated forms
- Checks and radios
- Floating labels
- New file input
- Simplified layout
- RTL is here!
- New utilities API
- Logical properties for spacing utilities
- Grid and layout
- Improved customizing
- Browser support
- JavaScript
- Migration guide

Bootstrap 5 Setup:

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with jsDelivr and a template starter page.

CSS

Copy-paste the stylesheet <link> into your <head> before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha385-1BmE5kBq78iYhFdvKuhfTAU6auU8tT95WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require our own JavaScript plugins and Popper. Place one of the following <script>s near the end of your pages, right before the closing </body> tag, to enable them.

Bundle

Include every Bootstrap JavaScript plugin and dependency with one of our two bundles. Both bootstrap.bundle.js and bootstrap.bundle.min.js include Popper for our tooltips and popovers. For more information about what's included in Bootstrap, please see our contents section.

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha385-ka7Sk0Gln5gmtz2MIQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov5+1p" crossorigin="anonymous"></script>
```

Separate

If you decide to go with the separate scripts solution, Popper must come first (if you're using tooltips or popovers), and then our JavaScript plugins.

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha385-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha385-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN5kzFN1RtK3zEFEIsxhlmWI5/YESvpZ13" crossorigin="anonymous"></script>
```

Create First Web Page With Bootstrap 5:

1. Add the HTML5 doctype

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>
  </html>
```

2. Bootstrap 5 is mobile-first

Bootstrap 5 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

To ensure proper rendering and touch zooming, add the following <meta> tag inside the <head> element:

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

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

CHAPTER 2: CONTAINERS

Bootstrap requires a containing element to wrap elements and contain its grid system (more on the grid system next). Bootstrap's container classes were created specifically for this purpose.

Bootstrap containers can be either **fixed or fluid**.

- 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

1. Fixed Containers:

A fixed container is a (responsive) fixed width container. As you resize your browser, its width remains intact, until it passes a certain breakpoint (as specified by you — more on that next), at which time it will resize to the new width for that break point.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container">
<h1>My First Bootstrap Page</h1>
<p>This part is inside a .container class.</p>
<p>The .container class provides a responsive fixed width container.</p>
<p>Resize the browser window to see that the container width will change at different breakpoints.</p>
</div>

</body>
</html>
```

Output:

My First Bootstrap Page

This part is inside a .container class.

The .container class provides a responsive fixed width container.

Resize the browser window to see that the container width will change at different breakpoints.

Note

The XXL breakpoint ($\geq 1500\text{px}$) is new in Bootstrap 5, while the largest breakpoint in Bootstrap 5 is Extra-large ($\geq 1200\text{px}$).

2. Fluid Containers:

A fluid container spans the full width of the viewport. It will expand and contract fluidly as you resize the browser. This is in contrast to the fixed width container which will appear to "jump" to the new size as you pass a given break point.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container-fluid class.</p>
  <p>The .container-fluid class provides a full width container, spanning the entire width of the viewport.</p>
</div>

</body>
</html>
```

Output:

My First Bootstrap Page

This part is inside a .container-fluid class.

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

Container Padding:

By default, containers have 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-5** means "add a large **top padding**":

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container pt-5">
  <h1>My First Bootstrap Page</h1>
  <p>This container has an extra top padding.</p>
  <p>Try to remove the .pt-5 class to see the difference.</p>
</div>

</body>
</html>
```

Output:

My First Bootstrap Page

This container has an extra top padding.

Try to remove the .pt-5 class to see the difference.

Container Border & Color:

Other utilities, such as borders and colors, are also often used together with containers:

Example:

```
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container p-5 my-5 border">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a border and some extra padding and margins.</p>
</div>

<div class="container p-5 my-5 bg-dark text-white">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a dark background color and a white text, and some extra padding and margins.</p>
</div>

<div class="container p-5 my-5 bg-primary text-white">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a blue background color and a white text, and some extra padding and margins.</p>
</div>
</body>
</html>
```

Output:

The image shows three vertically stacked containers, each with a different background color and some descriptive text. A red border surrounds the entire stack.

- Top Container:** White background. Text: "My First Bootstrap Page". Subtext: "This container has a border and some extra padding and margins."
- Middle Container:** Dark gray background. Text: "My First Bootstrap Page". Subtext: "This container has a dark background color and a white text, and some extra padding and margins."
- Bottom Container:** Blue background. Text: "My First Bootstrap Page". Subtext: "This container has a blue background color and a white text, and some extra padding and margins."

Responsive Containers:

You can also use the `.container-sm|md|lg|xl` classes to determine when the container should be responsive.

Class	Extra small <code><576px</code>	Small <code>≥576px</code>	Medium <code>≥768px</code>	Large <code>≥992px</code>	Extra-large <code>≥1200px</code>	XXL <code>≥1500px</code>
<code>.container-sm</code>	100%	550px	720px	960px	1150px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1150px	1320px

.container-lg	100%	100%	100%	960px	1150px	1320px
.container-xl	100%	100%	100%	100%	1150px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px

The **max-width** of the container will change on different screen sizes/viewports:

Example :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>
<body>

<div class="container pt-3">
  <h1>Responsive Containers</h1>
  <p>Resize the browser window to see the effect.</p>
</div>

<div class="container-sm border">.container-sm</div>
<div class="container-md mt-3 border">.container-md</div>
<div class="container-lg mt-3 border">.container-lg</div>
<div class="container-xl mt-3 border">.container-xl</div>
<div class="container-xxl mt-3 border">.container-xxl</div>

</body>
</html>
```

Output:

Responsive Containers

Resize the browser window to see the effect.

.container-sm

.container-md

.container-lg

.container-xl

.container-xxl

FutureVisionComputers.com

CHAPTER 3: GRID SYSTEM

Grid system enable you to create advanced layouts using rows and columns. The Bootstrap grid system can have up to **12 columns**, and you can specify how these columns scale for **different viewport sizes**.

Bootstrap grid system layout overview:

.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1	.col-sm-1								
.col-sm-2	.col-sm-3						.col-sm-7												
.col-sm-4			.col-sm-4			.col-sm-4													
.col-sm-5						.col-sm-7													
.col-sm-6						.col-sm-6													
.col-sm-12																			

The numbers at the end of each class name represent the number of columns that the column spans. So **.col-sm-1** spans one column and **.col-sm-8** spans eight. The **sm** means that the colspan applies to small devices and everything above. You can also use **md**, **lg**, and **xl** for medium, large, and extra-large.

Extra small devices are catered for by omitting the middle abbreviation. For example **.col-8** spans eight columns on extra small devices and up (in other words, all devices).

The Bootstrap 5 grid system has six classes:

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)
- **.col-xxl-** (xxlarge devices - screen width equal to or greater than 1500px)

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

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

Basic Structure of Bootstrap5 Grid:

The following is a basic structure of a Bootstrap 5 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, xl or xxl, 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, while three cols = 33.33% width to each col. Four cols = 25% width, etc. You can also use `.col-sm|md|lg|xl|xxl` to make the columns responsive.

Below we have collected some examples of basic Bootstrap 5 grid layouts.



1. Three Equal Columns:

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

  <div class="container-fluid mt-3">
    <h1>Three equal width columns</h1>
    <p>Note: Try to add a new div with class="col" inside the row class - this will create four equal-width columns.</p>
    <div class="row">
      <div class="col p-3 bg-primary text-white">.col</div>
      <div class="col p-3 bg-dark text-white">.col</div>
      <div class="col p-3 bg-primary text-white">.col</div>
    </div>
  </div>

</body>
</html>
```

Output:

Three equal width columns

Note: Try to add a new div with class="col" inside the row class - this will create four equal-width columns.



2. Responsive Columns:

The following example shows how to create four equal-width columns starting at tablets and scaling to extra-large desktops. **On mobile phones or screens that are less than 576px wide, the columns will automatically stack on top of each other:**

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container-fluid mt-3">
<h1>Responsive Columns</h1>
<p>Resize the browser window to see the effect.</p>
<p>The columns will automatically stack on top of each other when the screen is less than 576px wide.</p>
<div class="row">
<div class="col-sm-3 p-3 bg-primary text-white">.col</div>
<div class="col-sm-3 p-3 bg-dark text-white">.col</div>
<div class="col-sm-3 p-3 bg-primary text-white">.col</div>
<div class="col-sm-3 p-3 bg-dark text-white">.col</div>
</div>
</div>

</body>
</html>
```

Output:

Responsive Columns

Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 576px wide.



2. Two Unequal Responsive Columns:

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

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container-fluid mt-3">
  <h1>Two Unequal Responsive Columns</h1>
  <p>Resize the browser window to see the effect.</p>
  <p>The columns will automatically stack on top of each other when the screen is less than 576px wide.</p>
  <div class="row">
    <div class="col-sm-5 p-3 bg-primary text-white">.col</div>
    <div class="col-sm-8 p-3 bg-dark text-white">.col</div>
  </div>
</div>

</body>
</html>
```

Output:

Two Unequal Responsive Columns

Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 576px wide.



Things to Remember with Grids:

Containers:

Grids should be placed within a container (i.e. using either .container class or the **.container-fluid** class) for proper padding and alignment.

Rows & Columns

Rows contain one or more columns. Columns contain the content. Only columns can be immediate children of rows.

More than 12 Columns per Row?

If more than 12 columns are placed in a row, the columns will wrap to a new line. That is, columns will wrap as a group. So for example, if a row contains a **col-md-10** and **col-md-3**, the whole **col-md-3** will wrap to the new line.

CHAPTER 5 - TYPOGRAPHY

1. Body Text:

Bootstrap 5's global **default font-size is 16px** (applied to the <html> element) and the <body> element also has a font-size of 1rem applied (16px == 1rem)

Bootstrap uses a "native font stack" (user's system fonts), with a fallback to **Helvetica Neue, Arial, and sans-serif.**

All headings and **p elements** have their top margin removed. Headings have a bottom margin of .5rem and p elements have a bottom margin of 1rem.

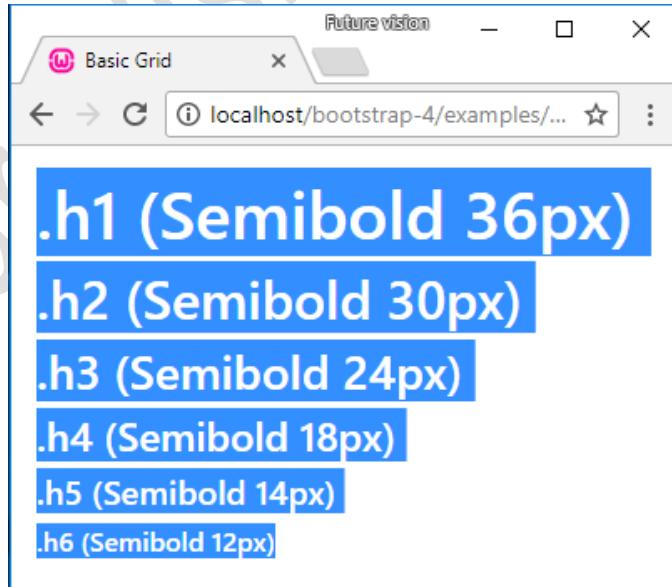
2. Heading Classes:

Bootstrap also includes heading classes from **.h1 to .h6** in case you need to render some inline text in the style of a particular heading.

Example:

```
<p class="h1">.h1 (Semibold 36px) </p>
<p class="h2">.h2 (Semibold 30px) </p>
<p class="h3">.h3 (Semibold 25px) </p>
<p class="h4">.h4 (Semibold 24px) </p>
<p class="h5">.h5 (Semibold 18px) </p>
<p class="h6">.h6 (Semibold 12px) </p>
```

Output:



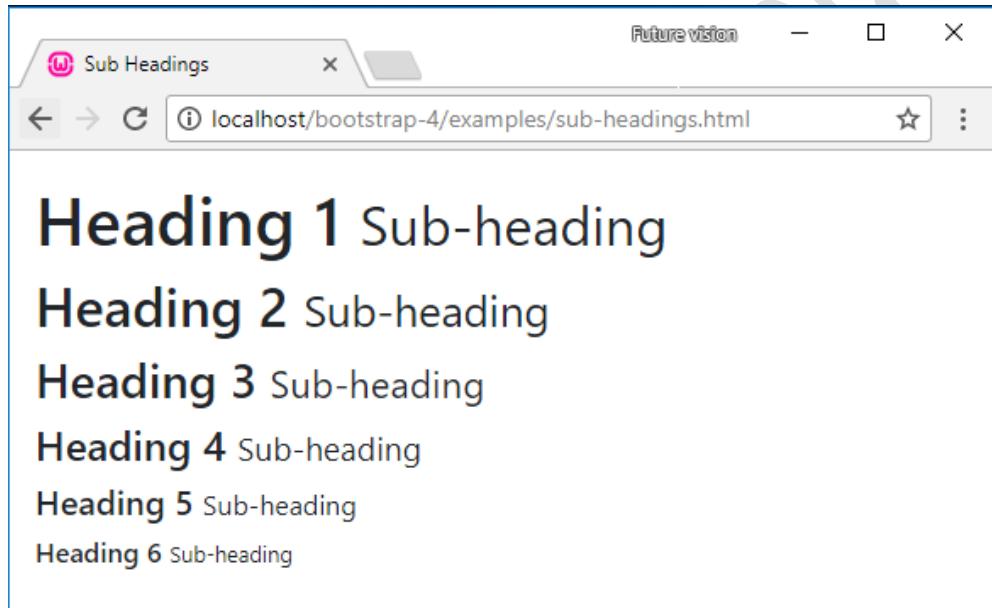
3. Sub Headings:

You can create a sub-heading or secondary text by placing text inside a <small> element within the heading.

Example:

```
<h1>Heading 1 <small>Sub-heading</small></h1>
<h2>Heading 2 <small>Sub-heading</small></h2>
<h3>Heading 3 <small>Sub-heading</small></h3>
<h5>Heading 5 <small>Sub-heading</small></h5>
<h5>Heading 5 <small>Sub-heading</small></h5>
<h6>Heading 6 <small>Sub-heading</small></h6>
```

Output:



5. Display Headings:

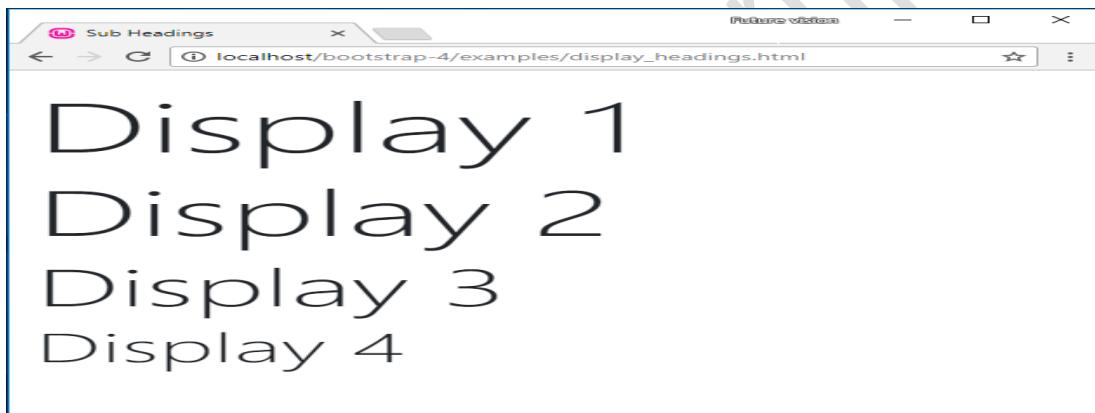
Display headings are designed to stand out more than the normal headings. There are 5 sizes of display heading (**.display-1**, **.display-2**, **.display-3**, **and .display-5**).

So for example, a `<h1>` element could be rendered in 5 different sizes using display heading classes. To create a display heading, add one of the above classes to the heading element.

Example:

```
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-5">Display 5</h1>
```

Output:



5. Lead Text, Marked Text, abbreviations:

Lead Text: You can make a paragraph stand out by using Bootstrap's **.lead** class.

Marked Text: The HTML `<mark>` element represents text as marked or highlighted for reference purposes, due to its relevance in another context.

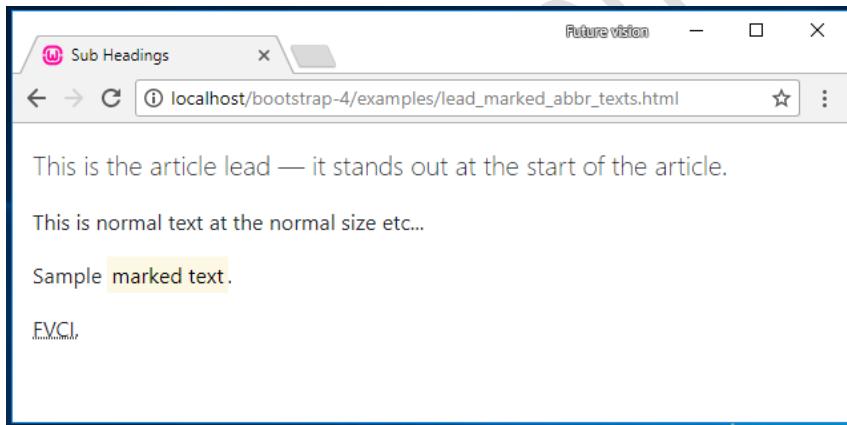
Abbreviation: In HTML, the `<abbr>` represents an abbreviation or acronym. The title attribute can be used to provide an expansion of the abbreviation. In Bootstrap, abbreviations with a title attribute are rendered with a light dotted bottom border and a help cursor on hover.

Example:

```
<div class="container-fluid">
<p class="lead">
This is the article lead — it stands out at the start of the article.
</p>
<p>
This is normal text at the normal size etc...
</p>
<p>Sample <mark>marked text</mark>.</p>
```

```
<abbr title="Future Vision Computer Institute">FVCI.</abbr>
</div>
```

Output:



6. Lists-unstyled, Lists-line:

You can use the Bootstrap **.list-unstyled** class to render lists without their default **list-style** and **left-margin**

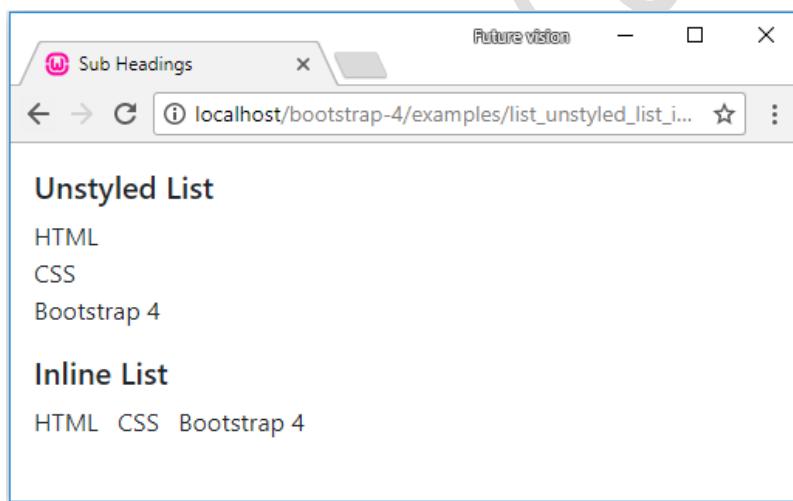
You can use the Bootstrap **.list-inline** and **.list-inline-item** classes to render lists as **display: inline-block** and to apply some padding

Example:

```
<div class="container-fluid">
    <ul class="list-unstyled">
        <li>HTML</li>
        <li>CSS</li>
        <li>Bootstrap 5</li>
    </ul>

    <ul class="list-inline">
        <li class="list-inline-item">HTML</li>
        <li class="list-inline-item">CSS</li>
        <li class="list-inline-item">Bootstrap 5</li>
    </ul>
</div>
```

Output:



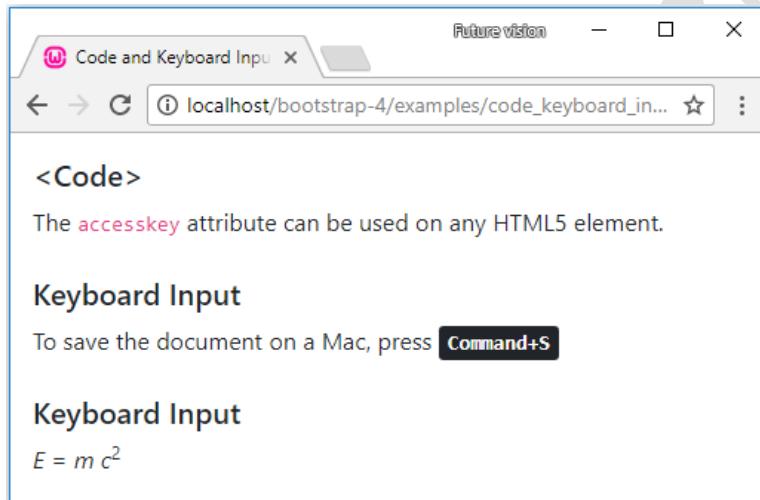
7. Code and Keyboard Input:

Example:

```
<div class="container-fluid">
    <h5>&lt;Code&gt;</h5>
        The <code>accesskey</code> attribute can be used on any HTML5
        element.
        <h5>Keyboard Input</h5>
        To save the document on a Mac, press
        <kbd><kbd>Command</kbd>+<kbd>S</kbd></kbd>

        <var>E</var> = <var>m</var> <var>c</var><sup>2</sup>
    </div>
```

Output:



CHAPTER 5 - TABLES

Bootstrap 5 added some new table classes that assist with building consistently styled and responsive tables.

1. Basic Table:

For a basic table with lightly padded cells and horizontal dividers, apply Bootstrap's **.table** class to the **<table>** element.

Example:

```
<div class="container-fluid">
    <table class="table">
        <thead>
            <tr>
                <th>Header 1</th>
                <th>Header 2</th>
                <th>Header 3</th>
            </tr>
        </thead>
        <tfoot>
            <tr>
                <th>Footer 1</th>
                <th>Footer 2</th>
                <th>Footer 3</th>
            </tr>
        </tfoot>
        <tbody>
            <tr>
                <td>Cell</td>
                <td>Cell</td>
                <td>Cell</td>
            </tr>
            <tr>
                <td>Cell</td>
                <td>Cell</td>
                <td>Cell</td>
            </tr>
            <tr>
                <td>Cell</td>
                <td>Cell</td>
                <td>Cell</td>
            </tr>
            <tr>
                <td>Cell</td>
                <td>Cell</td>
                <td>Cell</td>
            </tr>
        </tbody>
    </table>
</div>
```

Output:

Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

2. Dark Table

Dark tables are new in Bootstrap 5. You can use the **.table-dark** class to invert the colors so that it has light text on a dark background.

```
<table class="table table-dark">
```

Output:

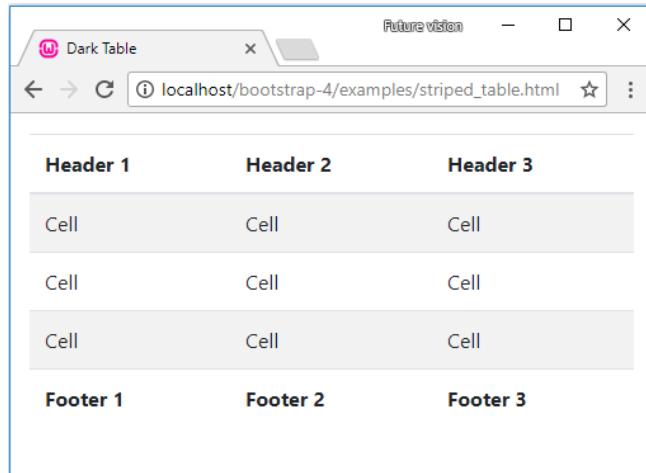
Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

3. Striped Table

For alternating background colors on the rows, add Bootstrap's **.table-striped** class (while keeping the **.table** class in place).

```
<table class="table table-striped">
```

Output:



A screenshot of a web browser window titled "Dark Table". The address bar shows "localhost/bootstrap-4/examples/stripped_table.html". The table has three columns labeled "Header 1", "Header 2", and "Header 3". It contains six rows: the first row is a header row, the next three rows are data rows, and the last two rows are footer rows. The data rows have a light gray background color, while the header and footer rows have white backgrounds. The footer rows also contain the same header text as the header row.

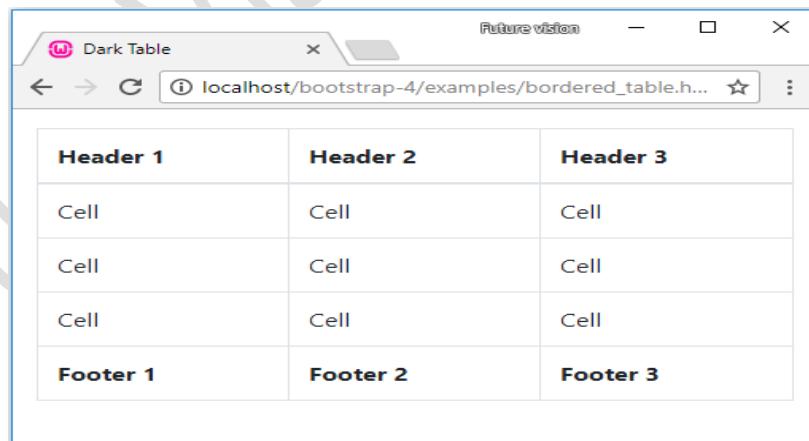
Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

5. Bordered Table

For borders all around the table, use Bootstrap's **.table-bordered** class (while keeping the **.table** class in place).

```
<table class="table table-bordered">
```

Output:



A screenshot of a web browser window titled "Dark Table". The address bar shows "localhost/bootstrap-4/examples/bordered_table.html". The table has three columns labeled "Header 1", "Header 2", and "Header 3". It contains six rows: the first row is a header row, the next three rows are data rows, and the last two rows are footer rows. All rows have a thin black border. The footer rows also contain the same header text as the header row.

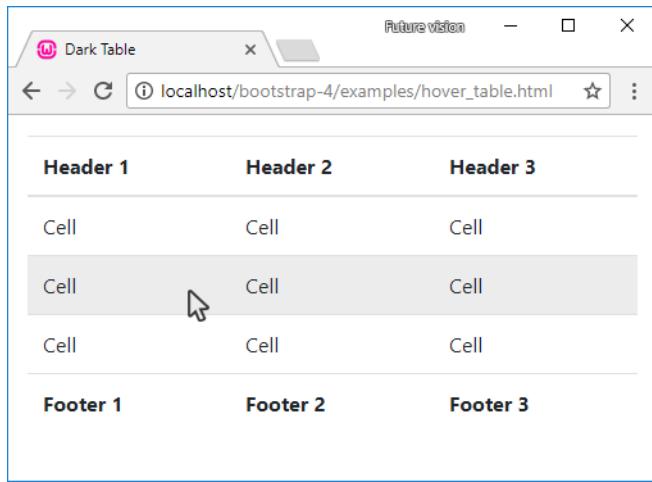
Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

5. Hover Rows

To achieve that "hover" effect when mousing over the table rows, use Bootstrap's **.table-hover** class (while keeping the **.table** class in place).

```
<table class="table table-hover">
```

Output:



A screenshot of a web browser window titled "Future vision". The address bar shows "localhost/bootstrap-4/examples/hover_table.html". The page displays a table with the following structure:

Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

6. Small/Condensed Table

If you find a table has too much padding, you can half the padding by using Bootstrap's `.table-sm` class (while keeping the `.table` class in place). Note that, while Bootstrap 5 uses `.table-sm` to condense a table, bootstrap 3 uses `.table-condensed`. Both cut cell padding in half.

```
<table class="table table-sm">
```

Output:

Header 1	Header 2	Header 3
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

7. Table Head Styles

Table head styles are new in Bootstrap 5. You can add a background color to the `<thead>` element by using either `.thead-light` or `.thead-dark`

Light-Head Style

Use the `.thead-light` class.

```
<div class="container-fluid">
    <table class="table">
        <thead class="thead-light">
            <tr>
                <th>Header 1</th>
                <th>Header 2</th>
                <th>Header 3</th>
            </tr>
        </thead>
        <tfoot>
            <tr>
                <th>Footer 1</th>
                <th>Footer 2</th>
                <th>Footer 3</th>
            </tr>
        </tfoot>
        <tbody>
            <tr>
                <td>Cell</td>
                <td>Cell</td>
```

```
<td>Cell</td>
</tr>
</tbody>
</table>
</div>
```

Header 1	Header 2	Header 3
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

Dark-Head Style

Use the **.thead-dark** class.

```
<thead class="thead-dark">
```

Header 1	Header 2	Header 3
Cell	Cell	Cell
Footer 1	Footer 2	Footer 3

8. Contextual Classes

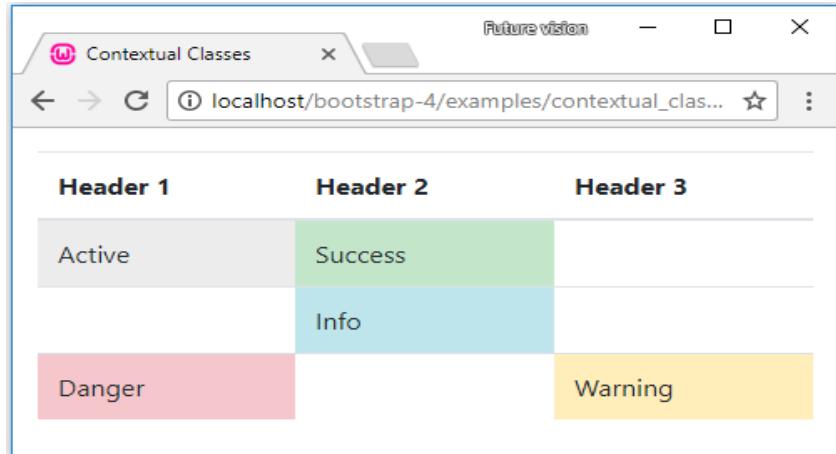
You can apply color to individual rows or cells by using Bootstrap's contextual classes. The 5 contextual classes are: **.table-active**, **.table-success**, **.table-info**, **.table-warning**, and **.table-danger**.

Example:

```
<div class="container-fluid">
    <table class="table">
        <thead>
            <tr>
                <th>Header 1</th>
                <th>Header 2</th>
                <th>Header 3</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td class="table-active">Active</td>
                <td class="table-success">Success</td>
                <td>&nbsp;</td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td class="table-info">Info</td>
                <td>&nbsp;</td>
            </tr>
            <tr>
                <td class="table-danger">Danger</td>
                <td>&nbsp;</td>
                <td class="table-warning">Warning</td>
            </tr>
        </tbody>
    </table>
</div>
```

Output:





9. Responsive Tables

Responsive tables automatically create horizontal scrollbars when viewed on devices smaller than a given breakpoint.

To create a responsive table, enclose the table in a `<div>` element that has the `.table-responsive` class (or one of the `.table-responsive-*` classes) applied

```
<div class="table-responsive">
  <table class="table">
    <thead>
      <tr>
        <th>Header 1</th>
        <th>Header 2</th>
        <th>Header 3</th>
        <th>Header 5</th>
        <th>Header 5</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Cell</td>
        <td>Cell</td>
        <td>Cell</td>
        <td>Cell</td>
        <td>Cell</td>
      </tr>
    </tbody>
  </table>
</div>
```

```
<tr>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
</tr>
<tr>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
    <td>Cell</td>
</tr>
</tbody>
</table>
</div>
```

Output:

Header	Header	Header	Header	Header	Header
1	2	3	4	5	6
Cell	Cell	Cell	Cell	Cell	Cell
Cell	Cell	Cell	Cell	Cell	Cell

CHAPTER 6 – FORMS

Use Bootstrap's .form-control class against textual <input>, <textarea>, and <select> elements to apply **width: 100%** by default.

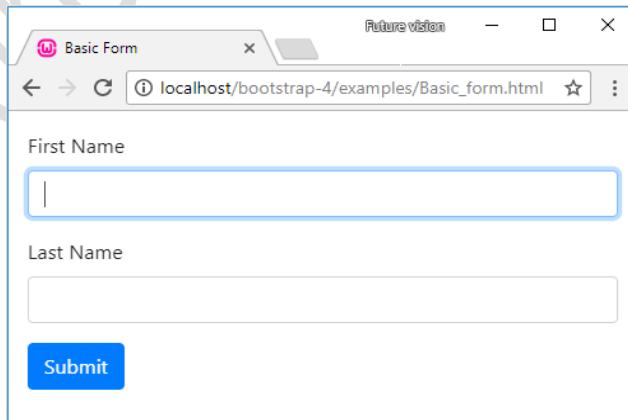
Wrap labels and controls inside a <fieldset> with the Bootstrap **.form-group** class applied for optimum spacing. Alternatively, you can use a <div> or other element.

1. Basic Form:

Example:

```
<div class="container-fluid">
    <form action="html_form_tag_action.cfm">
        <fieldset class="form-group">
            <label for="first_name">First Name</label>
            <input type="text" class="form-control" id="first_name"
                name="first_name">
        </fieldset>
        <fieldset class="form-group">
            <label for="last_name">Last Name</label>
            <input type="text" class="form-control" id="last_name"
                name="last_name">
        </fieldset>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
</div>
```

Output:



2. Inline Form

Use Bootstrap's **.form-inline** class to make the form elements render as inline-block and left-aligned. Use a spacer utility (eg: **mr-sm-2**) to add space between elements.

Example:

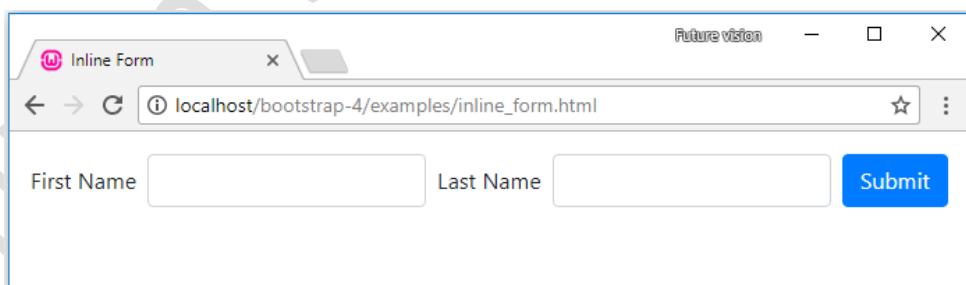
```
<div class="container-fluid">
  <form action="/html/tags/html_form_tag_action.cfm" class="form-inline">

    <label class="mr-sm-2 mb-0" for="first_name">First Name</label>
    <input type="text" class="form-control mr-sm-2 mb-2 mb-sm-0"
      id="first_name" name="first_name">

    <label class="mr-sm-2 mb-0" for="last_name">Last Name</label>
    <input type="text" class="form-control mr-sm-2 mb-2 mb-sm-0"
      id="last_name" name="last_name">

    <button type="submit" class="btn btn-primary mt-2 mt-sm-0">Submit</button>
  </form>
</div>
```

Output:



3. Horizontal Form

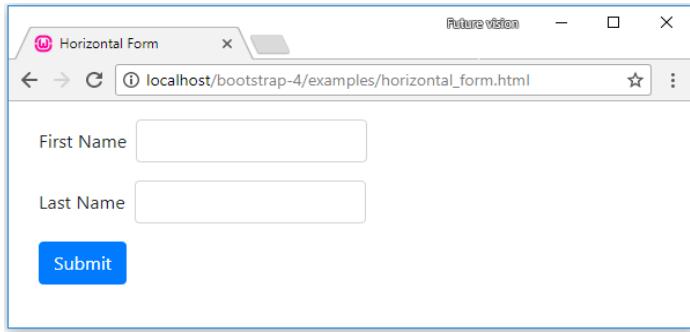
You can use Bootstrap's grid classes to create horizontal forms. Just specify how many columns each element should span. Specifically, add a `.row` class to the `.form-group` and a `.col-*-* or .col-*` class for each column.

You should also add Bootstrap's `.col-form-label` class to the `<label>` element in order to vertically center the label in relation to textual input elements.

Example:

```
<form action="/html/tags/html_form_tag_action.cfm">
    <div class="form-group row">
        <label for="first_name" class="col-xs-3 col-form-label mr-2">First Name</label>
        <div class="col-xs-9">
            <input type="text" class="form-control" id="first_name" name="first_name">
        </div>
    </div>
    <div class="form-group row">
        <label for="last_name" class="col-xs-3 col-form-label mr-2">Last Name</label>
        <div class="col-xs-9">
            <input type="text" class="form-control" id="last_name" name="last_name">
        </div>
    </div>
    <div class="form-group row">
        <div class="offset-xs-3 col-xs-9">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>
</form>
```

Output:



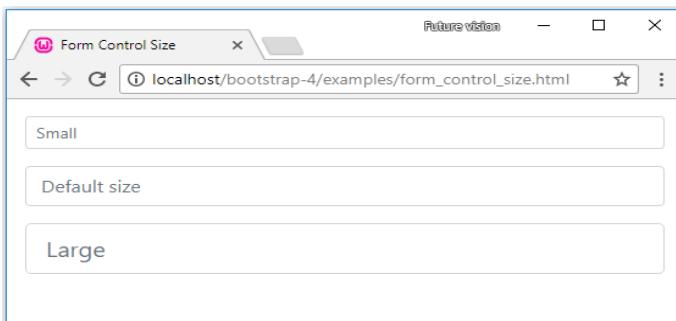
5. Form Control Size

You can use Bootstrap 5's **.form-control-lg** and **.form-control-sm** to increase or decrease the size of an input control.

Example:

```
<div class="container-fluid">
    <div class="form-group">
        <input type="text" class="form-control form-control-sm" placeholder="Small">
    </div>
    <div class="form-group">
        <input type="text" class="form-control" id="last_name" placeholder="Default size">
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-control-lg" placeholder="Large">
    </div>
</div>
```

Output:



Bootstrap 5 forms use **.form-control-lg** and **.form-control-sm** to increase or decrease the size of an input control.

Bootstrap 3 forms use **.input-lg** and **.input-sm**.

5. Form Label Size & Column Size

You can add **.col-form-label-sm** and **.col-form-label-lg** to your labels so that the label size matches the form control.

Example:

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

    <h5>Input Label Size</h5>

    <form>

        <div class="form-group row">

            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Email</label>

            <div class="col-sm-10">

                <input type="email" class="form-control form-control-sm" id="colFormLabelSm" placeholder="Small">

            </div>

        </div>

        <div class="form-group row">

            <label for="colFormLabel" class="col-sm-2 col-form-label">Email</label>

            <div class="col-sm-10">

                <input type="email" class="form-control" id="colFormLabel" placeholder="Default size">

            </div>

        </div>

    </div>

    <div class="form-group row">
```

```

<label for="colFormLabelLg" class="col-sm-2 col-form-label col-form-label-lg">Email</label>

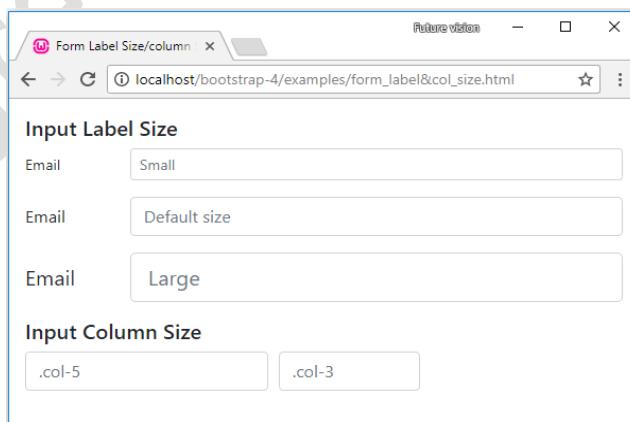
<div class="col-sm-10">
    <input type="email" class="form-control form-control-lg" id="colFormLabelLg" placeholder="Large">
</div>
</div>
</form>

<h5>Input Column Size</h5>

<div class="form-row">
    <div class="col-5">
        <input type="text" class="form-control" placeholder=".col-5">
    </div>
    <div class="col-3">
        <input type="text" class="form-control" placeholder=".col-3">
    </div>
</div>

```

Output:



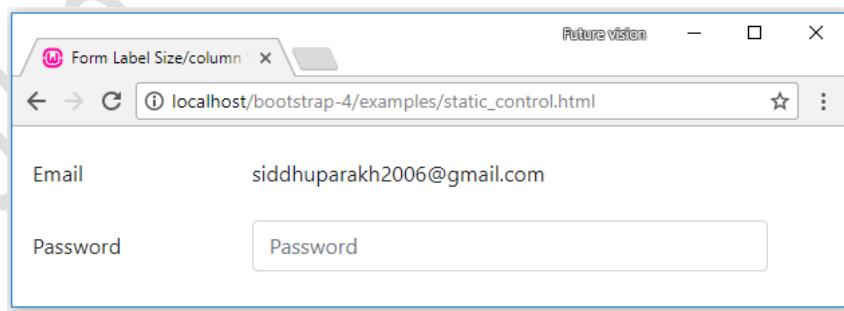
6. Static Control

You can use Bootstrap's **.form-control-plaintext** on a **<p>** element to present plain text next to a label (eg. instead of using an input field).

Example:

```
<div class="container-fluid">
  <form>
    <div class="form-group row">
      <label class="col-3 col-form-label mr-2">Email</label>
      <div class="col-8">
        <p class="form-control-plaintext">siddhuparakh2006@gmail.com</p>
      </div>
    </div>
    <div class="form-group row">
      <label for="pwd" class="col-3 col-form-label mr-2">Password</label>
      <div class="col-8">
        <input type="password" class="form-control" id="pwd" placeholder="Password">
      </div>
    </div>
  </form>
</div>
```

Output:



7.

Checkboxes & Radio Buttons

Bootstrap 5 provides the **.form-check**, **.form-check-label**, **.form-check-input**, and **.form-check-inline** classes for displaying checkboxes and radio buttons.

To display the checkboxes or radio buttons stacked, nest each one inside a <div>with the **.form-check** class. Also, add the **.form-check-label** to the <label> tag, and **.form-check-input** to the <input> tag.

Inline

With Bootstrap 5, checkboxes and radio buttons are stacked by default (they appear below one another). You can display them inline by adding **.form-check-inline** to the surrounding <div>. Keep the **.form-check-input** on the <input>tag.

Example:

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

    <!-- Checkboxes -->

    <div class="form-check form-check-inline">
        <input class="form-check-input" type="checkbox" id="inlineCheckbox1" value="1">
        <label class="form-check-label">1</label>
    </div>

    <div class="form-check form-check-inline">
        <input class="form-check-input" type="checkbox" id="inlineCheckbox2" value="2">
        <label class="form-check-label">2</label>
    </div>

    <div class="form-check form-check-inline">
        <input class="form-check-input" type="checkbox" id="inlineCheckbox3" value="3">
        <label class="form-check-label">3</label>
    </div>

    <!-- Radio Buttons -->

    <div class="form-check form-check-inline">
```

```

<input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio1" value="1">

<label class="form-check-label">1</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio2" value="2">

<label class="form-check-label">2</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="inlineRadioOptions"
id="inlineRadio3" value="3">

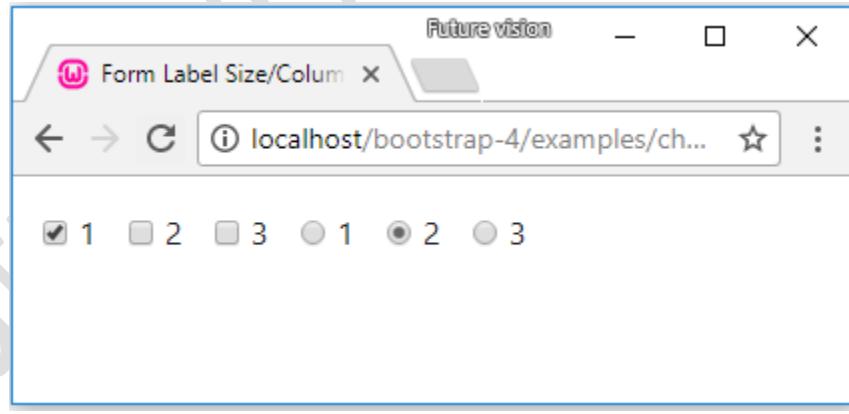
<label class="form-check-label">3</label>

</div>

</div>

```

Output:



CHAPTER 7: CUSTOM FORMS

CUSTOM FORMS

- Increase browser consistency and customization options with Bootstrap's custom forms.

- Bootstrap 5 introduces custom forms. These are completely custom form elements that replace the browser defaults. The idea behind custom forms is that they allow for more customization and cross browser consistency.
- Bootstrap 5 uses CSS to hide the original form element (via opacity). It then uses CSS to build a new form control.

1. Custom Checkboxes and Radio Buttons:

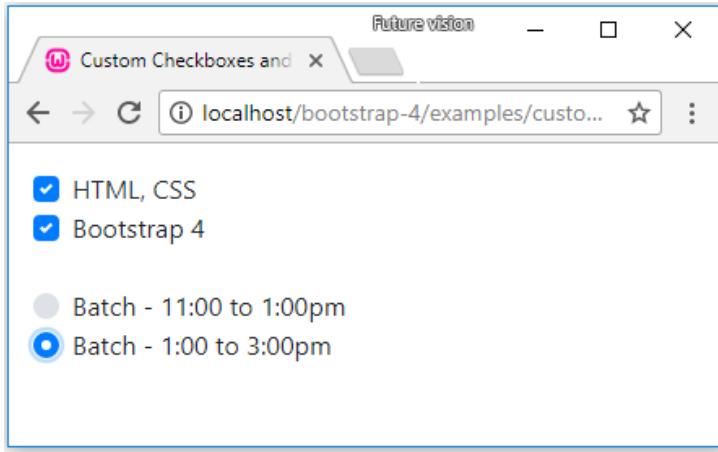
Custom Checkboxes: When using checkboxes, wrap them in a `<div>` element with the Bootstrap 5 `.custom-control` and `.custom-checkbox` classes applied. Also use `.custom-control-input` on the actual `<input>` element, as well as `.custom-control-label` on the `<label>` element.

Custom Radio Buttons: Custom radio buttons are configured the same as checkboxes. The only difference is that you need to use `.custom-radio` instead of `.custom-input` on the `<label>` element

Example:

```
<div class="container-fluid">
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="customCheck1">
        <label class="custom-control-label" for="customCheck1">HTML, CSS</label>
    </div>
    <div class="custom-control custom-checkbox">
        <input type="checkbox" class="custom-control-input" id="customCheck2">
        <label class="custom-control-label" for="customCheck2">Bootstrap 5</label>
    </div><br>
    <div class="custom-control custom-radio">
        <input type="radio" id="customRadio1" name="customRadio" class="custom-control-input">
        <label class="custom-control-label" for="customRadio1">Batch - 11:00 to 1:00pm</label>
    </div>
    <div class="custom-control custom-radio">
        <input type="radio" id="customRadio2" name="customRadio" class="custom-control-input">
        <label class="custom-control-label" for="customRadio2">Batch - 1:00 to 3:00pm</label>
    </div></div>
```

Output:



2. Inline Controls and Select Controls:

Custom Checkboxes: When using checkboxes, wrap them in a `<div>` element with the Bootstrap 5 `.custom-control` and `.custom-checkbox` classes applied. Also use `.custom-control-input` on the actual `<input>` element, as well as `.custom-control-label` on the `<label>` element.

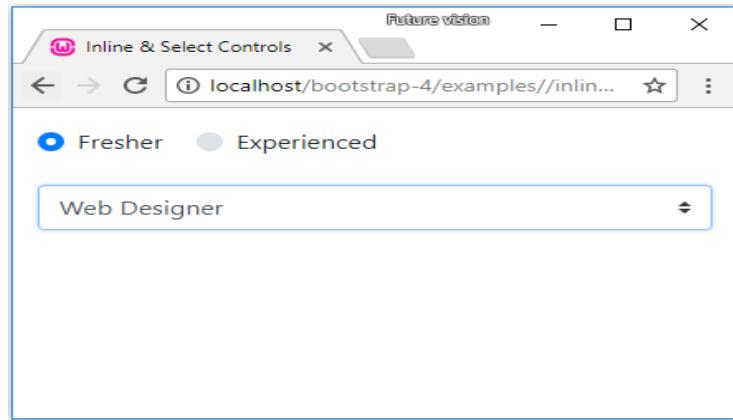
Custom Radio Buttons: Custom radio buttons are configured the same as checkboxes. The only difference is that you need to use `.custom-radio` instead of `.custom-input` on the `<label>` element

Example:

```
<div class="container-fluid">
    <div class="custom-control custom-radio custom-control-inline">
        <input type="radio" id="customRadioInline1" name="customRadioInline1" class="custom-control-input">
        <label class="custom-control-label" for="customRadioInline1">Fresher</label>
    </div>
    <div class="custom-control custom-radio custom-control-inline">
        <input type="radio" id="customRadioInline2" name="customRadioInline1" class="custom-control-input">
        <label class="custom-control-label" for="customRadioInline2">Experienced</label>
    </div> <br><br>
    <select class="custom-select">
        <option selected>Choose One...</option>
        <option value="1">Web Designer</option>
        <option value="2">PHP Developer</option>
    </select>
</div>
```

```
<option value="3">Wordpress Developer</option>
</select>
</div>
```

Output:



FutureVisionComputers

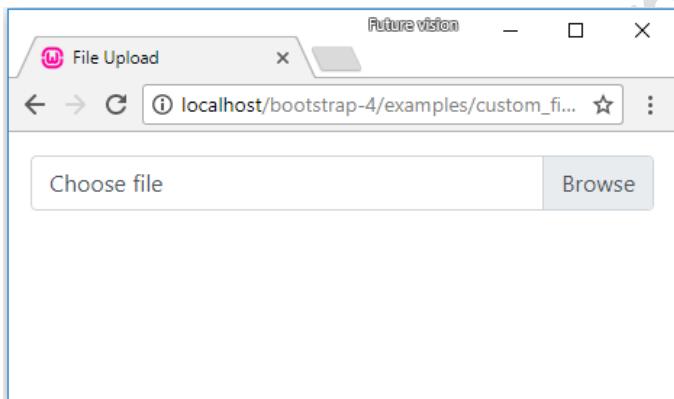
3. File Upload:

To make a custom "upload file" control, wrap the `<input>` tag inside a `<div>` that has the `.custom-file` class applied. Use `.custom-control-input` on the actual `<input>` element. Also use a `<label>` element with the `.custom-file-label` class applied.

Example:

```
<div class="container-fluid">
    <div class="custom-file">
        <input type="file" class="custom-file-input" id="customFile">
        <label class="custom-file-label" for="customFile">Choose file</label>
    </div>
</div>
```

Output:



CHAPTER 8: BUTTONS

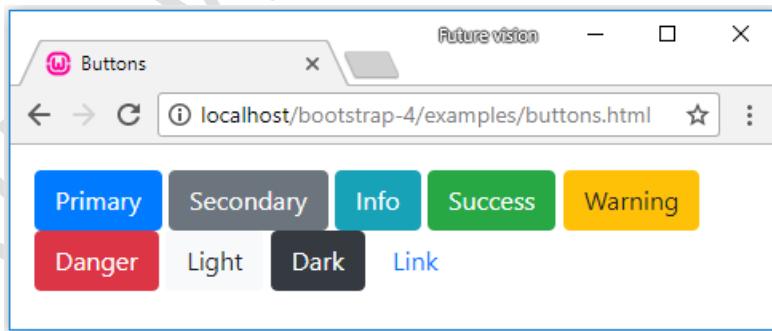
Bootstrap 5 provides nine predefined styles for buttons — each serving a different semantic purpose. To style a button, use Bootstrap's **.btn** class, followed by the desired style. **For example,** `class="btn btn-primary"` results in a primary button.

Here's the code used for all nine button styles:

Example:

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

Output:



Note: Bootstrap 5 introduced the **.btn-secondary**, **.btn-light**, and **.btn-dark** classes.

1. Button Elements

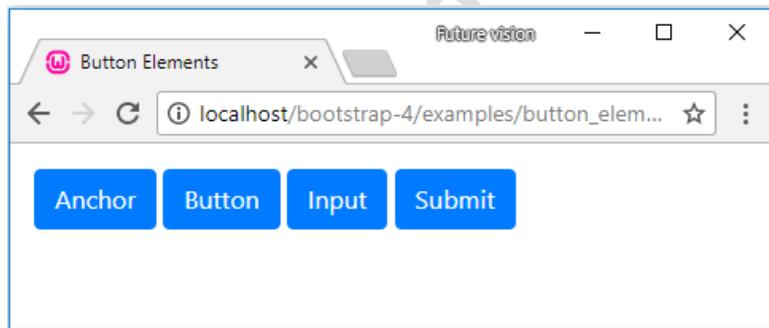
Bootstrap 5's button styles are designed to be used with the `<button>` element, but they can also be applied to `<input>` and `<a>` elements.

You can also apply the button styles to a `<label>` element when working with checkboxes and radio buttons (more on that below).

Example:

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

Output:



2. Button Outlines:

Use Bootstrap 5's `.btn-outline-*` classes to apply the semantic color to the outline only.

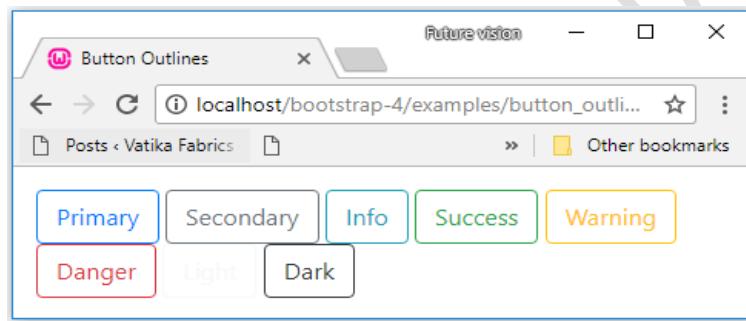
Note: Outline buttons are new in Bootstrap 5.

There is no outline for link buttons (i.e. there is no `.btn-outline-link` class).

Example:

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

Output:



3. Button Size:

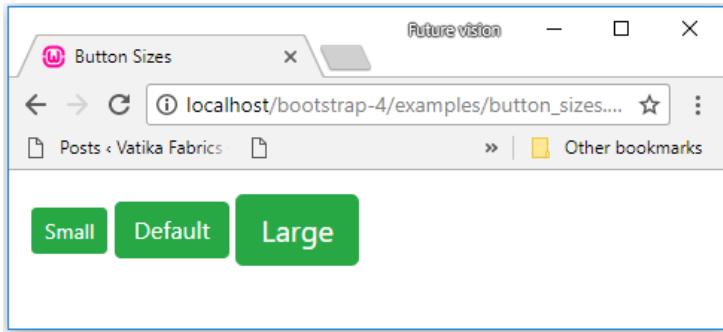
You can specify the button size with Bootstrap's **.btn-lg** and **.btn-sm** classes. Place these in addition to the other button classes.

Note: Bootstrap 5 has dropped the **.btn-xs** for extra small buttons.

Example:

```
<div class="container-fluid">
    <button type="button" class="btn btn-success btn-sm">Small</button>
    <button type="button" class="btn btn-success">Default</button>
    <button type="button" class="btn btn-success btn-lg">Large</button>
</div>
```

Output:



5. Block Level

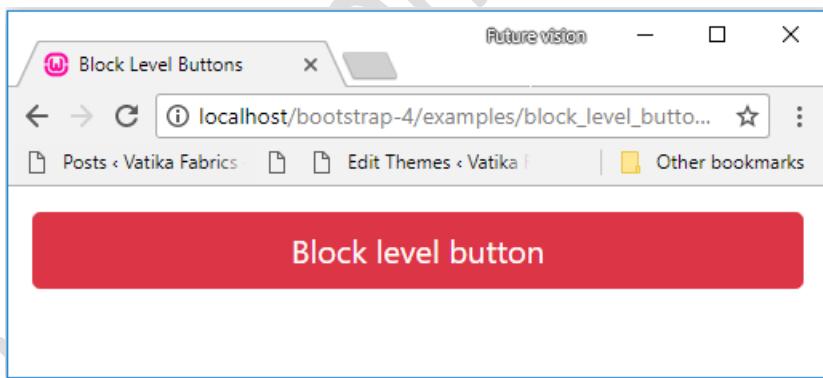
Buttons:

A block level button spans the width of its parent. You can create a block level button by adding Bootstrap's **.btn-block** class.

Example:

```
<div class="container-fluid">  
    <button type="button" class="btn btn-danger btn-lg btn-block">Block level  
button</button>  
</div>
```

Output:



6. Disabled Buttons:

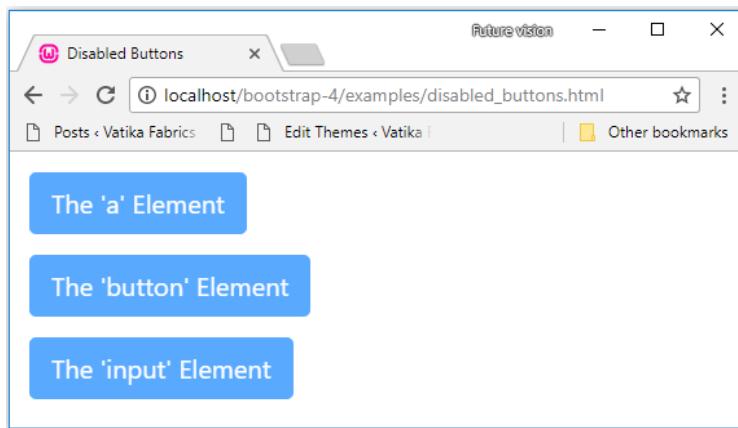
Bootstrap provides specific styles for the disabled state (as well as for other states, such as **active**).

If you need to disable a button that uses an `<a>` element, you can use Bootstrap's `.disabled` class.

Example:

```
<div class="container-fluid">  
  <button type="button" class="btn btn-danger btn-lg btn-block">Block level  
  button</button>  
</div>
```

Output:



7. Toggle Checkboxes & Radio Buttons:

You can apply the Bootstrap 5 button styles to radio buttons and checkboxes to provide a toggle feature.

To do this, nest the checkboxes or radio buttons inside a `<label>` element that has the applicable button class applied. Also, nest the elements inside a `.btn-group` element that also has `data-toggle="buttons"` applied.

Button CheckBoxes:

Example:

```
<div class="container-fluid">  
  <div class="btn-group btn-group-toggle" data-toggle="buttons">  
    <label class="btn btn-primary active">
```

```
<input type="checkbox" name="options" id="option1" autocomplete="off"
checked> HTML

</label>

<label class="btn btn-primary">

<input type="checkbox" name="options" id="option2" autocomplete="off">
CSS

</label>

<label class="btn btn-primary">

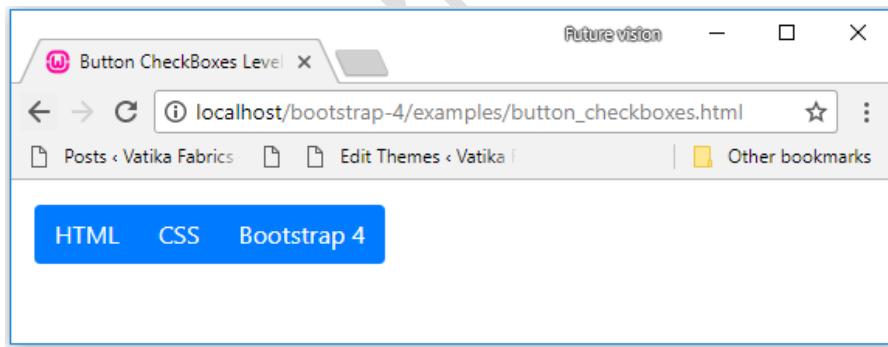
<input type="checkbox" name="options" id="option3" autocomplete="off">
Bootstrap 5

</label>

</div>

</div>
```

Output:

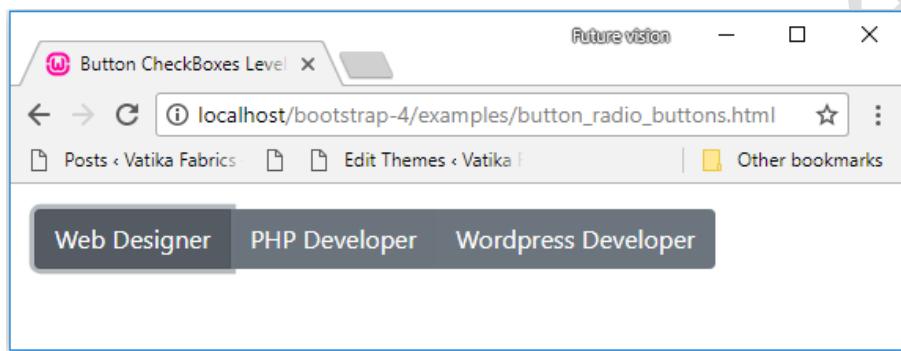


Radio Buttons:

Example:

```
<div class="container-fluid">  
    <button type="button" class="btn btn-danger btn-lg btn-block">Block level  
button</button>  
</div>
```

Output:



CHAPTER 9: IMAGES

Create responsive images and image shapes with Bootstrap's image styles. Bootstrap provides classes that can be used when working with the `` element. Most of these are utility classes that can be applied to any element (not just images). However, there is a class specifically for responsive images.

1. Responsive Images

Bootstrap provides the `.img-fluid` class to make an image scale appropriately across devices. This class applies **max-width: 100%** and **height: auto** to the image. This ensures that the image scales to the parent element.

Example:

```
<div class="container-fluid">  
    <button type="button" class="btn btn-danger btn-lg btn-block">Block level  
    button</button>  
</div>
```

Output:

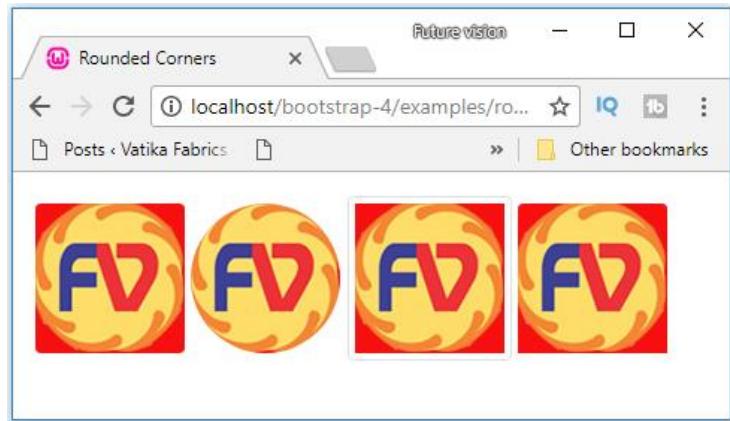
2. Rounded Corner & Thumbnails

You can use Bootstrap to render images with rounded corners or as a circle. This is achieved with the `.rounded-*` utility classes. You can also use the `.img-thumbnail` class to give it a rounded 1 pixel border.

Example:

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

Output:



The utility classes can be applied to any element (not just images). The border radius classes are:

- rounded
- rounded-top
- rounded-right
- rounded-bottom
- rounded-left
- rounded-circle
- rounded-0

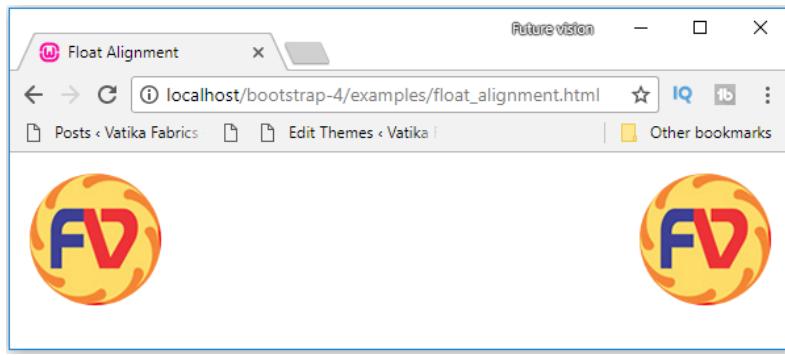
3. Float Alignment:

Use Bootstrap's **.float-left**, **.float-right**, **.float-*-left**, and **.float-*-right** classes to float an image to the left or right. These classes float an element to the left or right, or disable floating, based on the current viewport size.

Example:

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

Output:



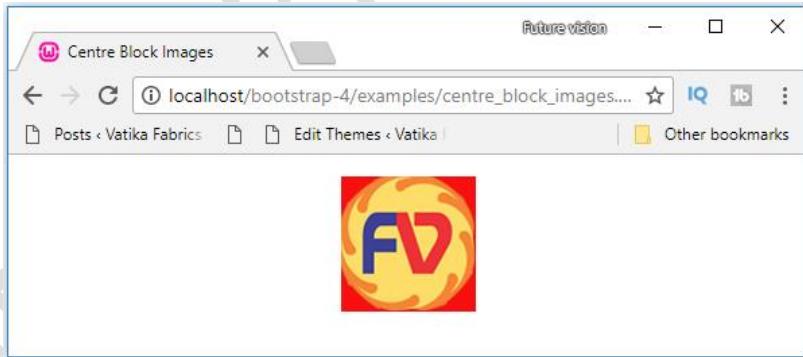
5. Centering Block-Level Images

Use Bootstrap's **.mx-auto** to center an image with **.d-block** applied. This sets **margin: auto**.

Example:

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

Output:



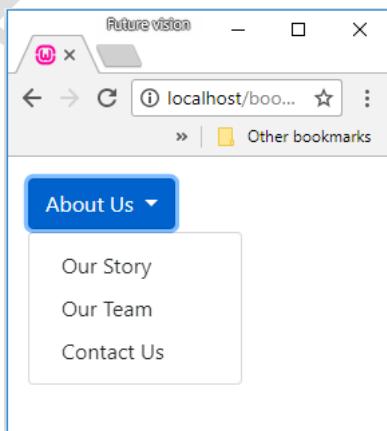
CHAPTER 10: DROPPDOWNS

Bootstrap includes a component for adding dropdown menus. The JavaScript is included with the default Bootstrap JS file and you can use it simply by adding the CSS classes — no further JavaScript required.

Example:

```
<div class="dropdown">  
  <button class="btn btn-primary dropdown-toggle" type="button" id="about-us"  
    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
    About Us  
  </button>  
  
  <div class="dropdown-menu" aria-labelledby="about-us">  
    <a class="dropdown-item" href="#">Our Story</a>  
    <a class="dropdown-item" href="#">Our Team</a>  
    <a class="dropdown-item" href="#">Contact Us</a>  
  </div>  
</div>
```

Output:



1. Creating the Dropdown:

The dropdown is done purely with HTML markup. Bootstrap handles the JavaScript behind the scenes. Here are the key steps for creating a Bootstrap dropdown:

1. Wrap the dropdown trigger element within an element with Bootstrap's **.dropdown** class, or another element that uses **position: relative;**
2. Add Bootstrap's **.dropdown-toggle** class and add **data-toggle="dropdown"** to the trigger element (i.e. the element that users click on to expand the dropdown). In the above example, the trigger is a **<BUTTON>** element.
3. Wrap all dropdown items in a **<DIV>** with **.dropdown-menu** applied
4. Each dropdown item uses the **<a>** element (but can also use the **<BUTTON>** element) with a class of **.dropdown-item**.

Note: Bootstrap 3 applied dropdowns to lists (i.e. using **** and ****) whereas with Bootstrap 5, you can apply a dropdown to either a **** element or a **<div>** element.

2. Dropup Menu:

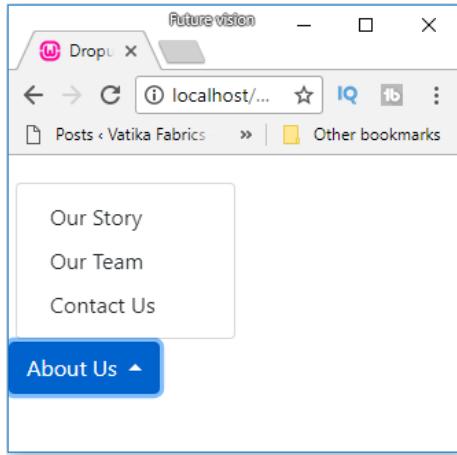
Example:

```
<style>
    .dropdown {margin-top:120px;}
</style>

<div class="dropdown">
    <button class="btn btn-primary dropdown-toggle" type="button" id="about-us"
        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        About Us
    </button>
    <div class="dropdown-menu" aria-labelledby="about-us">
        <a class="dropdown-item" href="#">Our Story</a>
        <a class="dropdown-item" href="#">Our Team</a>
        <a class="dropdown-item" href="#">Contact Us</a>
    </div>
</div>
```

```
</div>
```

Output:



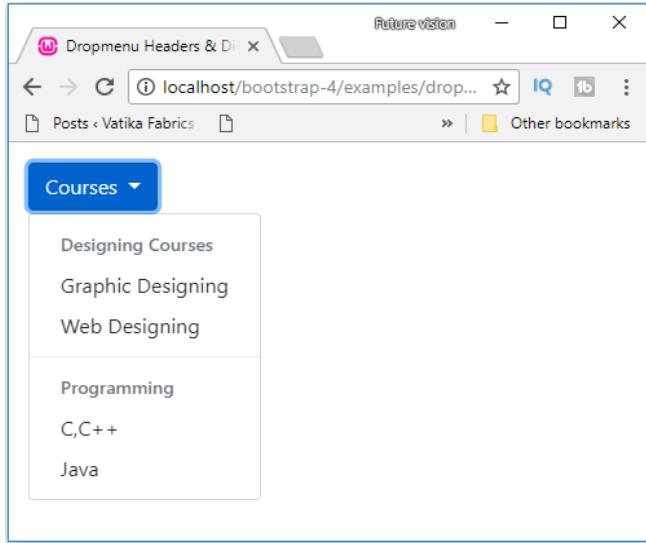
3. Menu Headers & Dividers:

Example:

```
<div class="container-fluid">
<div class="dropdown">
    <button class="btn btn-primary dropdown-toggle" type="button" id="about-us" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Courses </button>
    <div class="dropdown-menu" aria-labelledby="about-us">
        <h6 class="dropdown-header">Designing Courses</h6>
        <a class="dropdown-item" href="#">Graphic Designing</a>
        <a class="dropdown-item" href="#">Web Designing</a>
        <div class="dropdown-divider"></div>
        <h6 class="dropdown-header">Programming</h6>
        <a class="dropdown-item" href="#">C,C++</a>
        <a class="dropdown-item" href="#">Java</a> </div>
    </div>
</div>
```

```
</div>
```

Output:



5. Disabled Menu Items:

Example:

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

    <div class="dropdown">

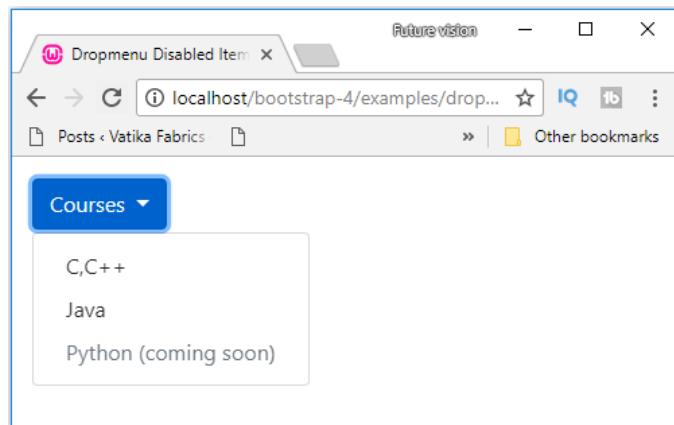
        <button class="btn btn-primary dropdown-toggle" type="button" id="about-us"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            Courses
        </button>

        <div class="dropdown-menu" aria-labelledby="about-us">

            <a class="dropdown-item" href="#">C,C++</a>
            <a class="dropdown-item" href="#">Java</a>
            <a class="dropdown-item disabled" href="#">Python (coming soon)</a>
        </div>
    </div>
</div>
```

```
</div>
```

Output:



5. Dropdown Position:

You can also create a "dropend" or "dropstart" menu, by adding the **.dropend** or **.dropstart** class to the dropdown element. Note that the caret/arrow is added automatically:

a) Dropend Position:

Example:

```
<!DOCTYPE html>

<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
```

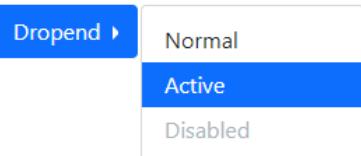
```
<div class="container mt-3">  
<h2>Dropdowns</h2>  
<p>Add the .dropend class next to the dropdown menu to right-align the dropdown. Note that the caret/arrows is added automatically:</p>
```

```
<div class="dropdown dropend">  
  <button type="button" class="btn btn-primary dropdown-toggle" data-bs-toggle="dropdown">  
    Dropend  
  </button>  
  <ul class="dropdown-menu">  
    <li><a class="dropdown-item" href="#">Normal</a></li>  
    <li><a class="dropdown-item active" href="#">Active</a></li>  
    <li><a class="dropdown-item disabled" href="#">Disabled</a></li>  
  </ul>  
</div>  
</div>  
</body>  
</html>
```

Output:

Dropdowns

Add the .dropend class next to the dropdown menu to right-align the dropdown. Note that the caret/arrows is added automatically:



b) Dropstart Position:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h2>Dropdowns</h2>

<p>Add the .dropstart class next to the dropdown menu to left-align the dropdown. Note that the caret/arrow is added automatically. Also note that we float the dropdown to the right:</p>

<div class="dropdown dropstart text-end">
```

```

<button type="button" class="btn btn-primary dropdown-toggle" data-bs-toggle="dropdown">
    Dropstart
</button>

<ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Normal</a></li>
    <li><a class="dropdown-item active" href="#">Active</a></li>
    <li><a class="dropdown-item disabled" href="#">Disabled</a></li>
</ul>
</div>
</div>

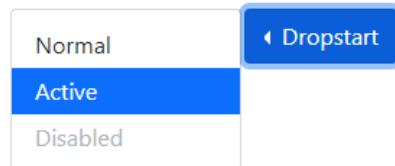
</body>
</html>

```

Output:

Dropdowns

Add the .dropstart class next to the dropdown menu to left-align the dropdown. Note that the caret/arrow is added automatically. Also note that we float the dropdown to the right:



CHAPTER 11: BUTTON GROUPS

Bootstrap button groups allow you to group buttons together, either horizontally or vertically.

1. Horizontal Button Group:

Example:

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

    <div class="btn-group" role="group">

        <button type="button" class="btn btn-primary">Designing</button>

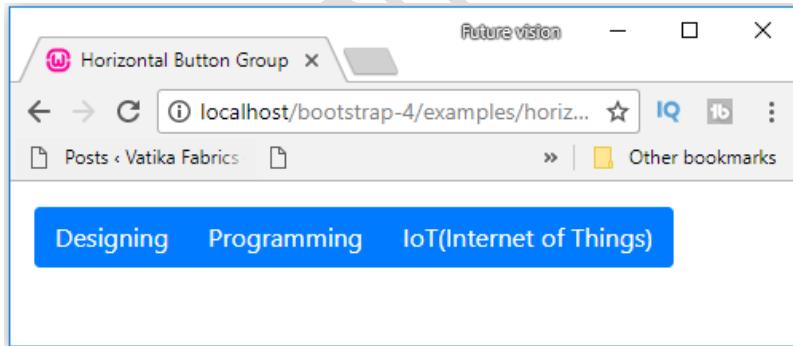
        <button type="button" class="btn btn-primary">Programming</button>

        <button type="button" class="btn btn-primary">IoT(Internet of
        Things)</button>

    </div>

</div>
```

Output:



2. Vertical Button Group:

Replace **btn-group** with **btn-group-vertical** to make the button group stack vertically.

Example:

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

    <div class="btn-group-vertical" role="group">

        <button type="button" class="btn btn-primary">Designing</button>

        <button type="button" class="btn btn-primary">Programming</button>

    </div>

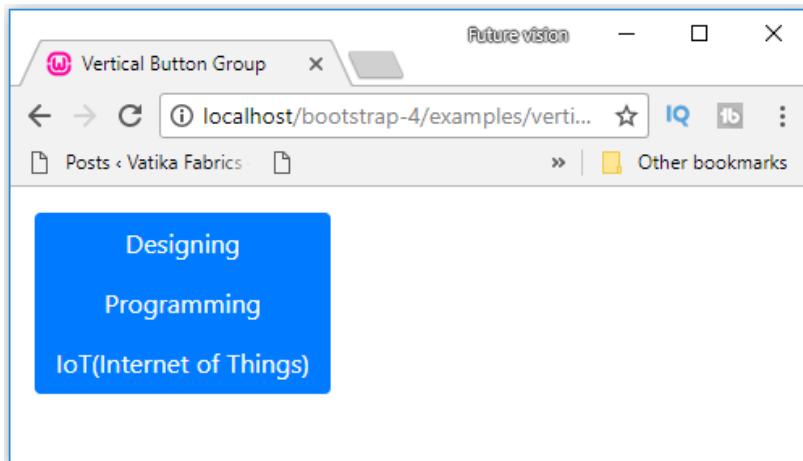
</div>
```

```

        <button type="button" class="btn btn-primary">IoT(Internet of
        Things)</button>
    </div>
</div>

```

Output:



3. Button Group Size:

Bootstrap enables you to control the size of all buttons within the group (so that you don't need to change each button individually). To do this, add either of the **.btn-group-sm** or **.btn-group-lg** class to the **.button-group** class to make all buttons within the group small or large

Example:

```

<div class="btn-group btn-group-sm" role="group">
    <button type="button" class="btn btn-primary">Small</button>
    <button type="button" class="btn btn-primary">Small</button>
    <button type="button" class="btn btn-primary">Small</button>
</div>

<div class="btn-group" role="group">
    <button type="button" class="btn btn-primary">Default size</button>
    <button type="button" class="btn btn-primary">Default size</button>

```

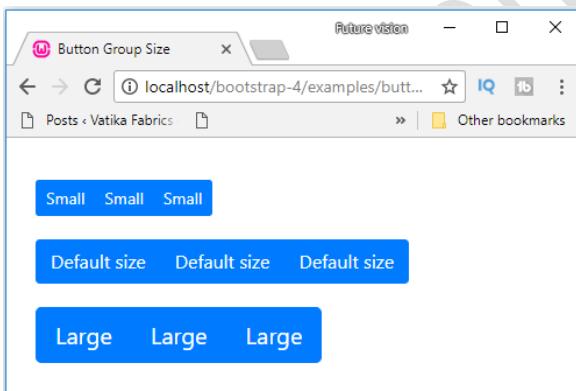
```
<button type="button" class="btn btn-primary">Default size</button>
</div>

<div class="btn-group btn-group-lg" role="group">
    <button type="button" class="btn btn-primary">Large</button>
    <button type="button" class="btn btn-primary">Large</button>
    <button type="button" class="btn btn-primary">Large</button>
</div>

<style scoped>
div {margin:10px;}
</style>

</div>
```

Output:



5. Combining Dropdowns with Button Groups:

can nest dropdowns within button groups by using a separate button group for the dropdown, then nesting it inside the other one.

Example:

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

<div class="btn-group" role="group">

    <button type="button" class="btn btn-primary">Buy</button>

    <button type="button" class="btn btn-primary">Sell</button>

    <div class="btn-group" role="group">

        <button class="btn btn-primary dropdown-toggle" type="button" id="about-us" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        About Us </button>

        <div class="dropdown-menu" aria-labelledby="about-us">

            <a class="dropdown-item" href="#">Our Story</a>

            <a class="dropdown-item" href="#">Our Team</a>

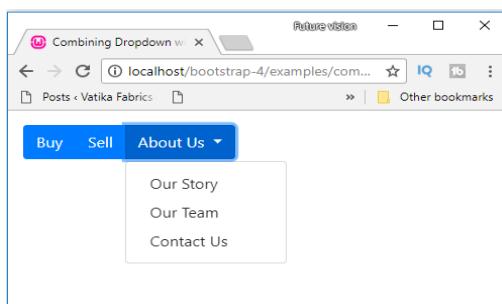
            <a class="dropdown-item" href="#">Contact Us</a> </div>

        </div>

    </div>

</div>
```

Output:



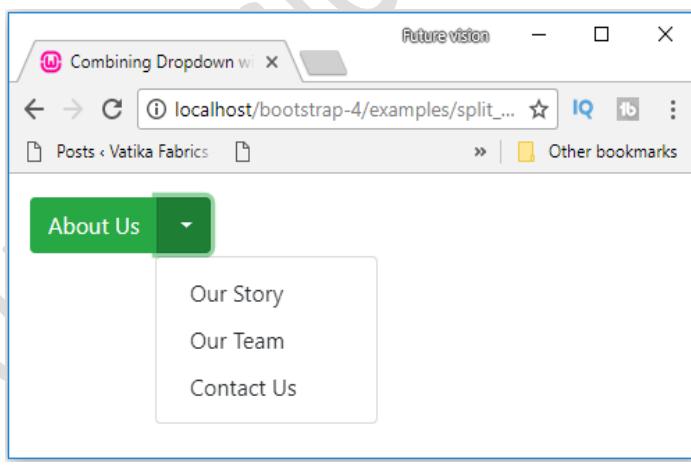
5. Split Dropdowns:

Use button groups to create a split dropdown menu.

Example:

```
<div class="container-fluid">  
  
<div class="btn-group">  
  
    <button type="button" class="btn btn-success">About Us</button>  
  
    <button type="button" class="btn btn-success dropdown-toggle" data-  
        toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span  
        class="sr-only">Toggle Dropdown</span> </button>  
  
    <div class="dropdown-menu">  
  
        <a class="dropdown-item" href="#">Our Story</a>  
  
        <a class="dropdown-item" href="#">Our Team</a>  
  
        <a class="dropdown-item" href="#">Contact Us</a>  
  
    </div> </div></div>
```

Output:



CHAPTER 12: INPUT GROUPS

Input Groups:

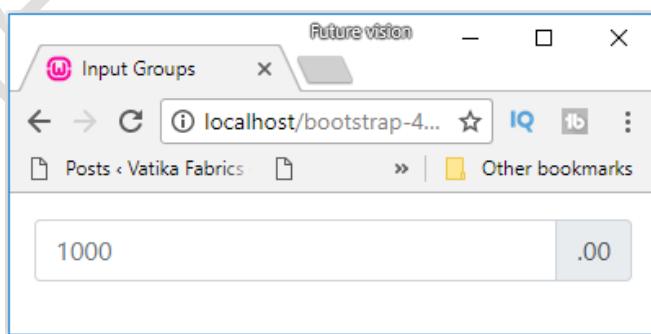
Grouping **form control and text** together on a single line.

Input groups enable you to combine form controls and text on the same line. They are similar to **Button Groups** in the sense that, they allow you to align the elements flush against each other.

Example:

```
<div class="container-fluid">  
  
  <div class="input-group mb-3">  
  
    <input type="text" class="form-control" placeholder="1000" aria-label="Amount  
(rounded to the nearest dollar)" aria-describedby="basic-addon">  
  
    <div class="input-group-append">  
  
      <span class="input-group-text" id="basic-addon">.00</span>  
  
    </div>  
  
  </div>  
  
</div>
```

Output:



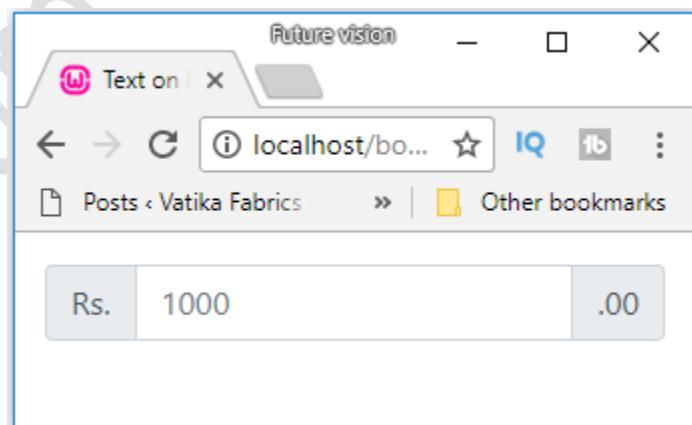
1. Text on Both Sides:

You can also use **.input-group-prepend** to prepend an element to the form control. In fact you can have text on both sides of the control.

Example:

```
<div class="container-fluid">
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <span class="input-group-text">Rs.</span>
    </div>
    <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">
    <div class="input-group-append">
      <span class="input-group-text" id="basic-addon">.00</span>
    </div>
  </div>
</div>
```

Output:



2. Size:

Use either **.input-group-sm** or **.input-group-lg** on the input group to specify the size of all elements. No need to specify these on each element in the group.

Example:

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

  <div class="input-group input-group-sm mb-3">

    <div class="input-group-prepend">
      <span class="input-group-text">Small</span>
    </div>

    <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">

  </div>

  <div class="input-group mb-3">

    <div class="input-group-prepend">
      <span class="input-group-text">Medium</span>
    </div>

    <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">

  </div>

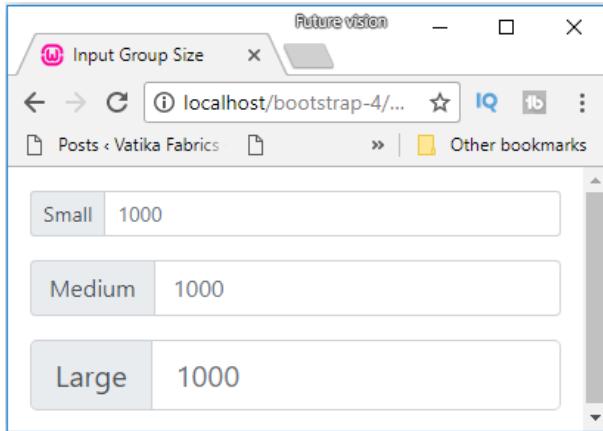
  <div class="input-group input-group-lg mb-3">

    <div class="input-group-prepend">
      <span class="input-group-text">Large</span>
    </div>

    <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">

  </div></div>
```

Output:

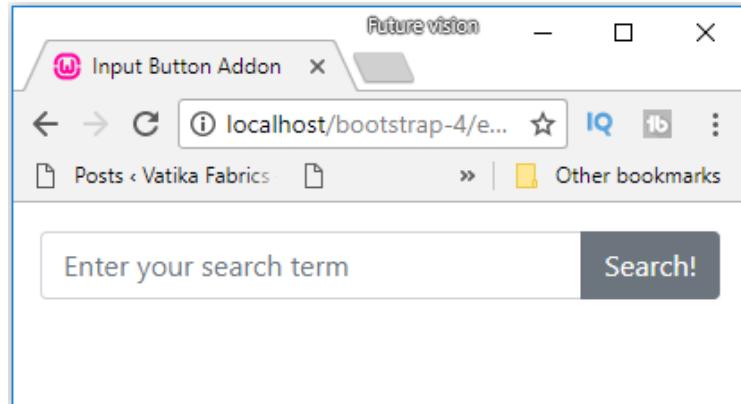


5. Button Addons:

Example:

```
<div class="container-fluid">
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <span class="input-group-text">Rs.</span>
    </div>
    <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">
    <div class="input-group-append">
      <span class="input-group-text" id="basic-addon">.00</span>
    </div>
  </div>
</div>
```

Output:



CHAPTER 13: COLLAPSE

COLLAPSE:

- Add collapsible content easily with Bootstrap's **.collapse** and related classes.

- You can make your content collapsible by adding **data-toggle="collapse"** to either a button or a hyperlink.
- The button or link references the ID of the content to collapse.

1. Using a link:

Use the **<a>** tag with an href value of the ID of the content to collapse.

On the collapsible content's container, add the **.collapse** class, and be sure to give it an ID.

Example:

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

    <p><a data-toggle="collapse" href="#footwear" aria-expanded="false" aria-controls="footwear">Footwear</a> </p>

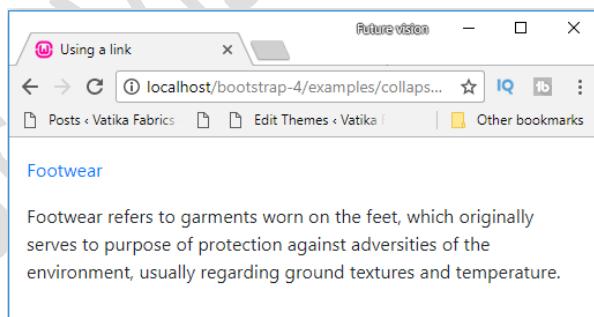
    <div class="collapse" id="footwear">

        <p>Footwear refers to garments worn on the feet, which originally serves to purpose of protection against adversities of the environment, usually regarding ground textures and temperature.</p>

    </div>

</div>
```

Output:



2. Using a button:

Use the **data-target** attribute with a value of the ID of the content to collapse.

On the collapsible content's container, add the **.collapse** class, and be sure to give it an ID.

Example:

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

    <button class="btn btn-info" type="button" data-toggle="collapse" data-
target="#footwear" aria-expanded="false" aria-controls="footwear"> Footwear
    </button>

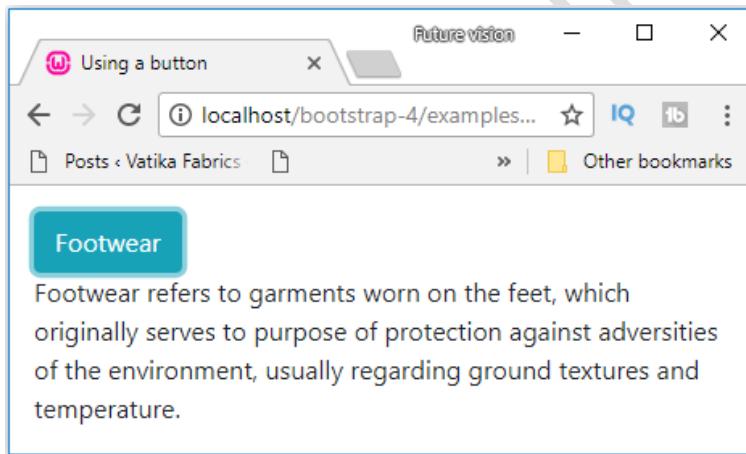
    <div class="collapse" id="footwear">

        <p>Footwear refers to garments worn on the feet, which originally serves to purpose of protection against adversities of the environment, usually regarding ground textures and temperature.</p>

    </div>

</div>
```

Output:



3. Accordion:

- Collapsible content is often used to make an "accordion". Accordions are popular for content such as FAQs, product overviews, etc
- You can use Bootstrap cards to style the accordion like below. Add the **.show** class to the collapsible content to have it expanded when the page loads. Also use **aria-expanded="true"** in this case.

Example:

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

    <div id="faq" role="tablist" aria-multiselectable="true">

        <div class="card">

            <div class="card-header" role="tab" id="questionOne">

                <h5 class="card-title"><a data-toggle="collapse" data-parent="#faq" href="#answerOne" aria-expanded="true" aria-controls="answerOne"> What if my boots are too big for my feet? </a></h5>

            </div>

            <div id="answerOne" class="collapse show" role="tabcard" aria-labelledby="questionOne">

                <div class="card-body"> Stuff your boots with newspaper or tissue. </div>

            </div>

        </div>

        <div class="card">

            <div class="card-header" role="tab" id="questionTwo">

                <h5 class="card-title"><a class="collapsed" data-toggle="collapse" data-parent="#faq" href="#answerTwo" aria-expanded="false" aria-controls="answerTwo"> Can I wear my boots inside? </a></h5>

            </div>

            <div id="answerTwo" class="collapse" role="tabcard" aria-labelledby="questionTwo">

                <div class="card-body"> No. Your mama should've told you about this. </div>

            </div>

        </div>

    </div>

</div>
```

```

        </div>

        </div>

        <div class="card">

            <div class="card-header" role="tab" id="questionThree">

                <h5 class="card-title"><a class="collapsed" data-toggle="collapse" data-parent="#faq" href="#answerThree" aria-expanded="true" aria-controls="answerThree"> What if my boots get slippery when wet? </a> </h5>

            </div>

            <div id="answerThree" class="collapse in" role="tabcard" aria-labelledby="questionThree">

                <div class="card-body"> Keep your boots dry. </div>

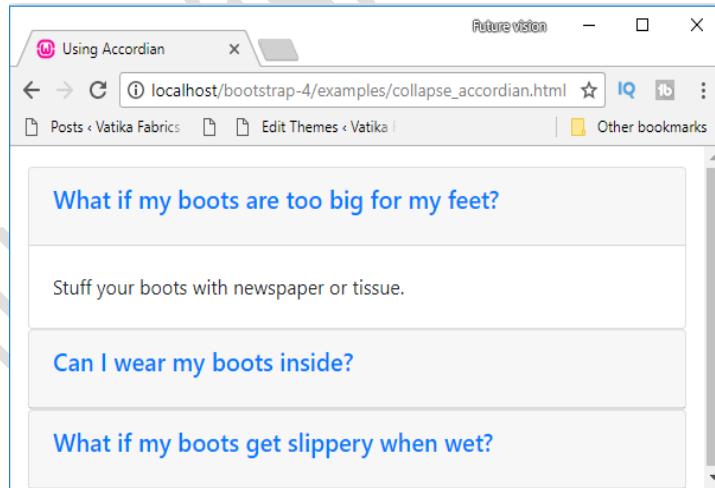
            </div>

        </div>

    </div>

```

Output:



CHAPTER 15: NAVS

Base Nav:

- Bootstrap 5 provides a base style that you can use with nav items. You can either add further Bootstrap styles to this base or customize it according to your own needs.
- The base nav component uses the **.nav** class on the outer navigation element such as a **** or **<nav> element**.

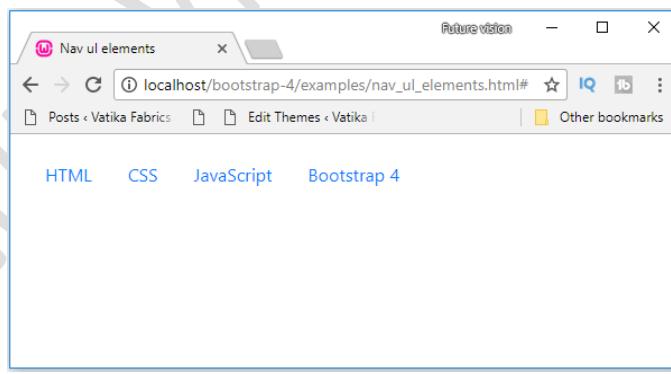
Base Nav on ul elements & <nav> elements

1. Base Nav on ul elements:

Example:

```
<div class="container-fluid">
  <ul class="nav">
    <li class="nav-item"><a class="nav-link active" href="#">HTML</a></li>
    <li class="nav-item"><a class="nav-link" href="#">CSS</a></li>
    <li class="nav-item"><a class="nav-link" href="#">JavaScript</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Bootstrap 5</a></li>
  </ul>
</div>
```

Output:



2. Base Nav on <nav> element:

Example:

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

```

<nav class="nav">

    <a class="nav-link active" href="#">HTML</a>

    <a class="nav-link" href="#">CSS</a>

    <a class="nav-link" href="#">JavaScript</a>

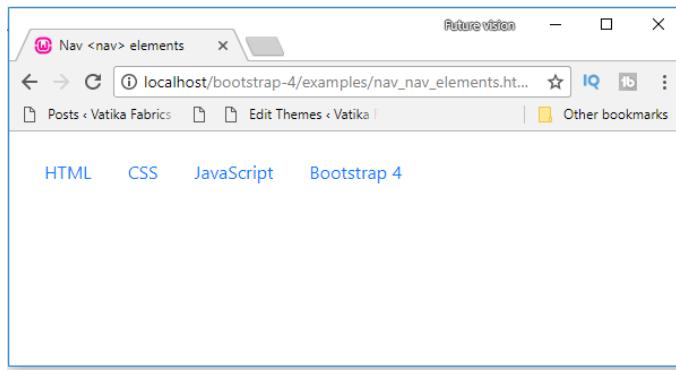
    <a class="nav-link" href="#">Bootstrap 5</a>

</nav>

</div>

```

Output:



Note:

- Bootstrap 3 doesn't use the .nav-item or .nav-link classes.
- Bootstrap 5 requires **.nav-item** to be applied to any elements.
- Bootstrap 5 requires **.nav-link** to be applied to the <a> element.
- Bootstrap 3 requires any **.active** class to be applied to the element.
- Bootstrap 5 requires any **.active** class to be applied to the <a> element.

3. Vertical Navs:

Add the **.flex-column** utility class to the **.nav** element to stack the nav items vertically.

Example:

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

```

<nav class="nav flex-column">

    <a class="nav-link active" href="#">HTML</a>

    <a class="nav-link" href="#">CSS</a>

    <a class="nav-link" href="#">JavaScript</a>

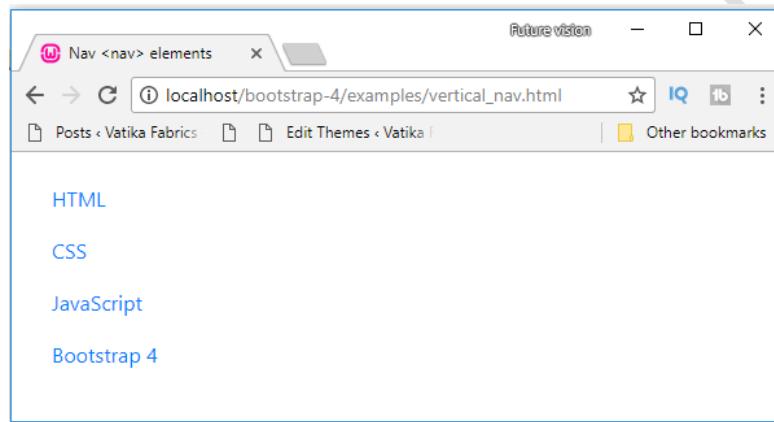
    <a class="nav-link" href="#">Bootstrap 5</a>

</nav>

</div>

```

Output:



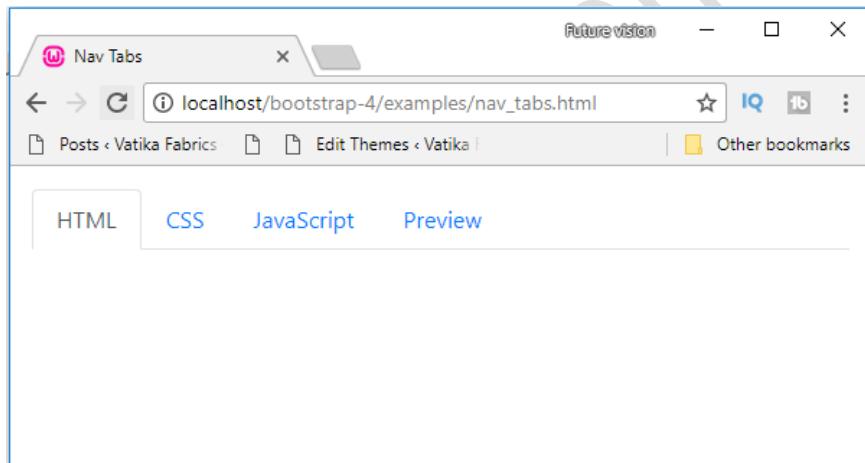
3. Tabs

- Add **class="nav nav-tabs"** to the **** element containing the list of navigation items.
- Also use **class="active"** to make a tab the selected tab.

Example:

```
<div class="container-fluid">  
    <ul class="nav nav-tabs">  
        <li class="nav-item"><a class="nav-link active" href="#">HTML</a></li>  
        <li class="nav-item"><a class="nav-link" href="#">CSS</a></li>  
        <li class="nav-item"><a class="nav-link" href="#">JavaScript</a></li>  
        <li class="nav-item"><a class="nav-link" href="#">Preview</a></li>  
    </ul>  
</div>
```

Output:



5. Pills:

Change .nav-tabs to .nav-pills to display pills instead.

Example:

```
<div class="container-fluid">  
    <ul class="nav nav-pills">
```

```

<li class="nav-item"><a class="nav-link active" href="#">HTML</a> </li>

<li class="nav-item"><a class="nav-link" href="#">CSS</a> </li>

<li class="nav-item"><a class="nav-link" href="#">JavaScript</a> </li>

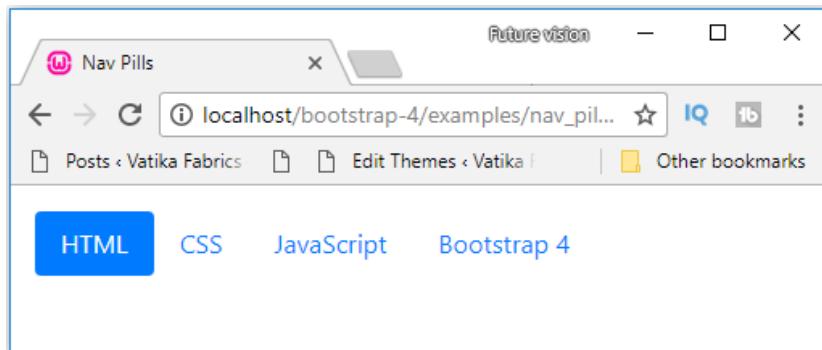
<li class="nav-item"><a class="nav-link" href="#">Bootstrap 5</a> </li>

</ul>

</div>

```

Output:



5. Vertical Pills:

As with any nav, you can stack pills vertically by adding the **.flex-column** utility class to the class list.

Example:

```

<div class="container-fluid">

<ul class="nav nav-pills flex-column">

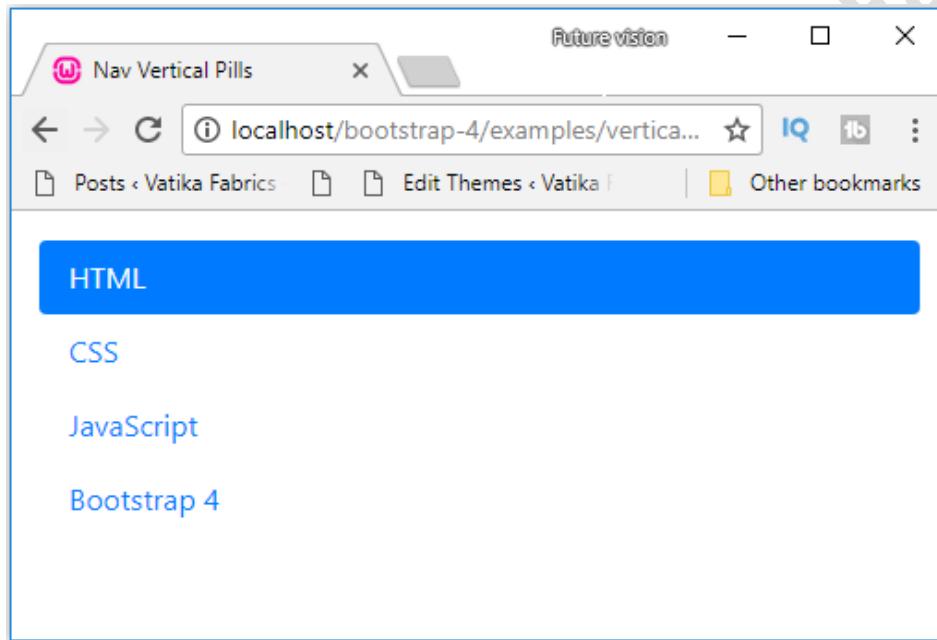
```

```

<li class="nav-item"><a class="nav-link active" href="#">HTML</a> </li>
<li class="nav-item"><a class="nav-link" href="#">CSS</a> </li>
<li class="nav-item"><a class="nav-link" href="#">JavaScript</a> </li>
<li class="nav-item"><a class="nav-link" href="#">Preview</a> </li>
</ul>
</div>

```

Output:



6. Tabs with Dropdown Menus:

You can add a dropdown menu to a tab . In this case, the **.dropdown** class is applied to a **** element (not a **<div>** element).

Example:

```

<div class="container-fluid">
<ul class="nav nav-tabs">
<li class="nav-item"><a class="nav-link active" href="#">HTML</a> </li>
<li class="nav-item"><a class="nav-link" href="#">CSS</a> </li>

```

```

<li class="nav-item"> <a class="nav-link" href="#">JavaScript</a> </li>

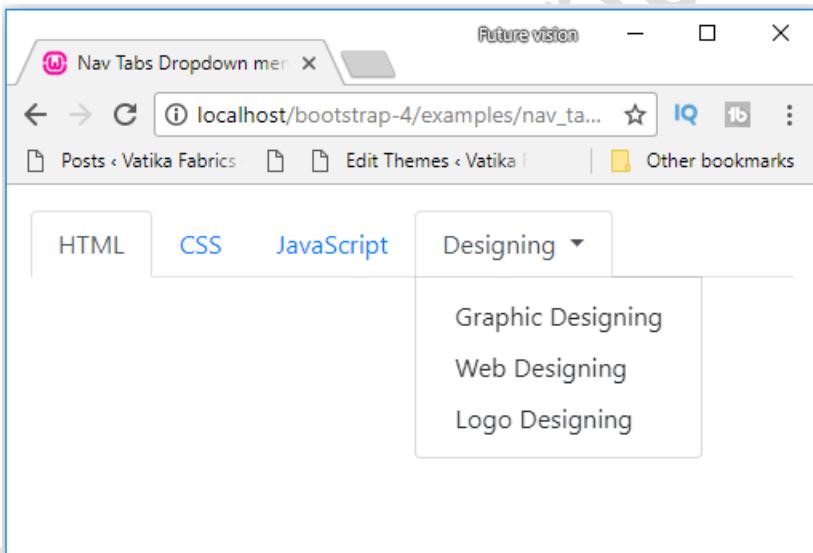
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false"> Designing </a>
    <div class="dropdown-menu" aria-labelledby="Designing">
        <a class="dropdown-item" href="#">Graphic Designing</a>
        <a class="dropdown-item" href="#">Web Designing</a>
        <a class="dropdown-item" href="#">Logo Designing</a>
    </div>
</li>

```

```

</ul>
</div>

```



Output:

7. Pills with Dropdown Menus:

You can add a dropdown menu to a pill . In this case, the **.dropdown** class is applied to a **** element (not a **<div>** element).

Example:

```

<div class="container-fluid">
    <ul class="nav nav-pills">
        <li class="nav-item"> <a class="nav-link active" href="#">HTML</a> </li>
        <li class="nav-item"> <a class="nav-link" href="#">CSS</a> </li>
        <li class="nav-item"> <a class="nav-link" href="#">JavaScript</a> </li>
    </ul>

```

```

<li class="nav-item dropdown">

    <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false"> Designing </a>

        <div class="dropdown-menu" aria-labelledby="Designing">

            <a class="dropdown-item" href="#">Graphic Designing</a>

            <a class="dropdown-item" href="#">Web Designing</a>

            <a class="dropdown-item" href="#">Logo Designing</a>

        </div>

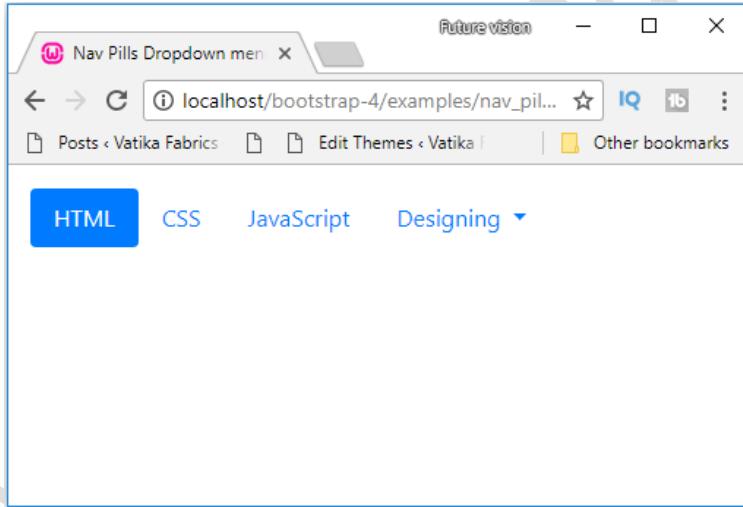
    </li>

</ul>

```

</div>

Output:



8. Tabbable Panes:

You can even use Bootstrap to create tabbable panels of content.

To do this, ensure each `<a>` in the navigation has `data-toggle="tab"` or `data-toggle="pill"` (depending on whether you're using tabs or pills). Then, beneath the tab list, add a `<div>` with the `.tab-content` class. This will contain the content. Then, for each piece of content, nest another `<div>` with the `.tab-pane` class.

Also, you can make tabs fade in by adding the `.fade` class to each `.tab-pane`. The first tab pane (or the active tab pane) must also have the `.show` class to properly fade in initial content.

Example:

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

```

<ul id="clothing-nav" class="nav nav-tabs" role="tablist">

    <li class="nav-item"><a class="nav-link active" href="#home" id="home-tab" role="tab" data-toggle="tab" aria-controls="home" aria-expanded="true">Home</a> </li>

    <li class="nav-item"><a class="nav-link" href="#hats" role="tab" id="hats-tab" data-toggle="tab" aria-controls="hats">Hats</a> </li>

    <!-- Dropdown -->

    <li class="nav-item dropdown"><a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">Footwear </a>

        <div class="dropdown-menu"><a class="dropdown-item" href="#dropdown-shoes" role="tab" id="dropdown-shoes-tab" data-toggle="tab" aria-controls="dropdownShoes">Shoes</a> <a class="dropdown-item" href="#dropdown-boots" role="tab" id="dropdown-boots-tab" data-toggle="tab" aria-controls="dropdownBoots">Boots</a> </div>

    </li>
</ul>

<!-- Content Panel -->

<div id="clothing-nav-content" class="tab-content">

    <div role="tabpanel" class="tab-pane fade show active" id="home" aria-labelledby="home-tab">

        <p>Welcome home! Click on the tabs to see the content change.</p>

    </div>

    <div role="tabpanel" class="tab-pane fade" id="hats" aria-labelledby="hats-tab">

        <p>A hat is a head covering. It can be worn for protection against the elements, ceremonial reasons, religious reasons, safety, or as a fashion accessory.</p>

    </div>

```

```

<div role="tabpanel" class="tab-pane fade" id="dropdown-shoes" aria-
labelledby="dropdown-shoes-tab">

    <p>A shoe is an item of footwear intended to protect and comfort the
    human foot while doing various activities. Shoes are also used as an item of
    decoration.</p>

</div>

<div role="tabpanel" class="tab-pane fade" id="dropdown-boots" aria-
labelledby="dropdown-boots-tab">

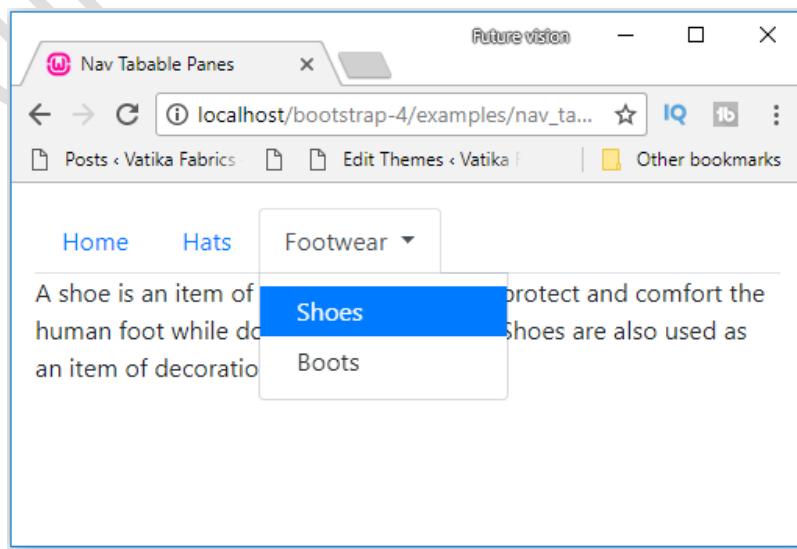
    <p>A boot is a type of footwear and a specific type of shoe. Most boots
    mainly cover the foot and the ankle, while some also cover some part of the
    lower calf. Some boots extend up the leg, sometimes as far as the knee or even
    the hip.</p>

</div>

</div>

```

Output:



FutureVision Computers



CHAPTER 15: NAVBAR

Add a fully-functional, responsive navbar to your website with minimal code.

The Bootstrap 5 navbar is a simple wrapper for positioning branding, navigation, and other elements into a navigation header.

1. Basic Navbar:

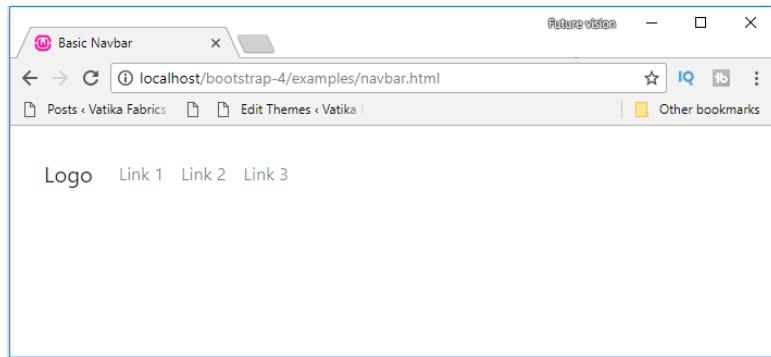
Example:

```
<div class="container-fluid">
  <nav class="navbar navbar-expand-sm navbar-light bg-faded">
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>

    <!-- Brand -->
    <a class="navbar-brand" href="#">Logo</a>

    <!-- Links -->
    <div class="collapse navbar-collapse" id="nav-content">
      <ul class="navbar-nav">
        <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Link 3</a></li>
      </ul>
    </div>
  </nav>
</div>
```

Output:



Here's an explanation:

- Wrap everything inside a nav element with the .navbar class and a **.navbar-expand{-sm|-md|-lg|-xl}** class, along with a color scheme.
- For the toggable menu on smaller devices, use a button element with **.navbar-toggler**. To display the "hamburger", use the **.navbar-toggler-icon** class on a span element.
- For a list of links, use a **ul** with the **.navbar-nav** class.
- For each individual list item, use **li** with **.nav-item**.
- For the individual links use a with **.nav-link**.

2. Navbars with Dropdowns:

You can add dropdown menus to your navbar. Just add the dropdown code to a **li** element with the **.dropdown** class applied.

Example:

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

  <nav class="navbar navbar-expand-sm navbar-light bg-faded">

    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>

    <!-- Brand -->

    <a class="navbar-brand" href="#">Logo</a>

    <!-- Links -->

    <div class="collapse navbar-collapse" id="nav-content">

      <ul class="navbar-nav">

        <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>

        <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>

        <li class="nav-item dropdown">

          <a class="nav-link dropdown-toggle" data-toggle="dropdown" id="Preview" href="#" role="button" aria-haspopup="true" aria-expanded="false">
Dropdown
          </a>
          <div class="dropdown-menu" aria-labelledby="Preview">

            <a class="dropdown-item" href="#">Dropdown Link 1</a>

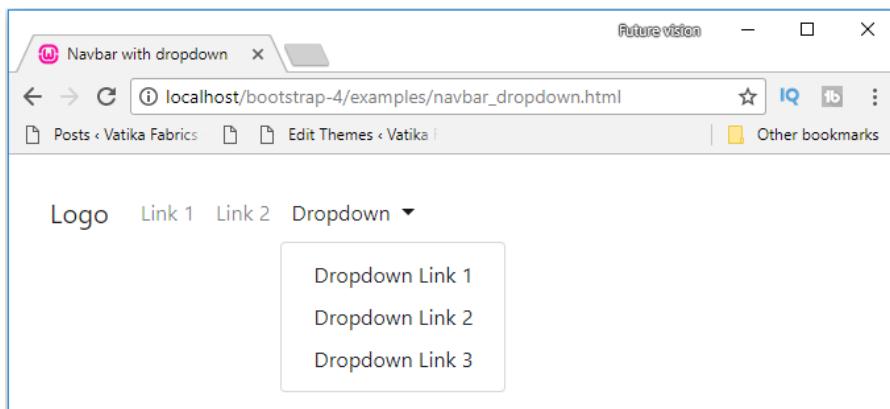
            <a class="dropdown-item" href="#">Dropdown Link 2</a>

            <a class="dropdown-item" href="#">Dropdown Link 3</a>

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

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

Output:



3. Navbar with Form Control:

You can include form controls within a navbar by adding the **.form-inline** to the form element. Here we also use **.mr-auto** on the previous element so that the form is aligned to the right.

Example:

```
<div class="container-fluid">  
  
<nav class="navbar navbar-expand-sm navbar-light bg-faded">  
  
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span> </button>  
  
    <!-- Brand -->  
  
    <a class="navbar-brand" href="#">Logo</a>  
  
    <!-- Links -->  
  
    <div class="collapse navbar-collapse" id="nav-content">  
  
        <ul class="navbar-nav mr-auto">  
  
            <li class="nav-item"><a class="nav-link" href="#">Link 1</a> </li>  
  
            <li class="nav-item"><a class="nav-link" href="#">Link 2</a> </li>
```

```
</ul>

<!-- Search -->

<form class="form-inline" role="search">

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

    <button type="submit" class="btn btn-secondary">Search</button>

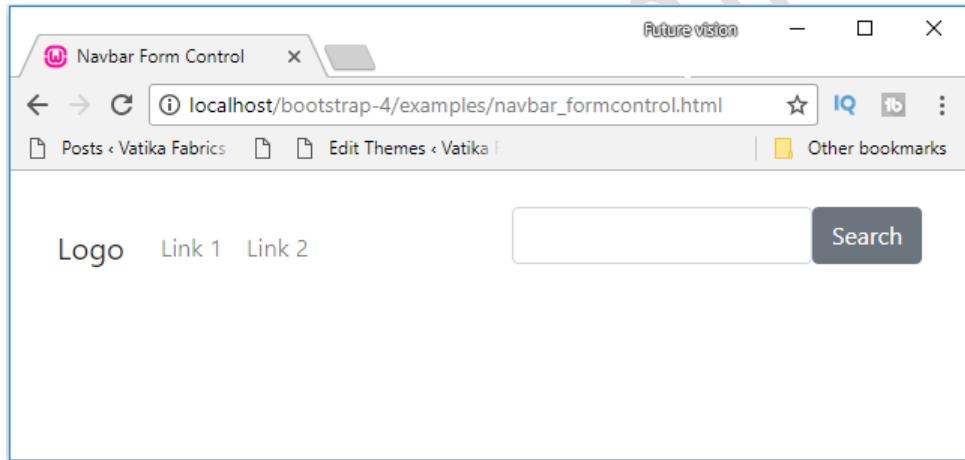
</form>

</div>

</nav>

</div>
```

Output:



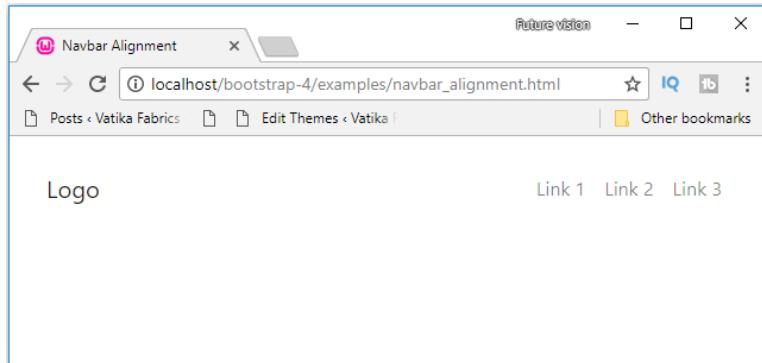
5. Alignment:

You can use any of the spacing utilities (such as mr-auto) or any of the flexbox alignment utilities (such as **.justify-content-end**) to align elements within the navbar.

Example:

```
<div class="container-fluid">  
  <nav class="navbar navbar-expand-sm navbar-light bg-faded">  
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>  
  
    <!-- Brand -->  
  
    <a class="navbar-brand" href="#">Logo</a>  
  
    <!-- Links -->  
  
    <div class="collapse navbar-collapse justify-content-end" id="nav-content">  
      <ul class="navbar-nav">  
        <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>  
        <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>  
        <li class="nav-item"><a class="nav-link" href="#">Link 3</a></li>  
      </ul>  
    </div> </nav>  
  </div>
```

Output:



Navbar Colors:

Bootstrap provides various options for adding color to navbars. The way it works is, you specify light or dark (via **.navbar-light** or **.navbar-dark**), then specify a color.

Color can be specified either via one of Bootstrap's color classes (eg, **.bg-primary**, **.bg-dark**), or by specifying your own color with CSS (eg, using background-color).

5. Primary Color, Light:

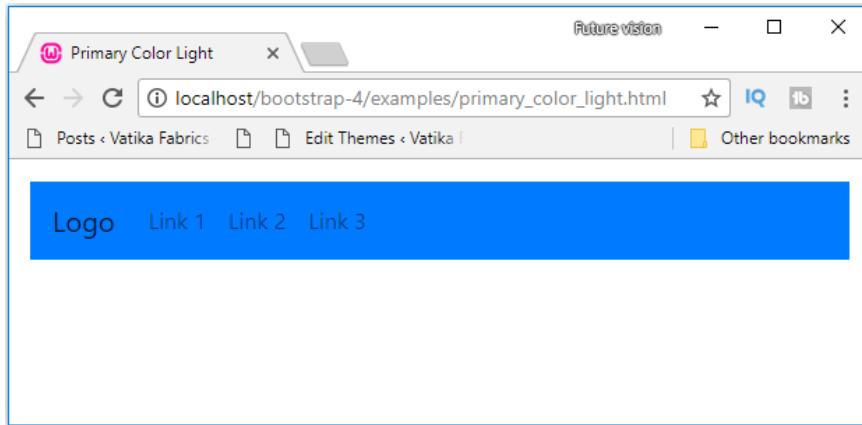
use **.bg-primary** along with **.navbar-light**:

Example:

```
<div class="container-fluid">
  <nav class="navbar navbar-expand-sm navbar-light bg-primary">
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"> <span class="navbar-toggler-icon"></span> </button>
    <!-- Brand -->
    <a class="navbar-brand" href="#">Logo</a>
    <!-- Links -->
    <div class="collapse navbar-collapse" id="nav-content">
      <ul class="navbar-nav">
        <li class="nav-item"> <a class="nav-link" href="#">Link 1</a> </li>
        <li class="nav-item"> <a class="nav-link" href="#">Link 2</a> </li>
        <li class="nav-item"> <a class="nav-link" href="#">Link 3</a> </li>
```

```
</ul>  
</nav>  
</div>
```

Output:



6. Primary Color, Dark:

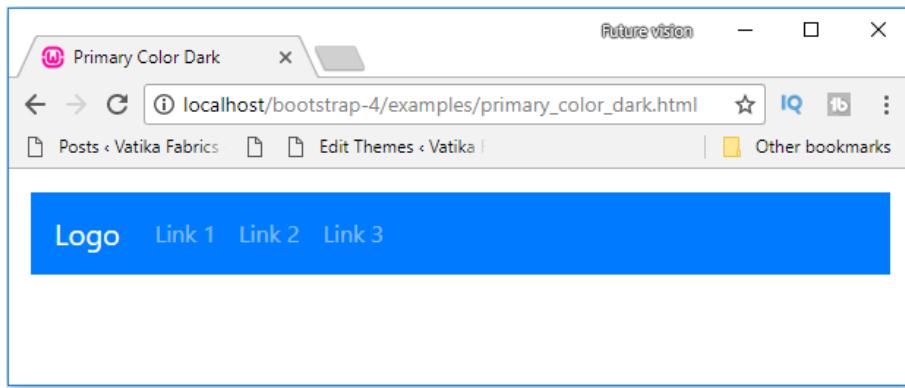
use **.bg-primary** along with **.navbar-dark**:

Example:

```
<div class="container-fluid">  
<nav class="navbar navbar-expand-sm navbar-dark bg-primary">  
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>  
  <!-- Brand -->  
  <a class="navbar-brand" href="#">Logo</a>  
  <!-- Links -->  
  <div class="collapse navbar-collapse" id="nav-content">  
    <ul class="navbar-nav">  
      <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>  
      <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>  
      <li class="nav-item"><a class="nav-link" href="#">Link 3</a></li>
```

```
</ul>  
</nav></div>
```

Output:



7. Custom Color:

To specify your own custom color, omit any .bg-* class (eg, remove .bg-primary) and add your own color using CSS. Bootstrap 5 introduced the .navbar-light class, and allows the .bg-* classes to be used on navbars.

Example:

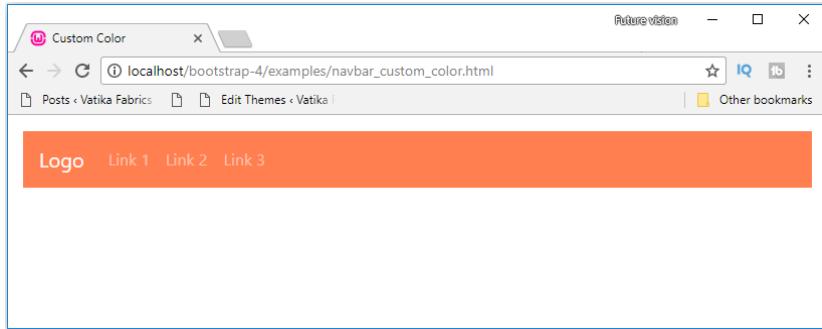
```
<div class="container-fluid">  
  
    <nav class="navbar navbar-expand-sm navbar-dark" style="background:  
coral;">  
  
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-  
target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-  
label="Toggle navigation"><span class="navbar-toggler-icon"></span>  
        </button>  
  
        <!-- Brand -->  
  
        <a class="navbar-brand" href="#">Logo</a>  
  
        <!-- Links -->  
  
        <div class="collapse navbar-collapse" id="nav-content">  
            <ul class="navbar-nav">  
  
                <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>  
  
                <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>
```

```

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

```

Output:



8. Fixed Top Navbar:

You can fix a navbar to the top of the viewport by using either **.fixed-top**.

Example:

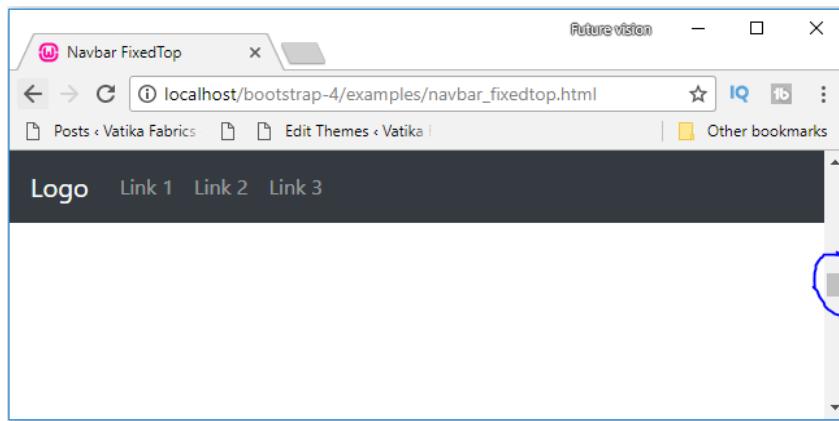
```

<div class="container-fluid">
    <nav class="navbar fixed-top navbar-expand-sm navbar-dark bg-dark">
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span>
        </button>
        <!-- Brand -->
        <a class="navbar-brand" href="#">Logo</a>
        <!-- Links -->
        <div class="collapse navbar-collapse" id="nav-content">
            <ul class="navbar-nav">
                <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>
                <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>
                <li class="nav-item"><a class="nav-link" href="#">Link 3</a></li>
            </ul>
        </div>
    </nav>
</div>

```

```
</ul>  
</nav>  
</div>
```

Output:



9. Fixed Bottom Navbar:

You can fix a navbar to the bottom of the viewport by using either **.fixed-bottom**.

Example:

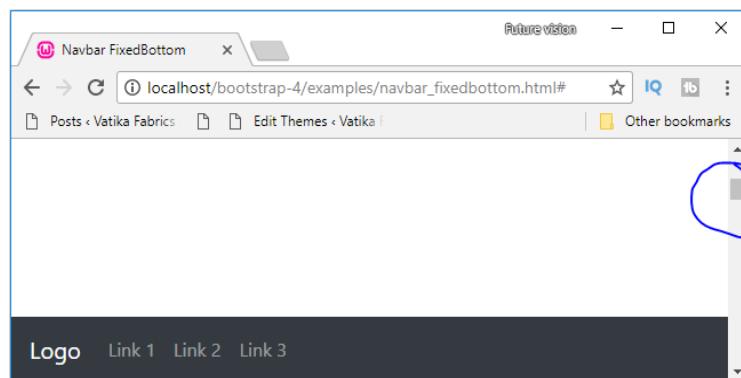
```
<div class="container-fluid">  
  
<nav class="navbar fixed-bottom navbar-expand-sm navbar-dark bg-dark">  
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-content" aria-controls="nav-content" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>  
  <!-- Brand -->  
  <a class="navbar-brand" href="#">Logo</a>  
  <!-- Links -->  
  <div class="collapse navbar-collapse" id="nav-content">  
    <ul class="navbar-nav">  
      <li class="nav-item"><a class="nav-link" href="#">Link 1</a></li>  
      <li class="nav-item"><a class="nav-link" href="#">Link 2</a></li>
```

```

<li class="nav-item"> <a class="nav-link" href="#">Link 3</a> </li>
</ul>
</nav>

```

Output:



10. Vertical Navbar:

Remove the `.navbar-expand-*` class to create a navigation bar that will always be vertical:

Example:

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

```

```

<nav class="navbar bg-light">
  <div class="container-fluid">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Digital Marketing</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Computer Hardware</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">UI/UX Development</a>
      </li>
    </ul>
  </div>
</nav>

<div class="container-fluid mt-3">
  <h3>Vertical Navbar Example</h3>
  <p>A navigation bar is a navigation header that is placed at the top of the page.</p>
</div>
</body>
</html>

```

Output:

Digital Marketing
Computer Hardware
UI/UX Development

Vertical Navbar Example

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

11. Brand/Logo in Navbar:

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

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">fv logo here</a>
  </div>
</nav>
```



```
<div class="container-fluid mt-3">  
  <h3>Brand / Logo Here</h3>  
  <p>The .navbar-brand class is used to highlight the brand/logo/project name of your page.</p>  
</div>  
  
</body>  
</html>
```

Output:

fv logo here

Brand / Logo Here

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

CHAPTER 16 – BREADCRUMBS

Breadcrumbs:

- On websites that have a lot of pages, breadcrumb navigation can greatly enhance the way users find their way around
- Bootstrap provides the **.breadcrumb** class which styles a list into a breadcrumb.
- Bootstrap adds separators (by using **:before and content**) to the list as well as **display: inline-block** to the **li** element — resulting in that "breadcrumb" effect.

1. Creating a Breadcrumb

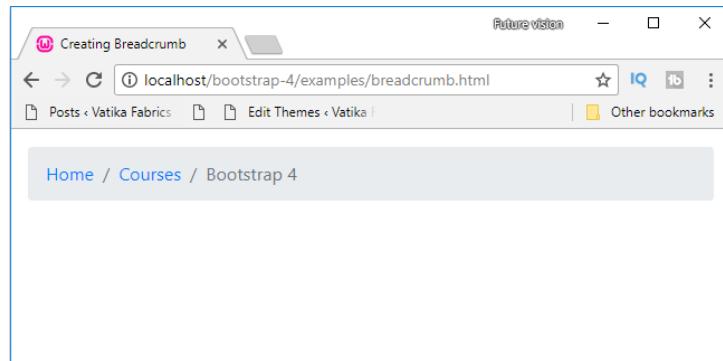
To create a breadcrumb:

- Add **class="breadcrumb"** to the **ul** element that represents the breadcrumb.
- Add the **.breadcrumb-item** class to each **li** element within the breadcrumb.
- Also add **class="active"** to the **li** that represents the current page/section (the page that the user is currently on) — which would almost always be the last crumb in the breadcrumb.

Example:

```
<div class="container-fluid">
    <ul class="breadcrumb">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item"><a href="#">Courses</a></li>
        <li class="breadcrumb-item active">Bootstrap 5</li>
    </ul>
</div>
```

Output:



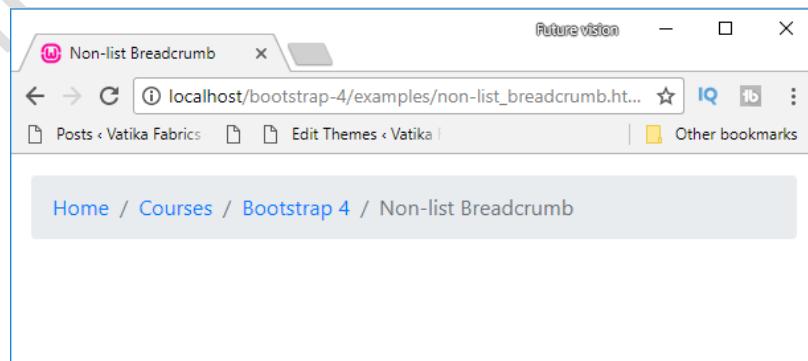
2. Non-List Breadcrumbs:

Breadcrumbs don't necessarily need to be in a list. You can use them like this if required:

Example:

```
<div class="container-fluid">  
    <ul class="breadcrumb">  
        <li class="breadcrumb-item"><a href="#">Home</a></li>  
        <li class="breadcrumb-item"><a href="#">Courses</a></li>  
        <li class="breadcrumb-item active">Bootstrap 5</li>  
    </ul>  
</div>
```

Output:



CHAPTER 17 – PAGINATION

- Create "Previous/Next" or "Next n" controls with Bootstrap's pagination classes.
- Pagination is typically where you have navigation that links to multiple pages within a series. For example, search results that return many pages will have pagination that enables the user to navigate to the next page or even jump forward several pages.

1. Default Pagination & Active Current Page

- To create default pagination, add **class="pagination"** to the **** element that represents the list of pages.
- Also add **.page-item** to each **** element and **.page-link** to each **<a>** element.
- Add the **.active** class to the **** element that represents the current page (the page that the user is currently on).

Example:

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

    <nav>

        <ul class="pagination">

            <li class="page-item"><a href="#" class="page-link" aria-label="Previous">
<span aria-hidden="true">&laquo;</span></a></li>

            <li class="page-item"><a href="#" class="page-link">1</a></li>

            <li class="page-item"><a href="#" class="page-link">2</a></li>

            <li class="page-item active"><a href="#" class="page-link">3</a></li>

            <li class="page-item "><a href="#" class="page-link">5</a></li>

            <li class="page-item"><a href="#" class="page-link">5</a></li>

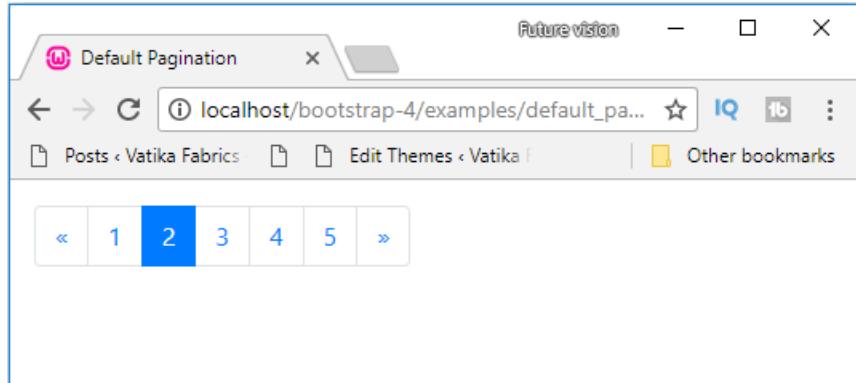
            <li class="page-item"><a href="#" class="page-link" aria-label="Next">
<span aria-hidden="true">&raquo;</span></a></li>

        </ul>

    </nav>

</div>
```

Output:



Note:

Bootstrap 3 only requires the **.pagination** class.

Bootstrap 5, in addition to the **.pagination** class, also requires the **.page-item** class to be added to each **** element and **.page-link** to each **<a>** element.

2. Disabling an Option

⋮ Add the **.disabled** class to the **** element to disable that option.

Example

```
<div class="container-fluid">
  <nav>
    <ul class="pagination">
      <li class="page-item"><a href="#" class="page-link" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
      </a></li>

      <li class="page-item"><a href="#" class="page-link">1</a></li>

      <li class="page-item disabled"><a href="#" class="page-link">2</a></li>

      <li class="page-item"><a href="#" class="page-link">3</a></li>

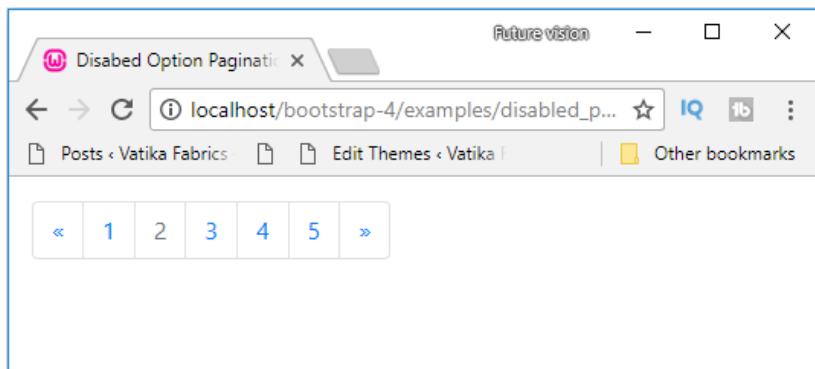
      <li class="page-item"><a href="#" class="page-link">5</a></li>

      <li class="page-item"><a href="#" class="page-link">5</a></li>

      <li class="page-item"><a href="#" class="page-link" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a></li>
    </ul>
  </nav>
</div>
```

```
</ul>  
</nav>  
</div>
```

Output:



FutureVisionComputers

3. Pagination Size

Add either the **.pagination-lg** or **.pagination-sm** to the **** element to increase or decrease the size of the pagination controls.

Example:

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

    <nav>

        <ul class="pagination pagination-lg">

            <li class="page-item"><a href="#" class="page-link" aria-label="Previous"><span aria-hidden="true">&laquo;</span></a></li>

            <li class="page-item"><a href="#" class="page-link">1</a></li>

            <li class="page-item"><a href="#" class="page-link">2</a></li>

            <li class="page-item"><a href="#" class="page-link">3</a></li>

            <li class="page-item"><a href="#" class="page-link">5</a></li>

            <li class="page-item"><a href="#" class="page-link">5</a></li>

            <li class="page-item"><a href="#" class="page-link">6</a></li>

            <li class="page-item"><a href="#" class="page-link">7</a></li>

            <li class="page-item"><a href="#" class="page-link" aria-label="Next"><span aria-hidden="true">&raquo;</span></a></li>

        </ul>

    </nav>

    <nav>

        <ul class="pagination">

            <li class="page-item"><a href="#" class="page-link" aria-label="Previous"><span aria-hidden="true">&laquo;</span></a></li>

            <li class="page-item"><a href="#" class="page-link">1</a></li>

            <li class="page-item"><a href="#" class="page-link">2</a></li>

            <li class="page-item"><a href="#" class="page-link">3</a></li>

        </ul>

    </nav>

</div>
```

```
<li class="page-item"><a href="#" class="page-link">5</a></li>

<li class="page-item"><a href="#" class="page-link">5</a></li>

<li class="page-item"><a href="#" class="page-link">6</a></li>

<li class="page-item"><a href="#" class="page-link">7</a></li>

<li class="page-item"><a href="#" class="page-link" aria-label="Next"><span aria-hidden="true">&raquo;</span></a></li>

</ul>

</nav>

<nav>

<ul class="pagination pagination-sm">

<li class="page-item"><a href="#" class="page-link" aria-label="Previous"><span aria-hidden="true">&laquo;</span></a></li>

<li class="page-item"><a href="#" class="page-link">1</a></li>

<li class="page-item"><a href="#" class="page-link">2</a></li>

<li class="page-item"><a href="#" class="page-link">3</a></li>

<li class="page-item"><a href="#" class="page-link">5</a></li>

<li class="page-item"><a href="#" class="page-link">5</a></li>

<li class="page-item"><a href="#" class="page-link">6</a></li>

<li class="page-item"><a href="#" class="page-link">7</a></li>

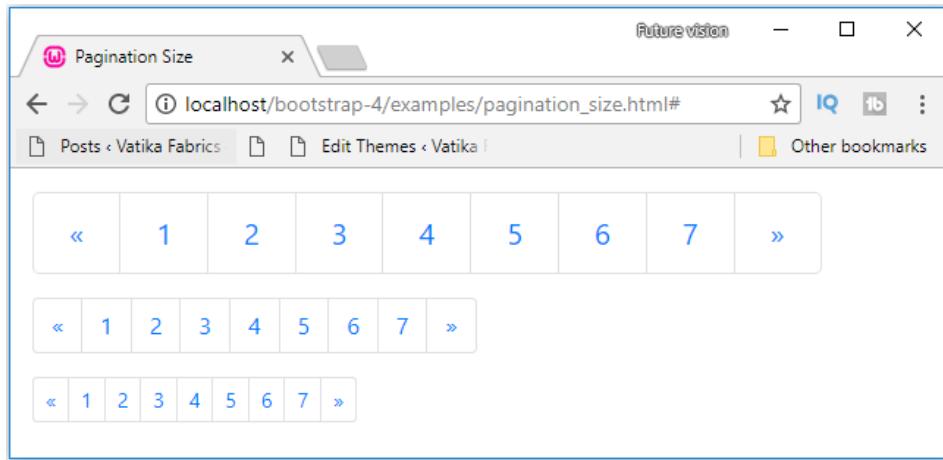
<li class="page-item"><a href="#" class="page-link" aria-label="Next"><span aria-hidden="true">&raquo;</span></a></li>

</ul>

</nav>

</div>
```

Output:



5. Pagination Alignment

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container mt-3">
    <h2>Pagination</h2>
    <p>Use utility classes to change the alignment of the pagination:</p>
```

```
<ul class="pagination">  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Previous</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">1</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">2</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Next</a></li>  
</ul>  
  
<ul class="pagination justify-content-center">  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Previous</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">1</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">2</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Next</a></li>  
</ul>  
  
<ul class="pagination justify-content-end">  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Previous</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">1</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">2</a></li>  
    <li class="page-item"><a class="page-link" href="javascript:void(0);">Next</a></li>  
</ul>  
</div>  
  
</body>  
</html>
```

Output:

Pagination

Use utility classes to change the alignment of the pagination:

Previous 1 2 Next

Previous 1 2 Next

Previous 1 2 Next

FutureVisionComputers

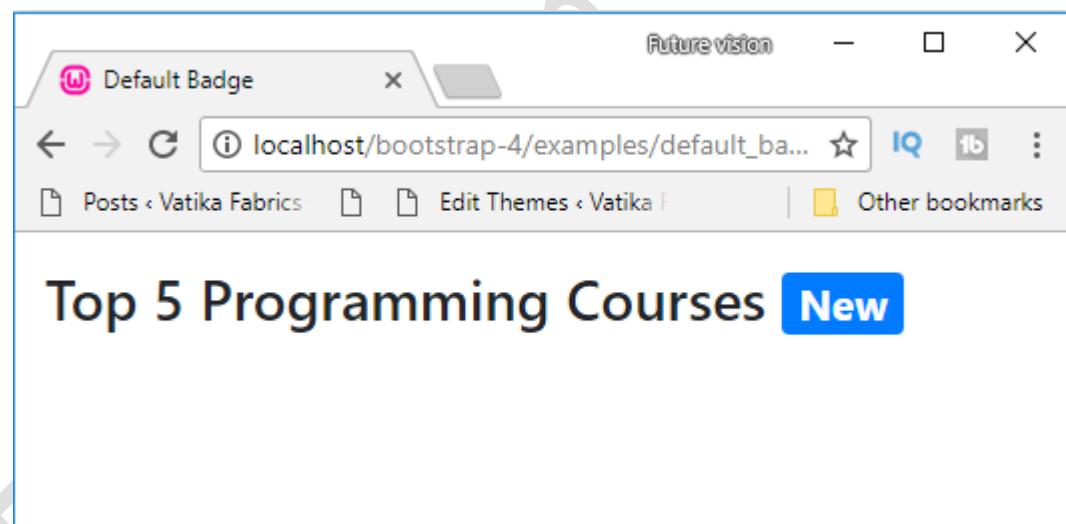
CHAPTER 18 – BADGES

- Bootstrap's badge classes can be used to highlight additional information that's appended to a string of text.
- To create a badge, apply the `.badge` class, as well as one of the `.badge-*` classes to the `` element that represents the badge.
- Here we create a default badge by using `class="badge badge-primary"`.

Example:

```
<div class="container-fluid">  
    <h3>Top 5 Programming Courses <span class="badge badge-primary">New</span></h3>  
</div>
```

Output:



Contextual Classes:

The following contextual classes are available for badges.

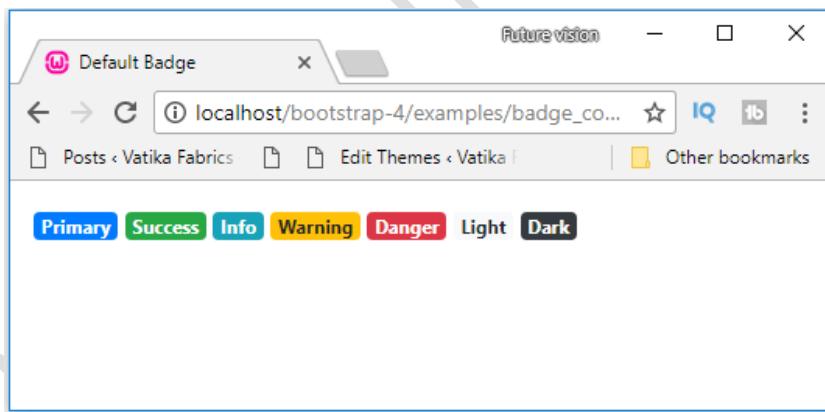
Example:

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

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

</div>
```

Output:



2. Pill Badges:

Make the corners more rounded with Bootstrap's **.badge-pill** class.

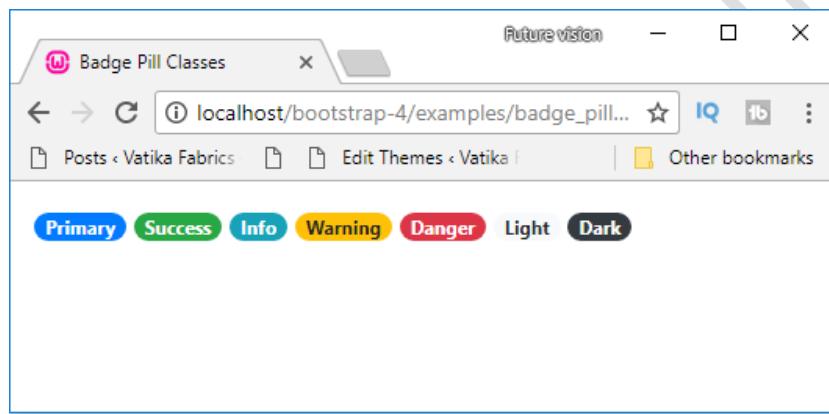
Example:

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

    <span class="badge badge-pill badge-primary">Primary</span>
    <span class="badge badge-pill badge-success">Success</span>
```

```
<span class="badge badge-pill badge-info">Info</span>  
<span class="badge badge-pill badge-warning">Warning</span>  
<span class="badge badge-pill badge-danger">Danger</span>  
<span class="badge badge-pill badge-light">Light</span>  
<span class="badge badge-pill badge-dark">Dark</span>  
</div>
```

Output:



Note:

Bootstrap 3 uses labels and badges, each for a slightly different purpose.

In Bootstrap 5, badges do what labels used to do in Bootstrap 3. Bootstrap 5 also introduced the **.badge-pill** class to provide rounded corners.

3. Badge inside an element:

An example of using a badge inside a button:

Example:

```
<!DOCTYPE html>

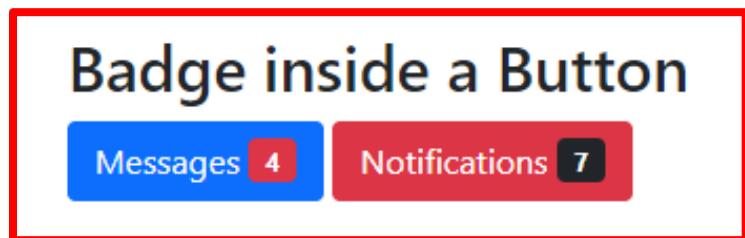
<html lang="en">
  <head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>

    <div class="container mt-3">
      <h2>Badge inside a Button</h2>
      <button type="button" class="btn btn-primary">
        Messages <span class="badge bg-danger">5</span>
      </button>
      <button type="button" class="btn btn-danger">
        Notifications <span class="badge bg-dark">7</span>
      </button>
    </div>

  </body>
```

</html>

Output:



FutureVisionComputers

CHAPTER 19 – JUMBOTRON

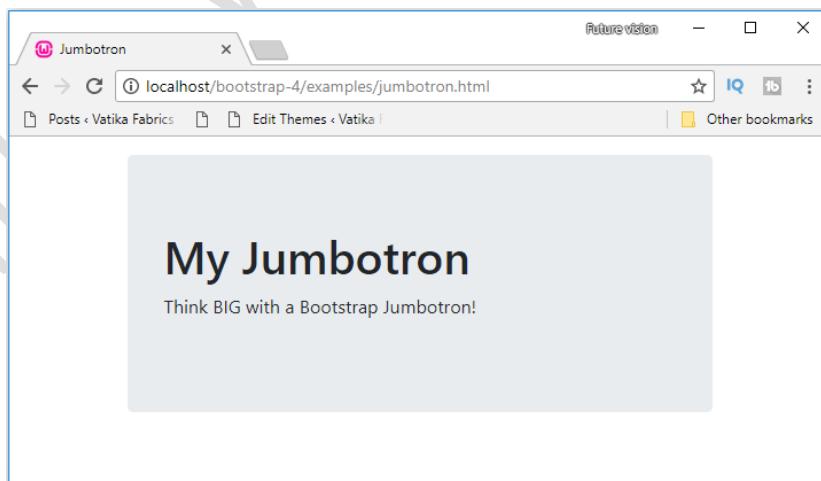
1. Jumbotron:

- Draw major attention with a Bootstrap jumbotron. Ideal for key marketing messages and calls to action.
- In Bootstrap, a jumbotron is a large callout that is styled with large text and heavy padding. By default, Bootstrap 5 jumbotrons are styled in light gray (#eceeef), but you can change this to any color.
- To create a jumbotron, apply the **.jumbotron** class to the **<div>** element that represents the jumbotron.

Example:

```
<div class="container">  
    <div class="jumbotron">  
        <h1>My Jumbotron</h1>  
        <p>Think BIG with a Bootstrap Jumbotron!</p>  
    </div>  
</div>
```

Output:



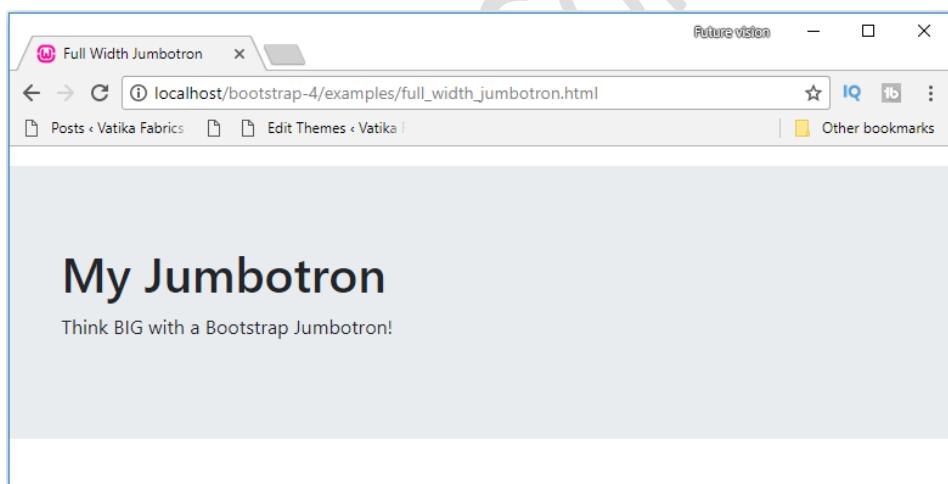
2. Full-Width Jumbotron:

To make the jumbotron full width and without the rounded corners, add the `.jumbotron-fluid` class, and nest a container inside the jumbotron.

Example:

```
<div class="jumbotron jumbotron-fluid">  
  <div class="container">  
    <h1>My Jumbotron</h1>  
    <p>Think BIG with a Bootstrap Full-Width Jumbotron!</p>  
  </div>  
</div>
```

Output:



Note: Bootstrap 5 introduced the `.jumbotron-fluid` class.

CHAPTER 20 – ALERTS

1. Alerts:

- Bootstrap provides an easy way to create a stylized alert message box with its alert component.
- To create an alert box, use the .alert class along with one of the .alert-* classes to specify the kind of alert.

Example:

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

    <div class="alert alert-success" role="alert">Success alert</div>

    <div class="alert alert-info" role="alert">Info alert</div>

    <div class="alert alert-warning" role="alert">Warning alert</div>

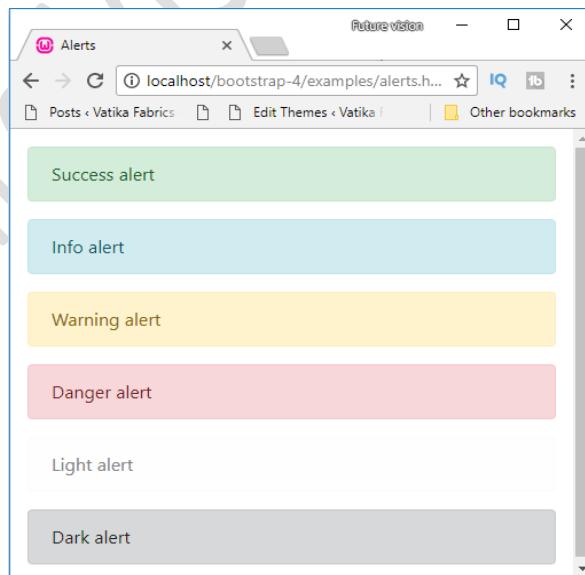
    <div class="alert alert-danger" role="alert">Danger alert</div>

    <div class="alert alert-light" role="alert">Light alert</div>

    <div class="alert alert-dark" role="alert">Dark alert</div>

</div>
```

Output:



2. Dismissible Alerts:

- You can create dismissible alerts by adding Bootstrap's **.alert-dismissible** class, along with a Close button. When the user clicks the Close button, the alert disappears.
- Here we use × to create the Close icon (X).

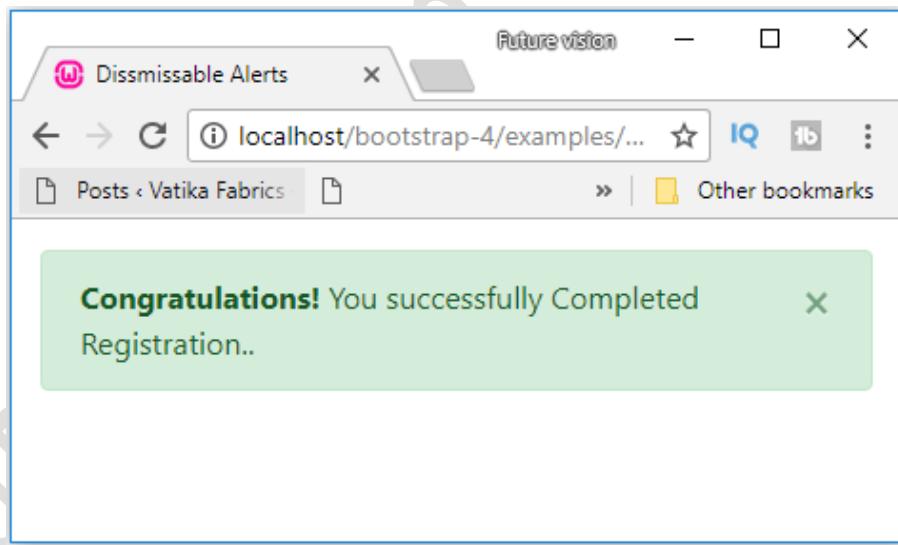
Example:

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

    <div class="alert alert-success alert-dismissible" role="alert">
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
        <strong>Congratulations!</strong> You successfully Completed Registration..
    </div>

</div>
```

Output:



3. Fading Effect and Links:

- You can give the alert a slight fading effect when closing them by adding the **.fade** and **.show** classes.
- Use Bootstrap's **.alert-link** class on the **<a>** tag to match link color with the text.

Example:

```

<div class="container-fluid">

    <div class="alert alert-success alert-dismissible fade show" role="alert">

        <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

            <span aria-hidden="true">&times;</span>

        </button>

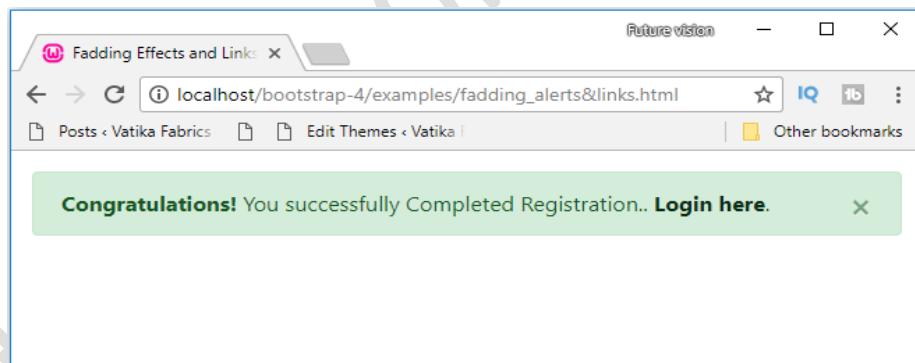
        <strong>Congratulations!</strong> You successfully Completed
Registration.. <a href="#" class="alert-link">Login here</a>.

    </div>

</div>

```

Output:



CHAPTER 21 – MODALS

1. Modal:

- Bootstrap enables you to add a modal dialog box to your site. A modal is a dialog box that takes the focus while the rest of the screen is dimmed or grayed out. This forces the user to take action on the dialog box before continuing.
- To create a modal, use the **.modal** class along with various other **.modal-*** classes to define each section of the modal.

Example:

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

    <button type="button" class="btn btn-primary btn-lg" data-toggle="modal"
    data-target="#flipFlop"> Click Me </button>

    <!-- The modal -->

    <div class="modal fade" id="flipFlop" tabindex="-1" role="dialog" aria-
    labelledby="modalLabel" aria-hidden="true">

        <div class="modal-dialog" role="document">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="modalLabel">Flip flop</h5>

                    <button type="button" class="close" data-dismiss="modal" aria-
                    label="Close"> <span aria-hidden="true">&times;</span> </button>

                </div>

                <div class="modal-body">

                    A type of open-toed sandal.

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-
                    dismiss="modal">Close</button>

                </div>

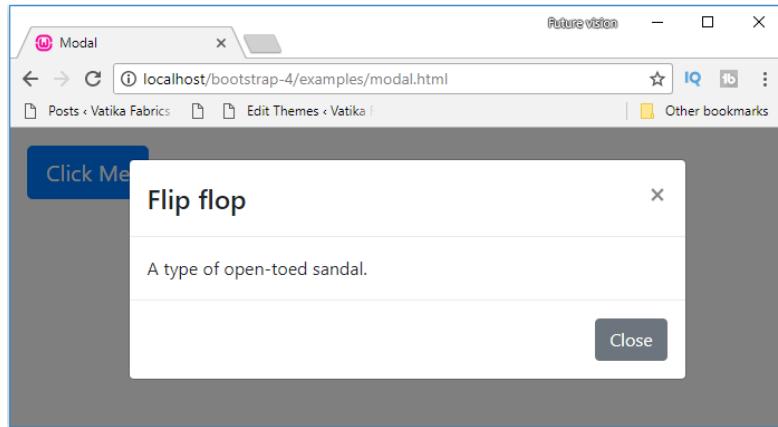
            </div>
        </div>
    </div>
</div>
```

```

        </div>
        </div>
        </div>
        </div>
</div>

```

Output:



2. Small Modal:

- Modals have a default size of medium. You can specify small or large as required.
- For a small modal dialog box, add the **.modal-sm** class to the **.modal-dialog**:

Example:

```

<div class="container-fluid">
    <button type="button" class="btn btn-primary btn-lg" data-toggle="modal"
    data-target="#flipFlop"> Click Me </button>
    <!-- The modal -->

    <div class="modal fade" id="flipFlop" tabindex="-1" role="dialog" aria-
    labelledby="modalLabel" aria-hidden="true">
        <div class="modal-dialog modal-sm" role="document">
            <div class="modal-content">

```

```

<div class="modal-header">

    <h5 class="modal-title" id="modalLabel">Flip flop</h5>

    <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span> </button>

</div>

<div class="modal-body">

    A type of open-toed sandal.

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>

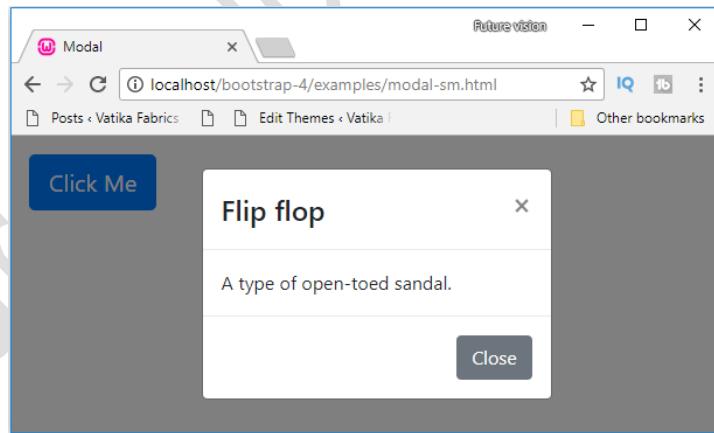
</div>

</div> </div>

</div> </div>

```

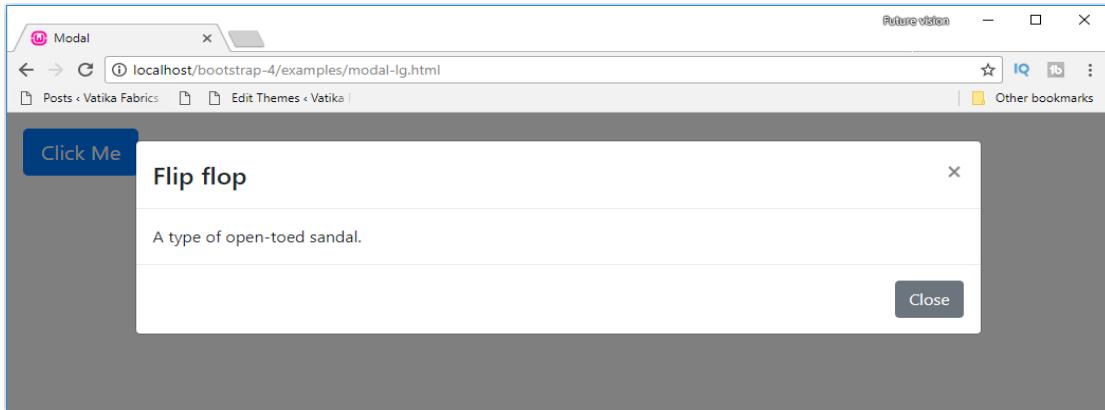
Output:



3. Large Modal:

For a large modal dialog box, add the **.modal-lg** class to the **.modal-dialog**:

Output:



5. Add Animations to Modal:

- Use the **.fade** class to add a fading effect when opening and closing the modal:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">
```

```
<h3>Fading Modal Example</h3>
```

```
<p>Click on the button to open the modal.</p>
```

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">  
    Open modal  
</button>  
</div>
```

```
<!-- The Modal -->
```

```
<div class="modal fade" id="myModal">  
    <div class="modal-dialog">  
        <div class="modal-content">  
            <!-- Modal Header -->  
            <div class="modal-header">  
                <h5 class="modal-title">Modal Heading</h5>  
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>  
            </div>
```

```
<!-- Modal body -->
```

```
<div class="modal-body">  
    Modal body..  
</div>
```

```
<!-- Modal footer -->
```

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

```
</div>

</div>

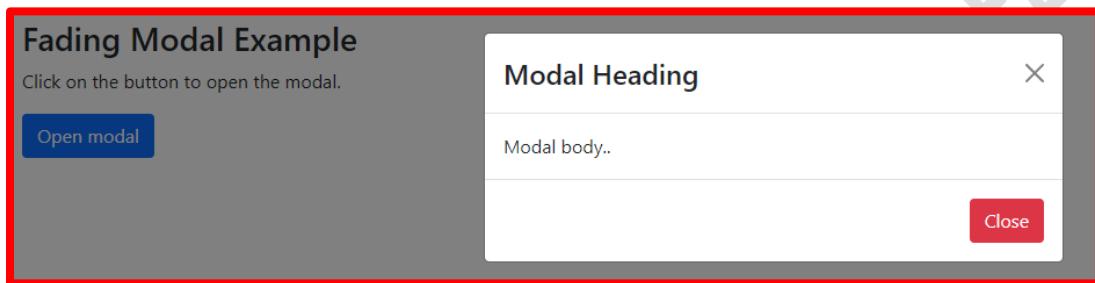
</div>

</div>

</body>

</html>
```

Output:



5. Full screen Modals:

- If you want the modal to span the whole width and height of the page, use the `.modal-fullscreen` class:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>
```

```
<div class="container mt-3">  
  <h3>Fullscreen Modal Example</h3>  
  <p>Click on the button to open the fullscreen modal.</p>  
  
  <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">  
    Open modal  
  </button>  
</div>  
  
<!-- The Modal -->  
<div class="modal" id="myModal">  
  <div class="modal-dialog modal fullscreen">  
    <div class="modal-content">  
  
      <!-- Modal Header -->  
      <div class="modal-header">  
        <h5 class="modal-title">Modal Heading</h5>  
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>  
      </div>  
  
      <!-- Modal body -->  
      <div class="modal-body">  
        Modal body..  
      </div>  
  
      <!-- Modal footer -->
```

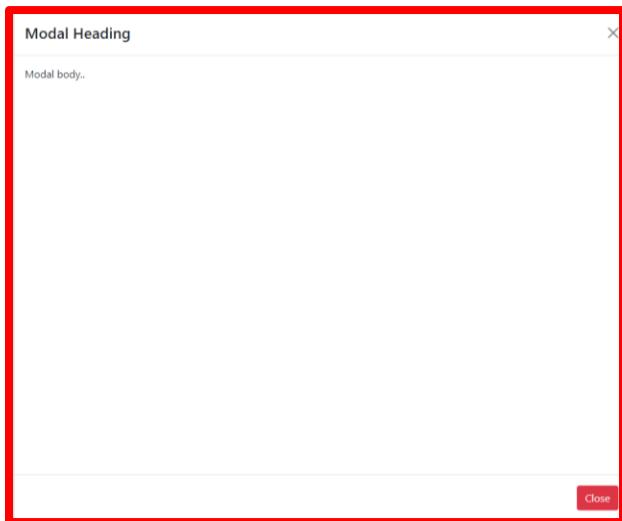


```

<div class="modal-footer">
    <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Close</button>
</div>
</div>
</div>
</body>
</html>

```

Output:



6. Centered Modals:

- Center the modal vertically and horizontally within the page, with the `.modal-dialog-centered` class:

Example:

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>

```

```
<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h3>Vertically Centered Modal Example</h3>

<p>Click on the button to open the modal.</p>

<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">

  Open modal

</button>

</div>
```

```
<!-- The Modal -->

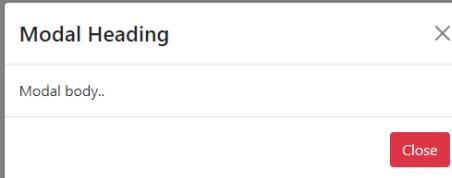
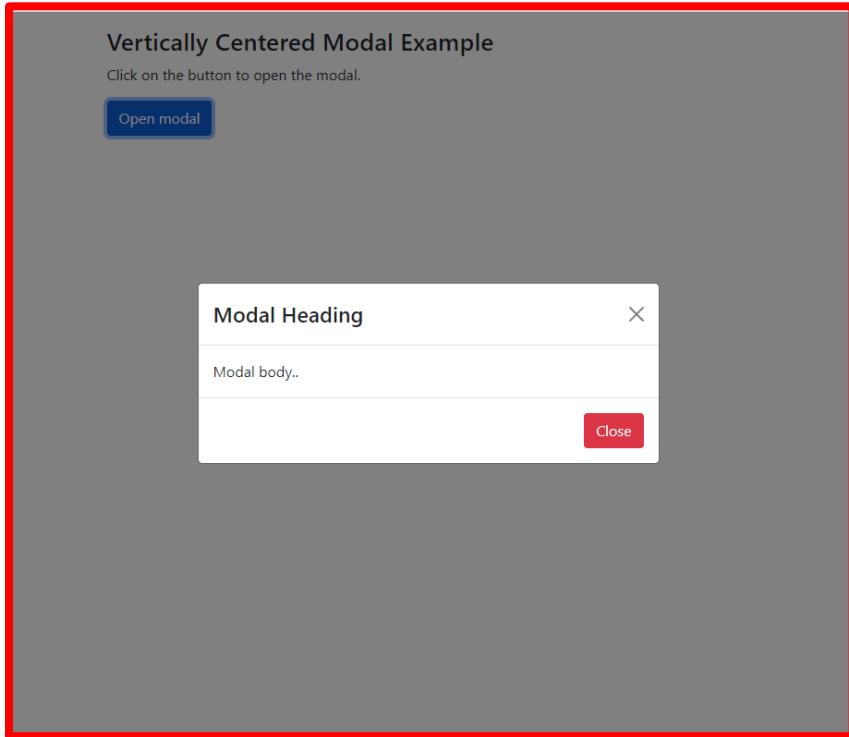
<div class="modal" id="myModal">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">

      <!-- Modal Header -->
      <div class="modal-header">
        <h5 class="modal-title">Modal Heading</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>

      <!-- Modal body -->
      <div class="modal-body">
        Modal body..
      </div>

      <!-- Modal footer -->
      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</body>
</html>
```

Output:



7. Scrolling Modals:

- When you have a lot of content inside the modal, a scrollbar is added to the page. See the examples below to understand it:

Example:

```
<!DOCTYPE html>

<html lang="en">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
```

```
<div class="container mt-3">

    <h2>Modal Scroll Example</h2>

    <p>Click on the button to open the modal.</p>




    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">

        Open modal

    </button>

</div>




<!-- The Modal -->

<div class="modal" id="myModal">

    <div class="modal-dialog">

        <div class="modal-content">

            <!-- Modal Header -->

            <div class="modal-header">

                <h5 class="modal-title">Modal Heading</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

            </div>

            <!-- Modal body -->

            <div class="modal-body">

                <h3>Some text to enable scrolling..</h3>

                <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur</p>

            </div>

        </div>

    </div>

</div>
```

adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

</div>

<!-- Modal footer -->

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-bs-dismiss="modal">Close</button>

</div>

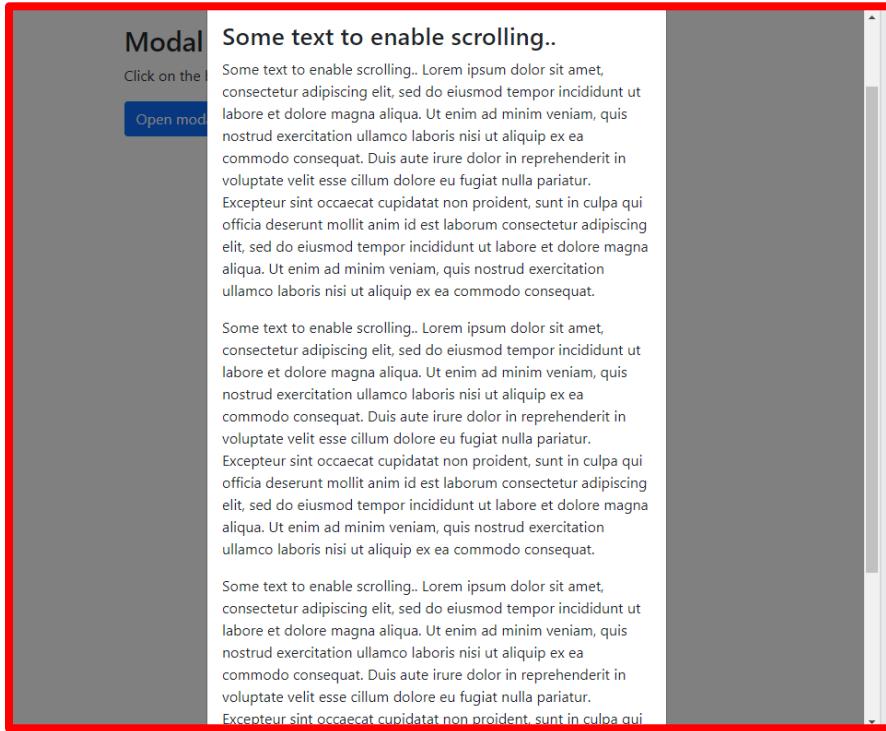
</div>

</div>

</body>

</html>

Output:



CHAPTER 22 – PROGRESS BARS

Bootstrap provides a component for displaying progress bars on a website. Progress bars are typically used to provide feedback to the user on the progress of a given process or action. Users can visualize instantly how much is complete and how much more there is to go.

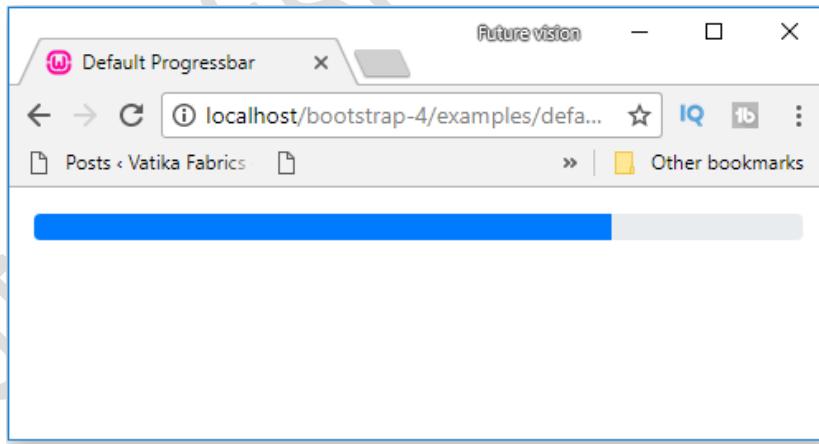
1. Default Progress Bar:

To create a default progress bar, use the **.progress** class with a **.progress-bar** nested inside it.

Example:

```
<div class="container-fluid">  
  
    <div class="progress">  
  
        <div class="progress-bar" role="progressbar" style="width: 75%" aria-  
        valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>  
  
    </div>  
  
</div>
```

Output:



2. Contextual Classes:

You can use the contextual classes to provide extra semantics via color.

Example:

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

    <div class="progress">

        <div class="progress-bar bg-success" role="progressbar" style="width: 25%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>

    </div>

    <br>

    <div class="progress">

        <div class="progress-bar bg-info" role="progressbar" style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>

    </div>

    <br>

    <div class="progress">

        <div class="progress-bar bg-warning" role="progressbar" style="width: 75%" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>

    </div>

    <br>

    <div class="progress">

        <div class="progress-bar bg-danger" role="progressbar" style="width: 100%" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>

    </div>

    <br>

    <div class="progress">

        <div class="progress-bar bg-light" role="progressbar" style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>

    </div> <br>

    <div class="progress">
```

```

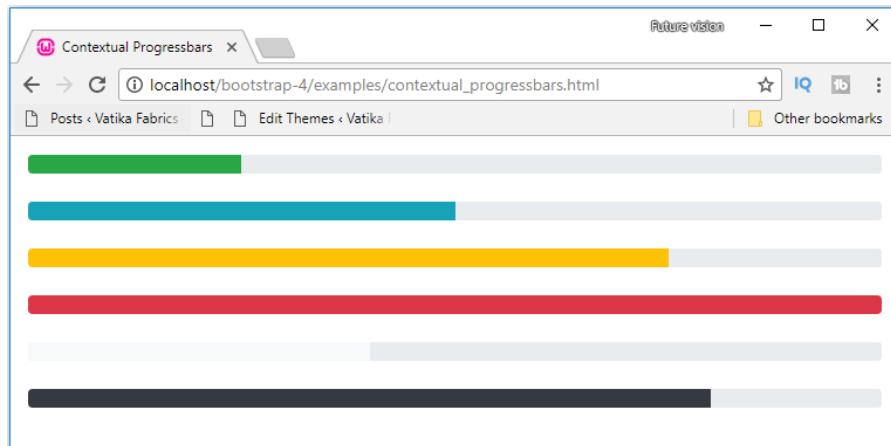
<div class="progress-bar bg-dark" role="progressbar" style="width: 80%" aria-
value="80" aria-valuemin="0" aria-valuemax="100"></div>

</div>

</div>

```

Output:



3. Striped Bar:

You can add stripes by adding the **.progress-bar-striped** class to the **.progress-bar** element.

Example:

```

<div class="container-fluid">

    <div class="progress">

        <div class="progress-bar progress-bar-striped" role="progressbar"
style="width: 10%" aria-valuenow="10" aria-valuemin="0" aria-
valuemax="100"></div>

    </div>

    <br>

    <div class="progress">

        <div class="progress-bar progress-bar-striped bg-success" role="progressbar"
style="width: 25%" aria-valuenow="25" aria-valuemin="0" aria-
valuemax="100"></div>

    </div>

```

```
<br>

<div class="progress">
    <div class="progress-bar progress-bar-striped bg-info" role="progressbar"
        style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-
        valuemax="100"></div>
</div>

<br>

<div class="progress">
    <div class="progress-bar progress-bar-striped bg-warning" role="progressbar"
        style="width: 75%" aria-valuenow="75" aria-valuemin="0" aria-
        valuemax="100"></div>
</div>

<br>

<div class="progress">
    <div class="progress-bar progress-bar-striped bg-danger" role="progressbar"
        style="width: 100%" aria-valuenow="100" aria-valuemin="0" aria-
        valuemax="100"></div>
</div>

<br>

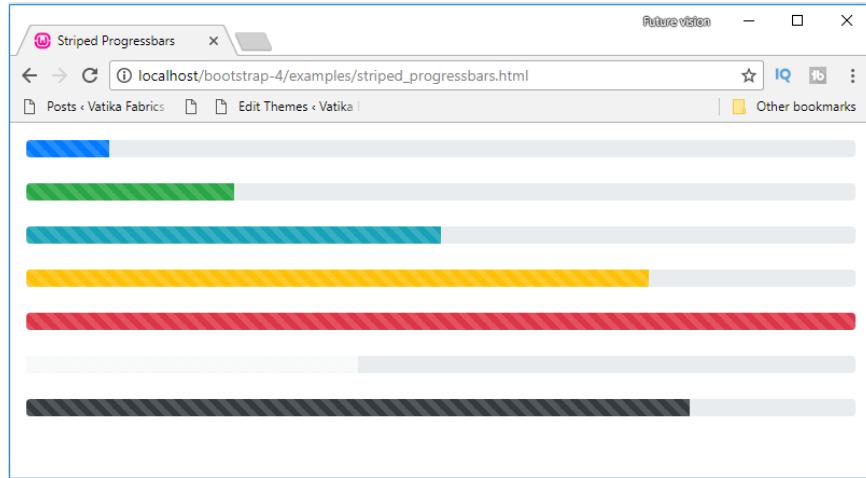
<div class="progress">
    <div class="progress-bar progress-bar-striped bg-light" role="progressbar"
        style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-
        valuemax="100"></div>
</div>

<br>

<div class="progress">
    <div class="progress-bar progress-bar-striped bg-dark" role="progressbar"
        style="width: 80%" aria-valuenow="80" aria-valuemin="0" aria-
        valuemax="100"></div>
</div>

</div>
```

Output:



5. Animated Striped Bar:

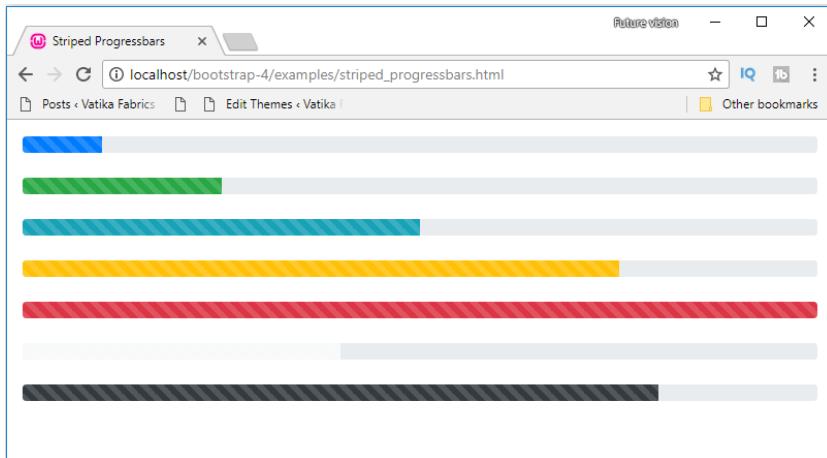
You can animate the stripes by adding the **.progress-animated** class to the **.progress** element.

Note that not all browsers support this feature at the time of writing.

Example:

```
<div class="container-fluid">  
  
    <div class="progress">  
  
        <div class="progress-bar progress-bar-striped progress-bar-animated"  
            role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"  
            style="width: 75%"></div>  
  
    </div>  
  
</div>
```

Output:



5. Multiple Progress Bars:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h2>Multiple Progress Bars</h2>

<p>Create a stacked progress bar by placing multiple bars into the same div with class="progress":</p>

<div class="progress">

<div class="progress-bar bg-success" style="width:50%">
```

```

Free Space

</div>

<div class="progress-bar bg-warning" style="width:10%">
    Warning
</div>

<div class="progress-bar bg-danger" style="width:20%">
    Danger
</div>
</div>

</div>

</body>
</html>

```

Output:

Multiple Progress Bars

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



CHAPTER 23 – LIST GROUPS

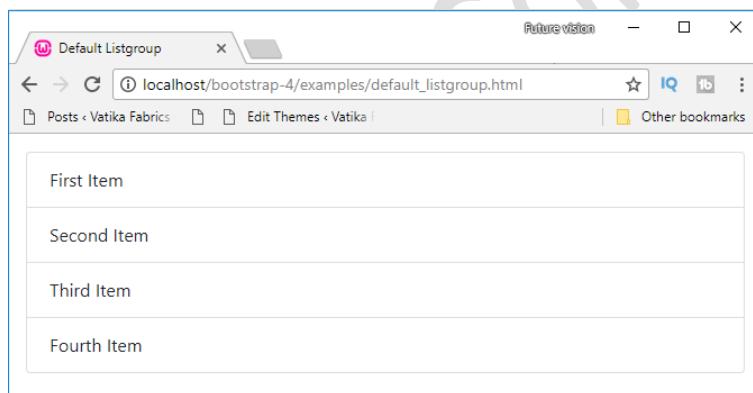
1. Create a Default List Group:

To create a default list group, apply the **.list-group** class to the **** tag, and the **.list-group-item** to each **** tag.

Example:

```
<div class="container-fluid">  
  <ul class="list-group">  
    <li class="list-group-item">First Item</li>  
    <li class="list-group-item">Second Item</li>  
    <li class="list-group-item">Third Item</li>  
    <li class="list-group-item">Fourth Item</li>  
  </ul>  
</div>
```

Output:



2. Linked List Group:

- You can use Bootstrap 5's **.list-group-item-action** to make linked list items. Linked list items result in the whole item being clickable (not just the text portion). The whole list item receives a "hover" style too.
- To create a linked list group, swap the **** tag for a **<div>** tag and the **** tags for **<a>**.

- You can also use Bootstrap's **.active** class to highlight a list item as the current one.

Example:

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

    <div class="list-group">

        <a href="#" class="list-group-item list-group-item-action">First Item</a>

        <a href="#" class="list-group-item list-group-item-action active">Second Item</a>

        <a href="#" class="list-group-item list-group-item-action">Third Item</a>

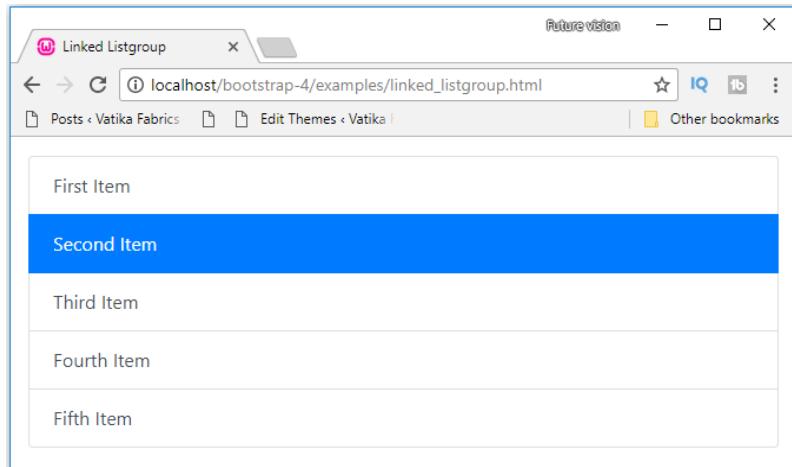
        <a href="#" class="list-group-item list-group-item-action">Fourth Item</a>

        <a href="#" class="list-group-item list-group-item-action">Fifth Item</a>

    </div>

</div>
```

Output:



3. Contextual Classes:

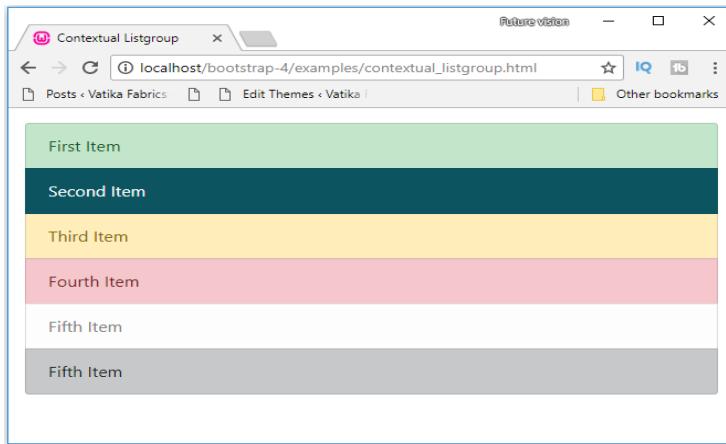
- You can also apply Bootstrap's contextual classes to list group items. To do this, add one of the `.list-group-item-*` classes to the list group item.
- When an item is `.active` its contextual class appears in a darkened version.

Example:

```
<div class="container-fluid">
  <div class="list-group">
    <a href="#" class="list-group-item list-group-item-action list-group-item-success">First Item</a>
    <a href="#" class="list-group-item list-group-item-action active list-group-item-info">Second Item</a>
    <a href="#" class="list-group-item list-group-item-action list-group-item-warning">Third Item</a>
    <a href="#" class="list-group-item list-group-item-action list-group-item-danger">Fourth Item</a>
    <a href="#" class="list-group-item list-group-item-action list-group-item-light">Fifth Item</a>
    <a href="#" class="list-group-item list-group-item-action list-group-item-dark">Fifth Item</a>
  </div>
```

```
</div>
```

Output:



5. Custom Content:

- List groups can contain other HTML elements too. And you can make it all clickable if it's nested inside a `<a>` element.
- Bootstrap 5 has the `.list-group-item-heading` class for headings and the `.list-group-item-text` class for text.

Example:

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

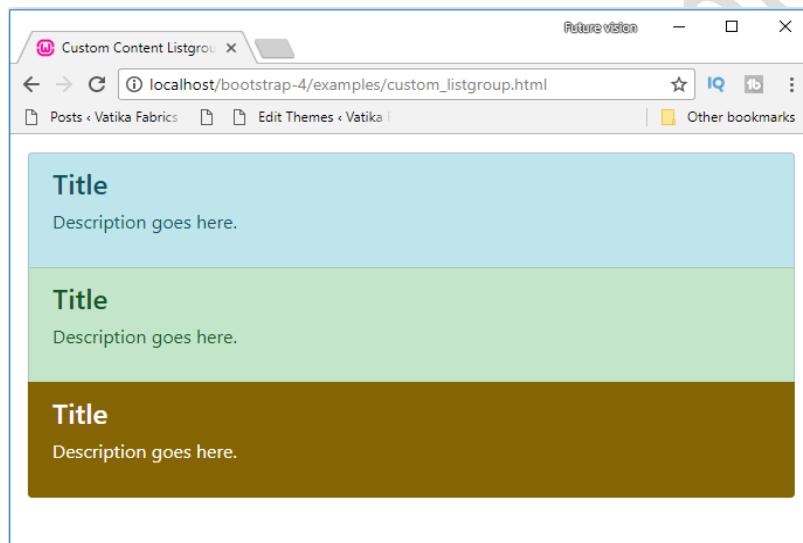
    <div class="list-group">

        <a href="#" class="list-group-item list-group-item-action list-group-item-info">
            <h5>Title</h5>
            <p>Description goes here.</p>
        </a>

        <a href="#" class="list-group-item list-group-item-action list-group-item-success">
            <h5>Title</h5>
            <p>Description goes here.</p>
        </a>
    </div>
</div>
```

```
<a href="#" class="list-group-item list-group-item-action list-group-item-warning active">  
    <h5>Title</h5>  
    <p>Description goes here.</p>  
    </a>  
</div>  
</div>
```

Output:



5. Flush/Remove Border:

- Use the `.list-group-flush` class to remove some borders and rounded corners:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h2>Flush / Remove Borders</h2>

<p>Use the .list-group-flush class to remove some borders and rounded corners:</p>

</div>

<div class="container mt-3">

<ul class="list-group list-group-flush">

<li class="list-group-item">First item</li>

<li class="list-group-item">Second item</li>

<li class="list-group-item">Third item</li>

<li class="list-group-item">Fourth item</li>

</ul>
```

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

Output:

Flush / Remove Borders

Use the `.list-group-flush` class to remove some borders and rounded corners:

```
First item  
Second item  
Third item  
Fourth item
```

6. Numbered List Groups:

- Use the `.list-group-numbered` class to create list items with numbers in front of them:

Example:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<title>Bootstrap Example</title>  
  
<meta charset="utf-8">  
  
<meta name="viewport" content="width=device-width, initial-scale=1">  
  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>  
  
</head>
```

```
<body>

<div class="container mt-3">
    <h2>Numbered List Group</h2>
    <p>Use the .list-group-numbered class to create list items with numbers in front of them:</p>
</div>

<div class="container mt-3">
    <ol class="list-group list-group-numbered">
        <li class="list-group-item">Graphic Designing</li>
        <li class="list-group-item">Website Designing</li>
        <li class="list-group-item">App Development</li>
    </ol>
</div>

</body>
</html>
```

Output:

Numbered List Group

Use the `.list-group-numbered` class to create list items with numbers in front of them:

1. Graphic Designing
2. Website Designing
3. App Development

7. Horizontal List Groups:

- If you want the list items to display horizontally instead of vertically (side by side instead of on top of each other), add the `.list-group-horizontal` class to `.list-group`:

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h2>Horizontal List Groups</h2>

<p>The .list-group-horizontal class displays the list items horizontally instead of vertically:</p>

<ul class="list-group list-group-horizontal">

<li class="list-group-item">Photoshop</li>
```

```
<li class="list-group-item">Coreldraw</li>  
<li class="list-group-item">Illustrator</li>  
<li class="list-group-item">After effects</li>  
</ul>  
</div>
```

```
</body>
```

```
</html>
```

Output:

Horizontal List Groups

The `.list-group-horizontal` class displays the list items horizontally instead of vertically:

```
Photoshop Coreldraw Illustrator After effects
```

8. List Groups with Badges:

- Combine `.badge` classes with utility/helper classes to add badges inside the list group:

Example:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<title>Bootstrap Example</title>  
  
<meta charset="utf-8">  
  
<meta name="viewport" content="width=device-width, initial-scale=1">  
  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```

```
</head>

<body>

<div class="container mt-3">

    <h2>List Group With Badges</h2>

    <p>Combine .badge classes with utility/helper classes to add badges inside the list group:</p>

    <ul class="list-group">

        <li class="list-group-item d-flex justify-content-between align-items-center">
            Facebook
            <span class="badge bg-primary rounded-pill">12</span>
        </li>

        <li class="list-group-item d-flex justify-content-between align-items-center">
            Instagram
            <span class="badge bg-primary rounded-pill">50</span>
        </li>

        <li class="list-group-item d-flex justify-content-between align-items-center">
            Whatsapp
            <span class="badge bg-primary rounded-pill">99</span>
        </li>
    </ul>
</div>
</body>
</html>
```

Output:

List Group With Badges

Combine .badge classes with utility/helper classes to add badges inside the list group:

Facebook

12

Instagram

50

Whatsapp

99

FutureVisionComputers

CHAPTER 25 – CARDS

Cards are new in Bootstrap 5. Cards replace functionality that was previously provided by panels, wells, thumbnails.

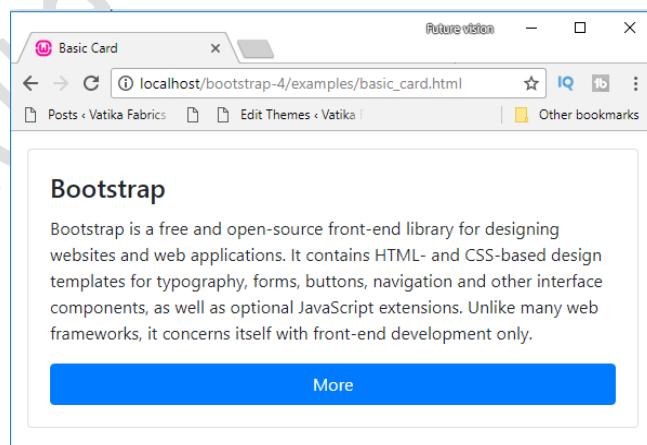
1. Basic Card:

- To create a basic card, apply the **.card** and **.card-body** classes to an element to create the outer card container.
- Add **.card-title** to any heading elements and **.card-text** to text elements.

Example:

```
<div class="container-fluid">  
  
<div class="card card-body">  
  
    <h5 class="card-title">Bootstrap</h5>  
  
    <p class="card-text">Bootstrap is a free and open-source front-end library for  
designing websites and web applications. It contains HTML- and CSS-based design  
templates for typography, forms, buttons, navigation and other interface components,  
as well as optional JavaScript extensions. Unlike many web frameworks, it concerns  
itself with front-end development only.</p>  
  
    <a href="#" class="btn btn-primary">More</a>  
  
</div>
```

Output:



2. Header & Footer:

You can add a header and/or footer by adding a `<div>` with `.card-header` or `.card-footer`

Example:

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

    <div class="card" style="max-width: 20rem;">

        <div class="card-header"> Sponsored Content </div>

        <div class="card-body">

            <h5 class="card-title">Future Vision Computers</h5>

            <p class="card-text">Computer Training Institute. G-50, Navmangalam Complex, Citylight, Surat</p>

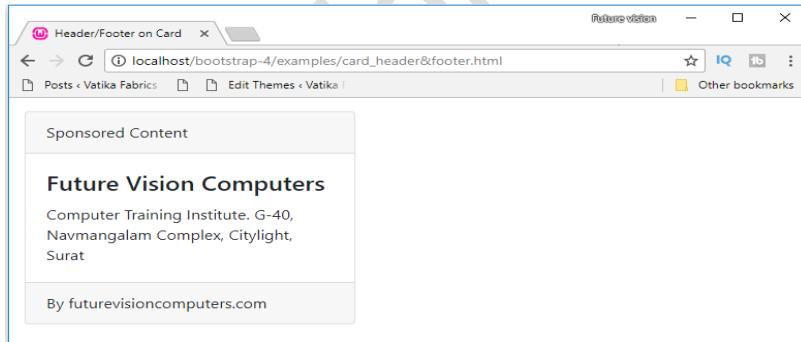
        </div>

        <div class="card-footer"> By futurevisioncomputers.com </div>

    </div>

</div>
```

Output:



Note:

You can also apply the `.card-header` class to any HTML header element (i.e. `<h1>` - `<h6>` tags).

3. Card Image:

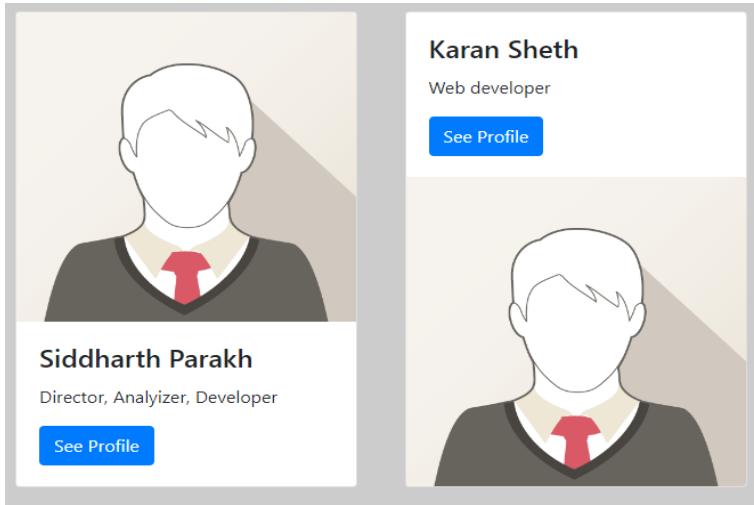


Add **.card-img-top** or **.card-img-bottom** to an to place it at the top or at the bottom inside the card. Note that we have added the image outside of the card-block to span the entire width:

Example:

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-3">
      <div class="card" style="width:300px">
        
        <div class="card-body">
          <h5 class="card-title">Siddharth Parakh</h5>
          <p class="card-text">Director, Analyzer, Developer</p>
          <a href="#" class="btn btn-primary">See Profile</a>
        </div>
      </div>
    </div>
    <div class="col-sm-3">
      <div class="card" style="width:300px">
        <div class="card-body">
          <h5 class="card-title">Karan Sheth</h5>
          <p class="card-text">Web developer</p>
          <a href="#" class="btn btn-primary">See Profile</a>
        
      </div>
    </div>
  </div>
</div>
```

Output:



5. Card Image Overlay:

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

Example:

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

    <div class="card" style="width:300px">

        <div class="card-img-overlay">

            <h5 class="card-title">Siddharth Parakh</h5>

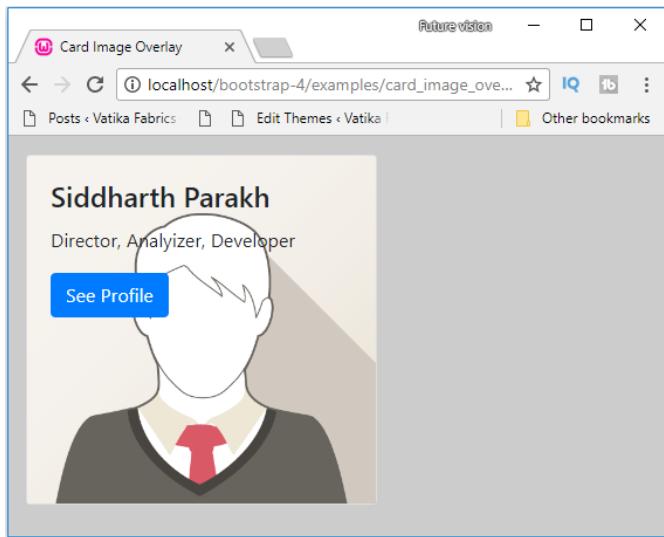
            <p class="card-text">Director, Analyzer, Developer</p>

            <a href="#" class="btn btn-primary">See Profile</a>

        </div>
    </div>

</div>
```

Output:



5. Contextual Cards:

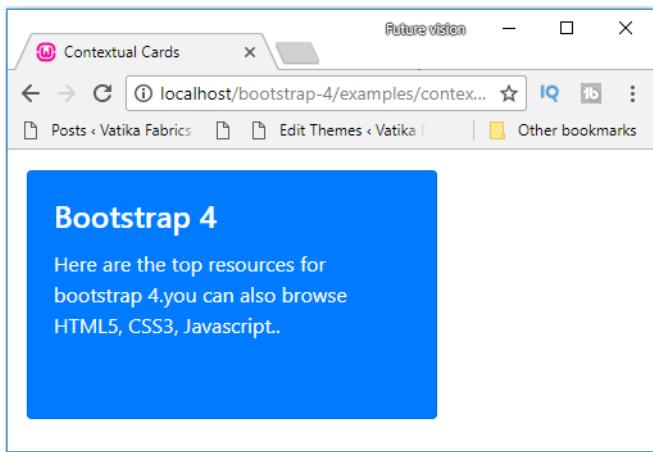
To add a background color the card, use contextual classes (**.bg-primary**, **.bg-success**, **.bg-info**, **.bg-warning**, **.bg-danger**, **.bg-secondary**, **.bg-dark** and **.bg-light**)

Example:

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

    <div class="card text-white bg-primary" style="max-width: 20rem;">
        <div class="card-body">
            <h5 class="card-title">Bootstrap 5</h5>
            <p class="card-text">Here are the top resources for bootstrap 5.you can also browse HTML5, CSS3, Javascript..</p>
            <!-- Links -->
            <a href="#" class="card-link">Sun Gone</a> <a href="#" class="card-link">Still Gone</a>
        </div>
    </div>
</div>
```

Output:



FutureVisionComputers

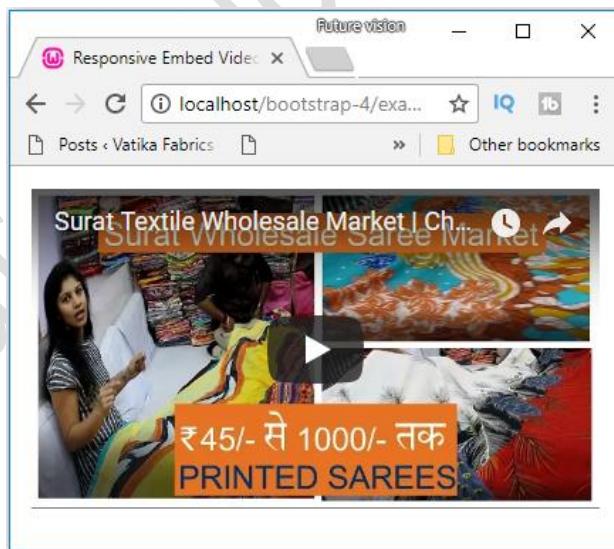
CHAPTER 25 – RESPONSIVE EMBED

- Bootstrap's responsive embed component provides an easy way to scale videos and slide shows to their containing block.
- The responsive embed component can be used on `<iframe>`, `<video>`, `<object>`, and `<embed>` elements.
- To apply the responsive embed component to one of these elements, wrap the element within a `<div>` element with the `.embed-responsive` class, along with an intrinsic ratio class (eg, `.responsive-embed-16by9` or `.responsive-embed-5by3`).

Example:

```
<div class="container-fluid">  
    <div class="embed-responsive embed-responsive-16by9">  
        <iframe class="embed-responsive-item"  
            src="//www.youtube.com/embed/nERIBg2Lfb8" allowfullscreen></iframe>  
    </div>  
</div>
```

Output:



CHAPTER 26 – CAROUSEL

1. Basic Carousel:

- Add scrolling images or text to your website with the Bootstrap carousel component.
- The Bootstrap carousel component enables you to add scrolling images and text that slide in, pause, then slide out. Controls enable the user to scroll forwards or backwards within the set. Basically a scrolling marquee with user controls.

How To create a basic carousel:

- Apply **.carousel** and **.slide** to an outer container (with its own ID).
- For the scrollable contents, wrap all items in a **.carousel-inner** and give each item a **.carousel-item** class.
- Also, you must apply **.active** to one of the slides in the carousel, otherwise the carousel won't be visible. This class allows you to set one slide as the initial slide (i.e. the starting slide).

Example:

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

    <div id="myCarousel" class="carousel slide w-50 ml-auto mr-auto" data-
        ride="carousel">

        <div class="carousel-inner" role="listbox">

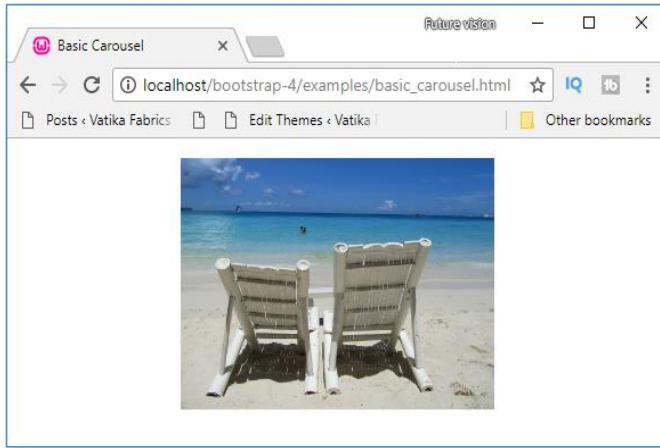
            <div class="carousel-item active">  </div>

            <div class="carousel-item">  </div>

            <div class="carousel-item">  </div>

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

Output:



2. Add Controls:

- To add "Previous/Next" controls, use an `<a>` with `.carousel-control-prev` or `carousel-control-next`. Also add a `` with `.carousel-control-prev-icon` or `.carousel-control-next-icon`.
- To cater for screenreaders, add ``s with the words "Previous" or "Next" and give them the `.sr-only` class. This will hide this text from browsers, but read out the text in screenreaders.

Example:

```

<div class="container-fluid">

    <div id="myCarousel" class="carousel slide w-50 ml-auto mr-auto" data-
ride="carousel">

        <div class="carousel-inner" role="listbox">

            <div class="carousel-item active"></div>

            <div class="carousel-item"></div>

            <div class="carousel-item"></div>

        </div>

        <a class="carousel-control-prev" href="#myCarousel" role="button" data-
slide="prev">

            <span class="carousel-control-prev-icon" aria-hidden="true"></span>

            <span class="sr-only">Previous</span>

        </a>
    
```

```

<a class="carousel-control-next" href="#myCarousel" role="button" data-slide="next">

    <span class="carousel-control-next-icon" aria-hidden="true"></span>

    <span class="sr-only">Next</span>

</a>

</div>

</div>

```

Output:



3. Add Indicators:

To add "indicator" controls (the little clickable areas at the bottom, center), add a list with **.carousel-indicators**, as well as the applicable **.data-target** and **.data-slide-to** attributes.

Example:

```

<div class="container-fluid">

    <div id="myCarousel" class="carousel slide w-50 ml-auto mr-auto" data-ride="carousel">

        <ol class="carousel-indicators">

            <li data-target="#myCarousel" data-slide-to="0" class="active"></li>

            <li data-target="#myCarousel" data-slide-to="1"></li>

            <li data-target="#myCarousel" data-slide-to="2"></li>

        </ol>

        <div class="carousel-inner" role="listbox">

```

```
<div class="carousel-item active"></div>

<div class="carousel-item"></div>

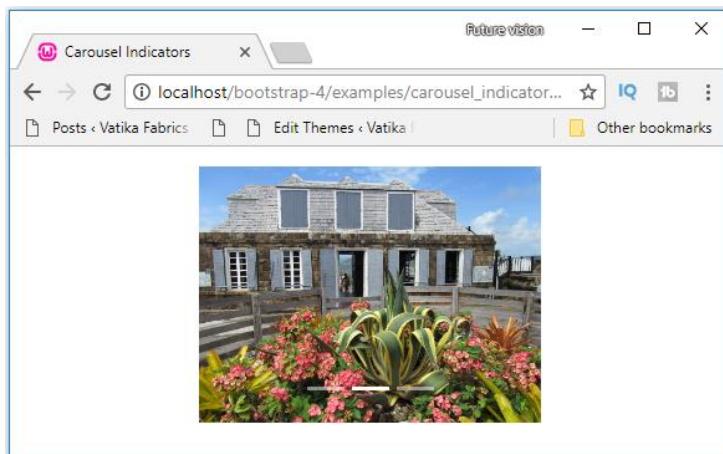
<div class="carousel-item"></div>

</div>

</div>

</div>
```

Output:



5. Add Captions:

You can add captions to your slideshow by adding a `<div>` with the `.carousel-caption` class applied. This goes inside each `.carousel-item`.

Example:

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

    <div id="myCarousel" class="carousel slide w-50 ml-auto mr-auto" data-
ride="carousel">

        <ol class="carousel-indicators">

            <li data-target="#myCarousel" data-slide-to="0" class="active"></li>

            <li data-target="#myCarousel" data-slide-to="1"></li>

            <li data-target="#myCarousel" data-slide-to="2"></li>

        </ol>

        <div class="carousel-inner" role="listbox">

            <div class="carousel-item active">

                <div class="carousel-caption">

                    <h3>First slide</h3>

                    <p>This is the first slide.</p>

                </div>

            </div>

            <div class="carousel-item">

                <div class="carousel-caption">

                    <h3>Second slide</h3>

                    <p>This is the second slide.</p>

                </div>

            </div>

        </div>

    </div>
```

```
<div class="carousel-item"> 

<div class="carousel-caption">

<h3>Third slide</h3>

<p>This is the third slide.</p>

</div>

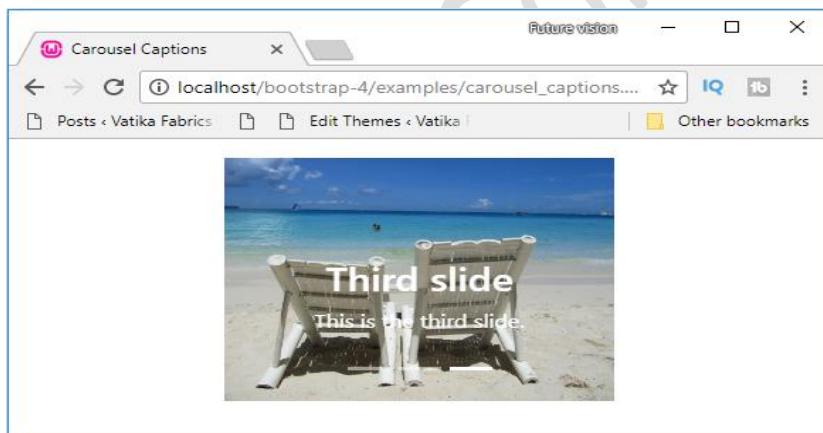
</div>

</div>

</div>

</div>
```

Output:



CHAPTER 27 – TOOLTIPS

Tooltip

- Add a stylized tooltip with the Bootstrap tooltip functionality.
- Bootstrap provides tooltip functionality that allows a stylized tooltip to appear when the user hovers over an item. The text of the tooltip comes from the title attribute.
- Tooltips are generally more suitable for <a> and <button> elements.
- Tooltips rely on the 3rd party library Tether for positioning. Therefore, you need to ensure that you're calling the tether.min.js file on your web page

Tooltip Initialization:

- Bootstrap tooltips are not enabled by default. They require initialization before you can use them.
- One way to do this is to place the following code somewhere after the call to Jquery:

```
<script>

$(function () {

    $('[data-toggle="tooltip"]').tooltip()

})

</script>
```

Tooltip on link:

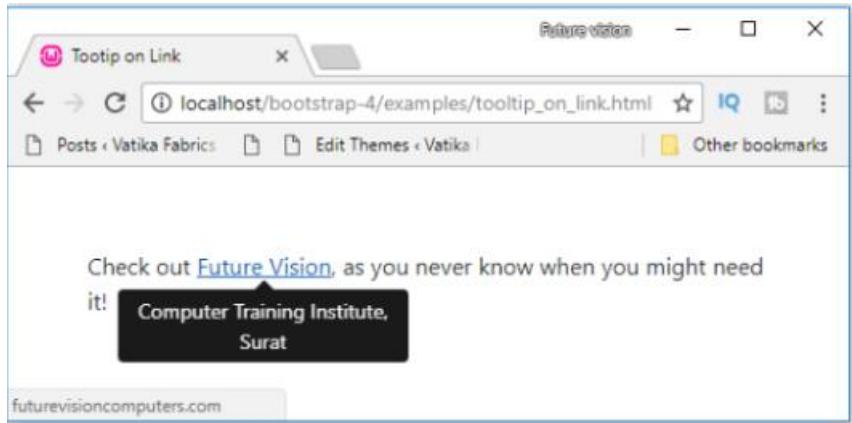
Example:

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

    <p style="margin:50px;">Check out <a target="_blank"
        href="http://futurevisioncomputers.com" data-toggle="tooltip" data-placement="top"
        title="Computer Training Institute, Surat">Future Vision</a>, as you never know when
        you might need it!</p>

</div>
```

Output:



Tooltip on button:

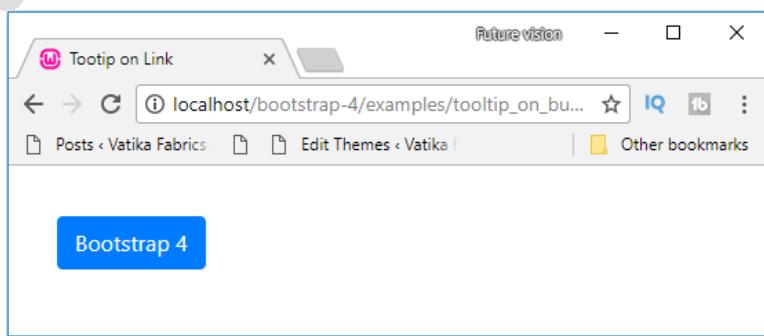
Example:

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

    <button type="button" class="btn btn-primary" data-toggle="tooltip" title="Top
Designing Framework" style="margin:20px;">
        Bootstrap 5
    </button>

</div>
```

Output:



Tooltip Position:

You can specify for the tooltip to appear at the top, bottom, left or right

Example:

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

    <button type="button" class="btn btn-primary" data-toggle="tooltip" data-placement="top" title="New HTML Topics" style="margin:20px;"> HTML

    </button>

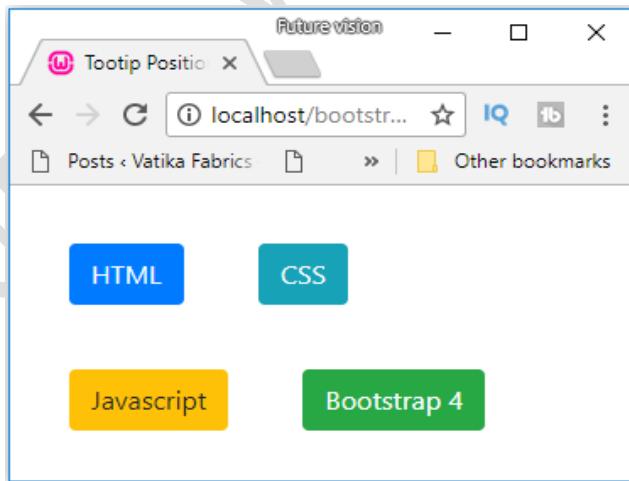
    <button type="button" class="btn btn-info" data-toggle="tooltip" data-placement="right" title="New CSS Topics" style="margin:20px;">
        CSS
    </button>

    <button type="button" class="btn btn-warning" data-toggle="tooltip" data-placement="bottom" title="New Javascript Topics" style="margin:20px;">
        Javascript
    </button>

    <button type="button" class="btn btn-success" data-toggle="tooltip" data-placement="left" title="New Bootstrap5 Topics" style="margin:20px;">
        Bootstrap 5
    </button>

</div>
```

Output:



CHAPTER 28 – POPOVERS

- Bootstrap provides a popover component that displays small overlay content, similar to that found in iOS.

- Popovers rely on the 3rd party library Tether for positioning. Therefore, you need to ensure that you're calling the popper.js file on your web page .
- Popovers also require the tooltips plugin.

1. Popover Initialization:

Bootstrap popovers are not enabled by default. They require initialization before you can use them.

One way to do this is to place the following code somewhere after the call to JQuery:

```
<script>
$(function () {
    $('[data-toggle="popover"]').popover()
})
</script>
```

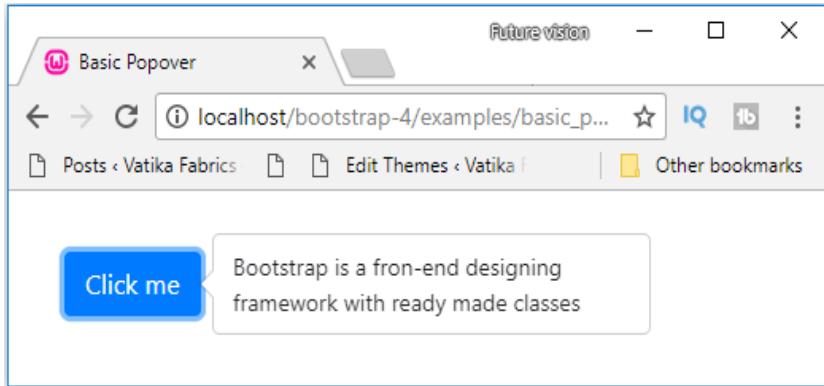
2. Basic Popover:

Use **data-toggle="popover"** to specify the popover, and **data-content=""** to provide the content for the popover.

Example:

```
<div class="container-fluid">
    <button type="button" class="btn btn-primary" data-toggle="popover" data-
        content="Bootstrap is a front-end designing framework with ready made classes"
        style="margin:20px;">
        Click me
    </button>
</div>
```

Output:



3. Popover Title:

Any **title** attribute you provide on the element will become the title of the popup.

Example:

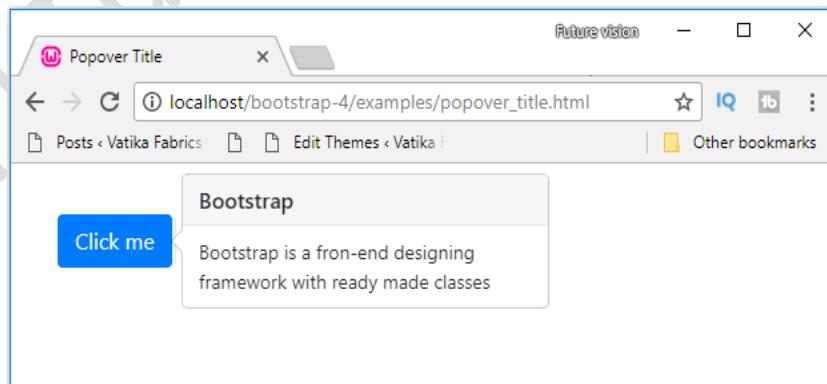
```
<div class="container-fluid">

    <button type="button" class="btn btn-primary" title="Bootstrap" data-
        toggle="popover" data-content="Bootstrap is a front-end designing framework
        with ready made classes" style="margin:20px;">

        Click me
    </button>

</div>
```

Output:



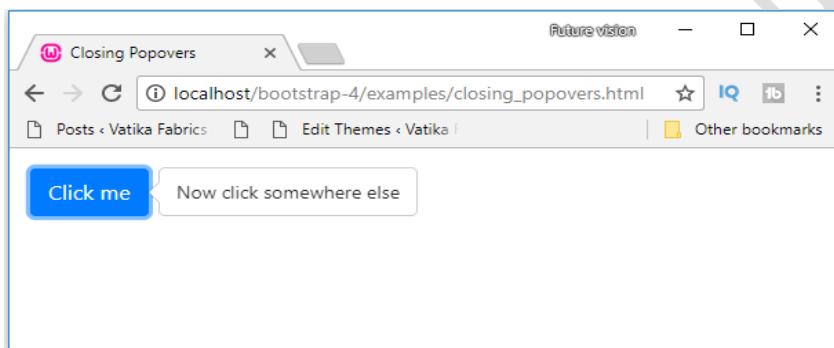
4. Closing Popovers:

- By default, popovers will close when the user clicks the element again (i.e. the one that triggered the popover).
- You can make a popover close when the user clicks anywhere within the document (as opposed to the element that triggered it).
- To close a popover like this, use **data-trigger="focus"**:

Example:

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

Output:



5. Popovers on Hover:

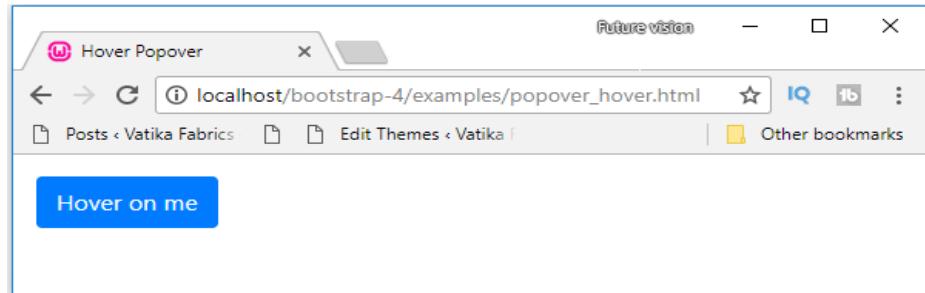
You can change it so that the popover is triggered by a mouseover event, rather than a click.

To do this, use **data-trigger="hover"**:

Example:

```
<button type="button" class="btn btn-primary" data-toggle="popover" data-trigger="hover" data-placement="right" data-content="Now hover out.">  
    Hover over me  
</button>
```

Output:



FutureVisionComputers

CHAPTER 29 – SCROLLSPY

The Bootstrap Scrollspy component allows you to automatically update nav targets based on scroll position.

Scrollspy allows you to **highlight the current position in a menu, based on the user's scroll position**. As the user scrolls down the page, the applicable menu item is highlighted, based one where the scroll position is.

Scrollspy Example:

Here's an example of Scrollspy in action. Scroll down the page/frame and see each menu item highlighted as you reach the corresponding position in the page:

Example:

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

    <nav id="scrollspy-nav" class="navbar navbar-default">
        <ul class="nav nav-pills">
            <li class="nav-item"><a class="nav-link" href="#html">HTML</a></li>
            <li class="nav-item"><a class="nav-link" href="#css">CSS</a></li>
            <li class="nav-item"><a class="nav-link" href="#bootstrap">Bootstrap</a></li>
        </ul>
    </nav>

    <article data-spy="scroll" data-target="#scrollspy-nav" data-offset="0"
    style="overflow: scroll; width: 60vw; height: 70vh; font-size: 5vw;">
        <h5 id="html">HTML</h5>
        <p>HTML5 is the next major revision of the HTML standard superseding HTML 5.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.</p>
        <h5 id="css">CSS</h5>
        <p>CSS is used to control the style of a web document in a simple and easy way. CSS is the acronym for "Cascading Style Sheet". This tutorial covers both the versions CSS1, CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.</p>
        <h5 id="bootstrap">Bootstrap</h5>
    </article>

```

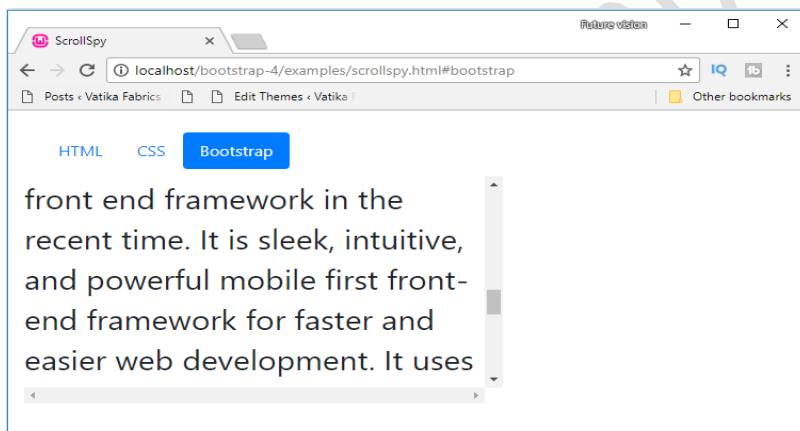
```
<p>Bootstrap is the most popular front end framework in the recent time. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript. This tutorial will teach you the basics of Bootstrap Framework using which you can create web projects with ease. The tutorial is divided into sections such as Bootstrap Basic Structure, Bootstrap CSS, Bootstrap Layout Components and Bootstrap Plugins. Each of these sections contain related topics with simple and useful examples.</p>

</div>

</article>

</div>
```

Output:



CHAPTER 30 – Off Canvas

- Off canvas is similar to modals (hidden by default and shown when activated), except that is often used as a sidebar navigation menu.

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="offcanvas offcanvas-start" id="demo">

<div class="offcanvas-header">

<h1 class="offcanvas-title">Heading</h1>

<button type="button" class="btn-close" data-bs-dismiss="offcanvas"></button>

</div>

<div class="offcanvas-body">

<p>Some text lorem ipsum.</p>

<p>Some text lorem ipsum.</p>

<p>Some text lorem ipsum.</p>

<button class="btn btn-secondary" type="button">A Button</button>

</div>

</div>


```

```

</div>

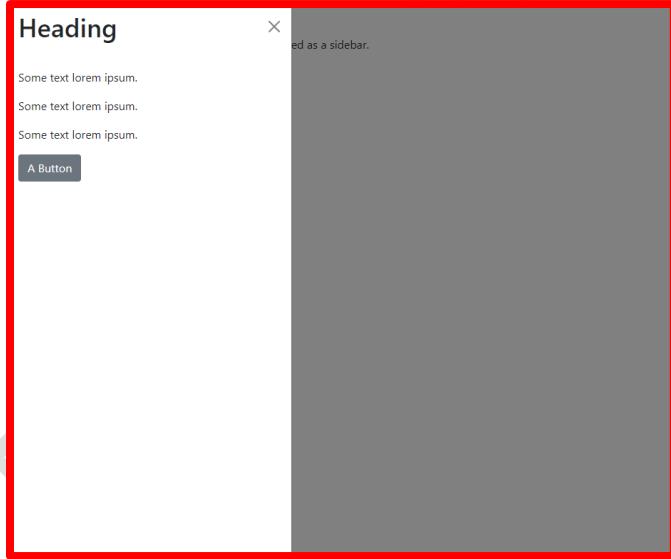
</div>

<div class="container-fluid mt-3">
    <h3>Offcanvas Sidebar</h3>
    <p>Offcanvas is similar to modals, except that it is often used as a sidebar.</p>
    <button class="btn btn-primary" type="button" data-bs-toggle="offcanvas" data-bs-target="#demo">
        Open Offcanvas Sidebar
    </button>
</div>

</body>
</html>

```

Output:



Example Explained

The `.offcanvas` class creates the offcanvas sidebar.

The `.offcanvas-start` class positions the offcanvas, and makes it 500px wide. See examples below for more positioning classes.

The `.offcanvas-title` class ensures proper margins and line-height.

Then, add your content inside the `.offcanvas-body` class.

To open the offcanvas sidebar, you must use a `<button>` or an `<a>` element that points to the id of the `.offcanvas` container (`#demo` in our example).

To open the offcanvas sidebar with an `<a>` element, you can point to `#demo` with the `href` attribute, instead of `data-bs-target` attribute.

2. Off Canvas Position:

Use the `.offcanvas-start|end|top|bottom` to position the offcanvas to the left, right, top or bottom:

Example of top position:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <title>Bootstrap Example</title>

    <meta charset="utf-8">

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

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

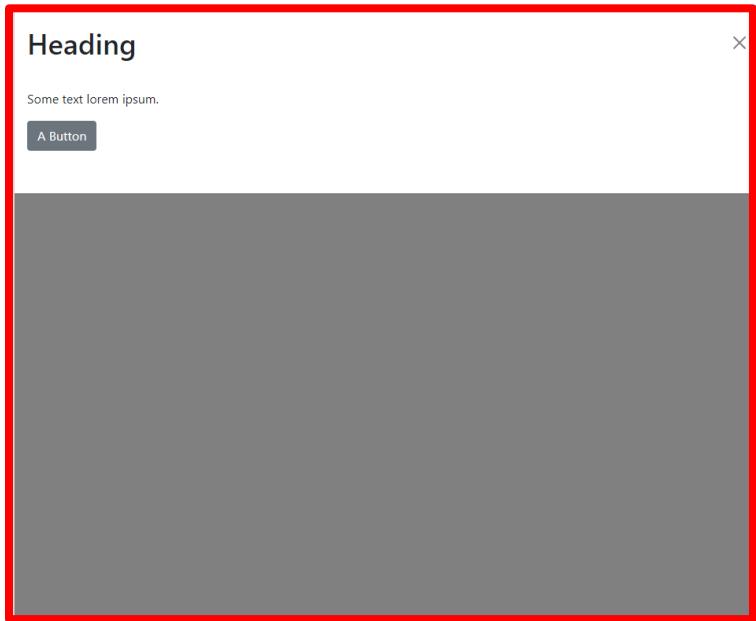
<div class="offcanvas offcanvas-top" id="demo">

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

```
<h1 class="offcanvas-title">Heading</h1>
<button type="button" class="btn-close" data-bs-dismiss="offcanvas"></button>
</div>
<div class="offcanvas-body">
<p>Some text lorem ipsum.</p>
<button class="btn btn-secondary" type="button">A Button</button>
</div>
</div>

<div class="container-fluid mt-3">
<h3>Top Offcanvas</h3>
<p>The .offcanvas-top class positions the offcanvas at the top of the page.</p>
<button class="btn btn-primary" type="button" data-bs-toggle="offcanvas" data-bs-target="#demo">
    Toggle Top Offcanvas
</button>
</div>
</body>
</html>
```

Output:



FutureVision Computers

CHAPTER 31 – Spinners

- To create a spinner/loader, use the `.spinner-border` class:

Example:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Bootstrap Example</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>
    <div class="container mt-3">
      <h2>Spinners</h2>
      <p>To create a spinner/loader, use the <code>.spinner-border</code> class:</p>
      <div class="spinner-border"></div>
    </div>
  </body>
</html>
```

Output:

Spinners

To create a spinner/loader, use the `.spinner-border` class:



Colored Spinner Example:

- Use any **text color utilites** to add a color to the spinner:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Bootstrap Example</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>

    <div class="container mt-3">
      <h2>Colored Spinners</h2>
      <p>Use any <strong>text color utilites</strong> to add a color to the spinner:</p>
      <div class="spinner-border text-muted"></div>
    </div>
  </body>
</html>
```

```
<div class="spinner-border text-primary"></div>
<div class="spinner-border text-success"></div>
<div class="spinner-border text-info"></div>
<div class="spinner-border text-warning"></div>
<div class="spinner-border text-danger"></div>
<div class="spinner-border text-secondary"></div>
<div class="spinner-border text-dark"></div>
<div class="spinner-border text-light"></div>
</div>
</body>
</html>
```

Output:

Colored Spinners

Use any **text color utilities** to add a color to the spinner:



Growing Spinner Example:

- Use the `.spinner-grow` class if you want the spinner/loader to grow instead of "spin":

```
<!DOCTYPE html>

<html>
  <head>
    <title>Bootstrap Example</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>

    <div class="container mt-3">
      <h2>Growing Spinners</h2>
      <p>Use the <code>.spinner-grow</code> class if you want the spinner/loader to grow instead of "spin":</p>

      <div class="spinner-grow text-muted"></div>
      <div class="spinner-grow text-primary"></div>
      <div class="spinner-grow text-success"></div>
      <div class="spinner-grow text-info"></div>
      <div class="spinner-grow text-warning"></div>
      <div class="spinner-grow text-danger"></div>
      <div class="spinner-grow text-secondary"></div>
      <div class="spinner-grow text-dark"></div>
      <div class="spinner-grow text-light"></div>
    </div>
  </body>

```

```
</div>
```

```
</body>
```

```
</html>
```

Output:

Growing Spinners

Use the `.spinner-grow` class if you want the spinner/loader to grow instead of "spin":



Spinner Button Example:

- You can also add spinners to a button, with or without text:

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Example</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container mt-3">
<h2>Spinner Buttons</h2>
<p>Add spinners to buttons:</p>
<button class="btn btn-primary">
```

```
<span class="spinner-border spinner-border-sm"></span>
</button>

<button class="btn btn-primary">
<span class="spinner-border spinner-border-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>
</html>
```



Output:

Spinner Buttons

Add spinners to buttons:



FutureVisionComputers

CHAPTER 32 – Toasts

- The toast component is like an alert box that is only shown for a couple of seconds when something happens (i.e. when the user clicks on a button, submits a form, etc.).

How to create a toast ?

To create a toast, use the `.toast` class, and add a `.toast-header` and a `.toast-body` inside of it.

Note: Toasts are hidden by default. Use the `.show` class if you want to display it. To close it, use a `<button>` element and add `data-bs-dismiss="toast"`:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container mt-3">

<h3>Toast Example</h3>

<p>A toast is like an alert box that is only shown for a couple of seconds when something happens (i.e. when a user clicks on a button, submits a form, etc.).</p>

<p>In this example, we use the .show class to always show the toast by default. You can close it by clicking on the close (x) icon inside the toast header.</p>

<div class="toast show">

<div class="toast-header">

<strong class="me-auto">Toast Header</strong>
```

```
<button type="button" class="btn-close" data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">
<p>Some text inside the toast body</p>
</div>

</div>

</div>

</body>

</html>
```

Output:

Toast Example

A toast is like an alert box that is only shown for a couple of seconds when something happens (i.e. when a user clicks on a button, submits a form, etc.).

In this example, we use the .show class to always show the toast by default. You can close it by clicking on the close (x) icon inside the toast header.



Open a toast:

- To show a toast with a click of a button, you must initialize it with JavaScript: select the specified element and call the `toast()` method.
- The following code will show all "toasts" in the document when you click on a button:

```
<!DOCTYPE html>

<html lang="en">

<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>

<body>

<div class="container mt-3">
    <h3>Toast Example</h3>
    <p>In this example, we use a button to show the toast message.</p>
    <button type="button" class="btn btn-primary" id="toastbtn">Show Toast</button>
<div class="toast">
    <div class="toast-header">
        <strong class="me-auto">Toast Header</strong>
        <button type="button" class="btn-close" data-bs-dismiss="toast"></button>
    </div>
    <div class="toast-body">
        <p>Some text inside the toast body</p>
    </div>
</div>
</div>
</body>
```

```
<script>

document.getElementById("toastbtn").onclick = function() {

var toastElList = [].slice.call(document.querySelectorAll('.toast'))

var toastList = toastElList.map(function(toastEl) {

    return new bootstrap.Toast(toastEl)

}))

toastList.forEach(toast => toast.show()

}

</script>

</body>

</html>
```

Output:

Toast Example

In this example, we use a button to show the toast message.

Show Toast

Toast Header

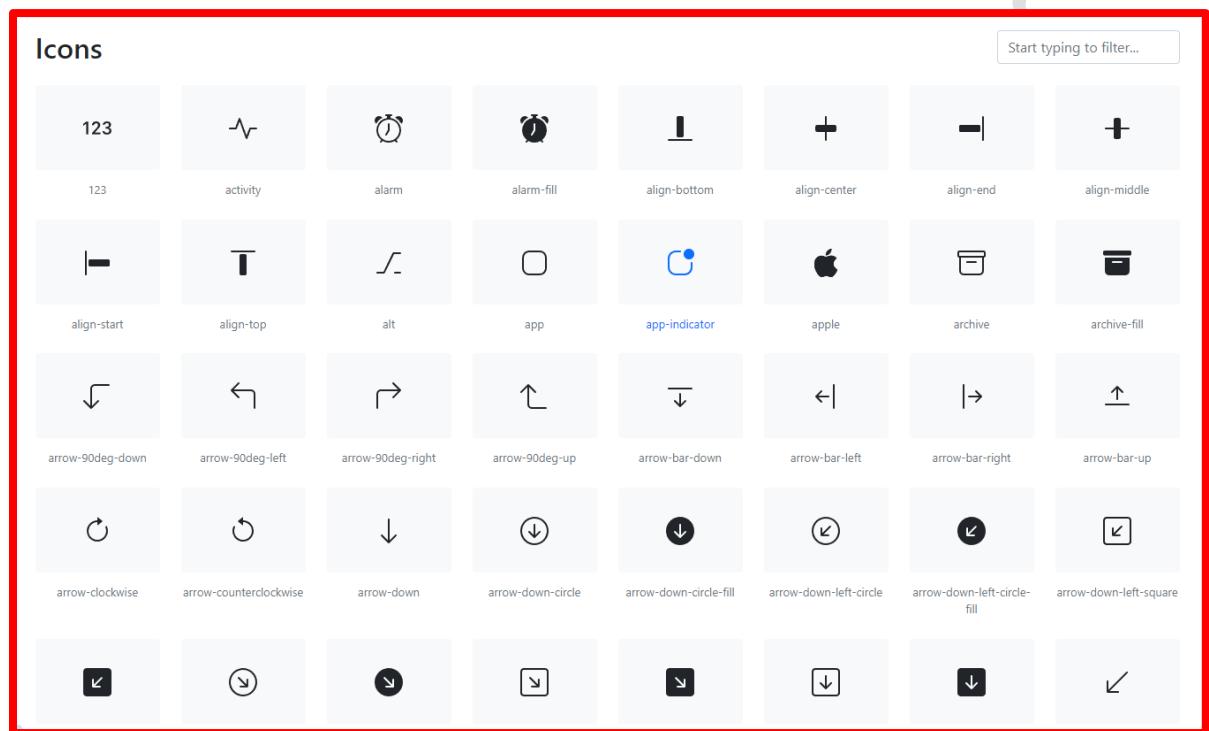
X

Some text inside the toast body

CHAPTER 33 – Icons

Free, high quality, open source icon library with over 1,600 icons. Include them anyway you like—SVGs, SVG sprite, or web fonts. Use them with or without [Bootstrap](#) in any project.

Link: - <https://icons.getbootstrap.com/>



CHAPTER 35 – Bootstrap Landing Page

Create your own bootstrap landing page project to build a great portfolio

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap 5 Website Example</title>

<meta charset="utf-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

<style>

.fakeimg {

height: 200px;

background: #aaa;

}

</style>

</head>

<body>

<div class="p-5 bg-primary text-white text-center">

<h1>Future Vision Computers landing page</h1>

<p>Resize this responsive page to see the effect!</p>


```

```
</div>

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

<div class="container-fluid">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link active" href="#">Active</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#">Disabled</a>

</li>

</ul>

</div>

</nav>
```

```
<div class="container mt-5">

<div class="row">

<div class="col-sm-5">

<h2>About Me</h2>
```



```
<h5>Photo of me:</h5>

<div class="fakeimg">Fake Image</div>

<p>We are teaching since 2006</p>

<h3 class="mt-5">Some Links</h3>

<p>Lorem ipsum dolor sit ame.</p>

<ul class="nav nav-pills flex-column">

<li class="nav-item">

<a class="nav-link active" href="#">Active</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#">Disabled</a>

</li>

</ul>

<hr class="d-sm-none">

</div>

<div class="col-sm-8">

<h2>Web Designing</h2>

<h5>by Future Vision Computers, Apr 20, 2022</h5>

<div class="fakeimg">Fake Image</div>
```

<p>Read more..</p>

<p>Web design encompasses many different skills and disciplines in the production and maintenance of websites. </p>

Mobile App Development

<h5>by Future Vision Computers, Apr 20, 2022</h5>

Fake Image

<p>Read more..</p>

<p>Mobile app development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, enterprise digital assistants or mobile phones.</p>

</div>

</div>

</div>

<p>Footer © Future Vision Computers 2022</p>

</div>

</body>

</html>

Output:

Future Vision Computers landing page
Resize this responsive page to see the effect!

Active Link Link Disabled

About Me
Photo of me:
Fake Image

We are teaching since 2006

Web Designing
by Future Vision Computers, Apr 20, 2022
Fake Image

Read more..

Some Links
Lorem ipsum dolor sit ame.
Active
Link
Link
Disabled

Mobile App Development
by Future Vision Computers, Apr 20, 2022
Fake Image

Mobile App Development
by Future Vision Computers, Apr 20, 2022
Fake Image

Read more..
Mobile app development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, enterprise digital assistants or mobile phones.

Footer © Future Vision Computers 2022