

Bootstrap 5

1.1 About Bootstrap

- Bootstrap is HTML, CSS, and JavaScript framework to quickly create **responsive webpages** by using its built-in components.
- It is a sleek, intuitive front-end framework which follows the **mobile-first approach**.
- It helps in quick development of responsive websites by providing HTML layout and CSS based templates for UI components like Forms, Tables, Navigation menus, Dropdowns, Carousel, etc.
- It uses HTML, CSS, and JS for faster and easier web development.

When we already have CSS for styling, why do we need Bootstrap ?

Despite the various advantages CSS has, it has certain disadvantages as well:

- Though CSS is compatible with all browsers, the level of compatibility varies with each browser.
- All enterprise level applications are huge and difficult to maintain.

1.2 Bootstrap Features

Bootstrap:

- was built at Twitter Inc. by Mark Otto and Jacob Thornton as a framework for internal tools
- is a complete CSS framework offering Grid system and configurations, Typography classes, UI components like forms, tables and more
- is a widely used framework in the development of responsive websites
- has very good documentation which can be found in Bootstrap site
<https://getbootstrap.com/>
- One of the most popular and widely CSS frameworks.

1.3 Responsive web design

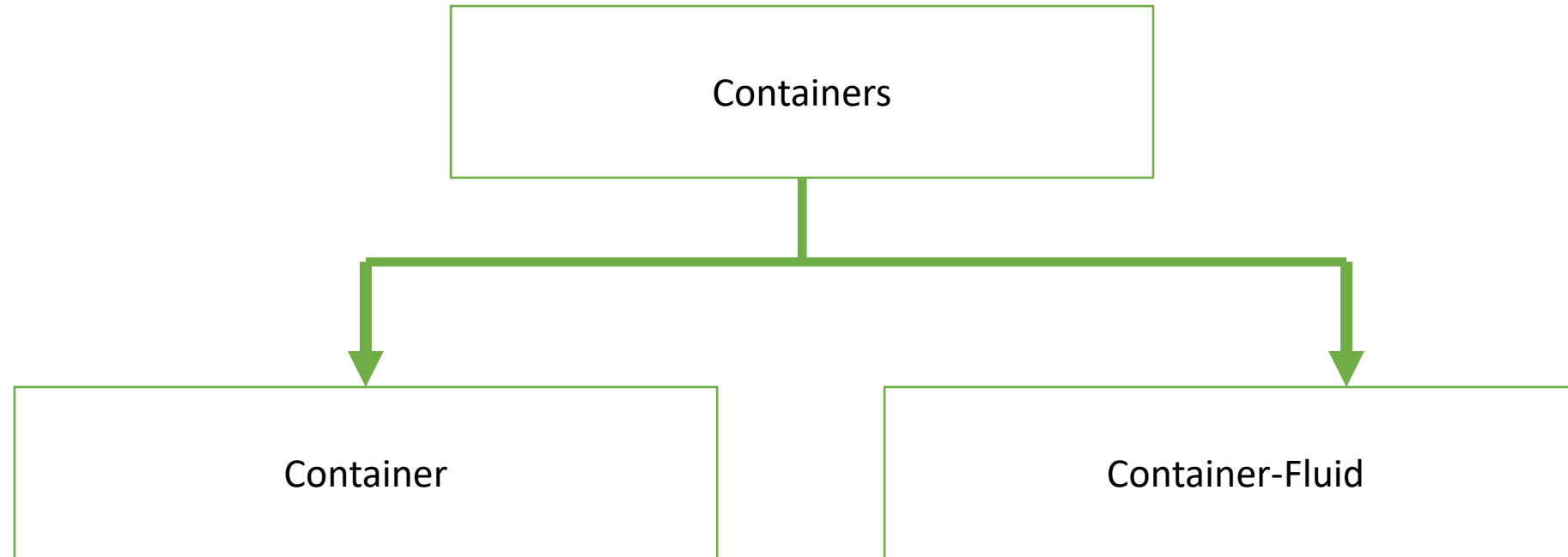
- Responsive web design is all about providing an optimal viewing experience across a wide variety of devices. This means that the entire content of the site should appear without loss of any information along with the maintenance of its appeal, when it is viewed in a mobile or a tablet or a desktop.
- The basic features of responsive web design are as follows:
- **Fluid Layout:** Layout grows and shrinks based on the size of the device browser
- **Flexible Images:** Images adapt to the size of the device browser
- **Responsiveness:** Selectively apply CSS based on the size of browser/device using responsive patterns



1.4 Starter Code

```
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css"
integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMP0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js"
integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
```

1.5 Bootstrap Containers



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

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**

1.5 Bootstrap Containers

According to the width of the viewport, the **.container** class gives fixed width.

container-fluid will constantly resize as you change the width of viewport.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra Large ≥1200px	XXL ≥1400px
max-width	100%	540px	720px	960px	1140px	1320px

A screenshot of a web browser's developer tools, specifically the 'Styles' panel. The panel shows a list of CSS rules for the Bootstrap container class. The rules are organized into media queries based on minimum width. The first rule is for min-width: 1400px, setting max-width: 1320px. The second rule is for min-width: 1200px, setting max-width: 1140px. The third rule is for min-width: 992px, setting max-width: 960px. The fourth rule is for min-width: 768px, setting max-width: 720px. The fifth rule is for min-width: 576px, setting max-width: 540px. The rules are for the .container, .container-lg, .container-md, .container-sm, .container-xl, and .container-xxl classes. The source file is _containers.scss:21.

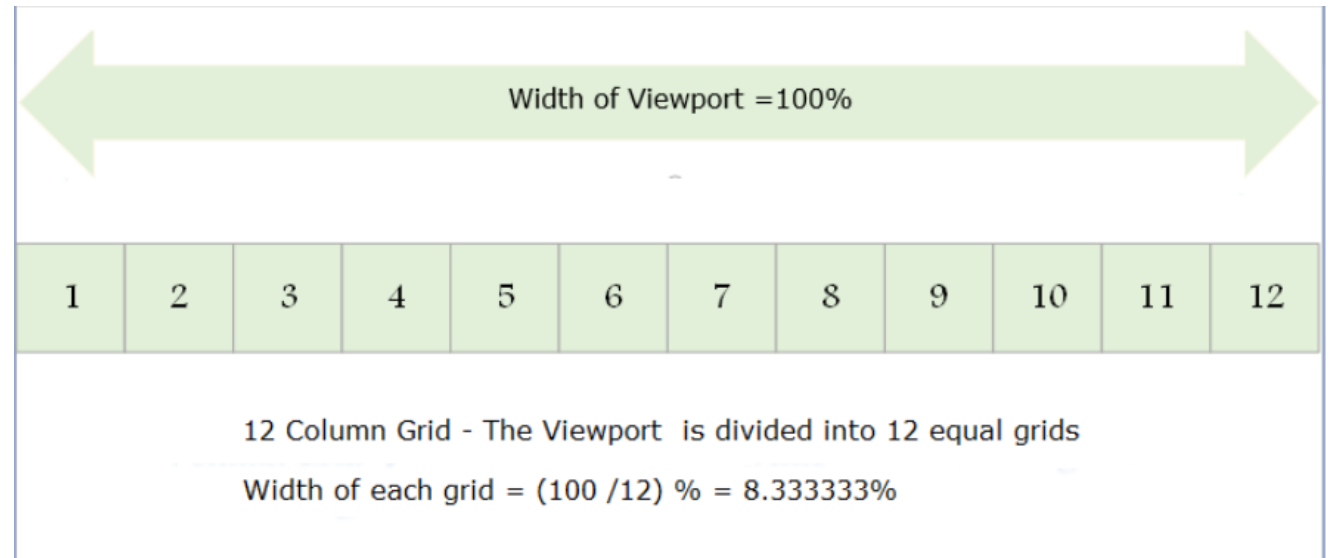
```
Styles Computed Layout Event Listeners >>
Filter :hov .cls + [ ]
@media (min-width: 1400px)
.container, .container-lg, .container-md, .container-sm, .container-xl, .container-xxl {
  max-width: 1320px;
}
@media (min-width: 1200px)
.container, .container-lg, .container-md, .container-sm, .container-xl {
  max-width: 1140px;
}
@media (min-width: 992px)
.container, .container-lg, .container-md, .container-sm {
  max-width: 960px;
}
@media (min-width: 768px)
.container, .container-md, .container-sm {
  max-width: 720px;
}
@media (min-width: 576px)
.container, .container-sm {
  max-width: 540px;
}
```

1.6 Grid System

In Bootstrap, each webpage can be divided into

- rows
- column-grids,

Each row has **12** column-grids.



1.6.1 Row

- To create a row in Bootstrap, `.row` class needs to be used.
- When a row is used inside a container, container's padding adjusts the row's margin, so the entire content of the row is visible.

1.6.2 Column

- To create a column in Bootstrap, `.col-<sq>-<n>` class needs to be used.
- Here `<sq>` is the size qualifier and `<n>` is the number of grids we want the column to occupy.

Size qualifiers can be of four types:

- **lg** (for large devices with screen width $\geq 1200\text{px}$)
- **md** (for medium devices with screen width $< 1200\text{px}$ and $\geq 992\text{px}$)
- **sm** (for small devices with screen width $< 992\text{px}$ and $\geq 768\text{px}$)
- **xs** (for extra small devices with screen width $< 768\text{px}$)

Number of grids in a column can range between 0 to 12.

1.6.3 Grid Overflow



The above scenario is that of an overflow. Overflow can happen in two ways.

1.If you add up all the grids, it is coming up to $2+5+7=14$ grids. This cannot be accommodated in one line. Once you have **.col-md-2** and **.col-md-5**, you are left with 5 grids. However, the next column **.col-md-7** needs 7 grids which cannot be accommodated there. Thus, **it moves to the next line of the same row.**

2.As we can see in the image above, the first column **.col-md-2** is occupying 2 grids, but the text content in it cannot fit into the two grids and so it overflows vertically. So **.col-md-7** moves to the next line under **.col-md-2**.

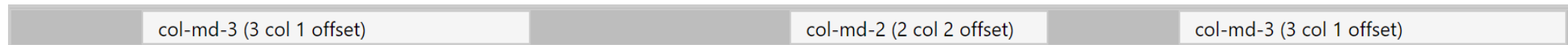
1.6.4 Grid Offset

If we want to add space between the column grid, we can use offset.

We can do this by using `.offset-<n>` class.

```
class="col-md-3 offset-1"
```

This will create a column with 3 grids and to the left of it, will create a column of 1 blank grids.



1.6.5 Layout for multiple devices

Requirement:

The configuration requirement is as follows:

- On desktops, both images must appear one beside the other
- On a tablet screen, each image must appear one below the other, in the center of the screen
- On a mobile screen, the images should be displayed one below the other occupying the entire width of the screen

```
class="col-lg-4 offset-lg-1 col-md-5 offset-md-3 col-sm-12"
```

1.6.6 Typography

1.6.6.1 Font Size

Bootstrap has modified most of the element selectors and has also introduced some classes.

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <small> are the element selectors modified by Bootstrap to make our headings attractive.

1.6.6.2 Display

Bootstrap 4 provides display headings to make headings that stand out better in between text. These headings are created by just adding the **display-x** class to the heading tag.

Display 1

Display 2

Display 3

Display 4

```
<h1 class="display-1">Display 1</h1>  
<h1 class="display-2">Display 2</h1>  
<h1 class="display-3">Display 3</h1>  
<h1 class="display-4">Display 4</h1>
```

1.6.6 Typography

1.6.6.3 Text Alignment

```
<p class="text-left">...</p>  
<p class="text-center">.text-center</p>  
<p class="text-right">.text-right</p>  
<p class="text-justify">...</p>
```

1.6.6.4 Text Emphasis

```
<p class="text-success"> I am a Success Text </p>  
<p class="text-primary"> I am a Primary Text </p>  
<p class="text-info"> I am a Info Text </p>  
<p class="text-warning"> I am a Warning Text </p>  
<p class="text-danger"> I am a Danger Text </p>  
<p class="text-muted"> I am a Muted Text </p>
```

1.6.6.5 Text Transform

```
<p class="text-capitalize"> text capitalize </p>  
<p class="text-uppercase"> text UPPERCASE </p>  
<p class="text-lowercase"> TEXT lowercase </p>
```

1.6.7 Bootstrap Tables

HTML provides tables to present data in a tabular format. Bootstrap provides different classes to make the appearance of these tables attractive.

.table class provides horizontal dividers and small cell padding . In order to group head and body section, we should use <thead> and <tbody> inside <table> element.

.table-bordered along with .table class to add borders.

.table-sm along with .table class to make the table look compact.

.table-hover to the .table class to enable the hover state of tables.

.table-striped to the .table class to provide alternate colors (like zebras)

.table-responsive to get the border and a horizontal scrollbar below the table for those devices only when the entire text content cannot be shown in the available screen width. Above that, there is no effect on the table.

1.6.8 Bootstrap Images

In order to style the images and make them responsive, Bootstrap has provided different classes like `.rounded`, `.rounded-circle`, `.img-thumbnail`, `.img-fluid`.

`class="rounded"`



`class="rounded-circle"`



`class="img-thumbnail"`



1.6.9 Bootstrap Cards

A card in Bootstrap is a bordered box with some padding around its content.

Cards can have a header, body, and footer.

They come in various colors and can align content in various ways.

- Basic Card ***.card*** and ***.card-body*** classes
- Header and Footer in Cards add a header and footer to our card using ***.card-header*** and ***.card-footer*** classes
- We can add a background color to our cards using the different contextual classes like ***.bg-primary***, ***.bg-success***, ***.bg-danger*** etc
- We can add a title to our card by using the ***.card-title*** class to any heading in our card
- We can add images to our cards by using the ***.card-img-top***, ***.card-img-bottom*** or ***.card-img-overlay*** to an `` tag

1.6.10 UI Components

UI (User Interface) components are those components using which a user interacts with a webpage.

HTML pages may have several UI components such as input fields, select menus, text areas, buttons etc. These elements can be styled using HTML itself, but this may not be visually appealing.

In order to overcome this limitation of HTML, Bootstrap provides a wide range of predefined classes for styling UI components.

1.6.10 UI Components

1.6.10.1 Form Control

.form-control can be used for form elements like `<input>`, `<textarea>`, `<select>` can be used. Elements with the class `.form-control` are set to width of 100% of parent element.

.form-check to style radio buttons and checkboxes.

The default alignment on using `.form-check` creates vertically aligned radio buttons.

.form-check-inline will create horizontal alignment to the radio buttons.

1.6.10.2 Buttons

.btn is used for buttons with `<button>` element or `<input type= "button/reset/submit">`.

.btn-block to create block level button (width of button=width of the parent element)

1.6.10.3 Badges

.badge are indicators for numerical values and are generally used to show how many items are there in association with a particular list/link.

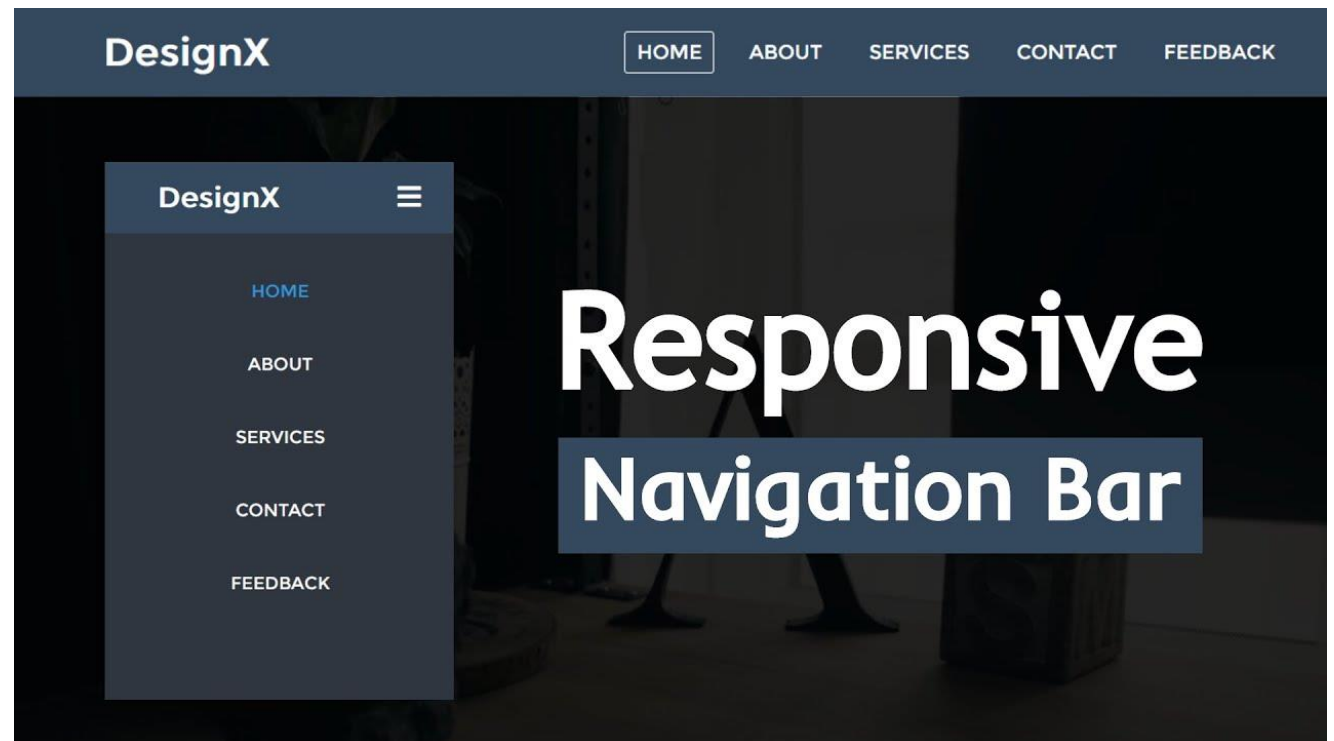
1.6.11 Navigation Bar

Navigation bar (**navbar**) is a section intended to aid users in accessing different webpages of a website. HTML5 provides us with `<nav>` element to create a navigation bar.

Bootstrap has different classes which help to provide better appearance to the navbar.

Navbar Components

1. The **brand** or **logo** is positioned in the top left corner. The brand position is not affected by resizing the window.
2. The **links** in the navbar are visible on large or medium screens. On small screens, the links are placed inside the collapsible drawer.
3. The **navbar toggler button** is used to expand or minimize the drawer on small screens as required.



1.6.11 Navigation Bar

1.6.12 Navbar Brand and Toggle Button

Use the **navbar-brand** class with the <a> tag to add a brand to the top left corner of the navbar.

Use the **navbar-toggler** class with a button to create a collapsible navbar on smaller screens.

- **data-target="#nav-collapse"** points to the div which has the **id="nav-collapse"**

- **data-toggle="collapse"** tells Bootstrap to switch between visual states by collapsing whenever a user interaction happens i.e. whenever the user clicks on the button, it should **"collapse"** something

1.6.13 Navbar Links

- **.navbar-nav** will by default align the links on the left side and keep them all on one line

- adding **.active** or **.disabled** class to the <a> makes a link appear pressed so that it will look as if it is the current page or disabled to make it unavailable

- **id="nav-collapse"** is an identifier for the div which contains all the links (both left and right) so that the button which appears upon collapse can point to this set of links.

- **.collapse** and **.navbar-collapse** give the appropriate styling when collapsed

```
<a class="navbar-brand" href="#">Brand Name</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nav-collapse">
  <span class="navbar-toggler-icon"></span>
</button>
```

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

1.6.11 Navigation Bar

1.6.11.14 Navbar Dark

On adding the **.bg-color** class to the nav tag, the color of the navbar will change as shown.

The standard colors can be applied using *bg-primary*, *bg-secondary*, *bg-dark*, etc.

We can also use the **navbar-dark** class to make all the text white colored. Or use **navbar-light** to make the text black colored.

1.6.11.15 Navbar Links Alignment

.ml-auto is a margin utility class that sets the left-margin to auto and hence pushes elements to the right.

```
<ul class="navbar-nav ml-auto">
  <li class="nav-item"> <a class="nav-link" href="#">Profile</a></li>
  <li class="nav-item"> <a class="nav-link" href="#">Logout</a></li>
</ul>
```

1.6.12 Form Elements

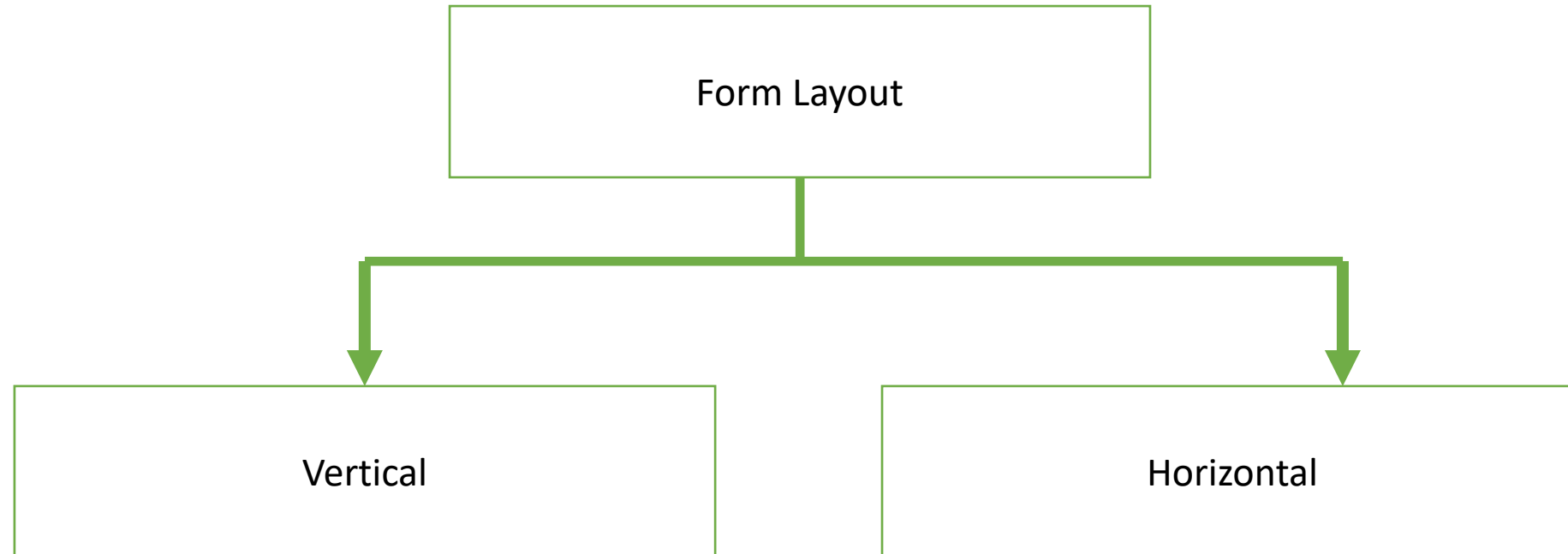
In order to have consistent look and feel and optimum spacing between the form elements, across various resolutions, we can make use of standard layout of the form.

Steps to be followed to achieve the standard layout:

- Every <label> and its corresponding form element should be wrapped in a <div> having **.form-group**
- Every form input element (except button) should have class **.form-control**

```
<div class="form-group">  
  <label>LABEL FOR FORM ELEMENT</label>  
  <form-element class="form-control">FORM ELEMENT</form-element>  
</div>
```


1.16.13 Form Layout



1.6.13.1 Vertical Form Layout

Bootstrap by **default** provides vertical layout to the forms. We need not use any separate Bootstrap class for this (as the <form> element selector has been modified in bootstrap.css).

Here the labels and the corresponding form elements will be **left aligned** and **stacked one below the other**.

1.16.13.2 Horizontal Form Layout

.form-inline class aligns the label and form elements one beside the other in a single row. Each unit consisting of label and form element will be appearing one beside the other.

Inline form

Username Password

Vertical form

Username

Password

1.16.14

Carousel

- Carousel / Slideshow
- The Carousel is a slideshow for cycling through elements.

Class	Description
.carousel	Creates a carousel
.carousel-indicators	Adds indicators for the carousel. These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing)
.carousel-inner	Adds slides to the carousel
.carousel-item	Specifies the content of each slide
.carousel-control-prev	Adds a left (previous) button to the carousel, which allows the user to go back between the slides
.carousel-control-next	Adds a right (next) button to the carousel, which allows the user to go forward between the slides
.carousel-control-prev-icon	Used together with .carousel-control-prev to create a "previous" button
.carousel-control-next-icon	Used together with .carousel-control-next to create a "next" button
.slide	Adds a CSS transition and animation effect when sliding from one item to the next. Remove this class if you do not want this effect

1.6.15 Modals

Modals are built with HTML, CSS, and JavaScript. They're positioned over everything else in the document and remove scroll from the <body> so that modal content scrolls instead.

Clicking on the modal “backdrop” will automatically close the modal.

Bootstrap only supports one modal window at a time. Nested modals aren't supported as we believe them to be poor user experiences.

Modals use position: fixed, which can sometimes be a bit particular about its rendering. Whenever possible, place your modal HTML in a top-level position to avoid potential interference from other elements. You'll likely run into issues when nesting a .modal within another fixed element.

1.6.16 Tooltip

The Tooltip component is small pop-up box that appears when the user moves the mouse pointer over an element

To create a tooltip, add the **data-bs-toggle="tooltip"** attribute to an element.

Use the **title** attribute to specify the text that should be displayed inside the tooltip.

1.6.16.1 Positioning Tooltips

By default, the tooltip will appear on top of the element.

Use the **data-bs-placement** attribute to set the position of the tooltip on top, bottom, left or the right side of the element

1.6.17 Buttons

Bootstrap provides several classes for buttons.

1.6.17.1 Button Style

1.16.17.2 Button Outline

1.16.17.3 Button Size

1.16.7.4 Button Active/Disabled

1.16.7.5 Spinner Button

1.16.7.6 Button Groups

1.16.7.6 Vertical Button Groups

1.16.7.7 Horizontal Button Groups

1.6.18 Progress Bar

To create a default progress bar, add a **.progress** class to a container element and add the **.progress-bar** class to its child element.

Use the CSS **width** property to set the width of the progress bar.

1.6.18.1 Progress Bar Height

1.6.18.2 Progress Bar Labels

1.6.18.3 Colored Progress Bar

1.6.18.4 Striped Progress Bar

1.6.18.5 Animated Progress Bar

1.6.18.6 Multiple Progress Bar

1.6.19 Spinners

To create a spinner/loader, use the .spinner-border class

1.6.19.1 Color Spinners

1.6.19.2 Growing Spinners

1.6.19.3 Spinner Size

1.6.19.4 Spinner Buttons

1.16.20 Pagination

Basic Pagination

If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.

To create a basic pagination, add the **.pagination** class to an `` element. Then add the **.page-item** to each `` element and a **.page-link** class to each link inside ``.

The **.active** class is used to "highlight" the current page.

.disabled class is used for un-clickable links

.pagination-lg for larger blocks or **.pagination-sm** for smaller blocks

Pagination Alignment

1.16.20 Pagination

Basic Pagination

If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.

To create a basic pagination, add the **.pagination** class to an `` element. Then add the **.page-item** to each `` element and a **.page-link** class to each link inside ``.

The **.active** class is used to "highlight" the current page.

.disabled class is used for un-clickable links

.pagination-lg for larger blocks or **.pagination-sm** for smaller blocks

Pagination Alignment

1.16.21 Collapse

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

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

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

1.16.21.1 Accordion

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