

## What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

## Why Use Bootstrap?

Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

## Bootstrap Versions

This tutorial follows **Bootstrap 3**, which was released in 2013. However, we also cover newer versions; Bootstrap 4 (released 2018) and Bootstrap 5 (released 2021).

## Where to Get Bootstrap?

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

You can:

- Download Bootstrap from [getbootstrap.com](https://getbootstrap.com)
- Include Bootstrap from a CDN

MaxCDN provides CDN support for Bootstrap's CSS and JavaScript.

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
```

```
<!-- jQuery library -->
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
```

## Create First Web Page With Bootstrap

### 1. Add the HTML5 doctype

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.

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

### 2. Bootstrap 3 is mobile-first

Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

```
<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.

### 3. Containers

Bootstrap also requires a containing element to wrap site contents. There are two container classes to choose from:

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

### Creating Containers with Bootstrap

Containers are the most basic layout element in Bootstrap and are required when using the grid system. Containers are basically used to wrap content with some padding. They are also used to align the content horizontally center on the page in case of fixed width layout.

Bootstrap provides three different types containers:

- `.container`, which has a max-width at each responsive breakpoint.
- `.container-fluid`, which has 100% width at all breakpoints.
- `.container-{breakpoint}`, which has 100% width until the specified breakpoint.

Classes Bootstrap Grid System	X-Small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px

<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

## Creating Fluid Containers

You can use the `.container-fluid` class to create a full width container. The width of the fluid container will always be 100% irrespective of the devices or screen sizes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Fluid Containers</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
  <div class="container-fluid">
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </div>
</body>
</html>
```

## Specify Responsive Breakpoints for Containers

Since Bootstrap v4.4, you can also create containers that is 100% wide until the specified breakpoint is reached, after which max-width for each of the higher breakpoints will be applied.

For example, `.container-xl` will be 100% wide until the xl breakpoint is reached (i.e., viewport width  $\geq$  1200px), after which max-width for xl breakpoint is applied, which is 1140px.

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Containers with Responsive Breakpoints</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
  <div class="container-sm border py-3 my-3">100% wide until screen size less
than 576px</div>
  <div class="container-md border py-3 my-3">100% wide until screen size less
than 768px</div>
  <div class="container-lg border py-3 my-3">100% wide until screen size less
than 992px</div>
  <div class="container-xl border py-3 my-3">100% wide until screen size less
than 1200px</div>
  <p class="m-3"><strong>Tip:</strong> Open the output in a new blank tab
(Click the arrow next to "Show Output" button) and resize the browser window to
understand how the Bootstrap responsive grid system works.</p>
</body>
</html>

```

## Adding Background and Borders to Containers

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Set Background and Borders to Bootstrap Containers</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
  <!-- Container with dark background and white text color -->
  <div class="container bg-dark text-white border py-3 my-3">
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </div>

  <!-- Container with light background -->
  <div class="container bg-light py-3 my-3">
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </div>

```

```

<!-- Container with border -->
<div class="container border py-3 my-3">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
</body>
</html>

```

## Applying Paddings and Margins to Containers

By default, containers have padding of 12px on the left and right sides, and no padding on the top and bottom sides.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Set Paddings and Margins to Bootstrap Containers</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
  <!-- Container with border, extra paddings and margins -->
  <div class="container border py-3 my-3">
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
  </div>
</body>
</html>

```

## What is Bootstrap Grid System?

Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes. It is built with **flexbox** with mobile-first approach. You can use the Bootstrap's predefined grid classes for quickly making the layouts for different types of devices like mobile phones, tablets, laptops, desktops, and so on. For example, you can use the `.col-*` classes to create grid columns for extra small devices like mobile phones in portrait mode, and the `.col-sm-*` classes for mobile phones in landscape mode.

Similarly, you can use the `.col-md-*` classes to create grid columns for medium screen devices like tablets, the `.col-lg-*` classes for devices like small laptops, the `.col-xl-*` classes for laptops and desktops, and the `.col-xxl-*` classes for large desktop screens.

Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
Number of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

## Creating Two Column Layouts

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Two Column Grid Layouts for Tablets and Desktops</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .demo-content{
    padding: 15px;
    font-size: 18px;
    background: #dbdfe5;
    margin-bottom: 15px;
  }
  .demo-content.bg-alt{
    background: #abb1b8;
  }
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">Open the output in a new blank tab (Click the
  arrow next to "Show Output" button) and resize the browser window to understand
  how the Bootstrap responsive grid system works.</div>

```

```

<div class="container">
  <!--Row with two equal columns-->
  <div class="row">
    <div class="col-md-6">
      <div class="demo-content">.col-md-6</div>
    </div>
    <div class="col-md-6">
      <div class="demo-content bg-alt">.col-md-6</div>
    </div>
  </div>

  <!--Row with two columns divided in 1:2 ratio-->
  <div class="row">
    <div class="col-md-4">
      <div class="demo-content">.col-md-4</div>
    </div>
    <div class="col-md-8">
      <div class="demo-content bg-alt">.col-md-8</div>
    </div>
  </div>

  <!--Row with two columns divided in 1:3 ratio-->
  <div class="row">
    <div class="col-md-3">
      <div class="demo-content">.col-md-3</div>
    </div>
    <div class="col-md-9">
      <div class="demo-content bg-alt">.col-md-9</div>
    </div>
  </div>
</div>
</body>
</html>

```

## Creating Three Column Layouts

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Three Column Grid Layouts for Tablets (landscape) and
Desktops</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */

```

```

.demo-content{
  padding: 15px;
  font-size: 18px;
  background: #dbdfe5;
  margin-bottom: 15px;
}
.demo-content.bg-alt{
  background: #abb1b8;
}
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">Open the output in a new blank tab (Click the
  arrow next to "Show Output" button) and resize the browser window to understand
  how the Bootstrap responsive grid system works.</div>
  <div class="container">
    <!--Row with three equal columns-->
    <div class="row">
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content bg-alt">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
    </div>

    <!--Row with three columns divided in 1:4:1 ratio-->
    <div class="row">
      <div class="col-lg-2">
        <div class="demo-content">.col-lg-2</div>
      </div>
      <div class="col-lg-8">
        <div class="demo-content bg-alt">.col-lg-8</div>
      </div>
      <div class="col-lg-2">
        <div class="demo-content">.col-lg-2</div>
      </div>
    </div>

    <!--Row with three columns divided unevenly-->
    <div class="row">
      <div class="col-lg-3">
        <div class="demo-content">.col-lg-3</div>
      </div>

```



```

        <div class="col-lg-7">
            <div class="demo-content bg-alt">.col-lg-7</div>
        </div>
        <div class="col-lg-2">
            <div class="demo-content">.col-lg-2</div>
        </div>
    </div>
</div>
</body>
</html>

```

## Bootstrap Auto-layout Columns

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Auto-layout Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
    /* Some custom styles to beautify this example */
    .row{
        margin-top: 1rem;
        background: #dbdfe5;
    }
    .col{
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0,0,0,0.5)
    }
</style>
</head>
<body>
<div class="container">
    <!--Row with two equal columns-->
    <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
    </div>

    <!--Row with three equal columns-->
    <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
        <div class="col">.col</div>
    </div>

```

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

## Creating Multi-Column Layouts with Bootstrap

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Grid Layouts for Large Devices</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  p{
    padding: 50px;
    font-size: 32px;
    font-weight: bold;
    text-align: center;
    background: #dbdfe5;
  }
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">Open the output in a new blank tab (Click the
  arrow next to "Show Output" button) and resize the browser window to understand
  how the Bootstrap responsive grid system works.</div>
  <div class="container-lg">
    <div class="row">
      <div class="col-xl-4"><p>Box 1</p></div>
      <div class="col-xl-4"><p>Box 2</p></div>
      <div class="col-xl-4"><p>Box 3</p></div>
      <div class="col-xl-4"><p>Box 4</p></div>
      <div class="col-xl-4"><p>Box 5</p></div>
      <div class="col-xl-4"><p>Box 6</p></div>
      <div class="col-xl-4"><p>Box 7</p></div>
      <div class="col-xl-4"><p>Box 8</p></div>
      <div class="col-xl-4"><p>Box 9</p></div>
      <div class="col-xl-4"><p>Box 10</p></div>
      <div class="col-xl-4"><p>Box 11</p></div>
      <div class="col-xl-4"><p>Box 12</p></div>
    </div>
  </div>
```

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

## Nesting of Grid Columns

The Bootstrap grid columns are also nestable, that means you can put rows and columns inside an existing column. However, the formula for placing the columns will be the same, i.e. the sum of column numbers should be equal to 12 or less within a single row.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Nested Rows and Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
    /* Some custom styles to beautify this example */
    .main-content{
        min-height: 230px;
        background: #dbdfe5;
    }
    .sidebar-content{
        min-height: 100px;
        margin-bottom: 30px;
        background: #b4bac0;
    }
</style>
</head>
<body>
<div class="container mt-3">
    <div class="row">
        <div class="col-sm-8">
            <div class="main-content"></div>
        </div>
        <div class="col-sm-4">
            <!--Nested rows within a column-->
            <div class="row">
                <div class="col-12">
                    <div class="sidebar-content"></div>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-6">
            <div class="sidebar-content"></div>
        </div>
    </div>
</div>
```

```

        <div class="col-6">
            <div class="sidebar-content"></div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</body>
</html>

```

## Creating Variable Width Columns

You can use the `col-{breakpoint}-auto` classes to size columns based on the natural width of their content. Try out the following example to see how it works:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Sizing Columns Based on the Width of their Content</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
<style>
    /* Some custom styles to beautify this example */
    .row{
        margin-top: 1rem;
        background: #dbdfe5;
    }
    [class^=col]{
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0,0,0,0.5)
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row justify-content-md-center">
            <div class="col-md-3">Column left</div>
            <div class="col-md-auto">Variable width column</div>
            <div class="col-md-3">Column right</div>
        </div>
        <div class="row">
            <div class="col">Column left</div>
            <div class="col-auto">Variable width column</div>
            <div class="col">Column right</div>
        </div>
    </div>

```

```

</div>
<p class="text-center mt-3"><strong>Note:</strong> Enter some more content
inside the variable width column to understand how it works.</p>
</body>
</html>

```

## Alignment of Grid Columns

You can use the flexbox alignment utilities to vertically and horizontally align grid columns inside a container. Try out the following examples to understand how it works:

### Vertical Alignment of Grid Columns

You can use the classes `.align-items-start`, `.align-items-center`, and `.align-items-end` to align the grid columns vertically at the top, middle and bottom of a container, respectively.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Vertical Alignments of Grid Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
    /* Some custom styles to beautify this example */
    .row{
        min-height: 10rem;
        margin-top: 1rem;
        background: #dbdfe5;
    }
    .col{
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0,0,0,0.5)
    }
</style>
</head>
<body>
<div class="container">
    <div class="row align-items-start">
        <div class="col">Column one</div>
        <div class="col">Column two</div>
        <div class="col">Column three</div>
    </div>
    <div class="row align-items-center">
        <div class="col">Column one</div>
        <div class="col">Column two</div>
        <div class="col">Column three</div>

```

```

</div>
<div class="row align-items-end">
  <div class="col">Column one</div>
  <div class="col">Column two</div>
  <div class="col">Column three</div>
</div>
</div>
</body>
</html>

```

### Another Example

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Vertical Alignments of Individual Grid Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .row{
    min-height: 10rem;
    margin-top: 1rem;
    background: #dbdfe5;
  }
  .col{
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0,0,0,0.5)
  }
</style>
</head>
<body>
<div class="container">
  <div class="row">
    <div class="col align-self-start">Column one</div>
    <div class="col align-self-center">Column two</div>
    <div class="col align-self-end">Column three</div>
  </div>
</div>
</body>
</html>

```

### Creating Three Column Layouts

Similarly, you can create other layouts based on the above principle. For instance, the following example will typically create three column layouts for laptops and desktops screens. It also works in tablets in landscape mode if screen resolution is more than or equal to 992 pixels (e.g. Apple iPad). However, in portrait mode the grid columns will be horizontal as usual.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Three Column Grid Layouts for Tablets (landscape) and
Desktops</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .demo-content{
    padding: 15px;
    font-size: 18px;
    background: #dbdfe5;
    margin-bottom: 15px;
  }
  .demo-content.bg-alt{
    background: #abb1b8;
  }
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">Open the output in a new blank tab (Click the
arrow next to "Show Output" button) and resize the browser window to understand
how the Bootstrap responsive grid system works.</div>
  <div class="container">
    <!--Row with three equal columns-->
    <div class="row">
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content bg-alt">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
    </div>
  </div>
```

```

<!--Row with three columns divided in 1:4:1 ratio-->
<div class="row">
  <div class="col-lg-2">
    <div class="demo-content">.col-lg-2</div>
  </div>
  <div class="col-lg-8">
    <div class="demo-content bg-alt">.col-lg-8</div>
  </div>
  <div class="col-lg-2">
    <div class="demo-content">.col-lg-2</div>
  </div>
</div>

<!--Row with three columns divided unevenly-->
<div class="row">
  <div class="col-lg-3">
    <div class="demo-content">.col-lg-3</div>
  </div>
  <div class="col-lg-7">
    <div class="demo-content bg-alt">.col-lg-7</div>
  </div>
  <div class="col-lg-2">
    <div class="demo-content">.col-lg-2</div>
  </div>
</div>
</div>
</body>
</html>

```

## Bootstrap Auto-layout Columns

You can also create *equal width columns* for all devices (x-small, small, medium, large, x-large, and xx-large) through simply using the class `.col`, without specifying any column number.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Auto-layout Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .row{
    margin-top: 1rem;
    background: #dbdfe5;

```



```

    }
    .col{
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0,0,0,0.5)
    }
</style>
</head>
<body>
<div class="container">
    <!--Row with two equal columns-->
    <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
    </div>

    <!--Row with three equal columns-->
    <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
        <div class="col">.col</div>
    </div>
</div>
</body>
</html>

```

## Creating a Simple Table with Bootstrap

You can create tables with basic styling that has horizontal dividers and small cell padding (8px by default), by just adding the Bootstrap's class `.table` to the `<table>` element.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Simple Table</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">

</head>
<body>
<div class="m-4">
    <table class="table">
        <thead>
            <tr>
                <th>#</th>
                <th>First Name</th>

```

```

        <th>Last Name</th>
        <th>Email</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```

## Creating Accented Tables

Bootstrap even provides a handful of contextual classes such as `.table-primary`, `.table-secondary`, `.table-success`, `.table-danger`, `.table-warning`, `.table-info`, `.table-light` and `.table-dark` to color tables, table rows or individual cells.

For example, you can create a dark version of the table (i.e. table with light text on dark backgrounds) by adding the contextual class `.table-dark` to the `.table` base class, like this:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Dark Table</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>

```

```

<body>
<div class="m-4">
  <table class="table table-dark">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>

```

Similarly, you can use other contextual classes. For instance, the following example uses the class `.table-success` on the `.table` to create green colored variant of a table.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Accented Tables</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">

```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/  
bootstrap.bundle.min.js"></script>
```

```
</head>
```

```
<body>
```

```
<div class="m-4">
```

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

```
    <thead>
```

```
      <tr>
```

```
        <th>#</th>
```

```
        <th>First Name</th>
```

```
        <th>Last Name</th>
```

```
        <th>Email</th>
```

```
      </tr>
```

```
    </thead>
```

```
    <tbody>
```

```
      <tr>
```

```
        <td>1</td>
```

```
        <td>Clark</td>
```

```
        <td>Kent</td>
```

```
        <td>clarkkent@mail.com</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>2</td>
```

```
        <td>Peter</td>
```

```
        <td>Parker</td>
```

```
        <td>peterparker@mail.com</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>3</td>
```

```
        <td>John</td>
```

```
        <td>Carter</td>
```

```
        <td>johncarter@mail.com</td>
```

```
      </tr>
```

```
    </tbody>
```

```
  </table>
```

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

```
    <thead>
```

```
      <tr>
```

```
        <th>#</th>
```

```
        <th>First Name</th>
```

```
        <th>Last Name</th>
```

```
        <th>Email</th>
```

```
      </tr>
```

```
    </thead>
```

```
    <tbody>
```

```
      <tr>
```

```
        <td>1</td>
```

```

        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

```

<table class="table table-success">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>

```

</table>

```
<table class="table table-danger">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>
```

```
<table class="table table-warning">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
```

```
</tr>
<tr>
  <td>2</td>
  <td>Peter</td>
  <td>Parker</td>
  <td>peterparker@mail.com</td>
</tr>
<tr>
  <td>3</td>
  <td>John</td>
  <td>Carter</td>
  <td>johncarter@mail.com</td>
</tr>
</tbody>
</table>
```

```
<table class="table table-info">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>
```

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

```

<thead>
  <tr>
    <th>#</th>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Email</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>1</td>
    <td>Clark</td>
    <td>Kent</td>
    <td>clarkkent@mail.com</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Peter</td>
    <td>Parker</td>
    <td>peterparker@mail.com</td>
  </tr>
  <tr>
    <td>3</td>
    <td>John</td>
    <td>Carter</td>
    <td>johncarter@mail.com</td>
  </tr>
</tbody>
</table>
</div>
</body>
</html>

```

Similar to the tables you can also use these contextual classes to emphasize the rows within a table. Here's an example of a table with emphasized rows, let's take a look:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Table with Accented Rows</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>

```



```
<div class="m-4">
  <table class="table">
    <thead>
      <tr>
        <th>#</th>
        <th>Bill</th>
        <th>Payment Date</th>
        <th>Payment Status</th>
      </tr>
    </thead>
    <tbody>
      <tr class="table-primary">
        <td>1</td>
        <td>Credit Card</td>
        <td>04/07/2021</td>
        <td>Waiting for statement</td>
      </tr>
      <tr class="table-secondary">
        <td>2</td>
        <td>Insurance</td>
        <td>02/07/2021</td>
        <td>Cancelled</td>
      </tr>
      <tr class="table-success">
        <td>3</td>
        <td>Water</td>
        <td>01/07/2021</td>
        <td>Paid</td>
      </tr>
      <tr class="table-info">
        <td>4</td>
        <td>Internet</td>
        <td>05/07/2021</td>
        <td>Change plan</td>
      </tr>
      <tr class="table-warning">
        <td>5</td>
        <td>Electricity</td>
        <td>03/07/2021</td>
        <td>Pending</td>
      </tr>
      <tr class="table-danger">
        <td>6</td>
        <td>Telephone</td>
        <td>06/07/2021</td>
        <td>Due</td>
      </tr>
      <tr class="table-light">
```

```

        <td>7</td>
        <td>Car Service</td>
        <td>08/07/2021</td>
        <td>Call in to confirm</td>
    </tr>
    <tr class="table-dark">
        <td>8</td>
        <td>Gas</td>
        <td>06/07/2021</td>
        <td>Payment failed</td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```

## Creating Tables with Striped Rows

You can also add zebra-stripping to the table rows within the `<tbody>` by simply adding an additional class `.table-striped` to the `.table` base class, as shown below:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Zebra-Striped Tables</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
    <table class="table table-striped">
        <thead>
            <tr>
                <th>#</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Clark</td>
                <td>Kent</td>

```

```

        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

```

<table class="table table-striped table-dark">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
        </tr>
        <tr>
            <td>3</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
        </tr>
    </tbody>
</table>

```

```
<table class="table table-striped table-primary">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>
```

```
<table class="table table-striped table-secondary">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
```

```

        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

```

<table class="table table-striped table-success">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

```

```

<table class="table table-striped table-danger">
  <thead>
    <tr>

```

```

        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

```

<table class="table table-striped table-warning">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>

```

```

        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

```

<table class="table table-striped table-info">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
        </tr>
        <tr>
            <td>3</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
        </tr>
    </tbody>
</table>

```

```

<table class="table table-striped table-light">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>

```

```

        <th>Email</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```

## Creating Borderless Tables

You can also create borderless tables using the class `.table-borderless` on the `.table` element.

```

<table class="table table-borderless">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Clark</td>

```



```

        <td>Kent</td>
        <td>clarkkent@mail.com</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

## Enabling Hover State on Table Rows

You can also enable a hover state on table rows within a `<tbody>` element by adding the modifier class `.table-hover` to the `.table` base class.

```

<table class="table table-hover">
    <thead>
        <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
        </tr>
        <tr>
            <td>2</td>

```

```

        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
    </tr>
    <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>

```

## Creating List Groups with Bootstrap

The list groups are very useful and flexible component for displaying lists of elements in a beautiful manner. In most basic form a list group is simply an **unordered list** with the class `.list-group` whereas, the list items having the class `.list-group-item`.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap List Group</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
    <ul class="list-group w-50">
        <li class="list-group-item">Pictures</li>
        <li class="list-group-item">Documents</li>
        <li class="list-group-item">Music</li>
        <li class="list-group-item">Videos</li>
    </ul>

```

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

## Indicate Disabled and Active Items

You can simply add the class `.active` to a `.list-group-item` to indicate the current active selection. Similarly, you can add `.disabled` to a `.list-group-item` to make it look like disabled.

```
<ul class="list-group w-50">
  <li class="list-group-item active">Pictures</li>
  <li class="list-group-item">Documents</li>
  <li class="list-group-item">Music</li>
  <li class="list-group-item disabled">Videos</li>
</ul>
```

## Edge-to-Edge List Groups

You can optionally add the class `.list-group-flush` to the `list-group` element to remove outer borders and rounded corners to create list groups that are edge-to-edge with their parent container.

```
<ul class="list-group list-group-flush w-50">
  <li class="list-group-item">Pictures</li>
  <li class="list-group-item">Documents</li>
  <li class="list-group-item">Music</li>
  <li class="list-group-item">Videos</li>
</ul>
```

## Creating Numbered List Groups

You can also create list groups where items are numbered through simply adding the modifier class `.list-group-numbered` on the `.list-group` element, like this:

```
<ol class="list-group list-group-numbered w-50">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
</ol>
```

## List Group with Checkboxes and Radios

You can also place Bootstrap's custom checkboxes and radio buttons within the list group items.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap List Group with Checkboxes</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="list-group w-50">
    <label class="list-group-item">
      <input type="checkbox" class="form-check-input me-1"
name="hobbies"> Music
    </label>
    <label class="list-group-item">
```

```

        <input type="checkbox" class="form-check-input me-1"
name="hobbies"> Travel & Adventure
    </label>
    <label class="list-group-item">
        <input type="checkbox" class="form-check-input me-1"
name="hobbies"> Reading
    </label>
</div>
</div>
</body>
</html>

```

Similarly, you can place custom radio buttons within list group items, like this:

```

<div class="list-group w-50">
    <label class="list-group-item">
        <input type="radio" class="form-check-
input me-1" name="gender"> Male
    </label>
    <label class="list-group-item">
        <input type="radio" class="form-check-
input me-1" name="gender"> Female
    </label>
</div>

```

## List Group with Linked Items

You can also [link](#) list group items with the little change in HTML markup.

Just replace the `<li>` with `<a>` tag and use `<div>` element as a parent instead of `<ul>`. You can also add [icons](#) and [badges](#) to this list group to make it more elegant. Here's an example:

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap List Group with Linked Items</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="list-group w-50">
    <a href="#" class="list-group-item list-group-item-action active">
      <i class="bi-house-fill"></i> Home
    </a>
    <a href="#" class="list-group-item list-group-item-action">
      <i class="bi-camera-fill"></i> Pictures
      <span class="badge rounded-pill bg-primary float-end">145</
span>
    </a>
    <a href="#" class="list-group-item list-group-item-action">
      <i class="bi-music-note-beamed"></i> Music
      <span class="badge rounded-pill bg-primary float-end">50</
span>
    </a>
    <a href="#" class="list-group-item list-group-item-action">
      <i class="bi-film"></i> Videos
      <span class="badge rounded-pill bg-primary float-end">8</
span>
    </a>
  </div>
</div>
</body>
</html>

```

## List Group with Contextual States

Like most of the other components you can also use contextual classes on the list group items to apply extra emphasis on them. Here's an example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap List Group with Contextual States</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <ul class="list-group w-50">
    <li class="list-group-item">A simple default list group item</li>
    <li class="list-group-item list-group-item-primary">A simple
primary list group item</li>
    <li class="list-group-item list-group-item-secondary">A simple
secondary list group item</li>
    <li class="list-group-item list-group-item-success">A simple
success list group item</li>
    <li class="list-group-item list-group-item-danger">A simple danger
list group item</li>
    <li class="list-group-item list-group-item-warning">A simple
warning list group item</li>
    <li class="list-group-item list-group-item-info">A simple info list
group item</li>
    <li class="list-group-item list-group-item-light">A simple light list
group item</li>
    <li class="list-group-item list-group-item-dark">A simple dark list
group item</li>
  </ul>
</div>
</body>
</html>
```

# Creating Forms with Bootstrap

HTML forms are an integral part of the web pages and applications, but creating the form layouts or styling the form controls manually one by one using CSS are often boring and tedious. Bootstrap greatly simplifies the process of styling and alignment of form controls like labels, input fields, selectboxes, textareas, buttons, etc. through predefined set of classes.

Bootstrap provides three different types of form layouts:

- Vertical Form (default form layout)
- Horizontal Form
- Inline Form

## Creating Vertical Form Layout

To create vertical form layouts simply use the predefined [margin utility classes](#) for grouping the labels, form controls, optional form text, and form validation messages.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Vertical Form Layout</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
```



```

    <form action="/examples/actions/confirmation.php"
method="post">
    <div class="mb-3">
        <label class="form-label" for="inputEmail">Email</
label>
        <input type="email" class="form-control"
id="inputEmail" placeholder="Email" required>
    </div>
    <div class="mb-3">
        <label class="form-label"
for="inputPassword">Password</label>
        <input type="password" class="form-control"
id="inputPassword" placeholder="Password" required>
    </div>
    <div class="mb-3">
        <div class="form-check">
            <input class="form-check-input" type="checkbox"
id="checkRemember">
            <label class="form-check-label"
for="checkRemember">Remember me</label>
        </div>
    </div>
    <button type="submit" class="btn btn-primary">Sign in</
button>
    </form>
</div>
</body>
</html>

```

## Creating Horizontal Form Layout

You can also create horizontal form layouts where labels and form controls are aligned side-by-side using the Bootstrap grid classes. To create a horizontal form layout add the class `.row` on form groups and use the `.col-*-*` grid classes to specify the width of your labels and controls.

```
<form>
  <div class="row mb-3">
    <label for="inputEmail" class="col-sm-2
col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-
control" id="inputEmail" placeholder="Email">
    </div>
  </div>
  <div class="row mb-3">
    <label for="inputPassword" class="col-sm-2
col-form-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-
control" id="inputPassword"
placeholder="Password">
    </div>
  </div>
  <div class="row mb-3">
    <div class="col-sm-10 offset-sm-2">
      <div class="form-check">
        <input class="form-check-
input" type="checkbox" id="checkRemember">
        <label class="form-check-
label" for="checkRemember">Remember me</label>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-10 offset-sm-2">
      <button type="submit" class="btn btn-
primary">Sign in</button>
    </div>
  </div>
```

```
</form>
```

## Creating Inline Form Layout

Sometimes you may want to display a series of form controls, and buttons in a single horizontal row to compact the layout.

```
<form>
  <div class="row align-items-center g-3">
    <div class="col-auto">
      <label class="visually-hidden"
for="inputEmail">Email</label>
      <input type="email" class="form-
control" id="inputEmail" placeholder="Email">
    </div>
    <div class="col-auto">
      <label class="visually-hidden"
for="inputPassword">Password</label>
      <input type="password" class="form-
control" id="inputPassword"
placeholder="Password">
    </div>
    <div class="col-auto">
      <div class="form-check">
        <input class="form-check-input"
type="checkbox" id="checkRemember">
        <label class="form-check-label"
for="checkRemember">Remember me</label>
      </div>
    </div>
    <div class="col-auto">
      <button type="submit" class="btn btn-
primary">Sign in</button>
    </div>
  </div>
```

`</form>`

## Height Sizing of Form Controls

You can easily change the height of your text inputs and select boxes to match the [button sizes](#).

Use the form control height sizing classes such as `.form-control-lg` and `.form-control-sm` on the text inputs to create it's larger or smaller sizes. Here's an example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Text Input Height Sizing</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="row mb-3">
    <label 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" placeholder="Large input">
    </div>
  </div>
  <div class="row mb-3">
    <label class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
```

```

        <input type="email" class="form-control"
placeholder="Default input">
    </div>
</div>
<div class="row">
    <label class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
        <input type="email" class="form-control form-control-
sm" placeholder="Small input">
    </div>
</div>
</div>
</body>
</html>

```

## Creating Custom Checkboxes

To create custom checkboxes wrap each checkbox `<input>` and their corresponding `<label>` in a `<div>` element, and apply the classes as shown in the following example:

```

<div class="form-check">
    <input type="checkbox" class="form-check-
input" name="customCheck" id="customCheck1">
    <label class="form-check-label"
for="customCheck1">Custom checkbox</label>
</div>
<div class="form-check">
    <input type="checkbox" class="form-check-
input" name="customCheck" id="customCheck2"
checked>
    <label class="form-check-label"
for="customCheck2">Another custom checkbox</label>
</div>

```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Custom Checkboxes</title>
<link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form>
    <div class="form-check">
      <input type="checkbox" class="form-
check-input" name="customCheck" id="customCheck1">
      <label class="form-check-label"
for="customCheck1">Custom checkbox</label>
    </div>
    <div class="form-check mt-2">
      <input type="checkbox" class="form-
check-input" name="customCheck" id="customCheck2"
checked>
      <label class="form-check-label"
for="customCheck2">Another custom checkbox</label>
    </div>
  </form>
</div>
</body>
```

```
</html>
```

## Creating Custom Radio Buttons

Similarly, you can create custom radio buttons using the Bootstrap like this:

```
<div class="form-check">
  <input type="radio" class="form-check-input"
name="customRadio" id="customRadio1">
  <label class="form-check-label"
for="customRadio1">Custom radio</label>
</div>
<div class="form-check">
  <input type="radio" class="form-check-input"
name="customRadio" id="customRadio2" checked>
  <label class="form-check-label"
for="customRadio2">Another custom radio</label>
</div>
```

## Inline Checkboxes and Radio Buttons

You can also place these custom checkboxes and radio buttons inline by simply adding the class `.form-check-inline` on the wrapper `.form-check` element.

```
<div class="form-check form-check-inline">
  <input type="checkbox" class="form-check-
input" name="customCheck" id="customCheck1">
  <label class="form-check-label"
for="customCheck1">Custom checkbox</label>
</div>
<div class="form-check form-check-inline">
```

```
<input type="checkbox" class="form-check-  
input" name="customCheck" id="customCheck2"  
checked>  
  <label class="form-check-label"  
for="customCheck2">Another custom checkbox</label>  
</div>
```

Similarly, you can place the radio buttons inline, as shown in the following example:

```
<div class="form-check form-check-inline">  
  <input type="radio" class="form-check-input"  
name="customRadio" id="customRadio1">  
  <label class="form-check-label"  
for="customRadio1">Custom radio</label>  
</div>  
<div class="form-check form-check-inline">  
  <input type="radio" class="form-check-input"  
name="customRadio" id="customRadio2" checked>  
  <label class="form-check-label"  
for="customRadio2">Another custom radio</label>  
</div>
```

## Disabling Custom Checkboxes and Radios

Custom checkboxes and radio buttons can also be disabled. Just add the boolean attribute `disabled` to the `<input>` element, as shown in the following example:

```
<div class="form-check">  
  <input type="checkbox" class="form-check-  
input" id="customCheck" disabled>
```



```
    <label class="form-check-label"
for="customCheck">Disabled custom checkbox</label>
</div>
<div class="form-check">
    <input type="radio" class="form-check-input"
id="customRadio" disabled>
    <label class="form-check-label"
for="customRadio">Disabled custom radio</label>
</div>
```

## Creating Toggle Switches

The switch markup is similar to custom checkbox—the only difference is—it uses the `.form-switch` class to render a toggle switch. Switches also support the `disabled` attribute.

```
<div class="form-check form-switch">
    <input class="form-check-input"
type="checkbox" id="switchDefault">
    <label class="form-check-label"
for="switchDefault">Default switch checkbox</
label>
</div>
<div class="form-check form-switch">
    <input class="form-check-input"
type="checkbox" id="switchChecked" checked>
    <label class="form-check-label"
for="switchChecked">Checked switch checkbox</
label>
</div>
<div class="form-check form-switch">
    <input class="form-check-input"
type="checkbox" id="switchDisabled" disabled>
```

```
    <label class="form-check-label"
for="switchDisabled">Disabled switch checkbox</
label>
</div>
```

## Creating Custom Select Menu

You can also customize the select dropdown menus by simply adding the class `.form-select` on the `<select>` element. However, this custom styling is limited only to the initial appearance of the `<select>` and cannot modify the `<option>`s due to browser limitations.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Custom Select Dropdown</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form>
    <select class="form-select">
      <option selected>Custom select menu</option>
      <option value="1">One</option>
      <option value="2">Two</option>
      <option value="3">Three</option>
    </select>
  </form>
</div>
```

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

You can also add the `disabled` attribute on a custom select to give it a grayed out appearance and remove pointer events, as shown in the following example:

```
<select class="form-select" disabled>
  <option selected>Custom select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
```

You can also create large and small variant of the custom selects to match the height of similarly sized [Bootstrap's text inputs](#) using the classes `.form-select-lg` and `.form-select-sm` on the `<select>` element, respectively. Let's take a look at the following example:

```
<select class="form-select form-select-lg">
  <option selected>Large custom select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
<select class="form-select mt-3">
  <option selected>Default custom select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
<select class="form-select form-select-sm mt-3">
```

```
<option selected>Small custom select menu</option>
<option value="1">One</option>
<option value="2">Two</option>
<option value="3">Three</option>
</select>
```

Bootstrap custom select also supports `multiple` and `size` attributes like normal select:

```
<select class="form-select" size="3" multiple>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
  <option value="4">Four</option>
</select>
```

## Creating Custom Range Inputs

To create custom range inputs just apply the class `.form-range` to the `<input type="range">`

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Custom Range</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
```

```

</head>
<body>
<div class="m-4">
  <form>
    <label for="customRange">Custom range</
label>
    <input type="range" class="form-range"
id="customRange">
  </form>
</div>
</body>
</html>

```

## Bootstrap Input Groups

```

<div class="row g-2">
  <div class="col-12">
    <div class="input-group">
      <span class="input-group-
text">Address</span>
      <textarea class="form-control"></
textarea>
    </div>
  </div>
  <div class="col-6">
    <div class="input-group">
      <label class="input-group-
text">Country</label>
      <select class="form-select">
        <option selected>Choose...</
option>
        <option>France</option>
        <option>Germany</option>
        <option>Hungary</option>
      </select>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
    <div class="col-6">
        <div class="input-group">
            <select class="form-select">
                <option selected>Choose...</
option>
                <option>One</option>
                <option>Two</option>
                <option>Three</option>
            </select>
            <button type="button" class="btn btn-
secondary">Submit</button>
        </div>
    </div>
</div>

```

Similarly, you can prepend or append addons to Bootstrap's custom file input, like this:

```

<div class="input-group">
    <input type="file" class="form-control">
    <button type="button" class="btn btn-
secondary">Upload</button>
</div>

```

## Checkboxes and Radio Buttons Addons

```

<div class="row">
    <div class="col-6">
        <div class="input-group">
            <span class="input-group-text">

```

```

        <input type="checkbox"
class="form-check-input mt-0">
        </span>
        <input type="text" class="form-
control">
    </div>
</div>
<div class="col-6">
    <div class="input-group">
        <span class="input-group-text">
            <input type="radio" class="form-
check-input mt-0">
        </span>
        <input type="text" class="form-
control">
    </div>
</div>
</div>

```

## Buttons Addons for Form Controls

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Input Groups with Buttons</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">

```

```

<div class="row">
  <div class="col-5">
    <div class="input-group">
      <input type="text" class="form-control"
placeholder="Search...">
      <button type="button" class="btn btn-secondary">
        <i class="bi-search"></i>
      </button>
    </div>
  </div>
  <div class="col-7">
    <div class="input-group">
      <input type="text" class="form-control" placeholder="Type
something...">
      <button type="submit" class="btn btn-primary">Submit</
button>
      <button type="reset" class="btn btn-danger">Reset</
button>
    </div>
  </div>
</div>
</body>
</html>

```

## Bootstrap Button Styles

Different classes are available in Bootstrap for styling the buttons as well as to indicate the different states or semantic. Button styles can be applied to any element. However, it is applied normally to the `<a>`, `<input>`, and `<button>` elements for the best rendering.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Button Styles</title>

```



```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary">Primary</button>
  <button type="button" class="btn btn-secondary">Secondary</
button>
  <button type="button" class="btn btn-success">Success</button>
  <button type="button" class="btn btn-danger">Danger</button>
  <button type="button" class="btn btn-warning">Warning</button>
  <button type="button" class="btn btn-info">Info</button>
  <button type="button" class="btn btn-dark">Dark</button>
  <button type="button" class="btn btn-light">Light</button>
  <button type="button" class="btn btn-link">Link</button>
</div>
</body>
</html>
```

## Bootstrap Outline Buttons

You can also create outline buttons by replacing the button modifier classes, like this:

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

```
<button type="button" class="btn btn-outline-dark">Dark</button>  
<button type="button" class="btn btn-outline-light">Light</button>
```

## Changing the Sizes of Buttons

Bootstrap gives you option further to scaling a button up or down.

To make buttons larger add an extra class `.btn-lg` to the buttons, like this:

```
<div class="m-4">  
  <button type="button" class="btn btn-primary btn-lg">Large button</button>  
  <button type="button" class="btn btn-secondary btn-lg">Large button</button>  
</div>
```

Small Button

```
<div class="m-4">  
  <button type="button" class="btn btn-primary btn-sm">Small button</button>  
  <button type="button" class="btn btn-secondary btn-sm">Small button</button>  
</div>
```

## Creating Spinner Buttons

With Bootstrap you can easily include spinner icon in a button to indicate the loading state in your application. Check out the following example to see how it works:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Spinner Buttons</title>
<link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
</head>
<body>
<div class="m-4">
    <button type="button" class="btn btn-primary"
disabled>
        <span class="spinner-border spinner-border-
sm"></span>
    </button>
    <button type="button" class="btn btn-primary"
disabled>
        <span class="spinner-border spinner-border-
sm"></span> Loading...
    </button>
    <button type="button" class="btn btn-primary"
disabled>
        <span class="spinner-grow spinner-grow-sm"></
span> Loading...
    </button>
</div>
</body>
</html>
```

# Creating Button Groups with Bootstrap

In the previous chapter you've learnt how to create different types of individual buttons and modify them with predefined classes. Bootstrap however, also allows you to group a series of buttons together in a single line through the button group component.

To create a button group just wrap a series of buttons with `.btn` class in a `<div>` element and apply the class `.btn-group` on it. You can additionally apply the class `.active` on an individual button to indicate the active state. Let's take a look at the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Button Groups</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="btn-group">
    <button type="button" class="btn btn-primary">Home</button>
    <button type="button" class="btn btn-primary active">About</
button>
    <button type="button" class="btn btn-primary">Services</
button>
  </div>
</div>
</body>
</html>
```

Similarly, you can also create button groups using outline buttons, like this:

```
<div class="btn-group">
  <button type="button" class="btn btn-outline-primary">Home</button>
  <button type="button" class="btn btn-outline-primary active">About</button>
  <button type="button" class="btn btn-outline-primary">Services</button>
</div>
```

## Mixed Styles Button Groups

You can also mix and match different button styles to create button groups like this:

```
<div class="btn-group">
  <button type="button" class="btn btn-success">
    <i class="bi-eye"></i> View
  </button>
  <button type="button" class="btn btn-warning">
    <i class="bi-pencil"></i> Edit
  </button>
  <button type="button" class="btn btn-danger">
    <i class="bi-trash"></i> Delete
  </button>
</div>
```

## Vertically Stacked Button Groups

You can also make the button groups appear vertically stacked rather than horizontally. To do this just replace the class `.btn-group` with the class `.btn-group-vertical`, like this:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Vertical Button Groups</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="btn-group-vertical">
    <a href="#" class="btn btn-primary">Home</a>
    <a href="#" class="btn btn-primary">About</a>
    <div class="btn-group">
      <a href="#" class="btn btn-primary dropdown-toggle"
data-bs-toggle="dropdown">Services</a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Web Design</
a>
        <a class="dropdown-item" href="#">Web
Development</a>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

## Creating a Basic Card

The card markup is pretty straight forward. The outer wrapper require the base class `.card`, whereas content can be placed

inside the `.card-body` element. The following example will show you how to create a card with a picture, mixed with some text content and a button.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Card</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="card" style="width: 300px;">
    
    <div class="card-body text-center">
      <h5 class="card-title">Alice Liddel</h5>
      <p class="card-text">Alice is a freelance web designer
and developer based in London. She is specialized in HTML5,
CSS3, JavaScript, Bootstrap, etc.</p>
      <a href="#" class="btn btn-primary">View Profile</a>
    </div>
  </div>
</div>
</body>
</html>
```

## Adding Navigation to Cards

You can also add [Bootstrap's nav components](#) such as tabs and pills to the card header.

To add tabs navigation to a card simply place the tabs markup inside the card header, and the tabs content inside the card body. You are also required to use an additional class `.card-header-tabs` on the `.nav` element along with the class `.nav-tabs` for proper alignment.

Let's try out the following example which creates an elegant tabbed navigation.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Card with Tab Navigation</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="card text-center">
    <div class="card-header">
      <ul class="nav nav-tabs card-header-tabs"
id="myTab">
        <li class="nav-item">
          <a href="#home" class="nav-link active" data-bs-
toggle="tab">Home</a>
        </li>
        <li class="nav-item">
          <a href="#profile" class="nav-link" data-bs-
toggle="tab">Profile</a>
```



```
        </li>
        <li class="nav-item">
            <a href="#messages" class="nav-link" data-bs-
toggle="tab">Messages</a>
        </li>
    </ul>
</div>
<div class="card-body">
    <div class="tab-content">
        <div class="tab-pane fade show active" id="home">
            <h5 class="card-title">Home tab content</h5>
            <p class="card-text">Here is some example text
to make up the tab's content. Replace it with your own text
anytime.</p>
            <a href="#" class="btn btn-primary">Go
somewhere</a>
        </div>
        <div class="tab-pane fade" id="profile">
            <h5 class="card-title">Profile tab content</h5>
            <p class="card-text">Here is some example text
to make up the tab's content. Replace it with your own text
anytime.</p>
            <a href="#" class="btn btn-primary">Go
somewhere</a>
        </div>
        <div class="tab-pane fade" id="messages">
            <h5 class="card-title">Messages tab content</
h5>
            <p class="card-text">Here is some example text
to make up the tab's content. Replace it with your own text
anytime.</p>
            <a href="#" class="btn btn-primary">Go
somewhere</a>
        </div>
    </div>
</div>
```

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

## Creating Basic Nav with Bootstrap

You can use the Bootstrap `.nav` class to create a basic navigation menu, like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Basic Nav</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav class="nav">
    <a href="#" class="nav-item nav-link active">Home</a>
    <a href="#" class="nav-item nav-link">Profile</a>
    <a href="#" class="nav-item nav-link">Messages</a>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
  </nav>
</div>
</body>
</html>
```

## Alignment of Nav Items

By default, navs are left-aligned, but you can easily align them to center or right using flexbox utilities.

The following example uses the class `.justify-content-center` to align nav items to center.

```
<nav class="nav justify-content-center">
  <a href="#" class="nav-item nav-link
active">Home</a>
  <a href="#" class="nav-item nav-
link">Profile</a>
  <a href="#" class="nav-item nav-
link">Messages</a>
  <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
</nav>
```

## Creating the Basic Tabs

Simply, add the class `.nav-tabs` to the basic nav to generate a tabbed navigation, like this:

```
<nav class="nav nav-tabs">
  <a href="#" class="nav-item nav-link
active">Home</a>
  <a href="#" class="nav-item nav-
link">Profile</a>
  <a href="#" class="nav-item nav-
link">Messages</a>
  <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
</nav>
```

You can also add icons to your tab items to make it more attractive, as shown here:

```
<nav class="nav nav-tabs">
  <a href="#" class="nav-item nav-link active">
```

```

        <i class="bi-house-door"></i> Home
    </a>
    <a href="#" class="nav-item nav-link">
        <i class="bi-person"></i> Profile
    </a>
    <a href="#" class="nav-item nav-link">
        <i class="bi-envelope"></i> Messages
    </a>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">
        <i class="bi-bar-chart"></i> Reports
    </a>
</nav>

```

## Creating the Pills Nav

Similarly, you can create pill based navigation by adding the class `.nav-pills` on the basic nav instead of class `.nav-tabs`, as shown in the following example:

```

<nav class="nav nav-pills">
    <a href="#" class="nav-item nav-link
active">Home</a>
    <a href="#" class="nav-item nav-
link">Profile</a>
    <a href="#" class="nav-item nav-
link">Messages</a>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
</nav>

```

Similarly, like nav tabs you can also add icons to your pills nav to make it more attractive:

```

<nav class="nav nav-pills">

```

```

    <a href="#" class="nav-item nav-link active">
      <i class="bi-house-door"></i> Home
    </a>
    <a href="#" class="nav-item nav-link">
      <i class="bi-person"></i> Profile
    </a>
    <a href="#" class="nav-item nav-link">
      <i class="bi-envelope"></i> Messages
    </a>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">
      <i class="bi-bar-chart"></i> Reports
    </a>
  </nav>

```

## Creating Tabs with Dropdowns

The following example will show you how to add simple dropdown menu to a tab.

```

<nav class="nav nav-tabs">
  <a href="#" class="nav-item nav-link
active">Home</a>
  <a href="#" class="nav-item nav-
link">Profile</a>
  <div class="nav-item dropdown">
    <a href="#" class="nav-link dropdown-
toggle" data-bs-toggle="dropdown">Messages</a>
    <div class="dropdown-menu">
      <a href="#" class="dropdown-
item">Inbox</a>
      <a href="#" class="dropdown-
item">Sent</a>
      <a href="#" class="dropdown-
item">Drafts</a>
    </div>
  </div>

```

```
        </div>
    </div>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
</nav>
```

## Creating Accordion Widget with Bootstrap

Accordion is a series of panels stacked on the top of each other.

Accordion menus and widgets are widely used in the web applications to manage the large amount of content and navigation lists in a small amount area. With Bootstrap collapse plugin you can either create accordions or show and hide content without writing any JavaScript code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Accordion</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
    <div class="accordion" id="myAccordion">
        <div class="accordion-item">
            <h2 class="accordion-header" id="headingOne">
```

```
        <button type="button" class="accordion-button
collapsed" data-bs-toggle="collapse" data-bs-
target="#collapseOne">1. What is HTML?</button>
```

```
    </h2>
```

```
    <div id="collapseOne" class="accordion-collapse
collapse" data-bs-parent="#myAccordion">
```

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

```
            <p>HTML stands for HyperText Markup Language.
HTML is the standard markup language for describing the
structure of web pages. <a href="https://
www.tutorialrepublic.com/html-tutorial/" target="_blank">Learn
more.</a></p>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

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

```
    <h2 class="accordion-header" id="headingTwo">
```

```
        <button type="button" class="accordion-button"
data-bs-toggle="collapse" data-bs-target="#collapseTwo">2.
What is Bootstrap?</button>
```

```
    </h2>
```

```
    <div id="collapseTwo" class="accordion-collapse
collapse show" data-bs-parent="#myAccordion">
```

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

```
            <p>Bootstrap is a sleek, intuitive, and powerful
front-end framework for faster and easier web development. It
is a collection of CSS and HTML conventions. <a href="https://
www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
target="_blank">Learn more.</a></p>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

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

```
    <h2 class="accordion-header" id="headingThree">
```

```
        <button type="button" class="accordion-button
collapsed" data-bs-toggle="collapse" data-bs-
target="#collapseThree">3. What is CSS?</button>
```

```
    </h2>
```

```
    <div id="collapseThree" class="accordion-collapse
collapse" data-bs-parent="#myAccordion">
```

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

```
            <p>CSS stands for Cascading Style Sheet. CSS
allows you to specify various style properties for a given HTML
element such as colors, backgrounds, fonts etc. <a
href="https://www.tutorialrepublic.com/css-tutorial/"
target="_blank">Learn more.</a></p>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## Bootstrap Accordion with Independent Panels

You can also make accordion panels stay open when another panel is opened by simply omitting the `data-bs-parent` attribute on each `.accordion-collapse` element.

```
<div class="accordion" id="myAccordion">
```

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

```
        <h2 class="accordion-header"
```

```
id="headingOne">
```

```
            <button type="button"
```

```
class="accordion-button collapsed" data-bs-
```

```
toggle="collapse" data-bs-target="#collapseOne">1.
```

```
What is HTML?</button>
```



```

        </h2>
        <div id="collapseOne" class="accordion-
collapse collapse">
            <div class="card-body">
                <p>HTML stands for HyperText
Markup Language. HTML is the standard markup
language for describing the structure of web
pages. <a href="https://www.tutorialrepublic.com/
html-tutorial/" target="_blank">Learn more.</a></
p>
            </div>
        </div>
    </div>
    <div class="accordion-item">
        <h2 class="accordion-header"
id="headingTwo">
            <button type="button"
class="accordion-button" data-bs-toggle="collapse"
data-bs-target="#collapseTwo">2. What is
Bootstrap?</button>
        </h2>
        <div id="collapseTwo" class="accordion-
collapse collapse show">
            <div class="card-body">
                <p>Bootstrap is a sleek,
intuitive, and powerful front-end framework for
faster and easier web development. It is a
collection of CSS and HTML conventions. <a
href="https://www.tutorialrepublic.com/twitter-
bootstrap-tutorial/" target="_blank">Learn more.</
a></p>
            </div>
        </div>
    </div>

```

```

    <div class="accordion-item">
      <h2 class="accordion-header"
id="headingThree">
        <button type="button"
class="accordion-button collapsed" data-bs-
toggle="collapse" data-bs-
target="#collapseThree">3. What is CSS?</button>
      </h2>
      <div id="collapseThree" class="accordion-
collapse collapse">
        <div class="card-body">
          <p>CSS stands for Cascading Style
Sheet. CSS allows you to specify various style
properties for a given HTML element such as
colors, backgrounds, fonts etc. <a href="https://
www.tutorialrepublic.com/css-tutorial/"
target="_blank">Learn more.</a></p>
        </div>
      </div>
    </div>
  </div>

```

## Creating Breadcrumbs with Bootstrap

A breadcrumb is a navigation scheme that indicates current page's location to the user within a website or application. Breadcrumb navigation can greatly enhance the accessibility of a website having a large number of pages or complex navigational hierarchy.

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Creating Breadcrumb in Bootstrap</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/
js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav>
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">Home</a></
li>
      <li class="breadcrumb-item"><a href="#">Products</
a></li>
      <li class="breadcrumb-item active">Accessories</li>
    </ol>
  </nav>
</div>
</body>
</html>

```

## Creating Pagination with Bootstrap

Pagination is the process of organizing content by dividing it into separate pages.

Pagination is used in some or other form quite often in almost every web application, for instance it is used by search engines for displaying a limited number of results on search results pages, or showing a limited number of posts for every page on a blog or forum.

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Pagination</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav>
    <ul class="pagination">
      <li class="page-item"><a href="#" class="page-
link">Previous</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">4</a></li>
      <li class="page-item"><a href="#" class="page-link">5</a></li>
      <li class="page-item"><a href="#" class="page-link">Next</
a></li>
    </ul>
  </nav>
</div>
</body>
</html>

```

## Creating Progress Bar with Bootstrap

Progress bars can be used for showing the progress of a task or action to the users.

The following example will show you how to create a simple progress bar with vertical gradient.

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Progress Bar</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
<style>
  /* Adding space at the bottom of progress bar */
  .progress{
    margin-bottom: 1rem;
  }
</style>
</head>
<body>
<div class="m-4">
  <div class="progress">
    <div class="progress-bar" style="width: 25%"></div>
  </div>
  <div class="progress">
    <div class="progress-bar" style="width: 50%"></div>
  </div>
  <div class="progress">
    <div class="progress-bar" style="width: 75%"></div>
  </div>
  <div class="progress">
    <div class="progress-bar" style="width: 100%"></div>
  </div>
</div>
</body>
</html>

```

## Creating Colored Spinner

You can use the [text color utility classes](#) to customize the color of spinners.

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Colored Spinners</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="spinner-border text-primary">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-secondary">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-success">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-danger">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-warning">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-info">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-dark">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-border text-light">
    <span class="visually-hidden">Loading...</span>
  </div>
</div>
</body>
</html>
```

Similarly, like the border spinners you can also customize the colors of growing spinners using the Bootstrap's [text color utility classes](#), as shown in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title>Bootstrap Colored Growing Spinners</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="spinner-grow text-primary">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-secondary">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-success">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-danger">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-warning">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-info">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-light">
    <span class="visually-hidden">Loading...</span>
  </div>
  <div class="spinner-grow text-dark">
    <span class="visually-hidden">Loading...</span>
  </div>
</div>
</body>
</html>
```