



# Bootstrap 5 Material

## 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.

## What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

## Bootstrap 5 Example

```
<div class="container-fluid p-5 bg-primary text-white text-center">  
  <h1>My First Bootstrap Page</h1>  
  <p>Resize this responsive page to see the effect!</p>  
</div>
```

```
<div class="container mt-5">  
  <div class="row">  
    <div class="col-sm-4">  
      <h3>Column 1</h3>  
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>  
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>  
    </div>  
    <div class="col-sm-4">  
      <h3>Column 2</h3>  
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>  
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>  
    </div>  
    <div class="col-sm-4">  
      <h3>Column 3</h3>  
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>  
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>  
    </div>  
  </div>  
</div>
```

</div>  
</div>

## Bootstrap Versions

Bootstrap 5 (released 2021) is the newest version of Bootstrap (released 2013); with new components, faster stylesheet and more responsiveness. Bootstrap 5 supports the latest, stable releases of all major browsers and platforms. However, Internet Explorer 11 and down is not supported.

### 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, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap 5 is compatible with all modern browsers (Chrome, Firefox, Edge, Safari, and Opera). **Note** that if you need support for IE11 and down, you must use either BS4 or BS3.

### Where to Get Bootstrap 5?

There are two ways to start using Bootstrap 5 on your own web site.

You can:

- Include Bootstrap 5 from a CDN
- Download Bootstrap 5 from [getbootstrap.com](https://getbootstrap.com)

### Bootstrap 5 CDN

If you don't want to download and host Bootstrap 5 yourself, you can include it from a CDN.

jsDelivr provides CDN support for Bootstrap's CSS and JavaScript:

**Max CDN:**

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></
script>
```

## 1. Add the HTML5 doctype

Bootstrap 5 uses HTML elements and CSS properties that require the HTML5 doctype.

Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct title and character set:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Bootstrap 5 Example</title>
    <meta charset="utf-8">
  </head>
</html>
```

## 2. Bootstrap 5 is mobile-first

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

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

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

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

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

### 3. Containers

Bootstrap 5 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

### Two Basic Bootstrap 5 Pages

The following example shows the code for a basic Bootstrap 5 page (with a responsive fixed width container):

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></
script>
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container class.</p>
  <p>The .container class provides a responsive fixed width container.</p>
</div>
```

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

## Container Fluid Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></
script>
</head>
<body>
<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container-fluid class.</p>
  <p>The .container-fluid class provides a full width container, spanning the entire width
of the viewport.</p>
</div>
</body>
</html>
```

## Fixed Container

Use the `.container` class to create a responsive, fixed-width container.

Note that its width (**max-width**) will change on different screen sizes:

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

Open the example below and resize the browser window to see that the container width will change at different breakpoints:

```
<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

## Container Padding

By default, containers have left and right padding, with no top or bottom padding. Therefore, we often use **spacing utilities**, such as extra padding and margins to make them look even better. For example, **.pt-5** means "add a large **top padding**":

### Example

```
<div class="container pt-5"></div>
```

## Container Border and Color

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

### Example

```
<div class="container p-5 my-5 border"></div>
<div class="container p-5 my-5 bg-dark text-white"></div>
<div class="container p-5 my-5 bg-primary text-white"></div>
```

You will learn much more about colors and border utilities in a later chapter.

## Responsive Containers

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

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

Class	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px	XXL ≥1400px
<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

### Example

```
<div class="container-sm">.container-sm</div>
<div class="container-md">.container-md</div>
<div class="container-lg">.container-lg</div>
<div class="container-xl">.container-xl</div>
<div class="container-xxl">.container-xxl</div>
```

## Bootstrap 5 Grid System

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



## Grid Classes

The Bootstrap 5 grid system has six classes:

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)
- **.col-xxl-** (xxlarge devices - screen width equal to or greater than 1400px)

## Three Equal Columns

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

### Example

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

## Responsive Columns

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

### Example

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

### Two Unequal Responsive Columns

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

### Example

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

### Bootstrap 5 Default Settings

Bootstrap 5 uses a default **font-size** of 1rem (16px by default), and its **line-height** is 1.5. In addition, all **<p>** elements have **margin-top: 0** and **margin-bottom: 1rem** (16px by default).

**<h1>** - **<h6>**

Bootstrap 5 styles HTML headings (**<h1>** to **<h6>**) with a bolder font-weight and a responsive font-size.

### Example

You can also use **.h1** to **.h6** classes on other elements to make them behave as headings if you want:

## Example

```
<p class="h1">h1 Bootstrap heading</p>
<p class="h2">h2 Bootstrap heading</p>
<p class="h3">h3 Bootstrap heading</p>
<p class="h4">h4 Bootstrap heading</p>
<p class="h5">h5 Bootstrap heading</p>
<p class="h6">h6 Bootstrap heading</p>
```

## Display Headings

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are six classes to choose from: `.display-1` to `.display-6`:

### `<small>`

In Bootstrap 5 the HTML `<small>` element (and the `.small` class) is used to create a smaller, secondary text in any heading:

### `<mark>`

Bootstrap 5 will style `<mark>` and `.mark` with a yellow background color and some padding:

## More Typography Classes

The Bootstrap 5 classes below can be added to style HTML elements further:

Class	Description
<code>.lead</code>	Makes a paragraph stand out
<code>.text-start</code>	Indicates left-aligned text
<code>.text-break</code>	Prevents long text from breaking layout
<code>.text-center</code>	Indicates center-aligned text

<code>.text-decoration-none</code>	Removes the underline from a link
<code>.text-end</code>	Indicates right-aligned text
<code>.text-nowrap</code>	Indicates no wrap text
<code>.text-lowercase</code>	Indicates lowercased text
<code>.text-uppercase</code>	Indicates uppercased text
<code>.text-capitalize</code>	Indicates capitalized text
<code>.list-inline</code>	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code>&lt;li&gt;</code> elements)

## Bootstrap 5 Colors

Bootstrap 5 has some contextual classes that can be used to provide "meaning through colors". The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body` (default body color/often black) and `.text-light`:

### Example

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

Dark grey text.

Body text.

Light grey text.

You can also add 50% opacity for black or white text with the `.text-black-50` or `.text-white-50` classes:

### Example

Black text with 50% opacity on white background

White text with 50% opacity on black background

## Background Colors

The classes for background colors are: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`.

The `.bg-color` classes above does not work well with text, or atleast then you have to specify a proper `.text-color` class to get the right text color for each background.

### Text Bg color classes

However, you can use the `.text-bg-color` classes and Bootstrap will automatically handle the appropriate text color for each background color:

### Example

```
<div class="container mt-3">
  <h2>Background Color with Contrasting Text Color</h2>
  <p class="text-bg-primary">This text is important.</p>
  <p class="text-bg-success">This text indicates success.</p>
  <p class="text-bg-info">This text represents some information.</p>
  <p class="text-bg-warning">This text represents a warning.</p>
  <p class="text-bg-danger">This text represents danger.</p>
```

```
<p class="text-bg-secondary">Secondary background color.</p>
<p class="text-bg-dark">Dark grey background color.</p>
<p class="text-bg-light">Light grey background color.</p>
</div>
```

## Basic Table

A basic Bootstrap 5 table has a light padding and horizontal dividers.

The `.table` class adds basic styling to a table:

```
<table class="table">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
```

```
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

### Striped Rows:

The `.table-striped` class adds zebra-stripes to a table:

```
<table class="table table-striped">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

## Bordered Table

The `.table-bordered` class adds borders on all sides of the table and cells:

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
```



```
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

## Hover Rows

The **.table-hover** class adds a hover effect (grey background color) on table rows:

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
      <td>Mary</td>
      <td>Moe</td>
      <td>mary@example.com</td>
    </tr>
    <tr>
      <td>July</td>
      <td>Dooley</td>
      <td>july@example.com</td>
    </tr>
  </tbody>
```

```
</table>
```

## Black/Dark Table

The `.table-dark` class adds a black background to the table:

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

## Dark Striped Table

Combine `.table-dark` and `.table-striped` to create a dark, striped table:

```
<table class="table table-dark table-striped">  
  <thead>  
    <tr>  
      <th>Firstname</th>  
      <th>Lastname</th>  
      <th>Email</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>  
      <td>John</td>  
      <td>Doe</td>  
      <td>john@example.com</td>  
    </tr>
```

```

<tr>
  <td>Mary</td>
  <td>Moe</td>
  <td>mary@example.com</td>
</tr>
<tr>
  <td>July</td>
  <td>Dooley</td>
  <td>july@example.com</td>
</tr>
</tbody>
</table>

```

## Hoverable Dark Table

The `.table-hover` class adds a hover effect (grey background color) on table rows:

Example

```

<table class="table table-dark table-hover">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>

```

```
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>john@example.com</td>
</tr>
<tr>
  <td>Mary</td>
  <td>Moe</td>
  <td>mary@example.com</td>
</tr>
<tr>
  <td>July</td>
  <td>Dooley</td>
  <td>july@example.com</td>
</tr>
</tbody>
</table>
```

## Borderless Table

The `.table-borderless` class removes borders from the table:

Example

```
<table class="table table-borderless">
</table>
```

## Contextual Classes

[Contextual classes](#) can be used to color the whole table (`<table>`), the table rows (`<tr>`) or table cells (`<td>`).

```
<table class="table">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Default</td>
      <td>Defaultson</td>
      <td>def@somemail.com</td>
    </tr>
    <tr class="table-primary">
      <td>Primary</td>
      <td>Joe</td>
      <td>joe@example.com</td>
    </tr>
    <tr class="table-success">
      <td>Success</td>
```

```
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr class="table-danger">
  <td>Danger</td>
  <td>Moe</td>
  <td>mary@example.com</td>
</tr>
<tr class="table-info">
  <td>Info</td>
  <td>Dooley</td>
  <td>july@example.com</td>
</tr>
<tr class="table-warning">
  <td>Warning</td>
  <td>Refs</td>
  <td>bo@example.com</td>
</tr>
<tr class="table-active">
  <td>Active</td>
  <td>Activeson</td>
  <td>act@example.com</td>
</tr>
<tr class="table-secondary">
  <td>Secondary</td>
  <td>Secondson</td>
```

```

        <td>sec@example.com</td>
    </tr>
    <tr class="table-light">
        <td>Light</td>
        <td>Angie</td>
        <td>angie@example.com</td>
    </tr>
    <tr class="table-dark">
        <td>Dark</td>
        <td>Bo</td>
        <td>bo@example.com</td>
    </tr>
</tbody>
</table>

```

### Example

Default	Defaultson	<a href="mailto:def@somemail.com">def@somemail.com</a>
Primary	Joe	<a href="mailto:joe@example.com">joe@example.com</a>
Success	Doe	<a href="mailto:john@example.com">john@example.com</a>
Danger	Moe	<a href="mailto:mary@example.com">mary@example.com</a>
Info	Dooley	<a href="mailto:july@example.com">july@example.com</a>
Warning	Refs	<a href="mailto:bo@example.com">bo@example.com</a>
Active	Activeson	<a href="mailto:act@example.com">act@example.com</a>
Secondary	Secondson	<a href="mailto:sec@example.com">sec@example.com</a>
Light	Angie	<a href="mailto:angie@example.com">angie@example.com</a>
		<a href="mailto:bo@example.com">bo@example.com</a>

The contextual classes that can be used are:

Class	Description
-------	-------------



<code>.table-primary</code>	Blue: Indicates an important action
<code>.table-success</code>	Green: Indicates a successful or positive action
<code>.table-danger</code>	Red: Indicates a dangerous or potentially negative action
<code>.table-info</code>	Light blue: Indicates a neutral informative change or action
<code>.table-warning</code>	Orange: Indicates a warning that might need attention
<code>.table-active</code>	Grey: Applies the hover color to the table row or table cell
<code>.table-secondary</code>	Grey: Indicates a slightly less important action
<code>.table-light</code>	Light grey table or table row background
<code>.table-dark</code>	Dark grey table or table row background

### Table Head Colors

You can also use any of the contextual classes to only add a background color to the table header:

Example

```
<table class="table">
  <thead class="table-dark">
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
```

```

        <td>Doe</td>
        <td>john@example.com</td>
    </tr>
    <tr>
        <td>Mary</td>
        <td>Moe</td>
        <td>mary@example.com</td>
    </tr>
    <tr>
        <td>July</td>
        <td>Dooley</td>
        <td>july@example.com</td>
    </tr>
</tbody>
</table>
<table class="table">
    <thead class="table-success">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>John</td>

```

```
<td>Doe</td>
<td>john@example.com</td>
</tr>
<tr>
<td>Mary</td>
<td>Moe</td>
<td>mary@example.com</td>
</tr>
<tr>
<td>July</td>
<td>Dooley</td>
<td>july@example.com</td>
</tr>
</tbody>
</table>
```

## Small table

The **.table-sm** class makes the table smaller by cutting cell padding in half:

Example

```
<table class="table table-bordered table-sm">
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Email</th>
```

```
</tr>
</thead>
<tbody>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  </tr>
  <tr>
    <td>Mary</td>
    <td>Moe</td>
    <td>mary@example.com</td>
  </tr>
  <tr>
    <td>July</td>
    <td>Dooley</td>
    <td>july@example.com</td>
  </tr>
</tbody>
</table>
```

## Responsive Tables

The `.table-responsive` class adds a scrollbar to the table when needed (when it is too big horizontally):

Example

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```

You can also decide when the table should get a scrollbar, depending on the screen width:

Class	Screen width
.table-responsive-sm	< 576px
.table-responsive-md	< 768px
.table-responsive-lg	< 992px
.table-responsive-xl	< 1200px
.table-responsive-xxl	< 1400px

Example

```
<div class="table-responsive-sm">
  <table class="table">
    ...
  </table>
</div>
```

## Bootstrap 5 Images

### Rounded Corners

The **.rounded** class adds rounded corners to an image:

Example

```

```

### Circle

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

### Example

```

```

### Thumbnail

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

### Example

```

```

### Aligning Images

Float an image to the left with the `.float-start` class or to the right with `.float-end`:

### Example

```

```

```

```

### Centered Image

Center an image by adding the utility classes `.mx-auto` (margin:auto) and `.d-block` (display:block) to the image:

### Example

```

```

### Responsive Images

Images come in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen. Create responsive images by adding an `.img-fluid` class to the `<img>` tag. The image will then scale nicely to the parent element.

### Example

```

```

## Bootstrap 5 Jumbotron

A jumbotron was introduced in Bootstrap 3 as a big padded box for calling extra attention to some special content or information. Jumbotrons are no longer supported in Bootstrap 5. **However**, you can use a `<div>` element and add special helper classes together with a color class to achieve the same effect:

### Example

```
<div class="mt-4 p-5 bg-primary text-white rounded">
  <h1>Jumbotron Example</h1>
  <p>Lorem ipsum...</p>
</div>
```

## Bootstrap 5 Alerts

Bootstrap 5 provides an easy way to create predefined alert messages:

Alerts are created with the `.alert` class, followed by one of the contextual classes `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`, `.alert-primary`, `.alert-secondary`, `.alert-light` or `.alert-dark`:

### Example

```
<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>
```

## Alert Links

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

### Example

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

```
</div>
```

## Closing Alerts

**Success!** This alert box could indicate a successful or positive action. To close the alert message, add a `.alert-dismissible` class to the alert container. Then add `class="btn-close"` and `data-bs-dismiss="alert"` to a link or a button element (when you click on this the alert box will disappear).

### Example

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
  <strong>Success!</strong> This alert box could indicate a successful or positive action.
</div>
```

## Animated Alerts

The `.fade` and `.show` classes adds a fading effect when closing the alert message:

### Example

```
<div class="alert alert-danger alert-dismissible fade show">
```

## Bootstrap 5 Buttons

Bootstrap 5 provides different styles of buttons

### Example

```
<button type="button" class="btn">Basic</button>
<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-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```



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

## Button Outline

Bootstrap 5 also provides eight outline/bordered buttons.

Move the mouse over them to see an additional "hover" effect

Example

```
<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-info">Info</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
<button type="button" class="btn btn-outline-light text-dark">Light</button>
```

## Button Sizes

Use the `.btn-lg` class for large buttons or `.btn-sm` class for small buttons:

```
<button type="button" class="btn btn-primary btn-lg">Large</button>
<button type="button" class="btn btn-primary">Default</button>
<button type="button" class="btn btn-primary btn-sm">Small</button>
```

## Block Level Buttons

To create a block level button that spans the entire width of the parent element, use the `.d-grid` "helper" class on the parent element

Example

```
<div class="d-grid gap-3">
  <button type="button" class="btn btn-primary btn-block">Full-Width Button</button>
  <button type="button" class="btn btn-primary btn-block">Full-Width Button</button>
</div>
```

If you have many block-level buttons, you can control the space between them with the `.gap-*` class

## Active/Disabled Buttons

A button can be set to an active (appear pressed) or a disabled (unclickable) state:

The class `.active` makes a button appear pressed, and the `disabled` attribute makes a button unclickable. Note that `<a>` elements do not support the disabled attribute and must therefore use the `.disabled` class to make it visually appear disabled.

Example

```
<button type="button" class="btn btn-primary active">Active Primary</button>
<button type="button" class="btn btn-primary disabled">Disabled Primary</button>
<a href="#" class="btn btn-primary disabled">Disabled Link</a>
```

## Bootstrap 5 Button Groups

Bootstrap 5 allows you to group a series of buttons together (on a single line) in a button group

Use a `<div>` element with class `.btn-group` to create a button group

Example

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
```

**Tip:** Instead of applying button sizes to every button in a group, use class `.btn-group-lg` for a large button group or the `.btn-group-sm` for a small button group

Bootstrap 5 also supports vertical button groups:

Use the class `.btn-group-vertical` to create a vertical button group

## Bootstrap 5 Badges

Badges are used to add additional information to any content

Use the `.badge` class together with a contextual class (like `.bg-secondary`) within `<span>` elements to create rectangular badges. Note that badges scale to match the size of the parent element (if any)

Example

```
<h1>Example heading <span class="badge bg-secondary">New</span></h1>
<h2>Example heading <span class="badge bg-secondary">New</span></h2>
<h3>Example heading <span class="badge bg-secondary">New</span></h3>
<h4>Example heading <span class="badge bg-secondary">New</span></h4>
<h5>Example heading <span class="badge bg-secondary">New</span></h5>
<h6>Example heading <span class="badge bg-secondary">New</span></h6>
```

## Contextual Badges

Use any of the contextual classes (**.bg-\***) to change the color of a badge

```
<span class="badge bg-success">Success</span>
```

## Pill Badges

Use the **.rounded-pill** class to make the badges more round:

```
<span class="badge rounded-pill bg-primary">Primary</span>
```

## Badge inside an Element

An example of using a badge inside a button

```
<button type="button" class="btn btn-primary">
  Messages <span class="badge bg-danger">4</span>
</button>
```

## Bootstrap 5 Progress Bars

A progress bar can be used to show how far a user is in a process.

To create a default progress bar, add a **.progress** class to a container element and add the **.progress-bar** class to its child element. Use the CSS **width** property to set the width of the progress bar

Example

```
<div class="progress">
  <div class="progress-bar" style="width:70%"></div>
</div>
```

## Progress Bar Height

The height of the progress bar is **1rem** (usually **16px**) by default. Use the CSS **height** property to change it. Note that you must set the same height for the progress container and the progress bar

Example

```
<div class="progress" style="height:20px">
  <div class="progress-bar" style="width:40%;height:20px"></div>
</div>
```

## Progress Bar Labels

Add text inside the progress bar to show the visible percentage

```
<div class="progress">
  <div class="progress-bar" style="width:70%">70%</div>
</div>
```

## Colored Progress Bars

By default, the progress bar is blue (primary). Use any of the contextual background classes to change its color

```
<!-- Green -->
<div class="progress">
  <div class="progress-bar bg-success" style="width:20%"></div>
</div>
```

## Striped Progress Bars

Use the **.progress-bar-striped** class to add stripes to the progress bars

```
<div class="progress">
  <div class="progress-bar progress-bar-striped" style="width:40%"></div>
</div>
```

## Animated Progress Bar

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

```
<div class="progress-bar progress-bar-striped progress-bar-animated" style="width:40%"></div>
```

## Multiple Progress Bars

Progress bars can also be stacked

```
<div class="progress">  
  <div class="progress-bar bg-success" style="width:40%">  
    Free Space  
  </div>  
  <div class="progress-bar bg-warning" style="width:10%">  
    Warning  
  </div>  
  <div class="progress-bar bg-danger" style="width:20%">  
    Danger  
  </div>  
</div>
```

## Bootstrap 5 Spinners

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

Example

```
<div class="spinner-border"></div>
```

## Colored Spinners

Use any **text color utilities** to add a color to the spinner

```
<div class="spinner-border text-success"></div>
```

## Growing Spinners

Use the `.spinner-grow` class if you want the spinner/loader to grow instead of "spin"

```
<div class="spinner-grow text-danger"></div>
```

## Spinner Size

Use `.spinner-border-sm` or `.spinner-grow-sm` to create a smaller spinner:

```
<div class="spinner-border spinner-border-sm"></div>  
<div class="spinner-grow spinner-grow-sm"></div>
```

## Spinner Buttons

You can also add spinners to a button, with or without text:

```
<button class="btn btn-primary">  
  <span class="spinner-border spinner-border-sm"></span>  
  Loading..  
</button>
```

## Bootstrap 5 Pagination

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

To create a basic pagination, add the `.pagination` class to an `<ul>` element. Then add the `.page-item` to each `<li>` element and a `.page-link` class to each link inside `<li>`:

Example

```
<ul class="pagination">  
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
```

```

<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>

```

## Active State

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

```

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

```

## Disabled State

The `.disabled` class is used for un-clickable links:

```

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

```

## Pagination Sizing

Pagination blocks can also be sized to a larger or a smaller size

Add class `.pagination-lg` for larger blocks or `.pagination-sm` for smaller blocks

```

<ul class="pagination pagination-lg"> </ul>

```

## Pagination Alignment

Use utility classes to change the alignment of the pagination

```

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

```

```

<!-- Center-aligned -->

```

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

<!-- Right-aligned -->

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

## Breadcrumbs

Another form for pagination, is breadcrumbs:

The `.breadcrumb` and `.breadcrumb-item` classes indicates the current page's location within a navigational

Example

```
<ul class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">Photos</a></li>
  <li class="breadcrumb-item"><a href="#">Summer 2017</a></li>
  <li class="breadcrumb-item"><a href="#">Italy</a></li>
  <li class="breadcrumb-item active">Rome</li>
</ul>
```

## Bootstrap 5 List Groups

The most basic list group is an unordered list with list items:

To create a basic list group, use an `<ul>` element with class `.list-group`, and `<li>` elements with class `.list-group-item`:

Example

```
<ul class="list-group">
  <li class="list-group-item">First item</li>
  <li class="list-group-item">Second item</li>
  <li class="list-group-item">Third item</li>
</ul>
```



## Active State

Use the `.active` class to highlight the current item:

```
<ul class="list-group">
  <li class="list-group-item active">Active item</li>
</ul>
```

## List Group With Linked Items

To create a list group with linked items, use `<div>` instead of `<ul>` and `<a>` instead of `<li>`. Optionally, add the `.list-group-item-action` class if you want a grey background color on hover:

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action">First item</a>
  <a href="#" class="list-group-item list-group-item-action">Second item</a>
  <a href="#" class="list-group-item list-group-item-action">Third item</a>
</div>
```

## Disabled Item

The `.disabled` class adds a lighter text color to the disabled item. And when used on links, it will remove the hover effect:

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">Disabled item</a>
  <a href="#" class="list-group-item disabled">Disabled item</a>
  <a href="#" class="list-group-item">Third item</a>
</div>
```

## Flush / Remove Borders

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

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

## Numbered List Groups

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

```
<ol class="list-group list-group-numbered">
  <li class="list-group-item">First item</li>
  <li class="list-group-item">Second item</li>
  <li class="list-group-item">Third item</li>
</ol>
```

## Horizontal List Groups

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

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

## Contextual Classes

Contextual classes can be used to add color to the list items

The classes for coloring list-items are: `.list-group-item-success`, `list-group-item-secondary`, `list-group-item-info`, `list-group-item-warning`, `.list-group-item-danger`, `.list-group-item-primary`, `list-group-item-dark` and `list-group-item-light`:

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Success item</li>
</ul>
```

## Link items with Contextual Classes

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action">Action item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-success">Success item</a>
</div>
```

## List Group with Badges

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

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

## Bootstrap 5 Cards

A card in Bootstrap 5 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.

### Basic Card

A basic card is created with the `.card` class, and content inside the card has a `.card-body` class:

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

### Header and Footer

The `.card-header` class adds a heading to the card and the `.card-footer` class adds a footer to the card:

```
<div class="card">
  <div class="card-header">Header</div>
  <div class="card-body">Content</div>
  <div class="card-footer">Footer</div>
</div>
```

## Contextual Cards

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

## Titles, text, and links

Use `.card-title` to add card titles to any heading element. The `.card-text` class is used to remove bottom margins for a `<p>` element if it is the last child (or the only one) inside `.card-body`. The `.card-link` class adds a blue color to any link, and a hover effect.

```
<div class="card">
  <div class="card-body">
    <h4 class="card-title">Card title</h4>
    <p class="card-text">Some example text. Some example text.</p>
    <a href="#" class="card-link">Card link</a>
    <a href="#" class="card-link">Another link</a>
  </div>
</div>
```

## Card Images

Add `.card-img-top` or `.card-img-bottom` to an `<img>` to place the image at the top or at the bottom inside the card. Note that we have added the image outside of the `.card-body` to span the entire width:

```
<div class="card" style="width:400px">
  
  <div class="card-body">
    <h4 class="card-title">Bhargavaprasad Dunga</h4>
    <p class="card-text">Some example text.</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>
```

## Card Image Overlays

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

```
<div class="card" style="width:500px">
  
  <div class="card-img-overlay">
    <h4 class="card-title">John Doe</h4>
    <p class="card-text">Some example text.</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>
```

## Bootstrap 5 Dropdowns

### Basic Dropdown

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

Example

```
<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-bs-
toggle="dropdown">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Link 1</a></li>
    <li><a class="dropdown-item" href="#">Link 2</a></li>
    <li><a class="dropdown-item" href="#">Link 3</a></li>
  </ul>
</div>
```

## Example Explained

The `.dropdown` class indicates a dropdown menu.

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

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

## Dropdown Divider

The `.dropdown-divider` class is used to separate links inside the dropdown menu with a thin horizontal border:

```
<li><hr class="dropdown-divider"></hr></li>
```

## Dropdown Header

The `.dropdown-header` class is used to add headers inside the dropdown menu:

```
<li><h5 class="dropdown-header">Dropdown header 1</h5></li>
```

## Disable and Active items

Highlight a specific dropdown item with the `.active` class (adds a blue background color).

To disable an item in the dropdown menu, use the `.disabled` class (gets a light-grey text color and a "no-parking-sign" icon on hover):

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

## Dropdown Position

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

```
<div class="dropdown dropend">
```

```
<div class="dropdown dropstart">
```

## Dropdown Menu Right

To right-align the dropdown menu, add the `.dropdown-menu-end` class to the element with `.dropdown-menu`:

```
<div class="dropdown-menu dropdown-menu-end">
```

## Dropup

If you want the dropdown menu to expand upwards instead of downwards, change the `<div>` element with `class="dropdown"` to `"dropup"`:

```
<div class="dropup">
```

## Dropdown Text

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

```
<ul class="dropdown-menu">
  <li><a class="dropdown-item" href="#">Link 1</a></li>
  <li><a class="dropdown-item" href="#">Link 2</a></li>
  <li><a class="dropdown-item" href="#">Link 3</a></li>
  <li><a class="dropdown-item-text" href="#">Text Link</a></li>
  <li><span class="dropdown-item-text">Just Text</span></li>
</ul>
```

## Grouped Buttons with a Dropdown

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <div class="btn-group">
    <button type="button" class="btn btn-primary dropdown-toggle" data-bs-
toggle="dropdown">Sony</button>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="#">Tablet</a></li>
      <li><a class="dropdown-item" href="#">Smartphone</a></li>
    </ul>
  </div>
</div>
```

## Bootstrap 5 Collapse

### Basic Collapsible

Collapsibles are useful when you want to hide and show large amount of content:

Example

```
<button data-bs-toggle="collapse" data-bs-target="#demo">Collapsible</button>

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

### Example Explained

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

To control (show/hide) the collapsible content, add the `data-bs-toggle="collapse"` attribute to an `<a>` or a `<button>` element.



Then add the `data-bs-target="#id"` attribute to connect the button with the collapsible content (`<div id="demo">`).

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

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

## Accordion

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

```
<div id="accordion">

  <div class="card">
    <div class="card-header">
      <a class="btn" data-bs-toggle="collapse" href="#collapseOne">
        Collapsible Group Item #1
      </a>
    </div>
    <div id="collapseOne" class="collapse show" data-bs-parent="#accordion">
      <div class="card-body">
        Lorem ipsum..
      </div>
    </div>
  </div>

  <div class="card">
    <div class="card-header">
      <a class="collapsed btn" data-bs-toggle="collapse" href="#collapseTwo">
        Collapsible Group Item #2
      </a>
    </div>
    <div id="collapseTwo" class="collapse" data-bs-parent="#accordion">
      <div class="card-body">
        Lorem ipsum..
      </div>
    </div>
  </div>
</div>
```

```

</div>
</div>

<div class="card">
  <div class="card-header">
    <a class="collapsed btn" data-bs-toggle="collapse" href="#collapseThree">
      Collapsible Group Item #3
    </a>
  </div>
  <div id="collapseThree" class="collapse" data-bs-parent="#accordion">
    <div class="card-body">
      Lorem ipsum..
    </div>
  </div>
</div>
</div>
</div>

```

## Bootstrap 5 Navs

If you want to create a simple horizontal menu, add the `.nav` class to a `<ul>` element, followed by `.nav-item` for each `<li>` and add the `.nav-link` class to their links:

```

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

```

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

## Aligned Nav

Add the `.justify-content-center` class to center the nav, and the `.justify-content-end` class to right-align the nav.

```
<!-- Centered nav -->
<ul class="nav justify-content-center">

<!-- Right-aligned nav -->
<ul class="nav justify-content-end">
```

## Vertical Nav

Add the `.flex-column` class to create a vertical nav:

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

## Tabs

Turn the nav menu into navigation tabs with the `.nav-tabs` class. Add the `.active` class to the active/current link. If you want the tabs to be toggleable, see the last example on this page.

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

## Pills

Turn the nav menu into navigation pills with the `.nav-pills` class. If you want the pills to be toggleable, see the last example on this page.

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

## Justified Tabs/pills

Justify the tabs/pills with the `.nav-justified` class (equal width):

```
<ul class="nav nav-pills nav-justified">..</ul>
<ul class="nav nav-tabs nav-justified">..</ul>
```

## Tabs with Dropdown

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-bs-
toggle="dropdown" href="#">Dropdown</a>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="#">Link 1</a></li>
      <li><a class="dropdown-item" href="#">Link 2</a></li>
      <li><a class="dropdown-item" href="#">Link 3</a></li>
    </ul>
  </li>
</ul>
```

## Toggleable / Dynamic Tabs

To make the tabs toggleable, add the `data-toggle="tab"` attribute to each link. Then add a `.tab-pane` class with a unique ID for every tab and wrap them inside a `<div>` element with class `.tab-content`.

If you want the tabs to fade in and out when clicking on them, add the `.fade` class to `.tab-pane`:

```
<!-- Nav tabs -->
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" data-bs-toggle="tab" href="#home">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" data-bs-toggle="tab" href="#menu1">Menu 1</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" data-bs-toggle="tab" href="#menu2">Menu 2</a>
  </li>
</ul>

<!-- Tab panes -->
<div class="tab-content">
  <div class="tab-pane container active" id="home">...</div>
  <div class="tab-pane container fade" id="menu1">...</div>
  <div class="tab-pane container fade" id="menu2">...</div>
</div>
```

## Bootstrap 5 Navbars

### Navigation Bars

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

## Basic Navbar

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xxl|xl|lg|md|sm` (stacks the navbar vertically on xxlarge, extra large, large, medium or small screens).

To add links inside the navbar, use either an `<ul>` element (or a `<div>`) with `class="navbar-nav"`. Then add `<li>` elements with a `.nav-item` class followed by an `<a>` element with a `.nav-link` class:

```
<!-- A grey horizontal navbar that becomes vertical on small screens -->
<nav class="navbar navbar-expand-sm bg-light">

  <div class="container-fluid">
    <!-- Links -->
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Link 1</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link 2</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link 3</a>
      </li>
    </ul>
  </div>

</nav>
```

## Centered Navbar

Add the `.justify-content-center` class to center the navigation bar:

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">
  ...
</nav>
```

## Colored Navbar

Use any of the `.bg-color` classes to change the background color of the navbar (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`)

**Tip:** Add a **white** text color to all links in the navbar with the `.navbar-dark` class, or use the `.navbar-light` class to add a **black** text color.

## Brand / Logo

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

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

When using the `.navbar-brand` class with images, Bootstrap 5 will automatically style the image to fit the navbar vertically.

## Navbar Text

Use the `.navbar-text` class to vertical align any elements inside the navbar that are not links (ensures proper padding and text color).

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <span class="navbar-text">Navbar text</span>
  </div>
</nav>
```

Very often, especially on small screens, you want to hide the navigation links and replace them with a button that should reveal them when clicked on.

To create a collapsible navigation bar, use a button with `class="navbar-toggler"`, `data-bs-toggle="collapse"` and `data-bs-target="#thetarget"`. Then wrap the navbar content (links, etc) inside a `<div>` element with `class="collapse navbar-collapse"`, followed by an id that matches the `data-bs-target` of the button: `"thetarget"`.

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Logo</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

## Fixed Navigation Bar

The navigation bar can also be fixed at the top or at the bottom of the page. A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

The `.fixed-top` class makes the navigation bar fixed at the **top**:



Use the `.fixed-bottom` class to make the navbar stay at the **bottom** of the page:

Use the `.sticky-top` class to make the navbar fixed/stay at the **top** of the page when you scroll **past** it. **Note:** This class does not work in IE11 and earlier (will treat it as `position:relative`).

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark sticky-top">
...
</nav>
```

## Bootstrap 5 Carousel

### Carousel / Slideshow

The Carousel is a slideshow for cycling through elements:

#### How To Create a Carousel

The following example shows how to create a basic carousel with indicators and controls:

```
<!-- Carousel -->
<div id="demo" class="carousel slide" data-bs-ride="carousel">

  <!-- Indicators/dots -->
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#demo" data-bs-slide-to="0" class="active"></button>
    <button type="button" data-bs-target="#demo" data-bs-slide-to="1"></button>
    <button type="button" data-bs-target="#demo" data-bs-slide-to="2"></button>
  </div>

  <!-- The slideshow/carousel -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
```

```

    

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

<!-- Left and right controls/icons -->
<button class="carousel-control-prev" type="button" data-bs-target="#demo" data-bs-
slide="prev">
  <span class="carousel-control-prev-icon"></span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#demo" data-bs-
slide="next">
  <span class="carousel-control-next-icon"></span>
</button>
</div>

```

Add elements inside `<div class="carousel-caption">` within each `<div class="carousel-item">` to create a caption for each slide:

```

<div class="carousel-item">
  
  <div class="carousel-caption">
    <h3>Los Angeles</h3>
    <p>We had such a great time in LA!</p>
  </div>
</div>

```

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

## Bootstrap 5 Modal

### Modals

The Modal component is a dialog box/popup window that is displayed on top of the current page

### How To Create a Modal

The following example shows how to create a basic modal:

```
<!-- Button to Open the Modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-
target="#myModal">
  Open modal
</button>

<!-- The Modal -->
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">

      <!-- Modal Header -->
      <div class="modal-header">
        <h4 class="modal-title">Modal Heading</h4>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>

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

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

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

## Add animation

Use the `.fade` class to add a fading effect when opening and closing the modal

```
<!-- Fading modal -->  
<div class="modal fade"></div>
```

```
<!-- Modal without animation -->  
<div class="modal"></div>
```

## Modal Size

Change the size of the modal by adding the `.modal-sm` class for small modals (max-width 300px), `.modal-lg` class for large modals (max-width 800px), or `.modal-xl` for extra large modals (max-width 1140px). Default is 500px max-width.

Add the size class to the `<div>` element with class `.modal-dialog`:

### Small Modal

```
<div class="modal-dialog modal-sm">
```

### Large Modal

```
<div class="modal-dialog modal-lg">
```

### Extra Large Modal

```
<div class="modal-dialog modal-xl">
```

## Fullscreen Modals

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

```
<div class="modal-dialog modal-fullscreen">
```

## Responsive Fullscreen Modals

You can also control when the modal should be in fullscreen, with the `.modal-fullscreen-*` classes:

Class	Description
<code>.modal-fullscreen-sm-down</code>	Fullscreen below 576px
<code>.modal-fullscreen-md-down</code>	Fullscreen below 768px
<code>.modal-fullscreen-lg-down</code>	Fullscreen below 992px
<code>.modal-fullscreen-xl-down</code>	Fullscreen below 1200px
<code>.modal-fullscreen-xxl-down</code>	Fullscreen below 1400px

## Centered Modal

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

```
<div class="modal-dialog modal-dialog-centered">
```

## Scrolling Modal

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

```
<div class="modal-dialog">
```

However, it is possible to only scroll inside the modal, instead of the page itself, by adding `.modal-dialog-scrollable` to `.modal-dialog`:

## Bootstrap 5 Tooltip

### Tooltips

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

### How To Create a Tooltip

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

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

```
<button type="button" class="btn btn-primary" data-bs-toggle="tooltip" title="Hooray!">Hover over me!</button>
```

**Note:** Tooltips must be initialized with JavaScript to work.

The following code will enable all tooltips in the document:

```
<script>
var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'))
var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {
  return new bootstrap.Tooltip(tooltipTriggerEl)
})
</script>
```

### Positioning Tooltips

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

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

```
<a href="#" data-bs-toggle="tooltip" data-bs-placement="top" title="Hooray!">Hover</a>
<a href="#" data-bs-toggle="tooltip" data-bs-placement="bottom" title="Hooray!">Hover</a>
<a href="#" data-bs-toggle="tooltip" data-bs-placement="left" title="Hooray!">Hover</a>
<a href="#" data-bs-toggle="tooltip" data-bs-placement="right" title="Hooray!">Hover</a>
```

## Bootstrap 5 Popover

### Popovers

The Popover component is similar to tooltips; it is a pop-up box that appears when the user clicks on an element. The difference is that the popover can contain much more content.

### How To Create a Popover

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

Use the `title` attribute to specify the header text of the popover, and use the `data-bs-content` attribute to specify the text that should be displayed inside the popover's body:

```
<button type="button" class="btn btn-primary" data-bs-  
toggle="popover" title="Popover Header" data-bs-content="Some content inside the  
popover">Toggle popover</button>
```

**Note:** Popovers must be initialized with JavaScript to work.

The following code will enable all popovers in the document:

```
<script>  
var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-  
toggle="popover"]'))  
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {  
  return new bootstrap.Popover(popoverTriggerEl)  
})  
</script>
```

### Positioning Popovers

By default, the popover will appear on the right side of the element.

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

```
<a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="top" data-  
content="Content">Top</a>
```



```
<a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="bottom" data-  
content="Content">Bottom</a>  
<a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="left" data-  
content="Content">Left</a>  
<a href="#" title="Header" data-bs-toggle="popover" data-bs-placement="right" data-  
content="Content">Right</a>
```

## Closing Popovers

By default, the popover is closed when you click on the element again. However, you can use the `data-bs-trigger="focus"` attribute which will close the popover when clicking outside the element:

```
<a href="#" title="Dismissible popover" data-bs-toggle="popover" data-bs-  
trigger="focus" data-bs-content="Click anywhere in the document to close this  
popover">Click me</a>
```

## Hoverable Popover

**Tip:** If you want the popover to be displayed when you move the mouse pointer over the element, use the `data-bs-trigger` attribute with a value of `"hover"`:

```
<a href="#" title="Header" data-bs-toggle="popover" data-bs-trigger="hover" data-bs-  
content="Popover text">Hover over me</a>
```

## Bootstrap 5 Toasts

### Toasts

The toast component is like an alert box that is only shown for a couple of seconds when something happens (i.e. when the user clicks on a button, submits a form, etc.).

### How To Create a Toast

To create a toast, use the `.toast` class, and add a `.toast-header` and a `.toast-body` inside of it.

**Note:** Toasts are hidden by default. Use the `.show` class if you want to display it. To close it, use a `<button>` element and add `data-bs-dismiss="toast"`:

```
<div class="toast show">
  <div class="toast-header">
    Toast Header
    <button type="button" class="btn-close" data-bs-dismiss="toast"></button>
  </div>
  <div class="toast-body">
    Some text inside the toast body
  </div>
</div>
```

## Open a Toast

To show a toast with a click of a button, you must initialize it with JavaScript: select the specified element and call the `toast()` method.

The following code will show all "toasts" in the document when you click on a button:

```
<script>
document.getElementById("toastbtn").onclick = function() {
  var toastElList = [].slice.call(document.querySelectorAll('.toast'))
  var toastList = toastElList.map(function(toastEl) {
    return new bootstrap.Toast(toastEl)
  })
  toastList.forEach(toast => toast.show())
}
</script>
```

## Bootstrap 5 Scrollspy

### Scrollspy

Scrollspy is used to automatically update links in a navigation list based on **scroll** position.

### How To Create a Scrollspy

The following example shows how to create a scrollspy:

```
<!-- The scrollable area -->
<body data-bs-spy="scroll" data-bs-target=".navbar" data-bs-offset="50">

<!-- The navbar - The <a> elements are used to jump to a section in the scrollable area -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">
...
  <ul class="navbar-nav">
    <li><a href="#section1">Section 1</a></li>
    ...
  </ul>
</nav>

<!-- Section 1 -->
<div id="section1">
  <h1>Section 1</h1>
  <p>Try to scroll this page and look at the navigation bar while scrolling!</p>
</div>
...

</body>
```

### Example Explained

Add **data-bs-spy="scroll"** to the element that should be used as the scrollable area (often this is the **<body>** element).

Then add the **data-bs-target** attribute with a value of the id or the class name of the navigation bar (**.navbar**). This is to make sure that the navbar is connected with the scrollable area.

Note that scrollable elements must match the ID of the links inside the navbar's list items (**<div id="section1">** matches **<a href="#section1">**).

The optional **data-bs-offset** attribute specifies the number of pixels to offset from top when calculating the position of scroll. This is useful when you feel that the links inside the navbar changes the active state too soon or too early when jumping to the scrollable elements. Default is 10 pixels.

## Bootstrap 5 Offcanvas

### Offcanvas

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

### How To Create an Offcanvas Sidebar

The following example shows how to create an offcanvas sidebar:

```
<!-- Offcanvas Sidebar -->
<div class="offcanvas offcanvas-start" id="demo">
  <div class="offcanvas-header">
    <h1 class="offcanvas-title">Heading</h1>
    <button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas"></button>
  </div>
  <div class="offcanvas-body">
    <p>Some text lorem ipsum.</p>
    <p>Some text lorem ipsum.</p>
    <button class="btn btn-secondary" type="button">A Button</button>
  </div>
</div>

<!-- Button to open the offcanvas sidebar -->
<button class="btn btn-primary" type="button" data-bs-toggle="offcanvas" data-bs-target="#demo">
  Open Offcanvas Sidebar
</button>
```

### Example Explained

The `.offcanvas` class creates the offcanvas sidebar.

The `.offcanvas-start` class positions the offcanvas, and makes it 400px wide. See examples below for more positioning classes.

The `.offcanvas-title` class ensures proper margins and line-height.

Then, add your content inside the `.offcanvas-body` class.

To open the offcanvas sidebar, you must use a `<button>` or an `<a>` element that points to the id of the `.offcanvas` container (`#demo` in our example).

To open the offcanvas sidebar with an `<a>` element, you can point to `#demo` with the `href` attribute, instead of `data-bs-target` attribute.

## Offcanvas Position

Use the `.offcanvas-start|end|top|bottom` to position the offcanvas to the left, right, top or bottom:

```
<div class="offcanvas offcanvas-end" id="demo">
```

## Responsive OffCanvas Menu

You can also control when you want to hide or show the offcanvas menu on different screen widths, with the `.offcanvas-sm|md|lg|xl|xxl` classes:

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

## Dark OffCanvas Menu

Use the `.text-bg-dark` class to create a dark offcanvas menu.

**Tip:** We have also added the `.btn-close-white` class to `.btn-close`, to create a white close button that looks nice with the dark background:

```
<div class="offcanvas offcanvas-end" id="demo">
<button type="button" class="btn-close btn-close-white" data-bs-
dismiss="offcanvas"></button>
```

## Bootstrap 5 Utilities

### Utilities / Helper Classes

Bootstrap 5 has a lot of utility/helper classes to quickly style elements without using any CSS code.

#### Borders

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

```
<span class="border"></span>  
<span class="border border-0"></span>  
<span class="border border-top-0"></span>  
<span class="border border-end-0"></span>  
<span class="border border-bottom-0"></span>  
<span class="border border-start-0"></span>  
<br>
```

```
<span class="border-top"></span>  
<span class="border-end"></span>  
<span class="border-bottom"></span>  
<span class="border-start"></span>
```

#### Border Width

Use **.border-1** to **.border-5** to change the width of the border:

```
<span class="border border-1"></span>  
<span class="border border-2"></span>  
<span class="border border-3"></span>  
<span class="border border-4"></span>  
<span class="border border-5"></span>
```

#### Border Color

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

```
<span class="border border-primary"></span>  
<span class="border border-secondary"></span>
```

## Border Radius

Add rounded corners to an element with the **rounded** classes:

```
<span class="rounded"></span>
<span class="rounded-top"></span>
<span class="rounded-end"></span>
<span class="rounded-bottom"></span>
<span class="rounded-start"></span>
<span class="rounded-circle"></span>
<span class="rounded-pill" style="width:130px"></span>
<span class="rounded-0"></span>
<span class="rounded-1"></span>
<span class="rounded-2"></span>
<span class="rounded-3"></span>
<span class="rounded-4"></span>
<span class="rounded-5"></span>
```

## Float and Clearfix

Float an element to the right with the **.float-end** class or to the left with **.float-start**, and clear floats with the **.clearfix** class:

```
<div class="clearfix">
  <span class="float-start">Float left</span>
  <span class="float-end">Float right</span>
</div>
```

## Responsive Floats

Float an element to the left or to the right depending on screen width, with the responsive float classes (**.float-*\**-start|end** - where *\** is **sm** ( $\geq 576\text{px}$ ), **md** ( $\geq 768\text{px}$ ), **lg** ( $\geq 992\text{px}$ ), **xl** ( $\geq 1200\text{px}$ ) or **xxl** ( $\geq 1400\text{px}$ )):

## Center Align

Center an element with the **.mx-auto** class (adds margin-left and margin-right: auto):

```
<div class="mx-auto bg-warning" style="width:150px">Centered</div>
```

## Width

Set the width of an element with the w-\* classes (.w-25, .w-50, .w-75, .w-100, .mw-auto, .mw-100):

```
<div class="w-25 bg-warning">Width 25%</div>
<div class="w-50 bg-warning">Width 50%</div>
<div class="w-75 bg-warning">Width 75%</div>
<div class="w-100 bg-warning">Width 100%</div>
<div class="w-auto bg-warning">Auto Width</div>
<div class="mw-100 bg-warning">Max Width 100%</div>
```

## Height

Set the height of an element with the h-\* classes (.h-25, .h-50, .h-75, .h-100, .mh-auto, .mh-100):

```
<div style="height:200px;background-color:#ddd">
  <div class="h-25 bg-warning">Height 25%</div>
  <div class="h-50 bg-warning">Height 50%</div>
  <div class="h-75 bg-warning">Height 75%</div>
  <div class="h-100 bg-warning">Height 100%</div>
  <div class="h-auto bg-warning">Auto Height</div>
  <div class="mh-100 bg-warning" style="height:500px">Max Height
100%</div>
</div>
```

## Spacing

Bootstrap 5 has a wide range of responsive margin and padding utility classes. They work for all breakpoints: **xs** (<=576px), **sm** (>=576px), **md** (>=768px), **lg** (>=992px), **xl** (>=1200px) or **xxl** (>=1400px):

The classes are used in the format: {property}{sides}-{size} for **xs** and {property}{sides}-{breakpoint}-{size} for **sm**, **md**, **lg**, **xl** and **xxl**.

Where *property* is one of:

**m** - sets **margin**      **p** - sets **padding**



Where *sides* is one of:

- **t** - sets **margin-top** or **padding-top**
- **b** - sets **margin-bottom** or **padding-bottom**
- **s** - sets **margin-left** or **padding-left**
- **e** - sets **margin-right** or **padding-right**
- **x** - sets both **padding-left** and **padding-right** or **margin-left** and **margin-right**
- **y** - sets both **padding-top** and **padding-bottom** or **margin-top** and **margin-bottom**
- blank - sets a **margin** or **padding** on all 4 sides of the element

Where *size* is one of:

- **0** - sets **margin** or **padding** to **0**
- **1** - sets **margin** or **padding** to **.25rem**
- **2** - sets **margin** or **padding** to **.5rem**
- **3** - sets **margin** or **padding** to **1rem**
- **4** - sets **margin** or **padding** to **1.5rem**
- **5** - sets **margin** or **padding** to **3rem**
- **auto** - sets **margin** to **auto**

## Shadows

Use the **shadow-** classes to add shadows to an element

```
<div class="shadow-none p-4 mb-4 bg-light">No shadow</div>
<div class="shadow-sm p-4 mb-4 bg-white">Small shadow</div>
<div class="shadow p-4 mb-4 bg-white">Default shadow</div>
<div class="shadow-lg p-4 mb-4 bg-white">Large shadow</div>
```

## Vertical Align

Use the **align-** classes to change the alignment of elements (only works on inline, inline-block, inline-table and table cell elements):

```
<span class="align-baseline">baseline</span>
<span class="align-top">top</span>
<span class="align-middle">middle</span>
<span class="align-bottom">bottom</span>
<span class="align-text-top">text-top</span>
```

## Aspect Ratio

Create responsive video or slideshows based on the width of the parent.

Add the `.ratio` class together with an aspect ratio of your choice `.ratio-*` to a parent element, and add the embed (video or iframe) inside of it:

```
<!-- Aspect ratio 1:1 -->
<div class="ratio ratio-1x1">
  <iframe src="https://www.youtube.com/embed/tgbNymZ7vqY"></iframe>
</div>

<!-- Aspect ratio 4:3 -->
<div class="ratio ratio-4x3">
  <iframe src="https://www.youtube.com/embed/tgbNymZ7vqY"></iframe>
</div>

<!-- Aspect ratio 16:9 -->
<div class="ratio ratio-16x9">
  <iframe src="https://www.youtube.com/embed/tgbNymZ7vqY"></iframe>
</div>

<!-- Aspect ratio 21:9 -->
<div class="ratio ratio-21x9">
  <iframe src="https://www.youtube.com/embed/tgbNymZ7vqY"></iframe>
</div>
```

## Visibility

Use the `.visible` or `.invisible` classes to control the visibility of elements. **Note:** These classes do not change the CSS display value. They only add `visibility:visible` or `visibility:hidden`

```
<div class="visible">I am visible</div>
<div class="invisible">I am invisible</div>
```

## Close icon

Use the `.btn-close` class to style a close icon. This is often used for alerts and modals.

## Colors

As described in the Colors chapter, here is a list of all text and background color classes:

The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body` (default body color/often black) and `.text-light`:

You can also add 50% opacity for black or white text with the `.text-black-50` or `.text-white-50` classes:

## Background Colors

The classes for background colors are: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`.

The `.bg-color` classes above does not work well with text, or atleast then you have to specify a proper `.text-color` class to get the right text color for each background.

However, you can use the `.text-bg-color` classes and Bootstrap will automatically handle the appropriate text color for each background color.

## Bootstrap 5 Flex

### Flexbox

The biggest difference between Bootstrap 3 and Bootstrap 4 & 5 is that Bootstrap 5 now uses flexbox, instead of floats, to handle the layout.

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

To create a flexbox container and to transform direct children into flex items, use the `d-flex` class:

```
<div class="d-flex p-3 bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
</div>
```

To create an inline flexbox container, use the `d-inline-flex` class:

```
<div class="d-inline-flex p-3 bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```

## Horizontal Direction

Use `.flex-row` to display the flex items horizontally (side by side). This is default.

**Tip:** Use `.flex-row-reverse` to right-align the horizontal direction:

```
<div class="d-flex flex-row bg-secondary">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>

<div class="d-flex flex-row-reverse bg-secondary">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```

## Vertical Direction

Use `.flex-column` to display the flex items vertically (on top of each other), or `.flex-column-reverse` to reverse the vertical direction

## Justify Content

Use the `.justify-content-*` classes to change the alignment of flex items. Valid classes are `start` (default), `end`, `center`, `between` or `around`

```
<div class="d-flex justify-content-start">...</div>
<div class="d-flex justify-content-end">...</div>
```

## Fill / Equal Widths

Use `.flex-fill` on flex items to force them into equal widths:

```
<div class="d-flex">
  <div class="p-2 bg-info flex-fill">Flex item 1</div>
  <div class="p-2 bg-warning flex-fill">Flex item 2</div>
  <div class="p-2 bg-primary flex-fill">Flex item 3</div>
</div>
```

## Grow

Use `.flex-grow-1` on a flex item to take up the rest of the space. In the example below, the first two flex items take up their necessary space, while the last item takes up the rest of the available space:

```
<div class="d-flex">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary flex-grow-1">Flex item 3</div>
</div>
```

**Tip:** Use `.flex-shrink-1` on a flex item to make it shrink if necessary

## Order

Change the visual order of a specific flex item(s) with the `.order` classes. Valid classes are from 0 to 5, where the lowest number has highest priority (order-1 is shown before order-2, etc.):

```
<div class="d-flex bg-secondary">
  <div class="p-2 bg-info order-3">Flex item 1</div>
  <div class="p-2 bg-warning order-2">Flex item 2</div>
  <div class="p-2 bg-primary order-1">Flex item 3</div>
</div>
```

## Auto Margins

Easily add auto margins to flex items with `.ms-auto` (push items to the right), or by using `.me-auto` (push items to the left):

```
<div class="d-flex bg-secondary">
  <div class="p-2 ms-auto bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```

```
<div class="d-flex bg-secondary">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 me-auto bg-primary">Flex item 3</div>
</div>
```

## Wrap

Control how flex items wrap in a flex container with `.flex-nowrap` (default), `.flex-wrap` or `.flex-wrap-reverse`.

Click on the buttons below to see the difference between the three classes, by changing the wrapping of the flex items in the example box:

```
<div class="d-flex flex-wrap">..  
</div>

<div class="d-flex flex-wrap-reverse">..  
</div>

<div class="d-flex flex-nowrap">..  
</div>
```

## Align Content

Control the vertical alignment of **gathered** flex items with the `.align-content-*` classes. Valid classes are `.align-content-start` (default), `.align-content-end`, `.align-content-center`, `.align-content-between`, `.align-content-around` and `.align-content-stretch`.

**Note:** These classes have no effect on single rows of flex items.

```

<div class="d-flex flex-wrap align-content-start">..</div>
<div class="d-flex flex-wrap align-content-end">..</div>
<div class="d-flex flex-wrap align-content-center">..</div>
<div class="d-flex flex-wrap align-content-around">..</div>
<div class="d-flex flex-wrap align-content-stretch">..</div>

```

## Align Items

Control the vertical alignment of **single rows** of flex items with the `.align-items-*` classes. Valid classes are `.align-items-start`, `.align-items-end`, `.align-items-center`, `.align-items-baseline`, and `.align-items-stretch` (default).

```

<div class="d-flex align-items-start">..</div>
<div class="d-flex align-items-end">..</div>
<div class="d-flex align-items-center">..</div>
<div class="d-flex align-items-baseline">..</div>
<div class="d-flex align-items-stretch">..</div>

```

## Bootstrap 5 Forms

### Stacked Form

All textual `<input>` and `<textarea>` elements with class `.form-control` get proper form styling:

```

<form action="/action_page.php">
  <div class="mb-3 mt-3">
    <label for="email" class="form-label">Email:</label>
    <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
  </div><button type="submit" class="btn btn-primary">Submit</button>
</form>

```

Also note that we add a `.form-label` class to each label element to ensure correct padding.

Checkboxes have different markup. They are wrapped around a container element with `.form-check`, and labels have a class of `.form-check-label`, while checkboxes and radio buttons use `.form-check-input`.

```
<label for="comment">Comments:</label>
<textarea class="form-
control" rows="5" id="comment" name="text"></textarea>
```

## Form Control Size

You can change the size of `.form-control` inputs with `.form-control-lg` or `.form-control-sm`:

```
<input type="text" class="form-control form-control-lg" placeholder="Large
input">
<input type="text" class="form-control" placeholder="Normal input">
<input type="text" class="form-control form-control-sm" placeholder="Small
input">
```

## Disabled and Readonly

Use the `disabled` and/or `readonly` attributes to disable the input field:

```
<input type="text" class="form-control" placeholder="Normal input">
<input type="text" class="form-control" placeholder="Disabled
input" disabled>
<input type="text" class="form-control" placeholder="Readonly
input" readonly>
```

## Plain text Inputs

Use the `.form-control-plaintext` class to style an input field without borders, but keep proper margins and padding:

```
<input type="text" class="form-control-plaintext" placeholder="Plaintext
input">
<input type="text" class="form-control" placeholder="Normal input">
```



## Color Picker

To style an input with type="color" properly, use the `.form-control-color` class:

```
<input type="color" class="form-control form-control-color" value="#CCCCCC">
```

## Bootstrap 5 Select

Select menus are used if you want to allow the user to pick from multiple options. To style a select menu in Bootstrap 5, add the `.form-select` class to the `<select>` element:

```
<select class="form-select">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
</select>
```

## Select Menu Size

Use the `.form-select-lg` or `.form-select-sm` class to change the size of the select menu:

```
<select class="form-select form-select-lg">
<select class="form-select">
<select class="form-select form-select-sm">
```

## Disabled Select Menu

Use the `disabled` attribute to disable the select menu

```
<select class="form-select" disabled>
```

## Data Lists

Bootstrap will also style data lists, which is a list of pre-defined options for an `<input>` element:

Choose your browser from the list:

```
<label for="browser" class="form-label">Choose your browser from the  
list:</label>  
<input class="form-control" list="browsers" name="browser" id="browser">  
<datalist id="browsers">  
  <option value="Edge">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

## Bootstrap 5 Checkboxes and Radio buttons

### Checkboxes

Checkboxes are used if you want the user to select any number of options from a list of preset options.

```
<div class="form-check">  
  <input class="form-check-  
input" type="checkbox" id="check1" name="option1" value="something" checked  
>  
  <label class="form-check-label">Option 1</label>  
</div>
```

### *Example Explained*

To style checkboxes, use a wrapper element with `class="form-check"` to ensure proper margins for labels and checkboxes.

Then, add the `.form-check-label` class to label elements, and `.form-check-input` to style checkboxes properly inside the `.form-check` container.

Use the `checked` attribute if you want the checkbox to be checked by default.

### Radio buttons

Radio buttons are used if you want to limit the user to just one selection from a list of preset options.

```

<div class="form-check">
  <input type="radio" class="form-check-
input" id="radio1" name="optradio" value="option1" checked>Option 1
  <label class="form-check-label" for="radio1"></label>
</div>
<div class="form-check">
  <input type="radio" class="form-check-
input" id="radio2" name="optradio" value="option2">Option 2
  <label class="form-check-label" for="radio2"></label>
</div>
<div class="form-check">
  <input type="radio" class="form-check-input" disabled>Option 3
  <label class="form-check-label"></label>
</div>

```

## Toggle Switches

If you want your checkbox to be styled as a toggle switch, use the `.form-switch` class together with the `.form-check` container:

```

<div class="form-check form-switch">
  <input class="form-check-
input" type="checkbox" id="mySwitch" name="darkmode" value="yes" checked>
  <label class="form-check-label" for="mySwitch">Dark Mode</label>
</div>

```

## Bootstrap 5 Range

### Custom Range

To style a range menu, add the `.form-range` class to the input element with `type="range"`:

```

<label for="customRange" class="form-label">Custom range</label>
<input type="range" class="form-range" id="customRange">

```

## Steps

By default, the interval between the range numbers is 1. You can change it by using the `step` attribute:

## Min and Max

By default, the minimum value is 0 and maximum value is 100. You can use the `min` and/or `max` attribute change it:

```
<input type="range" class="form-range" min="0" max="4">
```

## Bootstrap 5 Input Groups

### Input Groups

The `.input-group` class is a container to enhance an input by adding an icon, text or a button in front or behind the input field as a "help text".

To style the specified help text, use the `.input-group-text` class:

```
<form>
  <div class="input-group">
    <span class="input-group-text">@</span>
    <input type="text" class="form-control" placeholder="Username">
  </div>

  <div class="input-group">
    <input type="text" class="form-control" placeholder="Your Email">
    <span class="input-group-text">@example.com</span>
  </div>
</form>
```

### Input Group Size

Use the `.input-group-sm` class for small input groups and `.input-group-lg` for large inputs groups:

### Multiple Inputs and Helpers

Add multiple inputs or addons:

```

<!-- Multiple inputs -->
<div class="input-group mb-3">
  <span class="input-group-text">Person</span>
  <input type="text" class="form-control" placeholder="First Name">
  <input type="text" class="form-control" placeholder="Last Name">
</div>

<!-- Multiple addons / help text -->
<div class="input-group mb-3">
  <span class="input-group-text">One</span>
  <span class="input-group-text">Two</span>
  <span class="input-group-text">Three</span>
  <input type="text" class="form-control">
</div>

```

## Input Group with Checkboxes and Radios

You can also use checkboxes or radio buttons instead of text:

```

<div class="input-group mb-3">
  <div class="input-group-text">
    <input type="checkbox">
  </div>
  <input type="text" class="form-control" placeholder="Some text">
</div>

```

## Input Group Buttons

```

<div class="input-group mb-3">
  <button class="btn btn-outline-primary" type="button">Basic
  Button</button>
  <input type="text" class="form-control" placeholder="Some text">
</div>

```

## Input Group with Dropdown Button

Add a dropdown button in the input group. Note that you don't need the .dropdown wrapper, as you normally would.

```

<div class="input-group mt-3 mb-3">
  <button type="button" class="btn btn-primary dropdown-toggle" data-bs-
toggle="dropdown">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Link 1</a></li>
    <li><a class="dropdown-item" href="#">Link 2</a></li>
    <li><a class="dropdown-item" href="#">Link 3</a></li>
  </ul>
  <input type="text" class="form-control" placeholder="Username">
</div>

```

## Bootstrap 5 Form Floating Labels

### Floating Labels / Animated Labels

By default, when using labels, they normally appear on top of the input field:

With floating labels, you can insert the label inside the input field, and make them float/animate when you click on the input field:

```

<div class="form-floating mb-3 mt-3">
  <input type="text" class="form-control" id="email" placeholder="Enter
email" name="email">
  <label for="email">Email</label>
</div>

<div class="form-floating mt-3 mb-3">
  <input type="text" class="form-control" id="pwd" placeholder="Enter
password" name="pswd">
  <label for="pwd">Password</label>
</div>

```

Notes on floating labels: The <label> elements must come after the <input> element, and the placeholder attribute is required for each <input> element (even though it is not shown).

## Bootstrap 5 Form Validation

You can use different validation classes to provide valuable feedback to users. Add either `.was-validated` or `.needs-validation` to the `<form>` element, depending on whether you want to provide validation feedback before or after submitting the form. The input fields will have a green (valid) or red (invalid) border to indicate what's missing in the form. You can also add a `.valid-feedback` or `.invalid-feedback` message to tell the user explicitly what's missing, or needs to be done before submitting the form.

```
<form action="/action_page.php" class="was-validated">
  <div class="mb-3 mt-3">
    <label for="uname" class="form-label">Username:</label>
    <input type="text" class="form-control" id="uname" placeholder="Enter
username" name="uname" required>
    <div class="valid-feedback">Valid.</div>
    <div class="invalid-feedback">Please fill out this field.</div>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

## Bootstrap 5 Grid System

### The Grid System

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

## Grid Classes

The Bootstrap 5 grid system has six classes:

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)
- **.col-xxl-** (xxlarge devices - screen width equal to or greater than 1400px)

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

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

## Bootstrap 5 Grid

### Grid Example

	XSmall	Small	Medium	Large	Extra Large	XXL
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px	>=1400px

Assume we have a simple layout with two columns. We want the columns to split 25%/75% for **ALL** devices.

We will add the following classes to our two columns:

```
<div class="col-3">.....</div>
<div class="col-9">.....</div>
```



```
<div class="col-sm-3">.....</div>
<div class="col-sm-9">.....</div>
```

```
<div class="col-sm-3 col-md-6">..... </div>
<div class="col-sm-9 col-md-6">..... </div>
```

```
<div class="col-sm-3 col-md-6 col-lg-4">..... </div>
<div class="col-sm-9 col-md-6 col-lg-8">..... </div>
```

```
<div class="col-sm-3 col-md-6 col-lg-4 col-xl-2">.....</div>
<div class="col-sm-9 col-md-6 col-lg-8 col-xl-10">.....</div>
```

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-6 col-xxl-3">
      <p>Lorem ipsum.....</p>
    </div>
    <div class="col-md-6 col-xxl-9">
      <p>Sed ut perspiciatis.....</p>
    </div>
  </div>
</div>
```