**Bootstrap 5**

Bootstrap is the most popular and powerful front-end (HTML, CSS, and JavaScript) framework for faster and easier responsive web development.It includes HTML and CSS based design templates for creating common user interface components like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and so on.Bootstrap was originally created by a designer and a developer at Twitter in mid-2010. Before being an open-sourced framework, Bootstrap was known as Twitter Blueprint.

* create responsive websites.
* You can quickly create multi-column layout with pre-defined classes.
* You can quickly create different types of form layouts.
* You can quickly create different variation of navigation bar.
* You can easily create components like accordions, modals, etc. without writing any JS code.
* You can easily create dynamic tabs to manage large amount of content.
* You can easily create tooltips and popovers to show hint text.
* You can easily create carousel or image slider to showcase your content.
* You can quickly create different types of alert boxes.

The list does not end here, there are many other interesting things that you can do with Bootstrap. You will learn about all of them in detail in upcoming chapters.

**Advantages of Using Bootstrap**

If you have had some experience with any front-end framework, you might be wondering what makes Bootstrap so special. Here are some advantages why one should opt for Bootstrap framework:

* **Save lots of time** — You can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.
* **Responsive features** — Using Bootstrap you can easily create responsive websites that appear more appropriately on different devices and screen resolutions without any change in markup.
* **Consistent design** — All Bootstrap components share the same design templates and styles through a central library, so the design and layout of your web pages will be consistent.
* **Easy to use** — Bootstrap is very easy to use. Anybody with the basic working knowledge of HTML, CSS and JavaScript can start development with Bootstrap.
* **Compatible with browsers** — Bootstrap is created with modern web browsers in mind and it is compatible with all modern browsers such as Chrome, Firefox, Safari, Internet Explorer, etc.
* **Open Source** — And the best part is, it is completely free to download and use.

**Note:** Bootstrap is responsive by default with a mobile first approach. Bootstrap 5 is the latest and most stable version of the Bootstrap. Bootstrap 5 is supported in all major modern browsers such Google Chrome, Firefox, Safari, Internet Explorer 10 and above, etc.

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

Bootstrap provides three different types containers:

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

| **Classes**  **Bootstrap  Grid System** | **X-Small**  **<576px** | **Small**  **≥576px** | **Medium**  **≥768px** | **Large**  **≥992px** | **X-Large**  **≥1200px** | **XX-Large**  **≥1400px** |
| --- | --- | --- | --- | --- | --- | --- |
| .container | 100% | 540px | 720px | 960px | 1140px | 1320px |
| .container-sm | 100% | 540px | 720px | 960px | 1140px | 1320px |
| .container-md | 100% | 100% | 720px | 960px | 1140px | 1320px |
| .container-lg | 100% | 100% | 100% | 960px | 1140px | 1320px |
| .container-xl | 100% | 100% | 100% | 100% | 1140px | 1320px |
| .container-xxl | 100% | 100% | 100% | 100% | 100% | 1320px |
| .container-fluid | 100% | 100% | 100% | 100% | 100% | 100% |

**Creating Responsive Fixed-width Containers**

You can simply use the .container class to create a responsive, fixed-width container. The width of the container will change at different breakpoints or screen sizes, as shown above.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Responsive Fixed-width Containers</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="container">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

<p class="mt-3 text-info"><strong>Tip:</strong> Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</p>

</div>

</body>

</html>

FLUID CONTAINERS: to create a full width container, that will always span the entire width of the screen (width is always 100%):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Fluid Containers</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="container-fluid">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</div>

</body>

</html>

## Specify Responsive Breakpoints for Containers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Containers with Responsive Breakpoints</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="container-sm border py-3 my-3">100% wide until screen size less than 576px</div>

<div class="container-md border py-3 my-3">100% wide until screen size less than 768px</div>

<div class="container-lg border py-3 my-3">100% wide until screen size less than 992px</div>

<div class="container-xl border py-3 my-3">100% wide until screen size less than 1200px</div>

<p class="m-3"><strong>Note:</strong> The classes ".border", ".py-3", and ".my-3" are <a href="/twitter-bootstrap-tutorial/bootstrap-helper-classes.php" target="\_top">utility classes</a> to add borders, paddings, and margins to the element, respectively.</p>

<p class="m-3"><strong>Tip:</strong> Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</p>

</body>

</html>

## Adding Background and Borders to Containers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Set Background and Borders to Bootstrap Containers</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<!-- Container with dark background and white text color -->

<div class="container bg-dark text-white border py-3 my-3">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</div>

<!-- Container with light background -->

<div class="container bg-light py-3 my-3">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</div>

<!-- Container with border -->

<div class="container border py-3 my-3">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</div>

<p class="m-3"><strong>Note:</strong> The classes ".py-3", and ".my-3" are <a href="/twitter-bootstrap-tutorial/bootstrap-helper-classes.php" target="\_top">spacing utility classes</a> to add padding, and margin of 1rem to the top and bottom sides of the element, respectively.</p>

</body>

</html>

## Applying Paddings and Margins to Containers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Set Paddings and Margins to Bootstrap Containers</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<!-- Container with border, extra paddings and margins -->

<div class="container border py-3 my-3">

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</div>

<p class="m-3"><strong>Note:</strong> The classes ".py-3", and ".my-3" are <a href="/twitter-bootstrap-tutorial/bootstrap-helper-classes.php" target="\_top">spacing utility classes</a> to add padding, and margin of 1rem to the top and bottom sides of the element, respectively.</p>

</body>

</html>

**Tip:** Avoid setting left and right margin on fixed and responsive containers, because the value auto is applied automatically to margin-left and margin-right property by the Bootstrap at certain breakpoints to align the container horizontally center.

**Bootstrap Grid System**

The Bootstrap grid system is the fastest and easy way to create responsive website layout.

**What is Bootstrap Grid System?**

Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes. It is built with [flexbox](https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php) with mobile-first approach. Also, it is fully responsive and uses twelve column system (12 columns available per row) and six default responsive tiers.

You can use the Bootstrap's predefined grid classes for quickly making the layouts for different types of devices like mobile phones, tablets, laptops, desktops, and so on. For example, you can use the .col-\* classes to create grid columns for extra small devices like mobile phones in portrait mode, and the .col-sm-\* classes for mobile phones in landscape mode.

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

The following table summarizes the key features of the Bootstrap's grid system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features**  **Bootstrap  Grid System** | **X-Small (xs)**  **<576px** | **Small (sm)**  **≥576px** | **Medium (md)**  **≥768px** | **Large (lg)**  **≥992px** | **X-Large (xl)**  **≥1200px** | **XX-Large (xxl)**  **≥1400px** |
| Container max-width | None (auto) | 540px | 720px | 960px | 1140px | 1320px |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- | .col-xxl- |
| Number of columns | 12 | | | | | |
| Gutter width | 1.5rem (.75rem on left and right) | | | | | |
| Custom gutters | Yes | | | | | |
| Nestable | Yes | | | | | |
| Column ordering | Yes | | | | | |

Above table demonstrates one important thing, applying any .col-sm-\* class to an element will not only have an effect on small devices, but also on medium, large and extra large devices (viewport width ≥768px), if a .col-md-\*, .col-lg-\*, .col-xl-\*, or .col-xxl-\* class is not present.

Similarly, the .col-md-\* class will not only have an effect on medium devices, but also on large and extra large devices if a .col-lg-\*, .col-xl-\*, or .col-xxl-\* class is not present.

Now the question arises how to create rows and columns using this 12 column responsive grid system. The answer is pretty simple, at first create a container that acts as a wrapper for your rows and columns using any [container classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-containers.php) such as .container, after that create rows inside the container using the .row class, and to create columns inside any row you can use the .col-\*, .col-sm-\*, .col-md-\*, .col-lg-\*, .col-xl-\* and .col-xxl-\* classes.

The columns are actual content area where we will place our contents. In the following sections we will put all these things into real action and see how it actually works:

**Creating Two Column Layouts**

The following example will show you how to create two column layouts for medium, large and extra large devices like tables, laptops and desktops etc. However, on mobile phones (screen width less than 768px), the columns will automatically become horizontal (2 rows, 1 column).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Two Column Grid Layouts for Tablets and Desktops</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container">

<!--Row with two equal columns-->

<div class="row">

<div class="col-md-6">

<div class="demo-content">.col-md-6</div>

</div>

<div class="col-md-6">

<div class="demo-content bg-alt">.col-md-6</div>

</div>

</div>

<!--Row with two columns divided in 1:2 ratio-->

<div class="row">

<div class="col-md-4">

<div class="demo-content">.col-md-4</div>

</div>

<div class="col-md-8">

<div class="demo-content bg-alt">.col-md-8</div>

</div>

</div>

<!--Row with two columns divided in 1:3 ratio-->

<div class="row">

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-9">

<div class="demo-content bg-alt">.col-md-9</div>

</div>

</div>

</div>

</body>

</html>

**Note:** In a grid layout, content must be placed inside the columns (.col and .col-\*) and only columns may be the immediate children of rows (.row). Also, rows should be placed inside a [container (either fixed or fluid)](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-containers.php) for proper padding and alignment.

**Tip:** Grid column widths are set in percentages, so they're always fluid and sized relative to their parent element. In addition, each column has horizontal padding (called a gutter) for controlling the space between individual columns.

Since the Bootstrap grid system is based on 12 columns, therefore to keep the columns in a one line (i.e. side by side), the sum of the grid column numbers within a single row should not be greater than 12. If you go through the above example code carefully you will find the numbers of grid columns (i.e. col-md-\*) add up to twelve (6+6, 4+8 and 3+9) for every row.

## Creating Three Column Layouts

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Three Column Grid Layouts for Tablets (landscape) and Desktops</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container">

<!--Row with three equal columns-->

<div class="row">

<div class="col-lg-4">

<div class="demo-content">.col-lg-4</div>

</div>

<div class="col-lg-4">

<div class="demo-content bg-alt">.col-lg-4</div>

</div>

<div class="col-lg-4">

<div class="demo-content">.col-lg-4</div>

</div>

</div>

<!--Row with three columns divided in 1:4:1 ratio-->

<div class="row">

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

<div class="col-lg-8">

<div class="demo-content bg-alt">.col-lg-8</div>

</div>

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

</div>

<!--Row with three columns divided unevenly-->

<div class="row">

<div class="col-lg-3">

<div class="demo-content">.col-lg-3</div>

</div>

<div class="col-lg-7">

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

</div>

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

</div>

</div>

</body>

</html>

**Note:** If more than 12 grid columns are placed within a single row, then each group of extra columns, as a whole, will wrap onto a new line. See [column wrapping behavior](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php#column-wrapping).

**Auto-layout Columns**

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

Let's try out the following example to understand how it exactly works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Auto-layout Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<!--Row with two equal columns-->

<div class="row">

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

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

</div>

<!--Row with three equal columns-->

<div class="row">

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

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

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

</div>

</div>

</body>

</html>

Additionally, you can also set the width of one column and let the sibling columns automatically resize around it equally. You may use the predefined grid classes or inline widths.

If you try the following example you'll find columns in a row with class .col has equal width.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Auto-resize Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

[class^=col]{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<!--Row with two equal columns-->

<div class="row">

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

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

</div>

<!--Row with three columns divided in 1:2:1 ratio-->

<div class="row">

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

<div class="col-sm-6">.col-sm-6</div>

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

</div>

</div>

</body>

</html>

## Column Wrapping Behavior

Now we are going to create more flexible layouts that changes the column orientation based on the viewport size. The following example will create a three column layout on large devices like laptops and desktops, as well as on tablets (e.g. Apple iPad) in landscape mode, but on medium devices like tablets in portrait mode (768px ≤ screen width < 992px), it will change into a two column layout where the third column moves at the bottom of the first two columns.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Column Wrapping</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Column Wrapping</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap column wrapping works.</div>

<div class="container">

<div class="row">

<div class="col-md-4 col-lg-3">

<div class="demo-content">.col-md-4 .col-lg-3</div>

</div>

<div class="col-md-8 col-lg-6">

<div class="demo-content bg-alt">.col-md-8 .col-lg-6</div>

</div>

<div class="col-md-12 col-lg-3">

<div class="demo-content">.col-md-12 .col-lg-3</div>

</div>

</div>

</div>

</body>

</html>

As you can see in the example above the sum of the medium grid column numbers (i.e. col-md-\*) is 3 + 9 + 12 = 24 > 12, therefore the third [<div>](https://www.tutorialrepublic.com/html-reference/html-div-tag.php) element with the class .col-md-12 that is adding the extra columns beyond the maximum 12 columns in a .row, gets wrapped onto a new line as one contiguous unit on the medium screen size devices.

Similarly, you can create even more adaptable layouts for your websites using the Bootstrap's grid column wrapping feature. Here're some ready to use [Bootstrap grid examples](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-examples.php).

**Creating Multi-Column Layouts with Bootstrap**

With the new Bootstrap mobile first flexbox grid system you can easily control how your website layout will render on different types of devices that have different screen or viewport sizes like mobile phones, tablets, desktops, etc. Let's consider the following illustration.



In the above illustration there are total 12 content boxes in all devices, but its placement varies according to the device screen size, like in mobile device the layout is rendered as one column grid layout which has 1 column and 12 rows placed above one another, whereas in tablet it is rendered as two column grid layout which has 2 columns and 6 rows.

Further, in large screen size devices like laptops and desktops it is rendered as three column grid layout which has 3 columns and 4 rows and finally in extra large screen devices like large desktops it is rendered as four column grid layout which has 4 columns and 3 rows.

Now the question is how we can create such responsive layouts using this Bootstrap flexbox grid system. Let's start with the primary target device. Suppose our primary target device is laptop or normal desktop. Since our laptop layout has 3 columns and 4 rows i.e. 3x4 grid layout, so the HTML code for making such grid structure would look something like this.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Grid Layouts for Large Devices</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

p{

padding: 50px;

font-size: 32px;

font-weight: bold;

text-align: center;

background: #dbdfe5;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container-lg">

<div class="row">

<div class="col-xl-4"><p>Box 1</p></div>

<div class="col-xl-4"><p>Box 2</p></div>

<div class="col-xl-4"><p>Box 3</p></div>

<div class="col-xl-4"><p>Box 4</p></div>

<div class="col-xl-4"><p>Box 5</p></div>

<div class="col-xl-4"><p>Box 6</p></div>

<div class="col-xl-4"><p>Box 7</p></div>

<div class="col-xl-4"><p>Box 8</p></div>

<div class="col-xl-4"><p>Box 9</p></div>

<div class="col-xl-4"><p>Box 10</p></div>

<div class="col-xl-4"><p>Box 11</p></div>

<div class="col-xl-4"><p>Box 12</p></div>

</div>

</div>

</body>

</html>

**Tip:** The .container-lg class makes the container 100% wide if the width of the viewport is less than 992px, thus utilizing the full available width on smaller screens.

If you see the output of the above example in a large device such as a laptop or desktop which has screen or viewport width greater than or equal to 1200px but less than 1400px, you will find the layout has 4 rows where each row has 3 equal columns resulting in 3x4 grid layout.

Now it's time to customize our layout for other devices. Let's first start by customizing it for medium devices like tablets (768px ≤ viewport width < 1200px). Since on tablet our layout rendered as 2x6 grids (i.e. 2 columns and 6 rows). So, go ahead and add the class .col-md-6 on every column.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Grid Layouts for Large Devices</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

p{

padding: 50px;

font-size: 32px;

font-weight: bold;

text-align: center;

background: #dbdfe5;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container-lg">

<div class="row">

<div class="col-xl-4 col-md-6"><p>Box 1</p></div>

<div class="col-xl-4 col-md-6"><p>Box 2</p></div>

<div class="col-xl-4 col-md-6"><p>Box 3</p></div>

<div class="col-xl-4 col-md-6"><p>Box 4</p></div>

<div class="col-xl-4 col-md-6"><p>Box 5</p></div>

<div class="col-xl-4 col-md-6"><p>Box 6</p></div>

<div class="col-xl-4 col-md-6"><p>Box 7</p></div>

<div class="col-xl-4 col-md-6"><p>Box 8</p></div>

<div class="col-xl-4 col-md-6"><p>Box 9</p></div>

<div class="col-xl-4 col-md-6"><p>Box 10</p></div>

<div class="col-xl-4 col-md-6"><p>Box 11</p></div>

<div class="col-xl-4 col-md-6"><p>Box 12</p></div>

</div>

</div>

</body>

</html>

**Tip:** For convenience choose your primary target device and create layout for that device first after that add classes to make it responsive for other devices.

Similarly, you can customize the layout for extra extra large devices such as a large desktop screen by adding the class .col-xxl-3 on each column, as every row in that layout contains 4 columns (i.e. 4x3 grids layout). Here's the final code after combining the whole process.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Grid Layouts for All Devices</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

p{

padding: 50px;

font-size: 32px;

font-weight: bold;

text-align: center;

background: #dbdfe5;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container-lg">

<div class="row">

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 1</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 2</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 3</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 4</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 5</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 6</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 7</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 8</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 9</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 10</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 11</p></div>

<div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 12</p></div>

</div>

</div>

</body>

</html>

**Tip:** According to the above illustration there is no need to customize the layout for mobile phones; since columns on extra small devices will automatically become horizontal and rendered as 1x12 column grid layout in absence of .col-\* or .col-sm-\* classes.

**Nesting of Grid Columns**

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nested Rows and Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.main-content{

min-height: 230px;

background: #dbdfe5;

}

.sidebar-content{

min-height: 100px;

margin-bottom: 30px;

background: #b4bac0;

}

</style>

</head>

<body>

<div class="container mt-3">

<div class="row">

<div class="col-sm-8">

<div class="main-content"></div>

</div>

<div class="col-sm-4">

<!--Nested rows within a column-->

<div class="row">

<div class="col-12">

<div class="sidebar-content"></div>

</div>

</div>

<div class="row">

<div class="col-6">

<div class="sidebar-content"></div>

</div>

<div class="col-6">

<div class="sidebar-content"></div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

## Creating Variable Width Columns

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Sizing Columns Based on the Width of their Content</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

[class^=col]{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row justify-content-md-center">

<div class="col-md-3">Column left</div>

<div class="col-md-auto">Variable width column</div>

<div class="col-md-3">Column right</div>

</div>

<div class="row">

<div class="col">Column left</div>

<div class="col-auto">Variable width column</div>

<div class="col">Column right</div>

</div>

</div>

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

</body>

</html>

## Alignment of Grid Columns

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

### Vertical Alignment of Grid Columns

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Vertical Alignments of Grid Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

min-height: 10rem;

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row align-items-start">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

<div class="row align-items-center">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

<div class="row align-items-end">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

</div>

</body>

</html>

Individual columns inside a row can also be aligned vertically. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Vertical Alignments of Individual Grid Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

min-height: 10rem;

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row">

<div class="col align-self-start">Column one</div>

<div class="col align-self-center">Column two</div>

<div class="col align-self-end">Column three</div>

</div>

</div>

</body>

</html>

**Note:** You can skip the number in .col-\* grid class and just use the .col class to create equal size columns for all devices (extra small, small, medium, large, and extra large).

### Horizontal Alignment of Grid Columns

You can use the classes .justify-content-start, .justify-content-center, and .justify-content-end to align the grid columns horizontally at the left, center and right of a container, respectively. Let's check out the following example to see how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Horizontal Alignments of Grid Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

[class^=col-]{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row justify-content-start">

<div class="col-4">Column one</div>

<div class="col-4">Column two</div>

</div>

<div class="row justify-content-center">

<div class="col-4">Column one</div>

<div class="col-4">Column two</div>

</div>

<div class="row justify-content-end">

<div class="col-4">Column one</div>

<div class="col-4">Column two</div>

</div>

</div>

</body>

</html>

Alternatively, you can use the class .justify-content-around to distribute grid columns evenly with half-size spaces on either end, whereas you can use the class .justify-content-between to distribute the grid columns evenly where the first column placed at the start and the last column placed at the end. Try out the following example to see how it actually works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Justify Grid Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

[class^=col-]{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row justify-content-around">

<div class="col-4">Column one</div>

<div class="col-4">Column two</div>

</div>

<div class="row justify-content-between">

<div class="col-4">Column one</div>

<div class="col-4">Column two</div>

</div>

</div>

</body>

</html>

Please check out the tutoiral on [css3 flexbox](https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php) to learn more about flex items alignment.

**Reordering of Grid Columns**

You can even change the visual order of your grid columns without changing their order in actual markup. Use the class .order-last to order the column in last, whereas use the class .order-first to order the column at first place. Let's checkout an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Reorder Grid Columns in Bootstrap</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row">

<div class="col order-last">First, but ordered at last</div>

<div class="col">Second, but unordered</div>

<div class="col order-first">Last, but ordered at first</div>

</div>

</div>

</body>

</html>

You can also use the .order-\* classes to order the grid columns depending on the order numbers. Grid column with higher order number comes after the grid column with lower order number or grid column with no order classes. It includes support for 1 through 12 across all five grid tiers.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Order Grid Columns Using Order Number in Bootstrap</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row">

<div class="col order-4">First, but ordered at last</div>

<div class="col">Second, but ordered at first</div>

<div class="col order-1">Last, but ordered at second</div>

</div>

</div>

</body>

</html>

## Offsetting the Grid Columns

You can also move grid columns to the right for alignment purpose using the column offset classes like .offset-sm-\*, .offset-md-\*, .offset-lg-\*, and so on.

These classes offset the columns by simply increasing its left margin by specified number of columns. For example, the class .offset-md-4 on column .col-md-8 moves it to the right over four columns from its original position. Try out the following example to see how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Columns Offsetting</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 10px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<div class="container mt-3">

<div class="row">

<div class="col-md-4">

<div class="demo-content">.col-md-4</div>

</div>

<div class="col-md-8">

<div class="demo-content bg-alt">.col-md-8</div>

</div>

</div>

<div class="row">

<div class="col-md-8 offset-md-4">

<div class="demo-content bg-alt">.col-md-8 .offset-md-4</div>

</div>

</div>

<hr>

<div class="row">

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-9">

<div class="demo-content bg-alt">.col-md-9</div>

</div>

</div>

<div class="row">

<div class="col-md-6 offset-md-3">

<div class="demo-content bg-alt">.col-md-6 .offset-md-3</div>

</div>

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

</div>

<hr>

<div class="row">

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-3">

<div class="demo-content bg-alt">.col-md-3</div>

</div>

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-3">

<div class="demo-content bg-alt">.col-md-3</div>

</div>

</div>

<div class="row">

<div class="col-md-3 offset-md-3">

<div class="demo-content bg-alt">.col-md-3 .offset-md-3</div>

</div>

<div class="col-md-3 offset-md-3">

<div class="demo-content">.col-md-3 .offset-md-3</div>

</div>

</div>

</div>

</body>

</html>

You can also offset columns using the margin utility classes. These classes are useful in the situations where the width of the offset is not fixed. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Columns Offsetting Using Margin Utilities</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 10px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<div class="container mt-3">

<div class="row">

<div class="col-md-4">

<div class="demo-content">.col-md-4</div>

</div>

<div class="col-md-4 ms-auto">

<div class="demo-content bg-alt">.col-md-4 .ms-auto</div>

</div>

</div>

<div class="row">

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-3 ms-md-auto">

<div class="demo-content bg-alt">.col-md-3 .ms-md-auto</div>

</div>

</div>

<div class="row">

<div class="col-auto me-auto">

<div class="demo-content">.col-auto .me-auto</div>

</div>

<div class="col-auto">

<div class="demo-content bg-alt">.col-auto</div>

</div>

</div>

</div>

</body>

</html>

**ote:** You can use the class .col-auto to create columns that only take up as much space as needed, i.e. the column sizes itself based on the contents.

**Creating Compact Columns**

You can remove the default gutters between columns to create compact layouts by adding the class .g-0 on .row. This class removes the negative margins from row and the horizontal padding from all immediate children columns. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Compact Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

min-height: 50px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center">Columns with Gutters</h2>

<div class="container mt-3">

<div class="row">

<div class="col-4">

<div class="demo-content">.col-4</div>

</div>

<div class="col-4">

<div class="demo-content bg-alt">.col-4</div>

</div>

<div class="col-4">

<div class="demo-content">.col-4</div>

</div>

</div>

</div>

<hr>

<h2 class="text-center">Columns without Gutters</h2>

<div class="container mt-3">

<div class="row g-0">

<div class="col-4">

<div class="demo-content">.col-4</div>

</div>

<div class="col-4">

<div class="demo-content bg-alt">.col-4</div>

</div>

<div class="col-4">

<div class="demo-content">.col-4</div>

</div>

</div>

</div>

</body>

</html>

## Breaking Columns to a New Line

You can also create equal-width columns that span multiple rows by inserting a <div> with .w-100 class where you want the columns to break to a new line. Additionally, you can make these breaks responsive by combining the .w-100 class with [responsive display utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Breaking Columns to a New Lines</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

[class^=col]{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<h4 class="mt-3">Breaking columns on all devices</h4>

<div class="row">

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

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

<div class="w-100"></div>

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

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

</div>

<h4 class="mt-3">Breaking columns on all devices except extra large devices</h4>

<div class="row">

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

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

<div class="w-100 d-xl-none"></div>

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

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

</div>

<p class="mt-3"><strong>Note:</strong> Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how it works.</p>

</div>

</body>

</html>

We hope you've understood the basics of new Bootstrap 5 grid system. In next few chapters you'll learn how to create basic web page layouts using this flexbox grid system.

# Bootstrap Fixed Layout

## Creating Fixed Layout with Bootstrap

With Bootstrap you can still create web page layouts based on fixed number of pixels, however the container width vary depending on the viewport width and the layout is responsive too.

The process of creating the fixed yet responsive layout basically starts with the .container class. After that you can create rows with the .row class to wrap the horizontal groups of columns. Rows must be placed within a .container for proper alignment and padding.

Further columns can be created inside a row using the predefined grid classes such as .col, col-{xs|sm|md|lg|xl|xxl}-\*, where \* represent grid number and should be from 1 to 12. Please check out the tutorial on [Bootstrap grid system](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php) to learn more about grid classes..

**Note:** Actual content like text, images, videos, tables, etc. should be placed within columns, and only columns may be the immediate children of rows.

The following example will create a fixed width responsive layout that is 720px pixels wide on medium devices like tablets (viewport ≥ 768px), whereas 960px wide on large devices like small laptops (viewport ≥ 992px), 1140px wide on extra large devices like desktops (viewport ≥ 1200px), and 1320px wide on extra-extra large devices like large desktops (viewport ≥ 1400px).

However, on small devices such as mobile phones (576px ≤ viewport < 768px) the layout will be 540px wide. But, on extra-small devices (viewport < 576px) the layout will cover 100% width. Also, columns will be stacked vertically and navbar will be collapsed in both cases.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Fixed Layout Example</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

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

</div>

</div>

</div>

</nav>

<div class="container">

<div class="p-5 my-4 bg-light rounded-3">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" class="text-success" target="\_blank">tutorialrepublic.com</a> you will learn the essential web development technologies along with real life practice examples, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-success btn-lg">Get started today</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>HTML</h2>

<p>HTML is the standard markup language for describing the structure of the web pages. Our HTML tutorials will help you to understand the basics of latest HTML5 language, so that you can create your own web pages or website.</p>

<p><a href="https://www.tutorialrepublic.com/html-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>CSS</h2>

<p>CSS is used for describing the presentation of web pages. CSS can save a lot of time and effort. Our CSS tutorials will help you to learn the essentials of latest CSS3, so that you can control the style and layout of your website.</p>

<p><a href="https://www.tutorialrepublic.com/css-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Bootstrap</h2>

<p>Bootstrap is a powerful front-end framework for faster and easier web development. Our Bootstrap tutorials will help you to learn all the features of latest Bootstrap 4 framework so that you can easily create responsive websites.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

</div>

<hr>

<footer>

<div class="row">

<div class="col-md-6">

<p>Copyright &copy; 2021 Tutorial Republic</p>

</div>

<div class="col-md-6 text-md-end">

<a href="#" class="text-dark">Terms of Use</a>

<span class="text-muted mx-2">|</span>

<a href="#" class="text-dark">Privacy Policy</a>

</div>

</div>

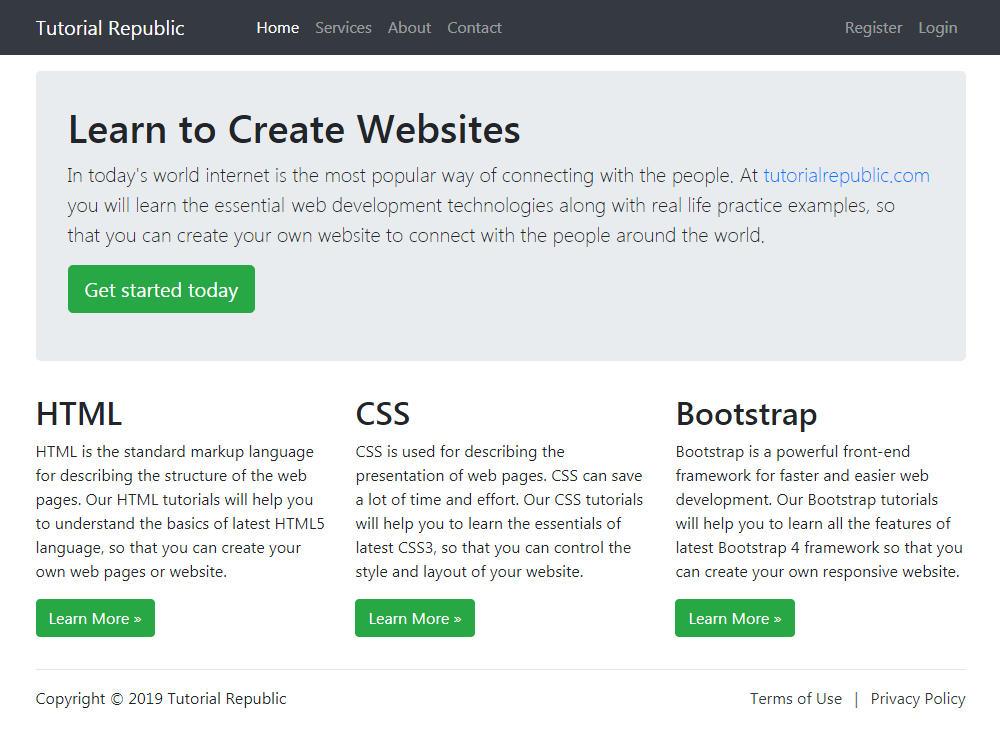
</footer>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=fixed-layout)

We've used the margin utility classes such as .mb-3, .ml-auto, mx-2 etc. to adjust spacing between the elements. Whereas the classes .text-dark, .text-muted, .text-md-right are text utility classes to adjust color and alignment of text. You'll learn about them in later chapters.

# Bootstrap Fluid Layout

In this tutorial you will learn how to create fluid layouts with Bootstrap.

## Creating Fluid Layout with Bootstrap

In Bootstrap you can use the class .container-fluid to create fluid layouts to utilize the 100% width of the viewport across all devices (extra small, small, medium, large, extra large, and extra-extra large).

The class .container-fluid simply applies the width: 100% instead of different width for different viewport sizes. However, the layout will still responsive and you can use the grid classes as usual. See the tutorial on [Bootstrap grid system](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php) to learn more about grid classes.

The following example will create a fluid layout that covers 100% width of the screen.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Fluid Layout</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

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

</div>

</div>

</div>

</nav>

<div class="container-fluid">

<div class="p-5 my-4 bg-light rounded-3">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" class="text-success" target="\_blank">tutorialrepublic.com</a> you will learn the essential web development technologies along with real life practice examples, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-success btn-lg">Get started today</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>HTML</h2>

<p>HTML is the standard markup language for describing the structure of the web pages. Our HTML tutorials will help you to understand the basics of latest HTML5 language, so that you can create your own web pages or website.</p>

<p><a href="https://www.tutorialrepublic.com/html-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>CSS</h2>

<p>CSS is used for describing the presentation of web pages. CSS can save a lot of time and effort. Our CSS tutorials will help you to learn the essentials of latest CSS3, so that you can control the style and layout of your website.</p>

<p><a href="https://www.tutorialrepublic.com/css-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Bootstrap</h2>

<p>Bootstrap is a powerful front-end framework for faster and easier web development. Our Bootstrap tutorials will help you to learn all the features of latest Bootstrap 4 framework so that you can easily create responsive websites.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

</div>

<hr>

<footer>

<div class="row">

<div class="col-md-6">

<p>Copyright &copy; 2021 Tutorial Republic</p>

</div>

<div class="col-md-6 text-md-end">

<a href="#" class="text-dark">Terms of Use</a>

<span class="text-muted mx-2">|</span>

<a href="#" class="text-dark">Privacy Policy</a>

</div>

</div>

</footer>

</div>

</body>

</html>

**Bootstrap Respo nsive Layout**

In this tutorial you will learn how to create responsive websites with Bootstrap framework.

**What is Responsive Web Design**

Responsive web design is a process of designing and building websites to provide better accessibility and optimal viewing experience to the user by optimizing it for different devices.

With the growing trend of smart phones and tablets, it has become almost unavoidable to ignore the optimization of sites for mobile devices. Responsive web design is a preferable alternative and an efficient way to target a wide range of devices with much less efforts.

Responsive layouts automatically adjust and adapts to any device screen size, whether it is a desktop, a laptop, a tablet, or a mobile phone. See the following Illustration.



**Creating Responsive Layout with Bootstrap**

With the Bootstrap powerful mobile first [flexbox](https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php) grid system creating the responsive and mobile friendly websites and applications has become much easier.

Bootstrap is responsive and mobile friendly from the start. Its [six tier grid classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php) provides better control over the layout as well as how it will be rendered on different types of devices like mobile phones, tablets, laptops and desktops, large screen devices, and so on.

The following example will create a responsive layout that is rendered as 4 column layout in extra-large devices (viewport ≥ 1200px), and 3 column layout in large devices (992px ≤ viewport < 1200px), whereas 2 column layout in medium devices (768px ≤ viewport < 992px), and 1 column layout in small and extra-small devices (viewport < 768px). Let's try it out and see how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap 4 Responsive Layout</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

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

</div>

</div>

</div>

</nav>

<div class="container">

<div class="p-5 my-4 bg-light rounded-3">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" class="text-success" target="\_blank">tutorialrepublic.com</a> you will learn the essential web development technologies along with real life practice examples, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-success btn-lg">Get started today</a></p>

</div>

<div class="row g-3">

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>HTML</h2>

<p>HTML is the standard markup language for describing the structure of the web pages. Our HTML tutorials will help you to understand the basics of latest HTML5 language, so that you can create your own website.</p>

<p><a href="https://www.tutorialrepublic.com/html-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>CSS</h2>

<p>CSS is used for describing the presentation of web pages. CSS can save a lot of time and effort. Our CSS tutorials will help you to learn the essentials of latest CSS3, so that you can control the style and layout of your website.</p>

<p><a href="https://www.tutorialrepublic.com/css-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>JavaScript</h2>

<p>JavaScript is the most popular and widely used client-side scripting language. Our JavaScript tutorials will provide in-depth knowledge of the JavaScript including ES6 features, so that you can create interactive websites.</p>

<p><a href="https://www.tutorialrepublic.com/javascript-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>Bootstrap</h2>

<p>Bootstrap is a powerful front-end framework for faster and easier web development. Our Bootstrap tutorials will help you to learn all the features of latest Bootstrap 4 framework so that you can easily create responsive websites.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>PHP</h2>

<p>PHP is the most popular server-side scripting language for creating dynamic web pages. Our PHP tutorials will help you to learn all the features of latest PHP7 scripting language so that you can easily create dynamic websites.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>SQL</h2>

<p>SQL is a standard language designed for managing data in relational database management system. Our SQL tutorials will help you to learn the fundamentals of the SQL language so that you can efficiently manage your databases.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>References</h2>

<p>Our references section outlines all the standard HTML5 tags and CSS3 properties along with other useful references such as color names and values, character entities, web safe fonts, language codes, HTTP messages, and more.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

<div class="col-md-6 col-lg-4 col-xl-3">

<h2>FAQ</h2>

<p>Our Frequently Asked Questions (FAQ) section is an extensive collection of FAQs that provides quick and working solution of common questions and queries related to web design and development with live demo.</p>

<p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="\_blank" class="btn btn-success">Learn More &raquo;</a></p>

</div>

</div>

<hr>

<footer>

<div class="row">

<div class="col-md-6">

<p>Copyright © 2021 Tutorial Republic</p>

</div>

<div class="col-md-6 text-md-end">

<a href="#" class="text-dark">Terms of Use</a>

<span class="text-muted mx-2">|</span>

<a href="#" class="text-dark">Privacy Policy</a>

</div>

</div>

</footer>

</div>

</body>

</html>

**Tip:** Open the output of this example in new blank window (click the New Window link inside the CodeLab editor) and resize the screen you will see the orientation of content boxes changes when viewport width crossing or approaching the certain limit (i.e. [breakpoints](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php#grid-features)).

# Bootstrap Typography

In this tutorial you will learn about the styling and formatting of text content like headings, paragraphs, blockquotes, and so on with Bootstrap.

## Working with Headings

You can define all [HTML headings](https://www.tutorialrepublic.com/html-tutorial/html-headings.php), <h1> through <h6> — In the same way you define in simple HTML document. You can also utilize the heading classes .h1 through .h6 on other elements, if you want to apply the style on element's text same as headings.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Headings</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<h1>h1. Bootstrap heading</h1>

<h2>h2. Bootstrap heading</h2>

<h3>h3. Bootstrap heading</h3>

<h4>h4. Bootstrap heading</h4>

<h5>h5. Bootstrap heading</h5>

<h6>h6. Bootstrap heading</h6>

</div>

</body>

</html>

**Customizing Headings**

Additionally, you can use the [<small>](https://www.tutorialrepublic.com/html-reference/html-small-tag.php) tag with .text-muted class to display the secondary text of any heading in a smaller and lighter variation. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Headings with Secondary Text</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<h2>

Fancy display heading

<small class="text-muted">With faded secondary text</small>

</h2>

</div>

</body>

</html>

## Display Headings

Bootstrap also provides display headings that can be used when you need a heading to stand out. Display headings are displayed in larger [font-size](https://www.tutorialrepublic.com/css-reference/css-font-size-property.php) but lighter [font-weight](https://www.tutorialrepublic.com/css-reference/css-font-weight-property.php).

Six different display headings are available. Here's is an example:

#### Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Display Headings</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<h1 class="display-1">Display Heading 1</h1>

<h1 class="display-2">Display Heading 2</h1>

<h1 class="display-3">Display Heading 3</h1>

<h1 class="display-4">Display Heading 4</h1>

<h1 class="display-5">Display Heading 5</h1>

<h1 class="display-6">Display Heading 6</h1>

</div>

</body>

</html>

**Working with Paragraphs**

Bootstrap's global default [font-size](https://www.tutorialrepublic.com/css-reference/css-font-size-property.php) is **1rem** (typically 16px), with a [line-height](https://www.tutorialrepublic.com/css-reference/css-line-height-property.php) of **1.5** (typically 24px), which is applied to the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element as well as all the paragraphs i.e. the [<p>](https://www.tutorialrepublic.com/html-reference/html-p-tag.php) elements. In addition to that [margin-bottom](https://www.tutorialrepublic.com/css-reference/css-margin-bottom-property.php) of **1rem** is also applied to all the paragraphs.

You can also make a paragraph stand out by adding the class .lead on it.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Paragraphs Styling</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p>This is how a normal paragraph looks like in Bootstrap.</p>

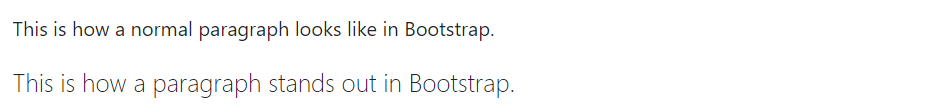
<p class="lead">This is how a paragraph stands out in Bootstrap.</p>

</div>

</body>

</html>

The HTML code in the above examples will produce the following result:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=paragraphs)

**Tip:** In CSS rem stands for "root em". 1rem is equal to the font size of the root element (i.e. the [<html>](https://www.tutorialrepublic.com/html-reference/html-html-tag.php) element), which is 16px in most browsers by default.

**Text Alignment**

You can easily align text to left, right, and center using the text alignment classes.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p class="text-start">Left aligned text on all viewport sizes.</p>

<p class="text-center">Center aligned text on all viewport sizes.</p>

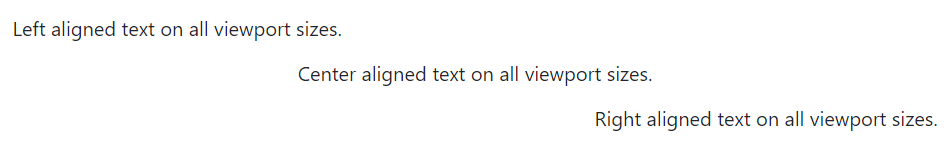
<p class="text-end">Right aligned text on all viewport sizes.</p>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-alignment)

You can also align text based on screen size using the responsive text alignment classes. These classes use the same viewport width breakpoints as the grid system.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Responsive Text Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p class="text-sm-center">Text will be center aligned on small sized (sm) viewports and up.</p>

<p class="text-md-center">Text will be center aligned on medium sized (md) viewports and up.</p>

<p class="text-lg-center">Text will be center aligned on large sized (lg) viewports and up.</p>

<p class="text-xl-center">Text will be center aligned on extra-large sized (xl) viewports and up.</p>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text Formatting</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p><b>This is bold text</b></p>

<p><code>This is computer code</code></p>

<p><em>This is emphasized text</em></p>

<p><i>This is italic text</i></p>

<p><mark>This is highlighted text</mark></p>

<p><small>This is small text</small></p>

<p><strong>This is strongly emphasized text</strong></p>

<p>This is <sub>subscript</sub> and <sup>superscript</sup></p>

<p><ins>This text is inserted to the document</ins></p>

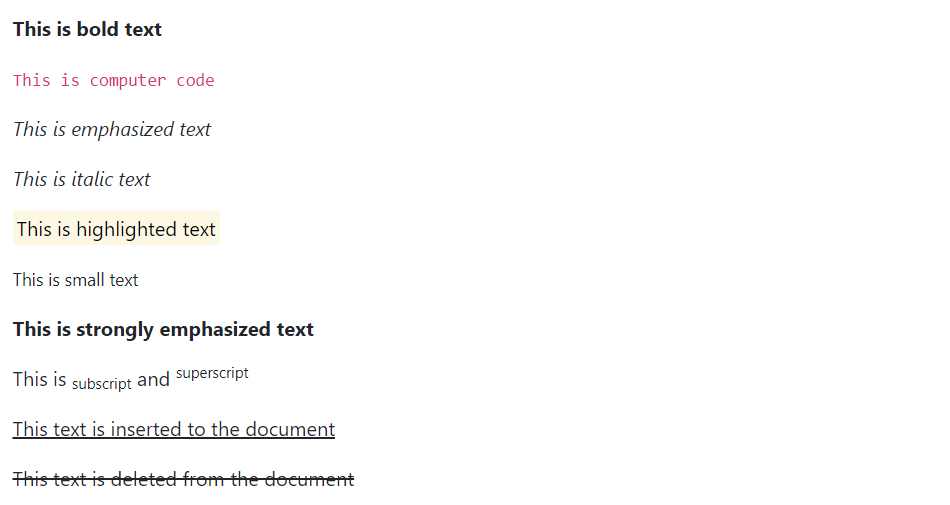
<p><del>This text is deleted from the document</del></p>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-formatting)

**Text Transformation**

You can also transform the text to lowercase, uppercase or make them capitalize.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text Transformation</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p class="text-lowercase">The quick brown fox jumps over the lazy dog.</p>

<p class="text-uppercase">The quick brown fox jumps over the lazy dog.</p>

<p class="text-capitalize">The quick brown fox jumps over the lazy dog.</p>

</div>

</body>

</html>

**Text Coloring**

Colors are the powerful method of conveying important information in website design.

Bootstrap has handful of emphasis utility classes that can be used for this purpose such as showing success message in green color, warning or error message in red color, etc.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text Emphasis Classes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<p class="text-primary">Primary: Please read the instructions carefully before proceeding.</p>

<p class="text-secondary">Secondary: This is featured has been removed from the latest version.</p>

<p class="text-success">Success: Your message has been sent successfully.</p>

<p class="text-info">Info: You must agree with the terms and conditions to complete the sign up process.</p>

<p class="text-warning">Warning: There was a problem with your network connection.</p>

<p class="text-danger">Danger: An error has been occurred while submitting your data.</p>

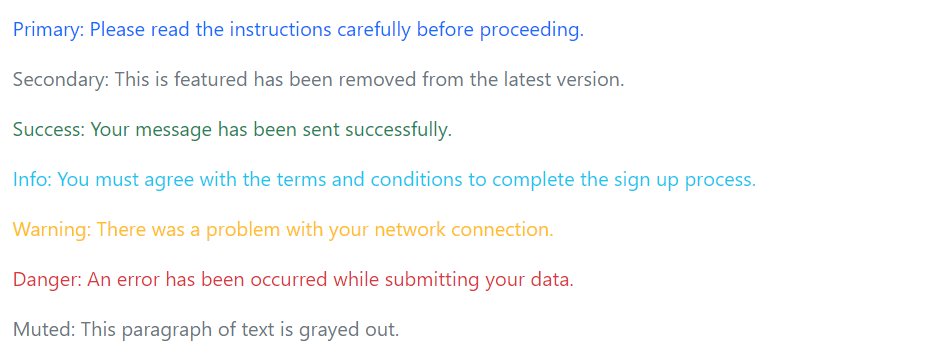
<p class="text-muted">Muted: This paragraph of text is grayed out.</p>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-emphasis-classes)

Please, check out the [Bootstrap helper classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php) chapter to learn about other text coloring and background coloring classes, as well as various other utility classes.

**Styling Blockquotes**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Styling Blockquotes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<blockquote class="blockquote">

<p>Imagination is more important than knowledge.</p>

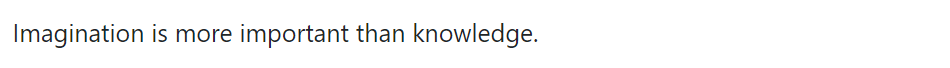
</blockquote>

</div>

</body>

</html>

he output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=blockquotes)

When providing attribution, wrap your <blockquote> in a <figure> element and use a <figcaption> or a block level element (e.g., <p>) with the .blockquote-footer class, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Styling Blockquote with Attribution</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<figure>

<blockquote class="blockquote">

<p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>

</blockquote>

<figcaption class="blockquote-footer">by <cite>Albert Einstein</cite></figcaption>

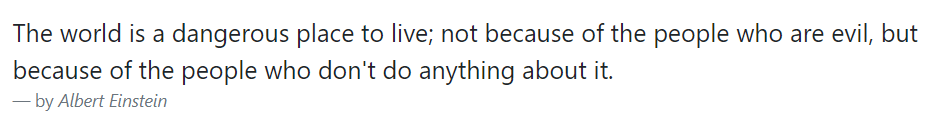
</figure>

</div>

</body>

</html>

The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=blockquote-with-attribution)

You can also align blockquotes to the right or center by simply applying the text alignment classes .text-end or .text-center on the <blockquote> or <figure> element.

**Truncating Long Text**

For longer text, you can use the class .text-truncate to truncate the text with an ellipsis. The display property value of the element must be inline-block or block.

It is particularly helpful in a situation where you want to display a piece of text in a single line but there is no enough space available. Let's try out an example and see how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text Truncation</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<!-- Block level element -->

<div class="row">

<div class="col-2 text-truncate">

The quick brown fox jumps over the lazy dog.

</div>

</div>

<!-- Inline level element -->

<span class="d-inline-block text-truncate" style="max-width: 100px;">

The quick brown fox jumps over the lazy dog.

</span>

</div>

</body>

</html>

## Text wrapping and Overflow

You can use the class .text-wrap to wrap the text within an element by overwriting its [white-space](https://www.tutorialrepublic.com/css-reference/css-white-space-property.php) property if it is set to pre or nowrap, such as [Bootstrap badge components](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-badges.php).

Similarly, you can use the class .text-nowrap to prevent text from wrapping within an element.

Let's try out the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Text wrapping and Overflow</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="badge bg-primary text-wrap" style="width: 6rem;">

This text will wrap.

</div>

<div class="bg-warning text-nowrap mt-4" style="width: 6rem;">

This text will overflow the element's box.

</div>

</div>

</body>

</html>

**Wrapping Long Word**

You can use the class .text-break to prevent long word from breaking your layout.

Let's try out the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Word Break</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="container">

<h2>Without Word Break</h2>

<div class="row">

<div class="col-2">

<p class="bg-warning">veryveryveryveryveryveryverylongword</p>

</div>

</div>

<hr />

<h2>With Word Break</h2>

<div class="row">

<div class="col-2">

<p class="bg-warning text-break">veryveryveryveryveryveryverylongword</p>

</div>

</div>

</div>

</div>

</body>

</html>

**Bootstrap Tables**

## What is Table?

The HTML tables are used to present data in grid manner like row and columns. Using Bootstrap you can greatly improve the appearance of table in a quick and easy way.

See the tutorial on [HTML tables](https://www.tutorialrepublic.com/html-tutorial/html-tables.php) to learn more about tables.

## Creating a Simple Table with Bootstrap

You can create tables with basic styling that has horizontal dividers and small cell padding (8px by default), by just adding the Bootstrap's class .table to the [<table>](https://www.tutorialrepublic.com/html-reference/html-table-tag.php) element.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Simple Table</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

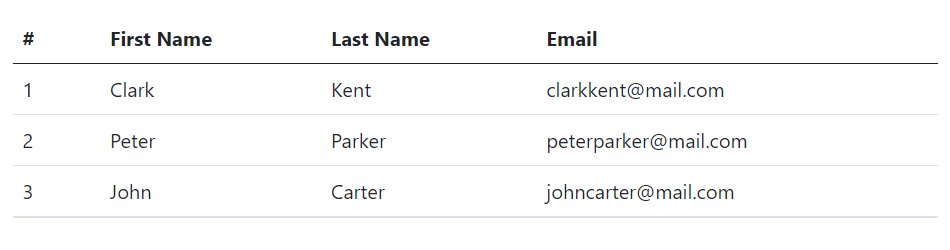
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=simple-table)

**Creating Accented Tables**

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

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Dark Table</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

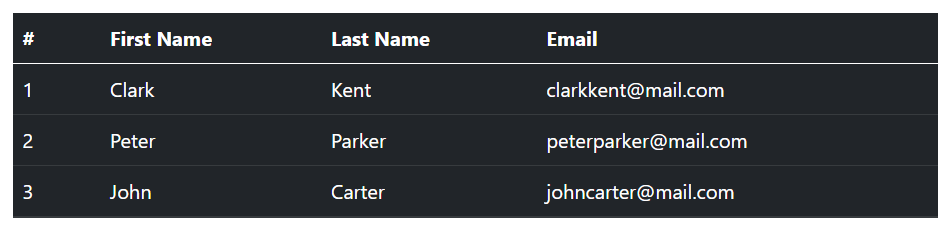
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dark-table)

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Accented Tables</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

**<table class="table table-success">**

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

**<table class="table table-danger">**

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

**<table class="table table-warning">**

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

**<table class="table table-info">**

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

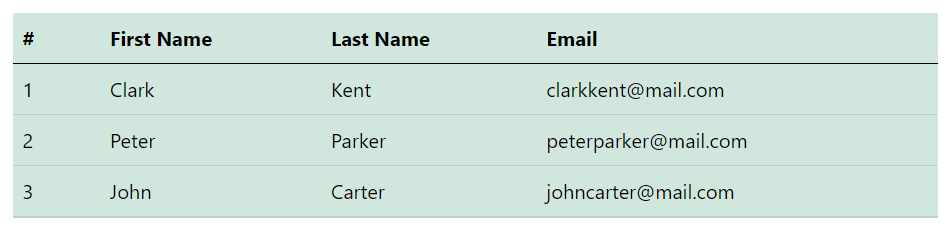
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=accented-table)

Check out the snippets section for examples of some [beautifully designed Bootstrap tables](https://www.tutorialrepublic.com/snippets/gallery.php?tag=table).

**Tip:** You can use these contextual classes on the .table base class to create colored version of any table such as stripped, hoverable, bordered, compact table, and so on.

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Table with Accented Rows</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table">

<thead>

<tr>

<th>#</th>

<th>Bill</th>

<th>Payment Date</th>

<th>Payment Status</th>

</tr>

</thead>

<tbody>

<tr class="table-primary">

<td>1</td>

<td>Credit Card</td>

<td>04/07/2021</td>

<td>Waiting for statement</td>

</tr>

<tr class="table-secondary">

<td>2</td>

<td>Insurance</td>

<td>02/07/2021</td>

<td>Cancelled</td>

</tr>

<tr class="table-success">

<td>3</td>

<td>Water</td>

<td>01/07/2021</td>

<td>Paid</td>

</tr>

<tr class="table-info">

<td>4</td>

<td>Internet</td>

<td>05/07/2021</td>

<td>Change plan</td>

</tr>

<tr class="table-warning">

<td>5</td>

<td>Electricity</td>

<td>03/07/2021</td>

<td>Pending</td>

</tr>

<tr class="table-danger">

<td>6</td>

<td>Telephone</td>

<td>06/07/2021</td>

<td>Due</td>

</tr>

<tr class="table-light">

<td>7</td>

<td>Car Service</td>

<td>08/07/2021</td>

<td>Call in to confirm</td>

</tr>

<tr class="table-dark">

<td>8</td>

<td>Gas</td>

<td>06/07/2021</td>

<td>Payment failed</td>

</tr>

</tbody>

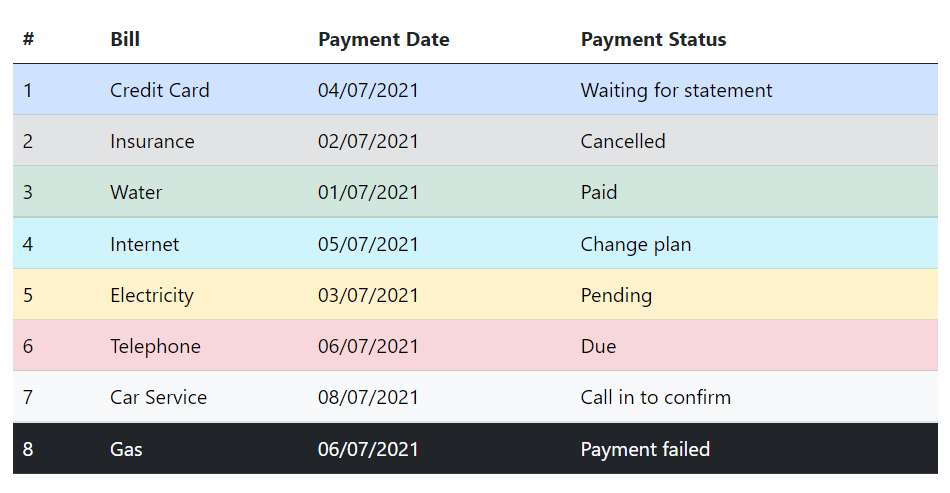
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-with-accented-rows)

**Creating Tables with Striped Rows**

You can also add zebra-striping to the table rows within the [<tbody>](https://www.tutorialrepublic.com/html-reference/html-tbody-tag.php) by simply adding an additional class .table-striped to the .table base class, as shown below:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Zebra-Striped Tables</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table table-striped">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

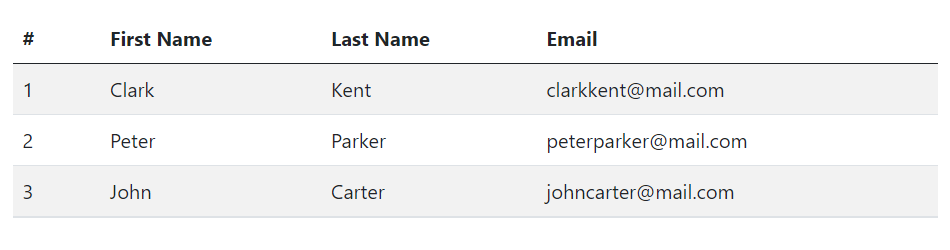
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-striped)

**Creating Bordered Tables**

You can add borders on all sides of the table and cells by adding the modifier class .table-bordered to the .table base class, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Bordered Tables</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table table-bordered">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

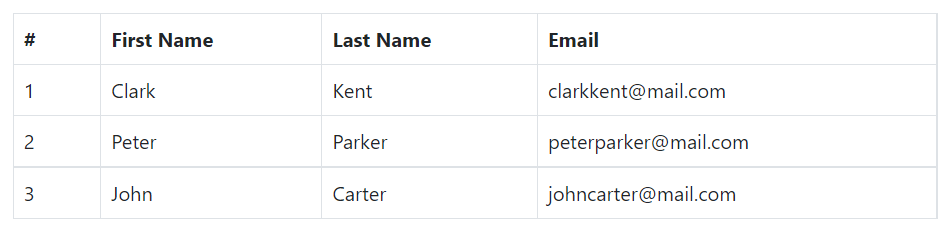
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-bordered)

**Creating Borderless Tables**

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Borderless Tables</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table table-borderless">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

<table class="table table-borderless table-success">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

<table class="table table-borderless table-danger">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

<table class="table table-borderless table-warning">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

<table class="table table-borderless table-info">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

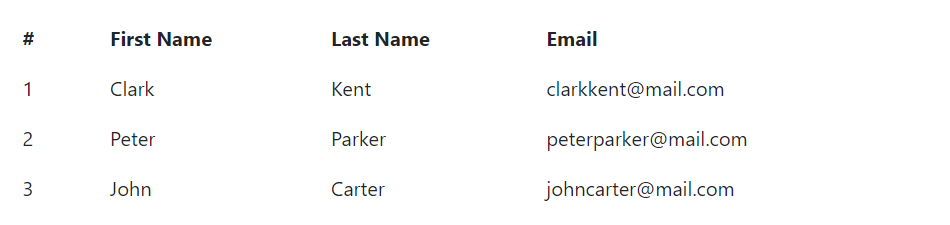
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=borderless-table)

**Enabling Hover State on Table Rows**

You can also enable a hover state on table rows within a [<tbody>](https://www.tutorialrepublic.com/html-reference/html-tbody-tag.php) element by adding the modifier class .table-hover to the .table base class. Let's try out the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Tables with Hover States</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table table-hover">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

</div>

</body>

</html>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

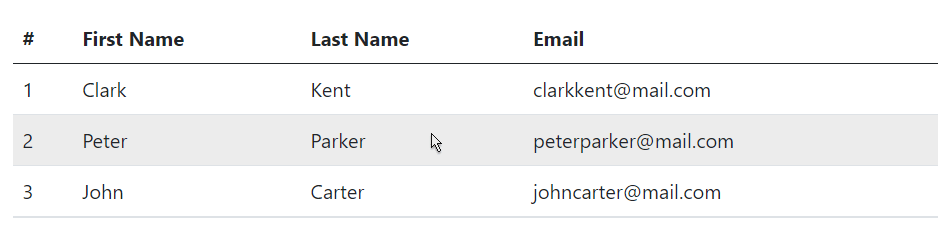
<td>johncarter@mail.com</td>

</tr>

</tbody>

</table>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-with-hover-states)

## Creating Small or Compact Tables

You can also make your tables more compact and save the space through adding the modifier class .table-sm to the .table base class. The .table-sm class makes the table compact by cutting all cell padding in half. Let's take a look at the following example:

#### Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Compact Tables</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table table-sm">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

</table>

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

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

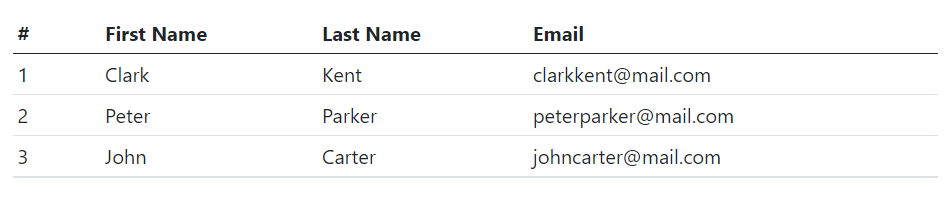
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=compact-table)

**Setting Table Head Colors**

Similar to light and dark tables, you can use the modifier classes .table-light or .table-dark on the <thead> element to make it appear in light or dark gray.

The following example will create a table with light gray background head.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Table Head Light</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table">

<thead class="table-light">

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

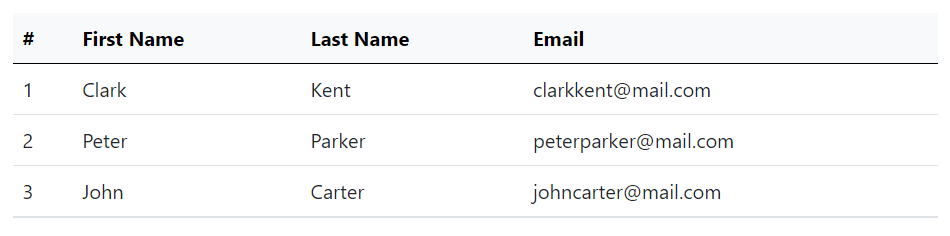
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-head-light)

The following example will create a table with dark gray background head.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Table Head Dark</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<table class="table">

<thead class="table-dark">

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

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

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

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

</tr>

</tbody>

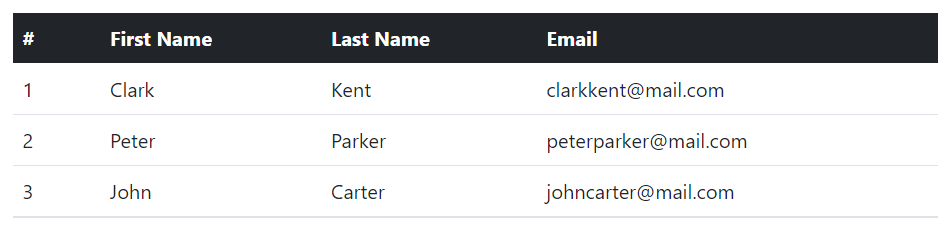
</table>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=table-head-dark)

## Creating Responsive Tables with Bootstrap

You can also create responsive tables to enable horizontal scrolling on small devices.

To make any table responsive just place it inside a [<div>](https://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the .table-responsive class on it. You can also specify when the table should have a scrollbar, based on the viewport width (i.e. [breakpoints](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php#grid-features)), using the classes .table-responsive{-sm|-md|-lg|-xl}.

Let's try out the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Responsive Table</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Custom style to make this example easy to understand \*/

table \* {

white-space: nowrap;

}

</style>

</head>

<body>

<div class="m-4">

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

<th>Biography</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

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

<td>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</td>

</tr>

<tr>

<td>2</td>

<td>John</td>

<td>Carter</td>

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

<td>Vestibulum consectetur scelerisque bibendum scelerisque purus.</td>

</tr>

<tr>

<td>3</td>

<td>Peter</td>

<td>Parker</td>

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

<td>Integer pulvinar leo id risus interdum vel metus dignissim.</td>

</tr>

</tbody>

</table>

</div>

<p class="mt-4"><strong>Note:</strong> Change the editor layout/orientation to see how responsive table works.</p>

</div>

</body>

</html>

**Tip:** Text inside the cells of <thead> are always vertical aligned to the bottom. Whereas, text inside the cells of <tbody> inherit their alignment from <table> and are vertically aligned to the top by default. Use the [vertical align classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#vertical-alignment-classes) to re-align text where needed.

**Bootstrap Cards**

In this tutorial you will learn how to use Bootstrap card component.

**Using the Bootstrap Cards**

Bootstrap card is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. Card replaces panel, well, and thumbnail components in old Bootstrap 3 version.

In the following sections, you will see what you can do with the card component.

**Creating a Basic Card**

The card markup is pretty straight forward. The outer wrapper require the base class .card, whereas content can be placed inside the .card-body element. The following example will show you how to create a card with a picture, mixed with some text content and a button.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<img src="/examples/images/sample.svg" class="card-img-top" alt="Sample Image">

<div class="card-body text-center">

<h5 class="card-title">Alice Liddel</h5>

<p class="card-text">Alice is a freelance web designer and developer based in London. She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.</p>

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

</div>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card)

**Note:** Card doesn't have fixed width, they'll occupy the full width of its parent element. So, if you need a fixed width card you need to apply the width property on card yourself. Also, card have no margin by default, use [spacing utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#spacing-classes) if needed.

## Content Types for Card Component

The card component support a wide variety of content, including images, text, list groups, links, navs, and more. Here are the examples of what's supported by the card.

### Body Only Card

You can simply use .card with .card-body within, whenever you need to create a padded box.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Basic Card</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card">

<div class="card-body">This is some text within a padded box.</div>

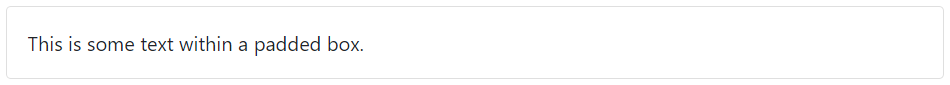
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-card)

### Card with Titles, Text, and Links

Further, you can also place title and links inside the card along with text, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Titles, Text, and Links</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<div class="card-body">

<h5 class="card-title">Eiffel Tower</h5>

<h6 class="card-subtitle mb-3 text-muted">Champ de Mars, Paris, France</h6>

<p class="card-text">Built in 1889 Eiffel Tower is one of the most iconic landmarks in the world.</p>

<a href="#" class="card-link">View pictures</a>

<a href="#" class="card-link">Learn more</a>

</div>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-titles-text-and-links)

### Card with Header and Footer

You can also add header and footer within your cards using the .card-header and .card-footer class, respectively. Let's take a look at the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Header and Footer</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card text-center">

<div class="card-header">Featured</div>

<div class="card-body">

<h5 class="card-title">NASA Launched Solar Probe</h5>

<p class="card-text">NASA launched Parker space probe in 2018 with the mission of making observations of the outer corona of the Sun. It is the first-ever mission to "touch" the Sun.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

<div class="card-footer text-muted">3 years ago</div>

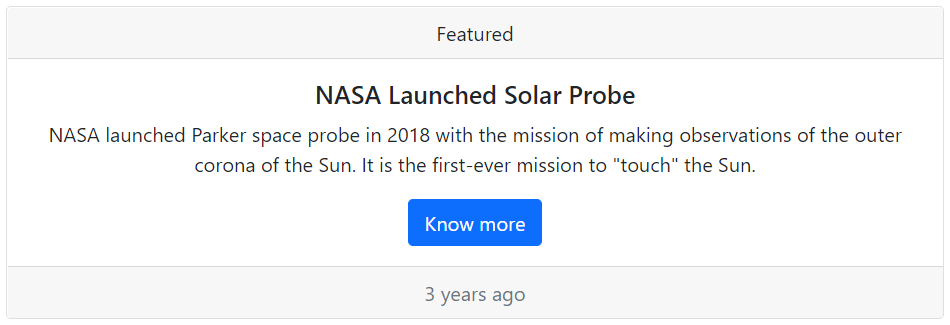
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-header-and-footer)

**Tip:** You can use text align utility classes such as .text-center and .text-end to align card's content to the center and right end, respectively. By default they're left aligned.

### Placing List Groups within Card

You can also place list groups inside the card along with other content types, as shown here.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with List Group</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<div class="card-header">Featured</div>

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

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

<li class="list-group-item">A second item</li>

<li class="list-group-item">A third item</li>

</ul>

<div class="card-body">

<a href="#" class="card-link">Add More</a>

<a href="#" class="card-link">Share</a>

</div>

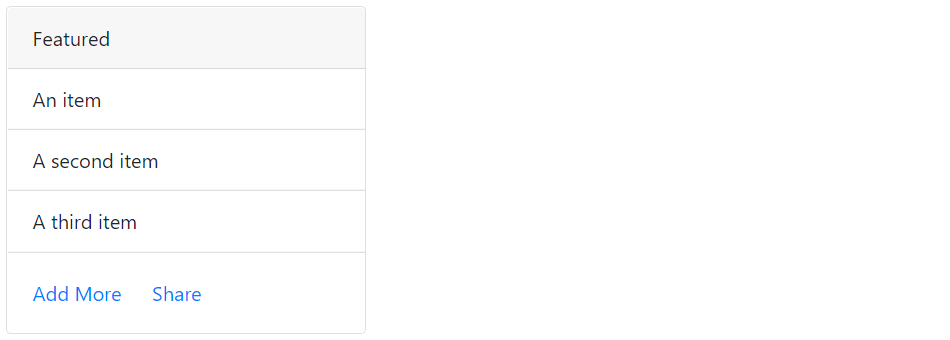
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-list-group)

### Mix and Match Multiple Content Types within Card

Feel free to mix and match multiple content types to create the card you need. The following example will create a fixed-width card with an image, text, list group, and hyperlinks.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Multiple Content Types</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<img src="/examples/images/thumbnail.svg" class="w-100 border-bottom" alt="Sample Image">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Here is some example text to make up the card's content. Replace it with your own text anytime.</p>

</div>

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

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

<li class="list-group-item">A second item</li>

<li class="list-group-item">A third item</li>

</ul>

<div class="card-body">

<a href="#" class="card-link">Card link</a>

<a href="#" class="card-link">Another link</a>

</div>

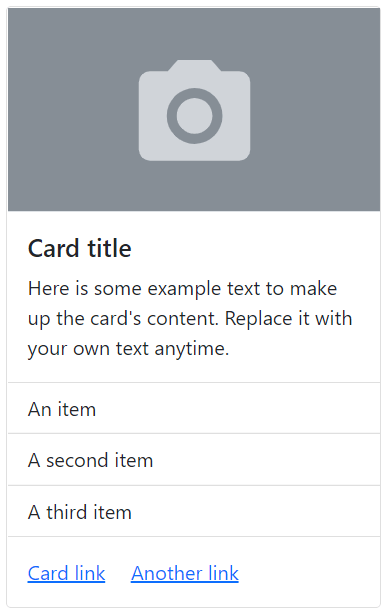
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-multiple-content-types)

**Adding Navigation to Cards**

You can also add [Bootstrap's nav components](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navs.php) such as tabs and pills to the card header.

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

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Tab Navigation</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card text-center">

<div class="card-header">

<ul class="nav nav-tabs card-header-tabs" id="myTab">

<li class="nav-item">

<a href="#home" class="nav-link active" data-bs-toggle="tab">Home</a>

</li>

<li class="nav-item">

<a href="#profile" class="nav-link" data-bs-toggle="tab">Profile</a>

</li>

<li class="nav-item">

<a href="#messages" class="nav-link" data-bs-toggle="tab">Messages</a>

</li>

</ul>

</div>

<div class="card-body">

<div class="tab-content">

<div class="tab-pane fade show active" id="home">

<h5 class="card-title">Home tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

<div class="tab-pane fade" id="profile">

<h5 class="card-title">Profile tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

<div class="tab-pane fade" id="messages">

<h5 class="card-title">Messages tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

</div>

</div>

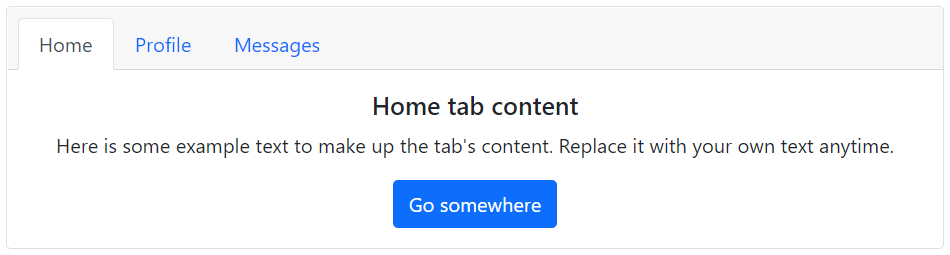
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-tabs-navigation)

Similarly, you can add pills nav to the card by using an additional class .card-header-pills along with the class .nav-pills on the .nav element, as shown below:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Pills Navigation</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card text-center">

<div class="card-header">

<ul class="nav nav-pills card-header-pills">

<li class="nav-item">

<a href="#home" class="nav-link active" data-bs-toggle="tab">Home</a>

</li>

<li class="nav-item">

<a href="#profile" class="nav-link" data-bs-toggle="tab">Profile</a>

</li>

<li class="nav-item">

<a href="#messages" class="nav-link" data-bs-toggle="tab">Messages</a>

</li>

</ul>

</div>

<div class="card-body">

<div class="tab-content">

<div class="tab-pane fade show active" id="home">

<h5 class="card-title">Home tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

<div class="tab-pane fade" id="profile">

<h5 class="card-title">Profile tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

<div class="tab-pane fade" id="messages">

<h5 class="card-title">Messages tab content</h5>

<p class="card-text">Here is some example text to make up the tab's content. Replace it with your own text anytime.</p>

<a href="#" class="btn btn-primary">Go somewhere</a>

</div>

</div>

</div>

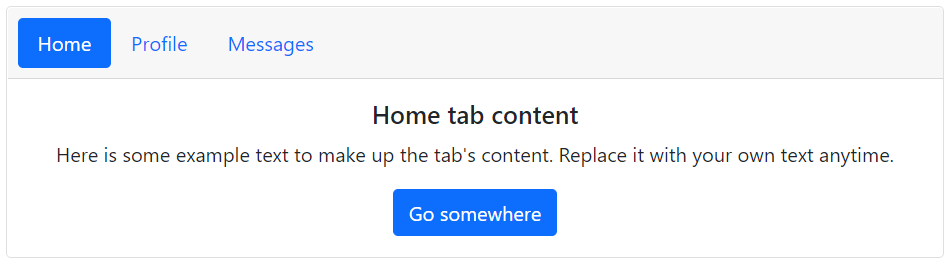
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-with-pills-navigation)

## Customizing the Card Styles

There are several options available for customizing the card's backgrounds, borders, and color.

### Customizing Background and Color

You can simply use the [background](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) and [color](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes) utility classes to change the appearance of a card. Let's try out the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Styles</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="row">

<div class="col-6">

<div class="card text-white bg-primary mb-4">

<div class="card-body">

<h5 class="card-title">Primary card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-secondary mb-4">

<div class="card-body">

<h5 class="card-title">Secondary card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-success mb-4">

<div class="card-body">

<h5 class="card-title">Success card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-danger mb-4">

<div class="card-body">

<h5 class="card-title">Danger card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-warning mb-4">

<div class="card-body">

<h5 class="card-title">Warning card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-info mb-4">

<div class="card-body">

<h5 class="card-title">Info card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card text-white bg-dark">

<div class="card-body">

<h5 class="card-title">Dark card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card bg-light">

<div class="card-body">

<h5 class="card-title">Light card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

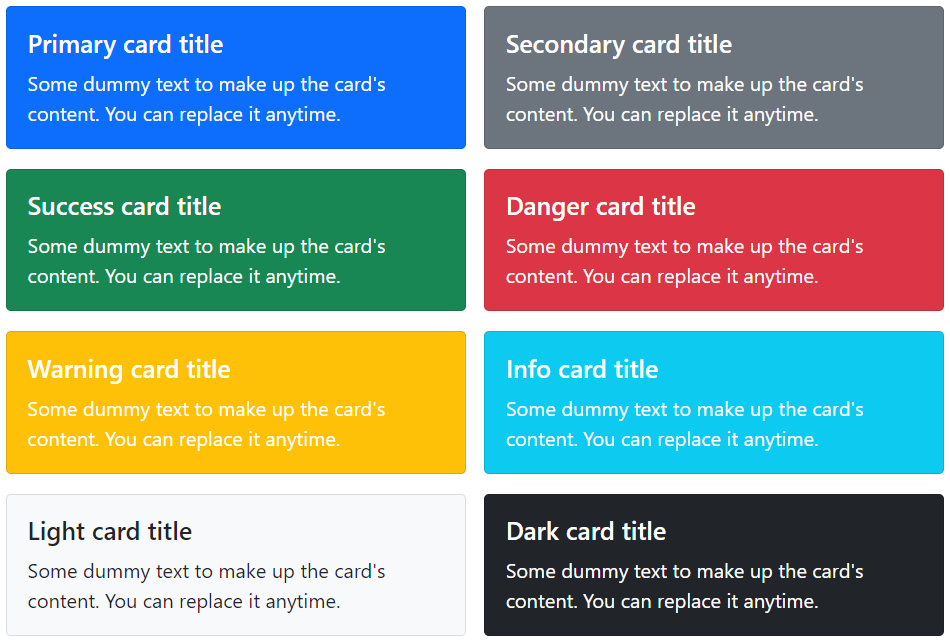
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-styles)

### Customizing Border and Color

Similarly, you can customize the border and text color of any card using the text and border utility classes. Just apply these classes on the .card or its child elements, as shown below:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Styles Outline</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="row">

<div class="col-6">

<div class="card border-primary mb-4">

<div class="card-body text-primary">

<h5 class="card-title">Primary card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-secondary mb-4">

<div class="card-body text-secondary">

<h5 class="card-title">Secondary card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-success mb-4">

<div class="card-body text-success">

<h5 class="card-title">Success card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-danger mb-4">

<div class="card-body text-danger">

<h5 class="card-title">Danger card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-warning mb-4">

<div class="card-body text-warning">

<h5 class="card-title">Warning card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-info mb-4">

<div class="card-body text-info">

<h5 class="card-title">Info card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-dark mb-4">

<div class="card-body text-dark">

<h5 class="card-title">Dark card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

<div class="col-6">

<div class="card border-light mb-4">

<div class="card-body text-muted">

<h5 class="card-title">Light card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

</div>

</div>

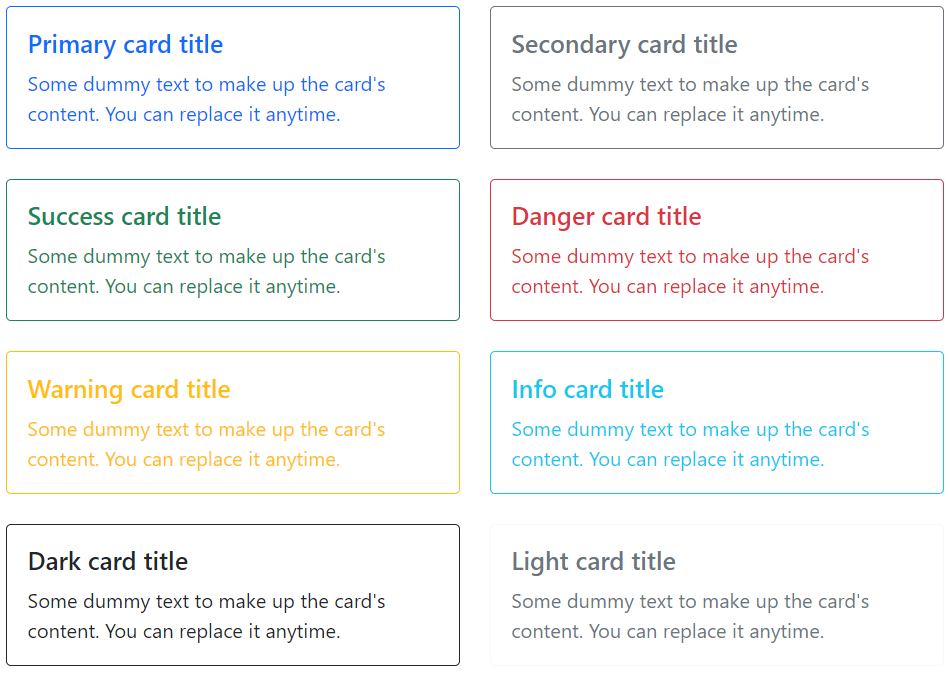
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-styles-outline)

## Card Layout Options

In addition to styling of the cards, Bootstrap also includes a few options for laying out the series of cards. However, these layouts are not responsive yet.

### Creating the Card Groups

You can use card groups to render cards as a single, attached element with equal width and height columns. However, cards inside a card group become horizontally stacked on extra small devices (i.e. viewport width <576px). Let's try out an example and see how it actually works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Group</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card-group">

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

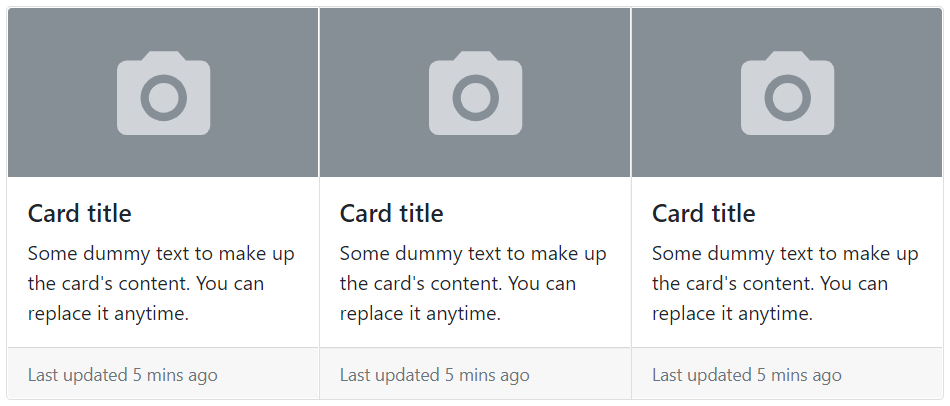
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=cards-group)

**Creating the Card Grids**

You can use the Bootstrap grid system and its .row-cols-\* classes to control how many grid columns (wrapped around your cards) to show per row. For example, you can use the class .row-cols-1 to show one card per row, similarly you can use the class .row-cols-md-2 to show two cards per row, from the medium breakpoint up (i.e. viewport width ≥768px).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Cards Grid</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="row row-cols-1 row-cols-md-2 g-4">

<div class="col">

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

</div>

<div class="col">

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

</div>

<div class="col">

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

</div>

<div class="col">

<div class="card">

<img src="/examples/images/thumbnail.svg" class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 5 mins ago</small>

</div>

</div>

</div>

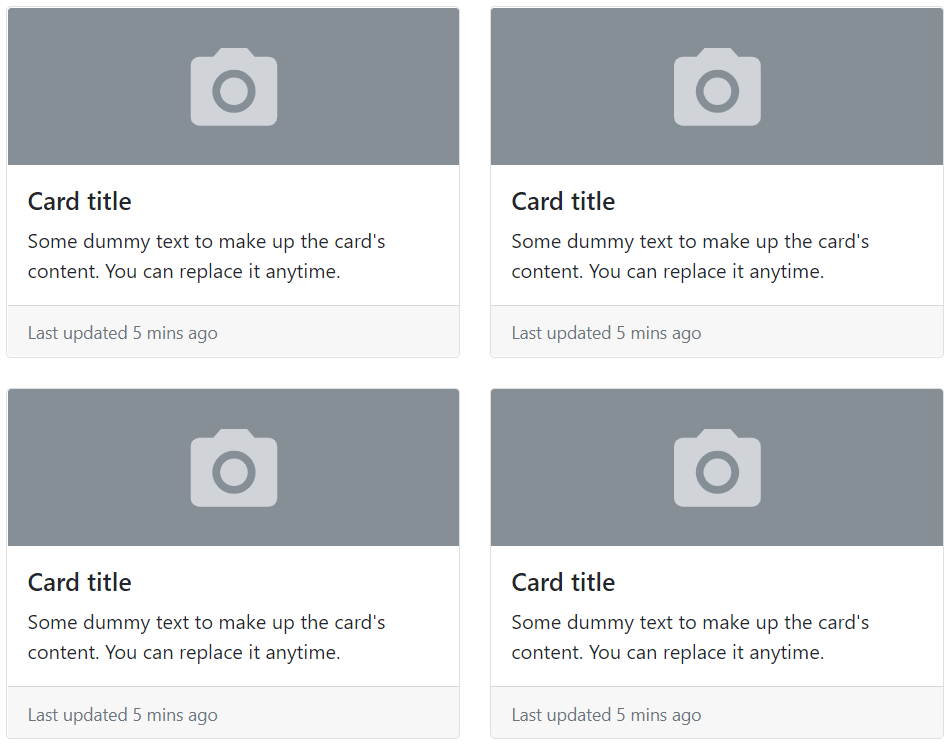
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=cards-grid)

**Creating Horizontal Cards**

You can also create horizontal cards where image and text content are placed side-by-side using a combination of grid and utility classes, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Horizontal Card</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card" style="max-width: 500px;">

<div class="row g-0">

<div class="col-sm-5" style="background: #868e96;">

<img src="/examples/images/sample.svg" class="card-img-top h-100" alt="...">

</div>

<div class="col-sm-7">

<div class="card-body">

<h5 class="card-title">Alice Liddel</h5>

<p class="card-text">Alice is a freelance web designer and developer based in London. She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.</p>

<a href="#" class="btn btn-primary stretched-link">View Profile</a>

</div>

</div>

</div>

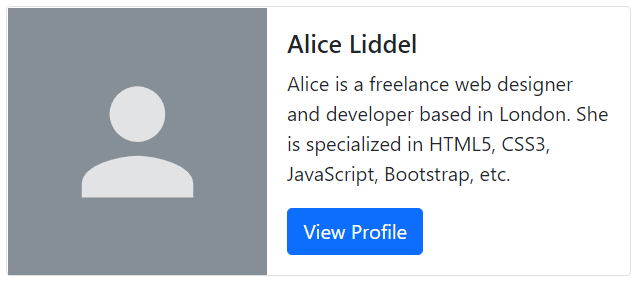
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=horizontal-card)

**Card Image Overlays**

You can even turn an image into a card background and place the card's text on the top it using the class .card-img-overlay in place of .card-body. Depending on the image, you may need additional styles for better adjustments. Let's check out an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Image Overlays</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="card text-white" style="width: 350px;">

<img src="/examples/images/sample.svg" class="card-img-top" alt="...">

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

<h5 class="card-title">Alice Liddel</h5>

<p class="card-text">Alice is a freelance web designer and developer based in London. She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.</p>

<a href="#" class="btn btn-primary stretched-link">View Profile</a>

</div>

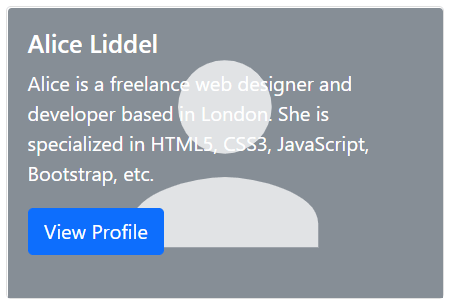
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-image-overlays)

**Note:** The card content should not be larger than the height of the image. If content is larger than the image the content will be displayed outside the image.

**Text Alignment inside Card**

You can easily change the text alignment of any card—entirely or specific parts—with the text alignment utility classes. For example, you can use the class .text-center and .text-end to align the card's text content to the center and to the right end, respectively.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Text Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="row row-cols-1 row-cols-md-3 g-3">

<div class="col">

<!-- Card with default left text alignment -->

<div class="card">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

<div class="col">

<!-- Card with center text alignment -->

<div class="card text-center">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

<div class="col">

<!-- Card with right text alignment -->

<div class="card text-end">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

</div>

</div>

</body>

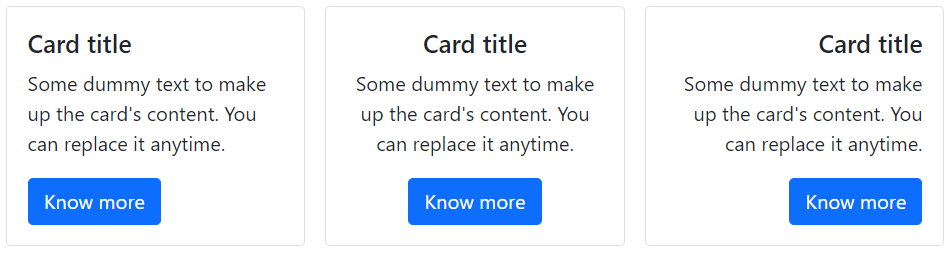
</html>

— The output of the above example will look something like this:

**Specifying Card Size**

Cards have no specific width, they are 100% wide by default. However, you can change this as needed with custom CSS, grid classes, or sizing utility classes.

Let's try out the following example to understand how it basically works:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=card-text-alignment)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card Sizing</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="container-fluid">

<!-- Card sizing using grid markup -->

<div class="row mb-4">

<div class="col-sm-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

<div class="col-sm-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

</div>

<!-- Card sizing using sizing utility classes -->

<div class="card w-75 mb-4">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

<!-- Card sizing using sizing utility classes -->

<div class="card" style="width: 15rem;">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some dummy text to make up the card's content. You can replace it anytime.</p>

<a href="#" class="btn btn-primary">Know more</a>

</div>

</div>

</div>

</div>

</body>

</html><!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Card with Stretched Link</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<img src="/examples/images/sample.svg" class="card-img-top" alt="...">

<div class="card-body text-center">

<h5 class="card-title">Alice Liddel</h5>

<p class="card-text">Alice is a freelance web designer and developer based in London. She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.</p>

<a href="#" class="btn btn-primary stretched-link">View Profile</a>

</div>

</div>

</div>

</body>

</html>

## Card with Stretched Link

You can add the class .stretched-link to a link inside the card to make the whole card clickable (i.e. whole card act like a link). Multiple links are not recommended with stretched links.

Try out the following example to see how this actually works:

# Bootstrap Media Objects

In this tutorial you will learn how to create the media objects in Bootstrap.

## Using the Media Object in Bootstrap

Bootstrap media object has been discontinued from version 5. However, you can still create a layout that contains left- or right-aligned media object like images or video alongside the textual content such as blog comments, Tweets, etc. using the flex and spacing utility classes.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Media Objects</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="d-flex">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Jhon Carter <small class="text-muted"><i>Posted on January 10, 2021</i></small></h5>

<p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>

</div>

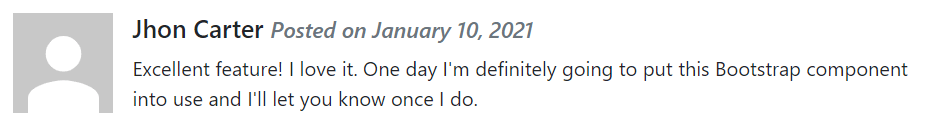
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=media-objects)

You can also create other variations of media object. Apply the image modifier classes like .rounded or .rounded-circle to the image to create rounded corner or circular image.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Rounded Media Objects</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="d-flex">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" class="rounded-circle" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Jhon Carter <small class="text-muted"><i>Posted on January 10, 2021</i></small></h5>

<p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>

</div>

</div>

<hr>

<div class="d-flex">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" class="rounded" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Jhon Carter <small class="text-muted"><i>Posted on January 10, 2021</i></small></h5>

<p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>

</div>

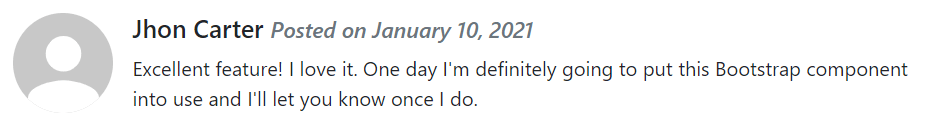
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=rounded-media-objects)

**Creating Nested Media Objects**

Media objects can also be nested inside other media object. It can be very useful for creating comment threads in a blog post. Let's check out an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nested Media Objects</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="d-flex">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" class="rounded-circle" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Jhon Carter <small class="text-muted"><i>Posted on January 10, 2021</i></small></h5>

<p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>

<!-- Nested media object -->

<div class="d-flex mt-4">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" class="rounded-circle" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Clark Kent <small class="text-muted"><i>Posted on January 12, 2021</i></small></h5>

<p>Thanks, you found this component useful. Don't forget to check out other Bootstrap components as well. They're also very useful.</p>

</div>

</div>

</div>

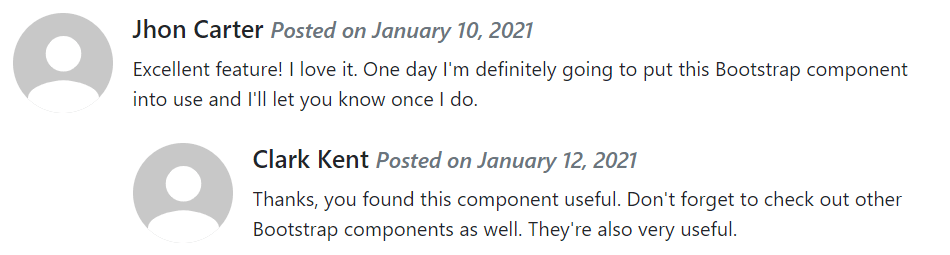
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nested-media-objects)

**Alignment of Media Objects**

You can also change the horizontal alignment of content and media by simply tweaking the HTML code itself, as demonstrated in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Horizontal Alignment of Bootstrap Media Objects</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="d-flex">

<div class="flex-grow-1 me-3">

<h5>Jhon Carter <small class="text-muted"><i>Posted on January 10, 2021</i></small></h5>

<p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>

</div>

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" width="80" height="80" alt="Sample Image">

</div>

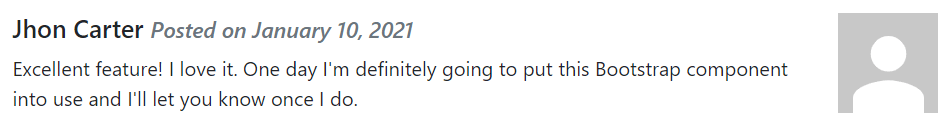
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=horizontal-alignment-of-media-objects)

Besides that you can also align the images or other media at the middle or bottom of the content block using the flexbox utilities classes, for example, you can use the class .align-self-center for vertical center alignment, and the class .align-self-end for bottom alignment.

By default, the media inside a media object is top aligned. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Vertical Alignment of Bootstrap Media Objects</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<!--Top aligned media-->

<div class="d-flex">

<div class="flex-shrink-0">

<img src="/examples/images/avatar.svg" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Top aligned media <small class="text-muted"><i>This is Default</i></small></h5>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis.</p>

<p>Amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.</p>

</div>

</div>

<hr>

<!--Middle aligned media-->

<div class="d-flex">

<div class="flex-shrink-0 align-self-center">

<img src="/examples/images/avatar.svg" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Middle Aligned Media</h5>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis.</p>

<p>Amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.</p>

</div>

</div>

<hr>

<!--Bottom aligned media-->

<div class="d-flex">

<div class="flex-shrink-0 align-self-end">

<img src="/examples/images/avatar.svg" width="80" height="80" alt="Sample Image">

</div>

<div class="flex-grow-1 ms-3">

<h5>Bottom Aligned Media</h5>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis.</p>

<p>Amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.</p>

</div>

</div>

</div>

</body>

</html>

# Bootstrap Icons

In this tutorial you will learn how to include and use Bootstrap icons on a web page.

## Using Icons in Bootstrap 5

Bootstrap now includes over [1,300 high quality icons](https://www.tutorialrepublic.com/bootstrap-icons-classes.php), which are available in SVGs, SVG sprite, or web fonts format. You can use them with or without Bootstrap in any project.

The advantage of using font icons is, you can create icons of any color just through applying the CSS [color](https://www.tutorialrepublic.com/css-reference/css-color-property.php) property. Also, to change the size of icons you can simply use the CSS [font-size](https://www.tutorialrepublic.com/css-reference/css-font-size-property.php) property.

Now, let's see how to include and use Bootstrap icons on a web page.

## Including Bootstrap Icons in a Web Page

The simplest way to include Bootstrap icons in a web page is using the CDN link. This CDN link basically points to a remote CSS file that includes all the necessary classes to generate font icons.

You can include Bootstrap icons in a Bootstrap template as well as in a simple web page without using the Bootstrap framework. Let's take a look at the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Including Bootstrap Icons in HTML</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

<!-- Bootstrap Font Icon CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

</head>

<body>

<h1 class="m-4"><i class="bi-globe"></i> Hello, world!</h1>

<!-- Bootstrap JS Bundle with Popper -->

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

</body>

</html>

**How to Use Bootstrap Icons in Your Code**

To use Bootstrap icons in your code you'll require an [<i>](https://www.tutorialrepublic.com/html-reference/html-i-tag.php) tag with an individual icon class .bi-\* applied on it. The general syntax for using Bootstrap icons is:

<i class="bi-*class-name*"></i>

Where ***class-name*** is the name of the particular icon class, e.g. search, person, calendar, star, globe, facebook, twitter, and so on. See the list of all [Bootstrap icons classes](https://www.tutorialrepublic.com/bootstrap-icons-classes.php).

For example, to place search icon inside a button you could do something like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Buttons with Icons</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

<!-- Bootstrap Font Icon CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

</head>

<body>

<div class="m-4">

<button type="submit" class="btn btn-primary"><span class="bi-search"></span> Search</button>

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

</div>

<!-- Bootstrap JS Bundle with Popper -->

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

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=buttons-with-icons)

Similarly, you can place icons inside the navs, forms, tables, paragraphs or anywhere you want. In the next chapter you will see how to use these icons in [Bootstrap nav components](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navs.php).

**Note:** Remember to leave a space after the closing tag of icon element (i.e. after </i> tag), when using the icons along with the strings of text such as inside buttons or navigation links, to ensure that there is proper spacing between the icon and text.

**Using Font Awesome Icons in Bootstrap**

You can also use external icon libraries in Bootstrap. One of the most popular and highly compatible external icon library for Bootstrap is Font Awesome. It provides over 675 icons which are available in SVG, PNG, as well as in web font format for better usability and scalability.

You can simply use the freely available font-awesome CDN link to include it in your project. Let's take a look at the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Including Font Awesome Icons in Bootstrap</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

<!-- Font Awesome CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<h1 class="m-4"><i class="fa fa-globe"></i> Hello, world!</h1>

<!-- Bootstrap JS Bundle with Popper -->

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

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Including Font Awesome Icons in Bootstrap</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

<!-- Font Awesome CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<h1 class="m-4"><i class="fa fa-globe"></i> Hello, world!</h1>

<!-- Bootstrap JS Bundle with Popper -->

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

</body>

</html>

**How to Use Font Awesome Icons in Your Code**

To use Font Awesome icons in your code you'll require an [<i>](https://www.tutorialrepublic.com/html-reference/html-i-tag.php) tag along with a base class .fa and an individual icon class .fa-\*. The general syntax for using font-awesome icons is:

<i class="fa fa-*class-name*"></i>

Where ***class-name*** is the name of the particular icon class, e.g. search, user, calendar, star, globe, facebook, twitter, and so on. See the list of all [Font Awesome icons classes](https://www.tutorialrepublic.com/font-awesome-icons-classes.php).

For example, you can place font-awesome search icon inside a button like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Buttons with Font Awesome Icons</title>

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

<!-- Font Awesome CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<div class="m-4">

<button type="submit" class="btn btn-primary"><span class="fa fa-search"></span> Search</button>

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

</div>

<!-- Bootstrap JS Bundle with Popper -->

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

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=buttons-with-font-awesome-icons)

Similarly, you can place Font Awesome icons inside the navs, forms, tables, paragraphs, and other components in the same way as you do with Bootstrap icons.

# Bootstrap Nav: Tabs and Pills

In this tutorial you will learn how to create navigation using Bootstrap nav component.

## Bootstrap Nav Components

Bootstrap provides an easy and quick way to create basic navigation as well as components like tabs and pills which are very flexible and elegant. All the Bootstrap's nav components, including tabs and pills, share the same base markup and styles through the base .nav class.

## Creating Basic Nav with Bootstrap

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Basic Nav</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav">

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

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

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

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

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav)

**Note:** You can use the class .disabled to make a link look like disabled. But, the .disabled class only changes the visual appearance of the link by making it gray and removing the hover effect, however the link will remain clickable unless you [remove the "href" attribute](https://www.tutorialrepublic.com/faq/how-to-remove-clickable-behavior-from-a-disabled-link-using-jquery.php).

**Alignment of Nav Items**

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

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Center Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav justify-content-center">

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

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

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

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

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-center-alignment)

Similarly, you can align nav items to right using the class .justify-content-end, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Right Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav justify-content-end">

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

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

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

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

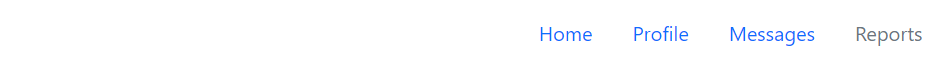
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-right-alignment)

Moreover, you can even vertically stack your nav items by changing the flex item direction with the class .flex-column. Also, if you want to stack your nav vertically on smaller viewports but not on others, use it with responsive breakpoint (e.g., .flex-sm-column).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Vertically Stacked Nav</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav flex-column">

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

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

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

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

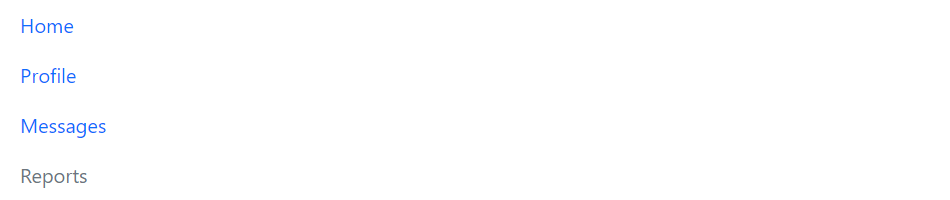
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-nav)

**Creating the Basic Tabs**

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Tabs</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-tabs">

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

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

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

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

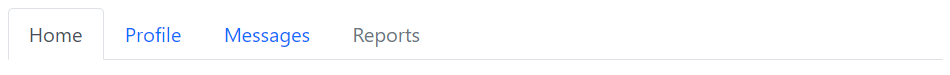
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-tabs)

See the tutorial on [Bootstrap tabs](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tabs.php) to learn how to create dynamic tab to toggle between content.

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Tabs with Icons</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-tabs">

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

<i class="bi-house-door"></i> Home

</a>

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

<i class="bi-person"></i> Profile

</a>

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

<i class="bi-envelope"></i> Messages

</a>

<a href="#" class="nav-item nav-link disabled" tabindex="-1">

<i class="bi-bar-chart"></i> Reports

</a>

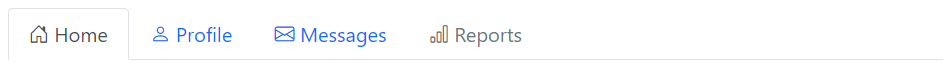
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tabs-with-icons)

See the tutorial on [Bootstrap icons](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-icons.php) to learn how to use icons in Bootstrap. Also, check out [Bootstrap icons classes](https://www.tutorialrepublic.com/bootstrap-icons-classes.php) to explore the icons provided by Bootstrap.

**Creating the Pills Nav**

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Pills</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills">

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

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

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

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

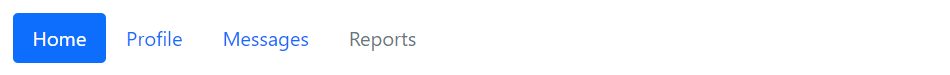
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-pills)

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pills Nav with Icons</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills">

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

<i class="bi-house-door"></i> Home

</a>

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

<i class="bi-person"></i> Profile

</a>

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

<i class="bi-envelope"></i> Messages

</a>

<a href="#" class="nav-item nav-link disabled" tabindex="-1">

<i class="bi-bar-chart"></i> Reports

</a>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pills-nav-with-icons)

Additionally, you can apply the class .flex-column on the .nav element to make the pills nav appear vertically stacked. You can alternatively use responsive versions (e.g., .flex-sm-column) if you need to stack them on specific viewports but not on others.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Vertically Stacked Pills Nav</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

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

<i class="bi-house-door"></i> Home

</a>

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

<i class="bi-person"></i> Profile

</a>

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

<i class="bi-envelope"></i> Messages

</a>

<a href="#" class="nav-item nav-link disabled" tabindex="-1">

<i class="bi-bar-chart"></i> Reports

</a>

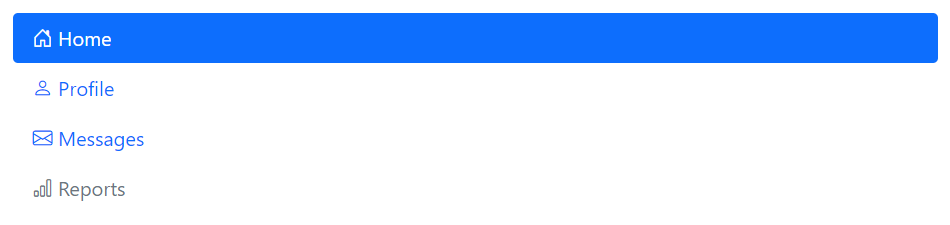
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-pills-nav)

## Bootstrap Nav with Dropdown Menus

You can add dropdown menus to a link inside tabs and pills nav with a little extra markup.

The four CSS classes .dropdown, .dropdown-toggle, .dropdown-menu and .dropdown-item are required in addition to the .nav, .nav-tabs or .nav-pills classes to create a simple dropdown menu inside tabs and pills nav without using any JavaScript code.

### Creating Tabs with Dropdowns

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Tabs with Dropdown Menus</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-tabs">

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

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

<div class="nav-item dropdown">

<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

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

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

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

</div>

</div>

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

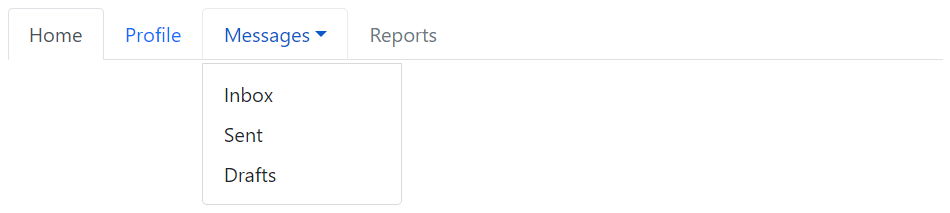
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tabs-with-dropdown-menus)

### Creating Pills with Dropdowns

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pills with Dropdown Menus</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills">

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

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

<div class="nav-item dropdown">

<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

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

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

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

</div>

</div>

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

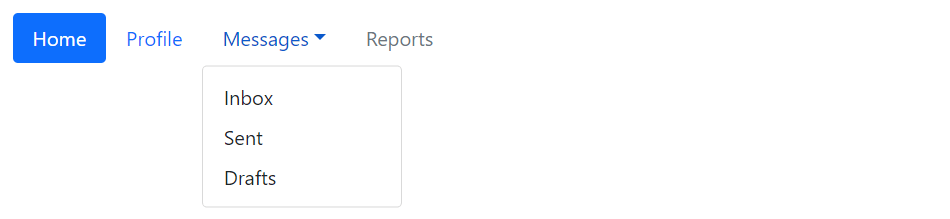
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pills-with-dropdown-menus)

You will learn more about dropdown menus later in [Bootstrap dropdowns](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-dropdowns.php) chapter.

**Fill and Justify Nav Component**

You can force your .nav-items to extend and proportionately fill all available width using the class .nav-fill on the .nav element. In the following example all horizontal space is occupied by the nav items, but every nav item doesn't have the same width.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Fill</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills nav-fill">

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

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

<a href="#" class="nav-item nav-link active">Explore Products</a>

<a href="#" class="nav-item nav-link">Contact Us</a>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-fill)

Alternatively, you can use the class .nav-justified instead of.nav-fill, if you want nav that fills all horizontal space as well as every nav item has the same width.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Nav Justified</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills nav-justified">

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

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

<a href="#" class="nav-item nav-link active">Explore Products</a>

<a href="#" class="nav-item nav-link">Contact Us</a>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=nav-justified)

# Bootstrap Navbar

In this tutorial you will learn how to create static and fixed positioned responsive navigation headers using the Bootstrap navbar component.

## Creating Navbar with Bootstrap

You can use the Bootstrap navbar component to create responsive navigation header for your website or application. These responsive navbar will be collapsed on devices having small viewports like mobile phones but expand when user click the toggle button. However, it will be horizontal as normal on the medium and large devices such as laptop or desktop.

You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort. The following example will show you how to create a simple static navbar with navigation links.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Static Navbar</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

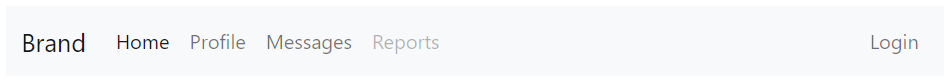
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=static-navbar)

**Tip:** Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl|-xxl} for responsive collapsing, and color scheme classes for appearance. Also, use the spacing and flex utility classes for controlling spacing and alignment within navbars.

**Note:** Navbars and their contents are fluid by default. Change the [container](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-containers.php) (e.g. .container{-sm|-md|-lg|-xl|-xxl}) to limit their horizontal width in different ways.

**Adding Logo Images to Navbars**

You can also place your logo image inside the navbar, instead of plain text. However, you need to set the logo height manually to fit it properly inside the navbar, as shown here:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar with Logo Image</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

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

<img src="/examples/images/logo.svg" height="28" alt="CoolBrand">

</a>

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

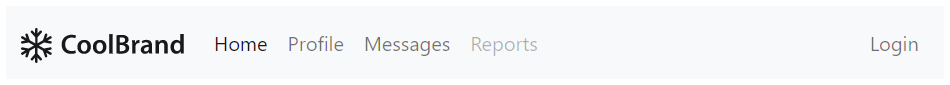
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=navbar-with-logo)

**Tip:** Use the [utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php) such as .ms-auto, .me-auto, .justify-content-between, etc. to align the nav links, forms, buttons or text inside the navbar.

**Adding Dropdowns to Navbar**

You can also include dropdowns and search box within the navbars, as shown here:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar with Dropdown</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

<div class="nav-item dropdown">

<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

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

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

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

</div>

</div>

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

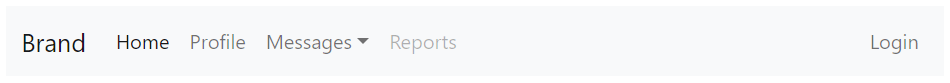
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=navbar-with-dropdown)

Check out the snippets section for examples of some [beautifully designed Bootstrap navbars](https://www.tutorialrepublic.com/snippets/gallery.php?tag=navbar).

**Placing Search Form inside Navbar**

Search form is a very common component of the navbars and you have seen it on various website quite often. You can place various form controls within a navbar using the class .d-flex on the [<form>](https://www.tutorialrepublic.com/html-reference/html-form-tag.php) element, as demonstrated in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar with Dropdown</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

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

</head>

<body>

<div class="m-4">

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">

<div class="navbar-nav">

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

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

<div class="nav-item dropdown">

<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

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

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

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

</div>

</div>

</div>

<form class="d-flex">

<div class="input-group">

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

<button type="button" class="btn btn-secondary"><i class="bi-search"></i></button>

</div>

</form>

<div class="navbar-nav">

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

</div>

</div>

</div>

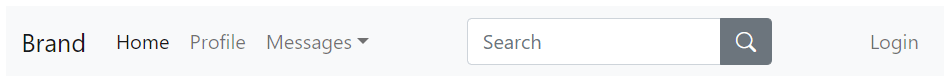
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=navbar-with-search-form)

**Changing the Color Scheme of Navbars**

You can also change the color scheme of the navbar by using the .navbar-light for the light background colors, or .navbar-dark for the dark background colors. Then, customize it with the [background color utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes), such as .bg-dark, .bg-primary, and so on.

Alternatively, you can also apply the CSS [background-color](https://www.tutorialrepublic.com/css-reference/css-background-color-property.php) property on the .navbar element yourself to customize the navbar theme, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar Color Schemes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

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

<style>

/\* Custom style \*/

.navbar{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse1">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse1">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse2">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse2">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-secondary">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse3">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse3">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-success">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse4">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse4">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-info">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse5">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse5">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-danger">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse6">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse6">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-dark bg-warning">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse7">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse7">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse8">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse8">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

</nav>

<nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd;">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse9">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse9">

<div class="navbar-nav">

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

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

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

</div>

<form class="d-flex ms-auto">

<input type="text" class="form-control me-sm-2" placeholder="Search">

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

</form>

</div>

</div>

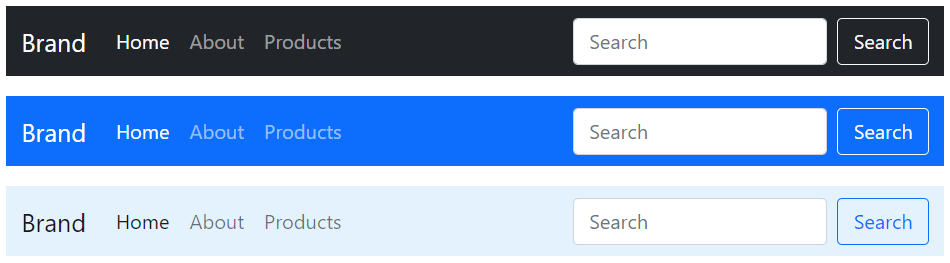
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=navbar-color-schemes)

## Bootstrap Fixed Navbars

Bootstrap also provides mechanism to create navbar that is fixed to the top, fixed to the bottom, or stickied to the top (i.e. scrolls with the page until it reaches the top, then stays there).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar Fixed to Top</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Custom CSS to prevent the content

to go underneath the navbar \*/

body{

padding-top: 70px;

}

</style>

</head>

<body>

<div class="m-4">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

</nav>

<!-- Demo Content -->

<div class="container">

<div class="p-5 mb-4 bg-light rounded-3">

<h1>Fixed Top Navbar</h1>

<p class="lead">The fixed-top navbar bar is fixed at the top of the viewport and does not scroll with the rest of the page.</p>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus. Tibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed.</p>

<p>Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.</p>

<p>Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id.Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et, viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel. Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus. Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis ultricies, magna quam rhoncus erat, in lacinia libero magna a ipsum.</p>

<p>Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.</p>

</div>

</div>

</body>

</html>

### Navbar Fixed to the Bottom

Similarly, add the class .fixed-bottom to the .navbar element to fix the navbar at the bottom of the viewport. It also won't scroll with the page. Let's see how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar Fixed to Bottom</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Custom CSS to prevent the content

to go underneath the navbar \*/

body{

padding-bottom: 70px;

}

</style>

</head>

<body>

<div class="m-4">

<nav class="navbar fixed-bottom navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

</nav>

<!-- Demo Content -->

<div class="container">

<div class="p-5 mb-4 bg-light rounded-3">

<h1>Fixed Bottom Navbar</h1>

<p class="lead">The fixed-bottom navbar bar is fixed at the bottom of the viewport and does not scroll with the rest of the page.</p>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus. Tibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed.</p>

<p>Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.</p>

<p>Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id.Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et, viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel. Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus. Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis ultricies, magna quam rhoncus erat, in lacinia libero magna a ipsum.</p>

<p>Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.</p>

</div>

</div>

</body>

</html>

### Navbar Stickied to the Top

You can also create sticky top navbar that scrolls with the page until it reaches the top, then stays there, by simply using the .sticky-top class on the .navbar element, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Navbar Stickied to the Top</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="p-5 bg-warning">

<h1>Demo Header</h1>

</div>

<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

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

<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarCollapse">

<div class="navbar-nav">

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

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

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

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

</div>

<div class="navbar-nav ms-auto">

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

</div>

</div>

</div>

</nav>

<!-- Demo Content -->

<div class="container mt-4">

<div class="p-5 mb-4 bg-light rounded-3">

<h1>Sticky Top Navbar</h1>

<p class="lead">The static-top navbar scrolls with the page until it reaches the top, then stays there.</p>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus. Tibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed.</p>

<p>Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.</p>

<p>Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id.Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et, viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel. Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus. Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis ultricies, magna quam rhoncus erat, in lacinia libero magna a ipsum.</p>

<p>Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.</p>

</div>

</body>

</html>

**Note:** Remember to add [padding](https://www.tutorialrepublic.com/css-reference/css-padding-property.php) (at least 70px) to the top or bottom of the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element to prevent the content to go underneath the navbar while implementing the fixed top or fixed bottom navbar. Also, be sure to add your custom [style sheet](https://www.tutorialrepublic.com/html-tutorial/html-styles.php) after the Bootstrap's CSS file, otherwise style rules in your style sheet may be overridden by the Bootstrap's one.

**Tip:** Place .navbar content inside the .container, .container-fluid or .container{-sm|-md|-lg|-xl|-xxl} for proper padding and alignment with the rest of the content.

# Bootstrap Accordion

In this tutorial you will learn how to create accordion with Bootstrap.

## Creating Accordion Widget with Bootstrap

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

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

Let's see how to build a simple accordion widget using the Bootstrap collapsible plugin.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Accordion</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<div class="accordion-item">

<h2 class="accordion-header" id="headingOne">

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

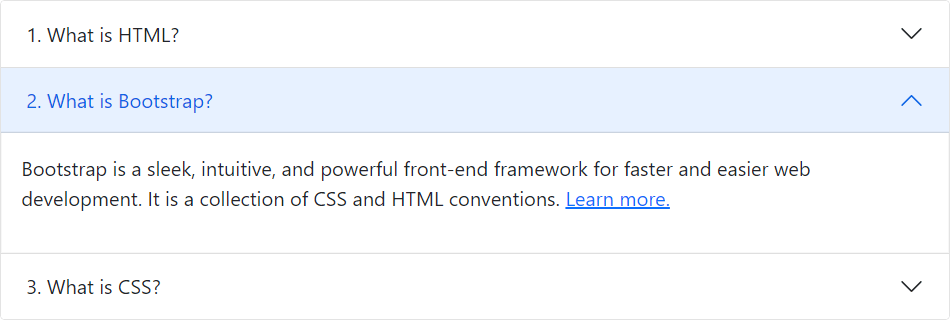
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=accordion)

**Bootstrap Accordion with Plus Minus Icons**

You can also use plus minus icons in the Bootstrap accordion widget (in place of chevron down and up icons) with a few lines of custom CSS code. Let's try out the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Accordion with Plus Minus Icon</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Custom style \*/

.accordion-button::after {

background-image: url("data:image/svg+xml,%3csvg viewBox='0 0 16 16' fill='%23333' xmlns='http://www.w3.org/2000/svg'%3e%3cpath fill-rule='evenodd' d='M8 0a1 1 0 0 1 1 1v6h6a1 1 0 1 1 0 2H9v6a1 1 0 1 1-2 0V9H1a1 1 0 0 1 0-2h6V1a1 1 0 0 1 1-1z' clip-rule='evenodd'/%3e%3c/svg%3e");

transform: scale(.7) !important;

}

.accordion-button:not(.collapsed)::after {

background-image: url("data:image/svg+xml,%3csvg viewBox='0 0 16 16' fill='%23333' xmlns='http://www.w3.org/2000/svg'%3e%3cpath fill-rule='evenodd' d='M0 8a1 1 0 0 1 1-1h14a1 1 0 1 1 0 2H1a1 1 0 0 1-1-1z' clip-rule='evenodd'/%3e%3c/svg%3e");

}

</style>

</head>

<body>

<div class="m-4">

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

<div class="accordion-item">

<h2 class="accordion-header" id="headingOne">

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

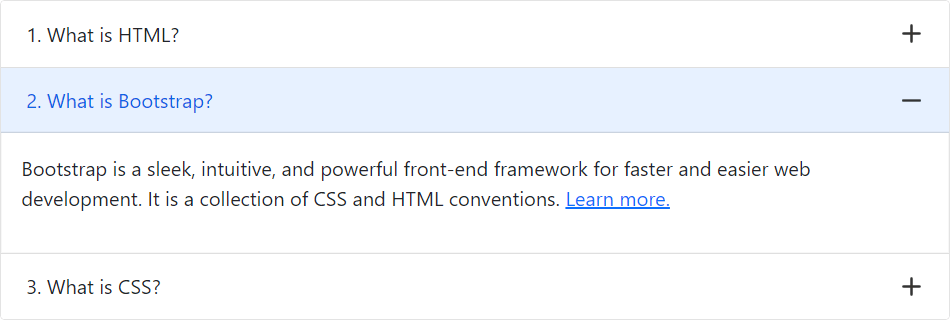
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=accordion-with-plus-minus-icon)

**Bootstrap Edge-to-Edge Accordions**

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

Let's check out the following example to understand how it actually works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Edge-to-Edge Accordion</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body class="bg-light">

<div class="accordion accordion-flush" id="myAccordion">

<div class="accordion-item">

<h2 class="accordion-header" id="headingOne">

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

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

<div class="card-body">

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

</div>

</div>

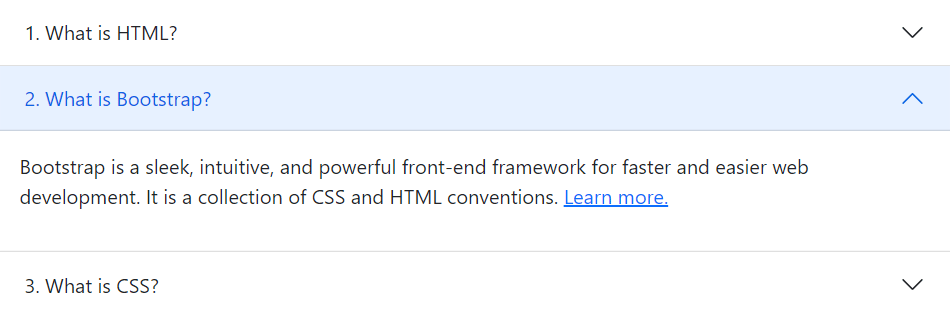
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=accordion)

**Bootstrap Accordion with Independent Panels**

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

Let's try out the following example and see how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Accordion with Independent Collapsible Panels</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<div class="accordion-item">

<h2 class="accordion-header" id="headingOne">

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

</h2>

<div id="collapseOne" class="accordion-collapse collapse">

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

<div id="collapseTwo" class="accordion-collapse collapse show">

<div class="card-body">

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

</div>

</div>

</div>

<div class="accordion-item">

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

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

</h2>

<div id="collapseThree" class="accordion-collapse collapse">

<div class="card-body">

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

</div>

</div>

</div>

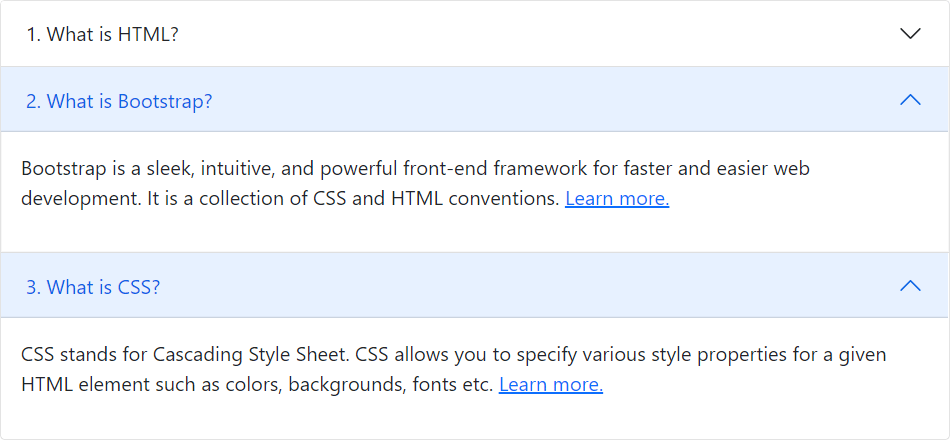
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=accordion-with-independent-collapsible-panels)

# Bootstrap Breadcrumbs

In this tutorial you will learn how to create breadcrumbs with Bootstrap.

## Creating Breadcrumbs with Bootstrap

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

You can create static breadcrumbs layouts with Bootstrap simply using the class .breadcrumb on the [ordered lists](https://www.tutorialrepublic.com/html-tutorial/html-lists.php), as demonstrated in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Creating Breadcrumb in Bootstrap</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item"><a href="#">Products</a></li>

<li class="breadcrumb-item active">Accessories</li>

</ol>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Breadcrumb](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=breadcrumb)

The default breadcrumb separator is /. But, you can change it with a little custom CSS, for example, if you want to use > as separator, you can apply the following style rules:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Changing Bootstrap Breadcrumb Separator</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Custom style for changing separator \*/

.breadcrumb-item + .breadcrumb-item::before {

content: ">";

}

</style>

</head>

<body>

<div class="m-4">

<nav>

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item"><a href="#">Products</a></li>

<li class="breadcrumb-item active">Accessories</li>

</ol>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Breadcrumb with Custom Separator](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=changing-breadcrumb-separator)

**Tip:** The separators inside the breadcrumb component are added automatically via CSS through the [::before](https://www.tutorialrepublic.com/css-tutorial/css-pseudo-elements.php) pseudo-element and the [content](https://www.tutorialrepublic.com/css-reference/css-content-property.php) property.

# Bootstrap Pagination

In this tutorial you will learn how to create pagination with Bootstrap.

## Creating Pagination with Bootstrap

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

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pagination</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ul class="pagination">

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

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

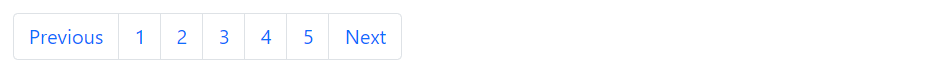
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pagination)

**Pagination with Disabled and Active States**

Links inside Bootstrap pagination can further be customized for different circumstances, like when user approaches to an end or start, or indicating current page number to the user. Use the class .disabled for making the links disabled and .active to indicate the current page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pagination with Disabled and Active State</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ul class="pagination">

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

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pagination-with-disabled-and-active-state)

**Note:** The .disabled class only displays links as it disabled it doesn't remove the click functionality, to do this you can swap active or disabled anchors with spans.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Using span Element for Disabled Link in Bootstrap Pagination</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ul class="pagination">

<li class="page-item disabled"><span class="page-link">Previous</span></li>

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

</nav>

</div>

</body>

</html>

## Changing the Sizes of Pagination

You can also change the sizes of pagination to increase or decrease the clickable area.

Add the relative sizing classes like .pagination-lg, or .pagination-sm to the .pagination base class for creating larger or smaller pagination. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pagination Sizes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<!-- Large pagination -->

<nav>

<ul class="pagination pagination-lg">

<li class="page-item disabled"><span class="page-link">Previous</span></li>

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

</nav>

<!-- Default pagination -->

<nav>

<ul class="pagination">

<li class="page-item disabled"><span class="page-link">Previous</span></li>

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

</nav>

<!-- Small pagination -->

<nav>

<ul class="pagination pagination-sm">

<li class="page-item disabled"><span class="page-link">Previous</span></li>

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

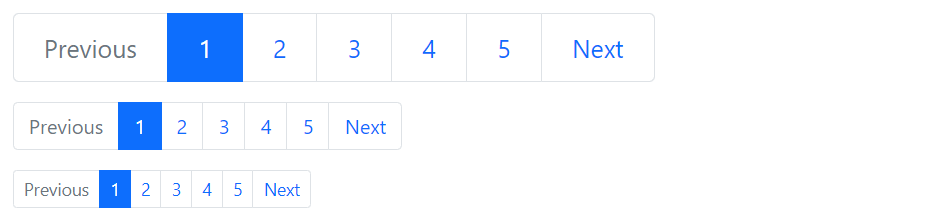
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pagination-sizes)

**Alignment of Pagination**

By default pagination is aligned horizontally left. To align it in the center of the page add the class .justify-content-center to the .pagination base class, as shown below:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pagination Center Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ul class="pagination justify-content-center">

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

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

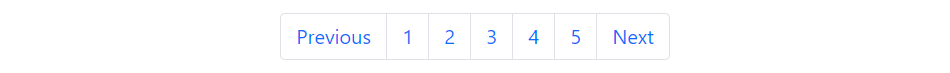
</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pagination-center-alignment)

Similarly, you can use the class .justify-content-end to align the pagination to the right:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pagination Right Alignment</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav>

<ul class="pagination justify-content-end">

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

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

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

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

<li class="page-item"><a href="#" class="page-link">4</a></li>

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

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

</ul>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pagination-right-alignment)

# Bootstrap Badges

In this tutorial you will learn how to create badges with Bootstrap.

## Creating Badges with Bootstrap

Badges are generally used to indicate some valuable information on the web pages such as important heading, warning messages, notification counter, etc.

The following example will show you how to create inline badges using the Bootstrap.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Badges</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<h1>Bootstrap heading <span class="badge bg-secondary">New</span></h1>

<h2>Bootstrap heading <span class="badge bg-secondary">New</span></h2>

<h3>Bootstrap heading <span class="badge bg-secondary">New</span></h3>

<h4>Bootstrap heading <span class="badge bg-secondary">New</span></h4>

<h5>Bootstrap heading <span class="badge bg-secondary">New</span></h5>

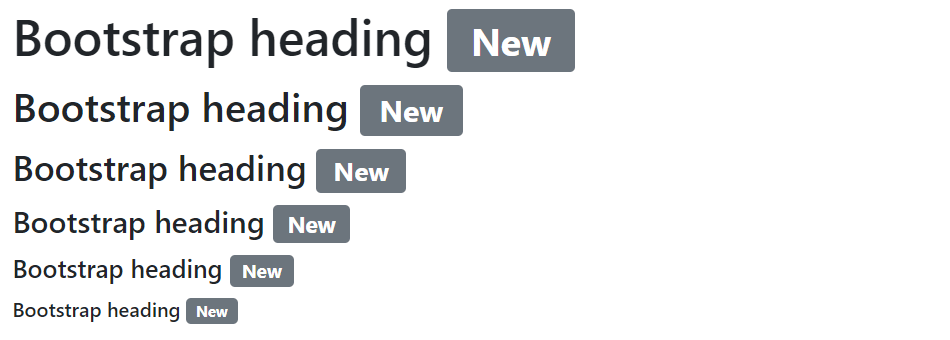
<h6>Bootstrap heading <span class="badge bg-secondary">New</span></h6>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=badges)

**Tip:** Bootstrap badges scale automatically to match the size of the immediate parent element by using the relative font sizing and em units.

**Changing the Appearance of Badges**

You can use the Bootstrap inbuilt [background color utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) to quickly change the appearance of the badges. Let's take a look at the following example.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Colored Badges</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<span class="badge bg-secondary">Secondary</span>

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

<span class="badge bg-danger">Danger</span>

<span class="badge bg-warning text-dark">Warning</span>

<span class="badge bg-info text-dark">Info</span>

<span class="badge bg-dark">Dark</span>

<span class="badge bg-light text-dark">Light</span>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Colored Badges](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=colored-badges)

**Note:** When using the light background color, such as Bootstrap's .bg-light, you'll need to use a dark text color utility, like .text-dark for proper text visibility. This is because background color utility classes do not set anything except background-color.

**Creating Pill Badges**

Similarly, you can create pill shape badges (i.e. badges with more rounded corners) using the .rounded-pill modifier class, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Pill Badges</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

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

<span class="badge rounded-pill bg-secondary">Secondary</span>

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

<span class="badge rounded-pill bg-danger">Danger</span>

<span class="badge rounded-pill bg-warning text-dark">Warning</span>

<span class="badge rounded-pill bg-info text-dark">Info</span>

<span class="badge rounded-pill bg-dark">Dark</span>

<span class="badge rounded-pill bg-light text-dark">Light</span>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Pill Badges](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=pill-badges)

**Showing Counter with Badges**

You can also use badges as part of links or buttons to provide a counter, such as number of received or unread messages, number of notifications etc. They're most commonly found in email clients, application dashboards, social networking websites, etc. Here's an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Badges inside Nav</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<nav class="nav nav-pills">

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

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

<a href="#" class="nav-link active">Messages <span class="badge bg-light text-primary">24</span></a>

<a href="#" class="nav-link">Notification <span class="badge bg-primary">5</span></a>

</nav>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=badges-inside-nav)

**Creating Positioned Badges**

You can use positioning utilities to modify a .badge and position it in the corner of a link or button.

Let's take a look at the following example to understand how it basically works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Positioned Badge</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<button type="button" class="btn btn-primary position-relative">

Inbox

<span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger">50+</span>

</button>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=positioned-badge)

# Bootstrap Progress Bars

In this tutorial you will learn how to create progress bars using Bootstrap.

## Creating Progress Bar with Bootstrap

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

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Progress Bar</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar" style="width: 25%"></div>

</div>

<div class="progress">

<div class="progress-bar" style="width: 50%"></div>

</div>

<div class="progress">

<div class="progress-bar" style="width: 75%"></div>

</div>

<div class="progress">

<div class="progress-bar" style="width: 100%"></div>

</div>

</div>

</body>

</html>

**Note:** The wrapper, i.e. the .progress element indicates the max value of the progress bar, whereas the inner .progress-bar element indicates the progress so far. Also, the .progress-bar requires an inline style, or custom CSS to set their width.

**Creating Progress Bar with Label**

You can also display the progress status as a percentage label just by placing the text within the .progress-bar element, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Progress Bar with Label</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar" style="width: 25%">

25%

</div>

</div>

<div class="progress">

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

50%

</div>

</div>

<div class="progress">

<div class="progress-bar" style="width: 75%">

75%

</div>

</div>

<div class="progress">

<div class="progress-bar" style="width: 100%">

100%

</div>

</div>

</div>

</body>

</html>

f you are showing percentage label you should also apply a [min-width](https://www.tutorialrepublic.com/css-reference/css-min-width-property.php) to the progress bar to ensure that the label text remains readable even for low percentage, as shown here:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Setting Minimum Width fro Bootstrap Progress Bars</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar" style="min-width: 20px;">

0%

</div>

</div>

<div class="progress">

<div class="progress-bar" style="min-width: 20px; width: 2%;">

2%

</div>

</div>

</div>

</body>

</html>

## Setting the Height of Progress Bars

The default height of the progress-bar is 16px, but you can also set its height according to your need by setting the CSS [height](https://www.tutorialrepublic.com/css-reference/css-height-property.php) property on the .progress element, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Height Sizing of Bootstrap Progress Bar</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<!-- Progress bar with 1px height -->

<div class="progress" style="height: 1px;">

<div class="progress-bar" style="width: 50%;"></div>

</div>

<!-- Progress bar with 20px height -->

<div class="progress" style="height: 20px;">

<div class="progress-bar" style="width: 50%;"></div>

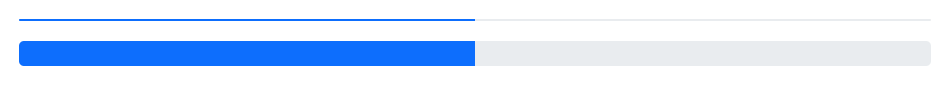
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=progress-bar-height-sizing)

## Creating Stripped Progress Bar

To create the stripped progress bar just add an extra class .progress-bar-striped to the .progress-bar element, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Stripped Progress Bar</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar progress-bar-striped" style="width: 60%;"></div>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Stripped Progress Bar](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stripped-progress-bar)

**Tip:** The stripe is generated via CSS gradient over the progress bar's background color. See the tutorial on [CSS3 Gradients](https://www.tutorialrepublic.com/css-tutorial/css3-gradients.php) to learn how to create gradient colors using CSS.

**Creating Animated Progress Bar**

You can also animate the stripped progress-bar. Add the class .progress-bar-animated to the .progress-bar element, it will animate the stripes from right to left via [CSS3 animations](https://www.tutorialrepublic.com/css-tutorial/css3-animations.php).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Animated Progress Bar</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar progress-bar-striped progress-bar-animated" style="width: 60%"></div>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

**Changing Progress Bar Value Dynamically**

Static progress bars aren't very impressive. The following example will give you a rough idea of how to update the status of a Bootstrap progress bar dynamically using jQuery.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Dynamically Change Bootstrap Progress Bar Value Using jQuery</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

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

</head>

<body>

<div class="m-4">

<!-- Progress bar HTML -->

<div class="progress">

<div class="progress-bar progress-bar-striped" style="min-width: 20px;"></div>

</div>

<!-- jQuery Script -->

<script>

var i = 0;

function makeProgress(){

if(i < 100){

i = i + 1;

$(".progress-bar").css("width", i + "%").text(i + "%");

}

// Wait for sometime before running this script again

setTimeout("makeProgress()", 100);

}

makeProgress();

</script>

</div>

</body>

</html>

## Creating Stacked Progress Bar

You can also place multiple progress bars in a progress component to stack them.

Here's an example that demonstrates how it actually works.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Stacked Progress Bars</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="progress">

<div class="progress-bar bg-success" style="width: 40%">

Program Files (40%)

</div>

<div class="progress-bar bg-warning" style="width: 25%">

Residual Files (25%)

</div>

<div class="progress-bar bg-danger" style="width: 15%">

Junk Files (15%)

</div>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[Bootstrap Stacked Progress Bar](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stacked-progress-bar)

**Creating Progress Bars of Different Colors**

You can additionally use [background color utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) to create progress bars of various colors in order to convey meaning through color, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Progress Bars with Emphasis Classes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<div class="progress">

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

</div>

<div class="progress">

<div class="progress-bar bg-success" style="width: 40%"></div>

</div>

<div class="progress">

<div class="progress-bar bg-warning" style="width: 80%"></div>

</div>

<div class="progress">

<div class="progress-bar bg-danger" style="width: 90%"></div>

</div>

<hr>

<div class="progress">

<div class="progress-bar bg-info" style="width: 20%">

Disk Space: 80% free

</div>

</div>

<div class="progress">

<div class="progress-bar bg-success" style="width: 40%">

Enough Disk Space: 60% free

</div>

</div>

<div class="progress">

<div class="progress-bar bg-warning" style="width: 80%">

Low Disk Space: 80% full

</div>

</div>

<div class="progress">

<div class="progress-bar bg-danger" style="width: 90%">

Not Enough Disk Space: 90% full

</div>

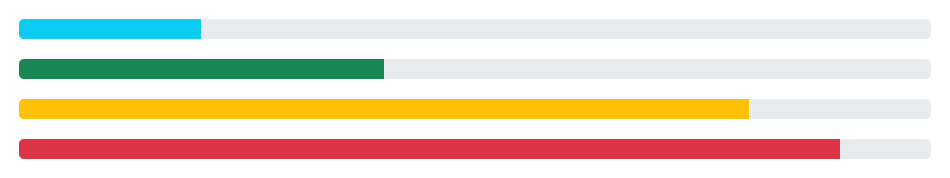
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=progress-bar-with-emphasis)

**Making Striped Progress Bars of Different Colors**

Similar to the solid colors, you can also create different colored striped progress bars using the same background color utility classes. Let's take a look at the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Striped Progress Bars with Emphasis Classes</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

<style>

/\* Adding space at the bottom of progress bar \*/

.progress{

margin-bottom: 1rem;

}

</style>

</head>

<body>

<div class="m-4">

<div class="progress">

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

</div>

<div class="progress">

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

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-warning" style="width: 80%"></div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-danger" style="width: 90%"></div>

</div>

<hr />

<div class="progress">

<div class="progress-bar progress-bar-striped bg-info" style="width: 20%">

Disk Space: 80% free

</div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-success" style="width: 40%">

Enough Disk Space: 60% free

</div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-warning" style="width: 80%">

Low Disk Space: 80% full

</div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-danger" style="width: 90%">

Not Enough Disk Space: 90% full

</div>

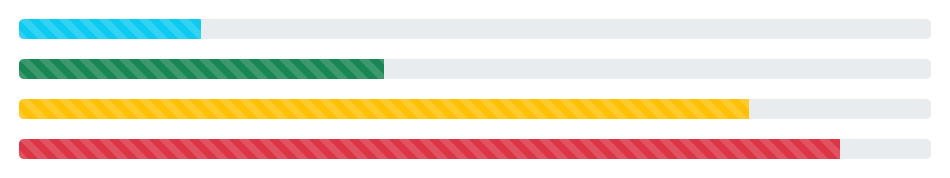
</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=striped-progress-bar-with-emphasis)

# Bootstrap Spinners

In this tutorial you will learn how to use the Bootstrap spinner component.

## Creating the Spinners with Bootstrap

Bootstrap introduces the new spinner component that you can use to show the loading state in your applications. Spinners are typically loading icons and they're built only with HTML and CSS. However, you need some custom JavaScript to show or hide them on a web page.

Moreover, you can easily customize the appearance, alignment, and sizing of the spinners with the pre-defined utility classes. Now, let's see how to create them.

## Creating Border Spinner

You can create a border spinner or a lightweight loading indicator, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Border Spinner</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

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

</head>

<body>

<div class="m-4">

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

The special .visually-hidden class hides an element to all devices except screen readers.

**Creating Colored Spinner**

You can use the [text color utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes) to customize the color of spinners.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

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

<title>Bootstrap Colored Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<div class="spinner-border text-primary">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-secondary">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-success">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-danger">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-warning">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-info">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-dark">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-border text-light">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

**Creating Growing Spinners**

You can also create growing spinners that repeatedly grow and fade out, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Growing Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<div class="spinner-grow">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

Similarly, like the border spinners you can also customize the colors of growing spinners using the Bootstrap's [text color utility classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes), as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Colored Growing Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<div class="spinner-grow text-primary">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-secondary">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-success">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-danger">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-warning">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-info">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-light">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow text-dark">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

— The output of the above example will look something like this:

**Sizing of Spinners**

You can use the class .spinner-border-sm and .spinner-grow-sm to make a smaller spinner that can quickly be used within other components such as buttons.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Small Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<div class="spinner-border spinner-border-sm">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow spinner-grow-sm">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Custom Sizing of Bootstrap Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<div class="spinner-border" style="width: 40px; height: 40px;">

<span class="visually-hidden">Loading...</span>

</div>

<div class="spinner-grow" style="width: 40px; height: 40px;">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</body>

</html>

## Using Spinners within Buttons

You can also use spinners within buttons to indicate an action is currently processing or taking place.

Here's an example where we've placed the spinners inside the disabled buttons.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Placing Bootstrap Spinners inside Buttons</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<!-- Border spinners inside buttons -->

<button class="btn btn-primary" type="button" disabled>

<span class="spinner-border spinner-border-sm"></span>

<span class="visually-hidden">Loading...</span>

</button>

<button class="btn btn-primary" type="button" disabled>

<span class="spinner-border spinner-border-sm"></span>

Loading...

</button>

<!-- Growing spinners inside buttons -->

<button class="btn btn-primary" type="button" disabled>

<span class="spinner-grow spinner-grow-sm"></span>

<span class="visually-hidden">Loading...</span>

</button>

<button class="btn btn-primary" type="button" disabled>

<span class="spinner-grow spinner-grow-sm"></span>

Loading...

</button>

</div>

</body>

</html>

— The output of the above example will look something like this:

  Loading...   Loading...

**Alignment of Spinners**

You can easily align the spinners to left, right or center using the flexbox, text alignment, or float utility classes. Let's try out the following example and see how it actually works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Left, Right and Center Alignment of Bootstrap Spinners</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<!-- Default alignment left -->

<div>

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<hr>

<!-- Center alignment using flex utilities -->

<div class="d-flex justify-content-center">

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<hr>

<!-- Center alignment using text alignment utilities -->

<div class="text-center">

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<hr>

<!-- Right alignment using flex utilities -->

<div class="d-flex justify-content-end">

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<hr>

<!-- Right alignment using text alignment utilities -->

<div class="text-end">

<div class="spinner-border">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<hr>

<!-- Right alignment using float utilities -->

<div class="clearfix">

<div class="spinner-border float-end">

<span class="visually-hidden">Loading...</span>

</div>

</div>

</div>

</body>

</html>

# Bootstrap Jumbotron

In this tutorial you will learn how to create jumbotron in Bootstrap.

## Showcasing Contents with Jumbotron

The Bootstrap jumbotron provides an excellent way to showcase the key content or information on a web page. However, the Bootstrap's default jumbotron component has been dropped from the version 5. But, you can still create custom jumbotrons using the Bootstrap inbuilt utility classes.

Let's take a look at an example to understand how to create a custom jumbotron:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Custom Jumbotron</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container-lg my-4">

<div class="p-5 mb-4 bg-light border rounded-3">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" target="\_blank">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-primary btn-lg">Start learning today</a></p>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

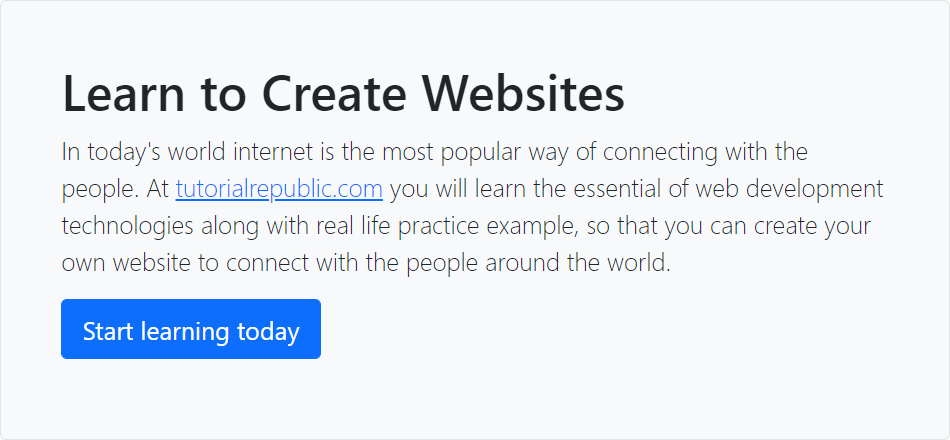
<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

</div>

</body>

</html>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=jumbotron)

**Tip:** The class .p-5 apply the padding of 3rem on all four sides, while the class .mb-4 apply bottom margin of 1.5rem to the jumbotron. Likewise, the classes .border and .rounded-3 applies a border, and a border radius of .3rem on all four sides, respectively. See the [Bootstrap Helper Classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php) chapter to learn more about the various utility classes.

You can also create full width jumbotron with centered content, without rounded corners and borders, by simply wrapping its inner content in .container element, and removing the .border and .rounded-\* class from the wrapper element, as shown in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Full Width Jumbotron</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="py-5 mb-4 bg-light">

<div class="container">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" target="\_blank">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-primary btn-lg">Start learning today</a></p>

</div>

</div>

<div class="container">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Full Width Jumbotron</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="py-5 mb-4 bg-light">

<div class="container">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" target="\_blank">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-primary btn-lg">Start learning today</a></p>

</div>

</div>

<div class="container">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

</div>

</body>

</html>

## Changing the Color Scheme of Jumbotron

Similarly, you can create other variant of jumbotron using the Bootstrap's [color](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes) and [background](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) utility classes. The following example will create a dark variant jumbotron.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Dark Variant of Bootstrap Jumbotron</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container-lg my-4">

<div class="p-5 mb-4 bg-dark text-white rounded-3">

<h1>Learn to Create Websites</h1>

<p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" target="\_blank" class="text-white">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>

<p><a href="https://www.tutorialrepublic.com" target="\_blank" class="btn btn-light btn-lg">Start learning today</a></p>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

</div>

</body>

</html>

**Bootstrap Helper Classes**

In this tutorial you will learn about the Bootstrap helper/utility classes.

**Color Classes**

You can use the contextual color classes to emphasize the text and convey meaning through color. Here's the complete list of available contextual color classes.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .text-primary | Apply blue color (#007bff) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-secondary | Apply grey color (#6c757d) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-success | Apply green color (#28a745) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-danger | Apply red color (#dc3545) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-warning | Apply yellow color (#ffc107) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-info | Apply teal color (#17a2b8) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-light | Apply very light grey color (#f8f9fa) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-dark | Apply dark grey color (#343a40) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-body | Apply Bootstrap's default body text color (#212529) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-muted | Apply light grey color (#6c757d) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-white | Apply white color (#fff) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-black-50 | Apply 50% transparent black color (rgba(0,0,0,.5)) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |
| .text-white-50 | Apply 50% transparent white color (rgba(255,255,255,.5)) on text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=color-classes) |

**Background Color Classes**

Similar to the contextual color classes, you can use the contextual background color classes to set the [background-color](https://www.tutorialrepublic.com/css-tutorial/css-background.php) of an element to apply extra emphasis on them.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .bg-primary | Apply blue background-color (#007bff) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-secondary | Apply grey background-color (#6c757d) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-success | Apply green background-color (#28a745) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-danger | Apply red background-color (#dc3545) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-warning | Apply yellow background-color (#ffc107) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-info | Apply teal background-color (#17a2b8) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-light | Apply very light grey background-color (#f8f9fa) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-dark | Apply dark grey background-color (#343a40) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-white | Apply white background-color (#fff) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |
| .bg-transparent | Apply transparent background-color (transparent) on an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=background-color-classes) |

**Text Alignment Classes**

You can use these text alignment classes to align and format your text content.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .text-start | Align the text to the left. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-center | Align the text to the center. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-end | Align the text to the right. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-wrap | Wrap the overflowing text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-nowrap | Prevent text from wrapping. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-truncate | Truncate the text with an ellipsis. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-break | Break the long words to prevent overflow. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-lowercase | Transform the text to lowercase. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-uppercase | Transform the text to uppercase. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-capitalize | Capitalize the first letter of each word. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fw-bold | Set the font-weight of an element to **bold**. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fw-bolder | Set the font-weight of an element to **bolder** (relative to the parent). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fw-normal | Set the font-weight of an element to normal. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fw-light | Set the light font-weight for an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fw-lighter | Set the lighter font-weight for an element (relative to the parent). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fst-italic | Set the font-style of an element to *italic*. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .fst-normal | Set the font-style of an element to normal. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .font-monospace | Set the font-family of an element to monospace font (fixed-width) | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-reset | Reset the color of a text or link (inherits the color from its parent). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-decoration-none | Remove the text decoration such as underline from a text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-decoration-underline | Add underline to text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |
| .text-decoration-line-through | Add line through the middle of text. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=text-classes) |

**Border Classes**

You can use the border utility classes to quickly style the border and border-radius of an element. They are very useful for images, buttons, or any other element.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .border | Add a border on all sides of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-additive-classes) |
| .border-top | Add a border on the top side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-additive-classes) |
| .border-end | Add a border on the right side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-additive-classes) |
| .border-bottom | Add a border on the bottom side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-additive-classes) |
| .border-start | Add a border on the left side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-additive-classes) |
| .border-0 | Remove the border from all sides of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-subtractive-classes) |
| .border-top-0 | Remove the border from top side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-subtractive-classes) |
| .border-end-0 | Remove the border from right side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-subtractive-classes) |
| .border-bottom-0 | Remove the border from bottom side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-subtractive-classes) |
| .border-start-0 | Remove the border from left side of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-subtractive-classes) |
| .border-primary | Change the border-color of an element to blue (#007bff). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-secondary | Change the border-color of an element to grey (#6c757d). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-success | Change the border-color of an element to green (#28a745). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-danger | Change the border-color of an element to red (#dc3545). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-warning | Change the border-color of an element to yellow (#ffc107). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-info | Change the border-color of an element to teal (#17a2b8). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-light | Change the border-color of an element to very light grey (#f8f9fa). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-dark | Change the border-color of an element to dark grey (#343a40). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .border-white | Change the border-color of an element to white (#fff). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-color-classes) |
| .rounded | Round all corners of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-top | Round top-left and top-right corners of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-end | Round top-right and bottom-right corners of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-bottom | Round bottom-left and bottom-right corners of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-start | Round top-left and bottom-left corners of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-circle | Round an element into circle shape. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-pill | Round an element into pill shape. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-0 | Remove round corners from an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-1 | Set border-radius of 0.2rem for an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-2 | Set border-radius of 0.25rem for an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |
| .rounded-3 | Set border-radius of 0.3rem for an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=border-radius-classes) |

**Spacing Classes**

You can use the spacing utility classes to quickly adjust the margin and padding of an element.

The # character in the classes .mt-#, .pt-#, etc. represents the size of the margin or padding. The value of the size can be an integer from 0 to 5, or auto.

The size 0 completely removes the margin or padding by setting it to 0, whereas the value auto sets the margin to auto. The integer numbers from 1 to 5 sets the margin or padding incrementally (ranging from 0.25rem to 3rem). Try out the examples to see how it works:

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .mt-# | Set the top margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=margin-top-classes) |
| .mb-# | Set the bottom margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=margin-bottom-classes) |
| .ms-# | Set the left margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=margin-left-classes) |
| .me-# | Set the right margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=margin-right-classes) |
| .mx-# | Set the left and right margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=horizontal-margin-classes) |
| .my-# | Set the top and bottom margin of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-margin-classes) |
| .pt-# | Set the top padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=padding-top-classes) |
| .pb-# | Set the bottom padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=padding-bottom-classes) |
| .ps-# | Set the left padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=padding-left-classes) |
| .pe-# | Set the right padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=padding-right-classes) |
| .px-# | Set the left and right padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=horizontal-padding-classes) |
| .py-# | Set the top and bottom padding of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-padding-classes) |

**Tip:** You can also apply negative margins using the margin utility classes. Just append the size number with the character n, like .mt-n1, .mb-n2, and so on.

**Float Classes**

You can use the float utility classes to quickly float an element to the left or right or remove float from an element altogether. Here's the list of available float classes.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .float-start | Float an element to the left. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=float-classes) |
| .float-end | Float an element to the right. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=float-classes) |
| .float-none | Disable floating from an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=float-classes) |
| .clearfix | Clear floats to prevent parent element from collapsing. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=float-classes) |
| .float-\*-start | Float an element to the left on based on the viewport size. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=responsive-float-left-classes) |
| .float-\*-end | Float an element to the right on based on the viewport size. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=responsive-float-right-classes) |
| .float-\*-none | Disable floating from an element on based on the viewport size. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=responsive-float-none-classes) |

**Tip:** The \* character in the classes .float-\*-left and .float-\*-right represents responsive breakpoint which can be sm, md, lg, xl, or xxl. See [Bootstrap grid system](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php) for more detail.

**Shadow Classes**

You can use the shadow utility classes to quickly add or remove shadows to elements.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .shadow | Add a shadow to an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=shadow-classes) |
| .shadow-sm | Add a small shadow to an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=shadow-classes) |
| .shadow-lg | Add a larger shadow to an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=shadow-classes) |
| .shadow-none | Remove shadow from an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=shadow-classes) |

**Sizing Classes**

You can use the sizing utility classes to easily make an element as wide or as tall.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .w-25 | Set the width of an element to 25% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=width-classes) |
| .w-50 | Set the width of an element to 50% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=width-classes) |
| .w-75 | Set the width of an element to 75% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=width-classes) |
| .w-100 | Set the width of an element to 100% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=width-classes) |
| .w-auto | Set the width of an element to auto, i.e. the browser will calculate and select a width for the specified element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=width-classes) |
| .mw-100 | Set the max-width of an element to 100%. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=max-width-and-height-classes) |
| .h-25 | Set the height of an element to 25% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=height-classes) |
| .h-50 | Set the height of an element to 50% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=height-classes) |
| .h-75 | Set the height of an element to 75% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=height-classes) |
| .h-100 | Set the height of an element to 100% of its parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=height-classes) |
| .h-auto | Set the height of an element to auto, i.e. the browser will calculate and select a height for the specified element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=height-classes) |
| .mh-100 | Set the max-height of an element to 100%. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=max-width-and-height-classes) |
| .vw-100 | Set the width of an element to 100% of the width of the viewport. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=viewport-width-classes) |
| .min-vw-100 | Set the min-width of an element to 100% of the width of the viewport. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=viewport-width-classes) |
| .vh-100 | Set the height of an element to 100% of the height of the viewport. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=viewport-height-classes) |
| .min-vh-100 | Set the min-height of an element to 100% of the height of the viewport. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=viewport-height-classes) |

**Vertical Alignment Classes**

You can use the vertical alignment utility classes to quickly change the vertical alignment of inline, inline-block, inline-table, and table cell elements.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .align-baseline | Align the baseline of the element with the baseline of its parent. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |
| .align-top | Align the top of the element with the top of the entire line. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |
| .align-middle | Align the element in the middle of the parent element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |
| .align-bottom | Align the bottom of the element with the bottom of the entire line. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |
| .align-text-top | Align the top of the element with the top of the parent element's font. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |
| .align-text-bottom | Align the bottom of the element with the bottom of the parent element's font. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertical-align-classes) |

**Position Classes**

You can use the position utility classes to quickly configuring the position of an element.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .position-static | Set the position of element to static. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=position-static-class) |
| .position-relative | Set the position of element to relative. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=position-relative-class) |
| .position-absolute | Set the position of element to absolute. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=position-absolute-class) |
| .position-fixed | Set the position of element to fixed. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=position-fixed-class) |
| .position-sticky | Set the position of element to sticky. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=position-sticky-class) |
| .fixed-top | Position an element at the top of the viewport, from edge to edge. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=fixed-top-class) |
| .fixed-bottom | Position an element at the bottom of the viewport, from edge to edge. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=fixed-bottom-class) |
| .sticky-top | Position an element at the top of the viewport, from edge to edge, but that will only happen after you scroll past it. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=sticky-top-class) |

**Display Classes**

You can use the display utility classes to quickly and responsively toggle the display value of an element. You can also control an element's display when printing the page.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .d-none | Hide an element. It does not take up any space in the layout. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-none-classes) |
| .d-inline | Force an element to generate an inline-level box. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-inline-classes) |
| .d-inline-block | Force an element to generate a block box that will be flowed with surrounding content i.e. in the same line as adjacent content like inline box. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-inline-block-classes) |
| .d-block | Force an element to generate a block-level box. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-block-classes) |
| .d-table | Force an element to behave like a [<table>](https://www.tutorialrepublic.com/html-reference/html-table-tag.php) element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-table-classes) |
| .d-table-row | Force an element to behave like a [<tr>](https://www.tutorialrepublic.com/html-reference/html-tr-tag.php) element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-table-classes) |
| .d-table-cell | Force an element to behave like a [<td>](https://www.tutorialrepublic.com/html-reference/html-td-tag.php) element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-table-classes) |
| .d-flex | Force an element to generate a block-level [flex container box](https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php). | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-flex-classes) |
| .d-inline-flex | Force an element to generate an inline-level flex container box. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=display-inline-flex-classes) |

You can also use the responsive breakpoints (sm, md, lg, xl, and xxl.) with these display utility classes to show or hide elements depending on viewport width using the format .d-{breakpoint}-{value}, for example, .d-sm-none, .d-md-block, .d-lg-block, .d-xl-block, .d-xxl-block, and so on.

For a complete set of flex utility classes, please check out the [Bootstrap flex reference page](https://getbootstrap.com/docs/5.0/utilities/flex/).

**Tip:** You can create print variant of the display classes to change the display value of elements when printing by simply inserting the string print- after .d-, for example, .d-print-none, .d-print-inline, .d-print-block, .d-print-table, and so on.

**Miscellaneous Classes**

Here's the list some more utility classes that can used for various purposes.

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Test** |
| .btn-close | Create generic close icon for dismissing components like modals and alerts. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=close-class) |
| .overflow-auto | Provide scrollbars to view overflowed content of an element if required. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=overflow-classes) |
| .overflow-hidden | Hide overflowed content of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=overflow-classes) |
| .overflow-visible | Show overflowed content of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=overflow-classes) |
| .overflow-scroll | Provide scrollbars to view overflowed content of an element. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=overflow-classes) |
| .stretched-link | Make an element clickable by stretching a nested link. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stretched-link-class) |
| .visible | Make an element visible, without modifying the display. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=visibility-classes) |
| .invisible | Make an element invisible, but it still take up space in the page. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=visibility-classes) |
| .visually-hidden | Hide an element to all devices except screen readers. | [Example](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=visually-hidden-class) |

**Bootstrap Modals**

In this tutorial you will learn how to create modals with Bootstrap.

**Creating Modals with Bootstrap**

Modal is basically a dialog box or popup window that is used to provide important information to the user or prompt user to take necessary actions before moving on. Modals are widely used to warn users for situations like session time out or to receive their final confirmation before going to perform any critical actions such as saving or deleting important data.

You can easily create very smart and flexible dialog boxes with the Bootstrap modal plugin. The following example oulines the basic structure to create a simple modal with a header, message body and the footer containing action buttons for the user.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-modal-on-page-load-using-jquery)

<!-- jQuery Code (to Show Modal on Page Load) -->

<script>

$(document).ready(function(){

$("#myModal").modal("show");

});

</script>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

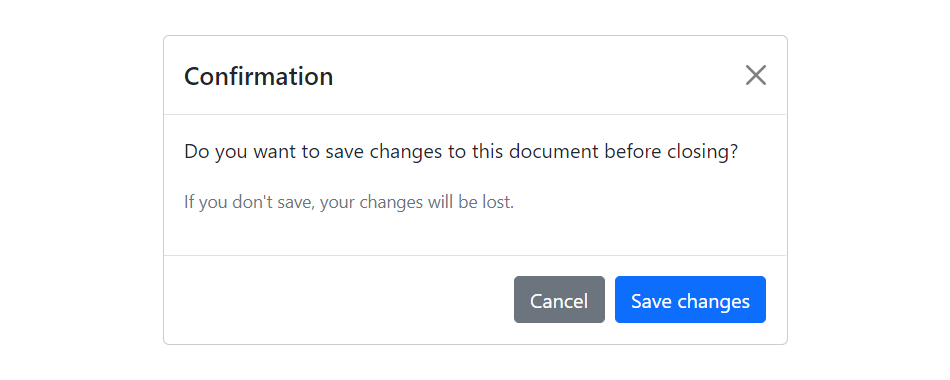
</div>

</div>

</div>

</div>

— If you try out the above example, it will launches the modal window automatically when the DOM is fully loaded via JavaScript. The output will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=modal)

**Tip:** Always try to place your modal HTML in a top-level position in your document, preferably before closing of the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) tag (i.e. </body>) to avoid interference from other elements, otherwise it may affect modal's appearance or functionality.

Check out the snippets section for examples of some [beautifully designed Bootstrap modals](https://www.tutorialrepublic.com/snippets/gallery.php?tag=modal).

**Activate Modals via Data Attributes**

You can activate a Bootstrap modal by clicking on the button or link via data attributes without writing any JavaScript code. Take a look at the following example to see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-modals-via-data-attribute)

<!-- Button HTML (to Trigger Modal) -->

<a href="#myModal"class="btn btn-lg btn-primary"data-bs-toggle="modal">Launch Demo Modal</a>

<!-- Modal HTML -->

<div id="myModal"class="modal fade">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

The above example launches the modal window on click of the "Launch Demo Modal" button. Let's go through each part of this modal code one by one for a better understanding.

**Explanation of Code**

To activate a Bootstrap modal via data attributes we basically need two components — the controller element like a button or link, and the modal element itself.

* The outermost container of every modal in a document must have a unique id (in this case id="myModal", *line no-5*), so that it can be targeted via data-bs-target (for buttons) or href (for hyperlinks) attribute of the controller element (*line no-2*).
* The attribute data-bs-toggle="modal" is required to add on the controller element (*line no-2*), like a button or an anchor, along with a attribute data-bs-target="#myModal" or href="#myModal" to target a specific modal to toggle.
* The .modal-dialog class (*line no-6*) sets the width as well as horizontal and vertical alignment of the modal box. Whereas the class .modal-content sets the styles like text and background color, borders, rounded corners etc.

Rest of the thing is self explanatory, such as the .modal-header element defines a header for the modal that usually contains a modal title and a close button, whereas the .modal-body element contains the actual content like text, images, forms etc. and the .modal-footer element defines the footer that typically contains action buttons for the user.

**Note:** The .fade class on the .modal element adds a fading and sliding animation effect while showing and hiding the modal window. If you want the modal that simply appear without any effect you can just remove this class. Also, when modals become too long for the user's viewport or device, they scroll independent of the page itself.

**Activate Modals via JavaScript**

You may also activate a Bootstrap modal window via JavaScript — just call the modal() Bootstrap method with the modal id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-modals-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myModal").modal("show");

});

});

</script>

**Changing the Size of Modals**

Bootstrap gives you option further to scaling a modal up or down. You can create small, large, as well as extra-large modals by adding an extra class .modal-sm, .modal-lg, and .modal-xl class, respectively on the .modal-dialog. Here's an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=large-and-small-modal-sizes)

<!-- Extra Large modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#extraLargeModal">Extra Large modal</button>

<div id="extraLargeModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-xl">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Extra Large Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-xl</code> class on <code>.modal-dialog</code> to create this extra large modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

<!-- Large modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#largeModal">Large modal</button>

<div id="largeModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Large Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-lg</code> class on <code>.modal-dialog</code> to create this large modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

<!-- Small modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#smallModal">Small modal</button>

<div id="smallModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-sm">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Small Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-sm</code> class on <code>.modal-dialog</code> to create this small modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

**Tip:** The maximum width of the default modal will be 500px, whereas the maximum width of the small, large, and extra-large modal will be 300px, 800px, 1140px, respectivley.

**Changing Modal Content Based on Trigger Button**

Often several modal on a web page has almost same content with minor differences.

You can use the [modal events](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#modal-events) to create slightly different modal windows based on the same modal HTML. The following example will show you how to change the title of the modal window according to the trigger button's data-title attribute value.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=change-modal-content-based-on-trigger-button-using-jquery)

<script>

$(document).ready(function(){

$("#myModal").on("show.bs.modal",function(event){

// Get the button that triggered the modal

var button =$(event.relatedTarget);

// Extract value from the custom data-\* attribute

var titleData = button.data("title");

// Change modal title

$(this).find(".modal-title").text(titleData);

});

});

</script>

**Creating Vertically Centered Modal**

Simply add the class .modal-dialog-centered to .modal-dialog element to vertically center the modal. If modal has long content you can additionally apply the class .modal-dialog-scrollable on .modal-dialog to make the modal body scrollable. Here's an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-centering-the-modal)

<!-- Button HTML (to Trigger Modal) -->

<a href="#modalCenter"role="button"class="btn btn-primary"data-bs-toggle="modal">Vertically Centered Modal</a>

<!-- Modal HTML -->

<div id="modalCenter"class="modal fade"tabindex="-1">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Vertical Alignment Demo</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Lorem ipsum dolor sit amet, consectetur elit...</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Close</button>

<button type="button"class="btn btn-primary">OK, Got it!</button>

</div>

</div>

</div>

</div>

**Using the Grid inside Modals**

You can also utilize the [Bootstrap grid system](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php) to create grid layouts within a modal. Simply, use the .row class to create rows and use .col-\*, .col-sm-\*, .col-md-\*, .col-lg-\* and .col-xl-\* classes to create columns within the .modal-body. Let's check out an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-row-columns-inside-modal)

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Send Us a Message</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<div class="row">

<div class="col-6">

<div class="form-group">

<label>First Name</label>

<input type="text"class="form-control">

</div>

</div>

<div class="col-6">

<div class="form-group">

<label>Last Name</label>

<input type="text"class="form-control">

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="form-group">

<label>Email</label>

<input type="email"class="form-control">

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="form-group">

<label>Comments</label>

<textarea class="form-control"></textarea>

</div>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Send Email</button>

</div>

</div>

</div>

</div>

**Loading Content in Modal via Ajax**

You can also load remote content inside the Bootstrap modal via Ajax.

In the following example content inside the modal body will be inserted from a remote file upon activation using the jQuery load() method and Bootstrap show.bs.modal event.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=loading-content-in-modal-body-via-ajax-using-jquery)

<!-- jQuery Code (to Load Content via Ajax) -->

<script>

$(document).ready(function(){

$("#myModal").on("show.bs.modal",function(){

// Place the returned HTML into the selected element

$(this).find(".modal-body").load("remote.php");

});

});

</script>

<!-- Button HTML (to Trigger Modal) -->

<button type="button"class="btn btn-lg btn-primary"data-bs-toggle="modal"data-bs-target="#myModal">Launch Demo Modal</button>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Ajax Loading Demo</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<!-- Content will be loaded here from "remote.php" file -->

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Close</button>

<button type="button"class="btn btn-primary">OK, Got it!</button>

</div>

</div>

</div>

</div>

**Tip:** You can also place [tooltips](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php) and [popovers](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php) within the modals as needed. When modals are closed, any tooltips and popovers within are also automatically dismissed.

**Options**

There are certain options which can be passed to modal() Bootstrap method to customize the functionality of a modal. Options can be passed via data attributes or JavaScript.

For setting the modals options via data attributes, just append the option name to data-bs-, such as data-bs-backdrop="static", data-bs-keyboard="false", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| backdrop | boolean or the string 'static' | true | Includes a modal-backdrop (black overlay area) element. Alternatively, you may specify static for a backdrop which doesn't close the modal on click. |
| keyboard | boolean | true | Closes the modal window on press of escape key. |
| focus | boolean | true | Puts the focus on the modal when initialized. |
| show | boolean | true | Shows the modal when initialized or activate. |

Data attributes provides an easy way for setting the modal options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#passing-options) examples in the methods section below to know how to set the options for modals using JavaScript.

In the following example we've set the backdrop option to static (*line no-5*) which prevents the modal from closing when clicking outside of the modal i.e. the black overlay area.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=modal-with-static-backdrop)

<!-- Button HTML (to Trigger Modal) -->

<button type="button"class="btn btn-lg btn-primary"data-bs-target="#myModal"data-bs-toggle="modal">Launch Demo Modal</button>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"data-bs-backdrop="static"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

**Methods**

These are the standard bootstrap's modals methods:

**Passing options**

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#modal-options) to the modal using options object.

Let's try out the following example which will prevent the modal from closing when a user clicks on the backdrop (i.e. the black overlay area behind the modal).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-hiding-modal-on-backdrop-click-using-jquery)

<script>

$(document).ready(function(){

$("#myModal").modal({

backdrop:"static"

});

});

</script>

The following example will prevent the modal from closing on press of the escape key.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=prevent-hiding-modal-on-escape-key-press-using-jquery)

<script>

$(document).ready(function(){

$("#myModal").modal({

keyboard:false

});

});

</script>

**toggle**

This method can be used to toggle a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toggle-modal-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myModal").modal("toggle");

});

});

</script>

**show**

This method can be used to open a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-modal-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myModal").modal("show");

});

});

</script>

**hide**

This method can be used to hide a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=hide-modal-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myModal").modal("hide");

});

});

</script>

**handleUpdate**

This method readjusts the modal's position to counter the jerk that is occurring due to the appearance of the viewport scrollbar in case if the modal height changes in such a way that it becomes higher than the viewport height while it is open.

A common example of this scenario is showing the hidden elements inside the modal via JavaScript or loading content inside the modal using Ajax after activation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=handle-modal-update-using-jquery)

<script>

$(document).ready(function(){

$("#showText").click(function(){

$("#textBlock").show();

$("#myModal").modal("handleUpdate");

});

});

</script>

**dispose**

This method destroys an element's modal (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-modal-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myModal = bootstrap.Modal.getInstance($("#myModal")[0]);

console.log(myModal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

$("#myModal").modal("dispose");

console.log(myModal);

// {\_element: null, \_config: null, \_dialog: null, \_backdrop: null, \_isShown: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the modal instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-modal-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var modal = bootstrap.Modal.getInstance($("#myModal")[0]);

console.log(modal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the modal instance associated with a DOM element, or create a new one in case if the modal wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-modal-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var modal = bootstrap.Modal.getOrCreateInstance($("#myModal")[0]);

console.log(modal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

**Events**

Bootstrap's modal class includes few events for hooking into modal functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.modal | This event fires immediately when the show instance method is called. |
| shown.bs.modal | This event is fired when the modal has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.modal | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.modal | This event is fired when the modal has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hidePrevented.bs.modal | This event is fired when the modal is shown, its backdrop option is set to static and a click outside the modal performed, or keyboard option is set to false and an escape key press is performed. |

The following example displays an alert message to the user when fade out transition of the modal window has been fully completed.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listen-to-hidden-bs-modal-event-using-jquery)

<script>

$(document).ready(function(){

$("#myModal").on("hidden.bs.modal",function(){

alert("Modal window has been completely closed.");

});

});

</script>

The following example displays an alert message if you try to close the modal by clicking the dark area.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listen-to-hideprevented-bs-modal-event-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal = document.getElementById("myModal");

myModal.addEventListener("hidePrevented.bs.modal",function(){

alert("Modal can't be closed by clicking outside, because the backdrop option is set to static.");

});

});

</script>

**Tip:** See also the [Bootstrap FAQ](https://www.tutorialrepublic.com/faq.php#bootstrap-less) section for more examples on modals, like setting vertical alignment, changing default width, embedding YouTube videos, etc.

**Bootstrap Modals**

In this tutorial you will learn how to create modals with Bootstrap.

**Creating Modals with Bootstrap**

Modal is basically a dialog box or popup window that is used to provide important information to the user or prompt user to take necessary actions before moving on. Modals are widely used to warn users for situations like session time out or to receive their final confirmation before going to perform any critical actions such as saving or deleting important data.

You can easily create very smart and flexible dialog boxes with the Bootstrap modal plugin. The following example oulines the basic structure to create a simple modal with a header, message body and the footer containing action buttons for the user.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-modal-on-page-load-using-javascript)

<!-- JavaScript Code (to Show Modal on Page Load) -->

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal =newbootstrap.Modal(document.getElementById("myModal"));

myModal.show();

});

</script>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

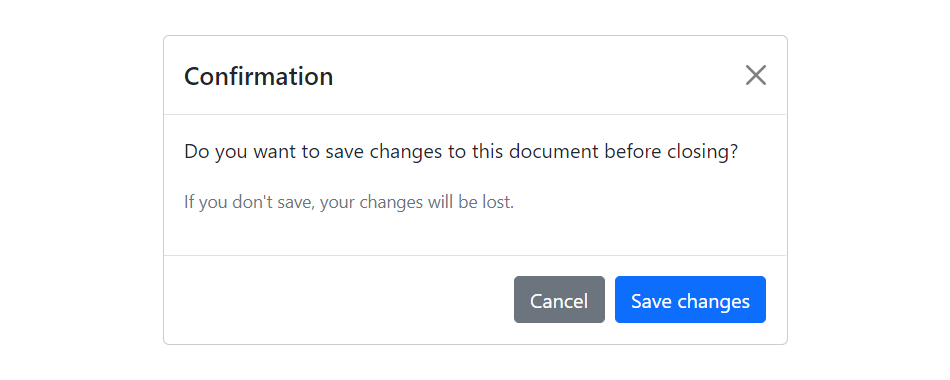
</div>

</div>

</div>

</div>

— If you try out the above example, it will launches the modal window automatically when the DOM is fully loaded via JavaScript. The output will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=modal)

**Tip:** Always try to place your modal HTML in a top-level position in your document, preferably before closing of the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) tag (i.e. </body>) to avoid interference from other elements, otherwise it may affect modal's appearance or functionality.

Check out the snippets section for examples of some [beautifully designed Bootstrap modals](https://www.tutorialrepublic.com/snippets/gallery.php?tag=modal).

**Activate Modals via Data Attributes**

You can activate a Bootstrap modal by clicking on the button or link via data attributes without writing any JavaScript code. Take a look at the following example to see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-modals-via-data-attribute)

<!-- Button HTML (to Trigger Modal) -->

<a href="#myModal"class="btn btn-lg btn-primary"data-bs-toggle="modal">Launch Demo Modal</a>

<!-- Modal HTML -->

<div id="myModal"class="modal fade">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

The above example launches the modal window on click of the "Launch Demo Modal" button. Let's go through each part of this modal code one by one for a better understanding.

**Explanation of Code**

To activate a Bootstrap modal via data attributes we basically need two components — the controller element like a button or link, and the modal element itself.

* The outermost container of every modal in a document must have a unique id (in this case id="myModal", *line no-5*), so that it can be targeted via data-bs-target (for buttons) or href (for hyperlinks) attribute of the controller element (*line no-2*).
* The attribute data-bs-toggle="modal" is required to add on the controller element (*line no-2*), like a button or an anchor, along with a attribute data-bs-target="#myModal" or href="#myModal" to target a specific modal to toggle.
* The .modal-dialog class (*line no-6*) sets the width as well as horizontal and vertical alignment of the modal box. Whereas the class .modal-content sets the styles like text and background color, borders, rounded corners etc.

Rest of the thing is self explanatory, such as the .modal-header element defines a header for the modal that usually contains a modal title and a close button, whereas the .modal-body element contains the actual content like text, images, forms etc. and the .modal-footer element defines the footer that typically contains action buttons for the user.

**Note:** The .fade class on the .modal element adds a fading and sliding animation effect while showing and hiding the modal window. If you want the modal that simply appear without any effect you can just remove this class. Also, when modals become too long for the user's viewport or device, they scroll independent of the page itself.

**Activate Modals via JavaScript**

You may also activate a Bootstrap modal window via JavaScript — just call the modal() Bootstrap method with the modal id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-modals-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

btn.addEventListener("click",function(){

var myModal =newbootstrap.Modal(document.getElementById("myModal"));

myModal.show();

});

});

</script>

**Changing the Size of Modals**

Bootstrap gives you option further to scaling a modal up or down. You can create small, large, as well as extra-large modals by adding an extra class .modal-sm, .modal-lg, and .modal-xl class, respectively on the .modal-dialog. Here's an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=large-and-small-modal-sizes)

<!-- Extra Large modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#extraLargeModal">Extra Large modal</button>

<div id="extraLargeModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-xl">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Extra Large Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-xl</code> class on <code>.modal-dialog</code> to create this extra large modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

<!-- Large modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#largeModal">Large modal</button>

<div id="largeModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Large Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-lg</code> class on <code>.modal-dialog</code> to create this large modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

<!-- Small modal -->

<button class="btn btn-primary"data-bs-toggle="modal"data-bs-target="#smallModal">Small modal</button>

<div id="smallModal"class="modal fade"tabindex="-1"role="dialog">

<div class="modal-dialog modal-sm">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Small Modal</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Add the <code>.modal-sm</code> class on <code>.modal-dialog</code> to create this small modal.</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">OK</button>

</div>

</div>

</div>

</div>

**Tip:** The maximum width of the default modal will be 500px, whereas the maximum width of the small, large, and extra-large modal will be 300px, 800px, 1140px, respectivley.

**Changing Modal Content Based on Trigger Button**

Often several modal on a web page has almost same content with minor differences.

You can use the [modal events](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#modal-events) to create slightly different modal windows based on the same modal HTML. The following example will show you how to change the title of the modal window according to the trigger button's data-title attribute value.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=change-modal-content-based-on-trigger-button-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal = document.getElementById("myModal");

myModal.addEventListener("show.bs.modal",function(event){

// Get the button that triggered the modal

var button = event.relatedTarget;

// Extract value from the custom data-\* attribute

var titleData = button.getAttribute("data-title");

// Change modal title

myModal.querySelector(".modal-title").innerText = titleData;

});

});

</script>

**Creating Vertically Centered Modal**

Simply add the class .modal-dialog-centered to .modal-dialog element to vertically center the modal. If modal has long content you can additionally apply the class .modal-dialog-scrollable on .modal-dialog to make the modal body scrollable. Here's an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-centering-the-modal)

<!-- Button HTML (to Trigger Modal) -->

<a href="#modalCenter"role="button"class="btn btn-primary"data-bs-toggle="modal">Vertically Centered Modal</a>

<!-- Modal HTML -->

<div id="modalCenter"class="modal fade"tabindex="-1">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Vertical Alignment Demo</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Lorem ipsum dolor sit amet, consectetur elit...</p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Close</button>

<button type="button"class="btn btn-primary">OK, Got it!</button>

</div>

</div>

</div>

</div>

**Using the Grid inside Modals**

You can also utilize the [Bootstrap grid system](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php) to create grid layouts within a modal. Simply, use the .row class to create rows and use .col-\*, .col-sm-\*, .col-md-\*, .col-lg-\* and .col-xl-\* classes to create columns within the .modal-body. Let's check out an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-row-columns-inside-modal)

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Send Us a Message</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<div class="row">

<div class="col-6">

<div class="form-group">

<label>First Name</label>

<input type="text"class="form-control">

</div>

</div>

<div class="col-6">

<div class="form-group">

<label>Last Name</label>

<input type="text"class="form-control">

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="form-group">

<label>Email</label>

<input type="email"class="form-control">

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="form-group">

<label>Comments</label>

<textarea class="form-control"></textarea>

</div>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Send Email</button>

</div>

</div>

</div>

</div>

**Loading Content in Modal via Ajax**

You can also load remote content inside the Bootstrap modal via Ajax.

In the following example content inside the modal body will be inserted from a remote file upon activation using the jQuery load() method and Bootstrap show.bs.modal event.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=loading-content-in-modal-body-via-ajax-using-javascript)

<!-- JavaScript Code (to Load Content via Ajax) -->

<script>

// Define function to fetch content from a remote file

functionloadContent(url){

var httpRequest =newXMLHttpRequest();

httpRequest.onreadystatechange=function(){

// Processing the server response

if(httpRequest.readyState === XMLHttpRequest.DONE){

if(httpRequest.status ===200){

updateModal(httpRequest.responseText);

}else{

alert("There was a problem with the request.");

}

}

};

httpRequest.open("GET", url,true);

httpRequest.send();

};

// Define function to update modal based on response

functionupdateModal(response){

var myModal = document.getElementById("myModal");

myModal.querySelector(".modal-body").innerHTML = response;

}

// Load content in modal upon activation

document.addEventListener("DOMContentLoaded",function(){

myModal.addEventListener("show.bs.modal",function(){

loadContent("remote.php");

});

});

</script>

<!-- Button HTML (to Trigger Modal) -->

<button type="button"class="btn btn-lg btn-primary"data-bs-toggle="modal"data-bs-target="#myModal">Launch Demo Modal</button>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Ajax Loading Demo</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<!-- Content will be loaded here from "remote.php" file -->

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Close</button>

<button type="button"class="btn btn-primary">OK, Got it!</button>

</div>

</div>

</div>

</div>

**Tip:** You can also place [tooltips](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php) and [popovers](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php) within the modals as needed. When modals are closed, any tooltips and popovers within are also automatically dismissed.

**Options**

There are certain options which can be passed to modal() Bootstrap method to customize the functionality of a modal. Options can be passed via data attributes or JavaScript.

For setting the modals options via data attributes, just append the option name to data-bs-, such as data-bs-backdrop="static", data-bs-keyboard="false", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| backdrop | boolean or the string 'static' | true | Includes a modal-backdrop (black overlay area) element. Alternatively, you may specify static for a backdrop which doesn't close the modal on click. |
| keyboard | boolean | true | Closes the modal window on press of escape key. |
| focus | boolean | true | Puts the focus on the modal when initialized. |
| show | boolean | true | Shows the modal when initialized or activate. |

Data attributes provides an easy way for setting the modal options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#passing-options) examples in the methods section below to know how to set the options for modals using JavaScript.

In the following example we've set the backdrop option to static (*line no-5*) which prevents the modal from closing when clicking outside of the modal i.e. the black overlay area.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=modal-with-static-backdrop)

<!-- Button HTML (to Trigger Modal) -->

<button type="button"class="btn btn-lg btn-primary"data-bs-target="#myModal"data-bs-toggle="modal">Launch Demo Modal</button>

<!-- Modal HTML -->

<div id="myModal"class="modal fade"data-bs-backdrop="static"tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

<button type="button"class="btn-close"data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<p>Do you want to save changes to this document before closing?</p>

<p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>

</div>

<div class="modal-footer">

<button type="button"class="btn btn-secondary"data-bs-dismiss="modal">Cancel</button>

<button type="button"class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

**Methods**

These are the standard bootstrap's modals methods:

**Passing options**

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php#modal-options) to the modal using options object.

Let's try out the following example which will prevent the modal from closing when a user clicks on the backdrop (i.e. the black overlay area behind the modal).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-hiding-modal-on-backdrop-click-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal =newbootstrap.Modal(document.getElementById("myModal"),{

backdrop:"static"

});

});

</script>

The following example will prevent the modal from closing on press of the escape key.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=prevent-hiding-modal-on-escape-key-press-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal =newbootstrap.Modal(document.getElementById("myModal"),{

keyboard:false

});

});

</script>

**toggle**

This method can be used to toggle a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toggle-modal-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myModal").modal("toggle");

});

});

</script>

**show**

This method can be used to open a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-modal-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var myModal =newbootstrap.Modal(document.getElementById("myModal"));

btn.addEventListener("click",function(){

myModal.show();

});

});

</script>

**hide**

This method can be used to hide a modal window manually.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=hide-modal-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal =newbootstrap.Modal(document.getElementById("myModal"));

var btn = document.getElementById("myBtn");

btn.addEventListener("click",function(){

myModal.hide();

});

});

</script>

**handleUpdate**

This method readjusts the modal's position to counter the jerk that is occurring due to the appearance of the viewport scrollbar in case if the modal height changes in such a way that it becomes higher than the viewport height while it is open.

A common example of this scenario is showing the hidden elements inside the modal via JavaScript or loading content inside the modal using Ajax after activation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=handle-modal-update-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("showText");

btn.addEventListener("click",function(){

document.getElementById("textBlock").style.display ="block";

var myModal = bootstrap.Modal.getInstance(document.getElementById("myModal"));

myModal.handleUpdate();

});

});

</script>

**dispose**

This method destroys an element's modal (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-modal-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

btn.addEventListener("click",function(){

var myModal = bootstrap.Modal.getInstance(document.getElementById("myModal"));

console.log(myModal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

myModal.dispose();

console.log(myModal);

// {\_element: null, \_config: null, \_dialog: null, \_backdrop: null, \_isShown: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the modal instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-modal-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

btn.addEventListener("click",function(){

var myModal = document.getElementById("myModal");

var modal = bootstrap.Modal.getInstance(myModal);

console.log(modal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the modal instance associated with a DOM element, or create a new one in case if the modal wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-modal-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

btn.addEventListener("click",function(){

var myModal = document.getElementById("myModal");

var modal = bootstrap.Modal.getOrCreateInstance(myModal);

console.log(modal);

// {\_element: div#myModal.modal.fade, \_config: {…}, \_dialog: div.modal-dialog, \_backdrop: ke, \_isShown: false, …}

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

**Events**

Bootstrap's modal class includes few events for hooking into modal functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.modal | This event fires immediately when the show instance method is called. |
| shown.bs.modal | This event is fired when the modal has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.modal | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.modal | This event is fired when the modal has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hidePrevented.bs.modal | This event is fired when the modal is shown, its backdrop option is set to static and a click outside the modal performed, or keyboard option is set to false and an escape key press is performed. |

The following example displays an alert message to the user when fade out transition of the modal window has been fully completed.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listen-to-hidden-bs-modal-event-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal = document.getElementById("myModal");

myModal.addEventListener("hidden.bs.modal",function(){

alert("Modal window has been completely closed.");

});

});

</script>

The following example displays an alert message if you try to close the modal by clicking the dark area.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listen-to-hideprevented-bs-modal-event-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myModal = document.getElementById("myModal");

myModal.addEventListener("hidePrevented.bs.modal",function(){

alert("Modal can't be closed by clicking outside, because the backdrop option is set to static.");

});

});

</script>

**Tip:** See also the [Bootstrap FAQ](https://www.tutorialrepublic.com/faq.php#bootstrap-less) section for more examples on modals, like setting vertical alignment, changing default width, embedding YouTube videos, etc.

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php)

**Bootstrap Dropdowns**

In this tutorial you will learn how to add dropdown menus to various Bootstrap components.

**Creating Dropdown Menus with Bootstrap**

The dropdown menu is typically used inside the navigation header to display a list of related links when a user mouse hover or click on the trigger element.

You can use the Bootstrap dropdown plugin to add toggleable dropdown menus (i.e. open and close on click) to almost anything such as links, buttons or button groups, navbar, tabs and pills nav etc. without even writing a single line of JavaScript code.

**Adding Dropdowns via Data Attributes**

Bootstrap provides an easy and elegant mechanism for adding the dropdown menu to an element via data attributes. The following example will show you the minimum markup required for adding a dropdown menu to the hyperlink via data attributes.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-dropdowns-via-data-attributes)

<div class="dropdown">

<a href="#"class="dropdown-toggle"data-bs-toggle="dropdown">Dropdown</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

</div>

</div>

The above example demonstrates the most basic form of the Bootstrap dropdowns. Let's understand each part of the Bootstrap dropdown component one by one.

**Explanation of Code**

The Bootstrap dropdown has basically two components — the dropdown trigger element which can be a hyperlink or button, and the dropdown menu itself.

* The .dropdown class specifies a dropdown menu.
* The .dropdown-toggle class defines the trigger element, which is a hyperlink in our case, whereas the attribute data-bs-toggle="dropdown" is required on the trigger element to toggle the dropdown menu.
* The [<div>](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-dropdowns.php) element with the class .dropdown-menu is actually building the dropdown menu that typically contains the related links or actions.

Similarly, you can add the dropdowns to the buttons and nav components. The following section will show you some common implementation of the Bootstrap dropdown.

**Dropdowns within a Navbar**

The following examples will show you how to add dropdowns to navbar.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-navbar)

<nav class="navbar navbar-expand-sm navbar-light bg-light">

<div class="container-fluid">

<a href="#"class="navbar-brand">Brand</a>

<button type="button"class="navbar-toggler"data-bs-toggle="collapse"data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div id="navbarCollapse"class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="nav-item">

<a href="#"class="nav-link">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Inbox</a>

<a href="#"class="dropdown-item">Drafts</a>

<a href="#"class="dropdown-item">Sent Items</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Trash</a>

</div>

</li>

</ul>

<ul class="nav navbar-nav ms-auto">

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Admin</a>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Reports</a>

<a href="#"class="dropdown-item">Settings</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Logout</a>

</div>

</li>

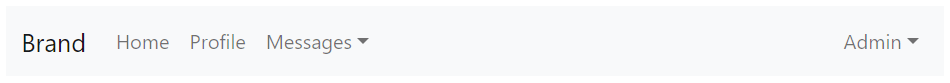
</ul>

</div>

</div>

</nav>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-navbar)

**Tip:** You can draw a divider line to separate the links inside a dropdown menu by adding the class .dropdown-divider on a blank <div> element.

**Dropdowns within Navs**

The following example will show you how to add dropdowns to pills navs.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-nav)

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Inbox</a>

<a href="#"class="dropdown-item">Drafts</a>

<a href="#"class="dropdown-item">Sent Items</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Trash</a>

</div>

</li>

<li class="nav-item dropdown ms-auto">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Admin</a>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Reports</a>

<a href="#"class="dropdown-item">Settings</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Logout</a>

</div>

</li>

</ul>

— The output of the above example will look something like this:

[Bootstrap Dropdowns within Nav](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-nav)

You can simply convert the above example to a tab dropdown by replacing the class .nav-pills with the .nav-tabs, without any further change in markup.

**Dropdowns within Buttons**

The following examples will show you how to add dropdowns to a primary button.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-buttons)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Primary</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— Similarly you can add dropdowns to other variants of the buttons, as shown here:

[Bootstrap Dropdowns within Buttons](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-buttons)

**Bootstrap Split Button Dropdowns**

The following examples will show you how to add dropdowns to split buttons.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=split-button-dropdowns)

<div class="btn-group">

<button type="button"class="btn btn-primary">Primary</button>

<button type="button"class="btn btn-primary dropdown-toggle dropdown-toggle-split"data-bs-toggle="dropdown">

<span class="visually-hidden">Toggle Dropdown</span>

</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— Similarly you can add dropdowns to other variants of the buttons, as shown here:

[Bootstrap Split Button Dropdowns](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=split-button-dropdowns)

**Tip:** You can use the Bootstrap's button relative sizing classes like .btn-lg and .btn-sm on the .btn element to further [resizing the buttons dropdowns](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=sizing-button-dropdowns).

**Dropdowns Inside Button Groups**

To create dropdown menus inside a button group just place a .btn-group along with the dropdown markup within another .btn-group, as shown in the following example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-button-groups)

<div class="btn-group">

<button type="button"class="btn btn-primary">Button</button>

<button type="button"class="btn btn-primary">Another Button</button>

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropdown</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Dropdowns within Button Groups](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-button-groups)

Similarly, you can crate dropdown inside vertically stacked button groups, like this:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-vertically-stacked-button-groups)

<div class="btn-group-vertical">

<button type="button"class="btn btn-primary">Button</button>

<button type="button"class="btn btn-primary">Another Button</button>

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropdown</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

</div>

**Creating Dropup, Dropleft and Dropright Menus**

You can even trigger the dropdown menus above, as well as, at the left and right of the elements by adding an extra class .dropup, .dropstart and .dropend, respectively to the parent element (i.e. the .btn-group element), as shown in the following example.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropup-dropleft-and-dropright-menus)

<!-- Dropup menu -->

<div class="btn-group dropup">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropup</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

<!-- Dropleft menu -->

<div class="btn-group dropstart">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropleft</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

<!-- Dropright menu -->

<div class="btn-group dropend">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropright</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Dropup, Dropleft and Dropright Menus](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropup-dropleft-and-dropright-menus)

**Creating the Right Aligned Dropdown Menus**

By default, the top-left corner of the dropdown menu is positioned at the bottom-left corner of its parent element i.e. 100% from the top and along the left side. To right align the dropdown menu just add an extra class .dropdown-menu-end to the .dropdown-menu base class.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=right-aligned-dropdown-menus)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Right-aligned Dropdown Menu</button>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Right-aligned Dropdown Menus](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=right-aligned-dropdown-menus)

**Adding Headers to Dropdown Items**

You can optionally add a menu header to label a section of menu items inside a dropdown menu by adding the class .dropdown-header to the <div> element, like this:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdown-header)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Products</button>

<div class="dropdown-menu">

<div class="dropdown-header">ELECTRONICS</div>

<a href="#"class="dropdown-item">Mobiles</a>

<a href="#"class="dropdown-item">Tablets</a>

<a href="#"class="dropdown-item">Laptops</a>

<div class="dropdown-header">FASHION</div>

<a href="#"class="dropdown-item">Clothing</a>

<a href="#"class="dropdown-item">Sunglasses</a>

</div>

</div>

**Disable Items within a Dropdown**

You can apply the class .disabled to the menu items in the dropdown to make them look like disabled. However, the link is still clickable, to disable this you can typically [remove the anchor's href attribute](https://www.tutorialrepublic.com/faq/how-to-remove-clickable-behavior-from-a-disabled-link-using-jquery.php) either using the JavaScript or manually.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disabling-dropdown-items)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Reports</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">View</a>

<a href="#"class="dropdown-item">Download</a>

<a href="#"class="dropdown-item disabled"tabindex="-1">Edit / Delete</a>

</div>

</div>

**Adding Dropdowns via JavaScript**

You may also add dropdowns manually using the JavaScript — just call the dropdown() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the link or button element in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-dropdowns-via-jquery)

<script>

$(document).ready(function(){

$(".dropdown-toggle").dropdown();

});

</script>

**Note:** The data-bs-toggle="dropdown" is still required for the dropdown's trigger element regardless of whether you call the dropdown via JavaScript or data-api.

**Options**

There are certain options which can be passed to dropdown() Bootstrap method to customize the functionality of a dropdown. Options can be passed via data attributes or JavaScript.

For setting the dropdown options via data attributes, just append the option name to data-bs-, like data-bs-display="static", and so on. Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-autoClose="false", use data-bs-auto-close="false".

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the dropdown menu (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| reference | string | element | object | 'toggle' | Reference element of the dropdown menu. Accepts the values of 'toggle', 'parent', an HTMLElement reference or an object providing getBoundingClientRect. |
| display | string | 'dynamic' | By default, Bootstrap use Popper for dynamic positioning. You can disable this with 'static'. |
| offset | array | string | function | [0, 2] | Specify the offset of the dropdown relative to its target. You can pass a string in data attributes with comma separated values like: data-bs-offset="10,20". |
| autoClose | boolean | string | true | Configure the auto close behavior of the dropdown:   * true - the dropdown will be closed by clicking outside or inside the dropdown menu. * false - the dropdown will be closed by clicking the toggle button and manually calling the hide or toggle method. Also, dropdown will not be closed by pressing esc key. * 'inside' - the dropdown will be closed (only) by clicking inside the dropdown menu. * 'outside' - the dropdown will be closed (only) by clicking outside the dropdown menu. |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

See the [Popper.js's documentation](https://popper.js.org/popper-documentation.html) for more information on options mentioned above.

**Methods**

This is the standard bootstrap's dropdown method:

**toggle**

This method toggles the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myDropdown").dropdown("toggle");

});

});

</script>

**show**

This method shows the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myDropdown").dropdown("show");

});

});

</script>

**hide**

This method hides the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myDropdown").dropdown("hide");

});

});

</script>

**update**

This method updates the position of an element's dropdown.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myDropdown").dropdown("update");

});

});

</script>

**dispose**

This method destroys an element's dropdown (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myDropdown").dropdown("dispose");

});

});

</script>

**getInstance**

This is a static method which allows you to get the dropdown instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-dropdown-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myDropdown = bootstrap.Dropdown.getInstance($("#myDropdown")[0]);

console.log(myDropdown);

// {\_element: button#myDropdown.btn.btn-primary.dropdown-toggle, \_popper: {…}, \_config: {…}, \_menu: div.dropdown-menu, \_inNavbar: false}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the dropdown instance associated with a DOM element, or create a new one in case if the dropdown wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-dropdown-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myDropdown = bootstrap.Dropdown.getOrCreateInstance($("#myDropdown")[0]);

console.log(myDropdown);

// {\_element: button#myDropdown.btn.btn-primary.dropdown-toggle, \_popper: null, \_config: {…}, \_menu: div.dropdown-menu, \_inNavbar: false}

});

});

</script>

**Events**

These are the standard Bootstrap events to enhance the dropdown functionality.

All dropdown events are fired at the toggling element and then bubbled up. So you can also add event listeners on the .dropdown-menu's parent element. Also, you can use the event.relatedTarget to target the toggling anchor element.

| **Event** | **Description** |
| --- | --- |
| show.bs.dropdown | This event fires immediately when the show instance method is called. |
| shown.bs.dropdown | This event is fired when the dropdown has been made visible to the user and CSS transitions have completed. |
| hide.bs.dropdown | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.dropdown | This event is fired when the dropdown has finished being hidden from the user and CSS transitions have completed. |

The following example displays the text content of dropdown link when the users click on it.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-dropdown-events-using-jquery)

<script>

$(document).ready(function(){

$(".dropdown").on("show.bs.dropdown",function(e){

var linkText =$(e.relatedTarget).text();// Get the link text

alert("Click on OK button to view the dropdown menu for - "+ linkText);

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php)

**Bootstrap Dropdowns**

In this tutorial you will learn how to add dropdown menus to various Bootstrap components.

**Creating Dropdown Menus with Bootstrap**

The dropdown menu is typically used inside the navigation header to display a list of related links when a user mouse hover or click on the trigger element.

You can use the Bootstrap dropdown plugin to add toggleable dropdown menus (i.e. open and close on click) to almost anything such as links, buttons or button groups, navbar, tabs and pills nav etc. without even writing a single line of JavaScript code.

**Adding Dropdowns via Data Attributes**

Bootstrap provides an easy and elegant mechanism for adding the dropdown menu to an element via data attributes. The following example will show you the minimum markup required for adding a dropdown menu to the hyperlink via data attributes.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-dropdowns-via-data-attributes)

<div class="dropdown">

<a href="#"class="dropdown-toggle"data-bs-toggle="dropdown">Dropdown</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

</div>

</div>

The above example demonstrates the most basic form of the Bootstrap dropdowns. Let's understand each part of the Bootstrap dropdown component one by one.

**Explanation of Code**

The Bootstrap dropdown has basically two components — the dropdown trigger element which can be a hyperlink or button, and the dropdown menu itself.

* The .dropdown class specifies a dropdown menu.
* The .dropdown-toggle class defines the trigger element, which is a hyperlink in our case, whereas the attribute data-bs-toggle="dropdown" is required on the trigger element to toggle the dropdown menu.
* The [<div>](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-dropdowns.php) element with the class .dropdown-menu is actually building the dropdown menu that typically contains the related links or actions.

Similarly, you can add the dropdowns to the buttons and nav components. The following section will show you some common implementation of the Bootstrap dropdown.

**Dropdowns within a Navbar**

The following examples will show you how to add dropdowns to navbar.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-navbar)

<nav class="navbar navbar-expand-sm navbar-light bg-light">

<div class="container-fluid">

<a href="#"class="navbar-brand">Brand</a>

<button type="button"class="navbar-toggler"data-bs-toggle="collapse"data-bs-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div id="navbarCollapse"class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="nav-item">

<a href="#"class="nav-link">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Inbox</a>

<a href="#"class="dropdown-item">Drafts</a>

<a href="#"class="dropdown-item">Sent Items</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Trash</a>

</div>

</li>

</ul>

<ul class="nav navbar-nav ms-auto">

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Admin</a>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Reports</a>

<a href="#"class="dropdown-item">Settings</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Logout</a>

</div>

</li>

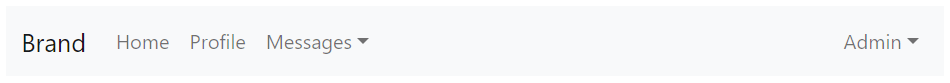
</ul>

</div>

</div>

</nav>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-navbar)

**Tip:** You can draw a divider line to separate the links inside a dropdown menu by adding the class .dropdown-divider on a blank <div> element.

**Dropdowns within Navs**

The following example will show you how to add dropdowns to pills navs.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-nav)

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Messages</a>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Inbox</a>

<a href="#"class="dropdown-item">Drafts</a>

<a href="#"class="dropdown-item">Sent Items</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Trash</a>

</div>

</li>

<li class="nav-item dropdown ms-auto">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Admin</a>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Reports</a>

<a href="#"class="dropdown-item">Settings</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Logout</a>

</div>

</li>

</ul>

— The output of the above example will look something like this:

[Bootstrap Dropdowns within Nav](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-nav)

You can simply convert the above example to a tab dropdown by replacing the class .nav-pills with the .nav-tabs, without any further change in markup.

**Dropdowns within Buttons**

The following examples will show you how to add dropdowns to a primary button.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-buttons)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Primary</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— Similarly you can add dropdowns to other variants of the buttons, as shown here:

[Bootstrap Dropdowns within Buttons](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-buttons)

**Bootstrap Split Button Dropdowns**

The following examples will show you how to add dropdowns to split buttons.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=split-button-dropdowns)

<div class="btn-group">

<button type="button"class="btn btn-primary">Primary</button>

<button type="button"class="btn btn-primary dropdown-toggle dropdown-toggle-split"data-bs-toggle="dropdown">

<span class="visually-hidden">Toggle Dropdown</span>

</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— Similarly you can add dropdowns to other variants of the buttons, as shown here:

[Bootstrap Split Button Dropdowns](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=split-button-dropdowns)

**Tip:** You can use the Bootstrap's button relative sizing classes like .btn-lg and .btn-sm on the .btn element to further [resizing the buttons dropdowns](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=sizing-button-dropdowns).

**Dropdowns Inside Button Groups**

To create dropdown menus inside a button group just place a .btn-group along with the dropdown markup within another .btn-group, as shown in the following example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-button-groups)

<div class="btn-group">

<button type="button"class="btn btn-primary">Button</button>

<button type="button"class="btn btn-primary">Another Button</button>

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropdown</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Dropdowns within Button Groups](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-button-groups)

Similarly, you can crate dropdown inside vertically stacked button groups, like this:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdowns-within-vertically-stacked-button-groups)

<div class="btn-group-vertical">

<button type="button"class="btn btn-primary">Button</button>

<button type="button"class="btn btn-primary">Another Button</button>

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropdown</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

</div>

**Creating Dropup, Dropleft and Dropright Menus**

You can even trigger the dropdown menus above, as well as, at the left and right of the elements by adding an extra class .dropup, .dropstart and .dropend, respectively to the parent element (i.e. the .btn-group element), as shown in the following example.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropup-dropleft-and-dropright-menus)

<!-- Dropup menu -->

<div class="btn-group dropup">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropup</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

<!-- Dropleft menu -->

<div class="btn-group dropstart">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropleft</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

<!-- Dropright menu -->

<div class="btn-group dropend">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Dropright</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Dropup, Dropleft and Dropright Menus](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropup-dropleft-and-dropright-menus)

**Creating the Right Aligned Dropdown Menus**

By default, the top-left corner of the dropdown menu is positioned at the bottom-left corner of its parent element i.e. 100% from the top and along the left side. To right align the dropdown menu just add an extra class .dropdown-menu-end to the .dropdown-menu base class.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=right-aligned-dropdown-menus)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Right-aligned Dropdown Menu</button>

<div class="dropdown-menu dropdown-menu-end">

<a href="#"class="dropdown-item">Action</a>

<a href="#"class="dropdown-item">Another action</a>

<div class="dropdown-divider"></div>

<a href="#"class="dropdown-item">Separated link</a>

</div>

</div>

— The output of the above example will look something like this:

[Bootstrap Right-aligned Dropdown Menus](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=right-aligned-dropdown-menus)

**Adding Headers to Dropdown Items**

You can optionally add a menu header to label a section of menu items inside a dropdown menu by adding the class .dropdown-header to the <div> element, like this:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dropdown-header)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Products</button>

<div class="dropdown-menu">

<div class="dropdown-header">ELECTRONICS</div>

<a href="#"class="dropdown-item">Mobiles</a>

<a href="#"class="dropdown-item">Tablets</a>

<a href="#"class="dropdown-item">Laptops</a>

<div class="dropdown-header">FASHION</div>

<a href="#"class="dropdown-item">Clothing</a>

<a href="#"class="dropdown-item">Sunglasses</a>

</div>

</div>

**Disable Items within a Dropdown**

You can apply the class .disabled to the menu items in the dropdown to make them look like disabled. However, the link is still clickable, to disable this you can typically [remove the anchor's href attribute](https://www.tutorialrepublic.com/faq/how-to-remove-clickable-behavior-from-a-disabled-link-using-jquery.php) either using the JavaScript or manually.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disabling-dropdown-items)

<div class="btn-group">

<button type="button"class="btn btn-primary dropdown-toggle"data-bs-toggle="dropdown">Reports</button>

<div class="dropdown-menu">

<a href="#"class="dropdown-item">View</a>

<a href="#"class="dropdown-item">Download</a>

<a href="#"class="dropdown-item disabled"tabindex="-1">Edit / Delete</a>

</div>

</div>

**Adding Dropdowns via JavaScript**

You may also add dropdowns manually using the JavaScript — just call the dropdown() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the link or button element in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-dropdowns-via-jquery)

<script>

$(document).ready(function(){

$(".dropdown-toggle").dropdown();

});

</script>

**Note:** The data-bs-toggle="dropdown" is still required for the dropdown's trigger element regardless of whether you call the dropdown via JavaScript or data-api.

**Options**

There are certain options which can be passed to dropdown() Bootstrap method to customize the functionality of a dropdown. Options can be passed via data attributes or JavaScript.

For setting the dropdown options via data attributes, just append the option name to data-bs-, like data-bs-display="static", and so on. Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-autoClose="false", use data-bs-auto-close="false".

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the dropdown menu (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| reference | string | element | object | 'toggle' | Reference element of the dropdown menu. Accepts the values of 'toggle', 'parent', an HTMLElement reference or an object providing getBoundingClientRect. |
| display | string | 'dynamic' | By default, Bootstrap use Popper for dynamic positioning. You can disable this with 'static'. |
| offset | array | string | function | [0, 2] | Specify the offset of the dropdown relative to its target. You can pass a string in data attributes with comma separated values like: data-bs-offset="10,20". |
| autoClose | boolean | string | true | Configure the auto close behavior of the dropdown:   * true - the dropdown will be closed by clicking outside or inside the dropdown menu. * false - the dropdown will be closed by clicking the toggle button and manually calling the hide or toggle method. Also, dropdown will not be closed by pressing esc key. * 'inside' - the dropdown will be closed (only) by clicking inside the dropdown menu. * 'outside' - the dropdown will be closed (only) by clicking outside the dropdown menu. |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

See the [Popper.js's documentation](https://popper.js.org/popper-documentation.html) for more information on options mentioned above.

**Methods**

This is the standard bootstrap's dropdown method:

**toggle**

This method toggles the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

// Create a dropdown instance

var myDropdown =newbootstrap.Dropdown(element);

btn.addEventListener("click",function(){

myDropdown.toggle();

});

});

</script>

**show**

This method shows the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

// Create a dropdown instance

var myDropdown =newbootstrap.Dropdown(element);

btn.addEventListener("click",function(){

myDropdown.show();

});

});

</script>

**hide**

This method hides the dropdown menu of a given navbar or tabbed navigation.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

// Create a dropdown instance

var myDropdown =newbootstrap.Dropdown(element);

btn.addEventListener("click",function(){

myDropdown.hide();

});

});

</script>

**update**

This method updates the position of an element's dropdown.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

// Create a dropdown instance

var myDropdown =newbootstrap.Dropdown(element);

btn.addEventListener("click",function(){

myDropdown.update();

});

});

</script>

**dispose**

This method destroys an element's dropdown (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-dropdown-methods-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

// Create a dropdown instance

var myDropdown =newbootstrap.Dropdown(element);

btn.addEventListener("click",function(){

myDropdown.dispose();

});

});

</script>

**getInstance**

This is a static method which allows you to get the dropdown instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-dropdown-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

btn.addEventListener("click",function(){

var myDropdown = bootstrap.Dropdown.getInstance(element);

console.log(myDropdown);

// {\_element: button#myDropdown.btn.btn-primary.dropdown-toggle, \_popper: {…}, \_config: {…}, \_menu: div.dropdown-menu, \_inNavbar: false}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the dropdown instance associated with a DOM element, or create a new one in case if the dropdown wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-dropdown-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myDropdown");

btn.addEventListener("click",function(){

var myDropdown = bootstrap.Dropdown.getOrCreateInstance(element);

console.log(myDropdown);

// {\_element: button#myDropdown.btn.btn-primary.dropdown-toggle, \_popper: null, \_config: {…}, \_menu: div.dropdown-menu, \_inNavbar: false}

});

});

</script>

**Events**

These are the standard Bootstrap events to enhance the dropdown functionality.

All dropdown events are fired at the toggling element and then bubbled up. So you can also add event listeners on the .dropdown-menu's parent element. Also, you can use the event.relatedTarget to target the toggling anchor element.

| **Event** | **Description** |
| --- | --- |
| show.bs.dropdown | This event fires immediately when the show instance method is called. |
| shown.bs.dropdown | This event is fired when the dropdown has been made visible to the user and CSS transitions have completed. |
| hide.bs.dropdown | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.dropdown | This event is fired when the dropdown has finished being hidden from the user and CSS transitions have completed. |

The following example displays the text content of dropdown link when the users click on it.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-dropdown-events-using-jquery)

<script>

$(document).ready(function(){

$(".dropdown").on("show.bs.dropdown",function(e){

var linkText =$(e.relatedTarget).text();// Get the link text

alert("Click on OK button to view the dropdown menu for - "+ linkText);

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-modals.php)[NE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tabs.php)

# Bootstrap Tabs

In this tutorial you will learn how to create dynamic tabs to toggle between the content using the Bootstrap tabs plugin.

## Creating Tabs with Bootstrap

Tab based navigations provides a powerful mechanism to handle huge amount of content within a small area through separating content into different panes where each pane is viewable one at a time.

User can quickly access the content through switching between the panes without leaving the page. The following example will show you how to create the basic tabs using the Bootstrap tab component.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-tabs)

<ul class="nav nav-tabs">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

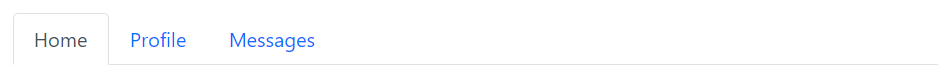
<li class="nav-item">

<a href="#"class="nav-link">Messages</a>

</li>

</ul>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-tabs)

The tabs plugin also works with pills nav. To create pill nav just replace the class .nav-tabs with .nav-pills in the tab markup, as shown in the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-pills)

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Messages</a>

</li>

</ul>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-pills)

## Creating Dynamic Tabs via Data Attributes

You can activate a tab or pill navigation without writing any JavaScript code — simply specify the data-bs-toggle="tab" on each tab, or data-bs-toggle="pill" on each pill, as well as create a .tab-pane with unique ID for every tab and wrap them in .tab-content.

Let's take a look at an example to understand how to create dynamic tabs via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-dynamic-tabs-via-data-attributes)

<ul class="nav nav-tabs">

<li class="nav-item">

<a href="#home"class="nav-link active"data-bs-toggle="tab">Home</a>

</li>

<li class="nav-item">

<a href="#profile"class="nav-link"data-bs-toggle="tab">Profile</a>

</li>

<li class="nav-item">

<a href="#messages"class="nav-link"data-bs-toggle="tab">Messages</a>

</li>

</ul>

<div class="tab-content">

<div class="tab-pane fade show active"id="home">

<p>Home tab content ...</p>

</div>

<div class="tab-pane fade"id="profile">

<p>Profile tab content ...</p>

</div>

<div class="tab-pane fade"id="messages">

<p>Messages tab content ...</p>

</div>

</div>

**Tip:** The .fade class is added to each .tab-pane to make tabs fade in while showing new tab content. Also, you must add .active class to a .nav-link, as well as .show and .active classes to the corresponding .tab-pane to make it initially visible.

## Creating Dynamic Tabs via JavaScript

You may also enable tabs via JavaScript. Each tab needs to be activated individually.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-dynamic-tabs-via-jquery)

<script>

$(document).ready(function(){

$("#myTab a").click(function(e){

e.preventDefault();

$(this).tab("show");

});

});

</script>

You can activate individual tabs in several ways. Let's take a look at the following example:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-individual-tabs-via-jquery)

$('#myTab a[href="#profile"]').tab("show");// show tab targeted by the selector

$("#myTab a:first").tab("show");// show first tab

$("#myTab a:last").tab("show");// show last tab

$("#myTab li:eq(1) a").tab("show");// show second tab (0-indexed, like an array)

## Methods

This is the standard bootstrap's tab method:

## show

Activates the given tab and shows its associated pane. Any other tab that was previously selected becomes unselected and its associated pane is hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-tab-using-jquery)

<script>

$(document).ready(function(){

$("#myTab a:last").tab("show");// show last tab

});

</script>

## dispose

This method destroys an element's tab (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-tab-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var lastTab = bootstrap.Tab.getInstance($("#myTab a:last")[0]);

console.log(lastTab);

// {\_element: a.nav-link}

$("#myTab a:last").tab("dispose");

console.log(lastTab);

// {\_element: null}

});

});

</script>

## getInstance

This is a static method which allows you to get the tab instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-tab-instance-using-jquery)

<script>

$(document).ready(function(){

// Get tab instance on button click

$("#myBtn").click(function(){

var lastTab = bootstrap.Tab.getInstance($("#myTab a:first")[0]);

console.log(lastTab);

// {\_element: a.nav-link.active}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the tab instance associated with a DOM element, or create a new one in case if the tab wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-tab-instance-using-jquery)

<script>

$(document).ready(function(){

// Get or create tab instance on button click

$("#myBtn").click(function(){

var lastTab = bootstrap.Tab.getOrCreateInstance($("#myTab a:first")[0]);

console.log(lastTab);

// {\_element: a.nav-link.active}

});

});

</script>

## Events

These are the standard Bootstrap events to enhance the tab functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.tab | This event fires on tab show, but before the new tab has been shown. You can use the event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively. |
| shown.bs.tab | This event fires on tab show after a tab has been shown. You can use the event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively. |
| hide.bs.tab | This event fires when the current active tab is to be hidden and thus a new tab is to be shown. You can use the event.target and event.relatedTarget to target the current active tab and the new tab which is going to be active very soon, respectively. |
| hidden.bs.tab | This event fires after the previous active tab is hidden and a new tab is shown. You can use the event.target and event.relatedTarget to target the previous active tab and the new active tab, respectively. |

The following example displays the names of active tab and previous tab to the user when transition of a tab has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-tab-events-using-jquery)

<script>

$(document).ready(function(){

$('a[data-bs-toggle="tab"]').on("shown.bs.tab",function(e){

console.log(e.target);// newly activated tab

console.log(e.relatedTarget);// previous active tab

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-dropdowns.php)

# Bootstrap Tabs

In this tutorial you will learn how to create dynamic tabs to toggle between the content using the Bootstrap tabs plugin.

## Creating Tabs with Bootstrap

Tab based navigations provides a powerful mechanism to handle huge amount of content within a small area through separating content into different panes where each pane is viewable one at a time.

User can quickly access the content through switching between the panes without leaving the page. The following example will show you how to create the basic tabs using the Bootstrap tab component.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-tabs)

<ul class="nav nav-tabs">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

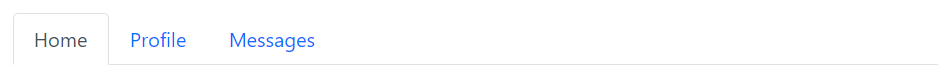
<li class="nav-item">

<a href="#"class="nav-link">Messages</a>

</li>

</ul>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-tabs)

The tabs plugin also works with pills nav. To create pill nav just replace the class .nav-tabs with .nav-pills in the tab markup, as shown in the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-pills)

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#"class="nav-link active">Home</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Profile</a>

</li>

<li class="nav-item">

<a href="#"class="nav-link">Messages</a>

</li>

</ul>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=basic-pills)

## Creating Dynamic Tabs via Data Attributes

You can activate a tab or pill navigation without writing any JavaScript code — simply specify the data-bs-toggle="tab" on each tab, or data-bs-toggle="pill" on each pill, as well as create a .tab-pane with unique ID for every tab and wrap them in .tab-content.

Let's take a look at an example to understand how to create dynamic tabs via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-dynamic-tabs-via-data-attributes)

<ul class="nav nav-tabs">

<li class="nav-item">

<a href="#home"class="nav-link active"data-bs-toggle="tab">Home</a>

</li>

<li class="nav-item">

<a href="#profile"class="nav-link"data-bs-toggle="tab">Profile</a>

</li>

<li class="nav-item">

<a href="#messages"class="nav-link"data-bs-toggle="tab">Messages</a>

</li>

</ul>

<div class="tab-content">

<div class="tab-pane fade show active"id="home">

<p>Home tab content ...</p>

</div>

<div class="tab-pane fade"id="profile">

<p>Profile tab content ...</p>

</div>

<div class="tab-pane fade"id="messages">

<p>Messages tab content ...</p>

</div>

</div>

**Tip:** The .fade class is added to each .tab-pane to make tabs fade in while showing new tab content. Also, you must add .active class to a .nav-link, as well as .show and .active classes to the corresponding .tab-pane to make it initially visible.

## Creating Dynamic Tabs via JavaScript

You may also enable tabs via JavaScript. Each tab needs to be activated individually.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-dynamic-tabs-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var tabList =[].slice.call(document.querySelectorAll("#myTab a"));

tabList.forEach(function(tab){

var tabTrigger =newbootstrap.Tab(tab);

tab.addEventListener("click",function(event){

event.preventDefault();

tabTrigger.show();

});

});

});

</script>

You can activate individual tabs in several ways. Let's take a look at the following example:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-individual-tabs-via-javascript)

// Show tab targeted by the selector

var profileTabElement = document.querySelector('#myTab a[href="#profile"]');

var profileTab =newbootstrap.Tab(profileTabElement);

profileTab.show();

// Show first tab

var firstTabElement = document.querySelector("#myTab li:first-child a");

var firstTab =newbootstrap.Tab(firstTabElement);

firstTab.show();

// Show last tab

var lastTabElement = document.querySelector("#myTab li:last-child a");

var lastTab =newbootstrap.Tab(lastTabElement);

lastTab.show();

// Show second tab (0-indexed, like an array);

var secondTabElement = document.querySelectorAll("#myTab li")[1].firstElementChild;

var secondTab =newbootstrap.Tab(secondTabElement);

secondTab.show();

## Methods

This is the standard bootstrap's tab method:

## show

Activates the given tab and shows its associated pane. Any other tab that was previously selected becomes unselected and its associated pane is hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=show-tab-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

// Show last tab

var lastTabElement = document.querySelector("#myTab li:last-child a");

var lastTab =newbootstrap.Tab(lastTabElement);

lastTab.show();

});

</script>

## dispose

This method destroys an element's tab (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-tab-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.querySelector("#myTab li:last-child a");

btn.addEventListener("click",function(){

var lastTab = bootstrap.Tab.getInstance(element);

console.log(lastTab);

// {\_element: a.nav-link}

lastTab.dispose();

console.log(lastTab);

// {\_element: null}

});

});

</script>

## getInstance

This is a static method which allows you to get the tab instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-tab-instance-using-jquery)

<script>

$(document).ready(function(){

// Get tab instance on button click

$("#myBtn").click(function(){

var lastTab = bootstrap.Tab.getInstance($("#myTab a:first")[0]);

console.log(lastTab);

// {\_element: a.nav-link.active}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the tab instance associated with a DOM element, or create a new one in case if the tab wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-tab-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.querySelector("#myTab li:first-child a");

// Get or create tab instance on button click

btn.addEventListener("click",function(){

var firstTab = bootstrap.Tab.getOrCreateInstance(element);

console.log(firstTab);

// {\_element: a.nav-link.active}

});

});

</script>

## Events

These are the standard Bootstrap events to enhance the tab functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.tab | This event fires on tab show, but before the new tab has been shown. You can use the event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively. |
| shown.bs.tab | This event fires on tab show after a tab has been shown. You can use the event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively. |
| hide.bs.tab | This event fires when the current active tab is to be hidden and thus a new tab is to be shown. You can use the event.target and event.relatedTarget to target the current active tab and the new tab which is going to be active very soon, respectively. |
| hidden.bs.tab | This event fires after the previous active tab is hidden and a new tab is shown. You can use the event.target and event.relatedTarget to target the previous active tab and the new active tab, respectively. |

The following example displays the names of active tab and previous tab to the user when transition of a tab has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-tab-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var tabList =[].slice.call(document.querySelectorAll('a[data-bs-toggle="tab"]'));

tabList.forEach(function(tab){

tab.addEventListener("shown.bs.tab",function(e){

console.log(e.target);// newly activated tab

console.log(e.relatedTarget);// previous active tab

});

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-dropdowns.php)[NEXT PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-collapse.php)

**Bootstrap Collapse**

In this tutorial you will learn how to show and hide elements with Bootstrap.

**Toggle Display of Content with Bootstrap**

You can use the Bootstrap collapse JavaScript plugin to easily show and hide (or expand and collapse) specific elements on a web page. Buttons and anchors (i.e. the <button> and <a> elements) are typically used as triggers that are mapped to the elements you want to toggle.

**Expand and Collapse Elements via Data Attributes**

You can show and hide elements in Bootstrap by clicking on a button or link via data attributes without writing any JavaScript code. Let's try out an example and see how it actually works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-collapsible-element-via-data-attributes)

<!-- Trigger Buttons HTML -->

<a href="#myCollapse"data-bs-toggle="collapse">Toggle Element</a>

<button type="button"class="btn btn-primary"data-bs-toggle="collapse"data-bs-target="#myCollapse">Toggle Element</button>

<!-- Collapsible Element HTML -->

<div class="collapse show"id="myCollapse">

<div class="card card-body">This is a simple example of showing and hiding specific element via data attributes. Click any trigger buttons to toggle this panel.</div>

</div>

We've just created a collapsible control without writing any JavaScript code. Well, let's go through each part of this code one by one for a better understanding.

**Explanation of Code**

The Bootstrap collapse plugin basically requires the two elements to work properly — the trigger element such as a button or hyperlink, and the collapsible element itself.

* The data-bs-toggle="collapse" attribute is added to the trigger or controller element along with a attribute data-bs-target (for buttons) or href (for anchors) to automatically assign control of one or more collapsible elements.
* The data-bs-target or href attribute accepts a [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) (such as id selector #myCollapse in our example) to apply the collapse to a specific element.
* The class .collapse is added to the collapsible element (*line no-6*).
* You can optionally add the class .show (*line no-6*) to the collapsible element in addition to the class .collapse to make it open by default.

To make the collapsible controls to work in group like accordion menu, you can utilize the [Bootstrap card](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-cards.php) component. See the tutorial on [Bootstrap accordion](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-accordion.php) to learn more about it.

**Expand and Collapse Elements via JavaScript**

You may also expand and collapse elements manually via JavaScript — just call the collapse() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-collapsible-element-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("toggle");

});

});

</script>

**Options**

There are certain options which may be passed to the collapse() Bootstrap method to customize the functionality of a collapsible element.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| parent | selector | false | All other collapsible elements under the specified parent will be closed while this collapsible item is shown on invocation. |
| toggle | boolean | true | Toggles the collapsible element on invocation. |

You can also set these options using the data attributes on accordion — just append the option name to data-bs-, like data-bs-parent="#myAccordion", data-bs-toggle="false".

**Methods**

These are the standard bootstrap's collapse methods:

**Passing options**

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-collapse.php#collapse-options) to the collapse using options object.

In following example the display of the collapsible element will not toggle on invocation, because the toggle option for the collapsible element is set to false.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=passing-options-to-collapse-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("toggle",{

toggle:false

});

});

});

</script>

**toggle**

This method toggles (show or hide) a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("toggle");

});

});

</script>

**show**

This method shows a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("show");

});

});

</script>

**hide**

This method hides a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("hide");

});

});

</script>

**dispose**

This method destroys an element's collapse (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-collapse-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCollapse = bootstrap.Collapse.getInstance($("#myCollapse")[0]);

console.log(myCollapse);

// {\_element: div#myCollapse.collapse.show, \_isTransitioning: false, \_config: {…}, \_triggerArray: Array(2), \_selector: "#myCollapse", …}

$("#myCollapse").collapse("dispose");

console.log(myCollapse);

// {\_element: null, \_isTransitioning: null, \_config: null, \_triggerArray: null, \_selector: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the collapse instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-collapse-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCollapse = bootstrap.Collapse.getInstance($("#myCollapse")[0]);

console.log(myCollapse);

// {\_element: div#myCollapse.collapse.show, \_isTransitioning: false, \_config: {…}, \_triggerArray: Array(2), \_selector: "#myCollapse", …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the collapse instance associated with a DOM element, or create a new one in case if the collapse wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-collapse-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCollapse = bootstrap.Collapse.getOrCreateInstance($("#myCollapse")[0]);

console.log(myCollapse);

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

**Events**

Bootstrap's collapse class includes few events for hooking into collapse functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.collapse | This event fires immediately when the show instance method is called. |
| shown.bs.collapse | This event is fired when a collapse element has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.collapse | This event is fired immediately when the hide method has been called. |
| hidden.bs.collapse | This event is fired when a collapse element has been hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |

The following example displays an alert message to the user when sliding transition of a collapsible element has been fully completed.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-collapse-events-using-jquery)

<script>

$(document).ready(function(){

$("#myCollapse").on("hidden.bs.collapse",function(){

alert("Collapsible element has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tabs.php)[NEXT PAG](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php)

**Bootstrap Collapse**

In this tutorial you will learn how to show and hide elements with Bootstrap.

**Toggle Display of Content with Bootstrap**

You can use the Bootstrap collapse JavaScript plugin to easily show and hide (or expand and collapse) specific elements on a web page. Buttons and anchors (i.e. the <button> and <a> elements) are typically used as triggers that are mapped to the elements you want to toggle.

**Expand and Collapse Elements via Data Attributes**

You can show and hide elements in Bootstrap by clicking on a button or link via data attributes without writing any JavaScript code. Let's try out an example and see how it actually works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-collapsible-element-via-data-attributes)

<!-- Trigger Buttons HTML -->

<a href="#myCollapse"data-bs-toggle="collapse">Toggle Element</a>

<button type="button"class="btn btn-primary"data-bs-toggle="collapse"data-bs-target="#myCollapse">Toggle Element</button>

<!-- Collapsible Element HTML -->

<div class="collapse show"id="myCollapse">

<div class="card card-body">This is a simple example of showing and hiding specific element via data attributes. Click any trigger buttons to toggle this panel.</div>

</div>

We've just created a collapsible control without writing any JavaScript code. Well, let's go through each part of this code one by one for a better understanding.

**Explanation of Code**

The Bootstrap collapse plugin basically requires the two elements to work properly — the trigger element such as a button or hyperlink, and the collapsible element itself.

* The data-bs-toggle="collapse" attribute is added to the trigger or controller element along with a attribute data-bs-target (for buttons) or href (for anchors) to automatically assign control of one or more collapsible elements.
* The data-bs-target or href attribute accepts a [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) (such as id selector #myCollapse in our example) to apply the collapse to a specific element.
* The class .collapse is added to the collapsible element (*line no-6*).
* You can optionally add the class .show (*line no-6*) to the collapsible element in addition to the class .collapse to make it open by default.

To make the collapsible controls to work in group like accordion menu, you can utilize the [Bootstrap card](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-cards.php) component. See the tutorial on [Bootstrap accordion](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-accordion.php) to learn more about it.

**Expand and Collapse Elements via JavaScript**

You may also expand and collapse elements manually via JavaScript — just call the collapse() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=creating-collapsible-element-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("toggle");

});

});

</script>

**Options**

There are certain options which may be passed to the collapse() Bootstrap method to customize the functionality of a collapsible element.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| parent | selector | false | All other collapsible elements under the specified parent will be closed while this collapsible item is shown on invocation. |
| toggle | boolean | true | Toggles the collapsible element on invocation. |

You can also set these options using the data attributes on accordion — just append the option name to data-bs-, like data-bs-parent="#myAccordion", data-bs-toggle="false".

**Methods**

These are the standard bootstrap's collapse methods:

**Passing options**

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-collapse.php#collapse-options) to the collapse using options object.

In following example the display of the collapsible element will not toggle on invocation, because the toggle option for the collapsible element is set to false.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=passing-options-to-collapse-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCollapse").collapse("toggle",{

toggle:false

});

});

});

</script>

**toggle**

This method toggles (show or hide) a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

// Create a collapse instance, toggles the collapse element on invocation

var myCollapse =newbootstrap.Collapse(element);

btn.addEventListener("click",function(){

myCollapse.toggle();

});

});

</script>

**show**

This method shows a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

// Create a collapse instance, toggles the collapse element on invocation

var myCollapse =newbootstrap.Collapse(element);

btn.addEventListener("click",function(){

myCollapse.show();

});

});

</script>

**hide**

This method hides a collapsible element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-collapse-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

// Create a collapse instance, toggles the collapse element on invocation

var myCollapse =newbootstrap.Collapse(element);

btn.addEventListener("click",function(){

myCollapse.hide();

});

});

</script>

**dispose**

This method destroys an element's collapse (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-collapse-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

btn.addEventListener("click",function(){

var myCollapse = bootstrap.Collapse.getInstance(element);

console.log(myCollapse);

// {\_element: div#myCollapse.collapse.show, \_isTransitioning: false, \_config: {…}, \_triggerArray: Array(2), \_selector: "#myCollapse", …}

myCollapse.dispose();

console.log(myCollapse);

// {\_element: null, \_isTransitioning: null, \_config: null, \_triggerArray: null, \_selector: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the collapse instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-collapse-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

btn.addEventListener("click",function(){

var myCollapse = bootstrap.Collapse.getInstance(element);

console.log(myCollapse);

// {\_element: div#myCollapse.collapse.show, \_isTransitioning: false, \_config: {…}, \_triggerArray: Array(2), \_selector: "#myCollapse", …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the collapse instance associated with a DOM element, or create a new one in case if the collapse wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-collapse-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCollapse");

btn.addEventListener("click",function(){

var myCollapse = bootstrap.Collapse.getOrCreateInstance(element);

console.log(myCollapse);

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

**Events**

Bootstrap's collapse class includes few events for hooking into collapse functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.collapse | This event fires immediately when the show instance method is called. |
| shown.bs.collapse | This event is fired when a collapse element has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.collapse | This event is fired immediately when the hide method has been called. |
| hidden.bs.collapse | This event is fired when a collapse element has been hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |

The following example displays an alert message to the user when sliding transition of a collapsible element has been fully completed.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-collapse-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myCollapse = document.getElementById("myCollapse");

myCollapse.addEventListener("hidden.bs.collapse",function(){

alert("Collapsible element has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tabs.php)[NEX](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php)

# Bootstrap Tooltips

In this tutorial you will learn how to create tooltips with Bootstrap.

## Creating the Tooltips with Bootstrap

A tooltip is a small pop up that appears when user places the mouse pointer over an element such as link or buttons to provide hint or information about the element being hovered.

Tooltips can be very helpful for the new visitors of your website because they enable the user to know the purpose of icons and links by placing the mouse pointer over them.

### Step 1: Adding the Tooltip Markup

To create a tooltip, you need to add the data-bs-toggle="tooltip" attribute to an element. Tolltip text that would display on hover can be specified using the title attribute.

Here is the standard markup for adding a tolltip to a hyperlink.

<a href="#" data-bs-toggle="tooltip" title="Some text">Hover over me</a>

Similarly, you can add tooltips to the other elements such as buttons, icons, etc.

### Step 2: Enabling the Tooltips

Tooltips can be enabled via JavaScript — just call the tooltip() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the target element in your JavaScript code.

You can either initialize tooltips individually or all in one go. The following jQuery code will initialize all tooltips on the page by selecting them by their data-bs-toggle attribute.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-tooltips-via-jquery)

<script>

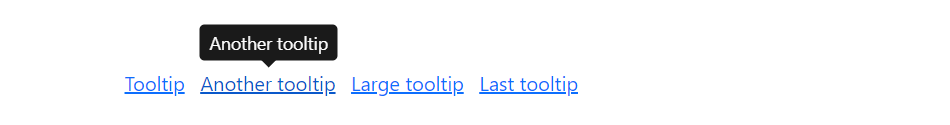
$(document).ready(function(){

$('[data-bs-toggle="tooltip"]').tooltip();

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-tooltips-via-jquery)

**Note:** For performance reasons the tooltip data-apis are opt in, means to use tooltips you must initialize them yourself. Also, tooltips with zero-length titles are never displayed, as well as triggering tooltips on hidden elements will not work.

**Tip:** Tooltips for .disabled or disabled elements must be triggered on a wrapper element. Also, when tooltips are triggered from hyperlinks that span multiple lines, it will be centered. You can use white-space: nowrap; on those hyperlinks to avoid this behavior.

## Setting the Directions of Tooltips

You can set tooltips to appear on top, right, bottom and left sides of an element.

### Positioning of Tooltips via Data Attributes

The following example will show you how to set the direction of tooltips via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tooltips-placement-via-data-attributes)

<a href="#"data-bs-toggle="tooltip"data-bs-placement="top"title="Tooltip on top">Tooltip on top</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="right"title="Tooltip on right">Tooltip on right</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="bottom"title="Tooltip on bottom">Tooltip on bottom</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="left"title="Tooltip on left">Tooltip on left</a>

### Positioning of Tooltips via JavaScript

The following example will show you how to set the direction of tooltips via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tooltips-placement-via-jquery)

<script>

$(document).ready(function(){

$("#tipTop").tooltip({ placement :"top"});

$("#tipRight").tooltip({ placement :"right"});

$("#tipBottom").tooltip({ placement :"bottom"});

$("#tipLeft").tooltip({ placement :"left"});

});

</script>

## Options

There are certain options which may be passed to tooltip() Bootstrap method to customize the functionality of the tooltip plugin.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the tooltip. |
| container | string | element | false | false | Appends the tooltip to a specific element.  Specify container: 'body' to avoid rendering problems in more complex components (like input groups, button groups, etc.) |
| delay | number | object | 0 | Time to delay in showing and hiding the tooltip (ms) — does not apply to manual trigger type.  If a number is supplied, delay is applied to both hide/show  Object structure is: delay: { "show": 500, "hide": 100 } |
| html | boolean | false | Insert HTML into the tooltip.  If true, HTML tags in the tooltip's title will be rendered in the tooltip. If false, innerText property will be used to insert content into the DOM.  Use simple text if you're worried about XSS attacks. |
| placement | string | function | 'top' | Sets the position of the tooltip — auto | top | bottom | left | right.  When auto value is specified, it will dynamically reorient the tooltip. |
| selector | string | false | false | If a selector is provided, tooltip objects will be attached to the specified targets.  This is normally used to apply tooltips to dynamically added DOM elements. |
| template | string | '<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-inner"></div></div>' | Base HTML to use when creating the tooltip.  The tooltip's title will be inserted into the .tooltip-inner element.  The .tooltip-arrow element will become the tooltip's arrow.  The outermost wrapper element should have the .tooltip class. |
| title | string | element | function | '' | Sets the default title value if title attribute isn't present. |
| trigger | string | 'hover focus' | Specify how tooltip is triggered — click | hover | focus | manual. You can pass multiple triggers; separated with a space.  The value manual indicates that the tooltip will be triggered programmatically via the .show(), .hide() and .toggle() methods; this value cannot be combined with any other trigger. |
| fallbackPlacements | array | ['top', 'right', 'bottom', 'left'] | Allows you to specify which placement Popper will use on fallback. |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the tooltip (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| customClass | string | function | '' | Add classes to the tooltip when it is shown. Please note that these classes will be added in addition to any classes specified in the template. To add multiple classes, separate them with spaces like: 'class1 class2'.  You can also pass a function that should return a single string containing additional class names. |
| sanitize | boolean | true | Enable or disable the sanitization. If activated 'template' and 'title' options will be sanitized. |
| allowList | object | [*Default value*](https://getbootstrap.com/docs/4.6/getting-started/javascript/#sanitizer) | Object which contains allowed attributes and tags. |
| sanitizeFn | null | function | null | Allows you to specify your own sanitize function. |
| offset | array | string | function | [0, 0] | Offset of the tooltip relative to its target. You can also pass a string in data attributes with comma separated values like: data-bs-offset="10,20" |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

You can set these options either through the use of data attributes or JavaScript. For setting the tooltips options via data attributes, just append the option name to data-bs- along with the correct value, like data-bs-animation="false", data-bs-placement="bottom" etc.

Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-customClass="my-class", use data-bs-custom-class="my-class".

However, JavaScript is the preferred way of setting these options as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php#passing-options) section below to know how to set the options for tooltips via JavaScript.

## Methods

These are the standard Bootstrap's tooltip methods:

## Passing options

You can additionally pass options to the tooltips using options object.

The following example will set the title text for the tooltips dynamically, if the value of the title attribute is omitted or missing from the selected elements:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-title-text-for-tooltips-using-jquery)

<script>

$(document).ready(function(){

$("#myTooltip").tooltip({

title :"It looks like title attribute is not present."

});

});

</script>

The following example will show you how to place the HTML content inside a tooltip:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=place-html-inside-a-tooltip-using-jquery)

<script>

$(document).ready(function(){

$("#myTooltip").tooltip({

title:"<h4><img src='images/smiley.png' width='30' alt='Smiley'> Hello, <b>I'm</b><i>Smiley!</i></h4>",

html:true

});

});

</script>

The following example will show you how to control the timing of showing and hiding the tooltip using the tooltip's delay option dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-tooltips-show-hide-time-delay-using-jquery)

<script>

$(document).ready(function(){

// Show and hide tooltip with same speed

$("#tinyTooltip").tooltip({

delay:100

});

// Show and hide tooltip with different speed

$("#largeTooltip").tooltip({

delay:{show:0, hide:2000}

});

});

</script>

The following example will show you how to create your own custom template for Bootstrap tooltips instead of using the default one dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-custom-template-for-tooltips-using-jquery)

<script>

$(document).ready(function(){

$("#myTooltip").tooltip({

template:'<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-head"><h3><i class="bi-info-circle"></i> Important Info</h3></div><div class="tooltip-inner"></div>'

});

});

</script>

The following example will insert the dynamically generated HTML code of the tooltip at the end of the #wrapper element instead of the default [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-tooltips-container-using-jquery)

<script>

$(document).ready(function(){

// Append tooltip HTML to wrapper element

$("#myTooltip").tooltip({

container:"#wrapper"

});

});

</script>

**Note:** Overriding the tooltip's default container option value does not produce any visible difference on page. To see the actual result you need to inspect the DOM. Press Ctrl+Shift+I (Windows / Linux) or Cmd+Opt+I (Mac) to open Developer tools or DOM Inspector.

Similarly, you can set other options for the tooltips. Let's check out the other methods of the Bootstrap tooltip plugin.

## show

This method reveals an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("show");

});

});

</script>

## hide

This method hides an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("hide");

});

});

</script>

## toggle

This method toggles an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("toggle");

});

});

</script>

## dispose

This method hides and destroys an element's tooltip (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("dispose");

});

});

</script>

## enable

This method gives an element's tooltip the ability to be shown. Tooltips are enabled by default.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("enable");

});

});

</script>

## disable

This method removes the ability for an element's tooltip to be shown. The tooltip will only be able to be shown if it is re-enabled.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("disable");

});

});

</script>

## toggleEnabled

This method toggles the ability for an element's tooltip to be shown or hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("toggleEnabled");

});

});

</script>

## update

This method updates the position of an element's tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myTooltip").tooltip("update");

});

});

</script>

## getInstance

This is a static method which allows you to get the tooltip instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-tooltip-instance-using-jquery)

<script>

$(document).ready(function(){

// Trigger the tooltip

$("#myTooltip").tooltip();

// Get tooltip instance on button click

$("#myBtn").click(function(){

var myTooltip = bootstrap.Tooltip.getInstance($("#myTooltip")[0]);

console.log(myTooltip);

// {\_element: a#myTooltip, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the tooltip instance associated with a DOM element, or create a new one in case if the tooltip wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-tooltip-instance-using-jquery)

<script>

$(document).ready(function(){

// Get or create tooltip instance on button click

$("#myBtn").click(function(){

var myTooltip = bootstrap.Tooltip.getOrCreateInstance($("#myTooltip")[0]);

console.log(myTooltip);

// {\_element: a#myTooltip, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## Events

Bootstrap's tooltip class includes few events for hooking into tooltip functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.tooltip | This event fires immediately when the show instance method is called. |
| shown.bs.tooltip | This event is fired when the tooltip has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.tooltip | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.tooltip | This event is fired when the tooltip has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| inserted.bs.tooltip | This event is fired after the show.bs.tooltip event when the tooltip template has been added to the DOM. |

The following example will display an alert message to the user when the fade out transition of the tooltip has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-tooltip-events-using-jquery)

<script>

$(document).ready(function(){

$("#myTooltip").on("hidden.bs.tooltip",function(){

alert("Tooltip has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-collapse.php)

# Bootstrap Tooltips

In this tutorial you will learn how to create tooltips with Bootstrap.

## Creating the Tooltips with Bootstrap

A tooltip is a small pop up that appears when user places the mouse pointer over an element such as link or buttons to provide hint or information about the element being hovered.

Tooltips can be very helpful for the new visitors of your website because they enable the user to know the purpose of icons and links by placing the mouse pointer over them.

### Step 1: Adding the Tooltip Markup

To create a tooltip, you need to add the data-bs-toggle="tooltip" attribute to an element. Tolltip text that would display on hover can be specified using the title attribute.

Here is the standard markup for adding a tolltip to a hyperlink.

<a href="#" data-bs-toggle="tooltip" title="Some text">Hover over me</a>

Similarly, you can add tooltips to the other elements such as buttons, icons, etc.

### Step 2: Enabling the Tooltips

Tooltips can be enabled via JavaScript — just call the tooltip() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the target element in your JavaScript code.

You can either initialize tooltips individually or all in one go. The following jQuery code will initialize all tooltips on the page by selecting them by their data-bs-toggle attribute.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-tooltips-via-jquery)

<script>

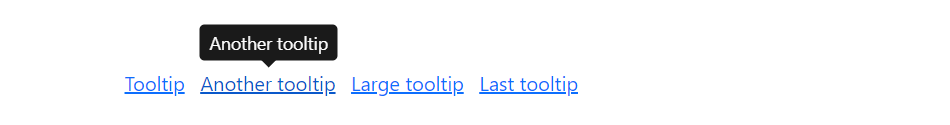
$(document).ready(function(){

$('[data-bs-toggle="tooltip"]').tooltip();

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-tooltips-via-jquery)

**Note:** For performance reasons the tooltip data-apis are opt in, means to use tooltips you must initialize them yourself. Also, tooltips with zero-length titles are never displayed, as well as triggering tooltips on hidden elements will not work.

**Tip:** Tooltips for .disabled or disabled elements must be triggered on a wrapper element. Also, when tooltips are triggered from hyperlinks that span multiple lines, it will be centered. You can use white-space: nowrap; on those hyperlinks to avoid this behavior.

## Setting the Directions of Tooltips

You can set tooltips to appear on top, right, bottom and left sides of an element.

### Positioning of Tooltips via Data Attributes

The following example will show you how to set the direction of tooltips via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tooltips-placement-via-data-attributes)

<a href="#"data-bs-toggle="tooltip"data-bs-placement="top"title="Tooltip on top">Tooltip on top</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="right"title="Tooltip on right">Tooltip on right</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="bottom"title="Tooltip on bottom">Tooltip on bottom</a>

<a href="#"data-bs-toggle="tooltip"data-bs-placement="left"title="Tooltip on left">Tooltip on left</a>

### Positioning of Tooltips via JavaScript

The following example will show you how to set the direction of tooltips via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=tooltips-placement-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

// Placement of tooltip on top

var tipTop = document.getElementById("tipTop");

var tooltipTop =newbootstrap.Tooltip(tipTop,{

placement :"top"

});

// Placement of tooltip on right

var tipRight = document.getElementById("tipRight");

var tooltipRight =newbootstrap.Tooltip(tipRight,{

placement :"right"

});

// Placement of tooltip on bottom

var tipBottom = document.getElementById("tipBottom");

var tooltipBottom =newbootstrap.Tooltip(tipBottom,{

placement :"bottom"

});

// Placement of tooltip on left

var tipLeft = document.getElementById("tipLeft");

var tooltipLeft =newbootstrap.Tooltip(tipLeft,{

placement :"left"

});

});

</script>

## Options

There are certain options which may be passed to tooltip() Bootstrap method to customize the functionality of the tooltip plugin.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the tooltip. |
| container | string | element | false | false | Appends the tooltip to a specific element.  Specify container: 'body' to avoid rendering problems in more complex components (like input groups, button groups, etc.) |
| delay | number | object | 0 | Time to delay in showing and hiding the tooltip (ms) — does not apply to manual trigger type.  If a number is supplied, delay is applied to both hide/show  Object structure is: delay: { "show": 500, "hide": 100 } |
| html | boolean | false | Insert HTML into the tooltip.  If true, HTML tags in the tooltip's title will be rendered in the tooltip. If false, innerText property will be used to insert content into the DOM.  Use simple text if you're worried about XSS attacks. |
| placement | string | function | 'top' | Sets the position of the tooltip — auto | top | bottom | left | right.  When auto value is specified, it will dynamically reorient the tooltip. |
| selector | string | false | false | If a selector is provided, tooltip objects will be attached to the specified targets.  This is normally used to apply tooltips to dynamically added DOM elements. |
| template | string | '<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-inner"></div></div>' | Base HTML to use when creating the tooltip.  The tooltip's title will be inserted into the .tooltip-inner element.  The .tooltip-arrow element will become the tooltip's arrow.  The outermost wrapper element should have the .tooltip class. |
| title | string | element | function | '' | Sets the default title value if title attribute isn't present. |
| trigger | string | 'hover focus' | Specify how tooltip is triggered — click | hover | focus | manual. You can pass multiple triggers; separated with a space.  The value manual indicates that the tooltip will be triggered programmatically via the .show(), .hide() and .toggle() methods; this value cannot be combined with any other trigger. |
| fallbackPlacements | array | ['top', 'right', 'bottom', 'left'] | Allows you to specify which placement Popper will use on fallback. |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the tooltip (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| customClass | string | function | '' | Add classes to the tooltip when it is shown. Please note that these classes will be added in addition to any classes specified in the template. To add multiple classes, separate them with spaces like: 'class1 class2'.  You can also pass a function that should return a single string containing additional class names. |
| sanitize | boolean | true | Enable or disable the sanitization. If activated 'template' and 'title' options will be sanitized. |
| allowList | object | [*Default value*](https://getbootstrap.com/docs/4.6/getting-started/javascript/#sanitizer) | Object which contains allowed attributes and tags. |
| sanitizeFn | null | function | null | Allows you to specify your own sanitize function. |
| offset | array | string | function | [0, 0] | Offset of the tooltip relative to its target. You can also pass a string in data attributes with comma separated values like: data-bs-offset="10,20" |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

You can set these options either through the use of data attributes or JavaScript. For setting the tooltips options via data attributes, just append the option name to data-bs- along with the correct value, like data-bs-animation="false", data-bs-placement="bottom" etc.

Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-customClass="my-class", use data-bs-custom-class="my-class".

However, JavaScript is the preferred way of setting these options as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php#passing-options) section below to know how to set the options for tooltips via JavaScript.

## Methods

These are the standard Bootstrap's tooltip methods:

## Passing options

You can additionally pass options to the tooltips using options object.

The following example will set the title text for the tooltips dynamically, if the value of the title attribute is omitted or missing from the selected elements:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-title-text-for-tooltips-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myTooltip");

var tooltip =newbootstrap.Tooltip(element,{

title :"It looks like title attribute is not present."

});

});

</script>

The following example will show you how to place the HTML content inside a tooltip:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=place-html-inside-a-tooltip-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myTooltip");

var tooltip =newbootstrap.Tooltip(element,{

title:"<h4><img src='images/smiley.png' width='30' alt='Smiley'> Hello, <b>I'm</b><i>Smiley!</i></h4>",

html:true

});

});

</script>

The following example will show you how to control the timing of showing and hiding the tooltip using the tooltip's delay option dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-tooltips-show-hide-time-delay-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var tinyTrigger = document.getElementById("tinyTooltip");

var largeTrigger = document.getElementById("largeTooltip");

// Showing and hiding tooltip with same speed

var tinyTooltip =newbootstrap.Tooltip(tinyTrigger,{

delay:100

});

// Showing and hiding tooltip with different speed

var largeTooltip =newbootstrap.Tooltip(largeTrigger,{

delay:{show:0, hide:2000}

});

});

</script>

The following example will show you how to create your own custom template for Bootstrap tooltips instead of using the default one dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-custom-template-for-tooltips-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myTooltip");

var tooltip =newbootstrap.Tooltip(element,{

template:'<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-head"><h3><i class="bi-info-circle"></i> Important Info</h3></div><div class="tooltip-inner"></div>'

});

});

</script>

The following example will insert the dynamically generated HTML code of the tooltip at the end of the #wrapper element instead of the default [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-tooltips-container-using-jquery)

<script>

$(document).ready(function(){

// Append tooltip HTML to wrapper element

$("#myTooltip").tooltip({

container:"#wrapper"

});

});

</script>

**Note:** Overriding the tooltip's default container option value does not produce any visible difference on page. To see the actual result you need to inspect the DOM. Press Ctrl+Shift+I (Windows / Linux) or Cmd+Opt+I (Mac) to open Developer tools or DOM Inspector.

Similarly, you can set other options for the tooltips. Let's check out the other methods of the Bootstrap tooltip plugin.

## show

This method reveals an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.show();

});

});

</script>

## hide

This method hides an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.hide();

});

});

</script>

## toggle

This method toggles an element's tooltip. This is considered a "manual" triggering of the tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.toggle();

});

});

</script>

## dispose

This method hides and destroys an element's tooltip (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.dispose();

});

});

</script>

## enable

This method gives an element's tooltip the ability to be shown. Tooltips are enabled by default.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.enable();

});

});

</script>

## disable

This method removes the ability for an element's tooltip to be shown. The tooltip will only be able to be shown if it is re-enabled.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.disable();

});

});

</script>

## toggleEnabled

This method toggles the ability for an element's tooltip to be shown or hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.toggleEnabled();

});

});

</script>

## update

This method updates the position of an element's tooltip.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-tooltip-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Create a tooltip instance

var myTooltip =newbootstrap.Tooltip(element);

btn.addEventListener("click",function(){

myTooltip.update();

});

});

</script>

## getInstance

This is a static method which allows you to get the tooltip instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-tooltip-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Trigger the tooltip

var myTooltip =newbootstrap.Tooltip(element);

// Get tooltip instance on button click

btn.addEventListener("click",function(){

var tooltip = bootstrap.Tooltip.getInstance(element);

console.log(tooltip);

// {\_element: a#myTooltip, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the tooltip instance associated with a DOM element, or create a new one in case if the tooltip wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-tooltip-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myTooltip");

// Get or create tooltip instance on button click

btn.addEventListener("click",function(){

var tooltip = bootstrap.Tooltip.getOrCreateInstance(element);

console.log(tooltip);

// {\_element: a#myTooltip, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## Events

Bootstrap's tooltip class includes few events for hooking into tooltip functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.tooltip | This event fires immediately when the show instance method is called. |
| shown.bs.tooltip | This event is fired when the tooltip has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.tooltip | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.tooltip | This event is fired when the tooltip has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| inserted.bs.tooltip | This event is fired after the show.bs.tooltip event when the tooltip template has been added to the DOM. |

The following example will display an alert message to the user when the fade out transition of the tooltip has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-tooltip-events-using-jquery)

<script>

$(document).ready(function(){

$("#myTooltip").on("hidden.bs.tooltip",function(){

alert("Tooltip has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-collapse.php)

# Bootstrap Popovers

In this tutorial you will learn how to create popovers with Bootstrap.

## Creating Popovers with Bootstrap

Popover is a small overlay of content that is used to display secondary information of any element when it is clicked by a user, like those on the iPad.

### Step 1: Adding the Popover Markup

To create a popover, you need to add the data-bs-toggle="popover" attribute to an element. Whereas, popover's title and its content that would display upon trigger or activation can be specified using the title and data-bs-content attribute respectively.

Here is the standard markup for adding a popover to a button.

<button type="button" data-bs-toggle="popover" title="Popover title" data-bs-content="Here's some amazing content.">Click to toggle popover</button>

Similarly, you can add popovers to the other elements such as links, icons, etc.

**Note:** For performance reasons the popovers data-apis are opt in like [tooltips](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php), means to use popovers you must initialize them yourself. Also, popovers with zero-length title and content values are never displayed, as well as triggering popovers on hidden elements will not work.

### Step 2: Enabling the Popovers

Popovers can be triggered via JavaScript — just call the popover() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the required element in your JavaScript code.

You can either initialize popovers individually or all in one go. The following example code will initialize all popovers on the page by selecting them by their data-bs-toggle attribute.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-popovers-via-jquery)

<script>

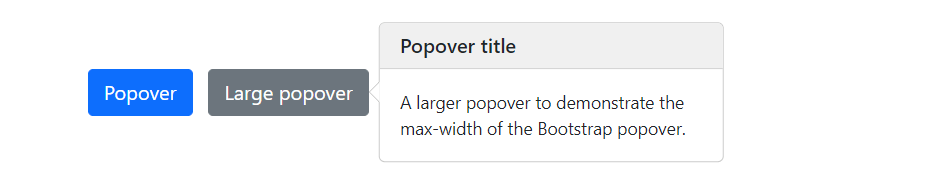
$(document).ready(function(){

$('[data-bs-toggle="popover"]').popover();

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-popovers-via-jquery)

## Setting the Directions of Popovers

You can set popovers to appear on top, right, bottom and left sides of an element.

### Positioning of Popovers via Data Attributes

The following example will show you how to set the direction of popovers via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=popovers-placement-via-data-attributes)

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="top"title="Popover title"data-bs-content="Popover on top">Popover on top</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="right"title="Popover title"data-bs-content="Popover on right.">Popover on right</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="bottom"title="Popover title"data-bs-content="Popover on bottom.">Popover on bottom</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="left"title="Popover title"data-bs-content="Popover on left.">Popover on left</button>

### Positioning of Popovers via JavaScript

The following example will show you how to set the direction of popovers via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=popovers-placement-via-jquery)

<script>

$(document).ready(function(){

$("#popTop").popover({placement :"top"});

$("#popRight").popover({placement :"right"});

$("#popBottom").popover({placement :"bottom"});

$("#popLeft").popover({ placement :"left"});

});

</script>

## Hiding the Popovers on Next Click

By default popovers are not hiding until you click the trigger element once again. You can use the focus trigger to hide the popovers when the user makes the next click.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismiss-popovers-on-next-click)

<a href="#"class="btn btn-primary"tabindex="0"data-bs-toggle="popover"data-bs-trigger="focus"title="Popover title"data-bs-content="Here's some amazing content.">Dismissible popover</a>

**Note:** To make this feature work properly across the browsers, you must use the [<a>](https://www.tutorialrepublic.com/html-reference/html-a-tag.php) tag, not the [<button>](https://www.tutorialrepublic.com/html-reference/html-button-tag.php) tag, and you also must include a tabindex attribute.

## Options

There are certain options which may be passed to popover() Bootstrap method to customize the functionality of the popover plugin.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the popover. |
| container | string | element | false | false | Appends the popover to a specific element.  Specify container: 'body' to avoid rendering problems in more complex components (like input groups, button groups, etc.) |
| content | string | element | function | '' | Sets default content value if data-bs-content attribute isn't present. |
| delay | number | object | 0 | Time to delay in showing and hiding the popover (ms) — does not apply to manual trigger type.  If a number is supplied, delay is applied to both hide/show  Object structure is: delay: { "show": 500, "hide": 100 } |
| html | boolean | false | Insert HTML into the popover.  If false, innerText property will be used to insert content into the DOM.  Use simple text if you're worried about XSS attacks. |
| placement | string | function | 'right' | Sets the position of the popover — auto | top | bottom | left | right.  When auto value is specified, it will dynamically reorient the popover. |
| selector | string | false | false | If a selector is provided, popover objects will be attached to the specified targets.  This is normally used to apply popovers to dynamically added DOM elements. |
| template | string | '<div class="popover"><div class="popover-arrow"></div><h3 class="popover-header"></h3><div class="popover-body"></div></div>' | Base HTML to use when creating the popover.  The popover's title will be inserted into the .popover-header element.  The popover's content will be inserted into the .popover-body element.  The .popover-arrow element will become the popover's arrow.  The outermost wrapper element should have the .popover class. |
| title | string | element | function | '' | Sets the default title value if title attribute isn't present. |
| trigger | string | 'click' | Specify how popover is triggered — click | hover | focus | manual. You can pass multiple triggers; separated with a space.  The value manual indicates that the popover will be triggered programmatically via the .show(), .hide() and .toggle() methods; this value cannot be combined with any other trigger. |
| fallbackPlacements | array | ['top', 'right', 'bottom', 'left'] | Allows you to specify which placement Popper will use on fallback. |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the popover (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| customClass | string | function | '' | Add classes to the popover when it is shown. Please note that these classes will be added in addition to any classes specified in the template. To add multiple classes, separate them with spaces like: 'class1 class2'.  You can also pass a function that should return a single string containing additional class names. |
| sanitize | boolean | true | Enable or disable the sanitization. If activated 'template', 'content' and 'title' options will be sanitized. |
| allowList | object | [*Default value*](https://getbootstrap.com/docs/4.6/getting-started/javascript/#sanitizer) | Object which contains allowed attributes and tags. |
| sanitizeFn | null | function | null | Allows you to specify your own sanitize function. |
| offset | array | string | function | [0, 8] | Offset of the popover relative to its target. You can also pass a string in data attributes with comma separated values like: data-bs-offset="10,20" |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

You can set these options either through the use of data attributes or JavaScript. For setting the popovers options via data attributes, just append the option name to data-bs- along with the correct value, like data-bs-animation="false", data-bs-placement="top" etc.

Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-customClass="my-class", use data-bs-custom-class="my-class".

However, JavaScript is the preferred way of setting these options as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php#passing-options) section below to know how to set the options for popovers via JavaScript.

## Methods

These are the standard Bootstrap's popover methods:

## Passing options

You can additionally pass options to the popovers using options object.

The following example will set the title text for the popovers dynamically, if the value of the title attribute is omitted or missing from the selected elements:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-title-text-for-popovers-using-jquery)

<script>

$(document).ready(function(){

$("#myPopover").popover({

title :"Default popover title"

});

});

</script>

The following example will show you how to place the HTML content inside a popover:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=place-html-inside-a-popover-using-jquery)

<script>

$(document).ready(function(){

$("#myPopover").popover({

title:'<h4 class="custom-title"><i class="bi-info-circle-fill"></i> Popover info</h4>',

content:'<p>This is a <em>simple example</em> demonstrating how to insert HTML code inside <strong>Bootstrap popover</strong>.</p>',

html:true

});

});

</script>

The following example will show you how to control the timing of showing and hiding the popover using the popover's delay option dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-popovers-show-hide-time-delay-using-jquery)

<script>

$(document).ready(function(){

// Show and hide popover with same speed

$("#tinyPopover").popover({

delay:100

});

// Show and hide popover with different speed

$("#largePopover").popover({

delay:{show:0, hide:2000}

});

});

</script>

The following example will show you how to create your own custom template for Bootstrap popovers instead of using the default one dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-custom-template-for-popovers-using-jquery)

<script>

$(document).ready(function(){

$('[data-bs-toggle="popover"]').popover({

template:'<div class="popover"><div class="popover-arrow"></div><h3 class="popover-header"></h3><div class="popover-body"></div><div class="popover-footer"><a class="btn btn-secondary btn-sm close">Close</a></div></div>'

});

// Close popover on button click

$(document).on("click",".popover .close",function(){

$(this).parents(".popover").popover("hide");

});

});

</script>

The following example will insert the dynamically generated HTML code of the popover at the end of the #wrapper element instead of the default [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-popovers-container-using-jquery)

<script>

$(document).ready(function(){

// Append popover HTML to wrapper element

$("#myPopover").popover({

container:"#wrapper"

});

});

</script>

**Note:** Overriding the popover's default container option value does not produce any visible difference on page. To see the actual result you need to inspect the DOM. Press Ctrl+Shift+I (Windows / Linux) or Cmd+Opt+I (Mac) to open Developer tools or DOM Inspector.

Similarly, you can set other options for the popovers. Let's check out the other methods of the Bootstrap popover plugin.

## show

This method reveals an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("show");

});

});

</script>

## hide

This method hides an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("hide");

});

});

</script>

## toggle

This method toggles an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("toggle");

});

});

</script>

## dispose

This method hides and destroys an element's popover (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("dispose");

});

});

</script>

## enable

This method gives an element's popover the ability to be shown. Popovers are enabled by default.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("enable");

});

});

</script>

## disable

This method removes the ability for an element's popover to be shown. The popover will only be able to be shown if it is re-enabled.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("disable");

});

});

</script>

## toggleEnabled

This method toggles the ability for an element's popover to be shown or hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("toggleEnabled");

});

});

</script>

## update

This method updates the position of an element's popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myPopover").popover("update");

});

});

</script>

## getInstance

This is a static method which allows you to get the popover instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-popover-instance-using-jquery)

<script>

$(document).ready(function(){

// Trigger the popover

$("#myPopover").popover();

// Get popover instance on button click

$("#myBtn").click(function(){

var myPopover = bootstrap.Popover.getInstance($("#myPopover")[0]);

console.log(myPopover);

// {\_element: button#myPopover.btn.btn-primary.btn-lg, \_isEnabled: true, \_timeout: 0, \_hoverState: null, \_activeTrigger: {…}, …}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the popover instance associated with a DOM element, or create a new one in case if the popover wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-popover-instance-using-jquery)

<script>

$(document).ready(function(){

// Get or create popover instance on button click

$("#myBtn").click(function(){

var myPopover = bootstrap.Popover.getOrCreateInstance($("#myPopover")[0]);

console.log(myPopover);

// {\_element: button#myPopover.btn.btn-primary.btn-lg, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## Events

Bootstrap's popover class includes few events for hooking into popover functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.popover | This event fires immediately when the show instance method is called. |
| shown.bs.popover | This event is fired when the popover has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.popover | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.popover | This event is fired when the popover has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| inserted.bs.popover | This event is fired after the show.bs.popover event when the popover template has been added to the DOM. |

The following example will display an alert message to the user when the fade out transition of the popover has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-popover-events-using-jquery)

<script>

$(document).ready(function(){

// Initialize popover

$("#myPopover").popover();

// Show alert when the popover has finished being hidden

$("#myPopover").on("hidden.bs.popover",function(){

alert("Popover has been completely closed.");

});

});

</script>

# Bootstrap Popovers

In this tutorial you will learn how to create popovers with Bootstrap.

## Creating Popovers with Bootstrap

Popover is a small overlay of content that is used to display secondary information of any element when it is clicked by a user, like those on the iPad.

### Step 1: Adding the Popover Markup

To create a popover, you need to add the data-bs-toggle="popover" attribute to an element. Whereas, popover's title and its content that would display upon trigger or activation can be specified using the title and data-bs-content attribute respectively.

Here is the standard markup for adding a popover to a button.

<button type="button" data-bs-toggle="popover" title="Popover title" data-bs-content="Here's some amazing content.">Click to toggle popover</button>

Similarly, you can add popovers to the other elements such as links, icons, etc.

**Note:** For performance reasons the popovers data-apis are opt in like [tooltips](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php), means to use popovers you must initialize them yourself. Also, popovers with zero-length title and content values are never displayed, as well as triggering popovers on hidden elements will not work.

### Step 2: Enabling the Popovers

Popovers can be triggered via JavaScript — just call the popover() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the required element in your JavaScript code.

You can either initialize popovers individually or all in one go. The following example code will initialize all popovers on the page by selecting them by their data-bs-toggle attribute.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-popovers-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var popoverTriggerList =[].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'));

var popoverList = popoverTriggerList.map(function(element){

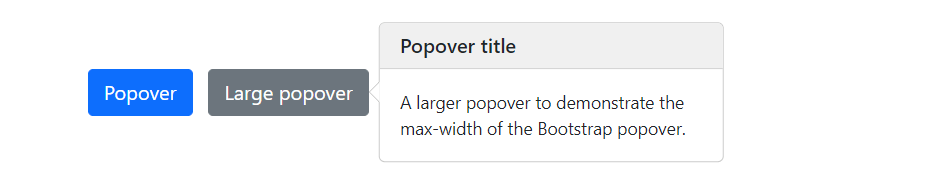
returnnewbootstrap.Popover(element);

});

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=enable-popovers-via-jquery)

## Setting the Directions of Popovers

You can set popovers to appear on top, right, bottom and left sides of an element.

### Positioning of Popovers via Data Attributes

The following example will show you how to set the direction of popovers via data attributes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=popovers-placement-via-data-attributes)

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="top"title="Popover title"data-bs-content="Popover on top">Popover on top</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="right"title="Popover title"data-bs-content="Popover on right.">Popover on right</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="bottom"title="Popover title"data-bs-content="Popover on bottom.">Popover on bottom</button>

<button type="button"class="btn btn-secondary"data-bs-toggle="popover"data-bs-placement="left"title="Popover title"data-bs-content="Popover on left.">Popover on left</button>

### Positioning of Popovers via JavaScript

The following example will show you how to set the direction of popovers via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=popovers-placement-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

// Placement of popover on top

var popTop = document.getElementById("popTop");

var popoverTop =newbootstrap.Popover(popTop,{

placement :"top"

});

// Placement of popover on right

var popRight = document.getElementById("popRight");

var popoverRight =newbootstrap.Popover(popRight,{

placement :"right"

});

// Placement of popover on bottom

var popBottom = document.getElementById("popBottom");

var popoverBottom =newbootstrap.Popover(popBottom,{

placement :"bottom"

});

// Placement of popover on left

var popLeft = document.getElementById("popLeft");

var popoverLeft =newbootstrap.Popover(popLeft,{

placement :"left"

});

});

</script>

## Hiding the Popovers on Next Click

By default popovers are not hiding until you click the trigger element once again. You can use the focus trigger to hide the popovers when the user makes the next click.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismiss-popovers-on-next-click)

<a href="#"class="btn btn-primary"tabindex="0"data-bs-toggle="popover"data-bs-trigger="focus"title="Popover title"data-bs-content="Here's some amazing content.">Dismissible popover</a>

**Note:** To make this feature work properly across the browsers, you must use the [<a>](https://www.tutorialrepublic.com/html-reference/html-a-tag.php) tag, not the [<button>](https://www.tutorialrepublic.com/html-reference/html-button-tag.php) tag, and you also must include a tabindex attribute.

## Options

There are certain options which may be passed to popover() Bootstrap method to customize the functionality of the popover plugin.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the popover. |
| container | string | element | false | false | Appends the popover to a specific element.  Specify container: 'body' to avoid rendering problems in more complex components (like input groups, button groups, etc.) |
| content | string | element | function | '' | Sets default content value if data-bs-content attribute isn't present. |
| delay | number | object | 0 | Time to delay in showing and hiding the popover (ms) — does not apply to manual trigger type.  If a number is supplied, delay is applied to both hide/show  Object structure is: delay: { "show": 500, "hide": 100 } |
| html | boolean | false | Insert HTML into the popover.  If false, innerText property will be used to insert content into the DOM.  Use simple text if you're worried about XSS attacks. |
| placement | string | function | 'right' | Sets the position of the popover — auto | top | bottom | left | right.  When auto value is specified, it will dynamically reorient the popover. |
| selector | string | false | false | If a selector is provided, popover objects will be attached to the specified targets.  This is normally used to apply popovers to dynamically added DOM elements. |
| template | string | '<div class="popover"><div class="popover-arrow"></div><h3 class="popover-header"></h3><div class="popover-body"></div></div>' | Base HTML to use when creating the popover.  The popover's title will be inserted into the .popover-header element.  The popover's content will be inserted into the .popover-body element.  The .popover-arrow element will become the popover's arrow.  The outermost wrapper element should have the .popover class. |
| title | string | element | function | '' | Sets the default title value if title attribute isn't present. |
| trigger | string | 'click' | Specify how popover is triggered — click | hover | focus | manual. You can pass multiple triggers; separated with a space.  The value manual indicates that the popover will be triggered programmatically via the .show(), .hide() and .toggle() methods; this value cannot be combined with any other trigger. |
| fallbackPlacements | array | ['top', 'right', 'bottom', 'left'] | Allows you to specify which placement Popper will use on fallback. |
| boundary | string | element | 'clippingParents' | Overflow constraint boundary of the popover (applies only to Popper's preventOverflow modifier). It can also accept an HTMLElement reference (via JavaScript only). |
| customClass | string | function | '' | Add classes to the popover when it is shown. Please note that these classes will be added in addition to any classes specified in the template. To add multiple classes, separate them with spaces like: 'class1 class2'.  You can also pass a function that should return a single string containing additional class names. |
| sanitize | boolean | true | Enable or disable the sanitization. If activated 'template', 'content' and 'title' options will be sanitized. |
| allowList | object | [*Default value*](https://getbootstrap.com/docs/4.6/getting-started/javascript/#sanitizer) | Object which contains allowed attributes and tags. |
| sanitizeFn | null | function | null | Allows you to specify your own sanitize function. |
| offset | array | string | function | [0, 8] | Offset of the popover relative to its target. You can also pass a string in data attributes with comma separated values like: data-bs-offset="10,20" |
| popperConfig | null | object | function | null | Allows you to change Bootstrap's default Popper config, see [Popper's configuration](https://popper.js.org/docs/v1/#Popper.Defaults). |

You can set these options either through the use of data attributes or JavaScript. For setting the popovers options via data attributes, just append the option name to data-bs- along with the correct value, like data-bs-animation="false", data-bs-placement="top" etc.

Also, when passing the options via data attributes make sure to change the case type of the option name from **camelCase** to **kebab-case**. For example, instead of using data-bs-customClass="my-class", use data-bs-custom-class="my-class".

However, JavaScript is the preferred way of setting these options as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php#passing-options) section below to know how to set the options for popovers via JavaScript.

## Methods

These are the standard Bootstrap's popover methods:

## Passing options

You can additionally pass options to the popovers using options object.

The following example will set the title text for the popovers dynamically, if the value of the title attribute is omitted or missing from the selected elements:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-title-text-for-popovers-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myPopover");

var tooltip =newbootstrap.Popover(element,{

title :"Default popover title"

});

});

</script>

The following example will show you how to place the HTML content inside a popover:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=place-html-inside-a-popover-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myPopover");

var popover =newbootstrap.Popover(element,{

title:'<h4 class="custom-title"><i class="bi-info-circle-fill"></i> Popover info</h4>',

content:'<p>This is a <em>simple example</em> demonstrating how to insert HTML code inside <strong>Bootstrap popover</strong>.</p>',

html:true

});

});

</script>

The following example will show you how to control the timing of showing and hiding the popover using the popover's delay option dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-popovers-show-hide-time-delay-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var tinyTrigger = document.getElementById("tinyPopover");

var largeTrigger = document.getElementById("largePopover");

// Showing and hiding popover with same speed

var tinyPopover =newbootstrap.Popover(tinyTrigger,{

delay:100

});

// Showing and hiding popover with different speed

var largePopover =newbootstrap.Popover(largeTrigger,{

delay:{show:0, hide:2000}

});

});

</script>

The following example will show you how to create your own custom template for Bootstrap popovers instead of using the default one dynamically via JavaScript.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=create-custom-template-for-popovers-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var popoverTriggerList =[].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'));

var popoverList = popoverTriggerList.map(function(element){

returnnewbootstrap.Popover(element,{

template:'<div class="popover"><div class="popover-arrow"></div><h3 class="popover-header"></h3><div class="popover-body"></div><div class="popover-footer"><a class="btn btn-secondary btn-sm close">Close</a></div></div>'

});

});

});

// Close popover on button click

document.addEventListener("click",function(e){

if(e.target && e.target.classList.contains("close")){

var popover = bootstrap.Popover.getInstance(e.target.closest(".popover"));

popover.hide();

}

});

</script>

The following example will insert the dynamically generated HTML code of the popover at the end of the #wrapper element instead of the default [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-popovers-container-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myPopover");

// Append popover HTML to wrapper element

var popover =newbootstrap.Popover(element,{

container:"#wrapper"

});

});

</script>

**Note:** Overriding the popover's default container option value does not produce any visible difference on page. To see the actual result you need to inspect the DOM. Press Ctrl+Shift+I (Windows / Linux) or Cmd+Opt+I (Mac) to open Developer tools or DOM Inspector.

Similarly, you can set other options for the popovers. Let's check out the other methods of the Bootstrap popover plugin.

## show

This method reveals an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.show();

});

});

</script>

## hide

This method hides an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.hide();

});

});

</script>

## toggle

This method toggles an element's popover. This is considered a "manual" triggering of the popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.toggle();

});

});

</script>

## dispose

This method hides and destroys an element's popover (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.dispose();

});

});

</script>

## enable

This method gives an element's popover the ability to be shown. Popovers are enabled by default.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.enable();

});

});

</script>

## disable

This method removes the ability for an element's popover to be shown. The popover will only be able to be shown if it is re-enabled.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.disable();

});

});

</script>

## toggleEnabled

This method toggles the ability for an element's popover to be shown or hidden.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.toggleEnabled();

});

});

</script>

## update

This method updates the position of an element's popover.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-popover-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Create a popover instance

var myPopover =newbootstrap.Popover(element);

btn.addEventListener("click",function(){

myPopover.update();

});

});

</script>

## getInstance

This is a static method which allows you to get the popover instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-popover-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Trigger the popover

var myPopover =newbootstrap.Popover(element);

// Get popover instance on button click

btn.addEventListener("click",function(){

var popover = bootstrap.Popover.getInstance(element);

console.log(popover);

// {\_element: button#myPopover.btn.btn-primary.btn-lg, \_isEnabled: true, \_timeout: 0, \_hoverState: null, \_activeTrigger: {…}, …}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the popover instance associated with a DOM element, or create a new one in case if the popover wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-popover-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myPopover");

// Get or create popover instance on button click

btn.addEventListener("click",function(){

var popover = bootstrap.Popover.getOrCreateInstance(element);

console.log(popover);

// {\_element: button#myPopover.btn.btn-primary.btn-lg, \_isEnabled: true, \_timeout: 0, \_hoverState: "", \_activeTrigger: {…}, …}

});

});

</script>

## Events

Bootstrap's popover class includes few events for hooking into popover functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.popover | This event fires immediately when the show instance method is called. |
| shown.bs.popover | This event is fired when the popover has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.popover | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.popover | This event is fired when the popover has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| inserted.bs.popover | This event is fired after the show.bs.popover event when the popover template has been added to the DOM. |

The following example will display an alert message to the user when the fade out transition of the popover has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-popover-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myPopover = document.getElementById("myPopover");

// Initialize popover

var popover =newbootstrap.Popover(myPopover);

// Show alert when the popover has finished being hidden

myPopover.addEventListener("hidden.bs.popover",function(){

alert("Popover has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-tooltips.php)

# Bootstrap Alerts

In this tutorial you will learn how to create alerts messages with Bootstrap.

## Creating Alert Messages with Bootstrap

Alert boxes are used quite often to stand out the information that requires immediate attention of the end users such as warning, error or confirmation messages.

With Bootstrap you can easily create elegant alert messages for various purposes by adding the contextual classes (e.g., .alert-success, .alert-warning, .alert-info etc.) to the .alert base class. You can also add an optional close button to dismiss any alert.

Bootstrap provides total 8 different types of alerts. The following example will show you the most commonly used alerts, which are: success, error or danger, warning and info alerts.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

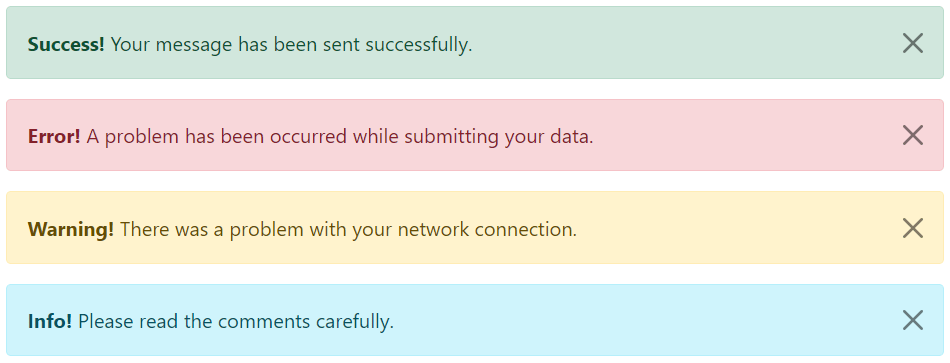
<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

Here're the remaining four Bootstrap alerts that can be used for various purposes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> This is a simple primary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> This is a simple secondary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> This is a simple dark alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

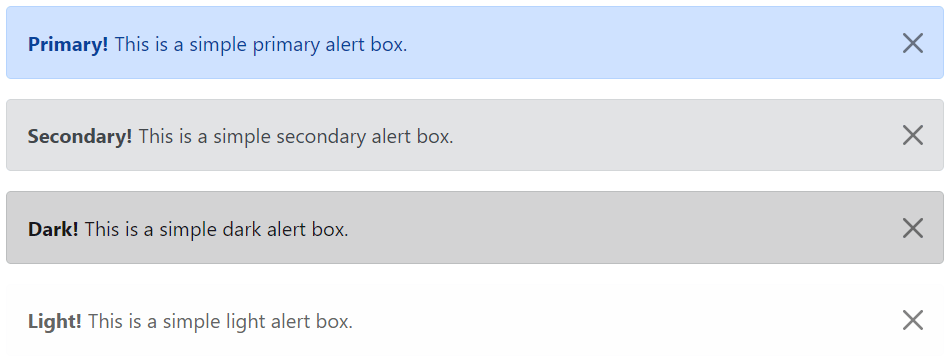
<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> This is a simple light alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

**Tip:** The .fade and .show classes on the .alert element enable the fading transition effect while closing the alert boxes. If you don't want animation just removes these classes. Also, the class .alert-dismissible is required on the .alert element for proper positioning of the .btn-close. If your alert doesn't have a close button you can skip this class.

## Adding Icons to Bootstrap Alerts

You can also place icons inside Bootstrap alerts. You can either use [Bootstrap icons](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-icons.php) or third-party icons like Font Awesome. Let's take a look at the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible d-flex align-items-center fade show">

<i class="bi-check-circle-fill"></i>

<strong class="mx-2">Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-octagon-fill"></i>

<strong class="mx-2">Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-triangle-fill"></i>

<strong class="mx-2">Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible d-flex align-items-center fade show">

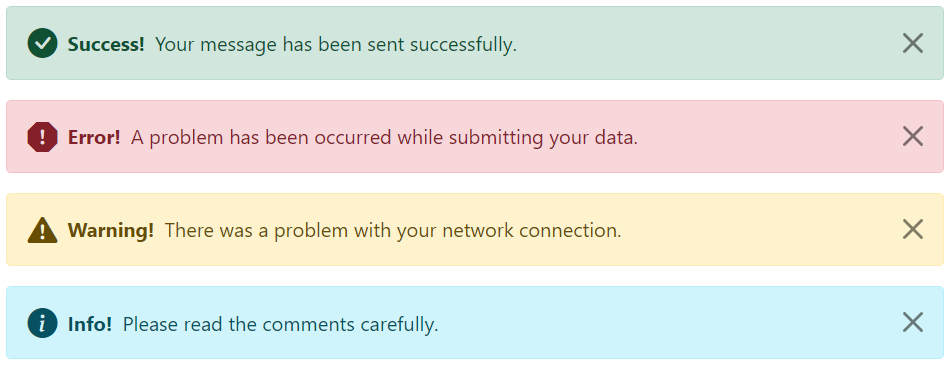
<i class="bi-info-circle-fill"></i>

<strong class="mx-2">Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

## Additional Content inside Alerts

You can also place additional HTML elements like headings, paragraphs and dividers inside an alert. To manage spacing between the elements you can use margin utility classes, as here:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

<div class="alert alert-danger alert-dismissible fade show">

<h4 class="alert-heading"><i class="bi-exclamation-octagon-fill"></i> Oops! Something went wrong.</h4>

<p>Please enter a valid value in all the required fields before proceeding. If you need any help just place the mouse pointer above info icon next to the form field.</p>

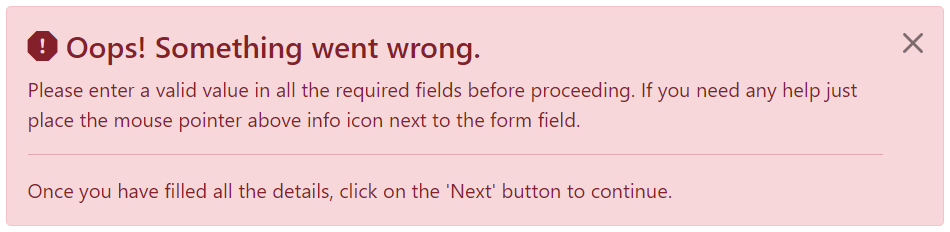
<hr>

<p class="mb-0">Once you have filled all the details, click on the 'Next' button to continue.</p>

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

## Matching Links Color inside Alerts

You can apply the utility class .alert-link to the links inside any alert boxes to quickly create matching colored links, as shown in the example below:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

Similarly, you can match links inside other alert boxes. Let's try out the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=matching-links-inside-different-alerts)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> A simple success alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A simple danger alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> A simple info alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> A simple primary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> A simple secondary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> A simple dark alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> A simple light alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

## Closing Alerts via Data Attribute

Data attributes provides a simple and easy way to add close functionality to the alert boxes.

Just add the data-bs-dismiss="alert" to the close button and it will automatically enable the dismissal of the containing alert message box. Also, add the class .alert-dismissible to the .alert element for proper positioning of the .btn-close button.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-data-attributes)

<div class="alert alert-info alert-dismissible fade show">

<strong>Note!</strong> This is a simple example of dismissible alert.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

Use the [<button>](https://www.tutorialrepublic.com/html-reference/html-button-tag.php) element for creating the close button for consistent behavior across all devices.

## Closing Alerts via JavaScript

You may also dismiss an alert via JavaScript. Let's try out an example and see how it works:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("close");

});

});

</script>

## Methods

These are the standard bootstrap's alerts methods:

## close

This method closes an alert by removing it from the DOM. If the .fade and .show classes are present on the element, the alert will fade out before it is removed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=close-alerts-using-javascript)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("close");

});

});

</script>

## dispose

This method destroys an element's alert (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-alerts-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("dispose");

});

});

</script>

## getInstance

This is a static method which allows you to get the alert instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-alert-instance-using-jquery)

<script>

$(document).ready(function(){

// Create alert instance

$("#myAlert").alert();

// Get alert instance on button click

$("#myBtn").click(function(){

var myAlert = bootstrap.Alert.getInstance($("#myAlert")[0]);

console.log(myAlert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the alert instance associated with a DOM element, or create a new one in case if the alert wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-alert-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myAlert = bootstrap.Alert.getOrCreateInstance($("#myAlert")[0]);

console.log(myAlert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## Events

Bootstrap's alert class includes few events for hooking into alert functionality.

| **Event** | **Description** |
| --- | --- |
| close.bs.alert | This event fires immediately when the close instance method is called. |
| closed.bs.alert | This event is fired when the alert has been closed and CSS transitions have completed. |

The following example displays an alert message to the user when fade out transition of an alert message box has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-alert-events-using-jquery)

<script>

$(document).ready(function(){

$("#myAlert").on("closed.bs.alert",function(){

alert("Alert message box has been closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php)

# Bootstrap Alerts

In this tutorial you will learn how to create alerts messages with Bootstrap.

## Creating Alert Messages with Bootstrap

Alert boxes are used quite often to stand out the information that requires immediate attention of the end users such as warning, error or confirmation messages.

With Bootstrap you can easily create elegant alert messages for various purposes by adding the contextual classes (e.g., .alert-success, .alert-warning, .alert-info etc.) to the .alert base class. You can also add an optional close button to dismiss any alert.

Bootstrap provides total 8 different types of alerts. The following example will show you the most commonly used alerts, which are: success, error or danger, warning and info alerts.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

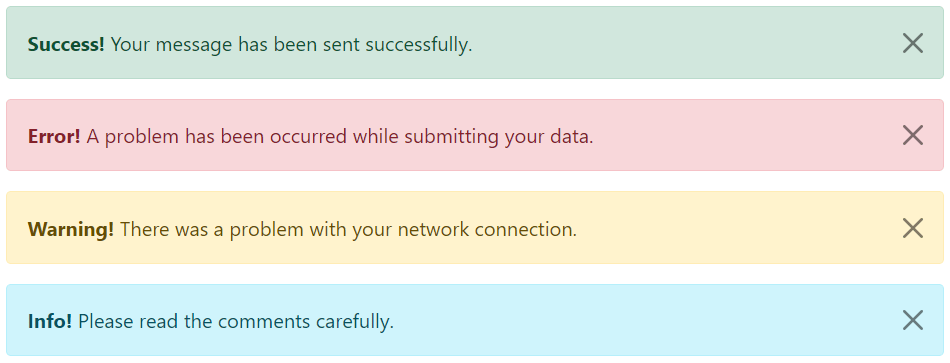
<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

Here're the remaining four Bootstrap alerts that can be used for various purposes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> This is a simple primary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> This is a simple secondary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> This is a simple dark alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

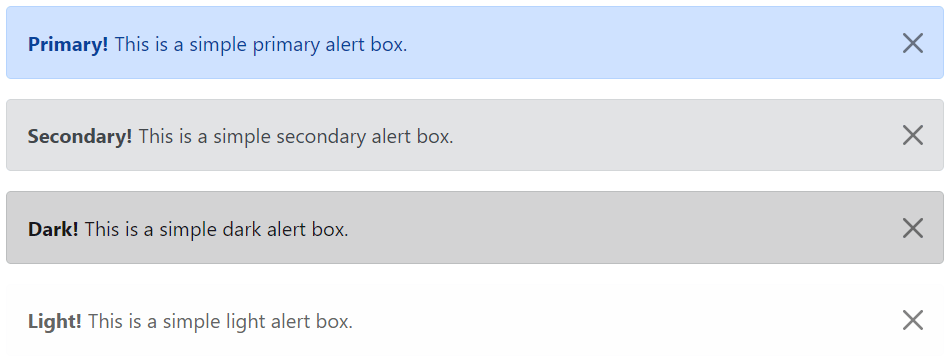
<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> This is a simple light alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

**Tip:** The .fade and .show classes on the .alert element enable the fading transition effect while closing the alert boxes. If you don't want animation just removes these classes. Also, the class .alert-dismissible is required on the .alert element for proper positioning of the .btn-close. If your alert doesn't have a close button you can skip this class.

## Adding Icons to Bootstrap Alerts

You can also place icons inside Bootstrap alerts. You can either use [Bootstrap icons](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-icons.php) or third-party icons like Font Awesome. Let's take a look at the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible d-flex align-items-center fade show">

<i class="bi-check-circle-fill"></i>

<strong class="mx-2">Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-octagon-fill"></i>

<strong class="mx-2">Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-triangle-fill"></i>

<strong class="mx-2">Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible d-flex align-items-center fade show">

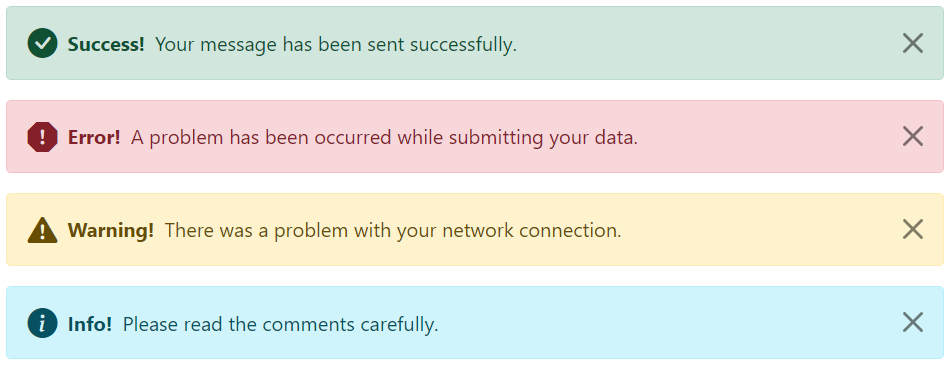
<i class="bi-info-circle-fill"></i>

<strong class="mx-2">Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

## Additional Content inside Alerts

You can also place additional HTML elements like headings, paragraphs and dividers inside an alert. To manage spacing between the elements you can use margin utility classes, as here:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

<div class="alert alert-danger alert-dismissible fade show">

<h4 class="alert-heading"><i class="bi-exclamation-octagon-fill"></i> Oops! Something went wrong.</h4>

<p>Please enter a valid value in all the required fields before proceeding. If you need any help just place the mouse pointer above info icon next to the form field.</p>

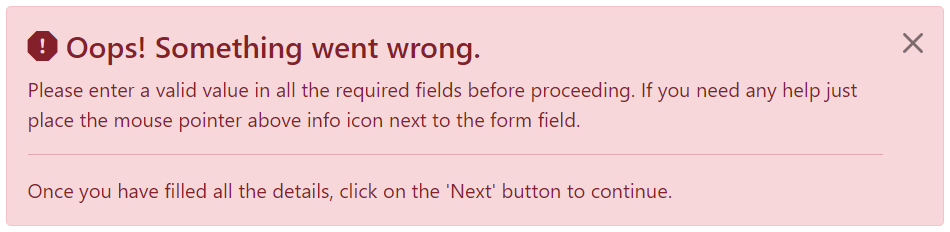
<hr>

<p class="mb-0">Once you have filled all the details, click on the 'Next' button to continue.</p>

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

## Matching Links Color inside Alerts

You can apply the utility class .alert-link to the links inside any alert boxes to quickly create matching colored links, as shown in the example below:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

Similarly, you can match links inside other alert boxes. Let's try out the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=matching-links-inside-different-alerts)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> A simple success alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A simple danger alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> A simple info alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> A simple primary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> A simple secondary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> A simple dark alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> A simple light alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

## Closing Alerts via Data Attribute

Data attributes provides a simple and easy way to add close functionality to the alert boxes.

Just add the data-bs-dismiss="alert" to the close button and it will automatically enable the dismissal of the containing alert message box. Also, add the class .alert-dismissible to the .alert element for proper positioning of the .btn-close button.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-data-attributes)

<div class="alert alert-info alert-dismissible fade show">

<strong>Note!</strong> This is a simple example of dismissible alert.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

Use the [<button>](https://www.tutorialrepublic.com/html-reference/html-button-tag.php) element for creating the close button for consistent behavior across all devices.

## Closing Alerts via JavaScript

You may also dismiss an alert via JavaScript. Let's try out an example and see how it works:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myAlert");

// Create alert instance

var myAlert =newbootstrap.Alert(element);

btn.addEventListener("click",function(){

myAlert.close();

});

});

</script>

## Methods

These are the standard bootstrap's alerts methods:

## close

This method closes an alert by removing it from the DOM. If the .fade and .show classes are present on the element, the alert will fade out before it is removed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=close-alerts-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myAlert");

// Create alert instance

var myAlert =newbootstrap.Alert(element);

btn.addEventListener("click",function(){

myAlert.close();

});

});

</script>

## dispose

This method destroys an element's alert (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-alerts-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myAlert");

// Create alert instance

var myAlert =newbootstrap.Alert(element);

btn.addEventListener("click",function(){

myAlert.dispose();

});

});

</script>

## getInstance

This is a static method which allows you to get the alert instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-alert-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myAlert");

// Create alert instance

var myAlert =newbootstrap.Alert(element);

// Get tooltip instance on button click

btn.addEventListener("click",function(){

var alert = bootstrap.Alert.getInstance(element);

console.log(alert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the alert instance associated with a DOM element, or create a new one in case if the alert wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-alert-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myAlert");

btn.addEventListener("click",function(){

var myAlert = bootstrap.Alert.getOrCreateInstance(element);

console.log(myAlert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## Events

Bootstrap's alert class includes few events for hooking into alert functionality.

| **Event** | **Description** |
| --- | --- |
| close.bs.alert | This event fires immediately when the close instance method is called. |
| closed.bs.alert | This event is fired when the alert has been closed and CSS transitions have completed. |

The following example displays an alert message to the user when fade out transition of an alert message box has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-alert-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myAlert = document.getElementById("myAlert");

myAlert.addEventListener("closed.bs.alert",function(){

alert("Alert message box has been closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php)

# Bootstrap Alerts

In this tutorial you will learn how to create alerts messages with Bootstrap.

## Creating Alert Messages with Bootstrap

Alert boxes are used quite often to stand out the information that requires immediate attention of the end users such as warning, error or confirmation messages.

With Bootstrap you can easily create elegant alert messages for various purposes by adding the contextual classes (e.g., .alert-success, .alert-warning, .alert-info etc.) to the .alert base class. You can also add an optional close button to dismiss any alert.

Bootstrap provides total 8 different types of alerts. The following example will show you the most commonly used alerts, which are: success, error or danger, warning and info alerts.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

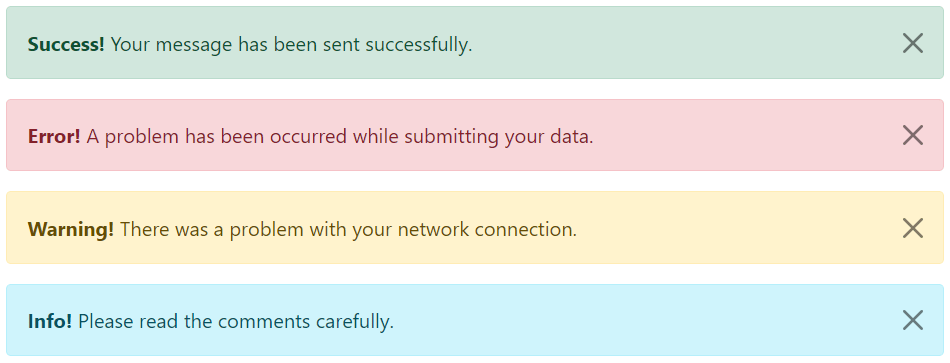
<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=common-alert-messages)

Here're the remaining four Bootstrap alerts that can be used for various purposes.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> This is a simple primary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> This is a simple secondary alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> This is a simple dark alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

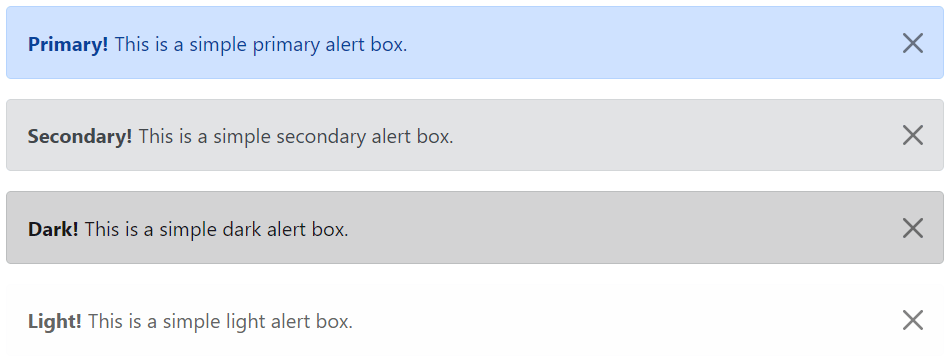
<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> This is a simple light alert box.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=new-alert-boxes)

**Tip:** The .fade and .show classes on the .alert element enable the fading transition effect while closing the alert boxes. If you don't want animation just removes these classes. Also, the class .alert-dismissible is required on the .alert element for proper positioning of the .btn-close. If your alert doesn't have a close button you can skip this class.

## Adding Icons to Bootstrap Alerts

You can also place icons inside Bootstrap alerts. You can either use [Bootstrap icons](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-icons.php) or third-party icons like Font Awesome. Let's take a look at the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible d-flex align-items-center fade show">

<i class="bi-check-circle-fill"></i>

<strong class="mx-2">Success!</strong> Your message has been sent successfully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-octagon-fill"></i>

<strong class="mx-2">Error!</strong> A problem has been occurred while submitting your data.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible d-flex align-items-center fade show">

<i class="bi-exclamation-triangle-fill"></i>

<strong class="mx-2">Warning!</strong> There was a problem with your network connection.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible d-flex align-items-center fade show">

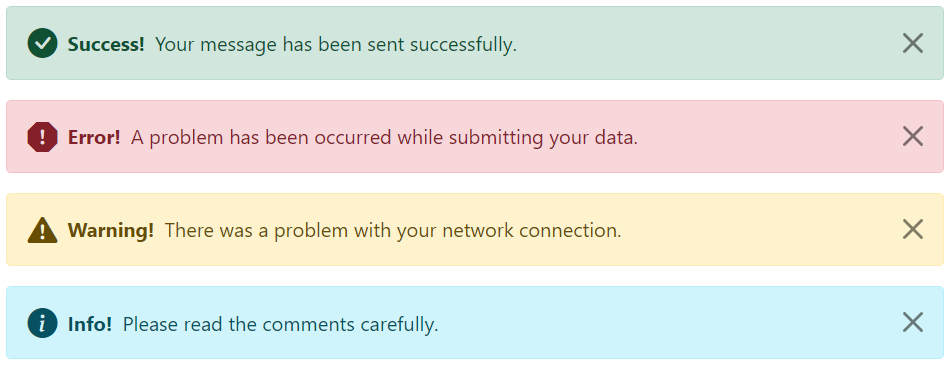
<i class="bi-info-circle-fill"></i>

<strong class="mx-2">Info!</strong> Please read the comments carefully.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-icons)

## Additional Content inside Alerts

You can also place additional HTML elements like headings, paragraphs and dividers inside an alert. To manage spacing between the elements you can use margin utility classes, as here:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

<div class="alert alert-danger alert-dismissible fade show">

<h4 class="alert-heading"><i class="bi-exclamation-octagon-fill"></i> Oops! Something went wrong.</h4>

<p>Please enter a valid value in all the required fields before proceeding. If you need any help just place the mouse pointer above info icon next to the form field.</p>

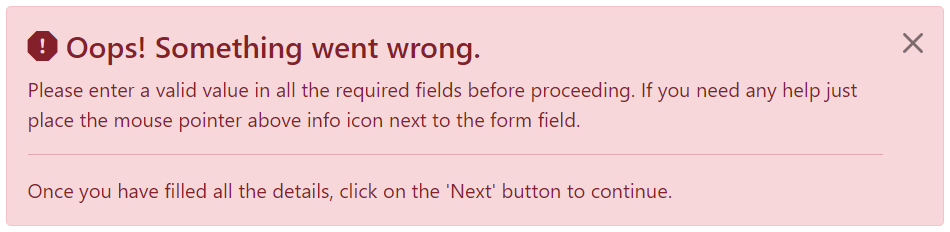
<hr>

<p class="mb-0">Once you have filled all the details, click on the 'Next' button to continue.</p>

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-with-additional-content)

## Matching Links Color inside Alerts

You can apply the utility class .alert-link to the links inside any alert boxes to quickly create matching colored links, as shown in the example below:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=alert-links)

Similarly, you can match links inside other alert boxes. Let's try out the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=matching-links-inside-different-alerts)

<!-- Success Alert -->

<div class="alert alert-success alert-dismissible fade show">

<strong>Success!</strong> A simple success alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Error Alert -->

<div class="alert alert-danger alert-dismissible fade show">

<strong>Error!</strong> A simple danger alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Warning Alert -->

<div class="alert alert-warning alert-dismissible fade show">

<strong>Warning!</strong> A simple warning alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Info Alert -->

<div class="alert alert-info alert-dismissible fade show">

<strong>Info!</strong> A simple info alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Primary Alert -->

<div class="alert alert-primary alert-dismissible fade show">

<strong>Primary!</strong> A simple primary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Secondary Alert -->

<div class="alert alert-secondary alert-dismissible fade show">

<strong>Secondary!</strong> A simple secondary alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Dark Alert -->

<div class="alert alert-dark alert-dismissible fade show">

<strong>Dark!</strong> A simple dark alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

<!-- Light Alert -->

<div class="alert alert-light alert-dismissible fade show">

<strong>Light!</strong> A simple light alert with <a href="#"class="alert-link">an example link</a>.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

## Closing Alerts via Data Attribute

Data attributes provides a simple and easy way to add close functionality to the alert boxes.

Just add the data-bs-dismiss="alert" to the close button and it will automatically enable the dismissal of the containing alert message box. Also, add the class .alert-dismissible to the .alert element for proper positioning of the .btn-close button.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-data-attributes)

<div class="alert alert-info alert-dismissible fade show">

<strong>Note!</strong> This is a simple example of dismissible alert.

<button type="button"class="btn-close"data-bs-dismiss="alert"></button>

</div>

Use the [<button>](https://www.tutorialrepublic.com/html-reference/html-button-tag.php) element for creating the close button for consistent behavior across all devices.

## Closing Alerts via JavaScript

You may also dismiss an alert via JavaScript. Let's try out an example and see how it works:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dismissal-of-alerts-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("close");

});

});

</script>

## Methods

These are the standard bootstrap's alerts methods:

## close

This method closes an alert by removing it from the DOM. If the .fade and .show classes are present on the element, the alert will fade out before it is removed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=close-alerts-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("close");

});

});

</script>

## dispose

This method destroys an element's alert (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-alerts-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myAlert").alert("dispose");

});

});

</script>

## getInstance

This is a static method which allows you to get the alert instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-alert-instance-using-jquery)

<script>

$(document).ready(function(){

// Create alert instance

$("#myAlert").alert();

// Get alert instance on button click

$("#myBtn").click(function(){

var myAlert = bootstrap.Alert.getInstance($("#myAlert")[0]);

console.log(myAlert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the alert instance associated with a DOM element, or create a new one in case if the alert wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-alert-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myAlert = bootstrap.Alert.getOrCreateInstance($("#myAlert")[0]);

console.log(myAlert);

// {\_element: div#myAlert.alert.alert-info.alert-dismissible.fade.show}

});

});

</script>

## Events

Bootstrap's alert class includes few events for hooking into alert functionality.

| **Event** | **Description** |
| --- | --- |
| close.bs.alert | This event fires immediately when the close instance method is called. |
| closed.bs.alert | This event is fired when the alert has been closed and CSS transitions have completed. |

The following example displays an alert message to the user when fade out transition of an alert message box has been fully completed.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-alert-events-using-jquery)

<script>

$(document).ready(function(){

$("#myAlert").on("closed.bs.alert",function(){

alert("Alert message box has been closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-popovers.php)

# Bootstrap Stateful Buttons

In this tutorial you will learn how to create toggle buttons with Bootstrap.

## Controlling Button States

In the previous section you've learnt about the [Bootstrap button styling and the modifications](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-buttons.php) as well as how to create [button groups and toolbars](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-button-groups.php). With Bootstrap you can do even more with the buttons like controlling the states of buttons, make checkbox and radio inputs behaves like toggle buttons, and so on. In the following section we will discuss them in detail.

## Creating Single Toggle Button

You can activate toggling (i.e. change the normal state of a button to a push state and vice versa) on a single button by simply adding the data attribute data-bs-toggle="button".

If you're pre-toggling a button, you must add the .active class manually.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=single-toggle-button)

<button type="button"class="btn btn-outline-primary"data-bs-toggle="button"autocomplete="off">Toggle button</button>

<button type="button"class="btn btn-outline-primary active"data-bs-toggle="button"autocomplete="off">Active toggle button</button>

<button type="button"class="btn btn-outline-primary"data-bs-toggle="button"autocomplete="off"disabled>Disabled toggle button</button>

— The toggle button upon clicking will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=single-toggle-button)

**Note:** The Mozilla Firefox browser persists the button state across page loads, to prevent this behavior, you may simply set the attribute autocomplete="off" on the form containing the buttons, or directly to the input or button element.

## Creating Checkbox Button Groups

You can also combine checkboxes to create checkbox style toggling on button groups. Let's try out the following example to understand how it basically works:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=checkbox-button-groups)

<div class="btn-group">

<input type="checkbox"class="btn-check"name="options"id="check1"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="check1">Checkbox 1</label>

<input type="checkbox"class="btn-check"name="options"id="check2"autocomplete="off">

<label class="btn btn-outline-primary"for="check2">Checkbox 2</label>

<input type="checkbox"class="btn-check"name="options"id="check3"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="check3">Checkbox 3</label>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=checkbox-button-groups)

**Note:** Don't use the .active class to preselect checkbox or radio in button groups, as it only changes the visual appearance to make them look like selected. To actually preselect them you will need to apply the checked attribute on the input element yourself.

## Creating Radio Button Groups

Similarly, you can create radio buttons style toggling on button groups, like this:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=radio-button-groups)

<div class="btn-group">

<input type="radio"class="btn-check"name="options"id="radio1"autocomplete="off">

<label class="btn btn-outline-primary"for="radio1">Radio 1</label>

<input type="radio"class="btn-check"name="options"id="radio2"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="radio2">Radio 2</label>

<input type="radio"class="btn-check"name="options"id="radio3"autocomplete="off">

<label class="btn btn-outline-primary"for="radio3">Radio 3</label>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=radio-button-groups)

## Methods

These are the standard bootstrap's buttons methods:

## toggle

This method toggles push state of the button. It changes the appearance of the button, and makes it look like that it has been activated. You can also enable auto toggling of a button by simply using the data-bs-toggle="button" attribute. Let's take a look at the following example:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toggle-button-state-using-jquery)

<script>

$(document).ready(function(){

$(".btn").click(function(){

$(this).button("toggle");

});

});

</script>

## dispose

This method destroys an element's button (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-buttons-using-jquery)

<script>

$(document).ready(function(){

$("#disposeBtn").click(function(){

var myButton = bootstrap.Button.getInstance($("#myButton")[0]);

console.log(myButton);

// {\_element: button#myButton.btn.btn-outline-primary.active}

myButton.dispose();

console.log(myButton);

// {\_element: null}

});

});

</script>

## getInstance

This is a static method which allows you to get the button instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-button-instance-using-jquery)

<script>

$(document).ready(function(){

$("#getBtn").click(function(){

var myButton = bootstrap.Button.getInstance($("#myButton")[0]);

console.log(myButton);

// {\_element: button#myButton.btn.btn-outline-primary.active}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the button instance associated with a DOM element, or create a new one in case if the button wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-button-instance-using-jquery)

<script>

$(document).ready(function(){

$("#getBtn").click(function(){

var myButton = bootstrap.Button.getOrCreateInstance($("#myButton")[0]);

console.log(myButton);

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-alerts.php)

# Bootstrap Stateful Buttons

In this tutorial you will learn how to create toggle buttons with Bootstrap.

## Controlling Button States

In the previous section you've learnt about the [Bootstrap button styling and the modifications](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-buttons.php) as well as how to create [button groups and toolbars](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-button-groups.php). With Bootstrap you can do even more with the buttons like controlling the states of buttons, make checkbox and radio inputs behaves like toggle buttons, and so on. In the following section we will discuss them in detail.

## Creating Single Toggle Button

You can activate toggling (i.e. change the normal state of a button to a push state and vice versa) on a single button by simply adding the data attribute data-bs-toggle="button".

If you're pre-toggling a button, you must add the .active class manually.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=single-toggle-button)

<button type="button"class="btn btn-outline-primary"data-bs-toggle="button"autocomplete="off">Toggle button</button>

<button type="button"class="btn btn-outline-primary active"data-bs-toggle="button"autocomplete="off">Active toggle button</button>

<button type="button"class="btn btn-outline-primary"data-bs-toggle="button"autocomplete="off"disabled>Disabled toggle button</button>

— The toggle button upon clicking will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=single-toggle-button)

**Note:** The Mozilla Firefox browser persists the button state across page loads, to prevent this behavior, you may simply set the attribute autocomplete="off" on the form containing the buttons, or directly to the input or button element.

## Creating Checkbox Button Groups

You can also combine checkboxes to create checkbox style toggling on button groups. Let's try out the following example to understand how it basically works:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=checkbox-button-groups)

<div class="btn-group">

<input type="checkbox"class="btn-check"name="options"id="check1"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="check1">Checkbox 1</label>

<input type="checkbox"class="btn-check"name="options"id="check2"autocomplete="off">

<label class="btn btn-outline-primary"for="check2">Checkbox 2</label>

<input type="checkbox"class="btn-check"name="options"id="check3"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="check3">Checkbox 3</label>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=checkbox-button-groups)

**Note:** Don't use the .active class to preselect checkbox or radio in button groups, as it only changes the visual appearance to make them look like selected. To actually preselect them you will need to apply the checked attribute on the input element yourself.

## Creating Radio Button Groups

Similarly, you can create radio buttons style toggling on button groups, like this:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=radio-button-groups)

<div class="btn-group">

<input type="radio"class="btn-check"name="options"id="radio1"autocomplete="off">

<label class="btn btn-outline-primary"for="radio1">Radio 1</label>

<input type="radio"class="btn-check"name="options"id="radio2"autocomplete="off"checked>

<label class="btn btn-outline-primary"for="radio2">Radio 2</label>

<input type="radio"class="btn-check"name="options"id="radio3"autocomplete="off">

<label class="btn btn-outline-primary"for="radio3">Radio 3</label>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=radio-button-groups)

## Methods

These are the standard bootstrap's buttons methods:

## toggle

This method toggles push state of the button. It changes the appearance of the button, and makes it look like that it has been activated. You can also enable auto toggling of a button by simply using the data-bs-toggle="button" attribute. Let's take a look at the following example:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toggle-button-state-using-jquery)

<script>

$(document).ready(function(){

$(".btn").click(function(){

$(this).button("toggle");

});

});

</script>

## dispose

This method destroys an element's button (i.e. removes stored data on the DOM element).

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-buttons-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var disposeBtn = document.getElementById("disposeBtn");

var myButton = document.getElementById("myButton");

disposeBtn.addEventListener("click",function(){

var bsButton = bootstrap.Button.getInstance(myButton);

console.log(bsButton);

// {\_element: button#myButton.btn.btn-outline-primary.active}

bsButton.dispose();

console.log(bsButton);

// {\_element: null}

});

});

</script>

## getInstance

This is a static method which allows you to get the button instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-button-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var getBtn = document.getElementById("getBtn");

var myButton = document.getElementById("myButton");

getBtn.addEventListener("click",function(){

var bsButton = bootstrap.Button.getInstance(myButton);

console.log(bsButton);

// {\_element: button#myButton.btn.btn-outline-primary.active}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the button instance associated with a DOM element, or create a new one in case if the button wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-button-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myBtn = document.getElementById("getBtn");

var myButton = document.getElementById("myButton");

getBtn.addEventListener("click",function(){

var bsButton = bootstrap.Button.getOrCreateInstance(myButton);

console.log(bsButton);

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-alerts.php)

**Bootstrap Carousel**

In this tutorial you will learn how to create carousels with Bootstrap.

**Creating Carousels with Bootstrap**

The carousel also known as slideshow or image slider is some of the best way of showcasing the huge amount of contents within a small space on the web pages. It is a dynamic presentation of contents where text and images are made visible or accessible to the user by cycling through several items.

The following example will show you how to build a simple carousel or slideshow with previous/next controls and indicators using the Bootstrap carousel plugin.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

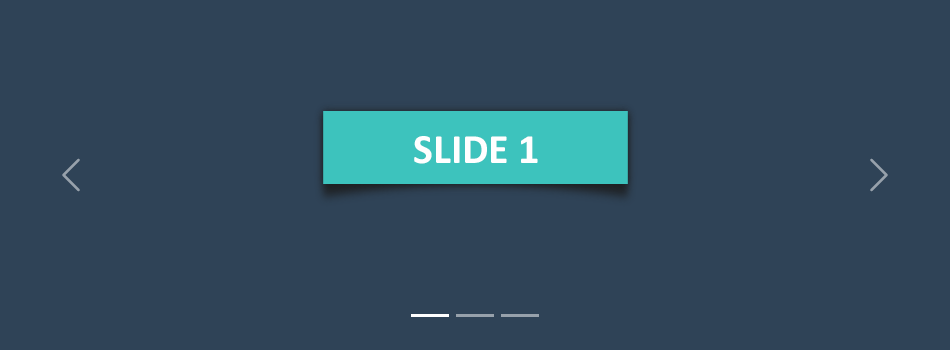
<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel)

**Creating Carousel with Captions**

You can also add captions to carousel slides easily with the .carousel-caption element within any .carousel-item. You can optionally use display utility classes to hide captions on smaller viewports.

Let's try out the following example to understand how it basically works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-with-captions)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

<div class="carousel-caption d-none d-md-block">

<h5>First slide label</h5>

<p>Some placeholder content for the first slide.</p>

</div>

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

<div class="carousel-caption d-none d-md-block">

<h5>Second slide label</h5>

<p>Some placeholder content for the second slide.</p>

</div>

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

<div class="carousel-caption d-none d-md-block">

<h5>Third slide label</h5>

<p>Some placeholder content for the third slide.</p>

</div>

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-with-captions)

**Tip:** The classes .d-none and .d-md-block on the .carousel-caption elements in the example above makes the carousel captions visible on the medium devices (i.e. viewport width ≥768px), but hide them on the smaller viewports.

**Creating Dark Variant of Carousel**

You can additionally add .carousel-dark to the .carousel to create darker controls, indicators, and captions in case your slides are lighter in color. Let's check out an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-dark)

<div id="myCarousel"class="carousel carousel-dark slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1-light.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2-light.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3-light.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-dark)

Check out the snippets section for examples of some [beautifully designed Bootstrap carousels](https://www.tutorialrepublic.com/snippets/gallery.php?tag=carousel).

You can also add captions such as heading or description text to the individual slides of the carousel, please check out the next example in the following section.

**Activate Carousels via Data Attributes**

With Bootstrap you can create carousels very easily via data attributes without writing a single line of JavaScript code. Let's take a look at the following example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-carousel-via-data-attributes)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

You might be wondering what this code was all about. Well, let's go through each part of this carousel example code one by one for a better understanding.

**Explanation of Code**

The Bootstrap carousel generally has three components — carousel indicators (*small rectangles*), carousel controls (*previous and next arrows*) and the carousel items or slides.

* The outermost container of every carousel (i.e. the .carousel element) requires a unique id (in our case id="myCarousel") so that it can be targeted by the carousel indicators (*line no-4,5,6*) and carousel controls (*line no-23,26*) to function properly.
* The data-bs-ride="carousel" attribute of the .carousel element tells the Bootstrap to start animating the carousel immediately when the page load.
* The data-bs-slide-to attribute (*line no-4,5,6*) move the slide position to a particular item (index beginning with 0) when clicking on the specific carousel indicator.
* The slides are specified within the .carousel-inner (*line no-10*) and the content of each slide is defined within the .carousel-item element that can be text and images.
* The data-bs-slide attribute on carousel controls (*line no-23,26*) accepts the keywords prev or next, which alters the slide position relative to its current position.

Rest of the thing is self-explanatory, for example, the .carousel element specifies the Bootstrap carousel, the .carousel-indicators element indicates how many slides are there in the carousel and which slide is currently active, .carousel-control-prev, .carousel-control-next elements defines previous and next controls to move between carousel slides, and so on.

**Tip:** It is required to add the class .active to one of the carousel slides (i.e. on the .carousel-item element), otherwise carousel will not be visible.

**Note:** The .slide class on the .carousel element adds CSS slide transition animation to the carousel that makes the carousel items slide when showing the new item.

**Activate Carousels via JavaScript**

You may also activate a carousel manually using the JavaScript — just call the carousel() method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the wrapper element in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-carousel-via-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel();

});

</script>

**Tip:** Manually activating a carousel via JavaScript can be helpful in a situation when you don't want your carousel to start sliding or animating at page load.

**Options**

There are certain options which can be passed to carousel() Bootstrap method to customize the functionality of a carousel. Options can be passed via data attributes or JavaScript.

For setting the modals options via data attributes, just append the option name to data-bs, such as data-bs-interval="3000", data-bs-pause="hover", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| interval | number | 5000 | Specifies the amount of time to delay (in milliseconds) between one slide to another in automatic cycling. If false, carousel will not automatically cycle. |
| keyboard | boolean | true | Specifies whether the carousel should react to keyboard events. By default it is true that means if carousel has focus you can go to its previous and next slide using the left and right arrow keys on the keyboard. |
| pause | string | boolean | 'hover' | Pauses the cycling of the carousel when mouse pointer enters the carousel and resumes the cycling when mouse pointer leaves the carousel, by default. If set to false, hovering over the carousel won't pause it. |
| ride | string | boolean | false | Autoplays the carousel after the user manually cycles the first item. If set to 'carousel', autoplays the carousel on load. |
| wrap | boolean | true | Specifies whether the carousel should cycle continuously or have hard stops (i.e stop at the last slide). |
| touch | boolean | true | Specifies whether the carousel should support left/right swipe interactions on touchscreen devices. |

Data attributes provides an easy way for setting the carousel options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-carousel.php#passing-options) section below to know how to set the options for carousels using JavaScript.

**Methods**

These are the standard bootstrap's carousels methods:

**Passing options**

You can additionally pass options to the carousels using options object.

The following example will **turn off auto sliding** in the carousel. By default Bootstrap carousel is started playing or sliding automatically when the page loads.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disable-carousel-autoplay-via-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel({

interval:false

});

});

</script>

The following example will stop carousel from auto-sliding once the last slide has been reached.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-carousel-auto-sliding-at-the-end-of-the-slide-via-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel({

wrap:false

});

});

</script>

**cycle**

This method start carousel for cycling through the items from left to right.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCarousel").carousel("cycle");

});

});

</script>

**pause**

This method stops the carousel from cycling through items.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCarousel").carousel("pause");

});

});

</script>

**prev**

This method cycles the carousel to the previous item.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCarousel").carousel("prev");

});

});

</script>

**next**

This method cycles the carousel to the next item.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCarousel").carousel("next");

});

});

</script>

**nextWhenVisible**

Don't cycle carousel to next when the page isn't visible or the carousel or its parent isn't visible.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel("nextWhenVisible");

});

</script>

**to**

This method cycles the carousel to a particular frame (start with 0, similar to an array).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

$("#myCarousel").carousel(2);

});

});

</script>

**dispose**

This method destroys an element's carousel (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-carousel-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCarousel = bootstrap.Carousel.getInstance($("#myCarousel")[0]);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

$("#myCarousel").carousel("dispose");

console.log(myCarousel);

// {\_element: null, \_items: null, \_interval: null, \_activeElement: null, \_isPaused: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the carousel instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-carousel-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCarousel = bootstrap.Carousel.getInstance($("#myCarousel")[0]);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the carousel instance associated with a DOM element, or create a new one in case if the carousel wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-carousel-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myCarousel = bootstrap.Carousel.getOrCreateInstance($("#myCarousel")[0]);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

});

});

</script>

**Events**

Bootstrap's carousel class includes few events for hooking into carousel functionality.

| **Event** | **Description** |
| --- | --- |
| slide.bs.carousel | This event fires immediately when the slide instance method is called. |
| slid.bs.carousel | This event is fired when the carousel has completed its slide transition. |

The following example displays an alert message when sliding transition of a carousel item has been fully completed. Let's try it out and see how it actually works.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-carousel-events-using-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").on("slid.bs.carousel",function(){

alert("The sliding transition of previous carousel item has been fully completed.");

});

});

</script>

**Bootstrap Carousel**

In this tutorial you will learn how to create carousels with Bootstrap.

**Creating Carousels with Bootstrap**

The carousel also known as slideshow or image slider is some of the best way of showcasing the huge amount of contents within a small space on the web pages. It is a dynamic presentation of contents where text and images are made visible or accessible to the user by cycling through several items.

The following example will show you how to build a simple carousel or slideshow with previous/next controls and indicators using the Bootstrap carousel plugin.

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

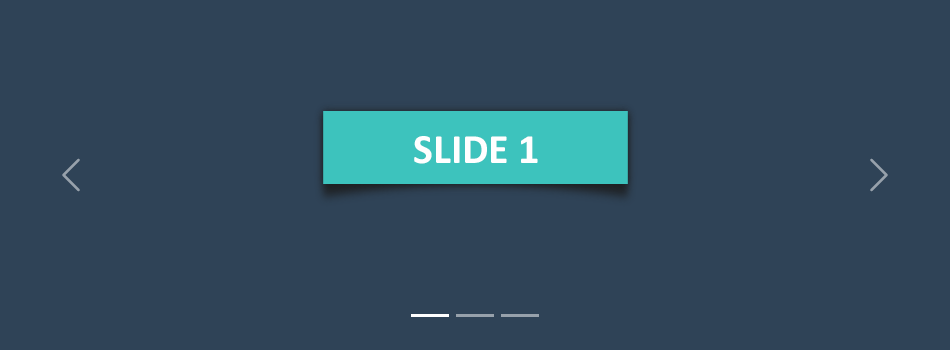
<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel)

**Creating Carousel with Captions**

You can also add captions to carousel slides easily with the .carousel-caption element within any .carousel-item. You can optionally use display utility classes to hide captions on smaller viewports.

Let's try out the following example to understand how it basically works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-with-captions)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

<div class="carousel-caption d-none d-md-block">

<h5>First slide label</h5>

<p>Some placeholder content for the first slide.</p>

</div>

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

<div class="carousel-caption d-none d-md-block">

<h5>Second slide label</h5>

<p>Some placeholder content for the second slide.</p>

</div>

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

<div class="carousel-caption d-none d-md-block">

<h5>Third slide label</h5>

<p>Some placeholder content for the third slide.</p>

</div>

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-with-captions)

**Tip:** The classes .d-none and .d-md-block on the .carousel-caption elements in the example above makes the carousel captions visible on the medium devices (i.e. viewport width ≥768px), but hide them on the smaller viewports.

**Creating Dark Variant of Carousel**

You can additionally add .carousel-dark to the .carousel to create darker controls, indicators, and captions in case your slides are lighter in color. Let's check out an example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-dark)

<div id="myCarousel"class="carousel carousel-dark slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1-light.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2-light.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3-light.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=carousel-dark)

Check out the snippets section for examples of some [beautifully designed Bootstrap carousels](https://www.tutorialrepublic.com/snippets/gallery.php?tag=carousel).

You can also add captions such as heading or description text to the individual slides of the carousel, please check out the next example in the following section.

**Activate Carousels via Data Attributes**

With Bootstrap you can create carousels very easily via data attributes without writing a single line of JavaScript code. Let's take a look at the following example:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-carousel-via-data-attributes)

<div id="myCarousel"class="carousel slide"data-bs-ride="carousel">

<!-- Carousel indicators -->

<ol class="carousel-indicators">

<li data-bs-target="#myCarousel"data-bs-slide-to="0"class="active"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="1"></li>

<li data-bs-target="#myCarousel"data-bs-slide-to="2"></li>

</ol>

<!-- Wrapper for carousel items -->

<div class="carousel-inner">

<div class="carousel-item active">

<img src="images/slide1.png"class="d-block w-100"alt="Slide 1">

</div>

<div class="carousel-item">

<img src="images/slide2.png"class="d-block w-100"alt="Slide 2">

</div>

<div class="carousel-item">

<img src="images/slide3.png"class="d-block w-100"alt="Slide 3">

</div>

</div>

<!-- Carousel controls -->

<a class="carousel-control-prev"href="#myCarousel"data-bs-slide="prev">

<span class="carousel-control-prev-icon"></span>

</a>

<a class="carousel-control-next"href="#myCarousel"data-bs-slide="next">

<span class="carousel-control-next-icon"></span>

</a>

</div>

You might be wondering what this code was all about. Well, let's go through each part of this carousel example code one by one for a better understanding.

**Explanation of Code**

The Bootstrap carousel generally has three components — carousel indicators (*small rectangles*), carousel controls (*previous and next arrows*) and the carousel items or slides.

* The outermost container of every carousel (i.e. the .carousel element) requires a unique id (in our case id="myCarousel") so that it can be targeted by the carousel indicators (*line no-4,5,6*) and carousel controls (*line no-23,26*) to function properly.
* The data-bs-ride="carousel" attribute of the .carousel element tells the Bootstrap to start animating the carousel immediately when the page load.
* The data-bs-slide-to attribute (*line no-4,5,6*) move the slide position to a particular item (index beginning with 0) when clicking on the specific carousel indicator.
* The slides are specified within the .carousel-inner (*line no-10*) and the content of each slide is defined within the .carousel-item element that can be text and images.
* The data-bs-slide attribute on carousel controls (*line no-23,26*) accepts the keywords prev or next, which alters the slide position relative to its current position.

Rest of the thing is self-explanatory, for example, the .carousel element specifies the Bootstrap carousel, the .carousel-indicators element indicates how many slides are there in the carousel and which slide is currently active, .carousel-control-prev, .carousel-control-next elements defines previous and next controls to move between carousel slides, and so on.

**Tip:** It is required to add the class .active to one of the carousel slides (i.e. on the .carousel-item element), otherwise carousel will not be visible.

**Note:** The .slide class on the .carousel element adds CSS slide transition animation to the carousel that makes the carousel items slide when showing the new item.

**Activate Carousels via JavaScript**

You may also activate a carousel manually using the JavaScript — just call the carousel() method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the wrapper element in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=activate-carousel-via-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel();

});

</script>

**Tip:** Manually activating a carousel via JavaScript can be helpful in a situation when you don't want your carousel to start sliding or animating at page load.

**Options**

There are certain options which can be passed to carousel() Bootstrap method to customize the functionality of a carousel. Options can be passed via data attributes or JavaScript.

For setting the modals options via data attributes, just append the option name to data-bs, such as data-bs-interval="3000", data-bs-pause="hover", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| interval | number | 5000 | Specifies the amount of time to delay (in milliseconds) between one slide to another in automatic cycling. If false, carousel will not automatically cycle. |
| keyboard | boolean | true | Specifies whether the carousel should react to keyboard events. By default it is true that means if carousel has focus you can go to its previous and next slide using the left and right arrow keys on the keyboard. |
| pause | string | boolean | 'hover' | Pauses the cycling of the carousel when mouse pointer enters the carousel and resumes the cycling when mouse pointer leaves the carousel, by default. If set to false, hovering over the carousel won't pause it. |
| ride | string | boolean | false | Autoplays the carousel after the user manually cycles the first item. If set to 'carousel', autoplays the carousel on load. |
| wrap | boolean | true | Specifies whether the carousel should cycle continuously or have hard stops (i.e stop at the last slide). |
| touch | boolean | true | Specifies whether the carousel should support left/right swipe interactions on touchscreen devices. |

Data attributes provides an easy way for setting the carousel options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-carousel.php#passing-options) section below to know how to set the options for carousels using JavaScript.

**Methods**

These are the standard bootstrap's carousels methods:

**Passing options**

You can additionally pass options to the carousels using options object.

The following example will **turn off auto sliding** in the carousel. By default Bootstrap carousel is started playing or sliding automatically when the page loads.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disable-carousel-autoplay-via-jquery)

<script>

$(document).ready(function(){

$("#myCarousel").carousel({

interval:false

});

});

</script>

The following example will stop carousel from auto-sliding once the last slide has been reached.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-carousel-auto-sliding-at-the-end-of-the-slide-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myCarousel");

var myCarousel =newbootstrap.Carousel(element,{

wrap:false

});

});

</script>

**cycle**

This method start carousel for cycling through the items from left to right.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

btn.addEventListener("click",function(){

myCarousel.cycle();

});

});

</script>

**pause**

This method stops the carousel from cycling through items.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

btn.addEventListener("click",function(){

myCarousel.pause();

});

});

</script>

**prev**

This method cycles the carousel to the previous item.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

btn.addEventListener("click",function(){

myCarousel.prev();

});

});

</script>

**next**

This method cycles the carousel to the next item.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

btn.addEventListener("click",function(){

myCarousel.next();

});

});

</script>

**nextWhenVisible**

Don't cycle carousel to next when the page isn't visible or the carousel or its parent isn't visible.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

myCarousel.nextWhenVisible();

});

</script>

**to**

This method cycles the carousel to a particular frame (start with 0, similar to an array).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-carousel-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

// Create a carousel instance

var myCarousel =newbootstrap.Carousel(element);

btn.addEventListener("click",function(){

myCarousel.to(2);

});

});

</script>

**dispose**

This method destroys an element's carousel (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-carousel-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

btn.addEventListener("click",function(){

var myCarousel = bootstrap.Carousel.getInstance(element);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

myCarousel.dispose();

console.log(myCarousel);

// {\_element: null, \_items: null, \_interval: null, \_activeElement: null, \_isPaused: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the carousel instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-carousel-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

btn.addEventListener("click",function(){

var myCarousel = bootstrap.Carousel.getInstance(element);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the carousel instance associated with a DOM element, or create a new one in case if the carousel wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-carousel-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myCarousel");

btn.addEventListener("click",function(){

var myCarousel = bootstrap.Carousel.getOrCreateInstance(element);

console.log(myCarousel);

// {\_element: div#myCarousel.carousel.slide, \_items: Array(3), \_interval: 9, \_activeElement: div.carousel-item.active, \_isPaused: false, …}

});

});

</script>

**Events**

Bootstrap's carousel class includes few events for hooking into carousel functionality.

| **Event** | **Description** |
| --- | --- |
| slide.bs.carousel | This event fires immediately when the slide instance method is called. |
| slid.bs.carousel | This event is fired when the carousel has completed its slide transition. |

The following example displays an alert message when sliding transition of a carousel item has been fully completed. Let's try it out and see how it actually works.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-carousel-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myCarousel = document.getElementById("myCarousel");

myCarousel.addEventListener("slid.bs.carousel",function(){

alert("The sliding transition of previous carousel item has been fully completed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-stateful-buttons.php)

# Bootstrap Typeahead

In this tutorial you will learn how to create typeaheads with Bootstrap.

## Creating Typeaheads with Bootstrap

The typeahead input fields are very popular in modern web forms. The main purpose of using typeahead is to improve the user experience by supplying hints or a list of possible choices based on the text they've entered while filling a form or searching something — like the Google instant search.

Typeahead functionality also saves time and reduces the number of potential errors, because the user has less likelihood of making a spelling mistake. Typeahead plugin has been dropped from the Bootstrap (v3.0+), in favor of using [Twitter typeahead](https://twitter.github.io/typeahead.js/).

Twitter typeaheads is a fast and fully-featured autocomplete library inspired by twitter.com's autocomplete search functionality. To create Twitter typeaheads first download typeahead.js from their official page — <https://twitter.github.io/typeahead.js/> and include in your project, after that you can turn any text-based [<input>](https://www.tutorialrepublic.com/html-reference/html-input-tag.php) element into a typeahead.

Twitter typeaheads require jQuery 1.9+ to work. Non-jQuery version is not available.

## Creating Twitter Typeahead with Local Dataset

The following example will show you how to create Twitter typeahead with local dataset.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=typeahead-with-local-dataset)

<script>

$(document).ready(function(){

// Defining the local dataset

var cars =['Audi','BMW','Bugatti','Ferrari','Ford','Lamborghini','Mercedes Benz','Porsche','Rolls-Royce','Volkswagen'];

// Constructing the suggestion engine

var cars =newBloodhound({

datumTokenizer: Bloodhound.tokenizers.whitespace,

queryTokenizer: Bloodhound.tokenizers.whitespace,

local: cars

});

// Initializing the typeahead

$('.typeahead').typeahead({

hint:true,

highlight:true,/\* Enable substring highlighting \*/

minLength:1/\* Specify minimum characters required for showing suggestions \*/

},

{

name:'cars',

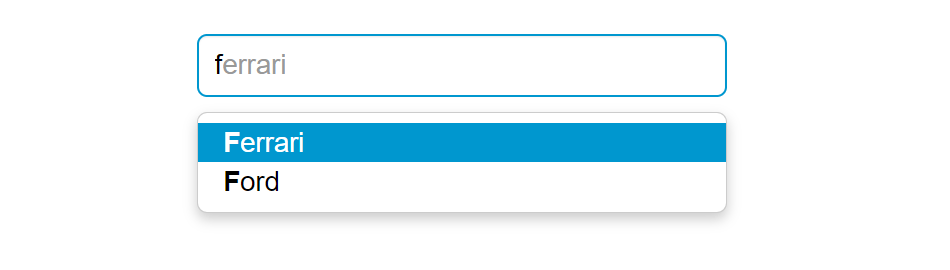
source: cars

});

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=typeahead-with-local-dataset)

**Note:** Bloodhound is the typeahead.js suggestion engine. It is very flexible and offers advanced functionalities such as prefetching remote data, fast lookups through intelligent caching using the browser [local storage](https://www.tutorialrepublic.com/html-tutorial/html5-web-storage.php), etc.

**Tip:** Set autocomplete="off" for the input box if you want to prevent default browser menus from appearing over the Bootstrap type-ahead dropdown.

## Creating Twitter Typeahead External Dataset

You can also specify external dataset through a URL pointing to a JSON file containing an array of datums. The individual units that compose datasets are called datums.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=typeahead-with-external-dataset)

<script>

$(document).ready(function(){

// Sonstructs the suggestion engine

var countries =newBloodhound({

datumTokenizer: Bloodhound.tokenizers.whitespace,

queryTokenizer: Bloodhound.tokenizers.whitespace,

// The url points to a json file that contains an array of country names

prefetch:'data/countries.json'

});

// Initializing the typeahead with remote dataset without highlighting

$('.typeahead').typeahead(null,{

name:'countries',

source: countries,

limit:10/\* Specify max number of suggestions to be displayed \*/

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-carousel.php)

**Bootstrap ScrollSpy**

In this tutorial you will learn how to create scrollspy with Bootstrap.

**Creating ScrollSpy with Bootstrap**

The Bootstrap scrollspy is a navigation mechanism that automatically highlights the nav links based on the scroll position to indicate the visitor where they are currently on the page.

The scrollspy will make your web page more elegant and accessible, if you're using the bookmark links for directing the visitors to the different sections of a page that has a huge amount of content.

Scrollspy has the following requirements to function properly:

* It must be used on a Bootstrap [nav](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navs.php), or [list group](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-list-groups.php) component.
* You must apply the style position: relative; on the element you're spying on, which is usually the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element. But, if you're spying a [<div>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) or any element other than the <body> be sure to additionally apply a height and overflow-y: scroll; on it.
* Anchors ([<a>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php)) are required and must point to an element with that id.

Here's an example of a scrollspy using the list group. Let's try it out and see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=scrollspy)

<body data-bs-spy="scroll"data-bs-offset="15"data-bs-target="#myScrollspy">

<div class="container">

<div class="row">

<div class="col-sm-3"id="myScrollspy">

<div class="list-group">

<a class="list-group-item list-group-item-action active"href="#section1">Section One</a>

<a class="list-group-item list-group-item-action"href="#section2">Section Two</a>

<a class="list-group-item list-group-item-action"href="#section3">Section Three</a>

</div>

</div>

<div class="col-sm-9">

<div id="section1">

<h2>Section One</h2>

<p>This is section one content...</p>

</div>

<hr>

<div id="section2">

<h2>Section Two</h2>

<p>This is section two content...</p>

</div>

<hr>

<div id="section3">

<h2>Section Three</h2>

<p>This is section three content...</p>

</div>

</div>

</div>

</div>

</body>

**Creating ScrollSpy via Data Attributes**

You can easily add scrollspy behavior to your navbar via data attributes without writing a single line of JavaScript code. Let's try out the following example to see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-scrollspy-via-data-attributes)

<body data-bs-spy="scroll"data-bs-offset="60"data-bs-target="#myNavbar">

<nav id="myNavbar"class="navbar navbar-light bg-light fixed-top">

<div class="container-fluid">

<a href="#"class="navbar-brand">Navbar</a>

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#section1"class="nav-link">Section 1</a>

</li>

<li class="nav-item">

<a href="#section2"class="nav-link">Section 2</a>

</li>

<li class="nav-item">

<a href="#section3"class="nav-link">Section 3</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Section 4</a>

<div class="dropdown-menu">

<a href="#section4dot1"class="dropdown-item">Section 4.1</a>

<a href="#section4dot2"class="dropdown-item">Section 4.2</a>

<a href="#section4dot3"class="dropdown-item">Section 4.3</a>

</div>

</li>

<li>

<a href="#section5"class="nav-link">Section 5</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div id="section1">

<h2>Section 1</h2>

<p>This is section 1 content...</p>

</div>

<hr>

<div id="section2">

<h2>Section 2</h2>

<p>This is section 2 content...</p>

</div>

<hr>

<div id="section3">

<h2>Section 3</h2>

<p>This is section 3 content...</p>

</div>

<hr>

<h2>Section 4</h2>

<p>This is section 4 content</p>

<div id="section4dot1">

<h3>Section 4.1</h3>

<p>This is section 4.1 content...</p>

</div>

<div id="section4dot2">

<h3>Section 4.2</h3>

<p>This is section 4.2 content...</p>

</div>

<div id="section4dot3">

<h3>Section 4.3</h3>

<p>This is section 4.3 content...</p>

</div>

<hr>

<div id="section5">

<h2>Section 5</h2>

<p>This is section 5 content...</p>

</div>

</div>

</body>

You might be wondering what this code was all about. Well, let's go through each part of this scrollspy example code one by one for a better understanding.

**Explanation of Code**

The Bootstrap scrollspy has basically two components — the target (e.g. nav or list group) and the scrollable area to spy on, which is often the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) section.

* The data-bs-spy="scroll" attribute (*line no-01*) is applied to the element you want to spy on, which is simply the <body> element in our case.
* The data-bs-target attribute is added to the element we're spying on (i.e. the <body> element) with the ID or class of the parent element of any Bootstrap .nav component, so that nav links can be targeted by the scrollspy for highlighting purpose.
* The optional data-bs-offset attribute specifies the number of pixels to offset from top when calculating the position of scroll. Adjust the offset value if the targeted links are highlighting too early or too late. The default value is 10 pixels.

Rest of the thing is self explanatory, such as the .navbar element specifies a [Bootstrap navbar](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navbar.php), whereas the element <a href="#section1">Section 1</a> (*line no-7*) creates a bookmark link, the .dropdown element (*line no-15*) creates a dropdown menu, and so on.

**Creating ScrollSpy via JavaScript**

You may also add scrollspy manually using the JavaScript — just call the scrollspy() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the navbar in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-scrollspy-via-jquery)

<script>

$(document).ready(function(){

$("body").scrollspy({

target:"#myNavbar"

})

});

</script>

**Options**

There are certain options which may be passed to scrollspy() Bootstrap method to customize the functionality of a scrollspy. Options can be passed via data attributes or JavaScript.

For setting the scrollspy options via data attributes, just append the option name to data-bs-, like data-bs-offset="0", data-bs-method="position", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| offset | number | 10 | Number of pixels to offset from top when calculating position of scroll. |
| method | string | auto | Finds which section the spied element is in. The value auto will choose the best method get scroll coordinates.  Whereas, the value offset will use jQuery offset method to get scroll coordinates, and the value position will use jQuery position method to get scroll coordinates. |
| target | string | jQuery object | DOM element |  | Specifies element to apply Scrollspy plugin. |

**Methods**

These are the standard bootstrap's scrollspy methods:

**Passing options**

You can additionally pass options to the scrollspy using options object.

The following example will set offset from top when calculating scroll position.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=set-scrollspy-offset-via-jquery)

<script>

$(document).ready(function(){

$("body").scrollspy({

offset:70

})

});

</script>

**refresh**

You'll need to call this method when you're using scrollspy in conjunction with adding or removing elements from the DOM. Let's try out an example and see how it really works:

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=refresh-scrollspy-using-jquery)

<script>

$(document).ready(function(){

$('[data-spy="scroll"]').each(function(){

$(this).scrollspy("refresh");

});

});

</script>

**dispose**

This method destroys an element's scrollspy (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-scrollspy-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myScrollspy = bootstrap.ScrollSpy.getInstance($("#myContent")[0]);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

$("#myContent").scrollspy("dispose");

console.log(myScrollspy);

// {\_element: null, \_scrollElement: null, \_config: null, \_selector: null, \_offsets: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the scrollspy instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-scrollspy-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myScrollspy = bootstrap.ScrollSpy.getInstance($("#myContent")[0]);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the scrollspy instance associated with a DOM element, or create a new one in case if the scrollspy wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-scrollspy-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myScrollspy = bootstrap.ScrollSpy.getOrCreateInstance($("#myContent")[0]);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

});

});

</script>

**Events**

Bootstrap's scrollspy class includes few events for hooking into scrollspy functionality.

| **Event** | **Description** |
| --- | --- |
| activate.bs.scrollspy | This event fires whenever a new item becomes activated by the scrollspy. |

The following example displays an alert message when a new item becomes highlighted by the scrollspy. Let's try it out and see how it actually works.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-scrollspy-event-using-jquery)

<script>

$(document).ready(function(){

$("#myNavbar").on("activate.bs.scrollspy",function(){

var currentItem =$(".nav li.active > a").text();

$("#info").empty().html("Currently you are viewing - "+ currentItem);

})

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-typeahead.php)

**Bootstrap ScrollSpy**

In this tutorial you will learn how to create scrollspy with Bootstrap.

**Creating ScrollSpy with Bootstrap**

The Bootstrap scrollspy is a navigation mechanism that automatically highlights the nav links based on the scroll position to indicate the visitor where they are currently on the page.

The scrollspy will make your web page more elegant and accessible, if you're using the bookmark links for directing the visitors to the different sections of a page that has a huge amount of content.

Scrollspy has the following requirements to function properly:

* It must be used on a Bootstrap [nav](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navs.php), or [list group](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-list-groups.php) component.
* You must apply the style position: relative; on the element you're spying on, which is usually the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element. But, if you're spying a [<div>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) or any element other than the <body> be sure to additionally apply a height and overflow-y: scroll; on it.
* Anchors ([<a>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php)) are required and must point to an element with that id.

Here's an example of a scrollspy using the list group. Let's try it out and see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=scrollspy)

<body data-bs-spy="scroll"data-bs-offset="15"data-bs-target="#myScrollspy">

<div class="container">

<div class="row">

<div class="col-sm-3"id="myScrollspy">

<div class="list-group">

<a class="list-group-item list-group-item-action active"href="#section1">Section One</a>

<a class="list-group-item list-group-item-action"href="#section2">Section Two</a>

<a class="list-group-item list-group-item-action"href="#section3">Section Three</a>

</div>

</div>

<div class="col-sm-9">

<div id="section1">

<h2>Section One</h2>

<p>This is section one content...</p>

</div>

<hr>

<div id="section2">

<h2>Section Two</h2>

<p>This is section two content...</p>

</div>

<hr>

<div id="section3">

<h2>Section Three</h2>

<p>This is section three content...</p>

</div>

</div>

</div>

</div>

</body>

**Creating ScrollSpy via Data Attributes**

You can easily add scrollspy behavior to your navbar via data attributes without writing a single line of JavaScript code. Let's try out the following example to see how it works:

**Example**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-scrollspy-via-data-attributes)

<body data-bs-spy="scroll"data-bs-offset="60"data-bs-target="#myNavbar">

<nav id="myNavbar"class="navbar navbar-light bg-light fixed-top">

<div class="container-fluid">

<a href="#"class="navbar-brand">Navbar</a>

<ul class="nav nav-pills">

<li class="nav-item">

<a href="#section1"class="nav-link">Section 1</a>

</li>

<li class="nav-item">

<a href="#section2"class="nav-link">Section 2</a>

</li>

<li class="nav-item">

<a href="#section3"class="nav-link">Section 3</a>

</li>

<li class="nav-item dropdown">

<a href="#"class="nav-link dropdown-toggle"data-bs-toggle="dropdown">Section 4</a>

<div class="dropdown-menu">

<a href="#section4dot1"class="dropdown-item">Section 4.1</a>

<a href="#section4dot2"class="dropdown-item">Section 4.2</a>

<a href="#section4dot3"class="dropdown-item">Section 4.3</a>

</div>

</li>

<li>

<a href="#section5"class="nav-link">Section 5</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div id="section1">

<h2>Section 1</h2>

<p>This is section 1 content...</p>

</div>

<hr>

<div id="section2">

<h2>Section 2</h2>

<p>This is section 2 content...</p>

</div>

<hr>

<div id="section3">

<h2>Section 3</h2>

<p>This is section 3 content...</p>

</div>

<hr>

<h2>Section 4</h2>

<p>This is section 4 content</p>

<div id="section4dot1">

<h3>Section 4.1</h3>

<p>This is section 4.1 content...</p>

</div>

<div id="section4dot2">

<h3>Section 4.2</h3>

<p>This is section 4.2 content...</p>

</div>

<div id="section4dot3">

<h3>Section 4.3</h3>

<p>This is section 4.3 content...</p>

</div>

<hr>

<div id="section5">

<h2>Section 5</h2>

<p>This is section 5 content...</p>

</div>

</div>

</body>

You might be wondering what this code was all about. Well, let's go through each part of this scrollspy example code one by one for a better understanding.

**Explanation of Code**

The Bootstrap scrollspy has basically two components — the target (e.g. nav or list group) and the scrollable area to spy on, which is often the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) section.

* The data-bs-spy="scroll" attribute (*line no-01*) is applied to the element you want to spy on, which is simply the <body> element in our case.
* The data-bs-target attribute is added to the element we're spying on (i.e. the <body> element) with the ID or class of the parent element of any Bootstrap .nav component, so that nav links can be targeted by the scrollspy for highlighting purpose.
* The optional data-bs-offset attribute specifies the number of pixels to offset from top when calculating the position of scroll. Adjust the offset value if the targeted links are highlighting too early or too late. The default value is 10 pixels.

Rest of the thing is self explanatory, such as the .navbar element specifies a [Bootstrap navbar](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-navbar.php), whereas the element <a href="#section1">Section 1</a> (*line no-7*) creates a bookmark link, the .dropdown element (*line no-15*) creates a dropdown menu, and so on.

**Creating ScrollSpy via JavaScript**

You may also add scrollspy manually using the JavaScript — just call the scrollspy() Bootstrap method with the id or class [selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the navbar in your JavaScript code.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-scrollspy-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myScrollSpy =newbootstrap.ScrollSpy(document.body,{

target:"#myNavbar"

})

});

</script>

**Options**

There are certain options which may be passed to scrollspy() Bootstrap method to customize the functionality of a scrollspy. Options can be passed via data attributes or JavaScript.

For setting the scrollspy options via data attributes, just append the option name to data-bs-, like data-bs-offset="0", data-bs-method="position", and so on.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| offset | number | 10 | Number of pixels to offset from top when calculating position of scroll. |
| method | string | auto | Finds which section the spied element is in. The value auto will choose the best method get scroll coordinates.  Whereas, the value offset will use jQuery offset method to get scroll coordinates, and the value position will use jQuery position method to get scroll coordinates. |
| target | string | jQuery object | DOM element |  | Specifies element to apply Scrollspy plugin. |

**Methods**

These are the standard bootstrap's scrollspy methods:

**Passing options**

You can additionally pass options to the scrollspy using options object.

The following example will set offset from top when calculating scroll position.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=adding-scrollspy-offset-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myScrollSpy =newbootstrap.ScrollSpy(document.body,{

offset:70

})

});

</script>

**refresh**

You'll need to call this method when you're using scrollspy in conjunction with adding or removing elements from the DOM. Let's try out an example and see how it really works:

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=refresh-scrollspy-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var scrollspyList =[].slice.call(document.querySelectorAll('[data-bs-spy="scroll"]'));

scrollspyList.forEach(function(element){

bootstrap.ScrollSpy.getInstance(element).refresh();

})

});

</script>

**dispose**

This method destroys an element's scrollspy (i.e. removes stored data on the DOM element).

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=dispose-scrollspy-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myContent");

btn.addEventListener("click",function(){

var myScrollspy = bootstrap.ScrollSpy.getInstance(element);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

myScrollspy.dispose();

console.log(myScrollspy);

// {\_element: null, \_scrollElement: null, \_config: null, \_selector: null, \_offsets: null, …}

});

});

</script>

**getInstance**

This is a static method which allows you to get the scrollspy instance associated with a DOM element.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-scrollspy-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myContent");

btn.addEventListener("click",function(){

var myScrollspy = bootstrap.ScrollSpy.getInstance(element);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

});

});

</script>

**getOrCreateInstance**

This is a static method which allows you to get the scrollspy instance associated with a DOM element, or create a new one in case if the scrollspy wasn't initialized.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-scrollspy-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myContent");

btn.addEventListener("click",function(){

var myScrollspy = bootstrap.ScrollSpy.getOrCreateInstance(element);

console.log(myScrollspy);

// {\_element: div#myContent, \_scrollElement: div#myContent, \_config: {…}, \_selector: "#myNavbar .nav-link, #myNavbar .list-group-item, #myNavbar .dropdown-item", \_offsets: Array(5), …}

});

});

</script>

**Events**

Bootstrap's scrollspy class includes few events for hooking into scrollspy functionality.

| **Event** | **Description** |
| --- | --- |
| activate.bs.scrollspy | This event fires whenever a new item becomes activated by the scrollspy. |

The following example displays an alert message when a new item becomes highlighted by the scrollspy. Let's try it out and see how it actually works.

**Example**

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-scrollspy-event-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var scrollspyList =[].slice.call(document.querySelectorAll('[data-bs-spy="scroll"]'));

scrollspyList.forEach(function(element){

bootstrap.ScrollSpy.getInstance(element).refresh();

})

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-typeahead.php)

# Bootstrap Toasts

In this tutorial you will learn how to use the Bootstrap toast component.

## Creating the Toasts with Bootstrap

The toast component is newly introduced in Bootstrap 4. They are lightweight notifications similar to push notifications that are shown by web browsers on computer screens. They're built with flexbox, so you can easily align and position them on a web page.

Additionally, toasts are opt-in for performance reasons, so you must initialize them yourself with toast() method. Also, toasts will automatically hide after 5000 milliseconds (5 seconds), if you do not specify autohide: false option. Now let's see how to create a toast.

### Step 1: Adding the Toast Markup

Toasts markups are pretty straightforward. The following example will show you how to create a toast component with a header, body, and a close button.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast)

<div class="toast"id="myToast">

<div class="toast-header">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#">Click here!</a>

</div>

</div>

### Step 2: Triggering the Toasts

Toasts can be triggered via JavaScript — just call the toast() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the target element in your JavaScript code.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=trigger-toast-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

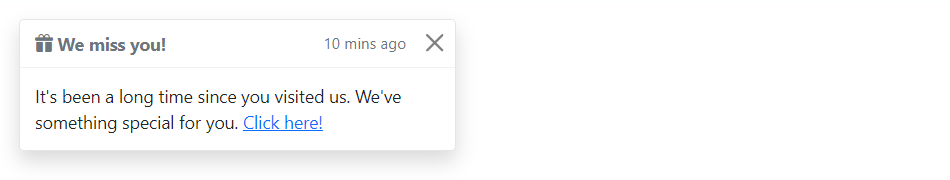
$("#myToast").toast("show");

});

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast)

## Changing the Toast's Color Schemes

You can use the [color](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes) and [background](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) utility classes to create toasts with different color schemes.

The following example will create a toast with blue background and white text.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast-with-different-color-schemes)

<div class="toast bg-primary text-white fade show">

<div class="toast-header bg-primary text-white">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close btn-close-white"data-bs-dismiss="toast"></button>

</div>

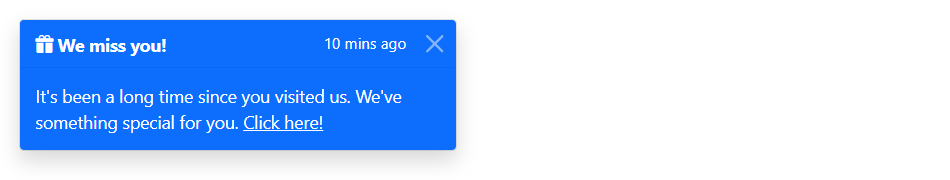
<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#"class="text-white">Click here!</a>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast-with-different-color-schemes)

## Stacking Toasts Vertically

You can stack multiple toasts vertically by simply wrapping them in a toast container, which will vertically add some spacing. Let's take a look at the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-toasts)

<div class="toast-container">

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>just now</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

This is a basic toast message.

</div>

</div>

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>5 seconds ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

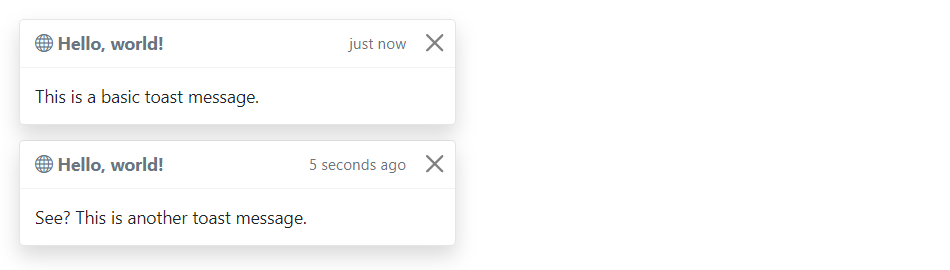
See? This is another toast message.

</div>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-toasts)

## Placement of Toasts

You can place toasts anywhere on your web page using custom [CSS positioning](https://www.tutorialrepublic.com/css-tutorial/css-position.php). However, the top right, bottom right, or top middle side is recommended for notifications.

Also, if you only want to show one toast at a time, put the positioning styles inline (i.e. directly on the .toast element). Let's try out an example and see how it works:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toasts-placement)

<div class="toast-container" style="position: absolute;top: 10px;right: 10px;">

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>just now</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

This is a basic toast message.

</div>

</div>

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>5 seconds ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

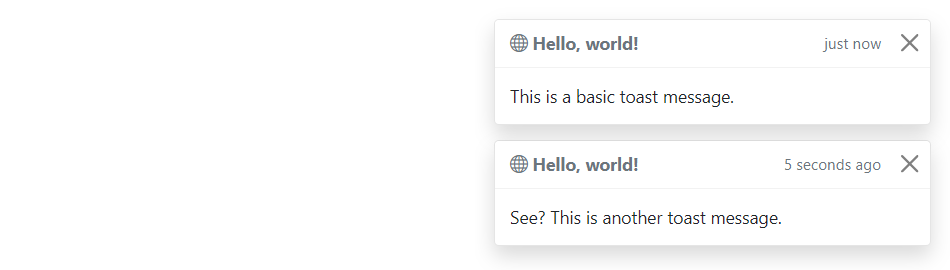
See? This is another toast message.

</div>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toasts-placement)

## Options

There are certain options which can be passed to toast() Bootstrap method to customize the functionality of a toast. Options can be passed via data attributes or JavaScript.

For setting the toast options via data attributes, just append the option name to data-bs-, such as data-bs-autohide="false", data-bs-delay="3000", etc.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the toast. |
| autohide | boolean | true | Auto hide the toast. |
| delay | number | 5000 | Delay hiding the toast (ms). |

Data attributes provides an easy way for setting the toast options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-toasts.php#passing-options) examples in the methods section below to know how to set the options for toasts using JavaScript.

In the following example we've set the autohide option to false using the data attribute (*line no-1*) which prevents the toast from closing automatically.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-hiding-toast-automatically-via-data-attributes)

<div class="toast"id="myToast"data-bs-autohide="false">

<div class="toast-header">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#">Click here!</a>

</div>

</div>

## Methods

These are the standard bootstrap's toast methods:

## Passing options

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-toasts.php#toast-options) to the toast using options object.

The following example code will prevent the toast from closing automatically.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disable-toast-autohide-via-jquery)

<script>

$(document).ready(function(){

$("#myToast").toast({

autohide:false

});

});

</script>

The following example code will increase the autohide time of toast to 10 seconds.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=increase-toast-autohide-time-via-jquery)

<script>

$(document).ready(function(){

$("#myToast").toast({

delay:10000

});

});

</script>

## show

This method is used to display the toast.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-jquery)

<script>

$(document).ready(function(){

$(#myBtn").click(function(){

$("#myToast").toast("show");

});

});

</script>

## hide

This method is used to hide toast. You've to manually call this method if you set autohide to false.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-jquery)

<script>

$(document).ready(function(){

$(#myBtn").click(function(){

$("#myToast").toast("hide");

});

});

</script>

## dispose

This method hides an element's toast. The toast will remain on the DOM but won't show anymore.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-jquery)

<script>

$(document).ready(function(){

$(#myBtn").click(function(){

$("#myToast").toast("dispose");

});

});

</script>

## getInstance

This is a static method which allows you to get the toast instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-toast-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myToast = bootstrap.Toast.getInstance($("#myToast")[0]);

console.log(myToast);

// {\_element: div#myToast.toast.fade.show, \_config: {…}, \_timeout: null, \_hasMouseInteraction: false, \_hasKeyboardInteraction: false}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the toast instance associated with a DOM element, or create a new one in case if the toast wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-toast-instance-using-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

var myToast = bootstrap.Toast.getOrCreateInstance($("#myToast")[0]);

console.log(myToast);

// {\_element: div#myToast.toast.fade.show, \_config: {…}, \_timeout: null, \_hasMouseInteraction: false, \_hasKeyboardInteraction: false}

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

## Events

Bootstrap's modal class includes few events for hooking into modal functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.toast | This event fires immediately when the show instance method is called. |
| shown.bs.toast | This event is fired when the toast has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.toast | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.toast | This event is fired when the toast has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |

The following example will display an alert message to the user when the fade out transition of the toast has been fully completed. Let's try it out and see how it works:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-toast-events-using-jquery)

<script>

$(document).ready(function(){

$("#myToast").on("hidden.bs.toast",function(){

alert("Toast component has been completely closed.");

});

});

</script>

[PREVIOUS PAGE](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-scrollspy.php)

# Bootstrap Toasts

In this tutorial you will learn how to use the Bootstrap toast component.

## Creating the Toasts with Bootstrap

The toast component is newly introduced in Bootstrap 4. They are lightweight notifications similar to push notifications that are shown by web browsers on computer screens. They're built with flexbox, so you can easily align and position them on a web page.

Additionally, toasts are opt-in for performance reasons, so you must initialize them yourself with toast() method. Also, toasts will automatically hide after 5000 milliseconds (5 seconds), if you do not specify autohide: false option. Now let's see how to create a toast.

### Step 1: Adding the Toast Markup

Toasts markups are pretty straightforward. The following example will show you how to create a toast component with a header, body, and a close button.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast)

<div class="toast"id="myToast">

<div class="toast-header">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#">Click here!</a>

</div>

</div>

### Step 2: Triggering the Toasts

Toasts can be triggered via JavaScript — just call the toast() Bootstrap method with the id, class or any [CSS selector](https://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the target element in your JavaScript code.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=trigger-toast-via-jquery)

<script>

$(document).ready(function(){

$("#myBtn").click(function(){

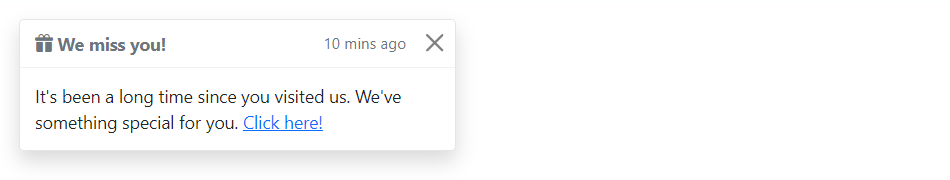
$("#myToast").toast("show");

});

});

</script>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast)

## Changing the Toast's Color Schemes

You can use the [color](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#color-classes) and [background](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-helper-classes.php#background-color-classes) utility classes to create toasts with different color schemes.

The following example will create a toast with blue background and white text.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast-with-different-color-schemes)

<div class="toast bg-primary text-white fade show">

<div class="toast-header bg-primary text-white">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close btn-close-white"data-bs-dismiss="toast"></button>

</div>

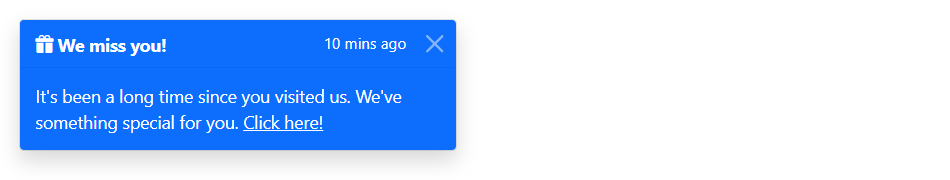
<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#"class="text-white">Click here!</a>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toast-with-different-color-schemes)

## Stacking Toasts Vertically

You can stack multiple toasts vertically by simply wrapping them in a toast container, which will vertically add some spacing. Let's take a look at the following example:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-toasts)

<div class="toast-container">

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>just now</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

This is a basic toast message.

</div>

</div>

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>5 seconds ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

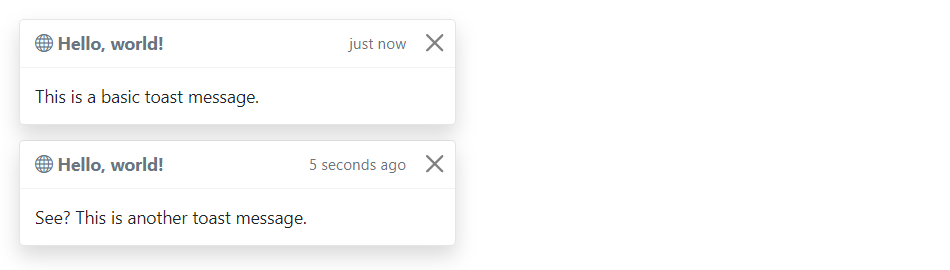
See? This is another toast message.

</div>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=vertically-stacked-toasts)

## Placement of Toasts

You can place toasts anywhere on your web page using custom [CSS positioning](https://www.tutorialrepublic.com/css-tutorial/css-position.php). However, the top right, bottom right, or top middle side is recommended for notifications.

Also, if you only want to show one toast at a time, put the positioning styles inline (i.e. directly on the .toast element). Let's try out an example and see how it works:

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toasts-placement)

<div class="toast-container" style="position: absolute;top: 10px;right: 10px;">

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>just now</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

This is a basic toast message.

</div>

</div>

<div class="toast fade show">

<div class="toast-header">

<strong class="me-auto"><i class="bi-globe"></i> Hello, world!</strong>

<small>5 seconds ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

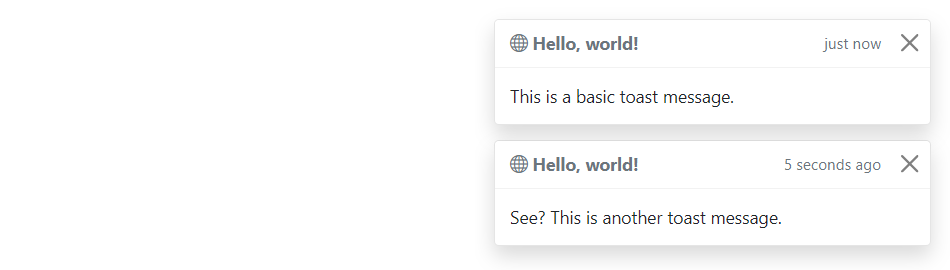
See? This is another toast message.

</div>

</div>

</div>

— The output of the above example will look something like this:

[](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=toasts-placement)

## Options

There are certain options which can be passed to toast() Bootstrap method to customize the functionality of a toast. Options can be passed via data attributes or JavaScript.

For setting the toast options via data attributes, just append the option name to data-bs-, such as data-bs-autohide="false", data-bs-delay="3000", etc.

| **Name** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| animation | boolean | true | Apply a CSS fade transition to the toast. |
| autohide | boolean | true | Auto hide the toast. |
| delay | number | 5000 | Delay hiding the toast (ms). |

Data attributes provides an easy way for setting the toast options, however JavaScript is the more preferable way as it prevents you from repetitive work. See the [passing options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-toasts.php#passing-options) examples in the methods section below to know how to set the options for toasts using JavaScript.

In the following example we've set the autohide option to false using the data attribute (*line no-1*) which prevents the toast from closing automatically.

#### Example

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=stop-hiding-toast-automatically-via-data-attributes)

<div class="toast"id="myToast"data-bs-autohide="false">

<div class="toast-header">

<strong class="me-auto"><i class="bi-gift-fill"></i> We miss you!</strong>

<small>10 mins ago</small>

<button type="button"class="btn-close"data-bs-dismiss="toast"></button>

</div>

<div class="toast-body">

It's been a long time since you visited us. We've something special for you. <a href="#">Click here!</a>

</div>

</div>

## Methods

These are the standard bootstrap's toast methods:

## Passing options

You can additionally pass [options](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-toasts.php#toast-options) to the toast using options object.

The following example code will prevent the toast from closing automatically.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=disable-toast-autohide-via-jquery)

<script>

$(document).ready(function(){

$("#myToast").toast({

autohide:false

});

});

</script>

The following example code will increase the autohide time of toast to 10 seconds.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=increase-toast-autohide-time-via-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var element = document.getElementById("myToast");

var myToast =newbootstrap.Toast(element,{

delay:10000

});

});

</script>

## show

This method is used to display the toast.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myToast");

// Create toast instance

var myToast =newbootstrap.Toast(element);

btn.addEventListener("click",function(){

myToast.show();

});

});

</script>

## hide

This method is used to hide toast. You've to manually call this method if you set autohide to false.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myToast");

// Create toast instance

var myToast =newbootstrap.Toast(element);

btn.addEventListener("click",function(){

myToast.hide();

});

});

</script>

## dispose

This method hides an element's toast. The toast will remain on the DOM but won't show anymore.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=calling-toast-methods-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myToast");

// Create toast instance

var myToast =newbootstrap.Toast(element);

btn.addEventListener("click",function(){

myToast.dispose();

});

});

</script>

## getInstance

This is a static method which allows you to get the toast instance associated with a DOM element.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-toast-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myToast");

btn.addEventListener("click",function(){

var myToast = bootstrap.Toast.getInstance(element);

console.log(myToast);

// {\_element: div#myToast.toast.fade.show, \_config: {…}, \_timeout: null, \_hasMouseInteraction: false, \_hasKeyboardInteraction: false}

});

});

</script>

## getOrCreateInstance

This is a static method which allows you to get the toast instance associated with a DOM element, or create a new one in case if the toast wasn't initialized.

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=get-or-create-toast-instance-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var btn = document.getElementById("myBtn");

var element = document.getElementById("myToast");

btn.addEventListener("click",function(){

var myToast = bootstrap.Toast.getOrCreateInstance(element);

console.log(myToast);

// {\_element: div#myToast.toast.fade.show, \_config: {…}, \_timeout: null, \_hasMouseInteraction: false, \_hasKeyboardInteraction: false}

});

});

</script>

**Tip:** Static methods cannot be called on instances of the class (i.e. objects). They're called on the class itself. The keyword static is used to define a static method for a class.

## Events

Bootstrap's modal class includes few events for hooking into modal functionality.

| **Event** | **Description** |
| --- | --- |
| show.bs.toast | This event fires immediately when the show instance method is called. |
| shown.bs.toast | This event is fired when the toast has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired. |
| hide.bs.toast | This event is fired immediately when the hide instance method has been called. |
| hidden.bs.toast | This event is fired when the toast has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired. |

The following example will display an alert message to the user when the fade out transition of the toast has been fully completed. Let's try it out and see how it works:

#### Example

**jQuery** **JavaScript**

[**Try this code »**](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=listening-to-toast-events-using-javascript)

<script>

document.addEventListener("DOMContentLoaded",function(){

var myToast = document.getElementById("myToast");

myToast.addEventListener("hidden.bs.toast",function(){

alert("Toast component has been completely closed.");

});

});

</script>

https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/