

Bootstrap 4 Tutorial

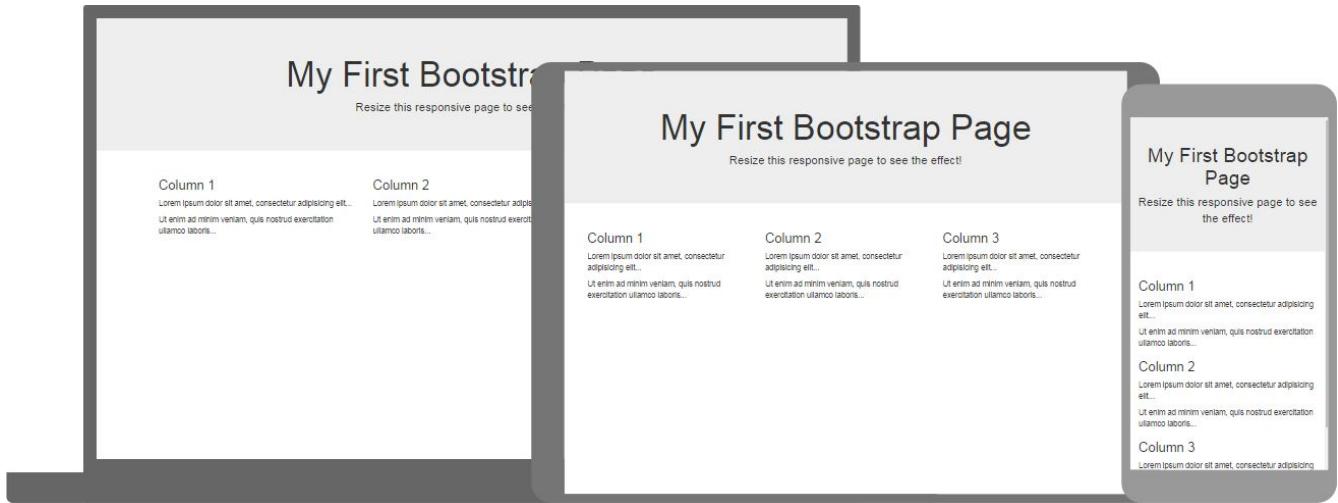
Bootstrap 4 is the newest version of [Bootstrap](#), which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Bootstrap 4 is completely free to download and use!

[Learn Bootstrap 4](#)

Try it Yourself Examples

This Bootstrap 4 tutorial contains hundreds of Bootstrap 4 examples.

With our online editor, you can edit the code, and click on a button to view the result.



Bootstrap 4 Example

```
<div class="jumbotron text-center">
  <h1>My First Bootstrap Page</h1>
  <p>Resize this responsive page to see the effect!</p>
</div>

<div class="container">
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum dolor..</p>
      <p>Ut enim ad..</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum dolor..</p>
      <p>Ut enim ad..</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 3</h3>
      <p>Lorem ipsum dolor..</p>
      <p>Ut enim ad..</p>
    </div>
  </div>
</div>
```

Click on the "Try it Yourself" button to see how it works.

Bootstrap 3 vs. Bootstrap 4

Bootstrap 4 is the newest version of Bootstrap; with new components, faster stylesheet and more responsiveness.

Bootstrap 4 supports the latest, stable releases of all major browsers and platforms. However, Internet Explorer 9 and down is not supported.

If you require IE8-9 support, use Bootstrap 3. It is the most stable version of Bootstrap, and it is still supported by the team for critical bugfixes and documentation changes. However, new features will NOT be added to it.

Bootstrap 4 Basic Template

We have created a responsive starter template with Bootstrap 4. You are free to modify, save, share, and use it in your projects:

[Demo](#)[Try it Yourself](#)

Bootstrap References

A complete list of all Bootstrap 4 CSS classes with description and examples:

- [Bootstrap All CSS Classes](#) Popular

If you want to do more with Bootstrap components, see our complete Bootstrap 4 JavaScript/jQuery reference - all with "Try it Yourself" examples:

- [Bootstrap JS Alert](#)
- [Bootstrap JS Button](#)
- [Bootstrap JS Carousel](#)
- [Bootstrap JS Collapse](#)
- [Bootstrap JS Dropdown](#)
- [Bootstrap JS Modal](#)
- [Bootstrap JS Popover](#)
- [Bootstrap JS Scrollspy](#)
- [Bootstrap JS Tab](#)
- [Bootstrap JS Tooltip](#)

Did You Know?

W3.CSS is an excellent alternative to Bootstrap 4.

W3.CSS is smaller, faster, and easier to use.

If you want to learn W3.CSS, go to our [W3.CSS Tutorial](#).

Bootstrap 4 Grids

Bootstrap 4 Grid System

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

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

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1				
span 4				span 4				span 4							
span 4						span 8									
span 6						span 6									
span 12															

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

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

Grid Classes

The Bootstrap 4 grid system has five classes:

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

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

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

Basic Structure of a Bootstrap 4 Grid

The following is a basic structure of a Bootstrap 4 grid:

```
<!-- Control the column width, and how they should appear on different devices -->
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>

<!-- Or let Bootstrap automatically handle the layout -->
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

First example: create a row (`<div class="row">`). Then, add the desired number of columns (tags with appropriate `.col-*-*` classes). The first star (*) represents the responsiveness: sm, md, lg or xl, while the second star represents a number, which should add up to 12 for each row.

Second example: instead of adding a number to each `col`, let bootstrap handle the layout, to create equal width columns: two `"col"` elements = 50% width to each col. three cols = 33.33% width to each col. four cols = 25% width, etc. You can also use `.col-sm|md|lg|xl` to make the columns responsive.

Below we have collected some examples of basic Bootstrap 4 grid layouts.

Three Equal Columns

.col

.col

.col

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

Example

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

Responsive Columns

.col-sm-3

.col-sm-3

.col-sm-3

.col-sm-3

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

Example

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

Two Unequal Responsive Columns

.col-sm-4

.col-sm-8

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

Example

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

Tip: You will learn more about Bootstrap 4 grids later in this tutorial.

Bootstrap 4 Text/Typography

Bootstrap 4 Default Settings

Bootstrap 4 uses a default `font-size` of 16px, and its `line-height` is 1.5.

The default `font-family` is "Helvetica Neue", Helvetica, Arial, sans-serif.

In addition, all `<p>` elements have `margin-top: 0` and `margin-bottom: 1rem` (16px by default).

<h1> - <h6>

Bootstrap 4 styles HTML headings (`<h1>` to `<h6>`) with a bolder font-weight and an increased font-size:

Example

h1 Bootstrap heading (2.5rem = 40px)

h2 Bootstrap heading (2rem = 32px)

h3 Bootstrap heading (1.75rem = 28px)

h4 Bootstrap heading (1.5rem = 24px)

h5 Bootstrap heading (1.25rem = 20px)

h6 Bootstrap heading (1rem = 16px)

Display Headings

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are four classes to choose from: `.display-1`, `.display-2`, `.display-3`, `.display-4`

Example

Display 1

Display 2

Display 3

Display 4

<small>

In Bootstrap 4 the HTML `<small>` element is used to create a lighter, secondary text in any heading:

Example

h1 heading secondary text

h2 heading secondary text

h3 heading secondary text

h4 heading secondary text

h5 heading secondary text

h6 heading secondary text

<mark>

Bootstrap 4 will style the HTML `<mark>` element with a yellow background color and some padding:

Example

Use the mark element to highlight text.

<abbr>

Bootstrap 4 will style the HTML `<abbr>` element with a dotted border bottom:

Example

The WHO (World Health Organization) was founded in 1948.

<blockquote>

Add the `.blockquote` class to a `<blockquote>` when quoting blocks of content from another source:

Example

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

— From WWF's website

<dl>

Bootstrap 4 will style the HTML `<dl>` element in the following way:

Example

Coffee

- black hot drink

Milk

- white cold drink

<code>

Bootstrap 4 will style the HTML `<code>` element in the following way:

Example

The following HTML elements: `span`, `section`, and `div` defines a section in a document.

<kbd>

Bootstrap 4 will style the HTML `<kbd>` element in the following way:

Example

Use `ctrl + p` to open the Print dialog box.

<pre>

Bootstrap 4 will style the HTML `<pre>` element in the following way:

Example

```
Text in a pre element  
is displayed in a fixed-width  
font, and it preserves  
both      spaces and  
line breaks.
```

More Typography Classes

The Bootstrap 4 classes below can be added to style HTML elements further:

Class	Description	Example
<code>.font-weight-bold</code>	Bold text	Try it
<code>.font-italic</code>	Italic text	Try it
<code>.font-weight-light</code>	Light weight text	Try it
<code>.font-weight-normal</code>	Normal text	Try it
<code>.lead</code>	Makes a paragraph stand out	Try it
<code>.small</code>	Indicates smaller text (set to 85% of the size of the parent)	Try it

.text-left	Indicates left-aligned text	Try it
.text-* -left	Indicates left-aligned text on small, medium, large or xlarge screens	Try it
.text-center	Indicates center-aligned text	Try it
.text-* -center	Indicates center-aligned text on small, medium, large or xlarge screens	Try it
.text-right	Indicates right-aligned text	Try it
.text-* -right	Indicates right-aligned text on small, medium, large or xlarge screens	Try it
.text-justify	Indicates justified text	Try it
.text-monospace	Monospaced text	Try it
.text nowrap	Indicates no wrap text	Try it
.text-lowercase	Indicates lowercased text	Try it
.text-uppercase	Indicates upercased text	Try it
.text-capitalize	Indicates capitalized text	Try it
.initialism	Displays the text inside an <code><abbr></code> element in a slightly smaller font size	Try it
.list-unstyled	Removes the default list-style and left margin on list items (works on both <code></code> and <code></code>). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)	Try it
.list-inline	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)	Try it
.pre-scrollable	Makes a <code><pre></code> element scrollable	Try it

Complete Bootstrap 4 CSS Reference

For a complete reference of all CSS classes available in Bootstrap 4, visit our [Bootstrap 4 All Classes Reference](#).

Bootstrap 4 Colors

Text Colors

Bootstrap 4 has some contextual classes that can be used to provide "meaning through colors".

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

Example

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

Dark grey text.

Body text.

Light grey text.

Contextual text classes can also be used on links, which will add a darker hover color:

Example

Muted link. Primary link. Success link. Info link. Warning link. Danger link. Secondary link. Dark grey link. Body/black link. Light grey link.

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

Example

Black text with 50% opacity on white background

White text with 50% opacity on black background

Background Colors

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

Note that background colors do not set the text color, so in some cases you'll want to use them together with a `.text-*` class.

Example

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

Bootstrap 4 Tables

Bootstrap 4 Basic Table

A basic Bootstrap 4 table has a light padding and horizontal dividers.

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Striped Rows

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Bordered Table

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Hover Rows

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Black/Dark Table

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

Example

Dark Striped Table

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

Example

Hoverable Dark Table

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

Example

Borderless Table

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Contextual Classes

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

Example

Firstname	Lastname	Email
Default	Defaultson	def@somemail.com
Primary	Joe	joe@example.com
Success	Doe	john@example.com
Danger	Moe	mary@example.com
Info	Dooley	july@example.com
Warning	Refs	bo@example.com
Active	Activeson	act@example.com
Secondary	Secondson	sec@example.com
Light	Angie	angie@example.com
Dark	Bo	bo@example.com

The contextual classes that can be used are:

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

Table Head Colors

The `.thead-dark` class adds a black background to table headers, and the `.thead-light` class adds a grey background to table headers:

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Small table

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

Example

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Responsive Tables

The `.table-responsive` class creates a responsive table: an horizontal scrollbar is added to the table on screens that are less than 992px wide (if needed). When viewing on anything larger than 992px wide, there is no difference:

Example

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

#	Firstname	Lastname	Age	City	Country	Sex	Example	Ex
1	Anna	Pitt	35	New York	USA	Female	Yes	Ye

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

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

Example

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

Bootstrap 4 Images

Bootstrap 4 Image Shapes

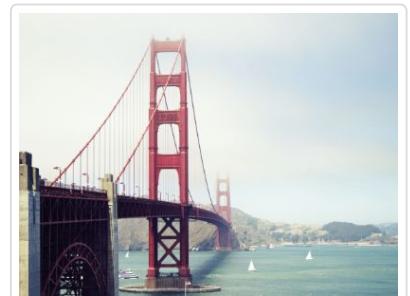
Rounded Corners:



Circle:



Thumbnail:



Rounded Corners

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

Example

```

```

Circle

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

Example

```

```

Thumbnail

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

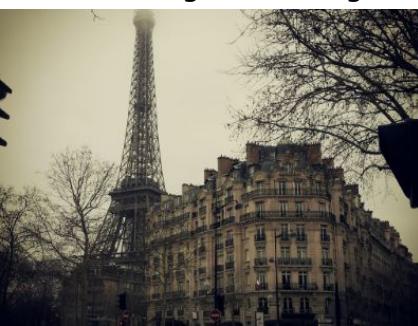
Example

```

```

Aligning Images

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



Example

```


```

Centered Image

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



Example

```

```

Responsive Images

Images come in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.

Create responsive images by adding an `.img-fluid` class to the `` tag. The image will then scale nicely to the parent element.

The `.img-fluid` class applies `max-width: 100%;` and `height: auto;` to the image:

Example

```

```

Bootstrap 4 Jumbotron

Bootstrap 4 Jumbotron

A jumbotron indicates a big grey box for calling extra attention to some special content or information.

Tip: Inside a jumbotron you can put nearly any valid HTML, including other Bootstrap elements/classes.

Bootstrap Tutorial

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web.

Use a `<div>` element with class `.jumbotron` to create a jumbotron:

Example

```
<div class="jumbotron">
  <h1>Bootstrap Tutorial</h1>
  <p>Bootstrap is the most popular HTML, CSS...</p>
</div>
```

Full-width Jumbotron

If you want a full-width jumbotron without rounded borders, add the `.jumbotron-fluid` class and a `.container` or `.container-fluid` inside of it:

Example

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1>Bootstrap Tutorial</h1>
    <p>Bootstrap is the most popular HTML, CSS...</p>
  </div>
</div>
```

Bootstrap 4 Alerts

Bootstrap 4 Alerts

Bootstrap 4 provides an easy way to create predefined alert messages:

Success! This alert box indicates a successful or positive action. X

Info! This alert box indicates a neutral informative change or action. X

Warning! This alert box indicates a warning that might need attention. X

Danger! This alert box indicates a dangerous or potentially negative action. X

Primary! This alert box indicates an important action. X

Secondary! This alert box indicates a less important action. X

Dark! Dark grey alert box. X

Light! Light grey alert box. X

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

Example

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

Alert Links

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

Success! You should [read this message](#).

Info! You should [read this message](#).

Warning! You should [read this message](#).

Danger! You should [read this message](#).

Primary! You should [read this message](#).

Secondary! You should [read this message](#).

Dark! You should [read this message](#).

Light! You should [read this message](#).

Example

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

Closing Alerts

Click on the "x" symbol to the right to close me.



To close the alert message, add a `.alert-dismissible` class to the alert container. Then add `class="close"` and `data-dismiss="alert"` to a link or a button element (when you click on this the alert box will disappear).

Example

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="close" data-dismiss="alert">&times;</button>
  <strong>Success!</strong> Indicates a successful or positive action.
</div>
```

Tip: × (x) is an HTML entity that is the preferred icon for close buttons, rather than the letter "x". For a list of all HTML Entities, [visit our HTML Entities Reference](#).

Animated Alerts

Click on the "x" symbol to the right to close me. I will "fade" out.



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

Example

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

Complete Bootstrap 4 Alert Reference

For a complete reference of all alert options, methods and events, go to our [Bootstrap 4 JS Alert Reference](#).

Bootstrap 4 Buttons

Button Styles

Bootstrap 4 provides different styles of buttons:

Basic Primary Secondary Success Info Warning Danger Dark Light

[Link](#)

Example

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

The button classes can be used on an `<a>`, `<button>`, or `<input>` element:

Example

```
<a href="#" class="btn btn-info" role="button">Link Button</a>
<button type="button" class="btn btn-info">Button</button>
<input type="button" class="btn btn-info" value="Input Button">
<input type="submit" class="btn btn-info" value="Submit Button">
```

Why do we put a # in the href attribute of the link? Since we do not have any page to link it to, and we do not want to get a "404" message, we put # as the link. In real life it should of course been a real URL to the "Search" page.

Button Outline

Bootstrap 4 provides eight outline/bordered buttons:

Primary

Secondary

Success

Info

Warning

Danger

Dark

Light

Example

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

Button Sizes

Bootstrap 4 provides different button sizes:

Large

Default

Small

Example

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

Block Level Buttons

Add class `.btn-block` to create a block level button that spans the entire width of the parent element.

Full-Width Button

Example

```
<button type="button" class="btn btn-primary btn-block">Full-Width  
Button</button>
```

Active/Disabled Buttons

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

Active Primary Disabled Primary

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

Example

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

Bootstrap 4 Button Groups

Button Groups

Bootstrap 4 allows you to group a series of buttons together (on a single line) in a button group:

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

Example

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

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

Large Buttons:

Apple Samsung Sony

Default Buttons:

Apple Samsung Sony

Small Buttons:

Apple Samsung Sony

Example

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

Vertical Button Groups

Bootstrap 4 also supports vertical button groups:

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

Example

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

Nesting Button Groups & Dropdown Menus

Apple Samsung Sony ▾

Nest button groups to create dropdown menus (you will learn more about dropdowns in a later chapter):

Example

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

Split Button Dropdowns

Sony ▾

Example

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Sony</button>
  <button type="button" class="btn btn-primary dropdown-toggle dropdown-
toggle-split" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Tablet</a>
    <a class="dropdown-item" href="#">Smartphone</a>
  </div>
</div>
```

Vertical Button Group w/ Dropdown

Example

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

Button Groups Side by Side

Button groups are "inline" by default, which makes them appear side by side when you have multiple groups:

```
BMW  Mercedes  Volvo
```

Example

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

<div class="btn-group">
  <button type="button" class="btn btn-primary">BMW</button>
  <button type="button" class="btn btn-primary">Mercedes</button>
  <button type="button" class="btn btn-primary">Volvo</button>
</div>
```

Bootstrap 4 Badges

Example heading New

Badges are used to add additional information to any content. Use the `.badge` class together with a contextual class (like `.badge-secondary`) within `` elements to create rectangular badges. Note that badges scale to match the size of the parent element (if any):

Example

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

Contextual Badges

Primary Secondary Success Danger Warning Info Light Dark

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

Example

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

Pill Badges

Primary Secondary Success Danger Warning Info Light Dark

Use the `.badge-pill` class to make the badges more round:

Example

```
<span class="badge badge-pill badge-default">Default</span>
<span class="badge badge-pill badge-primary">Primary</span>
<span class="badge badge-pill badge-success">Success</span>
<span class="badge badge-pill badge-info">Info</span>
<span class="badge badge-pill badge-warning">Warning</span>
<span class="badge badge-pill badge-danger">Danger</span>
```

Badge inside an Element

An example of using a badge inside a button:

Messages 4

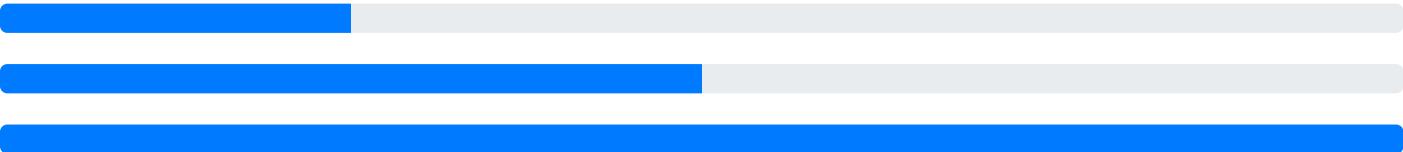
Example

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

Bootstrap 4 Progress Bars

Basic Progress Bar

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

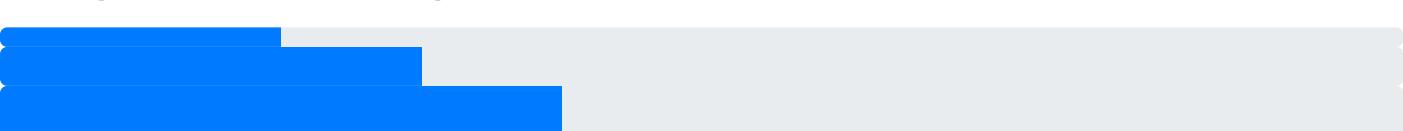


To create a default progress bar, add a `.progress` class to a container element and add the `.progress-bar` class to its child element. Use the CSS `width` property to set the width of the progress bar:

Example

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

Progress Bar Height



The height of the progress bar is 16px by default. Use the CSS `height` property to change it. Note that you must set the same height for the progress container and the progress bar:

Example

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

Progress Bar Labels

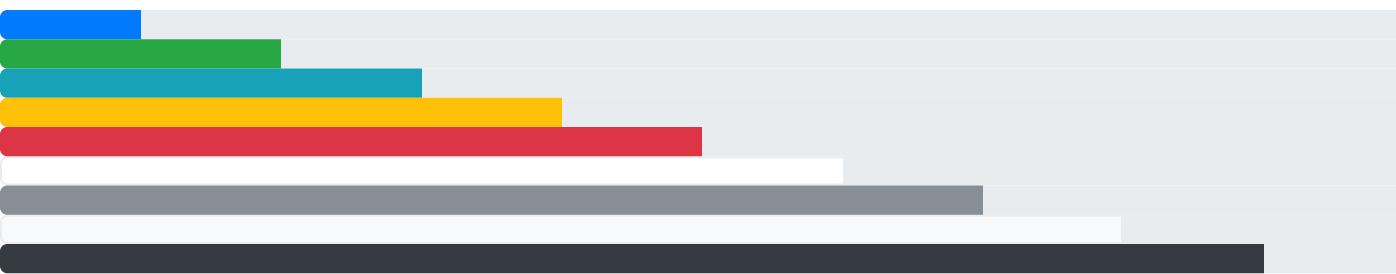
Add text inside the progress bar to show the visible percentage:

70%

Example

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

Colored Progress Bars



By default, the progress bar is blue (primary). Use any of the Bootstrap 4 contextual background classes to its color:

Example

```
<!-- Blue -->
<div class="progress">
  <div class="progress-bar" style="width:10%"></div>
</div>

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

<!-- Turquoise -->
<div class="progress">
  <div class="progress-bar bg-info" style="width:30%"></div>
</div>

<!-- Orange -->
<div class="progress">
  <div class="progress-bar bg-warning" style="width:40%"></div>
```

```

</div>

<!-- Red -->


<div class="progress-bar bg-danger" style="width:50%"></div>
</div>

<!-- White -->


<div class="progress-bar bg-white" style="width:60%"></div>
</div>

<!-- Grey -->


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

<!-- Light Grey -->


<div class="progress-bar bg-light" style="width:80%"></div>
</div>

<!-- Dark Grey -->


<div class="progress-bar bg-dark" style="width:90%"></div>
</div>


```

Striped Progress Bars



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

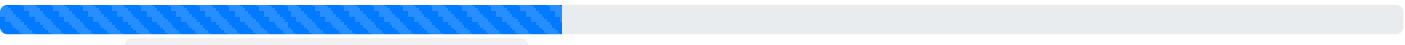
Example

```

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

```

Animated Progress Bar



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

Example

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

Multiple Progress Bars

Progress bars can also be stacked:

Free Space Warning Danger

Bootstrap 4 Pagination

Basic Pagination

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

Previous 1 2 3 Next

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

Example

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

Active State

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

Previous 1 2 3 Next

Example

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

Disabled State

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

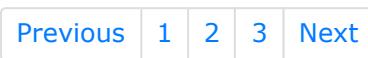
Previous [1](#) [2](#) [3](#) [Next](#)

Example

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

Pagination Sizing

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



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

Example

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

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

Pagination Alignment

Use utility classes to change the alignment of the pagination:

```
Previous 1 2 3 Next
```

```
Previous 1 2 3 Next
```

```
Previous 1 2 3 Next
```

Example

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

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

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

Tip: Read more about Bootstrap 4 Utility/Helper classes in our [BS4 Utilities Chapter](#).

Breadcrumbs

Another form for pagination, is breadcrumbs:

Photos / Summer 2017 / Italy / Rome

Example

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

Bootstrap 4 List Groups

Basic List Groups

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

First item

Second item

Third item

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

Example

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

Active State

Active item

Second item

Third item

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

Example

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

List Group With Linked Items

First item

Second item

Third item

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

Example

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

Disabled Item

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

Disabled item

Disabled item

Third item

Flush / Remove Borders

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

First item

Second item

Third item

Fourth item

Example

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

Contextual Classes

Contextual classes can be used to color list items:

Success item

Secondary item

Info item

Warning item

Danger item

Primary item

Dark item

Light item

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

Example

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Success item</li>
  <li class="list-group-item list-group-item-secondary">Secondary item</li>
  <li class="list-group-item list-group-item-info">Info item</li>
  <li class="list-group-item list-group-item-warning">Warning item</li>
  <li class="list-group-item list-group-item-danger">Danger item</li>
  <li class="list-group-item list-group-item-primary">Primary item</li>
  <li class="list-group-item list-group-item-dark">Dark item</li>
  <li class="list-group-item list-group-item-light">Light item</li>
</ul>
```

Link items with Contextual Classes

Action item

Success item

Secondary item

Info item

Warning item

Danger item

Primary item

Dark item

Light item

Example

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action">Action
  item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-success">Success item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-secondary">Secondary item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-info">Info item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-warning">Warning item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-danger">Danger item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-primary">Primary item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-dark">Dark item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-
  item-light">Light item</a>
</div>
```

List Group with Badges

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

Inbox	12
Ads	50
Junk	99

Example

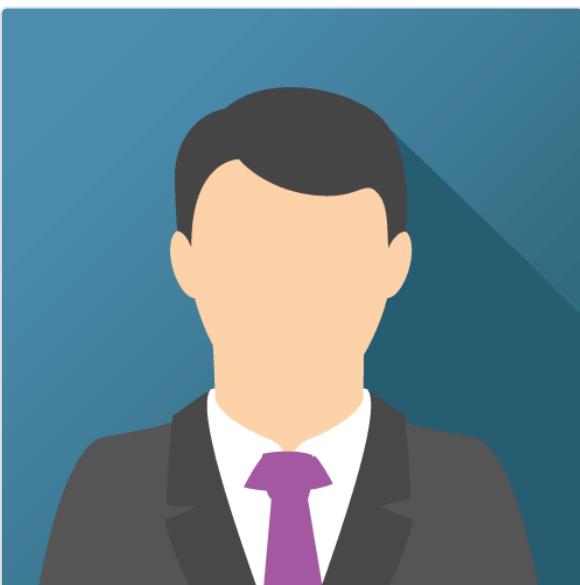
```
<ul class="list-group">
  <li class="list-group-item d-flex justify-content-between align-items-center">
    Inbox
    <span class="badge badge-primary badge-pill">12</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    Ads
    <span class="badge badge-primary badge-pill">50</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    Junk
    <span class="badge badge-primary badge-pill">99</span>
  </li>
</ul>
```

Tip: Read more about Bootstrap 4 Utility/Helper classes in our [BS4 Utilities Chapter](#).

Bootstrap 4 Cards

Cards

A card in Bootstrap 4 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.



John Doe

Some example text some
example text. John Doe is an
architect and engineer

[See Profile](#)

Basic Card

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

```
Basic card
```

Example

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

If you are familiar with Bootstrap 3, cards replace old panels, wells, and thumbnails.

Header and Footer

```
Header
```

```
Content
```

```
Footer
```

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

Example

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

Contextual Cards

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

Example

Basic card

Primary card

Success card

Info card

Warning card

Danger card

Secondary card

Dark card

Light card

Titles, text, and links

Card title

Some example text. Some example text.

Card link Another link

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

Example

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

Card Images



John Doe

Some example text some example text. John Doe is an architect and engineer

[See Profile](#)

Jane Doe

Some example text some example text. Jane Doe is an architect and engineer

[See Profile](#)



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

Example

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

Card Image Overlays

John Doe

Some example text some example text. Some example text some example text. Some example text some example text.

[See Profile](#)



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

Example

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

Card Columns

Some text inside the first card

Some text inside the third card

Some text inside the fifth card

Some text inside the second card

Some text inside the fourth card

Some text inside the sixth card

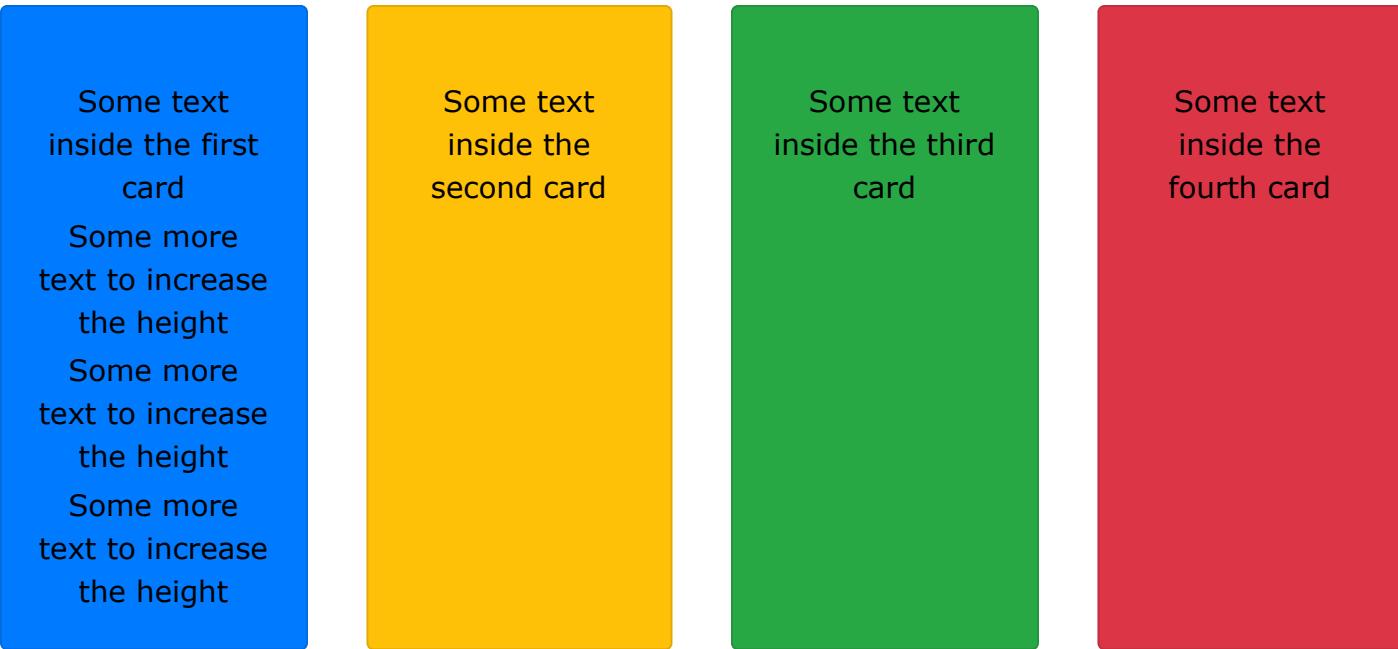
The `.card-columns` class creates a masonry-like grid of cards (like pinterest). The layout will automatically adjust as you insert more cards.

Note: The cards are displayed vertically on small screens (less than 576px):

Example

```
<div class="card-columns">
  <div class="card bg-primary">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the first card</p>
    </div>
  </div>
  <div class="card bg-warning">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the second card</p>
    </div>
  </div>
  <div class="card bg-success">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the third card</p>
    </div>
  </div>
  <div class="card bg-danger">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the fourth card</p>
    </div>
  </div>
  <div class="card bg-light">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the fifth card</p>
    </div>
  </div>
  <div class="card bg-info">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the sixth card</p>
    </div>
  </div>
</div>
```

Card Deck



The `.card-deck` class creates a grid of cards that are of **equal height and width**. The layout will automatically adjust as you insert more cards.

Note: The cards are displayed vertically on small screens (less than 576px):

Example

```
<div class="card-deck">
  <div class="card bg-primary">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the first card</p>
    </div>
  </div>
  <div class="card bg-warning">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the second card</p>
    </div>
  </div>
  <div class="card bg-success">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the third card</p>
    </div>
  </div>
  <div class="card bg-danger">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the fourth card</p>
    </div>
  </div>
</div>
```

Card Group

Some text inside the first card Some more text to increase the height Some more text to increase the height Some more text to increase the height	Some text inside the second card	Some text inside the third card	Some text inside the fourth card
--	----------------------------------	---------------------------------	----------------------------------

The `.card-group` class is similar to `.card-deck`. The only difference is that the `.card-group` class removes left and right margins between each card.

Note: The cards are displayed vertically on small screens (less than 576px), **WITH** top and bottom margin:

Example

```
<div class="card-group">
  <div class="card bg-primary">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the first card</p>
    </div>
  </div>
  <div class="card bg-warning">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the second card</p>
    </div>
  </div>
  <div class="card bg-success">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the third card</p>
    </div>
  </div>
  <div class="card bg-danger">
    <div class="card-body text-center">
      <p class="card-text">Some text inside the fourth card</p>
    </div>
  </div>
</div>
```

Bootstrap 4 Dropdowns

Basic Dropdown

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

Dropdown button ▾

Example

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

Example Explained

The `.dropdown` class indicates a dropdown menu.

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

Add the `.dropdown-menu` class to a `<div>` element to actually build the dropdown menu.

Then add the `.dropdown-item` class to each element (links or buttons) inside the dropdown menu.

Dropdown Divider

Dropdown button ▾

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

Example

```
<div class="dropdown-divider"></div>
```

Dropdown Header

Dropdown button ▾

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

Example

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

Disable and Active items

Dropdown button ▾

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

Example

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

Dropdown Position

Dropright ▾

Dropleft ▾

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

Dropright

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

Dropleft

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

Dropdown Menu Right

Wide dropdown button to demonstrate this example ▾

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

Example

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

Dropup

Dropdown button ▾

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

Example

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

Dropdown Text

Dropdown button ▾

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

Example

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

Grouped Buttons with a Dropdown

Apple Samsung Sony ▾

Example

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

Split Button Dropdowns

Primary ▾ Secondary ▾ Success ▾ Info ▾ Warning ▾ Danger ▾

Example

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

Vertical Button Group w/ Dropdown

Example

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

Complete Bootstrap 4 Dropdown Reference

For a complete reference of all dropdown options, methods and events, go to our [Bootstrap 4 JS Dropdown Reference](#).

Bootstrap 4 Collapse

Basic Collapsible

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

Click Me

Example

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

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

Example Explained

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

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

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

Example

```
<a href="#demo" data-toggle="collapse">Collapsible</a>

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

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

Example

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

Accordion

Collapsible Group Item #1

Lore ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Collapsible Group Item #2

Collapsible Group Item #3

The following example shows a simple accordion by extending the card component.

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

Example

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

```

</div>

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

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

</div>

```

Complete Bootstrap 4 Collapse Reference

For a complete reference of all collapse options, methods and events, go to our [Bootstrap 4 JS Collapse Reference](#).

Bootstrap 4 Navs

Nav Menus

Link Link Link Disabled

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

Example

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

Aligned Nav

Link Link Link Disabled

Link Link Link Disabled

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

Example

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

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

Vertical Nav

Link

Link

Link

Disabled

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

Example

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

Tabs

Active Link Link Disabled

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

Example

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

Pills

Active Link Link Disabled

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

Example

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

Justified Tabs/pills

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

Active

Link

Link

Disabled

Active

Link

Link

Disabled

Example

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

Pills with Dropdown

Active Dropdown ▾ Link Disabled

Example

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

Tabs with Dropdown

Active Dropdown ▾ Link Disabled

Example

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

Toggleable / Dynamic Tabs

Home Menu 1 Menu 2

HOME

 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.

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

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

Example

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

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

Toggleable / Dynamic Pills

Home Menu 1 Menu 2

HOME

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

The same code applies to pills; only change the data-toggle attribute to `data-toggle="pill"` :

Example

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

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

Complete Bootstrap 4 Nav Reference

For a complete reference of all tab options, methods and events, go to our [Bootstrap 4 JS Tab Reference](#).

Bootstrap 4 Navigation Bar

Navigation Bars

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



Basic Navbar

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size. A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xl|lg|md|sm` (stacks the navbar vertically on extra large, large, medium or small screens).

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

```
Link 1 Link 2 Link 3
```

Example

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

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

</nav>
```

Vertical Navbar

Remove the `.navbar-expand-xl|lg|md|sm` class to create a vertical navigation bar:

Link 1

Link 2

Link 3

Example

```
<!-- A vertical navbar -->
<nav class="navbar bg-light">

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

</nav>
```

Centered Navbar

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

The following example will center the navigation bar on medium, large and extra large screens. On small screens it will be displayed vertically and left-aligned (because of the `.navbar-expand-sm` class):

Link 1 Link 2 Link 3

Example

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">
  <a class="navbar-brand" href="#">Logo</a>
  ...
</nav>
```

Colored Navbar

Active Link Link Disabled

Active Link Link Disabled

Active Link Link Disabled

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

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

Example

```
<!-- Grey with black text -->
<nav class="navbar navbar-expand-sm bg-light navbar-light">
  <ul class="navbar-nav">
    <li class="nav-item active">
      <a class="nav-link" href="#">Active</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Disabled</a>
    </li>
  </ul>
</nav>

<!-- Black with white text -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">...</nav>

<!-- Blue with white text -->
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">...</nav>
```

Active/disabled state: Add the `.active` class to an `<a>` element to highlight the current link, or the `.disabled` class to indicate that the link is un-clickable.

Brand / Logo

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

```
Logo Link 1 Link 2 Link 3
```

Example

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

When using the `.navbar-brand` class on images, Bootstrap 4 will automatically style the image to fit the navbar vertically.



```
Link 1 Link 2 Link 3
```

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <a class="navbar-brand" href="#">
    
  </a>
  ...
</nav>
```

Collapsing The Navigation Bar

Navbar



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

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

Example

```
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
    <!-- Brand -->
    <a class="navbar-brand" href="#">Navbar</a>

    <!-- Toggler/collapsible Button -->
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>

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

Tip: You can also remove the .navbar-expand-md class to ALWAYS hide navbar links and display the toggler button.

Navbar With Dropdown

Logo Link 1 Link 2 Dropdown link ▾

Navbars can also hold dropdown menus:

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Logo</a>

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

  <!-- Dropdown -->
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-
    toggle="dropdown">
      Dropdown link
    </a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Link 1</a>
      <a class="dropdown-item" href="#">Link 2</a>
      <a class="dropdown-item" href="#">Link 3</a>
    </div>
  </li>
</ul>
</nav>
```

Navbar Forms and Buttons

Search

Search

Add a `<form>` element with `class="form-inline"` to group inputs and buttons side-by-side:

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <form class="form-inline" action="/action_page.php">
    <input class="form-control mr-sm-2" type="text" placeholder="Search">
    <button class="btn btn-success" type="submit">Search</button>
  </form>
</nav>
```

You can also use other input classes, such as `.input-group-prepend` or `.input-group-append` to attach an icon or help text next to the input field. You will learn more about these classes in the Bootstrap Inputs chapter.

@ Username

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <form class="form-inline" action="/action_page.php">
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">@</span>
      </div>
      <input type="text" class="form-control" placeholder="Username">
    </div>
  </form>
</nav>
```

Navbar Text

Link 1 Link 2 Navbar text

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

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

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

  <!-- Navbar text-->
  <span class="navbar-text">
    Navbar text
  </span>

</nav>
```

Fixed Navigation Bar

Top Fixed Navbar

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

Scroll this page to see the effect

The navigation bar can also be fixed at the top or at the bottom of the page.

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

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

Example

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

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

Bottom Fixed Navbar

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

Scroll this page to see the effect

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-bottom">  
  ...  
</nav>
```

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

Sticky Navbar

A sticky navigation bar stays fixed at the top of the page when you scroll past it.

Scroll this page to see the effect. **Note:** sticky-top does not work in IE11 and earlier.

Some example text. Some example text. Some example text. Some example text. Some example text

Some example text. Some example text. Some example text. Some example text. Some example text

Example

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

Bootstrap 4 Forms

Bootstrap 4's Default Settings

Form controls automatically receive some global styling with Bootstrap:

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Bootstrap 4 Form Layouts

Bootstrap provides two types of form layouts:

- Stacked (full-width) form
- Inline form

Bootstrap 4 Stacked Form

Email:

Password:

Remember me

Submit

The following example creates a stacked form with two input fields, one checkbox, and a submit button.

Add a wrapper element with `.form-group`, around each form control, to ensure proper margins:

Example

```
<form action="/action_page.php">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="form-group form-check">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Bootstrap Inline Form

Email:

Password:

 Remember meSubmit

In an inline form, all of the elements are inline and left-aligned.

Note: This only applies to forms within viewports that are at least 576px wide. On screens smaller than 576px, it will stack horizontally.

Additional rule for an inline form:

- Add class `.form-inline` to the `<form>` element

The following example creates an inline form with two input fields, one checkbox, and one submit button:

Example

```
<form class="form-inline" action="/action_page.php">
  <label for="email">Email address:</label>
  <input type="email" class="form-control" id="email">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd">
  <div class="form-check">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Inline Form with Utilities

The inline form above feels "compressed", and will look much better with Bootstrap's spacing utilities. The following example adds a right margin (`.mr-sm-2`) to each input on all devices (small and up). And a margin bottom class (`.mb-2`) is used to style the input field when it breaks (goes from horizontal to vertical due to not enough space/width):

Email:

Password:

 Remember meSubmit

Example

```
<form class="form-inline" action="/action_page.php">
  <label for="email" class="mr-sm-2">Email address:</label>
  <input type="email" class="form-control mb-2 mr-sm-2" id="email">
  <label for="pwd" class="mr-sm-2">Password:</label>
  <input type="password" class="form-control mb-2 mr-sm-2" id="pwd">
  <div class="form-check mb-2 mr-sm-2">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary mb-2">Submit</button>
</form>
```

You will learn more about spacing and other "helper" classes in our [Bootstrap 4 Utilities Chapter](#).

Bootstrap 4 Form Inputs

Supported Form Controls

Bootstrap supports the following form controls:

- input
- textarea
- checkbox
- radio
- select

Bootstrap Input

Bootstrap supports all the HTML5 input types: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, and color.

Note: Inputs will NOT be fully styled if their type is not properly declared!

Name:

Password:

The following example contains two input elements; one of `type="text"` and one of `type="password"`. As we mentioned in the Forms chapter, we use the `.form-control` class to style inputs with full-width and proper padding, etc:

Example

```
<div class="form-group">
  <label for="usr">Name:</label>
  <input type="text" class="form-control" id="usr">
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd">
</div>
```

Bootstrap Textarea

Comment:

The following example contains a textarea:

Example

```
<div class="form-group">
  <label for="comment">Comment:</label>
  <textarea class="form-control" rows="5" id="comment"></textarea>
</div>
```

Bootstrap Checkboxes

- Option 1
- Option 2
- Option 3

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

The following example contains three checkboxes. The last option is disabled:

Example

```
<div class="form-check">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 1
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 2
  </label>
</div>
<div class="form-check disabled">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="" disabled>Option 3
  </label>
</div>
```

Example explained

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

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

Inline Checkboxes

Use the `.form-check-inline` class if you want the checkboxes to appear on the same line:

- Option 1
- Option 2
- Option 3

Example

```
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 1
  </label>
</div>
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 2
  </label>
</div>
<div class="form-check-inline disabled">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="" disabled>Option 3
  </label>
</div>
```

Bootstrap Radio Buttons

- Option 1
- Option 2
- Option 3

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

The following example contains three radio buttons. The last option is disabled:

Example

```
<div class="form-check">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio">Option 1
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio">Option 2
  </label>
</div>
<div class="form-check disabled">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio"
disabled>Option 3
  </label>
</div>
```

As with checkboxes, use the `.form-check-inline` class if you want the radio buttons to appear on the same line:

- Option 1
- Option 2
- Option 3

Example

```
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio">Option 1
  </label>
</div>
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio">Option 2
  </label>
</div>
<div class="form-check-inline disabled">
  <label class="form-check-label">
    <input type="radio" class="form-check-input" name="optradio"
disabled>Option 3
  </label>
</div>
```

Bootstrap Select List

Select list (select one):

1

Multiple select list (hold ctrl or shift (or drag with the mouse) to select more than one):

1

2

3

4

...

Select lists are used if you want to allow the user to pick from multiple options.

The following example contains a dropdown list (select list):

Example

```
<div class="form-group">
  <label for="sel1">Select list:</label>
  <select class="form-control" id="sel1">
    <option>1</option>
    <option>2</option>
    <option>3</option>
    <option>4</option>
  </select>
</div>
```

Form Control Sizing

Change the size of the form control with `.form-control-sm` or `.form-control-lg`:

Small form control

Default form control

Large form control

Example

```
<input type="text" class="form-control form-control-sm">
<input type="text" class="form-control form-control">
<input type="text" class="form-control form-control-lg">
```

Form Control with Plain Text

Use the `.form-control-plaintext` if you want to style the input field as plain text:

Default form control

Form control with plain tex

Example

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

Form Control File and Range

Add the `.form-control-range` class to `input type="range"` or `.form-control-file` to `input type="file"` to style a range control or a file field with full-width:



Example

```
<input type="range" class="form-control-range">
<input type="file" class="form-control-file border">
```

Bootstrap 4 Inputs Group

Bootstrap 4 Input Groups

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

Use `.input-group-prepend` to add the help text in front of the input, and `.input-group-append` to add it behind the input.

At last, add the `.input-group-text` class to style the specified help text.

@

Your Email

@example.com

Example

```
<form>
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <span class="input-group-text">@</span>
    </div>
    <input type="text" class="form-control" placeholder="Username">
  </div>

  <div class="input-group mb-3">
    <input type="text" class="form-control" placeholder="Your Email">
    <div class="input-group-append">
      <span class="input-group-text">@example.com</span>
    </div>
  </div>
</form>
```

Tip: We use the `.mb-3` utility class to ensure that the input group gets a proper margin bottom. Read more about utility classes in our [BS4 Utilities Chapter](#).

Input Group Sizing

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

Small

Default

Large

Example

```
<form>
  <div class="input-group mb-3 input-group-sm">
    <div class="input-group-prepend">
      <span class="input-group-text">Small</span>
    </div>
    <input type="text" class="form-control">
  </div>
</form>
<form>
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <span class="input-group-text">Default</span>
    </div>
    <input type="text" class="form-control">
  </div>
</form>
<form>
  <div class="input-group mb-3 input-group-lg">
    <div class="input-group-prepend">
      <span class="input-group-text">Large</span>
    </div>
    <input type="text" class="form-control">
  </div>
</form>
```

Multiple Inputs and Helpers

Add multiple inputs or addons:

Person	First Name	Last Name
One Two Three		

Example

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

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

Input Group with Checkboxes and Radios

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

- Some text
- Some text

Example

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

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

Input Group Buttons

Basic Button Some text

Search

Go

Something clever..

OK

Cancel

Example

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

<div class="input-group mb-3">
  <input type="text" class="form-control" placeholder="Search">
  <div class="input-group-append">
    <button class="btn btn-success" type="submit">Go</button>
  </div>
</div>

<div class="input-group mb-3">
  <input type="text" class="form-control" placeholder="Something clever..">
  <div class="input-group-append">
    <button class="btn btn-primary" type="button">OK</button>
    <button class="btn btn-danger" type="button">Cancel</button>
  </div>
</div>
```

Input Group with Dropdown Button

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

```
Dropdown button ▾ Username
```

Example

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

Input Group Labels

Put labels outside of the input group, and remember that the value of the for attribute should match the id of the input.

Click on the label and it will bring focus to the input:

Write your email here:

Email

@example.com

Example

```
<label for="demo">Write your email here:</label>
<div class="input-group mb-3">
  <input type="text" class="form-control" placeholder="Email" id="demo"
name="email">
  <div class="input-group-append">
    <span class="input-group-text">@example.com</span>
  </div>
</div>
```

Bootstrap 4 Custom Forms

Bootstrap 4 Custom Forms

Bootstrap 4 comes with customized form elements, that are meant to replace browser defaults:

Custom checkbox

Default checkbox

Custom radio

Default radio

Custom Range:



Default Range:



Custom select menu ▾

Default select menu ▾

Custom file upload

Custom Checkbox

To create a custom checkbox, wrap a container element, like `<div>`, with a class of `.custom-control` and `.custom-checkbox` around the checkbox. Then add the `.custom-control-input` to the input with `type="checkbox"`.

Tip: If you use labels for accompanying text, add the `.custom-control-label` class to it.

Note that the value of the `for` attribute should match the id of the checkbox:

Custom checkbox

Default checkbox

Example

```
<form>
  <div class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input" id="customCheck" name="example1">
    <label class="custom-control-label" for="customCheck">Check this custom checkbox</label>
  </div>
</form>
```

Custom Radio buttons

To create a custom radio button, wrap a container element, like `<div>`, with a class of `.custom-control` and `.custom-radio` around the radio button. Then add the `.custom-control-input` to the input with type="radio".

Tip: If you use labels for accompanying text, add the `.custom-control-label` class to it.

Note that the value of the `for` attribute should match the id of the radio:

Custom radio

Default radio

Example

```
<form>
  <div class="custom-control custom-radio">
    <input type="radio" class="custom-control-input" id="customRadio"
name="example1" value="customEx">
    <label class="custom-control-label" for="customRadio">Custom
radio</label>
  </div>
</form>
```

Inline Custom Form Controls

If you want the custom form controls to sit side by side (inline), add the `.custom-control-inline` to the wrapper/container:

Custom radio Custom radio

Example

```
<form>
  <div class="custom-control custom-radio custom-control-inline">
    <input type="radio" class="custom-control-input" id="customRadio"
name="example" value="customEx">
    <label class="custom-control-label" for="customRadio">Custom radio
1</label>
  </div>
  <div class="custom-control custom-radio custom-control-inline">
    <input type="radio" class="custom-control-input" id="customRadio2"
name="example" value="customEx">
    <label class="custom-control-label" for="customRadio2">Custom radio
2</label>
  </div>
</form>
```

Custom Select Menu

To create a custom select menu, add the `.custom-select` class to the `<select>` element:

Custom Select Menu ▾ Default select menu ▾

Example

```
<form>
  <select name="cars" class="custom-select-sm">
    <option selected>Custom Select Menu</option>
    <option value="volvo">Volvo</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
</form>
```

Custom Select Menu Size

Use the `.custom-select-sm` class to create a small select menu and the `.custom-select-lg` class for a large one:

Small Custom Select Menu  Default Custom Select Menu  Large Custom Select Menu 

Example

```
<form>
  <!-- Small -->
  <select name="cars" class="custom-select-sm">
    <option selected>Small Custom Select Menu</option>
    <option value="volvo">Volvo</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>

  <!-- Large -->
  <select name="cars" class="custom-select-lg">
    <option selected>Large Custom Select Menu</option>
    <option value="volvo">Volvo</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
</form>
```

Custom Range

To create a custom select menu, add the `.custom-range` class to an input with type="

<range>":

Custom range



Example

```
<form>
  <label for="customRange">Custom range</label>
  <input type="range" class="custom-range" id="customRange" name="points1">
</form>
```

Custom File Upload

To create a custom file upload, wrap a container element with a class of `.custom-file` around the input with type="file". Then add the `.custom-control-input` to it.

Tip: If you use labels for accompanying text, add the `.custom-control-label` class to it.

Note that the value of the for attribute should match the id of the checkbox:

Choose file

ファイルを選択 選択されていません

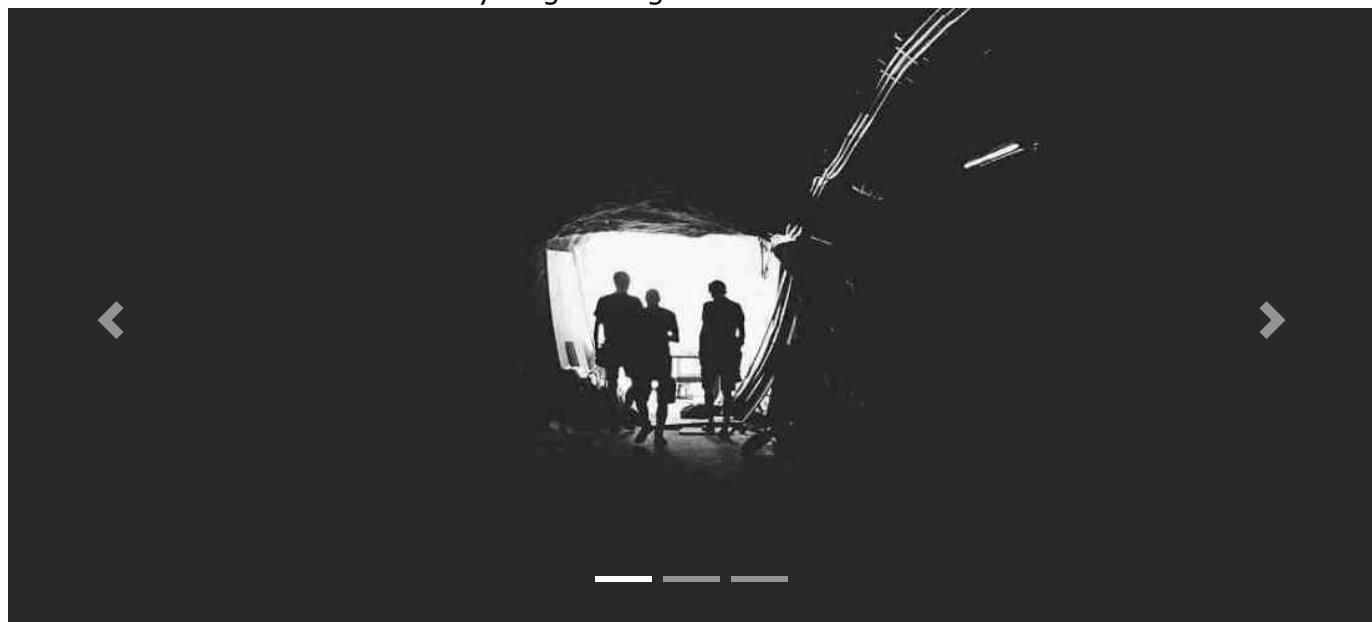
Example

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

Bootstrap 4 Carousel

Bootstrap 4 Carousel

The Carousel is a slideshow for cycling through elements.



How To Create a Carousel

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

Example

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

  
  <ul class="carousel-indicators">
    <li data-target="#demo" data-slide-to="0" class="active"></li>
    <li data-target="#demo" data-slide-to="1"></li>
    <li data-target="#demo" data-slide-to="2"></li>
  </ul>

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

  
  <a class="carousel-control-prev" href="#demo" data-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </a>
  <a class="carousel-control-next" href="#demo" data-slide="next">
    <span class="carousel-control-next-icon"></span>
  </a>

</div>
```

Example explained

A description of what each class from the example above do:

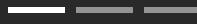
Class	Description
.carousel	Creates a carousel
.carousel-indicators	Adds indicators for the carousel. These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing)
.carousel-inner	Adds slides to the carousel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

Add Captions to Slides



Los Angeles

We had such a great time in LA!



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

Example

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

Complete Bootstrap Carousel Reference

For a complete reference of all carousel options, methods and events, go to our [Bootstrap JS Carousel Reference](#).

Bootstrap 4 Modal

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

[Open modal](#)

How To Create a Modal

The following example shows how to create a basic modal:

Example

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

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

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

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

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

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

Add animation

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

Example

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

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

Modal Size

Change the size of the modal by adding the `.modal-sm` class for small modals or `.modal-lg` class for large modals.

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

Small Modal

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

Large Modal

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

By default, modals are medium in size.

Centered Modal

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

Example

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

Complete Bootstrap Modal Reference

For a complete reference of all modal options, methods and events, go to our [Bootstrap JS Modal Reference](#).

Bootstrap 4 Tooltip

Bootstrap 4 Tooltip

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

Hover over me Hover over me

How To Create a Tooltip

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

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

```
<a href="#" data-toggle="tooltip" title="Hooray!">Hover over me</a>
```

Note: Tooltips must be initialized with jQuery: select the specified element and call the `tooltip()` method.

The following code will enable all tooltips in the document:

Example

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

Positioning Tooltips

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

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

Example

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

Complete Bootstrap Tooltip Reference

For a complete reference of all tooltip options, methods and events, go to our [Bootstrap JS Tooltip Reference](#).

Bootstrap 4 Popover

Bootstrap 4 Popover

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

[Toggle popover](#) [Toggle popover](#)

How To Create a Popover

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

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

```
<a href="#" data-toggle="popover" title="Popover Header" data-content="Some content inside the popover">Toggle popover</a>
```

Note: Popovers must be initialized with jQuery: select the specified element and call the `popover()` method.

The following code will enable all popovers in the document:

Example

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

Positioning Popovers

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

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

Example

```
<a href="#" title="Header" data-toggle="popover" data-placement="top" data-content="Content">Click</a>
<a href="#" title="Header" data-toggle="popover" data-placement="bottom" data-content="Content">Click</a>
<a href="#" title="Header" data-toggle="popover" data-placement="left" data-content="Content">Click</a>
<a href="#" title="Header" data-toggle="popover" data-placement="right" data-content="Content">Click</a>
```

Closing Popovers

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

Example

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

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

Example

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

Complete Bootstrap Popover Reference

For a complete reference of all popover options, methods and events, go to our [Bootstrap JS Popover Reference](#).

Bootstrap 4 Scrollspy (Advanced)

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

Section 1

Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this section and look at the navigation bar while scrolling!

Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this section and look at the navigation bar while scrolling!

How To Create a Scrollspy

The following example shows how to create a scrollspy:

Example

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

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

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

Example Explained

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

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

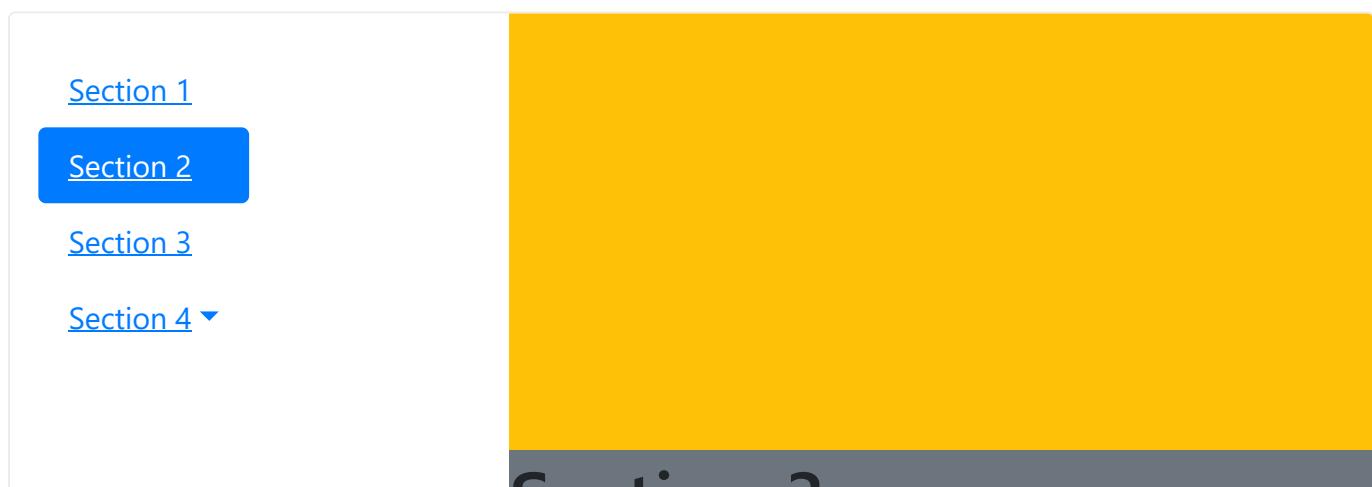
Note that scrollable elements must match the ID of the links inside the navbar's list items (`<div id="section1">` matches ``).

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

Default is 10 pixels.

Requires relative positioning: The element with `data-spy="scroll"` requires the CSS `position` property, with a value of "relative" to work properly.

Scrollspy Vertical Menu



In this example, we use Bootstrap's vertical navigation pills as menu:

Example

```
<body data-spy="scroll" data-target="#myScrollspy" data-offset="1">

<div class="container-fluid">
  <div class="row">
    <nav class="col-sm-3 col-4" id="myScrollspy">
      <ul class="nav nav-pills flex-column">
        <li class="nav-item">
          <a class="nav-link active" href="#section1">Section 1</a>
        </li>
        ...
      </ul>
    </nav>
    <div class="col-sm-9 col-8">
      <div id="section1">
        <h1>Section 1</h1>
        <p>Try to scroll this page and look at the menu while scrolling!
      </p>
      </div>
      ...
      </div>
    </div>
  </div>
</body>
```

Complete Bootstrap Scrollspy Reference

For a complete reference of all scrollspy options, methods and events, go to our [Bootstrap JS Scrollspy Reference](#).

Bootstrap 4 Utilities

Bootstrap 4 has a lot of utility/helper classes to quickly style elements without using any CSS code.

Borders

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

Example



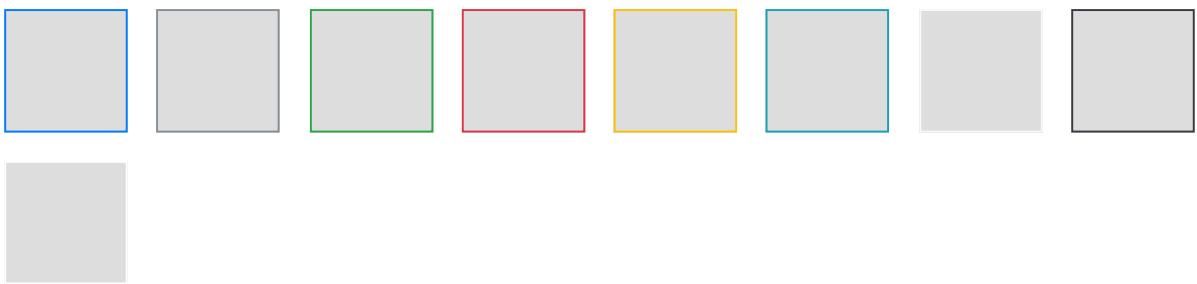
Example

```
<span class="border"></span>
<span class="border border-0"></span>
<span class="border border-top-0"></span>
<span class="border border-right-0"></span>
<span class="border border-bottom-0"></span>
<span class="border border-left-0"></span>
```

Border Color

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

Example



Example

```
<span class="border border-primary"></span>
<span class="border border-secondary"></span>
<span class="border border-success"></span>
<span class="border border-danger"></span>
<span class="border border-warning"></span>
<span class="border border-info"></span>
<span class="border border-light"></span>
<span class="border border-dark"></span>
<span class="border border-white"></span>
```

Border Radius

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

Example



Example

```
<span class="rounded"></span>
<span class="rounded-top"></span>
<span class="rounded-right"></span>
<span class="rounded-bottom"></span>
<span class="rounded-left"></span>
<span class="rounded-circle"></span>
<span class="rounded-0"></span>
```

Float and Clearfix

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

Example

Float left

Float right

Example

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

Responsive Floats

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

Example

Float right on medium screens or wider

Float right on small screens or wider

Float right on large screens or wider

Float right on extra large screens or wider

Float none

```
<div class="float-sm-right">Float right on small screens or wider</div><br>
<div class="float-md-right">Float right on medium screens or wider</div>
<br>
<div class="float-lg-right">Float right on large screens or wider</div><br>
<div class="float-xl-right">Float right on extra large screens or
wider</div><br>
<div class="float-none">Float none</div>
```

Center Align

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

Example

```
Centered
```

Example

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

Width

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

Example

Width 25%

Width 50%

Width 75%

Width 100%

Max Width 100%

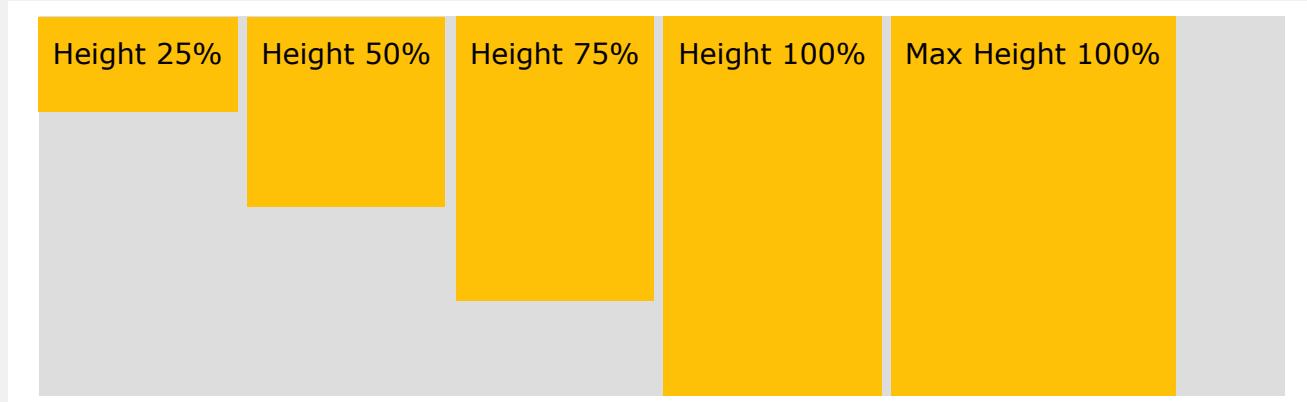
Example

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

Height

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

Example



Example

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

Spacing

Bootstrap 4 has a wide range of responsive margin and padding utility classes. They work for all breakpoints: `xs` ($\leq 576\text{px}$), `sm` ($\geq 576\text{px}$), `md` ($\geq 768\text{px}$), `lg` ($\geq 992\text{px}$) or `xl` ($\geq 1200\text{px}$):

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

Where `property` is one of:

- `m` - sets `margin`
- `p` - sets `padding`

Where `sides` is one of:

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

Where size is one of:

- `0` - sets `margin` or `padding` to `0`
- `1` - sets `margin` or `padding` to `.25rem` (4px if font-size is 16px)
- `2` - sets `margin` or `padding` to `.5rem` (8px if font-size is 16px)
- `3` - sets `margin` or `padding` to `1rem` (16px if font-size is 16px)
- `4` - sets `margin` or `padding` to `1.5rem` (24px if font-size is 16px)
- `5` - sets `margin` or `padding` to `3rem` (48px if font-size is 16px)
- `auto` - sets `margin` to auto

Example

I only have a top padding (`1.5rem` = 24px)

I have a padding on all sides (`3rem` = 48px)

I have a margin on all sides (`3rem` = 48px) and a bottom padding (`3rem` = 48px)

Example

```
<div class="pt-4 bg-warning">I only have a top padding (1.5rem = 24px)
</div>
<div class="p-5 bg-success">I have a padding on all sides (3rem = 48px)
</div>
<div class="m-5 pb-5 bg-info">I have a margin on all sides (3rem = 48px)
and a bottom padding (3rem = 48px)</div>
```

More Spacing Examples

.m-# / m-*-#	margin on all sides	Try it
.mt-# / mt-*-#	margin top	Try it
.mb-# / mb-*-#	margin bottom	Try it
.ml-# / ml-*-#	margin left	Try it
.mr-# / mr-*-#	margin right	Try it
.mx-# / mx-*-#	margin left and right	Try it
.my-# / my-*-#	margin top and bottom	Try it
.p-# / p-*-#	padding on all sides	Try it
.pt-# / pt-*-#	padding top	Try it
.pb-# / pb-*-#	padding bottom	Try it
.pl-# / pl-*-#	padding left	Try it
.pr-# / pr-*-#	padding right	Try it
.py-# / py-*-#	padding top and bottom	Try it
.px-# / px-*-#	padding left and right	Try it

Shadows

Use the `shadow-` classes to add shadows to an element:

Example

No shadow

Small shadow

Default shadow

Large shadow

Example

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

Vertical Align

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

Example

```
baseline top middle bottom text-top text-bottom
```

Example

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

Responsive Embeds

Create responsive video or slideshow embeds based on the width of the parent.

Add the `.embed-responsive-item` to any embed elements (like `<iframe>` or `<video>`) in a parent element with `.embed-responsive` and an aspect ratio of your choice:

Example

Example

```
<!-- 21:9 aspect ratio -->
<div class="embed-responsive embed-responsive-21by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 16:9 aspect ratio -->
<div class="embed-responsive embed-responsive-16by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 4:3 aspect ratio -->
<div class="embed-responsive embed-responsive-4by3">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 1:1 aspect ratio -->
<div class="embed-responsive embed-responsive-1by1">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

Visibility

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

Example

```
I am visible
```

Example

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

Position

Use the `.fixed-top` class to make any element fixed/stay at the **top** of the page:

Top Fixed Navbar

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

Scroll this page to see the effect

Example

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

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

Bottom Fixed Navbar

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

Scroll this page to see the effect

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-bottom">  
  ...  
</nav>
```

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

Sticky Navbar

A sticky navigation bar stays fixed at the top of the page when you scroll past it.

Scroll this page to see the effect. **Note:** sticky-top does not work in IE11 and earlier.

Some example text. Some example text. Some example text. Some example text. Some example text

Some example text. Some example text. Some example text. Some example text. Some example text

Example

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

Close icon

Use the `.close` class to style a close icon. This is often used for alerts and modals. Note that we use the `×` symbol to create the actual icon (a better looking "x"). Also note that it floats right by default:

Example

X

Example

```
<button type="button" class="close">&times;</button>
```

Screenreaders

Use the `.sr-only` class to hide an element on all devices, except screen readers:

Example

```
<span class="sr-only">I will be hidden on all screens except for screen  
readers.</span>
```

Typography/Text Classes

As described in the [Typography](#) chapter, here is a list of all typography/text classes:

Class	Description	Example
<code>.display-*</code>	Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are four classes to choose from: <code>.display-1</code> , <code>.display-2</code> , <code>.display-3</code> , <code>.display-4</code>	Try it
<code>.font-weight-bold</code>	Bold text	Try it
<code>.font-weight-normal</code>	Normal text	Try it
<code>.font-weight-light</code>	Light weight text	Try it
<code>.font-italic</code>	Italic text	Try it
<code>.lead</code>	Makes a paragraph stand out	Try it
<code>.small</code>	Indicates smaller text (set to 85% of the size of the parent)	Try it

.text-left	Indicates left-aligned text	Try it
.text-center	Indicates center-aligned text	Try it
.text-right	Indicates right-aligned text	Try it
.text-justify	Indicates justified text	Try it
.text-monospace	Monospaced text	Try it
.text nowrap	Indicates no wrap text	Try it
.text-lowercase	Indicates lowercased text	Try it
.text-uppercase	Indicates uppercased text	Try it
.text-capitalize	Indicates capitalized text	Try it
.initialism	Displays the text inside an <code><abbr></code> element in a slightly smaller font size	Try it
.list-unstyled	Removes the default list-style and left margin on list items (works on both <code></code> and <code></code>). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well)	Try it
.list-inline	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)	Try it
.pre-scrollable	Makes a <code><pre></code> element scrollable	Try it

Block Elements

To make an element into a block element, add the `.d-block` class. Use any of the `d-*-block` classes to control WHEN the element should be a block element on a specific screen width:

Example

```
d-block
d-sm-block d-md-block d-lg-block d-xl-block
```

Example

```
<span class="d-block bg-success">d-block</span>
<span class="d-sm-block bg-success">d-sm-block</span>
<span class="d-md-block bg-success">d-md-block</span>
```

```
<span class="d-lg-block bg-success">d-lg-block</span>
<span class="d-xl-block bg-success">d-xl-block</span>
```

Flex

Use `.flex-*` classes to control the layout with flexbox.

Read more about [Bootstrap 4 Flex](#), in our next chapter.

Example

Horizontal:



Vertical:



Bootstrap 4 Flex

Bootstrap 4 Flex

Use flex classes to control the layout of Bootstrap 4 components.

Flexbox

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

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning. If you are new to flex, you can read about it in our [CSS Flexbox Tutorial](#).

Note: Flexbox is not supported in IE9 and earlier versions.

If you require IE8-9 support, use [Bootstrap 3](#). It is the most stable version of Bootstrap, and it is still supported by the team for critical bugfixes and documentation changes. However, no new features will be added to it.

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

Example

A horizontal flexbox container with three items: 'Flex item 1' (teal), 'Flex item 2' (yellow), and 'Flex item 3' (blue).

Example

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

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

Example

```
Flex item 1 Flex item 2 Flex item 3
```

Example

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

Horizontal Direction

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

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

Example

```
Flex item 1 Flex item 2 Flex item 3
```

```
Flex item 3 Flex item 2 Flex item 1
```

Example

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

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

Vertical Direction

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

Example



Example

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

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

Justify Content

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

Example



Example

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

Fill / Equal Widths

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

Example



Example

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

Grow

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

Example



Flex item 1 Flex item 2 Flex item 3

Example

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

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

Order

Change the visual order of a specific flex item(s) with the `.order` classes. Valid classes are from 0 to 12, where the lowest number has highest priority (order-1 is shown before order-2, etc..):

Example



Flex item 3 Flex item 2 Flex item 1

Example

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

Auto Margins

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

Example



Example

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

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

Wrap

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

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

`flex-wrap` `flex-wrap-reverse` `flexnowrap`

Example

Flex item 1	Flex item 2	Flex item 3	Flex item 4	Flex item 5	Flex item 6
Flex item 7	Flex item 8	Flex item 9	Flex item 10	Flex item 11	Flex item 12
Flex item 13	Flex item 14	Flex item 15		Flex item 16	Flex item 17
Flex item 18	Flex item 19	Flex item 20		Flex item 21	Flex item 22
Flex item 23	Flex item 24	Flex item 25			

Example

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

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

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

Align Content

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

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

`align-content-start` `align-content-end` `align-content-center` `align-content-around`
`align-content-stretch`

Example

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

Align Items

Control the vertical alignment of **single rows** of flex items with the `.align-items-*` classes.

Valid classes are `.align-items-start`, `.align-items-end`, `.align-items-center`,
`.align-items-baseline`, and `.align-items-stretch` (default).

Click on the buttons below to see the difference between the five classes:

`align-items-start` `align-items-end` `align-items-center` `align-items-baseline`
`align-items-stretch`

```
Flex item 1 Flex item 2 Flex item 3
```

Example

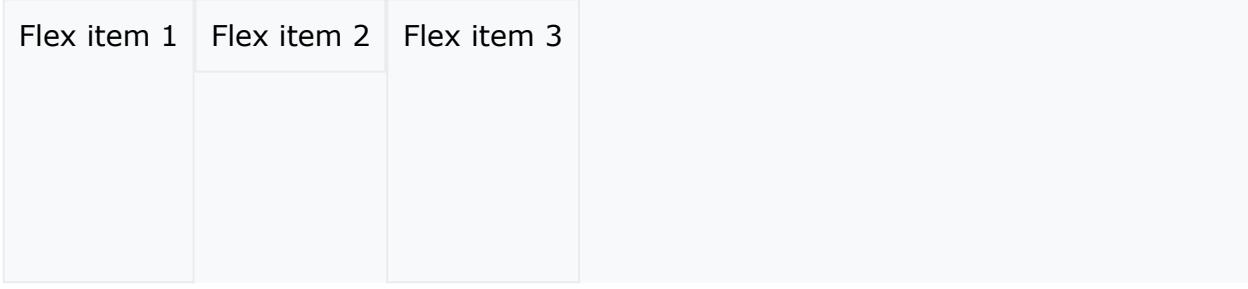
```
<div class="d-flex align-items-start">..</div>  
  
<div class="d-flex align-items-end">..</div>  
  
<div class="d-flex align-items-center">..</div>  
  
<div class="d-flex align-items-baseline">..</div>  
  
<div class="d-flex align-items-stretch">..</div>
```

Align Self

Control the vertical alignment of **a specified flex item** with the `.align-self-*` classes.
Valid classes are `.align-self-start`, `.align-self-end`, `.align-self-center`, `.align-self-baseline`, and `.align-self-stretch` (default).

Click on the buttons below to see the difference between the five classes:

`align-self-start` `align-self-end` `align-self-center` `align-self-baseline`
`align-self-stretch`



Flex item 1 Flex item 2 Flex item 3

Example

```
<div class="d-flex bg-light" style="height:150px">
  <div class="p-2 border">Flex item 1</div>
  <div class="p-2 border align-self-start">Flex item 2</div>
  <div class="p-2 border">Flex item 3</div>
</div>
```

Responsive Flex Classes

All flex classes comes with additional responsive classes, which makes it easy to set a specific flex class on a specific screen size.

The * symbol can be replaced with sm, md, lg or xl, which represents small, medium, large or xlarge screens.

Search for a specific flex class..

Class	Description	Example
Flex Container		
.d-*-flex	Creates a flexbox container for different screens	Try it
.d-*-inline-flex	Creates an inline flexbox container for different screens	Try it
Direction		
.flex-* -row	Display flex items horizontally on different screens	Try it
.flex-* -row-reverse	Display flex items horizontally, and right-aligned, on different screens	Try it
.flex-* -column	Display flex items vertically on different screens	Try it

`.flex-* -column-reverse`

Display flex items vertically, with reversed order, on different screens

[Try it](#)

Justified Content

`.justify-content-* -start`

Display flex items from the start (left-aligned) on different screens

[Try it](#)`.justify-content-* -end`

Display flex items at the end (right-aligned) on different screens

[Try it](#)`.justify-content-* -center`

Display flex items in the center of a flex container on different screens

[Try it](#)`.justify-content-* -between`

Display flex items in "between" on different screens

[Try it](#)`.justify-content-* -around`

Display flex items "around" on different screens

[Try it](#)

Fill / Equal Width

`.flex-* -fill`

Force flex items into equal widths on different screens

[Try it](#)

Grow

`.flex-* -grow-0`

Don't make the items grow on different screens

`.flex-* -grow-1`

Make items grow on different screens

Shrink

`.flex-* -shrink-0`

Don't make the items shrink on different screens

`.flex-* -shrink-1`

Make items shrink on different screens

Order

`.order-* -0-12`

Change the order from 0 to 12 on small screens

[Try it](#)

Wrap

`.flex-* -nowrap`

Don't wrap items on different screens

[Try it](#)`.flex-* -wrap`

Wrap items on different screens

[Try it](#)`.flex-* -wrap-reverse`

Reverse the wrapping of items on different screens

[Try it](#)

Align Content

.align-content-* - start	Align gathered items from the start on different screens	Try it
.align-content-* - end	Align gathered items at the end on different screens	Try it
.align-content-* - center	Align gathered items in the center on different screens	Try it
.align-content-* - around	Align gathered items "around" on different screens	Try it
.align-content-* - stretch	Stretch gathered items on different screens	Try it

Align Items

.align-items-* - start	Align single rows of items from the start on different screens	Try it
.align-items-* - end	Align single rows of items at the end on different screens	Try it
.align-items-* - center	Align single rows of items in the center on different screens	Try it
.align-items-* - baseline	Align single rows of items on the baseline on different screens	Try it
.align-items-* - stretch	Stretch single rows of items on different screens	Try it

Align Self

.align-self-* - start	Align a flex item from the start on different screens	Try it
.align-self-* - end	Align a flex item at the end on different screens	Try it
.align-self-* - center	Align a flex item in the center on different screens	Try it
.align-self-* - baseline	Align a flex item on the baseline on different screens	Try it
.align-self-* - stretch	Stretch a flex item on different screens	Try it

Bootstrap 4 Media Objects

Media Objects

Bootstrap provides an easy way to align media objects (like images or videos) together with content. Media objects are often used to display blog comments, tweets and so on:



John Doe *Posted on February 19, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



John Doe *Posted on February 20, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Basic Media Object



John Doe *Posted on February 19, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

To create a media object, add the `.media` class to a container element, and place media content inside a child container with the `.media-body` class. Add padding and margins as needed, with the spacing utilities:

Example

```
<div class="media border p-3">
  
  <div class="media-body">
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>
    <p>Lorem ipsum...</p>
  </div>
</div>
```

Nested Media Objects

Media objects can also be nested (a media object inside a media object):



John Doe *Posted on February 19, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Jane Doe *Posted on February 20, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

To nest media objects, place a new `.media` container inside the `.media-body` container:

Example

```
<div class="media border p-3">
  
  <div class="media-body">
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>
    <p>Lorem ipsum...</p>
    <div class="media p-3">
      
      <div class="media-body">
        <h4>Jane Doe <small><i>Posted on February 20 2016</i></small></h4>
        <p>Lorem ipsum...</p>
      </div>
    </div>
  </div>
</div>
```

Right-Aligned Media Image

John Doe *Posted on February 19, 2016*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Example

```
<div class="media border p-3">
  <div class="media-body">
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>
    <p>Lorem ipsum...</p>
  </div>
  
</div>
```

Top, Middle or Bottom Alignment

Use the flex utilities, `align-self-*` classes to place the media object on the top, middle or at the bottom:



Media Top

`align-self-top`

Media object is aligned to the top of the container.

Media object is aligned to the top of the container.

Media Middle

`align-self-center`

Media object is aligned to the center of the container.

Media object is aligned to the center of the container.

Media Bottom

`align-self-bottom`

Media object is aligned to the bottom of the container.

Media object is aligned to the bottom of the container.

Example

```
<!-- Media top -->
<div class="media">
  
  <div class="media-body">
    <h4>Media Top</h4>
    <p>Lorem ipsum...</p>
  </div>
</div>

<!-- Media middle -->
<div class="media">
  
  <div class="media-body">
    <h4>Media Middle</h4>
    <p>Lorem ipsum...</p>
  </div>
</div>

<!-- Media bottom -->
<div class="media">
  
  <div class="media-body">
    <h4>Media Bottom</h4>
    <p>Lorem ipsum...</p>
  </div>
</div>
```

Bootstrap 4 Filters (Advanced)

Bootstrap 4 Filters

Bootstrap does not have a component that allows filtering. However, we can use `jQuery` to filter / search for elements.

Filter Tables

Perform a case-insensitive search for items in a table:

Example

Type something in the input field to search the table for first names, last names or emails:

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@mail.com
July	Dooley	july@greatstuff.com
Anja	Ravendale	a_r@test.com

jQuery

```
<script>
$(document).ready(function(){
    $("#myInput").on("keyup", function() {
        var value = $(this).val().toLowerCase();
        $("#myTable tr").filter(function() {
            $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1);
        });
    });
</script>
```

Example explained: We use jQuery to loop through each table rows to check if there are any text values that matches the value of the input field. The `toggle` method hides the row (`display:none`) that does not match the search. We use the `toLowerCase()` method to convert the text to lower case, which makes the search case insensitive (allows "john", "John", and even "JOHN" on search).

Filter Lists

Perform a case-insensitive search for items in a list:

Example

Type something in the input field to search the list for items:

First item

Second item

Third item

Fourth

Filter Anything

Perform a case-insensitive search for text inside a div element:

Example

I am a paragraph.

I am a div element inside div.

I am a button

Another button

Another paragraph.

Bootstrap 4 Grid Stacked-to-horizontal

Bootstrap 4 Grid Example: Stacked-to-horizontal

We will create a basic grid system that starts out stacked on extra small devices, before becoming horizontal on larger devices.

The following example shows a simple "stacked-to-horizontal" two-column layout, meaning it will result in a 50%/50% split on all screens, except for extra small screens, which it will automatically stack (100%):



Example: Stacked-to-horizontal

```
<div class="container">
  <div class="row">
    <div class="col-sm-6 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-6 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Tip: The numbers in the `.col-sm-*` classes indicates how many columns the div should span (out of 12). So, `.col-sm-1` spans 1 column, `.col-sm-4` spans 4 columns, `.col-sm-6` spans 6 columns, etc. **Note:** Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns):

Tip: You can turn any fixed-width layout into a **full-width** layout by changing the `.container` class to `.container-fluid`:

Example: Fluid container

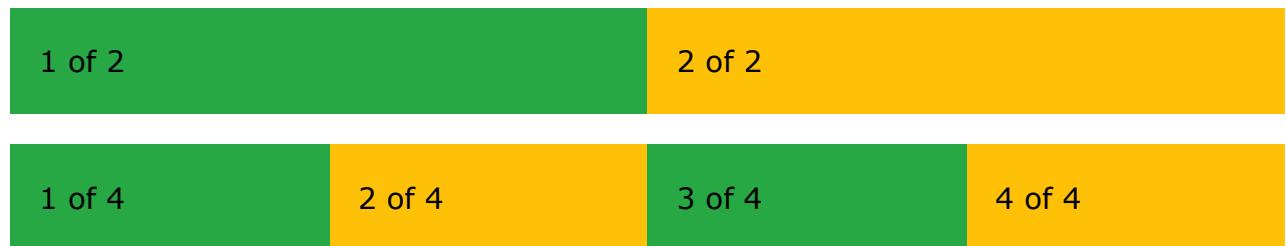
```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-6">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-6">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-size-*` and only use the `.col-size` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width. The *size* classes determines when the columns should be responsive:

```
<!-- Two columns: 50% width on all screens, except for extra small (100% width) -->
<div class="row">
  <div class="col-sm">1 of 2</div>
  <div class="col-sm">2 of 2</div>
</div>

<!-- Four columns: 25% width on all screens, except for extra small (100% width)-->
<div class="row">
  <div class="col-sm">1 of 4</div>
  <div class="col-sm">2 of 4</div>
  <div class="col-sm">3 of 4</div>
  <div class="col-sm">4 of 4</div>
</div>
```



Bootstrap 4 Grid Extra Small

Extra Small Grid Example

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

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

We will add the following classes to our two columns:

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

The following example will result in a 25%/75% split on all devices (extra small, small, medium, large and xllarge).

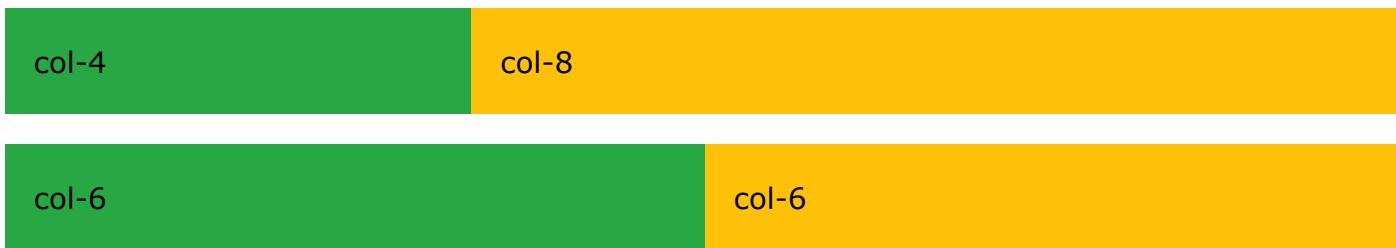


Example

```
<div class="container-fluid">
  <div class="row">
    <div class="col-3 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-9 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

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

For a 33.3%/66.6% split, you would use `.col-4` and `.col-8` (and for a 50%/50% split, you would use `.col-6` and `.col-6`):



Example

```
<!-- 33.3%/66.6% split -->
<div class="container-fluid">
  <div class="row">
    <div class="col-4 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-8 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>

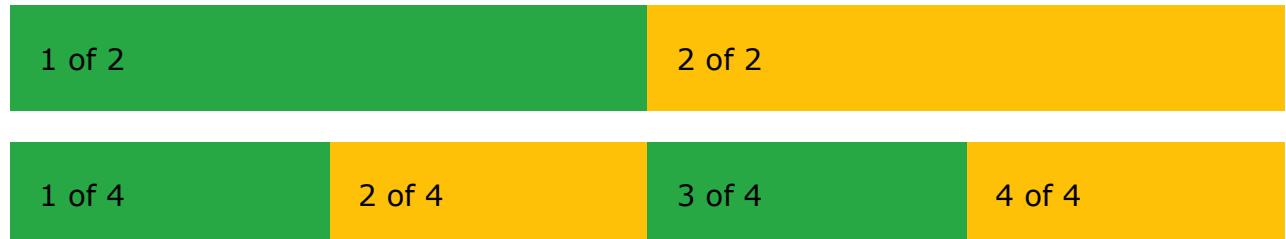
<!-- 50%/50% split -->
<div class="container-fluid">
  <div class="row">
    <div class="col-6 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-6 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-*` and only use the `.col` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width:

```
<!-- Two columns: 50% width-->
<div class="row">
  <div class="col">1 of 2</div>
  <div class="col">2 of 2</div>
</div>

<!-- Four columns: 25% width-->
<div class="row">
  <div class="col">1 of 4</div>
  <div class="col">2 of 4</div>
  <div class="col">3 of 4</div>
  <div class="col">4 of 4</div>
</div>
```



The next chapter shows how to add a different split percent for small devices.

Bootstrap 4 Grid Small

Small Grid Example

	Extra small		Medium	Large	Extra Large
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Screen width	<576px		>=768px	>=992px	>=1200px

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

Small devices are defined as having a screen width from **576 pixels to 767 pixels**.

For small devices we will use the `.col-sm-*` classes.

We will add the following classes to our two columns:

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

Now Bootstrap is going to say "at the small size, look for classes with -sm- in them and use those".

The following example will result in a 25%/75% split on small (and medium, large and xlarge) devices. On extra small devices, it will automatically stack (100%):

.col-sm-3

.col-sm-9

Example

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-3 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

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

For a 33.3%/66.6% split, you would use `.col-sm-4` and `.col-sm-8` (and for a 50%/50% split, you would use `.col-sm-6` and `.col-sm-6`):



Example

```
<!-- 33.3/66.6% split: -->
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-4 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-8 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>

<!-- 50%/50% split: -->
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-6 bg-success">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-6 bg-warning">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

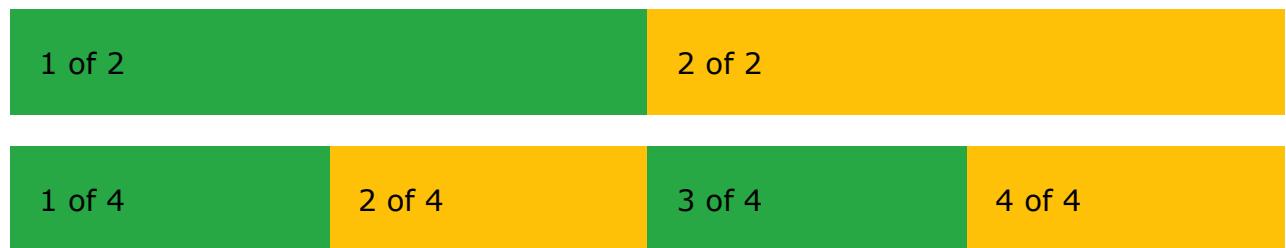
Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-sm-*` and only use the `.col-sm` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width.

If the screen size is **less than 576px**, the columns will stack horizontally:

```
<!-- Two columns: 50% width on all screens, except for extra small (100% width) -->
<div class="row">
  <div class="col-sm">1 of 2</div>
  <div class="col-sm">2 of 2</div>
</div>

<!-- Four columns: 25% width on all screens, except for extra small (100% width)-->
<div class="row">
  <div class="col-sm">1 of 4</div>
  <div class="col-sm">2 of 4</div>
  <div class="col-sm">3 of 4</div>
  <div class="col-sm">4 of 4</div>
</div>
```



The next chapter shows how to add a different split percent for medium devices.

Bootstrap 4 Grid Medium

Medium Grid Example

	Extra small	Small		Large	Extra Large
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Screen width	<576px	>=576px		>=992px	>=1200px

In the previous chapter, we presented a grid example with classes for small devices. We used two divs (columns) and we gave them a 25%/75% split:

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

But on medium devices the design may be better as a 50%/50% split.

Medium devices are defined as having a screen width from **768 pixels to 991 pixels**.

For medium devices we will use the `.col-md-*` classes:

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

Now Bootstrap is going to say "at the small size, look at classes with -sm- in them and use those. At the medium size, look at classes with -md- in them and use those".

The following example will result in a 25%/75% split on small devices and a 50%/50% split on medium (and large and xlarge) devices. On extra small devices, it will automatically stack (100%):

.col-sm-3 .col-md-6

.col-sm-9 .col-md-6

Example

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

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

Using Only Medium

In the example below, we only specify the `.col-md-6` class (without `.col-sm-*`). This means that medium, large and extra large devices will split 50%/50% - because the class scales up. However, for small and extra small devices, it will stack vertically (100% width):

Example

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

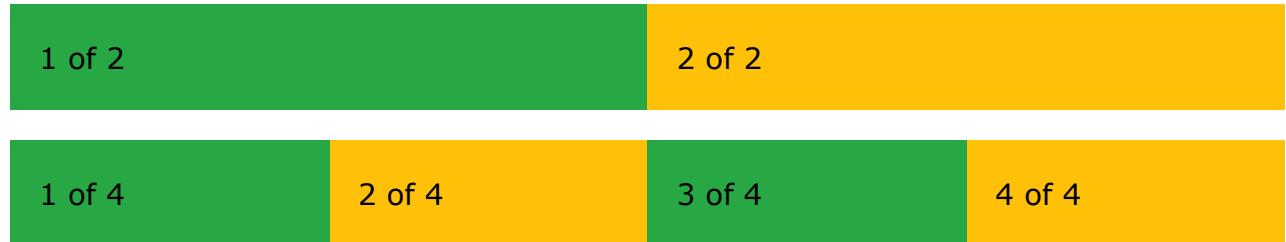
Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-md-*` and only use the `.col-md` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width.

If the screen size is **less than 768px**, the columns will stack horizontally:

```
<!-- Two columns: 50% width on medium and up-->
<div class="row">
  <div class="col-md">1 of 2</div>
  <div class="col-md">2 of 2</div>
</div>

<!-- Four columns: 25% width on medium and up -->
<div class="row">
  <div class="col-md">1 of 4</div>
  <div class="col-md">2 of 4</div>
  <div class="col-md">3 of 4</div>
  <div class="col-md">4 of 4</div>
</div>
```



The next chapter shows how to add a different split percent for large devices.

Bootstrap 4 Grid - Large

Large Grid Example

	Extra small	Small	Medium		Extra Large
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Screen width	<576px	=576px	=768px		=1200px

In the previous chapter, we presented a grid example with classes for small and medium devices. We used two divs (columns) and we gave them a 25%/75% split on small devices, and a 50%/50% split on medium devices:

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

But on large devices the design may be better as a 33%/66% split.

Large devices are defined as having a screen width from **992 pixels to 1199 pixels**.

For large devices we will use the `.col-lg-*` classes:

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

Now Bootstrap is going to say "at the small size, look at classes with -sm- in them and use those. At the medium size, look at classes with -md- in them and use those. At the large size, look at classes with the word -lg- in them and use those".

The following example will result in a 25%/75% split on small devices, a 50%/50% split on medium devices, and a 33%/66% split on large and xlarge devices. On extra small devices, it will automatically stack (100%):

.col-sm-3 .col-md-6 .col-lg-4	.col-sm-9 .col-md-6 .col-lg-8
-------------------------------	-------------------------------

Example

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-3 col-md-6 col-lg-4">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9 col-md-6 col-lg-8">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

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

Using Only Large

In the example below, we only specify the `.col-lg-6` class (without `.col-md-*` and/or `.col-sm-*`). This means that large and xlarge devices will split 50%/50%. However, for medium, small AND extra small devices, it will stack vertically (100% width):

Example

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

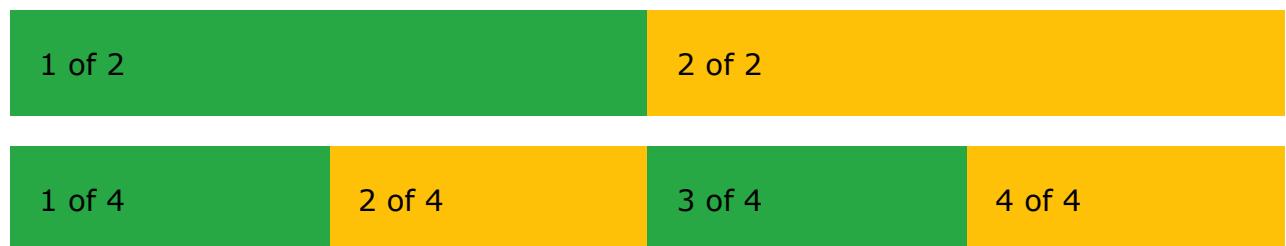
Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-lg-*` and only use the `.col-lg` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width.

If the screen size is **less than 992px**, the columns will stack horizontally:

```
<!-- Two columns: 50% width on large and up-->
<div class="row">
  <div class="col-lg">1 of 2</div>
  <div class="col-lg">2 of 2</div>
</div>

<!-- Four columns: 25% width on large and up -->
<div class="row">
  <div class="col-lg">1 of 4</div>
  <div class="col-lg">2 of 4</div>
  <div class="col-lg">3 of 4</div>
  <div class="col-lg">4 of 4</div>
</div>
```



Bootstrap 4 Grid - Extra Large

XLarge Grid Example

	Extra small	Small	Medium	Large	
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Screen width	<576px	=576px	=768px	=992px	

In the previous chapter, we presented a grid example with classes for small and medium devices. We used two divs (columns) and we gave them a 25%/75% split on small devices, and a 50%/50% split on medium devices and a 33%/66% split on large devices:

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

But on xlARGE devices the design may be better as a 20%/80% split.

Extra large devices are defined as having a screen width from **1200 pixels and above**.

For xlARGE devices we will use the `.col-xl-*` classes:

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

The following example will result in a 25%/75% split on small devices, a 50%/50% split on medium devices, and a 33%/66% split on large and a 20%/80% split on xlarge devices. On extra small devices, it will automatically stack (100%):

col-sm-3 col-md-6
col-lg-4 col-xl-2

col-sm-9 col-md-6 col-lg-8 col-xl-10

Example

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-3 col-md-6 col-lg-4 col-xl-2">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9 col-md-6 col-lg-8 col-xl-10">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Note: Make sure that the sum always adds up to 12.

Using Only XLarge

In the example below, we only specify the `.col-xl-6` class (without `.col-lg-*`, `.col-md-*` and/or `.col-sm-*`). This means that xlarge devices will split 50%/50%. However, for large, medium, small AND extra small devices, it will stack vertically (100% width):

Example

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

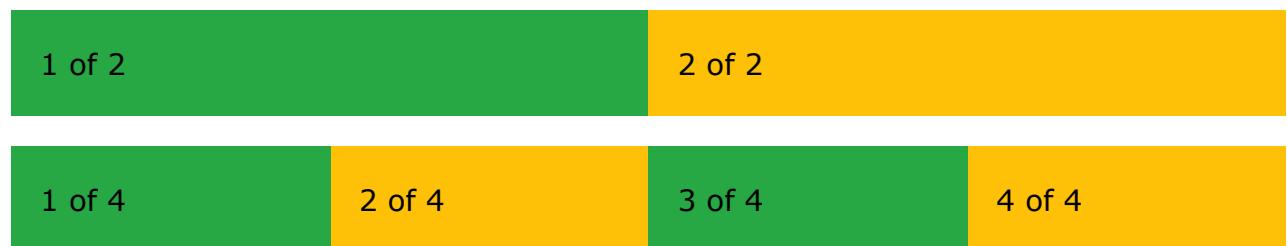
Auto Layout Columns

In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from `.col-xl-*` and only use the `.col-xl` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width.

If the screen size is **less than 1200px**, the columns will stack horizontally:

```
<!-- Two columns: 50% width on xlarge and up-->
<div class="row">
  <div class="col-xl">1 of 2</div>
  <div class="col-xl">2 of 2</div>
</div>

<!-- Four columns: 25% width on xlarge and up -->
<div class="row">
  <div class="col-xl">1 of 4</div>
  <div class="col-xl">2 of 4</div>
  <div class="col-xl">3 of 4</div>
  <div class="col-xl">4 of 4</div>
</div>
```



Bootstrap 4 Grid Examples

Below we have collected some examples of Bootstrap 4 grid layouts.

Three Equal Columns

Use the `.col` class on a specified number of elements and Bootstrap will recognize how many elements there are (and create equal-width columns). In the example below, we use three col elements, which gets a width of 33.33% each.



Example

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

Three Equal Columns Using Numbers

You can also use numbers to control the column width. Just make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns):



Example

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

Three Unequal columns

To create unequal columns, you have to use numbers. The following example will create a 25%/50%/25% split:



Example

```
<div class="row">
  <div class="col-3">col-3</div>
  <div class="col-6">col-6</div>
  <div class="col-3">col-3</div>
</div>
```

Setting One Column Width

However, it is enough to only set the width of one column, and have the sibling columns automatically resize around it. The following example will create a 25%/50%/25% split:



Example

```
<div class="row">
  <div class="col">col-3</div>
  <div class="col-6">col-6</div>
  <div class="col">col-3</div>
</div>
```

More Equal Columns

1 of 2	2 of 2				
1 of 4	2 of 4	3 of 4	4 of 4		
1 of 6	2 of 6	3 of 6	4 of 6	5 of 6	6 of 6

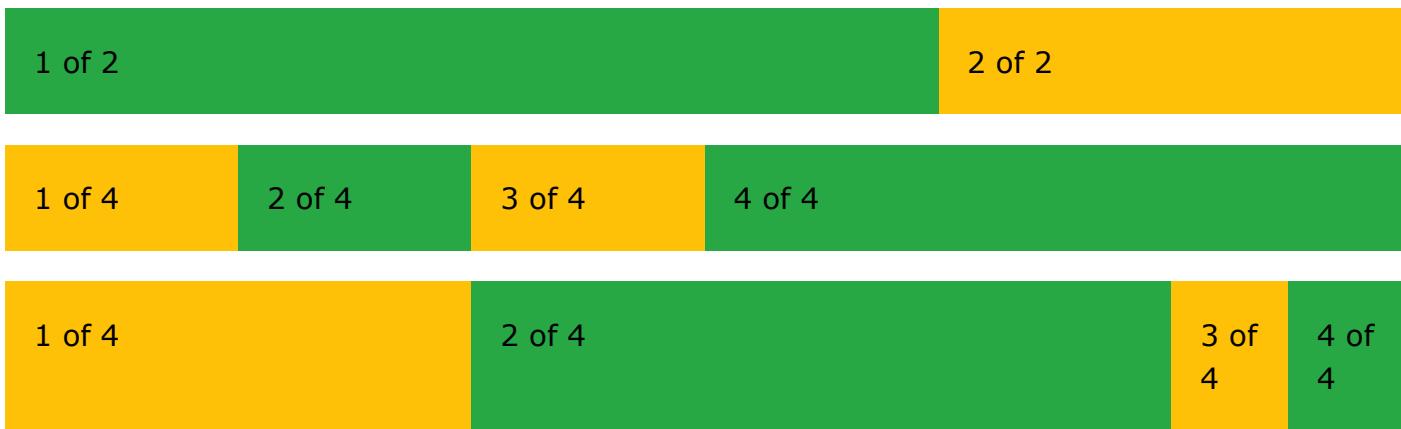
Example

```
<!-- Two equal columns -->
<div class="row">
  <div class="col">1 of 2</div>
  <div class="col">2 of 2</div>
</div>

<!-- Four equal columns -->
<div class="row">
  <div class="col">1 of 4</div>
  <div class="col">2 of 4</div>
  <div class="col">3 of 4</div>
  <div class="col">4 of 4</div>
</div>

<!-- Six equal columns -->
<div class="row">
  <div class="col">1 of 6</div>
  <div class="col">2 of 6</div>
  <div class="col">3 of 6</div>
  <div class="col">4 of 6</div>
  <div class="col">5 of 6</div>
  <div class="col">6 of 6</div>
</div>
```

More Unequal Columns



Example

```
<!-- Two Unequal Columns -->
<div class="row">
  <div class="col-8">1 of 2</div>
  <div class="col-4">2 of 2</div>
</div>

<!-- Four Unequal Columns -->
<div class="row">
  <div class="col-2">1 of 4</div>
  <div class="col-2">2 of 4</div>
  <div class="col-2">3 of 4</div>
  <div class="col-6">4 of 4</div>
</div>

<!-- Setting two column widths -->
<div class="row">
  <div class="col-4">1 of 4</div>
  <div class="col-6">2 of 4</div>
  <div class="col">3 of 4</div>
  <div class="col">4 of 4</div>
</div>
```

Equal Height

If one of the column is taller than the other (due to text or CSS height), the rest will follow:

Lorem ipsum dolor sit amet, cibo sensibus interesseret no sit. Et dolor possim volutpat qui. No malis tollit iriure eam, et vel tale zril blandit, rebum vidisse nostrum qui eu. No nostrud dolorem legendos mea, ea eum mucius oporteat platonem. Eam an case scribentur, ei clita causae cum, alia debet eu vel.	col	col
--	-----	-----

Example

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

Nested Columns



The following example shows how to create a two column layout, with another two columns inside one of the columns:

Example

```
<div class="row">
  <div class="col-8">
    .col-8
    <div class="row">
      <div class="col-6">.col-6</div>
      <div class="col-6">.col-6</div>
    </div>
  </div>
  <div class="col-4">.col-4</div>
</div>
```

Responsive Classes

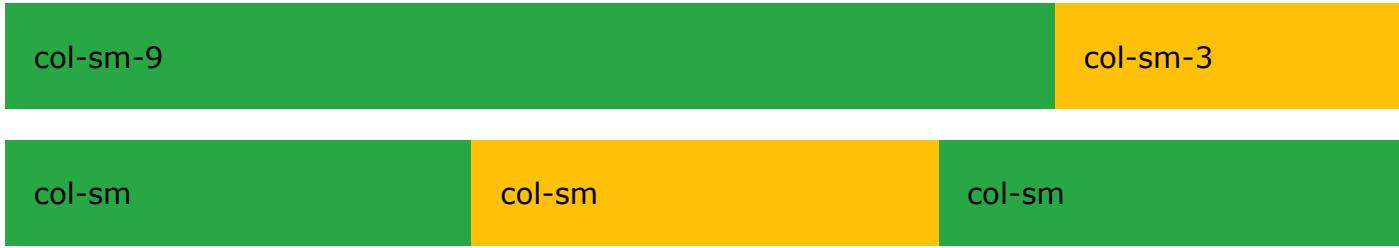
The Bootstrap 4 grid system has five classes:

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

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

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

Stacked to Horizontal

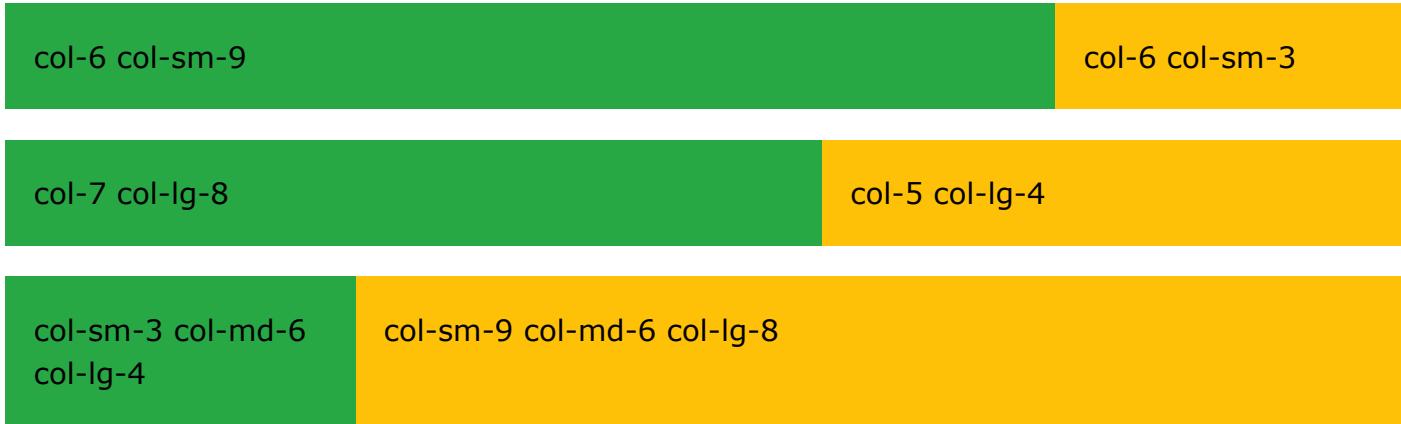


The following example shows how to create a column layout that starts out stacked on extra small devices, before becoming horizontal on larger devices (sm, md, lg and xl):

Example

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

Mix and Match



Example

```
<!-- 50%/50% split on extra small devices and 75%/25% split on larger devices -->
<div class="row">
  <div class="col-6 col-sm-9">col-6 col-sm-9</div>
  <div class="col-6 col-sm-3">col-6 col-sm-3</div>
</div>

<!-- 58%/42% split on extra small, small and medium devices and 66.3%/33.3% split on large and xlarge devices -->
<div class="row">
  <div class="col-7 col-lg-8">col-7 col-lg-8</div>
  <div class="col-5 col-lg-4">col-5 col-lg-4</div>
</div>

<!-- 25%/75% split on small devices, a 50%/50% split on medium devices, and a 33%/66% split on large and xlarge devices. On extra small devices, it will automatically stack (100%) -->
<div class="row">
  <div class="col-sm-3 col-md-6 col-lg-4">col-sm-3 col-md-6 col-lg-4</div>
  <div class="col-sm-9 col-md-6 col-lg-8">col-sm-9 col-md-6 col-lg-8</div>
</div>
```

No Gutters

Add the `.no-gutters` class to the `.row` container to remove gutters (extra space):

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
--	---

Example

```
<div class="row no-gutters">
```

Bootstrap 4 Class Reference

Complete List of All Bootstrap 4 Classes

Complete list of all Bootstrap 4 CSS classes with description and examples:

Class	Description	Example	Category
.active	Adds a white text color to the active link in a navbar .	Try it	Navbar
.active	Adds a blue background color to the active list item in a list group	Try it	List Groups
.active	Adds a dark-blue background color to simulate a "pressed" button	Try it	Buttons
.active	Adds a blue background color to the active dropdown item in a dropdown	Try it	Dropdowns
.active	Adds a blue background color to the active pagination link (to highlight the current page)	Try it	Pagination
.active	Displays/shows the current carousel item	Try it	Carousel
.alert	Creates an alert message box	Try it	Alerts
.alert-danger	Red alert. Indicates a dangerous or potentially negative action	Try it	Alerts
.alert-dark	Dark alert. Dark grey alert box	Try it	Alerts
.alert-dismissible	Indicates a closable alert box. Together with the <code>.close</code> class, this class is used to close the alert (adds extra padding)	Try it	Alerts
.alert-heading	Adds <code>color: inherit</code> to the specified element	Try it	Alerts
.alert-info	Teal alert. Indicates a neutral informative change or action	Try it	Alerts
.alert-light	Light alert. Light grey alert box	Try it	Alerts

Class	Description	Example	Category
.alert-link	Used on links inside alerts to provide matching colored links	Try it	Alerts
.alert-primary	Blue alert. Indicates an important action	Try it	Alerts
.alert-secondary	Grey alert. Indicates a "less" important action	Try it	Alerts
.alert-success	Green alert. Indicates a successful or positive action	Try it	Alerts
.alert-warning	Yellow alert. Indicates caution should be taken with this action	Try it	Alerts
.align-baseline	The element is aligned with the baseline of the parent. This is default	Try it	Utilities
.align-bottom	The element is aligned with the lowest element on the line	Try it	Utilities
.align-middle	The element is placed in the middle of the parent element	Try it	Utilities
.align-top	The element is aligned with the top of the tallest element on the line	Try it	Utilities
.align-text-top	The element is aligned with the top of the parent element's font	Try it	Utilities
.align-text-bottom	The element is aligned with the bottom of the parent element's font	Try it	Utilities
.align-content-around	Align gathered items "around"	Try it	Flex
.align-content-*~	Align gathered items "around" on different screens	Try it	Flex
.align-content-center	Align gathered items in the center	Try it	Flex
.align-content-*~center	Align gathered items in the center on different screens	Try it	Flex
.align-content-end	Align gathered items at the end	Try it	Flex

Class	Description	Example	Category
.align-content-end	Align gathered items at the end on different screens	Try it	Flex
.align-content-start	Align gathered items from the start	Try it	Flex
.align-content-*start	Align gathered items from the start on different screens	Try it	Flex
.align-content-stretch	Stretch gathered items	Try it	Flex
.align-content-*stretch	Stretch gathered items on different screens	Try it	Flex
.align-items-start	Align single rows of items from the start	Try it	Flex
.align-items-*start	Align single rows of items from the start on different screens	Try it	Flex
.align-items-end	Align single rows of items at the end	Try it	Flex
.align-items-*end	Align single rows of items at the end on different screens	Try it	Flex
.align-items-center	Align single rows of items in the center	Try it	Flex
.align-items-*center	Align single rows of items in the center on different screens	Try it	Flex
.align-items-baseline	Align single rows of items at the baseline	Try it	Flex
.align-items-*baseline	Align single rows of items at the baseline on different screens	Try it	Flex
.align-items-stretch	Stretch single rows of items	Try it	Flex
.align-items-*stretch	Stretch single rows of items on different screens	Try it	Flex
.align-self-start	Align a flex item from the start	Try it	Flex

Class	Description	Example	Category
.align-self-start	Align a flex item from the start on different screens	Try it	Flex
.align-self-end	Align a flex item at the end	Try it	Flex
.align-self-*-end	Align a flex item at the end on different screens	Try it	Flex
.align-self-center	Align a flex item in the center	Try it	Flex
.align-self-*-center	Align a flex item in the center on different screens	Try it	Flex
.align-self-baseline	Align a flex item at the baseline	Try it	Flex
.align-self-*-start	Align a flex item at the baseline on different screens	Try it	Flex
.align-self-stretch	Stretch a flex item	Try it	Flex
.align-self-*-stretch	Stretch a flex item on different screens	Try it	Flex
.badge	Creates a circular badge (grey circle - often used as a numerical indicator)	Try it	Badges
.badge-danger	Red badge. Indicates a dangerous or potentially negative action	Try it	Badges
.badge-dark	Dark badge. Dark grey alert box	Try it	Badges
.badge-info	Teal badge. Indicates a neutral informative change or action	Try it	Badges
.badge-light	Light badge. Light grey alert box	Try it	Badges
.badge-pill	Makes a badge more round	Try it	Badges
.badge-primary	Blue badge. Indicates an important action	Try it	Badges
.badge-secondary	Grey badge. Indicates a "less" important action	Try it	Badges

Class	Description	Example	Category
.badge-success	Green badge. Indicates a successful or positive action	Try it	Badges
.badge-warning	Yellow badge. Indicates caution should be taken with this action	Try it	Badges
.bg-danger	Adds a red background color to an element. Represents danger or a negative action	Try it	Colors
.bg-dark	Adds a dark grey background color to an element	Try it	Colors
.bg-info	Adds a teal background color to an element. Represents some information	Try it	Colors
.bg-light	Adds a light grey background color to an element	Try it	Colors
.bg-primary	Adds a blue background color to an element. Represents something important	Try it	Colors
.bg-secondary	Adds a grey background color to an element. Indicates a "less" important action	Try it	Colors
.bg-success	Adds a green background color to an element. Indicates success or a positive action	Try it	Colors
.bg-warning	Adds a yellow/orange background color to an element. Represents a warning or a negative action	Try it	Colors
.blockquote	Styles quoted blocks of content from another source (adds a larger font-size (1.25rem))	Try it	Typography
.blockquote-footer	Styles the source title inside the blockquote (light grey text with indentation)	Try it	Typography
.border	Adds a border to an element	Try it	Utilities
.border-bottom-0	Removes the bottom border from an element	Try it	Utilities

Class	Description	Example	Category
.border-danger	Adds a red border to an element (indicates danger)	Try it	Utilities
.border-dark	Adds a dark border to an element	Try it	Utilities
.border-info	Adds a teal border to an element (indicates information)	Try it	Utilities
.border-left-0	Removes the left border from an element	Try it	Utilities
.border-light	Adds a light grey border to an element	Try it	Utilities
.border-primary	Adds a blue border to an element	Try it	Utilities
.border-right-0	Removes the right border from an element	Try it	Utilities
.border-top-0	Removes the top border from an element	Try it	Utilities
.border-secondary	Adds a grey border to an element	Try it	Utilities
.border-success	Adds a green border to an element (indicates success)	Try it	Utilities
.border-warning	Adds a orange border to an element (indicates warning)	Try it	Utilities
.border-white	Adds a white border to an element	Try it	Utilities
.border-0	Removes all borders from an element	Try it	Utilities
.breadcrumb	A pagination. Indicates the current page's location within a navigational hierarchy	Try it	Pagination
.breadcrumb-item	Styles list items or links inside the breadcrumb	Try it	Pagination
.btn	Creates a basic button (gray background and rounded corners)	Try it	Buttons
.btn-block	Creates a block level button that spans the entire width of the parent element	Try it	Buttons

Class	Description	Example	Category
.btn-dark	Dark grey button	Try it	Buttons
.btn-danger	Red button. Indicates danger or a negative action	Try it	Buttons
.btn-group	Groups buttons together on a single line	Try it	Button Groups
.btn-group-lg	Large button group (makes all buttons in a button group larger - increased font-size and padding)	Try it	Button Groups
.btn-group-sm	Small button group (makes all buttons in a button group smaller)	Try it	Button Groups
.btn-group-vertical	Makes a button group appear vertically stacked	Try it	Button Groups
.btn-info	Teal button. Represents a neutral informative change or action	Try it	Buttons
.btn-light	Light grey button	Try it	Buttons
.btn-link	Makes a button look like a link (get button behavior)	Try it	Buttons
.btn-lg	Large button	Try it	Buttons
.btn-outline-dark	Dark grey bordered/outlined button	Try it	Buttons
.btn-outline-danger	Red bordered/outlined button. Indicates danger or a negative action	Try it	Buttons
.btn-outline-info	Teal bordered/outlined button. Represents a neutral informative change or action	Try it	Buttons
.btn-outline-light	Light grey bordered/outlined button	Try it	Buttons
.btn-outline-primary	Blue bordered/outlined button.	Try it	Buttons
.btn-outline-secondary	Grey bordered/outlined button. Indicates a "less" important action	Try it	Buttons

Class	Description	Example	Category
.btn-outline-success	Green bordered/outlined button. Indicates success or a positive action	Try it	Buttons
.btn-outline-warning	Orange bordered/outlined button. Represents warning or a negative action	Try it	Buttons
.btn-primary	Blue button. Indicates a something important	Try it	Buttons
.btn-sm	Small button	Try it	Buttons
.btn-secondary	Grey button. Indicates a "less" important action	Try it	Buttons
.btn-success	Green button. Indicates success or a positive action	Try it	Buttons
.btn-toolbar	Combine sets of button groups into button toolbars for more complex components	Try it	Button Groups
.btn-warning	Orange button. Represents warning or a negative action	Try it	Buttons
.card	Creates a card	Try it	Cards
.card-body	Container for card content	Try it	Cards
.card-columns	Container to create a masonry-like grid of cards	Try it	Cards
.card-danger	Adds a red background color to the card. Represents danger or a negative action	Try it	Cards
.card-dark	Adds a grey background color to the card	Try it	Cards
.card-deck	Container to create a grid of cards that are of equal height and width	Try it	Cards
.card-footer	Card footer	Try it	Cards
.card-group	Container to create a grid of cards that are of equal height and width, without side margins	Try it	Cards
.card-header	Card header	Try it	Cards

Class	Description	Example	Category
.card-header-tabs	Styles navigation tabs inside the card header	Try it	Cards
.card-header-pills	Styles navigation pills inside the card header	Try it	Cards
.card-img-bottom	Place the image at the bottom inside a card	Try it	Cards
.card-img-overlay	Turns an image into a card background. Often used to add text on top of the image	Try it	Cards
.card-img-top	Place the image at the top inside a card	Try it	Cards
.card-info	Adds a teal background color to the card. Represents some information	Try it	Cards
.card-light	Adds a light grey background color to the card	Try it	Cards
.card-link	Adds a blue color to any link and a hover effect inside the card	Try it	Cards
.card-primary	Adds a blue background color to the card. Represents something important	Try it	Cards
.card-secondary	Adds a grey background color to the card. Represents something "less" important	Try it	Cards
.card-subtitle	The <code>.card-subtitle</code> is used after a <code>.card-title</code> , and adds the following to an element: <code>margin-top: -.375rem; margin-bottom: 0;</code>	Try it	Cards
.card-success	Adds a green background color to the card. Indicates success or a positive action	Try it	Cards
.card-text	Used to remove bottom margins for a <code>p</code> element if it is the last child (or the only one), inside <code>.card-body</code>	Try it	Cards
.card-title	Adds a title to any heading element inside the card	Try it	Cards
.card-warning	Adds a yellow/orange background color to the card. Represents a warning or a negative action	Try it	Cards

Class	Description	Example	Category
.carousel	Creates a carousel (slideshow)	Try it	Carousel
.carousel-caption	Creates a caption text for each slide in the carousel	Try it	Carousel
.carousel-control-next	Container for "next" carousel/item link	Try it	Carousel
.carousel-control-next-icon	Used together with <code>.carousel-control-next</code> to create a "next" icon/button (right-pointed arrow)	Try it	Carousel
.carousel-control-prev	Container for "previous" carousel/item link	Try it	Carousel
.carousel-control-prev-icon	Used together with <code>.carousel-control-prev</code> to create a "previous" icon/button (left-pointed arrow)	Try it	Carousel
.carousel-indicators	Adds little dots/indicators at the bottom of each slide (which indicates how many slides there is in the carousel, and which slide the user are currently viewing)	Try it	Carousel
.carousel-inner	Container for slide items	Try it	Carousel
.carousel-item	Specifies the content of each slide	Try it	Carousel
.clearfix	Clears floats	Try it	Utilities
.close	Styles a close icon. This is often used for alerts and modals. Often used together with the × symbol to create the actual icon (a better looking "x"). It floats right by default	Try it	Utilities
.col-*	Creates a column layout for extra small devices (and up/all devices , if not combined with other column classes). The * can be a number between 1 and 12	Try it	Grid System
.col-sm-*	Creates a column layout for small devices (and up , if not combined with other column classes). The * can be a number between 1 and 12	Try it	Grid System

Class	Description	Example	Category
.col-md-*	Creates a column layout for medium devices (and up , if not combined with other column classes). The <code>*</code> can be a number between 1 and 12	Try it	Grid System
.col-lg-*	Creates a column layout for large devices (and up , if not combined with other column classes). The <code>*</code> can be a number between 1 and 12	Try it	Grid System
.col-xl-*	Creates a column layout for extra large devices. The <code>*</code> can be a number between 1 and 12	Try it	Grid System
.collapse	Indicates collapsible content - which can be hidden or shown on demand	Try it	Collapse
.collapse show	Show the collapsible content by default	Try it	Collapse
.container	Fixed width container with widths determined by screen sizes. Equal margin on the left and right.	Try it	Containers
.container-fluid	A container that spans the full width of the screen	Try it	Containers
.custom-checkbox	A wrapper/container for custom checkboxes	Try it	Custom Forms
.custom-control	A wrapper/container for custom forms	Try it	Custom Forms
.custom-control-input	Customized form control	Try it	Custom Forms
.custom-control-inline	Inline (horizontally - side by side) customized form controls	Try it	Custom Forms
.custom-control-label	Customized label, when used together with a custom form control	Try it	Custom Forms
.custom-file	Customized file upload	Try it	Custom Forms
.custom-radio	A wrapper/container for custom radio buttons	Try it	Custom Forms

Class	Description	Example	Category
.custom-range	Customized range control	Try it	Custom Forms
.custom-select	Customized select menu	Try it	Custom Forms
.custom-select-lg	Large customized select menu	Try it	Custom Forms
.custom-select-sm	Small customized select menu	Try it	Custom Forms
.disabled	Disables a button (adds opacity and a "no-parking-sign" icon on hover)	Try it	Buttons
.disabled	Disables a dropdown item (adds a grey text color and a "no-parking-sign" icon on hover)	Try it	Dropdowns
.disabled	Disables a pagination link (cannot be clicked - adds a grey text color and a "no-parking-sign" icon on hover)	Try it	Pagination
.disabled	Disables a list item in a list group (cannot be clicked - adds a light grey color and removes the hover effect on list item links)	Try it	List Groups
.dropdown	Creates a toggleable menu that allows the user to choose one value from a predefined list	Try it	Dropdowns
.dropdown-divider	Used to separate links in the dropdown menu with a thin horizontal border	Try it	Dropdowns
.dropdown-header	Used to add headers inside the dropdown menu	Try it	Dropdowns
.dropdown-item	Creates a dropdown item (added to links or buttons inside .dropdown-menu)	Try it	Dropdowns
.dropdown-item-text	Used to add plain text to a dropdown item, or used on links for default link styling	Try it	Dropdowns
.dropdown-menu	Adds the default styles for the dropdown menu container	Try it	Dropdowns

Class	Description	Example	Category
.dropdown-menu-right	Right-aligns a dropdown menu	Try it	Dropdowns
.dropdown-toggle	Used on the button that should hide and show (toggle) the dropdown menu	Try it	Dropdowns
.dropleft	Left-aligns the dropdown	Try it	Dropdowns
.dropright	Right-aligns the dropdown	Try it	Dropdowns
.dropdown	Indicates a dropdown menu (upwards instead of downwards)	Try it	Dropdowns
.d-block	Creates a block element (adds <code>display:block</code>)	Try it	Utilities
.d-*-block	Creates a block element on a specific screen width	Try it	Utilities
.d-flex	Creates a flexbox container and transforms direct children into flex items	Try it	Flex
.d-*-flex	Creates a flexbox container on a specific screen size	Try it	Flex
.d-inline-flex	Creates an inline flexbox container	Try it	Flex
.d-*inline-flex	Creates an inline flexbox container on a specific screen size	Try it	Flex
.embed-responsive	Container for embedded content. Makes videos or slideshows scale properly on any device	Try it	Images
.embed-responsive-16by9	Container for embedded content. Creates an 16:9 aspect ratio embedded content	Try it	Images
.embed-responsive-4by3	Container for embedded content. Creates an 4:3 aspect ratio embedded content	Try it	Images
.embed-responsive-item	Used inside <code>.embed-responsive</code> . Scales the video nicely to the parent element	Try it	Images
.fade	Adds a fading effect when closing an alert box	Try it	Alerts

Class	Description	Example	Category
.fade	Adds a fading effect when showing tab/pill content	Try it	Navs
.fade	Adds a fading effect when opening a modal	Try it	Modal
.fixed-bottom	Makes an element stay at the bottom of the screen (sticky/fixed)	Try it	Utilities
.fixed-top	Makes an element stay at the top of the screen (sticky/fixed)	Try it	Utilities
.flex-column	Display flex items vertically	Try it	Flex
.flex-*column	Display flex items vertically on different screen sizes:	Try it	Flex
.flex-column-reverse	Display flex items vertically, reversed	Try it	Flex
.flex-*column-reverse	Display flex items vertically, reversed, on different screen sizes	Try it	Flex
.flex-fill	Used on flex items to force it/them into equal width columns	Try it	Flex
.flex-*fill	Force flex items into equal widths on different screens	Try it	Flex
.flex-grow-0 1	Used on a single flex item to take up the rest of the available space	Try it	Flex
.flexnowrap	Don't wrap flex items	Try it	Flex
.flex-*nowrap	Don't wrap items on different screens	Try it	Flex
.flex-shrink-0 1	Used on a single flex item to shrink it if necessary		Flex
.flex-row	Display flex items horizontally (side by side)	Try it	Flex
.flex-*row	Display flex items horizontally on a specific screen size	Try it	Flex
.flex-row-reverse	Display flex items right-aligned and horizontally	Try it	Flex

Class	Description	Example	Category
.flex-*~row-reverse	Display flex items right-aligned and horizontally on a specific screen size	Try it	Flex
.flex-wrap	Wrap flex items	Try it	Flex
.flex-*~wrap	Wrap items on different screens	Try it	Flex
.flex-wrap~reverse	Wrap flex items, in reversed order	Try it	Flex
.flex-*~wrap~reverse	Wrap flex items, in reversed order on different screens	Try it	Flex
.float-left	Floats an element to the left	Try it	Utilities
.float-*~left	Floats an element to the left on different screens	Try it	Utilities
.float-none	Remove floats from an element	Try it	Utilities
.float-right	Floats an element to the right	Try it	Utilities
.float-*~right	Floats an element to the left on different screens	Try it	Utilities
.font-weight-bold	Bold text	Try it	Typography
.font-weight-italic	<i>Italic</i> text	Try it	Typography
.font-weight-light	Light weight text (font-weight:300)	Try it	Typography
.font-weight-normal	Normal text (font-weight:400)	Try it	Typography
.form-check	Container for checkboxes. Adds proper padding	Try it	Forms
.form-check~inline	Makes checkboxes appear on the same line (horizontally)	Try it	Forms
.form-check~input	Styles checkboxes with proper margins	Try it	Forms

Class	Description	Example	Category
.form-check-label	Ensures proper margins for labels used together with checkboxes	Try it	Forms
.form-control	Used on input, textarea, and select elements to span the entire width of the page and make them responsive	Try it	Forms
.form-control-file	Adds <code>display:block</code> and <code>width:100%</code> to input filed with type="file"	Try it	Forms
.form-control-lg	Large form control	Try it	Forms
.form-control-plaintext	Styles a form control as plain text	Try it	Forms
.form-control-range	Adds <code>display:block</code> and <code>width:100%</code> to input filed with type="range"	Try it	Forms
.form-control-sm	Small form control	Try it	Forms
.form-group	Container for form input and label	Try it	Forms
.form-inline	Makes a <form> left-aligned with inline-block controls (This only applies to forms within viewports that are at least 768px wide)	Try it	Forms
.h1 – .h6	Makes an element look like a heading of the chosen class (h1-h6)	Try it	Typography
.h-25	Sets the height of an element to 25%	Try it	Utilities
.h-50	Sets the height of an element to 50%	Try it	Utilities
.h-75	Sets the height of an element to 75%	Try it	Utilities
.h-100	Sets the height of an element to 100%	Try it	Utilities
.img-fluid	Responsive image (adds <code>max-width:100%</code> and <code>height:auto</code>)	Try it	Images
.img-thumbnail	Shapes an image to a thumbnail (thin light grey borders)	Try it	Images

Class	Description	Example	Category
.initialism	Displays the text inside an <code><abbr></code> element in a slightly smaller font size	Try it	Typography
.input-group	Container to enhance an input by adding an icon, text or a button in front or behind the input field as a "help text"	Try it	Input Group
.input-group-append	Input group container for adding help text behind an input field	Try it	Input Group
.input-group-lg	Large input group	Try it	Input Group
.input-group-prepend	Input group container for adding help text in front of an input field	Try it	Input Group
.input-group-sm	Small input group	Try it	Input Group
.input-group-text	Styles the specified help text in an input group	Try it	Input Group
.input-lg	Large input field	Try it	Input Sizing
.input-sm	Small input field	Try it	Input Sizing
.invisible	Make an element invisible	Try it	Utilities
.jumbotron	Creates a padded grey heading/box with rounded corners that enlarges the font sizes of the text inside it. Used for calling extra attention to some special content or information	Try it	Jumbotron
.jumbotron-fluid	Creates a full-width jumbotron (grey padded heading) without rounded borders	Try it	Jumbotron
.justify-content-*	Aligns flex items from the start , at the end , centered , in between and "around"	Try it	Flex
.justify-content-*=around	Aligns flex items "around" on different screen sizes	Try it	Flex
.justify-content-*=between	Aligns flex items in "between" on different screen sizes	Try it	Flex

Class	Description	Example	Category
.justify-content-center	Aligns flex items in the center on different screen sizes	Try it	Flex
.justify-content-end	Aligns flex items at the end on different screen sizes	Try it	Flex
.justify-content-start	Aligns flex items from the start on different screen sizes	Try it	Flex
.lead	Increase the font size and line height of a paragraph	Try it	Typography
.list-group	Creates a bordered list group for elements	Try it	List Group
.list-group-flush	Removes some borders and rounded corners from list items in a list group	Try it	List Group
.list-group-item	Added to each element in the list group	Try it	List Group
.list-group-item-action	Added to links inside the list group to make them stand out on hover (darker background)	Try it	List Group
.list-group-item-danger	Red background color for a list item in a list group	Try it	List Group
.list-group-item-dark	Dark grey background color for a list item in a list group	Try it	List Group
.list-group-item-info	Light-blue background color for a list item in a list group	Try it	List Group
.list-group-item-light	Light grey background color for a list item in a list group	Try it	List Group
.list-group-item-primary	Blue background color for a list item in a list group	Try it	List Group
.list-group-item-success	Green background color for a list item in a list group	Try it	List Group
.list-group-item-warning	Yellow background color for a list item in a list group	Try it	List Group

Class	Description	Example	Category
.list-inline	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)	Try it	Typography
.list-inline-item	Places all list items on a single line (used together with <code>.list-inline</code> on the parent <code></code> element)	Try it	Typography
.list-unstyled	Removes all default list-style (bullets, left margin, etc.) styling from a <code></code> or <code></code> list	Try it	Typography
.mark	Highlights text: <code>Highlighted text</code>	Try it	Typography
.media	Aligns media objects together with content (like images or videos - often used for comments in a blog post etc)	Try it	Media Objects
.media-body	Container for media content	Try it	Media Objects
.modal	Identifies the content as a modal and brings focus to it	Try it	Modals
.modal-body	Defines the style for the body of the modal. Add any HTML markup here (p, img, etc)	Try it	Modals
.modal-content	Styles the modal (border, background-color, etc). Inside this, add the modal's header, body and footer, if needed	Try it	Modals
.modal-dialog-centered	Centers the modal vertically and horizontally within the page	Try it	Modals
.modal-footer	The footer of the modal (often contains an action button and a close button)	Try it	Modals
.modal-header	The header of the modal (often contains a title and a close button)	Try it	Modals
.modal-lg	Large modal (wider than default)	Try it	Modals
.modal-sm	Small modal (less width)	Try it	Modals

Class	Description	Example	Category
.m-# / m-*#	Responsive margin classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.mt-# / mt-*#	Responsive top margin classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.mb-# / mb-*#	Responsive bottom margin classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.ml-# / ml-*#	Responsive left margin classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.mr-# / mr-*#	Responsive right margin classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.mx-# / mx-*#	Responsive left and right margin auto (horizontal) classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.my-# / my-*#	Responsive top and bottom margin auto (vertical) classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.mx-auto	Centers an element horizontally	Try it	Utilities
.nav nav-tabs	Creates a tabbed menu	Try it	Tabs
.nav nav-pills	Creates a pill menu	Try it	Tabs
.nav-justified	Justifies tab/pill links with an equal width	Try it	Tabs
.navbar	Creates a navigation bar	Try it	Navbar
.navbar-nav	Container for navigation links inside the .navbar container	Try it	Navbar
.navbar-brand	Added to a link or a header element inside the navbar to represent a logo or a header	Try it	Navbar

Class	Description	Example	Category
.navbar-collapse	Collapses the navbar (hidden and replaced with a menu/hamburger icon on mobile phones and small tablets)	Try it	Navbar
.navbar-expand-*	Responsive collapsible class - stacks the navbar vertically on small (sm), medium (md), large (lg) or extra large (xl) screens	Try it	Navbar
.navbar-dark	Adds a white text color to all links in the navbar	Try it	Navbar
.navbar-light	Adds a black text color to all links in the navbar	Try it	Navbar
.navbar-text	Vertically align any elements inside the navbar that are not links (ensures proper padding)	Try it	Navbar
.navbar-toggler	Styles the button that should open the navbar on small screens. Automatically styled as a hamburger/three bars	Try it	Navbar
.nav-link	Used to style links/anchors inside the navbar	Try it	Navbar
.nav-item	Used to style list items inside the navbar	Try it	Navbar
.no-gutters	Remove gutters/extra space from columns	Try it	Grid System
.page-item	Styles list items inside a pagination	Try it	Pagination
.page-link	Styles links inside a pagination	Try it	Pagination
.pagination	Creates a pagination (Useful when you have a web site with lots of pages)	Try it	Pagination
.pagination-lg	Large pagination (each pagination link gets a bigger font-size and more padding)	Try it	Pagination
.pagination-sm	Small pagination (each pagination link gets a smaller font size and less padding)	Try it	Pagination
.pre-scrollable	Makes a <code><pre></code> element scrollable (<code>max-height</code> of 350px and provide a y-axis scrollbar)	Try it	Helpers

Class	Description	Example	Category
.progress	Container for progress bars	Try it	Progress Bars
.progress-bar	Creates a progress bar	Try it	Progress Bars
.progress-bar-animated	Animates the progress bar (used together with stripes)	Try it	Progress Bars
.progress-bar-striped	Adds stripes to the progress bar	Try it	Progress Bars
.p-# / p-*-*#	Responsive padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.pt-# / pt-*-*#	Responsive top padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.pb-# / pb-*-*#	Responsive bottom padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.pl-# / pl-*-*#	Responsive left padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.pr-# / pr-*-*#	Responsive right padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.py-# / py-*-*#	Responsive top and bottom padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.px-# / px-*-*#	Responsive left and right padding classes. * can be sm, md, lg or xl. # can be a number between 0 and 5	Try it	Utilities
.rounded	Adds rounded corners to an element	Try it	Utilities
.rounded-bottom	Adds bottom rounded corners to an element	Try it	Utilities
.rounded-circle	Shapes an element to a circle (not supported in IE8 and earlier)	Try it	Utilities

Class	Description	Example	Category
.rounded-left	Adds left rounded corners of an element	Try it	Utilities
.rounded-right	Adds right rounded corners to an element	Try it	Utilities
.rounded-top	Adds top rounded corners to an element	Try it	Utilities
.rounded-0	Removes rounded corners from an element	Try it	Utilities
.row	Container for responsive columns	Try it	Grid System
.shadow	Adds a shadow to an element	Try it	Utilities
.shadow-lg	Adds a large shadow to an element	Try it	Utilities
.shadow-md	Adds a medium shadow to an element	Try it	Utilities
.shadow-none	Removes shadows from an element	Try it	Utilities
.shadow-sm	Adds a small shadow to an element	Try it	Utilities
.small	Creates a lighter, secondary text in any heading	Try it	Typography
.sr-only	Hides an element on all devices except for screen readers	Try it	Utilities
.sr-only-focusable	Hides an element on all devices except for screen readers	Try it	Utilities
.sticky-top	Makes an element stay sticky/fixed at the top of the page when you scroll past it	Try it	Utilities
.tab-content	Used together with <code>.tab-pane</code> to creates toggleable/dynamic tabs/pills	Try it	Tabs
.tab-pane	Used together with <code>.tab-content</code> to creates toggleable/dynamic tabs/pills	Try it	Tabs
.table	Adds basic styling to a table (padding, bottom borders, etc)	Try it	Tables

Class	Description	Example	Category
.table-active	Adds a grey background color to the table row (<code><tr></code> or table cell (<code><td></code>) (same color used on hover)	Try it	Tables
.table-bordered	Adds borders on all sides of the table and cells	Try it	Tables
.table-borderless	Remove borders from a table	Try it	Tables
.table-condensed	Makes a table more compact by cutting cell padding in half	Try it	Tables
.table-dark	Adds a black background with white text to the table	Try it	Tables
.table-hover	Creates a hoverable table (adds a grey background color on table rows on hover)	Try it	Tables
* .table-responsive-	Makes a table responsive (adds a horizontal scrollbar when needed). By default, the scrollbar is added to the table on screens that are less than 992px wide (if needed). There's no difference when viewing anything larger than 992px wide. However, you can use sm md lg xl to decide WHEN the table should get a scrollbar, depending on the screen width	Try it	Tables
.table-striped	Adds zebra-stripes to a table	Try it	Tables
.text-capitalize	Indicates capitalized text	Try it	Typography
.text-center	Center-aligns text	Try it	Typography
.text-*center	Center-aligns text on different screens	Try it	Typography
.text-danger	Red text color. Indicates danger	Try it	Colors
.text-dark	Dark grey text color	Try it	Typography
.text-hide	Hides text (helps replace an element's text content with a background image)	Try it	Typography

Class	Description	Example	Category
.text-info	Light-blue text color. Indicates information	Try it	Colors
.text-light	Light grey text color	Try it	Colors
.text-justify	Indicates justified text	Try it	Typography
.text-left	Aligns the text to the left	Try it	Typography
.text-*-left	Left-aligns text on different screens	Try it	Typography
.text-lowercase	Changes text to lowercase	Try it	Typography
.text-muted	Grey text color	Try it	Colors
.text-nowrap	Prevents the text from wrapping	Try it	Typography
.text-primary	Blue text color. Indicates something important	Try it	Colors
.text-secondary	Grey text color. Indicates something "less" important	Try it	Colors
.text-right	Aligns text to the right	Try it	Typography
.text-*-right	Right-aligns text on different screens	Try it	Typography
.text-success	Green text color. Indicates success	Try it	Colors
.text-uppercase	Makes text uppercase	Try it	Typography
.text-warning	Yellow/orange text color. Indicates warning	Try it	Colors
.text-white	White text color	Try it	Colors
.thead-dark	Adds a black background color to table headers	Try it	Tables
.thead-light	Adds a grey background color to table headers	Try it	Tables

Class	Description	Example	Category
.visible	Make an element visible	Try it	Utilities
.w-25	Sets the width of an element to 25%	Try it	Utilities
.w-50	Sets the width of an element to 50%	Try it	Utilities
.w-75	Sets the width of an element to 75%	Try it	Utilities
.w-100	Sets the width of an element to 100%	Try it	Utilities

Bootstrap 4 JS Alert

Alert CSS Classes

For a tutorial about Alerts, read our [Bootstrap Alerts Tutorial](#).

Class	Description	Example
.alert	Creates an alert message box	Try it
.alert-danger	Red alert. Indicates a dangerous or potentially negative action	Try it
.alert-dark	Dark alert. Dark grey alert box	Try it
.alert-dismissible	Indicates a closable alert box. Together with the <code>.close</code> class, this class is used to close the alert (adds extra padding)	Try it
.alert-heading	Adds <code>color: inherit</code> to the specified element	Try it
.alert-info	Light-blue alert. Indicates a neutral informative change or action	Try it
.alert-light	Light alert. Light grey alert box	Try it
.alert-link	Used on links inside alerts to provide matching colored links	Try it
.alert-primary	Blue alert. Indicates an important action	Try it
.alert-secondary	Grey alert. Indicates a "less" important action	Try it
.alert-success	Green alert. Indicates a successful or positive action	Try it
.alert-warning	Yellow alert. Indicates caution should be taken with this action	Try it
.close	Styles the close button for the alert message (floats right with a specified font-size, color, etc.)	Try it

Close Alerts Via data-* Attributes

Add `data-dismiss="alert"` to a link or a button element to close the alert message.

Example

```
<a href="#" class="close" data-dismiss="alert">&times;</a>
```

Close Alerts Via JavaScript

Close manually with:

Example

```
$('.close').alert("close");
```

Alert Options

None

Alert Methods

The following table lists all available alert methods.

Method	Description	Try it
<code>.alert("close")</code>	Closes the alert message	Try it
<code>.alert("dispose")</code>	Destroys an element's alert.	

Alert Events

The following table lists all available alert events.

Event	Description	Try it
<code>close.bs.alert</code>	Occurs when the alert message is about to be closed	Try it
<code>closed.bs.alert</code>	Occurs when the alert message has been closed (will wait for CSS transitions to complete)	Try it

Bootstrap 4 JS Button

For a tutorial about Buttons, read our [Bootstrap Buttons Tutorial](#).

The classes below can be used to style any <a>, <button>, or <input> element:

Class	Description	Example
.btn	Adds basic styling to any button	Try it
.btn-block	Makes a block-level button (spans the full width of the parent element)	Try it
.btn-danger	Indicates a dangerous or potentially negative action	Try it
.btn-dark	Dark grey button	Try it
.btn-default	Indicates a default/standard button	Try it
.btn-info	Contextual button for informational alert messages	Try it
.btn-lg	Makes a large button	Try it
.btn-light	Light grey button	Try it
.btn-link	Makes a button look like a link (will still have button behavior)	Try it
.btn-outline-*	Creates an outlined/bordered button. Use any of the contextual classes as * (btn-outline-primary, btn-outline-success, etc)	Try it
.btn-primary	Provides extra visual weight and identifies the primary action in a set of buttons	Try it
.btn-sm	Makes a small button	Try it
.btn-success	Indicates a successful or positive action	Try it
.btn-secondary	Indicates a "less" important action	Try it
.btn-toolbar	Combine sets of button groups into button toolbars for more complex components	Try it
.btn-warning	Indicates caution should be taken with this action	Try it
.active	Makes the button appear pressed	Try it
.disabled	Makes the button disabled	Try it

Via JavaScript

Enable manually with:

```
$('.btn').button();
```

Button Options

None

Button Methods

The following table lists all available button methods.

Note: Methods can also be passed via data attributes; append the method name to data-, as in data-toggle or data-dispose.

Method	Description	Try it
.button("toggle")	Makes the button look pressed	Try it
.button("dispose")	Destroys an element's button	

Bootstrap 4 JS Carousel

Carousel CSS Classes

For a tutorial about Carousels, read our [Bootstrap Carousel Tutorial](#).

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

[Try it Yourself »](#)

Via data-* Attributes

The `data-ride="carousel"` attribute activates the carousel.

The `data-slide` and `data-slide-to` attributes specifies which slide to go to.

The `data-slide` attribute accepts two values: `prev` or `next`, while `data-slide-to` accept numbers.

Example

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

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

    <!-- Carousel Controls -->
    <a class="carousel-control-prev" href="#myCarousel" data-slide="prev">
        <span class="carousel-control-prev-icon"></span>
    </a>
```

Via JavaScript

Enable manually with:

Example

```
// Activate Carousel
$("#myCarousel").carousel();

// Enable Carousel Indicators
$(".item").click(function(){
    $("#myCarousel").carousel(1);
});

// Enable Carousel Controls
$(".carousel-control-prev").click(function(){
    $("#myCarousel").carousel("prev");
});
```

Carousel Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to data-, as in data-interval="".

Name	Type	Default	Description	Try it
interval	number, or the boolean	5000 false	Specifies the delay (in milliseconds) between each slide. Note: Set interval to false to stop the items from automatically sliding	Using JS Using data
keyboard	boolean	true	Specifies whether the carousel should react to keyboard events: <ul style="list-style-type: none">• true - the carousel can be navigated (next and previous) with the keyboard left and right arrows• false - the carousel can NOT be navigated with the keyboard left and right arrows	Using JS Using data
pause	string, or the boolean	"hover" false	Pauses the carousel from going through the next slide when the mouse pointer enters the carousel, and resumes the sliding when the mouse pointer leaves the carousel. Note: Set pause to false to stop the ability to pause on hover	Using JS Using data
wrap	boolean	true	Specifies whether the carousel should go through all slides continuously, or stop at the last slide <ul style="list-style-type: none">• true - cycle continuously• false - stop at the last item	Using JS Using data

Carousel Methods

The following table lists all available carousel methods.

Method	Description	Try it
.carousel(<i>options</i>)	Activates the carousel with an option. See options above for valid values	Try it
.carousel("cycle")	Goes through the carousel items from left to right	Try it
.carousel("pause")	Stops the carousel from going through items	Try it
.carousel(<i>number</i>)	Goes to a specified item (zero-based: first item is 0, second item is 1, etc..)	Try it
.carousel("prev")	Goes to the previous item	Try it
.carousel("next")	Goes to the next item	Try it
.carousel("dispose")	Destroys a carousel	

Carousel Events

The following table lists all available carousel events.

Event	Description	Try it
slide.bs.carousel	Occurs when the carousel is about to slide from one item to another	Try it
slid.bs.carousel	Occurs when the carousel has finished sliding from one item to another	Try it

The slid and slide events also has additional properties:

Property	Description	Try it
direction	Returns the direction the carousel is sliding (left or right)	
relatedTarget	Returns the DOM element that is being slid into place as the active item	Try it
from	Returns the index of where the previous item came from, when moving on to the next one	Try it
to	Returns the index of the next item	Try it

Bootstrap 4 JS Collapse

Collapse CSS Classes

For a tutorial about Collapsibles, read our [Bootstrap Collapse Tutorial](#).

Class	Description	Example
.collapse	Hides the content	Try it
.collapse show	Shows the collapsible content by default	Try it
.collapsing	Added when the transition starts, and removed when it finishes	Try it

Via data-* Attributes

Just add `data-toggle="collapse"` and a `data-target` to element to automatically assign control of a collapsible element. The data-target attribute accepts a CSS selector to apply the collapse to. Be sure to add the class collapse to the collapsible element. If you'd like it to default open, add the additional class "show".

Example

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

<div id="demo" class="collapse">
Some text..
</div>
```

Tip: To add accordion-like group management to a collapsible control, add the data attribute `data-parent="#selector"`.

Via JavaScript

Enable manually with:

```
$('.collapse').collapse()
```

Collapse Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to data-, as in data-parent="".

Name	Type	Default	Description	Try it
parent	selector	false	All collapsible elements under the specified parent will be closed when this collapsible item is shown. (similar to traditional accordion behavior)	Try it
toggle	boolean	true	Toggles the collapsible element on invocation	Try it

Collapse Methods

The following table lists all available collapse methods.

Method	Description	Try it
.collapse(<i>options</i>)	Activates the collapsible element with an option. See options above for valid values	
.collapse("toggle")	Toggles the collapsible element	Try it
.collapse("show")	Shows the collapsible element	Try it
.collapse("hide")	Hides the collapsible element	Try it
.collapse("dispose")	Destroys the collapsible element	

Collapse Events

The following table lists all available collapse events.

Event	Description	Try it
show.bs.collapse	Occurs when the collapsible element is about to be shown	Try it
shown.bs.collapse	Occurs when the collapsible element is fully shown (after CSS transitions have completed)	Try it
hide.bs.collapse	Occurs when the collapsible element is about to be hidden	Try it
hidden.bs.collapse	Occurs when the collapsible element is fully hidden (after CSS transitions have completed)	Try it

Bootstrap JS Dropdown

Dropdown CSS Classes

For a tutorial about Dropdowns, read our [Bootstrap Dropdowns Tutorial](#).

Class	Description	Example
.dropdown	Indicates a dropdown menu	Try it
.dropdown-item	Style links inside the dropdown menu with proper padding etc	Try it
.dropdown-item-text	Style text or text links inside the dropdown menu with proper padding etc	Try it
.dropdown-menu	Builds the dropdown menu	Try it
.dropdown-menu-right	Right-aligns a dropdown menu	Try it
.dropdown-header	Adds a header inside the dropdown menu	Try it
.dropup	Indicates a dropup menu	Try it
.disabled	Disables an item in the dropdown menu	Try it
.active	Styles the active element in a dropdown menu	Try it
.divider	Separes items inside the dropdown menu with a horizontal line	Try it

Via data-* Attributes

Add `data-toggle="dropdown"` to a link or a button to toggle a dropdown menu.

Example

```
<button type="button" class="dropdown-toggle" data-
toggle="dropdown">Dropdown Example</button>
```

Via JavaScript

Enable manually with:

Example

```
$('.dropdown-toggle').dropdown();
```

Note: The data-toggle="dropdown" attribute is required regardless of whether you call the dropdown() method.

Dropdown Options

None

Dropdown Methods

The following table lists all available dropdown methods.

Method	Description	Try it
.dropdown("toggle")	Toggles the dropdown. If set, it will open the dropdown menu by default	Try it
.dropdown("update")	Updates the position of an element's dropdown	
.dropdown("dispose")	Destroys an element's dropdown	

Dropdown Events

The following table lists all available dropdown events.

Event	Description	Try it
show.bs.dropdown	Occurs when the dropdown is about to be shown.	Try it
shown.bs.dropdown	Occurs when the dropdown is fully shown (after CSS transitions have completed)	Try it
hide.bs.dropdown	Occurs when the dropdown is about to be hidden	Try it
hidden.bs.dropdown	Occurs when the dropdown is fully hidden (after CSS transitions have completed)	Try it

Tip: Use jQuery's `event.relatedTarget` to get the element which triggered the dropdown:

Example

```
$(".dropdown").on("show.bs.dropdown", function(event){  
    var x = $(event.relatedTarget).text(); // Get the text of the element  
    alert(x);  
});
```

Bootstrap JS Modal

Modal CSS Classes

For a tutorial about Modals, read our [Bootstrap Modal Tutorial](#).

Class	Description	Example
.modal	Creates a modal	Try it
.modal-content	Styles the modal properly with border, background-color, etc. Use this class to add the modal's header, body, and footer	Try it
.modal-dialog-centered	Centers the modal vertically and horizontally within the page	Try it
.modal-header	Defines the style for the header of the modal	Try it
.modal-body	Defines the style for the body of the modal	Try it
.modal-footer	Defines the style for the footer in the modal. Note: This area is right-aligned by default. To change this, add the justify-content-start or justify-content-center together with the .modal-footer class	Try it
.modal-sm	Specifies a small modal	Try it
.modal-lg	Specifies a large modal	Try it
.fade	Adds an animation/transition effect which fades the modal in and out	Try it

Trigger the Modal Via data-* Attributes

Add `data-toggle="modal"` and `data-target="#modalID"` to any element.

Note: For `<a>` elements, omit `data-target`, and use `href="#modalID"` instead:

Example

```
<!-- Buttons -->
<button type="button" data-toggle="modal" data-target="#myModal">Open
Modal</button>

<!-- Links -->
<a data-toggle="modal" href="#myModal">Open Modal</a>

<!-- Other elements -->
<p data-toggle="modal" data-target="#myModal">Open Modal</p>
```

Trigger Via JavaScript

Enable manually with:

Example

```
$("#myModal").modal()
```

Modal Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to data-, as in data-backdrop="".

Name	Type	Default	Description	Try it
backdrop	boolean or the string "static"	true	<p>Specifies whether the modal should have a dark overlay:</p> <ul style="list-style-type: none">• true - dark overlay• false - no overlay (transparent) <p>If you specify the value "static", it is not possible to close the modal when clicking outside of it</p>	Using JS Using data
keyboard	boolean	true	Specifies whether the modal can be closed with the escape key (Esc):	Using JS Using data
show	boolean	true	Specifies whether to show the modal when initialized	Using JS Using data

Modal Methods

The following table lists all available modal methods.

Method	Description	Try it
.modal(options)	Activates the content as a modal. See options above for valid values	Try it
.modal("toggle")	Toggles the modal	Try it
.modal("show")	Opens the modal	Try it
.modal("hide")	Hides the modal	Try it

Modal Events

The following table lists all available modal events.

Event	Description	Try it
show.bs.modal	Occurs when the modal is about to be shown	Try it
shown.bs.modal	Occurs when the modal is fully shown (after CSS transitions have completed)	Try it
hide.bs.modal	Occurs when the modal is about to be hidden	Try it
hidden.bs.modal	Occurs when the modal is fully hidden (after CSS transitions have completed)	Try it

Bootstrap JS Popover

JS Popover

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

For a tutorial about Popovers, read our [Bootstrap Popover Tutorial](#).

Enable via data-* Attributes

The `data-toggle="popover"` activates the popover.

The `title` attribute specifies the header text of the popover.

The `data-content` attribute specifies the text that should be displayed inside the popover's body.

Example

```
<a href="#" data-toggle="popover" title="Popover Header" data-content="Some content inside the popover">Toggle popover</a>
```

Via JavaScript

Popovers are not CSS-only plugins, and must therefore be initialized with jQuery: select the specified element and call the `popover()` method.

Example

```
// Select all elements with data-toggle="popover" in the document
 $('[data-toggle="popover"]').popover();

// Select a specified element
 $('#myPopover').popover();
```

Popover Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-placement=""`.

Name	Type	Default	Description	Try it
------	------	---------	-------------	--------

animation	boolean	true	Specifies whether to add a CSS fade transition effect when opening and closing the popover <ul style="list-style-type: none"> • true - Add a fading effect • false - Do not add a fading effect 	Try it
container	string, or the boolean	false	Appends the popover to a specific element. Example: container: 'body'	Try it
content	string	""	Specifies the text inside the popover's body	Try it
delay	number, or object	0	Specifies the number of milliseconds it will take to open and close the popover. To specify a delay for opening and another one for closing, use the object structure: delay: {show: 500, hide: 100} - which will take 500 ms to open the popover, but only 100 ms to close it	Try it
html	boolean	false	Specifies whether to accept HTML tags in the popover: <ul style="list-style-type: none"> • true - Accept HTML tags • false - Do not accept HTML tags <p>Note: The HTML must be inserted in the title attribute (or using the title option). When set to false (default), jQuery's <code>text()</code> method will be used. Use this if you are worried about XSS attacks</p>	Try it

placement	string	"right"	Specifies the popover position. Possible values: <ul style="list-style-type: none">• "top" - Popover on top• "bottom" - Popover on bottom• "left" - Popover on left• "right" - Popover on right• "auto" - Lets the browser decide the position of the popover. For example, if the value is "auto left", the popover will display on the left side when possible, otherwise on the right. If the value is "auto bottom", the popover will display at the bottom when possible, otherwise on the top	Try it
selector	string, or the boolean false	false	Adds the popover to a specified selector	Try it
template	string		Base HTML to use when creating the popover. The popover's title will be injected into the .popover-header. The popover's content will be injected into the .popover-body..arrow will become the popover's arrow. The outermost wrapper element should have the .popover class.	
title	string	""	Specifies the header text of the popover	Try it

trigger	string	"click"	Specifies how the popover is triggered. Possible values: <ul style="list-style-type: none"> • "click" - Trigger the popover with a click • "hover" - Trigger the popover on hover • "focus" - Trigger the popover when it gets focus (by tabbing or clicking .e.g) • "manual" - Trigger the popover manually 	Try it
offset	number or a string	0	Offset of the popover relative to its target	
fallbackPlacement	string or an array	"flip"	Specifies which position Popper will use on fallback	
boundary	string or element	"scrollParent"	Overflow constraint boundary of the popover. Accepts the values "viewport", "window" or "scrollParent", or an HTML element	

Popover Methods

The following table lists all available popover methods.

Method	Description	Try it
.popover(<i>options</i>)	Activates the popover with an option. See options above for valid values	Try it
.popover("show")	Shows the popover	Try it
.popover("hide")	Hides the popover	Try it
.popover("toggle")	Toggles the popover	Try it
.popover("dispose")	Hides and destroys the popover	Try it
.popover("enable")	Enables the popover the ability to be shown. This is default	
.popover("disable")	Removes the ability to show a popover. The popover can only be shown if it is re-enabled again	
.popover("toggleEnabled")	Toggles the ability for the popover to be shown or hidden	
.popover("update")	Updates the position of the popover	

Popover Events

The following table lists all available popover events.

Event	Description	Try it
show.bs.popover	Occurs when the popover is about to be shown	Try it
shown.bs.popover	Occurs when the popover is fully shown (after CSS transitions have completed)	Try it
hide.bs.popover	Occurs when the popover is about to be hidden	Try it
hidden.bs.popover	Occurs when the popover is fully hidden (after CSS transitions have completed)	Try it
inserted.bs.popover	Occurs after the show.bs.popover event when the popover template has been added to the DOM	

Bootstrap JS Scrollspy

JS Scrollspy

Scrollspy is used to automatically update links in a navigation list based on scroll position. For a tutorial about Scrollspy, read our [Bootstrap Scrollspy Tutorial](#).

Via data-* Attributes

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

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

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

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

Requires relative positioning: The element with `data-spy="scroll"` requires the CSS **position** property, with a value of "relative" to work properly.

Example

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

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

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

Via JavaScript

Enable manually with:

Example

```
$( 'body' ).scrollspy({target: ".navbar"})
```

Scrollspy Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to data-, as in data-offset="".

Name	Type	Default	Description	Try it
offset	number	10	Specifies the number of pixels to offset from top when calculating the position of scroll	Try it

Scrollspy Methods

The following table lists all available scrollspy methods.

Method	Description
.scrollspy("refresh")	When adding and removing elements from the scrollspy, this method can be used to refresh the document
.scrollspy("dispose")	Destroys the scrollspy

Scrollspy Events

The following table lists all available scrollspy events.

Event	Description
activate.bs.scrollspy	Occurs when a new item becomes activated by the scrollspy

Bootstrap JS Tab

Tab CSS Classes

Tabs are used to separate content into different panes where each pane is viewable one at a time.

For a tutorial about Tabs, read our [Bootstrap Tabs/Pills Tutorial](#).

Class	Description	Example
.nav .nav-tabs	Creates navigation tabs	Try it
.nav .nav-pills	Creates navigation pills	Try it
.nav-item	Creates tab items	Try it
.nav-link	Styles links inside the navigation tab	Try it
.nav-justified	Makes navigation tabs/pills equal widths of their parent, at screens wider than 768px. On smaller screens, the nav tabs are stacked	Try it
.tab-content	Together with .tab-pane and data-toggle="tab", it makes the tab toggleable	Try it
.tab-pane	Together with .tab-content and data-toggle="tab", it makes the tab toggleable	Try it

Via data-* Attributes

Add `data-toggle="tab"` to each tab, and add a `.tab-pane` class with a unique ID for every tab and wrap them in a `.tab-content` class.

Example

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

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

Via JavaScript

Enable manually with:

Example

```
// Select all tabs
$('.nav-tabs a').click(function(){
    $(this).tab('show');
})

// Select tab by name
$('.nav-tabs a[href="#home"]').tab('show')

// Select first tab
$('.nav-tabs a:first').tab('show')

// Select last tab
$('.nav-tabs a:last').tab('show')

// Select fourth tab (zero-based)
$('.nav-tabs li:eq(3) a').tab('show')
```

Tab Options

None

Tab Methods

The following table lists all available tab methods.

Method	Description	Try it
.tab("show")	Shows the tab	Try it

Tab Events

The following table lists all available tab events.

Event	Description	Try it
show.bs.tab	Occurs when the tab is about to be shown.	Try it
shown.bs.tab	Occurs when the tab is fully shown (after CSS transitions have completed)	Try it
hide.bs.tab	Occurs when the tab is about to be hidden	Try it
hidden.bs.tab	Occurs when the tab is fully hidden (after CSS transitions have completed)	Try it

Tip: Use jQuery's `event.target` and `event.relatedTarget` to get the active tab and the previous active tab:

Example

```
$('.nav-tabs a').on('shown.bs.tab', function(event){
    var x = $(event.target).text();           // active tab
    var y = $(event.relatedTarget).text();    // previous tab
});
```

Bootstrap JS Tooltip

JS Tooltip

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

For a tutorial about Tooltips, read our [Bootstrap Tooltip Tutorial](#).

Via data-* Attributes

The `data-toggle="tooltip"` activates the tooltip.

The `title` attribute specifies the text that should be displayed inside the tooltip.

Example

```
<a href="#" data-toggle="tooltip" title="Hooray!">Hover over me</a>
```

Via JavaScript

Tooltips are not CSS-only plugins, and must therefore be initialized with jQuery: select the specified element and call the `tooltip()` method.

Example

```
// Select all elements with data-toggle="tooltips" in the document
 $('[data-toggle="tooltip"]').tooltip();

// Select a specified element
 $('#myTooltip').tooltip();
```

Tooltip Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-placement=""`.

Name	Type	Default	Description	Try it
------	------	---------	-------------	--------

animation	boolean	true	Specifies whether to add a CSS fade transition effect when showing and hiding the tooltip <ul style="list-style-type: none"> • true - Add a fading effect • false - Do not add a fading effect 	Try it
container	string, or the boolean	false	Appends the tooltip to a specific element. Example: container: 'body'	Try it
delay	number, or object	0	Specifies the number of milliseconds it will take to show and hide the tooltip. To specify a delay for showing and another one for hiding, use the object structure:delay: {show: 500, hide: 100} - which will take 500 ms to show the tooltip, but only 100 ms to hide it	Try it
html	boolean	false	Specifies whether to accept HTML tags in the tooltip: <ul style="list-style-type: none"> • true - Accept HTML tags • false - Do not accept HTML tags <p>Note: The HTML must be inserted in the title attribute (or using the title option). When set to false (default), jQuery's text() method will be used. Use this if you are worried about XSS attacks</p>	Try it

placement	string	"top"	Specifies the tooltip position. Possible values: <ul style="list-style-type: none">• "top" - Tooltip on top• "bottom" - Tooltip on bottom• "left" - Tooltip on left• "right" - Tooltip on right• "auto" - Lets the browser decide the position of the tooltip. For example, if the value is "auto left", the tooltip will display on the left side when possible, otherwise on the right. If the value is "auto bottom", the tooltip will display at the bottom when possible, otherwise on the top	Try it
selector	string, or the boolean false		Adds the tooltip to a specified selector	Try it
template	string		Base HTML to use when creating the tooltip. The tooltip's title will be inserted into the element having the class .tooltip-inner and the element with the class .tooltip-arrow will become the tooltip's arrow. The outermost wrapper element should have the .tooltip class.	
title	string	""	Specifies the text that should be displayed inside the tooltip	Try it

trigger	string	"hover focus"	Specifies how the tooltip is triggered. Possible values: <ul style="list-style-type: none">• "click" - Trigger the tooltip with a click• "hover" - Trigger the tooltip on hover• "focus" - Trigger the tooltip when it gets focus (by tabbing or clicking .e.g)• "manual" - Trigger the tooltip manually	Try it
offset	number or a string	0	Offset of the tooltip relative to its target	
fallbackPlacement	string or an array	"flip"	Specifies which position Popper will use on fallback	
boundary	string or element	"scrollParent"	Overflow constraint boundary of the tooltip. Accepts the values "viewport", "window" or "scrollParent", or an HTML element	

Tooltip Methods

The following table lists all available tooltip methods.

Method	Description	Try it
.tooltip(<i>options</i>)	Activates the tooltip with an option. See options above for valid values	Try it
.tooltip("show")	Shows the tooltip	Try it
.tooltip("hide")	Hides the tooltip	Try it
.tooltip("toggle")	Toggles the tooltip	Try it
.tooltip("dispose")	Hides and destroys the tooltip	Try it

Tooltip Events

The following table lists all available tooltip events.

Event	Description	Try it
show.bs.tooltip	Occurs when the tooltip is about to be shown	Try it
shown.bs.tooltip	Occurs when the tooltip is fully shown (after CSS transitions have completed)	Try it
hide.bs.tooltip	Occurs when the tooltip is about to be hidden	Try it
hidden.bs.tooltip	Occurs when the tooltip is fully hidden (after CSS transitions have completed)	Try it