

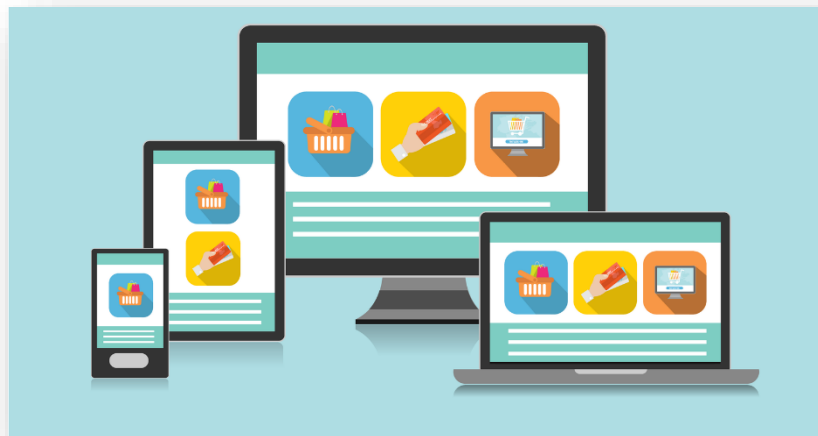
Web Design

CSS Framework:

BOOTSTRAP 5



**Build fast, responsive sites
with Bootstrap**



Bootstrap

Bootstrap is the most popular and powerful front-end (HTML, CSS, and JavaScript) framework for faster and easier responsive web development.

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for creating common user interface components like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and so on.

Bootstrap gives you ability to create flexible and responsive web layouts with much less efforts.

Bootstrap was originally created by a designer and a developer at Twitter in mid-2010. Before being an open-sourced framework, Bootstrap was known as Twitter Blueprint.

What You Can Do with Bootstrap

There are lot more things you can do with Bootstrap.

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

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

Advantages of Using Bootstrap

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

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

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

Creating Your First Web Page with Bootstrap

Now we're going to create a basic Bootstrap template by including the Bootstrap CSS and JS files via CDN. Bootstrap requires a third-party library [Popper.js](#) for some of its components like popovers and tooltips. You can either include it separately or simply include Bootstrap JS bundled with Popper.

We recommend adding Bootstrap in your project via CDN (Content Delivery Network) because CDN offers performance benefit by reducing the loading time, since they are hosting the files on multiple servers spread across the globe so that when a user requests the file it will be served from the server nearest to them. We're also using the CDN links in our examples:

Step 1: Creating a Basic HTML file

Open up your favorite code editor and create a new HTML file. Start with an empty window and type the following code and save it as "basic.html" on your desktop.

Example	Try this code »
1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="utf-8"></code>
5	<code> <meta name="viewport" content="width=device-width, initial-scale=1"></code>
6	<code> <title>Basic HTML File</title></code>
7	<code></head></code>
8	<code><body></code>
9	<code> <h1>Hello, world!</h1></code>
10	<code></body></code>
11	<code></html></code>

Tip: Always include the viewport `<meta>` tag inside the `<head>` section of your document to enable touch zooming and ensure proper rendering on mobile devices.

Step 2: Making this HTML File a Bootstrap Template

In order to make this plain HTML file a Bootstrap template, just include the Bootstrap CSS and JS files using their CDN links. Also, you should include JavaScript files at the bottom of the page, right before the closing `</body>` tag to improve the performance of your web pages.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>Basic Bootstrap Template</title>
  <! -- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body>
  <h1>Hello, world! </h1>
  <! -- Bootstrap JS Bundle with Popper -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bu
ndle.min.js" integrity="sha384-
MrCW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

</body>
</html>

```

And we're all set! After adding the Bootstrap's CSS and JS files to our HTML page, we can begin to develop any responsive site or application with the Bootstrap framework.

Note: The biggest change in Bootstrap 5 is it doesn't require jQuery anymore. However, you can still use jQuery to quickly implement Bootstrap component's methods and options. If Bootstrap detects jQuery in the window object it'll add all of its components in jQuery's plugin system. You will learn about them in advanced section of this tutorial series.

Bootstrap Containers

Creating Containers with Bootstrap

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

Bootstrap provides three different types containers:

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

The table below illustrates how each container's max-width changes across each breakpoint.

For example, when using the `.container` class the actual width of the container will be 100% if the viewport width is $<576\text{px}$, 540px if the viewport width is $\geq 576\text{px}$ but $<768\text{px}$, 720px if the viewport width is $\geq 768\text{px}$ but $<992\text{px}$, 960px if the viewport width is $\geq 992\text{px}$ but $<1200\text{px}$, 1140px if the viewport width is $\geq 1200\text{px}$ but $<1400\text{px}$, and 1320px if the viewport width is $\geq 1400\text{px}$.

Similarly, when you use the `.container-lg` class the actual width of the container will be 100% until the viewport width is $<992\text{px}$, 960px if the viewport width is $\geq 992\text{px}$ but $<1200\text{px}$, 1140px if the viewport width $\geq 1200\text{px}$ but $<1400\text{px}$, and 1320px if the viewport width is $\geq 1400\text{px}$.

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

Creating Responsive Fixed-width Containers

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

```
<div class="container">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

Creating Fluid Containers

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

```
<div class="container-fluid">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

Specify Responsive Breakpoints for Containers

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

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

```
<div class="container-sm">100% wide until screen size less than
576px</div>
<div class="container-md">100% wide until screen size less than
768px</div>
<div class="container-lg">100% wide until screen size less than
992px</div>
<div class="container-xl">100% wide until screen size less than
1200px</div>
```

Adding Background and Borders to Containers

By default, container doesn't have any `background-color` or `border`. But if you need you can apply your own styles, or simply use the Bootstrap background-color and border [utility classes](#) to add background-color or border on them, as shown in the following example.

```
<!-- Container with dark background and white text color -->
<div class="container bg-dark text-white">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>

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

<!-- Container with border -->
<div class="container border">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```


Bootstrap Helper Classes

Color Classes

You can use the contextual color classes to emphasize the text and convey meaning through color. Here's the complete list of available contextual color classes.

Class	Description
<code>.text-primary</code>	Apply blue color (■#007bff) on text.
<code>.text-secondary</code>	Apply grey color (■#6c757d) on text.
<code>.text-success</code>	Apply green color (■#28a745) on text.
<code>.text-danger</code>	Apply red color (■#dc3545) on text.
<code>.text-warning</code>	Apply yellow color (■#ffc107) on text.
<code>.text-info</code>	Apply teal color (■#17a2b8) on text.
<code>.text-light</code>	Apply very light grey color (□#f8f9fa) on text.
<code>.text-dark</code>	Apply dark grey color (■#343a40) on text.
<code>.text-body</code>	Apply Bootstrap's default body text color (■#212529) on text.
<code>.text-muted</code>	Apply light grey color (■#6c757d) on text.
<code>.text-white</code>	Apply white color (□#fff) on text.
<code>.text-black-50</code>	Apply 50% transparent black color (■ <code>rgba(0,0,0,.5)</code>) on text.
<code>.text-white-50</code>	Apply 50% transparent white color (□ <code>rgba(255,255,255,.5)</code>) on text.

Background Color Classes

Similar to the contextual color classes, you can use the contextual background color classes to set the `background-color` of an element to apply extra emphasis on them.

Class	Description
<code>.bg-primary</code>	Apply blue background-color (■#007bff) on an element.
<code>.bg-secondary</code>	Apply grey background-color (■#6c757d) on an element.
<code>.bg-success</code>	Apply green background-color (■#28a745) on an element.
<code>.bg-danger</code>	Apply red background-color (■#dc3545) on an element.
<code>.bg-warning</code>	Apply yellow background-color (■#ffc107) on an element.
<code>.bg-info</code>	Apply teal background-color (■#17a2b8) on an element.
<code>.bg-light</code>	Apply very light grey background-color (□#f8f9fa) on an element.
<code>.bg-dark</code>	Apply dark grey background-color (■#343a40) on an element.
<code>.bg-white</code>	Apply white background-color (□#fff) on an element.
<code>.bg-transparent</code>	Apply transparent background-color (□transparent) on an element.

Text Alignment Classes

You can use these text alignment classes to align and format your text content.

Class	Description
<code>.text-start</code>	Align the text to the left.
<code>.text-center</code>	Align the text to the center.
<code>.text-end</code>	Align the text to the right.
<code>.text-wrap</code>	Wrap the overflowing text.
<code>.text-nowrap</code>	Prevent text from wrapping.
<code>.text-truncate</code>	Truncate the text with an ellipsis.
<code>.text-break</code>	Break the long words to prevent overflow.
<code>.text-lowercase</code>	Transform the text to lowercase.
<code>.text-uppercase</code>	Transform the text to uppercase.
<code>.text-capitalize</code>	Capitalize the first letter of each word.
<code>.fw-bold</code>	Set the font-weight of an element to bold .
<code>.fw-bolder</code>	Set the font-weight of an element to bolder (relative to the parent).
<code>.fw-normal</code>	Set the font-weight of an element to normal.
<code>.fw-light</code>	Set the light font-weight for an element.

.fw-lighter	Set the lighter font-weight for an element (relative to the parent).
.fst-italic	Set the font-style of an element to <i>italic</i> .
.fst-normal	Set the font-style of an element to normal.
.font-monospace	Set the font-family of an element to monospace font (fixed-width)
.text-reset	Reset the color of a text or link (inherits the color from its parent).
.text-decoration-none	Remove the text decoration such as underline from a text.
.text-decoration-underline	Add underline to text.
.text-decoration-line-through	Add line through the middle of text.

Text Alignment Classes

You can use these text alignment classes to align and format your text content.

Class	Description
.text-start	Align the text to the left.
.text-center	Align the text to the center.
.text-end	Align the text to the right.
.text-wrap	Wrap the overflowing text.
.text-nowrap	Prevent text from wrapping.
.text-truncate	Truncate the text with an ellipsis.
.text-break	Break the long words to prevent overflow.
.text-lowercase	Transform the text to lowercase.
.text-uppercase	Transform the text to uppercase.
.text-capitalize	Capitalize the first letter of each word.
.fw-bold	Set the font-weight of an element to bold .
.fw-bolder	Set the font-weight of an element to bolder (relative to the parent).
.fw-normal	Set the font-weight of an element to normal.
.fw-light	Set the light font-weight for an element.
.fw-lighter	Set the lighter font-weight for an element (relative to the parent).
.fst-italic	Set the font-style of an element to <i>italic</i> .
.fst-normal	Set the font-style of an element to normal.
.font-monospace	Set the font-family of an element to monospace font (fixed-width)
.text-reset	Reset the color of a text or link (inherits the color from its parent).
.text-decoration-none	Remove the text decoration such as underline from a text.
.text-decoration-underline	Add underline to text.
.text-decoration-line-through	Add line through the middle of text.

Border Classes

You can use the border utility classes to quickly style the border and border-radius of an element. They are very useful for images, buttons, or any other element.

Class	Description
.border	Add a border on all sides of an element.
.border-top	Add a border on the top side of an element.
.border-end	Add a border on the right side of an element.
.border-bottom	Add a border on the bottom side of an element.
.border-start	Add a border on the left side of an element.
.border-0	Remove the border from all sides of an element.
.border-top-0	Remove the border from top side of an element.
.border-end-0	Remove the border from right side of an element.
.border-bottom-0	Remove the border from bottom side of an element.
.border-start-0	Remove the border from left side of an element.
.border-primary	Change the border-color of an element to blue (#007bff).
.border-secondary	Change the border-color of an element to grey (#6c757d).
.border-success	Change the border-color of an element to green (#28a745).
.border-danger	Change the border-color of an element to red (#dc3545).
.border-warning	Change the border-color of an element to yellow (#ffc107).
.border-info	Change the border-color of an element to teal (#17a2b8).
.border-light	Change the border-color of an element to very light grey (#f8f9fa).
.border-dark	Change the border-color of an element to dark grey (#343a40).
.border-white	Change the border-color of an element to white (fff).
.rounded	Round all corners of an element.
.rounded-top	Round top-left and top-right corners of an element.
.rounded-end	Round top-right and bottom-right corners of an element.
.rounded-bottom	Round bottom-left and bottom-right corners of an element.
.rounded-start	Round top-left and bottom-left corners of an element.
.rounded-circle	Round an element into circle shape.
.rounded-pill	Round an element into pill shape.
.rounded-0	Remove round corners from an element.
.rounded-1	Set border-radius of 0.2rem for an element.
.rounded-2	Set border-radius of 0.25rem for an element.
.rounded-3	Set border-radius of 0.3rem for an element.

Spacing Classes

You can use the spacing utility classes to quickly adjust the margin and padding of an element.

The # character in the classes `.mt-#`, `.pt-#`, etc. represents the size of the margin or padding. The value of the size can be an integer from 0 to 5, or `auto`.

The size 0 completely removes the margin or padding by setting it to 0, whereas the value `auto` sets the margin to auto. The integer numbers from 1 to 5 sets the margin or padding incrementally (ranging from 0.25rem to 3rem). Try out the examples to see how it works:

Class	Description
<code>.mt-#</code>	Set the top margin of an element.
<code>.mb-#</code>	Set the bottom margin of an element.
<code>.ms-#</code>	Set the left margin of an element.
<code>.me-#</code>	Set the right margin of an element.
<code>.mx-#</code>	Set the left and right margin of an element.
<code>.my-#</code>	Set the top and bottom margin of an element.
<code>.pt-#</code>	Set the top padding of an element.
<code>.pb-#</code>	Set the bottom padding of an element.
<code>.ps-#</code>	Set the left padding of an element.
<code>.pe-#</code>	Set the right padding of an element.
<code>.px-#</code>	Set the left and right padding of an element.
<code>.py-#</code>	Set the top and bottom padding of an element.

Tip: You can also apply negative margins using the margin utility classes. Just append the size number with the character `n`, like `.mt-n1`, `.mb-n2`, and so on.

Float Classes

You can use the float utility classes to quickly float an element to the left or right or remove float from an element altogether. Here's the list of available float classes.

Class	Description
<code>.float-start</code>	Float an element to the left.
<code>.float-end</code>	Float an element to the right.
<code>.float-none</code>	Disable floating from an element.
<code>.clearfix</code>	Clear floats to prevent parent element from collapsing.

<code>.float-*-start</code>	Float an element to the left on based on the viewport size.
<code>.float-*-end</code>	Float an element to the right on based on the viewport size.
<code>.float-*-none</code>	Disable floating from an element on based on the viewport size.

Tip: The * character in the classes `.float-*-left` and `.float-*-right` represents responsive breakpoint which can be `sm`, `md`, `lg`, `xl`, or `xxl`.

See <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php> for more detail.

Shadow Classes

You can use the shadow utility classes to quickly add or remove shadows to elements.

Class	Description
<code>.shadow</code>	Add a shadow to an element.
<code>.shadow-sm</code>	Add a small shadow to an element.
<code>.shadow-lg</code>	Add a larger shadow to an element.
<code>.shadow-none</code>	Remove shadow from an element.

Sizing Classes

You can use the sizing utility classes to easily make an element as wide or as tall.

Class	Description
<code>.w-25</code>	Set the width of an element to 25% of its parent element.
<code>.w-50</code>	Set the width of an element to 50% of its parent element.
<code>.w-75</code>	Set the width of an element to 75% of its parent element.
<code>.w-100</code>	Set the width of an element to 100% of its parent element.
<code>.w-auto</code>	Set the width of an element to auto, i.e. the browser will calculate and select a width for the specified element.
<code>.mw-100</code>	Set the max-width of an element to 100%.
<code>.h-25</code>	Set the height of an element to 25% of its parent element.
<code>.h-50</code>	Set the height of an element to 50% of its parent element.
<code>.h-75</code>	Set the height of an element to 75% of its parent element.
<code>.h-100</code>	Set the height of an element to 100% of its parent element.
<code>.h-auto</code>	Set the height of an element to auto, i.e. the browser will calculate and select a height for the specified element.
<code>.mh-100</code>	Set the max-height of an element to 100%.

<code>.vw-100</code>	Set the width of an element to 100% of the width of the viewport.
<code>.min-vw-100</code>	Set the min-width of an element to 100% of the width of the viewport.
<code>.vh-100</code>	Set the height of an element to 100% of the height of the viewport.
<code>.min-vh-100</code>	Set the min-height of an element to 100% of the height of the viewport.

Vertical Alignment Classes

You can use the vertical alignment utility classes to quickly change the vertical alignment of inline, inline-block, inline-table, and table cell elements.

Class	Description
<code>.align-baseline</code>	Align the baseline of the element with the baseline of its parent.
<code>.align-top</code>	Align the top of the element with the top of the entire line.
<code>.align-middle</code>	Align the element in the middle of the parent element.
<code>.align-bottom</code>	Align the bottom of the element with the bottom of the entire line.
<code>.align-text-top</code>	Align the top of the element with the top of the parent element's font.
<code>.align-text-bottom</code>	Align the bottom of the element with the bottom of the parent element's font.

Position Classes

You can use the position utility classes to quickly configuring the position of an element.

Class	Description
<code>.position-static</code>	Set the position of element to static.
<code>.position-relative</code>	Set the position of element to relative.
<code>.position-absolute</code>	Set the position of element to absolute.
<code>.position-fixed</code>	Set the position of element to fixed.
<code>.position-sticky</code>	Set the position of element to sticky.
<code>.fixed-top</code>	Position an element at the top of the viewport, from edge to edge.
<code>.fixed-bottom</code>	Position an element at the bottom of the viewport, from edge to edge.
<code>.sticky-top</code>	Position an element at the top of the viewport, from edge to edge, but that will only happen after you scroll past it.

Display Classes

You can use the display utility classes to quickly and responsively toggle the display value of an element. You can also control an element's display when printing the page.

Class	Description
<code>.d-none</code>	Hide an element. It does not take up any space in the layout.
<code>.d-inline</code>	Force an element to generate an inline-level box.
<code>.d-inline-block</code>	Force an element to generate a block box that will be flowed with surrounding content i.e. in the same line as adjacent content like inline box.
<code>.d-block</code>	Force an element to generate a block-level box.
<code>.d-table</code>	Force an element to behave like a <code><table></code> element.
<code>.d-table-row</code>	Force an element to behave like a <code><tr></code> element.
<code>.d-table-cell</code>	Force an element to behave like a <code><td></code> element.
<code>.d-flex</code>	Force an element to generate a block-level flex container box .
<code>.d-inline-flex</code>	Force an element to generate an inline-level flex container box.

You can also use the responsive breakpoints (`sm`, `md`, `lg`, `xl`, and `xxl`.) with these display utility classes to show or hide elements depending on viewport width using the format `.d-{breakpoint}-{value}`, for example, `.d-sm-none`, `.d-md-block`, `.d-lg-block`, `.d-xl-block`, `.d-xxl-block`, and so on.

For a complete set of flex utility classes, please check out the <https://getbootstrap.com/docs/5.0/utilities/flex/>

Tip: You can create print variant of the display classes to change the display value of elements when printing by simply inserting the string `print-` after `.d-`, for example, `.d-print-none`, `.d-print-inline`, `.d-print-block`, `.d-print-table`, and so on.

Miscellaneous Classes

Here's the list some more utility classes that can used for various purposes.

Class	Description
<code>.btn-close</code>	Create generic close icon for dismissing components like modals and alerts.
<code>.overflow-auto</code>	Provide scrollbars to view overflowed content of an element if required.
<code>.overflow-hidden</code>	Hide overflowed content of an element.
<code>.overflow-visible</code>	Show overflowed content of an element.
<code>.overflow-scroll</code>	Provide scrollbars to view overflowed content of an element.
<code>.stretched-link</code>	Make an element clickable by stretching a nested link.
<code>.visible</code>	Make an element visible, without modifying the display.
<code>.invisible</code>	Make an element invisible, but it still take up space in the page.
<code>.visually-hidden</code>	Hide an element to all devices except screen readers.

Bootstrap Grid System

What is Bootstrap Grid System?

Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes. It is built with [flexbox](#) with mobile-first approach. Also, it is fully responsive and uses twelve column system (12 columns available per row) and six default responsive tiers.

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

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

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

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

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

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

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

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

Creating Two Column Layouts

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

```
<div class="container">
  <!--Row with two equal columns-->
  <div class="row">
    <div class="col-md-6">Column left</div>
    <div class="col-md-6">Column right</div>
  </div>

  <!--Row with two columns divided in 1:2 ratio-->
  <div class="row">
    <div class="col-md-4">Column left</div>
    <div class="col-md-8">Column right</div>
  </div>

  <!--Row with two columns divided in 1:3 ratio-->
  <div class="row">
    <div class="col-md-3">Column left</div>
    <div class="col-md-9">Column right</div>
  </div>
</div>
```

Note: In a grid layout, content must be placed inside the columns (`.col` and `.col-*`) and only columns may be the immediate children of rows (`.row`). Also, rows should be placed inside a [container \(either fixed or fluid\)](#) for proper padding and alignment.

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Two Column Grid Layouts for Tablets and Desktops</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"rel
="stylesheet"/>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .demo-content {
        padding: 15px;
        font-size: 18px;
        background: #dbdfe5;
        margin-bottom: 15px;
      }
      .demo-content.bg-alt {
        background: #abb1b8;
      }
    </style>
  </head>
  <body>
    <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
    <div class="text-center my-3">
      Open the output in a new blank tab (Click the arrow next to "Show Output"
      button) and resize the browser window to understand how the Bootstrap
      responsive grid system works.
    </div>
    <div class="container">
      <!--Row with two equal columns-->
      <div class="row">
        <div class="col-md-6">
          <div class="demo-content">.col-md-6</div>
        </div>
        <div class="col-md-6">
          <div class="demo-content bg-alt">.col-md-6</div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

<!--Row with two columns divided in 1:2 ratio-->
<div class="row">
  <div class="col-md-4">
    <div class="demo-content">.col-md-4</div>
  </div>
  <div class="col-md-8">
    <div class="demo-content bg-alt">.col-md-8</div>
  </div>
</div>
<!--Row with two columns divided in 1:3 ratio-->
<div class="row">
  <div class="col-md-3">
    <div class="demo-content">.col-md-3</div>
  </div>
  <div class="col-md-9">
    <div class="demo-content bg-alt">.col-md-9</div>
  </div>
</div>
</div>
</body>
</html>

```

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

Creating Three Column Layouts

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

```

<div class="container">
  <!--Row with three equal columns-->
  <div class="row">
    <div class="col-lg-4">Column left</div>
    <div class="col-lg-4">Column middle</div>
    <div class="col-lg-4">Column right</div>
  </div>

```

```

<!--Row with three columns divided in 1:4:1 ratio-->
<div class="row">
  <div class="col-lg-2">Column left</div>
  <div class="col-lg-8">Column middle</div>
  <div class="col-lg-2">Column right</div>
</div>

<!--Row with three columns divided unevenly-->
<div class="row">
  <div class="col-lg-3">Column left</div>
  <div class="col-lg-7">Column middle</div>
  <div class="col-lg-2">Column right</div>
</div>
</div>

```

Note: If more than 12 grid columns are placed within a single row, then each group of extra columns, as a whole, will wrap onto a new line.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>
      Bootstrap Three Column Grid Layouts for Tablets (landscape) and Desktops
    </title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .demo-content {
        padding: 15px;
        font-size: 18px;
        background: #dbdfe5;
        margin-bottom: 15px;
      }
      .demo-content.bg-alt {
        background: #abb1b8;
      }
    </style>

```

```

</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">
    Open the output in a new blank tab (Click the arrow next to "Show Output"
    button) and resize the browser window to understand how the Bootstrap
    responsive grid system works.
  </div>
  <div class="container">
    <!--Row with three equal columns-->
    <div class="row">
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content bg-alt">.col-lg-4</div>
      </div>
      <div class="col-lg-4">
        <div class="demo-content">.col-lg-4</div>
      </div>
    </div>

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

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

```

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

Bootstrap Auto-layout Columns

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

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

```
<div class="container">
  <!--Row with two equal columns-->
  <div class="row">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
  </div>

  <!--Row with three equal columns-->
  <div class="row">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
</div>
```

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

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

```
<div class="container">
  <!--Row with two equal columns-->
  <div class="row">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
  </div>

  <!--Row with three columns divided in 1:2:1 ratio-->
  <div class="row">
```

```

        <div class="col">Column one</div>
        <div class="col-sm-6">Column two</div>
        <div class="col">Column three</div>
    </div>
</div>

////////////////////////////////////

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

      <!--Row with three columns divided in 1:2:1 ratio-->
      <div class="row">
        <div class="col">.col</div>

```



```

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

```

Column Wrapping Behavior

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

```

<div class="container">
  <div class="row">
    <div class="col-md-4 col-lg-3">Column one</div>
    <div class="col-md-8 col-lg-6">Column two</div>
    <div class="col-md-12 col-lg-3">Column three</div>
  </div>
</div>

```

////////////////////////////////////

```

<body>
  <h2 class="text-center mt-3">Bootstrap Column Wrapping</h2>
  <div class="text-center my-3">
    Open the output in a new blank tab (Click the arrow next to "Show Output"
    button) and resize the browser window to understand how the Bootstrap
    column wrapping works.
  </div>
  <div class="container">
    <div class="row">
      <div class="col-md-4 col-lg-3">
        <div class="demo-content">.col-md-4 .col-lg-3</div>
      </div>
      <div class="col-md-8 col-lg-6">
        <div class="demo-content bg-alt">.col-md-8 .col-lg-6</div>
      </div>
      <div class="col-md-12 col-lg-3">
        <div class="demo-content">.col-md-12 .col-lg-3</div>
      </div>
    </div>
  </div>

```

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

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

Similarly, you can create even more adaptable layouts for your websites using the Bootstrap's grid column wrapping feature

Creating Multi-Column Layouts with Bootstrap

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



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

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

Now the question is how we can create such responsive layouts using this Bootstrap flexbox grid system. Let's start with the primary target device. Suppose our primary target device is laptop or normal desktop. Since our laptop layout

has 3 columns and 4 rows i.e. 3x4 grid layout, so the HTML code for making such grid structure would look something like this.

```
<div class="container-lg">
  <div class="row">
    <div class="col-xl-4"><p>Box 1</p></div>
    <div class="col-xl-4"><p>Box 2</p></div>
    <div class="col-xl-4"><p>Box 3</p></div>
    <div class="col-xl-4"><p>Box 4</p></div>
    <div class="col-xl-4"><p>Box 5</p></div>
    <div class="col-xl-4"><p>Box 6</p></div>
    <div class="col-xl-4"><p>Box 7</p></div>
    <div class="col-xl-4"><p>Box 8</p></div>
    <div class="col-xl-4"><p>Box 9</p></div>
    <div class="col-xl-4"><p>Box 10</p></div>
    <div class="col-xl-4"><p>Box 11</p></div>
    <div class="col-xl-4"><p>Box 12</p></div>
  </div>
</div>
```

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Grid Layouts for Large Devices</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      p {
        padding: 50px;
        font-size: 32px;
        font-weight: bold;
        text-align: center;
        background: #dbdfe5;
      }
    </style>
```

```

</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
  <div class="text-center my-3">
    Open the output in a new blank tab (Click the arrow next to "Show Output"
    button) and resize the browser window to understand how the Bootstrap
    responsive grid system works.
  </div>
  <div class="container-lg">
    <div class="row">
      <div class="col-xl-4"><p>Box 1</p></div>
      <div class="col-xl-4"><p>Box 2</p></div>
      <div class="col-xl-4"><p>Box 3</p></div>
      <div class="col-xl-4"><p>Box 4</p></div>
      <div class="col-xl-4"><p>Box 5</p></div>
      <div class="col-xl-4"><p>Box 6</p></div>
      <div class="col-xl-4"><p>Box 7</p></div>
      <div class="col-xl-4"><p>Box 8</p></div>
      <div class="col-xl-4"><p>Box 9</p></div>
      <div class="col-xl-4"><p>Box 10</p></div>
      <div class="col-xl-4"><p>Box 11</p></div>
      <div class="col-xl-4"><p>Box 12</p></div>
    </div>
  </div>
</body>
</html>

```

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

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

```

<div class="container-lg">
  <div class="row">
    <div class="col-xl-4 col-md-6"><p>Box 1</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 2</p></div>

```

```
<div class="col-xl-4 col-md-6"><p>Box 3</p></div>
<div class="col-xl-4 col-md-6"><p>Box 4</p></div>
<div class="col-xl-4 col-md-6"><p>Box 5</p></div>
<div class="col-xl-4 col-md-6"><p>Box 6</p></div>
<div class="col-xl-4 col-md-6"><p>Box 7</p></div>
<div class="col-xl-4 col-md-6"><p>Box 8</p></div>
<div class="col-xl-4 col-md-6"><p>Box 9</p></div>
<div class="col-xl-4 col-md-6"><p>Box 10</p></div>
<div class="col-xl-4 col-md-6"><p>Box 11</p></div>
<div class="col-xl-4 col-md-6"><p>Box 12</p></div>
</div>
</div>
```

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

////////////////////////////////////

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Grid Layouts for Large Devices</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      p {
        padding: 50px;
        font-size: 32px;
        font-weight: bold;
        text-align: center;
        background: #dbdfe5;
      }
    </style>
  </head>
  <body>
    <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
    <div class="text-center my-3">
      Open the output in a new blank tab (Click the arrow next to "Show Output"
```

button) and resize the browser window to understand how the Bootstrap responsive grid system works.

```
</div>
<div class="container-lg">
  <div class="row">
    <div class="col-xl-4 col-md-6"><p>Box 1</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 2</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 3</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 4</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 5</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 6</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 7</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 8</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 9</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 10</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 11</p></div>
    <div class="col-xl-4 col-md-6"><p>Box 12</p></div>
  </div>
</div>
</body>
</html>
```

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

```
<div class="container-lg">
  <div class="row">
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 1</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 2</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 3</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 4</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 5</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 6</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 7</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 8</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 9</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 10</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 11</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 12</p></div>
  </div>
</div>
```

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

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

```

    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 7</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 8</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 9</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 10</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 11</p></div>
    <div class="col-xl-4 col-md-6 col-xxl-3"><p>Box 12</p></div>
  </div>
</div>
</body>
</html>

```

Nesting of Grid Columns

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

```

<div class="container">
  <div class="row">
    <div class="col-sm-8">Column left</div>
    <div class="col-sm-4">
      <!--Column right with nested rows and columns-->
      <div class="row">
        <div class="col-12"></div>
      </div>
      <div class="row">
        <div class="col-6"></div>
        <div class="col-6"></div>
      </div>
    </div>
  </div>
</div>
////////////////////////////////////

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

```



```
    rel="stylesheet"
  />
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
  <style>
    /* Some custom styles to beautify this example */
    .main-content {
      min-height: 230px;
      background: #dbdfe5;
    }
    .sidebar-content {
      min-height: 100px;
      margin-bottom: 30px;
      background: #b4bac0;
    }
  </style>
</head>
<body>
  <div class="container mt-3">
    <div class="row">
      <div class="col-sm-8">
        <div class="main-content"></div>
      </div>
      <div class="col-sm-4">
        <!--Nested rows within a column-->
        <div class="row">
          <div class="col-12">
            <div class="sidebar-content"></div>
          </div>
        </div>
        <div class="row">
          <div class="col-6">
            <div class="sidebar-content"></div>
          </div>
          <div class="col-6">
            <div class="sidebar-content"></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Creating Variable Width Columns

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

```
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col-md-3">Column left</div>
    <div class="col-md-auto">Variable width column</div>
    <div class="col-md-3">Column right</div>
  </div>
  <div class="row">
    <div class="col">Column left</div>
    <div class="col-auto">Variable width column</div>
    <div class="col">Column right</div>
  </div>
</div>
////////////////////////////////////

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

```

<body>
  <div class="container">
    <div class="row justify-content-md-center">
      <div class="col-md-3">Column left</div>
      <div class="col-md-auto">Variable width column</div>
      <div class="col-md-3">Column right</div>
    </div>
    <div class="row">
      <div class="col">Column left</div>
      <div class="col-auto">Variable width column</div>
      <div class="col">Column right</div>
    </div>
  </div>
  <p class="text-center mt-3">
    <strong>Note:</strong> Enter some more content inside the variable width
    column to understand how it works.
  </p>
</body>
</html>

```

Alignment of Grid Columns

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

Vertical Alignment of Grid Columns

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

```

<div class="container">
  <div class="row align-items-start">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
  <div class="row align-items-center">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
  <div class="row align-items-end">

```

```

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

```

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

```

<div class="container">
  <div class="row">
    <div class="col align-self-start">Column one</div>
    <div class="col align-self-center">Column two</div>
    <div class="col align-self-end">Column three</div>
  </div>
</div>

```

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

```

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

```

```

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

```

Horizontal Alignment of Grid Columns

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

```

<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
</div>

```

Alternatively, you can use the class `.justify-content-around` to distribute grid columns evenly with half-size spaces on either end, whereas you can use the

class `.justify-content-between` to distribute the grid columns evenly where the first column placed at the start and the last column placed at the end. Try out the following example to see how it actually works:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Horizontal Alignments of Grid Columns</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .row {
        margin-top: 1rem;
        background: #dbdfe5;
      }
      [class^="col-"] {
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0, 0, 0, 0.5);
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row justify-content-start">
        <div class="col-4">Column one</div>
        <div class="col-4">Column two</div>
      </div>
      <div class="row justify-content-center">
        <div class="col-4">Column one</div>
        <div class="col-4">Column two</div>
      </div>
      <div class="row justify-content-end">
        <div class="col-4">Column one</div>
        <div class="col-4">Column two</div>
      </div>
    </div>
  </body>
</html>
```

```

    </body>
</html>
////////////////////////////////////
<div class="container">
  <div class="row justify-content-around">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
  <div class="row justify-content-between">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
</div>
////////////////////////////////////

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Justify Grid Columns</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .row {
        margin-top: 1rem;
        background: #dbdfe5;
      }
      [class^="col-"] {
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0, 0, 0, 0.5);
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row justify-content-around">
        <div class="col-4">Column one</div>

```

```

        <div class="col-4">Column two</div>
    </div>
    <div class="row justify-content-between">
        <div class="col-4">Column one</div>
        <div class="col-4">Column two</div>
    </div>
</div>
</body>
</html>

```

Reordering of Grid Columns

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

```

<div class="container">
  <div class="row">
    <div class="col order-last">First, but ordered at last</div>
    <div class="col">Second, but unordered</div>
    <div class="col order-first">Last, but ordered at first</div>
  </div>
</div>

```

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

```

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

```



```

/* Some custom styles to beautify this example */
.row {
  margin-top: 1rem;
  background: #dbdfe5;
}
.col {
  padding: 10px 15px;
  background: #abb1b8;
  border: 1px solid rgba(0, 0, 0, 0.5);
}
</style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col order-last">First, but ordered at last</div>
      <div class="col">Second, but unordered</div>
      <div class="col order-first">Last, but ordered at first</div>
    </div>
  </div>
</body>
</html>

////////////////////////////////////

<div class="container">
  <div class="row">
    <div class="col order-4">First, but ordered at last</div>
    <div class="col">Second, but ordered at first</div>
    <div class="col order-1">Last, but ordered at second</div>
  </div>
</div>
////////////////////////////////////

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

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
<style>
  /* Some custom styles to beautify this example */
  .row {
    margin-top: 1rem;
    background: #dbdfe5;
  }
  .col {
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0, 0, 0, 0.5);
  }
</style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col order-4">First, but ordered at last</div>
      <div class="col">Second, but ordered at first</div>
      <div class="col order-1">Last, but ordered at second</div>
    </div>
  </div>
</body>
</html>

```

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

Creating Compact Columns

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

```

<div class="row g-0">
  <div class="col-4">Column one</div>
  <div class="col-4">Column two</div>
  <div class="col-4">Column three</div>
</div>

```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Compact Columns</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .demo-content {
        padding: 15px;
        font-size: 18px;
        min-height: 50px;
        background: #dbdfe5;
        margin-bottom: 15px;
      }
      .demo-content.bg-alt {
        background: #abb1b8;
      }
    </style>
  </head>
  <body>
    <h2 class="text-center">Columns with Gutters</h2>
    <div class="container mt-3">
      <div class="row">
        <div class="col-4">
          <div class="demo-content">.col-4</div>
        </div>
        <div class="col-4">
          <div class="demo-content bg-alt">.col-4</div>
        </div>
        <div class="col-4">
          <div class="demo-content">.col-4</div>
        </div>
      </div>
    </div>
    <hr />
    <h2 class="text-center">Columns without Gutters</h2>
    <div class="container mt-3">

```

```

<div class="row g-0">
  <div class="col-4">
    <div class="demo-content">.col-4</div>
  </div>
  <div class="col-4">
    <div class="demo-content bg-alt">.col-4</div>
  </div>
  <div class="col-4">
    <div class="demo-content">.col-4</div>
  </div>
</div>
</div>
</body>
</html>

```

Breaking Columns to a New Line

You can also create equal-width columns that span multiple rows by inserting a `<div>` with `.w-100` class where you want the columns to break to a new line.

```

<div class="container">
  <!-- Break columns on all devices -->
  <div class="row">
    <div class="col">.col</div>
    <div class="col">.col</div>
    <div class="w-100"></div>
    <div class="col">.col</div>
    <div class="col">.col</div>
  </div>
  <!-- Break columns on all devices except extra large devices -->
  <div class="row">
    <div class="col">.col</div>
    <div class="col">.col</div>
    <div class="w-100 d-xl-none"></div>
    <div class="col">.col</div>
    <div class="col">.col</div>
  </div>
</div>

```

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

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Breaking Columns to a New Lines</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
    <style>
      /* Some custom styles to beautify this example */
      .row {
        margin-top: 1rem;
        background: #dbdfe5;
      }
      [class^="col"] {
        padding: 10px 15px;
        background: #abb1b8;
        border: 1px solid rgba(0, 0, 0, 0.5);
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h4 class="mt-3">Breaking columns on all devices</h4>
      <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
        <div class="w-100"></div>
        <div class="col">.col</div>
        <div class="col">.col</div>
      </div>
      <h4 class="mt-3">
        Breaking columns on all devices except extra large devices
      </h4>
      <div class="row">
        <div class="col">.col</div>
        <div class="col">.col</div>
        <div class="w-100 d-xl-none"></div>
        <div class="col">.col</div>
        <div class="col">.col</div>
      </div>
    </div>
  </body>
</html>

```

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

Bootstrap Fixed Layout

Creating Fixed Layout with Bootstrap

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

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

Further columns can be created inside a row using the predefined grid classes such as `.col`, `col-{xs|sm|md|lg|x1|xx1}-*`, where `*` represent grid number and should be from 1 to 12.

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

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

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

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Fixed Layout Example</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Tutorial Republic</a>
    <button type="button" class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>
        <a href="#" class="nav-item nav-link">Services</a>
        <a href="#" class="nav-item nav-link">About</a>
        <a href="#" class="nav-item nav-link">Contact</a>
      </div>
      <div class="navbar-nav ms-auto">
        <a href="#" class="nav-item nav-link">Register</a>
        <a href="#" class="nav-item nav-link">Login</a>
      </div>
    </div>
  </div>
</nav>
<div class="container">
  <div class="p-5 my-4 bg-light rounded-3">
    <h1>Learn to Create Websites</h1>
    <p class="lead">In today's world internet is the most popular way of
connecting with the people. At <a href="https://www.tutorialrepublic.com"
class="text-success" target="_blank">tutorialrepublic.com</a> you will learn the
essential web development technologies along with real life practice examples, so
that you can create your own website to connect with the people around the
world.</p>
    <p><a href="https://www.tutorialrepublic.com" target="_blank" class="btn
btn-success btn-lg">Get started today</a></p>

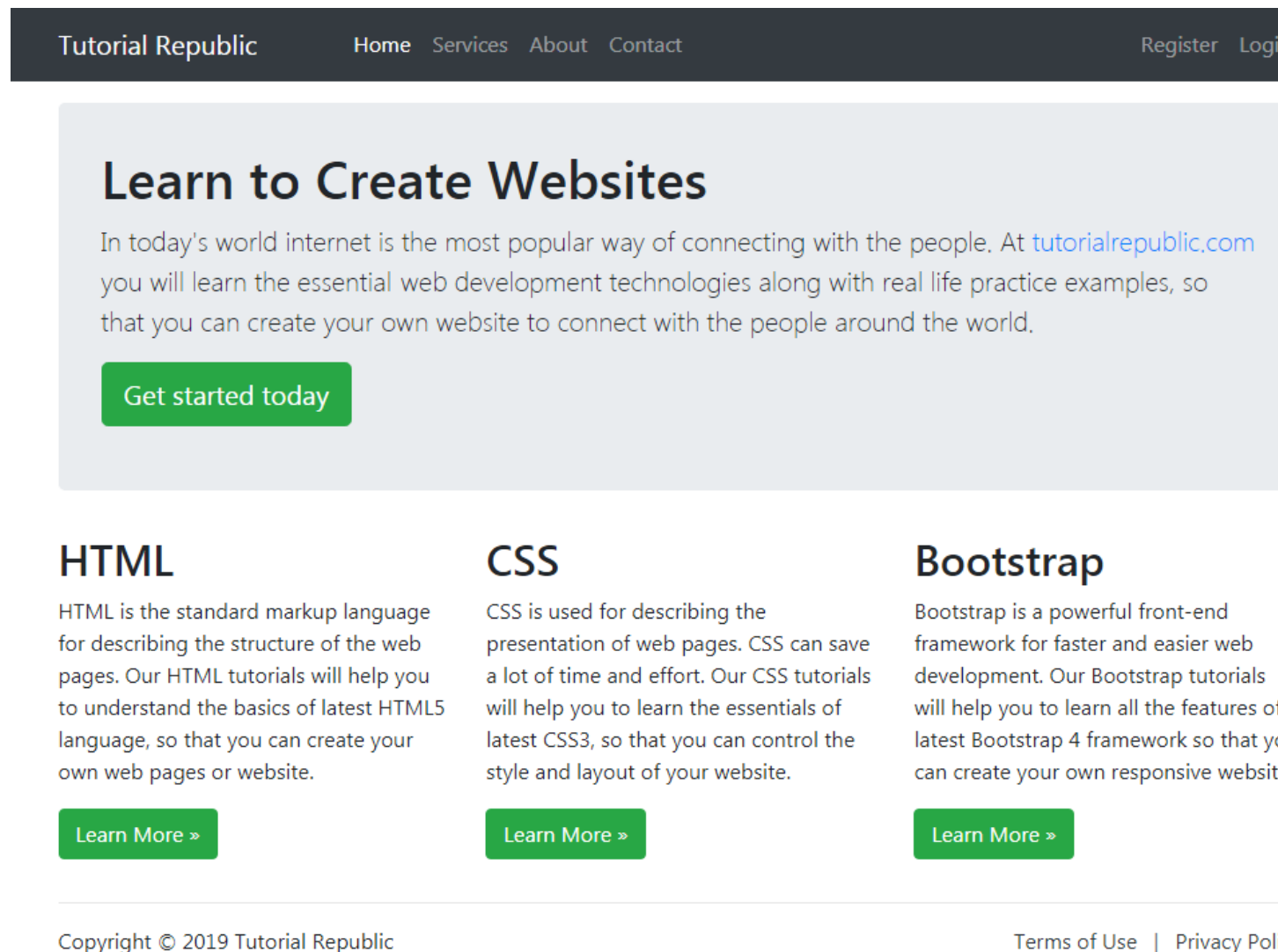
```

```

</div>
<div class="row">
  <div class="col-md-4">
    <h2>HTML</h2>
    <p>HTML is the standard markup language for describing the structure
of the web pages. Our HTML tutorials will help you to understand the basics of
latest HTML5 language, so that you can create your own web pages or website.</p>
    <p><a href="https://www.tutorialrepublic.com/html-tutorial/"
target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
  </div>
  <div class="col-md-4">
    <h2>CSS</h2>
    <p>CSS is used for describing the presentation of web pages. CSS can
save a lot of time and effort. Our CSS tutorials will help you to learn the
essentials of latest CSS3, so that you can control the style and layout of your
website.</p>
    <p><a href="https://www.tutorialrepublic.com/css-tutorial/"
target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
  </div>
  <div class="col-md-4">
    <h2>Bootstrap</h2>
    <p>Bootstrap is a powerful front-end framework for faster and easier
web development. Our Bootstrap tutorials will help you to learn all the features
of latest Bootstrap 4 framework so that you can easily create responsive
websites.</p>
    <p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-
tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
  </div>
</div>
<hr>
<footer>
  <div class="row">
    <div class="col-md-6">
      <p>Copyright &copy; 2021 Tutorial Republic</p>
    </div>
    <div class="col-md-6 text-md-end">
      <a href="#" class="text-dark">Terms of Use</a>
      <span class="text-muted mx-2">|</span>
      <a href="#" class="text-dark">Privacy Policy</a>
    </div>
  </div>
</footer>
</div>
</body>
</html>

```


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



Copyright © 2019 Tutorial Republic

[Terms of Use](#) | [Privacy Policy](#)

We've used the margin utility classes such as `.mb-3`, `.m1-auto`, `mx-2` etc. to adjust spacing between the elements. Whereas the classes `.text-dark`, `.text-muted`, `.text-md-right` are text utility classes to adjust color and alignment of text. You'll learn about them in later chapters.

Creating Fluid Layout with Bootstrap

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

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

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Fluid Layout</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a href="#" class="navbar-brand">Tutorial Republic</a>
        <button
          type="button"
          class="navbar-toggler"
          data-bs-toggle="collapse"
          data-bs-target="#navbarCollapse"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <div class="navbar-nav">
            <a href="#" class="nav-item nav-link active">Home</a>
            <a href="#" class="nav-item nav-link">Services</a>
            <a href="#" class="nav-item nav-link">About</a>
            <a href="#" class="nav-item nav-link">Contact</a>
          </div>
          <div class="navbar-nav ms-auto">
            <a href="#" class="nav-item nav-link">Login</a>
            <a href="#" class="nav-item nav-link">Register</a>
          </div>
        </div>
      </div>
    </nav>
  </body>
</html>
```

```

        </div>
    </div>
</div>
</nav>
<div class="container-fluid">
    <div class="p-5 my-4 bg-light rounded-3">
        <h1>Learn to Create Websites</h1>
        <p class="lead">
            In today's world internet is the most popular way of connecting with
            the people. At
            <a
                href="https://www.tutorialrepublic.com"
                class="text-success"
                target="_blank"
            >tutorialrepublic.com</a>
        >
        you will learn the essential web development technologies along with
        real life practice examples, so that you can create your own website
        to connect with the people around the world.
    </p>
    <p>
        <a
            href="https://www.tutorialrepublic.com"
            target="_blank"
            class="btn btn-success btn-lg"
        >Get started today</a>
    >
    </p>
</div>
<div class="row">
    <div class="col-md-4">
        <h2>HTML</h2>
        <p>
            HTML is the standard markup language for describing the structure of
            the web pages. Our HTML tutorials will help you to understand the
            basics of latest HTML5 language, so that you can create your own web
            pages or website.
        </p>
        <p>
            <a
                href="https://www.tutorialrepublic.com/html-tutorial/"
                target="_blank"
                class="btn btn-success"
            >Learn More &raquo;</a>
        >
    </p>

```

```
</div>
<div class="col-md-4">
  <h2>CSS</h2>
  <p>
    CSS is used for describing the presentation of web pages. CSS can
    save a lot of time and effort. Our CSS tutorials will help you to
    learn the essentials of latest CSS3, so that you can control the
    style and layout of your website.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/css-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
</div>
<div class="col-md-4">
  <h2>Bootstrap</h2>
  <p>
    Bootstrap is a powerful front-end framework for faster and easier
    web development. Our Bootstrap tutorials will help you to learn all
    the features of latest Bootstrap 4 framework so that you can easily
    create responsive websites.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
</div>
</div>
<hr />
<footer>
  <div class="row">
    <div class="col-md-6">
      <p>Copyright &copy; 2021 Tutorial Republic</p>
    </div>
    <div class="col-md-6 text-md-end">
      <a href="#" class="text-dark">Terms of Use</a>
      <span class="text-muted mx-2">|</span>
      <a href="#" class="text-dark">Privacy Policy</a>
    </div>
  </div>
</footer>
```

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

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

Learn to Create Websites

In today's world internet is the most popular way of connecting with the people. At tutorialrepublic.com you will learn the essential web development technologies along with real life practice examples, so that you can create your own website to connect with the people around the world.

Get started today

HTML

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

Learn More »

CSS

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

Learn More »

Bootstrap

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

Learn More »

Bootstrap Responsive Layout

Creating Responsive Layout with Bootstrap

With the Bootstrap powerful mobile first [flexbox](#) grid system creating the responsive and mobile friendly websites and applications has become much easier.

Bootstrap is responsive and mobile friendly from the start. Its [six tier grid classes](#) provides better control over the layout as well as how it will be rendered on different types of devices like mobile phones, tablets, laptops and desktops, large screen devices, and so on.

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap 4 Responsive Layout</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a href="#" class="navbar-brand">Tutorial Republic</a>
        <button
          type="button"
          class="navbar-toggler"
          data-bs-toggle="collapse"
          data-bs-target="#navbarCollapse"
```

```

    >
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
  <div class="navbar-nav">
    <a href="#" class="nav-item nav-link active">Home</a>
    <a href="#" class="nav-item nav-link">Services</a>
    <a href="#" class="nav-item nav-link">About</a>
    <a href="#" class="nav-item nav-link">Contact</a>
  </div>
  <div class="navbar-nav ms-auto">
    <a href="#" class="nav-item nav-link">Login</a>
    <a href="#" class="nav-item nav-link">Register</a>
  </div>
</div>
</div>
</nav>
<div class="container">
  <div class="p-5 my-4 bg-light rounded-3">
    <h1>Learn to Create Websites</h1>
    <p class="lead">
      In today's world internet is the most popular way of connecting with
      the people. At
      <a
        href="https://www.tutorialrepublic.com"
        class="text-success"
        target="_blank"
      >tutorialrepublic.com</a>
    >
    >
    you will learn the essential web development technologies along with
    real life practice examples, so that you can create your own website
    to connect with the people around the world.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com"
      target="_blank"
      class="btn btn-success btn-lg"
    >Get started today</a>
  >
  </p>
</div>
<div class="row g-3">
  <div class="col-md-6 col-lg-4 col-xl-3">
    <h2>HTML</h2>
    <p>

```

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

```
</p>
<p>
  <a
    href="https://www.tutorialrepublic.com/html-tutorial/"
    target="_blank"
    class="btn btn-success"
    >Learn More &raquo;</a>
  >
</p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <h2>CSS</h2>
  <p>
    CSS is used for describing the presentation of web pages. CSS can
    save a lot of time and effort. Our CSS tutorials will help you to
    learn the essentials of latest CSS3, so that you can control the
    style and layout of your website.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/css-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <h2>JavaScript</h2>
  <p>
    JavaScript is the most popular and widely used client-side scripting
    language. Our JavaScript tutorials will provide in-depth knowledge
    of the JavaScript including ES6 features, so that you can create
    interactive websites.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/javascript-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
```



```
</p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <h2>Bootstrap</h2>
  <p>
    Bootstrap is a powerful front-end framework for faster and easier
    web development. Our Bootstrap tutorials will help you to learn all
    the features of latest Bootstrap 4 framework so that you can easily
    create responsive websites.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <h2>PHP</h2>
  <p>
    PHP is the most popular server-side scripting language for creating
    dynamic web pages. Our PHP tutorials will help you to learn all the
    features of latest PHP7 scripting language so that you can easily
    create dynamic websites.
  </p>
  <p>
    <a
      href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
      target="_blank"
      class="btn btn-success"
      >Learn More &raquo;</a>
    >
  </p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <h2>SQL</h2>
  <p>
    SQL is a standard language designed for managing data in relational
    database management system. Our SQL tutorials will help you to learn
    the fundamentals of the SQL language so that you can efficiently
    manage your databases.
  </p>
  <p>
    <a
```

```

        href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
        target="_blank"
        class="btn btn-success"
        >Learn More &raquo;</a
    >
</p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <h2>References</h2>
    <p>
        Our references section outlines all the standard HTML5 tags and CSS3
        properties along with other useful references such as color names
        and values, character entities, web safe fonts, language codes, HTTP
        messages, and more.
    </p>
    <p>
        <a
            href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
            target="_blank"
            class="btn btn-success"
            >Learn More &raquo;</a
        >
    </p>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <h2>FAQ</h2>
    <p>
        Our Frequently Asked Questions (FAQ) section is an extensive
        collection of FAQs that provides quick and working solution of
        common questions and queries related to web design and development
        with live demo.
    </p>
    <p>
        <a
            href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/"
            target="_blank"
            class="btn btn-success"
            >Learn More &raquo;</a
        >
    </p>
</div>
</div>
<hr />
<footer>
    <div class="row">
        <div class="col-md-6">

```

```
        <p>Copyright © 2021 Tutorial Republic</p>
    </div>
    <div class="col-md-6 text-md-end">
        <a href="#" class="text-dark">Terms of Use</a>
        <span class="text-muted mx-2">|</span>
        <a href="#" class="text-dark">Privacy Policy</a>
    </div>
</div>
</footer>
</div>
</body>
</html>
```

Bootstrap Typography

Working with Headings

You can define all [HTML headings](#), `<h1>` through `<h6>` — In the same way you define in simple HTML document. You can also utilize the heading classes `.h1` through `.h6` on other elements, if you want to apply the style on element's text same as headings.

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

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

h1. Bootstrap heading

h2. Bootstrap heading

h3. Bootstrap heading

h4. Bootstrap heading

h5. Bootstrap heading

h6. Bootstrap heading

Customizing Headings

Additionally, you can use the `<small>` tag with `.text-muted` class to display the secondary text of any heading in a smaller and lighter variation. Here's an example:

```
<h2>  
  Fancy display heading  
  <small class="text-muted">With faded secondary text</small>  
</h2>
```

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

Fancy display heading With faded secondary text

Display Headings

Bootstrap also provides display headings that can be used when you need a heading to stand out. Display headings are displayed in larger `font-size` but lighter `font-weight`.

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

```
<h1 class="display-1">Display Heading 1</h1>  
<h1 class="display-2">Display Heading 2</h1>  
<h1 class="display-3">Display Heading 3</h1>  
<h1 class="display-4">Display Heading 4</h1>  
<h1 class="display-5">Display Heading 5</h1>  
<h1 class="display-6">Display Heading 6</h1>
```

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

Display Heading 1

Display Heading 2

Display Heading 3

Display Heading 4

Display Heading 5

Display Heading 6

Working with Paragraphs

Bootstrap's global default `font-size` is **1rem** (typically 16px), with a `line-height` of **1.5** (typically 24px), which is applied to the `<body>` element as well as all the paragraphs i.e. the `<p>` elements. In addition to that `margin-bottom` of **1rem** is also applied to all the paragraphs.

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

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

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

This is how a normal paragraph looks like in Bootstrap.

This is how a paragraph stands out in Bootstrap.

Tip: In CSS `rem` stands for "root em". 1rem is equal to the font size of the root element (i.e. the `<html>` element), which is 16px in most browsers by default.

Text Alignment

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

```
<p class="text-start">Left aligned text on all viewport sizes.</p>
<p class="text-center">Center aligned text on all viewport sizes.</p>
<p class="text-end">Right aligned text on all viewport sizes.</p>
```

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

Left aligned text on all viewport sizes.

Center aligned text on all viewport sizes.

Right aligned text on all viewport sizes.

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

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

Text Formatting

You are free to use text formatting tags like ``, `<i>`, `<small>` to make your text bold, italic, small and so on, in the same way you do in simple HTML page. Here's an example:

```
<p><b>This is bold text</b></p>
<p><code>This is computer code</code></p>
<p><em>This is emphasized text</em></p>
<p><i>This is italic text</i></p>
<p><mark>This is highlighted text</mark></p>
<p><small>This is small text</small></p>
```

```
<p><strong>This is strongly emphasized text</strong></p>
<p>This is <sub>subscript</sub> and <sup>superscript</sup></p>
<p><ins>This text is inserted to the document</ins></p>
<p><del>This text is deleted from the document</del></p>
```

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

This is bold text

This is computer code

This is emphasized text

This is italic text

This is highlighted text

This is small text

This is strongly emphasized text

This is _{subscript} and ^{superscript}

This text is inserted to the document

~~This text is deleted from the document~~

Text Transformation

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

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

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

the quick brown fox jumps over the lazy dog.

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.

The Quick Brown Fox Jumps Over The Lazy Dog.

Text Coloring

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

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

```
<p class="text-primary">Primary: Please read the instructions  
carefully before proceeding.</p>  
<p class="text-secondary">Secondary: This is featured has been removed  
from the latest version.</p>  
<p class="text-success">Success: Your message has been sent  
successfully.</p>  
<p class="text-info">Info: You must agree with the terms and  
conditions to complete the sign up process.</p>  
<p class="text-warning">Warning: There was a problem with your network  
connection.</p>  
<p class="text-danger">Danger: An error has been occurred while  
submitting your data.</p>  
<p class="text-muted">Muted: This paragraph of text is grayed out.</p>
```

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

Primary: Please read the instructions carefully before proceeding.

Secondary: This is featured has been removed from the latest version.

Success: Your message has been sent successfully.

Info: You must agree with the terms and conditions to complete the sign up process.

Warning: There was a problem with your network connection.

Danger: An error has been occurred while submitting your data.

Muted: This paragraph of text is grayed out.

Wrapping Long Word

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

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

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Word Break</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
"></script>
</head>
<body>
<div class="m-4">
  <div class="container">
    <h2>Without Word Break</h2>
    <div class="row">
      <div class="col-2">
        <p class="bg-warning">veryveryveryveryveryveryveryverylongword</p>
      </div>
    </div>
    <hr />
    <h2>With Word Break</h2>
```

```
<div class="row">
  <div class="col-2">
    <p class="bg-warning text-
break">veryveryveryveryveryveryveryverylongword</p>
  </div>
</div>
</div>
</div>
</body>
</html>
```

Without Word Break

veryveryveryveryveryveryveryverylongword

With Word Break

veryveryveryveryveryveryve
rylongword

Bootstrap Tables

What is Table?

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

Creating a Simple Table with Bootstrap

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Simple Table</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss"
      rel="stylesheet"
    />
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"></script>
  </head>
  <body>
    <div class="m-4">
      <table class="table">
        <thead>
          <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
          </tr>
          <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
          </tr>
          <tr>
            <td>3</td>
            <td>John</td>
```

```
        <td>Carter</td>
        <td>johncarter@mail.com</td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>
```

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Accented Tables

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

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

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

```

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

```

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

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

```

<table class="table">
  <thead>
    <tr>
      <th>#</th>
      <th>Bill</th>
      <th>Payment Date</th>
      <th>Payment Status</th>
    </tr>
  </thead>
  <tbody>
    <tr class="table-primary">
      <td>1</td>
      <td>Credit Card</td>
      <td>04/07/2021</td>

```

```
    <td>Waiting for statement</td>
</tr>
<tr class="table-secondary">
    <td>2</td>
    <td>Insurance</td>
    <td>02/07/2021</td>
    <td>Cancelled</td>
</tr>
<tr class="table-success">
    <td>3</td>
    <td>Water</td>
    <td>01/07/2021</td>
    <td>Paid</td>
</tr>
<tr class="table-info">
    <td>4</td>
    <td>Internet</td>
    <td>05/07/2021</td>
    <td>Change plan</td>
</tr>
<tr class="table-warning">
    <td>5</td>
    <td>Electricity</td>
    <td>03/07/2021</td>
    <td>Pending</td>
</tr>
<tr class="table-danger">
    <td>6</td>
    <td>Telephone</td>
    <td>06/07/2021</td>
    <td>Due</td>
</tr>
<tr class="table-light">
    <td>7</td>
    <td>Car Service</td>
    <td>08/07/2021</td>
    <td>Call in to confirm</td>
</tr>
<tr class="table-dark">
    <td>8</td>
    <td>Gas</td>
    <td>06/07/2021</td>
    <td>Payment failed</td>
</tr>
</tbody>
</table>
```

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

#	Bill	Payment Date	Payment Status
1	Credit Card	04/07/2021	Waiting for statement
2	Insurance	02/07/2021	Cancelled
3	Water	01/07/2021	Paid
4	Internet	05/07/2021	Change plan
5	Electricity	03/07/2021	Pending
6	Telephone	06/07/2021	Due
7	Car Service	08/07/2021	Call in to confirm
8	Gas	06/07/2021	Payment failed

Creating Tables with Striped Rows

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

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

```

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

```

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Bordered Tables

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

```

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

```



```

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

```

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Borderless Tables

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

```

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

```

```
<tbody>
  <tr>
    <td>1</td>
    <td>Clark</td>
    <td>Kent</td>
    <td>clarkkent@mail.com</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Peter</td>
    <td>Parker</td>
    <td>peterparker@mail.com</td>
  </tr>
  <tr>
    <td>3</td>
    <td>John</td>
    <td>Carter</td>
    <td>johncarter@mail.com</td>
  </tr>
</tbody>
</table>
```

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Enabling Hover State on Table Rows

You can also enable a hover state on table rows within a `<tbody>` element by adding the modifier class `.table-hover` to the `.table` base class. Let's try out the following example:

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

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Small or Compact Tables

You can also make your tables more compact and save the space through adding the modifier class `.table-sm` to the `.table` base class. The `.table-sm` class makes the table compact by cutting all cell padding in half. Let's take a look at the following example:

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

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Setting Table Head Colors

Similar to light and dark tables, you can use the modifier classes `.table-light` or `.table-dark` on the `<thead>` element to make it appear in light or dark gray.

The following example will create a table with light gray background head.

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

</table>

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

The following example will create a table with dark gray background head.

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

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

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Responsive Tables with Bootstrap

You can also create responsive tables to enable horizontal scrolling on small devices.

To make any table responsive just place it inside a `<div>` element and apply the `.table-responsive` class on it. You can also specify when the table should have a scrollbar, based on the viewport width (i.e. [breakpoints](#)), using the classes `.table-responsive{-sm|-md|-lg|-xl}`.

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

```
<div class="table-responsive">
  <table class="table">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
        <th>Biography</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
        <td>Lorem ipsum dolor sit amet...</td>
      </tr>
    </tbody>
  </table>
</div>
```

```
<tr>
  <td>2</td>
  <td>Peter</td>
  <td>Parker</td>
  <td>peterparker@mail.com</td>
  <td>Integer pulvinar leo id risus...</td>
</tr>
<tr>
  <td>3</td>
  <td>John</td>
  <td>Carter</td>
  <td>johncarter@mail.com</td>
  <td>Vestibulum consectetur scelerisque...</td>
</tr>
</tbody>
</table>
</div>
```

Tip: Text inside the cells of `<thead>` are always vertical aligned to the bottom. Whereas, text inside the cells of `<tbody>` inherit their alignment from `<table>` and are vertically aligned to the top by default. Use the [vertical align classes](#) to re-align text where needed.

Bootstrap Forms

Creating Forms with Bootstrap

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

Bootstrap provides three different types of form layouts:

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

The following section will give you the detailed overview of all these form layouts as well as the various form related Bootstrap components one by one. Well let's get started.

Creating Vertical Form Layout

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

Here's an example in which form controls are vertically stacked with labels on the top.

```
<body>
  <div class="m-4">
    <form action="/examples/actions/confirmation.php" method="post">
      <div class="mb-3">
        <label class="form-label" for="inputEmail">Email</label>
        <input
          type="email"
          class="form-control"
          id="inputEmail"
          placeholder="Email"
          required
        />
      </div>
      <div class="mb-3">
        <label class="form-label" for="inputPassword">Password</label>
        <input
          type="password"
          class="form-control"
          id="inputPassword"
          placeholder="Password"
          required
        />
      </div>
      <div class="mb-3">
        <div class="form-check">
          <input class="form-check-input" type="checkbox" id="checkRemember" />
          <label class="form-check-label" for="checkRemember"
            >Remember me</label>
        </div>
      </div>
    </form>
  </div>
</body>
```

```
        <button type="submit" class="btn btn-primary">Sign in</button>
    </form>
</div>
</body>
```

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

Email

Password

Remember me

Sign in

Note: All textual form controls, such as `<input>` and `<textarea>` requires the class `.form-control`, while `<select>` requires the class `.form-select` for general styling. These classes also makes the forms controls 100% wide. To change their width or use them inline, you can utilize the [Bootstrap's predefined grid classes](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php). <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php>

Tip: It is recommend to use margin-bottom utility classes (e.g., `mb-2`, `mb-3`, etc.) to add vertical spacing between the form groups. As, using single direction margin throughout in the form prevent margin collapsing and create more consist form.

Creating Horizontal Form Layout

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

Also, be sure to apply the class `.col-form-label` on the `<label>` elements, so that they're vertically centered with their associated form controls. Let's check out an example:

```
<body>
  <div class="m-4">
    <form action="/examples/actions/confirmation.php" method="post">
      <div class="row mb-3">
        <label for="inputEmail" class="col-sm-2 col-form-label">Email</label>
        <div class="col-sm-10">
          <input
            type="email"
            class="form-control"
            id="inputEmail"
            placeholder="Email"
            required
          />
        </div>
      </div>
      <div class="row mb-3">
        <label for="inputPassword" class="col-sm-2 col-form-label"
          >Password</label
        >
        <div class="col-sm-10">
          <input
            type="password"
            class="form-control"
            id="inputPassword"
            placeholder="Password"
            required
          />
        </div>
      </div>
      <div class="row mb-3">
        <div class="col-sm-10 offset-sm-2">
          <div class="form-check">
            <input
              class="form-check-input"
              type="checkbox"
              id="checkRemember"
            />
            <label class="form-check-label" for="checkRemember"
              >Remember me</label
            >
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-sm-10 offset-sm-2">
          <button type="submit" class="btn btn-primary">Sign in</button>
        </div>
      </div>
    </form>
  </div>
</body>
```

```
        </div>
      </div>
    </form>
  </div>
</body>
```

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

Email

Password

Remember me

Creating Inline Form Layout

Sometimes you may want to display a series of form controls, and buttons in a single horizontal row to compact the layout. You can do this easily by using the [Bootstrap's grid classes](#).

Let's take a look at following example and see how it actually works:

```
<body>
  <div class="m-4">
    <form action="/examples/actions/confirmation.php" method="post">
      <div class="row align-items-center g-3">
        <div class="col-auto">
          <label class="visually-hidden" for="inputEmail">Email</label>
          <input
            type="email"
            class="form-control"
            id="inputEmail"
            placeholder="Email"
            required
          />
        </div>
        <div class="col-auto">
          <label class="visually-hidden" for="inputPassword">Password</label>
          <input
            type="password"
            class="form-control"

```

```

        id="inputPassword"
        placeholder="Password"
        required
    />
</div>
<div class="col-auto">
    <div class="form-check">
        <input
            class="form-check-input"
            type="checkbox"
            id="checkRemember"
        />
        <label class="form-check-label" for="checkRemember"
            >Remember me</label>
    >
    </div>
</div>
<div class="col-auto">
    <button type="submit" class="btn btn-primary">Sign in</button>
</div>
</div>
</form>
</div>
</body>

```

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

Check out the snippets section for examples of some [beautifully designed Bootstrap forms](#).

<https://www.tutorialrepublic.com/snippets/gallery.php?tag=form>

Tip: It is recommended to include a label for every form inputs otherwise screen readers will have trouble with your forms. However, in case of inline form layouts you can hide the labels using the `.visually-hidden` class, so that only screen readers can read it.

Creating Responsive Form Layout

You can also make your forms responsive through using the grid classes with specific breakpoints.

The following example will create a form which laid out inline on medium devices and up (i.e., viewport width $\geq 768\text{px}$), but will become vertically stacked on smaller viewports.

```
<body>
  <div class="m-4">
    <form action="/examples/actions/confirmation.php" method="post">
      <div class="row align-items-center g-3">
        <div class="col-md-auto col-12">
          <label class="form-label d-md-none" for="inputEmail">Email</label>
          <input
            type="email"
            class="form-control"
            id="inputEmail"
            placeholder="Email"
            required
          />
        </div>
        <div class="col-md-auto col-12">
          <label class="form-label d-md-none" for="inputPassword"
            >Password</label>
          <input
            type="password"
            class="form-control"
            id="inputPassword"
            placeholder="Password"
            required
          />
        </div>
        <div class="col-md-auto col-12">
          <div class="form-check">
            <input
              class="form-check-input"
              type="checkbox"
              id="checkRemember"
            />
            <label class="form-check-label" for="checkRemember"
              >Remember me</label>
          </div>
        </div>
      </div>
    </form>
  </div>
</body>
```

```

        </div>
    </div>
    <div class="col-md-auto col-12">
        <button type="submit" class="btn btn-primary">Sign in</button>
    </div>
</div>
</form>
<p class="mt-4">
    <strong>Note:</strong> Please change the Editor layout/orientation to see
    how responsive layout actually works.
</p>
</div>
</body>

```

Creating Static Form Control

There might be a situation when you just want to display a plain text value next to a form label instead of a working form control. You can do this easily by replacing the class `.form-control` with the `.form-control-plaintext` and applying the attribute `readonly`.

The `.form-control-plaintext` class removes the default styling from the form field, but preserves the correct margin and padding. Let's take a look at an example:

```

<body>
  <div class="m-4">
    <form action="/examples/actions/confirmation.php" method="post">
      <div class="row mb-3">
        <label for="inputEmail" class="col-sm-2 col-form-label">Email</label>
        <div class="col-sm-10">
          <input
            type="email"
            readonly
            class="form-control-plaintext"
            id="inputEmail"
            value="peterparker@example.com"
          />
        </div>
      </div>
      <div class="row mb-3">
        <label for="inputPassword" class="col-sm-2 col-form-label"
          >Password</label>
        >
        <div class="col-sm-10">

```

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

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

Email peterparker@example.com

Password

Remember me

Column Sizing of Form Controls

You can also match the size of your inputs, textareas and select boxes to the Bootstrap grid column sizes. Simply, place your form controls (i.e. `<input>`, `<textarea>`, and `<select>`) in grid columns.

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

```
<body>
  <div class="m-4">
    <div class="row g-3">
      <div class="col-6">
        <input type="text" class="form-control" placeholder="City" />
      </div>
      <div class="col-4">
        <select class="form-select">
          <option>State</option>
        </select>
      </div>
      <div class="col-2">
        <input type="text" class="form-control" placeholder="Zip" />
      </div>
    </div>
  </div>
</body>
```

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



The image shows three form controls arranged horizontally. From left to right: a text input field with the placeholder text 'City', a select dropdown menu with the text 'State' and a downward-pointing arrow, and another text input field with the placeholder text 'Zip'.

Placing Checkboxes and Radio Buttons Inline

By default, any number of custom checkboxes and radio buttons that are immediate sibling will be vertically stacked and appropriately spaced with `.form-check` class.

But, you can also place these custom checkboxes and radio buttons inline (i.e., in the same line) by simply adding the class `.form-check-inline` to `.form-check` element, like this:

```
<body>
  <div class="m-4">
    <h3 class="mb-4">Default Placement of Checkboxes</h3>
    <div class="row">
      <div class="col-12">
        <div class="form-check mb-3">
          <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkMusic"
          />
          <label class="form-check-label" for="checkMusic">Music</label>
        </div>
        <div class="form-check mb-3">
          <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkTravel"
            checked
          />
          <label class="form-check-label" for="checkTravel">Travel</label>
        </div>
        <div class="form-check">
          <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkReading"
            checked
          />
          <label class="form-check-label" for="checkReading">Reading</label>
        </div>
      </div>
    </div>
    <hr />
    <h3 class="mb-4">Inline Placement of Checkboxes</h3>
    <div class="row">
      <div class="col-12">
        <div class="form-check form-check-inline">
          <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkMusic"
          />
```

```

        />
        <label class="form-check-label" for="checkMusic">Music</label>
    </div>
    <div class="form-check form-check-inline ms-3">
        <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkTravel"
            checked
        />
        <label class="form-check-label" for="checkTravel">Travel</label>
    </div>
    <div class="form-check form-check-inline ms-3">
        <input
            type="checkbox"
            class="form-check-input"
            name="hobbies"
            id="checkReading"
            checked
        />
        <label class="form-check-label" for="checkReading">Reading</label>
    </div>
</div>
</div>
</div>
</body>

```

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

Music Travel Reading

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

```

<div class="row">
    <div class="col-12">
        <div class="form-check form-check-inline">
            <input
                type="radio"
                class="form-check-input"
                name="gender"
                id="radioMale"
                checked
            />

```

```
<label class="form-check-label" for="radioMale">Male</label>
</div>
<div class="form-check form-check-inline ms-3">
  <input
    type="radio"
    class="form-check-input"
    name="gender"
    id="radioFemale"
  />
  <label class="form-check-label" for="radioFemale">Female</label>
</div>
</div>
</div>
```

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

Male Female

Adding Help Text to Form Controls

Placing help text for the form controls in an efficient way to guide users to enter the correct data in a form. You can place block level help text for a form control using the class `.form-text`. The block help text is typically displayed at the bottom of the control. Here's an example:

```
<label class="form-label" for="inputPassword">Password</label>
<input type="password" class="form-control" id="inputPassword">
<div class="form-text">
  Must be 8-20 characters long, contain letters, numbers and special
  characters, but must not contain spaces.
</div>
```

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

Password

Must be 8-20 characters long, contain letters, numbers and special characters, but must not contain spaces.

Similarly, you can also place inline help text using the `<small>` element. No need to use `.form-text` in this case. The following example shows how to implement this:

```

<div class="row align-items-center g-3">
  <div class="col-auto">
    <label class="col-form-label" for="inputPassword">Password</label>
  </div>
  <div class="col-auto">
    <input type="password" class="form-control" id="inputPassword" />
  </div>
  <div class="col-auto">
    <span class="form-text">Must be 8-20 characters long.</span>
  </div>
</div>

```

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

Password Must be 8-20 characters long.

Height Sizing of Form Controls

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

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

```

<body>
  <div class="m-4">
    <div class="row mb-3">
      <label class="col-sm-2 col-form-label col-form-label-lg">Email</label>
      <div class="col-sm-10">
        <input
          type="email"
          class="form-control form-control-lg"
          placeholder="Large input"
        />
      </div>
    </div>
  </div>
  <div class="row mb-3">
    <label class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" placeholder="Default input" />
    </div>
  </div>

```

```

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

```

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

Email	<input type="text" value="Large input"/>
Email	<input type="text" value="Default input"/>
Email	<input type="text" value="Small input"/>

Tip: Be sure to apply the class `.col-form-label-lg` OR `.col-form-label-sm` ON the `<label>` or `<legend>` elements to correctly resize the label according to the form controls.

Similarly, you can create larger and smaller variants of the select boxes using the `.form-select-lg` and `.form-select-sm` classes on the `<select>` element, like this:

```

<body>
  <div class="m-4">
    <div class="row mb-3">
      <label class="col-sm-2 col-form-label col-form-label-lg">State</label>
      <div class="col-sm-10">
        <select class="form-select form-select-lg">
          <option>Large select</option>
        </select>
      </div>
    </div>
  </div>
  <div class="row mb-3">
    <label class="col-sm-2 col-form-label">State</label>
    <div class="col-sm-10">

```

```

        <select class="form-select">
          <option>Default select</option>
        </select>
      </div>
    </div>
    <div class="row">
      <label class="col-sm-2 col-form-label col-form-label-sm">State</label>
      <div class="col-sm-10">
        <select class="form-select form-select-sm">
          <option>Small select</option>
        </select>
      </div>
    </div>
  </div>
</body>

```

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

State	<input type="text" value="Large select"/>
State	<input type="text" value="Default select"/>
State	<input type="text" value="Small select"/>

Bootstrap Input Groups

Extending Form Controls with Bootstrap

Bootstrap input group component is a very flexible and powerful component for creating interactive and elegant form controls, however, it is limited to text input, select, and textarea only.

In the following sections you will see how to extend form controls by adding the text, icons and buttons before, after, or on both sides of it to make your form more attractive.


Creating Prepended and Appended Inputs

Input groups are created using the class `.input-group`. It act as a container for inputs and addons.

Further, wrap the text or icon in a `` element as well as apply the class `.input-group-text` on it and place it before or after the input. Let's take a look at the following example:

```
<body>
  <div class="m-4">
    <div class="row g-2">
      <div class="col-6">
        <div class="input-group">
          <span class="input-group-text">
            <span class="bi-person"></span>
          </span>
          <input type="text" class="form-control" placeholder="Username" />
        </div>
      </div>
      <div class="col-6">
        <div class="input-group">
          <input type="text" class="form-control" placeholder="Amount" />
          <span class="input-group-text">.00</span>
        </div>
      </div>
      <div class="col-6">
        <div class="input-group">
          <span class="input-group-text">https://www.</span>
          <input type="text" class="form-control" placeholder="Domain name" />
        </div>
      </div>
      <div class="col-6">
        <div class="input-group">
          <span class="input-group-text">$</span>
          <input type="text" class="form-control" placeholder="US Dollar" />
          <span class="input-group-text">.00</span>
        </div>
      </div>
    </div>
  </div>
</body>
```

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

 Username	Amount	.00
https://www. Domain name	\$ US Dollar	.00

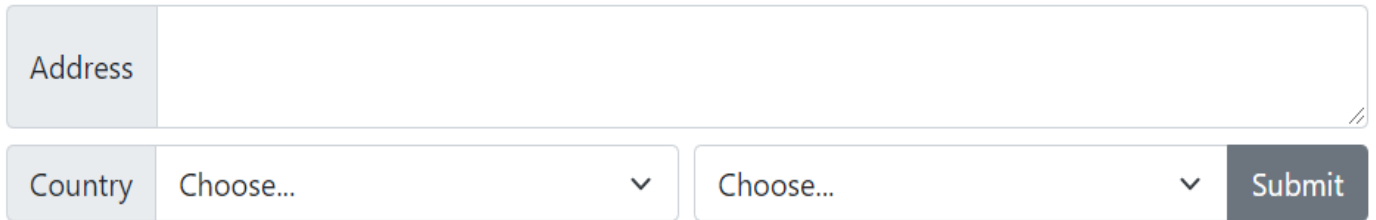
Since Bootstrap 5 you can also prepend or append select box dropdown and textarea form controls. Let's try out the following example and see how it basically works:

```

<body>
  <div class="m-4">
    <div class="row g-2">
      <div class="col-12">
        <div class="input-group">
          <span class="input-group-text">Address</span>
          <textarea class="form-control"></textarea>
        </div>
      </div>
      <div class="col-6">
        <div class="input-group">
          <label class="input-group-text">Country</label>
          <select class="form-select">
            <option selected>Choose...</option>
            <option>France</option>
            <option>Germany</option>
            <option>Hungary</option>
          </select>
        </div>
      </div>
      <div class="col-6">
        <div class="input-group">
          <select class="form-select">
            <option selected>Choose...</option>
            <option>One</option>
            <option>Two</option>
            <option>Three</option>
          </select>
          <button type="button" class="btn btn-secondary">Submit</button>
        </div>
      </div>
    </div>
  </div>
</body>

```

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



The image shows a Bootstrap form layout. It consists of a large text input field with the label 'Address' on the left. Below this, there are two dropdown menus, each with the label 'Country' and the text 'Choose...' and a downward arrow. To the right of the second dropdown menu is a dark grey button with the text 'Submit' in white.

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

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

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



The image shows a Bootstrap file input field. It features a 'Choose File' button on the left, the text 'No file chosen' in the center, and an 'Upload' button on the right.

//////////

Example:

```
<body>  
  <div class="m-4">  
    <h1 class="border-bottom pb-3 mb-4">Registration Form</h1>  
    <form  
      class="needs-validation"  
      action="/examples/actions/confirmation.php"  
      method="post"  
      novalidate  
    >  
      <div class="row mb-3">  
        <label class="col-sm-3 col-form-label" for="firstName">  
          >First Name:</label>  
        >  
        <div class="col-sm-9">  
          <input  
            type="text"  
            class="form-control"  
            id="firstName"  
            placeholder="First Name"  
            required  
          >
```

```
    />
  </div>
</div>
<div class="row mb-3">
  <label class="col-sm-3 col-form-label" for="lastName">Last Name:</label>
  <div class="col-sm-9">
    <input
      type="text"
      class="form-control"
      id="lastName"
      placeholder="Last Name"
      required
    />
  </div>
</div>
<div class="row mb-3">
  <label class="col-sm-3 col-form-label" for="emailAddress"
    >Email Address:</label
  >
  <div class="col-sm-9">
    <input
      type="email"
      class="form-control"
      id="emailAddress"
      placeholder="Email Address"
      required
    />
  </div>
</div>
<div class="row mb-3">
  <label class="col-sm-3 col-form-label" for="phoneNumber"
    >Mobile Number:</label
  >
  <div class="col-sm-9">
    <input
      type="number"
      class="form-control"
      id="phoneNumber"
      placeholder="Phone Number"
      required
    />
  </div>
</div>
<div class="row mb-3">
  <label class="col-sm-3 col-form-label">Date of Birth:</label>
  <div class="col-sm-3">
```

```
<select class="form-select" required>
  <option value="">Date</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
  <option value="9">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  <option value="16">16</option>
  <option value="17">17</option>
  <option value="18">18</option>
  <option value="19">19</option>
  <option value="20">20</option>
  <option value="21">21</option>
  <option value="22">22</option>
  <option value="23">23</option>
  <option value="24">24</option>
  <option value="25">25</option>
  <option value="26">26</option>
  <option value="27">27</option>
  <option value="28">28</option>
  <option value="29">29</option>
  <option value="30">30</option>
  <option value="31">31</option>
</select>
</div>
<div class="col-sm-3">
  <select class="form-select" required>
    <option value="">Month</option>
    <option value="1">January</option>
    <option value="2">February</option>
    <option value="3">March</option>
    <option value="4">April</option>
    <option value="5">May</option>
    <option value="6">June</option>
    <option value="7">July</option>
    <option value="8">August</option>
```

```

        <option value="9">September</option>
        <option value="10">October</option>
        <option value="11">November</option>
        <option value="12">December</option>
    </select>
</div>
<div class="col-sm-3">
    <select class="form-select">
        <option>Year</option>
    </select>
</div>
</div>
<div class="row mb-3">
    <label class="col-sm-3 col-form-label" for="postalAddress">
        Postal Address:</label>
    >
    <div class="col-sm-9">
        <textarea
            rows="3"
            class="form-control"
            id="postalAddress"
            placeholder="Postal Address"
            required
        ></textarea>
    </div>
</div>
<div class="row mb-3">
    <label class="col-sm-3 col-form-label" for="ZipCode">Zip Code:</label>
    <div class="col-sm-9">
        <input
            type="text"
            class="form-control"
            id="ZipCode"
            placeholder="Zip Code"
            required
        />
    </div>
</div>
<div class="row mb-3">
    <label class="col-sm-3 col-form-label" for="uploadImage">
        Upload Image:</label>
    >
    <div class="col-sm-9">
        <input type="file" class="form-control" id="uploadImage" />
    </div>
</div>

```

```

<div class="row mb-3">
  <label class="col-sm-3 col-form-label">Gender:</label>
  <div class="col-sm-9 mt-2">
    <div class="form-check form-check-inline">
      <input
        type="radio"
        class="form-check-input"
        name="gender"
        id="radioMale"
      />
      <label class="form-check-label" for="radioMale">Male</label>
    </div>
    <div class="form-check form-check-inline">
      <input
        type="radio"
        class="form-check-input"
        name="gender"
        id="radioFemale"
      />
      <label class="form-check-label" for="radioFemale">Female</label>
    </div>
  </div>
</div>
<div class="row mb-3">
  <div class="col-sm-9 offset-sm-3">
    <div class="form-check">
      <input
        class="form-check-input"
        type="checkbox"
        id="checkAgree"
        value="agree"
      />
      <label class="form-check-label" for="checkAgree"
        >I agree to the <a href="#">Terms and Conditions</a></label>
    </div>
  </div>
</div>
<div class="row mb-3">
  <div class="col-sm-9 offset-sm-3">
    <input type="submit" class="btn btn-primary" value="Submit" />
    <input type="reset" class="btn btn-secondary ms-2" value="Reset" />
  </div>
</div>
</form>
</div>

```

</body>

Registration Form

First Name:

Last Name:

Email Address:

Mobile Number:

Date of Birth:

Postal Address:

Zip Code:

Upload Image:

Gender:

Male Female

I agree to the [Terms and Conditions](#)

Bootstrap Buttons

Creating Buttons with Bootstrap

Buttons are the integral part of a website and application. They are used for various purposes like, submit or reset an [HTML form](#), performing interactive actions such as showing or hiding something on a web page on click of the

button, redirecting user to another page, and so on. Bootstrap provides a quick and easy way to create and customize the buttons.

Bootstrap Button Styles

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

The following example will show you how to create different styles of buttons in Bootstrap:

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

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



Bootstrap Outline Buttons

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

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

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



Changing the Sizes of Buttons

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

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

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

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



Similarly, to make buttons smaller add an extra class `.btn-sm` to the buttons, like this:

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

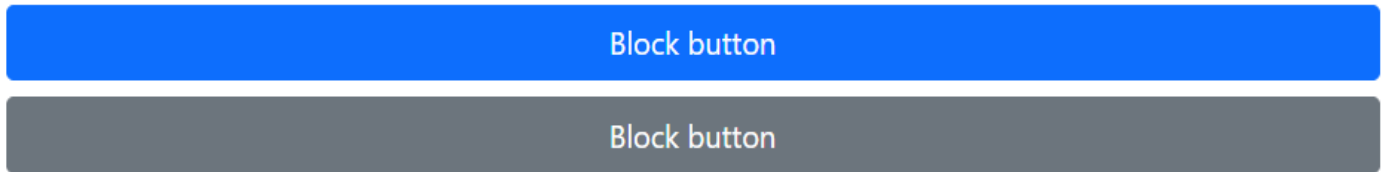
— The output of the above example will look something like this:



You can also create full-width or block buttons (buttons that covers the full width of the parent elements) through using the Bootstrap's display and gap utility classes. These utilities offers much greater control over spacing, alignment, and responsive behaviors.

```
<div class="d-grid gap-2">  
  <button type="button" class="btn btn-primary">Block  
button</button>  
  <button type="button" class="btn btn-secondary">Block  
button</button>  
</div>
```

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



Bootstrap Disabled Buttons

Sometimes we need to disable a button for certain reasons, for example, a user in case is not eligible to perform this particular action, or we want to ensure that user should performed all other required actions before proceed to this particular action. Let's see how to do that.

Creating Disabled Buttons Using Button and Input Element

Buttons created through `<button>` or `<input>` tag can be disabled by adding the `disabled` attribute to the respective element, as shown in the following example:

```
<button type="button" class="btn btn-primary disabled">Primary  
button</button>  
<button type="button" class="btn btn-secondary disabled">Secondary  
button</button>
```

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



Creating Disabled Buttons Using Anchor Elements

Buttons created through `<a>` tag can be disabled by adding the class `.disabled`, like this:

```
<a href="#" class="btn btn-primary disabled">Primary link</a>  
<a href="#" class="btn btn-secondary disabled">Secondary link</a>
```

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



Note: The `.disabled` class only make links visually appear like disabled, however the link will remain clickable unless you remove the `href` attribute from it.

Bootstrap Button Groups

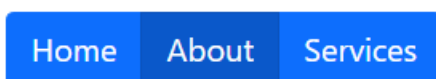
Creating Button Groups with Bootstrap

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

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

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

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



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

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

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

Mixed Styles Button Groups

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

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

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



Creating Button Toolbar

You can also combine sets of button groups together for creating more complex components like button toolbar. To create button toolbar just wrap sets of button groups in a `<div>` element and apply the class `.btn-toolbar` on it, as shown in the following example:

```
<div class="btn-toolbar">
  <div class="btn-group me-2">
    <button type="button" class="btn btn-primary">
      <i class="bi-fonts"></i>
    </button>
    <button type="button" class="btn btn-primary">
      <i class="bi-type-bold"></i>
    </button>
    <button type="button" class="btn btn-primary">
      <i class="bi-type-italic"></i>
    </button>
  </div>
</div>
```

```

    <button type="button" class="btn btn-primary">
      <i class="bi-type-underline"></i>
    </button>
  </div>
  <div class="btn-group me-2">
    <button type="button" class="btn btn-primary">
      <i class="bi-text-left"></i>
    </button>
    <button type="button" class="btn btn-primary">
      <i class="bi-text-center"></i>
    </button>
    <button type="button" class="btn btn-primary">
      <i class="bi-text-right"></i>
    </button>
    <button type="button" class="btn btn-primary">
      <i class="bi-justify"></i>
    </button>
  </div>
  <div class="btn-group">
    <button type="button" class="btn btn-primary">
      <i class="bi-code"></i>
    </button>
  </div>
</div>

```

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



Height Sizing of Button Groups

Instead of applying button sizing classes to each button in a group, you can simply apply button group sizing classes like `.btn-group-lg` or `.btn-group-sm` directly to each `.btn-group` element to create larger or smaller button groups, as shown in the example below:

```

<!-- Large button group -->
<div class="btn-group mb-2 btn-group-lg">
  <button type="button" class="btn btn-primary">Home</button>
  <button type="button" class="btn btn-primary">About</button>
  <button type="button" class="btn btn-primary">Services</button>
</div>
<br />

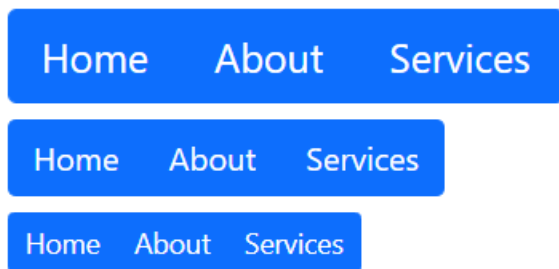
```

```

<!-- Default button group -->
<div class="btn-group mb-2">
  <button type="button" class="btn btn-primary">Home</button>
  <button type="button" class="btn btn-primary">About</button>
  <button type="button" class="btn btn-primary">Services</button>
</div>
<br />
<!-- Small button group -->
<div class="btn-group btn-group-sm">
  <button type="button" class="btn btn-primary">Home</button>
  <button type="button" class="btn btn-primary">About</button>
  <button type="button" class="btn btn-primary">Services</button>
</div>

```

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



Nesting Button Groups

Button groups can also be nested. The following example demonstrates how to place a `.btn-group` within another `.btn-group` to create dropdown menus inside button groups.

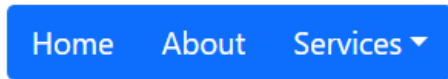
```

<div class="btn-group">
  <a href="#" class="btn btn-primary">Home</a>
  <a href="#" class="btn btn-primary">About</a>
  <div class="btn-group">
    <a
      href="#"
      class="btn btn-primary dropdown-toggle"
      data-bs-toggle="dropdown"
      >Services</a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Web Design</a>
      <a class="dropdown-item" href="#">Web Development</a>
    </div>
  </div>

```

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

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

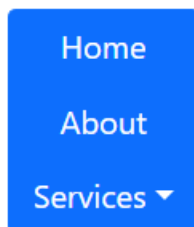


Vertically Stacked Button Groups

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

```
<div class="btn-group-vertical">
  <a href="#" class="btn btn-primary">Home</a>
  <a href="#" class="btn btn-primary">About</a>
  <div class="btn-group">
    <a
      href="#"
      class="btn btn-primary dropdown-toggle"
      data-bs-toggle="dropdown"
      >Services</a>
    >
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Web Design</a>
      <a class="dropdown-item" href="#">Web Development</a>
    </div>
  </div>
</div>
```

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

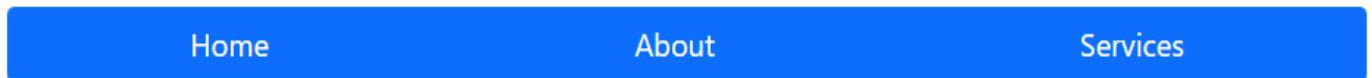


Creating Justified Button Groups

You can also stretch your button groups to fill all the available width by applying the flex utility class `.d-flex` to the `.btn-group` element. Every button has equal width in this case.

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

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



Bootstrap Cards

Using the Bootstrap Cards

Bootstrap card is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. Card replaces panel, well, and thumbnail components in old Bootstrap 3 version.

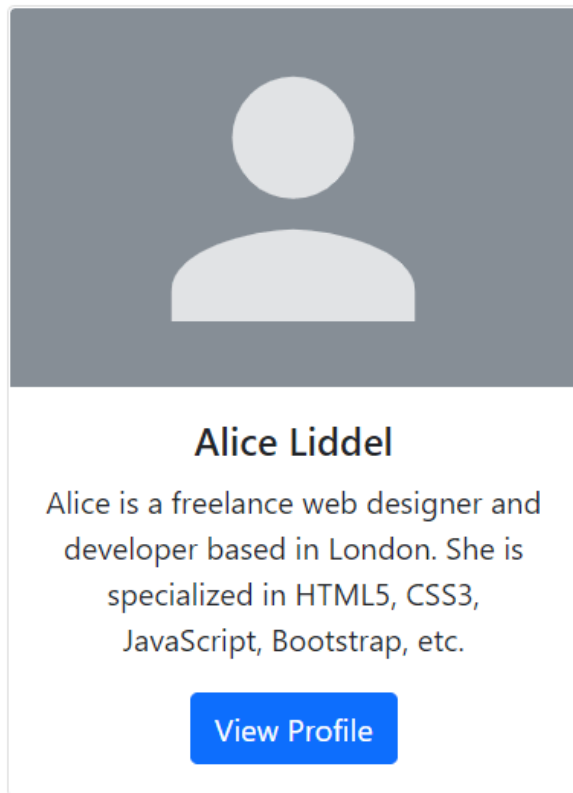
In the following sections, you will see what you can do with the card component.

Creating a Basic Card

The card markup is pretty straight forward. The outer wrapper require the base class `.card`, whereas content can be placed inside the `.card-body` element. The following example will show you how to create a card with a picture, mixed with some text content and a button.


```
<div class="card" style="width: 300px;">
  
  <div class="card-body text-center">
    <h5 class="card-title">Alice Liddel</h5>
    <p class="card-text">Alice is a freelance web designer and
developer based in London. She is specialized in HTML5, CSS3,
JavaScript, Bootstrap, etc.</p>
    <a href="#" class="btn btn-primary">View Profile</a>
  </div>
</div>
```

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



Note: Card doesn't have fixed width, they'll occupy the full width of its parent element. So, if you need a fixed width card you need to apply the width property on card yourself. Also, card have no margin by default,

Content Types for Card Component

The card component support a wide variety of content, including images, text, list groups, links, navs, and more. Here are the examples of what's supported by the card.

Body Only Card

You can simply use `.card` with `.card-body` within, whenever you need to create a padded box.

```
<div class="card">
  <div class="card-body">This is some text within a padded
  box.</div>
</div>
```

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

This is some text within a padded box.

Card with Titles, Text, and Links

Further, you can also place title and links inside the card along with text, like this:

```
<div class="card" style="width: 300px;">
  <div class="card-body">
    <h5 class="card-title">Eiffel Tower</h5>
    <h6 class="card-subtitle mb-3 text-muted">Champ de Mars,
    Paris, France</h6>
    <p class="card-text">Built in 1889 Eiffel Tower is one of the
    most iconic landmarks in the world.</p>
    <a href="#" class="card-link">View pictures</a>
    <a href="#" class="card-link">Discover history</a>
  </div>
</div>
```

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

Eiffel Tower

Champ de Mars, Paris, France

Built in 1889 Eiffel Tower is one of the most iconic landmarks in the world.

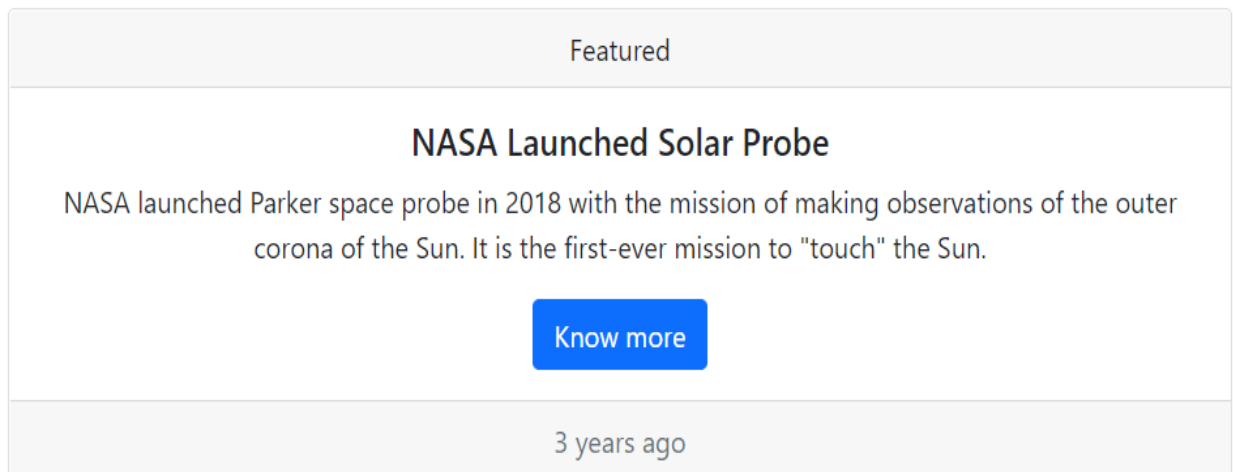
[View pictures](#) [Discover history](#)

Card with Header and Footer

You can also add header and footer within your cards using the `.card-header` and `.card-footer` class, respectively. Let's take a look at the following example:

```
<div class="card text-center">
  <div class="card-header">Featured</div>
  <div class="card-body">
    <h5 class="card-title">NASA Launched Solar Probe</h5>
    <p class="card-text">NASA launched Parker space probe in 2018
with the mission of making observations of the outer corona of the
Sun. It is the first-ever mission to "touch" the Sun.</p>
    <a href="#" class="btn btn-primary">Know more</a>
  </div>
  <div class="card-footer text-muted">3 years ago</div>
</div>
```

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

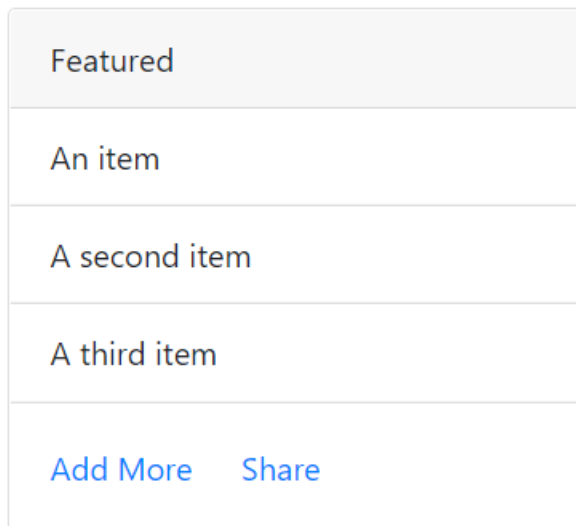


Placing List Groups within Card

You can also place list groups inside the card along with other content types, as shown here.

```
<div class="card" style="width: 300px;">
  <div class="card-header">Featured</div>
  <ul class="list-group list-group-flush">
    <li class="list-group-item">An item</li>
    <li class="list-group-item">A second item</li>
    <li class="list-group-item">A third item</li>
  </ul>
  <div class="card-body">
    <a href="#" class="card-link">Add More</a>
    <a href="#" class="card-link">Share</a>
  </div>
</div>
```

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



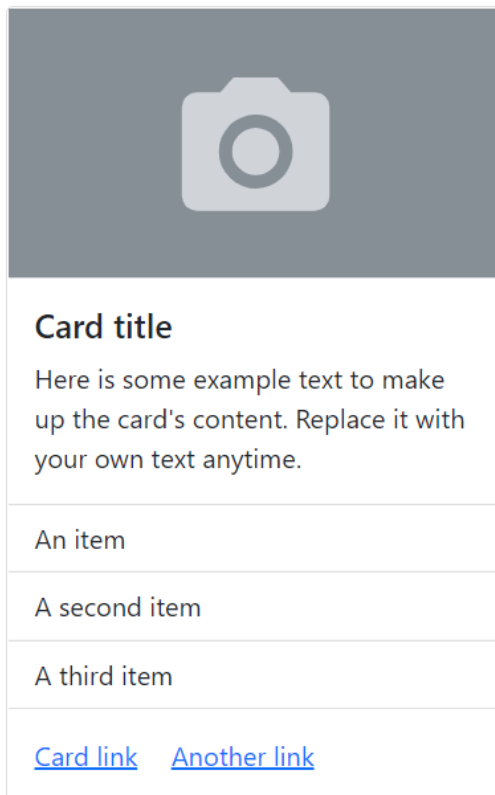
Mix and Match Multiple Content Types within Card

Feel free to mix and match multiple content types to create the card you need. The following example will create a fixed-width card with an image, text, list group, and hyperlinks.

```
<div class="card" style="width: 300px;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
```

```
<p class="card-text">Here is some example text to make up the
card's content. Replace it with your own text anytime.</p>
</div>
<ul class="list-group list-group-flush">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<div class="card-body">
  <a href="#" class="card-link">Card link</a>
  <a href="#" class="card-link">Another link</a>
</div>
</div>
```

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



Adding Navigation to Cards

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

To add tabs navigation to a card simply place the tabs markup inside the card header, and the tabs content inside the card body. You are also required to use

an additional class `.card-header-tabs` on the `.nav` element along with the class `.nav-tabs` for proper alignment.

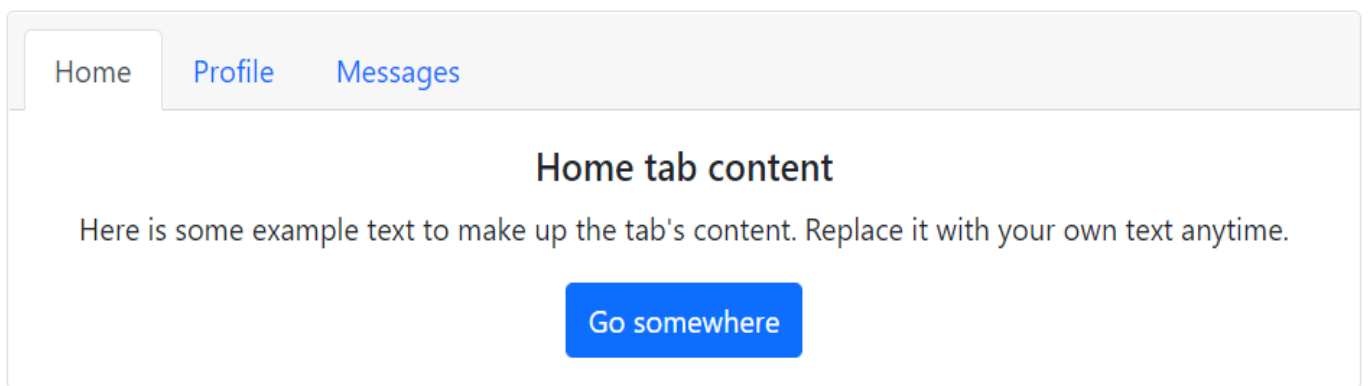
Let's try out the following example which creates an elegant tabbed navigation.

```
<body>
  <div class="m-4">
    <div class="card text-center">
      <div class="card-header">
        <ul class="nav nav-tabs card-header-tabs" id="myTab">
          <li class="nav-item">
            <a href="#home" class="nav-link active" data-bs-toggle="tab"
              >Home</a>
          </li>
          <li class="nav-item">
            <a href="#profile" class="nav-link" data-bs-toggle="tab">Profile</a>
          </li>
          <li class="nav-item">
            <a href="#messages" class="nav-link" data-bs-toggle="tab"
              >Messages</a>
          </li>
        </ul>
      </div>
      <div class="card-body">
        <div class="tab-content">
          <div class="tab-pane fade show active" id="home">
            <h5 class="card-title">Home tab content</h5>
            <p class="card-text">
              Here is some example text to make up the tab's content. Replace it
              with your own text anytime.
            </p>
            <a href="#" class="btn btn-primary">Go somewhere</a>
          </div>
          <div class="tab-pane fade" id="profile">
            <h5 class="card-title">Profile tab content</h5>
            <p class="card-text">
              Here is some example text to make up the tab's content. Replace it
              with your own text anytime.
            </p>
            <a href="#" class="btn btn-primary">Go somewhere</a>
          </div>
          <div class="tab-pane fade" id="messages">
            <h5 class="card-title">Messages tab content</h5>
            <p class="card-text">
```

Here is some example text to make up the tab's content. Replace it with your own text anytime.

```
</p>
<a href="#" class="btn btn-primary">Go somewhere</a>
</div>
</div>
</div>
</div>
</div>
</div>
</body>
```

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



Similarly, you can add pills nav to the card by using an additional class `.card-header-pills` along with the class `.nav-pills` on the `.nav` element, as shown below:

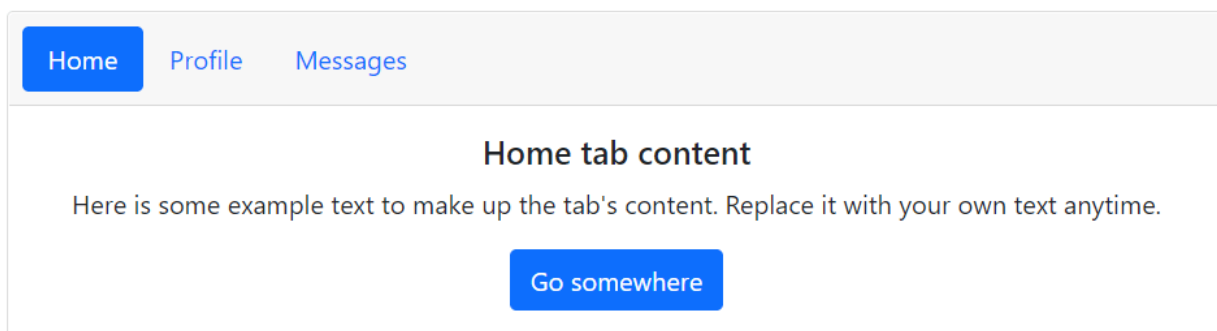
```
<body>
<div class="m-4">
  <div class="card text-center">
    <div class="card-header">
      <ul class="nav nav-pills card-header-pills">
        <li class="nav-item">
          <a href="#home" class="nav-link active" data-bs-toggle="tab"
            >Home</a>
        </li>
        <li class="nav-item">
          <a href="#profile" class="nav-link" data-bs-toggle="tab">Profile</a>
        </li>
        <li class="nav-item">
          <a href="#messages" class="nav-link" data-bs-toggle="tab"
            >Messages</a>
        </li>
      </ul>
    </div>
  </div>
</div>
```

```

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

```

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



Customizing the Card Styles

There are several options available for customizing the card's backgrounds, borders, and color.

Customizing Background and Color

You can simply use the [background](#) and [color](#) utility classes to change the appearance of a card. Let's try out the following example to understand how it basically works:

```
<body>
  <div class="m-4">
    <div class="row">
      <div class="col-6">
        <div class="card text-white bg-primary mb-4">
          <div class="card-body">
            <h5 class="card-title">Primary card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
        </div>
      </div>
      <div class="col-6">
        <div class="card text-white bg-secondary mb-4">
          <div class="card-body">
            <h5 class="card-title">Secondary card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
        </div>
      </div>
      <div class="col-6">
        <div class="card text-white bg-success mb-4">
          <div class="card-body">
            <h5 class="card-title">Success card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
        </div>
      </div>
    </div>
  </div>
```

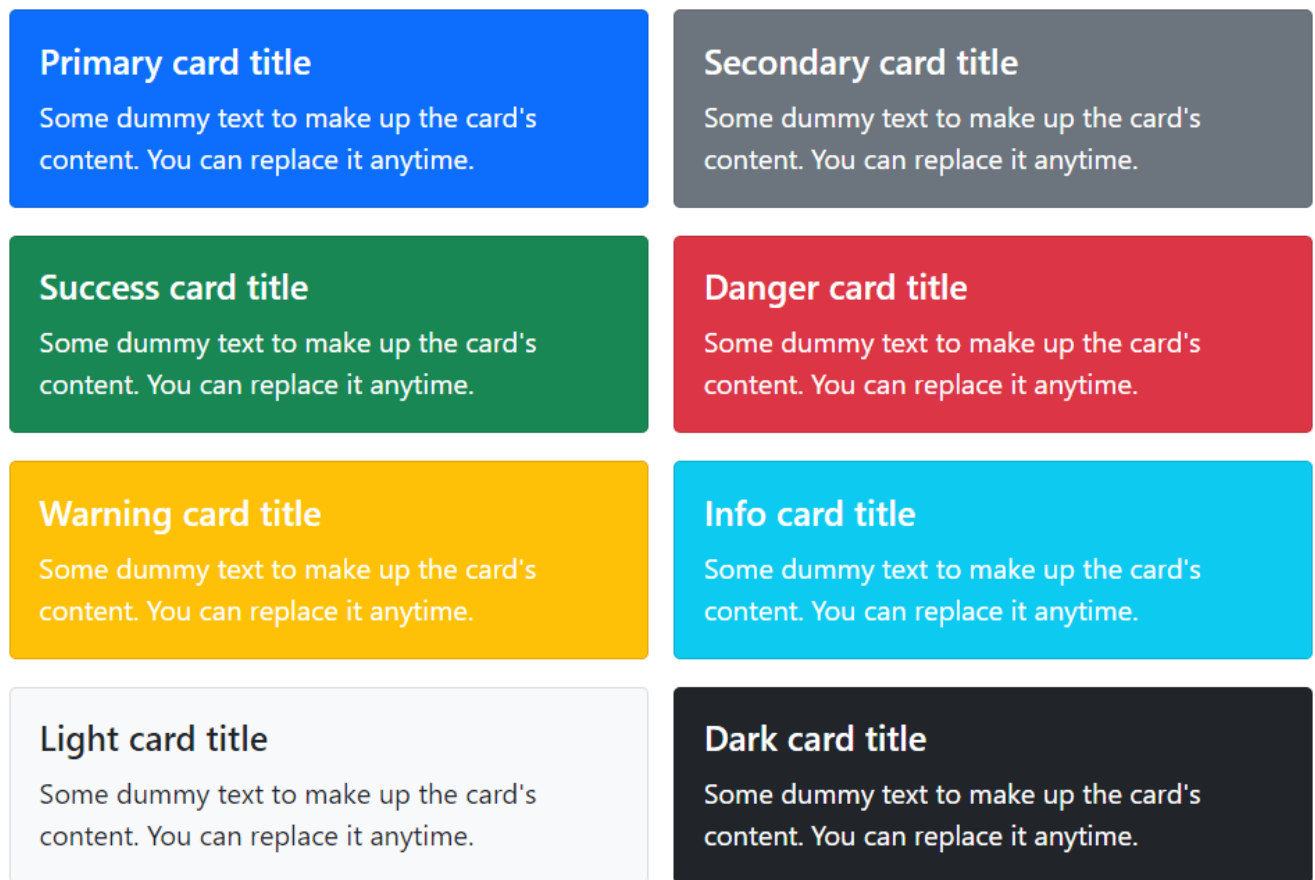
```
</div>
</div>
<div class="col-6">
  <div class="card text-white bg-danger mb-4">
    <div class="card-body">
      <h5 class="card-title">Danger card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
    </div>
  </div>
</div>
<div class="col-6">
  <div class="card text-white bg-warning mb-4">
    <div class="card-body">
      <h5 class="card-title">Warning card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
    </div>
  </div>
</div>
<div class="col-6">
  <div class="card text-white bg-info mb-4">
    <div class="card-body">
      <h5 class="card-title">Info card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
    </div>
  </div>
</div>
<div class="col-6">
  <div class="card text-white bg-dark">
    <div class="card-body">
      <h5 class="card-title">Dark card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
    </div>
  </div>
</div>
</div>
```

```

<div class="col-6">
  <div class="card bg-light">
    <div class="card-body">
      <h5 class="card-title">Light card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
    </div>
  </div>
</div>
</div>
</div>
</div>
</body>

```

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



Card Layout Options

In addition to styling of the cards, Bootstrap also includes a few options for laying out the series of cards. However, these layouts are not responsive yet.

Creating the Card Groups

You can use card groups to render cards as a single, attached element with equal width and height columns. However, cards inside a card group become horizontally stacked on extra small devices (i.e. viewport width <576px). Let's try out an example and see how it actually works:

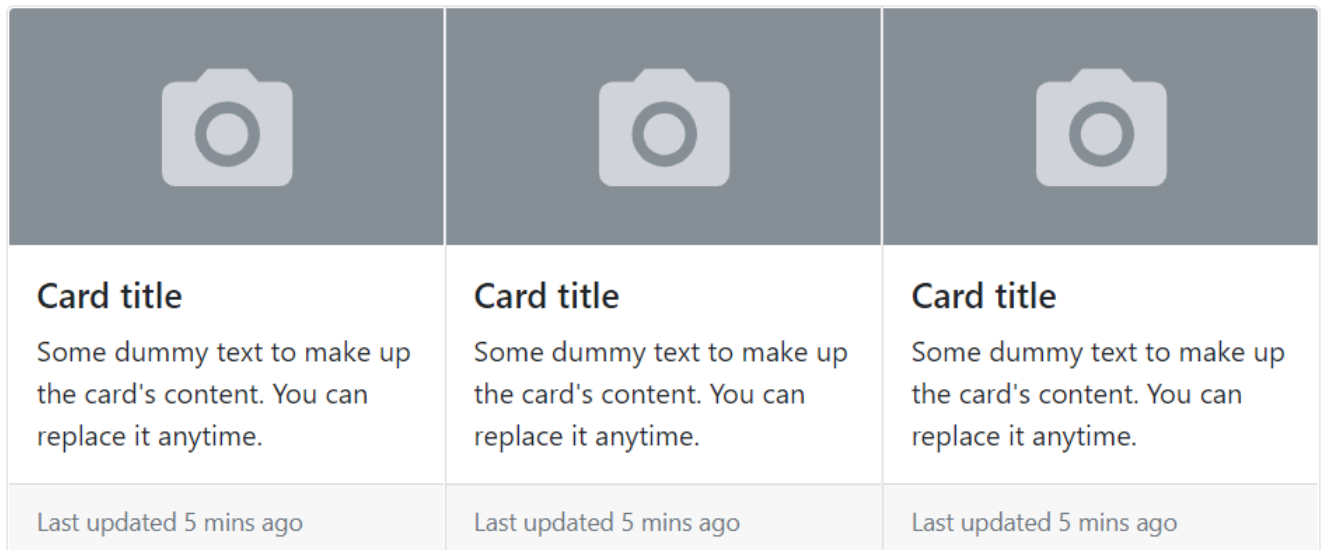
```
<body>
  <div class="m-4">
    <div class="card-group">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
          <p class="card-text">
            Some dummy text to make up the card's content. You can replace it
            anytime.
          </p>
        </div>
        <div class="card-footer">
          <small class="text-muted">Last updated 5 mins ago</small>
        </div>
      </div>
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title">Card title</h5>
          <p class="card-text">
            Some dummy text to make up the card's content. You can replace it
            anytime.
          </p>
        </div>
        <div class="card-footer">
          <small class="text-muted">Last updated 5 mins ago</small>
        </div>
      </div>
    </div>
  </div>
</body>
```

```


<div class="card-body">
  <h5 class="card-title">Card title</h5>
  <p class="card-text">
    Some dummy text to make up the card's content. You can replace it
    anytime.
  </p>
</div>
<div class="card-footer">
  <small class="text-muted">Last updated 5 mins ago</small>
</div>
</div>
</div>
</div>
</body>

```

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



Creating the Card Grids

You can use the Bootstrap grid system and its `.row-cols-*` classes to control how many grid columns (wrapped around your cards) to show per row. For example, you can use the class `.row-cols-1` to show one card per row, similarly you can use the class `.row-cols-md-2` to show two cards per row, from the medium breakpoint up (i.e. viewport width $\geq 768\text{px}$).

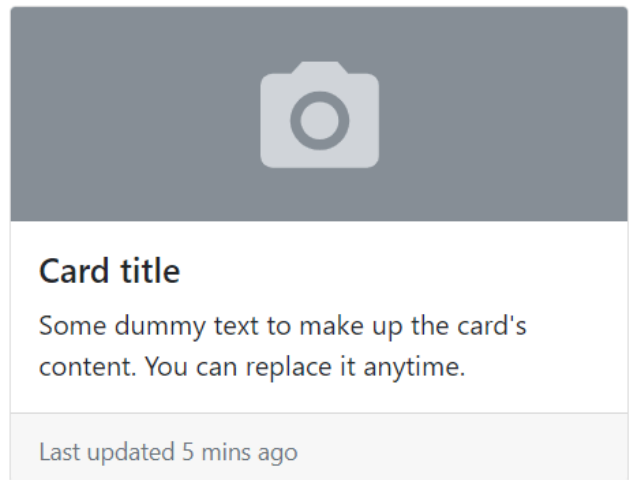
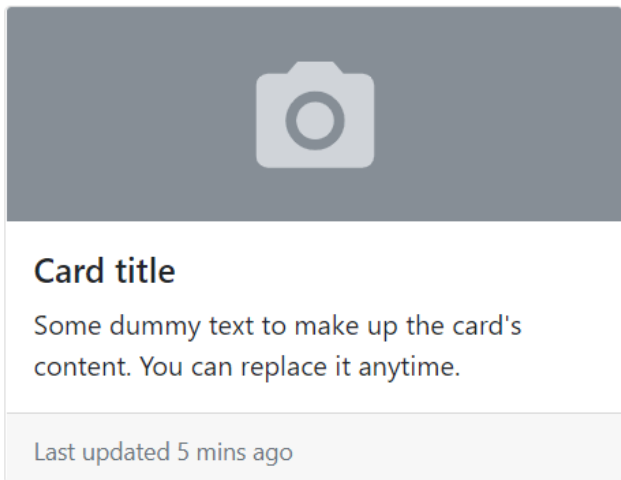
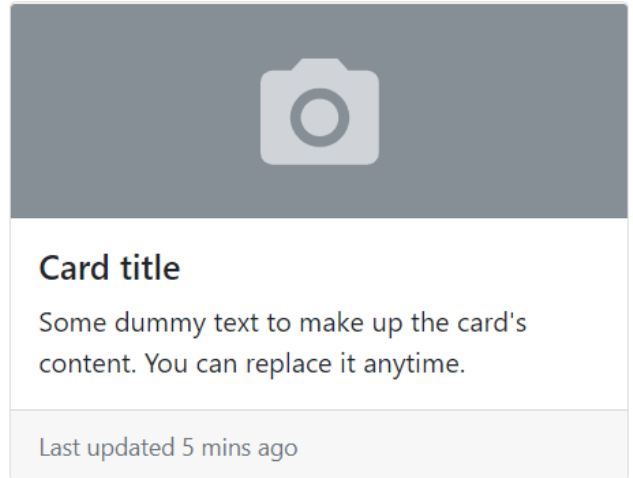
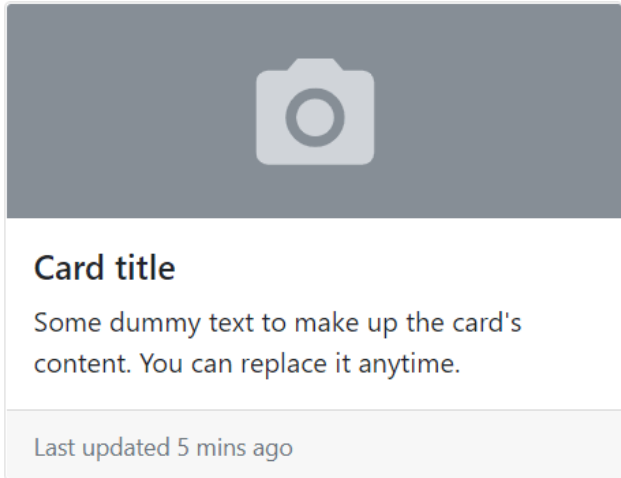
```
<body>
  <div class="m-4">
    <div class="row row-cols-1 row-cols-md-2 g-4">
      <div class="col">
        <div class="card">
          
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
          <div class="card-footer">
            <small class="text-muted">Last updated 5 mins ago</small>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="card">
          
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
          <div class="card-footer">
            <small class="text-muted">Last updated 5 mins ago</small>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="card">
          
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
          </div>
          <div class="card-footer">
            <small class="text-muted">Last updated 5 mins ago</small>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```

        alt="..."
    />
    <div class="card-body">
        <h5 class="card-title">Card title</h5>
        <p class="card-text">
            Some dummy text to make up the card's content. You can replace it
            anytime.
        </p>
    </div>
    <div class="card-footer">
        <small class="text-muted">Last updated 5 mins ago</small>
    </div>
</div>
<div class="col">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
                Some dummy text to make up the card's content. You can replace it
                anytime.
            </p>
        </div>
        <div class="card-footer">
            <small class="text-muted">Last updated 5 mins ago</small>
        </div>
    </div>
</div>
</div>
</div>
</body>

```

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



Creating Horizontal Cards

You can also create horizontal cards where image and text content are placed side-by-side using a combination of grid and utility classes, as shown in the following example:

```
<body>
  <div class="m-4">
    <div class="card" style="max-width: 500px">
      <div class="row g-0">
        <div class="col-sm-5" style="background: #868e96">
          
        </div>
      </div>
    </div>
  </div>
```

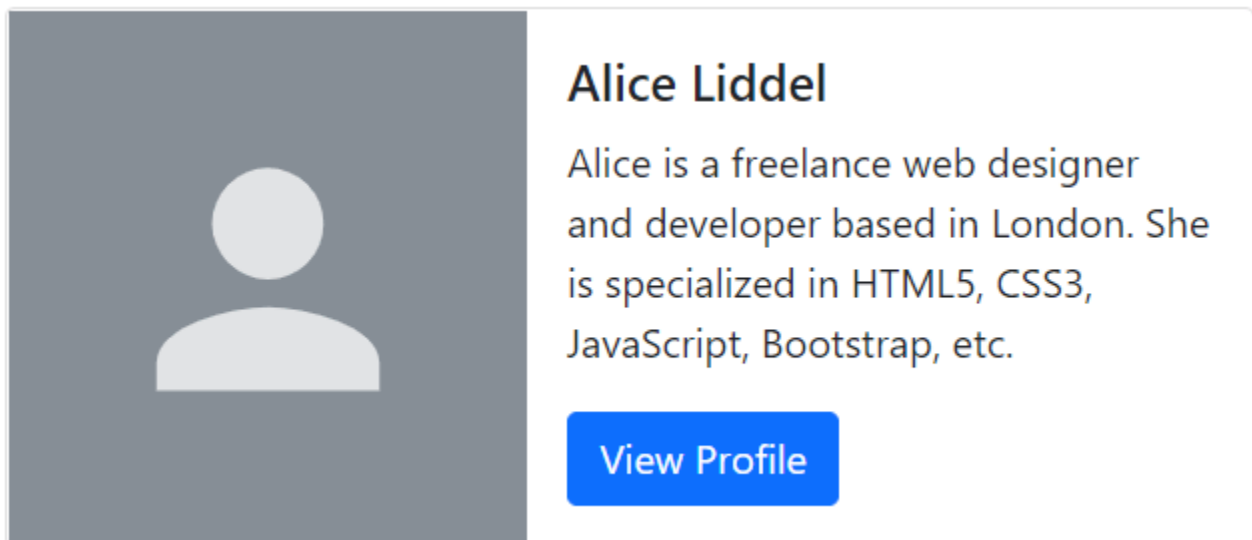


```

<div class="col-sm-7">
  <div class="card-body">
    <h5 class="card-title">Alice Liddel</h5>
    <p class="card-text">
      Alice is a freelance web designer and developer based in London.
      She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.
    </p>
    <a href="#" class="btn btn-primary stretched-link">View Profile</a>
  </div>
</div>
</div>
</div>
</div>
</body>

```

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



Card Image Overlays

You can even turn an image into a card background and place the card's text on the top it using the class `.card-img-overlay` in place of `.card-body`. Depending on the image, you may need additional styles for better adjustments. Let's check out an example:

```

<body>
  <div class="m-4">
    <div class="card text-white" style="width: 350px">
      
      <div class="card-img-overlay">

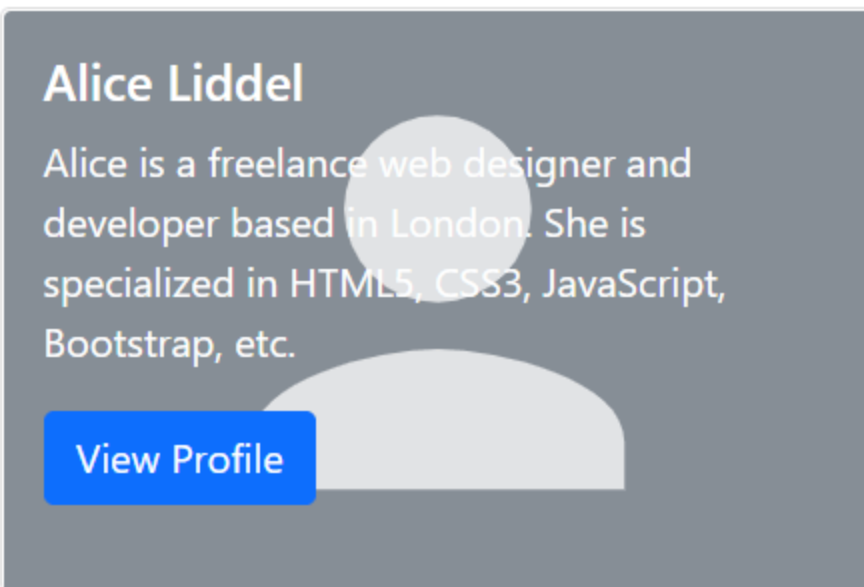
```

```

<h5 class="card-title">Alice Liddel</h5>
<p class="card-text">
  Alice is a freelance web designer and developer based in London. She
  is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.
</p>
<a href="#" class="btn btn-primary stretched-link">View Profile</a>
</div>
</div>
</div>
</body>

```

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



Text Alignment inside Card

You can easily change the text alignment of any card—entirely or specific parts—with the text alignment utility classes. For example, you can use the class `.text-center` and `.text-end` to align the card's text content to the center and to the right end, respectively.

```

<body>
  <div class="m-4">
    <div class="row row-cols-1 row-cols-md-3 g-3">
      <div class="col">
        <!-- Card with default left text alignment -->
        <div class="card">
          <div class="card-body">
            <h5 class="card-title">Card title</h5>

```

```

    <p class="card-text">
      Some dummy text to make up the card's content. You can replace it
      anytime.
    </p>
    <a href="#" class="btn btn-primary">Know more</a>
  </div>
</div>
</div>
<div class="col">
  <!-- Card with center text alignment -->
  <div class="card text-center">
    <div class="card-body">
      <h5 class="card-title">Card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
      <a href="#" class="btn btn-primary">Know more</a>
    </div>
  </div>
</div>
<div class="col">
  <!-- Card with right text alignment -->
  <div class="card text-end">
    <div class="card-body">
      <h5 class="card-title">Card title</h5>
      <p class="card-text">
        Some dummy text to make up the card's content. You can replace it
        anytime.
      </p>
      <a href="#" class="btn btn-primary">Know more</a>
    </div>
  </div>
</div>
</div>
</div>
</body>

```

Specifying Card Size

Cards have no specific width, they are 100% wide by default. However, you can change this as needed with custom CSS, grid classes, or sizing utility classes.

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

```
<body>
  <div class="m-4">
    <div class="row row-cols-1 row-cols-md-3 g-3">
      <div class="col">
        <!-- Card with default left text alignment -->
        <div class="card">
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
            <a href="#" class="btn btn-primary">Know more</a>
          </div>
        </div>
      </div>
      <div class="col">
        <!-- Card with center text alignment -->
        <div class="card text-center">
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
            <a href="#" class="btn btn-primary">Know more</a>
          </div>
        </div>
      </div>
      <div class="col">
        <!-- Card with right text alignment -->
        <div class="card text-end">
          <div class="card-body">
            <h5 class="card-title">Card title</h5>
            <p class="card-text">
              Some dummy text to make up the card's content. You can replace it
              anytime.
            </p>
            <a href="#" class="btn btn-primary">Know more</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Card title

Some dummy text to make up the card's content. You can replace it anytime.

[Know more](#)

Card title

Some dummy text to make up the card's content. You can replace it anytime.

[Know more](#)

Card title

Some dummy text to make up the card's content. You can replace it anytime.

[Know more](#)

Card title

Some dummy text to make up the card's content. You can replace it anytime.

[Know more](#)

Bootstrap Navbar

Creating Navbar with Bootstrap

You can use the Bootstrap navbar component to create responsive navigation header for your website or application. These responsive navbar will be collapsed on devices having small viewports like mobile phones but expand when user click the toggle button. However, it will be horizontal as normal on the medium and large devices such as laptop or desktop.

You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort. The following example will show you how to create a simple static navbar with navigation links.

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Brand</a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>
        <a href="#" class="nav-item nav-link">Profile</a>
        <a href="#" class="nav-item nav-link">Messages</a>
        <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
      </div>
      <div class="navbar-nav ms-auto">
        <a href="#" class="nav-item nav-link">Login</a>
      </div>
    </div>
  </div>
</nav>

```

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



Tip: Navbars require a wrapping `.navbar` with `.navbar-expand{-sm|-md|-lg|-xl|-xxl}` for responsive collapsing, and color scheme classes for appearance. Also, use the spacing and flex utility classes for controlling spacing and alignment within navbars.

Note: Navbars and their contents are fluid by default. Change the [container](#) (e.g. `.container{-sm|-md|-lg|-xl|-xxl}`) to limit their horizontal width in different ways.

Adding Logo Images to Navbars

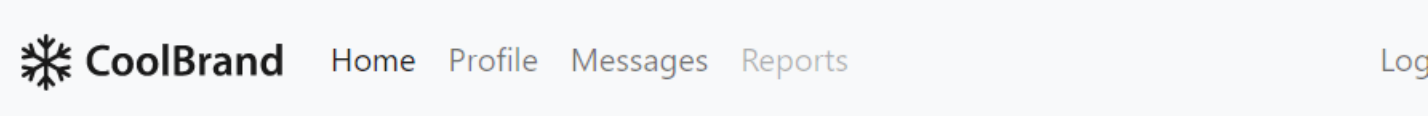
You can also place your logo image inside the navbar, instead of plain text. However, you need to set the logo height manually to fit it properly inside the navbar, as shown here:

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">
      
    </a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>
        <a href="#" class="nav-item nav-link">Profile</a>
        <a href="#" class="nav-item nav-link">Messages</a>
        <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
      </div>
      <div class="navbar-nav ms-auto">
        <a href="#" class="nav-item nav-link">Login</a>
      </div>
    </div>
  </div>
</nav>

```

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



CoolBrand Home Profile Messages Reports Login

Tip: Use the [utility classes](#) such as `.ms-auto`, `.me-auto`, `.justify-content-between`, etc. to align the nav links, forms, buttons or text inside the navbar.

Adding Dropdowns to Navbar

You can also include dropdowns and search box within the navbars, as shown here:

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Brand</a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">

```

```

    <div class="navbar-nav">
      <a href="#" class="nav-item nav-link active">Home</a>
      <a href="#" class="nav-item nav-link">Profile</a>
      <div class="nav-item dropdown">
        <a href="#" class="nav-link dropdown-toggle" data-
bs-toggle="dropdown">Messages</a>
        <div class="dropdown-menu">
          <a href="#" class="dropdown-item">Inbox</a>
          <a href="#" class="dropdown-item">Sent</a>
          <a href="#" class="dropdown-item">Drafts</a>
        </div>
      </div>
      <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
    </div>
    <div class="navbar-nav ms-auto">
      <a href="#" class="nav-item nav-link">Login</a>
    </div>
  </div>
</div>
</nav>

```

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

```

Brand Home Profile Messages ▾ Reports Login

```

Placing Search Form inside Navbar

Search form is a very common component of the navbars and you have seen it on various website quite often. You can place various form controls within a navbar using the class `.d-flex` on the `<form>` element, as demonstrated in the following example:

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Brand</a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse justify-content-between"
id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>

```

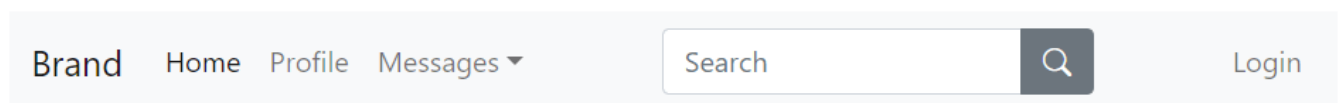


```

    <a href="#" class="nav-item nav-link">Profile</a>
    <div class="nav-item dropdown">
      <a href="#" class="nav-link dropdown-toggle" data-
bs-toggle="dropdown">Messages</a>
      <div class="dropdown-menu">
        <a href="#" class="dropdown-item">Inbox</a>
        <a href="#" class="dropdown-item">Sent</a>
        <a href="#" class="dropdown-item">Drafts</a>
      </div>
    </div>
  </div>
  <form class="d-flex">
    <div class="input-group">
      <input type="text" class="form-control"
placeholder="Search">
      <button type="button" class="btn btn-secondary"><i
class="bi-search"></i></button>
    </div>
  </form>
  <div class="navbar-nav">
    <a href="#" class="nav-item nav-link">Login</a>
  </div>
</div>
</div>
</nav>

```

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



Changing the Color Scheme of Navbars

You can also change the color scheme of the navbar by using the `.navbar-light` for the light background colors, or `.navbar-dark` for the dark background colors. Then, customize it with the [background color utility classes](#), such as `.bg-dark`, `.bg-primary`, and so on.

Alternatively, you can also apply the CSS `background-color` property on the `.navbar` element yourself to customize the navbar theme, as shown in the following example:

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

```

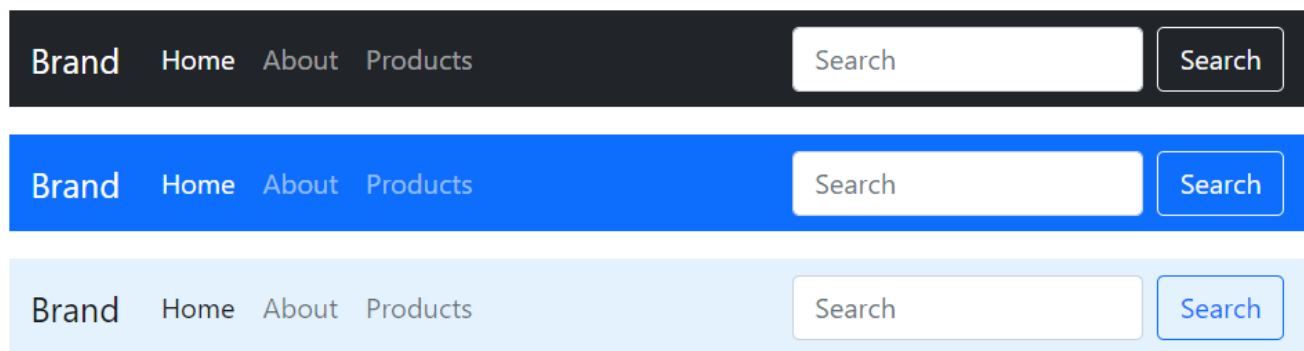
    <!-- Navbar content -->
</nav>

<nav class="navbar navbar-dark bg-primary">
    <!-- Navbar content -->
</nav>

<nav class="navbar navbar-light" style="background-color: #ddeeff;">
    <!-- Navbar content -->
</nav>

```

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



Bootstrap Fixed Navbars

Bootstrap also provides mechanism to create navbar that is fixed to the top, fixed to the bottom, or stickied to the top (i.e. scrolls with the page until it reaches the top, then stays there).

Navbar Fixed to the Top

Apply the position utility class `.fixed-top` to the `.navbar` element to fix the navbar at the top of the viewport, so that it won't scroll with the page. Here's an example:

```

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">
    <!-- Navbar content -->
</nav>

```

Navbar Fixed to the Bottom

Similarly, add the class `.fixed-bottom` to the `.navbar` element to fix the navbar at the bottom of the viewport. It also won't scroll with the page. Let's see how it works:

```
<nav class="navbar fixed-bottom navbar-expand-lg navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>
```

Navbar Sticked to the Top

You can also create sticky top navbar that scrolls with the page until it reaches the top, then stays there, by simply using the `.sticky-top` class on the `.navbar` element, like this:

```
<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>
```

Note: Remember to add `padding` (at least 70px) to the top or bottom of the `<body>` element to prevent the content to go underneath the navbar while implementing the *fixed top* or *fixed bottom* navbar. Also, be sure to add your custom `style sheet` after the Bootstrap's CSS file, otherwise style rules in your style sheet may be overridden by the Bootstrap's one.

Tip: Place `.navbar` content inside the `.container`, `.container-fluid` OR `.container{-sm|-md|-lg|-xl|-xxl}` for proper padding and alignment with the rest of the content.

Ex: sticky

```
<body>
  <div class="p-5 bg-warning">
    <h1>Demo Header</h1>
  </div>
  <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button
        type="button"
        class="navbar-toggler"
        data-bs-toggle="collapse"
        data-bs-target="#navbarCollapse"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <div class="navbar-nav">
```

```

    <a href="#" class="nav-item nav-link active">Home</a>
    <a href="#" class="nav-item nav-link">Profile</a>
    <a href="#" class="nav-item nav-link">Messages</a>
    <a href="#" class="nav-item nav-link disabled" tabindex="-1"
      >Reports</a>
  >
</div>
<div class="navbar-nav ms-auto">
  <a href="#" class="nav-item nav-link">Login</a>
</div>
</div>
</div>
</nav>

<!-- Demo Content -->
<div class="container mt-4">
  <div class="p-5 mb-4 bg-light rounded-3">
    <h1>Sticky Top Navbar</h1>
    <p class="lead">
      The static-top navbar scrolls with the page until it reaches the top,
      then stays there.
    </p>
  </div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem
    tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis
    vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
    scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis.
    In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu
    in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo,
    blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit
    suscipit, id interdum neque porttitor. Integer faucibus ligula. Quis quam
    ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit
    tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at
    mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis
    gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis
    justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget
    mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat.
    Maecenas mattis lectus enim, quis tincidunt dui molestie euismod.
    Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus. Tibulum
    consectetur scelerisque lacus, ac fermentum lorem convallis sed.
  </p>
  <p>
    Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis
    lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus
    scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id

```

mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero. Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.

</p>

<p>

Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id. Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.

</p>

<p>

Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit. Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et,

```

viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel.
Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus.
Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed
lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et
ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis
ultrices, magna quam rhoncus erat, in lacinia libero magna a ipsum.
</p>
<p>
Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus.
Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis
natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere
venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor
quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac
erat blandit vehicula ut eget urna. In hac habitasse platea dictumst.
Nullam ut iaculis nibh, eget eleifend elit.
</p>
</div>
</body>

```

////

Example: bottom navbar

```

<body>
  <div class="m-4">
    <nav class="navbar fixed-bottom navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a href="#" class="navbar-brand">Brand</a>
        <button
          type="button"
          class="navbar-toggler"
          data-bs-toggle="collapse"
          data-bs-target="#navbarCollapse"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <div class="navbar-nav">
            <a href="#" class="nav-item nav-link active">Home</a>
            <a href="#" class="nav-item nav-link">Profile</a>
            <a href="#" class="nav-item nav-link">Messages</a>
            <a href="#" class="nav-item nav-link disabled" tabindex="-1"
              >Reports</a>
          </div>
        </div>
      <div class="navbar-nav ms-auto">

```

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

```
<!-- Demo Content -->
```

```
<div class="container">
```

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

```
    <h1>Fixed Bottom Navbar</h1>
```

```
    <p class="lead">
```

```
      The fixed-bottom navbar bar is fixed at the bottom of the viewport and
      does not scroll with the rest of the page.
```

```
    </p>
```

```
  </div>
```

```
<p>
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem
  tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec
  turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
  scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis.
  In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in
  arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl
  leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam
  ut elit suscipit, id interdum neque porttitor. Integer faucibus
  ligula. Quis quam ut magna consequat faucibus. Pellentesque eget nisi a
  mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra
  sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida
  nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit,
  commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue
  dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis,
  bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui
  molestie euismod. Curabitur et diam tristique, accumsan nunc eu,
  hendrerit tellus. Tibulum consectetur scelerisque lacus, ac fermentum
  lorem convallis sed.
```

```
</p>
```

```
<p>
```

```
  Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a
  iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus.
  Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus
  metus id mi gravida, faucibus convallis neque pretium. Maecenas quis
  sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet
  tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi
  vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales
  libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum
  ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia
  Curae; Sed eu elementum nibh, quis varius libero. Nam eget purus nec est
```

consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.

</p>

<p>

Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id. Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.

</p>

<p>

Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit. Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et, viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel. Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus. Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis ultricies, magna quam rhoncus erat, in


```

    lacinia libero magna a ipsum.
  </p>
  <p>
    Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus.
    Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis
    natoque penatibus et magnis dis parturient montes, nascetur ridiculus
    mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere
    venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor
    quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio
    ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst.
    Nullam ut iaculis nibh, eget eleifend elit.
  </p>
</div>
</div>
</body>

```

/////

Fixed top navbar

```

<body>
  <div class="m-4">
    <nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a href="#" class="navbar-brand">Brand</a>
        <button
          type="button"
          class="navbar-toggler"
          data-bs-toggle="collapse"
          data-bs-target="#navbarCollapse"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <div class="navbar-nav">
            <a href="#" class="nav-item nav-link active">Home</a>
            <a href="#" class="nav-item nav-link">Profile</a>
            <a href="#" class="nav-item nav-link">Messages</a>
            <a href="#" class="nav-item nav-link disabled" tabindex="-1"
              >Reports</a>
          </div>
          <div class="navbar-nav ms-auto">
            <a href="#" class="nav-item nav-link">Login</a>
          </div>
        </div>
      </div>
    </nav>
  </div>
</body>

```

```
</nav>
```

```
<!-- Demo Content -->
```

```
<div class="container">
```

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

```
    <h1>Fixed Top Navbar</h1>
```

```
    <p class="lead">
```

```
      The fixed-top navbar bar is fixed at the top of the viewport and does  
      not scroll with the rest of the page.
```

```
    </p>
```

```
  </div>
```

```
<p>
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem  
  tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec  
  turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum  
  scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis.  
  In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in  
  arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl  
  leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam  
  ut elit suscipit, id interdum neque porttitor. Integer faucibus  
  ligula. Quis quam ut magna consequat faucibus. Pellentesque eget nisi a  
  mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra  
  sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida  
  nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit,  
  commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue  
  dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis,  
  bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui  
  molestie euismod. Curabitur et diam tristique, accumsan nunc eu,  
  hendrerit tellus. Tibulum consectetur scelerisque lacus, ac fermentum  
  lorem convallis sed.
```

```
</p>
```

```
<p>
```

```
  Orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a  
  iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus.  
  Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus  
  metus id mi gravida, faucibus convallis neque pretium. Maecenas quis  
  sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet  
  tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi  
  vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales  
  libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum  
  ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia  
  Curae; Sed eu elementum nibh, quis varius libero. Nam eget purus nec est  
  consectetur vehicula. Nullam ultrices nisl risus, in viverra libero  
  egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada.  
  Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus  
  hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare
```

odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.

</p>

<p>

Pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id. Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.

</p>

<p>

Enim arcu, interdum vel metus dignissim, venenatis feugiat purus. Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit. Convallis eget pretium eu, bibendum non leo. Proin suscipit purus adipiscing dolor gravida, in fermentum sapien blandit. Praesent pellentesque ligula dui, in gravida turpis vehicula ac. Pellentesque hendrerit nunc ut luctus hendrerit. Aliquam nec tincidunt urna. Ut interdum nec odio non interdum. Curabitur ligula justo, dapibus non ligula tristique, dapibus tristique nulla. Aliquam pulvinar dapibus eros, rutrum pretium urna iaculis ut. Nam est est, tempus id egestas et, viverra in dui. Aliquam gravida orci tortor, sed congue justo ornare vel. Cras in quam consectetur eros varius scelerisque. Ut vel fermentum purus. Nullam interdum blandit turpis, id pellentesque massa feugiat at. Ut sed lectus lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla rutrum, ante quis convallis ultricies, magna quam rhoncus erat, in lacinia libero magna a ipsum.

</p>

<p>

Fusce enim arcu, interdum vel metus dignissim, venenatis feugiat purus.

Nulla posuere orci ut leo sodales, sed cursus dolor ornare. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet quam orci. Nulla sollicitudin lectus eget posuere venenatis. Sed vestibulum elementum sagittis. Quisque tristique tortor quis feugiat sollicitudin. Ut pellentesque luctus vulputate. Ut at odio ac erat blandit vehicula ut eget urna. In hac habitasse platea dictumst. Nullam ut iaculis nibh, eget eleifend elit.

```
</p>  
</div>  
</div>  
</body>
```