**BOOTSTRAP**

What is Twitter Bootstrap?

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS, and Javascript.

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

### Why Use Bootstrap?

* · Mobile first approach: Bootstrap 3 framework consists of Mobile first styles throughout the entire library instead of them in separate files.
* Browser Support: It is supported by all popular browsers.
* Easy to get started: With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
* Responsive design: Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter Bootstrap Responsive Design.
* Provides a clean and uniform solution for building an interface for developers.
* It contains beautiful and functional built-in components which are easy to customize.
* It also provides web-based customization.
* And best of all it is an open source.

**BOOTSTRAP -GRID SYSTEM**

What is a Bootstrap grid system ?

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes predefined classes for easy layout options, as well as powerful mixins for generating more semantic layouts.

Let us understand the above statement. Bootstrap 3 is mobile first in the sense that the code for Bootstrap now starts by targeting smaller screens like mobile devices, tablets, and then “expands” components and grids for larger screens such as laptops, desktops.

## **Working of Bootstrap Grid System**

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works:

· Rows must be placed within a **.container** class for proper alignment and padding.

· Use rows to create horizontal groups of columns.

· Content should be placed within the columns, and only columns may be the immediate children of rows.

· Predefined grid classes like **.row and .col-xs-4** are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.

· Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on **.rows**.

· Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three **.col-xs-4**.

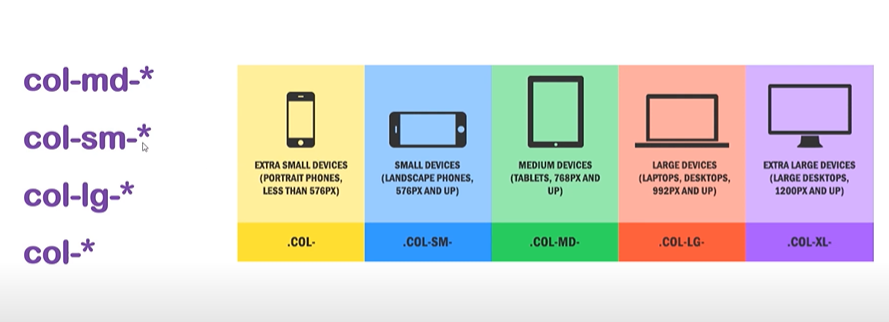
**GRID OPTIONS**

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

The Bootstrap grid system has four classes:

* xs (for phones - screens less than 768px wide)
* sm (for tablets - screens equal to or greater than 768px wide)
* md (for small laptops - screens equal to or greater than 992px wide)
* lg (for laptops and desktops - screens equal to or greater than 1200px wide)

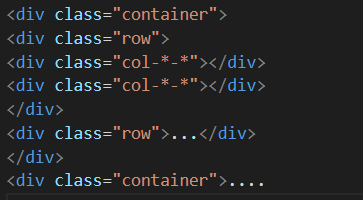
The classes above can be combined to create more dynamic and flexible layouts.



### 

### Basic Grid Structure

## Following is basic structure of Bootstrap grid:



### Details

## · <div class="container">...</div> element is added to ensure proper centering and maximum width for layout.

## · Once a container is added, next you need to think in terms of rows. Add <div class="row">...</div> and columns <div class="col-md-6"></div> inside the rows.

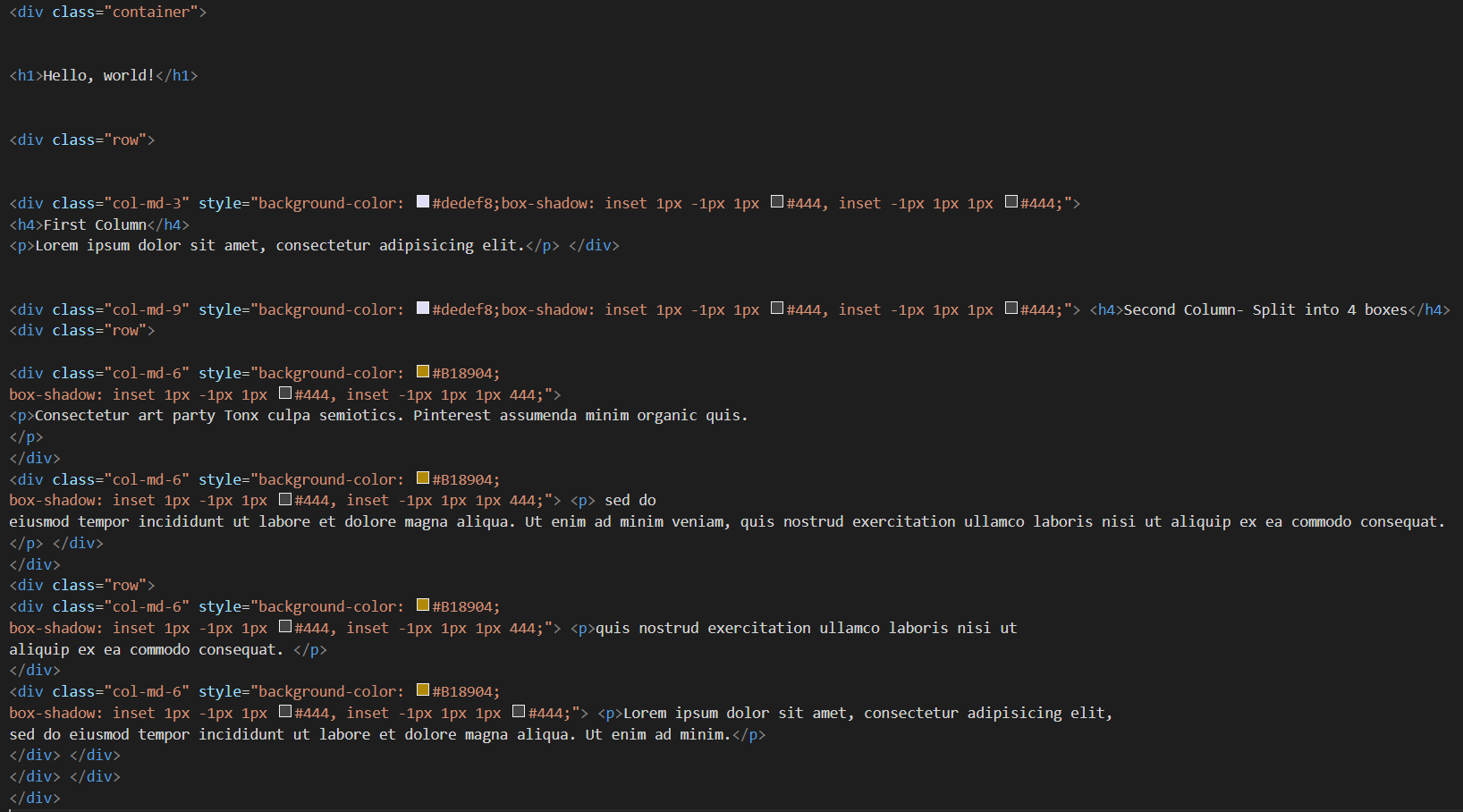
## · Every row in the grid is made up of 12 units and you can define the desired size of your columns using those units. In our example we have two columns each made of 6 units wide i.e 6+6=12.

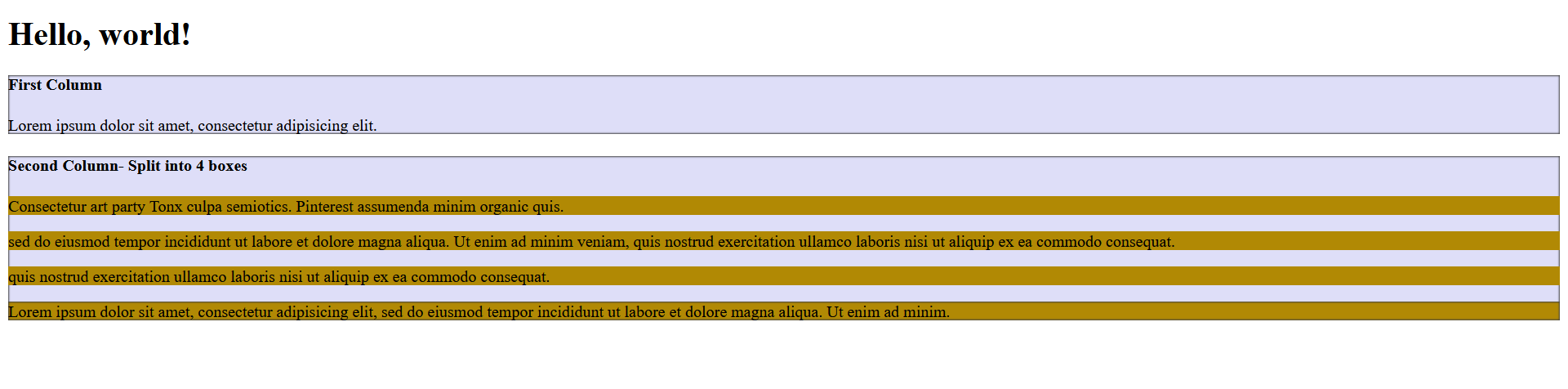
## **Nesting Columns**

To nest your content with the default grid, add a new **.row** and set of **.col-md-**

**\*** columns within an existing **.col-md-\*** column. Nested rows should include a set of columns that add up to 12.

In the following example, the layout has two columns, with the second one being split into four boxes over two rows.

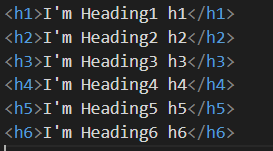




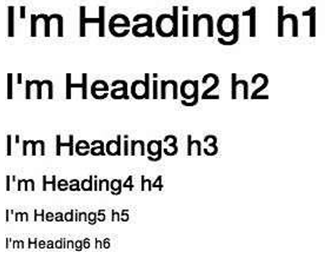
**BOOTSTRAP - TYPOGRAPHY**

## **Headings**

All HTML headings (h1 to h6) are styled in Bootstrap. An example is shown below:

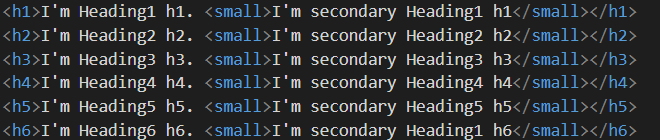


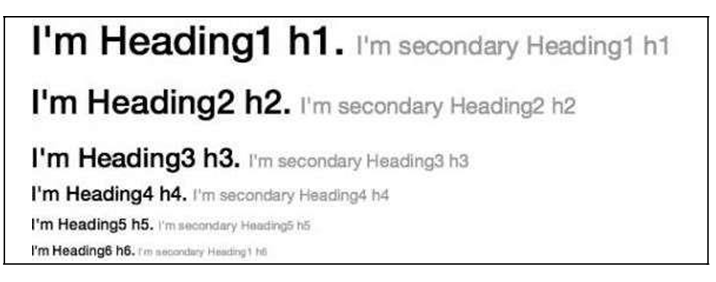
OUTPUT :-



### **Inline Subheadings**

To add an inline subheading to any of the headings, simply add <small> around any of the elements or add **.small** class and you will get smaller text in a lighter color as shown in the example below:



The above code segment with Bootstrap will produce the following result:

## 

## 

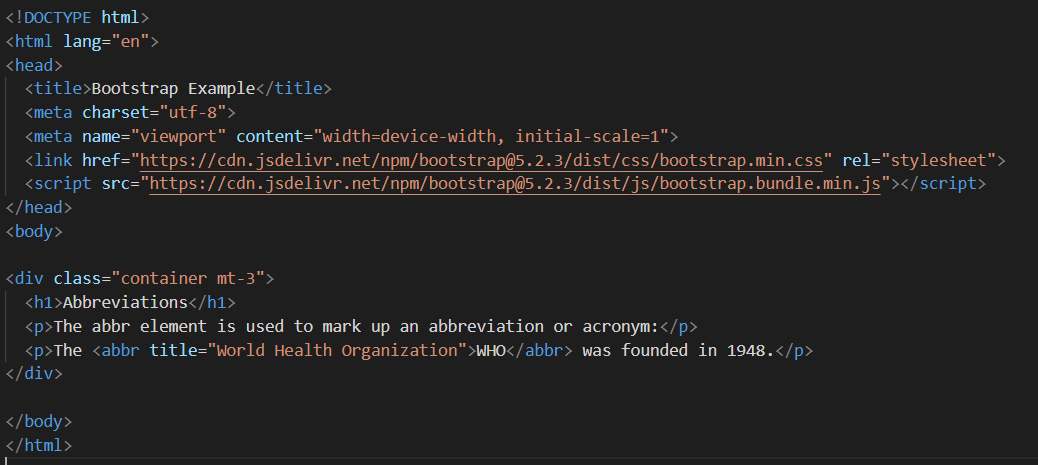
## **<mark>**

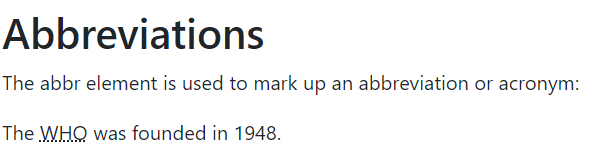
Bootstrap 5 will style <mark> and .mark with a yellow background color and some padding:



**<abbr>**

Bootstrap 5 will style the HTML <abbr> element with a dotted border bottom and a cursor with question mark on hover:



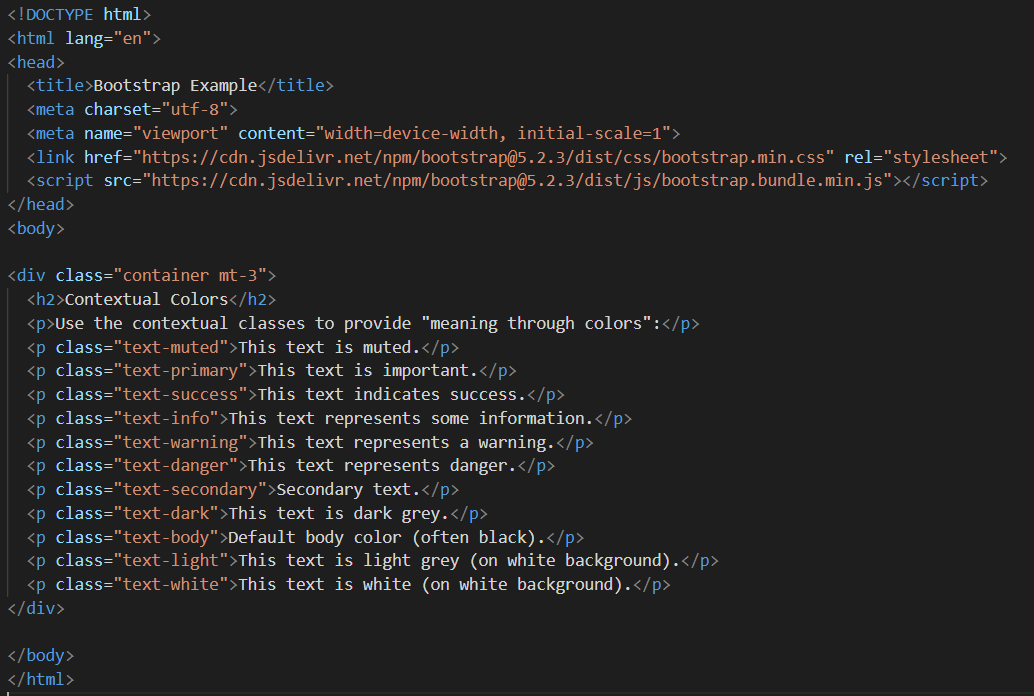


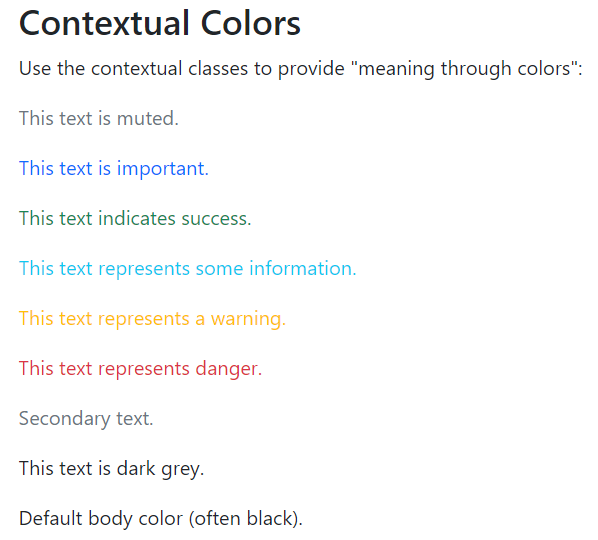
**BOOTSTRAP COLORS**

**Text Colors**

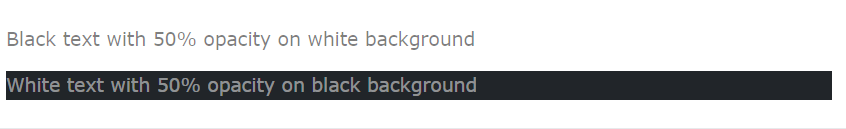
Bootstrap 5 has some contextual classes that can be used to provide "meaning through colors".

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



****

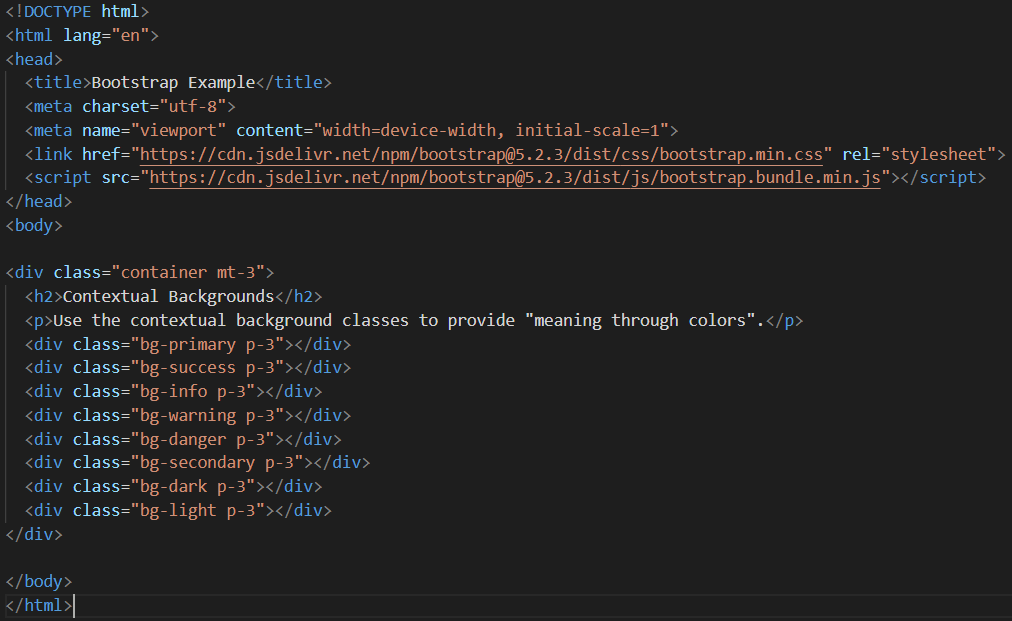
You can also add 50% opacity for black or white text with the .text-black-50 or .text-white-50 classes:

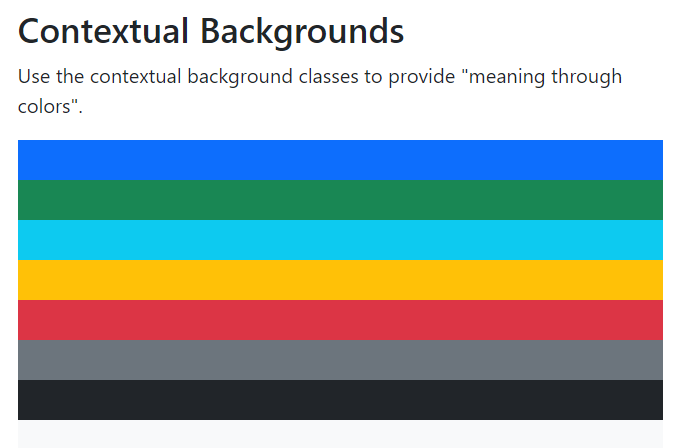


## 

## **Background Colors**

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



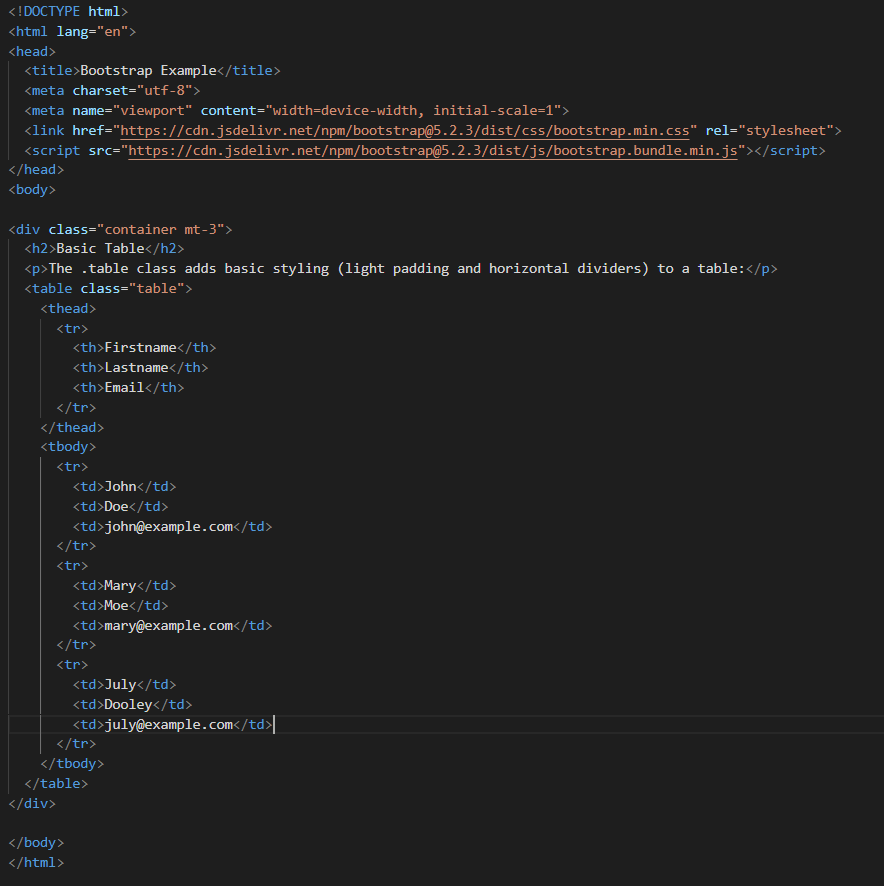


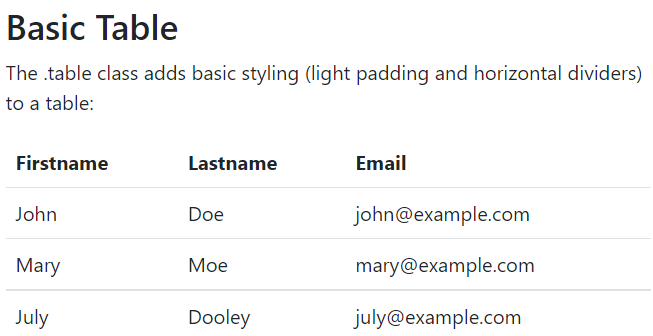
However, you can use the .text-bg-*color* classes and Bootstrap will automatically handle the appropriate text color for each background color.

**BOOTSTRAP TABLES**

## **Basic Table**

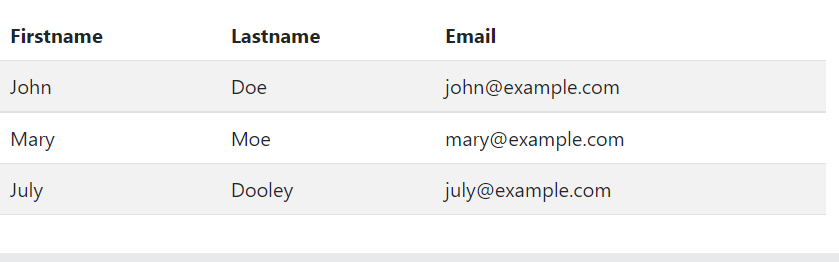
A basic Bootstrap 5 table has a light padding and horizontal dividers.

The .table class adds basic styling to a table:

****

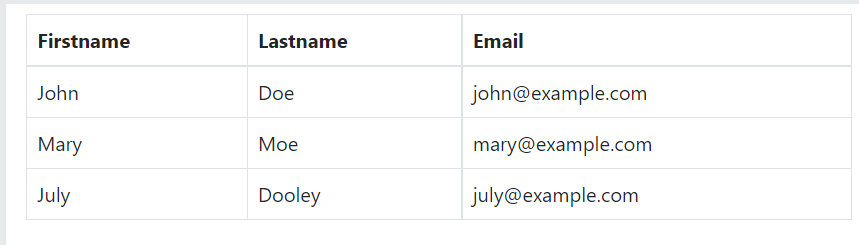
## Striped Rows

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

****

## Bordered Table

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



## 

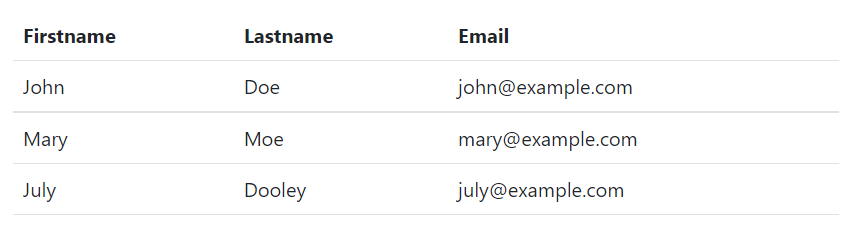
## 

## 

## 

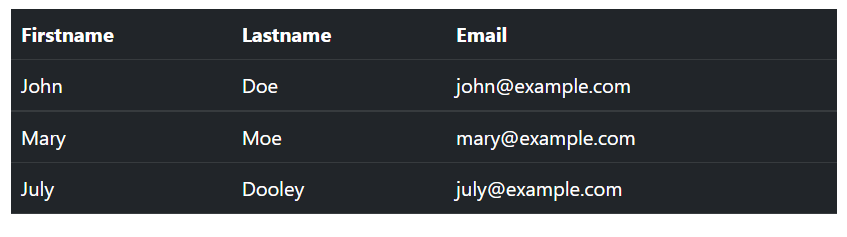
## Hover Rows

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



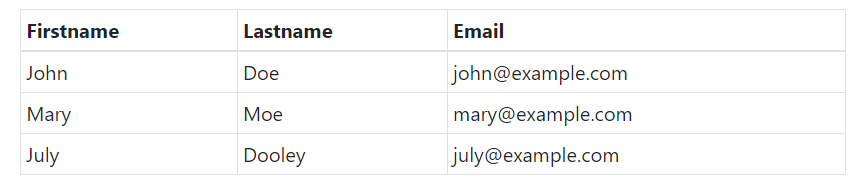
## Black/Dark Table

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



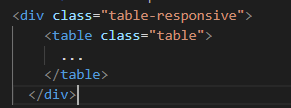
## Small table

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

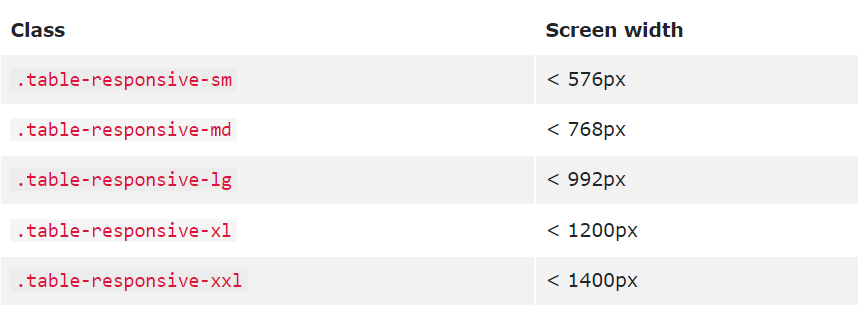


## Responsive Tables

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



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



# **Bootstrap Images**

## 

## **Rounded Corners**

The .rounded class adds rounded corners to an image:



## **Circle**

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



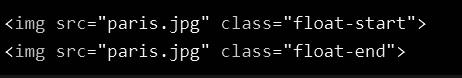
## **Thumbnail**

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



## **Aligning Images**

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



## **Centered Image**

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



## **Responsive Images**

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

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

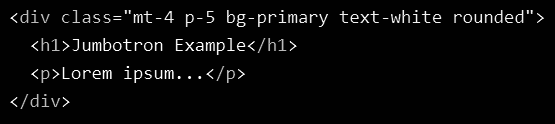
The .img-fluid class applies max-width: 100%; and height: auto; to the image:



## **Bootstrap Jumbotron**

A jumbotron was introduced in Bootstrap 3 as a big padded box for calling extra attention to some special content or information.

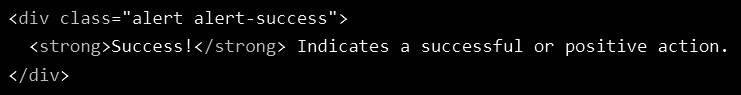
Jumbotrons are no longer supported in Bootstrap 5. However, you can use a <div> element and add special helper classes together with a color class to achieve the same effect:



# **Bootstrap Alerts**

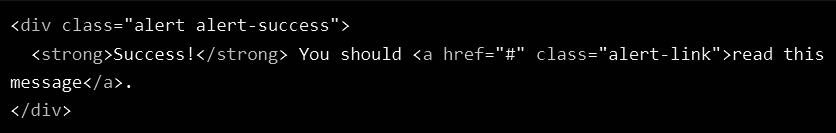
Bootstrap 5 provides an easy way to create predefined alert messages:

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



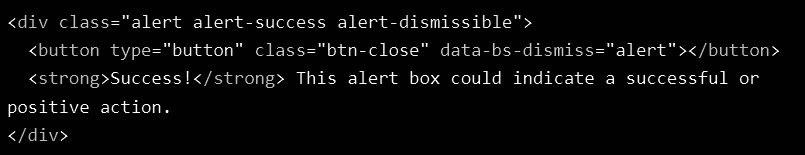
## **Alert Links**

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



## **Closing Alerts**

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



## **Animated Alerts**

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



# 

# 

# 

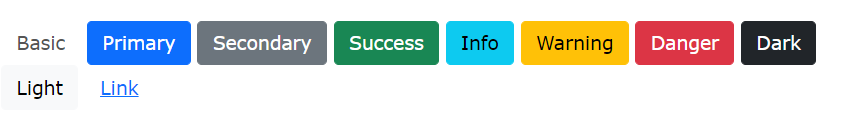
# 

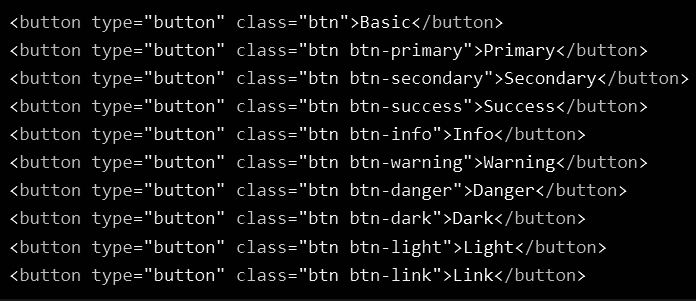
# 

# **Bootstrap Buttons**

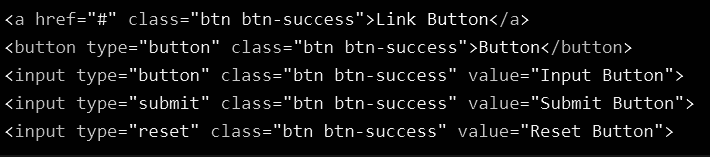
## **Button Styles**

Bootstrap 5 provides different styles of buttons:





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

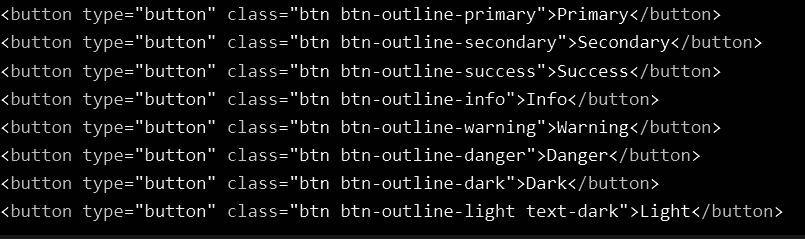


## **Button Outline**

Bootstrap 5 also provides eight outline/bordered buttons.

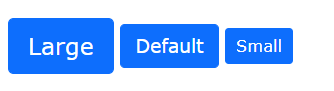
Move the mouse over them to see an additional "hover" effect:

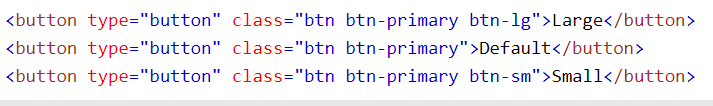




## **Button Sizes**

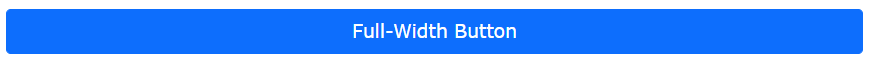
Use the .btn-lg class for large buttons or .btn-sm class for small buttons:

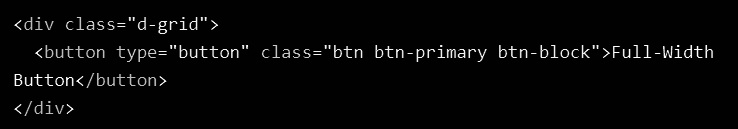




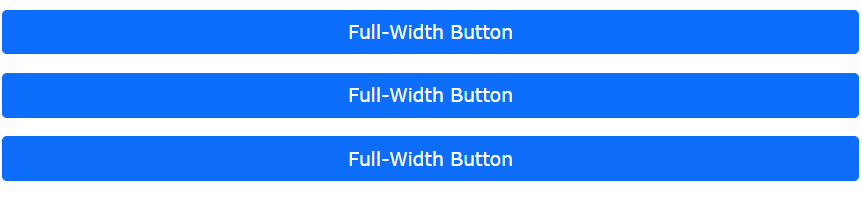
## **Block Level Buttons**

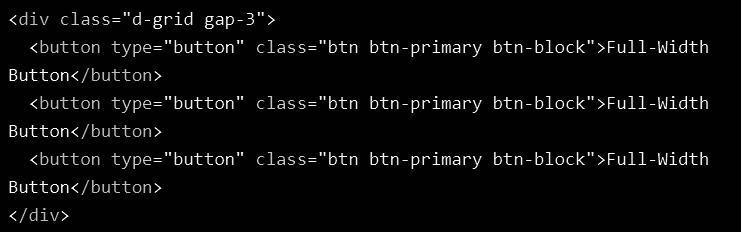
To create a block level button that spans the entire width of the parent element, use the .d-grid "helper" class on the parent element:





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





## 

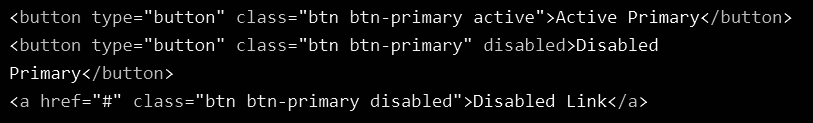
## 

## **Active/Disabled Buttons**

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



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



## 

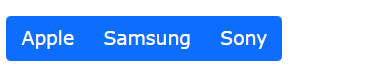
## 

## 

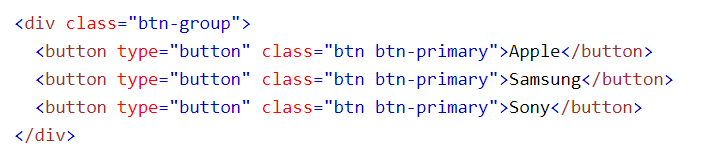
## 

## **Button Groups**

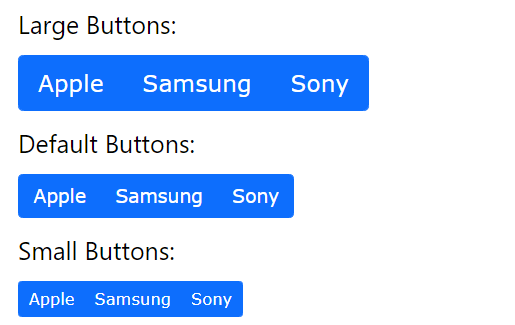
Bootstrap 5 allows you to group a series of buttons together (on a single line) in a button group:

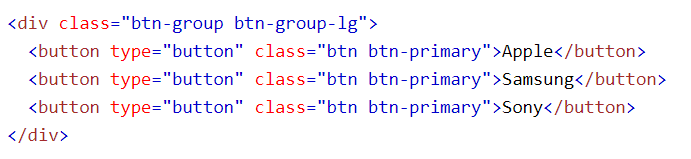


Use a <div> element with class .btn-group to create a button group:



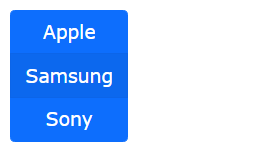
Tip: Instead of applying button sizes to every button in a group, use class .btn-group-lg for a large button group or the .btn-group-sm for a small button group:



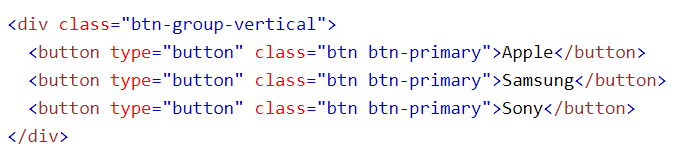


## **Vertical Button Groups**

Bootstrap 5 also supports vertical button groups:

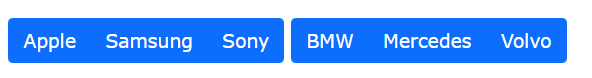


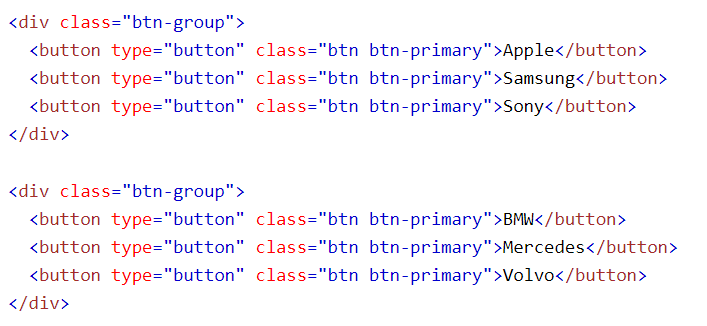
Use the class .btn-group-vertical to create a vertical button group:



## **Button Groups Side by Side**

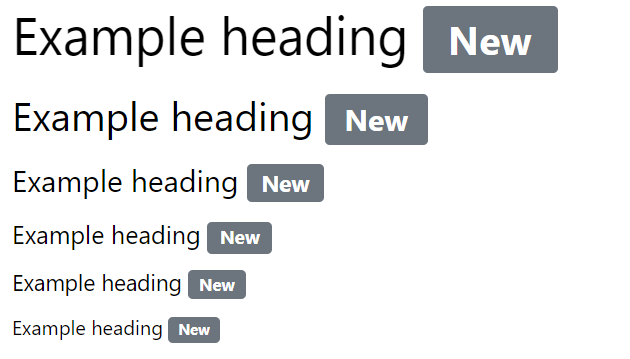
Button groups are "inline" by default, which makes them appear side by side when you have multiple groups:



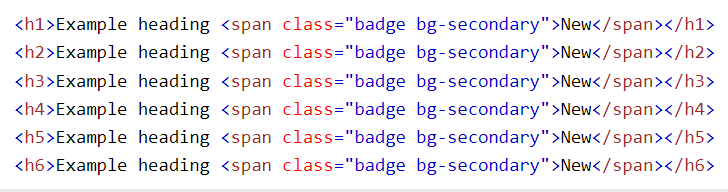


# **Bootstrap Badges**

Badges are used to add additional information to any content:



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



## Contextual Badges



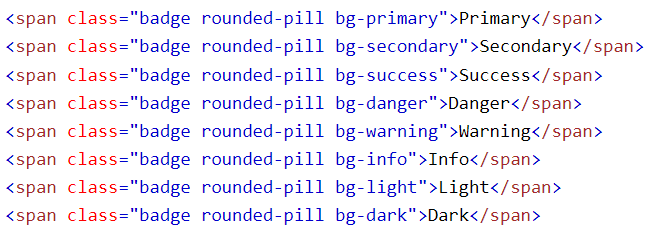
Use any of the contextual classes (.bg-\*) to change the color of a badge:



## Pill Badges

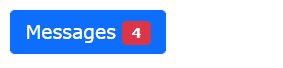


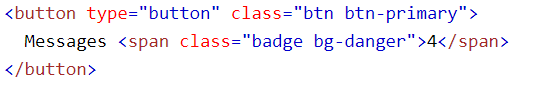
Use the .rounded-pill class to make the badges more round:



## Badge inside an Element

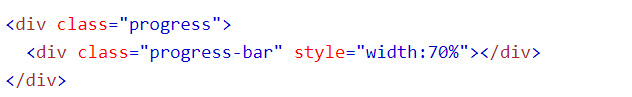
An example of using a badge inside a button:





# **Bootstrap Progress Bars**

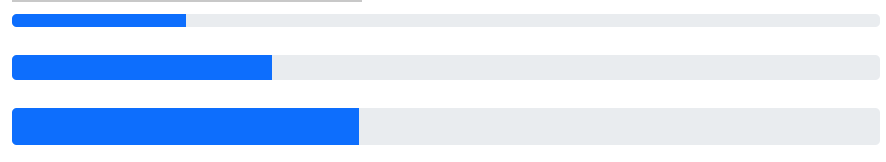
A progress bar can be used to show how far a user is in a process.



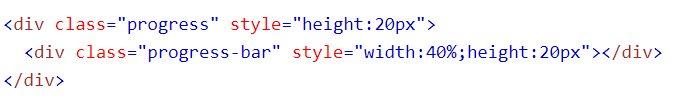
To create a default progress bar, add a .progress class to a container element and add the .progress-bar class to its child element. Use the CSS width property to set the width of the progress bar:



## Progress Bar Height



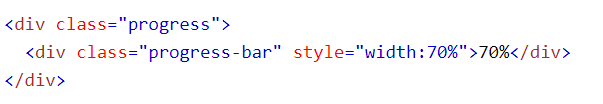
The height of the progress bar is 1rem (usually 16px) by default. Use the CSS height property to change it. Note that you must set the same height for the progress container and the progress bar:



## Progress Bar Labels

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





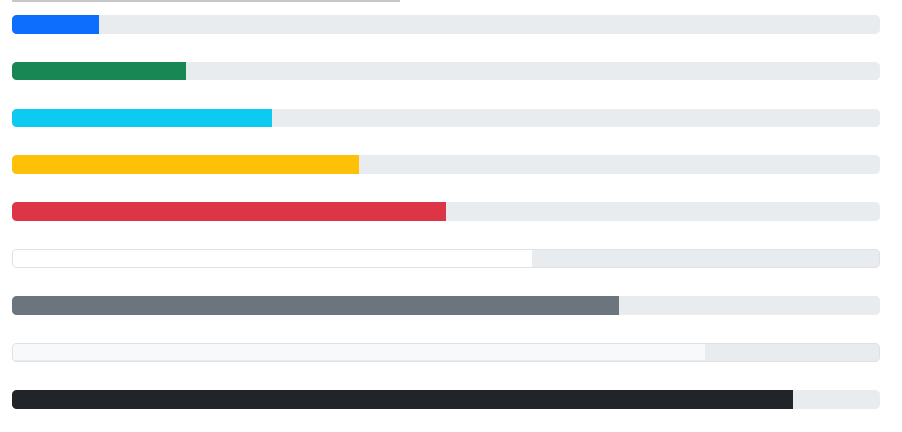
## 

## 

## 

## 

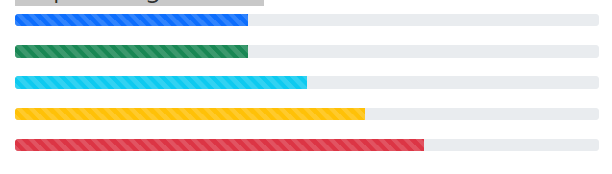
## Colored Progress Bars



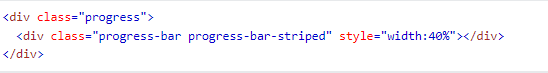
By default, the progress bar is blue (primary). Use any of the contextual background classes to change its color:



## Striped Progress Bars



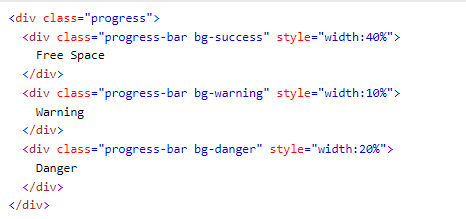
Use the .progress-bar-striped class to add stripes to the progress bars:



## Multiple Progress Bars

Progress bars can also be stacked:





# **Bootstrap Spinners**

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

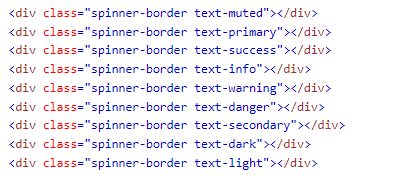




## Colored Spinners

Use any text color utilites to add a color to the spinner:

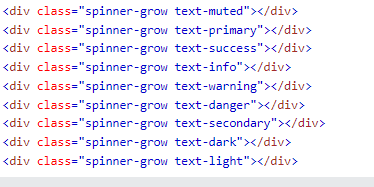




## Growing Spinners

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





## 

## Spinner Size

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





## 

## Spinner Buttons

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



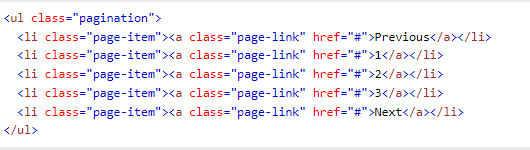


# Bootstrap Pagination

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



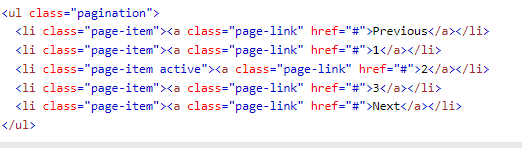
To create a basic pagination, add the .pagination class to an <ul> element. Then add the .page-item to each <li> element and a .page-link class to each link inside <li>:



## Active State

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





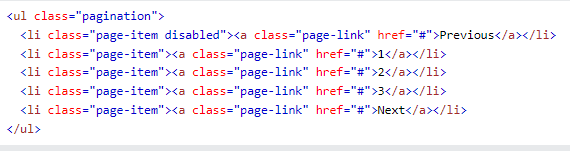
## 

## 

## Disabled State

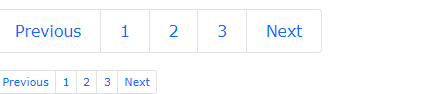
The .disabled class is used for un-clickable links:



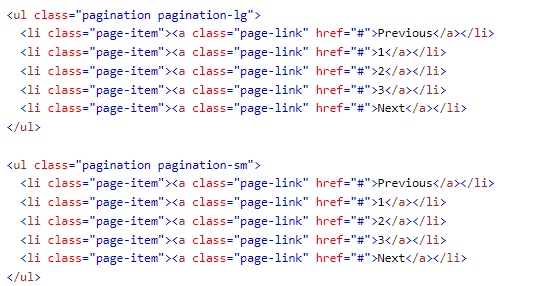


## Pagination Sizing

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



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



## Pagination Alignment

Use utility classes to change the alignment of the pagination:





## 

## Breadcrumbs

Another form for pagination, is breadcrumbs:



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

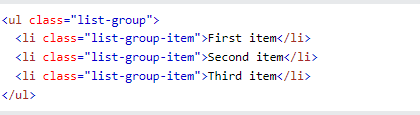


# Bootstrap List Groups

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



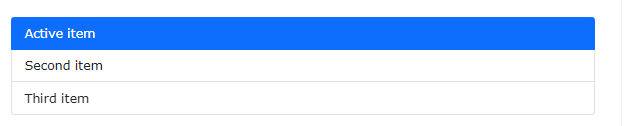
To create a basic list group, use an <ul> element with class .list-group, and <li> elements with class .list-group-item:



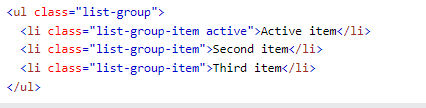
## 

## 

## Active State

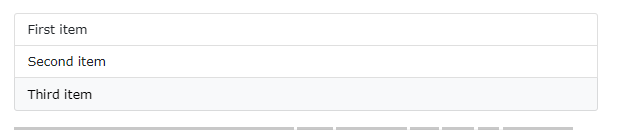


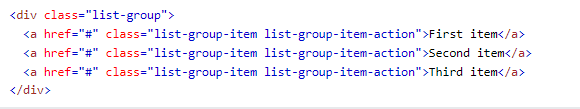
Use the .active class to highlight the current item:



## List Group With Linked Items

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





## 

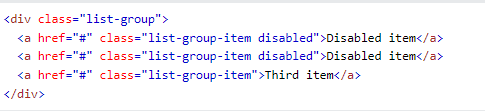
## 

## 

## Disabled Item

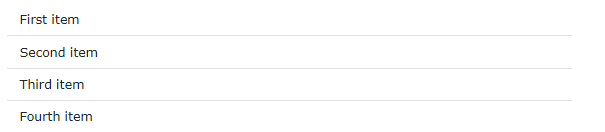
The .disabled class adds a lighter text color to the disabled item. And when used on links, it will remove the hover effect:

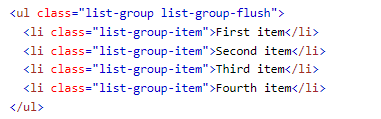




## Flush / Remove Borders

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





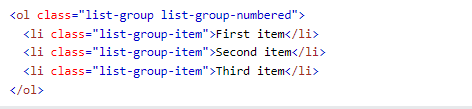
## 

## 

## Numbered List Groups

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

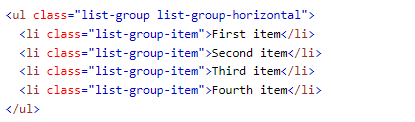




## Horizontal List Groups

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

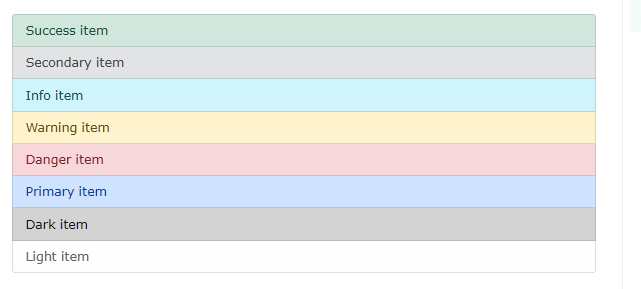




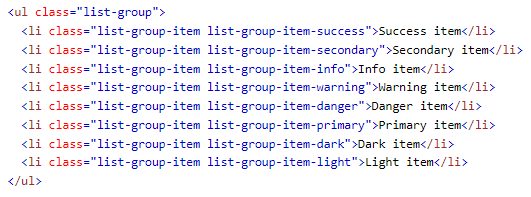
## 

## Contextual Classes

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



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



# 

# 

# Bootstrap Cards

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

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

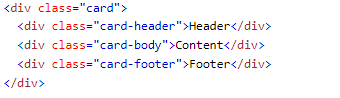




## Header and Footer



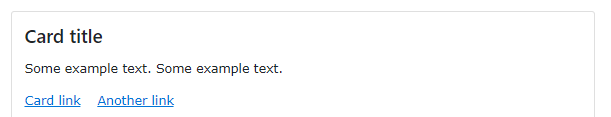
The .card-header class adds a heading to the card and the .card-footer class adds a footer to the card:



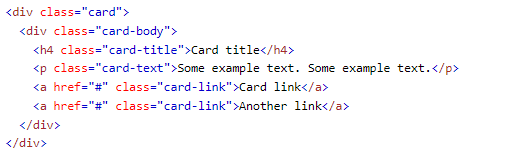
## 

## 

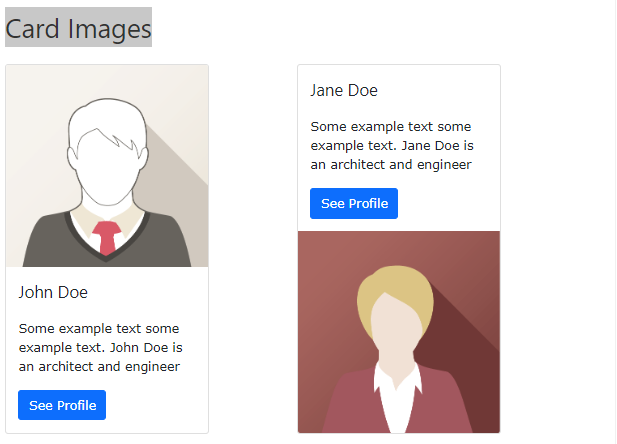
## Titles, text, and links



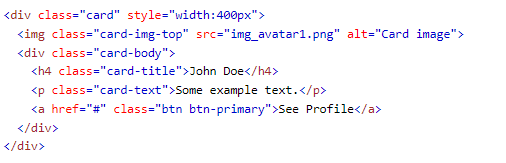
Use .card-title to add card titles to any heading element. The .card-text class is used to remove bottom margins for a <p> element if it is the last child (or the only one) inside .card-body. The .card-link class adds a blue color to any link, and a hover effect.



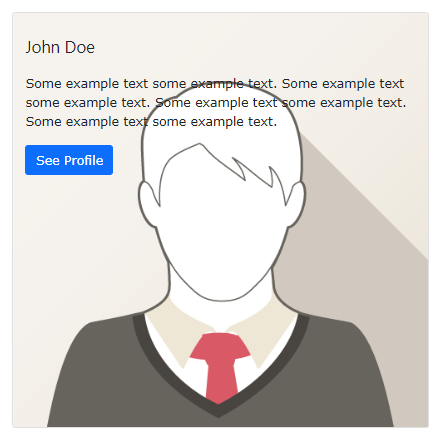
## Card Images



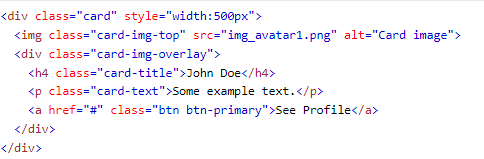
Add .card-img-top or .card-img-bottom to an <img> to place the image at the top or at the bottom inside the card. Note that we have added the image outside of the .card-body to span the entire width:



## Card Image Overlays



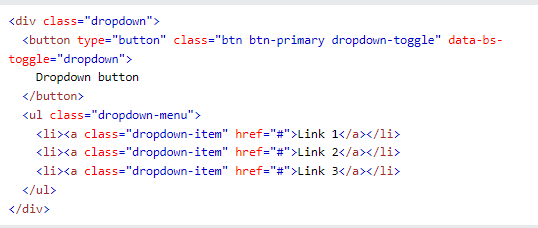
Turn an image into a card background and use .card-img-overlay to add text on top of the image:



# **Bootstrap Dropdowns**

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





The .dropdown class indicates a dropdown menu.

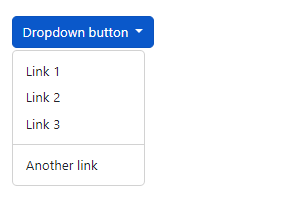
To open the dropdown menu, use a button or a link with a class of .dropdown-toggle and the data-bs-toggle="dropdown" attribute.

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

## 

## Dropdown Divider

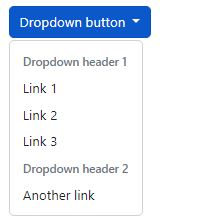
The .dropdown-divider class is used to separate links inside the dropdown menu with a thin horizontal border:





## Dropdown Header

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

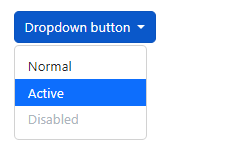


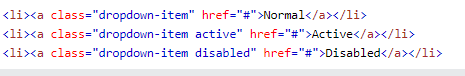


## Disable and Active items

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

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



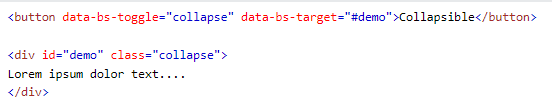


# 

# **Bootstrap Collapse**

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



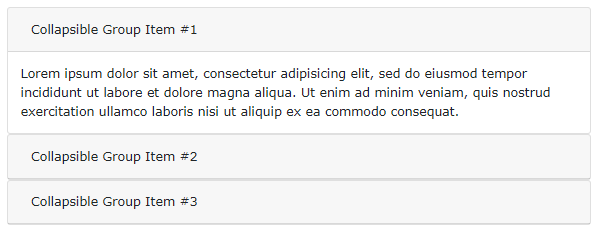


The .collapse class indicates a collapsible element (a <div> in our example); this is the content that will be shown or hidden with a click of a button.

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

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

## Accordion



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

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



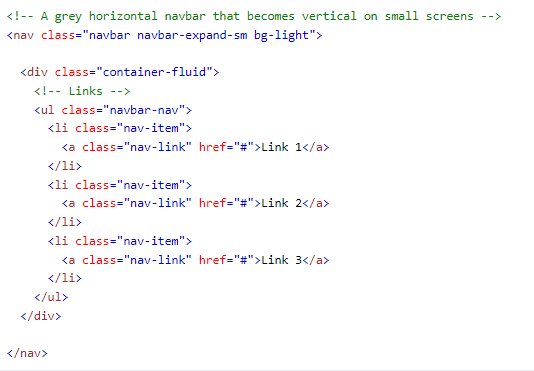
# **Bootstrap Navbars**

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

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

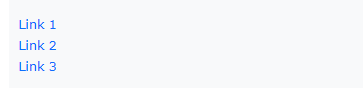
To add links inside the navbar, use either an <ul> element (or a <div>) with class="navbar-nav". Then add <li> elements with a .nav-item class followed by an <a> element with a .nav-link class:

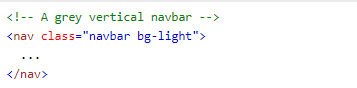




## Vertical Navbar

Remove the .navbar-expand-\* class to create a navigation bar that will always be vertical:

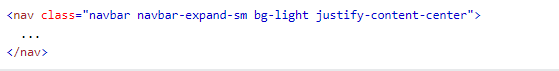




## Centered Navbar

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





## 

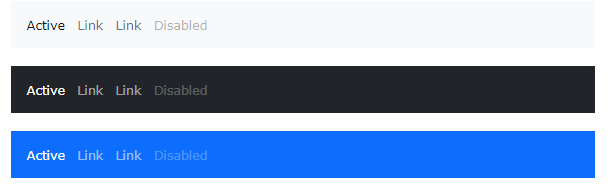
## 

## 

## 

## 

## Colored Navbar



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

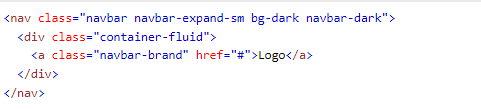
Tip: Add a white text color to all links in the navbar with the .navbar-dark class, or use the .navbar-light class to add a black text color.



## Brand / Logo

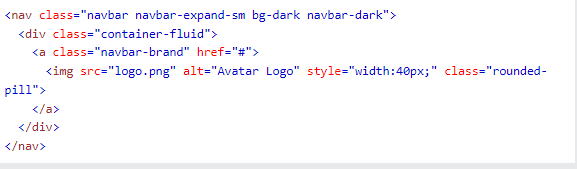
The .navbar-brand class is used to highlight the brand/logo/project name of your page:





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





## 

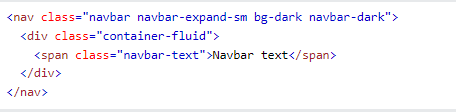
## 

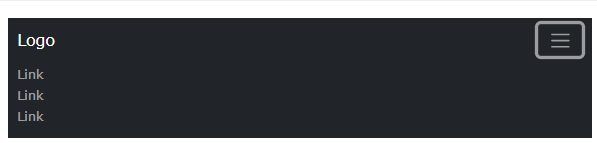
## 

## Navbar Text



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





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

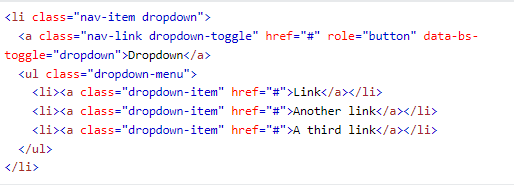
To create a collapsible navigation bar, use a button with class="navbar-toggler", data-bs-toggle="collapse" and data-bs-target="#*thetarget*". Then wrap the navbar content (links, etc) inside a <div> element with class="collapse navbar-collapse", followed by an id that matches the data-bs-target of the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



## Navbar With Dropdown

Navbars can also hold dropdown menus:





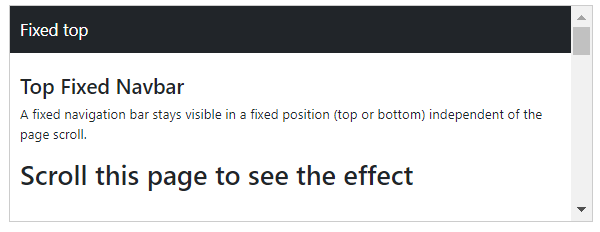
## Navbar Forms and Buttons



You can also include forms inside the navigation bar:



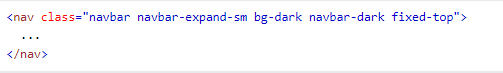
## Fixed Navigation Bar



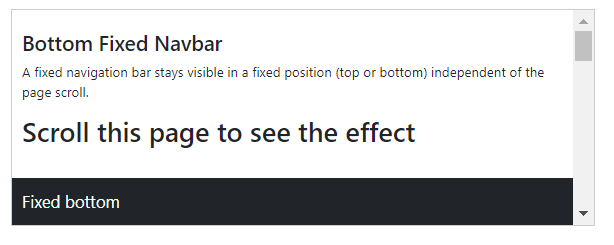
The navigation bar can also be fixed at the top or at the bottom of the page.

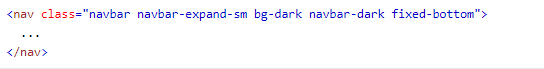
A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

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

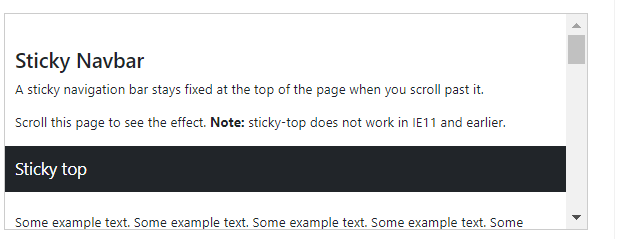


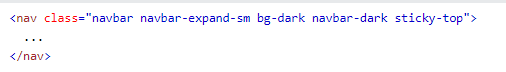
Use the .fixed-bottom class to make the navbar stay at the bottom of the page:





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





# Bootstrap Carousel

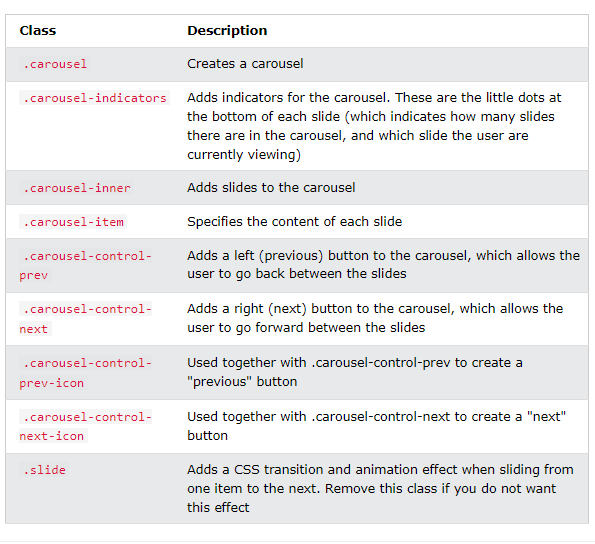
## How To Create a Carousel

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

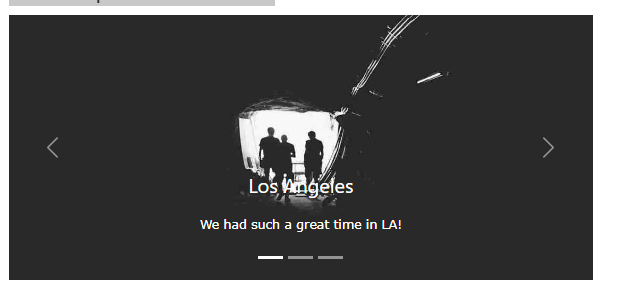




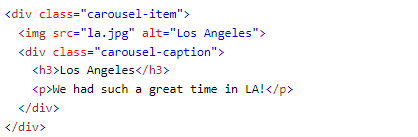
A description of what each class from the example above do:



## Add Captions to Slides

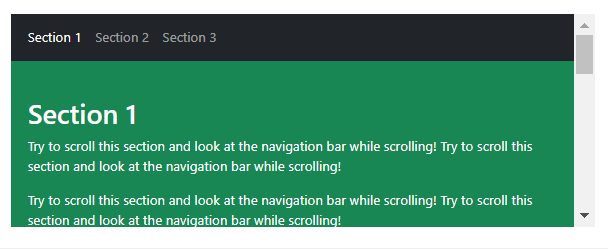


Add elements inside <div class="carousel-caption"> within each <div class="carousel-item"> to create a caption for each slide:



# 

# **Bootstrap Scrollspy**

****

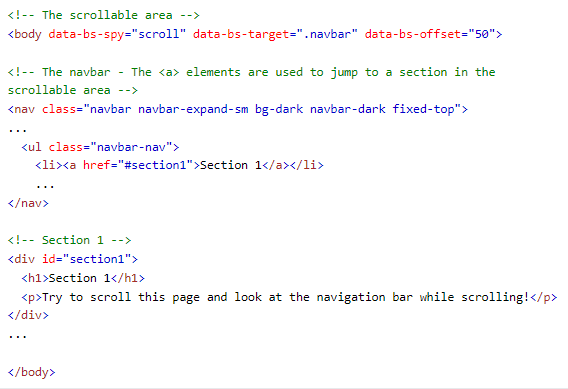
Scrollspy is used to automatically update links in a navigation list based on scroll position.

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

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

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

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



# 

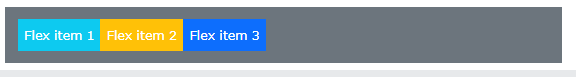
# 

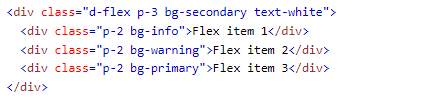
# **Bootstrap Flex**

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

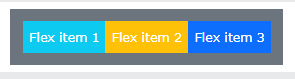
The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

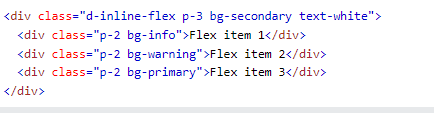
To create a flexbox container and to transform direct children into flex items, use the d-flex class:





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



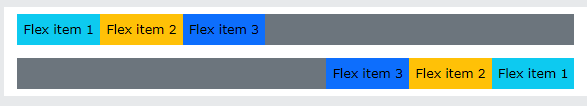


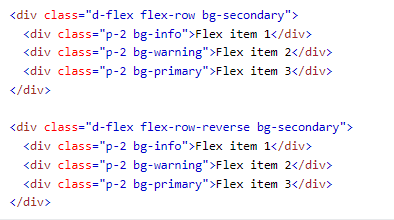
## 

## Horizontal Direction

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

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





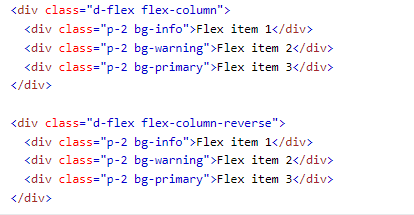
## 

## 

## Vertical Direction

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



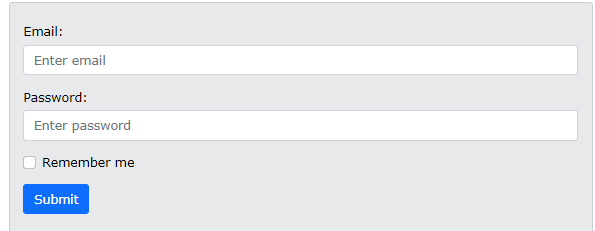


# 

# Bootstrap Forms

## Stacked Form

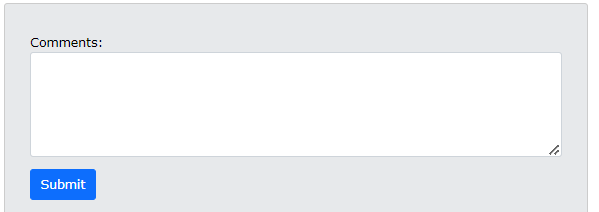
All textual <input> and <textarea> elements with class .form-control get proper form styling:





## 

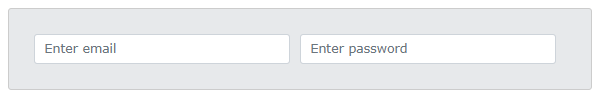
## Textarea



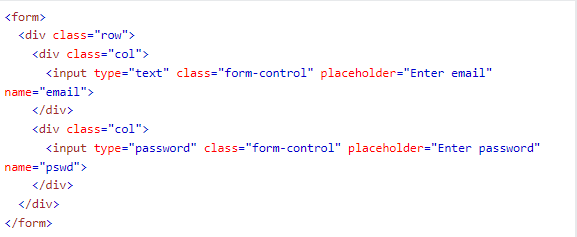


## 

## Form Row/Grid (Inline Forms)

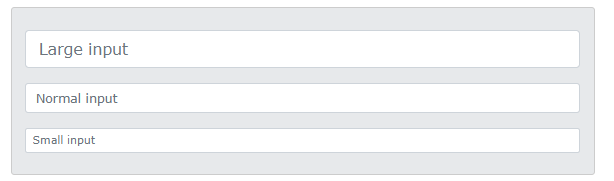


If you want your form elements to appear side by side, use .row and .col:

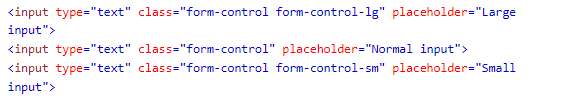


## 

## Form Control Size



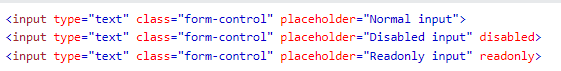
You can change the size of .form-control inputs with .form-control-lg or .form-control-sm:



## Disabled and Readonly



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



## Plain text Inputs



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



## Color Picker





# **Bootstrap Select**

## Select Menu

Select menu (select one):

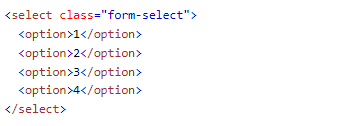


Multiple select menu (hold ctrl or shift (or drag with the mouse) to select more than one):



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

To style a select menu in Bootstrap 5, add the .form-select class to the <select> element:



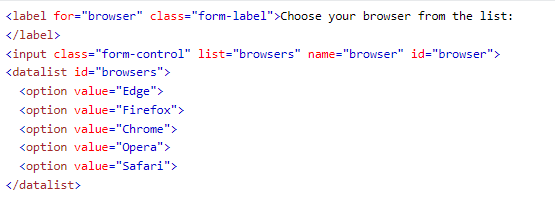
Use the .form-select-lg or .form-select-sm class to change the size of the select menu.

## Data Lists

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

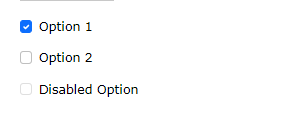
Choose your browser from the list:

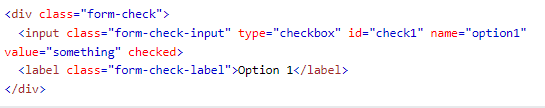




## Checkboxes

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





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

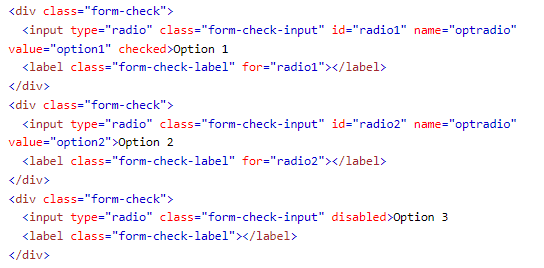
Then, add the .form-check-label class to label elements, and .form-check-input to style checkboxes properly inside the .form-check container.

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

## Radio buttons

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

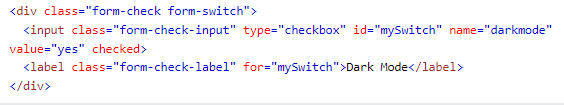




## Toggle Switches

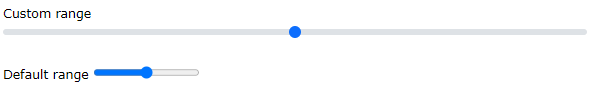
If you want your checkbox to be styled as a toggle switch, use the .form-switch class together with the .form-check container:





## Custom Range

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





## Steps

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





## Min and Max

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

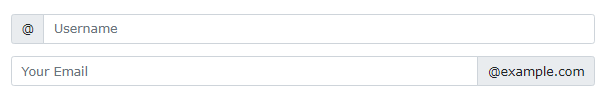


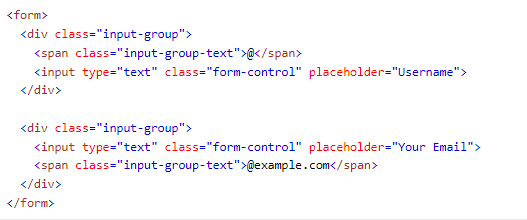


## Input Groups

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

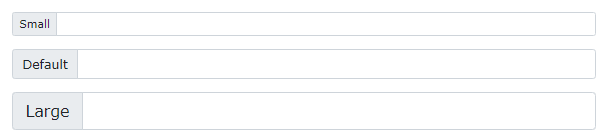
To style the specified help text, use the .input-group-text class:

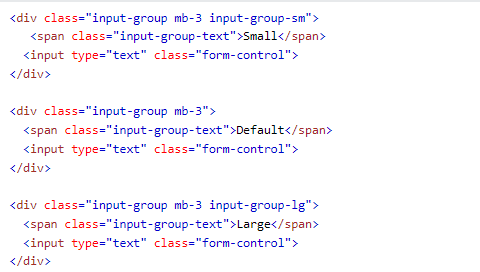




## Input Group Size

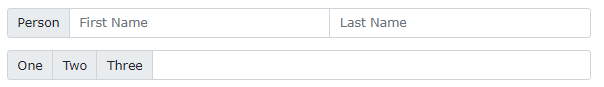
Use the .input-group-sm class for small input groups and .input-group-lg for large inputs groups:

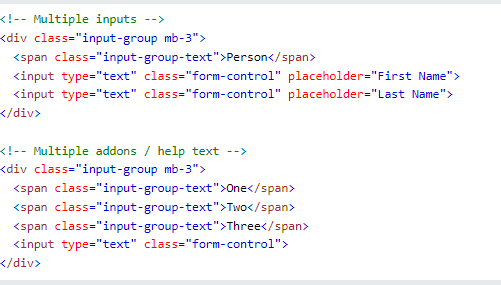




## Multiple Inputs and Helpers

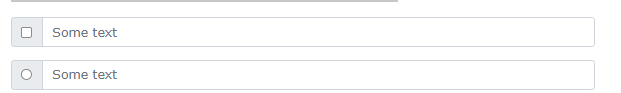
Add multiple inputs or addons:

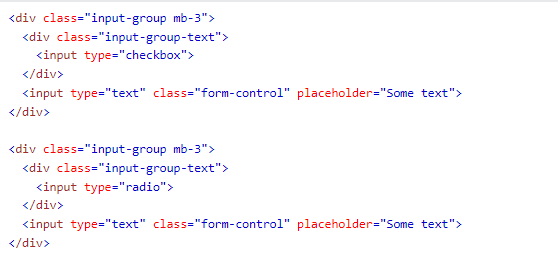




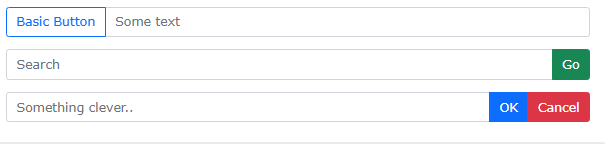
## Input Group with Checkboxes and Radios

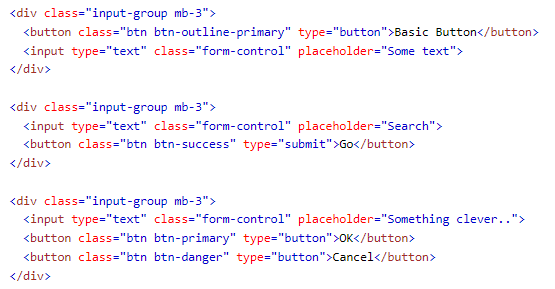
You can also use checkboxes or radio buttons instead of text:





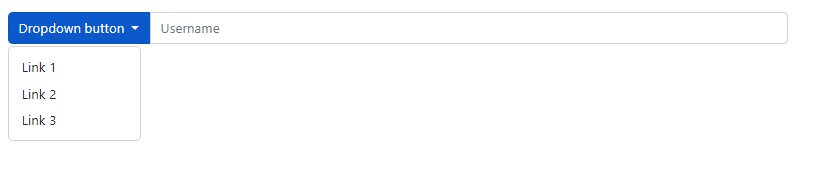
## Input Group Buttons

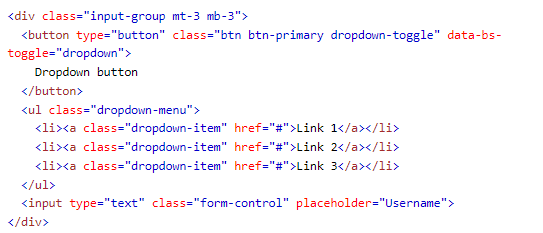




## Input Group with Dropdown Button

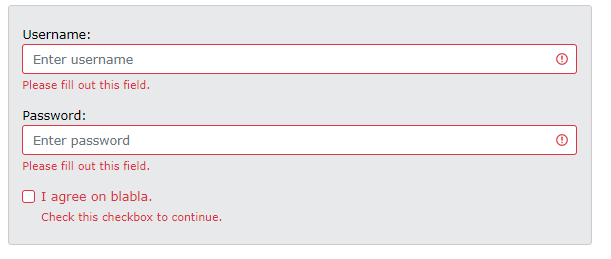
Add a dropdown button in the input group. Note that you don't need the .dropdown wrapper, as you normally would.





## 

## Form Validation



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

