**Environment Setup**

There are 2 ways use Bootstrap, one is downloading files directly from getbootstrap.com and other is by including it from CDN (Content Delivery Network).

A **content delivery network** (**CDN**) refers to a geographically distributed group of servers which **work** together to provide fast delivery of Internet content. A **CDN** allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos.

Bootstrap provides the CDN urls at their website, which you need to use above the </head> section of your HTML document.

<!-- Latest compiled and minified CSS --> <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

<!-- Latest compiled JavaScript --> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

**Create First Web Page With Bootstrap**

**1. Add the HTML5 doctype**

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype. Hence you are required to include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

</head>

</html>

**2. Bootstrap is mobile-first**

Bootstrap is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework. To ensure proper rendering and touch zooming, add the followingtag inside theelement:

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

Note:

* The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
* The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

**3. Bootstrap requires a containing element to wrap site contents**

We will further discuss the containing element later in the guide, for now just think there is a class called as “container” or “container-fluid” whose function is to wrap the HTML content inside the.

–

Hence your document with bootstrap will look something like:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

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

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>My First Bootstrap Page</h1>

<p>This is some text.</p>

</div>

</body>

</html>

**Bootstrap Container**

You learned from the previous section that Bootstrap requires a containing element to wrap site contents. The current Bootstrap version uses two kind of container classes to wrap the page’s contents. It contains:

* **.container** − It represents a fixed width container.
* **.container-fluid** − It represents a full width container.

**Container**

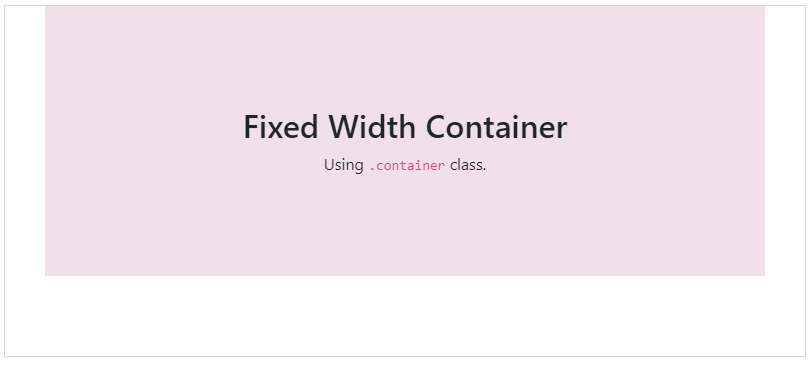
The .container class is used to wrap the page content with fixed width and content can be placed in the center easily by using the .container class as shown below.

<div class = "container">

...

</div>

The output will look something like:



**Fluid Container**

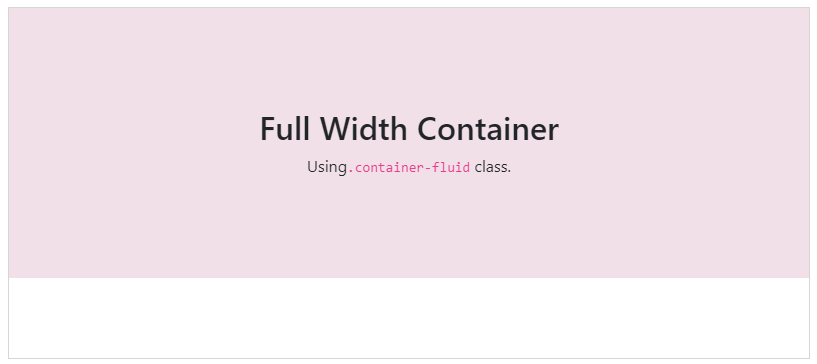
You can create a full width container by using the .container-fluid class as shown below.

<div class = "container-fluid">

...

</div>

The output will look something like:



**Bootstrap Grid**

Bootstrap grid is built with flexbox and allows up to 12 columns across the page. provides responsive, mobile first fluid grid system which scales the columns as the device or viewport size increases.

**Working of Grid System**

* Rows must be placed within a .container class for proper alignment and padding.
* For responsive width use .container class and for fixed width across all viewport, use the .container-fluid class.
* 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.
* Columns contain padding for controlling the space between them.
* If you place more than 12 columns in a row, then the columns will be placed in a new line.
* Columns create gaps between column content via padding. Therefore, you can remove the margin from rows and padding from columns with .no-gutters class on the row.
* You can make grid system responsive by using five grid breakpoints such as extra small, small, medium, large, and extra large.
* Predefined grid classes like .col-4 are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.

**Grid Classes**

The Bootstrap grid system has five classes:

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

You do not need to remember them all, just need to put into practice and you will understand it all. To make it easy, we will recall the text during the challenges as well.

**Examples using Grid Structure**

Suppose we want a fluid structure (i.e 100% width of the site) and 2 boxes in parallel, both of equal size. Something like this image below:



To achieve this we will use .container-fluid class along with the row and col. The code will look something like this below:

<div class="container-fluid">

<div class="row">

<div class="col" style="background-color:#ccc; padding:50px;">.col</div>

<div class="col" style="background-color:#f0f0f0;padding:50px;">.col</div>

</div>

</div>

Note: The inline styling in the above example is just for background color and additional padding for more visiblity.

Lets take another example. Suppose we need four equal-width columns starting at tablets and scaling to extra large desktops. On mobile phones or screens that are less than 576px wide, the columns will automatically stack on top of each other.



To achieve this we will use .container-fluid class along with the row and col. The code will look something like this below:

<div class="row">

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

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

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

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

</div>

In the above code, if you notice we have used .col-sm-3 rather .col used previously. That is becaue .col-sm- (small devices – screen width equal to or greater than 576px) handles the smaller devices screen i.e your mobile.

### Bootstrap Content

Bootstrap 4 uses collection of content methods for displaying the text, blocks of code, responsive images, data in a tabular format etc on the web page.

To keep it brief, we will only look at the typography and images.

##### Typography

The typography feature creates headings, paragraphs, lists and other inline elements. It specifies how text elements should be rendered on the web page.

The Bootstrap 4 uses a default font-size of 16px and the line-height of 1.5. The default font-family is “Helvetica Neue”, Helvetica, Arial, sans-serif. In addition, all <p> elements have margin-top: 0 and margin-bottom: 1rem (16px by default).

Let’s look at how some of the HTML elements can be twisted using bootstrap.

# **1. Headings**

All HTML headings, <h1> through <h6>, are available. You can also use the class .h1 through .h6 incase you want to match the font styling of a heading but cannot use the associated HTML element. For example, we can get the paragraphs to have same styling as H1 element by just using .h1 class.

<p class="h1">h1. Bootstrap heading</p>

# **2. Display headings**

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

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

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

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

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

To learn more about the Typography, can visit the [official documentation](https://getbootstrap.com/docs/4.1/content/typography/)

##### Responsive images

Images in Bootstrap are made responsive with .img-fluid. The code “max-width: 100%; and height: auto;” are applied to the image so that it scales with the parent element. Example for the code is give below:

<img src="..." class="img-fluid" alt="Responsive image">

The additional classes that are available with images are:

1. The .rounded class adds rounded corners to an image.  
2. The .rounded-circle class shapes the image to a circle.  
3. The .img-thumbnail class shapes the image to a thumbnail (bordered).

To learn more about the images, can visit the [official documentation](https://getbootstrap.com/docs/4.1/content/images/)

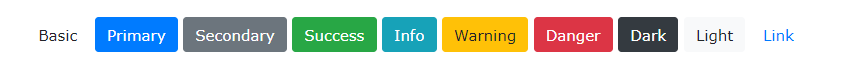
### Bootstrap Components

Bootstrap 4 has dozen of reusable components built to provide iconography, drop downs, input groups, navigation, alerts, and much more.

To keep it brief, we will only look at the buttons and forms.

##### Buttons

“Bootstrap 4” provides different styles of buttons:



They can be generated using the class “btn” along with additional classes like “btn-primary”, “btn-secondary” etc. Each one can be used in your web apps / website depending upon the situation.

<button type="button" class="btn">Basic</button>

<button type="button" class="btn btn-primary">Primary</button>

<button type="button" class="btn btn-secondary">Secondary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-light">Light</button>

<button type="button" class="btn btn-link">Link</button>

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

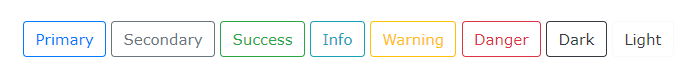
<a href="#" class="btn btn-info" role="button">Link Button</a>

<button type="button" class="btn btn-info">Button</button>

<input type="button" class="btn btn-info" value="Input Button">

<input type="submit" class="btn btn-info" value="Submit Button">

If in case you want to create outline/bordered buttons, then those can also be generated using “outline” keyword as a class:



<button type="button" class="btn btn-outline-primary">Primary</button>

<button type="button" class="btn btn-outline-secondary">Secondary</button>

<button type="button" class="btn btn-outline-success">Success</button>

<button type="button" class="btn btn-outline-info">Info</button>

<button type="button" class="btn btn-outline-warning">Warning</button>

<button type="button" class="btn btn-outline-danger">Danger</button>

<button type="button" class="btn btn-outline-dark">Dark</button>

<button type="button" class="btn btn-outline-light text-dark">Light</button>

To control sizes, use: .btn-lg class for large buttons or .btn-sm class for small buttons.

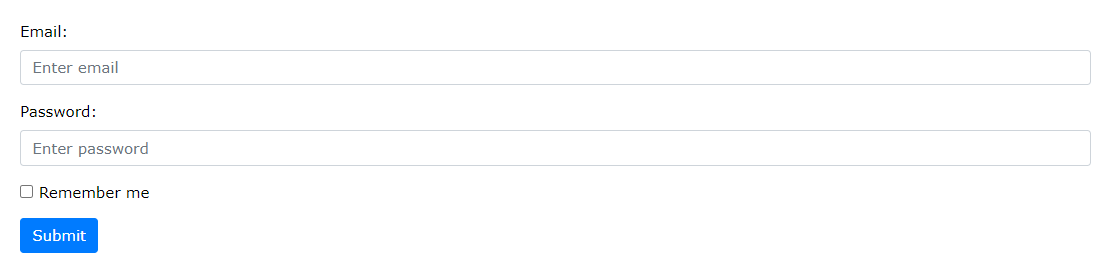
More options can be explored at their [official website](https://getbootstrap.com/docs/4.1/components/buttons/).

##### Bootstrap Forms

Bootstrap provides two types of form layouts:

### 1. Stacked (full-width) form

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



<form action="/action\_page.php">

<div class="form-group">

<label for="email">Email address:</label>

<input type="email" class="form-control" placeholder="Enter email" id="email">

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" placeholder="Enter password" id="pwd">

</div>

<div class="form-group form-check">

<label class="form-check-label">

<input class="form-check-input" type="checkbox"> Remember me

</label>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

Note we added a wrapper element with “.form-group” around each form control, to ensure proper margins.

### 2. Inline Form

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

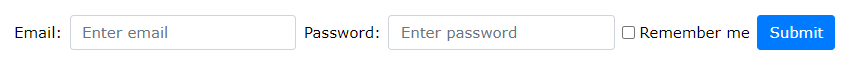
Note: This only applies to forms within viewports that are at least 576px wide. On screens smaller than 576px, it will stack horizontally.

To create a inline form, add class “.form-inline” to the

Top of Form

element

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



<form class="form-inline" action="/action\_page.php">

<label for="email">Email address:</label>

<input type="email" class="form-control" placeholder="Enter email" id="email">

<label for="pwd">Password:</label>

<input type="password" class="form-control" placeholder="Enter password" id="pwd">

<div class="form-check">

<label class="form-check-label">

<input class="form-check-input" type="checkbox"> Remember me

</label>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

More options can be explored at their [official website](https://getbootstrap.com/docs/4.1/components/forms/).

Bottom of Form

### Bootstrap Utilities

Bootstrap 4 uses collection of utilities for displaying borders, text color, embeding video etc on the web page.

To keep it brief, we will only look at the colors and float.

##### Color

Use the contextual classes to change the color of text, link (using classes like text-primary, text-secondary etc) and background color (using classes like bg-primary, bg-secondary etc) of an element.

The following example shows changing color of the text by using contextual classes.

<p class="text-primary">Color : text-primary </p>

<p class="text-secondary">Color : text-secondary</p>

<p class="text-success">Color : text-success</p>

<p class="text-danger">Color : text-danger</p>

<p class="text-warning">Color : text-warning</p>

<p class="text-info">Color : text-info</p>

<p class="text-black-50">Color : text-black-50</p>

<p class="text-dark">Color : text-dark</p>

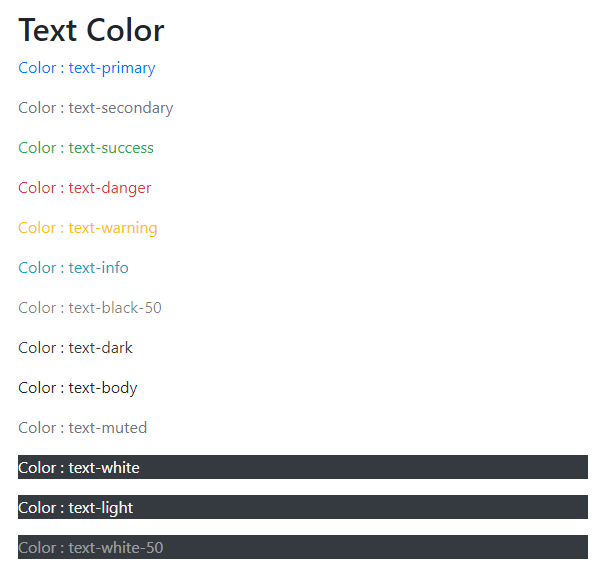
<p class="text-body">Color : text-body</p>

<p class="text-muted">Color : text-muted</p>

<p class="text-white bg-dark">Color : text-white</p>

<p class="text-light bg-dark">Color : text-light</p>

<p class="text-white-50 bg-dark">Color : text-white-50</p>



##### Float

The float utility is used to float an element to left or right side of the container by using the float-left or float-right classes. If you don’t want to float the element, then use the “float-none” class.

The following example shows the use of float classes over the div.

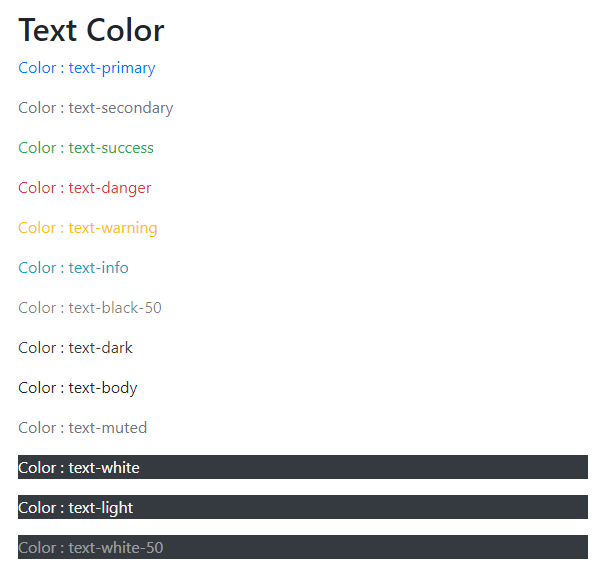
<div class = "float-left">Floating on left side (using 'float-left' class)</div>

<br>

<div class =" float-right">Floating on right side (using 'float-right' class)</div>

<br>

<div class = "float-none">No floating (using 'float-none' class)</div>



More options can be explored at their [official website](https://getbootstrap.com/docs/4.1/utilities/).