What is Bootstrap?

* Bootstrap is a free front-end framework for faster and easier web development
* Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
* Bootstrap also gives you the ability to easily create responsive designs

**What is Responsive Web Design?**  
  
Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

## Bootstrap Versions

This tutorial follows **Bootstrap 4**, which was released in 2018, as an upgrade to [Bootstrap 3](https://www.w3schools.com/bootstrap/default.asp), with new components, faster stylesheetc, more responsiveness, etc.

[Bootstrap 5](https://www.w3schools.com/bootstrap5/index.php) (released 2021) is the newest version of [Bootstrap](https://www.w3schools.com/bootstrap/default.asp); It supports the latest, stable releases of all major browsers and platforms. However, Internet Explorer 11 and down is not supported.

The main differences between Bootstrap 5 and Bootstrap 3 & 4, is that Bootstrap 5 has switched to [JavaScript](https://www.w3schools.com/js/default.asp) instead of [jQuery](https://www.w3schools.com/jquery/default.asp).

**Note:** [Bootstrap 3](https://www.w3schools.com/bootstrap/default.asp) and Bootstrap 4 is still supported by the team for critical bugfixes and documentation changes, and it is perfectly safe to continue to use them. However, new features will NOT be added to them.

Why Use Bootstrap?

Advantages of Bootstrap:

* **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
* **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
* **Mobile-first approach:** In Bootstrap, mobile-first styles are part of the core framework
* **Browser compatibility:** Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

Where to Get Bootstrap 4?

There are two ways to start using Bootstrap 4 on your own web site.

You can:

* Include Bootstrap 4 from a CDN
* Download Bootstrap 4 from getbootstrap.com

## Bootstrap 4 CDN

If you don't want to download and host Bootstrap 4 yourself, you can include it from a CDN (Content Delivery Network).

**jsDelivr** provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery:

### **jsDelivr:**

<!-- Latest compiled and minified CSS -->  
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">  
  
<!-- jQuery library -->  
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>  
  
<!-- Popper JS -->  
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>  
  
<!-- Latest compiled JavaScript -->  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>

## Downloading Bootstrap 4

If you want to download and host Bootstrap 4 yourself, go to <https://getbootstrap.com/>, and follow the instructions there.

## Create First Web Page With Bootstrap 4

**1. Add the HTML5 doctype**

Bootstrap 4 uses HTML elements and CSS properties that require the HTML5 doctype.

Always 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 4 is mobile-first**

Bootstrap 4 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 following <meta> tag inside the <head> element:

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

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. Containers**

Bootstrap 4 also requires a containing element to wrap site contents.

There are two container classes to choose from:

1. The .container class provides a responsive **fixed width container**
2. The .container-fluid class provides a **full width container**, spanning the entire width of the viewport

.container

.container-fluid

## Two Basic Bootstrap 4 Pages

The following example shows the code for a basic Bootstrap 4 page (with a responsive fixed width container):

### **Container Example**

<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>Bootstrap 4 Example</title>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">  
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>  
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>  
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>  
</head>  
<body>  
  
<div class="container">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>  
</div>  
  
</body>  
</html>

The following example shows the code for a basic Bootstrap 4 page (with a full width container):

### **Container Fluid Example**

<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>Bootstrap 4 Example</title>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">  
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>  
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>  
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>  
</head>  
<body>  
  
<div class="container-fluid">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>  
</div>  
</body>  
</html>

# **Bootstrap 4 Containers**

## Containers

Bootstrap requires a containing element to wrap site contents.

Containers are used to pad the content inside of them, and there are two container classes available:

1. The .container class provides a responsive **fixed width container**
2. The .container-fluid class provides a **full width container**, spanning the entire width of the viewport

.container

.container-fluid

## Fixed Container

Use the .container class to create a responsive, fixed-width container.

Note that its width (max-width) will change on different screen sizes:

|  | **Extra small <576px** | **Small ≥576px** | **Medium ≥768px** | **Large ≥992px** | **Extra large ≥1200px** |
| --- | --- | --- | --- | --- | --- |
| max-width | 100% | 540px | 720px | 960px | 1140px |

Open the example below and resize the browser window to see that the container width will change at different breakpoints:

### **Example**

<div class="container">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>  
</div>

## Fluid Container

Use the .container-fluid class to create a full width container, that will always span the entire width of the screen (width is always 100%):

### **Example**

<div class="container-fluid">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>  
</div>

## Container Padding

By default, containers have 15px left and right padding, with no top or bottom padding. Therefore, we often use **spacing utilities**, such as extra padding and margins to make them look even better. For example, .pt-3 means "add a top padding of 16px":

### **Example**

<div class="container pt-3"></div>

## Container Border and Color

Other utilities, such as borders and colors, are also often used together with containers:

### **Example**

# **My First Bootstrap Page**

This container has a border and some extra padding and margins.

# **My First Bootstrap Page**

This container has a dark background color and a white text, and some extra padding and margins.

# **My First Bootstrap Page**

This container has a blue background color and a white text, and some extra padding and margins.

<div class="container p-3 my-3 border"></div>  
  
<div class="container p-3 my-3 bg-dark text-white"></div>  
  
<div class="container p-3 my-3 bg-primary text-white"></div>

| **Class** | **Extra small <576px** | **Small ≥576px** | **Medium ≥768px** | **Large ≥992px** | **Extra large ≥1200px** |
| --- | --- | --- | --- | --- | --- |
| .container-sm | 100% | 540px | 720px | 960px | 1140px |
| .container-md | 100% | 100% | 720px | 960px | 1140px |
| .container-lg | 100% | 100% | 100% | 960px | 1140px |
| .container-xl | 100% | 100% | 100% | 100% | 1140px |

## Responsive Containers

You can also use the .container-sm|md|lg|xl classes to create responsive containers.

The max-width of the container will change on different screen sizes/viewports:

### **Example**

<div class="container-sm">.container-sm</div>  
<div class="container-md">.container-md</div>  
<div class="container-lg">.container-lg</div>  
<div class="container-xl">.container-xl</div>

# **Bootstrap 4 Grids**

## Bootstrap 4 Grid System

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

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 |
| span 4 | | | | span 4 | | | | span 4 | | | |
| span 4 | | | | span 8 | | | | | | | |
| span 6 | | | | | | span 6 | | | | | |
| span 12 | | | | | | | | | | | |

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

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

## Grid Classes

The Bootstrap 4 grid system has five classes:

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

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

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

## Basic Structure of a Bootstrap 4 Grid

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

<!-- Control the column width, and how they should appear on different devices -->  
<div class="row">  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
</div>  
<div class="row">  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
</div>  
  
<!-- Or let Bootstrap automatically handle the layout -->  
<div class="row">  
  <div class="col"></div>  
  <div class="col"></div>  
  <div class="col"></div>  
</div>

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

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

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

## Three Equal Columns

.col

.col

.col

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

### **Example**

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

## Responsive Columns

.col-sm-3

.col-sm-3

.col-sm-3

.col-sm-3

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

### **Example**

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

## Unequal Responsive Columns

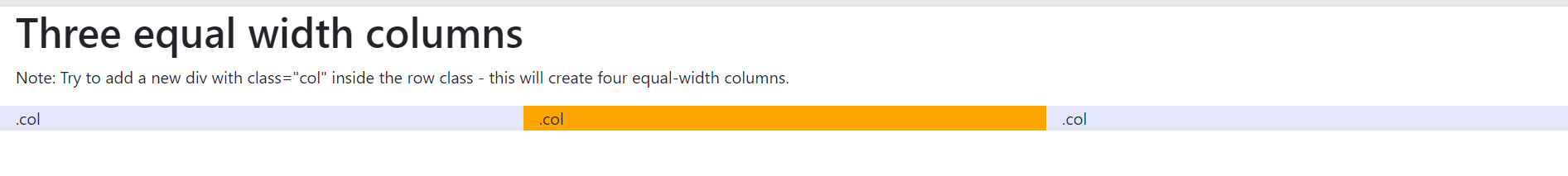
.col-sm-4

.col-sm-8

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

### **Example**

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



<body>

<div class="container-fluid">

<h1>Three equal width columns</h1>

<p>Note: Try to add a new div with class="col" inside the row class - this will create four equal-width columns.</p>

<div class="row">

<div class="col" style="background-color:lavender;">.col</div>

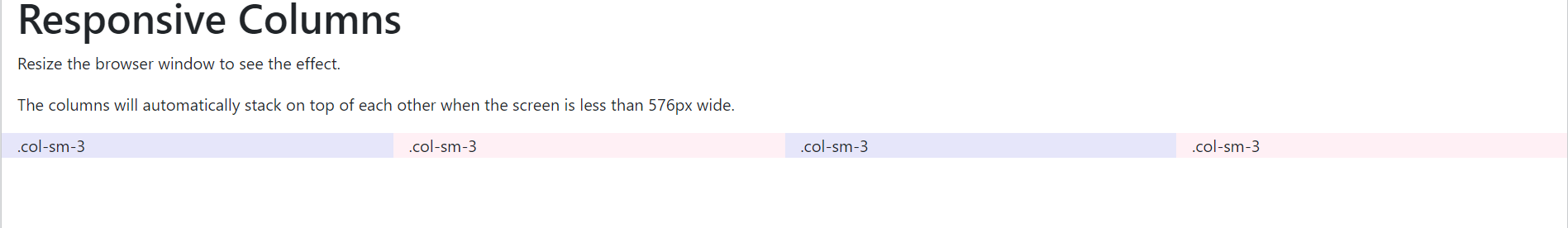
<div class="col" style="background-color:orange;">.col</div>

<div class="col" style="background-color:lavender;">.col</div>

</div>

</div>

</body>



<div class="container-fluid">

<h1>Responsive Columns</h1>

<p>Resize the browser window to see the effect.</p>

<p>The columns will automatically stack on top of each other when the screen is less than 576px wide.</p>

<div class="row">

<div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>

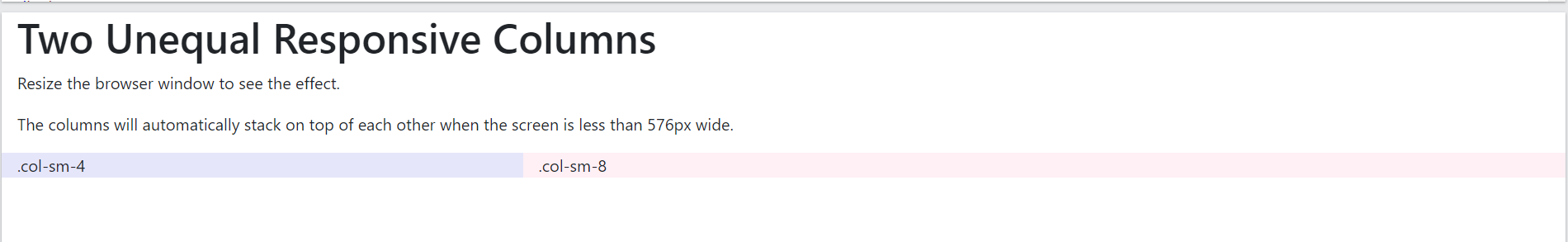
<div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>

<div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>

<div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>

</div>

</div>



<div class="container-fluid">

<h1>Two Unequal Responsive Columns</h1>

<p>Resize the browser window to see the effect.</p>

<p>The columns will automatically stack on top of each other when the screen is less than 576px wide.</p>

<div class="row">

<div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>

<div class="col-sm-8" style="background-color:lavenderblush;">.col-sm-8</div>

</div>

</div>

## Bootstrap 4 Alerts

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

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

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

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

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

×**Primary!** This alert box indicates an important action.

×**Secondary!** This alert box indicates a less important action.

×**Dark!** Dark grey alert box.

×**Light!** Light grey alert box.

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

### **Example**

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

## Alert Links

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

**Success!** You should [**read this message**](javascript:void(0)).

**Info!** You should [**read this message**](javascript:void(0)).

**Warning!** You should [**read this message**](javascript:void(0)).

**Danger!** You should [**read this message**](javascript:void(0)).

**Primary!** You should [**read this message**](javascript:void(0)).

**Secondary!** You should [**read this message**](javascript:void(0)).

**Dark!** You should [**read this message**](javascript:void(0)).

**Light!** You should [**read this message**](javascript:void(0)).

### **Example**

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

## Closing Alerts

[**×**](javascript:void(0))Click on the "x" symbol to the right to close me.

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

### **Example**

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

**Tip:** &times; (×) is an HTML entity that is the preferred icon for close buttons, rather than the letter "x".  
  
For a list of all HTML Entities, [visit our HTML Entities Reference](https://www.w3schools.com/charsets/ref_html_entities_4.asp).

## Animated Alerts

[**×**](https://www.w3schools.com/bootstrap4/bootstrap_alerts.asp)Click on the "x" symbol to the right to close me. I will "fade" out.

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

### **Example**

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

## Bootstrap 4 Table

## Bootstrap 4 Basic Table

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

The .table class adds basic styling to a table:

### **Example**

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

## Striped Rows

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

### **Example**

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

## Bordered Table

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

### **Example**

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

## Hover Rows

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

### **Example**

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

## Black/Dark Table

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

### **Example**

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

## Dark Striped Table

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

### **Example**

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

## Hoverable Dark Table

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

### **Example**

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

## Borderless Table

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

### **Example**

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

## Contextual Classes

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

### **Example**

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

The contextual classes that can be used are:

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

## Table Head Colors

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

### **Example**

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

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

## Small table

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

### **Example**

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

## Responsive Tables

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

### **Example**

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

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

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

### **Example**

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

# **Bootstrap 4 Colors**

## Text Colors

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

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

### **Example**

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

Dark grey text.

Body text.

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

### **Example**

[Muted link.](javascript:void(0)) [Primary link.](javascript:void(0)) [Success link.](javascript:void(0)) [Info link.](javascript:void(0)) [Warning link.](javascript:void(0)) [Danger link.](javascript:void(0)) [Secondary link.](javascript:void(0)) [Dark grey link.](javascript:void(0)) [Body/black link.](javascript:void(0)) [Light grey link.](javascript:void(0))

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

### **Example**

Black text with 50% opacity on white background

White text with 50% opacity on black background

## Background Colors

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

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

### **Example**

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

# **Bootstrap 4 Forms**

## Bootstrap 4's Default Settings

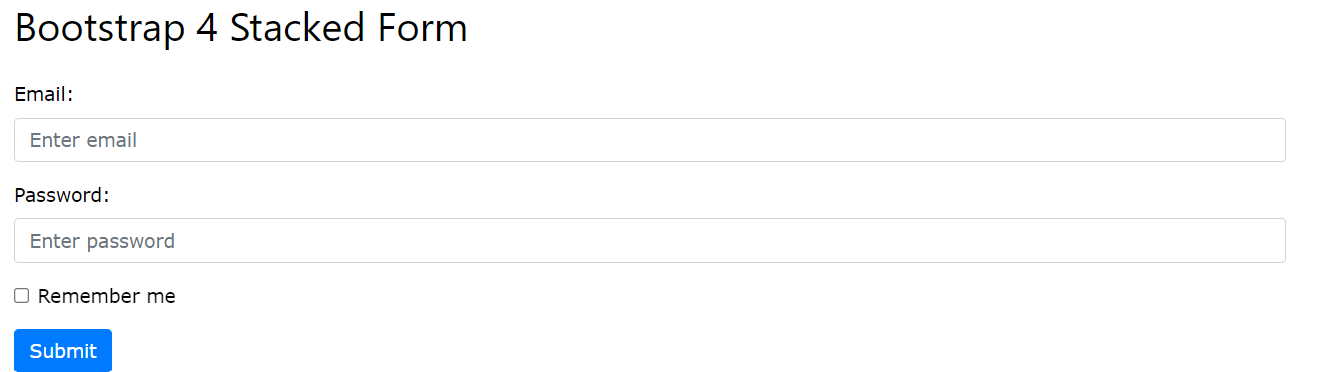
Form controls automatically receive some global styling with Bootstrap:

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

## Bootstrap 4 Form Layouts

Bootstrap provides two types of form layouts:

* Stacked (full-width) form
* Inline form

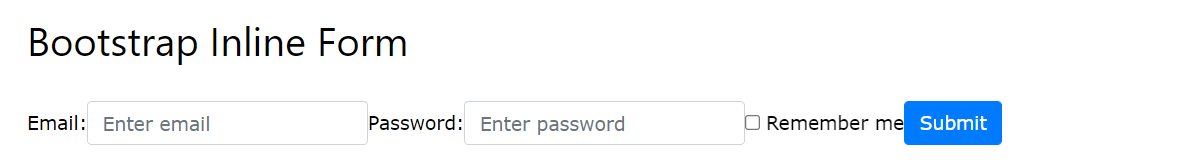


The following example creates a stacked form with two input fields, one checkbox, and a submit button.

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

### **Example**

<form action="/action\_page.php">  
  <div class="form-group">  
    <label for="email">Email address:</label>  
    <input type="email" class="form-control" 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>



Bottom of 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.**

Additional rule for an inline form:

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

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

### **Example**

<form class="form-inline" action="/action\_page.php">  
  <label for="email">Email address:</label>  
  <input type="email" class="form-control" 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>

## Inline Form with Utilities

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

Top of Form

Bottom of Form

### **Example**

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

## Form Row/Grid

You can also use columns (.col) to control the width and alignment of form inputs without using spacing utilities. Just remember to put them inside a .row container.

In the example below, we use two columns that will appear side by side. 

### **Example**

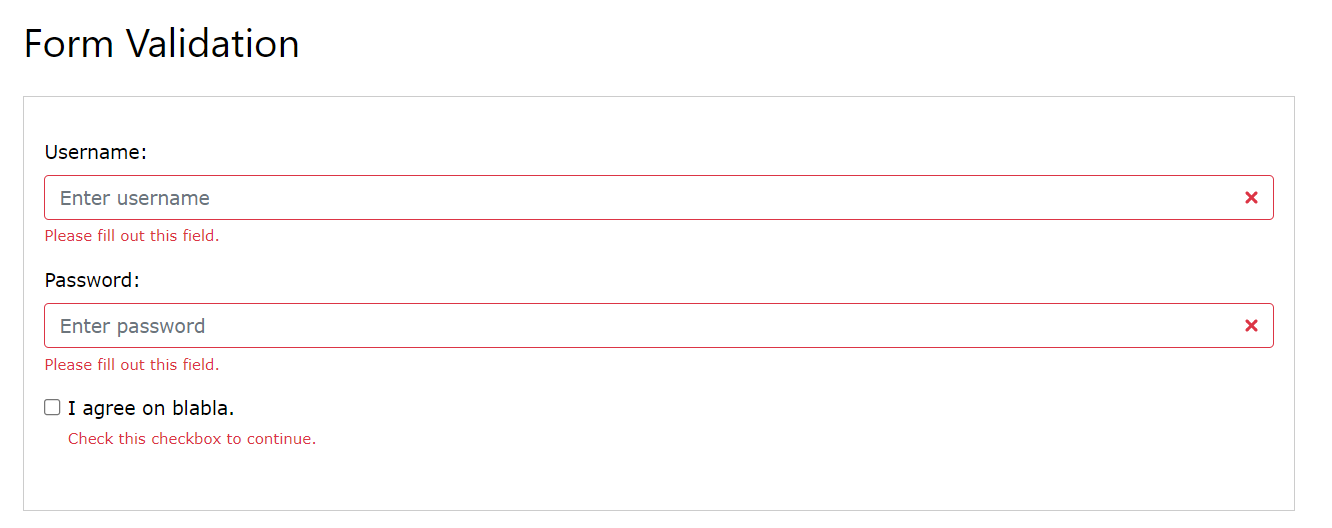
<form>  
  <div class="row">  
    <div class="col">  
      <input type="text" class="form-control" id="email" placeholder="Enter email" name="email">  
    </div>  
    <div class="col">  
      <input type="password" class="form-control" placeholder="Enter password" name="pswd">  
    </div>  
  </div>  
</form>

If you want less grid margins (override default column gutters), use .form-row instead of .row:



### **Example**

<form>  
  <div **class="form-row"**>  
    <div class="col">  
      <input type="text" class="form-control" id="email" placeholder="Enter email" name="email">  
    </div>  
    <div class="col">  
      <input type="password" class="form-control" placeholder="Enter password" name="pswd">  
    </div>  
  </div>  
</form>

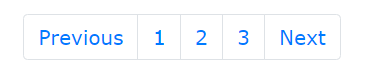
Bottom of Form

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

# **Bootstrap 4 Pagination**

## Basic Pagination

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



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>:

### **Example**

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

## Active State

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

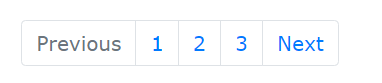


### **Example**

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

## Disabled State

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



### **Example**

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

## Pagination Sizing

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



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

### **Example**

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

## Pagination Alignment

Use utility classes to change the alignment of the pagination:

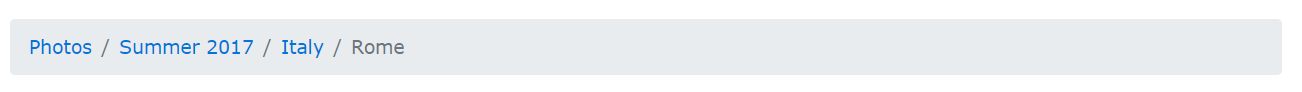


### **Example**

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

## Breadcrumbs

Another form for pagination, is breadcrumbs:



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

### **Example**

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

# **Bootstrap 4 Buttons**

## Button Styles

Bootstrap 4 provides different styles of buttons:

Basic Primary Secondary Success Info Warning Danger Dark Light Link



### **Example**

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

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

### **Example**

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

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

## Button Outline

Bootstrap 4 provides eight outline/bordered buttons:

Primary Secondary Success Info Warning Danger Dark Light

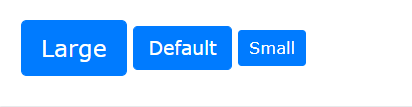


### **Example**

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

## Button Sizes

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



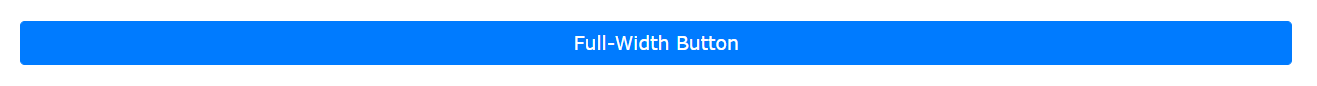
Large Default Small

### **Example**

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

## Block Level Buttons

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



Full-Width Button

### **Example**

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

## Active/Disabled Buttons

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



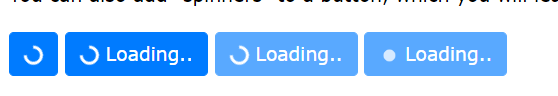
Active Primary Disabled Primary

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

### **Example**

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

## Spinner Buttons



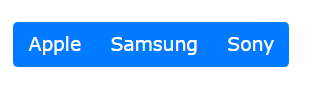
### **Example**

<button class="btn btn-primary">  
  <span class="spinner-border spinner-border-sm"></span>  
</button>  
  
<button class="btn btn-primary">  
  <span class="spinner-border spinner-border-sm"></span>  
  Loading..  
</button>  
  
<button class="btn btn-primary" disabled>  
  <span class="spinner-border spinner-border-sm"></span>  
  Loading..  
</button>  
  
<button class="btn btn-primary" disabled>  
  <span class="spinner-grow spinner-grow-sm"></span>  
  Loading..  
</button>

# **Bootstrap 4 Button Groups**

## Button Groups

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

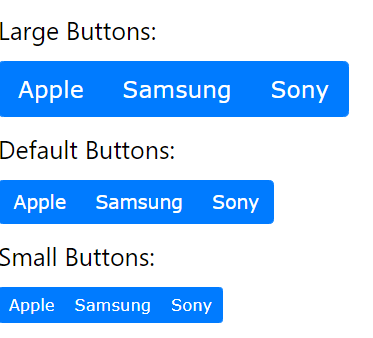


Use a <div> element with class .btn-group to create a button group:

### **Example**

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

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

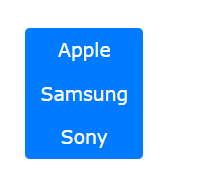


### **Example**

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

## Vertical Button Groups

Bootstrap 4 also supports vertical button groups:



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

### **Example**

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

# **Bootstrap 4 Images**

## Bootstrap 4 Image Shapes



## Rounded Corners

The .rounded class adds rounded corners to an image:

### **Example**

<img src="cinqueterre.jpg" class="rounded" alt="Cinque Terre">

## Circle

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

### **Example**

<img src="cinqueterre.jpg" class="rounded-circle" alt="Cinque Terre">

## Thumbnail

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

### **Example**

<img src="cinqueterre.jpg" class="img-thumbnail" alt="Cinque Terre">

## Aligning Images

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

### **Example**

<img src="paris.jpg" class="float-left">  
<img src="paris.jpg" class="float-right">

## 

## Centered Image

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



### **Example**

<img src="paris.jpg" class="mx-auto d-block">

## 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:

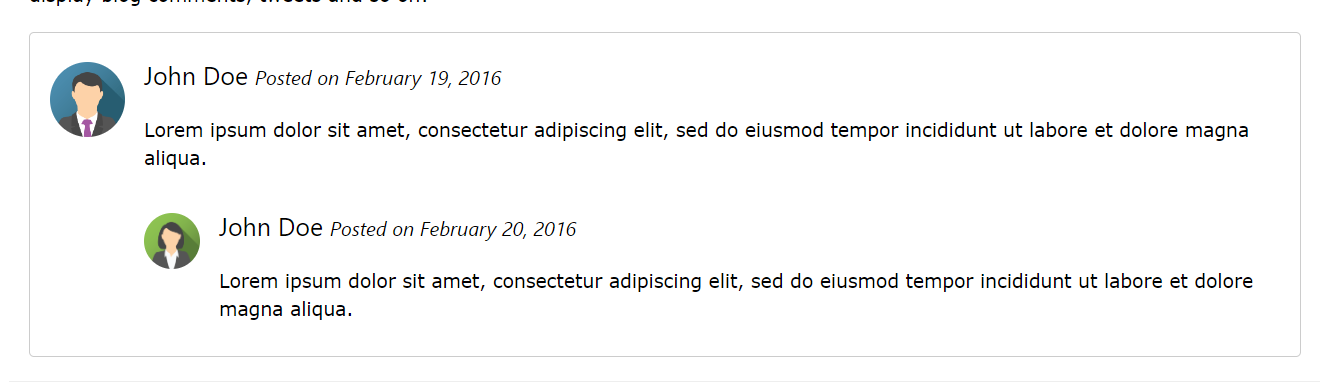
### **Example**

<img class="img-fluid" src="img\_chania.jpg" alt="Chania">

# **Bootstrap 4 Media Objects**

## Media Objects

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



## Basic Media Object



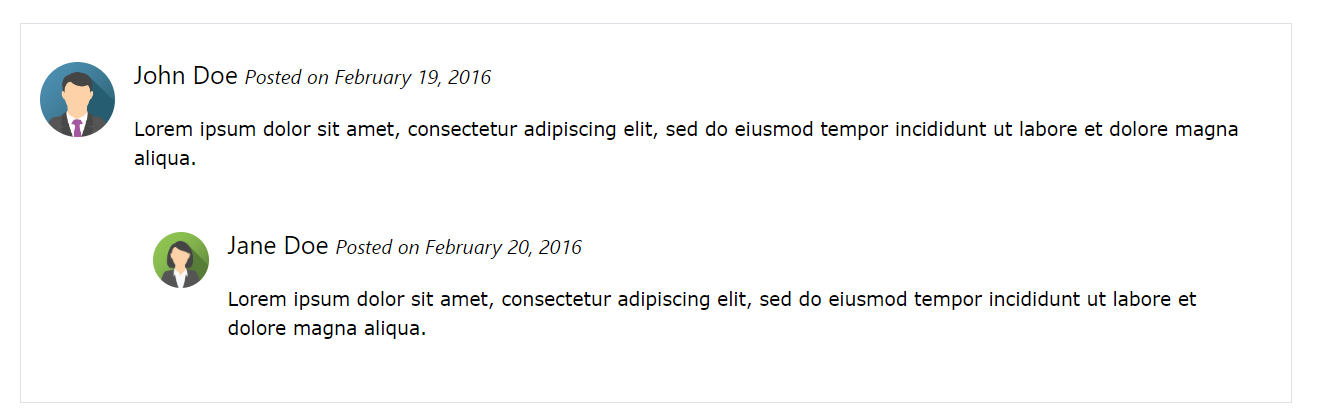
To create a media object, add the .media class to a container element, and place media content inside a child container with the .media-body class. Add padding and margins as needed, with the spacing utilities:

### **Example**

<div class="media border p-3">  
  <img src="img\_avatar3.png" alt="John Doe" class="mr-3 mt-3 rounded-circle" style="width:60px;">  
  <div class="media-body">  
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>

## Nested Media Objects

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

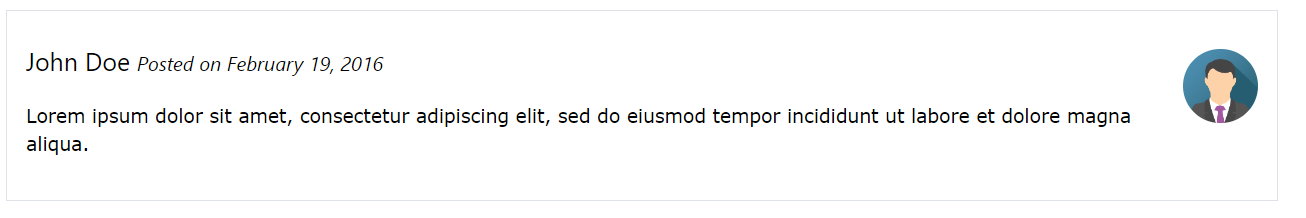


To nest media objects, place a new .media container inside the .media-body container:

### **Example**

<div class="media border p-3">  
  <img src="img\_avatar3.png" alt="John Doe" class="mr-3 mt-3 rounded-circle" style="width:60px;">  
  <div class="media-body">  
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>  
    <p>Lorem ipsum...</p>  
    <div class="media p-3">  
      <img src="img\_avatar2.png" alt="Jane Doe" class="mr-3 mt-3 rounded-circle" style="width:45px;">  
      <div class="media-body">  
        <h4>Jane Doe <small><i>Posted on February 20 2016</i></small></h4>  
        <p>Lorem ipsum...</p>  
      </div>  
    </div>   
  </div>  
</div>

## Right-Aligned Media Image



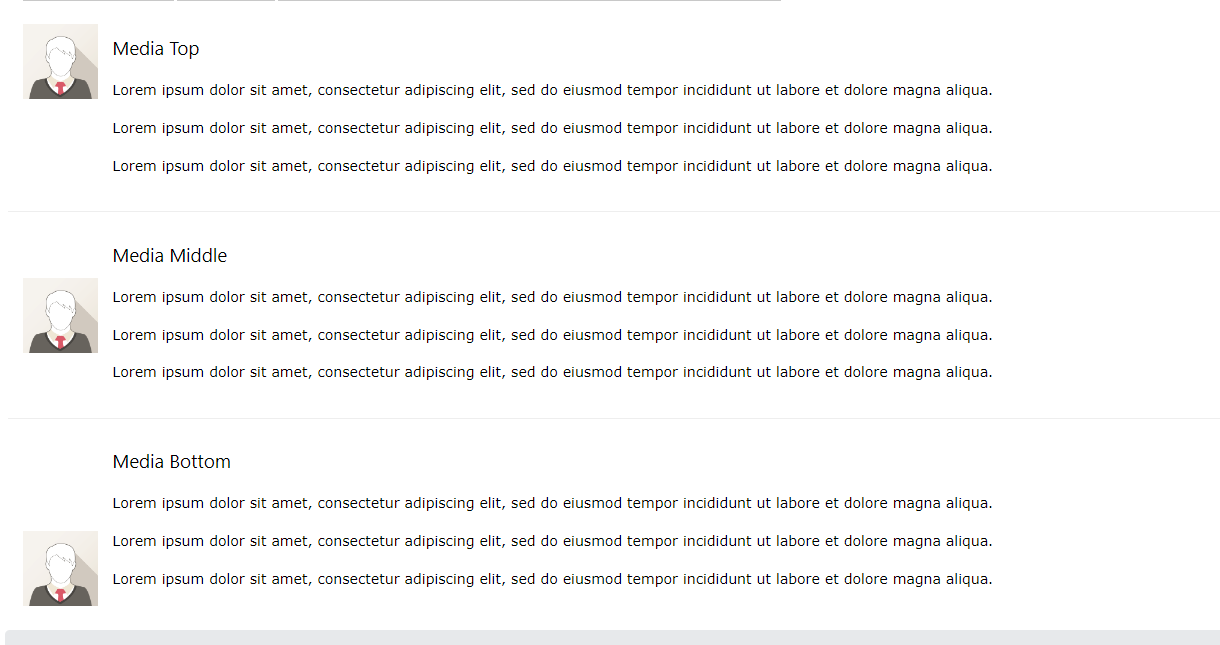
To right-align the media image, add the image after the .media-body container:

### **Example**

<div class="media border p-3">  
  <div class="media-body">  
    <h4>John Doe <small><i>Posted on February 19, 2016</i></small></h4>  
    <p>Lorem ipsum...</p>  
  </div>  
  <img src="img\_avatar3.png" alt="John Doe" class="ml-3 mt-3 rounded-circle" style="width:60px;">  
</div>

## Top, Middle or Bottom Alignment

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



### **Example**

<!-- Media top -->  
<div class="media">  
  <img src="img\_avatar1.png" class="align-self-start mr-3" style="width:60px">  
  <div class="media-body">  
    <h4>Media Top</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>  
  
<!-- Media middle -->  
<div class="media">  
  <img src="img\_avatar1.png" class="align-self-center mr-3" style="width:60px">  
  <div class="media-body">  
    <h4>Media Middle</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>  
  
<!-- Media bottom -->  
<div class="media">  
  <img src="img\_avatar1.png" class="align-self-end mr-3" style="width:60px">  
  <div class="media-body">  
    <h4>Media Bottom</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>

## Bootstrap 4 Grid System

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:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 |
| span 4 | | | | span 4 | | | | span 4 | | | |
| span 4 | | | | span 8 | | | | | | | |
| span 6 | | | | | | span 6 | | | | | |
| span 12 | | | | | | | | | | | |

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

Grid Classes

The Bootstrap 4 grid system has five classes:

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

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

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

Grid System Rules

Some Bootstrap 4 grid system rules:

* Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding
* Use rows to create horizontal groups of columns
* Content should be placed within columns, and only columns may be immediate children of rows
* Predefined classes like .row and .col-sm-4 are available for quickly making grid layouts
* Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows
* Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three .col-sm-4
* Column widths are in percentage, so they are always fluid and sized relative to their parent element
* The biggest **difference between Bootstrap 3 and Bootstrap 4** is that Bootstrap 4 now uses flexbox, instead of floats. One big advantage with flexbox is that grid columns without a specified width will automatically layout as "equal width columns" (and equal height). Example: Three elements with .col-sm will each automatically be 33.33% wide from the small breakpoint and up

## Basic Structure of a Bootstrap 4 Grid

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

<!-- Control the column width, and how they should appear on different devices -->  
<div class="row">  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
</div>  
  
<!-- Or let Bootstrap automatically handle the layout -->  
<div class="row">  
  <div class="col"></div>  
  <div class="col"></div>  
  <div class="col"></div>  
  <div class="col"></div>  
</div>

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

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

## Grid Options

The following table summarizes how the Bootstrap 4 grid system works across different screen sizes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Extra small (<576px)** | **Small (>=576px)** | **Medium (>=768px)** | **Large (>=992px)** | **Extra Large (>=1200px)** |
| **Class prefix** | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| **Grid behaviour** | Horizontal at all times | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints |
| **Container width** | None (auto) | 540px | 720px | 960px | 1140px |
| **Suitable for** | Portrait phones | Landscape phones | Tablets | Laptops | Laptops and Desktops |
| **# of columns** | 12 | 12 | 12 | 12 | 12 |
| **Gutter width** | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) |
| **Nestable** | Yes | Yes | Yes | Yes | Yes |
| **Offsets** | Yes | Yes | Yes | Yes | Yes |
| **Column ordering** | Yes | Yes | Yes | Yes | Yes |

## Bootstrap 4 Grid Example: Stacked-to-horizontal

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

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



### **Example: Stacked-to-horizontal**

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

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

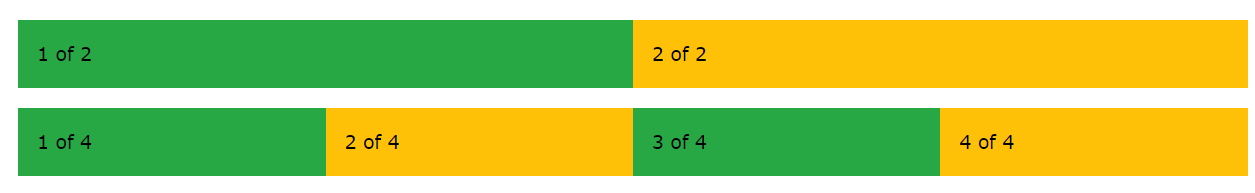
### **Example: Fluid container**

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

Auto Layout Columns

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

<!-- Two columns: 50% width on all screens, except for extra small (100% width) -->  
<div class="row">  
  <div class="col-sm">1 of 2</div>  
  <div class="col-sm">2 of 2</div>  
</div>  
  
<!-- Four columns: 25% width on all screens, except for extra small (100% width)-->  
<div class="row">  
  <div class="col-sm">1 of 4</div>  
  <div class="col-sm">2 of 4</div>  
  <div class="col-sm">3 of 4</div>  
  <div class="col-sm">4 of 4</div>  
</div>



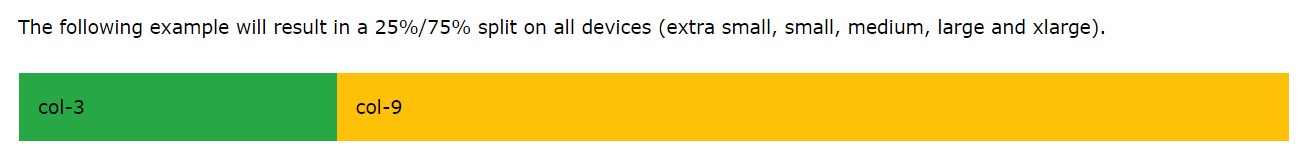
Extra Small Grid Example

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

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

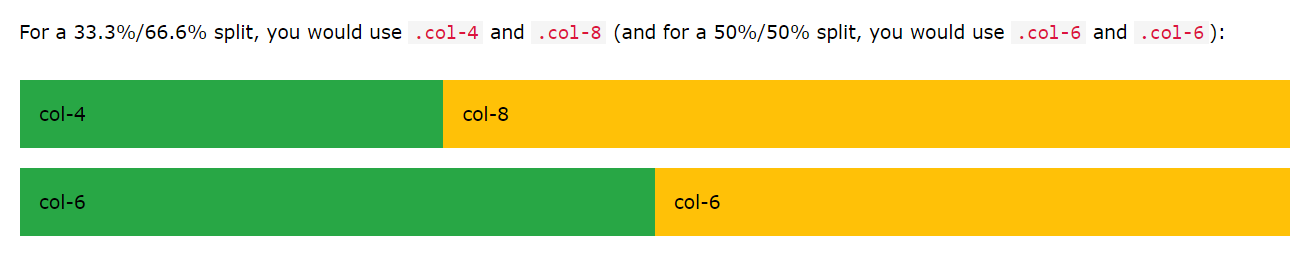
We will add the following classes to our two columns:

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



### **Example**

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

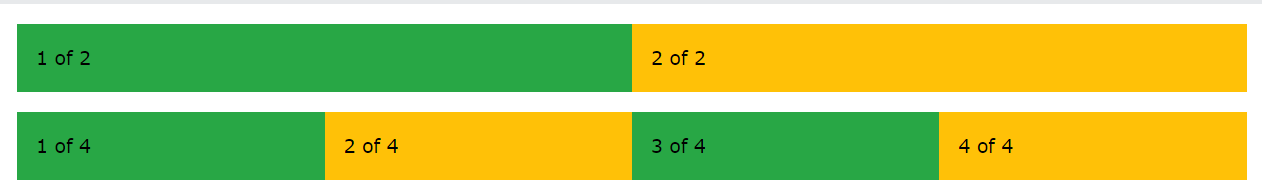


<!-- 33.3%/66.6% split -->  
<div class="container-fluid">  
  <div class="row">  
    <div class="col-4 bg-success">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-8 bg-warning">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>  
  
<!-- 50%/50% split -->  
<div class="container-fluid">  
  <div class="row">  
    <div class="col-6 bg-success">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-6 bg-warning">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

Auto Layout Columns

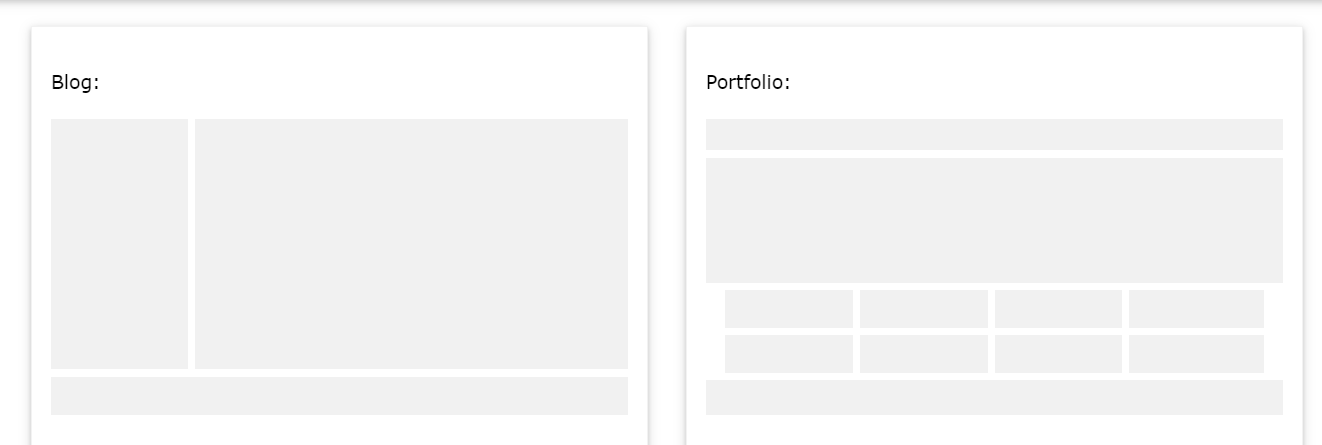
In Bootstrap 4, there is an easy way to create equal width columns for all devices: just remove the number from .col-\* and only use the .col class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width:

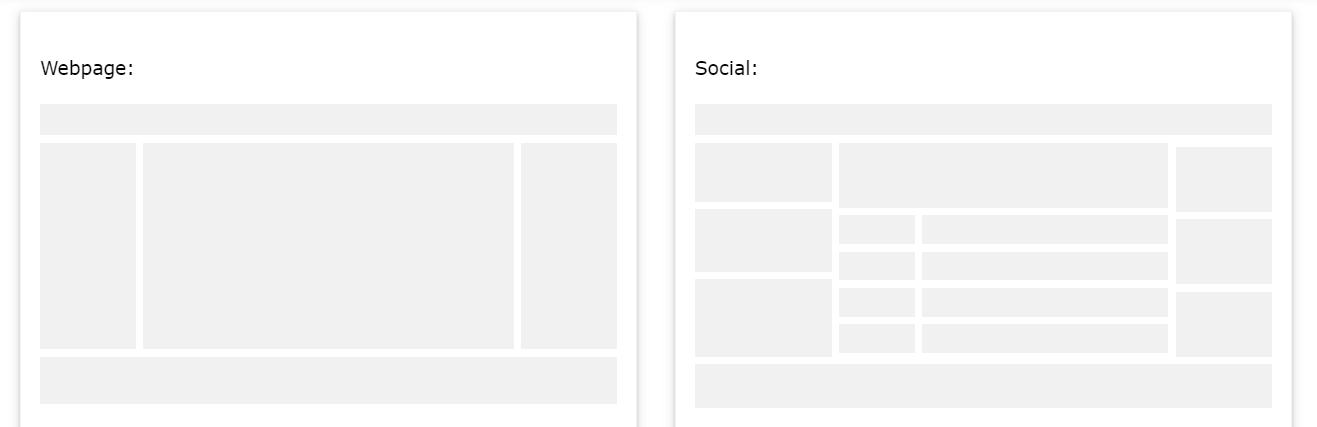
<!-- Two columns: 50% width-->  
<div class="row">  
  <div class="col">1 of 2</div>  
  <div class="col">2 of 2</div>  
</div>  
  
<!-- Four columns: 25% width-->  
<div class="row">  
  <div class="col">1 of 4</div>  
  <div class="col">2 of 4</div>  
  <div class="col">3 of 4</div>  
  <div class="col">4 of 4</div>  
</div>

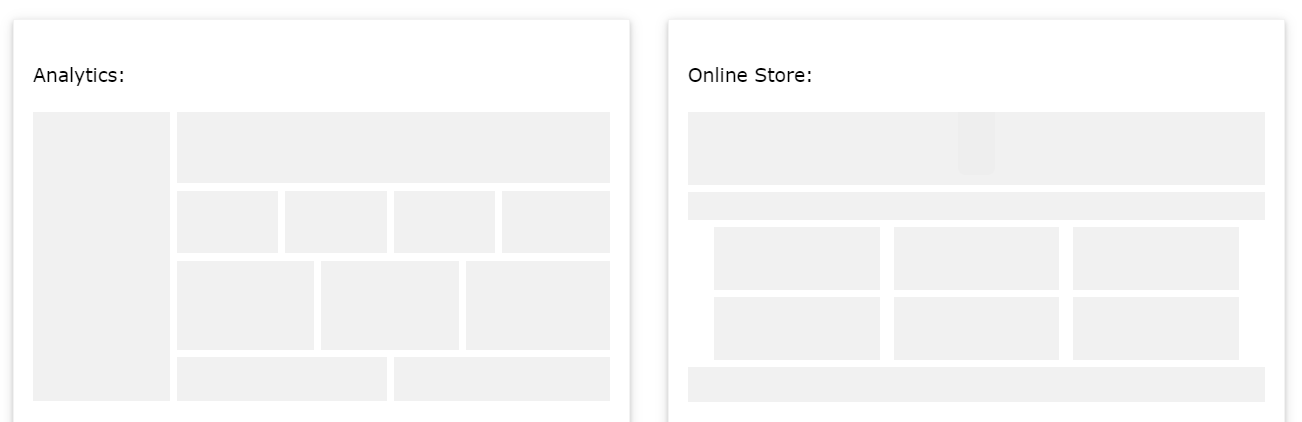


**NOTE: you can check all this exampeles with all the other grid type options like md,lg,xl**

# **Bootstrap Templates**

****

****

****

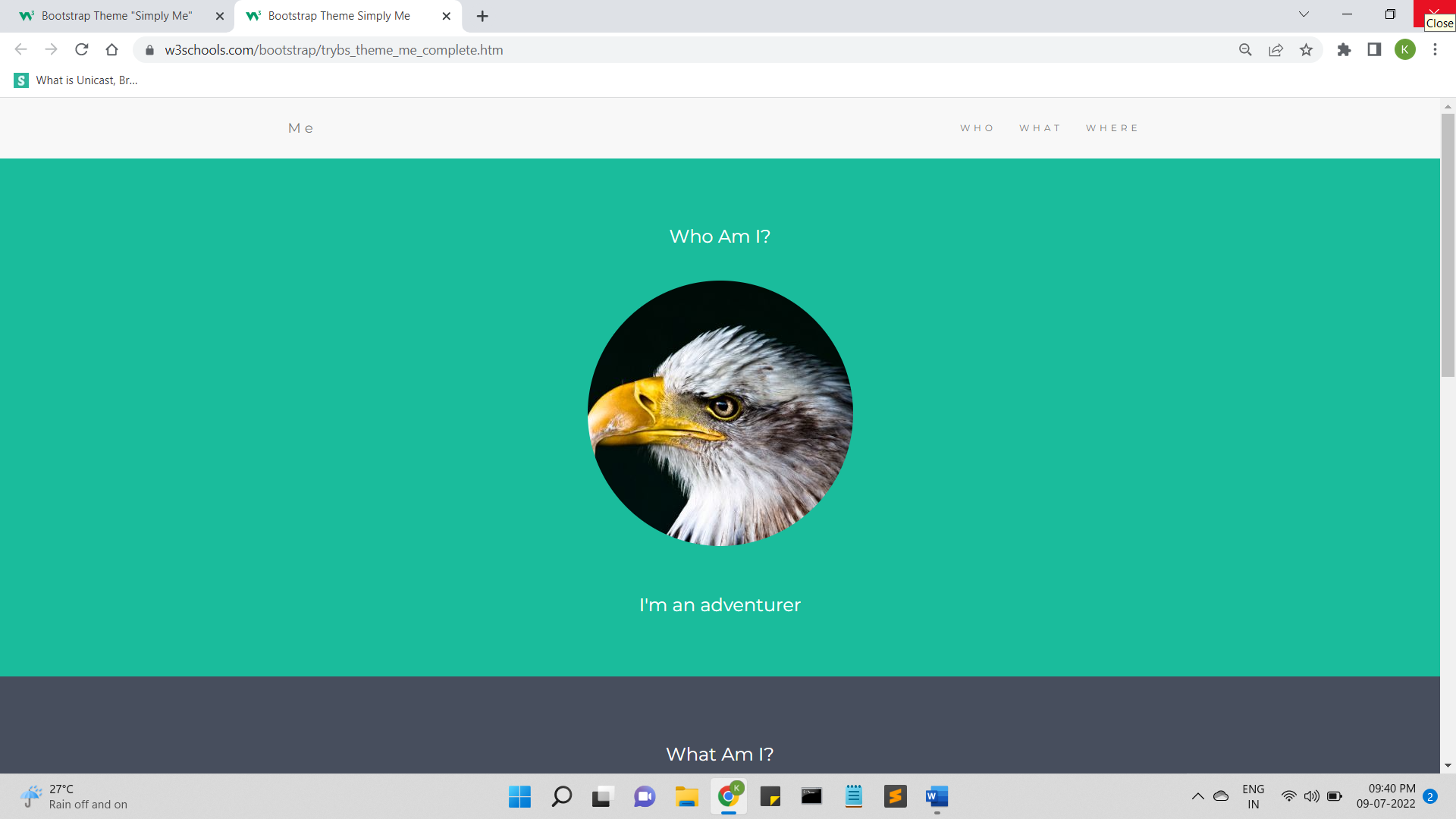
# **Bootstrap Theme "Simply Me"**

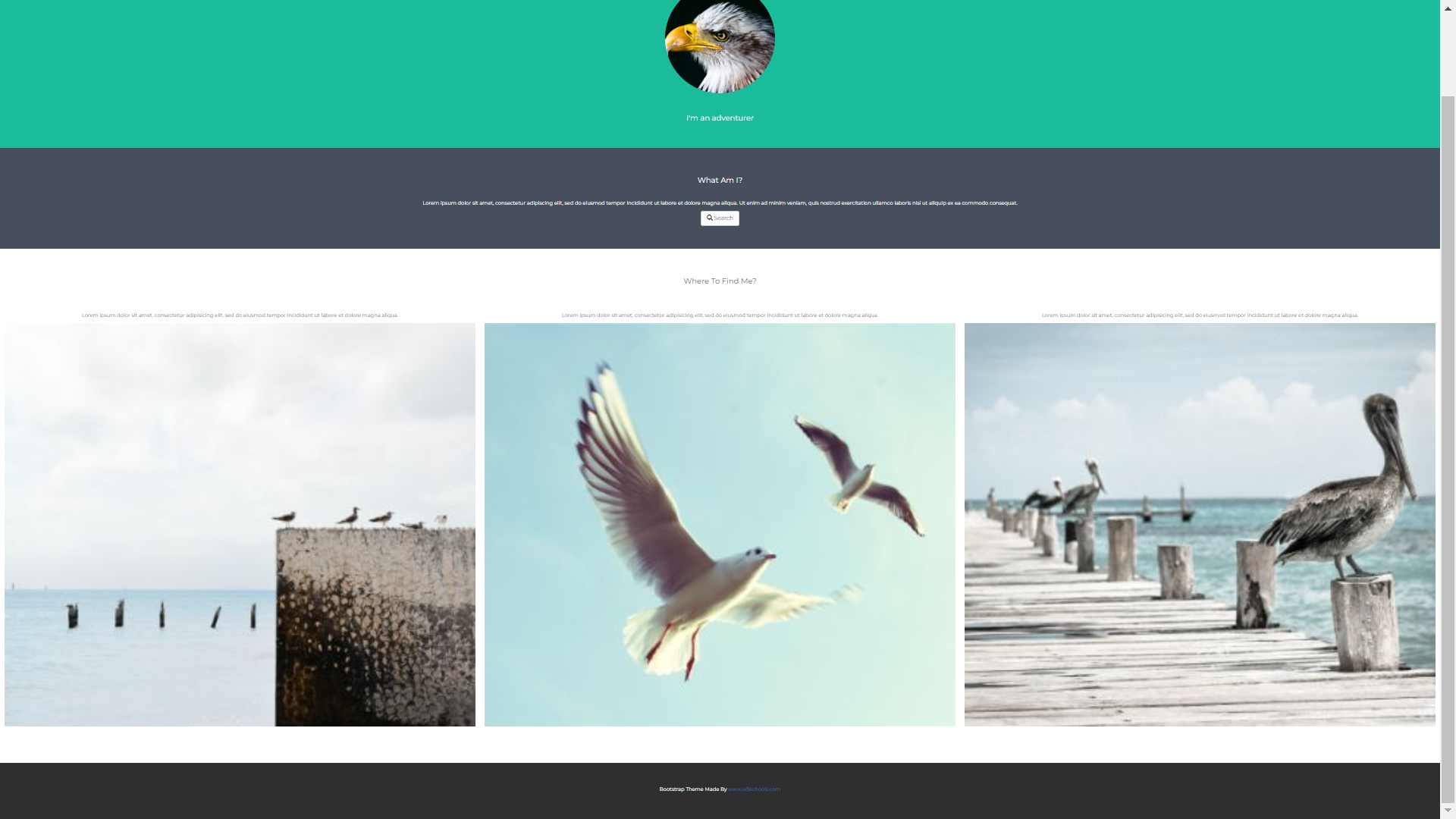
## Create a Theme: "Simply Me"

This page will show you how to build a Bootstrap theme from scratch.

We will start with a simple HTML page, and then add more and more components, until we have a fully functional, personal and responsive website.

The result will look like this, and you are free to modify, save, share, use or do whatever you want with it:





<!DOCTYPE html>

<html lang="en">

<head>

<!-- Theme Made By www.w3schools.com - No Copyright -->

<title>Bootstrap Theme Simply Me</title>

<meta charset="utf-8">

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

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<style>

body {

font: 20px Montserrat, sans-serif;

line-height: 1.8;

color: #f5f6f7;

}

p {font-size: 16px;}

.margin {margin-bottom: 45px;}

.bg-1 {

background-color: #1abc9c; /\* Green \*/

color: #ffffff;

}

.bg-2 {

background-color: #474e5d; /\* Dark Blue \*/

color: #ffffff;

}

.bg-3 {

background-color: #ffffff; /\* White \*/

color: #555555;

}

.bg-4 {

background-color: #2f2f2f; /\* Black Gray \*/

color: #fff;

}

.container-fluid {

padding-top: 70px;

padding-bottom: 70px;

}

.navbar {

padding-top: 15px;

padding-bottom: 15px;

border: 0;

border-radius: 0;

margin-bottom: 0;

font-size: 12px;

letter-spacing: 5px;

}

.navbar-nav li a:hover {

color: #1abc9c !important;

}

</style>

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-default">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

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

</div>

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

<ul class="nav navbar-nav navbar-right">

<li><a href="#">WHO</a></li>

<li><a href="#">WHAT</a></li>

<li><a href="#">WHERE</a></li>

</ul>

</div>

</div>

</nav>

<!-- First Container -->

<div class="container-fluid bg-1 text-center">

<h3 class="margin">Who Am I?</h3>

<img src="bird.jpg" class="img-responsive img-circle margin" style="display:inline" alt="Bird" width="350" height="350">

<h3>I'm an adventurer</h3>

</div>

<!-- Second Container -->

<div class="container-fluid bg-2 text-center">

<h3 class="margin">What Am I?</h3>

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

<a href="#" class="btn btn-default btn-lg">

<span class="glyphicon glyphicon-search"></span> Search

</a>

</div>

<!-- Third Container (Grid) -->

<div class="container-fluid bg-3 text-center">

<h3 class="margin">Where To Find Me?</h3><br>

<div class="row">

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

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

<img src="birds1.jpg" class="img-responsive margin" style="width:100%" alt="Image">

</div>

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

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

<img src="birds2.jpg" class="img-responsive margin" style="width:100%" alt="Image">

</div>

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

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

<img src="birds3.jpg" class="img-responsive margin" style="width:100%" alt="Image">

</div>

</div>

</div>

<!-- Footer -->

<footer class="container-fluid bg-4 text-center">

<p>Bootstrap Theme Made By <a href="https://www.w3schools.com">www.w3schools.com</a></p>

</footer>

</body>

</html>