

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.

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

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

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 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

Where to Get Bootstrap?

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

You can:

- Download Bootstrap from getbootstrap.com
- Include Bootstrap from a CDN

Downloading Bootstrap

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there.

Bootstrap CDN

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

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

MaxCDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/
```

```
css/bootstrap.min.css">
```

```
<!-- jQuery library -->
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

One advantage of using the Bootstrap CDN:

Many users already have downloaded Bootstrap from MaxCDN when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

jQuery

Bootstrap uses jQuery for JavaScript plugins (like modals, tooltips, etc). However, if you just use the CSS part of Bootstrap, you don't need jQuery.

Create First Web Page With Bootstrap

1. Add the HTML5 doctype

Bootstrap 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 3 is mobile-first

Bootstrap 3 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 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 Pages

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

Example

```
<!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/3.4.0/css/bootstrap.min.css">  
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/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>
```

[Try it Yourself »](#)

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

Example

```
<!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/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.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 Grids

[❏ Previous](#)[Next ❏](#)

Bootstrap Grid System

Bootstrap's grid system 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 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 automatically depending on the screen size.

Grid Classes

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 Structure of a Bootstrap Grid

The following is a basic structure of a Bootstrap grid:

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

First; create a row (`<div class="row">`). Then, add the desired number of columns (tags with appropriate `.col-*-*` classes). Note that numbers in `.col-*-*` should always add up to 12 for each row.

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

Three Equal Columns

`.col-sm-4`

`.col-sm-4`

`.col-sm-4`

The following example shows how to get a three equal-width columns starting at tablets and scaling to large desktops. On mobile phones or screens that are less than 768px wide, the columns will automatically stack:

Example

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

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

[Try it Yourself »](#)

Two Unequal 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 desktops:

Example

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

Bootstrap Tables

[❏ Previous](#)[Next ❏](#)

Bootstrap Basic Table

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

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

Example

Firstname	Lastname	Email
-----------	----------	-------

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

[Try it Yourself »](#)

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

[Try it Yourself »](#)

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

[Try it Yourself »](#)

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

[Try it Yourself »](#)

Condensed Table

The `.table-condensed` class makes a table more compact 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

[Try it Yourself »](#)

Contextual Classes

Contextual classes can be used to color table rows (`<tr>`) or table cells (`<td>`):

Example

Firstname	Lastname	Email
Default	Defaultson	def@somemail.com
Success	Doe	john@example.com
Danger	Moe	mary@example.com
Info	Dooley	july@example.com

Firstname	Lastname	Email
Warning	Refs	bo@example.com
Active	Activeson	act@example.com
Try it Yourself »		

The contextual classes that can be used are:

Class	Description
<code>.active</code>	Applies the hover color to the table row or table cell
<code>.success</code>	Indicates a successful or positive action
<code>.info</code>	Indicates a neutral informative change or action
<code>.warning</code>	Indicates a warning that might need attention
<code>.danger</code>	Indicates a dangerous or potentially negative action

Responsive Tables

The `.table-responsive` class creates a responsive table. The table will then scroll horizontally on small devices (under 768px). When viewing on anything larger than 768px wide, there is no difference:

Example

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

```
...
</table>
</div>
```

Try it Yourself »

Exercise:

Add a class attribute to style the table as a basic Bootstrap table.

```
<table 
```

Bootstrap Images

[❏ Previous](#)[Next ❏](#)

Bootstrap Image Shapes

Rounded Corners:



Circle:



Thumbnail:



Rounded Corners

The `.img-rounded` class adds rounded corners to an image (IE8 does not support rounded corners):

Example

```

```

[Try it Yourself »](#)

Circle

The `.img-circle` class shapes the image to a circle (IE8 does not support rounded corners):

Example

```

```

[Try it Yourself »](#)

Thumbnail

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

Example

```

```

[Try it Yourself »](#)

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-responsive` class to the `` tag. The image will then scale nicely to the parent element.

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

Example

```

```

[Try it Yourself »](#)

Image Gallery

You can also use Bootstrap's grid system in conjunction with the `.thumbnail` class to create an image gallery.



Lorem ipsum donec id elit non mi porta gravida at eget metus.



Lorem ipsum donec id elit non mi porta gravida at eget metus.



Lorem ipsum donec id elit non mi porta gravida at eget metus.

Note: You will learn more about the Grid System later in this tutorial (how to create a layout with different amount of columns).

Example

```
<div class="row">
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="/w3images/lights.jpg">
        
        <div class="caption">
          <p>Lorem ipsum...</p>
        </div>
      </a>
    </div>
  </div>
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="/w3images/nature.jpg">
```

```

        
        <div class="caption">
            <p>Lorem ipsum...</p>
        </div>
    </a>
</div>
</div>
<div class="col-md-4">
    <div class="thumbnail">
        <a href="/w3images/fjords.jpg">
            
            <div class="caption">
                <p>Lorem ipsum...</p>
            </div>
        </a>
    </div>
</div>
</div>
</div>

```

[Try it Yourself »](#)

Responsive Embeds

Also let videos or slideshows scale properly on any device.

Classes can be applied directly to <iframe>, <embed>, <video>, and <object> elements.

The following example creates a responsive video by adding an `.embed-responsive-item` class to an <iframe> tag (the video will then scale nicely to the parent element). The containing <div> defines the aspect ratio of the video:

Example

```

<div class="embed-responsive embed-responsive-16by9">
    <iframe class="embed-responsive-item" src="..."></iframe>
</div>

```

[Try it Yourself »](#)

What is aspect ratio?

The aspect ratio of an image describes the proportional relationship between its

width and its height. Two common video aspect ratios are 4:3 (the universal video format of the 20th century), and 16:9 (universal for HD television and European digital television).

You can choose between two aspect ratio classes:

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

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

[Try it Yourself »](#)

Exercise:

Use a Bootstrap class to shape the image as a circle.

```

```

Bootstrap Jumbotron and Page Header

[Previous](#) [Next](#)

Creating a Jumbotron

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

A jumbotron is displayed as a grey box with rounded corners. It also enlarges the font sizes of the text inside it.

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

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

Bootstrap Tutorial

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

Jumbotron Inside Container

Place the jumbotron inside the `<div class="container">` if you want the jumbotron to NOT extend to the edge of the screen:

Example

```
<div class="container">
  <div class="jumbotron">
    <h1>Bootstrap Tutorial</h1>
    <p>Bootstrap is the most popular HTML, CSS, and JS framework for
developing
    responsive, mobile-first projects on the web.</p>
  </div>
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```

[Try it Yourself »](#)

Jumbotron Outside Container

Place the jumbotron outside the `<div class="container">` if you want the jumbotron to extend to the screen edges:

Example

```
<div class="jumbotron">
  <h1>Bootstrap Tutorial</h1>
  <p>Bootstrap is the most popular HTML, CSS, and JS framework for developing
responsive,
  mobile-first projects on the web.</p>
</div>
<div class="container">
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```

[Try it Yourself »](#)

Creating a Page Header

A page header is like a section divider.

The `.page-header` class adds a horizontal line under the heading (+ adds some extra space around the element):

Example Page Header

Use a `<div>` element with class `.page-header` to create a page header:

Example

```
<div class="page-header">
  <h1>Example Page Header</h1>
</div>
```

Bootstrap Wells

Wells

The `.well` class adds a rounded border around an element with a gray background color and some padding:

Basic Well

Example

```
<div class="well">Basic Well</div>
```

[Try it Yourself »](#)

Well Size

Small Well

Normal Well

Large Well

Change the size of the well by adding the `.well-sm` class for small wells or `.well-lg` class for large wells:

Example

```
<div class="well well-sm">Small Well</div>
<div class="well well-lg">Large Well</div>
```

[Try it Yourself »](#)

Bootstrap Alerts

Alerts

Bootstrap provides an easy way to create predefined alert messages:

xSuccess! This alert box indicates a successful or positive action.

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

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

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

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

Example

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

<div class="alert alert-info">
  <strong>Info!</strong> Indicates a neutral informative change or action.
</div>

<div class="alert alert-warning">
  <strong>Warning!</strong> Indicates a warning that might need attention.
</div>

<div class="alert alert-danger">
  <strong>Danger!</strong> Indicates a dangerous or potentially negative
  action.
</div>
```

[Try it Yourself »](#)

Alert Links

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

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

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

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

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

Example

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

[Try it Yourself »](#)

Closing Alerts

[✕](#) Click on the "x" symbol to the right to close me.

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

Example

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

[Try it Yourself »](#)

The aria-* attribute and × explanation

To help improve accessibility for people using screen readers, you should include the `aria-label="close"` attribute, when creating a close button.

× (×) is an HTML entity that is the preferred icon for close buttons, rather than the letter "x".

For a list of all HTML Entities, [visit our HTML Entities Reference](#).

Animated Alerts

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

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

Example

```
<div class="alert alert-danger fade in">
```

[Try it Yourself »](#)

Exercise:

Add a Bootstrap "alert" class to display the result of a successful action.

```
<div class="alert">  
  ✕  
  Success!  
</div>
```

Bootstrap Buttons

[❏ Previous](#)[Next ❏](#)

Button Styles

Bootstrap provides different styles of buttons:

Basic Default Primary Success Info Warning Danger Link

To achieve the button styles above, Bootstrap has the following classes:

- `.btn`
- `.btn-default`
- `.btn-primary`
- `.btn-success`
- `.btn-info`
- `.btn-warning`
- `.btn-danger`
- `.btn-link`

The following example shows the code for the different button styles:

Example

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</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-link">Link</button>
```

[Try it Yourself »](#)

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

Example

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

[Try it Yourself »](#)

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 our examples. It should be a real URL to a specific page.

Button Sizes

Bootstrap provides four button sizes:

Large Medium Small XSmall

The classes that define the different sizes are:

- `.btn-lg`
- `.btn-md`
- `.btn-sm`
- `.btn-xs`

The following example shows the code for different button sizes:

Example

```
<button type="button" class="btn btn-primary btn-lg">Large</button>
<button type="button" class="btn btn-primary btn-md">Medium</button>
<button type="button" class="btn btn-primary btn-sm">Small</button>
<button type="button" class="btn btn-primary btn-xs">XSmall</button>
```

[Try it Yourself »](#)

Block Level Buttons

A block level button spans the entire width of the parent element.

Button 1Button 2

Add class `.btn-block` to create a block level button:

Example

```
<button type="button" class="btn btn-primary btn-block">Button 1</button>
```

[Try it Yourself »](#)

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 class `.disabled` makes a button unclickable:

Example

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

[Try it Yourself »](#)

Exercise:

Add a Bootstrap class to style the button properly as a "danger" button.

```
<button class="❌">Danger</button>
```

Bootstrap Button Groups

[❏ Previous](#)[Next ❏](#)

Button Groups

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

AppleSamsungSony

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>
```

[Try it Yourself »](#)

Tip: Instead of applying button sizes to every button in a group, use class `.btn-group-lg|sm|xs` to size all buttons in the 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>
```

[Try it Yourself »](#)

Vertical Button Groups

Bootstrap also supports vertical button groups:

AppleSamsungSony

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>
```

[Try it Yourself »](#)

Justified Button Groups

To span the entire width of the screen, use the `.btn-group-justified` class:

Apple

Samsung

Sony

Example with `<a>` elements:

Example

```
<div class="btn-group btn-group-justified">
  <a href="#" class="btn btn-primary">Apple</a>
  <a href="#" class="btn btn-primary">Samsung</a>
  <a href="#" class="btn btn-primary">Sony</a>
</div>
```

[Try it Yourself »](#)

Note: For `<button>` elements, you must wrap each button in a `.btn-group` class:

Example

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



```

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

```

[Try it Yourself »](#)

Nesting Button Groups & Dropdown Menus

AppleSamsung

Sony

Nest button groups to create dropdown menus:

Example

```

<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <div class="btn-group">
    <button type="button" class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">
      Sony <span class="caret"></span></button>
    <ul class="dropdown-menu" role="menu">
      <li><a href="#">Tablet</a></li>
      <li><a href="#">Smartphone</a></li>
    </ul>
  </div>
</div>
</div>

```

[Try it Yourself »](#)

Split Button Dropdowns

Sony

Example

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

Try it Yourself »

Exercise:

Add a Bootstrap class to group the buttons together.

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

Bootstrap Glyphicons

❏ PreviousNext ❏

Glyphicons

Bootstrap provides 260 glyphicons from the [Glyphicons](#) Halflings set.

Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc.

Here are some examples of glyphicons:

Envelope glyphicon:

Print glyphicon:

Search glyphicon:

Download glyphicon:

Glyphicon Syntax

A glyphicon is inserted with the following syntax:

```
<span class="glyphicon glyphicon-name"></span>
```

The *name* part in the syntax above must be replaced with the proper name of the glyphicon.

Glyphicon Example

The following example shows different ways to use glyphicons:

Example

```
<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
```

```
<p>Envelope icon as a link:
```

```
  <a href="#"><span class="glyphicon glyphicon-envelope"></span></a>
</p>
```

```
<p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
```

```
<p>Search icon on a button:
```

```
  <button type="button" class="btn btn-default">
    <span class="glyphicon glyphicon-search"></span> Search
```

```

    </button>
</p>
<p>Search icon on a styled button:
    <button type="button" class="btn btn-info">
        <span class="glyphicon glyphicon-search"></span> Search
    </button>
</p>
<p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
<p>Print icon on a styled link button:
    <a href="#" class="btn btn-success btn-lg">
        <span class="glyphicon glyphicon-print"></span> Print
    </a>
</p>

```

[Try it Yourself »](#)

Exercise:

Use a Glyphicon class value to make the span display as a "search" icon.

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

Bootstrap Badges and Labels

[Previous](#) [Next](#)

Badges

Badges are numerical indicators of how many items are associated with a link:

[News](#) **5**

[Comments](#) **10**

[Updates](#) **2**

The numbers (5, 10, and 2) are the badges.

Use the `.badge` class within `` elements to create badges:

Example

```
<a href="#">News <span class="badge">5</span></a><br>
<a href="#">Comments <span class="badge">10</span></a><br>
<a href="#">Updates <span class="badge">2</span></a>
```

[Try it Yourself »](#)

Badges can also be used inside other elements, such as buttons:

Primary **7**

The following example shows how to add badges to buttons:

Example

```
<button type="button" class="btn btn-
primary">Primary <span class="badge">7</span></button>
```

[Try it Yourself »](#)

Labels

Labels are used to provide additional information about something:

Example **New**

Example New

Example New

Example New

Example New

Example New

Use the `.label` class, followed by one of the six contextual classes `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` or `.label-danger`, within a `` element to create a label:

Example

```
<h1>Example <span class="label label-default">New</span></h1>
<h2>Example <span class="label label-default">New</span></h2>
<h3>Example <span class="label label-default">New</span></h3>
<h4>Example <span class="label label-default">New</span></h4>
<h5>Example <span class="label label-default">New</span></h5>
<h6>Example <span class="label label-default">New</span></h6>
```

[Try it Yourself »](#)

The following example shows all contextual label classes:

Default Label Primary Label Success Label Info Label Warning Label Danger Label

Example

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

Bootstrap Progress Bars

[❏ Previous](#)[Next ❏](#)

Basic Progress Bar

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

Bootstrap provides several types of progress bars.

A default progress bar in Bootstrap looks like this:



To create a default progress bar, add a `.progress` class to a `<div>` element:

Example

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="70"
    aria-valuemin="0" aria-valuemax="100" style="width:70%">
    <span class="sr-only">70% Complete</span>
  </div>
</div>
```

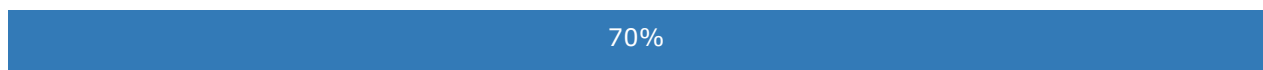
[Try it Yourself »](#)

Note: Progress bars are not supported in Internet Explorer 9 and earlier (because they use CSS3 transitions and animations to achieve some of their effects).

Note: To help improve accessibility for people using screen readers, you should include the `aria-*` attributes.

Progress Bar With Label

A progress bar with a label looks like this:



Remove the `.sr-only` class from the progress bar to show a visible percentage:

Example

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="70"
```

```
aria-valuemin="0" aria-valuemax="100" style="width:70%">
  70%
</div>
</div>
```

[Try it Yourself »](#)

Colored Progress Bars

Contextual classes are used to provide "meaning through colors".

The contextual classes that can be used with progress bars are:

- `.progress-bar-success`
- `.progress-bar-info`
- `.progress-bar-warning`
- `.progress-bar-danger`



The following example shows how to create progress bars with the different contextual classes:

Example

```
<div class="progress">
  <div class="progress-bar progress-bar-success" role="progressbar" aria-
valuenow="40"
    aria-valuemin="0" aria-valuemax="100" style="width:40%">
    40% Complete (success)
  </div>
</div>
```

```
<div class="progress">
  <div class="progress-bar progress-bar-info" role="progressbar" aria-
valuenow="50"
```



```

    aria-valuemin="0" aria-valuemax="100" style="width:50%">
      50% Complete (info)
    </div>
  </div>

  <div class="progress">
    <div class="progress-bar progress-bar-warning" role="progressbar" aria-
    valuenow="60"
      aria-valuemin="0" aria-valuemax="100" style="width:60%">
        60% Complete (warning)
      </div>
    </div>

  <div class="progress">
    <div class="progress-bar progress-bar-danger" role="progressbar" aria-
    valuenow="70"
      aria-valuemin="0" aria-valuemax="100" style="width:70%">
        70% Complete (danger)
      </div>
    </div>

```

[Try it Yourself »](#)

Striped Progress Bars

Progress bars can also be striped:



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

Example

```

<div class="progress">
  <div class="progress-bar progress-bar-success progress-bar-
  striped" role="progressbar"
    aria-valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width:40%">

```

```

    40% Complete (success)
  </div>
</div>

<div class="progress">
  <div class="progress-bar progress-bar-info progress-bar-
striped" role="progressbar"
    aria-valuenow="50" aria-valuemin="0" aria-valuemax="100" style="width:50%">
    50% Complete (info)
  </div>
</div>

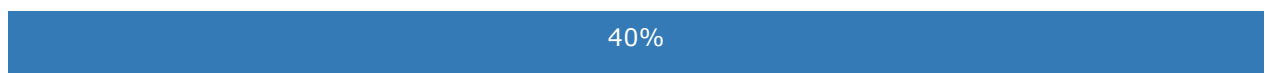
<div class="progress">
  <div class="progress-bar progress-bar-warning progress-bar-
striped" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100" style="width:60%">
    60% Complete (warning)
  </div>
</div>

<div class="progress">
  <div class="progress-bar progress-bar-danger progress-bar-
striped" role="progressbar"
    aria-valuenow="70" aria-valuemin="0" aria-valuemax="100" style="width:70%">
    70% Complete (danger)
  </div>
</div>

```

[Try it Yourself »](#)

Animated Progress Bar



Add class `.active` to animate the progress bar:

Example

```

<div class="progress">
  <div class="progress-bar progress-bar-striped active" role="progressbar"
    aria-valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width:40%">

```

```
    40%  
</div>  
</div>
```

[Try it Yourself »](#)

Stacked Progress Bars

Progress bars can also be stacked:



Create a stacked progress bar by placing multiple bars into the same `<div class="progress">`:

Example

```
<div class="progress">  
  <div class="progress-bar progress-bar-  
success" role="progressbar" style="width:40%">  
    Free Space  
  </div>  
  <div class="progress-bar progress-bar-  
warning" role="progressbar" style="width:10%">  
    Warning  
  </div>  
  <div class="progress-bar progress-bar-  
danger" role="progressbar" style="width:20%">  
    Danger  
  </div>  
</div>
```

[Try it Yourself »](#)

Exercise:

Add the correct classes to make this HTML code behave as a progress bar.

```
<div class="❌">  
  <div class="❌"  
    role="progressbar"  
    style="width:70%">  
  </div>  
</div>
```