



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

# Bootstrap

# What is Bootstrap?

- ▶▶ Bootstrap is the most popular framework for quickly styling your website.
- ▶▶ Bootstrap lets us skip writing CSS and focus instead on HTML.

# Using Bootstrap

Bootstrap (currently v3.3.5) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

# Using Bootstrap

## Bootstrap CDN

The folks over at [MaxCDN](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN](#) links.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
```

Copy

# Mobile Friendly

▶▶ Bootstrap 3 is mobile friendly by default.

- ▶▶ Responsive to screen size
- ▶▶ Touch screen zooming

▶▶ Add following to your <head> to enable touch zooming.

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

- ▶ width=device-width: sets width of page to screen width of the device
- ▶ initial-scale=1: sets initial zoom level when the page is first loaded

# 1) .Container

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

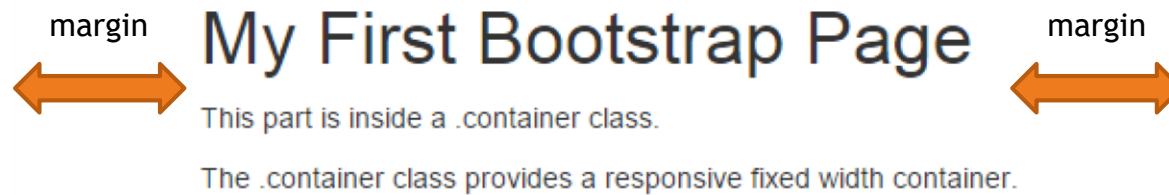
- Responsively add margins, center, and wrap your content.

## ► Example

```
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container class.</p>
  <p>The .container class provides a responsive fixed width container.</p>
</div>

</body>
```



## 2) Grid System

- Bootstrap divides a page into a grid of 12 columns and multiple rows for easier positioning of elements.
- Grid system is responsive and columns will rearrange automatically depending on the screen size.

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											



# Grid System

## Grid options

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	<b>Extra small devices</b> Phones (<768px)	<b>Small devices</b> Tablets (≥768px)	<b>Medium devices</b> Desktops (≥992px)	<b>Large devices</b> Desktops (≥1200px)
<b>Grid behavior</b>	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
<b>Container width</b>	None (auto)	750px	970px	1170px
<b>Class prefix</b>	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
<b># of columns</b>	12			
<b>Column width</b>	Auto	~62px	~81px	~97px

# Grid System - Basics

- ▶ Place columns inside for each row.

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

- ▶ Set each column width using class prefixes:

- ▶ `.col-xs-*`, `.col-sm-*`, `.col-md-*`, `.col-lg-*`

# Example

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

# Grid System - Multiple Devices

- ▶▶ Sometimes columns are stacked vertically on smaller screens
- ▶▶ Use `col-xs-*`, `col-sm-*`, `col-md-*` together to set different column widths for different screen sizes

# Example

Allow 2 columns in a row for mobile, 3 for desktop

```
<div class="row show-grid">  
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>  
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>  
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>  
</div>
```

On Desktop:

.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
---------------------	---------------------	---------------------

On Mobile:

.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
.col-xs-6 .col-md-4	

# Grid System - Column Resets

Sometimes at breakpoints, columns may be aligned in wrong places.

```
<div class="row show-grid">
  <div class="col-xs-6 col-sm-3">
    .col-xs-6 .col-sm-3
    <br>
    Resize your viewport or check it out on your phone for an example.
  </div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>
```

Problem:

.col-xs-6 .col-sm-3 Resize your viewport or check it out on your phone for an example.	.col-xs-6 .col-sm-3
	.col-xs-6 .col-sm-3
.col-xs-6 .col-sm-3	

Correct Output:

.col-xs-6 .col-sm-3 Resize your viewport or check it out on your phone for an example.	.col-xs-6 .col-sm-3
.col-xs-6 .col-sm-3	.col-xs-6 .col-sm-3

# Grid System - Column Resets

## Fix

```
<!-- Add the extra clearfix for only the required viewport -->  
<div class="clearfix visible-xs"></div>
```

- ▶▶ `.clearfix` clears `float` CSS attribute
- ▶▶ `visible-xs` is one of many `responsive utilities`
- ▶▶ `clearfix` is applied to only mobile devices (`visible-xs`)

# Grid System - Column Resets

## Fix

```
<div class="row">
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>

  <!-- Add the extra clearfix for only the required viewport -->
  <div class="clearfix visible-xs-block"></div>

  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>
```

Go to next line





# Responsive utilities

## Available classes

Use a single or combination of the available classes for toggling content across viewport breakpoints.

	<b>Extra small devices</b> Phones (<768px)	<b>Small devices</b> Tablets (≥768px)	<b>Medium devices</b> Desktops (≥992px)	<b>Large devices</b> Desktops (≥1200px)
<code>.visible-xs</code>	Visible	Hidden	Hidden	Hidden
<code>.visible-sm</code>	Hidden	Visible	Hidden	Hidden
<code>.visible-md</code>	Hidden	Hidden	Visible	Hidden
<code>.visible-lg</code>	Hidden	Hidden	Hidden	Visible
<code>.hidden-xs</code>	Hidden	Visible	Visible	Visible
<code>.hidden-sm</code>	Visible	Hidden	Visible	Visible
<code>.hidden-md</code>	Visible	Visible	Hidden	Visible
<code>.hidden-lg</code>	Visible	Visible	Visible	Hidden

# Grid System - Nested columns

Level 1: .col-md-9

Level 2: .col-md-6

Level 2: .col-md-6

```
<div class="row">
  <div class="col-md-9">
    Level 1: .col-md-9
    <div class="row">
      <div class="col-md-6">
        Level 2: .col-md-6
      </div>
      <div class="col-md-6">
        Level 2: .col-md-6
      </div>
    </div>
  </div>
</div>
```

# 3) Typography - Headings

Supports `<h1>` to `<h6>`, same as HTML

## EXAMPLE

h1. Bootstrap heading	Semibold 36px
h2. Bootstrap heading	Semibold 30px
h3. Bootstrap heading	Semibold 24px
h4. Bootstrap heading	Semibold 18px
h5. Bootstrap heading	Semibold 14px
h6. Bootstrap heading	Semibold 12px

# Typography - Alignment

## EXAMPLE

Left aligned text.

Center aligned text.

Right aligned text.

Justified text.

```
<p class="text-left">Left aligned text.</p>  
<p class="text-center">Center aligned text.</p>  
<p class="text-right">Right aligned text.</p>  
<p class="text-justify">Justified text.</p>
```

# Typography - Addresses

## EXAMPLE

### **Twitter, Inc.**

795 Folsom Ave, Suite 600  
San Francisco, CA 94107  
P: (123) 456-7890

### **Full Name**

[first.last@example.com](mailto:first.last@example.com)

```
<address>
  <strong>Twitter, Inc.</strong><br>
  795 Folsom Ave, Suite 600<br>
  San Francisco, CA 94107<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>

<address>
  <strong>Full Name</strong><br>
  <a href="mailto:#">first.last@example.com</a>
</address>
```

append **<br>** to preserve formatting

# Typography - Description

## EXAMPLE

### Description lists

A description list is perfect for defining terms.

```
<dl>
  <dt>...</dt>
  <dd>...</dd>
</dl>
```

# Typography - Horizontal Description

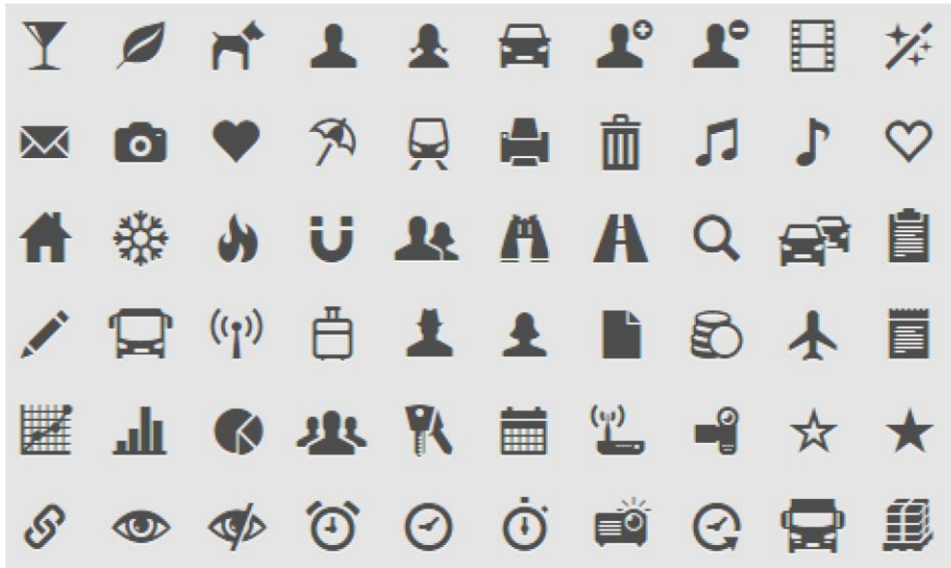
## EXAMPLE

**Description lists** A description list is perfect for defining terms.

```
<dl class="dl-horizontal">
  <dt>...</dt>
  <dd>...</dd>
</dl>
```

## 4) Glyphicons

- Bootstrap comes with 200 icons to use, named “glyphicons”



# Glyphicon Syntax

## ►► Syntax

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

## ►► Examples

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

Envelope icon: ✉

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

Search icon: 🔍

```
<p>Search icon on a styled button:  
  <button type="button" class="btn btn-info">  
    <span class="glyphicon glyphicon-search"></span> Search  
  </button>  
</p>
```

Search icon on a styled button:

🔍 Search



# Resizing Glyphicons

- ▶ To resize glyphicons, simply increase its font size in CSS!
- ▶ Example

```
<style>
  .features .glyphicon {
    font-size: 32px;
  }
</style>
```

## Output

With font-size: 32px 

With font-size: 14px 

```
<body>
  <div class="container">
    <div class="row features">
      <p>With font-size: 32px <span class="glyphicon glyphicon-briefcase"></span></p>
    </div>
    <div class="row">
      <p>With font-size: 14px <span class="glyphicon glyphicon-briefcase"></span></p>
    </div>
  </div>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
</body>
```

# 5) Forms

- ▶▶ Automatically applies some global styling with Bootstrap.
- ▶ Examples

Textarea

☐ Option one is this and that—be sure to include why it's great

☐ Option two is disabled

☒ Option one is this and that—be sure to include why it's great

☐ Option two can be something else and selecting it will deselect option one

☐ Option three is disabled

@

✓

# Forms

## ▶▶ Layouts supported by Bootstrap

- ▶▶ Vertical Form
- ▶▶ Horizontal Form
- ▶▶ Inline Form

## ▶▶ Tips on using forms with Bootstrap

- ▶▶ `<form role="form">` *Improves accessibility*
- ▶▶ `<div class="form-group">` *Improves spacing of form elements*
- ▶▶ `.form-control` *Give width of 100% to text form elements*

# Forms - Vertical Layout

Email:

Enter email

Password:

Enter password

☐ Remember me

Submit

```
<form role="form">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

# Forms - Horizontal Layout

**Email:**

**Password:**

☐ Remember me

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="control-label col-sm-2" for="email">Email:</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="email" placeholder="Enter email">
    </div>
  </div>
  <div class="form-group">
    <label class="control-label col-sm-2" for="pwd">Password:</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="pwd" placeholder="Enter password">
    </div>
  </div>
</form>
```

# Forms - Inline Layout

Email:  Password:  ☐ Remember me

```
<form class="form-inline" role="form">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

# Forms - Controls

Supports HTML form elements

▶▶ <input>

▶▶ <textarea>

▶▶ <input type="checkbox">

▶▶ <input type="radio">

▶▶ <select>

▶▶ ...

# Forms - Validation states

- ▶▶ Bootstrap allows you to give validation status to your form elements.
- ▶▶ Bootstrap supports

- ▶▶ `.has-warning`

- ▶▶ `.has-error`

- ▶▶ `.has-success`

Input with success

Input with warning

Input with error

☐ Checkbox with success

☐ Checkbox with warning

☐ Checkbox with error



## ►► Example

# Validation States - Input

Input with success

Input with warning

Input with error

```
<div class="form-group" has-success>
  <label class="control-label" for="inputSuccess1">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess1">
</div>
<div class="form-group" has-warning>
  <label class="control-label" for="inputWarning1">Input with warning</label>
  <input type="text" class="form-control" id="inputWarning1">
</div>
<div class="form-group" has-error>
  <label class="control-label" for="inputError1">Input with error</label>
  <input type="text" class="form-control" id="inputError1">
</div>
```

# Forms - Validation states with icons

To add feedback icons

1. add **has-feedback** class to parent
2. insert `<span>` specifying glyphicon for the control's feedback

```
<span class="glyphicon glyphicon-ok form-control-feedback"></span>
```

## EXAMPLE

Input with success

Input with warning

Input with error

# Validation states with Icons

Input with success

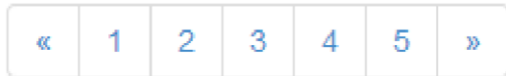
 

```
<div class="form-group has-success has-feedback">  
  <label class="control-label" for="inputSuccess2">Input with success</label>  
  <input type="text" class="form-control" id="inputSuccess2" aria-describedby="inputSuccess2Status">  
  <span class="glyphicon glyphicon-ok form-control-feedback" aria-hidden="true"></span>  
  <span id="inputSuccess2Status" class="sr-only">(success)</span>  
</div>
```

## 6) Pagination

►► You can use Bootstrap to divide your website into pages.

EXAMPLE



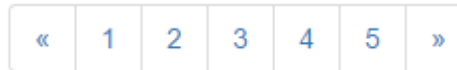
Simply add **pagination** class to ul element.

```
<ul class="pagination">
```

Each page is a list of href element. e.g. `<li><a href="#">1</a></li>`

Use **pagination-lg** or **pagination-sm** to size page button sizes

# Example - Pagination



```
<ul class="pagination">
  <li>
    <a href="#" aria-label="Previous">
      <span aria-hidden="true">&laquo;</span>
    </a>
  </li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li>
    <a href="#" aria-label="Next">
      <span aria-hidden="true">&raquo;</span>
    </a>
  </li>
</ul>
```

# 7) Pager

Simpler version with “Previous” or “Next” buttons

EXAMPLE

Previous

Next

```
<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>
```

You can also align buttons using `.previous` and `.next`

EXAMPLE

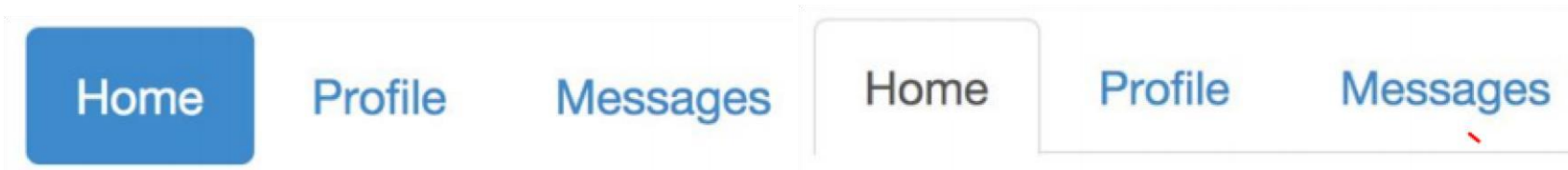
← Older

Newer →

```
<ul class="pager">
  <li class="previous"><a href="#">&larr; Older</a></li>
  <li class="next"><a href="#">Newer &rarr;</a></li>
</ul>
```

## 8) Navigation & Navigation Bars

### ►► Navigation (nav)



### ►► Navigation Bar (navbar)



# Navigation (nav)

- ▶▶ Use unordered list and give `.nav` class attribute.

- ▶ Example



Page 1   Page 2   Page 3

```
<div>
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Page 1</a></li>
    <li><a href="#">Page 2</a></li>
    <li><a href="#">Page 3</a></li>
  </ul>
</div>
```

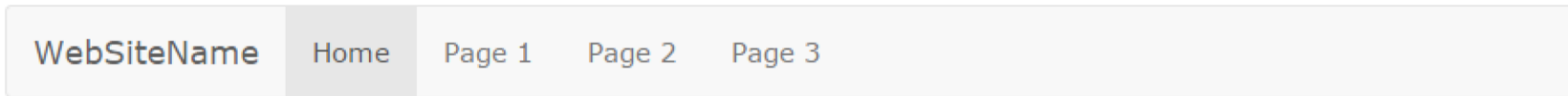
Makes navigation inline



# Navigation Bar (.navbar)

►► Creates a visible bar with navigation elements.

► Example

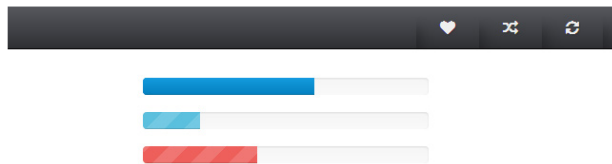


```
<body>
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Page 1</a></li>
        <li><a href="#">Page 2</a></li>
        <li><a href="#">Page 3</a></li>
      </ul>
    </div>
  </div>
</nav>
```

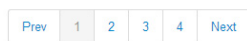
# REFERENCES

▶▶ Go over <http://getbootstrap.com/getting-started/#examples>

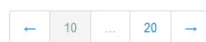
Task# 01: Use the Bootstrap feature progress bar and design it in the form of given output:



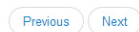
Task # 02: Create the “Pagination” tool of Bootstrap and give the following output:



**Centered**



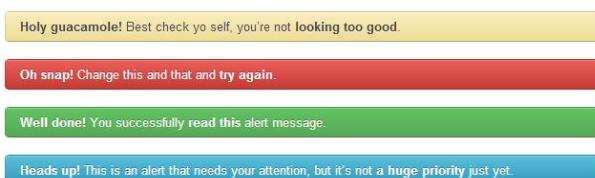
**Pager (Centered by default)**



**Pager (each link aligned to sides)**



Task # 03: Use the concept of Alerts in Bootstrap and provide the given output.



Task # 04 create a form using bootstrap for designing You can use JS or html validation for validating the form inputs.

**Company name\***

Please enter your company name. This field is required.

**Phone\***

Please enter a phone # like 123-456-7890. This field is required.

**Website\***

Please enter a valid URL. This field is required.