BOOTSTRAP

What is Bootstrap?

- **Bootstrap** is a free, open-source and is the most popular HTML, CSS, and JavaScript framework developed by twitter for creating responsive web applications.
- It can be used to create a web application built with any server side technologies like ASP.NET, JAVA, PHP etc.

History

- Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter (Twitter Blueprint). It was released as an open source product in August 2011 on GitHub.
- Version 2.0 supports Responsive web design.
- Version 3.0 adopted a mobile- first design.
- Version 4.0 alpha added Sass and Flexbox support.

Why use Bootstrap?

- Mobile first approach:

  The framework consists of Mobile first styles throughout the entire library instead of in separate files.

- Browser Support:

  It is supported by all popular browsers.

- Easy to get started:

  With just the knowledge of HTML and CSS anyone can get started with Bootstrap.

- Responsive design:

  Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles.

What Bootstrap Package Includes?

- **Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, background.
- **CSS:** Bootstrap comes with feature of global CSS settings, fundamental HTML elements and enhanced with extensible classes, and an advanced grid system.
- **Components:** Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, popovers, and much more.

What Bootstrap Package Includes?

- **JavaScript Plugins**: Bootstrap contains a variety of customized jQuery plugins. We can easily include them all, or one by one.
- **Customize:** We can customize Bootstrap's components and jQuery plugins to get your very own version.

Download Bootstrap

- You can download the latest version of Bootstrap from http://getbootstrap.com/.

Bootstrap CDN Links

- <!-- Latest compiled and minified CSS -->
- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

- <!-- Optional theme -->
- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" integrity="sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp" crossorigin="anonymous">

- <!-- Latest compiled and minified JavaScript -->
- <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>

Download Bootstrap

Original source files are included as listed below.

- Jquery.js
- Bootstrap.min.css
- Bootstrap.js

Example:

- <script type="text/javascript" src="js/jquery-3.1.1.js"></script>
- <script type="text/javascript" src="lib/bootstrap/js/bootstrap.js"></script>
- <link rel="stylesheet" type="text/css" href="lib/bootstrap/css/bootstrap.css">

Containers

.container class used for a responsive fixed-width.

Syntax:

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

.container-fluid for a full-width container.

Syntax:

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

What is a Grid?

- Grid is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal) lines used to structure the content.
- In web design, it is a very effective method to create a consistent layout rapidly and effectively using HTML and CSS.
- It organize and structure content, makes the websites easy to scan and reduces the load on users.

Grid System Description

- Grid System consists of 12 columns.
- The Grid system is used for creating page layouts through a series of rows and columns.
- Rows must be placed with a .container (fixed –width) or .container –fluid (full-width) for proper alignment and padding.
- Use rows to create horizontal group of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.

Grid System Structure

col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1 col-*-1

col- * - 6 col-*-6

col- * - 4 col- * - 4 col- * - 4

col- * - 4 col- * - 8

col- * - 12

Working of Bootstrap Grid System

Grid systems are used for creating page layouts through a series of rows and columns that house your content.

Here's how the Bootstrap grid system works:

- Rows must be placed within a .container class for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like .row and .col-*-* 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 twelve available columns you wish to span.
- If more than 12 columns are placed within a single row, each group of extra columns wrap onto a new line.

Grid Classes

The Bootstrap grid system has four classes:

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)

Device	Bootstrap Class
Extra small devices - Mobile Phones (< 768px)	.col-xs-*
Small devices - Tablets (≥768px)	.col-sm-*
Medium devices - Desktops (≥992px)	.col-md-*
Large devices - Desktops (≥1200px)	.col-lg-*

Grid options

The following table summarizes aspects of how Bootstrap grid system works across multiple devices:

	Extra small devices Phones (<768px)	**Small devices Tablets (≥768px)**	**Medium devices Desktops (≥992px)**	**Large devices Desktops (≥1200px)**
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	**.col-xs-**	**.col-sm-**	**.col-md-**	**.col-lg-**
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
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)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

BASIC GRID STRUCTURE

Following is basic structure of Bootstrap grid:

```
<div class="container">
    <div class="row">
        <div class="col-*-*"></div>
        <div class="col-*-*"></div>
    </div>
    <div class="row">...</div>
</div>
<div class="container">....
```

BASIC GRID STRUCTURE

Let us see some simple grid examples:

- Example: Stacked-to-horizontal
- Example: Medium and Large Device
- Example: Mobile, tablet, desktops

Bootstrap Grid System Example: Stacked-to-horizontal

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Stacked-to-horizontal

- Container class is added to ensure the proper centering and maximum width of the layout.
- Once container is added, we need to think in terms of rows. Add ** ... ** and columns ** ** inside the rows
- In the example we have two columns each made of 6 units wide.(6+6 =12).

Bootstrap Grid System Example: Medium and Large Device

- Here we have used **md** and **lg** column groups for support of medium and large size devices.
- For medium size devices, 2 div's will go from 50% by 50% of split.
- For large size devices, 2 div's will go form 33% by 66% of spilt.

Bootstrap Grid System Example: Mobile, Tablet, Desktops

**Now this gives us 3 different column layouts **

- **On a phone**, it will be 25% on the left, and 75% on the right.
- **On a tablet**, it will be 50%/50% again,
- **Large viewport**, it will be 33%/66%.
- 3 different layouts for each of the 3 responsive sizes.

Column Wrapping

- More than 12 columns are placed within a single row, Each group of extra columns will wrap onto a new line.

Responsive column resets

- Assume that four tiers of grids available, we run into issues where at certain breakpoints.
- Columns are don’t clear quite right as one is taller than the other.
- To fix this issue, We can use class .clearfix.

Offsetting columns

- It can be used for push columns over for more spacing.
- .col-md-offset-* class increases the left margin of a column range form 1 to 11.
- Offset class moves the content to the right depending upon the given value.

Nesting columns

- To nest your content with the default grid, add a new row and set of columns within an existing column.
- Nested rows should include a set of columns that add up to 12.

Column Ordering

- To write columns in an order and show them in one another one by using push and pull modifier classes where ranges form 1 to 11.
- Syntax:
- **.col-md-push-***
- **.col-md-pull-***

Typography

- Bootstrap uses Helvetica Neue, Helvetica, Arial, and sans-serif in its default font stack. Using typography feature of Bootstrap you can create headings, paragraphs, lists and other inline elements.
- Default font-size is 14px. with a line –height of 1.428.(Applied to <body> and all paragraphs).
- Paragraph <p> elements have a bottom margin that equals half their computed line-height (10px by default).

Headings:

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

<h1>I'm Heading1 h1</h1> Semibold 36px

<h2>I'm Heading2 h2</h2> Semibold 30px

<h3>I'm Heading3 h3</h3> Semibold 24px

<h4>I'm Heading4 h4</h4> Semibold 18px

<h5>I'm Heading5 h5</h5> Semibold 14px

<h6>I'm Heading6 h6</h6> Semibold 12px

Typography

INLINE SUBHEADINGS:

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

```
<h1>I'm Heading1 h1. <small>I'm secondary Heading1 h1</small></h1>

<h2>I'm Heading2 h2. <small>I'm secondary Heading2 h2</small></h2>

<h3>I'm Heading3 h3. <small>I'm secondary Heading3 h3</small></h3>

<h4>I'm Heading4 h4. <small>I'm secondary Heading4 h4</small></h4>

<h5>I'm Heading5 h5. <small>I'm secondary Heading5 h5</small></h5>

<h6>I'm Heading6 h6. <small>I'm secondary Heading1 h6</small></h6>
```

Inline text elements

Marked Text: Highlight the text using <mark> tag.

Deleted Text: Blocks of text have been deleted using <del> tag.

Strikethrough Text: Blocks of text that are no longer relevant using the <s> tag.

Inserted text: The <ins> tag defines a text that has been inserted into a document.

Underline Text: The <u>tag defines a underline text.

Small Text: Use the <small> tag to set the text at 85% the size of the parent.

Bold: Text with a heavier font-weight can be done by tag.

Italics: For Emphasizing of text with italics done by tag.

Alignment classes

Realign text can be done with text alignment classes.

- text-left class used for Left aligned text.
- text-right class used for Right aligned text.
- text-center class used for Center aligned text.
- text-justify class used for Justified text.
- text-nowrap class used for No wrap text.

Transformation classes

Transform text by using text capitalization classes.

- text-lowercase class used for Lowercased text.
- text-uppercase class used for Uppercased text.
- text-capitalize class used for Capitalized text.

Abbreviations

Basic Abbreviation :An abbreviation of the word attribute is attr.

Syntax :<abbr title="attribute">attr</abbr>

Initialism: Add .Initialism to an abbreviation for a slightly smaller font-size.

Syntax :<abbr title="HyperText Markup Language" class="initialism">HTML</abbr>

Addresses

To represent contact information by using <address> tag.

Syntax : <address>……. </address>

Blockquotes

Blocks of content from another source within your document.

Default blockquote

The blockquote element is used to present content from another source using <p> tag.

Block quote options

Naming a source : Style and content changes for simple variations on a standard <bolckquote>.Add a <footer> for identifying the source.

Alternate Displays: Add .bolckquote-reverse class right –aligned content.

Syntax:<blockquote class="blockquote-reverse">
</blockquote>

Lists

- **Unordered :**A list of items in which the order does *not* explicitly matter.

  Syntax :

  ...

- **Ordered :**A list of items in which the order *does* explicitly matter.

  Syntax :

  ...

- **Unstyled:** Removes the Default list-style and left margin for list items. It only applies to immediate children of list items.

  Syntax:<ul class="list-unstyled">

  ...

- **Inline:** Place all list items on a single line with padding.

  Syntax :<ul class="list-inline">

  ...

- **Description:** A list of terms with their associated descriptions.

  Syntax:<dl>

  <dt>...</dt>

  <dd>...</dd>

  </dl>

  Horizontal description

  Syntax:<dl class="dl-horizontal">

  <dt>...</dt>

  <dd>...</dd>

  </dl>

Tables

Basic Table:

If you want a nice, basic table style with just some light padding and horizontal dividers, add the base class of .table to any table .

Syntax:

```
<table class="table">

    ……..

</table>
```

Tables

STRIPED TABLE :

By adding the .table-striped class, you will get zebra-striping to any rows within the <tbody>.

Syntax:

```
<table class="table table-striped">
.....
</table>
```

Tables

BORDERED TABLE :

By adding the .table-bordered class, you will get borders surrounding every element and rounded corners around the entire table.

Syntax:

```
<table class="table table-bordered">
        ……….
</table>
```

Tables

HOVER TABLE :

By adding the .table-hover class, a light gray background will be added to rows while the cursor hovers over them.

Syntax:

```
<table class="table table-hover">
        ..............
</table>
```

Tables

Condensed table

By adding the .table-condesend class, to make tables more compat by cutting cell padding in half.

Syntax:

```
<table class="table table-condensed">
        ..............
</table>
```

Contextual classes

Use contextual classes to color table rows or individual cells.

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

Tables

RESPONSIVE TABLES:

By wrapping any .table in .table-responsive class, you will make the table scroll horizontally up to small devices (under 768px). When viewing on anything larger than 768px wide, you will not see any difference in these tables.

Syntax:

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

Forms

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%.

Form Layouts:

Bootstrap provides you with following types of form layouts:

- Vertical (default) form
- Inline form
- Horizontal form

Forms

VERTICAL(OR BASIC) FORM:

- Add a role form to the parent <form> element.
- Wrap labels and controls in a with class .form-group. This is needed for optimum spacing.
- Add a class of .form-control to all textual <input>, <textarea>, and <select> elements.

Forms

INLINE FORM :

To create a form where all of the elements are inline, left aligned and labels are alongside, add the class .forminline to the <form> tag.

Note: This only applies to forms within viewports that are at least 768px wide!

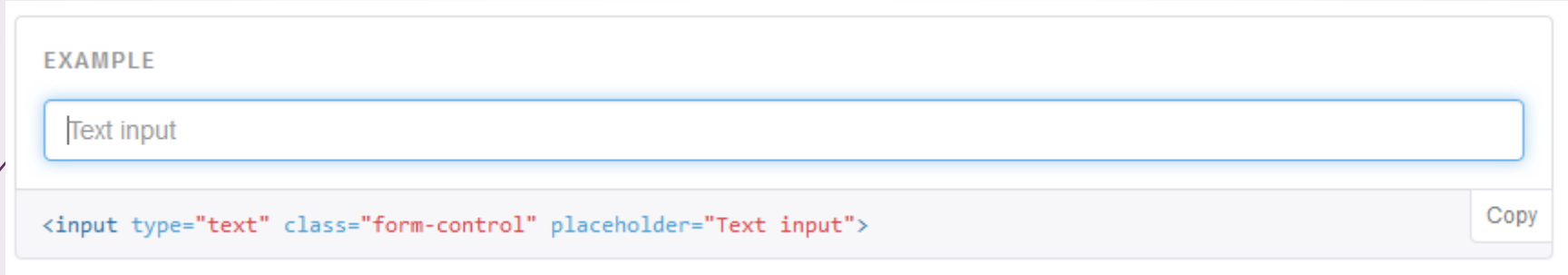
Forms

HORIZONTAL FORM:

- Add a class of .form-horizontal to the parent <form> element.
- Add a class of .control-label to all <labels> elements.

Supported controls

- **Inputs:** Bootstrap supports all the HTML5 input types: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, and color.
- Inputs will be fully styled if type is properly declared.



- **Textarea :** Form control supports multiples lines of text. Change rows attribute as required.
- **Checkboxes and Radio Buttons:** Checkboxes or Radiobuttons are used if you want the user to select any number of options from a list of preset options.

- .checkbox /.radio - Default (stacked orientation).
- .checkbox - inline /.radio - inline - Controls are appear on the same line.
- .disabled – Disable the input.

Selects:

Single select form no.of options

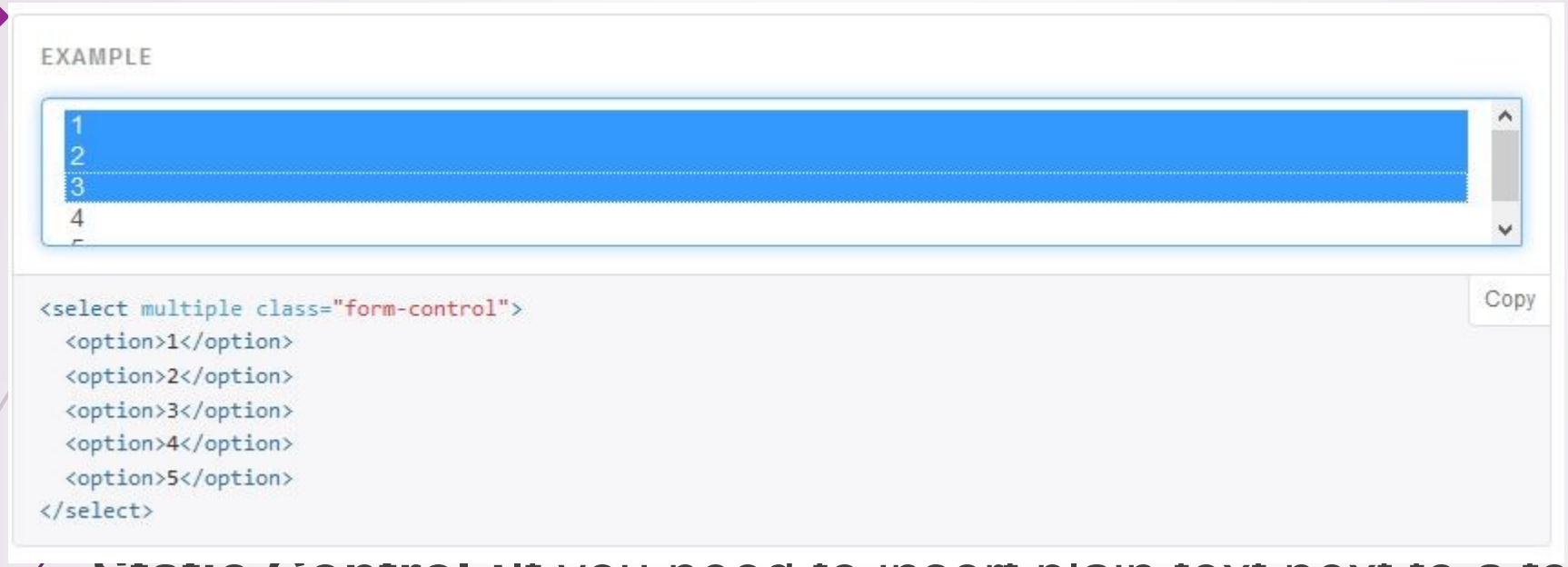
EXAMPLE

1

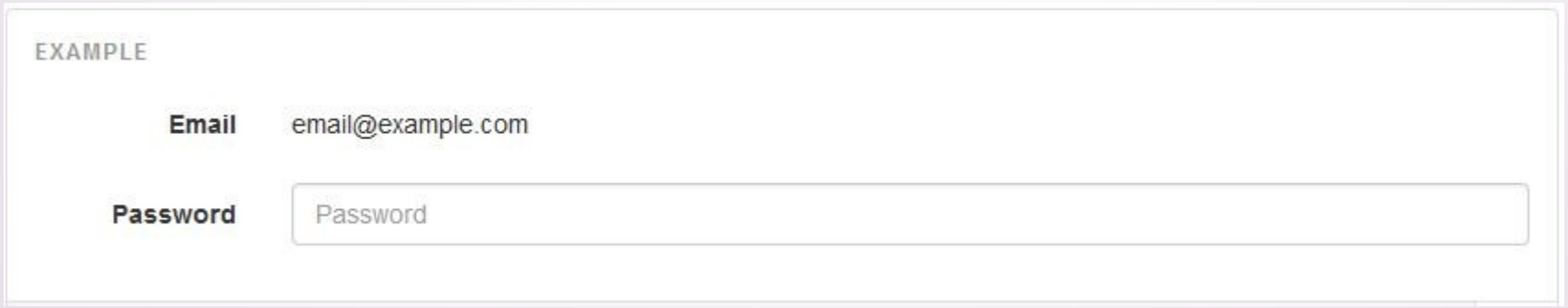
Copy

```
<select class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

- Select lists that are allow the user to pick from multiple options.



- **Static Control :**If you need to insert plain text next to a form label within a horizontal form, use the .form-control-static class on <p> element.



- **Focus state** : We remove the default outline styles on some form controls and apply a box- shadow in its place for :focus.

- **Disabled state:** Add the disabled Boolean attribute to an input. Disabled inputs appear lighter and add a not-allowed cursor.

Syntax:<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled>

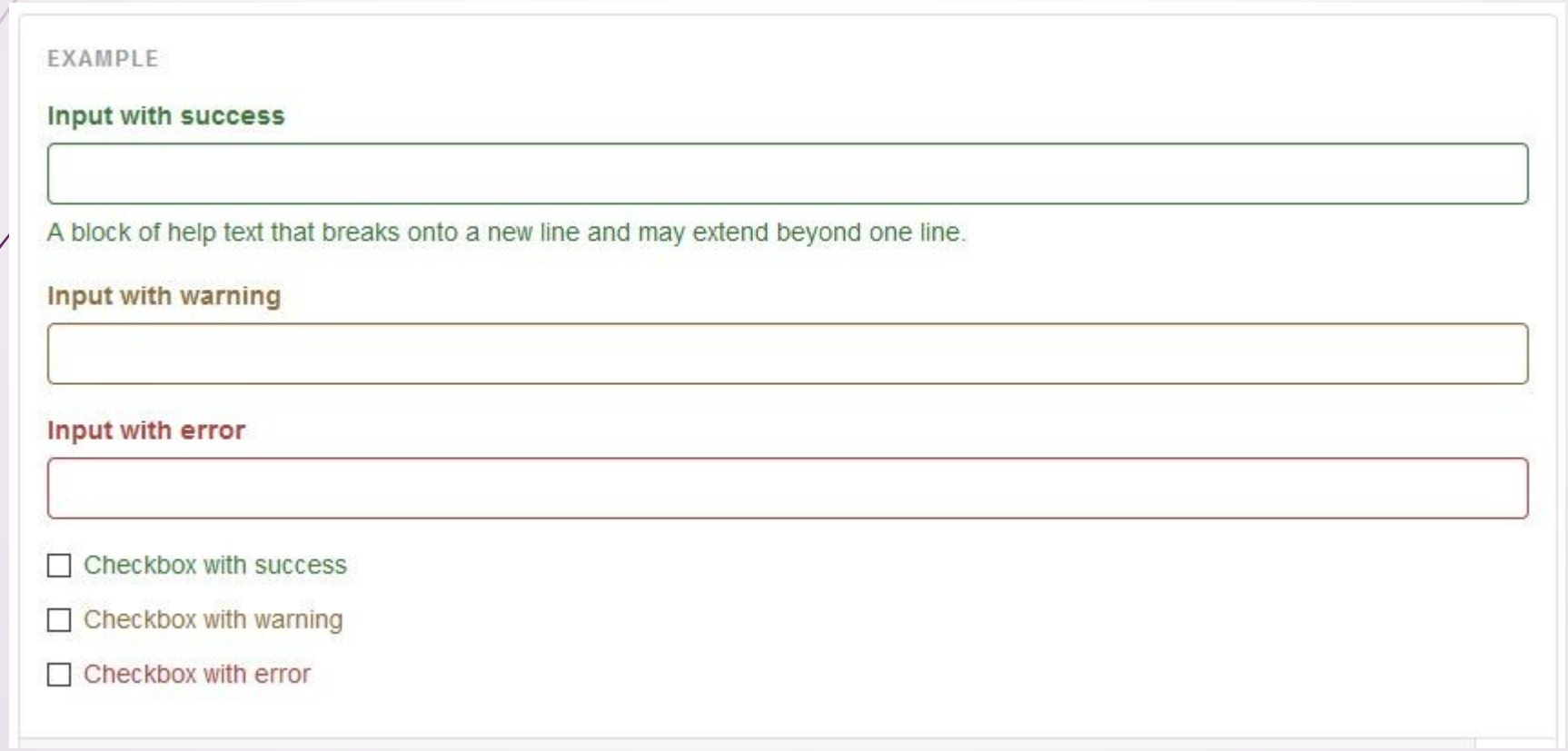
- **Readonly state:** add the readonly Boolean attribute on an input to prevent modification of the input's value. Read-only inputs appear lighter (just like disabled inputs), but retain the standard cursor.

- **Syntax:**<input class="form-control" type="text" placeholder="Readonly input here…" readonly>
- **Help Text :** A block of help text that breaks onto a new line and may extend beyond one line.
- **Syntax:**<label class="sr-only" for="inputHelpBlock">Input with help text</label>

  <input type="text" id="inputHelpBlock" class="form-control" aria-describedby="helpBlock">.

Validation states

- Bootstrap includes validation styles for error, warning, and success states on form controls. To use, add .has-warning, .has-error, or .has-success to the parent element. Any .control-label, .form-control, and .help-block within that element will receive the validation styles.



Bootstrap Form Control States

- **INPUT FOCUS** - The outline of the input is removed and a box-shadow is applied on focus
- **DISABLED INPUTS** - Add a disabled attribute to disable an input field
- **DISABLED FIELDSETS** - Add a disabled attribute to a fieldset to disable all controls within
- **READONLY INPUTS** - Add a readonly attribute to an input to prevent user input
- **VALIDATION STATES** - Bootstrap includes validation styles for error, warning, and success messages. To use, add .has-warning, .has-error, or .has-success to the parent element
- **ICONS** - You can add feedback icons with the .has-feedback class and an icon
- **HIDDEN LABELS** - Add a .sr-only class on non-visible labels

Control sizing

Set the heights of the input elements using classes likes .input-lg and .input-sm.

Set the widths of elements using grid column classes like .col-lg-* and .col-sm-*.

Height Sizing

Syntax:

```
<input class="form-control input-lg" type="text" placeholder=".input-lg">
<input class="form-control" type="text" placeholder="Default input">
<input class="form-control input-sm" type="text" placeholder=".input-sm">

<select class="form-control input-lg">...</select>
<select class="form-control">...</select>
<select class="form-control input-sm">...</select>
```

Horizontal form group sizes:

Quickly size labels and form controls within .form-horizontal by adding .form-group-lg or .form-group-sm.

Syntax:

```
<form class="form-horizontal">
  <div class="form-group form-group-lg">
    <label class="col-sm-2 control-label" for="formGroupInputLarge">Large label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputLarge" placeholder="Large input">
    </div>
  </div>
  <div class="form-group form-group-sm">
    <label class="col-sm-2 control-label" for="formGroupInputSmall">Small label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputSmall" placeholder="Small input">
    </div>
  </div>
</form>
```

Column sizing: Wrap inputs in grid columns, or any custom parent element, to easily enforce desired widths.

Syntax:

EXAMPLE


```
<div class="row">
  <div class="col-xs-2">
    <input type="text" class="form-control" placeholder=".col-xs-2">
  </div>
  <div class="col-xs-3">
    <input type="text" class="form-control" placeholder=".col-xs-3">
  </div>
  <div class="col-xs-4">
    <input type="text" class="form-control" placeholder=".col-xs-4">
  </div>
</div>
```

Buttons

Here we will discuss about how to use Bootstrap button with examples. Anything that is given a class of .btn will inherit the default look of a gray button with rounded corners.

Class	Description
Btn	Default/ Standard button.
btn-primary	Provides extra visual weight and identifies the primary action in a set of buttons.
btn-success	Indicates a successful or positive action.
btn-info	Contextual button for informational alert messages.
btn-warning	Indicates caution should be taken with this action.
btn-danger	Indicates a dangerous or potentially negative action.
btn-link	Deemphasize a button by making it look like a link while maintaining button behavior.

Buttons

Button Tags

The button classes used as <a> ,<button> or <input> element.

EXAMPLE


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

Button Options

Bootstrap provides seven styles of buttons:

EXAMPLE


```
<!-- Standard button -->
<button type="button" class="btn btn-default">Default</button>

<!-- Provides extra visual weight and identifies the primary action in a set of buttons -->
<button type="button" class="btn btn-primary">Primary</button>

<!-- Indicates a successful or positive action -->
<button type="button" class="btn btn-success">Success</button>

<!-- Contextual button for informational alert messages -->
<button type="button" class="btn btn-info">Info</button>

<!-- Indicates caution should be taken with this action -->
<button type="button" class="btn btn-warning">Warning</button>

<!-- Indicates a dangerous or potentially negative action -->
<button type="button" class="btn btn-danger">Danger</button>

<!-- Deemphasize a button by making it look like a link while maintaining button behavior -->
<button type="button" class="btn btn-link">Link</button>
```

Button Sizes

The classes that define the different sizes are:

.btn-lg

.btn-md

.btn-sm

.btn-xs

EXAMPLE


```
<p>
  <button type="button" class="btn btn-primary btn-lg">Large button</button>
  <button type="button" class="btn btn-default btn-lg">Large button</button>
</p>
<p>
  <button type="button" class="btn btn-primary">Default button</button>
  <button type="button" class="btn btn-default">Default button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-sm">Small button</button>
  <button type="button" class="btn btn-default btn-sm">Small button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-xs">Extra small button</button>
  <button type="button" class="btn btn-default btn-xs">Extra small button</button>
</p>
```

Create block level buttons—those that span the full width of a parent— by adding .btn-block.

Active state:

Buttons will appear pressed (with a darker background, darker border, and inset shadow) when active.

For <button> elements, this can be done by .active class.

Anchor element:

Add the .active class to <a> button.

EXAMPLE

Primary link Link

```
<a href="#" class="btn btn-primary btn-lg active" role="button">Primary link</a>
<a href="#" class="btn btn-default btn-lg active" role="button">Link</a>
```

Disabled state: Make buttons look unclickable by fading them back with opacity.

Add the disabled attribute to the <button> buttons.

EXAMPLE

Primary button Button

```
<button type="button" class="btn btn-lg btn-primary" disabled="disabled">Primary button</button>
<button type="button" class="btn btn-default btn-lg" disabled="disabled">Button</button>
```

Images

Bootstrap provides three classes that can be used to apply some simple styles to images:

- .img-rounded: class adds rounded corners to an image (IE8 does not support rounded corners)
- .img-circle: class shapes the image to a circle (IE8 does not support rounded corners):
- .img-thumbnail: class shapes the image to a thumbnail.

Syntax:

```
<img src="..." alt="..." class="img-rounded">
<img src="..." alt="..." class="img-circle">
<img src="..." alt="..." class="img-thumbnail">
```

OUTPUT:

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 <img> 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:

Syntax :

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

Helper classes

- Add meaning through text-colors with the classes
- Example :

Bootstrap Programming.

Bootstrap Programming.

Bootstrap Programming.

Bootstrap Programming.

Bootstrap Programming.

Bootstrap Programming.

Syntax:

```
<p class="text-muted">...</p>
<p class="text-primary">...</p>
<p class="text-success">...</p>
<p class="text-info">...</p>
<p class="text-warning">...</p>
<p class="text-danger">...</p>
```

Contextual backgrounds

Add meaning through background-colors with the classes below. Links will darken on hover just like text classes:

Class	Description
.bg-primary	Table cell is styled with class "bg-primary"
.bg-success	Table cell is styled with class "bg-success"
.bg-info	Table cell is styled with class "bg-info"
.bg-warning	Table cell is styled with class "bg-warning"
.bg-danger	Table cell is styled with class "bg-danger"

Syntax:

```
<p class="bg-primary">...</p>
<p class="bg-success">...</p>
<p class="bg-info">...</p>
<p class="bg-warning">...</p>
<p class="bg-danger">...</p>
```

Close icon :

Use the generic close icon for dismissing content like modals and alerts. Use the class close to get the close icon.

EXAMPLE

×

```
<button type="button" class="close" aria-label="Close"><span aria-hidden="true">&times;</span></button>
```

Carets:

Use carets to indicate dropdown functionality and direction. To get this functionality use the class caret with a element.

EXAMPLE

▾

```
<span class="caret"></span>
```

Quick floats

Float an element to the left or right with a class.

Syntax : ...

...

Center content blocks: Sets an element to display:block and center with margin-right:auto and margin-left:auto.

Syntax : ...

Showing and hiding content:

.show class Forces an element to be shown (display:block)

.hidden class Forces an element to be hidden (display:none)

Syntax: ...

...

Responsive utilities

These classes are used to show and/or hide content by device via media queries.

Available classes :

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

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

As of v3.2.0, the .visible-*-* classes for each breakpoint come in three variations, one for each CSS display property value listed below.

Group of classes	CSS `display`
`.visible-*-block`	`display: block;`
`.visible-*-inline`	`display: inline;`
`.visible-*-inline-block`	`display: inline-block;`

E.g. for small (sm) screens, the available .visible-*-* classes are: .visible-sm-block, .visible-sm-inline, and .visible-sm-inline-block.

The classes .visible-xs, .visible-sm, .visible-md, and .visible-lg are deprecated as of v3.2.0. They are approximately equivalent to .visible-*-block