Lists

Bootstrap supports ordered lists, unordered lists, and definition lists.

- **Ordered lists** – An ordered list is a list that falls in some sort of sequential order and is prefaced by numbers.
- **Unordered lists** – An unordered list is a list that doesn't have any particular order and is traditionally styled with bullets. If you do not want the bullets to appear, then you can remove the styling by using the class *.list-unstyled*. You can also place all list items on a single line using the class *.list-inline*.
- **Definition lists** – In this type of list, each list item can consist of both the <dt> and the <dd> elements. <dt> stands for *definition term*, and like a dictionary, this is the term (or phrase) that is being defined. Subsequently, the <dd> is the definition of the <dt>. You can make terms and descriptions in <dl> line up side-by-side using class *dl-horizontal*.

The following example demonstrates each of these types –

```
<h4>Example of Ordered List</h4>
<ol>
   <li>Item 1</li>
   <li>Item 2</li>
   <li>Item 3</li>
   <li>Item 4</li>
</ol>
<h4>Example of UnOrdered List</h4>
<ul>
   <li>Item 1</li>
   <li>Item 2</li>
   <li>Item 3</li>
   <li>Item 4</li>
</ul>
<h4>Example of Unstyled List</h4>
<ul class = "list-unstyled">
   <li>Item 1</li>
   <li>Item 2</li>
   <li>Item 3</li>

   <li>Item 4</li>

</ul>

<h4>Example of Inline List</h4>

<ul class = "list-inline">

   <li>Item 1</li>

   <li>Item 2</li>

   <li>Item 3</li>

   <li>Item 4</li>

</ul>

<h4>Example of Definition List</h4>

<dl>

   <dt>Description 1</dt>

   <dd>Item 1</dd>

   <dt>Description 2</dt>

   <dd>Item 2</dd>

</dl>

<h4>Example of Horizontal Definition List</h4>

<dl class = "dl-horizontal">

   <dt>Description 1</dt>

   <dd>Item 1</dd>

   <dt>Description 2</dt>

   <dd>Item 2</dd>

</dl>
```

Bootstrap - Forms

In this chapter, we will study how to create forms with ease using Bootstrap. Bootstrap makes it easy with the simple HTML markup and extended classes for different styles of forms. In this chapter we will study how to create forms with ease using Bootstrap.

Form Layout

Bootstrap provides you with following types of form layouts −

- Vertical (default) form
- In-line form
- Horizontal form

Vertical or Basic Form

The basic form structure comes with Bootstrap; individual form controls automatically receive some global styling. To create a basic form do the following −

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

```
<form role = "form">
   <div class = "form-group">
      <label for = "name">Name</label>
      <input type = "text" class = "form-control" id = "name" placeholder = "Enter Name">
   </div>
   <div class = "form-group">
      <label for = "inputfile">File input</label>
      <input type = "file" id = "inputfile">
      <p class = "help-block">Example block-level help text here.</p>
   </div>
   <div class = "checkbox">
      <label><input type = "checkbox"> Check me out</label>
   </div>
   <button type = "submit" class = "btn btn-default">Submit</button>
</form>
```

Inline Form

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

```
<form class = "form-inline" role = "form">
```

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

      <label class = "sr-only" for = "name">Name</label>

      <input type = "text" class = "form-control" id = "name" placeholder = "Enter Name">

   </div>

   <div class = "form-group">

      <label class = "sr-only" for = "inputfile">File input</label>

      <input type = "file" id = "inputfile">

   </div>

   <div class = "checkbox">

      <label><input type = "checkbox"> Check me out</label>

   </div>

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

</form>
```

- By default inputs, selects, and textareas have 100% width in Bootstrap. You need to set a width on the form controls when using inline form.
- Using the class *.sr-only* you can hide the labels of the inline forms.

Horizontal Form

Horizontal forms stands apart from the others not only in the amount of markup, but also in the presentation of the form. To create a form that uses the horizontal layout, do the following −

- Add a class of *.form-horizontal* to the parent <form> element.
- Wrap labels and controls in a <div> with class *.form-group*.
- Add a class of *.control-label* to the labels.

```
<form class = "form-horizontal" role = "form">

   <div class = "form-group">

      <label for = "firstname" class = "col-sm-2 control-label">First Name</label>

      <div class = "col-sm-10">

         <input type = "text" class = "form-control" id = "firstname" placeholder = "Enter
First Name">

      </div>

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

      <label for = "lastname" class = "col-sm-2 control-label">Last Name</label>

      <div class = "col-sm-10">

         <input type = "text" class = "form-control" id = "lastname" placeholder = "Enter Last Name">

      </div>

   </div>

   <div class = "form-group">

      <div class = "col-sm-offset-2 col-sm-10">

         <div class = "checkbox">

            <label><input type = "checkbox"> Remember me</label>

         </div>

      </div>

   </div>

   <div class = "form-group">

      <div class = "col-sm-offset-2 col-sm-10">

         <button type = "submit" class = "btn btn-default">Sign in</button>

      </div>

   </div>

</form>
```

Supported Form Controls

Bootstrap natively supports the most common form controls mainly *input, textarea, checkbox, radio, and select.*

Textarea

The textarea is used when you need multiple lines of input. Change *rows*attribute as necessary (fewer rows = smaller box, more rows = bigger box).

```
<form role = "form">

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

```
      <label for = "name">Text Area</label>

      <textarea class = "form-control" rows = "3"></textarea>

   </div>

</form>
```

CheckBoxes and Radio Buttons

Checkboxes and radio buttons are great when you want users to choose from a list of preset options.

- When building a form, use *checkbox* if you want the user to select any number of options from a list. Use *radio* if you want to limit the user to just one selection.
- Use *.checkbox-inline* or *.radio-inline* class to a series of checkboxes or radios for controls appear on the same line.

The following example demonstrates both (default and inline) types −

```
<label for = "name">Example of Default Checkbox and radio button </label>

<div class = "checkbox">

   <label>

      <input type = "checkbox" value = "">Option 1

   </label>

</div>

<div class = "checkbox">

   <label>

      <input type = "checkbox" value = "">Option 2

   </label>

</div>

<div class = "radio">

   <label>

      <input type = "radio" name = "optionsRadios" id = "optionsRadios1" value = "option1" 
checked> Option 1

   </label>

</div>

<div class = "radio">

   <label>
```

```
      <input type = "radio" name = "optionsRadios" id = "optionsRadios2" value = "option2">

      Option 2 - selecting it will deselect option 1

   </label>

</div>

<label for = "name">Example of Inline Checkbox and radio button </label>

<div>

   <label class = "checkbox-inline">

      <input type = "checkbox" id = "inlineCheckbox1" value = "option1"> Option 1

   </label>

   <label class = "checkbox-inline">

      <input type = "checkbox" id = "inlineCheckbox2" value = "option2"> Option 2

   </label>

   <label class = "checkbox-inline">

      <input type = "checkbox" id = "inlineCheckbox3" value = "option3"> Option 3

   </label>

   <label class = "checkbox-inline">

      <input type = "radio" name = "optionsRadiosinline" id = "optionsRadios3" value = "option1" checked> Option 1

   </label>

   <label class = "checkbox-inline">

      <input type = "radio" name = "optionsRadiosinline" id = "optionsRadios4" value = "option2"> Option 2

   </label>

</div>
```

Selects

A select is used when you want to allow the user to pick from multiple options, but by default it only allows one.

- Use <select> for list options with which the user is familiar, such as states or numbers.
- Use *multiple = "multiple"* to allow the users to select more than one option.

The following example demonstrates both (select and multiple) types –

```
<form role = "form">
   <div class = "form-group">
      <label for = "name">Select list</label>
      <select class = "form-control">
         <option>1</option>
         <option>2</option>
         <option>3</option>
         <option>4</option>
         <option>5</option>
      </select>
      <label for = "name">Mutiple Select list</label>
      <select multiple class = "form-control">
         <option>1</option>
         <option>2</option>
         <option>3</option>
         <option>4</option>
         <option>5</option>
      </select>
   </div>
</form>
```

Static Control

Use the class *.form-control-static* on a <p>, when you need to place plain text next to a form label within a horizontal form.

```
<form class = "form-horizontal" role = "form">
   <div class = "form-group">
      <label class = "col-sm-2 control-label">Email</label>
      <div class = "col-sm-10">
         <p class = "form-control-static">email@example.com</p>
      </div>
   </div>
   <div class = "form-group">

      <label for = "inputPassword" class = "col-sm-2 control-label">Password</label>

      <div class = "col-sm-10">

         <input type = "password" class = "form-control" id = "inputPassword" placeholder
= "Password">

      </div>

   </div>

</form>
```

Form Control States

In addition to the *:focus* (i.e., a user clicks into the input or tabs onto it) state, Bootstrap offers styling for disabled inputs and classes for form validation.

Input Focus

When an input receives *:focus*, the outline of the input is removed and a *box-shadow* is applied.

Disabled Inputs

If you need to disable an input, simply adding the *disabled* attribute will not only disable it; it will also change the styling and the mouse cursor when the cursor hovers over the element.

Disabled Fieldsets

Add the disabled attribute to a <fieldset> to disable all the controls within the <fieldset> at once.

Help Text

Bootstrap form controls can have a block level help text that flows with the inputs. To add a full width block of content, use the *.help-block* after the <input>. The following example demonstrates this −

```
<form role = "form">

   <span>Example of Help Text</span>

   <input class = "form-control" type = "text" placeholder = "">

   <span class = "help-block">

      A longer block of help text that breaks onto a new line and may extend beyond one
line.

   </span>
```

```
</form>
```

Bootstrap - Buttons

This chapter covers the use age of Bootstrap button with examples. Anything that is given a class of **.btn** will inherit the default look of a gray button with rounded corners. However, Bootstrap provides some options to style buttons, which are summarized in the following table –

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.

The following example demonstrates all the above button classes –

```
<!-- Standard button -->

<button type = "button" class = "btn btn-default">Default Button</button>

<!-- Provides extra visual weight and identifies the primary action in a set of buttons --
>

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

<!-- Indicates a successful or positive action -->

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

<!-- Contextual button for informational alert messages -->

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

<!-- Indicates caution should be taken with this action -->

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

<!-- Indicates a dangerous or potentially negative action -->

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

Button Size

The following table summarizes the classes used to get buttons of various sizes—

Class	Description
.btn-lg	This makes the button size large.
.btn-sm	This makes the button size small.
.btn-xs	This makes the button size extra small.
.btn-block	This creates block level buttons—those that span the full width of a parent.

The following example demonstrates this –

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

Button Tags

You may use button classes with <a>, <button>, or <input> element. But it is recommended that you use it with <button> elements mostly to avoid cross browser inconsistency issues.

The following example demonstrates this –

```
<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">
```