

# Bootstrap

# 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 others, as well as optional Javascript plugins
- Bootstrap also gives you the ability to easily create responsive designs.  
(Looks good on all devices, from small phones to large desktops.)

# Advantages of Bootstrap

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

# Where to Get Bootstrap 4?

- There are two ways to start using Bootstrap 4 on your own website
  - Include Bootstrap 4 from a CDN (recommended)
  - Download Bootstrap 4 from [getbootstrap.com](https://getbootstrap.com).

# Create First Web Page with Bootstrap 4

## 1. Add the HTML5 boilerplate

- a. Bootstrap 4 uses HTML elements and CSS properties that require the HTML5 doctype.
- b. Always include the HTML5 doctype at the beginning of the page.

## 2. Bootstrap 4 is mobile-first

- a. Bootstrap 4 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.
- b. 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.

The `initial-scale=1` part sets the initial zoom level when the page is first loaded by the browser.

# Create First Web Page with Bootstrap 4 cont.

## 3. Containers

- a. Bootstrap 4 also requires a containing element to wrap site contents.
- b. There are two container classes to choose from:
  - i. The `.container` class provides a responsive fixed width container
  - ii. The `.container-fluid` class provides a full width container, spanning the entire width of the viewport.

# Container Examples

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

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

A visual representation of the .container class. It shows a light gray rectangular box centered within a larger white rectangular area, illustrating how the container class restricts content to a fixed width.

.container

A visual representation of the .container-fluid class. It shows a light gray rectangular box that spans the entire width of the white rectangular area, illustrating how the container-fluid class allows content to stretch across the full width of the page.

.container-fluid

# Bootstrap 4 Grid System

- Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
- If you do not want to use all 12 columns individually, you can group the columns together to create wider columns.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

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



# Grid Classes

- The bootstrap 4 grid system has five classes:
  - `.col-` (extra small devices - screen width less than 576px)
  - `.col-sm-` (small devices - screen width equal to or greater than 576px)
  - `.col-md-` (medium devices - screen width equal to or greater than 768px)
  - `.col-lg-` (large devices - screen width equal to or greater than 992px)
  - `.col-xl-` (xlarge devices - screen width equal to or greater than 1200px)
- The classes above can be combined to create more dynamic and flexible layouts.
- Each class scales up, so if you wish to set the same widths for sm and md, you only need to specify sm.

# Basic Structure of a Bootstrap 4 Grid

```
<!-- Control the column width, and how they should appear on different devices -->  
<div class="row">  
  <div class="col-*-*"></div>  
  <div class="col-*-*"></div>  
</div>  
<div class="row">  
  <div class="col-*-*"></div>  
  <div class="col-*-*"></div>  
  <div class="col-*-*"></div>  
</div>
```

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

# Basic Structure of a Bootstrap 4 Grid cont.

```
<!-- Or let Bootstrap automatically handle the layout -->  
<div class="row">  
  <div class="col"></div>  
  <div class="col"></div>  
  <div class="col"></div>  
</div>
```

- Instead of adding a number to each `col`, let bootstrap handle the layout, to create equal width columns: two “col” elements = 50% width to each col. Three cols = 33.33% width to each col. Four cols = 25% width, etc. You can also use `.col-sm | md | lg | xl` to make the columns responsive.

# Three Equal Columns

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

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

# Responsive Columns

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

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

# Two Unequal Responsive Columns

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

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

# Assignment 3.1 - Recreate Bootstrap Grid

- Recreate [this](#) grid using Bootstrap and CSS.

# Bootstrap 4 Colors - Text Colors

- Bootstrap 4 has some contextual classes that can be used to provide “meaning through colors”.
- The classes for text colors are:
  - `.text-muted`
  - `.text-primary`
  - `.text-success`
  - `.text-info`
  - `.text-warning`
  - `.text-danger`
  - `.text-secondary`
  - `.text-white`
  - `.text-dark`
  - `.text-body`
  - `.text-light`



# Bootstrap 4 Colors - Background Colors

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

# Bootstrap 4 Images - Rounded Corners

- The `.rounded` class adds rounded corners to an image

```

```

# Bootstrap 4 Images - Circle

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

```

```

# Bootstrap 4 Images - Thumbnail

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

```

```

# Aligning Images

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

```
  

```

# Centered Image

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

```

```

# Responsive Images

- Images come in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.
- Create responsive images by adding an `.img-fluid` class to the `<img>` tag. The image will then scale nicely to the parent element.
- The `.img-fluid` class applies `max-width: 100%;` and `height: auto;` to the image.

```

```

# Bootstrap 4 Tables - Bootstrap 4 Basic Table

- A basic Bootstrap 4 table has a light padding and horizontal dividers.
- The `.table` class adds basic styling to a table.

```
<table class="table">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  </tr>
</table>
```



# Striped Rows

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

```
<table class="table table-striped">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  </tr>
</table>
```

# Bordered Table

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

```
<table class="table table-bordered">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  </tr>
</table>
```

# Hover Rows

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

```
<table class="table table-hover">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  </tr>
</table>
```



# Bootstrap 4 Buttons - Button Styles

- Bootstrap 4 provides different styles of buttons
  - .btn
  - .btn btn-primary
  - .btn btn-secondary
  - .btn btn-success
  - .btn btn-info
  - .btn btn-warning
  - .btn btn-danger
  - .btn btn-dark
  - .btn btn-light
  - .btn btn-link
- The button classes can be used on `<a>`, `<button>`, or `<input>` elements.

# Button Sizes

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

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

# Block Level Buttons

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

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

# Bootstrap Forms

- Bootstrap 4's Default Settings
  - Form controls automatically receive some global styling with Bootstrap:
  - All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.



# Bootstrap 4 Stacked Form

- The following example creates a stacked form with two input fields, one checkbox, and a submit button.
- Add a wrapper element with `.form-group`, around each form control, to ensure proper margins.

```
<form action="/action_page.php">
  <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="form-group form-check">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

# Bootstrap 4 Inline Form

- In an inline form, all of the elements are inline and left-aligned.
  - This only applies to forms within viewports that are at least 576px wide. On screens smaller than 576px, it will stack horizontally.
- Add class `.form-inline` to the `<form>` element.
- The following example creates an inline form with two input fields, one checkbox, and one submit button.

```
<form class="form-inline" action="/action_page.php">
  <label for="email">Email address:</label>
  <input type="email" class="form-control" id="email">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd">
  <div class="form-check">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

# Bootstrap 4 Form Inputs

- Bootstrap supports the following form controls
  - Input
  - Textarea
  - Checkbox
  - Radio
  - select

# Bootstrap Input

- 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 NOT be fully styled if their type is not properly declared.
- The following example contains two input elements; one of `type="text"` and one of `type="password"`. As mentioned before, use the `.form-control` class to style inputs with full-width and proper padding, etc.

```
<div class="form-group">
  <label for="usr">Name:</label>
  <input type="text" class="form-control" id="usr">
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd">
</div>
```

# Bootstrap Checkboxes

- Use a wrapper element with `class="form-check"` to ensure proper margins for labels and checkboxes.
- Add the `.form-check-label` class to label elements, and `.form-check-input` to style checkboxes properly inside the `.form-check` container.

```
<div class="form-check">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 1
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 2
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="" disabled>Option 3
  </label>
</div>
```

# Inline Checkboxes

- Use the `.form-check-inline` class if you want the checkboxes to appear on the same line.

```
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 1
  </label>
</div>
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="">Option 2
  </label>
</div>
<div class="form-check-inline">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" value="" disabled>Option 3
  </label>
</div>
```

# Bootstrap Radio Buttons

- Use the same classes as checkboxes to style these.

# Bootstrap Select List

- Use the `.form-control` class inside the `<select>` tag.

```
<div class="form-group">
  <label for="sel1">Select list:</label>
  <select class="form-control" id="sel1">
    <option>1</option>
    <option>2</option>
    <option>3</option>
    <option>4</option>
  </select>
</div>
```



# Bootstrap 4 Navigation Bar

- A navigation bar is a navigation header that is placed at the top of the page.

# Basic Navbar

- With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.
- A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xl | lg | md | sm` (stacks the navbar vertically on extra large, large, medium or small screens).
- To add links inside the navbar, use a `<ul>` element with `class="navbar-nav"`. Then add `<li>` elements with a `.nav-item` class followed by an `<a>` element with a `.nav-link` class.

# Basic Navbar Example

```
<!-- A grey horizontal navbar that becomes vertical on small screens -->
<nav class="navbar navbar-expand-sm bg-light">

  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 3</a>
    </li>
  </ul>

</nav>
```

# Vertical Navbar

- Remove the `.navbar-expand-xl | lg | md | sm` class to create a vertical navigation bar.

```
<!-- A grey horizontal navbar that becomes vertical on small screens -->
<nav class="navbar navbar-expand-sm bg-light">

  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 3</a>
    </li>
  </ul>

</nav>
```

# Centered Navbar

- Add the `.justify-content-center` class to center the navigation bar.
- The following example will center the navigation bar on medium, large, and extra large screens. On small screens it will be displayed vertically and left-aligned (because of the `.navbar-expand-sm` class).

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">  
  ...  
</nav>
```

# Colored Navbar

- Use any of the `.bg-color` classes to change the background color of the navbar (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark`, and `.bg-light`)
- Add a white text color to all links in the navbar with the `.navbar-dark class`, or use the `.navbar-light` class to add a black text color.

# Colored Navbar Example

```
<!-- Grey with black text -->
<nav class="navbar navbar-expand-sm bg-light navbar-light">
  <ul class="navbar-nav">
    <li class="nav-item active">
      <a class="nav-link" href="#">Active</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Disabled</a>
    </li>
  </ul>
</nav>

<!-- Black with white text -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">...</nav>

<!-- Blue with white text -->
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">...</nav>
```

# Brand / Logo

- The `.navbar-brand` class is used to highlight the brand/logo/project name of your page.
- When using the `.navbar-brand` class on images, Bootstrap 4 will automatically style the image to fit the navbar vertically.

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Logo</a>
  ...
</nav>

<!-- Example with Image -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <a class="navbar-brand" href="#">
    
  </a>
  ...
</nav>
```



# Collapsing The Navigation Bar

- Very often, especially on small screens, you want to hide navigation links and replace them with a button that should reveal them when clicked on.
- To create a collapsible navigation bar, use a button with `class="navbar-toggler"`, `data-toggle="collapse"`, and `data-target="#thetarget"`. Then wrap the navbar contents (links, etc) inside a `div` element with `class="collapse navbar-collapse"`, followed by an `id` that matches the `data-target` of the button: "thetarget"

# Collapsing the Navigation Bar Example

```
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Navbar</a>

  <!-- Toggler/collapsible Button -->
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>

  <!-- Navbar links -->
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
    </ul>
  </div>
</nav>
```

# Navbar with Dropdown

- Navbars can also hold dropdown menus.

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Logo</a>

  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>

    <!-- Dropdown -->
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
        Dropdown link
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Link 1</a>
        <a class="dropdown-item" href="#">Link 2</a>
        <a class="dropdown-item" href="#">Link 3</a>
      </div>
    </li>
  </ul>
</nav>
```

# Bootstrap 4 Flex

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.
- To create a flexbox container and to transform direct children into flex items, use the `d-flex` class.

```
<div class="d-flex p-3 bg-secondary text-white">  
  <div class="p-2 bg-info">Flex item 1</div>  
  <div class="p-2 bg-warning">Flex item 2</div>  
  <div class="p-2 bg-primary">Flex item 3</div>  
</div>
```



# Horizontal Direction

- Use `.flex-row` to display the flex items horizontally (side by side). This is default.
- Use `.flex-row-reverse` to right align the horizontal direction.

```
<div class="d-flex flex-row bg-secondary">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>

<div class="d-flex flex-row-reverse bg-secondary">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



# Justify Content

- Use the `.justify-content-*` classes to change the alignment of flex items.
- Valid classes are `start` (default), `end`, `center`, `between`, or `around`.

```
<div class="d-flex justify-content-start">...</div>
<div class="d-flex justify-content-end">...</div>
<div class="d-flex justify-content-center">...</div>
<div class="d-flex justify-content-between">...</div>
<div class="d-flex justify-content-around">...</div>
```



# Align Content

- Control the vertical alignment of gathered flex items with the `.align-content-*` classes. Valid classes are `start` (default), `end`, `center`, `between`, `around`, and `stretch`.
- These classes have no effect on single rows of flex items.

```
<div class="d-flex flex-wrap align-content-start">..</div>
```

```
<div class="d-flex flex-wrap align-content-end">..</div>
```

```
<div class="d-flex flex-wrap align-content-center">..</div>
```

```
<div class="d-flex flex-wrap align-content-around">..</div>
```

```
<div class="d-flex flex-wrap align-content-stretch">..</div>
```

# Align Items

- Control the vertical alignment of single rows of flex items with the `.align-items-*` classes. Valid classes are `start`, `end`, `center`, `baseline`, and `stretch` (default).

```
<div class="d-flex flex-wrap align-content-start">..</div>
```

```
<div class="d-flex flex-wrap align-content-end">..</div>
```

```
<div class="d-flex flex-wrap align-content-center">..</div>
```

```
<div class="d-flex flex-wrap align-content-around">..</div>
```

```
<div class="d-flex flex-wrap align-content-stretch">..</div>
```



# Assignment 3.2 - Recreate Apple Page

- Recreate [this](#) Apple page using HTML, Bootstrap, and CSS.
  - Only worry about the design. Not the functionality!