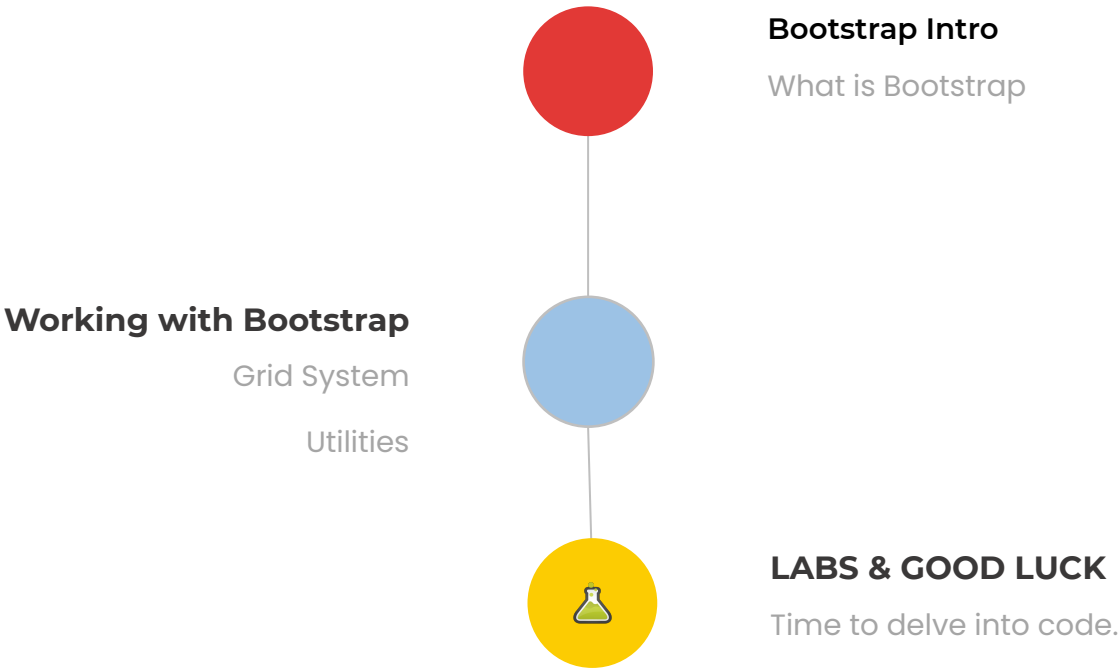




INTERACTIVE DEVELOPMENT

Utilizing Bootstrap



Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

[Get started](#)[Download](#)

Currently v4.5.3

DEV ☒ 3.5.8
TEST ☒ 3.5.5
PROD ☒ 3.5.5



Octopus Deploy

Break down dev & ops silos by automating deployments & IT ops runbooks from a single place. Get started free.

ads via Carbon

STORY: SPEED

Bootstrap has come a long way and it was developed out of “boredom” by developers who wanted a faster way to create User Interfaces and to resolve all the nooks and crannies imposed by all the various browsers and devices.

With today’s variety of devices, we have an issue in keeping up to technology advances.

Bootstrap is one such framework that allows us to simplify our development process.

A set of user interface libraries and utilities are placed inside Bootstrap.

Bootstrap uses

- 1. **Flexbox grid** for it’s layout
- 2. rem for measurements
- 3. Has device breakpoints to accommodate for different media sizes

col-xl-2 // class used for extra large screens like your gaming rigs
col-lg-2 // class used for large devices like laptops
col-md-2 // class used for medium devices like tablets
col-sm-2 // class used for small devices like mobile phones

Grid options

While Bootstrap uses **ems** or **rems** for defining most sizes, **pxs** are used for grid breakpoints and container widths. This is because the viewport width is in pixels and does not change with the **font size**.

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

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

The Bootstrap Grid system helps you to create your layout and easily build a responsive website.

The grid is divided into **12 columns**, so your layout will be based on this. The most basic column is calling the “col” class then we can decide which device breakpoint do we want.

Example if say col-sm-4 meaning this column size will be applied to small devices and take up 4 columns.

If no size is specified, Bootstrap will auto-fill in the column size

Grid options

While Bootstrap uses **ems** or **rems** for defining most sizes, **pxs** are used for grid breakpoints and container widths. This is because the viewport width is in pixels and does not change with the **font size**.

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

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

To use the grid system you'll have to add a **.row** class to the main **div**.

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
```

In this instance, bootstrap will auto calculate and assign 4 columns to each col-sm giving a total of 12 columns

CONTAINER & COLUMNS

The above example creates three equal-width columns on small, medium, large, and extra large devices using our predefined grid classes. Those columns are centered in the page with the parent `.container`.

Breaking it down, here's how it works:

- Containers provide a means to center and horizontally pad your site's contents. Use `.container` for a responsive pixel width or `.container-fluid` for `width: 100%` across all viewport and device sizes.
- Rows are wrappers for columns. Each column has horizontal `padding` (called a gutter) for controlling the space between them. This `padding` is then counteracted on the rows with negative margins. This way, all the content in your columns is visually aligned down the left side.
- In a grid layout, content must be placed within columns and only columns may be immediate children of rows.
- Thanks to flexbox, grid columns without a specified `width` will automatically layout as equal width columns. For example, four instances of `.col-sm` will each automatically be 25% wide from the small breakpoint and up. See the [auto-layout columns](#) section for more examples.
- Column classes indicate the number of columns you'd like to use out of the possible 12 per row. So, if you want three equal-width columns across, you can use `.col-4`.
- Column `widths` are set in percentages, so they're always fluid and sized relative to their parent element.
- Columns have horizontal `padding` to create the gutters between individual columns, however, you can remove the `margin` from rows and `padding` from columns with `.no-gutters` on the `.row`.
- To make the grid responsive, there are five grid breakpoints, one for each [responsive breakpoint](#): all breakpoints (extra small), small, medium, large, and extra large.
- Grid breakpoints are based on minimum width media queries, meaning **they apply to that one breakpoint and all those above it** (e.g., `.col-sm-4` applies to small, medium, large, and extra large devices, but not the first `xs` breakpoint).
- You can use predefined grid classes (like `.col-4`) or [Sass mixins](#) for more semantic markup.

Adapted guide from Bootstrap documentation.

The Bootstrap documentation provides an excellent resource of examples and explanations. Learn to read the docs

We can use additional flexbox methods but these is simplified using Bootstrap classes

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

STARTING OUT

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

1. Create a base HTML file
2. Add in the Bootstrap CSS and JS (here we add the minified codes). You can copy the ones used on the bootstrap intro page OR simply download bootstrap and include the files in
3. The CDN used allows bootstrap to be easily accessible across servers.
4. Start creating your interface using Bootstrap

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/boot
```

[Copy](#)

Separate

If you decide to go with the separate scripts solution, Popper.js must come first, and then our JavaScript plugins.

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2l  
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integ  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" inte
```

[Copy](#)

<https://repl.it/@malcolmyam/wk08-simple-bootstrap#index.html>

Headings

All HTML headings, `<h1>` through `<h6>`, are available.

Heading	Example
<code><h1></h1></code>	h1. Bootstrap heading
<code><h2></h2></code>	h2. Bootstrap heading
<code><h3></h3></code>	h3. Bootstrap heading
<code><h4></h4></code>	h4. Bootstrap heading
<code><h5></h5></code>	h5. Bootstrap heading
<code><h6></h6></code>	h6. Bootstrap heading

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

[Copy](#)

`.h1` through `.h6` classes are also available, for when you want to match the font styling of a heading but cannot use the associated HTML element.

Bootstrap applies a reset css within itself. This is similar to what we applied for Assignment 1 using the normalize.css .

It allows a good reboot of the preset properties of the HTML elements

<https://getbootstrap.com/docs/4.5/content/typography/>

Responsive images

Images in Bootstrap are made responsive with `.img-fluid`. `max-width: 100%`; and `height: auto`; are applied to the image so that it scales with the parent element.



```

```

Image thumbnails

In addition to our [border-radius utilities](#), you can use `.img-thumbnail` to give an image a rounded 1px border appearance.



```

```

Copy

<https://getbootstrap.com/docs/4.5/content/images/>

COMPONENTS

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

Navbar

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

Components are what really accelerates the development process. This are pre-built utilities that allows us to create common user interfaces just by calling the appropriate CSS class and HTML elements

Used in combination with Bootstrap Grid layout system, this is how we can create responsive interfaces

Examples

Alerts are available for any length of text, as well as an optional dismiss button. For proper styling, use one of the eight **required** contextual classes (e.g., `.alert-success`). For inline dismissal, use the [alerts jQuery plugin](#).

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

A simple dark alert—check it out!

```
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
<div class="alert alert-danger" role="alert">
  A simple danger alert—check it out!
</div>
<div class="alert alert-warning" role="alert">
  A simple warning alert—check it out!
</div>
<div class="alert alert-info" role="alert">
  A simple info alert—check it out!
</div>
<div class="alert alert-light" role="alert">
  A simple light alert—check it out!
</div>
<div class="alert alert-dark" role="alert">
  A simple dark alert—check it out!
</div>
```

<https://getbootstrap.com/docs/4.5/components/alerts/>

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

Navbar

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

Forms are quite a hassle to design and customise. We use Bootstrap to create responsive form designs together with quick utilities for responsiveness and common form customisation tools

Overview

Bootstrap's form controls expand on [our Rebooted form styles](#) with classes. Use these classes to opt into their customized displays for a more consistent rendering across browsers and devices.

Be sure to use an appropriate `type` attribute on all inputs (e.g., `email` for email address or `number` for numerical information) to take advantage of newer input controls like email verification, number selection, and more.

Here's a quick example to demonstrate Bootstrap's form styles. Keep reading for documentation on required classes, form layout, and more.

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp">
    <small id="emailHelp" class="form-text
text-muted">We'll never share your email with anyone
else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input"
id="exampleCheck1">
    <label class="form-check-label"
for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn
btn-primary">Submit</button>
</form>
```

COMPONENTS :: NAVBAR

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

Navbar

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

Creating responsive navigation that allows resizing and mobile designs can be tricky. Bootstrap simplifies this process for us.

How it works

Here's what you need to know before getting started with the navbar:

- Navbars require a wrapping `.navbar` with `.navbar-expand-{sm|md|lg|xl}` for responsive collapsing and `color scheme` classes.
- Navbars and their contents are fluid by default. Use `optional containers` to limit their horizontal width.
- Use our `spacing` and `flex` utility classes for controlling spacing and alignment within navbars.
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.
- Navbars are hidden by default when printing. Force them to be printed by adding `.d-print` to the `.navbar`. See the `display` utility class.
- Ensure accessibility by using a `<nav>` element or, if using a more generic element such as a `<div>`, add a `role="navigation"` to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

Supported content

Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:

- `.navbar-brand` for your company, product, or project name.
- `.navbar-nav` for a full-height and lightweight navigation (including support for dropdowns).
- `.navbar-toggler` for use with our collapse plugin and other `navigation toggling` behaviors.
- `.form-inline` for any form controls and actions.
- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.

Here's an example of all the sub-components included in a responsive light-themed navbar that automatically collapses at the `lg` (large) breakpoint.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
    data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span
class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
          Dropdown
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#" tabindex="-1"
aria-disabled="true">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search"
aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
    </form>
  </div>
</nav>
```

<https://getbootstrap.com/docs/4.5/components/navbar/>

UTILITIES :: VIDEO EMBEDS

Example

Wrap any embed like an `<iframe>` in a parent element with `.embed-responsive` and an aspect ratio. The `.embed-responsive-item` isn't strictly required, but we encourage it.



```
<div class="embed-responsive
embed-responsive-16by9">
  <iframe
class="embed-responsive-item"
src="https://www.youtube.com/embed/zp
0ULjyy-n8?rel=0"
allowfullscreen></iframe>
</div>
```

ACTIVITY

"Simple About" (30 mins - 1hr)

#TODO ACTIVITY 6

Using your previous CA on a simple about page, create a simple **ABOUT ME** page using bootstrap css classes and utilities. The site should be responsive.

#TOHAVE

Bootstrap Components, utilities and your own content:

- Include a Navbar
- About me text content
- A 4 column layout for your portfolio to show 4 different responsive images.

Consider using Card layouts or using grid

<https://getbootstrap.com/docs/4.5/components/card/>

- Include a simple Contact

Additional Challenge: Style your site using added CSS and typography of your choice

You may use Bootstrap examples to start off

<https://getbootstrap.com/docs/4.5/examples/>

If you want to modify or add on your own CSS, place the CSS script after Bootstrap css

SUBMISSION

Create folder **wk08-simple-click** and place your codes for this exercise

Compile all the activities for the week and submit as a ZIP file

wk08-studentid-studentname.zip file

Name each activity properly

Eg.

/wk08-studentid-studentname

../wk08-template-10

../wk08-simple-click

../wk08-circles

../wk08-randomuser

../wk08-weather

../wk08-about-me

Bootstrap

Summary

What have we covered?

Starting with Bootstrap

Bootstrap Utilities

Bootstrap Components

Key Takeaway?

Practise & Practise

Bootstrap is just one such library for your CSS framework. There are many CSS frameworks out there that are being used. Always **reference the documentation**

Learn the concepts of how bootstrap is being applied instead of memorizing everything in bootstrap.

As developers, we do not memorize but learn how to use things to create “new stuff”