



# **Week 3 Workshop**

Web Development  
Fundamentals



# Check-In

- How was this week for you? Any particular challenges or accomplishments?
- Did you understand the Exercises and were you able to complete them?
- How were the Challenges and Quiz this week?



# Agenda

Activity	Estimated Duration
Welcome & check-in	10 mins
Week 3 Review	1 hour 30 mins
Break	10 mins
Workshop Introduction & Demo	10 mins
Workshop Assignment	1 hour 30 mins
Code review & wrap-up	30 mins



# Week 3: New Concepts

- What is Bootstrap?
- Responsive design
- Breakpoints
- Bootstrap grid
- Bootstrap utility classes

- Bootstrap static components:
  - Navbars
  - Breadcrumbs
  - Icons
  - Buttons
  - Forms
  - Tables
  - Cards
  - Images



# Review: Responsive Web Design

## Discuss together:

- What does **mobile-first design** mean?
- What does **responsive web design** mean?
- What is a **UI framework**?
- How does **Bootstrap** work under the hood?



# Review: Breakpoints

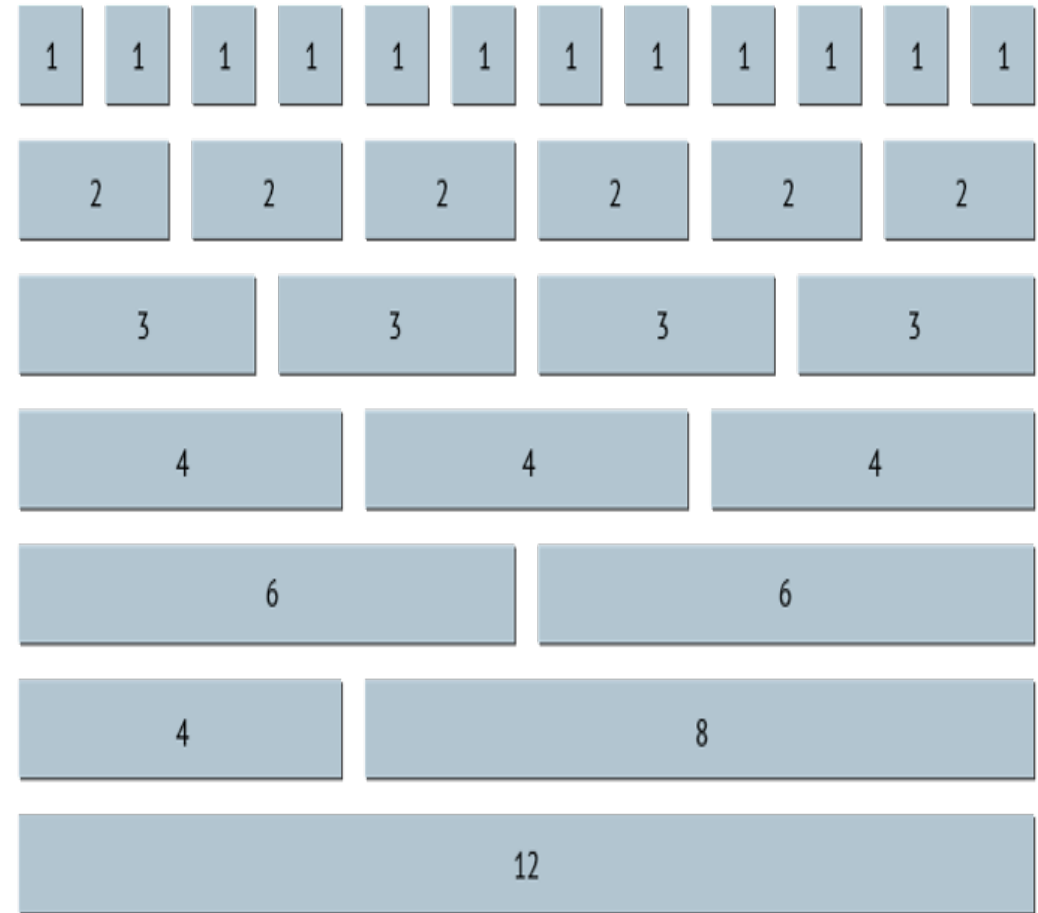
## Discuss:

- What is a **media query**, and what is a **breakpoint**?
- Remember, you will not have to write any media queries/define breakpoints yourself to use Bootstrap. You simply access the breakpoints by using responsive classes that let you specify **sm**, **md**, **lg**, **xl**, and **xxl**.
- However, it's good to have some understanding of how the code works under the hood.
- Because Bootstrap is designed mobile-first, the **XS** breakpoint is the default and does not need to be specified.



# Review: The Bootstrap Grid

- Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content.
- It's built with flexbox and media queries/breakpoints to be fully responsive.
- Rows are wrappers for columns.
- Content **must** be placed within columns, and only columns may be immediate children of rows.
- There are 12 maximum columns to each row.





# Review: The Bootstrap Grid System

## Discuss:

- What does `col-md-6` do?
- What does `col-12` do?
- What will this row look like in a Bootstrap grid, and why?

```
<div class="row">  
  <div class="col">  
    1 of 3  
  </div>  
  <div class="col">  
    2 of 3  
  </div>  
  <div class="col">  
    3 of 3  
  </div>  
</div>
```





# Review: Bootstrap Documentation

- **Bonus points\*** for the first person to look through the Bootstrap online documentation (<https://getbootstrap.com/docs/5.2>) and say out loud the name of the class used to create a **Button component with a yellow outline**.
- The importance of getting used to reading code documentation cannot be overstated. **It's very difficult to memorize everything**, there's too much information, and it's constantly evolving!
- Instead, ***practice finding the information when you need it.***
- Luckily, Bootstrap's documentation is quite thorough and well maintained.
- Be forewarned. Not all documentation will be this friendly! You will often have to 'connect the dots' to make up for poorly written or missing documentation.
- So when there is good documentation on a project, appreciate it and use it to the full extent.

*\*no actual bonus points will be awarded; good job though!*



# Review: Bootstrap Order-\* Classes

- Bootstrap's responsive **order-\*** classes can be used with the Bootstrap grid to change the order of columns.
- You can use them with breakpoints to specify that they should only apply within certain viewport sizes, or leave them out to apply to all.
- You can set the order of the column by number from **0-5**. For example, to make a column appear first in a row, you can give it the class **order-0**.
- You can also use **order-first** and **order-last**



# Review: Bootstrap Offset-\* Classes

- Use the responsive **offset-\*** classes to force a column to move to the right in a row
- You can optionally add a breakpoint, such as **offset-sm-4**
- If you leave out the breakpoint, it will apply to all viewport sizes



# Review: Adding Custom Styles

- If adding custom styles, create an external stylesheet and link it **below** the link to the **bootstrap.css** file.
- You can write rules for existing Bootstrap classes and it will override the Bootstrap rules where there is any conflict, and keep the Bootstrap rules where there is no conflict -- as long as your custom stylesheet is linked **below** the Bootstrap css link.



# Review: Bootstrap Alignment Classes

- Bootstrap has many different classes to help with alignment.
- In this week, you learned about `align-items-center` and `align-self-center` to help with vertical alignment.
- You also learned about `text-center` to help with horizontal text alignment.
- **Discuss:** How would you make `text-center` into a responsive class that will only apply to **LG** viewports and up?



# Review: Color & Spacing Utility Classes

## Discuss together:

- What does ms-3 do?

Adds a left margin to an element of size 3 (1rem / 16px default)

- What does p-5 do?

Adds a padding all around an element of size 5 (3rem / 48px default)

- What does py-1 do?

Adds a Top and Bottom padding to an element of size 1 (.25rem / 4px default)

- What does mx-auto do?

Horizontally center an element within its parent element

**Note:** Element must be fixed-width and block level (**display: block** and **width** set)

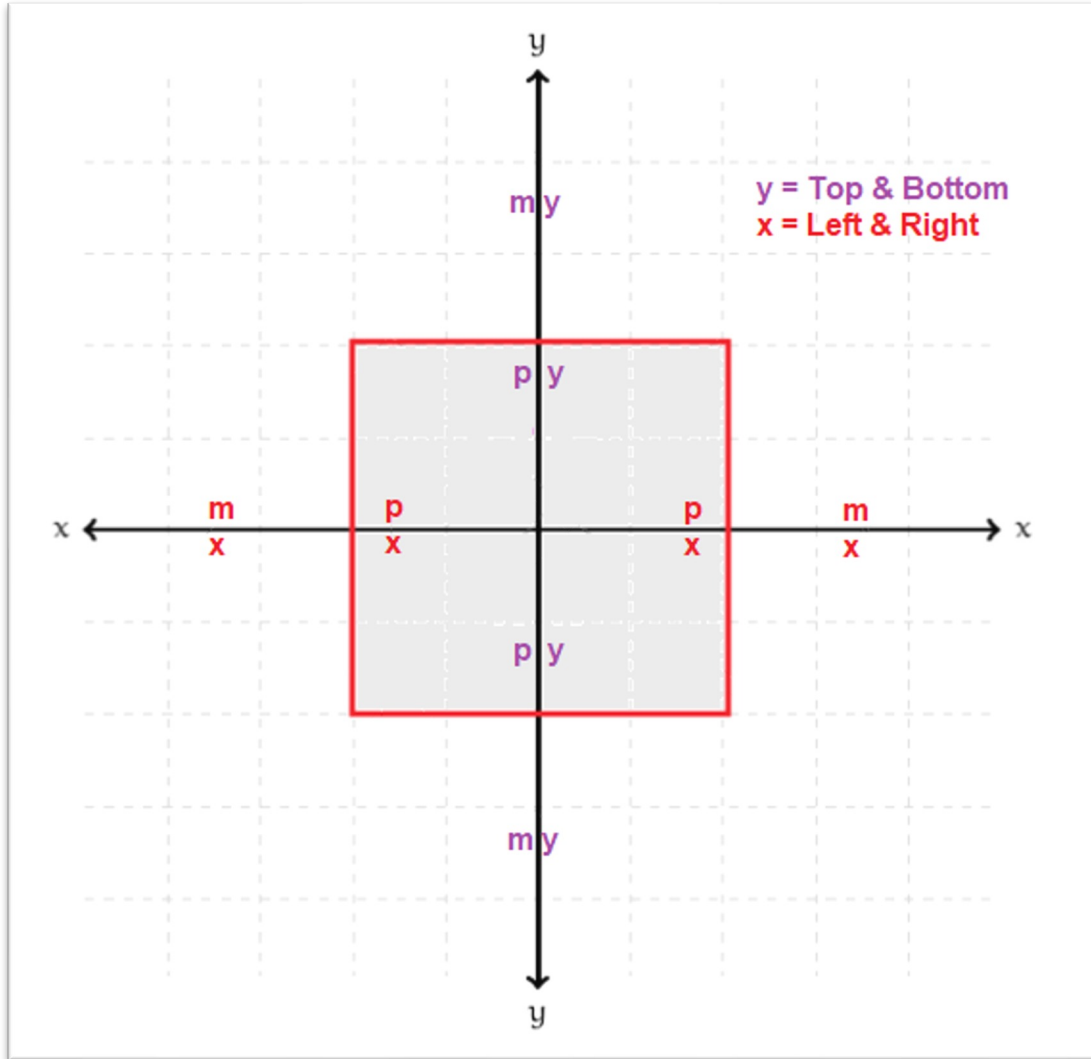
- When is it better to use these classes to set the margin and padding, versus using your custom stylesheet (styles.css), and vice versa?

Use a combination starting with Bootstrap classes and then override it with custom classes to get exactly what you're looking for

**Note:** These types of questions are not meant to make you feel as if you need to memorize the answers, but to continue to get you accustomed to finding information from the documentation! If you can't answer this in-class by memory, by all means, look it up.



# Review: MX, MY, PX, PY



As on a graph:

**y** = Top and Bottom

**x** = Left and Right

**m** = Margin (outside the element)

**p** = Padding (inside the element)

- `0` - sets `margin` or `padding` to `0`
- `1` - sets `margin` or `padding` to `.25rem` (4px if font-size is 16px)
- `2` - sets `margin` or `padding` to `.5rem` (8px if font-size is 16px)
- `3` - sets `margin` or `padding` to `1rem` (16px if font-size is 16px)
- `4` - sets `margin` or `padding` to `1.5rem` (24px if font-size is 16px)
- `5` - sets `margin` or `padding` to `3rem` (48px if font-size is 16px)
- `auto` - sets `margin` to `auto`

**Note:** margins can also be negative, by adding an "n" in front of *size*:

- `n1` - sets `margin` to `-.25rem` (-4px if font-size is 16px)
- `n2` - sets `margin` to `-.5rem` (-8px if font-size is 16px)
- `n3` - sets `margin` to `-1rem` (-16px if font-size is 16px)
- `n4` - sets `margin` to `-1.5rem` (-24px if font-size is 16px)
- `n5` - sets `margin` to `-3rem` (-48px if font-size is 16px)



# Review: Navbar

```
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#nucampNavbar">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="nucampNavbar">
  <ul class="navbar-nav">
    <li class="nav-item"><a class="nav-link active" href="#"><i class="fa fa-home fa-lg"></i> Home</a></li>
    <li class="nav-item"><a class="nav-link" href="aboutus.html"><i class="fa fa-info fa-lg"></i> About</a></li>
    <li class="nav-item"><a class="nav-link" href="#"><i class="fa fa-list fa-lg"></i> Sites</a></li>
    <li class="nav-item"><a class="nav-link" href="contactus.html"><i class="fa fa-address-card fa-lg"></i> Contact</a></li>
  </ul>
</div>
```

## Discuss:

- What does the `.navbar-expand-sm` class do?

States to **expand** the navbar on small+ viewports

- What happens to a Navbar if you use the class `.navbar-expand` without specifying any breakpoint?

It would keep the navbar **expanded** in any viewport (even mobile)

- What happens to a Navbar if you leave out any `.navbar-expand` class completely?

The navbar would be **collapsed** in all viewports (default behavior of navbar without this class)

- What does the `data-bs-target` attribute in a navbar's `<button>` element with a class of `.navbar-toggler` need to contain?

Needs to contain the `"id"` value of the parent `<div>` element that wraps the `.navbar-nav` list and `.nav-item` items





# Review: Breadcrumbs

## Discuss:

- Three types of breadcrumbs:
  - Location
  - Attribute
  - Path (history trail)

<b>Location</b>	Static and show WHERE the page is located in the website hierarchy (i.e. Library > Events) - <b>MOST COMMON</b>
<b>Attribute</b>	Give information that CATEGORIZES the current page based on attributes chosen by the user ( <i>filtered results</i> usually) (i.e. Rugs > Round)
<b>Path</b>	Shows the users steps they too to get to the current page - also known as " <i>history trail</i> ". Tends to be confusing and the least used. Not based on hierarchy but user clicks/nav history



# Review: Icon Fonts

- Icon Fonts and SVG icons are the modern-day answer to icons.

**Discuss:** What are some advantages of using Icon Fonts over a Raster image (i.e. PNG of an icon)?

<b>Size</b>	Smaller in size
<b>Performance</b>	Since smaller in size, they will load faster (SVG's render faster than icon fonts)
<b>Flexibility</b>	Can change color, shadow and other features using only CSS or editing SVG attributes
<b>Scalability</b>	Scale in quality much better (doesn't degrade quality)



# Review: Font-Awesome

## KEY POINTS:

- Font-Awesome is an extremely popular font and icon toolkit.
- Has a free version and a pro version.
- Easy to use, quick grab and paste icon styles.
- Use with `<span>` or `<i>` elements, both are OK:

`<i class="fa-solid fa-phone"></i>`

`<span class="fa-solid fa-phone"></span>`





# Review: Buttons

- The Button component uses the `.btn` base class along with more classes for color and size.



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```



# Review: Buttons

## Discuss:

- What are the three HTML elements most often used for a Button component?

Select the three most common HTML elements that can be made into buttons using Bootstrap classes.

<input checked="" type="checkbox"/>	button ✓
<input checked="" type="checkbox"/>	input ✓
<input checked="" type="checkbox"/>	a ✓

- Which one should you use?

### **Which one?**

So should you use an anchor, input or button? When you're navigating the user to a page or resource, use an anchor. Otherwise, both inputs and buttons are valid. Personally, I prefer to use inputs for submitting and resetting forms and buttons for custom behavior because I think it makes the intent clearer. However, the element you use is entirely up to you. Go nuts!



# Review: Forms

## Discuss:

- What form class do you need to add before you can use the `.form-control-lg` and `.form-control-sm` classes?

The `.form-control` BASE class is also needed if you are going to use the `.form-control-lg` and `.form-control-sm` classes

- What Bootstrap Form class helps you improve the layout and behavior of a checkbox or radio input type?

The `.form-check-input` class applies layout styles for `checkbox` and `radio button` input types



# Review: Tables

**NOTE:** The documentation for Bootstrap Tables is found in the **Content** section

- Tables are built with the `<table>` HTML element.
- Tables should not be used for page layout – we're not in 1995! :)
- Build Bootstrap Tables by adding the `.table` base class to the `<table>` start tag.
- Optional styling classes include: `.table-striped`, `.table-bordered`, `.table-sm`, `.table-dark`, `.table-hover`, more.

**Discuss:** What is the purpose of the `table-responsive` class?  
(hint: check the docs!)



# Review: Cards

- There is **no** `<card>` HTML element.
- Cards are built using the `<div>` element and other HTML elements.
- Cards have no margin and no fixed width to start, and stretch to fill their parent element.
- The optional `card-body` class will add a padded section.
- Other useful, optional Card classes include:
  - `card-title`
  - `card-header`
  - `card-footer`

**Discuss:** Name a few other Card classes and describe what they do.





# Review: Cards

## Discuss:

What card classes is the card below using?

The image shows a 'Reserve a Table' card with the following components and annotations:

- .card**: A red bracket on the left side of the card container.
- .card-header**: A red bracket on the right side of the orange header bar.
- .card-body**: A purple bracket on the right side of the main content area.

The card content includes:

- Header**: 'Reserve a Table' in white text on an orange background.
- Form Fields**:
  - Number of Guests**: Radio buttons for 1, 2, 3, 4, 5, and 6.
  - Date and Time**: A date picker (calendar icon) and a time picker (clock icon).
- Buttons**: A blue 'Reserve' button.



# Review: Images

- Was everyone able to download and extract the img zip file?
- Check: Does everyone have the img folder saved correctly (as **/nucampsite/img/** and **not** as /nucampsite/**img/img/** for example)
- Note: Like Tables, the documentation on Images is in the **Content** section of Bootstrap's docs rather than the Components section
- The **.img-fluid** Bootstrap class makes images responsive by applying these two styles:

```
max-width: 100%;  
height: auto;
```

Expand the image width to the size of the containing element

Scale the image height in relation to the width



# Week 3 Workshop Assignment

- **Goals:**

- ✓ Add an image to the “Featured Campsites” section and style it
  - ✓ Create a reservation form and place it inside a card
  - ✓ Create a button in the header that will scroll down to the form when you click it
- All students should aim to finish and submit your assignment before the end of class today
- Work in your groups, talk to each other, and figure things out together!



# Code Review and Check-Out

- **Submit 2 files:**
  - your updated **index.html** file
  - a **screenshot** of your updated home page.
- Submit your Week 3 Feedback
- Work on your portfolio project
- Have fun learning dynamic components next week!



**Happy  
learning!**