# Bootstrap: Week 3

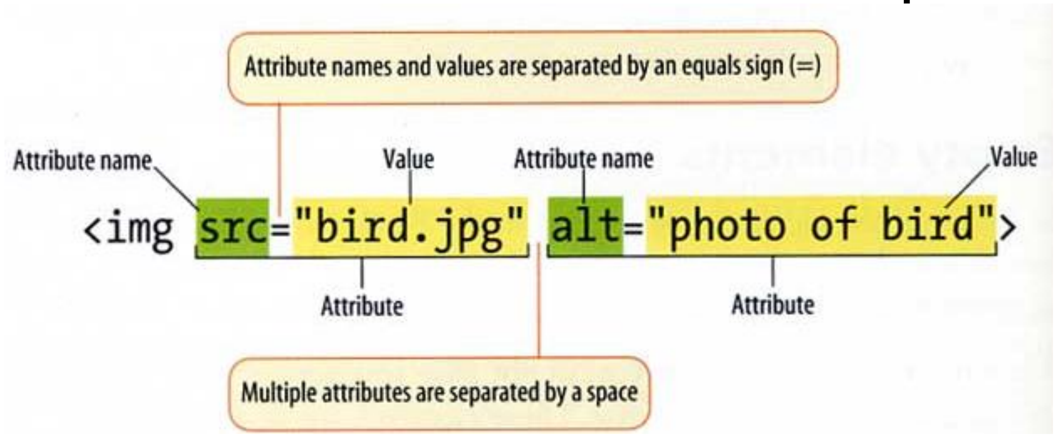
Workshop Presentation

## Today's Agenda

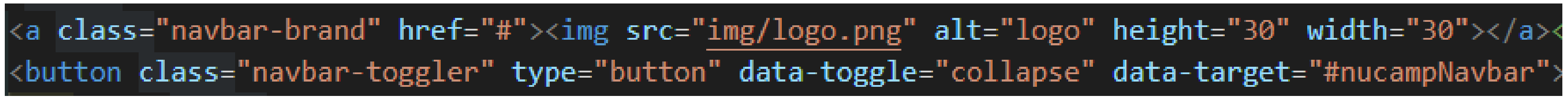
| Activity | Estimated Duration |
|---|---|
| Check-In & Set-Up | 15 mins |
| Review | 75 mins |
| Workshop Assignment Task 1 | 30 mins |
| Break | 15 mins |
| Workshop Assignment Task 2 | 45 mins |
| Workshop Assignment Task 3 | 45 mins |
| Check-Out (Feedback & Wrap-Up) | 15 mins |

# Review: Bootstrap JavaScript Components

- The components you were introduced to this week make use of JavaScript for their functionality, but they do not require you to write any JavaScript.
- Instead, you are using HTML5 custom data-* attributes that Bootstrap has defined in its code to access the JavaScript functionality.


Attribute names and values are separated by an equals sign (=)
Attribute name
Value
Attribute name
Value
Attribute
Attribute
Multiple attributes are separated by a space

```
<img src="bird.jpg" alt="photo of bird">
```



```
<a class="navbar-brand" href="#"><img src="img/logo.png" alt="logo" height="30" width="30"></a><
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#nucampNavbar">
```

## Review: The Nav Component and Tabs

- Tabs are a variant on the base .nav class.

**Discuss together:**

- What Bootstrap class can you use to add a fade effect to tabs when switching between them? What tab class do you use this effect with?

**.fade .tab-pane**

https://getbootstrap.com/docs/4.5/components/navs/#fade-effect

# Review: Collapse and Accordion

**Discuss together:**

- What is the purpose of using the data-parent attribute with a Collapse component?

If **data-parent** is provided, then all collapsible elements under the specified parent will be closed when this collapsible item is shown. (accordion effect)

https://getbootstrap.com/docs/4.5/components/collapse/#accordion-example

# Review: Tooltips

- Find the Options section in the documentation on Tooltips and answer these questions as a class:
  - What is the default location for a tooltip if you don't specify its placement?

    `data-placement="top"`

  - What are the four options for how a tooltip is triggered?

    **click | hover | focus | manual**

## Review: Modals

**Discuss:**

- How is a Modal similar to a JavaScript alert() function's popup?

They both open a dialog box

- How are they different?

You can highly customize a **modal** to include any content you want
e.g. forms, notifications, lightboxes (fill screen with image)

**alert** is more of a push communication to alert the user of something

- Can you use Tooltips and Popovers inside a Modal component?

**Yes**, however keep in mind that a single button can only hold one type of data-toggle (modal/tooltip, etc)

- What does the "show" Modal option do?

It will manually open the modal

# Review: Carousel

**Discuss:**

- What is the difference between **data-slide** and **data-slide-to**?

  `data-slide` accepts the keywords **`prev`** or **`next`**, which alters the slide position relative to its current position

  `data-slide-to` allows you to jump to a specific slide index (zero-based)

- What is the default Carousel interval in milliseconds?

  **5000 ms (5 seconds)**

- What does the Carousel component's "ride" option do, and what is its default value?

  The **data-ride** option/attribute by default will autoplay the carousel AFTER the user manually cycles the first item. Set this to "carousel" if you want it to start cycling by default

# Review: JavaScript Data Types

- The eight data types of JavaScript as of the most recent ECMAScript standard are:
  - **Number** ?
  - **String** ?
  - **Boolean** ?
  - **Undefined** ?
  - Null
  - Object
  - Symbol
  - BigInt

## Review: JavaScript Variables

**Discuss:**

- What is the difference between variable declaration, assignment, and initialization?

**Declaration:** Defining a name for your variable for use in your code (var, **let** & **const**). e.g. **let varName;**

**Assignment:** Sets/re-sets the value of your variable( **=** , +=, -=, *=, and /= ). e.g. **varName = "hello"**

**Initialization:** Specifying an initial value for your variable to start with. e.g. **let varName = "hello";**

- Can you re-declare a variable using the var or let keywords? That is, can you use var twice with the same variable name? What about let?

**var** lets you redeclare variables … beware, since you are more likely to accidentally overwrite something.
**let** does NOT let you redeclare which is generally good practice. Just re-assign the variable instead

# Review: Truthy vs Falsy

Discuss:

- What do the terms **truthy** and **falsy** mean?

**Falsy – Any value that is 0, an empty string (""), undefined, null, or NaN will be evaluated as false**

**Truthy – Any other value that is NOT falsy will be evaluated as true**

- How are they different from **true** and **false**?

**Logical comparison operators ( i.e. && & || ) perform something called "short circuiting" based on whether an operand is truthy/falsy:**

**Used with &&, the operand will either return the last truthy value OR the first falsy value**

- **(1 && 2) will return 2 since they are both truthy values**
- **(1 && undefined) will return undefined since it is the first falsy value**

**Used with ||, the operand will either return the first truthy value OR the last falsy value**

- **(1 || 2) will return 1 since 1 is the first truthy value**
- **(1 || undefined) will return 1 since 1 is the first truthy value**
- **(undefined || null) will return null since the first value is falsy**

# Review: JavaScript Operators

- Discuss:
  - What is the difference between == and ===?

    **== does NOT evaluate the data type (e.g. 1 == "1" will return true)**

    **=== is a strict equality where the data types must match (e.g. 1 === "1" will return false)**

  - Which is considered best practice to use, and why?

    **=== is best practice to avoid any unintended comparisons**

  - What is the meaning of type coercion, and can you give an example of an operator that can cause type coercion?

    **When there is an implicit conversion of a data type (e.g. Number to a String )**

    **The + operator can concatenate a String and Number to a String ( "hello" + 32 = "hello32")**

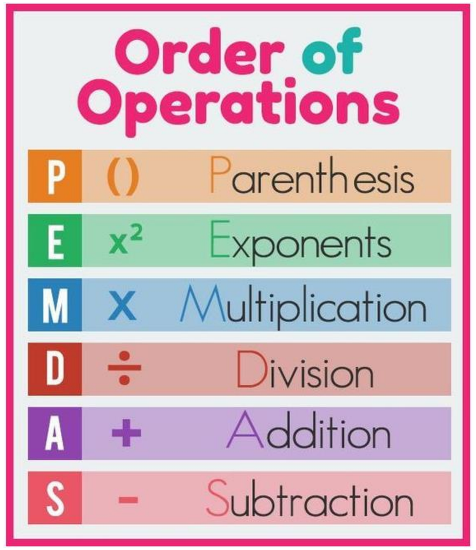
  - What would be returned from this expression?:

```
"banana" || "robot"
```

**"banana" since non-empty Strings are truthy and that's the first truthy value of a Logical Or statement**

# Review: JavaScript Operators (cont)

- Operator precedence: Like in math, there is an order of operations.


Order of Operations
P () Parenthesis
E x² Exponents
M x Multiplication
D ÷ Division
A + Addition
S - Subtraction

- What would the result of these two operations?

```
2 + 3 * 5
```

**17**

```
10 - 4 / 2
```

**8**

- What would you guess would be the result of these two operations?

```
3 > 5 && 2
```

**false**

**Short Circuiting means that once 3 > 5 comes back as false, there is no need to evaluate the right side of the && so the operation returns false**

```
3 > (5 && 2)
```

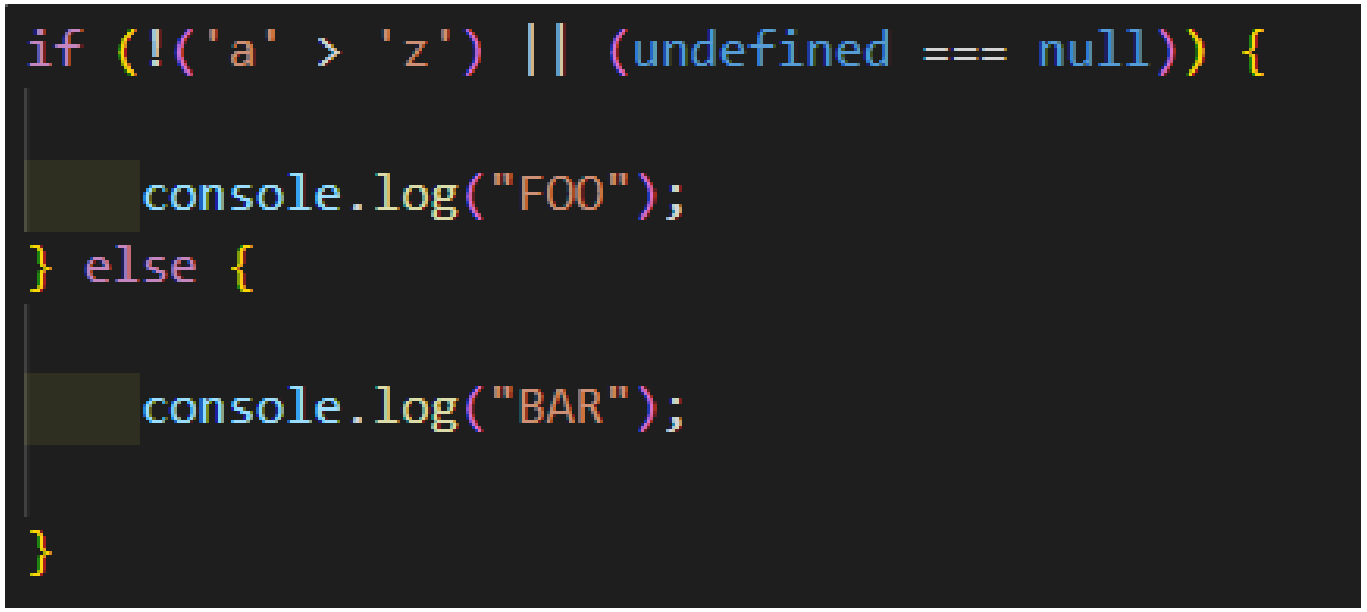
**true**

**(5 && 2) evaluates to 2 (last truthy value); then 3 > 2 returns true**

**Note:** [MDN's Operator Precedence documentation]

## Review: If / ElseIf / Else

- What will print to the console?



```
if (!('a' > 'z') || (undefined === null)) {

    console.log("FOO");
} else {

    console.log("BAR");

}
```

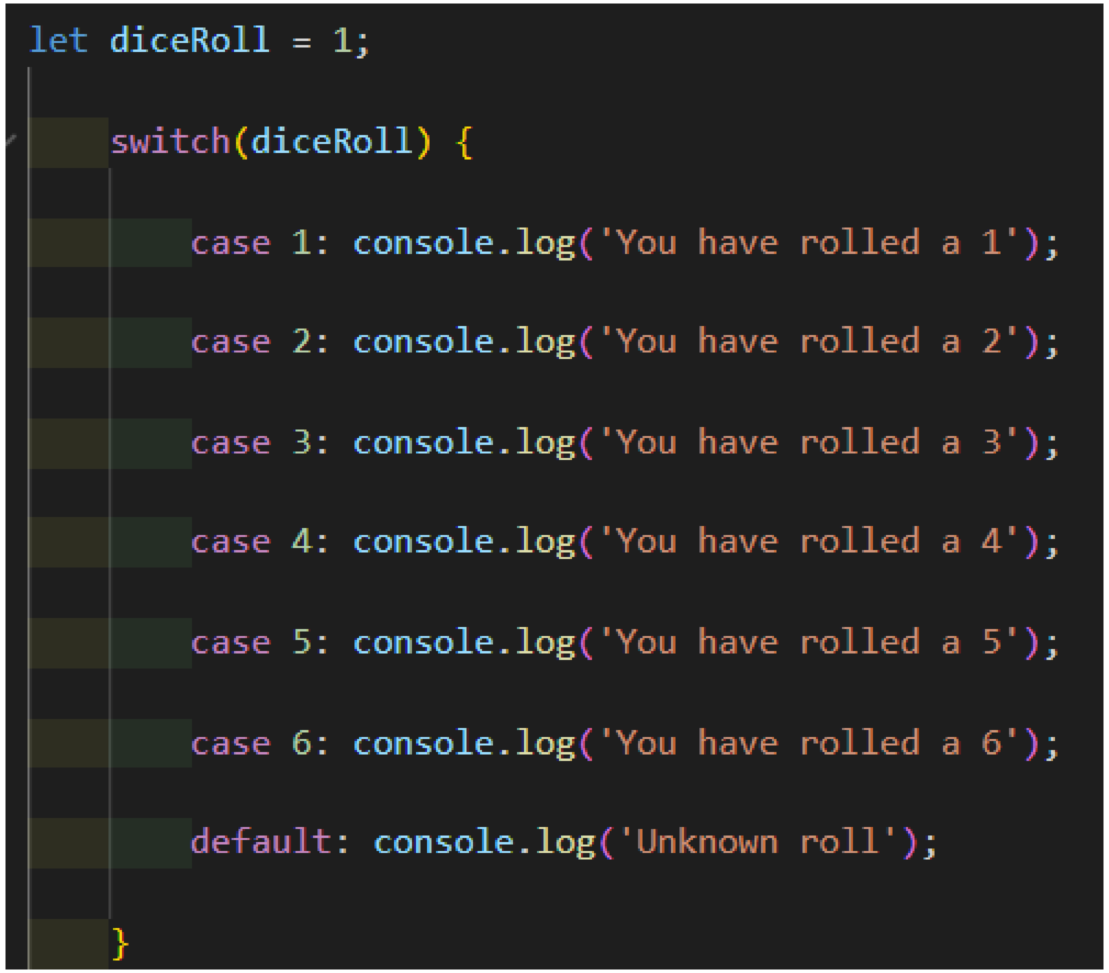
**"FOO" would get printed**

! logical "not" (opposite)

'a' > 'z' = false
!('a' > 'z') = "not" false (true)

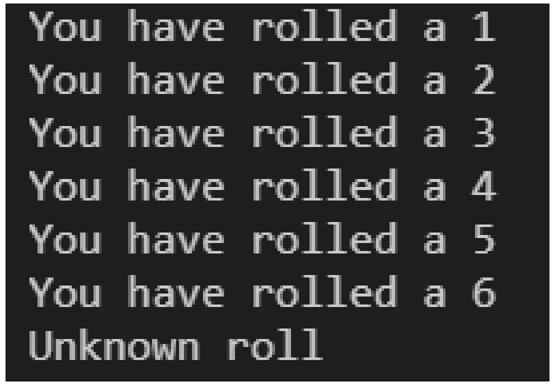
# Review: Switch

- What is wrong with this code? What would happen if you ran it?



```
let diceRoll = 1;

switch(diceRoll) {

        case 1: console.log('You have rolled a 1');

        case 2: console.log('You have rolled a 2');

        case 3: console.log('You have rolled a 3');

        case 4: console.log('You have rolled a 4');

        case 5: console.log('You have rolled a 5');

        case 6: console.log('You have rolled a 6');

        default: console.log('Unknown roll');

}
```

There is no "**break;**" for each case which will result in running the matched case + all cases below it



```
You have rolled a 1
You have rolled a 2
You have rolled a 3
You have rolled a 4
You have rolled a 5
You have rolled a 6
Unknown roll
```

## Week 3 Workshop Assignment

- All students should aim to finish and submit your assignment before you leave today.
- Work in pairs, or groups of three. Talk to each other and figure things out together!
- 10-minute rule during workshops: If you and your paired partner have spent more than 10 minutes trying to figure something out, ask your instructor for help.

Happy learning!

# Review: Challenges / Quiz

- It is important that students have ample time to complete the assignment during the workshop.
- If there is time left *after* students have completed the Workshop Assignment, review the Week 3 challenges and quiz together.