Bootstrap: Week 5

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	60 mins
Portfolio Project Presentations	30 mins
Check-Out (Feedback & Wrap-Up)	15 mins

Review: Function Declarations

Discuss:

- What does it mean to say that function declarations are "hoisted"?

  Hoisted means that at runtime JS will move all functions to the top of their scope

- In the code below, which is the parameter and which is the argument?

```
function inchesToCM(inches) {
    return inches * 2.54;
}

const lengthInches = 3;
const lengthCm = inchesToCM(lengthInches);
```

Argument = lengthInches
(pass variables to functions)

Parameter = inches
(take in arguments as local variable)

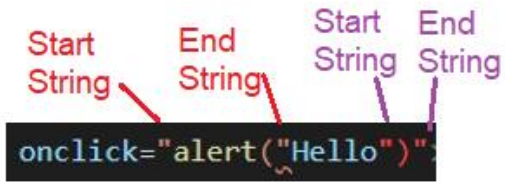
- In the code above, what would happen if you tried to access the variable named **inches** from outside of the function?

  JS would throw an error that "inches" is not defined since it's only defined in the function's scope

Review: JavaScript Events

Discuss:

- In the following inline onevent handler: `onclick="alert('Hello')"`



  - Why is it necessary to use single quotes around the 'Hello!' Instead of double?

To prevent JavaScript from mistaking "alert(" as a string, since once a single or double quote is opened, it will look for another corresponding single or double quote to close the string

- What is wrong with the following code? Assume that there is a valid button element and a valid function named runSomeCode.

```
document.querySelector("button").onclick="runSomeCode()";
```

The onclick JS method is used incorrectly.
Should be .onclick=runSomeCode; (no quotes or parenthesis)

- What is the main difference between using the on*event* handlers (such as the onclick event handlers shown above) vs event listeners?

Even listeners are the more modern way to implement event handlers and support more than 1 event listener to an element

```
// Multiple listeners can be added to the same event and element
button.addEventListener('click', changeText)
button.addEventListener('click', alertText)
```

Review: Function Expressions

Discuss:

- Here is a function declaration. How would you write this as an (anonymous) function expression?

```
function getFullName(firstName, lastName){
    return firstName + " " + lastName;
}
```

```
// Anonymous Function (function with no name)
const getFullName = function (firstName, lastName){
    return firstName + " " + lastName;
}

// Calling the anonymous function
console.log("hello", getFullName("Steve", "Thompson"));
```

- Are function expressions hoisted?

No, they are only valid after you reach them in code

- Are semicolons used after the ending curly brace of a function expression?

No

Review: Arrow Functions

Discuss:

- When do you need to use the curly braces and the return keyword with an arrow function?

  When your function body contains more than a single return statement (more than 1 line)

- When can you omit the parentheses around the parameter list for an arrow function?

  When there is only 1 parameter

```
const double = a => a * 2;
```

- How would you write the same function from the previous slide as an arrow function?

```
function getFullName(firstName, lastName) {
    return firstName + " " + lastName;
}
```

```
const getFullName = (firstName, lastName)  => firstName + " " + lastName;
```

Review: Var, Const, Let

Discuss:

- When should you use const instead of let?

  When you don't want the variable value to be reassigned

- What type of scope does var have?

  Function block scope

- What type of scope do const and let have?

  Block scope (any block of code, including but not limited to functions)

- What is the meaning of global scope?

  Global scope is accessible to your whole project (not very secure and prone to bugs)

Week 5 Workshop Assignment

- Task 1: 10:30am – 11:00am
- Break Time: 11:00am – 11:15pm

- Task 2: 11:15am – 12:15pm
- Portfolio Presentation/Overviews: 12:15pm – 12:45pm
  - Be prepared to screenshare

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 5 challenges and quiz together.

Review: Week 5 Quiz

When an arrow function only has a single statement inside its function block, you can omit the curly braces around that statement as well as the return ✓ keyword.

What is the value returned by evaluating the below expression?

109 && 21 && 0 && 88

Select one:

- a. 109
- b. 88
- c. 21
- d. 0 ✓ Correct. If there is a falsy value as a Logical And operand, evaluation will stop at the falsy value and it will be returned as the result of the operation.

Match the following.

Variables declared with the **var** keyword have this kind of scope.	function ✓
Variables declared with the **let** keyword have this kind of scope.	block ✓
Variables declared with the **const** keyword have this kind of scope.	block ✓

Your answer is correct.

The correct answer is: Variables declared with the **var** keyword have this kind of scope. → function, Variables declared with the **let** keyword have this kind of scope. → block, Variables declared with the **const** keyword have this kind of scope. → block

Review: Week 5 Quiz

What is the value returned by evaluating the below expression?

null || undefined || ""

Select one:

- a. undefined
- b. null
- c. "" ✓ Correct. The Logical Or operator will test each of its operands and will return the first operand with a truthy value. If there are none, it will return the final operand (with a falsy value).

You cannot assign a value to a variable without first using **var, let,** or **const** to declare the variable.

Select one:

- True
- False ✓

Correct. Unless you are using strict mode, JavaScript allows you assign a value to a variable without first declaring it (then gives it global scope), but this is bad practice and a common mistake. You should never deliberately assign a value to a variable without first declaring it, to avoid creating global variables unintentionally.

The correct answer is 'False'.

Function declarations are hoisted, but function expressions and arrow functions are not hoisted.

Select one:

- True ✓
- False