IST 363

# Lab 07

## This Lab Covers

* JavaScript – variables, datatypes, loops, random numbers

## Lab Corrections

No corrections this week. Woot!

## JavaScript 1

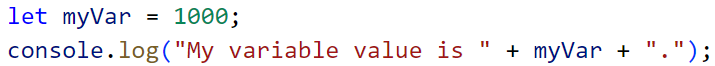
In lecture this week you saw:

* [JavaScript comments](https://www.w3schools.com/js/js_comments.asp)
* [where to put your JavaScript (embedding or external)](https://www.geeksforgeeks.org/what-is-external-javascript/)
* [Console.log(); and using the inspector to view the console](https://www.w3schools.com/jsref/met_console_log.asp)
* [assigning basic variables](https://www.w3schools.com/js/js_variables.asp)
* [Data types](https://www.w3schools.com/js/js_datatypes.asp)
* [if…else statements](https://www.w3schools.com/js/js_if_else.asp)

The links above will take you to documentation about all these concepts in case you need to review. Below we are going to add onto our learning about variable data types with arrays, objects and arrays of objects. We’ll also talk about random numbers and loops.

**Directions**: You will write the code for the following questions in one HTML file with embedded JavaScript. Please use comments (// #1) to number the questions so I know where to look for each questions’ answer.

### Logging Multiple Variables

1. You can console log more than one variable at a time along with text by just adding the plus sign between text, and variables. Here’s an example of what I mean:  
     
     
     
   This would output “My variable value is 1000.” to the console.  
     
   **Task – Logging Text and Variables**  
   Set a variable called favNumber equal to your favorite number write code to console log this variable along with the text. and console log this.  
     
   My favorite number is 5.

### Arrays

1. An array (you may know them as lists) is a special variable that can store multiple values in a single variable. Arrays in JavaScript are defined using square brackets [ ], and each value inside is separated by a comma. Here’s an example of an array that stores a list of colors:  
     
   A close-up of a computer code

   AI-generated content may be incorrect.  
   1. Arrays are zero-indexed, meaning the first item is at index 0.
   2. You can access an element using its index (colors[1] gets "blue").
   3. The .push() method adds an item to the end of an array. So to add the value “orange” to the colors array you could say colors.push("orange");

**Task Working with Arrays**Create an array named favoriteFoods that contains at least four of your favorite foods.  
Print the second item in the array to the console.  
Add a new food to the array using .push().  
Print the entire updated array to the console.

### Looping Arrays

1. When working with arrays, you often need to access and process each element. The best way to do this is by using a loop. A for loop is commonly used to iterate over an array. Since arrays have indexes, a for loop can run from 0 to the last index of the array. The last index of an array with 4 items is 3 (array.length - 1). This is because the first item is at 0.   
     
   Here’s an example of an array that loops over the colors array.

How it works:

* The loop starts at i = 0 (first element).
* It runs as long as i < favoriteFoods.length (until it reaches the last element).
* i++ increases the index by 1 each time, moving to the next element.  
    
  A computer code with text

  AI-generated content may be incorrect.

**Task Looping thru an Array**Write a for loop to print out your favorite foods. Add some text to help you give context so that the output looks like the example below. You will need to use your skills from #1 to do this.  
  
Favorite Food 1: pizza  
Favorite Food 2: sushi  
etc…

### Filtering Arrays

1. In class we worked with if…else statements. Remember an if statement is used to execute code only if a certain condition is true. You can combine if statements with loops to check each item in an array and perform an action based on a condition.  
     
   **Task Filtering an Array**Copy and paste your loop from #3 into this question and rewrite it to exclude console logging the second entry. For example, my array contains pizza, sushi, chocolate, and French fries. When I console log it for this question I won’t see the sushi entry.

### Objects

1. In JavaScript, an object is a collection of related data and functionality, stored as key-value pairs. Objects allow you to group related information together in a structured way.

Why Use Objects?

* Instead of using multiple separate variables, you can store everything in one object.
* Objects are great for representing real-world entities like a person, a car, or a student.
* They help keep your code organized and easier to manage.

Here’s an example object:  
  
A computer code with text

AI-generated content may be incorrect.  
Objects use keys (e.g., name, age) to store values ("Emily", 20).

You access values using dot notation (objectName.key) or bracket notation (objectName["key"]).

**Task Creating and Using Objects**Create an object called book with the following properties:

* title (a string for the book title)
* author (a string for the author's name)
* yearPublished (a number for the publication year)
* isAvailable (a boolean indicating if the book is available)

Print the book’s title and author to the console.

Use an if statement to check if the book is available. If isAvailable is true, print "This book is available for checkout." Otherwise, print "Sorry, this book is currently checked out."

### Other Kinds of Loops

1. There are other ways to loop through things in JavaScript besides the for loop you used above. A while loop runs as long as a condition is true. Use it when you don’t know beforehand how many times the loop should run.

Example: Printing numbers while x is less than 5  
  
A computer screen shot of a code

AI-generated content may be incorrect.  
  
**Task While Loops**Write a while loop that console logs even numbers and stops when you get to 20.

### Random Numbers

1. In JavaScript, you can generate random numbers using the Math.random() function. This function returns a decimal number between 0 (inclusive) and 1 (exclusive).

Generating Random Whole Numbers

To get a random whole number, you can multiply by a range and use Math.floor() to round down.  
  
A close-up of a text

AI-generated content may be incorrect.

* Math.random() generates a number between 0 and 1.
* Multiplying by 10 scales it to a range of 0-9.999.
* Math.floor() removes the decimal, keeping numbers 0-9.
* Adding 1 shifts the range to 1-10.

**Task Using a Random Number**

Write JavaScript that simulates rolling a six-sided die and returns a random number between 1 and 6. Console log the result.

**All of this week is going to be magically combined next week to do something wonderful to our web pages. The End!**