Week 5 Lesson 10 Readings

***An Introduction to Functions, Execution Context and the Call Stack Video:***

JavaScript executes code line by line.

We are declaring constants.

Stores data in memory.

Execution context = process of going code line by line and the memory we store any variables or functions that get declared or announced.

Live memory of variables with data known as a Global Variable Environment.

Single threaded = one thing at a time.

Synchronous execution

The thread of execution = we go through the code in the function line by line.

Local memory = variable environment where anything defined in the function is stored.

We keep track of the functions being called in JavaScript with a “Call Stack.”

Tracks which execution context we are in – that is, what function is currently being run and where to return to after an execution context is popped off the stack.

One global execution context, multiple function contexts.

***How to Understand Callbacks & Higher Order Functions Video:***

Function copyArrayAndMultiplyBy2(array) {

Let output = [];

For (let I = 0; i < array.length; i++) {

}

Return output;

Const myArray = [1,2,3]

Let result = copArrayAndMultiplyBy2(myArray)

CopyArrayAndMultiplyBy2 = function

myArray = 1,2,3

results = 2,4,6

local memory:

array = 1,2,3

output = 2,4,6 = the results

By doing all these exercises, we are breaking the repeating rules.

We could generalize our function so that we pass in our specific instructions only when we run the

copyArrayAndManipulate function.

How is this possible?

Functions in JavaScript = first class objects

They can co-exist with and can be treated like any other JavaScript object.

Assigned to variables and properties of other objects

Passed as arguments into functions

Returned as values from functions

Higher-order functions

Takes in a function or passes out a function

Just a term to describe these functions – any function that does it we call that – but there’s nothing different about them inherently.

Callbacks and higher order functions simplify our code and keep it DRY = Don’t Repeat Yourselves

And they do something even more powerful

They allow us to run asynchronous code.