**Part 1 (Total 55 minutes)**

**From DOM practice to coding Functions**

# Reflection / discussion (15 minutes)

* What is the basic anatomy of a function?
* What is the basic anatomy of an object ?
* How is a function different from the commands we’ve been writing so far?

**Writing functions in JavaScript (40 minutes)**

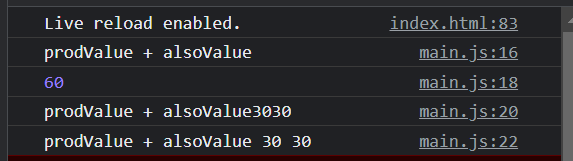
A function encapsulates a set of commands that we want should be executed together. It can optionally take in some input in the first line, and it can return some output in the last line.

Putting statements and expressions into functions can make code more modular, less prone to errors, and easier to debug.

**Function with no arguments and no return value**

1.1 Open Session04.class/appscripts/main.js Paste the code in the first box into main.js. MSWord can create formatting issues.You can instead copy and paste the text from secondbox.js provided in the appscripts folder.

1.2 Save. Open Session04.class/index.html in the browser, and use the developer tools to open the console window. What is printed?



**//function declaration**

function myFunction(p1, p2) {

// this function takes in two ARGUMENTS, but any number of arguments are possible

var prod = p1\*p2; // this variable is declared inside the function so it does not exist once the function call ends

console.log(prod); // The function prints the product of p1 and p2 to the console

}

**//function call**

myFunction(5,6);

var a = 5;

var b = 6;

myFunction(a,b);

1.3 Now “comment out” all the previous code by selecting all and pressing Control + / on your windows machine, or command + / on your Mac

Because it is commented out, if you save and run it now, nothing will be printed to the console. Test it out, or check with your partner to see what worked for them.

**From pseudocode to function**

1.4 Now, to take the statements you wrote in 1.3, and then paste them within the brackets in the first box. Try it out. Does it still work?

**Part 2 Variable scope**

**Summary of learnings**

Variable scope (the difference between variables declared inside function and those declared outside functions)

* ***Variable Declaration:*** *Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the var keyword or the let keyword.*
* ***Variable Initialization:*** *This usually occurs when a variable is declared. Here the variable is assigned a memory or space by the JavaScript engine. Because of this, once a variable is declared, it takes a value of undefined even before assignment.*
* ***Variable Assignment:*** *Variable assignment is usually the most important step when using a variable. Here the variable is assigned data which is a value using the assignment operator "=". Values in JavaScript take one of the standard JavaScript datatypes (string, number, boolean, null, undefined)*

**Reference:** https://www.tutorialspoint.com/javascript/javascript\_variables.htm

**Long-term and short-term memory loss with variable scope (20 minutes)**

2.1 Let’s play with a new code snippet. It is provided as part2.js in the appscripts folder. Notice the pre-function block of statements before the function, the function block statements within the function, and the post-function block. You do not need to paste this code anywhere, as it is already linked to your index.html through a line within the head element. You should now uncomment this line in index.html.

//PRE-FUNCTION BLOCK

var counter2; //LINE 1: declaring a variable. Current value = undefined  
counter2 = 5; //LINE 2: assigning a variable. Current value = 5

function foo() {

//FUNCTION BLOCK

counter2 = counter2 + 1;  
 console.log(“Inside the function block: counter2’s value is:” + counter2);

}

//POST-FUNCTION BLOCK

console.log(“Post-function block: counter2’s value is:” + counter2);

foo();

2.2 Now, to understand variable scope, we will edit part2.js. Shuffle Line 1 and Line 2 between different positions in the pre-, post- and the function block. Remember that Line 1 should always occur before Line 2!

2.3 In other words, move your declaration and assignment of counter2 so that you try out all the conditions in the Table 1.

**Please fill up the following table with your findings. Note the ORDER in which console.log statements are printed. Why is this so?**

**Table 1**

|  |  |  |
| --- | --- | --- |
|  | **Output on the console** | **Why** |
| **Line 1 and Line 2 are in current position** | Post: 5 Inside: 6 | Cause changing a variable inside a function-scope doesn’t affect the global variable |
| **Line 2 is inside function block BEFORE “counter2 = counter2+1” expression** | Post: undefined Inside: 6 | Counter2 is still unassigned while foo() function is executed  First, inside the foo() function counter2 is assigned 6, but only in the function-scope |
| **Line 2 is inside function block AFTER “counter2 = counter2+1” expression** | Post: undefined Inside: 5 | Counter2 undefined until it being assigned 5 on LINE 2, before the print().  Counter2 is still undefined in the global scope |
| **Line 2 is in post-function block, after function call** | Post: undefined  Inside: NaN (Not-a-Number) | By trying to do a arithmetic operation on a “undefined”-value, counter2 is being reassigned to NaN.  Since the print is before the assignment of the variable, it still prints that counter2 is still undefined. |

**Part 3 Objects**

**Initializing objects (5 minutes)**

* 1. Create two variables with names 'point1' and point2'
* We want point1 and point2 to refer to two objects which have the same properties. For example, they can both be points with an x and a y coordinate.
* So, create two variables point1 and point2, and assign them to objects that have two properties, 'x' and 'y', giving 'x' and 'y' different values for each of the points.
  1. Create a variable with the name result. It should have three properties, “sum”, “difference” and “product”.

**References:**

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects>

**Returning an object instead of a single value (15 minutes)**

* 1. Create a new function "multi" that takes 2 numbers as arguments, and returns an object called result with 3 properties, "sum", "difference" and "product", which are assigned the values for the sum, the difference, and the product of the function arguments.
* Call the multi function. Since it takes in some parameters and returns a value, how would you need to call it?

3.4 Add 3 sentences to the innerHTML of the article element presenting the different properties of the object returned by your call to multi. The message should start as “**Output of Step 3.6 is:”**

* Take a moment to add comments to your code.

**Operations on object properties (15 minutes)**

3.5 Create a new function called 'pointsum', that expects two points as input arguments (that is, two objects each with an 'x' and a 'y' property). The function should return a new point object that is the sum of the two input points.

* The sum is calculated by summing the individual x and y properties of the two arguments to each other.
  1. As before, add sentence to your article that expresses the results of calling your function. The message should start as “**Output of Step 3.6 is**:”
* Hint: JSON is a built-in object that has a stringify method that "pretty prints" objects provided as an argument. Use JSON.stringify() to print the result of the call to your pointsum function.
* Take a moment to add comments to your code.

**References**

<https://levelup.gitconnected.com/5-ways-to-log-an-object-to-the-console-in-javascript-7b995c56af5a>

**Reflection and discussion**

* What would an object look like that has a function as a property?
* How would you call a function that is a property of an object?
  + 1. Now paste the code in the second box into your main.js.

MSWord can create formatting issues.You can instead copy and paste the text from secondbox.js provided in the appscripts folder.

**//function declaration**

function myFunction(p1, p2) {

// this function takes in two ARGUMENTS, but any number of arguments are possible

var prod = p1\*p2; // this variable is declared inside the function so it does not exist once the function call ends

return prod; // The function returns the product of p1 and p2

};

//function call

prodValue = myFunction(5,6);

var a = 5;

var b = 6;

alsoValue = myFunction(a,b);

alsoValue = myFunction(a,b);

//what will be printed?

console.log("prodValue + alsoValue");

//what will be printed?

console.log(prodValue + alsoValue);

//what will be printed?

console.log("prodValue + alsoValue" + prodValue + alsoValue);

//what will be printed?

console.log("prodValue + alsoValue" + " " + prodValue + " " + alsoValue);

* + 1. Save and reload the index.html. Open the console window and record the outputs on the console log. The line number next to the output will tell you which line number of your main.js it corresponds to.



**2.2.4 Please fill up this table according to the output that was generated.**

|  |  |  |
| --- | --- | --- |
| Statement | Output | Explain why |
| console.log("prodValue + alsoValue"); | prodValue + alsoValue | Prints string |
| console.log(prodValue + alsoValue); | 60 | Prints the result of (prodValue + alsoValue) |
| console.log("prodValue + alsoValue" + prodValue + alsoValue); | prodValue + alsoValue3030 | Prints a string first, then concatenates it with prodValue and alsoValue. Happens because we don’t have parentheses around (prodValue + alsoValue) |
| console.log("prodValue + alsoValue" + " " + prodValue + " " + alsoValue); | prodValue + alsoValue 30 30 | Similar to the one above, but added the whitespaces |

**References:**

1. <https://www.w3schools.com/js/js_functions.asp>
2. <https://www.geeksforgeeks.org/javascript-console-log-with-examples/>

**(Suggested 10 minute break)**

**Part 3 (Total 1 hour++)**

* **All the functions to follow should be added into your Session03.class/appscripts/main.js**
* **We will judge the completeness of the work based on the console messages that are printed.**

Summary of learnings at the end of this part:

* If you create a new function, you also have to make sure that you call it otherwise it will not execute.
* If it takes in arguments, then you have to pass it arguments when you call it.
* If it returns a value, then you have to assign it to a variable when you call it.
  + Eg var myFactorialValue = myFactorial(5);
  1. **Passing arguments to functions (10 minutes)**

3.1.1 In your main.js, add a new function named 'sumMaker ' that takes no arguments. All this function should do ifs print a console message saying that you are in the function.

* First, create an empty function, similar to the one in the first box.
* Then, fix its name from myFunction to sumMaker,
* Then, remove anything in the round brackets because it takes no parameters.
  + 1. Reload the page. Check the console window. The function will not be executed, because you have to call a function for it to be executed. So far you have only defined it.
* Now, add a function call to main.js after the statements that define the sumMaker function, similar to the last few lines in the first box. Save and reload in the browser. Does it work now?
  + 1. Now modify sumMaker so that it takes two numbers as arguments, prints a console message saying you are in the function.

**You can print the outputs to the console log by including this statement in the sumMaker function:**

**console.log(“Output of Step 3.1.3 is : I am in the function“);**

* + 1. Save and reload. It will throw an error because it expected arguments, but your function call is not passing any arguments to it.

Before you call sumMaker, create two variables, 'x', and 'y', and assign numbers to them. Now use them in the sumMaker() call e.g., sumMaker(x,y);

Does it work now? Take a moment to add comments to your code.

* 1. **Saving the output from a function into a variable (15 minutes)**

3.2.1 Now create a function sumMaker2 that does the exact same thing as before, but now it also returns the difference between the two numbers it received as arguments.

Now you will have to change the way you call the function. When you call the function, store the difference in another variable.

* + 1. Use the variables and the value returned by calling the 'sumMaker2' function on them to construct a sentence printed in your article element. The sentence should read something like: **"Output of Step 3.2.2 is: The difference between 7 and 100 is 93"**
  + Hint: if you don't want to overwrite the text already in your article element, put a line break ( <br>) at the end of your previous welcome message, then use '+=' to add the new text.
  + Take a moment to add comments to your code.