NM2207

Week 3

Session 03 Classwork

Overview of what we'll do today:

- Practice writing pseudocode
- Practice the use of objects and functions

Before attempting the challenges, you are expected to have watched and coded along with the Lecture videos. A tutorial is meant to practice the skills presented in the video lecture, and show you more applications of it. Tutors will explain the challenges and answer the questions you may have**.**

The challenges are due to be completed at the end of class each week for full credit which is also attendance. Submitting by midnight of the same day (ie Thursday night) accounts for half the credit.

Part 1 (Total 35 minutes)

Summary of learnings of Part 1

- Remember the first step in writing any pseudocode or code is always to declare and initialize your variables.
- If you are extracting the HTML content of an element on your index.html
    - The first step is to assign that element to a variable using getElementById(). See Table 1 for how it's done.
    - Inside the brackets of getElementById() put the id of the element you want in quotes
    - And then to set its innerHTML to whatever you want.
    - If you are extracting more than one value, you will correspondingly need to do the getElementById() step each time, with different id names. Remember to assign them to different variables. See Table 1.

From Pseudocode to JavaScript DOM

First, we will write pseudocode ie a list of steps to solve a problem.
If you need help, please refer to the slides of Lecture 3.

1.1 Warm up

Write the pseudocode to convert a Celsius temperature reading input by a user into a Fahrenheit temperature reading and return the result

1.2 From problem to pseudocode (15 minutes)

We reuse this in Part 2 so do it well.

1.2 Write the pseudocode to calculate the distance traveled, given a speed and a time input by the user, and return the result.

1.2.1 Modify the pseudocode to extract the speed and the time from the innerHTML property of different elements in the Document Object Model. See Table 1 for how it's done.

1.2.2 Modify the pseudocode to print the distance traveled to the innerHTML of a third element instead of returning the result.

1.2.3 In preparation for coding this up, go to your Session03.class/appscripts/main.js in the Sublime editor. Use code to add speed and time values to two HTML elements in index.html. See Table 1 for how it's done.

1.3 Practice with the Document Object Model

Now, practice getting and setting text and style properties for different HTML elements on W3schools

https://www.w3schools.com/JSREF/prop_html_innerhtml.asp
https://www.w3schools.com/JSREF/prop_node_innertext.asp
https://www.w3schools.com/JSREF/prop_html_style.asp

Table 1: For practice. (20 minutes)

Description	**Code**
Command to get an HTML element and assign it to a variable	var speedEl = document.getElementById("aside"); //var can take any name, and look for any id
Now we can refer to all the properties of the HTML element, such as	
its HTML content	speedEl.innerHTML
Its color (works for headings, paragraphs and divs)	speedEl.style.color
backgroundColor (works for paragraphs and divs)	speedEl.style.backgroundColor
height (works for paragraphs and divs)	speedEl.style.height
margin (works for paragraphs and divs)	
padding (works for paragraphs and divs)	
overflow (works for paragraphs and divs)	

Part 2 (Total 55 minutes)

From DOM practice to coding Functions

Discussion (to be covered by the tutor now or during Part 2)
Slides available on LumiNUS

Reflection / discussion (15 minutes)

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

For later: Video H3.1 on LumiNUS explains how putting statements and expressions into functions can make code more modular, less prone to errors, and easier to debug.

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.

2.1 Function with no arguments and no return value

2.1.1 Open Session03.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.

2.1.2 Save. Open Session03.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);
```

2.2 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.

2.2 From pseudocode to function

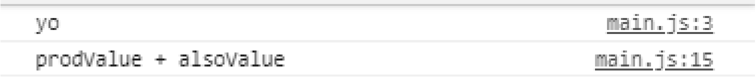
2.2.1 Now, to take the statements you wrote in 1.3.3, and then paste them within the brackets in the first box. Try it out. Does it still work?

2.2.2 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);
```

2.2.3 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.



```
yo                                          main.js:3
prodValue + alsoValue                       main.js:15
```

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

Statement	Output	Explain why
console.log("prodValue + alsoValue");		
console.log(prodValue + alsoValue);		
console.log("prodValue + alsoValue" + prodValue + alsoValue);		
console.log("prodValue + alsoValue" + " " + prodValue + " " + alsoValue);		

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);

3.1 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 is 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.

3.1.2 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?

3.1.3 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");

3.1.4 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.

3.2 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.

3.2.2 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.

Bonus

The next few questions get you more practice with math and functions. We will use them in forthcoming weeks, so its good to try to understand their logic right away.

2.3 Notice the map() function provided below. Run through the function “manually”:

```
//function declaration
// maps x in  the interval [a,b] into the interval [m, n]
let map =function (x, a, b, m, n){
  let range = n-m;
  // x is 'proportion' of the way from a to b
  // e.g. if a=10, b=20, and x=15, x is half (.5) of the way from a to b
  let proportion = (x-a)/(b-a);
  return (m + proportion*range);
}
```

2.3.1 Explain what the function is doing:

Question	Answer
What is its purpose?	
What does it take as input?	
What does it provide as output?	
What happens when x is equal to a?	
What happens when x is equal to b?	
What happens when x is exactly halfway between a and b?	

2.3.2 Now, paste it into main.js and test it with arguments supplied in the function call. Were you right?

2.4 Given the formula x = r sin(angle in radians) and y = rcos(angle in radians), write the pseudocode to return the linear x and y components of a point on the radius of a circle, given r and angle.

- If this body is moving along a circular path at a speed “rate”, calculate the angle at any time as a function of initial angle and rate
- Now that you know the angle, write the pseudocode to return its x and y components at any given time. How can you make sure that y is always positive?

Reference:

1. We will use this to move objects in two dimensions, using Math.sin and Math.cos functions in JavaScript. https://nm2207.org/creativeweb/tutorials/timeranimationweb/SinCos.html

---

This next section should be attempted after watching and coding along to the Assignment videos for this week. However, it is provided here because it directly builds upon the skills of Part 3.

Initializing objects (5 minutes)

3.3 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.

3.4 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)

3.5 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?

3.6 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)

3.7 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.

3.8 As before, add sentence to your article that expresses the results of calling your function. The message should start as “**Output of Step 3.8 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 resultof 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

- (Not practicing today but usually on the quiz) What would an object look like that has a function as a property?
- (Not practicing today but usually on the quiz) How would you call a function that is a property of an object?