aWeek 3 Day 4&5 18/5/23

# Journey so far

So far, we’ve looked at:

* Variables
* Comparison and Logical operators
* Conditionals
* Loops
* Data Types
* Objects
* Arrays

# What are Functions?

A function is a type of object and is a block of organized code that is used to perform a task.

* We “call” or “invoke” a function at any point in our program if we require to
* run it.
* A function can be called by other code, by itself, or by a variable that refers to
* the function.

Functions are generally used to perform some action .

* E. g: Sending an email when a user clicks on a button. We define the code for
* sending the email in our function, but we only call the function when the
* button is pressed.

# Functions

* A picture containing text, font, screenshot, line

  Description automatically generatedA function is a block of organized, reusable lines of code that is used to perform a single, related action.
  + A function definition (also called a function declaration, or function statement) consists of the **function keyword**, followed by:
* The ***name*** of the function.
* A ***list of parameters*** to the function, enclosed in parentheses and separated by commas.
* The JavaScript ***statements*** that define the function, enclosed in curly brackets {...}.

# Function syntax

* Let’s look at how we write functions in our code

Function declaration (below).

A picture containing text, font, screenshot, line

Description automatically generated

# Function calls

A picture containing text, font, screenshot, line

Description automatically generated

To call a function we simply specify it’s name and then open and close the parentheses().

# Function parameters

* + When creating a function, you can define a set of ***optional parameters*** that it can take.
    - * These parameters will be defined in the parentheses that we’ve previously left empty.
    1. A picture containing text, font, screenshot

       Description automatically generated
       - In this example, we are allowing our function to take two parameters num1 and num2
* Separate each parameter with a comma (you can have as many as you'd like!)
* When we call our function, we can define what we want the values of our parameters to be, in this example **num1 = 3** and **num2 = 4.**

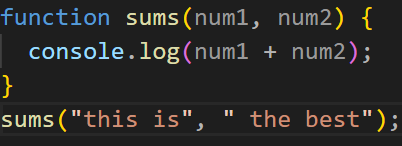
# Exercise 1

* Create a function that takes in two strings as parameters and logs them to the console.
* Share a screenshot when you’re done.
  + 1. A picture containing text, font, screenshot, line

       Description automatically generated

# Exercise 2

Create a function that takes in two strings as parameters, combines the two strings and logs the result to the console.



# Return statement

A return statement is the value that is given back by the function to whoever called it.

Let’s look at the previous example.

* In this example we saw the result of num1+ num2 in the console because we told the function to log that result to the console.

A picture containing text, font, screenshot

Description automatically generated

* If we decided to change the console.log to a return statement, now we can store the value of that result to be used later.

A picture containing text, font, screenshot

Description automatically generated

A picture containing text, font, screenshot

Description automatically generated

* + Changing our function to this produces no visual output in the console. That’s because we have nothing logged anything to the console.
  + We can think of the function call as being the value of the return statement.
  + To prove that, we can console.log the function call itself which should show us the result in the console.

A picture containing text, font, screenshot, line

Description automatically generated

# return statement continued…

* We can also store the function call into a variable that we can use later

1. A picture containing text, font, screenshot

   Description automatically generated

* We can also store the function itself into a variable that we can use later

1. A picture containing text, screenshot, font

   Description automatically generated
2. This is an anonymous function

# return statement continued…

1. Note: the return statement acts like a “break” for that function. So any code after that return statement won't be read.
2. A picture containing text, screenshot, font, line

   Description automatically generated

* In this example, the console log comes after the return statement and would not be executed.

# Exercise 3

Create a function that takes in a string and returns ***false*** if the string is ***empty*** and ***true*** if it is not.

(Hint: an empty string always === false).

1. After the result is returned, console.log the result to the browser.

A screen shot of a computer code

Description automatically generated with low confidence

# Functions in Objects

* A JavaScript function can be a value in an object.
* When functions are stored as object properties, they are called methods.

A computer code on a black background

Description automatically generated with low confidence

* You access an object method with the following syntax:

A picture containing text, font, screenshot, graphics

Description automatically generated

# Exercise 4

1. Write a function called half Number that will take one argument (a number), divide it by 2, and return the result.
   * Assign the return value of the function to a variable called halved Number.
   * Print out a log like "Half of 5 is 2.5.“ using the variable.
2. Write a function called time in Seconds that takes in an integer *minutes* as a parameter and returns seconds.

*[Hint: You might need to convert the string output from the prompt to an integer]*

* + Prompt the user to enter minutes.
  + Call the function and alert the output.

