WEEK 3 ARRAYS AND FUNCTIONS

RESEARCH ASSIGNMENT

Student: Aymar Dossou

Prompts

1. Select **five methods** that can be used on an Array and describe the following for each:

1) what the method signature is

2) what the method does

3) why would this method be useful (how could you use it)?

2. What is the difference between == and ===?

3. What is a closure and how does it work? Provide an example.

4. What is your favorite thing you learned this week?

Instructions

As developers, research is a constant part of our job. A common saying is that 90% of software development is Googling, and while that is an exaggeration, Google is a highly used tool in the role. This Research assignment is meant to go beyond the course curriculum and increase your understanding of relevant topics while exposing you to online resources you'll frequently use on the job. Please write a paragraph for **two (2)** of the above prompts and include URLs from where you found the information to cite your sources. Do not copy and paste text from the internet or any other source; use the information you find in your research, summarize, in your own words, the concepts. Plagiarism will result in a zero for the assignment as well as disciplinary actions.

Student responses

1. Five methods that can be used on an Array
    - push - length – splice – join – unshift

1) Methods signatures:

push:	`array.push(item1, item2, ..., itemX)`
length:	`array.length = number`
splice:	`array.splice(index, howmany, item1, ....., itemX)`
join:	`array.join(separator)`
unshift:	`array.unshift(item1, item2, ..., itemX)`

2) What the method does:

push:	*Adds new elements to the end of an array and returns the new length.*
length:	*Sets or returns the number of elements in an array.*
splice:	*Adds/Removes elements from an array.*
join:	*Joins all elements of an array into a string.*
unshift:	*Adds new elements to the beginning of an array, and returns the new length*

3) Why would this method be useful (how could you use it)?

Any methos used on an array has in most case a descriptive nomination which allows the user to adapt different methods to their arrays depending on the need they have at that moment in time. In cases in which a method's nomination is not descriptive of the action it performs; the user can then refer to the use executed by a previous user. For example:
The length method sets or returns the number of elements in an array as described above, which suggest that this method can be used to determine how many elements are in an array. In the other end, the Unshift method is not explicit, so in this case the user will refer to a previous user's execution of this method or to the description of the method itself.

2. What is the difference between == and ===?

The operators == and === are used to compare two type of data or variables. The == operator is used to compare or equate two numerical variables as opposed to the === operator which used to compare different type of variables (strings, boolean) including numerical variables. According to https://www.scaler.com/topics/javascript/difference-between-double-equals-and-triple-equals-in-javascript/ " The main difference between the == and === operator in JavaScript is that the == operator does the type conversion of the operands before comparison, whereas the === operator compares the values as well as the data types of the operands." This explanation point to the fundamental use of these two operators.

3. What is a closure and how does it work? Provide an example.

   A closure is a combination of function with one within the other. A clearer explanation as well as an illustration is given on https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures " A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created, at function creation time." See the following example

```
function init() {
  var name = 'Mozilla'; // name is a local variable created by init
  function displayName() {
    // displayName() is the inner function, a closure
    console.log(name); // use variable declared in the parent function
  }
  displayName();
}
init();
```

4. What is your favorite thing you learned this week?

   The favorite thing I learned this week is inserting a new element at the beginning of an array by using array.unshift()