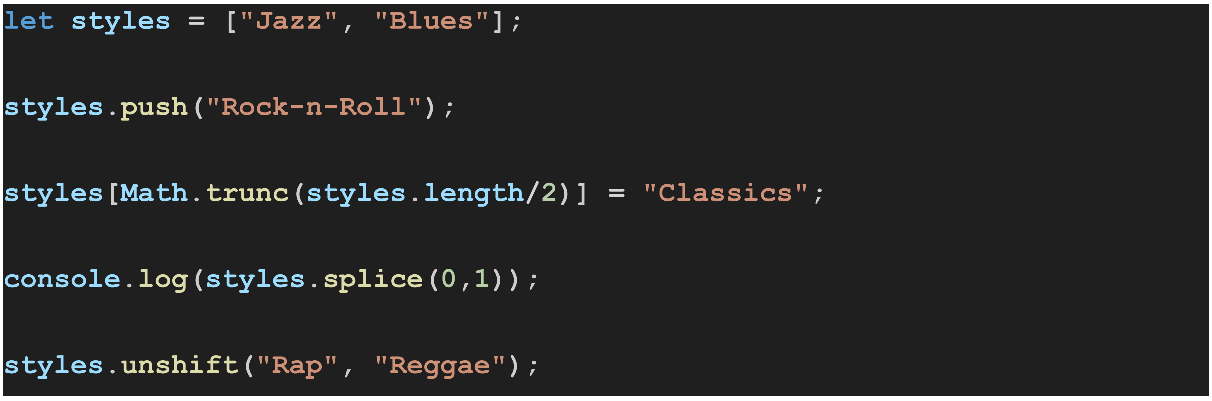
Alex Samos Muñoz

1. What is this code going to show?

let fruits = ["Apples", "Pear", "Orange"];
// push a new value into the "copy"
let shoppingCart = fruits;
shoppingCart.push("Banana");
// what's in fruits?
alert(fruits.length); // ?

El resultado del alert será 4

2. Let's try 5 array operations.
a) Create an array styles with items "Jazz" and "Blues".
b) Append "Rock-n-Roll" to the end.
c) Replace the value in the middle with "Classics". Your code for finding the middle value should work for any arrays with odd length.
d) Strip off the first value of the array and show it.
e) Prepend Rap and Reggae to the array.



```
let styles = ["Jazz", "Blues"];

styles.push("Rock-n-Roll");

styles[Math.trunc(styles.length/2)] = "Classics";

console.log(styles.splice(0,1));

styles.unshift("Rap", "Reggae");
```

3. What is the result? Why?

let arr = ["a", "b"];
arr.push(function() {
alert(this); });

arr[2](); // ?
El resultado de sacar la tercera posición del array será a, b, function(){ alert(this);}. Porque con el push le añadimos la función entera al array y al sacar su posición la llamamos y al tener un alert(this) muestra todo lo que hay dentro del array.

4. Write the function sumInput() that:
• Asks the user for values using prompt and stores the values in the array.
• Finishes asking when the user enters a non-numeric value, an empty string, or presses "Cancel".
• Calculates and returns the sum of array items.
P.S. A zero 0 is a valid number, please don't stop the input on zero

```
function sumInput() {
  let suma = 0;
  let bool = true;
  let arr = Array();
  while (bool) {
    let valorIntroducido = prompt("Introduce un numero: ");
    if (isNaN(valorIntroducido) || valorIntroducido === null ||
valorIntroducido === "") {
      bool = false;
    }else{
      let conversion = parseInt(valorIntroducido);
      arr.push(conversion);
    }
  }

  for(let num of arr){
    suma += num;
  }
  return suma;
}

let resultado = sumInput();

console.log(resultado);
```

5. The input is an array of numbers arr = [1, -2, 3, 4, -9, 6] and the task is to find the contiguous subarray of arr with the maximal sum of items. Write the function getMaxSubSum(arr) that will return that sum. Please try to think of a fast solution: O(n^2) or even O(n) if you can

getMaxSubSum([-1, 2, 3, -9]) == 5
getMaxSubSum([2, -1, 2, 3, -9]) == 6
getMaxSubSum([-1, 2, 3, -9, 11]) == 11
getMaxSubSum([-2, -1, 1, 2]) == 3
getMaxSubSum([100, -9, 2, -3, 5]) == 100
getMaxSubSum([1, 2, 3]) == 6 (take all)

Alex Samos Muñoz

```
function getMaxSubSum(arr) {
  let sumaActual = arr[0];
  let mayorSuma = 0;

  for (let i = 0; i < arr.length; i++) {
      sumaActual = (arr[i] > sumaActual + arr[i]) ? arr[i] : sumaActual
+ arr[i];
      mayorSuma = (sumaActual > mayorSuma) ? sumaActual : mayorSuma;
    }
  return mayorSuma;
}
```