

POO2 - Práctica 1: JavaScript

1. Escribir una función que calcule el cuadrado de un numero sin utilizar llaves ni paréntesis.
2. Escribir una función **creaResta** que reciba por parámetro un entero **x** y devuelva otra función que recibe como parámetro un entero **y**. Esta segunda función devuelve la resta:

$$y - x$$

Crear una variable llamada **resta** a partir de la función **creaResta** enviando como parámetro un numero entero **n**.

Ejecutar la función **resta** enviándole un numero entero **m**.

¿**x** toma el valor de **n** e **y** toma el valor de **m**, o es al revés?

3. Escribir una función **repeat** que reciba como parámetros una función Lambda **functionToRepeat** y un entero **n**.

repeat ejecutará **n** veces **functionToRepeat** pasándole por parámetro el número de ejecución (la primera vez será 1, la segunda 2, hasta **n**) y guardando los resultados en un array que finalmente lo retorna.

Escribir otra función **PrintString** que reciba por parámetro un entero **n** y devuelva "Execution number: n".

Ejemplo de ejecución con **n = 4**:

```
[ "Execution number: 1",  
  "Execution number: 2",  
  "Execution number: 3",  
  "Execution number: 4"]
```

4. Se tiene el siguiente arreglo que contiene un listado de pilotos de F1 ordenados según la posición de llegada de la última carrera.

```
const pilotos = ["Verstappen", "Hamilton", "Russell", "Sainz", "Perez", "Leclerc", "Norris",  
                 "Alonso", "Ocon", "Vettel"];
```

a. Obtener la posición de llegada de Russell.

b. ¿Quién llegó sexto?

c. Obtener un listado con todos los pilotos que contengan la letra "a".

Resultado esperado: ["Verstappen", "Hamilton", "Sainz", "Alonso"]

d. A partir del siguiente arreglo ["Russell", "Bottas", "Perez"], obtener otro arreglo con true, si el piloto se encuentra en el listado original, o false, si no está.

Resultado esperado: [true, false, true]

e. Los organizadores de la carrera se equivocaron, Perez no llegó en quinto puesto sino en segundo. Corregir el listado.

5. Escribir una función **howManyTimesAppears** que reciba como parámetro un arreglo de números y un número. Esta función devolverá la cantidad de veces que aparece el número (recibido por parámetro) en el arreglo. En caso de no aparecer, la función debe devolver "0" (cero).

Ejemplos:

```
Const array = [ 3, 6, 9, 3, 1, 5, 2, 10]
howManyTimesAppears(array, 3)    -->    Devuelve: 2
howManyTimesAppears(array, 5)    -->    Devuelve: 1
howManyTimesAppears(array, 7)    -->    Devuelve: 0
```

6. Se tiene un arreglo numérico y se debe sumar a cada elemento del mismo, el valor del número más chico dentro del arreglo.

Ejemplos:

```
Array1 = [ 4, 8, 2, 13, 20]    -->    Resultado esperado: [ 6, 10, 4, 15, 22]
Array2 = [ 4, 8, 2, -5, 20]    -->    Resultado esperado: [ -1, 3, -3, -10, 15]
```

7. Teniendo una lista de strings con "Nombre Apellido".
- 1) Ordenar la lista alfabéticamente por nombre.
 - 2) Invertir los strings de "Nombre Apellido" a "Apellido Nombre".
 - 3) Ordenar la nueva lista alfabéticamente por apellido.

Aclaración: No tener en cuenta nombres ni apellidos compuestos.

```
const personas = ["Lionel Messi", "Rodrigo Depaul", "Emiliano Martinez", "Angel Dimaria",
"Soledad Jaimes", "Yamila Rodriguez", "Florencia Bonsegundo"];
```