

## Ejercicios básicos de JavaScript

### [C.01] Mensajes

Realizar una aplicación que muestre los siguientes mensajes utilizando **`console.log()`**:

**HOLA MUNDO!!!**

**Puedo mostrar comillas 'simples'.**

**Y comillas "dobles".**

**Nota:** El mensaje se mostrará en una sola instrucción. Utilice caracteres de escape. Emplee plantillas de string (tilde invertido).

### [C.02] Mostrando impares

Crear un programa que use un bucle **for** para mostrar los números del 1 al 20, pero que salte los números pares (usando *continue*) y se detenga cuando alcance el número 15 (usando *break*).

**Nota:** Utilizar `console.log()`

### [C.03] ¿Cuál es el número?

Cree una aplicación que muestre, a través de un Array, los nombres de los meses de un año y el número al que ese mes corresponde. Utilizar una estructura repetitiva para escribir en la consola.

**Nota:** Utilizar `console.log()`

### [C.04] ¿Es par o impar?

Realizar una función que reciba un número y que muestre (por consola) un mensaje como el siguiente:

**El número 5 es impar**, siendo 5 el número recibido como parámetro.

### [C.05] Repetite la palabrita

Realizar una función que reciba un parámetro requerido de tipo numérico y otro opcional de tipo cadena. Si el segundo parámetro es recibido, se mostrará tantas veces por consola, como lo indique el primer parámetro. En caso de no recibir el segundo parámetro, se mostrará el valor inverso del primer parámetro.

**Nota:** Utilizar `console.log()`

### [C.06] Mostrame el nombre y apellido

Guardar su nombre y apellido en dos variables distintas. Dichas variables serán pasadas como parámetro de la función **mostrarNombreApellido**, que mostrará el apellido en mayúscula y el nombre solo con la primera letra en mayúsculas y el resto en minúsculas.

El apellido y el nombre se mostrarán separados por una coma (,).

**Nota:** Utilizar `console.log()`

### [C.07] Trunca la cadenita

Crear una función **truncate(cadena, longitud)** que verifique la longitud de **cadena** y, si excede **longitud**, reemplaza el final de *cadena* con el carácter de puntos suspensivos "...", para hacer su longitud igual al parámetro longitud.

El resultado de la función debe ser la cadena truncada (si es necesario).

### [C.08] Arrays con rest

Crea una función llamada **crearArray** que reciba **n** números como argumentos usando el parámetro rest (...numeros) y devuelva un array.

Ejemplo:

```
crearArray(1, 2, 3, 4); // [1, 2, 3, 4]
```

### [C.09] Arrays con spread

Usando el operador spread (...), crea una copia de un array y agrega un nuevo elemento al final.

Ejemplo:

```
// Original: const frutas = ["manzana", "banana"];
```

```
// Resultado: ["manzana", "banana", "naranja"]
```

### [C.10] Música

Definir un array nombrado **estilos**, con los ítems **"Jazz"** y **"Blues"**. Se pide:

1. Agregar **"Rock-n-Roll"** al final.
2. Reemplazar el valor del medio por **"Heavy Metal"**. El código para encontrar el valor medio debe funcionar con cualquier array de longitud impar.
3. Quitar el primer valor del array.
4. Anteponer Rap y Reggae al array.

Mostrar el array durante el proceso, debería quedar así:

**Jazz, Blues**

**Jazz, Blues, Rock-n-Roll**

**Jazz, Heavy Metal, Rock-n-Roll**

**Heavy Metal, Rock-n-Roll**

**Rap, Reggae, Heavy Metal, Rock-n-Roll**

**Nota:** Utilizar console.log()

### [C.11] ¿Son todas mayúsculas, minúsculas o qué?

Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

### [C.12] Palíndro qué?

Definir una función que determine si la cadena de texto que se le pasa como parámetro es un **palíndromo**, es decir, si se lee de la misma forma desde la izquierda y desde la derecha.

Ejemplo de palíndromo complejo: "**La ruta nos aporoto otro paso natural**".

### [C.13] Se me prendió fuego el cerebro

Definir la función **obtenerSumaMaxima(arr)**. Recibirá como parámetro un array de números, por ejemplo: [1, -2, 3, 4, -9, 6].

La tarea es encontrar, dentro de 'arr', el subarray de elementos contiguos que tenga la suma máxima.

Por ejemplo:

```
obtenerSumaMaxima([-1, 2, 3, -9]) //5
obtenerSumaMaxima([2, -1, 2, 3, -9]) //6
obtenerSumaMaxima([-1, 2, 3, -9, 11]) //11
obtenerSumaMaxima([-2, -1, 1, 2]) //3
obtenerSumaMaxima([100, -9, 2, -3, 5]) //100
```

```
obtenerSumaMaxima([1, 2, 3]) //6
```

**Nota:** si todos los valores son negativos, no se toma ningún elemento y su suma es cero.

```
obtenerSumaMaxima([-1, -2, -3]) // 0
```

### [C.14] ¿De qué signo sos?

Crear una función que reciba como único parámetro una cadena que contenga el día, mes y año de nacimiento de una persona (con formato **dd-mm-yyyy**). La función mostrará por consola a qué signo corresponde dicha fecha de nacimiento.

### [C.15] Se me prendió fuego el cerebro... otra vez

Un **número de Smith** es un número entero tal que la suma de sus dígitos es igual a la suma de los dígitos de los números restantes tras la factorización en primos (la factorización debe estar escrita sin exponentes, repitiendo los números todas las veces necesarias).

Por ejemplo, **378** =  $2 \times 3 \times 3 \times 3 \times 7$  es un número de Smith en base 10, porque  $3 + 7 + 8 = 2 + 3 + 3 + 3 + 7$ .

Por definición, se deben contar los dígitos de los factores.

Por ejemplo, **22** en base 10 es  $2 \times 11$ , y se deben contar los tres dígitos: 2, 1, 1. Por lo tanto 22 es un número de Smith porque  $2 + 2 = 2 + 1 + 1$

**Nota:** Utilice tres funciones, una realiza la comparación, otra descompone el número en sus factores primos y suma los coeficientes, y la última función suma cada término.