



## EJERCICIOS CONDICIONALES

- 1. Pide al usuario un número y muestra si es par o impar.**
- 2. Pide al usuario dos números, súmalos y muestra si la suma es par o impar.**
- 3. Pide al usuario un número y muestra si es primo o no (un número primo es un número mayor que 1 que tiene únicamente dos divisores positivos distintos: él mismo y el 1. Por ejemplo, el 7 sólo tiene como divisores el 7 y el 1, por tanto es primo. Mientras que el 6 tiene como divisores el 6, el 1, el 2 y el 3).**
- 4. Pide al usuario dos números y muestra cuál es el mayor.**
- 5. Pide al usuario una letra y muestra si es vocal o no.**
- 6. Pide al usuario un número entre 1 y 7 y muestra a qué día de la semana corresponde.**



## EJERCICIOS CONDICIONALES

**7. Escribe un programa que pida al usuario su nombre y su edad y muestre en pantalla si es mayor de edad (18 años o más) y si su nombre empieza por una letra anterior a la "M" en el abecedario (pista: piensa en acceder a la primera letra del nombre considerando que las palabras (o strings) son un array de caracteres).**

**8. Escribe un programa que solicite al usuario ingresar dos números y determine cuál es el mayor. Si ambos números son iguales, mostrará un mensaje que diga "Ambos números son iguales".**

**9. Escribe un programa que solicite al usuario ingresar dos números y determine si el primer número es divisible entre el segundo número. Si el primer número es divisible entre el segundo número, mostrará un mensaje que diga "El primer número es divisible entre el segundo número". De lo contrario, mostrará un mensaje que diga "El primer número no es divisible entre el segundo número".**



## EJERCICIOS CONDICIONALES

**10. Escribe un programa que solicite al usuario ingresar tres números y determine cuál es el mayor. Si los tres números son iguales, mostrará un mensaje que diga "Los tres números son iguales".**



## EJERCICIOS BUCLES

- 1. Imprime los primeros n números naturales utilizando un bucle while, donde n será un número que introducirá el usuario. Por ejemplo, si el usuario introduce 5, el programa debe mostrar 1, 2, 3, 4 y 5.**
- 2. Imprime la tabla de multiplicar del número introducido utilizando un bucle for.**
- 3. Calcula la suma de los primeros n números pares utilizando un bucle while. Por ejemplo, si el usuario introduce 6, el programa calcular la suma de  $2 + 4 + 6$ , y mostrar el resultado.**
- 4. Calcula el factorial del número introducido utilizando un bucle for**
- 5. Dado un número n, imprime por la consola todos los números desde 1 hasta n que sean divisibles por 3 o 5**
- 6. Dado un número n, imprimir en la consola todos los números desde n hasta 1 en orden descendente**
- 7. Dado un número n, imprimir en la consola la suma de todos los números desde 1 hasta n que sean múltiplos de 3 y 5 (es decir, que al dividirlos por 3 o por 5, el residuo de la división sea 0).**



## EJERCICIOS BUCLES

- 8. Pide al usuario un número y muestra por consola todos los números primos menores o iguales a ese número.**
- 9. Pide al usuario una palabra y mostrar por consola todas las letras de la palabra.**
- 10. Muestra los primeros n términos de la serie de Fibonacci utilizando un bucle while. La serie de fibonacci es una secuencia de números donde cada número es igual a la suma de sus dos anteriores. Por ejemplo, la secuencia en caso de que n fuera 5 seria: 0, 1, 1, 2, 3, 5**



## EJERCICIOS ARRAYS

- 1. Crea un array con los números del 1 al 10. Luego, utiliza un bucle for para sumar todos los números del array y mostrar el resultado por consola.**
- 2. Crea un array con 5 nombres. Luego, utiliza un bucle while para mostrar cada uno de los nombres por consola.**
- 3. Crea un array con 3 números. Luego, utiliza un bucle for para mostrar por consola los cuadrados de cada uno de los números. (Investiga cómo hacer el cuadrado de un número en javascript).**
- 4. Crea un array con los números del 1 al 5 y otro array con los números del 6 al 10. Luego, utiliza un bucle for para mostrar por consola la suma de cada par de elementos correspondientes de los dos arrays.**
- 5. Crea un array con 5 números y muestra por consola el número mayor.**
- 6. Crea un array con 10 palabras y muestra por consola la cantidad de palabras que empiezan con la letra "a".**



## EJERCICIOS ARRAYS

- 7. Crea un array con 7 números y muestra por consola la suma de los números en posiciones pares (0, 2, 4, etc.).**
- 8. Crea un array con 6 números y muestra por consola los números que son pares.**
- 9. Crea un array con 4 números y muestra por consola la diferencia entre el mayor y el menor número.**
- 10. Encuentra el índice del primer elemento par en un array. (pista: Investiga sobre la palabra reservada "break" y cómo utilizarla a tu favor en los bucles).**



## EJERCICIOS FUNCIONES

- 1. Define una función llamada "sumar" que reciba dos parámetros y retorne la suma de ambos.**
- 2. Define una función llamada "restar" que reciba dos parámetros y retorne la resta del segundo al primero.**
- 3. Define una función llamada "multiplicar" que reciba dos parámetros y retorne el producto de ambos.**
- 4. Define una función llamada "dividir" que reciba dos parámetros y retorne la división del primero entre el segundo. Si el segundo parámetro es 0, la función debe retornar un mensaje de error indicando que la división entre 0 no está definida.**
- 5. Define una función llamada "calculadora" que reciba tres parámetros: num1, num2 y operacion. La función debe retornar el resultado de la operación entre num1 y num2, donde la operación es un string que puede ser "suma", "resta", "multiplicacion" o "division". Si la operación no es una de las cuatro opciones mencionadas, la función debe retornar un mensaje de error indicando que la operación introducida no es válida.**



## EJERCICIOS FUNCIONES

- 6. Define una función llamada "esPar" que reciba un parámetro de tipo número y retorne un valor booleano indicando si el número es par o no.**
- 7. Define una función llamada "esPrimo" que reciba un parámetro de tipo número y retorne un valor booleano indicando si el número es primo o no. Un número es primo si sólo es divisible entre 1 y sí mismo.**
- 8. Define una función llamada "factorial" que reciba un parámetro de tipo número y retorne el factorial de ese número. El factorial de un número n es el producto de todos los números enteros positivos desde 1 hasta n.**
- 9. Define una función llamada "palindromo" que reciba un parámetro que sea una palabra y retorne un valor booleano indicando si la palabra es un palíndromo o no. Un palíndromo es una palabra o frase que se lee igual de izquierda a derecha que de derecha a izquierda.**



## EJERCICIOS FUNCIONES

**10. Define una función llamada "contarVocales" que reciba una palabra y retorne la cantidad de vocales que tiene. Las vocales permitidas son "a", "e", "í", "o" y "u" (en minúsculas). No hace falta que controles si la palabra tiene alguna vocal en mayúscula, asume que las vocales de la palabra siempre estarán en minúscula**



## EJERCICIOS MATRICES

- 1. Crea una matriz cuadrada de tamaño n, que contenga todos ceros.**
- 2. Crea una matriz de 3x3. Luego, muestra por consola la suma de los valores de la diagonal principal de la matriz.**
- 3. Crea una matriz de tamaño n x n, donde n es un número introducido por el usuario, y muéstralala por pantalla. Luego, muestra por consola la suma de los valores de la diagonal de la matriz.**
- 4. Crea dos matrices de 2x2. Luego, muestra por consola la suma de ambas matrices.**
- 5. Crea una matriz de tamaño 4x4. Luego, muestra por consola la matriz resultante de multiplicar cada valor por 2.**
- 6. Crea una matriz de tamaño 5x5. Luego, muestra por consola la suma de todos los valores de la matriz.**
- 7. Crea una matriz de tamaño n x m, que contenga números en orden ascendente desde 1 hasta n\*m.**



## EJERCICIOS EXTRA (TEMÁTICA JUEGOS)

- 1. Adivina el número.** El programa creará un número aleatorio entre 1 y 100. El jugador tiene que adivinar el número. El programa le indica al jugador si el número es mayor o menor al que ha introducido. El juego termina cuando el jugador adivina el número.
- 2. Crea el piedra, papel, tijeras en javascript.** El usuario jugará contra el programa. Para que el programa pueda jugar, generará un número aleatorio. Si este número es menor que 3, la elección del programa será piedra. Si el número es menor o igual a 6 y no es menor a 3, la elección del programa será papel. En cualquier otro caso, la elección del programa será tijeras.

## EJERCICIOS EXTRA (TEMÁTICA JUEGOS)

**3. Crea el ahorcado.** El programa elige una palabra aleatoria de un conjunto predefinido de palabras. El usuario debe adivinar la palabra introduciendo una letra en cada intento. Si la letra está en la palabra, el programa muestra la letra en la palabra. Si la letra no está en la palabra, el programa agrega un intento fallido a un contador. Muestra la palabra oculta con guiones bajos. El usuario gana si adivina la palabra antes de que el contador de intentos fallidos llegue a un número predefinido (como 10). Por ejemplo, si la palabra es bitcoin, primero lo mostramos con \_\_\_\_\_. Si el usuario introduce la letra "b", mostraremos b\_\_\_\_\_. Si falla, restaremos un intento. Pista: investiga las funciones "repeat", y "includes" de los strings en javascript.