

Ejercicios JavaScript básico I

1. Ejercicio 1 JS: Tipos de datos

Tipos de datos primitivos:

1.1. Declara una variable de tipo string y asígnale un mensaje de saludo. Luego, imprime la variable en la consola.

1.2. Declara una variable de tipo number y asígnale tu edad. Luego, imprime la variable en la consola.

1.3. Declara una variable de tipo boolean y asígnale true o false según tu preferencia. Luego, imprime la variable en la consola.

1.4. Declara una variable de tipo null y otra de tipo undefined. Luego, imprime ambas variables en la consola.

Tipos de datos no primitivos (objetos):

1.5. Crea un objeto que represente un artículo con propiedades como name, price y category. Imprime el objeto en la consola.

1.6. Declara un array que contenga nombres de ciudades. Imprime el array en la consola.

Variables:

1.7. Declara una variable y asígnale tu color favorito. Luego, modifica la variable con otro color e imprime ambos valores en la consola.

Declaración de variables:

1.8. Declara dos variables, una sin asignarle ningún valor y otra con el valor null. Imprime ambas variables en la consola.

Alcance de las variables:

1.9. Crea dos variables, una dentro de una función y otra fuera de ella. Intenta acceder a ambas variables desde dentro y fuera de la función, y observa los resultados en la consola.

Reglas de nomenclatura de variables:

1.10. Declara variables con nombres que sigan las reglas de JavaScript y otras que no las sigan. Imprime los nombres de las variables en la consola.

Operadores:

1.11. Utiliza diferentes operadores (+, -, *, /, %, ++, --) con variables que contengan números. Imprime los resultados en la consola.

Strings:

1.12. Declara una variable con una cadena de texto de varias líneas usando caracteres de escape (\n, \t). Imprime la cadena resultante en la consola.

2. Ejercicio 2 JS: Estructuras de control

Condicionales (if, else if, else):

2.1. Escribe un programa que verifique si un número es positivo, negativo o cero e imprima el resultado.

2.2. Crea un programa que determine si una persona es menor de edad, adulta o anciana según su edad.

Bucles (for, while, do-while):

2.3. Utiliza un bucle for para imprimir los valores de i desde 0 hasta 4.

2.4. Utiliza un bucle while para imprimir los valores del contador desde 0 hasta 4.

2.5. Emplea un bucle do-while para imprimir los valores de x desde 0 hasta 4.

Interrupción de bucles (break, continue):

2.6. Crea un bucle for que se detenga cuando i sea igual a 3 usando la instrucción break.

2.7. Implementa un bucle for que salte la iteración cuando j sea igual a 2 usando la instrucción continue.

Condicional (switch):

2.8. Convierte un número del 1 al 12 en el nombre del mes correspondiente. El programa debe solicitar al usuario que ingrese un número del 1 al 12 y luego mostrar el nombre del mes asociado.

Ejemplo:

- Ingresar un número del 1 al 12 para saber el mes correspondiente: 9
- El mes correspondiente al número 9 es: Septiembre

3. Ejercicio 3 JS: Funciones

3.1. Escribe un programa que solicite al usuario el radio de un círculo y luego calcule y muestre el área y el perímetro del círculo.

- Solicita al usuario que ingrese el radio del círculo.
- Define dos funciones: una para calcular el área del círculo y otra para calcular el perímetro del círculo.
- Usa las fórmulas matemáticas apropiadas dentro de estas funciones para calcular el área ($\pi * \text{radio}^2$) y el perímetro ($2 * \pi * \text{radio}$) del círculo.
- Llama a estas funciones con el radio proporcionado por el usuario como argumento.
- Muestra los resultados del área y el perímetro al usuario.

3.2. Escribe una función llamada `calculatePower` que tome dos parámetros: `base` y `exponent`. Esta función debe calcular y devolver el resultado de elevar la base al exponente.

Ejemplo:

```
// Llamada a la función  
  
let result = calculatePower(2, 3);  
  
console.log(result); // Resultado esperado: 8
```

4. Ejercicio 4 JS: Arrays

4.1. Crea un programa que solicite al usuario ingresar una lista de números y luego encuentre el número más grande en esa lista.

- Solicita al usuario que ingrese la cantidad de números que quiere introducir.
- Solicita al usuario que introduzca cada número uno por uno.
- Almacena estos números en un array.
- Escribe una función llamada `findLargestNumber` que tome un array de números como entrada y devuelva el número más grande en el array.
- Dentro de la función, itera a través del array usando un bucle `for` para encontrar el número más grande.
- Muestra el número más grande encontrado al usuario.

Asegúrate de que el programa utilice arrays para almacenar los números y la función para encontrar el número más grande.

Ejemplo:

- Ingresas la cantidad de números: 5
- Ingresas el número 1: 10
- Ingresas el número 2: 25
- Ingresas el número 3: 7
- Ingresas el número 4: 18
- Ingresas el número 5: 12
- El número más grande es: 25

5. Ejercicio 5 JS: Bucle for

5.1. Bucle for

- Imprimir los números del 20 al 30 en la consola.
- Imprimir los números pares del 30 al 50 en la consola.
- Calcular la suma de los primeros 50 números naturales.
- Imprimir la tabla de multiplicar del 8.
- Imprimir los elementos de un array.
- Imprimir un patrón de asteriscos en forma de triángulo, con altura 9.
- Calcular la suma de los números pares del 1 al 50.
- Imprimir los números del 30 al 20 en orden descendente.
- Calcular el promedio de un array de números.