

Guía para Estudiantes: Lógica de Programación con JavaScript

Objetivo

Aprender los fundamentos de la **lógica de programación** a través de **JavaScript**, un lenguaje de programación popular y accesible. Vamos a abordar los siguientes temas:

1. **Introducción a JavaScript**
 2. **Variables, tipos de datos y operadores**
-

1. Introducción a JavaScript

JavaScript es un **lenguaje de programación** utilizado principalmente para crear interactividad en las páginas web. Es un lenguaje **interpretado** y **dinámico**, lo que significa que el código se ejecuta directamente en el navegador sin necesidad de compilarlo. Aunque inicialmente se utilizaba solo para scripts en el lado del cliente, hoy en día también se usa en el lado del servidor con **Node.js**.

Características de JavaScript:

- **Versatilidad:** Se puede usar tanto para desarrollo web como para aplicaciones móviles, juegos y servidores.
- **Lenguaje de alto nivel:** Esto significa que su sintaxis está más cerca del lenguaje humano, lo que facilita el aprendizaje.
- **Lenguaje interpretado:** El código es ejecutado directamente por el navegador sin necesidad de compilación previa.

2. Variables, Tipos de Datos y Operadores

En programación, las **variables** son espacios de memoria donde podemos almacenar valores. Los valores que almacenamos pueden ser de diferentes tipos, como números, cadenas de texto, etc. A continuación, veremos cómo trabajar con variables, tipos de datos y operadores en JavaScript.


2.1 Variables en JavaScript

Las **variables** en JavaScript se usan para almacenar datos. Para declarar una variable, se utilizan las palabras clave `var`, `let` o `const`:

- `var`: Es la forma más antigua de declarar variables. Ya no se recomienda en la mayoría de los casos debido a problemas de alcance.
- `let`: Es la manera más moderna y flexible de declarar una variable. Tiene **alcance de bloque**, lo que significa que solo existe dentro de la función o bloque de código donde se define.
- `const`: Similar a `let`, pero para declarar **constantes**, es decir, valores que no deben cambiar durante la ejecución del programa.

Ejemplo de declaración de variables:

javascript

 Copiar código

```
let edad = 25;      // Variable que puede cambiar
const nombre = "Juan"; // Variable constante
```

2.2 Tipos de Datos

En JavaScript, los datos que almacenamos en las variables pueden ser de distintos **tipos**. Los principales son:

- **Números (Number)**: Representa valores numéricos, ya sean enteros o decimales.


javascript

 Copiar código

```
let edad = 25;      // Entero
let precio = 19.99; // Decimal
```

- **Cadenas de texto (String):** Se utilizan para almacenar texto. Se definen entre comillas simples (') o dobles (").

javascript

 Copiar código

```
let nombre = "Carlos";  
let saludo = 'Hola, ¿cómo estás?';
```

- **Booleanos (Boolean):** Solo puede ser `true` o `false`. Se usan para decisiones lógicas.

javascript

 Copiar código

```
let esEstudiante = true;  
let tieneLicencia = false;
```

- **Arreglos (Array):** Almacena múltiples valores en una sola variable.

javascript

 Copiar código

```
let numeros = [1, 2, 3, 4, 5];  
let colores = ["rojo", "verde", "azul"];
```

- **Objetos (Object):** Son colecciones de pares clave-valor. Se usan para representar estructuras más complejas.

javascript

 Copiar código

```
let persona = {  
  nombre: "Ana",  
  edad: 30,  
  ciudad: "Madrid"  
};
```


2.3 Operadores

Los **operadores** son símbolos que se utilizan para realizar operaciones sobre los valores o variables. En JavaScript, existen varios tipos de operadores:

1. **Operadores Aritméticos:** Se usan para realizar operaciones matemáticas.
 - + (Suma)
 - - (Resta)
 - * (Multiplicación)
 - / (División)
 - % (Módulo, devuelve el residuo de una división)

Ejemplo:

javascript

 Copiar código

```
let a = 10;  
let b = 5;  
let resultado = a + b; // 15  
let residuo = a % b;   // 0
```

2. **Operadores de Comparación:** Compara dos valores y devuelve un valor booleano (true o false).
 - == (Igual a)
 - != (Distinto de)
 - > (Mayor que)

- < (Menor que)
- >= (Mayor o igual que)
- <= (Menor o igual que)

Ejemplo:

javascript

 Copiar código


```
let x = 10;  
let y = 20;  
console.log(x == y); // false  
console.log(x < y); // true
```

3. Operadores Lógicos: Se usan para combinar expresiones booleanas.

- && (Y lógico)
- || (O lógico)
- ! (No lógico)

Ejemplo:

javascript

 Copiar código

```
let esMayor = true;  
let tieneLicencia = false;  
console.log(esMayor && tieneLicencia); // false  
console.log(esMayor || tieneLicencia); // true
```

4. **Operadores de Asignación:** Permiten asignar valores a las variables.
- = (Asignación simple)
 - +=, -=, *=, /= (Operadores de asignación combinados)

Ejemplo:

javascript

 Copiar código

```
let saldo = 100;  
saldo += 50; // saldo = saldo + 50  
console.log(saldo); // 150
```

Resumen

- **JavaScript** es un lenguaje fundamental para el desarrollo web y tiene una sintaxis sencilla.
- Las **variables** son fundamentales para almacenar datos, y se pueden declarar con `let` y `const`.
- Los **tipos de datos** más comunes en JavaScript son números, cadenas de texto, booleanos, arreglos y objetos.
- Los **operadores** en JavaScript permiten hacer operaciones matemáticas, comparaciones y lógica entre valores.

Ejemplo Explicativo en JavaScript

Imaginemos que vamos a crear un programa que calcule el **perímetro** y el **área** de un rectángulo, utilizando las fórmulas matemáticas correspondientes. Posteriormente, mostraremos los resultados de estos cálculos en la consola.

Enunciado:

Vamos a pedir al usuario que ingrese las dimensiones de un rectángulo (largo y ancho), luego calcularemos su **perímetro** y su **área**, y mostraremos los resultados en la consola.

Fórmulas utilizadas:

- **Área del rectángulo:** $A'rea = largo \times ancho$
 $\text{\texttt{\text{Área}}} = \text{\texttt{\text{largo}}} \times \text{\texttt{\text{ancho}}}$
- **Perímetro del rectángulo:** $Peri'metro = 2 \times (largo + ancho)$
 $\text{\texttt{\text{Perímetro}}} = 2 \times (\text{\texttt{\text{largo}}} + \text{\texttt{\text{ancho}}})$

Código en JavaScript:

```
// Declaración de variables para las dimensiones del rectángulo
let largo = 50;      // Tipo de dato: Number (número entero)
let ancho = 20;      // Tipo de dato: Number (número entero)

// Cálculo del área del rectángulo
let area = largo * ancho; // Operador de multiplicación (*)

// Cálculo del perímetro del rectángulo
let perimetro = 2 * (largo + ancho); // Operadores de suma (+) y
multiplicación (*)

// Mostramos los resultados en la consola
console.log("El área del rectángulo es: " + area); // Muestra el área
console.log("El perímetro del rectángulo es: " + perimetro); // Muestra el
perímetro

// Cambiar las dimensiones del rectángulo
largo = 30;          // Asignación: cambiamos el valor de largo
ancho = 40;          // Asignación: cambiamos el valor de ancho
```

```
// Recalculamos el área y el perímetro con las nuevas dimensiones
area = largo * ancho; // Nueva operación aritmética para el área
perimetro = 2 * (largo + ancho); // Nueva operación aritmética para el
perímetro

// Mostramos los nuevos resultados
console.log("El área del rectángulo con nuevas dimensiones es: " + area);
console.log("El perímetro del rectángulo con nuevas dimensiones es: " +
perimetro);
```

Explicación del Código:

1. **Declaración de variables:**
 - largo y ancho son variables de tipo **número** (Number), que representan las dimensiones del rectángulo.
 - Inicialmente, largo tiene un valor de 50 y ancho tiene un valor de 20.
 2. **Cálculo del área:**
 - El área se calcula usando el operador de **multiplicación** (*), y se almacena en la variable area.
 3. **Cálculo del perímetro:**
 - El perímetro se calcula con la fórmula $2 \times (\text{largo} + \text{ancho})$. Utilizamos los operadores de **suma** (+) y **multiplicación** (*).
 4. **Mostrar resultados:**
 - Usamos console.log() para imprimir los resultados de **área** y **perímetro** en la consola.
 5. **Cambio de dimensiones:**
 - Modificamos las dimensiones del rectángulo, asignando nuevos valores a largo y ancho.
 6. **Recalcular con las nuevas dimensiones:**
 - Realizamos de nuevo el cálculo de **área** y **perímetro** con las nuevas dimensiones, y mostramos los resultados en la consola.
-

Conceptos Abordados:

1. **Variables:** Se utilizaron variables para almacenar las dimensiones (largo, ancho), el **área** y el **perímetro**.



2. **Tipos de datos:** Las variables `largo`, `ancho`, `area` y `perimetro` son de tipo **número** (`Number`), ya que los valores que se almacenan son valores numéricos.
3. **Operadores:**
 - **Aritméticos:** Se usaron operadores de **suma** (+) y **multiplicación** (*) para calcular el área y el perímetro.
 - **Asignación:** Se usó el operador de asignación (=) para cambiar el valor de las variables.

Conclusión:

Este ejercicio demuestra cómo utilizar **variables**, **tipos de datos** y **operadores** para realizar cálculos matemáticos en JavaScript sin necesidad de condicionales. Es un ejercicio que se centra en las operaciones básicas y en cómo manipular datos dentro de un programa.

¡Claro! Aquí tienes 10 enunciados de ejercicios para resolver en JavaScript sin utilizar condicionales. Estos ejercicios están diseñados para trabajar con **variables**, **tipos de datos** y **operadores**.

Ejercicios para Resolver en JavaScript:

- Suma de tres números:**
 - Declara tres variables con valores numéricos. Suma los tres números y muestra el resultado en la consola.
- Multiplicación de dos decimales:**
 - Declara dos variables de tipo number con valores decimales (por ejemplo, 5.5 y 2.3). Multiplica ambos valores y muestra el resultado en la consola.
- Promedio de cuatro notas:**
 - Declara cuatro variables con valores numéricos que representen notas. Calcula el promedio de esas notas y muestra el resultado en la consola.
- Área de un triángulo:**
 - Declara dos variables base y altura, y usa la fórmula del área de un triángulo ($\text{área} = (\text{base} \times \text{altura}) / 2$) para calcular el área del triángulo. Muestra el resultado en la consola.
- Raíz cuadrada de un número:**
 - Declara una variable con un número positivo. Usa la función `Math.sqrt()` para calcular la raíz cuadrada del número y muestra el resultado en la consola.
- Conversión de grados Celsius a Fahrenheit:**
 - Declara una variable con un valor en grados Celsius. Convierte ese valor a grados Fahrenheit usando la fórmula $F = (C \times (9/5)) + 32$ y muestra el resultado en la consola.
- Potencia de un número:**
 - Declara dos variables, una con el valor de la base y otra con el valor del exponente. Usa el operador de **potenciación** (`**`) para calcular la base elevada al exponente y muestra el resultado.
- Concatenación de nombres:**
 - Declara dos variables de tipo string, una con el nombre de una persona y otra con su apellido. Une estas dos cadenas de texto y muestra el resultado completo en la consola.
- Calcular el perímetro de un círculo:**
 - Declara una variable con el valor del radio de un círculo. Calcula el perímetro utilizando la fórmula $P = 2 \times \pi \times \text{radio}$, y muestra el resultado en la consola.
- Restar dos números grandes:**



- Declara dos variables con números grandes (por ejemplo, 50000 y 12000). Resta el segundo número del primero y muestra el resultado en la consola.

Consejos para Resolver los Ejercicios:

- Utiliza **let** para declarar las variables y asigna valores adecuados según el tipo de dato que se te indique en el enunciado.
- Asegúrate de usar los operadores correctos para las operaciones matemáticas: + para suma, - para resta, * para multiplicación, / para división y **** para potenciación.
- Para algunas operaciones, como el cálculo de la raíz cuadrada o la conversión de Celsius a Fahrenheit, puedes usar funciones predefinidas de JavaScript como Math.sqrt().
- Revisa si la fórmula que estás utilizando es correcta y asegúrate de mostrar los resultados usando **console.log()**.