

JAVA SCRIPT

```
// **GUÍA DE JAVASCRIPT: VARIABLES, OPERADORES Y ESTRUCTURAS**

// =====
// 1. Uso de Variables
// =====

// 1.1. `var`
// - Es la forma más antigua de declarar variables.
// - Tiene un alcance de función.
// - Si se usa una variable con `var` antes de estar definida, no marca
// error, pero su valor es `undefined`.

var nombre = 'EDER VEGA';
console.log(nombre); // "EDER VEGA"
console.log(typeof(nombre)); // "string"

var edad = 23;
console.log(edad); // 23
console.log(typeof(edad)); // "number"

edad = 'cinco'; // Cambio de tipo de dato
console.log(edad); // "cinco"
console.log(typeof(edad)); // "string"

var sueldo = 10;
console.log(sueldo); // 10
console.log(typeof(sueldo)); // "number"

var tinetrabajo = false;
console.log(tinetrabajo); // false
console.log(typeof(tinetrabajo)); // "boolean"

var puesto;
console.log(typeof(puesto)); // "undefined" (solo declarada)
```

JAVA SCRIPT

```
// -----  
// 1.2. `let`  
// - Introducido en ES6 (ECMAScript 2015).  
// - Tiene un alcance de bloque.  
// - No se puede usar antes de declararla (genera un error).  
  
let nombreLet = "Ana";  
console.log(nombreLet); // "Ana"  
  
// Ejemplo en un bloque (if)  
if (true) {  
    let y = 20;  
    console.log(y); // 20  
}  
// console.log(y); // ReferenceError: y is not defined (alcance de bloque)  
  
// -----  
// 1.3. `const`  
// - Introducido en ES6.  
// - Se utiliza para declarar constantes.  
// - No se puede reasignar su valor después de la declaración.  
  
const PI = 3.1416;  
console.log(PI); // 3.1416
```

JAVA SCRIPT

```
// -----  
// 1.4. Tipos de Datos  
  
// Primitivos  
// - `string`: Cadenas de texto.  
// - `number`: Números enteros o decimales.  
// - `boolean`: `true` o `false`.  
// - `undefined`: Variable no asignada.  
// - `null`: Ausencia intencional de valor.  
// - `symbol`: Valor único e inmutable.  
// - `bigint`: Números enteros grandes.  
  
let id = Symbol("id");  
console.log(typeof id); // "symbol"  
  
// Objetos  
// - `object`: Colección de propiedades (pares clave-valor).  
// - `array`: Lista ordenada de valores.  
// - `function`: Bloque de código invocable.  
// - `Date`, `RegExp`, `Error`: Objetos especializados.  
  
let persona = {  
  nombre: "Ana",  
  edad: 30  
};  
console.log(typeof persona); // "object"  
  
let colores = ["rojo", "verde", "azul"];  
console.log(typeof colores); // "object"  
console.log(Array.isArray(colores)); // true  
  
function sumar(a, b) {  
  return a + b;  
}  
console.log(typeof sumar); // "function"  
  
let fecha = new Date();  
console.log(typeof fecha); // "object"
```

JAVA SCRIPT

```
// =====  
// 2. Operadores Básicos  
// =====  
  
// 2.1. Aritméticos  
// - `+`, `-`, `*`, `/`, `%` (residuo).  
  
var edadAna = 34;  
var edadMaria = 28;  
var yearActual = 2025;  
  
var diferenciaEdad = edadAna - edadMaria;  
var sumaEdad = edadAna + edadMaria;  
  
var yearAna = yearActual - edadAna;  
var yearMaria = yearActual - edadMaria;  
  
console.log(diferenciaEdad); // 6  
console.log(sumaEdad); // 62  
console.log('Año en que nació Ana: ' + yearAna); // 1991  
console.log('Año en que nació María: ' + yearMaria); // 1997  
  
// -----  
// 2.2. Lógicos  
// - `<`, `>`, `<=`, `>=`, `==`, `!=`.  
  
var mayorAna = edadAna > edadMaria;  
console.log(mayorAna); // true  
  
var igualdad = !(edadAna == edadMaria);  
console.log(igualdad); // true  
  
// -----  
// 2.3. Unarios  
// - `++` (incremento), `--` (decremento).  
  
edadAna++; // Incremento  
console.log(edadAna); // 35  
  
console.log(edadAna++); // 35 (incremento posterior)  
console.log(edadAna); // 36  
  
console.log(++edadAna); // 37 (incremento previo)
```

JAVA SCRIPT

```
// -----  
// 2.4. Asignación  
// - `+=`, `-=` , `*=` , `/=` , `%=` .  
  
var a = 12;  
var b = 5;  
  
a += b; // a = a + b  
console.log(a); // 17  
  
var c = a % b; // Residuo  
console.log(c); // 2  
  
// =====  
// 3. Estructuras de Código  
// =====  
  
// 3.1. Paréntesis `()`  
// - Llamadas a funciones:  
function sumar(a, b) {  
    return a + b;  
}  
let resultado = sumar(3, 5);  
console.log(resultado); // 8  
  
// - Definición de funciones:  
function saludar(nombre) {  
    console.log("Hola, " + nombre);  
}  
saludar("Juan"); // "Hola, Juan"  
  
// - Agrupación de expresiones:  
let resultadoAgrupado = (2 + 3) * 4;  
console.log(resultadoAgrupado); // 20
```

JAVA SCRIPT

```
// -----  
// 3.2. Llaves `{}`  
// - Bloques de código:  
if (edad >= 18) {  
    console.log("Eres mayor de edad");  
}  
  
// - Objetos:  
let personaObj = {  
    nombre: "Ana",  
    edad: 25  
};  
console.log(personaObj.nombre); // "Ana"  
  
// - Funciones:  
function restar(a, b) {  
    return a - b;  
}  
  
// -----  
// 3.3. Corchetes `[]`  
// - Arrays:  
let coloresArray = ["rojo", "verde", "azul"];  
console.log(coloresArray[0]); // "rojo"  
  
// - Acceso dinámico a propiedades:  
let personaDinamica = { nombre: "Juan", edad: 30 };  
let propiedad = "nombre";  
console.log(personaDinamica[propiedad]); // "Juan"  
  
// - Modificación de arrays:  
let numeros = [1, 2, 3];  
numeros[3] = 4;  
console.log(numeros); // [1, 2, 3, 4]
```

JAVA SCRIPT

```
// =====  
// 4. Objetos  
// =====  
  
const personaConst = { nombre: "Ana", ape1: "Vega" };  
personaConst.nombre = "Carlos"; // Modificación válida  
console.log(personaConst); // { nombre: "Carlos", ape1: "Vega" }  
console.log(personaConst.ape1); // "Vega"
```