

# Sintaxis y Estructura del Lenguaje de JavaScript

## 1. Introducción a JavaScript

JavaScript es un lenguaje de programación de alto nivel, dinámico y orientado a objetos, utilizado principalmente para crear interactividad en páginas web. Su sintaxis es similar a otros lenguajes como C o Java, lo que facilita su aprendizaje si ya se tiene experiencia en esos lenguajes.

## 2. Sintaxis Básica de JavaScript

### a) Declaración de Variables

Las variables en JavaScript se pueden declarar utilizando var, let, o const:

```
var nombre = "Juan"; // Puede ser redeclarada y reasignada
```

```
let edad = 25;      // Puede ser reasignada, pero no redeclarada en el mismo ámbito
```

```
const PI = 3.14;    // No puede ser reasignada ni redeclarada
```

- var: Declara una variable con ámbito global o de función.
- let: Declara una variable con ámbito de bloque.
- const: Declara una constante de bloque, que no puede ser reasignada.

### b) Tipos de Datos

JavaScript es un lenguaje de tipado dinámico. Los principales tipos de datos son:

- Números: Enteros y decimales (let x = 5;)
- Cadenas de texto (Strings): Texto encerrado entre comillas (let saludo = "Hola";)
- Booleanos: Verdadero o falso (let esVerdadero = true;)
- Arreglos: Listas de valores (let colores = ["rojo", "verde", "azul"];)
- Objetos: Colección de pares clave-valor (let persona = {nombre: "Juan", edad: 30};)
- Nulo y indefinido: null (sin valor intencional) y undefined (valor no asignado).

### c) Operadores

- Aritméticos: +, -, \*, /, %
- De asignación: =, +=, -=, \*=, /=
- De comparación: ==, ===, !=, !==, >, <, >=, <=
- Lógicos: &&, ||, !

### d) Estructuras de Control

Las estructuras de control permiten ejecutar diferentes bloques de código en función de condiciones o repetir código.

- Condicionales: if, else if, else

```
if (edad > 18) {  
    console.log("Eres adulto");  
} else {  
    console.log("Eres menor de edad");  
}
```

- Bucles:

- for: Repite un bloque de código un número específico de veces.
- while: Repite mientras una condición sea verdadera.
- do...while: Similar a while, pero se ejecuta al menos una vez.

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

### e) Funciones

Las funciones en JavaScript son bloques de código reutilizables.

- Declaración de función:

```
function saludar(nombre) {
```

```
    return "Hola, " + nombre;
}
```

- Funciones anónimas y flecha:

```
let sumar = (a, b) => a + b;
```

### **3. Objetos y Programación Orientada a Objetos (POO)**

JavaScript soporta la programación orientada a objetos a través de objetos y prototipos.

- Creación de objetos:

```
let persona = {
  nombre: "Carlos",
  edad: 30,
  saludar: function() {
    return "Hola, soy " + this.nombre;
  }
};
```

- Clases: Introducidas en ES6, permiten una sintaxis más clara para definir objetos y herencia.

```
class Persona {
  constructor(nombre, edad) {
    this.nombre = nombre;
    this.edad = edad;
  }
}
```

```
saludar() {  
    return Hola, soy ${this.nombre};  
}  
}
```

```
let juan = new Persona("Juan", 25);
```

#### **4. Manipulación del DOM**

JavaScript interactúa con HTML a través del DOM (Document Object Model). Algunas operaciones comunes incluyen:

- Seleccionar elementos:

```
let titulo = document.getElementById("titulo");
```

- Modificar contenido:

```
titulo.textContent = "Nuevo Título";
```

- Escuchar eventos:

```
titulo.addEventListener("click", function() {  
    alert("Título clickeado");  
});
```

Comprender la sintaxis y la estructura básica de JavaScript es esencial para desarrollar aplicaciones web modernas. Este lenguaje, con su flexibilidad y capacidad para interactuar con el DOM, se ha convertido en una herramienta fundamental para los desarrolladores web. Dominar estos conceptos es el primer paso hacia el desarrollo de aplicaciones dinámicas e interactivas.