

# Curso de Programación Desde Cero

## Tema 3: JavaScript Fundamentals

### 3.1 Introducción a JavaScript

JavaScript es el lenguaje de programación que da vida a las páginas web. Es el único lenguaje que los navegadores web entienden nativamente para crear interactividad, manipular contenido dinámicamente y comunicarse con servidores. Dominar JavaScript es esencial para cualquier desarrollador web.

#### ¿Qué es JavaScript?

JavaScript es un lenguaje de programación interpretado, dinámico y de alto nivel que permite crear experiencias interactivas en la web. A pesar de su nombre, no tiene relación directa con Java - son lenguajes completamente diferentes.

#### Historia y Evolución

- 1995: Brendan Eich crea JavaScript en Netscape en solo 10 días
- 1997: Estandarización como ECMAScript (ES1)
- 2009: ES5 - Estabiliza el lenguaje con nuevas características
- 2015: ES6/ES2015 - Revolución con clases, arrow functions, let/const
- 2016-presente: Actualizaciones anuales con nuevas características

#### ¿Dónde se Ejecuta JavaScript?

- Frontend: En navegadores web (Chrome, Firefox, Safari, Edge)
- Backend: En servidores con Node.js
- Móvil: Aplicaciones híbridas (React Native, Ionic)
- Desktop: Aplicaciones de escritorio (Electron)
- IoT: Dispositivos conectados y microcontroladores

#### Características de JavaScript

- Interpretado: No necesita compilación previa
- Dinámico: Las variables pueden cambiar de tipo
- Orientado a objetos: Soporta programación basada en objetos
- Funcional: Las funciones son ciudadanos de primera clase
- Event-driven: Responde a eventos del usuario
- Case-sensitive: Distingue entre mayúsculas y minúsculas

### 3.2 Configuración del Entorno

#### Herramientas Necesarias

Para empezar a programar en JavaScript solo necesitas:

- Un navegador web moderno (Chrome recomendado)
- Un editor de código (Visual Studio Code recomendado)

## 3.3 Variables y Tipos de Datos en JavaScript

### Declaración de Variables

JavaScript tiene tres formas de declarar variables:

```
// var - forma antigua, evitar usar
var nombre = "Juan";

// let - para variables que pueden cambiar
let edad = 25;
edad = 26; // Puede cambiar

// const - para constantes (no pueden cambiar)
const PI = 3.14159;
const nombre = "María";
// PI = 3.15; // Error! No se puede cambiar
```

### Reglas para Nombrar Variables

- Deben comenzar con letra, \$ o \_
- No pueden empezar con números
- No pueden contener espacios
- Son case-sensitive (edad vs Edad)
- No pueden ser palabras reservadas

```
// Nombres válidos
let nombreUsuario = "Ana";
let _contador = 0;
let $precio = 99.99;
let variable2 = "válido";

// Nombres inválidos
// let 2variable = "error";
// let nombre usuario = "error";
// let class = "error"; // palabra reservada
```

### Tipos de Datos Primitivos

#### 1. Number (Números)

JavaScript maneja enteros y decimales como el mismo tipo:

```
let entero = 42;
let decimal = 3.14;
let negativo = -17;
let científico = 2.5e6; // 2,500,000
let infinito = Infinity;
let noNúmero = NaN; // Not a Number
```

#### 2. String (Cadenas de Texto)

```
let nombre = "María";
let apellido = 'García';
let mensaje = `Hola ${nombre} ${apellido}`; // Template literal
let multilinea = `Primera línea
Segunda línea
Tercera línea`;
```

#### 3. Boolean (Booleanos)

```
let esVerdadero = true;
let esFalso = false;
let esMayor = 25 > 18; // true
let tienePermiso = esVerdadero && esMayor;
```

## 3.4 Operadores y Expresiones

### Operadores Aritméticos

Realizan operaciones matemáticas:

```
let a = 10;
let b = 3;

console.log(a + b); // 13 (suma)
console.log(a - b); // 7 (resta)
console.log(a * b); // 30 (multiplicación)
console.log(a / b); // 3.333... (división)
console.log(a % b); // 1 (módulo - resto)
console.log(a ** b); // 1000 (exponencial)

// Incremento y decremento
let contador = 5;
contador++; // Incrementa en 1 (ahora es 6)
contador--; // Decrementa en 1 (ahora es 5)
++contador; // Pre-incremento
--contador; // Pre-decremento
```

### Operadores de Asignación

```
let x = 10;
x += 5; // x = x + 5 (ahora x = 15)
x -= 3; // x = x - 3 (ahora x = 12)
x *= 2; // x = x * 2 (ahora x = 24)
x /= 4; // x = x / 4 (ahora x = 6)
x %= 5; // x = x % 5 (ahora x = 1)
x **= 3; // x = x ** 3 (ahora x = 1)
```

### Operadores de Comparación

```
let edad1 = 25;
let edad2 = "25";

// Igualdad (convierte tipos)
console.log(edad1 == edad2); // true
console.log(edad1 == 25); // true

// Igualdad estricta (no convierte tipos)
console.log(edad1 === edad2); // false
console.log(edad1 === 25); // true

// Desigualdad
console.log(edad1 != edad2); // false
console.log(edad1 !== edad2); // true

// Comparaciones numéricas
console.log(25 > 20); // true
console.log(25 < 30); // true
console.log(25 >= 25); // true
console.log(25 <= 24); // false
```

### Operadores Lógicos

```
let esAdulto = true;
let tieneLicencia = false;
let edad = 20;

// AND (&&) - Ambas condiciones deben ser true
let puedeConducir = esAdulto && tieneLicencia; // false

// OR (||) - Al menos una condición debe ser true
let puedeEntrar = esAdulto || edad >= 18; // true
```

## 3.5 Estructuras de Control

### Condicionales: if, else if, else

Permiten tomar decisiones basadas en condiciones:

```
// Condicional simple
let edad = 18;
if (edad >= 18) {
    console.log("Eres mayor de edad");
}

// Condicional con else
if (edad >= 18) {
    console.log("Eres mayor de edad");
} else {
    console.log("Eres menor de edad");
}

// MÃºltiples condiciones
let calificacion = 85;
if (calificacion >= 90) {
    console.log("Excelente - A");
} else if (calificacion >= 80) {
    console.log("Muy bien - B");
} else if (calificacion >= 70) {
    console.log("Bien - C");
} else if (calificacion >= 60) {
    console.log("Suficiente - D");
} else {
    console.log("Insuficiente - F");
}
```

### Switch: MÃºltiples Opciones

Ãštil cuando tienes muchas opciones especÃ-ficas:

```
let diaSemana = 3;
let nombreDia;
switch (diaSemana) {
    case 1:
        nombreDia = "Lunes";
        break;
    case 2:
        nombreDia = "Martes";
        break;
    case 3:
        nombreDia = "MiÃ©rcoles";
        break;
    case 4:
        nombreDia = "Jueves";
        break;
    case 5:
        nombreDia = "Viernes";
        break;
    case 6:
    case 7:
        nombreDia = "Fin de semana";
        break;
    default:
        nombreDia = "DÃ-a invÃ¡lido";
}
```

## 3.6 Funciones en JavaScript

### Declaración de Funciones

Las funciones permiten agrupar código reutilizable:

```
// Función básica
function saludar() {
    console.log("¡Hola mundo!");
}

// Llamar a la función
saludar(); // "¡Hola mundo!"

// Función con parámetros
function saludarPersona(nombre) {
    console.log(`¡Hola ${nombre}!`);
}

saludarPersona("Ana"); // "¡Hola Ana!"

// Función con múltiples parámetros
function sumar(a, b) {
    return a + b;
}

let resultado = sumar(5, 3);
console.log(resultado); // 8
```

### Expresiones de Función

```
// Función anónima asignada a variable
const multiplicar = function(a, b) {
    return a * b;
};

console.log(multiplicar(4, 7)); // 28

// Arrow functions (ES6)
const dividir = (a, b) => {
    return a / b;
};

// Arrow function simplificada
const cuadrado = x => x * x;
const saludar = nombre => `Hola ${nombre}`;

console.log(cuadrado(5)); // 25
console.log(saludar("Carlos")); // "Hola Carlos"
```

### Parámetros por Defecto

```
function crearSaludo(nombre = "Usuario", hora = "día") {
    return `Buenos ${hora}, ${nombre}`;
}

console.log(crearSaludo()); // "Buenos día, Usuario"
console.log(crearSaludo("Ana")); // "Buenos día, Ana"
console.log(crearSaludo("Carlos", "tarde")); // "Buenos tarde, Carlos"
```

### Funciones de Orden Superior

Funciones que reciben otras funciones como parámetros:

```
// Función que recibe otra función
function procesar(numeros, operacion) {
    let resultado = [];
    for (let numero of numeros) {
        resultado.push(operacion(numero));
```

## 3.7 Manipulación del DOM

### ¿Qué es el DOM?

El DOM (Document Object Model) es la representación en memoria de la estructura HTML de una página. JavaScript puede manipular el DOM para cambiar contenido, estilos y estructura dinámicamente.

### Seleccionar Elementos

```
// Por ID  
let titulo = document.getElementById("mi-titulo");  
  
// Por clase (devuelve el primero)  
let primerBoton = document.getElementsByClassName("boton")[0];  
  
// Por etiqueta  
let todosLosParrafos = document.getElementsByTagName("p");  
  
// Selectores CSS (más flexibles)  
let elemento = document.querySelector("#mi-id");  
let primerElemento = document.querySelector(".mi-clase");  
let todosLosElementos = document.querySelectorAll(".mi-clase");
```

### Modificar Contenido

```
let titulo = document.getElementById("titulo");  
  
// Cambiar texto  
titulo.textContent = "Nuevo título";  
titulo.innerText = "Otro título";  
  
// Cambiar HTML  
titulo.innerHTML = "<strong>Título en negrita</strong>";  
  
// Modificar atributos  
let imagen = document.querySelector("img");  
imagen.src = "nueva-imagen.jpg";  
imagen.alt = "Nueva descripción";  
imagen.setAttribute("title", "Título de la imagen");
```

### Modificar Estilos

```
let elemento = document.getElementById("mi-elemento");  
  
// Estilos individuales  
elemento.style.color = "red";  
elemento.style.backgroundColor = "yellow";  
elemento.style.fontSize = "20px";  
elemento.style.display = "none"; // Ocultar  
  
// Agregar/quitar clases CSS  
elemento.classList.add("nueva-clase");  
elemento.classList.remove("clase-vieja");  
elemento.classList.toggle("activo"); // Cambia si existe  
elemento.classList.contains("mi-clase"); // true/false
```

### Crear y Agregar Elementos

```
// Crear nuevo elemento  
let nuevoParrafo = document.createElement("p");  
nuevoParrafo.textContent = "Este es un nuevo párrafo";  
nuevoParrafo.className = "parrafo-dinamico";  
  
// Agregar al DOM  
let contenedor = document.getElementById("contenedor");  
contenedor.appendChild(nuevoParrafo);
```

## 3.8 Resumen del Tema 3

### Conceptos Fundamentales Aprendidos

En este tema hemos dominado JavaScript, el lenguaje que da vida a la web:

#### âœ“ Fundamentos de JavaScript

- â€¢ Historia y evoluciÃ³n del lenguaje
- â€¢ ConfiguraciÃ³n del entorno de desarrollo
- â€¢ Formas de ejecutar cÃ³digo JavaScript
- â€¢ CaracterÃ-sticas del lenguaje

#### âœ“ Variables y Tipos de Datos

- â€¢ DeclaraciÃ³n con let, const y var
- â€¢ Tipos primitivos: number, string, boolean, undefined, null
- â€¢ Tipos complejos: arrays y objetos
- â€¢ ConversiÃ³n y verificaciÃ³n de tipos

#### âœ“ Operadores y Expresiones

- â€¢ AritmÃ©ticos, de comparaciÃ³n y lÃ³gicos
- â€¢ Operador ternario para condicionales simples
- â€¢ Precedencia y asociatividad
- â€¢ Template literals para strings

#### âœ“ Estructuras de Control

- â€¢ Condicionales: if, else if, else, switch
- â€¢ Bucles: for, while, do-while, for...of, for...in
- â€¢ Control de flujo: break y continue

#### âœ“ Funciones

- â€¢ DeclaraciÃ³n de funciones y expresiones
- â€¢ Arrow functions y sintaxis moderna
- â€¢ ParÃ¡metros, valores de retorno y scope
- â€¢ Funciones de orden superior y mÃ©todos de array

#### âœ“ ManipulaciÃ³n del DOM

- â€¢ SelecciÃ³n de elementos HTML
- â€¢ ModificaciÃ³n de contenido y estilos
- â€¢ CreaciÃ³n y eliminaciÃ³n de elementos
- â€¢ Manejo de eventos para interactividad

### Ejercicios de AutoevaluaciÃ³n

Para consolidar tu aprendizaje, desarrolla estos proyectos:

#### 1. Calculadora Interactiva

Crea una calculadora completa con interfaz web y todas las operaciones.

#### 2. Juego de Adivinanza

Implementa un juego donde el usuario adivine un nÃºmero aleatorio.

#### 3. Lista de Tareas Avanzada

Expande la lista bÃ¡sica con ediciÃ³n, eliminaciÃ³n y persistencia local.

#### 4. Validador de Formularios

Crea un formulario con validaciÃ³n en tiempo real usando JavaScript.