

Curso de Programaci3n Desde Cero

Tema 3: JavaScript Fundamentals

3.1 Introducci3n a JavaScript

JavaScript es el lenguaje de programaci3n que da vida a las p3ginas web. Es el 3nico lenguaje que los navegadores web entienden nativamente para crear interactividad, manipular contenido din3micamente y comunicarse con servidores. Dominar JavaScript es esencial para cualquier desarrollador web.

Â¿Qu3 es JavaScript?

JavaScript es un lenguaje de programaci3n interpretado, din3mico y de alto nivel que permite crear experiencias interactivas en la web. A pesar de su nombre, no tiene relaci3n directa con Java - son lenguajes completamente diferentes.

Historia y Evoluci3n

- â€¢ 1995: Brendan Eich crea JavaScript en Netscape en solo 10 d3as
- â€¢ 1997: Estandarizaci3n como ECMAScript (ES1)
- â€¢ 2009: ES5 - Estabiliza el lenguaje con nuevas caracter3sticas
- â€¢ 2015: ES6/ES2015 - Revoluci3n con clases, arrow functions, let/const
- â€¢ 2016-presente: Actualizaciones anuales con nuevas caracter3sticas

Â¿D3nde se Ejecuta JavaScript?

- â€¢ Frontend: En navegadores web (Chrome, Firefox, Safari, Edge)
- â€¢ Backend: En servidores con Node.js
- â€¢ M3vil: Aplicaciones h3bridas (React Native, Ionic)
- â€¢ Desktop: Aplicaciones de escritorio (Electron)
- â€¢ IoT: Dispositivos conectados y microcontroladores

Caracter3sticas de JavaScript

- â€¢ Interpretado: No necesita compilaci3n previa
- â€¢ Din3mico: Las variables pueden cambiar de tipo
- â€¢ Orientado a objetos: Soporta programaci3n basada en objetos
- â€¢ Funcional: Las funciones son ciudadanos de primera clase
- â€¢ Event-driven: Responde a eventos del usuario
- â€¢ Case-sensitive: Distingue entre may3sculas y min3sculas

3.2 Configuraci3n del Entorno

Herramientas Necesarias

Para empezar a programar en JavaScript solo necesitas:

- â€¢ Un navegador web moderno (Chrome recomendado)
- â€¢ Un editor de c3digo (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-a";
// 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 ≠ 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 cientifico = 2.5e6; // 2,500,000
let infinito = Infinity;
let noNumero = NaN; // Not a Number
```

2. String (Cadenas de Texto)

```
let nombre = "María-a";
let apellido = 'García-a';
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");
}

// Múltiples 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: Múltiples Opciones

Estil cuando tienes muchas opciones específicas:

```
let diaSemana = 3;
let nombreDia;
switch (diaSemana) {
  case 1:
    nombreDia = "Lunes";
    break;
  case 2:
    nombreDia = "Martes";
    break;
  case 3:
    nombreDia = "Miércoles";
    break;
  case 4:
    nombreDia = "Jueves";
    break;
  case 5:
    nombreDia = "Viernes";
    break;
  case 6:
  case 7:
    nombreDia = "Fin de semana";
    break;
  default:
    nombreDia = "Día inválido";
}
```

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 = "dÃ-a") {
  return `Buenos ${hora}, ${nombre}`;
}

console.log(crearSaludo()); // "Buenos dÃ-a, Usuario"
console.log(crearSaludo("Ana")); // "Buenos dÃ-a, Ana"
console.log(crearSaludo("Carlos", "tardes")); // "Buenos tardes, 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:

“ Fundamentos de JavaScript

- Historia y evolución del lenguaje
- Configuración del entorno de desarrollo
- Formas de ejecutar código JavaScript
- Características del lenguaje

“ Variables y Tipos de Datos

- Declaración con let, const y var
- Tipos primitivos: number, string, boolean, undefined, null
- Tipos complejos: arrays y objetos
- Conversión y verificación de tipos

“ Operadores y Expresiones

- Aritméticos, de comparación y lógicos
- Operador ternario para condicionales simples
- Precedencia y asociatividad
- Template literals para strings

“ Estructuras de Control

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

“ Funciones

- Declaración de funciones y expresiones
- Arrow functions y sintaxis moderna
- Parámetros, valores de retorno y scope
- Funciones de orden superior y métodos de array

“ Manipulación del DOM

- Selección de elementos HTML
- Modificación de contenido y estilos
- Creación y eliminación de elementos
- Manejo de eventos para interactividad

Ejercicios de Autoevaluación

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 número aleatorio.

3. Lista de Tareas Avanzada

Expande la lista básica con edición, eliminación y persistencia local.

4. Validador de Formularios

Crea un formulario con validación en tiempo real usando JavaScript.