




Introducción a JavaScript

 2026-01   50 min.

¿Qué es JavaScript?

JavaScript

JavaScript es el lenguaje de programación de la web.

Agrega **interactividad** y **comportamiento** a las páginas web.

Es el único lenguaje que funciona nativamente en todos los navegadores.

Función primaria: Brindar interactividad a páginas web

HTML, CSS, y JavaScript

HTML: Contenido

CSS: Estilos

JavaScript: Interactividad

Juntos forman la base del desarrollo web frontend.

Historia de JavaScript

- Creado en **1995** por una persona
- Desarrollado en **10 días**
- Poco en común con Java
- Visualmente se parece a Java, pero son lenguajes muy diferentes
- Creado específicamente para el navegador
- Durante la Web 1.0 (< 2004), JS era exclusivamente para el navegador

Características de JavaScript

- No requiere compilación
- Las variables pueden cambiar de tipo
- Puede manejar operaciones asíncronas
- Reacciona a eventos del usuario o del sistema

Cómo agregar JavaScript a HTML?

Inline (en línea):

```
<button onclick="alert('Hola!')">Click</button>
```

Script interno:

```
<script>  
  console.log('Hola desde JavaScript');  
</script>
```

Script externo:

```
<script src="script.js"></script>
```

Herramientas de desarrollo

Console del navegador:

- F12 o Click derecho → Inspeccionar

```
console.log('Hola mundo');  
console.log(variable);  
console.log('Valor:', 42);
```


Hola Mundo

```
// Comentario de una línea

/*
  Comentario
  de múltiples
  líneas
*/

// Imprimir en la consola
console.log('Hola Mundo');

// Mostrar alerta en el navegador
alert('Hola Mundo');

// Modificar el contenido de la página
document.body.innerHTML = '<h1>Hola Mundo</h1>';
```

Frameworks y el desarrollo frontend moderno

Los tres grandes frameworks

React, creado por Facebook.

Vue, creado por Evan You.

Angular, creado por Google.

Ecosistema JavaScript

Frameworks/Librerías Frontend:

- React, Vue, Angular, Svelte

Backend:

- Node.js, Express, NestJS

Build Tools:

- Webpack, Vite, Parcel

Package Managers:

- npm, yarn, pnpm

Entonces, ¿qué haremos en este curso?

- Crear componentes de React
- Aplicar estilos a los componentes
- Manejar el estado de la aplicación
- Implementar routing y navegación
- Consumir APIs REST desde el frontend
- Utilizar un repositorio de Git para manejar versionamiento

¿Qué se necesita para comenzar?

- NodeJS instalado
- Algún entorno de desarrollo (como VS Code)
- Un navegador moderno (Chrome, Firefox, Edge)
- Git instalado

