



Universidad Politécnica de Lázaro Cárdenas

Proyecto: Historia y evolución de JavaScript

Asignatura: Programación Web

Integrante(s): Moisés Eduardo Valencia Hernández

Docente: Ing. Andrés Luna Jaimes

18 de marzo de 2025

Tabla de Contenido

¿Cómo Nace?	2
¿Quién Es Su Creador?	3
¿Por Qué Es Un Lenguaje De Programación?	4
¿Por Qué Es Un Estándar En La Web?	6
¿Cómo Funciona?	7
Situación Actual	8

¿Cómo nace?

JavaScript nació en 1995, desarrollado por Brendan Eich mientras trabajaba en Netscape Communications Corporation. Netscape buscaba agregar dinamismo a las páginas web, ya que hasta entonces solo se podían estructurar con HTML y estilizar con CSS.

Brendan Eich creó JavaScript en solo 10 días, basándose en elementos de los lenguajes Scheme y Self, pero con una sintaxis similar a Java por razones de marketing. Inicialmente, el lenguaje se llamó Mocha, luego LiveScript, y finalmente fue renombrado como JavaScript.

Microsoft desarrolló su propia versión llamada JScript, lo que generó incompatibilidades entre navegadores. Para solucionar esto, en 1997 se estableció el estándar ECMAScript, que regula el desarrollo de JavaScript hasta la actualidad.

¿Quién es su creador?

Brendan Eich: El Creador de JavaScript

nació en 1995, desarrollado por Brendan Eich mientras trabajaba en Netscape Communications Corporation. Netscape buscaba agregar dinamismo a las páginas web, ya que hasta entonces solo se podían estructurar con HTML y estilizar con CSS.

Brendan Eich, un programador estadounidense con experiencia en lenguajes como Scheme y Self, fue el encargado de desarrollar este nuevo lenguaje de scripting. En un tiempo récord de 10 días, Eich diseñó JavaScript con una sintaxis similar a Java, aunque su funcionalidad y propósito eran diferentes. El lenguaje pasó por varios nombres antes de establecerse como JavaScript, inicialmente se llamó Mocha y luego LiveScript.

La decisión de darle una sintaxis parecida a Java fue principalmente una estrategia de marketing, ya que en ese momento Java era un lenguaje muy popular y Netscape quería atraer la atención de los desarrolladores. Sin embargo, JavaScript fue concebido como un lenguaje más accesible y orientado a la manipulación del contenido web en el navegador.

La Estandarización y Evolución de JavaScript Con la creciente popularidad del lenguaje, Microsoft lanzó su propia versión llamada JScript, lo que provocó incompatibilidades entre navegadores. Para solucionar este problema, en 1997 se estableció el estándar ECMAScript, que regula la evolución de JavaScript hasta la actualidad.

Brendan Eich continuó desempeñando un papel clave en el desarrollo de la tecnología web y más tarde se convirtió en CTO y CEO de Mozilla, la organización responsable de Firefox. Su contribución al mundo de la programación sigue siendo fundamental, ya que JavaScript es ahora uno de los lenguajes más utilizados en el desarrollo web moderno, tanto en el frontend como en el backend con tecnologías como Node.js.

Gracias al trabajo de Brendan Eich, JavaScript se ha convertido en una pieza clave de la web, permitiendo la creación de sitios y aplicaciones dinámicas que forman parte de la vida digital de millones de personas en todo el mundo.

¿Por qué es un lenguaje de programación?

JavaScript es un lenguaje de programación porque permite a los desarrolladores escribir instrucciones que una computadora puede interpretar y ejecutar. A diferencia de HTML y CSS, que solo estructuran y estilizan el contenido de una página web, JavaScript proporciona **lógica y dinamismo**, permitiendo la manipulación de datos, interactividad y automatización de tareas.

Características que lo definen como un lenguaje de programación:

1. Ejecución de instrucciones: JavaScript permite ejecutar líneas de código en un orden específico para realizar tareas como cálculos matemáticos, manipulación de elementos en una página web y validaciones de formularios.
2. Variables y Tipos de Datos: Permite almacenar y manipular datos con variables (var, let, const) y soporta múltiples tipos de datos como números, cadenas de texto, booleanos, objetos y arreglos.
3. Estructuras de Control: Utiliza condicionales (if, else, switch) y bucles (for, while) para controlar el flujo de ejecución del programa.
4. Funciones: JavaScript permite la creación de funciones para encapsular código reutilizable y mejorar la modularidad del programa.
5. Programación Orientada a Objetos (POO): Aunque JavaScript no es un lenguaje puramente orientado a objetos, permite la creación de objetos, clases y prototipos para organizar el código de manera eficiente.
6. Interacción con el entorno: Puede manipular el DOM (Document Object Model) para cambiar dinámicamente el contenido y estilo de una página web en respuesta a eventos del usuario.
7. Asincronía y eventos: Soporta programación asíncrona mediante promesas, async/await y el modelo basado en eventos, lo que permite realizar tareas en segundo plano sin bloquear la ejecución del programa.

Gracias a estas características, JavaScript se considera un lenguaje de programación versátil y esencial en el desarrollo web moderno.

¿Por qué es un estándar en la web?

JavaScript es un estándar en la web porque es el único lenguaje de programación que los navegadores ejecutan de forma nativa, permitiendo la interactividad y dinamismo en las páginas web sin necesidad de complementos adicionales. Su adopción universal ha sido clave para su consolidación como un pilar fundamental en el desarrollo web.

Razones por las que JavaScript es un estándar: Compatibilidad universal: Todos los navegadores modernos (Chrome, Firefox, Safari, Edge, etc.) soportan JavaScript sin necesidad de instalación adicional.

1. Interactividad web: Permite agregar funcionalidades como validaciones de formularios, animaciones, efectos visuales y comunicación con servidores en tiempo real.
2. Ecosistema robusto: Cuenta con una enorme comunidad de desarrolladores y una gran cantidad de bibliotecas y frameworks (React, Angular, Vue, etc.) que facilitan su uso en aplicaciones web complejas.
3. Estandarización (ECMAScript): Desde 1997, JavaScript ha sido regulado por ECMAScript, asegurando su evolución constante con nuevas características y mejoras.

4. Uso en backend: Con la llegada de Node.js, JavaScript dejó de ser solo un lenguaje para navegadores y se convirtió en una tecnología de servidor, ampliando aún más su relevancia en la web.

Gracias a su versatilidad, compatibilidad y constante evolución, JavaScript se ha mantenido como el estándar de la web, siendo un lenguaje imprescindible para cualquier desarrollador que trabaje en entornos digitales.

¿Cómo funciona?

JavaScript funciona ejecutándose dentro de un motor de JavaScript en el navegador o en entornos de servidor como Node.js. Su funcionamiento se basa en la interpretación y ejecución de código en tiempo real, lo que permite la manipulación dinámica del contenido web.

1. Principales aspectos del funcionamiento de JavaScript: Ejecutado por un motor de JavaScript: Cada navegador tiene su propio motor de JavaScript que interpreta y ejecuta el código. Algunos ejemplos son V8 (Chrome y Node.js), SpiderMonkey (Firefox) y JavaScriptCore (Safari).
2. Basado en eventos: JavaScript permite manejar eventos como clics, movimientos del mouse y cambios en formularios para responder dinámicamente a la interacción del usuario.
3. Manipulación del DOM: Puede acceder y modificar el DOM para cambiar el contenido, estilo y estructura de una página web en tiempo real.

4. Ejecución síncrona y asíncrona: Aunque JavaScript se ejecuta de forma secuencial, puede manejar tareas asíncronas con `setTimeout`, `setInterval`, promesas y `async/await`, lo que permite ejecutar código sin bloquear la página.
5. Comunicación con servidores: A través de AJAX y Fetch API, JavaScript puede enviar y recibir datos de servidores sin necesidad de recargar la página.
6. Uso en backend: Con Node.js, JavaScript puede ejecutarse en servidores, manejando bases de datos, autenticación de usuarios y lógica de negocio en aplicaciones web.

Gracias a estas características, JavaScript es un lenguaje potente y flexible que permite desarrollar desde pequeñas interacciones en páginas web hasta aplicaciones web complejas.

Situación actual

En la actualidad, JavaScript sigue siendo uno de los lenguajes de programación más utilizados en el mundo. Su ecosistema continúa expandiéndose con nuevas tecnologías y herramientas que lo hacen más eficiente y versátil. Algunas tendencias y aspectos relevantes de su estado actual incluyen:

1. Crecimiento de frameworks y bibliotecas: Herramientas como React.js, Vue.js y Angular han impulsado el desarrollo de aplicaciones web modernas y eficientes.
2. Expansión en el backend: Gracias a Node.js, JavaScript no solo domina el frontend, sino también el desarrollo del lado del servidor, permitiendo la creación de aplicaciones escalables y eficientes.

3. Popularidad en la comunidad: JavaScript es constantemente actualizado y mejorado por la comunidad de desarrolladores y la organización ECMAScript, asegurando su relevancia a largo plazo.
4. Desarrollo de aplicaciones móviles y de escritorio: Tecnologías como React Native y Electron han permitido el uso de JavaScript para desarrollar aplicaciones móviles y de escritorio multiplataforma.
5. Mejoras en rendimiento y seguridad: Motores como V8 han optimizado la ejecución de JavaScript, y herramientas como TypeScript han mejorado la estructura y seguridad del código.

Gracias a su constante evolución, JavaScript sigue siendo la tecnología base del desarrollo web y su influencia se extiende a múltiples áreas del desarrollo de software.