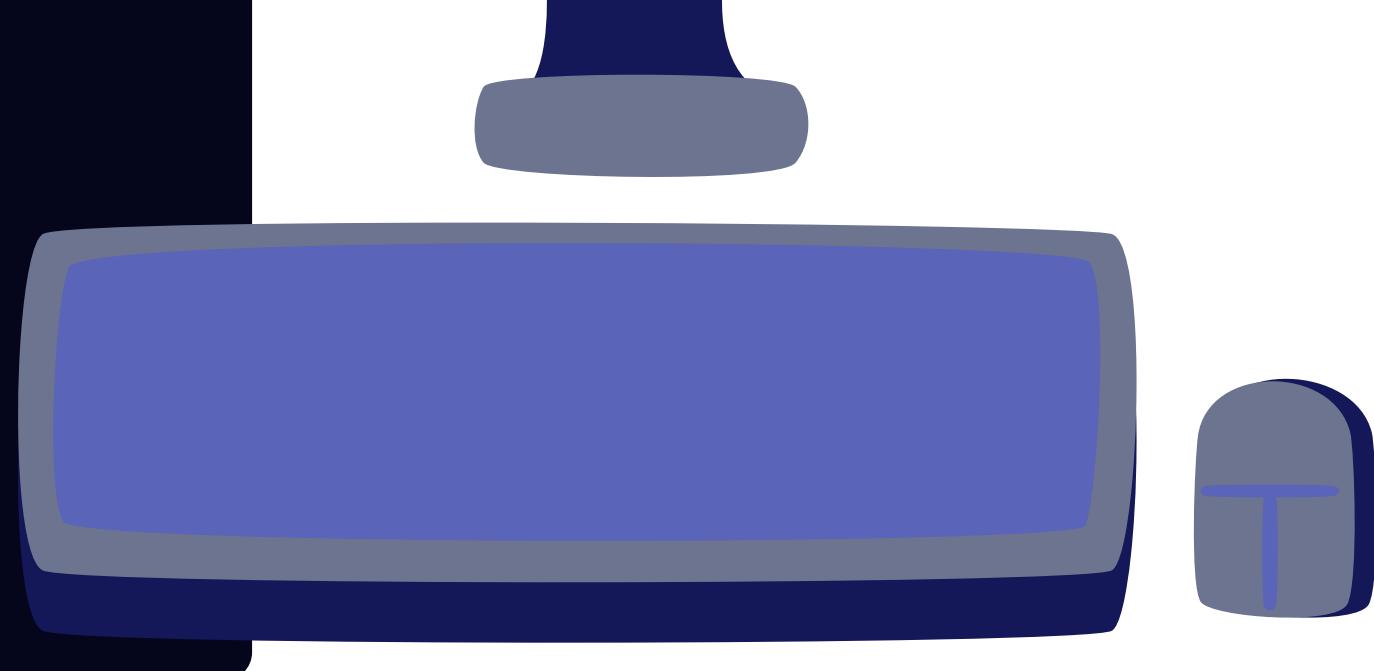
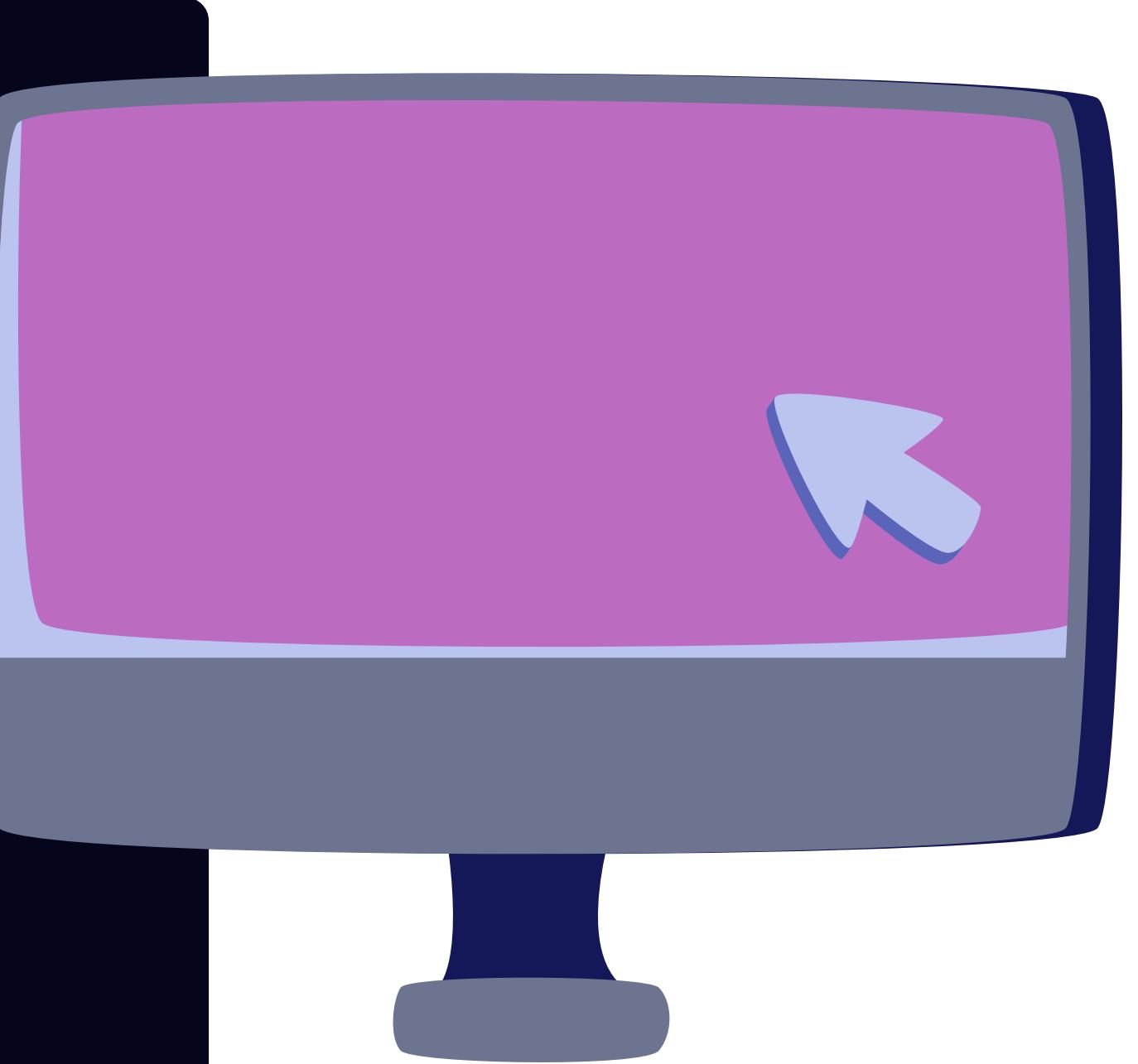
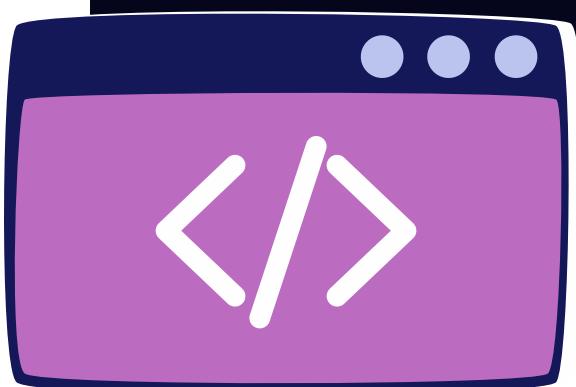


Taller Realizado por Damian Trajster

JAVASCRIPT

Introducción

JavaScript es un lenguaje de programación interpretado que se ejecuta principalmente en los navegadores web. Su propósito principal es añadir interactividad y dinamismo a las páginas web, permitiendo, por ejemplo, la actualización de contenido sin recargar la página, la validación de formularios en tiempo real, y la creación de animaciones, entre muchas otras funcionalidades.



1. JAVASCRIPT COMO LENGUAJE INTERPRETADO

Un lenguaje interpretado se ejecuta línea por línea a medida que el navegador lo interpreta, sin necesidad de una compilación previa. Esto hace que JavaScript sea ideal para el desarrollo web, ya que los cambios en el código pueden verse inmediatamente en el navegador.

HISTORIA Y EVOLUCIÓN DE JAVASCRIPT

JavaScript fue creado en 1995 por Brendan Eich mientras trabajaba en Netscape Communications Corporation, los desarrolladores del popular navegador Netscape Navigator. Originalmente, JavaScript fue creado en solo 10 días, y se llamó "Mocha", luego "LiveScript", y finalmente "JavaScript" en un esfuerzo por aprovechar la popularidad del lenguaje Java, aunque no están directamente relacionados.

Principales hitos en su evolución:

01

1997 -
ESTANDARIZACIÓN
COMO ECMASCIPT:

JavaScript fue estandarizado bajo el nombre de ECMAScript (ES) para asegurar compatibilidad entre navegadores. ECMAScript define las especificaciones del lenguaje, y cada actualización importante se conoce por su versión, como ES5, ES6, etc.

02

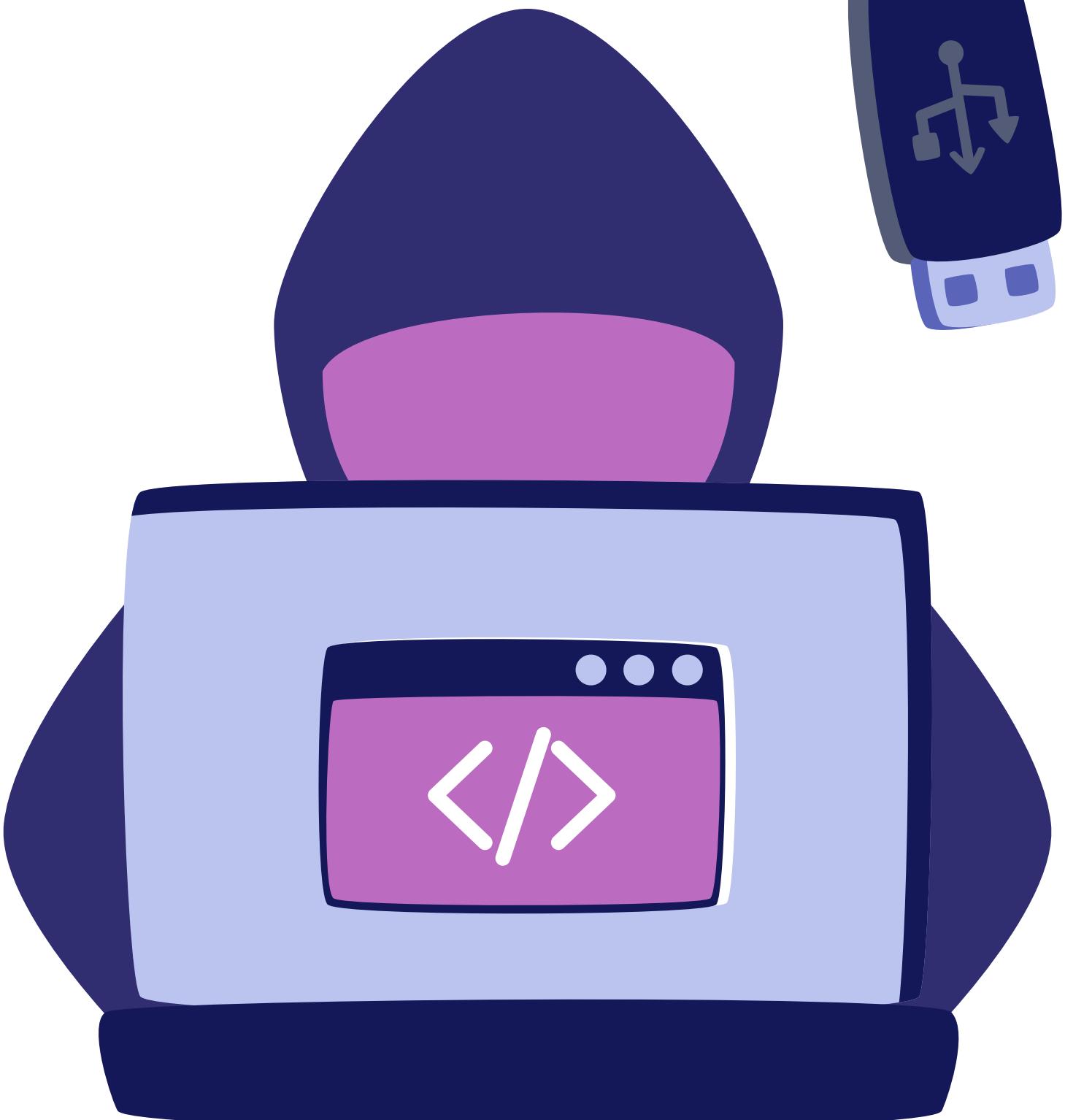
2009 - ES5
(ECMASCRIPT 5):

Introdujo importantes mejoras, como JSON, el método `Array.forEach()`, y mejoras en el manejo de propiedades y objetos.

03

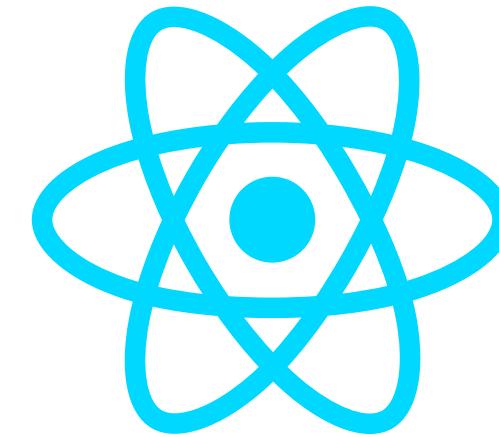
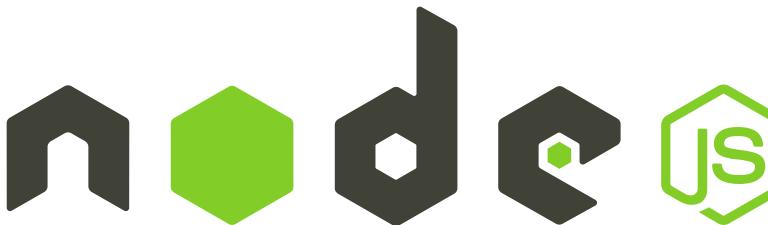
2015 - ES6
(ECMASCRIPT 6 0
ES2015)

Esta actualización transformó JavaScript en un lenguaje mucho más poderoso, agregando funcionalidades como `let` y `const` para definir variables, funciones de flecha, clases, módulos, y Promises, entre otras características.



3. JavaScript Hoy: Más Allá del Navegador

- **Backend (Node.js):** Con la llegada de Node.js, JavaScript se expandió al backend, permitiendo a los desarrolladores usar un mismo lenguaje en el servidor y en el cliente.
- **Aplicaciones de Escritorio y Móviles:** JavaScript es clave para el desarrollo de aplicaciones móviles y de escritorio, con frameworks como Electron (para aplicaciones de escritorio), React Native (para aplicaciones móviles) y Ionic.
- **Desarrollo Full Stack:** Al usar JavaScript en el frontend y el backend, es posible crear aplicaciones completas con frameworks como Express (backend), React , Angular (frontend) o Astro el framework mas nuevo de JS (2022)



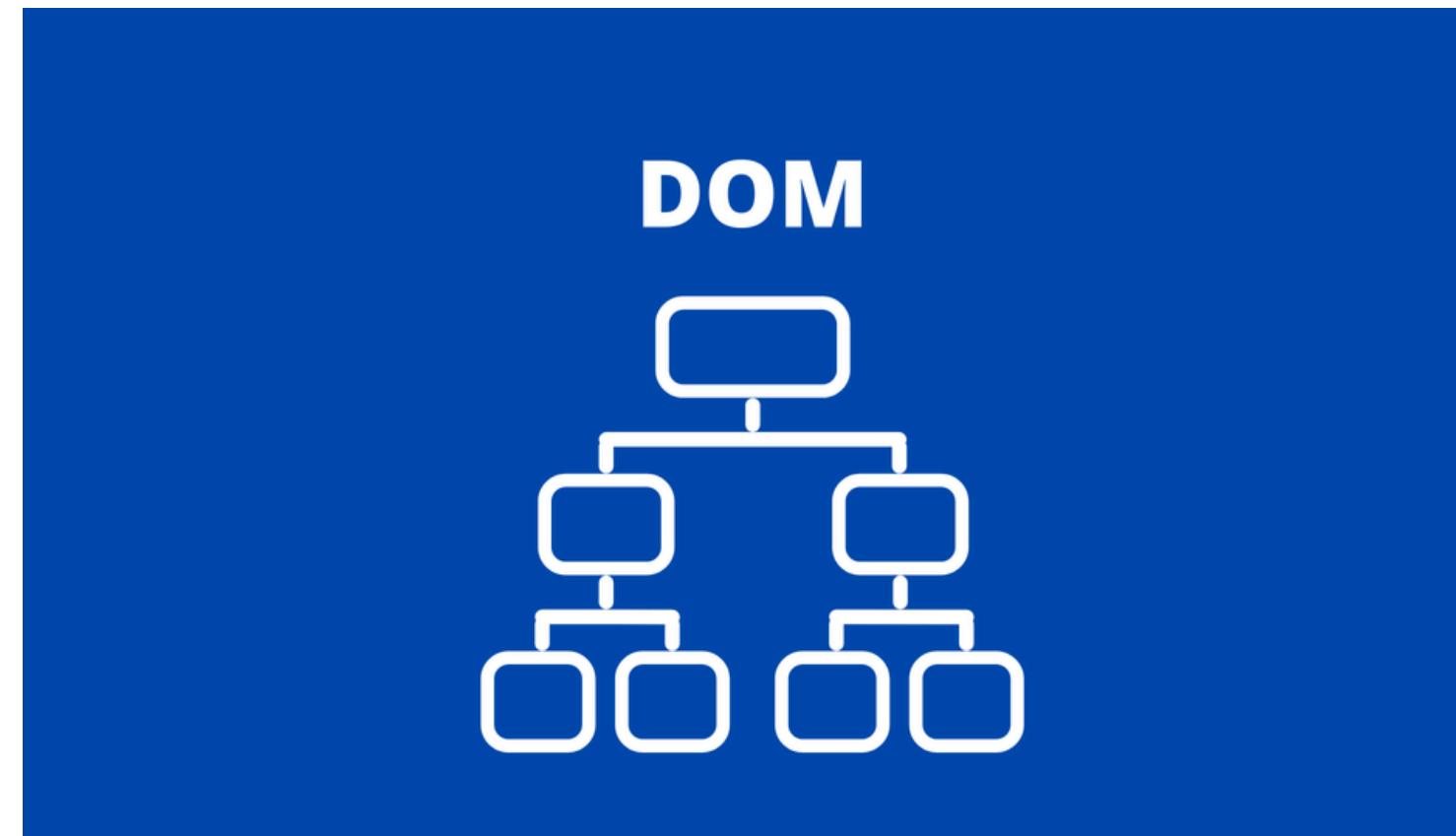
DOM (DOCUMENT OBJECT MODEL)

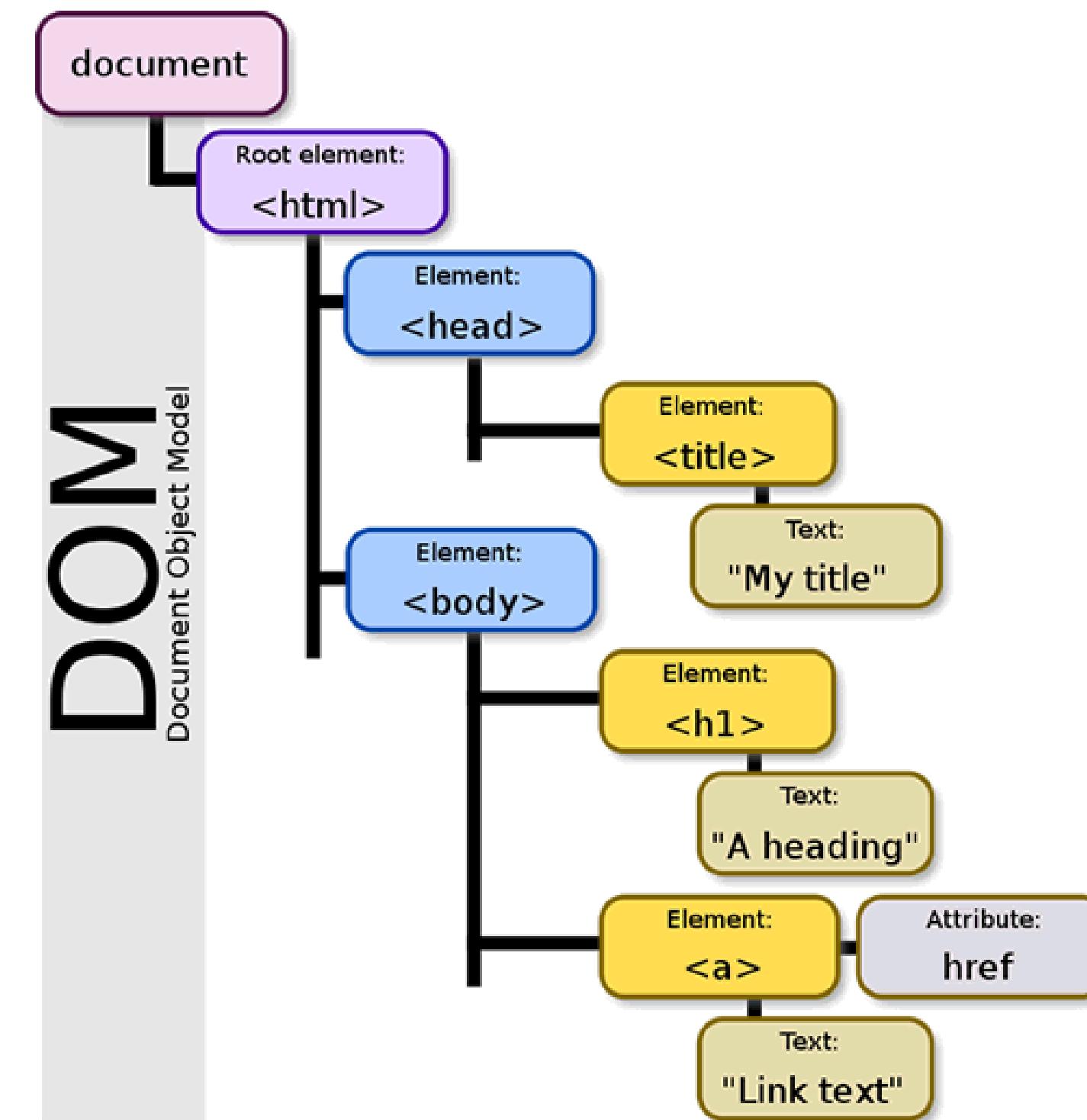
El DOM (Document Object Model) es una representación en forma de árbol de todos los elementos de una página web. Cada etiqueta HTML, atributo y contenido de texto se convierte en un "nodo" o "objeto" en el DOM, al cual JavaScript puede acceder y manipular directamente. Esto permite modificar dinámicamente el contenido, el estilo y los atributos de los elementos en una página.

¿Cómo JavaScript interactúa con el DOM?

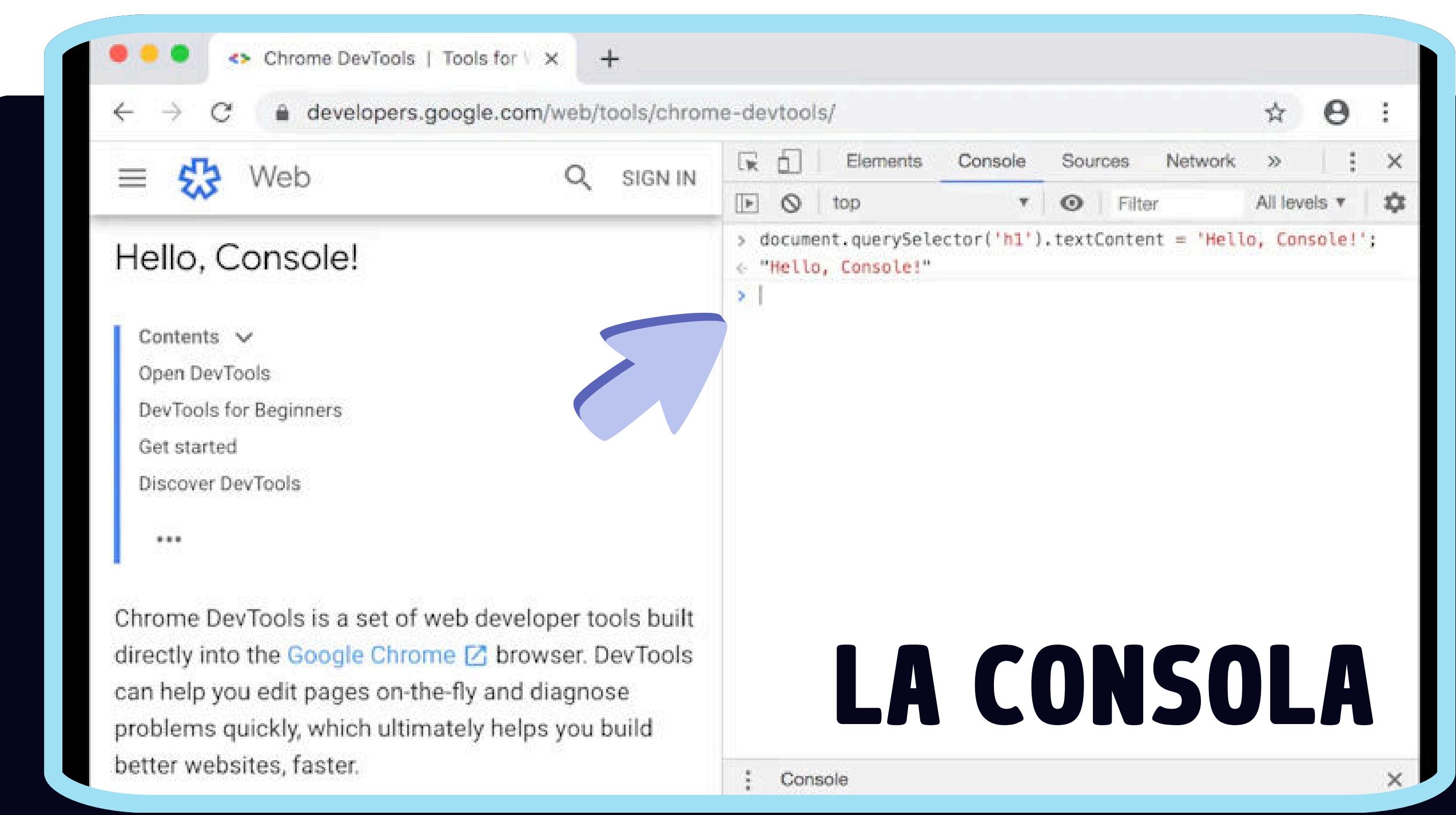
A través del DOM, JavaScript puede:

1. **Modificar el contenido:** Cambiar o actualizar el texto o los elementos en la página.
2. **Estilizar elementos:** Cambiar colores, tamaños, visibilidad, etc., de los elementos.





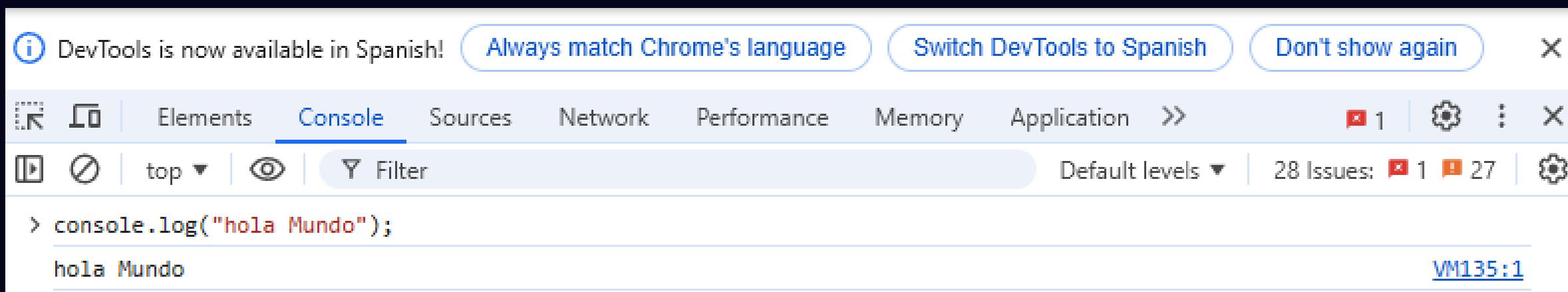
La manipulación del DOM es fundamental en el desarrollo de páginas interactivas y dinámicas. JavaScript, al interactuar con el DOM, permite que las aplicaciones respondan a las acciones del usuario, generando una experiencia más rica e interactiva en tiempo real.



Para abrir las herramientas de desarrollo en la mayoría de los navegadores (como Chrome, Firefox o Edge), usa uno de los siguientes métodos:

- Tecla F12.
- Ctrl + Shift + I (Windows/Linux) o Cmd + Option + I (Mac).
- Clic derecho en la página web y selecciona "Inspeccionar" o "Inspeccionar elemento".

Las Herramientas de Desarrollo de los navegadores web son una herramienta fundamental para cualquier desarrollador de JavaScript, ya que permiten realizar pruebas, depuración y análisis del código directamente en el navegador.



La consola es una de las herramientas más útiles para los desarrolladores, permitiendo ejecutar comandos JavaScript en tiempo real, ver mensajes de error y hacer pruebas sin necesidad de modificar el código de la página.

CONCLUSIÓN

JavaScript es un lenguaje que ha evolucionado desde su inicio como un simple lenguaje de scripts a convertirse en una pieza esencial del desarrollo en múltiples entornos, trascendiendo el navegador y consolidándose como un lenguaje versátil y potente en la programación.

A CODEAR!!!