<!--Estudio Shonos-->

Introduccion a javascript {

<Presentada por="Los miembros del equipo 4"/>

}

JS

Contenidos

Historia {

En 1994 nació Netscape Navigator, el primer navegador masivo que llevó la web al público general. Antes existía Mosaic, usado solo por la élite tecnológica. Netscape revolucionó el acceso a internet y, tras salir a bolsa, atrajo la atención de Bill Gates, quien lanzó Internet Explorer, iniciando la famosa guerra de navegadores.

En 1995, Netscape buscaba hacer la web más interactiva y contrató a Brendan Eich, quien creó LiveScript en solo una semana. Poco después fue renombrado JavaScript por motivos de marketing, aunque no guarda relación con Java. Microsoft respondió con JScript, lo que generó incompatibilidades. Para resolverlo, en 1997 nació ECMAScript 1, la primera estandarización del lenguaje. Sin embargo, tras ECMAScript 3 en 1999, el desarrollo se estancó; la versión 4 nunca salió y JavaScript fue visto como un lenguaje limitado, útil solo para animaciones simples, el “patito feo” de la programación.

}

Historia {

En 2002, Netscape liberó su código, dando origen a Mozilla Firefox, símbolo de libertad frente a Internet Explorer. En 2006 apareció jQuery, que simplificó el código y aseguró compatibilidad entre navegadores, dominando casi una década.

La revolución llegó en 2008 con Google Chrome y su motor V8, mucho más rápido que la competencia. Esto permitió crear aplicaciones web más complejas. En 2009, Ryan Dahl tomó V8 y lo llevó al servidor con Node.js, permitiendo usar JavaScript en el backend y transformando el desarrollo web.

}

Caracteristicas {

Lenguaje Interpretado

No necesita compilación previa; el navegador lo interpreta directamente

Tipado débil y dinámico

Las variables no requieren declaración de tipo, y pueden cambiar de tipo según el valor asignado

Imperativo y estructurado

Las instrucciones se ejecutan de forma secuencial, y permite estructuras como bucles, condicionales y funciones

Orientado a objetos

basado en prototipos

Aunque no usa clases como Java o C++, permite crear objetos y herencia mediante prototipos

Multiplataforma

Funciona en cualquier navegador y sistema operativo, tanto en el cliente como en el servidor (gracias a Node.js)

Lenguaje ligero y sencillo

Fácil de aprender y usar, ideal para principiantes y muy utilizado en desarrollo web

}

Ventajas {

Universalidad: Funciona en todos los navegadores modernos sin necesidad de instalación extra.
Rapidez: El código se ejecuta directamente en el navegador, sin necesidad de compilación.
Interactividad: Permite hacer páginas dinámicas, animaciones, validaciones y mejorar la experiencia del usuario.
Versatilidad: Se puede usar en frontend (navegador) y backend (con Node.js).
Gran comunidad: Tiene muchas librerías y frameworks (React, Angular, Vue, etc.).
Evolución constante: Cada año recibe mejoras con nuevas funciones y herramientas.
Multiplataforma: Sirve para aplicaciones web, móviles, de escritorio e incluso inteligencia artificial.

Desventajas {

> Seguridad: Puede ser vulnerable a ataques
> Compatibilidad: En navegadores antiguos puede no funcionar bien o tener diferencias.
> Rendimiento: Más lento que lenguajes compilados como C++ o Java en tareas muy pesadas.
> Tipado débil: Los errores pueden aparecer en ejecución porque no es fuertemente tipado.
> Dependencia del navegador: En frontend, algunas funciones dependen del soporte del navegador.

Usos principales {

PerfectoDesarrollo web (Frontend):
Crear interactividad en páginas web (0menús, validaciones de formularios, animaciones, sliders, etc.).
Modificar dinámicamente el contenido de una página (DOM).
Mejorar la experiencia del usuario con interfaces dinámicas (por ejemplo, redes sociales o buscadores).

Desarrollo del lado del servidor (Backend):
Con Node.js se pueden crear servidores, manejar bases de datos etc.

Aplicaciones móviles:
Con frameworks como React Native, Ionic o NativeScript se pueden desarrollar apps para Android e iOS.

Aplicaciones de escritorio:
Con Electron.js se crean programas de computadora (como Visual Studio Code o Discord).

Videojuegos:
Se usa en desarrollo de juegos web con motores como Phaser.js o en 3D con Three.js.

Internet de las cosas (IoT):
Permite programar dispositivos inteligentes con frameworks como Johnny-Five.

Inteligencia Artificial y Ciencia de Datos:
Con librerías como TensorFlow.js se hacen modelos de IA directamente en el navegador.

}

HERRAMIENTAS

Cuando hablamos de herramientas que ocupa JavaScript, podemos entenderlo en dos sentidos:

Herramientas propias del lenguaje (lo que JavaScript trae por sí mismo: estructuras, objetos, métodos, APIs del navegador).

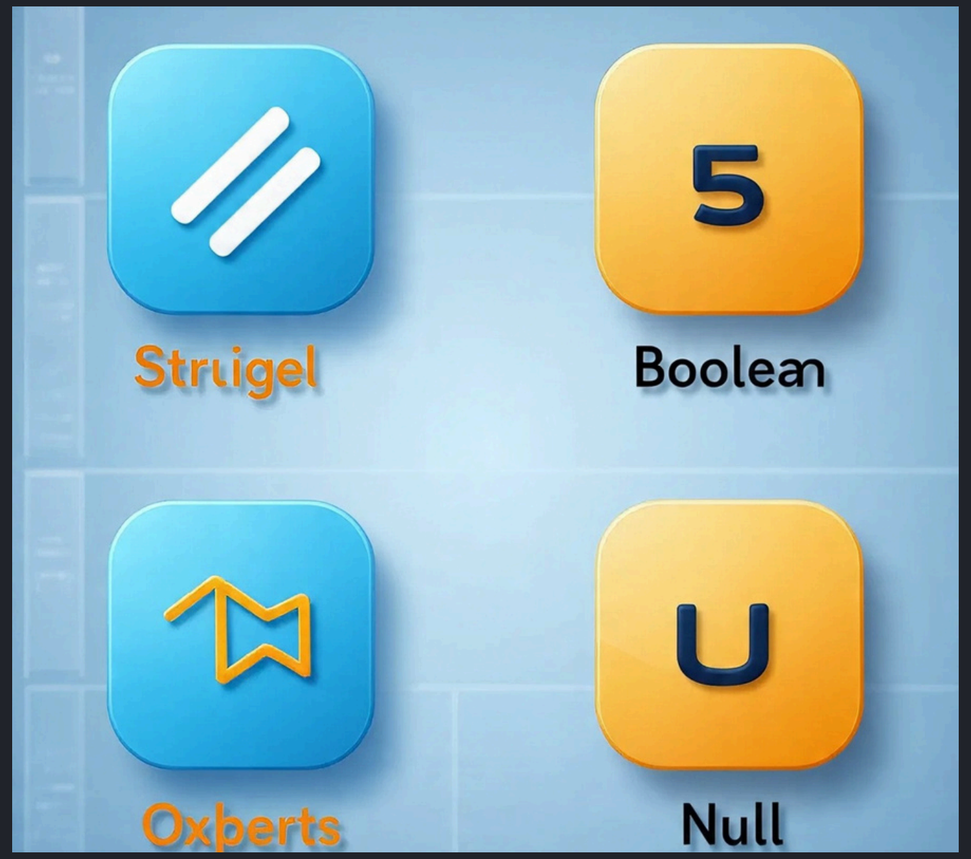
Herramientas externas que se usan para trabajar, desarrollar ymejorar proyectos en JavaScript

Herramientas nativas

Tipos ae datos

Primitivos: Números, cadenas de texto, booleanos, null y undefined
Objetos: Colecciones de propiedades y métodos que pueden ser utilizados para representar datos y comportamientos complejos.

Enteros: let edad = 25;
Decimales: let precio = 19.99;
Notación científica: let distancia 1.2e5;



Estructuras de Control

Condicionales: if/else, switch.
Bucles: for, while, do-while
Funciones: Bloques de código reutilizables que pueden ser llamados desde diferentes partes del programa.

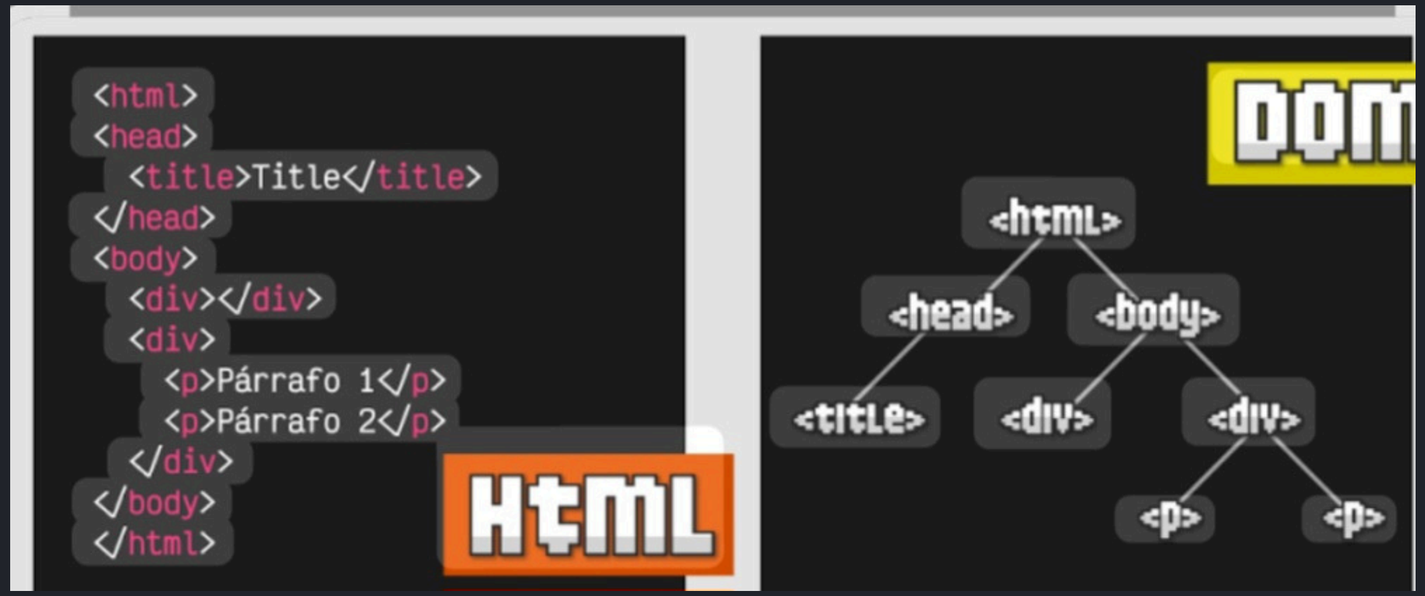
```
let edad = 25;
if (edad >= 18) {
  console.log("Eres mayor de edad");
} else {
  console.log("Eres menor de edad");
}
```

APIs del Navegador

DOM: Document Object Model, permiteinteractuar con los elementos HTML de una página web.

Eventos: Permiten responder a acciones del usuario, como clics o
pulsaciones de teclas.

Fetch API: Permite realizar solicitudes
HTTP para obtener datos de servidores.



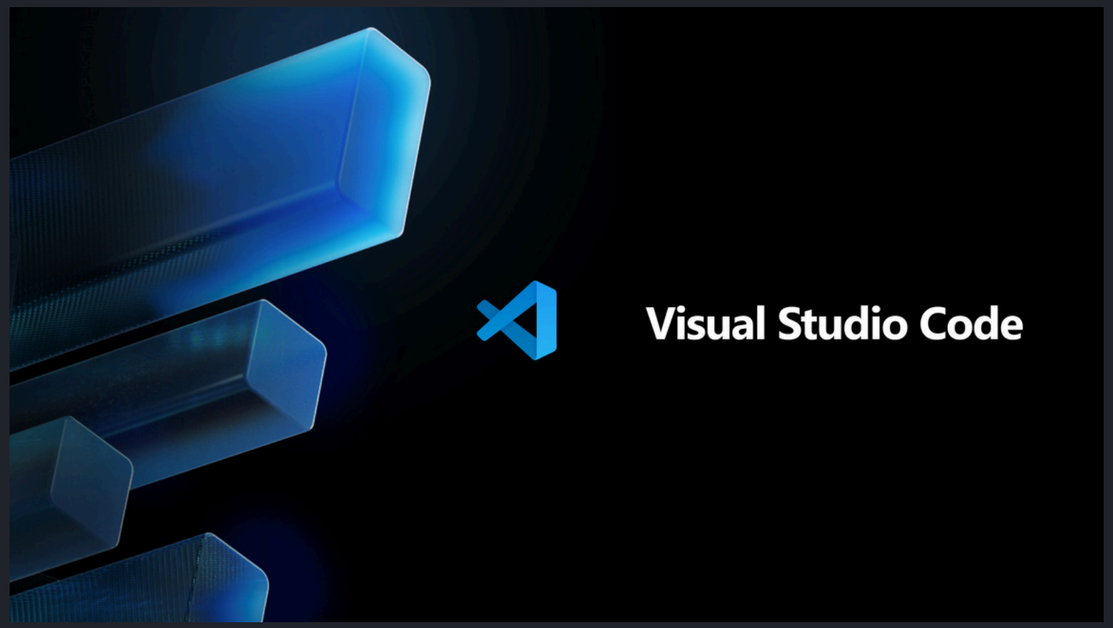
Herramientas externas

Editores de Código

Visual Studio Code: Un editor de código ligero y personalizable con soporte para JavaScript y muchas otras lenguajes

Sublime Text: Un editor de código rápido y flexible con una gran cantidad de plugins y temas disponibles.

Atom: Un editor de código personalizable y extensible con soporte para JavaScript y muchas otras lenguajes.



Entorno de ejecución

Google Chrome: Desarrollado por Google, Chrome es uno de los navegadores más utilizados en todo el
mundo.

Mozilla Firefox: Desarrollado por Mozilla, Firefox es un navegador de código abierto que se enfoca en la privacidad y la seguridad.

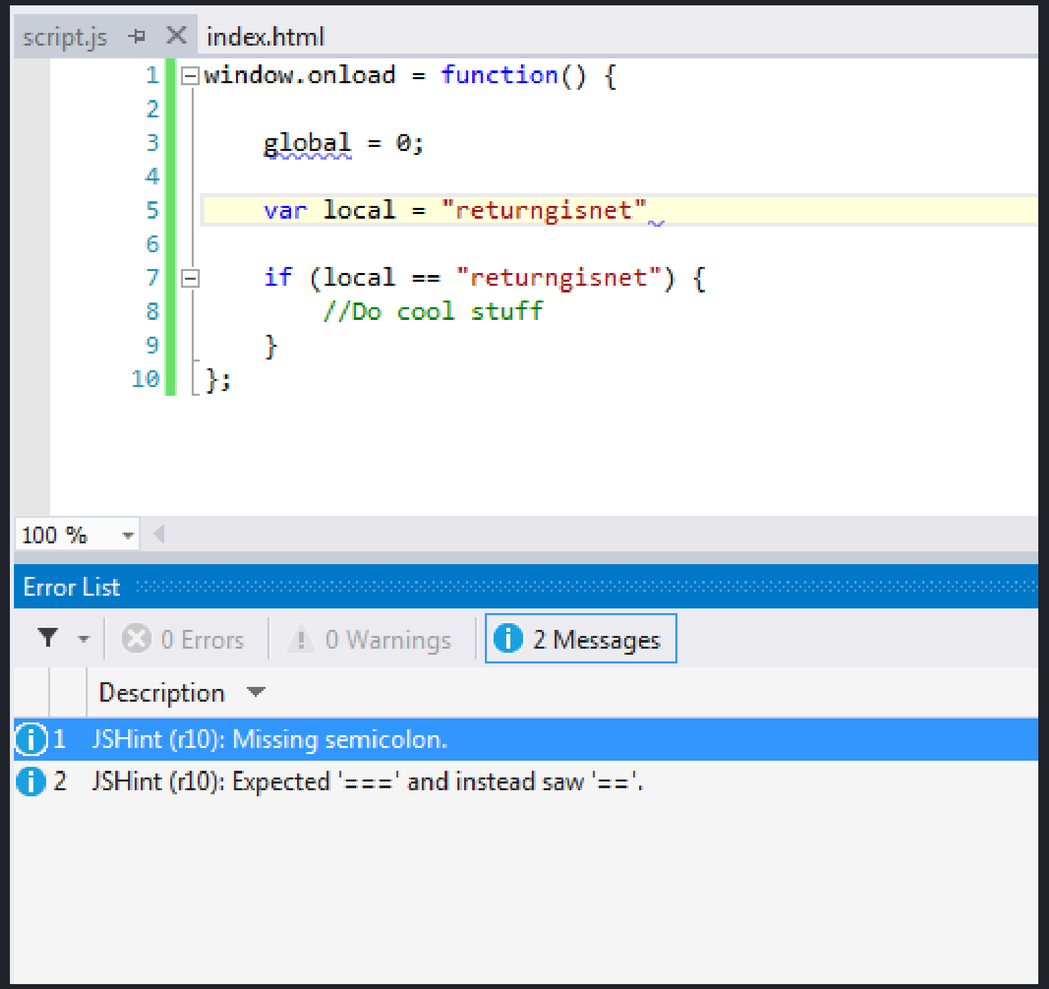
Microsoft Edge: Desarrollado por Microsoft, Edge es un navegador que se enfoca en la velocidad y la seguridad.

Herramientas de Análisis de Código

ESLint: Una herramienta de análisis de código que permite detectar errores y problemas de estilo en el código JavaScript.

JSHint: Una herramienta de análisis de código que permite detectar errores y problemas de estilo en el códigoJavaScript.

CodeCoverage: Una herramienta que permite medir la cobertura de código de las pruebas unitarias y de integración.



```
Gracias {

}
```