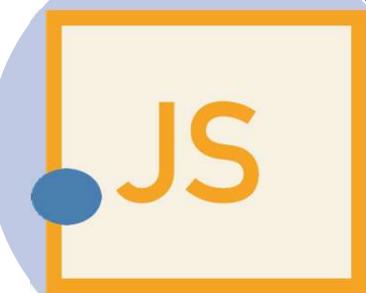


Jorge García Flores

# Desarrollo Web en Entorno Cliente





# Introducción

javascript

Motor V8

# CONTENT

Instalación de  
Software

Angular

TypeScript

Usos de  
JavaScript

Motores javascript

Frameworks  
javascript



# JavaScript

JavaScript, comúnmente abreviado como JS, es un lenguaje de programación de tipo interpretado basado en el estándar ECMAScript, y a pesar de ser débilmente tipado y dinámico, también se define como un lenguaje orientado a objetos.

JavaScript es *case sensitive* o sensible a mayúsculas y minúsculas.

JS toma su mayor esencia del lenguaje C y algunas convenciones propias de Java, aunque Java y JavaScript no guardan relación alguna entre sí.

Las últimas versiones de JS permiten que sea un lenguaje del lado del cliente, así como también un lenguaje del lado del servidor (*Node.js*).



# JavaScript

**JavaScript no es Java.**

La causa de esta similitud en los nombres se debe al origen de JavaScript.

Inicialmente fue desarrollado por la empresa Netscape en 1995 con el nombre de LiveScript. Posteriormente pasó a llamarse JavaScript, quizás tratando de aprovechar que existía un lenguaje de programación de gran popularidad llamado Java y que un nombre similar podía hacer que el nuevo lenguaje fuera atractivo.

Microsoft también adoptó este lenguaje, pero lo llamó JS.



# JavaScript

En los años 90 se utilizaban unos dispositivos llamados módems para conectarse a Internet a una velocidad de 28Kbps (3.500 caracteres por segundo)

Cuando un navegador accedía a una página web y, por ejemplo, el usuario rellenaba un formulario se enviaba al servidor, si el usuario se había olvidado o cometía un error en algún campo, como en el DNI, el servidor le enviaba otra página indicándoselo para que el usuario repitiera el proceso.

¡Esto era LENTO y CARO!

Por este motivo, a Brendan Eich, programador en Netscape, se le ocurrió crear un lenguaje (LiveScript) que corriera en el host cliente, en el navegador, que pudiera detectar estos errores y corregirlos sin la necesidad de enviarlo al servidor.



yahoo\_1997.gif

Tamaño: 20.787 bytes

# JavaScript

La Asociación Europea de Fabricantes de Computadoras (ECMA o *European Computer Manufacturers Association*) pasó a denominarse Ecma International para expresar su alcance internacional.

Desde el año 1997, ECMA establece el estándar para JavaScript DOM (*Document Object Model*) con el fin de evitar incompatibilidades entre los distintos navegadores existentes.

Su primera versión fue ECMA-262 bajo la ISO/IEC 16262, en el año 1997.



[ECMA-262 - Ecma International \(ecma-international.org\)](http://ecma-international.org)

# JavaScript

[https://fullstackseries.com/post\\_series/javascript-moderno/](https://fullstackseries.com/post_series/javascript-moderno/)

<https://www.makeuseof.com/ecmascript-2023-es14-what-to-expect/>

<https://appmaster.io/es/news/javascripts-ecmascript-2023-aprobacion>

Versiones de JavaScript:

- 1996 LiveScript pasó a llamarse JavaScript
- 1997 ES1 (ECMAScript1)
- 2009 ES5 (ECMAScript5) muchas características nuevas, como por ejemplo la declaración de variables con var...
- 2015 ES6/2015 (ECMA2015) que fue la actualización más grande de JavaScript hasta el momento: símbolos, constantes, declaración de variables con *let*... y se estableció que cada lanzamiento de una nueva especificación de JavaScript sería anual.
- ES7/ES2016, ES8/ES2017, ES9/ES2018, ES10/2019, ES11/2020 ES12/2021 ...
- ES14 (2023)



ES5

ES6

Un *polyfill* es un fragmento de código (generalmente JavaScript) que se utiliza para proporcionar una funcionalidad moderna en navegadores antiguos que no lo admiten de forma nativa.

JavaScript

### **ES5:**

- Soportado por todos los navegadores web
- Mayor compatibilidad.

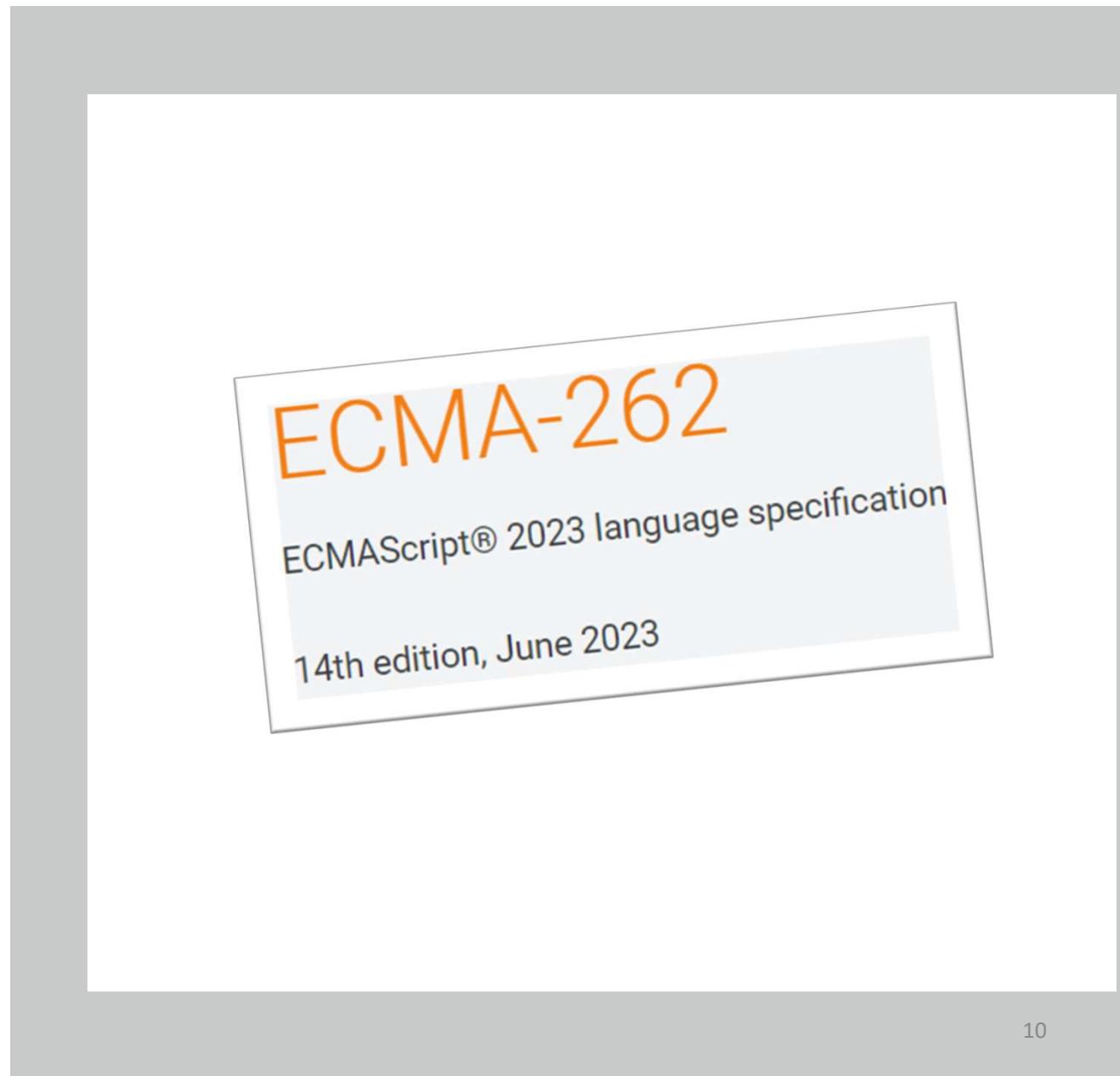
### ***ES6/ES2015, ES7/ES2016, ES8/ES2017, ES9/ES2018,...:***

- Soportado por la mayoría de navegadores modernos.
- Se pueden usar muchas de las características nuevas.
- Se pierde la compatibilidad (sobre todo con navegadores antiguos presentes todavía en ciertos países, como en Latinoamérica).
- Posibilidad de usar polyfills para utilizar características modernas en navegadores antiguos.

# JavaScript

Especificaciones Online:

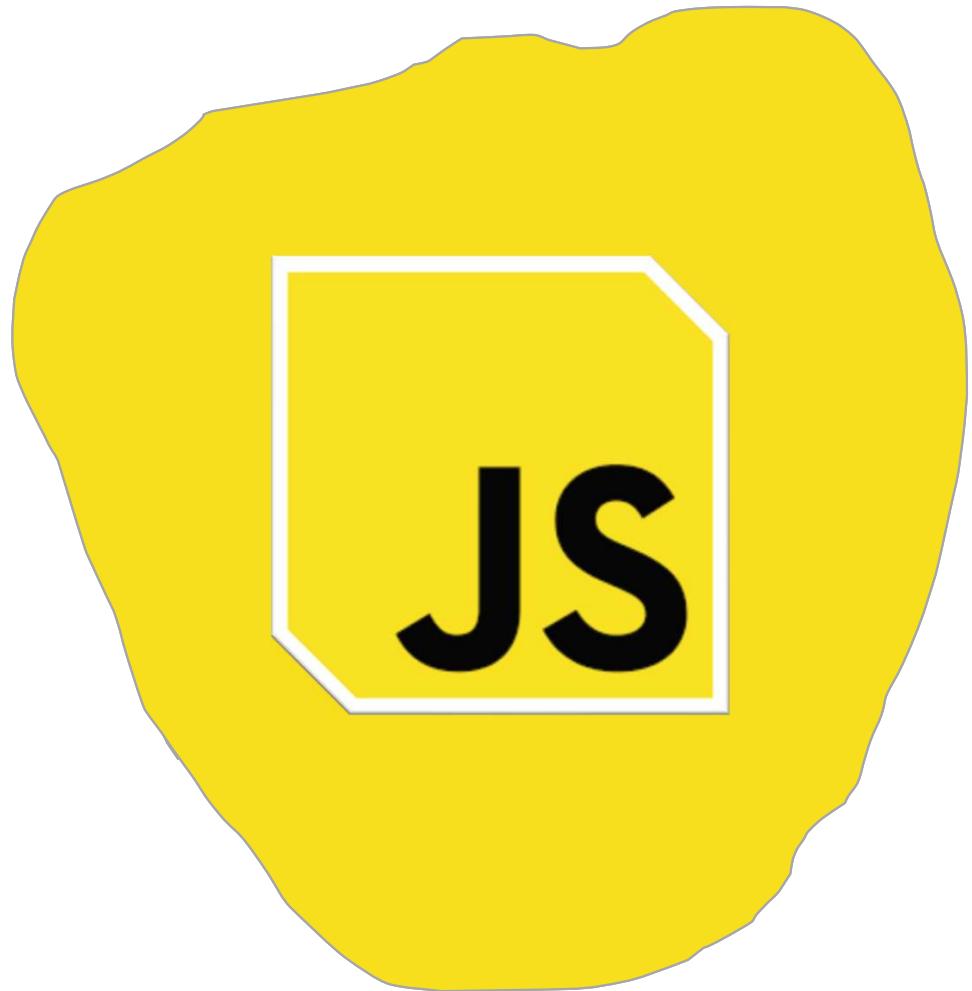
- [ECMA-262 5.1 edition, June 2011](#)
- [ECMA-262, 6th edition, June 2015](#)
- [ECMA-262, 7th edition, June 2016](#)
- [ECMA-262, 8th edition, June 2017](#)
- [ECMA-262, 9th edition, June 2018](#)
- [ECMA-262, 10th edition, June 2019](#)
- [ECMA-262, 11th edition, June 2020](#)
- [ECMA-262, 12th edition, June 2021](#)
- [ECMA-262, 13th edition, June 2022](#)



# JavaScript

Cuando se utiliza JS *embedido* dentro de páginas web, su ejecución se realiza en el ordenador del cliente, dentro del navegador, donde se está mostrando la página.

En este entorno, por seguridad, no es posible acceder de forma normal a ningún recurso local que no sea expresamente permitido por el propio navegador, sólo a las propias propiedades de la página o del navegador en sí.



# JavaScript

Suele ser frecuente utilizar JavaScript para:

- Controlar las ventanas del navegador y su contenido.
- Evitar la dependencia del servidor web para cálculos sencillos.
- Capturar eventos generados por el usuario y responder a ellos.
- Validar los datos que introduce un usuario en un formulario antes de enviarlos al servidor...

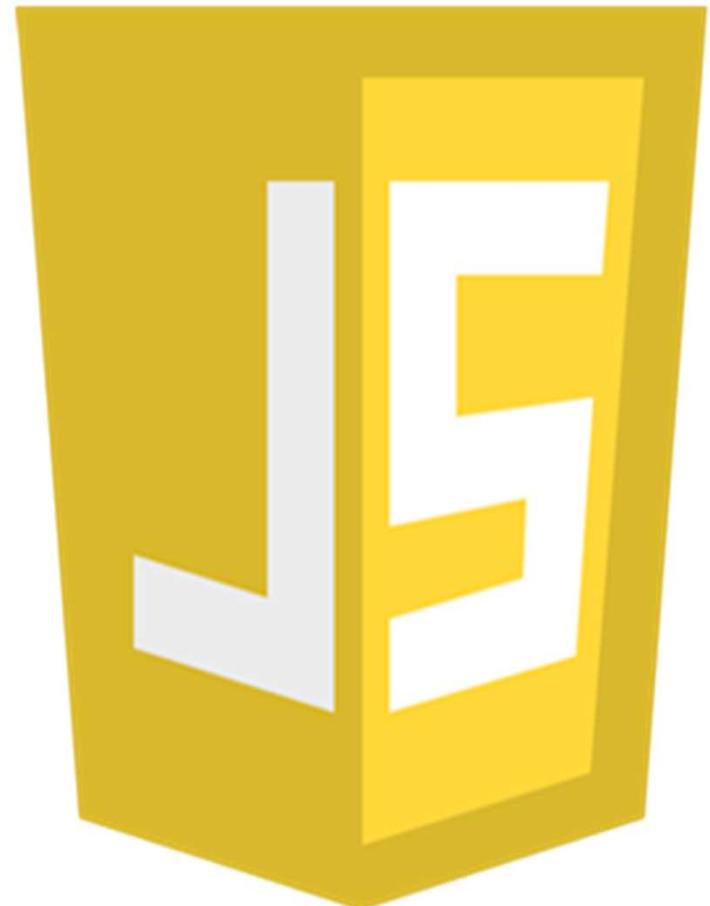


# JavaScript

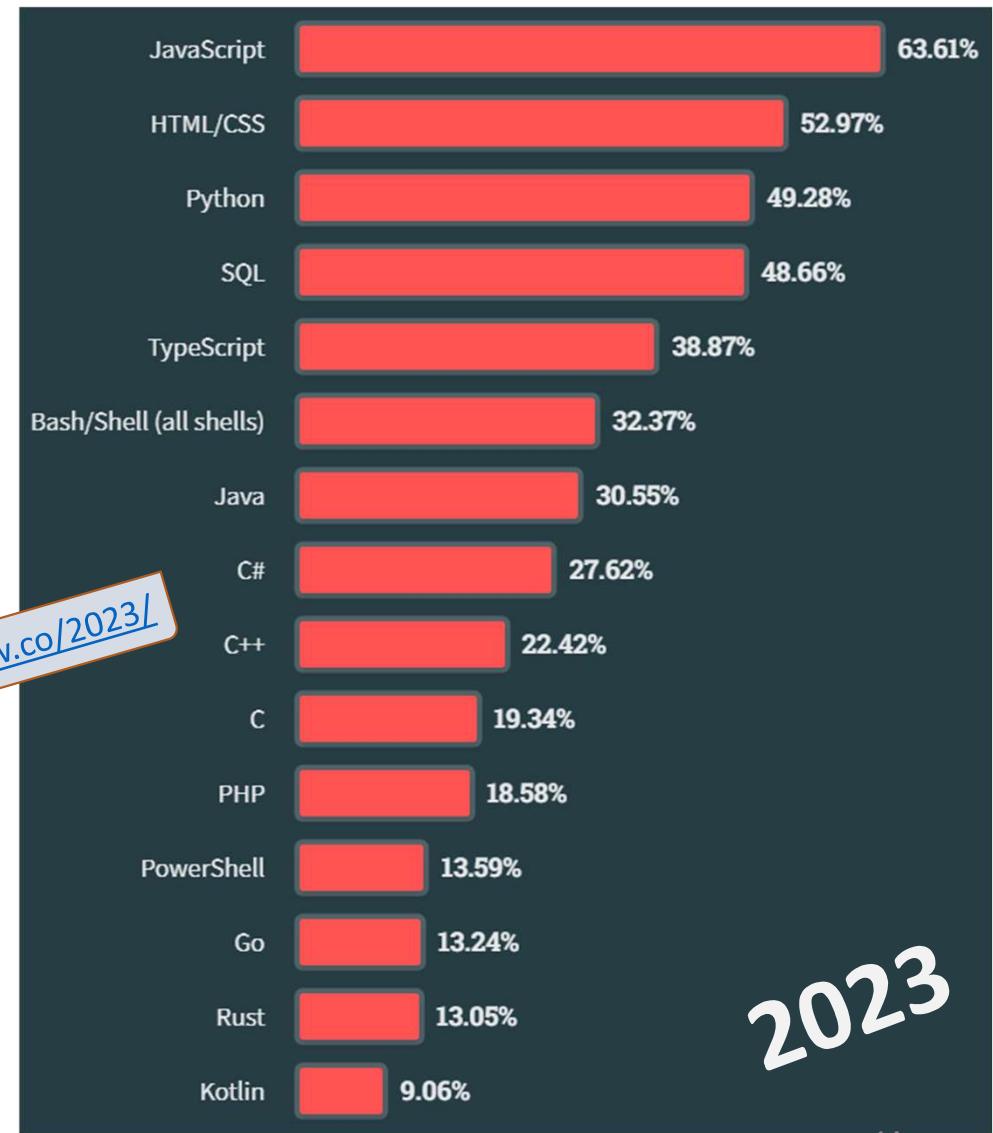
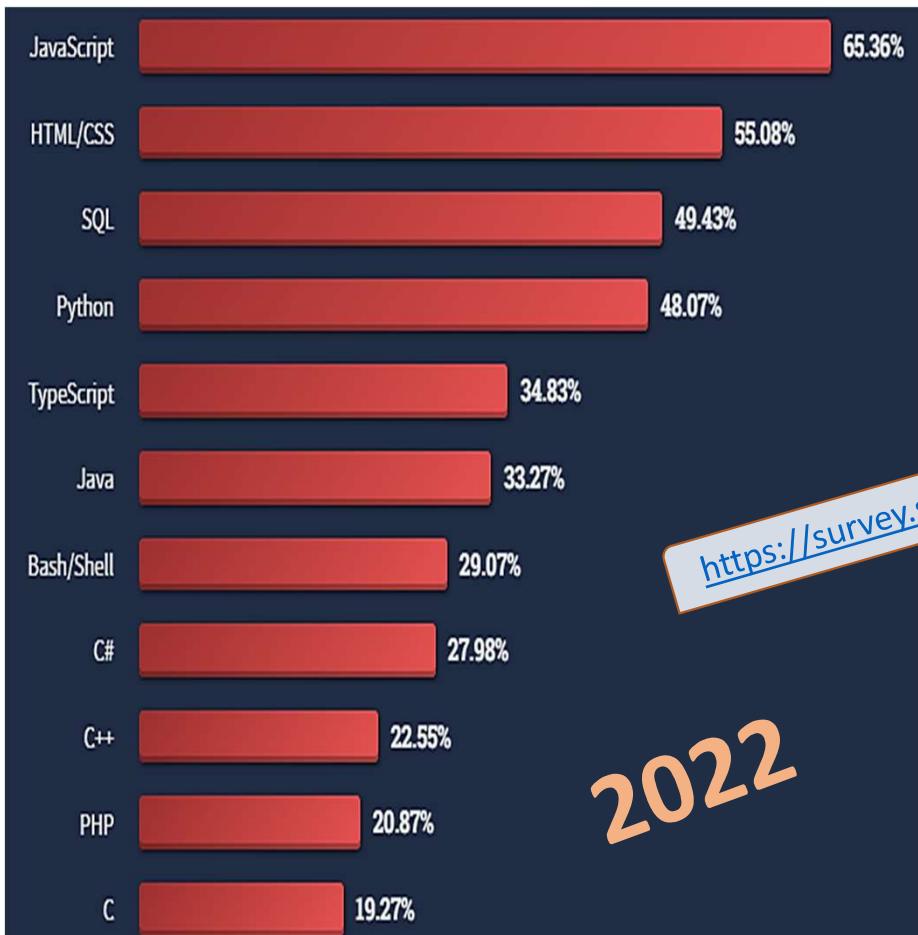
Es un lenguaje de programación de tipo interpretado:

Cuando se accede a una aplicación web desde un navegador, éste descarga los archivos `.js` que contienen el código con las instrucciones a ejecutar.

Es, entonces, cuando el motor JavaScript del navegador interpreta las instrucciones del código y emplea un proceso, en tiempo real, conocido como *Just-In-Time Compiler (JIT Compiler)* en dónde convierte dicho código en algo que sí puede entender el ordenador.



# Programming, scripting, and markup languages



## Motores JavaScript JavaScript Engine

V8 usado navegadores basados en Chromium (Chrome, Microsoft Edge, Opera) y en Node.js y Deno

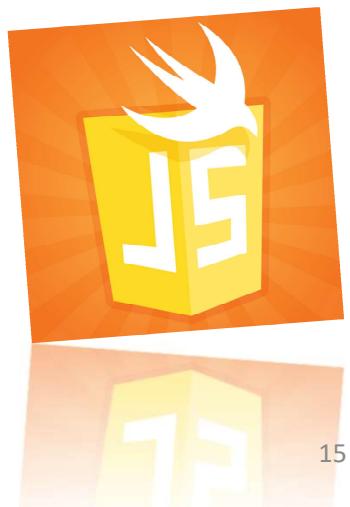


SpiderMonkey (Firefox)



Debido a la gran variedad de navegadores también existe una gran variedad de motores JS, siendo los más importantes V8 empleado en Chrome, SpiderMonkey empleado en Mozilla y JavaScriptCore empleado en Safari.

JavaScriptCore usado en navegadores basados en Webkit (Safari)

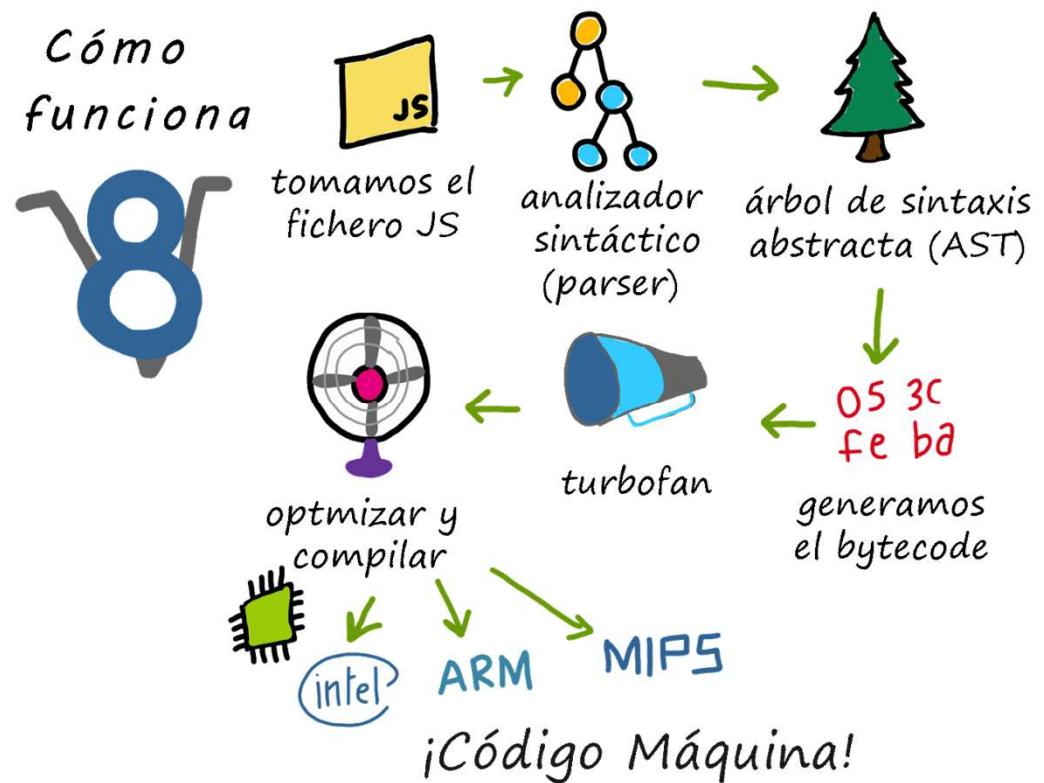


# [ Motor V8 ]

En la actualidad, V8 es uno de los motores líderes de JavaScript debido a su rendimiento, estabilidad y evolución, siendo empleado en importantes navegadores como Chrome y Microsoft Edge, y también se usa en NodeJS y Deno.

Nació en el año 2008 de la mano de Google y está desarrollado en C++.

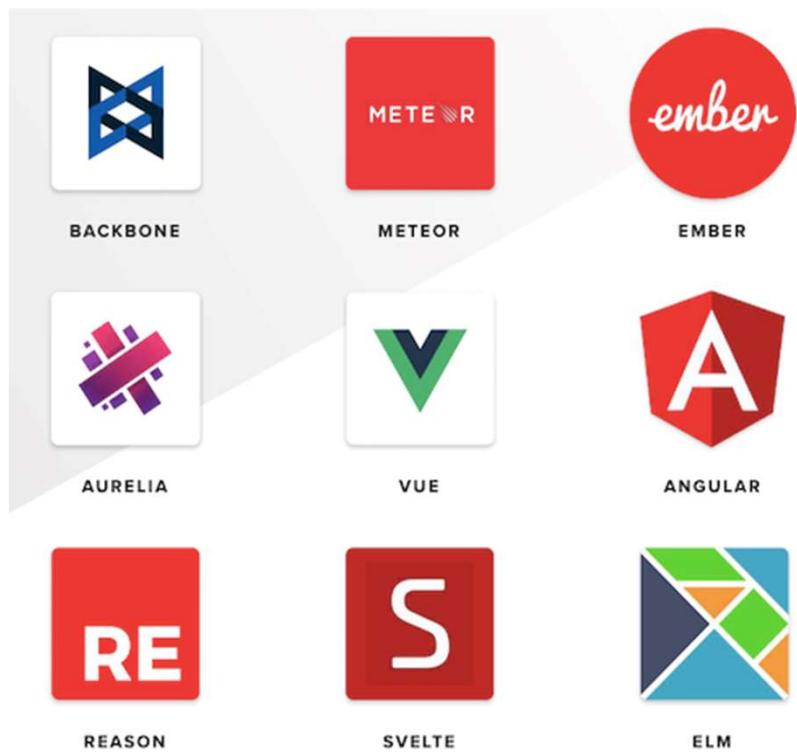
Ejecuta JavaScript empleando una estrategia que combina un intérprete para transformar el código y un compilador para optimizar el resultado.



<https://filisantillan.com/blog/js-engine/>

# Frameworks JavaScript

El término *framework* se emplea en muchos ámbitos del desarrollo de sistemas software.



Existen *frameworks* para el desarrollo de aplicaciones médicas, para el desarrollo de juegos,...

Un framework es una infraestructura para la creación de otros programas porque funciona como base para desarrollar programas futuros de forma rápida.

Se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta.

# Frameworks JavaScript



Un framework contiene librerías de código y módulos ya listos que resumen las tareas de creación de elementos recurrentes en el desarrollo de aplicaciones, a la vez que define una arquitectura para el desarrollo de software.

Uno de los éxitos de los frameworks se debe a que están bien documentados.

La cantidad de documentación que podremos encontrar sobre un framework suele ser enorme y además cuentan con una gran comunidad detrás, respondiendo preguntas y desarrollando nuevas funcionalidades.

Cada framework, como contiene sus propios programas, tiene su modo de funcionar.



UI9WGMOLK

ventajas de los frameworks web



Facilita y agiliza el proceso de desarrollo de aplicaciones Web.

Permite reutilizar código ya existente y promueve buenas prácticas de desarrollo. Evita reinventar la rueda.

Debido a la existencia de múltiples navegadores que se comportan de diferente forma, es necesario escribir códigos diferentes para cada uno. Un framework permite la “abstracción” del navegador y centrarse en la funcionalidad del código.

El código resultante utilizando un framework es más claro y reutilizable por otros programadores.

Están altamente testeados, con lo que las páginas suelen ser más fiables.

## Frameworks JavaScript

Existen decenas de frameworks o implementaciones alternativas (Vue.JS, Ember.JS y Angular entre otros) que utilizan JavaScript como lenguaje y que, a través de diferentes APIs, mejoran la funcionalidad o la creación de aplicaciones con JS.

API o interfaz de programación de aplicaciones, es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

Por ejemplo: usando una API de imágenes no hace falta programar como se lee y se dibuja un fichero JPG simplemente se llama a una o varias funciones de la API.



The Angular logo consists of the word "Angular" in blue lowercase letters enclosed in a yellow rounded rectangle.

Angular, comúnmente llamado Angular 2+ o Angular 2, es un framework de JS desarrollado en TypeScript de código abierto y mantenido por Google.

Se utiliza para crear y mantener aplicaciones web de una sola página (SPA o *Single Page Application*).

Angular separa completamente el *frontend* del *backend* en la aplicación, y evita escribir código repetido, manteniendo todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador).

Consiguiendo desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.



TypeScript es un lenguaje que añade a JS una capa de tipado estático y algunas otras incorporaciones de la Programación Orientada a Objetos (OOP) tradicional.

<https://www.typescriptlang.org/>

Estas características son simplemente para ayudar a trabajar con JavaScript en tiempo de diseño, ya que TypeScript compila todo a JS tradicional.

TypeScript es un superconjunto de JavaScript, que añade tipados estáticos y objetos basados en clases.

Es desarrollado y mantenido por Microsoft, es open-source.

Extiende la sintaxis de JavaScript por lo que cualquier código de JavaScript existente funcionaría perfectamente.



# TypeScript

El tipado se refiere a cómo se declaran los tipos de variables. Estos pueden variar desde tipos simples como *number* y *string* hasta estructuras complejas.

Existen dos categorías en los lenguajes de programación: tipado estático y tipado dinámico.

En lenguajes con tipado estático, el tipo de variable se conoce en el tiempo de compilación, mientras que, en lenguajes con tipado dinámico, el tipo de variable se conoce solamente cuando se ejecuta el programa.



JavaScript no admite tipado estático:

```
<script>
Let a=5;
a="Jorge";
document.write(a);
</script>
```

# TypeScript



El tipado dinámico, generalmente, proporciona una libertad, pero conduce a errores y disminuye la productividad.

Utilizar tipados estáticos en JS permite:

- Evitar errores ocultos.
- Refactorizar el código sin grandes efectos secundarios.

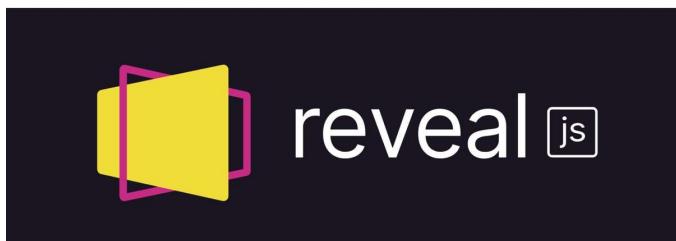
No hay advertencias del editor en archivos JavaScript ;

Este código falla en tiempo de ejecución!

```
function compact(arr) {  
    if (arr.length > 10)  
        return arr.trim(0, 10)  
    return arr  
}
```

# Usos de JavaScript

<https://revealjs.com/>

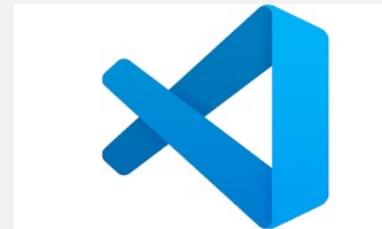


<https://revealjs.com/#/1>

<https://js13kgames.com/games/underrun/index.html>



<https://js13kgames.com/entries/2022>



`</tag>`

## INSTALACIÓN DE SOFTWARE



[Visual Studio Code](#)



[Activitus Bar](#)

[Live Server](#)

*Extensiones Opcionales:*



[vscode-icons](#)

[Material Icon Theme](#)

[Monokai Night Theme](#)

[Auto Close Tag](#)

# Bibliografía

- <https://www.ecma-international.org/>
- <https://www.ecma-international.org/publications-and-standards/standards/ecma-402/>
- <https://developer.mozilla.org/es/docs/Glossary/Polyfill>
- <https://filisantillan.com/blog/js-engine/>
- <https://khru.gitbooks.io/typescript/content/>
- <https://www.typescriptlang.org/>
- [https://fullstackseries.com/post\\_series/javascript-moderno/](https://fullstackseries.com/post_series/javascript-moderno/)
- <https://statefull.medium.com/ember-js-el-framework-8d77b43b10b2/>