

# Javascript

---

## Javascript Introducción

Inicialmente Javascript fue creado para dar vida a las páginas web y ejecutarse solamente en los navegadores.

Pero hoy en día se puede utilizar en muchas areas por ejemplo:

1. Desarrollo Front-End
2. Desarrollo Back-End
3. Aplicaciones de línea de comandos
4. Aplicaciones de escritorio
5. Aplicaciones móviles
6. Cualquier combinación de las anteriores.

## Javascript Front-End (En el navegador)

Son simples archivos de texto con la extensión `.js` y no necesitan compilarse. Son llamados simplemente Scripts pueden ser incluidos dentro del código HTML y corren automáticamente cuando la página se carga. Javascript **en el Navegador** normalmente es usado para:

- Agregar nuevo código/contenido html a una página, cambiar su contenido, modificar los estilos.
- Reaccionar a la interacción del usuario con la página, clicks y movimiento del ratón, entrada de teclado.
- Enviar solicitudes a un servidor (AJAX) para obtener y enviar información.
- Hacer preguntas y mostrar mensajes al usuario

### Javascript y Java

En sus inicios (1995) Javascript se llamó Mocha y después LiveScript, pero como Java era muy popular en esos años, se decidió posicionar este nuevo lenguaje como si fuera un *hermano menor* de Java y se cambió el nombre a Javascript. En 1997 fue adoptado como estándar por la ECMA con el nombre de ECMAScript y poco después por la ISO.

# Javascript

---

## Javascript Especificación y Referencia

### Especificación de Javascript

La especificación ECMA-262 contiene la definición detallada y formal de Javascript, es decir lo define, pero es muy difícil de entender.

<https://www.ecma-international.org/publications/standards/Ecma-262.htm>

### Referencia de Javascript

Existen actualmente en la web muchas referencias del lenguaje pero una de las mas completas es la de Mozilla Developer Network contiene mucha información detallada, además de ejemplos acerca de las funciones, métodos etc.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

- **ECMAScript:** Se refiere a la especificación estándar.
- **Javascript:** Es una de las implementaciones del estándar (como JScript y ActionScript hoy ya obsoletos)

## Javascript Motor

Los navegadores web tienen embebido un motor de Javascript. Este motor es el encargado de leer el script compilarlo y ejecutarlo.

Existen diferentes motores en cada navegador, usualmente hacen **Just in Time Compilation**. Estos motores son generalmente desarrollados y mantenidos por los proveedores de navegadores. Hoy en día Javascript puede correr en cualquier dispositivo que tenga un Motor de Javascript. Por ejemplo el motor V8 de Chrome es parte esencial del entorno de ejecución Node.js

### Just in Time Compilation:

Compilación del código en tiempo de ejecución.

# Javascript

---

## Javascript Supersets

Hoy en día existen varios lenguajes que son transpilados (convertidos) a Javascript. Mediante el uso de diversas herramientas el proceso de transpilación es muy rápido y transparente. Permitiendo a los desarrolladores escribir en otro lenguaje y *auto-convertirlo* a Javascript. Algunos de estos lenguajes son:

- **Type Script (Microsoft):** Agrega tipos de datos
- **Coffee Script:** Tiene una sintaxis mas corta, que permite escribir código mas claro y preciso
- **Dart (Google):** Lenguaje independiente con su propio motor que puede ser transpilado a Javascript

**Compilación:** Convertir código fuente de un lenguaje a otro que está a un nivel más bajo.

**Transpilación:** Convertir código fuente de un lenguaje a otro que está en el mismo nivel de abstracción.