

Lenguajes de script de cliente

1.1. Definición

Los lenguajes de script de cliente son lenguajes de programación que se ejecutan directamente en el navegador web del usuario, en otras palabras, en el lado del cliente. Estos lenguajes permiten a las páginas web interactuar dinámicamente con el usuario, sin necesidad de enviar constantemente solicitudes al servidor. Esto mejora la experiencia del usuario al hacer que las páginas sean más rápidas y reactivas.

En el navegador JavaScript es capaz de:

- Agregar nuevo HTML a la página, cambiar el contenido existente y modificar estilos.
- Reaccionar a las acciones del usuario, ejecutarse con los clics del ratón, movimientos del puntero y al oprimir teclas.
- Enviar solicitudes de red a servidores remotos, descargar y cargar archivos (Tecnologías llamadas AJAX y COMET).
- Obtener y configurar cookies, hacer preguntas al visitante y mostrar mensajes.
- Recordar datos en el lado del cliente con el almacenamiento local ("local storage")

JavaScript se distingue por:

- Completar la integración con HTML y CSS.
- Las cosas simples se hacen de manera simple.
- Soporta la mayoría de los navegadores y se habilita de forma predeterminada.

1.2. Características

- Ejecución en el cliente: Se ejecutan en el navegador del usuario en lugar del servidor.
- Interactividad: Permite modificar dinámicamente el contenido de la página.
- Integración con HTML y CSS: Trabajan en conjunto con HTML y CSS para mejorar la experiencia del usuario.
- Event-driven: Responden a eventos como clics, movimientos del mouse o entrada de datos.
- Asincronismo: Permiten realizar peticiones al servidor sin interrumpir la navegación del usuario (AJAX).
- Independencia del servidor: Reducen la carga en el servidor al ejecutar ciertas funciones en el cliente.

El uso de JavaScript no solo mejora la interactividad de las páginas web, sino que también permite el desarrollo de aplicaciones de página única (SPA) que ofrecen una experiencia similar a la de una aplicación de escritorio o móvil. Esto se logra mediante la carga dinámica de contenido sin necesidad de recargar la página completa, lo que resulta en una navegación más fluida y rápida.

1.3. Sintaxis básica

1. Variables: Las variables son contenedores que almacenan datos. En JavaScript, se definen utilizando las palabras clave como “**var**”, “**let**” o “**const**”.

```
let nombre = "Alex"; // Variable que almacena un nombre
const edad = 19;     // Variable constante, cuyo valor no puede cambiar
```

2. Operadores: Los operadores permiten realizar varias operaciones sobre variables y valores. Los más comunes son:

- Aritméticos: “+”, “-”, “*” y “/”
- Lógicos: **&&** (AND), **||** (OR), **!** (NOT)

```
let suma = 5 + 3; // Resultado: 8
let esAdulto = edad >= 18; // Compara si la edad es mayor o igual a 18
```

3. Condicionales: Las estructuras condicionales permiten ejecutar fragmentos de código basados en la evaluación de una condición. Las más comunes son “**if**”, “**else if**” y “**else**”.

```
if (edad >= 18) { // Si la edad que a introducido es mayor que 18 entonces:
  console.log("Eres adulto"); // Sale el mensaje "Eres adulto"
} else { // En el cual quier otro caso:
  console.log("Eres menor de edad"); // Sale el mensaje "Eres menor de edad"
}
```

4. Funciones: Las funciones son bloques de código reutilizables que realizan una tarea específica. Se definen utilizando la palabra clave “**function**”.

```
function saludar(nombre) { // Crea la función de saludar
  console.log("Hola, " + nombre); // Lo que se va mostrar en la pantalla
}

saludar("Juan"); // Llama a la función con el argumento "Juan"
```

5. Bucles (Loops): Los bucles permiten ejecutar un bloque de código repetidamente. Uno de los bucles más comunes es el **"for"**.

```
for (let i = 0; i < 5; i++) { // Crea la función i que es igual a 0, y mientras sea menor que 5, se le vaya añadiendo 1
  console.log(i); // Imprime los números del 0 al 4
}
```

6. Comentarios: Los comentarios son anotaciones que se utilizan para explicar el código y no afectan a su ejecución. En JavaScript, los comentarios de una sola línea comienzan con **"//"**, y los de varias líneas se encierran entre **"/*"** y **"*/"**.

```
// Este es un comentario de una línea

/* Este es un comentario
   de varias líneas */
```

1.4. Estándares

Debido a que JavaScript es un lenguaje funcional, imperativo y orientado a objetos (sabemos que es más bien orientado a prototipos) tuvo tanto éxito que 1 año después lanzan la versión 1.1, y a la par Microsoft decide adoptarlo para Internet Explorer, realmente creó una copia de JavaScript llamada JScript (esto con el fin de evitar problemas legales); Netscape decide estandarizar el lenguaje de JavaScript así que en 1997 decide enviar su versión al organismo ECMA (European Computer Manufacturers Association)

- ECMAScript (ES): Especificación que define el estándar para JavaScript.

- HTML DOM (Document Object Model): Modelo de objetos que permite la manipulación del contenido HTML y CSS desde el script.

- W3C y WHATWG: Organizaciones que establecen los estándares web relacionados con el scripting en el cliente.

1.5. Ejemplos

Ejemplos de Lenguajes de Script de Cliente:

JavaScript

Es el lenguaje más comúnmente utilizado para scripting del lado del cliente en la web. Se usa para crear interacciones dinámicas en las páginas web, manipulación del DOM, validaciones de formularios, y más.

- **Uso:** Se ejecuta directamente en el navegador del usuario y se integra perfectamente con HTML y CSS para crear aplicaciones de web dinámicas.
- **Ejemplo: JavaScript** se usa en la mayoría de los sitios web interactivos modernos como Google, Facebook y Twitter.

TypeScript

Es la solución a muchos de los problemas de JavaScript, está pensado para el desarrollo de aplicaciones robustas, implementando características en el lenguaje que nos permitan desarrollar herramientas más avanzadas para el desarrollo de aplicaciones.

Es un superset de JavaScript que añade tipado estático. Aunque se escribe y se usa de manera similar a JavaScript, TypeScript es transpilado a JavaScript antes de ejecutarse.

- **Uso:** Se utiliza principalmente en aplicaciones más grandes y complejas, proporcionando ventajas como una mayor seguridad en el tipo de datos.
- **Ejemplo:** TypeScript es utilizado en aplicaciones como Angular (framework para aplicaciones web).

VBScript (Visual Basic Script)

Fue un lenguaje de programación integrado en Internet Explorer competidor de Javascript, además de un lenguaje usado para productos Microsoft diversos como las ASP.

Es un lenguaje de scripting desarrollado por Microsoft y fue comúnmente utilizado en Internet Explorer para crear scripts del lado del cliente.

- **Uso:** Aunque su uso ha disminuido con la adopción de JavaScript y otros lenguajes, **VBScript** fue muy utilizado para automatizar tareas y validaciones en páginas web en la era de los navegadores antiguos.
- **Ejemplo: VBScript** se utilizó principalmente en sitios web que solo eran accesibles desde Internet Explorer.

ActionScript

Actionscript es un lenguaje de programación creado por Adobe, y que se utiliza para la plataforma llamada Adobe Flash Player, que se trata de una plataforma que pertenece a la misma empresa. Este tipo de lenguaje, por un lado, permite programar de una manera más interactiva, y, por otro lado, permite una mayor eficiencia a la hora de generar animaciones de cualquier tipo en la plataforma Flash.

Se utiliza en las aplicaciones desarrolladas con Adobe Flash. Aunque Flash ha sido descontinuado, ActionScript fue muy popular en la creación de aplicaciones multimedia interactivas en la web.

- **Uso: ActionScript** permitía la creación de animaciones y aplicaciones ricas en medios para la web.
- **Ejemplo:** Antes de su discontinuación, **ActionScript** se usaba en juegos y animaciones interactivas en Flash.

CoffeeScript

Es un lenguaje de programación que transcompila a JavaScript. La sintaxis de CoffeeScript es más concisa y expresiva que la de JavaScript, pero finalmente se convierte en JavaScript antes de ejecutarse.

Fue diseñado para hacer que el código JavaScript sea más legible y eficiente, centrándose en la simplicidad y la elegancia.

- **Uso:** Algunos desarrolladores lo usan como una alternativa más limpia y sencilla para escribir código JavaScript.
- **Ejemplo:** CoffeeScript fue utilizado por algunas aplicaciones web en el pasado, aunque su uso ha disminuido frente al crecimiento de JavaScript moderno y TypeScript.

Bibliografía

<https://es.javascript.info/intro>

https://aws.amazon.com/es/what-is/javascript/?utm_source=chatgpt.com

<https://togrowagencia.com/caracteristicas-del-lenguaje-javascript/>

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar_and_types

<https://mauriciogc.medium.com/javascript-historia-estándar-motores-aeb453d9d9e>

<https://codigofacilito.com/articulos/typescript>

<https://desarrolloweb.com/home/vbscript>

<https://www.euroinnova.com/blog/actionscript>

<https://rolecatcher.com/es/habilidades/conocimiento/tecnologias-de-la-informacion-y-las-comunicaciones/desarrollo-y-analisis-de-software-y-aplicaciones/cafescript/>