# JavaScript

JavaScript (JS) es un lenguaje de programación dinámico y multiparadigma que se usa para crear contenido web interactivo y dinámico, tanto en el navegador (lado del cliente) como en el servidor (lado del servidor, como con Node.js). Permite a los desarrolladores añadir funcionalidades como menús, animaciones, validación de formularios y actualización de contenido sin recargar la página.

## ¿Qué hace JavaScript?

* **Interactividad en la web:**

Es el lenguaje principal para añadir características interactivas a los sitios web, como juegos, efectos visuales, y formularios dinámicos.

* **Manipulación del DOM:**

Permite interactuar con el Document Object Model (DOM), que es la estructura de una página web, para modificar su contenido, estilo y estructura.

* **Actualización de contenido (**[**AJAX**](https://www.google.com/search?sca_esv=f2b4c39e3966b02c&sxsrf=AE3TifMSj6GQRLFOBNX1cAHppbKTKe715w%3A1756382901411&q=AJAX&sa=X&ved=2ahUKEwido7aXvK2PAxUoVTABHdeVGYIQxccNegUIlAEQAQ&mstk=AUtExfBuICkEOTR4NK8Qtxr1iNpoim6FfCQuJli37FAgR_yJyevQkf7cPahohdPlihIuJ630xGdQR9PKJ-a2stNHVFNfXkUafZljz_ixQcYkG24zzNhSouR3MrtxwaSj9ilwBijfc9BWSmMc8f3mR80qwi_NWs4Z_P_CLoQrq94FDAP2nCqdNfqGYbNwbeoI1E30nDIEqzPmQf0CPtjDUMak9IS2MlO-xDPKm11mbKzDWevk5YAGvLy0WYW1J4aDf1fLjUkQ5B5Plxk2vOkqOo7aovep&csui=3)**):**

Facilita el envío y recepción de datos al servidor en segundo plano, lo que permite actualizar partes de una página sin recargarla por completo, como en las redes sociales.

* **Desarrollo del lado del servidor:**

Con Node.js, JavaScript puede ejecutarse fuera del navegador, lo que permite crear aplicaciones web completas y del lado del servidor.

## Características clave:

* **Lenguaje de scripting:**

Se ejecuta para realizar tareas específicas, especialmente en el navegador.

* **Tipado dinámico:**

Los tipos de datos se pueden inferir en tiempo de ejecución, lo que le da flexibilidad.

* **Multiparadigma:**

Admite múltiples estilos de programación, incluyendo orientada a objetos y funcional.

## ¿Dónde se usa?

* **Navegadores web:**

Es el lenguaje de scripting más utilizado en los navegadores, con un motor JavaScript integrado en cada uno.

* **Servidores:**

Entornos como Node.js permiten usar JavaScript para el desarrollo de servidores y aplicaciones web.

* **Aplicaciones móviles:**

También se utiliza para el desarrollo de aplicaciones móviles, mejorando la experiencia del usuario.

## Importancia de su habilitación:

Para que las funcionalidades de una página web con JavaScript operen correctamente, este debe estar habilitado en el navegador. Si está deshabilitado, el contenido o la interactividad de la página pueden ser limitados o no estar disponibles.

# Enlazar JavaScript

Para enlazar JavaScript a HTML, utiliza la etiqueta <script> y dos métodos principales: incrustar el código JavaScript directamente dentro de la etiqueta, o usar el atributo src para referenciar un archivo .js externo, ubicando la etiqueta en la sección <head> o <body> de tu HTML para controlar su momento de carga.

**Método 1: Incrustar JavaScript en línea**

1. Abre tu archivo HTML en un editor de código.
2. Agrega la etiqueta <script> en la sección <head> o <body>.
3. Escribe tu código JavaScript dentro de las etiquetas <script> y </script>.

Código:

<!DOCTYPE html>  
 <html>  
 <head>  
 <title>Ejemplo de JavaScript en línea</title>  
 </head>  
 <body>  
 <h1>Mi Página Web</h1>

<script>

alert("¡Hola desde JavaScript en línea!"); // Esto se ejecutará al cargar la página  
 </script>

</body>  
 </html>

**Método 2: Enlazar un archivo JavaScript externo**

1. Crea un archivo nuevo con la extensión .js (por ejemplo, script.js) en la misma carpeta que tu archivo HTML.
2. Escribe tu código JavaScript en este archivo.

JavaScript:

*// En script.js*  
 console.log("¡Hola desde el archivo externo!");

1. En tu archivo HTML, usa la etiqueta <script> con el atributo src.

Código:

<!DOCTYPE html>  
 <html>  
 <head>  
 <title>Ejemplo de JavaScript externo</title>  
 </head>  
 <body>  
 <h1>Mi Página Web</h1>  
 <script src="script.js"></script>  
 </body>  
 </html>

**Ubicación de la etiqueta <script>:**

* **En el <head>:**

El código JavaScript se cargará y ejecutará antes de que el contenido del <body> sea procesado.

* **Al final del <body>:**

El código se ejecutará después de que el contenido HTML haya sido renderizado, lo cual es útil para scripts que interactúan con el DOM y evita retrasos en la carga de la página.

# Variables

En JavaScript, los tipos de variables más importantes son los tipos de datos primitivos como String, Number, BigInt, Boolean, Undefined, Null y Symbol, los cuales almacenan un solo valor. Aparte de estos, existe el tipo de dato no primitivo Object, que puede almacenar colecciones de datos como objetos y funciones.

**Tipos de datos primitivos**

Estos tipos de datos no son objetos y no tienen métodos asociados.

* String: Representa texto, como "Hola Mundo".
* Number: Almacena números, tanto enteros como decimales, como 10 o 3.14.
* BigInt: Se utiliza para números enteros muy grandes que no caben en el tipo Number.
* Boolean: Contiene valores lógicos, true (verdadero) o false (falso).
* Undefined: Indica que una variable ha sido declarada pero aún no tiene un valor asignado.
* Null: Representa la ausencia intencional de un valor, con un único valor null.
* Symbol: Es un valor único e inmutable, utilizado como clave en objetos para asegurar que no haya colisiones.

**Tipos de datos no primitivos**

* Object: Es el único tipo de dato no primitivo y se utiliza para agrupar información. Puede contener otras variables, como objetos o funciones.

**Para declarar variables en JavaScript, se utilizan las palabras clave var, let y const.**

* var:

Declara una variable con alcance de función, pero su uso es limitado en el código moderno.

* let:

Permite declarar variables con alcance de bloque y reasignarles un valor en el futuro.

* const:

Declara una constante, una variable cuyo valor no se puede reasignar después de su inicialización.

# Prompt

En JavaScript, prompt() es un método del objeto Window que muestra una ventana emergente (o diálogo) con un mensaje y un campo de entrada para que el usuario ingrese texto. Este método es útil para solicitar información sencilla del usuario, como su nombre o edad, y devuelve el texto ingresado como una cadena de texto o null si el usuario cancela la operación.

**Sintaxis:**

La sintaxis general del método prompt() es la siguiente:

let respuesta = prompt(mensaje, valorPorDefecto);

* mensaje: Un string opcional que se muestra al usuario en el diálogo.
* valorPorDefecto: Un string opcional que aparece en el campo de entrada del diálogo.

**Cómo funciona:**

1. **Muestra un diálogo:**

Al ejecutar prompt(), se abre una ventana modal que detiene la ejecución del script.

1. **Solicita información:**

El usuario ve el mensaje y un campo de texto con el valorPorDefecto (si se proporciona).

1. **Recopila datos:**

El usuario puede escribir su respuesta, usar el valor predeterminado o cancelar.

1. **Devuelve el resultado:**

Si el usuario hace clic en "Aceptar", la función devuelve el texto ingresado como una cadena de texto.

Si el usuario hace clic en "Cancelar" o presiona la tecla Esc, la función devuelve null.

**Ejemplo práctico:**

let nombre = prompt("Por favor, introduce tu nombre:", "Anónimo");  
alert("Hola, " + nombre + "!");

* Este código primero mostrará un prompt pidiendo el nombre, con "Anónimo" como valor por defecto.
* Luego, el nombre que el usuario introduzca se guardará en la variable nombre.
* Finalmente, se mostrará una alerta con el mensaje "Hola, [nombre introducido]".