

JAVASCRIPT

El dinamismo de la web

The logo consists of the letters 'JS' in a bold, dark blue, sans-serif font. The 'J' and 'S' are connected. This logo is centered within a light yellow square, which is itself centered on the overall yellow background.

JS

¿Qué es Javascript?

Es un **lenguaje de programación** creado en el año 1995 y basado en el estándar **ECMAScript** quien **determina** cómo los navegadores deben **interpretar este lenguaje**.

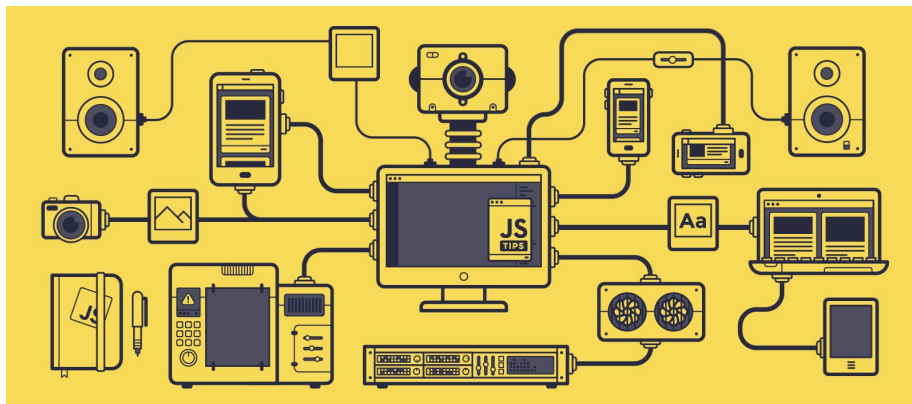
A **Javascript** se le denomina *lenguaje "del lado del cliente"* porque se ejecuta en **contexto del navegador** (cliente web) a diferencia de otros lenguajes que **corren en el servidor**.

¿Cómo funciona Javascript?

Javascript es un lenguaje **orientado a prototipos, multiparadigma e interpretado**.

Esto quiere decir que **podemos usar** programación **funcional**, **orientada a objetos** o **imperativa**.

Además, al ser un **lenguaje interpretado** nuestro código será leído y procesado en tiempo de ejecución a diferencia de los **lenguajes compilados** que leen todo el código antes de **comenzar a correr el programa**.



¿Para qué se usa?

JavaScript en el navegador puede hacer todo lo relacionado con la **manipulación del documento web, la interacción con el usuario y el servidor**.

- ▶ **Cambiar** todo el **contenido** de un documento web (tipo de letra, colores, animaciones, etc.)
- ▶ **Enviar información** a través de la red a servidores remotos, descargar archivos.
- ▶ **Almacenamiento local en el navegador** (recuperar, almacenar información durante la ejecución y visualización del documento web).

¿Para qué NO se usa?

JavaScript **NO** puede acceder a las **circuitos integrados de una computadora** tales como:

- ▶ Disco Duro (Acceso a eliminar información, modificar o leer).
- ▶ Acceso a la memoria **RAM, ROM**.
- ▶ Acceso a la **tarjeta de RED o Procesadores**.
- ▶ **Trabajar del lado del servidor.**

El objetivo de JavaScript en el navegador solo **se limita** al uso exclusivo de todo lo que un **documento web** te puede brindar.

Vincular nuestro Javascript

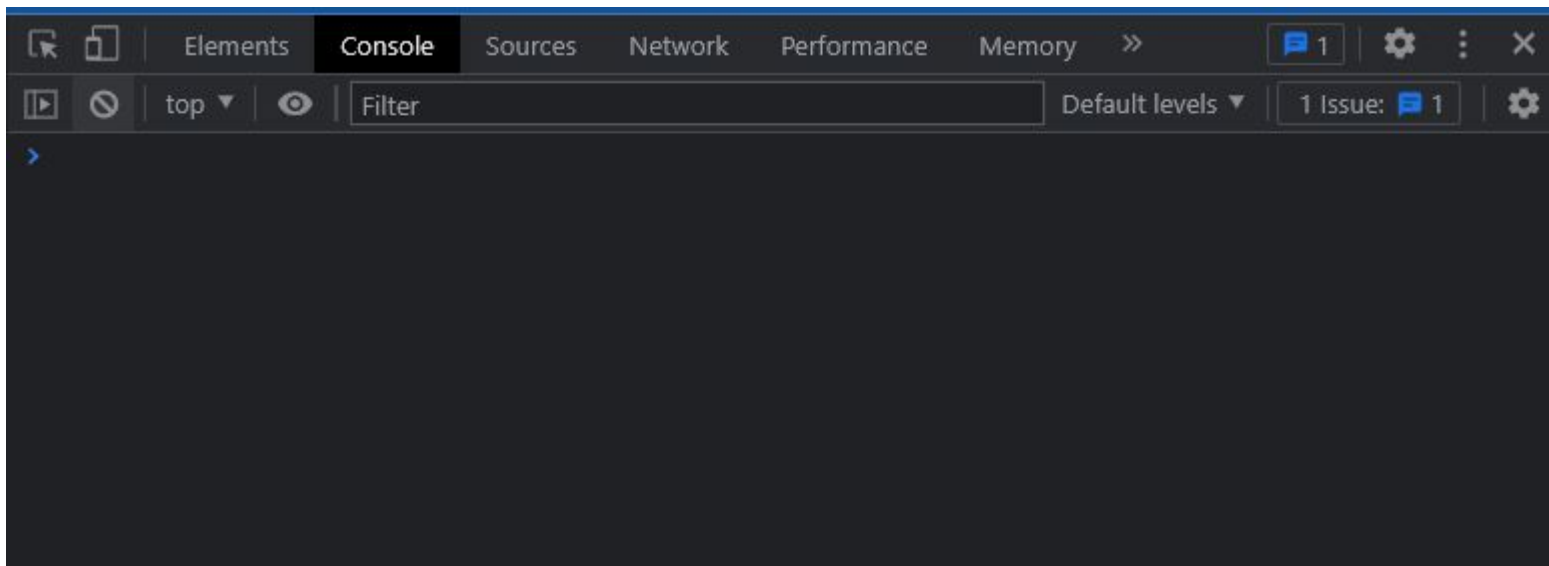
Hay **dos formas** de incluir nuestro código JavaScript en nuestro documento HTML:

Interna: dentro de una etiqueta `<script></script>` antes del cierre del `</body>` en nuestro html.

Externa: en un archivo con extensión .js vinculado a través de una etiqueta `<script src="index.js"></script>` usando el **atributo source** para indicar la ruta al archivo.

¿Cómo vemos los resultados de nuestro código?

Para esto usaremos **la consola del navegador** a la cual podemos acceder presionando **f12**, ctrl + Mayús + J o simplemente **haciendo click derecho** y presionando la opción “inspeccionar”.



Entrada y Salida de datos

Estas **funciones** las utilizamos para **tomar datos de entrada** del usuario o para **representar datos de salida** sin entrometer directamente el HTML.

Hacen uso de mensajes desplegados y la consola del navegador.

Entrada y salida de información.

prompt(): despliega un mensaje en la ventana del navegador con una casilla para ingresar un valor. El **valor ingresado** será tomado como un string.

alert(): despliega un mensaje en la ventana del navegador con el texto que reciba por parámetro.

console.log(): envía lo que recibe por parámetro a la consola del navegador.

```
1 let entrada = prompt('Ingresa tu nombre: ');
2
3 alert(entrada);
4
5 console.log(entrada);
```

Esta página dice
Ingresa tu nombre:

Aceptar Cancelar

Esta página dice
Pepe
Aceptar

Variables

Una variable es un espacio en memoria reservado para alojar información de nuestro programa durante la ejecución del mismo.

Necesitan ser declaradas al momento de escribir nuestro programa y asignadas a un nombre único que pueda identificarlas.

Variables

Para **declarar una variable** debemos utilizar las palabras reservadas `var` o `let` seguidas por el **nombre** que deseamos asignarle.

Existen ciertas restricciones sobre los nombres o caracteres a utilizar:

- El nombre debe contener solo letras, dígitos o los símbolos \$ o _
- El **primer carácter** `no` debe ser un número.
- **No** debe ser una palabra reservada del lenguaje.

```
var miVariable = 'Hola Mundo';  
  
let otrVariable = 50;
```

Nuestras variables son mutables ya que pueden cambiar con el tiempo a diferencia de las **constantes** que `no` se les puede reasignar un valor.

```
const PI = 3.14;
```

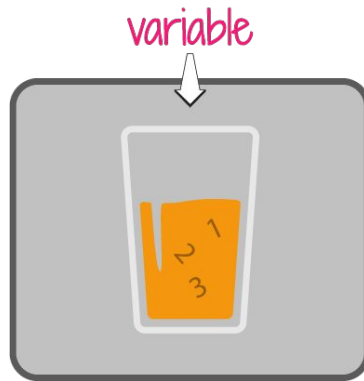
Diferencia entre Var y Let

VAR

Pertenecen **al ámbito o scope global** de nuestro documento, por lo que **pueden** ser **accedidas** y **reasignadas** desde cualquier lugar del mismo.

El uso de ésta, puede dar resultados inesperados, por eso, hay que tener cuidado de cómo se usa.

```
var nombreVariable;  
var a;  
var b;
```



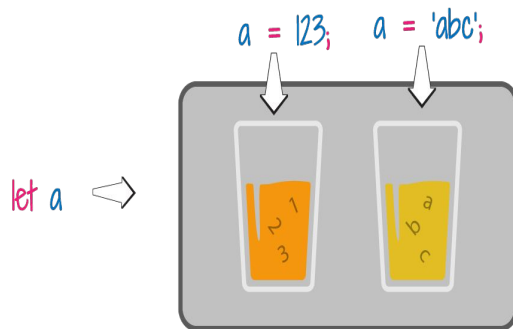
Diferencia entre Var y Let

LET

El **alcance** de estas variables **es local**, solo pueden ser **accedidas** dentro del bloque donde se definen.

También, permiten que su valor pueda ser reasignado.

```
let nombreVariable = 'texto';  
let a = 'abc';  
a = 123;  
let b = 1;  
b = 5;
```



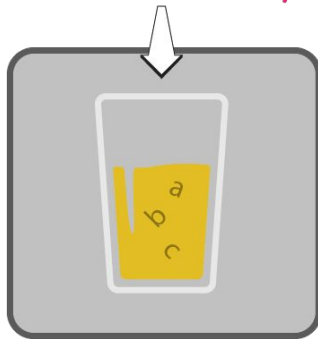
Las constantes

CONST

Solo pueden ser accedidas dentro del bloque donde están definidas, pero **no permite** que su **valor** sea **reasignado**, es decir, la **variable** se vuelve **inmutable**.

```
const nombreVariable = 'texto';  
const a = 'Hola Mundo';  
const b = 'abc';  
const c = 123;
```

const b = 'abc';



Tipos de Datos

Javascript es un **lenguaje débilmente tipado** o de tipado “dinámico” donde una variable **no estará atada** a un **mismo tipo de dato** como sucede en otros lenguajes, si no que **podemos cambiarlo** durante la **ejecución** de nuestro programa.

En Javascript ***existen diferentes tipos de datos*** que podemos utilizar, vamos a conocerlos...

Tipos de datos

string: secuencia de caracteres que representan un valor. (**cadena de texto**)

number: valor **numérico**, entero o decimal.

boolean: valores **true** o **false**.

null: valor **nulo**.

undefined: valor **sin definir**.

symbol: tipo de dato cuyos casos son **únicos e inmutables**.

object: colección de datos en un conjunto de **propiedad/valor**.



```
1 let cadena = 'Hola Mundo';
2 let numero = 23;
3 let booleano = true;
4 let nulo = null;
5 let indefinido = undefined;
6 let symbol = Symbol()
7 let objeto = {
8     propiedad: 'valor'
9 }
```