<https://es.wikipedia.org/wiki/JavaScript>

**JavaScript** (abreviado comúnmente **JS**) es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)), dialecto del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript). Se define como [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos),[2](https://es.wikipedia.org/wiki/JavaScript#cite_note-2)​ [basado en prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos), [imperativo](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_imperativa&action=edit&redlink=1), débilmente tipado(al declarar variables no hace falta indicar el tipo) y dinámico(poder ponerle varios tipos).

<https://es.wikipedia.org/wiki/Document_Object_Model#:~:text=Document%20Object%20Model%20o%20DOM,sobre%20c%C3%B3mo%20pueden%20combinarse%20dichos>

***Document Object Model*** o **DOM** ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos [HTML](https://es.wikipedia.org/wiki/HTML), [XHTML](https://es.wikipedia.org/wiki/XHTML) y [XML](https://es.wikipedia.org/wiki/Extensible_Markup_Language),[1](https://es.wikipedia.org/wiki/Document_Object_Model#cite_note-1)​ un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

<https://www.w3schools.com/js/js_datatypes.asp>

Tipos de datos de JavaScript

Las variables de JavaScript pueden contener muchos **tipos de datos** : números, cadenas, objetos y más:

var length = 16;                               // Number  
var lastName = "Johnson";                      // String  
var x = {firstName:"John", lastName:"Doe"};    // Object

## El concepto de tipos de datos

En programación, los tipos de datos son un concepto importante.

Para poder operar con variables, es importante saber algo sobre el tipo.

Sin tipos de datos, una computadora no puede resolver esto de manera segura:

var x = 16 + "Volvo";

¿Tiene algún sentido agregar "Volvo" a dieciséis? ¿Producirá un error o producirá un resultado?

JavaScript tratará el ejemplo anterior como:

var x = "16" + "Volvo";

Al agregar un número y una cadena, JavaScript tratará el número como una cadena.

### **Ejemplo**

var x = 16 + "Volvo";

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_addstring)

### **Ejemplo**

var x = "Volvo" + 16;

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_addstring2)

JavaScript evalúa expresiones de izquierda a derecha. Diferentes secuencias pueden producir diferentes resultados:

### **JavaScript:**

var x = 16 + 4 + "Volvo";

Resultado:

20Volvo

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_addstrings_1)

### **JavaScript:**

var x = "Volvo" + 16 + 4;

Resultado:

Volvo164

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_addstrings_2)

En el primer ejemplo, JavaScript trata 16 y 4 como números, hasta que llega a "Volvo".

En el segundo ejemplo, dado que el primer operando es una cadena, todos los operandos se tratan como cadenas.

## Los tipos de JavaScript son dinámicos: los principales son:textos, numéricos y booleanos.

JavaScript tiene tipos dinámicos. Esto significa que la misma variable se puede usar para contener diferentes tipos de datos:

### **Ejemplo**

var x;           // Now x is undefined  
x = 5;           // Now x is a Number  
x = "John";      // Now x is a String

## Cadenas de JavaScript

Una cadena (o una cadena de texto) es una serie de caracteres como "John Doe".

Las cadenas se escriben con comillas. Puede utilizar comillas simples o dobles:

### **Ejemplo**

var carName1 = "Volvo XC60";   // Using double quotes  
var carName2 = 'Volvo XC60';   // Using single quotes

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_string_quotes)

Puede usar comillas dentro de una cadena, siempre que no coincidan con las comillas que rodean la cadena:

### **Ejemplo**

var answer1 = "It's alright";             // Single quote inside double quotes  
var answer2 = "He is called 'Johnny'";    // Single quotes inside double quotes  
var answer3 = 'He is called "Johnny"';    // Double quotes inside single quotes

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_string)

Aprenderá más sobre cadenas más adelante en este tutorial.

## Números de JavaScript

JavaScript tiene solo un tipo de números.

Los números se pueden escribir con o sin decimales(se utilizá puntos . no comas):

### **Ejemplo**

var x1 = 34.00;     // Written with decimals  
var x2 = 34;        // Written without decimals

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_numbers)

Los números extra grandes o extra pequeños se pueden escribir con notación científica (exponencial):

### **Ejemplo**

var y = 123e5;      // 12300000  
var z = 123e-5;     // 0.00123

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_numbers_large)

Aprenderá más sobre los números más adelante en este tutorial.

## Booleanos de JavaScript

Los booleanos solo pueden tener dos valores: trueo false.

### **Ejemplo**

var x = 5;  
var y = 5;  
var z = 6;  
(x == y)       // Returns true  
(x == z)       // Returns false

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_bolean)

Los booleanos se utilizan a menudo en pruebas condicionales.

Aprenderá más sobre las pruebas condicionales más adelante en este tutorial.

## Matrices de JavaScript

Las matrices de JavaScript se escriben entre corchetes.

Los elementos de la matriz están separados por comas.

El siguiente código declara (crea) una matriz llamada cars, que contiene tres elementos (nombres de automóviles):

### **Ejemplo**

var cars = ["Saab", "Volvo", "BMW"];

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_array)

Los índices de matriz están basados ​​en cero, lo que significa que el primer elemento es [0], el segundo es [1] y así sucesivamente.

Aprenderá más sobre matrices más adelante en este tutorial.

## Objetos JavaScript

Los objetos de JavaScript se escriben con llaves {}.

Las propiedades de los objetos se escriben como pares de nombre: valor, separados por comas.

### **Ejemplo**

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_object)

El objeto (persona) del ejemplo anterior tiene 4 propiedades: nombre, apellido, edad y color de ojos.

Aprenderá más sobre los objetos más adelante en este tutorial.

## El tipo de operador

Puede utilizar el typeofoperador de JavaScript para encontrar el tipo de una variable de JavaScript.

El typeofoperador devuelve el tipo de variable o expresión:

### **Ejemplo**

typeof ""             // Returns "string"  
typeof "John"         // Returns "string"  
typeof "John Doe"     // Returns "string"

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_typeof_string)

### **Ejemplo**

typeof 0              // Returns "number"  
typeof 314            // Returns "number"  
typeof 3.14           // Returns "number"  
typeof (3)            // Returns "number"  
typeof (3 + 4)        // Returns "number"

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_typeof_number)

## Indefinido

En JavaScript, una variable sin valor tiene el valor undefined. El tipo también lo es undefined.

### **Ejemplo**

var car;    // Value is undefined, type is undefined

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_undefined)

Cualquier variable puede vaciarse estableciendo el valor en undefined. El tipo también lo será undefined.

### **Ejemplo**

car = undefined;    // Value is undefined, type is undefined

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_undefined_2)

## Valores vacíos

Un valor vacío no tiene nada que ver undefined.

Una cadena vacía tiene tanto un valor legal como un tipo.

### **Ejemplo**

var car = "";    // The value is "", the typeof is "string"

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_empty)

## Nulo

En JavaScript nulles "nada". Se supone que es algo que no existe.

Desafortunadamente, en JavaScript, el tipo de datos de nulles un objeto.

Puede considerarlo un error en JavaScript que typeof nulles un objeto. Debería serlo null.

Puede vaciar un objeto configurándolo en null:

### **Ejemplo**

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
person = null;    // Now value is null, but type is still an object

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_null)

También puede vaciar un objeto configurándolo en undefined:

### **Ejemplo**

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
person = undefined;   // Now both value and type is undefined

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_undefined_1)

## Diferencia entre indefinido y nulo

undefinedy nullson iguales en valor pero diferentes en tipo:

typeof undefined           // undefined  
typeof null                // object  
  
null === undefined         // false  
null == undefined          // true

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_undefined_3)

## Datos primitivos

Un valor de datos primitivo es un valor de datos simple y único sin propiedades ni métodos adicionales.

El typeofoperador puede devolver uno de estos tipos primitivos:

* string
* number
* boolean
* undefined

### **Ejemplo**

typeof "John"              // Returns "string"  
typeof 3.14                // Returns "number"  
typeof true                // Returns "boolean"  
typeof false               // Returns "boolean"  
typeof x                   // Returns "undefined" (if x has no value)

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_typeof_primitive)

## Datos complejos

El typeofoperador puede devolver uno de dos tipos complejos:

* function
* object

El typeofoperador devuelve "objeto" para objetos, matrices y nulos.

El typeofoperador no devuelve "objeto" para las funciones.

### **Ejemplo**

typeof {name:'John', age:34} // Returns "object"  
typeof [1,2,3,4]             // Returns "object" (not "array", see note below)  
typeof null                  // Returns "object"  
typeof function myFunc(){}   // Returns "function"

[Inténtalo tú mismo "](https://www.w3schools.com/js/tryit.asp?filename=tryjs_datatypes_typeof_complex)

<https://www.w3schools.com/jsref/met_document_queryselector.asp>

<https://www.w3schools.com/jsref/event_key_keycode.asp>

<https://www.w3schools.com/js/js_string_methods.asp>

<https://lenguajejs.com/javascript/dom/crear-elementos-dom/>

<https://www.w3schools.com/js/js_operators.asp>

https://www.w3schools.com/js/js\_arithmetic.asp

<https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=794:operadores-aritmeticos-javascript-mod-o-resto-de-division-precedencia-concatenar-strings-cu01116e&catid=78&Itemid=206>

https://www.w3schools.com/js/js\_array\_methods.asp

<https://www.w3schools.com/js/js_objects.asp>

<https://www.w3schools.com/js/js_classes.asp>

<https://es.wikipedia.org/wiki/JSON#:~:text=JSON%20(acr%C3%B3nimo%20de%20JavaScript%20Object,para%20el%20intercambio%20de%20datos>.

<https://www.w3schools.com/tags/att_a_download.asp>

<https://www.w3schools.com/js/js_arrow_function.asp>

<https://www.w3schools.com/js/js_api_fetch.asp>

<http://sigdeletras.com/2020/crear-una-fake-reat-api-con-json-server-copy/>

https://programandoointentandolo.com/2017/10/reset-css-normalizar-estilos.html

<https://www.w3schools.com/jsref/event_buttons.asp>

<https://ed.team/blog/las-variables-css-son-una-realidad-y-te-explico-como-usarlas>

https://www.w3schools.com/jsref/prop\_geo\_position.asp