



La especificación ECMAScript

A finales de 1996 **Netscape Communications Corporation** traspasó la responsabilidad de definir y estandarizar JavaScript a **Ecma International**, una organización internacional para el desarrollo de estándares en el sector de tecnologías de la información y comunicación. Para junio de 1997, JavaScript fue oficialmente adoptado como un estándar ECMA, y publicado bajo el nombre de **ECMAScript**.

Este estándar fue desarrollado por un grupo de trabajo que tenía como objetivo la definición de la sintaxis del lenguaje, de su semántica, así como de las librerías y tecnologías suplementarias que facilitan el desarrollo con este lenguaje.

De esta manera, ECMAScript se convirtió en la especificación que define cómo debe funcionar y comportarse JavaScript. Dicha especificación es adoptada por las diferentes empresas o entidades que desarrollan los navegadores web para que estos reconozcan JavaScript.

Obviamente, los navegadores web intentan cumplir la especificación ECMAScript en su totalidad, aunque no siempre es posible, debido a que pueden existir características que no funcionan en un navegador, pero en otros sí. Aun así, a medida que se van lanzando



nuevas versiones de los navegadores web, la compatibilidad con ECMAScript va aumentando.

Diferencia entre ECMAScript y JavaScript

En realidad, no existe ninguna diferencia, si bien se pueden distinguir ciertos matices. ECMAScript es un estándar definido por ECMA International que es implementado por distintas empresas en sus navegadores web. Para aludir al lenguaje o implementación que exactamente reconoce un navegador, se suele utilizar el término **JavaScript** en lugar de ECMAScript. Además, "JavaScript" es una marca registrada en los Estados Unidos¹ propiedad de **Oracle Corporation**. La marca comercial originalmente se emitió a nombre de **Sun Microsystems** el 6 de mayo de 1997 y se transfirió a Oracle cuando adquirieron Sun Microsystems en 2009².

Versiones de ECMAScript

La lista de versiones de ECMAScript publicadas hasta el momento son las siguientes:

¹ "U.S. Trademark Serial No. 75026640". uspto.gov. *United States Patent and Trademark Office*. 1997-05-06. *Archived from the original on 2021-07-13. Retrieved 2021-05-08.* / "Legal Notices". oracle.com. *Oracle Corporation*. *Archived from the original on 2021-06-05. Retrieved 2021-05-08.*

² "Oracle to buy Sun in \$7.4-bn deal - The Economic Times". *The Economic Times*.



Edición	Fecha de publicación	Nombre	Cambios desde la edición anterior
1	Junio de 1997	ECMAScript 1997 (ES1)	Primera edición.
2	Junio de 1998	ECMAScript 1998 (ES2)	Cambios editoriales para mantener la especificación completa alineada con el estándar internacional ISO/IEC 16262
3	Diciembre de 1999	ECMAScript 1999 (ES3)	Se agregaron expresiones regulares, mejor manejo de strings, nuevo control de declaraciones, manejo de excepciones con try/catch, definición más estricta de errores, formato para la salida numérica y otras mejoras.
4	AGO/2008	ECMAScript 2008 (ES4)	Esta versión fue abandonada y muchas de las características propuestas para la misma fueron propuestas para la edición ECMAScript Harmony.

Edición	Fecha de publicación	Nombre	Cambios desde la edición anterior
5	DIC/2009	ECMAScript 2009 (ES5)	Agrega el modo estricto, agrega algunas nuevas características, como getters y setters, librería para el soporte de JSON, y una nueva definición sobre las propiedades de los objetos.
5.1	DIC/2011	ECMAScript 2011 (ES5.1)	Cambios leves alineados con la tercera edición del estándar internacional ISO/IEC 16262:2011.
6	JUN/2015	ES2015, también conocida como "ES6 Harmony".	La sexta edición marcó un antes y un después en el desarrollo del lenguaje, con cambios significativos en la sintaxis para escribir aplicaciones complejas, incluyendo clases y módulos, iteradores for/of loops, generadores y generador de expresiones estilo Python, funciones de dirección, datos binarios, colecciones (mapas, sets, mapas débiles), y proxies (metaprogramación para objetos virtuales y wrappers).

Edición	Fecha de publicación	Nombre	Cambios desde la edición anterior
7	JUN/2016	ES2016	Incorpora el método <code>Array.prototype.includes()</code> y el operador exponencial <code>(**)</code> .
8	JUN/2017	ES2017	Incluye los constructores <code>async/await</code> .
9	JUN/2018	ES2018	Incluye los operadores <code>rest/spread</code> para variables, iteración asincrónica, <code>Promise.prototype.finally()</code>
10	JUN/2019	ES2019	Incorporó <code>Array.flat()</code> , <code>Array.flatMap()</code> , <code>String.trimStart()</code> , <code>String.trimEnd()</code> , errores opcionales en el bloque <code>catch</code> , <code>Object.fromEntries()</code> , <code>Symbol.description</code>

Edición	Fecha de publicación	Nombre	Cambios desde la edición anterior
11	JUN/2020	ES2020	Además de nuevas funcionalidades, esta versión incluye el tipo primitivo BigInt para enteros de tamaño arbitrario primitivo, el operador nullish coalescing, optional chaining y el objeto globalThis.
12	JUN/2021	ES2021	Promise any(), String replaceAll(), Numeric Separators (_)