



ECMASCRIPT

Especificación estándar de
un lenguaje.

AGENDA

ECMAScript

ES1

ES2

ES3

ES4

ES5

Gustavo Adolfo Garcia Blanco

ES6

ES7

ES8

ES9

ES10

ES11

ECMAScript

- Es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO/IEC 22275:2018.

Gustavo Adolfo García Blanco

- ECMAScript define un lenguaje de tipos dinámicos ligeramente inspirado en JAVa y otros lenguajes del estilo de C. Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudoclases.

ECMAScript

- La mayoría de navegadores de Internet incluyen una implementación del estándar ECMAScript, al igual que un acceso al Document Object Model para manipular páginas web. JavaScript está implementado en la mayoría de navegadores, Internet Explorer de Microsoft usa JScript. El navegador Opera tenía su propio intérprete de ECMAScript con extensiones para soportar algunas características de JavaScript y JScript, actualmente Opera está basado en Chromium (y utiliza su intérprete). Cada navegador tiene extensiones propias al estándar ECMAScript, pero cualquier código que se adecúe al estándar debería funcionar en todos ellos.
- Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

ES1

- Primera edición publicada en Junio de 1997.
- Estándar completo en el siguiente link:
https://www.ecma-international.org/wp-content/uploads/ECMA-262_1st_edition_june_1997.pdf

ES2

- Segunda edición publicada en Agosto de 1998.
- Cambios editoriales para mantener la especificación completa alineada con el estándar internacional ISO/IEC 16262 con respect a ES1.

Gustavo Adolfo Garcia Blanco

- Estándar completo en el siguiente link:
https://www.ecma-international.org/wp-content/uploads/ECMA-262_2nd_edition_august_1998.pdf

ES3

- Tercera edición publicada en Diciembre de 1999.
- 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 con respecto a ES2.
- Estándar completo en el siguiente link: https://www.ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf

ES4

- La cuarta edición fue abandonado, nunca se publico.

Gustavo Adolfo Garcia Blanco

- La cuarta edición fue abandonada debido a diferencias políticas respecto a la complejidad del lenguaje. Muchas características propuestas para la cuarta edición fueron completamente abandonadas; algunas fueron propuestas para la edición ECMAScript Harmony.

ES5

- Quinta edición publicada en Diciembre de 2009.
- Agrega el modo estricto ("**strict mode**"), un subconjunto destinado a proporcionar una mejor comprobación de errores y evitar constructores propensos a errores. Aclara varias ambigüedades de la tercera edición, y afina el comportamiento de las implementaciones del "mundo real" que difieren consistentemente desde esa especificación. Agrega algunas nuevas características, como getters y setters, librería para el soporte de JSON, y una más completa reflexión sobre las propiedades de los objetos con respecto a ES3.
- Estándar completo en el siguiente link: https://www.ecma-international.org/wp-content/uploads/ECMA-262_5th_edition_december_2009.pdf

ES5.1

- 5.1 edición publicada en Junio de 2011.
- Esta edición 5.1 de la ECMAScript Standard está completamente alineada con la tercera edición del estándar internacional ISO/IEC 16262:2011.
- Estándar completo en el siguiente link:
https://www.ecma-international.org/wp-content/uploads/ECMA-262_5.1_edition_june_2011.pdf

ES6

- Sexta edición publicada en Junio de 2015.
- La sexta edición agrega cambios significativos en la sintaxis para escribir aplicaciones complejas, incluyendo clases y módulos, definiéndolos semánticamente en los mismos términos del modo estricto de la edición ECMAScript 5. Otras nuevas características incluyen 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). Al ser la primera especificación "ECMAScript Harmony", es también conocida como "ES6 Harmony".
- Estándar completo en el siguiente link: https://www.ecma-international.org/wp-content/uploads/ECMA-262_6th_edition_june_2015.pdf

ES7

- Séptima edición publicada en Junio de 2016.
- La séptima edición fue una mera actualización de la versión 6. Incorpora el método `Array.prototype.includes()` y el operador exponencial `**`.
- Estándar completo en el siguiente link: <https://262.ecma-international.org/7.0/>

ES8

- Octava edición publicada en Junio de 2017.
- La 8ª edición, oficialmente conocida como ECMAScript 2017, fue finalizada en junio de 2017. Incluye constructores `async/await`, los cuales funcionan usando generadores y promesas.
- Estándar completo en el siguiente link:
<https://262.ecma-international.org/8.0/>

ES9

- Novena edición publicada en Junio de 2018.
- La 9.^a edición, oficialmente conocida como ECMAScript 2018, incluye operadores rest/spread para variables (tres puntos: `...identificador`), iteración asincronica, `Promise.prototype.finally()`
- Estándar completo en el siguiente link:
<https://262.ecma-international.org/9.0/>

ES10

- Décima edición publicada en Junio de 2019.
- La 10.^a edición, oficialmente conocida como ECMAScript 2019, incorporó `Array.flat()`, `Array.flatMap()`, `String.trimStart()`, `String.trimEnd()`, errores opcionales en el bloque `catch`, `Object.fromEntries()`, `Symbol.description`.
- Estándar completo en el siguiente link: <https://262.ecma-international.org/10.0/>

ES11

- Undécima edición publicada en Junio de 2020.
- La 11.^a edición, oficialmente conocida como ECMAScript 2020, incluye el tipo primitivo **BigInt**, el operador de unión nula (`??`), el encadenamiento opcional (`?.`) y el objeto `globalThis`.
- Estándar completo en el siguiente link:
<https://262.ecma-international.org/11.0/>