¿Qué es ECMAScript?

- ECMAScript es una especificación mantenida por **Ecma International** (principalmente por el comité *TC39*) que define el lenguaje JavaScript: su sintaxis, tipos, APIs básicas, comportamiento de objetos, etc. (Wikipedia)
- JavaScript es una implementación de ECMAScript (junto con APIs del navegador, de Node, etc.).
- Desde 2015, las versiones del estándar se publican cada año (ES2015, ES2016, …) con características nuevas ya finalizadas. (Wikipedia)

Últimas cuatro versiones y sus mejoras

Voy a cubrir ES2022, ES2023, ES2024, ES2025 (o lo que ya se ha publicado/finalizado) con sus funcionalidades más destacadas.

Versión	Año	Nuevas funcionalidades destacadas
ES2022	2022	Algunas de las más relevantes: • Campos públicos y privados en clases, incluyendo métodos privados, accesores privados. • Bloques estáticos (static { … }) dentro de clases para inicialización específica de clase. • El operador in para campos privados: permite comprobar si un objeto tiene un campo privado. • at método para acceder a índices relativos en *Strings*, *Arrays* y *TypedArrays*. • Propiedades cause en objetos Error (para mantener la cadena de causas cuando se encadenan errores). • match indices en expresiones regulares con la bandera d, para obtener índices de inicio y fin de coincidencias. (Wikipedia)
ES2023	2023	Algunas mejoras introducidas: • Nuevos métodos en Array.prototype y TypedArray.prototype: findLast, findLastIndex. • Métodos que devuelven nuevas colecciones o versiones modificadas de arrays, por ejemplo toSorted, toReversed, with, toSpliced (sin mutar el original). • Soporte para *shebang* (#!) al inicio de archivos JS para facilitar scripts ejecutables. • Que la mayoría de los Symbols se puedan usar como claves en colecciones débiles (WeakMaps, etc.) donde antes algunas restricciones hacían que no todos los símbolos fueran permitidos. (Wikipedia)
ES2024	2024	Algunas de las características esperadas o ya finalizadas incluyen: • Métodos groupBy en Object y Map (“agrupar” colecciones basado en callback). • Promise.withResolvers, un método estático que devuelve un objeto que contiene la promesa junto con sus funciones resolve, reject. • Operaciones de conjunto (Set.prototype) más ricas. • La bandera /v para expresiones regulares unicode. (Wikipedia)
ES2025	2025	La versión más reciente ya publicada (junio de 2025). Algunas de sus novedades mencionadas: • Objeto Iterator que puede envolver iteradores como Arrays y proporcionar una interfaz funcional con evaluación perezosa (*lazy evaluation*). • Promise.try: permite llamar algo que puede o no ser una promesa y tratarlo como promesa siempre. • Nuevos métodos de Set: intersection, difference, symmetricDifference, isSubsetOf, isSupersetOf, isDisjointFrom. • RegExp.escape: función para escapar una cadena para ser usada dentro de un patrón regex. • Importar archivos JSON directamente como módulos. • Float16Array y métodos asociados en *TypedArrays*. (Wikipedia)

Versión más reciente

- La versión más reciente es **ECMAScript 2025** (también llamada ES2025), publicada en junio de 2025. (Wikipedia)
- Algunos de los cambios más relevantes que aporta:

    1. **Operaciones con Set extendidas**: ahora puedes hacer operaciones de teoría de conjuntos directamente con métodos como intersección, diferencia, etc. Muy útil si trabajas con datos en los que necesitas comparar conjuntos, combinar, etc.
    2. **Importar JSON como módulos**: esto simplifica cargar datos JSON directamente usando la sintaxis de módulo, lo que mejora la legibilidad y opción de optimización.
    3. **RegExp.escape**: ayuda a construir patrones de regex dinámicamente de forma segura sin tener que escribir tu propia función escapadora.
    4. **Iterator con evaluación perezosa**: permite procesar datos iterables sin materializar todos los valores de golpe, lo cual puede ser muy eficiente si los datos son grandes o la cadena de operaciones es larga.