

# Ejercicios ECMAScript vs JavaScript

Puedes ver la última especificación aquí: <https://tc39.es/ecma262/2023/>

Ejercicio: Contesta las siguientes preguntas:

- ¿Qué partes contiene la especificación?
- ¿Qué es un agente?
- ¿Qué funciones y propiedades tiene el prototipo de "Date" que no hayas usado y puedas usar.

Aquí se puede ver la lista de compatibilidad con Node (V8) desde ES2015 (ES6): <https://node.green/>

Investigación:

- ¿Qué partes heredó JS de Java?
- ¿Qué partes heredó JS de Scheme?
- ¿Qué partes heredó JS de Self?

## Ejercicio sobre Motores de JS vs Entornos de ejecución.

Descargar un **motor de JavaScript**:

- V8 (<https://v8.dev/docs/build>, es un buen reto contruirlo desde los archivos fuentes)
- Rhino ([https://github.com/mozilla/rhino/releases/tag/Rhino1\\_7\\_14\\_Release](https://github.com/mozilla/rhino/releases/tag/Rhino1_7_14_Release)),
- JavaScriptCore (<https://developer.apple.com/documentation/javascriptcore>)
- ChakraCore (<https://github.com/chakra-core/ChakraCore/releases/tag/v1.11.24>)

También puedes instalar JSVU (<https://www.npmjs.com/package/jsvu>) para instalar de manera más sencilla un binario.

Ejecuta un archivo, puede ser un simple "HelloWorld" y examina la API que exponen.

- ¿Cuáles son algunas de las cosas que todos los engines ofrecen?
- ¿En qué tipos de lenguajes están desarrollados?
- ¿Qué harías si necesitas que tu programa desarrollado en otro lenguaje pueda ejecutar JS?

Explora cómo puedes usarlo. Reto extra: agrégalo en un proyecto de otro lenguaje.

## Ejercicio sobre JIT

Contesta las siguientes preguntas:

- ¿Qué puedes hacer que tu código se aproveche mejor de las optimizaciones del JIT?

- ¿Qué engines tienen JIT y cuáles no?
- ¿Por qué simplemente no compilamos directamente JS?