Parece haber una gran confusión sobre este tema con las diversas respuestas que contradicen otras respuestas, con algunas respuestas simplemente equivocadas.

JavaScript solía ser puramente interpretado, pero eso fue hace muchos años. Hoy en día, se compila con JIT para el código de máquina nativo en todas las principales implementaciones de JavaScript. JavaScript no es un lenguaje interpretado. Desearía que la gente dejara de responder preguntas, diciendo que sí. En algunos casos, partes de un programa de JavaScript pueden interpretarse brevemente; consulte la explicación a continuación.

Exactamente cuando se compila en código máquina varía según la implementación. En el V8 actual (usado en Chrome y Node.js), comienza usando un intérprete ya que hay pocas razones para dedicar tiempo a compilar código que solo se ejecuta una vez. Sin embargo, si una función se ejecuta más de un par de veces, se compila inmediatamente en un código de máquina nativo optimizado. No hay forma de que un motor de JavaScript pueda competir con otras implementaciones de JavaScript sin compilar el código de la máquina, ya que el JavaScript moderno es realmente bastante rápido. V8 solía compilar el código de la máquina inmediatamente y luego tratar de optimizar las funciones "activas" de manera iterativa. Ese enfoque fue abandonado recientemente a favor de lo que describí anteriormente. Sin embargo, JavaScript es completamente diferente de sus scripts de shell estándar, Python o PHP u otros lenguajes interpretados. Si compila V8 desde el código fuente, puede usar "d8" para volcar el código de ensamblaje (representación textual uno a uno del código de máquina) que produce el código JavaScript.

Una respuesta a continuación sugiere que debido a que puede ver el código fuente en el navegador, puede decir que está interpretado. Esto no es verdad. Puede ver la fuente porque así es como se entrega al motor de JavaScript antes de que se compile JIT.

Una nueva característica en todas las implementaciones modernas de JavaScript es algo llamado WebAssembly, que es un código compilado AOT (de lenguajes que no son JavaScript, como C, por ejemplo) empaquetado en un formato más eficiente. En esos casos, no podrá ver el código fuente original.

También hay un AST binario en proceso que también evitaría que veas el código fuente. Se cargaría extremadamente rápido y comenzaría a ejecutarse de inmediato.

Nota preventiva: Sí, sé que se pueden compilar algunos idiomas interpretados, como Perl, por ejemplo, pero esa es una característica especial y no cómo es ejecutada por defecto.

[Julian Jensen](https://www.quora.com/profile/Julian-Jensen), 35+ years of programming especially computer games, creator of The Elder Scolls

Javascript es un lenguaje interpretado completo. No necesita compilar, producir y ejecutar para ejecutarlo. Todo lo que necesita hacer es ejecutarlo y el código se ejecuta línea por línea.

Si está ejecutando en un entorno de navegador, su navegador interpretará todas y cada una de las líneas y luego lo ejecutará. Sin embargo, los navegadores modernos tienen tecnología JIT (Just In Time) que compila Javascript en un código de bytes que está a punto de ejecutarse.

Los motores de Javascript como V8 y Rhino, SpiderMonkey compilan Javascript en código nativo y también tienen compilación JIT para mejorar el rendimiento.

Sin embargo, en NodeJS, que tiene el V8 que ejecuta Javascript, puede ser diferente, ya que tiene V8 debajo, puede tener todas las optimizaciones originales de conversión de código nativo, compilación JIT.

Línea de fondo :

Javascript es interpretado. Como en el lenguaje se ha implementado de esa manera porque se suponía que debía usarse en la plataforma del navegador. Y dado que el lenguaje interpretado no es una especificación, los proveedores de navegadores lo han modificado para mejorar el rendimiento.

Espero que esto ayude

[Maharshi Dutta](https://www.quora.com/profile/Maharshi-Dutta), Working for quite some time in Javascript