



Frontend Tips

Funciones avanzadas de JavaScript





1. Closures

Los **closures** son funciones que recuerdan el entorno en el que fueron creadas. Esto significa que tienen acceso a las variables de su ámbito exterior, incluso después de que ese ámbito haya finalizado. Los closures son útiles para la creación de **funciones privadas** y la manipulación de **datos encapsulados**.



2. Async/Await

La **sintaxis async/await** simplifica el trabajo con promesas, haciendo el código asíncrono más legible y fácil de escribir. "**Async**" declara **funciones asíncronas** y "**await**" **pausa la ejecución** hasta que la promesa se resuelva, permitiendo un código más secuencial y claro.



3. Promises

Las **promesas** son objetos que representan el eventual **resultado** (o falla) de una **operación asíncrona**. Con **"then"** y **"catch"**, puedes manejar las respuestas y errores de estas operaciones. Las promesas facilitan la **gestión de múltiples operaciones** asíncronas y evitan el "callback hell".



4. Destructuring

La **deestructuración** permite extraer **valores de arrays** u objetos en **variables distintas**. Esto puede hacer que tu código sea más limpio y fácil de leer.



5. Spread Operator

El **operador spread** permite **expandir elementos** de un array o propiedades de un objeto. Es útil para copiar y concatenar arrays o combinar objetos.



profile.es

