

Hablemos de un punto importante que debes de tener en cuenta a la hora de elegir cuál será la implementación que utilizarás en tus proyectos para manejar de forma correcta el asincronismo en JavaScript.

Teniendo en cuenta lo aprendido en este curso quiero mostrarte las ventajas y desventajas que tienen cada una de las formas en las que podemos manejar asincronismo.

Callbacks

Ventajas

- *Simpleza*
Nos permite disponer de una sintaxis fácil de entender y comprender qué sucederá al ser ejecutada.
- *Compatibilidad*
Los Callbacks son funciones que no necesitan convertir tu código con un polyfill para que funcionen con todos los navegadores modernos o versiones anteriores.

Desventajas

Entre las principales desventajas podemos encontrar que disponen de una estructura que puede llegar a ser robusta, más cuando anidamos llamadas a otras funciones, su flujo se puede volver poco intuitivo lo cual nos hará no comprender claramente su estructura.

Manejo de Errores, con los Callbacks no tenemos un camino claro para manejar los errores lo cual se traduce en problemas a la hora de manejar la lógica de nuestro programa.

Promesas

Ventajas

- *Flujo fluido*
Con las promesas podemos manejar un flujo complejos, anidar llamadas y tener una sintaxis clara que nos permite entender nuestro programa o la lógica que implica su uso.
- *Manejo de Errores*
Las promesas nos proporcionan un forma clara de manejar errores, una sintaxis sencilla y una forma de entender qué sucederá cuando sean ejecutadas.

Desventajas

- *PolyFill*
Las promesas no son compatibles con todos los navegadores, si bien los navegadores modernos pueden interpretar sin problema alguno, navegadores como internet explorer 11 necesitan transpilar el código para que funcionen correctamente.

Async/Await

Ventajas

- *Sintaxis*
Tienen una sintaxis muy simple y clara de leer, lo que nos permite entender de forma muy

sencilla su funcionamiento.

- *Try/catch*
Podemos utilizar try/catch para el manejo de errores con lo cual podemos manejar una sintaxis clara para el manejo de los errores.

Desventajas

- *Polyfill*
Como las promesas, Async/Await aún no tienen toda la compatibilidad con los navegadores viejos, por lo cual necesitamos transpilar nuestro código para utilizarlos en cualquier navegador.

Conclusiones

Ahora que entiendes las ventajas y desventajas de los callbacks, promesas y async/await puedes tomar la decisión de cuál implementar en tus proyectos, teniendo en cuenta su uso, así como las implementaciones que estés realizando. En lo particular he dejado atrás a los Callbacks para pasar mi lógica que maneje asincronismo a las promesas y en casos particulares utilizar Async/Await.

Cuéntame, ¿cuáles han sido tus observaciones y cómo implementarías mejor estos recursos que dispones para manejar el asincronismo en JavaScript?