



Curso de Asincronismo con JavaScript



Aprobaste el curso

Felicitaciones, ya puedes acceder a tu [diploma digital](#)

9.33

Calificación

14 / 15

Aciertos

1. ¿Las promesas resuelven un principal problema de las callbacks?

callback hell



2. ¿Para qué nos sirve la clase XMLHttpRequest?

Retorna un archivo JSON dentro de una promesa

[REPASAR CLASE](#)

3. ¿El estado 4 de xhttp.readyState a qué hace referencia?

COMPLETED, La operación está terminada



4. ¿Para qué utilizamos `JSON.parse(xhr.responseText)`?

Convertir una respuesta de texto en un Objeto iterable



5. ¿La recomendación de la comunidad para anidar callbacks es?

Un máximo de 3 callbacks



6. ¿Cuáles son los argumentos que recibe una promesa?

resolve, reject



7. ¿Cuál es la forma correcta de retornar un Error en reject?

reject (new Error(`Error`))



8. ¿Para qué nos sirve xhttp.status === 200?

Verificamos que el estatus de la petición HTTP resuelva el estado 200



9. ¿Para qué nos sirve el método "catch()"?

Registra la razón del rechazo



10. ¿El método then() retorna?

Promesa



11. ¿Nos permite ejecutar una serie de promesas secuencialmente?

Promise.all()



12. ¿Nos permite definir una función asíncrona?

async



13. ¿Cuál es la expresión que pausa la ejecución de la función asíncrona y espera la resolución de la *promise*?

await



14. ¿Cuál es el método recomendado por la comunidad para manejar asincronismo en JavaScript?

Async/await



15. ¿Cómo aseguramos manejar los errores asincrónicos correctamente?

try { ...código } catch (error) { ...código }



REGRESAR