

Objetivo: Practicar con los conceptos fundamentales de callbacks, promesas y programación asíncrona en JavaScript. Esto incluye entender cómo funcionan los callbacks para manejar eventos y operaciones asíncronas, trabajar con promesas para manejar tareas que se resolverán en el futuro, y desarrollar habilidades para escribir código eficiente y no bloqueante.

1. Callback básico:

Crea una función saludar(callback) que tome un nombre como argumento y llame al callback con un saludo usando ese nombre.

2. Promesa simple:

Crea una función multiplicarPromise(a, b) que devuelva una promesa que resuelva el resultado de la multiplicación de a y b.

3. Callback anidado:

Modifica la función de multiplicación para que acepte un callback como tercer argumento y llame al callback con el resultado.

4. Promesa con retraso:

Crea una función retrasarPromesa(ms) que devuelva una promesa que se resuelva después de un cierto tiempo especificado en milisegundos.

5. Ejecución asíncrona:

Utiliza setTimeout dentro de una función para simular una operación asíncrona y llamar a un callback una vez que se complete.

6. Promesa encadenada:

Crea una cadena de promesas donde cada una dependa del resultado de la anterior, realizando operaciones diferentes en cada paso.

7. Callback con error:

Crea una función dividir(callback) que tome dos números como entrada y llame al callback con el resultado de la división, manejando el caso de división por cero.

8. Promesa con error:

Modifica la función de división para que devuelva una promesa y rechace con un mensaje de error si se intenta dividir por cero.

9. Ejecución paralela:

Crea un conjunto de funciones asíncronas y ejecútalas en paralelo utilizando promesas o callbacks, luego maneja los resultados una vez que todas hayan terminado.

10. Promesa combinada:

Combina varias promesas utilizando Promise.all para esperar a que todas se resuelvan antes de ejecutar una acción final.