

Memoria T3Actividad4

Ejercicio 4: Creación y gestión de un listado de libros utilizando `async/await`

Enunciado

El cuarto ejercicio consiste en replicar la funcionalidad del ejercicio anterior, donde se mostraba un listado de libros en pantalla con un retraso simulado. Sin embargo, en este ejercicio se deben utilizar las funciones propias de JavaScript `async/await` para gestionar la asincronía.

Análisis del Problema

En lugar de utilizar promesas directamente con `.then()`, se hace uso de `async/await`, lo cual proporciona una sintaxis más sencilla y legible para trabajar con operaciones asincrónicas en JavaScript. La funcionalidad sigue siendo la misma: mostrar un listado de libros después de un retraso de 2 segundos. La implementación debe incluir el manejo de errores para asegurar que cualquier problema durante la ejecución sea capturado y tratado adecuadamente.

Solución Propuesta

La solución se implementa en cuatro pasos:

1. **Creación del Array de Objetos:** Se mantiene el mismo array de objetos del ejercicio anterior, donde cada objeto representa un libro con las propiedades `title`, `year`, `isbn` y `author`.
2. **Función para Simular el Retraso:** Se implementa una función `esperar(ms)` que retorna una promesa, la cual se resuelve después de un tiempo determinado (en este caso, 2000 ms).
3. **Función Asíncrona para Mostrar los Libros:** Se crea una función `async` que utiliza `await` para pausar la ejecución durante el tiempo especificado y luego construye el listado de libros en formato HTML.
4. **Ejecución de la Función:** Se llama a la función asíncrona para que realice la operación y muestre el resultado en la página.

Código Fuente

HTML

El código HTML se encuentra en el archivo `T3Actividad4.html` del proyecto.

JavaScript

El código JavaScript está adjunto en el archivo `T3Actividad4.js` del proyecto.

Explicación del Código

1. **Creación del Array de Libros:**
 - Se define un array llamado `libros` que contiene cuatro objetos con la información de los libros. Cada objeto tiene las propiedades `title`, `year`, `isbn` y `author`.
2. **Función `esperar(ms)`:**
 - Esta función simula un retraso utilizando `setTimeout` y retorna una promesa que se resuelve después de un tiempo específico, en este caso 2000 milisegundos.
3. **Función Asíncrona `mostrarLibrosConRetraso()`:**
 - La función utiliza `await` para esperar que se complete la promesa de la función `esperar(2000)` antes de continuar con la ejecución.
 - Después de la espera, se construye una cadena de texto en formato HTML con la información de cada libro en una lista no ordenada (``).
 - El resultado se muestra en un elemento HTML con el id `resultado`.
 - Se incluye un bloque `try/catch` para manejar cualquier error que pueda ocurrir durante la ejecución de la función.
4. **Ejecución de la Función:**
 - Se llama a `mostrarLibrosConRetraso()` para que realice el proceso completo y presente el listado en la página.

Pruebas Realizadas

Se llevaron a cabo diferentes pruebas para verificar que la implementación con `async/await` funcionaba correctamente:

1. **Retraso en la visualización:** Se comprobó que el listado de libros se mostraba en pantalla después de un retraso de 2 segundos.

2. **Formato del resultado:** Se verificó que los libros se mostraban en una lista HTML con el formato correcto (``).
 3. **Gestión de errores:** Se simuló un error en el código para asegurar que el bloque `catch` capturaba el problema y mostraba un mensaje de error en la página.
-

Conclusión

El uso de `async/await` simplifica la implementación de la funcionalidad asincrónica, proporcionando una sintaxis más clara y fácil de mantener. El ejercicio cumple con los requisitos solicitados, mostrando el listado de libros en pantalla después de un retraso de 2 segundos y gestionando posibles errores de manera adecuada.
