

ASYNC AWAIT IN JAVASCRIPT



Introduction

Certainly! `async/await` is a way to write asynchronous code that looks and reads like synchronous code. This can make it easier to understand and debug your code especially if you are working with a lot of async code.

Async function

An async function is a function that is marked with the `async` keyword. It always returns a Promise, even if the function returns a value.

An async function can also contain an `await` expression, which is used to wait for a Promise to be resolved.

Await function

an await expression can only be used inside an async function. It is used to pause the execution of the async function until a Promise is resolved.

Try/Catch function

You can use a try/catch block to handle any errors that might occur while an async function is running. If an error occurs inside the try block, the execution of the function will be stopped, and the catch block will be executed.

A code example to make an HTTP request



```
async function getData() {  
  try {  
    const response = await  
fetch('https://some-api.com/endpoint');  
    const data = await response.json();  
    console.log(data);  
  } catch (error) {  
    console.error(error);  
  }  
}
```

Explanation of code

This code defines an async function called `getData`. The `async` keyword indicates that this is an asynchronous function, and it always returns a Promise.

Inside the function, there is a `try/catch` block that is used to handle any errors that might occur. The `try` block contains the code that might throw an error, and the `catch` block contains the code that will be executed if an error is thrown.

The first line of the `try` block uses the `await` keyword to wait for the Promise returned by the `fetch` function to be resolved. This means that the execution of the `getData` function will be paused until the `fetch` Promise is either fulfilled or rejected.

If the fetch Promise is fulfilled, the response variable will be set to the value of the Promise, which is the response object. If the fetch Promise is rejected, the catch block will be executed, and the error will be logged to the console.

The second line of the try block uses the await keyword to wait for the Promise returned by the response.json method to be resolved. This means that the execution of the getData function will be paused until the response.json Promise is either fulfilled or rejected.

If the response.json Promise is fulfilled, the data variable will be set to the value of the Promise, which is the parsed JSON data.

If the response.json Promise is rejected, the catch block will be executed, and the error will be logged to the console.

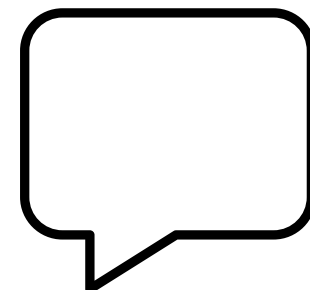
Finally, the console.log statement is executed, and the data variable is logged to the console.

Thanks for Watching!!

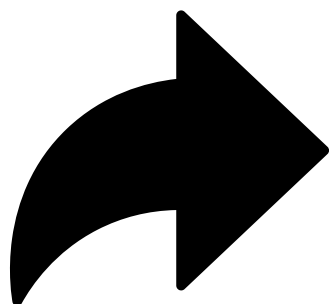
Like



Comment



Share



Follow us



shubhamsingh15