

## async and await in JavaScript

- **async** and **await** are used to handle asynchronous operations in JavaScript. They provide a more readable and synchronous-style approach to working with promises, making asynchronous code easier to manage and understand.

### async Keyword

- The **async** keyword is used to declare a function that always returns a **promise**.
- Even if the function does not explicitly return a promise, it automatically wraps the return value in a promise.

### await Keyword

- The **await** keyword can only be used inside an **async** function.
- It pauses the execution of the function until the promise resolves, and it returns the resolved value.
- The function execution continues only after the promise has settled, making the code look synchronous.

### Handling Errors with async/await

- **try...catch** blocks are used with **async** and **await** to handle errors that might occur during asynchronous operations.

### # Example:

```
javascript
async function fetchWithErrorHandling() {
    try {

    } catch (error) {

    }
}

fetchWithErrorHandling();
```

```
let getData = async ()=>{  
  try{  
    let data = await fetch("https://api.github.com/users")  
  
    // console.log(data)  
  
    let jsonData = await data.json()  
  
    console.log(jsonData) // it will give the original data.  
  }  
  catch(err)  
  {  
    console.log(err)  
  }  
}  
getData()
```