

2025 | FrontEnd Dev | Falak Naz | Personal Branding

Async Await

in JavaScript



@falku_naz



What is Async/Await?

async and **await** are modern JavaScript features used to handle asynchronous operations in a more readable and maintainable way compared to traditional callbacks or Promises. They allow you to write asynchronous code that looks and behaves like **synchronous** code, making it easier to understand.



Key Concepts

Async/Await is a way to make your code wait for an asynchronous operation to finish before moving on to the next step.

- **Async:** A function declared with the `async` keyword always returns a Promise. It allows the use of `await` inside the function.
- **Await:** The `await` keyword pauses the execution of the `async` function until the Promise is resolved or rejected. It can only be used inside an `async` function.



Why is it helpful?

- **Improves Readability:** Makes asynchronous code look synchronous, reducing complexity.
- **Error Handling:** Easier error handling using try-catch blocks compared to chaining .then() and .catch().
- **Avoids Callback Hell:** Provides a clean and flat structure, unlike nested callbacks.



Problem with Promises

Without Async/Await (Promises):

```
● ● ●  
fetchUser()  
  .then(user => fetchPosts(user.id))  
  .then(posts => console.log(posts))  
  .catch(error => console.error(error));
```

Chaining promises can make the code harder to read and maintain.



Solution with Async/Await

With Async/Await:

```
● ● ●

async function getUserPosts() {
  try {
    const user = await fetchUser();
    const posts = await fetchPosts(user.id);
    console.log(posts);
  } catch (error) {
    console.error(error);
  }
}
```

This approach makes the code look synchronous, making it more readable.



Conclusion

Using **async** and **await**:

- Simplifies handling of asynchronous operations.
- Reduces boilerplate code.
- Makes it easier to debug and maintain.

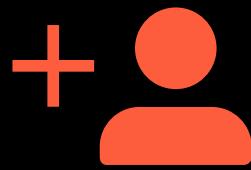
In modern JavaScript development, especially in environments with many asynchronous tasks (e.g., web APIs, databases), **async/await has become the preferred approach.**



2024

REPOST ❤

That's it!



Follow me!



Like this!



Share it!

