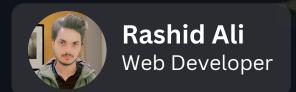
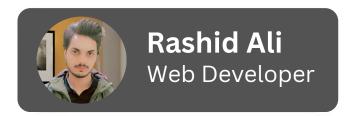
JS

Asynchronous function in JavaScript



async

An asynchronous function is a function that operates asynchronously via the event loop, using an implicit Promise to return its result. This keyword was introduced in **ECMAScript 2017** (ES8) and provides a cleaner and more readable way to write asynchronous code compared to traditional Promise-based or callback-based approaches.



How to write asynchronous function?

```
async function myAsyncFunction() {
    // Asynchronous operations
}
```

Using arrow function

```
O ● O ■ JS index.js

const myAsyncFunction = async () ⇒ {
    // Asynchronous operations
};
```



Example

from where data will be fetch, await is used here to pause the execution of the function.

```
async function fetchData() {
   const response = await fetch('https://api.example.com/data');
   const data = await response.json();
   return data;
}
```

Inside an async function, you can use the await keyword to pause the execution of the function until a promise is settled (resolved or rejected). The await keyword can only be used within an async function.

Return Values

```
async function example() {
   return 'Hello, World!';
}

example().then((result) ⇒ {
   console.log(result); // Output: Hello, World!
});
```

async functions always return a **promise**. If the function returns a value, the promise will be resolved with that value. If the function throws an error, the promise will be rejected with that error.





Rashid Ali

Web Developer

Did You Find it Useful? Share Your thoughts.



