

Async/Await

Async/await is a modern feature in JavaScript that simplifies working with asynchronous code, especially when dealing with Promises. It allows you to write asynchronous code in a synchronous-like manner, making it easier to read, write, and maintain.

1. **Async Functions:** An async function is a function that operates asynchronously via the event loop. You declare an async function by prefixing the function declaration with the **async** keyword.

```
async function myAsyncFunction() {  
  // Asynchronous code here  
}
```

2. **Await Keyword:** The await keyword is used inside an async function to pause the execution of the function until a Promise is settled (resolved or rejected). It allows you to write code that looks synchronous but behaves asynchronously.

```
async function myAsyncFunction() {  
  const result = await somePromise;  
  // Code here executes after somePromise is resolved  
}
```

Example

```
//promise is created  
function apromise() {
```

```
return new Promise(function (res, rej) {  
  var a = 20;  
  if (a % 2 == 0) {  
    res("num is even");  
  } else {  
    rej("num is odd");  
  }  
});  
}
```

//resolving the promise value using async/await

```
async function asyncfun() {  
  var v = await apromise();  
  console.log(v);  
}  
asyncfun();
```

//callback hell

```
async function executor(){  
  var addres=await add(10);  
  var subres=await sub(addres);  
  var mulres=await mul(subres);  
  var divres=await div(mulres);  
  console.log(divres);  
}  
executor()
```

//asynchronous

```
async function executor() {  
  let result1 = await promise1;  
  console.log(result1);  
  
  let result2 = await promise2;  
  console.log(result2);  
  
  let result3 = await promise3;  
  console.log(result3);  
  
  let result4 = await promise4;  
  console.log(result4);  
}  
  
executor();
```

How to fetch the api data using aync and await

```
async function apidata(){  
  const response = await fetch("https://fakestoreapi.com/products");  
  const data = await response.json();  
  
  console.log(data);  
}  
apidata()
```

fetch the data in local server

```
fetch("http://localhost:3000/data")  
  .then(response=>{  
  
    if(response.ok){
```

```
        return response.json()
    } else {
        return "error code" + response.statusText
    }
}).then(data=>{
    console.log(data);
})
```