Nishika Verma
LinkedIn

Async-Await

Javascript

@nishikaverma

JS
Async-Await

Nishika Verma
LinkedIn

•JavaScript is everywhere. Millions of webpages are built on JS.

•A few examples will help you understand the JavaScript Async await Keywords in this post.

@nishikaverma

Nishika Verma
LinkedIn

Async-Await

We use the async keyword with a function to represent that the function is an asynchronous function. The async function returns a promise.

The syntax of async function is:

```
async function name(parameter1, parameter2, ...paramaterN) {
// statements
}
```

Here,

name - name of the function.

parameters - parameters that are passed to the function.

Nishika Verma
LinkedIn

Async Function

```
// async function example

async function f() {
  console.log('Async function.');
  return Promise.resolve(1);
}

f();
```

Output

```
Async function.
```

In the above program, the async keyword is used before the function to represent that the function is asynchronous.

Nishika Verma
LinkedIn

Async Function

**Since this function returns a promise,
We can also use the chaining method 'then()' like this:**

```
async function f() {
  console.log('Async function.');
  return Promise.resolve(1);
}

f().then(function(result) {
  console.log(result)
});
```

Output

```
Async function
1
```

In the above program, the f() function is resolved and the then() method gets executed.

Nishika Verma
LinkedIn

JavaScript await Keyword

The await keyword is used inside the async function to wait for the asynchronous operation.

The syntax to use await is:

```
let result = await promise;
```

The use of await pauses the async function until the promise returns a result (resolve or reject) value.
For example,

```
// a promise
let promise = new Promise(function (resolve, reject) {
   setTimeout(function () {
   resolve('Promise resolved')}, 4000);
});

// async function
async function asyncFunc() {

   // wait until the promise resolves
   let result = await promise;

   console.log(result);
   console.log('hello');
}

// calling the async function
asyncFunc();
```

Output

```
Promise resolved
hello
```

JS
Async-Await

Nishika Verma
LinkedIn

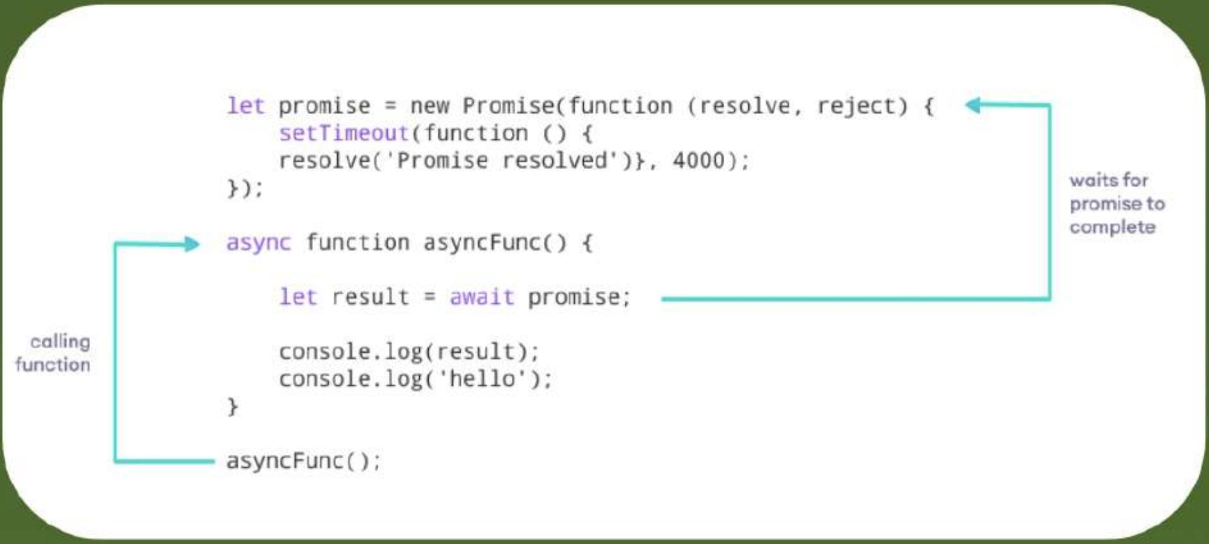
Explanation :

• In the above program, a Promise object is created and it gets resolved after 4000 milliseconds.

•Here, the asyncFunc() function is written using the async function.

• Hence, hello is displayed only after promise value is available to the result variable.

•The await keyword waits for the promise to be complete.



```
let promise = new Promise(function (resolve, reject) {
    setTimeout(function () {
    resolve('Promise resolved')}, 4000);
});

async function asyncFunc() {

    let result = await promise;

    console.log(result);
    console.log('hello');
}

asyncFunc();
```

@nishikaverma

Nishika Verma
LinkedIn

Benefits of Using async Function

- The code is more readable than using a callback or a promise.
- Error handling is simpler.
- Debugging is easier.

Note:

These two keywords async/await were introduced in the newer version of JavaScript (ES8). Some older browsers may not support the use of async/await.

Thank you!

Nishika Verma

LinkedIn