

Async Patterns



Mosh Hamedani
programmingwithmosh.com





Activities

Visual Studio Code

index.js - Applications - Visual Studio Code

Tue 11:48

0B/s ↓ 0B/s ↑



File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

APPLICATIONS

01. Synchronous Vs Asynchronous Code

async-demo

JS index.js

package.json

README.md

02. Patterns For Dealing With Asynchronous ...

async-demo

JS index.js

package.json

README.md

03. Callbacks

04. Callback Hell

05. Named Functions To Rescue

06. Promises

07. Replacing Callbacks With Promises

08. Consuming Promises

09. Creating Settled Promises

10. Running Parallel Promises

11. Async And Await

12. Exercise

README.md

OUTLINE

TIMELINE

JS index.js

02. Patterns For Dealing With Asynchronous Code > async-demo > JS index.js > getUser

```
1 console.log('Before');
2 // Synchronous or Blocking : Synchronous basically means that you can only execute one thing at a time.
3 // Asynchronous means that you can execute multiple things at a time
4 // and you don't have to finish executing the current thing in order to move on to next one.
5 const user = getUser(1);
6 // It will return undefined because returning value will not be available at the time of calling getUser function.
7 // Solutions to above problem are: Callbacks, Promises, Async/await
8 // Async/await is actually just syntax sugar built on top of promises.
9 console.log(user);
10 console.log('After');
11 // Node. js is a single threaded language which in background uses multiple threads to execute asynchronous code.
12 // Node. js is non-blocking which means that all functions ( callbacks ) are delegated to the event loop
13 // and they are ( or can be ) executed by different threads.
14
15 Complexity is 3 Everything is cool!
16 function getUser(id){
17     setTimeout(() => { // Asynchronous
18         console.log('Reading a user from the database...');
19         return { id: id, gitHubUsername: 'ankit'};
20     }, 2000); // Wait of two seconds
21 }
```

DEBUG CONSOLE PROBLEMS 1 OUTPUT TERMINAL

2: node

```
dandy@My-Ubuntu:~/Keep/GitHub Repositories/Node.js/06. Asynchronous JavaScript/Applications/02. Patterns For Dealing With Asynchronous Code/async-demo$ nodemon index.js
[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Before
undefined
After
Reading a user from the database...
[nodemon] clean exit - waiting for changes before restart
```

master*

0 0 0 1

✓ javascript | ✓ index.js

Ln 20, Col 2 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style