

Lesson 7 Demo 1

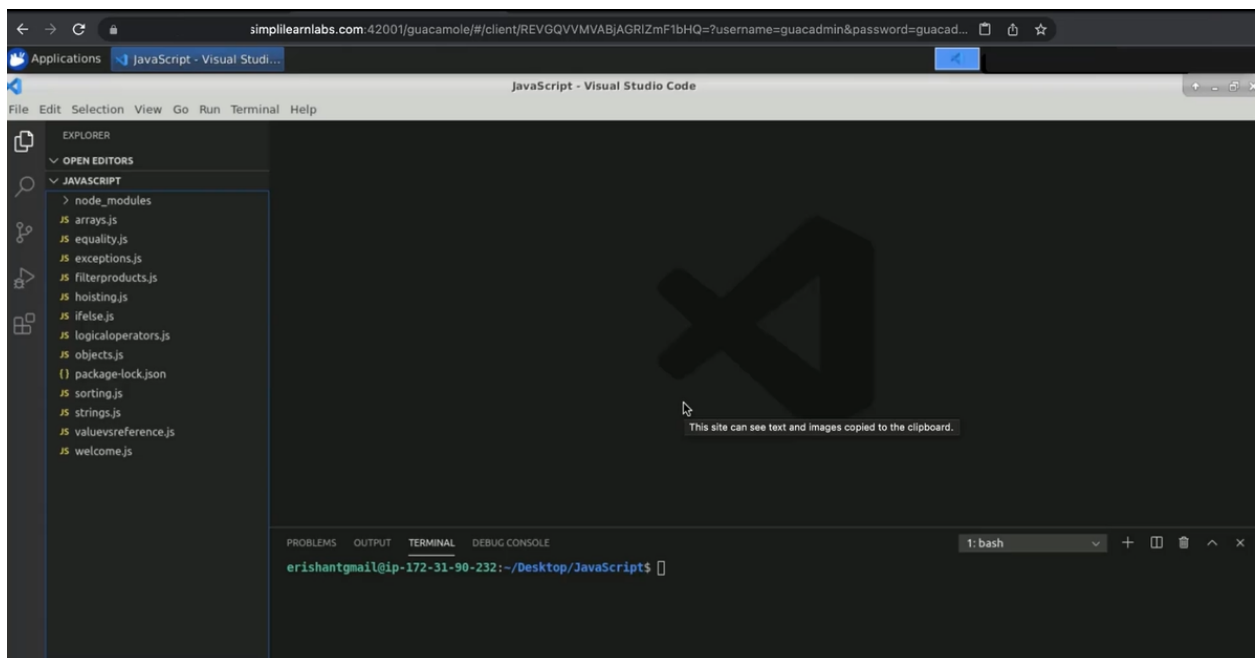
Async Programming in JS

Objective: To work with async programming in JS

Prerequisites: None

Steps to perform:

1. Open the visual studio code



2. Create a file name async.js

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar is open, displaying a file tree under 'JAVASCRIPT'. The files listed are: node_modules, arrays.js, async.js, equality.js, exceptions.js, filterproducts.js, hoisting.js, ifelse.js, logicaloperators.js, objects.js, package-lock.json, sorting.js, strings.js, valuesreference.js, and welcome.js. The 'async.js' file is selected. The main editor area shows an open file named 'async.js' with a single line of code: `1 |`. The bottom panel shows the 'TERMINAL' tab with the prompt `erishantgmail@ip-172-31-90-232: ~/Desktop/JavaScript$`.

3. Create a function to upload a file. Enter 'let uploadFile = () => "File Uploaded Successfully";'.

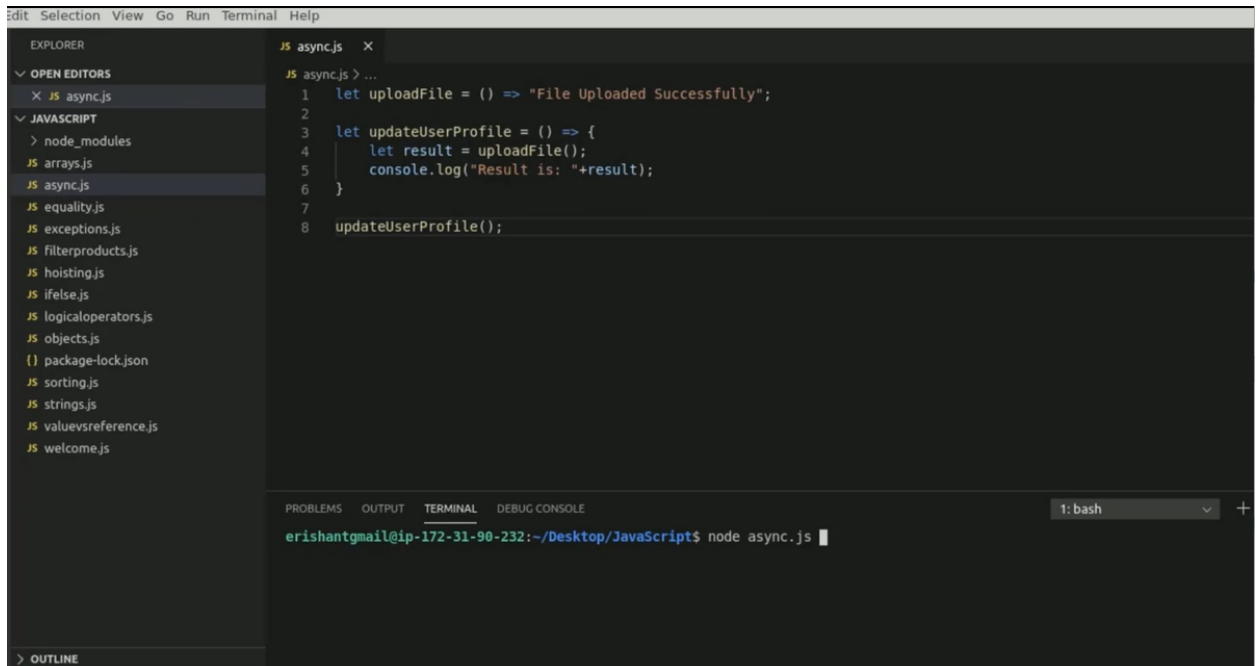
The screenshot shows the Visual Studio Code interface. The Explorer sidebar is open, and the 'async.js' file is selected. The main editor area shows the 'async.js' file with the following code: `1 let uploadFile = () => "File Uploaded Successfully";` and `2 |`. The bottom panel shows the 'TERMINAL' tab with the prompt `1: bash`.

4. Create one more function.

```
Enter 'let updateUserProfile = () => {
  let result = uploadFile();
  console.log("Result is: "+result);
}
```

5. Execute updateUserProfile() method.

6. Enter 'node async.js' in terminal.

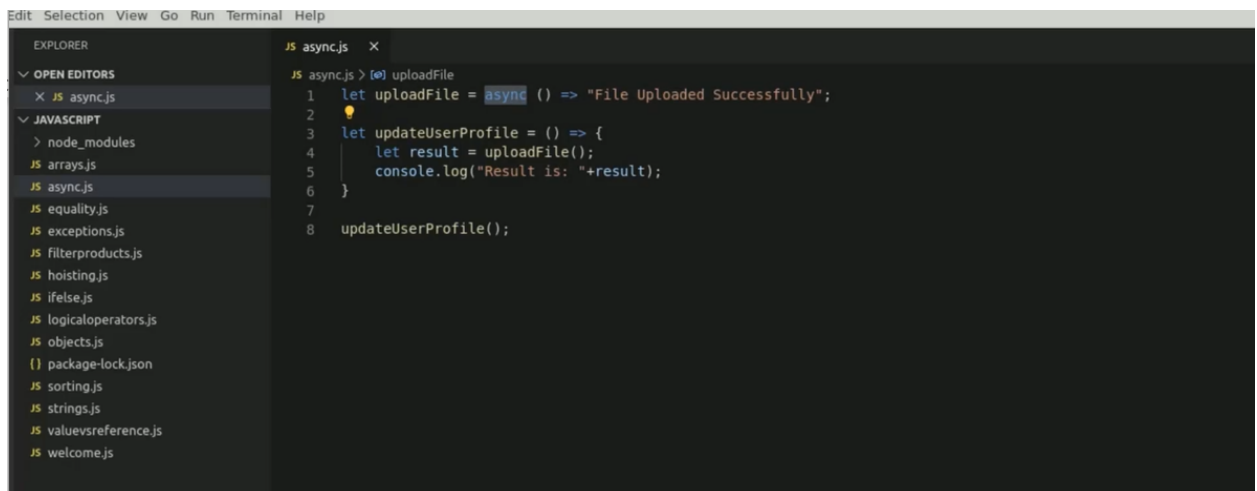


The screenshot shows the VS Code editor interface. On the left, the Explorer sidebar is open, showing a list of files under the 'JAVASCRIPT' folder, including 'async.js'. The main editor area displays the content of 'async.js', which contains the following JavaScript code:

```
1 let uploadFile = () => "File Uploaded Successfully";
2
3 let updateUserProfile = () => {
4   let result = uploadFile();
5   console.log("Result is: "+result);
6 }
7
8 updateUserProfile();
```

At the bottom, the Terminal window is open, showing the command prompt '1: bash' and the command 'node async.js' entered by the user.

7. Enter 'async' keyword with uploadFile() function.



The screenshot shows the VS Code editor interface. The main editor area displays the content of 'async.js', which is the same as in the previous screenshot, but with the 'async' keyword added to the 'uploadFile' function definition:

```
1 let uploadFile = async () => "File Uploaded Successfully";
2
3 let updateUserProfile = () => {
4   let result = uploadFile();
5   console.log("Result is: "+result);
6 }
7
8 updateUserProfile();
```

The 'async' keyword is highlighted in blue in the first line of code.

8. Run the code in terminal.

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$

```

9. Add another method. Enter result.then(
 (value) => console.log("Value is: "+value);
);

```

JS async.js
JS async.js > updateUserProfile
1 let uploadFile = async () => "File Uploaded Successfully";
2
3 let updateUserProfile = () => {
4   let result = uploadFile();
5   console.log("Result is: "+result);
6
7   result.then([
8     (value) => console.log("Value is: "+value);
9   ]);
10 }
11
12 updateUserProfile();

```

10. Run the code in terminal.

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$

```

11. Enter 'async' keyword with updateUserProfile() function.

```
JS async.js •
JS async.js > [⌘] updateUserProfile
1  let uploadFile = async () => "File Uploaded Successfully";
2
3  let updateUserProfile = async () => {
4    let result = uploadFile();
5    console.log("Result is: "+result);
6
7    result.then(
8      (value) => console.log("Value is: "+value)
9    );
10 }
11
12 updateUserProfile();
```

12. Enter 'await' in result upload file function().

```
JS async.js •
JS async.js > [⌘] updateUserProfile
1  let uploadFile = async () => "File Uploaded Successfully";
2
3  let updateUserProfile = async () => {
4    let result = await uploadFile();
5
6    /*console.log("Result is: "+result);
7
8    result.then(
9      (value) => console.log("Value is: "+value)
10   */
11 }
12
13 updateUserProfile();
```

13. Enter 'console.log("Result is: "+result);'.

```

JS async.js x
JS async.js > [0] updateUserProfile
1 let uploadFile = async () => "File Uploaded Successfully";
2
3 let updateUserProfile = async () => {
4   let result = await uploadFile();
5   console.log("Result is: "+result);
6
7   /*console.log("Result is: "+result);
8
9   result.then(
10    (value) => console.log("Value is: "+value)
11   );*/
12 }
13
14 updateUserProfile();

```

14. Run the code in terminal.

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$

```

15. Create new promise object. Enter 'let myPromise = new Promise(function(resolve, reject){ }).

```

JS async.js
JS async.js > [0] myPromise
1  let uploadFile = async () => "File Uploaded Successfully";
2
3  let updateUserProfile = async () => {
4      let result = await uploadFile();
5      console.log("Result is: "+result);
6
7      /*console.log("Result is: "+result);
8
9      result.then(
10         (value) => console.log("Value is: "+value)
11     );*/
12 }
13
14 updateUserProfile();
15
16 let myPromise = new Promise(
17     function(resolve, reject){
18         // Algo goes here
19     }
20 );
21

```

16. Inside {}, Enter '//Algo goes here
 let isDataFetched =true;
 if (isDataFetched){
 resolve();
 }else{
 reject();
 }.

```

JS async.js
JS async.js > [0] myPromise > <function>
11  );*/
12  }
13
14  updateUserProfile();
15
16  let myPromise = new Promise(
17      function(resolve, reject){
18          // Algo goes here
19
20          let isDataFetched = true;
21
22          if(isDataFetched){
23              resolve();
24          }else{
25              reject();
26          }
27      }
28  );

```


17. Enter 'myPromise.then(
 () => console.log("Data Fetched Successfully")
).catch(
 () => console.log("Data Fetched Operation Failed")
);'.

```

JS async.js x
JS async.js > catch() callback
17     function(resolve, reject){
18         // Algo goes here
19
20         let isDataFetched = true;
21
22         if(isDataFetched){
23             resolve(); // resolves to then
24         }else{
25             reject(); // rejects to catch
26         }
27     }
28 };
29
30 myPromise.then(
31     () => console.log("Data Fetched Successfully")
32 ).catch(
33     () => console.log("Data Fetched Operation Failed")
34 );
35
36 );

```

18. Run the code in terminal.

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: bash
Result is: [object Promise]
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: [object Promise]
Value is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetched Successfully

```

19. Enter 'isDataFetched = False;'.

```

JS async.js x
JS async.js > myPromise > <function> > isDataFetched
17 function(resolve, reject){
18     // Algo goes here
19
20     let isDataFetched = false;
21     if(isDataFetched){
22         resolve(); // resolves to then
23     }else{
24         reject(); // rejects to catch
25     }
26 }
27
28 );
29
30 myPromise.then(
31     () => console.log("Data Fetched Successfully")
32 ).catch(
33     () => console.log("Data Fetch Operation Failed")
34 );
35
36 );

```

20. Run the code in terminal.

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: bash

Value is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetched Successfully
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node async.js
Result is: File Uploaded Successfully
Data Fetch Operation Failed
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$

```