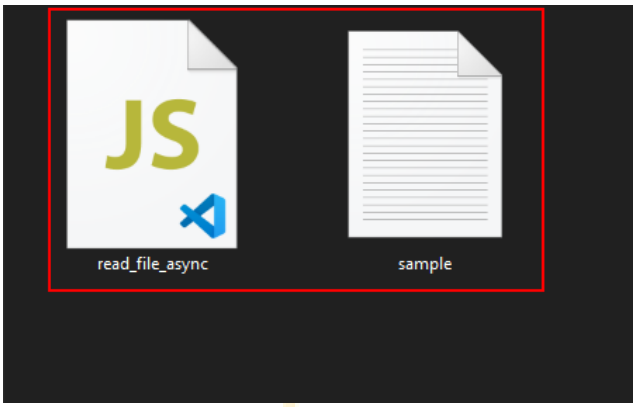
NodeJS Module 3: Hands-On:3

Reading the Contents of a File Asynchronously

Step 1: Create a file named **read_sync.js** and a text file named **sample.txt** with some sample text in it

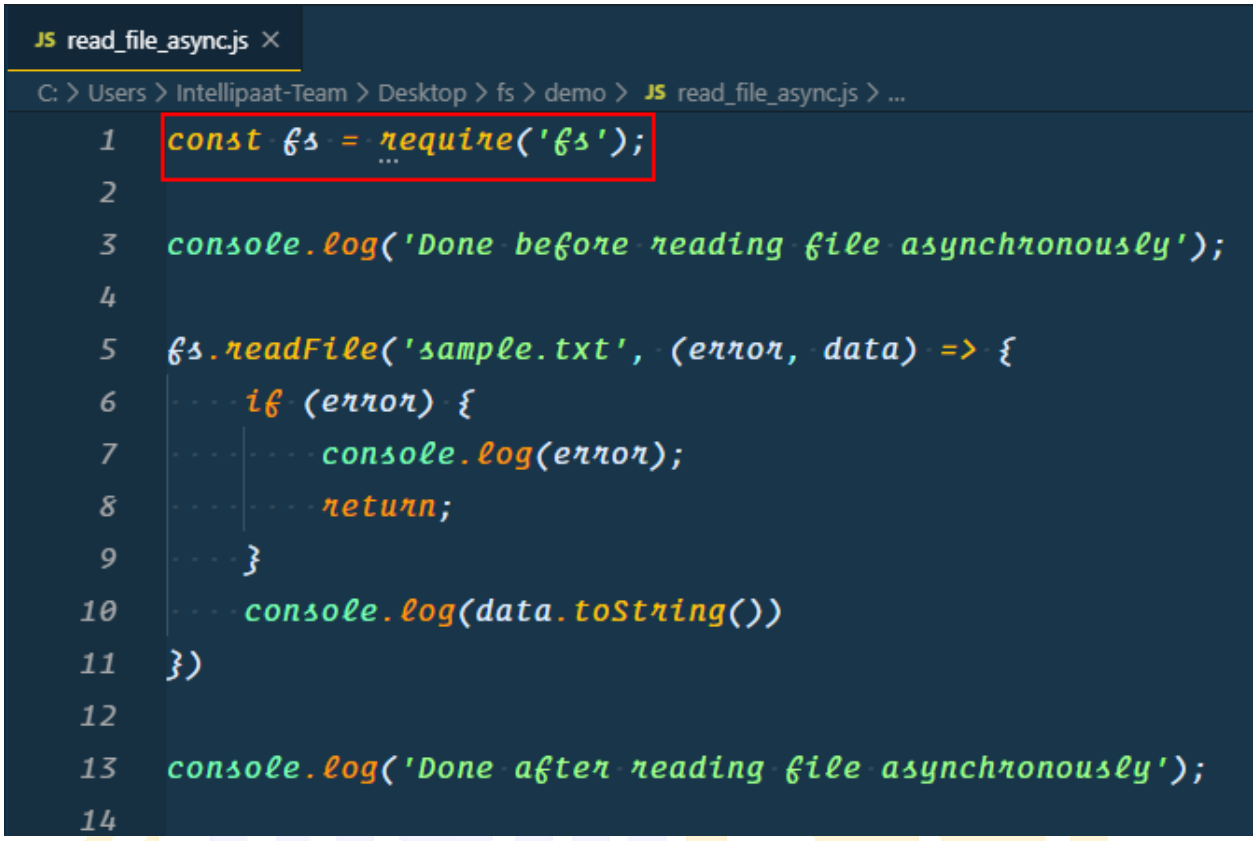


Step 2: Open **read_sync.js** in any text editor

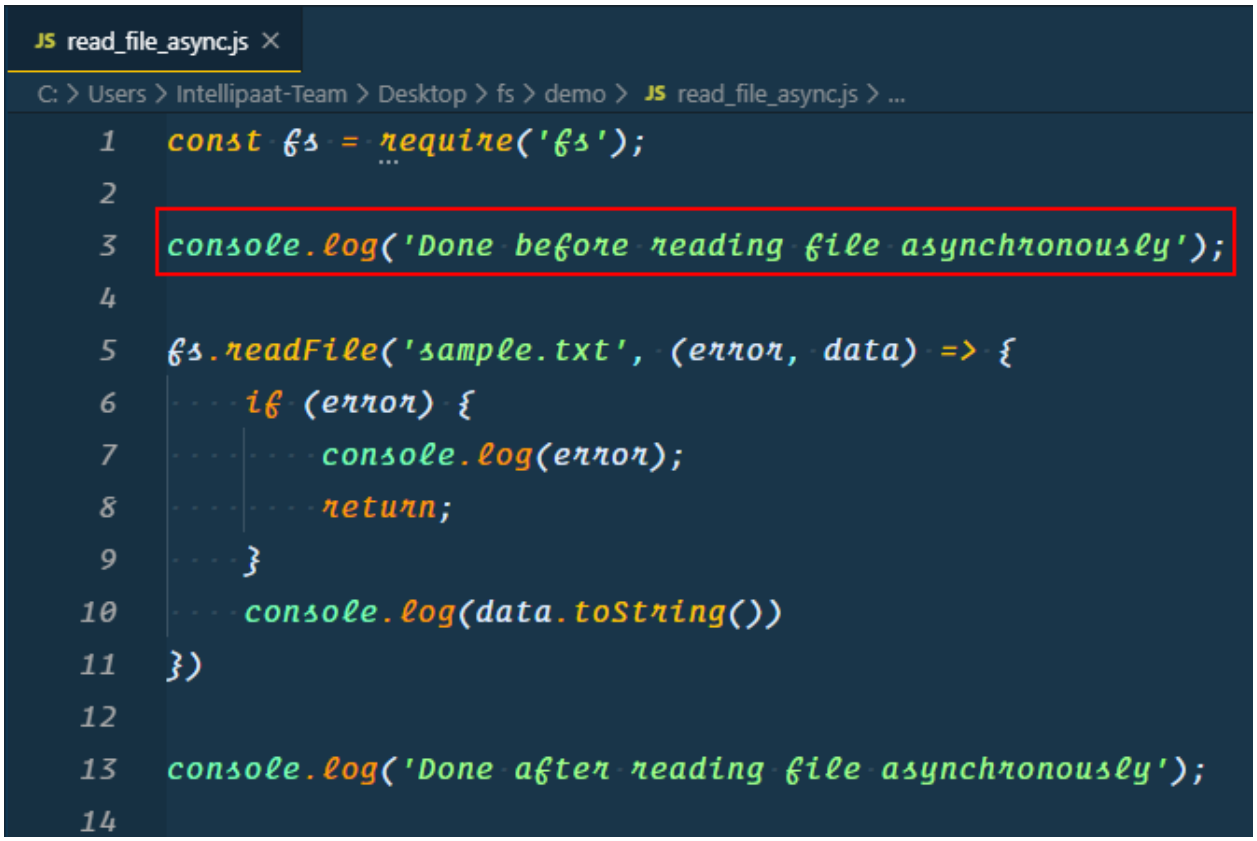
Step 3: Type the following code:

```
JS read_file_async.js ×
C: > Users > Intellipaat-Team > Desktop > fs > demo > JS read_file_async.js > ...
1   const fs = require('fs');
2
3   console.log('Done before reading file asynchronously');
4
5   fs.readFile('sample.txt', (error, data) => {
6       if (error) {
7           console.log(error);
8           return;
9       }
10      console.log(data.toString())
11  })
12
13  console.log('Done after reading file asynchronously');
14
```

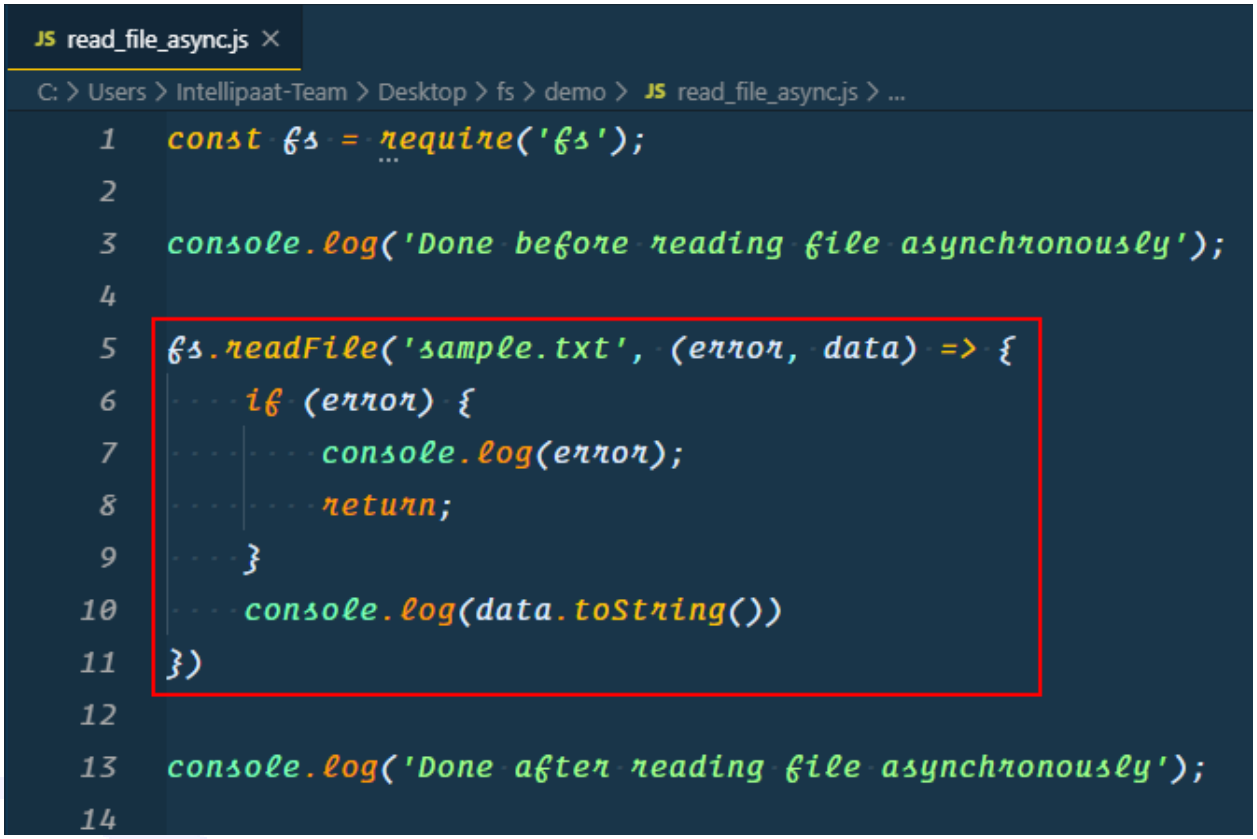
Step 3.1: Import the fs module using the 'require' keyword



Step 3.2: Write a console log before reading the code that reads the file contents asynchronously

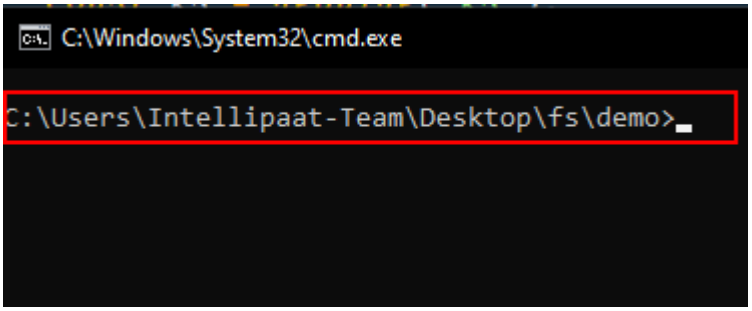


Step 3.3: Read the file contents asynchronously using the method 'readFile' from the fs module and pass in the filename 'sample.txt' and a callback that accepts errors and data. It will log the error if any, and if not then it will log the the data



Step 3.4: Write a console log after reading the code that reads the file contents asynchronously

Step 4: Open the command prompt in the same directory as the file



Step 5: Run the file using the command 'node read_file_async.js'. Note that the output does not appear in the same order in which you wrote it in code because it is read asynchronously. So, if it is a really big file, then your code will have stop until the file is read entirely; rather, the reading of file is pushed to a new thread and out program continues to work. After it finishes the synchronous code, it starts executing the callbacks specified by the asynchronous code