Node.js

Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.

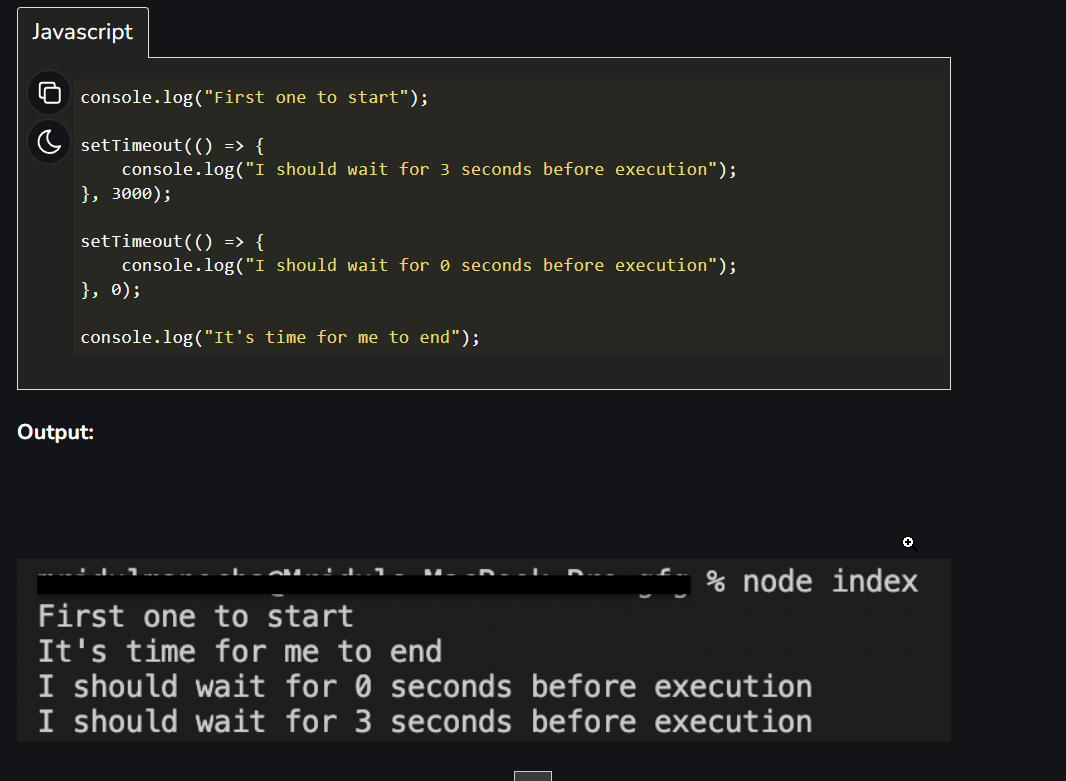
It is based on non blocking io model.

NodeJS has the ability to do multiple things at the same time is called asynchronous programming.

Think of a waiter working in a restaurant. He goes around the restaurant taking orders from customers and serving them when their respective food is ready. What happens when the waiter takes an order from a customer and waits until the food is prepared, serves it and then proceeds to the next customer. This way the waiting time for each customer increases and the restaurant would be a huge flop. The latter represents synchronous programming and the earlier one represents asynchronous programming.

Non-Blocking: Non-Blocking nature of node.js simply means that node.js proceeds with the execution of  the program instead of waiting for long I/O operations or HTTP requests. i.e the non-JavaScript related code is processed in the background by different threads or by the browser which gets executed when node.js completes execution of the main program and when the data required is successfully fetched. This way, the long time taking operations do not block the execution of the remaining part of the program. Hence the name Non-Blocking.

Example: Have a look at the below code-snippet for getting a better understanding of the non-blocking nature of Node.js.



Npm is node package manager.

* npm init

This command is used to initialize/set up the project. It also created package.json file that contains details about project.

* npm install express --save

This command is used to save install any new package i.e npm install(In the above command we are actually installing express framework)

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.

The size of node\_modules is so big that we don’t push or commit it to our repository.

That is why it is mentioned in package.json file.

We may install all the dependencies on our local by using the below command:

* npm install

To install any package globally(i.e can be used in any node project)

* npm i -g nodemon

g stands for global, nodemon is a package name here.

Nodemon is a command-line tool that helps with the speedy development of Node. js applications. It monitors your project directory and automatically restarts your node application when it detects any changes. This means that you do not have to stop and restart your applications in order for your changes to take effect.

**Dev Dependency**

This dependency is used only for development-related work. Below command is used to install dev dependencies.

* npm install –save-dev package\_name

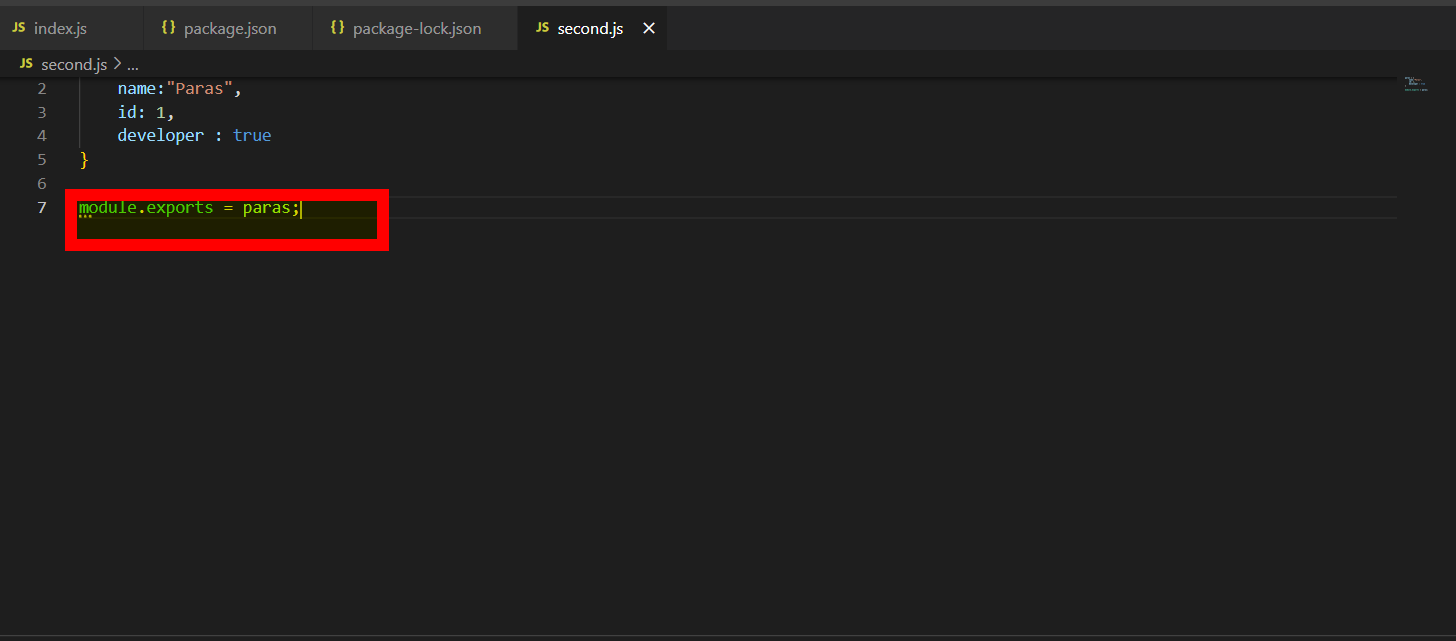
npm install –save-dev nodemon

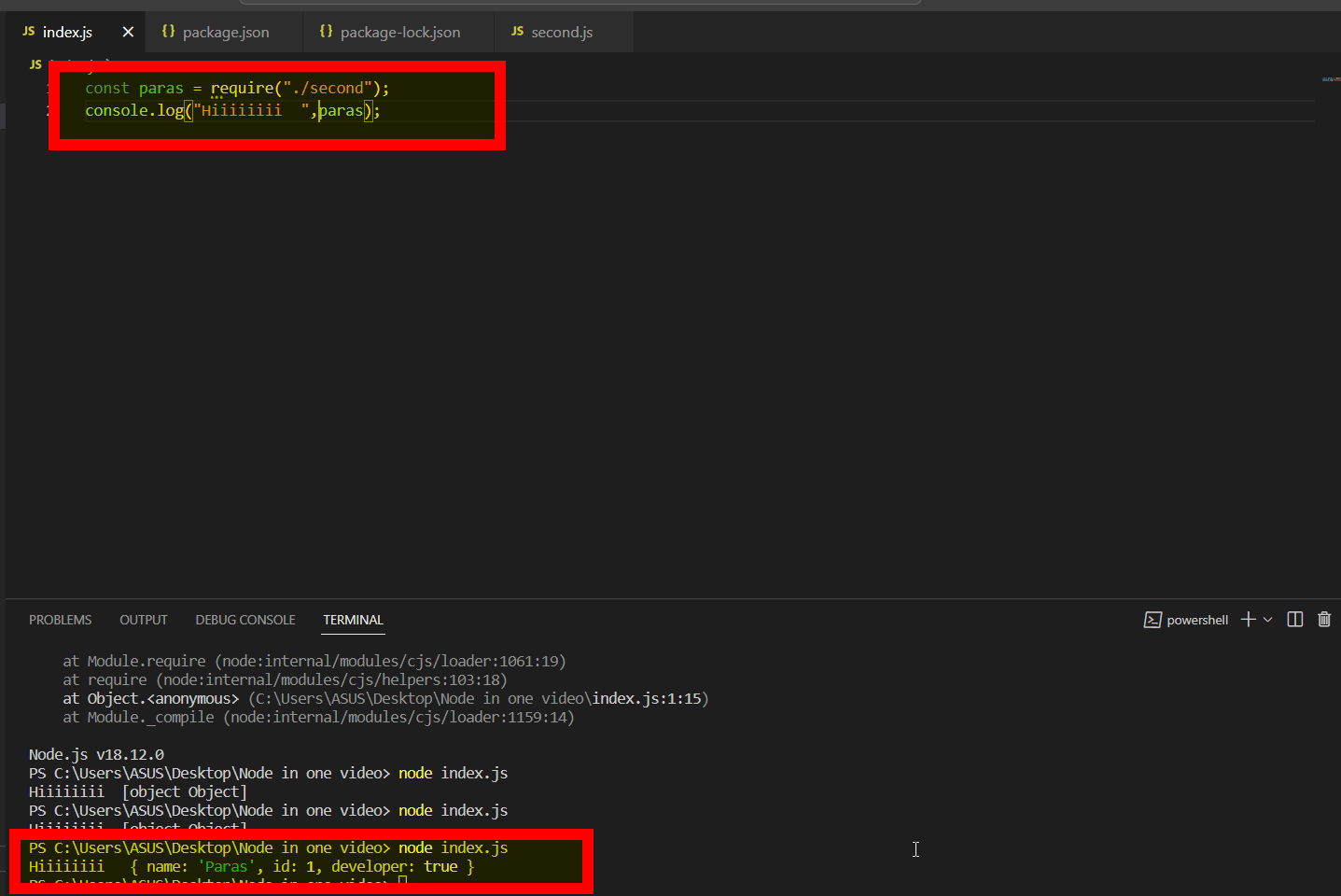
To uninstall any package use below command

* npm uninstall package\_name

package-lock.json – This file contains the dependency tree, package.json contains the dependency name but this file contains the version and related dependencies as well.

In order to use any object in multiple files, we first export that object and import it into the file in which it needs to be used.



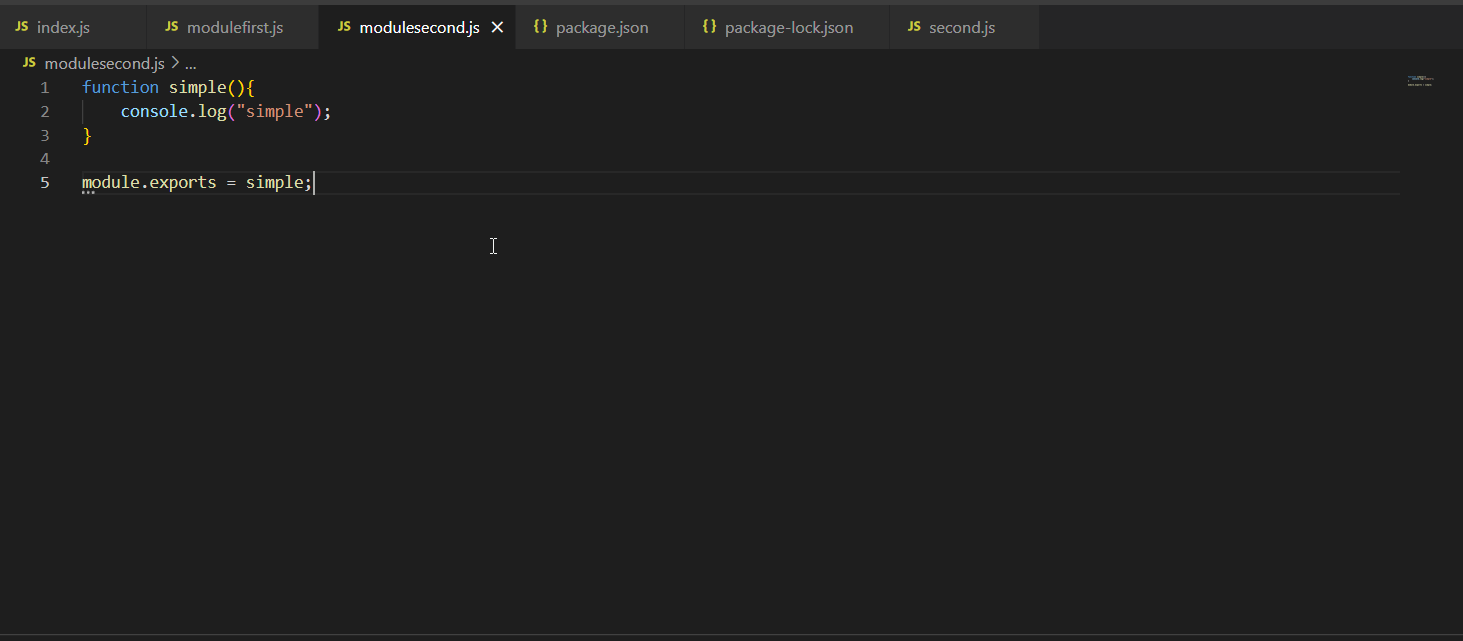


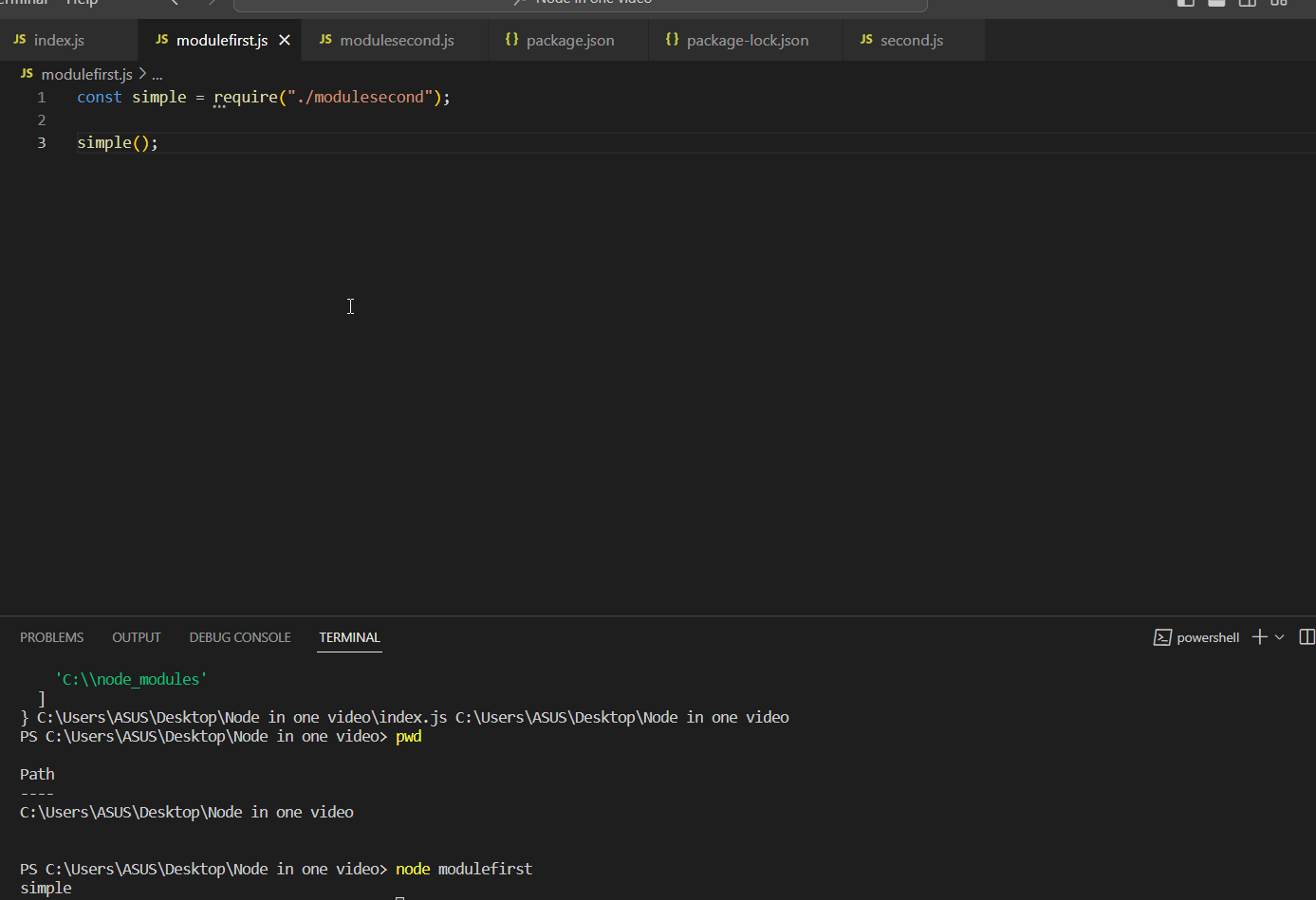
This system (exporting and importing ) is called common js model.

Learn about promises and callback in javascript.

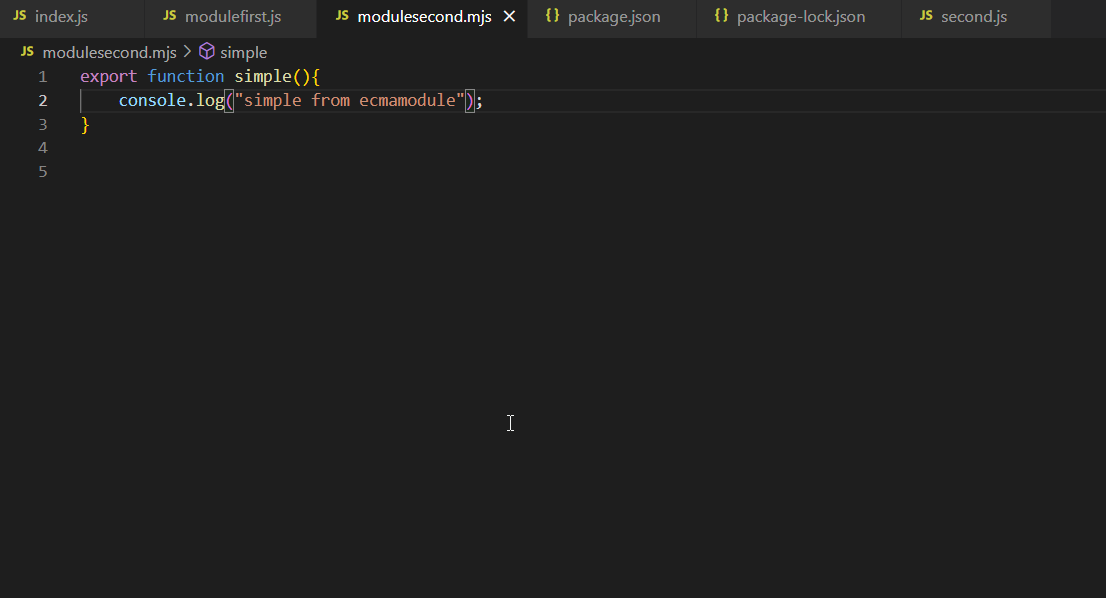
There are 2 types of modules in node.js:

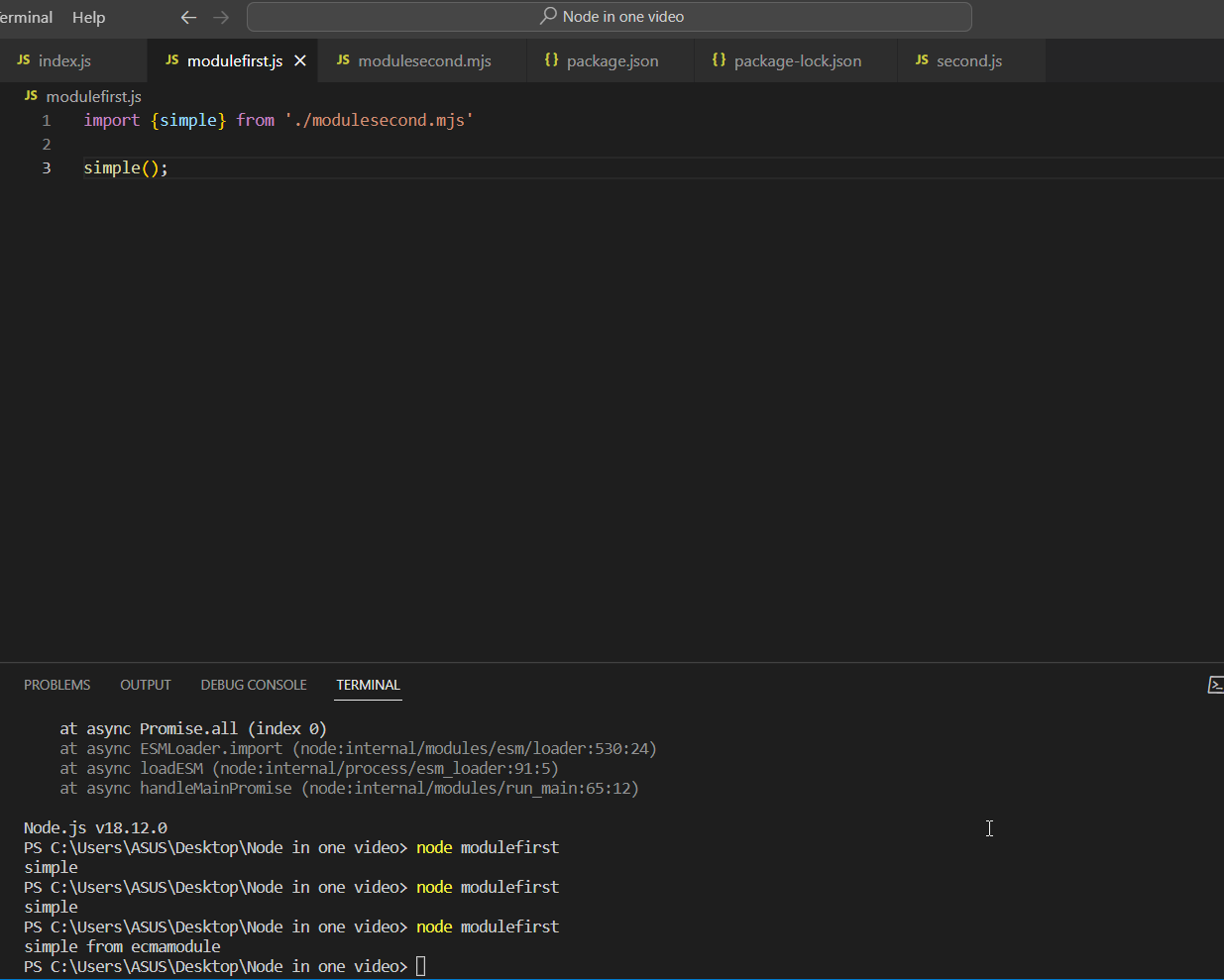
1. Common JS module





1. ECMAScript modules





In order to import all the objects/functions from a file, we may use the below sample code:

import \* as alias\_name from “./file\_name”;

const port = process.env.PORT (It is used to fetch the port number on which the application is running)