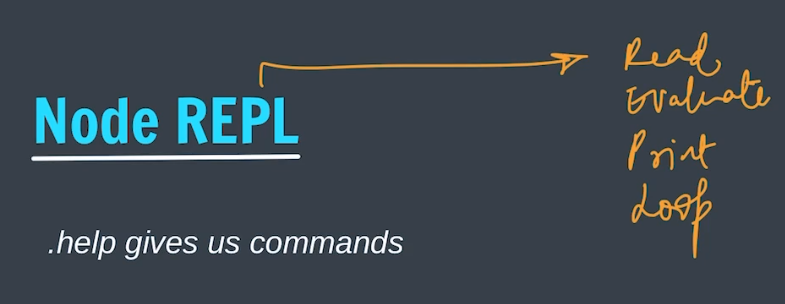
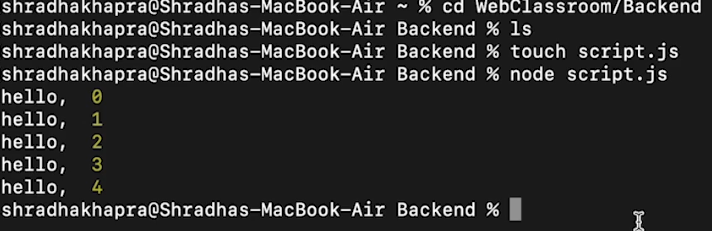
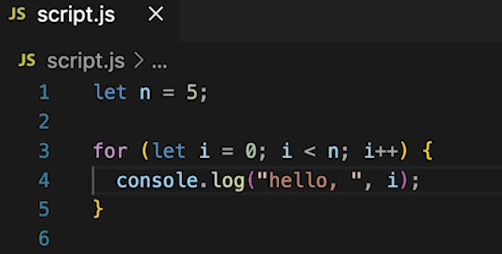
****

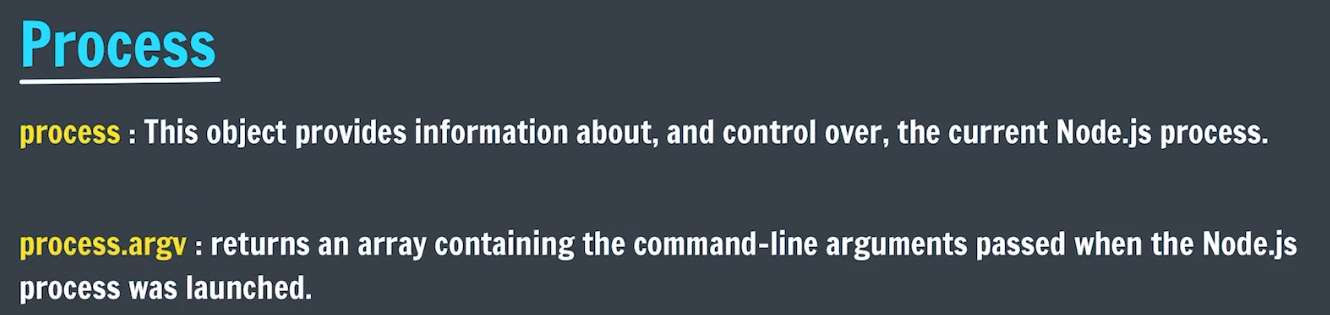
****

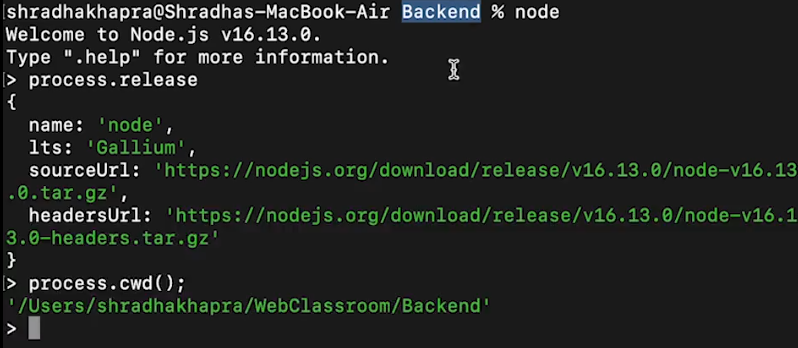
**Write node in the terminal and enter into Node REPL, Where you can write and execute the JS code.**

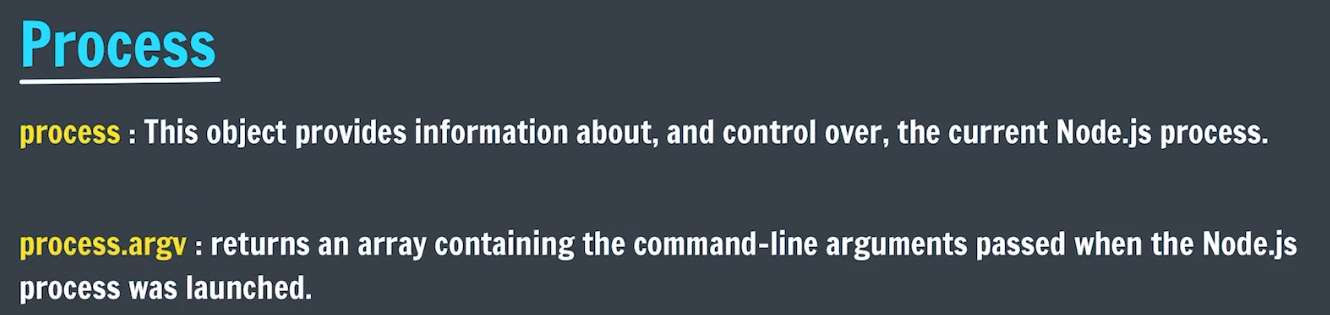
**To run a js file using node:**

1. **Just run it in vs code normally**
2. **Go to the terminal then to the directory of the file then use command, node file.js**

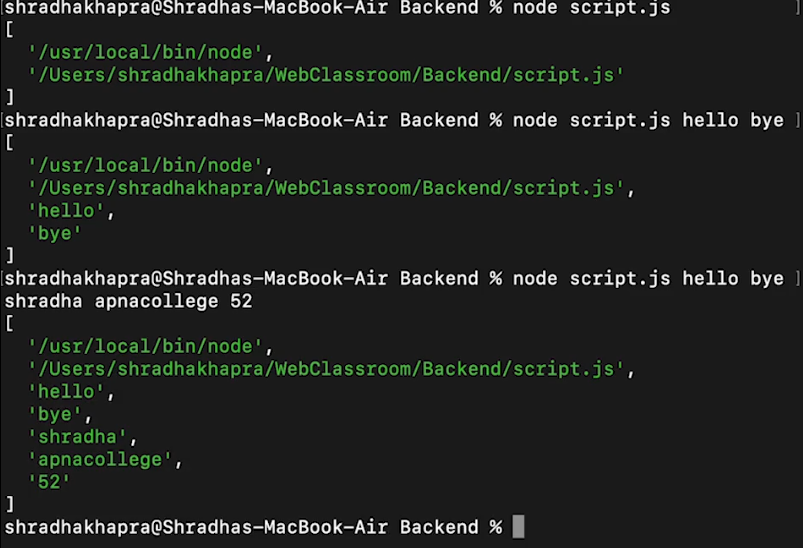
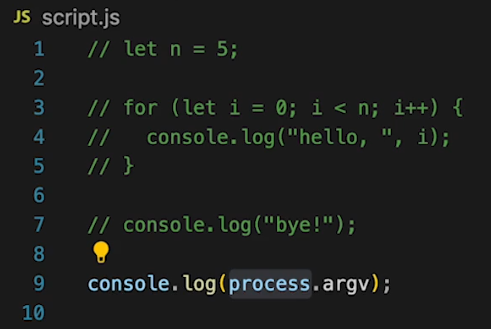
****

****

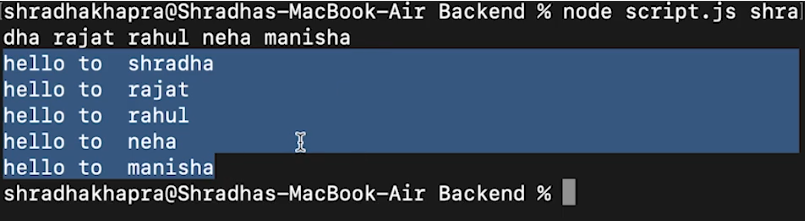
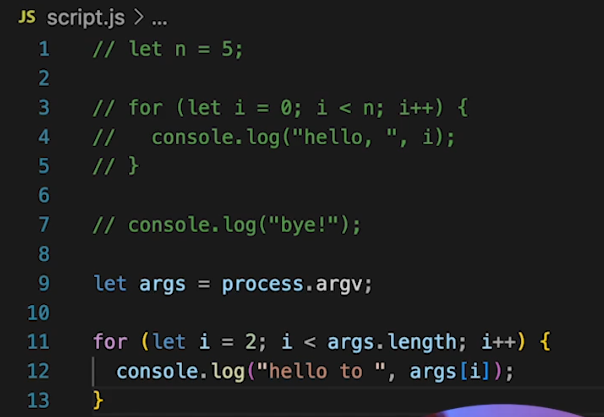
****

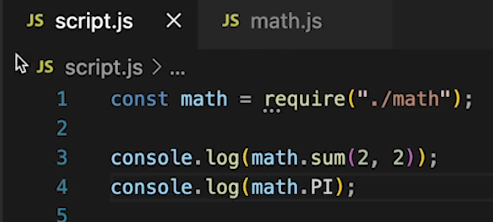
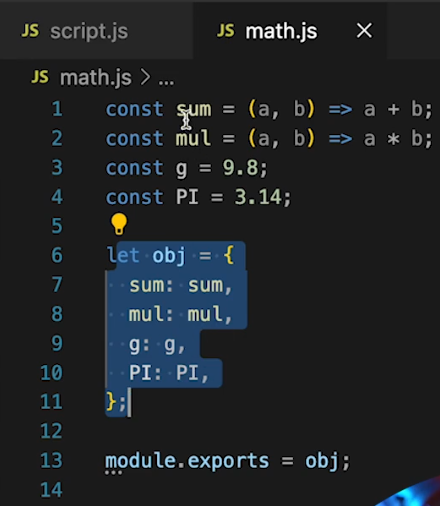
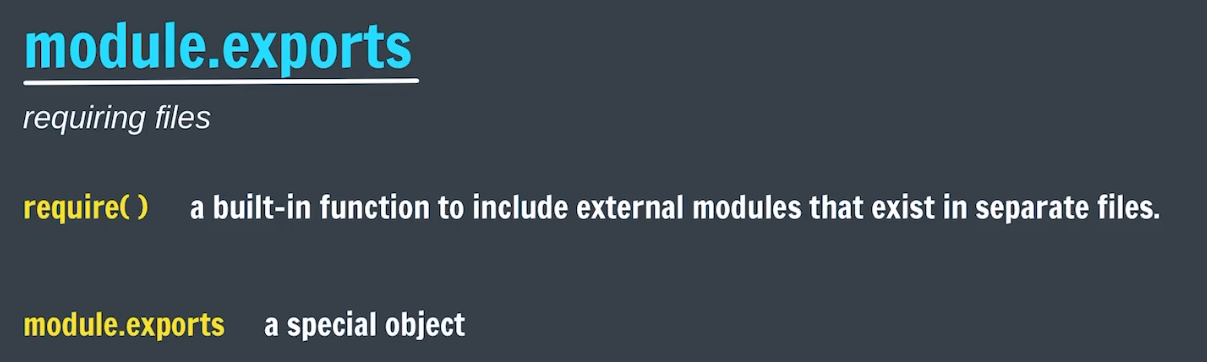
****

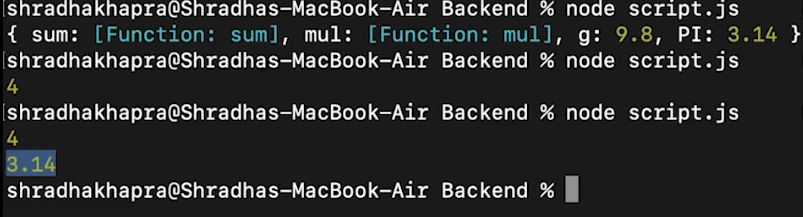
**These Arguments are passed during running a file using node command**

****

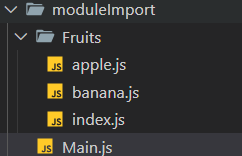
**We can also access these arguments:**

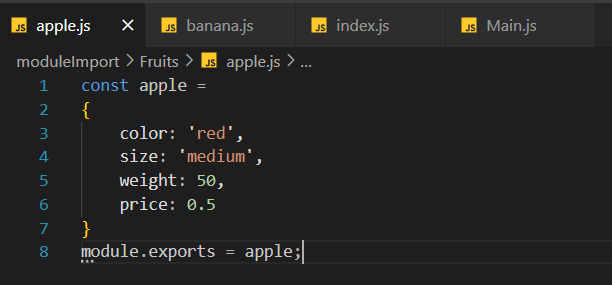
****

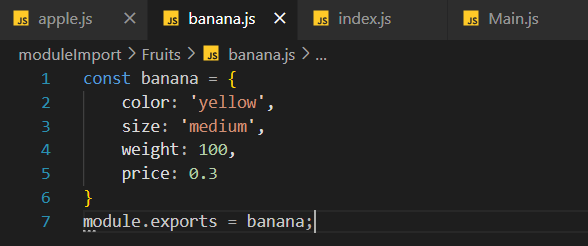
****

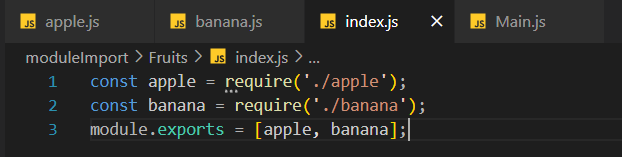
****

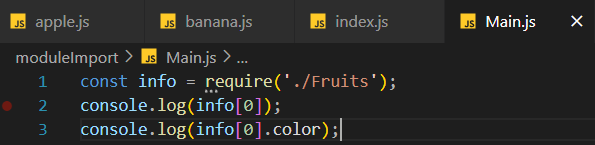
**How to export a directory: here we are importing whole fruits directory in main.js.**

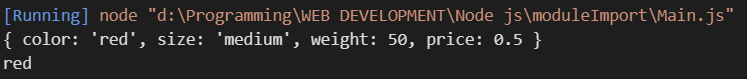
****

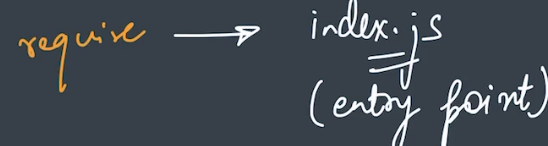
****

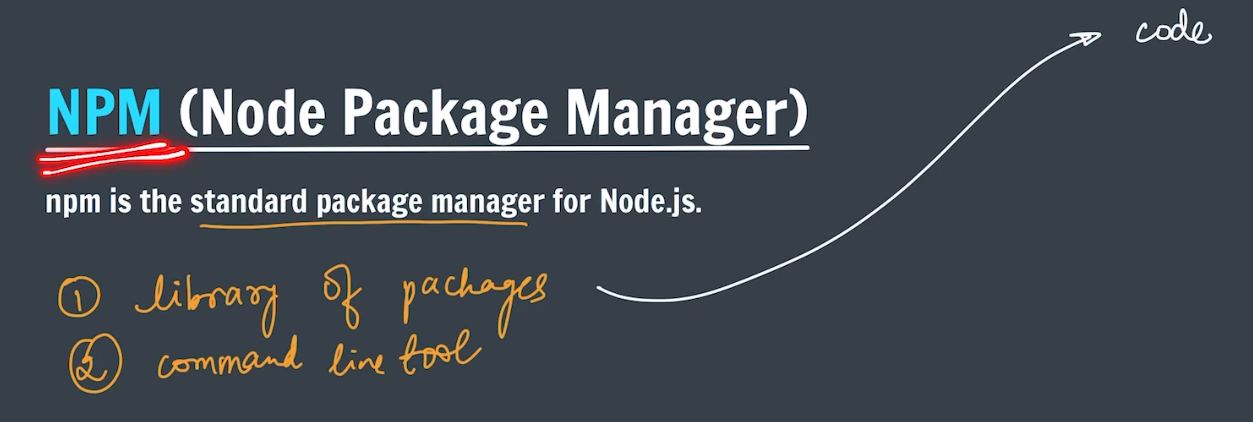
****

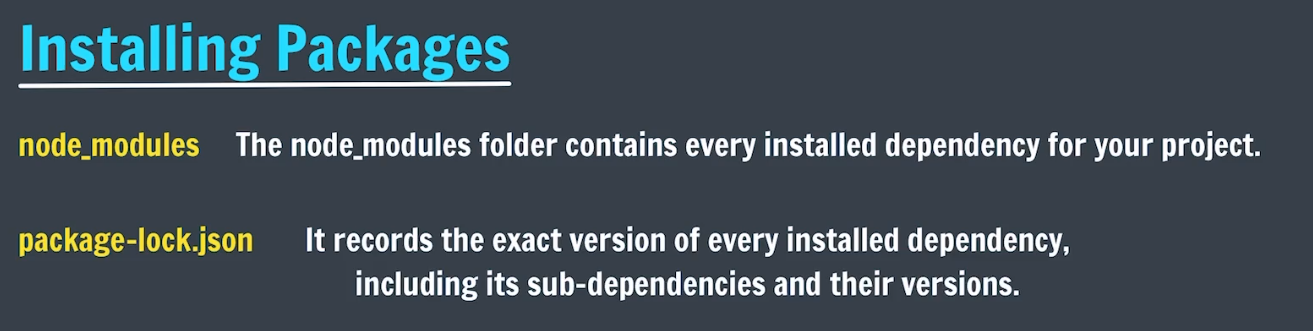
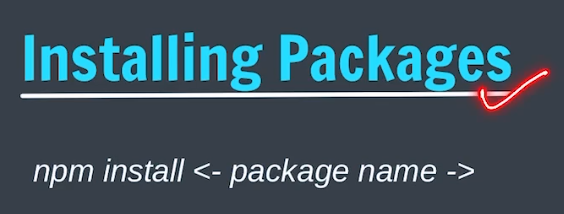
****

****

****

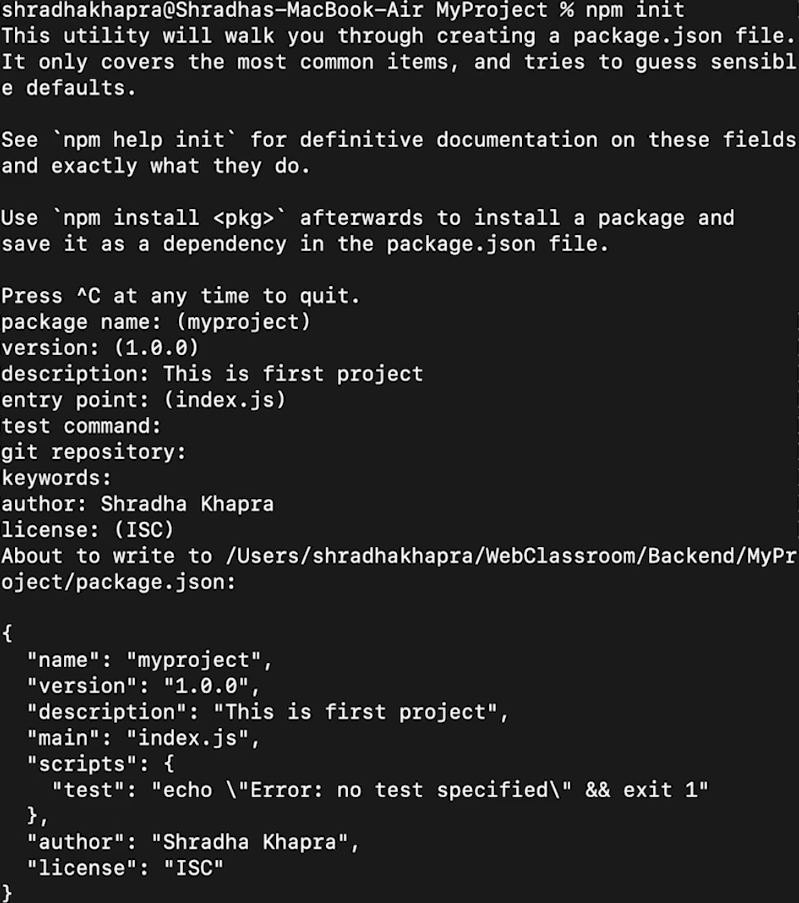
****

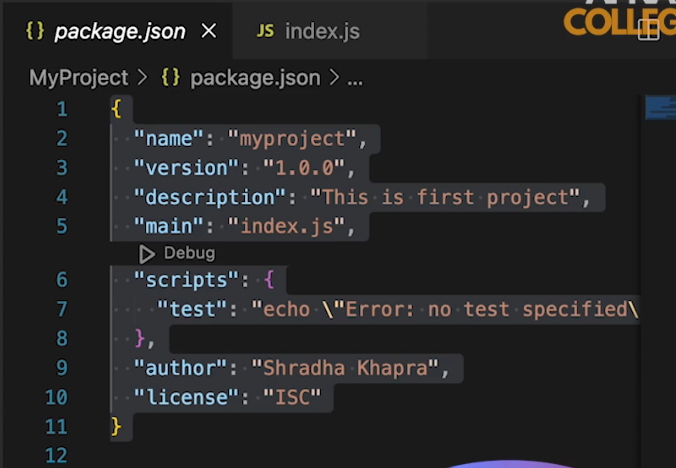
****

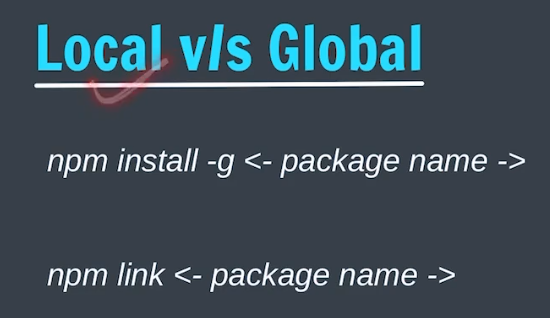
****

****

**To create a package.json for our project we run, npm init and then fill the relevent information**

****

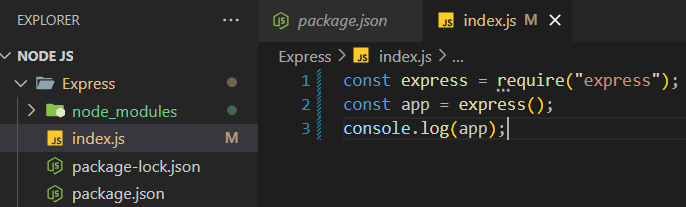
****

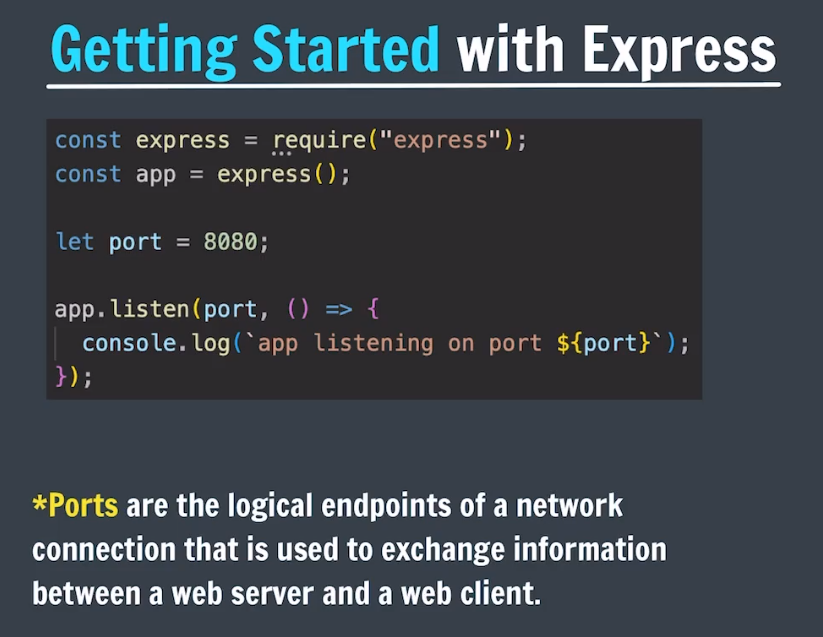
****

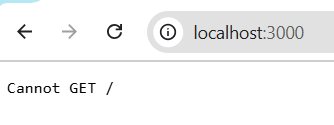
**We can also install packages globally using npm install -g packageName, after this we can use it in any directory but before using we have to go to that directory and then link that package using the command npm link packageName. But this is not preffered we should always install locally.**

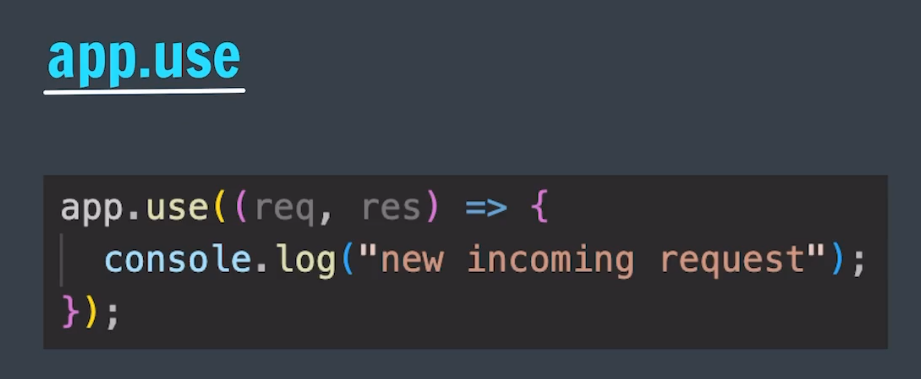
****

**Here ‘app’ allows us to create server side webApp and it is an object**

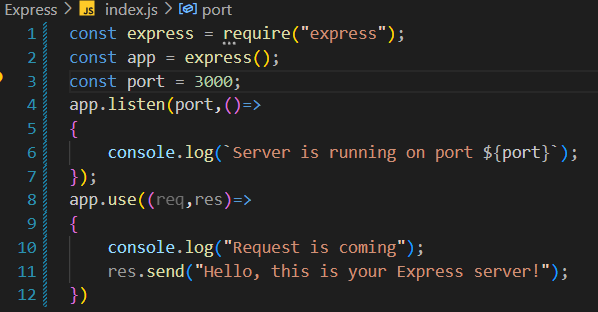
****

****

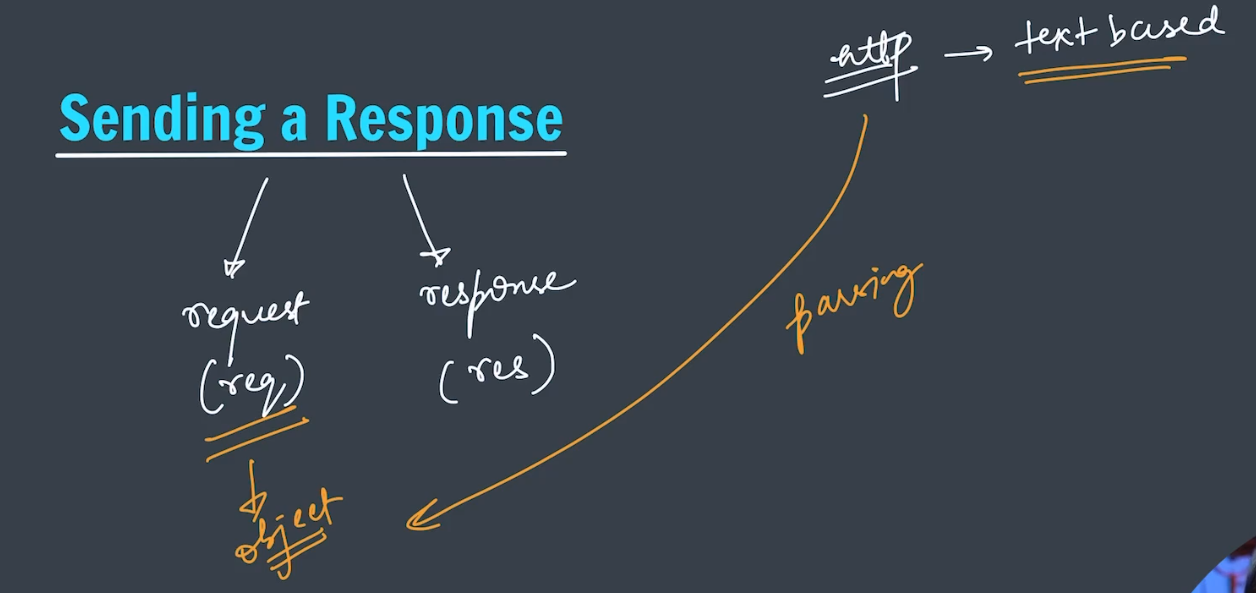
* **Express creates a web server for listining the incomming API requests with the help of a function of ‘app’ I.e Listens().**
* **When the server starts, “app is listening on port 8080” will be printed**
* **To start server = node index.js**
* **After starting server, to send a request on that server go to localhost:3000**
* ** WE get this because we have not written code for handling any server request.**
* **To stop server = Ctrl + C**

****

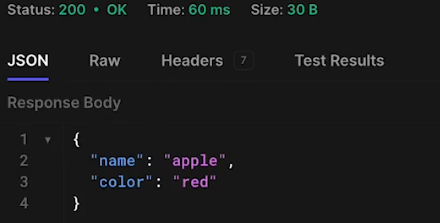
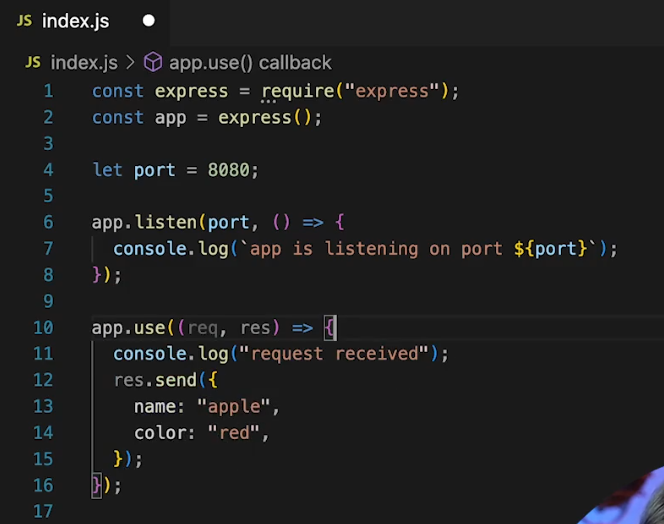
* **This method listens for all types of requests(get, post etc) and then according to the request callbacks inside it executes**

****

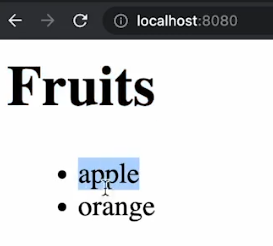
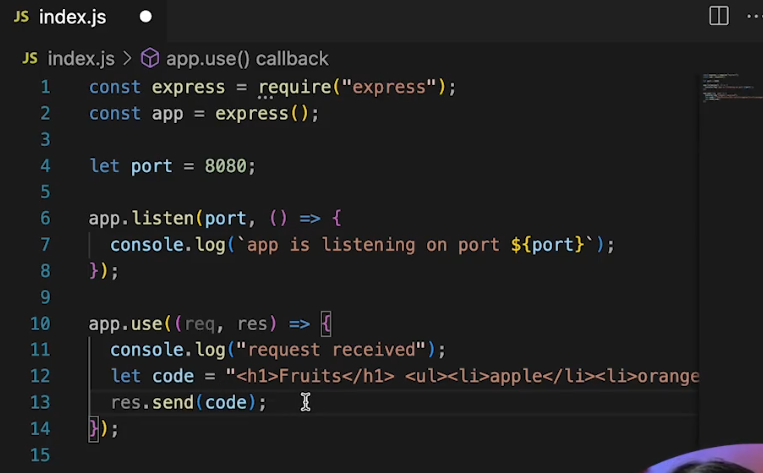
* **Now when we get a request on our server “request received” will prints on our system terminal and “Hello, this is your Express server!” will be the response to the req sender.**

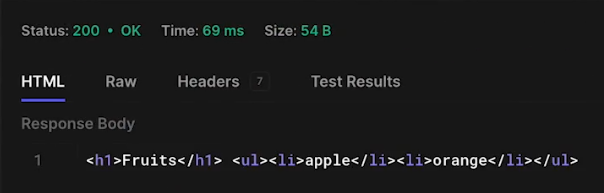
****

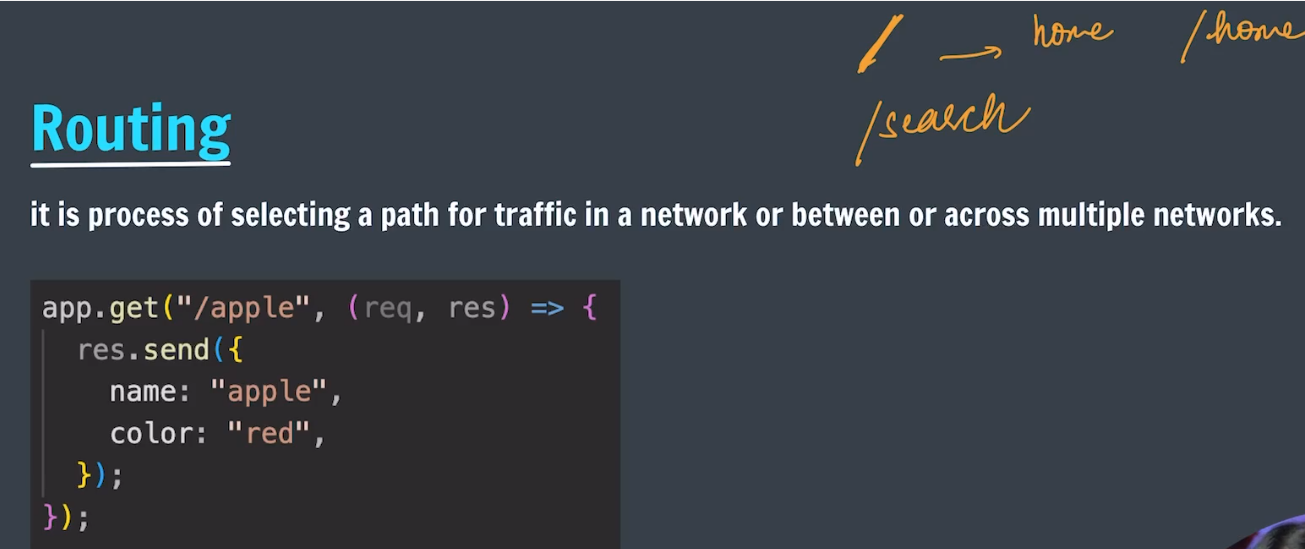
**Sending object as a response:**

****

**Sending HTML as response:**

****

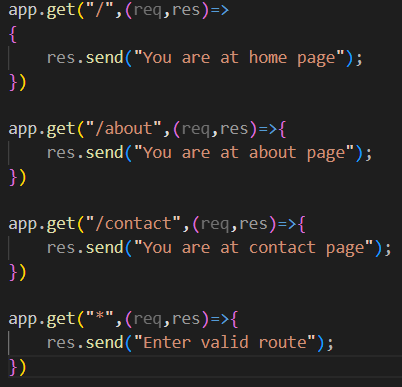
****

****

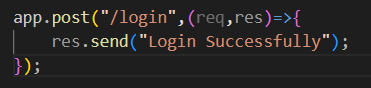
* **Routing helps us to define on which path we are sending the request over the network**
* **So that we can send different different responses from different routes/paths**
* **If we use app.use() we will get same response on all the paths which is specified inside the use().**
* **That’s why we use app.get(), it allows us to send different response at different routes.**

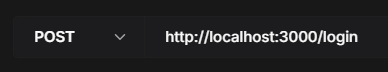


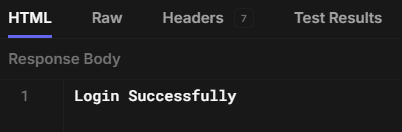
* **For any other path not mentioned we can specify custom response using “\*”:**

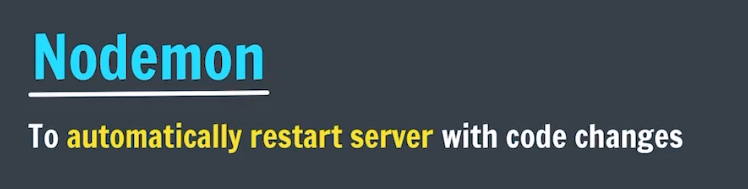


* **Simillary you can also send the post request:**

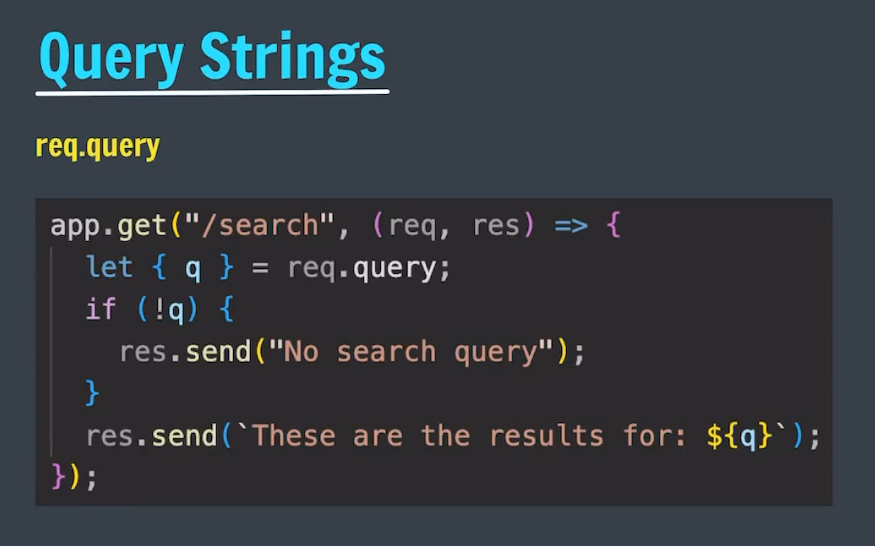








* **npm install -g nodemon**
* **Nodemon index.js**



* **Queries passed during get request are accessed like this: [www.npmjs.com/search?q=figlet&name=faziel,](https://www.npmjs.com/search?q=figlet,) here figlet, name=faziel are the query strings**

