

BACKEND

Intro to backend

Full stack = Frontend + Backend

- Backend is invisible to the user but full of logics that make the web app work.



Inspired from App Brewery

Server Application Database

- Server – 24/7. **Listens to the browser request** all over the world. Any computer that is connected to the network can act as a server. Local host is also server using our own computer.
- Application – The logic for the app. It determines **how to respond to the request** from the browser. User clicks a button and that button returns an HTML page from the server to the application. Other than HTML, Data or status code (404 error) can be sent to. Application does it. Server and Application communicate via API.
- Database – Where all **the data is stored** for the application to use to serve client's request.
- Watch - <https://www.youtube.com/watch?v=gOghS3BmaxI>



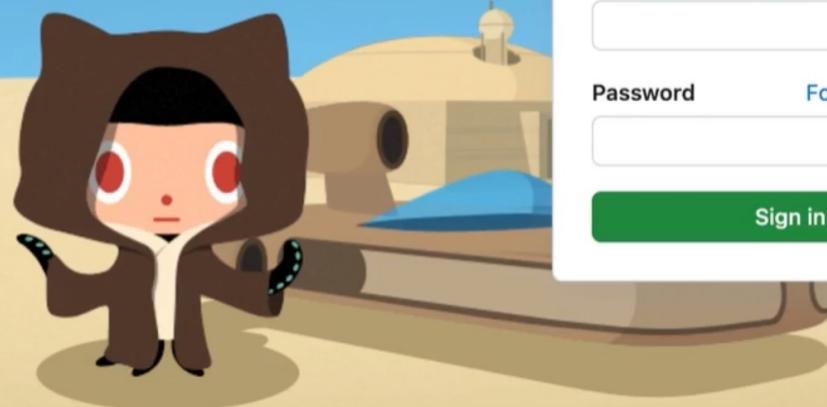
Product ▾ Solutions ▾ Open Source ▾ Pricing

Search GitHub

Sign in Sign up

404

This is not the
web page you
are looking for.



Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

Sign in

Find code, projects, and people on GitHub:

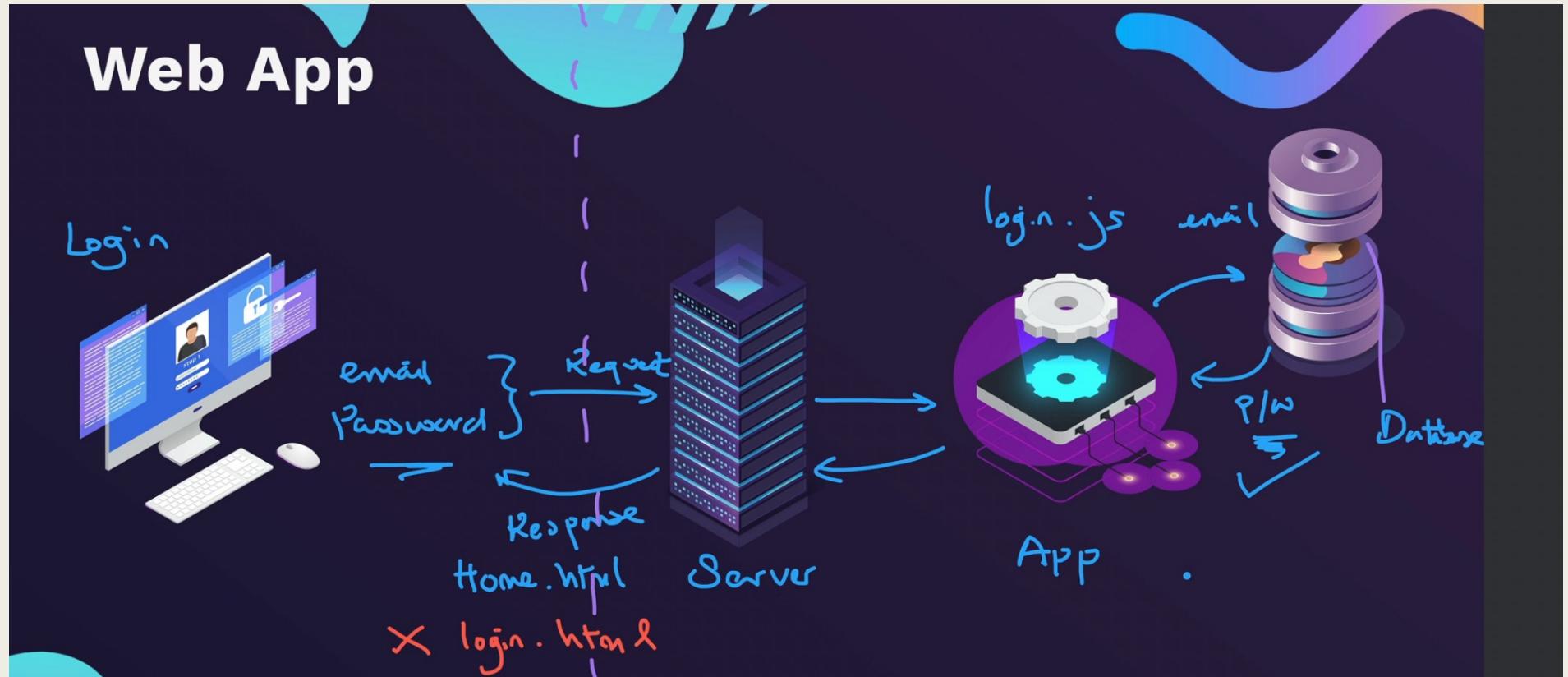
Search

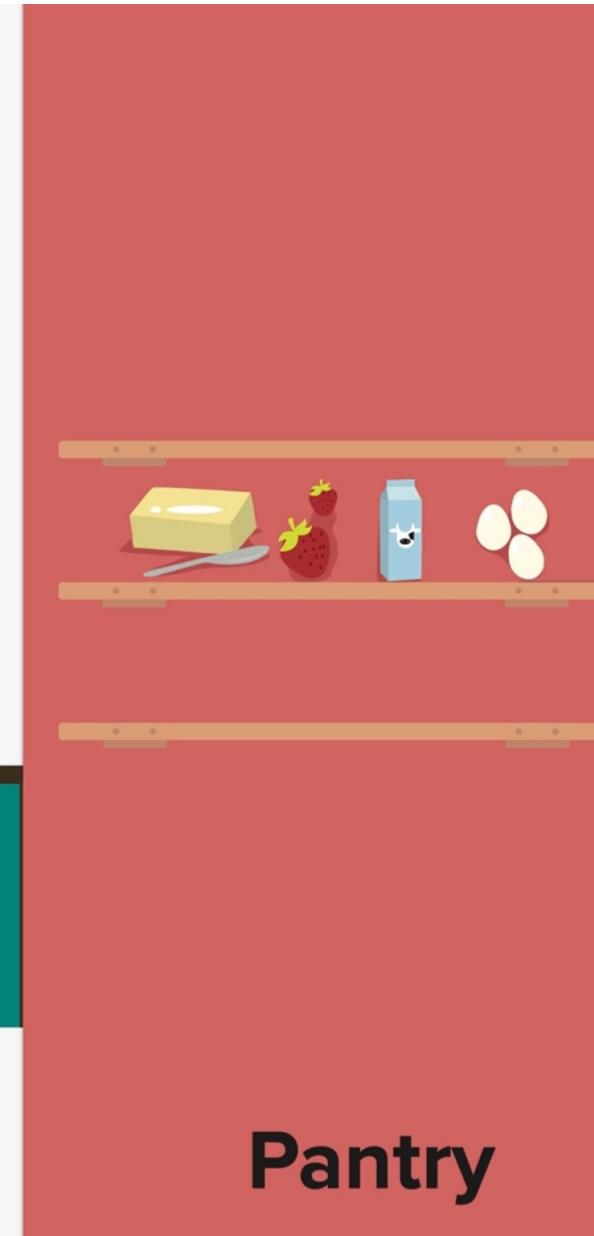
[Contact Support](#) – [GitHub Status](#) – [@githubstatus](#)

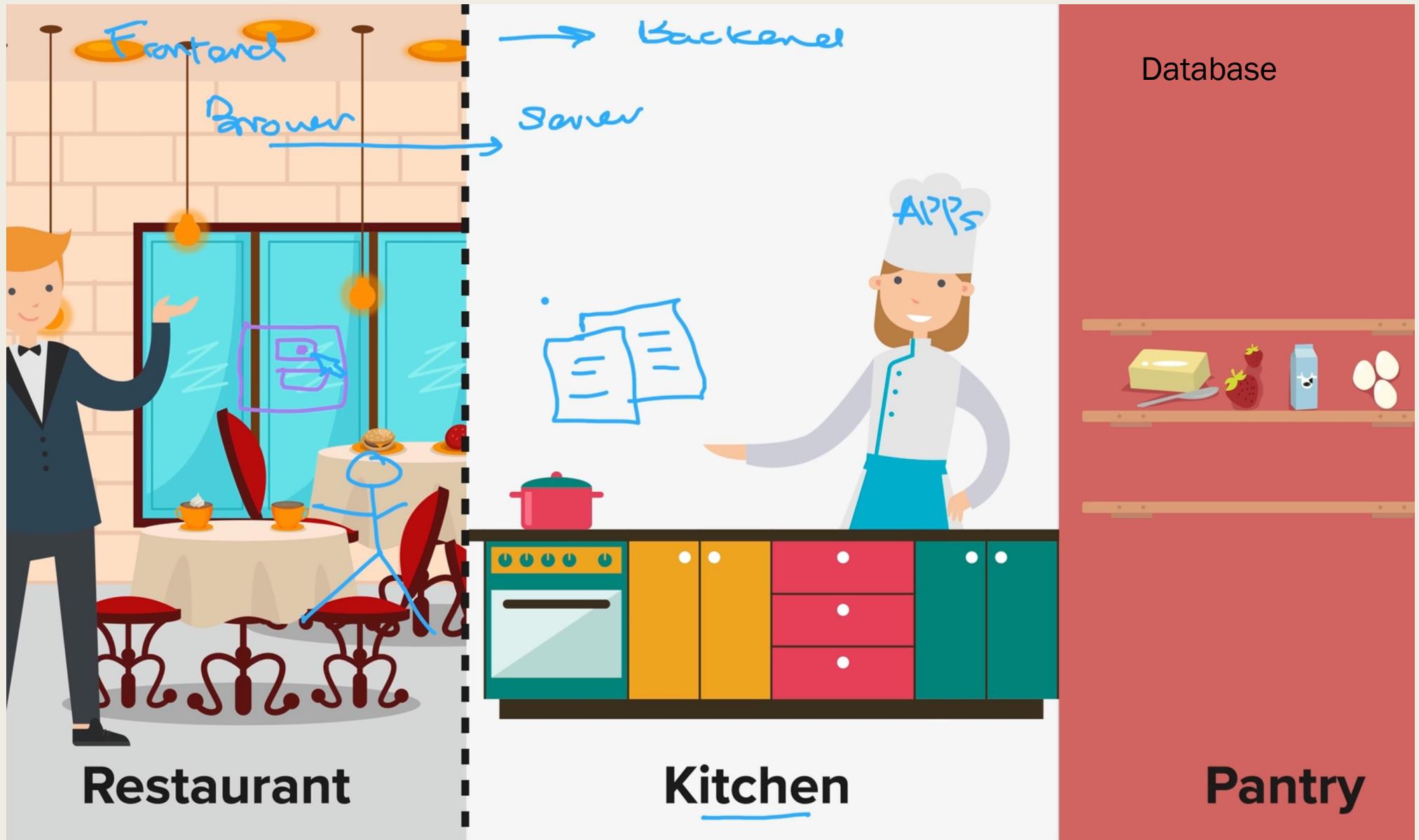
From Stack Overflow

- **Server** - Server on which your website is hosted. This server will have installed web servers such as IIS, apache, etc.
- **Application** - Logics of your created applications which are utilizing your database, web service, etc. This application server will host business layer (wrapped with web services), scheduled jobs, windows services, etc.
- **Database** - Database server will have your one or more database hosted such as Oracle, Sql Server, MySql, etc.

What's the difference between webpage/website and web app?



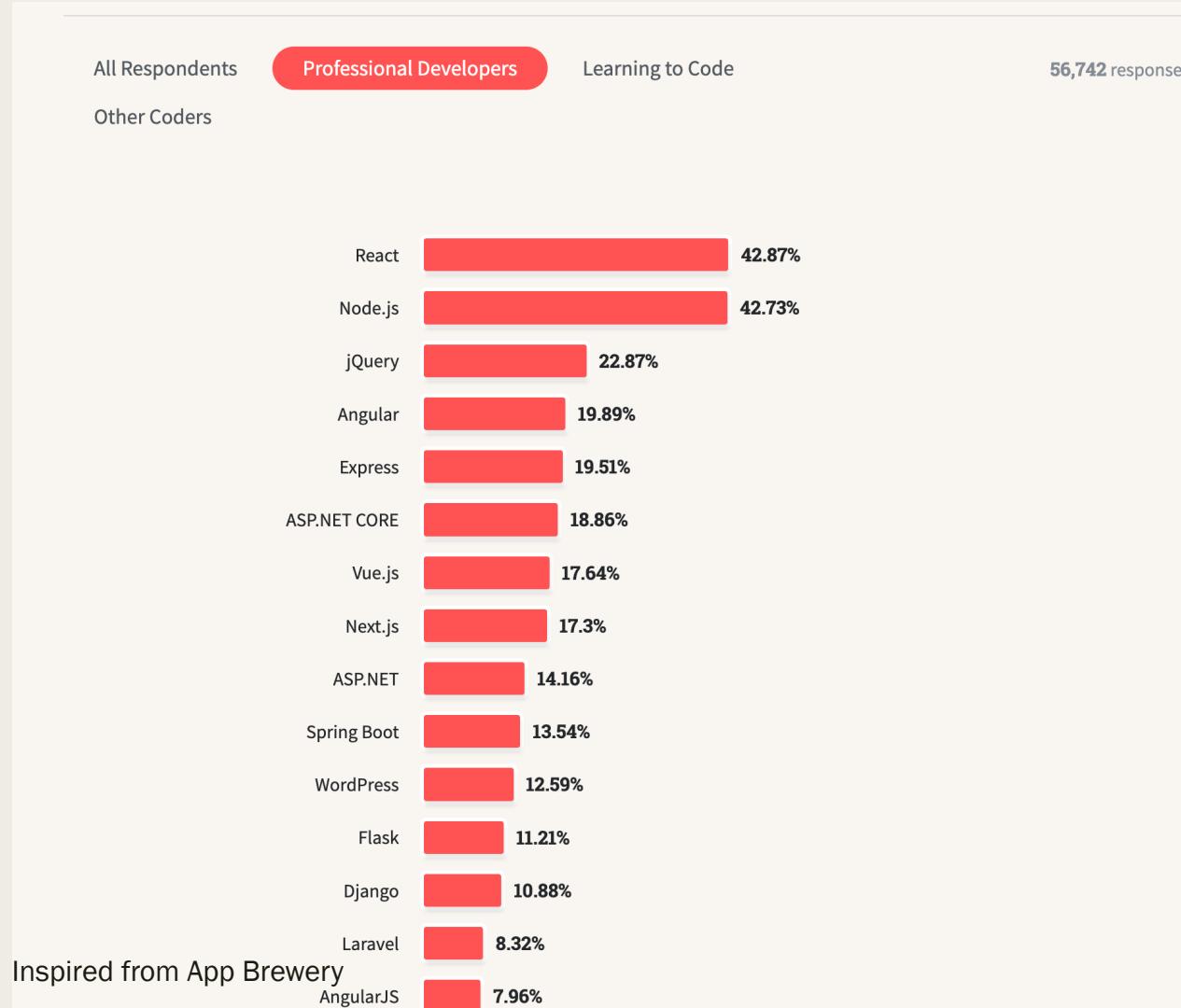




IP address, DNS Server

- Watch - <https://www.youtube.com/watch?v=5o8CwafCxnU>

Why Node : Stack overflow survey



Why do we need a framework, why can't we code everything from scratch?

- Think cooking.

What's node

- Is an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.
- Official documentation - <https://nodejs.org/en/about>
- Originally JS was developed to work on browser only. Node made JS work on servers too.

Optional: Synchronous vs Asynchronous programming

- Watch - <https://www.youtube.com/watch?v=Kpn2ajSa92c>

The image shows two side-by-side code editors and their respective consoles.

Top Example (Synchronous):

```
1 let a = 1
2 let b = 2
3
4 console.log('Synchronous')
5
6 console.log(a)
7 console.log(b)
```

Console Output:

Output	File	Line
Synchronous	async.js	4
1	async.js	6
2	async.js	7

Bottom Example (Asynchronous):

```
1 let a = 1
2 let b = 2
3
4 setTimeout(function() {
5   console.log('Async')
6 }, 100)
7
8 console.log('Synchronous')
9
10 console.log(a)
11 console.log(b)
```

Console Output:

Output	File	Line
Synchronous	async.js	8
1	async.js	10
2	async.js	11
Async	async.js	5

Inspired from <https://www.youtube.com/watch?v=Kpn2ajSa92c>

Install Node.js on Mac

■ Installation Steps

1. Download the Mac installer from the [Nodes.js® web site](#).
 2. Choose the Long Term Support version that's shown on the left. (recommended version)
 3. **Run the installer** (double-click on the .pkg file you downloaded in the previous step.)
 4. **Follow the prompts in the installer** (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).
 5. Confirm that node has been installed successfully by opening a Hyper terminal and entering the command `node --version`
- You should see the version of node you have just installed.

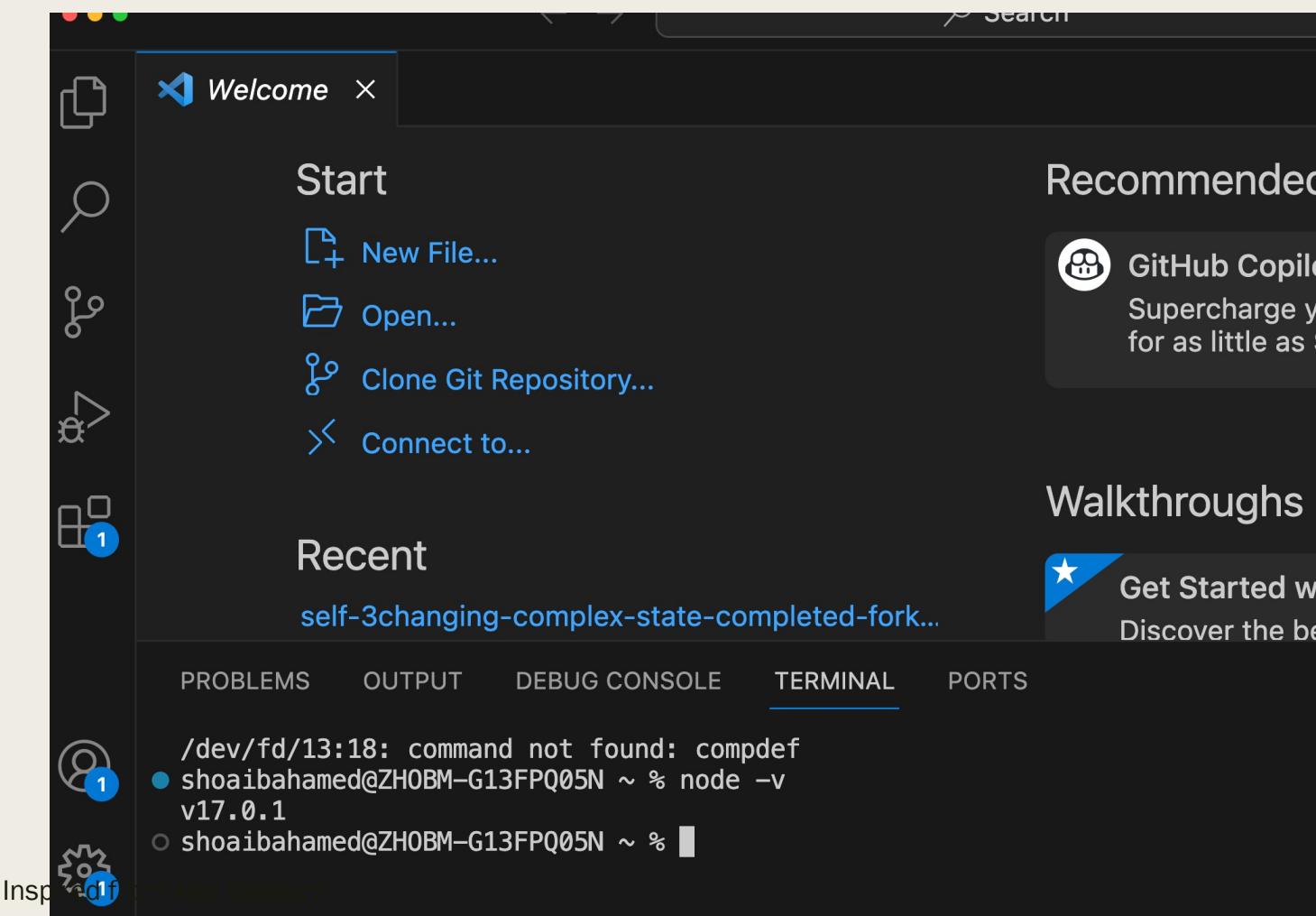
Install Node.js on Windows

■ Installation Steps

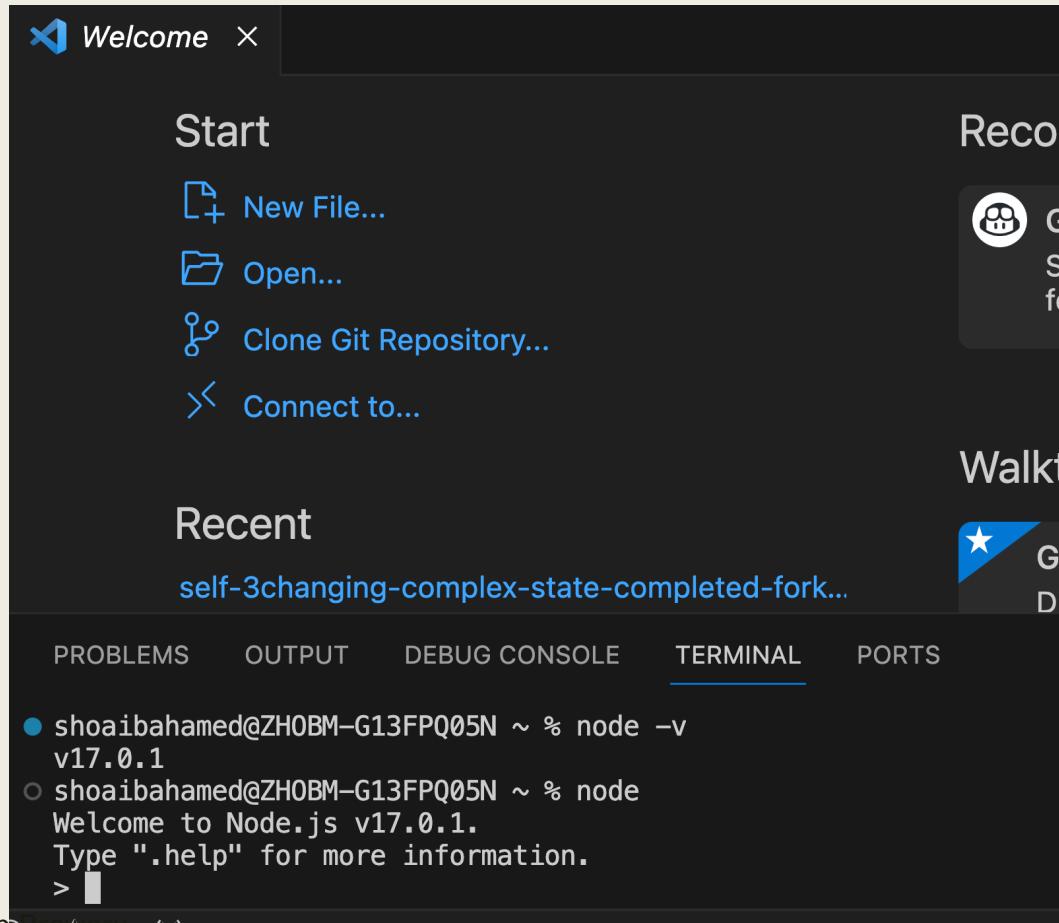
1. Download the Windows installer from the [Nodes.js® web site.](#)
 2. Choose the LTS version that's shown on the left. (Recommended version).
 3. Run the installer (the .msi file you downloaded in the previous step.)
 4. Follow the prompts in the installer (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).
 5. Restart your computer. You won't be able to run Node.js® until you restart your computer.
 6. Confirm that Node has been installed successfully on your computer by opening a Hyper terminal and typing in the commands node --version
- You should see the version of node you just installed.

Next: Using node

Checking node version in VS code



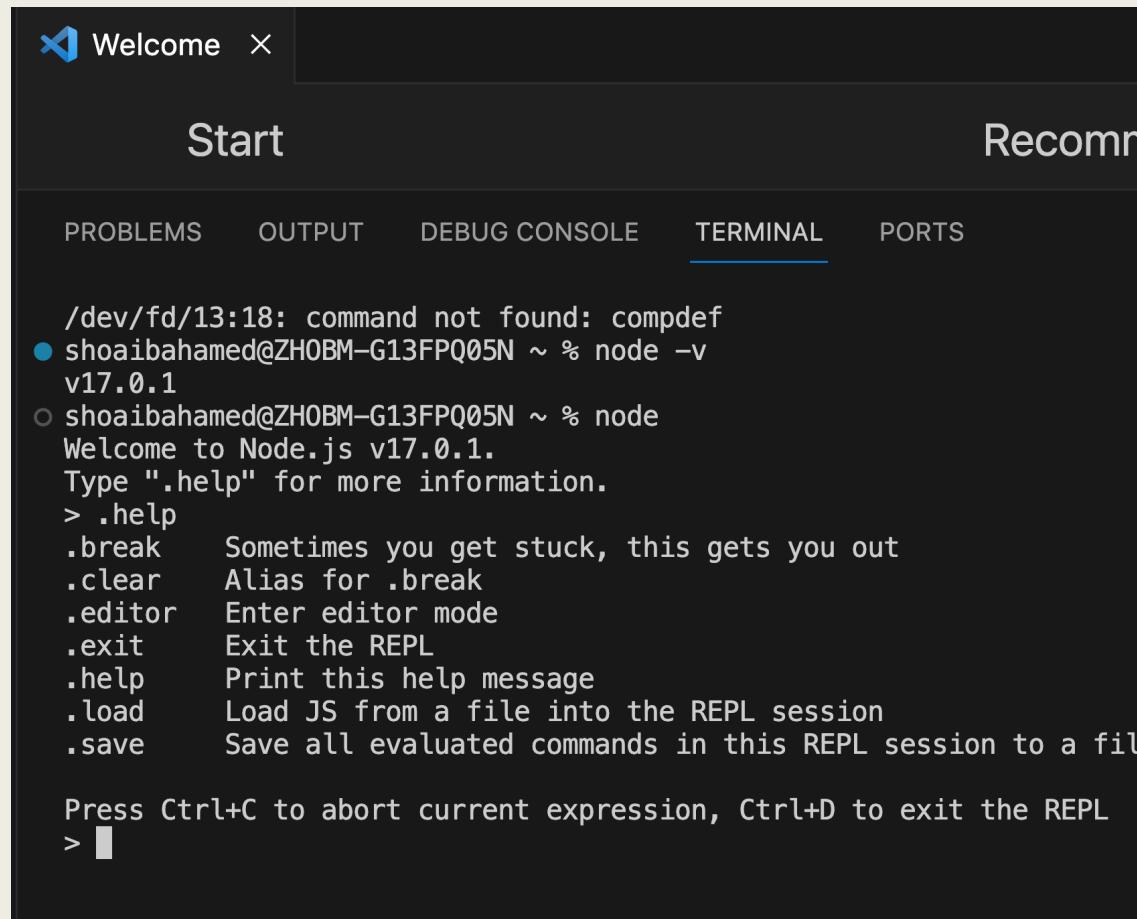
Node REPL : type node



A screenshot of the Visual Studio Code interface. The title bar says "Welcome". The left sidebar has sections for "Start" (New File..., Open..., Clone Git Repository..., Connect to...) and "Recent" (self-3changing-complex-state-completed-fork...). The bottom navigation bar shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The main area is a terminal window displaying the following text:

```
shoibahamed@ZH0BM-G13FPQ05N ~ % node -v
v17.0.1
shoibahamed@ZH0BM-G13FPQ05N ~ % node
Welcome to Node.js v17.0.1.
Type ".help" for more information.
> 
```

.help shows basic node commands



A screenshot of a terminal window titled "Welcome". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal content shows a command-line session:

```
/dev/fd/13:18: command not found: compdef
● shoaibahamed@ZHOBM-G13FPQ05N ~ % node -v
v17.0.1
○ shoaibahamed@ZHOBM-G13FPQ05N ~ % node
Welcome to Node.js v17.0.1.
Type ".help" for more information.
> .help
  .break    Sometimes you get stuck, this gets you out
  .clear    Alias for .break
  .editor   Enter editor mode
  .exit     Exit the REPL
  .help     Print this help message
  .load     Load JS from a file into the REPL session
  .save     Save all evaluated commands in this REPL session to a file

Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
> █
```

Read Evaluate Print Loop

```
✉ shoaibahamed@ZHOBM-G13FPQ05N ~ % 5+7
      zsh: command not found: 5+7
○ shoaibahamed@ZHOBM-G13FPQ05N ~ % node
      Welcome to Node.js v17.0.1.
      Type ".help" for more information.
> 5+7
12
>
```

Node is a JS runtime, we can write any JS in node environment

```
Uncaught ReferenceError
> let a = 5;
undefined
> a
5
> 
```

Writing node in text editor

- Create a directory.
- Create a index.js file in that directory.
- In the index.js file write `console.log('Hello from node');`
- In the terminal cd to the directory
- In the terminal type `node index.js`

Solution:

The screenshot shows a dark-themed instance of Visual Studio Code. In the Explorer sidebar, a folder named "2.1 USING NODE" is expanded, showing an "index.js" file. The code editor displays the following content:

```
JS index.js
1 console.log("hello from node");
```

The Terminal tab is active, showing the following command-line session:

```
/dev/fd/13:18: command not found: compdef
● shoaibahamed@ZHOBM-G13FPQ05N 2.1 Using Node % cd '/Users/shoaibahamed/Downloads/2
.1 Using Node'
● shoaibahamed@ZHOBM-G13FPQ05N 2.1 Using Node % node index.js
hello from node
○ shoaibahamed@ZHOBM-G13FPQ05N 2.1 Using Node %
```

Practice challenge: Create another JS file inside of the directory then write and run some code there.

Native Node modules

- Modules provided by node to make build applications easy.
- Documentation - <https://nodejs.org/dist/latest-v20.x/docs/api/>
- Download the starter folder from here –
<https://drive.google.com/file/d/1jPT6DePMAaK0wXXVhCnHURJdccIOW2F7/view?usp=sharing>
- We will use this write function (`fs.writeFile(file, data[, options], callback)`) - <https://nodejs.org/dist/latest-v20.x/docs/api/fs.html#fswritefd-buffer-options-callback>

The `mode` option only affects the newly created file. See `fs.open()` for more details.

```
import { writeFile } from 'node:fs';
import { Buffer } from 'node:buffer';

const data = new Uint8Array(Buffer.from('Hello Node.js'));
writeFile('message.txt', data, (err) => {
  if (err) throw err;
  console.log('The file has been saved!');
});
```

If `options` is a string, then it specifies the encoding:

The screenshot shows a dark-themed interface of the Visual Studio Code code editor. At the top, there are two tabs: 'JS index.js' (active) and 'JS solution.js'. Below the tabs, the code editor displays the following Node.js script:

```
1 const fs = ...
2 fs.writeFile("message.txt","Hello from NodeJs",(err) => {
3     if (err) throw err;
4     console.log('The file has been saved!');
5 })
```

Below the code editor is a 'Output (⌃⌘U)' button. Underneath the editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS. The TERMINAL tab is active, showing the output of running the script:

- shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules % node index.js
- The file has been saved!
- shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules % █

On the right side of the terminal area, there is a dropdown menu with options: > zsh + ▾.

Message.txt file got created.

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays files: index.js (selected), message.txt, and solution.js. In the center, the code editor shows index.js with the following content:

```
const fs = require("fs")
fs.writeFile("message.txt","Hello from NodeJs",(err) => {
  if (err) throw err;
  console.log('The file has been saved!');
});
```

At the bottom, the Terminal tab is active, showing the command line output:

```
/dev/fd/13:18: command not found: compdef
shoaibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules % cd '/Users/shoaibahamed/2.2 Native Modules'
shoaibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules % node index.js
The file has been saved!
shoaibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules %
```

Challenge: read file

- Following this (`fs.readFile(path[, options], callback)`)- <https://nodejs.org/dist/latest-v20.x/docs/api/fs.html#fsreadfilepath-options-callback>
- Write another line in message.txt. Read message.txt file.

Notice what gets printed!

The screenshot shows a dark-themed instance of Visual Studio Code. On the left, the Explorer sidebar lists three files under 'OPEN EDITORS': 'index.js' (selected), 'message.txt', and 'solution.js'. It also shows a folder '2.2 NATIVE MODULES' containing 'index.js', 'message.txt', and 'solution.js'. The 'index.js' editor tab is active, displaying the following code:

```
1  const fs = require("fs")
2  // fs.writeFile("message.txt","Hello from NodeJs",(err) => {
3  //     if (err) throw err;
4  //     console.log('The file has been saved!');
5  // });
6
7  fs.readFile('./message.txt', (err, data) => {
8      if (err) throw err;
9      console.log(data);
10 });

The terminal at the bottom shows the output of running the script:
```

```
s/2.2 Native Modules'
shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules % node index.js
The file has been saved!
shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules % pwd
/Users/shoaibahamed/Downloads/2.2 Native Modules
shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules % node index.js
<Buffer 48 65 6c 6c 6f 20 66 72 6f 6d 20 4e 6f 64 65 4a 73 21 0a 48 65 6c 6c 6f 20
66 72 6f 6d 20 49 6e 73 74 72 75 63 74 6f 72 21>
shoaibahamed@ZH0BM-G13FPQ05N 2.2 Native Modules %
```

Why is this buffer?

```
readFile('/etc/passwd', (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

The callback is passed two arguments `(err, data)`, where `data` is the contents of the file.

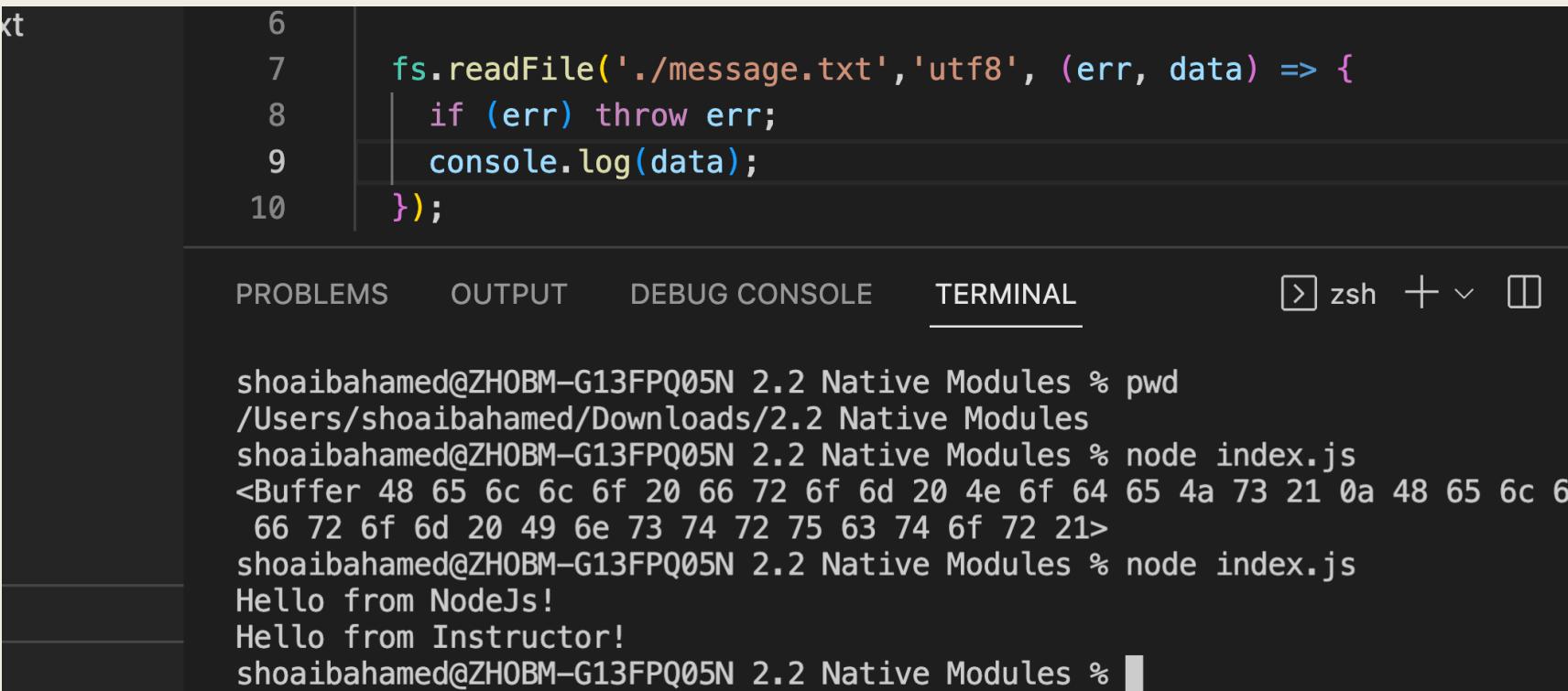
If no encoding is specified, then the raw buffer is returned.

If `options` is a string, then it specifies the encoding:

```
import { readFile } from 'node:fs';

readFile('/etc/passwd', 'utf8', callback);
```

Specify 'utf8'



A screenshot of a terminal window within a dark-themed code editor interface. At the top, there's a code editor pane showing a snippet of Node.js code:

```
xt
6
7   fs.readFile('./message.txt','utf8', (err, data) => {
8     if (err) throw err;
9     console.log(data);
10    });

```

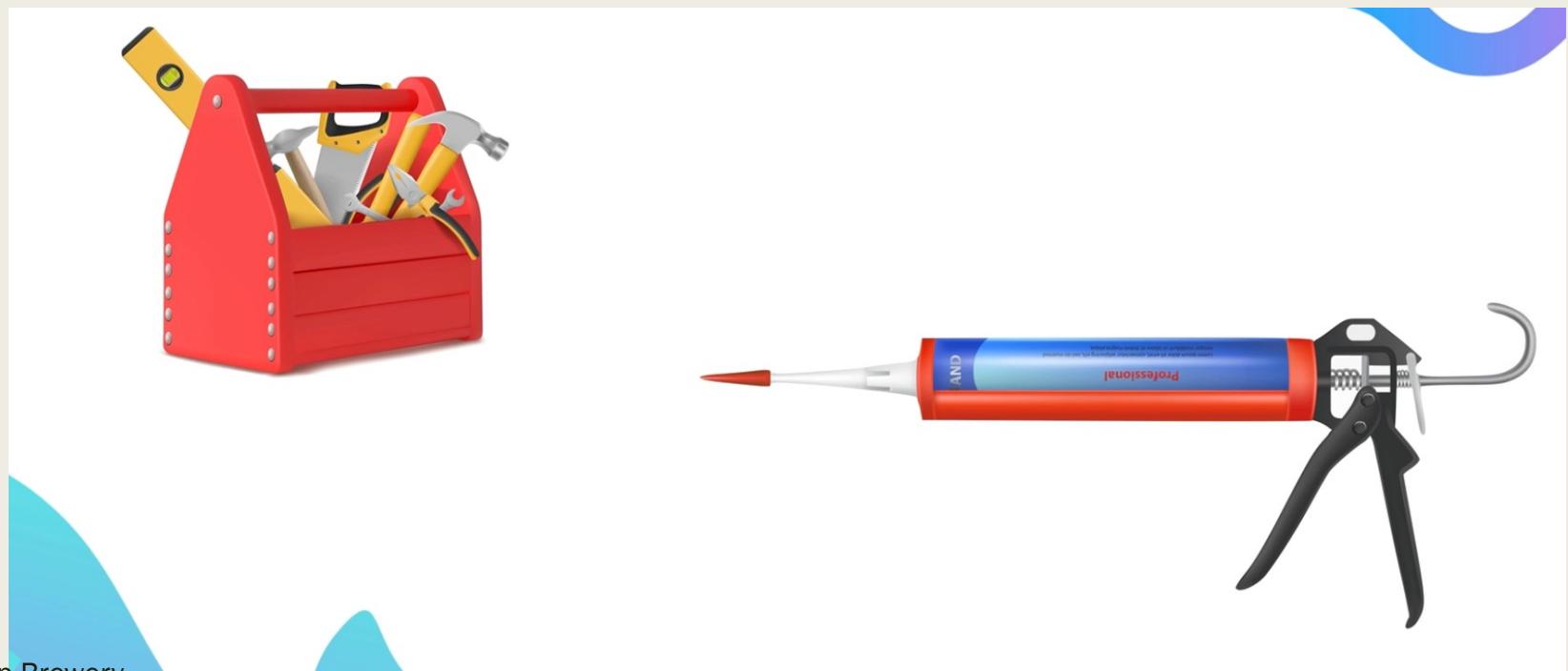
Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, indicated by a underline. To the right of the tabs is a toolbar with icons for zsh, a plus sign, a downward arrow, and a square.

The terminal output shows the following sequence of commands and responses:

```
shoibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules % pwd
/Users/shoibahamed/Downloads/2.2 Native Modules
shoibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules % node index.js
<Buffer 48 65 6c 6c 6f 20 66 72 6f 6d 20 4e 6f 64 65 4a 73 21 0a 48 65 6c 6
66 72 6f 6d 20 49 6e 73 74 72 75 63 74 6f 72 21>
shoibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules % node index.js
Hello from NodeJs!
Hello from Instructor!
shoibahamed@ZHOBM-G13FPQ05N 2.2 Native Modules %
```

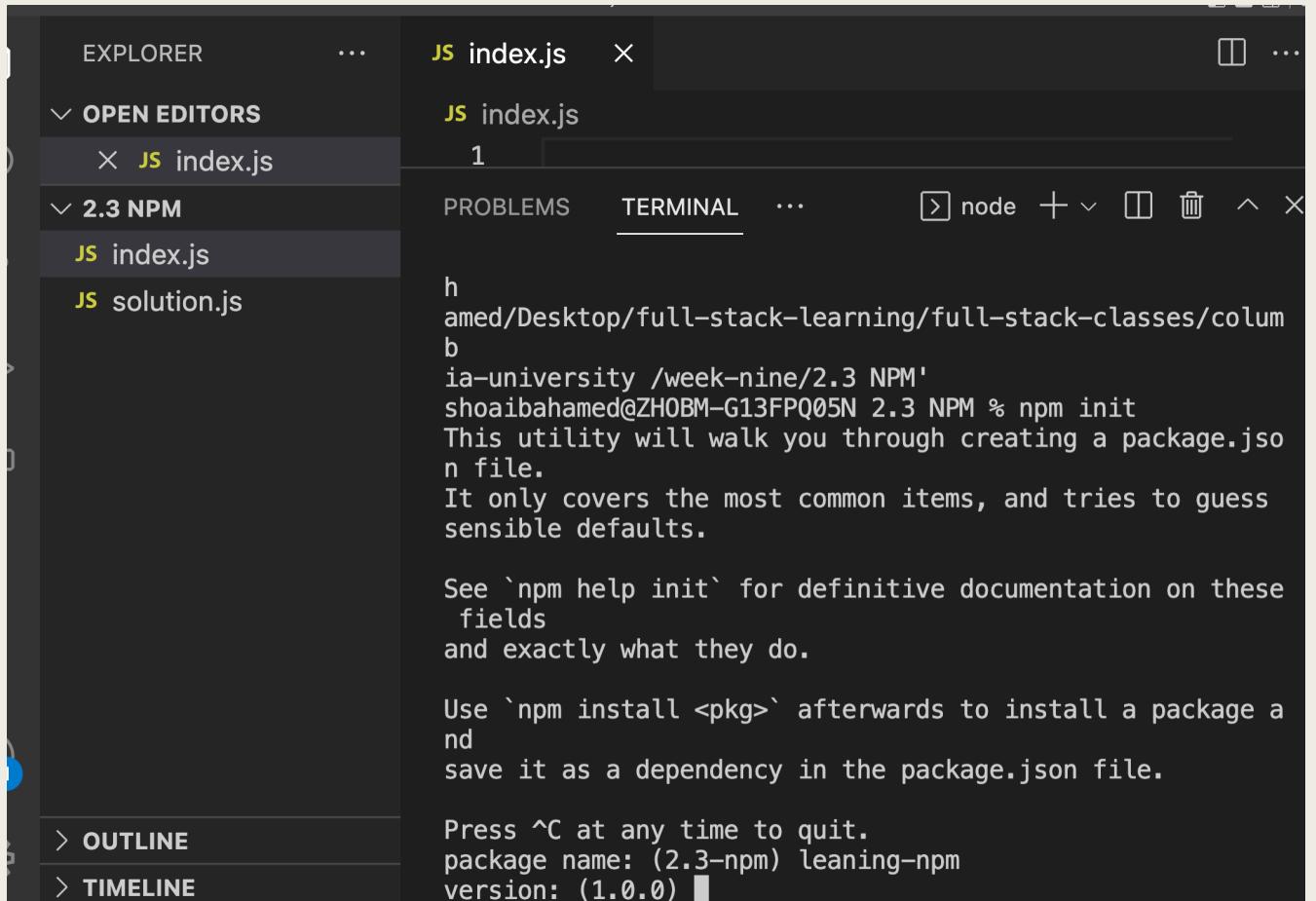
Learning Node Package Manager(NPM)

- Open sourced node packages.
- A place by github to find the modules other developers built.
- Native node modules are the basic/common modules. NPM more diversified/special modules.



Let's configure and use npm!

- NPM come pre bundled with node meaning if you have node installed, you have npm installed as well.
- Download the initial directory. Open up index.js file in VS code. Cd inside the directory via terminal. Directory link - <https://drive.google.com/file/d/1EUTL5skqrfAjXH1EA-wzu3uplxvflFOo/view?usp=sharing>
- Then type **npm init** in your terminal.
- It will ask some questions:



The screenshot shows the Visual Studio Code interface. The left sidebar has sections for 'OPEN EDITORS' (with files 'index.js' and 'solution.js') and '2.3 NPM' (with file 'index.js'). The main area is the 'TERMINAL' tab, which displays the following text:

```

h
amed/Desktop/full-stack-learning/full-stack-classes/colum
b
ia-university /week-nine/2.3 NPM'
shoabahamed@ZHOBM-G13FPQ05N 2.3 NPM % npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess
sensible defaults.

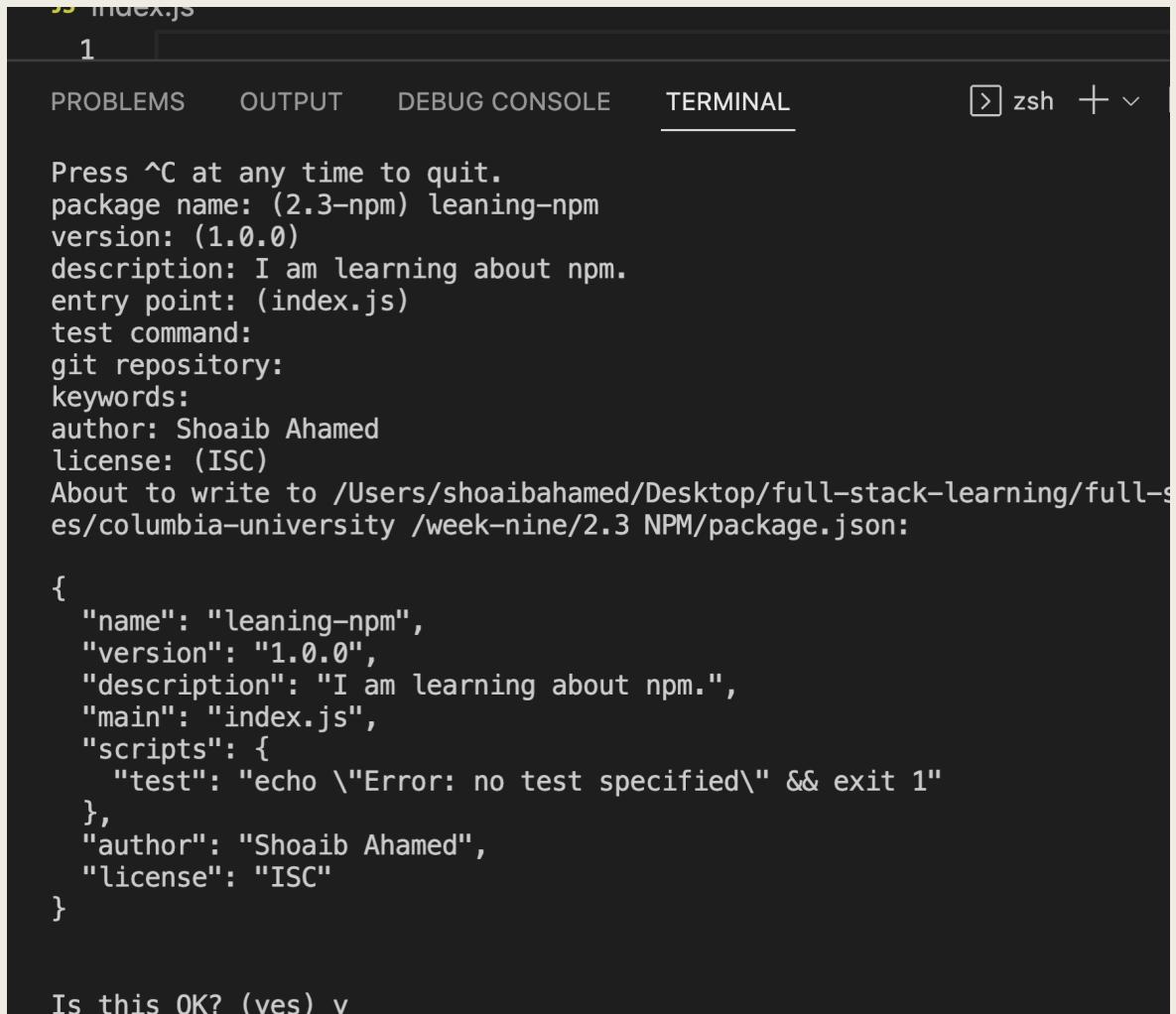
See `npm help init` for definitive documentation on these
fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (2.3-npm) leaning-npm
version: (1.0.0)

```

It will ask some questions:



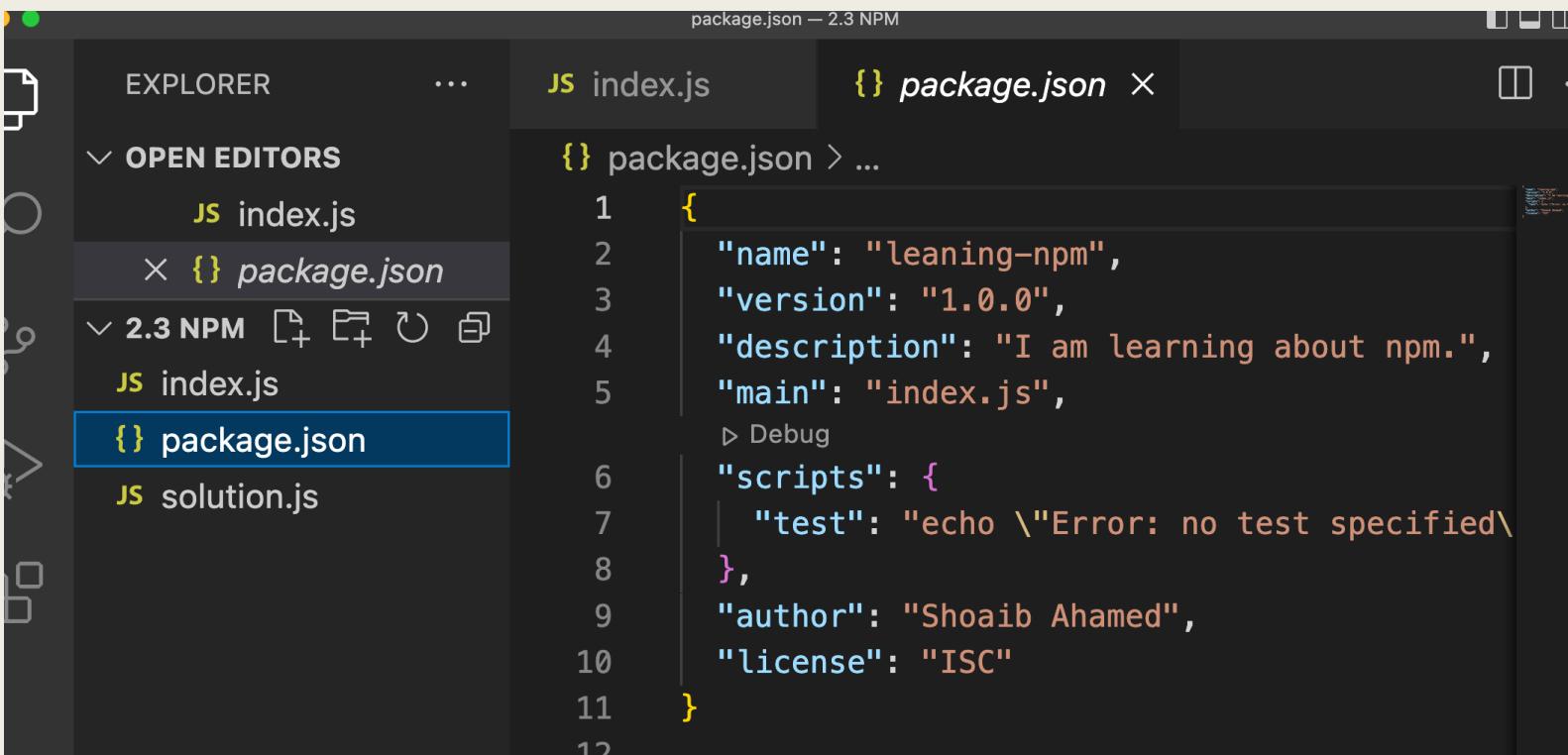
The screenshot shows a terminal window with the following content:

```
index.js
1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh + ▾
Press ^C at any time to quit.
package name: (2.3-npm) leaning-npm
version: (1.0.0)
description: I am learning about npm.
entry point: (index.js)
test command:
git repository:
keywords:
author: Shoaib Ahamed
license: (ISC)
About to write to /Users/shoaibahamed/Desktop/full-stack-learning/full-s
es/columbia-university /week-nine/2.3 NPM/package.json:

{
  "name": "leaning-npm",
  "version": "1.0.0",
  "description": "I am learning about npm.",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "Shoaib Ahamed",
  "license": "ISC"
}

Is this OK? (yes) y
```

Finally package.json file gets created.



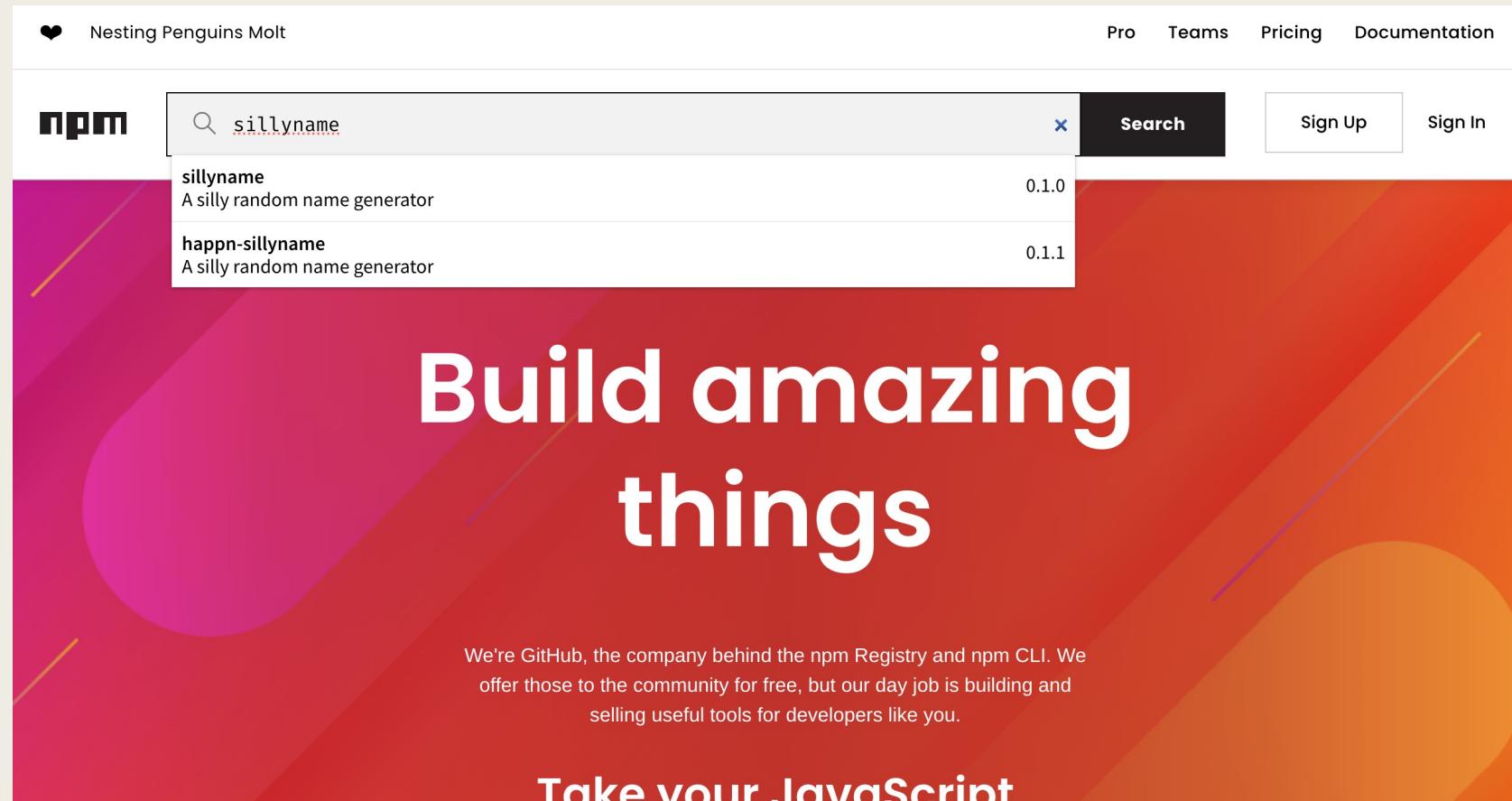
The screenshot shows the Visual Studio Code interface with the title bar "package.json — 2.3 NPM". The left sidebar displays the "OPEN EDITORS" section, which includes "index.js", "package.json" (which is currently selected and highlighted with a blue border), and "solution.js". The main editor area shows the contents of the "package.json" file:

```
1 {  
2   "name": "leaning-npm",  
3   "version": "1.0.0",  
4   "description": "I am learning about npm.",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\"",  
8   },  
9   "author": "Shoaib Ahamed",  
10  "license": "ISC"  
11}  
12
```

Now that we have configured npm, next job is to install npm packages.

```
npm install <name of the package>
```

we can find the packages here – npmjs.com



To install this package type in your terminal – **npm i sillyname**

The screenshot shows the npmjs.com package page for 'sillyname'. At the top, there's a search bar with 'Search packages' and a 'Search' button. Below the search bar, the package name 'sillyname' is displayed in bold, with its version '0.1.0' and status 'Public'. It was published 8 years ago. The 'Code' tab is selected, showing a 'Beta' badge. Other tabs include 'Readme' (highlighted), '0 Dependencies', '97 Dependents', and '4 Versions'. The main content area features a heading 'Silly name' with a link icon. Below it is the description 'A random name generator.' and a code snippet:

```
var generateName = require('sillyname');
var sillyName = generateName();
```

Underneath the code, there's a section for 'Some example results:' which is currently empty. To the right, there's an 'Install' section with a command line input field containing 'npm i sillyname'. Below that are sections for 'Repository' (link to github.com/thedeveloper/sillyname) and 'Homepage' (link to github.com/thedeveloper/sillyname#re...). At the bottom, there's a 'Weekly Downloads' chart.

Once installed you can find it as dependency

- node_modules also gets downloaded.

The screenshot shows the VS Code interface with the following details:

- EXPLORER View:** Shows files and folders:
 - index.js
 - package.json (highlighted with a blue selection bar)
 - 2.3 NPM
 - node_modules
 - index.js
 - package-lock.json
 - package.json (highlighted with a blue selection bar)
 - solution.js
- Code Editor:** Displays the package.json file content:

```
8  },
9 "author": "Shoaib Ahamed",
10 "license": "ISC",
11 "dependencies": {
12   "sillyname": "^0.1.0"
13 }
14 }
15 }
```
- TERMINAL View:** Shows the command output of an npm install:

```
e/2.3 NPM/package.json:
shoaibahamed@ZH0BM-G13FPQ05N 2.3 NPM % npm i sillyname

added 1 package, and audited 2 packages in 1s

found 0 vulnerabilities
shoaibahamed@ZH0BM-G13FPQ05N 2.3 NPM %
```

Steps to use this package

1. Require this package in index.js (as per the documentation). Then console.

```
var generateName = require('sillyname');
var sillyName = generateName();
console.log(`my name is ${sillyName}.`)
```

Output :

JS index.js X

JS index.js > ...

```
1 var generateName = require('sillyname');
2 var sillyName = generateName();
3 console.log(`my name is ${sillyName}.`)
4 // console.log("my name is " + sillyName + ".");
```

PROBLEMS

TERMINAL

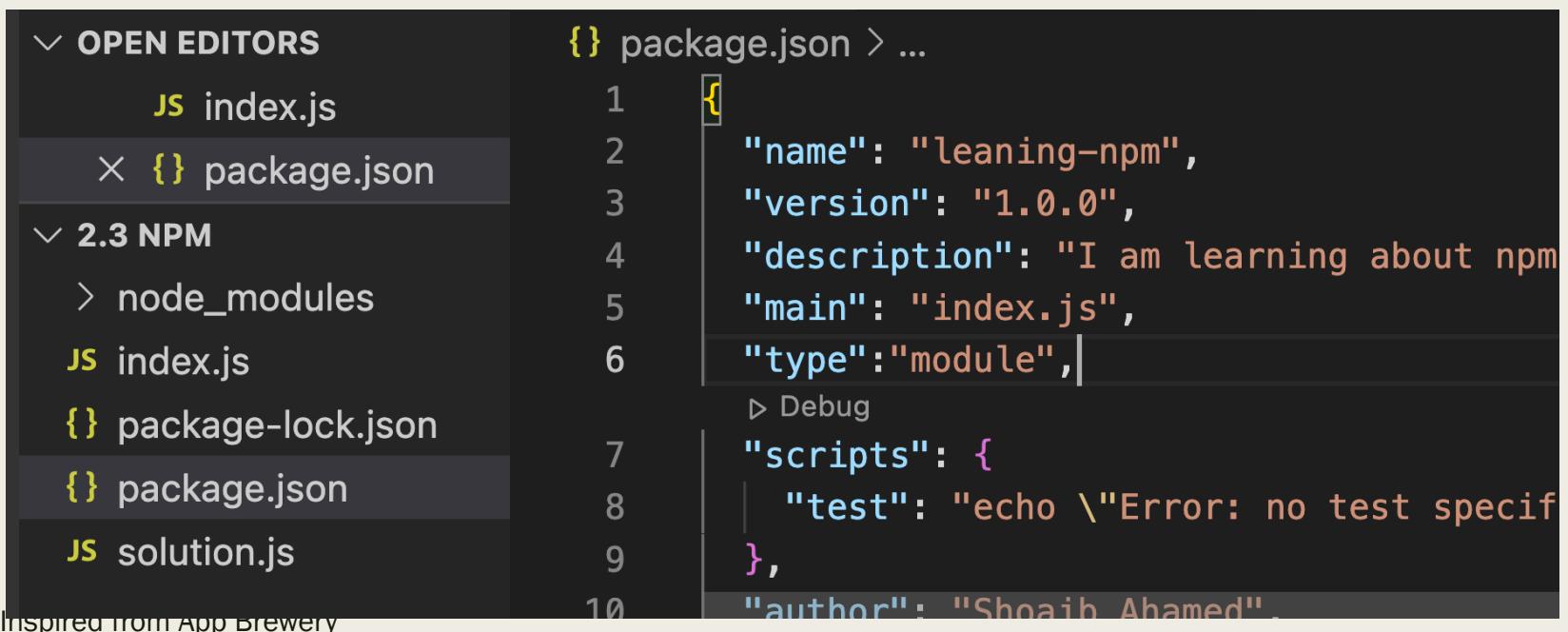
...

zsh + ^ X

```
shoabahamed@ZHOBM-G13FPQ05N 2.3 NPM % node index.js
my name is Starzebra Robin.
shoabahamed@ZHOBM-G13FPQ05N 2.3 NPM %
```

CJS vs ESM

- CJS uses require keyword.
- ESM (ECMA script modules) uses import keyword.
- In order to use ESM we have to configure package.json file. Add `"type": "module"`,

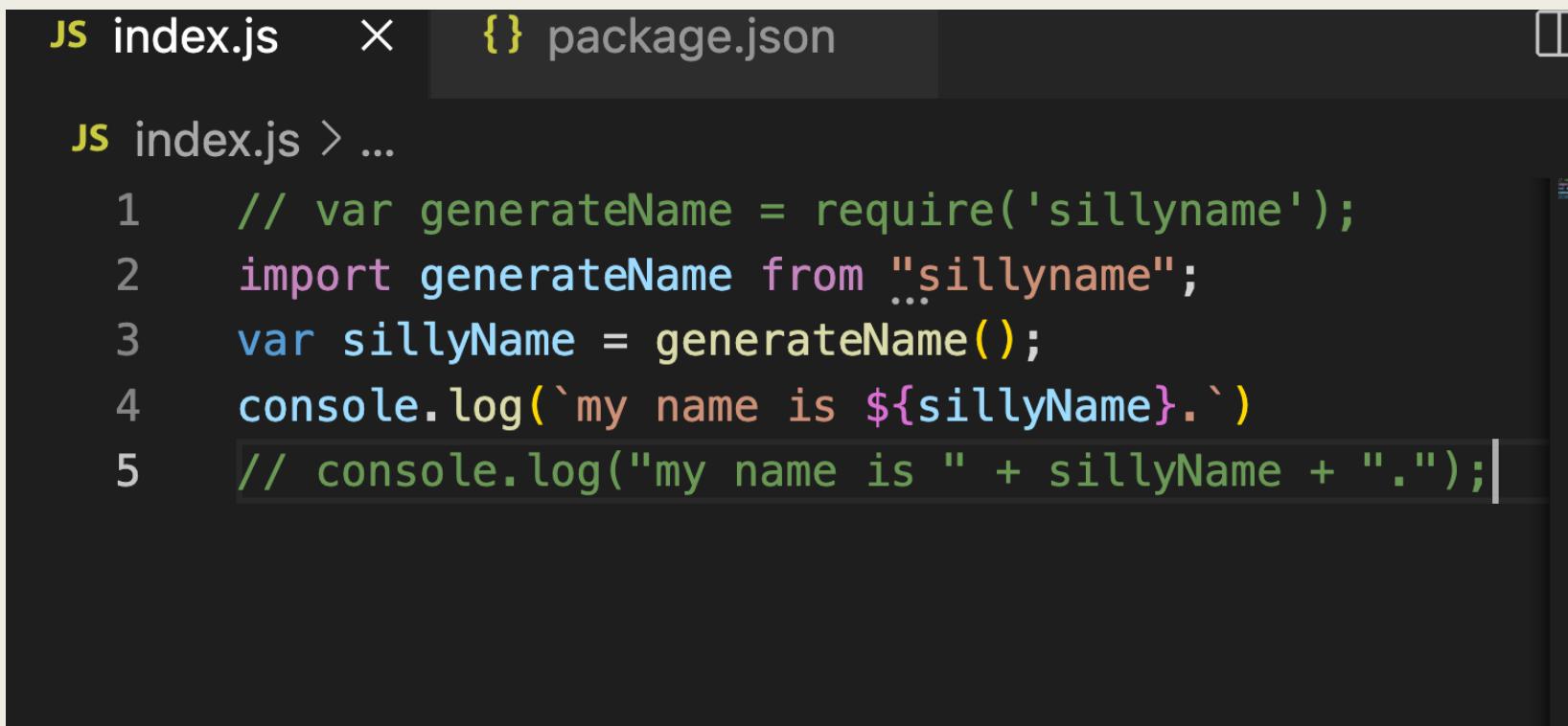


The image shows a code editor interface with two panes. The left pane, titled 'OPEN EDITORS', lists several files: index.js, package.json, node_modules, index.js, package-lock.json, package.json, and solution.js. The 'package.json' file in the left pane is currently selected and highlighted. The right pane displays the contents of the selected package.json file, which is being edited. The code is as follows:

```
{}
{
  "name": "leaning-npm",
  "version": "1.0.0",
  "description": "I am learning about npm",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\""
  },
  "author": "Shoaib Ahamed"
}
```

In the bottom left corner of the editor, there is a small footer text: 'Inspired from App Brewery'.

Import method from "name of module"



The image shows a screenshot of a code editor with two tabs: 'index.js' and 'package.json'. The 'index.js' tab is active, displaying the following code:

```
JS index.js > ...
1 // var generateName = require('sillyname');
2 import generateName from "...";
3 var sillyName = generateName();
4 console.log(`my name is ${sillyName}.`)
5 // console.log("my name is " + sillyName + ".")
```

Challenge

- Using this superheroes package - <https://www.npmjs.com/package/superheroes>
- Can you console log something like – I am (superhero name)!

Solution

```
~/Desktop/full-stack-learning/full-stack-  
classes/columbia-university /week-nine/2.3  
NPM/index.js  
1 // var generateName = require('sillyname');  
2 import generateName from "sillyname";  
3 var sillyName = generateName();  
4 console.log(`my name is ${sillyName}.`)  
5 // console.log("my name is " + sillyName + ".");  
6  
7 import superheroes from "superheroes";  
8 var superheroName = superheroes.random();  
9  
10 console.log(`my superheroname is ${superheroName}.`)
```

PROBLEMS

TERMINAL

...

❯ zsh + ⌂ ⚡

Node.js v17.0.1

shoaibahamed@ZH0BM-G13FPQ05N 2.3 NPM % node index.js

my name is Marblesword Myth.

my superheroname is Mysterio.

shoaibahamed@ZH0BM-G13FPQ05N 2.3 NPM % node index.js

my name is Gravecrest Python.

my superheroname is Dorothy.

shoaibahamed@ZH0BM-G13FPQ05N 2.3 NPM %

At your own -

- Research and learn more about usages of Node Package Manager /npm.

Homework

- `/*`
- 1. Use the inquirer npm package to get user input.
- 2. Use the qr-image npm package to turn the user entered URL into a QR code image.
- 3. Create a txt file to save the user input using the native fs node module.
- `*/`