# Contents

# Nodejs

- It is javascript run time environment
- It is used for server side programming
- Nodejs is not a language, or library or framework
- With the help of nodejs we can run javascript even outside the environment of browsers

# Nodejs Installation

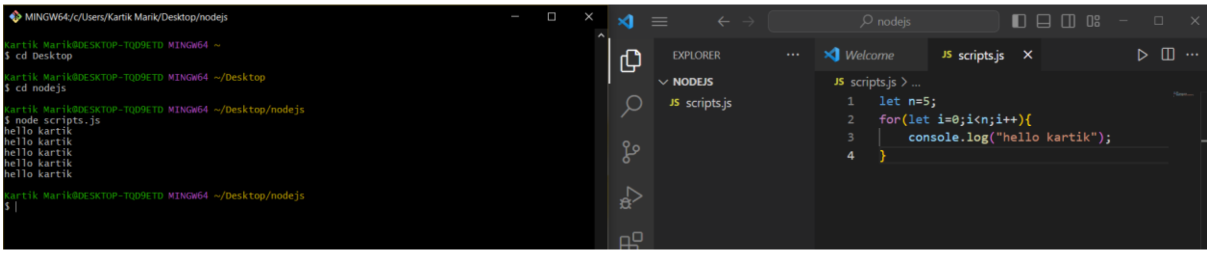
- https://nodejs.org/en
- Open gitbash and type for check installation or not
  - Node ------>to run node via terminal---->to move out press ctrl+d or ctrl+c(2 times)
  - Node -v ------->to check node version
  - Npm –version ---------->to check npm version

# Nodejs REPL(read evaluation print loop)

- Open gitbash and type
  - Node
  - Global -------------->its basically a object
  - .help

# Node files

- Syntax to run any js file ---->node filename.js
- create a folder named nodejs and make a file scripts.js and perform the below activity

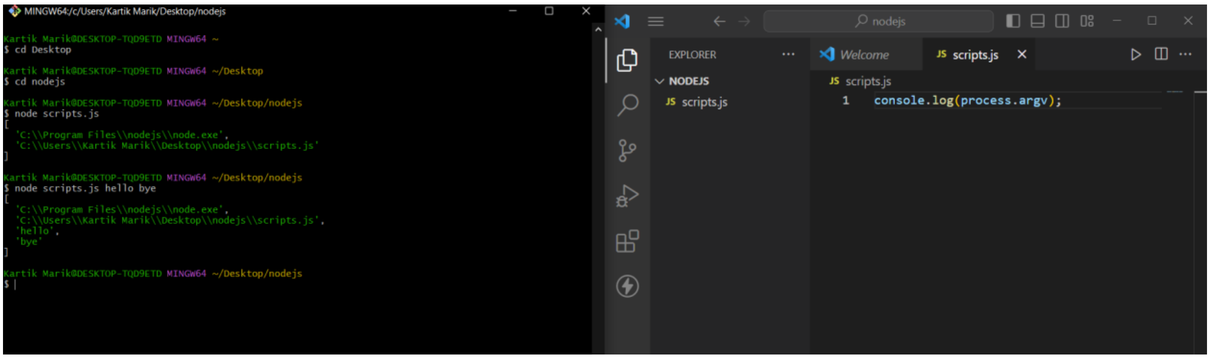


```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~
$ cd Desktop

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop
$ cd nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
hello kartik
hello kartik
hello kartik
hello kartik
hello kartik

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ |
```

```
let n=5;
for(let i=0;i<n;i++){
    console.log("hello kartik");
}
```

# Process in node

- **Process----->this object provides information about and control over the current node.js process**
- **Process.argv-----> return an array containing the command line arguments passed when node.js process was launched.**
- **Examples are below so do these steps**
  - **Open terminal and type below statements**
  - **Node**
  - **Process**
  - **Process.version**
  - **Process.release**
  - **Process.cwd**
  - **Process.argv**



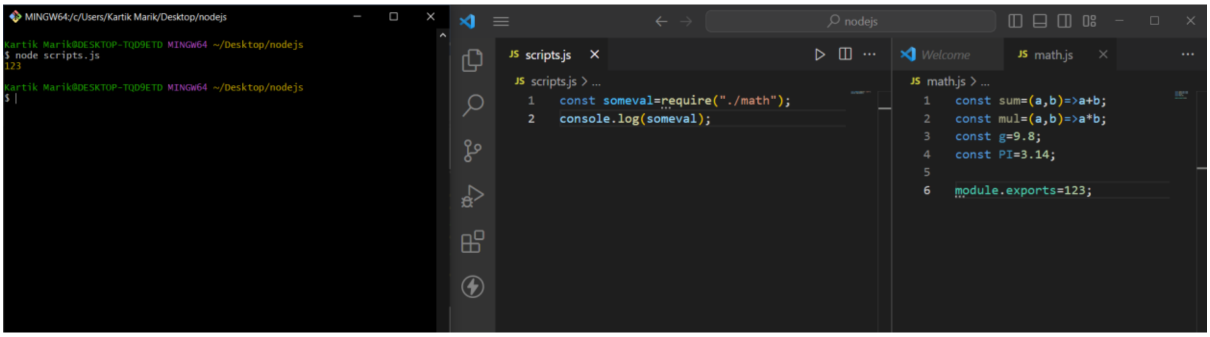
```
MINGW64:/c/Users/Kartik Marik/Desktop/nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~
$ cd Desktop

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop
$ cd nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\Kartik Marik\\Desktop\\nodejs\\scripts.js'
]

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js hello bye
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\Kartik Marik\\Desktop\\nodejs\\scripts.js',
  'hello',
  'bye'
]

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ |
```

```js
console.log(process.argv);
```

## Export in files

- Make two files name scripts.js and math.js  inside folder nodejs to perform  below exmples

## Method-1



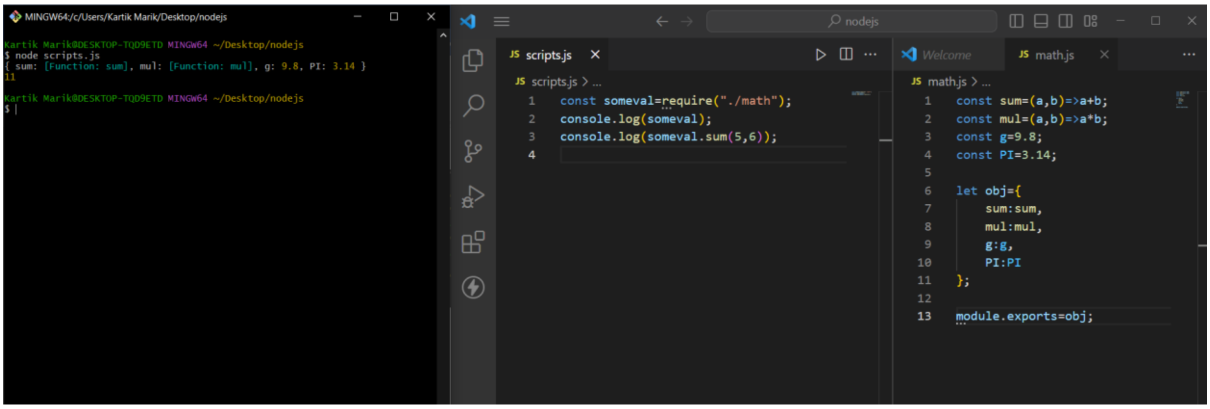
```
MINGW64:/c/Users/Kartik Marik/Desktop/nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
123

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ |
```

scripts.js

```js
const someval=require("./math");
console.log(someval);
```

math.js

```js
const sum=(a,b)=>a+b;
const mul=(a,b)=>a*b;
const g=9.8;
const PI=3.14;

module.exports=123;
```



```
MINGW64:/c/Users/Kartik Marik/Desktop/nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
{ sum: [Function: sum], mul: [Function: mul], g: 9.8, PI: 3.14 }
11

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ |
```

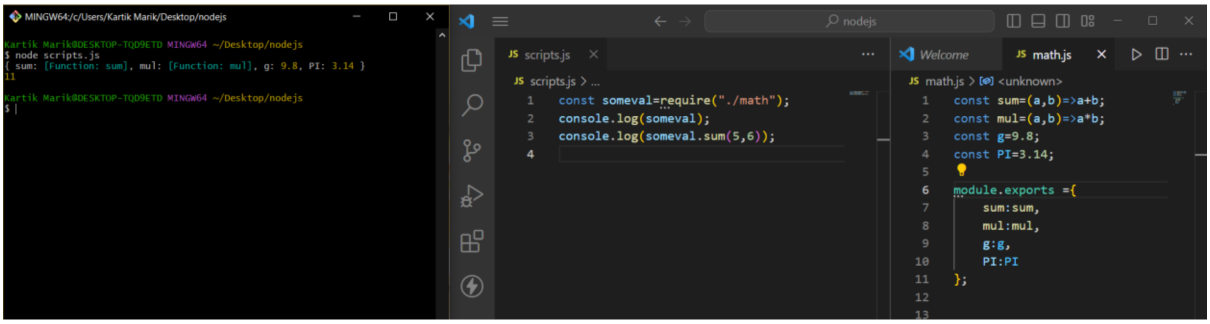
scripts.js

```js
const someval=require("./math");
console.log(someval);
console.log(someval.sum(5,6));

```

math.js

```js
const sum=(a,b)=>a+b;
const mul=(a,b)=>a*b;
const g=9.8;
const PI=3.14;

let obj={
    sum:sum,
    mul:mul,
    g:g,
    PI:PI
};

module.exports=obj;
```

## Method-2



```
MINGW64:/c/Users/Kartik Marik/Desktop/nodejs

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
{ sum: [Function: sum], mul: [Function: mul], g: 9.8, PI: 3.14 }
11

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ |
```

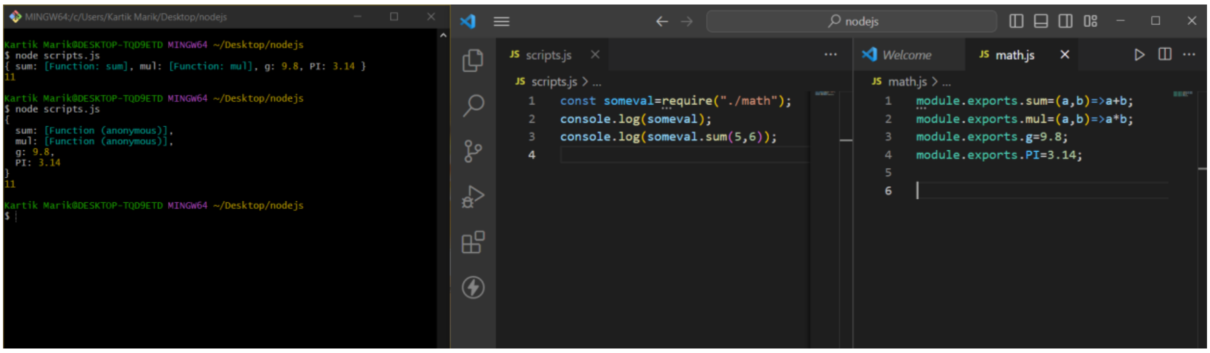
scripts.js

```js
const someval=require("./math");
console.log(someval);
console.log(someval.sum(5,6));

```

math.js

```js
const sum=(a,b)=>a+b;
const mul=(a,b)=>a*b;
const g=9.8;
const PI=3.14;

module.exports ={
    sum:sum,
    mul:mul,
    g:g,
    PI:PI
};

```

## Method-3



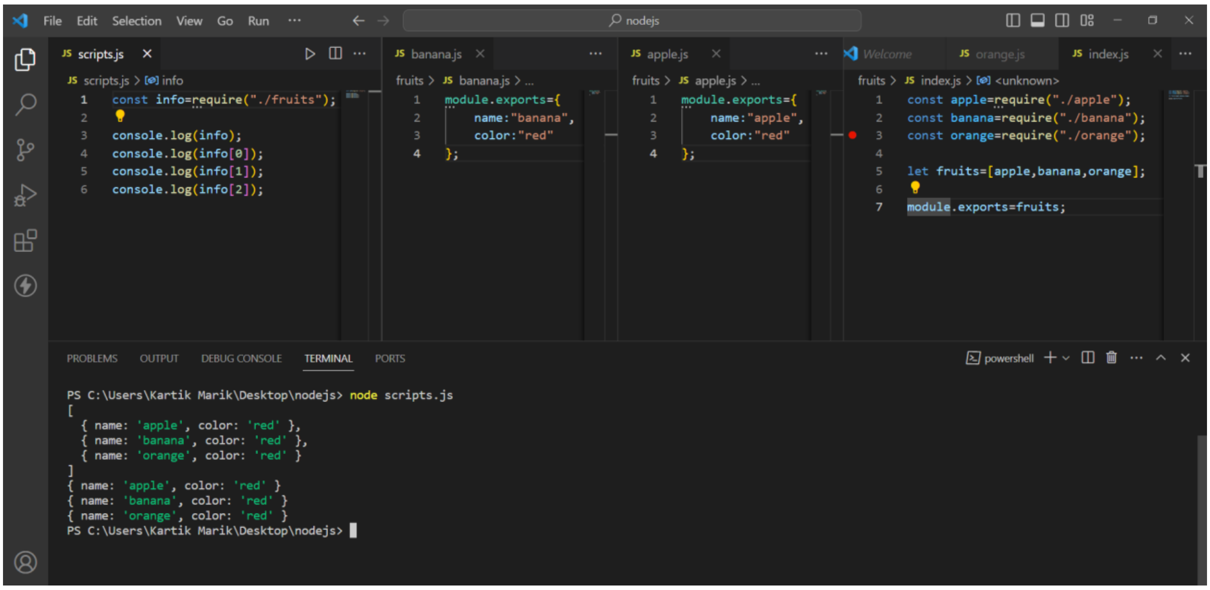
```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
{ sum: [Function: sum], mul: [Function: mul], g: 9.8, PI: 3.14 }
11

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ node scripts.js
{
  sum: [Function (anonymous)],
  mul: [Function (anonymous)],
  g: 9.8,
  PI: 3.14
}
11

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$
```

```
const someval=require("./math");
console.log(someval);
console.log(someval.sum(5,6));
```

```
module.exports.sum=(a,b)=>a+b;
module.exports.mul=(a,b)=>a*b;
module.exports.g=9.8;
module.exports.PI=3.14;
```

# Export in directories

- Make another folder  named fruits inside nodejs folder
- Make three files inside fruits folder named
  - Apple.js
  - Banana.js
  - Orange.js
  - Note  write similar thing also inside orange.js file and run it.



```
const info=require("./fruits");

console.log(info);
console.log(info[0]);
console.log(info[1]);
console.log(info[2]);
```

```
module.exports={
    name:"banana",
    color:"red"
};
```

```
module.exports={
    name:"apple",
    color:"red"
};
```

```
const apple=require("./apple");
const banana=require("./banana");
const orange=require("./orange");

let fruits=[apple,banana,orange];

module.exports=fruits;
```

```
PS C:\Users\Kartik Marik\Desktop\nodejs> node scripts.js
[
  { name: 'apple', color: 'red' },
  { name: 'banana', color: 'red' },
  { name: 'orange', color: 'red' }
]
{ name: 'apple', color: 'red' }
{ name: 'banana', color: 'red' }
{ name: 'orange', color: 'red' }
PS C:\Users\Kartik Marik\Desktop\nodejs>
```

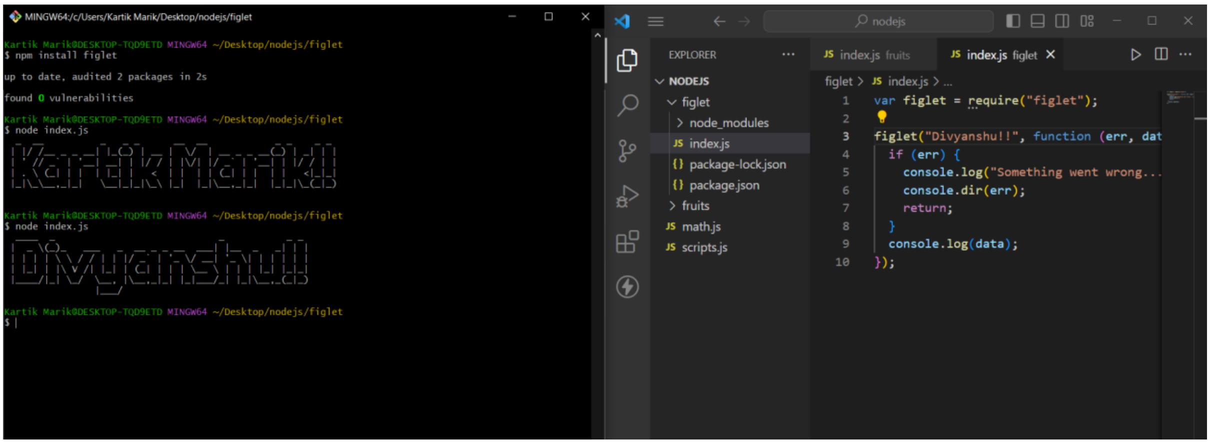
# NPM-(node package manager)

- Npm is the standard pacakage manager for node.js
- Library of pacakages
- Command linetool
- Npm is preinstalled comes with node so to verify write below code in gitbash
  - Npm
- Synatax to install any pacakage
  - Npm install <-pacakage name->
- https://www.npmjs.com/    -------------->search any pacakages
  - Exaples of some pacakages are:-

- Reactjs
- Expressjs
- Figlet
- Give-me-a-joke

## Examples-1 to demonstrate of installing pacakage and using it

- Create a folder named figlet inside nodejs
- Go that that folder location by terminal
- Npm install figlet
- Create index.js file inside figlet folder and run below code
- https://www.npmjs.com/package/figlet



```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/figlet
$ npm install figlet

up to date, audited 2 packages in 2s

found 0 vulnerabilities

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/figlet
$ node index.js

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/figlet
$ node index.js

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/figlet
$ |
```

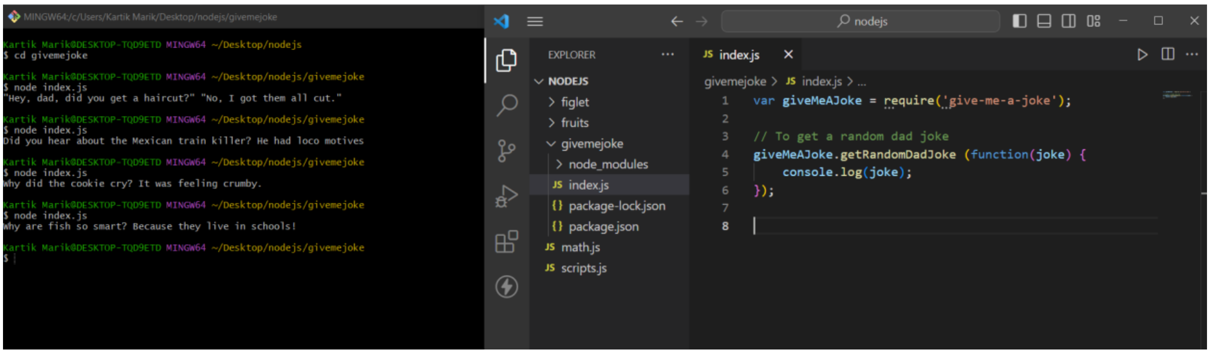
```
var figlet = require("figlet");

figlet("Divyanshu!!", function (err, dat
  if (err) {
    console.log("Something went wrong...
    console.dir(err);
    return;
  }
  console.log(data);
});
```

Note---> three files when we install any pacakage

- Node_modules----->the node_modules folder contains every installed dependency to your project
- Pacakage-lock.json-------->it records the exact version of every installed dependency, including its sub dependencies and their versions
- Pacakage.json---------------->this file contains descriptive and functional metadata about project such as a name, version, and dependencies.

## Examples-2 to demonstrate of installing pacakage and using it

- https://www.npmjs.com/package/give-me-a-joke
- Create a folder givemejoke inside nodejs folder
- Go to location and install this pacakage
- Create index.js file and have fun

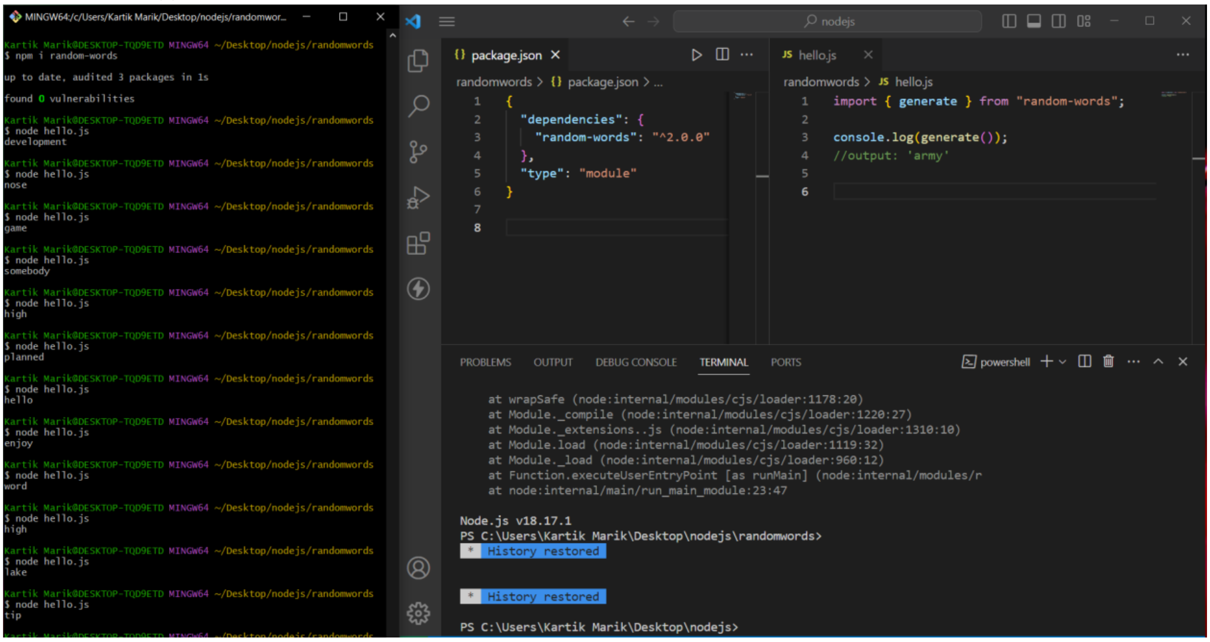


```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs
$ cd givemejoke

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/givemejoke
$ node index.js
"Hey, dad, did you get a haircut?" "No, I got them all cut."

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/givemejoke
$ node index.js
Did you hear about the Mexican train killer? He had loco motives

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/givemejoke
$ node index.js
Why did the cookie cry? It was feeling crumby.

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/givemejoke
$ node index.js
Why are fish so smart? Because they live in schools!

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/givemejoke
$
```

```
var giveMeAJoke = require('give-me-a-joke');

// To get a random dad joke
giveMeAJoke.getRandomDadJoke (function(joke) {
    console.log(joke);
});

```

## Examples-3 to demonstrate  of installing pacakage and using it

- Create another folder named randomwords inside nodejs folder
- Go to location and install this pacakage
- Create index,js and have fun
- https://www.npmjs.com/package/random-words
- Note it used import keyword instead of require so do change for correcting error
- See where “type”:”module” is written and never do mistakes…….



```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ npm i random-words

up to date, audited 3 packages in 1s

found 0 vulnerabilities

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
development

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
nose

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
game

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
somebody

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
high

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
planned

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
hello

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
enjoy

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
word

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
high

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
lake

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
$ node hello.js
tip

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/nodejs/randomwords
```

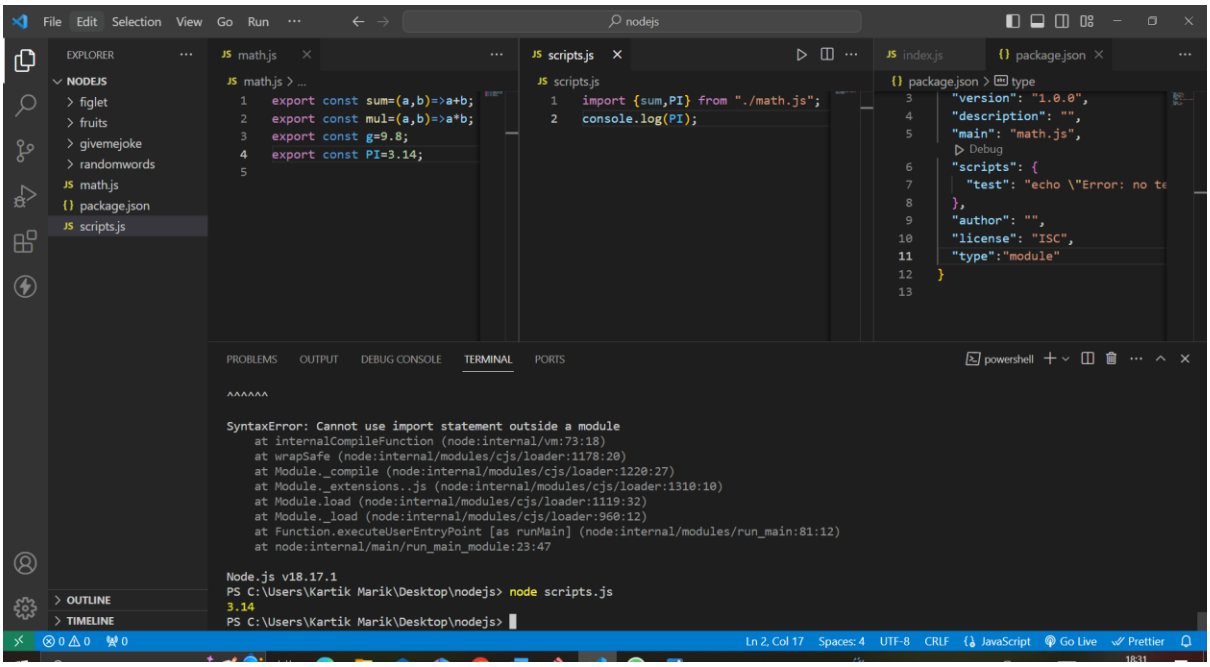
```
{
  "dependencies": {
    "random-words": "^2.0.0"
  },
  "type": "module"
}

```

```
import { generate } from "random-words";

console.log(generate());
//output: 'army'

```

```
    at wrapSafe (node:internal/modules/cjs/loader:1178:20)
    at Module._compile (node:internal/modules/cjs/loader:1220:27)
    at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
    at Module.load (node:internal/modules/cjs/loader:1119:32)
    at Module._load (node:internal/modules/cjs/loader:960:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/r
    at node:internal/main/run_main_module:23:47

Node.js v18.17.1
PS C:\Users\Kartik Marik\Desktop\nodejs\randomwords>
 *  History restored


 *  History restored

PS C:\Users\Kartik Marik\Desktop\nodejs>
```

## require vs import

- Require keyword---> previously used
- Import  keyword is better than require keyword to use
- Example ---> how to use import by writing some changes in package.json



```
export const sum=(a,b)=>a+b;
export const mul=(a,b)=>a*b;
export const g=9.8;
export const PI=3.14;
```

```
import {sum,PI} from "./math.js";
console.log(PI);
```

```
  "version": "1.0.0",
  "description": "",
  "main": "math.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no te
  },
  "author": "",
  "license": "ISC",
  "type":"module"
}
```

```
^^^^^^

SyntaxError: Cannot use import statement outside a module
    at internalCompileFunction (node:internal/vm:73:18)
    at wrapSafe (node:internal/modules/cjs/loader:1178:20)
    at Module._compile (node:internal/modules/cjs/loader:1220:27)
    at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
    at Module.load (node:internal/modules/cjs/loader:1119:32)
    at Module._load (node:internal/modules/cjs/loader:960:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:23:47

Node.js v18.17.1
PS C:\Users\Kartik Marik\Desktop\nodejs> node scripts.js
3.14
PS C:\Users\Kartik Marik\Desktop\nodejs>
```