NAME: NGUYEN TAN PHAT

NODEJS LAP 1

Table of Contents

[Task 1, Check NodeJS version. 2](#_Toc156816369)

[Task 2, Example of runtime environment 2](#_Toc156816370)

[1, Definition of runtime environment 2](#_Toc156816371)

[2, NodeJS example of runtime environment 2](#_Toc156816372)

[Task 3, Provide a screenshot to show that Nodejs is a runtime supported language? 2](#_Toc156816373)

[Task 4, Run the below code 3](#_Toc156816374)

[1, Capture screen and report 3](#_Toc156816375)

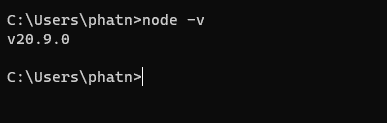
[2, Write about the role of console.log 3](#_Toc156816376)

[Task 5, Code exercises 3](#_Toc156816377)

[1, Source code 3](#_Toc156816378)

[2, Screenshot of the result 5](#_Toc156816379)

# Task 1, Check NodeJS version.



# Task 2, Example of runtime environment

## 1, Definition of runtime environment

A runtime environment is a set of software and hardware components that enable a program to run on a specific platform.

## 2, NodeJS example of runtime environment

// Import the fs module

const fs = require('fs');

// Create a new file called hello.txt

fs.writeFile('hello.txt', 'Hello, world!', (err) => {

// Handle any errors

if (err) {

console.error(err);

} else {

// Log a success message

console.log('The file was created successfully!');

}

});

A runtime environment with Node.js using the fs module is a software stack that uses to create and run JavaScript programs that can interact with the file system on your computer. The fs module is a built-in module that provides functions for reading, writing, deleting, and modifying files and directories.

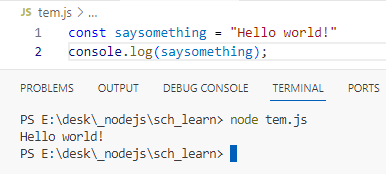
# Task 3, Provide a screenshot to show that Nodejs is a runtime supported language?

# Task 4, Run the below code

const saysomething = "Hello world!"

console.log(saysomething);

## 1, Capture screen and report



## 2, Write about the role of console.log

The role of console.log in Node.js is similar to the role of console.log in the browser. It is a function that prints messages or values to the standard output stream, which is usually the terminal or the console window where run Node.js program.

# Task 5, Code exercises

## 1, Source code

1. const fs = require('fs');

2. const fsp = require('fs/promises');

3.

4. // 5.1

5. try{

6.     const index = fs.readFileSync('txt/start.txt', 'utf-8');

7.     console.log("\*\*Below is the content of start.txt\*\*", '\n', index);

8. } catch (err){

9.     console.error(err);

10. }

11. // 5.1 \*\*

12.

13. // 5.2

14. async function readandwrite(){

15.     try{

16.         const index2 = await fsp.readFile('txt/read-this.txt', 'utf-8');

17.         console.log("\*\*Below is the content of read-this.txt\*\*", '\n', index2);

18.         console.log("\*\*Writing read-this.txt to start.txt\*\*");

19.         await fsp.writeFile('txt/start.txt', index2, 'utf-8');

20.         console.log("File written successfully");

21.     } catch (err){

22.         console.error(err);

23.     }

24. }

25. // 5.2 \*\*

26.

27. // 5.3

28. async function readandwrite2(){

29.     try{

30.         console.log("\*\*Reading append.txt\*\*");

31.         const index3 = fs.readFileSync('txt/append.txt', 'utf-8');

32.         console.log("\*\*Writing append.txt to start.txt\*\*");

33.         fs.appendFileSync('txt/start.txt', '\n' + index3, 'utf-8');

34.     } catch(err){

35.         console.log(err);

36.     }

37. }

38. // 5.3 \*\*

39.

40. // 5.4 - 5.5

41. async function rename(){

42.     try{

43.         console.log('Renaming start.txt to final2.txt');

44.         fs.renameSync('txt/start.txt', 'txt/final2.txt');

45.         const index4 = await fs.readFileSync('txt/final2.txt', 'utf-8');

46.     console.log("\*\*Below is the content of final2.txt\*\*", '\n', index4);

47.     }catch(err){

48.         console.log(err);

49.     }

50. }

51. // 5.4 - 5.5 \*\*

52.

53. async function main(){

54.     await readandwrite();

55.     await readandwrite2();

56.     await rename();

57. }

58. main();

59.

60.

## 2, Screenshot of the result

