

**code :** [wap/NodeFolder at main · chinawutao1979/wap · GitHub](https://github.com/chinawutao1979/wap/tree/main/wap/NodeFolder)

## Exercise N1

**Q:**

- Download Node.js from the official Node.js website and install it: <https://nodejs.org>.
- Create a folder for your Node.js files (NodeFolder).
- Save the Hello World program in a file using VS Code:

```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200,  
    {'Content-Type': 'text/html'});  
  res.end('Hello World!');  
}).listen(8080);
```

Open a Command Window

- Run command:

```
C:\NodeFolder> npm init
```

- Run the Hello World program:

```
C:\NodeFolder> node HelloWorld.js
```

- In your Browser, enter the URL:

<http://localhost:8080>

**A:**

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project named 'NODEFOLDER' containing a file 'package.json'. The main editor area is split into two panes. The left pane shows the 'package.json' file with its contents: 

```
{
  "name": "nodefolder",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "tao wu",
  "license": "MIT"
}
```

. The right pane shows the terminal output of the 'npm init' command, which prompts for package details and confirms the creation of the 'package.json' file. The terminal text includes: 'Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.', 'Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows', 'PS C:\Users\china\Documents\GitHub\NodeFolder> 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: (nodefolder)', 'version: (1.0.0)', 'description:', 'entry point: (index.js)', 'test command:', 'git repository:', 'keywords:', 'author: tao wu', 'license: (ISC) MIT', 'About to write to C:\Users\china\Documents\GitHub\NodeFolder\package.json:', the JSON object, 'Is this OK? (yes) y', and 'PS C:\Users\china\Documents\GitHub\NodeFolder>'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> 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: (nodefolder)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: tao wu
license: (ISC) MIT
About to write to C:\Users\china\Documents\GitHub\NodeFolder\package.json:

{
  "name": "nodefolder",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "tao wu",
  "license": "MIT"
}

Is this OK? (yes) y
PS C:\Users\china\Documents\GitHub\NodeFolder>
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project named 'NODEFOLDER' containing a file 'package.json'. The main editor area is split into two panes. The left pane shows the 'package.json' file with its contents: 

```
{
  "name": "nodefolder",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "tao wu",
  "license": "MIT"
}
```

. The right pane shows the terminal output of the 'npm init' command, which prompts for package details and confirms the creation of the 'package.json' file. The terminal text includes: 'Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.', 'Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows', 'PS C:\Users\china\Documents\GitHub\NodeFolder> 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: (nodefolder)', 'version: (1.0.0)', 'description:', 'entry point: (index.js)', 'test command:', 'git repository:', 'keywords:', 'author: tao wu', 'license: (ISC) MIT', 'About to write to C:\Users\china\Documents\GitHub\NodeFolder\package.json:', the JSON object, 'Is this OK? (yes) y', and 'PS C:\Users\china\Documents\GitHub\NodeFolder>'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> 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: (nodefolder)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: tao wu
license: (ISC) MIT
About to write to C:\Users\china\Documents\GitHub\NodeFolder\package.json:

{
  "name": "nodefolder",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "tao wu",
  "license": "MIT"
}

Is this OK? (yes) y
PS C:\Users\china\Documents\GitHub\NodeFolder>
```

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays a file tree for a project named 'NODEFOLDER'. Inside, there's a 'week3Monday' folder containing 'exerciseN1', which has subfolders 'play' and 'helloModule.js', and files 'app.js', 'app2.js', 'Hello\_World.js', and 'package.json'. The main editor shows the content of 'Hello\_World.js', which is a simple HTTP server using the 'http' module. The terminal at the bottom shows a PowerShell session where the user navigates to the project directory and runs 'node Hello\_World.js'.

```
File Edit Selection View Go Run Terminal Help
Hello_World.js - NodeFolder - Visual Studio Code

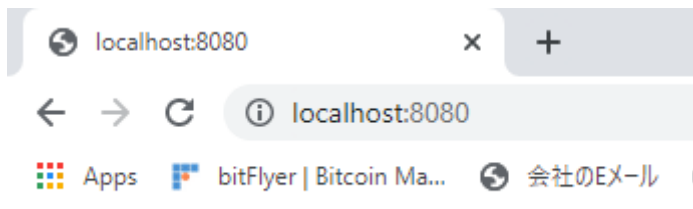
EXPLORER
NODEFOLDER
  week3Monday
    exerciseN1
      play
      JS app.js
      JS app2.js
      JS Hello_World.js
      JS helloModule.js
    exerciseN2
    exerciseN3
    exerciseN4
    package.json

JS Hello_World.js
week3Monday > exerciseN1 > JS Hello_World.js > ...
1 var http = require('http');
2
3 http.createServer(function (req, res) {
4   res.writeHead(200,
5     {'Content-Type': 'text/html'});
6   res.end('Hello World!');
7 }).listen(8080);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

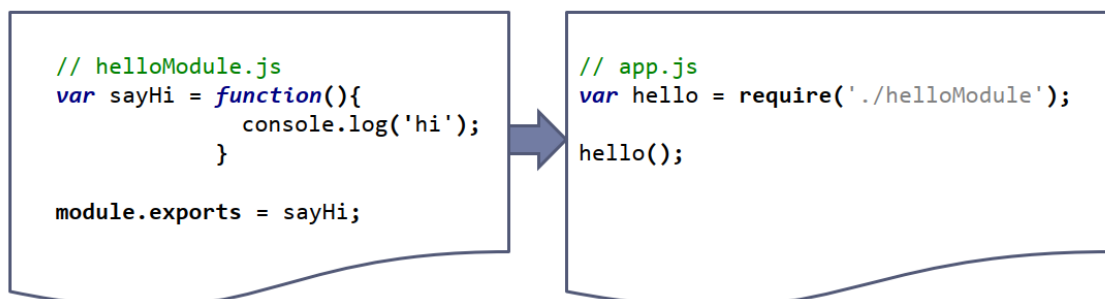
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1> node Hello_World.js
```



Q:

`module.exports`

- ▶ Think of this object (`module.exports`) as a return statement.



A:

```
JS app.js JS helloModule.js X JS helloModule.js JS app2.js
week3Monday > exerciseN1 > JS helloModule.js > ...
1 var sayHi = function () {
2   console.log("hi");
3 };
4
5 module.exports = sayHi;
6

week3Monday > exerciseN1 > JS app2.js > ...
1 var hello = require("./helloModule");
2 hello();
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1> node app2.js
hi
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1>
```

Q:

## Create your own module

play folder

```
// play/violin.js
const play = function() { console.log("First Violin is playing!"); }
module.exports = play;
```

```
// play/clarinet.js
const play = function() { console.log("Clarinet is playing!"); }
module.exports = play;
```

```
// play/index.js
const violin = require('./violin');
const clarinet = require('./clarinet');
module.exports = { 'violin': violin, 'clarinet': clarinet };
```

```
// app.js
const play = require('./play');
play.violin();
play.clarinet();
```

A:

```
JS violin.js X JS clarinet.js JS index.js
week3Monday > exerciseN1 > play > JS violin.js > ...
1 // play/violin.js
2 const play = function() { console.log("First Violin is playing!"); }
3 module.exports = play;
```

```
JS violin.js JS clarinet.js X JS index.js
week3Monday > exerciseN1 > play > JS clarinet.js > ...
1 // play/clarinet.js
2 const play = function() { console.log("Clarinet is playing!"); }
3 module.exports = play;
```

```
JS violin.js JS clarinet.js JS index.js X
week3Monday > exerciseN1 > play > JS index.js > ...
1 // play/index.js
2 const violin = require('./violin');
3 const clarinet = require('./clarinet');
4 module.exports = { 'violin': violin, 'clarinet': clarinet };
```

```
EXPLORER ... JS violin.js JS clarinet.js JS index.js JS app.js X
v NODEFOLDER
v week3Monday no
v exerciseN1
v play
JS clarinet.js
JS index.js
JS violin.js
JS app.js
JS app2.js
JS Hello_World.js
JS helloModule.js
> exerciseN2
> exerciseN3
> exerciseN4
{} package.json

week3Monday > exerciseN1 > JS app.js > ...
1 // app.js
2 const play = require('./play');
3 play.violin();
4 play.clarinet();
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1> node app.js
First Violin is playing!
Clarinet is playing!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN1> |
```

## Exercise N2

Q:

```
myDate function () {  
  return Date();  
};  
exports.myDate = myDate;
```

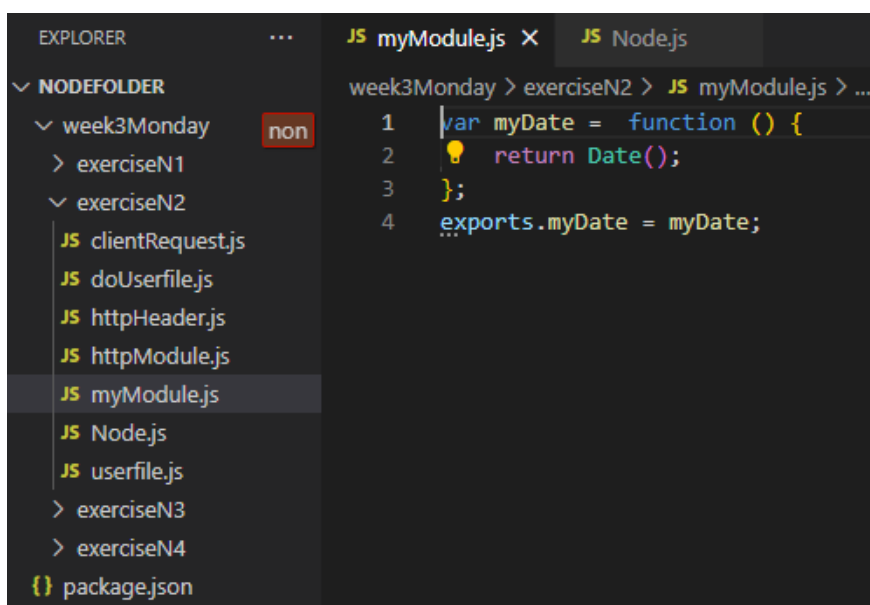
- Use the exports keyword to make properties and methods available outside the module file.

- Save this in a file called myModule.js

Use the module myModule in a Node.js file:

```
var http = require('http');  
var dt = require('./myModule');  
http.createServer(function (req, res) {  
  res.writeHead(200,  
    {'Content-Type': 'text/html'});  
  res.write("The date and time are  
    currently: " + dt.myDate());  
  res.end();  
}).listen(8080);
```

A:



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree for a project named 'NODEFOLDER'. It contains subfolders 'week3Monday', 'exerciseN1', and 'exerciseN2'. Under 'exerciseN2', there are files: 'clientRequest.js', 'doUserfile.js', 'httpHeader.js', 'httpModule.js', 'myModule.js', 'Node.js', 'userfile.js', and 'package.json'. The 'Node.js' file is selected. The main editor area shows the content of 'Node.js', which is a simple HTTP server using the 'http' module. The code is as follows:

```
1 var http = require('http');
2 var dt = require('./myModule');
3
4 http.createServer(function (req, res) {
5   res.writeHead(200,
6     {'Content-Type': 'text/html'});
7   res.write("The date and time are currently: " + dt.myDate());
8   res.end();
9 }).listen(8080);
```

The screenshot shows the VS Code Terminal window. The title bar indicates 'Windows PowerShell'. The terminal output shows the following commands and their results:

```
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2> node Node.js
```

The screenshot shows a web browser window with the address bar set to 'localhost:8080'. The page content displays the output of the Node.js server:

The date and time are currently: Mon May 09 2022 16:24:47 GMT-0500 (Central Daylight Time)

Q:

### Another Module Example

```
const getName = () => {
  return 'Jim';
};

const getLocation = () => {
  return 'Munich';
};

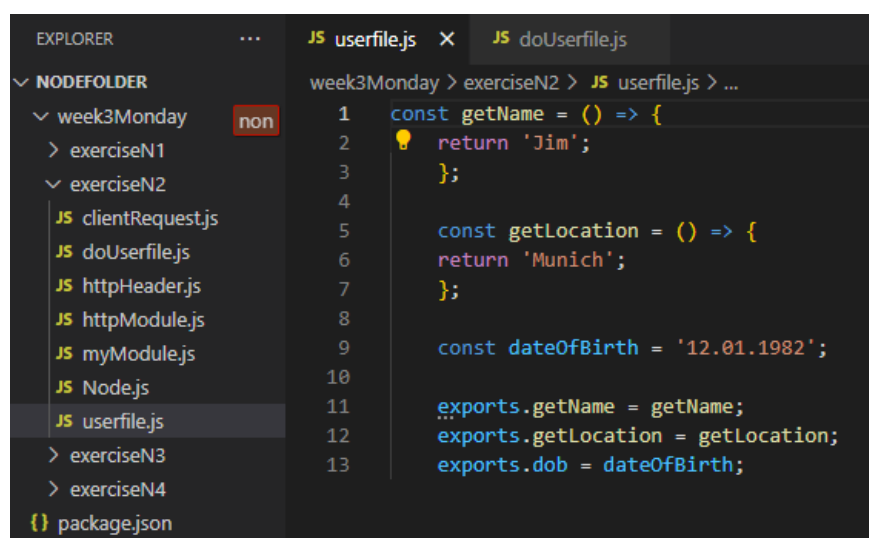
const dateOfBirth = '12.01.1982';
exports.getName = getName;
exports.getLocation = getLocation;
exports.dob = dateOfBirth;
```

## Using the New Module

```
const user = require('./userfile');  
console.log(user.getName() + ' lives  
in ' + user.getLocation() + ' and was  
born on ' + user.dob);
```

- The require operation needs the file name of the module.
- The constant user accesses the public methods.

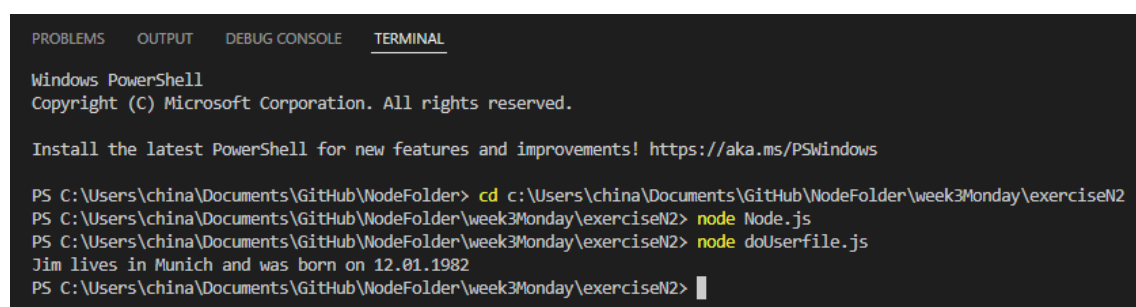
A:



```
EXPLORER  ...  JS userfile.js  JS doUserfile.js  
week3Monday > exerciseN2 > JS userfile.js > ...  
1  const getName = () => {  
2    return 'Jim';  
3  };  
4  
5  const getLocation = () => {  
6    return 'Munich';  
7  };  
8  
9  const dateOfBirth = '12.01.1982';  
10  
11  exports.getName = getName;  
12  exports.getLocation = getLocation;  
13  exports.dob = dateOfBirth;
```



```
EXPLORER  ...  JS userfile.js  JS doUserfile.js  
week3Monday > exerciseN2 > JS doUserfile.js > ...  
1  const user = require('./userfile');  
2  console.log(user.getName() + ' lives in ' + user.getLocation() + ' and was born on ' + user.dob);
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2  
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2> node Node.js  
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2> node doUserfile.js  
Jim lives in Munich and was born on 12.01.1982  
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2>
```

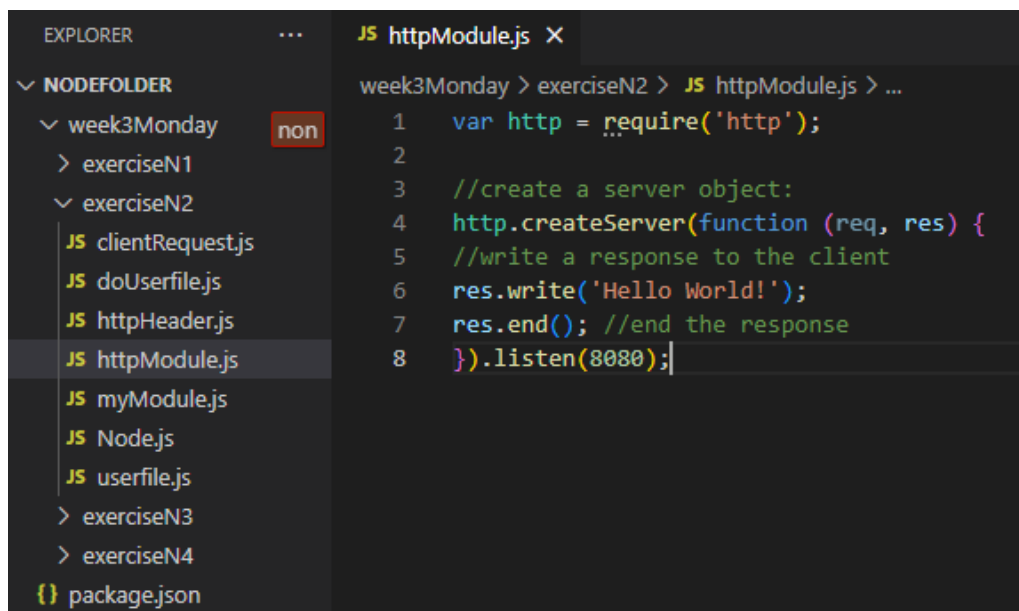


Q:

## HTTP Module

```
var http = require('http');  
//create a server object:  
http.createServer(function (req, res) {  
  //write a response to the client  
  res.write('Hello World!');  
  res.end(); //end the response  
}).listen(8080);  
//the server object listens on port 8080
```

A:

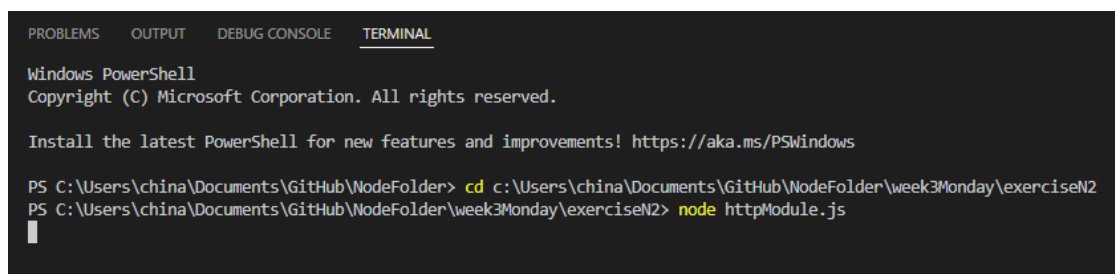


The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree with the following structure:

- NODEFOLDER
  - week3Monday
    - exerciseN1
    - exerciseN2
      - clientRequest.js
      - doUserfile.js
      - httpHeader.js
      - httpModule.js (selected)
      - myModule.js
      - Node.js
      - userfile.js
    - exerciseN3
    - exerciseN4
  - package.json

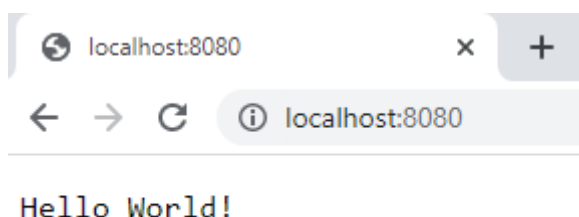
The main editor window shows the content of `httpModule.js`:

```
1 var http = require('http');  
2  
3 //create a server object:  
4 http.createServer(function (req, res) {  
5   //write a response to the client  
6   res.write('Hello World!');  
7   res.end(); //end the response  
8 }).listen(8080);
```



The screenshot shows the VS Code Terminal window with the following PowerShell commands and output:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2  
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN2> node httpModule.js
```



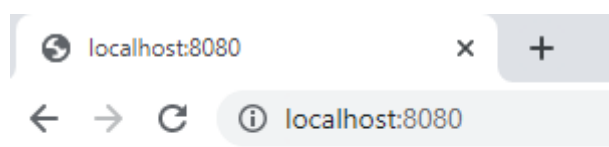
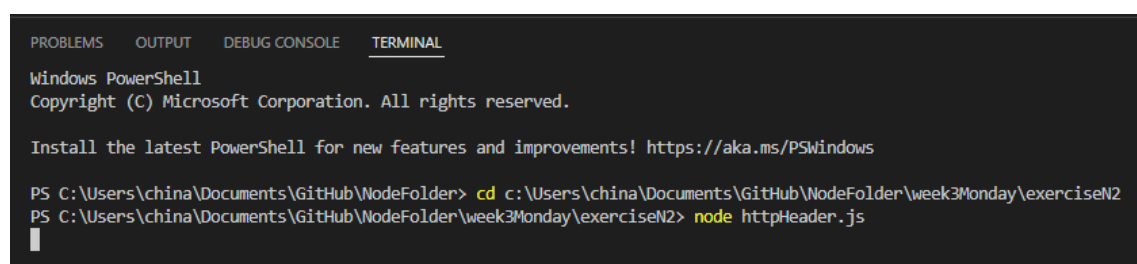
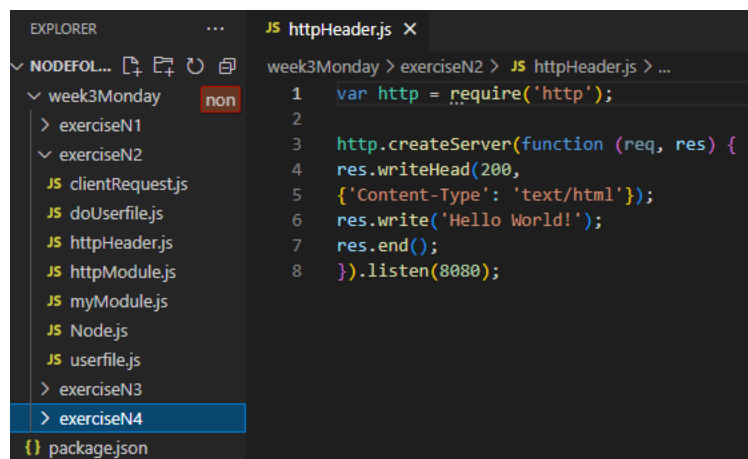
Q:

## HTTP Header

If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200,
    {'Content-Type': 'text/html'});
  res.write('Hello World!');
  res.end();
}).listen(8080);
```

A:



Hello World!

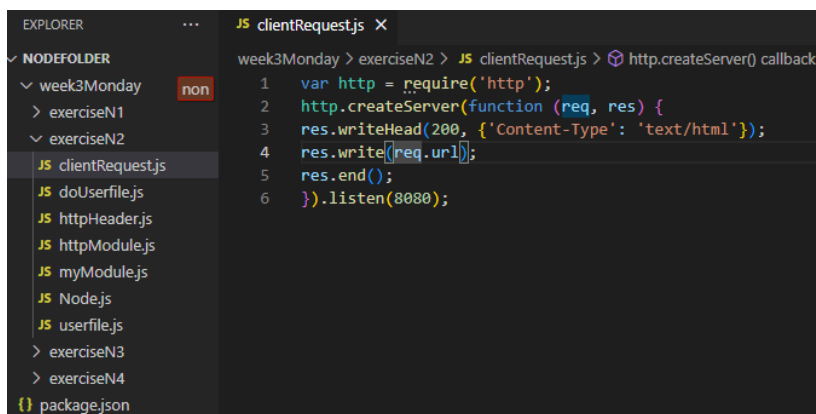
Q:

### Client Request

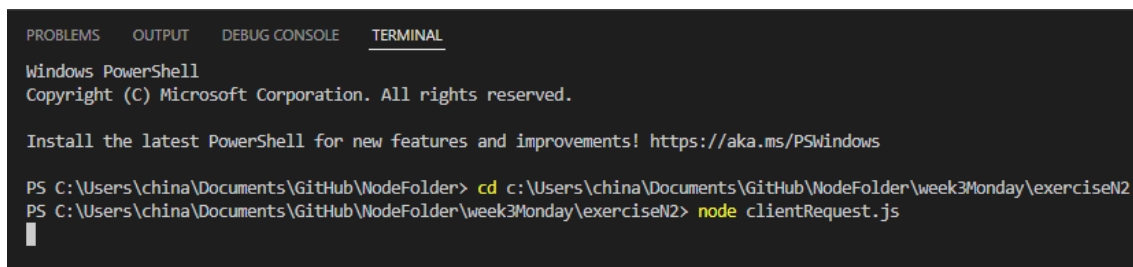
- The req argument represents the request from the client.
- The property url holds the part of the url that comes after the domain name.

```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write(req.url);  
  res.end();  
}).listen(8080);
```

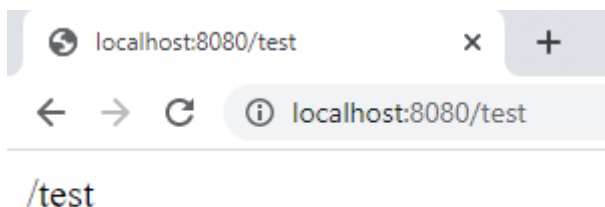
A:



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree for a project named 'NODEFOLDER'. The tree includes a 'week3Monday' directory containing 'exerciseN1', 'exerciseN2', and 'package.json'. Under 'exerciseN2', several JavaScript files are listed: 'clientRequest.js', 'doUserfile.js', 'httpHeader.js', 'httpModule.js', 'myModule.js', 'Node.js', and 'userfile.js'. The 'clientRequest.js' file is selected and its content is displayed in the main editor. The code in the editor matches the code provided in the previous block, showing the setup of an HTTP server that responds with the request URL.



The screenshot shows a Windows PowerShell terminal window. The title bar indicates it's a 'Windows PowerShell' session. The prompt shows the current directory is 'C:\Users\china\Documents\GitHub\NodeFolder'. The user has navigated to the 'week3Monday\exerciseN2' subdirectory and executed the command 'node clientRequest.js' to run the server.



## Exercise N3

Q:

Run the code shown below and then enter the following in your

Browser:

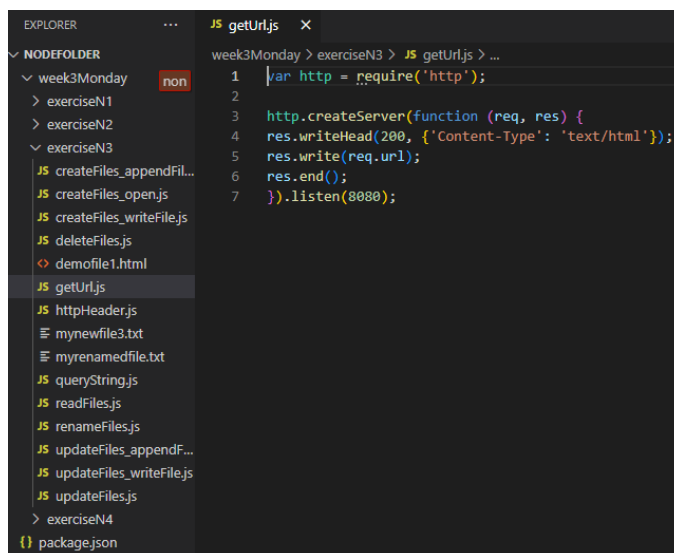
`http://localhost:8080/apples`

`http://localhost:8080/pears`

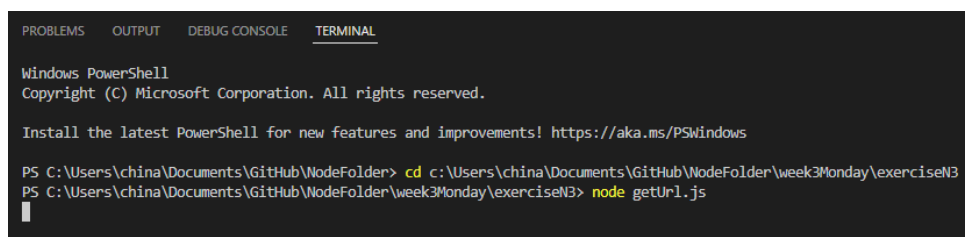
`http://localhost:8080/strawberries`

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url);
  res.end();
}).listen(8080);
```

A:



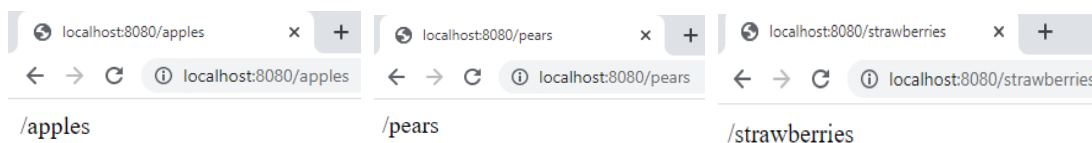
```
EXPLORER
week3Monday > exerciseN3 > JS getUrl.js > ...
1 | var http = require('http');
2
3 | http.createServer(function (req, res) {
4 |   res.writeHead(200, {'Content-Type': 'text/html'});
5 |   res.write(req.url);
6 |   res.end();
7 | }).listen(8080);
JS createFiles_appendFil...
JS createFiles_open.js
JS createFiles_writeFile.js
JS deleteFiles.js
demoFile1.html
JS getUrl.js
JS httpHeader.js
mynewfile3.txt
myrenamedfile.txt
JS queryString.js
JS readFiles.js
JS renameFiles.js
JS updateFiles_appendF...
JS updateFiles_writeFile.js
JS updateFiles.js
exerciseN4
package.json
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node getUrl.js
```



Q:

### Query String

The following code reads the values from the query string corresponding to the names year and month:

```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(8080);
```

A:

```
week3Monday > exerciseN3 > JS queryString.js > ...
1  var http = require('http');
2  var url = require('url');
3
4  http.createServer(function (req, res) {
5    res.writeHead(200, {'Content-Type': 'text/html'});
6    var q = url.parse(req.url, true).query;
7    var txt = q.year + " " + q.month;
8    res.end(txt);
9  }).listen(8080);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3  
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node queryString.js

localhost:8080/queryString.js?year=2022&month=03

2022 03

Q:

## HTTP Header

If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200,
    {'Content-Type': 'text/html'});
  res.write('Hello World!');
  res.end();
}).listen(8080);
```

A:

```
week3Monday > exerciseN3 > JS httpHeader.js > ...
```

```
1  var http = require('http');
2
3  http.createServer(function (req, res) {
4    res.writeHead(200,
5      {'Content-Type': 'text/html'});
6    res.write('Hello World!');
7    res.end();
8  }).listen(8080);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

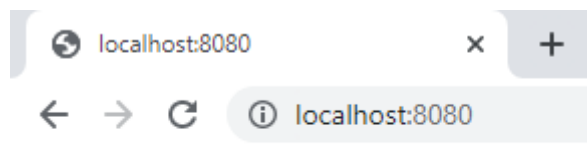
Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3

PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node httpHeader.js



Hello World!

Q:

## Node.js File System

```
var fs = require('fs');
```

Common use for the File System module:

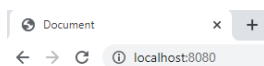
- Read files
- Create files
- Update files
- Delete files
- Rename files

A: Read files

```
JS readFiles.js X demofile1.html
week3Monday > exerciseN3 > JS readFiles.js > ...
1 var http = require('http');
2 var fs = require('fs');
  Complexity is 3 Everything is cool!
3 http.createServer(function (req, res) {
4   fs.readFile('demofile1.html', function(err, data) {
5     res.writeHead(200, {'Content-Type': 'text/html'});
6     res.write(data);
7     return res.end();
8   });
9 }).listen(8080);
```

The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree for 'week3Monday'. The 'exerciseN3' folder is expanded, showing files like 'createFiles\_appendFile.js', 'createFiles\_open.js', 'createFiles\_writeFile.js', 'deleteFiles.js', 'demofile1.html', 'getUrl.js', 'httpHeader.js', 'myrenamedfile.txt', 'queryString.js', 'readFiles.js', 'renameFiles.js', 'updateFiles\_appendFile.js', 'updateFiles\_writeFile.js', and 'updateFiles.js'. The 'demofile1.html' file is selected. The main editor area shows the content of 'demofile1.html', which is an HTML document with a doctype, meta tags for charset, language, and viewport, a title 'Document', and a body containing an h1 header 'My Header' and a paragraph 'My paragraph'.

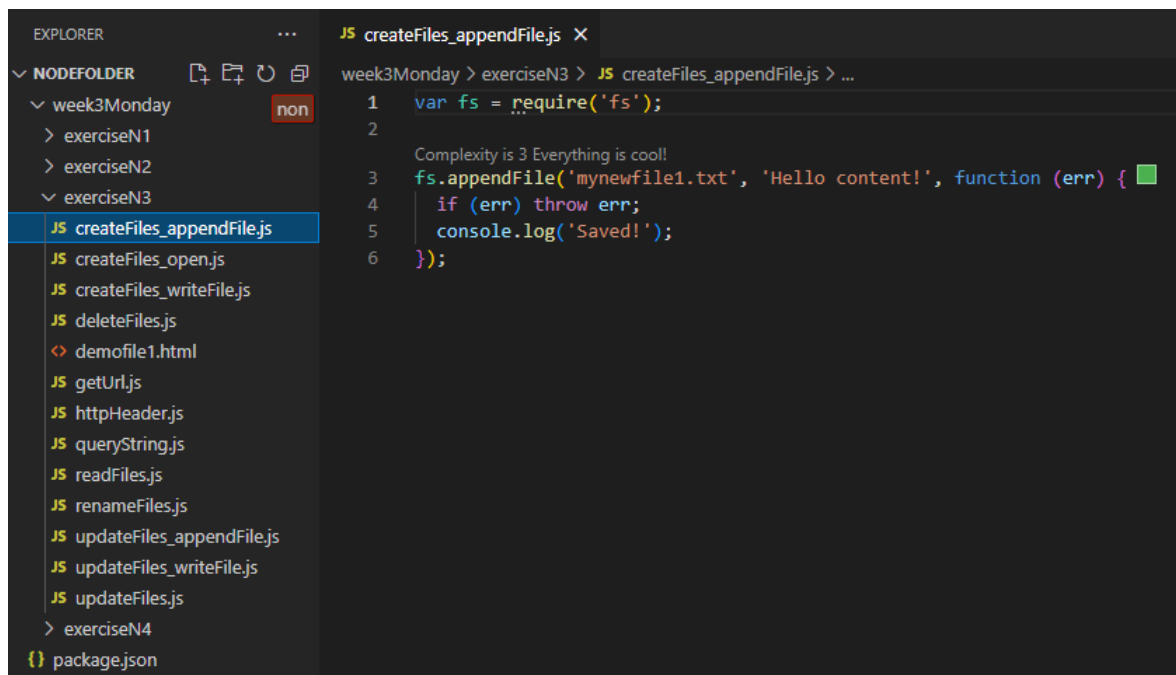
The screenshot shows the VS Code Terminal window. It displays the Windows PowerShell prompt. The user has navigated to the directory 'C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3' and run the command 'node readFiles.js'. The terminal output shows the text 'Complexity is 3 Everything is cool!' which is the content of the 'demofile1.html' file being served by the Node.js application.



## My Header

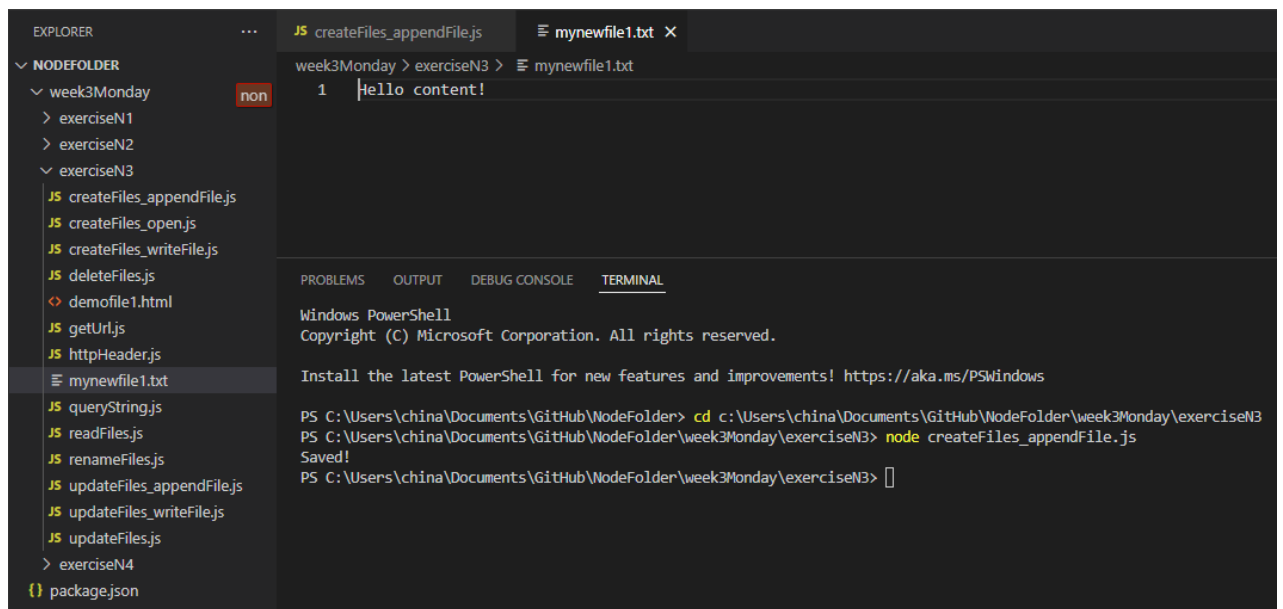
My paragraph.

## Create files –appendFile



The screenshot shows the VS Code interface. In the Explorer on the left, the file `createFiles_appendFile.js` is selected under the `exerciseN3` folder. The Editor on the right displays the content of this file:

```
1 var fs = require('fs');
2
3 Complexity is 3 Everything is cool!
4 fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
5   if (err) throw err;
6   console.log('Saved!');
7 });
```



The screenshot shows the VS Code interface with the `mynewfile1.txt` file open in the Editor. The file contains the text `Hello content!`. Below the Editor, the TERMINAL tab is active, showing the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node createFiles_appendFile.js
Saved!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3>
```



## Create files –open

The image shows two screenshots of the Visual Studio Code editor. The top screenshot shows the Explorer sidebar with a project structure under 'NODEFOLDER'. The 'week3Monday' folder is expanded, showing subfolders 'exerciseN1', 'exerciseN2', and 'exerciseN3'. Inside 'exerciseN3', several JavaScript files are listed, including 'createFiles\_appendFile.js', 'createFiles\_open.js' (which is selected), 'createFiles\_writeFile.js', 'deleteFiles.js', 'demoFile1.html', 'getUrl.js', 'httpHeader.js', 'mynewfile1.txt', 'queryString.js', 'readFiles.js', 'renameFiles.js', 'updateFiles\_appendFile.js', 'updateFiles\_writeFile.js', and 'updateFiles.js'. The main editor area shows the content of 'createFiles\_open.js', which contains a Node.js script using the 'fs' module to create a file named 'mynewfile2.txt' and log 'Saved!' to the console. The bottom screenshot shows the same project structure, but now 'mynewfile2.txt' is open in the editor. Below the editor, the 'TERMINAL' panel is active, showing the command prompt output of running 'node createFiles\_open.js' in the 'week3Monday/exerciseN3' directory, which results in 'Saved!' being printed.

```
EXPLORER  ...  JS createFiles_open.js X
week3Monday > exerciseN3 > JS createFiles_open.js > ...
1  var fs = require('fs');
2
   Complexity is 3 Everything is cool!
3  fs.open('mynewfile2.txt', 'w', function (err, file) {
4    if (err) throw err;
5    console.log('Saved!');
6  });
```

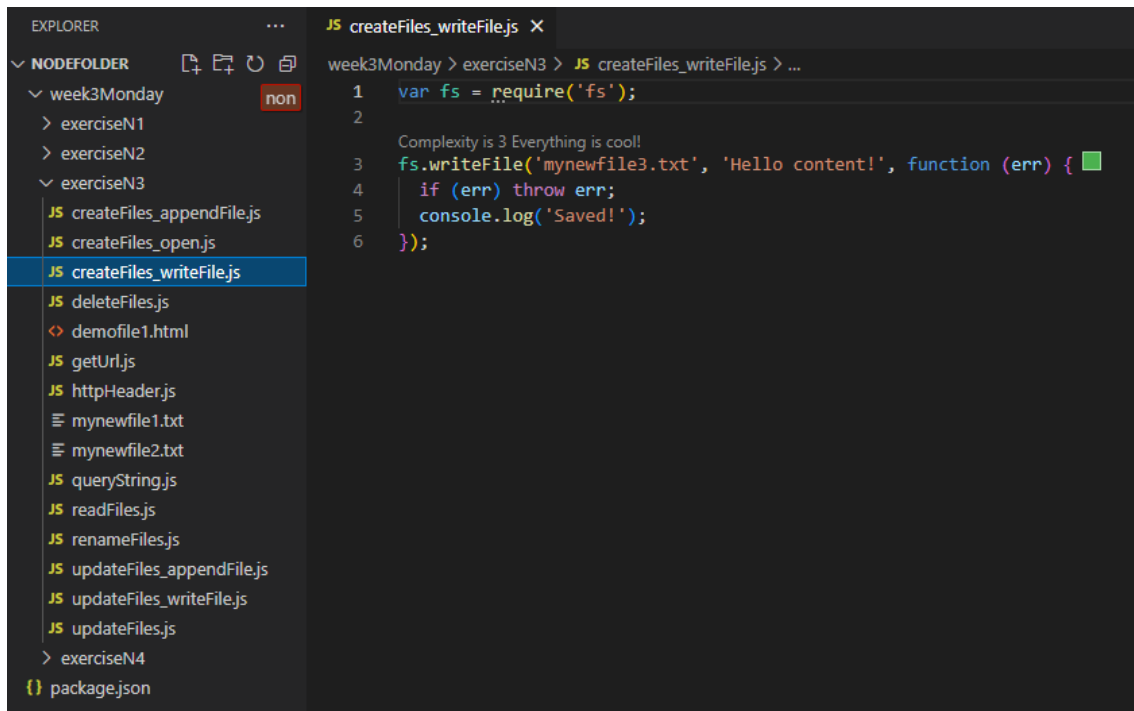
```
EXPLORER  ...  JS createFiles_open.js  mynewfile2.txt X
week3Monday > exerciseN3 > mynewfile2.txt
1

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

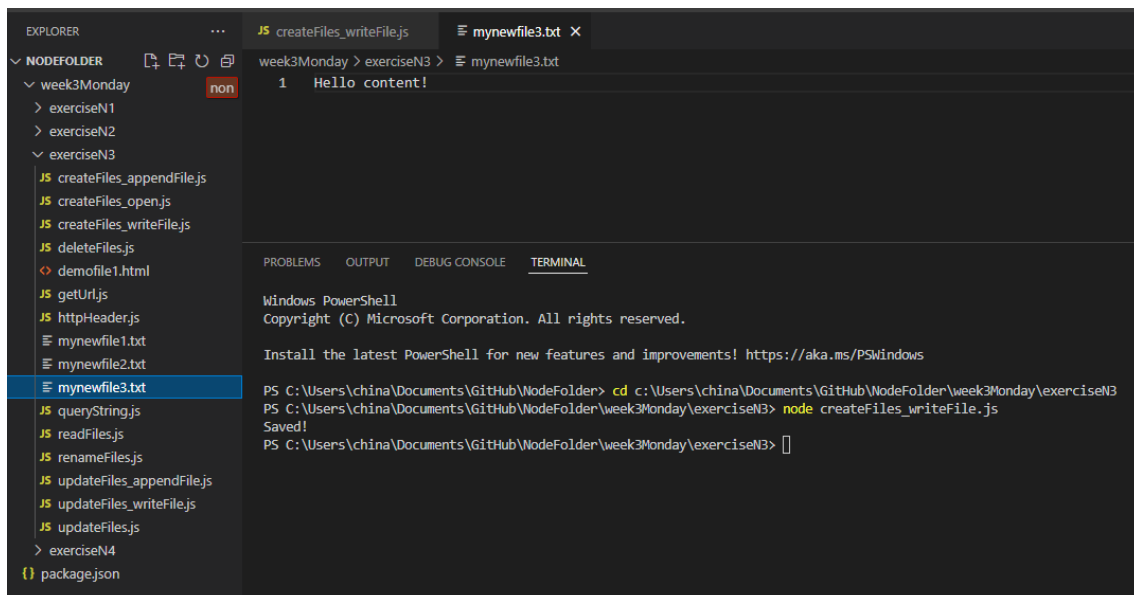
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node createFiles_open.js
Saved!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3>
```

## Create files –writeFile



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree with folders 'week3Monday', 'exerciseN1', 'exerciseN2', and 'exerciseN3'. Under 'exerciseN3', several JavaScript files are listed, including 'createFiles\_writeFile.js' which is currently selected. The main editor pane shows the code for 'createFiles\_writeFile.js' with the following content:

```
1 var fs = require('fs');
2
3 Complexity is 3 Everything is cool!
4 fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {
5   if (err) throw err;
6   console.log('Saved!');
7 });
```



This screenshot shows the same VS Code interface after running the script. The Explorer sidebar is identical. The Editor pane now shows 'mynewfile3.txt' with the content 'Hello content!'. Below the editor, the Terminal pane is active, displaying the command prompt output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node createFiles_writeFile.js
Saved!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3>
```

## Update files –appendFile

```
EXPLORER  ...  JS updateFiles_appendFile.js  mynewfile1.txt
week3Monday > exerciseN3 > JS updateFiles_appendFile.js > ...
  week3Monday non
    > exerciseN1
    > exerciseN2
    > exerciseN3
      JS createFiles_appendFile.js
      JS createFiles_open.js
      JS createFiles_writeFile.js
      JS deleteFiles.js
      demoFile1.html
      JS getUrl.js
      JS httpHeader.js
      mynewfile1.txt
      mynewfile2.txt
      mynewfile3.txt
      JS queryString.js
      JS readFiles.js
      JS renameFiles.js
      JS updateFiles_appendFile.js
      JS updateFiles_writeFile.js
    > exerciseN4
  () package.json

JS updateFiles_appendFile.js
1 var fs = require('fs');
2
3 Complexity is 3 Everything is cool!
4 fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {
5   if (err) throw err;
6   console.log('Updated!');
7 });
```

```
EXPLORER  ...  JS updateFiles_appendFile.js  mynewfile1.txt
week3Monday > exerciseN3 > mynewfile1.txt
  week3Monday non
    > exerciseN1
    > exerciseN2
    > exerciseN3
      JS createFiles_appendFile.js
      JS createFiles_open.js
      JS createFiles_writeFile.js
      JS deleteFiles.js
      demoFile1.html
      JS getUrl.js
      JS httpHeader.js
      mynewfile1.txt
      mynewfile2.txt
      mynewfile3.txt
      JS queryString.js
      JS readFiles.js
      JS renameFiles.js
      JS updateFiles_appendFile.js
      JS updateFiles_writeFile.js
    > exerciseN4
  () package.json

JS updateFiles_appendFile.js
1 Hello content!
```

```
EXPLORER  ...  JS updateFiles_appendFile.js  mynewfile1.txt
week3Monday > exerciseN3 > mynewfile1.txt
  week3Monday non
    > exerciseN1
    > exerciseN2
    > exerciseN3
      JS createFiles_appendFile.js
      JS createFiles_open.js
      JS createFiles_writeFile.js
      JS deleteFiles.js
      demoFile1.html
      JS getUrl.js
      JS httpHeader.js
      mynewfile1.txt
      mynewfile2.txt
      mynewfile3.txt
      JS queryString.js
      JS readFiles.js
      JS renameFiles.js
      JS updateFiles_appendFile.js
      JS updateFiles_writeFile.js
    > exerciseN4
  () package.json

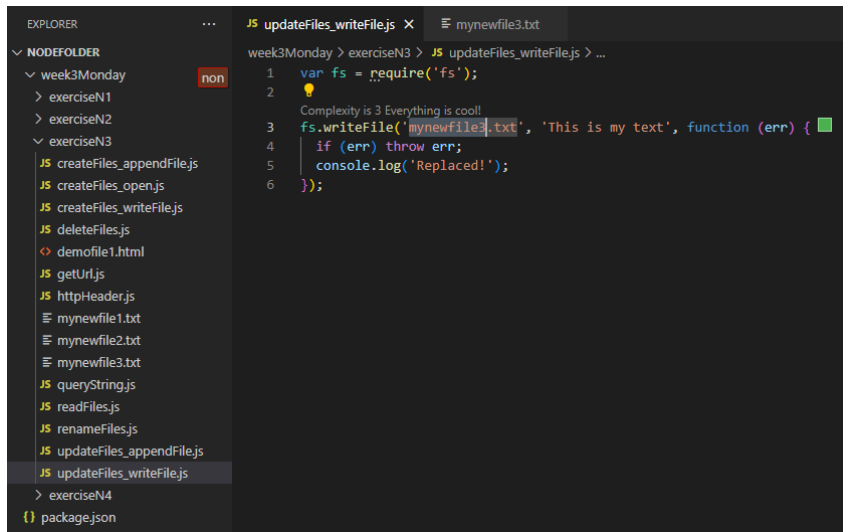
JS updateFiles_appendFile.js
1 Hello content! This is my text.

TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

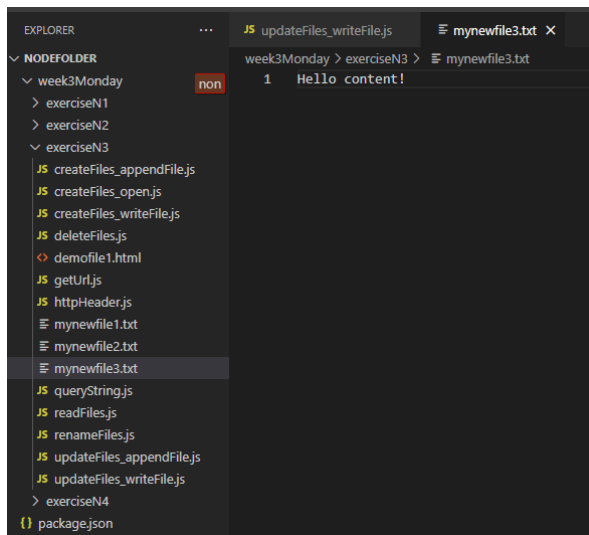
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node updateFiles_appendFile.js
Updated!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> |
```

## Update files –writeFile



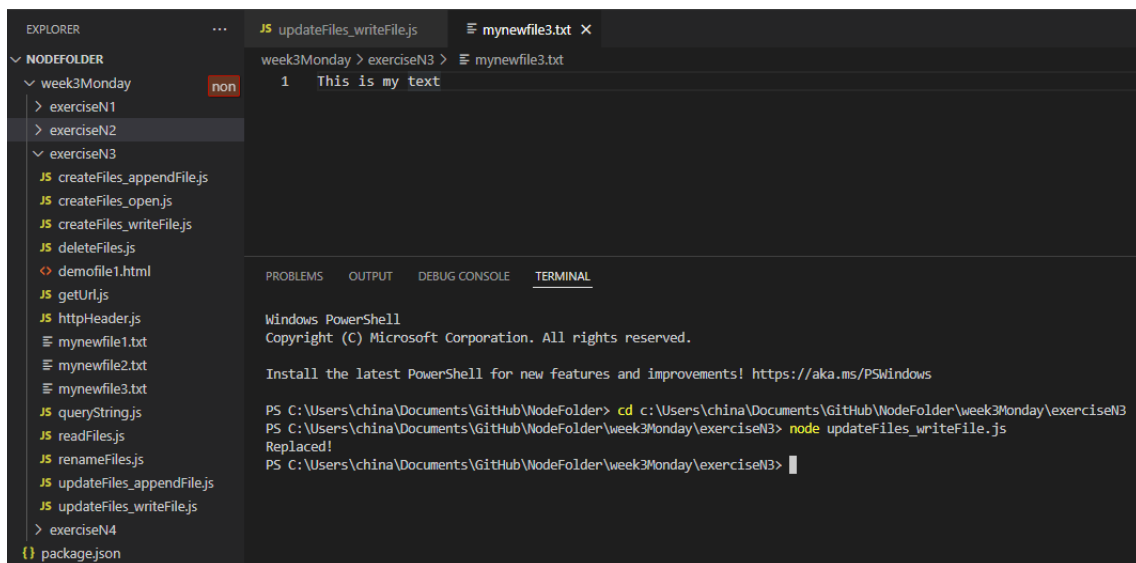
The Explorer sidebar shows a project structure with folders week3Monday, exerciseN1, exerciseN2, and exerciseN3. The exerciseN3 folder is expanded, showing files like createFiles\_appendFile.js, createFiles\_open.js, createFiles\_writeFile.js, deleteFiles.js, demoFile1.html, getUrl.js, httpHeader.js, mynewfile1.txt, mynewfile2.txt, mynewfile3.txt, queryString.js, readFiles.js, renameFiles.js, updateFiles\_appendFile.js, and updateFiles\_writeFile.js. The updateFiles\_writeFile.js file is selected. The Editor shows the code for updateFiles\_writeFile.js, which uses fs.writeFile to write 'This is my text' to mynewfile3.txt. A tooltip for fs.writeFile is visible.

```
1 var fs = require('fs');
2
3 fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {
4   if (err) throw err;
5   console.log('Replaced!');
6 });
```



The Explorer sidebar shows the same project structure. The mynewfile3.txt file is selected. The Editor shows the content of mynewfile3.txt, which is 'Hello content!'.

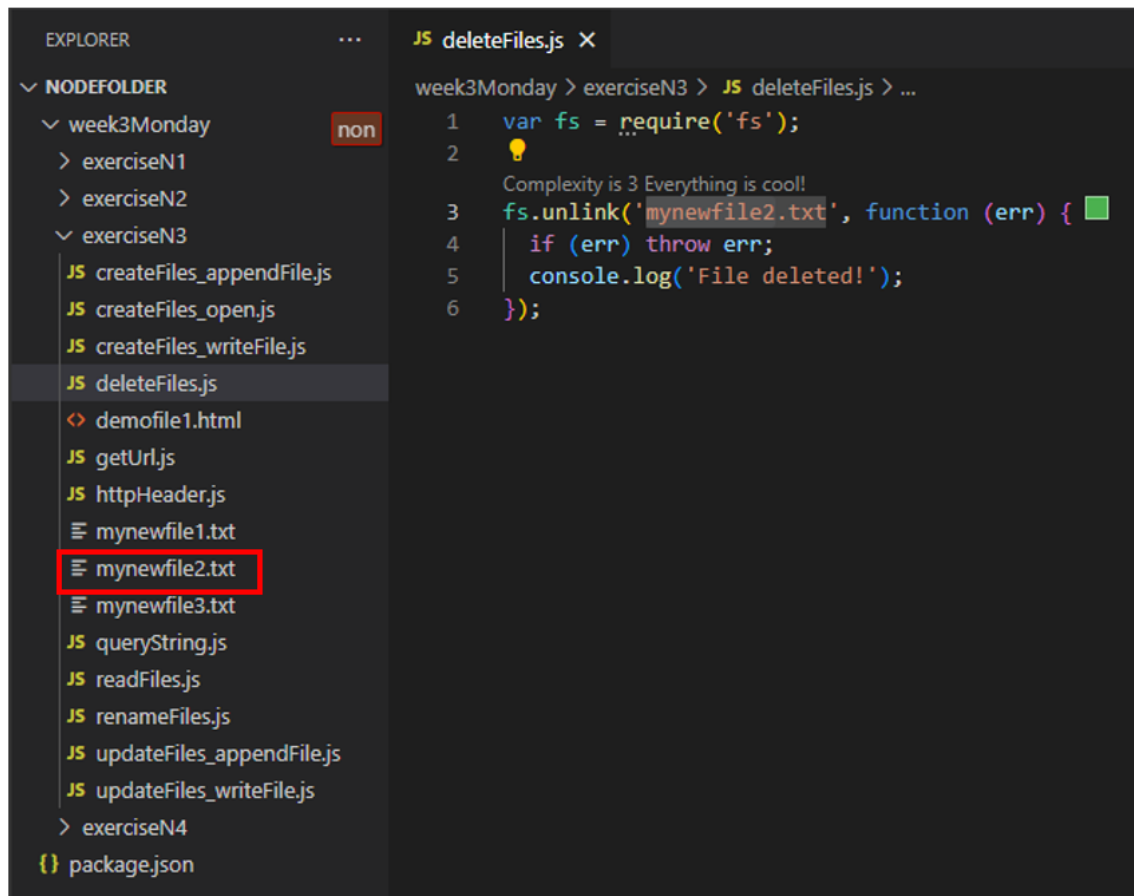
```
1 Hello content!
```



The Explorer sidebar shows the same project structure. The exerciseN3 folder is expanded, showing the mynewfile3.txt file. The Editor shows the content of mynewfile3.txt, which is 'This is my text'. The Terminal shows the command to run the updateFiles\_writeFile.js file, which outputs 'Replaced!'.

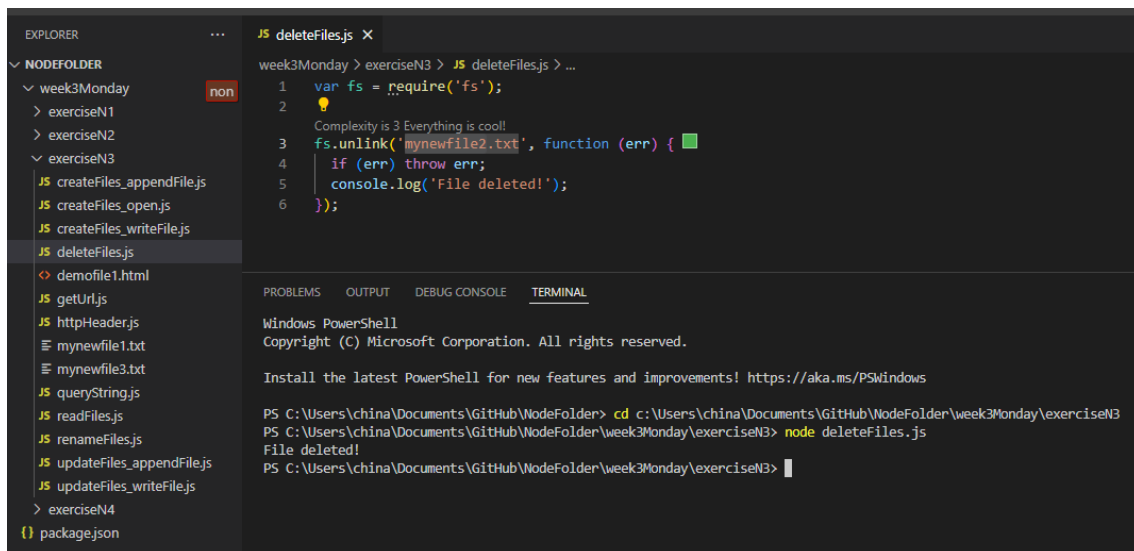
```
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node updateFiles_writeFile.js
Replaced!
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3>
```

## Delete files



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree for a project named 'NODEFOLDER'. The tree includes a 'week3Monday' folder, which contains subfolders 'exerciseN1', 'exerciseN2', and 'exerciseN3'. Inside 'exerciseN3', there are several JavaScript files: 'createFiles\_appendFile.js', 'createFiles\_open.js', 'createFiles\_writeFile.js', 'deleteFiles.js' (which is selected and highlighted), 'demoFile1.html', 'getUrl.js', 'httpHeader.js', 'mynewfile1.txt', 'mynewfile2.txt' (highlighted with a red rectangle), 'mynewfile3.txt', 'queryString.js', 'readFiles.js', 'renameFiles.js', 'updateFiles\_appendFile.js', and 'updateFiles\_writeFile.js'. The main Editor pane shows the content of 'deleteFiles.js'. The code is as follows:

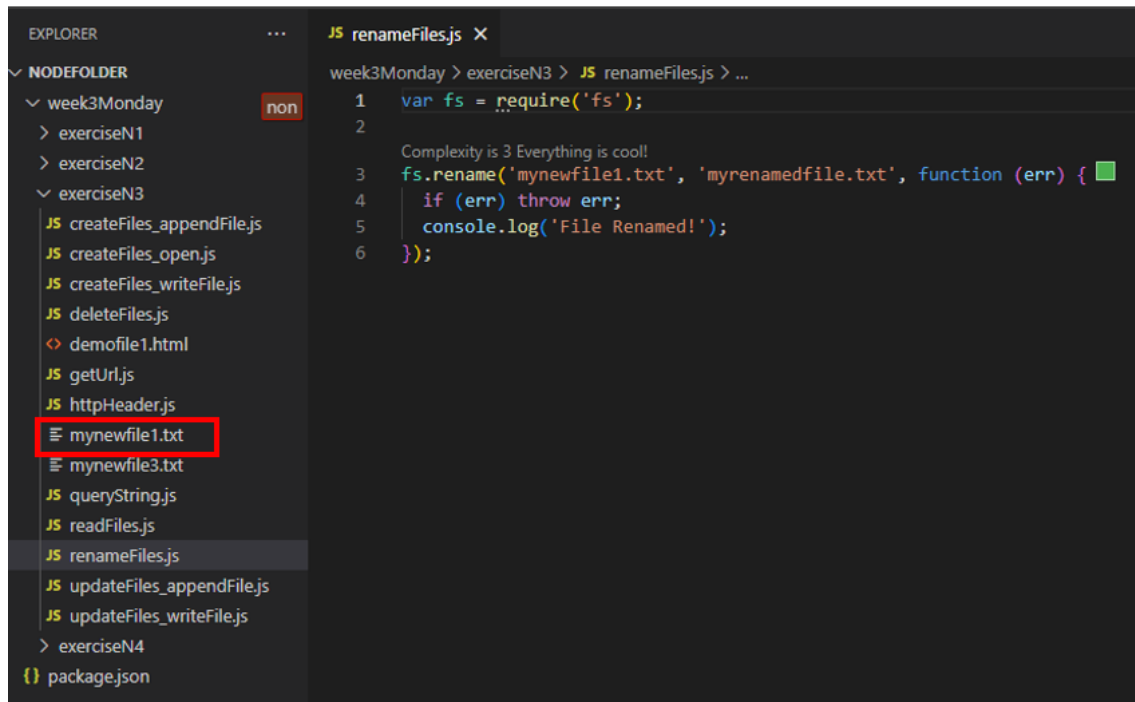
```
1 var fs = require('fs');
2
3 fs.unlink('mynewfile2.txt', function (err) {
4   if (err) throw err;
5   console.log('File deleted!');
6 });
```



This screenshot shows the same VS Code interface as the previous one, but with the 'TERMINAL' tab active at the bottom. The terminal displays the command to run the script and its output:

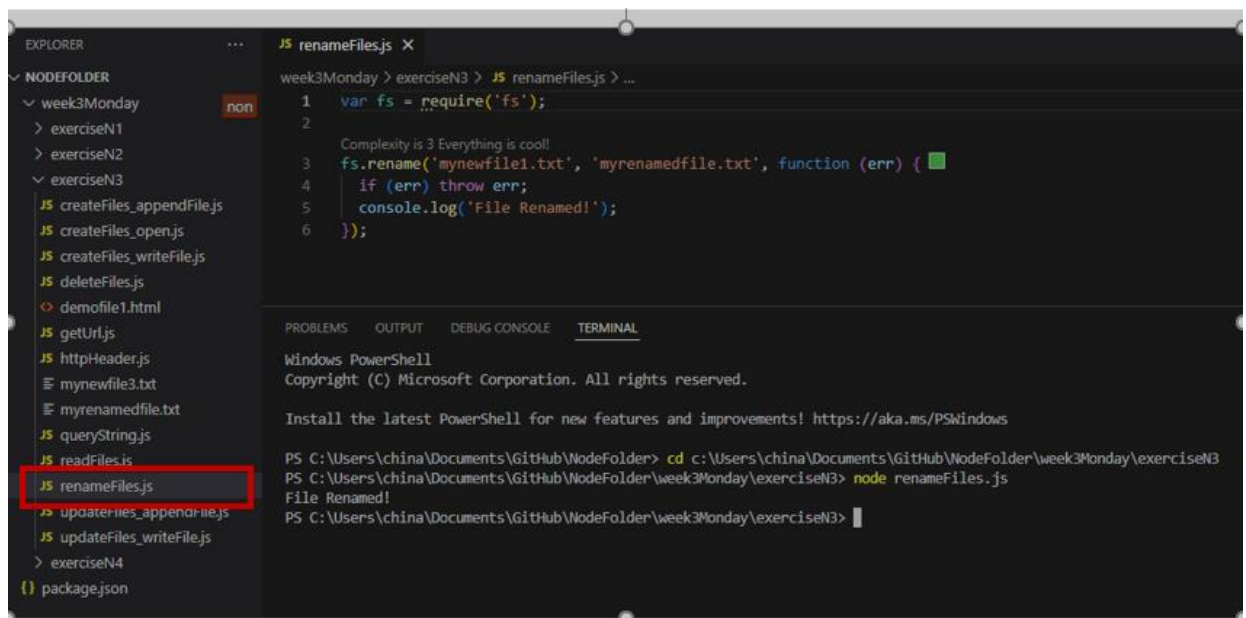
```
PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN3> node deleteFiles.js
File deleted!
```

## Rename files



This screenshot shows the VS Code Explorer on the left and the editor on the right. In the Explorer, the file `mynewfile1.txt` is highlighted with a red box. The editor displays the content of `renameFiles.js`, which uses the `fs.rename` method to rename the file.

```
1 var fs = require('fs');
2
3 Complexity is 3 Everything is cool!
4 fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {
5   if (err) throw err;
6   console.log('File Renamed!');
7 });
```



This screenshot shows the VS Code Explorer, editor, and terminal. In the Explorer, `renameFiles.js` is highlighted with a red box. The terminal shows the command `node renameFiles.js` being executed, resulting in the output `File Renamed!`.

```
PS C:\Users\china\Documents\Github\NodeFolder> cd c:\Users\china\Documents\Github\NodeFolder\week3Monday\exerciseN3
PS C:\Users\china\Documents\Github\NodeFolder\week3Monday\exerciseN3> node renameFiles.js
File Renamed!
PS C:\Users\china\Documents\Github\NodeFolder\week3Monday\exerciseN3>
```

## Exercise N4: Simple Adder

**Q:**

Opening HTML page

(SimpleAdder.html)

```
<form action =  
"http://localhost:8080/add.js">  
<div>Enter two numbers  
<input type = "text" name="first"/><br/>  
<input type = "text" name="second"/><br/>  
<input type="submit" value="Click"/>  
</div>  
</form>
```

Module to Perform Addition

addmod.js

```
exports.add = function (req,res,vals) {  
  var sum = parseInt(vals.first) + parseInt(vals.second);  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("<!DOCTYPE html>");  
  res.write("<html>");  
  res.write("<head><meta charset='utf-8'/>");  
  res.write("<title>Calculator Web Site</title>");  
  res.write("</head>");  
  res.write("<body>");  
  res.write("<p>The sum is: ");  
  res.write(String(sum));  
  res.write("</p>");  
  res.write("</body>");  
  res.write("</html>");  
  return res.end();  
};
```

Web Server for the Adder

AdderWebServer.js

```
var http = require('http');  
var url = require('url');
```

```

var fs = require('fs');
var addmod = require('./addmod.js');

http.createServer(function (req, res) {
  var q = url.parse(req.url, true);
  var filename = "." + q.pathname;
  if (q.pathname==="/add.js")
    addmod.add(req,res,q.query)
  else
    fs.readFile(filename, function(err, data) {
      if (err) {
        res.writeHead(404, {'Content-Type': 'text/html'});
        return res.end("404 Not Found");
      }
      res.writeHead(200); // Content-Type not included
      res.write(data);
      return res.end();
    });
}).listen(8080);};

```

Run the Simple Adder

- To run the Simple Adder website, use the following URL:

<http://localhost:8080/SimpleAdder.html>

A:



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with a folder named 'week3Monday' containing subfolders 'exerciseN1', 'exerciseN2', 'exerciseN3', and 'exerciseN4'. Inside 'exerciseN4', there are files 'AdderWebServer.js', 'addmod.js', and 'SimpleAdder.html'. The 'addmod.js' file is selected and its content is displayed in the main editor area. The code defines an Express.js route for 'add' that calculates the sum of two numbers and returns an HTML response.

```
1 exports.add = function (req, res, vals) {
2   var sum = parseInt(vals.first) + parseInt(vals.second);
3   res.writeHead(200, { "Content-Type": "text/html" });
4   res.write("<!DOCTYPE html>");
5   res.write("<html>");
6   res.write('<head><meta charset="utf-8"/>');
7   res.write("<title>Calculator Web Site</title>");
8   res.write("</head>");
9   res.write("<body>");
10  res.write("<p>The sum is: ");
11  res.write(String(sum));
12  res.write("</p>");
13  res.write("</body>");
14  res.write("</html>");
15  return res.end();
16 };
17
```

The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows the same project structure as the previous screenshot. The 'AdderWebServer.js' file is selected and its content is displayed in the main editor area. The code sets up an Express.js server that listens on port 8080. It uses the 'addmod.js' module to handle the 'add.js' route. Comments in the code indicate the complexity of the code at different stages.

```
1 var http = require("http");
2 var url = require("url");
3 var fs = require("fs");
4 var addmod = require("../addmod.js");
5
6 http
7   Complexity is 7 It's time to do something...
8   .createServer(function (req, res) {
9     var q = url.parse(req.url, true);
10    var filename = "." + q.pathname;
11    if (q.pathname == "/add.js") addmod.add(req, res, q.query);
12    else
13      Complexity is 4 Everything is cool!
14      fs.readFile(filename, function (err, data) {
15        if (err) {
16          res.writeHead(404, { "Content-Type": "text/html" });
17          return res.end("404 Not Found");
18        }
19        res.writeHead(200); // Content-Type not included
20        res.write(data);
21        return res.end();
22      });
23  })
24  .listen(8080);
```

The screenshot shows the VS Code terminal with the PowerShell prompt. The terminal displays the Windows PowerShell logo and copyright information. It then shows the command to run the AdderWebServer.js file using node.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\china\Documents\GitHub\NodeFolder> cd c:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN4
PS C:\Users\china\Documents\GitHub\NodeFolder\week3Monday\exerciseN4> node AdderWebServer.js
```

