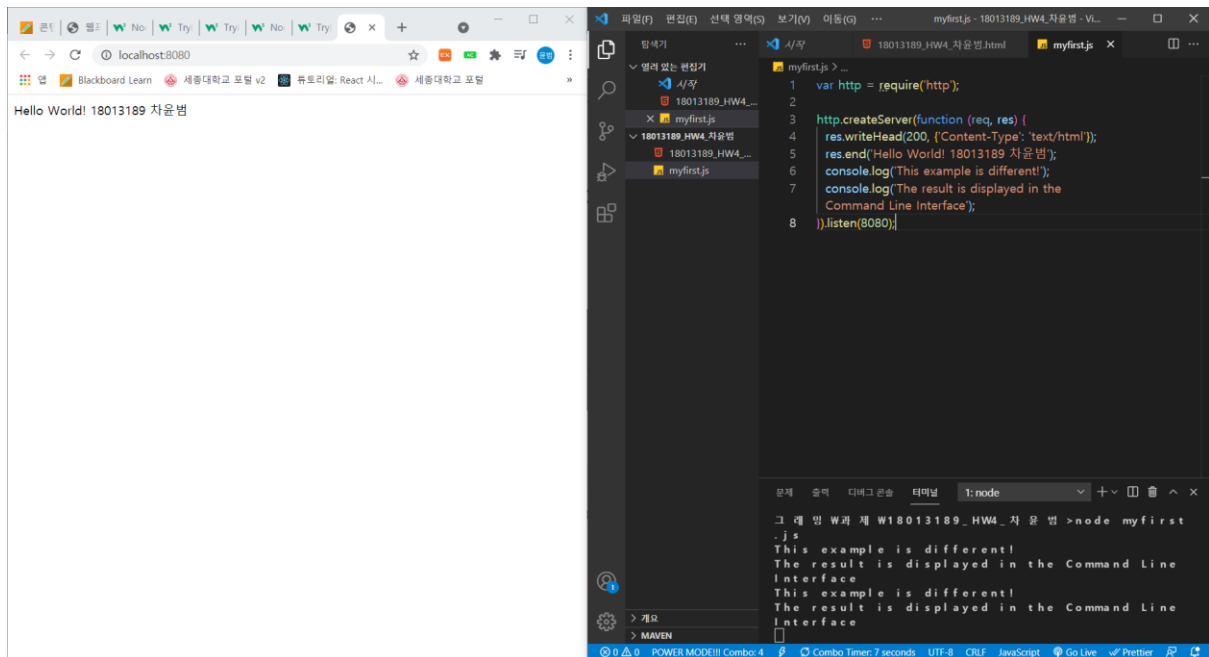


웹프로그래밍 HW4

18013189 컴퓨터공학과 차윤범

Node.js Tutorial Chapter

실행 코드와 결과



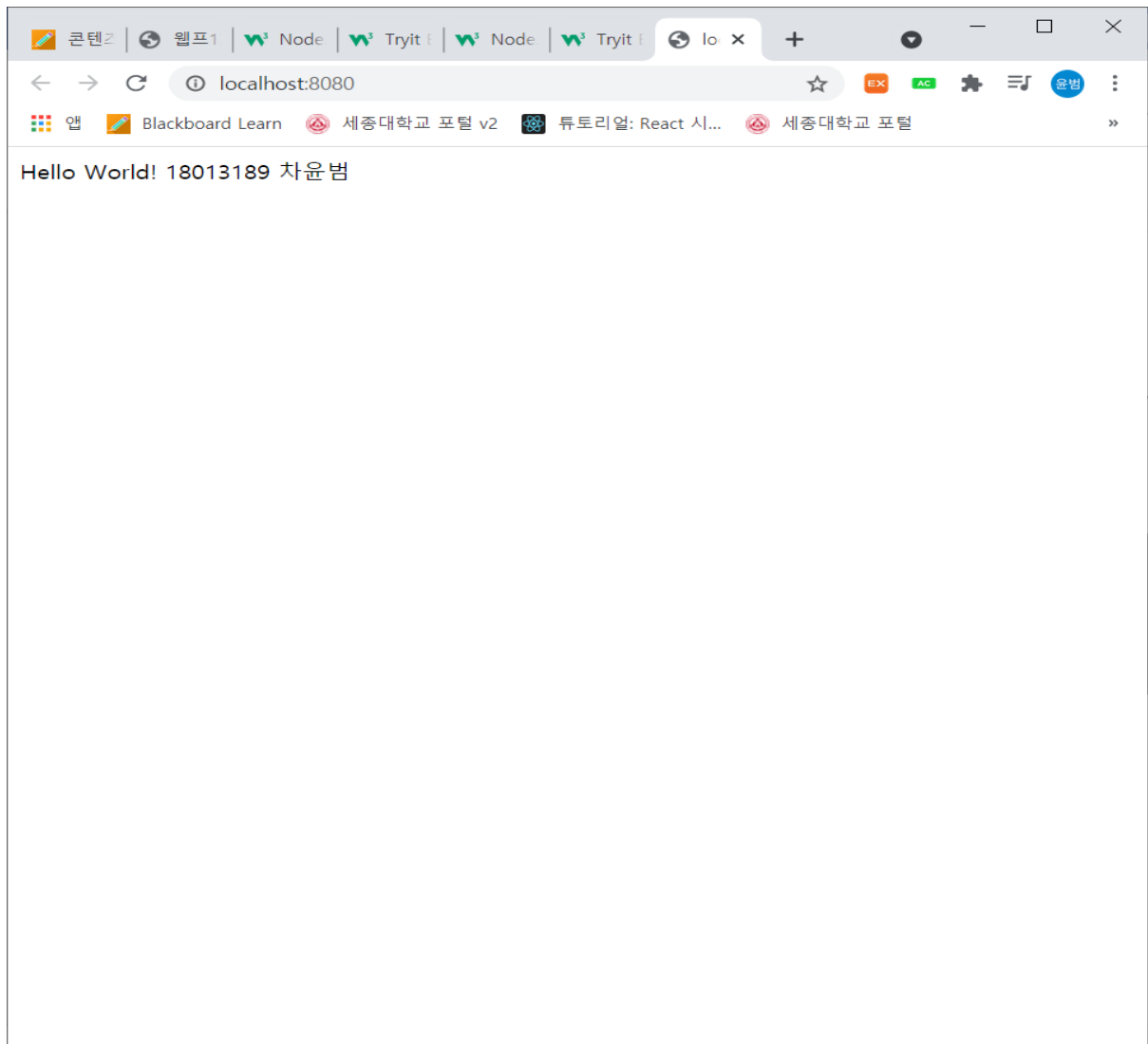
Node.js Get Started Chapter Run Example

실행 코드

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end('Hello World! 18013189 차윤범');
}).listen(8080);
```

실행 결과



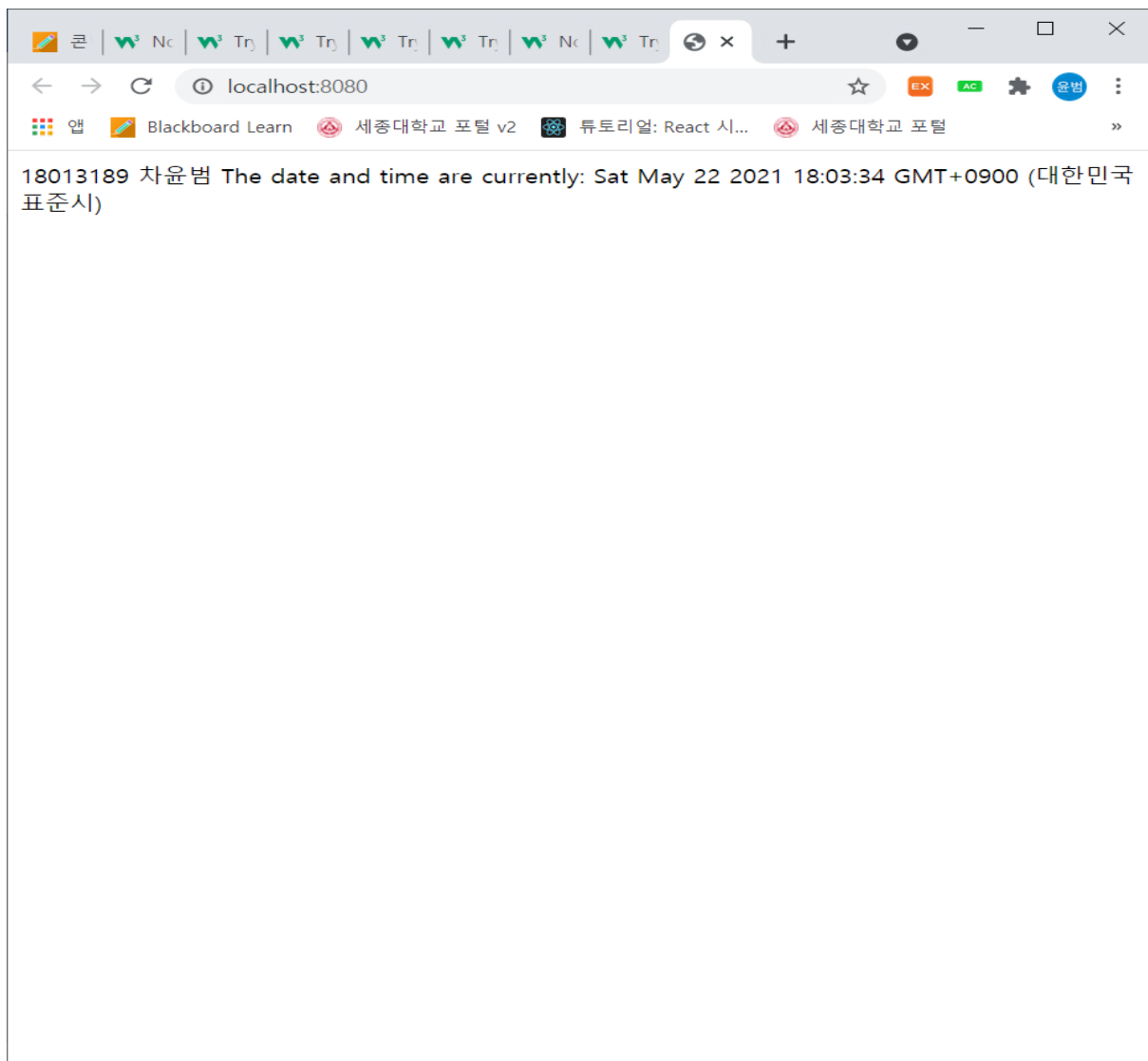
Node.js Modules Chapter

실행 코드

```
var http = require('http');
var dt = require('./myfirstmodule');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("18013189 차윤범");
  res.write("The date and time are currently: " + dt.myDateTime());
  res.end();
}).listen(8080);
```

실행 결과



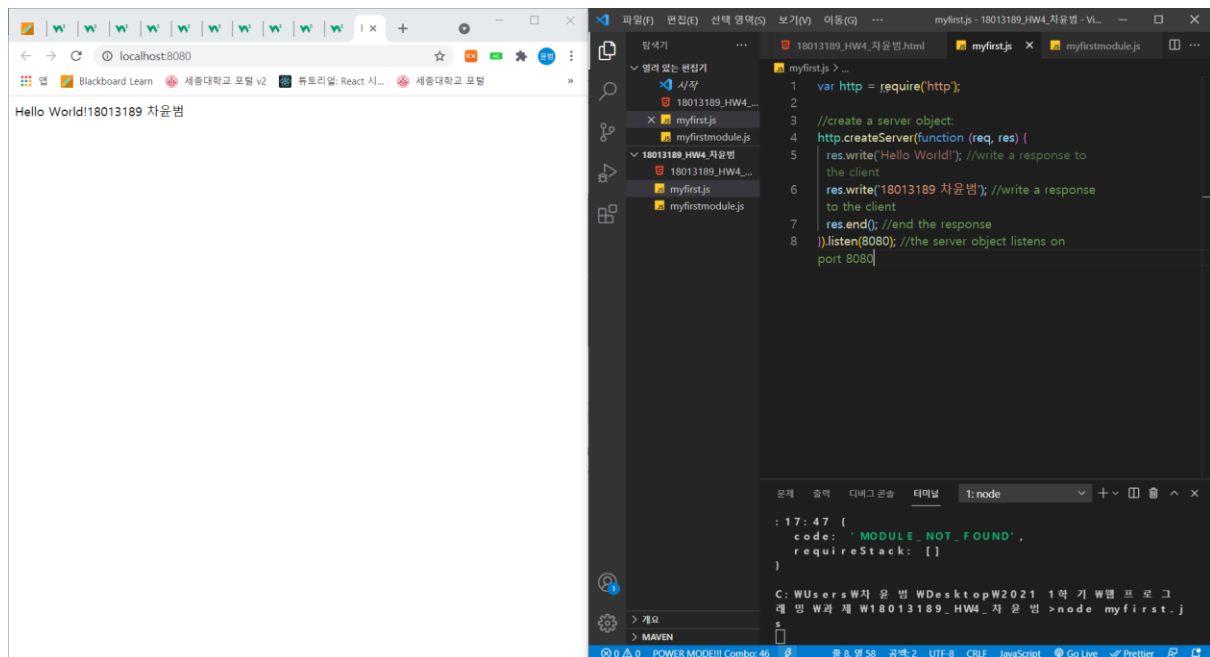
Node.js HTTP Module Chapter

실행 코드

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

//create a server object:
http.createServer(function (req, res) {
  res.write('Hello World!'); //write a response to the client
  res.write('18013189 차윤범'); //write a response to the client
  res.end(); //end the response
}).listen(8080); //the server object listens on port 8080
```

실행 결과



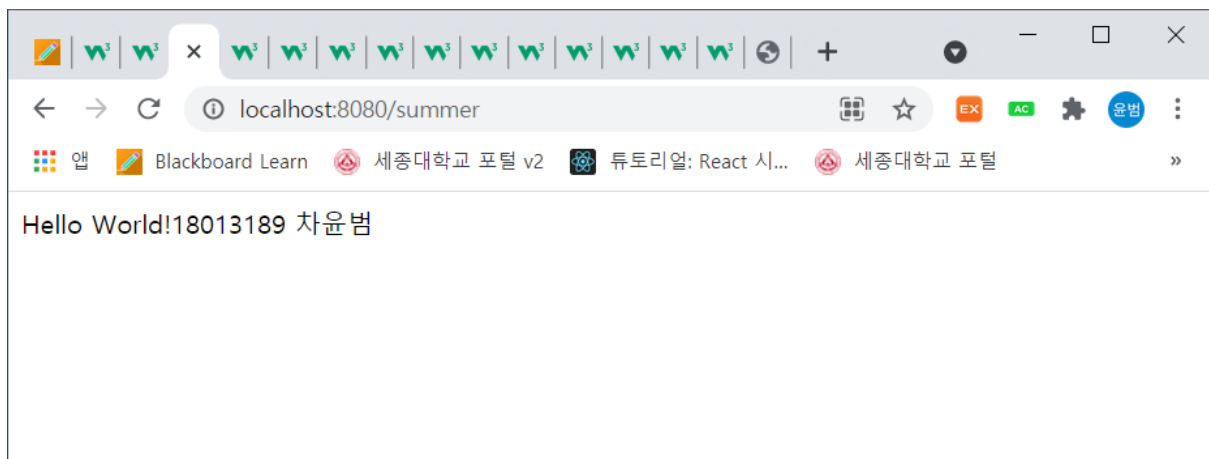
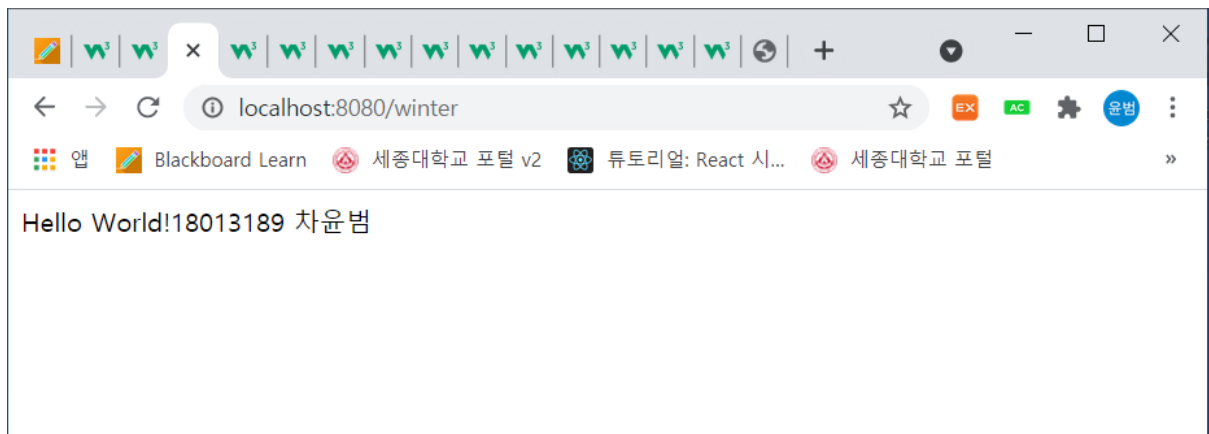
Node.js HTTP Module Chapter

Add an HTTP Header Part and Read the Query String Part

실행 코드

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url);
  res.write("18013189 차윤범");
  res.end();
}).listen(8080);
```

실행 결과



Split the Query String Part

실행 코드

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("18013189 차윤범 ");

  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(8080);
```

실행 결과



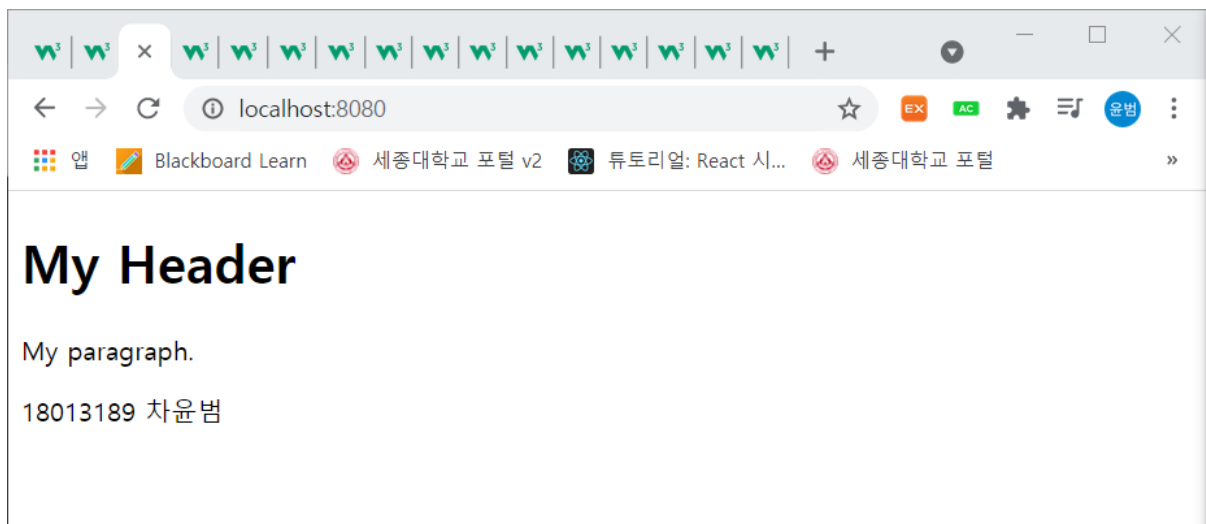
Node.js File System Module Chapter

Read Files Part

실행 코드

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('18013189_HW4_차윤범.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

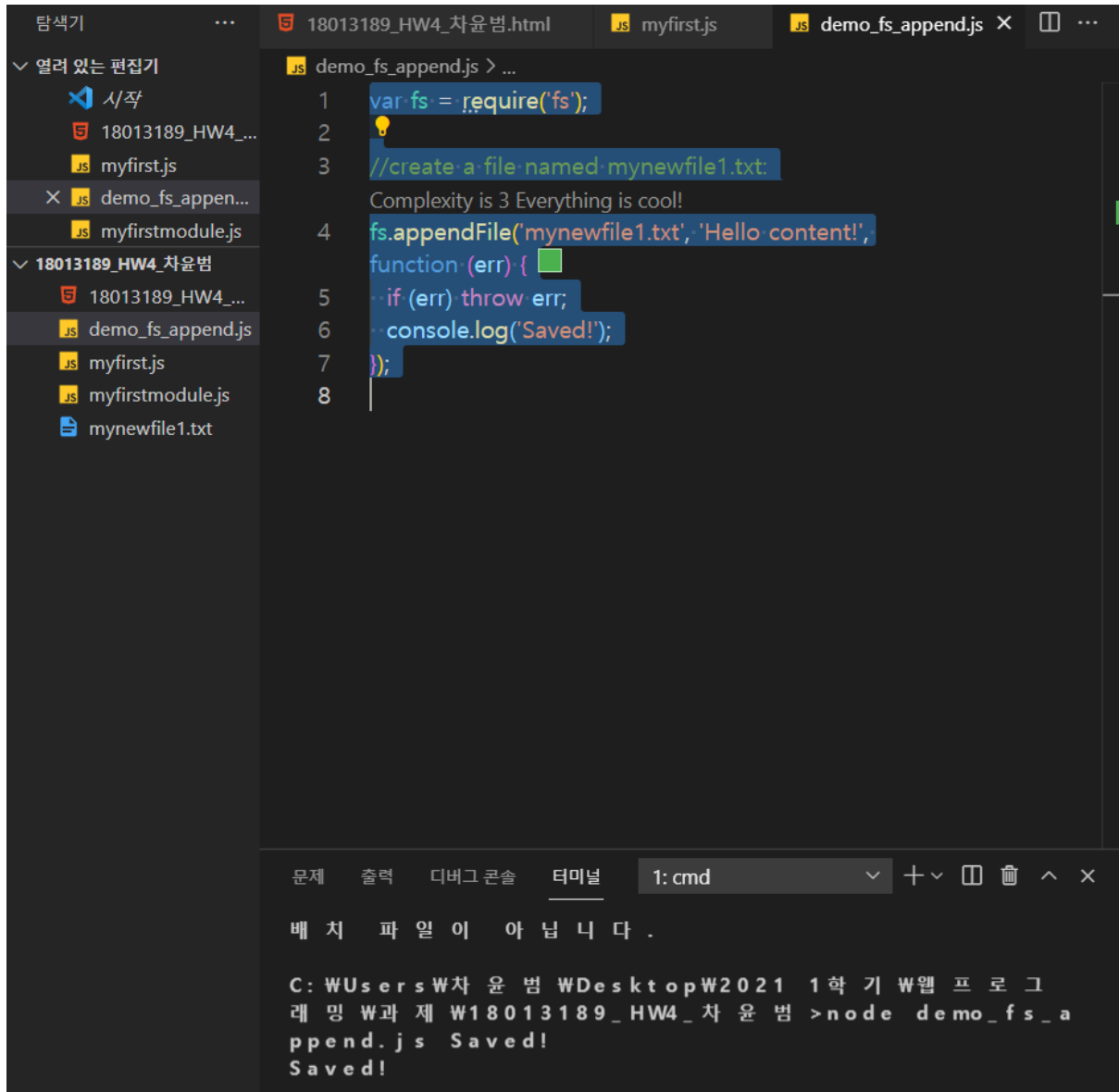
실행결과



Create Files Part

fs.appendFile()

실행 코드와 실행 결과



The image shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named '18013189_HW4_차윤범' with files 'demo_fs_append.js', 'myfirst.js', 'myfirstmodule.js', and 'mynewfile1.txt'. The code editor shows the following JavaScript code in 'demo_fs_append.js':

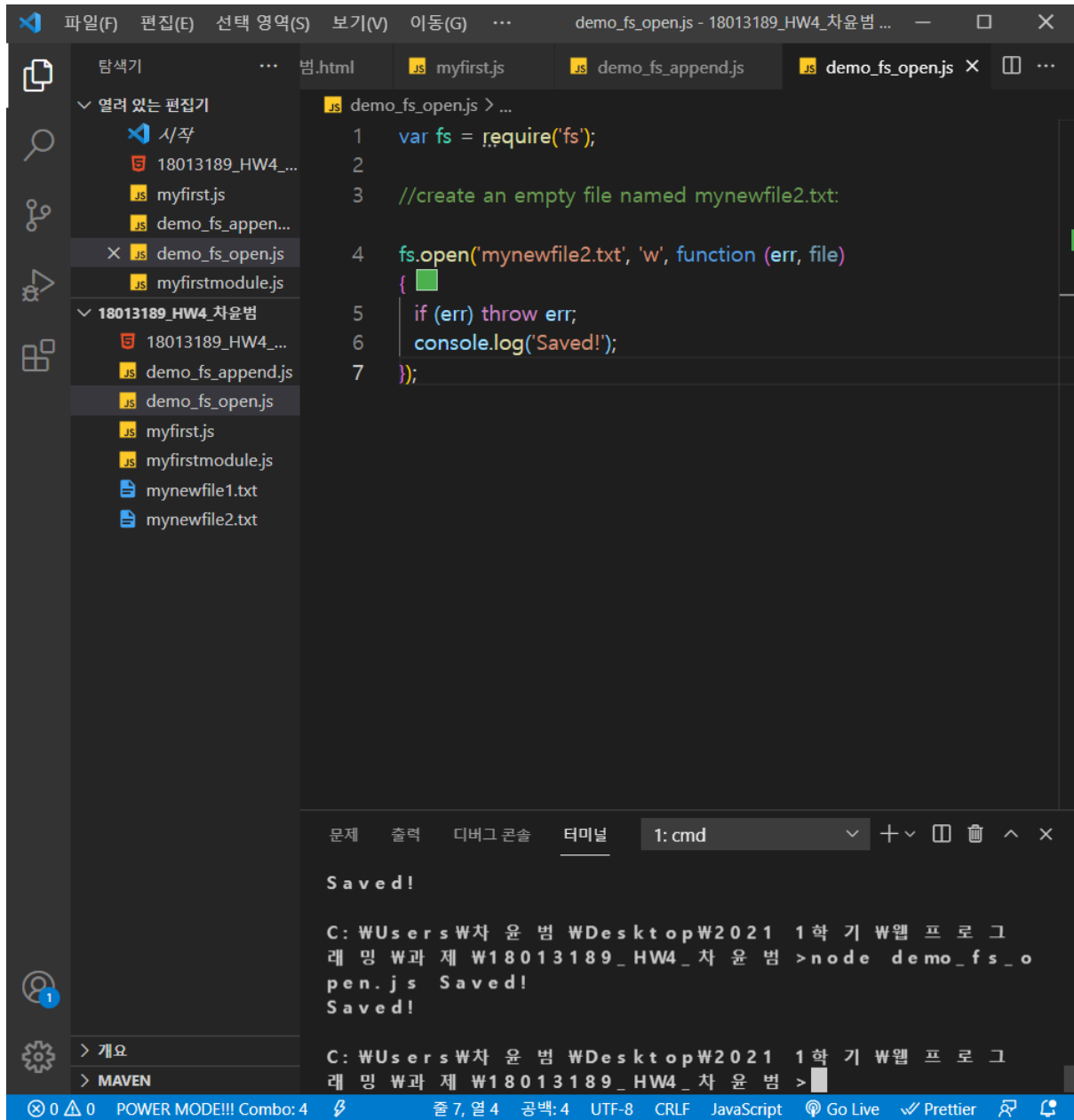
```
1 var fs = require('fs');
2
3 //create a file named mynewfile1.txt:
4 fs.appendFile('mynewfile1.txt', 'Hello content!',
5   function (err) {
6     if (err) throw err;
7     console.log("Saved!");
8   });
```

The terminal at the bottom shows the command 'node demo_fs_append.js' being executed, resulting in the output 'Saved!'.

```
C:\Users\차윤범\Desktop\2021 1학기 웹 프로그래밍 과제\18013189_HW4_차윤범>node demo_fs_append.js
Saved!
```


fs.open()

실행 코드와 결과



The screenshot shows the Visual Studio Code editor with the file `demo_fs_open.js` open. The code in the editor is as follows:

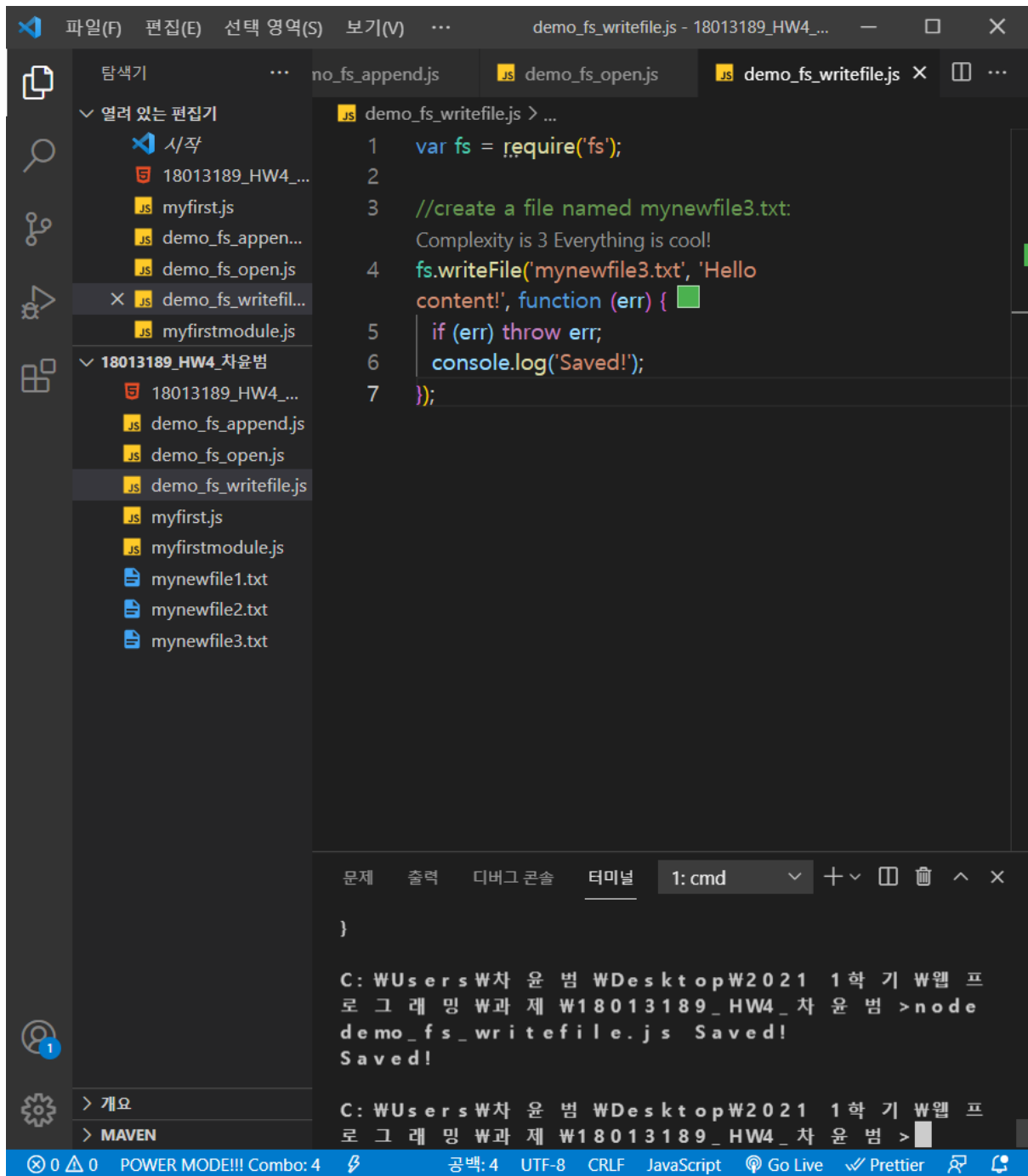
```
1 var fs = require('fs');
2
3 //create an empty file named mynewfile2.txt:
4 fs.open('mynewfile2.txt', 'w', function (err, file)
5 {
6   if (err) throw err;
7   console.log('Saved!');
8 });
```

The terminal output shows the command `node demo_fs_open.js` being executed, resulting in the message `Saved!` being printed to the console.

```
C:\Users\W차 윤 범\Desktop\W2021 1학기 웹 프로그래밍 과제 W18013189_HW4_차 윤 범>node demo_fs_open.js
Saved!
```

fs.writeFile()

실행 코드와 결과



The screenshot shows the Visual Studio Code editor with the file `demo_fs_writefile.js` open. The code in the editor is as follows:

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

The file explorer on the left shows the project structure, including the `demo_fs_writefile.js` file. The terminal at the bottom shows the command `node demo_fs_writefile.js` being executed, resulting in the output:

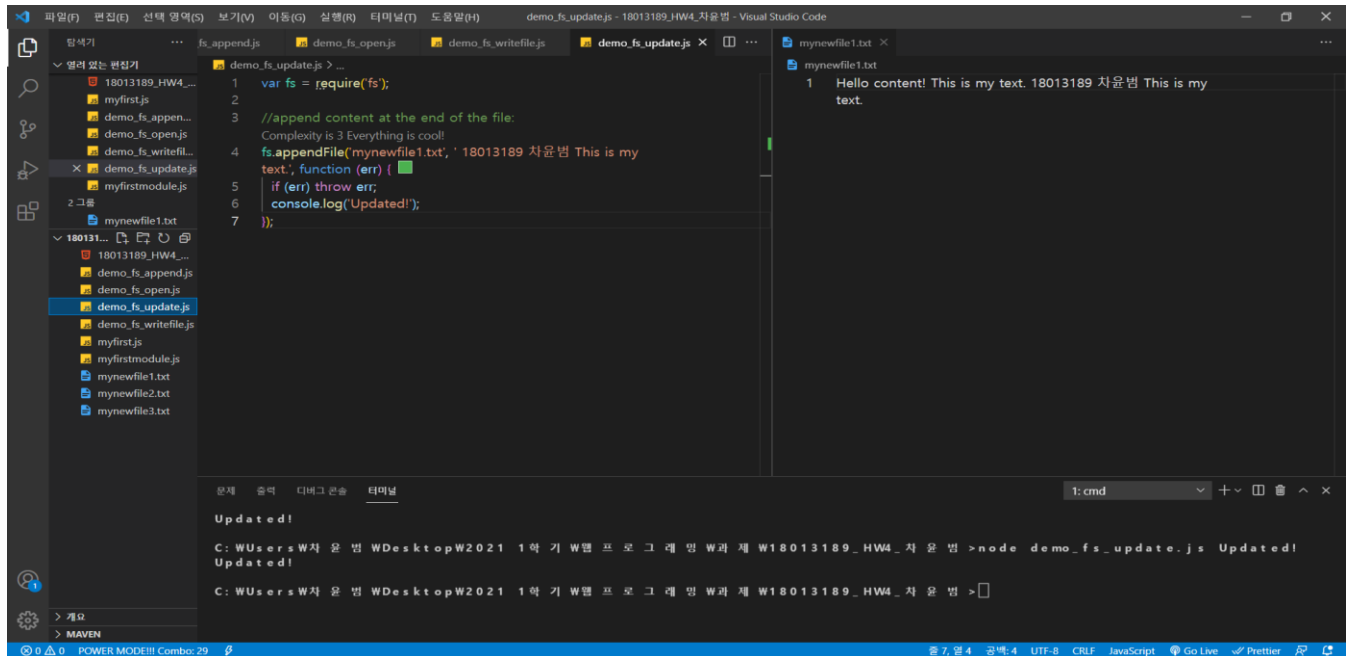
```
C: \Users\W차 윤 범\Desktop\2021 1 학 기 웹 프
로 그 래 밍 과 제 W18013189_HW4_차 윤 범 >node
demo_fs_writefile.js Saved!
Saved!
```

The status bar at the bottom indicates the current mode is `POWER MODE!!!` and the file encoding is `UTF-8`.

Update Files Part

fs.appendFile()

실행 코드와 결과



The screenshot shows the Visual Studio Code editor with the file `demo_fs_update.js` open. The code uses `fs.appendFile` to append content to `mynewfile1.txt`. The terminal output shows the command `node demo_fs_update.js` being executed, resulting in `Updated!` being printed twice.

```
1 var fs = require('fs');
2
3 //append content at the end of the file:
4 fs.appendFile('mynewfile1.txt', ' 18013189 차윤범 This is my
5 text.', function (err) {
6   if (err) throw err;
7   console.log('Updated!');
8 });
```

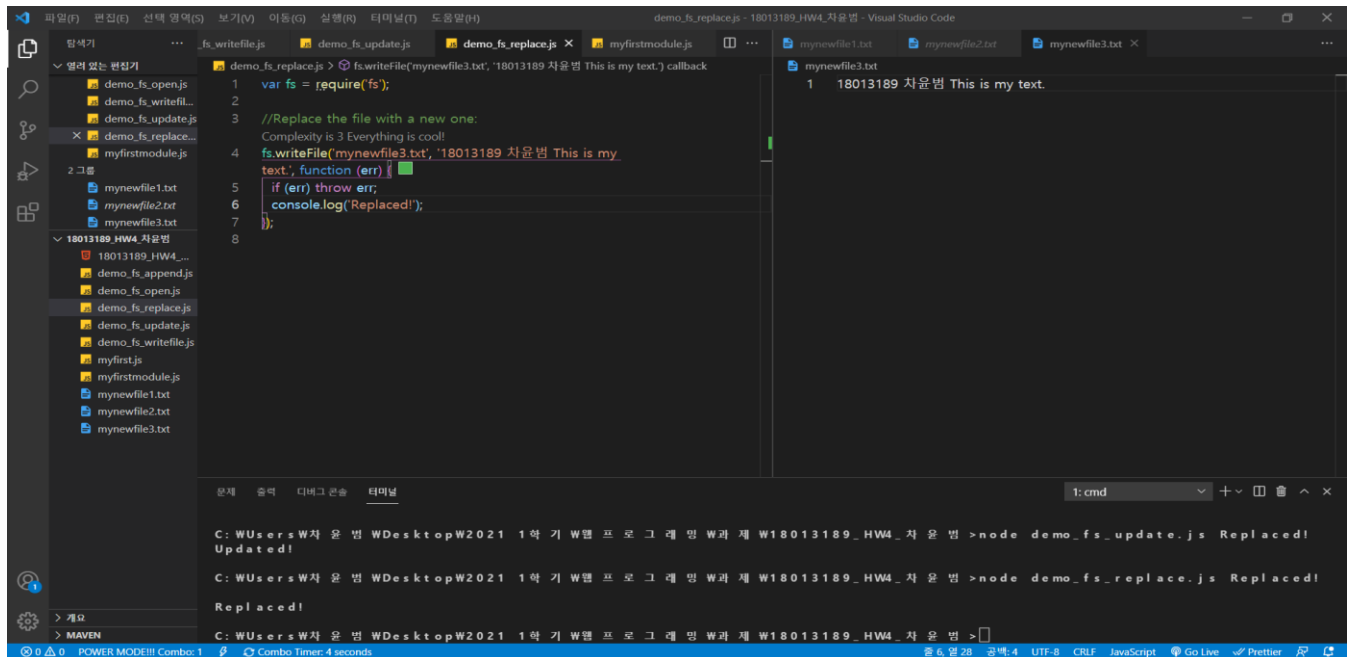
```
Updated!
Updated!
```

```
C:\Users\W자\온\W\Desktop\W2021\1학기\W\프로그래밍\과제\W18013189_HW4_차윤범>node demo_fs_update.js Updated!
Updated!
```

```
C:\Users\W자\온\W\Desktop\W2021\1학기\W\프로그래밍\과제\W18013189_HW4_차윤범>
```

fs.writeFile()

실행 코드와 결과



The screenshot shows the Visual Studio Code editor with the file `demo_fs_replace.js` open. The code uses `fs.writeFile` to replace the content of `mynewfile3.txt`. The terminal output shows the command `node demo_fs_replace.js` being executed, resulting in `Replaced!` being printed twice.

```
1 var fs = require('fs');
2
3 //Replace the file with a new one:
4 fs.writeFile('mynewfile3.txt', '18013189 차윤범 This is my
5 text.', function (err) {
6   if (err) throw err;
7   console.log('Replaced!');
8 });
```

```
Replaced!
Replaced!
```

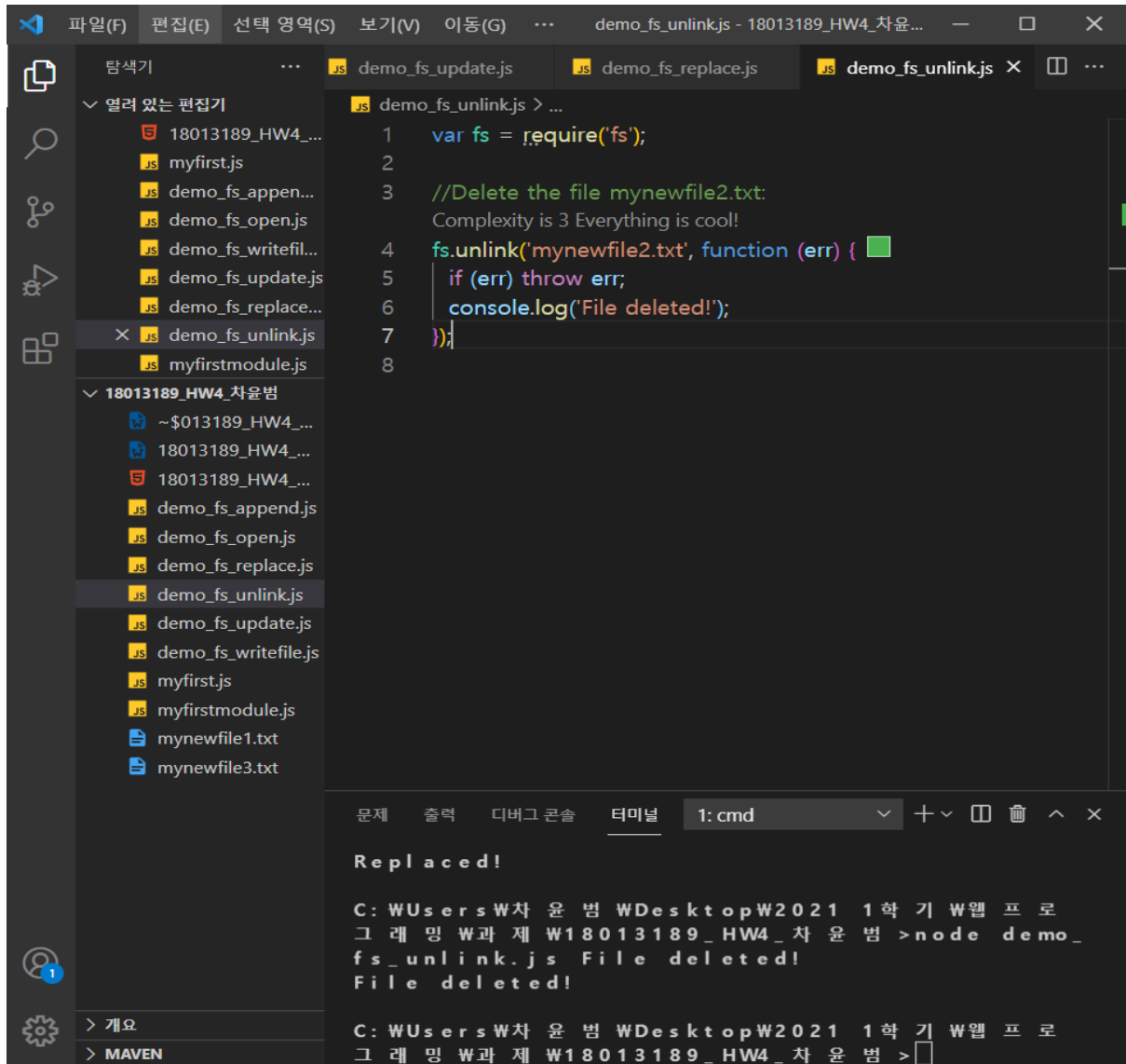
```
C:\Users\W자\온\W\Desktop\W2021\1학기\W\프로그래밍\과제\W18013189_HW4_차윤범>node demo_fs_replace.js Replaced!
Replaced!
```

```
C:\Users\W자\온\W\Desktop\W2021\1학기\W\프로그래밍\과제\W18013189_HW4_차윤범>
```

Delete Files Part

fs.unlink()

실행 코드와 결과



The screenshot shows the Visual Studio Code editor with the file `demo_fs_unlink.js` open. The code in the editor is as follows:

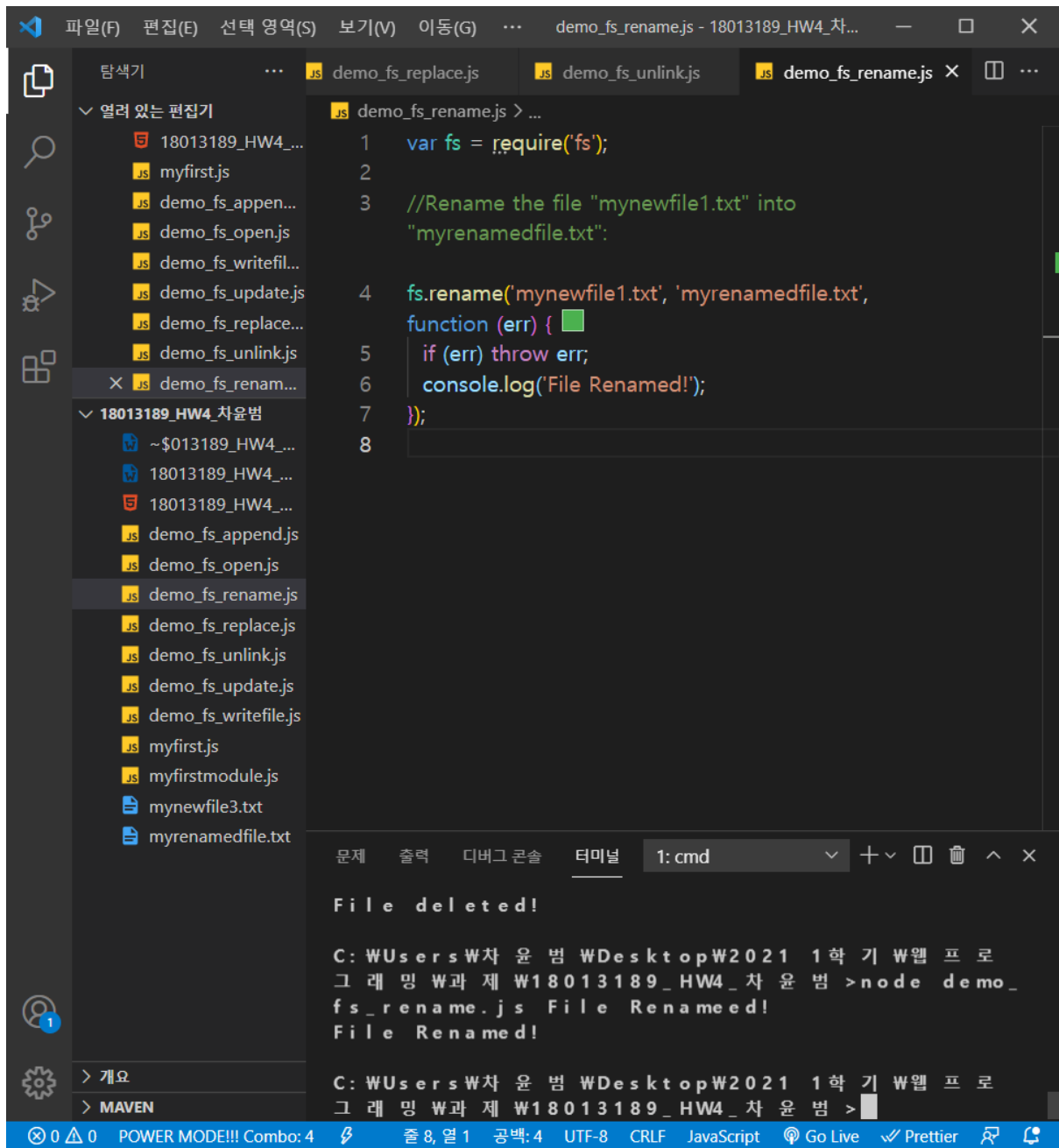
```
1 var fs = require('fs');
2
3 //Delete the file mynewfile2.txt:
4 Complexity is 3 Everything is cool!
5 fs.unlink('mynewfile2.txt', function (err) {
6   if (err) throw err;
7   console.log('File deleted!');
8 })
```

The file explorer on the left shows the project structure, including files like `18013189_HW4_차윤범`, `demo_fs_append.js`, `demo_fs_open.js`, `demo_fs_replace.js`, `demo_fs_unlink.js`, `demo_fs_update.js`, `demo_fs_writefile.js`, `myfirst.js`, `myfirstmodule.js`, `mynewfile1.txt`, and `mynewfile3.txt`.

The terminal at the bottom shows the command `node demo_fs_unlink.js` being executed, resulting in the output:

```
Replaced!
C: \Users\W차 윤 범\Desktop\W2021 1학 기 웹 프 로 그 래 밍 W과 제 W18013189_HW4_차 윤 범 >node demo_
fs_unlink.js File deleted!
File deleted!
```

fs.rename()



Node.js URL Module Chapter

Split a web address into readable parts

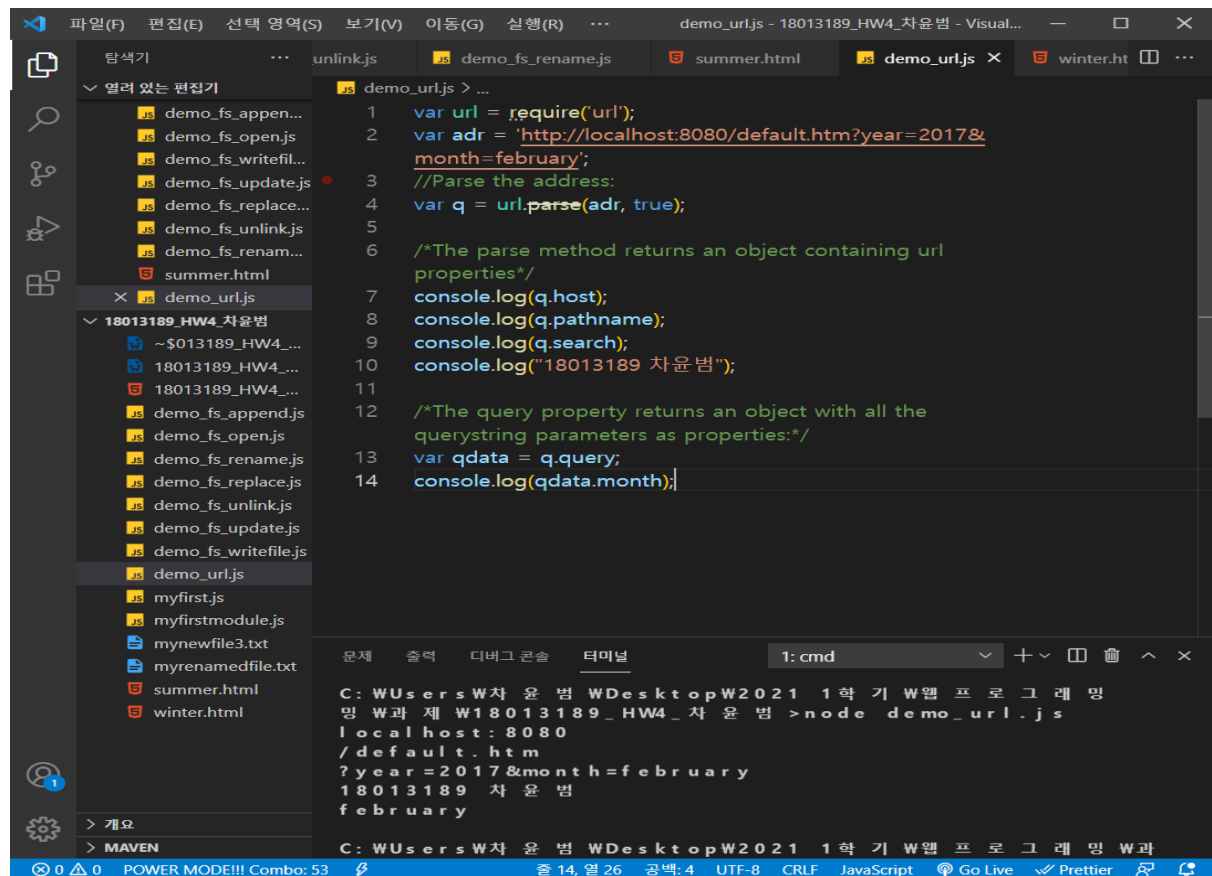
실행 코드

```
var url = require('url');
var adr = 'http://localhost:8080/default.htm?year=2017&month=february';
//Parse the address:
var q = url.parse(adr, true);

/*The parse method returns an object containing url properties*/
console.log(q.host);
console.log(q.pathname);
console.log(q.search);
console.log("18013189 차윤범");

/*The query property returns an object with all the querystring parameters
as properties:*/
var qdata = q.query;
console.log(qdata.month);
```

실행 결과



```
C: WUsersW차 윤 범 WDesktopW2021 1학 기 W웹 프 로 그 레 밍
명 W과 제 W18013189_ HW4_ 차 윤 범 >node demo_ url . js
localhost : 8080
/ default . htm
? year = 2017 & month = february
18013189 차 윤 범
february

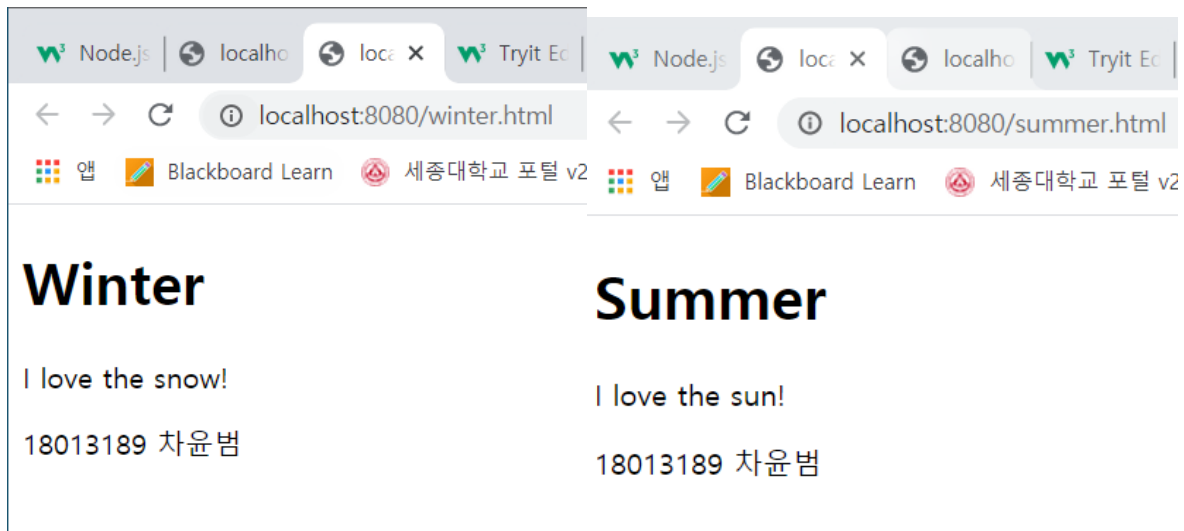
C: WUsersW차 윤 범 WDesktopW2021 1학 기 W웹 프 로 그 레 밍 W과
```

Node.js File Server Part

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

http.createServer(function (req, res) {
  var q = url.parse(req.url, true);
  var filename = "." + q.pathname;
  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': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

실행 결과

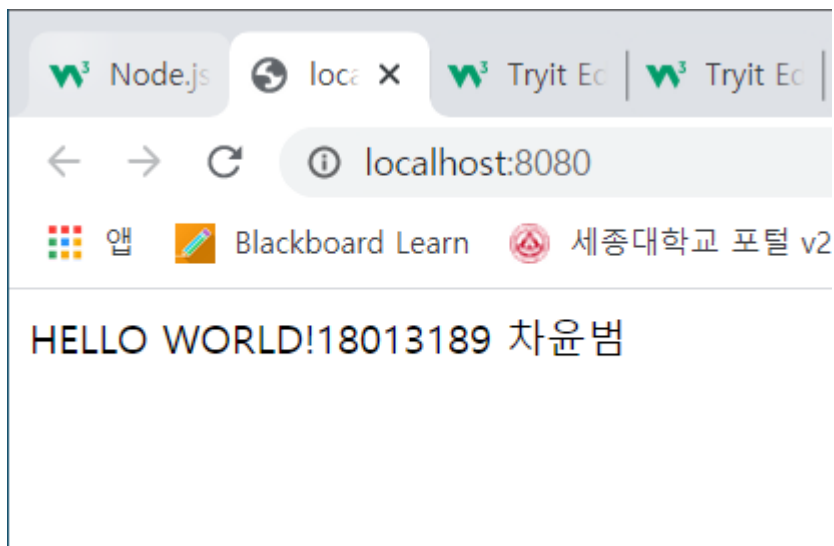


Node.js NPM Chapter

실행 코드

```
var http = require('http');
var uc = require('upper-case');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  /*Use our upper-case module to upper case a string:*/
  res.write(uc.toUpperCase("Hello World!"));
  res.write(uc.toUpperCase("18013189 차윤범"));
  res.end();
}).listen(8080);
```

실행 결과



Node.js Events

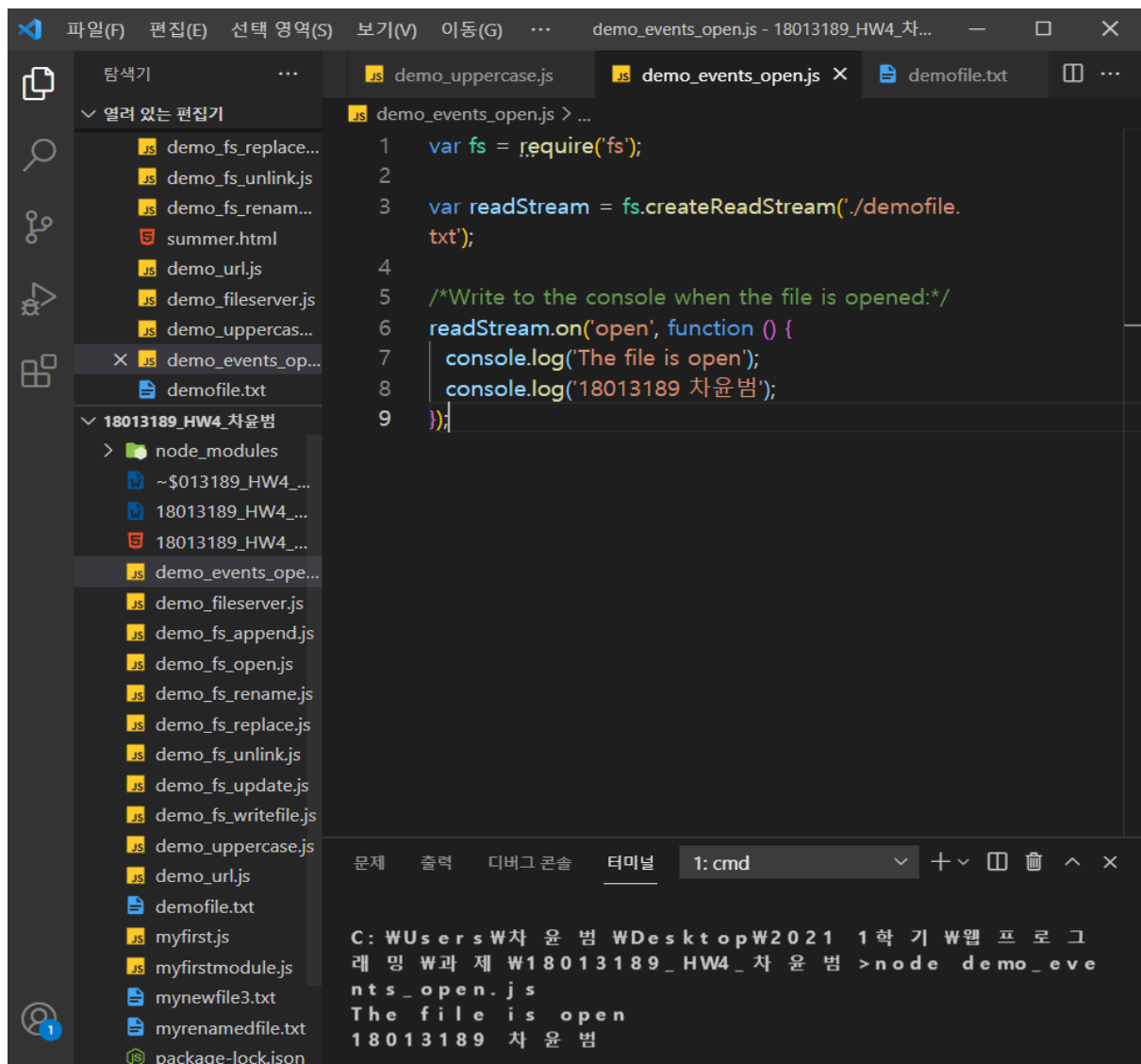
실행 코드

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

var readStream = fs.createReadStream('./demofile.txt');

/*Write to the console when the file is opened:*/
readStream.on('open', function () {
  console.log('The file is open');
  console.log('18013189 차윤범');
});
```

실행 결과



```
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) ... demo_events_open.js - 18013189_HW4_차... — □ ×
탐색기 ... JS demo_uppercase.js JS demo_events_open.js X demofile.txt □ ...
열려 있는 편집기
JS demo_fs_replace...
JS demo_fs_unlink.js
JS demo_fs_renam...
summer.html
JS demo_url.js
JS demo_fileserver.js
JS demo_uppercas...
X JS demo_events_op...
demofile.txt
18013189_HW4_차윤범
  > node_modules
    ~$013189_HW4_...
    18013189_HW4_...
    18013189_HW4_...
    JS demo_events_ope...
    JS demo_fileserver.js
    JS demo_fs_append.js
    JS demo_fs_open.js
    JS demo_fs_rename.js
    JS demo_fs_replace.js
    JS demo_fs_unlink.js
    JS demo_fs_update.js
    JS demo_fs_writefile.js
    JS demo_uppercase.js
    JS demo_url.js
    demofile.txt
    myfirst.js
    myfirstmodule.js
    mynewfile3.txt
    myrenamedfile.txt
    package-lock.json
demo_events_open.js > ...
1 var fs = require('fs');
2
3 var readStream = fs.createReadStream('./demofile.
  txt');
4
5 /*Write to the console when the file is opened:*/
6 readStream.on('open', function () {
7   console.log('The file is open');
8   console.log('18013189 차윤범');
9 });
문제 출력 디버그 콘솔 터미널 1: cmd
C: WUsersW차 윤 범 WDesktopW2021 1학 기 W웹 프 로 그
래 밍 W과 제 W18013189_HW4_차 윤 범 >node demo_eve
nts _ open . js
The file is open
18013189 차 윤 범
```

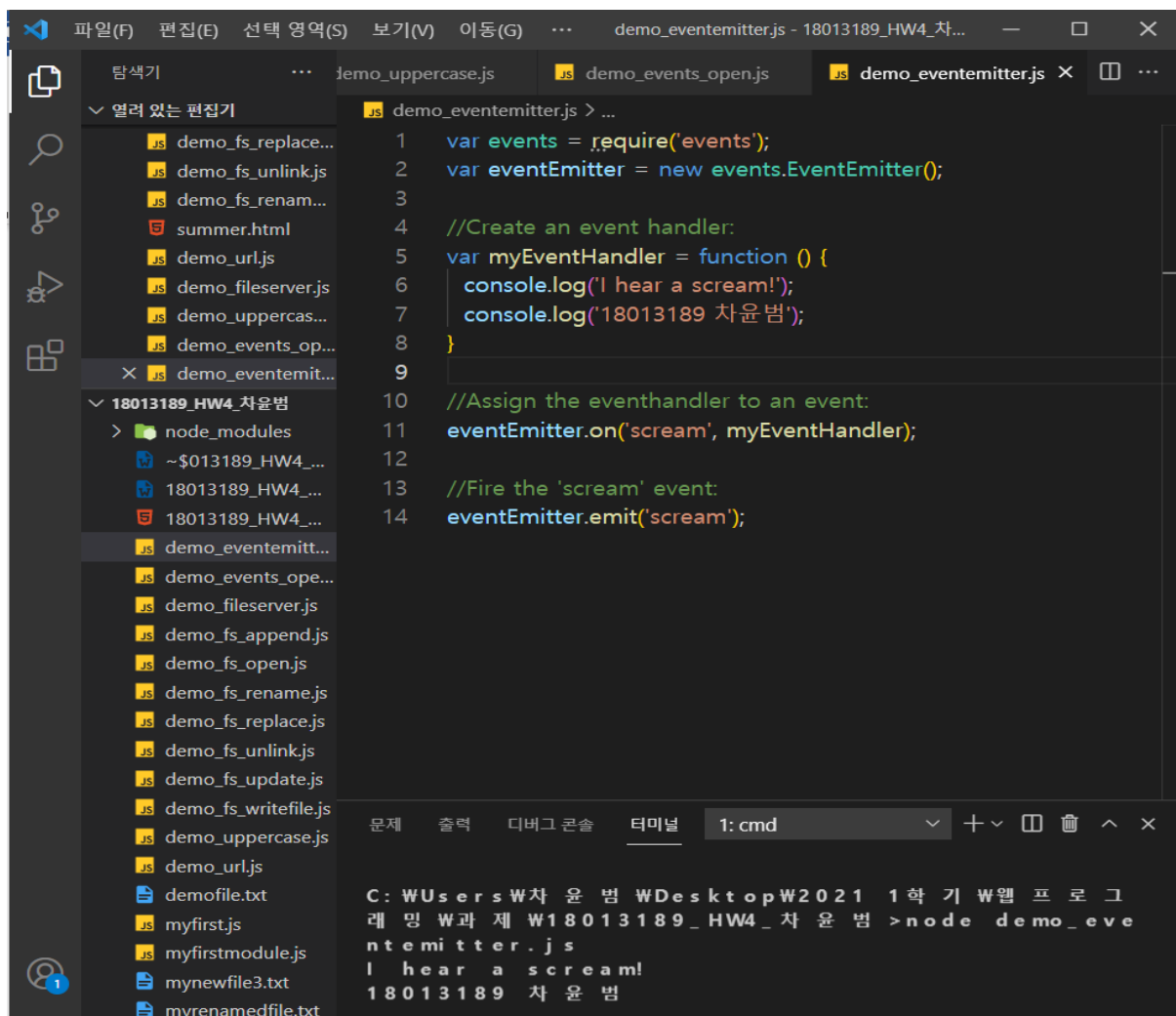
The EventEmitter Object Part

실행 코드

```
var events = require('events');
var EventEmitter = new events.EventEmitter();
//Create an event handler:
var myEventHandler = function () {
  console.log('I hear a scream!');
  console.log('18013189 차윤범');
}
//Assign the eventhandler to an event:
eventEmitter.on('scream', myEventHandler);

//Fire the 'scream' event:
eventEmitter.emit('scream');
```

실행 결과



```
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) ... demo_eventemitter.js - 18013189_HW4 차... — □ ×
탐색기 ... demo_uppercase.js demo_events_open.js demo_eventemitter.js × □ ...
열려 있는 편집기
demo_eventemitter.js > ...
1 var events = require('events');
2 var EventEmitter = new events.EventEmitter();
3
4 //Create an event handler:
5 var myEventHandler = function () {
6   console.log('I hear a scream!');
7   console.log('18013189 차윤범');
8 }
9
10 //Assign the eventhandler to an event:
11 eventEmitter.on('scream', myEventHandler);
12
13 //Fire the 'scream' event:
14 eventEmitter.emit('scream');
```

문제 출력 디버그 콘솔 터미널 1: cmd

```
C: \Users\W차 윤 범\Desktop\2021 1 학 기 웹 프 로 그
래 밍 과 제 W18013189_HW4_차 윤 범>node demo_eve
ntemi tter . j s
I h e a r a s c r e a m !
1 8 0 1 3 1 8 9 차 윤 범
```

Node.js Upload Files Chapter

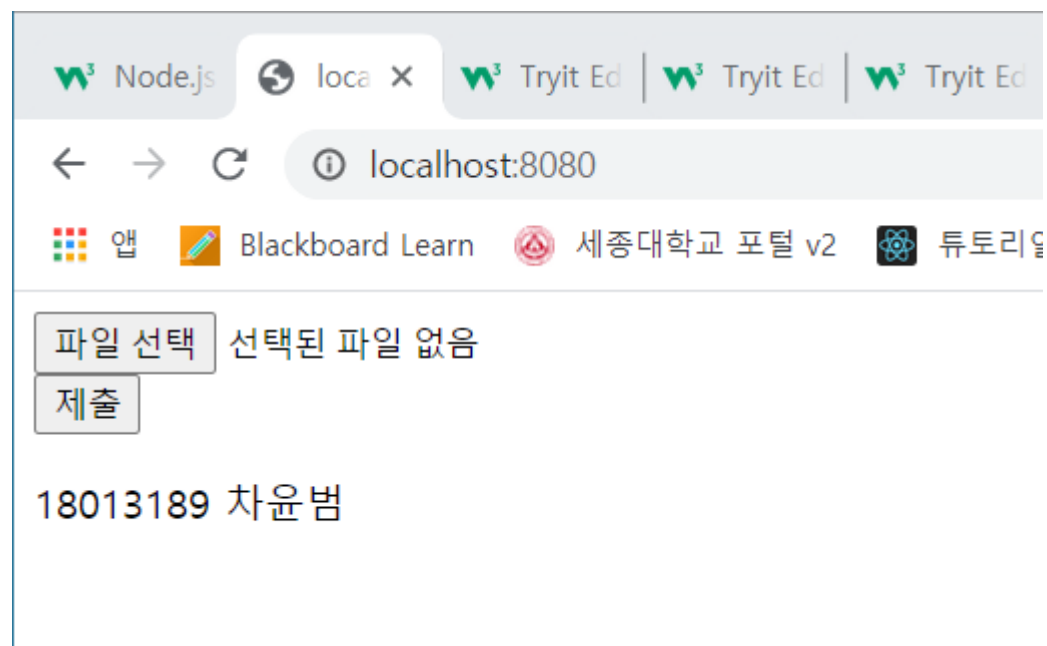
Step1

실행 코드

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
  res.write('<input type="file" name="filetoupload"><br>');
  res.write('<input type="submit">');
  res.write('</form>');
  res.write("18013189 차윤범");
  return res.end();
}).listen(8080);
```

실행 결과



Step2 and Step3

Step2 실행 코드

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

http.createServer(function (req, res) {
  if (req.url == '/fileupload') {
    var form = new formidable.IncomingForm();
    form.parse(req, function (err, fields, files) {
      res.write('File uploaded');
      res.end();
    });
  } else {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
    res.write('<input type="file" name="filetoupload"><br>');
    res.write('<input type="submit">');
    res.write('</form>');
    res.write("18013189 차윤범");
    return res.end();
  }
}).listen(8080);
```

Step3 실행 코드

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

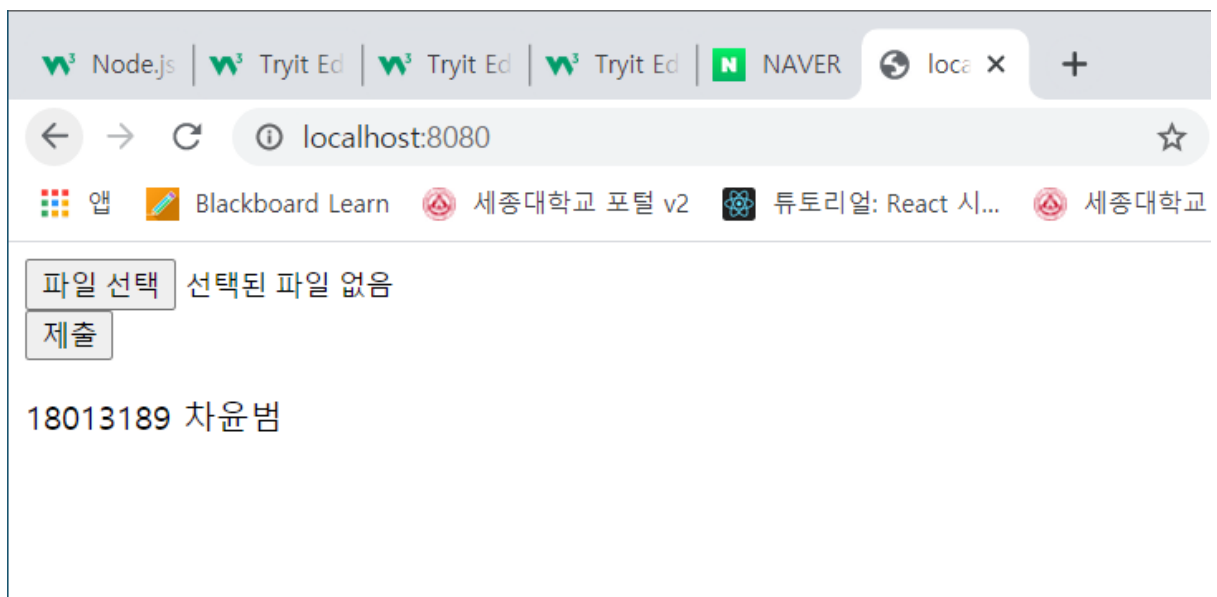
http.createServer(function (req, res) {
  if (req.url == '/fileupload') {
    var form = new formidable.IncomingForm();
    form.parse(req, function (err, fields, files) {
      var oldpath = files.filetoupload.path;
      var newpath = 'C:/Users/Your Name/' + files.filetoupload.name;
      fs.rename(oldpath, newpath, function (err) {
        if (err) throw err;
        res.write('File uploaded and moved!');
        res.end();
      });
    });
  }
});
```

```

    });
  });
  } else {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
    res.write('<input type="file" name="filetoupload"><br>');
    res.write('<input type="submit">');
    res.write('</form>');
    res.write('18013189 차윤범');
    return res.end();
  }
}).listen(8080);

```

실행 결과는 Step 1과 동일하다.



Node.js Send an Email Chapter

`npm install nodemailer` 명령어로 모듈을 설치합니다.

선택한 이메일 제공 업체의 사용자 이름과 비밀번호를 사용하여 이메일을 보냅니다. 이 튜토리얼은 Gmail 계정을 사용하여 이메일을 보내는 방법을 보여줍니다.

실행 코드

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

var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'youremail@gmail.com',
    pass: 'yourpassword'
  }
});

var mailOptions = {
  from: 'youremail@gmail.com',
  to: 'myfriend@yahoo.com',
  subject: 'Sending Email using Node.js',
  text: 'That was easy!'
};

transporter.sendMail(mailOptions, function(error, info){
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
  }
});
```

둘 이상의 주소로 이메일을 보내려면

```
var mailOptions = {
  from: 'youremail@gmail.com',
  to: 'myfriend@yahoo.com, myotherfriend@yahoo.com',
  subject: 'Sending Email using Node.js',
  text: 'That was easy!'
}
```

var mailOptions의 변수를 위와 같이 변경해주면 됩니다.

HTML 형식의 텍스트를 이메일로 보내려면 "text"속성 대신 "html"속성을 사용해야 합니다.

```
var mailOptions = {  
  from: 'youremail@gmail.com',  
  to: 'myfriend@yahoo.com',  
  subject: 'Sending Email using Node.js',  
  html: '<h1>Welcome</h1><p>That was easy!</p>'  
}
```

이렇게 마찬가지로 변수를 위와 같이 변경하면 됩니다.