**웹프로그래밍 HW4**

**18013189 컴퓨터공학과 차윤범**

**Node.js Tutorial Chapter**

실행 코드와 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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'});

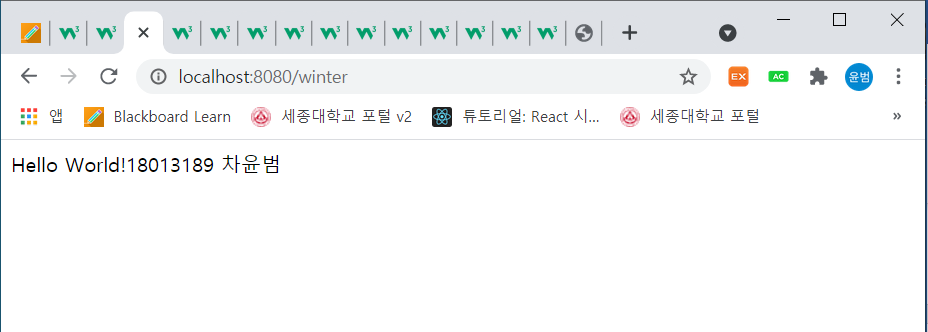
  res.write(req.url);

  res.write("18013189 차윤범");

  res.end();

}).listen(8080);

실행 결과



텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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'});

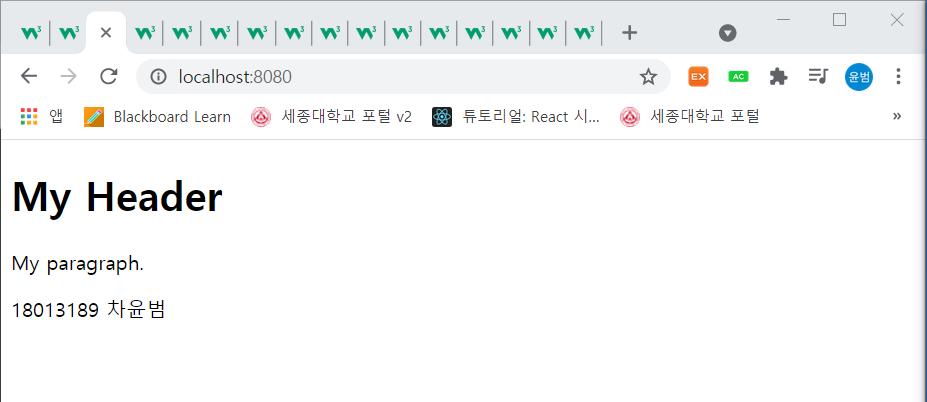
    res.write(data);

    return res.end();

  });

}).listen(8080);

실행결과



**Create Files Part**

**fs.appendFile()**

실행 코드와 실행 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**fs.open()**

실행 코드와 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**fs.writeFile()**

실행 코드와 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**Update Files Part**

**fs.appendFile()**

실행 코드와 결과

텍스트, 스크린샷, 모니터, 검은색이(가) 표시된 사진

자동 생성된 설명

**fs.writeFile()**

실행 코드와 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**Delete Files Part**

**fs.unlink()**

실행 코드와 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**fs.rename()**

텍스트이(가) 표시된 사진

자동 생성된 설명

**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);

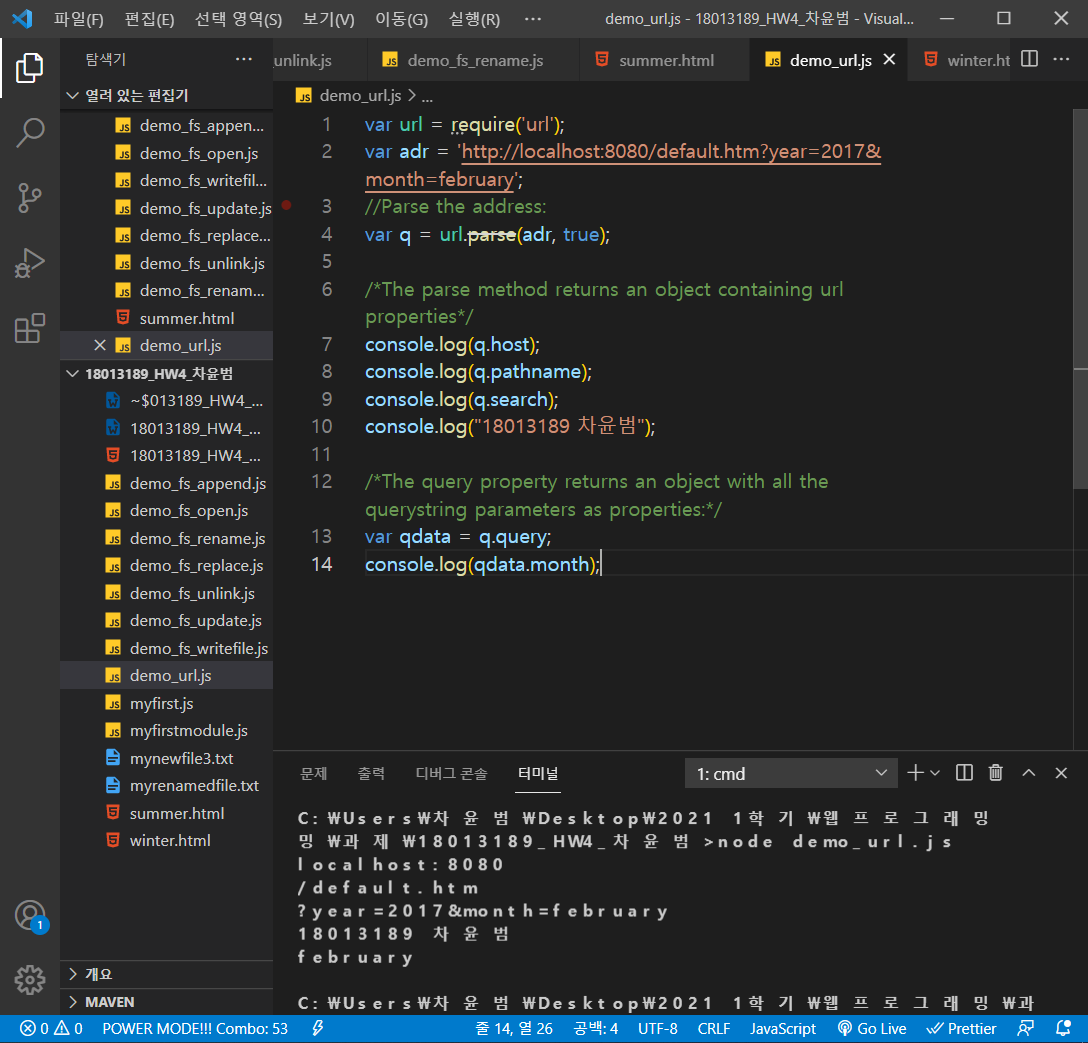
console.log("18013189 차윤범");

/\*The query property returns an object with all the querystring parameters as properties:\*/

var qdata = q.query;

console.log(qdata.month);

실행 결과



**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'});

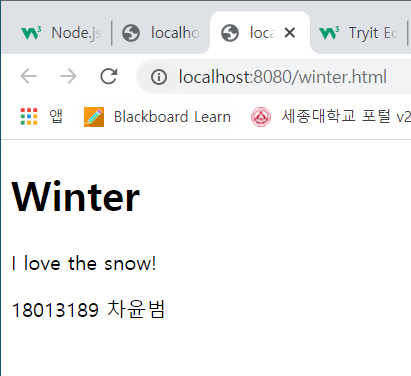
    res.write(data);

    return res.end();

  });

}).listen(8080);

실행 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**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.upperCase("Hello World!"));

  res.write(uc.upperCase("18013189 차윤범"));

  res.end();

}).listen(8080);

실행 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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

텍스트이(가) 표시된 사진

자동 생성된 설명

**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.

텍스트이(가) 표시된 사진

자동 생성된 설명

**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.