12차시

- cmd
 	
notepad notehttp.js
 	
- notehttp 메모장
 	
var http = require('http'); // http 모듈을 불러온다.
http.createServer(function (req, res) { // 웹 브라우저를 만들고
res.writeHead(200, {'Content-Type': 'text/html'}); // text나 html를 회신할 것이다.
res.end('Hello World!'); // 출력하고자 하는 문장
}).listen(8080); // 포트 넘버를 적는다. 도메인주소에 쓰는 숫자
 	
- cmd
 	
node notehttp.js
	실습 1번
	HTTP 모듈 사용해보기

- cmd
 	
notepad mymodule.js
 	
- mymodule 메모장
 	
exports.myDateTime = function () { // myDateTime이름으로 사용자 모듈을 만든다.
return Date(); // 현재 날짜를 반환한다.
};
 	
- cmd
 	
notepad notehttp.js
 	
- notehttp 메모장
 	
var http = require('http'); // http 모듈을 불러온다.
var dt = require('./mymodule'); // 사용자 지정 모듈을 불러온다.
 	
http.createServer(function (req, res) { // 웹 브라우저를 만들고
res.writeHead(200, {'Content-Type': 'text/html'}); // text나 html를 회신할 것이다.
res.write("Date: " + dt.myDateTime()); // 모듈이름.사용하고자 하는 함수이름
res.end('Hello World!'); // 출력하고자 하는 문장
}).listen(8080); // 포트 넘버를 적는다. 도메인주소에 쓰는 숫자
 	
- cmd
 	
node notehttp.js
	실습 2번
	사용자 모듈을 만들어서 시간 출력하기

- cmd
 	
notepad nodeurl.js
 	
- nodeurl 메모장
 	
var url = require('url'); // url 모듈을 가져온다.
var adr = 'http://localhost:8080/default.htm?year=2023&month=May';
var q = url.parse(adr, true); // adr에 들어있는 도메인주소를 parse를 통해 잘라서 저장한다.
 	
console.log(q.host); // returns 'localhost:8080'
console.log(q.pathname); // returns '/default.htm'
console.log(q.search); // returns '?year=2023&month=May'
console.log(q.query); // [Object: null prototype] { year: '2023', month: 'May' }
	실습 3
	url 주소의 속성을 파싱으로 객체를 분리

```
var qdata = q.query;
console.log(qdata.month);  //  May
console.log(qdata.year);  //  2023
```

- cmd

```
node nodeurl.js
```

- cmd

```
notepad hturl.js
```

- hturl 메모장

```
var http = require('http');  // http모듈을 가져온다.
var url = require('url');   // url모듈을 가져온다.

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

- cmd

```
node hturl.js
```

- 도메인 주소창
http://localhost:8080/?year=2023&month=May&name=OJS

실습 4
url 쿼리 문자열 분할

####### 오류가 남 ##
- cmd

```
notepad demo.html
```

- demo HTML

```
<html>
  <body>
    <h1>My Header</h1>  // 제목
    <p>My paragraph.</p>
  </body>
</html>
```

- cmd

```
notepad nodefs.js
```

- nodefs 메모장

```
var http = require('http');
var fs = require('fs');  // 파일 시스템 모듈을 포함

http.createServer(function (req, res) {
  // fs를 사용하여 demo.html에 있는 데이터를 불러온다.
  fs.readFile('demo.html', function(err, data) {
    res.writeHead(200, {'Content- Type' : 'test/html'});
    res.write(data);  // 가져온 data를 출력
    return res.end();
  });
```

실습 5
html를 사용하여 문자영을 출력하기

}).listen(8080);

- cmd

node nodefs.js

- cmd

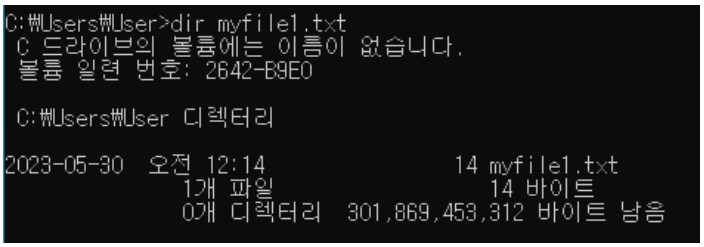
notepad myfile1.js

- myfile1 메모장

```
var fs = require('fs');
// appendFile() 메소드를 통해 Hello content!가 적혀있는 파일을 생성
fs.appendFile('myfile1.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

- cmd

node myfile1.js



실습 6
appendFile() 메소드를 통해 값이 있는 파일을 생성

- cmd

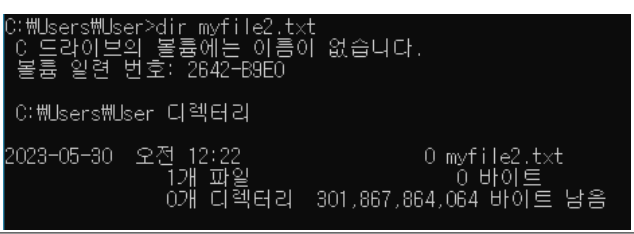
notepad myfile2.js

- myfile2 메모장

```
var fs = require('fs');
// open() 메소드를 이용하여 비어있는 새 파일을 작성
fs.open('myfile2.txt', 'w', function (err, file) {
  if (err) throw err;
  console.log('Saved!');
});
```

- cmd

node myfile2.js



실습 7
open() 메소드를 통해 비어있는 파일을 생성

- cmd

실습 8
writeFile() 메소드를 사용하여 지정된 파

notepad myfile3.js

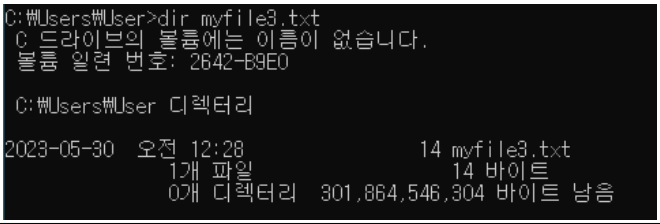
- myfile3 메모장

```
var fs = require('fs');
// writeFile() 메소드를 지정된 파일과 내용이 없으면 대체 또는 새파일 생성
fs.writeFile('myfile3.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

일과 내용이 없으면 대체 또는 새파일 생성

- cmd

node myfile3.js



- cmd

notepad myfile4.js

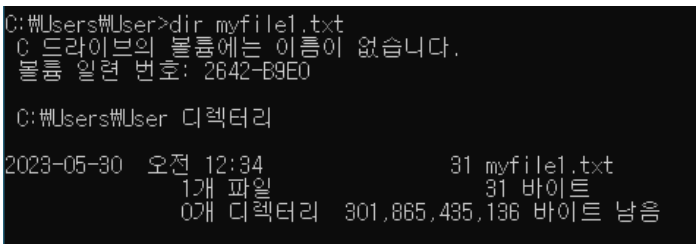
- myfile4 메모장

```
var fs = require('fs');
// appendFile() 메소드를 사용하여 업데이트를 실행한다.
fs.appendFile('myfile1.txt', ' This is my text.', function (err) {
  if (err) throw err;
  console.log('Updated!'); // cmd화면에 값을 띄운다.
});
```

실습 9
appendFile() 메소드를 사용하여 값을 추가한다.(업데이트)

- cmd

node myfile4.js



myfile1의 저장 값이 늘어났다.

- cmd

notepad myfile5.js

- myfile5 메모장

```
var fs = require('fs');
// writeFile() 메소드를 사용하여 기존의 값을 새로운 값으로 대체하여 덮어씌운다.
fs.writeFile('myfile3.txt', 'This is my text', function (err) {
  if (err) throw err;
  console.log('Replaced!');
});
```

실습 10
writeFile() 메소드를 사용하여 기존의 값을 새로운 값으로 대체하여 덮어씌운다.

- cmd

node myfile5.js

```
C:₩Users₩User>type myfile3.txt
Hello content!
C:₩Users₩User>notepad myfile5.js

C:₩Users₩User>node myfile5.js
Replaced!

C:₩Users₩User>type myfile3.txt
This is my text
C:₩Users₩User>
```

- cmd

notepad myfile6.js

- myfile6 메모장

```
var fs = require('fs');
// unlink() 메소드를 이용하여 파일을 삭제한다.
fs.unlink('myfile2.txt', function (err) {
  if (err) throw err;
  console.log('File deleted!');
});
```

- cmd

node myfile6.js

```
C:₩Users₩User>dir myfile2.txt
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 2642-B9E0

 C:₩Users₩User 디렉터리

2023-05-30  오전 12:22                 0 myfile2.txt
               1개 파일                   0 바이트
               0개 디렉터리  301,864,001,536 바이트 남음

C:₩Users₩User>node myfile6.js
File deleted!

C:₩Users₩User>dir myfile2.txt
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 2642-B9E0

 C:₩Users₩User 디렉터리

파일을 찾을 수 없습니다.

C:₩Users₩User>
```

실습 11
unlink() 메소드를 이용하여 파일을 삭제한다.

- cmd

notepad myfile7.js

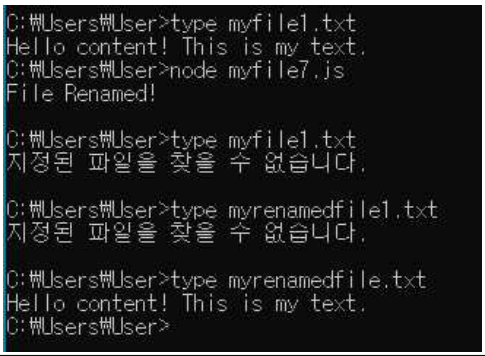
- myfile7 메모장

```
var fs = require('fs');
// rename() 메소드를 이용하여 파일 이름을 바꾼다.
fs.rename('myfile1.txt', 'myrenamedfile.txt', function (err) {
  if (err) throw err;
  console.log('File Renamed!');
});
```

- cmd

실습 12
rename() 메소드를 이용하여 파일 이름을 바꾼다.

node myfile7.js



- cmd

notepad winter.html

- winter.html

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Winter</h1>
    <p>I love the snow!</p>
  </body>
</html>
```

- cmd

notepad summer.html

- summer.html

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Summer</h1>
    <p>I love the sun!</p>
  </body>
</html>
```

- cmd

notepad urlpath.js

- urlpath 메모장

```
var http = require('http');  // http 정보를 저장
var url = require('url');  // url 정보를 저장
var fs = require('fs');  // fs 정보를 저장

http.createServer(function (req, res) {  // 도메인 창을 열었을때
  var q = url.parse(req.url, true);  // 도메인 주소창에 적힌 값을 parse로 쪼개서 저장
  var filename = "." + q.pathname;  // 쪼갠 값 중 pathname값 맨 앞에 .을 붙여서 저장
  fs.readFile(filename, function(err, data) {  // 위에서 저장한 값의 이름으로 된 파일을 불러온다.
    if (err) {  // 에러시
      res.writeHead(404, {'Content-Type': 'text/html'});
     return res.end("404 Not Found");
    }
    res.writeHead(200, {'Content-Type': 'text/html'}); // 정상적으로 값이 읽혀지면
    res.write(data);  // 파일안에 data를 출력
```

실습 12
winter과 summer html를 선언하여 도메인에 입력값에 따라 다른 값 출력하기

```
    return res.end();
  });
}).listen(8080);  // 포트 넘버
```

- cmd

```
node urlpath.js
```

- 도메인 주소창에

http://localhost:8080/summer.html
또는
http://localhost:8080/winter.html
입력

- cmd

```
notepad nodeeve.js
```

- nodeeve 메모장

```
var events = require('events');
var eventEmitter = new events.EventEmitter();

//이벤트 처리기를 생성한다.
var myEventHandler = function () {
  console.log('I hear a scream!');
}

//'scream' 이벤트에 이벤트 처리기를 할당한다.
eventEmitter.on('scream', myEventHandler);

//'scream' 이벤트를 시작합니다.
eventEmitter.emit('scream');
```

- cmd

```
node nodeeve.js
```

실습 13
// 이벤트 모듈을 사용하여 파일 실행

--- html

- cmd

```
notepad index.html
```

- index,html

```
<!doctype html>
<html lang="ko">
        <head>
                <meta charset="UTF-8">
                <meta name="Generator" content="EditPlus®">
                <title>아두이노-자바스크립트 통신 예제</title>

                <script src="https://cdn.socket.io/4.5.0/socket.io.min.js"></script>
                <style>
                        #messages {
                                background-color: blue;
                                color: black;
                                width: 600px;
                                height: 150px;
                                background-color: #f5d682;
```

실습 14
습도 센서와 LED센서를 연동시키면서 토글을 하는 코드

```
                        border: 1px solid red;
                        text-align: left;
                        padding: 10px;
                        font-size: 28px;
                }
                .data {
                        font-size: 50px;
                        color: #0099ff;
                }
                .card {
                        background-color: #F8F7F9;;
                        box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);
                        padding-top:10px;
                        padding-bottom:20px;
                }
                .button {
                        padding: 15px 50px;
                        font-size: 24px;
                        text-align: center;
                        outline: none;
                        color: #fff;
                        background-color: #0f8b8d;
                        border: none;
                        border-radius: 5px;
                        -webkit-touch-callout: none;
                        -webkit-user-select: none;
                        -khtml-user-select: none;
                        -moz-user-select: none;
                        -ms-user-select: none;
                        user-select: none;
                        -webkit-tap-highlight-color:
                        rgba(0,0,0,0);
                }
                /*.button:hover {background-color: #0f8b8d}*/
                .button:active {
                        background-color: #0f8b8d;
                        box-shadow: 2 2px #CDCDCD;
                        transform: translateY(2px);
                }
                .state {
                        font-size: 1.5rem;
                        color:#8c8c8c;
                        font-weight: bold;
                }
        </style>
</head>
<body>
        <h1>아두이노와 Node.js 웹브라우저 예제</h1>
        <div id="messages">
                <p>조도: <span id="light" class="data"></span>°C</p>
        </div>
        <form id="form" action="">
                <input id="input" autocomplete="off" /><button>Send</button>
        </form>
        <div class="content">
                <div class="card">
                        <p class="state">state: <span id="state">ledoff</span></p>
                        <p><button                                              id="button"
class="button">Toggle</button></p>
                </div>
```

```
                </div>
                <script>
                        var socket = io();
                        var temp = document.getElementById('light');
                        var form = document.getElementById('form');
                        var btn = document.getElementById('button');
                        var state = document.getElementById('state');
                        var input = document.getElementById('input');

                        form.addEventListener('submit', function(e) {
                          e.preventDefault();
                          if (input.value) {
                            socket.emit('message',  input.value);
                            input.value = '';
                          }
                        });

                        // 토글을 클릭하면
                        btn.addEventListener('click', function(e) {
                          e.preventDefault();
                          if(state.innerText == "ledoff"){ // off가 떠있는 상태에서 클릭되면
                             // 소켓으로 ledon이라는 문자를 보내고
                            socket.emit('message', "ledon");
                            state.innerText = "ledon"; // 화면에 ledon을 띄운다.
                          }
                          else {  // 아니면
                            // 소켓으로 ledoff라는 문자를 보내고
                            socket.emit('message', "ledoff");
                            state.innerText = "ledoff";  // 화면에 ledoff를 띄운다.
                          }
                        });

                        socket.on('data', (msg) => { // msg를 받으면
                          //console.log(msg);
                          const obj = JSON.parse(msg); //msg를 parse로 자른 다음에
                          console.log(obj); // 콘솔창(cmd)에 작성을 하고
                          // json의 light에 저장되어 있는 값을 화면에 띄운다.
                          temp.innerText = obj.light;
                        });
                </script>
        </body>
</html>


--- node

- cmd

notepad add2.js

- add 메모장

const { SerialPort } = require('serialport')
const { ReadlineParser } = require('@serialport/parser-readline')
const port = new SerialPort({ path: 'COM3', baudRate: 9600 })
const parser = new ReadlineParser()
port.pipe(parser)

const express = require('express');
const app = express();
```

```
const http = require('http');
const server = http.createServer(app);
const { Server } = require("socket.io");
const io = new Server(server);

app.get('/', (req, res) => {
  res.sendFile(__dirname + "/index.html"); // html 문서를 지정해 준다.
});

//소켓 연결시
io.on('connection', (socket) => {
  console.log('a user connected');
  socket.on('disconnect', () => {
    console.log('user disconnected');
  });

  //dht센서의 값을 전송하자.
  parser.on('data', function(data) {
    console.log(data);
    //소켓 발신
    socket.emit('data', data);  // HTML로 받은 값을 보낸다.
  });

   socket.on('message', (msg) => {   // HTML로 부터 값을 받고
    console.log("클라이언트의 요청이 있습니다.");
    console.log(msg);
    if(msg == 'ledon'){
      port.write("o");
    }
    if(msg == 'ledoff') {
      port.write("f");
    }

    socket.emit('result', `수신된 메세지는 "${ msg }" 입니다.`);
  });
});

server.listen(3000, () => {
  console.log("server is listening at localhost: 3000"); //'가 아니고 키보드 맨 왼쪽위 ~ 결 표시 아래
있는것임
});

--- 아두이노

- 아두이노

#include <ArduinoJson.h>

StaticJsonDocument<48> doc;
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN,OUTPUT);
}

void loop() {
  int temp = analogRead(0);
  doc["light"] = temp;
  if( temp > 500){
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
```

```
  }
  else {
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
  }

  serializeJson(doc, Serial);
  Serial.println();

  if(Serial.available() > 0) {// Read from serial port
    char ReaderFromNode; // Store current character
    ReaderFromNode = (char) Serial.read();
    convertToState(ReaderFromNode); // 사용자 지정 함수 사용
  }
  delay(2000);
}

void convertToState(char chr) {
  if(chr=='o'){
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
  }
  if(chr=='f'){
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
  }
}
```