

<div><div><div>- cmd</div><div>>>> notepad test1.js</div></div><div><div>- test1 메모장</div></div><div><div>const http = require("http"); // http를 요청하여 변수에 입력</div><div>const server = http.createServer((req, res) => { // 새로운 서버를 만듦</div><div>if (req.url === "/") { // '/'로 끝나는 주소가 들어오면 반환</div><div>res.write("<h1>Hello from nodejs</h1>");</div><div>} else { // '/'로 끝나는 주소가 아닌 다른 주소가 들어오면 반환</div><div>res.write("<h1>You have entered this url : \${req.url}</h1>");</div><div>}</div><div>res.end();</div><div>});</div><div><div>server.listen(3000, () => {</div><div>console.log("The server is listening on port 3000");</div><div>});</div><div><div>- cmd</div><div>>>> node test1.js</div></div><div><div>- 크롬 주소창</div><div>>>> http://localhost:3000/</div></div></div></div></div>	<div><div>실습 1</div><div>test1 메모장을 사용해서 http 패키지 사</div><div>용해보기</div></div>
<div><div><div>- cmd</div><div>>>> notepad test2.js</div></div><div><div>- test2 메모장</div></div><div><div>const express = require("express"); // express를 불러와서 변수에 저장</div><div>const server = express(); // 서버를 생성하기 위해</div><div>server.get("/", (req, res) => { // 기본적인 도메인을 받으면 출력</div><div>res.send("<h1>Hello, and help me, plz~~</h1>");</div><div>});</div><div>server.listen(3000, (err) => { // 3000이 아닌 값을 대입되면,</div><div>if (err) return console.log(err);</div><div>console.log("The server is listening on port 3000"); // 콘솔창에 출력</div><div>});</div><div><div>- cmd</div><div>>>> node test2.js</div></div><div><div>- 크롬 주소창</div><div>>>> http://localhost:3000/</div></div></div></div>	<div><div>실습 2</div><div>test2 메모장을 사용해서 express 패키</div><div>지 사용해보기</div></div>
<div><div><div>- cmd</div><div>>>> notepad index.html</div></div><div><div>- index.html 메모장</div></div><div><div><!DOCTYPE html></div><div><html lang="en"></div><div><head></div><div><meta charset="UTF-8" /></div><div><title>Home</title></div><div></head></div><div><body></div></div></div>	<div><div>실습 3</div><div>index.html , test3 메모장을 사용해서</div><div>express 패키지 사용해보기</div></div>

<pre><h1>This is HOME/ABOUT</h1> </body> </html> - cmd >>> notepad test3.js - test3 메모장 const express = require("express"); const server = express(); server.get("/", (req, res) => { res.sendFile(__dirname + "/index.html"); }); server.get("/about", (req, res) => { res.sendFile(__dirname + "/about.html"); }); server.listen(3000, (err) => { if (err) return console.log(err); console.log("The server is listening on port 3000"); }); - cmd >>> node test3.js</pre>	
<pre>- 아두이누 void setup() { Serial.begin(9600); // pinMode(LED_BUILTIN, OUTPUT); // 내장 LED를 사용한다. digitalWrite(LED_BUILTIN, LOW); // 디폴트 값으로 불을 끈다. } void loop() { if(Serial.available() > 0) // 도메인에 입력된 값이 시리얼 입력값으로 들어왔을 경우 { char ReaderFromNode; ReaderFromNode = (char) Serial.read(); // 시리얼로 입력된 값을 불러서 변수에 저장한다. convertToState(ReaderFromNode); // 사용자지정함수를 불러와서 LED를 조절한다. } delay(1000); } void convertToState(char chr) { if(chr=='o'){ // 매개변수로 들어온 값이 o인 경우 LED - ON digitalWrite(LED_BUILTIN, HIGH); delay(100); } if(chr=='f'){ // 매개변수로 들어온 값이 f인 경우 LED - OFF digitalWrite(LED_BUILTIN, LOW); delay(100); } } - cmd >>> notepad test4.js - test4 메모장</pre>	<p>실습 4 도메인 값을 입력하면, 그 값에 따라 보드의 LED출력하기</p>

```

const express = require('express');
const app = express();
const { SerialPort } = require('serialport') ;
const { ReadlineParser } = require('@serialport/parser-readline') ;
const arduinoSerialPort = new SerialPort({ path: 'COM3', baudRate: 9600 });
// 연결 포트와 시리얼 출력속도를 입력한다.
var port = 3000;

arduinoSerialPort.on('open',function() { // 아두이노와 연결이 잘 되었으면,
    console.log('Serial Port ' + arduinoSerialPort + ' is opened.');// cmd프롬프트 창에 출
    력
});

app.get('/', function (req, res) { // 도메인 창이 준비 되었으면 출력
    return res.send('Working');
})

app.get('/:action', function (req, res) {
    var action = req.params.action || req.param('action');// 도메인 값 뒤에 액션이 들어오
    면,
    if(action == 'led'){ // /led 라고 입력했을때
        arduinoSerialPort.write("o");// 아두이노 시리얼에 o 라고 값을 입력
        return res.send('Led light is on!');
    }
    if(action == 'off') { // /off 라고 입력했을때
        arduinoSerialPort.write("f");// 아두이노 시리얼에 f 라고 값을 입력
        return res.send("Led light is off!");
    }
    return res.send('Action: ' + action);
});

app.listen(port, function () {
    // cmd프롬프트 창에 출력
    console.log('Example app listening on port http://0.0.0.0:' + port + '!');
});

- cmd

>>> node test4.js

- 도메인 창에 led, off를 입력한다.

```

```

- 아두이노 창
코드를 실행하기 전에 라이브러리 매니저에서 DHT sensor library를 설치한다.

#include <ArduinoJson.h>
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11

// 2번핀에 연결되어있는 온도습도측정기를 DHT11모드로 값을 받아서 dht에 저장한다.
DHT dht(DHTPIN, DHTTYPE);

StaticJsonDocument<48> doc; // Json 값으로 변수를 저장하기 위해 선언

float hum; //습도 값을 저장할 변수
float temp; //온도 값을 저장할 변수

void setup() {
    Serial.begin(9600);
    pinMode(LED_BUILTIN, OUTPUT); // 13번 LED 출력모드

```

실습 5, 6
 습도 측정을 보드를 통해, 측정을 하여
 크롬 창에 출력하고, 크롬 창 입력란에
 값을 입력하면 그 값에 따라 led를 출력
 한다.
 led - 13(노), GND(검)
 온도습도계 - 2(노), 5v(빨), GND(검)

```
digitalWrite(LED_BUILTIN, LOW); // 디폴트 값으로 불을 끈다.
dht.begin();
}

void loop() {
  hum = dht.readHumidity(); // dht에 저장된 값 중 Humidity를 읽어서 저장한다.
  temp = dht.readTemperature(); // dht에 저장된 값 중 Temperature를 읽어서 저장한다.

  doc["temp"] = temp; // Json 변수인 doc에 값을 저장한다.
  doc["humidity"] = hum; // Json 변수인 doc에 값을 저장한다.

  serializeJson(doc, Serial); // doc를 serial를 통해 보낸다.
  Serial.println();
  if(Serial.available() > 0) // Read from serial port
  {
    char ReaderFromNode; // Store current character
    ReaderFromNode = (char) Serial.read();
    convertToState(ReaderFromNode); // 사용자 지정함수를 불러온다.
  }
  delay(3000);
}

void convertToState(char chr) { // 매개변수를 통해 들어온 값이
  if(chr=='o'){ // o면 led on
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
  }
  if(chr=='f'){ // f면 led off
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
  }
}
```

- cmd

>>> notepad index.html

- index.html 메모장

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="Generator" content="EditPlus@">
  <meta name="Author" content="">
  <meta name="Keywords" content="">
  <meta name="Description" content="">
  <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: 300px;
      background-color: #f5d682;
      border: 1px solid red;
      text-align: left;
      padding: 10px;
```

```

        font-size: 28px;
    }

    .data {
        font-size: 50px;
        color: #0099ff;
    }
</style>

</head>
<body>
    <h1>아두이노와 Node.js 웹브라우저 시리얼, 소켓 통신 예제</h1>

    <div id="messages">
        <p>온도: <span id="temp" class="data"></span>°C</p>
        <p>습도: <span id="hum" class="data"></span>%</p>
    </div>

    <form id="form" action="">
        <input id="input" autocomplete="off" /><button>Send</button>
    </form>

    <script>
        var socket = io();

        var temp = document.getElementById('temp');
        var hum = document.getElementById('hum');
        var form = document.getElementById('form');
        var input = document.getElementById('input');
        form.addEventListener('submit', function(e) {
            e.preventDefault();
            if (input.value) {
                socket.emit('message', input.value); // 보내고
                input.value = '';
                console.log("sending OK");
            }
        });

        socket.on('data', (msg) => { // 받고
            console.log(msg);
            const obj = JSON.parse(msg);
            console.log(obj);
            temp.innerText = obj.temp;
            hum.innerText = obj.humidity;
        });

    </script>
</body>
</html>

- cmd

>>> notepad add2.js

- add2 메모장

const { SerialPort } = require('serialport')
const { ReadlineParser } = require('@serialport/parser-readline')
const arduinport = new SerialPort({ path: 'COM3', baudRate: 9600 })
// 아두이노포트와 연결속도를 입력하여 아두이노와 연결한다.
const parser = new ReadlineParser()

```

arduinoport.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'); // 연결시 cmd프롬프트에 출력
  socket.on('disconnect', () => { // 연결이 안될 시 cmd프롬프트에 출력
    console.log('user disconnected');
  });

  //dht센서의 값을 전송하자.
  parser.on('data', function(data) {
    console.log(data); //data를 cmd프롬프트에 출력

    //소켓 발신
    socket.emit('data', data); // 소켓에 data를 보낸다.
  });

  socket.on('message', (msg) => { //소켓으로 메시지를 받고
    // cmd프롬프트에 출력
    console.log("클라이언트의 요청이 있습니다.");
    // cmd프롬프트에 받은 메시지 출력
    console.log(msg); // cmd프롬프트에 받은 메시지 출력
    // 받은 메시지가 ledon이면 아두이노에 o를 보낸다.
    if(msg == "ledon") arduinoport.write("o");
    // 받은 메시지가 ledoff이면 아두이노에 f를 보낸다.
    if(msg == "ledoff") arduinoport.write("f");
    // 소켓에 다음의 값을 보낸다.
    socket.emit('result', '수신된 메세지는 "${ msg }" 입니다.');
```

});

```
server.listen(3000, () => {
  console.log("server is listening at localhost: 3000");
});

- cmd

>>> node add2.js

- 크롬 소켓 창에 ledon, ledoff를 입력해본다.
```