



Arduino-IoT

[wk02]

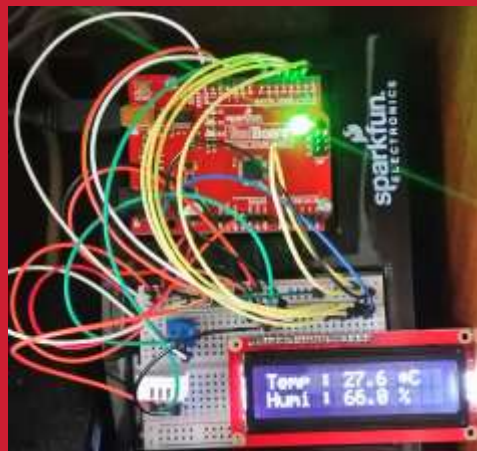
node.js intro

Visualization of Signals using Arduino,
Node.js & storing signals in MongoDB
& mining data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2020

Email : chaos21c@gmail.com





My ID

1분반-목요일 (2학년)

- AA1-01: 강서현
- AA1-02: 강태민
- AA1-03: 김세은
- AA1-04: 여수민
- AA1-05: 정영훈
- AA1-06: 차혁준
- AA1-07: 하태현
- AA1-08: 김경욱
- AA1-09: 김민욱
- AA1-10: 김민성
- AA1-11: 김민준
- AA1-12: 김인수
- AA1-13: 김현식
- AA1-14: 장성운
- AA1-15: 전승진
- AA1-16: 정희철
- AA1-17: 조동현
- AA1-18: 전동빈
- AA1-19: 신종원

2분반-수요일 (3학년)

- AA2-01: 강민수
- AA2-02: 구병준
- AA2-03: 김종민
- AA2-04: 박성철
- AA2-05: 이승현
- AA2-06: 이창호
- AA2-07: 손성빈
- AA2-08: 안예찬
- AA2-09: 유종인
- AA2-10: 이석민
- AA2-11: 이정문
- AA2-12: 이주원
- AA2-13: 정재영
- AA2-14: 하태성
- AA2-15: 김경미
- AA2-16: 김규년
- AA2-17: 김유빈
- AA2-18: 송다은
- AA2-19: 정주은
- AA2-20: 권준표



[Review]

◆ [wk01]

➤ test : aann.html

wk01 : aann.html

- ◆ [Target of this week] Make **aann.html** using Sublime text3



```

D:\Portable\Sublime Text Build 3143 x64\Data\aa00\aaann.html (aa00) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Settings Tools Project Preferences Help
FOLDERS
  aa00
    aann.html
1  <html>
2      <head>
3
4      </head>
5
6      <body>
7          <h2>Hello aa00!</h2>
8      </body>
9
10 </html>
  
```

wk01 : aann.html – result!

D:\Portable\Sublime Text Build 3143 x64\Data\aa00\aaann.html (aa00) - Sublime Text (UNREGISTERED)

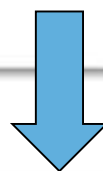
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- aa00
 - aaann.html

```

1  <html>
2      <head>
3
4      </head>
5
6      <body>
7          <h2>Hello aa00!</h2>
8      </body>
9
10 </html>
  
```





1.0 What is node.js?





1.0 What is node.js?

← → ↻ 🏠 🔒 안전함 | https://www.w3schools.com/nodejs/nodejs_intro.asp

🏠 HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP HOW TO JQUERY MORE ▼

Node.js Tutorial
Node.js HOME
Node.js Intro
Node.js Get Started
Node.js Modules
Node.js HTTP Module
Node.js File System
Node.js URL Module
Node.js NPM
Node.js Events
Node.js Upload Files
Node.js Email

Node.js MySQL
MySQL Get Started
MySQL Create Database
MySQL Create Table

Node.js Introduction

◀ Previous

What is Node.js?

- Node.js is an open source server framework
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

Why Node.js?

Node.js uses asynchronous programming!

https://www.w3schools.com/nodejs/nodejs_intro.asp



1.0 What is node.js?

Javascript on Server

Node.js is an open-source, runtime environment for developing **server-side and network applications in JavaScript**.

Its **event-driven architecture and non-blocking I/O model** makes it ideal for building **real-time applications that run across distributed devices**.



1.1 What is node.js?



홈 | ABOUT | 다운로드 | 문서 | 재단 | 참여하기 | 보안 | 뉴스

Node.js®에 대해서

비동기 이벤트 주도 JavaScript 런타임으로써 Node는 확장성 있는 네트워크 애플리케이션을 만들 수 있도록 설계되었습니다. 다음 "hello world" 예제는 다수의 연결을 동시에 처리할 수 있습니다. 각 연결에서 콜백이 실행되는데 실행할 작업이 없다면 Node는 대기합니다.

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

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```



1.2 What is node.js?

- **Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.**
- **Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.**
- **Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.**



1.3 Non-blocking IO

Blocking Code

```
var contents = readFile('/path/some/file.txt');  
console.log(contents);  
console.log('Another independent operation');
```

Non-Blocking Code

```
var contents = readFile('/path/some/file.txt', function(err, contents){  
    console.log(err || contents);  
});  
  
console.log('Another independent operation');
```

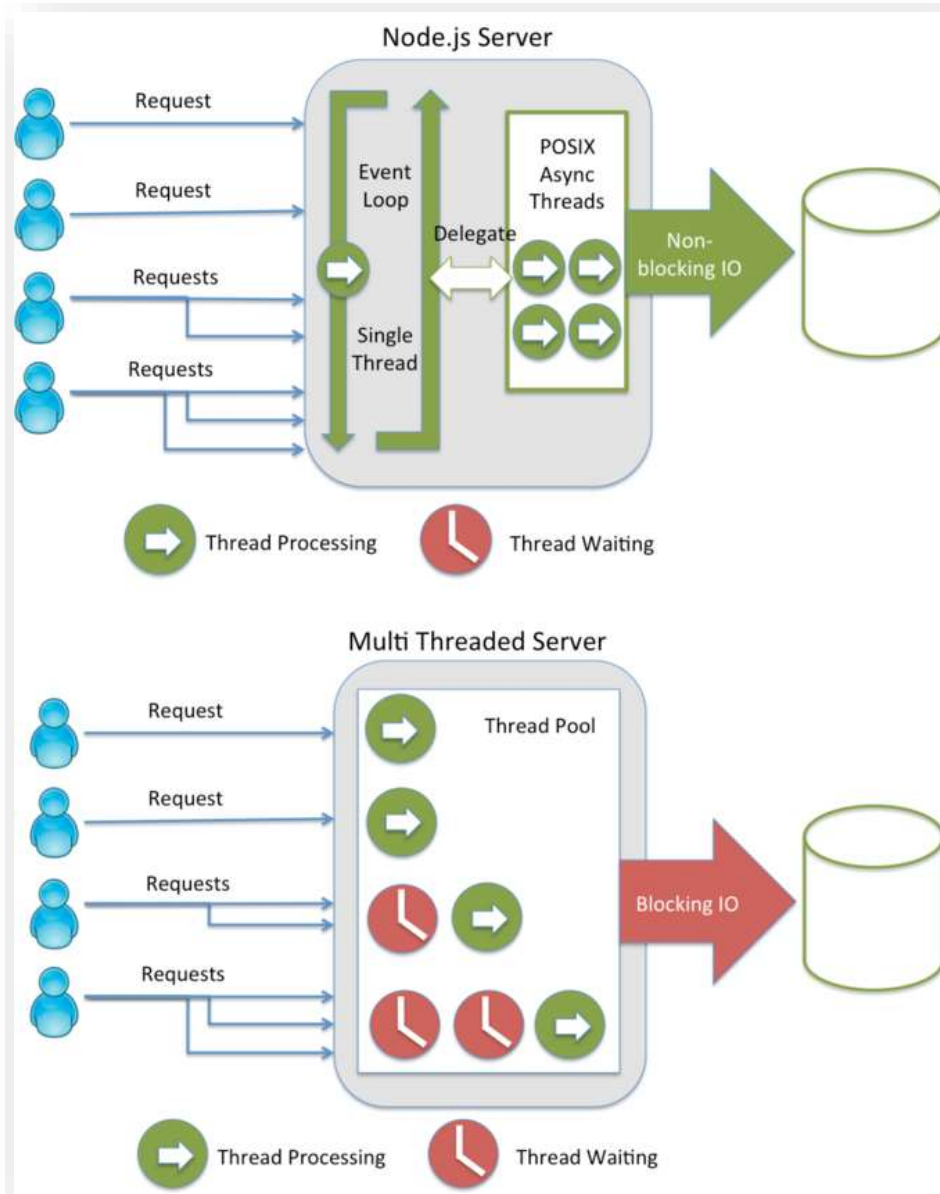
[Quiz] 계란 1개를 삶을 때 10분이 걸린다.

그러면 계란 10개를 삶을 때 걸리는 시간은?

Blocking : ?

Non-blocking : ?

1.3 Non-blocking IO

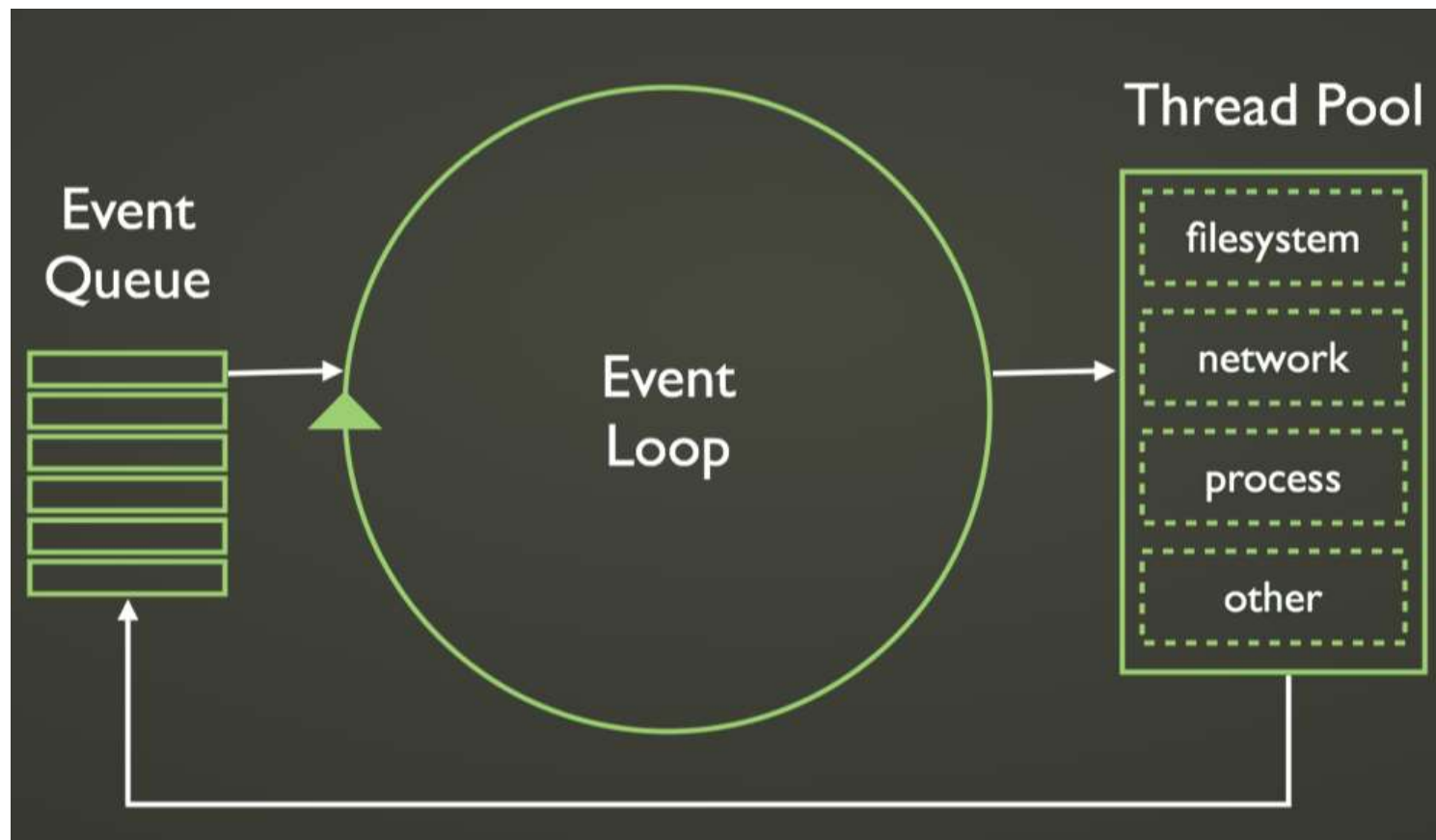


Single thread
Non-blocking IO

Multi thread
Blocking IO



1.4 Non-blocking event loop





2.1 Install node.js



다운로드

최신 LTS 버전: 12.18.3 (includes npm 6.14.6)

플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.



Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8

<https://nodejs.org/ko/download/>



2.2 Verify installation

C:\> 명령 프롬프트

```
Microsoft Windows [Version 10.0.19041.450]  
(c) 2020 Microsoft Corporation. All rights reserved.
```

```
C:\Users\life21c>node -v  
v14.9.0
```

```
C:\Users\life21c>npm -v  
6.14.8
```

```
C:\Users\life21c>_
```

node -v

npm -v



2.3 Testing node.js – node shell

명령 프롬프트

```
C:\Users\life21c>node
Welcome to Node.js v14.9.0.
Type ".help" for more information.
> a=2
2
> b=5
5
> c=a+b
7
> a*b
10
> a/b
0.4
> a%b
2
> b%a
1
>
(To exit, press ^C again or ^D or type .exit)
>
C:\Users\life21c>_
```

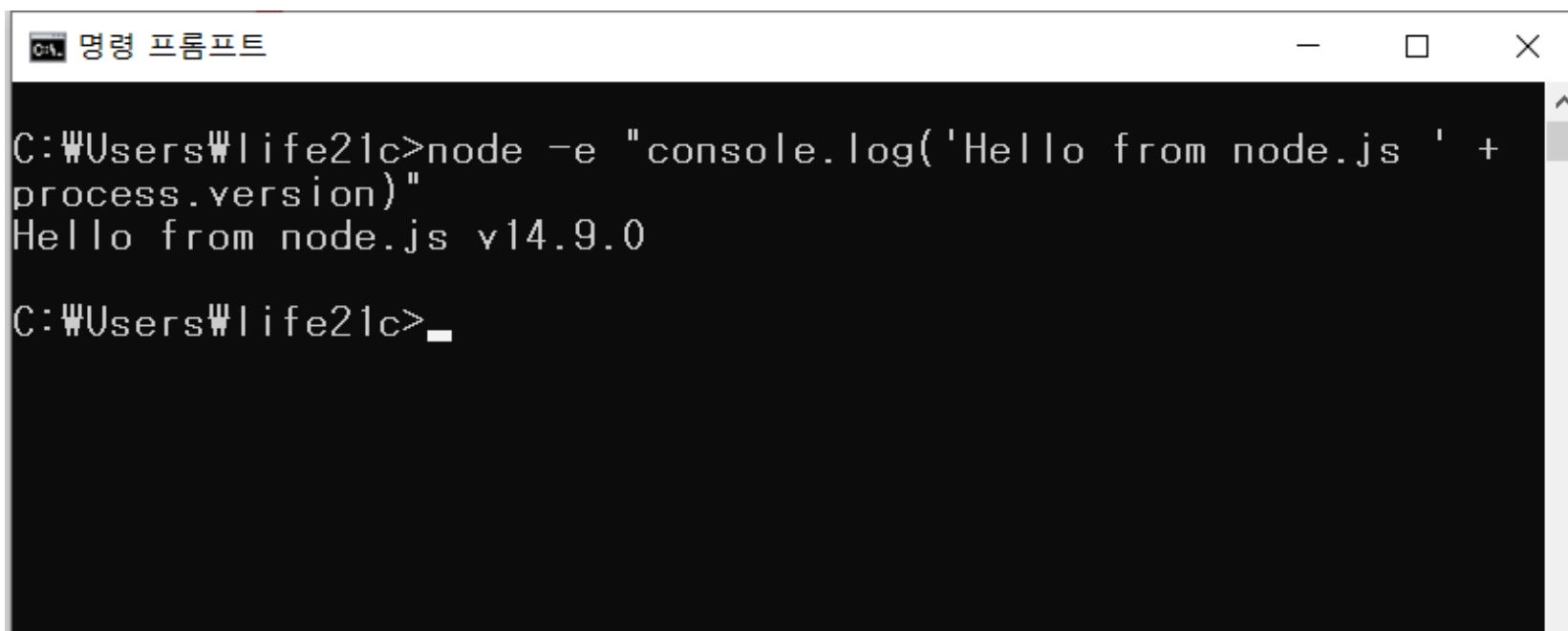
// node shell
node
>
>
// exit
^C^C



2.4 Testing node.js – final test

Final test if Node.js was installed correctly:

```
node -e "console.log('Hello from node.js ' + process.version)"
```



A screenshot of a Windows Command Prompt window. The title bar shows "명령 프롬프트" (Command Prompt) and the path "C:\>". The command prompt shows the following text:

```
C:\>node -e "console.log('Hello from node.js ' + process.version)"
Hello from node.js v14.9.0
C:\>_
```



Using node.js in VSCode

→ **Portable VSCode**

New editor: VScode



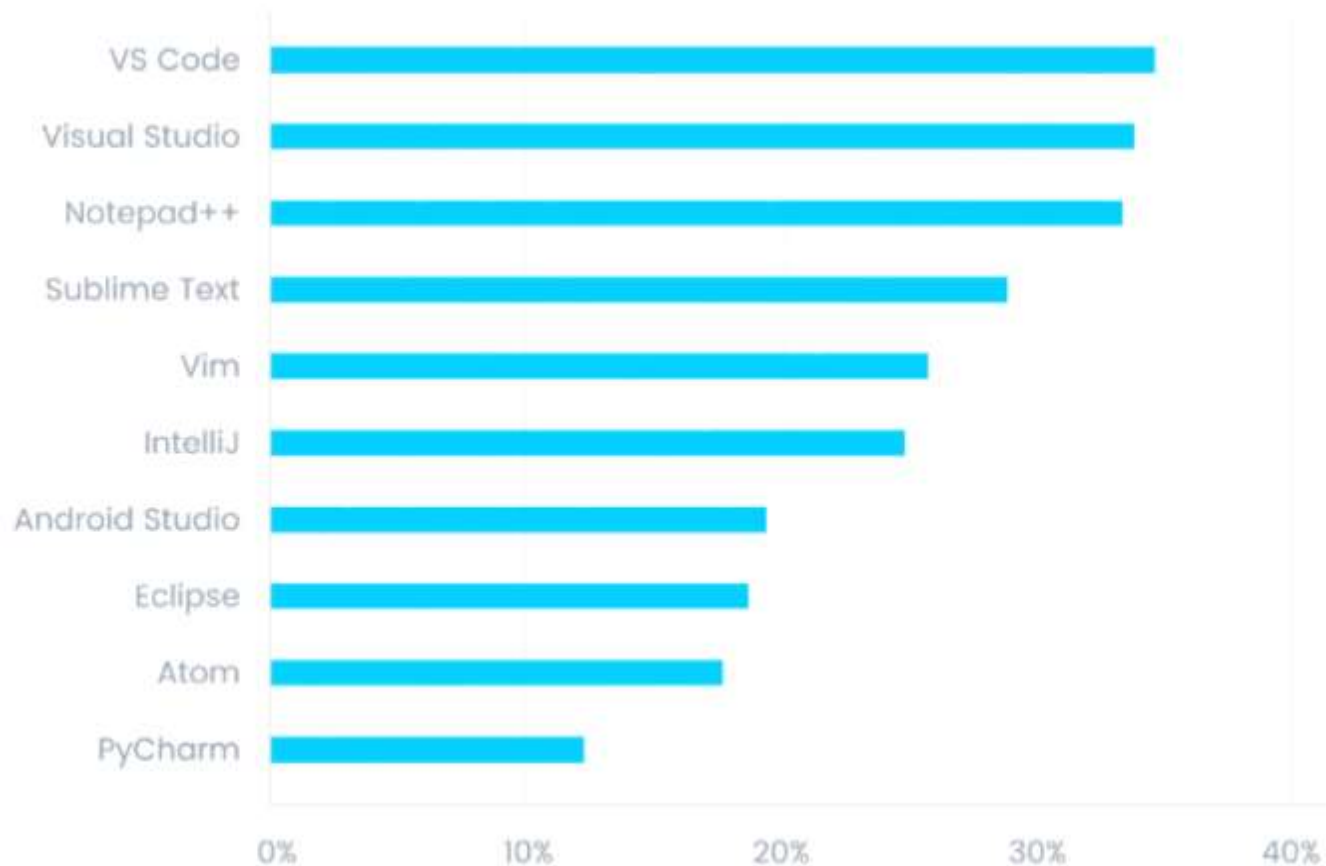
Visual Studio Code portable

A source code editor developed by Microsoft for Windows, Linux and macOS.

<https://portapps.io/app/vscode-portable/>

New editor: VScode

Most Popular Developer Environments



New editor: VScode

Download

Version

1.48.2-37

Platform

Windows 64-bits

Format

Portable setup (recommended)



Download

vscode-portable-win64-1.48.2-37-setup.exe

내 PC > DATA (D:) > Portable > vscode-portable >

이름

app

CHANGELOG

code

portapp

README

vscode-portable

하드디스크에 **portable** 폴더를 만들고
vscode-portable를 **portable** 폴더에
설치

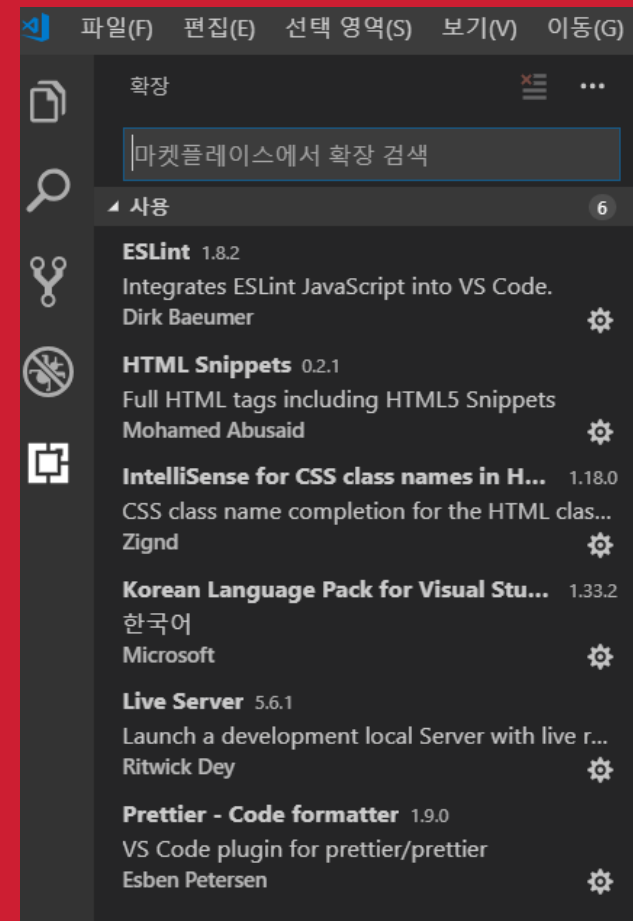
<https://portapps.io/app/vscode-portable/>

New editor: VScode

확장프로그램 설치 (설치 후 vscode 재시작)

1. Korean language pack
2. HTML snippet
3. IntelliSense for CSS class names in HTML
4. Javascript (ES6)
5. ESLint
6. Prettier
7. Live server (for HTML preview)
8. GitLens, Git History
9. Material Icon Theme
10. Python

C, C++, Java,
Node.js
Python, Jupyter
... all coding!



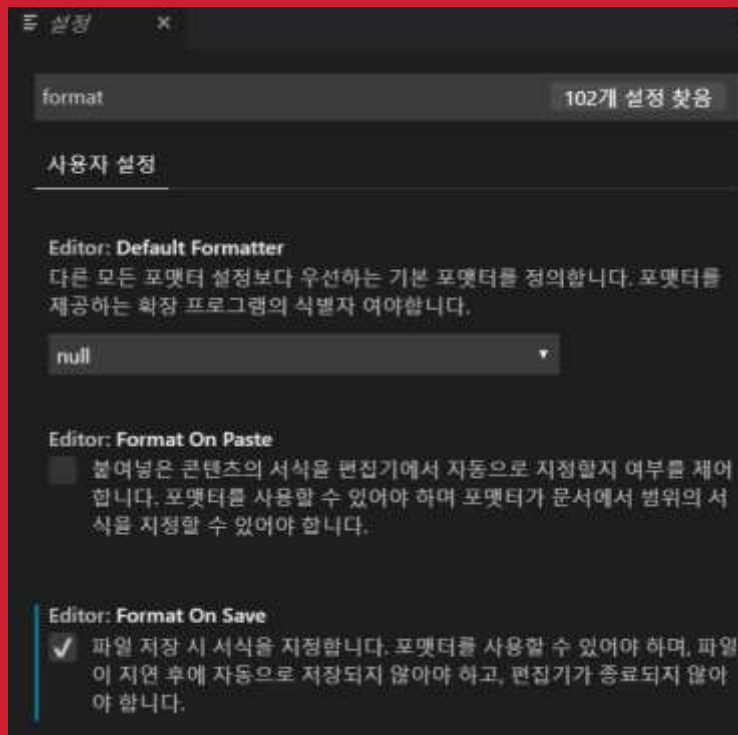
New editor: VScode

확장프로그램 설정 (각각 설정 후 종료/재시작)

1. Prettier

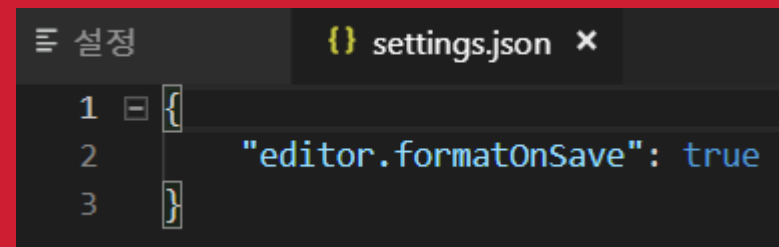
2. Live server (for HTML preview)

파일 > 기본설정 > 설정 → 사용자 설정



검색: format

Editor: Format on Save (check)



설정 → Ctrl + Shift + P

실습 준비: 작업영역 aann 설정

[파일] 메뉴

→ 작업영역에 폴더추가

하드디스크

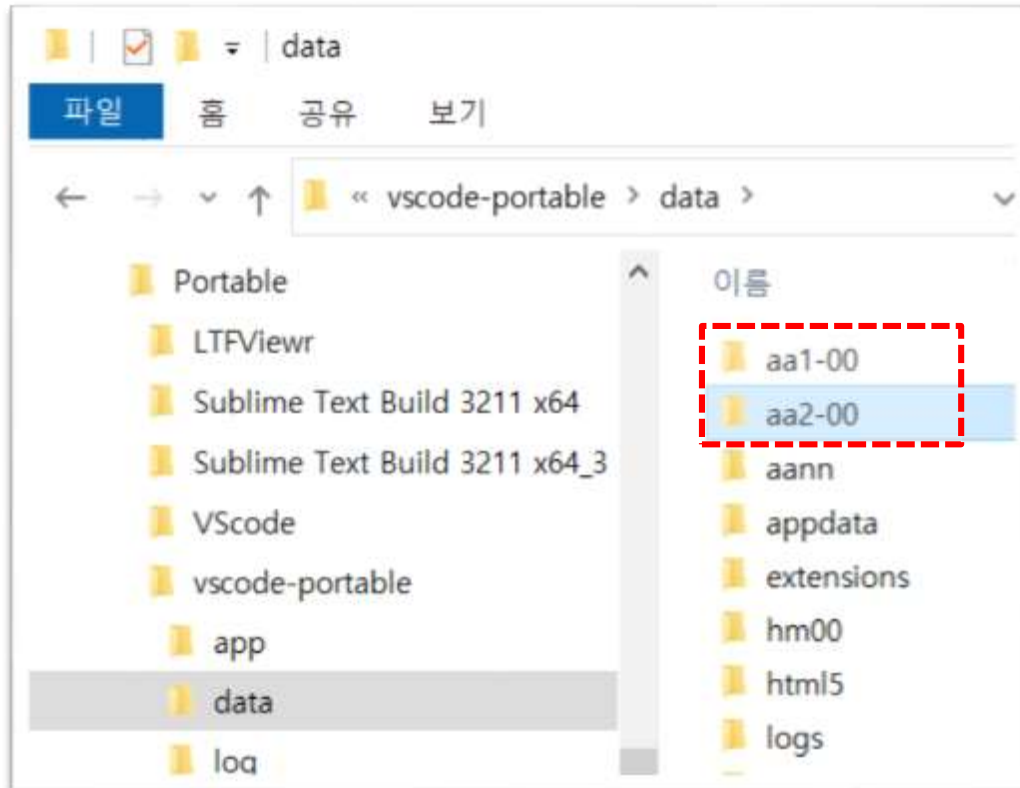
D:\Portable\vscode-
portable\data에 “aax-nn”
폴더 만들기

VSCode 탐색기에서

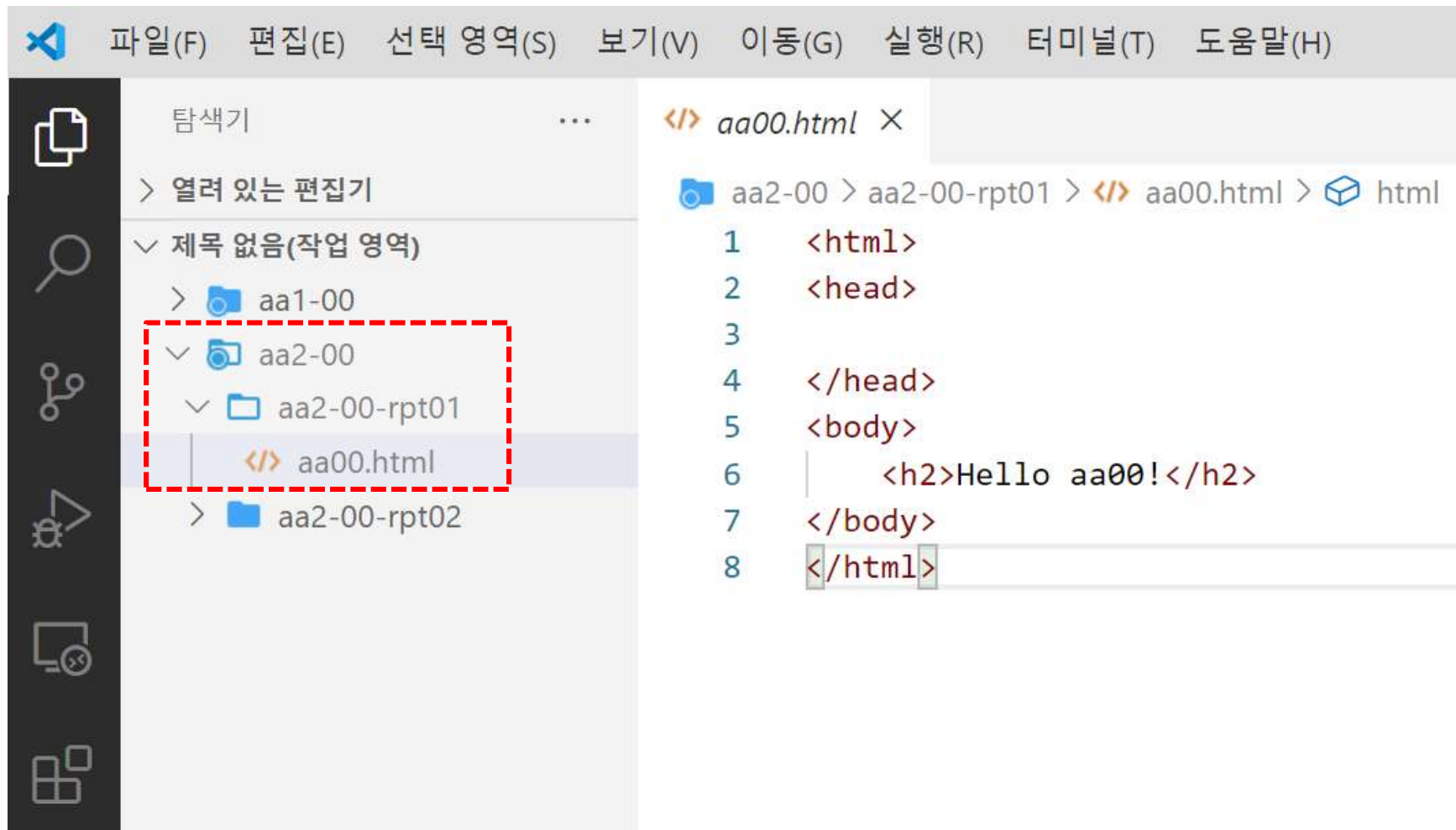
작업영역 클릭 → ‘aax-nn’
지정

‘aax-nn’ 에서 오른쪽마우스
클릭하여 새폴더를
aax-nn-rpt02 로 만든다.

aax-nn-rpt01 을 작업영역에
복사



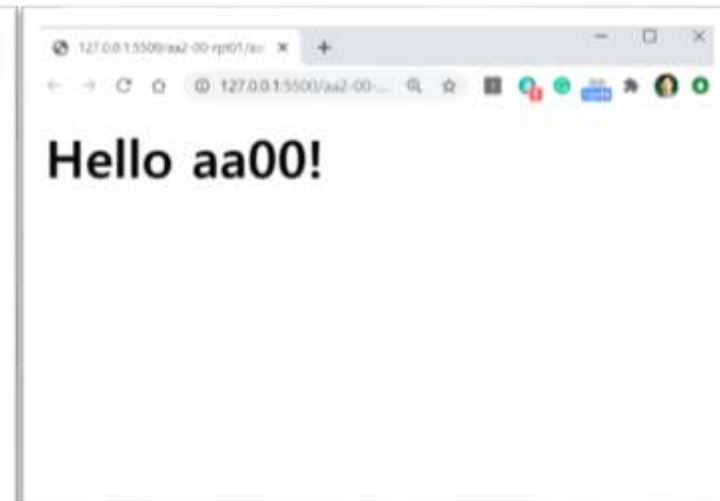
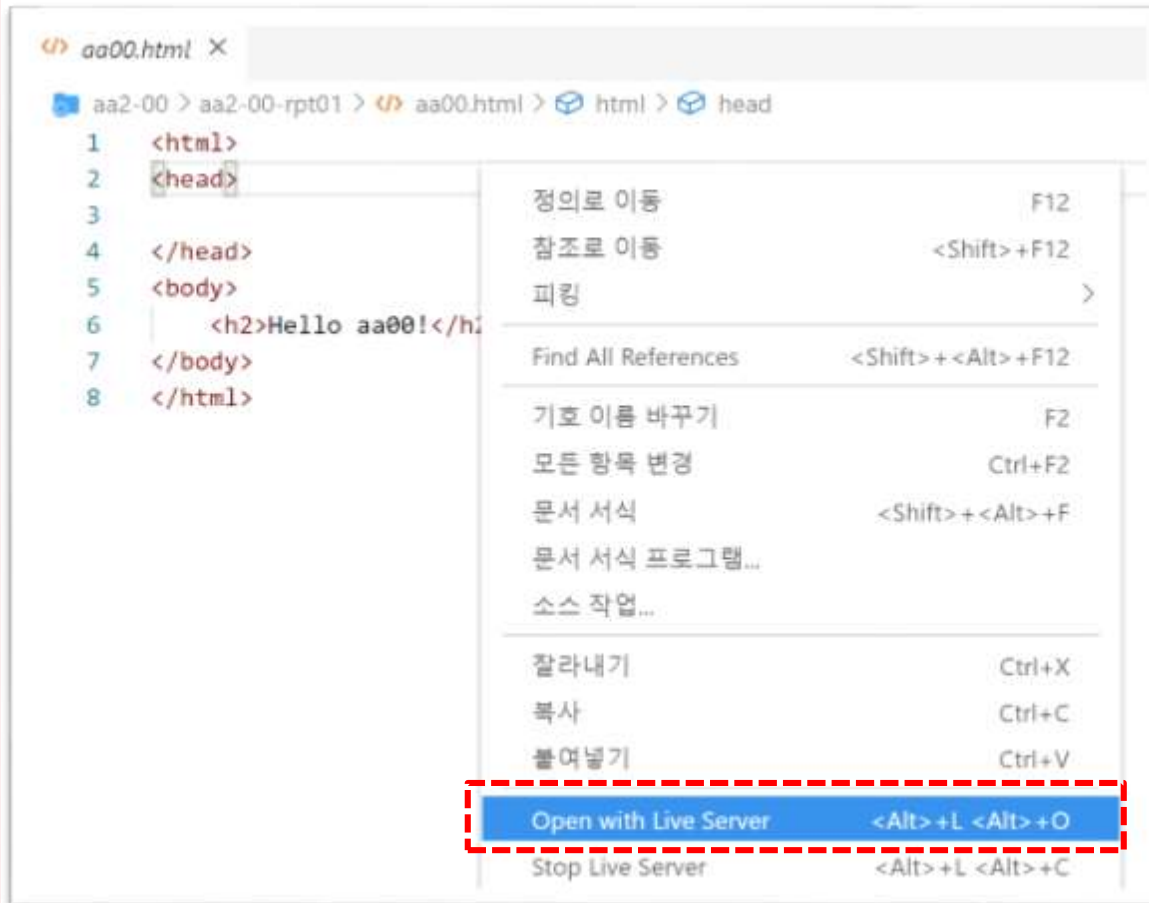
실습 준비: 작업영역 설정 – 코드 작성/복사



The screenshot displays the Visual Studio Code (VS Code) interface. The top menu bar includes options: 파일(F), 편집(E), 선택 영역(S), 보기(V), 이동(G), 실행(R), 터미널(T), and 도움말(H). The left sidebar shows the Explorer view with a tree structure under '제목 없음(작업 영역)'. The tree includes folders 'aa1-00', 'aa2-00', and 'aa2-00-rpt02'. Under 'aa2-00', there is a sub-folder 'aa2-00-rpt01' which contains the file 'aa00.html'. This file is highlighted with a red dashed box. The right pane shows the code editor for 'aa00.html'. The editor's title bar indicates the file path: 'aa2-00 > aa2-00-rpt01 > aa00.html'. The code content is as follows:

```
1 <html>
2 <head>
3
4 </head>
5 <body>
6 |   <h2>Hello aa00!</h2>
7 </body>
8 </html>
```

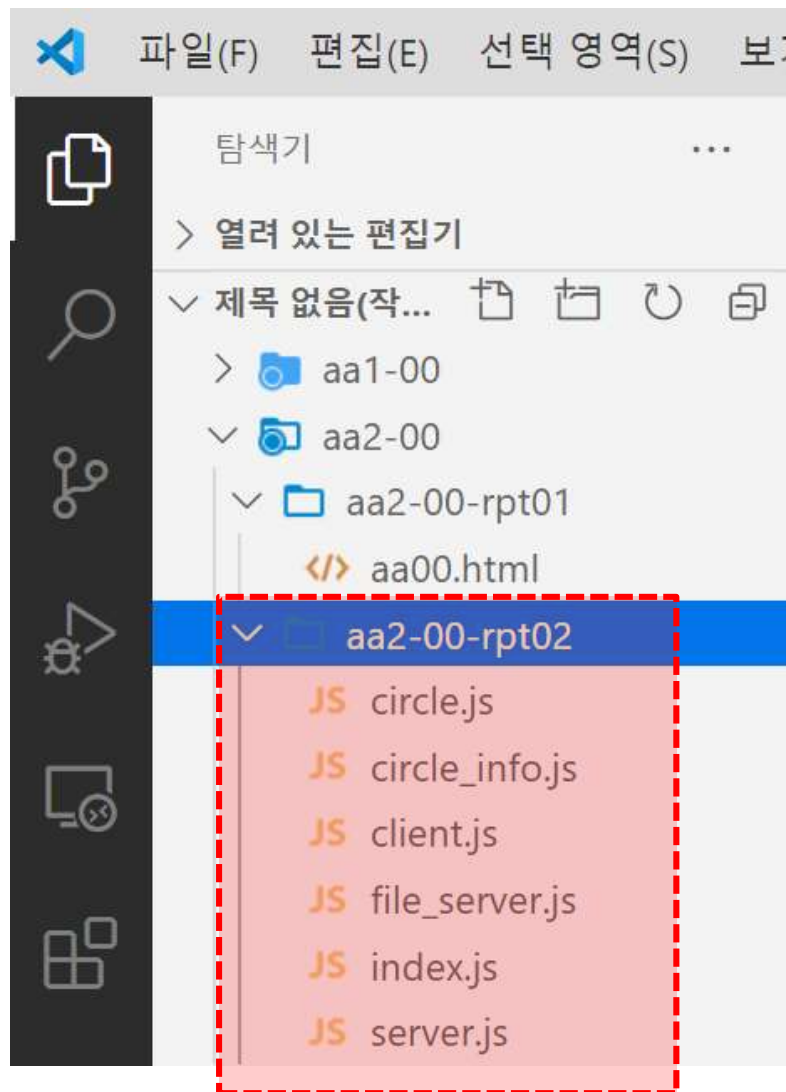
실습 준비: wk01 report: aann.html 실행



실습 준비: 작업영역 안에 소스 파일 복사

원도우 탐색기에서 github
에서 다운받아 압축을 풀은
소스 폴더 내의 모든 파일을
aax-nn-rpt02 폴더에 복
사

VSCode 탐색기에서
작업영역 클릭



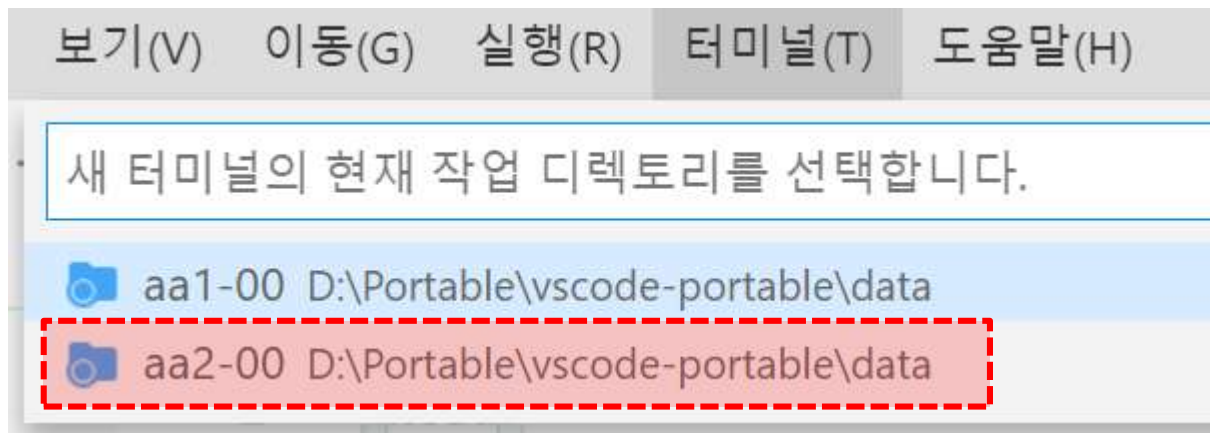


Node.js Project

npm init

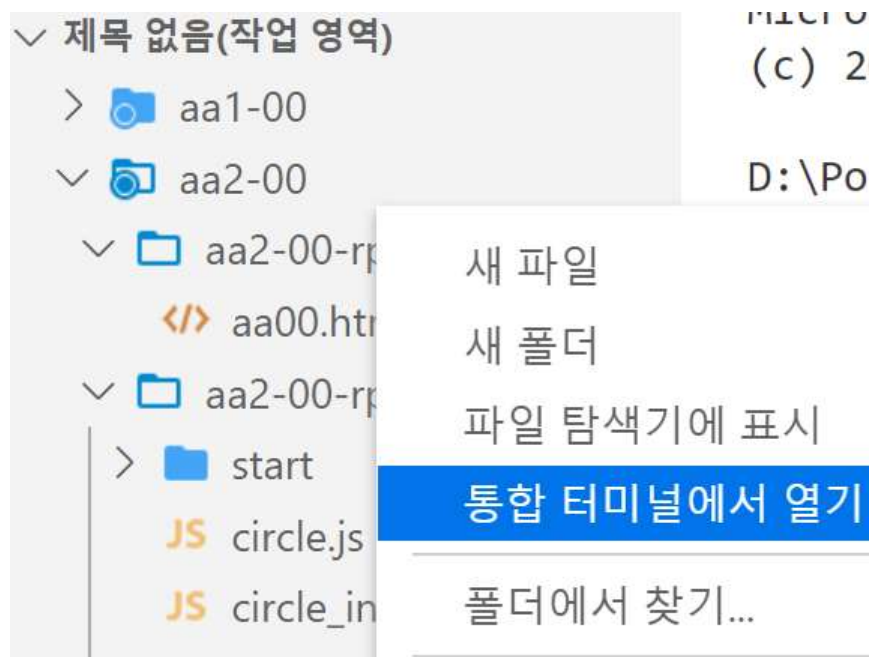


4.1 Node project : start terminal



// node cmd

md start
cd start





4.1 Node project : start terminal

문제 출력 디버그 콘솔 터미널

2: cmd



Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02>md start  
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02>cd start  
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>|
```

// node cmd

**md start
cd start**



4.2 Node project : **npm init**

문제 출력 디버그 콘솔 터미널

2: node

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>**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: (start)
version: (1.0.0)
description: start node project
entry point: (index.js)
test command:
git repository:
keywords: test
author: aa00
license: (ISC) MIT

// node project

npm init



4.3.1 Node project : package.json

About to write to D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start\package.json:

```
{
  "name": "start",
  "version": "1.0.0",
  "description": "start node project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "test"
  ],
  "author": "aa00",
  "license": "MIT"
}
```

Is this OK? (yes)

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>dir

D 드라이브의 볼륨: DATA

볼륨 일련 번호: 82D1-4852

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start 디렉터리

2020-09-08	오후 04:00	<DIR>	.
2020-09-08	오후 04:00	<DIR>	.
2020-09-08	오후 04:00		255 package.json
		1개 파일	255 바이트
		2개 디렉터리	2,515,994,062,848 바이트 남음



4.3.2 Node project : package.json

package.json

Node 프로젝트 설정 파일
- json file

The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'aa1-00' and 'aa2-00'. Under 'aa2-00', there are subfolders 'aa2-00-rpt01' and 'aa2-00-rpt02'. Inside 'aa2-00-rpt02', there is a folder 'start' which contains a file 'package.json'. The 'package.json' file is highlighted with a red dashed box. The code editor shows the content of 'package.json' with the following JSON structure:

```
1  
2  "name": "start",  
3  "version": "1.0.0",  
4  "description": "start node project",  
5  "main": "index.js",  
6  "scripts": {  
7    "test": "echo \"Error: no test specified\" && exit 1"  
8  },  
9  "keywords": [  
10    "test"  
11  ],  
12  "author": "aa00",  
13  "license": "MIT"  
14  
15
```

Below the code editor, there is a yellow box with the text: Save as AAnn_package.png



5. Node Apps

Node Apps



5.1.1 Hello Node! – hello.js

제목 없음(작업 영역)

> aa1-00

aa2-00

aa2-00-rpt01

aa00.html

aa2-00-rpt02

start

pac

circle

circle

client

새 파일

새 폴더

파일 탐색기에 표시

통합 터미널에서 열기

2
3
4
5
6
7

package.json

JS hello.js

X

aa2-00 > aa2-00-rpt02 > start > JS hello.js

```
1 // hello.js
2 console.log("Hello Node! by aa00");
3
```

문제 출력 디버그 콘솔 터미널

2: cmd

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>dir

D 드라이브의 볼륨: DATA

볼륨 일련 번호: 82D1-4852

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start 디렉터리

```
2020-09-08 오후 04:09 <DIR> .
2020-09-08 오후 04:09 <DIR> ..
2020-09-08 오후 04:11      50 hello.js
2020-09-08 오후 04:00    255 package.json
                   2개 파일          305 바이트
                   2개 디렉터리 2,515,993,612,288 바이트 남음
```

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node hello
Hello Node! by aa00
```

VSCode 터미널에서 실행 : node js-file-name



5.1.2 Hello Node! – npm start 로 실행

npm package.json X JS hello.js

aa2-00 > aa2-00-rpt02 > start > npm package.json > {} scripts

```
2   "name": "start",
3   "version": "1.0.0",
4   "description": "start node project",
5   "main": "index.js",
6   "scripts": {
7     "start": "node hello.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9  },
10  "keywords": [
11    "test"
12  ]
13}
```

문제 출력 디버그 콘솔 터미널 2: cmd

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node hello
Hello Node! by aa00

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>npm start

> start@1.0.0 start D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start
> node hello.js

Hello Node! by aa00
```



Using function in node



5.2.1 Using function in node

package.json JS hello_function.js X JS hello.js

aa2-00 > aa2-00-rpt02 > start > JS hello_function.js > ...

```
1 function hello(what) {  
2   console.log("Hello " + what + "!");  
3 }  
4  
5 hello("aa00");  
6 hello("Redwoods, 홍길동");  
7
```

문제 출력 디버그 콘솔 터미널

2: cmd

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node hello_function

Hello aa00!

Hello Redwoods, 홍길동!

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>



5.2.2 Using user-module: hello_module.js

JS hello_module.js X

aa2-00 > aa2-00-rpt02 > start > JS hello_module.js > ...

```
1 module.exports = function (what) {  
2   console.log("Hello " + what + "!");  
3 };  
4
```

사용자 정의 모듈

JS hello_my_module.js X

aa2-00 > aa2-00-rpt02 > start > JS hello_my_module.js > ...

```
1 var olleh = require("./hello_module");  
2  
3 olleh("aa00");  
4 olleh("Node");  
5
```

문제 출력 디버그 콘솔 터미널

2: cmd

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node hello_my_module  
Hello aa00!  
Hello Node!
```

Call user-module 'hello_module.js' from hello_my_module.js



[extra code] local module : circle.js

circle_info.js uses local module circle.js.

JS hello_module.js JS circle.js X

aa2-00 > aa2-00-rpt02 > JS circle.js > ...

```
1 // circle.js
2 var PI = Math.PI;
3
4 module.exports.area = function (r) {
5   return PI * r * r;
6 };
7
8 module.exports.circumference = function (r) {
9   return 2 * PI * r;
10 };
11
```

circle.js

JS hello_my_module.js JS circle_info.js ●

aa2-00 > aa2-00-rpt02 > JS circle_info.js > ...

```
1 // circle_info.js
2 var circle = require("../circle");
3 console.log("The area of a circle of radius 4 is " + circle.area(4));
4 console.log(
5   "The circumference of a circle of radius 4 is " + circle.circumference(4));
6
```

circle_info.js

문제 출력 디버그 콘솔 터미널

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node ../circle_info
The area of a circle of radius 4 is 50.26548245743669
The circumference of a circle of radius 4 is 25.132741228718345
```




Node.js Server

1. http, tcp, file

2. Express

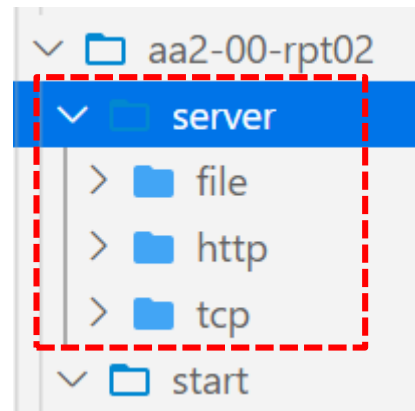
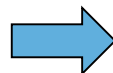
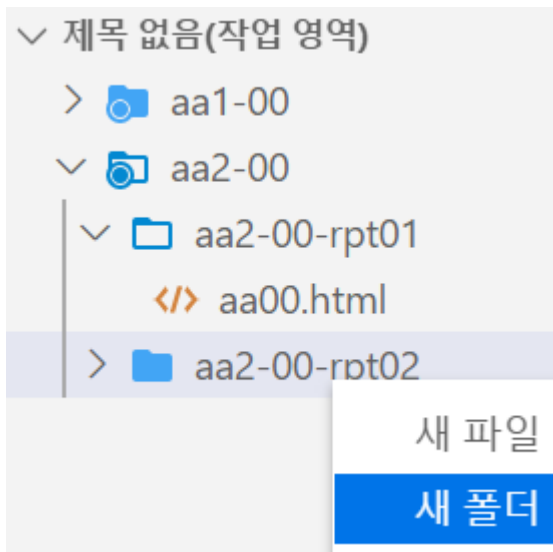


6. Node Server

Node Server I.

- 1. HTTP server**
- 2. TCP server**
- 3. File upload**

6.1 Node server : working folders





6.1.1 http server

JS index.js

aa2-00 > aa2-00-rpt02 > server > http > JS index.js > ...

```
1 // http server : index.js
2 var http = require("http");
3 port = 3000;
4
5 var server = http.createServer(function (request, response) {
6     response.writeHead(200, {
7         "Content-Type": "text/plain",
8     });
9     response.write("Hello HTTP server from node.js"); // WEB response
10    response.write("\n");
11    response.end();
12 });
13
14 server.listen(port);
15 console.log("Server Running on " + port + ".\nLaunch http://localhost:" + port);
```

문제 출력 디버그 콘솔 터미널

1: node



D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\http>node index

Server Running on 3000.

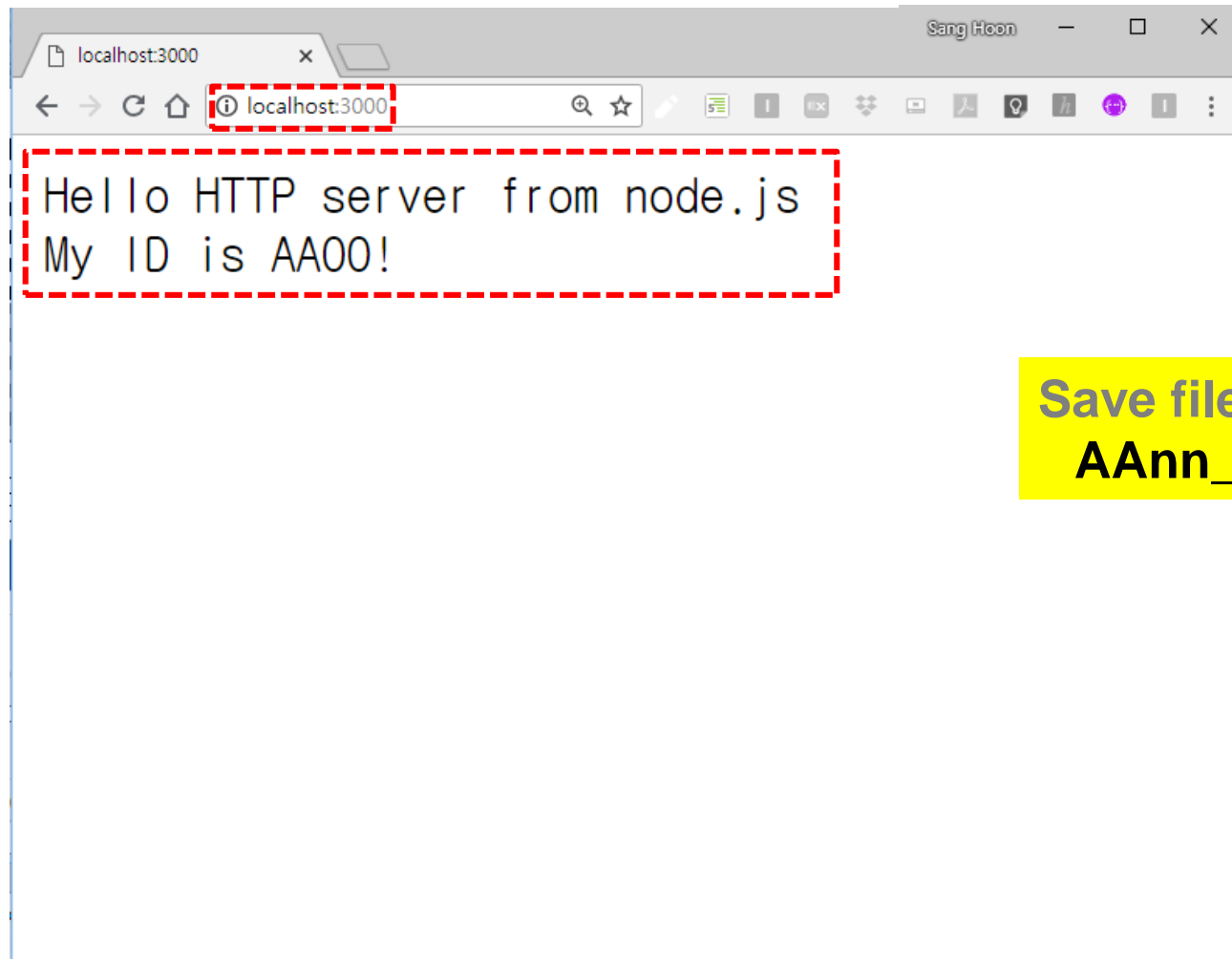
Launch http://localhost:3000



6.1.2 http server : result 1

Server Running on 3000.

Launch `http://localhost:3000`



Save file

AAnn_HTTP.png



6.1.3 http server – stop server !!! → ^C

문제 출력 디버그 콘솔 터미널

1: cmd

Server Running on 3000.

Launch http://localhost:3000

^C

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\http>

[Tip] port number

★사용해도 되는 포트번호와 사용할 수 없는 포트 번호

1. 잘 알려진 포트는 0~1023 까지입니다.

(특정 프로그램들이 사용하기로 예약되어 있기 때문에 쓸 수 없는 포트 번호)

2. 등록된 포트는 1024~49151 까지입니다. (사용가능)

3. 동적 및/또는 개인 포트는 49152~65535 까지입니다. (사용가능)

참고: <http://support.microsoft.com/kb/174904/ko>

[Tip] listen EADDRINUSE 오류 해결 -1

1. listen **EADDRINUSE** 오류

-사용중인 포트이거나 포트를 중지시키지 않고 종료시켰을 경우 계속 포트가 사용되고 있는데 연결 시키려고 할 때 나타나는 오류

```
Server Running on 3000.  
Launch http://localhost:3000  
events.js:154  
    throw er; // Unhandled 'error' event  
    ^  
  
Error: listen EADDRINUSE :::3000  
    at Object.exports._errnoException (util.js:856:11)  
    at exports._exceptionWithHostPort (util.js:879:20)
```

2. 해결 방법

- (1) cmd창에서 **netstat -ano**를 입력한 후, 로컬주소에서 사용 중인 포트 번호를 확인
- (2) 사용중인 포트번호를 확인하고 그 해당포트의 **pid번호를 확인**한다.

[Tip] listen EADDRINUSE 오류 해결 -2

```
Node
v5.7.0
D:\Portable\nodejs\portable\data>netstat -ano

활성 연결

프로토콜  로컬 주소          외부 주소          상태          PID
TCP        0.0.0.0:135     0.0.0.0:0          LISTENING      1040
TCP        0.0.0.0:445     0.0.0.0:0          LISTENING      4
TCP        0.0.0.0:3000    0.0.0.0:0          LISTENING      14332
TCP        0.0.0.0:14430   0.0.0.0:0          LISTENING      24316
TCP        0.0.0.0:14440   0.0.0.0:0          LISTENING      24316
TCP        0.0.0.0:17500   0.0.0.0:0          LISTENING      25096
TCP        0.0.0.0:21300   0.0.0.0:0          LISTENING      16228
TCP        0.0.0.0:30403   0.0.0.0:0          LISTENING      3660
TCP        0.0.0.0:30409   0.0.0.0:0          LISTENING      3668
```

2. 해결 방법 (cmd에서 다음 명령 실행 후 port 3000의 pid가 제거됨을 확인) **taskkill /pid PID_number**

```
D:\Portable\nodejs\portable\data>taskkill /pid 14332
성공: 프로세스(PID 14332)에 종료 신호를 보냈습니다.
```

```
D:\Portable\nodejs\portable\data>netstat -ano
```

활성 연결

프로토콜	로컬 주소	외부 주소	상태	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1040
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:14430	0.0.0.0:0	LISTENING	24316



6.2.1 tcp server (socket connection)

JS server.js ×

aa2-00 > aa2-00-rpt02 > server > tcp > JS server.js > ...

```
1 // tcp server (network server)
2 var net = require("net");
3 var port = 3000;
4
5 // Network connection using socket
6 var server = net.createServer(function (socket) {
7     console.log("Connection from " + socket.remoteAddress);
8     socket.end("Hello AA00! from localhost:3000");
9 });
10
11 server.listen(port, "127.0.0.1");
12 console.log("Network server started at port : " + port);
13
```

Socket으로 전송

문제 출력 디버그 콘솔 터미널

1: node

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>node server
Network server started at port : 3000
```



6.2.2 tcp client

JS client.js X

aa2-00 > aa2-00-rpt02 > server > tcp > JS client.js > ...

```
1 // tcp client
2 var net = require("net");
3 var port = 3000;
4 var client = new net.Socket();
5
6 // Connection using socket
7 client.connect(port, "127.0.0.1");
8
9 // Receive data from socket, callback function
10 client.on("data", function (data) {
11   console.log("Data: " + data);
12   client.destroy();
13 });
14
15 // Add a 'close' event handler for the client socket
16 client.on("close", function () {
17   console.log("Connection closed");
18 });
19
```

Socket으로 전송되는
데이터를 처리하고 종료

1: node, cmd

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed
```

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>
```



6.2.3 tcp server & client : result

```
JS server.js x
aa2-00 > aa2-00-rpt02 > server > tcp > JS server.js > ...
1 // tcp server (network server)
2 var net = require("net");
3 var port = 3000;
4
5 // Network connection using socket
6 var server = net.createServer(function (socket) {
7   console.log("Connection from " + socket.remoteAddress);
8   socket.end("Hello AA00! from localhost:3000");
9 });
10
11 server.listen(port, "127.0.0.1");
12 console.log("Network server started at port : " + port);
13

JS client.js x
aa2-00 > aa2-00-rpt02 > server > tcp > JS client.js > ...
1 // tcp client
2 var net = require("net");
3 var port = 3000;
4 var client = new net.Socket();
5
6 // Connection using socket
7 client.connect(port, "127.0.0.1");
8
9 // Receive data from socket, callback function
10 client.on("data", function (data) {
11   console.log("Data: " + data);
12   client.destroy();
13 });
14
15 // Add a 'close' event handler for the client socket
16 client.on("close", function () {
17   console.log("Connection closed");
18 });
19

문제 줄러 디버그 콘솔 터미널
Network server started at port : 3000
Connection from 127.0.0.1
Connection from 127.0.0.1
Connection from 127.0.0.1
Connection from 127.0.0.1

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>
```

Save file

AAnn_TCP_Log.png



6.3.1 file upload using module 'formidable'

JS file_server.js X

aa2-00 > aa2-00-rpt02 > server > file > JS file_server.js > http.createServer() callback > form.parse

```
1 // File upload using formidable node module
2 var formidable = require("formidable"),
3     http = require("http"),
4     util = require("util"),
5     port = 3663;
6
7 http
8   .createServer(function (req, res) {
9     if (req.url == "/upload" && req.method.toLowerCase() == "post") {
10      // parse a file upload
11      var form = new formidable.IncomingForm();
12
13      form.parse(req, function (err, fields, files) {
14        res.writeHead(200, { "content-type": "text/plain" });
15        res.write("received upload:\n\n");
16        res.end(util.inspect({ fields: fields, files: files }));
17      });
18      return;
19    }
20  })
```

문제 출력 디버그 콘솔 터미널

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>node file_server
File server Running on 3663.
Launch http://localhost:3663
```




6.3.2 file upload : npm install formidable

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>npm install formidable
npm WARN saveError ENOENT: no such file or directory, open 'D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file\package.json'
npm WARN file No description
npm WARN file No repository field.
npm WARN file No README data
npm WARN file No license field.
```

```
+ formidable@1.2.2
added 1 package and audited 1 package in 0.41s
```

```
1 package is looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

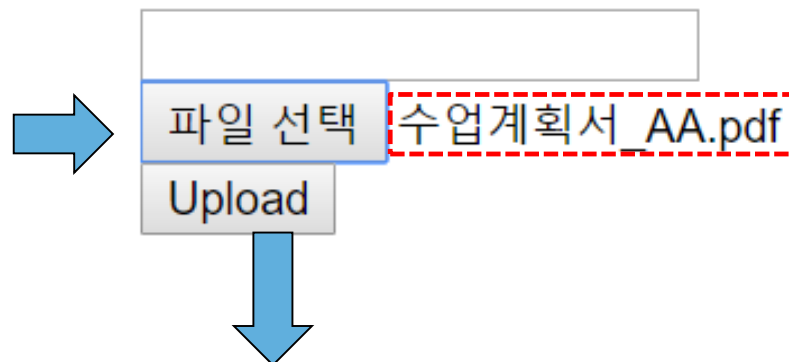
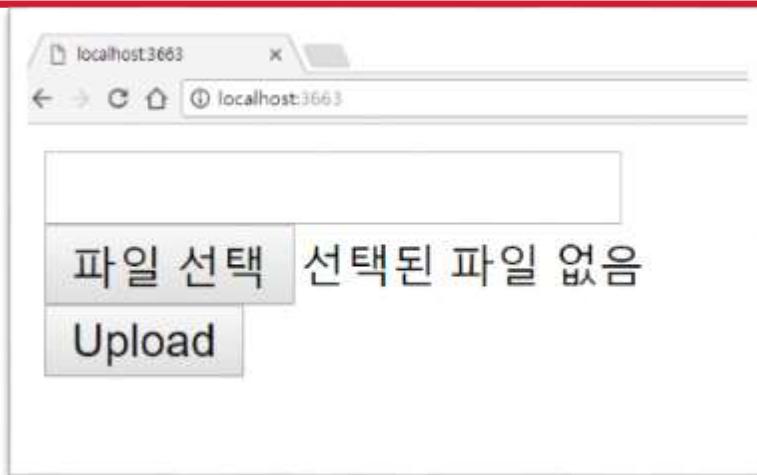
```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>dir
```

```
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 82D1-4852
```

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file 디렉터리
```

```
2020-09-08 오후 05:56 <DIR>      .
2020-09-08 오후 05:56 <DIR>      ..
2020-09-08 오후 05:54      1,116 file_server.js
2020-09-08 오후 05:56 <DIR>      node_modules
2020-09-08 오후 05:56      325 package-lock.json
                2개 파일          1,439 바이트
                3개 디렉터리 2,515,992,223,744 바이트 남음
```

6.3.3 file upload



localhost:3663/upload

received upload:

```

{
  fields: { title: '' },
  files: {
    upload: File {
      _events: [Object: null prototype] {},
      _eventsCount: 0,
      _maxListeners: undefined,
      size: 174524,
      path:
        'C:\\\\Users\\\\life21c\\\\AppData\\\\Local\\\\Temp\\\\upload_137cb331c7ded8fa36c80
        d5eb8dc9cea'
      name: '&#49688;&#50629;&#44228;&#54925;&#49436;_AA.pdf'
      type: 'application/pdf',
      hash: null,
      lastModifiedDate: 2020-09-08T09:03:02.527Z,
      _writeStream: [WriteStream],
      [Symbol(kCapture)]: false
    }
  }
}

```

Save file
AAnn_Upload.png



[Practice]

◆ [wk02]

- Node module : **aanninfo.js**
- Upload folder: **aax-nn-rpt02**



[practice] local module : aanninfo.js

index_aann.js uses local module **aanninfo.js** in start subfolder.



```
1 // index_aann.js
2
3 var myinfo = require('./aanninfo');
4
5 myinfo("aa00", "Redwoods", '010-1234-5678');
6
7 myinfo("aa55", "Comsi", '010-5678-1234');
```

My Info
ID : aa00
Name : Redwoods
Phone : 010-1234-5678

My Info
ID : aa55
Name : Comsi
Phone : 010-5678-1234

[Finished in 0.2s]

Save as
AAnn_info.png

[practice] local module : aanninfo.js

How to make aanninfo.js in start subfolder.

1. Make local module – aanninfo.js
2. Call aanninfo.js from index_aann.js.
3. Capture your result.

```

1 // aanninfo.js
2
3 module.exports = function(id, name, phone) {
4     console.log("My Info");
5     console.log("ID : " + id);
6     console.log("Name : " + name);
7     console.log("Phone : " + phone + "\n");
8 }
    
```

[\[참고\] Node local module 만들기](#)

◆ [Target of this week]

My Info using node module – aanninfo.js

Upload folder : aax-nn-rpt02

- 제출할 파일들

- ① **AAnn_package.png**
- ② **AAnn_HTTP.png**
- ③ **AAnn_TCP_Log.png**
- ④ **AAnn_Upload.png**
- ⑤ **AAnn_info.png**
- ⑥ **start folder**
- ⑦ **server folder**

[Upload to github]

◆ [wk02]

- upload all work of this week
- Use repo “aax-nn” in github
- upload folder “aax-nn-rpt02”
in your github.

● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <https://plot.ly/> plotly
- ✓ <https://www.mongodb.com/> MongoDB
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub



주교재 및 참고도서

아두이노와 Node.js에 기반한 IOT 신호 시각화

| 저자 이 상 훈 |

인제대학교 출판부

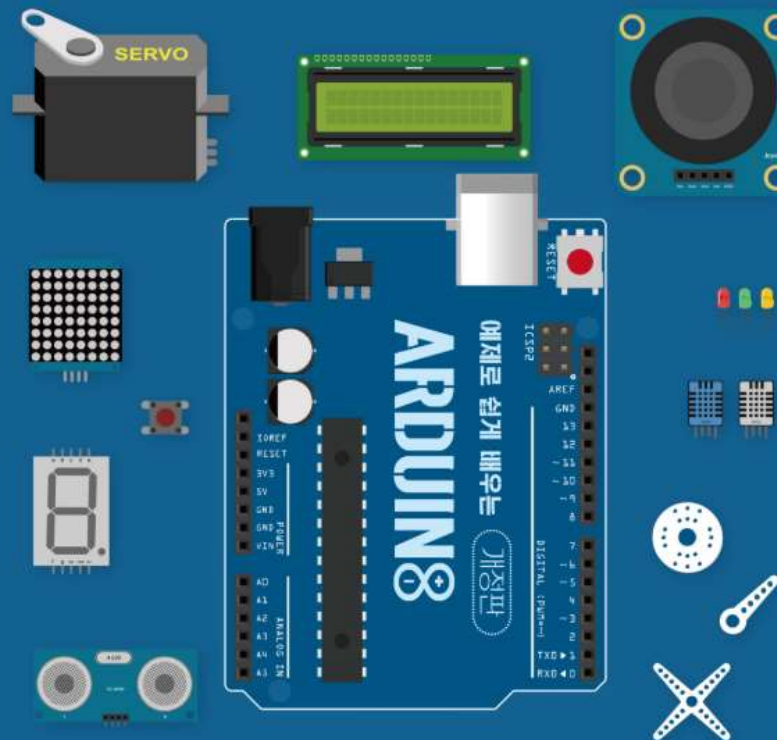
아두이노와 Node.js에 기반한

IOT 신호 시각화

| 저자 이 상 훈 |



인제대학교 출판부



예제로 쉽게 배우는

아두이노

개정판

장성용 · 김진환 지음

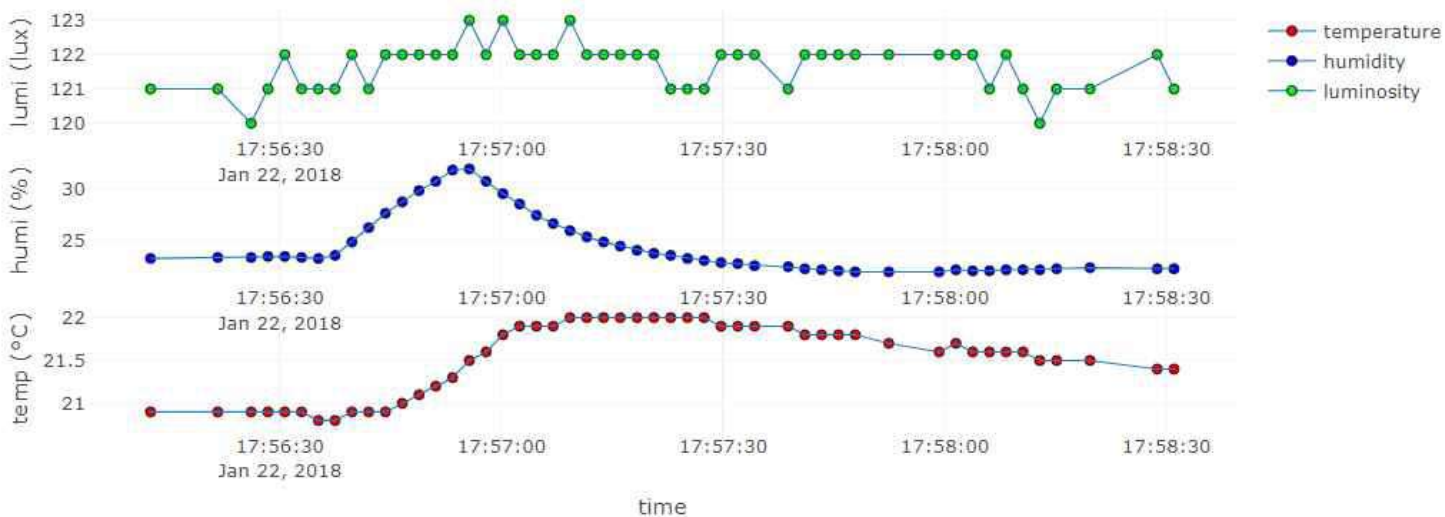


Target of this class

Real-time Weather Station from sensors



on Time: 2018-01-22 17:58:31.012

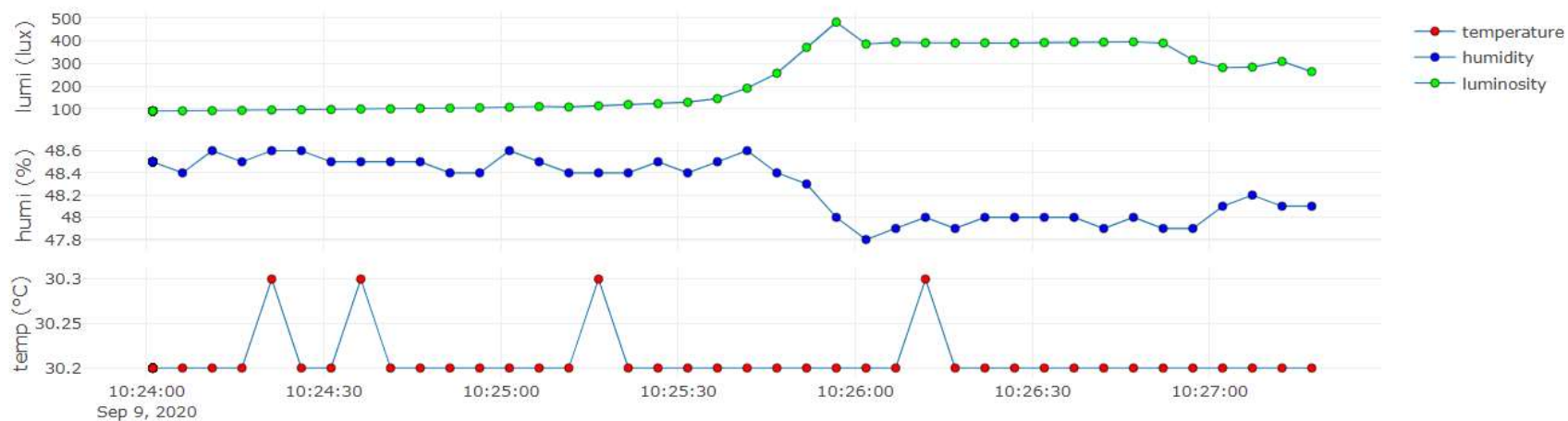


Target of this class

Real-time Weather Station from nano 33 BLE sensors



on Time: 2020-09-09 10:27:17.321



Another target of this class

PPG with rangeslider

