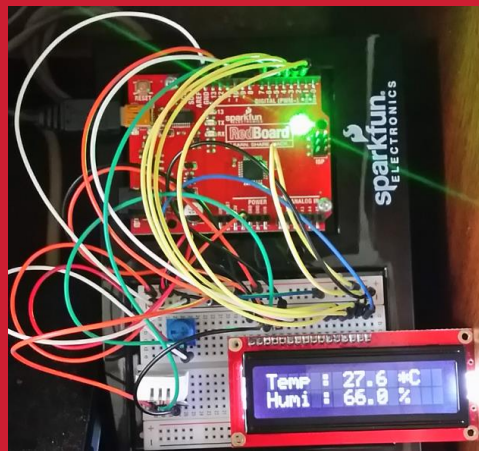
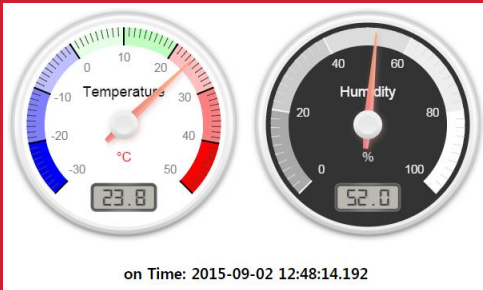




Healthcare-IoT

[wk02]

Start Node.js

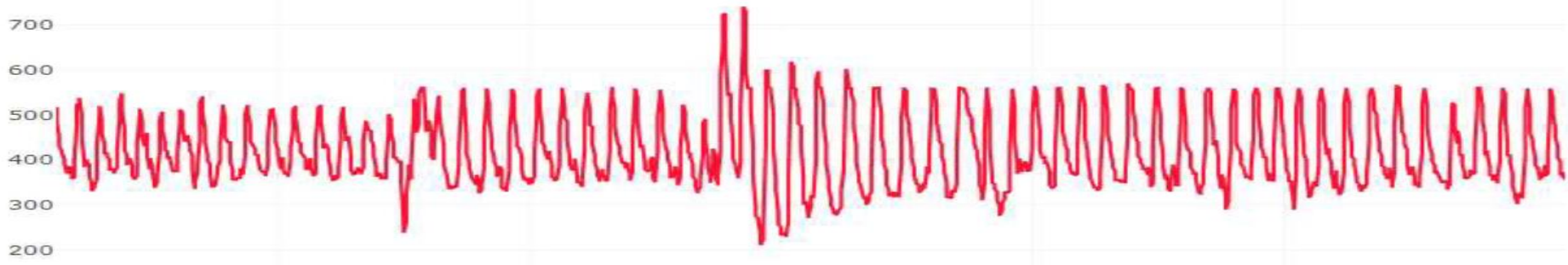


Visualization of Healthcare Signals using
Arduino & Node.js

HCit, INJE University

1st semester, 2018

Email : chaos21c@gmail.com





My ID

오전

성명	ID
김민선	HS01
김영걸	HS02
김주란	HS03
김주현	HS04
김태민	HS05
여준하	HS06
이수민	HS07
정민지	HS08
정유현	HS09
정재은	HS10
주하영	HS11
한준영	HS12

오후

성명	ID
신영주	HS21
오가영	HS22
윤민수	HS23
윤진아	HS24
이진영	HS25
임상은	HS26
임재형	HS27
최민영	HS28
황유빈	HS29

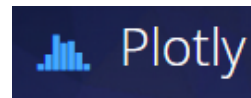
주간계획서

주간계획서			
주차	수업방법	수업내용	과제물
1	강의/실습	수업 및 실습 안내 - 포터블 소프트웨어 설치	
2	강의/실습	Node.js I - Node.js 코드의 기본 구조 - 기초 Node 서버 및 클라이언트	실습확인
3	강의/실습	Node.js II - Node.js Express 서버	실습확인
4	강의/실습/발표	Arduino I - 아날로그 신호 회로 - LCD를 이용한 센서 신호 모니터링	실습확인
5	강의/실습	Arduino II - 단일 센서 회로와 Node.js 연결 - 다중 센서 회로와 Node.js 연결	실습확인
6	강의/실습	프로젝트I - 생체 센서 회로와 Node.js 연결 - 생체 신호 소개	프로젝트I
7	강의/실습/발표	IOT 데이터 시각화 I (Plotly.js) - 데이터 및 시계열 차트 - 데이터 스트리밍	실습확인
8	시험	중간고사	
9	강의/실습	IOT 데이터 시각화 II (Plotly.js) - 다중 센서 데이터 시각화 - 다중 센서 데이터 스트리밍	실습확인
10	강의/실습/발표	프로젝트II - 생체 센서 데이터 시각화 - 생체 센서 데이터 스트리밍	프로젝트II
11	강의/실습	IOT 데이터 저장과 처리 - MongoDB 설치 및 Mongo shell - MongoDB와 Node.js 연결 및 데이터 저장	실습확인
12	강의/실습	프로젝트III - MongoDB에 IOT 데이터 저장 및 모니터링 - 생체 센서 데이터 저장 및 시각화	프로젝트III
13	강의/실습	IOT 데이터 마이닝 - 아두이노에서 발생한 데이터 관리 - 데이터마이닝 소개	실습확인
14	강의/실습/발표	프로젝트IV - 생체 센서 데이터 관리 - 생체 센서 데이터 마이닝	프로젝트IV
15	시험	기말고사	

Purpose of HS

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리
4. 생체 센서 발생 신호 처리, 시각화 및 저장
5. 생체 센서 발생 신호 저장 및 분석
6. 생체 신호 장비 활용 능력





[Review]

◆ [wk01]

➤ test : [hsnn.html](#)

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



The screenshot shows the Sublime Text 3 interface. The title bar indicates the file path is D:\Portable\Sublime Text Build 3143 x64\Data\hs00\hs00.html (hs00) - Sublime Text (UNREGISTERED). The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. On the left, the 'FOLDERS' panel shows a folder named 'hs00' with a sub-file 'hs00.html' highlighted by a red dashed rectangle. The main editor area shows the content of 'hs00.html' with line numbers 1 through 10. The code is as follows:

```

1  <html>
2      <head>
3
4      </head>
5
6      <body>
7          <h2>Hello hs00!</h2>
8      </body>
9
10 </html>

```

wk01 : hsnn.html (2nd method)

서브라임텍스트3 에서 html파일 크롬 으로 실행하기

1. 댓글

서브라임텍스트3에서 html파일을 작업하다가 페이지를 바로 크롬 브라우저로 실행하는 방법이다.

1. 서브라임텍스트3 에서 Tools > Build System > New Build System

```
1
2 {
3   "cmd":
4     ["c:\\ProgramFiles\\Google\\Chrome\\Application\\chrome.exe", "$file"]
5 }
```

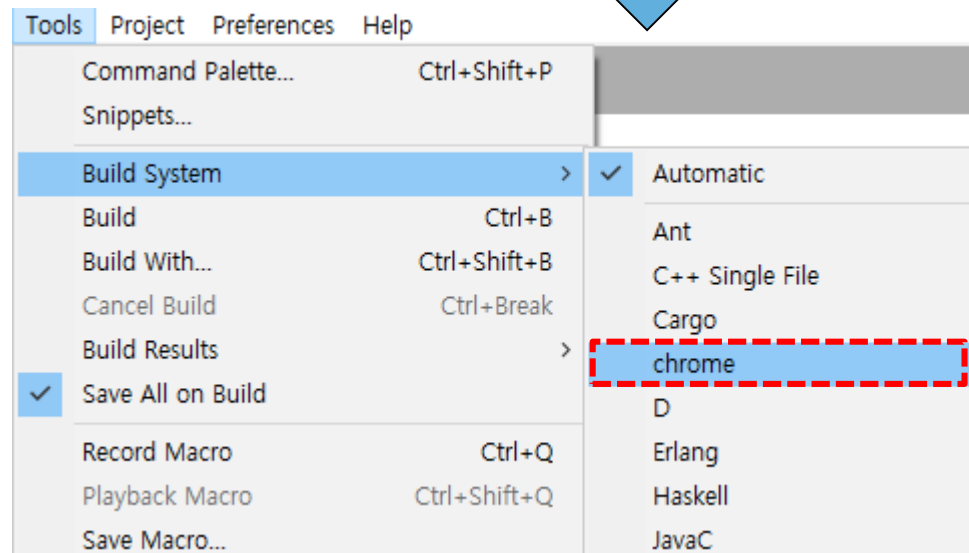
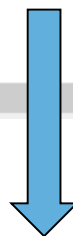
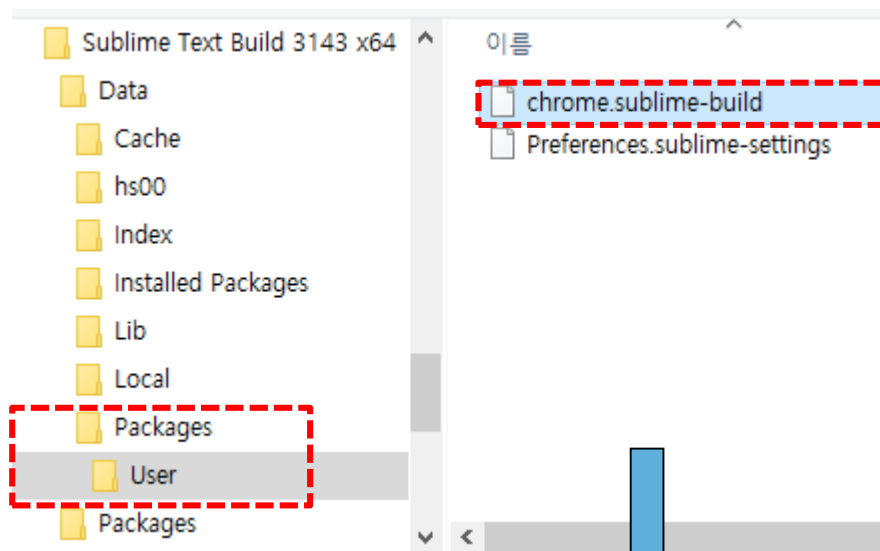
2. 위와 같이 작성하고 Chrome.sublime-build 이름으로 저장한다.

여기서 주의해야할 사항은 Chrome.exe 위치가 다를 수도 있으니 확인해보고 작성한다. 나는 programFilex (x86) 에 있어서 수정하고 적용했다. 그리고 역슬래시는 두개씩(\\) 들어간다.

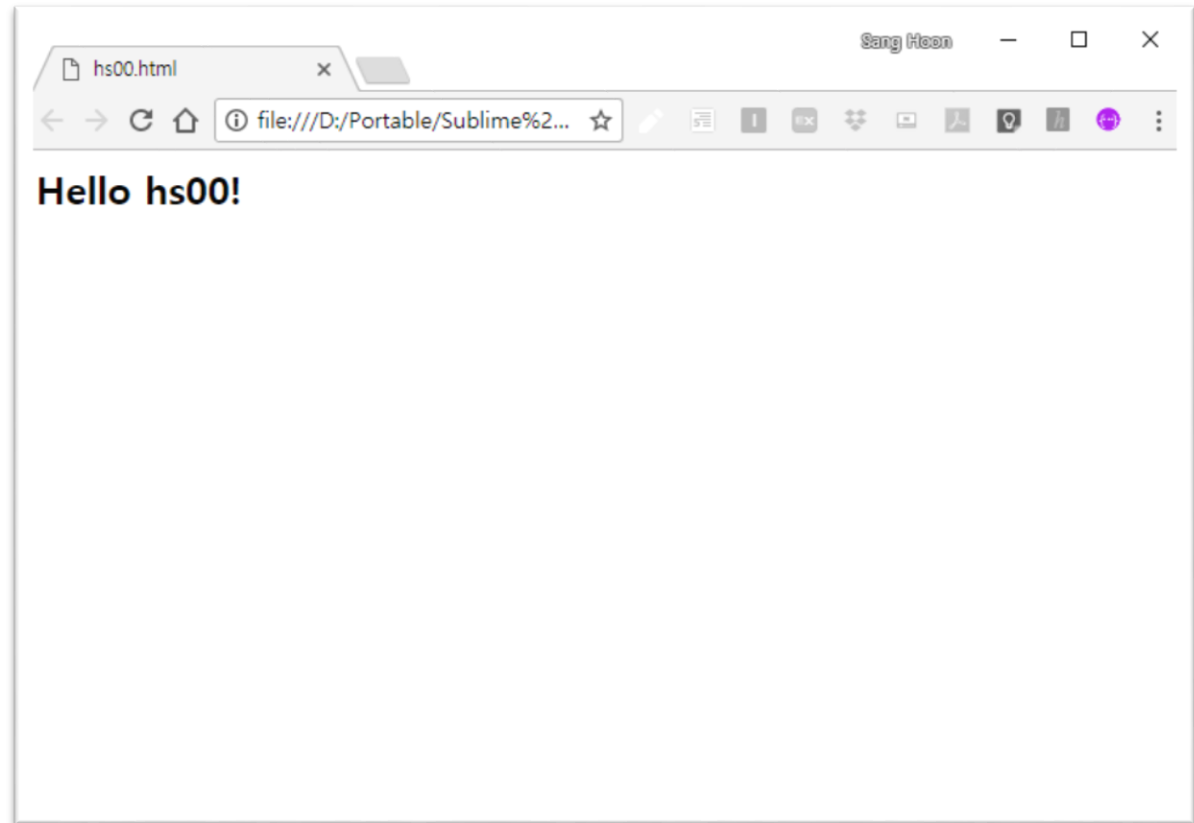
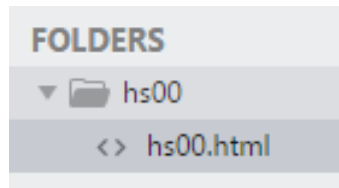
3. 프로그램을 재실행 한 후 Tools > Build System에 Chrome이라고 생겼으면 확인하고 선택한다.

4. html 파일에서 Ctrl + b를 누르면 크롬에서 바로 실행된다.

wk01 : hsnn.html



wk01 : hsnn.html – result!





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');
```

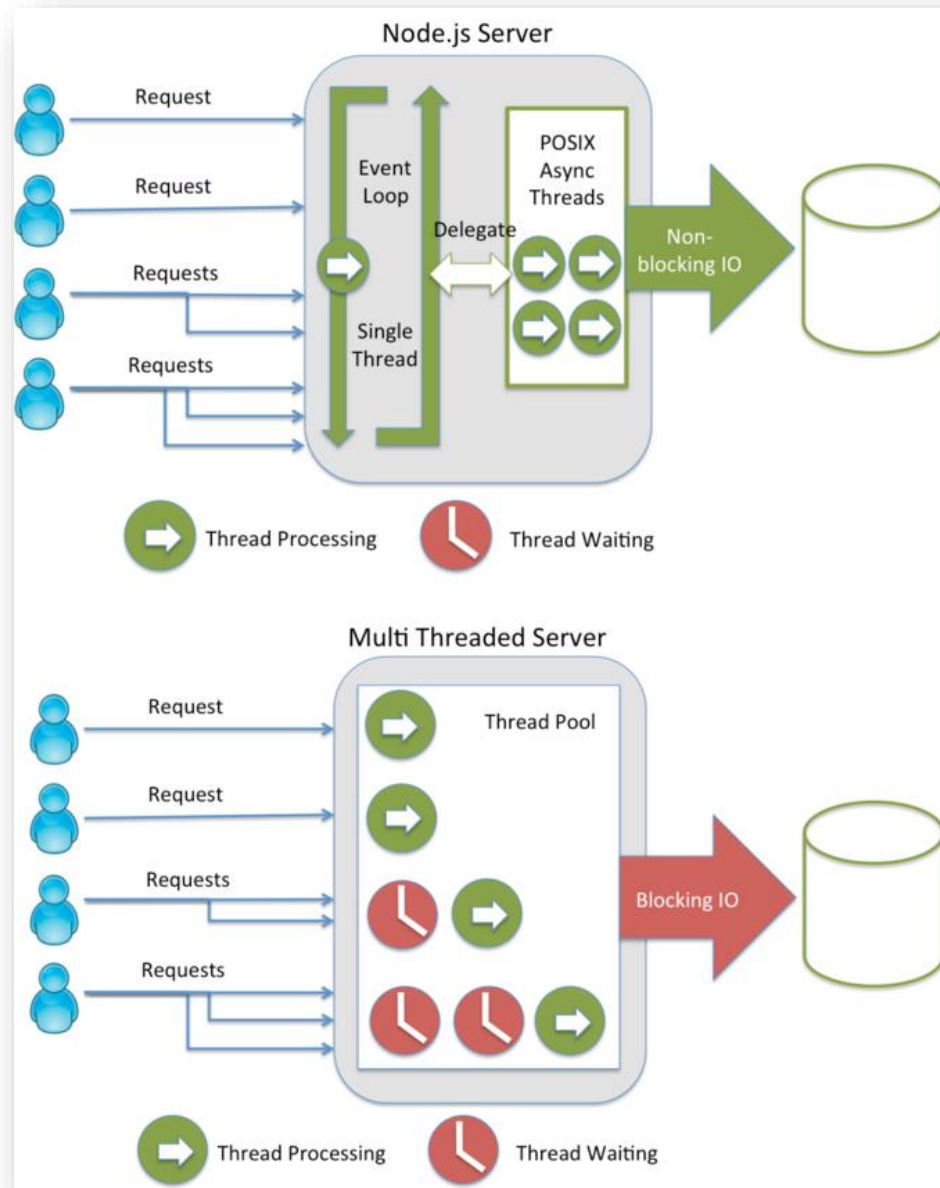
[Quiz] 계란 1개를 삶을 때 10분이 걸린다.

그러면 계란 10개를 삶을 때 걸리는 시간은?

Blocking : ?

Non-blocking : ?

1.3 Non-blocking IO

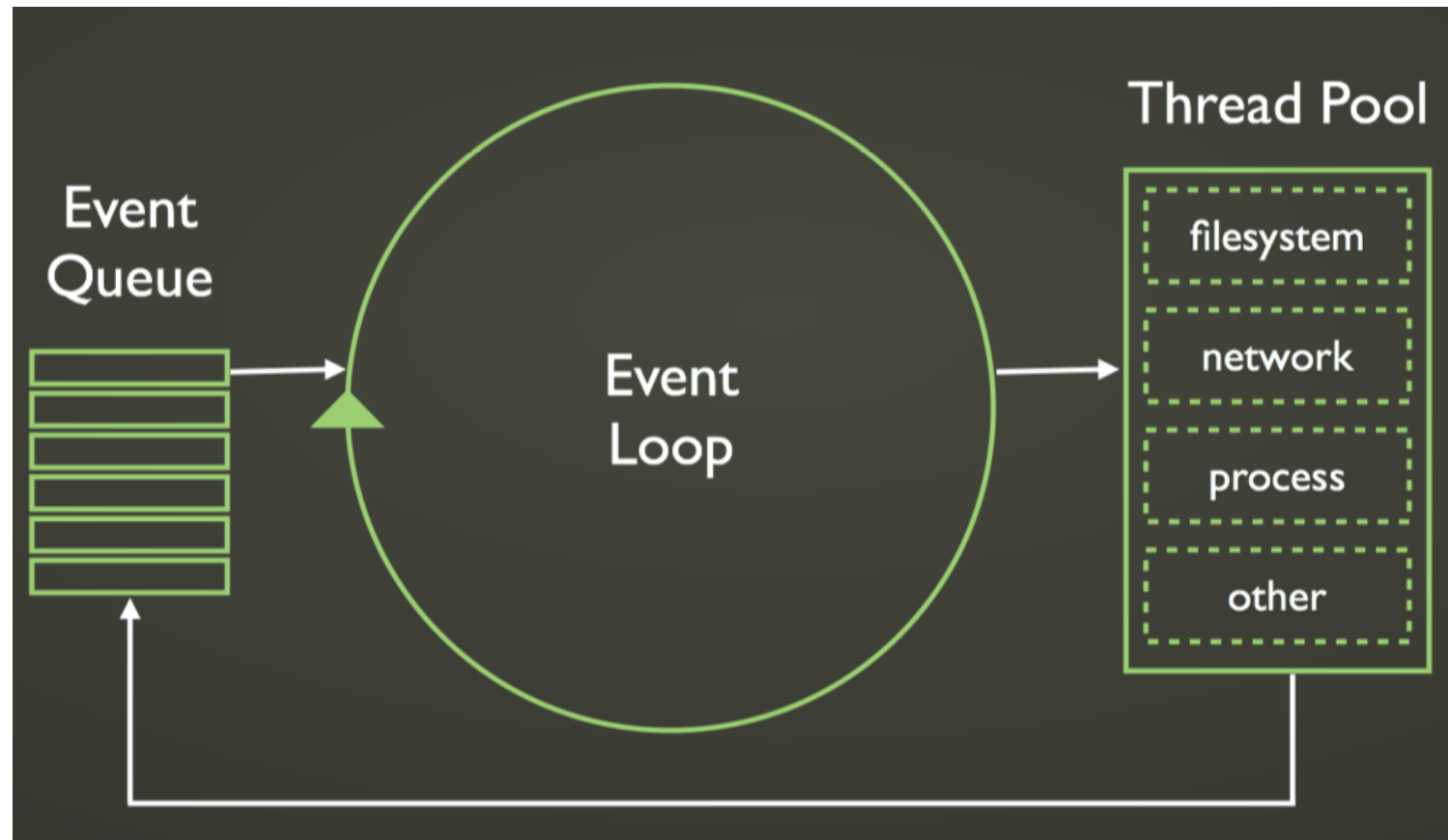


Single thread
Non-blocking IO

Multi thread
Blocking IO



1.4 Non-blocking event loop





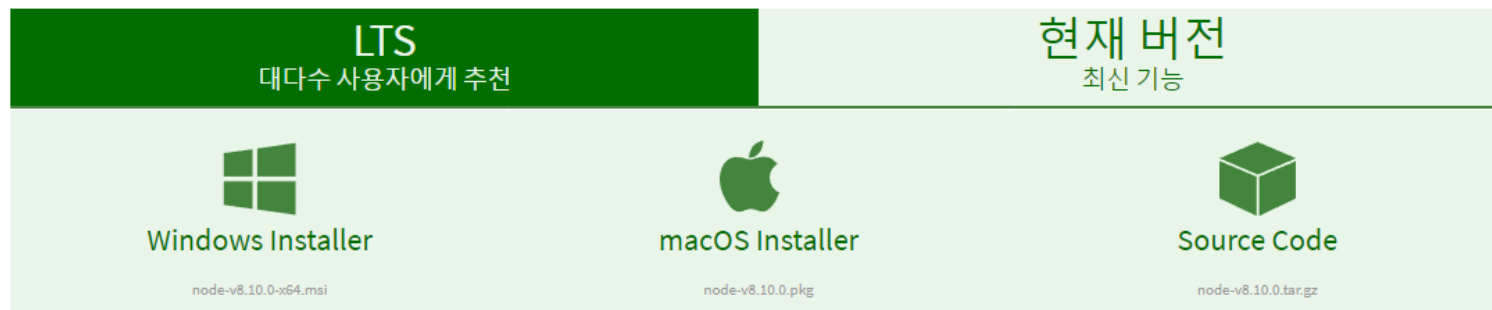
2.1 Install node.js



다운로드

최신 LTS 버전: 8.10.0 (includes npm 5.6.0)

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



Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x86/x64)

Linux Binaries (ARM)

Source Code

32-bit		64-bit	
32-bit		64-bit	
64-bit			
64-bit			
32-bit		64-bit	
ARMv6	ARMv7		ARMv8
node-v8.10.0.tar.gz			

<https://nodejs.org/ko>



2.2 Install node.js portable






<https://gareth.flowers/nodejs-portable/>



2.2 Install node.js portable

← → ↻ 🏠 GitHub, Inc. [US] | <https://github.com/garethflowers/nodejs-portable/releases/> 🔍 ☆ 📄 ⓘ

 This repository Search Pull requests Issues Marketplace Explore

 [garethflowers / nodejs-portable](#)  Watch ▾ 8


Code Issues 6 Pull requests 3 Projects 0 Wiki Insights ▾

Releases Tags

Latest release


🔖 v5.7.0
🔗 3d91ba6


NodeJS Portable v5.7.0


 garethflowers released this on 26 Feb 2016 · 2 commits to master since this release

- Updated to NodeJS v5.7.0.

Downloads

 [NodeJSPortable_5.7.0.paf.exe](#)

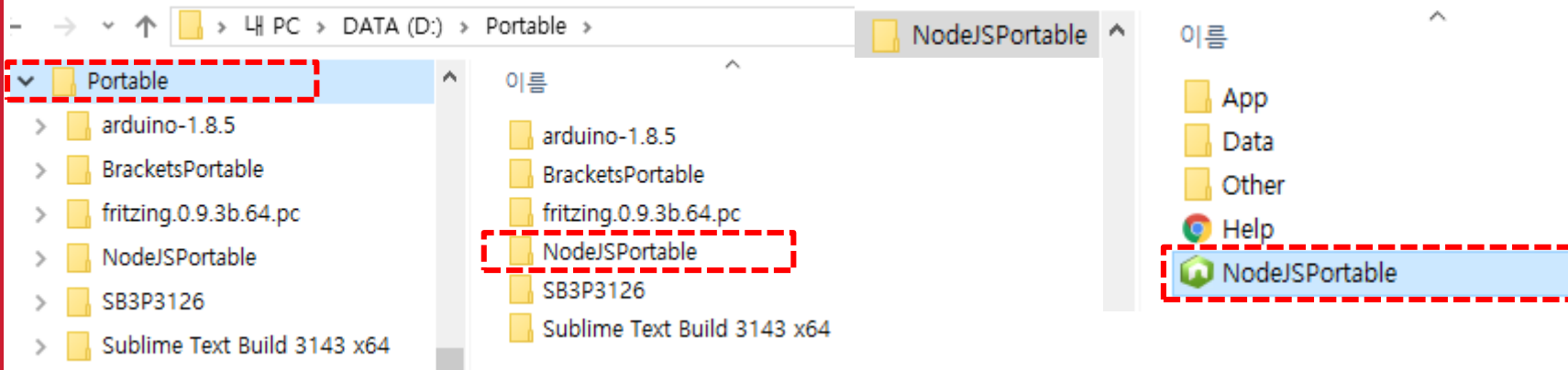
 [Source code \(zip\)](#)

 [Source code \(tar.gz\)](#)

<https://github.com/garethflowers/nodejs-portable/releases/>



2.2 Install node.js portable





2.3.1 Verify installation

NodeJS

```
D:\Portable\NodeJSPortable\Data>node -v  
v5.7.0
```

```
D:\Portable\NodeJSPortable\Data>npm -v  
3.5.3
```

```
D:\Portable\NodeJSPortable\Data>
```

node -v

npm -v



2.3.2 Check node folders.

```
NodeJS
D:\Portable\NodeJSPortable\Data>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data 디렉터리

2017-12-28 오후 01:53 <DIR> .
2017-12-28 오후 01:53 <DIR> ..
2016-02-26 오전 04:36 6,148 .DS_Store
2017-12-28 오후 01:53 <DIR> node_modules
2015-10-15 오전 07:21 296 npm
2015-10-15 오전 07:21 204 npm.cmd
2017-12-28 오후 01:53 89 PortableApps.comLauncherRuntimeData-NodeJSPortable.ini
2017-12-28 오후 01:53 <DIR> settings
2017-12-28 오후 01:53 <DIR> Temp
4개 파일 6,737 바이트
5개 디렉터리 902,995,701,760 바이트 남음

D:\Portable\NodeJSPortable\Data>
```

Data folder



2.4 Testing node.js – node shell

NodeJS

```
D:\Portable\NodeJSPortable\Data>node
> a=2
a=2
2
> b=5
5
> c=a+b
7
> a*b
10
> a/b
0.4
>
(To exit, press ^C again or type .exit)
>
D:\Portable\NodeJSPortable\Data>
```

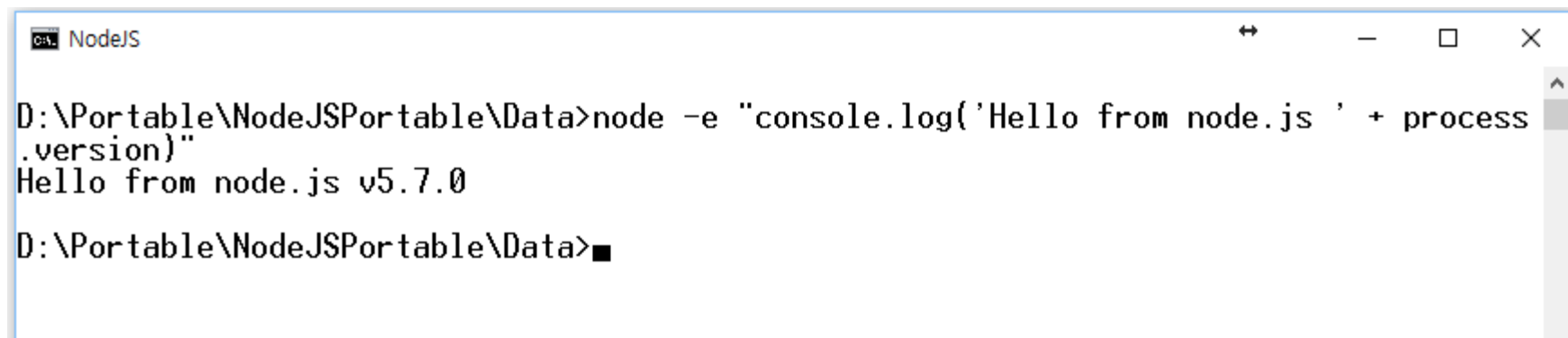
```
// node shell
node
>
>
// exit
^C^C
```



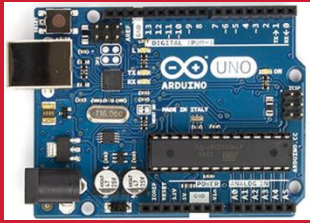

2.5 Testing node.js – final test

Final test if Node.js was installed correctly:

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



```
NodeJS
D:\Portable\NodeJSPortable\Data>node -e "console.log('Hello from node.js ' + process
.version)"
Hello from node.js v5.7.0
D:\Portable\NodeJSPortable\Data>
```



Using node.js

in Sublime text 3

→ Node build system



3.1 Using node.js in Sublime text 3

- Node.js 연결 : Tools > Build System > New Build System

```
{  
    "cmd": ["node.exe path", "$file"],  
    "selector": "source.js"  
}
```

Node.sublime-build 로 저장

D:\Portable\Sublime Text Build 3143 x64\Data\Packages\User\Node.sublime-build (hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

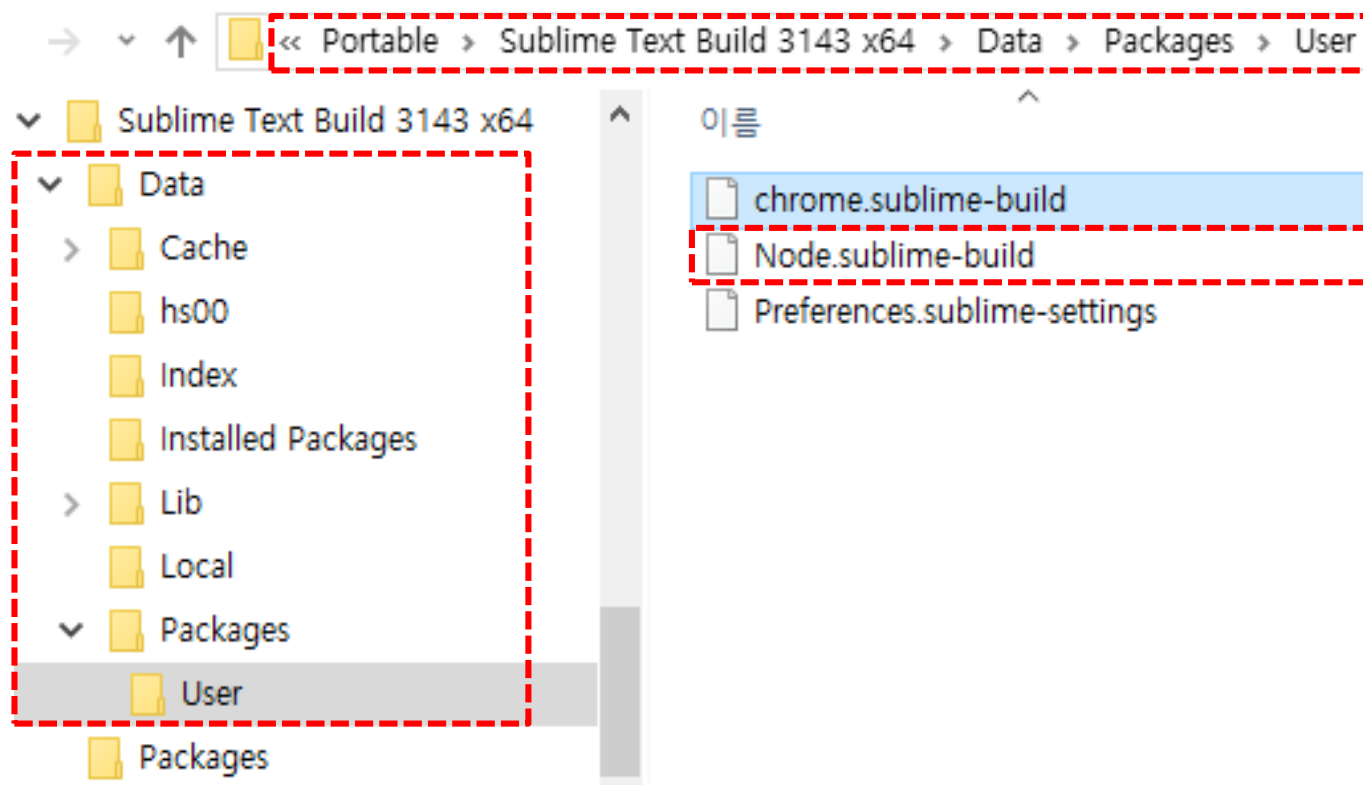
FOLDERS

hs00
hs00.html

```
1 {  
2   "cmd": ["D:\\Portable\\NodeJSPortable\\App\\NodeJS\\node.exe", "$file"],  
3   "selector": "source.js"  
4 }
```

3.2 Using node.js in Sublime text 3

● Node.js 설정 파일 저장 폴더 확인





3.3 Using node.js in Sublime text 3

● Node.js 연결 : Check node from Build System

D:\Portable\Sublime Text Build 3143 x64\Data\Packages\User\Node.sublime-build (hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

hs00
hs00.html

Node.

1 {
2
3
4 }
5

Command Palette... Ctrl+Shift+P

Snippets...

Build System >

Build Ctrl+B

Build With... Ctrl+Shift+B

Cancel Build Ctrl+Break

Build Results >

☒ Save All on Build

Record Macro Ctrl+Q

Playback Macro Ctrl+Shift+Q

Save Macro...

Macros >

Developer >

Install Package Control...

Automatic

Ant

C++ Single File

Cargo

☒ chrome

D

Erlang

Haskell

JavaC

Lua

Make

Node

Python

R

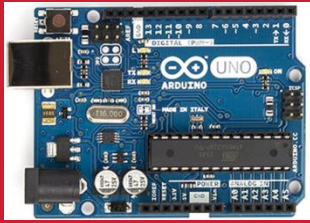
Ruby

Rust

Syntax Tests

New Build System...

```
able\\App\\NodeJS\\node.exe", "$file"]],
```



Node.js Project

npm init



4.1 Node project : md hsnn

NodeJS

```
D:\Portable\NodeJSPortable\Data>md hs00
```

```
D:\Portable\NodeJSPortable\Data>cd hs00
```

```
D:\Portable\NodeJSPortable\Data\hs00>dir
```

D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00 디렉터리

```
2018-03-14 오후 01:05 <DIR>      .
2018-03-14 오후 01:05 <DIR>      ..
                0개 파일                0 바이트
                2개 디렉터리  890,091,855,872 바이트 남음
```

```
D:\Portable\NodeJSPortable\Data\hs00>md start
```

```
D:\Portable\NodeJSPortable\Data\hs00>cd start
```

```
D:\Portable\NodeJSPortable\Data\hs00\start>dir
```

D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00\start 디렉터리

```
2018-03-14 오후 01:11 <DIR>      .
2018-03-14 오후 01:11 <DIR>      ..
                0개 파일                0 바이트
                2개 디렉터리  890,091,855,872 바이트 남음
```

```
D:\Portable\NodeJSPortable\Data\hs00\start>
```

// node cmd

md hs00

cd hs00



4.2 Node project : **npm init**

cmd npm

```
D:\Portable\NodeJSPortable\Data\hs00\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 json' for definitive documentation on these fields
and exactly what they do.
```

```
Use 'npm install <pkg> --save' afterwards to install a package and
save it as a dependency in the package.json file.
```

```
Press ^C at any time to quit.
```

```
name: (start)
```

```
version: (1.0.0)
```

```
description: start node project
```

```
entry point: (index.js)
```

```
test command:
```

```
git repository:
```

```
keywords: test
```

```
author: hs00
```

```
license: (ISC) MIT
```

```
About to write to D:\Portable\NodeJSPortable\Data\hs00\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": "hs00",
  "license": "MIT"
}
```

```
Is this ok? (yes) █
```

// node project

npm init



4.3.1 Node project : package.json

```
NodeJS
D:\Portable\NodeJSPortable\Data\hs00\start>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00\start 디렉터리

2018-03-14 오후 01:21 <DIR> .
2018-03-14 오후 01:21 <DIR>
2018-03-14 오후 01:21      255 package.json
                1개 파일
                2개 디렉터리  890,091,794,432 바이트 남음

D:\Portable\NodeJSPortable\Data\hs00\start>
```



4.3.2 Node project : package.json

package.json

type package.json

Use "cat" when
"type" does not
work

```
NodeJS
D:\Portable\NodeJSPortable\Data\hs00\start>type 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": "hs00",
  "license": "MIT"
}
D:\Portable\NodeJSPortable\Data\hs00\start>■
```

Save as
HSnn_package.png



4.4 Node project : package.json

Sublime Text 3

Project > Add Folder to Project

Select folder: **NodeJSPortable/Data/hs00**

D:\Portable\NodeJSPortable\Data\hs00\start\package.json (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

hs00

<> hs00.html

hs00

start

/* package.json

package.json

```
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": "hs00",
13  "license": "MIT"
14 }
```



5. Node Apps

Node Apps



5.1.1 Hello Node! – hello.js

The screenshot shows the Sublime Text editor interface. On the left, a 'FOLDERS' sidebar displays a project structure with folders 'hs00' and 'start'. A context menu is open over the 'start' folder, with 'New File' selected. The main editor window shows the file 'D:\Portable\NodeJS\Portable\Data\hs00\start\hello.js' being edited. The code contains two lines: a comment and a console log statement. Below the editor, the terminal output shows the successful execution of the script.

```
1 {  
2 "  
3 "  
New File  
Rename...  
New Folder...  
Delete Folder  
Find in Folder...  
FOLDERS  
hs00  
  hs00.html  
  hs00  
    start  
      /* hello.js  
      /* package.json  
File Edit Selection Find View Goto Tools Project Preferences Help  
hello.js package.json  
1 // hsnn's first node app  
2 console.log("Hello Node! by hs00");  
Hello Node! by hs00  
[Finished in 0.3s]
```

Build system을 Node로 지정 후 ^B로 실행



5.1.2 Hello Node! – “node hello.js”

Node cmd에서 “node hello” 로 실행

```
NodeJS
D:\Portable\NodeJSPortable\Data\hs00\start>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00\start 디렉터리

2018-03-14 오후 01:43 <DIR> .
2018-03-14 오후 01:43 <DIR> ..
2018-03-14 오후 01:43 61 hello.js
2018-03-14 오후 01:21 255 package.json
                2개 파일                316 바이트
                2개 디렉터리 890,091,540,480 바이트 남음

D:\Portable\NodeJSPortable\Data\hs00\start>node hello
Hello Node! by hs00

D:\Portable\NodeJSPortable\Data\hs00\start>
```

**// node cmd
node hello.js**



5.1.3 Hello Node! – npm start 로 실행

D:\Portable\NodeJSPortable\Data\hs00\start\package.json • (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- hs00
 - hs00.html
 - hs00
 - start
 - hello.js
 - package.json

```
1 {
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  "author": "hs00",
14  "license": "MIT"
15 }
```

D:\Portable\NodeJSPortable\Data\hs00\start>npm start

> start@1.0.0 start D:\Portable\NodeJSPortable\Data\hs00\start
> node hello.js

Hello Node! by hs00

D:\Portable\NodeJSPortable\Data\hs00\start>



Using function in node



5.2.1 Using function in node

D:\Portable\NodeJS\Portable\Data\hs00\start\hello_function.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- ▼ hs00
 - <> hs00.html
- ▼ hs00
 - ▼ start
 - /* hello.js
 - /* **hello_function.js**
 - /* package.json

```
1 // hsn's first node app
2 console.log("Hello Node! by hs00");
3
4 function hello(what) {
5     console.log("Hello " + what + "!");
6 }
7
8 hello("HCit");
9 hello("hs00");
```

```
Hello Node! by hs00
Hello HCit!
Hello hs00!
[Finished in 0.2s]
```



5.2.2 Using module in node



```
1 // hello_module.js
2 var util = require('util');
3
4 function hello(what) {
5     util.print("Hello " + what + "!\n");
6 }
7
8 hello("HCit");
9 hello("hs00");
```

```
(node) util.print is deprecated. Use console.log instead.
Hello HCit!
Hello hs00!
[Finished in 0.2s]
```

```
// Using module in node.
var util = require('util');
```

https://nodejs.org/api/util.html#util_util_print



5.2.3 Using local module – hello_mymodule.js

D:\Portable\NodeJS\Portable\Data\hs00\start\hello_mymodule.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- hs00
 - hs00
 - start
 - hello.js
 - hello_function.js
 - hello_module.js
 - hello_mymodule.js
 - package.json

```
1 // hello_mymodule.js
2
3 module.exports = function(what) {
4     console.log("Hello " + what + "!");
5 }
6
```

사용자 정의 모듈

FOLDERS

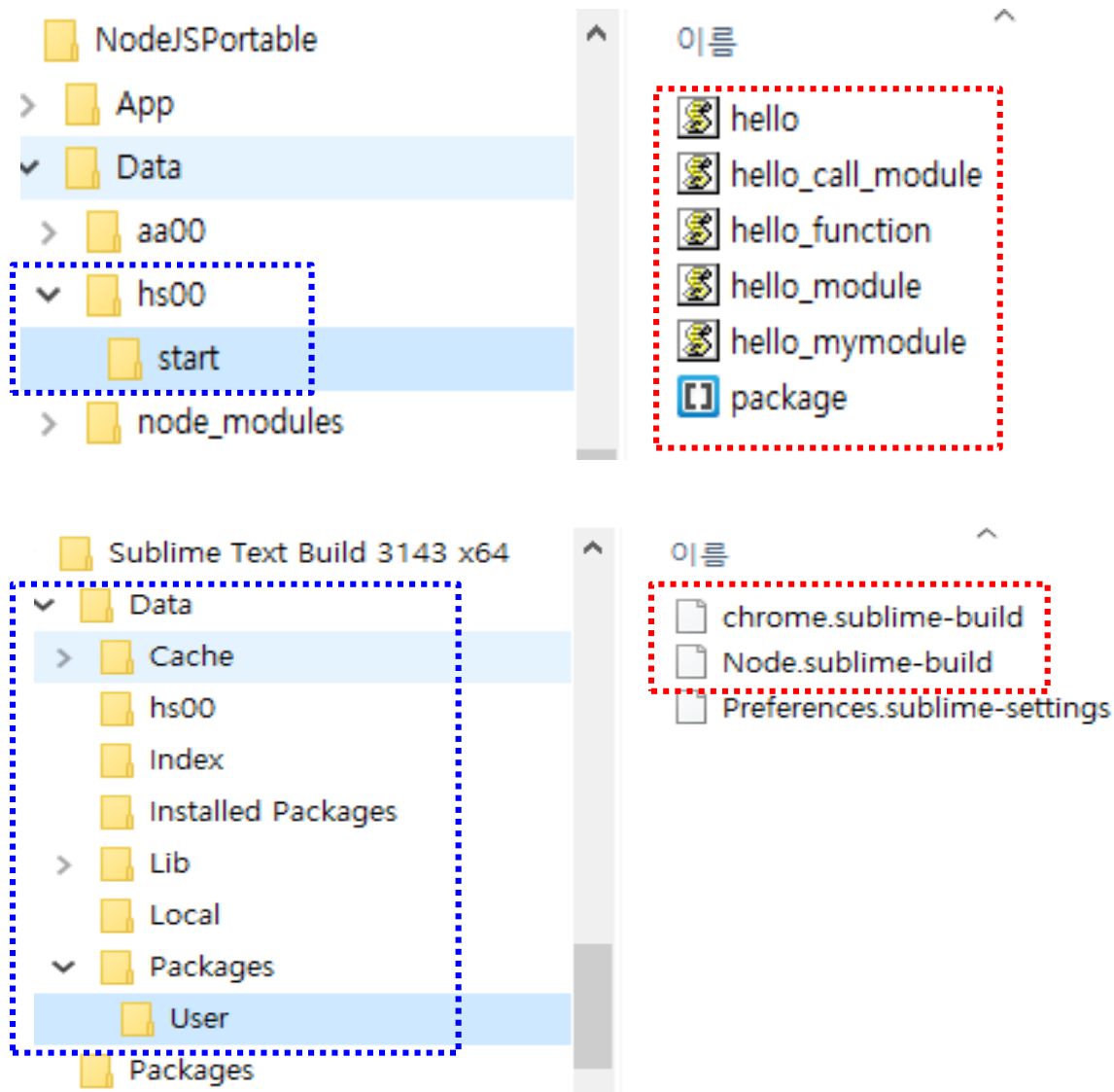
- hs00
 - hs00
 - start
 - hello.js
 - hello_call_module.js
 - hello_function.js
 - hello_module.js
 - hello_mymodule.js
 - package.json

```
1 // hello_call_module.js
2
3 var olleh = require('./hello_mymodule');
4
5 olleh("HCit");
6 olleh("hs00");
```

Hello HCit!
Hello hs00!
[Finished in 0.2s]

Call local module 'hello_mymodule.js' from hello_call_module.js

Checking – Portable working folders



The image shows two Windows File Explorer windows side-by-side, illustrating the portable working folders for Node.js and Sublime Text.

Top Window (Node.js Portable):

- Left pane: Shows the folder structure. The **Data** folder is expanded, and the **start** folder is selected. A blue dashed box highlights the **start** folder.
- Right pane: Shows the contents of the **start** folder. The files listed are **hello**, **hello_call_module**, **hello_function**, **hello_module**, **hello_mymodule**, and **package**. A red dashed box highlights these files.

Bottom Window (Sublime Text Build 3143 x64):

- Left pane: Shows the folder structure. The **Data** folder is expanded, and the **User** folder is selected. A blue dashed box highlights the **User** folder.
- Right pane: Shows the contents of the **User** folder. The files listed are **chrome.sublime-build**, **Node.sublime-build**, and **Preferences.sublime-settings**. A red dashed box highlights these files.



[extra code] local module : **circle.js**

circle_info.js uses local module **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 };
```

circle.js



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

circle_info.js

```
The area of a circle of radius 4 is 50.27
The circumference of a circle of radius 4 is 25.13
[Finished in 0.2s]
```



Node.js Server

1. http, tcp, file

2. Express



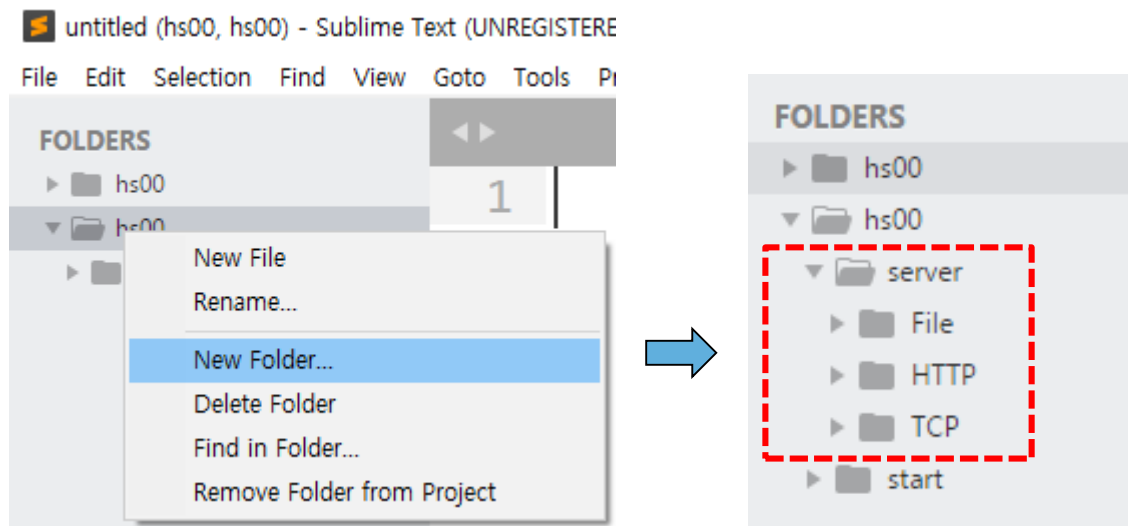
6. Node Server

Node Server I.

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



6.1 Node server : working folders





6.1.1 http server

D:\Portable\Node\SPortable\Data\hs00\server\HTTP\index.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- ▶ hs00
- ▼ hs00
 - ▶ server
 - ▶ File
 - ▼ HTTP
 - /* index.js
 - ▶ TCP
 - ▶ start

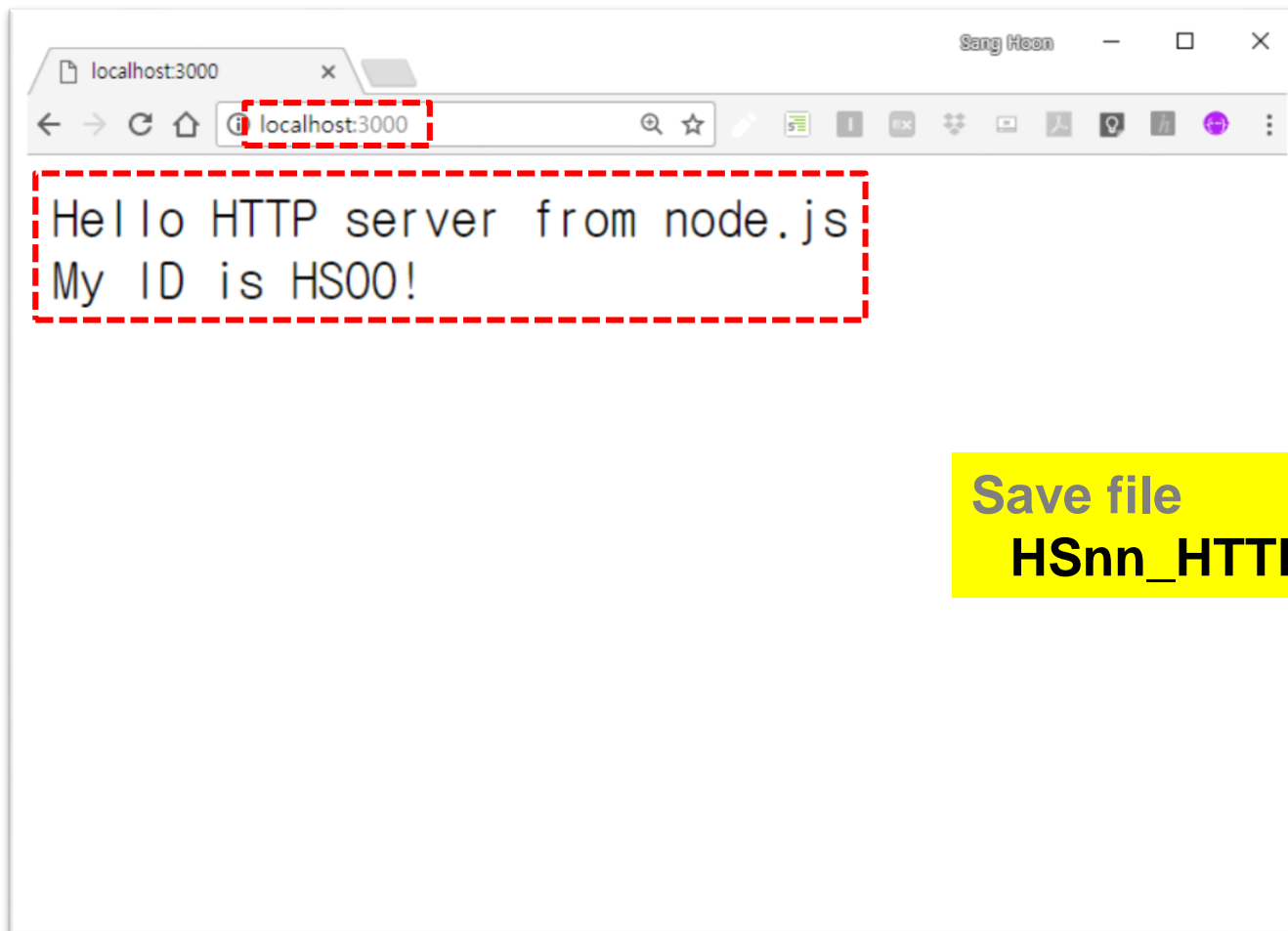
```
index.js x
1 // http server : index.js
2
3 var http = require('http');
4 port = 3000;
5
6 var server = http.createServer(function(request, response) {
7     response.writeHead(200, {
8         "Content-Type": "text/plain"
9     });
10    response.write("Hello HTTP server from node.js"); // WEB response
11    response.write("\nMy ID is HS00!");
12    response.end();
13
14 });
15
16 server.listen(port);
17 console.log("Server Running on " + port +
18     ".\nLaunch http://localhost:" + port);
19
```

```
Server Running on 3000.
Launch http://localhost:3000
```



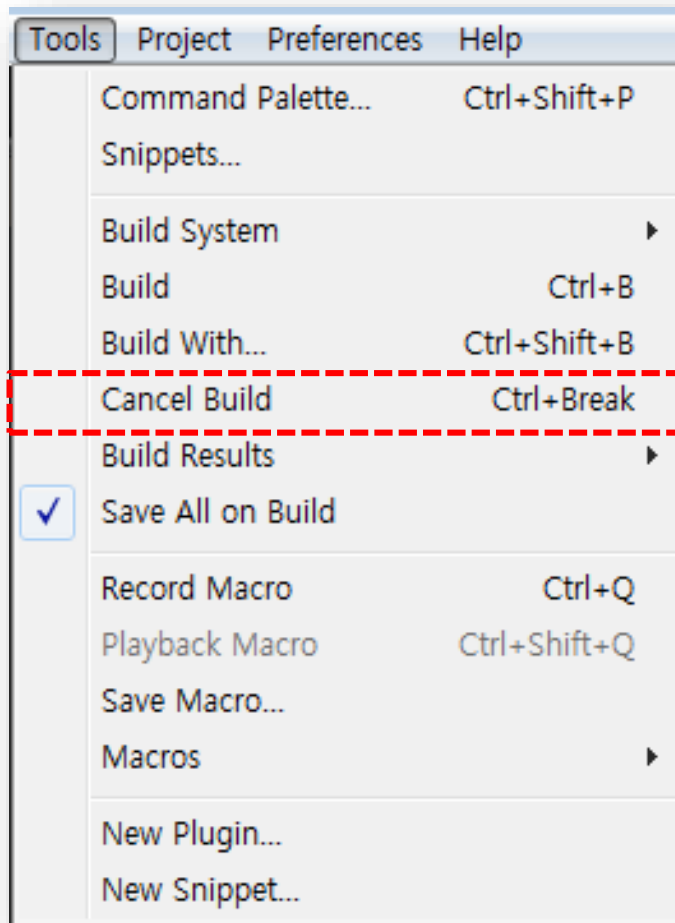
6.1.2 http server : result 1

Server Running on 3000.
Launch `http://localhost:3000`



Save file
HSnn_HTTP.png

6.1.3 http server – stop



```
Server Running on 3000.  
Launch http://localhost:3000  
[Cancelled]
```

[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\NodeJSPortable\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\NodeJSPortable\Data>taskkill /pid 14332
성공: 프로세스(PID 14332)에 종료 신호를 보냈습니다.
```

```
D:\Portable\NodeJSPortable\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)

D:\Portable\NodeJS\Portable\Data\hs00\server\TCP\server.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- ▶ hs00
- ▼ hs00
 - ▶ server
 - ▶ File
 - ▶ HTTP
 - /* index.js
 - ▼ TCP
 - /* client.js
 - /* server.js
 - ▶ start

```
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 HS00! from localhost:3000");
9 });
10
11 server.listen(port, "127.0.0.1");
12 console.log("Network server started at port : " + port);
13
```

Network server started at port : 3000



6.2.2 tcp server [Node cmd : node server.js]

NodeJS

D:\Portable\NodeJSPortable\Data\hs00\start>cd ..

D:\Portable\NodeJSPortable\Data\hs00>dir

D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00 디렉터리

```
2018-03-14 오후 02:57 <DIR> .
2018-03-14 오후 02:57 <DIR> ..
2018-03-14 오후 02:57 <DIR> server
2018-03-14 오후 02:52 <DIR> start
                0개 파일                0 바이트
                4개 디렉터리  890,091,003,904 바이트 남음
```

D:\Portable\NodeJSPortable\Data\hs00>cd server

D:\Portable\NodeJSPortable\Data\hs00\server>dir

D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00\server 디렉터리

```
2018-03-14 오후 02:57 <DIR> .
2018-03-14 오후 02:57 <DIR> ..
2018-03-14 오후 03:00 <DIR> File
2018-03-14 오후 03:00 <DIR> HTTP
2018-03-14 오후 03:00 <DIR> TCP
                0개 파일                0 바이트
                5개 디렉터리  890,091,003,904 바이트 남음
```

D:\Portable\NodeJSPortable\Data\hs00\server>cd tcp

D:\Portable\NodeJSPortable\Data\hs00\server\TCP>dir

D 드라이브의 볼륨: DATA
볼륨 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\hs00\server\TCP 디렉터리

```
2018-03-14 오후 03:00 <DIR> .
2018-03-14 오후 03:00 <DIR> ..
2017-12-29 오후 04:37      415 client.js
2018-03-14 오후 03:17      372 server.js
                2개 파일                787 바이트
                2개 디렉터리  890,091,003,904 바이트 남음
```

**Node cmd에서 TCP
서버 실행**

TCP 폴더로 가서

node server

D:\Portable\NodeJSPortable\Data\hs00\server\TCP>node server
Network server started at port : 3000



6.2.3 tcp client

D:\Portable\NodeJS\Portable\Data\hs00\server\TCP\client.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- ▶ hs00
- ▼ hs00
 - ▶ server
 - ▶ File
 - ▼ HTTP
 - /* index.js
 - ▼ TCP
 - /* client.js
 - /* server.js
 - ▶ start

```
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
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
```

```
Data: Hello HS00! from localhost:3000
Connection closed
[Finished in 0.3s]
```



6.2.4 tcp server 정지 (^C) 후 재 실행!

NodeJS - node server

```
D:\Portable\NodeJSPortable\Data\hs00\server\TCP>node server
Network server started at port : 3000
Connection from 127.0.0.1
^C
D:\Portable\NodeJSPortable\Data\hs00\server\TCP>node server
Network server started at port : 3000
```



6.2.5 tcp server & client : result

(A)

NodeJS - node server

```
D:\Portable\NodeJSPortable\Data\hs00\server\TCP>node server  
Network server started at port : 3000
```

(B)

```
Data: Hello HS00! from localhost:3000  
Connection closed  
[Finished in 0.3s]
```

**Sublime
Text 3**

(C)

NodeJS - node server

```
D:\Portable\NodeJSPortable\Data\hs00\server\TCP>node server  
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  
Connection from 127.0.0.1
```

Save file

HSnn_TCP_Log.png



6.3.1 file upload using module 'formidable'

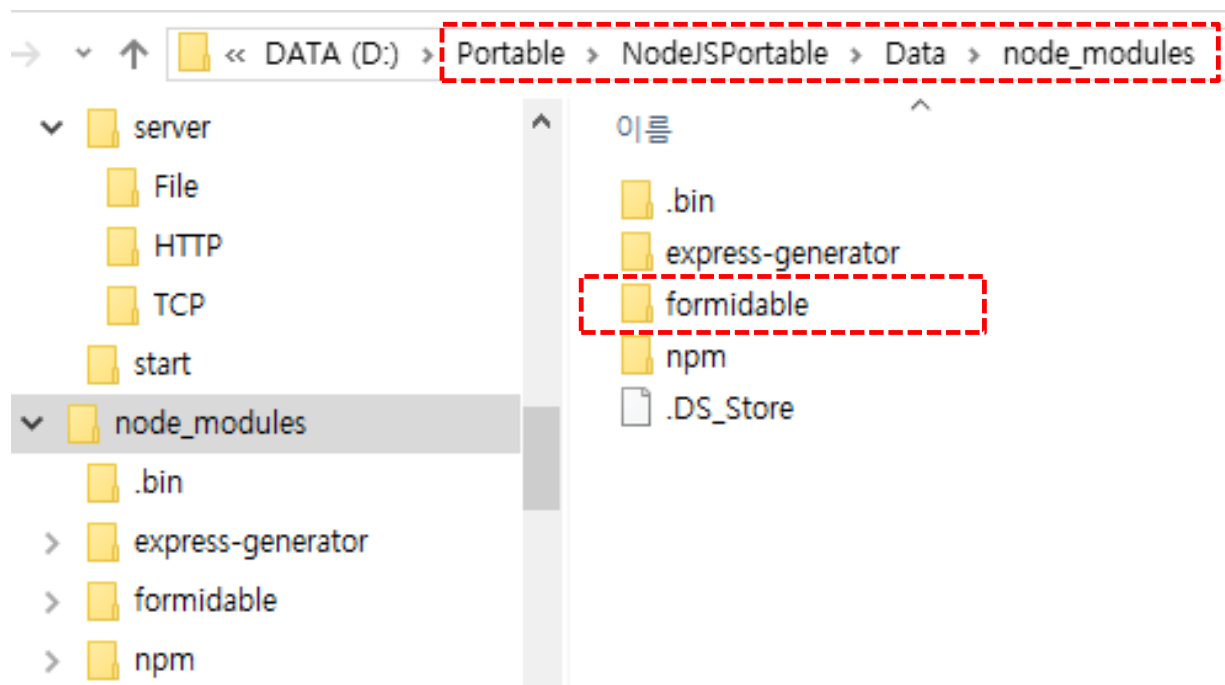
FOLDERS

- ▶ hs00
- ▼ hs00
 - ▼ server
 - ▼ File
 - /* file_server.js
 - ▼ HTTP
 - /* index.js
 - ▼ TCP
 - /* client.js
 - /* server.js
 - ▶ start

```
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.createServer(function(req, res) {
8   if (req.url == '/upload' && req.method.toLowerCase() == 'post') {
9     // parse a file upload
10    var form = new formidable.IncomingForm();
11
12    form.parse(req, function(err, fields, files) {
13      res.writeHead(200, {'content-type': 'text/plain'});
14      res.write('received upload:\n\n');
15      res.end(util.inspect({fields: fields, files: files}));
16    });
17    return;
18  }
19  // show a file upload form
20  res.writeHead(200, {'content-type': 'text/html'});
21  res.end(
22    '<form action="/upload" enctype="multipart/form-data" method="post">
23    <input type="text" name="title"><br>'+
```

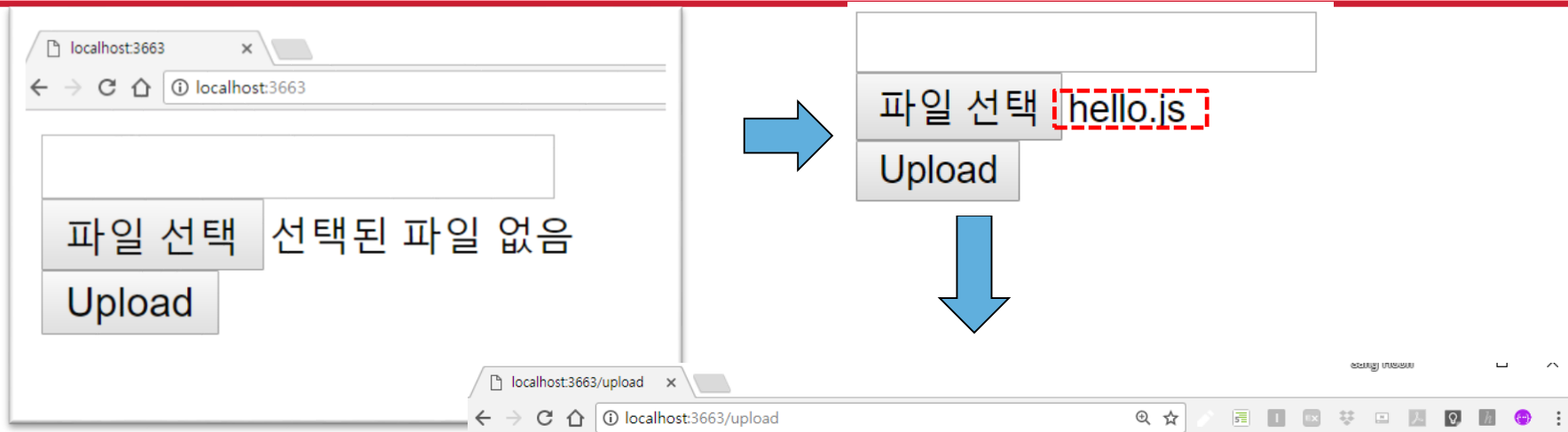
File server Running on 3663.
Launch <http://localhost:3663>

6.3.3 file upload : npm install formidable



**Portable node.js에서는
외부 node module이
Data/node_modules에 저장된다.**

6.3.4 file upload



received upload:

```
{ fields: { title: '' },
  files:
    { upload:
      File {
        domain: null,
        _events: {},
        _eventsCount: 0,
        _maxListeners: undefined,
        size: 61,
        path: 'C:\\Users\\WWW\\AppData\\Local\\Temp\\upload_e48bb8ec25d9dc3c0330080fc3a1b6a4\\',
        name: 'hello.js',
        type: 'application/javascript',
        hash: null,
        lastModifiedDate: Wed Mar 14 2018 15:53:38 GMT+0900 (KST),
        _writeStream: [Object] } } }
```

Save file
HSnn_Upload.png



[Practice]

◆ [wk02]

- Node module : **hsnninfo.js**
- Upload file name : **HSnn_Rpt01.zip**



[practice] local module : **hsnninfo.js**

index_hsnn.js uses local module **hsnninfo.js** in start subfolder.

D:\Portable\NodeJS\Portable\Data\hs00\start\index_hs00.js (hs00, hs00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- hs00
 - hs00
 - server
 - start
 - circle.js
 - circle_info.js
 - hello.js
 - hello_call_module.js
 - hello_function.js
 - hello_module.js
 - hello_mymodule.js
 - hs00info.js
 - index_hs00.js**
 - package.json

```
1 // index_hsnn.js
2
3 var myinfo = require('./hs00info');
4
5 myinfo("hs00", "Redwoods", '010-1234-5678');
6
7 myinfo("hs77", "HCit", '010-5678-1234');
```

My Info
ID : hs00
Name : Redwoods
Phone : 010-1234-5678

My Info
ID : hs77
Name : HCit
Phone : 010-5678-1234

[Finished in 0.2s]

Save as
HSnn_info.png



[practice] local module : hsninfo.js

How to make hsninfo.js in start subfolder.

1. Make local module – hsninfo.js
2. Call hsninfo.js from index_hsn.js.
3. Capture your result.

```
hs00info.js  x  index_hs00.js  x
1 // hsninfo.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 – hsnninfo.js

제출파일명 : **HSnn_Rpt01.zip**

- 압축할 파일들

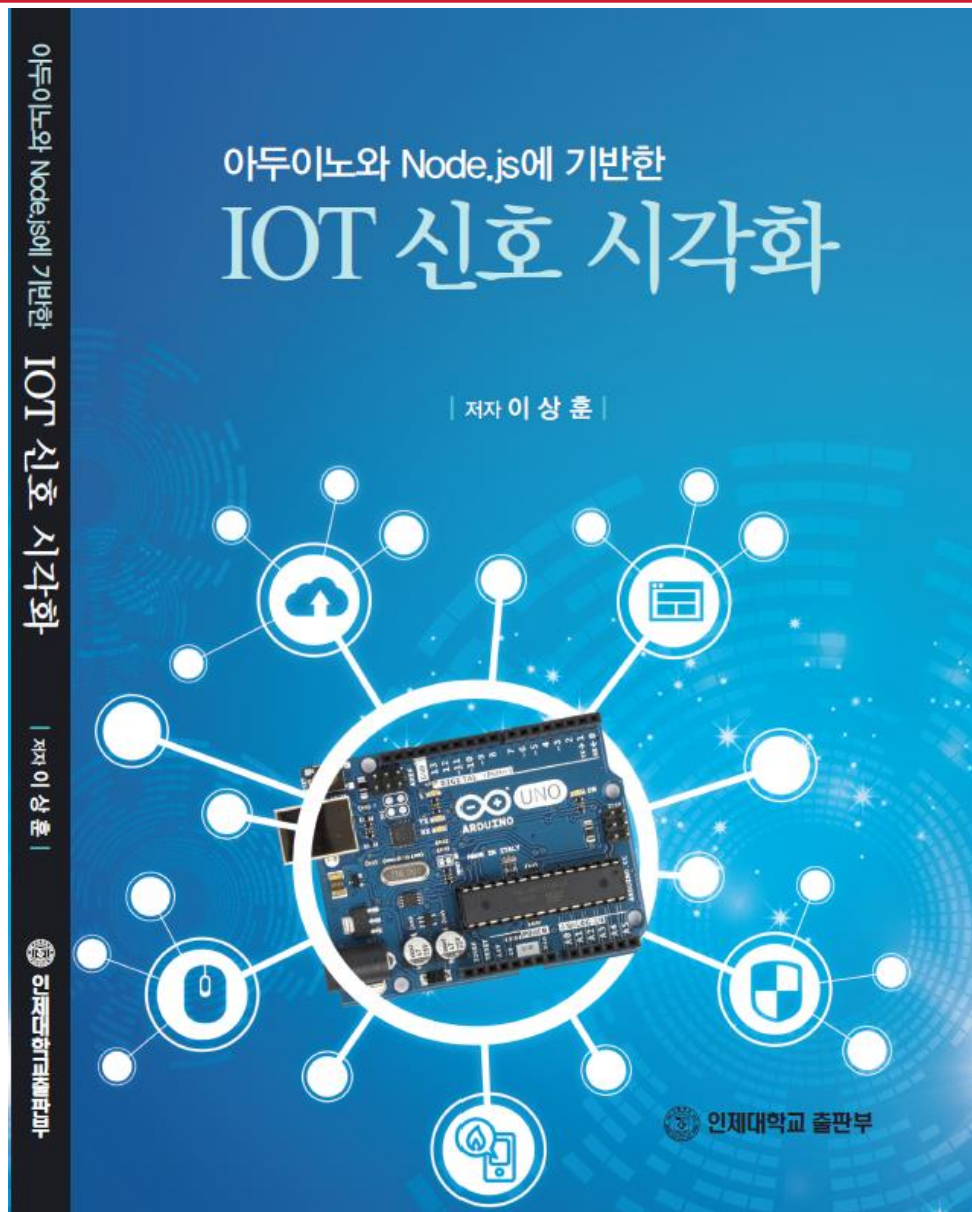
- ① **HSnn_package.png**
- ② **HSnn_HTTP.png**
- ③ **HSnn_TCP_Log.png**
- ④ **HSnn_Upload.png**
- ⑤ **HSnn_info.png**

● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



Syllabus of HS – 주교재 (인쇄 중 ..)

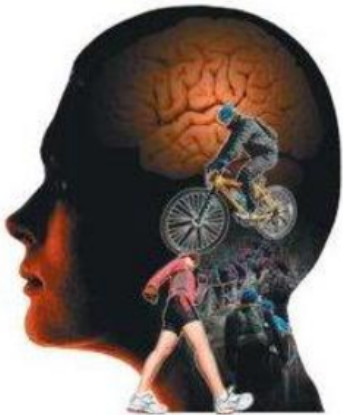




Features Business Explore Marketplace Pricing

Search GitHub

Sign in or Sign up



Redwoods Yi

Redwoods

Block or report user

📍 GimHae, Republic of Korea

Overview

Repositories 5

Stars 2

Followers 0

Following 0

Pinned repositories

dht22-iot-project

lot project to monitor data streaming from DHT22 wired at Arduino.

● HTML

Lec

All lectures by Redwoods in Inje University

arduino-nodejs-plotly-streaming

This repo introduces a simple and efficient way to plot the streaming data from Arduino with Easy Pulse ppg sensor or DHT11 sensor.

● HTML

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)

● Arduino



Redwoods / Lec

Unwatch 1

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

All lectures by Redwoods in Inje University

[Add topics](#)

81 commits

1 branch

0 releases

Branch: master

New pull request

Create new file

Upload files



Redwoods 2018 wk01 upload

Lat

advanced-Arduino-iot

wk16 exam upload

ev3

wk16 final exam. answers

healthcare-signal-iot

2018 wk01 upload

html5-basic

2018 wk01 upload

html5-mobile-simulation


wk15 lec upload

Lec.Rproj

2018 wk01 upload

README.md





wk03 upload and fix links


 This repository Search Pull requests Issues Marketplace Explore

Redwoods / Lec Unwatch 1

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master Lec / healthcare-signal-iot / Create new file U

 Redwoods 2018 wk01 upload Latest	
..	
 src	2018 wk01 upload
 README.md	2018 wk01 upload
 wk01_hs_Intro.pptx	2018 wk01 upload

 README.md

Lec : Introductionto Healthcare Signal Visualization

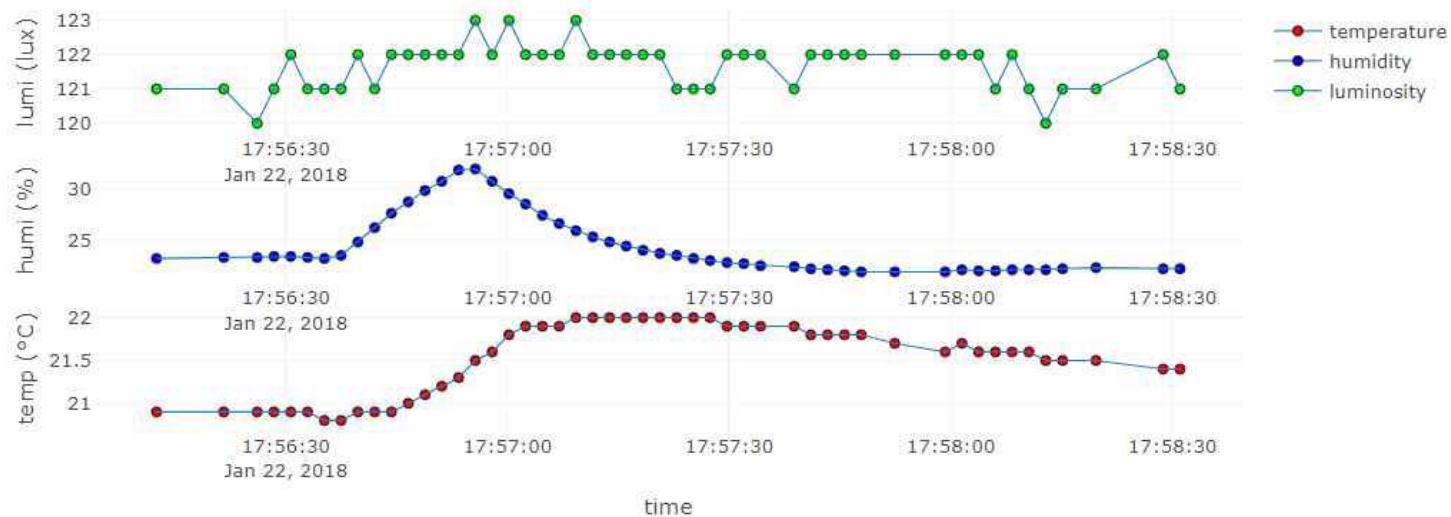
All lectures by Redwoods in Inje University from 2018 and 2017.

Target of this class

Real-time Weather Station from sensors



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



Project of this class

PPG with rangeslider

