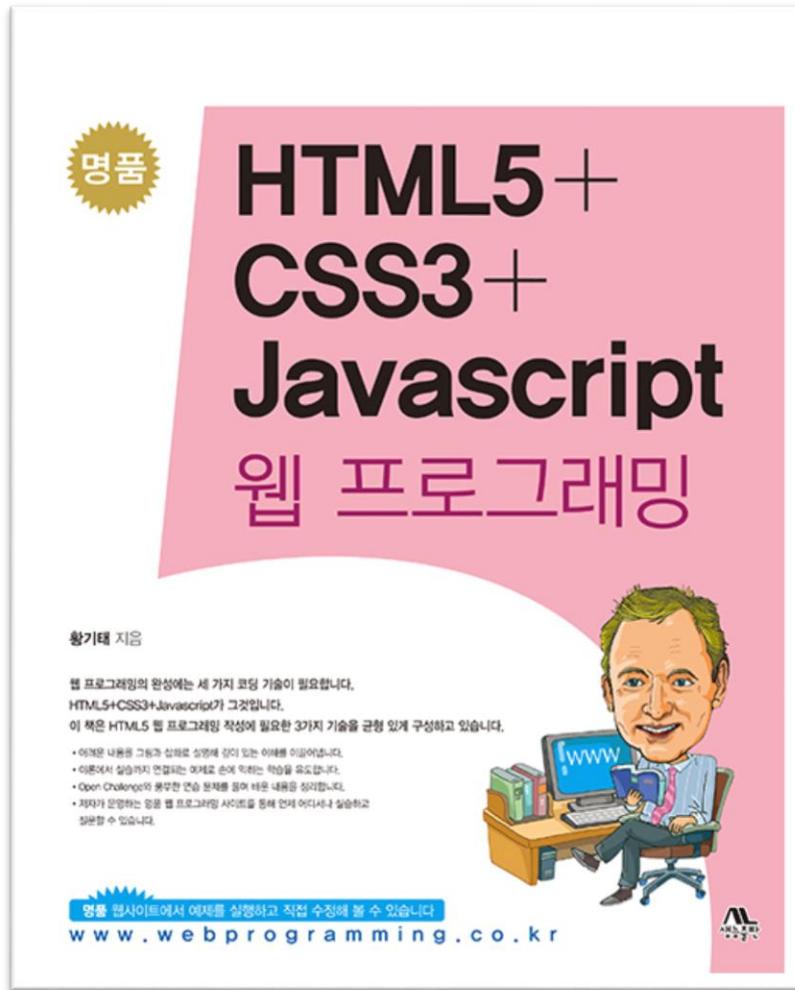


# Mobile Simulation



A screenshot of the w3schools.com website. The search bar at the top says "Search w3schools.com:". Below it is a Google Custom Search bar. The main content area features the w3schools logo and the text "JavaScript". Below this are two buttons: "JavaScript Tutorial" and "JavaScript Reference". Further down, another section for "JQuery" is shown with "JQuery Tutorial" and "JQuery Reference" buttons.



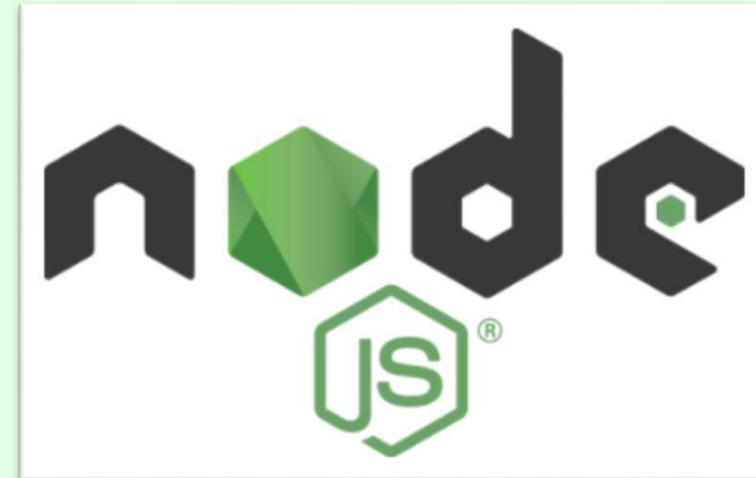
2017-2

# Weekly plan (HTML5, 1<sup>st</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state of HTML5**
- **wk02 : Making HTML5 documents**
- **wk03 : Table, iframe and media**
- **wk04 : Semantic tag and Form**
- **wk05 : CSS3 I. Basic**
- **wk06 : CSS3 II. Advanced**
- **wk07 : CSS3 III. Animation**
- **wk08 : Mid-term Exam.**
- **wk09 : Javascript : Data types & operators**
- **wk10 : Javascript : Loop & functions**
- **wk11 : Javascript : Core objects**
- **wk12 : Javascript : DOM**
- **wk13 : Javascript : Event handling I**
- **wk14 : Javascript : Event handling II**
- **wk15 : Final exam.**

# Weekly plan (Mobile Simulation, 2<sup>nd</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 : Canvas graphics II. Image & Transformation**
- **wk05 : Canvas graphics III. Animation**
- **wk06 : Canvas graphics IV. Game**
- **wk07 : 보강 기간에 보강 실시**
- **wk08 : Mid-term Exam.**
- **wk09 : jQuery I. Basic**
- **wk10 : jQuery II. Application**
- **wk11 : SVG, Drag & Drop**
- **wk12 : Google Map I: Intro**
- **wk13 : Google Map II: Apps**
- **wk14 : JS server - node.js**
- **wk15 : Final exam.**



# 과제10. msnn\_rpt10.zip

4

[실습과제10] Google Map apps

[1] 실습 결과 그림 4장 저장.

[2] [MSnn\\_Transit.png](#), [MSnn\\_geometry.png](#)  
[MSnn\\_MyHome.png](#), [MSnn\\_Flight\\_Stop.png](#)

[3.1] 나의 세계 일주 여행 지도를 만드시오.

[3.2] 꼭 가고 싶은 곳의 위치를 4 곳 이상 찾아서 추가하시오.

[3.3] 구글맵의 다양한 기능을 활용하시오. (아이콘, 애니메이션 변경)

html 파일을 [MSnn\\_Google\\_Travel.html](#) 로 저장하시오.

\*\*\*\* [MSnn\\_Rpt10.zip](#) 으로 압축해서 제출하시오.

[제출파일] [msnn\\_rpt10.zip](#) (11월21일 오후 6시 마감)

html 파일과 사용된 그림을 압축하여 이메일로 “msnn\_rpt10” 제목으로 제출

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)

# Results

5

지도 위성

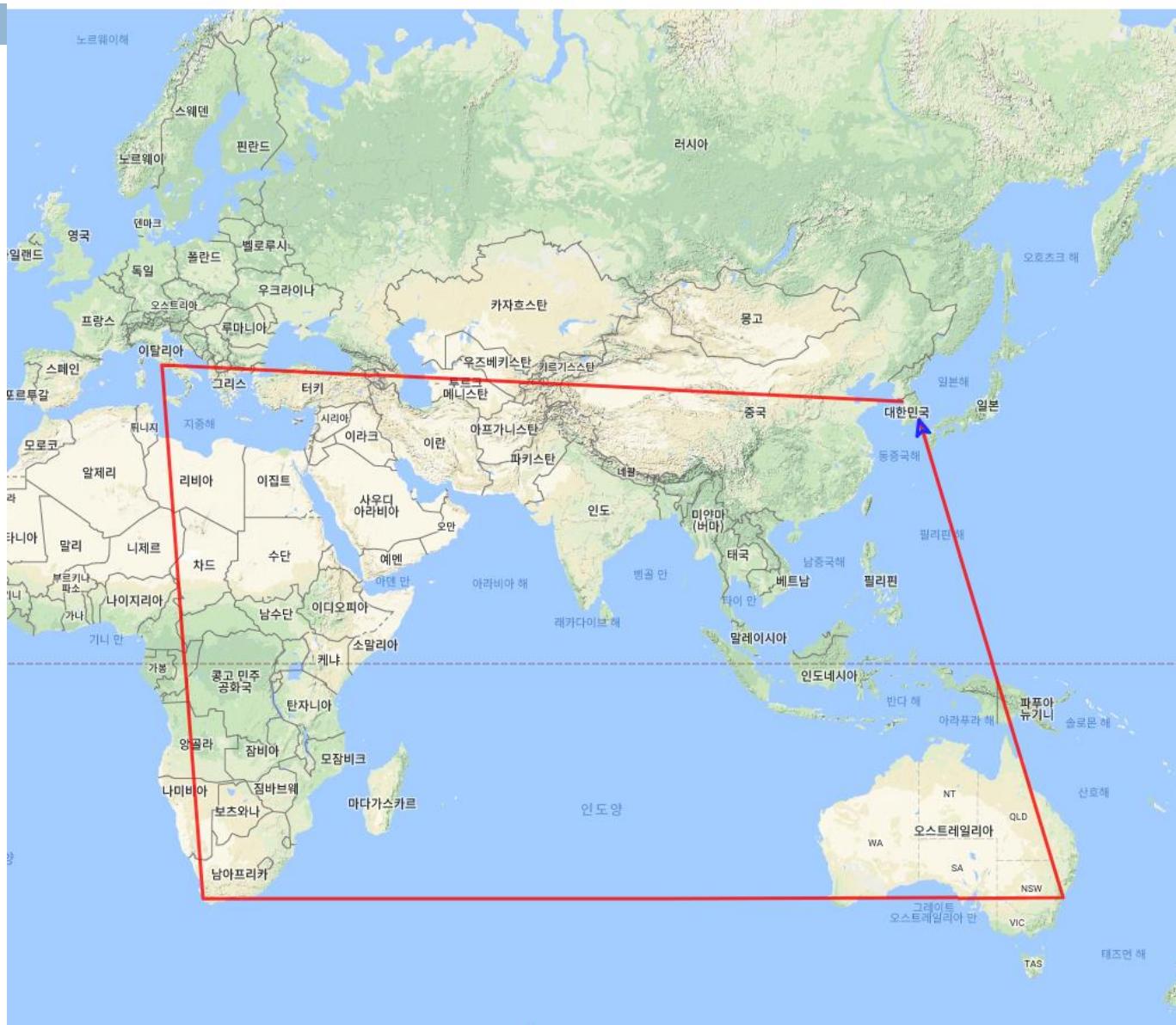
Foods Stores 서점 약국 Coffee shops Bar 의원 치과 미용실

Choose location: MS03 Home ▾

스타벅스 포항양덕점

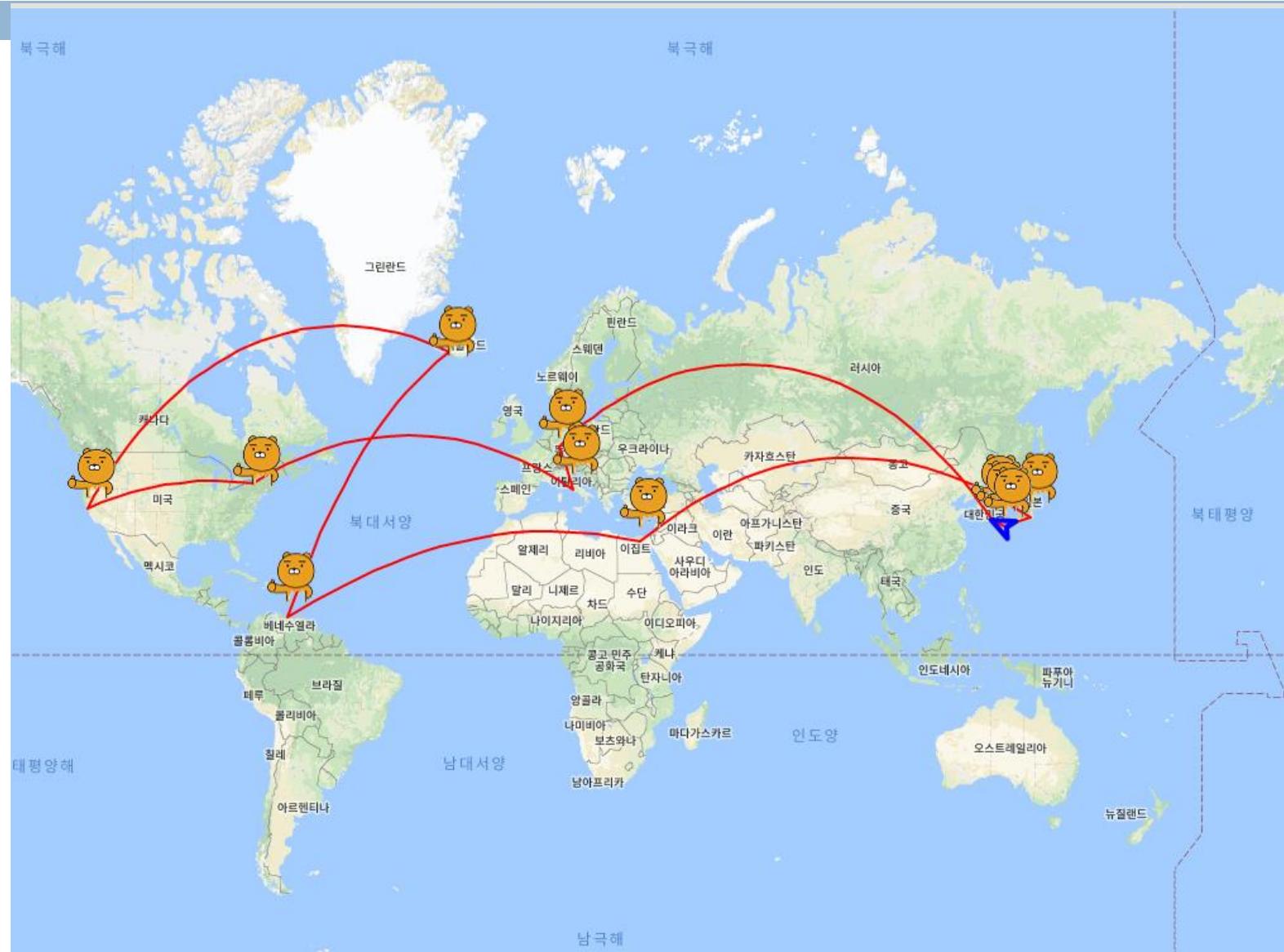
# Results

6



# Results

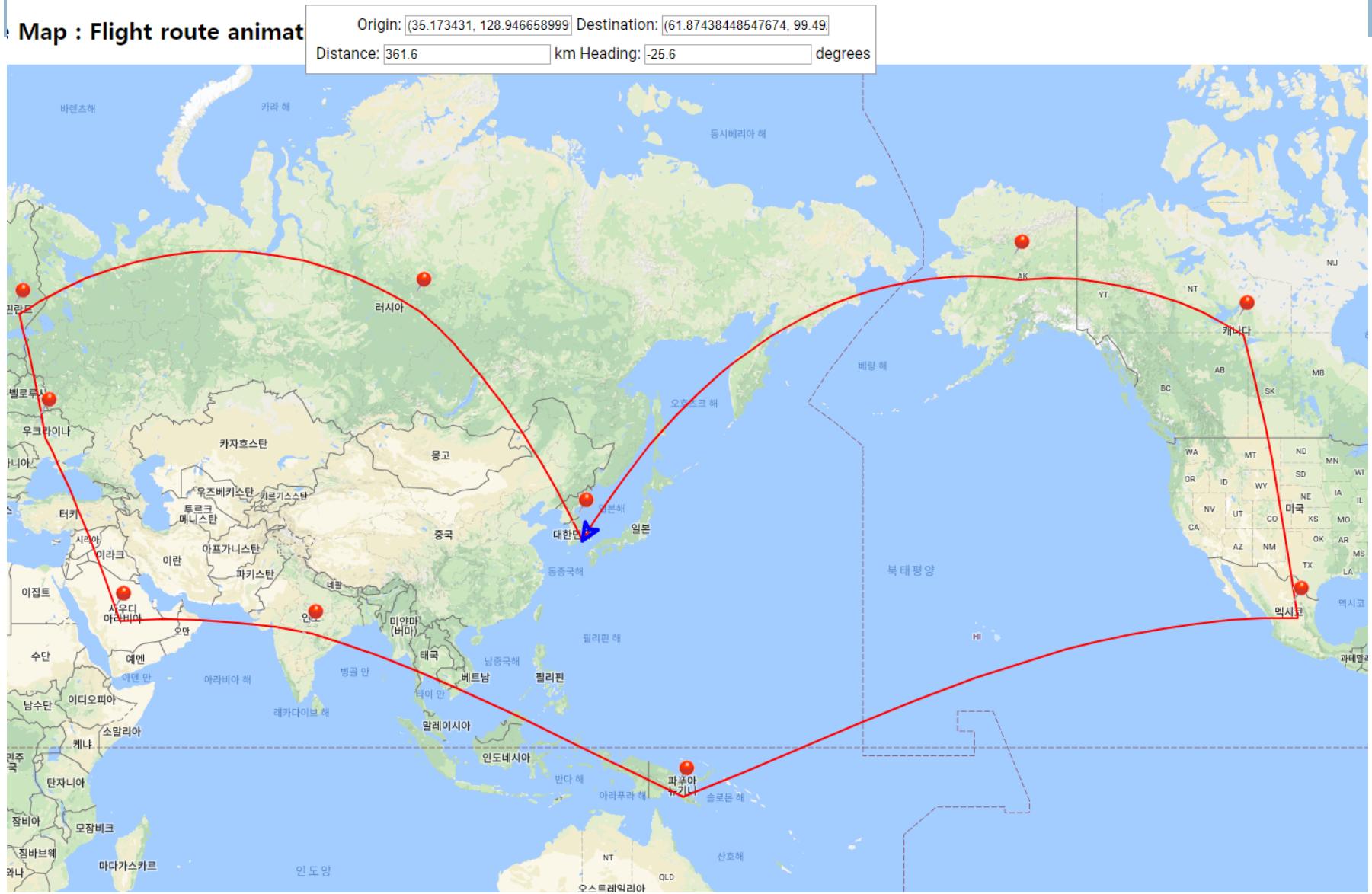
7



# Results

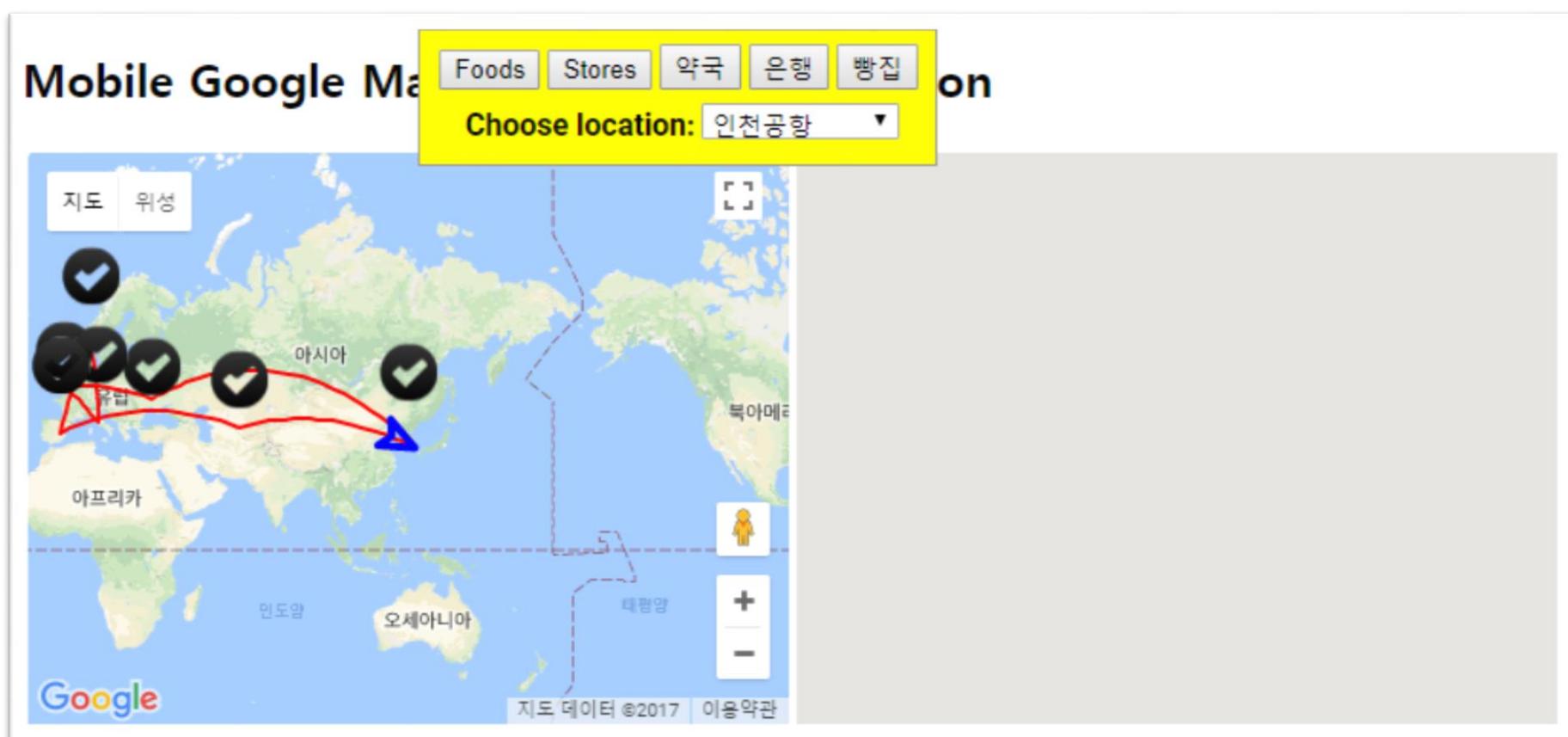
8

Map : Flight route animation



# Results

9



# Node.js



10





# Node.js in Brackets

11

The screenshot shows the Brackets Extension Manager interface. At the top, there are tabs for '설치 가능' (Installable), '테마' (Themes), '설치됨' (Installed) with a count of 3, and 'Default'. A search bar contains the text 'node'. Below the search bar, a button labeled 'Last Updated' with a downward arrow is visible.

The search results list the following extensions:

- Brackets Terminal X** by **ficristo**, version 0.2.0 (2017.03.11). It has 132,190 downloads and a '설치' (Install) button.
- NodeJS integration** by **yacut**, version 1.8.17 (2017.02.22). It has 63,374 downloads and a '설치' (Install) button. This item is highlighted with a red dashed border.
- Brackets SASS** by **Jason San Jose**, version 2.0.5 (2016.12.26). It has 26,890 downloads and a '설치' (Install) button.

Below the extension list, there is a note: '여기에 .zip 파일을 드래그하거나 URL에서 설치하세요...' (Drag your .zip file here or enter the URL to install). A blue '닫기' (Close) button is located in the bottom right corner.



# Node.js in Brackets

12

[yacut / brackets-nodejs-integration](#)

## Brackets - NodeJS integration

stars 26 forks 9 license MIT issues 13 open

Brackets - NodeJS integration is an extension for [Brackets](#) editor - it provides Nodejs and Mocha integration for Brackets. tested and works on any platform supported by Brackets (Windows, Mac OS X, GNU/Linux).

## Installation

### Dependencies:

To make [Brackets - NodeJS integration](#) work you'll need nodejs, npm and mocha installed in your system:

- Windows: [Nodejs](#) and [NPM for Windows](#)
- Mac OS X: [Nodejs](#) and [NPM for Mac](#)
- GNU/Linux: [Nodejs](#) and [NPM for Debian/Ubuntu](#)

### Extension installation:

Use [brackets-npm-registry](#)



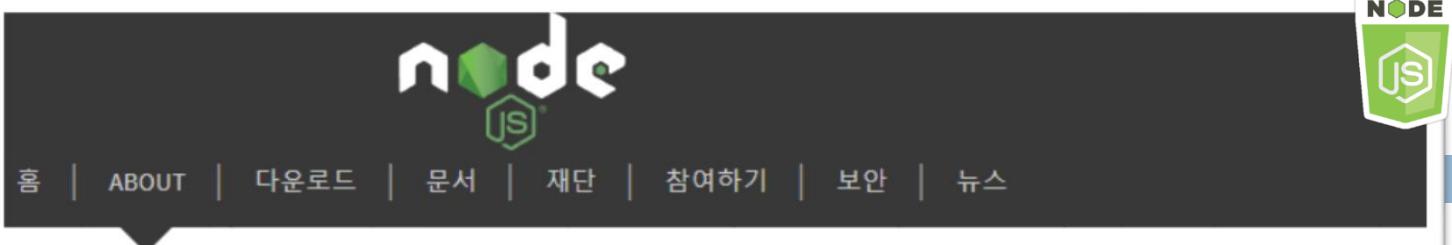
# Javascript on Server

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**.

# Node.js

14



## 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}/`);
});
```



# Node.js

15

- **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.**



# Node.js

16

## 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?

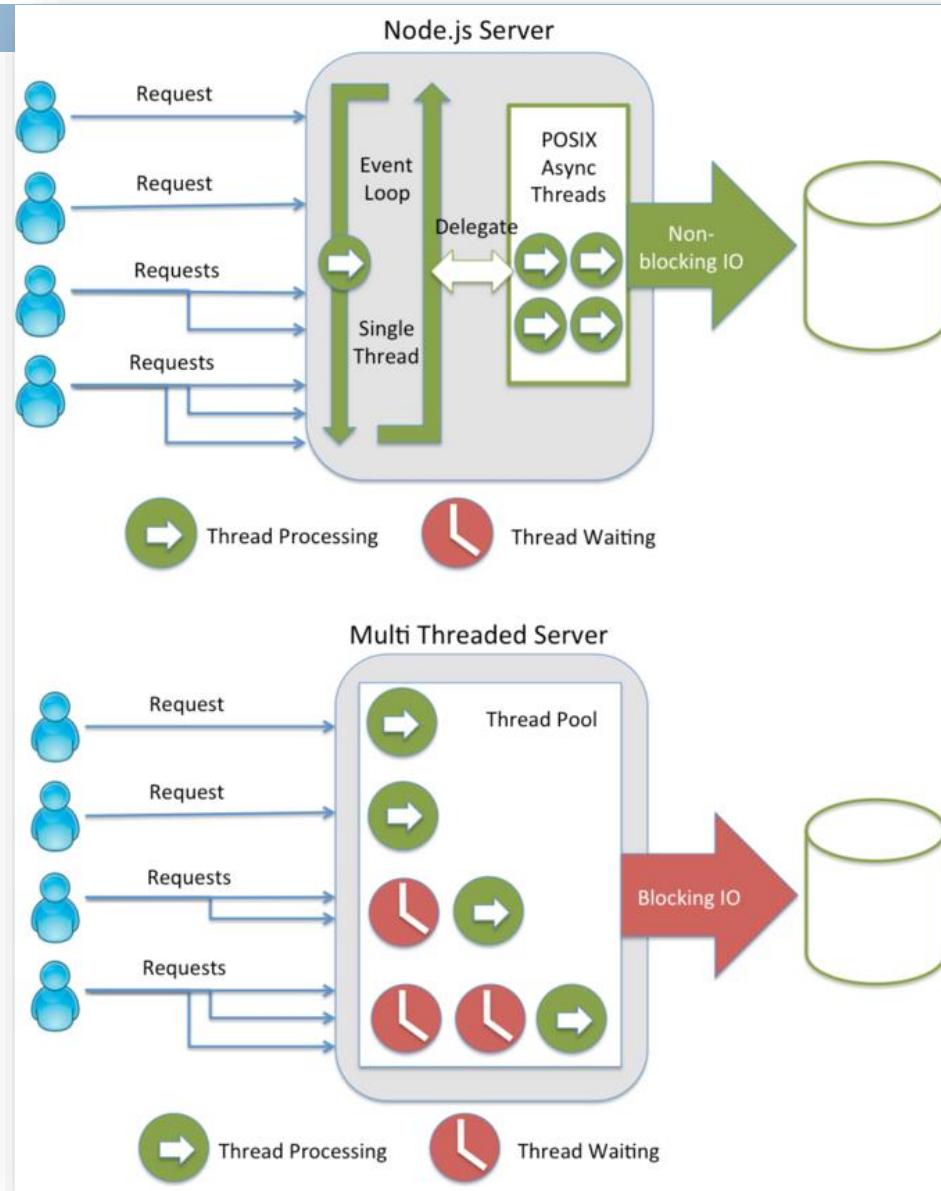
**Blocking : ?**

**Non-blocking : ?**

# Node.js : event loop vs. multi-threads



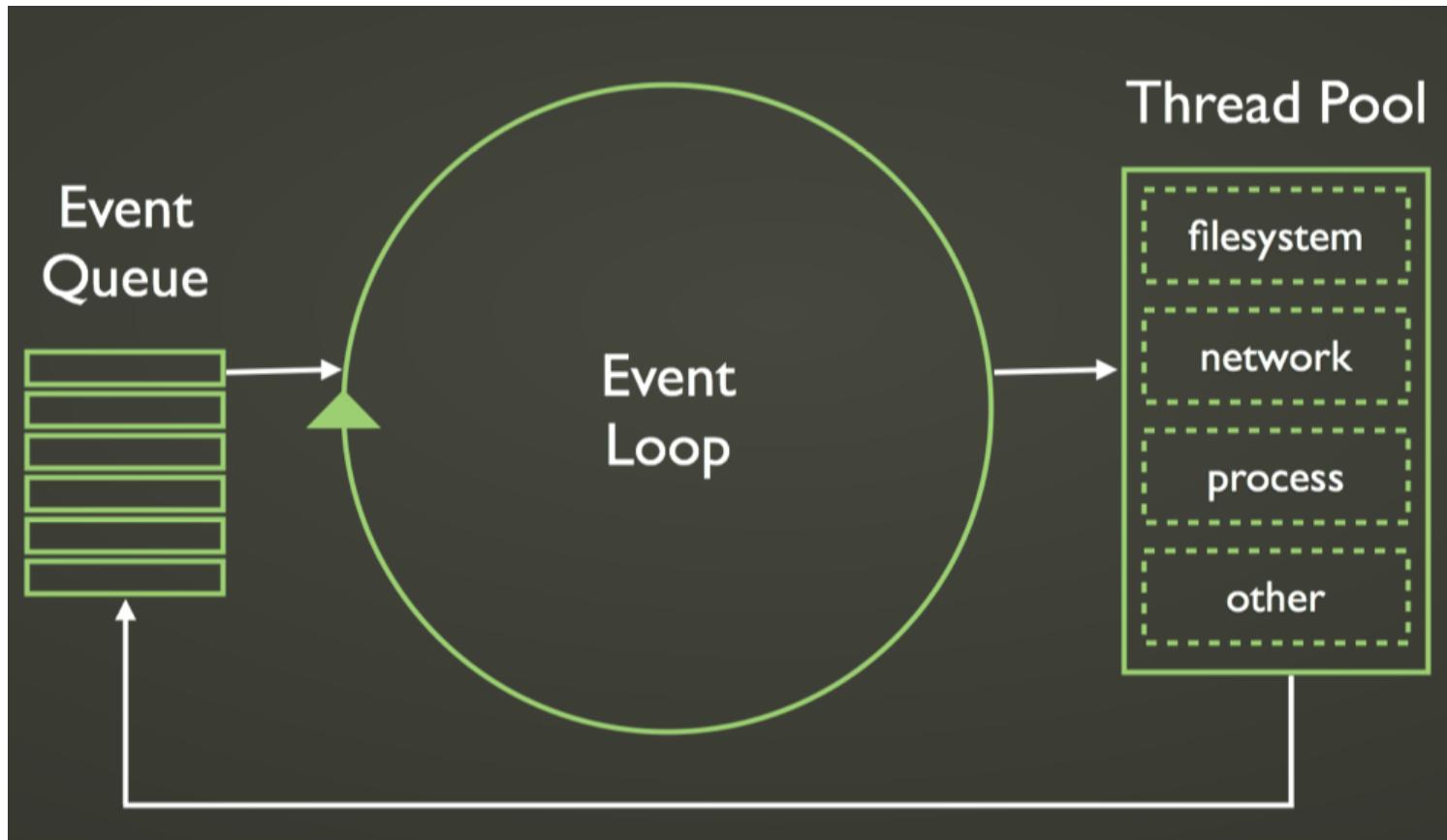
17



# Node.js : event loop



18



# Node.js : download



19

The screenshot shows the Node.js download page. At the top, there's a navigation bar with links for Home, ABOUT, 다운로드 (Download), 문서 (Documentation), 재단 (Foundation), 참여하기 (Participate), 보안 (Security), and 뉴스 (News). Below the navigation, a section titled '다운로드' (Download) is highlighted with a red dashed box. It says '최신 LTS 버전: v6.11.3 (includes npm 3.10.10)'. A message below it reads '플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.' (Download the pre-built Node.js installer or source code for your platform and start developing right away.).

The page is divided into two main sections: 'LTS' (Long Term Support) and '현재 버전' (Current Version). The LTS section is highlighted with a green background and contains links for 'Windows Installer' (with icon), 'Macintosh Installer' (with icon), and 'Source Code' (with icon). The current version section also contains these links.

Below these sections is a large table showing download options for different platforms and architectures. The table has columns for 'Platform' (Windows, Mac OS X, Linux) and 'Architecture' (32-bit, 64-bit). The rows list various file types: Windows Installer (.msi), Windows Binary (.zip), macOS Installer (.pkg), macOS Binaries (.tar.gz), Linux Binaries (x86/x64), Linux Binaries (ARM), and Source Code. The 'Windows Installer (.msi)' row is also highlighted with a red dashed box.

Platform	32-bit	64-bit	
Windows Installer (.msi)			
Windows Binary (.zip)	32-bit	64-bit	
macOS Installer (.pkg)		64-bit	
macOS Binaries (.tar.gz)		64-bit	
Linux Binaries (x86/x64)	32-bit	64-bit	
Linux Binaries (ARM)	ARMv6	ARMv7	ARMv8
Source Code	node-v6.11.3.tar.gz		

<https://nodejs.org/ko>

# Node.js : portable



20



A screenshot of a web browser showing the "NodeJS Portable" website. The URL in the address bar is <https://gareth.flowers/nodejs-portable/>. The page features a large, stylized title "NodeJS Portable". Below the title is a large green hexagonal logo with "JS" inside. To the right of the logo, there is a section titled "Portable version of NodeJS, for Windows" with two buttons: "Download NodeJS Portable" and "View on GitHub". At the bottom left, it says "NodeJS Portable".

Portable version of NodeJS, for Windows

[Download NodeJS Portable](#)

[View on GitHub](#)

NodeJS Portable

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

# Node.js : portable



21

The screenshot shows a GitHub repository page for <https://github.com/garethflowers/nodejs-portable/releases/>. The repository name is `garethflowers / nodejs-portable`. The 'Releases' tab is selected. The latest release is **v5.7.0**, released by garethflowers on 26 Feb 2016. The release notes indicate an update to NodeJS v5.7.0. Downloads available include `NodeJSPortable_5.7.0.paf.exe`, `Source code (zip)`, and `Source code (tar.gz)`.

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

# Node.js : node & npm



22

```
NodeJS  
D:\Portable\NodeJSPortable>node -v  
v5.7.0  
D:\Portable\NodeJSPortable>npm -v  
3.5.3  
D:\Portable\NodeJSPortable>
```

**node -v**

**npm -v**

# Node.js : node root folder



23

```
NodeJS
E:\Coder\NodeJSPortable>Data>dir
E 드라이브의 볼륨: Lecture
볼륨 일련 번호: 2618-C99F

E:\Coder\NodeJSPortable\Data 디렉터리

2017-09-06 오전 09:37 <DIR> .
2017-09-06 오전 09:37 <DIR> ..
2016-02-26 오전 04:36 6,148 .DS_Store
2017-09-06 오전 09:37 <DIR> node_modules
2015-10-15 오전 07:21 296 npm
2015-10-15 오전 07:21 204 npm.cmd
2017-09-06 오전 09:37 89 PortableApps.comLauncherRuntimeData=NodeJSPortabl
e.ini
2017-09-06 오전 09:37 <DIR> settings
        4개 파일 6,737 바이트
        4개 디렉터리 1,114,830,856,192 바이트 남음

E:\Coder\NodeJSPortable>Data>cd ..

E:\Coder\NodeJSPortable>dir
E 드라이브의 볼륨: Lecture
볼륨 일련 번호: 2618-C99F

E:\Coder\NodeJSPortable 디렉터리

2017-09-06 오전 09:37 <DIR> .
2017-09-06 오전 09:37 <DIR> ..
2017-09-06 오전 09:37 <DIR> App
2017-09-06 오전 09:37 <DIR> Data
2015-09-28 오전 07:22 5,003 Help.html
2015-09-28 오전 07:22 148,080 NodeJSPortable.exe
2017-09-06 오전 09:37 <DIR> Other
        2개 파일 153,083 바이트
        5개 디렉터리 1,114,830,856,192 바이트 남음

E:\Coder\NodeJSPortable>
```

**Data folder**

**Node.exe folder**

# Node.js : node shell



24

```
ca NodeJS

D:\Portable\NodeJSPortable>node
> a=2
a=2
2
> b=5
5
> c=a-b
-3
> c
-3
> a/b
0.4
>
(To exit, press ^C again or type .exit)
>

D:\Portable\NodeJSPortable>
```

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

# Node.js : node checkup



25

**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 titled "NodeJS". The window shows the command "node -e "console.log('Hello from node.js ' + process.version)" being run, followed by the output "Hello from node.js v5.7.0".

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

# Node.js : node project



26

NodeJS

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

D:\Portable\NodeJSPortable\Data 디렉터리

2017-11-28 오전 10:11 <DIR> .
2017-11-28 오전 10:11 <DIR> ..
2016-02-26 오전 04:36 <DIR> 6,148 .DS_Store
2017-11-08 오후 01:23 <DIR> aa00
2017-09-13 오후 12:18 express
2017-09-13 오후 12:18 express.cmd
2017-09-15 오후 04:41 <DIR> myApp
2017-09-13 오후 12:18 <DIR> node_modules
2015-10-15 오전 07:21 npm
2015-10-15 오전 07:21 npm.cmd
2017-11-01 오전 11:51 <DIR> npm_cache
2017-11-28 오전 10:09 PortableApps.comLauncherRuntimeData-NodeJSPortabl
e.ini
2017-09-06 오후 01:40 <DIR> settings
2017-11-08 오후 12:26 <DIR> Temp
      6개 파일          7,309 바이트
      8개 디렉터리    905,698,844,672 바이트 남음
```

```
D:\Portable\NodeJSPortable\Data>md ms00
```

```
D:\Portable\NodeJSPortable\Data>cd ms00
```

```
D:\Portable\NodeJSPortable\Data\ms00>md start
```

```
D:\Portable\NodeJSPortable\Data\ms00>cd start
```

```
D:\Portable\NodeJSPortable\Data\ms00\start>
```



# Node.js : npm init

27

npm init

```
cmd: npm
D:\Portable\NodeJSPortable\Data\ms00\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) ms00
ms00
version: (1.0.0)
description: start node
entry point: (index.js)
test command:
git repository:
keywords: node basic
author: ms00
license: (ISC) MIT
About to write to D:\Portable\NodeJSPortable\Data\ms00\start\package.json:

{
  "name": "ms00",
  "version": "1.0.0",
  "description": "start node",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [
    "node",
    "basic"
  ],
  "author": "ms00",
  "license": "MIT"
}

Is this ok? (yes)
```

# Node.js : node project



28

## package.json

type package.json

Use “cat” when  
“type” does not  
work

```
C:\ NodeJS
D:\Portable\NodeJSPortable\Data\ms00\start>dir
D 드라이브의 블루: DATA
블루 일련 번호: 7A01-106A

D:\Portable\NodeJSPortable\Data\ms00\start 디렉터리

2017-11-28 오전 10:40 <DIR> .
2017-11-28 오전 10:40 <DIR> ..
2017-11-28 오전 10:40 259 package.json
                      1개 파일 259 바이트
                      2개 디렉터리 905,698,844,672 바이트 남음

D:\Portable\NodeJSPortable\Data\ms00\start>cat package.json
{
  "name": "ms00",
  "version": "1.0.0",
  "description": "start node",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "node",
    "basic"
  ],
  "author": "ms00",
  "license": "MIT"
}

D:\Portable\NodeJSPortable\Data\ms00\start>
```

# Node.js : node project-package.json



29

start/package.json (ms00) - Brackets

파일 편집 검색 보기 탐색 디버그 도움말 NodeJS

ms00

start

package.json

```
1 - {  
2     "name": "ms00",  
3     "version": "1.0.0",  
4     "description": "start node",  
5     "main": "index.js",  
6     "scripts": {  
7         "test": "echo \\\"Error: no test specified\\\" && exit 1"  
8     },  
9     "keywords": [  
10         "node",  
11         "basic"  
12     ],  
13     "author": "ms00",  
14     "license": "MIT"  
15 }  
16
```

Save as  
**MSnn\_package.png**

# Node.js : node apps



30

# Node Apps

# Node.js : node apps (run in Brackets)



31

The screenshot shows the Brackets IDE interface. On the left, there's a sidebar with Korean file management options like '파일 만들기' (File Create), '폴더 만들기' (Folder Create), etc. The main workspace shows a file named 'hello.js' with the following code:

```
1 // msnn's first node app
2 console.log("Hello node!!!");
```

Below the code editor, the 'start' tab is selected, showing a terminal window with the output of the Node.js command:

```
Command: node "D:/Portable/NodeJSPortable/Data/ms00/basic/hello.js"
Hello node!!!
Program exited with code 0
```

A red dashed box highlights the terminal output. At the bottom of the terminal window, there's a button labeled 'Run current NodeJS file (Ctrl+Shift+N)'.

Run current NodeJS file (Ctrl+Shift+N)

# Node.js : node apps



32

Currently **Brackets - NodeJS integration** supports these features (this list may be incomplete as we add new features regularly):

- Run multiple nodejs processes inside editor with console output (actual limit is only 5 processes, because brackets connections limitation)
- Run current NodeJS file (Ctrl-Shift-N)
- Run current NodeJS project (Ctrl-Shift-P)
- Run current Mocha test (Ctrl-Shift-T)
- Run npm scripts
- Run gulp scripts
- Run mocha test inside editor with tree view results
- Open file via mouse click from error stack
- Show Actual/Expected difference for mocha test
- Double click on mocha test case to open file with it
- *Jump to require* command uses "Jump to declaration" and opens required file (Ctrl-Shift-J or Cmd-Shift-J)
- Code hints for require: actual directory content, NodeJS built-in libs and project packages from package.json (use Ctrl-Space or Cmd-Space while require)
- NodeJS and Mocha debugger integration (based on **TheBenji** work)

# Node.js : node apps (Run in cmd)



33

```
// node cmd  
cd basic  
node hello.js
```

```
cmd NodeJS  
D:\Portable\NodeJSPortable\Data\ms00>dir  
D 드라이브의 볼륨: DATA  
볼륨 일련 번호: 7A01-106A  
  
D:\Portable\NodeJSPortable\Data\ms00 디렉터리  
  
2017-11-28 오전 10:53 <DIR> .  
2017-11-28 오전 10:53 <DIR> ..  
2017-11-28 오전 10:55 <DIR> basic  
2017-11-28 오전 10:52 <DIR> start  
          0개 파일          0 바이트  
          4개 디렉터리  905,698,844,672 바이트 남음  
  
D:\Portable\NodeJSPortable\Data\ms00>cd basic  
  
D:\Portable\NodeJSPortable\Data\ms00\basic>dir  
D 드라이브의 볼륨: DATA  
볼륨 일련 번호: 7A01-106A  
  
D:\Portable\NodeJSPortable\Data\ms00\basic 디렉터리  
  
2017-11-28 오전 10:55 <DIR> .  
2017-11-28 오전 10:55 <DIR> ..  
2017-11-28 오전 10:55          55 hello.js  
          1개 파일          55 바이트  
          2개 디렉터리  905,698,844,672 바이트 남음  
  
D:\Portable\NodeJSPortable\Data\ms00\basic>node hello  
Hello node!!!  
  
D:\Portable\NodeJSPortable\Data\ms00\basic>
```

# Node.js : node apps (Run **npm start**)



34

```
현재 파일
  hello.js
• package.json

ms00 ▾
  ▾ basic
    hello.js
  ▾ start
    hello.js
    package.json
```

```
1 - {
2   "name": "ms00",
3   "version": "1.0.0",
4   "description": "start node",
5   "main": "index.js",
6   "scripts": {
7     "start": "node hello.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [
11    "node",
12    "basic"
13  ],
14  "author": "ms00",
15  "license": "MIT"
16}
17
```

```
D:\Portable\NodeJSPortable\Data\ms00\start>npm start
> ms00@1.0.0 start D:\Portable\NodeJSPortable\Data\ms00\start
> node hello.js
Hello node!!!
```

# Node.js : node apps – using function



35

The screenshot shows the Brackets IDE interface with the following details:

- File Menu:** 파일, 편집, 검색, 보기, 탐색, 디버그, 도움말, NodeJS
- Current File:** hello.js, package.json
- Project Structure:** ms00 / basic / hello.js, hello\_function.js; start / hello.js, package.json
- Code Editor:** basic/hello\_function.js (ms00) - Brackets
- Code Content:**

```
1 // msnn's first node app
2 console.log("Hello node!!!");
3
4 function hello(what) {
5     console.log("Hello " + what + " !");
6 }
7
8 hello("COMSI");
9 hello("ms00");
```

The code editor highlights the function definition from line 4 to line 6 with a red dashed box.
- Terminal:** hello\_function.js
- Output:** Command: node "D:/Portable/NodeJSPortable/Data/ms00/basic/hello\_function.js"  
Hello node!!!  
Hello COMSI!  
Hello ms00!  
Program exited with code 0

# Node.js : node apps – using module



36

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with file navigation. The main area displays a file named `hello_module.js` containing the following code:

```
1 // Using module in node
2 var util = require('util');
3
4 function hello(what) {
5     util.print("Hello " + what +" !");
6 }
7
8 hello("COMSI");
9 hello("ms00");
```

A red dashed box highlights the line `var util = require('util');`. Below the code editor is a terminal window titled `hello_module.js` showing the command and output:

```
Command: node "D:/Portable/NodeJSPortable/Data/ms00/basic/hello_module.js"
Hello COMSI !
Hello ms00 !
Program exited with code 0
```

# Node.js : node apps – user module

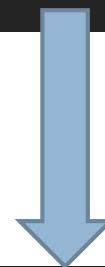


37

```
ms00
└── basic
    ├── hello.js
    ├── hello_function.js
    ├── hello_module.js
    ├── hello_mymodule.js
    └── hello_mymodule_call.js
```

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

Command: node  
Hello COMSI !  
Hello MS00 . !



```
ms00
└── basic
    ├── hello.js
    ├── hello_function.js
    ├── hello_module.js
    ├── hello_mymodule.js
    └── hello_mymodule_call.js
```

```
1 // Using user module in node
2 // hello_mymodule_call.js
3 var olleh = require('./hello_mymodule');
4
5 olleh("COMSI");
6 olleh("MS00");
```

Call user module 'hello\_mymodule.js' from hello\_mymodule\_call.js

# Node.js : node apps - example



38

```
1 // cicle.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 screenshot shows a terminal window with the following content:

- File tabs at the top: circle\_info.js
- Command line: Command: node "D:/Portable/NodeJSPortable/Data/ms00/basic/circle\_info.js"
- Output:
  - The area of a circle of radius 4 is 50.27
  - The circumference of a circle of radius 4 is 25.13
  - Program exited with code 0

# Node.js : node apps - folders



39

내 PC > DATA (D:) > Portable > NodeJSPortal > Data > ms00 > basic			
	이름	수정한 날짜	유형
Portable	circle	2017-11-28 오후...	JavaScript 파일
arduino-1.8.4	circle_info	2017-11-28 오후...	JavaScript 파일
NodeJSPortal	hello	2017-11-28 오전...	JavaScript 파일
App	hello_function	2017-11-28 오전...	JavaScript 파일
Data	hello_module	2017-11-28 오전...	JavaScript 파일
aa00	hello_mymodule	2017-11-28 오전...	JavaScript 파일
ms00	hello_mymodule_call	2017-11-28 오전...	JavaScript 파일
basic			
start			

Save as  
MSnn\_folder.png

# Node.js : node apps - server



40

```
1 // http server : http_server.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 in Brackets"); // 
WEB response
11  response.write("\nMy ID is MS00, COMSI!");
12  response.end();
13
14 });
15
16 server.listen(port);
17 console.log("Server Running on " + port +
18   "\nLaunch http://localhost:" + port);
```

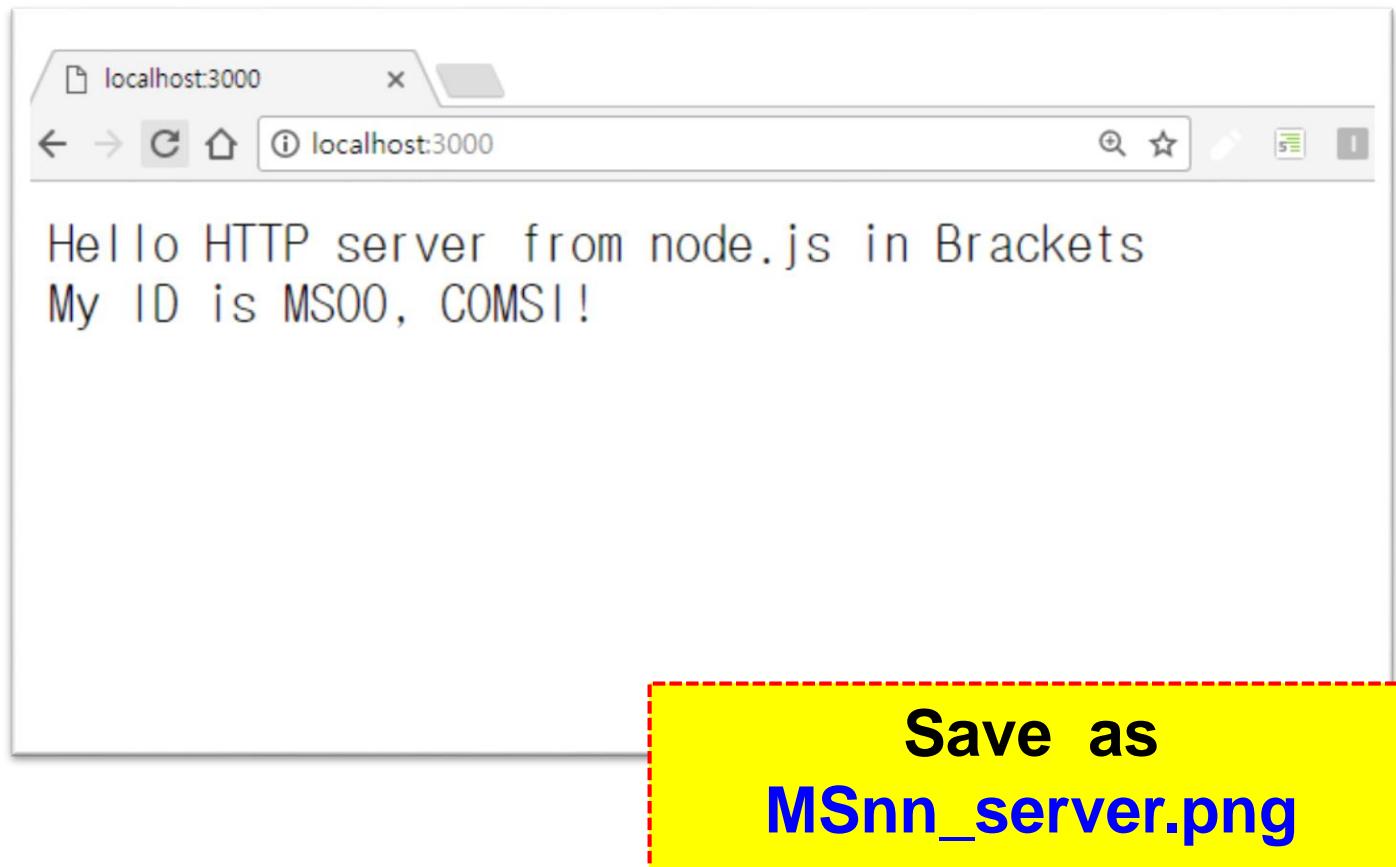
A screenshot of a code editor window. At the top, there are two tabs: one labeled "http\_server.js" with a blue icon and another labeled "http\_server.js" with a green icon. Below the tabs, the code from the previous block is displayed in a monospaced font. At the bottom of the editor, there is a status bar with the text "Command: node D:/Portable/NodeJSPortable/Data/ms00/server/http\_server.js" and "Server Running on 3000".

Command: node D:/Portable/NodeJSPortable/Data/ms00/server/http\_server.js  
Server Running on 3000.  
Launch http://localhost:3000

# Node.js : node apps - server



41



# Node.js : node apps – msnninfo.js



42

- ◆ [DIY of this week]

## My Info using node module – msnninfo.js

1. **Make local module – msnninfo.js**
2. **Call msnninfo.js from index\_msnn.js.**
3. **Capture your code: msnninfo.js**

```
My Info
ID : ms77
Name : 캠시
Phone : 010-5678-1234
```

# Node.js : node apps – msnninfo.js



43

**index\_msnn.js uses the user module msnninfo.js.**

```
1 // index_aann.js
2
3 var myinfo = require('./ms00info');
4
5 myinfo("ms00", "Redwoods", '010-1234-5678');
6
7 myinfo("ms77", "컴사", '010-5678-1234');
```

My Info  
ID : ms00  
Name : Redwoods  
Phone : 010-1234-5678

**Capture msnninfo.js code  
as MSnn\_info.png**

# 과제11. msnn\_rpt11.zip

44

## [실습과제11] Node.js Basic

[1] 실습 결과 그림 3장 저장.

[2] [MSnn\\_package.png](#)

[MSnn\\_folder.png](#)

[MSnn\\_server.png](#)

[MSnn\\_myinfo.png](#)

\*\*\*\* [MSnn\\_Rpt11.zip](#) 으로 압축해서 제출하시오.

[제출파일] [msnn\\_rpt11.zip](#) (11월14일 오후 6시 마감)

결과를 압축하여 이메일로 “[msnn\\_rpt11](#)” 제목으로 제출

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)



# Final exam.

[1] 12월 5일(화) 오후 2시 (장소 : E323)

[2] 필기 시험 (30점)

- 교재 13장 연습 문제 (선다형)
- **Javascript coding & general questions**  
(jQuery, SVG, D&D, Google Map, Node.js: 4지 선다형/단답형)

\*\*\*\* 잘 준비해서 웃기를 ... ^\_^

# 교재 WEB 강의 소개



◀ ▶ G ⓘ webprogramming.co.kr ☆

**명품 웹 프로그래밍**

**HTML5 + CSS3 + Javascript**

**웹 프로그래밍**

Home Introduction Notice Board Support Code

HTML5로 여러분의 무한한 상상력을 표현해 보세요!

Sir Tim Berners-Lee (1955.6.8 ~ )

명품 웹 프로그래밍 소개

“웹 프로그래밍을 가장 쉽게 익힐 수 있는 책”

처음 웹 프로그래밍을 공부하는 입문자들도 모든 주제를 직관적으로 이해하고 빠르게 파악할 수 있습니다.

자세히보기 →



강력한 Q&A 피드백 제공

“빠르고, 간결하고, 정확한 저자의 직접적인 답변”

‘이거 이해가 잘 안되는데.. 물어볼 사람도 없고..’  
더이상 고민하지 마세요.  
명품 웹 프로그래밍 홈페이지에서는  
누구나 저자가 직접 답변해주는  
Q&A 게시판을 이용할 수 있습니다.

자세히보기 →



즉석 실행 가능한 예제 프로그램

“백문이 불여일견, 백견이 불여일타(打)!”

코드로만 설명되어 있는 예제들,  
결과 화면이 있어도 이해가 잘 안되시죠?  
예제 소스를 바탕으로, 내맘대로 수정한  
코드를 즉석으로 웹 페이지로  
변환해주는 예제 프로그램을 통해  
모든 코드를 빠르고 쉽게  
이해할 수 있습니다.



자세히보기 →

Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

<http://webprogramming.co.kr>

# 관련 WEB 강의 소개 – w3schools.com

The screenshot shows the homepage of w3schools.com. The top navigation bar includes links for TUTORIALS, REFERENCES, and EXAMPLES. The main content area is divided into three main sections: HTML, CSS, and JavaScript.

**HTML Section:**

- ## HTML

The language for building web pages

[LEARN HTML](#) [HTML REFERENCE](#)

**HTML Example:**

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

[Try it Yourself »](#)

**CSS Section:**

- ## CSS

The language for styling web pages

[LEARN CSS](#) [CSS REFERENCE](#)

**CSS Example:**

```
body {
    background-color: lightblue;
}
h1 {
    color: white;
    text-align: center;
}
p {
    font-family: verdana;
    font-size: 20px;
}
```

[Try it Yourself »](#)

**JavaScript Section:**

- ## JavaScript

The language for programming web pages

**JavaScript Example:**

```
<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.innerHTML = "Hello";
}
```

<http://www.w3schools.com>