

```
1 1. node.js 설치
2 1)Visit nodejs 홈페이지(https://nodejs.org/en/)
3 2)각 OS에 맞는 설치파일 다운로드
4 3)각 OS에 맞게 설치
5 4)Windows 환경은 자동으로 실행 파일 경로 등록한다.
6 Path = %Path%;C:\Program Files\nodejs
7
8 5)설치가 성공적으로 끝나면 콘솔에서 버전 정보를 출력한다.
9 $ node --version or $ node -v
10 v6.10.2
11
12 6)Debian and Ubuntu based Linux distributions
13 refer to : https://nodejs.org/ko/download/package-manager/
14 - $ curl
15 The program 'curl' is currently not installed. You can install it by typing:
16 sudo apt install curl
17 - $ sudo apt-get install curl
18 - $curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -
19 - $ sudo apt-get install -y nodejs
20 or
21 - $ sudo apt-get update
22 - $ sudo apt-get install build-essential openssl libssl-dev pkg-config
23 - $ sudo apt-get install make gcc g++ python
24 - $ wget https://nodejs.org/dist/v6.10.2/node-v6.10.2.tar.gz
25 - $ tar xvfz node-v.x...tar.gz
26 - $ cd node...
27 - $ ./configure
28 - $ make
29 - $ sudo make install
30
31 7)Enterprise Linux와 Fedora
32 refer to : https://nodejs.org/ko/download/package-manager/
33 - # curl --silent --location https://rpm.nodesource.com/setup\_6.x | bash -
34 - # yum -y install nodejs
35
36 8)설치 디렉토리에서
37 -node.exe : node.js의 주 실행파일
38 -npm.cmd : node.js의 모듈을 설치하기 위해서 사용하는 npm(Node Package Manager)을 실행시키기 위한 파
39 일
40 -node_modules : node.js의 모듈이 설치되는 디렉토리. 초기 설정에는 npm 모듈만 설치되어 있다.
41
42 2. Console에서
43 1)$ node -h or $ node --help
44 $ node console command
45 $ node source.js [args]
46 $ node -v, --version
47 $ node -e, --eval script : script 평가
48 $ node -c, --check : 실행하지 않고 문법 체크
49 $ node -r, --require : 모듈을 미리 로딩
50
51
52 3. REPL
53 Refer to https://nodejs.org/dist/latest-v6.x/docs/api/repl.html
```

```
54 1)node.js가 설치되면 command prompt 에서도 node.js를 사용할 수 있다.
55 2)이것을 REPL(Read Eval Print Loop)라고 한다.
56 3)REPL은 웹 브라우저의 자바스크립트 console과 유사하게 명령어를 입력하여 바로 실행해 결과를 확인할 수 있는 도구이
   다.
57 4)간단한 코드 테스트나 문법 확인용으로 사용하면 좋다.
58 5)node REPL의 모든 명령어는 '.'으로 시작하는 특징이 있다.
59
60 6)$ node
61   > var str = 'Hello world';
62   undefined
63   > str
64   'Hello World'
65   > str.length
66   11
67   > str + '!'
68   'Hello World!'
69   >
70   >function sum(a, b) { return a + b; }
71   undefined
72   >sum(3,4);
73   7
74
75   >JSON
76   {}
77   >JSON.stringify({name:'Toby'});
78   '{"name":"Toby"}'
79
80   >var test = function(x, y){
81   ... var val = x * y;    <----아직 명령어가 완료되지 않았다는 것을 표시하기 위해 연속된 점(...)을 제공한다.
82   ... return val;
83   ... };
84   undefined
85   > test(3,4);
86   12
87
88   >test(4,
89   ... 5);
90   20
91
92   console.js파일의 내용을 복사해서 REPL 창에서 right-click 하면 붙여넣기 및 실행된다.
93   마지막 라인에서 커서가 멈춘다.
94   물론 화살표키를 이용해서 제한적이거나 REPL에서 입력했던 명령어들을 볼 수 있고, 수정할 수 있다.
95   >.help
96   break  sometimes you get stuck, this gets you out
97   clear  Alias for .break
98   exit   Exit the repl
99   help   Show repl options
100  load   Load JS from a file into the REPL session
101  save   Save all evaluated commands in this REPL session to a file
102  >
103
104  .break : 현재 구동중인 작업을 종료하고 대기 상태로 변경한다.
105  .clear : 현재 context를 초기화한다. 이는 node.js를 재실행하지 않더라도 사용중이었던 모든 변수 및 클로저 등을 깨끗
   이 정리해준다.
```

106 .exit : Node REPL을 종료한다. <Ctrl> + <D>키와 같다.
107 .help : 도움말 메뉴를 출력한다.
108 .load : Js 파일을 REPL 세션으로 읽어들인다.
109 .save : 현재 수행 중인 REPL 세션에서 실행했던 모든 명령어들을 파일로 저장한다.

110
111

112 4. 개발툴 설치

113 1)Eclipse, Visual Studio Code, Sublime Text 3 같은 개발툴

114 2)Visual Studio Code(<https://code.visualstudio.com/>)

115 -File > Preferences > Settings

116 -{

117 "editor.fontFamily": "Verdana",

118 "editor.fontSize": 16,

119 "editor.wrappingColumn": 0,

120 "editor.wordWrap": true,

121 "editor.rulers": [

122 100, 120

123],

124 "editor.cursorStyle": "block",

125 "editor.cursorBlinking": "phase",

126 "files.autoSave": "afterDelay"

127 }

128

129 3)Eclipse + ENIDE(Eclipse Node.js IDE) plug-in

130 -Help > Eclipse Marketplace... > In Search tab > Find : 'enide' > Enide(Studio) 2015

131 -Click 'Install'

132 -Check All > Confirm > I agree > Installation > 중간에 Warning

133 -Restart Eclipse

134 -Window > Perspective > Open Perspective > Other

135 -Select [Node] > OK

136 -New > Node.js Project

137 -Run As > Node Applicaton

138

139

140 5. Lab1

141 1)In the Explorer, go to D:\NodeJSHome

142 mkdir hello

143 cd hello

144 code . --> Visual Studio Code 창이 열림.

145 You can open files or folders directly from the command line. The period '.' refers to the current folder, therefore VS Code will start and open the Hello folder.

146

147 2)From the File Explorer tool bar, press the New File button:

148

149 3)And name the file hello.js:

150 By using the .js file extension, VS Code interprets this file as JavaScript and will evaluate the contents with the JavaScript language service.

151 Create a simple string variable in app.js and send the contents of the string to the console:

152 Note that when you typed console. IntelliSense on the console object was automatically presented to you. When editing JavaScript files, VS Code will automatically provide you with IntelliSense for the DOM.

153

154 4)hello.js

155 var msg = 'Hello World';

```
156 console.log(msg);
157
158 5)It's simple to run app.js with Node.js. From a terminal, just type:
159 $ node hello.js
160
161 6)You should see "Hello World" output to the terminal and then Node.js returns.
162 Hello World
163
164
165 6. Integrated Terminal
166 1)VS Code has an integrated terminal which you can use to run shell commands. You can run
Node.js directly from there and avoid switching out of VS Code while running command line
tools.
167 2)View > Integrated Terminal (Ctrl+` with the backtick character) will open the integrated
terminal and you can run node app.js there:
168
169
170 7. Lab2
171 1)helloworld.js
172 var http = require('http');
173
174 http.createServer(function (request, response) {
175     response.writeHead(200, {'Content-Type' : 'text/html'});
176     response.write('<h1>Hello, World</h1>');
177     response.end();
178 }).listen(80, function(){
179     console.log('Server running at http://localhost/');
180 });
181
182 2)$ node helloworld.js
183 Server running at http://localhost/
184
185 3)In Browser, http://localhost/로 Hello World가 나오는 페이지 확인
186
187
188 8. Linux에서 80 port 사용시 아래와 같은 오류발생
189 $ node helloworld.js
190 events.js:160
191     throw er; // Unhandled 'error' event
192     ^
193
194 Error : listen EACCES 0.0.0.0:80
195 ...
196
197 Refer to https://geunhokhim.wordpress.com/2016/03/29/nodejs-error-listen-eaccess-0-0-0-0-80/
198 root 권한이 없는 계정으로 Linux 내에서 Application을 수행할 때, 1024 보다 작은 Port에는 바인딩 할 수 없다.
199 Node.js에서 Web Server를 띄울 때에도 마찬가지로 인데, 아래와 같은 방법으로 처리 할 수 있다.
200
201 -Application을 root 권한으로 수행
202 -혹은, 1024 이상의 Port 번호를 바인딩 하도록 수정
203 -Application이 1024 보다 작은 Port에 바인딩 할 수 있도록 설정
```