

외부 모듈

# 외부모듈

일반 개발자가 만들어 배포한 모듈  
npm을 기반으로 모듈을 공유한다.

# NPM: Node Package Manager

Node.js의 패키지 / 모듈 저장소

Node.js 패키지 설치, 버전 및 호환성 관리할 수 있는 유틸리티

# NPM: Node Package Manager

## NPM 설치 확인방법

```
$ npm --version
```

```
4.5.0
```

```
Update available 4.5.0 → 5.3.0  
Run npm i -g npm to update
```

## 구버전의 NPM 새로운 버전으로 업데이트 방법

```
$ npm install npm -g # -g 옵션을 사용하면 컴퓨터에 저장이 됨
```

```
$ npm --version
```

```
5.3.0
```

# NPM: Node Package Manager

## package.json

npm은 프로젝트에 대한 설정을 package.json 이라는 파일에 의존하고 있고 이 파일에 JSON 형식으로 작성하여 프로젝트를 관리할 수 있습니다.

```
{
  "name": "node.js",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.17.2",
    "ejs": "^2.5.6",
    "express": "^4.15.3",
    "mysql": "^2.13.0",
    "nodemon": "^1.11.0",
    "path": "^0.12.7"
  },
}
```

# NPM: Node Package Manager

## 모듈 업데이트 방법

```
$ npm update express
```

## 모듈 검색 방법

```
$ npm search express
```

## 모듈 제거 방법

```
$ npm delete express
```

## 중복 모듈 정리

```
$ npm dedupe
```

<https://jungwoon.github.io/node.js/2017/08/01/Node-NPM/>

# NPM: Node Package Manager

## 기본 사용법

```
var express = require('express');
```

## Example

```
var express = require('express'); // express 모듈 사용하기 위함
var app = express(); // express 모듈을 app이라는 변수명으로 사용
var bodyParser = require('body-parser'); // body-parser를 bodyParser란 변수로 사용
var router = require('./router/index');
var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;
var session = require('express-session');
var flash = require('connect-flash');
```

ejs

## 템플릿 엔진 모듈

특정 형식의 문자열을 HTML 형식의 문자열로 변환

클라이언트 요청에 따라 웹문서 결과가 달라짐

정적인 부분과 동적인 부분을 따로 하기위해 사용





# ejs

## 1. 설치

```
npm install ejs
```

## 2. 추출

```
// ejs 모듈을 추출한다.  
var ejs = require('ejs')
```

## 3. 변환

```
// ejs 문자열을 html 문자열로 변경해준다.  
ejs.render(data);
```

Pass EJS a template string and some data. BOOM, you've got some HTML.

```
var ejs = require('ejs'),  
    people = ['geddy', 'neil', 'alex'],  
    html = ejs.render('<%= people.join(", "); %>', {people: people});
```

```
var template = ejs.compile(str, options);  
template(data);  
// => Rendered HTML string  
  
ejs.render(str, data, options);  
// => Rendered HTML string
```

```
// 파일 시스템 모듈 사용
var fs = require('fs');
// 서버 생성을 위한 http모듈 사용
var http = require('http');
// ejs 모듈 추출
var ejs = require('ejs');

// 서버 생성 및 실행
http.createServer(function(req, res){
    fs.readFile('ejsPage.ejs', 'utf-8', function(err, data){

        // 응답 스트림에 헤더를 작성
        //.writeHead 메소드는 헤더를 작성한다.
        // html 파일 형식으로 응답을 보낸다
        res.writeHead(200, {'Content-Type': 'text/html'});

        // ejs 문자열을 html 문자열로 변경해준다.
        res.end(ejs.render(data));

    });
}).listen(52273, function(){
    console.log('server run');
})
```

```
<% var name = "soomti" %>

<h1><%=name %></h1>
<p><%= 52 * 273 %></p>

<hr />

<% for(var i = 0 ; i < 10 ; i ++){ %>
    <h2>the square of <%= i %> is <%= i * i %></h2>
<% } %>
```

# soomti

14196

---

the square of 0 is 0

the square of 1 is 1

the square of 2 is 4

## ejs 특수 태그

- `<%` 'Scriptlet' tag, for control-flow, no output
- `<%=` Outputs the value into the template (HTML escaped)
- `<%-` Outputs the unescaped value into the template
- `<%#` Comment tag, no execution, no output
- `<%%` Outputs a literal '<%'
- `%>` Plain ending tag
- `-%>` Trim-mode ('newline slurp') tag, trims following newline

<https://opentutorials.org/course/50/191>

## 서버 실행 모듈

지역 모듈 : 자바스크립트 파일 내부에서 사용하는 모듈

전역 모듈 : 터미널에서 사용할 수 있는 모듈

전역 모듈에서는 -g 옵션을 사용해서 받는다

# supervisor 모듈

## supervisor 모듈

`$ node app.js` 의 경우, 서버를 키고 파일 내용을 변경 시에 인식을 못한다.

`supervisor` 모듈은 파일 내용 변경을 인식하는 모듈로, 자동으로 인식하여 실행을 종료시킨후에 다시 실행시킨다.

### 1.설치

```
$ sudo npm install -g supervisor  
[sudo] password for USER : 비밀번호 입력
```

### 2. 설치 확인

```
$ supervisor
```

```
----- 실행 결과 -----
```

```
Node Supervisor is used to restart programs when they crash.  
It can also be used to restart programs when a *.js file changes.
```

```
Usage:
```

```
  supervisor [options] <program>  
  supervisor [options] -- <program> [args ...]
```



# supervisor 모듈

## example

```
// 터미널 실행
$ supervisor test.server.js

Running node-supervisor with
  program 'test.server.js'
  --watch '.'
  --extensions 'node,js'
  --exec 'node'

Starting child process with 'node test.server.js'
Watching directory '/Users/Soomti/STUDY/Javascript/NODEJS/modernJS/chap07' for changes.
Press rs for restarting the process.
test!
// 파일 변경후 저장 버튼 누른 후
crashing child
Starting child process with 'node test.server.js'
test changed!
```

# forever 모듈

## forever 모듈

단일 스레드 기반인 node.js 웹 서비스는 예외 하나로 웹 서비스가 크게 죽어버리는데, 이런 상황을 대비하고자 만든 모듈

`$ node app.js` 의 경우, 서버를 키고 파일 내용을 변경 시에 인식을 못한다.

`supervisor` 모듈은 파일 내용 변경을 인식하는 모듈로, 자동으로 인식하여 실행을 종료시킨후에 다시 실행시킨다.

### 1.설치

```
$ sudo npm install -g supervisor  
[sudo] password for USER : 비밀번호 입력
```

### 2. 설치 확인

```
$ supervisor
```

```
----- 실행 결과 -----
```

근데오류남 ㅠㅠㅠㅠ

```
i@hansumin-ui-MacBook-Pro: ~/STUDY/Javascript/NODEJS/modernJS/chap07 $ forever start app.js
--minUptime not set. Defaulting to: 1000ms
--spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
Forever processing file: app.js
```



2016인기모듈 30가지 ㅎㅎ

<https://colorlib.com/wp/npm-packages-node-js/>