

# Nodejs

2016/12/04

# 진행

- ▶ module 문법 (require/exports)
- ▶ 내장 객체
- ▶ API Documents
- ▶ NPM
  - ▶ express

# 모듈 (exports/require)

- ▶ mod-math.js / module.js로 실습
  - ▶ <https://github.com/owo-study/nodejs2016/blob/master/2weeks/mod-math.js>
  - ▶ <https://github.com/owo-study/nodejs2016/blob/master/2weeks/module.js>
- ▶ 접근 제어(Access control)에 대해서
  - ▶ exports에 넣지 않는 메소드나 변수는 private하게 된다.
  - ▶ exports에 넣은 멤버(변수/메소드)들은 public하게 된다.

mod-math.js 파일을 아래와 같은 내용으로 작성합니다.

```
function private_func(){
    console.log("[mod-math.js] is Loaded");
}
private_func();

exports.multiply = function(aa, bb){
    return aa*bb;
};

function factorial( n ){
    return (n > 1) ? n * factorial(n-1) : 1 ;
}
exports.factorial = factorial;
```

app.js 파일을 아래와 같은 내용으로 작성 후

```
var myMath = require("./mod-math.js");

var result = myMath.multiply(10, 10)
console.log(result);

result = myMath.factorial(4);
console.log(result);

// myMath.private_call();
```

콘솔에서 node app으로 실행해 주세요

# 모듈의 종류

- ▶ 사용자 모듈
- ▶ 내장 모듈
  - ▶ nodejs에 설치하면서 기본으로 포함된 모듈
  - ▶ API Documents
    - ▶ Ver 6.x => <https://nodejs.org/dist/latest-v6.x/docs/api/>
    - ▶ Ver 7.x => <https://nodejs.org/dist/latest-v7.x/docs/api/>
- ▶ 전역(global) 모듈
  - ▶ <https://github.com/owo-study/nodejs2016/blob/master/2weeks/globals.js>

# 전역 모듈 globals

<https://nodejs.org/dist/latest-v6.x/docs/api/globals.html>

- ▶ 전역 변수
  - ▶ `__filename` / `__dirname`
- ▶ 전역 메소드
  - ▶ `setTimeout()` / `setInterval()`
- ▶ Console 객체
  - ▶ 로그용 메소드
    - ▶ `info()`, `log()`, `warn()`, `error()`
  - ▶ 성능(시간) 측정
    - ▶ `Time()~timeEnd()`
- ▶ Process 객체
  - ▶ 각종 정보들

# 내장 모듈

<https://github.com/owo-study/nodejs2016/blob/master/2weeks/inner-mods.js>

- ▶ OS 모듈
- ▶ URL 모듈
- ▶ HTTP 모듈
- ▶ Filesystem 모듈



# OS 모듈

<https://nodejs.org/dist/latest-v6.x/docs/api/os.html>

▶ 운영체제(서버측 시스템)에 대한 정보에 관한 객체

메소드	설명
os.arch()	운영체제 아키텍처를 가져옵니다
os.cpus()	<b>CPU</b> 정보를 가져옵니다
os.freemem()	사용가능한 메모리 정보를 가져옵니다
os.homedir()	현재 사용자의 <b>home</b> 디렉토리 정보를 가져옵니다
os.hostname()	서버의 <b>hostname</b> 을 가져옵니다
os.networkinterface()	네트워킹 인터페이스 정보를 가져옵니다
os.platform()	운영체제 플랫폼 정보를 가져옵니다
os.release()	운영체제의 버전을 가져옵니다
os.totalmem()	총 메모리 정보를 가져옵니다
os.type()	운영체제 이름을 가져옵니다
os.uptime()	운영체제가 실행된 시간을 가져옵니다.

# URL 모듈

<https://nodejs.org/dist/latest-v6.x/docs/api/url.html>

- ▶ 이름에서 알 수 있듯이 url과 관련된 처리를 위한 유틸리티 모듈

<b>url.format( urlObj )</b>
urlObject를 url문자열로 바꿔 줍니다
<b>url.parse(urlStr [, parseQueryString[, slashedDenoteHost] )</b>
url문자열을 urlObject로 바꿔 줍니다.
<b>url.resolve(from, to)</b>
from의 주소를 베이스로 to를 연결하여 url을 완성하여 줍니다.

# 파일 시스템 모듈

<https://nodejs.org/dist/latest-v6.x/docs/api/fs.html>

▶ 각자 과제로 해보기!

# HTTP 모듈

<https://nodejs.org/dist/latest-v6.x/docs/api/http.html>

- ▶ 웹(http)서버를 구축하기 위한 모듈

```
var http = require("http");

http.createServer(function(request, response){
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end("Hello nodejs Webserver");
}).listen(8000);

console.log("Server running...");
```

# Node Package Manager

<https://www.npmjs.com/>

- ▶ 다른 사람들이 만든 사용자 모듈을 묶어둔(Package)를 관리해주는 프로그램
- ▶ 다른 사용자가 만든 모듈이 여러 모듈들을 포함하여 만들었을 때 생기는 문제들 즉, 의존성 패키지 대한 관리를 알아서 해준다!
- ▶ 설치 : `npm install [패키지명]`
- ▶ 삭제 : `npm remove [패키지명]`
- ▶ 설치된 목록 : `npm list`
- ▶ 매뉴얼 : `npm help`
- ▶ 명령어별 매뉴얼 : `npm [명령어] -h`

# express 모듈 설치하기

<http://expressjs.com/ko/>

- ▶ `npm install express`

# express-generator 모듈 설치하기

<http://expressjs.com/ko/starter/generator.html>

- ▶ `npm install express-generator -g`

# 참고 자료

## ▶ express까지 빠르게

- ▶ 삼투월드님 동영상 강의 => <https://www.infllearn.com/course/node-js-%EB%85%B8%EB%93%9C%EC%A0%9C%EC%9D%B4%EC%97%90%EC%8A%A4-%EA%B0%95%EC%A2%8C/>

## ▶ 기초부터 차근히

- ▶ 생활코딩 => <https://opentutorials.org/module/938>
- ▶ Tacademy 온라인 강의 => <https://tacademy.sktechx.com/live/player/onlineLectureDetail.action>

## ▶ 참고 문서

- ▶ nodejs 입문 문서(번역) => <http://nodebeginner.org/index-kr.html>
- ▶ <https://www.zerocho.com/category/NodeJS/post/57387cb8715202c8679b3af1>