



Embedded Systems

who's next@gachon.ac.kr

Password

Log In

2018.11.1

임베디드 시스템

컴퓨터공학과 이병문

09 **Node**기반 웹프레임워크 기술

10 웹기반 임베디드시스템 제어**1**

11 웹기반 임베디드시스템 제어**2**

12 웹기반 임베디드시스템 제어**3**

13 **REST API**기반 임베디드 연동 **1**

14 **REST API**기반 임베디드 연동 **2**

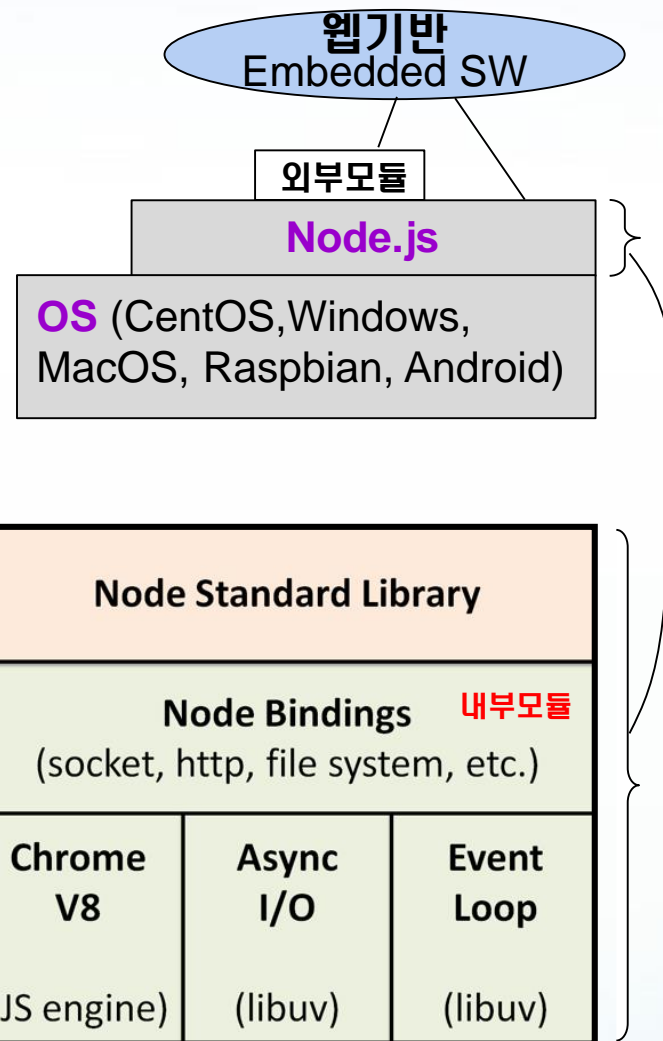
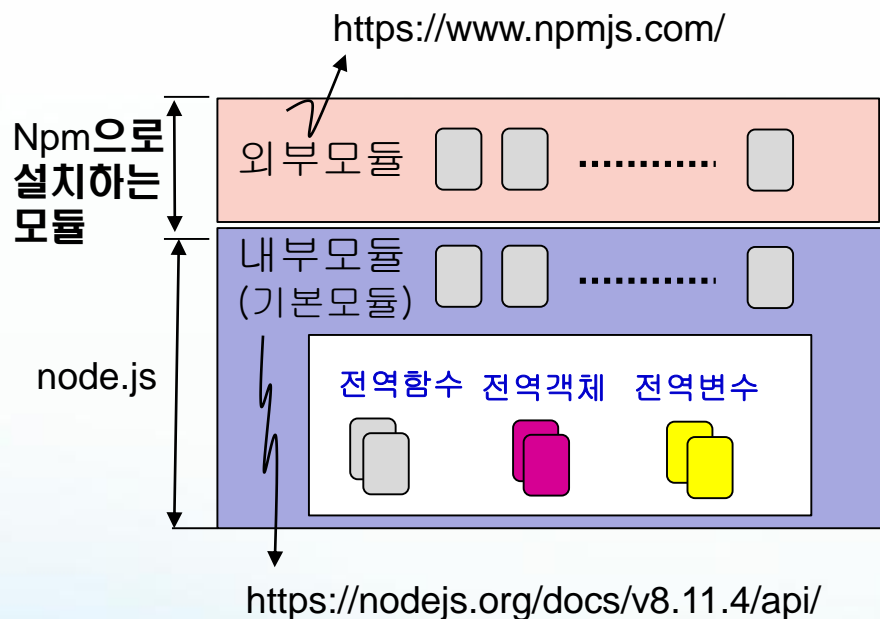
15 임베디드 무선연동기술

16 기말고사

■ Node.js 기본구조

☑ Node.js 의 전역함수

- Node.js 의 기본구조
- Node.js 에서 제공하는 모듈/객체/함수



■ 전역함수란 ?

☑ 선언하지 않고 사용할 수 있는 함수 ... Node.js 가 제공

예) `setTimeout()`, `clearTimeout()`, `setInterval()`, `clearInterval()`
`setImmediate()`, `clearImmediate()`

예제1) time.js

```
setTimeout (function ( ) { console.log("1초경과"); }, 1000);
setTimeout (function ( ) { console.log("2초경과"); }, 2000);
setTimeout ( ( ) => { console.log("3초경과"); }, 3000);
setTimeout (function ( ) { console.log("5초경과"); }, 5000);
setTimeout (function ( ) { console.log("7초경과"); }, 7000);
const id = setTimeout ( ( ) =>
    { console.log("취소될 함수 π π"); }, 10000);
clearTimeout (id); // 타이머취소 ^^;
setInterval ( ( ) =>
    { console.log("4초마다 호출"); }, 4000);
```

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[20160000@localhost svr]$ vi time.js
[20160000@localhost svr]$ node time
1초 경 과
2초 경 과
3초 경 과
4초 마 다   호 출
5초 경 과
7초 경 과
4초 마 다   호 출
4초 마 다   호 출
4초 마 다   호 출
^C
[20160000@localhost svr]$
```

■ 전역객체란 ?

☑ 프로그램 전체에서 사용할 수 있는 객체 (예, console, export, process, ...) ... Node.js 가 제공

예) **console**.log ("Hello world ! "); <- 어떤 선언이나 준비없이도 프로그램에서 바로 사용

예) **process** 객체

→ require() 함수로 객체를 추출할 필요가 없음

☑ Console 객체

```
var    sum = 0;
const  code = 5;

console.time('실행시간');
for (var cnt = 0; cnt < 100; cnt++) {
    sum = sum + cnt;
}
console.timeEnd('실행시간');
console.log("반복횟수: %d", cnt);
console.log("반복횟수:", cnt);
console.log(cnt);
console.log(`반복횟수와 총합계: ${ cnt + sum }`);
console.log("%d, %s, %j", sum, "그리고", { age: 24 });
console.error('error # %d', code);
```

라벨이름이 같아야 함!!

Grave이용하면
템플릿문자열사용가능
(표현식을 넣을 수 있음)

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

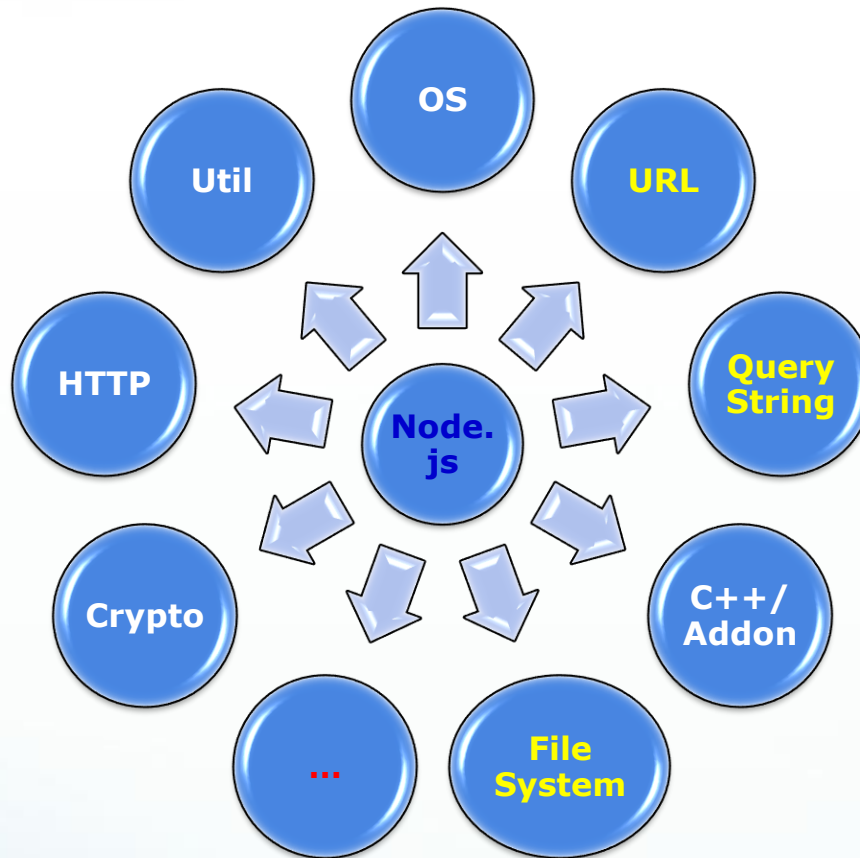
```
[leebyungmun@localhost 3week]$ node con.js
실행 시간: 1ms
반복 횟수: 100
반복 횟수: 100
100
반복 횟수와 총 합 계: 5050
4950, 그리고, , { "age": 24}
error #5
[leebyungmun@localhost 3week]$
```

%d 숫자
%s 문자열
%j JSON포맷

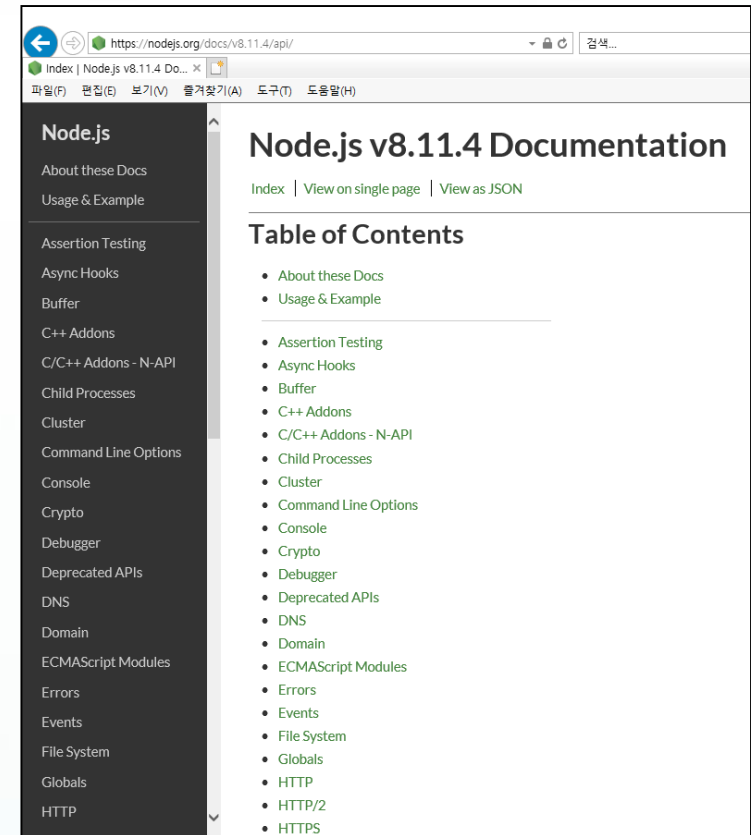
예제2 (con.js)

■ Node.js 의 내부모듈

☑ Node.js 프레임워크에서 제공하는 모듈들



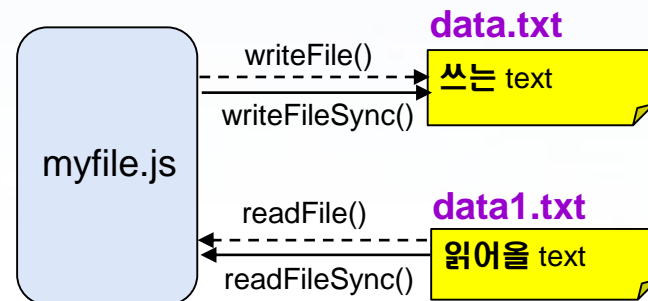
<https://nodejs.org/docs/v8.11.4/api/>



■ File System 모듈

☑ Directory/File I/O 기능을 제공하기 위한 모듈/객체

- 파일에서 읽기, 파일로 쓰기, 파일존재 유무, mode/소유자변경 등 동기/비동기메소드 제공
- 비동기는 파라미터로 **callback** 함수가 반드시 포함됨



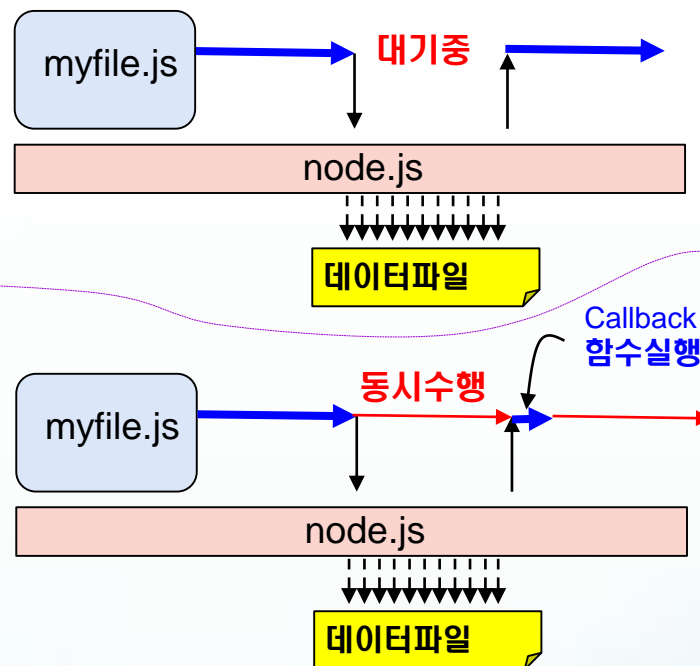
Synchronous I/O (동기식 I/O)

Thread waits during I/O operation



Asynchronous I/O (비동기식 I/O)

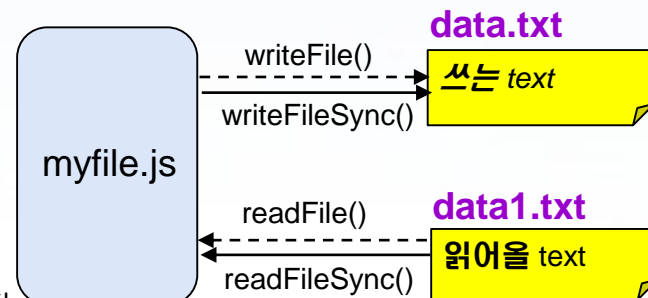
Thread DON'T wait during I/O operation



■ File System 모듈

☑ Directory/File I/O 기능을 제공하기 위한 모듈/객체

- 파일읽기, 파일쓰기, 파일존재 유무,
mode/소유자변경 등 동기식과 비동기식 메소드 제공
- 비동기식은 파라미터로 **callback** 함수가 반드시 포함됨



// 모듈추출

```
const fs = require('fs');
var text = '쓰는 text';
```

// 모듈사용

```
fs.writeFile('data.txt', text, 'utf8', function (error) { console.log('비동기식 쓰기'); } ); // 비동기식 I/O
fs.writeFileSync('data.txt', text, 'utf8'); // 동기식 I/O
console.log('동기식 쓰기');
```

```
var readtext;
```

```
fs.readFile('data.txt', 'utf8', function (error, readtext) { console.log('비동기식으로 읽음 :', readtext); } );
try {
  readtext = fs.readFileSync('data1.txt', 'utf8');
  console.log('동기식으로 읽음 : %s', readtext);
} catch (error) {
  console.log(error);
}
```

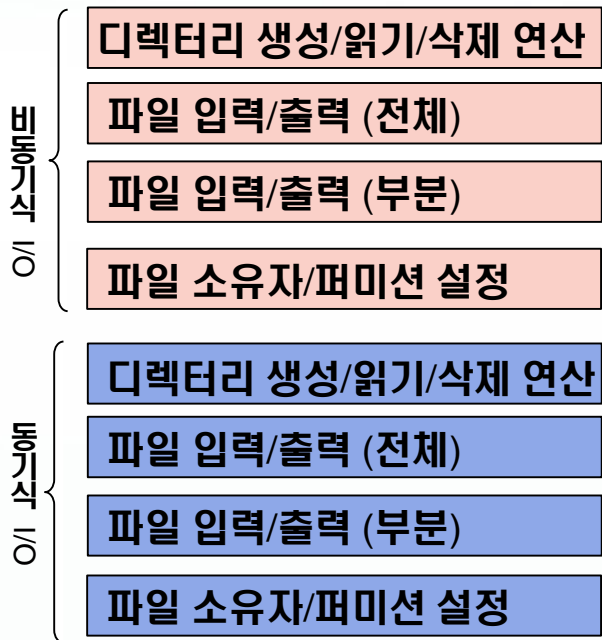
예제3 (myfile.js)



```
20160000@raspberrypi: ~/mypgm/filetest
myfile.js
20160000@raspberrypi:~/mypgm/filetest $ node myfile.js
동기식 쓰기
{ Error: ENOENT: no such file or directory, open 'data1.txt'
  at Object.fs.openSync (fs.js:646:18)
  at Object.fs.readFileSync (fs.js:551:33)
  at Object.<anonymous> (/home/20160000/mypgm/filetest/myfile.js:12:23)
  at Module._compile (module.js:652:30)
  at Object.Module._extensions..js (module.js:663:10)
  at Module.load (module.js:565:32)
  at tryModuleLoad (module.js:535:12)
  at Function.Module._load (module.js:497:3)
  at Function.Module.runMain (module.js:693:10)
  at startup (bootstrap_node.js:191:16) errno: -2, code: 'ENOENT', syscall: 'open', path: 'data1.txt'
}
비동기식 쓰기
비동기식으로 읽음 : 쓰는 text
20160000@raspberrypi:~/mypgm/filetest $ ls
data.txt myfile.js
```

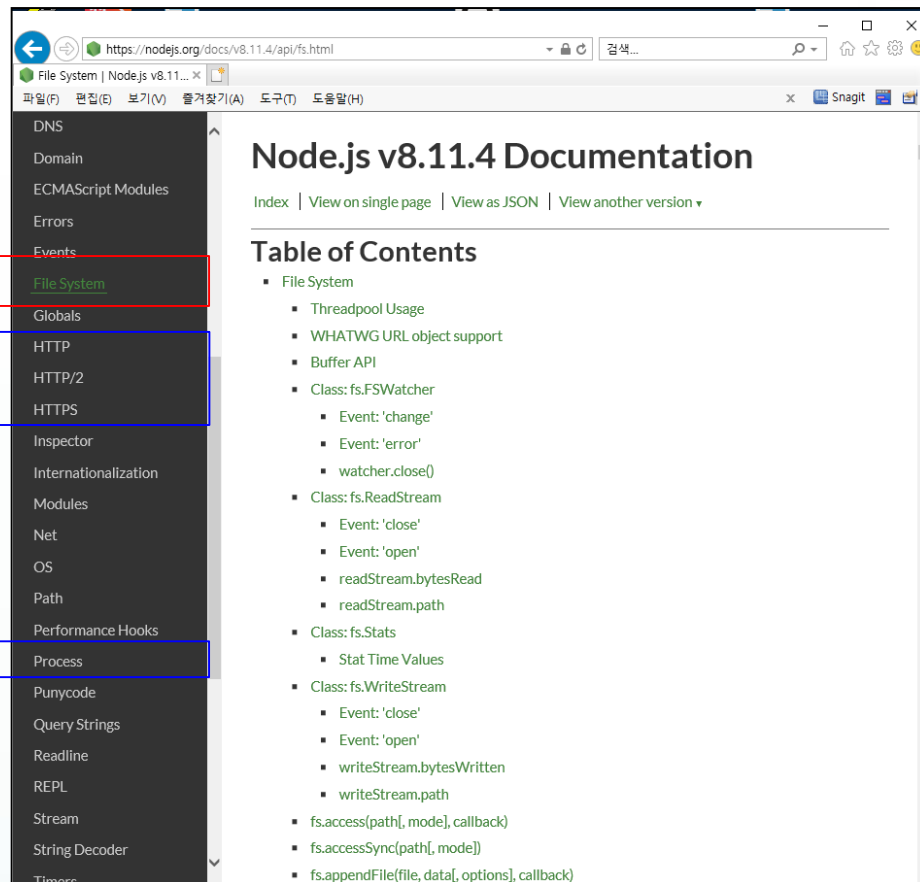

■ File System 모듈

- 기본적으로 비동기식 I/O이며, 동기식 I/O 기능도 모두 제공하고 있음



```
const exitHandler = function(code) {  
  console.log('프로세스가 종료됩니다.');
```

process.on('exit', exitHandler);



<https://nodejs.org/docs/v8.11.4/api/>

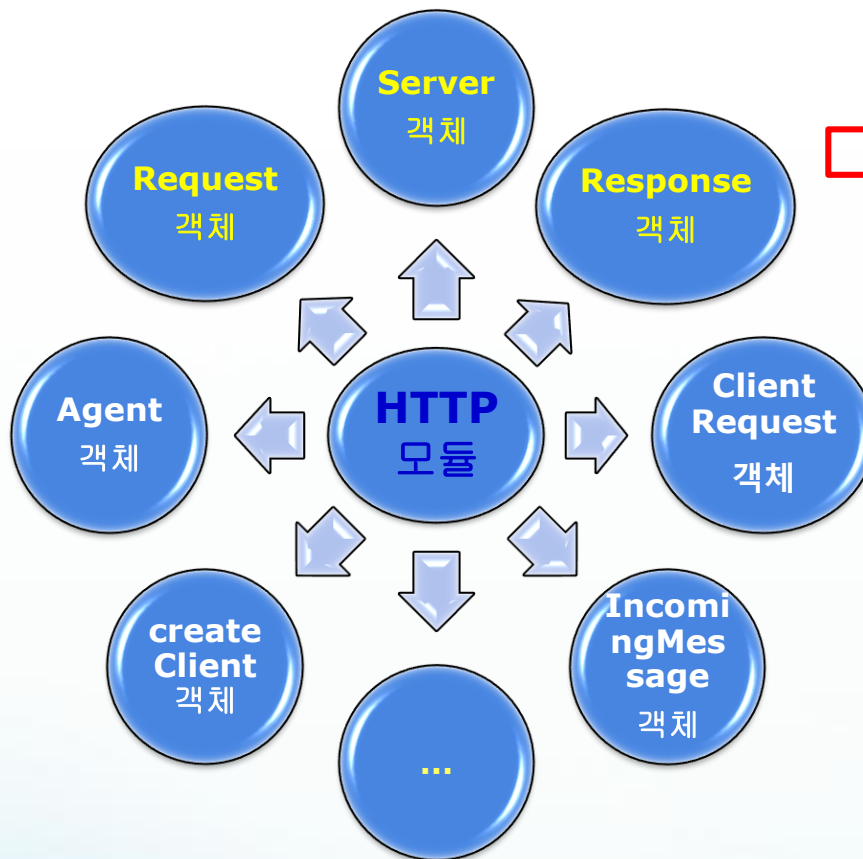
■ 실습일지

☑ 실습1



■ HTTP모듈의 객체들

☑ HTTP 기본모듈에서 제공하는 객체들

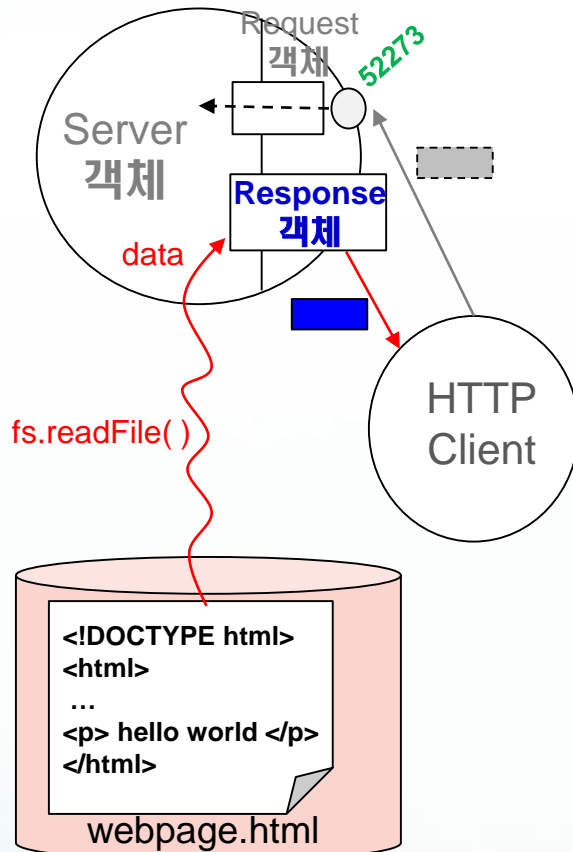


A screenshot of the Node.js documentation page for the `http` module. The left sidebar shows the "ECMAScript Modules" list with "HTTP" highlighted in a red box. The main content area displays the `http.createServer([requestListener])` and `http.get(options[, callback])` methods, including their signatures, return types, and usage examples.

<https://nodejs.org/docs/v8.11.4/api/>

■ Response 객체

- ☑ file system 모듈로 webpage.html 을 읽어서, 보내는 웹서버 예제



예제4 (readpage.js)

// 모듈을 추출

```
const fs = require('fs');
const http = require('http');
```

// 서버생성/서버실행

```
http.createServer( (request, response) => {
  // HTML 파일을 읽습니다.
  fs.readFile('webpage.html', function (error, data) {
    if (error) {
      response.writeHead(500, { 'Content-Type': 'text/plain' });
      response.end('Server: File error');
    }
    else {
      response.writeHead(200, { 'Content-Type': 'text/html' });
      response.end(data);
    }
  });
}).listen(52273, () => {
  console.log('Server Running at http://127.0.0.1:52273');
});
```

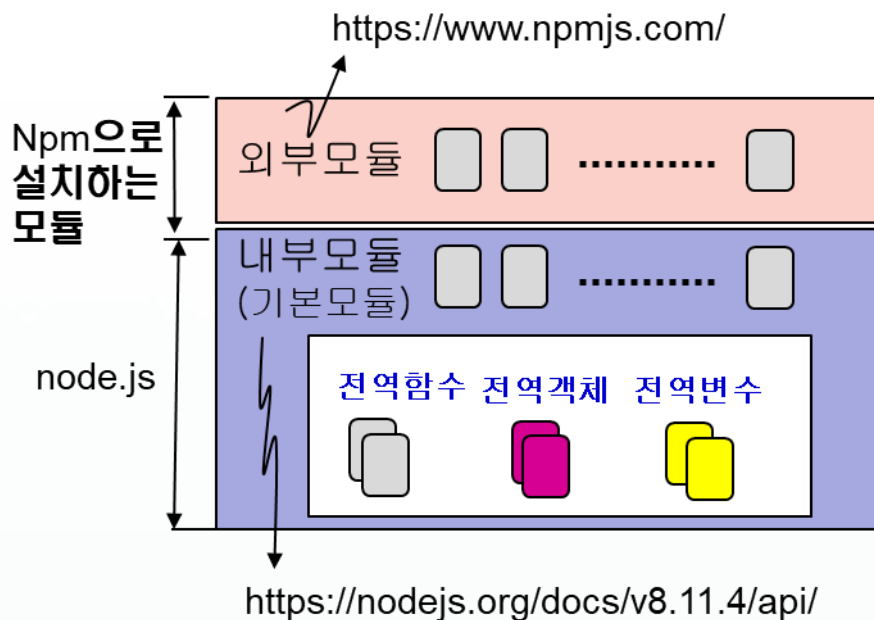
본인의
라즈베리파이 IP주소

```
20160000@raspberrypi: ~/mypgm
20160000@raspberrypi:~/mypgm $ node readpage.js
Server Running at http://102.9.80.241:52273
```

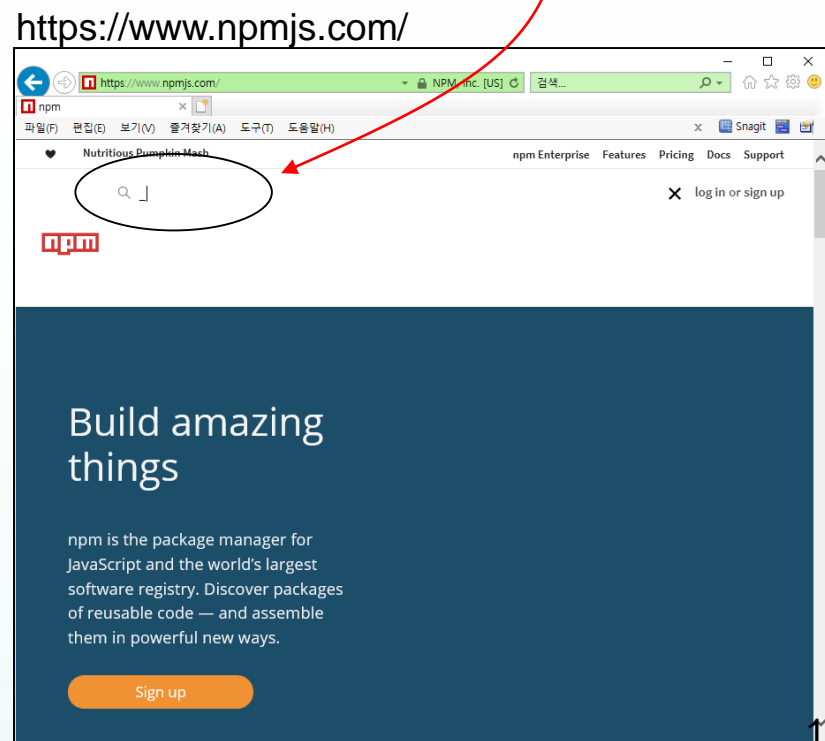
라즈베리파이 IP주소

■ 외부모듈

- ☑ 3rd party vendor, 개인이 개발하여 제공하는 모듈
- ☑ 다수의 프레임워크, 유틸리티, 함수기능을 제공



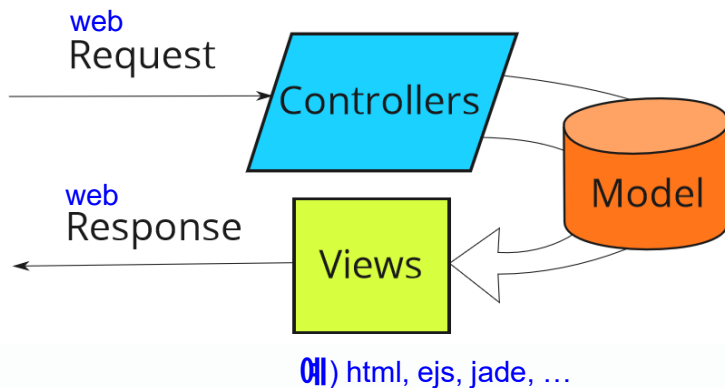
필요한 외부모듈을 검색한다.



■ 웹프레임워크

☑ 웹 디자인패턴(MVC = **M**odel, **V**iews, **C**ontrollers)

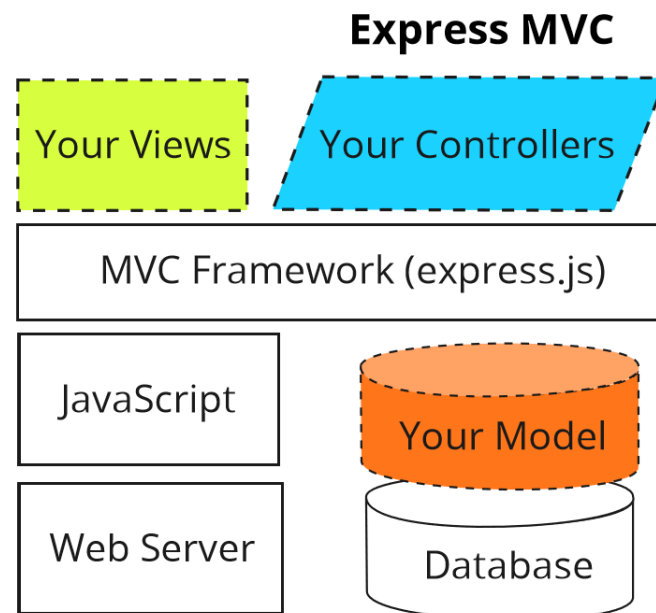
- 웹UI 와 비즈니스로직(처리코드)을 각각 분리시켜 설계하고 구현함으로써, 서로 영향없이 쉽게 수정/유지보수가 가능하게 하는 애플리케이션 개발방식



☑ 템플릿 모듈 (**ejs**, **jade** (= pug, pugJS))

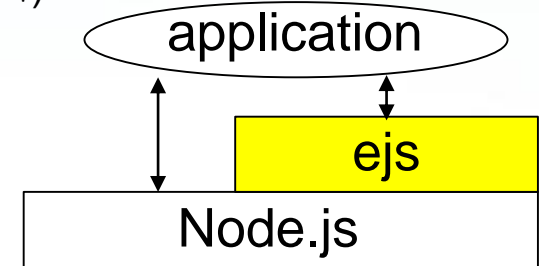
- HTML 문서에 Javascript 코드를 넣을 수 있도록 해서, 동적인 웹페이지를 생성하는 모듈

```
$ cd mypgm
$ npm install ejs
$ npm install jade
```



■ ejs 모듈 (= Embedded JavaScript Template 모듈)

- ☑ node.js 프레임워크에서는 *.html 파일만 읽어서 제공 (정적페이지)
- ☑ 확장모듈(예, ejs, jade)을 이용하면 동적페이지를 생성할 수 있음



```
const http = require('http');
const fs = require('fs');
const ejs = require('ejs'); ← 외부모듈(템플릿엔진)

// 서버생성, 서버실행
http.createServer(function (request, response) {
  // 7-8.ejs 파일을 읽습니다.
  fs.readFile('sample.ejs', 'utf8', function (error, data) {
    response.writeHead(200, { 'Content-Type': 'text/html' });
    response.write('<meta charset=utf8>');
    response.end(ejs.render(data, {
      name: 'ejs 실습문제',
      description: 'Hello ejs With Node.js .. !'
    }));
  });
}).listen(52273, function () {
  console.log('Server Running at http://127.0.0.1:52273');
});
```

예제5 (ejsmain.js)

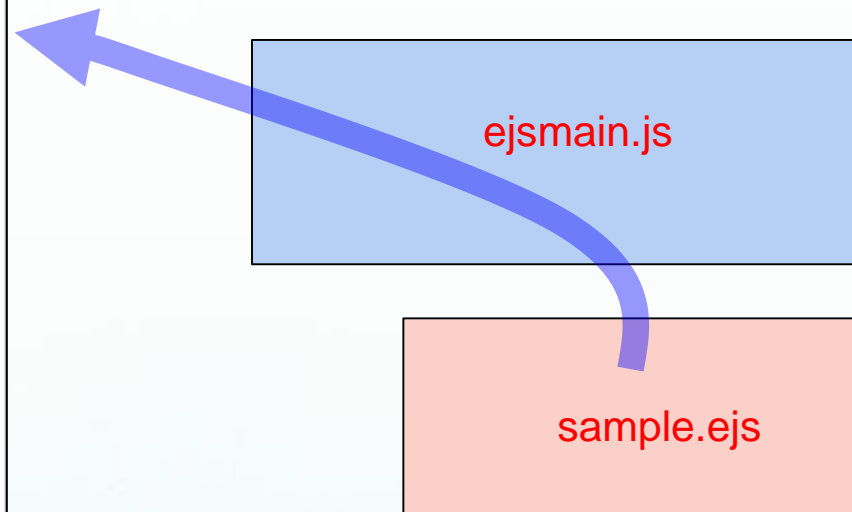
`<%= value %>`
`<% JS code %>`

sample.ejs

```
<h1><%= name %></h1>
<p><%= description %></p>
<hr />
<% for (var i = 0; i < 10; i++) { %>
  <h2>The Square of <%= i %>
    is <%= i * i %></h2>
  <% } %>
```

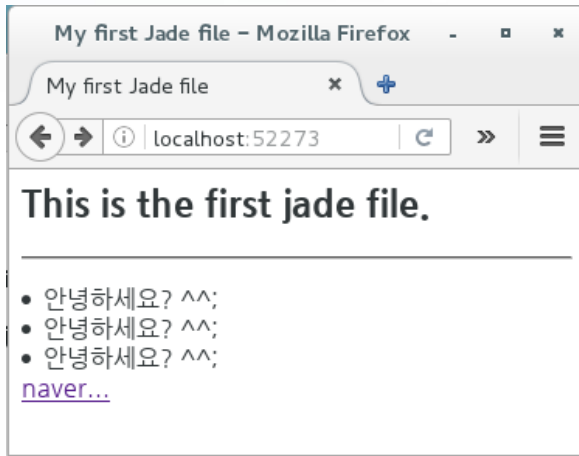
■ **ejs** 모듈 (= Embedded JavaScript Template 모듈)

☑ 실행결과



■ jade 템플릿 엔진

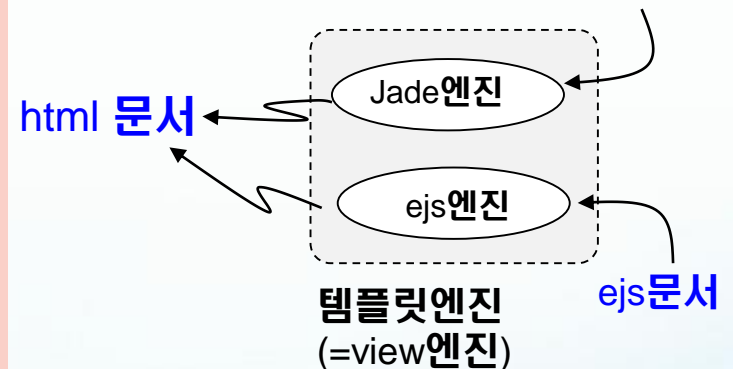
예제6) hello.jade



```
html
  head
    title My first Jade file
  body
    h2 This is the first jade file.
    hr
    ul
      - for (i=0; i < 3; i++)
        li 안녕하세요? ^^
    a(href="http://www.naver.com") naver...
```

```
<html>
<head>
  <title>My first Jade file</title>
</head>
<body>
  <h2>This is the first jade file.</h2>
  <hr/>
  <ul>
    <li>안녕하세요 ? ^^; </li>
    <li>안녕하세요 ? ^^; </li>
    <li>안녕하세요 ? ^^; </li>
  </ul>
  <a href="http://www.naver.com">naver...</a>
</body>
</html>
```

jade문서



■ jade 언어(Language)



```
html
  head
    title My first Jade file
  body
    h2 This is the first jade file.
    hr
    ul
      - for (i=0; i < 3; i++)
        li 안녕하세요? ^^;
    a(href="http://www.naver.com") naver...
    div= currentTime
```

```
doctype html
html
  head
    title Index Page
  body (bgcolor = yellow)
    h1 # { name } .. !
    h2= description
    hr
    - for(var i = 0; i < 10; i++) {
      p
        a(href="http://hanbit.co.kr") Go To Hanbit Media # { i }
    - }
```

=== 작성지침(문법) ===

1. 반드시 **들여쓰기** 하여야 함!
(탭 또는 5칸씩 띄기)
2. 줄 바꾸면, **맨앞**을 Tag로 인식!
3. 자바스크립트 코드 앞에 -
(예, - for (i=0; i<3; i++)
4. 변수사용은 **# { 변수명 }**
=변수명
태그명과 = 사이에 공백없어야함
(예, # { currentTime }

변수값 셋팅은
render() 함수 를 호출하기
직전에 넘김

```
res.render('jade파일',  
  { currentTime:Date(); });
```

■ jade 언어(Language)

예제7) jademain.js

```
const http = require('http');
const fs = require('fs');
const jade = require('jade'); ← 외부모듈(템플릿엔진)

http.createServer(function (request, response) {
  if (request.method == 'GET') {
    if (request.url == '/') {
      fs.readFile('hello.jade', 'utf8', function (error, data) {
        const fn = jade.compile(data);
        response.writeHead(200, {'Content-Type': 'text/html; charset=utf8'});
        response.end(fn());
      });
    }
  }
}).listen(52273, function () {
  console.log('Server Running at http://127.0.0.1:52273 ...');
});
```

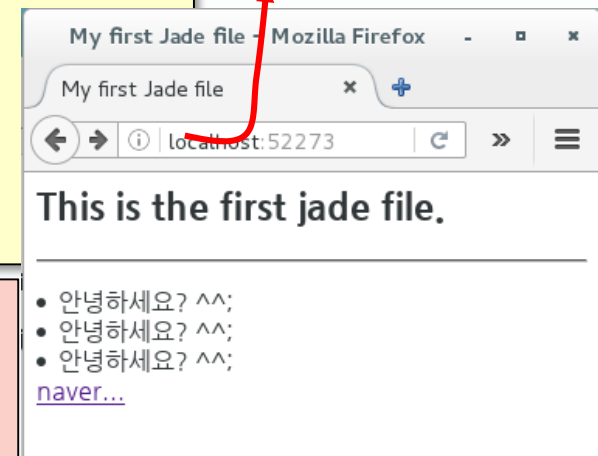
```
$ vi hello.jade
$ vi jademain.js
$ node jademain.js
Server Running at http://127.0.0.1:52273 ...
```

```
html
head
  title My first Jade file
body
  h2 This is the first jade file.
  hr
  ul
    - for (i=0; i < 3; i++)
      li 안녕하세요? ^^;
    a(href="http://www.naver.com") naver...
```

hello.jade

본인의
라즈베리파이
IP주소

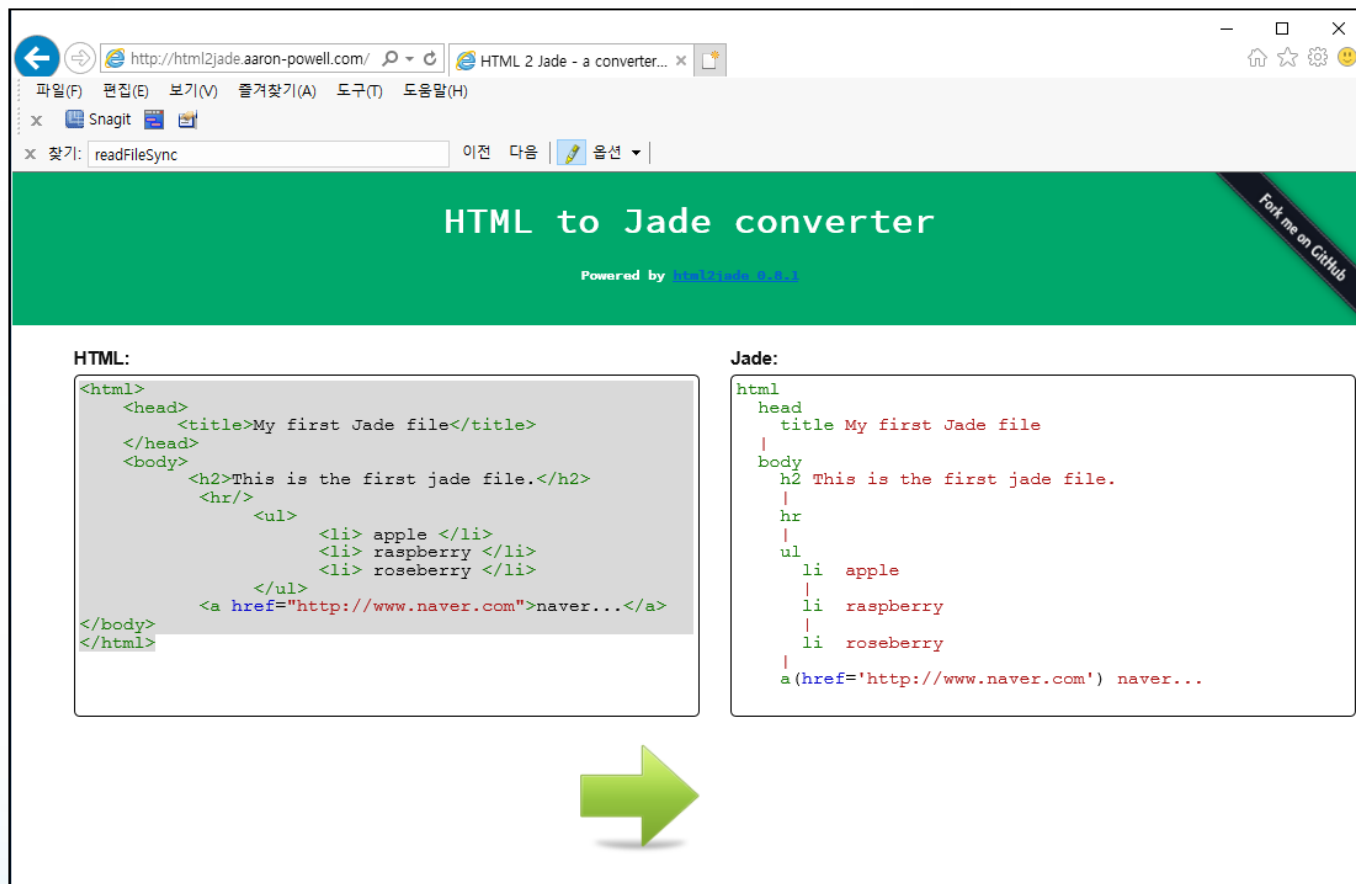
본인의
라즈베리파이
IP주소



■ jade 언어(Language)

☑ HTML to JADE (변환기)

예) <http://html2jade.aaron-powell.com/>



The screenshot shows a web browser window with the URL <http://html2jade.aaron-powell.com/>. The page title is "HTML to Jade converter" and it is powered by "html2jade 0.9.1". A banner on the right says "Fork me on GitHub".

The page displays two code blocks: "HTML:" and "Jade:". A large green arrow points from the HTML block to the Jade block, indicating the conversion process.

HTML:

```
<html>
  <head>
    <title>My first Jade file</title>
  </head>
  <body>
    <h2>This is the first jade file.</h2>
    <hr/>
    <ul>
      <li>apple </li>
      <li>raspberry </li>
      <li>roseberry </li>
    </ul>
    <a href="http://www.naver.com">naver...</a>
  </body>
</html>
```

Jade:

```
html
  head
    title My first Jade file
  |
  body
    h2 This is the first jade file.
    |
    hr
    |
    ul
      li apple
      |
      li raspberry
      |
      li roseberry
    |
    a(href='http://www.naver.com') naver...
```


■ 실습일지

☑ 실습2

☑ 실습3

