Nodejs – Basic Functions

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Contents

- 간단한 웹서버
- 주소 문자열과 요청 파라미터 다루기
- 이벤트 이해하기
- 파일 다루기

간단한 웹서버

간단한 웹서버

- http 모듈 이용 – 웹 서버 기능을 담당하는 서버 객체 생성
- server = createServer()
- server.listen()

```
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');
});

server.listen(port, hostname, () => {
    console.log(`Server running at http://${hostname}:${port}`);
});
```

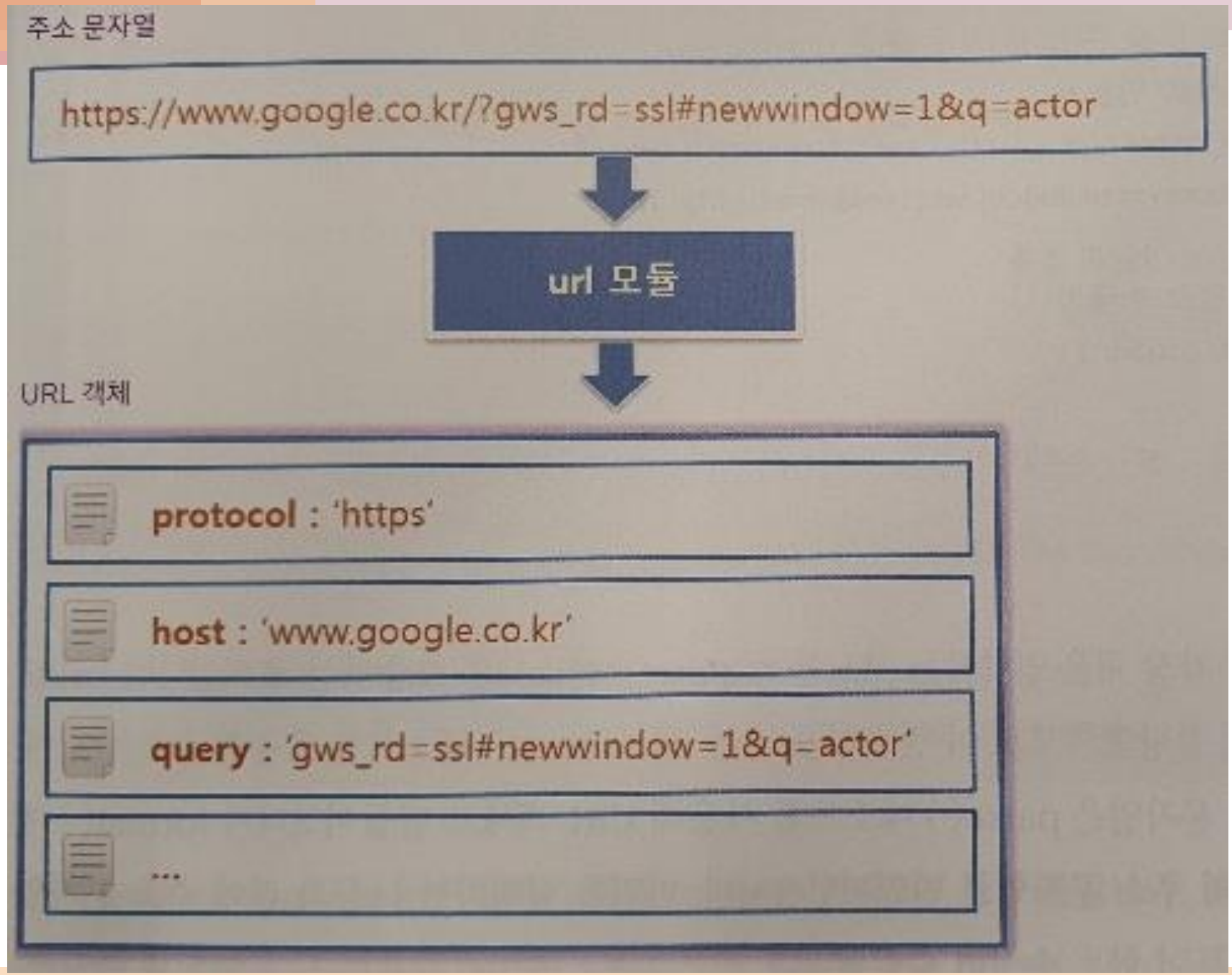
```
const http = require('http');

const port = 1000;

const server = http.createServer((req, res) => {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.end('Hello World!');
});

server.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

주소 문자열과

요청 파라미터 다루기

주소 문자열과 요청 파라미터 다루기

- url module: 주소 문자열을 url 객체로 변환
  - parse(): 주소 문자열 파싱 → url 객체
  - format(): url 객체 → 주소 문자열
- querystring module: 요청 파라미터 확인
  - parse(): 요청 파라미터 분석 → 객체
  - stringify(): 객체 안의 요청 파라미터 → 문자열

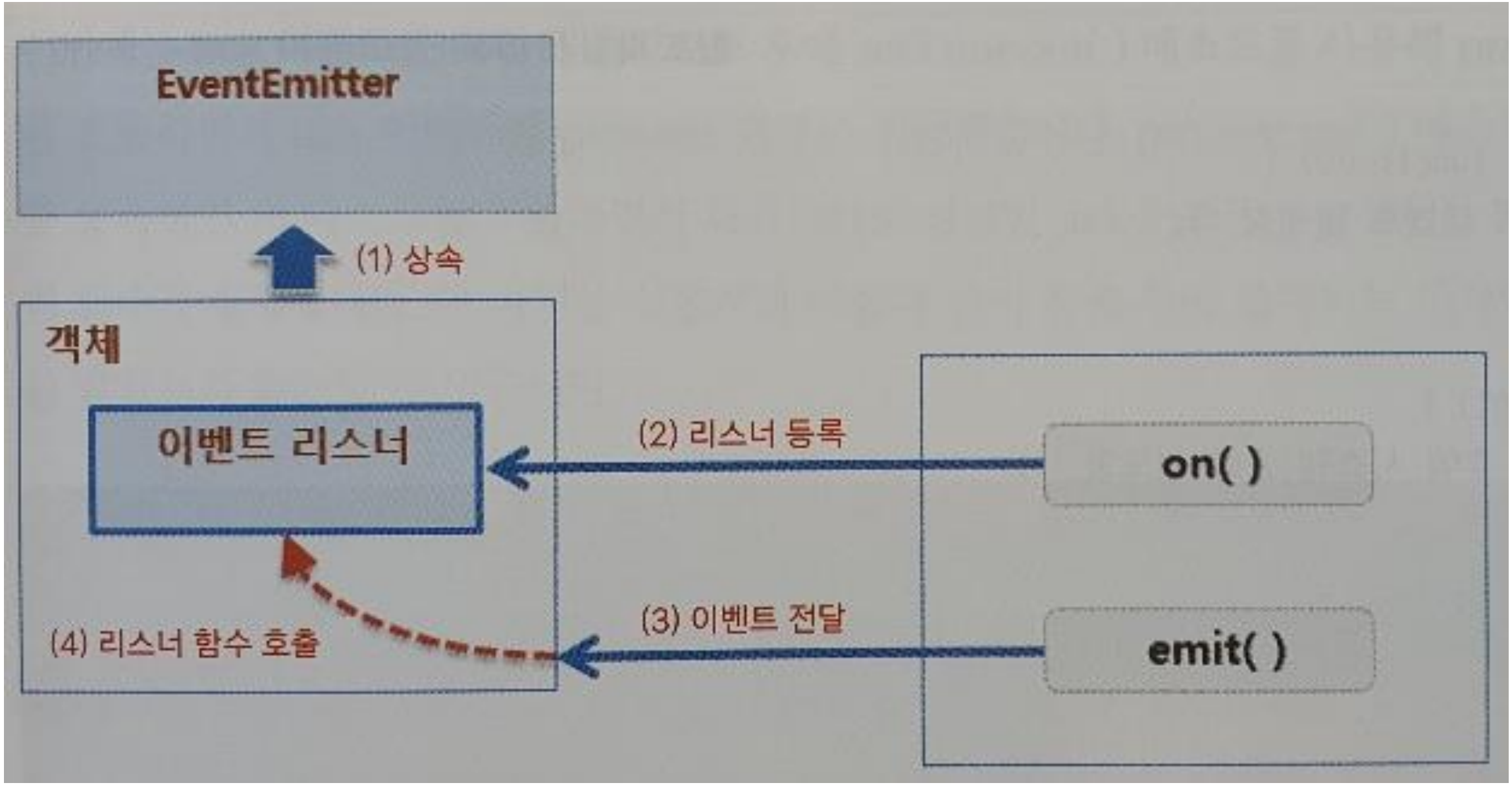

주소 문자열
https://www.google.co.kr/?gws_rd=ssl#newwindow=1&q=actor
url 모듈
URL 객체
protocol : 'https'
host : 'www.google.co.kr'
query : 'gws_rd=ssl#newwindow=1&q=actor'
...

```
let url = require('url');

let curURL =
url.parse('https://search.naver.com/search.naver?where=nexearch \
                    &sm=top_hty&fbm=1&ie=utf8&query=web+programming');
let curStr = url.format(curURL);

console.log('주소 문자열 : %s', curStr);
console.dir(curURL);

let qs = require('querystring');
let param = qs.parse(curURL.query);
console.dir(param);
console.log('요청 파라미터의 query : %s', param.query);
console.log('원본 요청 파라미터 : %s', qs.stringify(param));
```

이벤트 이해하기

이벤트 이해하기

- 비동기 방식 처리를 위해 서로 event를 보냄
- 노드 객체는 EventEmitter class를 상속 받음
  - on(event, listener)
  - once(event, listener)
  - removeListener(event, listener)


EventEmitter
(1) 상속
객체
이벤트 리스너
(2) 리스너 등록
on( )
(3) 이벤트 전달
emit( )
(4) 리스너 함수 호출

```
process.on('exit', function() {
    console.log('exit event 발생');
});

setTimeout(function() {
    console.log('2초 후에 시스템 종료 시도...');
    process.exit();
}, 2000);
```

```
process.on('tick', function(count) {
    console.log('count event 발생 : %s', count);
});

setTimeout(function() {
    console.log('2초 후에 tick 이벤트 전달 시도...');
    process.emit('tick', '2');
}, 2000);
```

이벤트 이해하기

- 서버 시작 → 변수 초기화 → 함수 선언 → 이벤트를 기다림
- event 모듈, EventEmitter 클래스: 이벤트와 이벤트 핸들러를 연결 (binding)

파일 다루기

파일 다루기

- 노드 파일 시스템
  - 파일 다루기 + 디렉토리 다루기
  - 동기 IO, 비동기 IO
- fs module: require('fs');
- file read/write
  - readFileSync(), writeFileSync()
  - readFile(), writeFile()

```
let fs = require('fs');

// read file - 동기식 IO
let data = fs.readFileSync('../Current/ex8-03.html', 'utf-8');
console.log(data);
console.log('file read request !!!');
```

```
let fs = require('fs');

fs.writeFile('./output.txt', 'Hello, world!\n',
function(err) {
    if (err) {
        console.log('Error: ' + err);
    }
});
```

```
let fs = require('fs');

// read file - 비동기식 IO
let data = fs.readFile('../Current/ex8-03.html', 'utf-8',
function(err, data) {
    if (err == null) {
        console.log(data);
    } else {
        console.log('error !!!');
    }
});
console.log('file read request !!!');
```

파일 다루기

- file을 직접 열고, 닫고, 읽고/쓰기
  - open(), read(), write(), close()
- directory 만들기, 삭제하기
  - mkdir()
  - rmdir()

공부한 것들

공부한 것들

- 간단한 웹서버
- 주소 문자열과 요청 파라미터 다루기
- 이벤트 이해하기
- 파일 다루기