



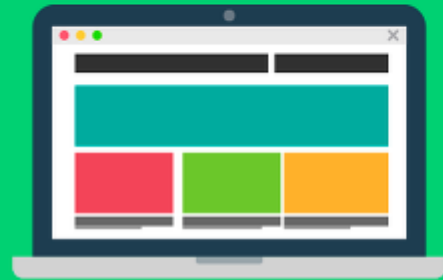
It's Your Life

with





Back-End



FRONTEND



BACKEND

추상화의 비밀 이야기

그리고

싶은 대로

그려여

글 이문민 그림 노성욱 감수 유재길 박남희





체시, 바르셀로나 '초대형' 선수 영입 임박...최종 합의 완료

[포르투=이규학]프렌키 더 용을 품는 클럽은 체시로 보인다. 더 용의 협상이 최종 합의에 이르렀다.바르셀로나 소식에 정통한 제라르 로메로 기자는 11일(한국...)

포르투 · 라리가



8년 전 방출한 FW에게 혼쭐난 맨유..."이럴 거면 다시 데려와! "

[사진] ©Gettyimages(무단전재 및 재배포 금지)[OSEN=고성환 기자] 맨체스터 유나이티드의 전설 리오 퍼디난드(44)가 대니 웰벡(32, 브라이...)

OSEN · 프리미어리그



'아무리 생각해도 이유를 모르겠다' KIA는 왜 피렐라와 승부했나

2022 KBO리그 삼성라이온즈와 KIA타이거즈의 경기가 10일 대구삼성라이온즈파크에서 열렸다. 삼성 피렐라가 10회말 끝내기 적시타를 치고 환호하고 있다..."

스포츠조선 · KBO리그



키움, SSG 대항마였는데...한 달 만에 격차는 2.5경기→10경기

4일 서울 구로구 고척스카이돔에서 열린 프로야구 '2022 신한은행 SOL KBO리그' SSG 랜더스와 키움 히어로즈의 경기, 키움 3루수 송성문이 10회초..."

뉴스1 · KBO리그



③ 3명 늘어난 엔트리, 벤투호에 승선할 태극전사 26명은?

FIFA 월드컵 최종 엔트리 23명→26명으로 확대손흥민황의조·김민재 등 '불박이'에 엄원상·정우영 합류 가능성↑ 손흥민의 인사(서울=연합뉴스) 이지은 기자..."

연합뉴스 · 대표팀

스포츠 창작자 영상

1 / 4 < >



[영국방송] "손흥민 놔두면 당하지!" BBC MOTD 토...

조회 31,043 · 한준TV



역시 국대 출신! 김지영2 프로의 드라이버샷과 특별...

조회 3,047 · 스포츠W - Spor...



손흥민급 감아차기ㄷ 추가시간에 터진 역대급 원...

조회 2,121 · 박측공 Football ...



뉴욕 메츠를 이끄는 공포의 슬라이더 트리오

조회 3,398 · 김형준의 야구야...



"어썸 김!" 외친 홈팬들 앞 2루타 김하성, 판독 끝 날...

조회 16,236 · 이현우의 MLBTV



"나 잡으려면 감독이랑 단장 찰라"

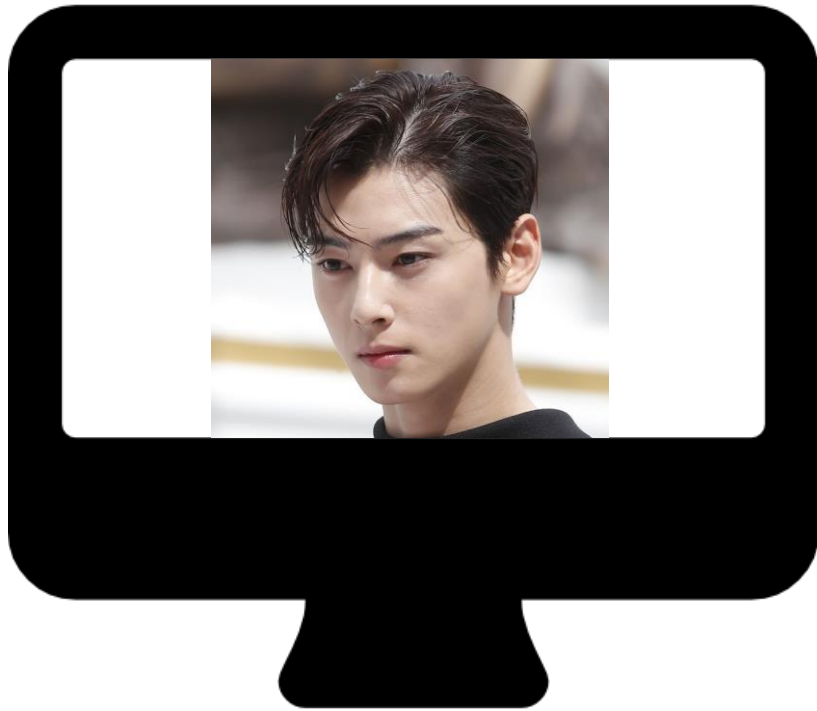
조회 10,807 · 농구대학







프론트-엔드



백-엔드



요청



클라이언트

서버

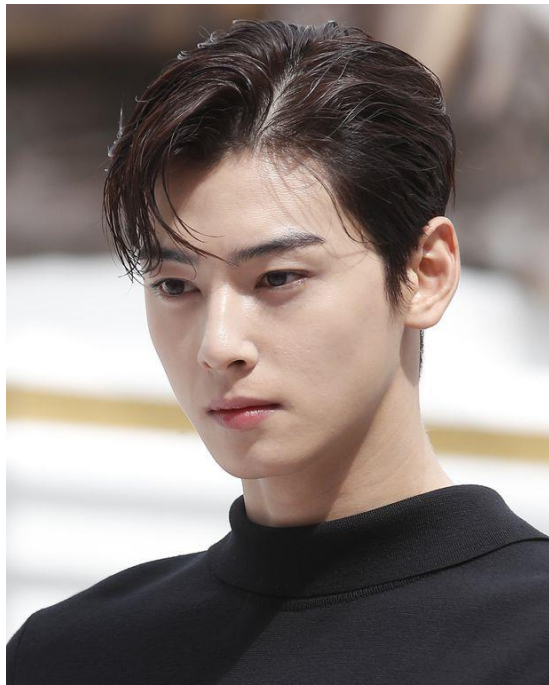


응답



남성 연예인

남성 연예인





Back-End





남성 연예인 목록

API



A

Application

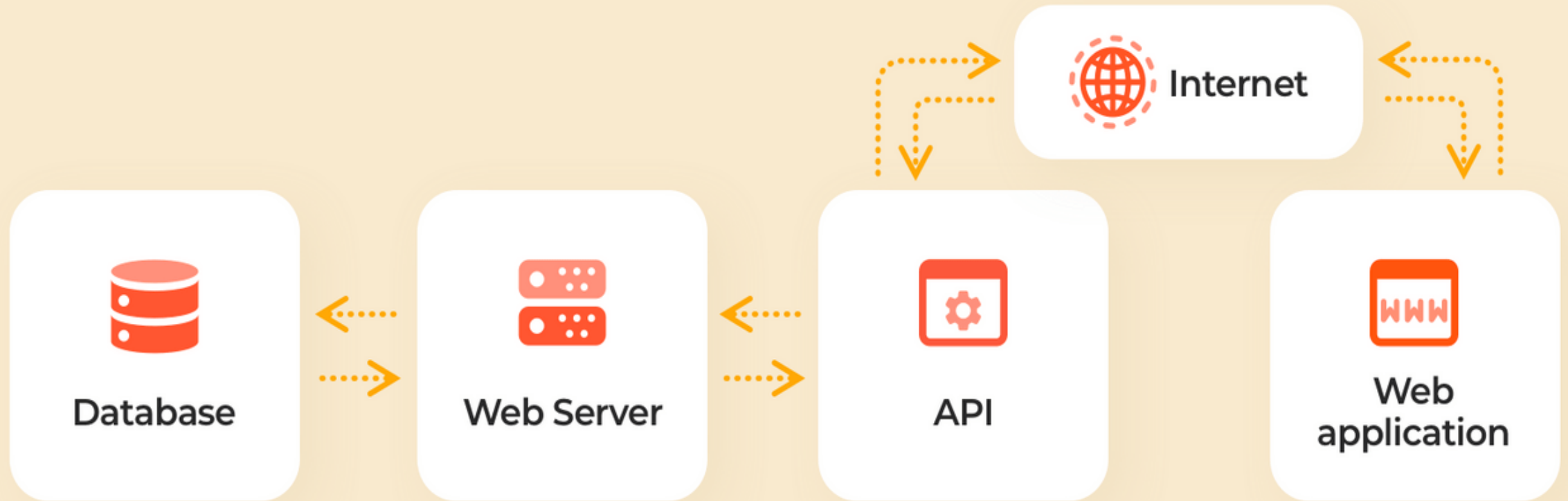
P

Programming

I

Interface

What is an API?



API(Application Programming Interface)



- 서로 다른 두 어플리케이션이 통신을 할 수 있도록 해주는 일종의 규약
- 예를 들어 백엔드 서버는 Java 로 되어 있고, 프론트엔드 페이지는 JS 로 되어 있다면!?
- 마치 영어만 가능한 사람과 카자흐스탄어만 가능한 사람이 대화 하는 것!
- 이때 API는 서로가 서로의 말로 통역을 해주는 것
- 영어만 가능한 사람이 카자흐말로 번역해서 데이터를 전달하는 것이죠!







API 주소

<http://api.tetzKB.com/male-stars>

https://apis.data.go.kr/B551011/KorService/areaBasedList?serviceKey=rfaoGpiapHFqOcUT6bqfERRY1WVxzOdOpEC3ChyAFPEfONdSMdRVNETTJKRhqTbPuZ2krpG2mQJMXDbyG74RA%3D%3D&numOfRows=687&pageNo=1&MobileOS=ETC&MobileApp=TripLog&_type=json&listYN=Y&arrange=B&contentTypeId=12&areaCode=1

http://api.tetzkdt.com/male-stars



```
[  
  {  
    name: "차은우",  
    image: "http://file.tetzKB.com/malestars/차은우.png"  
  },  
  {  
    name: "고수",  
    image: "http://file.tetzKB.com/malestars/고수.png"  
  },  
  {  
    name: "이동욱",  
    image: "http://file.tetzKB.com/malestars/이동욱.png"  
  }  
]
```



```
nodejs-express-middleware > JS server.js > ...
```

```
1  const express = require('express');
2  const app = express();
3
4  // Express is just a series of middleware calls, hmmm...
5  // Here, only middleware is registered
6  // nothing like app.get(/* ... params */) or app.post(/* ... params */)
7  // but when we open http://localhost:3000 in browser, then look at the console
8  // the output is:
9  // First middleware GET
10 // Second middleware GET
11 app.use(
12   (req, res, next) => {
13     console.log('First middleware', req.method);
14     return next();
15   },
16   (req, res, next) => {
17     console.log('Second middleware', req.method);
18     return next();
19   },
20   (req, res, next) => {
21     res.send(`Hello from ${req.method}
22     request to ${req.url}`);
23   }
24 );
25
26 app.listen(3000, () => console.log('Server started on: http://localhost:3000'));
27
```



{ REST : API }

Representational State Transfer API



<https://apis.data.go.kr/B551011/KorService/areaBasedList?serviceKey=rfaoGpiapHFqOcUT6bqfERRxy1WVxzOdOpEC3ChyAFPEfONdSMdRVNETTJKRhqTbPuZ2krpG2mQJMXDbyG74RA%3D%3D&numOfRows=687&pageNo=1&MobileOS=ETC&MobileApp=TripLog&type=json&listYN=Y&arrange=B&contentType=12&areaCode=1>

{JSON}
JavaScript Object Notation



```
const tetzObj = {
  name: '이효석',
  hobbies: ['Lol', 'Tekken', 'Bicycle'],
};

const strTetzObj = JSON.stringify(tetzObj);
console.log(typeof strTetzObj);
console.log(strTetzObj);
console.log(strTetzObj.name);

console.log('-----');

const objTetzBack = JSON.parse(strTetzObj);
console.log(typeof objTetzBack);
console.log(objTetzBack);
console.log(objTetzBack.name);
```

```
string
{"name":"이효석","hobbies":["Lol","Tekken","Bicycle"]}
undefined
-----
object
{ name: '이효석', hobbies: [ 'Lol', 'Tekken', 'Bicycle' ] }
이효석
```



Front-End

http://api.tetzkdt.com/male-stars



```
[  
  {  
    name: "차은우",  
    image: "http://file.tetzKB.com/malestars/차은우.png"  
  },  
  {  
    name: "고수",  
    image: "http://file.tetzKB.com/malestars/고수.png"  
  },  
  {  
    name: "이동욱",  
    image: "http://file.tetzKB.com/malestars/이동욱.png"  
  }  
]
```



대표적 방법

두 가지!

프론트에서 백엔드에게 정보를 요청(SPA)



```
async function fetchData() {
  const tokenResponse = await fetch(
    `https://kauth.kakao.com/oauth/token?grant_type`,
    {
      method: 'POST',
      headers: {
        'Content-type': 'application/x-www-form-urlencoded;charset=utf-8',
      },
    }
  );

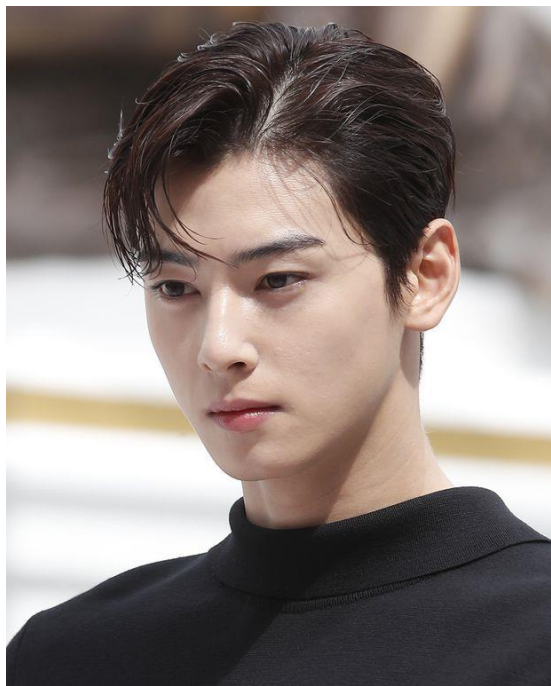
  if (tokenResponse.status === 200) {
    const tokenData = await tokenResponse.json();
    const userResponse = await fetch(`https://kapi.kakao.com/v2/user/me`, {
      method: 'POST',
      headers: {
        Authorization: `Bearer ${tokenData.access_token}`,
        'Content-type': 'application/x-www-form-urlencoded;charset=utf-8',
      },
    });
  }
}
```


백엔드에서 데이터를 보내서 그려주기(SSR)



```
<body>
  <section class="male-star">
    <% for(let i=0; i < maleStar.length; i++) { %>
      <p>
        <%=maleStar[i].name%>
      </p>
      
    <% } %>
  </section>
</body>
```

남성 연예인





여성 연예인





```
nodejs-express-middleware > JS server.js > ...
```

```
1  const express = require('express');
2  const app = express();
3
4  // Express is just a series of middleware calls, hmmm...
5  // Here, only middleware is registered
6  // nothing like app.get(/* ... params */) or app.post(/* ... params */)
7  // but when we open http://localhost:3000 in browser, then look at the console
8  // the output is:
9  // First middleware GET
10 // Second middleware GET
11 app.use(
12   (req, res, next) => {
13     console.log('First middleware', req.method);
14     return next();
15   },
16   (req, res, next) => {
17     console.log('Second middleware', req.method);
18     return next();
19   },
20   (req, res, next) => {
21     res.send(`Hello from ${req.method}
22     request to ${req.url}`);
23   }
24 );
25
26 app.listen(3000, () => console.log('Server started on: http://localhost:3000'));
27
```

http://api.tetzkdt.com/female-stars



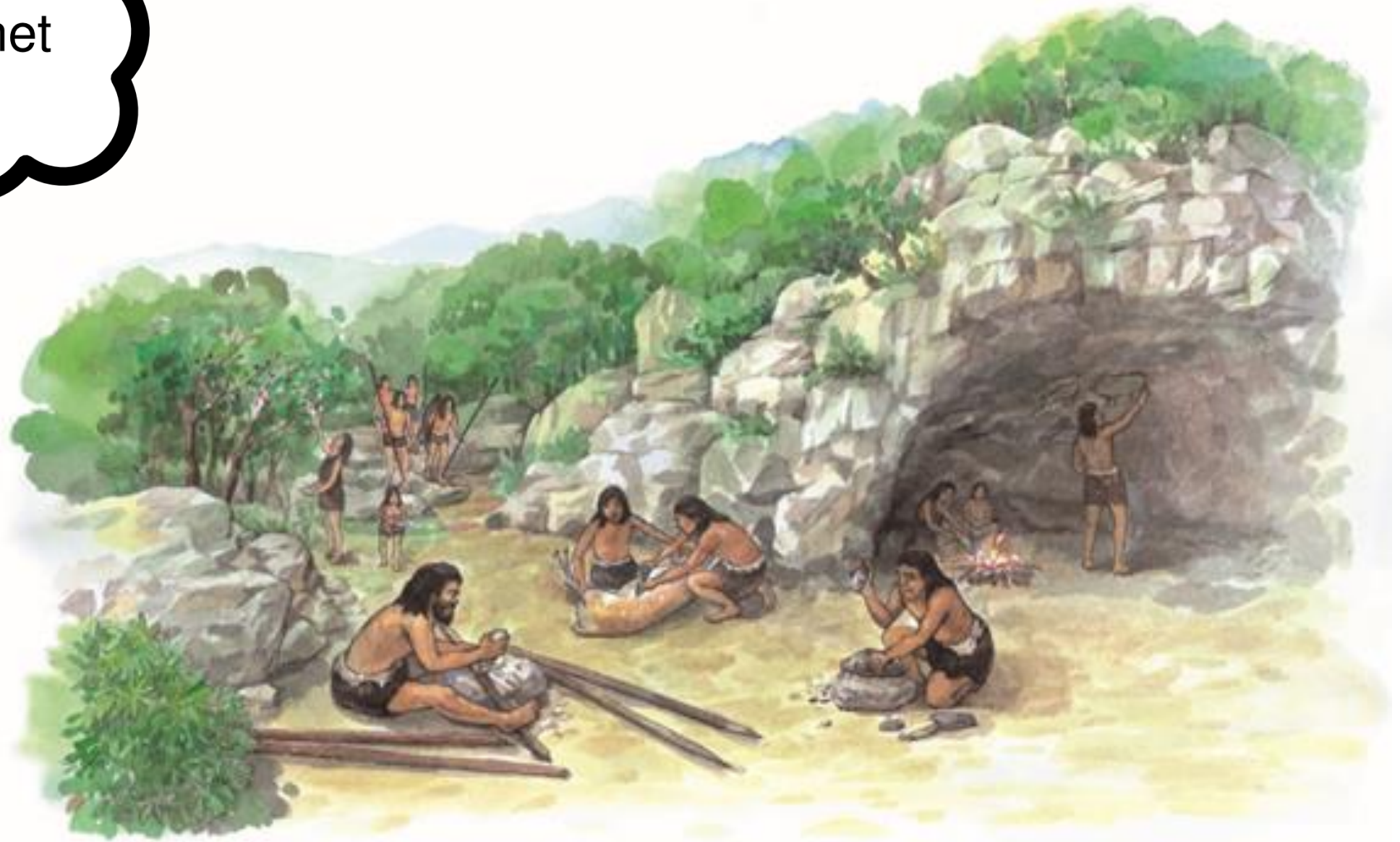
```
[  
  {  
    name: "아이린",  
    image: "http://file.tetzKB.com/malestars/아이린.png"  
  },  
  {  
    name: "카리나",  
    image: "http://file.tetzKB.com/malestars/카리나.png"  
  },  
  {  
    name: "김태희",  
    image: "http://file.tetzKB.com/malestars/김태희.png"  
  }  
]
```



Node-JS



Internet





신문검색 | 국외검색



서치센터 | 광고안내

맞춤정보 서비스, **MyNAVER**
메일메일 핫&콜, **WebToday**

삼성생명 오픈 이벤트
?

네이버를 나의 홈페이지로...
네오넷의 부동산 급매물 서비스!

NAVER 검색

분류에서 | 검색
☐ 제목에서 검색 [Quick Help](#) [확장검색](#)

NAVER 분류

- [건강/의학](#)
건강관리, 병원, 의학
- [교육](#)
대학, 시험, 자격증, 유학
- [뉴스/미디어](#)
신문, 잡지, 텔레비전
- [레크리에이션](#)
스포츠, 게임, 여행, 레저
- [사업/경제](#)
기업, 취업, 온라인쇼핑
- [사회과학](#)
경제학, 사회학, 언어학
- [사회/문화](#)
결혼, 기관/단체
- [연예/오락](#)
연예인, 영화, 음악, 유머
- [인문/예술](#)
디자인, 인문과학, 박물관
- [자연과학](#)
공학, 컴퓨터과학, 생물학
- [정부/공공기관](#)
정치, 한국정부, 국제기구
- [지역정보](#)
대한민국, 서울, 국가
- [참고자료](#)
도서관, 사람찾기 사전
- [컴퓨터](#)
인터넷, S/W, O/S, 통신

NAVER 소식

네이버 포털서비스 오픈! **MyNAVER**
네이버 웹문서수 3,141,838 건

NAVER 추천

New! 크리스마스
[행사, 이벤트](#) [여행](#) [온라인쇼핑](#)
[영화, 비디오](#) [취업](#) [채용](#) [투자](#) [재테크](#)



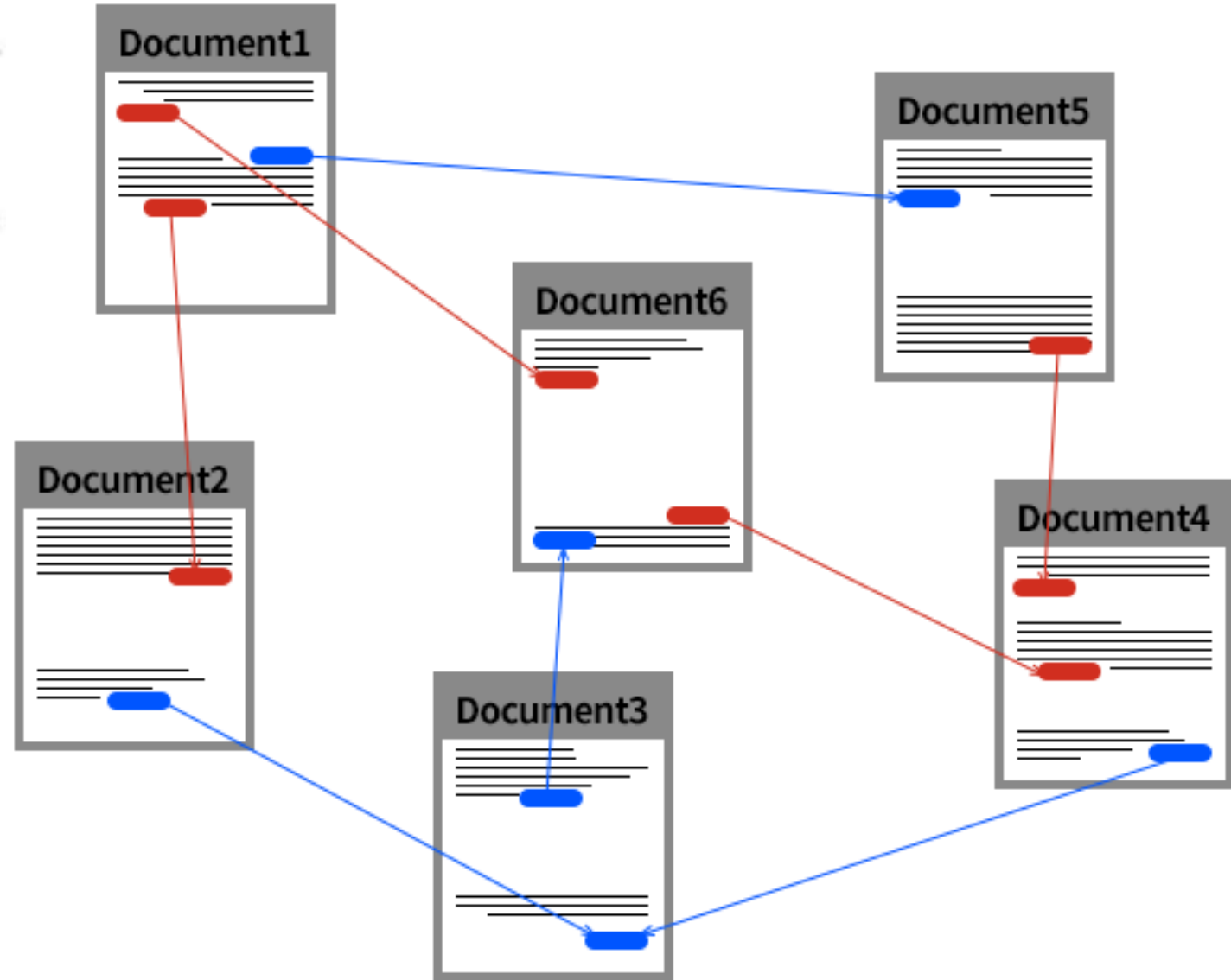
AT & T 전화접속서비스

[도움말](#) | [새소식](#) | [홈페이지등록](#) | [다른검색사이트](#)





ComputerHope.com





인간의 욕심은 끝이 없고



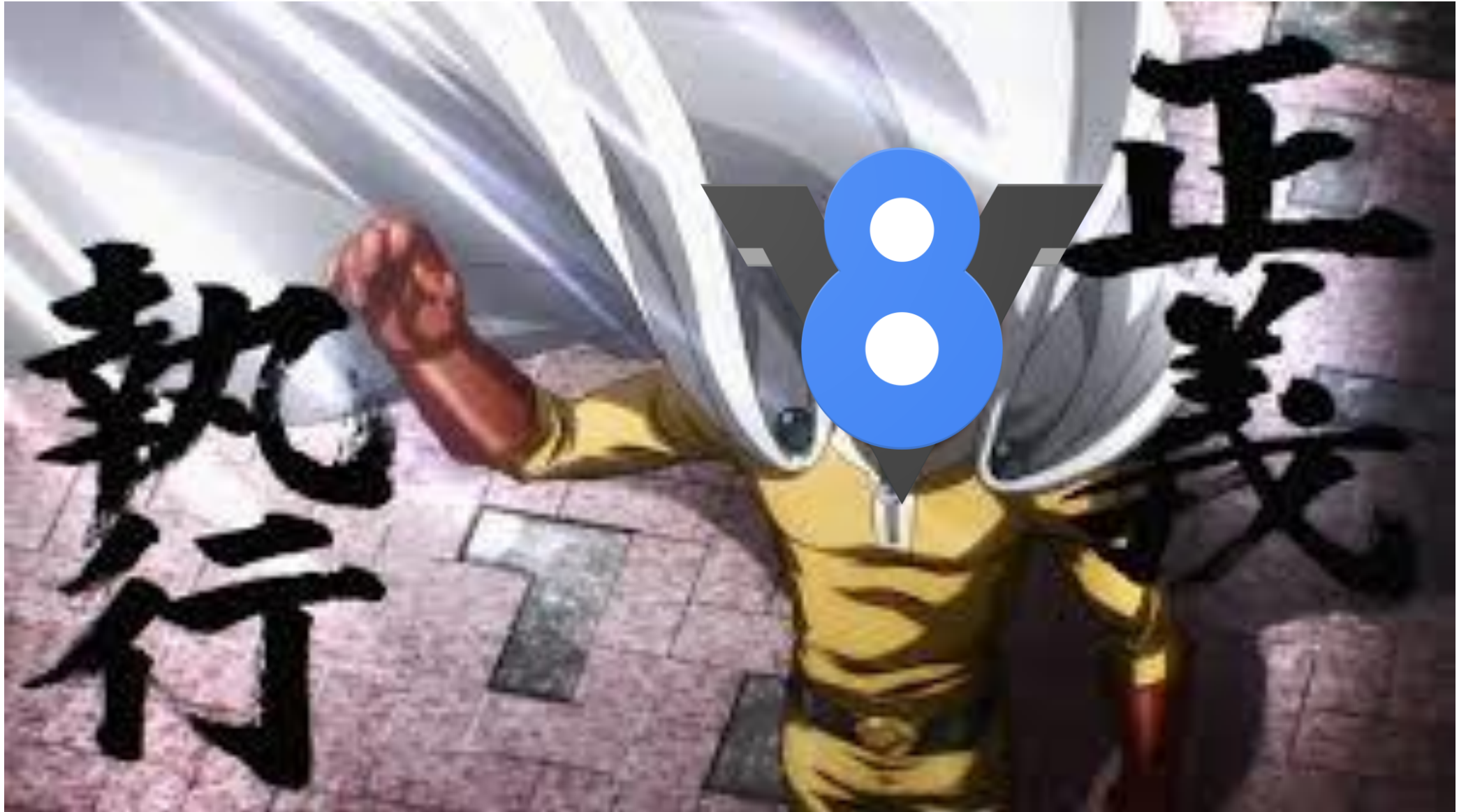
JavaScript





2008





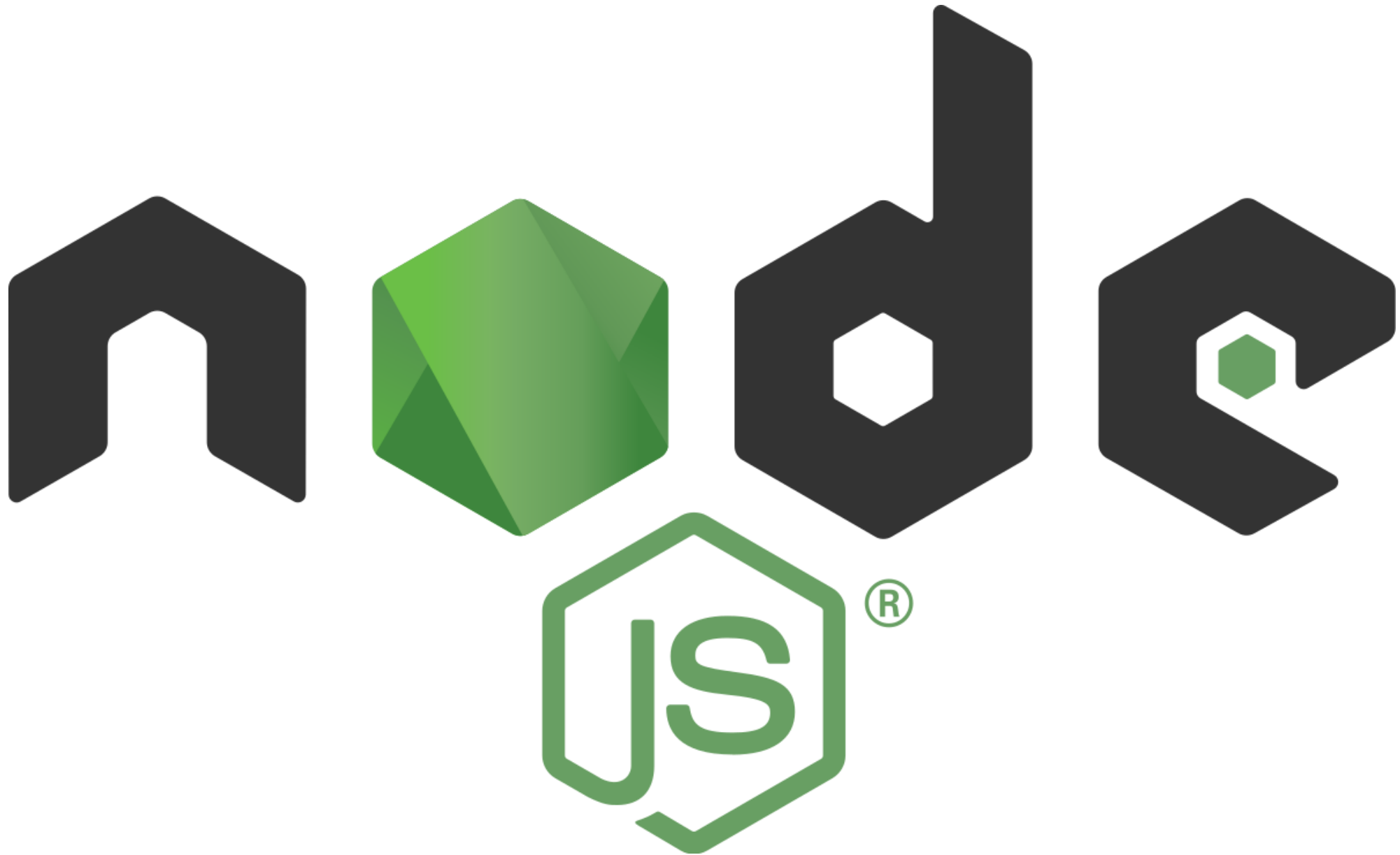


open source



1. 스크립트 언어라 어느 환경에서도 동작
2. 성능이 빠르고 좋음
3. Javascript 를 쓰는 개발자가 많음
4. Ryan Dahl이 서버용 기능을 개발!











대한민국 No.1
롯데렌터카

쉽고빠른



NAVER

kakao

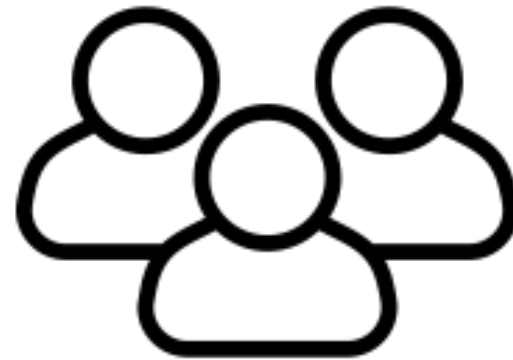


 KB 국민은행

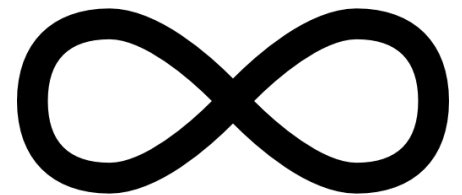
coupang

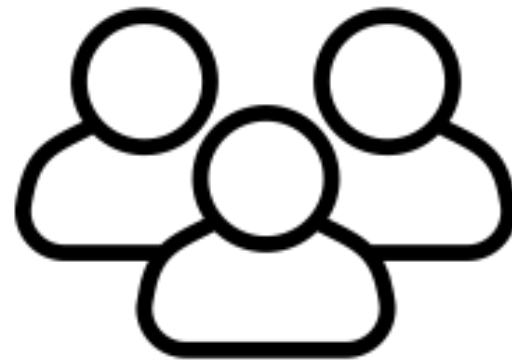
 spring[®]

배달의민족 



x





x

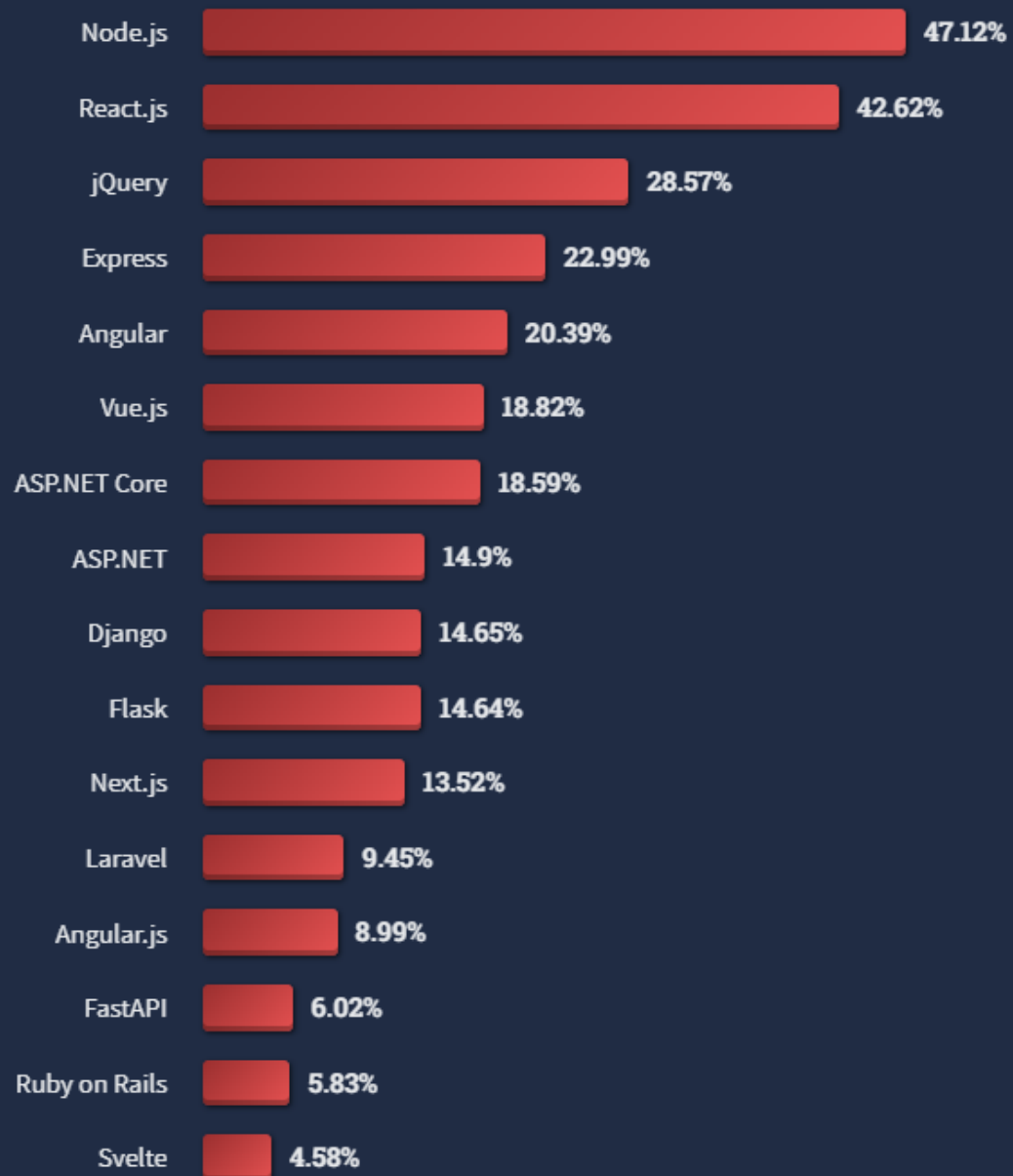
?



2022



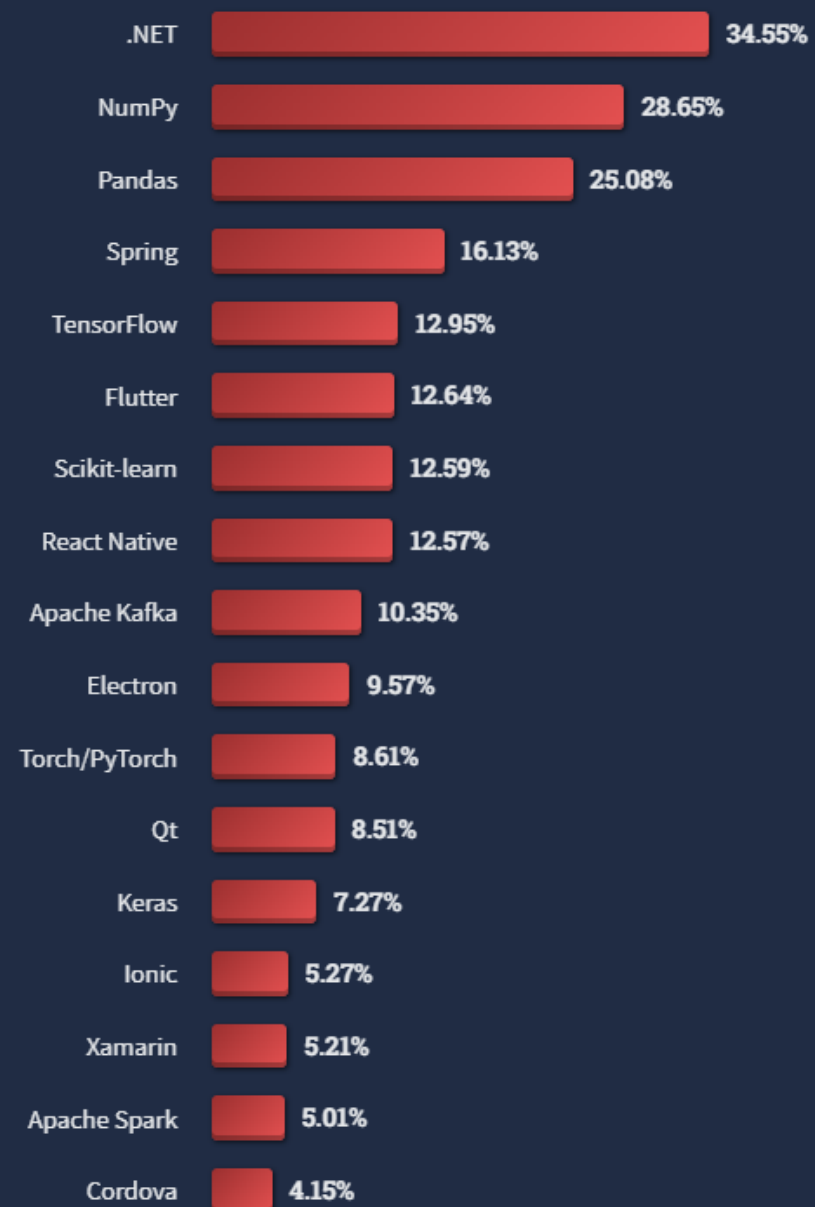
Web frameworks and technologies



2022



Other frameworks and libraries





<https://coinpan.com> 코인판 on 2018-01-09





NPM?

(Node Package Manager)



1,614,651 개의 패키지!

By the numbers

Packages

1,614,651

Downloads · Last Week

30,687,522,522

Downloads · Last Month

126,688,929,038





NPM(Node Package Manager)?



NPM(Node Package Manager)?



패키지 관리가 필요하겠죠?



- 이러한 패키지를 관리하는 것이 바로
- Node Package Manager → NPM 입니다!



패키지 관리가 필요하겠죠?



- 프로젝트를 위해 어떤 패키지를 설치 해야 할까요?
- 패키지 버전에 따라 충돌이 발생할 수 있지 않을까요?
- 이러한 문제를 하나의 JSON 파일로 관리해 줍니다!

패키지 관리 시작하기!



- npm init -y

```
tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/node-setting
$ npm init -y
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Wrote to C:\Users\tetz\Desktop\node-setting\package.json:

{
  "name": "setting",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

패키지 관리 시작하기!



- Package.json 파일이 생성 됩니다!
- Node.js 관련 package의 모든 정보가 여기에 들어 갑니다!

```
{
  "name": "setting",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Package.json 파일 알아보기!



```
{
  "name": "setting",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```


Package.json 파일 알아보기!



- npm run test

```
tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/node-setting
$ npm run test
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> setting@1.0.0 test
> echo "Error: no test specified" && exit 1

"Error: no test specified"

tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/node-setting
$ █
```

Package.json 파일 변경하기!



```
{
  "scripts": {
    "test": "echo \"Hello, Node.js\""
  }
}
```

- npm run test

```
tetz@DESKTOP-P7Q4OLL MINGW64 ~/Desktop/node-setting
$ npm run test
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> test
> echo "Hello, Node.js"

"Hello, Node.js"
```

Package 설치하기!



- npm install left-pad

```
lhs@DESKTOP-86MUCGC MINGW64 /d/git/kdt-5th-proj1 (main)
$ npm install left-pad
npm WARN deprecated left-pad@1.3.0: use String.prototype.padStart()

added 1 package, and audited 174 packages in 647ms

12 packages are looking for funding
  run `npm fund` for details

4 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

```
{
  "dependencies": {
    "left-pad": "^1.3.0"
  }
}
```

Package 삭제하기



- npm uninstall left-pad

```
lhs@DESKTOP-86MUCGC MINGW64 /d/git/kdt-5th-proj1 (main)  
$ npm uninstall left-pad
```

```
removed 1 package, and audited 173 packages in 678ms
```

```
12 packages are looking for funding  
  run `npm fund` for details
```

```
4 high severity vulnerabilities
```

```
To address all issues (including breaking changes), run:  
  npm audit fix --force
```

```
Run `npm audit` for details.
```

```
{  
  "dependencies": {}  
}
```

Package.json 만 활용하기!



- 빈 폴더에 package.json 파일만 카피를 해봅시다
- Npm install
- 실행을 하면 package.json 를 참고하여 해당 package 가 설치 됩니다!
- 따라서, 추후에 협업 시 node_modules 폴더를 보낼 필요 없이 package.json 만 공유가 되면 편리하게 관리가 가능합니다~!



Node.js 가 뭐였죠?



Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

다운로드 - Windows (x64)

16.16.0 LTS

안정적, 신뢰도 높음

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

18.7.0 현재 버전

최신 기능

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

LTS 일정은 [여기서](#) 확인하세요

- 자바 스크립트 파일을 실행 시켜 봅시다!

Node.js 가 뭐였죠?



- main.js 파일 만들기

```
console.log("Hello, Node.js");
```

- Node main.js

```
tetz@DESKTOP-P7Q40LL MINGW64 ~/Desktop/node-setting  
$ node main.js  
Hello, Node.js
```



JSON?

Callback?

JSON? → JavaScript Object Notation



```
1  {
2    "string": "Hi",
3    "number": 2.5,
4    "boolean": true,
5    "null": null,
6    "object": { "name": "Kyle", "age": 24 },
7    "array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],
8    "arrayOfObjects": [
9      { "name": "Jerry", "age": 28 },
10     { "name": "Sally", "age": 26 }
11  ]
12 }
13
```

JSON? → JavaScript Object Notation



- JS의 객체 처럼 속성과 해당 속성(Key)에 대한 값(Value)을 쌍으로 가지는 객체형 자료형 입니다!

```
const tetz = {  
  name: '이효석',  
  isOld: true,  
};
```

JS의 객체

```
{  
  "name": "이효석",  
  "isOld": "true"  
}
```

JSON

JSON 을 실제로 봅시다!



```
const tetzObj = {
  name: '이효석',
  hobbies: ['Lol', 'Tekken', 'Bicycle'],
};

const strTetzObj = JSON.stringify(tetzObj);
console.log(typeof strTetzObj);
console.log(strTetzObj);
console.log(strTetzObj.name);

console.log('-----');

const objTetzBack = JSON.parse(strTetzObj);
console.log(typeof objTetzBack);
console.log(objTetzBack);
console.log(objTetzBack.name);
```

```
string
{"name":"이효석","hobbies":["Lol","Tekken","Bicycle"]}
undefined
-----
object
{ name: '이효석', hobbies: [ 'Lol', 'Tekken', 'Bicycle' ] }
이효석
```



{JSON}
JavaScript Object Notation



JSON 형태로 전달을 안하면!?

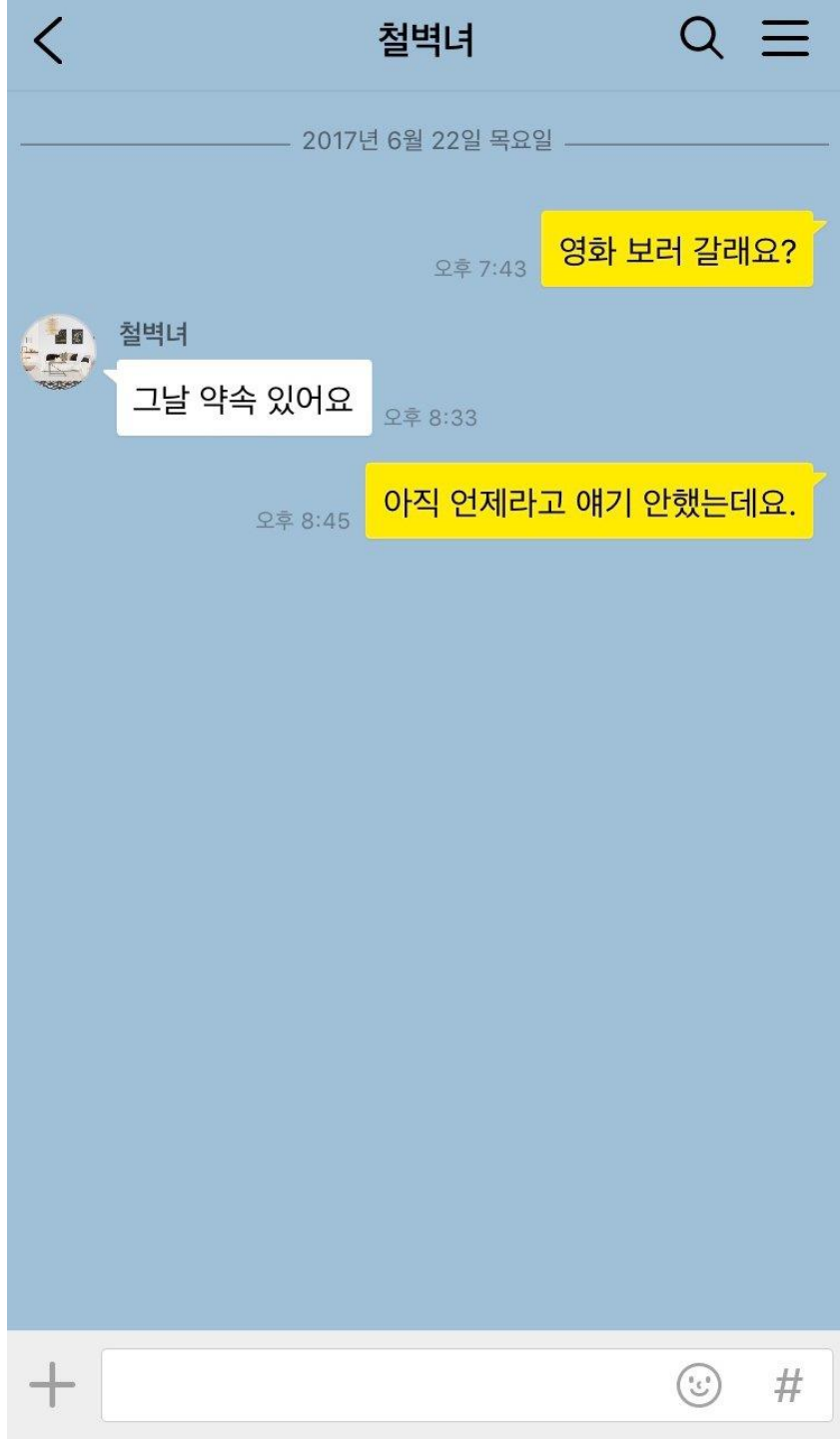


```
app.use('/', (req, res) => {  
  const str = '안녕하세요. 여기는 백엔드 입니다!';  
  res.send(str);  
});
```

backend/server.js

- 이렇게 문자열을 바로 전달하면!?

```
✖ ▶ Uncaught (in promise) SyntaxError: Unexpected token   VM130:1  
  '안', "안녕하세요. 여기는 백엔드 입니다!" is not valid JSON
```



비동기 통신

동기 / 비동기 프로그래밍



- 동기

- 하나의 요청이 완료가 되면 그 후에 다음 요청을 실행

- 비동기

- 하나의 요청이 완료가 되기 전에 다음 요청을 실행



SYNCHRONOUS VS ASYNCHRONOUS

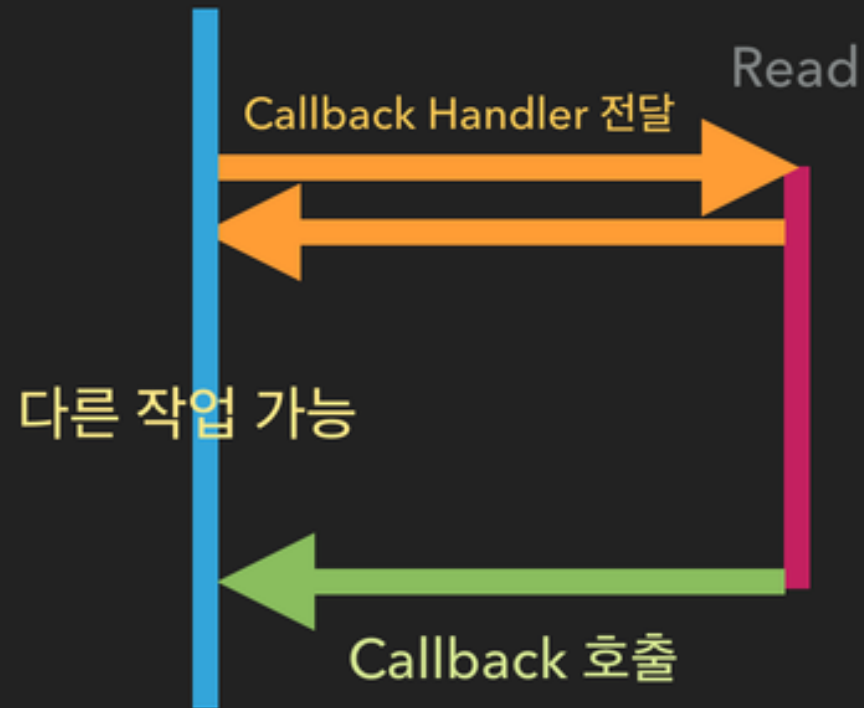
Sync 작업 완료 후 리턴

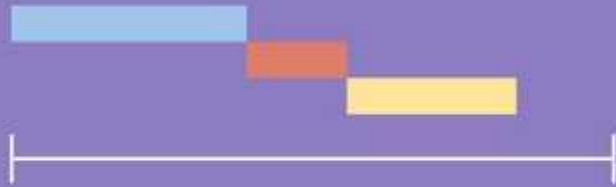
App Thread



Async 바로 리턴 + Callback

App Thread





Synchronous



Asynchronous



(a) 동기



(b) 비동기

비동기 프로그래밍



- 따라서 Node.js 는 컴퓨터의 각각의 코어에 다른 일을 시킬 수 있습니다!
- 즉, 동시에 여러가지 일을 한꺼번에 수행할 수 있습니다!
- 따라서 Node.js 는 가볍고 빠른 서버를 구성할 수 있죠!
- 하지만 이러한 비동기적 특성이 항상 좋은 것은 아닙니다!

비동기 프로그래밍



- 서버(백엔드) 통신을 할 때에는 동기적(절차적)으로 문제를 해결해야합니다.
- 예를 들어서 로그인 요청을 보내고 서버에서 로그인 결과에 대한 결과 값을 받은 다음에 로그인 처리를 해야겠죠?
- 하지만 Node.js 는 비동기적 특성을 가지므로 + 아무리 인터넷이 빨라 졌어도 node.js 의 처리 속도 보다는 느리기 때문에, 기존과 동일하게 코드를 작성하면 로그인 결과를 받기도 전에 node.js 는 로그인 처리를 하려고 할 것이고 이는 문제가 될 것입니다!

구원자, Callback!



- 위와 같은 비동기 프로그래밍의 문제를 해결하기 위해서 JS 에서 지원하는 방식이 바로 Callback 방식 입니다~!
- Callback 함수는 특정 함수의 작업이 종료 되어야만 다른 함수를 호출 하기 때문에 사용자가 원하는대로 절차적 프로그래밍이 가능해 집니다!
- 그럼 일단 간단한 코드로 확인해 보시죠!



```
function buySync(item, price, quantity) {  
  console.log(`${item} 상품을 ${quantity} 개 골라서 점원에게 주었습니다.`);  
  console.log('계산이 필요합니다.');
```

```
  const total = price * quantity;  
  return total;  
}  
  
function pay(tot) {  
  console.log(`${tot} 원을 지불하였습니다.`);  
}
```

```
const totalMoney = buySync('포켓몬빵', 1000, 5);
```

```
pay(totalMoney);
```

backend/callback.js