

WALE

S T U D I O



와플스튜디오 Backend Seminar

Instructors:

최한결 @Hank-Choi, 강지혁 @Jhvictor4

2021.08.28.(토) 10:00 - Zoom

0차시 세미나

Contributor 변다빈 @bdv111

서울대학교 앱/웹개발 동아리 와플스튜디오

Table of Contents

Introducing Backend

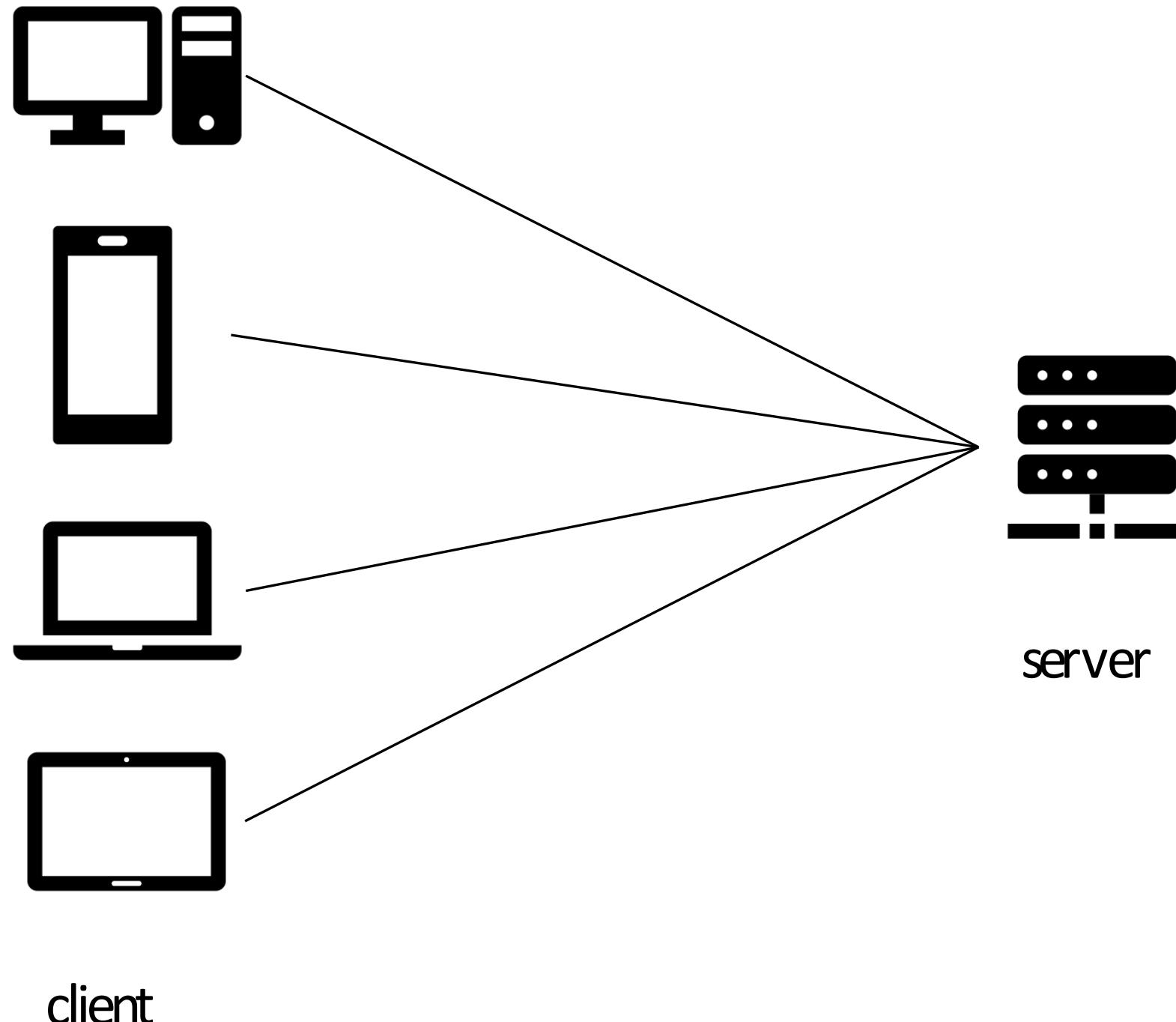
Database

Spring Boot / Django

Creating Basic API with



Client & Server



클라이언트

다른 프로그램에게 서비스를 요청하는 프로그램

서버

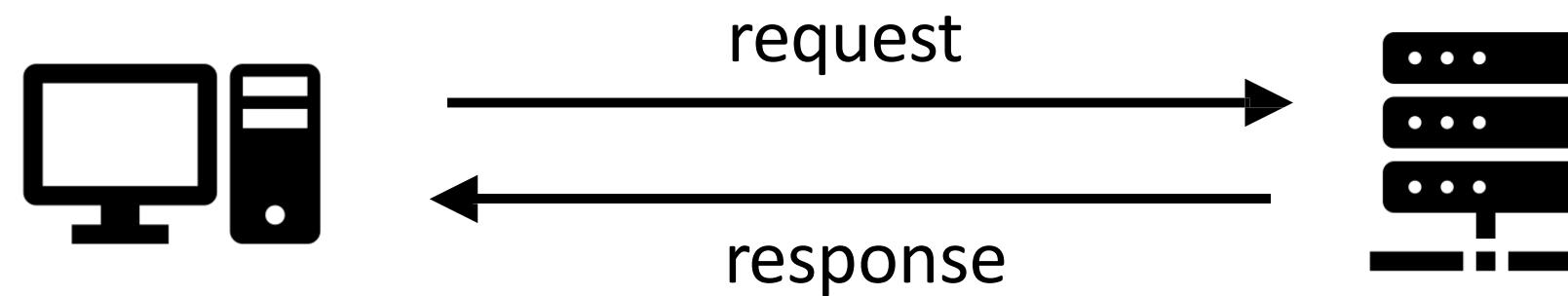
요청에 대해 필요한 응답을 해주는 프로그램

HTTP

HTTP(Hypertext Transfer Protocol)

프로토콜이란 상호 간에 정의한 규칙

Server와 client 간에 데이터를 주고받기 위해 정의



Http에 의해 서버는 항상 client가 요청할 때만
데이터를 전송

Request : request method + URL + header + body

Response: Status code + header + body

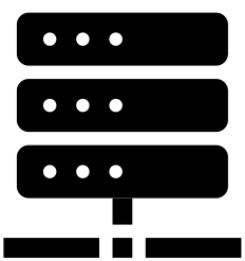
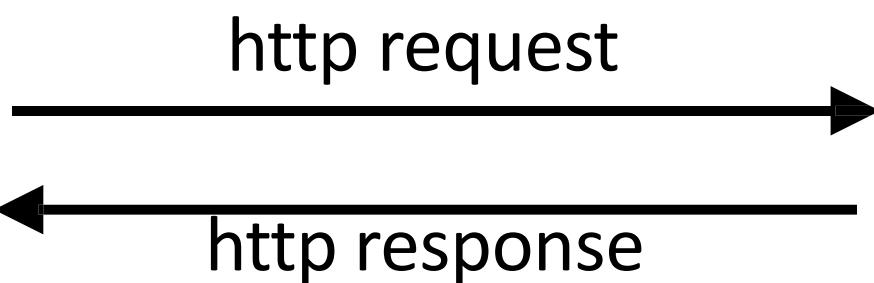
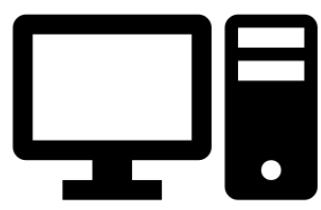
Request method: GET,POST,PUT,DELETE,PATCH

Status code: 200(ok), 401(unauthorized), 404(not found) ...

HTTP 예시

GET <https://www.naver.com>

request Method + URL + header + body



Status code + header + body

Status: 200 ok

```
<!doctype html>
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

HTTP 예시

The screenshot shows a REST client interface with the following details:

- Request URL:** www.naver.com
- Method:** GET
- Headers:** (7)
- Body:** none
- Response Status:** 200 OK
- Time:** 80 ms
- Size:** 214.53 KB
- Content Type:** HTML
- Content Preview:** (The preview shows a snippet of an HTML page from Naver, including a link to a post viewer, category names like '눈호강 여행', and a title '미국 샌프란시스코 풍경맛집! 최고의 전망포인트 TOP 5'. The numbers 2607 through 2612 are listed vertically on the left side of the preview area.)

HTTP Status code

10X - Informational

100 Continue

20X - Success

200 OK, 201 CREATED, 202 ACCEPTED

30X - Redirection

301 Moved Permanently , 303 See other, 307 Temporary Redirect

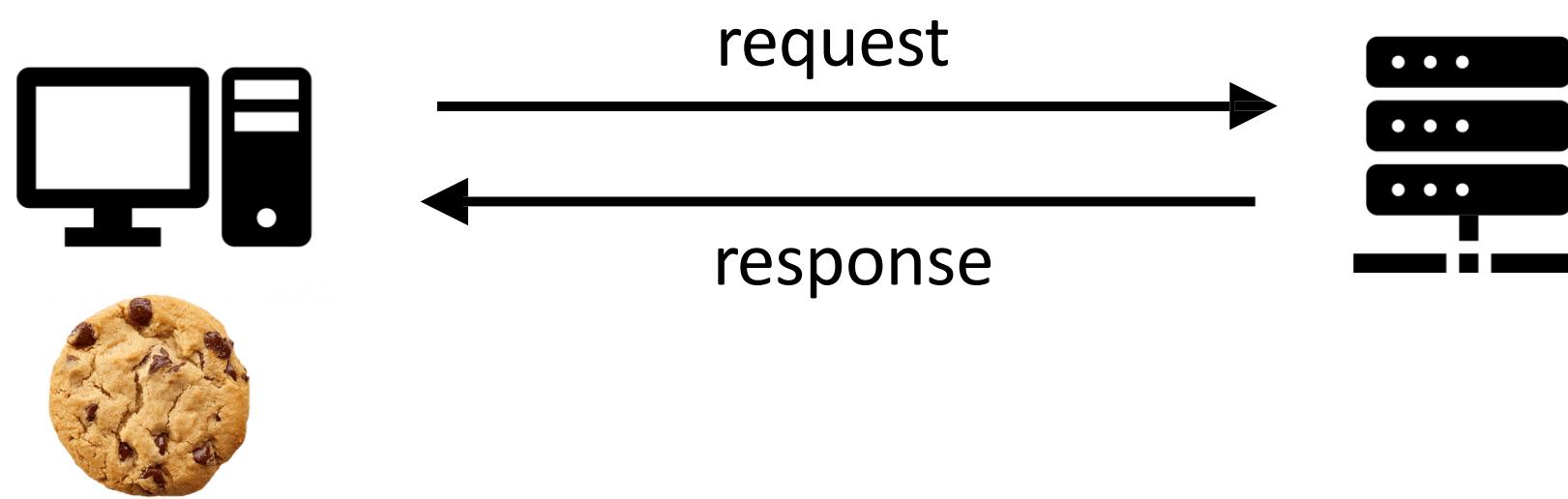
40X – Client Error

400 Bad Request, 401 Unauthorized, 404 Not Found, 409 Conflict

50X – Server Error

500 Internal Server Error -> 일으키지 말자

Cookie & Session



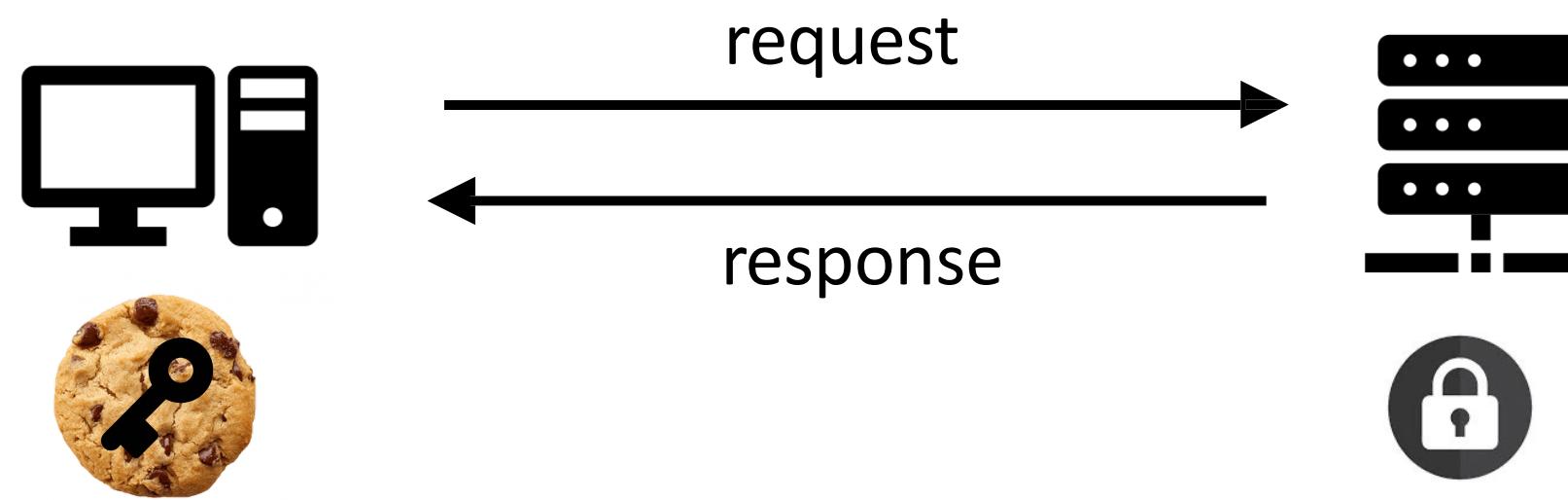
쿠키(Cookie)

서버에서 전달해 사용자의 로컬 컴퓨터에 저장하는 작은 데이터

쿠키는 같은 웹사이트를 방문할 때마다 request header에 추가되어 전달

쿠키를 통해 서버에게 client 정보와 상태를 전달 가능

Cookie & Session



세션(Session)

데이터를 사용자의 브라우저에 쿠키 형태가 아닌
접속한 서버 DB에 정보를 저장

클라이언트는 Session id를 쿠키로 메모리에 저장

Session id를 client의 메모리에 저장하므로
브라우저가 종료되면 사라지게 됨

API

(1) 자신의 정보 가져오기

- 자신의 정보를 가져옵니다.

endpoint

- GET /api/v1/user/me/

status code

- 200: OK
- 401: UNAUTHORIZED

result

example	description
<pre>1 { 2 "id": 3, 3 "username": "bdv111", 4 "email": "null@test.com" 5 }</pre>	<pre>1 { 2 "id": User id, 3 "username": 사용자 이름, 4 "email": 사용자 이메일 5 }</pre>

API

Application Programming Interface

특정 데이터를 공유할 경우 어떤 방식으로 정보를 요청해야 하는지, 그리고 어떤 데이터를 제공 받을 수 있을지에 대한 규격

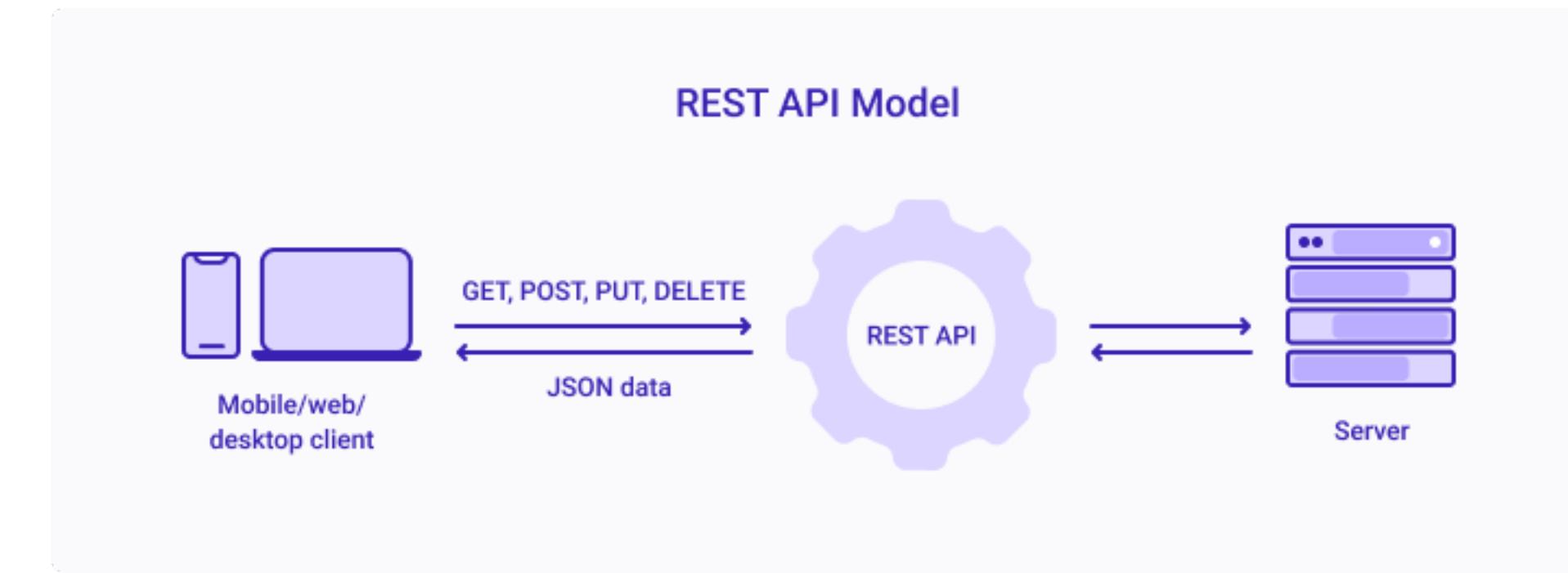
프로그램 구현을 알지 못해도 명세에 따라 프로그램 간의 데이터 이동을 가능하게 만듦

Front, back개발자는 API를 정리한 문서를 통해 소통

REST API

“Representational State Transfer” By Roy Fielding (Http Author)

소프트웨어 아키텍처의 일종
- 네트워킹을 더 잘 수행하기 위한 방법론



Principles

1. Uniform Interface : Resource [URI]에 대해 통일되고 한정적인 인터페이스 [HTTP Method]로 접근
 2. Stateless : 단순히 들어오는 요청을 수행할 뿐, 작업을 위한 상태 정보는 모름 -> cc. [멱등성](#)
 3. Client-Sever 구조 : 서버는 API 제공, 클라는 컨텍스트 관리 등으로 작업 분리 & 의존성 해소
- + Cacheable, Self-descriptiveness, Layer Architecture (proxy, load balancing ...)

Practice

- * Postman에서 실제 동작 확인하기
- * [Instagram API](#)
- * [API Docs](#)

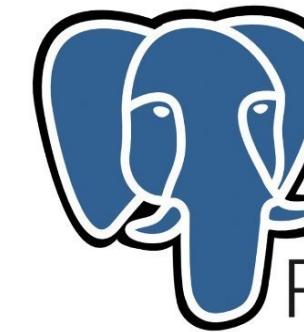
Database

Server

-> Serve What?

DBMS

->



-> RDBMS, NoSQL , etc.

-> MySQL로 실습

A screenshot of a Microsoft Excel spreadsheet titled "백엔드설문". The table has columns labeled A through H. Column A contains row numbers from 1 to 26. Column B contains dates and times. Columns C through H contain various numerical and categorical data. The last column, H, contains Korean text such as "내 학년은... (학번보다는 수? 내가 Backend 세미나를 했던 적이 있는가?)".

H2	A	B	C	D	E	F	G	H
1	타임스탬프	내 컴퓨터(개발을 위해 일상? 나는 Python / Kotlin 에 대한) 나는 관계형 데이터베이스에 나는 프로그래밍 또는 개발? 내 전공은...					내 학년은... (학번보다는 수? 내가 Backend 세미나를 했던 적이 있는가?)	
2	2021. 8. 26 오후 9:25:32	MacOS	1	1	1 타 전공	3학년		
3	2021. 8. 26 오후 9:27:40	MacOS	3	1	3 컴퓨터공학부 주전공	1학년		
4	2021. 8. 26 오후 9:27:44	MacOS	3	1	2 컴퓨터공학부 주전공	2학년		
5	2021. 8. 26 오후 9:28:08	MacOS	2	3	3 컴퓨터공학부 부전공 / 복수전공 / 복수전공 4학년 이상			
6	2021. 8. 26 오후 9:28:38	Windows	1	1	1 컴퓨터공학부 부전공 / 복수전공 / 복수전공 1학년			
7	2021. 8. 26 오후 9:31:11	Windows	1	2	2 타 전공	3학년		
8	2021. 8. 26 오후 9:31:29	Windows	2	2	3 컴퓨터공학부 주전공	4학년 이상		
9	2021. 8. 26 오후 9:31:47	MacOS	3	2	3 컴퓨터공학부 주전공	1학년		
10	2021. 8. 26 오후 9:32:38	Ubuntu (Linux)	2	1	3 타 전공	1학년		
11	2021. 8. 26 오후 9:34:16	Ubuntu (Linux)	1	1	3 타 전공	4학년 이상		
12	2021. 8. 26 오후 9:35:31	Windows	4	1	2 컴퓨터공학부 주전공	2학년		
13	2021. 8. 26 오후 9:43:30	MacOS	2	1	2 타 전공	1학년		
14	2021. 8. 26 오후 9:45:22	Ubuntu (Linux)	4	2	3 타 전공	3학년		
15	2021. 8. 26 오후 9:46:48	Windows	2	1	1 타 전공	4학년 이상		
16	2021. 8. 26 오후 9:48:21	MacOS	1	4	4 컴퓨터공학부 주전공	1학년		
17	2021. 8. 26 오후 9:49:39	Ubuntu (Linux)	3	1	2 타 전공	4학년 이상		
18	2021. 8. 26 오후 9:49:53	Windows	5	3	3 컴퓨터공학부 주전공	4학년 이상		
19	2021. 8. 26 오후 9:55:13	Ubuntu (Linux)	3	3	3 컴퓨터공학부 주전공	2학년		
20	2021. 8. 26 오후 9:56:28	Ubuntu (Linux)	3	3	3 컴퓨터공학부 주전공	2학년		
21	2021. 8. 26 오후 9:58:59	Ubuntu (Linux)	2	1	2 컴퓨터공학부 주전공	2학년		
22	2021. 8. 26 오후 10:00:33	MacOS	1	1	2 타 전공	4학년 이상		
23	2021. 8. 26 오후 10:07:11	MacOS	4	1	2 컴퓨터공학부 주전공	1학년		
24	2021. 8. 26 오후 10:11:07	Windows	2	2	2 컴퓨터공학부 주전공	1학년		
25	2021. 8. 26 오후 10:14:02	MacOS	4	2	3 컴퓨터공학부 주전공	1학년		
26	2021. 8. 26 오후 10:17:18	Ubuntu (Linux)	4	3	4 컴퓨터공학부 주전공	2학년		

RDB

Relational DataBase

-> 상식적인 이해

기본키(Primary Key)

외부키(Foreign Key)

선수번호	이름	팀 코드	포지션	등번호	키
1	김남일	K03	DF	33	177
2	박지성	K07	MF	7	178
3	이영표	K02	MF	22	176

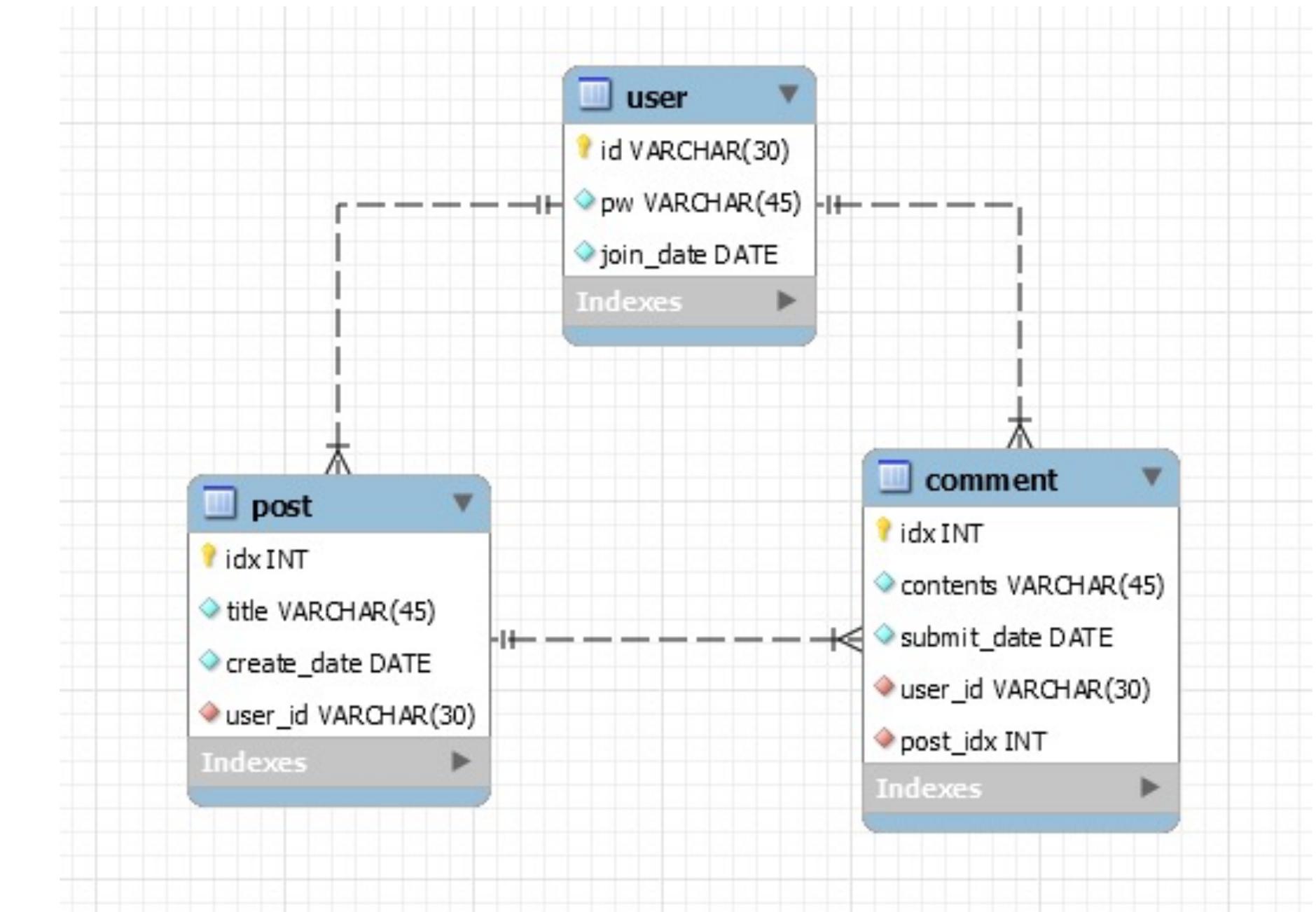
〈선수 테이블〉

기본키(Primary Key)

팀 코드	팀 명	연고지
K03	스틸러스	포항
K07	드래곤즈	전남
K02	블루윙즈	수원

〈구단 테이블〉

[그림 II-1-5] 테이블의 정규화



SQL

RDBMS에서 사용하는 '프로그래밍 언어'

DDL

종류	역할
CREATE	데이터베이스, 테이블 등을 생성하는 역할을 합니다.
ALTER	테이블을 수정하는 역할을 합니다.
DROP	데이터베이스, 테이블을 삭제하는 역할을 합니다.
TRUNCATE	테이블을 초기화시키는 역할을 합니다.

DML

종류	역할
SELECT	데이터를 조회하는 역할을 합니다.
INSERT	데이터를 삽입하는 역할을 합니다.
UPDATE	데이터를 수정하는 역할을 합니다.
DELETE	데이터를 삭제하는 역할을 합니다.

DCL

종류	역할
GRANT	특정 데이터베이스 사용자에게 특정 작업에 대한 수행권한 부여 합니다.
REVOKE	특정 데이터베이스 사용자에게 특정 작업에 대한 수행 권한을 박탈, 회수 합니다.
COMMIT	트랜잭션의 작업을 취소 및 원래로 복구하는 역할을 합니다.
ROLLBACK	트랜잭션의 작업을 취소 및 원래대로 복구하는 역할을 합니다.

실제 동작 확인하기 DataGrip
실제 확인하기 csv, tsv 파일

DB transaction

데이터베이스 작업의 단위

- 작업이란 : DB 상태를 변화시키는 일들
- 안전하게 수행됨을 보장할 수 있어야



Atomicity : 원자성. 트랜잭션은 작업의 최소 단위로서, 일부만 반영될 수 없음

Consistency : 일관성. 같은 상태에서 시작된 동일한 트랜잭션은 언제나 같은 결과를 도출

Isolation : 독립성. 처리 중인 트랜잭션끼리는 서로 간섭할 수 없다

Durability : 영구성. 성공한 트랜잭션의 결과는 영구적으로 반영되어야 한다

To Be Continued ...

Framework

Request -> Find data -> Do Some Stuff -> Serve Data (Send Response)

How?

```
younis@younis-linuxhint:~$ curl -X POST https://example.com/
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>411 - Length Required</title>
  </head>
  <body>
    <h1>411 - Length Required</h1>
  </body>
</html>
younis@younis-linuxhint:~$ █
```

Open Connection

Parse Request

Parse Parameters

Do some Business Logic

Build Response

Send it to Client

Close Connection

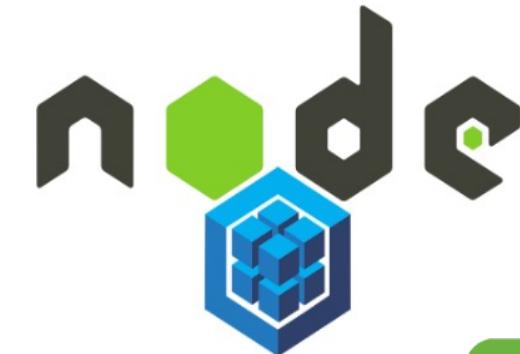


Framework

ANNOYING

Business Logic에만
집중하고 싶다.

Framework



With



간편해지는 유지 보수

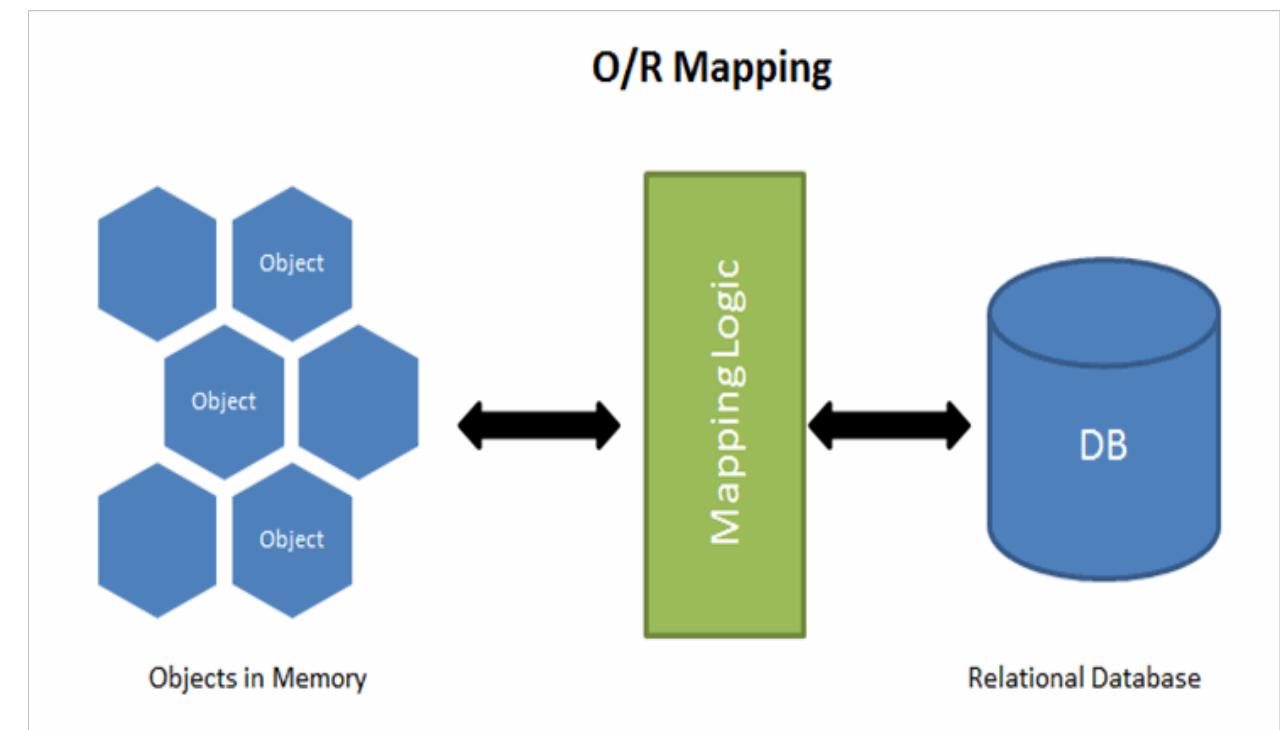
코드 스타일의 평준화

개발 생산성 UP

+ 인증, 직렬화 등의 다양한 편의기능 제공

ORM

- Object Relational Mapping
- DB 접근을 추상화 -> 더 쉽게 이용 가능
- Focus on business logic
- 재사용 및 유지보수 / 다양한 DBMS에 대응





장점

1. 성능이 좋아 대규모 트래픽 관리에 용이
2. 모듈화가 쉬워 대규모 프로젝트에 적합
3. 내장 서버를 사용해 간단하게 빌드 / 배포 가능
4. 많은 레퍼런스 / 사용자 풀 (특히 한국에서 bb)

단점

1. DB 지식이 조금 더 필요하다
2. DI 등의 개념이 러닝 커브 높음

NAVER

kakao

배달의민족

LINE

WA#LE
STUDIO



장점

1. Batteries-Included; 압도적인 개발 생산성
2. 모듈화 및 Customizing 용이
3. Testing / Health Check / 보안 등 풍부한 내장 기능 ex. [Django Admin](#)
4. 아주아주 친절한 공식 문서



단점

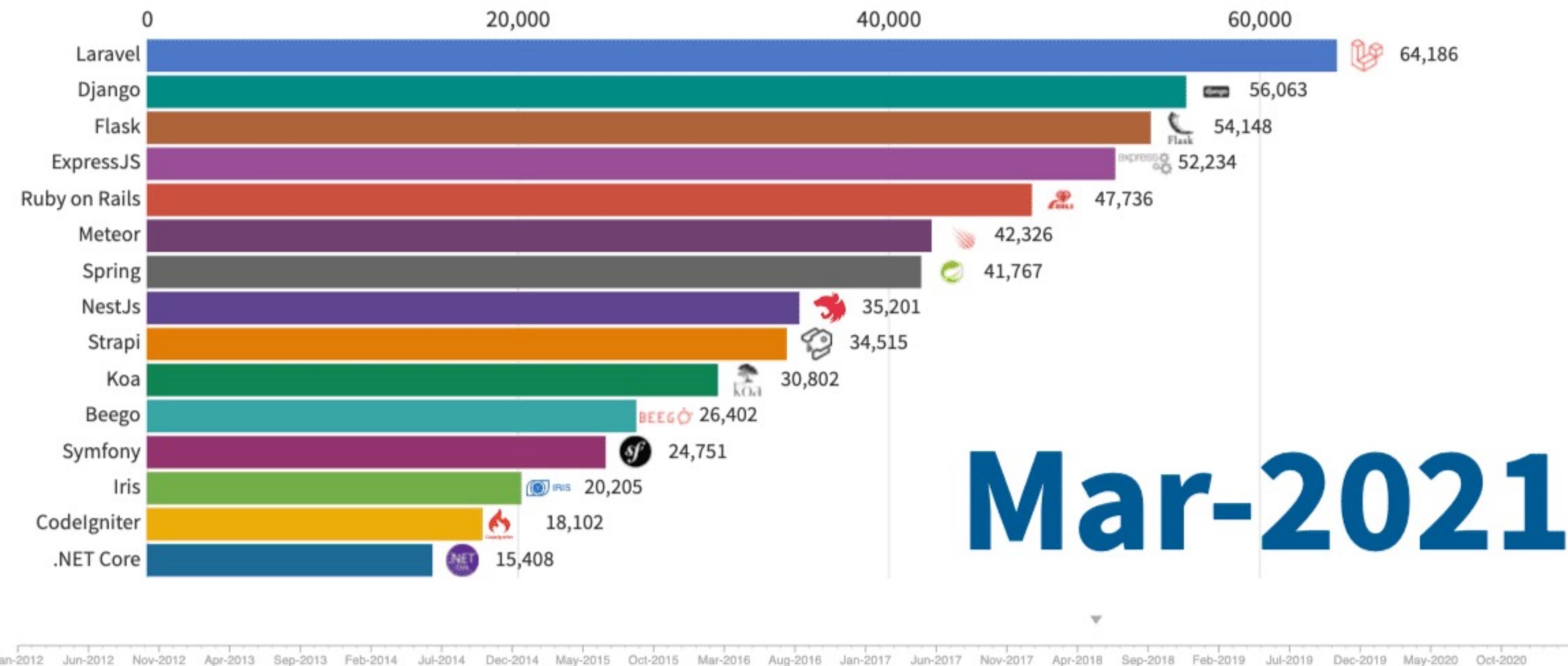
1. 성능 (Python의 한계)
2. Python-specific Feature에 대한 이해 필요
3. 그렇다고 낫지는 않은 러닝 커브 / 한국에서 비교적 적은 레퍼런스



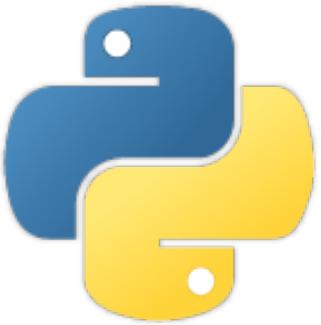
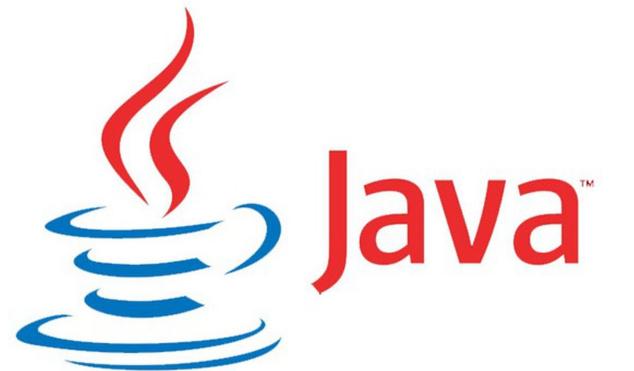


And Else

Most Popular Backend Frameworks



Know OOP

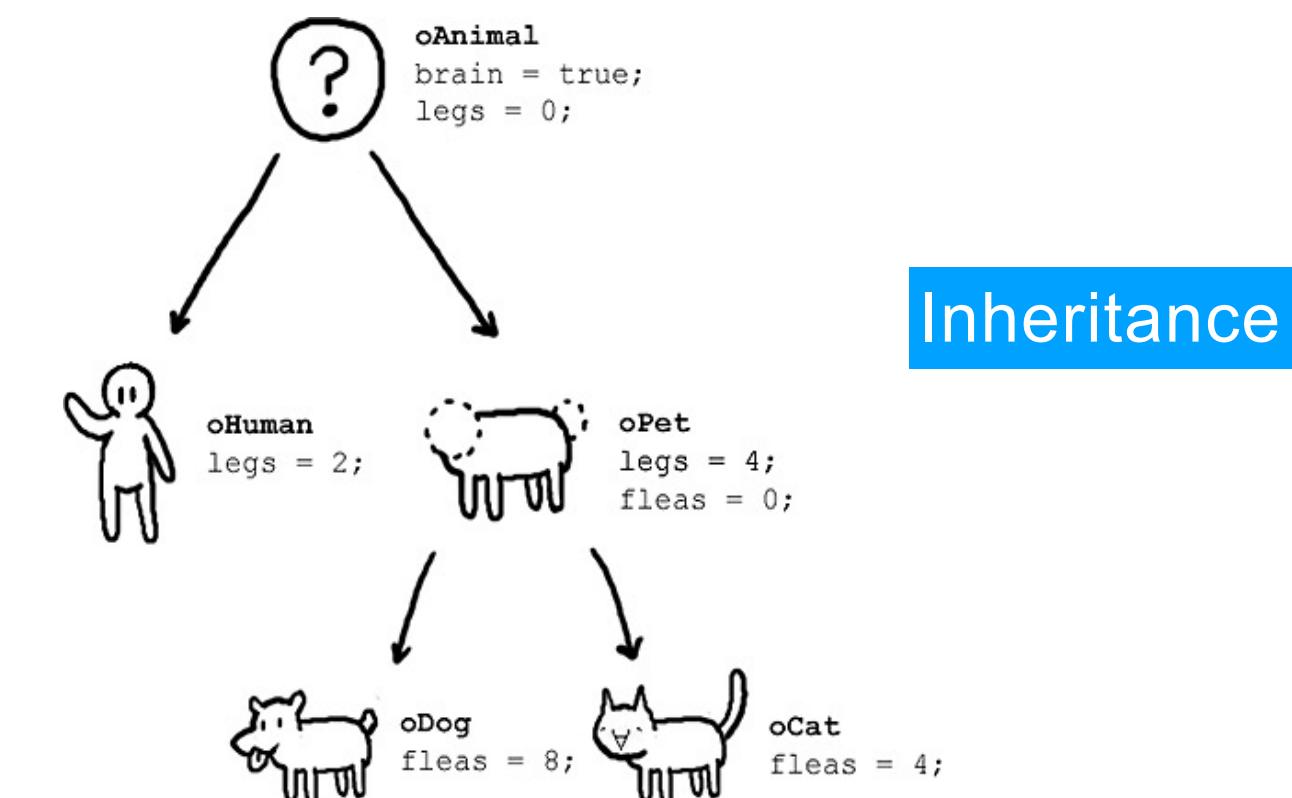
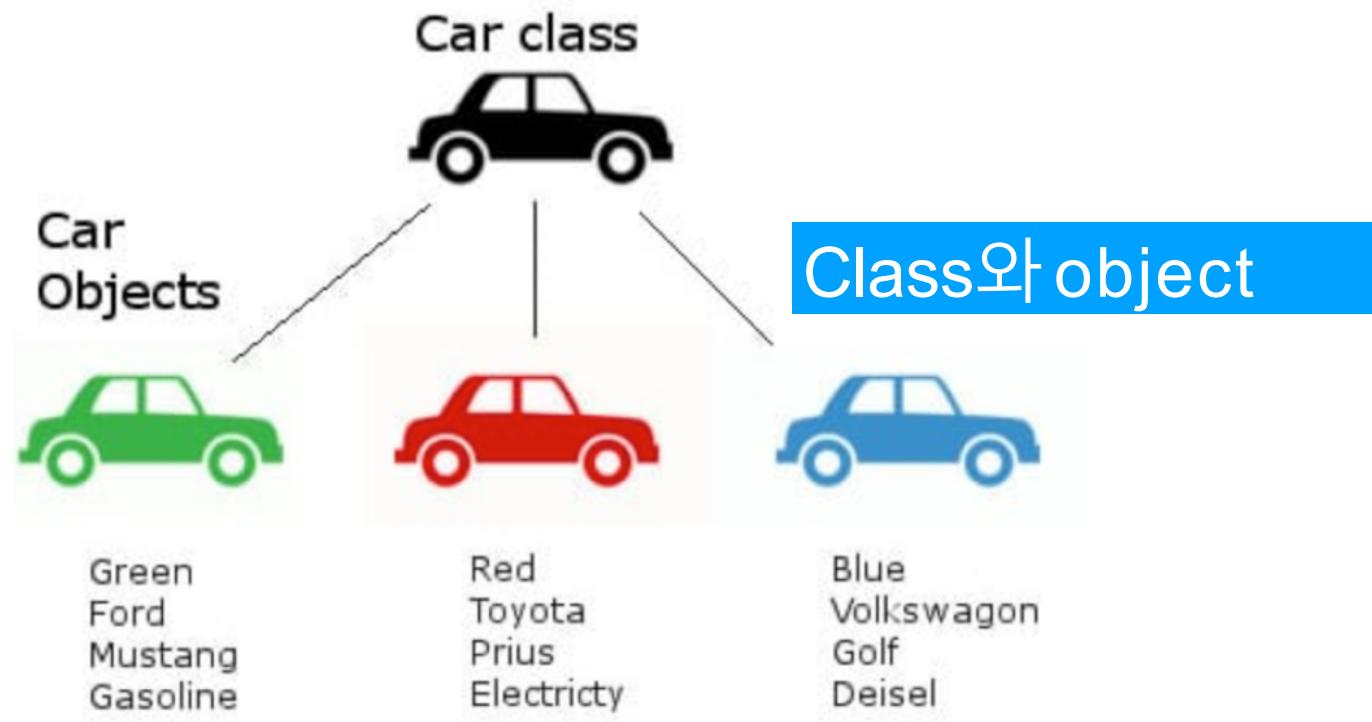


Object-Oriented Programming; 여러분이 3번에서 경험하신 그것

Abstraction / Encapsulation / Inheritance / Polymorphism

Spring, Django 는 물론, 프로그래밍 전반에 걸쳐 아주 흔하고 중요한 개념

두 세미나 전반에 걸쳐 계속해서 등장할 예정



오늘 수업 내용 실습 해보기 with **django**

천리길도 한 걸음부터!

- Python 설치하기
- Python 가상환경 세팅하기
- MySQL 설치하기
- MySQL Workbench(or DataGrip)
- Postman
- VSCode, PyCharm 등 IDE 세팅

How to Study Development?

당부의 말씀

- 세미나의 내용이나 과제에만 의존하지 말고 뭐라도 공부하고 프로그래밍하기
[**\[Nomad Coder\] 자꾸만 에러가 나오는데 왜그런걸까요?**](#)
- 에러를 안 내는 게 잘 하는 것이 아닌, 에러의 원인을 빠르게 파악해 고치는 것이 잘하는 것
- 질문하기 이전에 구글링, 구글링하기 이전에 생각하기, 로그 읽기
- (특히 Windows 분들) 자신의 환경은 자신이 가장 잘 알 수 밖에 없다

과제 안내

- Read [Assignment_0.md](#)