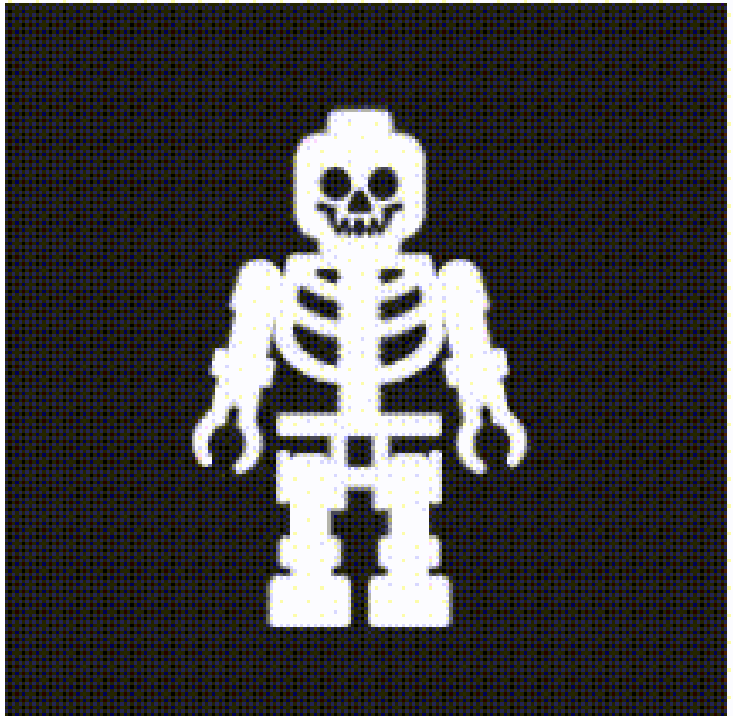
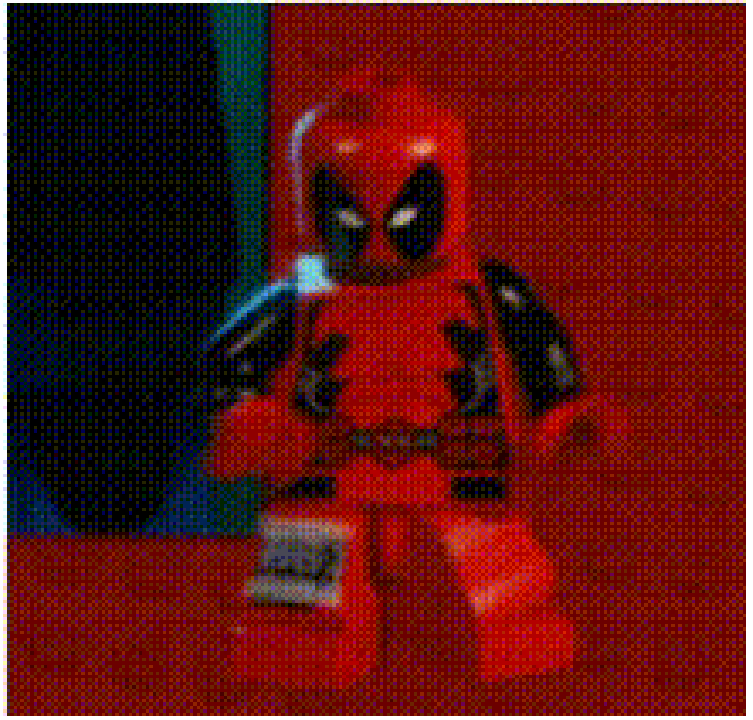
WEB

HTML
structure



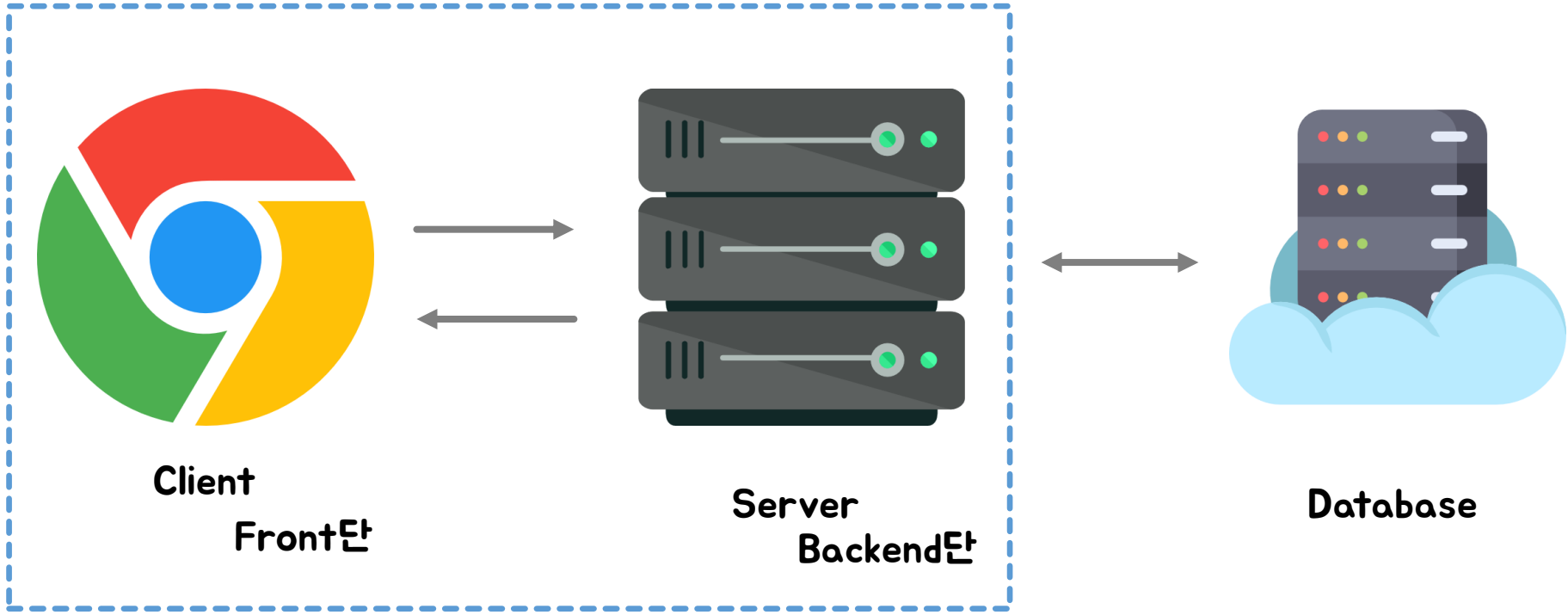
CSS
presentation/appearance

JavaScript
dynamism/action



WEB Programming

서버단 개발



Setting

Mac OS 사용자는 설치하지 마세요

Flask

```
$ pip install flask
```


```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "<p>Hello, Flask</p>"

if __name__ == "__main__":
    app.run()
```

Python이 처음이라면 참고

- https://github.com/EduProgramming/Python
- https://wikidocs.net/book/1

form

action

양식이 제출될 때 수행할 작업

method

양식을 제출할 때 HTTP method를 지정

(default) GET방식

URL변수: GET

HTTP post 트랜잭션: POST

참고 링크

https://www.w3schools.com/html/html_forms_attributes.asp

GET/POST

HTTP Method	GET 방식	POST 방식
데이터 담기는 곳	HTTP 패킷 Header	HTTP 패킷 Body
리소스 전달 방식	Query String	HTTP Body
HTTP 응답 코드	200	201
URL 데이터 노출 여부	O	X
캐싱 가능 여부	O	X
브라우저 기록 여부	O	X
북마크 추가	O	X
데이터 길이 제한	O	X
멱등성(idempotent)	O	X

*멱등성: 연산을 여러 번 하더라도 결과가 달라지지 않는 성질

GET방식

```
http://127.0.0.1:5000/user?user_id=123
```

POST방식

```
http://127.0.0.1:5000/user
```

form

accept-charset

양식 제출에 사용될 문자 인코딩 방식 설정

autocomplete

자동완성 기능 “ON”/“OFF” 설정

novalidate

양식 제출에 값들 검증하지 않음

ex) input:type=email을 사용했는데 submit눌렀을 때 email형식이 맞지 않아도 작동하게 됨

form
enctype

양식 데이터를 서버에 제출할 때 양식 데이터를 인코딩하는 방법 지정

method=POST방식일 때만 지정 사용 가능

`application/x-www-form-urlencoded`

(default) 기본값으로 모든 문자들을 서버로 보내기 전 인코딩됨 명시

`multipart/form-data`

모든 문자를 인코딩하지 않음 명시
form 태그에서 파일이나 이미지를 전송할 때 주로 사용

`text/plain`

공백 문자(space)는 “+”기호로 변환하지만 나머지 문자는 모두 인코딩되지 않음 명시

RESTful
REST

Representational State Transfer

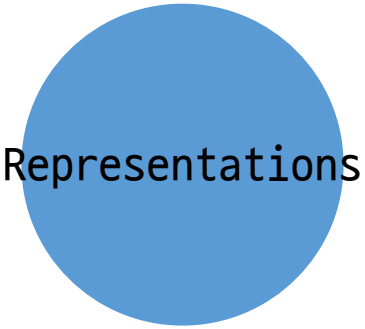
자원을 이름으로 구분하여 해당 자원의 상태를 주고 받는 모든 것을 의미

HTTP URI

자원

HTTP Method

자원에 대한 행위



HTTP Message Pay Load

자원에 대한 행위의 내용

Create: 데이터 생성(POST)

Read: 데이터 조회(GET)

Update: 데이터 수정(PUT)

Delete: 데이터 삭제(DELETE)

example

RESTful
REST

특징

1. 서버-클라이언트 구조
2. 무상태(Stateless)
3. 캐시 처리 가능(Cacheable)
4. 계층화(Layered System)
5. 인터페이스 일관성

Stateless란?

서버가 클라이언트의 세션 상태 및 세션 정보를 저장하지 않는 네트워크 프로토콜
- 요청에 대한 응답만 처리하는 방식

장점

- REST API 사용을 위한 별도 인프라 구축 필요 없음
- HTTP 프로토콜의 표준을 최대한 활용하여 여러 추가적 장점을 함께 사용 가능
- HTTP 표준 프로토콜에 따르는 모든 플랫폼에서 사용 가능
- Hypermedia API의 기본을 충실히 지키면서 범용성 보장
- REST API 메시지가 의도하는 바를 쉽게 파악 가능
- 여러 가지 서비스 디자인에서 생길 수 있는 문제 최소화
- 서버와 클라이언트의 역할을 명확하게 분리

단점

- 표준이 존재하지 않아 정의 필요
- HTTP Method형태 제한적
- 브라우저를 통해 테스트할 일이 많은 서비스라면 쉽게 고칠 수 있는 URL보다 Header정보의 값을 처리해야 하므로 전문성이 요구됨

RESTful
REST API

REST의 원리를 따르는 API 의미

설계 예시

1. URI는 동사보다 명사를, 대문자보다 소문자 사용
2. 마지막 슬래스(/)를 포함하지 않는다
3. 언더바(_) 대신 하이폰(-)를 사용
4. 파일확장자는 URI에 포함하지 않는다

   /movie/test.png x

   /movie/test O
5. 행위를 포함하지 않는다

   /put-movie/1 x

   /movie/1 O

RESTful

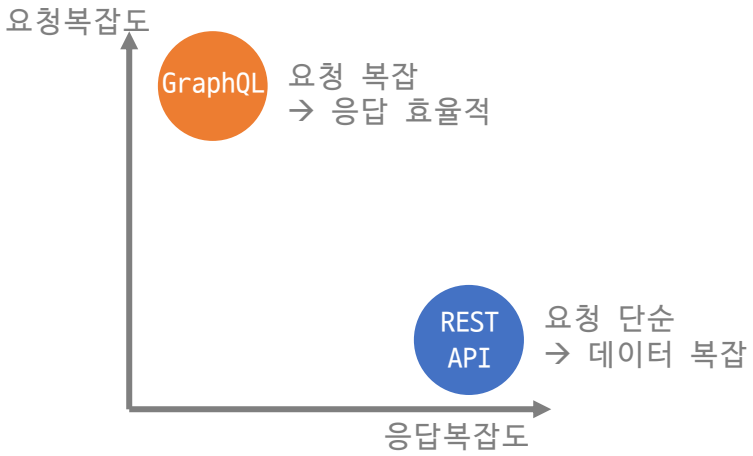
REST의 원리를 따르는 시스템

REST를 사용했다고 모두가 RESTful한 것은 아님

REST API의 설계 규칙을 올바르게 지킨 시스템을 RESTful하다 할 수 있다.

그런 REST API로 괜찮은가?

GraphQL이란?



Why PostgreSQL?

- **무료 데이터 베이스**
- OS 버전에 관계없이 설치 가능
- 엔지니어 없이 유지보수 간편
- 대용량 자료 취급할 수 있어야함

스타트업 기업들이 주로 사용하고 있어서 선정

장점

- 라이선스에 대한 비용문제가 없다
- 오픈소스의 안전성
- 지속적 발전중인 데이터베이스
- 독창적인 자료형 및 문법

  반대로 보면 단점으로 진입장벽이 높을 수 있음

단점

- MySQL과 같은 자리를 잡은 DB에 비해 인기도가 떨어짐 자리를 잡은 DB에 비해 커뮤니티 형성이 잘되어 있지 않음
- 속도에 민감한 경우 적합하지 않음

Table설계

```sql
CREATE TABLE IF NOT EXISTS Users (
    id              SERIAL              PRIMARY KEY,
    email           VARCHAR(60)         NOT NULL UNIQUE,
    password        VARCHAR(300)        NOT NULL
);

CREATE TABLE IF NOT EXISTS Likes (
    user_id         INT,
    CONSTRAINT fk_user_id FOREIGN KEY(user_id) REFERENCES Users(id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT pk_user_id PRIMARY KEY(user_id)
);
```

PostgreSQL CRUD

[C]: Create

```
INSERT INTO Users (email, password)
VALUES('test@test.com', '1234');
```

[R]: Read

```
SELECT *
FROM Users;
```

	id [PK] integer	email character varying (60)	password character varying (300)
1	1	test@test.com	1234

[U]: Update

```
UPDATE Users
SET email = 'test@naver.com', password = '4321'
WHERE id = 1;
```

```
SELECT *
FROM Users;
```

	id [PK] integer	email character varying (60)	password character varying (300)
1	1	test@naver.com	4321

[D]: Delete

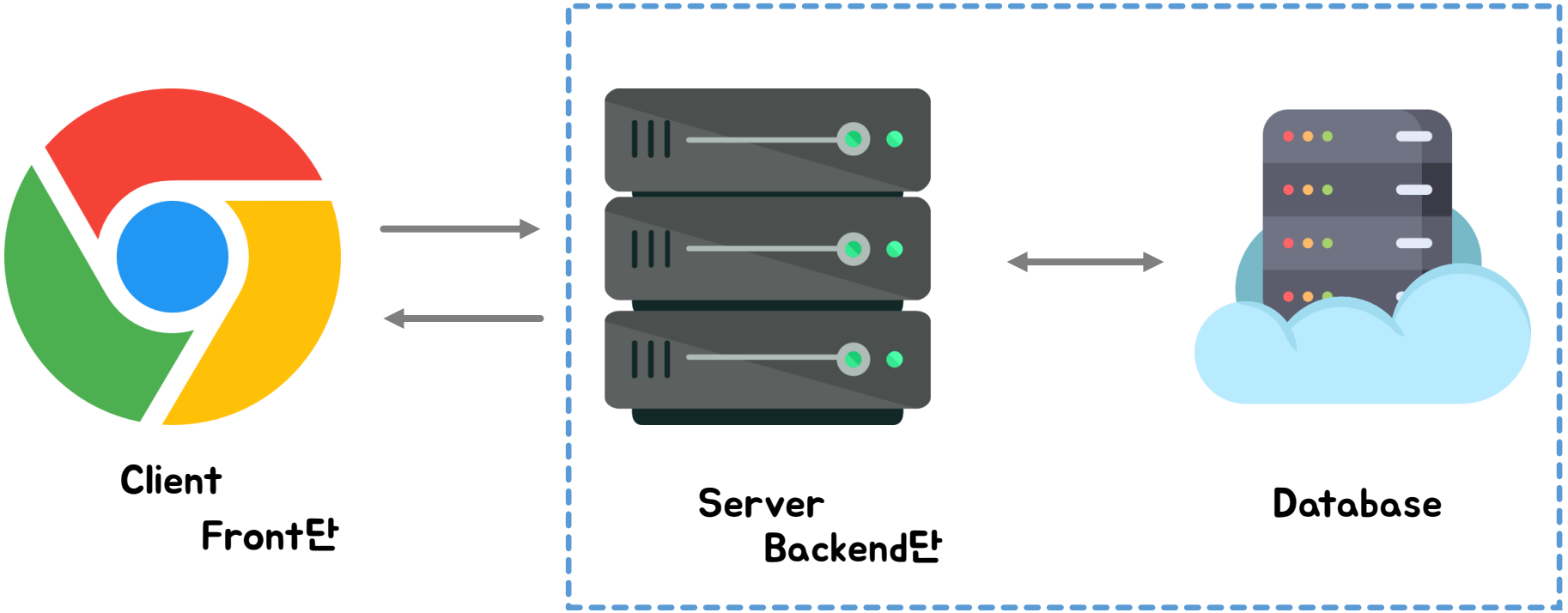
```
DELETE FROM Users
WHERE id = 1;
```

```
SELECT *
FROM Users;
```

id [PK] integer	email character varying (60)	password character varying (300)

프로그래머스:코딩테스트 연습
https://school.programmers.co.kr/learn/challenges

WEB Programming

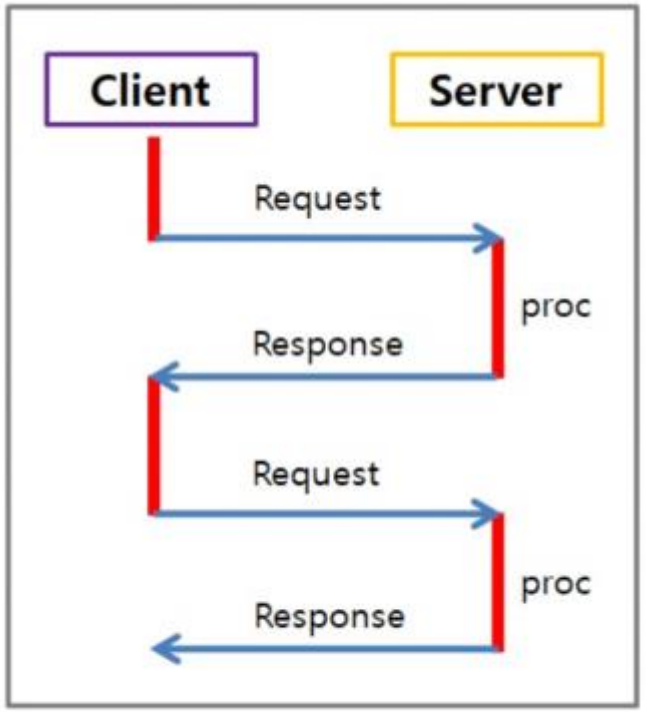


Backend – DB연동

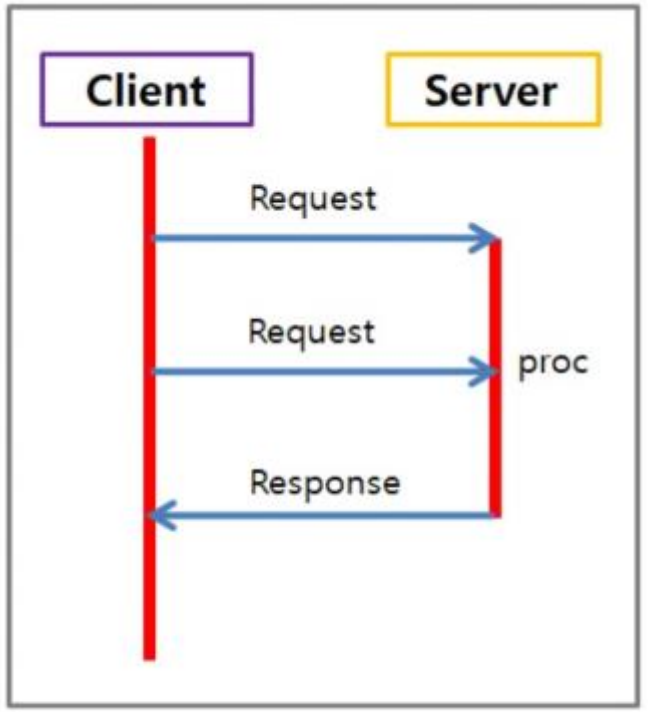
```
$ pip install psycopg2
```

Table Relationship

비동기 통신



<동기식>



<비동기식>

비동기 통신

XMLHttpRequest
(XHR)

Fetch API

Promise

async

위의 내용들을 알고 있는 것도 중요합니다.

비동기 통신

axios