

- Web server : 인터넷을 통해 client의 request가 전달되었을 때 이 요청을 처리하는 hardware & software  
→ 정적 resource 서비스

일반적인

• CGI (Common Gateway Interface) : webserver에서 application을 수행시키기 위한 규약, 규칙

(C, perl)

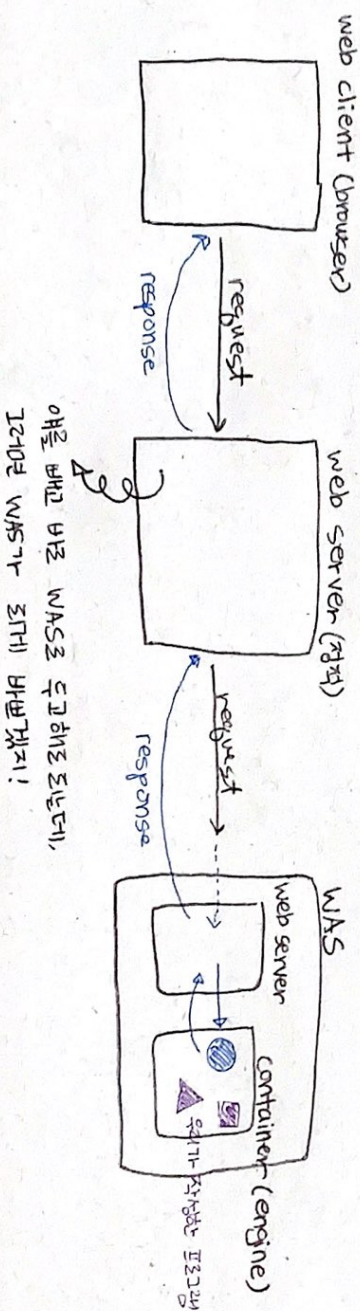
클라이언트 request → 웹서버(Apache, Nginx) → 프로그램을 실행

① 웹서버가 너무 많은 일을 한다.

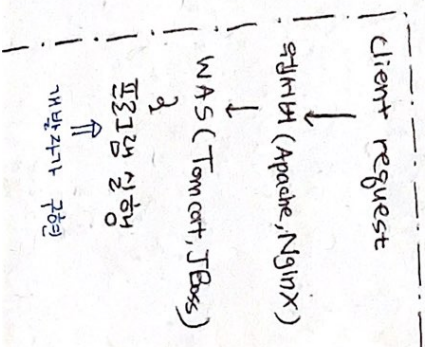
↓  
많은 양의 request를 수행하기 힘들다.

↓  
WAS 등장!

• WAS (Web Application Server) : 일반적인 웹서버와 web application을 동작시킬 수 있는 container를 분리!



이해를 위해 다른 WAS로 비교해보는데, 그러면 WAS가 크게 바뀌었지!



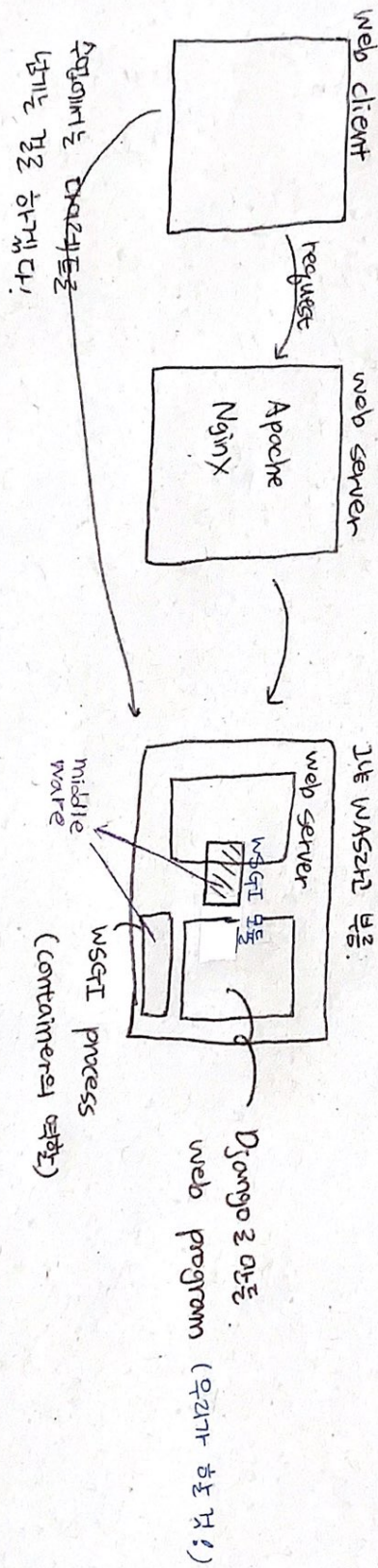


# WSGI (Web Server Gateway Interface)

: 파이썬에 종속된 개념 (즉, 파이썬에만 있는 개념!)

: python 의 python script (web application) 가 web server와 통신하기 위한 규약  
클라이언트 request → gateway (Apache, Nginx) → WSGI Server (middleware) → WSGI 규약을 준수하는 web application을 실행

by Django (플라가 호는 것!)  
(Flask, Django)



유지 (파이썬)



Django : python을 만든 자의 무조건 사옹한 수 있는 web application Framework!  
(open source)

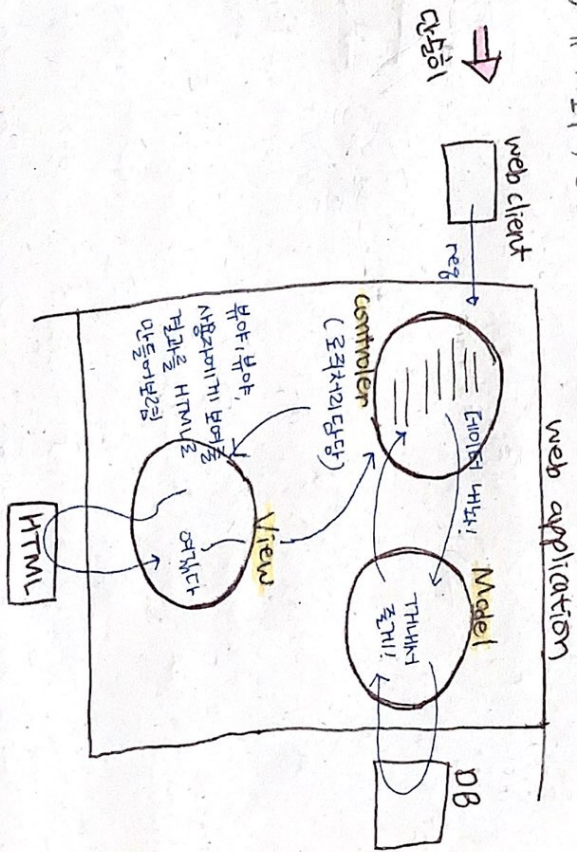
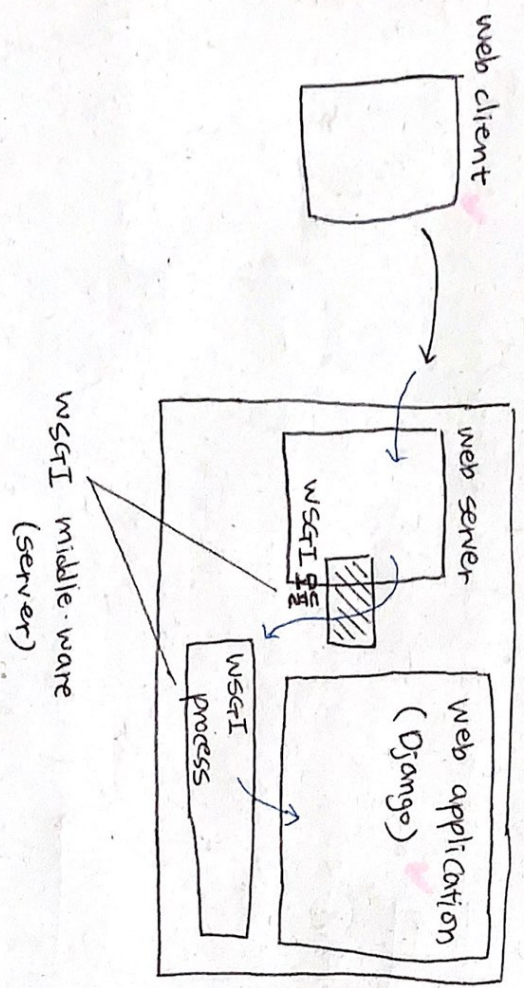
Django 특징

① **MVT** 패턴을 이용 : Design pattern (singleton pattern, observer pattern 등등...)  
중 가장 유명한 것이 **MVT!**

MVC 시작을 변화시킨 거.  
M : Model (model)  
V : View (logic)  
T : Template (UI)

MVC pattern : 프로그램 전체 구조에 대한 패턴 (모든 프로그램이 적용가능)  
(Model View Controller) Model : 데이터 데이터  
View : UI (사용자가 보거나 하는 것) 화면  
Controller : Business logic (그냥 로직이라고 생각하면 됨!) 로직 처리

구분 한 마다 다 분리해서 만들라 이야기지!  
for 협업, 유지·보수, 변경





# Django 튜링 (이어서)

## ② ORM (Object Relational Mapping)

객체 관계 매핑

~~~~~을 이용하여 알아야 할 것이 있어요...! ~~~~~

### Database

처음에는 계층형 db를 사용함. : 폴더 구조



(2차원만 프로그램적으로)

① 데이터 간의 연동이 쉽지 않음!

Network db : 128프 구조.

① 이걸로 풀어나 구현이 어려움  $\pi, \sigma$

relation 논리발표 (big 어떤 수학적자) :

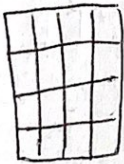


Table 구조, 여러가지 데이터에 대한 처리 등...

• 객체지향의 시대 도래!

⇒ 객체지향 DB 만들었으나 망함ㅋㅋㅋ

개념을 DB에 적용하기가 힘들고, table이 사용하기 불편함!

대신 객체지향 개념을 기회의 것이 추가함!

→ 객체-관계형 데이터베이스가 사용됨!  
(정형 데이터 처리)

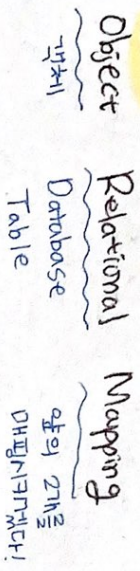
물고 DB  
↑ (비정형 데이터를 위한 DB)

→ IBM DB2라는 relational database 개발!  
→ Oracle, ... 등등도 만들거야! (관계형 데이터베이스)



CH1

# ORM



원래 database를 제어하려면 SQL 언어를 이용. (그래서 Django는 이를 간편하게!)>

python의 쿼리 언어를 이용해서 DB를 제어,

내부적으로 SQL이 자동으로 생성되어서 눈감기!

• 간편한 page가 자동 생성

• Elegant URL → 일반적으로 web program URL Design을 좀더 직관적이고 쉽게 표현이 가능.

• MVT Pattern

## Django 특징

- ① Django: 객체지향 DB 제어
- ② 간편한 page 자동 생성
- ③ URL 쉽고 직관적으로 표현이 가능
- ④ MVT pattern 사용



2.1)

- other IDE, JupyterLab, etc.
- conda install django
- python --version should 2.2.5 or higher (2.7.18)
- python --help (3.8.10) → professional 3.8.10 (2.7.18)