

Network, JSON, Serializer, 여러 Field 실습

# BE: 2번째 세션

---

NEXT X LIKELION 강미지

## 실습

OneToOne, ManyToMany, ForeignKey를 사용해봅시다

### 멋쟁이 피자처럼 메뉴판

#### 피자

페퍼로니 피자  
스테이크 피자  
하와이안 피자  
포테이토 피자

#### 음료 (피자 한판 당 택1)

콜라  
사이다  
오렌지주스

TIP.

Class는 피자, 음료 그리고 3개 타입의 재료에 대해 작성하면 좋을 것 같아요  
OnetoOne ManytoMany ForeignKey를 모두 사용해야하며, 중복이 있을 수 있습니다.

#### 재료

\*양파/버섯은 둘 중 택 1

#### 페퍼로니 피자

: 페퍼로니, 양파/버섯, 파슬리, 치즈, 도우, 토마토소스

#### 스테이크 피자

: 스테이크, 양파/버섯, 치즈, 도우, 토마토소스

#### 하와이안 피자

: 파인애플, 양파/버섯, 치즈, 도우, 토마토소스

#### 포테이토 피자

: 감자, 양파/버섯, 파슬리, 치즈, 도우, 토마토소스

# 실습

OneToOne, ManyToMany, ForeignKey를 사용해봅시다

```
##치즈 도우 토마토소스 파슬리
class Basic(models.Model):
    name = models.CharField(max_length=20)
    def __str__(self):
        return self.name
##양파 버섯
class Common(models.Model):
    name = models.CharField(max_length=20)
    def __str__(self):
        return self.name
##페퍼로니 스테이크 파인애플 감자
class Special(models.Model):
    name = models.CharField(max_length=20)
    def __str__(self):
        return self.name
```

```
class Drinks(models.Model):
    name = models.CharField(max_length=20)
    def __str__(self):
        return self.name

class Pizza(models.Model):
    name = models.CharField(max_length=20)
    basic_ingredients = models.ManyToManyField(Basic)
    common_ingredients = models.ForeignKey(Common,
                                            on_delete=models.CASCADE)
    special_ingredients = models.OneToOneField(Special,
                                              on_delete=models.CASCADE,primary_key=True)
    drinks = models.ForeignKey(Drinks,
                               on_delete=models.CASCADE)
    def __str__(self):
        return self.name
```

# 이번주에 배울 것은...

이미 반이나 끝났어요

## 0. 요구사항에 맞춰 모델 작성해보기(실습)

2-1. OneToOne, ManyToMany, ForeignKey

## 1. 네트워크 통신 맛보기

1-1. 클라이언트와 서버 통신

1-2. XML, JSON, YAML + serialization/deserialization

# 네트워크 통신이란

네트워크

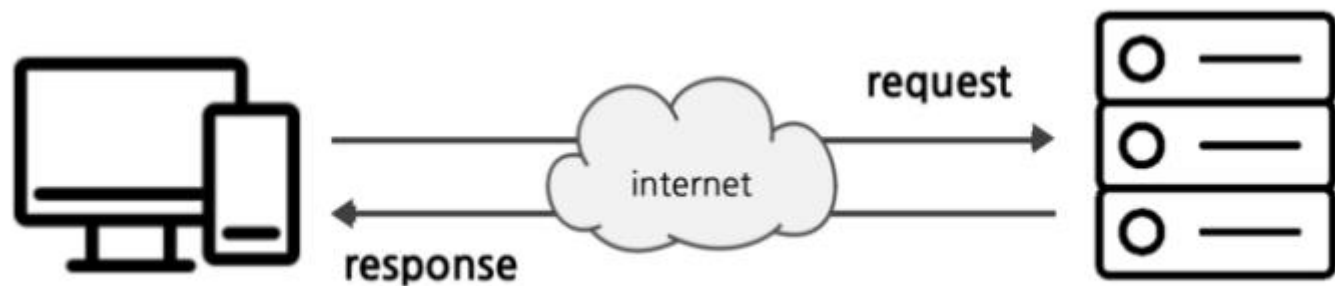
네트워크 : 컴퓨터간 데이터를 주고받을 수 있는 연결된 상태  
규모에 따라 LAN, WAN, 인터넷 등으로 불림



# 네트워크 통신이란

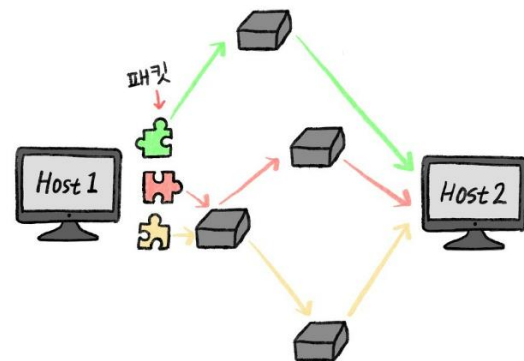
클라이언트, 서버

네트워크가 컴퓨터간 연결이라면

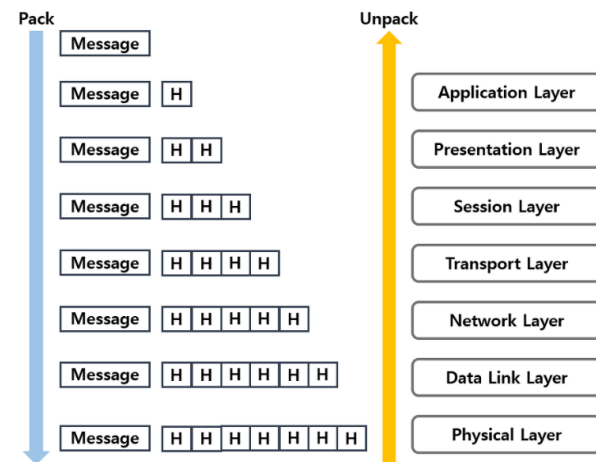


요청하는 컴퓨터  
(어플리케이션)

응답하는 컴퓨터  
(어플리케이션)

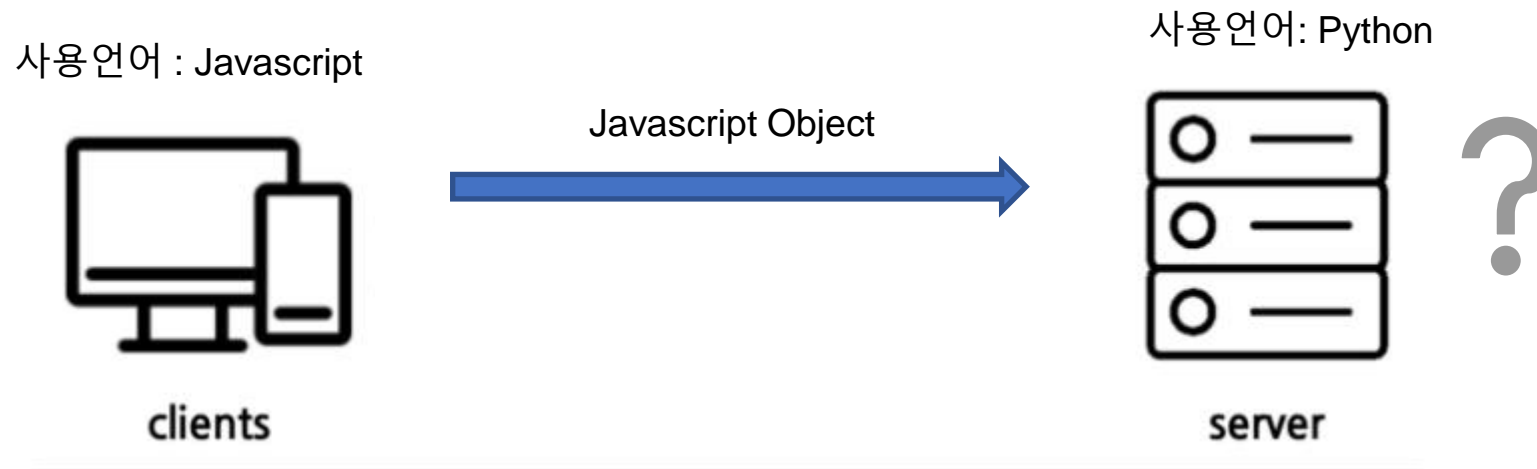


내 컴퓨터가 어떻게 다른 컴퓨터를 찾아가나요?  
→ OSI 7 Layer , TCP/IP계층



# 네트워크 통신이란

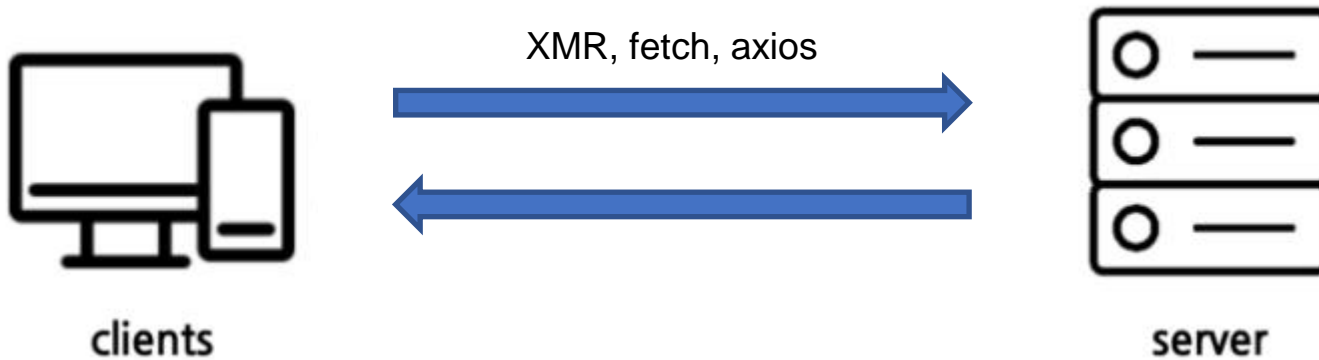
클라이언트, 서버간 통신



# 네트워크 통신이란

클라이언트, 서버간 통신

소통 규약: HTTP



소통 형식: XML, JSON, YAML



# JSON이란?

Javascript Object Notation

*JSON : Javascript Object Notation*

**1993년 나온 ECMAScript 버전3에 쓰여진 object에서 영감을 받아 만들어진 데이터 형식**

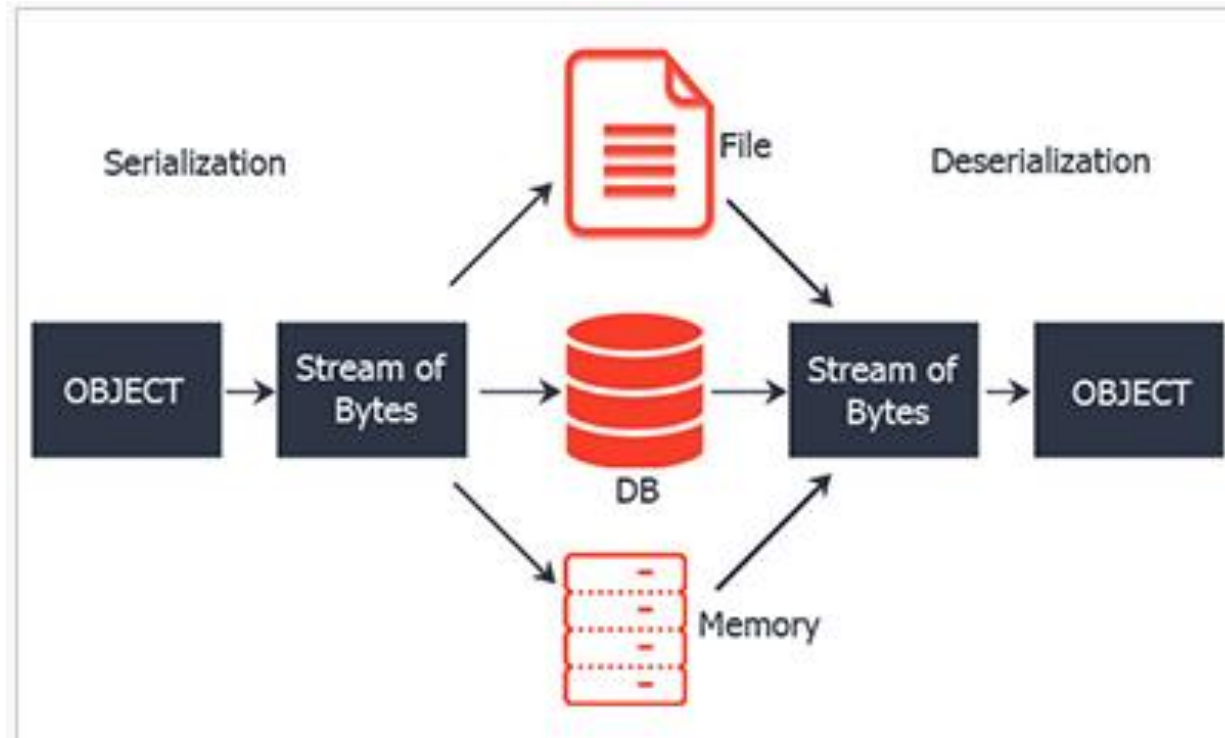
***Object {key:value}***

- 네트워크간 연결에서 데이터를 주고 받을 때 serialization과 transport를 위해 사용됨
- 프로그래밍 언어/플랫폼에 상관없이 사용 가능

**=> 어떤 언어의 object도 JSON으로 serialize 될 수 있고  
JSON(string type)도 어떠한 언어의 object로 deserialize될 수 있다.**

# JSON이란?

Javascript Object Notation



Object를 JSON형식으로 Serialization해서 클라이언트가 보내고 서버는 JSON을 deserialization(혹은 그대로)해서 사용한다

# 과제 1 : 서버 요구사항에 맞게 게시글과 댓글 CRUD + 좋아요 기능 구현하기

지난 과제의 `models.py`를 이용해주세요

1. 멋사 백엔드 스터디에서는 게시판 서비스를 지원하는 서버를 구축하려고 합니다!
2. 게시글(Post) 및 댓글(Comment)에 대한 CRUD(CREATE, READ, UPDATE, DELETE) 기능이 필요합니다.
3. 게시글은 카테고리로 구분되어야 하며, '카테고리 이름' 혹은 '카테고리 코드'로 특정 카테고리의 게시글만을 조회할 수 있어야 합니다.
4. 사용자(User) 정보를 관리해야 하고, 로그인한 사용자만이 게시글 및 댓글을 작성할 수 있습니다.
5. 사용자 정보로는 사용자의 이름, 아이디, 비밀번호, 그리고 사용자의 전공을 관리합니다.
6. 사용자의 전공 정보로는 본전공 외에 이중/복수/융합/심화 전공 정보를 포함해야 합니다.
7. 로그인한 사용자는 게시글에 "좋아요"를 표시하거나, 해제할 수 있어야 하며, 게시글에는 "좋아요를 누른 사용자 수"가 표시되어야 합니다.

## 과제 2

- ✓ DRF Docs Tutorial 4 연습하기
- ✓ 유튜브 강의 chapter 8 연습하기 (15분 분량)
- ✓ **2주차 발표자 분들은 오늘 하신 내용을 토대로 발표 준비 해주세요**

# 과제 관련 링크

파이팅!

[DRF 공식문서 Tutorial]

<https://www.django-rest-framework.org/tutorial/1-serialization/>

[유튜브 강의]

<https://www.youtube.com/watch?v=B38aDwUpcFc&t=3821s>