



Django Model Relationship

❖ 학습해야 할 내용

- ✓ Django Model Relation

1. M:N True or False

각 문항을 읽고 맞으면 T, 틀리면 F를 작성하고 틀렸다면 그 이유도 함께 작성하시오.

- 1) Django에서 1:N 관계는 ForeignKeyField를 사용하고 M:N 관계는 ManyToManyField를 사용한다.
- 2) ManyToManyField를 설정하고 만들어지는 테이블 이름은 "앱이름_클래스이름_지정한 필드이름"의 형태로 만들어진다.
- 3) ManyToManyField의 첫번째 인자는 참조할 모델, 두번째 인자는 related_name이 작성 되는데 두 가지 모두 필수적으로 들어가야 한다.

2. Like in templates

아래 빈 칸 (a)와 (b)에 들어갈 코드를 각각 작성하시오.

```
class Article(models.Model):
    ...
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    like_users = models.ManyToManyField(settings.AUTH_USER_MODEL, related_name=
'like_articles')
```

```
<!-- articles/index.html -->

{% for article in articles %}
    ...
    <p>{{ article.title }}</p>
    <a href="{% url 'articles:like' article.pk %}">
        {% if __ (a) __ in __ (b) __ %}
            <i class="fas fa-heart fa-lg" style="color:crimson;"></i>
        {% else %}
            <i class="fas fa-heart fa-lg" style="color:black;"></i>
        {% endif %}
    </a>
    <span>{{ __ (b) __ |length }}명이 이 글을 좋아합니다.</span>
{% endfor %}
```



Django Model Relationship

3. Follow in views

모델 정보가 다음과 같을 때 빈칸 (a)와 (b)에 들어갈 코드를 각각 작성하시오.

```
from django.db import models
from django.conf import settings
from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    followers = models.ManyToManyField(
        'self',
        symmetrical=False,
        related_name='followings'
    )
```

```
app_name = 'accounts'
urlpatterns = [
    ...,
    path('<int:user_pk>/follow/', views.follow, name='follow'),
]
```

```
from django.contrib.auth import get_user_model

User = get_user_model()

@require_POST
def follow(request, ____(a)____):
    person = get_object_or_404(User, pk=__(a)____)
    user = request.user

    if user != person:
        if person.__(b)____.__(c)____(pk=user.pk).exists():
            person.__(b)____.__(d)____(user)
        else:
            person.__(b)____.__(e)____(user)
    return redirect('accounts:profile', person.username)
```



Django Model Relationship

4. User AttributeError

아래와 같은 에러 메시지가 발생하는 이유와 이를 해결하기 위한 방법과 코드를 작성하십시오.

```
AttributeError at /accounts/signup/  
Manager isn't available; 'auth.User' has been swapped for 'accounts.User'  
  
Request Method: POST  
Request URL: http://127.0.0.1:8000/accounts/signup/  
Django Version: 2.1.15  
Exception Type: AttributeError  
Exception Value: Manager isn't available; 'auth.User' has been swapped for 'accounts.User'
```

5. related_name

아래의 경우 related_name을 필수적으로 설정해야 한다. 그 이유를 설명하십시오.

```
class Article(models.Model):  
    ...  
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)  
    like_users = models.ManyToManyField(settings.AUTH_USER_MODEL, related_name=  
    'like_articles')
```



Django Model Relationship

6. follow templates

person 변수에는 view함수에서 넘어온 유저 정보가 담겨 있고,
모델 정보가 아래와 같을 때
빈칸 a, b, c, d, e에 들어갈 알맞은 코드를 각각 작성하시오.

```
app_name = 'accounts'
urlpatterns = [
    ...,
    path('<int:user_pk>/follow/', views.follow, name='follow'),
]
```

```
from django.db import models
from django.contrib.auth.models import AbstractUser

# Create your models here.
class User(AbstractUser):
    followings = models.ManyToManyField('self', symmetrical=False, related_name='followers')
```

```
<h3>작성자 : {{ person.username }}</h3>

<div>팔로잉 : {{ __a__|length }}</div>
<div>팔로워 : {{ __b__|length }}</div>

<div>
    {% if __c__ != __d__ %}
        {% if __c__ in __b__ %}
            <a href="{% url 'accounts:follow' __e__ %}">Unfollow</a>
        {% else %}
            <a href="{% url 'accounts:follow' __e__ %}">Follow</a>
        {% endif %}
    {% endif %}
</div>
```