

04/27 (월) Django



❖ 학습해야 할 내용

- ✓ Django 1:N Model Relation

- 다음 중 맞으면 T, 틀리면 F를 작성하고 틀렸다면 이유를 함께 작성하시오.
 - ForeignKey는 부모 테이블의 데이터를 참조하기 위한 키이다.
 - 1:N 관계에서 1은 N의 데이터를 직접 참조 할 수 있다.
 - on_delete 속성은 ForeignKey 필드의 필수 인자이다.
 - 1:N 관계에서 ForeignKey는 반드시 부모 테이블의 PrimaryKey여야 한다.
- 다음과 같이 이름이 articles인 app의 models.py에 작성된 코드를 바탕으로 테이블이 만들어 졌을 때, 데이터베이스에 저장되는 ForeignKey 컬럼의 이름과 테이블의 이름이 무엇인지 작성하시오.

```
from django.db import models

class Question(models.Model):
    title = models.CharField(max_length=50)

class Comment(models.Model):
    answer = models.ForeignKey(Question, on_delete=models.CASCADE)
    content = models.CharField(max_length=100)
```

- 위 2번 문제 모델 관계를 바탕으로 어느 template 페이지가 다음과 같이 작성되어 있을 때, 질문(Question)에 작성된 모든 댓글(Comment)을 출력하고자 한다. 해당 template에서 Question 객체를 사용할 수 있다면 빈칸 (a)에 들어갈 알맞은 코드를 작성하시오.

```
{% for comment in __(a)__ %}
    <p>{{ comment.content }}</p>
{% endfor %}
```

04/27 (월) Django



4. 다음과 같이 게시글을 삭제하는 delete 함수와 로그인을 위한 login 함수가 작성되어 있다. 만약 비로그인 사용자가 삭제를 시도한다면 django는 해당 사용자를 url에 next 파라미터가 붙은 login 페이지로 redirect 한다. (ex /accounts/login/?next=/articles/1/delete/) redirect된 로그인 페이지에서 로그인에 성공했을 때 발생하는 HTTP response status code를 작성하고 발생한 원인과 해결을 위해 코드를 수정하십시오. (게시글 삭제는 HTTP POST method로만 가능하며 인증되지 않은 사용자는 메인페이지로 redirect 되어야 한다.)

```
from django.views.decorators.http import require_POST
from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect, get_object_or_404
from .models import Article
```

```
@login_required
@require_POST
def delete(request, article_pk):
    article = get_object_or_404(Article, pk=article_pk)
    article.delete()
    return redirect('articles:index')
```

```
from django.contrib.auth import login as auth_login
from django.contrib.auth.forms import AuthenticationForm

def login(request):
    if request.user.is_authenticated:
        return redirect('articles:index')
    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            auth_login(request, form.get_user())
            return redirect(request.GET.get('next') or 'articles:index')
    else:
        form = AuthenticationForm()
    context = {
        'form': form,
    }
    return render(request, 'accounts/login.html', context)
```