1:N, template inheritance

권중훈

오늘 만들 것

detail.html

책제목

HTML 4대 비극

느낀점

HTML/CSS/Python/Django
홈으로 수정하기 삭제하기

댓글

- Python이냐 JavaScript냐, 그것이 문제로다삭제하기

댓글 쓰기

오늘 할 것

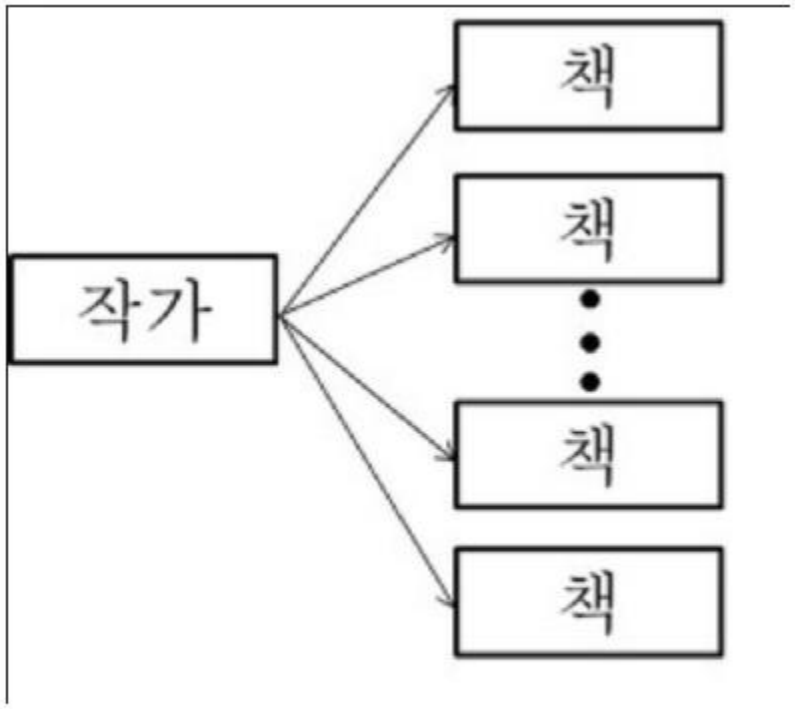
- 1:N 데이터베이스 구조
- 실습: 댓글 구현
- 템플릿 상속

1:N 데이터베이스

1:N

- 하나의 데이터가 여러 개의 데이터와 관계를 맺는 구조

- 글 - 댓글 / 유저 – 주문 등 실제로 가장 많이 사용되는 구조이다.



장고 모델 1:N

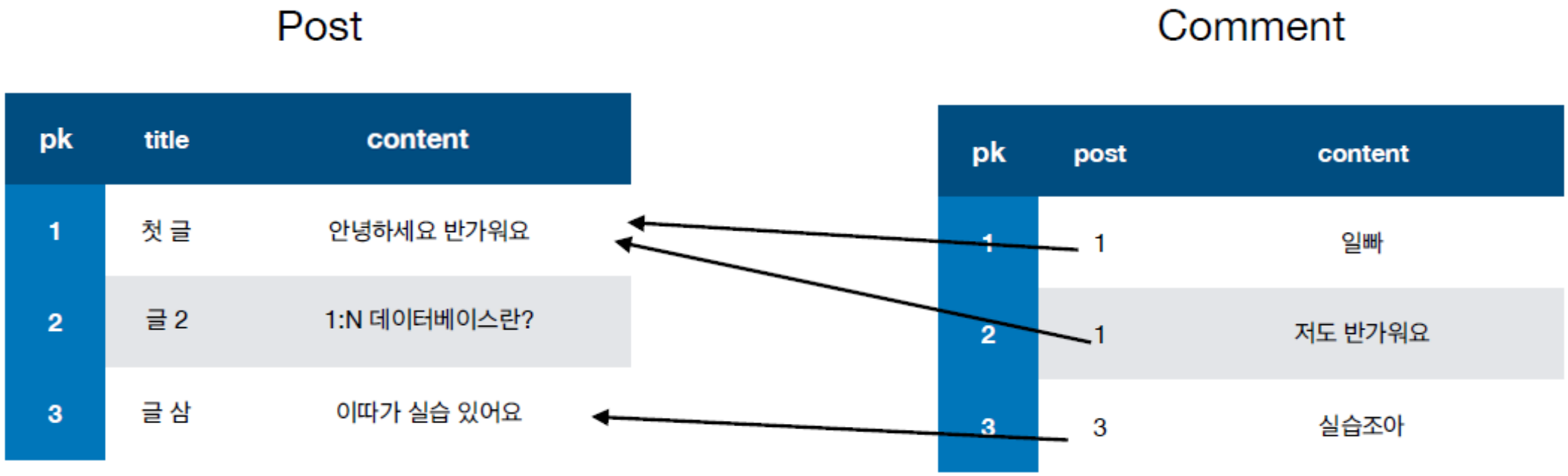
Post

pk	title	article
1	첫 글	안녕하세요 반가워요
2	글 2	1:N 데이터베이스란?
3	글 삼	이따가 실습 있어요

Comment

pk	post	content
1	1	일빠
2	1	저도 반가워요
3	3	실습조아

장고 모델 1:N


Post
pk
title
content
1
첫 글
안녕하세요 반가워요
2
글 2
1:N 데이터베이스란?
3
글 삼
이따가 실습 있어요
Comment
pk
post
content
1
1
일빠
2
1
저도 반가워요
3
3
실습조아

실습: 댓글 구현

- 지난 세션 때 만든 책 블로그에 댓글을 달아봅시다.

실습 준비

- mkdir session10
- cd session10
- git clone https://github.com/Gjunghoon/crud-1-n.git
- cd crud-1-n
- pipenv shell
- pipenv install (Pipfile.lock 정보를 바탕으로 가상환경에 모듈 설치)
- cd crudproject
- python manage.py makemigrations
- python manage.py migrate
- python manage.py createsuperuser
- python manage.py runserver

장고의 MTV 패턴

- 모델을 만들고 -> 템플릿을 만들고 -> 뷰를 만들고 -> url을 연결

models.py

```python
from django.db import models

# Create your models here.
class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()

    def __str__(self):
        return self.title

class Comment(models.Model):
    post = models.ForeignKey(Post, on_delete=models.CASCADE, related_name='comments')
    content = models.TextField()
```

모델이 변경되었습니다.

- (Ctrl + C) 눌러서 서버를 나와주시고
- python manage.py makemigrations
- python manage.py migrate
- python manage.py runserver 해주세요.

admin.py

- 만든 모델을 어드민에서 확인하고 싶으면 어드민에 등록해줘야합니다.

```python
from django.contrib import admin
from .models import Post, Comment

# Register your models here.
admin.site.register(Post)
admin.site.register(Comment)
```

detail.html

```
<div>
  <div>
    <div>{{ post.title }}</div>
    <div>{{ post.content }}</div>
  </div>
  <a href="{% url 'home' %}">홈으로</a>
  <a href="{% url 'edit' post.pk %}">수정하기</a>
  <a href="{% url 'delete' post.pk %}">삭제하기</a>

  <form method="POST">
    {% csrf_token %}
    <input type="text" name="content" placeholder="댓글을 입력해주세요">
    <button type="submit">댓글 쓰기</button>
  </form>
</div>
```

views.py

```python
from django.shortcuts import render, redirect
from .models import Post, Comment
```

- … 중략 …

```python
def detail(request, post_pk):
    post = Post.objects.get(pk=post_pk)

    if request.method == "POST":
        Comment.objects.create(
            post = post,
            content = request.POST['content']
        )
        return redirect('detail', post_pk)

    return render(request, 'detail.html', {'post': post})
```

detail.html

```
<div>
  <div>
    <div>{{ post.title }}</div>
    <div>{{ post.content }}</div>
  </div>
  <a href="{% url 'home' %}">홈으로</a>
  <a href="{% url 'edit' post.pk %}">수정하기</a>
  <a href="{% url 'delete' post.pk %}">삭제하기</a>

  {% for comment in post.comments.all %}
  <li>{{ comment.content }}</li>
  {% endfor %}

  <form method="POST">
    {% csrf_token %}
    <input type="text" name="content" placeholder="댓글을 입력해주세요">
    <button type="submit">댓글 쓰기</button>
  </form>
</div>
```

댓글이 잘 만들어졌습니다

제목입니다
내용입니다
홈으로 수정하기 삭제하기

- 댓글입니다

댓글을 입력해주세요 댓글 쓰기

댓글 삭제 기능을 만들어봅시다

제목입니다
내용입니다
홈으로 수정하기 삭제하기

- 댓글입니다

댓글삭제

댓글을 입력해주세요 댓글 쓰기

detail.html

```
<div>
  <div>
    <div>{{ post.title }}</div>
    <div>{{ post.content }}</div>
  </div>
  <a href="{% url 'home' %}">홈으로</a>
  <a href="{% url 'edit' post.pk %}">수정하기</a>
  <a href="{% url 'delete' post.pk %}">삭제하기</a>

  {% for comment in post.comments.all %}
  <li>{{ comment.content }}</li>
  <a href="{% url 'delete_comment' post.pk comment.pk %}">댓글삭제</a>
  {% endfor %}

  <form method="POST">
    {% csrf_token %}
    <input type="text" name="content" placeholder="댓글을 입력해주세요">
    <button type="submit">댓글 쓰기</button>
  </form>
</div>
```

views.py

- …중략…

```
def delete_comment(request, post_pk, comment_pk):
    comment = Comment.objects.get(pk=comment_pk)
    comment.delete()
    return redirect('detail', post_pk)
```

urls.py

```python
from django.contrib import admin
from django.urls import path
from app import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('new/', views.new, name="new"),
    path('detail/<int:post_pk>', views.detail, name="detail"),
    path('edit/<int:post_pk>', views.edit, name="edit"),
    path('delete/<int:post_pk>', views.delete, name="delete"),
    path('delete_comment/<int:post_pk>/<int:comment_pk>', views.delete_comment, name="delete_comment")
]
```

템플릿 상속

템플릿 상속

- https://docs.djangoproject.com/en/3.0/ref/templates/language/

Template inheritance

The most powerful – and thus the most complex – part of Django's template engine is template inheritance. Template inheritance allows you to build a base "skeleton" template that contains all the common elements of your site and defines **blocks** that child templates can override.

Let's look at template inheritance by starting with an example:

사용법

- {% extends ‘base.html’ %}
- {% block 태그이름 %}
- {% endblock 태그이름%}

원리

- base.html에서 틀을 짜고
- 나머지 템플릿에서 base.html을 extends(상속)해줍니다.
- 바꾸고 싶은 block만 자식 템플릿에서 내용을 바꿔줍니다.

base.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  <title>{% block title %}제목{% endblock %}</title>
</head>

<body>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>

  {% block content %}
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  {% endblock content %}

  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
</body>

</html>
```

home.html

```
{% extends 'base.html' %}
{% block title %}독후감 블로그{% endblock title %}

{% block content %}
{% for post in posts %}
<div>
  <ul>
    <li>
      <a href="{% url 'detail' post.pk %}">{{ post.title }}</a>
    </li>
  </ul>
</div>
{% endfor %}
<a href="{% url 'new' %}">글 쓰러가기</a>
{% endblock %}
```

localhost:8000

이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.
이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.

- 제목입니다

글 쓰러가기
이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.
이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.

base.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  <title>{% block title %}제목{% endblock %}</title>
</head>

<body>
  {% block content %}
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  {% endblock content %}
</body>

</html>
```

home.html

```
{% extends 'base.html' %}
{% block title %}독후감 블로그{% endblock title %}

{% block content %}
{% for post in posts %}
<div>
  <ul>
    <li>
      <a href="{% url 'detail' post.pk %}">{{ post.title }}</a>
    </li>
  </ul>
</div>
{% endfor %}
<a href="{% url 'new' %}">글 쓰러가기</a>
{% endblock content %}
```

base.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  <title>{% block title %}제목{% endblock %}</title>
</head>

<body>
  {% block content %}
  <!-- 다른 템플릿에서 base.html을 extends 해주면 -->
  <!-- 필요한 부분만 block 태그로 수정해줄 수 있습니다 -->
  {% endblock content %}
</body>

</html>
```

home.html

```
{% extends 'base.html' %}
{% block title %}독후감 블로그{% endblock title %}

{% block content %}
{% for post in posts %}
<div>
  <ul>
    <li>
      <a href="{% url 'detail' post.pk %}">{{ post.title }}</a>
    </li>
  </ul>
</div>
{% endfor %}
<a href="{% url 'new' %}">글 쓰러가기</a>
{% endblock content %}
```

```
{% extends 'base.html' %}
{% block title %}독후감 블로그{% endblock title %}

{% block content %}
{% for post in posts %}
<div>
  <ul>
    <li>
      <a href="{% url 'detail' post.pk %}">{{ post.title }}</a>
    </li>
  </ul>
</div>
{% endfor %}
<a href="{% url 'new' %}">글 쓰러가기</a>
{% endblock content %}
```

=

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}독후감 블로그{% endblock %}</title>
</head>

<body>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>

  {% block content %}
  {% for post in posts %}
  <div>
    <ul>
      <li>
        <a href="{% url 'detail' post.pk %}">{{ post.title }}</a>
      </li>
    </ul>
  </div>
  {% endfor %}
  <a href="{% url 'new' %}">글 쓰러가기</a>
  {% endblock content %}

  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
  <div>이 내용은 base.html을 extends해준 모든 템플릿에 들어갑니다.</div>
</body>

</html>
```

활용법?

- 헤더/푸터/네브바/sidebar 등등

- {% block 태그명 %}으로 지정만 해주면 활용법은 무궁무진

과제(5/25(월) 세션전까지)

- To-do-list 과제에 댓글 기능을 추가해주세요

- 1. detail.html에 각 할 일에 대한 모든 댓글을 볼 수 있어야 합니다.
- 2. detail.html에 댓글 생성과 삭제 기능을 넣어주세요.
- 3. base.html을 만들어서, 모든 템플릿에 'ㅇㅇㅇ의 to-do-list 블로그'(예시) 라는 h1 태그를 최상단에 보여주세요(모든 템플릿에 header를 다는 것이 핵심이므로, 멘트는 자유입니다)