



# 템플릿 언어, 상속

11기 운영진 김민지

멋쟁이사자처럼 대학 11기

# views.py 수정

---

```
views.py x
blog > views.py > index
1  from django.shortcuts import render
2  from .models import Post
3
4  # Create your views here.
5  def index(request):
6      posts = Post.objects.all()
7      return render(request, "index.html", {'posts' : posts})
8
```

from .models import Post

Post모델을 사용하기 위해 import

posts = Post.objects.all()

변수에 게시글 담기, Post 모델의 모든 내용을 posts라는 변수에 저장

return render(request, 'index.html', {'posts' : posts})

html로 변수 내보내기

해당 주소로 접속할 때, index.html의 내용을 보여주고, posts의 내용들을 전송

# 템플릿 언어

---

파이썬 변수 및 문법을 html 안에서 쓸 수 있도록 장고에서 제공해 주는 언어

파이썬 딕셔너리 형식으로 장고 데이터를 html에 전달

하지만 html에서는 파이썬 언어로 넘어온 데이터를 알지 못하기 때문에 템플릿 언어를 사용

## 변수

변수 : {{ 변수명 }}, ex) {{ post }}

변수의 속성 : {{ 변수명.속성 }}, ex) {{ post.title }}

## 필터

{{ 변수명|내장 필터 }}

변수의 값을 특정 형식으로 변환

여러 개의 필터를 연속적으로 사용 가능, 필터 직접 제작 가능

length : 값의 길이를 반환, ex) {{ post.title|length }}

slice : 표시되는 글자수 제한, ex) {{ post.title|truncatechars:7 }}

# 템플릿 언어

---

태그

for문

```
{% for post in posts %}  
    <h1>{{ post.title }}</h1>  
    <p>{{ post.body }}</p>  
{% endfor %}
```

if문

```
{% if post.title == "멋사" %}  
    <h1>{{ post.title }}</h1>  
    <p>{{ post.body }}</p>  
{% else %}  
    <p>제목이 멋사가 아닙니다</p>  
{% endif %}
```

url

```
{% url '이름' %}, ex) {% url 'detail' %}
```

[https://django-doc-test-kor.readthedocs.io/en/old\\_master/ref/templates/builtins.html](https://django-doc-test-kor.readthedocs.io/en/old_master/ref/templates/builtins.html)

# HTML에서 게시글 불러오기

---

index.html X

blog > templates > index.html > html > body

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     {% for post in posts %}
11         <h1>{{ post.title }}</h1>
12     {% endfor %}
13 </body>
14 </html>
```

{% for post in posts %}

<h1>{{ post.title }}</h1>

{% endfor %}

# 쿼리셋

---

Django ORM에서 제공하는 데이터 타입, 데이터베이스에서 전달받은 객체 목록

모든 객체 조회

`Post.objects.all()`

하나의 객체 조회, 결과가 1건 이상일 때는 에러

`Post.objects.get(id=1)`

정렬하기 : 특정 필드를 기준으로 정렬, -가 붙으면 내림차순을 의미

`Post.objects.order_by('created_date')`

`Post.objects.order_by('-created_date')`

필터

`Post.objects.filter(title='멋사')`

exact : 일치하는 조건을 입력할 때 사용

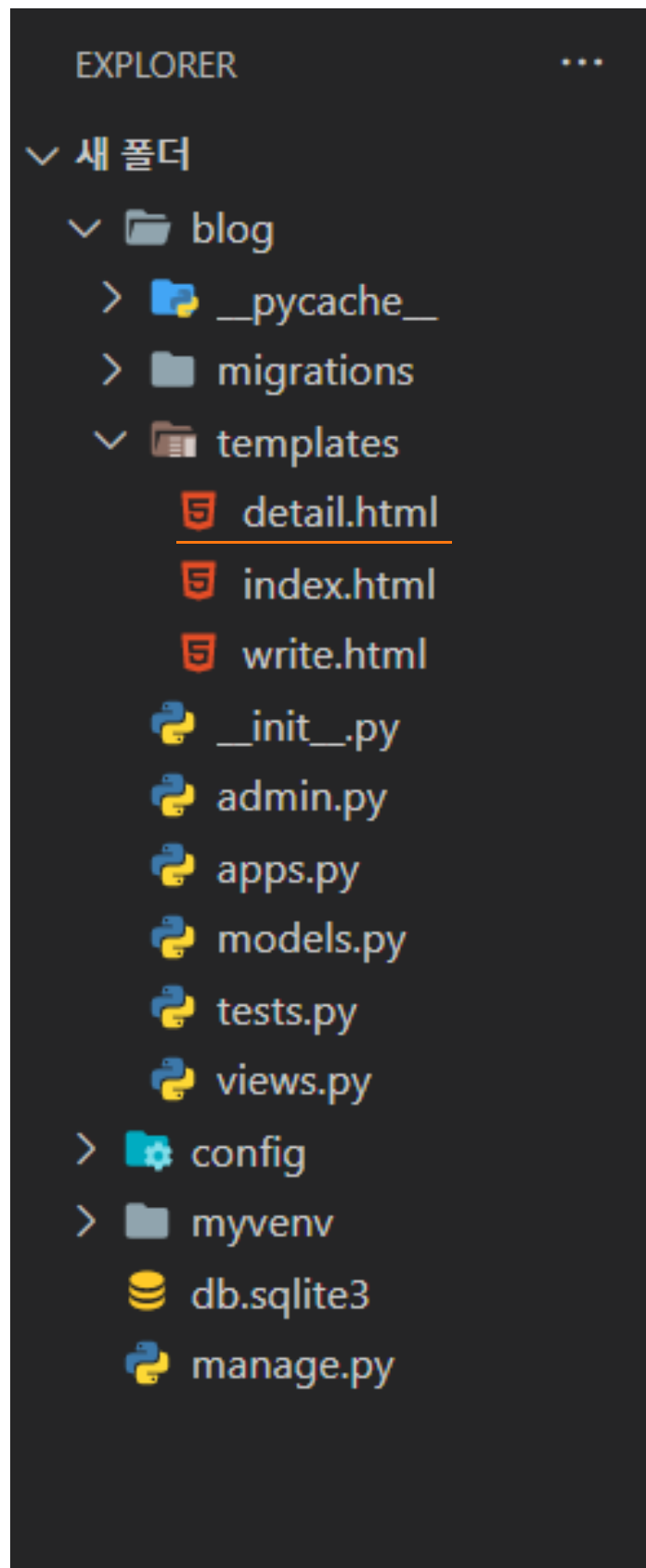
`Post.objects.filter(title__exact='멋사')`

contains : 부분 일치 조건 입력할 때 사용

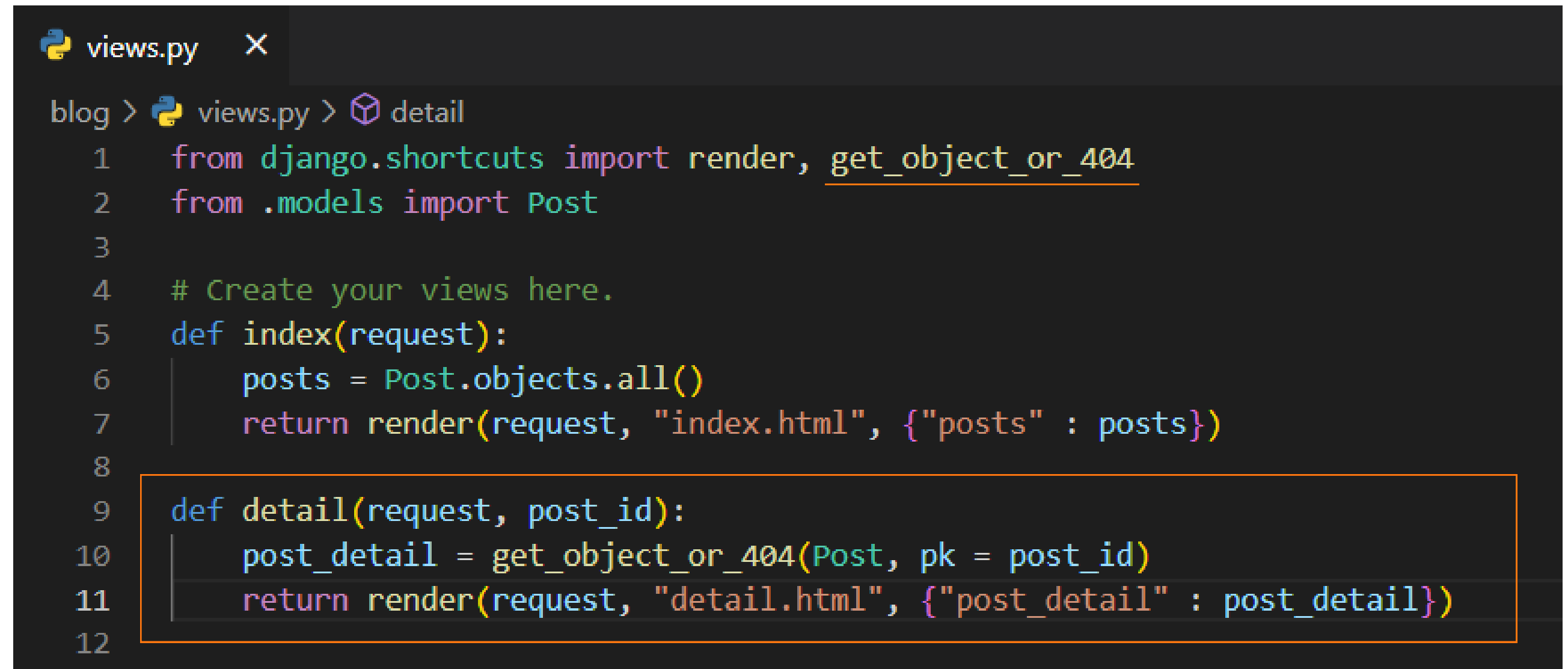
`Post.objects.filter(title__contains='멋사')`

# Read

## 1. detail.html 생성



## 2. views.py에 detail 함수 추가



# Read

---

## 3. urls.py에 경로 추가

```
17 from django.contrib import admin
18 from django.urls import path
19 from blog import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', views.index, name="index"),
24     path('detail/<int:post_id>', views.detail, name="detail"),
25 ]
26
```



# Read

---

## 4. detail.html

```
9  <body>
10  |   <h1>{{ post_detail.title }}</h1>
11  |   <p>{{ post_detail.body }}</p>
12  |   <p>{{ post_detail.pub_date }}</p>
13  </body>
```

## 5. index.html에서 a 태그로 detail 페이지 연결

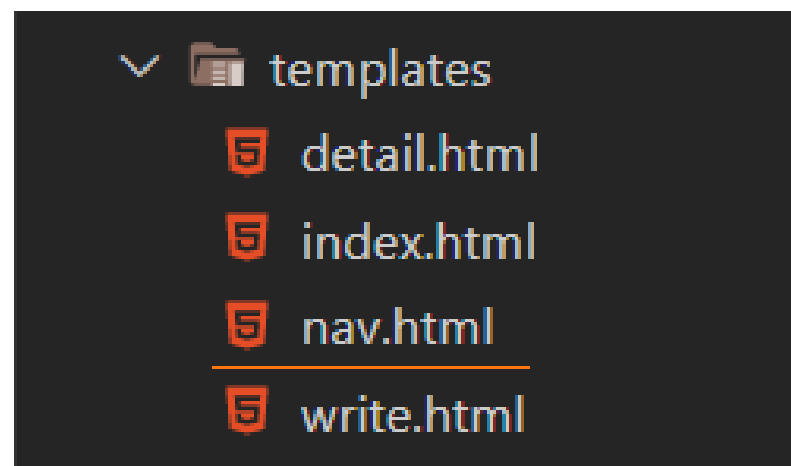
```
9  <body>
10  |   {% for post in posts %}
11  |   |   <a href="{% url 'detail' post.id %}">{{ post.title }}</a>
12  |   |   {% endfor %}
13  </body>
14  </html>
```

# 템플릿 상속

---

html 화면 상 계속해서 중복되는 부분을 별도의 html문서로 분류하여 한번만 작성하고 계속해서 재사용

## 1. nav.html 생성



## 2. settings.py에 경로 추가

```
55  ▼ TEMPLATES = [  
56  ▼      {  
57          'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58          'DIRS': ['blog/templates'],  
59          'APP_DIRS': True,  
60  ▼          'OPTIONS': {  
61  ▼              'context_processors': [  
62                  'django.template.context_processors.debug',  
63                  'django.template.context_processors.request',  
64                  'django.contrib.auth.context_processors.auth',  
65                  'django.contrib.messages.context_processors.messages',  
66              ],  
67          },  
68      },  
69  ]
```

# 템플릿 상속

---

## 3. nav.html 작성

```
nav.html X
blog > templates > nav.html > html > body
1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Document</title>
9  </head>
10 <body>
11     멧쟁이사자처럼 블로그
12     <br>
13
14     {% block content %}
15     {% endblock %}
16 </body>
17 </html>
```

# 템플릿 상속

---

## 4. 상속받은 템플릿에 작성

```
{% extends 'nav.html' %}  
{% load static %}  
{% block content%}
```

·  
·  
·

```
{% endblock %}
```

# 과제

---

템플릿 언어 사용해서 admin에서 게시물 작성하여  
게시글 목록과 상세페이지 만들어보기