

# 1장. PyWeb 프로젝트 만들기

## (index페이지 및 model 만들기)



# 목 차

1

Django로 pyweb 프로젝트 및 board앱 만들기

2

index 페이지 만들기 - 템플릿

3

Question, Answer 모델 만들기

# 장고(django) 프로젝트 생성하기

## ❖ 장고(django) 프로젝트 생성하기

1. 프로젝트 디렉터리(webproject) 아래에 pyweb 디렉터리 생성
2. 가상환경 진입하기

```
(mysite) C:\venvs\mysite\Scripts>cd c:/webproject/pyweb
```

3. 하위에 디렉터리( mysite)를 만들고 장고 프로젝트 설치

```
(mysite) C:\webproject\pyweb>django-admin startproject config .
```

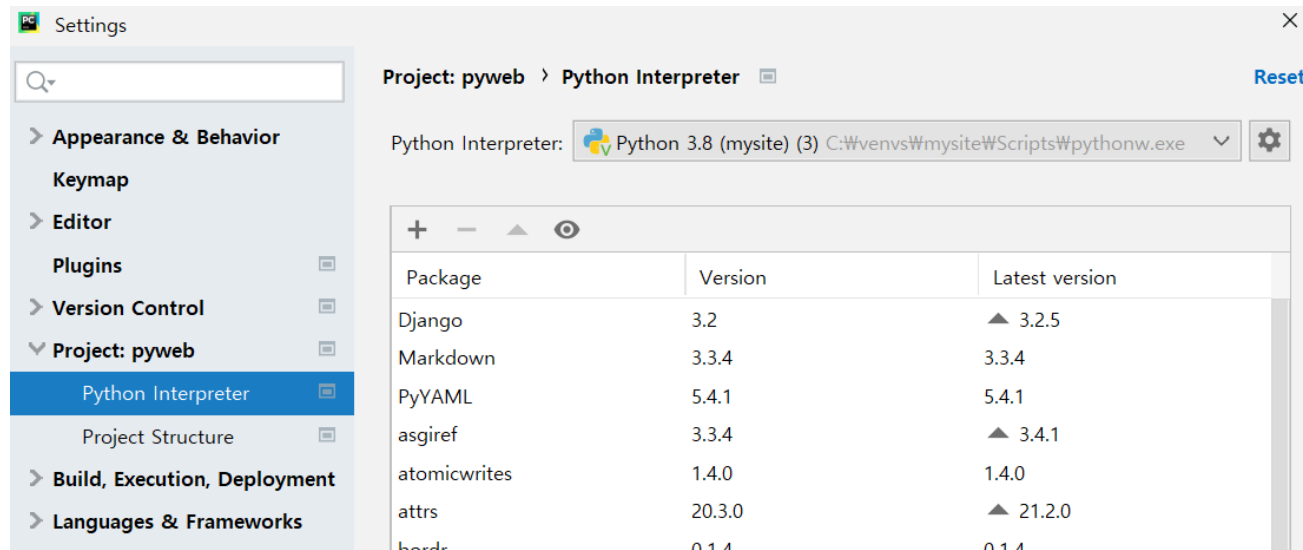
4. 개발 서버 구동하고 웹 사이트에 접속(127.0.0.1:8000)

```
(mysite) C:\projects\mysite>python manage.py runserver
```

# 파이참 인터프리터 설정

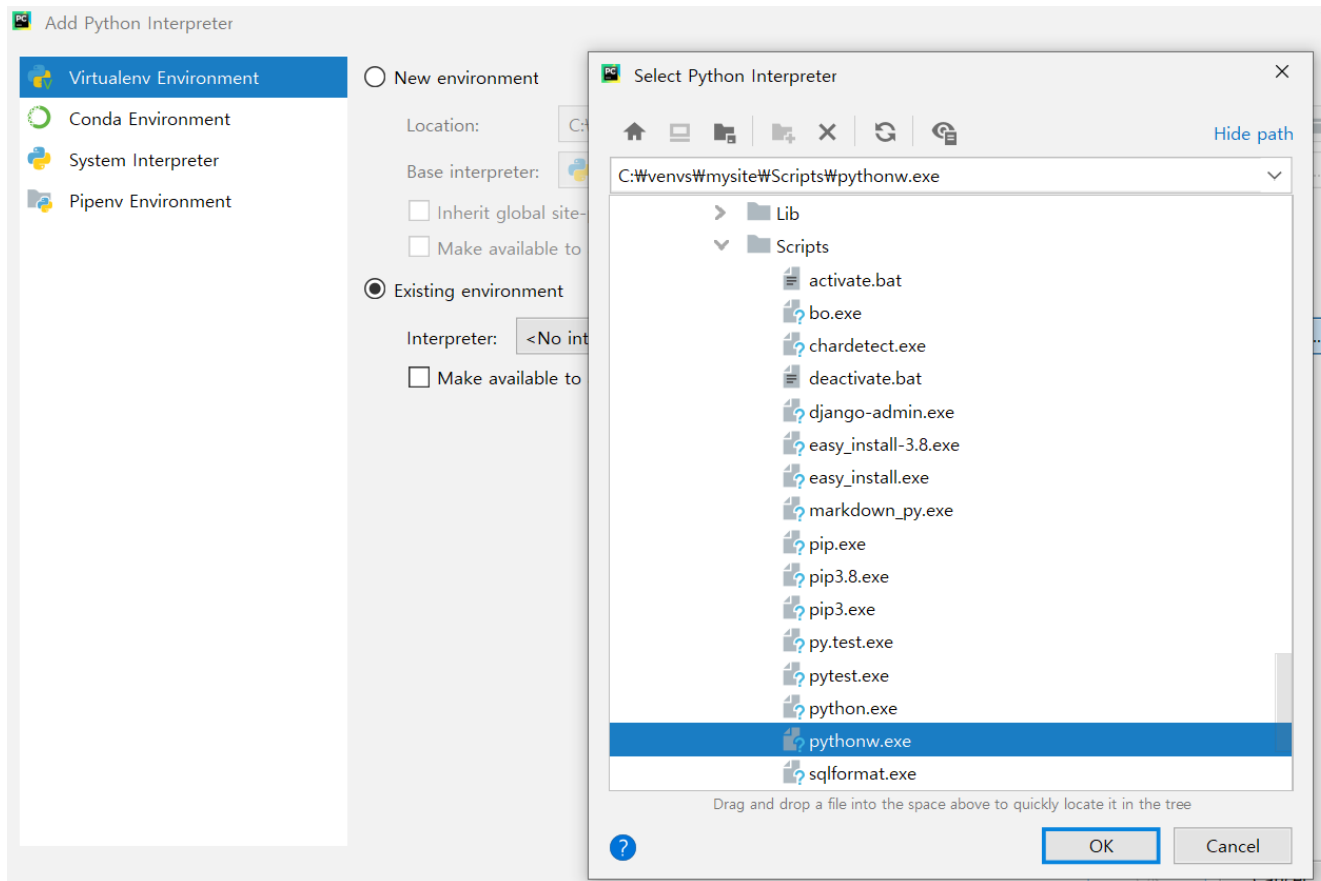
## ❖ 파이참 인터프리터 설정

1. 파이참 > Open > webproject/pyweb 선택
2. 파이썬 인터프리터 위치를 가상 환경 위치로 바꾸기  
[File-Settings] > [webproject/pyweb – Project Interpreter]  
오른쪽 위 설정 > add

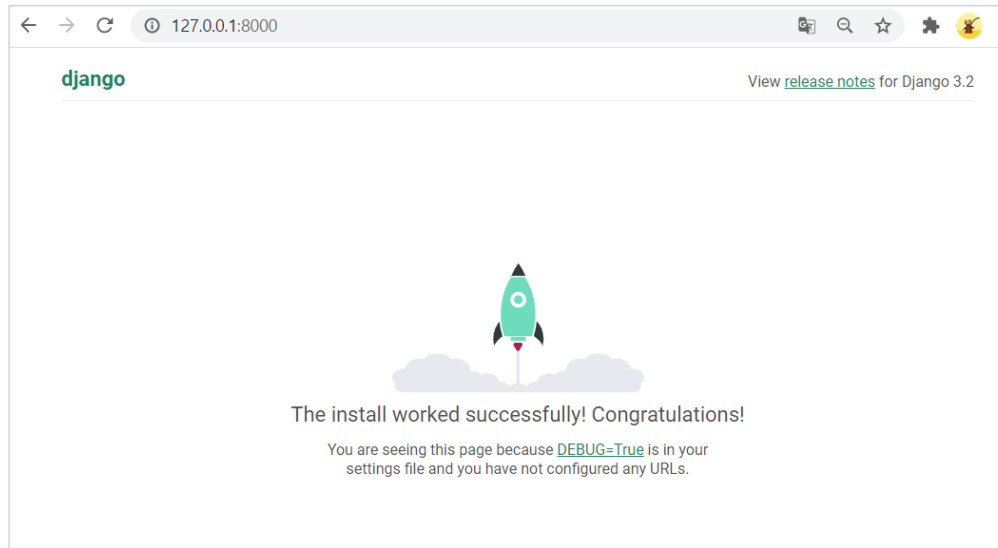


# 파이썬 인터프리터 설정

## ❖ 파이썬 인터프리터 설정



# 127.0.0.1:8000



- 언어 시간 변경 – config/settings.py

```
LANGUAGE_CODE = 'ko-kr'  
  
TIME_ZONE = 'Asia/Seoul'
```

## pyweb 관리자 슈퍼유저 생성

### 5. DB 사용을 위한 migrate 실행하기

```
(mysite) > C:\webproject\ pyweb >python manage.py migrate
```

### 6. 관리자 페이지 사용을 위한 슈퍼유저 생성하기

```
(mysite) > C:\webproject\ pyweb >python manage.py createsuperuser
```

```
c:\website\mypet>python manage.py createsuperuser
Username (leave blank to use 'suguin'): admin
Email address: sugu2100@daum.net
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

# pyweb 사이트 관리자 페이지

Django 관리

사용자 이름:

admin

비밀번호:

.....

로그인

## Django 관리

### 사이트 관리

#### POLL

##### Choices

+ 추가    ✎ 변경

##### Questions

+ 추가    ✎ 변경

#### 인증 및 권한

##### 그룹

+ 추가    ✎ 변경

##### 사용자(들)

+ 추가    ✎ 변경



# board 앱 생성하기

## ❖ board 앱 생성하기

1. django-admin의 startapp 명령 사용

```
(mysite) C:\webproject\ pyweb>python manage.py startapp board
```

2. config/setting.py 에 'board'앱 등록하기

```
INSTALLED_APPS = [  
    'board',  
    'django.contrib.admin',  
    'django.contrib.auth',  
]
```

# index 페이지 – 요청과 응답

## ❖ index 페이지에 문자열 출력하기

### 1. board/views.py 작성

```
from django.http import HttpResponse  
def index(request):  
    return HttpResponse("안녕하세요~ 방문을 환영합니다.")
```

### 2. views.index 경로 설정 - config/urls.py

```
from django.urls import path  
from board import views  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('board/', views.index),  
]
```

## index 페이지 – 요청과 응답

3. board 하위에 urls.py 생성하고, views.py 경로 설정 – board/urls.py

```
from django.urls import path
from board import views

app_name = 'board' # 네임 스페이스(소속)

urlpatterns = [
    # 127.0.0.1:8000/board
    path("", views.index, name='index'),
]
```

## index 페이지 – 요청과 응답

4. config/urls.py에서 pybo/urls.py include 함

```
from django.urls import path, include
from board import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.index),          # 127.0.0.1:8000
    path('board/', include('board.urls')),
]
```

# 템플릿으로 index 페이지 만들기

## ◆ 템플릿(template)

1. templates 디렉토리를 루트 디렉터리 바로 밑에 만든다.

```
(mysite) C:\webproject\mysite>mkdir templates
```

2. 템플릿 디렉터리 위치를 config/settings.py에 등록하기

```
TEMPLATES = [  
    .....  
    'DIRS': [BASE_DIR / 'templates'],  
    .....  
]
```

# index 페이지 만들기

## 1. index 페이지 제어 함수 정의 – board/views.py

```
from django.shortcuts import render
```

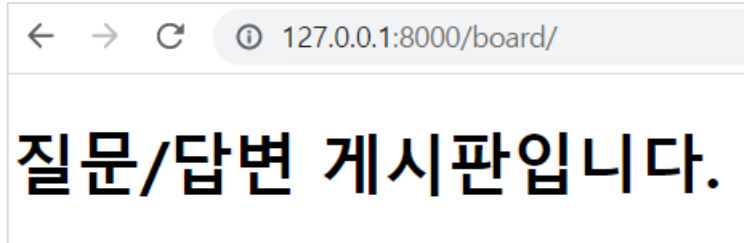
```
def index(request):
```

```
    return render(request, 'board/question_list.html')
```

```
    #return HttpResponse("안녕하세요~ 방문을 환영합니다.")
```

# 템플릿으로 질문 목록 페이지 만들기

## 2. question\_list.html 만들기 – templates/board/question\_list.html



```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>게시판 목록</title>
</head>
<body>
  <h1>질문/답변 게시판입니다.</h1>
</body>
</html>
```

TemplateDoesNotExist at /poll/  
poll/index.html

Request Method: GET  
Request URL: http://127.0.0.1:8000/poll/  
Django Version: 3.2.7  
Exception Type: TemplateDoesNotExist  
Exception Value: poll/index.html  
Exception Location: c:\venvs2\mysite\lib\site-packages\django  
Python Executable: c:\venvs2\mysite\Scripts\python.exe

에러 해결 – config/setting.py  
'DIRS': [**BASE\_DIR** / '**templates**'],

# 모델(Model) 만들기

## ◆ board 모델 – Question 생성하기

board/models.py에 Question 모델 생성

```
from django.db import models

class Question(models.Model):
    subject = models.CharField(max_length=200)
    content = models.TextField()
    create_date = models.DateTimeField()

    def __str__(self):
        return self.subject
```



# 모델(Model) 만들기

## ▪ 필드(속성) 타입

Field Type	설 명
<b>CharField</b>	제한된 문자열 타입, max length 옵션 사용
<b>TextField</b>	제한이 없는 가변 길이 대용량 문자열 타입
<b>IntegerField</b>	32bit 정수 타입 (-21억 ~ 21억)
<b>BigIntegerField</b>	64bit 정수 타입 (조, 경 이상)
<b>DateTimeField</b>	날짜와 시간을 갖는 타입
<b>BinaryField</b>	바이너리 데이터를 저장하는 필드
<b>ImageField</b>	이미지 업로드를 위한 필드
<b>FileField</b>	파일업로드를 위한 필드

# 모델(Model) 만들기

## ■ 필드(속성) 옵션

Field 옵션	설 명
<b>primary_key</b>	primary_key=True로 사용 (null=False, unique=True)
<b>null</b>	null=True이면 비어있는 데이터(허용)
<b>blank</b>	blank=True는 폼 유효성 검사에서 사용된다. null은 DB와 관련이 있고 blank는 form.is_valid()와 관련
<b>default</b>	필드의 default값 예) IntegerField(default=0)
<b>on_delete</b>	외래키에서 Foreign_key(on_delete=models.CASCADE) 연쇄 삭제 옵션
<b>related_name</b>	외래키 필드가 여러 개일 경우 관계를 지정해줌 related_name='author_question'

# 모델(Model) 만들기

- primary\_key 사용 예제

1. 자동 생성(Sequence- Auto Increment)

```
class Question(models.Model):
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    subject = models.CharField(max_length=200)
    content = models.TextField()
    create_date = models.DateTimeField()
```

2. primary\_key=True 옵션 사용

```
class Product(models.Model):
    p_id = models.CharField(max_length=10, primary_key=True) # 상품 아이디
    p_name = models.CharField(max_length=20) # 상품명
    p_price = models.IntegerField(default=0) # 가격
    p_description = models.TextField() # 상세 설명
    p_category = models.CharField(max_length=20) # 분류
```

# 모델(Model) 만들기

## ◆ board 모델 – Answer 생성

board/models.py에 Answer모델 생성

```
from django.db import models

class Answer(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    content = models.TextField()
    create_date = models.DateTimeField()

    def __str__(self):
        return self.content
```

# 데이터베이스와 테이블 생성

## ◆ makemigrations 명령으로 테이블 관리 시작하기

```
(mysite)C:\webproject\pyweb>python manage.py makemigrations
```

```
operations = [
    migrations.CreateModel(
        name='Question',
        fields=[
            ('id', models.BigAutoField(auto_created=True, primary_key=True,
            ('subject', models.CharField(max_length=200)),
            ('content', models.TextField()),
            ('create_date', models.DateTimeField()),
        ],
    ),
]
```

migrations/0001.initial.py

## ◆ makemigrations 후 다시 migrate 명령 실행

```
(mysite)C:\webproject\pyweb>python manage.py migrate
```

실행후 OK가 승인되고, sqlite3 db에 저장됨

# 데이터 저장 및 조회하기

## ◆ 장고 셸 실행하기

```
(mysite)C:\webproject\pyweb>python manage.py shell
```

## ◆ 질문 만들기 – Question 모델의 객체 생성

```
>>> from pybo.models import Question, Answer
>>> from django.utils import timezone
>>> q = Question(subject='pybo란 무엇인가', content='pybo에 대해
설명해 주세요', create_date=timezone.now())
>>> q.save() #저장, 수정시 반드시 사용 commit와 같음
>>> q.id
>>> q.subject
```

# 데이터 저장 및 조회하기

## ◆ 답변 만들기 – Answer 모델의 객체 생성

```
>>> q1 = Question.objects.get(id=1)
>>> a = Answer(question=q1, content='pybo란 파이썬 장고 프레임워크로 만든
게시판입니다.', create_date=timezone.now())
>>> a.save()
>>> q1.answer_set.all() # 연결된 데이터 조회하기
<QuerySet [<Answer: pybo란 파이썬 장고 프레임워크로 만든 게시판입니다.>]>
>>>
```

# 데이터 변경 및 삭제하기

## ◆ 데이터 수정

```
>>>q = Question.objects.get(id=2)
>>>q.subject = 'Django Model Question'
>>>q.save()
```

## ◆ 데이터 삭제

```
>>>q = Question.objects.get(id=1)
>>>q.delete()
```



# 장고 Admin

- ◆ 장고 Admin에서 모델 관리하기
  - board/admin.py 에 등록

```
from django.contrib import admin
from .models import Question, Answer

admin.site.register(Question)
admin.site.register(Answer)
```

# 장고 Admin

## ◆ 장고 Admin에서 모델 관리하기

### Django 관리

홈 > Pybo > Questions

PYBO	
Answers	+ 추가
Questions	+ 추가

인증 및 권한	
그룹	+ 추가
사용자(들)	+ 추가

### 변경할 question 선택

액션:

<input type="checkbox"/>	QUESTION
<input type="checkbox"/>	장고로 만든 유명한 사이트가 있나요?
<input type="checkbox"/>	pybo란 무엇인가요

2 questions

# 장고 Admin

## ◆ 장고 Admin에서 질문 추가하기

question 추가

**Subject:** 장고로 만든 유명한 사이트가 있나요?

**Content:** 장고로 만든 유명한 사이트가 있는지 소개해 주세요.

**Create date:**

날짜: 2021-10-11 오늘 | 

시각: 07:09:10 현재 | 

# 장고 Admin

## ◆ 장고 Admin에서 답변 추가하기

answer 추가

Question:

장고로 만든 유명한 사이트가 있나요? ▾  

Content:

외국 사이트로 인스타그램이 대표적이고  
우리나라는 요기요 사이트가 장고로 만든 사이트입니다.

Create date:

날짜: 2021-10-11 

시각: 07:23:49 

# 장고 Admin

## ◆ DB 브라우저로 보기 – Question, Answer 테이블

테이블(T): board\_question

	id	subject	content	create_date
...		필터	필터	필터
1	1	pybo란 무엇인가?	pybo에 대하여 설명해 주세요	2021-12-17 13:18:24.950375
2	2	장고 프레임워크로 만든 사이트	장고 프레임워크로 만든 유명 사이트...	2021-12-17 13:39:20

테이블(T): board\_answer

	id	content	create_date	question_id
...		필터	필터	필터
1	1	pybo란 파이썬 장고 프레임워크로 만든 게시판입니다.	2021-12-17 13:21:14.256848	1
2	2	외국사이트로는 인스타그램이 대표적이고...	2021-12-17 13:40:38	2