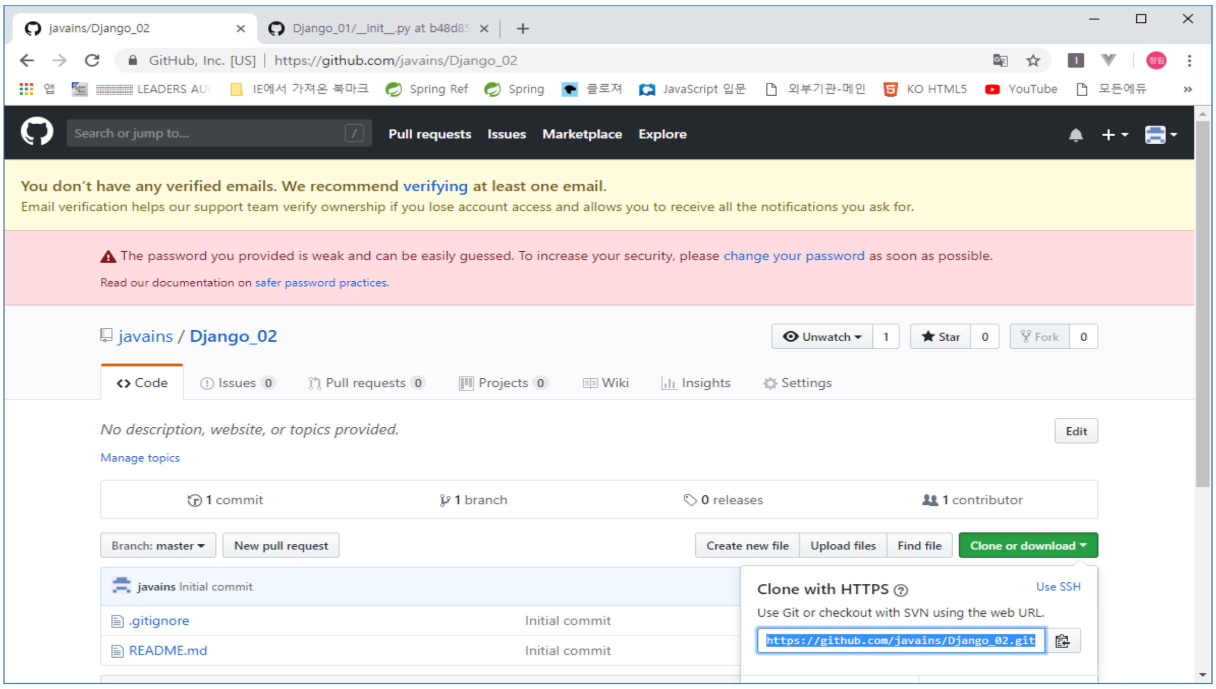
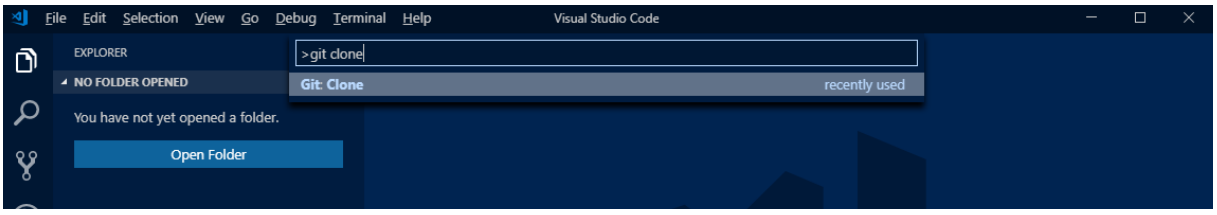
https://github.com/javains/Django_02.git 연동


Clone with HTTPS

Use Git or checkout with SVN using the web URL.

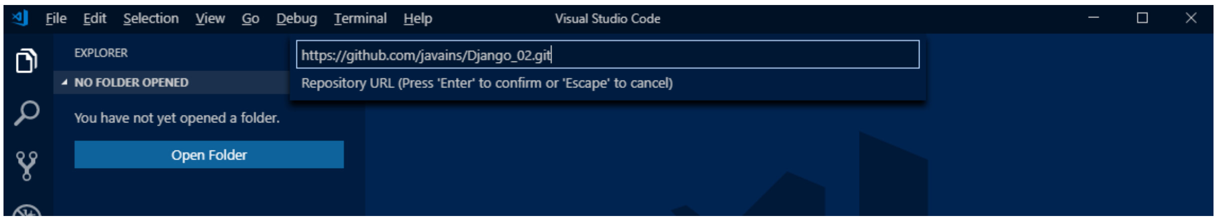
https://github.com/javains/Django_02.git

https://github.com/javains/Django_02.git

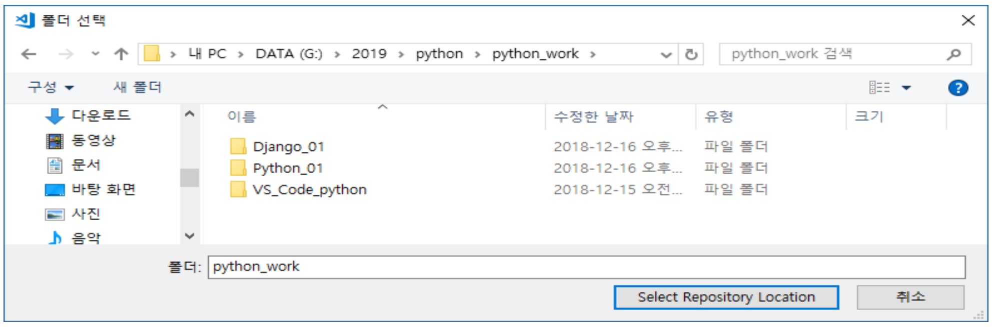

>git clone

Git: Clone recently used

ctrl+shift +p => git clone


https://github.com/javains/Django_02.git

Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)

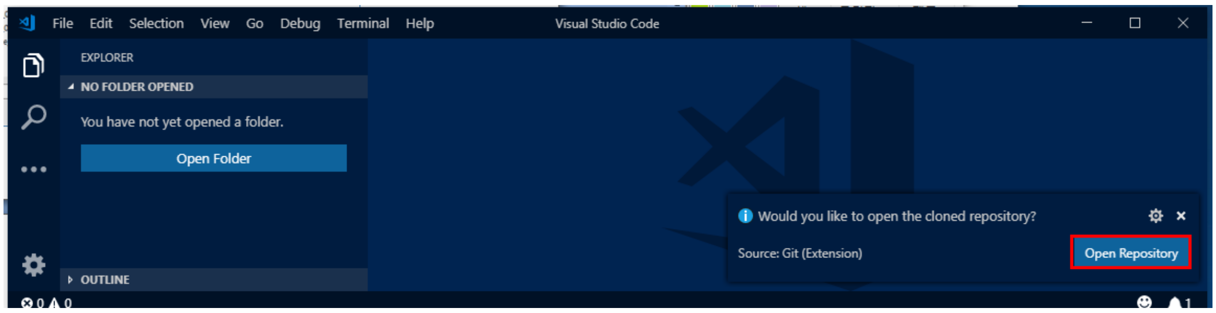

폴더 선택

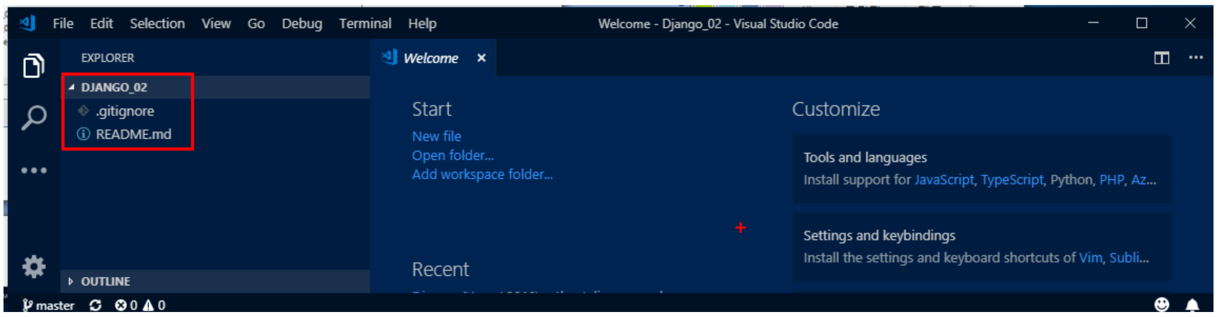
내 PC > DATA (G:) > 2019 > python > python_work

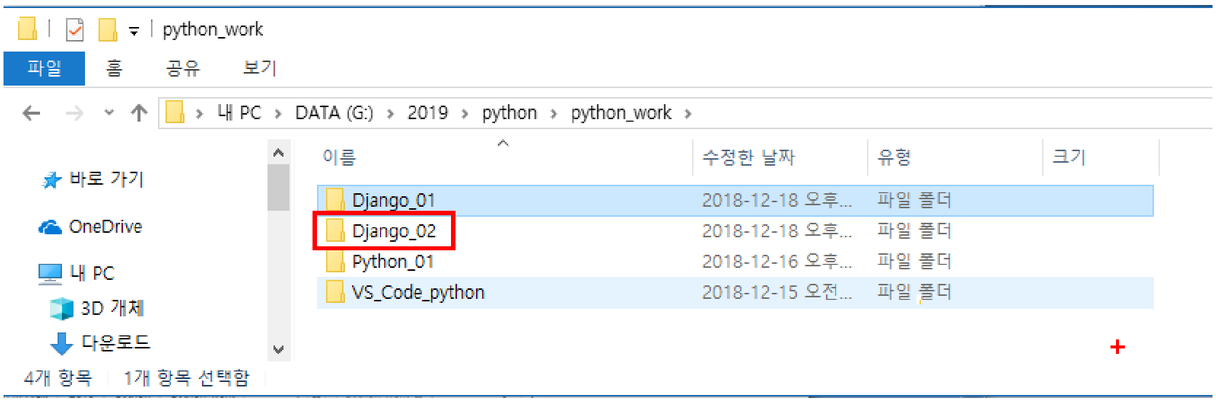
이름	수정한 날짜	유형	크기
Django_01	2018-12-16 오후...	파일 폴더	
Python_01	2018-12-16 오후...	파일 폴더	
VS_Code_python	2018-12-15 오전...	파일 폴더	

폴더: python_work

Select Repository Location


File Edit Selection View Go Debug Terminal Help
Visual Studio Code
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Would you like to open the cloned repository?
Source: Git (Extension)
Open Repository

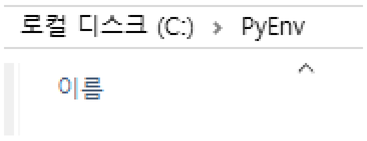

File Edit Selection View Go Debug Terminal Help
Welcome - Django_02 - Visual Studio Code
EXPLORER
DJANGO_02
.gitignore
README.md
OUTLINE
master
Welcome
Start
New file
Open folder...
Add workspace folder...
Recent
Customize
Tools and languages
Install support for JavaScript, TypeScript, Python, PHP, Az...
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Subli...


python_work
파일 홈 공유 보기
내 PC › DATA (G:) › 2019 › python › python_work ›
바로 가기
OneDrive
내 PC
3D 개체
다운로드

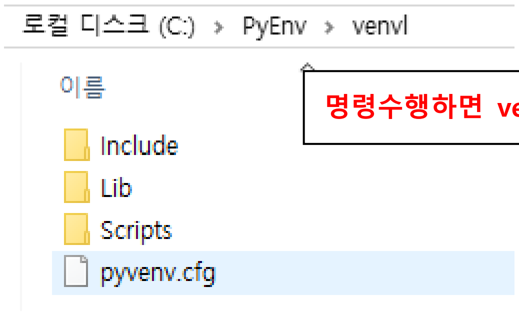
이름	수정한 날짜	유형	크기
Django_01	2018-12-18 오후...	파일 폴더	
Django_02	2018-12-18 오후...	파일 폴더	
Python_01	2018-12-16 오후...	파일 폴더	
VS_Code_python	2018-12-15 오전...	파일 폴더	

4개 항목 1개 항목 선택함

1.가상환경 만들 폴더 생성

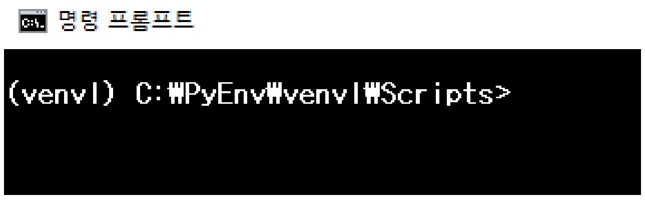


2. C:₩Users₩USER₩AppData₩Local₩Programs₩Python₩Python36-32>
   python Tools/Scripts/pyvenv.py c:/PyEnv/venvl



명령수행하면 venv1 폴더구조생성

3. C:₩PyEnv₩venvl₩Scripts>activate
윈도우즈에서 가상환경을 활성화(activate)



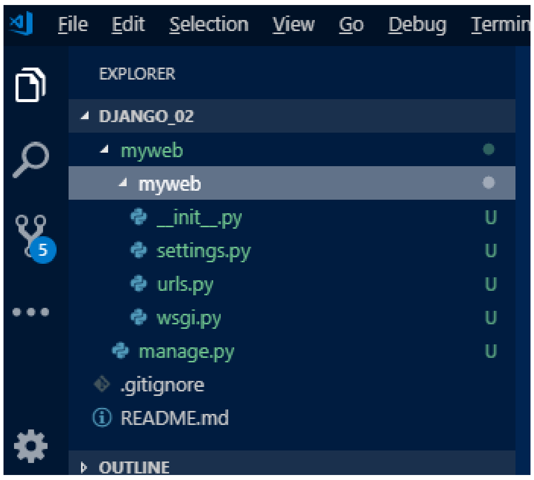
```
(venvl) C:₩PyEnv₩venvl₩Scripts>
```

종료는 exit

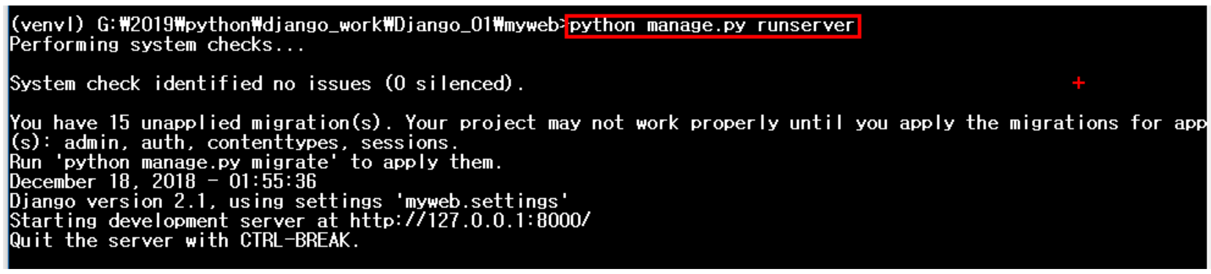
4. (venvl) C:₩PyEnv₩venvl₩Scripts>python -m pip install --upgrade pip
   pip upgrade

5. (venvl) C:₩PyEnv₩venvl₩Scripts>pip install django==2.1
   django 2.1 버전 설치

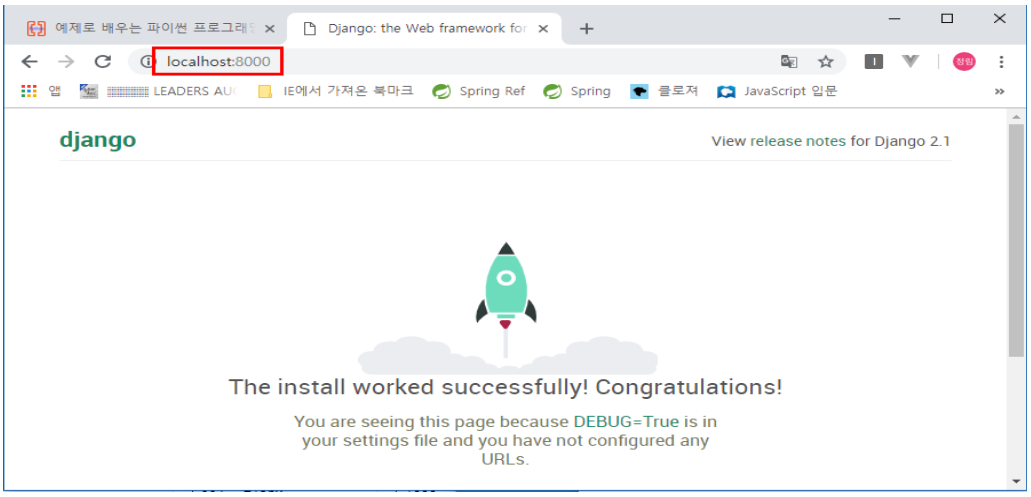
6. (venvl) G:₩2019₩python₩python_work₩Django_02>django-admin startproject myweb
프로젝트를 만들 디렉토리로 이동한 후, 아래와 같이 "django-admin startproject 프로젝트명" 를 실행하여 새 프로젝트를 생성한다.



7. (venvl) G:₩2019₩python₩django_work₩Django_02myweb>python manage.py runserver



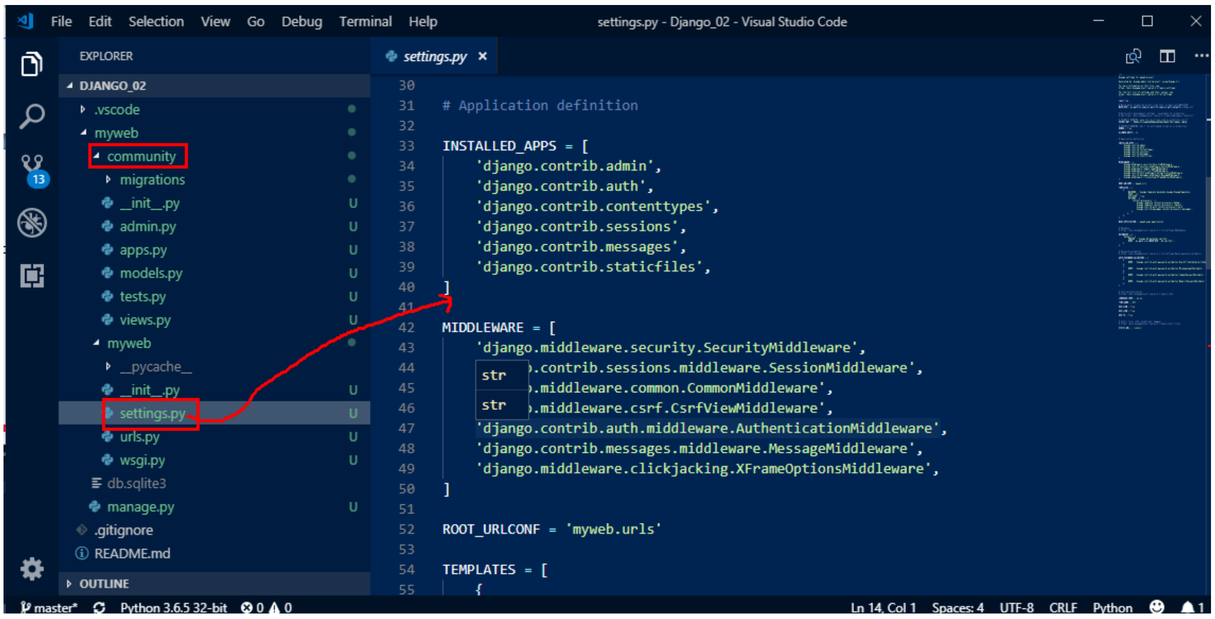
```
(venvl) G:₩2019₩python₩django_work₩Django_01₩myweb>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app
(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
December 18, 2018 - 01:55:36
Django version 2.1, using settings 'myweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```


예제로 배우는 파이썬 프로그래
Django: the Web framework for
localhost:8000
앱
LEADERS AU
IE에서 가져온 북마크
Spring Ref
Spring
클로저
JavaScript 입문
django
View release notes for Django 2.1

The install worked successfully! Congratulations!

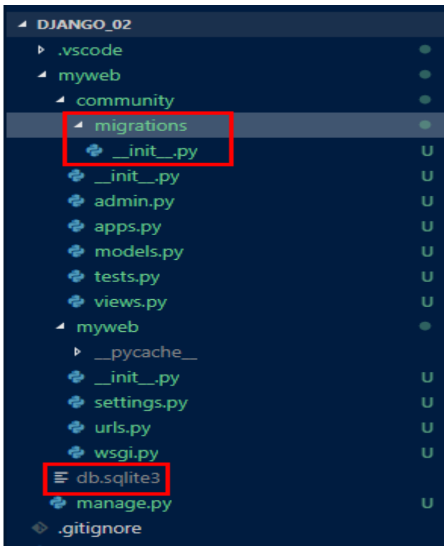
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

8. (venvl) G:₩2019~~_work₩Django_01₩myweb>python manage.py startapp community

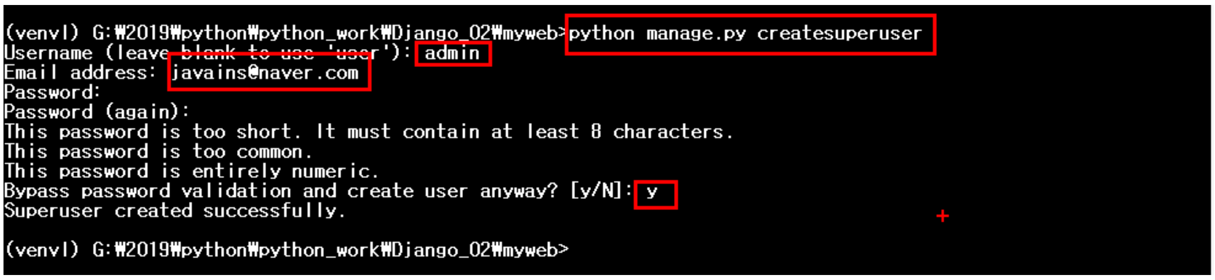


```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    str  ).contrib.sessions.middleware.SessionMiddleware',
         ).middleware.common.CommonMiddleware',
    str  ).middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'myweb.urls'

TEMPLATES = [
    {
```

PS G:₩2019₩python₩django_work₩Django_02₩myweb> python manage.py migrate



기본적인 DB가 만들어 진다.

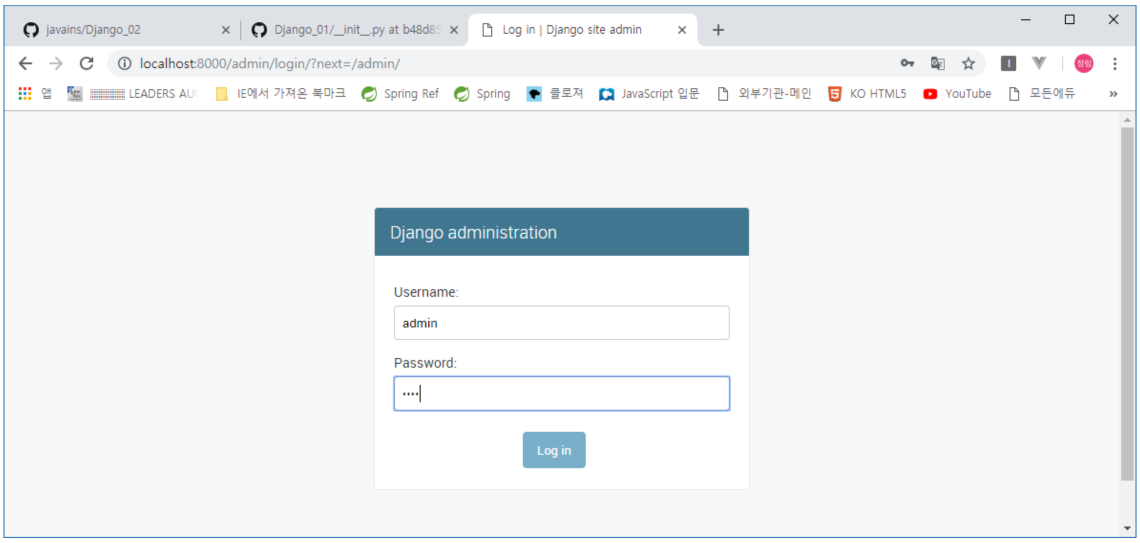


```
(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py createsuperuser
Username (leave blank to use 'user'): admin
Email address: javains@naver.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>
```

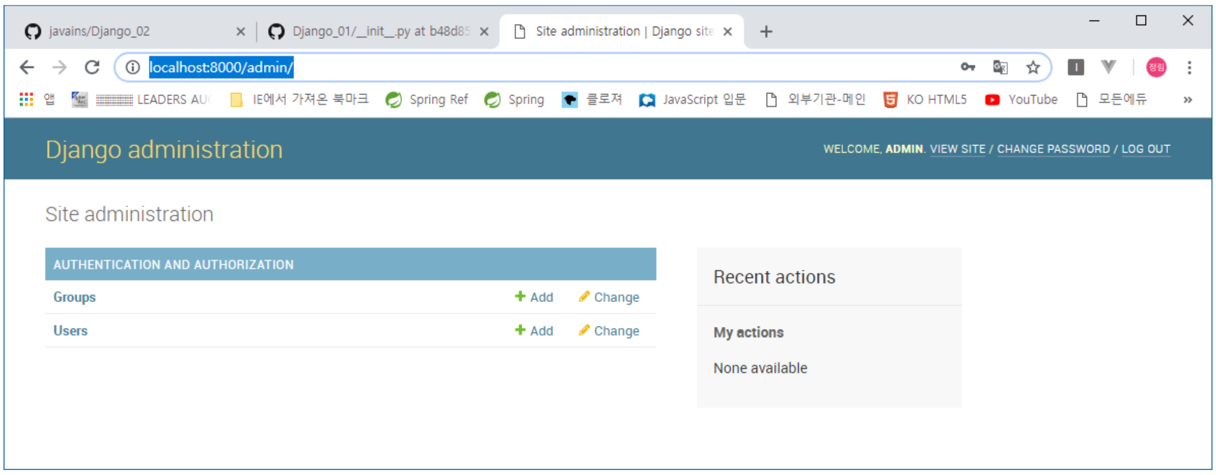
superuser : admin/1234

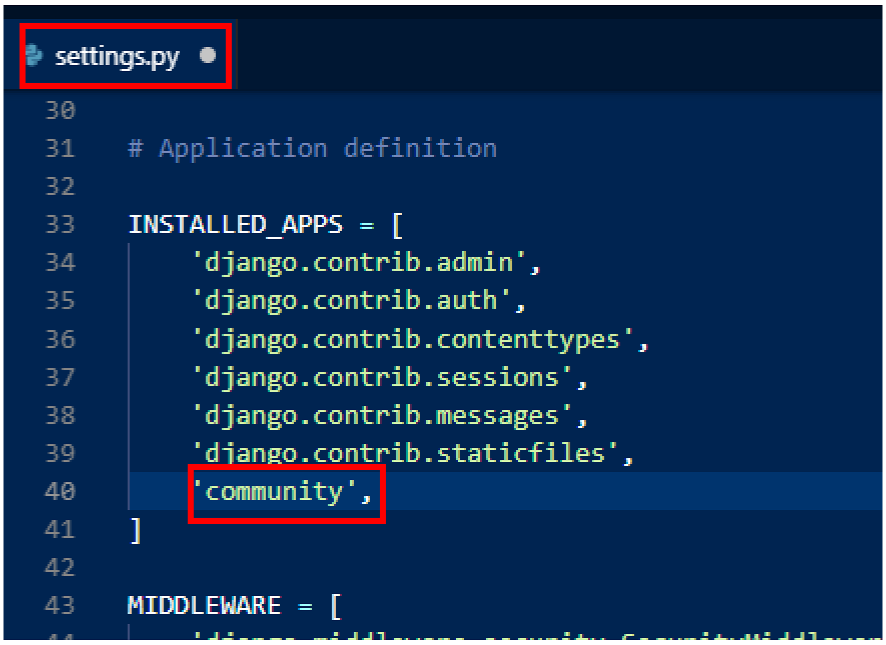
(venvl) G:₩~~_work₩Django_02₩myweb>python manage.py createsuperuser

```
(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py runserver
```


javains/Django_02
Django_01/__init__.py at b48d8
Log in | Django site admin
localhost:8000/admin/login/?next=/admin/
Django administration
Username:
admin
Password:
Log in

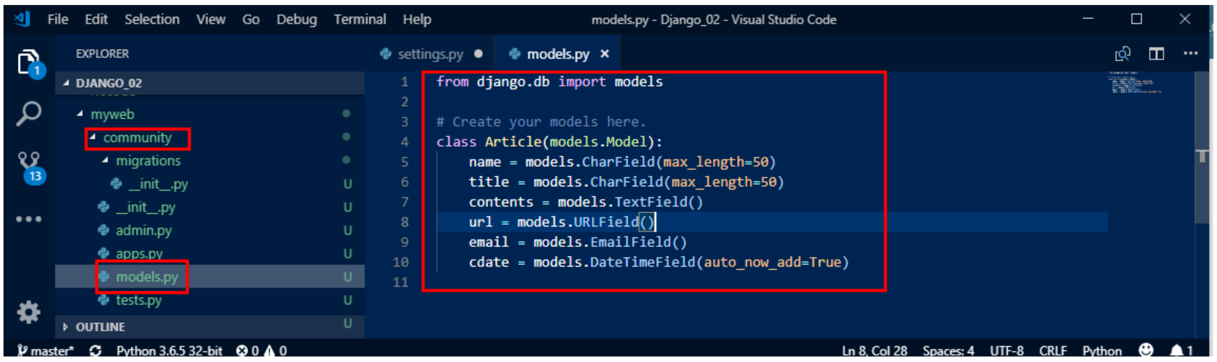
http://localhost:8000/admin/


javains/Django_02
Django_01/__init__.py at b48d8
Site administration | Django site
localhost:8000/admin/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
Recent actions
My actions
None available

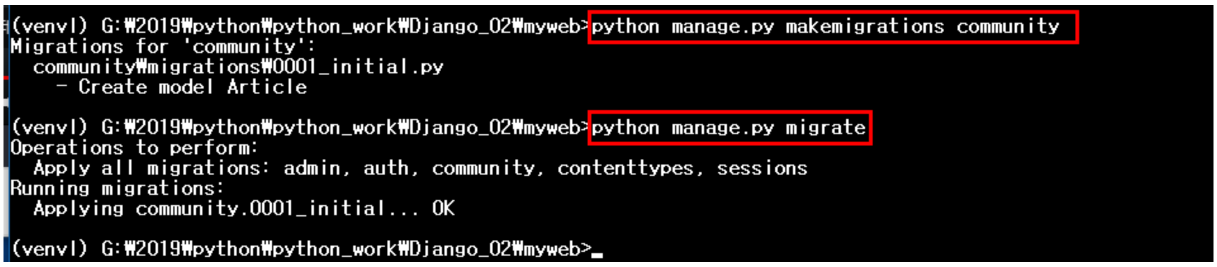


```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'community',
]

MIDDLEWARE = [
```

```python
from django.db import models
# Create your models here.
class Article(models.Model):
    name = models.CharField(max_length=50)
    title = models.CharField(max_length=50)
    contents = models.TextField()
    url = models.URLField()
    email = models.EmailField()
    cdate = models.DateTimeField(auto_now_add=True)
```



```python
from django.db import models

# Create your models here.
class Article(models.Model):
    name = models.CharField(max_length=50)
    title = models.CharField(max_length=50)
    contents = models.TextField()
    url = models.URLField()
    email = models.EmailField()
    cdate = models.DateTimeField(auto_now_add=True)
```



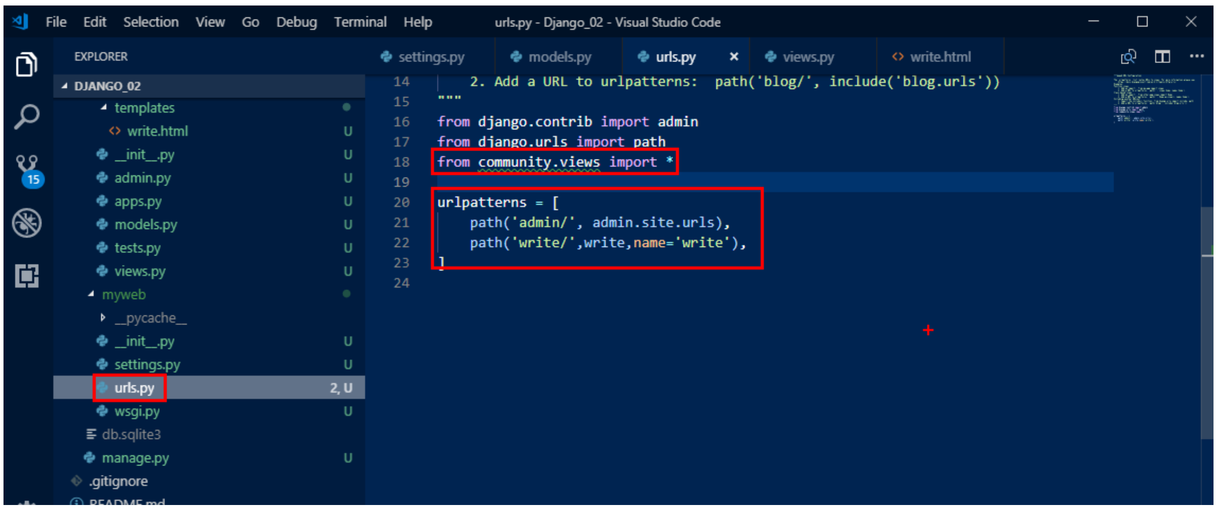
```
(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py makemigrations community
Migrations for 'community':
  community₩migrations₩0001_initial.py
    - Create model Article

(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, community, contenttypes, sessions
Running migrations:
  Applying community.0001_initial... OK

(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>_
```

(venvl) G~~python_work₩Django_02₩myweb>python manage.py makemigrations community

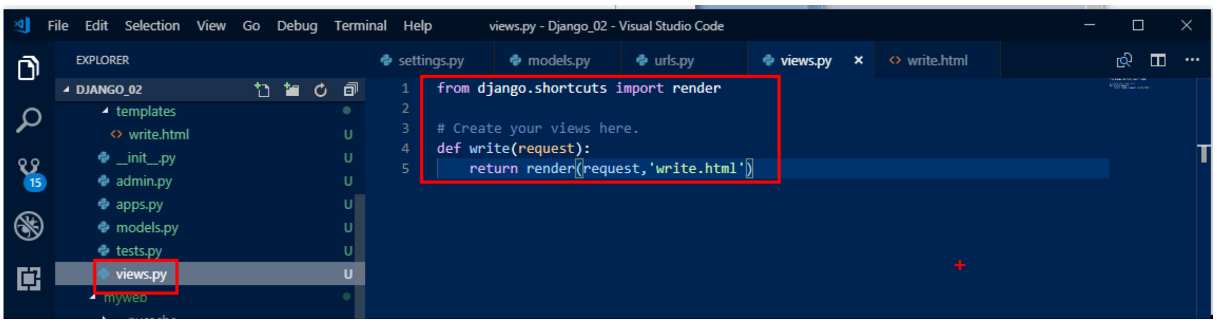
-> 앱에 변화 확인

(venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py migrate

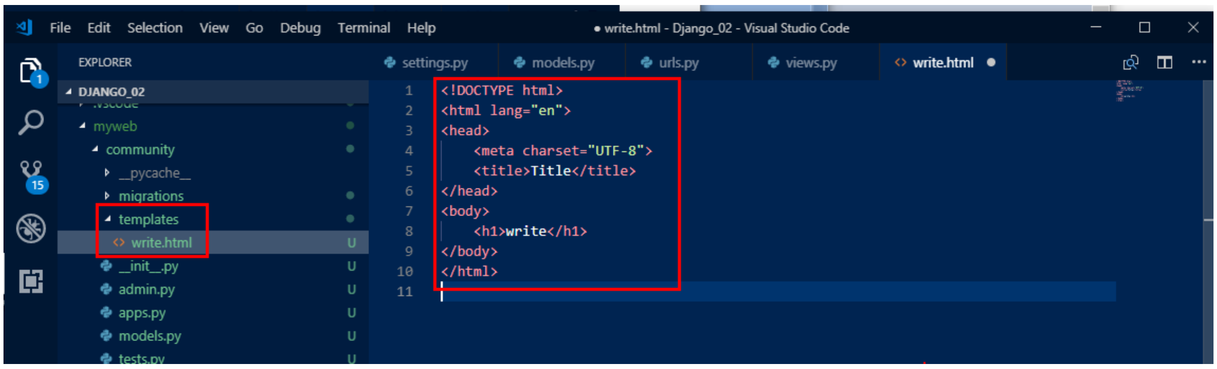
->실제 DB에 테이블 생성



```
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from community.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('write/',write,name='write'),
]
```

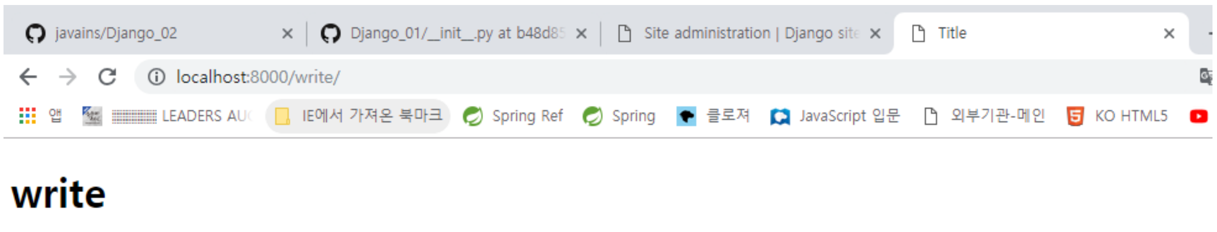


```
from django.shortcuts import render

# Create your views here.
def write(request):
    return render(request,'write.html')
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>write</h1>
</body>
</html>
```

venvl) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py runserver

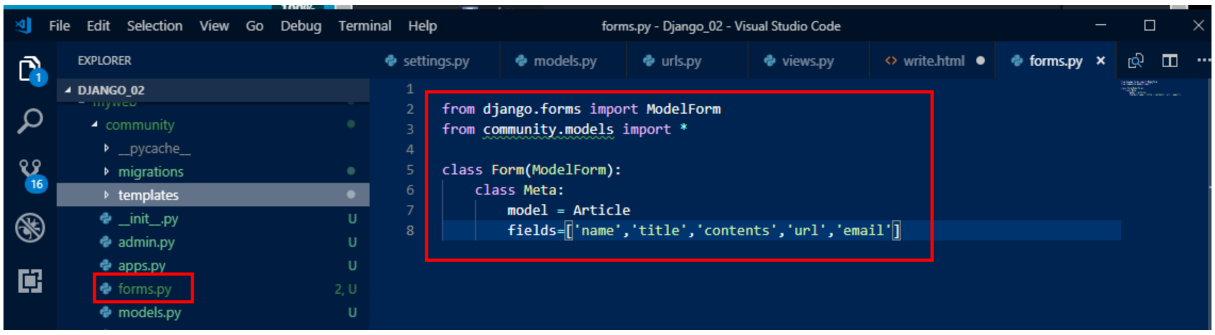


localhost:8000/write/

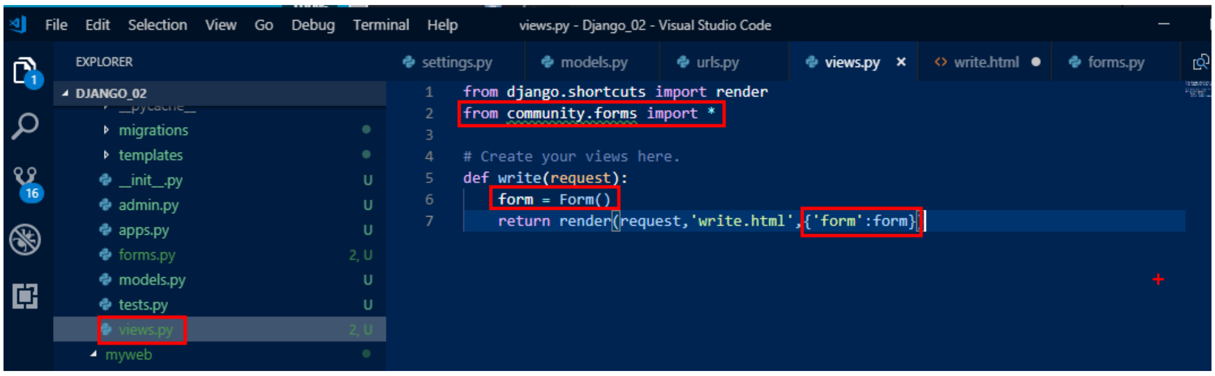
write

http://localhost:8000/write/

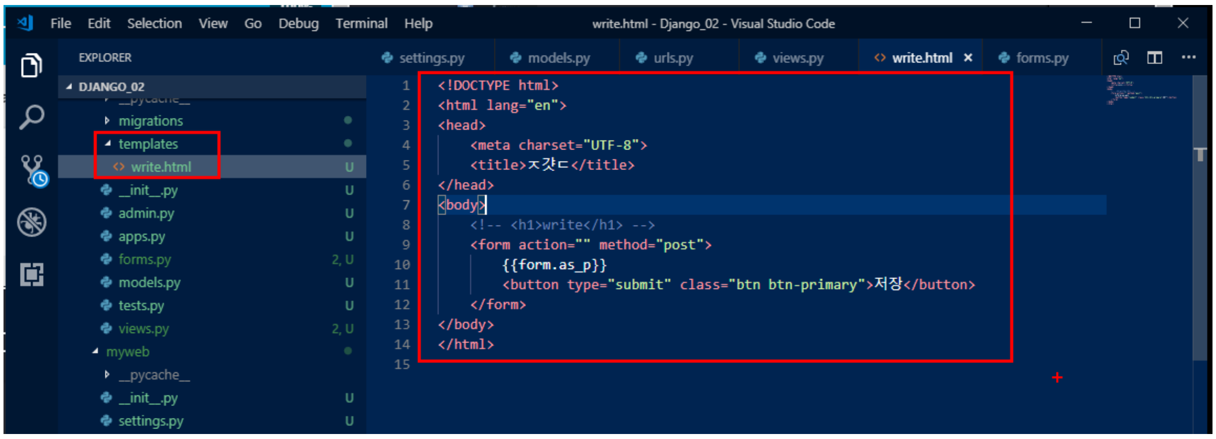
모델을 이용해서 글쓰기 폼 만들기



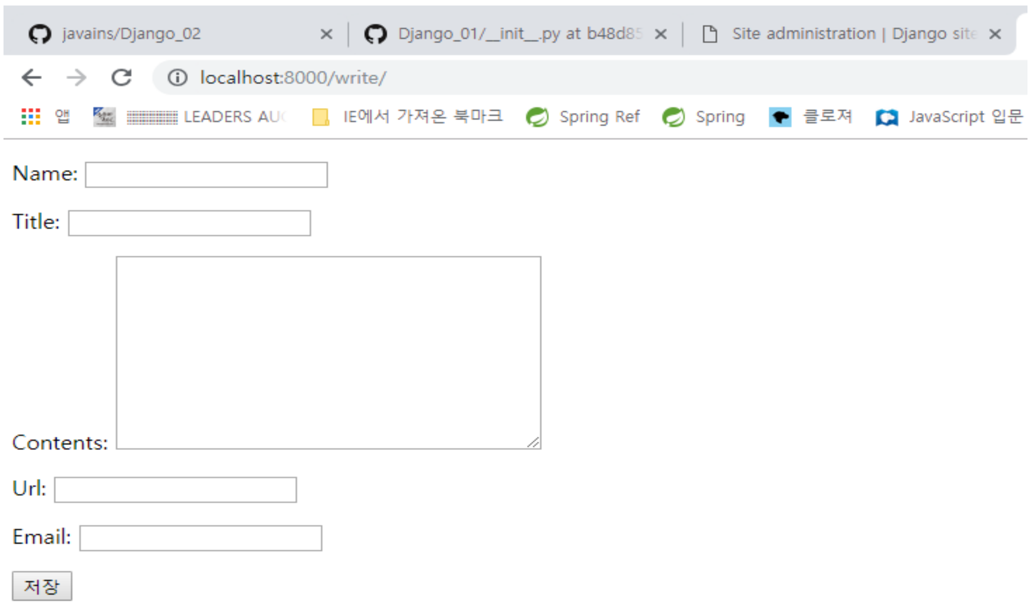
```python
from django.forms import ModelForm
from community.models import *

class Form(ModelForm):
    class Meta:
        model = Article
        fields=['name','title','contents','url','email']
```



```python
from django.shortcuts import render
from community.forms import *

# Create your views here.
def write(request):
    form = Form()
    return render(request,'write.html',{'form':form})
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ㅈ갸ㄷ</title>
</head>
<body>
    <!-- <h1>write</h1> -->
    <form action="" method="post">
        {{form.as_p}}
        <button type="submit" class="btn btn-primary">저장</button>
    </form>
</body>
</html>
```



localhost:8000/write/

Name:

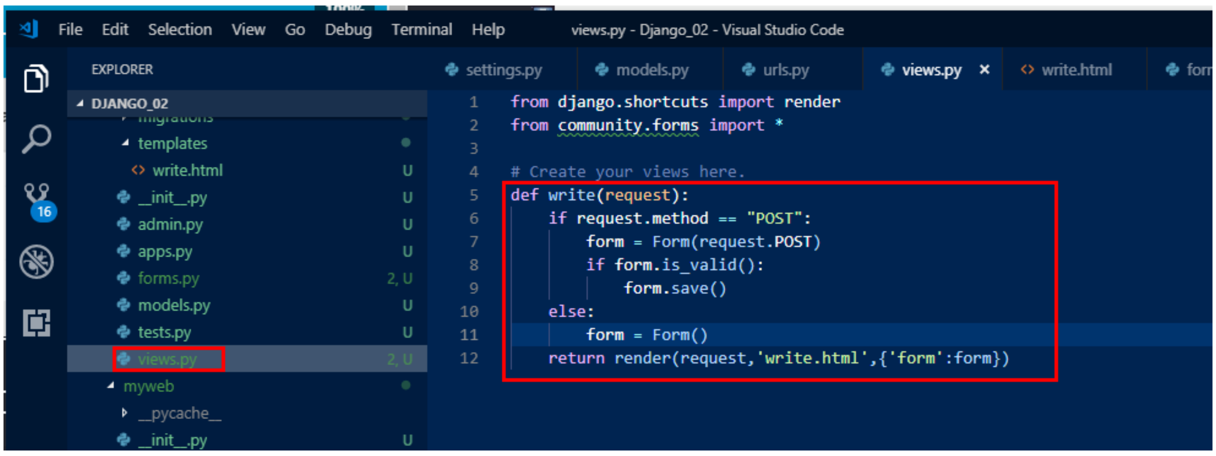
Title:

Contents:

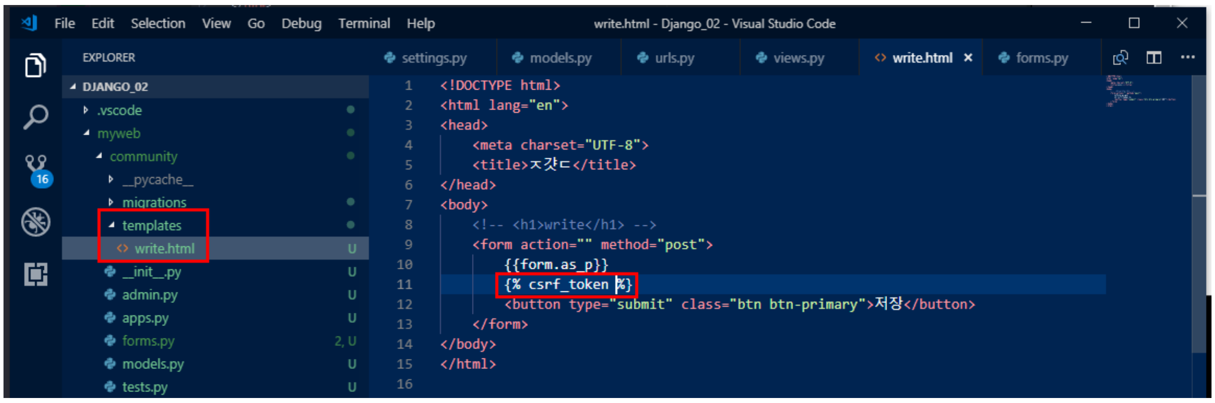
Url:

Email:

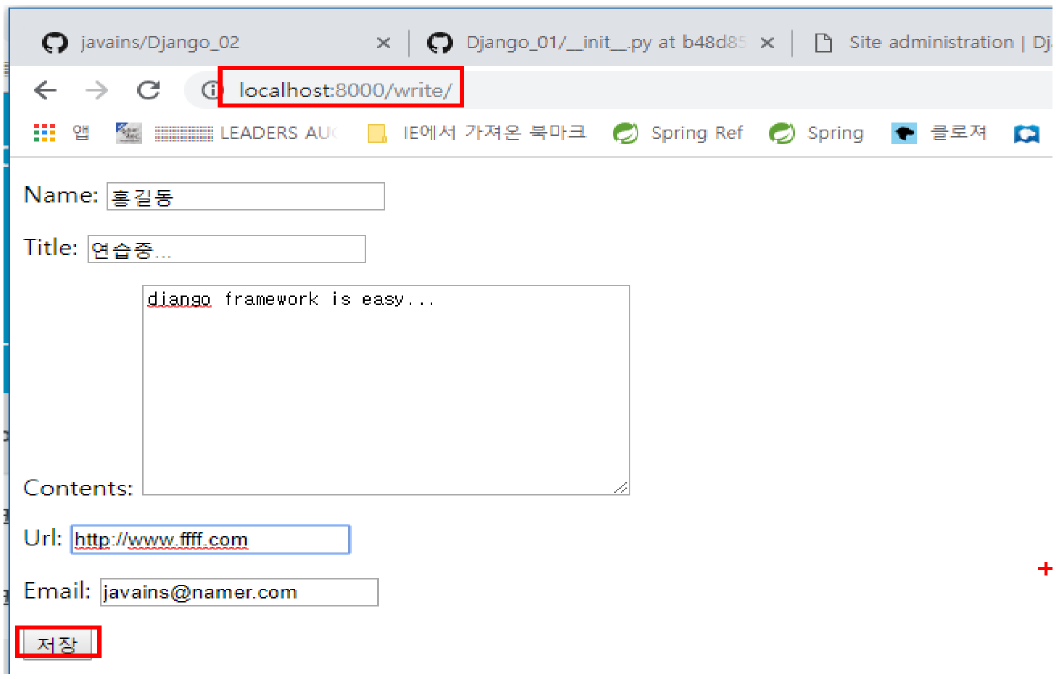
저장



```python
from django.shortcuts import render
from community.forms import *

# Create your views here.
def write(request):
    if request.method == "POST":
        form = Form(request.POST)
        if form.is_valid():
            form.save()
    else:
        form = Form()
    return render(request,'write.html',{'form':form})
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ㅈ갲ㄷ</title>
</head>
<body>
    <!-- <h1>write</h1> -->
    <form action="" method="post">
        {{form.as_p}}
        {% csrf_token %}
        <button type="submit" class="btn btn-primary">저장</button>
    </form>
</body>
</html>
```



Name: 홍길동

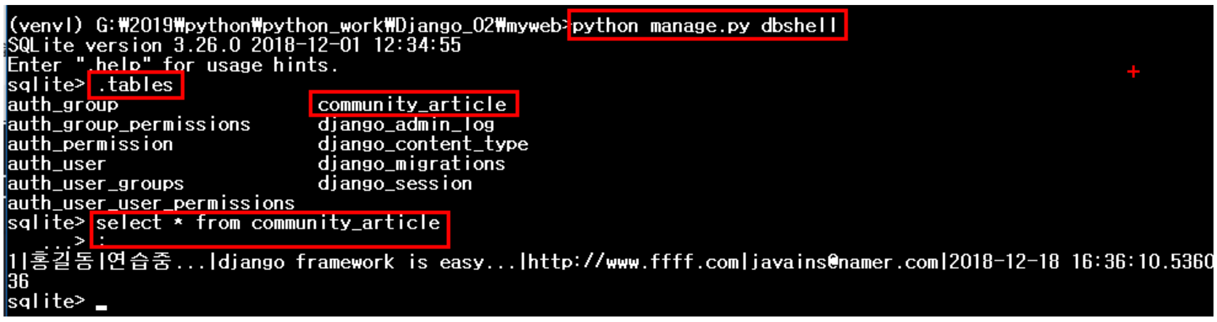
Title: 연습중...

Contents: django framework is easy...

Url: http://www.ffff.com

Email: javains@namer.com

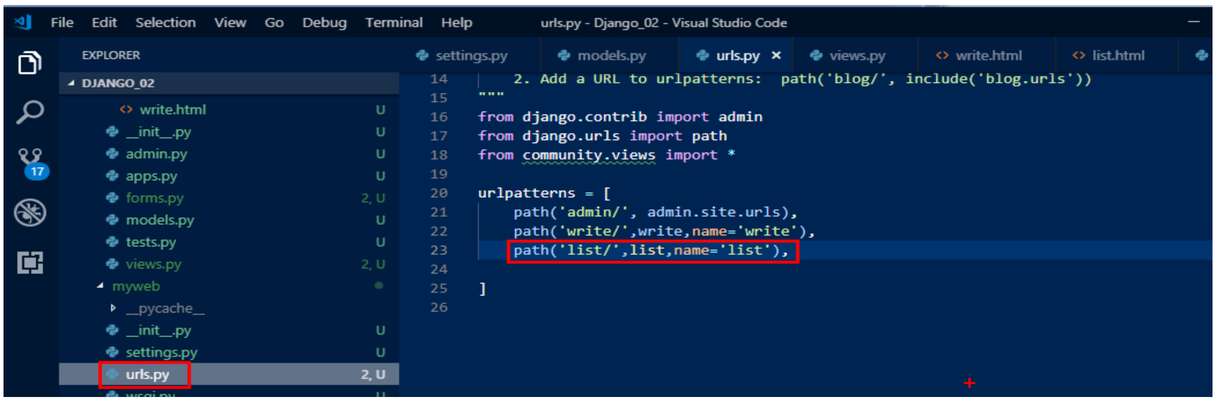
저장



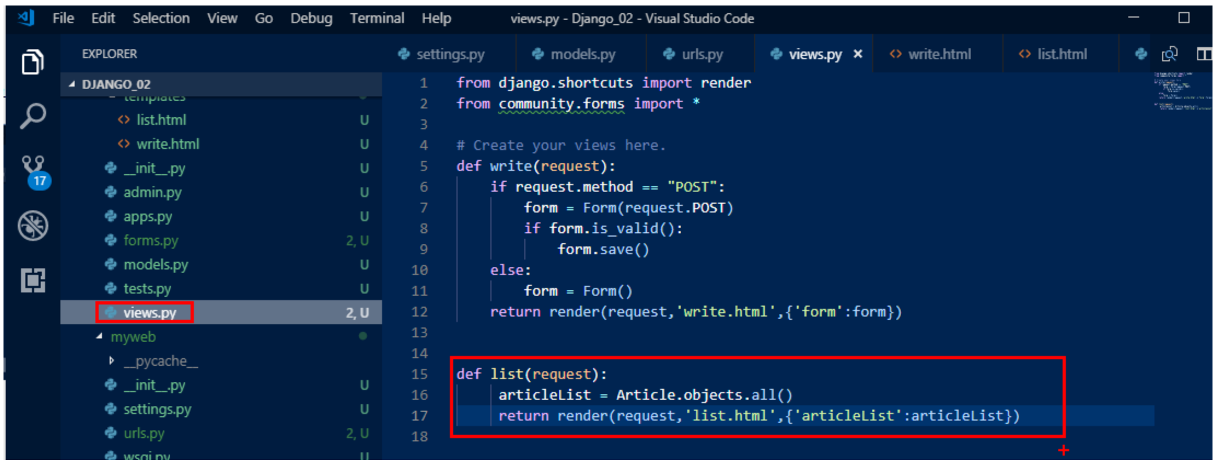
```
(venv1) G:₩2019₩python₩python_work₩Django_02₩myweb>python manage.py dbshell
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> .tables
auth_group                    community_article
auth_group_permissions        django_admin_log
auth_permission               django_content_type
auth_user                     django_migrations
auth_user_groups              django_session
auth_user_user_permissions
sqlite> select * from community_article
   ...> ;
1|홍길동|연습중...|django framework is easy...|http://www.ffff.com|javains@namer.com|2018-12-18 16:36:10.536036
sqlite> _
```

.quit

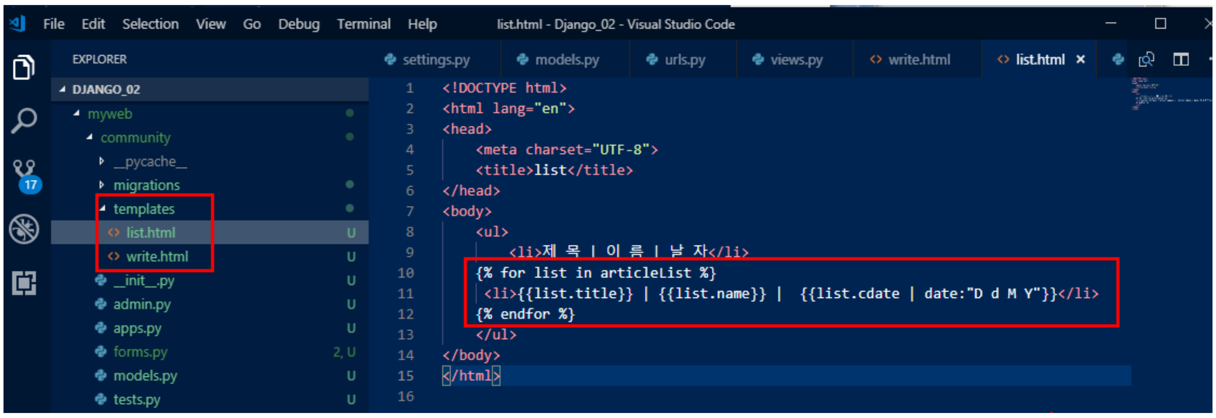
list 구현



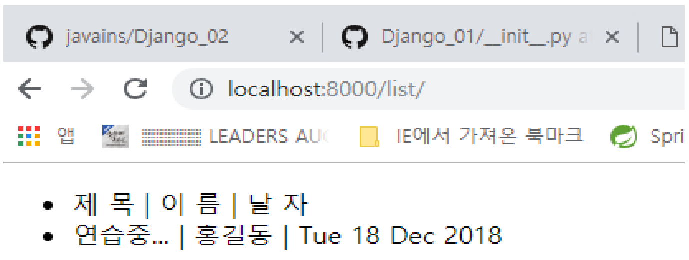
```python
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from community.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('write/',write,name='write'),
    path('list/',list,name='list'),

]
```



```python
from django.shortcuts import render
from community.forms import *

# Create your views here.
def write(request):
    if request.method == "POST":
        form = Form(request.POST)
        if form.is_valid():
            form.save()
    else:
        form = Form()
    return render(request,'write.html',{'form':form})

def list(request):
    articleList = Article.objects.all()
    return render(request,'list.html',{'articleList':articleList})

```

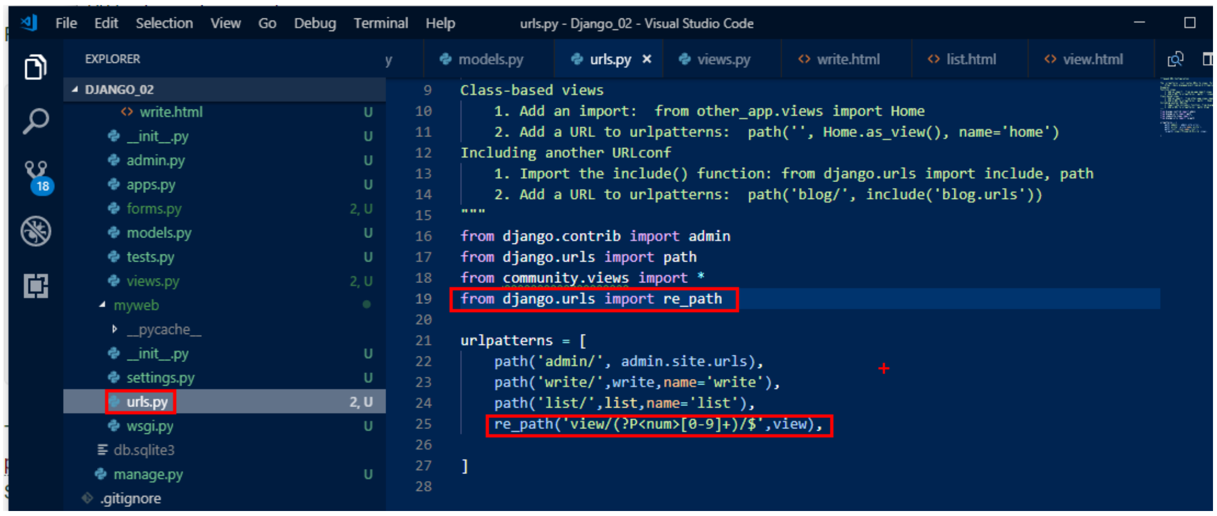


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>list</title>
</head>
<body>
    <ul>
        <li>제 목 | 이 름 | 날 자</li>
    {% for list in articleList %}
    <li>{{list.title}} | {{list.name}} |  {{list.cdate | date:"D d M Y"}}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

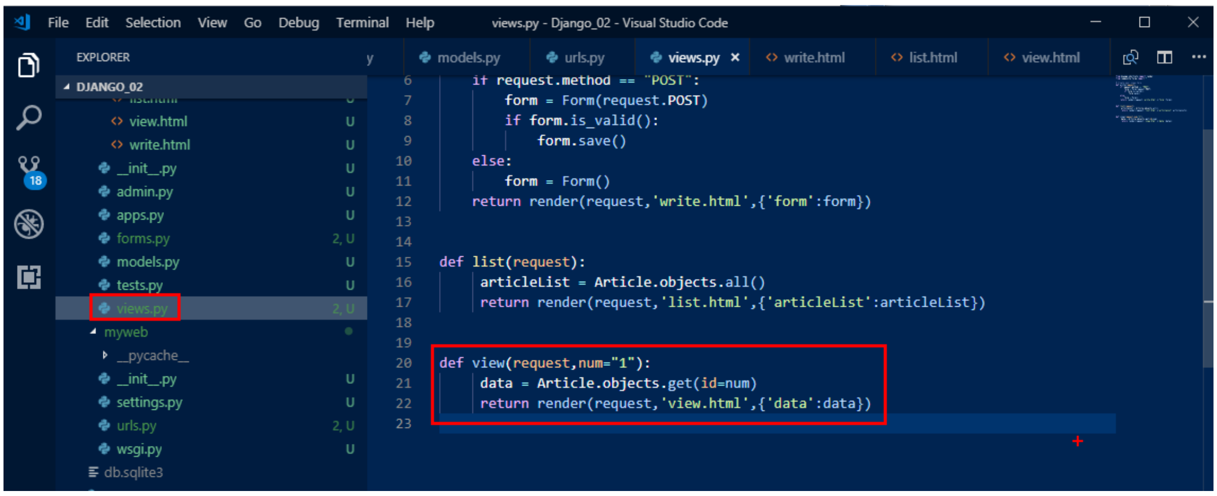


- 제 목 | 이 름 | 날 자
- 연습중... | 홍길동 | Tue 18 Dec 2018

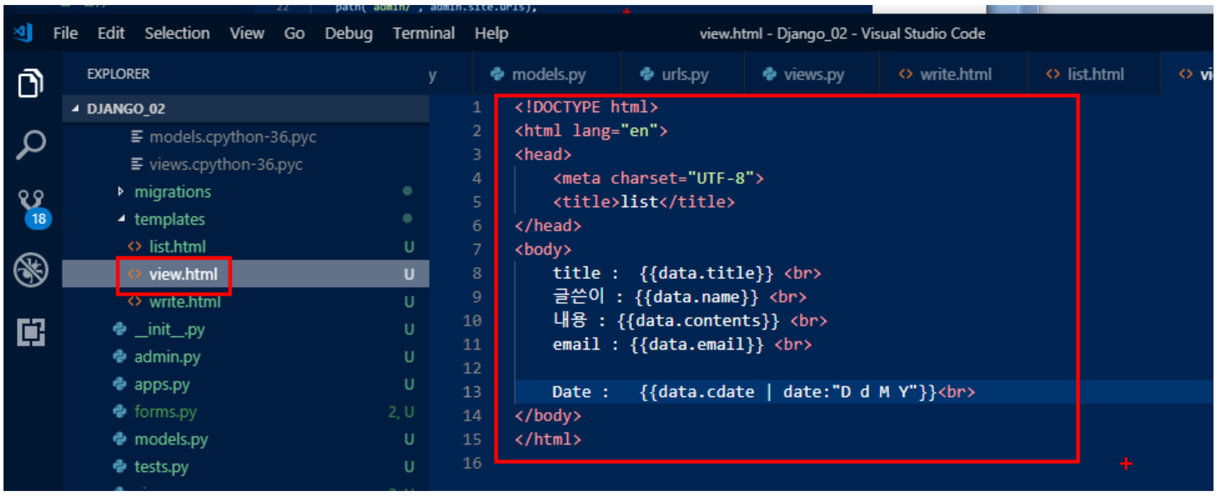
view 구현



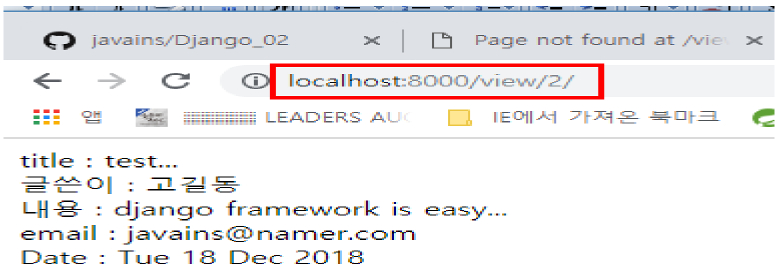
```
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from community.views import *
from django.urls import re_path

urlpatterns = [
    path('admin/', admin.site.urls),
    path('write/',write,name='write'),
    path('list/',list,name='list'),
    re_path('view/(?P<num>[0-9]+)/$',view),

]
```



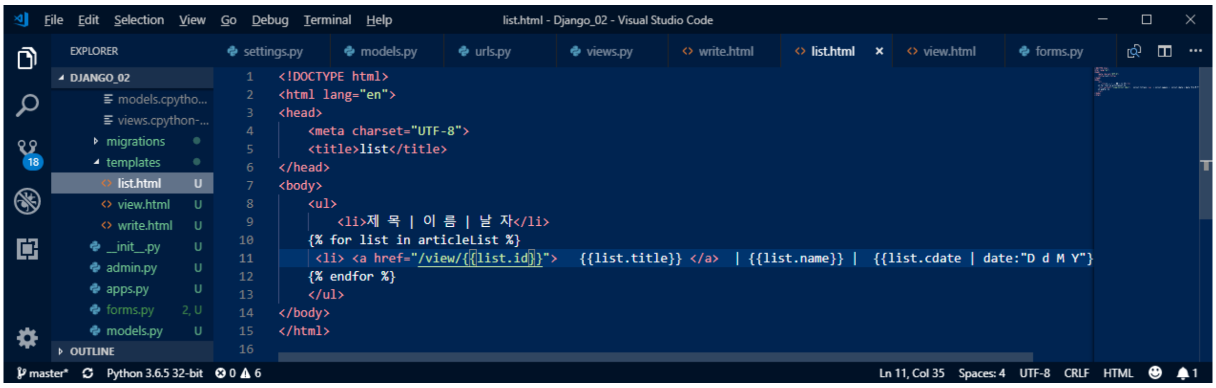
```
    if request.method == "POST":
        form = Form(request.POST)
        if form.is_valid():
            form.save()
    else:
        form = Form()
    return render(request,'write.html',{'form':form})


def list(request):
    articleList = Article.objects.all()
    return render(request,'list.html',{'articleList':articleList})


def view(request,num="1"):
    data = Article.objects.get(id=num)
    return render(request,'view.html',{'data':data})
```



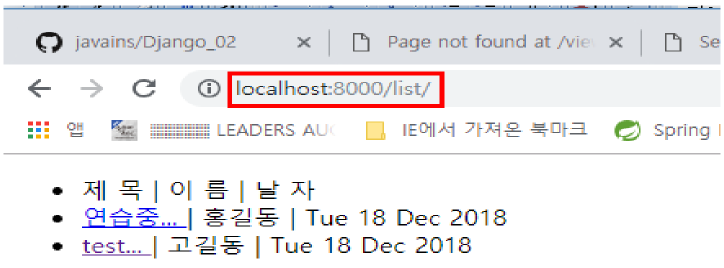
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>list</title>
</head>
<body>
    title :  {{data.title}} <br>
    글쓴이 : {{data.name}} <br>
    내용 : {{data.contents}} <br>
    email : {{data.email}} <br>

    Date :   {{data.cdate | date:"D d M Y"}}<br>
</body>
</html>
```



```
title : test...
글쓴이 : 고길동
내용 : django framework is easy...
email : javains@namer.com
Date : Tue 18 Dec 2018
```



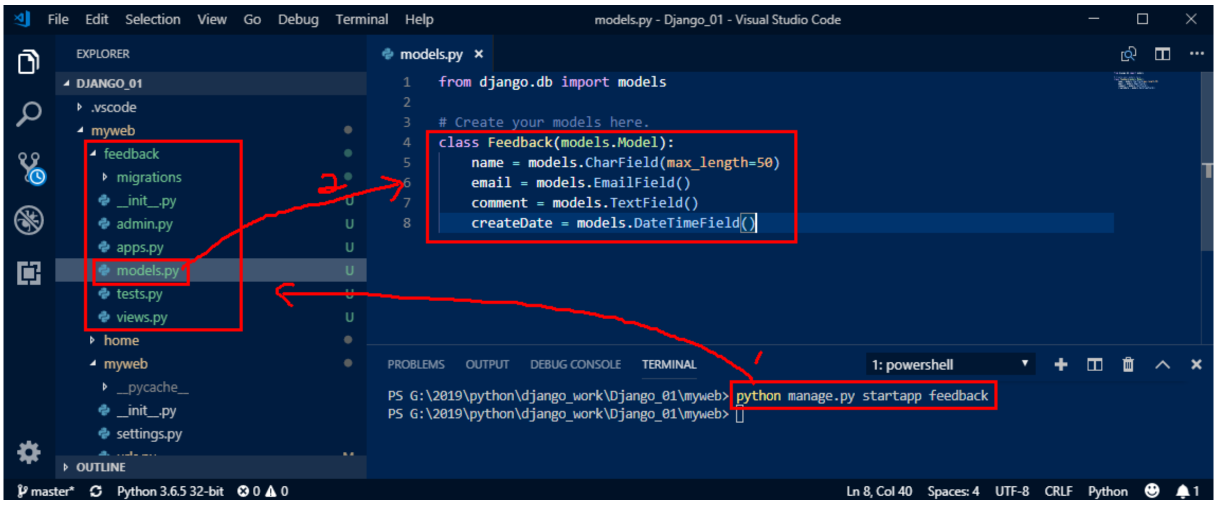
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>list</title>
</head>
<body>
    <ul>
        <li>제 목 | 이 름 | 날 자</li>
    {% for list in articleList %}
     <li> <a href="/view/{{list.id}}">  {{list.title}} </a>  | {{list.name}} |  {{list.cdate | date:"D d M Y"}
    {% endfor %}
    </ul>
</body>
</html>
```



localhost:8000/list/

- 제 목 | 이 름 | 날 자
- 연습중... | 홍길동 | Tue 18 Dec 2018
- test... | 고길동 | Tue 18 Dec 2018

Django 모델 (Model)

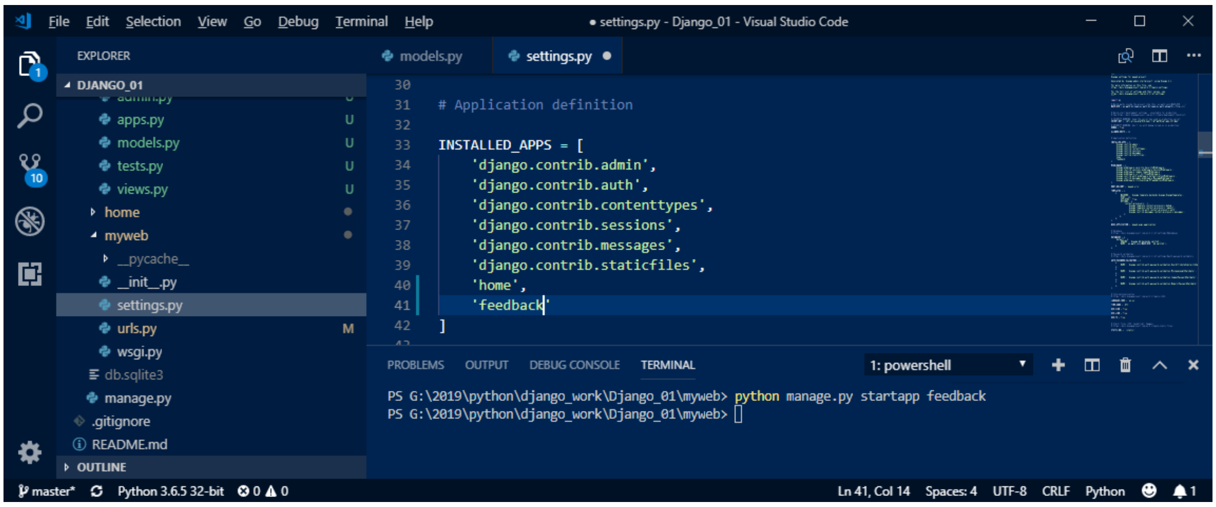

File Edit Selection View Go Debug Terminal Help
models.py - Django_01 - Visual Studio Code
EXPLORER
DJANGO_01
.vscode
myweb
feedback
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
home
myweb
__pycache__
__init__.py
settings.py
OUTLINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
master* Python 3.6.5 32-bit ⊗ 0 ⚠ 0
Ln 8, Col 40 Spaces: 4 UTF-8 CRLF Python

```python
from django.db import models

# Create your models here.
class Feedback(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField()
    comment = models.TextField()
    createDate = models.DateTimeField()
```

```
PS G:\2019\python\django_work\Django_01\myweb> python manage.py startapp feedback
PS G:\2019\python\django_work\Django_01\myweb>
```

```python
class Feedback(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField()
    comment = models.TextField(null=True)
    createDate = models.DateTimeField(auto_now_add=True)
```

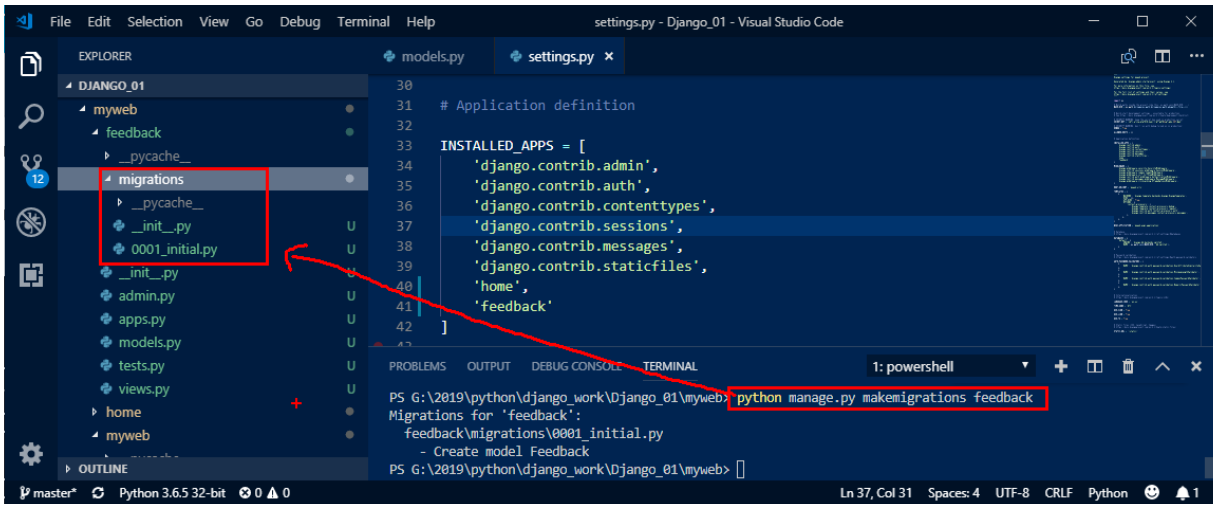
settings.py => INSTALLED_APPS 리스트에 'feedback'


File Edit Selection View Go Debug Terminal Help
● settings.py - Django_01 - Visual Studio Code
EXPLORER
DJANGO_01
admin.py
apps.py
models.py
tests.py
views.py
home
myweb
__pycache__
__init__.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
.gitignore
README.md
OUTLINE
models.py
settings.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
master* Python 3.6.5 32-bit ⊗ 0 ⚠ 0
Ln 41, Col 14 Spaces: 4 UTF-8 CRLF Python

```python

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home',
    'feedback'
]
```

```
PS G:\2019\python\django_work\Django_01\myweb> python manage.py startapp feedback
PS G:\2019\python\django_work\Django_01\myweb>
```

DB Migration

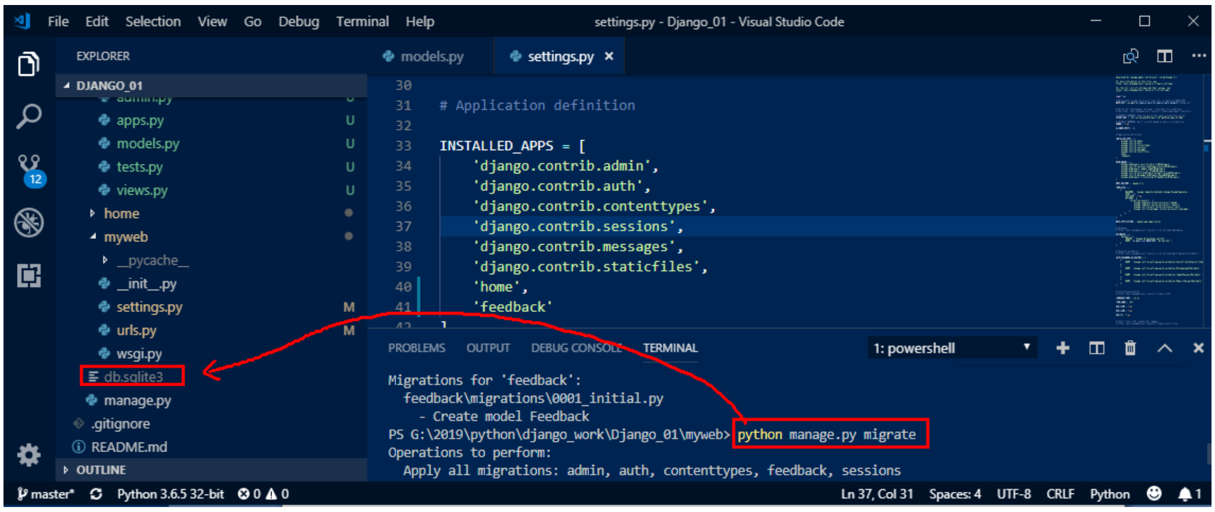


```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home',
    'feedback'
]
```

```
PS G:\2019\python\django_work\Django_01\myweb> python manage.py makemigrations feedback
Migrations for 'feedback':
  feedback\migrations\0001_initial.py
    - Create model Feedback
PS G:\2019\python\django_work\Django_01\myweb> 
```

1. PS G:₩~~~_work₩Django_01₩myweb> python manage.py makemigrations feedback
==> feedback app 에 model 변화가 있는지 확인만한다.

2. PS G:₩2019₩python₩django_work₩Django_01₩myweb> python manage.py migrate
==> model을 DataBase에 적용한다.



```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home',
    'feedback'
```

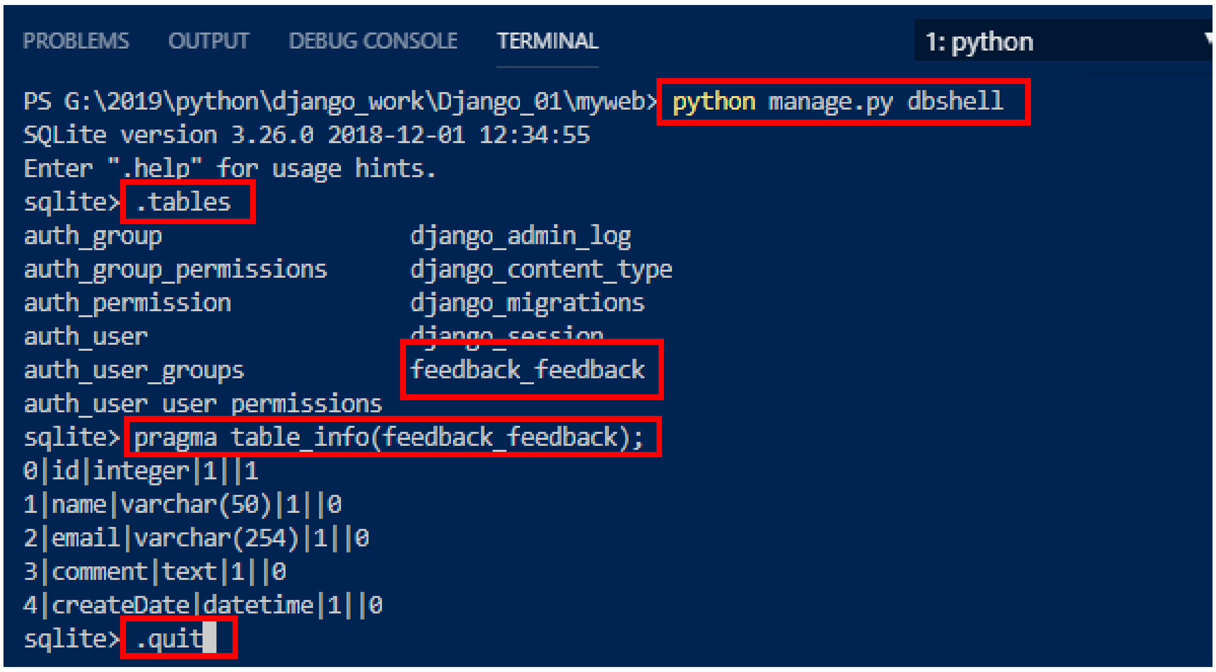
```
Migrations for 'feedback':
  feedback\migrations\0001_initial.py
    - Create model Feedback
PS G:\2019\python\django_work\Django_01\myweb> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, feedback, sessions
```

G:₩2019₩python₩django_work₩Django_01₩myweb> python manage.py dbshell

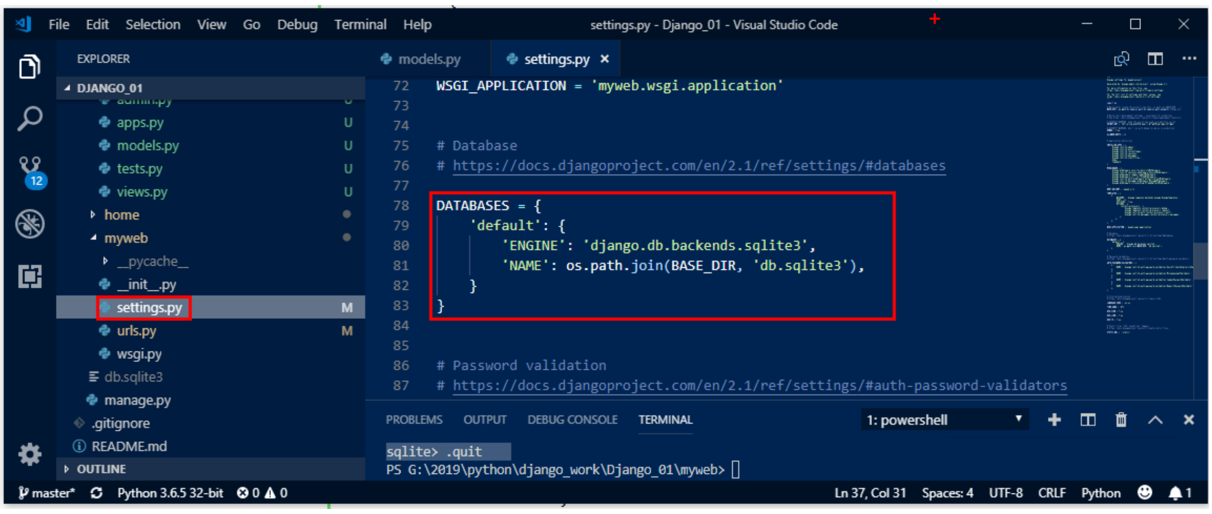
sqlite> .tables

sqlite> pragma table_info(feedback_feedback);

sqlite> .quit



```
PS G:\2019\python\django_work\Django_01\myweb> python manage.py dbshell
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> .tables
auth_group                  django_admin_log
auth_group_permissions      django_content_type
auth_permission             django_migrations
auth_user                   django_session
auth_user_groups            feedback_feedback
auth_user_user_permissions
sqlite> pragma table_info(feedback_feedback);
0|id|integer|1||1
1|name|varchar(50)|1||0
2|email|varchar(254)|1||0
3|comment|text|1||0
4|createDate|datetime|1||0
sqlite> .quit
```



```
WSGI_APPLICATION = 'myweb.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
# https://docs.djangoproject.com/en/2.1/ref/settings/#auth-password-validators
```

```
sqlite> .quit
PS G:\2019\python\django_work\Django_01\myweb>
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'MyDB',
        'USER': 'user1',
        'PASSWORD': 'pwd',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

- django.db.backends.postgresql
- django.db.backends.mysql
- django.db.backends.sqlite3
- django.db.backends.oracle

```
from feedback.models import *
from datetime import datetime
```

INSERT

```
# Feedback 객체 생성
fb = Feedback(name = 'Kim', email = 'kim@test.com', comment='Hi',
createDate=datetime.now())
fb.save()     # 새 객체 INSERT
```

SELECT

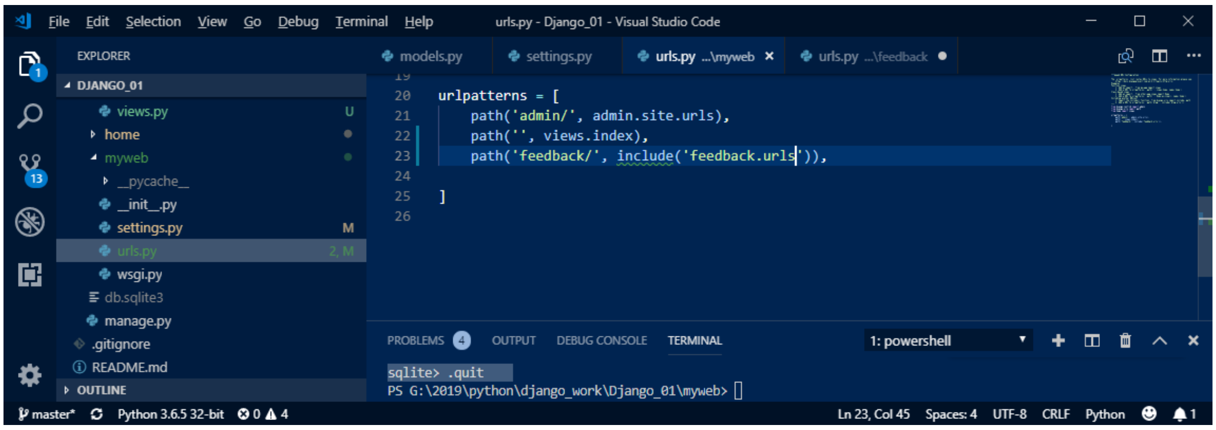
```
for f in Feedback.objects.all():
    s += str(f.id) + ' : ' + f.name + '₩n'

row = Feedback.objects.get(pk=1)
print(row.name)

rows = Feedback.objects.filter(name='Kim')
n = Feedback.objects.count()     # 데이타의 갯수(row 수)
rows = Feedback.objects.distinct('name')
rows = Feedback.objects.order_by('name').first()
rows = Feedback.objects.order_by('name').last()
```

UPDATE

```
fb = Feedback.objects.get(pk=1)
fb.name = 'Park'
fb.save()
```

DELETE

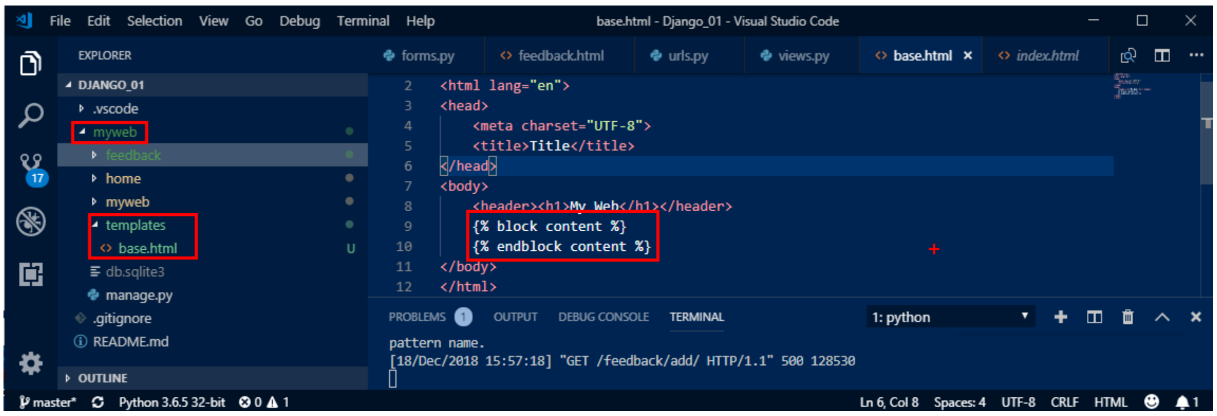
```
fb = Feedback.objects.get(pk=2)
fb.delete()
```



```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index),
    path('feedback/', include('feedback.urls')),

]
```

```
sqlite> .quit
PS G:\2019\python\django_work\Django_01\myweb>
```

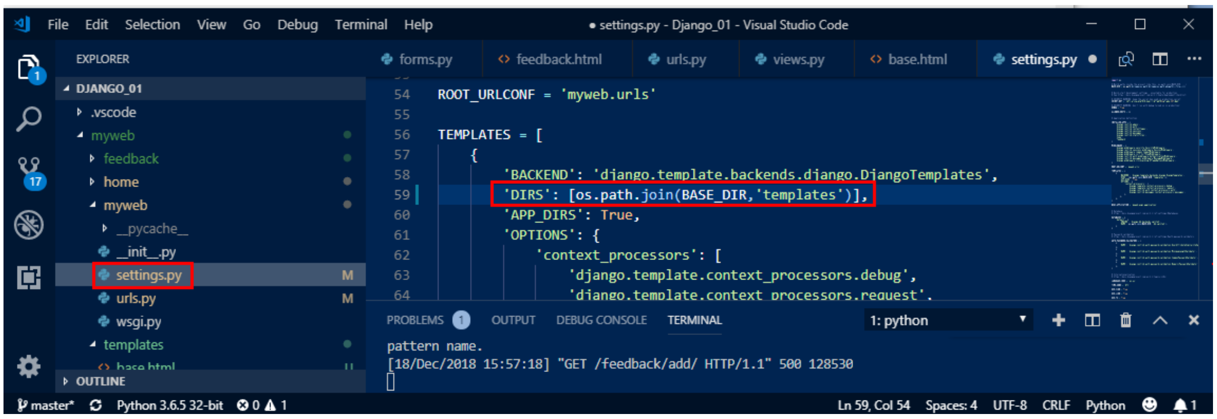
```
from django.contrib import admin
from django.urls import path
from feedback import views
urlpatterns = [
    path('add/', views.add),
]
```



```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <header><h1>My Web</h1></header>
    {% block content %}
    {% endblock content %}
</body>
</html>
```

```
pattern name.
[18/Dec/2018 15:57:18] "GET /feedback/add/ HTTP/1.1" 500 128530
```

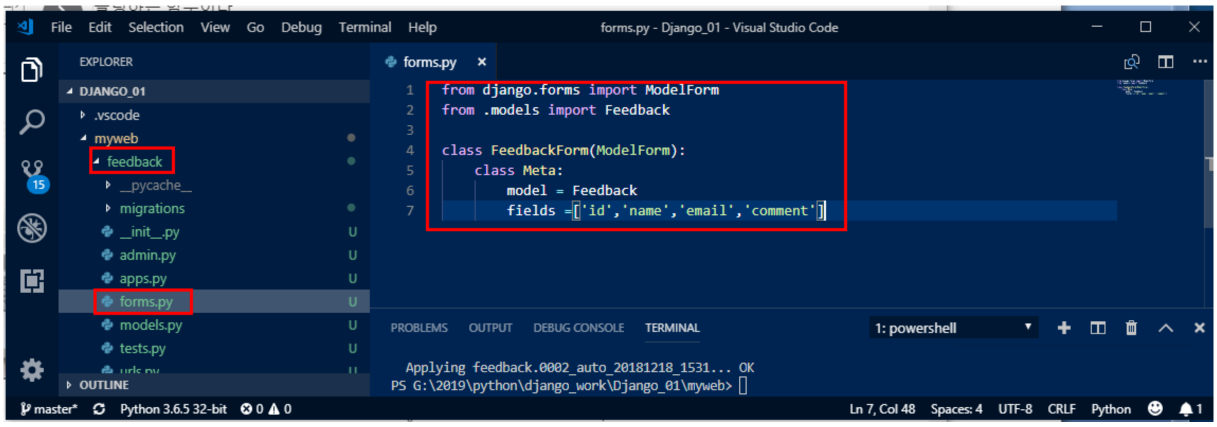
./templates/base.html 이라는 **Base** 템플릿을 만들었는데,
이 파일 안에 각 웹페이지에서 변경 혹은 삽입할 영역을 **{% block** 블럭명 **%}** 으로 지정한다. 여기서는 블럭명을 **content**로 정하여 **{% block content %}** 으로 표시



```
ROOT_URLCONF = 'myweb.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
```

```
pattern name.
[18/Dec/2018 15:57:18] "GET /feedback/add/ HTTP/1.1" 500 128530
```

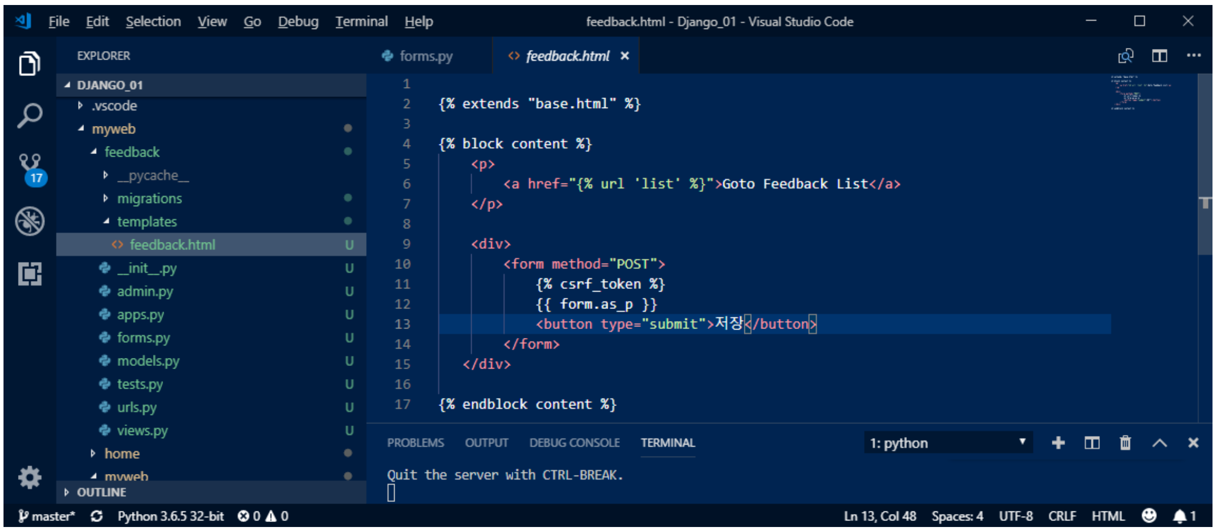
템플릿 위치 셋팅



```
from django.forms import ModelForm
from .models import Feedback

class FeedbackForm(ModelForm):
    class Meta:
        model = Feedback
        fields =['id','name','email','comment']
```

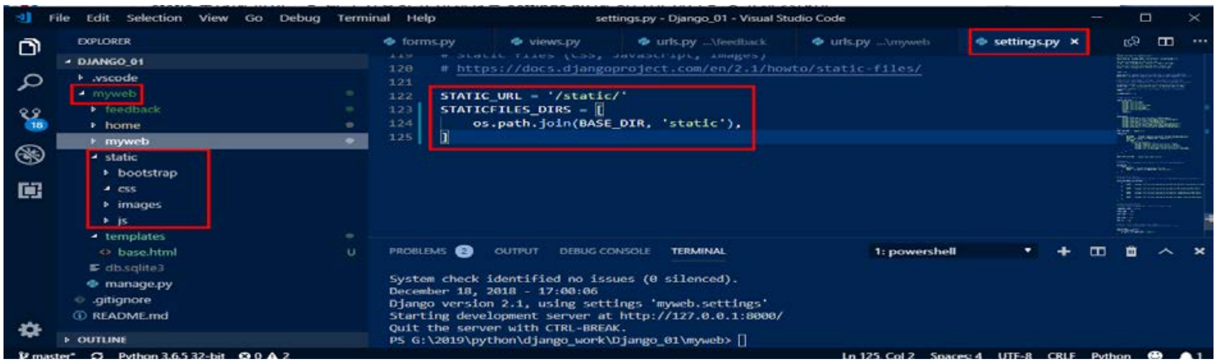
```
Applying feedback.0002_auto_20181218_1531... OK
PS G:\2019\python\django_work\Django_01\myweb>
```

CSRF (Cross Site Request Forgeries)는 웹 해킹 기법의 하나로 Django는 이를 방지하기 위한 기능을 기본적으로 제공하고 있다. Django에서 HTTP POST, PUT, DELETE을 할 경우 이 태그를 넣어 주어야 한다



```
{% extends "base.html" %}

{% block content %}
    <p>
        <a href="{% url 'list' %}">Goto Feedback List</a>
    </p>

    <div>
        <form method="POST">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">저장</button>
        </form>
    </div>

{% endblock content %}
```

```
Quit the server with CTRL-BREAK.
```



```
# https://docs.djangoproject.com/en/2.1/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static'),
]
```

```
System check identified no issues (0 silenced).
December 18, 2018 - 17:00:06
Django version 2.1, using settings 'myweb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
PS G:\2019\python\django_work\Django_01\myweb>
```