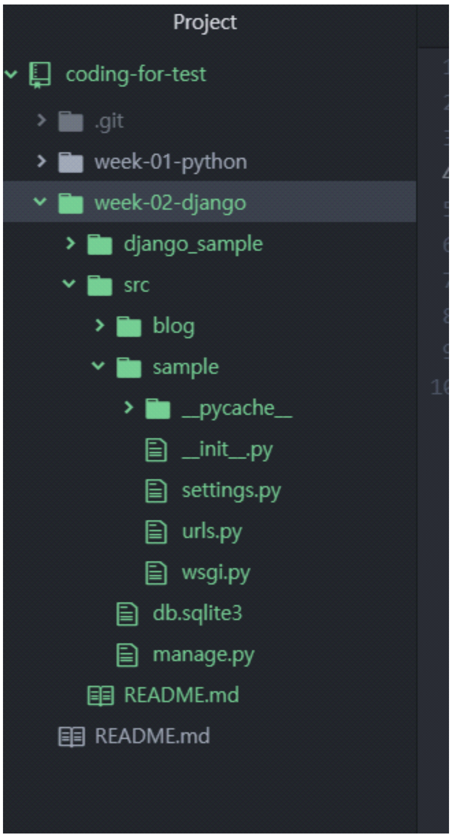
0

2019년 1월 12일 토요일 오후 4:24

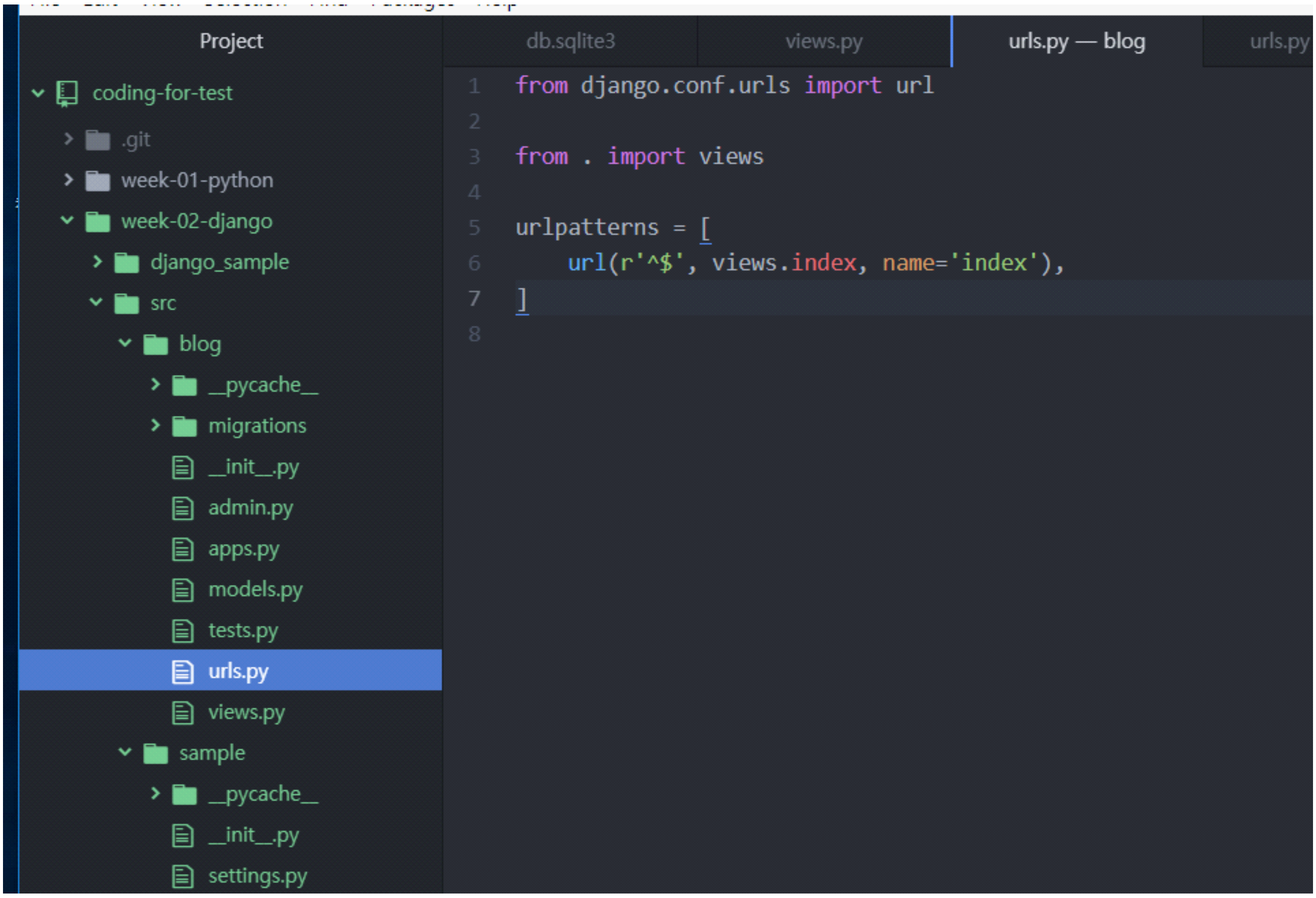
Django!
https://www.djangoproject.com/

01 runserver

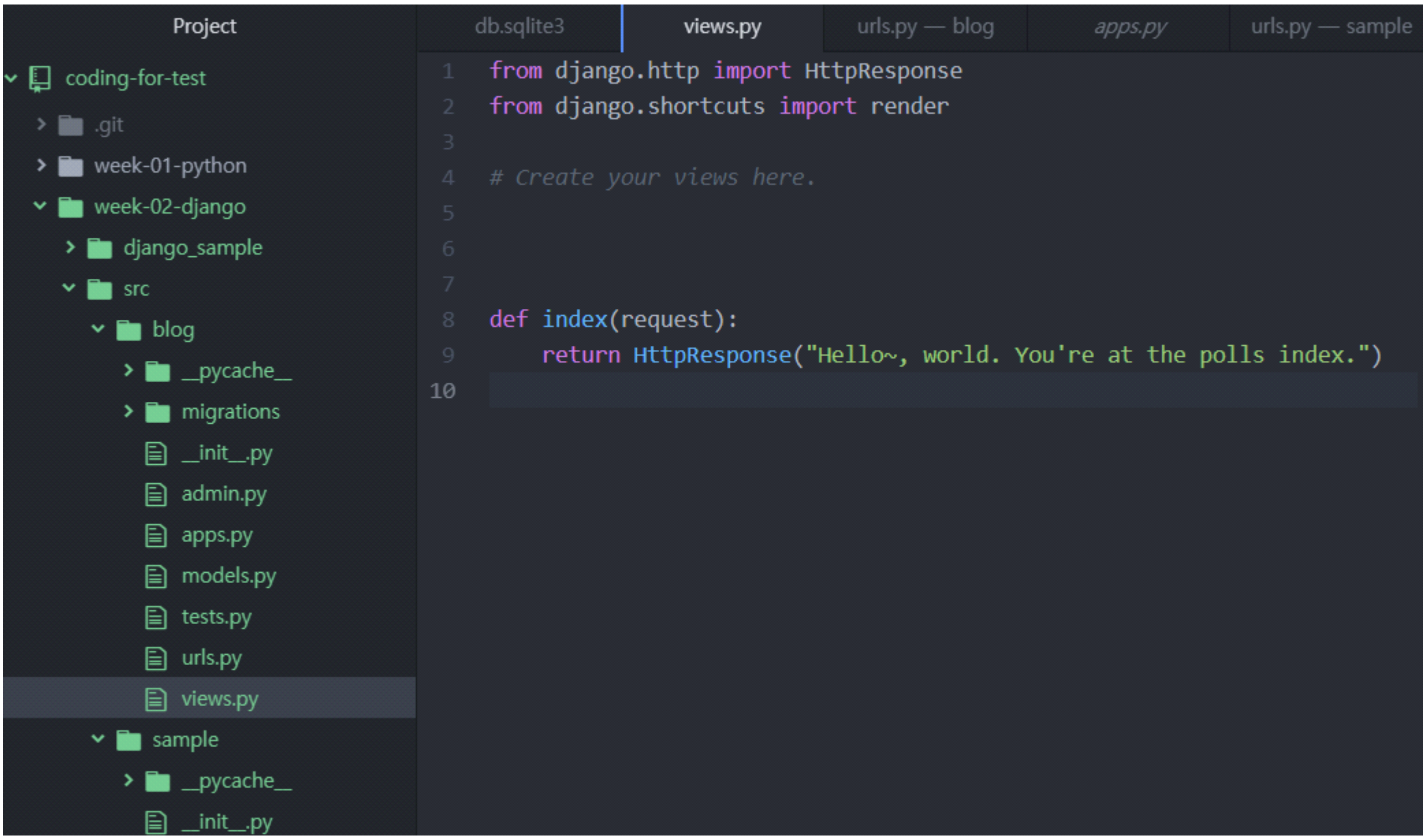
2018년 12월 23일 일요일 오후 8:18



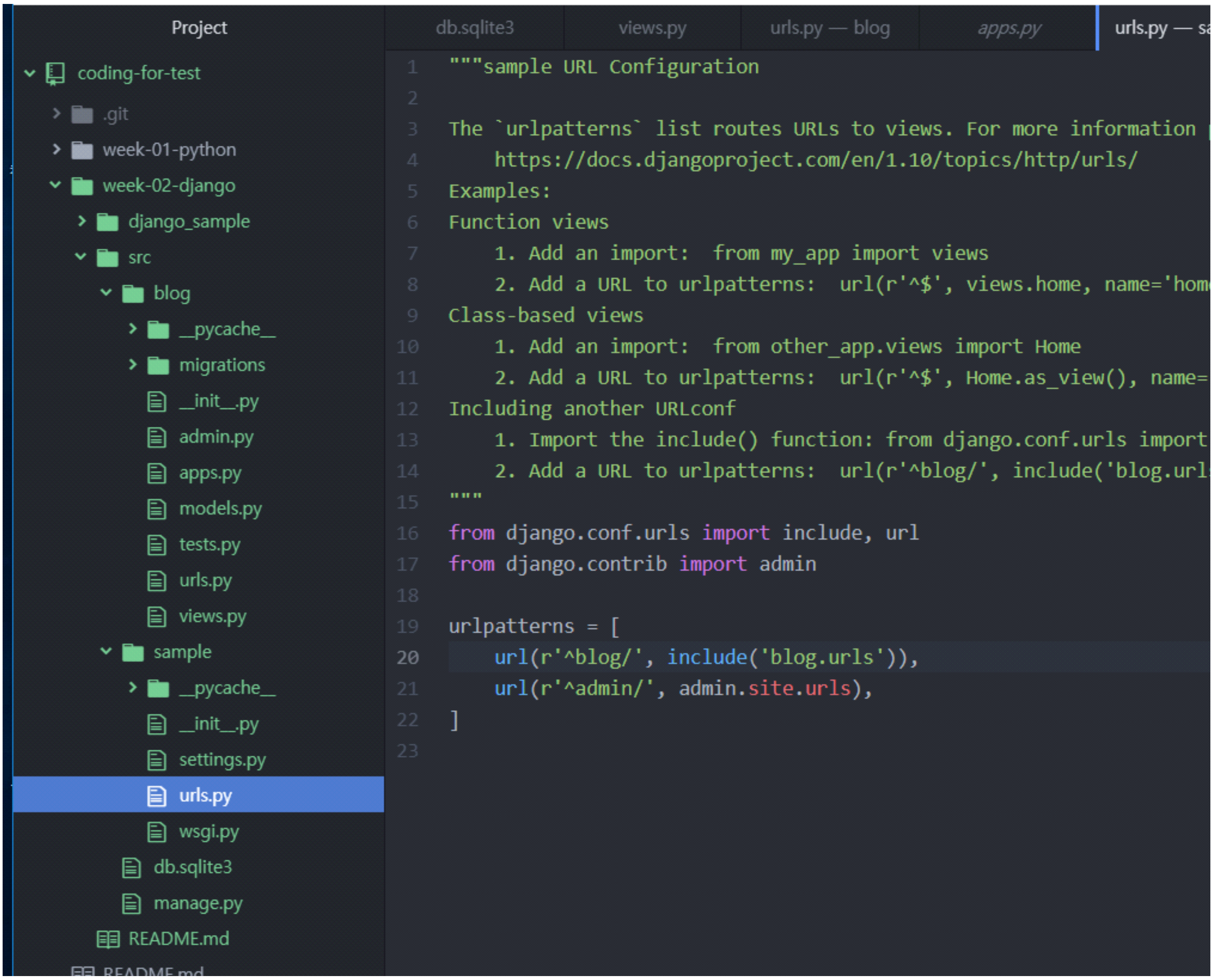
```
Project
coding-for-test
  .git
  week-01-python
  week-02-django
    django_sample
    src
      blog
      sample
        __pycache__
        __init__.py
        settings.py
        urls.py
        wsgi.py
      db.sqlite3
      manage.py
    README.md
  README.md
```



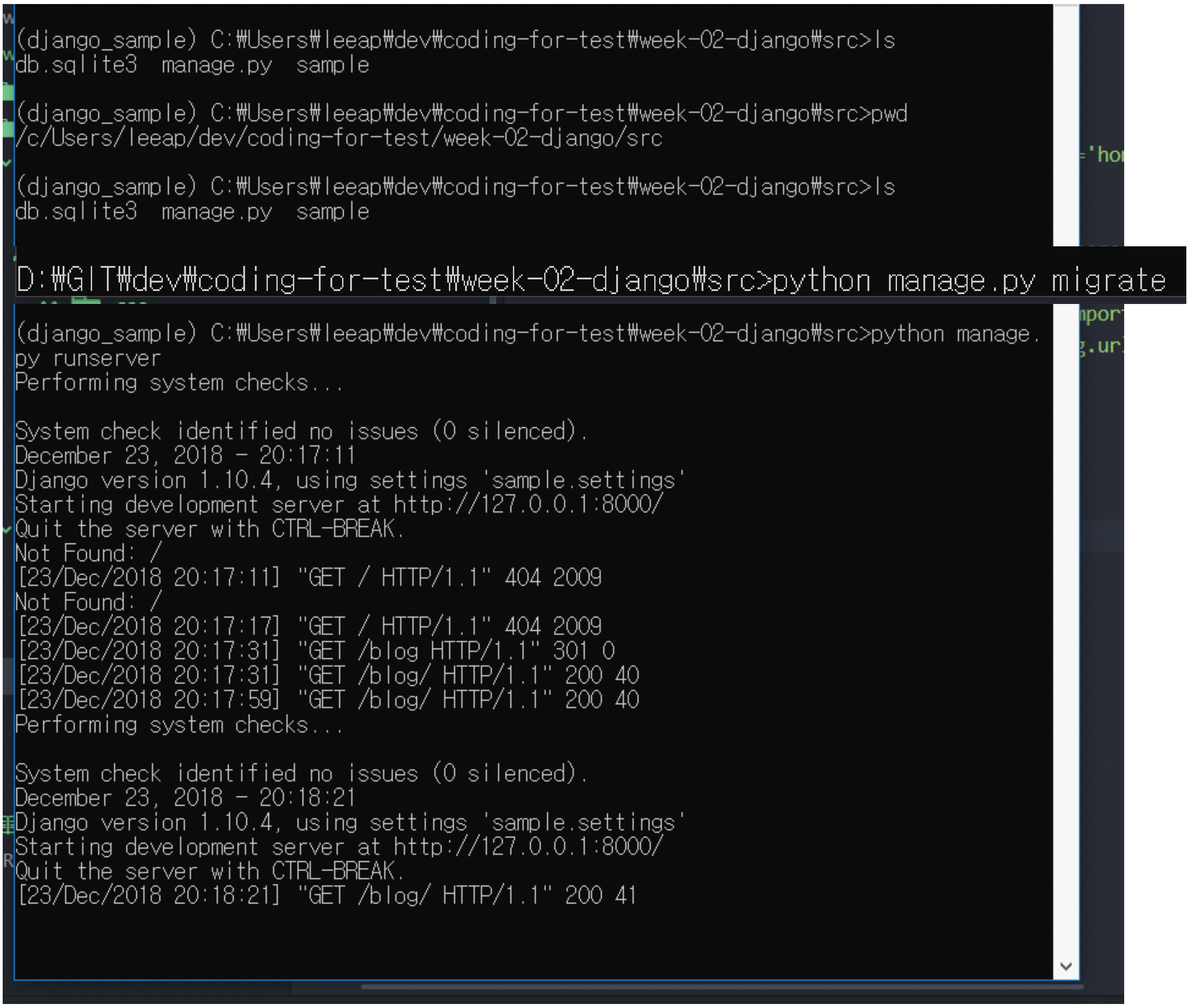
```python
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]

```



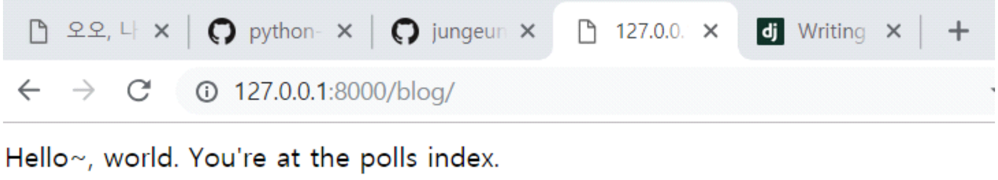
```python
from django.http import HttpResponse
from django.shortcuts import render

# Create your views here.



def index(request):
    return HttpResponse("Hello~, world. You're at the polls index.")

```



```
"""sample URL Configuration

The `urlpatterns` list routes URLs to views. For more information p
    https://docs.djangoproject.com/en/1.10/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name=
Including another URLconf
    1. Import the include() function: from django.conf.urls import
    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.url
"""
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    url(r'^blog/', include('blog.urls')),
    url(r'^admin/', admin.site.urls),
]
```



```
(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>ls
db.sqlite3  manage.py  sample

(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>pwd
/c/Users/leeap/dev/coding-for-test/week-02-django/src

(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>ls
db.sqlite3  manage.py  sample

D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py migrate

(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>python manage.
py runserver
Performing system checks...

System check identified no issues (0 silenced).
December 23, 2018 - 20:17:11
Django version 1.10.4, using settings 'sample.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Not Found: /
[23/Dec/2018 20:17:11] "GET / HTTP/1.1" 404 2009
Not Found: /
[23/Dec/2018 20:17:17] "GET / HTTP/1.1" 404 2009
[23/Dec/2018 20:17:31] "GET /blog HTTP/1.1" 301 0
[23/Dec/2018 20:17:31] "GET /blog/ HTTP/1.1" 200 40
[23/Dec/2018 20:17:59] "GET /blog/ HTTP/1.1" 200 40
Performing system checks...

System check identified no issues (0 silenced).
December 23, 2018 - 20:18:21
Django version 1.10.4, using settings 'sample.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[23/Dec/2018 20:18:21] "GET /blog/ HTTP/1.1" 200 41
```

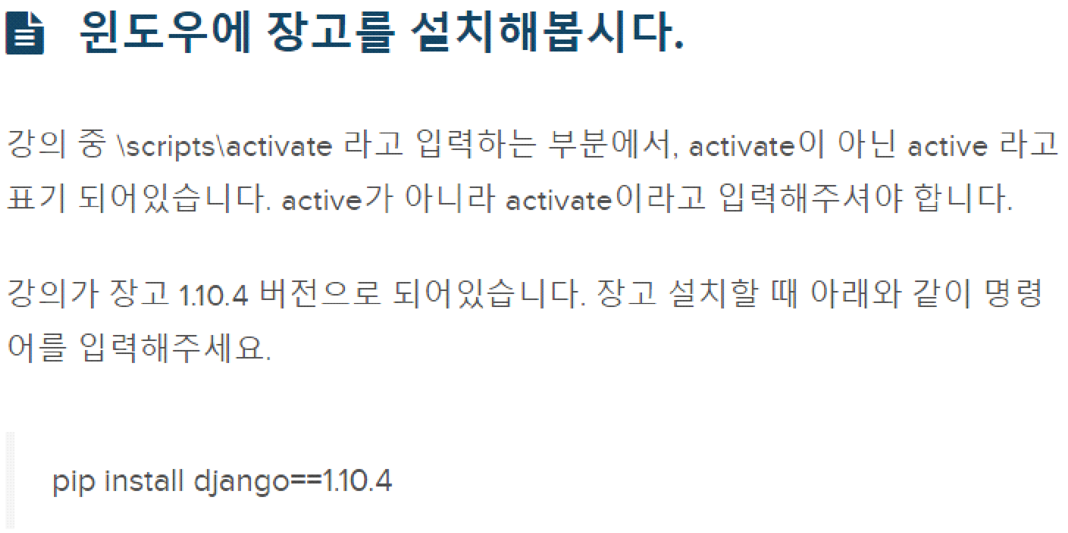


127.0.0.1:8000/blog/

Hello~, world. You're at the polls index.

01-2 독립환경실행

2019년 1월 5일 토요일 오후 4:09



윈도우에 장고를 설치해봅시다.

강의 중 \scripts\activate 라고 입력하는 부분에서, activate이 아닌 active 라고 표기 되어있습니다. active가 아니라 activate이라고 입력해주셔야 합니다.

강의가 장고 1.10.4 버전으로 되어있습니다. 장고 설치할 때 아래와 같이 명령어를 입력해주세요.

> pip install django==1.10.4

독립환경 설정 (Virtual Environment)

1. Virtualenv 라이브러리 설치하기: pip install Virtualenv
2. 프로젝트 폴더 생성: mkdir {{ python_project }}
3. 프로젝트 폴더 들어가기: cd {{ python_project }}
4. 독립환경 만들기: python -m venv {{ my_python }}
5. 독립환경 실행 시키기: {{ my_python }}₩scripts₩activate

장고 설치

1. 장고 설치하기: pip install django==1.10.4
2. 새로운 프로젝트 만들기: django-admin startproject {{firstsite}}
3. 기본 데이터베이스 설정하기: python manage.py migrate
4. 로컬 서버 시작하기: python manage.py runserver

- pip 공식문서: https://pip.pypa.io/en/stable/user_guide/
- 장고 공식문서: https://docs.djangoproject.com/en/1.10/

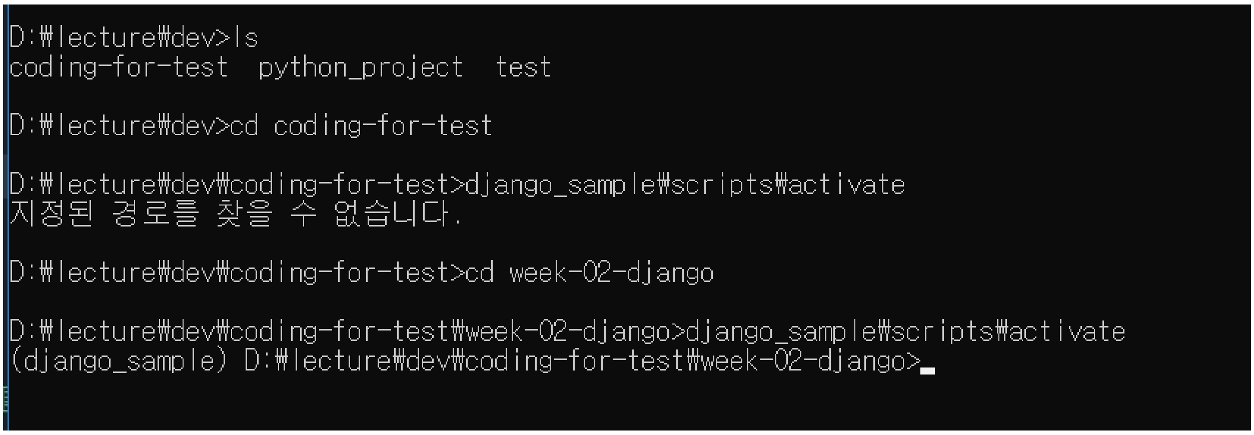
https://docs.djangoproject.com/en/1.10/intro/tutorial01/

pip 주요 명령어

- pip install : pip로 파이썬 패키지(라이브러리) 설치하기
- pip uninstall : pip로 파이썬 패키지(라이브러리) 삭제하기
- pip freeze : pip로 설치한 파이썬 패키지(라이브러리) 목록 표시
- pip freeze > requirements.txt : 위의 목록을 requirements.txt 라는 파일로 만들기
- pip install -r requirements.txt : requirements.txt 안의 패키지 전체 설치하기

cmd

```
(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>python manage.py runserver
ctrl+C
$ python manage.py migrate
(django_sample) C:₩Users₩leeap₩dev₩coding-for-test₩week-02-django₩src>python manage.py startapp blog
```

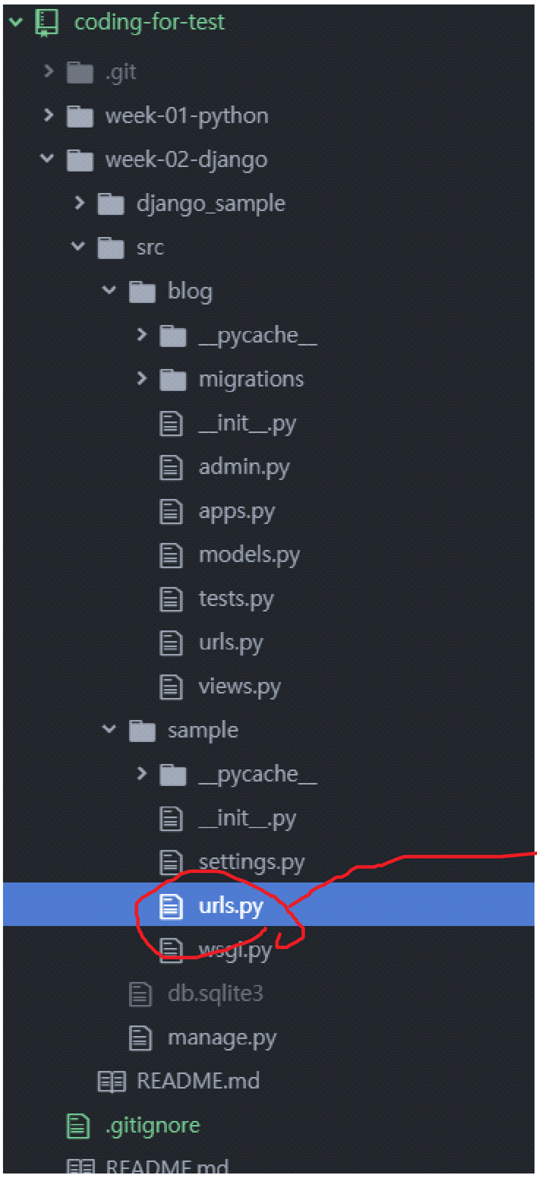


```
D:₩lecture₩dev>ls
coding-for-test  python_project  test

D:₩lecture₩dev>cd coding-for-test

D:₩lecture₩dev₩coding-for-test>django_sample₩scripts₩activate
지정된 경로를 찾을 수 없습니다.

D:₩lecture₩dev₩coding-for-test>cd week-02-django

D:₩lecture₩dev₩coding-for-test₩week-02-django>django_sample₩scripts₩activate
(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django>
```

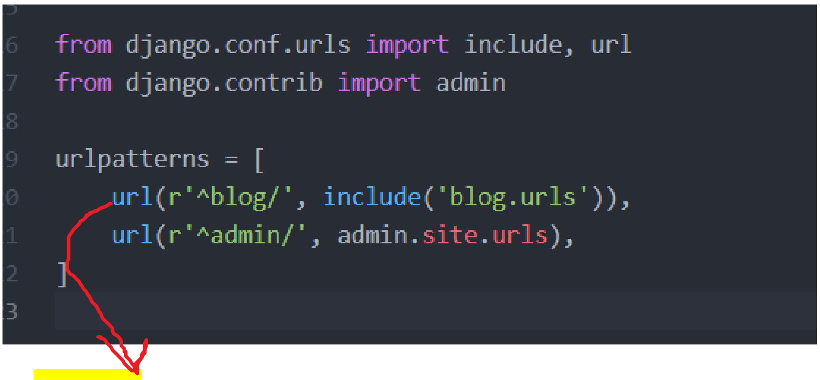
02 View & Url

2019년 1월 5일 토요일 오후 3:45

내 서비스 페이지마다 이름을 붙이자



```
coding-for-test
  .git
  week-01-python
  week-02-django
    django_sample
    src
      blog
        __pycache__
        migrations
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
        views.py
      sample
        __pycache__
        __init__.py
        settings.py
        urls.py
        wsgi.py
      db.sqlite3
      manage.py
    README.md
  .gitignore
  README.md
```



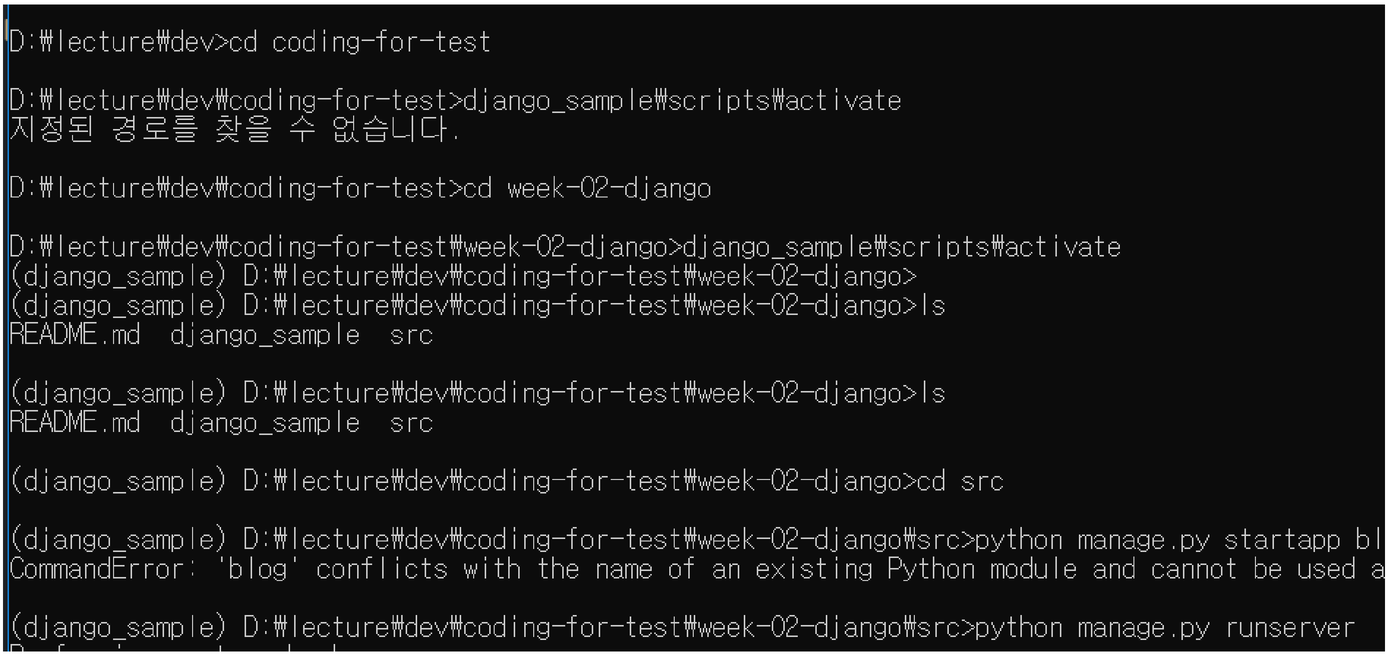
```python
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    url(r'^blog/', include('blog.urls')),
    url(r'^admin/', admin.site.urls),
]
```

주소역할

Blog폴더에 있는 urls.py걸 다 가져와서

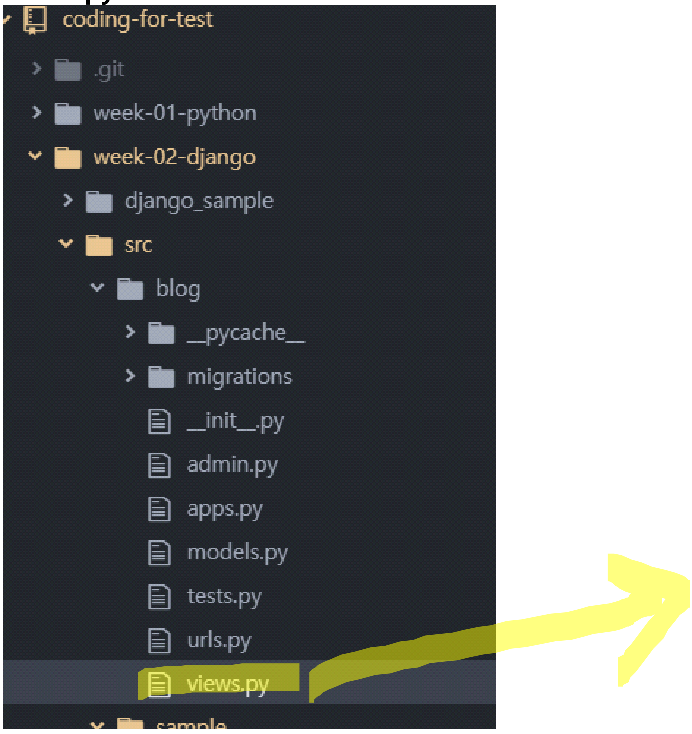
Blog라는 이름을 붙여라

Run server!



```
D:₩lecture₩dev>cd coding-for-test

D:₩lecture₩dev₩coding-for-test>django_sample₩scripts₩activate
지정된 경로를 찾을 수 없습니다.

D:₩lecture₩dev₩coding-for-test>cd week-02-django

D:₩lecture₩dev₩coding-for-test₩week-02-django>django_sample₩scripts₩activate
(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django>
(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django>ls
README.md  django_sample  src

(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django>ls
README.md  django_sample  src

(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django>cd src

(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django₩src>python manage.py startapp bl
CommandError: 'blog' conflicts with the name of an existing Python module and cannot be used a

(django_sample) D:₩lecture₩dev₩coding-for-test₩week-02-django₩src>python manage.py runserver
```

Views.py

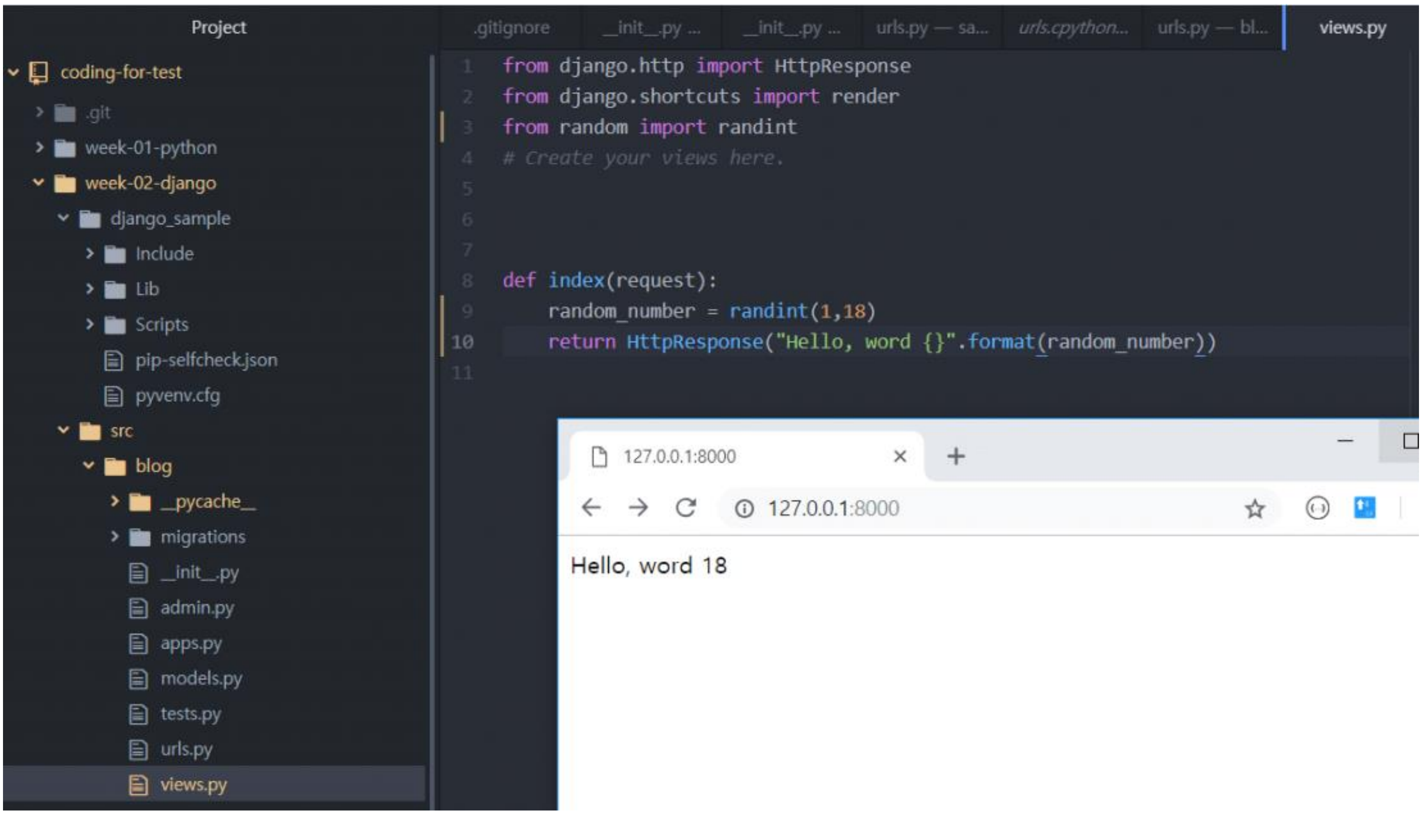


```
coding-for-test
  .git
  week-01-python
  week-02-django
    django_sample
    src
      blog
        __pycache__
        migrations
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
        views.py
      sample
```

View 역할

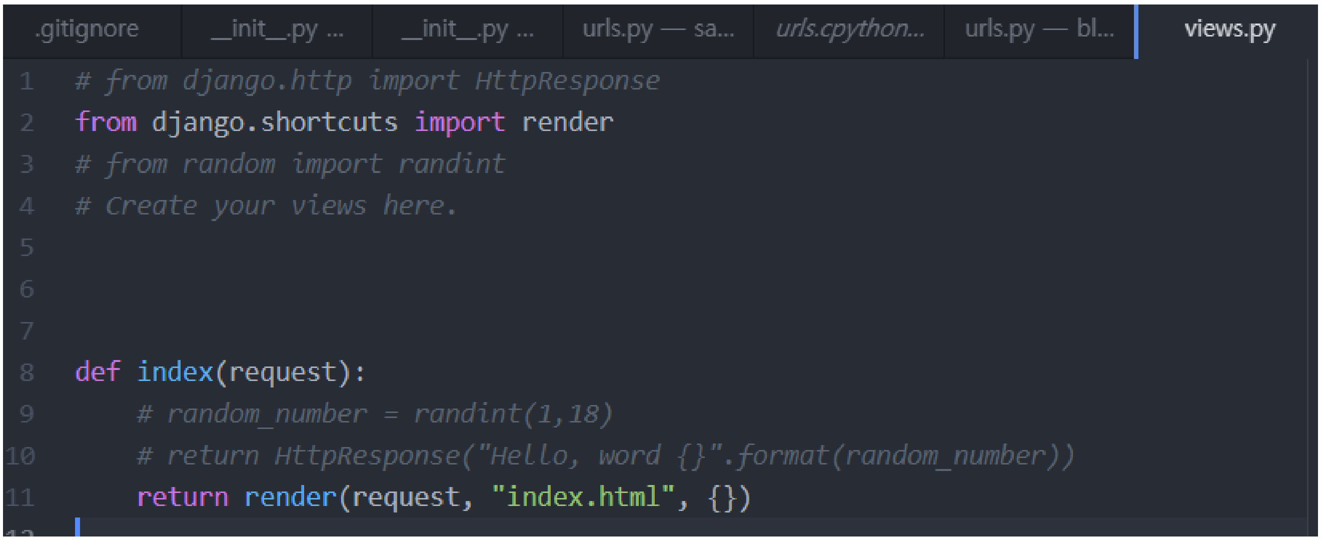
03. Templates, view 연결

2019년 1월 5일 토요일　　오후 4:04



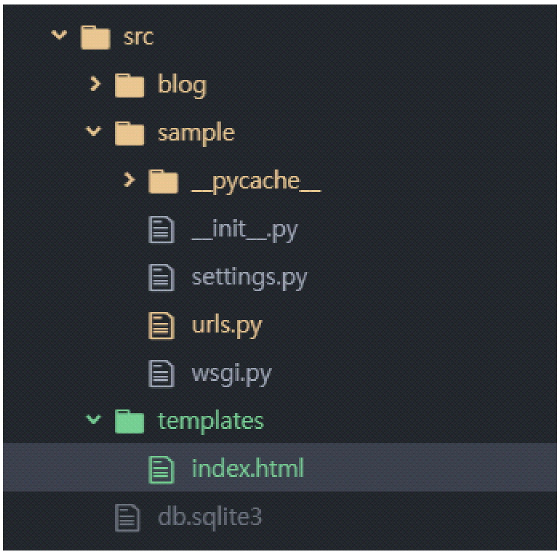
```python
from django.http import HttpResponse
from django.shortcuts import render
from random import randint
# Create your views here.



def index(request):
    random_number = randint(1,18)
    return HttpResponse("Hello, word {}".format(random_number))
```

```
Hello, word 18
```

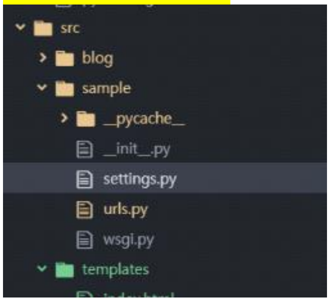


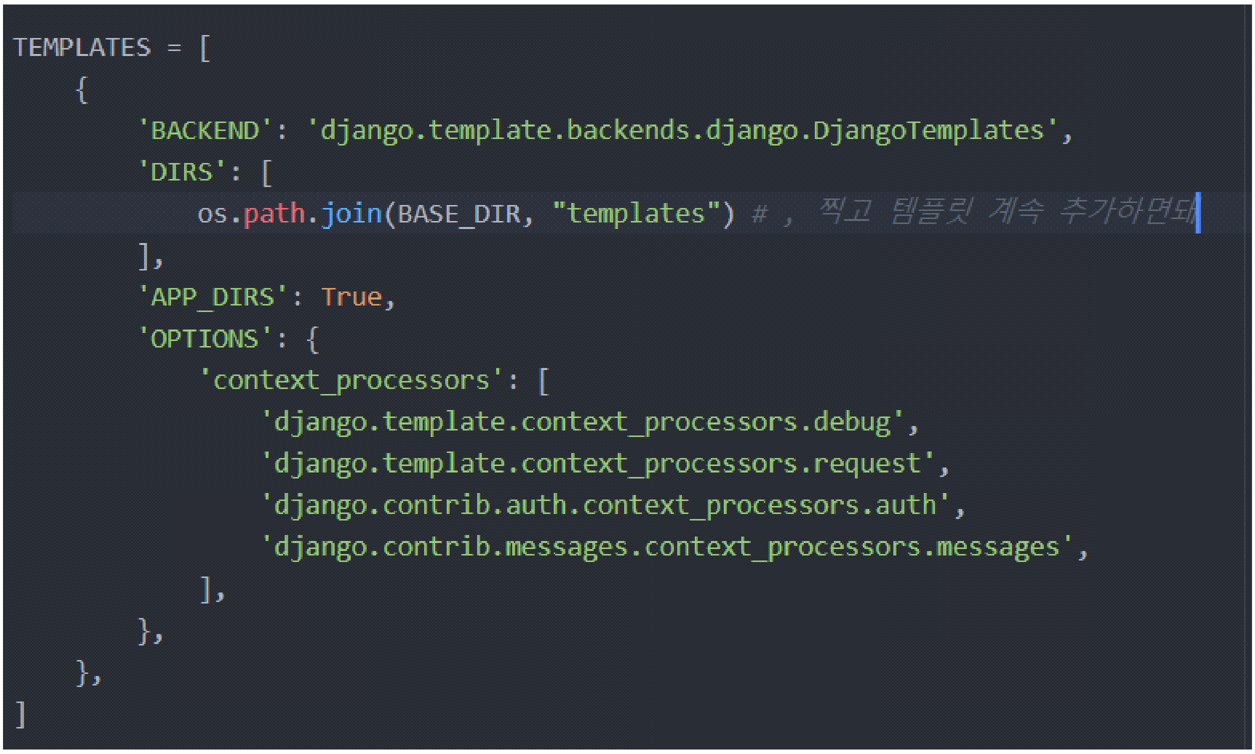
```python
# from django.http import HttpResponse
from django.shortcuts import render
# from random import randint
# Create your views here.



def index(request):
    # random_number = randint(1,18)
    # return HttpResponse("Hello, word {}".format(random_number))
    return render(request, "index.html", {})
```

Src₩templates₩index.html 만들기

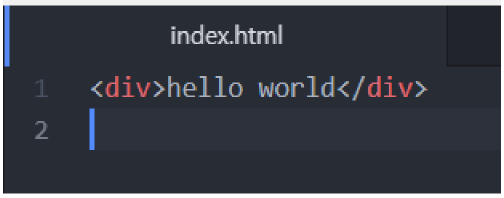


템플릿 경로 설정

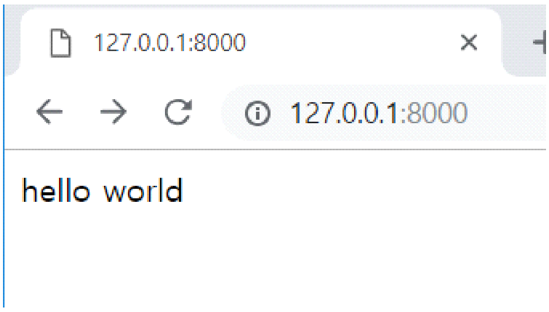




```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            os.path.join(BASE_DIR, "templates") # , 찍고 템플릿 계속 추가하면돼
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

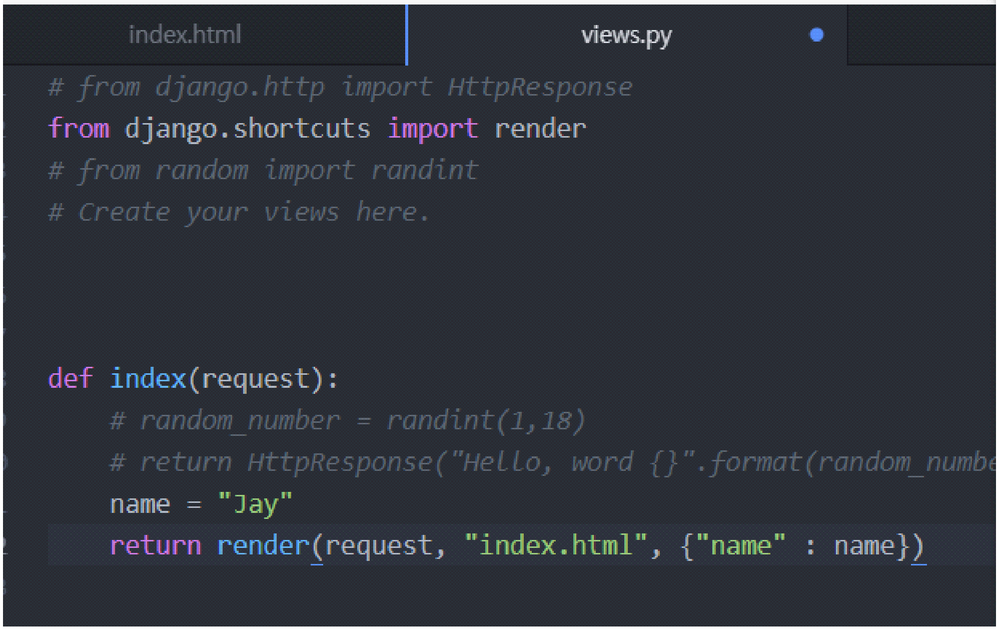


```
<div>hello world</div>
```

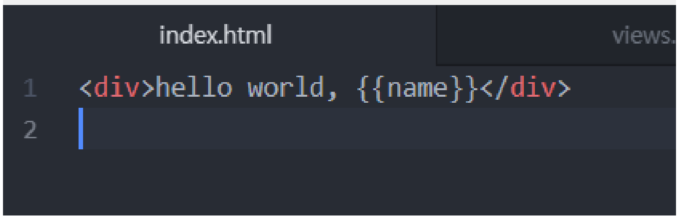


```
hello world
```

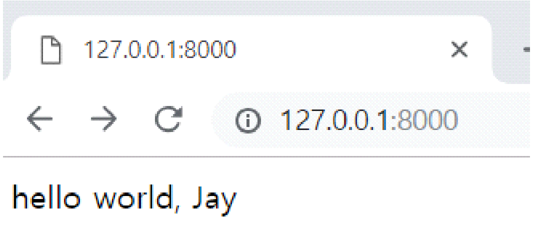
View.py를 통해 값을 넘겨 html에서 값 받기



```python
# from django.http import HttpResponse
from django.shortcuts import render
# from random import randint
# Create your views here.



def index(request):
    # random_number = randint(1,18)
    # return HttpResponse("Hello, word {}".format(random_numbe
    name = "Jay"
    return render(request, "index.html", {"name" : name})
```



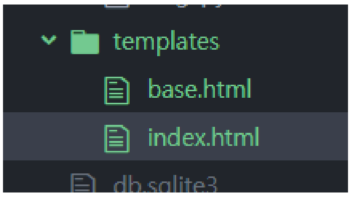
```html
<div>hello world, {{name}}</div>

```

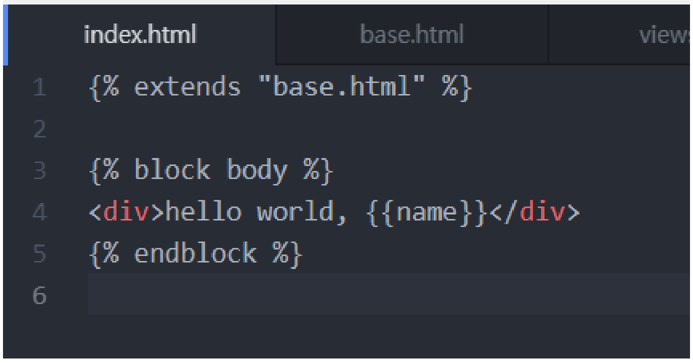


```
hello world, Jay
```

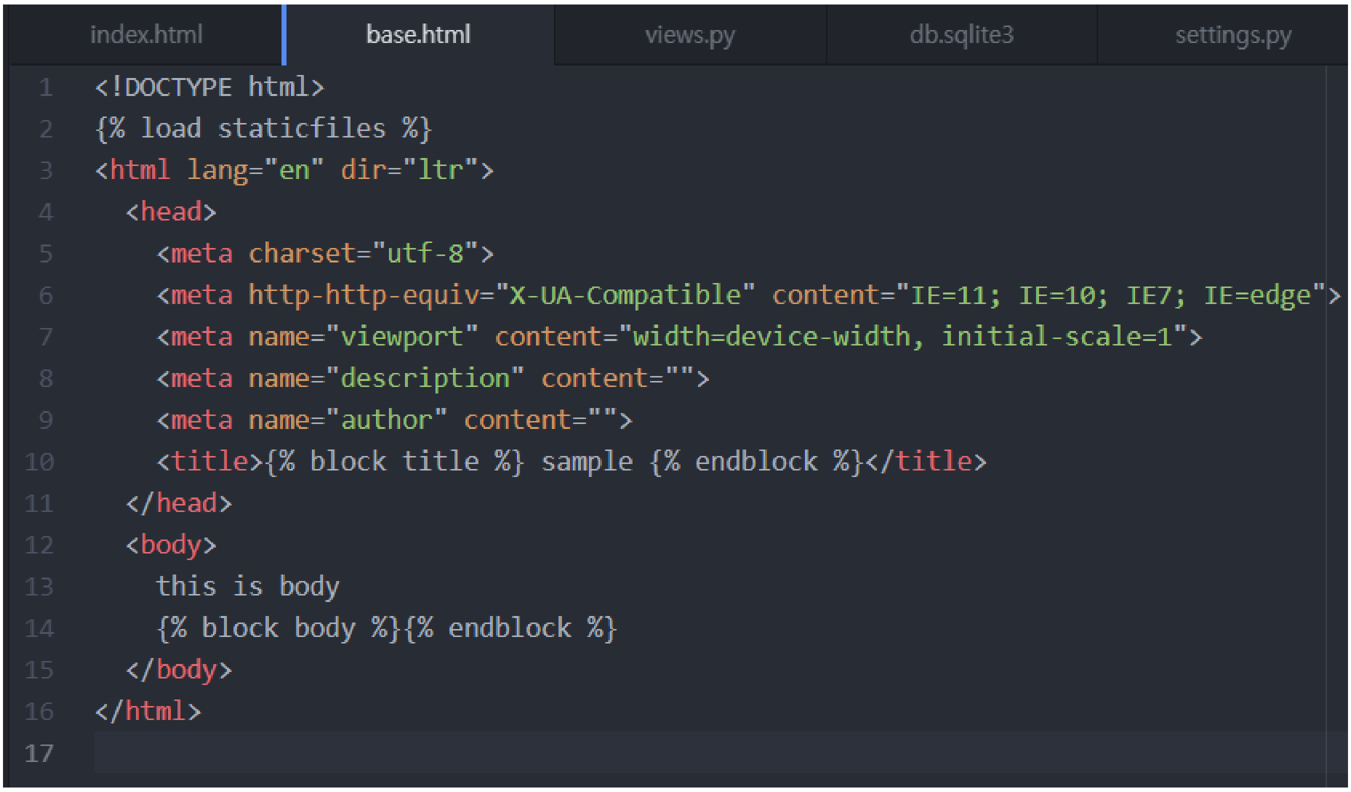
04. extends html

2019년 1월 5일 토요일　　오후 5:07

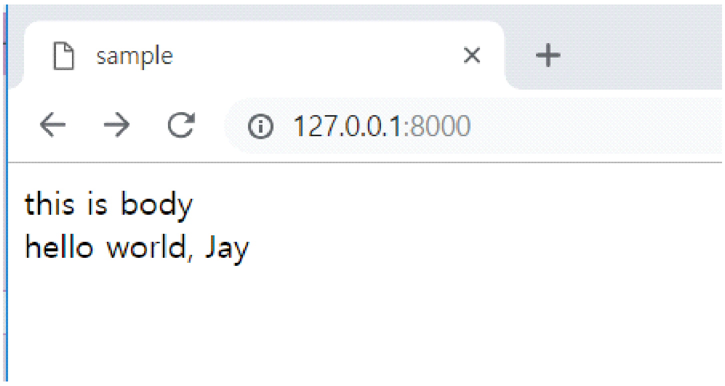




```
{% extends "base.html" %}

{% block body %}
<div>hello world, {{name}}</div>
{% endblock %}

```



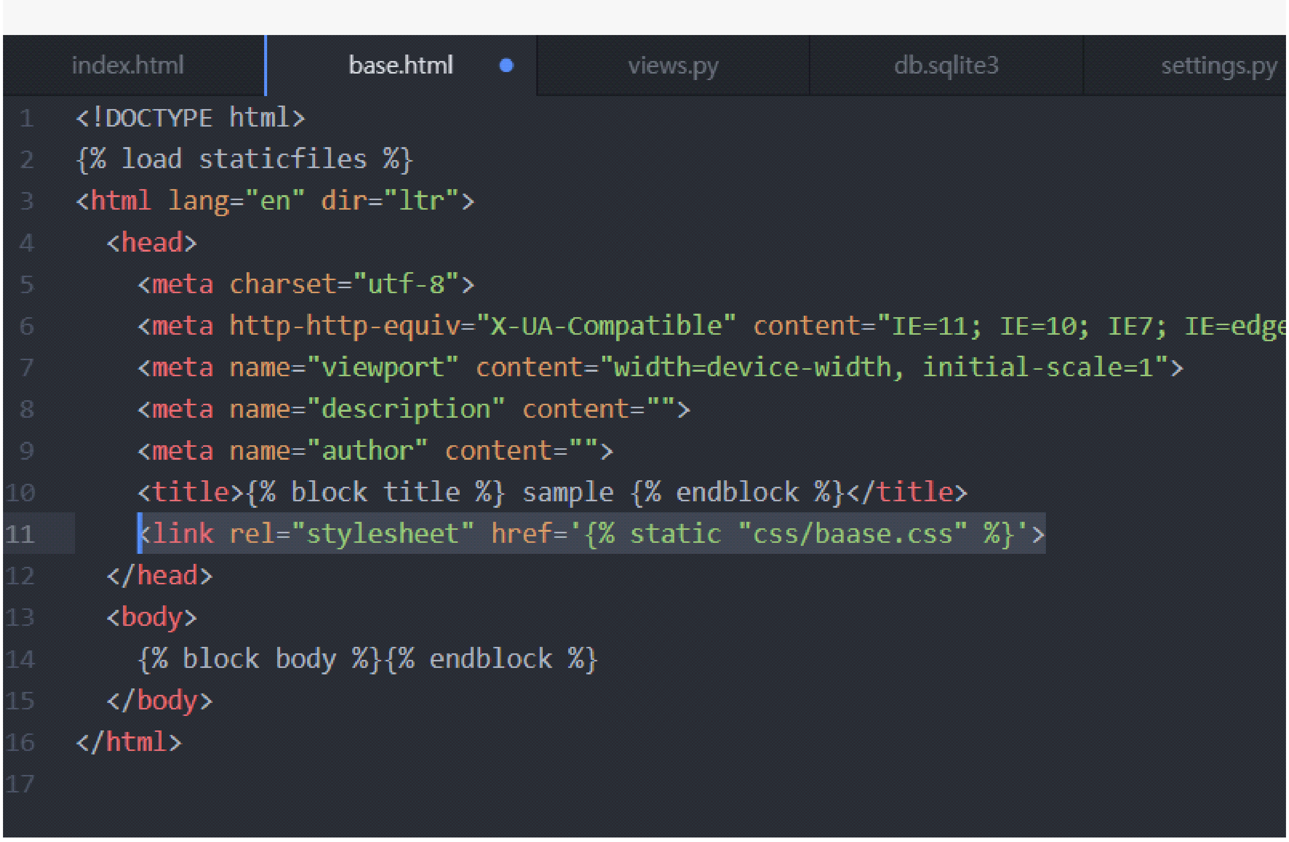
```
<!DOCTYPE html>
{% load staticfiles %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta http-http-equiv="X-UA-Compatible" content="IE=11; IE=10; IE7; IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>{% block title %} sample {% endblock %}</title>
  </head>
  <body>
    this is body
    {% block body %}{% endblock %}
  </body>
</html>

```



this is body
hello world, Jay

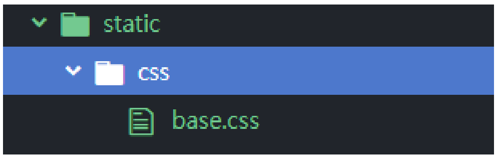
05. css

2019년 1월 6일 일요일 오후 10:30



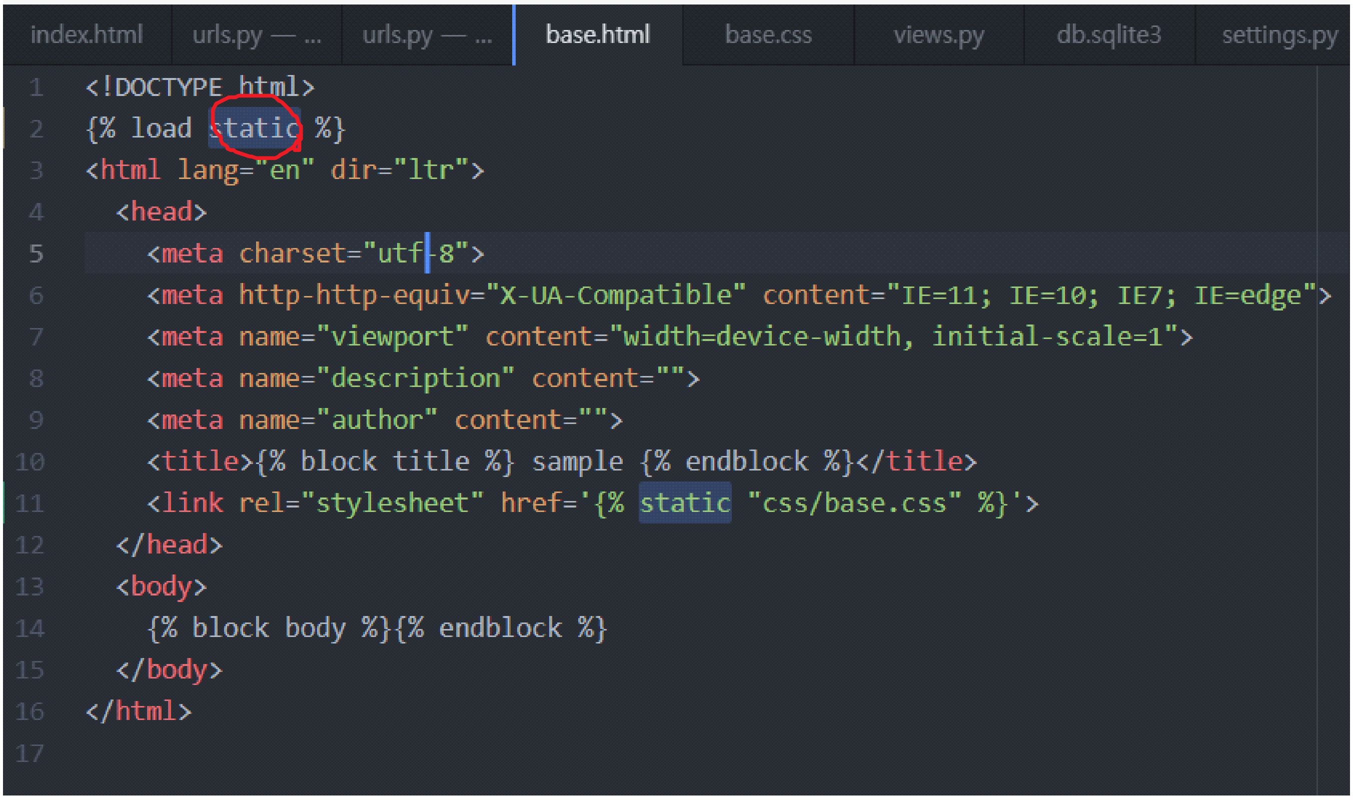
```
<!DOCTYPE html>
{% load staticfiles %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta http-http-equiv="X-UA-Compatible" content="IE=11; IE=10; IE7; IE=edge
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>{% block title %} sample {% endblock %}</title>
    <link rel="stylesheet" href='{% static "css/baase.css" %}'>
  </head>
  <body>
    {% block body %}{% endblock %}
  </body>
</html>

```

해당 경로 파일 만들어 주기

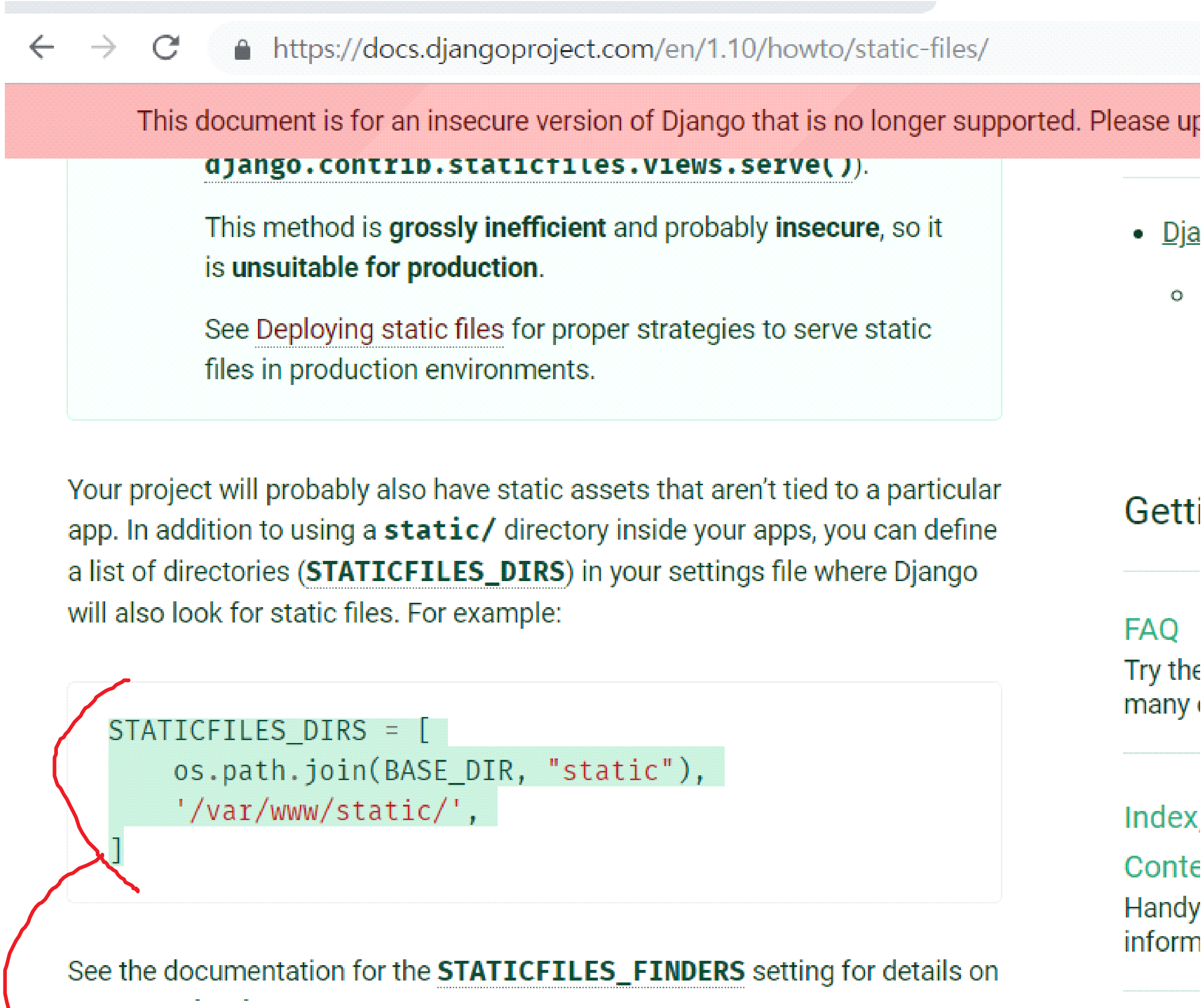


Settings.py

https://docs.djangoproject.com/en/1.10/howto/static-files/



```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta http-http-equiv="X-UA-Compatible" content="IE=11; IE=10; IE7; IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>{% block title %} sample {% endblock %}</title>
    <link rel="stylesheet" href='{% static "css/base.css" %}'>
  </head>
  <body>
    {% block body %}{% endblock %}
  </body>
</html>
```



This document is for an insecure version of Django that is no longer supported. Please up

django.contrib.staticfiles.views.serve()).

This method is **grossly inefficient** and probably **insecure**, so it is **unsuitable for production**.

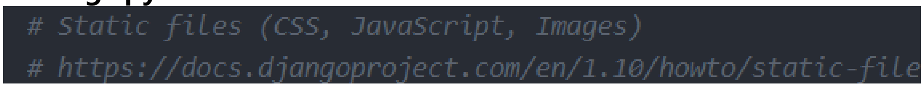
See Deploying static files for proper strategies to serve static files in production environments.

Your project will probably also have static assets that aren't tied to a particular app. In addition to using a `static/` directory inside your apps, you can define a list of directories (`STATICFILES_DIRS`) in your settings file where Django will also look for static files. For example:

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

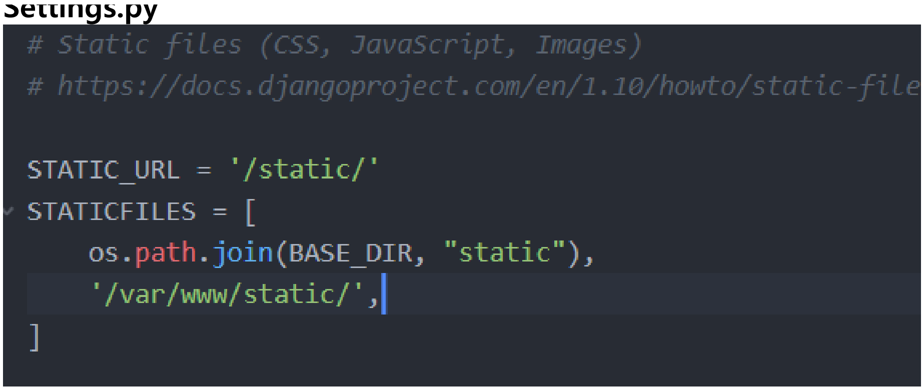
See the documentation for the `STATICFILES_FINDERS` setting for details on

Settings.py



```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.10/howto/static-file
```

Settings.py



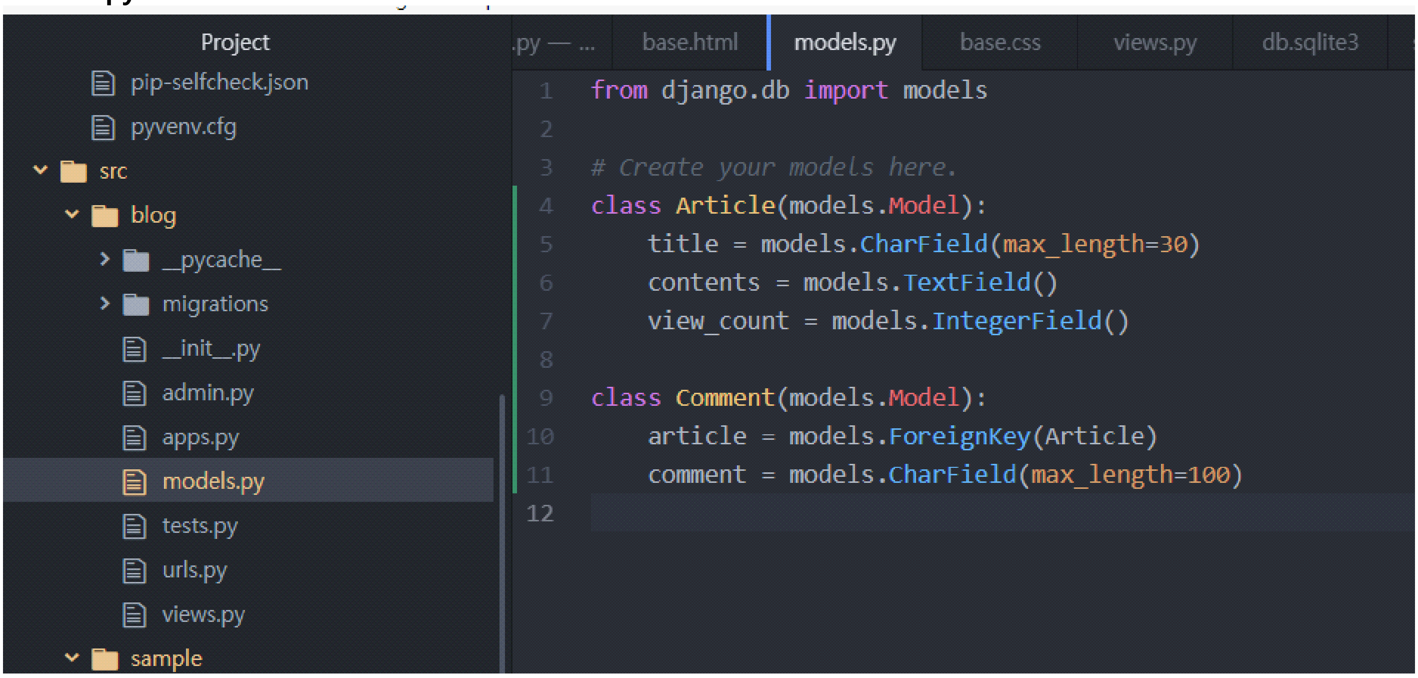
```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.10/howto/static-file

STATIC_URL = '/static/'
STATICFILES = [
    os.path.join(BASE_DIR, "static"),
    '/var/www/static/',
]
```

06. Models.py, 데이터베이스/SQL

2019년 1월 6일 일요일 오후 10:51

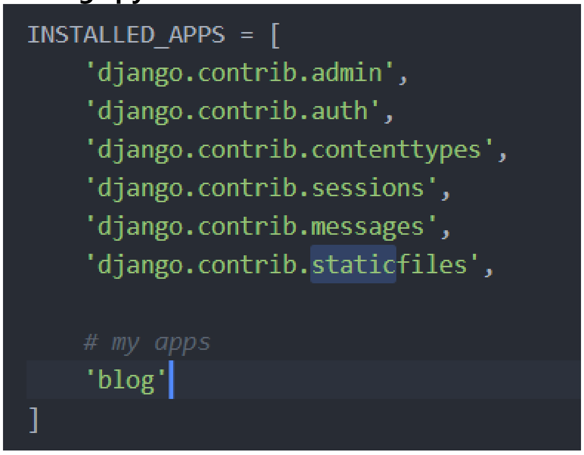
https://docs.djangoproject.com/en/2.1/topics/db/models/

Models.py



```python
from django.db import models

# Create your models here.
class Article(models.Model):
    title = models.CharField(max_length=30)
    contents = models.TextField()
    view_count = models.IntegerField()

class Comment(models.Model):
    article = models.ForeignKey(Article)
    comment = models.CharField(max_length=100)
```

Settings.py



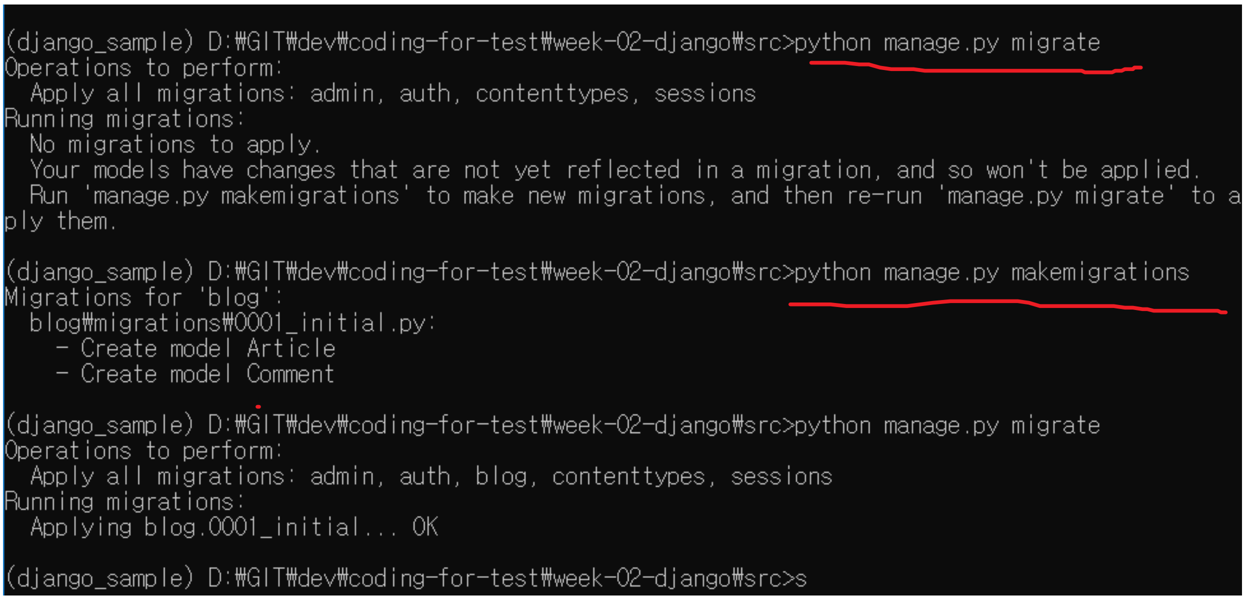
```python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # my apps
    'blog'
]
```

07. migrate

2019년 1월 6일 일요일　　오후 11:19

$ python manage.py migrate
--> error

$ python manage.py makemigrations



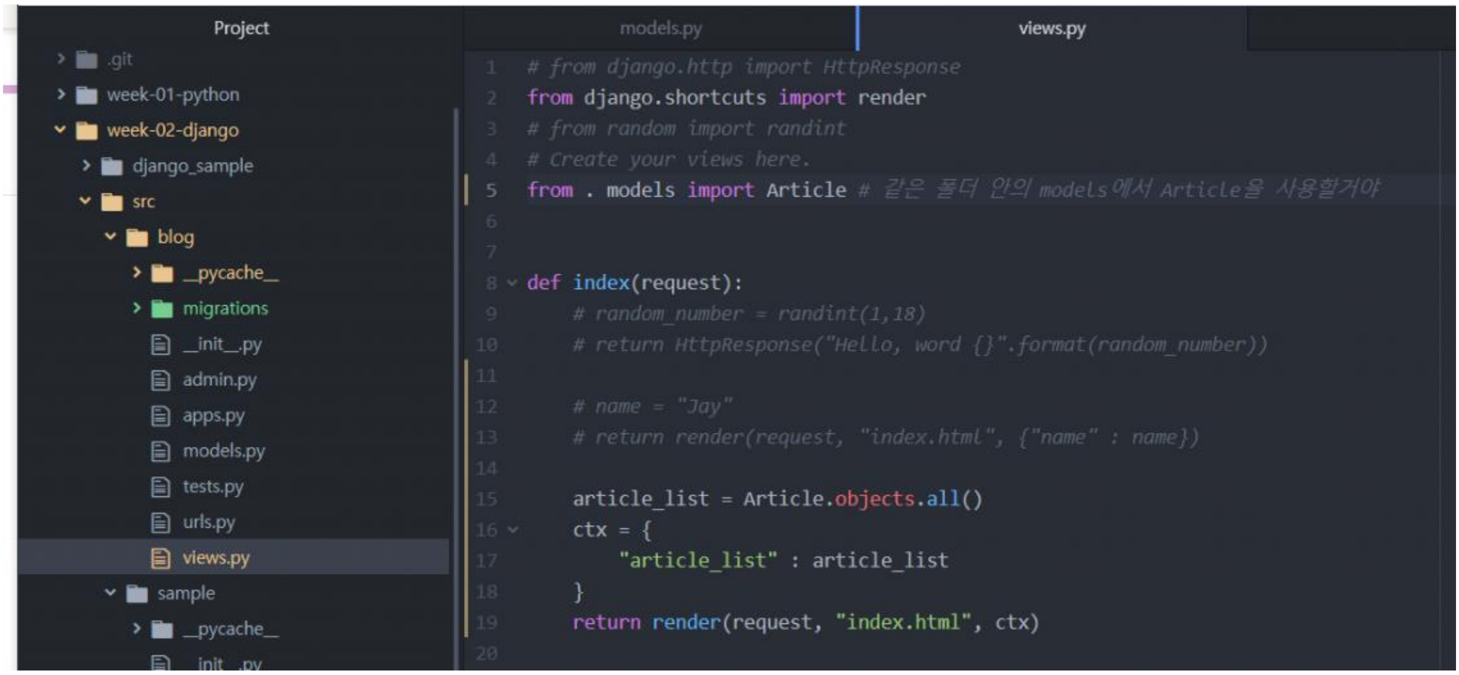
```
(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
  Your models have changes that are not yet reflected in a migration, and so won't be applied.
  Run 'manage.py makemigrations' to make new migrations, and then re-run 'manage.py migrate' to a
ply them.

(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py makemigrations
Migrations for 'blog':
  blog₩migrations₩0001_initial.py:
    - Create model Article
    - Create model Comment

(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, sessions
Running migrations:
  Applying blog.0001_initial... OK

(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>s
```

08. models database

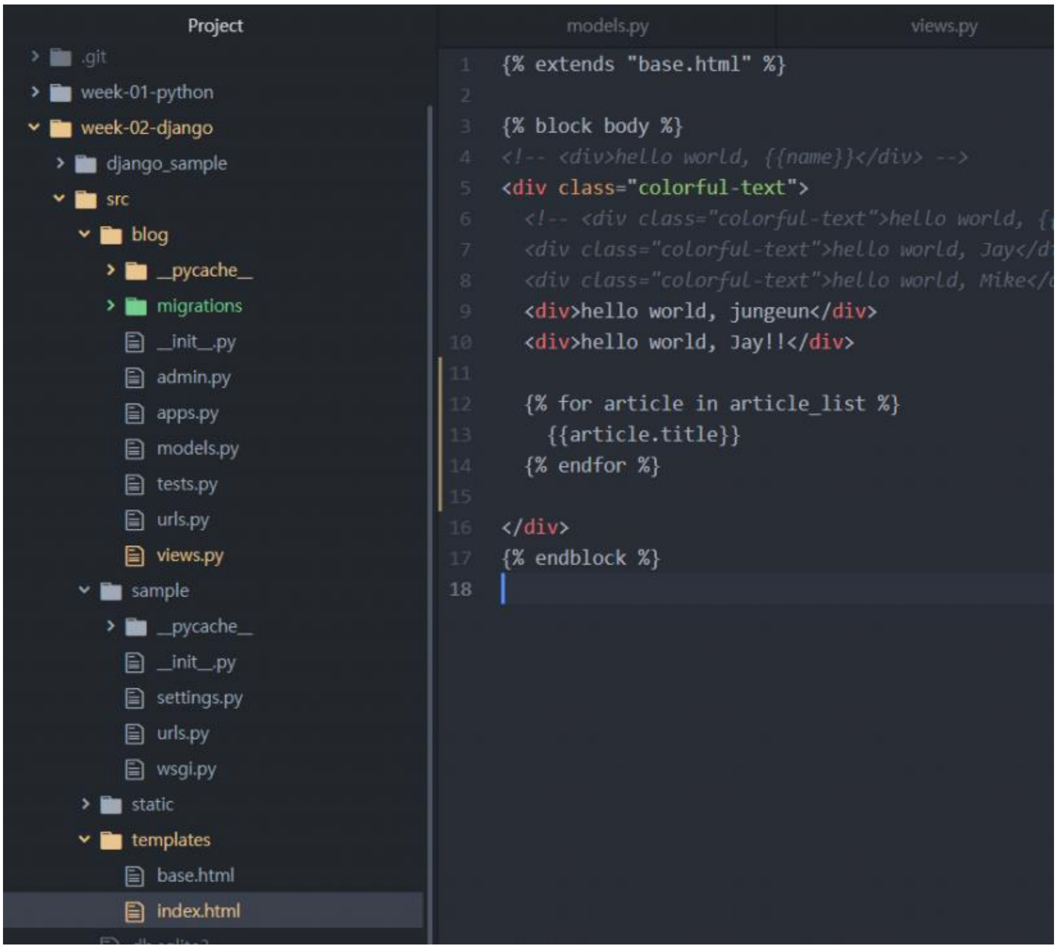
2019년 1월 12일 토요일 오후 3:23

Views.py에서 데이터 넣기

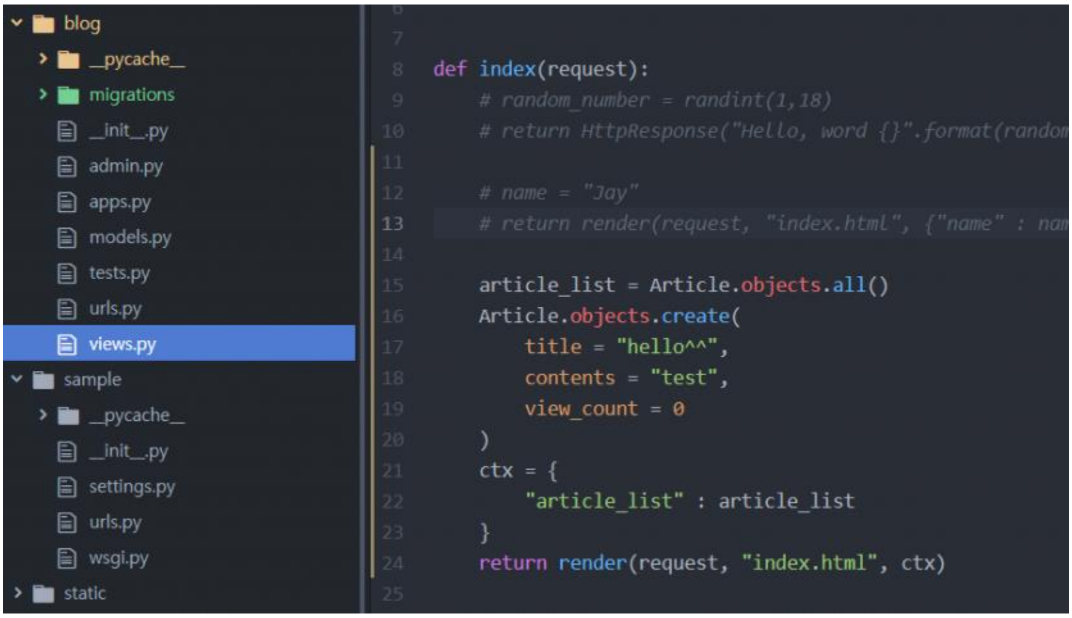
2019년 1월 12일 토요일 오후 3:48



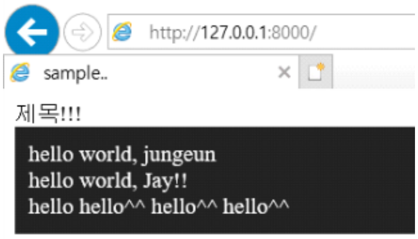
```python
# from django.http import HttpResponse
from django.shortcuts import render
# from random import randint
# Create your views here.
from . models import Article # 같은 폴더 안의 models에서 Article을 사용할거야


def index(request):
    # random_number = randint(1,18)
    # return HttpResponse("Hello, word {}".format(random_number))

    # name = "Jay"
    # return render(request, "index.html", {"name" : name})

    article_list = Article.objects.all()
    ctx = {
        "article_list" : article_list
    }
    return render(request, "index.html", ctx)

```



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

{% block body %}
<!-- <div>hello world, {{name}}</div> -->
<div class="colorful-text">
  <!-- <div class="colorful-text">hello world, {
  <div class="colorful-text">hello world, Jay</d
  <div class="colorful-text">hello world, Mike</
  <div>hello world, jungeun</div>
  <div>hello world, Jay!!</div>

  {% for article in article_list %}
    {{article.title}}
  {% endfor %}

</div>
{% endblock %}

```



```python

def index(request):
    # random_number = randint(1,18)
    # return HttpResponse("Hello, word {}".format(randon

    # name = "Jay"
    # return render(request, "index.html", {"name" : nam

    article_list = Article.objects.all()
    Article.objects.create(
        title = "hello^^",
        contents = "test",
        view_count = 0
    )
    ctx = {
        "article_list" : article_list
    }
    return render(request, "index.html", ctx)

```



제목!!!

```
hello world, jungeun
hello world, Jay!!
hello hello^^ hello^^ hello^^
```

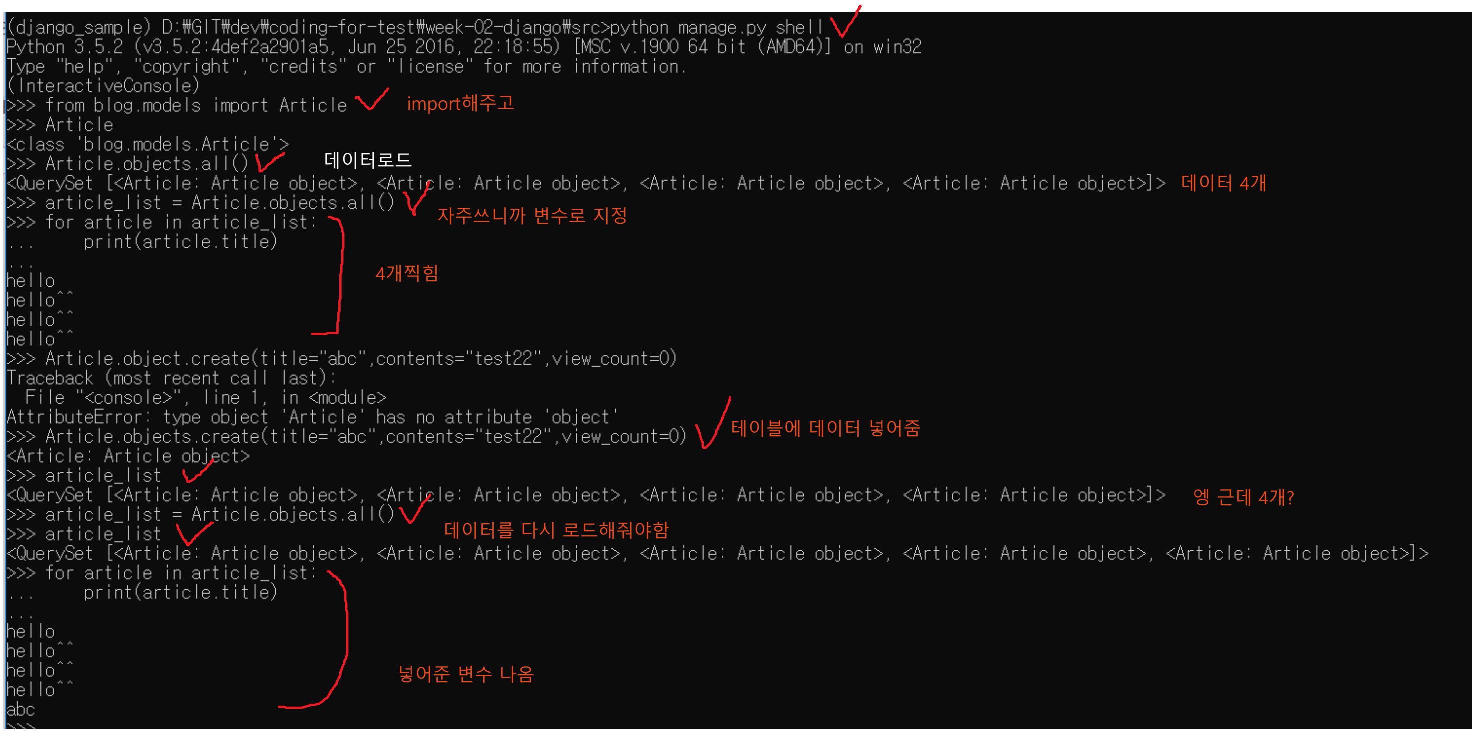
근데 이건 좋은 방법이 ㄴㄴ 왜냐면 계속 페이지 refresh할때마다 데이터가 계속 들어감

Cmd에서 데이터 넣기

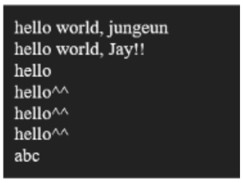
2019년 1월 12일 토요일 오후 3:48

명령프롬프트 :

python manage.py shell



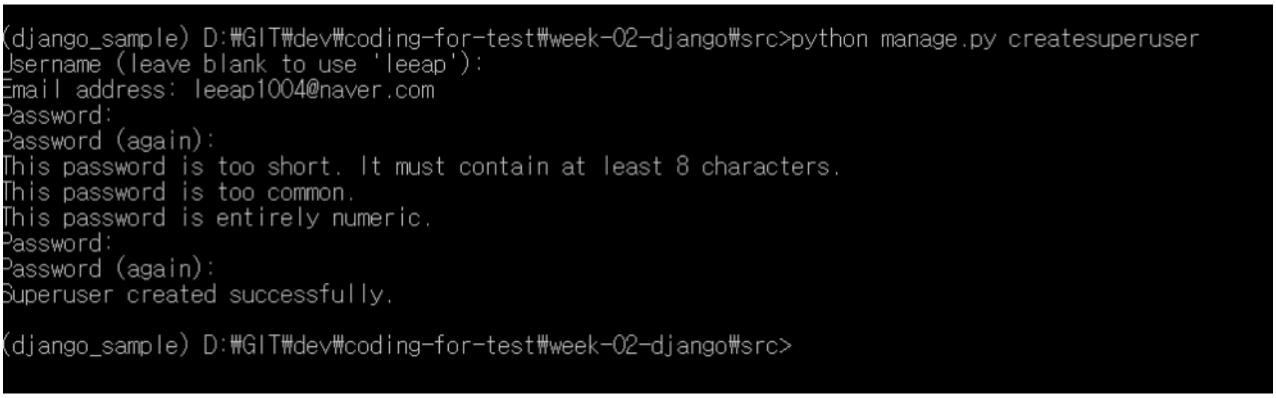
```
(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py shell
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from blog.models import Article
>>> Article
<class 'blog.models.Article'>
>>> Article.objects.all()
<QuerySet [<Article: Article object>, <Article: Article object>, <Article: Article object>, <Article: Article object>]>
>>> article_list = Article.objects.all()
>>> for article in article_list:
...     print(article.title)
...
hello
hello^^
hello^^
hello^^
>>> Article.object.create(title="abc",contents="test22",view_count=0)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: type object 'Article' has no attribute 'object'
>>> Article.objects.create(title="abc",contents="test22",view_count=0)
<Article: Article object>
>>> article_list
<QuerySet [<Article: Article object>, <Article: Article object>, <Article: Article object>, <Article: Article object>]>
>>> article_list = Article.objects.all()
>>> article_list
<QuerySet [<Article: Article object>, <Article: Article object>, <Article: Article object>, <Article: Article object>, <Article: Article object>]>
>>> for article in article_list:
...     print(article.title)
...
hello
hello^^
hello^^
hello^^
abc
>>>
```



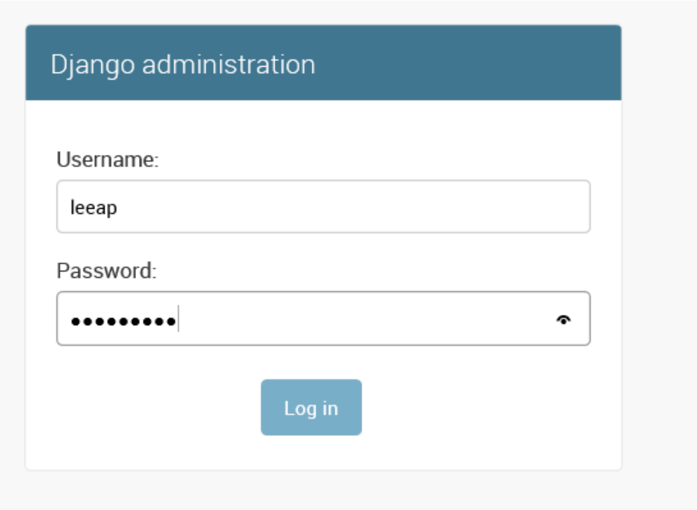
```
hello world, jungeun
hello world, Jay!!
hello
hello^^
hello^^
hello^^
abc
```

09. django관리자 등록

2019년 1월 12일 토요일　　오후 3:51



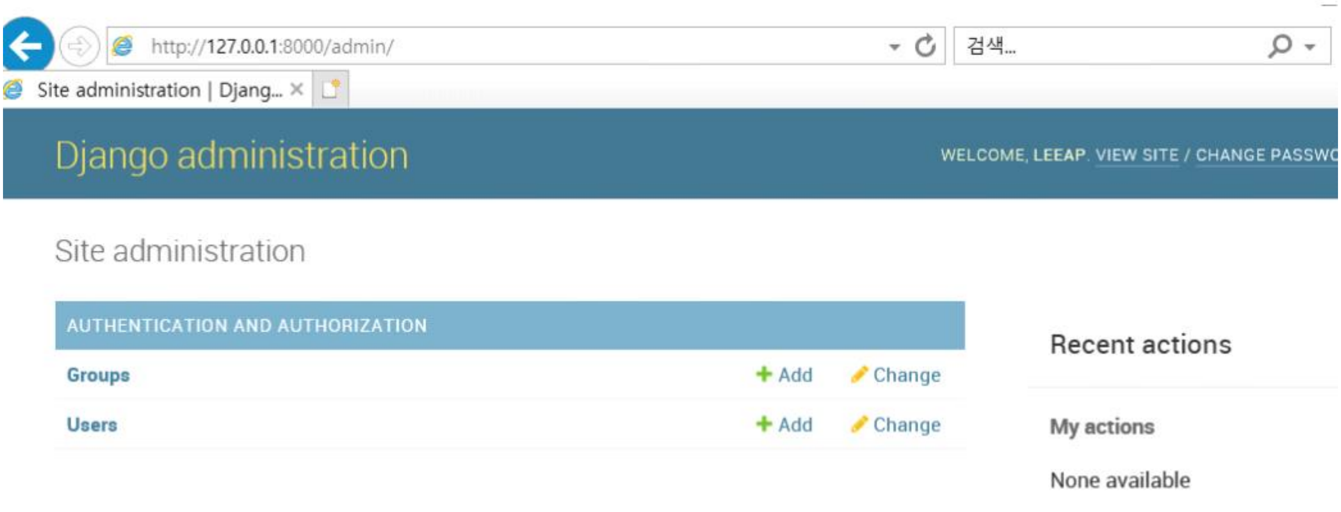
```
(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>python manage.py createsuperuser
Username (leave blank to use 'leeap'):
Email address: leeap1004@naver.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Password:
Password (again):
Superuser created successfully.

(django_sample) D:₩GIT₩dev₩coding-for-test₩week-02-django₩src>
```



Django administration

Username: leeap

Password: ●●●●●●●●●●



http://127.0.0.1:8000/admin/

Django administration

WELCOME, LEEAP. VIEW SITE / CHANGE PASSWO

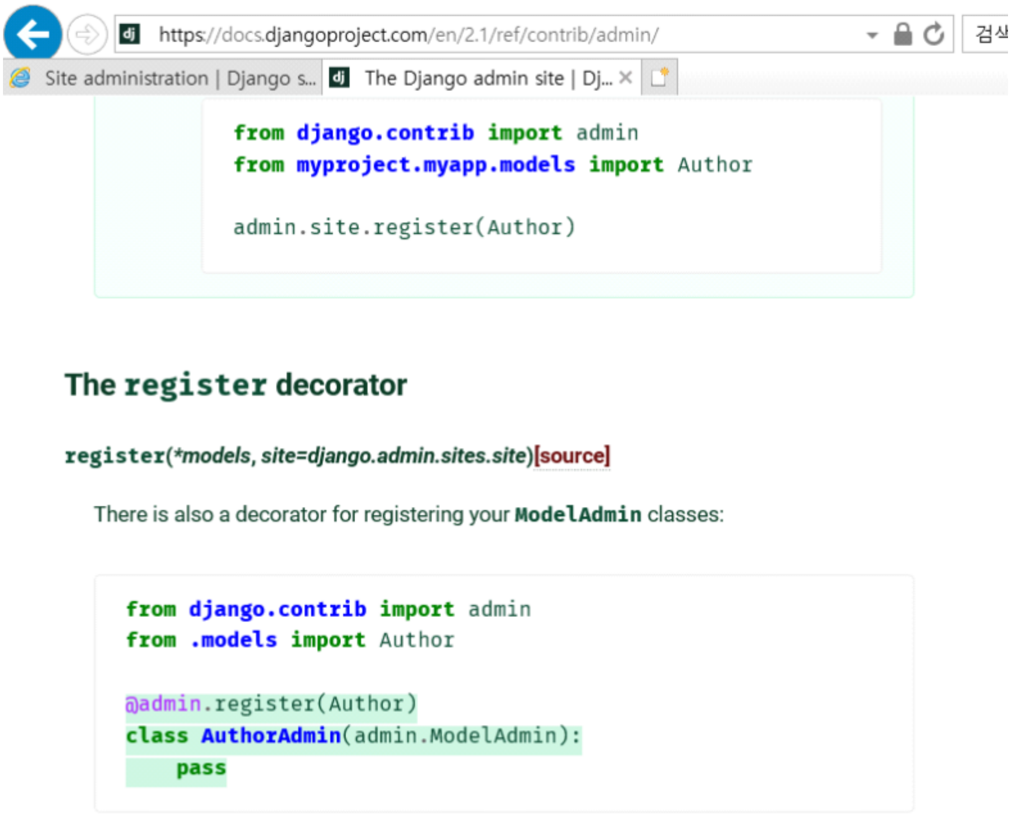
Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

Recent actions

My actions

None available


https://docs.djangoproject.com/en/2.1/ref/contrib/admin/
Site administration | Django s...
The Django admin site | Dj...
검색

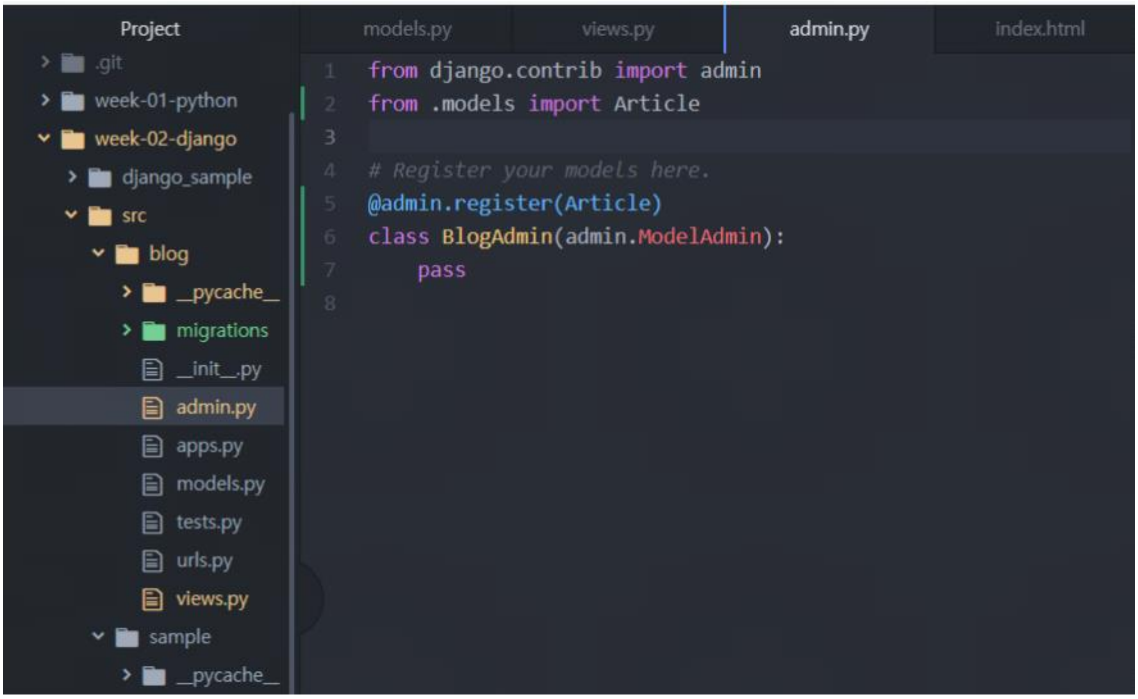
```
from django.contrib import admin
from myproject.myapp.models import Author

admin.site.register(Author)
```

The register decorator

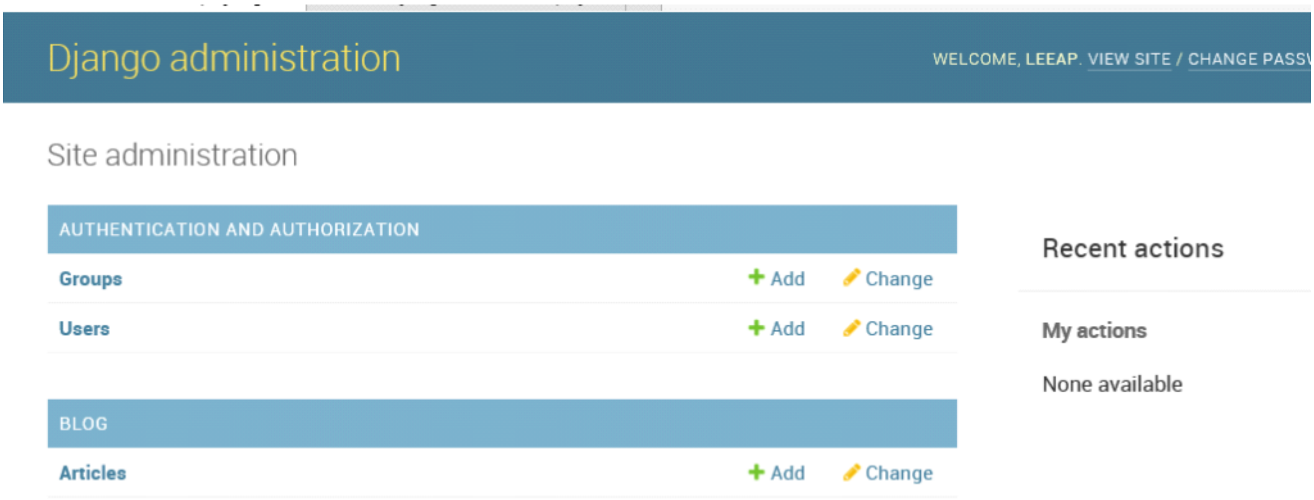
register(*models, site=django.admin.sites.site)[source]

There is also a decorator for registering your ModelAdmin classes:

```
from django.contrib import admin
from .models import Author

@admin.register(Author)
class AuthorAdmin(admin.ModelAdmin):
    pass
```



```
from django.contrib import admin
from .models import Article

# Register your models here.
@admin.register(Article)
class BlogAdmin(admin.ModelAdmin):
    pass
```



Django administration

WELCOME, LEEAP. VIEW SITE / CHANGE PASSW

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

BLOG		
Articles	+ Add	✎ Change

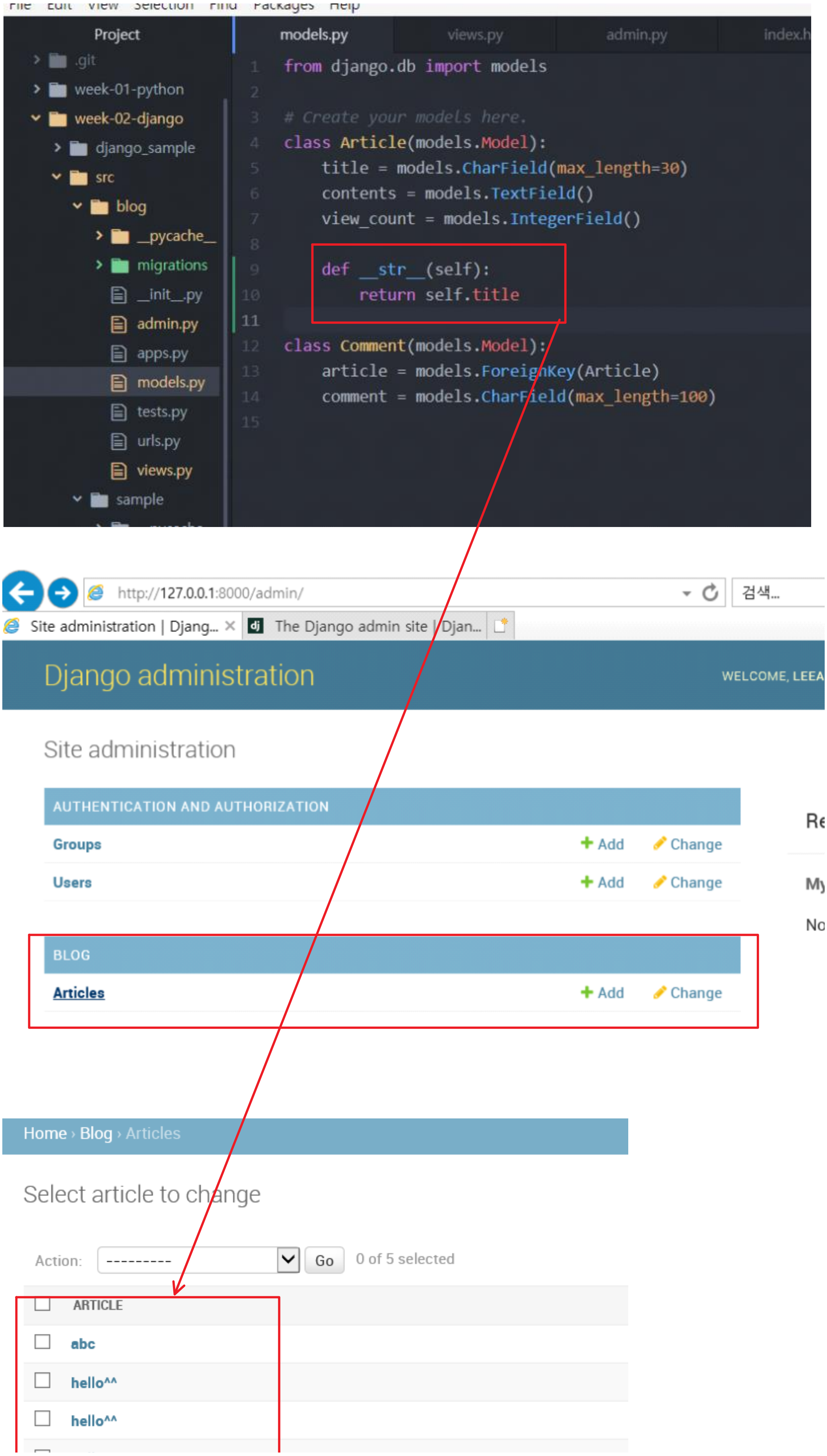
Recent actions

My actions

None available

관리자 페이지

2019년 1월 12일 토요일　　오후 4:22



```python
from django.db import models

# Create your models here.
class Article(models.Model):
    title = models.CharField(max_length=30)
    contents = models.TextField()
    view_count = models.IntegerField()

    def __str__(self):
        return self.title

class Comment(models.Model):
    article = models.ForeignKey(Article)
    comment = models.CharField(max_length=100)
```

http://127.0.0.1:8000/admin/
검색...
Site administration | Djang...
The Django admin site | Djan...
WELCOME, LEEA
Re
My
No

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

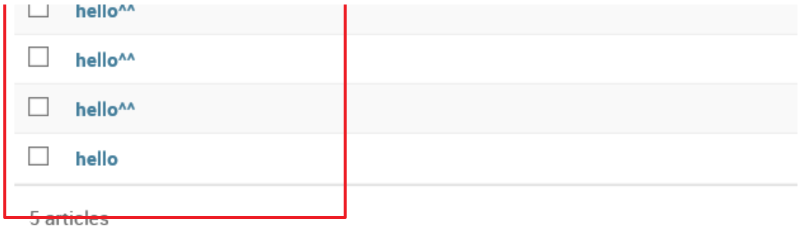
BLOG		
Articles	+ Add	Change

Home › Blog › Articles

Select article to change

Action: --------- Go 0 of 5 selected

☐	ARTICLE
☐	abc
☐	hello^^
☐	hello^^



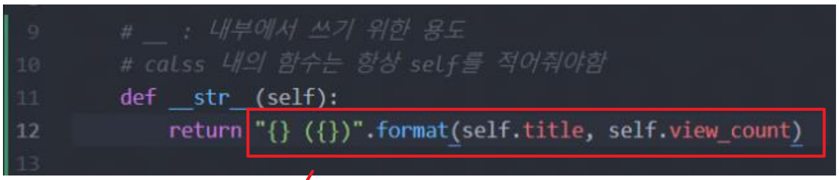
□ hello^^

□ hello^^

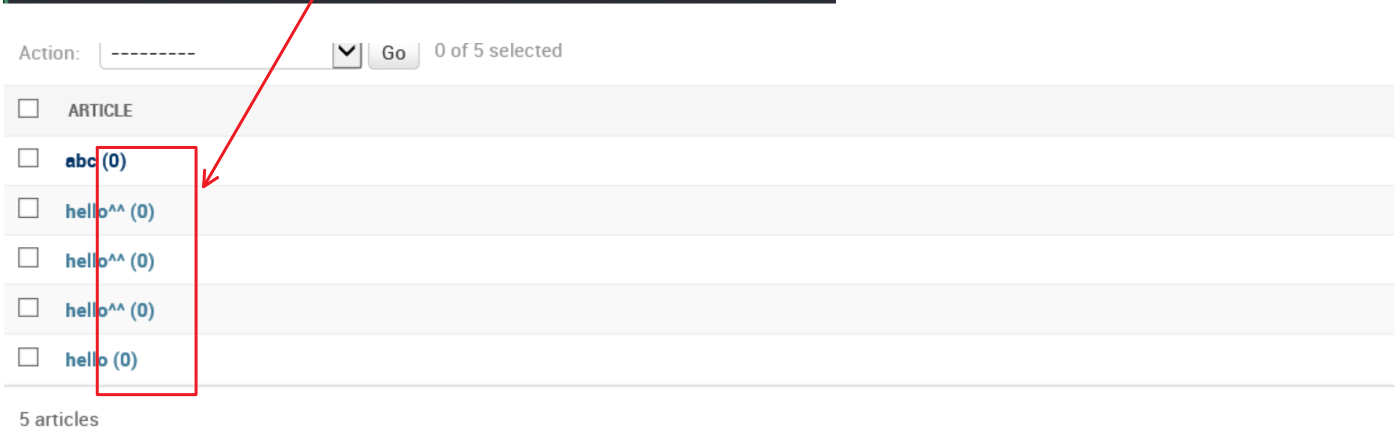
□ hello^^

□ hello

5 articles



```
    # __ : 내부에서 쓰기 위한 용도
    # calss 내의 함수는 항상 self를 적어줘야함
    def __str__(self):
        return "{} ({})".format(self.title, self.view_count)
```



Action: --------- Go 0 of 5 selected

□	ARTICLE
□	abc (0)
□	hello^^ (0)
□	hello^^ (0)
□	hello^^ (0)
□	hello (0)

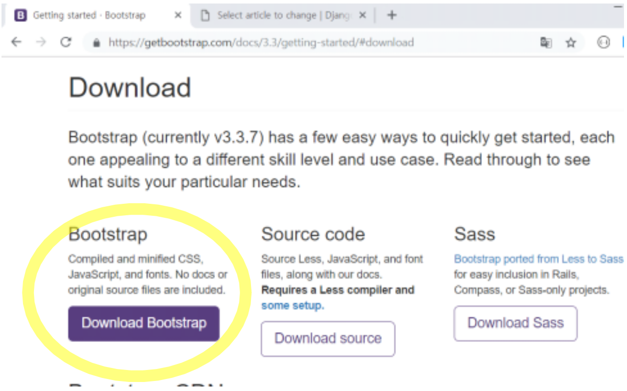
5 articles

10. bootstrap

2019년 1월 12일 토요일 오후 4:38

https://getbootstrap.com/docs/3.3/

부트스트랩 다운



Download

Bootstrap (currently v3.3.7) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

Download Bootstrap

Source code

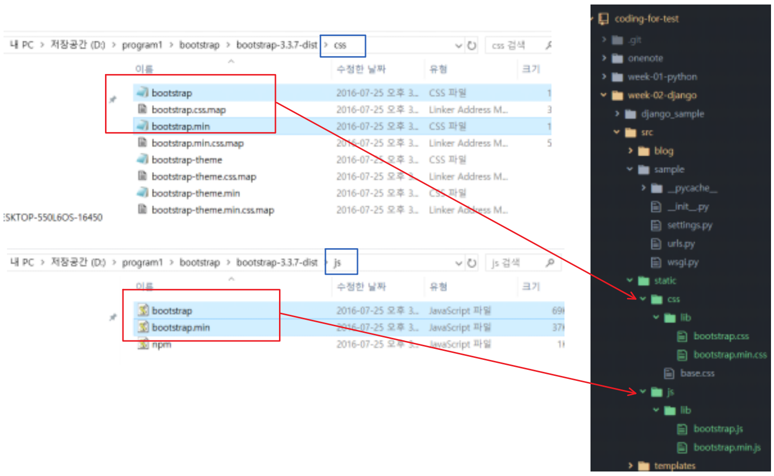
Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

Download source

Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

Download Sass

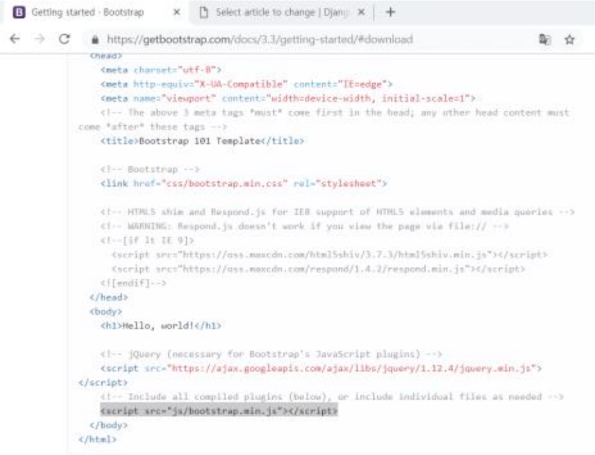


```
내 PC › 저장공간 (D:) › program1 › bootstrap › bootstrap-3.3.7-dist › css

이름                          수정한 날짜           유형               크기
bootstrap                     2016-07-25 오후 3...  CSS 파일           1
bootstrap.css.map             2016-07-25 오후 3...  Linker Address M...  3
bootstrap.min                 2016-07-25 오후 3...  CSS 파일           1
bootstrap.min.css.map         2016-07-25 오후 3...  Linker Address M...  5
bootstrap-theme               2016-07-25 오후 3...  CSS 파일
bootstrap-theme.css.map       2016-07-25 오후 3...  Linker Address M...
bootstrap-theme.min           2016-07-25 오후 3...  CSS 파일
bootstrap-theme.min.css.map   2016-07-25 오후 3...  Linker Address M...
```

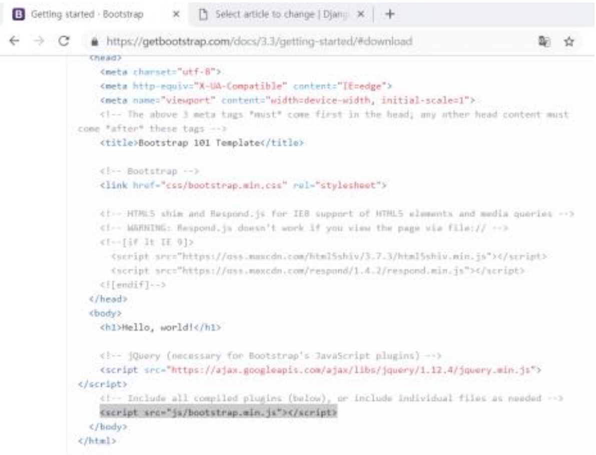
:SKTOP-550L6OS-16450

```
내 PC › 저장공간 (D:) › program1 › bootstrap › bootstrap-3.3.7-dist › js

이름                          수정한 날짜           유형               크기
bootstrap                     2016-07-25 오후 3...  JavaScript 파일    69
bootstrap.min                 2016-07-25 오후 3...  JavaScript 파일    37
npm                           2016-07-25 오후 3...  JavaScript 파일    1
```

```
coding-for-test
  › .git
  › onenote
  › week-01-python
  ⌄ week-02-django
    › django_sample
    ⌄ src
      › blog
      ⌄ sample
        › __pycache__
          __init__.py
          settings.py
          urls.py
          wsgi.py
      ⌄ static
        ⌄ css
          ⌄ lib
              bootstrap.css
              bootstrap.min.css
            base.css
        ⌄ js
          ⌄ lib
              bootstrap.js
              bootstrap.min.js
      › templates
```

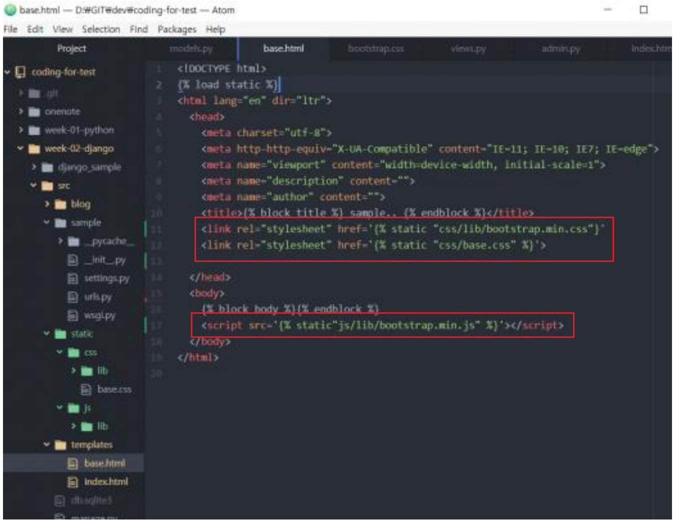


```
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
</script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```



Getting started · Bootstrap
Select article to change | Djang
https://getbootstrap.com/docs/3.3/getting-started/#download

```
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
</script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```



```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta http-http-equiv="X-UA-Compatible" content="IE=11; IE=10; IE7; IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>{% block title %} sample.. {% endblock %}</title>
    <link rel="stylesheet" href='{% static "css/lib/bootstrap.min.css"}'
    <link rel="stylesheet" href='{% static "css/base.css" %}'>

  </head>
  <body>
    {% block body %}{% endblock %}
    <script src='{% static"js/lib/bootstrap.min.js" %}'></script>
  </body>
</html>
```

https://getbootstrap.com/docs/3.4/examples/theme/
https://getbootstrap.com/docs/3.4/examples/jumbotron/
https://getbootstrap.com/docs/3.3/css/