django
INSTALLED APPS
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
PROJECT
Real Python

Introduction to Django

BY FIRAT TAMUR

What is the RoadMap

Steps:

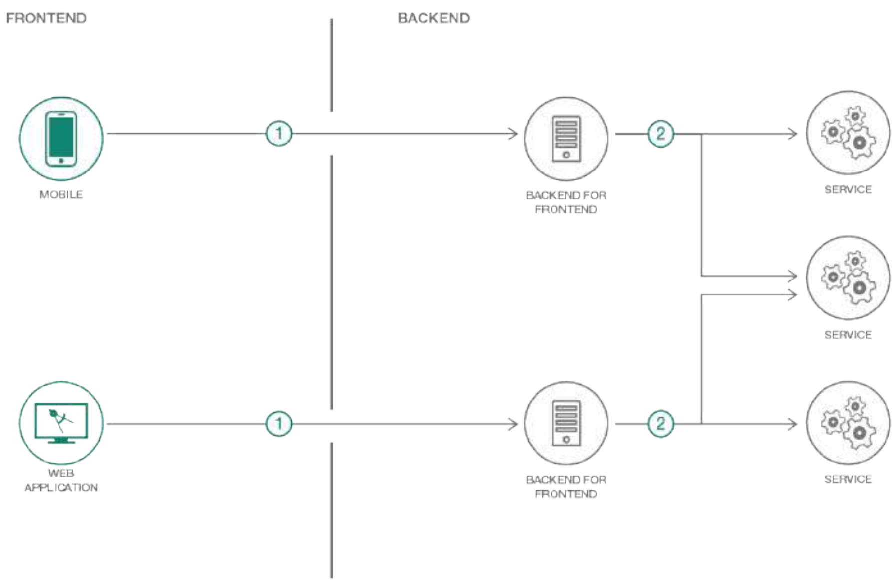
- What is the Front-end and Back-end.
- What is the Django and Why Django?
- Django Components.
- Create our first Django Project.
- Deploy our blog project with pythonanywhere.com site for free.

Front End – Back End Frameworks

- There are two sides to building websites.

- Front End frameworks are libraries which used to create the appearance of the websites.
- Shortly: Design, Appearance.

- Back End frameworks are libraries which used to create models and database connections.
- Shortly: Databases, Models, Functionality.

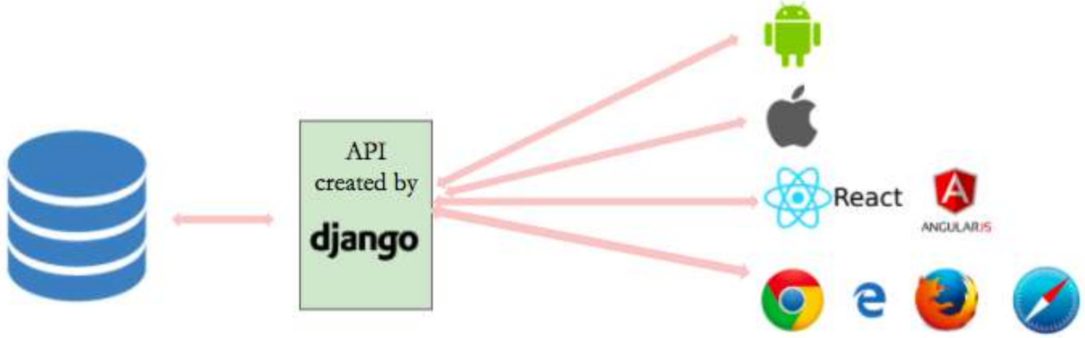


What is the Django? and Why Django?

- Django is a High-Level Python Web Back End Framework.

- Django helps programmers to create complex websites quickly and with less code.

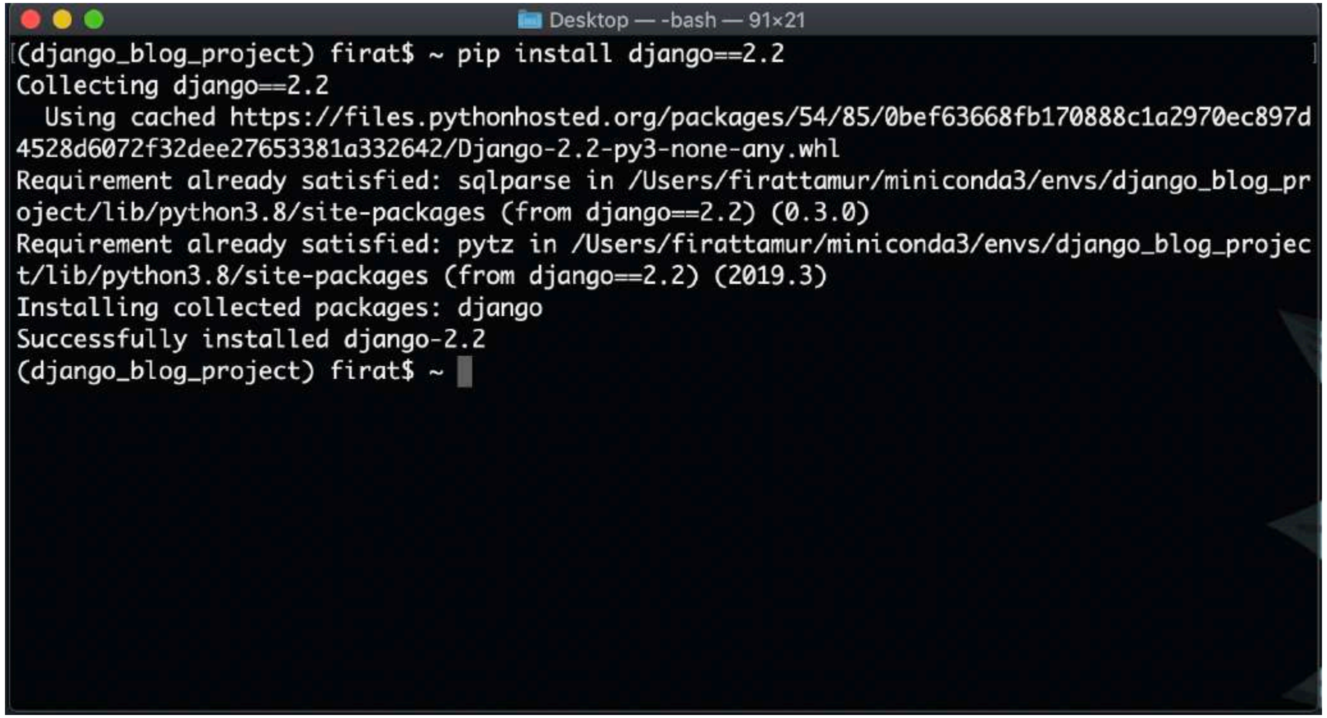
- It is fun to write websites in Python.



Django Main Components

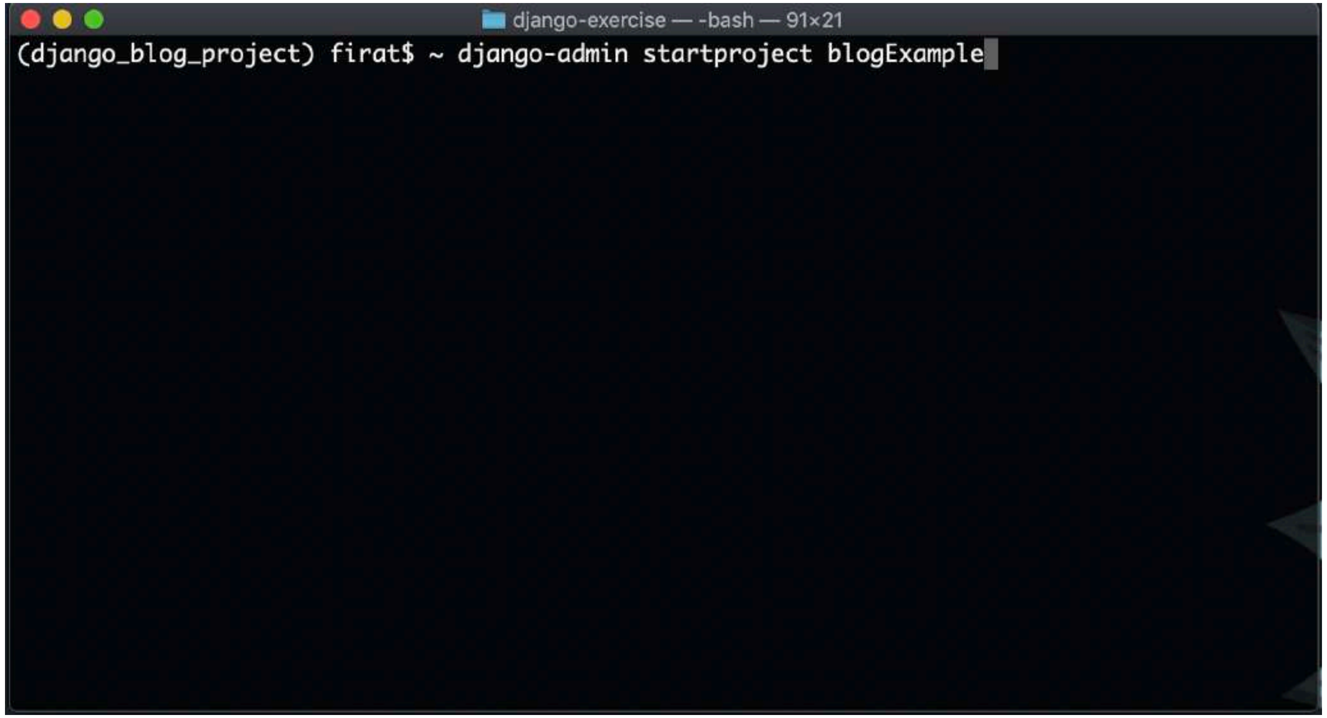
Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

Install Django



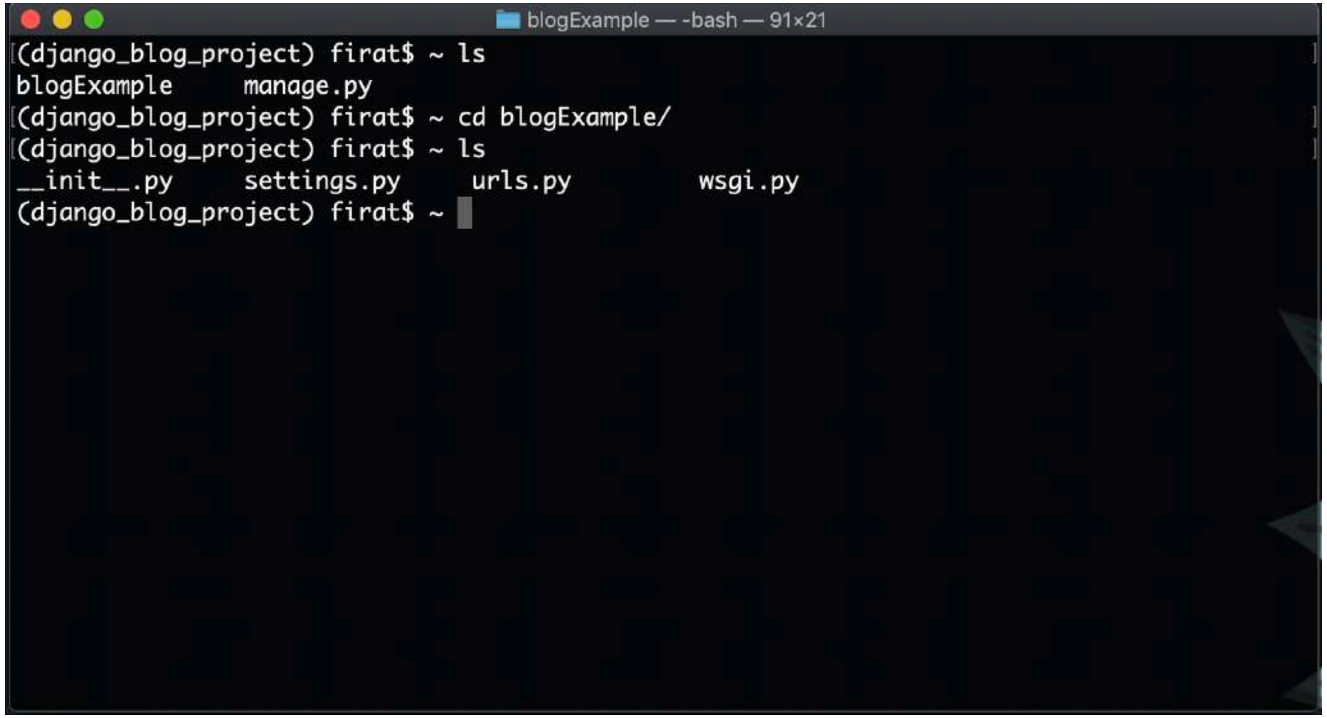
```
(django_blog_project) firat$ ~ pip install django==2.2
Collecting django==2.2
  Using cached https://files.pythonhosted.org/packages/54/85/0bef63668fb170888c1a2970ec897d
4528d6072f32dee27653381a332642/Django-2.2-py3-none-any.whl
Requirement already satisfied: sqlparse in /Users/firattamur/miniconda3/envs/django_blog_pr
oject/lib/python3.8/site-packages (from django==2.2) (0.3.0)
Requirement already satisfied: pytz in /Users/firattamur/miniconda3/envs/django_blog_projec
t/lib/python3.8/site-packages (from django==2.2) (2019.3)
Installing collected packages: django
Successfully installed django-2.2
(django_blog_project) firat$ ~
```

Start Project in Django



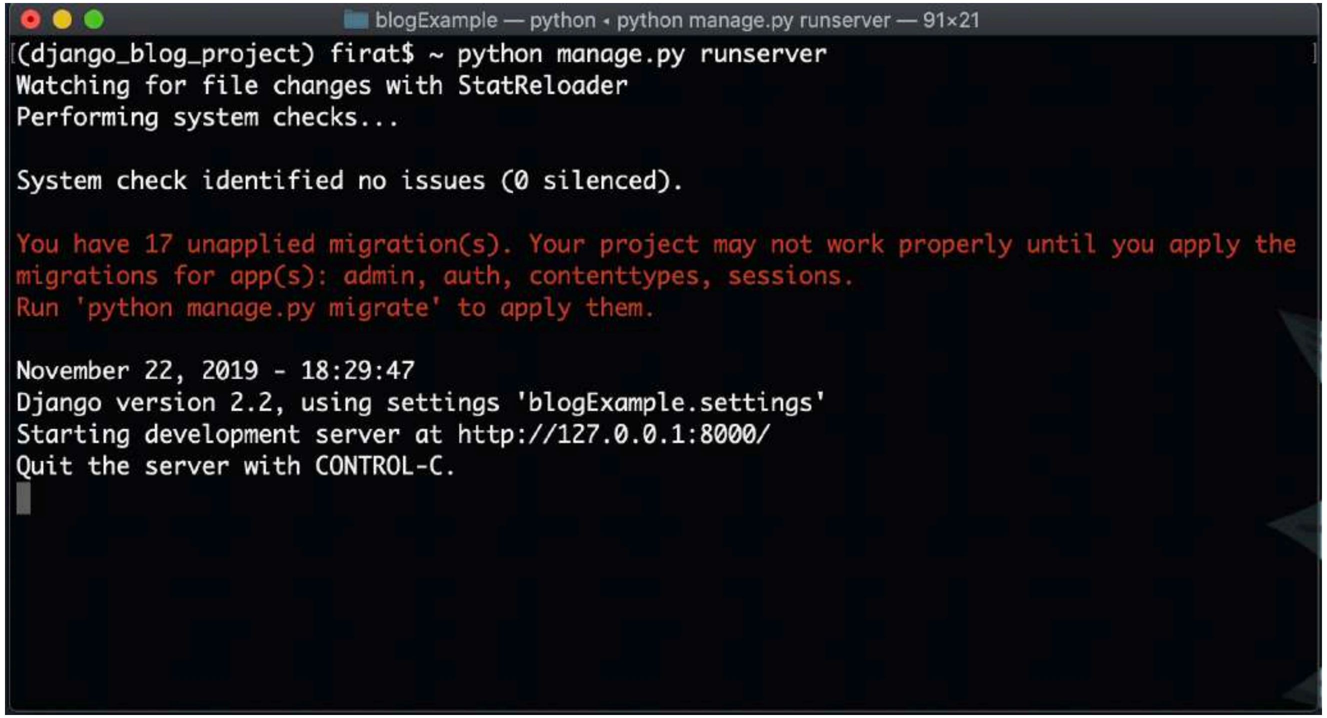
```
(django_blog_project) firat$ ~ django-admin startproject blogExample
```

Django Project Files

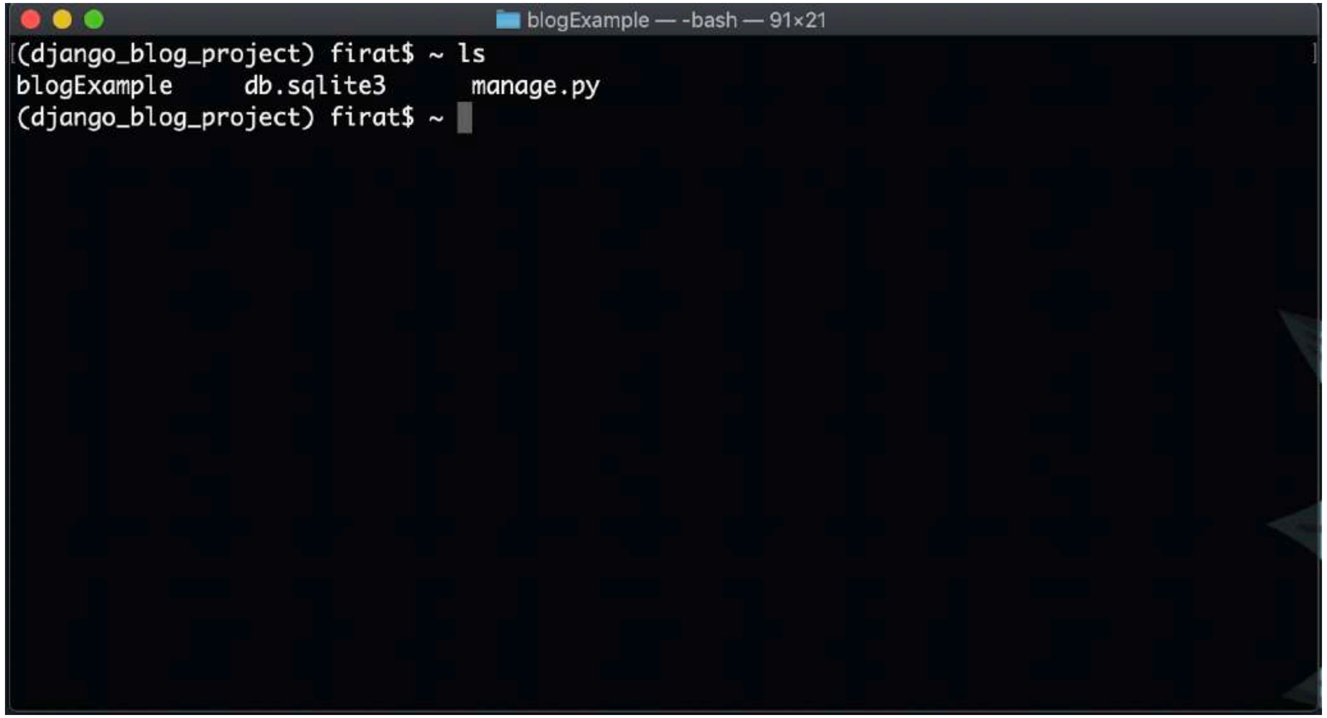


```
(django_blog_project) firat$ ~ ls
blogExample     manage.py
(django_blog_project) firat$ ~ cd blogExample/
(django_blog_project) firat$ ~ ls
__init__.py     settings.py     urls.py         wsgi.py
(django_blog_project) firat$ ~
```

Run Django project in first time.



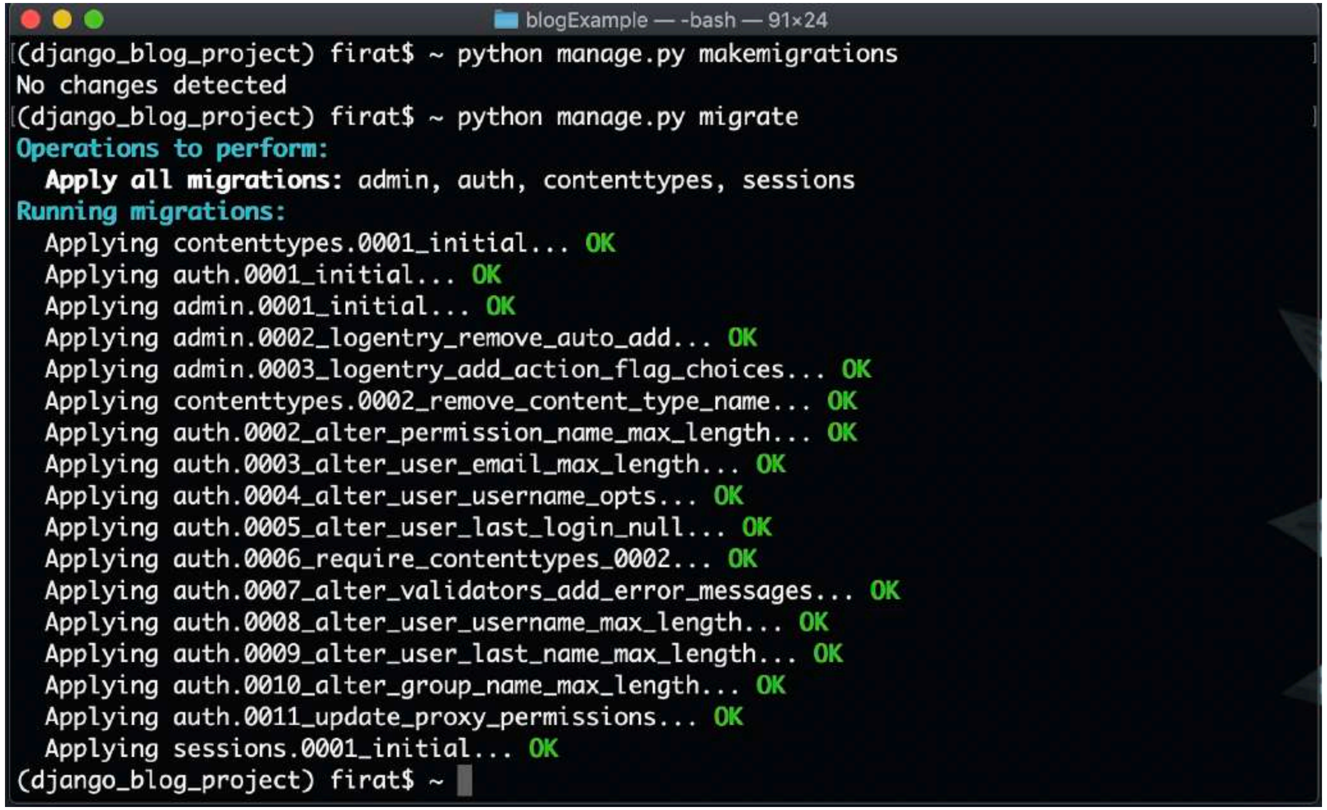
```
(django_blog_project) firat$ ~ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 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.

November 22, 2019 - 18:29:47
Django version 2.2, using settings 'blogExample.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```



```
(django_blog_project) firat$ ~ ls
blogExample      db.sqlite3       manage.py
(django_blog_project) firat$ ~ 
```

After first run Django project.

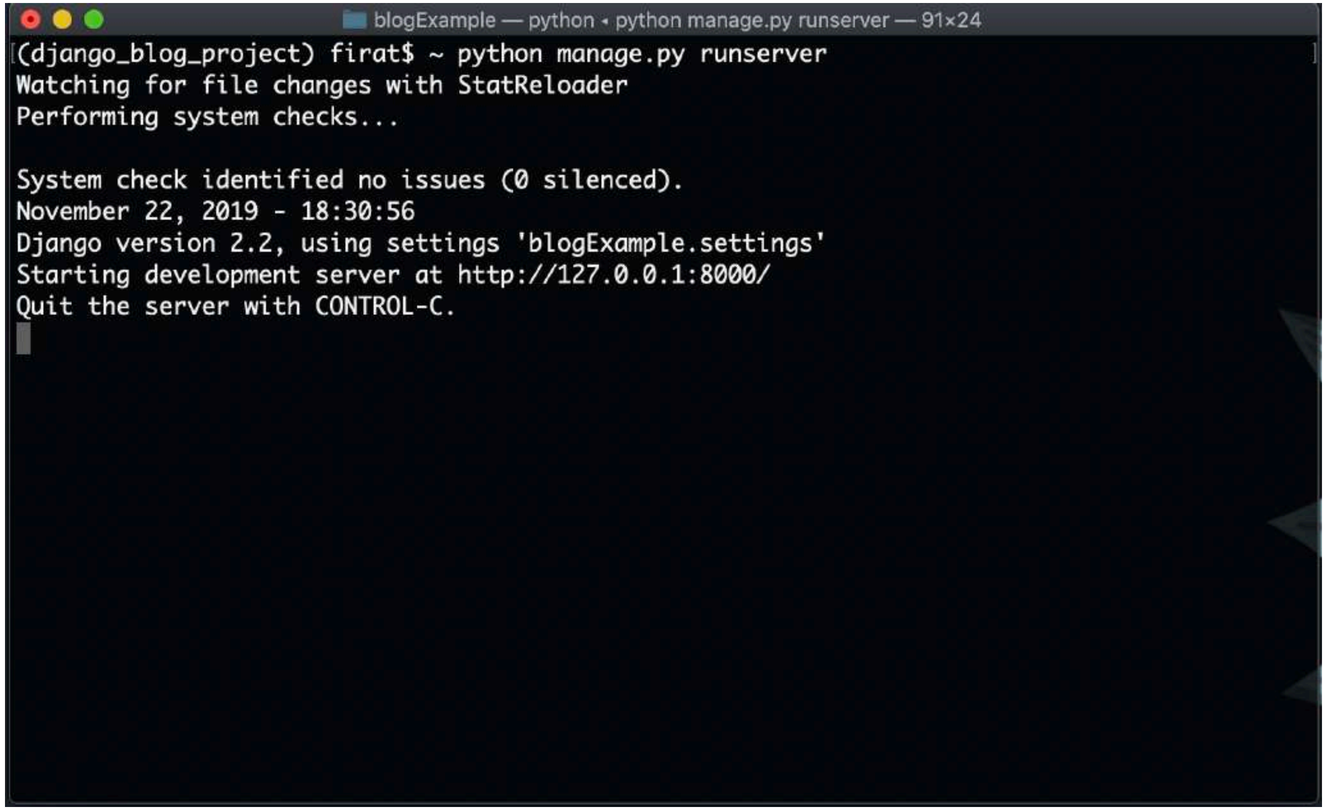
Django Project migrations



```
(django_blog_project) firat$ ~ python manage.py makemigrations
No changes detected
(django_blog_project) firat$ ~ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
(django_blog_project) firat$ ~
```

python manage.py migrate:
Run the migrations (execute the SQL commands).

Run Django project after migrations.

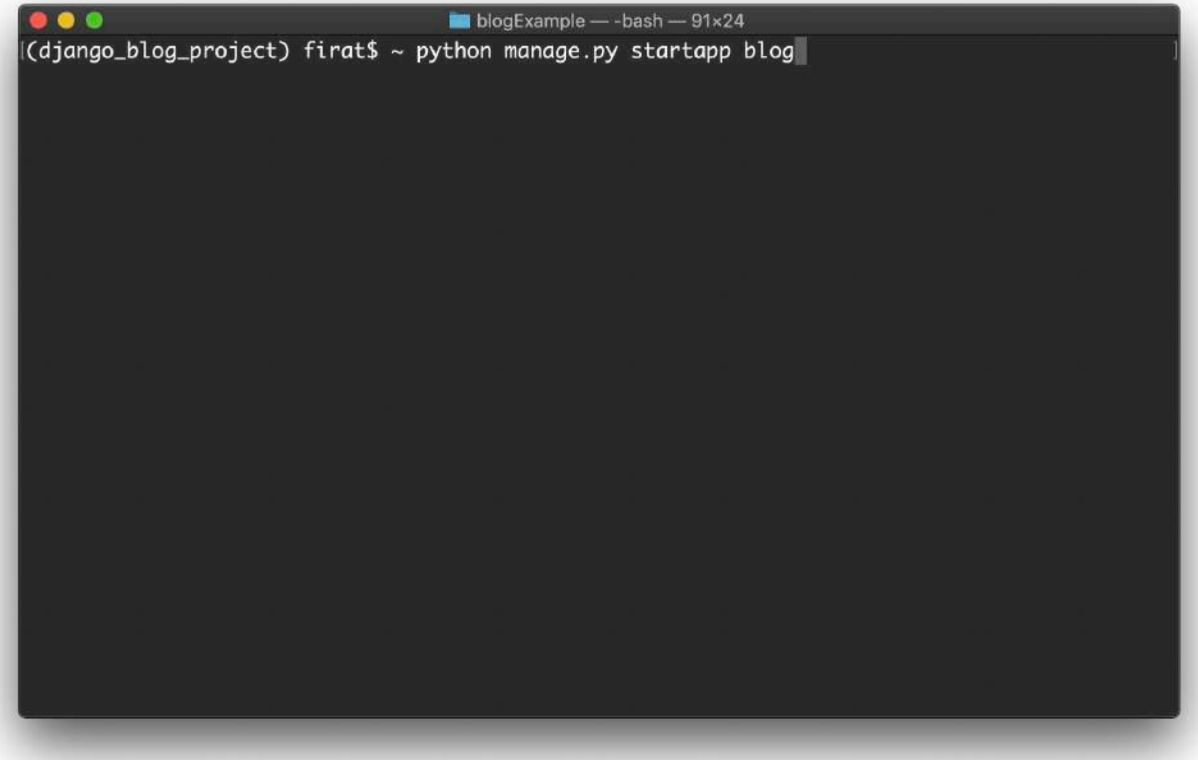


```
(django_blog_project) firat$ ~ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 22, 2019 - 18:30:56
Django version 2.2, using settings 'blogExample.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Django Main Components

Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

Create a Django App



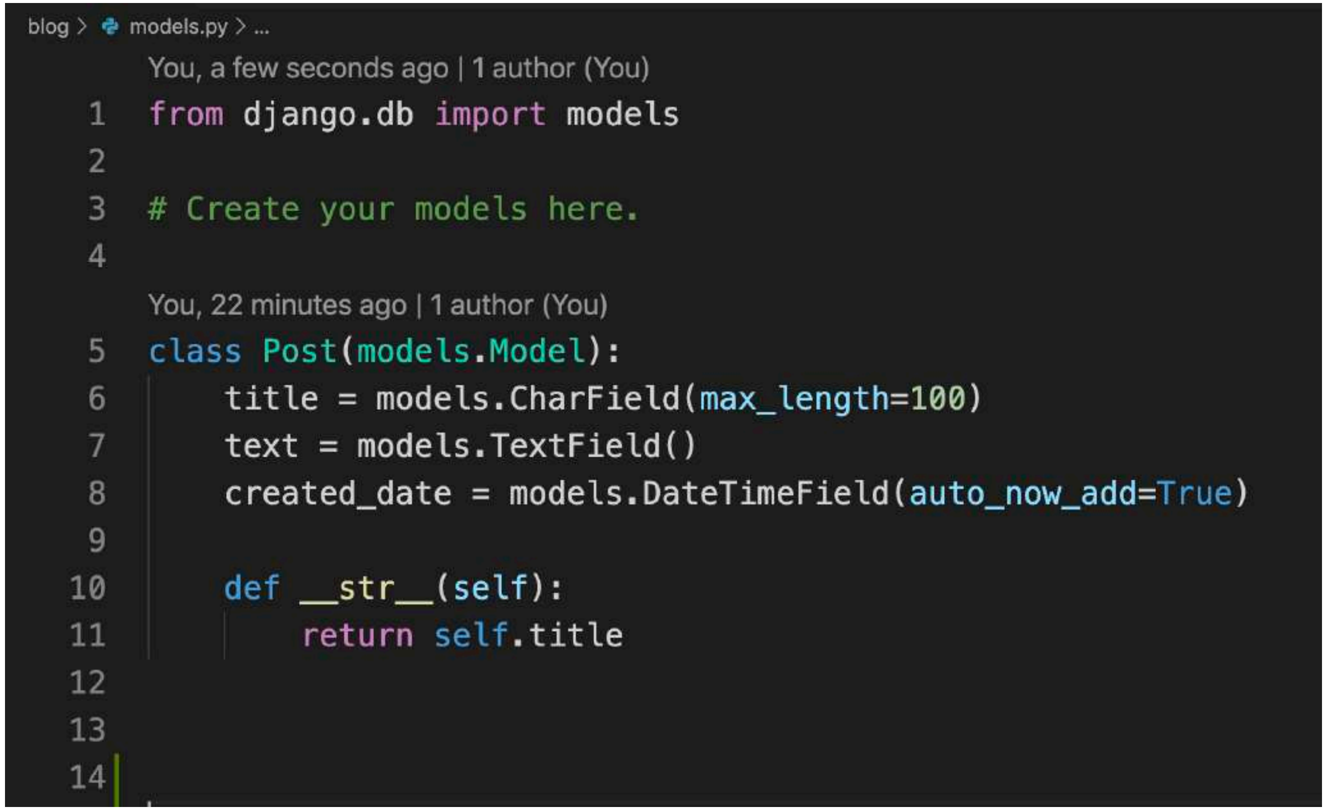
```
(django_blog_project) firat$ ~ python manage.py startapp blog
```

python manage.py makemigrations <app-name>: Create the migrations (generate the SQL commands).

Django Main Components

Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

A simple Post Model

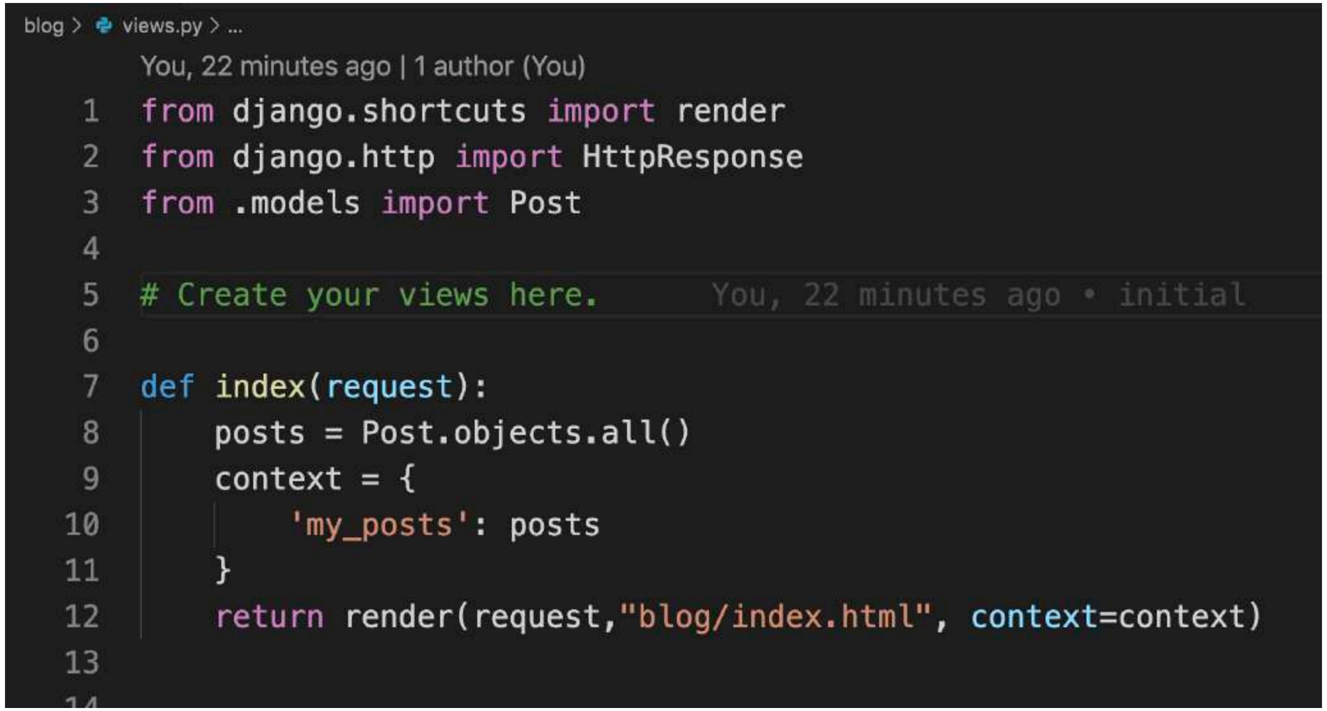

blog > models.py > ...
You, a few seconds ago | 1 author (You)
You, 22 minutes ago | 1 author (You)

```python
from django.db import models

# Create your models here.

class Post(models.Model):
    title = models.CharField(max_length=100)
    text = models.TextField()
    created_date = models.DateTimeField(auto_now_add=True)

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

Django Main Components

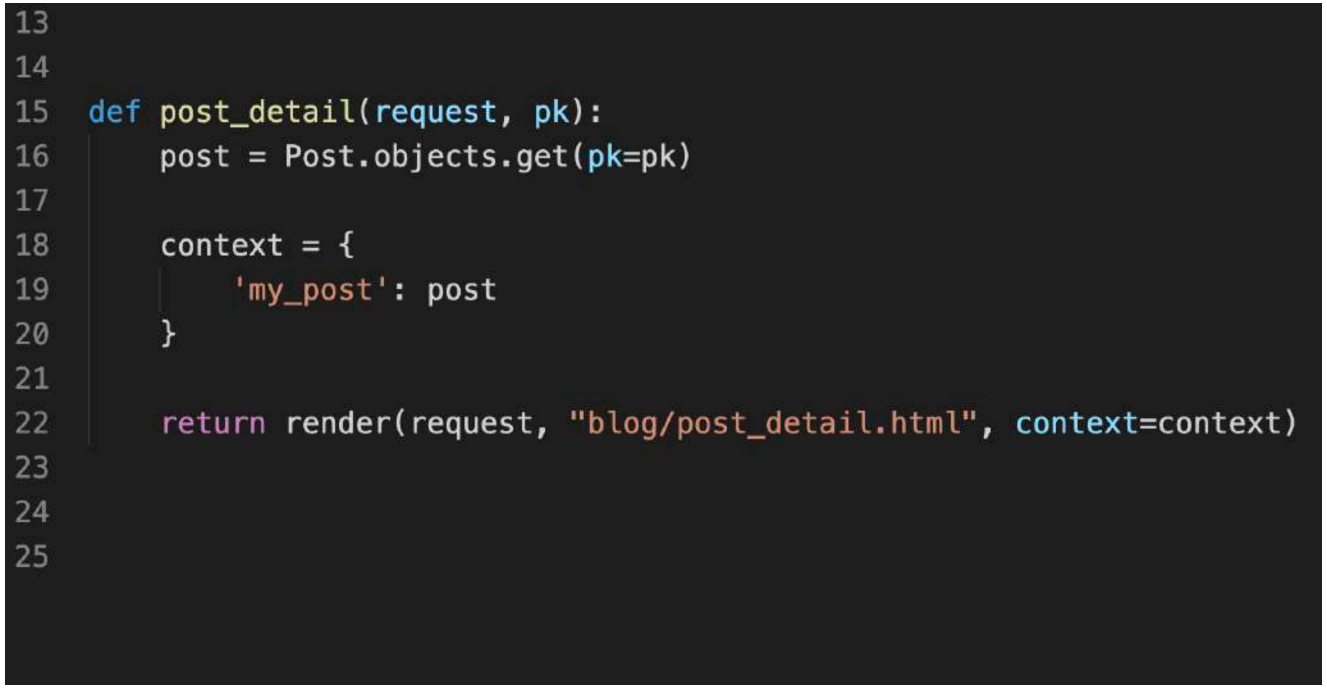
Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

Index View


blog > views.py > ...
You, 22 minutes ago | 1 author (You)
You, 22 minutes ago • initial

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

# Create your views here.

def index(request):
    posts = Post.objects.all()
    context = {
        'my_posts': posts
    }
    return render(request,"blog/index.html", context=context)

```

Post Detail View

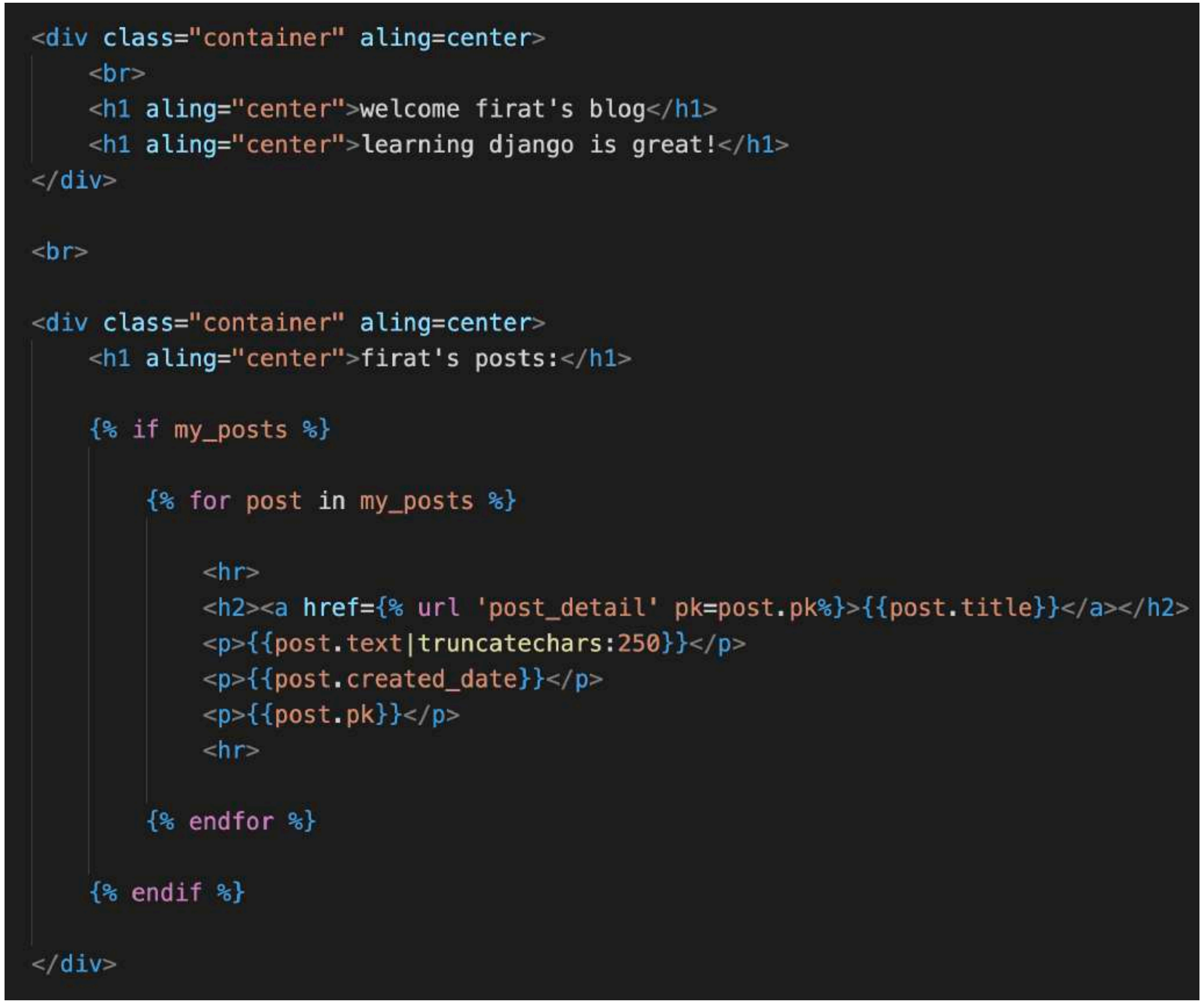


```python
def post_detail(request, pk):
    post = Post.objects.get(pk=pk)

    context = {
        'my_post': post
    }

    return render(request, "blog/post_detail.html", context=context)
```

Django Main Components

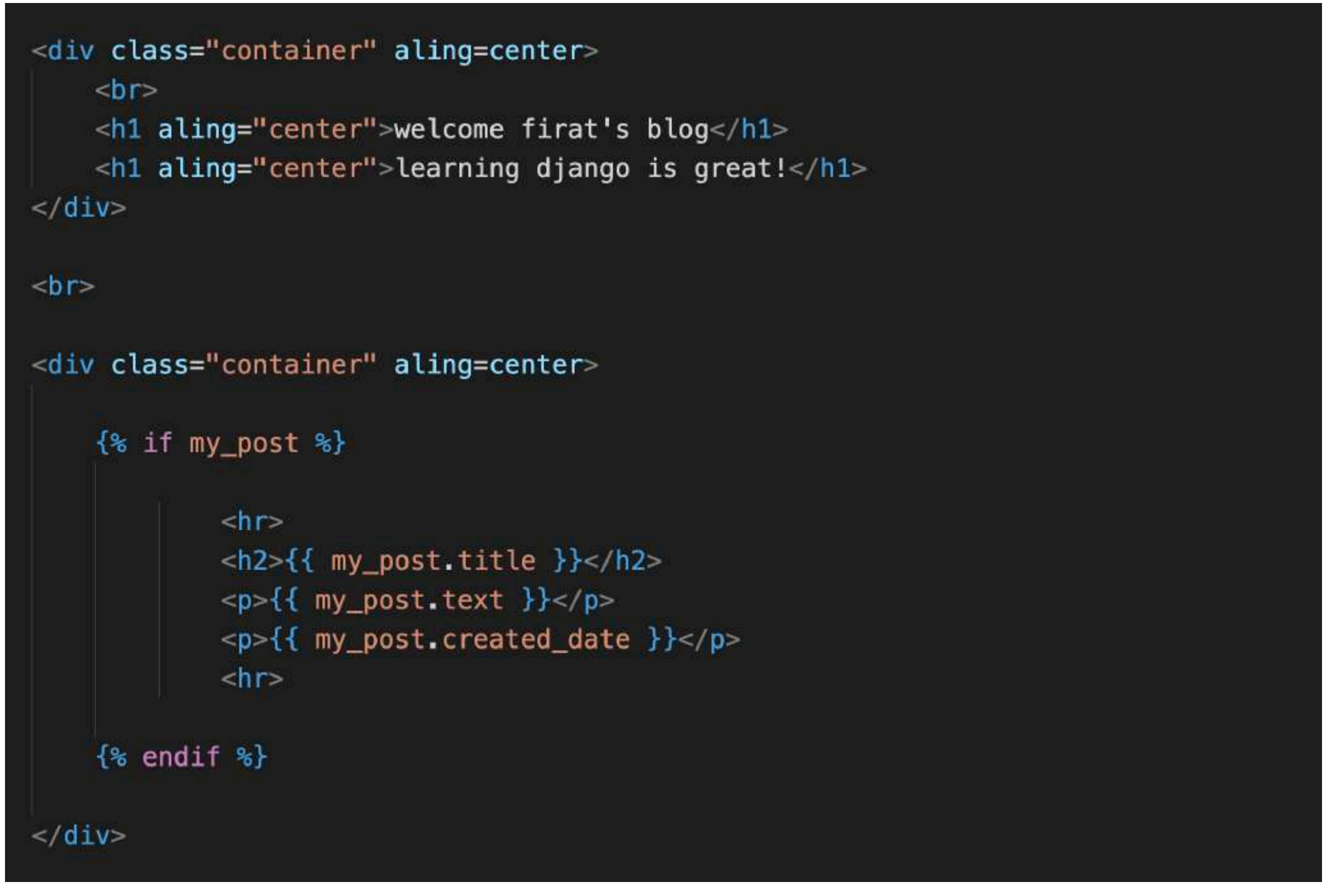
Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

Index Template



```
<div class="container" aling=center>
    <br>
    <h1 aling="center">welcome firat's blog</h1>
    <h1 aling="center">learning django is great!</h1>
</div>

<br>

<div class="container" aling=center>
    <h1 aling="center">firat's posts:</h1>

    {% if my_posts %}

        {% for post in my_posts %}

            <hr>
            <h2><a href={% url 'post_detail' pk=post.pk%}>{{post.title}}</a></h2>
            <p>{{post.text|truncatechars:250}}</p>
            <p>{{post.created_date}}</p>
            <p>{{post.pk}}</p>
            <hr>

        {% endfor %}

    {% endif %}

</div>
```

Post Detail Template

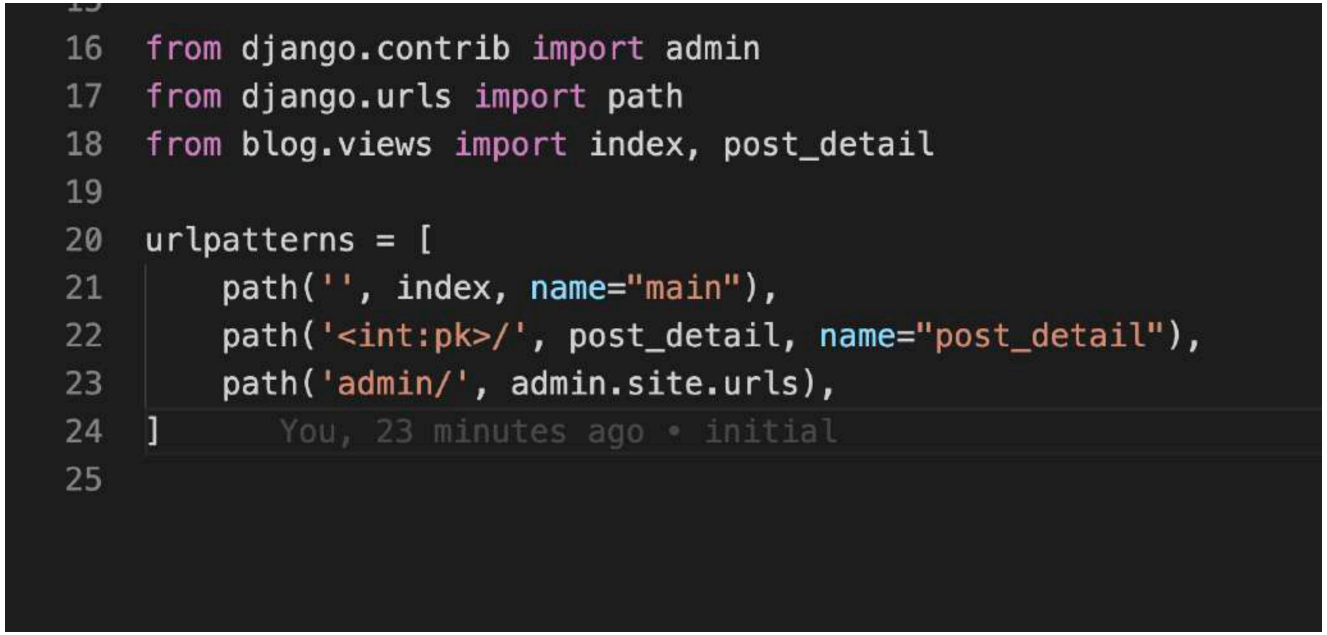


```
<div class="container" aling=center>
    <br>
    <h1 aling="center">welcome firat's blog</h1>
    <h1 aling="center">learning django is great!</h1>
</div>

<br>

<div class="container" aling=center>

    {% if my_post %}

            <hr>
            <h2>{{ my_post.title }}</h2>
            <p>{{ my_post.text }}</p>
            <p>{{ my_post.created_date }}</p>
            <hr>

    {% endif %}

</div>
```

Django Main Components

Apps	django apps
Models	django models
Views	django views
Templates	django templates
Urlpatterns	django urls

Urls.py File



```
from django.contrib import admin
from django.urls import path
from blog.views import index, post_detail

urlpatterns = [
    path('', index, name="main"),
    path('<int:pk>/', post_detail, name="post_detail"),
    path('admin/', admin.site.urls),
]      You, 23 minutes ago • initial

```

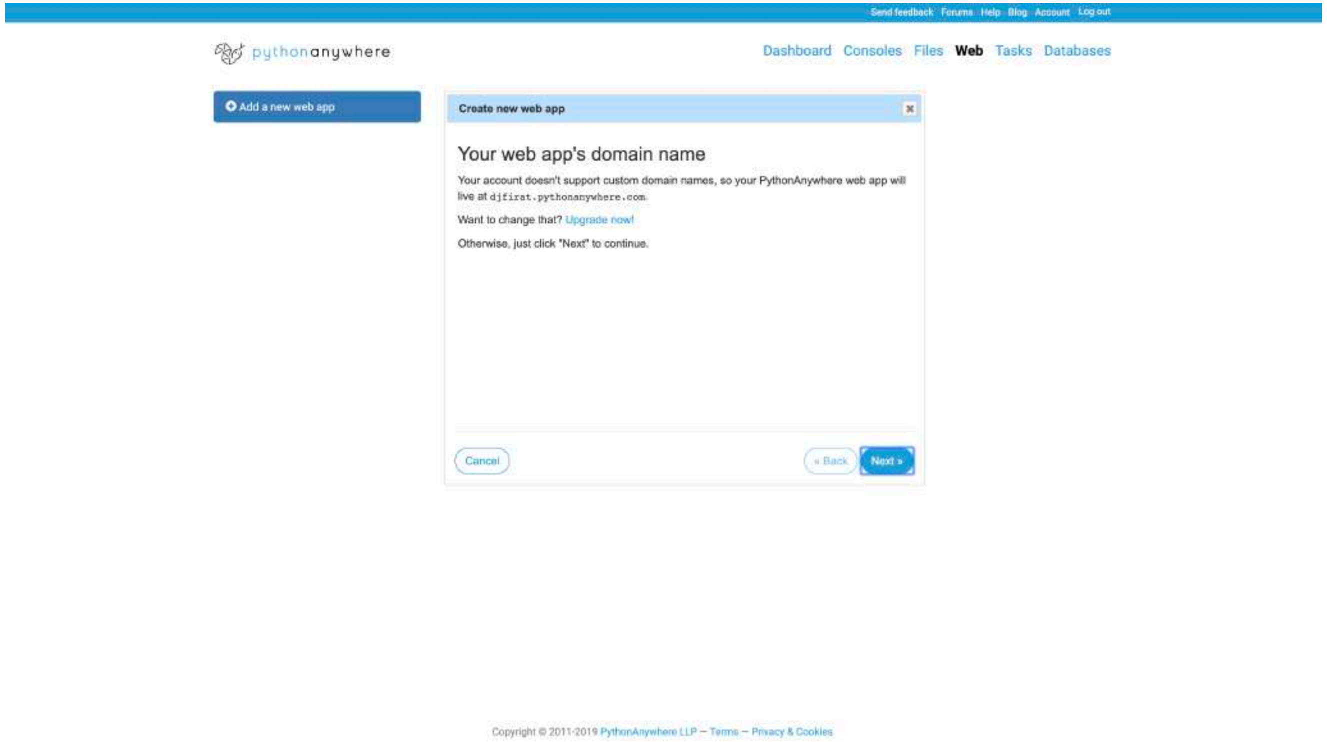
Deployment the Project.

Steps:

- **login/sign up pythonanywhere.com.**
- **open bash console.**
- **git clone project from GitHub.**
- **create venv in the project.**
- **activate venv and install requirements.**
- **remove database.**
- **apply migrations.**
- **create admin user.**
- **run all project.**

Last steps for deployment.

Add a new web app



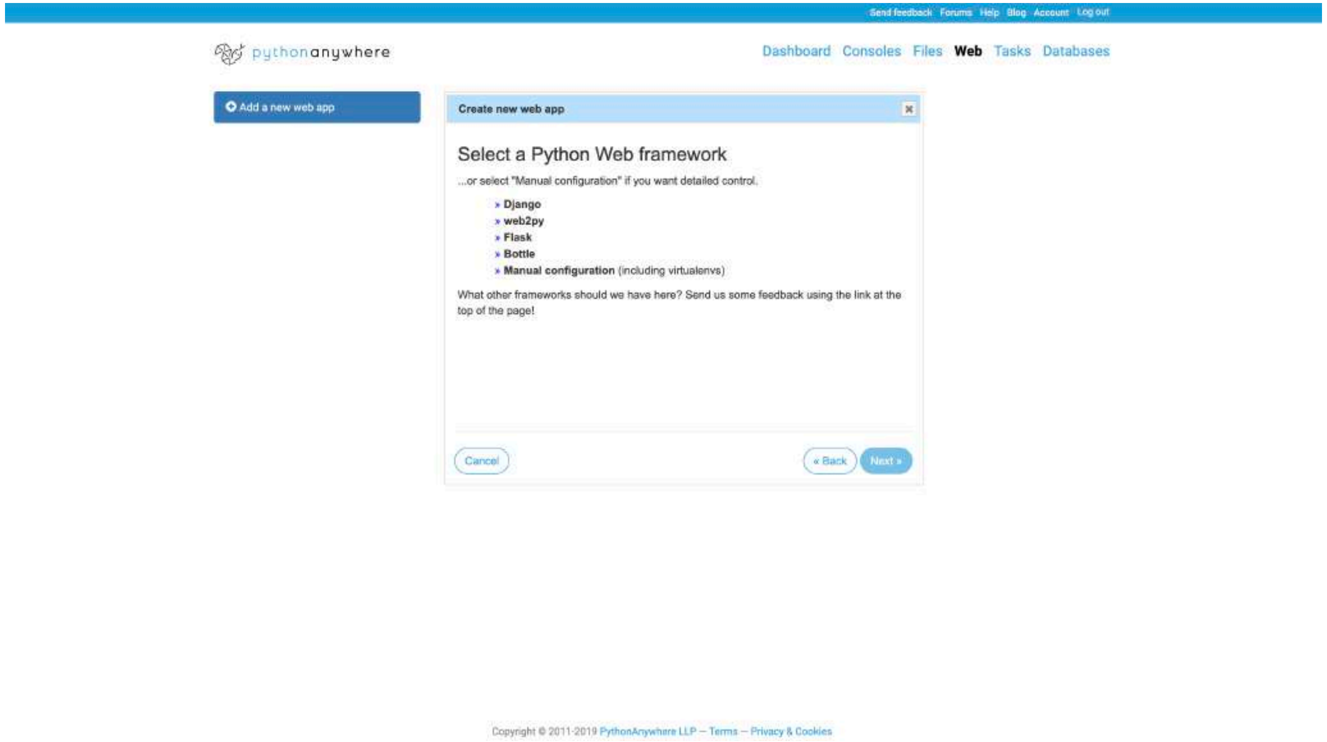
Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `djfirat.pythonanywhere.com`.

Want to change that? Upgrade now!

Otherwise, just click "Next" to continue.

Copyright © 2011-2019 PythonAnywhere LLP — Terms — Privacy & Cookies



Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

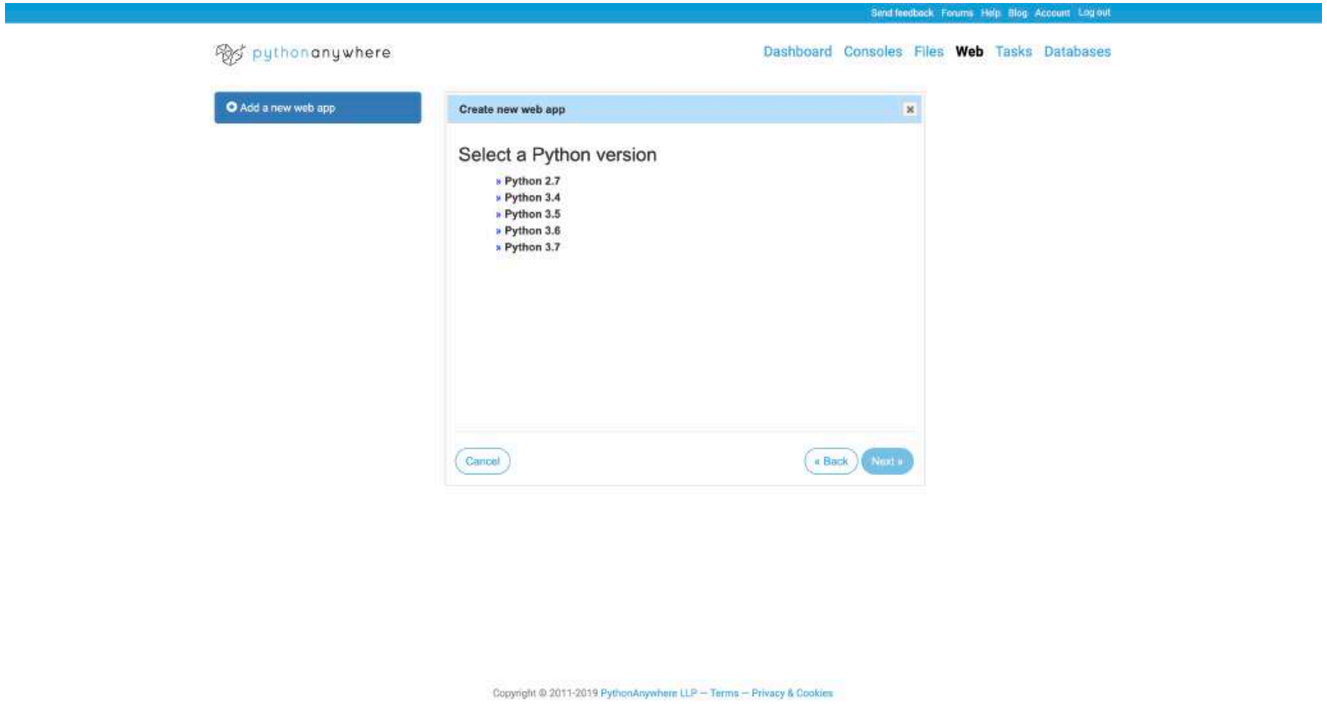
- **Django**
- **web2py**
- **Flask**
- **Bottle**
- **Manual configuration** (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Copyright © 2011-2019 PythonAnywhere LLP — Terms — Privacy & Cookies

Select the Manual Configuration

Select the Python version you used in the project.


Send feedback Forums Help Blog Account Log out
pythonanywhere
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Cancel
« Back
Next »

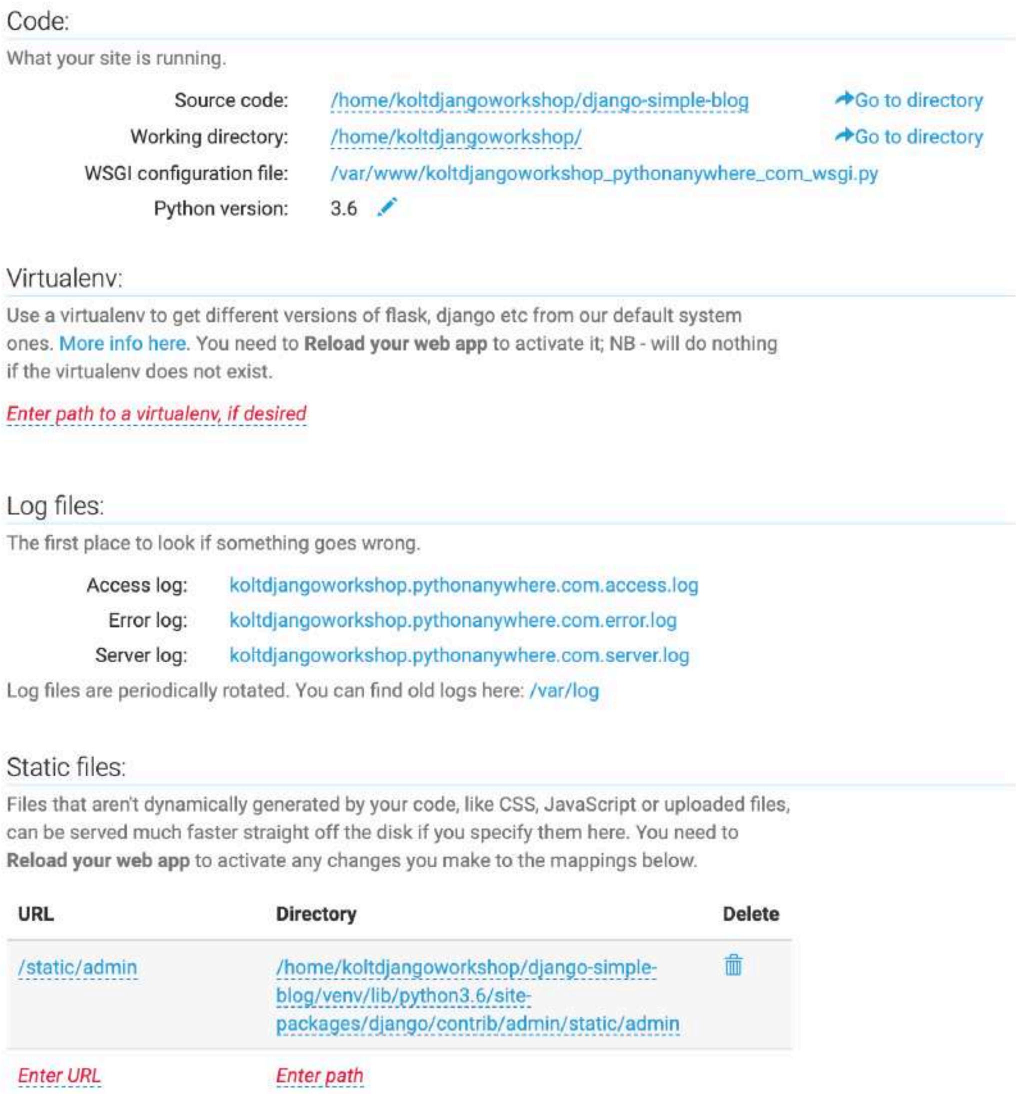
Create new web app

Select a Python version

- » Python 2.7
- » Python 3.4
- » Python 3.5
- » Python 3.6
- » Python 3.7

Copyright © 2011-2019 PythonAnywhere LLP — Terms — Privacy & Cookies

Directory Configuration



Code:

What your site is running.

Source code:	/home/koltdjangoworkshop/django-simple-blog	➦Go to directory
Working directory:	/home/koltdjangoworkshop/	➦Go to directory
WSGI configuration file:	/var/www/koltdjangoworkshop_pythonanywhere_com_wsgi.py	
Python version:	3.6 ✎	

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. More info here. You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

Enter path to a virtualenv, if desired

Log files:

The first place to look if something goes wrong.

Access log:	koltdjangoworkshop.pythonanywhere.com.access.log
Error log:	koltdjangoworkshop.pythonanywhere.com.error.log
Server log:	koltdjangoworkshop.pythonanywhere.com.server.log

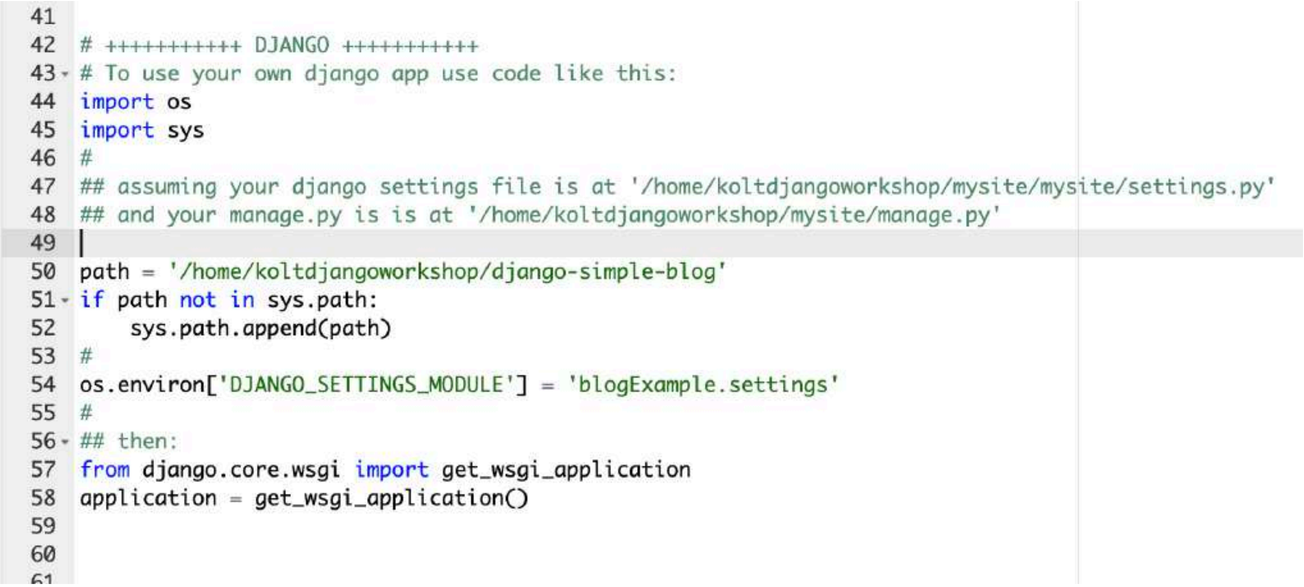
Log files are periodically rotated. You can find old logs here: /var/log

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/static/admin	/home/koltdjangoworkshop/django-simple-blog/venv/lib/python3.6/site-packages/django/contrib/admin/static/admin	🗑
Enter URL	*Enter path*	

Wsgi Configuration



```

# +++++++++++ DJANGO +++++++++++
# To use your own django app use code like this:
import os
import sys
#
## assuming your django settings file is at '/home/koltdjangoworkshop/mysite/mysite/settings.py'
## and your manage.py is is at '/home/koltdjangoworkshop/mysite/manage.py'

path = '/home/koltdjangoworkshop/django-simple-blog'
if path not in sys.path:
    sys.path.append(path)
#
os.environ['DJANGO_SETTINGS_MODULE'] = 'blogExample.settings'
#
## then:
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

References

Miro.Medium.Com, 2019, https://miro.medium.com/max/1538/1*oAvefOeC1V4InSgGqEpS0w.png

Files.Realpython.Com, 2019, https://files.realpython.com/media/Get-Started-With-Django_Watermarked.15a1e05597bc.jpg.

Developer.Ibm.Com, 2019, https://developer.ibm.com/developer/patterns/create-backend-for-frontend-application-architecture/images/backend-for-frontend-arch-diagram2.png