Django project

**Create your environment:**

------------------------------------------------------------------

| Mac/Linux: | python3 -m venv djangoPy3Env

-------------+----------------------------------------------------

| Windows: | virtualenv djangoPy3Env

>------------------------------------------------------------------

**Activate your environment:**

---------------------

copy

---------------------------------------------

| Mac/Linux: | source djangoPy3Env/bin/activate

------------------------------------------------------------------

| Windows (command prompt): | call djangoPy3Env\Scripts\activate

------------------------------------------------------------------

| Windows (git bash) : | source djangoPy3Env/Scripts/activate

------------------------------------------------------------------

**Install Django:**

(djangoPy3Env) Windows/Mac:| pip install Django==2.2

> cd python\_stack/django/django\_intro

django\_intro> django-admin startproject your\_project\_name\_here

django\_intro> cd your\_project\_name\_here

your\_project\_name\_here> python manage.py runserver

Go to localhost:8000 on your browser

Type ctrl-c to stop the server

your\_project\_name\_here> python manage.py startapp your\_app\_name\_here

**your\_project\_name\_here/your\_project\_name\_here/settings.py**

INSTALLED\_APPS = [

'your\_app\_name\_here', # added this line. Don't forget the comma!!

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]    # the trailing comma after the last item in a list, tuple, or dictionary is commonly accepted in Python

#### your\_project\_name\_here/your\_project\_name\_here/urls.py

from django.urls import path, include

urlpatterns = [

path('', include('your\_app\_name\_here.urls')),

]

#### your\_project\_name\_here/your\_app\_name\_here/urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.index),

]

#### your\_project\_name\_here/your\_app\_name\_here/views.py

from django.shortcuts import render, HttpResponse

*def* index(request):

return HttpResponse("this is the equivalent of @app.route('/')!")

your\_project\_name\_here> python manage.py runserver

python manage.py makemigrations

python manage.py migrate

python manage.py shell

from django.db import models

class Movie(models.Model):

title = models.CharField(max\_length=45)

    description = models.TextField()

    release\_date = models.DateTimeField()

    duration = models.IntegerField()

    created\_at = models.DateTimeField(auto\_now\_add=True)

    updated\_at = models.DateTimeField(auto\_now=True)

class Author(models.Model):

name = models.CharField(max\_length=255)

created\_at = models.DateTimeField(auto\_now\_add=True)

updated\_at = models.DateTimeField(auto\_now=True)

class Book(models.Model):

title = models.CharField(max\_length=255)

author = models.ForeignKey(Author, related\_name="books", on\_delete = models.CASCADE)

created\_at = models.DateTimeField(auto\_now\_add=True)

updated\_at = models.DateTimeField(auto\_now=True)

this\_author = Author.objects.get(id=2)

my\_book = Book.objects.create(title="Little Women", author=this\_author)

or

my\_book = Book.objects.create(title="Little Women", author=Author.objects.get(id=2))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

this\_author = Author.objects.get(id=2)

books = Book.objects.filter(author=this\_author)

or

books = Book.objects.filter(author=Author.objects.get(id=2))

class Book(models.Model):

title = models.CharField(max\_length=255)

created\_at = models.DateTimeField(auto\_now\_add=True)

updated\_at = models.DateTimeField(auto\_now=True)

class Publisher(models.Model):

name = models.CharField(max\_length=255)

books = models.ManyToManyField(Book, related\_name="publishers")

created\_at = models.DateTimeField(auto\_now\_add=True)

updated\_at = models.DateTimeField(auto\_now=True)

this\_book = Book.objects.get(id=4) # retrieve an instance of a book

this\_publisher = Publisher.objects.get(id=2) # retrieve an instance of a publisher

# 2 options that do the same thing:

this\_publisher.books.add(this\_book) # add the book to this publisher's list of books

# OR

this\_book.publishers.add(this\_publisher) # add the publisher to this book's list of publishers

this\_book = Book.objects.get(id=4) # retrieve an instance of a book

this\_publisher = Publisher.objects.get(id=2) # retrieve an instance of a publisher

# 2 options that do the same thing:

this\_publisher.books.remove(this\_book) # remove the book from this publisher's list of books

# OR

this\_book.publishers.remove(this\_publisher) # remove the publisher from this book's list of publishers

this\_publisher.books.all() # get all the books this publisher is publishing

this\_book.publishers.all() # get all the publishers for this book