

Django

Copyright © 2021 by
Robert M. Dondero, Ph.D
Princeton University

Objectives

- The lecture will provide *a taste of ...*
- The **Django** server-side web app framework
 - As a more powerful alternative to Flask
 - As a contrast with Flask

Goal

- Goal of this lecture
 - Give you enough information about Django to:
 - Help you decide if you want to use it
 - Help you get started with it

Agenda

- **Django overview**
- An example app
- Setting up
- Creating files and directories
- Creating the database
- Running the app
- The admin interface
- Concluding observations

Django Overview

**Jean “Django”
Reinhardt**
1910-1952

Jazz guitarist
and composer



Django Overview

- **Who:** Adrian Holovaty & Jacob Kaplan-Moss (Benevolent Dictators for Life)
- **When:** 1995
- **What:** A heavyweight Python server-side web framework
- **Why:** Help web developers do more, use good style, and avoid security vulnerabilities



Holovaty

Kaplan-Moss

Django Overview

Django characteristics:

"Django lets you build deep, dynamic, interesting sites in an extremely **short time**. Django is designed to let you focus on the **fun**, interesting parts of your job while easing the pain of the repetitive bits. In doing so, it provides **high-level abstractions** of common Web-development patterns, shortcuts for frequent programming tasks, and clear conventions on how to solve problems. At the same time, Django tries to **stay out of your way**, letting you work outside the scope of the framework as needed."

-- Holovaty and Kaplan-Moss

Django Overview

- Why study Django?
 - Popular
 - Different
 - Reasonable for COS 333 project

8

Django Overview

Why study Django?

Popular

Different

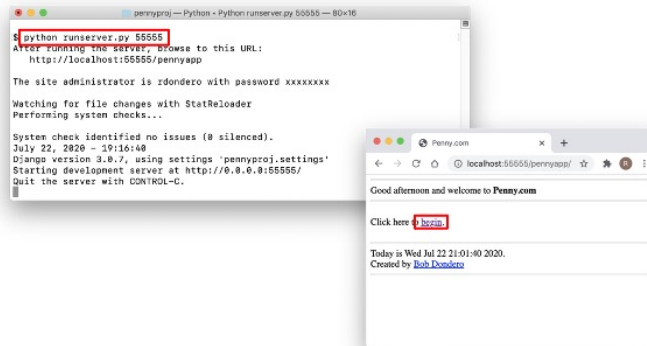
Reasonable for COS 333 project

Agenda

- Django overview
- **An example app**
- Setting up
- Creating files and directories
- Creating the database
- Running the app
- The admin interface
- Concluding observations

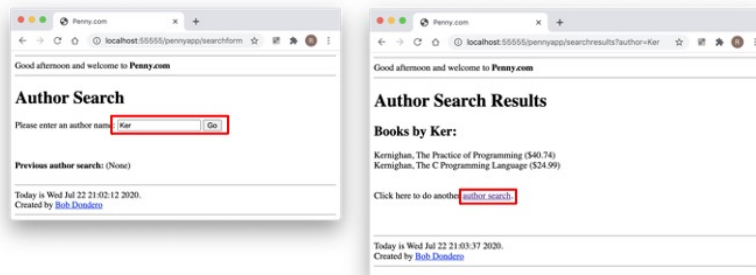
An Example App

- See **PennyDiango** app



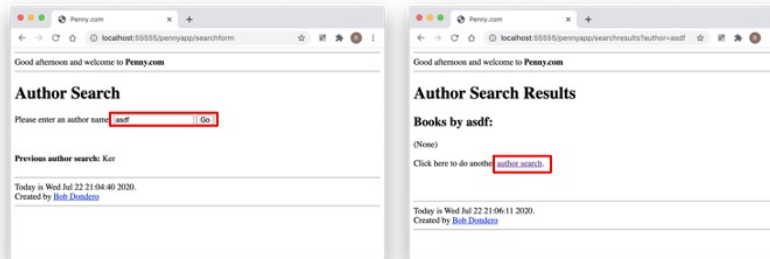
An Example App

- See **PennyDjango** app (cont.)



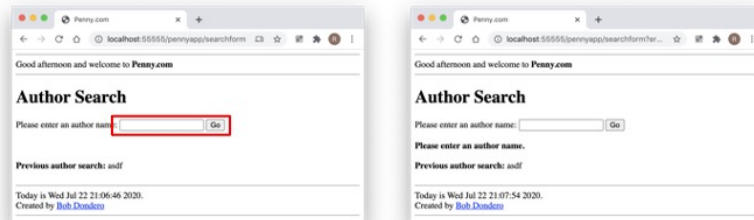
An Example App

- See **PennyDjango** app (cont.)



An Example App

- See **PennyDjango** app (cont.)



Agenda

- Django overview
- An example app
- **Setting up**
- Creating files and directories
- Creating the database
- Running the app
- The admin interface
- Concluding observations

Setting Up

- **Activate an appropriate Python virtual environment**
 - (OS specific)
- **Install Django**
 - `$ python -m pip install django`
- **Create and cd to some reasonable directory**
 - `$ mkdir PennyDjango`
 - `$ cd PennyDjango`

Setting Up

- **Create a Django project**
 - \$ `django-admin startproject pennyproj`
- **cd to your project directory**
 - \$ `cd pennyproj`
- **Create an application within the project**
 - \$ `python manage.py startapp pennyapp`

Agenda

- Django overview
- An example app
- Setting up
- **Creating files and directories**
- Creating the database
- Running the app
- The admin interface
- Concluding observations

Creating Files & Directories

Create/edit these files and directories:

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
  pennyapp/  
    __init__.py  
    admin.py  
    apps.py  
    models.py  
    templates/  
      pennyapp/  
        footer.html  
        header.html  
        index.html  
        searchform.html  
        searchresults.html  
    tests.py  
    urls.py  
    views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Boldface =>
Created or edited by Dondero

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Makes Django management
functionality available

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Dondero

SQL script
Used to populate DB

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Dondero

So need not remember
how to start server!

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Empty file
pennyproj is a package

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Unused

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited (2 lines) by Dondero

Specifies settings for pennyproj,
Esp. DB settings

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited (1 line) by Dondero

Maps pennyproj URLs to
pennapp URLs

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Connects project to WSGI

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Empty file
pennyapp is a Python package

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited by Dondero

Indicates that admin can
manipulate Book DB table

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Defines app name

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited heavily by Dondero

Defines DB schema

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Dondero

Define HTML templates
Note Django templating
language

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django

Unused

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited by Dondero

Maps pennyapp URLs
to functions

Creating Files & Directories

```
PennyDjango/  
  pennyproj/  
    manage.py  
    penny.sql  
    pennyapp/  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      templates/  
        pennyapp/  
          footer.html  
          header.html  
          index.html  
          searchform.html  
          searchresults.html  
      tests.py  
      urls.py  
      views.py
```

```
PennyDjango/ (cont.)  
  pennyproj/ (cont.)  
    pennyproj/  
      __init__.py  
      asgi.py  
      settings.py  
      urls.py  
      wsgi.py  
      runserver.py
```

Created by Django
Edited heavily by Dondero

Defines functions to handle
HTTP requests

Agenda

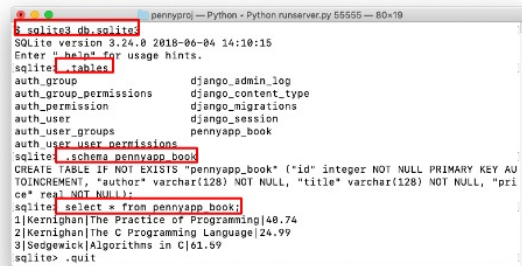
- Django overview
- An example app
- Setting up
- Creating files and directories
- **Creating the database**
- Running the app
- The admin interface
- Concluding observations

Creating the Database

- **Create the database**
 - \$ `python manage.py makemigrations`
 - \$ `python manage.py migrate`
 - \$ `sqlite3 db.sqlite3`
 - sqlite> `.read penny.sql`
 - sqlite> `.quit`

Creating the Database

- (Optional) Examine the database



```
pennysproj -- Python - Python runserver.py 55555 -- 80x19
$ sqlite3 db.sqlite
SQLite version 3.24.0 2018-06-04 14:10:15
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     pennyapp_book
auth_user_user_permissions
sqlite> .schema pennyapp_book
CREATE TABLE IF NOT EXISTS "pennyapp_book" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "author" varchar(128) NOT NULL, "title" varchar(128) NOT NULL, "price" real NOT NULL);
sqlite> select * from pennyapp_book;
1|Kernighan|The Practice of Programming|48.74
2|Kernighan|The C Programming Language|24.99
3|Sedgewick|Algorithms in C|61.89
sqlite> .quit
```

Note: DB has extra tables

Note: pennyapp_book table has extra fields

Agenda

- Django overview
- An example app
- Setting up
- Creating files and directories
- Creating the database
- **Running the app**
- The admin interface
- Concluding observations

Running the App

- Run the test server
 - `$ python runserver.py 55555`
- Run the application
 - Browse to
`http://localhost:55555/pennyapp`

Agenda

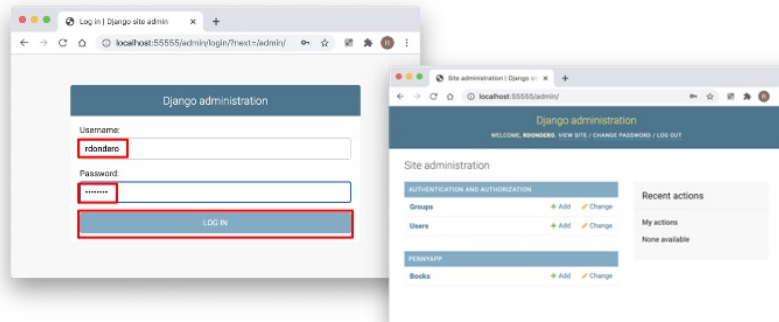
- Django overview
- An example app
- Setting up
- Creating files and directories
- Creating the database
- Running the app
- **The admin interface**
- Concluding observations

The Admin Interface

- Create a superuser
 - \$ `python manage.py createsuperuser`
 - Enter a username and password

The Admin Interface

- Run the *admin interface*
 - Browse to
`http://localhost:55555/admin`



Agenda

- Django overview
- An example app
- Setting up
- Creating files and directories
- Creating the database
- Running the app
- The admin interface
- **Concluding observations**

Concluding Observations

- Django:
 - Separates URL from file name
 - Provides a template engine
 - Provides an object-relational mapper (ORM)
 - Implements MVC architecture
 - Automatically provides an admin interface

44

Django Concluding Observations

Django separates URL from file name

Can use "pretty" URLs

Can map multiple URLs to same function

Can change function name without changing URL

Can change URL without changing function name

Django provides a template engine

Django = Flask + Jinja2

Can be used independent of Django framework

Django provides an Object-Relational Mapper (ORM)

Django = Flask + SQLAlchemy

Programmer need not know SQL

Django requires control of DB schema; realistic?

Django implements MVC architecture

Model (the database via the ORM)

Views (.html templates)

Controller (confusingly, views.py)

Django automatically provides an admin interface

- Allows site admins to manage users and groups

- Allows site admins to manage content of app's database tables

 - "A less powerful but more convenient version of your database's command-line utility"

- Not** a public interface to app's data

Summary

- The lecture has provided *a taste of...*
 - The **Django** server-side web app framework
 - As a more powerful alternative to Flask
 - As a contrast with Flask