Django

What is it?

- Python web framework
- Encourages rapid development
- A favorite for start ups

Who is using Django

Insta
Instagram

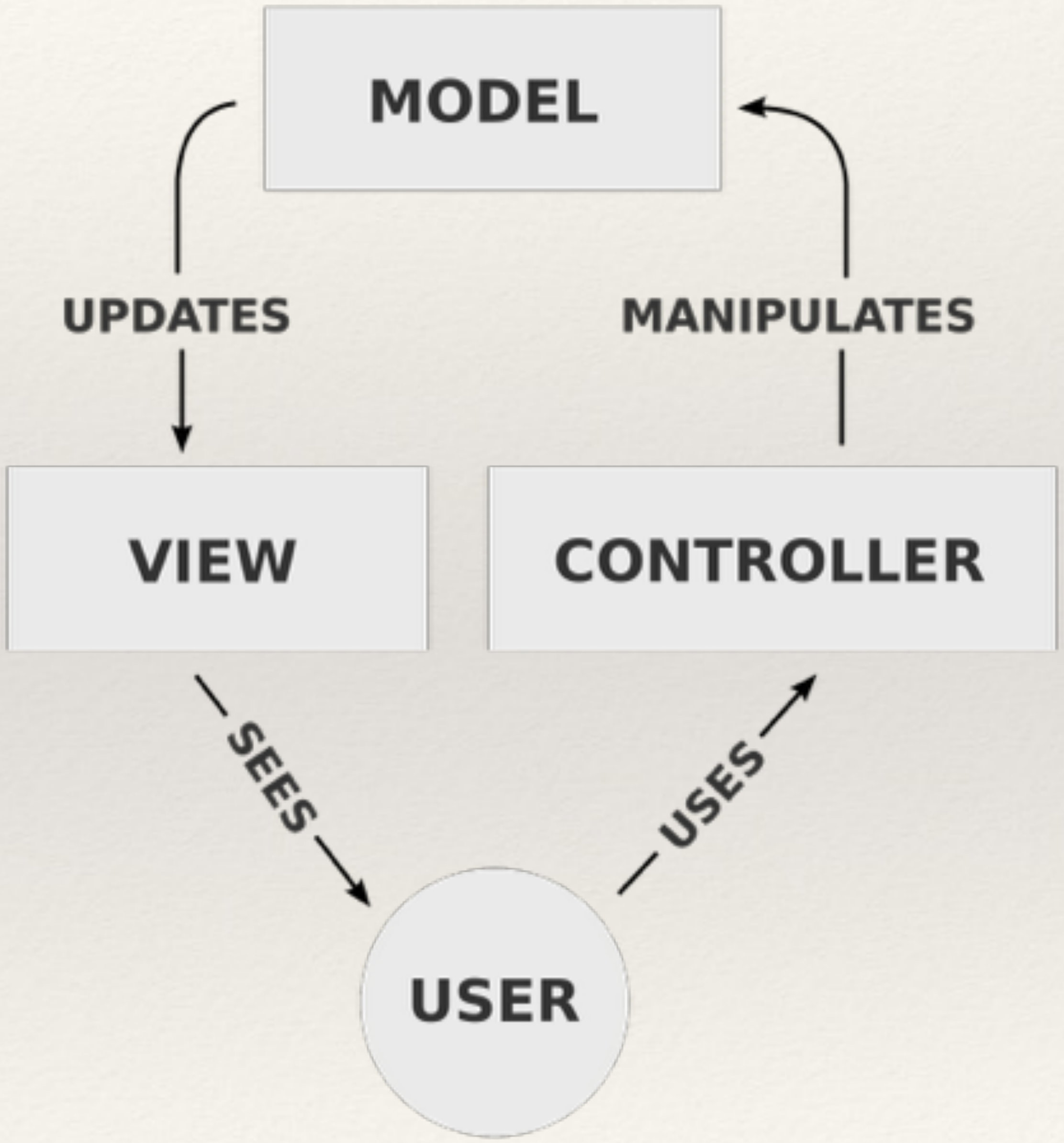
openstack™
CLOUD SOFTWARE

Why Use Django

- Easy to use framework
- Build a simple website in hours, not weeks
- Makes MVC (Model-View-Controller) more straightforward and easier to manage

Model-View-Controller

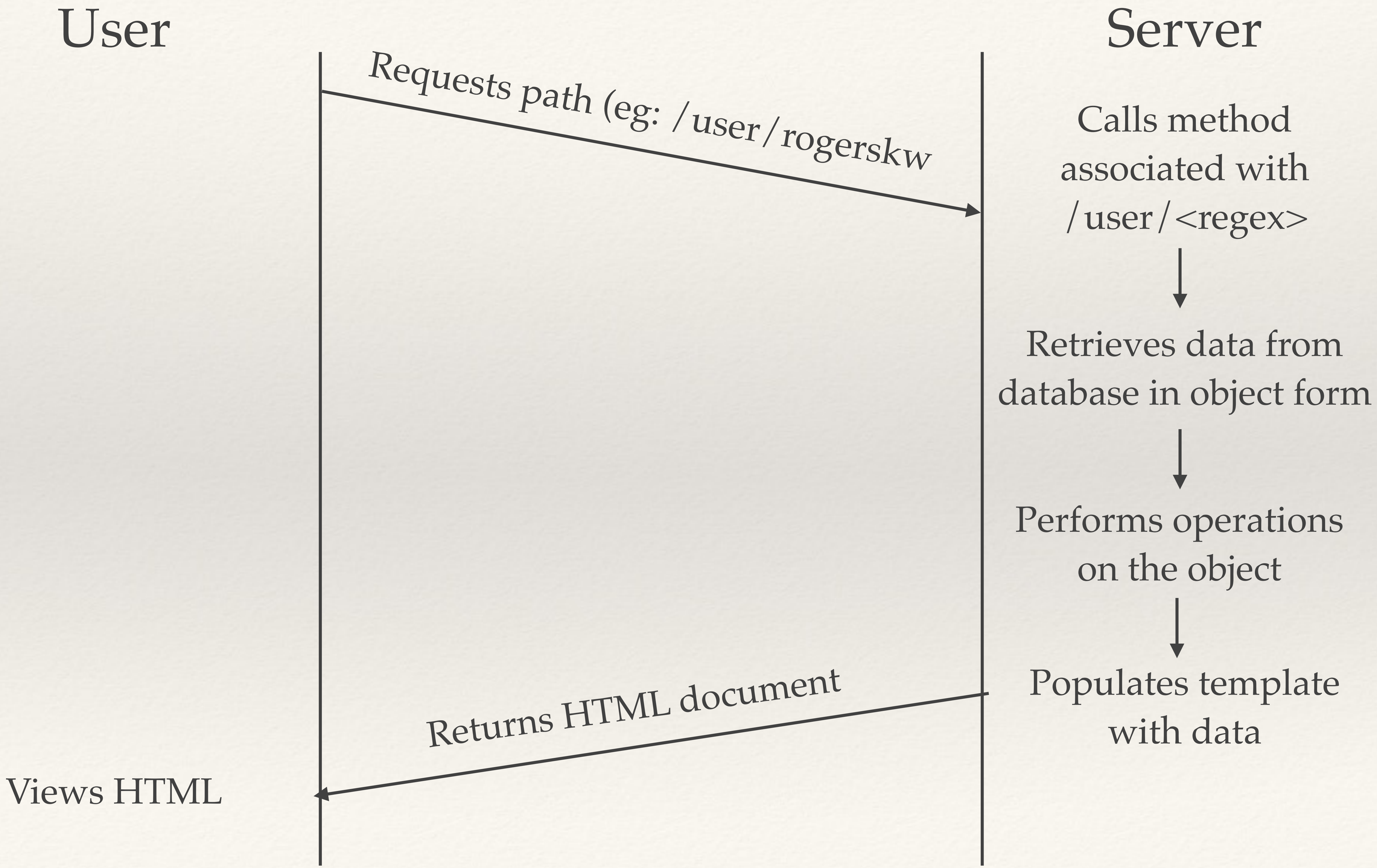
- Software architectural pattern for implementing user interfaces
- Model - Essentially the data
- View - a template that will be populated with data
- Controller - sends commands from the user to the model



MVC in Django

- Model - These are Python objects. Saved in a database as easily retrieved
- View - HTML templates. They will be combined with data to be sent to the user
- Controller - Code that is called by the user requesting a specific object

MVC in Django



User

Server

Requests path (eg: /user/rogerskw

Calls method associated with /user/<regex>

Retrieves data from database in object form

Performs operations on the object

Populates template with data

Returns HTML document

Views HTML

Disclaimer

Unfortunately, LNX01 does *not* have Django installed, so this will not be interactive

Let's Start

- Navigate to a directory where you will store your code
- Run:

```
$ django-admin.py startproject mysite
```

- Creates a new directory called `mysite`

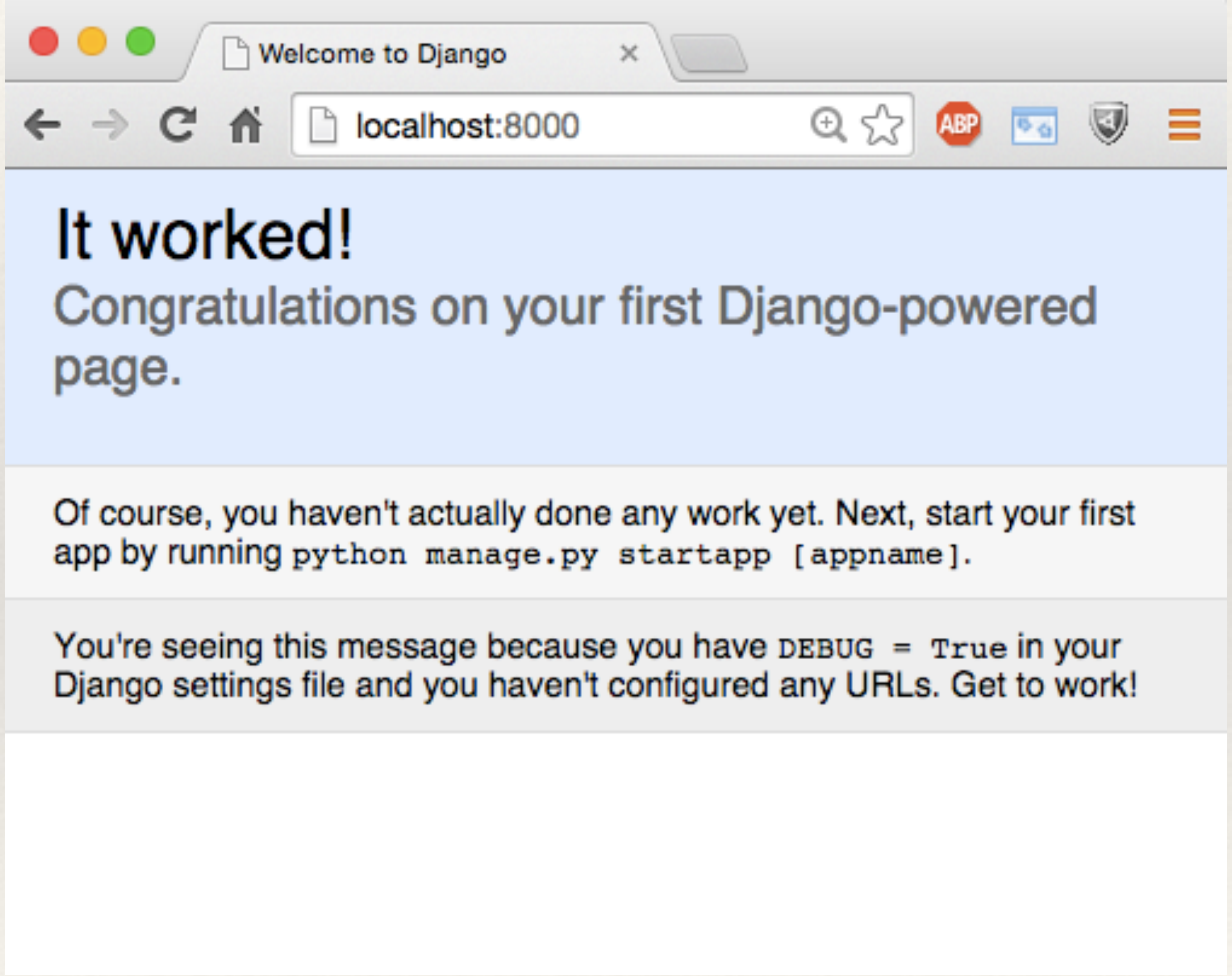
```
mysite/
    manage.py
    mysite/
        __init__.py
        settings.py
        urls.py
        wsgi.py
```

What Files are Created

- manage.py
  - This is how we will do admin ‘stuff’ throughout the development
- mysite/settings.py
  - This stores configurations for the project. Third party libraries, static URLs, database info and more are set here
- mysite/urls.py
  - This is where Django will look to see what method to call based on the HTTP request from the user
- mysite/wsgi.py
  - Web Server Gateway Interface - We will largely ignore this file

Test the Project

- We will use manage.py to test the code
  $ python manage.py runserver
- Open browser and navigate to localhost:8000
  - Django defaults to port 8000



It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Next, start your first app by running `python manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Make it do something…

- Obviously, that was not very exciting, so lets create a new ‘app’
- An app is a specific aspect of your website. Typically an app will consist of all pages for a given subdirectory
  - For example, if we were creating reddit, we might make a ‘subreddit’ app which would be all URLs with /r/<restofurl>
- Let’s create an app for conducting surveys

  ```
  $ python manage.py startapp polls
  ```

Our New App

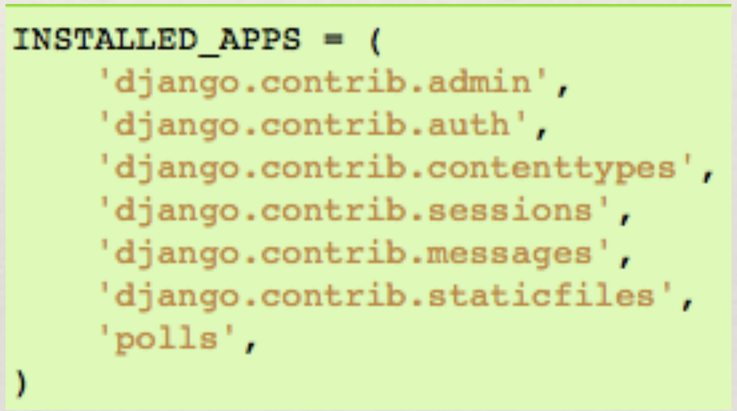
- Created a directory called 'polls'
- polls/
  views.py
  models.py
  test.py
  admin.py
- **models.py** - This is where we will store our classes. Objects for these classes can be saved to and retrieved from a database (specified in mysite/settings.py)
- **views.py** - This is where the code that creates the view is stored. Code here could also be considered the Controller to a certain extent.
- **test.py** - Unit tests are stored here. Good programmers will utilize tests to ensure that changes do not break anything.
- **admin.py** - You will need to tell Django when you have created a new model so that it can update the database.

Create Some Models

```
# mysite/polls/models.py

from django.db import models # Lets Python know where to find the necessary files

class Poll(models.Model): # We want to extend the Model class so that it has certain functionality
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    poll = models.ForeignKey(Poll) # Creates a pointer to a given Question object
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

Tell Django About the New App

- Add ‘polls’ to the INSTALLED_APPS set in mysite/settings.py
- Next, we need to prepare the SQL statement to create the tables
  $ python manage.py sql polls
- Now, execute the SQL statement
  $ python manage.py syncdb
- You will need to set up a superuser account



```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'polls',
)
```

Interactive Mode

- $ python manage.py shell
- Starts an interactive shell to do rapid testing
- Can retrieve objects from the database
- Create or modify objects and save them to the database

Do Stuff in Interactive Mode

```
>>> from polls.models import Poll, Choice

>>> Poll.objects.all()
 [ ] # No objects in the database yet

>>> from django.utils import timezone

>>> p = Poll(question=“Dinner?”,
 pub_date=timezone.now()) # Creates a new Poll object

>>> p.save() # Save the object to the database
 # Because it is new, it will create a new entry
```

Do Some More Interactive Stuff

```
>>> Poll.objects.all()
    [<Poll: Poll object>]

>>> p.id
    1

>>> p.question
    “Dinner?”
```

Admin Stuff

- Run the server again
- Go to localhost:8000/admin
    - This brings up the admin interface
    - From here we can do certain actions
- Enter superuser account info
- You will *not* see Poll or Choice listed

Make Poll and Choice Modifiable

```
# polls/admin.py

from django.contrib import admin
from polls.models import Poll
from polls.models import Choice

admin.site.register(Poll)
admin.site.register(Choice)

# You can now see Poll and Choice in the admin menu
```

Generate an HTML Page

```
# Add the following to polls/views.py
from django.http import HttpResponse

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

# We still need to map a URL to this method
```

Map a URL

```
# Modify polls/urls.py
from django.conf.urls import patterns, url
from polls import views

urlpatterns = patterns('',
    url(r'^$', views.index, name='index')
)
```

```
# Modify mysite/urls.py
from django.conf.urls import patterns, include, url

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    url(r'^polls/', include('polls.urls')),
    url(r'^admin/', include(admin.site.urls)),
)
```

Now Let's Check Out A Completed Website

github.com/ThetaTauMiami/ThetaTauMiami

thetataumiami.com