



# MIT

## Global Startup Labs

<http://aiti.mit.edu>

Indonesia Summer 2013  
Meetup 09 – Intro to Django



# Today's Meetup

- The Big Picture
- Django – A Web Application Framework
- Your First Django App
- Models in Django
- Today's Assignment

# The Big Picture

# So far we have an Android Frontend...



# What now?

We need to implement the backend:

- A web application framework, like
  - MonoRail, CppCMS, Apache Click, Grails, Spring, Stripes, multiple, Catalyst, CakePHP, Drupal, Symfony, CherryPy, Django, web2py, Ruby on Rails, Compujure
- A host, like
  - Heroku, Google App Engine

# The Big Picture



Web App Host

Your Django app

Android OS

Your Android app

# Django

## Web Application Framework

# Web Application Framework

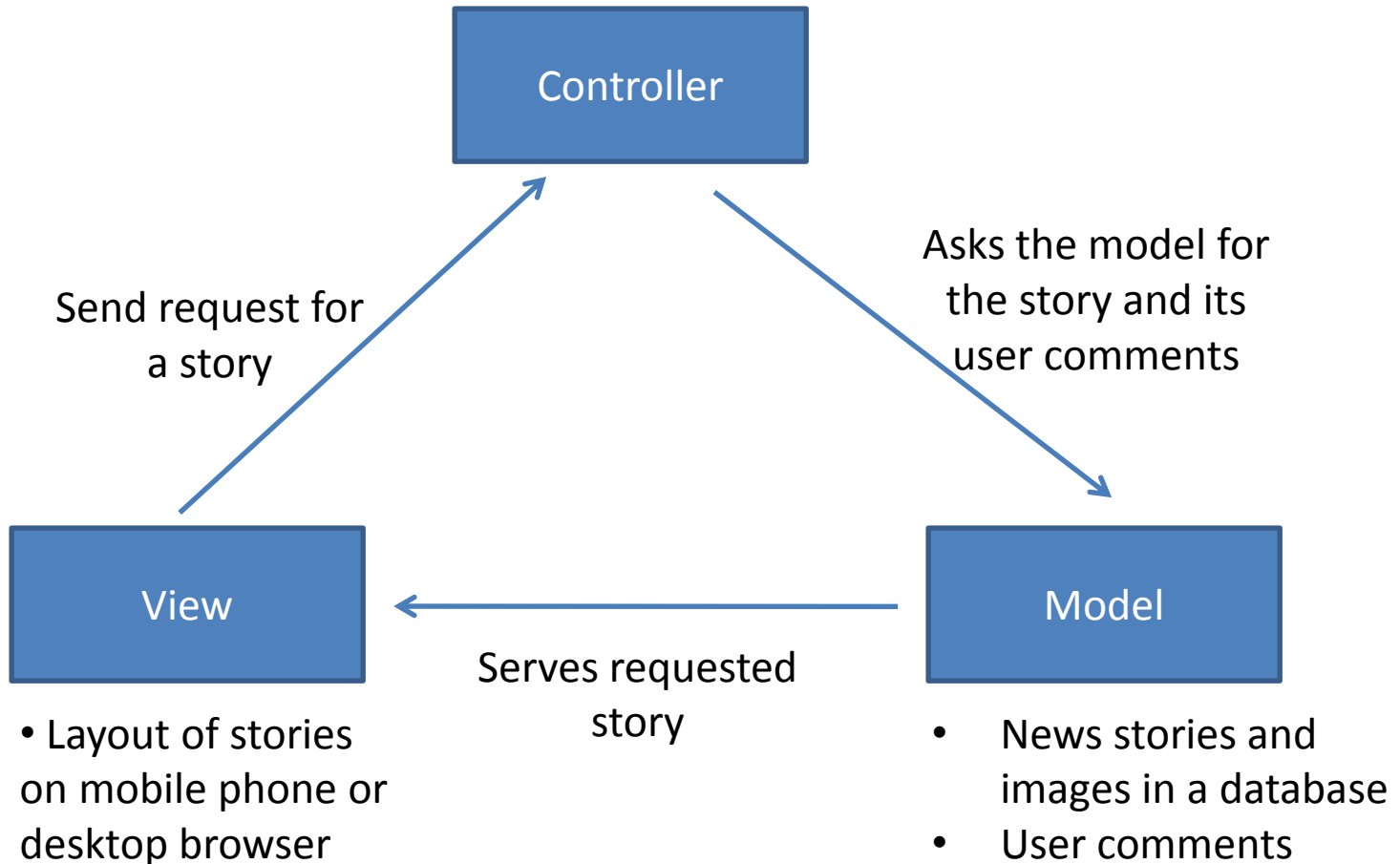
- A framework (code libraries) to help you make web applications or websites
- Supports you with
  - Handling HTTP requests
  - Templates for common HTML layouts
  - URL mapping
  - Database communication
  - Session management
  - Site security
- Allows you to focus on design and functionality rather than small details.



# Model-View-Controller (MVC)

- A pattern for organizing code often seen in web app frameworks
- Main idea is
  1. Separate the storage and manipulation of data (the model) and the presentation of data (view)
  2. Use the Controller to communicate between the model and view
- Advantages
  - Easier to develop and test model and view independently
  - Easier for others to understand

# Model-View-Controller (MVC) (news site example)



# What is Django?

- Web application framework, written in Python
- Released 2005
- Began with World Online, that needed to rapidly develop applications for news sites.
- Named after gypsie jazz guitarist Django Reinhardt (1910-53)
- Uses the MVC pattern, but with a twist on philosophy



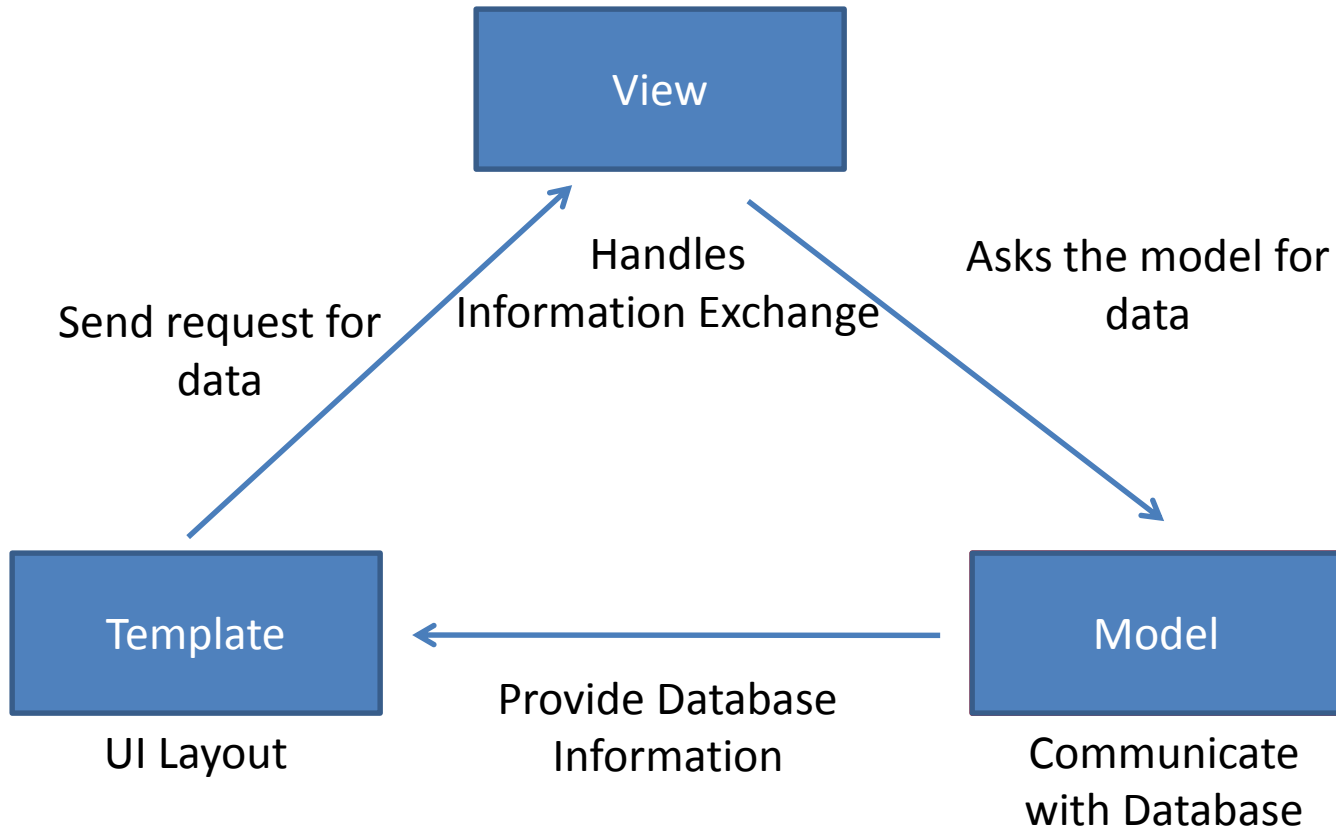
# Model-Template-View

In Django: Model-Template-View

(similar to MVC pattern)

- Model
  - describes database information
- Template
  - decides how to present information
- View
  - manages what information to output based on request

# Model-Template-View



# Why Django?

- Fast and easy development of web applications
  - Modular and re-useable. Don't Repeat Yourself (DRY) principle
  - BuiltIn SQLite support
- Active development and wide community support
- Successful Django sites <http://.djangosites.org/>
- Supported by Google App Engine & Heroku

# Your First Django App

# Installation

Follow the steps on:

<https://docs.djangoproject.com/en/dev/topics/install/>

- Skip Apache installation
  - For testing purposes built in webserver sufficient
- We will use SQLite as a database
  - No initial setup needed
- Install the official release Django 1.5.1
- Windows: Update your Path variable:
  - Add your Python directory
  - Add your django-admin.py directory



# Programming Interface

Command Promt/ Console

Python IDLE

```
Command Prompt
C:\Users\Markus\Documents\AITI\Programming Environment\DjangoWorkspace\nysite>python manage.py shell
IndentationError: unexpected indent (models.py, line 11)
C:\Users\Markus\Documents\AITI\Programming Environment\DjangoWorkspace\nysite>python manage.py shell
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
(InteractiveConsole)
>>> from polls.models import Poll, Choice
>>> Poll.objects.all()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: type object 'Poll' has no attribute 'Object'
>>> Poll.objects.all()
[<Poll: What's new?>]
>>> Choice.objects.all()
[]
>>> p = Poll.objects.get(pk=1)
>>> p.was_published_recently()
True
>>> p.choice_set.create(choice_text='Not much', votes=0)
<Choice: Not much>
>>> p.choice_set.create(choice_text='The sky', votes=0)
<Choice: The sky>
>>> p.choice_set.create(choice_text='The sky', votes=0)
<Choice: The sky>
>>> p.choice_set.create(choice_text='Just hacking again', votes=0)
<Choice: Just hacking again>
>>> c.poll
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'c' is not defined
>>> c = p.choice_set.create(choice_text='Just hacking again', votes=0)
>>> c.poll
[<Poll: What's new?>]
>>> p.choice_set.all()
[<Choice: Not much>, <Choice: The sky>, <Choice: The sky>, <Choice: Just hacking again>, <Choice: Just hacking again>]
>>> p.choice_set.count()
5
>>> exit()
C:\Users\Markus\Documents\AITI\Programming Environment\DjangoWorkspace\nysite>_

7% settings.py - C:\Users\Markus\Documents\AITI\Programming Environment\Django...
File Edit Format Run Options Windows Help
ADMINS = (
    # ('Your Name', 'your_email@example.com'),
)
MANAGERS = ADMINS
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3', # Add 'postgresql_psycopg2', 'my
        'NAME': 'C:\Users\Markus\Documents\AITI\Programming Environment\DjangoWo
        # The following settings are not used with sqlite3:
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': ''
    }
}
Ln: 123 Col: 50

7% models.py - C:\Users\Markus\Documents\AITI\Programming Environment\Django...
File Edit Format Run Options Windows Help
from django.db import models
import datetime
from django.utils import timezone

# Create your models here.

class Poll(models.Model):
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def __unicode__(self):
        return self.question

    def was_published_recently(self):
        return self.pub_date >= timezone.now() - datetime.timedelta(days=1)

class Choice(models.Model):
    poll = models.ForeignKey(Poll)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)

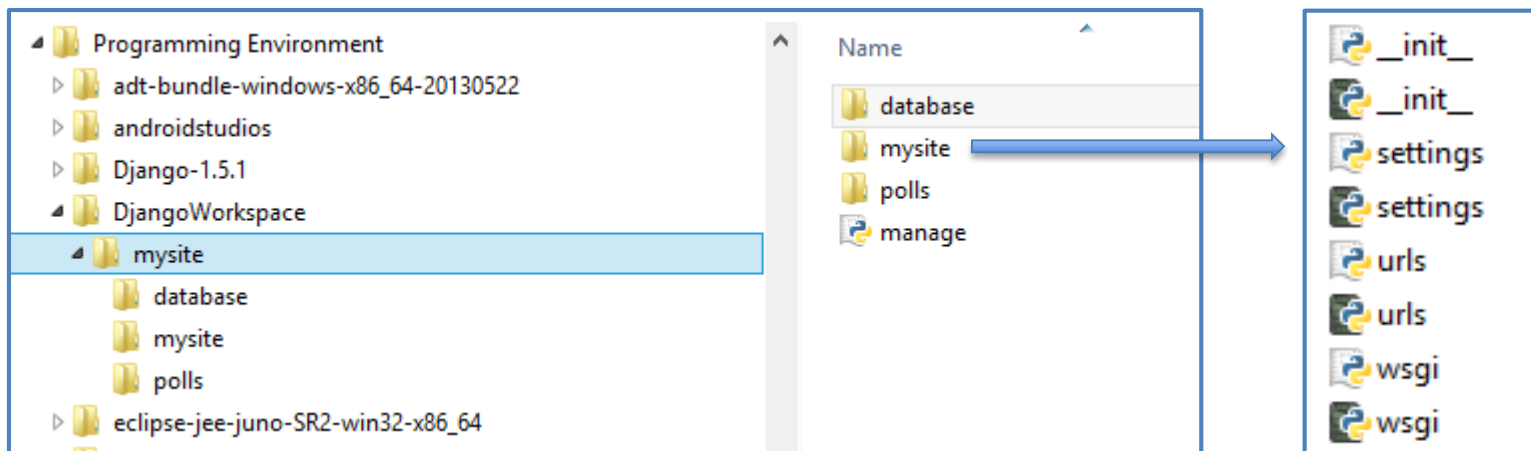
    def __unicode__(self):
        return self.choice_text
Ln: 24 Col: 0
```

# New Project

## Project Structure:

Whole project in one folder (mysite)

- MySite Python package → • MySite Python Package
- Applications (polls)
- (Database)
- manage.py
  - Use for interaction with your project
- \_\_init\_\_.py
- settings.py
- urls.py
- wsgi.py



# First Django Setup

Using the lightweight web server Django provides:

**it worked!**

Congratulations on your first Django-powered page.

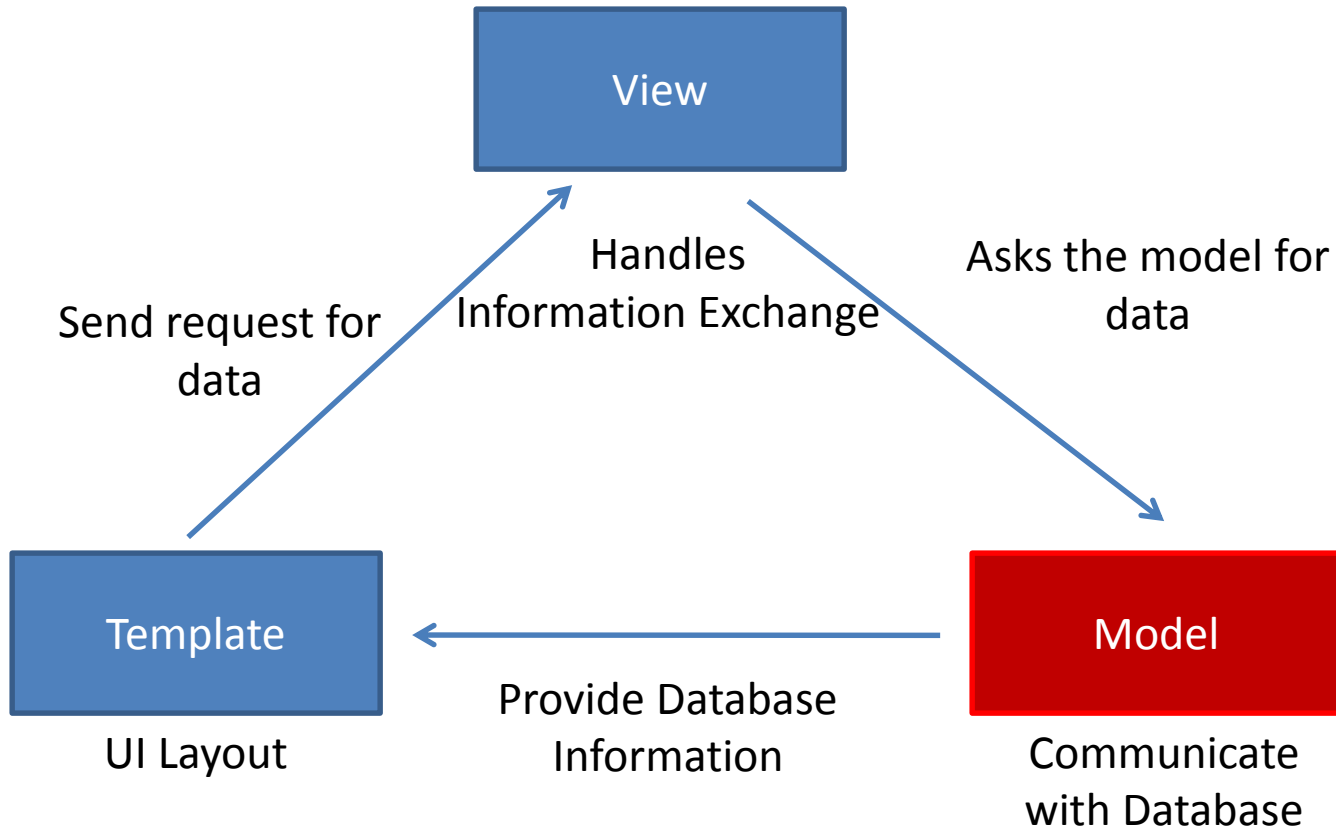
Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASES` setting in `mysite/settings.py`.
- 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!

# Models

# Models



# What is a Model

- A Python class describing data in your application
  - Subclass of `models.Model`
- Assigns attributes to each data field that is implemented
- Avoid direct work with the database
  - No need to handle database connections, timeouts, etc. Let Django do it for you.
  - Provides Schema for Database

# Django Model Syntax

Import statements

```
from django.db import models
import datetime
from django.utils import timezone
```

SubClass of  
models.Model Class

```
# Create your models here.
class Poll(models.Model):
```

Define fields

```
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
```

\_\_unicode\_\_  
corresponds to  
python \_\_str\_\_

```
    def __unicode__(self):
        return self.question
```

```
    def was_published_recently(self):
        return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
```

Can define more  
functions

# Django Fields

We can define fields directly in our model class

– No need to define manually in database

Example: create two fields in our Poll class

```
class Poll(models.Model):  
    question = models.CharField(max_length=200)  
    pub_date = models.DateTimeField('date published')
```

Define Type of Field

- E.g. `models.CharField`

Define arguments of field

- E.g. `max_length=200`

Django will automatically create fields in database



# Important Django Field Types

- BooleanField
  - Checkbox
- CharField(max\_length)
  - Single-line textbox
- DateField
  - Javascript calendar
- DateTimeField
  - Javascript calendar, time picker
- DecimalField(max\_digits, decimal\_places)
  - Decimal numbers
- EmailField
  - Charfield that validates email address
- FileField
  - File upload, stores path in database
- FloatField
  - Floating point numbers
- IntegerField
  - Integer textbox
- PositiveIntegerField
  - Integer textbos for positive integers
- TextField
  - Multi-line textbox

# Rules of Django Models

- When you update a model, ALWAYS RUN **python manage.py syncdb**
- All classes extend **models.Model**
- Models only live in **Apps**
- Django doesn't save objects until you call **save()** method

```
>>>a1 = Album(...)
```

```
# a1 is not saved to the database yet!
```

```
>>>a1.save()
```

```
# Now it is.
```

# Today's Assignment

# Today's Assignment

---

## Get started with Django!

- Install Django 1.5.1 on your workstation
- Set up your Python Environment
  - Use SQLite as your Database
- Create your first Django app:  
[Writing your first Django App – Part 1](#)

## Helpful Documentation:

- Chapters 1 - 3 of [The Django Book](#)