# Django

a high-level Python Web Framework

# Popular sites use Django

- NTHU OJ
- Disqus
- Pinterest
- Instagram
- Mozilla
- many newspaper websites (The Washington Post, UK's Guardian, The New York Times)

# Install

pip install django

# What is a Framework?

「框架就是制定一套規範或規則，
大家在該規範或規則下工作」

— 中文維基百科

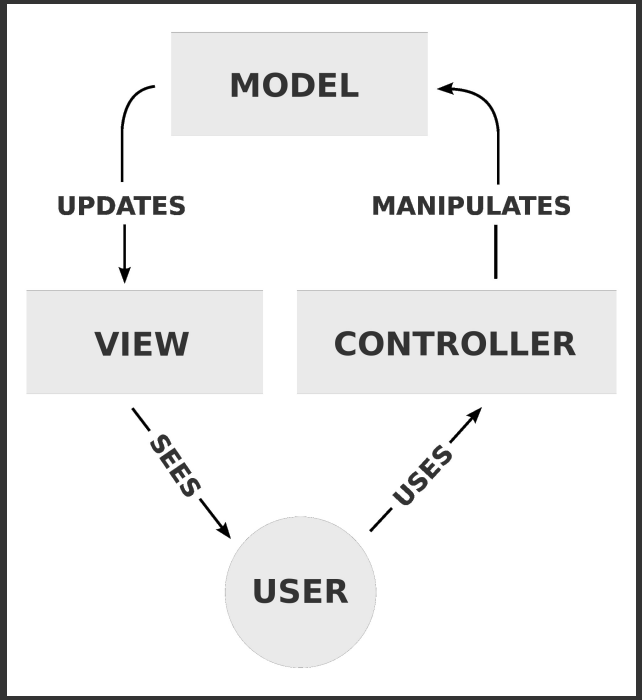
「就是使用別人搭建好的舞台，由你來表演！」

— 中文維基百科

# MVC

# MVC

- 常見的軟體架構模式
- 將軟體分成 Model, View, Controller 三個部份

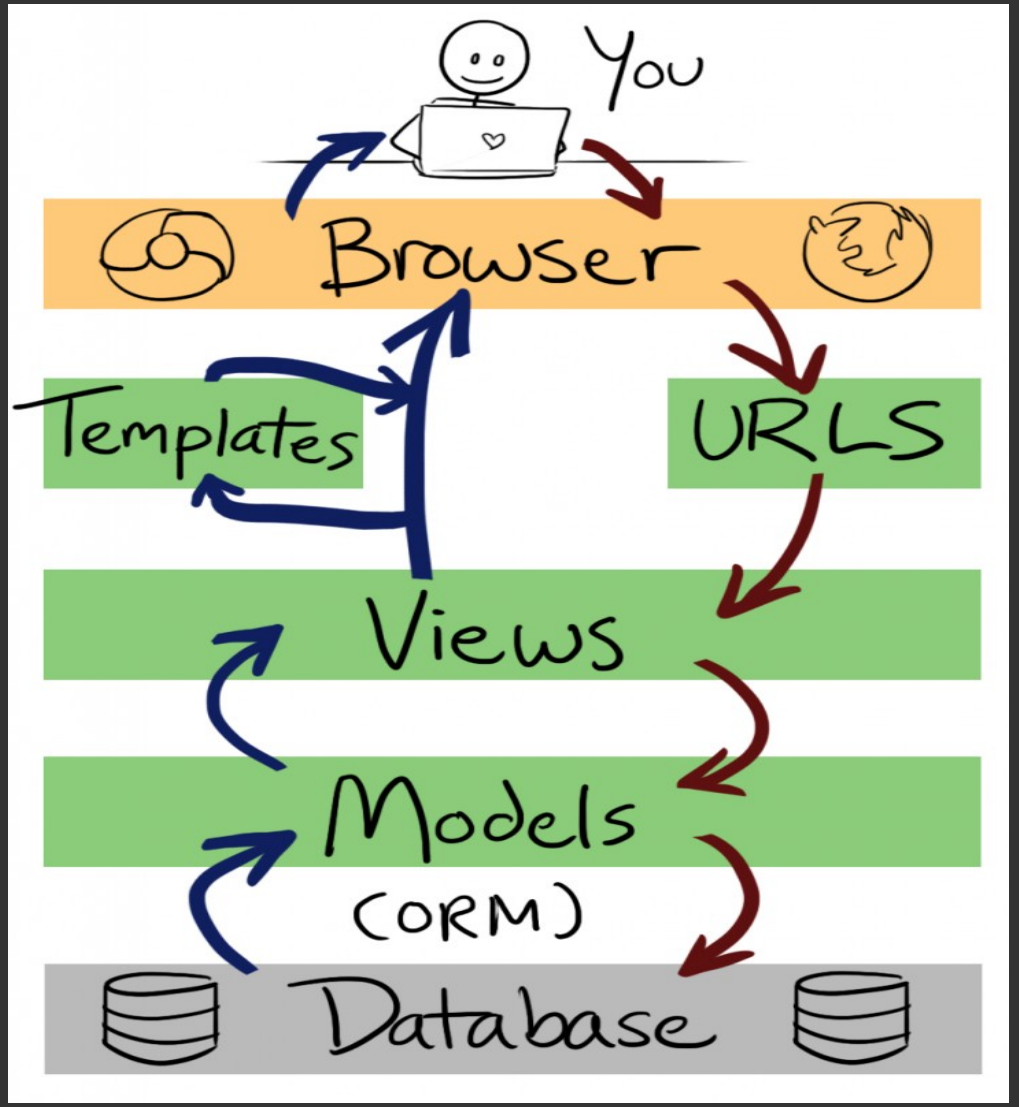


# MVC

- Model
  - 定義資料
- View
  - 呈現資料
- Controller
  - 操作資料的呈現方方式

# Django 的 MTV 架構

- 類似 MVC
- 分成 Model, Template, View
- 對應 MVC
  - Model: Model
  - Template: View
  - View: Controller


You
Browser
Templates
URLS
Views
Models
(ORM)
Database

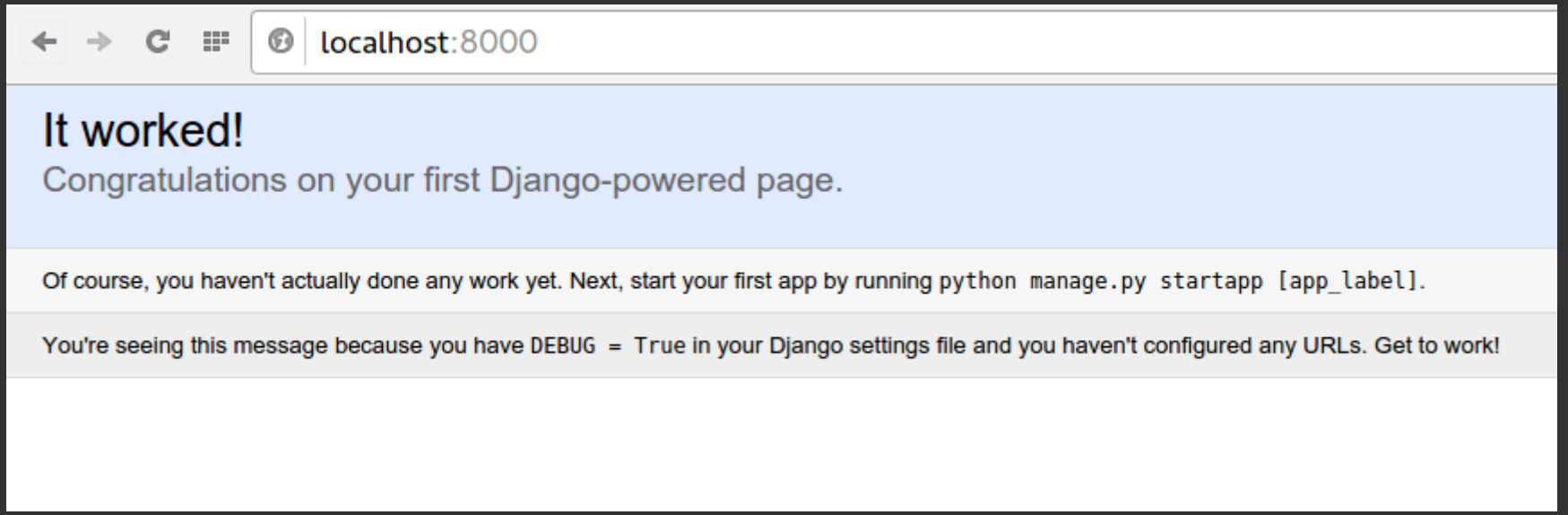
# Start a new project

- `django-admin startproject hello`

```
hello
├── hello
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py
```

# Try it

- `cd hello`
- `python manage.py runserver`



# Start a new app

- 一個 django project 是由許多小 app 所組成
- 可以上網找別人寫好的 app 加到自己的 project

# Start a new app

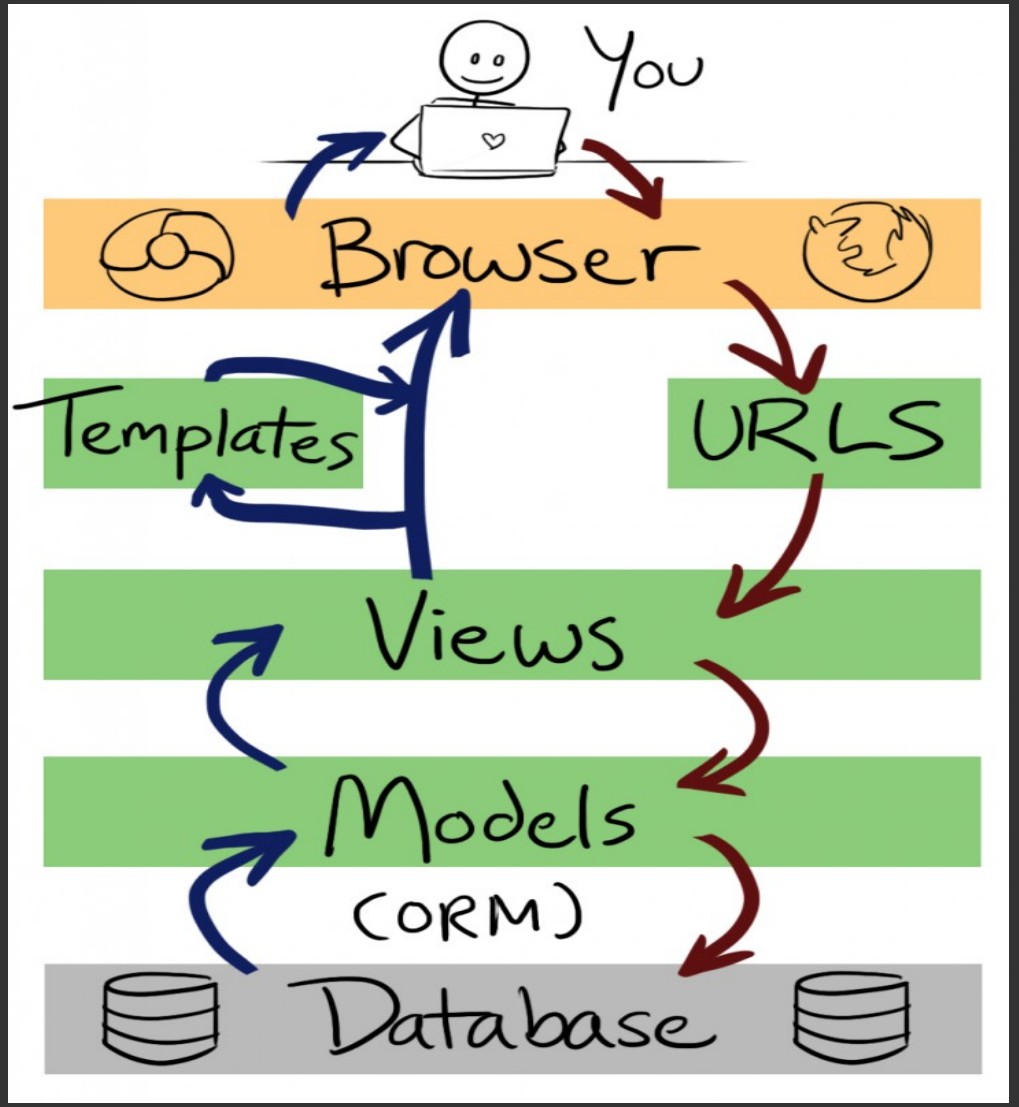
- `python manage.py startapp hihi`

```
hihi
├── admin.py
├── __init__.py
├── migrations
│   └── __init__.py
├── models.py
├── tests.py
└── views.py
```

# Add app to your project

- Edit hello/settings.py

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


You
Browser
Templates
URLS
Views
Models
(ORM)
Database

# views.py

- 接收 request
- 決定要取得什麼 model 的資料
- 處理資料
- 回傳 response

# views.py

- Edit hihi/view.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("hello world")
```

# Urls.py

- 解析網址
- www.abc.com/xxx/yyy/zzz
- 從第一個 / 後面開始解析
- 決定要呼叫哪一個 view

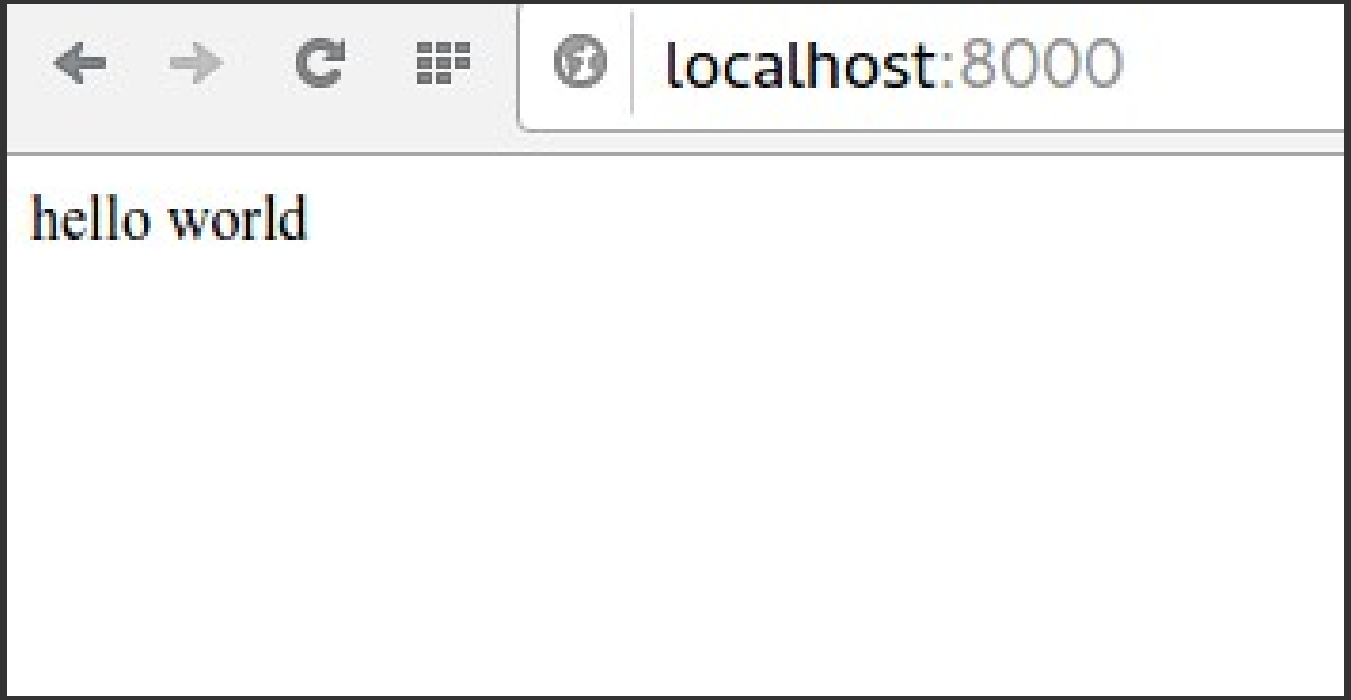
# Urls.py

- Edit hello/urls.py

```
from django.conf.urls import include, url
from django.contrib import admin
from hihi import views

urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
    url(r'^$', views.index, name=”index”),
]
```

```
python manage.py runserver
```



# Urls.py

- 全部塞在一起很難處理
- 可以用 include 的方式
- `url(r'^xxx/', include("xxx.urls"))`
- 假設有一個 url　/xxx/yyy/zzz/
- 拆成 /yyy/zzz/ 丟給 xxx/urls.py 來處理

# Urls.py

- 可以使用 regex
- 例如 `url(r'^(?P<yaya>\w+)', views.xxx, name=”xxx”)`
  - `()` 是一個 group
  - `?P<name>` 是給 match 到的字串的變數名稱
  - `\w` 是任何一個字元與數字以及 `'_'` ，意同 `[a-zA-Z0-9_]`
  - `+` 是一個以上

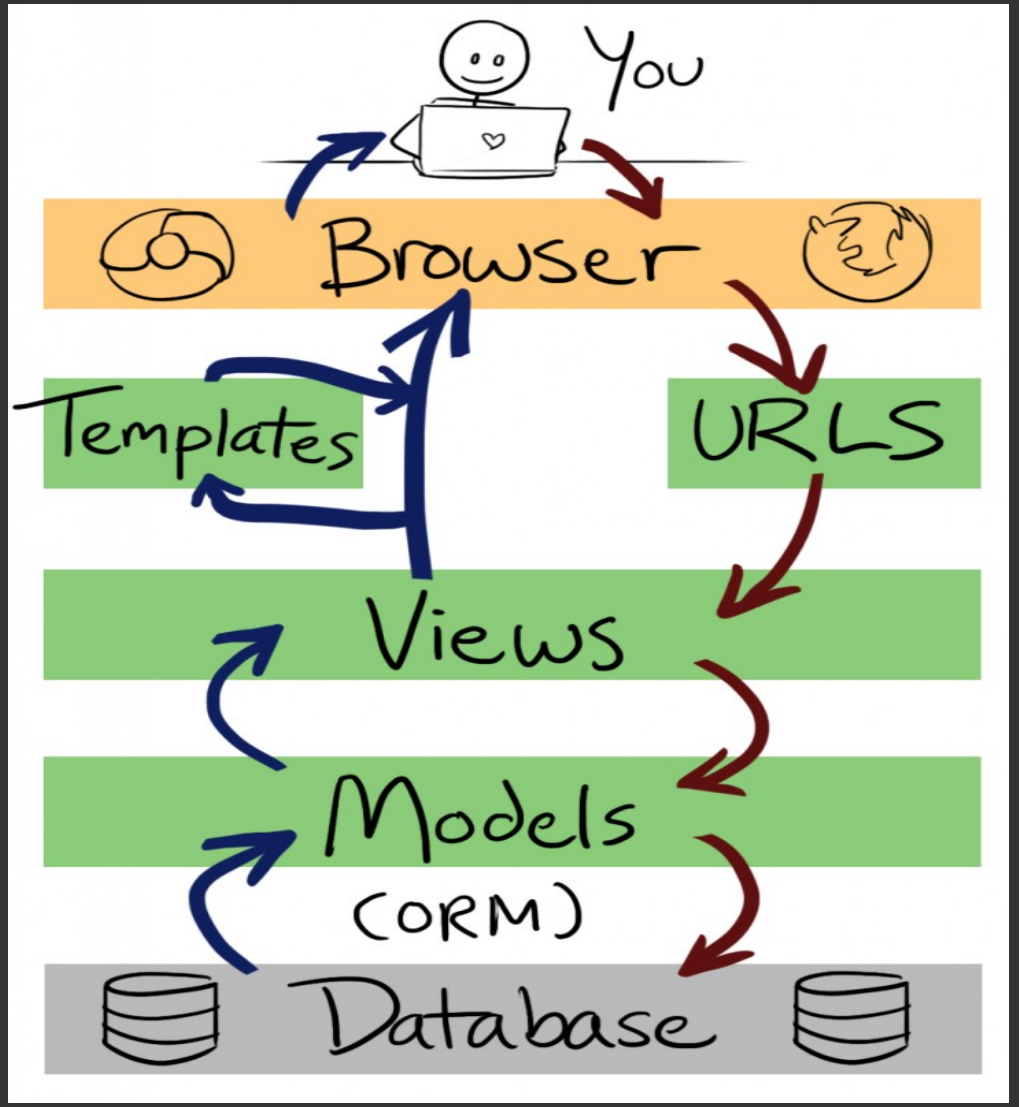
# Urls.py

- 在 view.py 裡面

```
def xxx(request, yaya):
    return HttpResponse(yaya)
```

# Urls.py

- Urls.py 的優點
  - 可以讓網址看起來很簡潔
  - 可以很容易的設計 restful api


You
Browser
Templates
URLS
Views
Models
(ORM)
Database

# Templates

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("<html>.....</html>")
```

# Create Templates

- 先在 app 的資料夾底下新增一個 tempates/app_name 的資料夾

```
hihi
├── admin.py
├── __init__.py
├── migrations
│   └── __init__.py
├── models.py
├── templates
│   └── hihi
│       └── index.html
├── tests.py
└── views.py
```

# Use Templates in View

```
from django.shortcuts import render
def index(request):
    return render(request, 'index.html')
```

# Template

- 樣板就是可以挖洞然後套用不同的東西
- 可以從 view 傳參數進去 template 裡面

# Template

```
from django.shortcuts import render
def index(request):
    return render(request, 'index.html',
                  {"msg": "Hello, world"})
```

# Template

```
<html>
  <body>
    {{ msg }}
  </body>
</html>
```

# Tags in template

- Templates 裡也可以放 if else 跟 for

```
{% if yaya %}
    <p>song la</p>
{% else %}
   <p>not song la</p>
{% endif %}
```

```
{% for x in xxx %}
   <p>{{ x }}</p>
{% endfor %}
```

# Static files

- Static files 就是 css, js, image 之類的檔案
- 在 app 裡面新增資料夾 static/app_name/
- 在 template 裡面使用的方式：

```
{% load staticfiles %}
<link rel="stylesheet" type="text/css" href="{% static app_name/xxx.css %}">
```

# Models

# Object Relational Mapping

- 以物件的方式操作資料庫
- 不用記複雜的 sql 語法， Django 幫你處理（爽）

# 連接資料庫

- 在 project_name/settings.py 裡面設定
- 預設是 sqlite ，一個簡易的資料庫

# Models.py

- 在 models.py 裡面定義物件
- django 就會幫你在資料庫裡面新增 table

# Models.py

編輯 hihi/models.py

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

# Models.py

- 執行 python manage.py makemigrations

```
Migrations for 'hihi':
  0001_initial.py:
    - Create model Post
```

# Models.py

- 執行 python manage.py migrate

```
Operations to perform:
  Synchronize unmigrated apps: staticfiles, messages
  Apply all migrations: admin, contenttypes, hihi, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... 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 hihi.0001_initial... OK
  Applying sessions.0001_initial... OK
```

# 操作 model

- 新增
  - `a = Song(x=”x”, y=”y”, z=”z”)`
  - `a.save()`
- 取得
  - `a = Song.objects.get(x=”x”)`
  - `aa = Song.objects.filter(x=”x”)`
  - `aaa = Song.objects.filter(x=”x”).order_by(“time”)`

# 操作 model

- 更新
  - a.x = “xxx”
  - a.save()
- 删除
  - a.delete()

# 更多操作

- Making queries in Django
  - https://docs.djangoproject.com/en/1.8/topics/db/queries/

# Models in template

編輯 hihi/views.py

```
from hihi.models import *

def show(request, pid):
    p = Post.objects.get(pk=pid)
    return render(request, "hihi/show.html", {"p": p})
```

# Models in template

編輯 `hihi/templates/hihi/show.html`

```
<html>
  <body>
    <p>{{ p.title }}</p>
    <p>{{ p.content }}</p>
  </body>
</html>
```

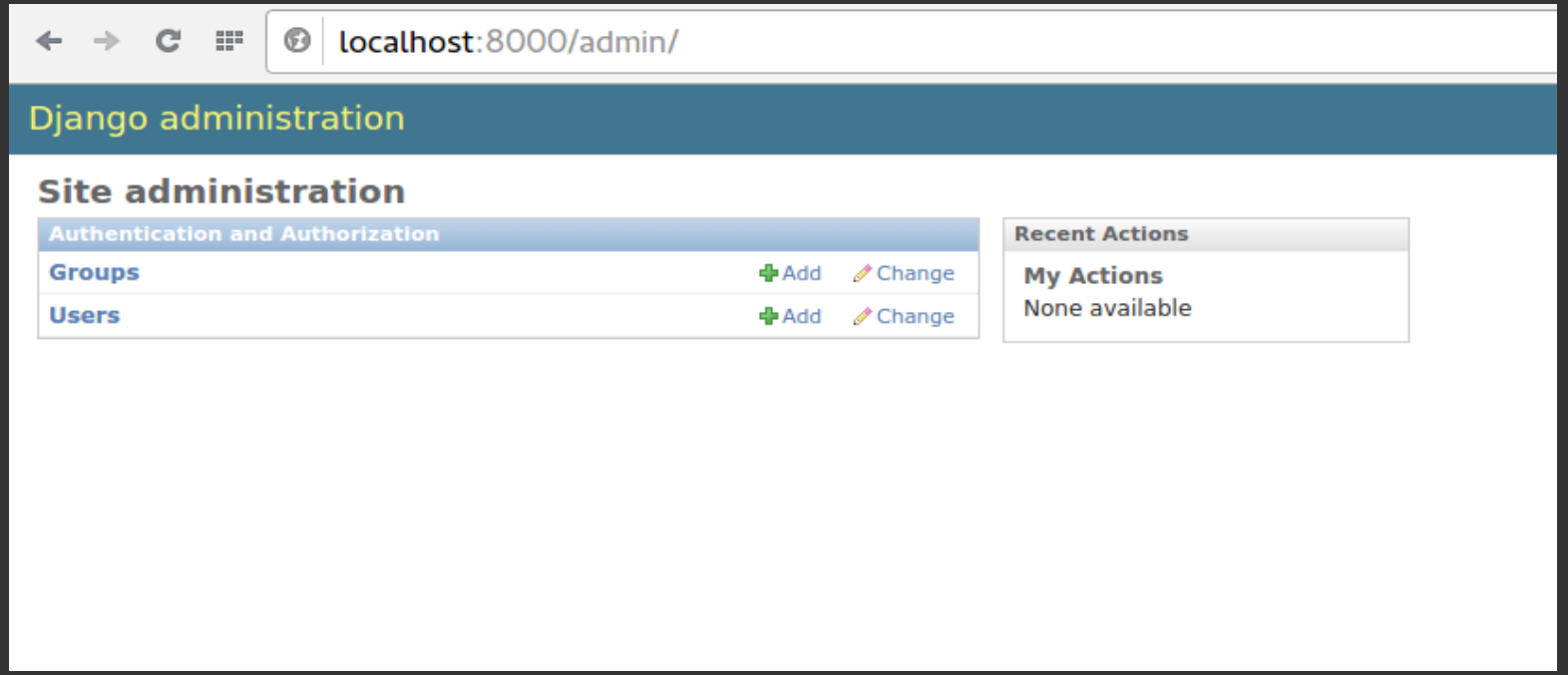
# Admin

# Admin

- 類似 phpMyAdmin
- 先創造一個 admin 帳號
- `python manage.py createsuperuser`

# Admin

- 進到 `localhost:8000/admin/`



# Admin

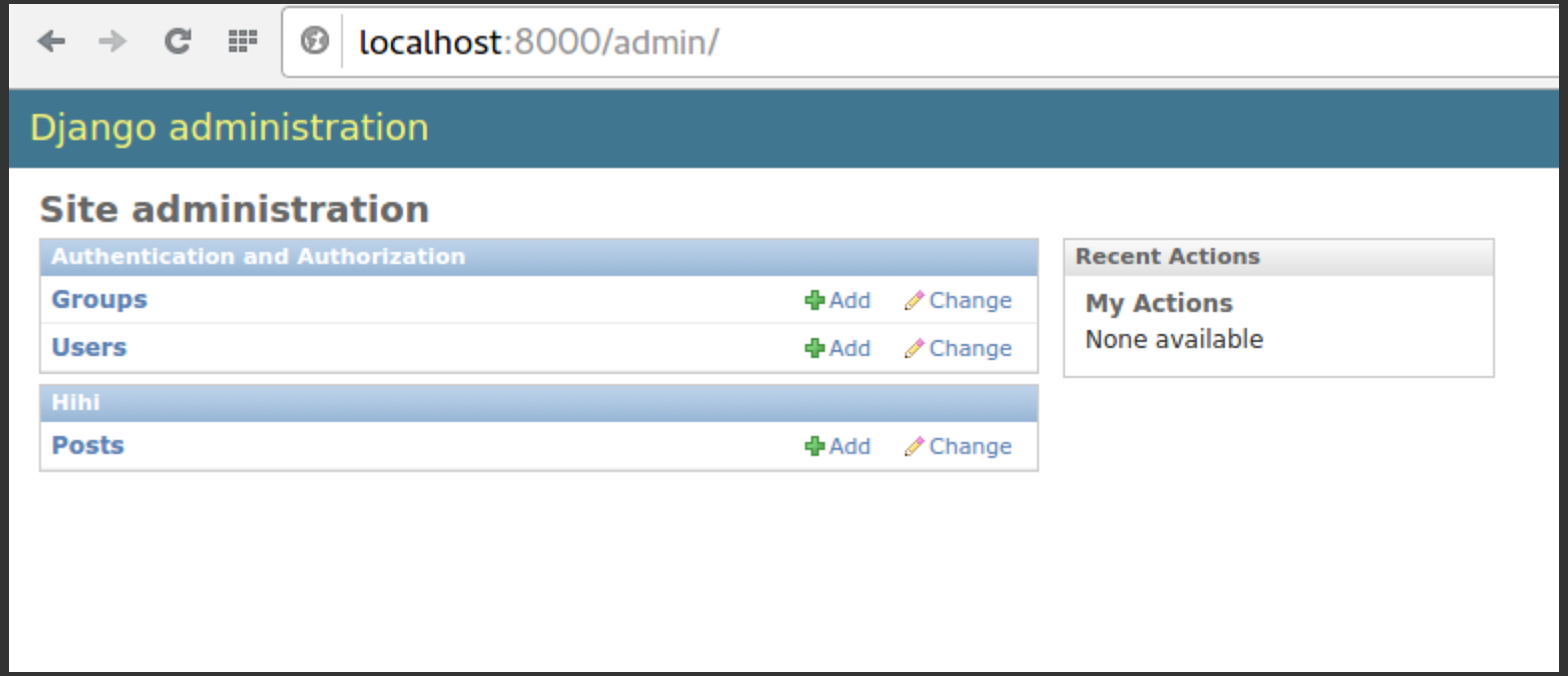
- 修改 hihi/admin.py

```
from django.contrib import admin
from hihi.models import *

admin.site.register(Post)
```

# Admin

- 進到 `localhost:8000/admin/`



# Forms

# Forms in django

- django 裡面也可以用物件的方式操作 form
- 可以直接用 model 來建立 form
- 方便！

# Forms in django

新增一個 hihi/forms.py

```
from django.forms import ModelForm
from hihi.models import *

class PostForm(ModelForm):
    class Meta:
        models = Post
```

# Forms in django

hihi/views.py

```
from hihi.forms import *
from django.shortcuts import render, redirect

def new_post(request):
    if request.method == 'POST':
        form = PostForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('/')
        else:
            return render(request, 'hihi/new_post.html', {'form': form})
    else:
        form = PostForm()
        return render(request, 'hihi/new_post.html', {'form': form})
```

# Forms in django

hihi/templates/hihi/new_post.html

```
<html>
  <body>
    <form method='post'>
      {% csrf_token %}
      {{ form }}
      <input type='submit' value='submit'>
    </form>
  </body>
</html>
```

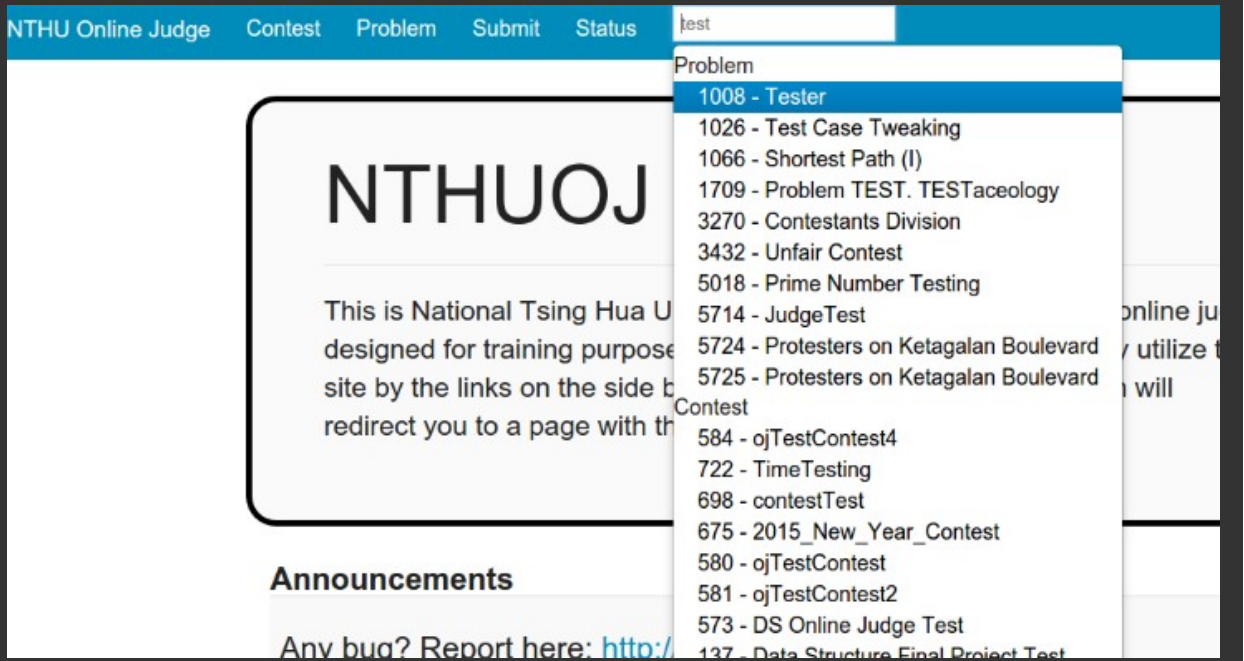
# User Athentication

- Django 已經寫好了 authentication 的機制
- User 的 model
- 登入登出
- 申請帳號，改密碼的 form
- 還有很多其他的
- 都不用自己寫
- OP!!!!

# DRY(dont repeat yourself)

# Reuse of App

- 網路上有超多別人寫好的 app 可以套用
- 例如： django-autocomplete-light



# Reference

Django 官網 ( 很完整 )