**0643010王儷蓉，0643011楊喻晴，0643024施欣佑**

**0643033潘玟晴，0643046洪于婷，0643047劉潔明**

按 ctrl+c 從part1部分進行至part2

* $ python manage.py makemigrations 創建更改的文件
* $ python manage.py migrate 將資料庫做更新
* $ python manage.py makemigrations創建更改的文件
* 在polls/models.py刪除原本內容，貼上:

from django.db import models

class Question(models.Model):

question\_text = models.CharField(max\_length=200)

pub\_date = models.DateTimeField('date published')

class Choice(models.Model):

question = models.ForeignKey(Question, on\_delete=models.CASCADE)

choice\_text = models.CharField(max\_length=200)

votes = models.IntegerField(default=0)

* 將polls.apps.PollsConfig文字貼在mysite/setting.py

INSTALLED\_APPS = [

'polls.apps.PollsConfig',

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

* $ python manage.py makemigrations創建更改的文件(創造了兩個model分別是choice.question)
* $ python manage.py migrate 將資料庫做更新
* $ python manage.py shell 進到資料庫裡
* >>> from polls.models import Question, Choice 將資料庫拖進來
* >>> Question.objects.all() 看Question的model里總共有哪些objects在裡面
* >>> from django.utils import timezone 創立一個新的資料
* >>> q = Question(question\_text="What's new?", pub\_date=timezone.now()) 開始寫入第一筆資料: What's new? 時間為系統時間
* >>> q.save() 將新資料儲存進去
* >>> q.id 找出q的ID
* >>> q.question\_text 會跑出的資料
* >>> q.pub\_date 文章發布時間
* >>> q.question\_text = "What's up?" 將q.question\_text 放進 "What's up?"
* >>> q.save() 將新資料儲存進去
* >>> Question.objects.all() show出目前objects有哪些
* >>> Question.objects.all()[0] 找值
* >>> Question.objects.all()[0]. question\_text
* 在polls/models.py裡的question下面新增code

def \_\_str\_\_(self):

return self.question\_text

* 在polls/models.py裡choice下面新增code

def \_\_str\_\_(self):

return self.choice\_text

* 在polls/models.py裡的最上面一行刪除from django.db import models 加入

import datetime

from django.db import models

from django.utils import timezone

* 在polls/models.py裡的question下面新增code

def was\_published\_recently(self):

return self.pub\_date >= timezone.now() - datetime.timedelta(days=1)

按ctrl+z離開

* python manage.py makemigrations 創建更改的文件
* python manage.py migrate將資料庫做更新
* python manage.py shell進到資料庫裡
* >>> from polls.models import Question, Choice 將model整合進去
* >>> Question.objects.all() 現在Question裡有哪些物件
* >>> Question.objects.filter(id=1) 找id=1是什麼
* >>> Question.objects.filter(question\_text\_\_startswith='What') 找出由What開頭的有那些
* >>> from django.utils import timezone import packages進來，目的為看現在的時間
* >>> current\_year = timezone.now().year 當年
* >>> Question.objects.get(pub\_date\_\_year=current\_year) 找出第一筆資料符合pub\_date\_\_year=current\_year
* >>> Question.objects.get(id=2) 找id=2是什麼
* >>> Question.objects.get(pk=1) primary-key=1會跑出
* >>> q = Question.objects.get(pk=1) 代表primary-key會跑出剛剛第一個sample
* >>> q.was\_published\_recently() 確定q的第一筆資料是否在一天內發表
* >>> q = Question.objects.get(pk=1) 將primary-key=1會跑出的值aside給q
* >>> q.choice\_set.all() 看Question裡到底有多少choice
* >>> q.choice\_set.create(choice\_text='Not much', votes=0) 新增一個choice
* >>> q.choice\_set.create(choice\_text='The sky', votes=0) 新增一個choice
* >>> c = q.choice\_set.create(choice\_text='Just hacking again', votes=0) 新增一個choice
* >>> c.question 問c對應到的question是什麼
* >>> q.choice\_set.all() 列出所有choice
* >>> q.choice\_set.count() 算出有幾個choice
* >>> Choice.objects.filter(question\_\_pub\_date\_\_year=current\_year) 過濾choice發表日哪些是今年的
* >>> c = q.choice\_set.filter(choice\_text\_\_startswith='Just hacking') 找出Just hacking 開頭的choice
* >>> c.delete() 刪除上者資料
* >>> q.choice\_set.count() 算出有幾個choice
* >>> q.choice\_set.all() 列出所有choice

按ctrl+z離開

* $ python manage.py createsuperuser 創造資料庫後端管理者

直接按enter

Email按enter

輸入密碼

* $python manage.py runserver 啟動開發伺服器
* 在網址那邊搜尋，刪除polls加上admin

輸入帳號密碼(密碼就是剛剛新創的密碼)

* 在polls/admin.py裡刪掉原本的加入

from django.contrib import admin

from .models import Question

admin.site.register(Question)

重新整理會出現question

**進入part3-1**

* 在polls/views.py下面貼上下列code記得不要刪

def detail(request, question\_id):

return HttpResponse("You're looking at question %s." % question\_id)

def results(request, question\_id):

response = "You're looking at the results of question %s."

return HttpResponse(response % question\_id)

def vote(request, question\_id):

return HttpResponse("You're voting on question %s." % question\_id)

* 在polls/urls.py裡面刪掉並貼上

from django.urls import path

from . import views

urlpatterns = [

# ex: /polls/

path('', views.index, name='index'),

# ex: /polls/5/

path('<int:question\_id>/', views.detail, name='detail'),

# ex: /polls/5/results/

path('<int:question\_id>/results/', views.results, name='results'),

# ex: /polls/5/vote/

path('<int:question\_id>/vote/', views.vote, name='vote'),

]

* 在polls/views.py

刪除:

from django.http import HttpResponse

def index(request):

return HttpResponse("Hello, world. You're at the polls index.")

新增:

from django.http import HttpResponse

from .models import Question

def index(request):

latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]

output = ', '.join([q.question\_text for q in latest\_question\_list])

return HttpResponse(output)

* 在polls裡新增一個資料夾 templates

templates裡再新增一個polls

然後在polls裡再新增一個index.html

貼上

{% if latest\_question\_list %}

<ul>

{% for question in latest\_question\_list %}

<li><a href="/polls/{{ question.id }}/">{{ question.question\_text }}</a></li>

{% endfor %}

</ul>

{% else %}

<p>No polls are available.</p>

{% endif %}

* 在polls/views.py

刪除:

from django.http import HttpResponse

from .models import Question

新增:

from django.http import HttpResponse

from django.template import loader

from .models import Question

* 在polls/views.py

刪除:

def index(request):

latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]

output = ', '.join([q.question\_text for q in latest\_question\_list])

return HttpResponse(output)

新增:

def index(request):

latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]

template = loader.get\_template('polls/index.html')

context = {

'latest\_question\_list': latest\_question\_list,

}

return HttpResponse(template.render(context, request))