首先要創造一個虛擬環境需要先創造一個空間，可防止版本不同、不相容。

打開terminal

→ sudo apt-get update

是將所有的package(套件）找出目前最新的版本，但沒有做更新

→ sudo apt-get install python3-pip

安裝第三版的pip

sudo：有時候在安裝指令的要有permission(授權)的狀況下才能安裝，如果授權被deny要加sudo這個關鍵字

ubuntu內建版本有兩個

python --version 版本為2

python3 --version 版本為3

→ pip3 install --upgrade pip

把pip的版本升級

→ pip3 --version

版本為9.0.1(python3.5)不要把3.5換成3.6會有bug

→ mkdir myproject

mkdir：make directory

開一個新的project

→ cd myproject

cd：change directory

跳到這個資料夾裡面

→ sudo pip3 install virtualenv

在這個裡面設立一個虛擬環境，把所有要用的套件全部裝進去，如果遇到PermissionError(授權的錯誤)要加上sudo

→ virtualenv VENV

開一個虛擬環境為VENV

→ source VENV/bin/activate

從這個路徑打開activate這個檔案

(VENV)所安裝的東西都會位於這個虛擬環境裡面

→ pip install django

一定要是第二版本

→ python

>>>import django

打import去call套件沒有錯誤就是安裝成功

按ctrl+z（暫停作業，稍後還可以恢復）跳開

→ django-admin startproject mysite

創建一個項目，第一次使用Django，需要些初始設置，您需要自動生成一些代碼來建立一個Django 項目 - 一個Django實例的設置集合，包括數據庫配置，Django特定的選項和特定於應用程序的設置。

~/myproject$ python manage.py runserver

會出現錯誤，因為manage.py在mysite而無法在myproject找到

→ cd mysite

→ python manage.py runserver

→ 此時開啟網頁 <http://127.0.0.1:8000/> ，出現火箭頭和 Congratulations！的頁面即成功。

按ctrl+c (終止一項工作，不能恢復過程。)離開

→ python manage.py startapp polls

開一個新的app叫做poll

→ 打開poll裡的views.py，從Django將code複製貼上

→ 在poll裡建一個新的檔案 urls.py ，打開後，從Django將code複製貼上

→ 打開mysite裡的urls.py，從Django將code複製貼上

→ python manage.py runserver

→ 此時開啟網頁 <http://localhost:8000/polls/>，要出現 Hello, world. You’re at the polls index.

--------------------------------------------part2--------------------------------------------------------------

按ctrl+c (終止一項工作，不能恢復過程。)離開

→python manage.py makemigrations

確認你的資料庫有哪些資料做了修改

→python manage.py migrate

更新你的資料庫 (記得最後一次一定要是 migrate)

接下來做的更新都是資料庫的更新。

→在polls/models.py下從djabgo教資料複製過來

→在mysite/settings.py裡面的INSTALLED\_APPS = []新增'polls.apps.PollsConfig',

→python manage.py makemigrations

檢查model是否有兩個修改

→python manage.py migrate

更新你的資料庫

→python manage.py shell 進入互動的shell模式

>>>是在shell的環境下

>>>from polls.models import Question, Choice

在model裡面加入Question, Choice

>>>Question.objects.all()

查看question裡面有那些資料

>>>from django.utils import timezone

加入timezone這個套件

>>>q = Question(question\_text="What's new?", pub\_date=timezone.now())

在Question這個table創造第一筆資料

在創造Question這個table必須明確告訴放什麼，有兩個是一定要放：question\_text、pub\_date，之後把這個值給q

>>>q.save()

創造的第一筆資料就被記錄進去了

>>>q.id

django本身忽略primary key這個東西，他是內鍵，因此用這個方式來了解主鍵是如何

>>>q.question\_text

會顯示剛剛寫入的文字What's new?

>>>q.pub\_date

會跑出時間，時間是根據timezone.now()

**>>>** q.question\_text = "What's up?"

把q這個物件的question\_text改成What's up

=等號是aside而不是等於，他是把What's up?這串字傳給question\_text

**>>>** q.save()

把資料存進去

>>>Question.objects.all()

把第一筆資料秀出來

Question到這邊結束

→在polls/models.py裡面新增一些指令(省略 太多ㄌ)

**from** **django.db** **import** models

**class** **Question**(models.Model):

*# ...* →保留

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

**return** self.question\_text

**class** **Choice**(models.Model):

*# ...* →保留

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

**return** self.choice\_text

目的→當你要去資料庫檢查一些內容的時候，會回傳東西

Str\_\_是兩個底線

→**import** **datetime**

**from** **django.db** **import** models

**from** **django.utils** **import** timezone

**class** **Question**(models.Model):

*# ...* **def** was\_published\_recently(self):

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

最近有沒有發表文章，而當你今天發表的文章是以今天為基準的一天內，就會回傳true，超過一天就會false

當要對資料庫做一些修正時

>>>Ctrl+z 先跳出，但不終止

→python manage.py makemigrations

檢查更新了什麼

→python manage.py migrate

更新你的資料庫

→python manage.py shell

進入到實作python的一個空間

Python ./manage.py shell在麻煩時這個指令比較好用

>>>**from** **polls.models** **import** Question, Choice

把model整合進去

>>>Question.objects.all()

查看有哪些物件

>>>Question.objects.filter(id=1)

做過濾的動作，第一個id為What's up?

**>>>**Question.objects.filter(question\_text\_\_startswith='What')

question\_text是由what開頭

>>> from django.utils import timezone 查看我現在的時間  
>>> current\_year = timezone.now().year 把今年當作current\_year  
>>> Question.objects.get(pub\_date\_\_year=current\_year)找出發表日跟current\_year的資料

>>> Question.objects.get(id=2) 出現錯誤，因為找不到ｉｄ＝２的資料

>>>Question.objects.get(pk=1)

只有一筆資料

>>> q = Question.objects.get(pk=1)

把值放進來  
>>> q.was\_published\_recently()

看q的第一筆資料是不是在一天內發文

>>> q = Question.objects.get(pk=1)  
>>> q.choice\_set.all()

>>> q.choice\_set.create(choice\_text='Not much', votes=0)

在Question創造第一個choice叫做Not much

>>> q.choice\_set.create(choice\_text='The sky', votes=0)

在Question創造第二個choice叫做The sky

>>> c = q.choice\_set.create(choice\_text='Just hacking again', votes=0)

在Question創造第三個choice叫做Just hacking again

>>> c.question

c這個choice所對應Question是什麼

>>> q.choice\_set.all()

總共有幾個

>>> q.choice\_set.count()

另外方法計算有幾個choice

>>> Choice.objects.filter(question\_\_pub\_date\_\_year=current\_year)

過濾choice發表日有哪些是今年的

>>> c = q.choice\_set.filter(choice\_text\_\_startswith='Just hacking')

尋找choice是just hacking開頭的

>>> c.delete()

→ctrl+z

→python manage.py createsuperuser (之後輸入帳號密碼)

→python manage.py runserver

點開網頁搜尋http://127.0.0.1:8000/admin/ 可以輸入帳號密碼之後代表成功

→在polls/admin.py貼上django的資料

-----------------------------------------------part 3------------------------------

→打開mysite/polls/views.py，從Django將code複製，增加上去，不可覆蓋到原本有的code

→打開mysite/polls/urls.py，從Django將code複製貼上

→打開mysite/polls/views.py，從Django將code複製，更改index以上的地方，detail、results、vote皆不變動

→在polls裡建一個資料夾templates

→在templates裡再建一個資料夾polls

→在polls裡開一個新檔案index.html

→打開檔案，從Django將code複製貼上

→打開mysite/polls/views.py，從Django將code複製，更改index以上的地方

→進入 <http://127.0.0.1:8000/polls/>，要出現What’s up? 點進去會出現You’re looking at question 1.

→進入<http://127.0.0.1:8000/polls/1/results/>，要出現You’re looking at the results of question 1.

→進入<http://127.0.0.1:8000/polls/1/vote/>，要出現You’re voting on question 1.

-----------------------------DJANGO PART3A shortcut: [render()](https://docs.djangoproject.com/en/2.0/topics/http/shortcuts/#django.shortcuts.render)---[-------------------------------](https://docs.djangoproject.com/en/2.0/intro/tutorial03/#a-shortcut-render)

→在polls/views.py裡面，用Django內的資料取代原本的from、index。

如何處理404error

→在polls/views.py裡面，用Django內的資料取代原本的from、detail資料。如此就不須又要再另外執行loader和HttpResponse。

→在polls/templates/polls裡開一個文件"detail.html"，之後在裡面貼入{{ question }}

→在polls/views.py裡面，用Django內的資料取代原本的from、detail資料。其中get\_object\_or\_404的功能為將Django模組視為第一參數兼關鍵字參數的隨機一數字，它會傳遞到此模組管理者的get()功能裡，因此如物件不存在，便會產生Http404。

以下三個程式碼會出現項目符號，網頁顯示結果會一樣

→<**h1**>{{ question.question\_text }}</**h1**>

<**ul**>

{% **for** choice **in** question.choice\_set.all %}

<**li**>{{ choice.choice\_text }}</**li**>

{% **endfor** %}

</**ul**>

<**li**><**a** href="{% **url** 'detail' question.id %}">{{ question.question\_text }}</**a**></**li**>

要去改它的url，change template code

<**li**><**a** href="{% **url** 'polls:detail' question.id %}">{{ question.question\_text }}</**a**></**li**>

因為有加app\_name，所以這段可以這樣寫

Part4

資料進去之後，外觀要改，要去改template

→<**h1**>{{ question.question\_text }}</**h1**>

{% **if** error\_message %}<**p**><**strong**>{{ error\_message }}</**strong**></**p**>{% **endif** %}

<**form** action="{% **url** 'polls:vote' question.id %}" method="post">

{% **csrf\_token** %}

{% **for** choice **in** question.choice\_set.all %}

<**input** type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />

<**label** for="choice{{ forloop.counter }}">{{ choice.choice\_text }}</**label**><**br** />

{% **endfor** %}

<**input** type="submit" value="Vote" />

</**form**>

會變成點選式項目符號

polls/results.html

到polls開一個新的檔案results.html

→<**h1**>{{ question.question\_text }}</**h1**>

<**ul**>

{% **for** choice **in** question.choice\_set.all %}

<**li**>{{ choice.choice\_text }} -- {{ choice.votes }}vote{{ choice.votes|pluralize }}</**li**>

{% **endfor** %}

</**ul**>

<**a** href="{% **url** 'polls:detail' question.id %}">Vote again?</**a**>

會顯示投票的數量

Part6  
首先在polls開新資料夾static，static裡面再開一個polls，polls裡面在開一個文件style.css 複製li a {  
 color: green;  
}  
然後去index call 你的index  
{% load static %}  
  
<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />  
  
圖要存到polls/static/polls/images/"圖名"."副檔名"裡面  
Polls/static再開一個images資料夾 然後圖檔放進裡面，按照圖片名稱加副檔名依序打上，再對路徑做更正。

Part7（基本上都是在更改admin的網頁外觀）

→修改polls/admin.py裡的資料，在admin網頁裡會增加一個欄位:Data Information

→修改polls/admin.py裡的資料，並且在底下新增（Choice)，在polls裡面會多出choice的欄位

→在polls/admin.py新增class ChoiceInline，可改變admin的外觀。之後修改class QuestionAdmin,兩個都修改完之後，就可以在新增問題時，一次輸入choice及對應的question的資料了

→修改polls/admin.py裡的class ChoiceInline(admin.TabularInline):內建的一些套件會讓你的網頁看起來更加的美觀

→修改polls/admin.py裡的class QuestionAdmin(admin.ModelAdmin)，

並新增list\_display = ('question\_text', 'pub\_date', 'was\_published\_recently')就會多出"問題"、"發布日期"跟"這個問題是否是最近發布的"。

→修改polls/models.py裡的def was\_published\_recently(self):

→ctrl+c跳出虛擬環境

→python manage.py makemigrations   
→python manage.py migrate

→python manage.py runserver

→在polls/admin.py的list\_display下面貼上list\_filter = ['pub\_date']，網頁的側邊會出現一個過濾器，更方便找到自己要的東西

→在polls/admin.py的list\_filter = ['pub\_date']下面貼上search\_fields = ['question\_text']就可以增加搜索功能

→在mysite/settings.py裡面修改整個templates

>>>>>>>這邊是出現Polls Administration之後如何改成能點開的網頁。

→在myproject/mysite(此時看的到manage.py)開一個"templates"資料夾

→在myproject/mysite/templates開一個"admin"資料夾

→點選VENV這個資料夾搜尋"Django"，之後沿著django/contrib/admin/templates/admin找到base\_site.html這個文件，將它複製貼上到剛剛新增的"admin”資料夾裡面

→之後將裡面的{% **block** branding %}~{% **endblock** %}的資料全部改成django裡面的新資料。

之後就是你這個作業完成了 ㄏㄏ