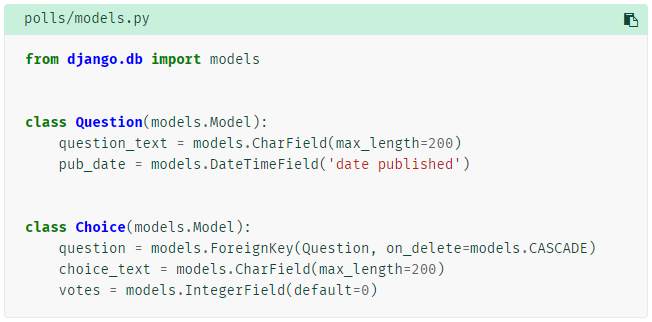
**Django app, part 2**



此行程式碼的目的為將model同步到資料庫，一般為先鍵入python manage.py makemigrations(建立更動後的資料檔)完再做python manage.py migrate。



從django.db中引入models

在poll這個app裡面，我們創建兩個model：Question、Choice，Question裏面包含了問題內容(設定為文字類別的資料，並限制長度最大值為200)、發布的時間(設定為日期格式)，Choice裡面有兩個欄位，choice的文字(設為文字類別的資料型態，並限制長度最大值為200)及相對應的vote(設定為整數的資料型態，並且預設值為0)，每個Choice皆與Question有關聯(使用ForeignKey，與Choice有關聯的為Question這個model，設定為on\_delete=models.CASCADE，代表設定當有關聯的Question遭刪除時，便將對應的Choice刪掉)。



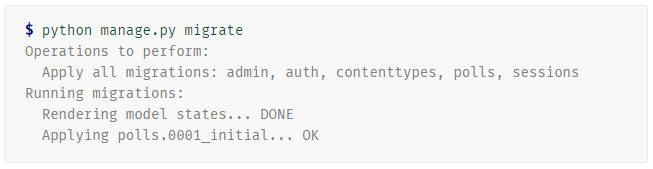
為了要將我們新增的poll這個app包含在project中，需要再mysite的settings.py的INSTALLED\_APPS加上polls.apps.PollsConfig。



makemigrations為建立一個更動model後的資料檔，執行此行程式碼後，django將會列出此次model的更動在哪些地方，第一次更動後將會產生一個0001\_initial.py的資料檔。



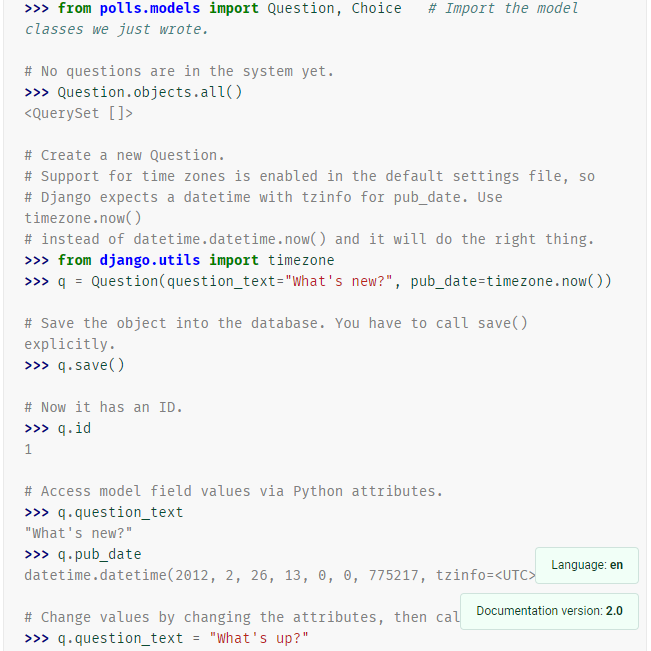
使用sqlmigrate可查看更動的一些細節。



執行此行程式碼將更動後的資料同步到資料庫。



進入python shell中，此為一互動式介面，可以直接執行指令。



重新給定question\_text的內容，並儲存。

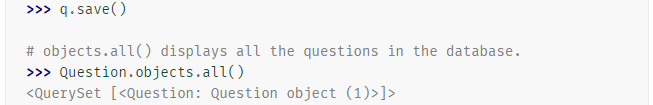
取得先前輸入的資料，因此鍵入對應的欄位名稱。

從django的utils這個套件中，引入timezone的功能，指定Question的文字為what’s new?、pub\_date為當前時區之時間。

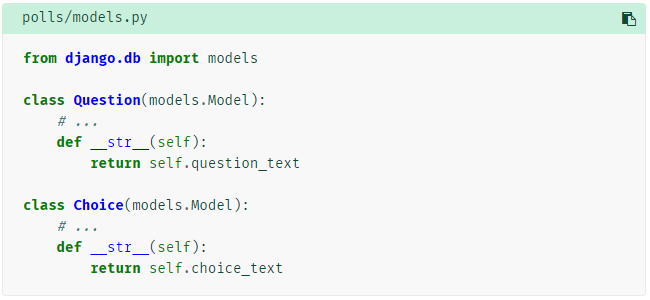
使用save()存取資料，此資料會有自己的id。

查詢全部的Question，但因為還未建立任何資料，因此不會有資料出現。

從poll的model中引入Question、Choice



這時候再使用objects.all()便可得知目前裡面有1項物件。



在Question、Choice皆新增一方法，並設定回傳值為其文字(\_\_str\_\_表示資料型態為字串)。



另外再引入datetime及從utils套件中引入timezone功能，並定義was\_published\_recently的回傳值。



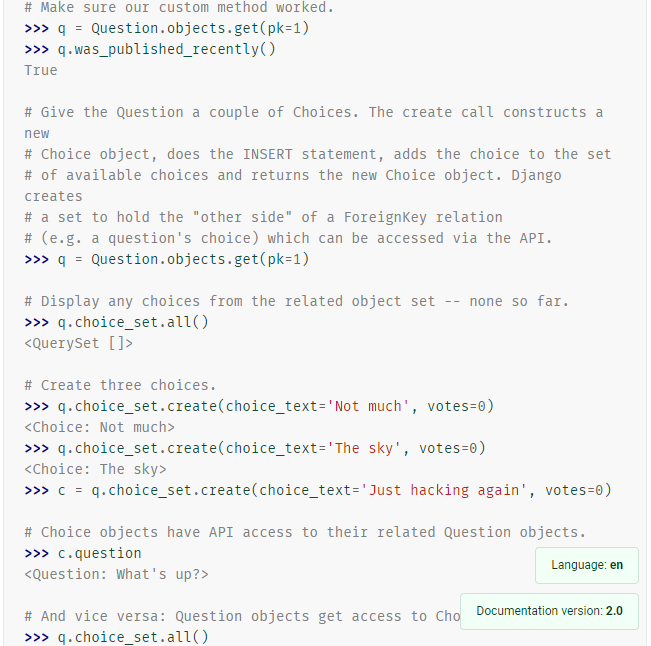
也可使用primary key取得資料。

取得id=2的資料，但是因為資料庫中並無此資料因此會有下列回傳訊息。

引入timezone，指定current\_year為時間中的年份，並取得與當年所發布的Question。

使用filter方法，指定id，可得到此筆資料內容，或是鍵入起始文字得到相對應資料。

再度進入python shell中，使用objects.all()將會看到回傳值已變為字串型態。



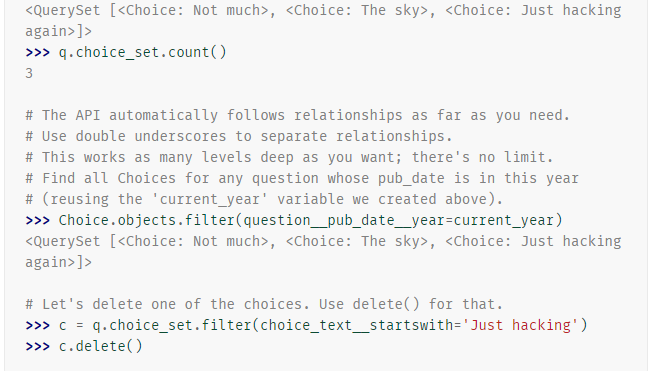
查看Choice內資料，可看到剛剛所輸入的資料。

查看與c對應的Question，得到回傳值。

建立三個Choice的資料。

顯示Choice的物件，目前裡面並無資料。

指定q為pk = 1的資料，並查看q的was\_published\_recently回傳值。



取得開頭為Just hacking的資料，指定給c，並刪除。

下面的指令可取得當年所發布的資料。

使用count()可得到資料筆數。



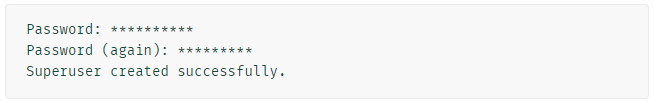
建立一個新的後台使用者。



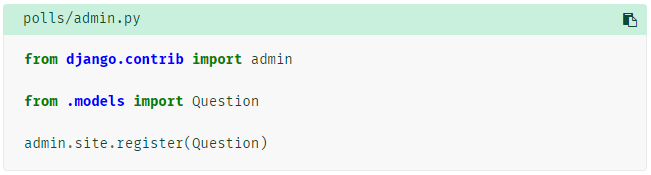
鍵入自己所取的使用者名稱。



輸入email。



輸入設定的密碼(兩次)。



在poll的admin.py中修改，讓Question在poll的首頁中呈現。