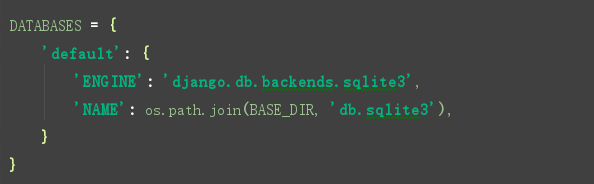
对于上一期，我们今天就开始来通过一个例子来了解Django模型

* 1. 制作图书管理

这里我们引用了《the Django book》中的例子，因为这个例子很好使

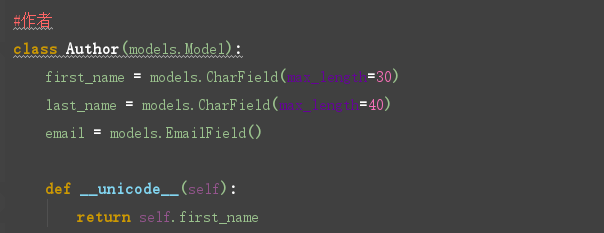
* 1. 数据库配置

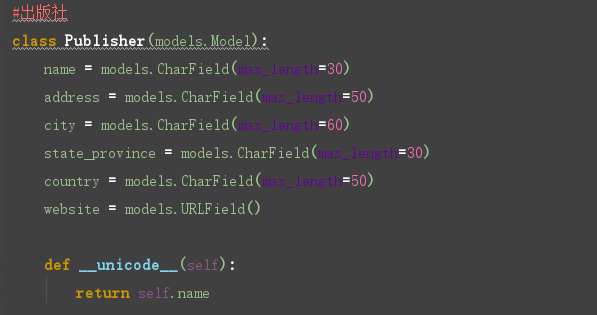
首先配置数据库，打开myweb/setting.py文件，查看是否为系统默认的设置：

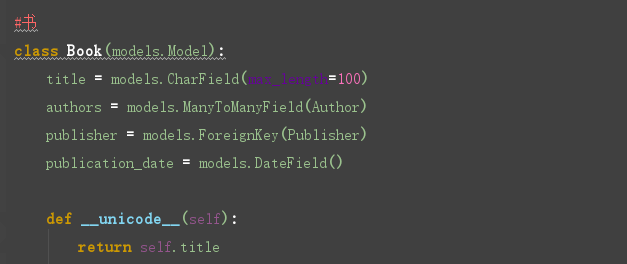


* 1. 创建表的映射

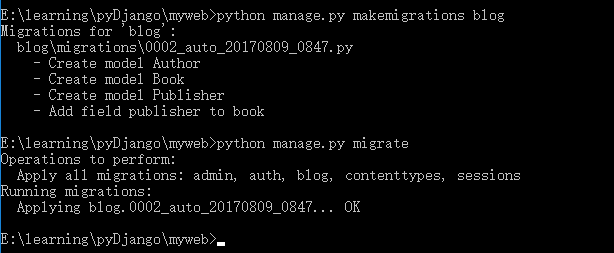
打开blog/models.py文件，创建作者、出版社和书







这些写完之后，我们就可以去执行数据库的生成和同步了，因为我们之前就创建了blog的数据库，所以现在直接同步到当前数据库就可以了



* 1. 创建表的后台管理

下面我们可以将创建的这几张表交给admin后台进行管理，我们来打开blog/admin.py文件：



看到这里可能很多都在思考，为什么之前我们创建blog的时候，不是在这里配置的呢？

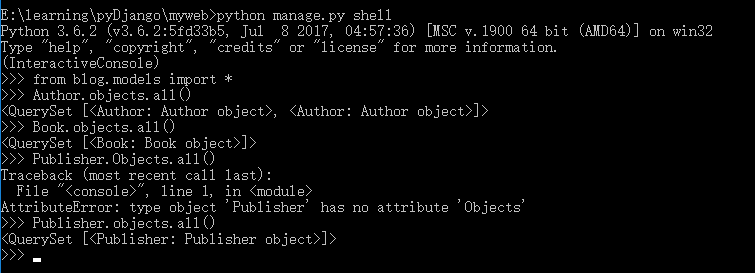
因为之前创建blog的时候，数据表只有两个，而admin的数据表只有一个，为了方便，之前的就没有用到admin.py文件。项目的发展，建议将前端和后台的代码区分开来。

登录admin后台: http://127.0.0.1:8000/admin/

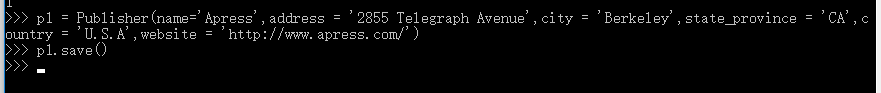


现在我们看到了新添加的三个数据表，我们为这三张数据表添加数据，细心的你就会发现一些问题了。

通过“python manage.py shell”命令进入shell



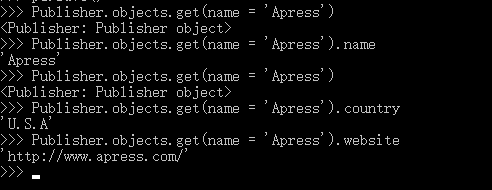
我们通过命令行方式为数据表添加数据



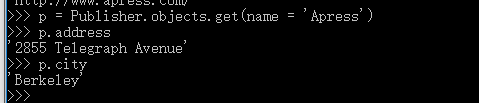
查询数据

接下来就学如何去查询数据，查询的方式也很多，下面列举一些常用的查询方式

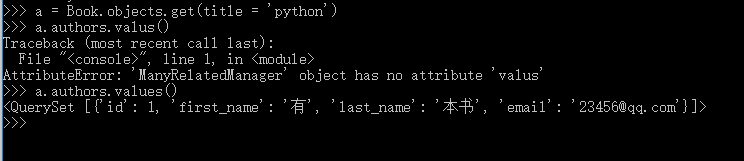
通过name=’Apress’条件查询：



通过变量查询:



查看title = python这本书的作者信息

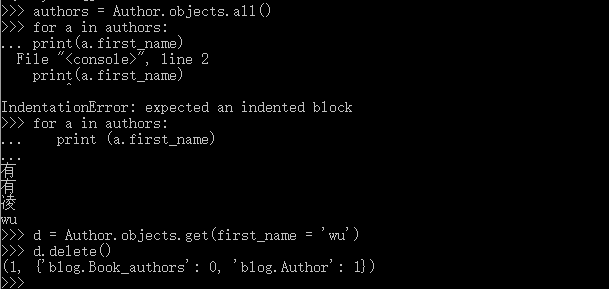


通过模糊查询：



* 1. 删除数据

删除 first name = wu的作者



我们首先通过for循环方式查询Authors表中first\_name的名字，再从中选择要删除的人，其中，通过shell方式来写代码，要注意Python代码的规范格式

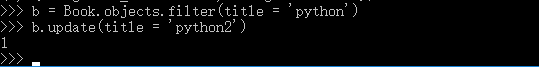
* 1. 更新数据

更新Book表中title = python的title为python2

更新前的数据：



执行更新命令：



更新后的数据：



关于数据库增删改查就先写到这里了