**数据库的设置**

**前提：mblog为项目名。Mainsite为其中的一个应用**

>修改mainsite/models.py：定义数据模型（即数据表及其字段）。Mainsite为该项目的一个app应用。本例增加了数据表Post

>使模型生效

(VENV) ~myDjango/mblog $ python manage.py makemigrations mainsite

(VENV) ~myDjango/mblog $ python manage.py migrate

>修改mainsite/admin.py将模型在Admin管理页面中注册：将Post纳入管理，

先导入（import）Post类（即Model），然后通过admin.site.register(Post)进行注册

>通过Admin管理界面增加Post数据表的记录

前提：创建Admin管理界面的管理员账号和密码——

Python manage.py createsuperuser

>修改admin.py，创建“表名Admin”类并注册：

创建PostAdmin类并注册，使Admin管理界面能够显示除第一个字段外的其他字段

**读取数据库的内容并展示到页面中**

>修改mainsite/view.py

导入模型： from .models import Post

定义内容输出函数：首页函数homepage和展示具体内容函数showpost

>修改mblog/urls.py

导入内容输出函数：from mainsite.views import homepage

在urlpatterns[ ]中，将网址和内容输出函数对应（批注A）

**模板的使用**

>创建templates文件夹。与mblog、mainsite同级

>修改mblog/settings.py

设置模板文件夹的位置：修改TMPLATES->DIRS

>在url.py中创建网址和views.py中函数的对应关系。（参考批注A）

>在templates文件夹中创建.html文件

>运行程序，以objects.all()在views.py中取得数据

>以render函数（渲染函数）把数据送到指定的模板文件（如index.html）中

>将得到的结果交给HttpResponse输出给浏览器