

## 1、如何分目录

可以按功能模块分，取决于开发者开发的思路

## 2、主要架构MTV

MTV (models、templates、views)

urls 路由，每个url对应到哪个函数

views 业务逻辑层，就是具体的处理函数，主要做逻辑处理

models 数据处理层，负责与数据的交互，例如定义表结构

如果实现一个特定的功能可以单独建立一个文件，例如发短信

## 3、数据库配置

如果新增应用，settings中INSTALLED\_APPS文件有所修改，需要执行

```
1 $ python manage.py migrate
```

如果仅仅是models文件有新增或者修改时，同步到数据库

```
python manage.py makemigrations app名
```

```
1 python manage.py makemigrations caradmin
```

```
python manage.py sqlmigrate app名 实际文件编号
```

```
1 python manage.py sqlmigrate caradmin 0002
```

## 4、创建应用

```
1 python manage.py startapp appname
```

## 5、models

django会根据Model类生成一个数据库镜像文件，然后再使用该镜像文件生成数据库，同时该文件将记录与数据库同步版本的变化，所以在使用django进行开发时不要手工去修改数据库，这样会造成django框架的版本记录不正确，从而无法正确的同步数据模型与数据库的内容

## 6、objects属性

objects是模型属性、Manager类型的对象，用于模型对象和数据库交互

### 1) objects获取数据

```
1 list = Test.objects.all() #获得所有数据行，相当于SQL中的SELECT * FROM
2 response2 = Test.objects.filter(id=1) #相当于SQL中的WHERE，可设置条件过滤结果
3 response3 = Test.objects.get(id=1) # 获取单个对象
4 response4 = Test.objects.order_by("id") #数据排序
```

objects.get()、objects.filter()的区别

objects.get()、objects.filter()两个方法都是用来查询数据库中的数据，不同之处get()查询到数据时会返回一个对象，查询不到数据时会报错，查询到多条数据也会报错；filter()查询数据时会返回一个对象列表，不到数据时会返回[]

### 2) objects新增数据

两种方式，1.objects.create()向数据库插入数据 2.先创建对象，再执行save函数

```
1 def testdb1(request):
2     test1 = Test(name='runoob')
3     test1.save()
4 def testdb2(request):
5     b = BlogMsg.objects.create(title=title, content=content)
```

### 3) objects删除数据

```
1 d = BlogMsg.objects.filter(id=bid).delete() #这里用filter而不用get是防止查询不到数据时报错
```

### 4) objects更新数据

两种方式，1.objects.update()修改数据 2.先获取要修改的数据，修改对应数据后再实行save函数

```
1 d=BlogMsg.objects.filter(id=bid).update(title=title, content=content)
```

```
1 def testdb(request):  
2     # 修改其中一个id=1的name字段, 再save, 相当于SQL中的UPDATE  
3     test1 = Test.objects.get(id=1)  
4     test1.name = 'Google'  
5     test1.save()
```