

# Django 学习笔记



*for my dear wife.*

Rails Chen

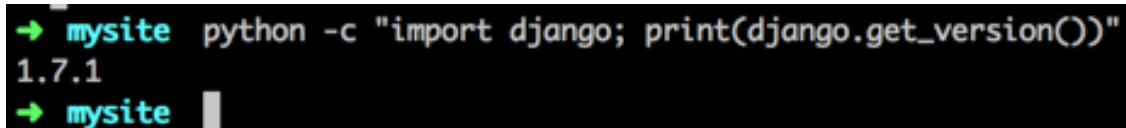
2014年12月

# Django app,part 1

## 1.1 Django 之 helloworld

### 1.查看Django版本

`python -c "import django; print(django.get_version())"`

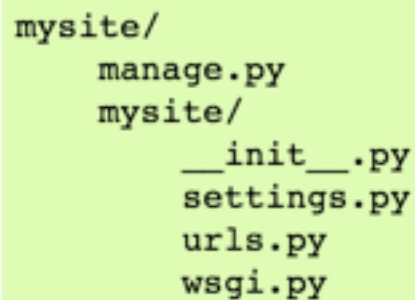


```
→ mysite python -c "import django; print(django.get_version())"
1.7.1
→ mysite
```

### 2.新建project

`django-admin.py startproject mysite`

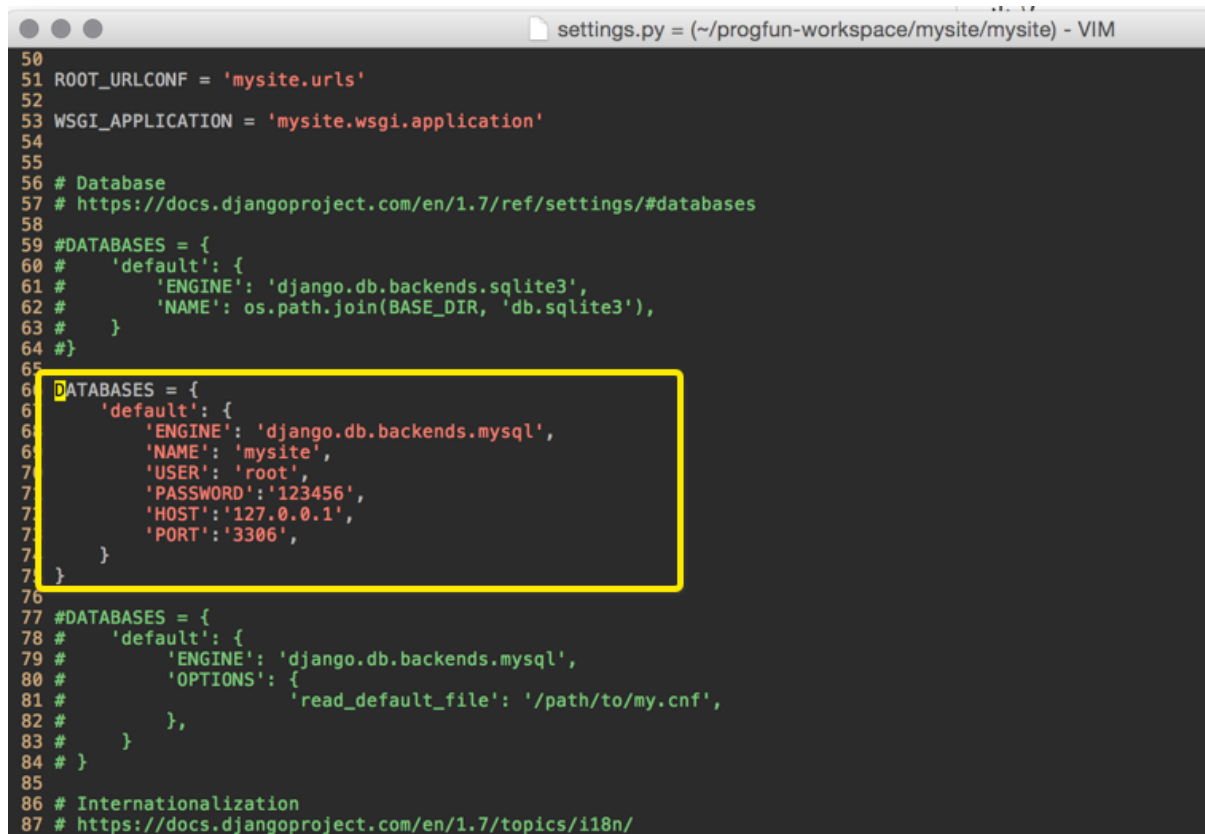
自动生成如下文件：



```
mysite/
  manage.py
  mysite/
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

### 3.配置MySQL数据库（数据库模块配置参考上一篇文章）

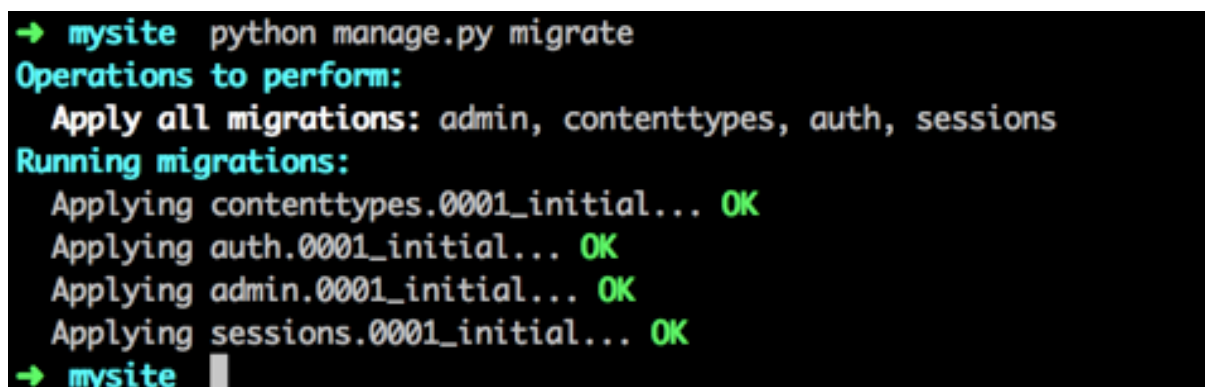
修改setting.py文件，以下是MySQL配置



```
50
51 ROOT_URLCONF = 'mysite.urls'
52
53 WSGI_APPLICATION = 'mysite.wsgi.application'
54
55
56 # Database
57 # https://docs.djangoproject.com/en/1.7/ref/settings/#databases
58
59 #DATABASES = {
60 #     'default': {
61 #         'ENGINE': 'django.db.backends.sqlite3',
62 #         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
63 #     }
64 #}
65
66 DATABASES = {
67     'default': {
68         'ENGINE': 'django.db.backends.mysql',
69         'NAME': 'mysite',
70         'USER': 'root',
71         'PASSWORD': '123456',
72         'HOST': '127.0.0.1',
73         'PORT': '3306',
74     }
75 }
76
77 #DATABASES = {
78 #     'default': {
79 #         'ENGINE': 'django.db.backends.mysql',
80 #         'OPTIONS': {
81 #             'read_default_file': '/path/to/my.cnf',
82 #         },
83 #     }
84 # }
85
86 # Internationalization
87 # https://docs.djangoproject.com/en/1.7/topics/i18n/
```

4.初始化数据（可选择运行）

python manage.py migrate



```
→ mysite python manage.py migrate
Operations to perform:
  Apply all migrations: admin, contenttypes, auth, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying sessions.0001_initial... OK
→ mysite
```

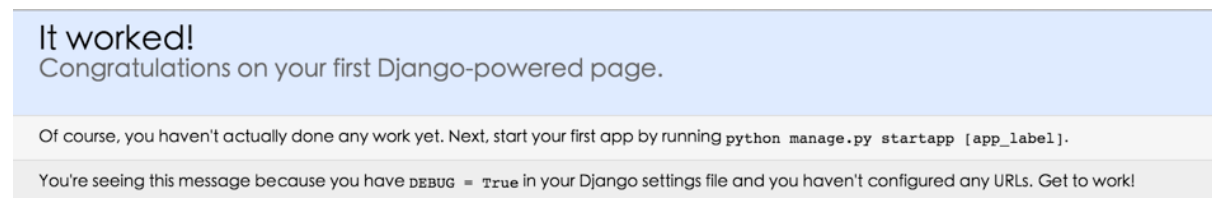
5.部署工程，启动服务

python manage.py runserver

```
→ mysite python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
December 14, 2014 - 15:46:33
Django version 1.7.1, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

访问：<http://127.0.0.1:8000/>，成功！



其他启动方式，端口与IP可选：

python manage.py runserver 8080

或

python manage.py runserver 0.0.0.0:8000

## 1.2 Django之创建 Models

1.进入与manage.py同一路径下:

执行: `python manage.py startapp polls`

创建以下文件:

```
polls/  
  __init__.py  
  admin.py  
  migrations/  
    __init__.py  
  models.py  
  tests.py  
  views.py
```

2.编辑polls/models.py, 建立数据模型 Question and Choice。

```
from django.db import models
```

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

```
    question_text = models.CharField(max_length=200)
```

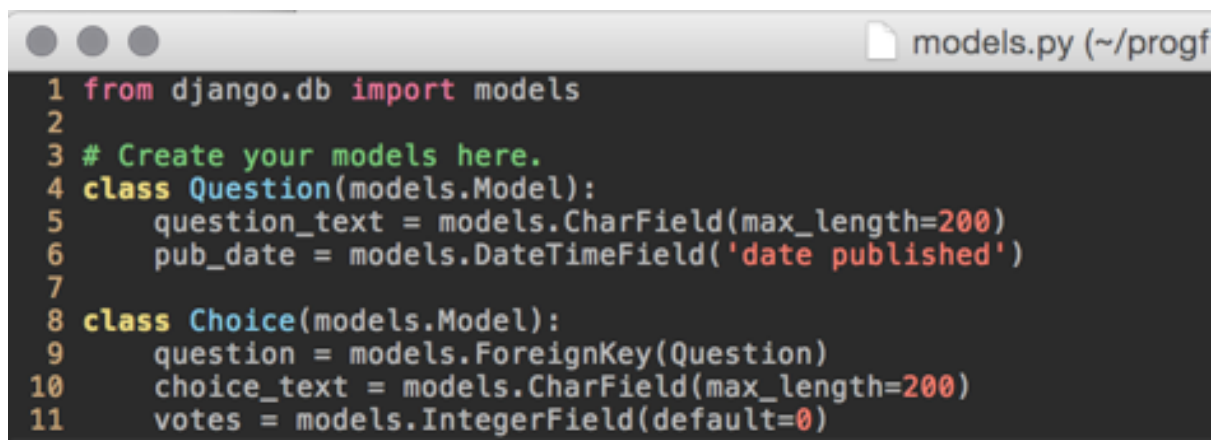
```
    pub_date = models.DateTimeField('date published')
```

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

```
    question = models.ForeignKey(Question)
```

```
    choice_text = models.CharField(max_length=200)
```

```
    votes = models.IntegerField(default=0)
```



```
models.py (~/progrf  
1 from django.db import models  
2  
3 # Create your models here.  
4 class Question(models.Model):  
5     question_text = models.CharField(max_length=200)  
6     pub_date = models.DateTimeField('date published')  
7  
8 class Choice(models.Model):  
9     question = models.ForeignKey(Question)  
10    choice_text = models.CharField(max_length=200)  
11    votes = models.IntegerField(default=0)
```

3. 接下来，告诉这个工程我们创建了一个模型。修改mysite/settings.py 文件, 在INSTALLED\_APPS 配置中加入 'polls'.

```
31
32 INSTALLED_APPS = (
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'polls',
40 )
41
```

执行命令：python manage.py makemigrations polls

```
→ mysite python manage.py makemigrations polls
Migrations for 'polls':
  0001_initial.py:
    - Create model Choice
    - Create model Question
    - Add field question to choice
→ mysite
```

4.提交到MySQL数据库中：

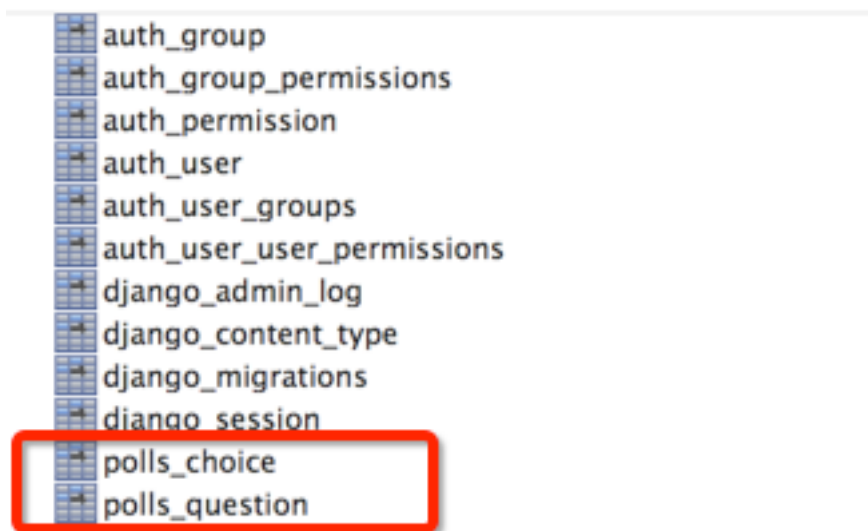
python manage.py sqlmigrate polls 0001

```
→ mysite python manage.py sqlmigrate polls 0001
BEGIN;
CREATE TABLE `polls_choice` (`id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY, `choice_text` varchar(200) NOT NULL, `votes` integer NOT NULL);
CREATE TABLE `polls_question` (`id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY, `question_text` varchar(200) NOT NULL, `pub_date` datetime NOT NULL);
ALTER TABLE `polls_choice` ADD COLUMN `question_id` integer NOT NULL;
ALTER TABLE `polls_choice` ALTER COLUMN `question_id` DROP DEFAULT;
CREATE INDEX polls_choice_7aa0f6ee ON `polls_choice` (`question_id`);
ALTER TABLE `polls_choice` ADD CONSTRAINT `polls_choice_question_id_246c99a640fb
bd72_fk_polls_question_id` FOREIGN KEY (`question_id`) REFERENCES `polls_questio
n` (`id`);
COMMIT;
→ mysite
```

提交: `python manage.py migrate`

```
→ mysite python manage.py migrate
Operations to perform:
  Apply all migrations: admin, contenttypes, polls, auth, sessions
Running migrations:
  Applying polls.0001_initial... OK
→ mysite
```

在MySQL中生成两张表:



## 1.3 database API的使用

### 1.进入shell

python manage.py shell

### 2.使用database API命令

➔ mysite python manage.py shell

Python 2.7.6 (default, Sep 9 2014, 15:04:36)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

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

```
>>> Question.objects.all()
```

```
[]
```

```
>>> from django.utils import timezone
```

```
>>> q = Question(question_text="What's new?", pub_date=timezone.now())
```

```
>>> q.save()
```

```
>>> q.id
```

```
1L
```

```
>>> q.question_text
```

```
"What's new?"
```

```
>>> q.pub_date
```

```
datetime.datetime(2014, 12, 15, 7, 10, 53, 169056, tzinfo=<UTC>)
```

```
>>> q.question_text = "What's up?"
```

```
>>> q.save()
```

```
>>> Question.objects.all()
```

```
[<Question: Question object>]
```

```
>>>
```



```

→ mysite python manage.py shell
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from polls.models import Question, Choice
>>> Question.objects.all()
[]
>>> from django.utils import timezone
>>> q = Question(question_text="What's new?", pub_date=timezone.now())
>>> q.save()
>>> q.id
1L
>>> q.question_text
"What's new?"
>>> q.pub_date
datetime.datetime(2014, 12, 15, 7, 10, 53, 169056, tzinfo=<UTC>)
>>> q.question_text = "What's up?"
>>> q.save()
>>> Question.objects.all()
[<Question: Question object>]
>>>

```

3. 增加模型的 `__str__()` 的方法，修改 `polls/models.py`

注意：Python 2 增加的函数为 `__unicode__()` 方法，参考资料：[https://docs.djangoproject.com/en/1.7/ref/models/instances/#django.db.models.Model.\\_\\_unicode\\_\\_](https://docs.djangoproject.com/en/1.7/ref/models/instances/#django.db.models.Model.__unicode__)

```

# Create your models here.
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def __str__(self):#Python 3
        return self.question_text

    def __unicode__(self):#Python 2
        return u'%s' % (self.question_text)

```

```

19
20 class Choice(models.Model):
21     question = models.ForeignKey(Question)
22     choice_text = models.CharField(max_length=200)
23     votes = models.IntegerField(default=0)
24
25     def __str__(self):#Python 3
26         return self.choice_text
27
28     def __unicode__(self):#Python 2
29         return u'%s' % (self.choice_text)
30

```

#### 4. 以下是database API的常用方法

➔ mysite python manage.py shell

Python 2.7.6 (default, Sep 9 2014, 15:04:36)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

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

```
>>> Question.objects.all()
```

```
[<Question: What's up?>]
```

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

```
[<Question: What's up?>]
```

```
>>> Question.objects.filter(question_text__startswith='What')
```

```
[<Question: What's up?>]
```

```
>>>
```

```
>>> from django.utils import timezone
```

```
>>> current_year = timezone.now().year
```

```
>>> Question.objects.get(pub_date__year=current_year)
```

```
<Question: What's up?>
```

```
>>>
```

```
>>>
```

```
>>> Question.objects.get(id=2)
```

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "/Library/Python/2.7/site-packages/Django-1.7.1-py2.7.egg/django/db/models/manager.py", line 92, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "/Library/Python/2.7/site-packages/Django-1.7.1-py2.7.egg/django/db/models/query.py", line 357, in get

self.model.\_meta.object\_name)

DoesNotExist: Question matching query does not exist.

```
>>>
```

```
>>>
```

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

```
<Question: What's up?>
```

```

>>>
>>> q=Question.objects.get(pk=1)
>>> q.was_published_recently()
True
>>> q=Question.objects.get(pk=1)
>>> q.choice_set.all()
[]
>>>
>>> q.choice_set.create(choice_text='Not much', votes=0)
<Choice: Not much>
>>> q.choice_set.create(choice_text='The sky', votes=0)
<Choice: The sky>
>>> c = q.choice_set.create(choice_text='Just hacking again', votes=0)
>>>
>>> c.question
<Question: What's up?>
>>>
>>> q.choice_set.all()
[<Choice: Not much>, <Choice: The sky>, <Choice: Just hacking again>]
>>> q.choice_set.count()
3
>>> Choice.objects.filter(question__pub_date__year=current_year)
[<Choice: Not much>, <Choice: The sky>, <Choice: Just hacking again>]
>>>
>>> c = q.choice_set.filter(choice_text__startswith='Just hacking')
>>> c.delete()
>>>

```