Tedu.cn
达内教育

Python开发

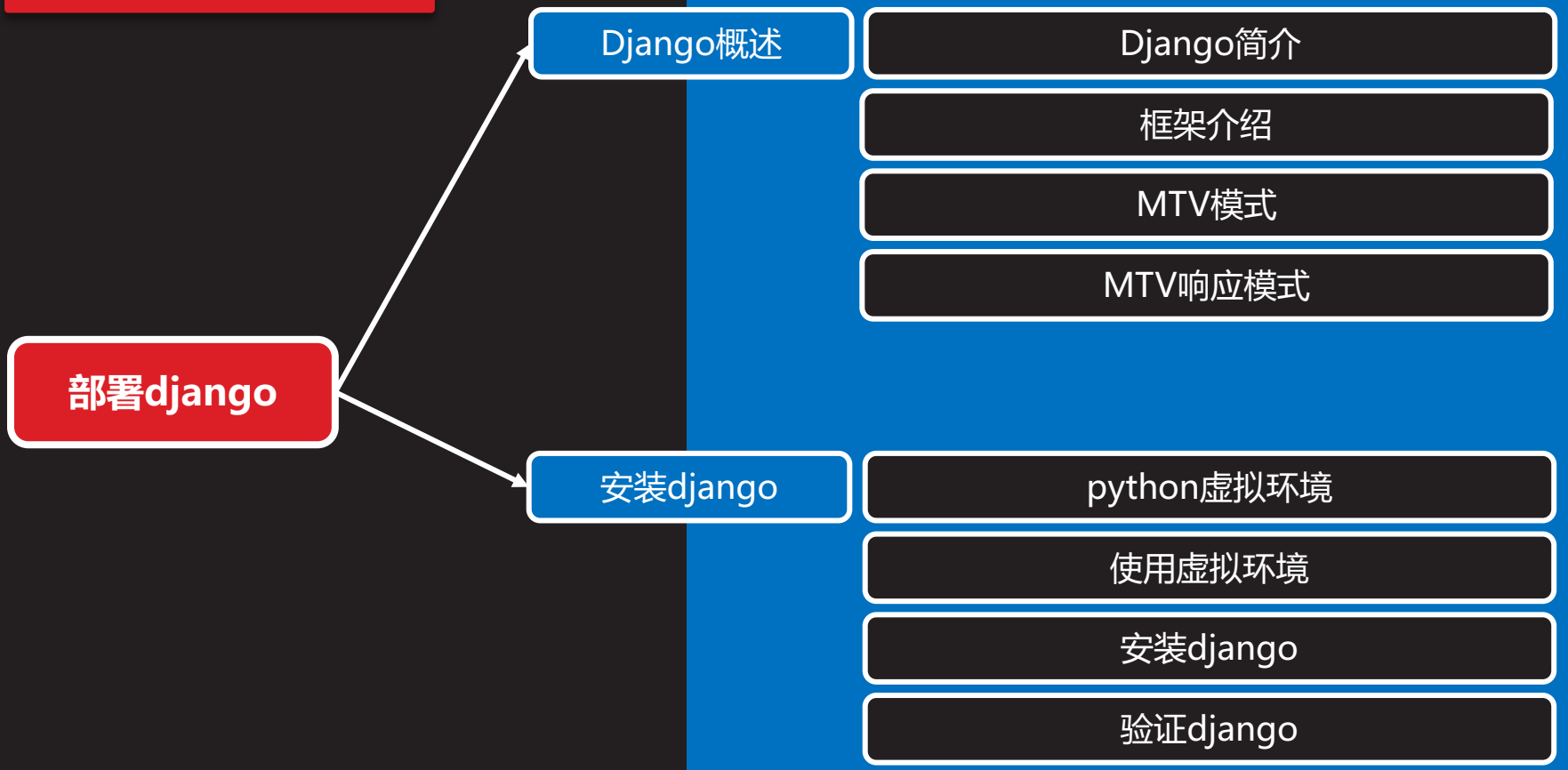
NSD DEVWEB

DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	部署django
	10:30 ~ 11:20	
	11:30 ~ 12:00	Django应用基础
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Django模型
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑

部署django


部署django
Django概述
Django简介
框架介绍
MTV模式
MTV响应模式
安装django
python虚拟环境
使用虚拟环境
安装django
验证django

Tedu.cn
达内教育

Django概述

Django简介

- Django是一个开放源代码的Web应用框架，由Python写成
- 最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站
- 2005年7月在BSD许可证下发布

知识讲解

框架介绍

- Django 框架的核心组件有：
  - 用于创建模型的对象关系映射
  - 为最终用户设计的完美管理界面
  - 一流的 URL 设计
  - 设计者友好的模板语言
  - 缓存系统

知识讲解

MTV模式

- Django的MTV模式本质上和MVC是一样的，也是为了各组件间保持松耦合关系，只是定义上有些许不同
  - M 代表模型（Model）：负责业务对象和数据库的关系映射(ORM)
  - T 代表模板 (Template)：负责如何把页面展示给用户(html)
  - V 代表视图（View）：负责业务逻辑，并在适当时候调用Model和Template
- 除了以上三层之外，还需要一个URL分发器，它的作用是将一个个URL的页面请求分发给不同的View处理，View再调用相应的Model和Template

知识讲解

MTV响应模式

- Web服务器（中间件）收到一个http请求
- Django在URLconf里查找对应的视图(View)函数来处理http请求
- 视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面
- 视图函数处理结束后返回一个http的响应给Web服务器
- Web服务器将响应发送给客户端

知识讲解

Tedu.cn
达内教育

安装django

python虚拟环境

- 每个项目都需要安装很多库，当项目结束后这些库不需要了，该如何清理？
- 在同一个主系统内，需要同时安装django1.11.6和django2.0.5如何实现
- 有了python虚拟环境，这些问题将迎刃而解

知识讲解

使用虚拟环境

- Python3中已经自带虚拟环境，只要创建并激活即可

```
[root@localhost ~]# mkdir pyproject
[root@localhost ~]# cd pyproject/
[root@localhost pyproject]# python3 -m venv django_env
[root@localhost pyproject]# source django_env/bin/activate
(django_env) [root@localhost pyproject]#
```

知识讲解

安装django

- Django不同版本有微小不同，本课程选用2.2版本

```
(django_env) [root@localhost pyproject]# pip install django==1.11.6
```

- 如果安装过于缓慢，可以使用国内镜像站点

```
(django_env) [root@localhost pyproject]# cat ~/.pip/pip.conf
[global]
index-url=http://pypi.douban.com/simple/
[install]
trusted-host=pypi.douban.com
```

知识讲解

验证django

- Django安装完毕后，就可以导入相关模块了

```
(django_env) [root@localhost pyproject]# python
Python 3.6.4 (default, Apr 27 2018, 08:26:23)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> django.__version__
' 1.11.6'
```

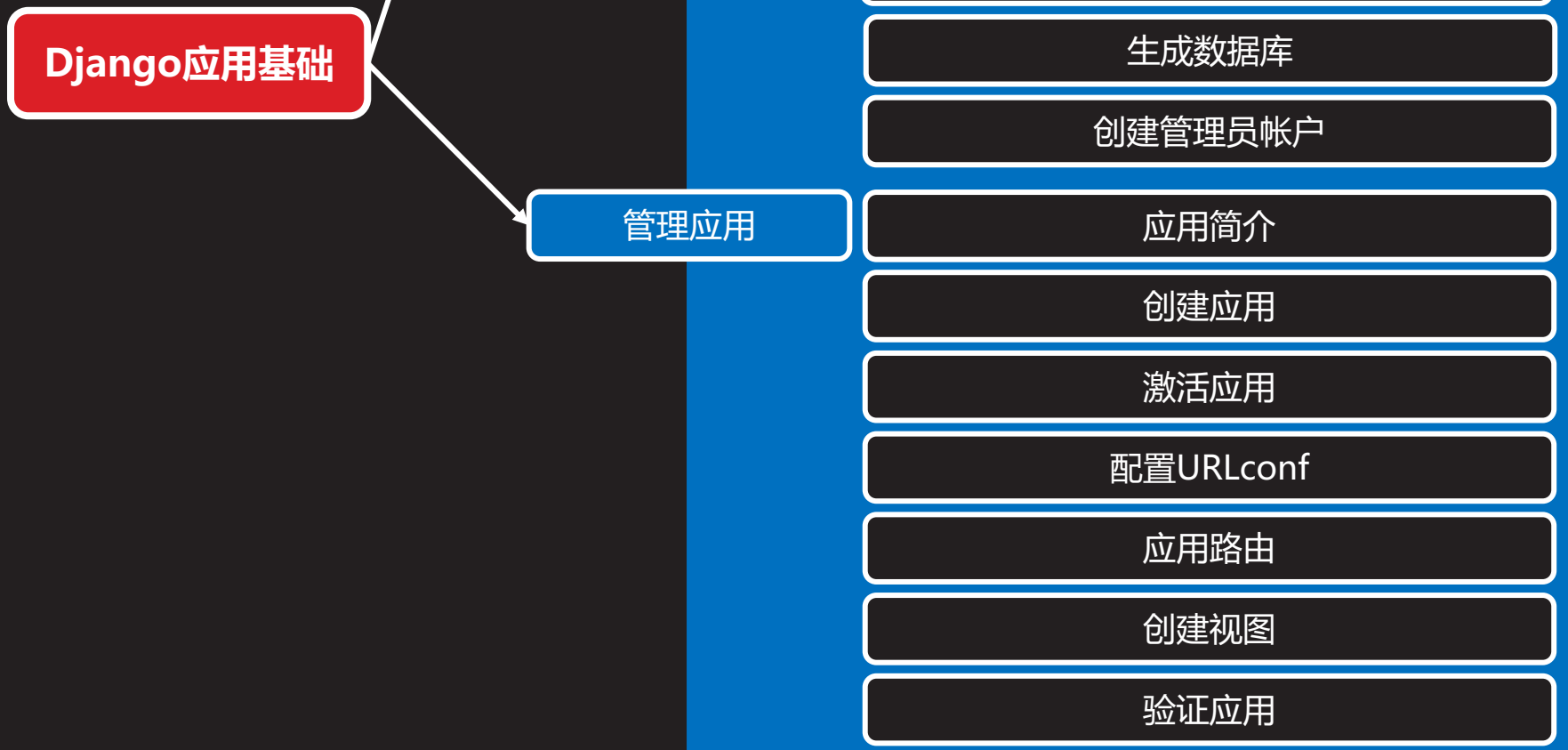
知识讲解

案例1：安装django

1. 创建python虚拟环境
2. 激活python虚拟环境
3. 在虚拟环境中安装django
4. 验证django是否安装正确

课堂练习

Django应用基础


Django应用基础
管理项目
创建项目
项目文件说明
开发服务器
访问django
修改设置
访问后台
生成数据库
创建管理员帐户
管理应用
应用简介
创建应用
激活应用
配置URLconf
应用路由
创建视图
验证应用

Tedu.cn
达内教育

管理项目

创建项目

知识讲解

- Djanog可以自动生成一些代码，这些代码创建一个Django项目：一个Django实例的设置集合，包括数据库的配置、Django有关的选项

```
(django_env) [root@localhost pyproject]# django-admin startproject
mysite
(django_env) [root@localhost pyproject]# tree mysite
mysite
├── manage.py
└── mysite
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py

1 directory, 5 files
```

项目文件说明

知识讲解

- 外层的mysite/根目录仅仅是项目的一个容器。 它的名字与Django无关；可以将其重命名为你喜欢的任何内容
- manage.py：一个命令行工具，可以使用多种方式对Django项目进行交互。 可以在django-admin和manage.py中读到关于manage.py的所有细节
- 内层的mysite/目录是项目的真正的Python包。 它是导入任何东西时将需要使用的Python包的名字（例如 mysite.urls）

项目文件说明（续1）

- mysite/__init__.py：一个空文件，它告诉Python这个目录应该被看做一个Python包
- mysite/settings.py：该Django 项目的设置/配置。Django settings 将告诉你这些设置如何工作。
- mysite/urls.py：此Django项目的URL声明；Django驱动的网站的“目录”
- mysite/wsgi.py：用于项目的与WSGI兼容的Web服务器入口

知识讲解

开发服务器

- Django自带一个开发服务器，默认运行于8000端口
- 该开发服务器不要应用在生产环境下

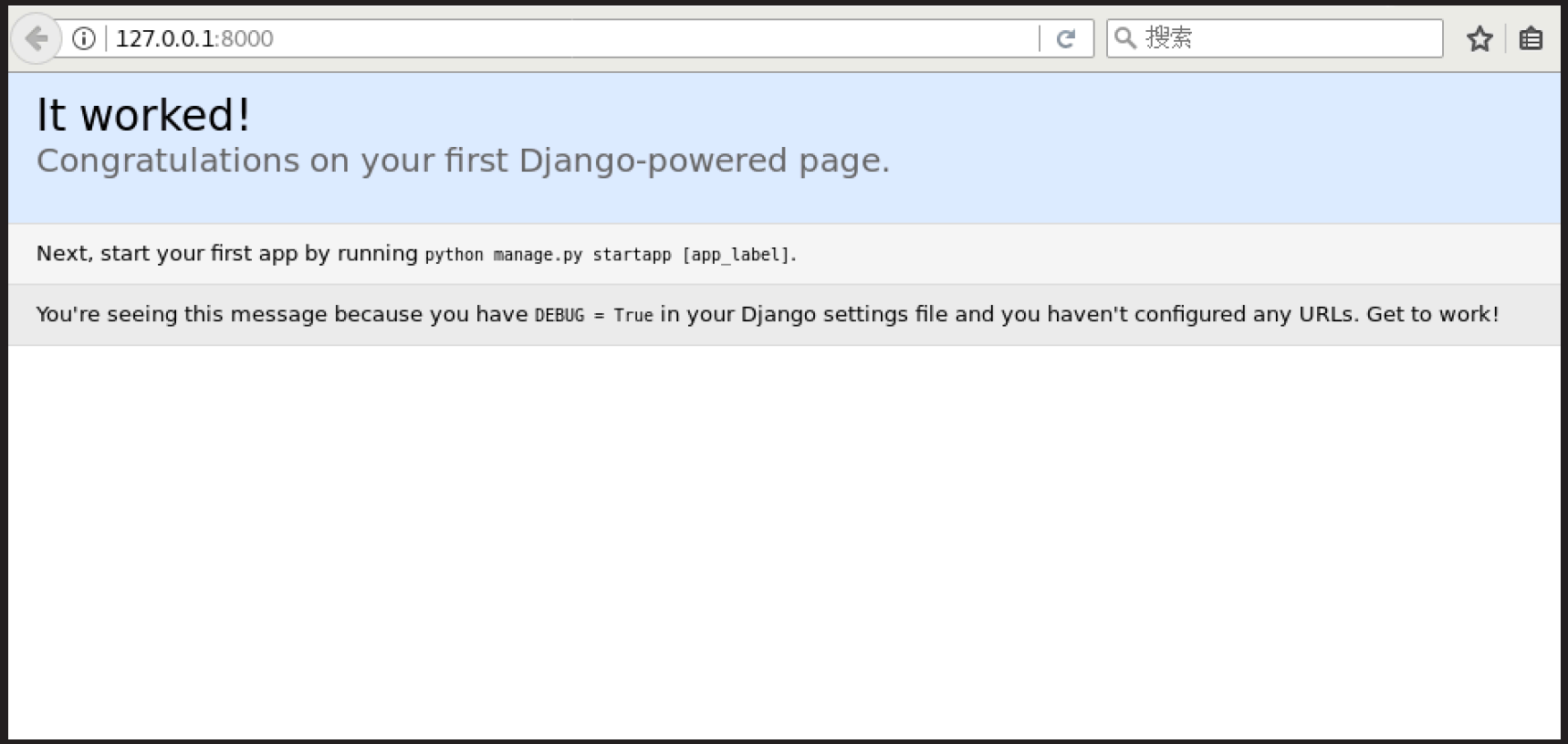
```
(django_env) [root@localhost mysite]# python manage.py runserver
Performing system checks...
System check identified no issues (0 silenced).

You have 13 unapplied migration(s). Your project may not work properly
until you apply the migrations for app(s): admin, auth, contenttypes,
sessions.
Run 'python manage.py migrate' to apply them.

May 09, 2018 - 23:13:04
Django version 1.11.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

知识讲解

访问django



知识讲解

修改设置

- Django默认只允许本机访问、英文环境，这些均可在settings.py中进行修改

```
ALLOWED_HOSTS = '*’

LANGUAGE_CODE = 'zh-Hans’
TIME_ZONE = 'Asia/Shanghai’
```

知识讲解

修改设置（续1）

知识讲解

访问后台

- 一旦创建了项目，django自动生成了后台管理页面
  http://127.0.0.1:8000/admin

知识讲解

生成数据库

- Django生成的项目，使用了很多预先编写好的应用，这些应用需要用到数据
- 只要执行以下语句，即可自动生成数据库

```
(django_env) [root@localhost mysite]# python manage.py migrate
```

知识讲解

创建管理员帐户

- 访问后台需要有超级用户身份
- 超级用户需要单独创建
- 用户将写到数据库中

```
(django_env) [root@localhost mysite]# python manage.py
createsuperuser
Username (leave blank to use 'root'): admin
Email address: zzg@tedu.cn
Password:
Password (again):
Superuser created successfully.
```

知识讲解

创建管理员帐户（续1）

- 创建好管理员用户后即可登陆后台

知识讲解

案例2：创建项目

1. 创建名为mysite的项目
2. 生成数据库
3. 创建超级用户
4. 登陆后台管理页面

课堂练习

Tedu.cn
达内教育

管理应用

应用简介

- 应用是一个Web应用程序，它完成具体的事项 —— 比如一个博客系统、一个存储公共档案的数据库或者一个简单的投票应用
- 项目是特定网站的配置和应用程序的集合
- 一个项目可以包含多个应用
- 一个应用可以运用到多个项目中去

知识讲解

创建应用

- 应用可以放在Python path上的任何位置
- 可以在manage.py文件同级目录创建应用，以便可以将它作为顶层模块导入，而不是mysite的子模块

```
(django_env) [root@localhost mysite]# python manage.py startapp polls
```

知识讲解

激活应用

- 创建应用后，需要将其安装到项目中，否则应用不会生效

修改mysite/settings.py:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'polls',
]
```

知识讲解

配置URLconf

- 为了整洁起见，用户与polls相关的url都交给polls应用进行路由
- 修改mysite/urls.py如下：

```
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^polls/', include('polls.urls'))
]
```

知识讲解

应用路由

- 创建polls应用的URLconf，配置访问polls应用首页的视图
- 创建polls/urls.py，内容如下：

```
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

知识讲解

应用路由（续1）

- URL正则采用的是match操作
  - r'^hello':匹配以hello开头的任意URL,如:/helloabc
  - r'^hello/$':匹配hello且后面无信息的URL,如:/hello,/hello/
  - r'^$':匹配 / 即空URL,通常用来设定应用的根,即默认入口。如: http://IP:port或者http://IP:port/

知识讲解

创建视图

- 视图是URLconf路由到某一URL时执行的函数
- 为上一步polls主页编写简单视图，编辑polls/views.py如下：

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hi! 这是polls应用的首页。")
```

知识讲解

验证应用

- 至此，用户已经可以访问polls应用的主页

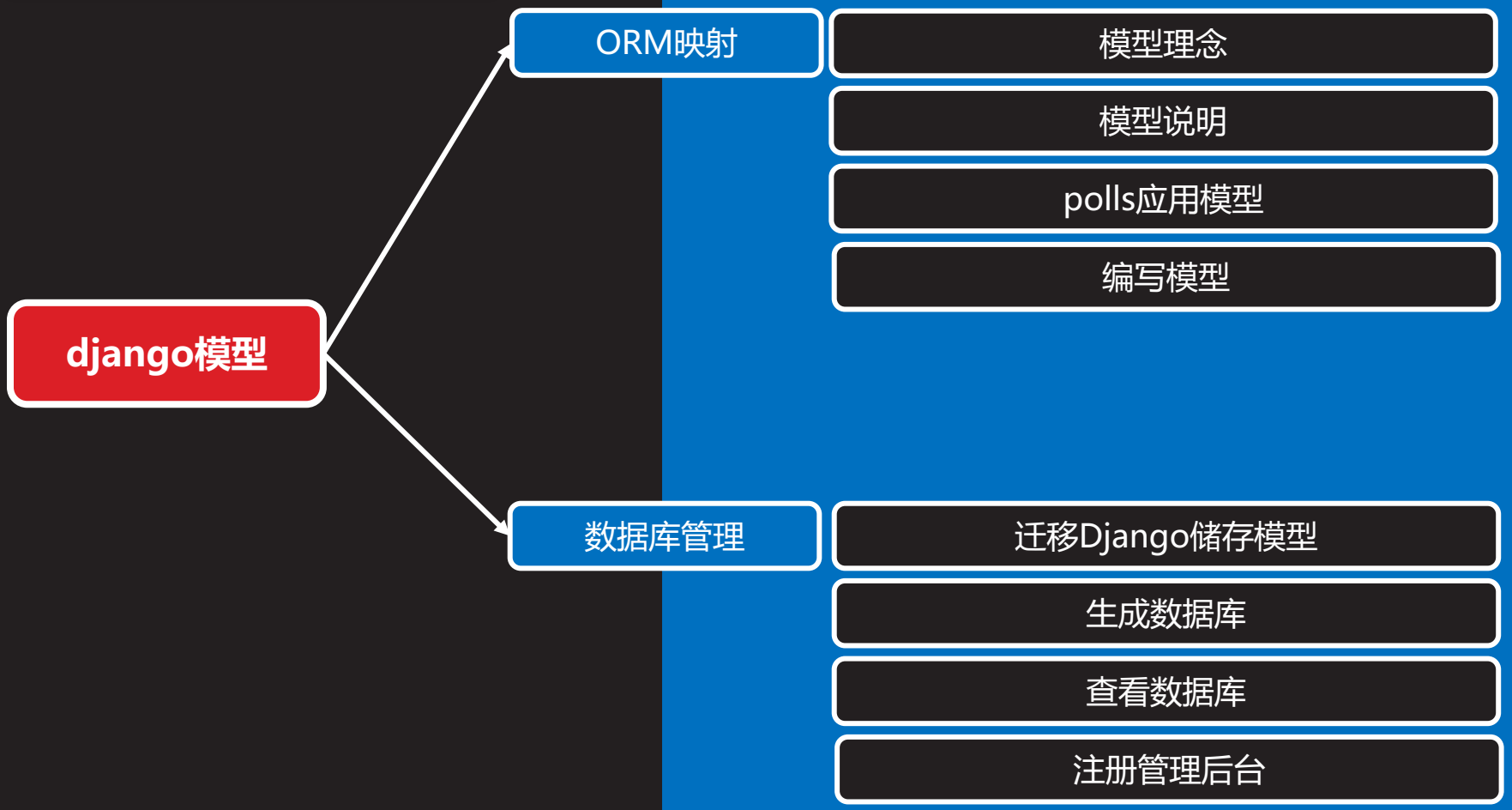
知识讲解

案例3：创建应用

1. 创建应用polls
2. 注册polls到项目中
3. 配置URLconf，当访问http://127.0.0.1:8000/polls时，由polls进行路由
4. 为polls应用配置主页视图

课堂练习

django模型


django模型
ORM映射
模型理念
模型说明
polls应用模型
编写模型
数据库管理
迁移Django储存模型
生成数据库
查看数据库
注册管理后台

Tedu.cn
达内教育

ORM映射

模型理念

- 模型指出了数据的唯一、明确的真实来源
- 它包含了正在存储的数据的基本字段和行为
- Django遵循DRY(Do not Repeat Yourself) 原则
- 这个原则的目标是在一个地方定义你的数据模型，并自动从它获得需要的信息

知识讲解

模型说明

- 每个模型都用一个类表示，该类继承自django.db.models.Model
- 每个模型都有一些类变量，在模型中每个类变量都代表了数据库中的一个字段
- 每个字段通过Field类的一个实例表示 —— 例如字符字段CharField和日期字段DateTimeField。 这种方法告诉Django，每个字段中保存着什么类型的数据

知识讲解

模型说明（续1）

- 每个Field 实例的名字就是字段的名字，并且是机器可读的格式
- 在Python代码中将使用到它的值，并且数据库将把它用作表的列名

知识讲解

polls应用模型

- 在这个简单的投票应用中，我们将创建两个模型：Question和Choice
- Question对象具有一个question_text（问题）属性和一个publish_date（发布时间）属性
- Choice有两个字段：选择的内容和选择的得票统计
- 每个Choice与一个Question关联

知识讲解

编写模型

- 编辑polls/models.py文件，输入以下内容：

```
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, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

知识讲解

案例4：创建模型

- 为polls应用创建模型
- 一个模型名为Question，用于记录问题
- 另一个模型名为Choice，每个Choice只能对应一个Question，但是一个Question可以对应多个Choice

课堂练习

Tedu.cn
达内教育

数据库管理

迁移Django储存模型

知识讲解

- polls应用建立了新的模型，该模型需要反馈到数据库中
- 通过运行makemigrations告诉Django，已经对模型做了一些更改，并且会将这些更改记录为迁移文件
- 迁移文件位于polls/migrations/目录下

```
(django_env) [root@localhost mysite]# python manage.py
makemigrations
Migrations for 'polls':
  polls/migrations/0001_initial.py
    - Create model Choice
    - Create model Question
    - Add field question to choice
```

生成数据库

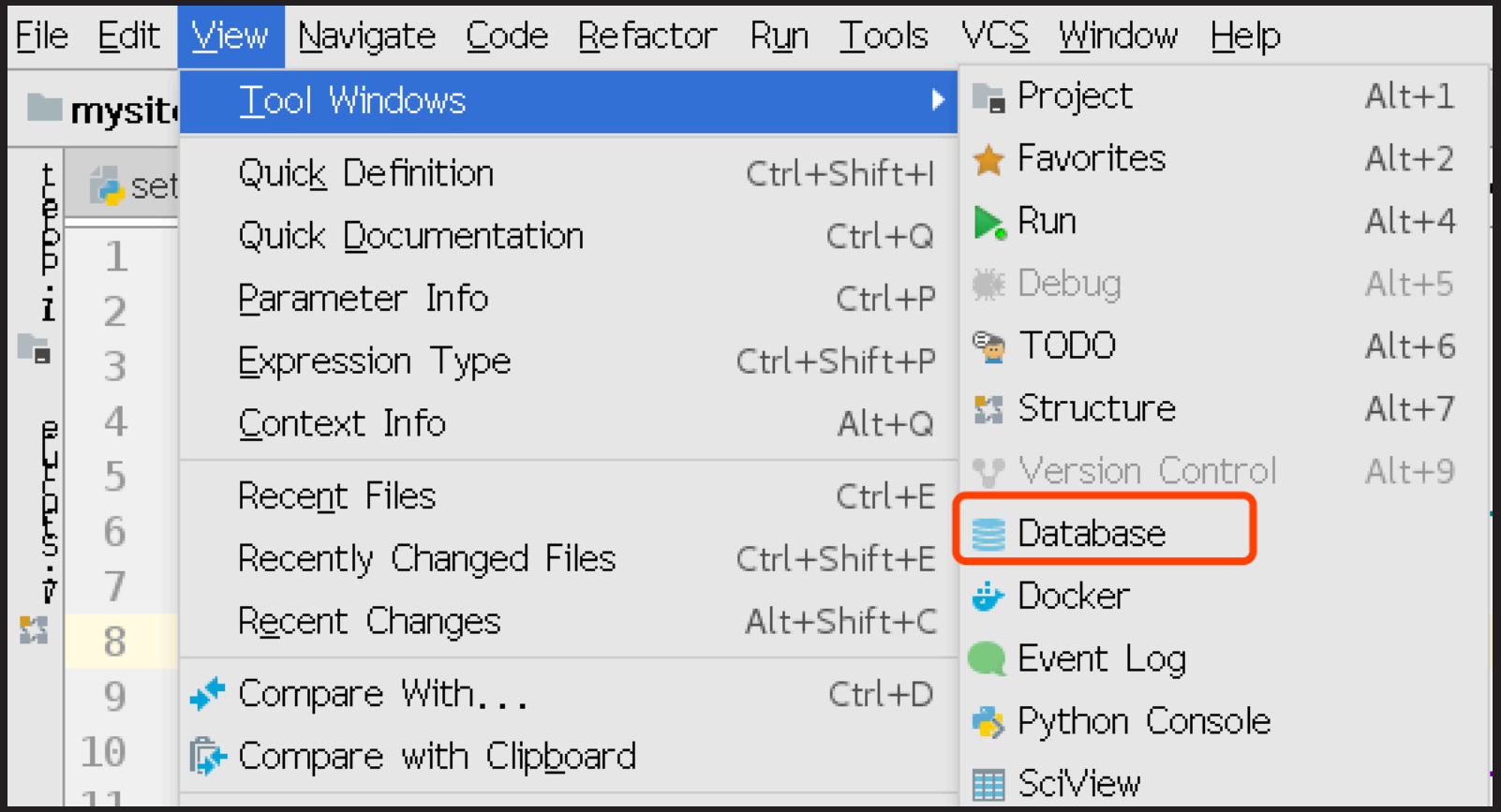
- makemigrations只是生成了迁移文件，并未真正的反馈到数据中
- migrate才是真正生成数据库

```
(django_env) [root@localhost mysite]# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Applying polls.0001_initial... OK
```

知识讲解

查看数据库

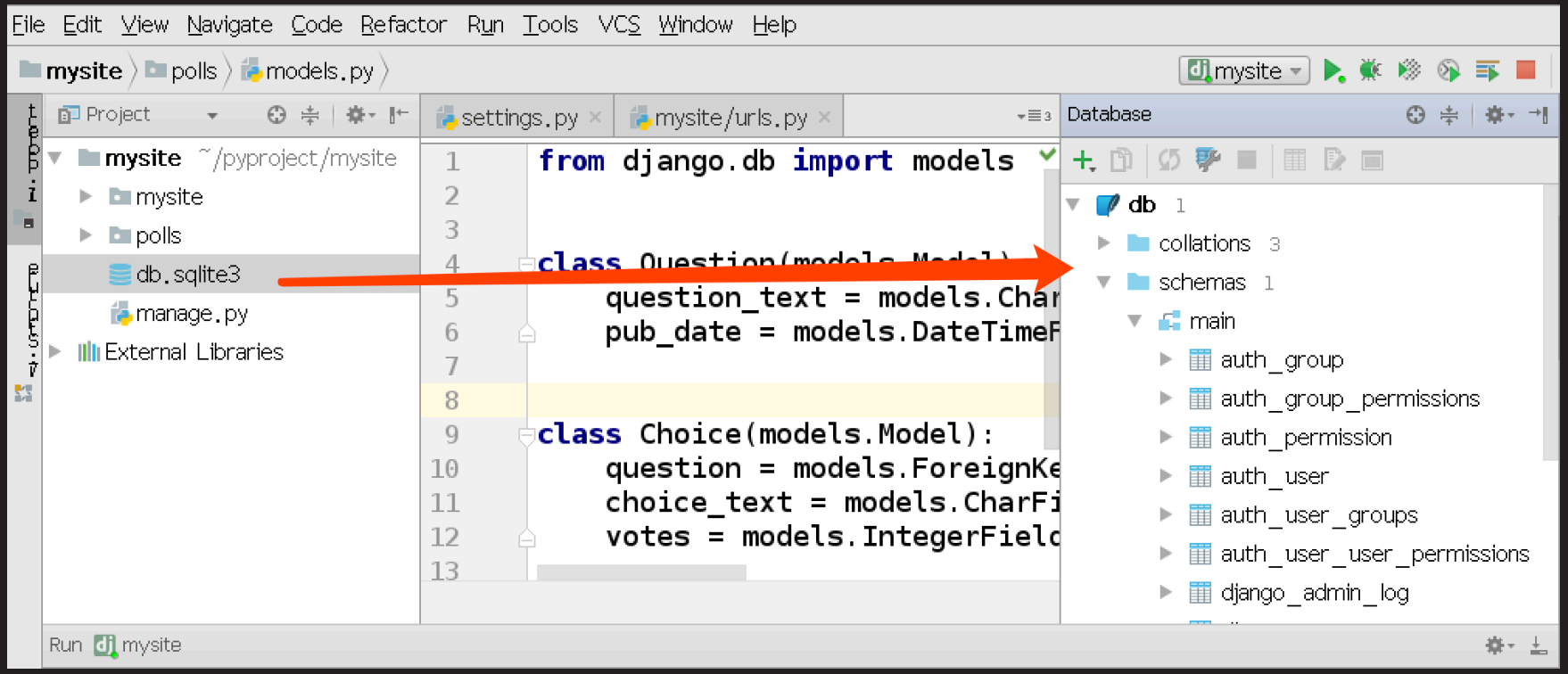
- 通过pycharm的Database窗口查看数据库



知识讲解

查看数据库（续1）

知识讲解



案例5：生成数据库

- 执行迁移命令生成数据库
- 在pycharm中打开数据库
- 观察数据库表名和字段名，发现与模型的关系

课堂练习

注册管理后台

- 投票的问题和选项，可以通过后台进行录入、编辑
- 只要把模型注册到后台管理界面即可
- 编辑polls/admin.py，修改如下：

```
from django.contrib import admin
from .models import Question, Choice

admin.site.register(Question)
admin.site.register(Choice)
```

知识讲解

注册管理后台（续1）

知识讲解

案例6：注册后台

- 将Question和Choice模型添加到后台管理
- 在后台页中填写几个问题和选项

课堂练习

Tedu.cn
达内教育

总结和答疑