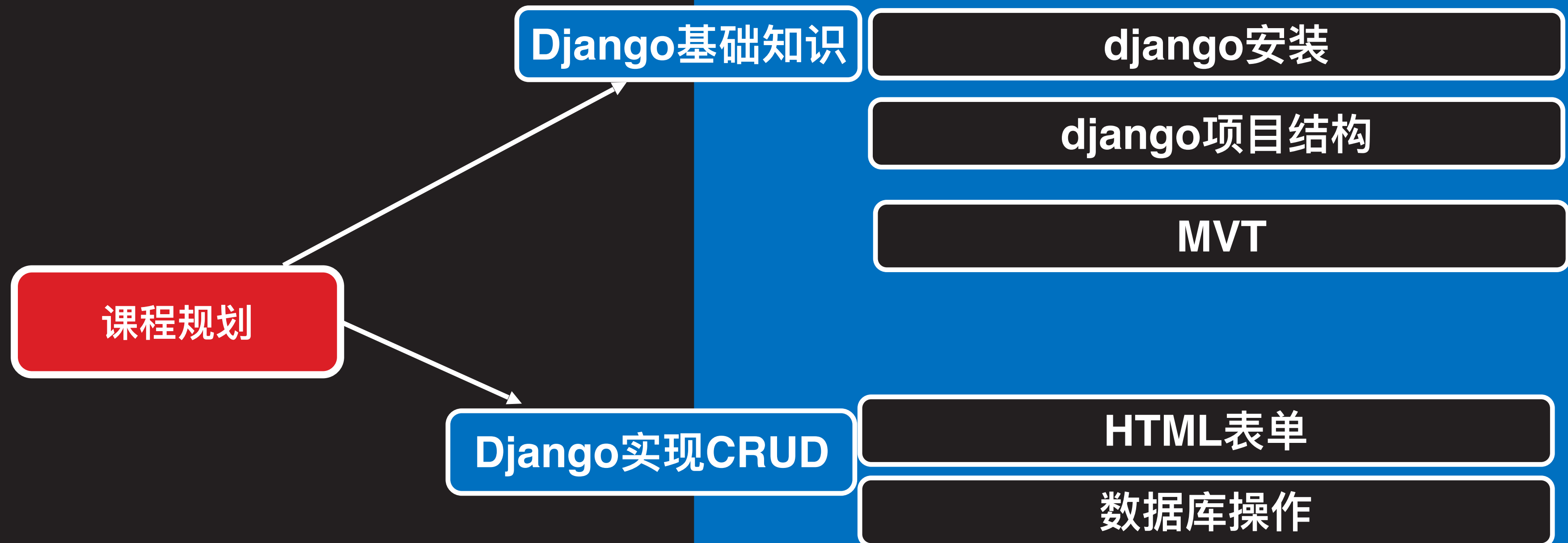


Python开发 Django基础

主题



Django安装

Python3环境准备

```
yum install zlib-devel openssl openssl-  
devel readline-devel mariadb-devel sqlite-devel  
libxml2-devel libxslt-devel
```

```
./setup.sh
```



Django 安装

```
pip3.6 install Django-1.9.6.tar.gz
```

代码
实践

或者解压后执行：

```
python3 setup.py install
```



Django 安装检查

```
python3 -m django version
```

代码
实践



Django项目建立

Django 初始化项目

django-admin startproject site01

代码
实践

```
|—— manage.py
|—— site01
    |—— __init__.py
    |—— settings.py
    |—— urls.py
    |—— wsgi.py
```



创建blog应用

```
cd site01  
python3 manage.py startapp blog
```

代码
实践



blog应用配置-注册应用

site01/settings.py

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

代码
实践



blog应用配置-URL配置1

site01/urls.py

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

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



blog应用配置-URL配置2

创建blog/urls.py

```
from django.conf.urls import url  
from .views import index
```

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



创建blog/views.py

```
from django.shortcuts import render, HttpResponse
```

```
def index(request):  
    retrun HttpResponse("hello")
```

代码
实践



启动服务

`python3 manage.py runserver`

浏览器访问: `http://127.0.0.1:8000/blog/`



创建django项目

- 1、创建django项目
- 2、添加一个应用
- 3、配置路由并访问应用首页

MVT



模板目录: `mkdir blog/templates`

静态文件目录: `mkdir blog/static`

模板文件: `touch blog/templates/index.html`

内容参考 `site196/blog/templates/index.html`



创建blog/views.py

```
from django.shortcuts import render, HttpResponse
```

```
def index(request):  
    retrun render(request,"index.html")
```

视图解析

代
码
实
践

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{% block title %} Default Title {% endblock %}</title>
  <link rel="stylesheet" href="{% static 'bootstrap/css/bootstrap.min.css' %}">
  {% block css %}
  {% endblock %}
</head>
<body>
  <div class="container">
    {% block content %}
    {{ variable }}
    {% endblock %}
    <ul>
      {% for item in items %}
        <li>{{ item }}</li>
      {% endfor %}
    </ul>
  </div>
</body>
</html>
```



Model

```
from django.db import models
import django.utils.timezone as timezone
```

```
class Post(models.Model):
    title = models.CharField(max_length=100)
    desc = models.CharField(max_length=500)
    content = models.TextField()
    author = models.CharField(max_length=50)
    add_date = models.DateTimeField('添加日期', default=timezone.now)
    mod_date = models.DateTimeField('最后修改日期', auto_now=True)
```

```
    def __str__(self):
        return self.title
```

```
class Comments(models.Model):
    post = models.ForeignKey(Post, on_delete=models.CASCADE)
    content = models.TextField(max_length=500)
```

```
    def __str__(self):
        return self.content
```



admin 注册model

blog/admin.py

```
from django.contrib import admin
```

```
# Register your models here.
```

```
from .models import Post, Comments
```

```
admin.site.register(Post)
```

```
admin.site.register(Comments)
```



数据库连接配置

创建数据库: `mysql>create database site01;`

site01/settings.py

```
DDATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'site01',  
        'USER': 'root',  
        'PASSWORD': 'root',  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
    }  
}  
# pip3 install pymysql  
site01/__init__.py  
import pymysql  
pymysql.install_as_MySQLdb()
```

代
码
实
践



执行migrate

```
python3 manage.py migrate  
python3 manage.py makemigrations  
python3 manage.py migrate
```



执行migrate/创建admin帐号

```
python3 manage.py migrate  
python3 manage.py makemigrations  
python3 manage.py migrate
```

代
码
实
践

```
python3 manage.py createsuperuser
```



View 读取数据

```
def index(request):  
    lstPost = Post.objects.all()  
    return render(request, "index.html", {'posts': posts})
```

模板继承-基础模板

代码
实践

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %} Default Title {% endblock %}</title>
    <link rel="stylesheet" href="{% static 'bootstrap/css/bootstrap.min.css' %}">
    {% block css %}
    {% endblock %}
</head>
<body>
    <div class="container">
        {% block content %}
        {% endblock %}
    </div>
</body>
</html>
```



参考base.html

模板继承-继承模板

代码实践

```
{% extends "base.html" %}
{% block content %}
    <a href="/blog/add" class="btn btn-primary">添加</a>
<table class="table table-striped table-bordered">
    <tr>
        <td>标题</td>
        <td>描述</td>
        <td>作者</td>
        <td>修改时间</td>
        <td>操作</td>
    </tr>
    {% for p in posts %}
        <tr>
            <td>{{ p.title }}</td>
            <td>{{ p.desc }}</td>
            <td>{{ p.author }}</td>
            <td>{{ p.mod_date| date:"Y-m-j" }}</td>
            <td>
                <a href="/blog/edit/{{ p.id }}" class="btn btn-success">编辑</a>
                <a href="/blog/del?id={{ p.id }}" class="btn btn-danger">删除</a>
            </td>
        </tr>
    {% endfor %}
</table>
{% endblock %}
```

参考list.html



练习MVT

- 1、新建model模式并使用admin管理
- 2、将admin中添加的数据在前端显示

CRUD

数据列表

代码实践

```
<table class="table table-striped table-bordered">
  <tr>
    <td>标题</td>
    <td>描述</td>
    <td>作者</td>
    <td>修改时间</td>
    <td>操作</td>
  </tr>
{% for p in posts %}
  <tr>
    <td>{{ p.title }}</td>
    <td>{{ p.desc }}</td>
    <td>{{ p.author }}</td>
    <td>{{ p.mod_date| date:"Y-m-j" }}</td>
    <td>
      <a href="/blog/edit/{{ p.id }}" class="btn btn-success">编辑</a>
      <a href="/blog/del?id={{ p.id }}" class="btn btn-danger">删除</a>
    </td>
  </tr>
{% endfor %}
</table>
```



数据列表

```
def showList(request):
```

```
    posts = Post.objects.all()
```

```
    return render(request, "list.html", {'posts': posts})
```

代码
实践



表单

代码实践

```
<form class="form-horizontal" action="{{ action }}" method="post">
  <div class="form-group">
    <label for="inputEmail3" class="col-sm-2 control-label">标题</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" value="{{ post.title }}" name="title"
placeholder="标题">
    </div>
  </div>
  .....
</div>
```

```
{% csrf_token %}
<div class="form-group">
  <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-default">保存</button>
  </div>
</div>
</form>
```



参考edit.html

表单取值

```
request.POST.get('title')  
request.GET.get("id")
```

代码
实践



添加Post数据

```
post = Post()
post.title = request.POST.get('title')
post.desc = request.POST.get('desc')
post.content = request.POST.get('content')
post.save()
```



修改Post

```
post = Post.objects.get(id=id)
post.title = request.POST.get('title')
post.desc = request.POST.get('desc')
post.content = request.POST.get('content')
post.save()
```



删除Post

```
post = Post.objects.get(id=request.GET.get("id"))  
post.delete()
```

代码
实践



ajax删除

```
$.ajax({  
    type: "GET",  
    url: "/blog/ajaxdel?id=1",  
    dataType: "json"  
});
```



练习CRUD

1、页面实现对数据库的增删改查

扩展：使用ajax提交回复