

# 1. 框架

## MVC

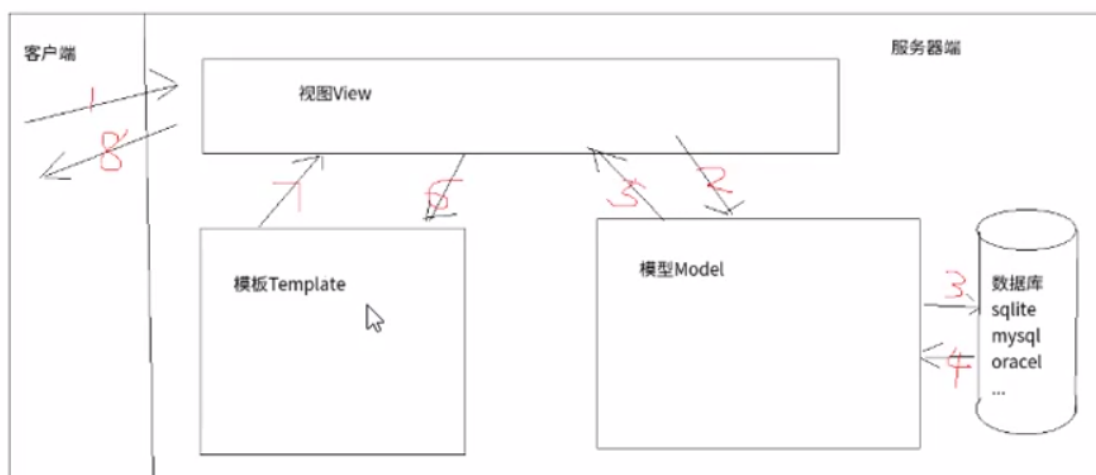
### MVC

- 大部分开发语言中都有MVC框架
- MVC框架的核心思想是：解耦
- 降低各功能模块之间的耦合性，方便变更，更容易重构代码，最大程度上实现代码的重用
- m表示model，主要用于对数据库层的封装
- v表示view，用于向用户展示结果
- c表示controller，是核心，用于处理请求、获取数据、返回结果

## MTV

### MVT

- Django是一款python的web开发框架
- 与MVC有所不同，属于MVT框架
- m表示model，负责与数据库交互
- v表示view，是核心，负责接收请求、获取数据、返回结果
- t表示template，负责呈现内容到浏览器



## 2. 内容简介

# 简介

- 通过简单示例，使用django完成基本流程的开发，学习django的主要的知识点，在后续课程中会逐个知识点进行深入讲解
- 以“图书-英雄”管理为示例



## 主要知识点介绍

- 环境搭建
- 定义模型
- 使用后台管理
- 编写视图
- 定义模板

## 主要知识点介绍

- 环境搭建
- 定义模型
- 使用后台管理
- 编写视图
- 定义模板

## 3. 虚拟环境的创建

---

### 什么是virtualenv?

- virtualenv是一个用于创建Python虚拟环境的工具。
- 虚拟环境又是什么呢？它是Python解释器的一个私有副本，在这个环境中可以安装私有包，而不会影响到系统中安装的全局Python解释器。

- 当一台主机上运行着多个Python应用时，使用虚拟环境可以有效的避免包的混乱和版本的冲突。

## 创建虚拟环境？

```
virtualenv -p python3 venv
```

## 激活和退出虚拟环境

```
source venv/bin/activate  
deactivate
```

## 在虚拟环境中安装依赖？

```
pip freeze  
pip3 install -r requirements.txt
```

## 删除虚拟环境？

```
rmvirtualenv venv
```

## 创建虚拟环境

- 创建：mkvirtualenv [虚拟环境名称]
- 删除：rmvirtualenv [虚拟环境名称]
- 进入：workon [虚拟环境名称]
- 退出：deactivate
- 所有的虚拟环境，都位于/home/.virtualenvs目录下
- 进入虚拟环境前的提示：

```
python@ubuntu:~$ workon h1
```

- 进入虚拟环境后的提示：

```
(h1) python@ubuntu:~$
```

- 查看当前的所有虚拟环境：workon [两次tab键]
- 查看虚拟环境中已经安装的包

```
pip list  
pip freeze
```


## 4. 创建项目

- 查看django版本的方法

```
python -m django --version
```

### 创建项目

- 命令django-admin startproject test1
- 进入test1目录，目录结构如下图：



```
├── manage.py
└── test1
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

### < 目录说明

- manage.py: 一个命令行工具，可以使你用多种方式对Django项目进行交互
- 内层的目录: 项目的真正的Python包
- \_\_init\_\_.py: 一个空文件，它告诉Python这个目录应该被看做一个Python包
- settings.py: 项目的配置
- urls.py: 项目的URL声明
- wsgi.py: 项目与WSGI兼容的Web服务器入口