# 1.0初始环境

## 1.1工具安装

python、pycharm、mysql、navicat、pymysql

## 1.2虚拟环境

virtualenv

virtualenv xxx-env

activate xxx-env

deactivate

virtualenvwrapper

创建虚拟环境

mkvirtualenv xxx-env

运行虚拟环境

workon xxx-env

退出虚拟环境

deactivate

删除虚拟环境

rmvirtualenv xxx-env

列出虚拟环境

lsvirtualenv

进入虚拟环境工作目录

cdvirtualenv

## 1.3安装Django

pip install diango==2.0

## 1.4创建实例（Django项目）

命令行创建

django-admin startproject [项目名称]

运行项目

进入项目根目录

python manage.py runserver

prcharm创建

新建项目->Django（注意选择编译环境）

## 1.5 Django目录解释

manage.py 负责和项目交互

settings.py 本项目的设置项

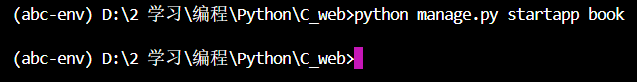
urls.py 项目的URL路径配置文件

wsgi.py 项目与WSGI协议兼容的web服务器入口，部署是用到

## 1.6 project和app

app是项目的组成部分，一个app代表项目中的一个模块所有URL请求的响应都是都是由app来处理的

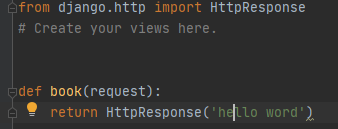
创建app



# 2.0 URL与视图

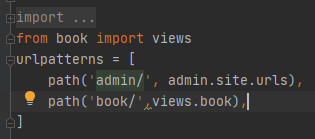
## 2.1视图

视图一般都写在app的views.py中。视图的第一参数永远是request



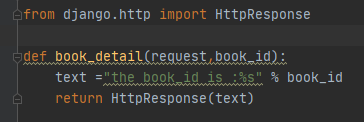
## 2.2 URL映射

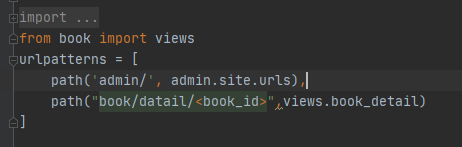
在urls.py文件中做URL映射，在用户输入URL后，做出响应



## 2.3 URL中添加参数

用来表示当前URL下的资源（参数一般为资源对应ID）





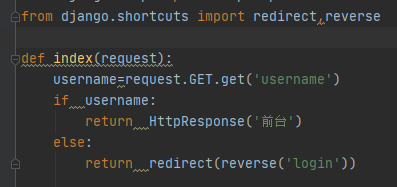
传递多参数

## 2.4 URL命名

原因：修改URL繁琐，为URL命名后，在使用URL的地方使用name反转之后即可

做法：在path中传递一个name参数

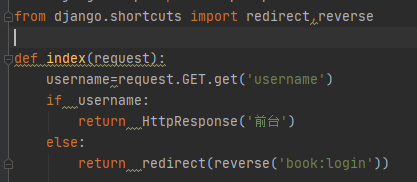




应用命名空间：name参数可能出现多个，导致URL调用混乱

做法：在app的urls.py中定义app\_name = ‘book’



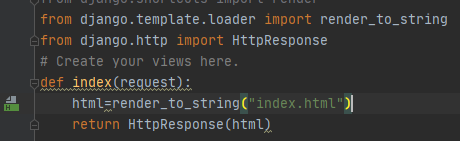


# 3.0模板

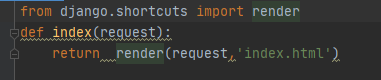
现实中呈现的网站都是经过HTML语言经过渲染实现的，DTL（Django Template Language）一种Django自带的，具有特殊语法的HTML文件，这个文件可被Django编译

## 3.1渲染模板

通过render\_to\_string找到模板，将模板编译后渲染成python字符串格式，再通过HttpResponse类包装进行返回



直接通过django.shortcuts类里的render直接实现



## 3.3模板查找路径

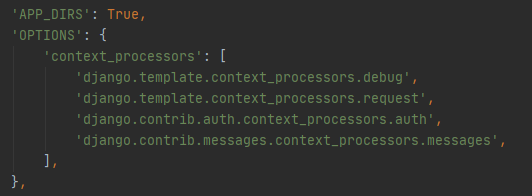
在settings.py文件中，有一个TEMPLATES配置，此配置包含了模板引擎、模板查找路径、模板上下文

模板查找路径存在两种添加方法：

DIRS中查找（默认优先查找DIRS中的路径）



APP\_DIRS在APP中查找模板文件true表示开启（若要实现在APP中查找则需要添加APP）



## 3.4 DTL模板变量

# 项目实践

## 环境配置

nvm安装

nvm（Node Version Manager）用来管理node.js

node安装