

我们有的时候会发现，一个电脑上有多个项目，一个依赖 Django 1.8，另一个比较旧的项目又要用 Django 1.5，这时候怎么办呢？

我们需要一个依赖包管理的工具来处理不同的环境。

如果不想搭建这个环境，只想用某一个版本的 Django 也可以，但是推荐学习此内容！

4.1 环境搭建

开发会用 virtualenv 来管理多个开发环境，virtualenvwrapper 使得virtualenv变得更好用

```
1 # 安装:
2 (sudo) pip install virtualenv virtualenvwrapper
```

Linux/Mac OSX 下:

修改 ~/.bash_profile 或其它环境变量相关文件(如 .bashrc 或用 ZSH 之后的 .zshrc)，添加以下语句

```
1 export WORKON_HOME=$HOME/.virtualenvs
2 export PROJECT_HOME=$HOME/workspace
3 source /usr/local/bin/virtualenvwrapper.sh
```

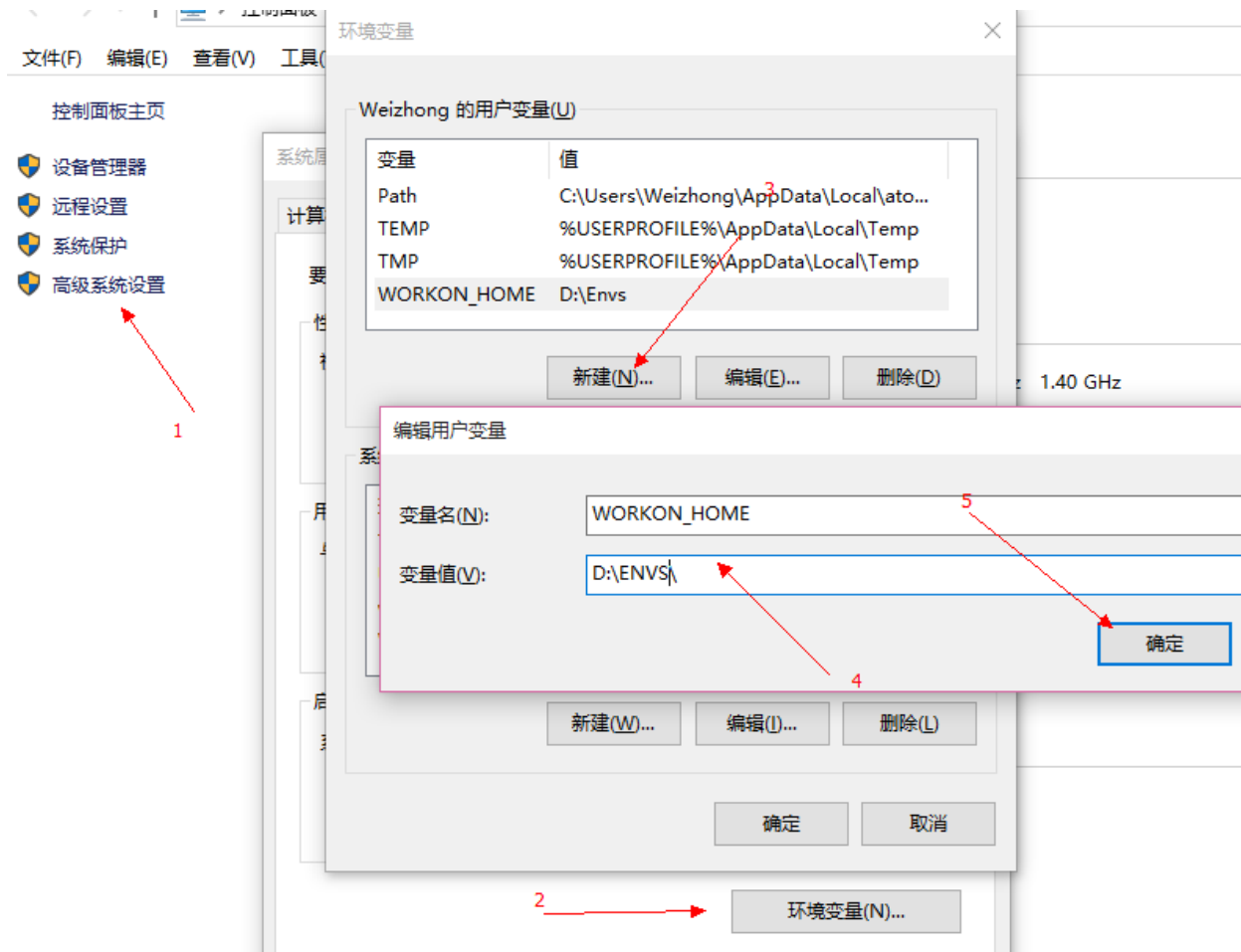
修改后使之立即生效(也可以重启终端使之生效):

```
1 source ~/.bash_profile
```

Windows 下:

```
1 pip install virtualenvwrapper-win
```

【可选】Windows下默认虚拟环境是放在用户名下面的Envs中的，与桌面，我的文档，下载等文件夹在一块的。更改方法：计算机，属性，高级系统设置，环境变量，添加WORKON_HOME，如图（windows 10 环境变量设置截图）：



workon_home.png

4.2 使用方法:

mkvirtualenv zqxt: 创建运行环境zqxt

workon zqxt: 工作在 zqxt 环境 或 从其它环境切换到 zqxt 环境

deactivate: 退出终端环境

其它的:

rmvirtualenv ENV: 删除运行环境ENV

mkproject mic: 创建mic项目和运行环境mic

mktmpenv: 创建临时运行环境

lsvirtualenv: 列出可用的运行环境

lsitepackages: 列出当前环境安装了的包

创建的环境是独立的，互不干扰，无需sudo权限即可使用 pip 来进行包的管理。