# virenvs简要说明

## virtualenv

用于创建独立的Python环境，多个Python相互独立，互不影响，它能够：  
1. 在没有权限的情况下安装新套件  
2. 不同应用可以使用不同的套件版本  
3. 套件升级不影响其他应用

### 一、安装

**sudo apt-get install python-virtualenv**

### 二、使用方法

**virtualenv [虚拟环境名称]**

----------------------------------------------

默认情况下，虚拟环境会依赖系统环境中的site packages，就是说系统中已经安装好的第三方package也会安装在虚拟环境中，如果不想依赖这些package，那么可以加上参数 --no-site-packages建立虚拟环境

**virtualenv --no-site-packages [虚拟环境名称]**

### 三、退出虚拟环境

**deactivate**

在虚拟环境安装Python套件

Virtualenv 附带有pip安装工具，因此需要安装的套件可以直接运行：

pip install [套件名称]

如果没有启动虚拟环境，系统也安装了pip工具，那么套件将被安装在系统环境中，为了避免发生此事，可以在~/.bashrc文件中加上：

export PIP\_REQUIRE\_VIRTUALENV=true

或者让在执行pip的时候让系统自动开启虚拟环境：

export PIP\_RESPECT\_VIRTUALENV=true

## Virtualenvwrapper

Virtaulenvwrapper是virtualenv的扩展包，用于更方便管理虚拟环境，它可以做：  
1. 将所有虚拟环境整合在一个目录下  
2. 管理（新增，删除，复制）虚拟环境  
3. 切换虚拟环境

### 1、安装

（1）**sudo easy\_install virtualenvwrapper**

**-------------------------------------------------------**

此时还不能使用virtualenvwrapper，默认virtualenvwrapper安装在/usr/local/bin下面，实际上你需要运行virtualenvwrapper.sh文件才行，打开这个文件看看,里面有安装步骤，我们照着操作把环境设置好。

（2）**创建目录用来存放虚拟环境**

mkdir $HOME/.virtualenvs

（3）在~/.bashrc中添加行： export WORKON\_HOME=$HOME/.virtualenvs

（4）在~/.bashrc中添加行：source /usr/local/bin/virtualenvwrapper.sh

**#这里目录位置要进行确认find / -name virtualenvwrapper.sh**

（5）运行： source ~/.bashrc

### 2、命令

workon/lsvirtualenv#列出虚拟环境列表

mkvirtualenv [虚拟环境名称] #新建虚拟环境

workon [虚拟环境名称] #启动/切换虚拟环境

rmvirtualenv [虚拟环境名称] #删除虚拟环境

deactivate #离开虚拟环境