[python三大神器之virtualenv](https://www.cnblogs.com/freely/p/8022923.html)

    pip, virtualenv, fabric通称为pythoner的三大神器。

**Virtualenv**

**Windows下，bin==scripts ,没有source命令，直接activate使用。**

    virtualenv------用来建立一个虚拟的python环境，一个专属于项目的python环境。用virtualenv 来保持一个干净的环境非常有用

测试环境：linux下

**1、基本使用**

    通过pip安装virtualenv：

pip install virtualenv

    测试安装:

virtualenv --version

    为一个工程项目搭建一个虚拟环境:

cd my\_project

virtualenv my\_project\_env

    另外，如果存在多个python解释器，可以选择指定一个Python解释器（比如``python2.7``），没有指定则由系统默认的解释器来搭建：

virtualenv -p /usr/bin/python2.7 my\_project\_env

    将会在当前的目录中创建一个名my\_project\_env的文件夹，这是一个独立的python运行环境，包含了Python可执行文件， 以及 pip 库的一份拷贝，这样就能安装其他包了，不过已经安装到系统Python环境中的所有第三方包都不会复制过来，这样，我们就得到了一个不带任何第三方包的“干净”的Python运行环境来。

 要开始使用虚拟环境，其需要被激活：

source my\_project\_env/bin/activate

    停用虚拟环境：

deactivate

    停用后将回到系统默认的Python解释器

**2、其他**

    用pip freeze查看当前安装版本

pip freeze

    另外：

pip freeze > requirements.txt

    这将会创建一个 requirements.txt 文件，其中包含了当前环境中所有包及 各自的版本的简单列表。您可以使用 “pip list”在不产生requirements文件的情况下， 查看已安装包的列表。这将会使另一个不同的开发者（或者是您，如果您需要重新创建这样的环境） 在以后安装相同版本的相同包变得容易。

pip install -r requirements.txt

    这能帮助确保安装、部署和开发者之间的一致性。

**virtualenvwrapper**

提供了一系列命令使得和虚拟环境工作变得愉快许多。它把您所有的虚拟环境都放在一个地方。

1. 将您的所有虚拟环境在一个地方。
2. 包装用于管理虚拟环境（创建，删除，复制）。
3. 使用一个命令来环境之间进行切换。

**1、安装**

    安装（确保 **virtualenv** 已经安装了）：

pip install virtualenvwrapper

export WORKON\_HOME=~/Envs #设置环境变量

mkdir -p $WORKON\_HOME #创建虚拟环境管理目录

find / -name virtualenvwrapper.sh #找到virtualenvwrapper.sh的路径

source 路径 #激活virtualenvwrapper.sh

    默认virtualenvwrapper安装在下面python解释器中的site-packages，实际上需要运行virtualenvwrapper.sh文件才行；所以需要先进行配置一下：

* 找到virtualenvwrapper.sh的路径：find / -name virtualenvwrapper.sh
* 运行virtualenvwrapper.sh文件：source 路径

**ps：每次要想使用virtualenvwrapper 工具时，都必须先激活virtualenvwrapper.sh,另外，如果创建前要将即将的环境保存到Envs中，就要先设置一下环境变量：**export WORKON\_HOME=~/Envs，再搭建

对于Windows，您可以使用 virtualenvwrapper-win

    安装（确保 **virtualenv** 已经安装了）：

pip install virtualenvwrapper-win

在Windows中，WORKON\_HOME默认的路径是 %USERPROFILE%Envs 。

**2、基本使用**

1、创建一个虚拟环境：

mkvirtualenv project\_env

   这会在Envs 中创建 project\_env虚拟环境

　  选择一个python解释器来搭建：

mkvirtualenv env --python=python2.7

2、在虚拟环境上工作：

workon project\_env

或者，您可以创建一个项目，它会创建虚拟环境，并在 $WORKON\_HOME 中创建一个项目目录。 当您使用 workon project\_env 时，会 cd -ed 到项目目录中。

mkproject project\_env

**virtualenvwrapper** 提供环境名字的tab补全功能。当您有很多环境， 并且很难记住它们的名字时，这就显得很有用。

workon 也能停止您当前所在的环境，所以您可以在环境之间快速的切换。

3、停止虚拟环境

deactivate

4、删除：

rmvirtualenv project\_env

**3、其他有用的命令**

[复制代码](javascript:void(0);)

lsvirtualenv #列举所有的环境。

cdvirtualenv #导航到当前激活的虚拟环境的目录中，比如说这样您就能够浏览它的 site-packages。

cdsitepackages # 和上面的类似，但是是直接进入到 site-packages 目录中。

lssitepackages #显示 site-packages 目录中的内容。

[复制代码](javascript:void(0);)

[virtualenvwrapper 命令的完全列表](https://virtualenvwrapper.readthedocs.io/en/latest/command_ref.html) 。

参考：http://pythonguidecn.readthedocs.io/zh/latest/dev/virtualenvs.html