包管理和虚拟环境

包管理

安装第三方包的方法有以下三种：

1. 通过Python社区开发的pip、easy_install等工具
2. 使用系统本身自带的包管理器（yum, apt-get等）
3. 通过源码安装（python setup.py install）

easy_install 是由PEAK(Python Enterprise Application Kit)开发的setuptools包里带的一个命令，所以使用easy_install实际上是在调用setuptools来完成安装模块的工作

easy_install的缺点

1. easy_install只支持安装，没有提供卸载、展示当前已安装的包列表等功能
2. 不能集中管理项目依赖列表，就是包得一个一个安装
3. 安装的包不能缓存使用，每次都要下载甚至编译
4. 终端上的输出不够友好
5. easy_install提供的Python应用打包部署方式(egg)已经落伍了

pip的优势

1. pip已经内置到Python 2.7.9和Python 3.4及其以上的版本里面。
2. easy_install只支持安装，没有提供卸载、展示当前已安装的包列表等功能。
3. pip支持二进制包使用wheel格式（后缀是.whl），而easy_install不支持。
4. pip能非常好地支持虚拟环境工具virtualenv。
5. 支持多种版本工具格式的包的下载和安装。
6. 可以集中管理项目依赖列表（文件名字一般叫作requirements.txt），使用-r选项安装这些依赖。
7. 更好的终端输出效果
8. 可以对下载的包进行缓存，下次直接从缓存目录取而不需要下载了。

```
❯ sudo apt-get install python-pip -yq
❯ sudo pip install pip -U -q  # -q表示静默安装, 减少过程输出
❯ pip --version
pip 9.0.3 from /usr/local/lib/python3.6/site-packages (python 3.6)
```

虚拟环境 - virtualenv

```
❯ sudo pip install virtualenv

❯ virtualenv venv
New python executable in /home/dongwm/venv/bin/python
Installing setuptools, pip, wheel...done.

❯ source venv/bin/activate

(venv)❯ which python
/home/dongwm/venv/bin/python

(venv)❯ deactivate
```

创建虚拟环境 - venv模块

```
python3 -m venv /path/to/new/virtual/environment

pyvenv /path/to/new/virtual/environment
# Python 3.6开始 pyvenv 脚本已经不可用
```

pipenv

1. 根据 Pipfile 自动寻找项目根目录。
2. 如果不存在，可以自动生成 Pipfile 和 Pipfile.lock。
3. 自动在项目目录的 .venv 目录创建虚拟环境。（当然这个目录地址通过设置WORKON_HOME改变）
4. 自动管理 Pipfile 新安装和删除的包。
5. 自动更新 pip。

autoenv

```
❯ sudo pip install autoenv

❯ source /usr/local/bin/activate.sh

❯ mkdir test

❯ cd test

❯ touch .env

❯ echo "source /home/dongwm/venv/bin/activate" > .env

❯ cd  # 先切换到其他目录
❯ cd test  # 出现如下提示
autoenv:
autoenv: WARNING:
autoenv: This is the first time you are about to source /home/dongwm/test/.env:
autoenv:
autoenv:     --- (begin contents) ----------------------------------------
autoenv:     source /home/dongwm/venv/bin/activate
autoenv:
autoenv:     --- (end contents) ------------------------------------------
autoenv:
autoenv: Are you sure you want to allow this? (y/N) y
(venv) ❯  #
```

不推荐pyenv的理由

1. 它用来解决切换Python版本问题，其实只不过是切换虚拟环境的另外一种途径，并没有存在的必要性
2. 如果同Python版本多个虚拟环境情况它是没有办法解决的，还得用虚拟环境，不如一开始都统一用虚拟环境
3. Python社区无感，它不是官方接受的解决方案，没有社区强力支持

延伸阅读

1. https://www.python.org/dev/peps/pep-0405/
2. https://docs.python.org/3/library/venv.html
3. https://zhuanlan.zhihu.com/p/32913361