

## virtualenvwrapper打造多版本Python环境



木制robot  
Python、Popping 爱好者

13 人赞了该文章

### 前言

面对多个 Python 开发项目时，需要针对不同的项目创建相应的开发环境。通常情况下，使用 virtualenv 创建一个虚拟的独立 Python 环境，但是 virtualenv 创建的环境相对分散不便于管理。这里推荐使用 virtualenvwrapper 来创建集中的便于管理的 Python 环境，同时可以结合 pyenv 为不同的项目选定不同的 Python 版本。

### virtualenvwrapper

#### 安装

通过 pip 安装 virtualenvwrapper (Ps: 记住加上 sudo) 。

```
$ sudo pip install virtualenvwrapper
```

#### 配置

在.bashrc(或.zshrc)中添加以下内容：

```
export WORKON_HOME=$HOME/.virtualenvs  
source /usr/local/bin/virtualenvwrapper.sh
```

接着执行`source ~/.bashrc(或./zshrc)`。

注意：修改.bashrc还是.zshrc取决于你使用的那种 shell。

#### 使用

- workon : 打印所有的虚拟环境；
- mkvirtualenv xxx : 创建 xxx 虚拟环境；
- workon xxx : 使用 xxx
- deactivate : 退出 xxx

▲ 赞同 13 ▼

● 5 条评论

➦ 分享

★ 收藏

...



- `rmvirtualenv xxx` : 删除 xxx 虚拟环境。

## pyenv

### 安装

#### Linux

使用下列命令自动安装 pyenv:

```
$ curl -L https://raw.githubusercontent.com/pyenv/pyenv-installer/master/bin/pyenv-installer.sh | bash
```

#### Mac OS X

使用 homebrew 直接安装:

```
$ brew update
$ brew install pyenv
```

### 配置

在.bashrc(或.zshrc)中添加下列内容:

```
export PYENV_ROOT="$HOME/.pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init -)"
```

然后重启 shell, 使配置生效。

```
$ exec "$SHELL"
```

### 安装 Python 时依赖环境

安装 Python 时需要提前安装好所需的依赖环境。

- Ubuntu/Debian:

```
sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev \
libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-dev \
xz-utils tk-dev libffi-dev liblzma-dev
```

- Fedora/CentOS/RHEL:

```
sudo yum install zlib-devel bzip2 bzip2-devel readline-devel sqlite sqlite-devel \
openssl-devel xz xz-devel libffi-devel
```

Alternative of openssl-devel:

```
sudo yum install compat-openssl10-devel --allowdowngrade
```

- openSUSE

```
zypper in zlib-devel bzip2 libbz2-devel readline-devel sqlite3 sqlite3-devel l
```

- macOS:

```
brew install readlin
```

## 使用

### 1. 查看所有 Python 版本

```
$ pyenv versions
```

执行命令将打印出安装的所有 Python 版本，\* 表示当前使用的 Python 版本。

### 2. 查询所有可安装的 Python 版本

```
$ pyenv install -l
```

### 3. 安装与卸载

```
$ pyenv install 2.7.14
```

安装 Python 2.7.14 版本。

```
$ pyenv uninstall 2.7.14
```

卸载 Python 2.7.14 版本

```
$ pyenv rehash
```

创建垫片路径（为所有已安装的可执行文件创建 shims，如：~/.pyenv/versions/bin-，因此，每当你增删了 Python 版本或带有可执行文件的包（如 pip）以后，都应该执行一次本命令）。

### 4. 设置 Python 版本

```
$ pyenv shell 2.7.14 # 设置面向 shell 的 Python 版本，通过设置当前 shell 的 PYENV_I
$ pyenv local 2.7.14 # 设置 Python 本地版本，通过将版本号写入当前目录下的 .python-ver
$ pyenv global 2.7.14 # 设置全局的 Python 版本，通过将版本号写入 ~/.pyenv/version 文
```

注意：shell > local > global。pyenv 会从当前目录开始向上逐级查找 .python-version 文件，直到根目录为止。若找不到，就用 global 版本。

Ps: 因为系统本身常常会依赖自带的 python 版本，所以尽量不要修改 global。

### 5. 取消 shell 或 local python 版本

```
$ pyenv shell --unset
$ pyenv local --unset
```

## 使用镜像加速下载

通过下列脚本使用国内搜狐的镜像快速安装各版本 Python：

```
v=3.6.3|wget http://mirrors.sohu.com/python/$v/Python-$v.tar.xz -P $(pyenv root)
```

注意：v 为 Python 版本号，请根据需要自行更改。

搜狐镜像貌似关闭了，可以自行下载指定的 Python 版本安装包放入，\$(pyenv root)/cache/文件夹下，然后执行 `pyenv install xxx(版本号)` 安装相应的 python 版本，安装时会使用缓存安装包。

## pyenv-virtualenvwrapper

### 安装

Linux

▲ 赞同 13 ▼    5 条评论    分享    ★ 收藏    ...

通过下列命令安装 pyenv-virtualenvwrapper:

```
$ git clone https://github.com/pyenv/pyenv-virtualenvwrapper.git $(pyenv root),
```

Mac OS X

Mac OS X 下可选择使用使用 Homebrew 来安装:

```
$ brew install pyenv-virtualenvwrapper
```

激活

安装完成后使用下列命令激活 pyenv-virtualenvwrapper:

```
$ pyenv virtualenvwrapper
```

使用

```
$ pyenv shell 3.6.3 # 设置当前 shell python 版本为 2.7.14
$ pip install virtualenvwrapper # 第一次使用新的 Python 环境需要安装此包, 否则创建的虚
$ mkvirtualenv python3 # 创建 python 3.6.3 版本的虚拟环境 python3
```

参考

- 1. [pyenv 文档](#)
- 2. [virtualenvwrapper 文档](#)
- 3. [pyenv-virtualenvwrapper 文档](#)
- 4. [使用 pyenv 管理 Python 版本](#)
- 5. [pyenv 安装配置与国内镜像加速](#)

编辑于 2018-09-13

Python virtualenv

文章被以下专栏收录

 Wooden Robot技术杂谈

进入专栏

推荐阅读

在 Linux 下安装基于 Python 的机器学习编程环境

试用了一下 EPD 的Canopy, 感觉不错。Anaconda由于不知名的原因, 在我的笔记本上装成功了, 但是没有运行起来, 可能是QT的原因吧。Anyway, 到现在, 我都不推荐了。还是坚持原来的方法, 因为...

肖智博 发表于文本挖掘研...



Python 大神 kennethreitz 又搞事了

刘志军 发表于Pytho...

黄哥Python本周(2017-12-29)答疑总结

以下内容来源于黄哥本周在知乎为网友答疑的内容, 总结如下, 供大家参考。一、关于网站选型, 用Django重写我的网站? 1、90年代开发软件, 是基于 c/s , 胖客户端, 瘦服务端, 当年本人用delphi...

黄哥

赞同 13

5 条评论

分享

收藏

...

这或pyt


Tun

5 条评论

切换为时间排序


写下你的评论...

😊

 零一1 年前


pyenv不是自带创建虚拟环境的功能吗

👍 赞

 木制robot (作者) 回复 零一1 年前

pyenv 是管理 python 版本的工具，virtualenvwrapper 是创建虚拟 python 环境的工具，pyenv-virtualenvwrapper 是创建不同 python 版本虚拟环境的工具。

👍 赞

 零一 回复 木制robot (作者)1 年前

pyenv virtualenv [python版本] [自定义环境名称] #创建基于[python版本]的虚拟环境

pyenv activate [环境名称] #激活虚拟环境

pyenv deactivate #退出虚拟环境

当然也可以"pyenv local/global [环境名称] " 设置全局/当前目录为某个虚拟环境

👍 赞

展开其他 2 条回复