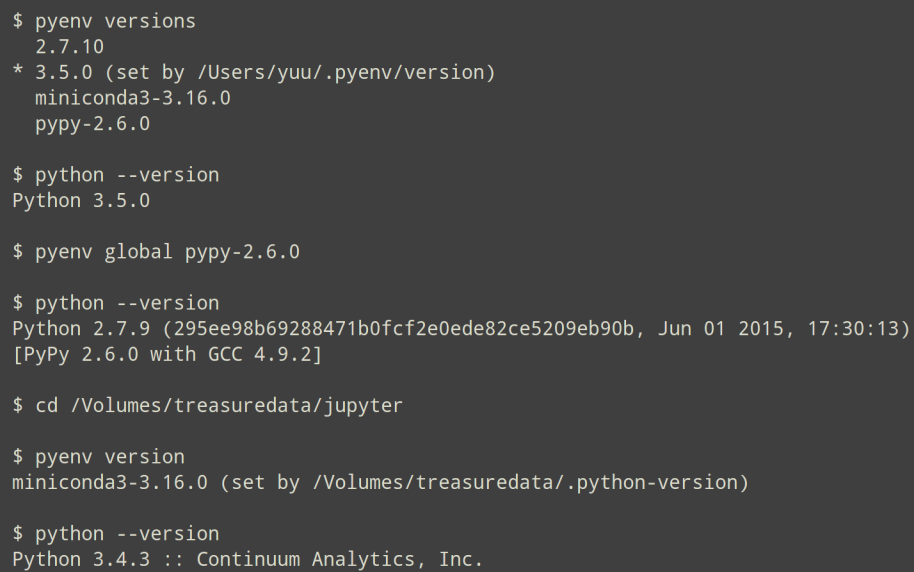


# 在 Linux 上控制 Python 版本 - Pyenv

Huang Po Hsun

2020 年 1 月 30 日



```
$ pyenv versions
2.7.10
* 3.5.0 (set by /Users/yuu/.pyenv/version)
  miniconda3-3.16.0
  pypy-2.6.0

$ python --version
Python 3.5.0

$ pyenv global pypy-2.6.0

$ python --version
Python 2.7.9 (295ee98b69288471b0fcf2e0ede82ce5209eb90b, Jun 01 2015, 17:30:13)
[PyPy 2.6.0 with GCC 4.9.2]

$ cd /Volumes/treasuredata/jupyter

$ pyenv version
miniconda3-3.16.0 (set by /Volumes/treasuredata/.python-version)

$ python --version
Python 3.4.3 :: Continuum Analytics, Inc.
```

# 1 前言

在 Linux 開發 Python 相關的項目，你是不是會碰到關於 Python 版本之類的問題？像是開發 Tensorflow 的時候碰到版本問題，**原本 Python 3.7 可以支援 Tensorflow，但是忽然 Python 從 3.7 更新到 3.8.0**，這時 Python 3.8 不支援 Tensorflow，你苦惱了，千辛萬苦的項目因為系統更新而導致版本不支援使項目暫停開發，這時 pyenv 將成為你的救星，如果你硬是不肯更新系統 Python 版本，你的系統將會得不到最新的體驗與安全並處於危險的不穩定狀態，系統更新真的很重要。

pyenv 是很棒的 Python 版本控制工具，讓你的電腦可以安裝多個 Python 版本。[pyenv](#) 是 Github 上的開源項目，關於使用須知該項目的 README.md 寫得很詳細。我這篇文章就分享一下我在 Arch Linux 的安裝方式。

# 2 安裝

從 Github 倉庫上直接 clone 下來。你也可以選擇直接在瀏覽器上下來。然後壓縮包解壓將文件內的文件放入 `~/.pyenv`。

## 2.1 設置代理

註：如果你在國內網使用 Github 網速過於緩慢，建議開 Proxy，然後給 [Git 設置代理](#)。

```
# set http
git config --global http.proxy 'socks5://127.0.0.1:1080'
# set https
git config --global https.proxy 'socks5://127.0.0.1:1080'
# unset http
git config --global --unset http.proxy
# unset https
git config --global --unset https.proxy

npm config delete proxy
```

Git 設置代理

## 2.2 下載

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv # Basic GitHub Checkout
```

## 3 環境變量

### 3.1 PYENV\_ROOT

將 PYENV\_ROOT 添加至環境變量。在 Konsole 輸入以下指令：

註：在 Arch 發行版中，`~/.bashprofile` 就是 `~/.profile`。如果你 Shell 使用的是 **ZSH**，為了在 **ZSH** 中也能使用，記得也要寫入 `~/.zshrc`。之所以要求要兩個都寫入是因為避免如果你換了其它 Shell 需要再重新設一次。

```
# .profile
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.profile
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.profile
# .zshrc
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.zshrc
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.zshrc
```

### 3.2 pyenv init

將 `pyenv init` 添加到您的 Shell 中以啟用填充和自動補全功能。請確保將 `eval "$(pyenv init-)"` 放在 Shell 配置文件的末尾，因為它在初始化期間會操縱 PATH。

```
# .profile
echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n  eval "$(pyenv init -)"\nfi' >> ~/.profile
# .zshrc
echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n  eval "$(pyenv init -)"\nfi' >> ~/.zshrc
```

## 4 重啟 SHELL

先重新加載環境變量文件後，重啟 Shell。

```
source ~/.profile  
source ~/.zshrc
```

接下來你輸入 `pyenv version`，應該就會顯示你目前 `pyenv` 使用的 Python 版本，輸入 `pyenv versions` 能顯示你 `pyenv` 中安裝的所有 Python 版本。

## 5 安裝 Python 版本

我目前要使用 Python 3.7.6 的版本。

```
pyenv install 3.7.6
```

## 6 設置 Shell 默認全局 Python 版本

```
pyenv global 3.7.6
```

接下來重新登出登入電腦。使用 `python --version` 查看 Python 版本就會發現版本已經變成 3.7.6 了。

## 7 Reference

- [pyenv / pyenv - Github](#)
- [Git 設置與取消代理](#)