

---

# **Python Project Management**

***Release 1.0***

**Peng Xiao**

**Nov 11, 2021**



<b>1</b>	<b>1. Python</b>	<b>1</b>
1.1	pyenv Mac . . . . .	1
1.2	pyenv Linux . . . . .	3
1.3	pyenv Windows . . . . .	4
1.4	pyenv . . . . .	4
1.5	dockerPython . . . . .	5
<b>2</b>	<b>2. Python</b>	<b>7</b>
2.1	pip . . . . .	7
2.2	virtualenv . . . . .	8
2.3	requirements.txt . . . . .	9
<b>3</b>	<b>Indices and tables</b>	<b>11</b>



## 1. PYTHON

**Warning:** LinuxMacPython

### 1.1 pyenv Mac

pyenv

<https://github.com/pyenv/pyenv>

#### 1.1.1

1. xcode `sudo xcode-select --install`
2. homebrew <https://brew.sh/>
3. git <https://git-scm.com/>
4. python build environment `brew install openssl readline sqlite3 xz zlib`

#### 1.1.2

1. homebrew

```
brew update
brew install pyenv
```

2. homebrew

```
curl https://pyenv.run | bash
```

bashzsh

```
export PATH="$HOME/.pyenv/bin:$PATH"

echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.zprofile
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.zprofile
echo 'eval "$(pyenv init --path)"' >> ~/.zprofile

echo 'eval "$(pyenv init -)"' >> ~/.zshrc
```

pyenv

```
~ pyenv
pyenv 2.2.0
Usage: pyenv <command> [<args>]

Some useful pyenv commands are:
--version    Display the version of pyenv
activate     Activate virtual environment
commands     List all available pyenv commands
deactivate   Deactivate virtual environment
doctor       Verify pyenv installation and development tools to build python(s).
exec         Run an executable with the selected Python version
global       Set or show the global Python version(s)
help         Display help for a command
hooks        List hook scripts for a given pyenv command
init         Configure the shell environment for pyenv
install      Install a Python version using python-build
local        Set or show the local application-specific Python version(s)
prefix       Display prefix for a Python version
rehash       Rehash pyenv shims (run this after installing executables)
root         Display the root directory where versions and shims are kept
shell        Set or show the shell-specific Python version
shims        List existing pyenv shims
uninstall    Uninstall a specific Python version
version      Show the current Python version(s) and its origin
version-file  Detect the file that sets the current pyenv version
version-name  Show the current Python version
version-origin Explain how the current Python version is set
versions     List all Python versions available to pyenv
virtualenv    Create a Python virtualenv using the pyenv-virtualenv plugin
virtualenv-delete Uninstall a specific Python virtualenv
virtualenv-init  Configure the shell environment for pyenv-virtualenv
virtualenv-prefix Display real_prefix for a Python virtualenv version
virtualenvs    List all Python virtualenvs found in `SPYENV_ROOT/versions/*'.
whence        List all Python versions that contain the given executable
which         Display the full path to an executable
```

See ``pyenv help <command>'` for information on a specific command.  
For full documentation, see: <https://github.com/pyenv/pyenv#readme>

~

### 1.1.3

python

```
~ pyenv install 3.9.7
python-build: use openssl@1.1 from homebrew
python-build: use readline from homebrew
Downloading Python-3.9.7.tar.xz...
-> https://www.python.org/ftp/python/3.9.7/Python-3.9.7.tar.xz
Installing Python-3.9.7...
```

(continues on next page)

(continued from previous page)

```
python-build: use tcl-tk from homebrew
python-build: use readline from homebrew
python-build: use zlib from xcode sdk

BUILD FAILED (OS X 12.0.1 using python-build 20180424)

Inspect or clean up the working tree at /var/folders/d1/5cgd23cn2c5fd_hr4vk6pgfh0000gn/T/
↳ python-build.20211029121146.25284
Results logged to /var/folders/d1/5cgd23cn2c5fd_hr4vk6pgfh0000gn/T/python-build.
↳ 20211029121146.25284.log

Last 10 log lines:
checking for python3... python3
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
checking MACHDEP... "darwin"
checking for gcc... clang
checking whether the C compiler works... no
configure: error: in `/var/folders/d1/5cgd23cn2c5fd_hr4vk6pgfh0000gn/T/python-build.
↳ 20211029121146.25284/Python-3.9.7':
configure: error: C compiler cannot create executables
See `config.log' for more details
make: *** No targets specified and no makefile found. Stop.
~ softwareupdate --all --install --force
Software Update Tool

Finding available software
No updates are available.
~ sudo rm -rf /Library/Developer/CommandLineTools
Password:
~ sudo xcode-select --install
```

```
globalpython
```

```
pyenv rehash
```

## 1.2 pyenv Linux

```
pyenv
```

<https://github.com/pyenv/pyenv>

---

**Note:** Ubuntu 20.10Linuxgithub

---

## 1.2.1

```
sudo apt-get update; sudo apt-get install make build-essential libssl-dev zlib1g-dev \
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm git vim \
libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev
```

## 1.2.2

```
curl https://pyenv.run | bash
```

## 1.2.3

zsh

```
echo 'export PYENV_ROOT="$HOME/.pyenv" >> ~/.zprofile
echo 'export PATH="$PYENV_ROOT/bin:$PATH" >> ~/.zprofile
echo 'eval "$(pyenv init --path)" >> ~/.zprofile

echo 'export PYENV_ROOT="$HOME/.pyenv" >> ~/.profile
echo 'export PATH="$PYENV_ROOT/bin:$PATH" >> ~/.profile
echo 'eval "$(pyenv init --path)" >> ~/.profile

echo 'eval "$(pyenv init -)" >> ~/.zshrc
```

<https://github.com/pyenv/pyenv>

## 1.3 pyenv Windows

<https://github.com/pyenv-win/pyenv-win#installation>

pip

Powershell or Git Bash:

```
pip install pyenv-win --target $HOME\\.pyenv
```

github

## 1.4 pyenv

### 1.4.1 python

```
pyenv install --list python
```

```
pyenv install 3.9.4 python
```



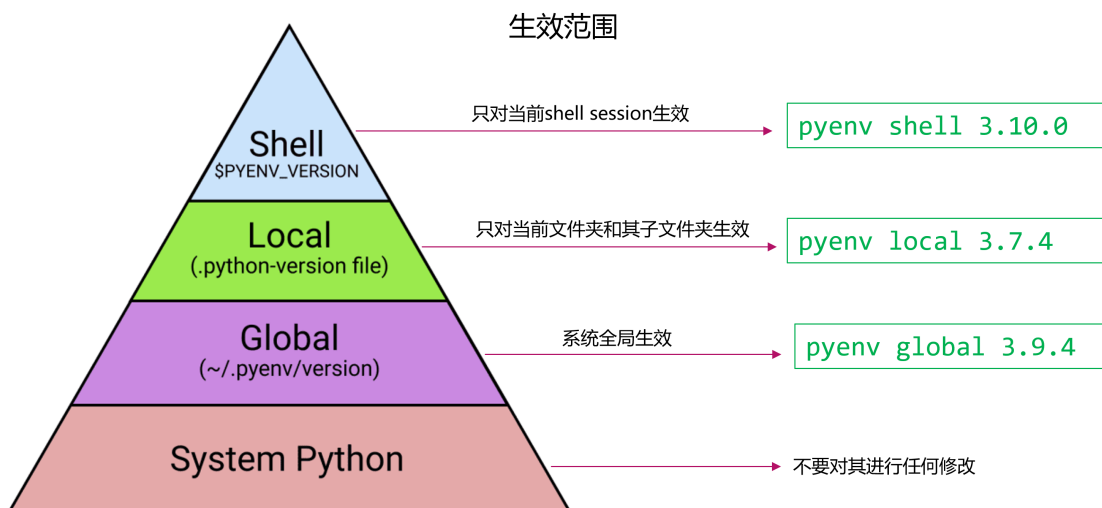
## 1.4.2 python

```
pyenv versions python
```

```
pyenv global 3.9.6 python
```

```
pyenv versions
3.7.8
3.8.10
* 3.9.6 (set by C:\Users\Peng Xiao\.pyenv\pyenv-win\version)
```

## 1.4.3 python



## 1.5 dockerPython



## 2. PYTHON

### 2.1 pip

#### 2.1.1 pip

pip

```
$ python -m pip --version
pip 21.1.3 from C:\Users\Peng Xiao\.pyenv\pyenv-win\versions\3.9.6\lib\site-packages\pip_
↪(python 3.9)
```

#### 2.1.2 pip

```
$ python -m pip install --upgrade pip setuptools wheel
```

#### 2.1.3 pip

<https://packaging.python.org/tutorials/installing-packages/>

#### 2.1.4 pip

Djangoproject A project BPython3.9project A Django 3 project BDjango2

Project	Python	Django
A	3.9	3
B	3.9	2

## 2.2 virtualenv

<https://docs.python.org/3/library/venv.html>

pip

```
manage-python-project git:(master) pyenv versions
system
3.10.0
3.8.7
* 3.9.7 (set by /home/penxiao/.pyenv/version)
manage-python-project git:(master) pip list
Package      Version
-----
pip           21.2.3
setuptools    57.4.0
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the '/home/penxiao/.pyenv/versions/3.9.7/bin/python3.9.
↪ -m pip install --upgrade pip' command.
```

### 2.2.1

```
manage-python-project git:(master) python -m venv venv
```

venvpython

### 2.2.2

```
manage-python-project git:(master) source venv/bin/activate
(venv) manage-python-project git:(master) pip list
Package      Version
-----
pip           21.2.3
setuptools    57.4.0
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the '/home/penxiao/tmp/manage-python-project/venv/bin/
↪ python -m pip install --upgrade pip' command.
(venv) manage-python-project git:(master)
```

```
pip python -m pip install --upgrade pip
```

windows

```
~ python --version
Python 3.8.7
~ cd .\python-demo\
python-demo git:(master) pwd

Path
----
C:\Users\Peng Xiao\python-demo
```

(continues on next page)

(continued from previous page)

```
python-demo git:(master) python -m venv venv
python-demo git:(master) .\venv\Scripts\activate
(venv) python-demo git:(master)
(venv) python-demo git:(master)
(venv) python-demo git:(master) pip list
Package      Version
-----
pip           20.2.3
setuptools    49.2.1
WARNING: You are using pip version 20.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'c:\users\peng xiao\python-demo\venv\scripts\
python.exe -m pip install --upgrade pip' command.
(venv) python-demo git:(master)
```

### 2.2.3

windowsMacLinux deactivate

## 2.3 requirements.txt

### 2.3.1 requirements.txt

```
# requirements.txt
Django
Django==3.1.2
Django>=2.2
Django<3.0
Django>=2.2,<3.0
Django~3.1.2 # Django>=3.1.2, ==3.1.*
Django~3.1   # Django>=3.1, ==3.*
```

```
# requirements.txt
git+https://gitprovider.com/user/project.git@{version}
```

versiontag, a branch, or a commitcommit

### 2.3.2 production and dev

productiondevrequirements

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`