

这是Linux/WSL2/macOS安装环境的一些步骤（基于pyenv和zsh）

1. 安装zsh

Linux/WSL2下安装zsh

```
sudo apt update
sudo apt-get install zsh
sudo usermod -s /usr/bin/zsh $(whoami)
sudo reboot
```

重启后会出现一下界面

```
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~). This function can help you with a few settings that should
make your use of the shell easier.

You can:

(q) Quit and do nothing. The function will be run again next time.
(0) Exit, creating the file ~/.zshrc containing just a comment.
    That will prevent this function being run again.
(1) Continue to the main menu.
(2) Populate your ~/.zshrc with the configuration recommended
    by the system administrator and exit (you will need to edit
    the file by hand, if so desired).

--- Type one of the keys in parentheses ---
```

请选择2

macOS下安装zsh

在macOS 10.13后都已经内置了zsh

1. Linux/WSL2 通过pyenv安装Python

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv

echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.zprofile

echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.zprofile

echo 'eval "$(pyenv init --path)"' >> ~/.zprofile

exec $SHELL

git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyenv
root)/plugins/pyenv-virtualenv

echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.zprofile

exec $SHELL

sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev
libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev xz-utils tk-dev
libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev

CONFIGURE_OPTS=--enable-shared pyenv install 3.8.8

pyenv virtualenv 3.8.8 pydev
```

2. macOS 通过pyenv安装Python

```
brew update
brew install pyenv

echo 'eval "$(pyenv init --path)"' >> ~/.zprofile

CONFIGURE_OPTS=--enable-shared pyenv install 3.8.8

pyenv virtualenv 3.8.8 pydev
```

关于macOS下brew的安装请参考(<https://brew.sh/>)

安装成功验证如下

```
% python3
Python 3.8.8 (default, Aug  3 2021, 11:44:45)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
```