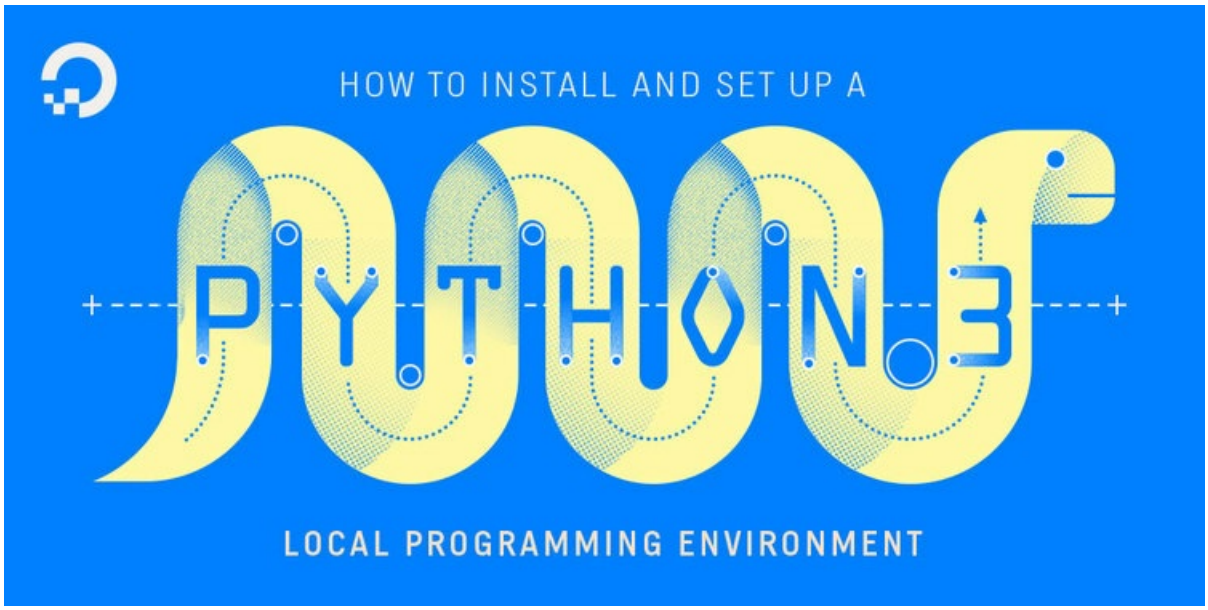

安装和配置

安装 python

安装 python 3

- 检查系统自带的版本:

```
import sys
print(sys.version)
!{sys.executable} --version
```

```
3.6.5 |Anaconda, Inc.| (default, Apr 29 2018, 16:14:56)
[GCC 7.2.0]
Python 3.6.5 :: Anaconda, Inc.
```

```
%%bash
python2 --version
python3 --version
```

```
Python 2.7.12
Python 3.6.5 :: Anaconda, Inc.
```

- 以 Ubuntu 16.04 LTS 为例, 确保系统中已安装:

```
sudo apt-get install python3 python3-dev
sudo apt-get install python3-tk
```

pip

安装 pip

检查 pip 是否安装以及版本

```
%%bash
pip --version
```

```
pip 10.0.1 from /home/lydia/miniconda3/lib/python3.6/site-packages/pip (python 3.6)
```

```
%%bash
python -m pip --version
```

```
pip 10.0.1 from /home/lydia/miniconda3/lib/python3.6/site-packages/pip (python 3.6)
```

安装:

```
sudo apt-get install python3-pip
```

用 pip 安装 python 包

```
pip install <package>
pip install --upgrade <package>
pip uninstall <package>
pip list
pip list --outdated
pip show <package>
```

安装到根目录(不推荐!)

```
sudo -H pip install <package>
```

[选项]

-U, --upgrade	升级, 配合 install 使用
--force-reinstall	强制重新安装依赖
-I, --ignore-installed	强制安装(无论是否已安装)
--no-cache-dir	不生成cache
-i *url*	使用指定源(更改配置见 pip 镜像源)
package==x.x.x	指定版本, 错误版本号(或为空)将返回可用版本号
"*package*<x.x.x"	指定小于某版本的最新版本, 必须有引号

检查包(尝试导入, 查看包版本):

```
python -c "import <package>"
python -c "import <package>; print(<package>.__version__)"   # 版本号
python -c "import <package>; print(<package>.version)"   # 安装位置
```

使 pip 安装 python 包到用户路径

```
%%bash
python -m site | grep 'USER_SITE'
```

```
USER_SITE: '/home/lydia/.local/lib/python3.6/site-packages' (doesn't exist)
ENABLE_USER_SITE: True
```

- 确认已创建用户包安装路径.
- 在 `~/.bashrc` 中设置 `PATH` 变量:

```
export PATH=$HOME/.local/bin:$PATH
```

- 使用 `install` 或 `list` 时加上选项 --user (注意: `uninstall` 不用)

> *使用 Anaconda 路径下的 pip 时不需要这个选项*

```
pip install --user <pkg_name>
pip list --user
pip list --outdated --user
```

Anaconda

安装 Anaconda或 Miniconda(推荐)

- 安装 miniconda 64位 (默认安装到 ~/miniconda3/)

  下载 https://conda.io/miniconda.html

```
%%bash
bash Miniconda3-latest-Linux-x86_64.sh
```

- 在 `~/.bashrc` 加入如下语句:

```
export PATH=$HOME/miniconda3/bin:$PATH
. $HOME/miniconda3/etc/profile.d/conda.sh
```

完成后执行:

```
source ~/.bashrc
```

- 安装好后先更新 conda 和 pip:

```
conda update conda pip
```

- 帮助

```
conda [cmd] --help
```

miniconda 下载 https://conda.io/miniconda.html
安装参考 https://conda.io/docs/user-guide/install/index.html

environment

默认环境为 base
`-n, --name <env_name>` 指定环境

```
conda create --name <env_name> [<packages>] python=3.6
conda create --name <env_name> --clone base

# e.g.
conda create --name intel intelpython3_core python=3
```

```
# 删除环境
conda-env remove --name <env_name>
*
# 列出信息
conda info -e/--envs
conda-env list

# 进入环境
source activate <env_name>
conda activate <env_name>

# 离开环境
source deactivate
conda deactivate

# 在某环境中执行操作
conda <commands> --name <env_name>
```

使用 conda 命令

```
conda info [<packages>]
conda list [<packages>]
conda install <packages>
conda update/upgrade <packages>
conda remove/uninstall <packages>
conda clean ...
```

见 `conda --help` 或 `conda cmd --help`

配置:

```
# 安装确认时显示源的 url, 默认已开启
conda config --set show_channel_urls True

# 禁止每次更新自动检查 conda 的更新, 默认True
conda config --set auto_update_conda False

# 禁止源优先而使用版本优先, 默认True
conda config --set channel_priority False

# 增加设置, 例如增加源(可使用 url 或别名)
conda config --add channels <channel>

# 删除某项设置的所有值, 例如删除所有自定义源
conda config --remove-key channels

# 删除某项设置的某个值, 例如删除某个源
conda config --remove channels <channel>
```

用户配置文件 `~/.condarc` (见 conda 镜像源)
参考链接 https://conda.io/docs/user-guide/configuration/use-condarc.html

也可以在不同的环境用不同的配置文件: `~/miniconda3/envs/<env_name>/.condarc`

镜像源

pip 镜像源

用户配置文件:

```
cat ~/.pip/pip.conf
```

```
[global]
index-url = https://pypi.douban.com/simple
trusted-host = pypi.doubanio.com
```

conda 镜像源

```
conda config --add/--prepend channels <new_channel>
conda config --append channels <new_channel>
```

清华源:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/

channels 的默认的优先级为: 前排优先
镜像源中 `pkgs/main` 通常比 `pkgs/free` 新, 因此将其放在最前.
或使用 `channel_priority: false`, 见上一节 conda 设置.

e.g.

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
```

此时 `~/.condarc` 的内容:

> channels:
> - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
> - defaults

Troubleshooting

Ubuntu 14.04 LTS 安装 python3 以及 pip

```
# 添加 ppa
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update

# 安装 python3.5:
sudo apt-get install python3.5 python3.5-dev python3.5-tk

# 安装 python3.6:
sudo apt-get install python3.6 python3.6-dev python3.6-tk

# python3.6 无法安装 pip 的解决方法:
#下载 https://bootstrap.pypa.io/get-pip.py
sudo mv -v /usr/lib/python3/dist-packages/pkg_resources.py /usr/lib/python3/dist-packages/pkg_resources.py.bak
sudo -H python3.6 get-pip.py
```

conda 安装时指定源且禁用其他源

```
conda install <packages> -c <channel> --override-channels
```

astropy 新版不再支持旧版 python

```
pip install "astropy<3.0" # for python < 3.5
```

安装常用包

numpy, scipy, matplotlib

```
conda install numpy
conda install scipy
conda install matplotlib
```

Or

```
pip install numpy
pip install scipy
pip install matplotlib
```

配置启动文件(Ubuntu), 在 `~/.bashrc` 中添加:

```
export PYTHONSTARTUP=$HOME/.pythonrc
```

```
cat ~/.pythonrc
```

```
# Python startup file
# ~/.pythonrc
import matplotlib.pyplot as plt
from matplotlib import cm
import numpy as np
import os

print(">>> import matplotlib.pyplot as plt")
print(">>> from matplotlib import cm")
print(">>> import numpy as np")
print(">>> import os")

try:
    from pprint import pprint
except ImportError:
    pass
else:
    print(">>> from pprint import pprint")

# Tab completion & history
try:
    import atexit, readline, rlcompleter
except ImportError:
    pass
else:
    readline.parse_and_bind('tab: complete') # Tab completion
    readline.parse_and_bind('''control-l:"    "''') # Indent, since TAB is occupied
    histfile = os.path.join(os.environ['HOME'], '.pyhistory') # History file
    try:
        readline.read_history_file(histfile)
    except IOError:
        pass
    atexit.register(readline.write_history_file, histfile)
    del histfile, readline, rlcompleter
    print("--- Tab completion: [ON] ---")
    print("--- History: [ON] ---")
    print("--- [Tab] -> [Ctrl+L] ---")
```

Astropy, SunPy

```
conda install astropy
conda install sunpy -c conda-forge
```

Or

```
pip install astropy
pip install sunpy[all] pytest
```

测试:

```
import astropy
astropy.test()
```

```
import sunpy
sunpy.self_test(online=False)
```

如果出错: '$USER'权限问题, 见下面修改 `~/.sunpy/sunpyrc` 说明.

查看包信息:

```
%%bash
conda search "sunpy 0.9.0"
```

```
Loading channels: ...working... done
# Name                  Version           Build  Channel
sunpy                     0.9.0          py27_0  conda-forge
sunpy                     0.9.0          py35_0  conda-forge
sunpy                     0.9.0          py36_0  conda-forge
```

```
%%bash
conda info "sunpy 0.9.0 py36_0"
```

```
sunpy 0.9.0 py36_0
------------------
file name   : sunpy-0.9.0-py36_0.tar.bz2
name        : sunpy
version     : 0.9.0
build string: py36_0
build number: 0
channel     : https://conda.anaconda.org/conda-forge/linux-64
size        : 5.7 MB
arch        : x86_64
constrains  : ()
license     : BSD 2-Clause
md5         : 3008791f706eadca029533ae3dd0a336
platform    : linux
subdir      : linux-64
url         : https://conda.anaconda.org/conda-forge/linux-64/sunpy-0.9.0-py36_0.tar.bz2
dependencies:
    astropy >=2.0.3
    beautifulsoup4
    drms
    glymur
    hypothesis
    matplotlib >=1.3
    mock
    numpy >=1.11
    pandas >=0.12.1
    pytest
    pytest-astropy
    pytest-mock
    python 3.6*
    requests
    scikit-image
    scipy
    sqlalchemy
    suds-jurko
```

用户文件位置: `~/.sunpy/sunpyrc`

```
cat ~/.sunpy/sunpyrc
```

```
[general]
working_dir = /home/$USER/software/sunpy-downloads

[downloads]
download_dir = data
sample_dir = data/sample_data

[database]
url = sqlite:////home/$USER/software/sunpy-downloads/sunpydb.sqlite
```

```
import sunpy
sunpy.print_config()
```

```
FILES USED:
  /home/lydia/miniconda3/lib/python3.6/site-packages/sunpy/data/sunpyrc
  /home/lydia/.sunpy/sunpyrc

CONFIGURATION:
  [general]
  time_format = %Y-%m-%d %H:%M:%S
  working_dir = /home/lydia/software/sunpy-downloads

  [downloads]
  download_dir = /home/lydia/software/sunpy-downloads/data
  sample_dir = /home/lydia/software/sunpy-downloads/data/sample_data

  [database]
  url = sqlite:////home/lydia/software/sunpy-downloads/sunpydb.sqlite
```

设置下载位置:
(参考 http://docs.sunpy.org/en/stable/guide/customization.html?highlight=sunpy.config.set**)**

e.g.

> [downloads]
> download_dir = ...

```
>>> import sunpy
>>> sunpy.config.set('downloads', 'download_dir',
                     '/home/<user>/<your_path>/sunpy-downloads/data')
```

退出后再进入 python 如果发现没有修改成功, 则需要手动修改 ~/.sunpy/sunpyrc 文件.
不能用 '$USER' 代替具体的用户名 '<user>'

Jupyter

```
conda install jupyter
```

Or

```
pip install jupyter
```

```
%%bash
jupyter --path
```

```
config:
    /home/lydia/.jupyter
    /home/lydia/miniconda3/etc/jupyter
    /usr/local/etc/jupyter
    /etc/jupyter
data:
    /home/lydia/.local/share/jupyter
    /home/lydia/miniconda3/share/jupyter
    /usr/local/share/jupyter
    /usr/share/jupyter
runtime:
    /run/user/1000/jupyter
```

- 设置双击打开 *.ipynb 文件

```
pip install nbopen
```

再执行:

Linux/BSD:

```
python3 -m nbopen.install_xdg<br>
```

Windows:

```
python3 -m nbopen.install_win<br>
```

Mac:
Clone the repository (https://github.com/takluyver/nbopen.git)
and run `./osx-install.sh`

之后即可在文件浏览器中选择文件的打开方式为 `Jupyter Notebook`.

> 注意可能会更改 bash 脚本的默认打开方式.

- Jupyter 插件集合
(https://github.com/ipython-contrib/jupyter_contrib_nbextensions)

```
conda install jupyter_contrib_nbextensions -c conda-forge
```

Or

```
pip install jupyter_contrib_nbextensions
```

> This also automatically installs the Javascript and CSS files:
> `jupyter contrib nbextension install --sys-prefix`

安装好后将同时启用 `Nbextensions` 选项卡, 手动选择需要的插件, 或者在命令行启用和禁用:

```
jupyter nbextension list # 查看
jupyter nbextension enable <path> # <path> 为上述 list 中的 <extension>/main
jupyter nbextension disable <path>
```

注意用如果 jupyter 安装在 conda 路径, 手动安装插件(`jupyter nbextension install`)时需要指定 --sys-prefix

- notebook主题 (https://github.com/dunovank/jupyter-themes)

```
pip install jupyterthemes
```

- 演示代码过程的插件 (https://github.com/lgpage/nbtutor)

```
conda install nbtutor -c conda-forge
```

Or

```
pip install nbtutor
jupyter nbextension install --overwrite --py nbtutor
jupyter nbextension enable --py nbtutor
```

载入:

```
# ipython/jupyter
%load_ext nbtutor
```

使用:
CodeCell 中首行加入下面语句, 然后执行 Cell (numpy 等需要在 Cell 内导入)

```
# ipython/jupyter
%%nbtutor -r -f
# 或
%%nbtutor -r -f -i  # 缩减显示
```

其他

- PeakUtils (http://peakutils.readthedocs.io/en/latest/)

```
git clone https://bitbucket.org/lucashnegri/peakutils.git
cd peakutils
python setup.py install
```

```
>>> import peakutils
```

- LMfit-py (https://github.com/lmfit/lmfit-py)

```
conda install lmfit -c conda-forge
# 或
pip install lmfit
```

```
>>> import lmfit
```

- HDF5包 (http://docs.h5py.org/en/latest/index.html)

pandas

```
conda install pytables pandas
```

Or

```
pip install tables pandas
```

h5py (http://docs.h5py.org/en/latest/index.html)

```
conda install h5py  # 推荐. 将同时安装 hdf5, 并得到 h5dump 等命令
```

Or

```
pip install h5py  # 需要 sudo apt-get install libhdf5-dev
```

- CDF包

安装 spacepy (https://pythonhosted.org/SpacePy)

或

```
pip install cdflib # https://github.com/MAVENSDC/cdflib
```

- EVTK (https://bitbucket.org/pauloh/pyevtk)

```
sudo apt-get install mercurial  # 得到 hg 命令

hg clone https://bitbucket.org/pauloh/pyevtk
cd pyevtk
python setup.py build --debug install
```

```
hg pull && hg update default # 更新
```

```
>>> from evtk.hl import gridToVTK
```

- Mayavi (http://docs.enthought.com/mayavi/mayavi)

需要先安装 cython (conda 或 pip)

```
pip install mayavi  # 将同时安装 vtk. (pip 是目前 mayavi 最保险的安装方式)
```

```
# shell
mayavi2
```