
Machine*LearningMcMasterUnivDocumentation*

Release 1.2

Fanwang Meng @ McMaster University

Mar 25, 2019

Contents:

1	Command Terminal	3
2	Git-hub	5
3	Text Editors for Python	7
4	Python, Its Tools and Libraries	9
4.1	Installization	9
4.2	Version Control of Python	10
4.3	Contragulations	11

Machine learning has been one of the most exciting field in modern artifical intelligence, which has chagend the way prople live and how scientist do their studies. Here we set up a setp-by-step tutorial to guide people how can they build theri python programming environment for machine learning.

Command Terminal

Set up a Terminal for Windows 10

1. Install Ubuntu

Follow the instruction of [this tutorial](#) and [another Chinese version tutorial](#). Now please restart your laptop.

2. Install Cmdr

Cmdr is a more user friendly command line terminal than cmd, which gave me a shock when I used it for the first time. Unzip the files to *C:\Program Files\Cmdr*.

Add the path where the *cmd.exe* lies in to the system environment, which enables you to use *Cmdr* with the *Win+r* shortcut followed by *cmdr*. You may follow the [this example](#). Open the cmd terminal as administrator, type in the following the command:

```
cmdr.exe /REGISTER ALL
```

Now you can start your cmdr with *Win+Alt+P*. Find the setting option at the right bottom of cmdr terminal and click the *Startup*, choose *Command line* and set it as `bash -cur_console:p`. Enter the bash shell with the following way: click the green icon at the right bottom, choose **3:{bash}** and **4:{bash}**, open the setting window, **Startup/tasks**, set it as `{bash::bash}` and set task parameters as `/icon "%CMDER_ROOT%\icons\cmdr.ico"`, confirm and exit. Now you can use it. Enjoy!

CHAPTER 2

Git-hub

Download the latest version Git from <https://git-scm.com/downloads> according to your operating system.

For windows 10 users, it's better for you to install them with *Cmder*. #. Add the sudo password which is essential for installing packages. "sudo passwd"

1. Install git.

```
$ sudo apt-get install git
```

2. Configure the git-hub account.

```
$ git config --global user.name "your_user_name"
$ git config --global user.email "your_email_address"
```

If you have a two factor authentication, you need a **Personal access token** to access the private repos.

Note

For Fedora and Centos users, try

```
$ sudo yum install git
where the administrative password is required.
```

If there a error message, try to add the admin password by

```
$ sudo passwd root``
```

Go to <https://education.github.com> for more trainings.

CHAPTER 3

Text Editors for Python

Two text editors are my favourites, **vim** and **sublime text 3**, both of which have been widely used in programming. Vim is very powerful, but a little difficult for learn at the beginning. Sublime is smart and easy to use.

1. Vim

Vim is pre-installed in most linux systems as well as Mac OSX systems. If you want to install vim from cmd command lines. You may follow this [tutorial](#) I have written.

2. Sublime

Download Sublime 3 from <https://www.sublimetext.com/3> and choose the right package accordingly. *Install subl command line tool* (you can skip this setp) Mac OSX:

```
$ sudo ln -s /Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl /usr/  
↪bin/subl
```

Linux: modify it accordingly. If it doesn't work, tyr

```
$ echo "export PATH=~/.bin:$PATH" >> ~/.profile`
```

Install Package Control Click **View > Show Console** to open the ST3 console and then copy this code into it:

```
#coding:utf-8  
  
import urllib.request,os,hashlib; h = 'df21e130d211cfc94d9b0905775a7c0f' +  
↪'1e3d39e33b79698005270310898eea76'; pf = 'Package Control.sublime-package'; ipp_  
↪= sublime.installed_packages_path(); urllib.request.install_opener( urllib.  
↪request.build_opener( urllib.request.ProxyHandler()) ); by = urllib.request.  
↪urlopen( 'http://packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh =_  
↪hashlib.sha256(by).hexdigest(); print('Error validating download (got %s_  
↪instead of %s), please try manual install' % (dh, h)) if dh != h else open(os.  
↪path.join( ipp, pf), 'wb' ).write(by)
```

You can now install packages by using the keyboard shortcut cmd+shift+P. Start typing install until Package Control: Install Package appears. Press enter and search for available packages.

Install SideBarEnhancements Package

Download or clone the contents of this repository to a folder named exactly as the package name into the Packages/ folder of ST. Otherwise, refer <https://packagecontrol.io/packages/SideBarEnhancements>.

Install Anaconda

Install Anaconda package with the help of Package Control. For a full description of its features, you may consult <https://packagecontrol.io/packages/Anaconda>. Don't forget to add {"anaconda_linting": false} in Anaconda User setting file.

Install Python-PEP8 Autoformat

Install python-pep8, which is a powerful tool for PEP8 formatting of python codes. It's easy to install it with Package Control.

Install sublimerge

This is a powerful tool for a lot of interesting things. It's the same way of installation. And the website is <https://packagecontrol.io/packages/Sublimerge>.

Install Sublime-ShellScriptImproved

This tool enables better ShellScript (Bash) syntax highlight for Sublime Text >= 3116, which makes your shell script more attractive: <https://github.com/jfcherng/Sublime-ShellScriptImproved>. It can be installed by the *packagecontrol*.

Install LaTeXTools

This tool enables the powerful preview of latex equations. You must enjoy it very much! More problems can refer to GitHub repo: <https://github.com/SublimeText/LaTeXTools>.

Other features You can make full use of this great article <https://realpython.com/blog/python/setting-up-sublime-text-3-for-full-stack-python-development/>.

It's very important to trim the trailing white space. You may configure your user setting file of sublime like:

```
#coding:utf-8

{
    "auto_indent": true,
    "font_size": 14,
    "ignored_packages":
    [
        "Vintage"
    ],
    "rulers":
    [
        79
    ],
    "smart_indent": true,
    "tab_size": 4,
    "trim_automatic_white_space": true,
    "trim_trailing_white_space_on_save": true,
    "use_tab_stops": true,
    "word_wrap": true,
    "wrap_width": 80
}
```

Once all the features have been configured, your vim should look like:

The Sublime should be equipped with powerful features like this:

Python, Its Tools and Libraries

In this part, we will first install python 3.6 and some libraries with the help of pip. Then we will show how to work with one of the best tools for python learning and teaching, ipython. However, if you don't like the command line way of programming, the easiest way is to install Anaconda, <https://www.continuum.io/downloads/>.

4.1 Installization

1. Install Python

Although Python 2 has been widely used, we recommend to use Python 3 instead. If you have some projects written in python 2, there are some optional ways for version control of python.

Go to <https://www.python.org/downloads/>, and download the latest version of python.

2. Install Pip

It is recommended to install pip 3 in consistence with python 3.6. A detailed instruction can be found: <https://pip.pypa.io/en/stable/installing/>. One convenient way install it is with this script: <https://bootstrap.pypa.io/get-pip.py>.

```
$ python get-pip.py
```

We can use pip to install, upgrade, uninstall python packages. Here are some examples.

```
$ pip search numpy
$ pip install numpy
#update python libraries
$ pip install numpy -U
#another way to update python libraries
$ pip install numpy --upgrade
```

3. Install machine learning python libraries

The often used python libraries are *numpy*, *pandas*, *matplotlib*, *pybrain*, *statsmodels*, and *scikit-learn*. A good online tutorial is worth reading: <http://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/>. Then we can make use of *pip* to

install python libraries. The official website of scikit-learn is a gold ore: <http://scikit-learn.org/stable/index.html>. Install scikit by pip. A good Scipy lecture notes can be found here: <http://www.scipy-lectures.org/>.

4. **Install IPython** IPython provides a rich architecture for interactive computing with: a powerful interactive shell and a kernel for Jupyter. It's quite easy to install IPython by *pip*:

```
$ pip3 install ipython
```

5. **Install Jupyter** Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science. But if Anaconda is not your choice, you can install Jupyter by *pip*.

```
$ pip3 install --upgrade pip
$ pip3 install jupyter
```

Once you have *Jupyter* installed, you can run notebook:

```
$ jupyter-notebook-3.6 test.ipynb
#you can also do it in this way
$ jupyter-3.6 notebook test.ipynb
```

An interesting book might help: <https://ipython.org/books.html>.

4.2 Version Control of Python

Sometimes, you have to work with both python 2 and python 3 and version control will become of a big concern. Otherwise, you will come across some unexpected errors.

For Mac Users

For now, maybe *MacPorts* is a good choice for people working with different python versions. The prerequisites to install *MacPorts* is that you have *Xcode* installed. Don't forget to open *Xcode* to accept the licence. Then go to the official website of MacPorts: <https://www.macports.org/install.php> and now you can install it by yourself. Now you can use *Macports* to do the version control.

```
$ port search python3*
$ sudo port install python36 @3.6.0
$ port search python2*
$ sudo port install python27 @2.7.13
$ port installed | grep python
$ sudo port select --set python python36 @3.6.0
```

For Linux Users

This is a more generalized way for python version control, where we will use *pyenv*. Here is how to install *pyenv*:

```
#install
$ curl -L https://raw.githubusercontent.com/pyenv/pyenv-installer/master/bin/pyenv-
  ↪installer | bash
#update
$ pyenv update
#check if the following have has been added in ~/.bashrc
#otherwise, add them
$ export PATH=~/.pyenv/bin:$PATH
$ eval "$(pyenv init -)"
$ eval "$(pyenv virtualenv-init -)"
```

Once you have pyenv installed, you can use it to install any specific version of python.

```
#this part is based on http://akbaribrahim.com/managing-multiple-python-versions-with-  
→pyenv/  
#search step need to be added  
$ pyenv install 3.6.0  
$ pyenv install 2.7.13  
#check th current version  
$ pyenv version  
#set specific version globally  
$ pyenv global 3.6.0  
#if you want to set a python version for the current shell session  
$ pyenv shell 3.6.0
```

For more information, you may refer to the *pyenv* <https://github.com/pyenv/pyenv#installation>.

4.3 Contragulations

You are all set for python!!!