

# Scrapy Tutorial #2: How To Install Scrapy On Mac

Last updated on Feb 25 2019 by Michael Yin

## Introduction:

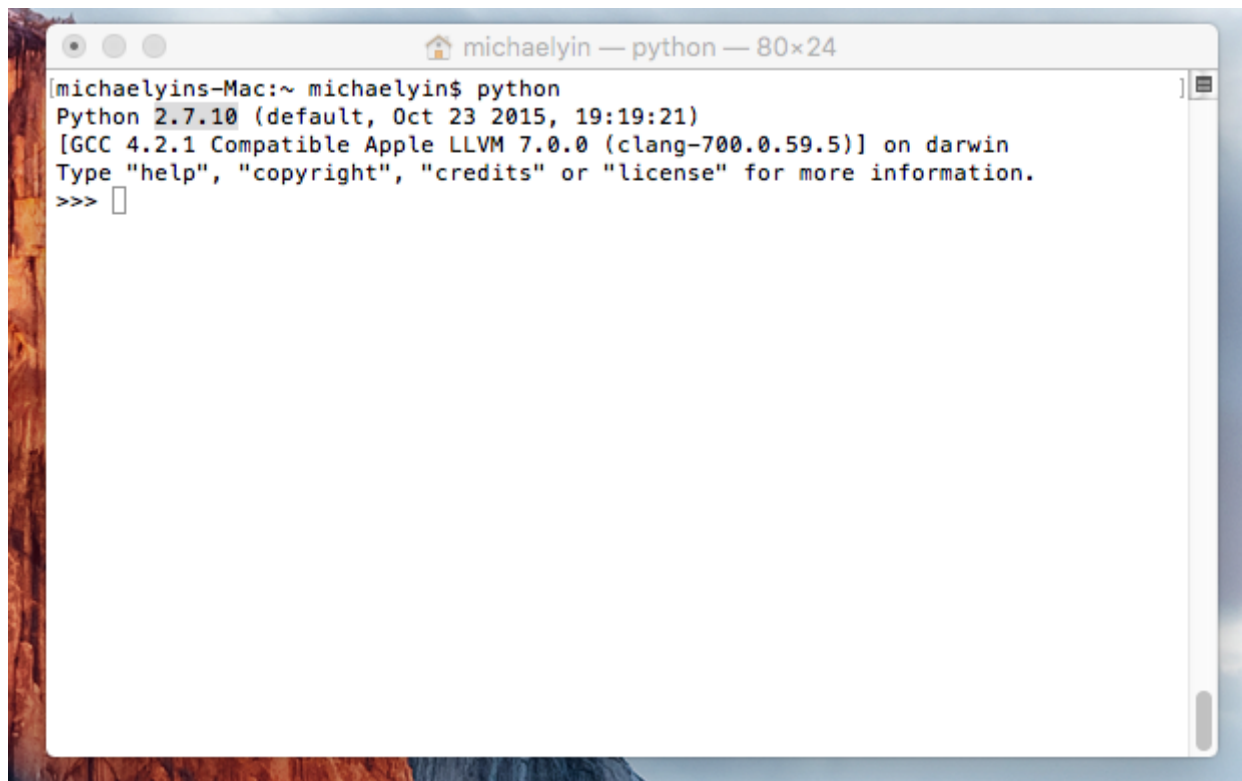
This is the #2 post of my [Scrapy Tutorial Series](#), in this Scrapy tutorial, I will talk about basic points of Python such as Python version, Python package and show you how to install Scrapy on your Mac. **In this tutorial, we would use python3 as our Python version, if you still want to use the old python2 version, just replace all python3 with python2 and pip3 with pip2 in code**

## Basic Points

Even you can successfully install Scrapy on your Mac without reading the basic points here, it is still recommended to read this section carefully because you will have a better understanding of Python, Scrapy, and pip.

### Python Version

The python version of your env we usually talk about is the version number of the Python interpreter. The easy way to check the version number is just type python in your terminal.

A screenshot of a macOS terminal window. The title bar at the top reads 'michaelyin — python — 80x24'. The terminal content shows the command 'python' being executed, resulting in the following output: 'Python 2.7.10 (default, Oct 23 2015, 19:19:21)', '[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.5)] on darwin', and 'Type "help", "copyright", "credits" or "license" for more information.'. The prompt '>>>' is visible at the bottom of the output.

As you can see, the default python interpreter of my Mac is 2.7.10, this version might vary from the different OSX system. Now there are mainly two versions Python 2 and python 3 for you to choose. The difference between them is

Short version: Python 2.x is legacy, Python 3.x is the present and future of the language

If you do not have a solid reason to use python 2, just embrace Python 3, which is the present and future of python.

## pip

pip is the preferred installer program in Python world, for example, we can use pip to install Scrapy by typing `pip install Scrapy`. It will handle all dependency for us and install them first, which is very convenient.

If you want to use pip which included with python2, use pip2, if you want to use pip which included with python3, use pip3, if you want to check the pip command version, just type `pip -V`

## Quick way to install Scrapy on Mac

If you want to get started quick and dirty, just use this way. First, we install homebrew on the Mac, which is the best package manager on Mac. **In this tutorial, we would use python3 as our Python version, if you still want to use the old python2 version, just replace all python3 with python2 and pip3 with pip2**

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master
```

When brew is installed, we can install the latest stable python2 and python3 as you like, you should know you can install them both because two versions can coexist in your Mac now.

```
brew install python3
# install scrapy using pip command
pip3 install scrapy
```

## More decent way to install Scrapy on Mac

Scrapy installed via the code above are global so they are available across all of your projects. That can be convenient at times, but it can also become problems. So how to install Scrapy on an isolated environment? This is why virtualenv created. On my Mac, only a few Python packages such as pip and virtualenv are globally available — other packages such as Scrapy, Django are installed in virtual environments.

After Python has been installed via brew, we install virtualenv

```
pip3 install virtualenv

# Now we can create a virtualenv for us to use
cd ~
mkdir Virtualenvs
cd Virtualenvs
virtualenv scrapy_env
# A virtualenv called scrapy_env now has been created
```

If we need to activate an virtualenv, just use `source $VIRTUALENV` to activate it, if we need to exit the virtualenv, just use deactivate command.

```
michaelyins-Mac:Virtualenvs michaelyin$ source scrapy_env/bin/activate
(scrapy_env) michaelyins-Mac:Virtualenvs michaelyin$
```

As you can see, we use `source scrapy_env/bin/activate` to activate the virtualenv, and now if you install python package, all of them would be in an isolated env, and the name of the virtualenv can be seen in the

shell prompt, which means we are in an virtualenv now. **If you want to know more about virtualenv, just check this [good python guide](#).** Since we are in an virtualenv, we can start to install Scrapy.

```
pip3 install scrapy
```

```
(scrapy_env) michaelyins-Mac:Virtualenvs michaelyin$ python
Python 3.6.2 (default, Jul 17 2017, 16:44:47)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import scrapy
>>> scrapy
<module 'scrapy' from '/Users/michaelyin/Virtualenvs/scrapy_env/lib/python3.6/site-packa
>>>
```

From the code above, you can see the scrapy is now located in virtualenv we just created, you can have many different Scrapy versions as you like now.

## ipython shell

Ipython shell is much more powerful than python console, Scrapy shell will use it if ipython has been installed instead of python console. I will talk about this in more detail in the future, but first, we need to install it.

```
# make sure we are in virtualenv
pip3 install ipython
```

```
(scrapy_env) michaelyins-Mac:Virtualenvs michaelyin$ ipython
Python 3.6.2 (default, Jul 17 2017, 16:44:47)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: import this
The Zen of Python, by Tim Peters
```

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```