总结：

反正两个有很多重合的功能部分，如果不记得就两个挨着试一下

如果官网说的方法太慢，试一下清华源

1. conda的作用：
2. conda可以给我们提供一个独立的环境，相当于python的virtualenv

conda create -name envname python=2.7

activate envname

1. conda info -envs 列出conda创建所有的环境

conda版本：conda --version

更新conda：conda update conda

切换回当前环境：deactivate

删除环境：conda remove --name flowers --all

制作环境副本：conda create --name flowers --clone snowflakes

查看在环境中安装的第三方包：conda list

搜索可安装的包：conda search

安装新软件：conda install --name packagename beautifulsoup4

也可以使用pip安装：pip install pkg

也可以从其它页面下载安装：conda install --channel https://conda.anaconda.org/pandas bottleneck

删除环境的第三方包：conda remove --name envname pck或者 pip uninstall pck

删除conda：rm -rf ~/miniconda OR rm -rf ~/anaconda

1. pip更新
2. python2.7更新pip命令

|  |
| --- |
| python -m pip install --upgrade pip |

1. python3.\*更新pip命令

|  |
| --- |
| python3 -m pip install --upgrade pip |

1. 国内镜像源：

https://pypi.tuna.tsinghua.edu.cn/simple   #清华

http://mirrors.aliyun.com/pypi/simple/       #阿里云

https://pypi.mirrors.ustc.edu.cn/simple/    #中国科技大学

http://pypi.hustunique.com/                   #华中理工大学

http://pypi.sdutlinux.org/                     #山东理工大学

http://pypi.douban.com/simple/            #豆瓣

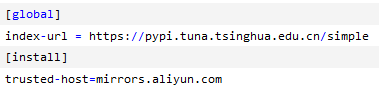
1. 临时修改镜像源

使用pip命令时，可以使用-i参数加上更新源地址，以完成安装和更新。例如（更改为阿里云镜像更新源，不再使用官方更新源）

pip install scrapy -i http://mirrors.aliyun.com/pypi/simple/

1. 1)Linux下修改配置文件的方法

修改~/.pip/pip.conf 该配置文件，如果文件不存在，则需自行创建文件。.表示Linux下的隐藏文件，.必须加上。



2)Windows下修改更新源文件的方法

在users文件下创建pip安装配置文件，文件名为pip.ini。一般路径为：C:\Users\\*\*\pip/pip.ini,其中\*\*表示计算机使用者的名字



Conda和pip通常被认为几乎完全相同。虽然这两个工具的某些功能重叠，但它们设计用于不同的目的。 [Pip](https://pip.pypa.io/en/stable/)是Python Packaging Authority推荐的用于从[Python Package Index](https://pypi.org/)安装包的工具。 Pip安装打包为wheels或源代码分发的Python软件。后者可能要求系统安装兼容的编译器和库。

[Conda](https://conda.io/docs/)是跨平台的包和环境管理器，可以安装和管理来自[Anaconda repository](https://repo.anaconda.com/)以 [Anaconda Cloud](https://anaconda.org/)的conda包。 Conda包是二进制文件，需要使用编译器来安装它们。另外，**conda包不仅限于Python软件。它们还可能包含C或C ++库，R包或任何其他软件。**

1. conda和pip之间的关键区别。

Pip安装Python包，而conda安装包可能包含用任何语言编写的软件的包。在使用pip之前，必须通过系统包管理器或下载并运行安装程序来安装Python解释器。而Conda可以直接安装Python包以及Python解释器。

另一个区别是conda能够创建可以包含不同版本的Python或其他软件包的隔离环境。在使用数据科学工具时，这非常有用，因为不同的工具可能包含冲突的要求，这些要求可能会阻止它们全部安装到单个环境中。 Pip没有内置的环境支持，而是依赖于[virtualenv](https://virtualenv.pypa.io/en/latest/)或[venv](https://docs.python.org/3/library/venv.html) 等其他工具来创建隔离环境。 pipenv，poetry和hatch wrap pip和virtualenv等工具提供了统一的方法来处理这些环境。

Pip和conda在如何实现环境中的依赖关系方面也有所不同。安装包时，pip会在递归的串行循环中安装依赖项。没有努力确保同时满足所有包的依赖性。如果较早安装的软件包与稍后安装的软件包具有不兼容的依赖性版本，则可能导致破坏的环境。conda使用可确保满足环境中安装的所有包的所有要求。此检查可能需要额外的时间，但有助于防止创建破坏的环境，前期关于依赖关系包的元数据是正确的。

考虑到conda和pip之间的相似性，有些人试图将这些工具结合起来创建数据科学环境也就不足为奇了。将pip与conda结合的主要原因是有些包只能通过pip安装。 Anaconda创酷提供超过1,500个软件包，包括最流行的数据科学，机器学习和AI框架。这些，以及包括conda-forge和bioconda在内的数据通过Anaconda云提供的数千个附加软件包，可以使用conda进行安装。尽管有大量的软件包，但与PyPI上提供的150,000多个软件包相比，它仍然很小。有时候需要的包没有conda包，但在PyPI上有，可以用pip安装。

1. Anaconda中的virtualenv和conda的关系

virtualenv / venv是允许用户创建与pip一起使用的隔离的Python环境的实用程序。 conda有自己的内置环境管理器，可以与conda和pip无缝工作，并且事实上比virtualenv / venv有几个优点：

conda环境集成了不同Python版本的管理，包括Python本身的安装和更新。 Virtualenvs必须在现有的，外部管理的Python可执行文件上创建。

conda环境可以跟踪非python依赖;例如无缝管理依赖性和基本工具(如LAPACK或OpenSSL)的并行版本

而不是构建在符号链接上的环境 - 这破坏了virtualenv的隔离，并且对于非Python依赖关系有时可能是脆弱的 - conda-envs是单个可执行路径中的真正隔离环境。

虽然virtualenvs与conda软件包不兼容，但conda环境与pip软件包完全兼容。第一个conda安装pip，然后你可以pip安装任何可用的包在那个环境中。您甚至可以在conda环境文件中显式地列出pip包，这意味着完整的软件堆栈可以从单个环境元数据文件完全重现。

　　也就是说，如果你想在你的virtualenv中使用conda，它是可能的：

　　$ virtualenv test\_conda

　　$ source test conda/bin/activate

　　$ pip install conda

　　$ conda install numpy