

【Python】No.1 Python安装部署和conda管理

NAU Analysts 2020-02-11 00:21

以下文章来源于一个特立独行的猫猫头，作者李清宇



一个特立独行的猫猫头
Keep Learning

开刊致辞

南审CFA编辑小组在发布【PyCFA】系列的文章后，考虑到本专业学生缺乏对计算机知识尤其是Python这门语言的了解，直接按照【PyCFA】进行编程可能存在一定的难度，所以编辑小组从今天起并行更新【Python】系列的教程，将从安装部署、常用语法、数据分析、网络爬虫、机器学习、软件/网站、数据库、区块链、大数据、云计算等方面和大家进行交流讨论。

由于我们编辑小组能力有限，如果推送中出现对某个知识点的错误解读，还请各位同学老师批评指正。如果各位同学有问题欢迎到南审CFA系QQ群进行讨论。本系列仅供南审CFA专业学生学习交流，不涉及任何商业活动和利益。

Python的安装部署&conda管理

1 Python是什么

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的开放源代码脚本语言。

2 为什么要学Python

社会形势

近年来，GDP和M2增速下降，宏观经济陷入困境，不少原先利润丰厚的企业面临着倒闭风险，金融行业形势也不乐观，去杠杆使行业规模减小，科技飞速发展使部分岗位的人才需求减小。然而，前几年金融业的蓬勃发展促使高校扩大了金融相关行业的招生，导致了金融人才市场供不应求的社会情况，使得如今毕业生的竞争加剧。因此，学习Python这样一门硬核技能，可以帮助金融专业的同学们在激烈的社会竞争中脱颖而出。

专业因素

Python的语法精确简洁，每个数学语句都能转变成一行Python代码，每行允许超过十万的计算量，易于实现金融算法和数学计算。此外，Python还拥有大量宝贵的第三方工具，这使其能被用于数据的爬取和分析，模型的构建及机器学习等多个方面，成为处理金融行业错综复杂事务的可靠选择，在校大学生们也可以通过Python对专业所学内容进行实现，从而深入理解专业知识。

第一性原理

计算机科学是一种应用型科学，Python作为其中的一部分，可以通过高度抽象化现实生活中的共性现象来解决问题，通过学习其算法和实际应用，能让你在应对各个行业的问题时都游刃有余。

3 Python的优缺点

优点

- 简单易学 —— Python 是代表简单主义思想的一种语言，简单易懂的语法使Python非常容易上手。
- 规范的代码 —— Python 采用强制缩进的方式使得代码具有极佳的可读性。
- 免费开源 —— Python 是 FLOSS（自由/开放源码软件）之一，可以不断被创新改进。简单地说，你可以自由地发布这个软件的拷贝、阅读其源代码、对它做改动或把它的一部分用于新的自由软件中。
- 可移植性 —— 基于开源本质，Python 可被移植在许多平台上。
- 解释性 —— 用Python 语言写的程序不需要编译成二进制代码，可以直接从源代码运行程序，这使 Python 使用起来更加简单，也帮助了 Python 程序移植到在不同平台上。
- 面向对象 —— Python 支持面向过程的编程，同时也通过一种非常强大又简单的方式（与其他主要的语言如 C++ 和 Java 相比）实现面向对象编程。
- 可扩展性 —— Python语言可以起到桥梁的作用，串联其他计算机语言，即你可以使用C或C++编写关键代码和不愿开放的算法，然后通过Python程序进行调用。
- 丰富的库 —— Python标准库非常庞大，它可以帮助处理各种工作，包括正则表达式、文档生成、数据库（Python提供所有主要商业数据库的接口）、网页浏览器和电子邮件等，除标准库以外，Python还有许多其他高质量的库，如Twisted和Python图像库等。

缺点

- 当前Python的国内市场较小，但随着时间推移，很多国内软件公司，尤其是游戏公司，也开始规模使用Python。
- 中文资料尚且匮乏，虽然一些优秀的教材已经被翻译成中文，但大部分高级内容还是只能看英文版。

4 关于Anaconda

什么是Anaconda

开源的Anaconda是操作Python和R语言最简单的途径，全世界有超过1900万的用户发行版本，是开发、测试及单机培训的行业标准，是数据科学家可以：

- 快速下载7500多个Python/R语言数据科学相关的开源包
- 运用conda管理 库（libraries）、依赖项与环境
- 运用scikit-learn、TensorFlow和Theano开发训练机器学习和深度学习模型
- 可利用性能优秀的数据分析工具，如：Dask、NumPy、pandas和Numba
- 通过Matplotlib、Boken、Datashader和Holoviews实现可视化

Anaconda内置包有很多，安装Anaconda之后可以免去我们下载部分包的时间。

Anaconda的安装条件

Anaconda可以在以下系统平台中安装和使用：

- Windows
- Mac OS X
- Linux （x86 / Power8）

Anaconda的安装条件：

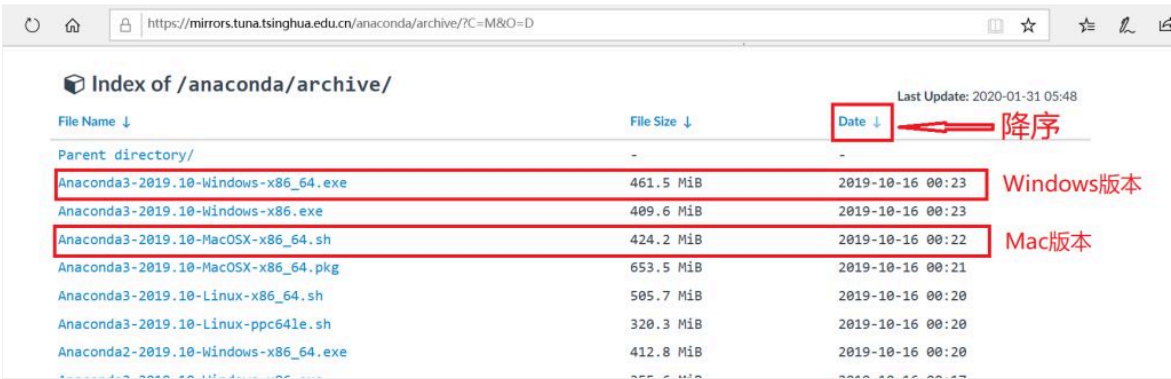
- 系统要求：32位或64位系统均可
- 下载文件大小：约500MB
- 所需空间大小：3GB空间大小

Anaconda下载

1 清华大学镜像下载（推荐）

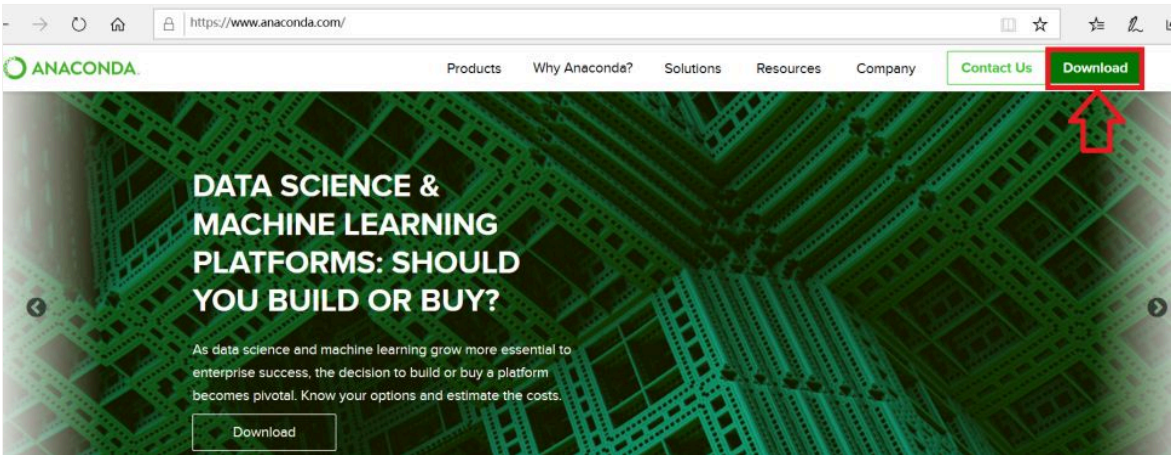
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/

根据用户电脑操作系统选择最新版本进行下载：



2 Anaconda官网下载（下载速度较慢）

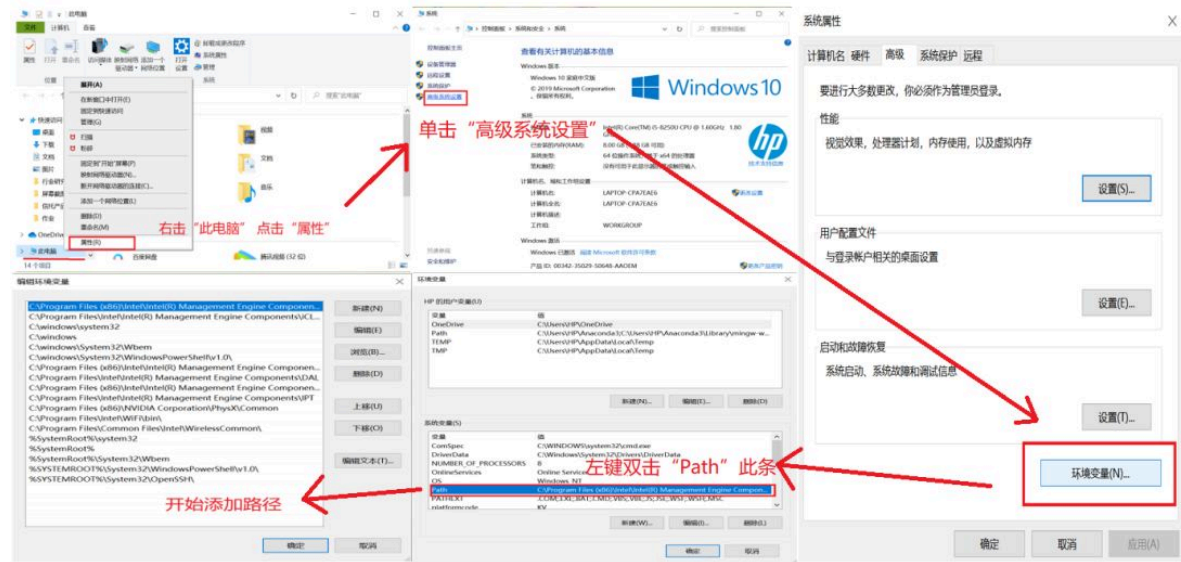
https://www.Anaconda.com



部署环境

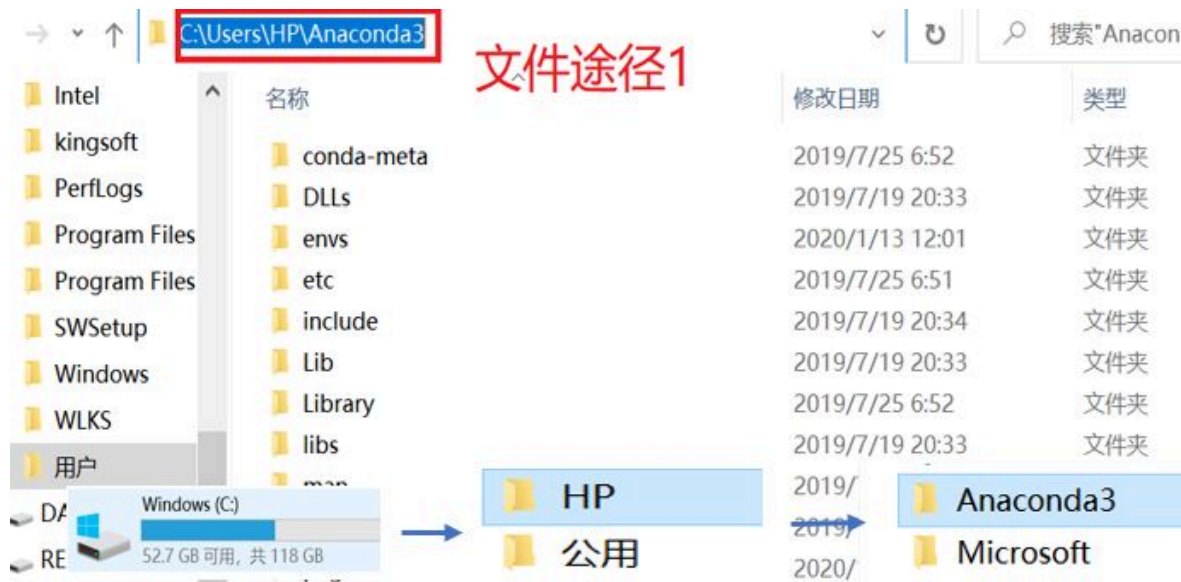
环境的部署需在环境变量中添加三条路径，可解决使用过程中导包失败的问题。

- 以Windows 10为例：

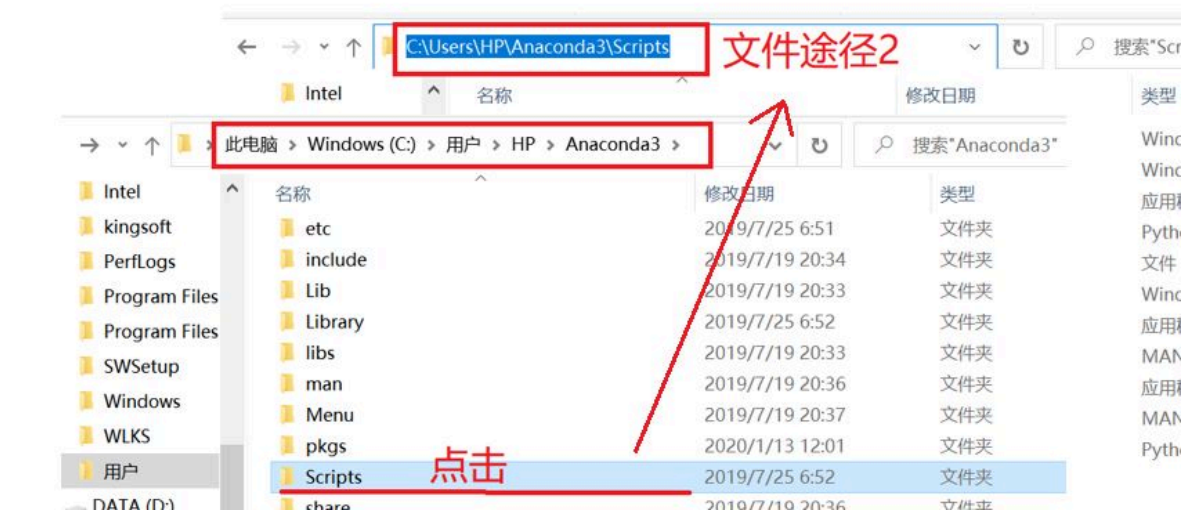


其中需添加的三个文件路径如下：

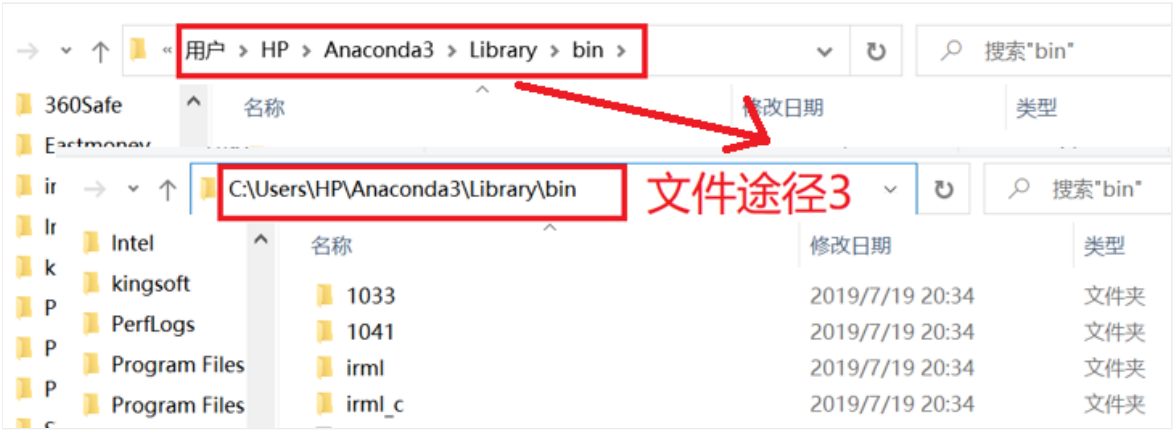
找到Anaconda3文件夹，默认安装情况下，Anaconda3文件可在“我的电脑”C盘>用户>用户名称（用户名因电脑不同而异，图示中为HP）中找到，红框中即为第一个需要添加的路径。



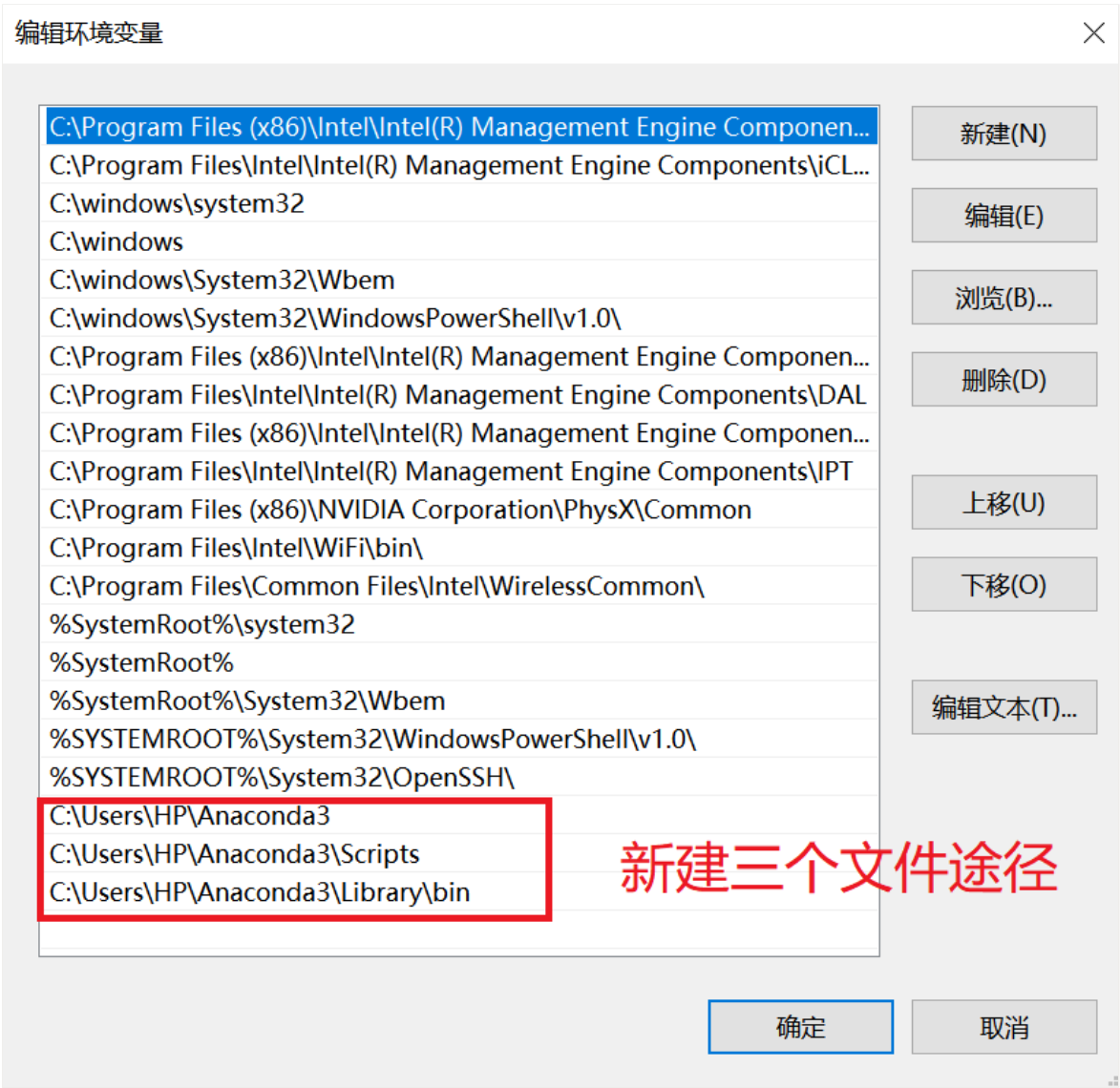
在Anaconda3文件夹中，点击进入Scripts文件夹，得到第二个文件路径。



在Anaconda3文件夹中，先进入Library文件夹，再进入bin文件夹，点击上方路径栏，即得到第三条需添加的路径。



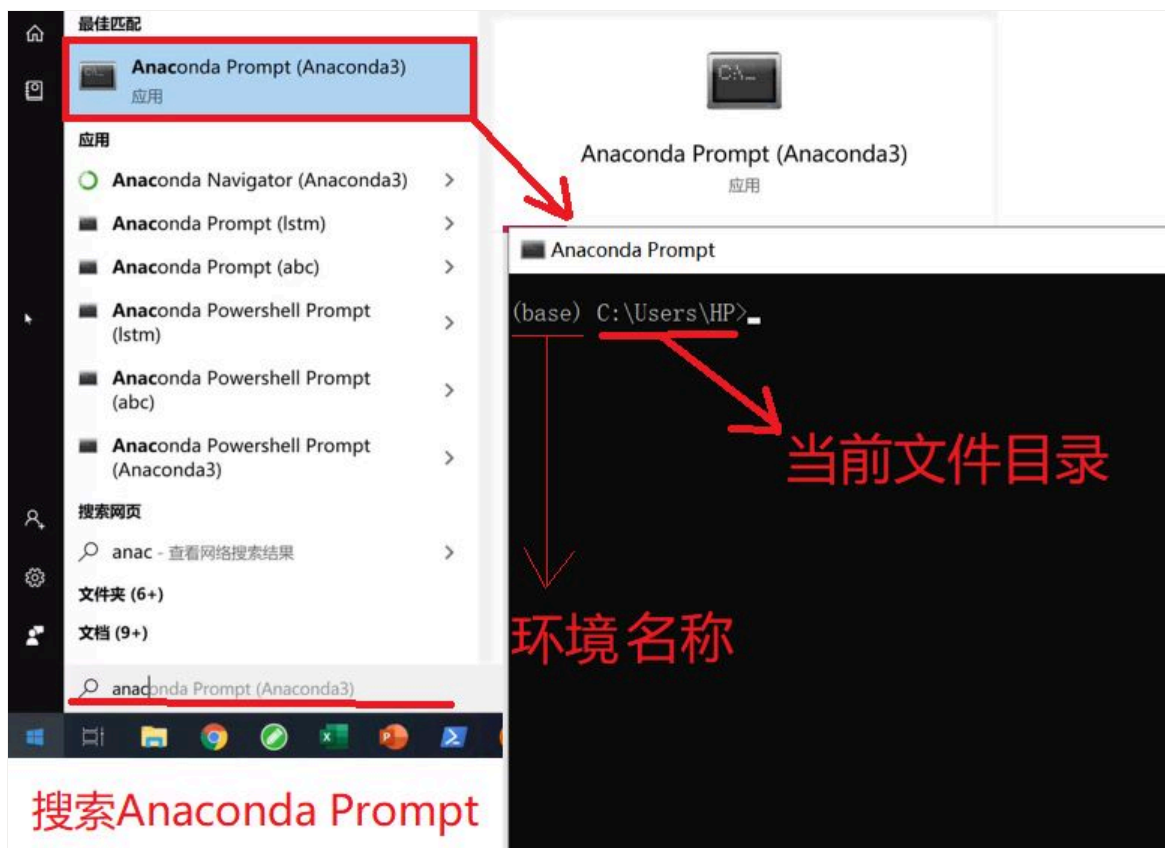
新建三个文件路径：



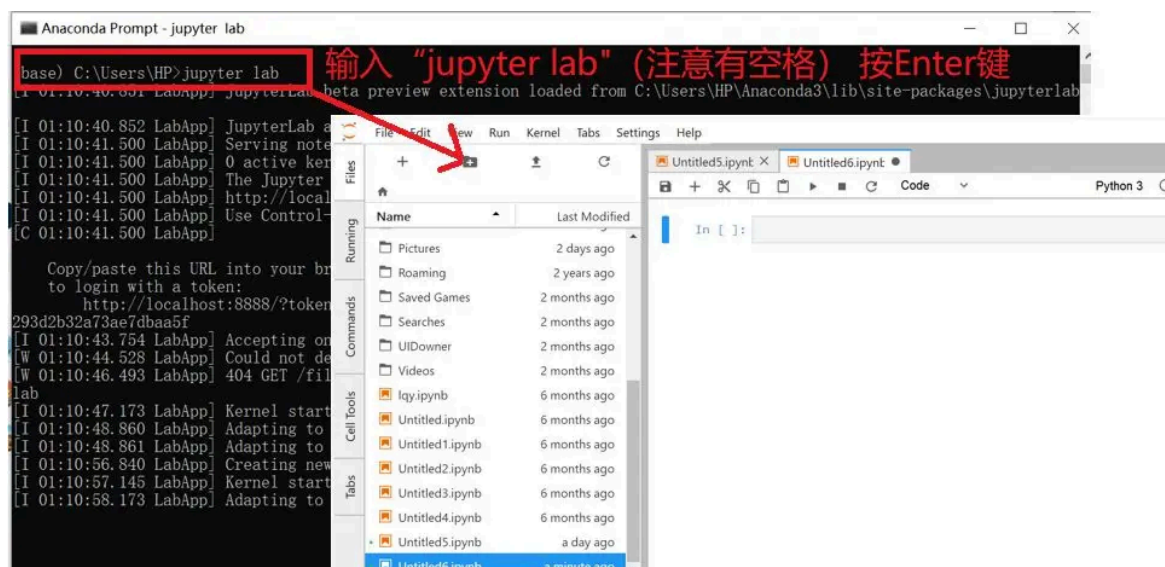
启用Anaconda

注意：Anaconda Prompt中全部使用英文输入法，每个命令中间要有空格。

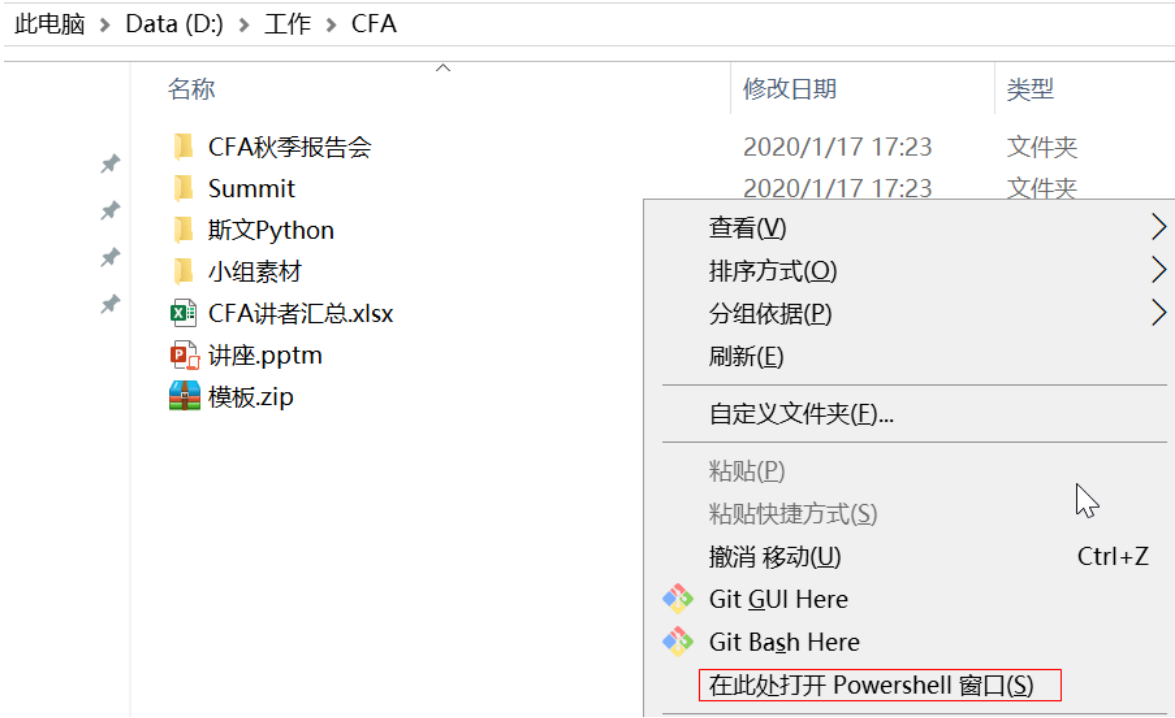
输入“jupyter lab”后，按Enter键开始运行，若在加载过程中有卡顿，jupyter lab的网页没有弹出，可以重复按Enter键，若网页仍未弹出，也可选择打开浏览器，在地址栏输入127.0.0.1:8888(英文输入法) 进入jupyter lab (有时候需要输入token，该值在prompt界面的“token=”处)。



成功启动后，开启jupyter lab的Anaconda Prompt界面不可关闭，否则jupyter lab将无法运行。需要注意的是，我们可以打开多个Anaconda Prompt终端，用来启动多个jupyter lab或者安装包。



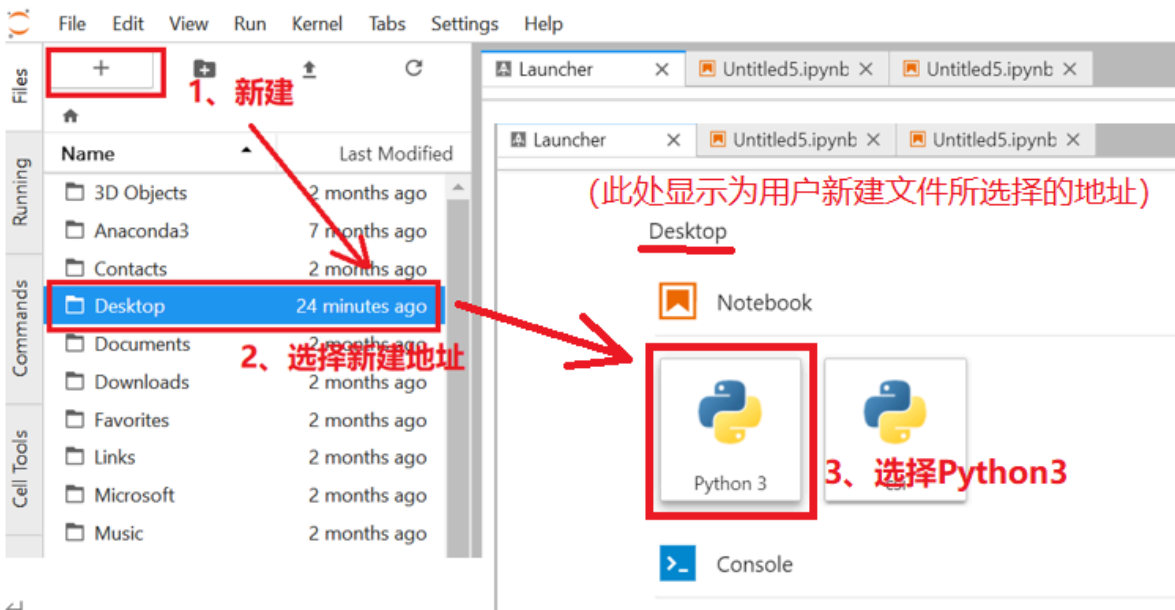
启动后会发现左侧的jupyter lab目录最外层只能到C盘>用户的位置，这样就无法访问到你其他硬盘的资源。Windows用户此时可以去指定目录下按住shift+鼠标右键，会显示有“在此处打开Powershell窗口”，打开后输入“jupyter lab”即可使用。此外，Mac和Windows用户均可在命令行界面下利用“cd 目录名”来更换目录（Windows用户需要先输入如“D:”来切换到D盘才可在该磁盘下执行cd命令）。



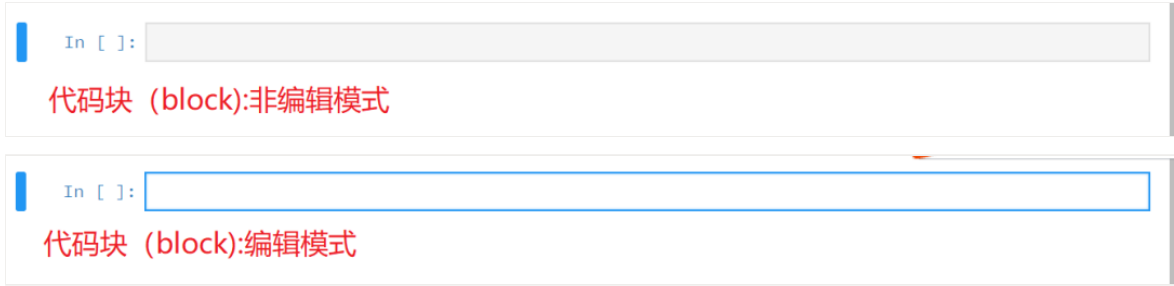
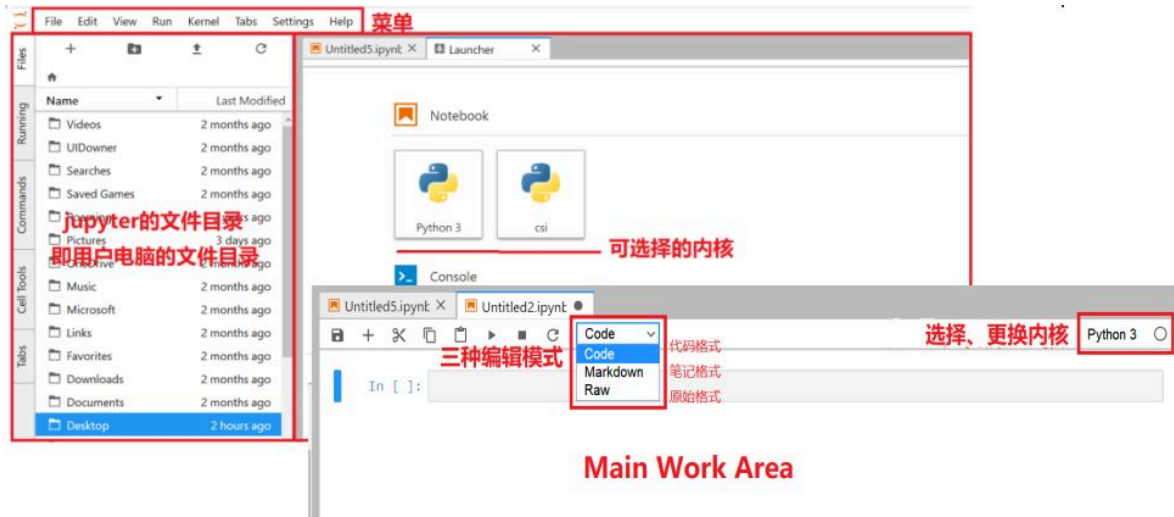
5 Jupyter的使用

新建文件

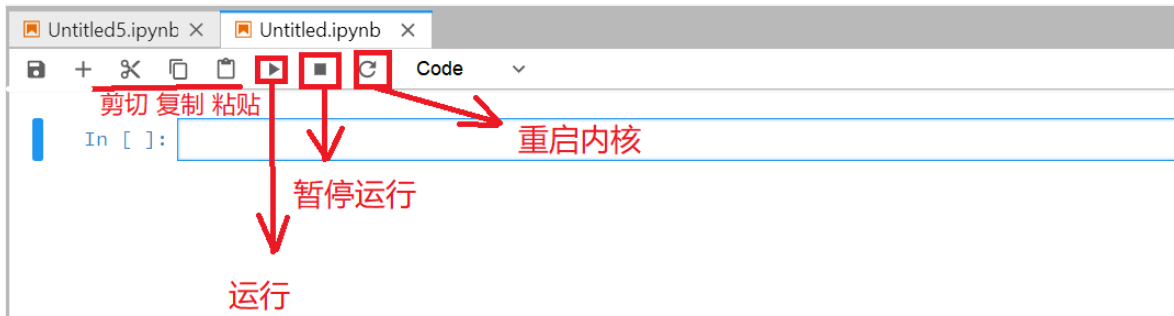
例：在桌面新建一个Python3（即base环境）的文件



使用面板介绍



重启内核相当于清空前面所有运行内容, "In[]"中括号中的数字将重新从"1"开始, 因此, 内核重启后需重新导入包并运行之前所需的内容。



常用操作

快捷键

- 在非编辑模式下操作单元格
 - X 剪切
 - V 粘贴
 - Z 回退
 - B向下插入输入框
 - A向上插入输入框
 - DD删除输入框
 - Shift+M 合并选中的单元格
 - 上下键 选择单元格

- 编辑模式下操作单元格内的内容
 - Shift+Ctrl+ - (减号键) 拆分单元格
 - history 查看误删除的历史代码
 - 以及其他常规编辑快捷键

- 两种模式均可操
 - F11 浏览器全屏
 - Ctrl+B 左侧栏隐藏
 - Ctrl+Enter 执行代码，但不自动移动到下一个输入框
 - Enter+Shift 执行代码且自动移动到下一个输入框

Enter 进入编辑模式
Esc 跳出编辑模式（进入非编辑模式）

文件重命名

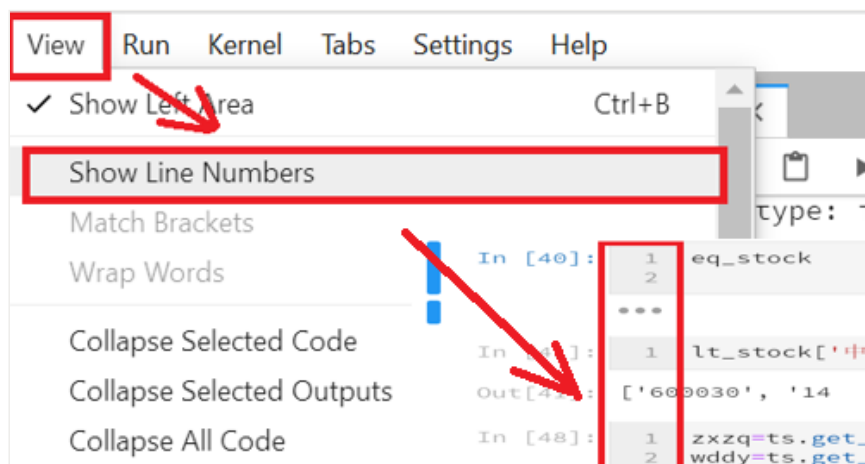
- 方法1:



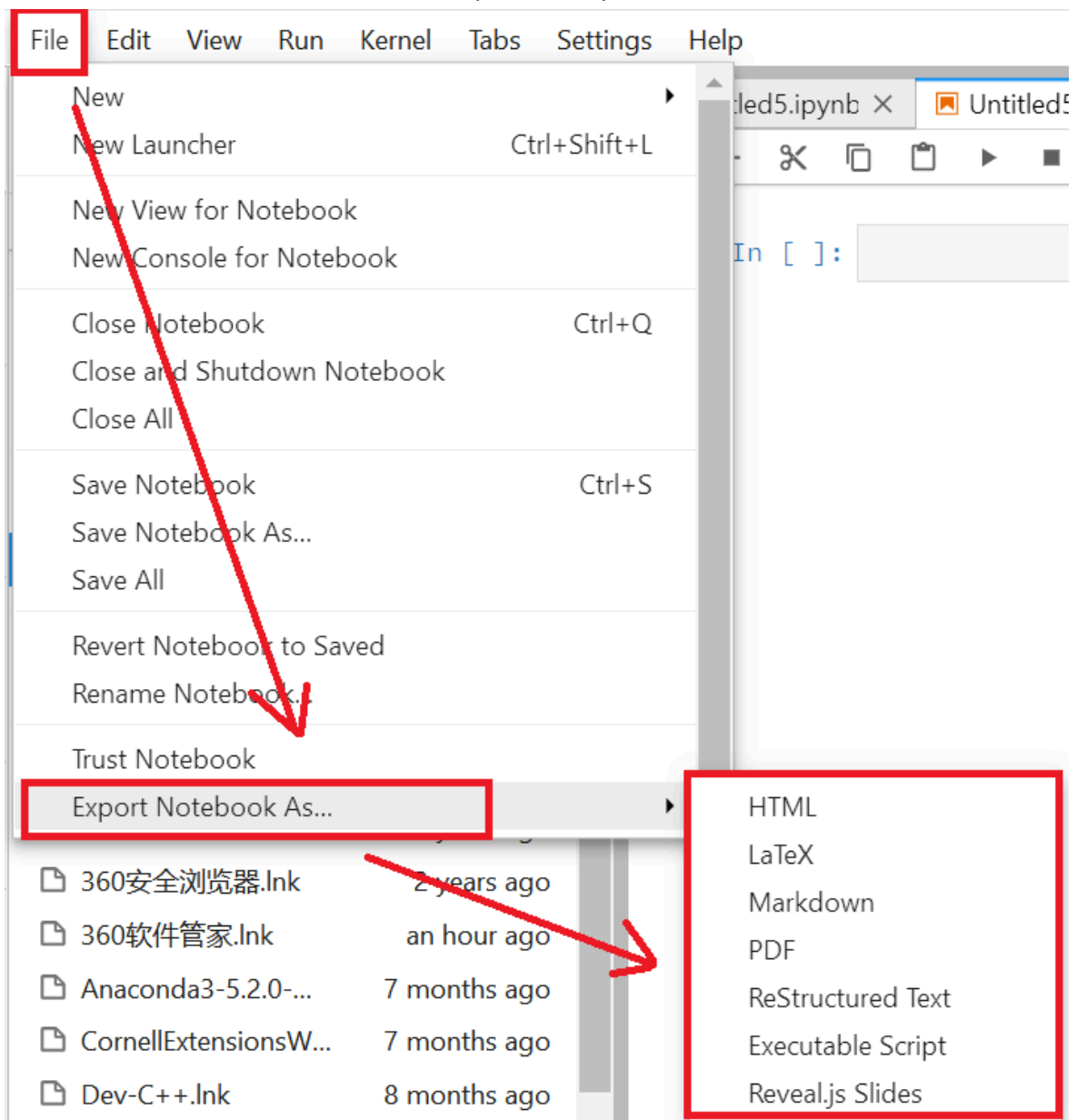
- 方法2:



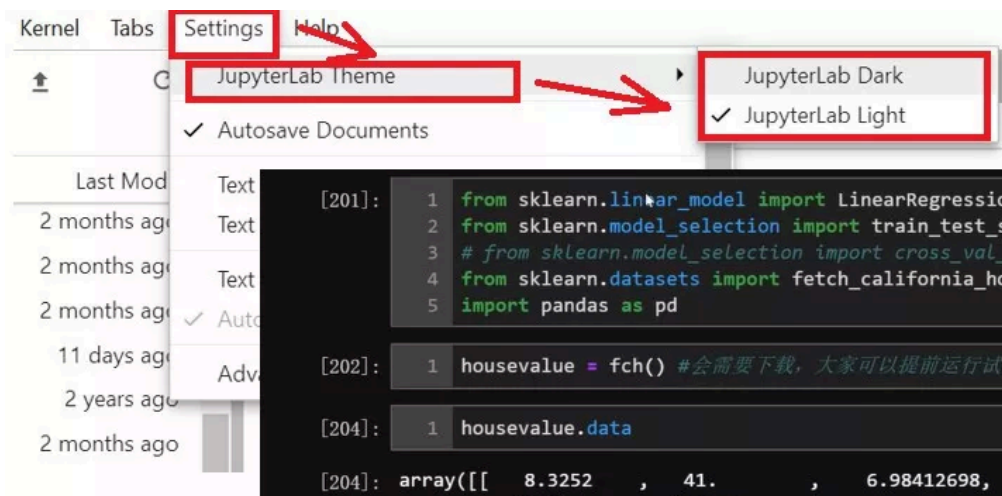
显示代码行数（方便调试bug）



调整导出文件格式，部分格式可能需要安装插件



更换背景色



6 关于Python

Python 常用包介绍

扩展库	简介
Numpy	提供数组支持，以及相应的高效的处理函数
Scipy	提供矩阵支持，以及矩阵相关的数值计算模块
Matplotlib	强大的数据可视化工具、作图库
Pandas	强大、灵活的数据分析和探索工具
StatsModels	统计建模和计量经济学，包括描述统计、统计模型估计和推断
Scikit-Learn	支持回归、分类、聚类等的强大的机器学习库
Keras	深度学习库，用于建立神经网络以及深度学习模型
Gensim	用来做文本主题模型的库，文本挖掘可能用到

这里首先介绍Python的三个基础扩展包的用途和下载渠道，这三个也是Python数据分析的三剑客：pandas（数据分析包）、numpy（科学计算包）、matplotlib（绘图包）

Pandas（Panel Data）是基于Numpy的一种工具，该工具是为了解决数据分析任务而创建的。Pandas纳入了大量库和一些标准的数据模型，提供了高效操作大型数据集所需的工具，最具有统计意味的工具包，其在某些方面优于R软件，数据结构含一维的Series，二维的DataFrame(类似于Excel或者SQL中的表)和三维的Panel。

- Pandas学习网站参考：
<https://www.py pandas.cn/>

Numpy(Numerical Python) 是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。NumPy 通常与 SciPy（Scientific Python）和 Matplotlib（绘图库）一起使用。

- Numpy学习网站参考：
<https://jalammar.github.io/visual-numpy/>

Matplotlib是受MATLAB的启发构建的。MATLAB是数据绘图领域广泛使用的面向过程、利用函数调用的语言和工具，matplotlib有一套完全仿照MATLAB的函数形式的绘图接口，在matplotlib.pyplot模块中，这套函数接口方便MATLAB用户过度到matplotlib包。

- Matplotlib学习网站参考：
<http://matplotlib.org/>
<http://matplotlib.org/examples/index.html>
<http://matplotlib.org/gallery.html>

常用包的安装

介绍完了这三个包后，让我们开始尝试下载安装。在Anaconda中由于内置了很多常用包，包括上述三个，但是如果以后需要安装新的Python包的话，还是需要自己利用下图所示的通过pip来安装，但是直接运行会因为一些众所周知的原因会下载缓慢。

```
(base) C:\Users\HP>pip install pandas  
(base) C:\Users\HP>pip install matplotlib
```

为了方便国内用户使用，很多机构都将这些包存储了一份到自己的镜像源中，国内用户在安装的时候可以直接通过 `pip install -i 镜像源名称 要下载的包名` 进行下载，若卸载包输入 `pip uninstall 要卸载的包名` 即可。（numpy为Anaconda自带的包，不用重新下载）

这里以使用清华镜像源下载/卸载pandas为例：

▪ 下载包：

```
(base) C:\Users\HP>pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple
```

▪ 卸载包：

```
(base) C:\Users\HP>pip uninstall pandas
```

其他获得包的途径有：

清华：

<https://pypi.tuna.tsinghua.edu.cn/simple>

阿里云：

<http://mirrors.aliyun.com/pypi/simple/>

中科大：

<https://pypi.mirrors.ustc.edu.cn/simple/>

华中理工大学：

<http://pypi.hustunique.com/>

山东理工大学：

<http://pypi.sdutlinux.org/>

豆瓣：

<http://pypi.douban.com/simple/>

相关包下载完毕后，还需将扩展包在jupyter lab中进行导入（pd、np、plt为三个包的常用缩写），导入之后在编写代码过程中，利用as的语句可以让我们通过包的缩写来调用。

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [4]: np.array(5)
```

```
Out[4]: array(5)
```


conda和pip是包的管理工具。之所以需要conda是因为在对Python的使用时我们会大量引用第三方别人编写好的Python包来加快我们完成数据分析或者开发的任务，但是由于Python版本的不断更新以及发布者不断丰富自己发布的包，所以会常常出现你要同时使用不同版本的包的问题，此时会带来环境的混乱。

在这样的需求之下，conda应运而生了。**conda是跨平台的包和环境管理器，可以安装和管理多个版本的软件包及其依赖关系；pip则是用于Python 安装包的工具。**本文中的Anaconda也属于conda类，是内置了很多Python包的conda环境，除此之外还有Miniconda这种无其他包的更轻便的conda环境。

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，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；而pip没有内置的环境支持，而是依赖其他工具来创建隔离环境。

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，虽然conda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。

总体上，conda相比于pip而言，其下载包的依赖更为齐全。只是conda包含的Python包是经过一定程度的筛选和优化的，而pip是从<https://pypi.org/>中获取所有的包，你我都可以对自己封装好的代码进行上传，所以conda在Python包的数量上可能远不如pip的多，但是使用起来更为安全。

类别	conda	pip
管理	二进制	wheel 或源码
需要编译器	no	yes
语言	any	Python
虚拟环境	支持	通过 virtualenv或venv等支持
依赖性检查	yes	屏幕提示用户选择
包来源	Anaconda repo和cloud	PyPi

Python 的 简 单 运 行

输入你的第一行代码：Hello World

Python的“Hello World”

Python的Hello World 1

```
[4]: print('Hello World')
Hello World
```

若出现bug，运行会弹出如下红框，可从中寻找错误原因，或将bug复制到浏览器上进行搜索，查找网上是否有给出针对此问题的解答。在debug过程中，注意要从根源错误处着手解决（例如下图中阴影部分的错误），并利用网络搜索资源，国内有CSDN和简书等技术交流网站。一般来说你碰到的bug前人都已经踩过雷了，解决病症的有效途径是冷静下来把自己的症状说清楚，要学会提问题。

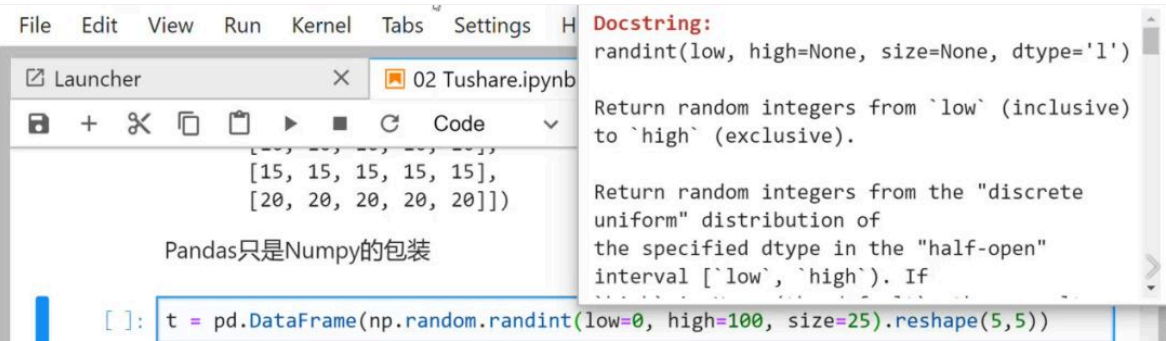
```
1 pd.set_option('display.max_rows', 100)

-----
OptionError                                Traceback (most recent call last)
<ipython-input-248-612259576f2e> in <module>
----> 1 pd.set_option('display.max_rows', 100)

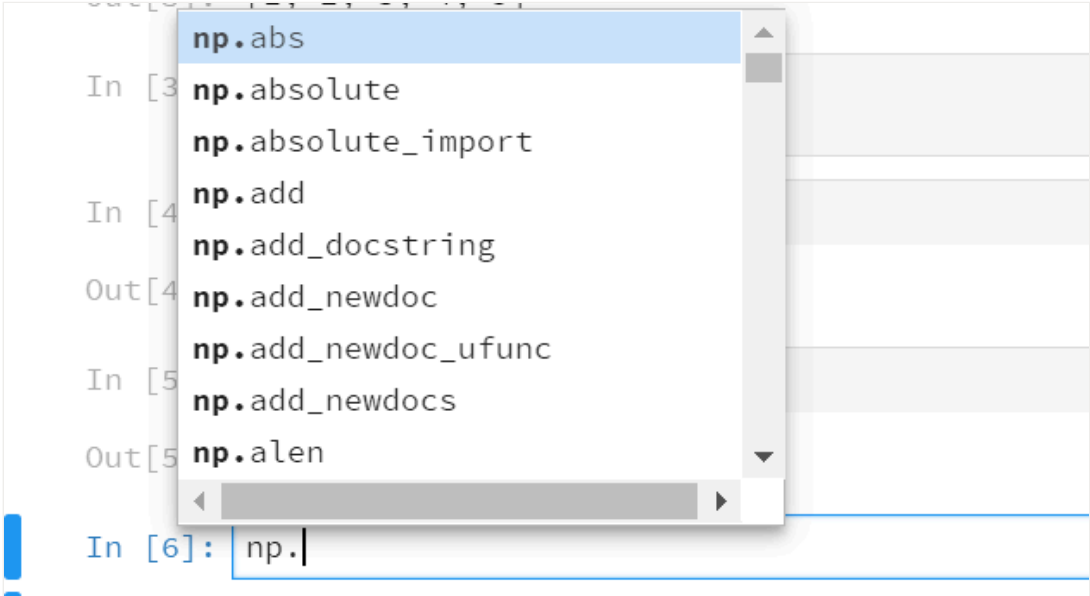
~\Anaconda3\envs\bili\lib\site-packages\pandas\_config\config.py in __call__(self, *args, **kwargs)
    231
    232     def __call__(self, *args, **kwargs):
--> 233         return self.__func__(*args, **kwargs)
    234
    235     @property

~\Anaconda3\envs\bili\lib\site-packages\pandas\_config\config.py in _set_option(*args, **kwargs)
    121
    122     for k, v in zip(args[::2], args[1::2]):
--> 123         key = _get_single_key(k, silent)
    124
```

当遇到不熟悉的函数，将光标移至函数后的括号中，先按住Shift键，再按Tab键（有时需要按两下Tab），即会弹出如图所示方框，内含对该函数的详细介绍。



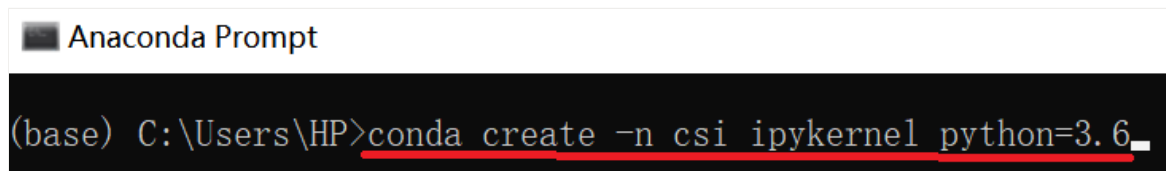
若希望知道某扩展包（图例中为numpy）的可用函数，将光标移到“np.”后，再按下Tab键，即可显示如下所有可用函数。



7 创建新环境（非必须操作）

前文介绍了conda是一个环境管理的工具，这里以创建Python 3.6的环境为例（由于一些众所周知的原因，利用conda安装包也会很慢，此时也需要更换镜像，大家有兴趣可以搜索"anaconda更换镜像"，这里不再赘述）：

- 在Anaconda Prompt中输入如下内容，其中“csi”为环境名称，可由用户自行拟取。



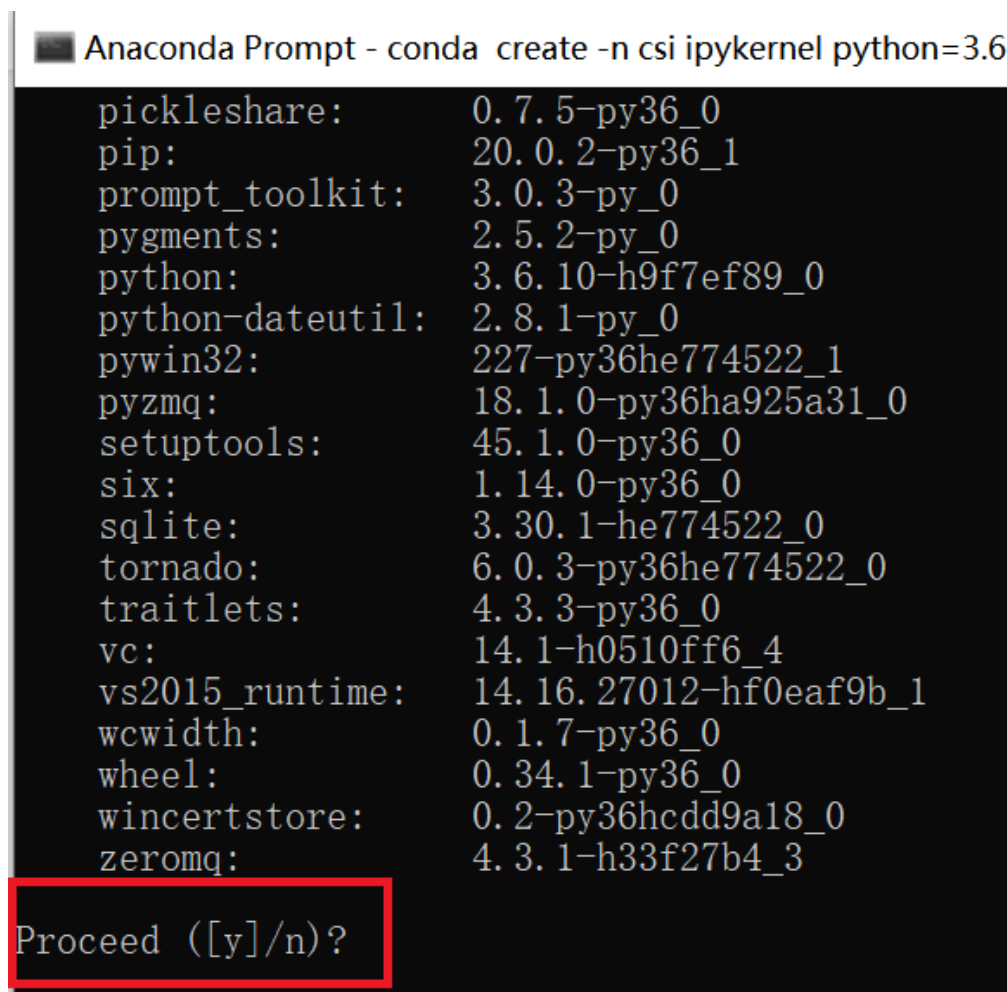
```

Anaconda Prompt

(base) C:\Users\HP>conda create -n csi ipykernel python=3.6_

```

- 若下载过程暂停在“Proceed”处，按Enter键，或先输入“y”再按Enter键。



```

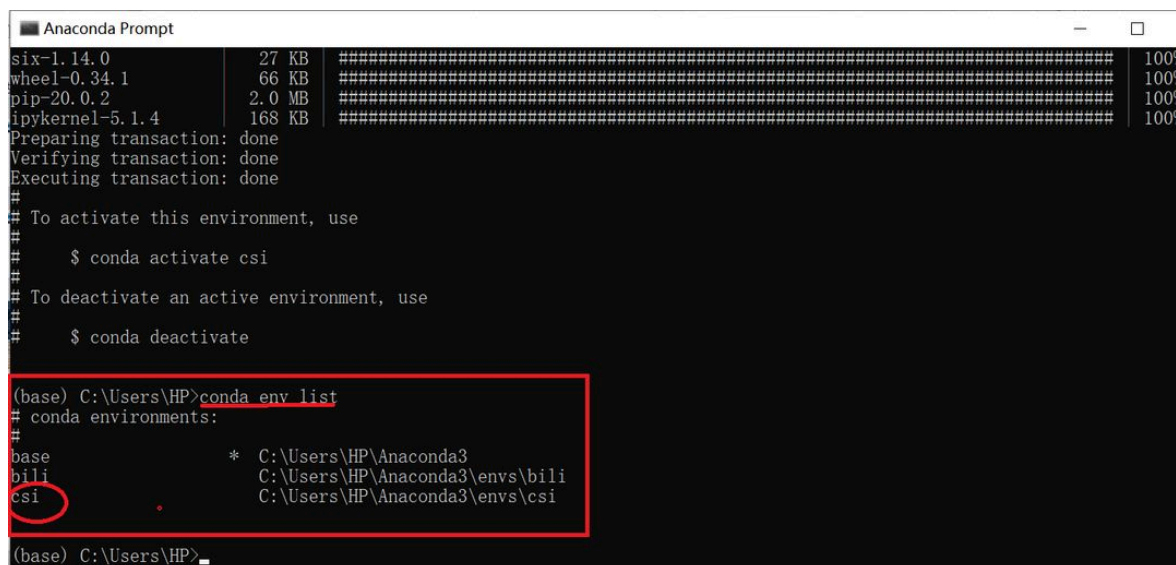
Anaconda Prompt - conda create -n csi ipykernel python=3.6

pickleshare:      0.7.5-py36_0
pip:              20.0.2-py36_1
prompt_toolkit:   3.0.3-py_0
pygments:        2.5.2-py_0
python:          3.6.10-h9f7ef89_0
python-dateutil: 2.8.1-py_0
pywin32:         227-py36he774522_1
pyzmq:           18.1.0-py36ha925a31_0
setuptools:      45.1.0-py36_0
six:             1.14.0-py36_0
sqlite:          3.30.1-he774522_0
tornado:         6.0.3-py36he774522_0
traitlets:       4.3.3-py36_0
vc:             14.1-h0510ff6_4
vs2015_runtime:  14.16.27012-hf0eaf9b_1
wcwidth:        0.1.7-py36_0
wheel:           0.34.1-py36_0
wincertstore:    0.2-py36hcdd9a18_0
zeromq:          4.3.1-h33f27b4_3

Proceed ([y]/n)?

```

- 下载成功后可输入如下指令，查看现有的所有环境并找到csi环境。



```

Anaconda Prompt

six-1.14.0          27 KB ##### 100%
wheel-0.34.1       66 KB ##### 100%
pip-20.0.2         2.0 MB ##### 100%
ipykernel-5.1.4    168 KB ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

# To activate this environment, use
#
#   $ conda activate csi
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\HP>conda env list
# conda environments:
#
base                * C:\Users\HP\Anaconda3
bili                C:\Users\HP\Anaconda3\envs\bili
csi                 C:\Users\HP\Anaconda3\envs\csi

(base) C:\Users\HP>

```

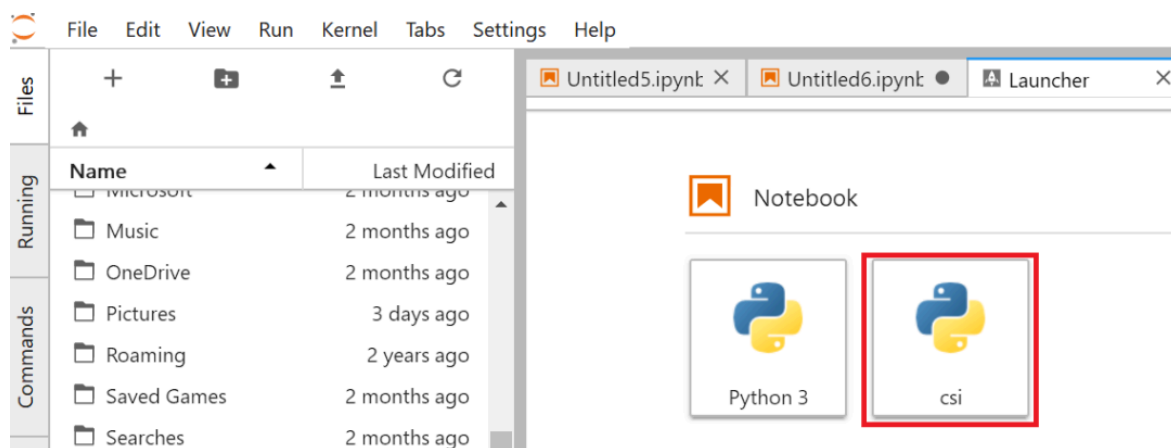
- 接着我们要激活csi环境（环境名称自拟）

```
(base) C:\Users\HP>conda activate csi
```

- 挂载至我们的base环境（默认环境，一进来就是）的Jupyter上，挂载只在第一次创建csi环境的时候需要，之后无需再进入csi环境。但是如果需要利用pip/conda安装包，还是需要激活csi环境进行操作。

```
(csi) C:\Users\HP>python -m ipykernel install --name csi --user
Installed kernelspec csi in C:\Users\HP\AppData\Roaming\jupyter\kernels\csi
```

- 切换回我们的base环境（注意：每次都是通过base环境启jupyter）再次启用jupyter lab后，此时可以看到新可选择内核中增添csi环境选项，点击即可选择编写Python程序所需内核。



8 Jupyter Lab与Jupyter Notebook

- 通过Anaconda Prompt开启 jupyter lab和 jupyter notebook

```
(base) C:\Users\HP>jupyter lab
```

```
(base) C:\Users\HP>jupyter notebook
```

- 两者比较

Jupyter Notebook 是一个款以网页为基础的交互计算环境，可以创建Jupyter的文档，支持多种语言，包括Python, Julia, R等等。广泛用于数据分析，数据可视化和其他的交互和探索性计算中。

Jupyter Lab 是包括了Notebook的下一代用户界面,其对于Jupyter Notebook有着完全的支持，通过使用JupyterLab，能够以灵活，集成和可扩展的方式处理文档和活动：

- 可以开启终端，用于交互式运行代码，完全支持丰富的输出
- 支持Markdown, Python, R, LaTeX等任何文本文件
- 增强notebook功能
- 更多插件支持

个人理解为：jupyter notebook功能性较强，而jupyter lab工程性较强。

本期撰写人：李清宇

南京审计大学2018级CFA3班

○
南京审计大学CFA
○



推进CFA学术交流，集聚南审CFA校友资源