

H02 - The Guidance to Install and Configure IDEs

Overview

- 本章节主要介绍学习Python的主要开发工具，以及一些能提高工作效率的小技巧，分安装部分与配置部分。
- 安装部分：
 - 为了使查看、编写代码的轻量级工作能够流畅运行，我们推荐使用Sublime Text 3这款软件。由于安装其他软件时也会涉及很多种格式文件的打开，因此我们首先安装它。
 - 在Python官网可以下载到原生的Python安装包，里面只涵盖了最基本的几个Python库，可不下载。
 - 在Anaconda官网可以下载到Anaconda安装包，它预先配置好了包含Python库的200多个常用库，其中的Jupyter Notebook组件是后续数据分析的首选平台，另外一个Spyder组件也是一个界面非常友好的IDE。
 - 为了支持后续更大工程量的开发工作，我们还提供了PyCharm专业版的下载方式与基于教育邮箱的激活方式。
- 配置部分：
 - 主要针对一些比较关键的启动速度、启动位置、插件安装等做了一些关键性描述，因为这些资料是比较少的，都是平时自己在开发过程中血和泪的经验，希望对大家有用。
 - 至于那些怎么正常使用软件的操作，知乎、CSDN等资料非常多，可在以后穿插性讲解，本章主要从系统的角度带大家理解如果正确安装软件与配置软件，尤其是扩展软件。
 - 加油！

Install

说在前面，必看！

首先，要说一下安装软件的方法论，很多非CS专业的学生在学习编程的时候很是痛苦，因为连软件都不知道怎么安装，为什么呢？因为现在的软件真的大啊，Adobe每款软件是2G级别的，专业的Visual Studio是10G级别的。

但，这不是装不了的借口，20G的单机游戏你是怎么装起来的？

Anyway，其实装软件是有固定流程的，也是有方法论的，只要想清楚一下几点就好了。

1. 个人使用的操作系统目前分Windows、Mac和Linux三大类，其中Windows的可执行程序文件的后缀是.exe，Mac是.dmg，Linux是.sh，下载的时候注意区分。
2. 要装Python2还是3版本的？装3，不用纠结了，再次强调，Python2的时代彻底谢幕了，此处应该来一首《Intro: the dawn》。
3. 要装专业版、企业版、社区版还是学生版？一般来说能装专业版就装专业版，功能更全一些；其次如果自己是学生，装学生版是最稳

的，一般只要验证一下你的.edu邮箱就可以，我们本章节要安装的PyCharm就是推荐大家以学生身份装专业版，这是极其完美的。

4. 要装全套组件还是部分组件？全能版还是核心功能版？这个有一些讲究，对于我来说，一般只要内存够、硬盘够、带的动，我都会装全套组件，比如本章节的Anaconda这款软件我们会推荐大家安装全套的，而非只有核心功能的MiniConda。但这种方法并不一定是永远合适的，毕竟Anaconda不大，如果你装过SAS，就知道你可能绝不会装全套的。
5. 要装安装版的（executable）还是便携版的（Portable）？一般来说，非专业人士装安装版的，这会在学习使用的时候提供最大的便利。我们要安装的Sublime就是安装版的，如果你装便携版的，你右键一个文件是没有“使用Sublime打开”这个选项的。因为，安装版的软件会一般会把软件的可执行程序写进注册表。我有一个学习软件工程的舍友，U盘里放了一个便携版的Dev，用来开发C++的体积很小的软件，公司、学校随插随用。
6. 要不要装插件？现在很多的软件其不单单只是提供完成一个目标的功能，更多优秀的软件都把自己做成了一个平台，在这些软件上会衍生大量为完成特定功能、由无数的极客贡献的插件，要不要装呢？装！必须装，我们会挑一些推荐给大家。其实不仅是开发Python，就是平时用来上网的浏览器，无论是谷歌、火狐还是360，甚至是猎豹之流，都应该装一些有用的插件，以提高浏览的体验。我们在之后的环节会讲一些。
7. 最后，如果有问题，请以我为准！接下来，请大家把网速拉到1000兆，下下下，装装装，配配配，不要恋战！

安装火狐浏览器：一款优秀的、程序员专用的、快速的、免费的、不用翻墙的、特别适合学习爬虫的浏览器

下载地址：[Firefox 火狐浏览器 - 全新、安全、快速 | 官方最新下载](#)



快，只为更好。

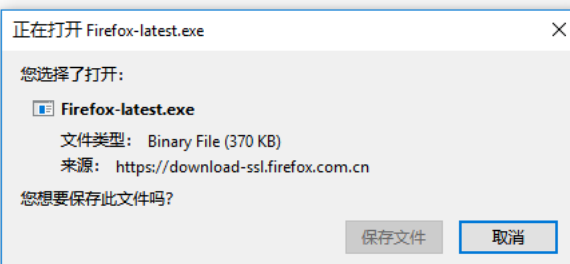
Fast for good.

立即下载

Firefox 隐私声明

[更多系统和语言下载](#)

[太平洋下载](#) [腾讯软件中心](#)

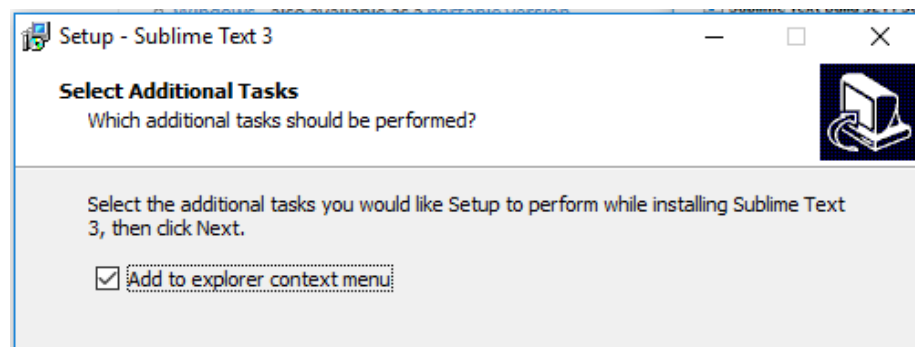
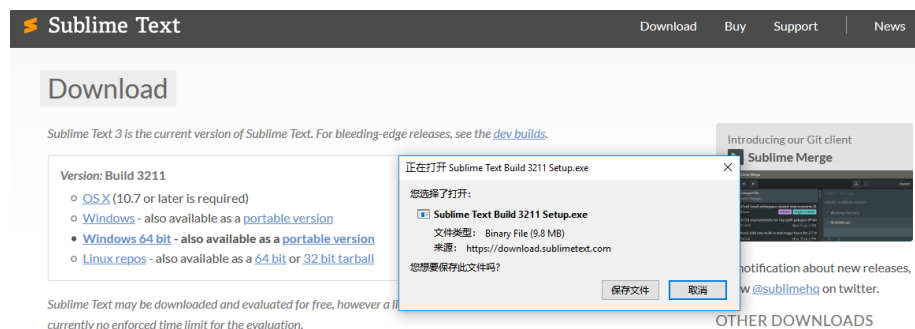




安装Sublime Text 3: 一款优秀的、程序员专用的、高颜值的、快速的、支持所有语言的、不用保存的、一用就会爱上的编辑器

下载地址: [Download - Sublime Text](#)

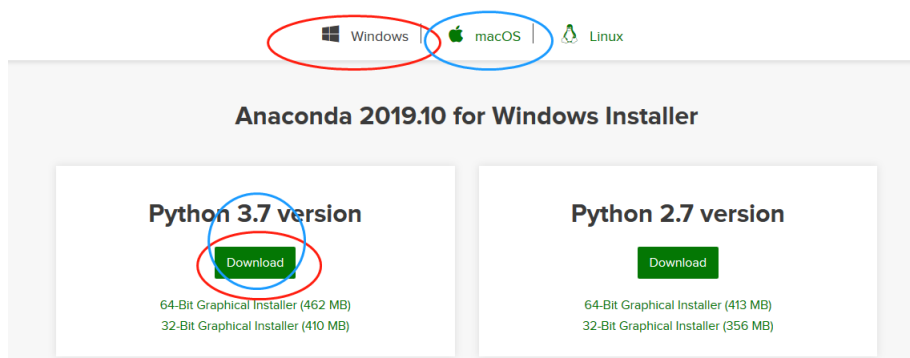
不要下载portable version, 除非你能master它。



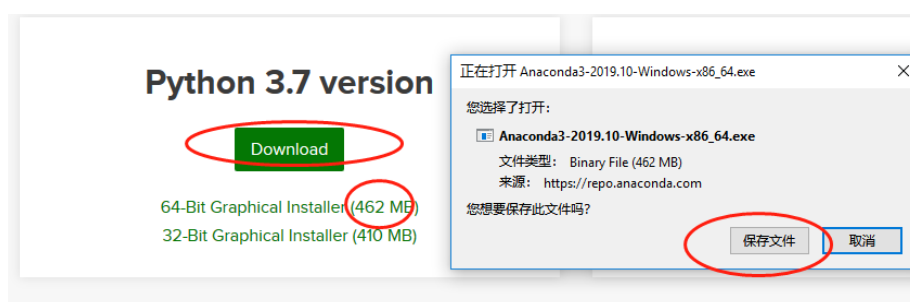
安装Anaconda 3: 一款革命性的、交互体验的、数据分析必备的、受众群体巨大的、十分强大的包管理平台、集成开发环境

下载地址: [Anaconda Python/R Distribution - Free Download](#)

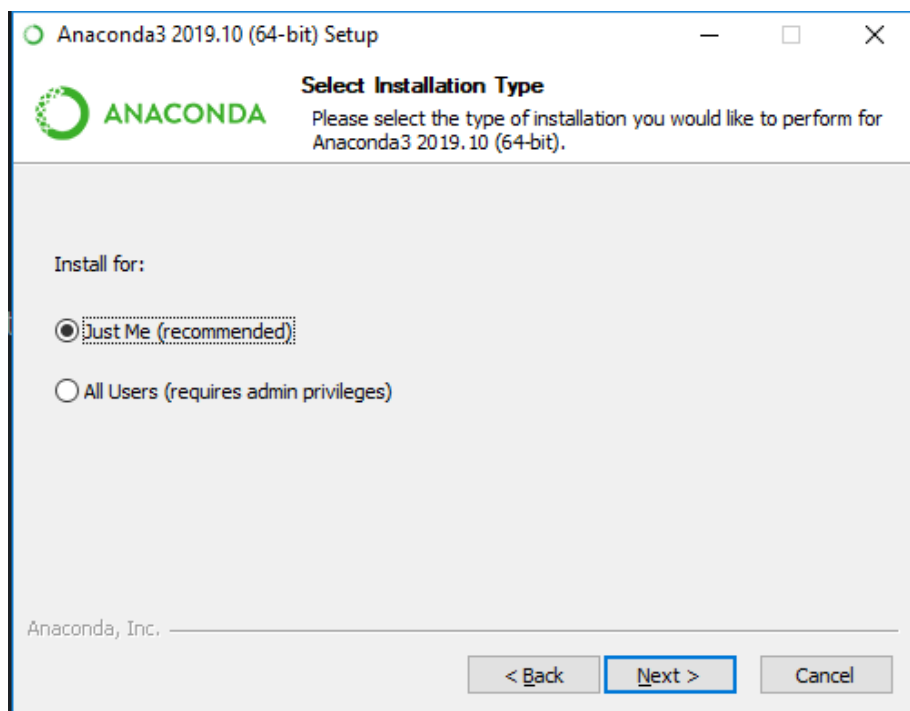
下载页面已经提供了Windows版和Mac版的链接，点击即可！



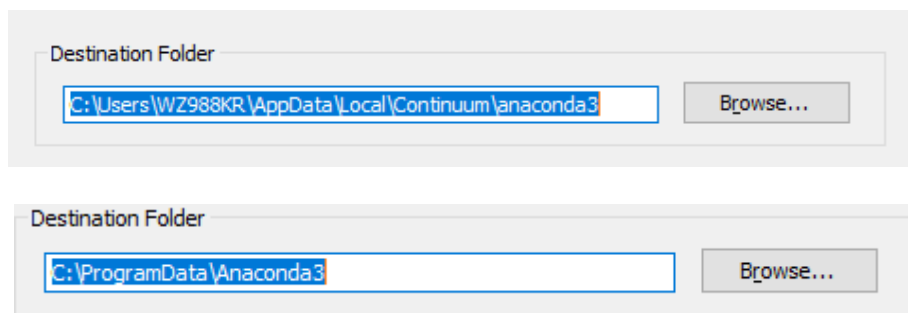
目前大家的电脑都是64位，所以直接点击下载即可，默认下载的就是462兆的64位安装包。



注意，在下图，有 **Just me** 和 **All Users** 两个选项，其实都可以，现在大家的电脑都是个人自己用，这两者严格意义上没有太大区别，一个明显的区别是安装的位置会不一样。以windows为例，**Just Me** 会安装在 **UserProfile** 文件夹下的深处（具体可参见我们《南川P.F.A教程》第一章关于环境变量的相关内容），而 **All Users** 会安装在C盘的 **ProgramData** 文件夹下，即 **C:\\ProgramData\\Anaconda3**。

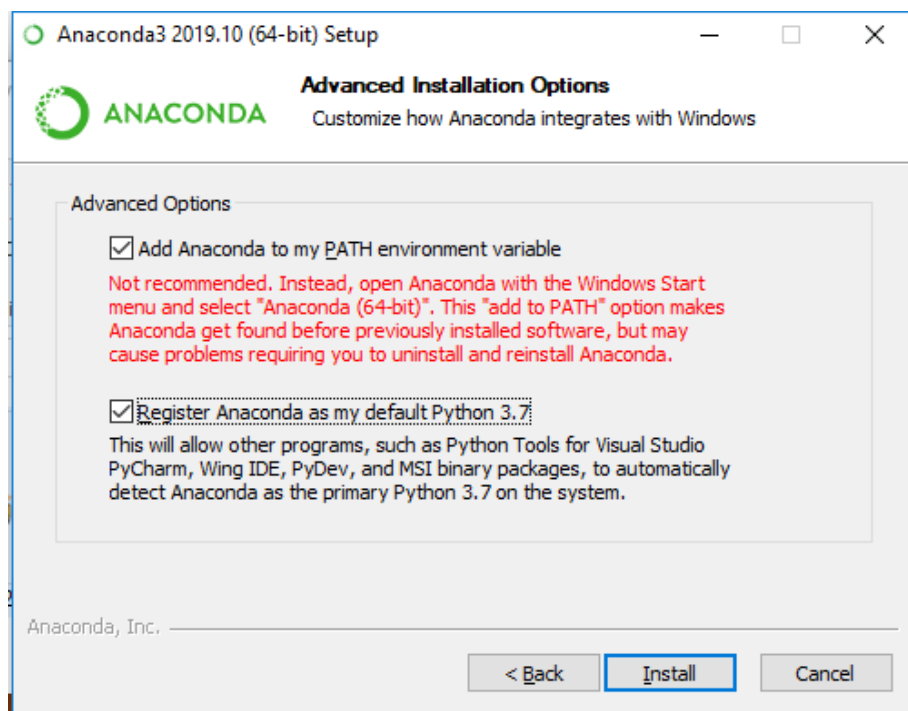


如下所示：



友情提醒，如果用的是公司的电脑，建议选择 **Just Me**，不然要经常提示输入密码以提高权限。

接下来的选项比较重要，两个都勾选：



如果第一个选项不勾选，就相当于装了一个便携版，这直接导致的结果就是，后续如果我们让你在 **cmd** 中输入 **Jupyter Notebook** 你会打不开。

而第二个选项不勾选的话，如果你已经装了其他Python，则可能会默认使用那个版本的Python，包括PIP等等。新手直接把Anaconda作为主要的工作环境，一个个地装一些需要的包，这种感觉还是很棒的，因为你会感受到你的Anaconda越来越强大，而且很少提示缺少包。

而对于老手来说，因为他们的工程可能会涉及到要发布，所以就要尽量只使用必要的包；并且由于项目多了，所以要隔绝环境避免污染，在这样的情况下，他们才会根据项目的需求考虑使用虚拟环境还是使用现有的环境，使用Anaconda的科学运算包、还是使用Python的原生环境、还是用之前配置的一个特殊用途的环境。

我想我应该讲清楚了叭，Good Luck!

安装PyCharm: 一款专业的、**Python**网站开发首选的、变态般强大的、沉浸式的、对学生友好的集成开发环境

为顺利安装与激活，此处要用学生邮箱。学生邮箱注册地址：[JetBrains Products for Learning](#)

Apply with: **UNIVERSITY EMAIL ADDRESS** ISIC/ITIC MEMBERSHIP OFFICIAL D

Status: ☒ I'm a student ☐ I'm a teacher

Level of study: Undergraduate

Is Computer Science or Engineering your major field of study?
☒ Yes ☐ No

Graduation date: Jun 01, 2021
Choose expected graduation date.

Email address: **xxxx@xxx.edu.cn**
I certify that the university email address provided above is valid and belongs to me.

Name: xx xxx
The software product(s) will be registered to your real name.

注意，邮箱是最重要的，其他一切都不重要！一定要要是你能登陆的含**edu**的教育邮箱！

Email address: 16...53@stu.r...edu.cn
I certify that the university email address provided above is valid and belongs to me.

Country / region: China

☐ I am under 13 years old

☒ I have read and I accept the [JetBrains Account Agreement](#)

☐ I consent to the use of my name, email address, and location data in email concerning JetBrains products held or services used by me or my organization

APPLY FOR FREE PRODUCTS

JetBrains Products for Learning

Thank you!

Please follow the instructions in the verification email we've sent you to 160[REDACTED]@stu.i[REDACTED].edu.cn. You can link JetBrains Educational Pack to another email address later.

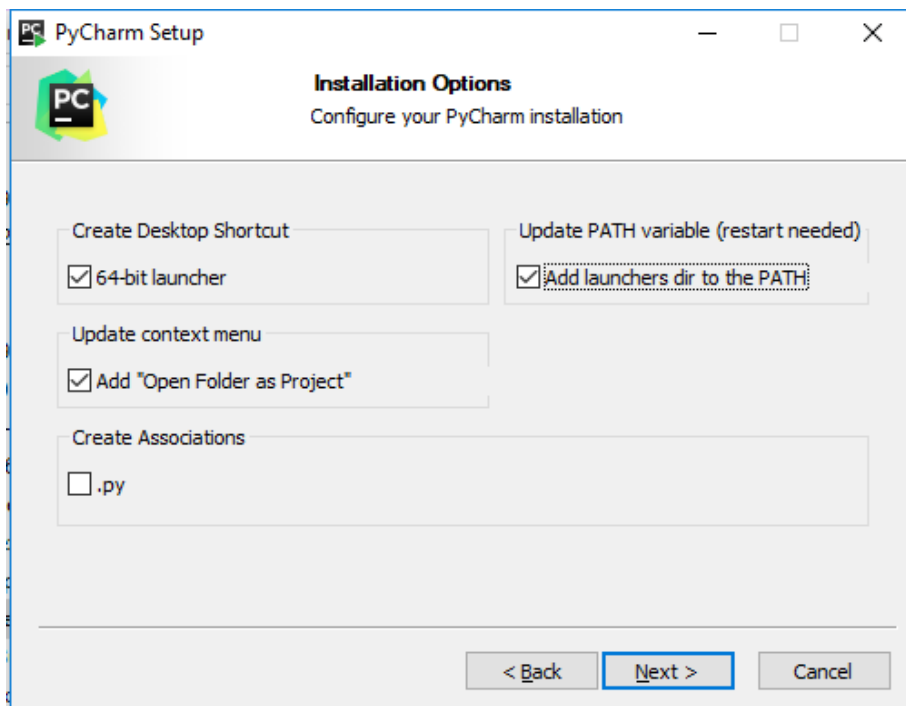
« [Discover our products](#)

接下来，你就要登陆你的教育邮箱进行激活，邮件可能会在垃圾箱内！。

激活成功后，便可以到另一个下载界面了，选择最新的、对应的PyCharm专用版下载即可！

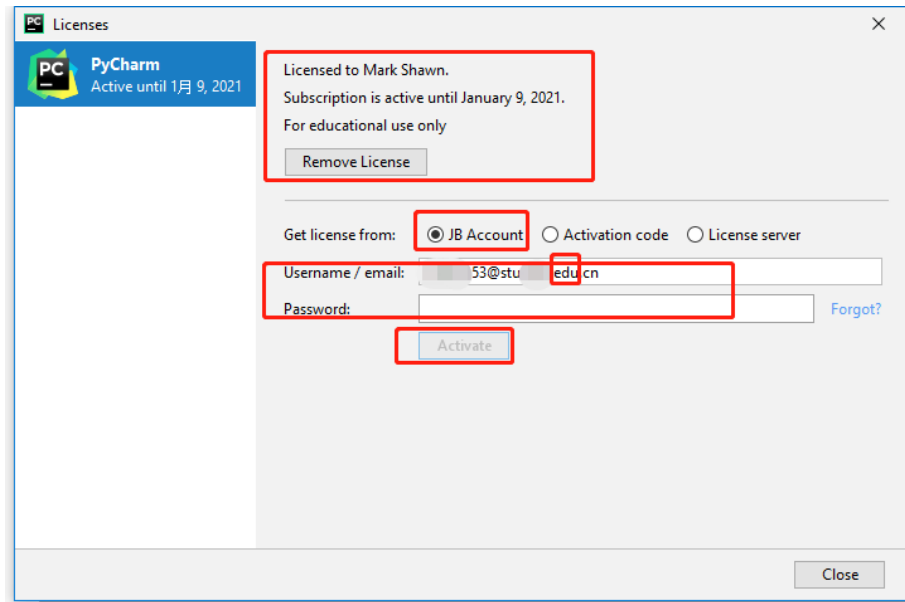
接下来，在安装PyCharm的时候要注意以下几项：

- 左1是创建64位的桌面运行快捷方式，建议勾选
- 右1是在系统/用户环境变量中添加PyCharm，一般来说当安装一些编程软件时，提示是否添加路径变量都是要勾选的，因为这可以方便程序（有时也是必须）找到可执行的文件，尤其是当装的软件并不多的时候也不大可能会出现变量冲突的问题，当然了如果不断地重装可能会引发一些问题
- 左2是在资源管理器中的文件夹上右击时显示一个“在Pycharm中打开文件夹”的选项，这是一个很有用的功能，当我们想深入研究一个Package的时候
- 左3是将Pycharm与.py类型的文件进行关联，不建议勾选，因为PyCharm的启动速度很慢，它更适合处理工程项目，而非单独查看某个.py的文件内容，我们更推荐使用SubLime、VSCode或IDLE等更轻量级的编辑器去打开.py文件



一路next后就会安装完成，之后会提示手动或自动重启（没有也没事，你可以自己重启，也可以不重启，但建议重启），重启之后PyCharm就正式安装好了。

接着就可以在PyCharm的启动界面内输入自己的注册信息，并得到一个License了，恭喜~！



友情提醒：

1. 如果不知道自己的教育网账号的可以询问一下身边的同学，或者老师。
2. 如果自己学校没有配给教育网账号的话，可以先使用free trial模式，可以坚持一个月，先凑合着吧（或者再过半个月就又有网上流出可以使用的激活码了）。

PyCharm安装补充解释：

在安装完PyCharm后重启打开PyCharm，选好配色等工作操作完就会出现一个软件授权界面，在这里，网上流传最多的一些激活码（非常长），以及让你修改host文件屏蔽JB（PyCharm厂商），但基本都已经没用了（至少2020年1月份这个节点我尝试了网上很多比如标题带有“亲测有用”的帖子都无一例外失效了，因为JB貌似加大了对软件盗版的打击力度）。

在此，我们并不推荐大家用激活码方式，由于本教程的受众大多数是学生，因此强烈推荐大家使用自己学校配给自己的教育网邮箱去官网注册账号，填写今年或者明年毕业的信息，然后邮箱确认后就可以得到一个一年有效期的专业版账号。在此，给JB的大方点赞，高等教育知识群体理应拥有使用先进生产力的权力，这也是JB奉行的理念之一。

安装Github Desktop:

下载地址: [GitHub Desktop | Simple collaboration from your desktop](#)



这个安装很简单，80兆，顺序安装即可。

Configure

我们推荐使用**Anaconda + PyCharm + Sublime**的组合学习Python，这能帮助你应对绝大多数情况下的开发问题，具有极强的扩展能力。当然，**Anaconda**下的**Spyder**可部分取代**PyCharm**，**VSCode**和**Sublime**也是互为替代品（VSCode比Sublime更强大一些），感兴趣的朋友可自行了解使用。

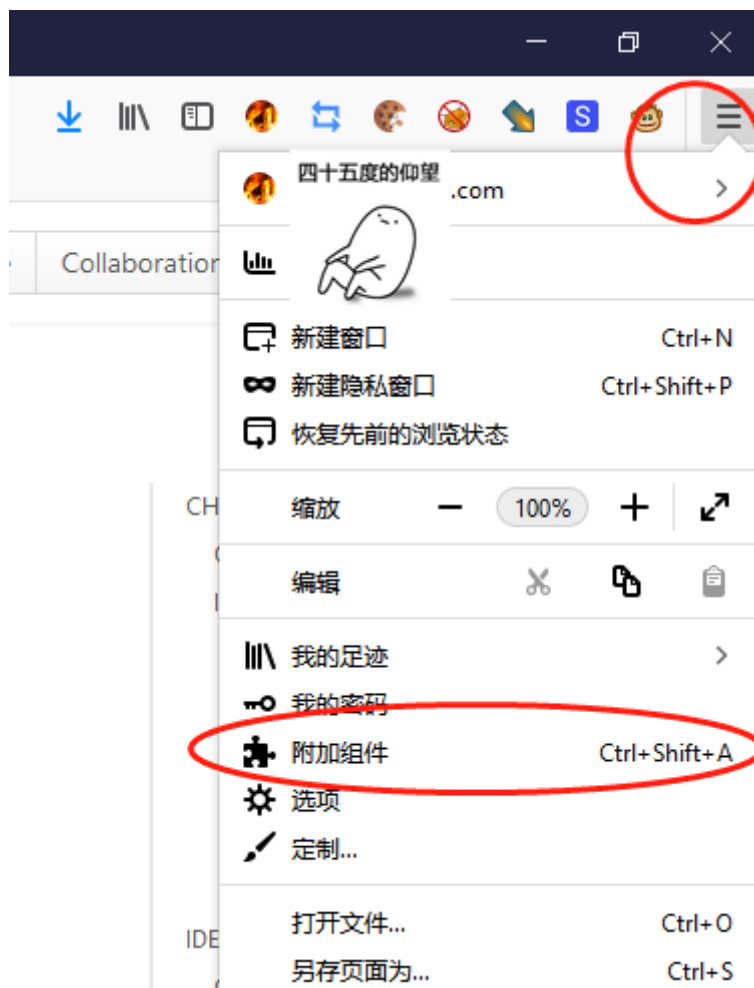
其中，Spyder和PyCharm都属于IDE（Integrated Development Environment，集成开发环境），即提供编写、编译、运行的一整套开发环境（一条龙服务）。而Sublime和VSCode都属于编辑器，就是提供很多扩展功能的比Windows的记事本更高级的记事本。

再说一说Anaconda，它是一个数据分析领域最流行的软件平台。当你安装完Anaconda之后，你的电脑便已经内置了一个Python环境，更准确地说是一个Python的安装包。它还内置了conda这个强大的包管理器，你可以启动Anaconda Navigator可视化地管理电脑中已经安装的python相关的安装包。有意思的是，严格意义上python是conda的一个子集，因为conda也管理着python这个安装包。

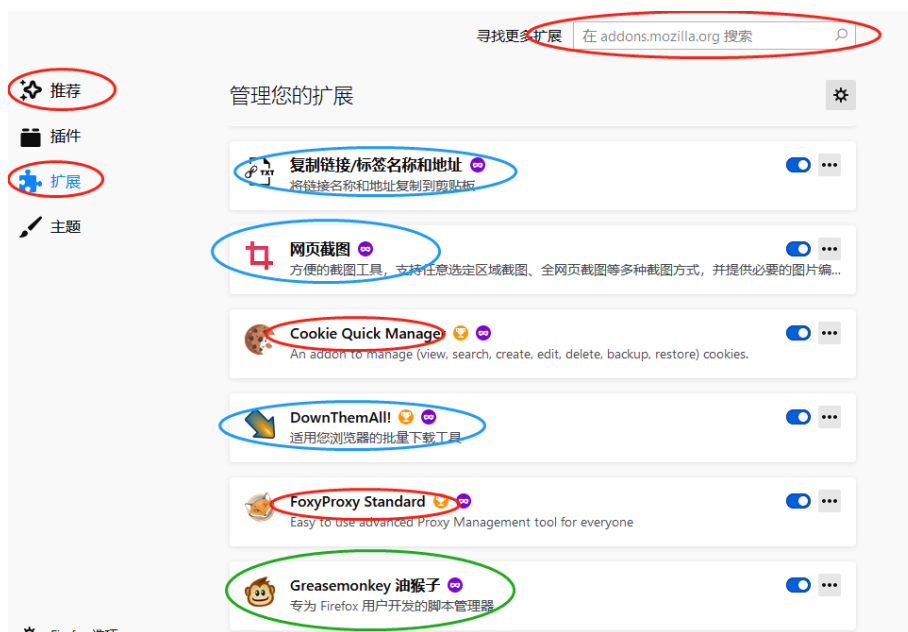
当然，最重要的是，Anaconda提供了Jupyter Notebook这个强大的交互式笔记本，和Spyder这样一个基本完备的IDE环境。尤其是Jupyter Notebook，这正是数以千万计的数据科学家无法抗拒的工作平台，也是PyCharm无法完全取代Anaconda的重要原因，以至于PyCharm内置了Jupyter Notebook的实现方式，你可以有趣地发现两个完全不同的公司开发的产品，它们之间相互影响、相互结合，共同为Python开发者们创造了一个几乎完美的开发环境。

接下来，为了让我们的软件更加强大，我们将为它们安装几个插件，并做一些配置。

火狐插件安装



在火狐浏览器的右上角的菜单按钮处点击展开，选择附加组件进入。

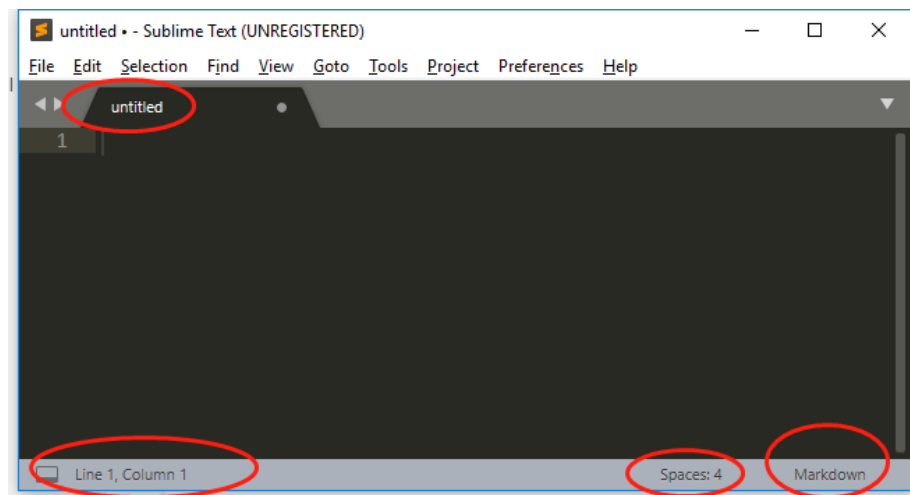


接着，你就来到了一个免费的、工具众多的插件市场，这里，我不会向大家强力推荐什么插件，毕竟每个插件都有其各自的功能，而且还存在替代品，我并不能保证我推给大家的就是最好的插件，一般对于这些够用就行。

大家打开后界面内存在的插件是和我不一样的，但可以通过推荐或者右上角的搜索栏搜索自己想要的插件。上面这张图中有6个插件，其中红色的是和爬虫相关的，一个管理着Cookie，一个管理着代理，非常重要，尤其是对于一些反爬的网站。另外三个蓝色的就是提供一些制作功能的，其中我最欣赏复制链接/标签名

称和地址插件，因为我要大量复制一些网页的地址与标签并保存成Markdown格式以便我的读者与我自己有更好的体验。而DownThemAll!是个有意思的插件，它可以罗列出网页上所有可下载的资源，包括各种文档，可以按类型筛选，有时候会有奇效。不过我知道大家有些人对视频更感兴趣，你也可以自己搜一搜一些视频插件，有些比如能破解会员视频，还是很吸引人的。最后绿色的那个插件油猴是插件中的插件平台，你可以通过它再开发小插件放在上面运行，非常强大。

Sublime Text 3使用介绍与插件安装



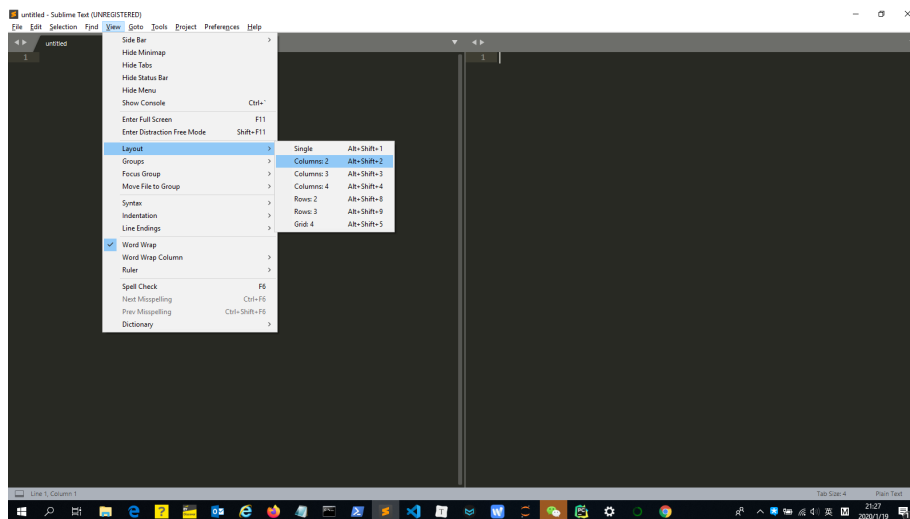
Sublime打开是一片黑白的，而且为你自动创建了一个untitled文件，但Sublime有一个很有意思的点，无论你有没有输入，你直接叉掉它它就直接关掉了，然后文件呢？自动保存了！等你下次打开，它们都在。这个特性是非常方便的，比如我有时需要速记点东西，于是我会打开windows的记事本，输入后再叉掉，可是它会弹窗问我保不保存，还要花我2秒钟分清一下哪个是不保存键，就这体验我再也很少用记事本了，即使速记它都不配，因为Sublime打开也非常快。

此外，看一下状态栏的一些信息，左下角是多少行多少列，这里提一下，编程世界里，经常会接触行和列，因为什么？因为数据结构要么是一维数组，要么是二维表格，要么是树状结构，要么是网状结构，而二维又是非常重要的，尤其对于财经领域的从业人员来说。记住行可以叫line，也可以叫row；列一般叫col或者column。当然行的序号也可以叫index，或者seq（即sequence）等。

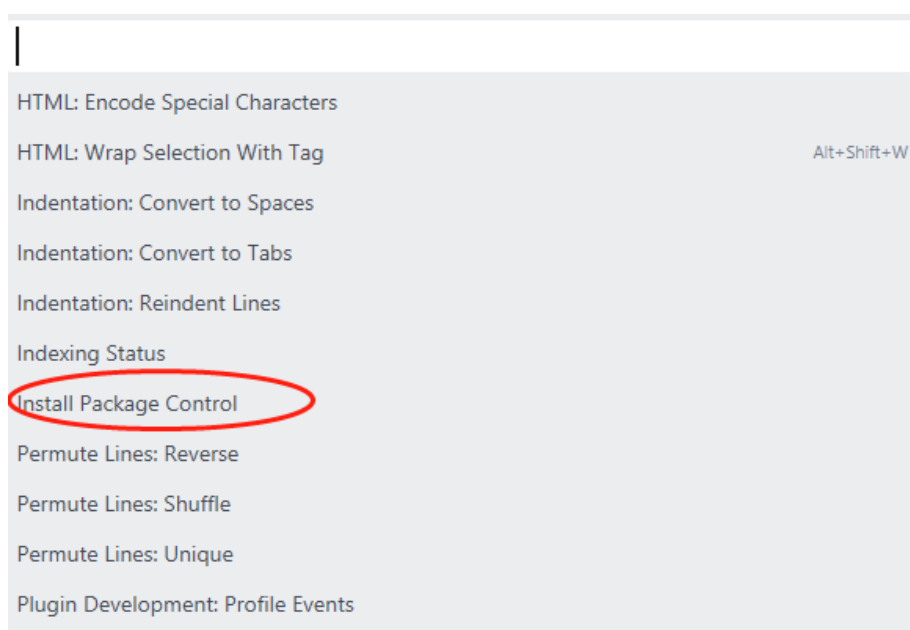
然后右下角的第一个关键信息是Space: 4，这个不学编程的人不会留意，但学Python的人要尤其注意，因为Python是靠缩进来表示语句结构的。关于缩进，因为有Tab和Space两种，它们看起来都能实现相同的效果，但其实原理完全不一样，具体可以见之后的引申阅读。

接着，很重要的信息在右下角，就是文件的格式，图中是Markdown，因为我修改了默认设置，这样我每次新建一个Sublime文档，都默认是Markdown格式的。你也修改它，单击之后将会打开新世界，其中你将会接着、打交道的有HTML | CSS | JavaScript | JSON | Python | SQL | XML等，尤其是JSON，我们会和它们经常打交道的。

紧接着，我们再看一个很重要的功能，那就是Layout，我们可以很轻松地把一个界面分成两块，不但可以对比不同的文件，还可以对比同一个文件的不同部分，不过要说起对比这种程序员刚需，就不得不推一些对比插件了。



按住 **Ctrl + Shift + P** 打开 **Sublime** 的快捷窗口，然后找到 **Install Package Control**，如果出错请参考这篇文章：[Installation - Package Control](#)

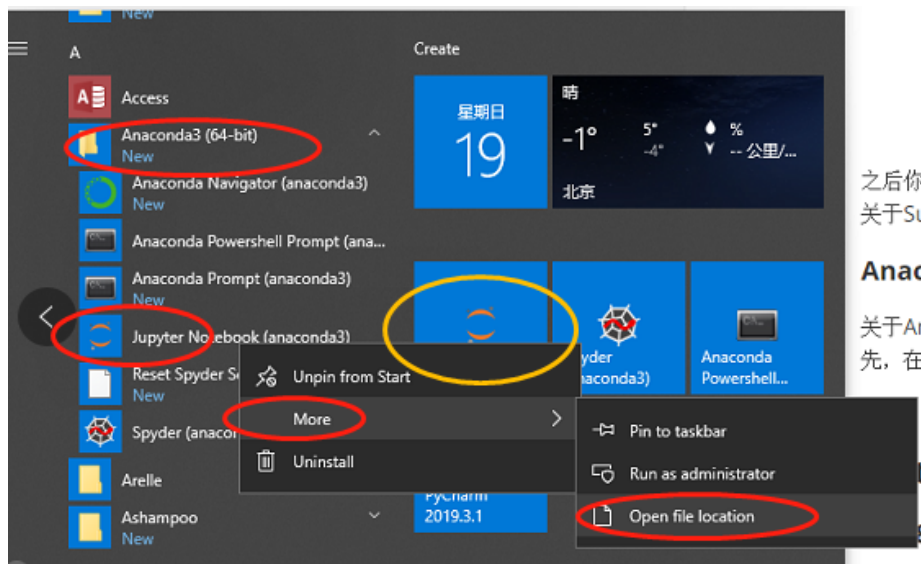


之后你就可以安装很多有用的插件，比如我对Markdown的书写体验很重视，就会下一些Markdown的增强插件。关于Sublime的插件，可以参考这篇文章：[sublime text 3 插件推荐？ - 知乎](#)

Anaconda配置与插件安装

修改Jupyter Notebook启动位置与默认浏览器

关于Anaconda，由于组件众多，我实在不能给出一个服众的方案，我只说一下我是怎么用Anaconda的吧，首先，在开始菜单中右击Jupyter Notebook，并选择打开文件位置，跳转到其快捷方式的位置。

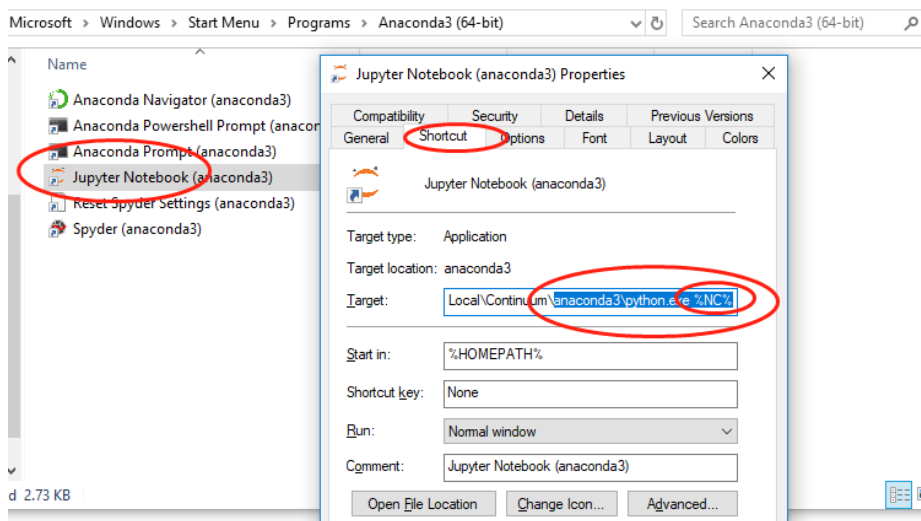


之后你
关于St

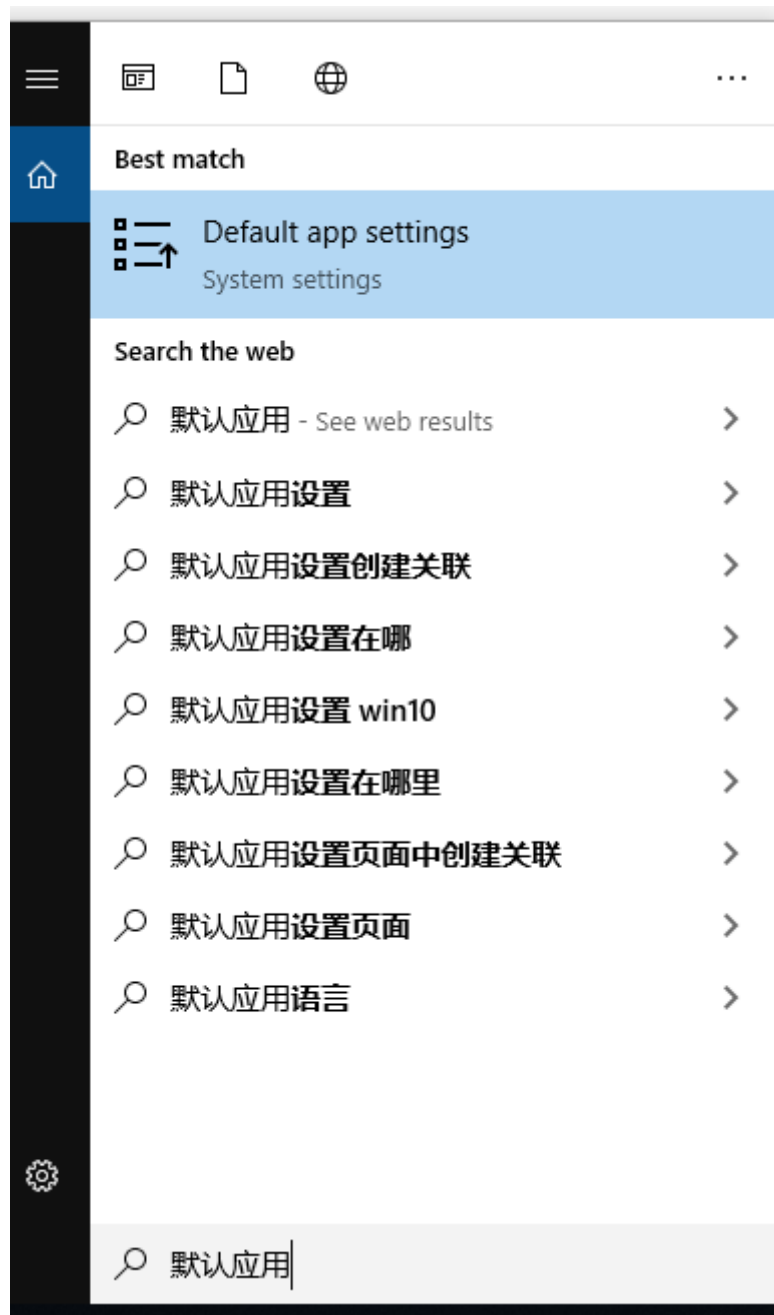
Anac

关于Ar
先, 在

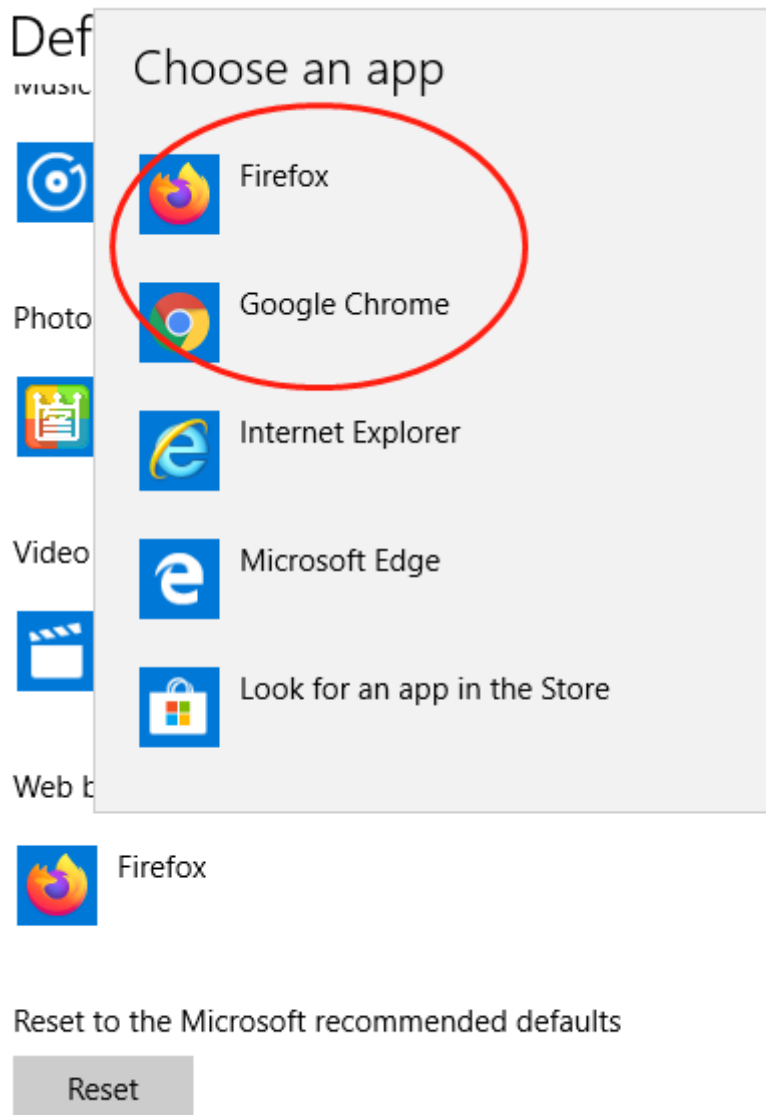
接着，右击其快捷方式，在属性里面设置它的目标位置，把`...anaconda3\python.exe`后面的文字都删掉，改成你想作为你的编程专用文件夹位置，我直接输入了`%NC%`，因为我在《南川P.F.A教程》第一章中已经在环境变量中定义了`NC`这个变量的含义，即我的`C:\\NC_WORKSPACE`文件夹路径，作为我的编程工作起始文件夹。



之后我还会在它运行之后，将其固定在我的任务栏处，这样我就能随时随地打开它。注意，打开一个Jupyter Notebook，它将首先开启一个黑窗，里面显示着运行过程中的一条条命令。这个目前不用管，严格意义上说，只要程序不出错，永远都可以不用管。不过要注意的是，在这个黑窗之后，你将能看到你的默认浏览器被打开了，因为Jupyter Notebook是基于网页的，在此建议将默认浏览器转为火狐或者谷歌浏览器。不同浏览器，尤其在之后的爬虫区别还是挺大的。



修改默认应用的快捷方法如上，直接在搜索框中输入“默认应用”即可，选择第一个匹配结果。接着再替换默认浏览器为二者之一。



修改智能提示：取消使用jedi

接下来我们在cmd中测试一下ipython，输完print后按一下tab键，看看发生什么。

```
C:\NC_WORKSPACE>ipython
Python 3.7.4 (default, Aug, 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.8.0 -- An enhanced Interactive Python. Type '?' for help.

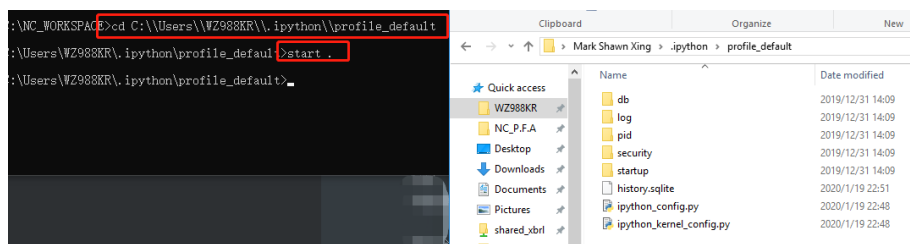
In [1]: print(
abs()      ascii      bin
all        AssertionError Blocking
any        AttributeError  bool
ArithmeticError BaseException breakpoi
```

可以看到这些提示很没啥用，现在看不懂没关系，会改就可以。

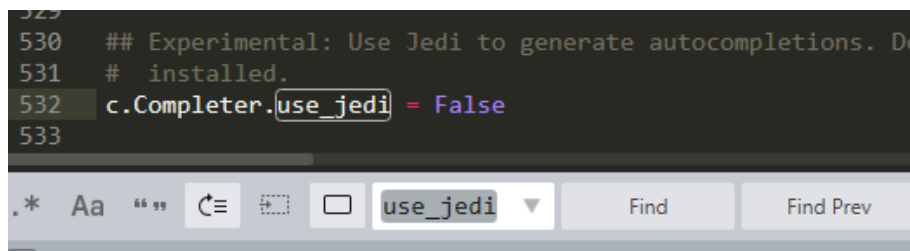
于是，这样操作，按ESC或Ctrl + D退出，然后重新输入ipython profile create，它就会在你的用户文件夹下生成两份配置文件。复制它的文件夹（非文件）路径到资源管理器中打开，

```
C:\NC_WORKSPACE>ipython profile create
[ProfileCreate] generating default config file: 'C:\Users\WZ983KR\ipython\profile_default\ipython_config.py'
[ProfileCreate] Generating default config file: 'C:\Users\WZ983KR\ipython\profile_default\ipython_kernel_config.py'
```

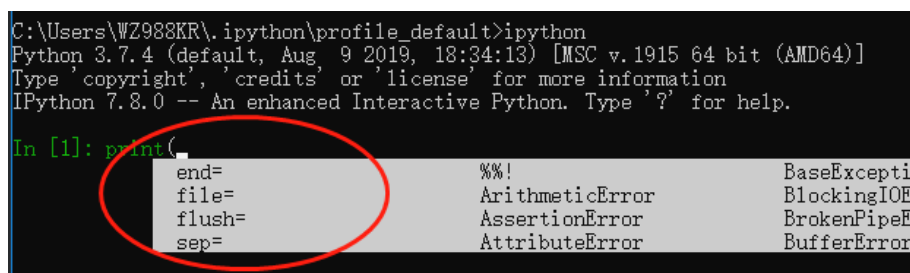
或者使用 `cd` 文件夹路径的方式切换到目标文件夹内，然后 `start` 打开。



接着，就是Sublime Text 3发威的时候了，右键 `ipython_config.py` 文件，然后选择使用Sublime打开（打开是不是特别流畅，并且右下角自动显示Python，接着 `Ctrl + F` 搜索 `use_jedi`，定位到第532行附近，把它开头的 `#` 号去掉，并修改成 `False`。然后按 `Ctrl + S` 保存，随着标题栏的星号一扫而过，文件就修改好了。



接下来重新进入cmd，重新输入 `ipython`，重新输入 `print()`，并且按 `tab` 键，看看是否不一样了？



没错，它把属于 `print` 函数的参数提到前面来了，这正是我们想要的！

By the way，说一下，只有 `ipython` 以及基于 `ipython` 的 `jupyter notebook` 需要这么修改，`Anaconda` 里的另一个开发环境 `Spyder` 是不需要的，`PyCharm` 自己也是不需要的，并且 `PyCharm` 在输入函数之后是按 `Ctrl + P` 键才是跳出参数，而非 `Tab` 键。

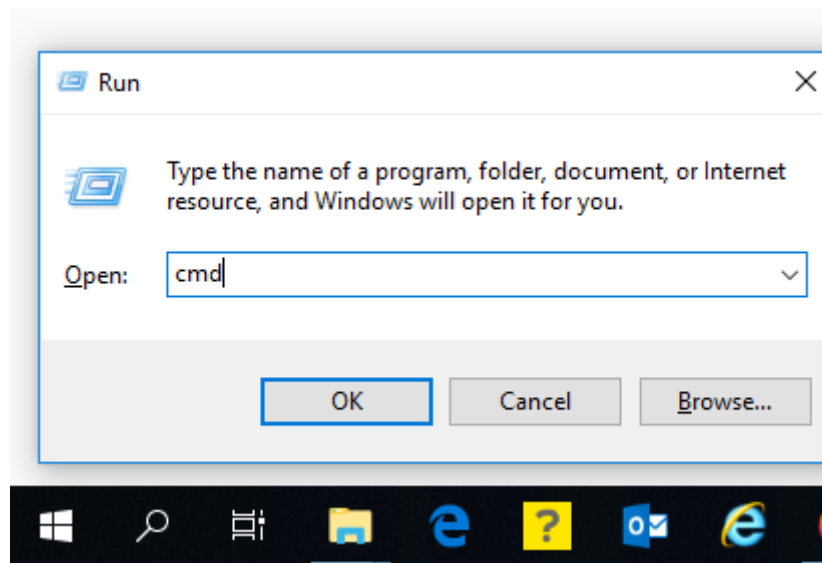
强烈建议研究一下 `Jupyter Notebook` 的快捷键，在 `help` 菜单里

此外，大家可以研究研究 `Jupyter notenbook` 的快捷键，很有意思！

pip更换镜像源

当然了，`Anaconda` 更加重要的操作是更换镜像源，所谓镜像源就是一些应用程序所能获取的位置，`Anaconda` 本身就有有一个云存储着大量的安装包资源，（`pip` 也是）但由于是在海外，因此直接从这个位置下载这些安装包，将会非常地慢。这个时候，就要将我们的默认镜像源（准确地说是更优先的镜像源）地址更改为国内的清华或者中科大站。这里，我们给出清华镜像源设置方法。

首先给出pip更换清华镜像源的方法，由于以后大多数包我们都是通过pip install xxx的形式安装的，所以首先要将pip更换成国内的。方法如下，Windows + R打开左下角的run窗口，输入cmd打开一个黑色的命令行窗口



接着复制输入如下内容：

```
pip install pip -U
pip config set global.index-url
https://pypi.tuna.tsinghua.edu.cn/simple
```

第一句是升级pip，有些人安装的位置可能有问题，不放心的话可以在后面加上--user关键字：

第二句话是将清华的这个镜像源设置为全局的默认的pip源地址。

这里特别指出，如果你是在公司内网（比如我现在的的环境的话），那就有可能连不上pip，所以就要先设置代理。我示范一下我在公司内网升级pip并且配置清华镜像源的操作：

```
cmd Command Prompt
Microsoft Windows [Version 10.0.17134.1184]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\NC_WORKSPACE>set http_proxy=http://[redacted].net:80
C:\NC_WORKSPACE>set https_proxy=https://[redacted].net:80

C:\NC_WORKSPACE>pip install pip -U
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting pip
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/00/b6/9cfa56b4081ad13874b0c6f96af8ce16cfec1/pip-19.3.1-py2.py3-none-any.whl (1.4MB)
    |#####| 1.4MB 2.2MB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
  Uninstalling pip-19.2.3:
    Successfully uninstalled pip-19.2.3
ERROR: Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: 'C:\Users\wz983kr\AppData\Local\Temp\pip-uninstall-r2y9hkxd\pip.exe'
Consider using the --user option or check the permissions.

C:\NC_WORKSPACE>pip install pip -U --user
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already up-to-date: pip in c:\users\wz983kr\anaconda3\lib\site-packages (19.3.1)

C:\NC_WORKSPACE>pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
Writing to C:\Users\wz983kr\AppData\Roaming\pip\pip.ini

C:\NC_WORKSPACE>
```

可以看到，需要首先设置 `http_proxy` 和 `https_proxy` 的代理，然后再 `pip` 升级，`-u` 的意思就是 `--update` 升级的意思，但有经验的人肯定知道我其实升级失败了，并且成功的把 `pip` 给卸载掉了，理论上我其实要重装 `python` 了，但为啥下一句加了 `--user` 就又好了呢？其实没好，你怕自己之前安装的时候可能不规范导致升级没成反而把 `pip` 卸载掉了，可以直接加 `--user`。

而我这里要解释的是，因为我的计算机里有多多个 `pip`，另外一个（也是我主要用的）`Anaconda3` 下的 `pip` 更新它自己成功了（毕竟它自己就是最新版），明白了叭。如果对这里不懂，自己也没有安装出问题，可以略过。如果自己出问题了，建议去看一下第一章的内容，那里写的很详细。

接下来 `Anaconda` 的镜像源更换，我就不展开了，`pip` 这个其实才是最关键的，有兴趣地可以参考以下几个链接：

- [Pypi | 镜像站使用帮助 | 清华大学开源软件镜像站 | Tsinghua Open Source Mirror](#)
- [Anaconda | 镜像站使用帮助 | 清华大学开源软件镜像站 | Tsinghua Open Source Mirror](#)

推荐阅读的一些关于 **Jupyter Notebook** 的资料

最后由于篇幅限制，给出安装 `Jupyter Notebook nbExtension` 的方法，我就不罗嗦了，非常好用，进阶必用。

- [Installing jupyter_contrib_nbextensions — jupyter_contrib_nbextensions 0.5.0 documentation](#)
- [Jupyter-contrib/jupyter_nbextensions_configurator: A jupyter notebook serverextension providing config interfaces for nbextensions.](#)

以及一些主流的、经典的、讨论已久的组装使用 `Jupyter Notebook` 的技巧：

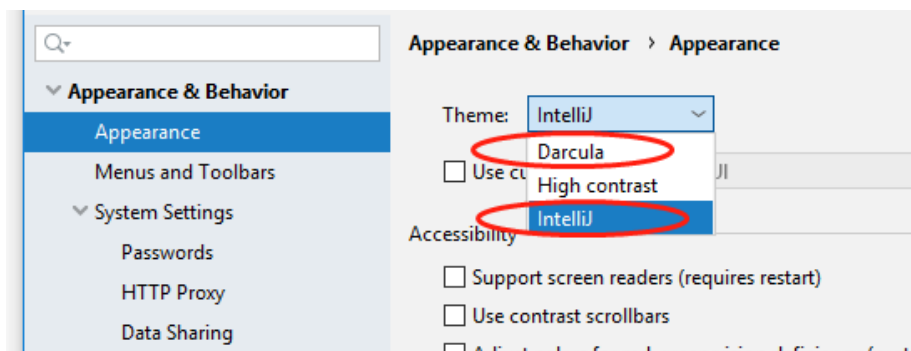
- 英文版：[28 Jupyter Notebook Tips, Tricks, and Shortcuts for Data Science](#)
- 中文版：[【精心解读】关于Jupyter Notebook的28个技巧 - 知乎](#)
- 中文+代码版：[Python编程神器Jupyter Notebook使用的28个秘诀（附代码） - 云+社区 - 腾讯云](#)

PyCharm的配置与插件安装

最后再谈谈 `PyCharm` 的配置和插件安装。

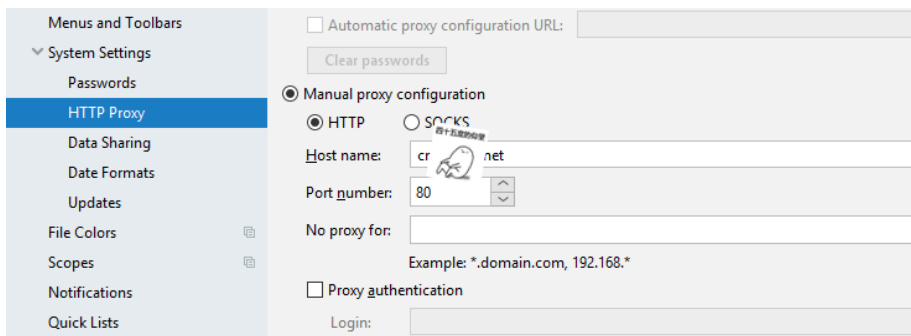
关于UI

首先，`PyCharm` 是完完全全的开发工具，因此主要集中在程序的逻辑、组织上，一个好的、大的项目，离不开 `PyCharm` 的全程支持，所以会长时间占用你的眼睛，因此 UI 很重要，在 `File - Settings - Appearance` 下可以修改自己的 theme，白天调成 `IntelliJ`，晚上调成 `Darcula`，还是非常有必要的。



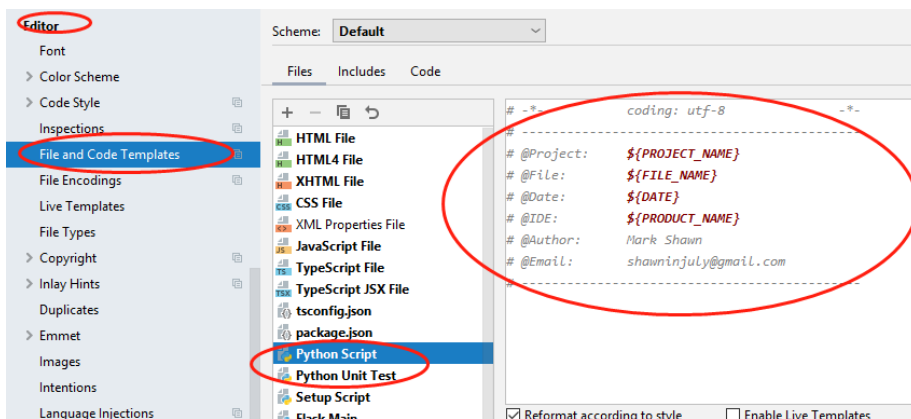
关于代理

接着，如果你在公司内网，就需要再配一下代理了。



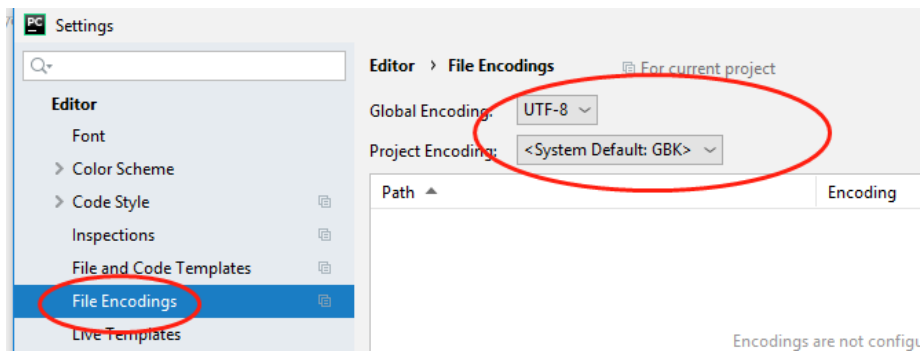
关于预设脚本模板

紧接着，在编辑器 - 文件与代码模板 - Python 脚本 里可以写入这样的文字，这样你每次新建一个Python文件后它都会自动生成一段预设的自动获取的文件名、时间、作者等信息，这可不止是要酷，也是专业的一种体现。



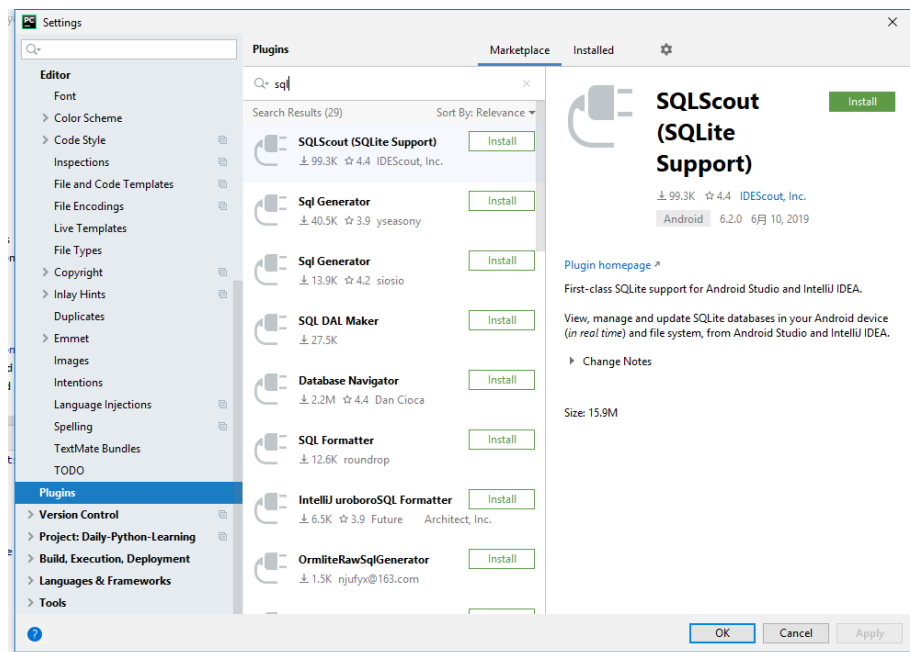
关于编码

在编码这里也要特别注意，可以看到全局设置的是UTF-8语言，然后项目编码是系统默认的GBK中文，可以暂时不用改，当编码有问题的时候可以来看看这边，具体想了解的可以参考一些博客文章。



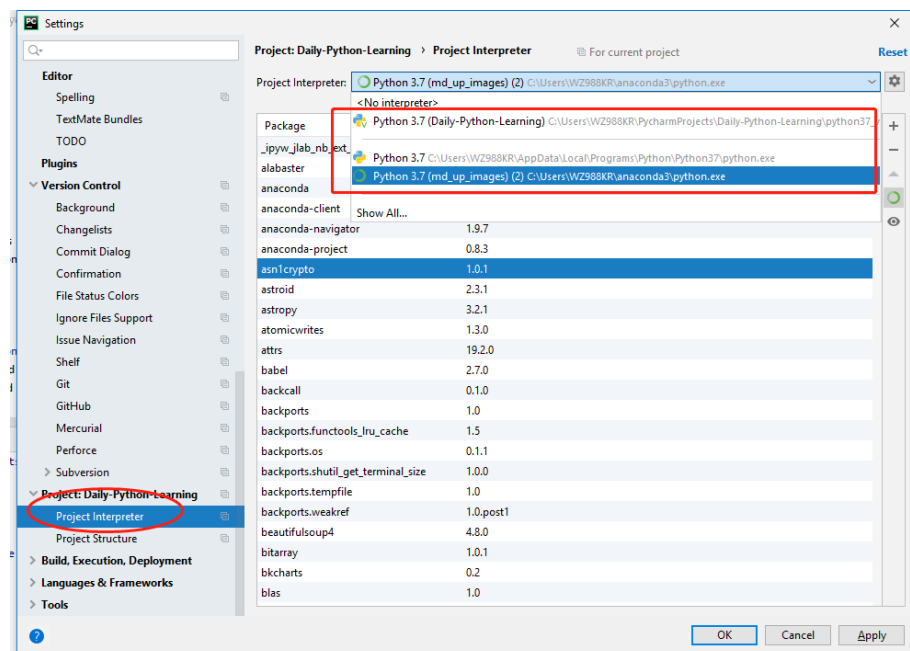
关于插件

接下来一个非常重要的菜单项是 **Plugin**，你可以找到很多有用的插件，比如专门用于显示MongoDB数据库的 **Mongo Plugin** 等。



关于程序编译器：最重要！

接下来的这个就更加关键了，它叫程序编译器，只有正确配置了相关Python程序的位置，你的程序才可以得以运行，比如下图：

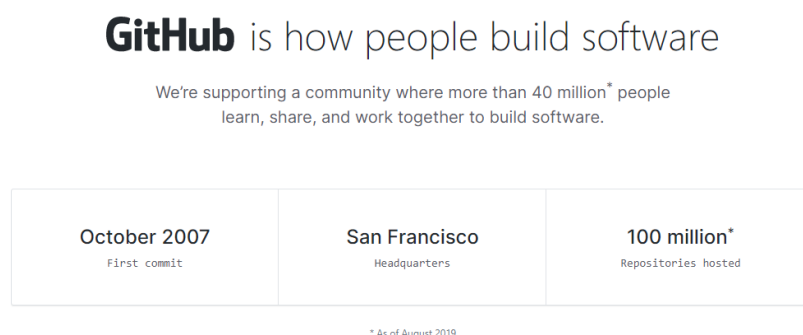


该图显示，我一共有3个不同的版本，其中红框内第二个是原生的python环境，里面只有两个包（猜猜是哪两个），而第一个python是我为开发另一个项目而基于原生python环境单独配置的，第三个大家都知道，它就是Anaconda提供给我们的包，如果你要测试、尝试开发个小东西，那就选择包最多的那个环境就好了，尤其选择Anaconda最舒服，当然了加载也会很慢。但如果你像我一样要发布一个项目，那就要考虑重新建一个效率更高、独立的环境了，这就是唯一的区别。充其量Anaconda也就是提供的包更多一些，管理起来更方便一些（有Anaconda Navigator管理），但在实际运行的时候调用的都是python的包。

其实大家有没有意识到，在这里，Anaconda和PyCharm两个巨无霸竟然合在一起了！没错，就是这种感觉，我们要的就是强强联合、双剑合璧，这样才能发挥出最大的作战威力。

Github Desktop

GitHub 是目前全球最大的开源代码库，以下是官网的截图：



可以说，学习编程的第一步和最后一步，都是Github，因为它是连接程序员的桥梁，它是如此重要，以至于你必须把它在浏览器里加个显眼的书签以表明自己无时无刻不在关注程序员世界的最新动态，某种意义上，这也是世界最先进生产力的窗口之一（如果不去计较那些军方的、大学的实验室的话）。

但Github的桌面程序其实没啥好说的，而且新手也看不懂，只要会用它Clone就可以了。

其实比起谈论Github Desktop，不如谈论PyCharm中的Git控制，这个也算进阶内容了，以后有机会吧。

Practical Problem: Tab or Space

我们以常规的一个**Tab**设置为4个缩进来说一下二者的区别。

```
# 下面是一个英文字符串，它每隔3个英文输入一个空格，于是有
abc def ghi klm nop
abc de f ghij k
# 上面也是一个英文字符串，由于每4个空格为1个Tab，你可以视为每个4个空
格为1个单元连载一起。
# 所以先输入abc之后，占了3个格子，那么再输入tab就会自动跳到下一个单
元，即第5个格子。
# 接着再输入de，占了2个格子，还剩2个格子，按tab也会跳到下一个单元的
起始位置，也就是第9个格子。
# 接着再输入f，占了1个格子，还剩3个格子，按tab依旧会跳到下一个单元的
起始位置，也即是第13个格子。
# 最后输入ghij，占满4个格子，按tab键就会跳到下下个单元的起始位置，也
就是第21个格子去了。
```

好的，我们来看看这个tab计数用Python怎么编，小小的感受一下。这题很简单，首先我们定义一下输入和输出，我们设置两个变量，一个是N即tab的位宽，一个k即当前字母的位置，再定义函数f(k, N)返回一个整数值，为了让程序更清晰明了，我们要放弃数学中f就是一切的思想，转而有意义的文字去替换它，比如写成这样：get_next_input_pos(k, N)。好了，其实这就写出了一个挺直观的函数了，并且是编程界的函数，而非数学界的函数。

基于此，你应该很快地反应过来，k和N可以也写得更具体、更有意义一些，没错！我们再修改一下，get_next_input_pos(cur_pos, TAB_WIDTH)，其中cur就是current地缩写，pos就是position的缩写，像这样的代码是极其可读的，你拿给任何程序员，它都是看得懂的，无非不能确定的是，你的pos是从0开始计还是从1开始计的。

那输入怎么输呢？输出又怎么输呢？函数该怎么定义？我们给一下模板，非常简单：

```
def get_next_input_pos(cur_pos, TAB_WIDTH):
    """
    用三个双引号写在函数下面，表示注释，而且这里的注释是对于整个函数的
    说明，并且非常重要，因为当你进阶后可能会涉及到编写程序文档，而这里的
    说明可以自动变成文档的一部分。

    在这一组三引号内你可以输入任何正常文字，我在下面给一个比较标准的
    注释，供大家参考。

    -----
    -----

    得到输入一个TAB键后，光标所在的输入位置

    :param cur_pos:      输入时，当前已经输入的英文字符数，由于中
    文字符是按2-3个计的，因此此处只考虑英文

    :param TAB_WIDTH:    每个TAB表示的固有的宽度，一般为4

    :return: int         应该返回一个整数值
```

```

"""

if cur_pos % TAB_WIDTH == 0:
    # A%B 表示模运算, A//B 表示整除运算
    # 当前正好处于位宽的结尾处, 则下一个TAB键后, 光标应该跳到下
    下个单元的开头
    print("Next pos will be: ", cur_pos + TAB_WIDTH +
1)
else:
    # 否则, 下个TAB键后, 光标应该跳到下个单元的开头
    ??? # 这里留给大家自己思考, 给我答案

# 以下是三个测试用例, 这块代码补完之后要能直接运行。
get_next_input_pos(4, 4)
get_next_input_pos(9, 4)
get_next_input_pos(5, 2)

```

Reference

参考网址

- [Installing on macOS — Anaconda documentation](#)
- [Jupyter-contrib/jupyter_nbextensions_configurator: A jupyter notebook serverextension providing config interfaces for nbextensions.](#)
- [28 Jupyter Notebook Tips, Tricks, and Shortcuts for Data Science](#)
- [【精心解读】关于Jupyter Notebook的28个技巧 - 知乎](#)
- [Python编程神器Jupyter Notebook使用的28个秘诀（附代码） - 云+社区 - 腾讯云](#)

Documentation

- [Getting started with Anaconda — Anaconda documentation](#)
- [User guide — conda 4.8.1.post77+375945f4 documentation](#)
- [The Jupyter Notebook — Jupyter Notebook 6.0.2 documentation](#)