

# Anaconda 下载安装及 Spyder 使用

唐银银

2018 年 12 月 3 日

一、下载安装 Anaconda（勾选 PATH） .....	2
二、Spyder 基本操作 .....	4
1、菜单栏 .....	4
2、全局设置 .....	5
3、Spyder 各个窗口操作 .....	8
4、安装包 .....	11
5、Python 基础语法 .....	11
三、Jupyter Notebook 基本介绍 .....	15
1、简介 .....	15
2、使用 .....	15
2.1、打开 Jupyter .....	15
2.2、更改默认路径 .....	15
2.3、新建 notebook .....	17
2.4、notebook 页面介绍 .....	17
2.5、有用的快捷键 .....	18
2.6、加载或运行本地 Python 文件 .....	19
2.7、隐藏 notebook 的输入单元格 .....	20
2.8、插入图片音乐等 .....	20
2.9、Markdown 简介 .....	21
2.10、关闭 jupyter .....	22
三、其他 IDE .....	23
四、自己动手做一做 .....	23

## 一、下载安装 Anaconda（勾选 PATH）

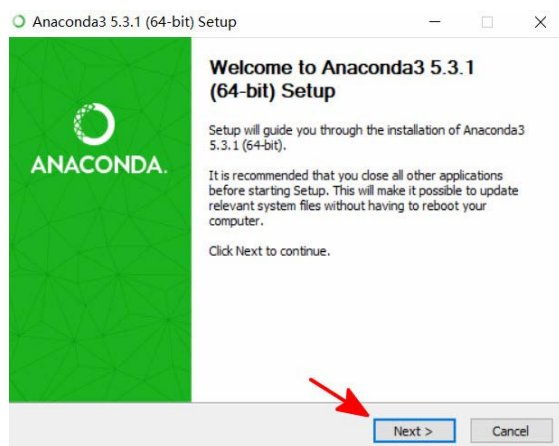
Anaconda 是专注于数据分析的 Python 发行版本，包含了 conda、Python 等 190 多个科学包及其依赖项。使用 Anaconda 的好处在于可以省去很多配置环境的步骤，省时省心又便于分析。

下载地址：<https://www.anaconda.com/download>，如果国外的网站下载不下来，可以用清华镜像下载：<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive>

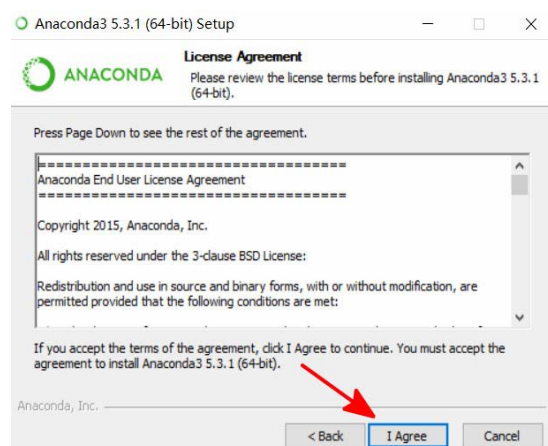
下载对应操作系统的 Anaconda 安装包，请注意安装最新版本的 Anaconda3-5.3.1，由于 Anaconda 自带 python，因此如果之前电脑里安装过 python 一定要先卸载原有的 python 再安装 Anaconda!!! 安装在两个 python 下的库不能共用，python 的运行环境会混乱。保证电脑里只有一个 python 可以免去很多麻烦。

下载好之后开始安装，整个安装过程耗时二十分钟左右（试验机型：DELL inspiron 5000，操作系统：windows 10），安装期间可能会弹出命令行窗口，不用对命令行进行操作，安装步骤如下：

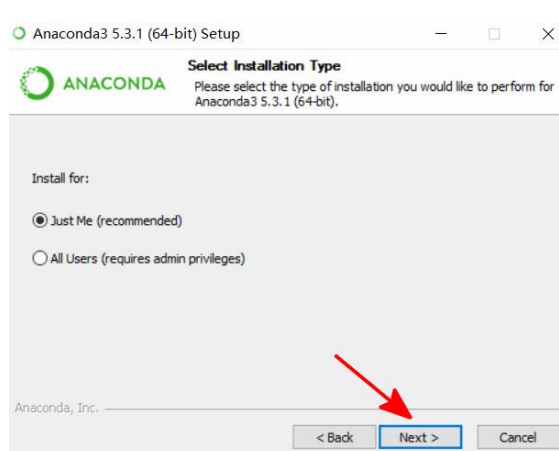
(1) 点击“next”



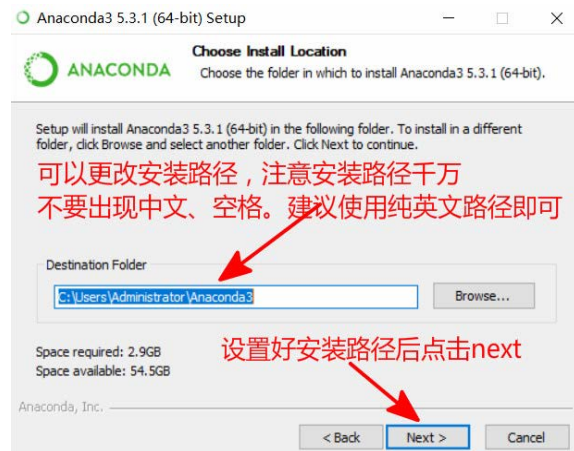
(2) 点击“I Agree”



(3) 点击“next”

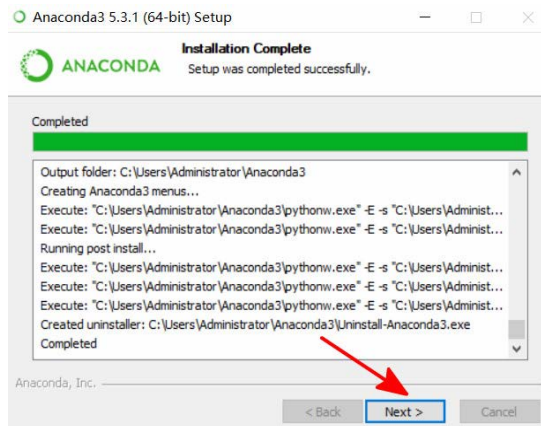
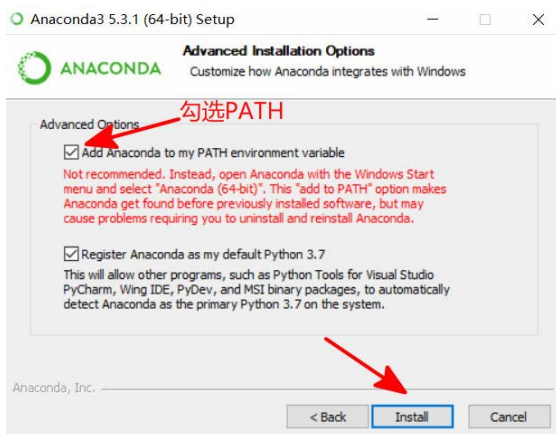


(4) 设置安装路径（注意图中红字）



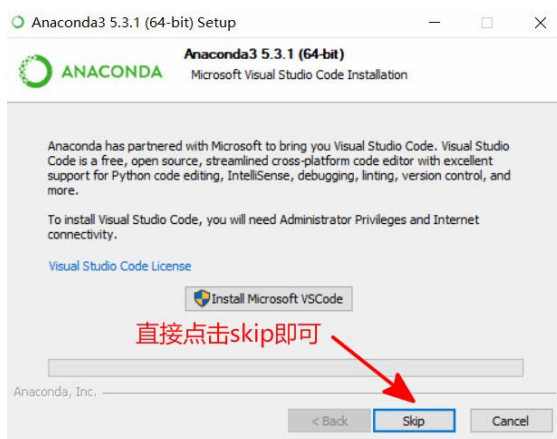
(5) 勾选“PATH”

(6) “Installation Complete”后点击“next”

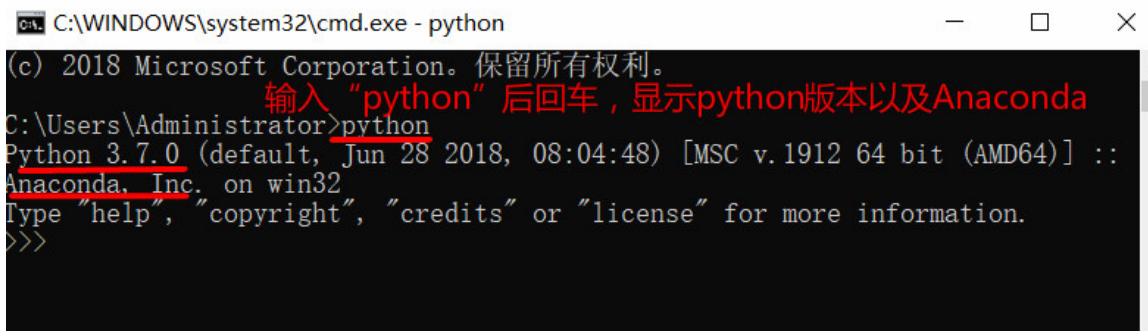


(7) 无需安装 VSCode, 点击“skip”

(8) 安装完成

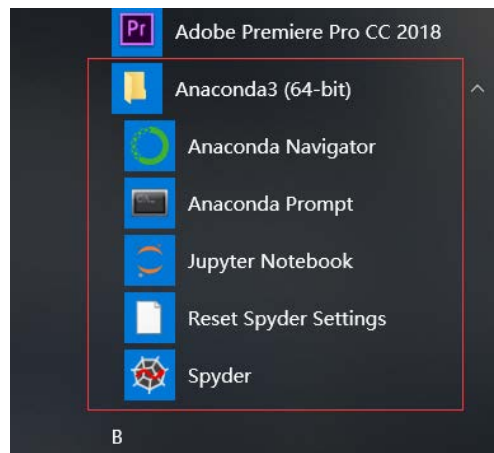


(9) 检查安装完成: 打开命令行窗口(快捷键 Win+R->输入“cmd”->回车)  
输入“python”后回车, 从命令行进入 Python, 出现 python 版本及 Anaconda 表示安装成功!



安装结束后会发现多了很多应用：Anaconda Navigator、Anaconda Prompt、Jupyter Notebook、Spyder。

在“开始”菜单栏中找到 Anaconda 文件夹，可以找到这些应用的快捷方式，打开这些应用。



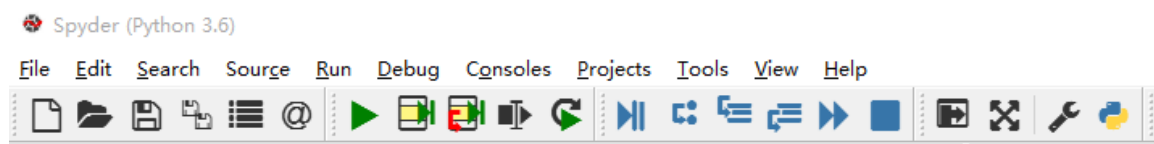
实在找不到可以通过在电脑里搜寻 Anaconda Navigator.lnk, Spyder.lnk 等，即快捷方式文件，然后双击打开（强烈推荐软件 [Everything](#)，找东西非常快）可以参考我的路径 C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Anaconda3 (64-bit)\，切记不要删掉这个文件夹，否则只能用下面提到的最后一种办法了。

最后如果还是找不到可以直接搜索 Spyder.exe 文件或者直接打开命令行输入软件名来打开（比如“Spyder”，“Jupyter Notebook”），不过在使用期间不要关闭 exe 或 cmd 窗口，不推荐这种方式，比较麻烦。

## 二、Spyder 基本操作

Spyder 是 Python 的集成开发环境。和其他的 Python 开发环境相比，它最大的优点就是模仿 MATLAB 的“工作空间”的功能，可以很方便地观察和修改数组的值。

### 1、菜单栏



- File: 新建、打开、打开最近文件、保存、关闭等
- Edit: 撤销、重做、复制、剪切、粘贴等
- View: Window layouts 可选择显示的界面类型，spyder、Rstudio、Matlab 等

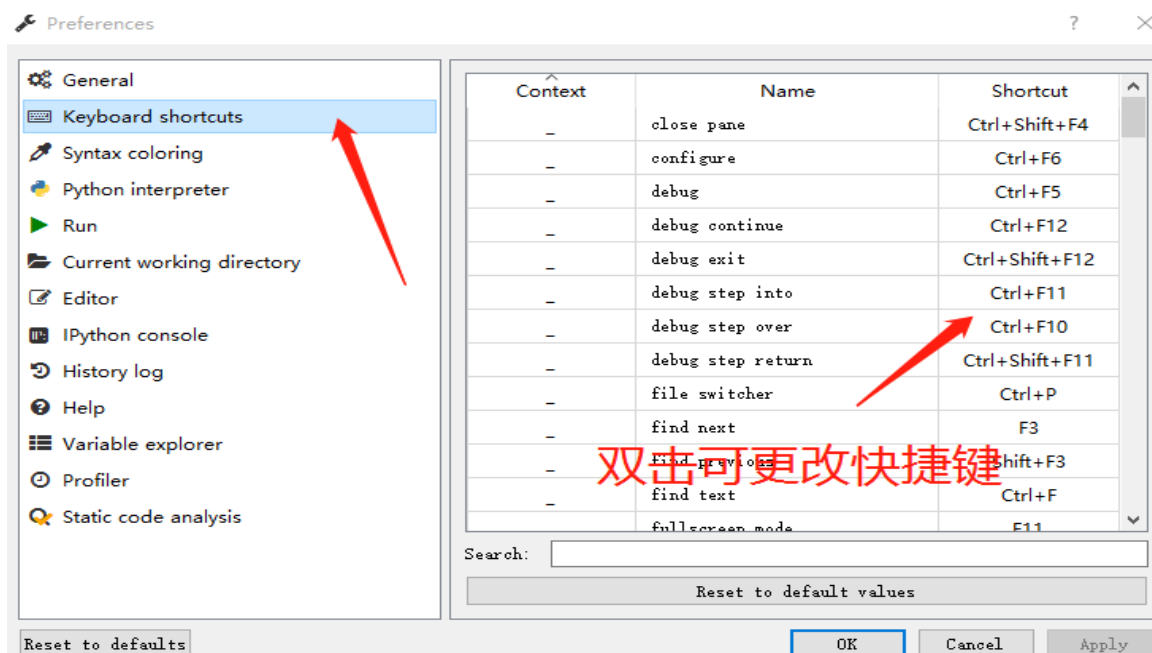
- Tools: 各种设置

常见操作的快捷键：新建、打开、保存、运行、最大化当前窗口、全屏、显示当前工作路径、修改当前工作路径（非默认更改）。

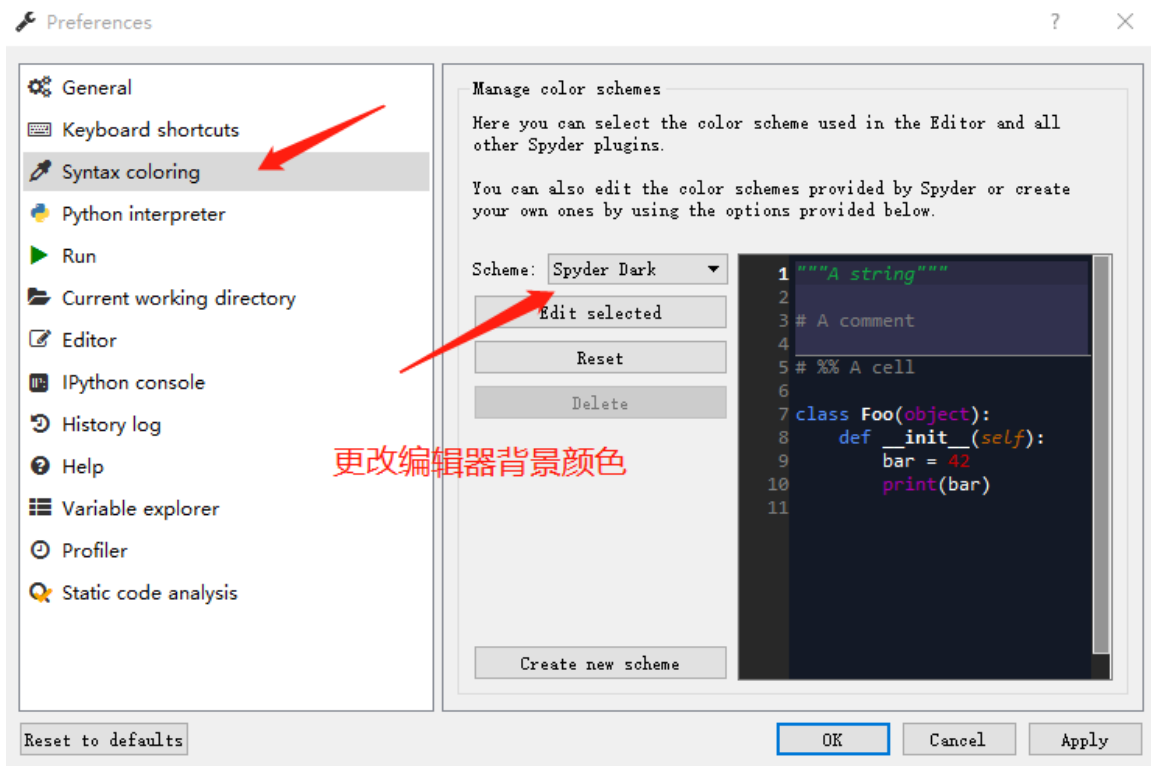
## 2、全局设置

Tools->preference（有兴趣可自行研究）

快捷键：

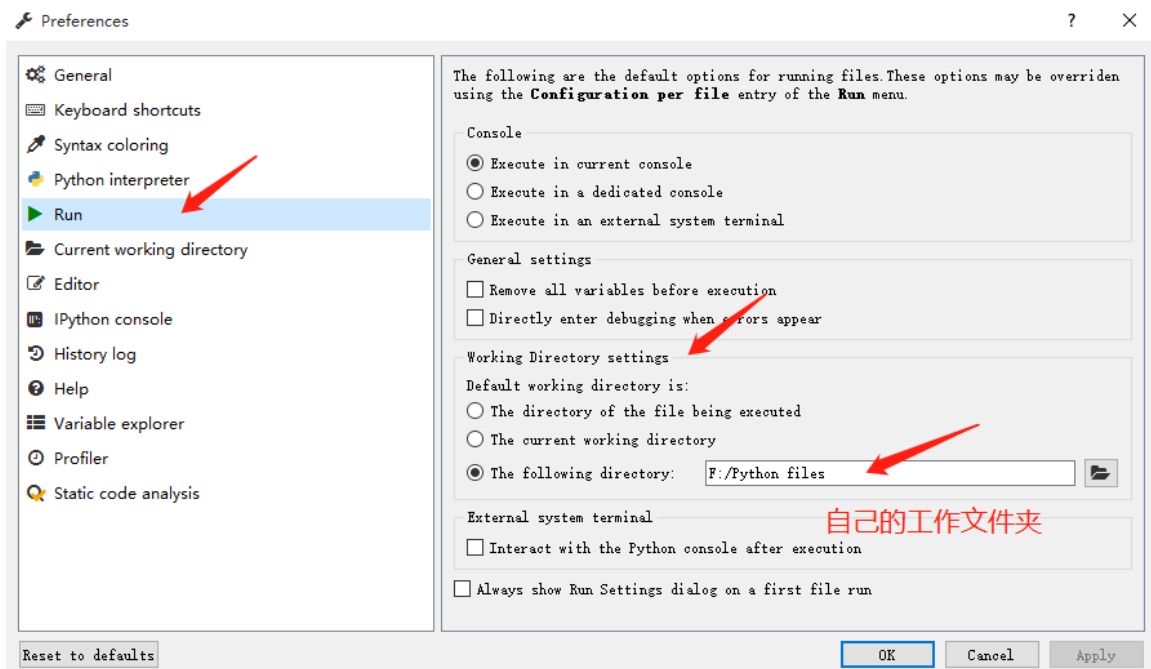


更改编辑器背景颜色：

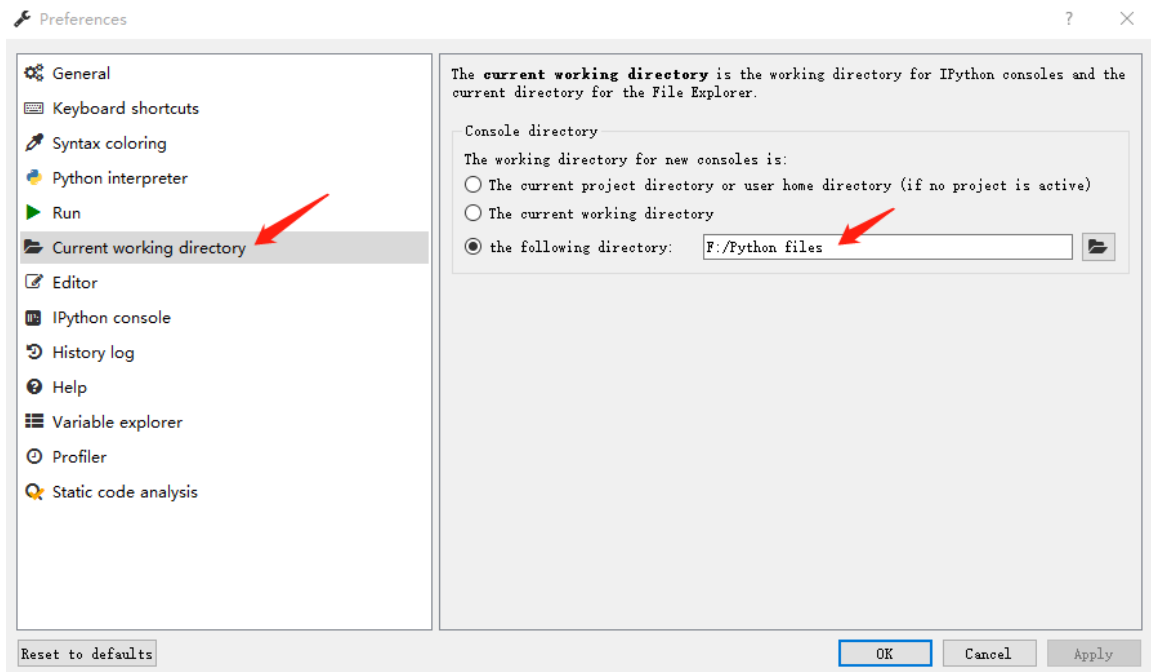


工作路径:

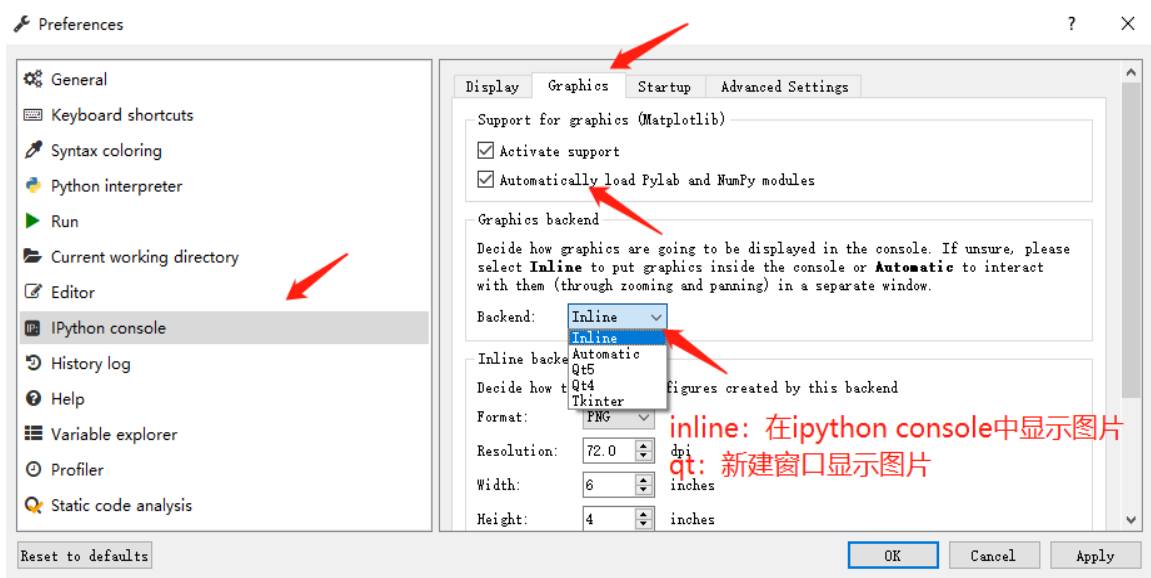
设置默认工作路径: Tools->Preferences->Run->Working directory settings->the following directory 选择自己的文件夹。



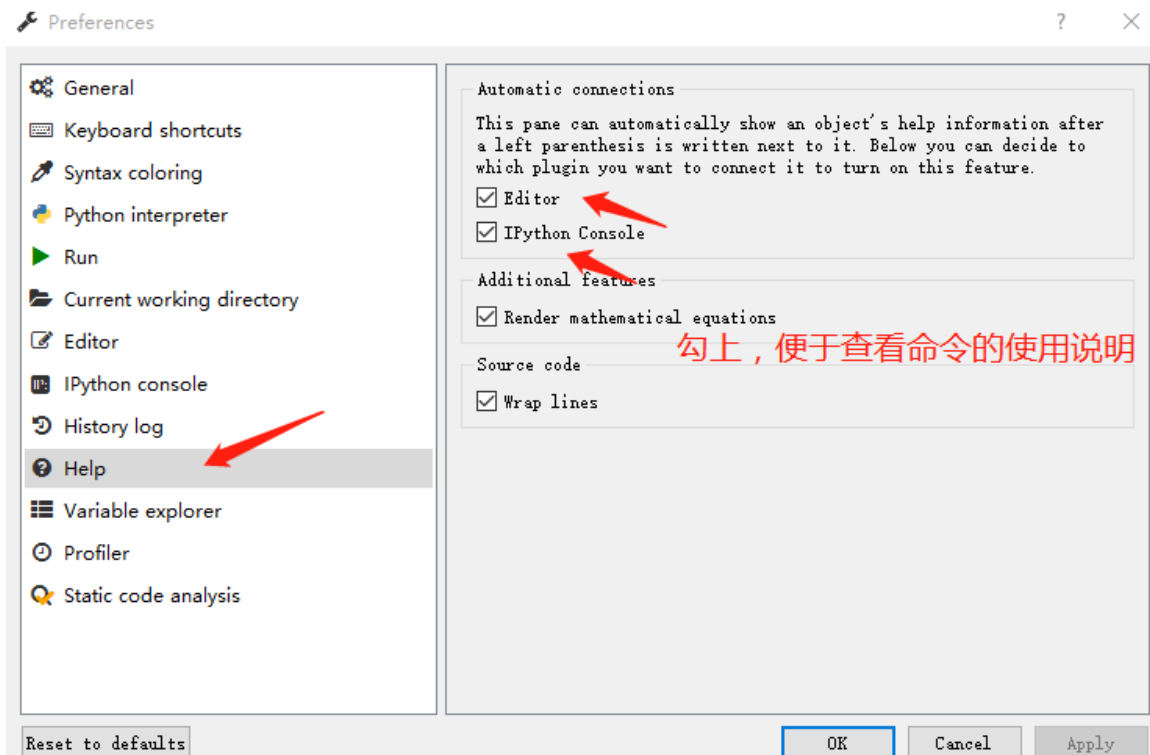
设置当前工作路径:



## IPython 控制台的更改:



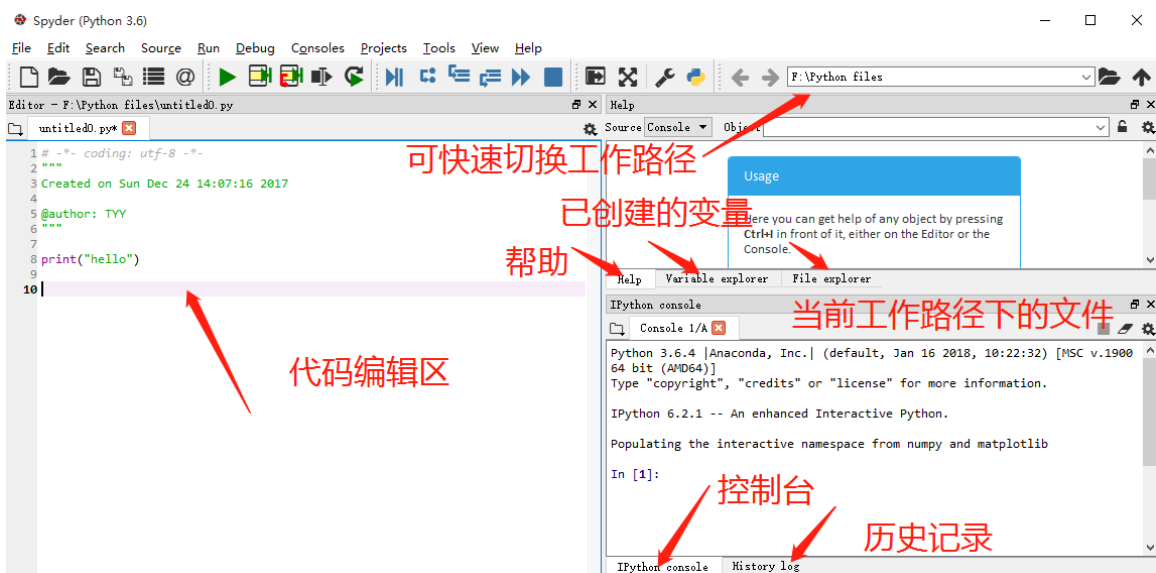
## 帮助设置:



设置之后需要重启!!!

### 3、Spyder 各个窗口操作

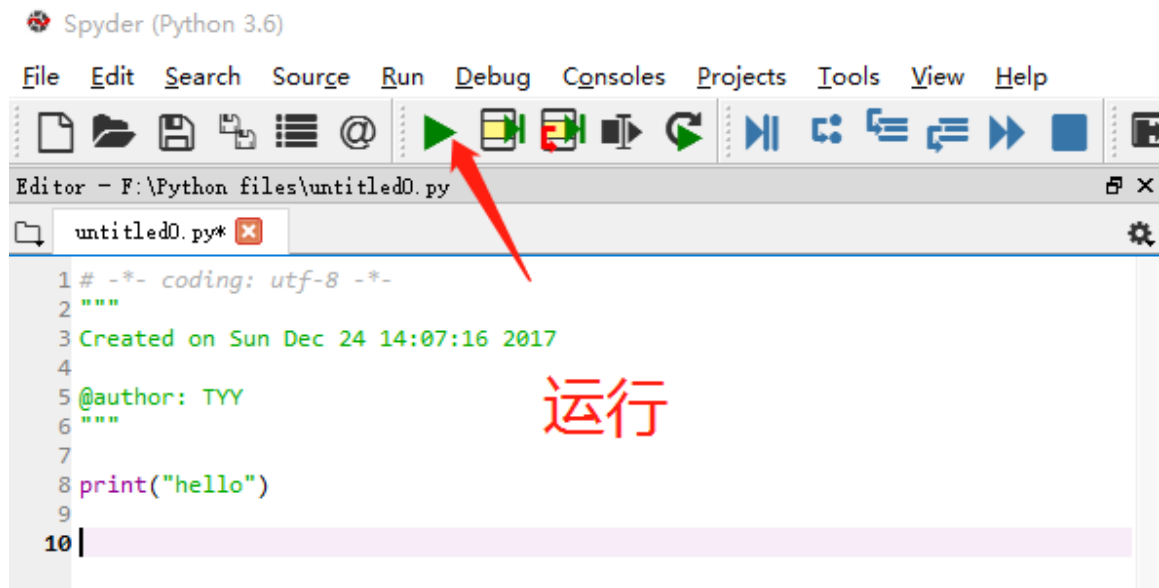
界面一览:



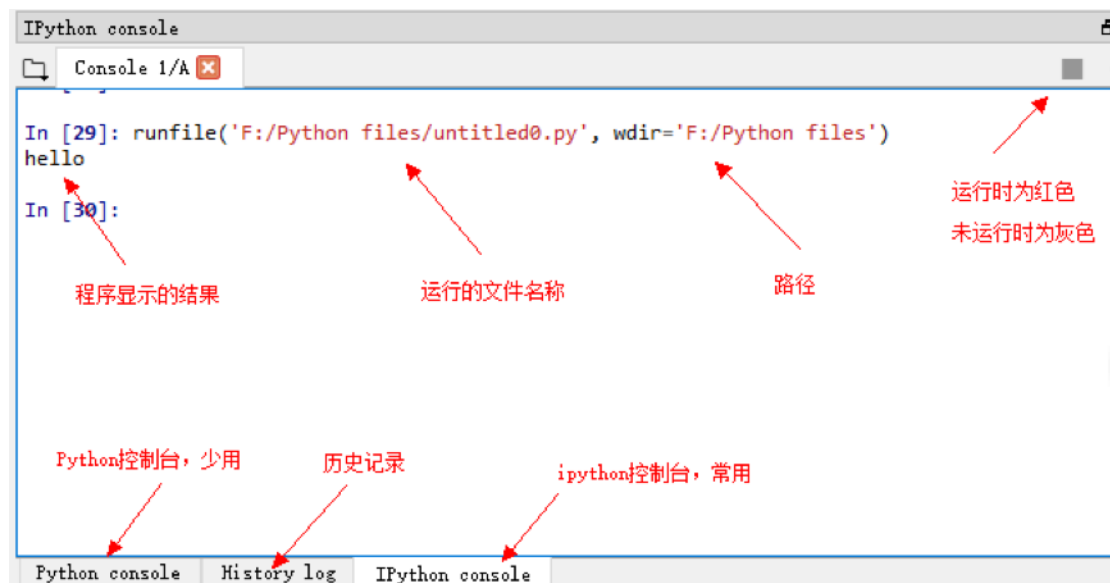
编辑器:



自己编写 py 文件，可全部运行（点击 run 或者 F5 快捷键），也可以部分运行（选中按 F9，单行可以将光标置于该行后，直接按 F9，多行则需先选中再按 F9），在运行之前会让你先进行保存，可选择自己想要保存的路径。保存后，便在控制台显示该文件的相关信息。**注意保存 py 文件尽量不要用中文路径，另外空格也尽量用下划线\_来代替，否则 Python 可能会认不出来，甚至出现很多奇怪的问题！**



控制台：显示结果。如：



输入命令，并显示（有的可以直接显示，有的需要使用 print）或者不显示结果。如：清屏可以使用快捷键 Ctrl+L 或者输入 %clear。

```

IPython console
Console 1/A

In [29]: runfile('F:/Python files/untitled0.py', wdir='F:/Python files')
hello

In [30]: 1+2
Out[30]: 3

In [31]: a=3

In [32]: print(a)
3

In [33]: print("hello")
hello

In [34]:
  
```

Python console   History log   IPython console

**变量环境：**包含四部分，分别是变量名称、对象类型、大小、对象。

清除所有变量在控制台输入 `reset`，选择 `y`

清除部分变量，比如，清除 `a` 和 `lst`，则输入 `del a,lst` 运行便可

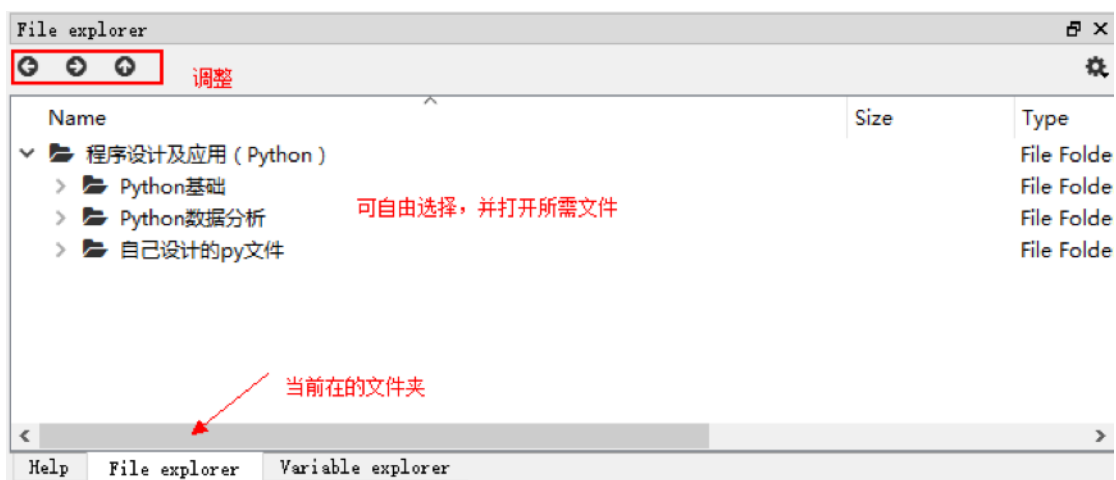
Name	Type	Size	Value
a	int	1	3
lst	list	2	[3, 'hi']
s	str	1	hello

变量名称   对象类型   变量大小   各个变量所赋的值

当前已有的变量

Help   File explorer   Variable explorer

**文件环境：**方便打开所需文件。

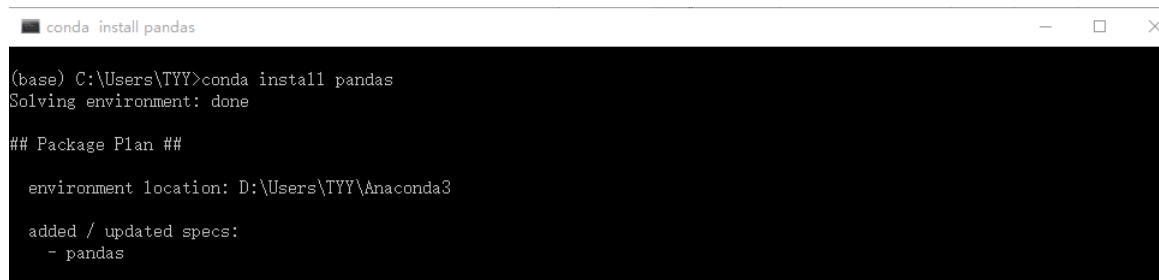


**帮助：**所需要了解的函数的介绍。

## 4、安装包

常用两种方法：conda 和 pip。由于 conda 的简单易操作，此处介绍用 conda 安装包。conda 的用法详见 <https://conda.io/docs/index.html>。如果想要用 pip 来安装包，则详见 <http://blog.csdn.net/lengqi0101/article/details/61921399>。此外还可以通过 pip 来安装静态编译包。

- 操作：打开终端 Anaconda Prompt
- 查看目前已安装的包：conda list
- 安装包：conda install somepackage，比如 conda install pandas，此外可以同时安装多个包，比如 conda install numpy scipy pandas，或限定版本 conda install numpy=1.9
- 更新包：conda upgrade somepackage，注意 conda、anaconda 和 Python 也可以由此更新
- 更新所有包：conda upgrade --all
- 卸载包：conda remove somepackage



```
conda install pandas

(base) C:\Users\TYT>conda install pandas
Solving environment: done

## Package Plan ##

  environment location: D:\Users\TYT\Anaconda3

added / updated specs:
- pandas
```

如何判断包已经安装好了？从命令行进入 python，运行 import somepackage 未报错即顺利安装。

## 5、Python 基础语法

**编码：**

默认 UTF-8 编码

**计算：**

符号：+、-、\*、/、\*\*、//、%、abs、round(3.1425,2)

含义：加、减、乘、除、幂、取整、取余、取绝对值、保留 2 位小数

特殊的：+=、-=、\*=、/=、//=、%=

含义：如果  $a=a+b$ ，那么  $a+=b$

### 判断：

符号：>、<、==、!=、>=、<=、and、or、not

含义：大于、小于、等于、不等于、大于等于、小于等于，且、或、非

### 变量：

类型有：整数、浮点数、字符串、列表、元组、字典和逻辑运算符

### 标识符：

第一个字符必须为字母或者下划线，其余部分可以为字母、数字和下划线。注意标识符对大小写敏感。

### Python 保留字：

不能用它来做标识符，它们各自有各自的含义。比如 True、if、else，是不能拿来作为变量或者函数名的。查找当前版本的保留字：输入 `import keyword`，然后输入 `keyword.kwlist`，便可。

### 注释：

单行注释用#开头，此行#后的所有字符不被运行。

多行注释有三种方法。

一为每行前面输入#，

快捷方式为，选中所需要注释的行，同时按 **Ctrl+1**（再按一次表示取消注释），或者按 **Ctrl+4**，进行块注释，按 **Ctrl+5**，取消块注释；

二为三个单引号，例如 `"""多行注释内容"""`

三为三个双引号，例如 `"""多行注释内容"""`

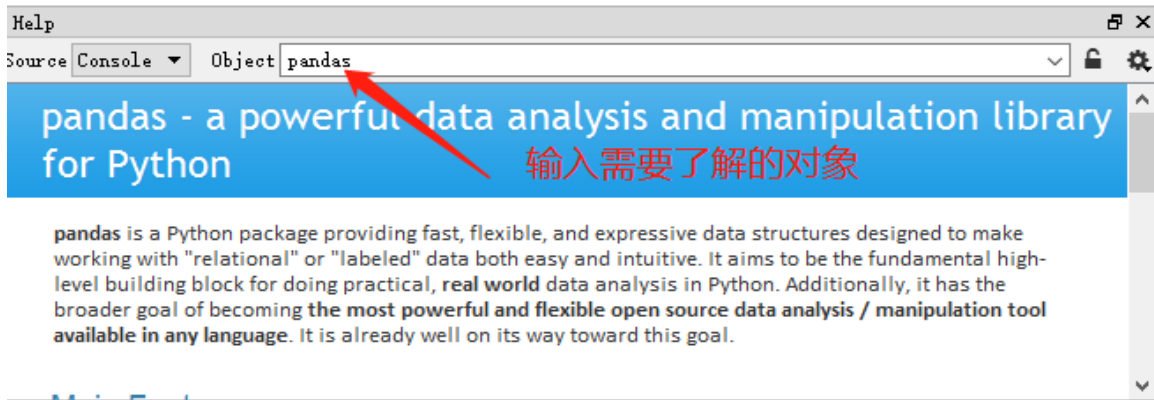
### print 与 input：

输出用 `print()`，输入用 `input()`

练习：询问对方名字，并问好

### 查看包、方法、变量的使用说明：

`help(object)` 或者在 help 窗口输入所需了解的对象。



缩进:

python 使用缩进（空格和制表符）来决定逻辑行的层次，从而用来决定语句的分组。缩进的空格数可变，但是同一个代码块的语句必须包含相同的缩进，一般为 4 个空格或一个 tab 键（注意同一个 py 文件里不要混合使用 tab 或四个空格，否则跨平台工作时可能会报错，建议选择一种方式一直使用）。如：输出数字 1 到 5

```
for n in range(1,6):  
    print(n)  
print("Done")
```

四个空格

多行语句:

若语句很长，一行写不完，则可以用反斜杠（\）来实现多行语句，如：

```
Total=item_one+\  
    item_two
```

但是在[], {}, () 中的多行语句，不需要使用反斜杠（\），直接换行即可。

导入包:

使用 import 或者 from...import

将整个包导入，格式为：import somemodule

从某个包导入某个函数，格式为：from somemodule import somefunction

从某个包导入多个函数，格式为：from somemodule import func1,func2,func3

从某个包导入全部函数，格式为：from somemodule import \*

注意：如果安装好了所需要的包，那么在这里导入的时候不会报错，如果报错了，说明此包需要先安装。

画图:

在控制台输出结果需要加命令：%matplotlib inline

在新窗口输出结果需要加命令：%matplotlib qt5

**控制流（for、while 和 if）：**

for 循环

#问好

```
friends = ['zw','ld','fy','yy']
```

```
for i in friends:
```

```
    print('hello! '+i)
```

```
print('Done')
```

while 循环

#求和

```
num = 1
```

```
sum_num = 0
```

```
while num<11:
```

```
    print(num)
```

```
    sum_num += num
```

```
    num += 1
```

```
print("sum_num = ",sum_num)
```

if 判断结构

#判断奇偶数

```
n = input("请输入一个整数： ")
```

```
n = int(n)
```

```
if n % 2 == 0:
```

```
    print("这是一个偶数")
```

```
else:
```

```
print("这是一个奇数")
```

## 三、Jupyter Notebook 基本介绍

### 1、简介

Jupyter Notebook 是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。——Jupyter Notebook 官方介绍

简而言之，Jupyter Notebook 是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示的程序。如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释。

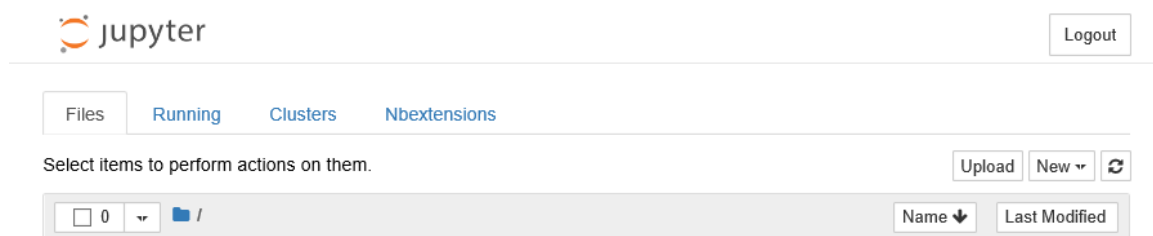
### 2、使用

#### 2.1、打开 Jupyter

- 方法 1：双击 Jupyter Notebook 图标；
- 方法 2：在 Anaconda Prompt 输入 `jupyter notebook`，回车便可。

注意：之后在 **Jupyter** 中的所有操作，都请不要关闭终端（即 Anaconda Prompt），一旦关闭，便会断开与本地服务器的连接，从而无法在 **Jupyter** 上进行其他操作。

打开 Jupyter 之后，主要页面如下所示：



#### 2.2、更改默认路径

- 先创建好自己需要创建的文件夹，我在 F 盘创建了一个 `Python_jupyter` 文件夹。
- 再在终端（Anaconda Prompt）输入命令：`jupyter notebook --generate-config`
- 此行命令表示获取配置文件所在路径，如下图（出现的[y/N]问题选择 N，若未出现问题，则再运行一次）：

```
(base) C:\Users\TTY>jupyter notebook --generate-config
Overwrite C:\Users\TTY\.jupyter\jupyter_notebook_config.py with default config? [y/N]N
```

• 在上图红色框中的.jupyter 文件夹中找到 jupyter\_notebook\_config.py 文件（或者用 search everything 软件查找文件也很方便），可以选择用记事本打开。查找下列内容：

```
## The directory to use for notebooks and kernels.
```

```
#c.NotebookApp.notebook_dir = ''
```

将之改为：

```
## The directory to use for notebooks and kernels.
```

```
c.NotebookApp.notebook_dir = 'F:\ Python_jupyter'
```

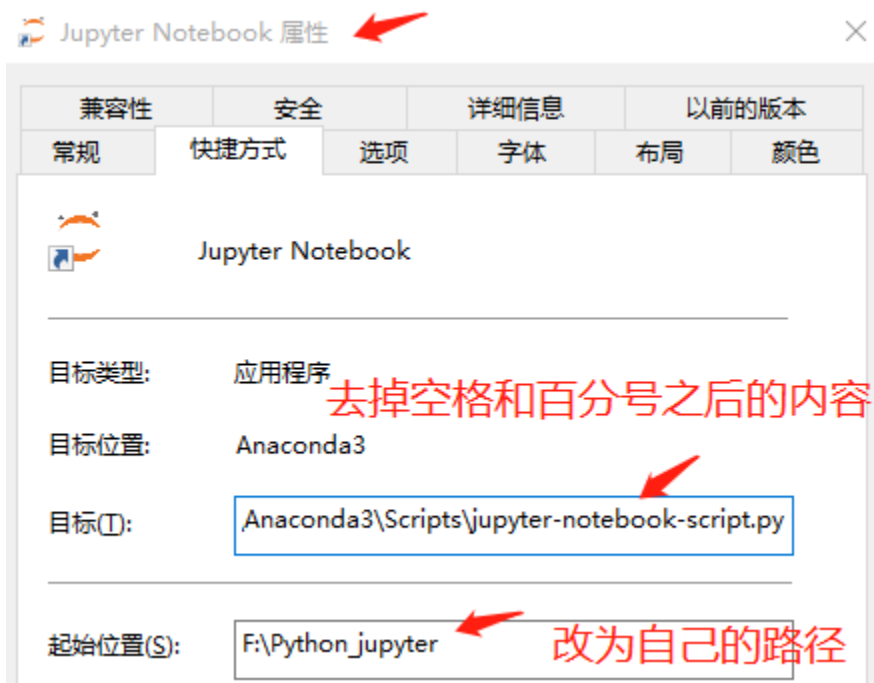
• 此处需要注意两点：

一是#c.NotebookApp.notebook\_dir = ''中的#需要删除，且前面无空格；

二是 F:\ Python\_jupyter 为自己想要保存的工作空间，需要提前新建好文件夹。

此时重新打开 jupyter，便可以看见一个清爽的页面。

如果此时还是不行，那么找到 Anaconda 创建的快捷方式，找到 Jupyter Notebook，右键打开文件所在位置，点击到 Jupyter Notebook，右键属性进入并修改起始位置的地址，删掉目标中的空格和百分号之后的内容，然后应用便可。

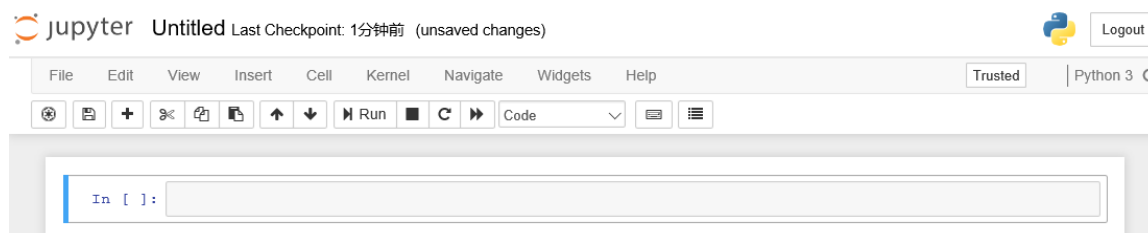




在新打开的 **notebook** 中输入 `%pwd` 就可查询当前的工作路径。

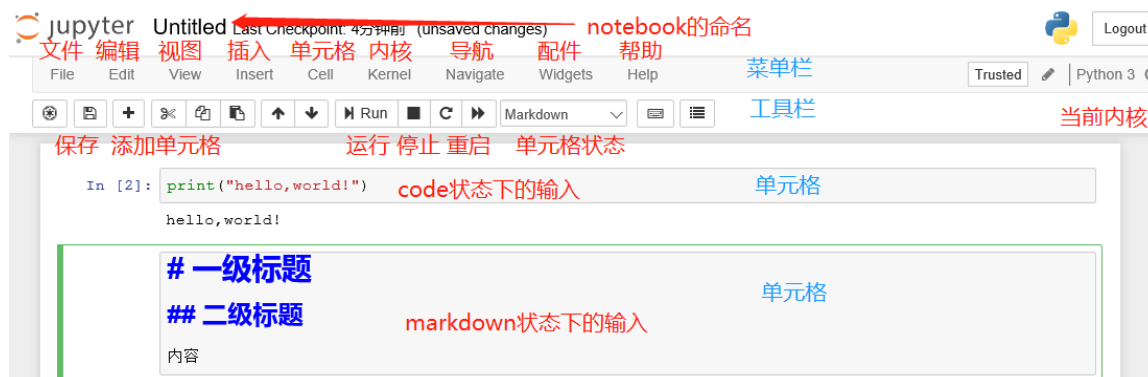
## 2.3、新建 notebook

点击主页面右上方的 **new** 按钮，选择你希望启动的内核，此处我们选择默认内核。在新打开的标签页面中，我们会看到 **notebook** 界面，如下：



## 2.4、notebook 页面介绍

Notebook 界面由四个部分组成：**notebook** 的名称、菜单栏、工具栏和单元格。



### notebook 名称

可以直接点击进行重命名，或者点击 **file** 选择重命名。

### 菜单栏

- **File**: 新建、打开、复制、重命名、保存等常见操作。需要注意的是 **download as** 可以选择多种格式，如.py.html.md 等，默认保存为.ipynb 格式
- **Edit**: 对单元格进行剪切、复制、粘贴等
- **Insert**: 在单元格上方或者下方插入单元格
- **Cell**: 运行单元格
- **Help**: 帮助文档，有需要可以查阅

### 工具栏

需要注意的是单元格状态，有 Code, Markdown, Heading, Raw NBconvert。最常用的是前两个，分别是代码状态，Markdown 编写状态，后两个较少使用。

## 单元格

Notebook 的主要区域是由单元格构成。如在单元格写上 1+2，按下 shift+enter 或者点击快捷键，便可得到如下结果：

```
In [4]: 1+2
Out[4]: 3
```

绿色边线框表示光标现在所处的位置，且可以进行编辑（蓝色边线框则不能编辑）。In 表示输入，Out 表示返回的值。有时不会有 out 出现，因为不需要返回值。如：

```
In [4]: 1+2
Out[4]: 3
In [5]: for i in range(3):
        print(i)
0
1
2
```

注意：每一个单元格在运行之后仍然可以进行修改，比如点击第一个单元格，将 1+2 修改为 2+3，运行得到新的结果，原来的结果会被替换。如：

```
In [6]: 2+3
Out[6]: 5
In [5]: for i in range(3):
        print(i)
0
1
2
```

## 2.5、有用的快捷键

快捷键列表：help > keyboard shortcuts。里面有所有的快捷键说明。

常用的有：

- 执行当前单元格，并自动跳到下一个单元格：Shift Enter
- 执行当前单元格，执行后不自动调到下一个单元格：Ctrl Enter
- 是当前的单元格进入编辑模式：Enter

- 退出当前单元格的编辑模式：Esc
- 删除当前的单元格：双 D
- 将当前的单元格转化为具有一级标题的 markdown：单 1
- 将当前的单元格转化为具有二级标题的 markdown：单 2
- 将当前的单元格转化为具有三级标题的 markdown：单 3
- 为一行或者多行添加/取消注释：Ctrl /
- 撤销对某个单元格的删除：z
- 快速跳转到首个单元格：Ctrl Home
- 快速跳转到最后一个单元格：Ctrl End
- 隐藏和显示输出的单元格结果：Ctrl O
- 选择多个单元格：Shift J（选择下一个），Shift K（选择上一个）
- 合并多个单元格：Shift M

## 2.6、加载或运行本地 Python 文件

### 加载本地 Python 文件

对于在工作路径下的文件，输入命令：%load Python 文件名称（含“.py”）

不在工作路径下的文件，输入命令：%load Python 文件的绝对路径。（路径不要有空格）

```
In [ ]: %load F:/hello_world.py #此py文件不在工作路径下
          写绝对路径，注意路径中不要有空格
In [ ]: %load say_hello.py      #此py文件位于工作路径下
```

注意 Python 文件的后缀为“.py”，在第一次执行之后输入的命令%load Python 文件的绝对路径会自动被注释掉。

### 运行本地 Python 文件

- 方法 1：输入命令：%run Python 文件名称
- 方法 2：输入命令：!python Python 文件名称

```
In [20]: %run say_hello.py
```

```
hello! zw  
hello! ld  
hello! fy  
hello! yy  
Done
```

```
In [21]: !python say_hello.py
```

```
hello! zw  
hello! ld  
hello! fy  
hello! yy  
Done
```

## 2.7、隐藏 notebook 的输入单元格

当使用的单元格较多时，有时我们只需要看输出的结果，而隐藏输入的命令。

代码 1:

```
from IPython.display import display_html
```

```
display_html("""<button  
onclick="$('input, .prompt, .output_stderr, .output_error').toggle();">Toggle  
Code</button>""", raw=True)
```

或者代码 2:

```
%%HTML
```

```
<button
```

```
onclick="$('input, .prompt, .output_stderr, .output_error, .output_result').toggle();">  
Toggle Code</button>
```

缺点：不能很好地适用于 markdown 单元格。

## 2.8、插入图片音乐等

绘图

为了让计算出来的图显示出来，需要先输入：

```
%matplotlib inline （在单元格中显示）
```

或者输入%matplotlib qt5 （在新的界面中显示）

再加载 matplotlib 包（需要先下载，通过在终端输入 conda install matplotlib 或者 pip install matplotlib）

代码及例子如下：

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(20)
y = x**2
plt.plot(x, y)
```

插入本地图片（代码）

```
from IPython.display import Image
Image(filename='name.png')
```

插入本地音乐（代码）

```
from IPython.display import Audio
Audio(filename="name.mp3")
```

插入网络图片、音乐、视频、网页等可参考文章：

[Ipython Notebook 使用教程](#)

## 2.9、Markdown 简介

一篇科普文：[为什么你应该用 Markdown 写作？](#)

Markdown 可以使用 LaTeX 语法来编辑数学公式，相对于 Word 来说方便很多。  
Jupyter Notebook 就可以实现使用 Markdown 写作：

将单元格状态改为 markdown，即可进行 markdown 编辑，或者蓝色框线时单击 M 便可从 code 状态改为 markdown 状态。

**标题：**生成一级标题：在内容前写上 # 和空格

生成二级标题：在内容前写上 ## 和空格

**数学公式：**

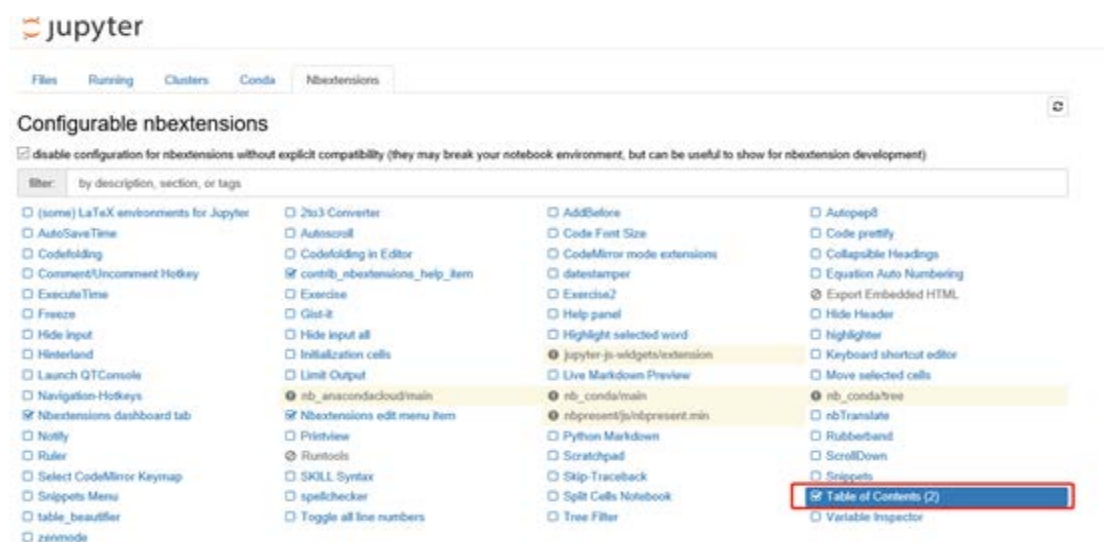
比如在 markdown 模式下输入并运行：

$${}_L\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left( \frac{x}{2} \right)^{2m + \alpha}$$

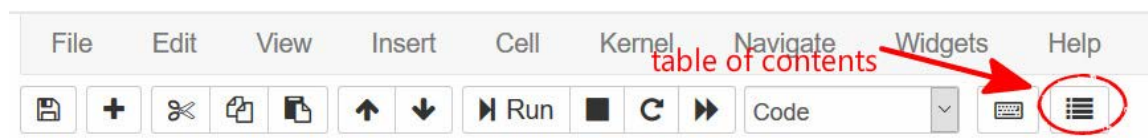
内容：直接编辑即可

生成目录：

- Jupyter 需要通过安装插件来生成目录（注意下列过程花费时间较长，请在时间充裕时操作）
- 在终端输入 `conda install -c conda-forge jupyter_contrib_nbextensions`
- 完成后，重新打开 jupyter，发现主页面多出了一个选项 Nbextensions，点击此标签，选择 Table of Contents(2)，操作如下：



然后在新打开的 notebook 中便可以看到目录快捷键了。



## 2.10、关闭 jupyter

- 先关闭 notebook 页面
- 回到 Home 页面，点击 running，找到对应的 notebook，点击右侧的 shutdown 按钮
- 仅关闭 notebook 页面并没有真正关闭该.ipynb 文件。

### 三、其他 IDE

除了 Spyder 以外，Python 还有其他好用的 IDE（集成开发环境），比如 Pycharm，文本编辑器 Sublime Text 和 Atom 通过安装插件也可以改造成优秀的 IDE，以及 Vsual Studio Code，根据不同的目的选择不同的 IDE。如果觉得用起来不顺手，改投其他 IDE 也是不错的选择。

不过建议大家先通过 Jupyter 以及 Spyder 掌握 Python 的基本用法之后再尝试不同的 IDE，这样能够更好地体会到不同工具的特点，选择更适合自己的工具，避免盲目更换 IDE 带来的时间和精力成本。

### 四、自己动手做一做

课后题：

猜年龄。写一个程序让别人猜自己的年龄，若对方猜对，则返回“恭喜你，猜对了”，否则接着猜，直到猜对。注意可能用上的命令：`while`、`if`、`int`、`input`、`print`。