

Jupyter notebook入门教程（上）

原创：红色石头 AI有道 4月8日



Jupyter notebook（又称IPython notebook）是一个交互式的笔记本，支持运行超过40种编程语言。本文中，我们将介绍Jupyter notebook的主要特点，了解为什么它能成为人们创造优美的可交互式文档和教育资源的一个强大工具。

在使用这个notebook之前，你需要先进行安装。你可以在Jupyter website上找到完整的安装步骤。这里，小编推荐使用Anaconda，因为Anaconda自带了Numpy、Scipy、Matplotlib等多种python开发包和Jupyter notebook。



下载网址：

<https://www.anaconda.com/download/>

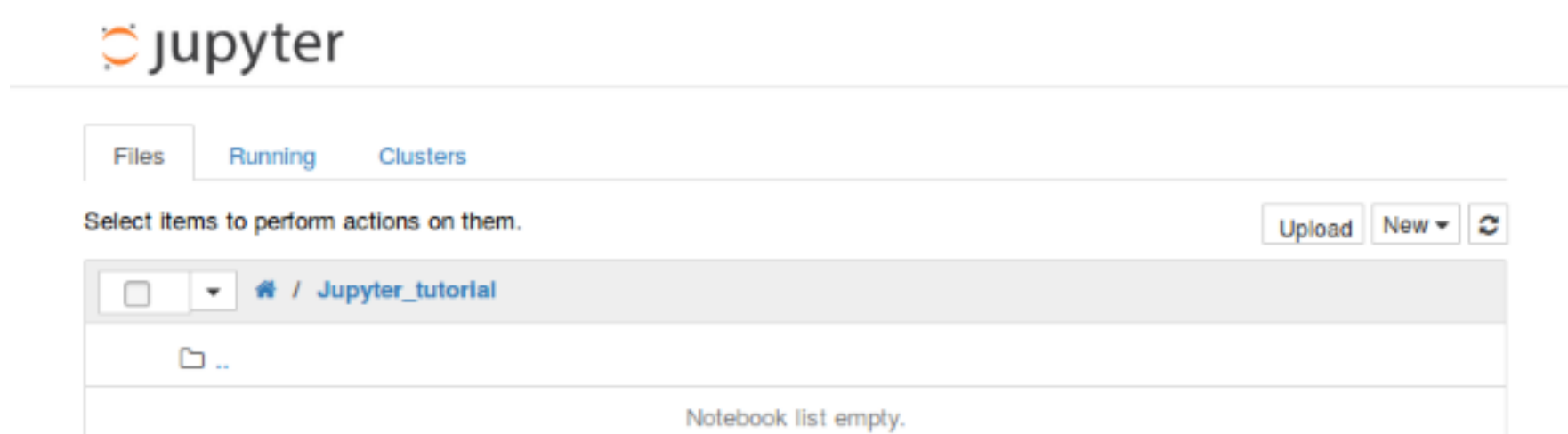
Anaconda有对应的Linux、Windows、macOS版本。这里以Linux系统为例来介绍Jupyter notebook教程。安装完毕后，在Linux终端运行下列命令：

```
jupyter notebook
```

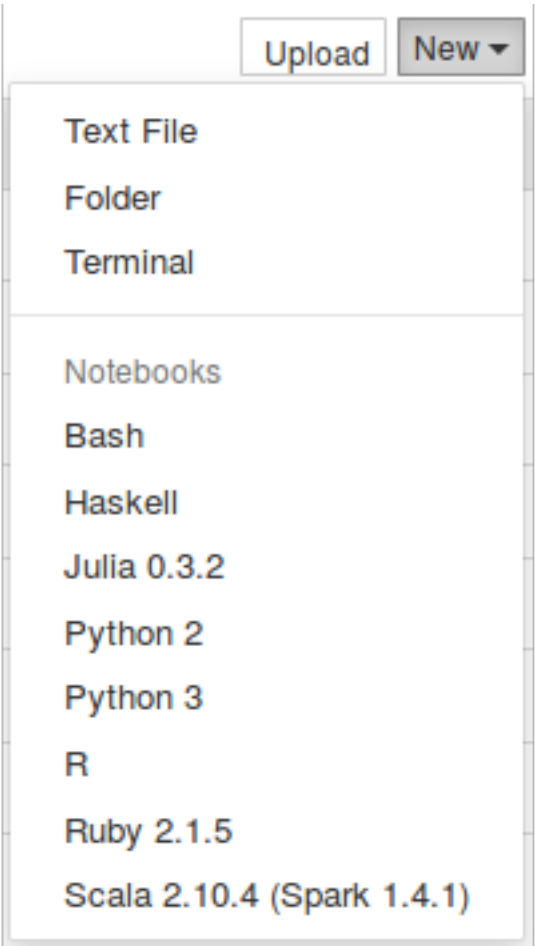
然后，你将看到类似下面显示的语句：

```
[I 20:06:36.367 NotebookApp] Writing notebook server cookie secret to /run/user/
[I 20:06:36.813 NotebookApp] Serving notebooks from local directory: /home/your_
[I 20:06:36.813 NotebookApp] 0 active kernels
[I 20:06:36.813 NotebookApp] The IPython Notebook is running at: http://localhost
[I 20:06:36.813 NotebookApp] Use Control-C to stop this server and shut down all
```

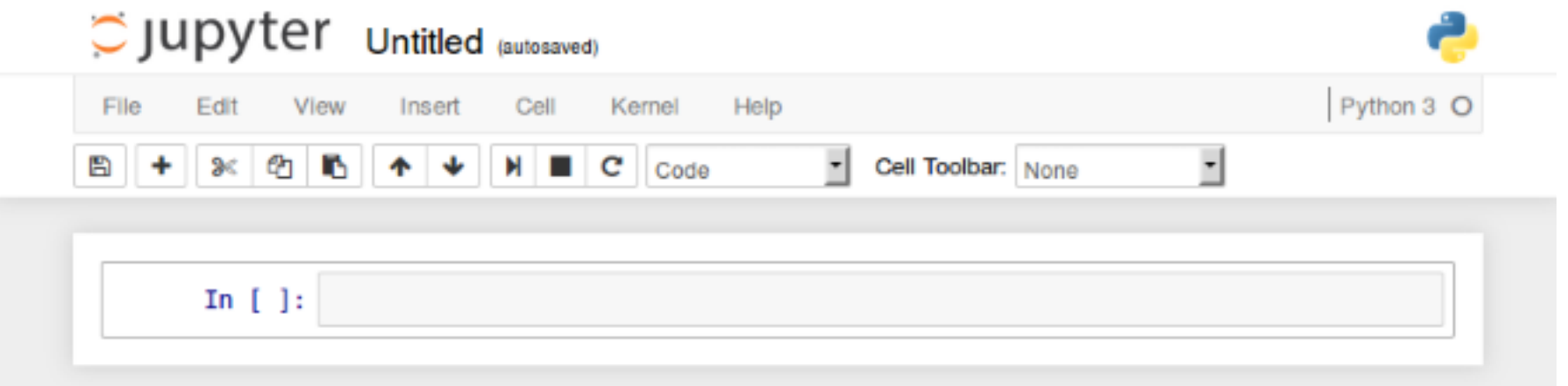
Jupyter的主窗口显示的路径一般是你启动notebook时位于的路径下（通常在你的用户目录下）。主界面如下所示：



若要创建一个新的notebook，只需鼠标左击New，在下拉选项选择一个你想启动的notebook类型即可。



因为我只安装了Python内核，所以启动Python notebook来进行演示，打开新的标签，得到了一个空的notebook界面。



你可以看到notebook由以下部分组成：

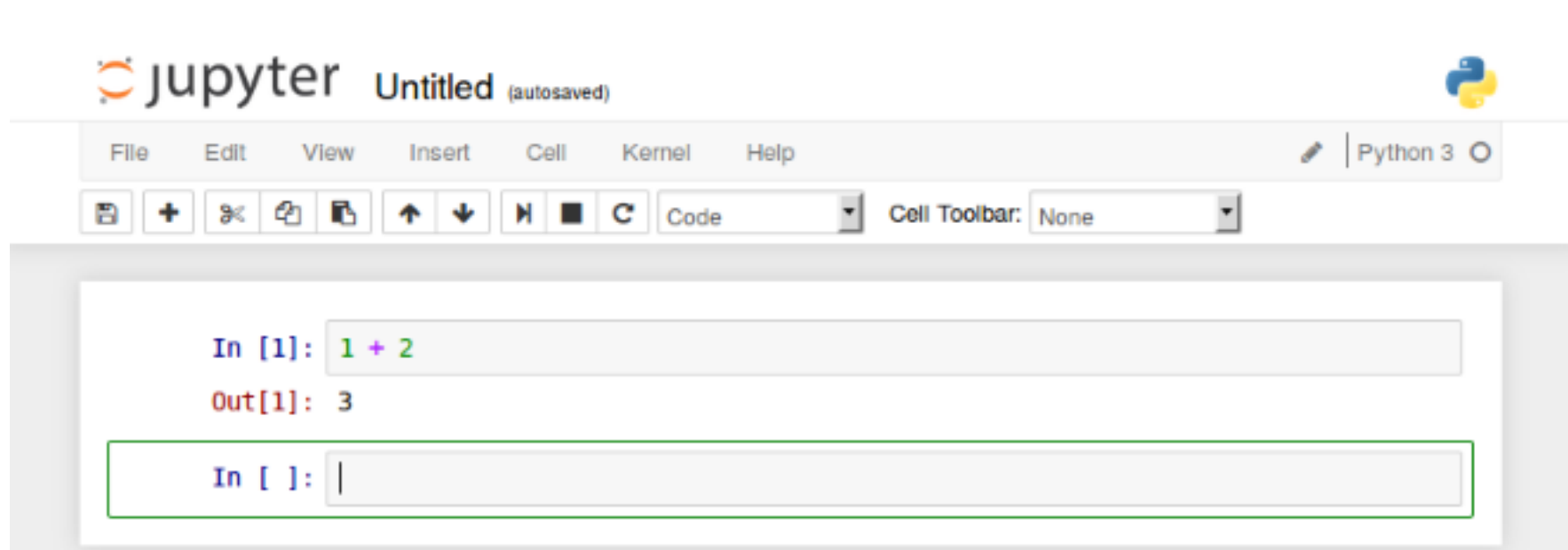
- notebook名字
- 主工具栏，包括保存notebook、导出、重载、重启内核等
- 快捷键
- notebook主要部分，notebook编辑区

花时间熟悉菜单，了解其中的选项。如果你想了解一些特殊模块、notebook或者一些库，可以点击菜单栏右侧的Help按钮。

在主区域（编辑区），你可以看到一个个单元（cell）。每个notebook都由许多cell

组成，每个cell有不同的功能。

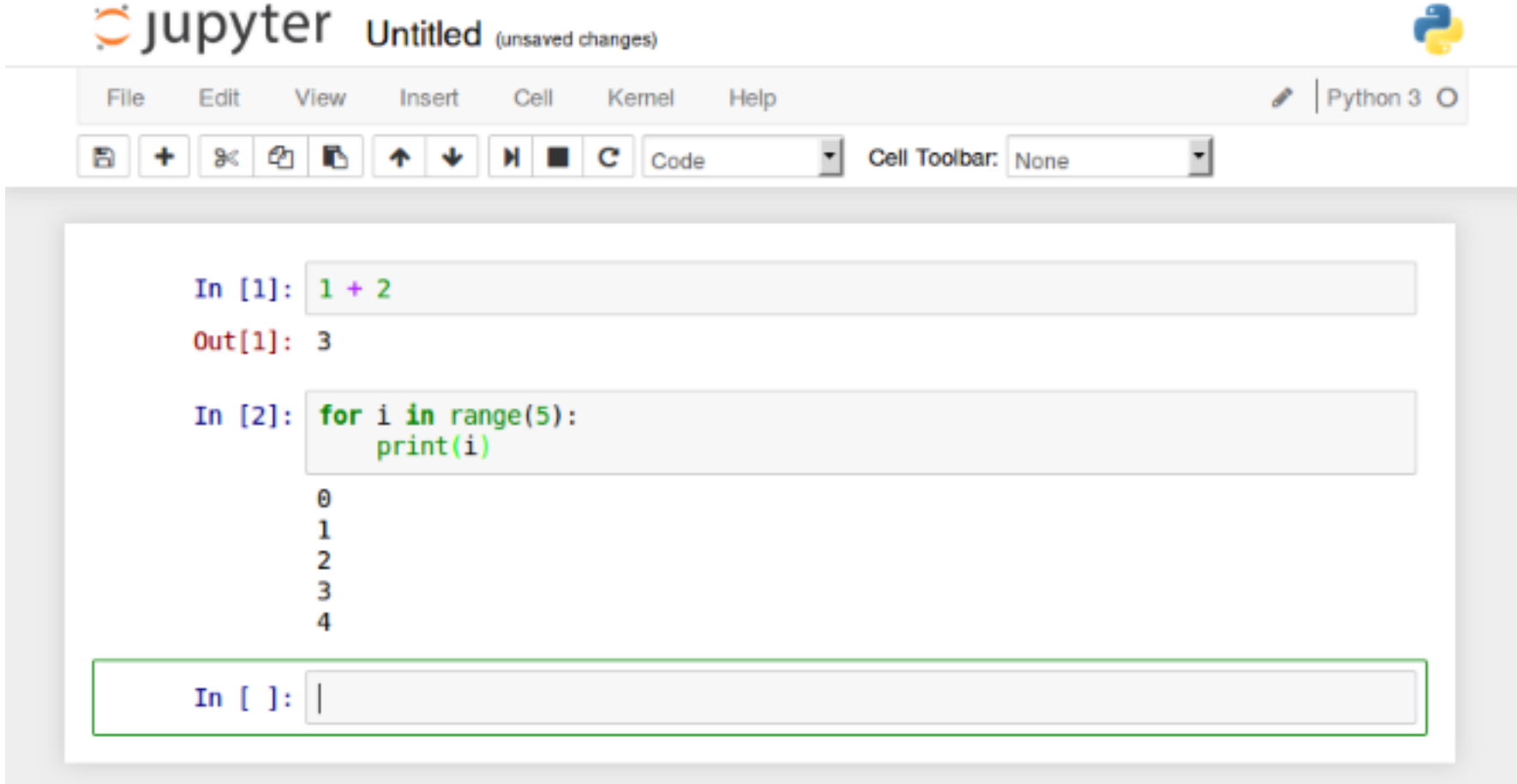
首先，第一个cell如下图所示，以“In[]”开头表示这是一个代码单元。在代码单元里，你可以输入任何代码并执行。例如，键盘输入1+2，然后按“Shift+Enter”，代码将被运行，并显示结果。同时，切换到新的cell中。



绿色的轮廓线所在的cell，就是当前工作单元。接下来，我们在第二个cell中，敲击下列代码：

```
for i in range(5):  
    print(i)
```

按“Shift+Enter”运行该单元，得到：

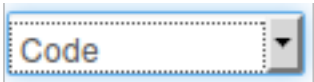


跟之前一样，代码被执行，结果也正确显示处理。但是你可能注意到这次没有 Out[2]，这是因为我们打印了结果，却没有返回值。

notebook一个非常有趣的特性是可以返回之前的cell，修改并重新运行，以此来跟新整个文档。例如，现在我们回到第一个cell中，将输入 1+2 改成 2+3，重新按“Shift+Enter”运行该单元，结果被立即更新成5。当你想使用不同参数调试方程又不想运行整个脚本的时候，这条特性非常有用。然而，你也可以通过菜单栏 Cell->Run all来重新运行整个notebook。

现在我们已经知道了如何输入代码，那为什么不尝试获得一个更加优美，解释性更强的notebook呢？这可以通过使用其他类型的cells来获得，例如Header和Markdown单元。

首先，我们在notebook的顶部添加一个标题。做法是，选择第一个cell，点击Insert->Insert cell above，则在这个cell上面会添加一个新的单元。但是这个新cell默认是code类型，我们可以通过cell type菜单选择Heading或Markdown替换之前的code类型，这样就可以添加标题了。



你可以选择Heading，然后弹出信息会告诉你如何创建不同级别的标题，这样你就有

了与code不同类型的cell。

#

In [1]: 1 + 2

Out[1]: 3

这个cell以#开头，表示一级标题。如果你想列子标题的话，参考下面的方法：

```
# : First level title
## : Second level title
### : Third level title
...
```

标题写在#后面，然后重新运行cell，你会发现该单元添加了标题，非常生动可读，示例如下：

My first title in Jupyter

A very simple operation

In [1]: 1 + 2

Out[1]: 3

Counter

In [2]: for i in range(5):
 print(i)

0
1
2
3
4

In []:

添加标题之后，我们要对每个代码类型的cell做相应的解释。为此，我们在相应的位置添加cell，将其类型改为Markdown，然后重新运行cell，这样就能让解释文本丰富起来了。

My first title in Jupyter

A very simple operation

Let's **add** two numbers:

```
In [1]: 1 + 2
```

```
Out[1]: 3
```

Counter

Let's *count* from 0 to 4:

```
In [2]: for i in range(5):  
        print(i)
```

```
0  
1  
2  
3  
4
```

```
In [ ]:
```

最后，你可以将新建的notebook重命名，点击File->Rename，然后输入新的名称即可。这样更改后的名字就会出现在窗口的左上角，位于Jupyter图标右侧。

在下一次的教程中，我们将对notebook的能力以及如何集成其他Python库进行更深入的研究。



推荐阅读：

[免费送！超全的人工智能资料（100G+）](#)



长按二维码扫描关注
红色石头的机器学习之路
ID: redstonewill



红色石头
个人微信: WillowRedstone
新浪微博: @RedstoneWill