**Jupyter Notebooks 是什么？**

Jupyter Notebooks 是一款开源的网络应用，我们可以将其用于创建和共享代码与文档。其提供了一个环境，你无需离开这个环境，就可以在其中**编写你的代码、运行代码、查看输出、可视化数据并查看结果**。因此，这是一款可执行端到端的数据科学工作流程的便捷工具，其中包括**数据清理、统计建模、构建和训练机器学习模型、可视化数据**等等。

当你还处于原型开发阶段时，Jupyter Notebooks 的作用更是引人注目。这是因为你的**代码是按独立单元的形式编写的，而且这些单元是独立执行的**。这让用户可以测试一个项目中的特定代码块，而无需从项目开始处执行代码。很多其它 IDE 环境（比如 RStudio）也有其它几种方式能做到这一点，但我个人觉得 Jupyter 的单个单元结构是最好的。

**一、创建一个新的notebook文件**

jupyter notebook

运行上面的命令之后，你将看到类似下面这样的输出：

[I 20:06:36.367 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook\_cookie\_secret

[I 20:06:36.813 NotebookApp] Serving notebooks from local directory: /home/your\_username

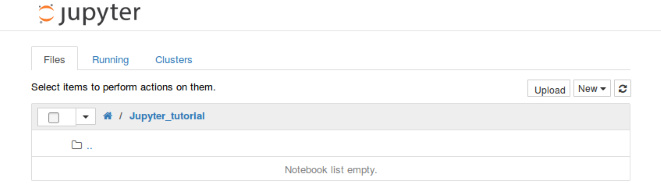
[I 20:06:36.813 NotebookApp] 0 active kernels

[I 20:06:36.813 NotebookApp] The IPython Notebook is running at: http://localhost:8888/

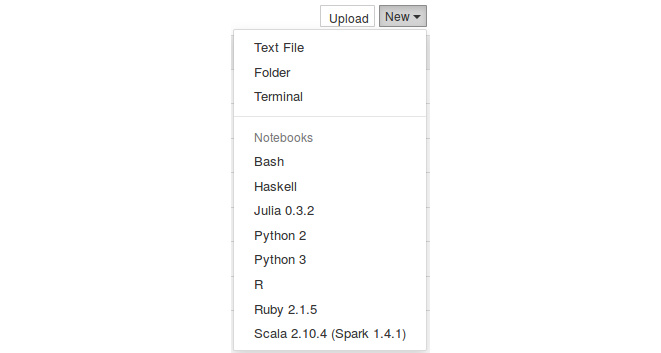
[I 20:06:36.813 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

同时，会在你开启 notebook 的文件夹中启动 Jupyter 主界面，如下所示：

打开笔记本后，你会看到顶部有三个选项卡：Files、Running 和 Clusters。其中，Files 基本上就是列出所有文件，Running 是展示你当前打开的终端和笔记本，Clusters 是由 IPython 并行提供的。



如果想新建一个 notebook，只需要点击New，选择你希望启动的 notebook 类型即可。

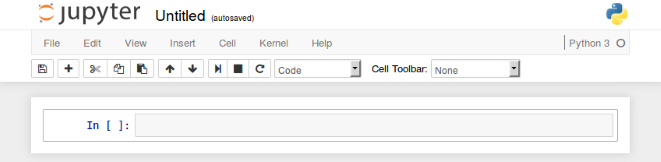


1、选择 Text File，你会得到一个空面板。你可以添加任何字母、单词和数字。其基本上可以看作是一个文本编辑器（类似于 Ubuntu 的文本编辑器）。你可以在其中选择语言（有很多语言选项），所以你可以在这里编写脚本。你也可以查找和替换该文件中的词。

2、选择 Folder 选项时，你会创建一个新的文件夹，你可以在其中放入文件，重命名或删除它。各种操作都可以。

Terminal 完全类似于在 Mac 或 Linux 机器上的终端（或 Windows 上的 cmd）。其能在你的网络浏览器内执行一些支持终端会话的工作。在这个终端输入 python，你就可以开始写你的 Python 脚本了！

3、这里，因为我只有一个 Python 内核，所以我们运行一个 Python notebook。在新打开的标签页中，我们会看到 notebook 界面，目前里面什么也没有。



notebook 界面由以下部分组成：

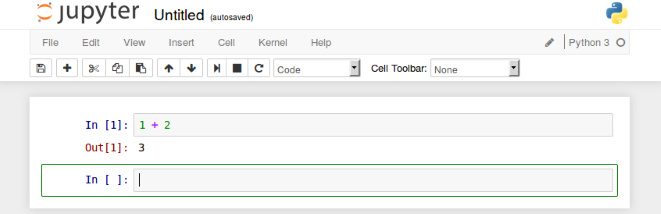
1. notebook 的名称
2. 主工具栏，提供了保存、导出、重载 notebook，以及重启内核等选项
3. 快捷键
4. notebook 主要区域，包含了 notebook 的内容编辑区

慢慢熟悉这些菜单和选项。如果想要详细了解有关 notebook 或一些库的具体话题，可以使用菜单栏右侧的帮助菜单。

下方的主要区域，由被称为单元格的部分组成。每个 notebook 由多个单元格构成，而每个单元格又可以有不同的用途。

**二、运行程序**

下方截图中看到的是**一个代码单元格（code cell）**，以[ ]开头。在这种类型的单元格中，**可以输入任意代码并执行**。例如，**输入1 + 2并按下Shift + Enter**。之后，单元格中的代码就会被计算，光标也会被移动动一个新的单元格中。你会得到如下结果：

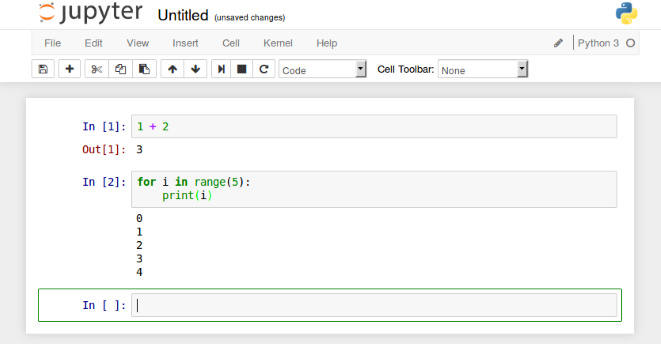


根据绿色边框线，我们可以轻松地识别出当前工作的单元格。接下来，我们在第二个单元格中输入些其他代码，例如：

for i in range(5):

print(i)

对上面的代码求值时，你会得到：



和前一个示例一样，代码被计算之后，马上就会显示结果。你应该注意到了，这次没有出现类似Out[2]这样的文字。这是因为我们将结果打印出来了，没有返回任何的值。

notebook 有一个非常有趣的特性，就是可以修改之前的单元格，对其重新计算，这样就可以更新整个文档了。试着把光标移回第一个单元格，并将1 + 2修改成2 + 3，然后按下Shift + Enter重新计算该单元格。你会发现结果马上就更新成了 5。如果你不想重新运行整个脚本，只想用不同的参数测试某个程式的话，这个特性显得尤其强大。不过，你也可以重新计算整个 notebook，只要点击Cell -> Run all即可。

**Ps:使用 Jupyter Notebooks 的神奇功能**

Jupyter Notebooks 的开发者已经在其中内置了一些预定义的神奇功能，能让你的生活更轻松，让你的工作更具交互性。你可以运行下面的命令来查看功能列表（注：% 符号通常不需要，因为自动补齐功能通常是开启的）：

%lsmagic

随后列出了很多选择%clear、%autosave、%debug 和 %mkdir 等功能你以前肯定见过。

神奇的命令可以以两种方式运行：**逐行方式**、**逐单元方式**

顾名思义，逐行方式是执行单行的命令，而逐单元方式则是执行不止一行的命令，而是执行整个单元中的整个代码块。

在逐行方式中，所有给定的命令必须以 % 字符开头；而在逐单元方式中，所有的命令必须以 %% 开头。我们看看下列示例以便更好地理解：

逐行方式：

%time a = range(10)

逐单元方式：

%%timeit a = range (10)min(a)

**三、给程序添加标题和Markdown注释**

现在我们已经知道了如何输入代码，为什么不尝试着让这个 notebook 更加漂亮、内容更丰富？为此，我们需要使用其他类型的单元格，即 **Header单元格和 Markdown单元格等。**

**可以添加的单元格类型概述**：

Code——不言而喻，就是写代码的地方。

Markdown——这是写文本的地方。你可以在运行一段代码后添加你的结论、添加注释等。

Raw NBConvert——这是一个可将你的笔记本转换成另一种格式（比如 HTML）的命令行工具。

Heading——这是你添加标题的地方，这样你可以将不同的章节分开，让你的笔记本看起来更整齐更清晰。这个现在已经被转换成 Markdown 选项本身了。输入一个「##」之后，后面输入的内容就会被视为一个标题。

**1**、首先，我们在顶部添加一个 notebook 的标题。选中第一个单元格，然后点击Insert -> Insert单元格above（在上方插入单元格）。你会发现，文档的顶部马上就出现了一个新的单元格。点击在快捷键栏中的单元格类型，将其变成一个标题单元格（heading cell）：

变成一个标题 cell

选中下拉选项中的 Heading。然后会出现一个弹出消息，告诉你如何创建不同层级的标题，这样你就有了一个不同类型的 cell：



这个单元格以#标记开头，意味着这是一个一级标题。如果需要子标题，可以使用以下标记表示（改变单元格类型时弹出消息中有解释）：

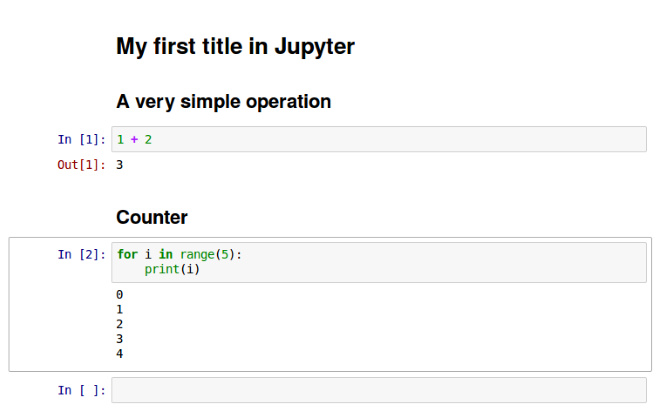
# : 一级标题

## : 二级标题

### : 三级标题

...

在#之后写下文档的标题，然后计算该单元格。你会发现一个样式非常好看的标题。作为示例和练习，我还添加了其他几个标题单元格：



**2**、添加好标题之后，我们在编写一些解释，介绍每个代码单元格中的情况。为此，我们要在相应的地方插入单元格，然后将其类型变成 Markdown。然后，计算新的单元格。就这样，你的解释文本就漂亮地渲染出来了！



最后，你可以重命名该 notebook，点击Fiel -> Rename，然后输入新的名称。这样，新的名称将会出现在窗口的左上角，在 Jupyter 的标志旁边。

**四、单元格操作**

**1、基本操作**

高级单元格操作，将让编写 notebook 变得更加方便。举例如下：

* 如果想删除某个单元格，可以选择该单元格，然后依次点击Edit -> Delete Cell；
* 如果想移动某个单元格，只需要依次点击Edit -> Move cell [up | down]；
* 如果想剪贴某个单元测，可以先点击Edit -> Cut Cell，然后在点击Edit -> Paste Cell [Above | Below]；
* 如果你的 notebook 中有很多单元格只需要执行一次，或者想一次性执行大段代码，那么可以选择合并这些单元格。点击Edit -> Merge Cell [Above | below]。

记住这些操作，它们可以帮助你节省许多时间。

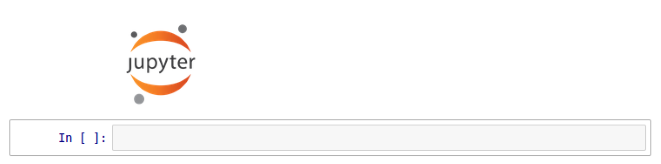
**2、Markdown 单元格高级用法**

我们再来看看 Markdown 单元格。虽然它的类型是 markdown，但是这类单元格也接受 HTML 代码。这样，你就可以在单元格类实现更加丰富的样式，添加图片，等等。例如，如果想在 notebook 中添加 Jupyter 的 logo，将其大小设置为 100px x 100px，并且放置在单元格左侧，可以这样编写：

<img src="http://blog.jupyter.org/content/images/2015/02/jupyter-sq-text.png"

style="width:100px;height:100px;float:left">

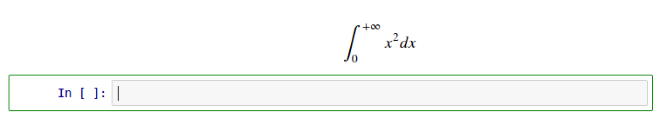
计算该单元格之后，会出现这样的结果：



另外，markdown 单元格还支持 LaTex 语法。例如：

$$\int\_0^{+\infty} x^2 dx$$

计算上述单元格，将获得下面的 LaTex 方程式：



**五、导出功能**

进入「Files」菜单，你会看到「Download As」选项：

你可以用 7 种可选格式保存你的笔记本。其中最常用的是 .ipynb 文件和 .html 文件。

1、使用 .ipynb 文件可让其他人将你的代码复制到他们的机器上，

2、使用 .html 文件能以网页格式打开（当你需要保存嵌入在笔记本中的图片时会很方便）。

3、你也可以使用 nbconvert 选项手动将你的笔记本转换成 HTML 或 PDF 等格式。

4、你也可以使用 jupyterhub，地址：https://github.com/jupyterhub/jupyterhub。其能让你将笔记本托管在它的服务器上并进行多用户共享。很多顶级研究项目都在使用这种方式进行协作。

5、导出 PDF 功能，可以让你不用写 LaTex 即可创建漂亮的 PDF 文档。你还可以将 notebook 作为网页发布在你的网站上。甚至，你可以导出为 ReST 格式，作为软件库的文档。

**六、键盘快捷键**

快捷方式是 Jupyter Notebooks 最大的优势之一。当你想运行任意代码块时，只需要按 Ctrl+Enter 就行了。Jupyter Notebooks 提供了很多键盘快捷键，可以帮助我们节省很多时间。

**1、Jupyter Notebooks 提供了两种不同的键盘输入模式——命令和编辑**。

你可以分别使用 **Esc 和 Enter** 在命令模式和编辑模式之间跳跃。

**（1）、命令模式**是将键盘和笔记本层面的命令绑定起来，并且由带有蓝色左边距的灰色单元边框表示。

进入命令模式之后（此时你没有活跃单元），你可以尝试以下快捷键：

A 会在活跃单元之上插入一个新的单元

B 会在活跃单元之下插入一个新单元。

连续按两次 D可以删除一个单元。

撤销被删除的单元，按 Z。

Y 会将当前活跃的单元变成一个代码单元。

按住 Shift +上或下箭头可选择多个单元。

在多选模式时，按住 Shift + M 可合并你的选择。按 F 会弹出「查找和替换」菜单。

**（2）、编辑模式**让你可以在活动单元中输入文本（或代码），用绿色单元边框表示。

处于编辑模式时（在命令模式时按 Enter 会进入编辑模式），你会发现下列快捷键很有用：

Ctrl + Home 到达单元起始位置。

Ctrl + S 保存进度。

Ctrl + Enter 会运行你的整个单元块。

Alt + Enter 不止会运行你的单元块，还会在下面添加一个新单元。

Ctrl + Shift + F 打开命令面板。

要查看键盘快捷键完整列表，可在命令模式按「H」或进入「Help > Keyboard Shortcuts」。你一定要经常看这些快捷键，因为常会添加新的。

**2、有用的 Jupyter Notebooks 扩展**

扩展/附加组件是一种非常有生产力的方式，能帮你提升在 Jupyter Notebooks 上的生产力。我认为安装和使用扩展的最好工具之一是 Nbextensions。在你的机器上安装它只需简单两步（也有其它安装方法，但我认为这个最方便）：

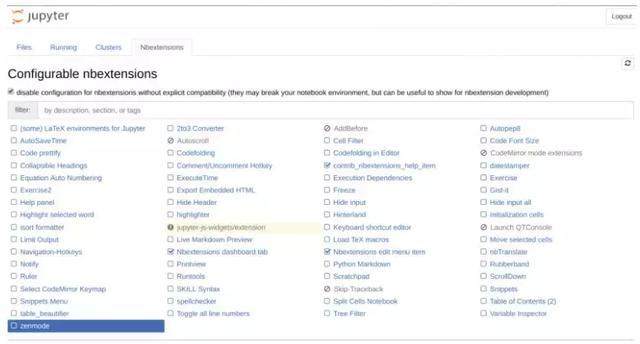
第一步：从 pip 安装它：

pip install jupyter\_contrib\_nbextensions

第二步：安装相关的 JavaScript 和 CSS 文件：

jupyter contrib nbextension install –user

完成这个工作之后，你会在你的 Jupyter Notebook 主页顶部看见一个 Nbextensions 选项卡。点击一下，你就能看到很多可在你的项目中使用的扩展。（**如果没有Nbextensions 选项卡，可以在编辑代码的界面，选择Edit-nbextensions config** ）



要启用某个扩展，只需勾选它即可。下面我给出了 4 个我觉得最有用的扩展：

1、Code prettify：它能重新调整代码块内容的格式并进行美化。

2、Printview：这个扩展会添加一个工具栏按钮，可为当前笔记本调用 jupyter nbconvert，并可以选择是否在新的浏览器标签页显示转换后的文件。

3、Scratchpad：这会添加一个暂存单元，让你可以无需修改笔记本就能运行你的代码。当你想实验你的代码但不想改动你的实时笔记本时，这会是一个非常方便的扩展。

4、Table of Contents (2)：这个很棒的扩展可以收集你的笔记本中的所有标题，并将它们显示在一个浮动窗口中。

**七、Matplotlib 集成**

如果你用 Python 绘制过图形，那你肯定知道 matplotlib。Matplotlib 是一个用于创建漂亮图形的 Python 库，结合 Jupyter notebook 使用时体验更佳。

要想在 Jupyter notebook 中使用 matplotlib，需要告诉 Jupyter 获取 matplotlib 生成的所有图形，并将其嵌入 notebook 中。为此，需要计算：

%matplotlib inline

译注：要想执行成功，需要先pip install matplotlib。

运行这个指令可能要花个几秒钟，但是在 notebook 中需要执行一次即可。接下来，我们来绘制一个图形，看看具体的集成效果：

import matplotlib.pyplot as plt

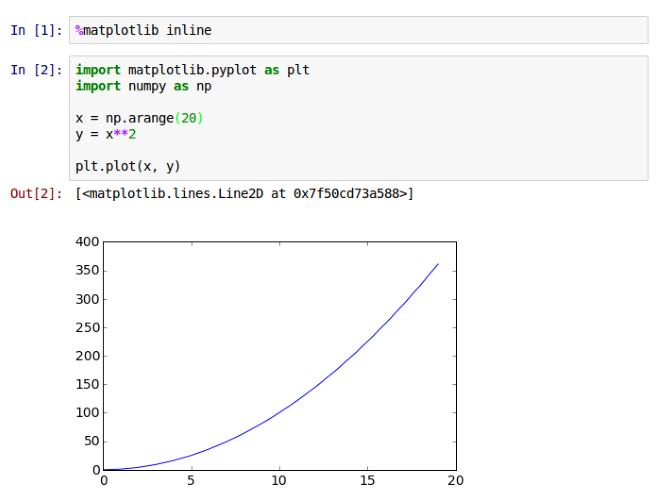
import numpy as np

x = np.arange(20)

y = x\*\*2

plt.plot(x, y)

上面的代码将绘制方程式 y=x^2 。计算单元格后，会得到如下图形：



我们看到，绘制出的图形直接添加在了 notebook 中，就在代码的下面。我们可以之后修改代码，重新计算，这时图形也会动态更新。这是每个数据科学家都想要的一个特性：将代码和图片放在同一个文件中，清楚地看出每段代码的效果。

**八、非本地内核**

我们可以非常容易地在一台电脑上启动 Jupyter，而且支持多人通过网络连接同一个 Jupyter 实例。在上一篇文章中，你有没有注意启动 Jupyter 时出现过这样一段话：

The IPython Notebook is running at: http://localhost:8888/

这意味着，你的 notebook 是本地运行的，可以在浏览器上打开 http://localhost:8888/ ，从而访问 notebook。你也可以修改下配置，让该 notebook 可以被公开访问。这样，任何知道 notebook 地址的人都可以连接到 notebook 进行远程修改。

**九、JupyterLab——Jupyter Notebooks 的进化**

JupyterLab 是今年二月份推出的，被认为是 Jupyter Notebooks 的进一步发展。其支持更加灵活和更加强大的项目操作方式，但具有和 Jupyter Notebooks 一样的组件。JupyterLab 环境与 Jupyter Notebooks 环境完全一样，但具有生产力更高的体验。

JupyterLab 让你能在一个窗口中排布你的笔记本、终端、文本文件和输出结果工作区！你只需拖放你需要的单元即可。你也可以编辑 Markdown、CSV 和 JSON 等常用文件格式并实时预览修改所造成的影响。

如果你想在你的机器上试用 JupyterLab，可查看安装说明：http://jupyterlab.readthedocs.io/en/stable/getting\_started/installation.html。JupyterLab 的开发者的长期目标是最终替代 Jupyter Notebooks。但目前来看这还需要一些时间。

**Ps**:这个 GitHub 库包含了一些有趣迷人的 Jupyter Notebooks：https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks