

Jupyter Notebooks 是数据科学/机器学习社区内一款非常流行的工具。Analytics Vidhya 的 Pranav Dar 近日发表了一篇上手使用 Jupyter Notebooks 的指南，从安装到基本功能进行了简洁清晰的介绍。

## 引言

应该使用哪个 IDE/环境/工具？这是人们在做数据科学项目时最常问的问题之一。可以想到，我们不乏可用的选择——从 R Studio 或 PyCharm 等语言特定的 IDE 到 Sublime Text 或 Atom 等编辑器——选择太多可能会让初学者难以下手。

如果说有什么每个数据科学家都应该使用或必须了解的工具，那非 Jupyter Notebooks 莫属了（之前也被称为 iPython 笔记本）。Jupyter Notebooks 很强大，功能多，可共享，并且提供了在同一环境中执行数据可视化的功能。

Jupyter Notebooks 允许数据科学家创建和共享他们的文档，从代码到全面的报告都可以。它们能帮助数据科学家简化工作流程，实现更高的生产力和更便捷的协作。由于这些以及你将在下面看到的原因，Jupyter Notebooks 成了数据科学家最常用的工具之一。



在本文中，我们将介绍 Jupyter Notebooks 并深入了解它的功能和优势。

读完本文之后，你就知道你应该为你的机器学习项目使用 Jupyter Notebooks 的原因了。你也会知道为什么人们会认为 Jupyter Notebooks 比该领域内的其它标准工具更好。

准备好学习了吗？现在开始吧！

## 目录

Jupyter Notebooks 是什么？如何安装 Jupyter Notebooks？开始上手！使用神奇的功能不只限于 Python——在 Jupyter Notebooks 中使用 R、Julia 和 JavaScript Jupyter Notebooks 中的交互式仪表盘——何乐不为？键盘快捷键——节省时间且更有生产力！有用的 Jupyter Notebooks 扩展保存和共享你的笔记本 JupyterLab——Jupyter Notebooks 的进化最佳实践和技巧

## 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。这可以让用户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。

正如你将在本文中看到的那样，这些笔记本非常灵活，能为数据科学家提供强大的交互能力和工具。它们甚至允许你运行 Python 之外的其它语言，比如 R、SQL 等。因为它们比单纯的 IDE 平台更具交互性，所以它们被广泛用于以更具教学性的方式展示代码。

## 如何安装 Jupyter Notebooks?

你可能已经猜到了，你首先需要在你的机器上安装 Python。Python 2.7 或 Python 3.3（或更新版本）都可以。

### 1.Anaconda

对新用户而言，一般的共识是你应该使用 Anaconda 发行版来安装 Python 和 Jupyter Notebooks。Anaconda 会同时安装这两个工具，并且还包含相当多数据科学和机器学习社区常用的软件包。你可以在[这里](https://www.anaconda.com/download)下载最新版的 Anaconda：

### 2.pip 方法

如果你因为某些原因不愿意使用 Anaconda，那么你需要确保你的机器运行着最新版的 pip。该怎么做呢？如果你已经安装了 Python，那么就已经有 pip 了。你可以使用以下代码升级到最新版的 pip：

```
#Linux and OSX
pip install -U pip setuptools
#Windows
python -m pip install -U pip setuptools
```

pip 安装好之后，继续安装 Jupyter：

```
#For Python2
pip install jupyter
#For Python3
pip3 install jupyter
```

你可以在[这里](https://jupyter.readthedocs.io/en/latest/install.html)查看官方的 Jupyter 安装文档：

## 开始上手！

现在你已经知道这些笔记本是什么以及如何将其安装到你的机器上了。现在开始使用吧！

要运行你的 Jupyter Notebooks，只需在命令行输入以下命令即可！

```
jupyter notebook
```

完成之后，Jupyter Notebooks 就会在你的默认网络浏览器打开，地址是：

```
http://localhost:8888/tree
```

在某些情况下，它可能不会自动打开。而是会在终端/命令行生成一个 URL，并带有令牌密钥提示。你需要将包含这个令牌密钥在内的整个 URL 都复制并粘贴到你的浏览器，然后才能打开一个笔记本。

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。

要打开一个新的 Jupyter 笔记本，点击页面右侧的「New」选项。你在这里会看到 4 个需要选择的选项：

Python 3Text FileFolderTerminal

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。

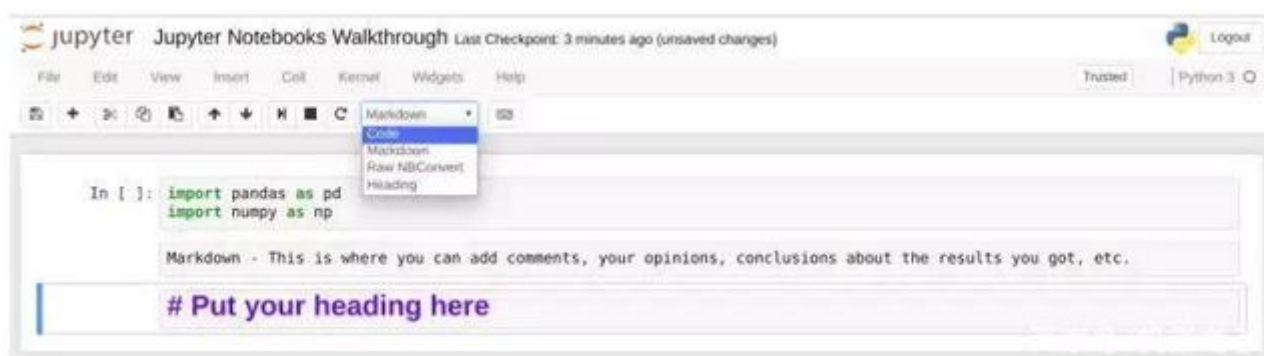
选择 Folder 选项时，你会创建一个新的文件夹，你可以在其中放入文件，重命名或删除它。各种操作都可以。

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！

但在本文中，我们重点关注的是笔记本，所以我们从 New 选项中选择 Python 3。你会看到下面的屏幕：



然后你可以从导入最常见的 Python 库开始：pandas 和 numpy。在代码上面的菜单中，你有一些操作各个单元的选项：添加、编辑、剪切、向上和向下移动单元、运行单元内的代码、停止代码、保存工作以及重启 kernel。



在上图所示的下拉菜单中，你还有 4 个选项：

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。

## 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
```

你会看到列出了很多选择，你甚至可能能认出其中一些！%clear、%autosave、%debug 和 %mkdir 等功能你以前肯定见过。现在，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)
```

我建议你运行这些代码，亲自看看它们的不同之处！

## 不只限于 Python——在 Jupyter Notebooks 中使用 R、Julia 和 JavaScript

神奇之处可不止这点。你甚至能在你的笔记本中使用其它语言，比如 R、Julia、JavaScript 等。我个人很喜欢 R 中的 ggplot2 软件包，所以使用它来进行探索性的数据分析具有很大很大的优势。

要在 Jupyter 中启用 R，你需要 IRKernel。这是针对 R 的专用 kernel，你可以在 GitHub 上获取。这需要 8 个步骤，已经有详细的解释了，另外还有截图指导，参阅：

<https://discuss.analyticsvidhya.com/t/how-to-run-r-on-jupyter-ipython-notebooks/5512>

如果你是一位 Julia 用户，你也能在 Jupyter Notebooks 中使用 Julia！你可以查看这篇为 Julia 用户学习数据科学而编写的全面介绍文章，其中有一个章节就是关于如何在 Jupyter 环境中使用 Julia：

<https://www.analyticsvidhya.com/blog/2017/10/comprehensive-tutorial-learn-data-science-julia-from-scratch/>

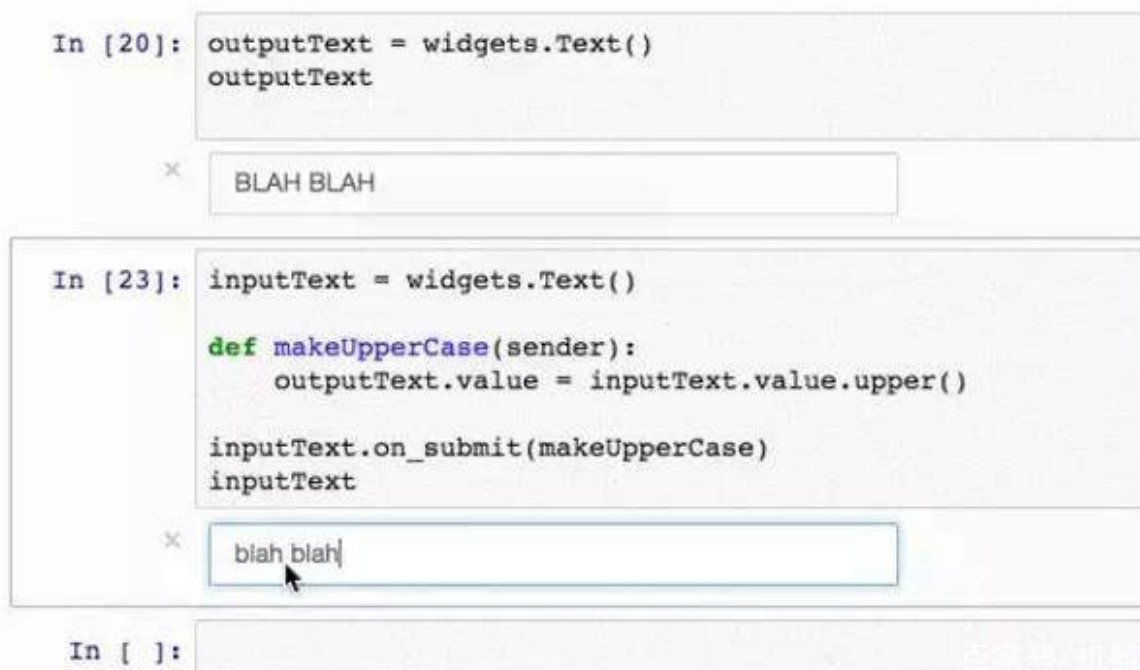
如果你更偏爱 JavaScript，那么我推荐使用 IJavascript kernel。这个 GitHub 库包含了在不同操作系统上安装这个 kernel 的各个步骤：<https://github.com/n-riesco/ijavascript>。注意，在使用它之前，你必需要先安装好 Node.js 和 npm。

## Jupyter Notebooks 中的交互式仪表盘——何乐不为？

在你考虑添加小部件之前，你需要导入 widgets 软件包：

```
from ipywidgets import widgets
```

小部件的基本类型有典型的文本输入小部件、基于输入的小部件和按钮小部件。下面的例子来自 Dominodatalab，给出了交互式小部件的一些外观：



关于小部件的完整指南，请参阅：<https://blog.dominodatalab.com/interactive-dashboards-in-jupyter/>

## 键盘快捷键——节省时间且更有生产力！

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。

下面是我们手动选择的一些对你的上手会有莫大帮助的快捷方式。我强烈建议你在阅读本文时逐一尝试一下。未来你会离不开它们的！

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

你可以分别使用 Esc 和 Enter 在命令模式和编辑模式之间跳跃。现在就试试看吧！

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。

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。如之前提到的，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。

### 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：

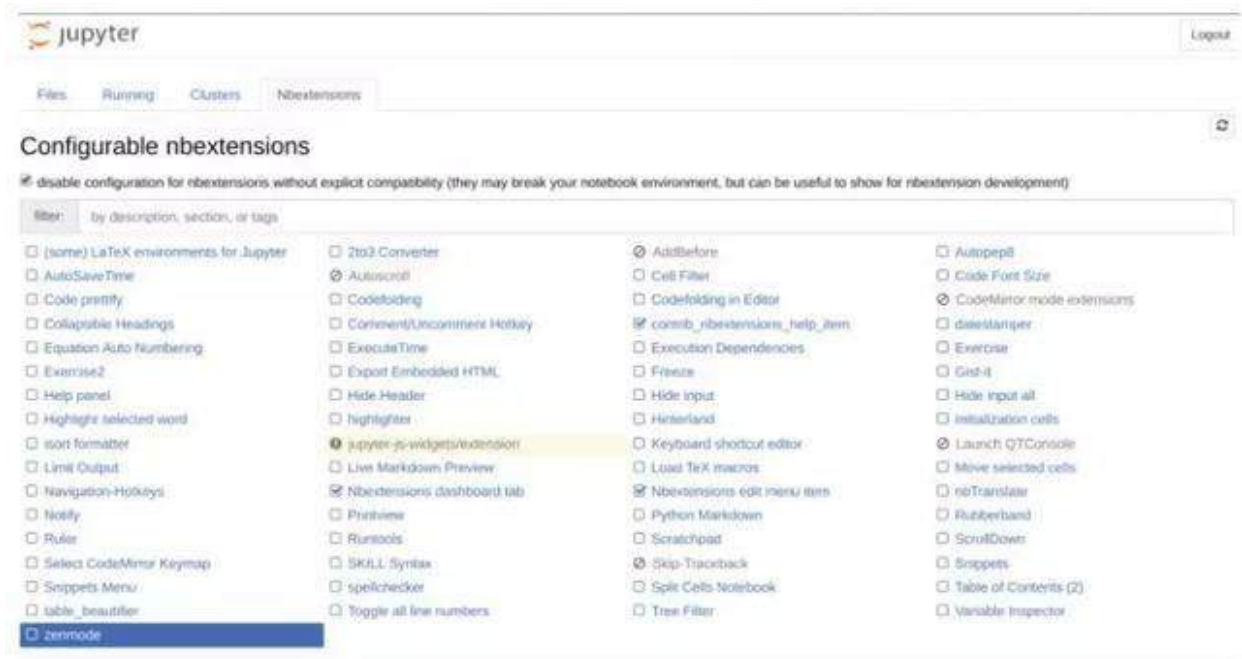
```
pip install jupyter_contrib_nbextensions
```

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。





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：

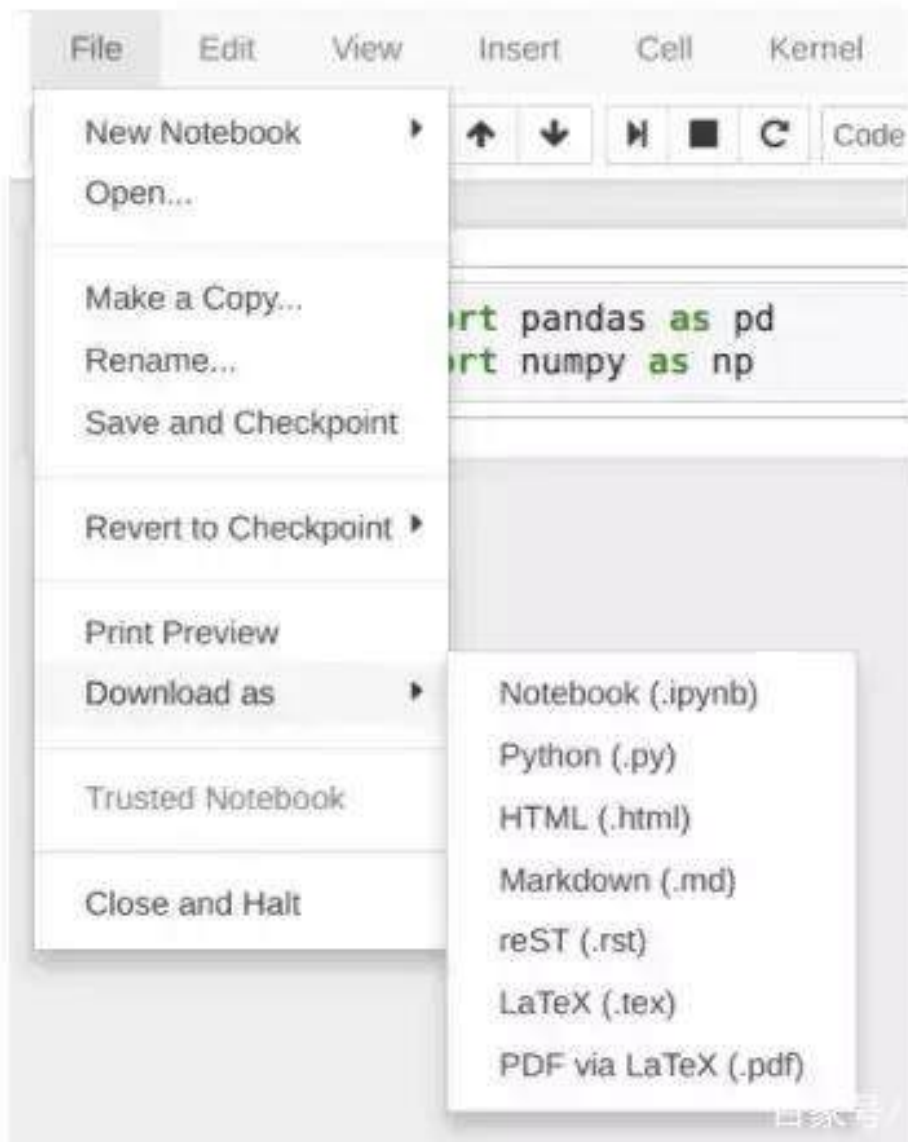
Code prettify：它能重新调整代码块内容的格式并进行美化。Printview：这个扩展会添加一个工具栏按钮，可为当前笔记本调用 jupyter nbconvert，并可以选择是否在新的浏览器标签页显示转换后的文件。Scratchpad：这会添加一个暂存单元，让你可以无需修改笔记本就能运行你的代码。当你想实验你的代码但不想改动你的实时笔记本时，这会是一个非常方便的扩展。Table of Contents (2)：这个很棒的扩展可以收集你的笔记本中的所有标题，并将它们显示在一个浮动窗口中。

这只是少量几个扩展。我强烈建议你查看完整扩展列表并实验它们的功能。

### 保存和共享你的笔记本

这是 Jupyter Notebooks 最重要且最出色的功能之一。当我必须写一篇博客文章时，我的代码和评论都会在一个 Jupyter 文件中，我需要首先将它们转换成另一个格式。记住这些笔记本是 json 格式的，这在进行共享时不会很有帮助。我总不能在电子邮件和博客上贴上不同单元块，对不对？

进入「Files」菜单，你会看到「Download As」选项：



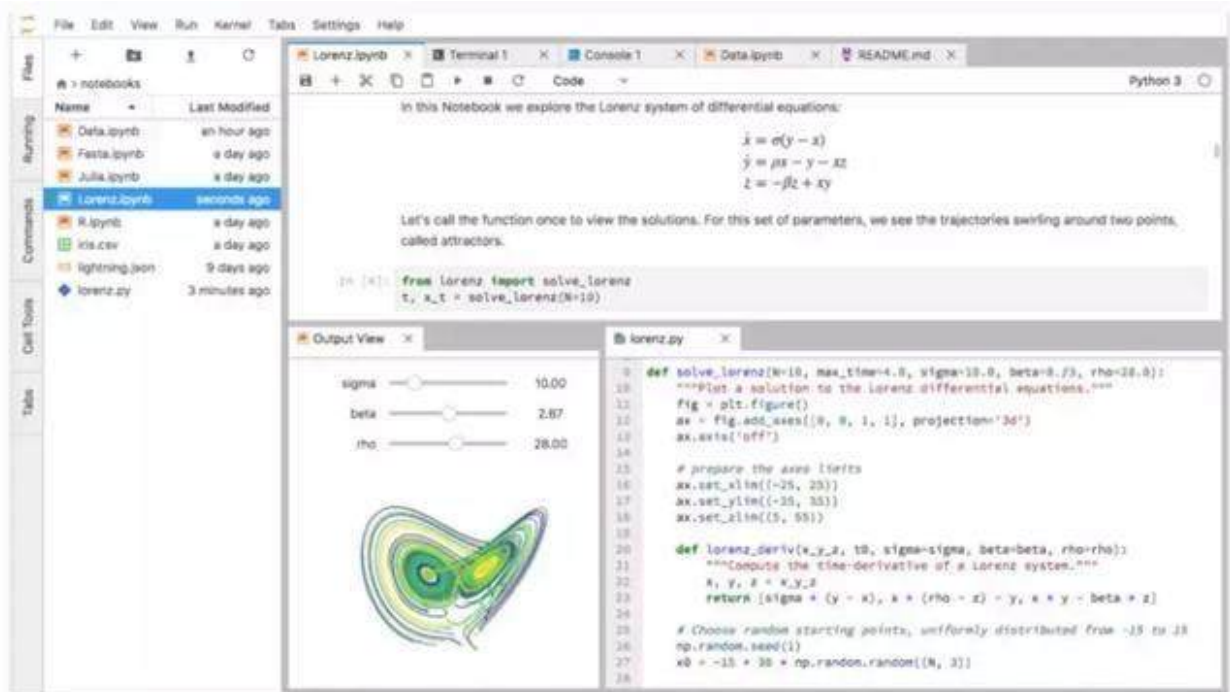
你可以用 7 种可选格式保存你的笔记本。其中最常用的是 .ipynb 文件和 .html 文件。使用 .ipynb 文件可让其他人将你的代码复制到他们的机器上，使用 .html 文件能以网页格式打开（当你需要保存嵌入在笔记本中的图片时会很方便）。

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。

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。

## 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](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。

## 最佳实践

尽管独自工作可能很有趣，但大多数时候你都是团队的一员。在这种情况下，遵循指导原则和最佳实践是很重要的，能确保你的代码和 Jupyter Notebooks 都有适当的注释，以便与你的团队成员保持一致。这里我列出了一些最佳实践指标，你在 Jupyter Notebooks 上工作时一定要遵守：

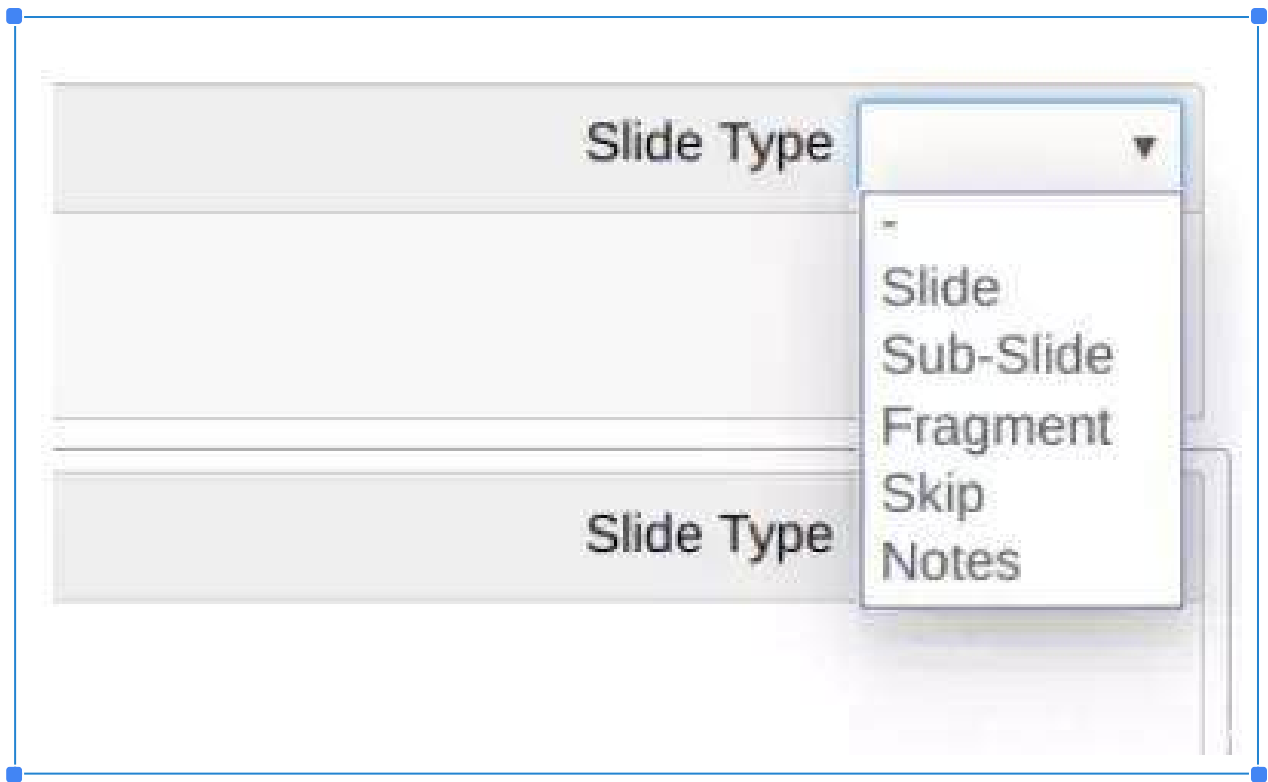
对任何程序员而言都是最重要的事情之一——总是确保你为你的代码添加了适当的注释！确保你的代码有所需的文档。考虑一个命名方案并贯彻始终。这能让其他人更容易遵循。不管你的代码需要什么库，都在你的笔记本起始处导入它们。（并在旁边添加注释说明你载入它们的目的）确保你的代码有适当的行距。你不要将你的循环和函数放在同一行——否则如果后面要引用它们，会让人抓狂的！有时候你的文件中有非常大量的代码。看看能不能将你认为不重要的某些代码隐藏起来，之后再引用。这能让你的笔记本看起来整洁清晰，这是非常可贵的。查看这个在 matplotlib 上的笔记本，看看可以如何简练地进行呈现：

<http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-4-Matplotlib.ipynb>

另一个额外技巧！在你想创建一个演示文稿时，你可能首先想到的工具是 PowerPoint 和 Google Slides。其实你的 Jupyter Notebooks 也能创建幻灯片！还记得我说过 Jupyter Notebooks 很灵活吗？我可没有夸大其辞。

要将你的笔记本转换成幻灯片，进入「View→Cell Toolbar」，然后点击「Slideshow」。现在，每个代码块右边都显示了一个「Slide Type」下拉选项。你能看到下列的 5 个选项：





你最好试试每个选项，以便更好地理解它们。这能改变你展示代码的方式！

## 结语

注意，这篇文章远没有完全覆盖 Jupyter Notebooks 的功能。还有很多东西要在你使用得更多之后才会用到。功能虽多，但关键在于实践出真知。

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>

这份指南只是你的数据科学旅程的起点，我很高兴能与你一起前行！