# 左手代码，右手写作：你必须会的Jupyter Notebook

在数据分析的道路上，你一定曾有过为新发现而激动不已的时刻，此时你急于将自己的发现告诉大家，却遇到了这样的问题：如何将我的分析过程清晰地表述出来呢？

为了能与同行们有效沟通，你需要重现整个分析过程，并将说明文字、代码、图表、公式、结论都整合在一个文档中。显然传统的文本编辑工具并不能满足这一需求，所以这儿隆重推荐一款神器 [Jupyter Notebook](http://jupyter.org/)，不仅能在文档中执行代码，还能以网页形式分享。

下图简单展示了Jupyter Notebook 文档的样式，更多示例可在 [nbviewer](http://nbviewer.jupyter.org/) 中找到。

Jupyter Notebook 文档示例

本文将按如下内容来展开：

一、Jupyter Notebook 介绍

* 文学编程
* 历史传承
* 优点

二、Jupyter Notebook 使用入门

* 安装
* 运行
* 界面

三、Jupyter Notebook 进阶功能

* 数学公式编辑
* 幻灯片制作
* 魔术关键字

## 一、Jupyter Notebook 介绍

### 文学编程

在介绍 Jupyter Notebook 之前，让我们先来看一个概念：[**文学编程**](https://zh.m.wikipedia.org/wiki/%E6%96%87%E5%AD%A6%E7%BC%96%E7%A8%8B) ( **Literate programming** )，这是由 Donald Knuth 提出的[编程方法](http://www.literateprogramming.com)。传统的结构化编程，人们需要按计算机的逻辑顺序来编写代码；与此相反，文学编程则可以让人们按照自己的思维逻辑来开发程序。

简单来说，文学编程的读者不是机器，而是人。 我们从写出让机器读懂的代码，过渡到向人们解说如何让机器实现我们的想法，其中除了代码，更多的是叙述性的文字、图表等内容。这么一看，这不正是数据分析人员所需要的编码风格么？不仅要当好一个程序员，还得当好一个作家。那么 Jupyter Notebook 就是不可或缺的一款集编程和写作于一体的效率工具。

### 历史传承

也许说到 Jupyter 你会觉得陌生，但想必你或多或少听过鼎鼎大名的 [IPython](https://ipython.org)。其实Jupyter 脱胎于 IPython 项目，IPython 顾名思义，是专注于 Python 的项目，但随着项目发展壮大，已经不仅仅局限于 Python 这一种编程语言了。Jupyter 的名字就很好地释义了这一发展过程，它是 **Ju**lia、**Pyt**hon 以及 **R** 语言的组合，字形相近于木星（Jupiter），而且现在支持的语言也远超这三种了。

### 优点

以下列举了 Jupyter Notebook 的众多优点：

* 极其适合数据分析  
  想象一下如下混乱的场景：你在终端中运行程序，可视化结果却显示在另一个窗口中，包含函数和类的脚本存在其他文档中，更可恶的是你还需另外写一份说明文档来解释程序如何执行以及结果如何。此时 Jupyter Notebook 从天而降，将所有内容收归一处，你是不是顿觉灵台清明，思路更加清晰了呢？
* 支持多语言  
  也许你习惯使用 R 语言来做数据分析，或者是想用学术界常用的 MATLAB 和 Mathematica，这些都不成问题，只要安装相对应的核（kernel）即可。[这里](https://github.com/jupyter/jupyter/wiki/Jupyter-kernels)列出了 Jupyter 支持的所有语言，供您参考。
* 分享便捷  
  支持以网页的形式分享，GitHub 中天然支持 Notebook 展示，也可以通过 [nbviewer](http://nbviewer.jupyter.org) 分享你的文档。当然也支持导出成 HTML、Markdown 、PDF 等多种格式的文档。
* 远程运行  
  在任何地点都可以通过网络链接远程服务器来实现运算，这里给出一个远程运行的[例子](https://try.jupyter.org)，可以体验一下 Jupyter Notebook。
* 交互式展现  
  不仅可以输出图片、视频、数学公式，甚至可以呈现一些互动的可视化内容，比如可以缩放的地图或者是可以旋转的三维模型。这就需要交互式插件（Interactive widgets）来支持，更多内容请参考[这里](http://jupyter.org/widgets.html)。

## 二、Jupyter Notebook 使用入门

### 安装

对于初学者，最简单的方法是安装 Anaconda，因为它自带了 Jupyter Notebook。如果想进一步了解 Anaconda 的使用方法，可参考我的上一篇文章[《致Python初学者们 - Anaconda入门使用指南》](https://www.jianshu.com/p/169403f7e40c)。

对于有经验的同学，自然使用命令行最为便捷。

如果在 conda 环境中，可以使用如下命令安装：

conda install jupyter notebook

或者直接通过 pip 安装：

pip install jupyter notebook

更多的安装说明请参考[官网](http://jupyter.org/install.html)。

### 运行

如果使用 Anaconda，可以在其 Navigator 图形界面中点击打开 Notebook。但是最快捷的方法还是在命令行中输入：

jupyter notebook

此时你的 Web 浏览器被自动打开，显示文件目录。通过点击右上角的 new 创建新文档。

进入 notebook

而关闭 notebook 文档可以通过选择文件名前的复选框后，点击 Shutdown 按钮实现。如果要关闭整个服务，则在原来的终端中按 Control + C 两次。

退出 notebook

### 界面

Notebook 文档是由一系列单元（Cell）构成，主要有两种形式的单元：

* 代码单元：这里是你编写代码的地方，通过按 Shift + Enter 运行代码，其结果显示在本单元下方。代码单元左边有 In [1]: 这样的序列标记，方便人们查看代码的执行次序。
* Markdown 单元：在这里对文本进行编辑，采用 markdown 的语法规范，可以设置文本格式、插入链接、图片甚至数学公式。同样使用 Shift + Enter 运行 markdown 单元来显示格式化的文本。

Jupyter Notebook 编辑界面

类似于 Linux 的 Vim 编辑器，在 notebook 中也有两种模式：

* 编辑模式：编辑文本和代码。选中单元并按 Enter 键进入编辑模式，此时单元左侧显示绿色竖线。
* 命令模式：用于执行键盘输入的快捷命令。通过 Esc 键进入命令模式，此时单元左侧显示蓝色竖线。

如果要使用快捷键，首先按 Esc 键进入命令模式，然后按相应的键实现对文档的操作。比如切换成代码单元（Y）或 markdown 单元（M），或者在本单元的下方增加一单元（B）。查看所有快捷命令可以按H。

尽管一开始需要花费一些学习成本，但熟练使用快捷键将大大提高工作效率。试想你在键盘上十指如飞时，如果还需要通过鼠标来操作文档，是不是很影响思考速度呢？

## 三、Jupyter Notebook 进阶功能

### 数学公式编辑

如果你曾做过严肃的学术研究，一定对 [LaTeX](https://www.latex-project.org) 并不陌生，这简直是写科研论文的必备工具，不但能实现严格的文档排版，而且能编辑复杂的数学公式。在 Jupyter Notebook 的 markdown 单元中我们也可以使用 LaTeX 的语法来插入数学公式。

在文本行中插入数学公式，使用一对 $符号，比如质能方程 $E = mc^2$。如果要插入一个数学区块，则使用一对 $$ 符号。比如下面公式表示 z=x/y：

$$ z = \frac{x}{y} $$

如何在 notebook 中使用 LaTeX，可进一步参考 [A Primer on Using LaTeX in Jupyter Notebooks](http://data-blog.udacity.com/posts/2016/10/latex-primer/) 这篇文章。

### 幻灯片制作

既然Jupyter Notebook 擅长展示数据分析的过程，除了通过网页形式分享外，当然也可以将其制作成幻灯片的形式。这里有一个幻灯片[示例](http://www.slideviper.oquanta.info/tutorial/slideshow_tutorial_slides.html#/)供参考，其制作风格简洁明晰。

那么如何用 Jupyter Notebook 制作幻灯片呢？首先在 notebook 的菜单栏选择 View > Cell Toolbar > Slideshow，这时在文档的每个单元右上角显示了 Slide Type 的选项。通过设置不同的类型，来控制幻灯片的格式。有如下5中类型：

* Slide：主页面，通过按左右方向键进行切换。
* Sub-Slide：副页面，通过按上下方向键进行切换。
* Fragment：一开始是隐藏的，按空格键或方向键后显示，实现动态效果。
* Skip：在幻灯片中不显示的单元。
* Notes：作为演讲者的备忘笔记，也不在幻灯片中显示。

Jupyter Notebook 幻灯片设置

当编写好了幻灯片形式的 notebook，如何来演示呢？这时需要使用 nbconvert：

jupyter nbconvert notebook.ipynb --to slides --post serve

在命令行中敲入上述代码后，浏览器会自动打开相应的幻灯片。

### 魔术关键字

魔术关键字（magic keywords），正如其名，是用于控制 notebook 的特殊的命令。它们运行在代码单元中，以 % 或者 %% 开头，前者控制一行，后者控制整个单元。

比如，要得到代码运行的时间，则可以使用 %timeit；如果要在文档中显示 matplotlib 包生成的图形，则使用 % matplotlib inline；如果要做代码调试，则使用 %pdb。但注意这些命令大多是在Python kernel 中适用的，其他 kernel 大多不适用。有许许多多的魔术关键字可以使用，更详细的清单请参考 [Built-in magic commands](http://ipython.readthedocs.io/en/stable/interactive/magics.html) 。

## 相关资源汇总

本文给出了许多扩展链接供参考，这里汇总如下，供您参考：

1. [Jupyter Notebook 官网](http://jupyter.org/)
2. [文学编程 Literate programming](http://www.literateprogramming.com)
3. [IPython 官网](https://ipython.org)
4. [LaTeX官网](https://www.latex-project.org)
5. [LaTeX 语法：A Primer on Using LaTeX in Jupyter Notebooks](http://data-blog.udacity.com/posts/2016/10/latex-primer/)
6. [魔术关键字：Built-in magic commands](http://ipython.readthedocs.io/en/stable/interactive/magics.html)

* 如果你现在迫不及待地想试一试 Jupyter Notebook了，请参考系列文章[Python数据分析的起手式](https://www.jianshu.com/p/39eb230e6f15)