

Jupyter Notebook 快速入门（下）



9 个月前

4671



这两天分享的 Jupyter Notebook 快速入门文章，比较基础，只涉及了基本功能介绍和演示。后面再找机会分享其他高级用法。

从上一篇文章中，我们发现 Jupyter notebook 的基本功能就可以支持完成许多事情。不过它背后的功能和选项并不止于此。本文将进一步介绍一些有用的操作。

单元格操作

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]**。

推荐阅读

热门文章

随机

20天持续压测，云存储性能哪家更强？

国内公有云大幅降价后，首份一手云计算产品评测报告

Python进阶、求职必看的前辈经验分享

硅谷码农用Python写了个机器人，租到了让女友满意的房子

使用 Python 进行科学计算：NumPy入门

十分钟入门Matplotlib

从零开发一个小游戏：PyGame 入门

好用！在 Notebook 中使用 Sublime Text 快捷键

十张GIFs让你弄懂递归等概念

热门标签

IDE PyCon

编译 Flask

Codewars

Postgresql Django

Docker Git

程序员 开发库

漫画 编码风格

记住这些操作，它们可以帮助你节省许多时间。

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：

```

```

计算该单元格之后，会出现这样的结果：



In []:

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：

$$\int_0^{+\infty} x^2 dx$$

In []:

导出功能

notebook 还有一个强大的特性，就是其导出功能。可以将 notebook 导出为多种格式：

- HTML
- Markdown
- ReST
- PDF（通过 LaTeX）
- Raw Python

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

经典书籍

《Think Python 2ed》最新中文

扫码关注编程派



腾讯云
云服务器 **1核1G** 低至 **1.**
稳定 高速 安全 | 送50G硬盘
立即购买>

可以导出为 ReST 格式，作为软件库的文档。

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：

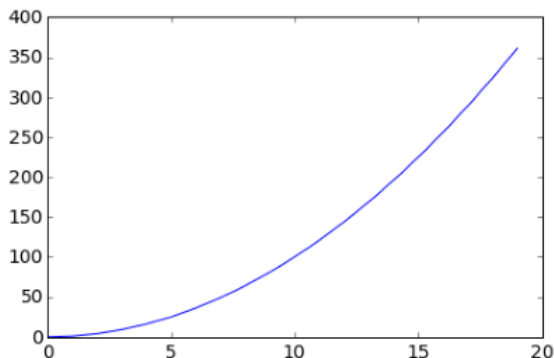
```
In [1]: %matplotlib inline
```

```
In [2]: import matplotlib.pyplot as plt
import numpy as np

x = np.arange(20)
y = x**2

plt.plot(x, y)
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x7f50cd73a588>]
```



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。

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。

结语

从这两篇快速入门介绍中，我们可以看到：**Jupyter notebook** 是一个非常强大的工具，可以创建漂亮的交互式文档，制作教学材料，等等。建议你马上开始使用 **Jupyter notebook**，探索更多 **notebook** 的强大功能。

[点此查看原文链接](#)
