

02 | Jupyter Notebook为什么是现代Python的必学技术？

2019-05-13 景霄

Python核心技术与实战

[进入课程 >](#)



讲述：冯永吉

时长 08:50 大小 8.10M

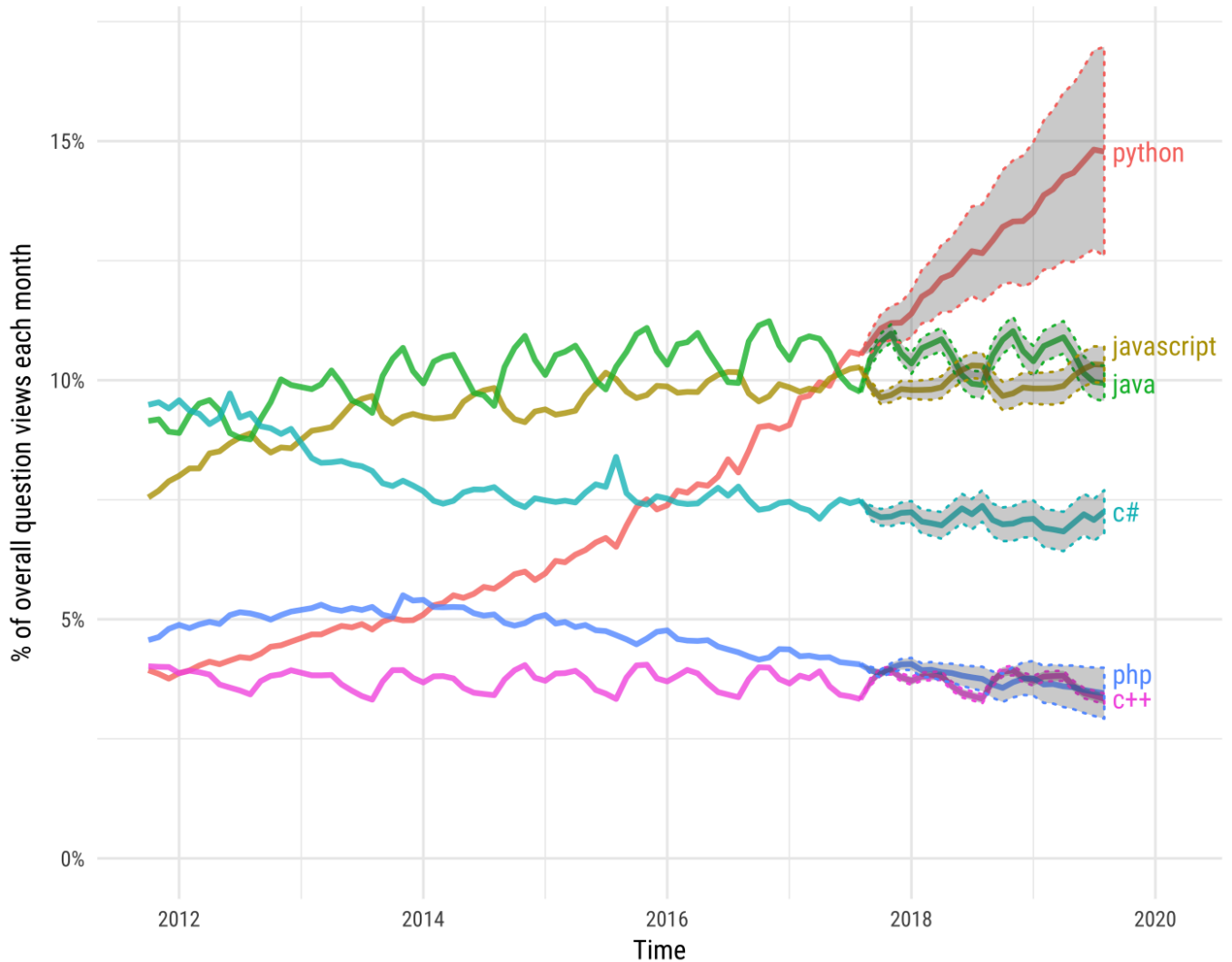


你好，我是景霄。

Stack Overflow 曾在 2017 年底，发布了在该站上各种语言的提问流量。其中，Python 已经超过了 JavaScript 成为了流量最高的语言，预测在 2020 年前会远远甩开 JavaScript。

Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.



可能你已经知道，Python 在 14 年后的“崛起”，得益于机器学习和数学统计应用的兴起。那为什么 Python 如此适合数学统计和机器学习呢？作为“老司机”的我可以肯定地告诉你，Jupyter Notebook (<https://jupyter.org/>) 功不可没。

毫不夸张地说，根据我对 Facebook 等硅谷一线大厂的了解，一个 Python 工程师如果现在还会使用 Jupyter Notebook 的话，可能就真的太落伍了。

磨刀不误砍柴工，高效的工具让我们的编程事半功倍。这一节课，我就来带你学习一下 Jupyter Notebook，为后面的 Python 学习打下必备基础。

什么是 Jupyter Notebook?

说了这么多，到底什么是 Jupyter Notebook? 按照 Jupyter 创始人 Fernando Pérez 的说法，他最初的梦想是做一个综合 Ju (Julia)、Py (Python) 和 R 三种科学运算语言的计算工具平台，所以将其命名为 Ju-Py-te-R。发展到现在，Jupyter 已经成为一个几乎支

持所有语言，能够把**软件代码、计算输出、解释文档、多媒体资源整合**在一起的多功能科学运算平台。

英文里说一图胜千言（A picture is worth a thousand words）。看下面这个图片，你就明白什么是 Jupyter Notebook 了。

▼ Rich, interactive outputs

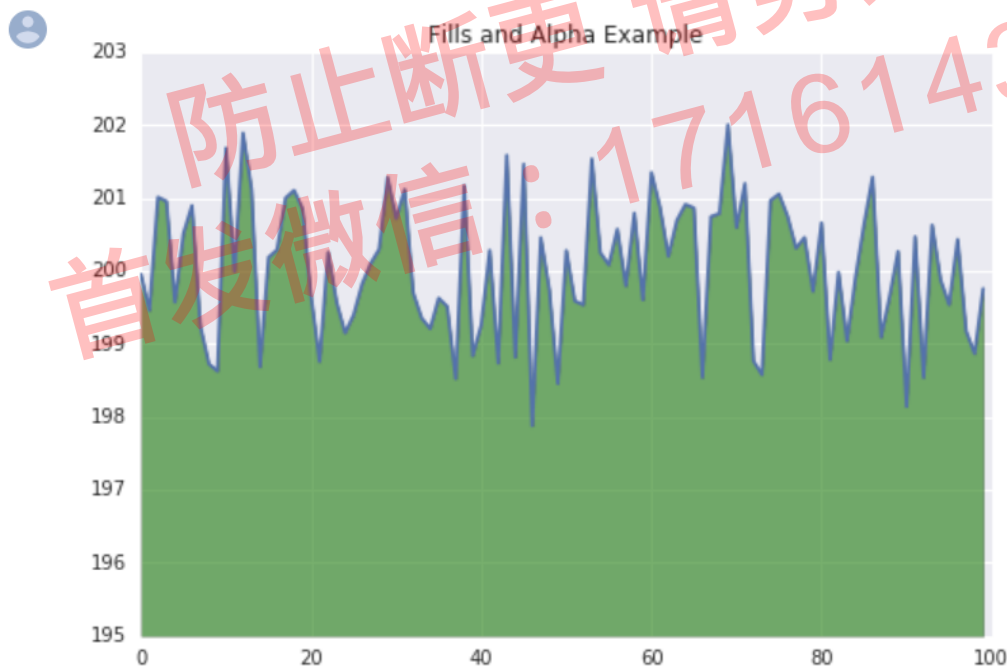
Until now all of the generated outputs have been text, but they can be more interesting, like the chart below.

```
[ ] import numpy as np
    from matplotlib import pyplot as plt

    ys = 200 + np.random.randn(100)
    x = [x for x in range(len(ys))]

    plt.plot(x, ys, '-')
    plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

    plt.title("Fills and Alpha Example")
    plt.show()
```



你在一个框框中直接输入代码，运行，它立马就在下面给你输出。怎么样，是不是很酷？你可能会纳闷儿，这样一个看起来“华而不实”的玩意儿，真的就成了 Python 社区的颠覆者吗？说实话放在几年前我也是不信的。所以 Jupyter Notebook 的影响究竟有多大呢？

Jupyter Notebook 的影响力

我们衡量一个技术的影响力，或者说要用自己的技术去影响世界时，必定绕不开这个技术对教育界的影响力。

就拿微软的 Word 文本处理系统来说吧。从纯技术角度来讲，Word 的单机设计理念早已落后时代 20 年。但以 Google Doc 为代表的在线文档系统，却并没有像想象中那样，实现对 Word 的降维打击。

直观的原因是用户习惯，使用 Word 修改文档，那就来回发几十遍呗，用着也还可以。但更深刻来想，之所以养成这样的用户习惯，是因为我们的教育根源。教育系统从娃娃抓起，用小学中学大学十几年的时间，训练了用户 Word 的使用习惯。到工作中，老员工又会带着新员工继续使用 Word，如此行程技术影响力生生不息的正向反馈。

回到我们今天的主题，我们来看 Jupyter Notebook。从 2017 年开始，已有大量的北美顶尖计算机课程，开始完全使用 Jupyter Notebook 作为工具。比如李飞飞的 CS231N《计算机视觉与神经网络》课程，在 16 年时作业还是命令行 Python 的形式，但是 17 年的作业就全部在 Jupyter Notebook 上完成了。再如 UC Berkeley 的《数据科学基础》课程，从 17 年起，所有作业也全部用 Jupyter Notebook 完成。

而 Jupyter Notebook 在工业界的影响力更甚。在 Facebook，虽然大规模的后台开发仍然借助于功能齐全的 IDE，但是几乎所有的中小型程序，比如内部的一些线下分析软件，机器学习模块的训练都是借助于 Jupyter Notebook 完成的。据我了解，在别的硅谷一线大厂，例如 Google 的 AI Research 部门 Google Brain，也是清一色地全部使用 Jupyter Notebook，虽然用的是他们自己的改进定制版，叫 Google Colab。

看到这里，相信你已经认可了 Jupyter Notebook 现如今的江湖地位。不过，说到技术的选择，有些人会说，这个技术流行，我们应该用；有些人认为，阿里已经在用这个技术了，这就是未来，我们也要用等等。不得不说，这些都是片面的认知。不管是阿里还是 Facebook 用的技术，其实不一定适用你的应用场景。

我经常鼓励技术同行，对于技术选择要有独立的思考，不要人云亦云。最起码你要去思考，Facebook 为什么选择这个技术？这个技术解决了哪些问题？Facebook 为什么不选择别的技术？有哪些局限？单从选择结果而言，Facebook 选择的技术很可能是因为它有几百个产品线，几万个工程师。而同样的技术，在一个十人的团队里，反而成了拖累。

在这里，我不想忽悠你任何技术，我想教会你的是辩证分析技术的思考方法。接下来，我们就来看看，Jupyter 究竟解决了哪些别人没有解决的问题。

Jupyter 的优点

整合所有的资源

在真正的软件开发中，上下文切换占用了大量的时间。什么意思呢？举个例子你就很好理解了，比如你需要切换窗口去看一些文档，再切换窗口去用另一个工具画图等等。这些都是影响生产效率的因素。

正如我前面提到的，Jupyter 通过把所有和软件编写有关的资源全部放在一个地方，解决了这个问题。当你打开一个 Jupyter Notebook 时，就已经可以看到相应的文档、图表、视频和相应的代码。这样，你就不需要切换窗口去找资料，只要看一个文件，就可以获得项目的所有信息。

交互性编程体验

在机器学习和数学统计领域，Python 编程的实验性特别强，经常出现的情况是，一小块代码需要重写 100 遍，比如为了尝试 100 种不同的方法，但别的代码都不想动。这一点和传统的 Python 开发有很大不同。如果是在传统的 Python 开发流程中，每一次实验都要把所有代码重新跑一遍，会花费开发者很多时间。特别是在像 Facebook 这样千万行级别的代码库里，即使整个公司的底层架构已经足够优化，真要重新跑一遍，也需要几分钟的时间。

而 Jupyter Notebook 引进了 Cell 的概念，每次实验可以只跑一小个 Cell 里的代码；并且，所见即所得，在代码下面立刻就可以看到结果。这样强的互动性，让 Python 研究员可以专注于问题本身，不被繁杂的工具链所累，不用在命令行直接切换，所有科研工作都能在 Jupyter 上完成。

零成本重现结果

同样在机器学习和数学统计领域，Python 的使用是非常短平快的。常见的场景是，我在论文里看到别人的方法效果很好，可是当我去重现时，却发现需要 pip 重新安装一堆依赖软件。这些准备工作可能会消耗你 80% 的时间，却并不是真正的生产力。

Jupyter Notebook 如何解决这个问题呢？

其实最初的 Jupyter Notebook 也是挺麻烦的，需要你先在本机上安装 IPython 引擎及其各种依赖软件。不过现在的技术趋势，则是彻底云端化了，例如 Jupyter 官方的 Binder 平台（介绍文档：<https://mybinder.readthedocs.io/en/latest/index.html>）和 Google 提供的 Google Colab 环境（介绍：<https://colab.research.google.com/notebooks/welcome.ipynb>）。它们让 Jupyter Notebook 变得和石墨文档、Google Doc 在线文档一样，在浏览器点开链接就能运行。

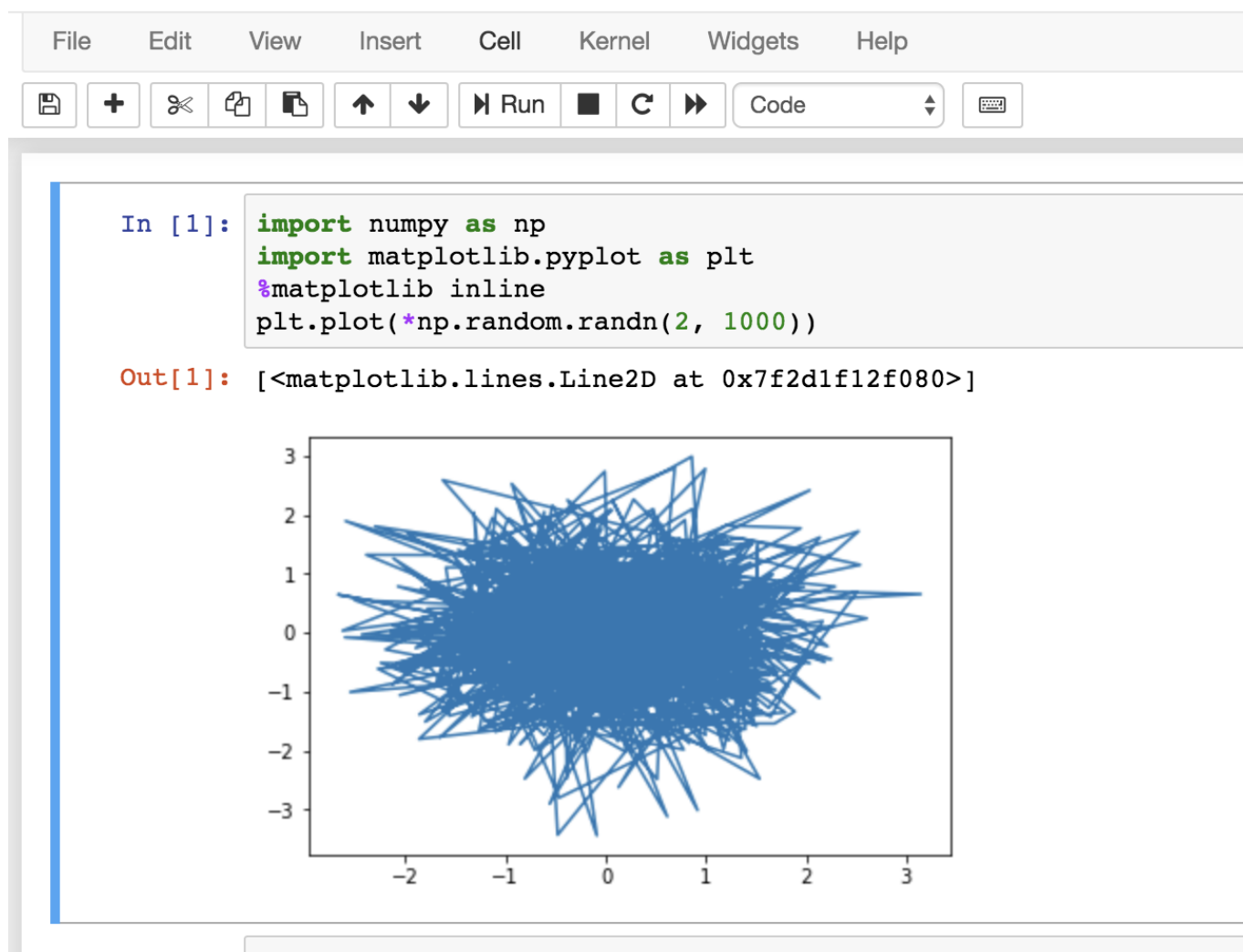
所以，现在当你用 Binder 打开一份 GitHub 上的 Jupyter Notebook 时，你不需要安装任何软件，直接在浏览器打开一份代码，就能在云端运行。

Jupyter Notebook 初体验

学习技术的最好方法就是用技术。不过，在今天的篇幅里，我不可能带你完全学会 Jupyter Notebook 的所有技巧。我想先带你直接感受一下，使用 Jupyter Notebook 的工作体验。

比如这样一个[GitHub 文件](#)。在[Binder](#)中，你只要输入其对应的 GitHub Repository 的名字或者 URL，就能在云端打开整个 Repository，选择你需要的[notebook](#)，你就能看到下图这个界面。

拼课微信：1718143661



每一个 Jupyter 的运行单元都包含了 In、Out 的 Cell。如图所示，你可以使用 Run 按钮，运行单独的一个 Cell。当然，你也可以在此基础上加以修改，或者新建一个 notebook，写成自己想要的程序。赶紧打开链接试一试吧！

另外，我还推荐下面这些 Jupyter Notebook，作为你实践的第一站。

第一个是 Jupyter 官方：https://mybinder.org/v2/gh/binder-examples/matplotlib-versions/mpl-v2.0/?filepath=matplotlib_versions_demo.ipynb

第二个是 Google Research 提供的 Colab 环境，尤其适合机器学习的实践应用：https://colab.research.google.com/notebooks/basic_features_overview.ipynb

如果你想在本地或者远程的机器上安装 Jupyter Notebook，可以参考下面的两个文档。

安装：<https://jupyter.org/install.html>

运行: <https://jupyter.readthedocs.io/en/latest/running.html#running>

总结

这节课，我为你介绍了 Jupyter Notebook，并告诉你它为什么日趋成为 Python 社区的必学技术。这主要是因为它的三大特点：**整合所有的资源、交互性编程体验和零成本重现结果**。但还是那句话，学习技术必须动手实操。这节课后，希望你能自己动手试一试 Jupyter Notebook，后面我们的一些课程代码，我也会用 Jupyter Notebook 的形式分享给你。

思考题

你尝试 Jupyter Notebook 了吗？欢迎在留言区和我分享你的使用体验。

 极客时间

Python 核心技术与实战

系统提升你的 Python 能力

景霄

Facebook 资深工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | 如何逐步突破，成为Python高手？

下一篇 03 | 列表和元组，到底用哪一个？



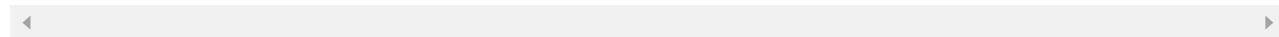
kyle

2019-05-13

👍 64

个人认为，工具类的介绍，一个工具没必要花一篇的时间去讲，文中提一下，读者自己去弄就行了~~

作者回复: 你的自学能力很强很好，但是我们的专栏设计要兼顾各种背景阶段的同学。



PyGeek

2019-05-13

👍 49

Jupyter本地的两种安装办法:

- 1.安装Anaconda。conda包含科学计算的几乎所有包，包含jupyter。
- 2.仅安装了Python。可以pip install ipython, jupyter。安装即可。

命令行启动jupyter: jupyter notebook



万能的芝麻...

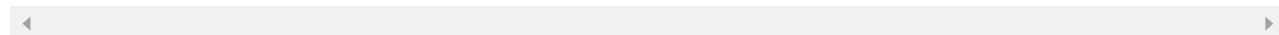
2019-05-14

👍 29

工欲善其事必先利其器，我觉得学任何东西，掌握优秀的工具都非常重要，很多时候可以达到先声夺人的效果。很多人说用Pycharm，又有多少人用出哪怕社区版10%的功能呢？我买这个课程反倒觉得，基础知识、常用库的介绍没那么重要，网上资料、官方文档随手可得。反而是了解FB这种大厂的Python工程是如何组织开发，这才最长见识。工程素养决定了你在生产工作中到底能走多远。...

展开 ∨

作者回复: 你说的很好



契卡

2019-05-13

👍 9

个人是一个生物信息学的研三学生，科研工作中大量涉及数据分析。我把Jupyter Notebook安装在课题组服务器上，用个人电脑远程连接使用。这样就可以在任何地方用一个浏览器就使用服务器的强大性能来处理课题组数据。ipynb也很方便和同学分享协作，他们可以很清晰准确地看到我处理数据的方法、过程和结果。向低年级新生传授知识时，Jupyter的交互性也使得演示变得快速简单直接。

展开 ∨



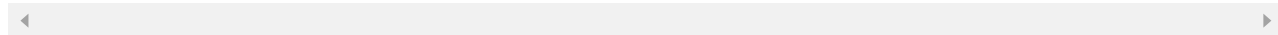
江湖海

2019-05-13

👍 8

工于善其事，必先利其器。把更多的精力和资源用到去解决实际最关键的问题，很好，谢谢老师的推荐。

作者回复: 嗯嗯



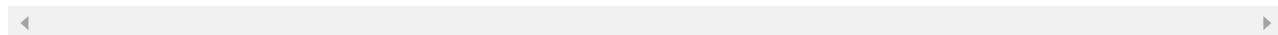
Fullmetal...

2019-05-13

👍 7

一直使用Pycharm的咋办？表示从来没有用过Jupyter
展开 ▾

作者回复: 如果你现在的问题都能解决也可以。



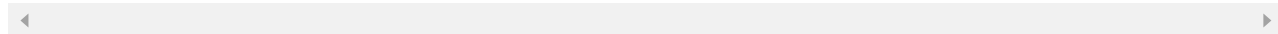
老贼

2019-05-13

👍 6

jupyter lab呢？
展开 ▾

作者回复: Jupyter lab和binder慢慢整合了，就是云端环境



AI悦创

2019-05-15

👍 5

详细的基础介绍才是这课程有价值的之一，之二就是进阶实践。而不是像某个评论一样，要是不用那和自学有什么区别。还有必要买课程吗？所以希望各位不用站在自身角度想问题，太片面也表示你格局。

展开 ▾



Geek_59f23...

2019-05-13

👍 4

pycharm2019.1已经支持jupyter notebooks了。老师好，我现在在一家培训机构学

python, 8月份学完出来就要找工作了, 那时候正好我们的课也结束了, 期待老师后面多分享点干货, 对初入职场的小白比较有用^_^

展开 ▾

作者回复: 好的

◀ ▶



cronusqiu

2019-05-13

👍 4

特地增加一篇来讲工具, 觉得不是很值得, 不同人使用工具习惯场景不同, 很像市面上绝大部分书开头都是介绍环境, 语法, hello world, 我觉得这个平台还是应该注重语言自身的核心, 已经哪些大厂如何很好地利用它来实现功能以及一些好的思想框架

作者回复: 你找到市面上哪本python书会介绍jupyter。孤立的看编程语言而看不到支撑开发的开发工具, 测试工具, 部署工具, 我不觉得能生产真正被用户使用的产品

◀ ▶



Geek_f4ef2...

2019-05-13

👍 4

安装Anaconda就有了这个工具, 可能用的少, 我并没有发觉到它带来的革命性。

作者回复: 可以尝试一下

◀ ▶



Anne颜

2019-05-22

👍 3

又不是所有人都像评论第一位那样, 对于我们没有基础的学员难道那位同学没有一点包容心?? 那样的评论作者没必要回复



Danpier

2019-05-13

👍 3

个人使用Anaconda+VSCode。Anaconda包含了conda和jupyter notebook, conda方便管理切换python环境, jupyter notebook非常适合用来学习数据分析。至于VSC既可以运行调试代码, 还可以直接新建一个终端在交互模式下跑代码, 支持powershell和

ipython。相比PyCharm，运存占用也比较低，对我那台老笔记非常友好。但是VSC刚上手需要学习配置一下工具环境。如果是小白，这点不介意的话，个人推荐...

展开 ▾



夜路破晓

2019-05-14

👍 2

自学Python之初就一直使用jupyter，就像老师专栏讲的，方便快捷、能够整合各种数据资源和所见即所得是其主要特点。

之前一直在用jupyter进行学习输出，但最近发现一个问题：我一般用jupyter写完推送github后，还要重复在博客上推送一次以便拓展接触面、认识更多小伙伴，毕竟还有很多人习惯博客得形式。每次从jupyter往博客搬运过程中就感到比较苦恼，因为你需要逐个ce...

展开 ▾



kevinsu

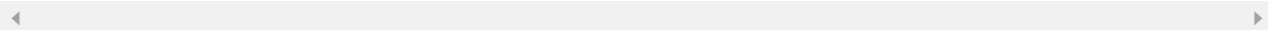
2019-05-14

👍 2

昨天在mac上安装解决了报错但是不太会用，今天研究了下很流畅，很爽！

展开 ▾

作者回复: 哈哈好的



郭杰

2019-05-13

👍 2

谢谢啦，我已经开始重视这个工具了

展开 ▾



坎中满

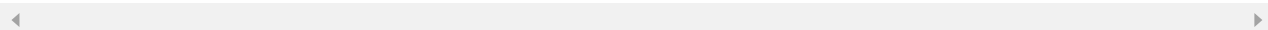
2019-05-13

👍 2

老师，我用的pharcm可以吗

展开 ▾

作者回复: 可以，另一个留言也回复到了。工具要看应用场景。





Tango

2019-05-13

👍 2

第三天打卡。除了丘比特笔记本，还喜欢用vscode。

展开 ▾

作者回复: 哈哈



志

2019-05-14

👍 1

1、Jupyter notebook 平时用来进行日常工作中的数据分析非常高效，「所见即所得」，结合内嵌的 Markdown 可以完整记录分析步骤和思路，同时代码运行结果可以导出 HTML或PDF分享给其他同事，甚至还能进行幻灯片放映～

2、看到有些同学说到代码自动补全的问题，其实Jupyter notebook 有很多插件，可以看看这篇介绍文章https://mp.weixin.qq.com/s/fpWtJAOb_Uz5ApndtXAlyw，里面第...

展开 ▾



马星

2019-05-14

👍 1

我本机装了anaconda，可以通过<http://localhost:8890/tree>访问jupyter,为什么别的机器通过<http://我的ip:8890/tree>访问不了呢，请指教！别的机器是能ping通我的ip的，我的远程访问也打开了，防火墙也关了

展开 ▾