

在教学和学习中使用 Jupyter

Teaching and Learning with Jupyter



在教学和学习中使用 Jupyter

Teaching and Learning with Jupyter

by

Student Name Student Number

First Surname 1234567

Instructor: I. Surname

Teaching Assistant: I. Surname

Project Duration: Month, Year - Month, Year

Faculty: Faculty of Aerospace Engineering, Delft

Cover: Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA
under CC BY-NC 2.0 (Modified)

Style: TU Delft Report Style, with modifications by Daan Zwaneveld



目录

1 前言	1
2 我们为什么使用 Jupyter 笔记本	2
2.1 我们为什么要使用 Jupyter?	2
2.2 但首先，什么是 Jupyter 笔记本?	2
2.3 课程的益处 & 趣闻轶事	4
2.3.1 公约	4
2.3.2 参与	5
2.3.3 增进了解	6
2.3.4 提高学生的成绩	7
2.3.5 增加学生的职业准备	7
2.4 学生福利	8
2.4.1 计算性思维	8
2.4.2 开放源代码	8
2.4.3 主动学习	9
2.5 教员福利	9
2.6 结论	9
3 Notebooks in teaching and learning	10
3.1 3.1 Oh the places your notebooks will go!	10
3.1.1 3.1.1 Introduction	10
3.1.2 3.1.2 Jupyter notebooks as textbooks	10
3.1.3 3.1.3 Notebooks as workbooks/primer	11
3.1.4 3.1.4 Notebooks as worksheets/drill sets	11
3.1.5 3.1.5 Notebooks as notepaper or course packets	12
3.1.6 3.1.6 Notebooks as an app	12
3.1.7 3.1.7 Notebooks as lab reports or assignments	12
3.1.8 3.1.8 Notebooks as interactive multimedia platforms	12
3.1.9 3.1.9 Notebooks as a demonstration platform	13
3.1.10 3.1.10 Notebooks as a live coding environment	14
3.1.11 3.1.11 Conclusion about places	15
3.2 3.2 Before You Begin...	15
3.2.1 3.2.1 Identify Your Teaching Goals	15
3.2.2 3.2.2 Understand Your Students	15
3.2.3 3.2.3 Cultivate Student Study Skills and Learning Strategies	16

3.2.4	3.2.4 Develop Your Content Strategy	16
3.2.5	3.2.5 Consider the Context of the Learning Environment	17
3.2.6	3.2.6 Crafting an experience: choose the right tool and approach for the task	17
3.2.7	3.2.7 Transitioning to and from Jupyter	18
3.2.8	3.2.8 Conclusion about teaching practices	18
4	Chapter 4 A catalogue of pedagogical patterns	19
4.1	4.1 Introduction	19
4.2	4.2 Shift-Enter for the win	19
4.3	4.4.3 Fill in the blanks	20
4.3.1	4.4 Target Practice	20
4.3.2	4.5 Tweak, twiddle, and frob	21
4.3.3	4.6 Notebook as an app	21
4.3.4	4.7 Win-day-one	22
4.3.5	4.8 Top-down sequence	22
4.3.6	4.9 Two bites at every apple	23
4.3.7	4.10 Coding as translation	24
4.3.8	4.11 Symbolic math over pencil + paper	24
4.3.9	4.12 Replace analysis with numerical methods	25
4.3.10	4.13 The API is the lesson	25
4.3.11	4.14 Proof by example, disproof by counterexample	26
4.3.12	4.15 The world is your dataset	27
4.3.13	4.16 Now you try (with different data or process)	28
4.3.14	4.17 Connect to external audiences	29
4.3.15	4.18 There can be only one	30
4.3.16	4.19 Hello, world!	31
4.3.17	4.20 Test driven development	33
4.3.18	4.21 Code reviews	34
4.3.19	4.22 Bug hunt	34
4.3.20	4.23 Adversarial programming	34
5	Jupyter Notebook ecosystem	36
5.1	5.1 Language support: kernels	36
5.2	5.2 Using Jupyter notebooks	36
5.3	5.3 Authoring Jupyter notebooks	37
5.3.1	5.3.1 Accessing documentation in the notebook	37
5.3.2	5.3.2 Widgets	37
5.3.3	5.3.3 Magics	39
5.3.4	5.3.4 Notebooks under version control	40
5.3.5	5.3.5 Testing notebooks	40
5.3.6	5.3.6 Essential Python libraries	40
5.3.7	5.3.7 Advanced topic: extensions	41

5.4	5.4 Tips and tricks	42
5.4.1	5.4.1 Reminders	42
5.4.2	5.4.2 Feedback	42
5.4.3	5.4.3 Explaining each cell	42
5.4.4	5.4.4 How to structure code cells	43
5.4.5	5.4.5 Custom styling	43
5.4.6	5.4.6 Length of notebooks	43
5.5	5.5 Gotchas	43
5.5.1	5.5.1 Programming language ≠Jupyter	43
5.5.2	5.5.2 Restart, restart, restart	44
5.5.3	5.5.3 Notebook hygiene	44
6	Chapter 6 Getting your class going with Jupyter	46
6.1	6.1 Local installation on students'or lab computers	46
6.1.1	6.1.1 Jupyter on student-owned computers	47
6.1.2	6.1.2 Jupyter on lab computers	47
6.2	6.2 Jupyter on remote servers	48
6.2.1	6.2.1 Running in a temporary environment in the cloud	48
6.2.2	6.2.2 Running on servers you control	49
6.2.3	6.2.3 Running Jupyter in the cloud	49
6.3	6.3 Distribution and collection of materials	50
6.3.1	6.3.1 Learning management systems	51
6.3.2	6.3.2 Web hosting	51
6.3.3	6.3.3 GitHub	51
6.3.4	6.3.4 JupyterHub	51
6.3.5	6.3.5 Using an LMS and nbgrader together:	52
6.4	6.4 Assessing student learning with Jupyter notebooks	52
6.5	6.5 How do you create Jupyter notebooks for reuse and sharing?	53
6.6	6.6 Jupyter: a 21st Century genre of Open Educational Resources and practices .	53
7	研究使用的案例	55
7.1	7.1 Jupyter 笔记本支持大面积招生的扩展性	55
7.1.1	7.1.1 支持加州大学伯克利分校的大量招生课程	55
7.1.2	7.1.2 在加拿大各地开展的活动大规模采用 Jupyter	55
7.1.3	7.1.3 快速转换：将现有课程转移到 Python 和 Jupyter (在最后一分钟)。	56
7.2	7.2 CFD Python ”的故事：以自己的节奏引导学习者	56
7.3	7.3 用 music21 分析音乐	57
7.4	7.4 计算机科学中的互动性（高中和初中）	58
7.5	7.5 使用 Jupyter 的交互式地球物理学	59
7.6	7.6 Investigating hurricanes	60
7.7	7.7 Riemann Problems and Jupyter Solutions	61

8 关于作者	63
8.1 项目负责人	63
8.2 定稿阶段的作者	63
9 Conclusion	66

1

前言

本手册适用于任何教授包括数据分析或计算以支持学习的主题的教育工作者。它不仅适用于教授工程或科学课程的教育工作者，也适用于数据新闻、商业和定量经济学、基于数据的决策科学和政策、定量健康科学和数字人文学科。它旨在提供一个切入点，以及对 **Jupyter** 在教育领域的广泛概述。无论你是已经在使用 **Jupyter** 进行教学，还是发现建立在 **Jupyter** 上的学习材料激起了你的好奇心，或者从未听说过 **Jupyter**，这本公开书中的材料都可以让你在教学中使用这一技术。

Jupyter 项目是一个广泛的合作项目，为互动和探索性计算开发开源工具。这些工具包括：100 多种计算机语言（重点是 **Python**），**Jupyter** 笔记本，**JupyterHub**，以及一个由大型社区贡献的扩展生态系统。自 2014 年底以来，**Jupyter** 笔记本的受欢迎程度呈爆炸式增长，这是因为它被作为从事数据科学的最受欢迎的环境而得到了推动。它也成长为一个在课堂上使用的平台，用于开发教学材料，分享课程和教程，以及创建计算故事。笔记本是包含有图像和数学的文本叙述的文件，与可执行代码（支持许多语言）和该代码的输出相结合。这种内容和代码的结合使得基于数据的交流成为一种强大的新形式。各地的教育工作者正在采用 **Jupyter** 进行教学。

新近采用 **Jupyter** 的教育工作者可能会因为要浏览工具和内容的生态系统而不知所措。他们可以研究许多例子，或者阅读大量的博客文章和讲座视频，以提炼出良好的实践模式和技术解决方案，为他们的学生提供最佳服务。几个早期采用者有很多经验可以分享，决定开始收集这些知识，并分享关于使用 **Jupyter** 进行教学的公开文件。其结果就是这本开放的书：一个记录社区成员在教育中使用 **Jupyter** 的经验的活文件。

在华盛顿特区举行的 **Jupyter** 社区研讨会（2018 年 11 月）开始了这一过程，以制作这本手册的第一版为目标进行了图书冲刺。这本合作编写的书整合了涵盖关键主题的解释和例子，包括：什么是 **Jupyter**，如何尝试 **Jupyter**，与学生分享笔记本，在本地安装 **Jupyter**，云服务，寻找示例笔记本，用 **Jupyter** 编写课程，为课程制作集合，用 **nbconvert** 导出到其他格式，用 **Jupyter** 编写教科书。使用 **Binder** 和 **JupyterHub**，制作作业和自动评分，制作在线课程，在课堂上用 **Jupyter** 教学，用 **Jupyter** 进行主动学习和翻转学习教学法，以及引导学习者在 **Jupyter** 中创建自己的内容。这本开放的手册将逐渐包含你在教学中需要了解的所有 **Jupyter**。

如果你觉得这些材料有帮助或有启发，请在 Twitter 上用 **#Jupyter4Edu** 给我们打招呼。我们希望你能这样做！

2

我们为什么使用 Jupyter 笔记本

2.1. 我们为什么要使用 Jupyter?

作为教师，我们要负责许多活动，包括创造课程、讲座、课程、作业和支持性环境；鼓励学生在课堂上的参与和表现；帮助学生学会批判性思考，使他们成为终身学习者和问题解决者；使材料与学生不同的兴趣和背景相关并有意义；评估学生的学习（包括评分和评价）；鼓励学生坚持情感劳动（反馈、沟通等）；尝试教学和学习实践，提高我们做所有这些事情的能力。

简而言之，我们设计学习环境和经验。

我们使用 **Jupyter** 笔记本来设计学习环境，帮助支持这些活动。我们相信，在教学中加入 **Jupyter** 笔记本使我们能够提高学生对课程内容的理解，提高学生对材料的参与度和对课堂的参与度，并使概念更有意义，与学生的不同兴趣相关。我们代表不同的学科，有许多不同的教学目标，所有这些都得到了使用 **Jupyter** 笔记本的支持。本手册的目的是为您提供一些想法，帮助您实现自己的教学和教学目标。

通过一系列的轶事，我们将说明作为一名教育工作者，你如何使用 **Jupyter** 笔记本来提高你的学生：1) 参与，2) 参与，3) 理解，4) 表现，以及 5) 为他们的职业生涯做准备。这些都是起点，我们相信你也会把这些例子带到新的和令人兴奋的方向。

2.2. 但首先，什么是 Jupyter 笔记本？

Jupyter 项目是三件事：一个标准的集合，一个社区，以及一套软件工具。**Jupyter Notebook** 是 **Jupyter** 的一部分，是创建 **Jupyter** 笔记本的软件。**Jupyter** 笔记本是一个支持混合可执行代码、方程式、可视化和叙述性文本的文件。具体来说，**Jupyter** 笔记本允许用户将数据、代码和散文结合起来，讲述一个互动的、计算性的故事。无论是分析美国文学语料库、创作音乐和艺术，还是说明数字信号处理背后的工程概念，笔记本都可以将传统的教科书中的解释与应用程序的互动性相结合。

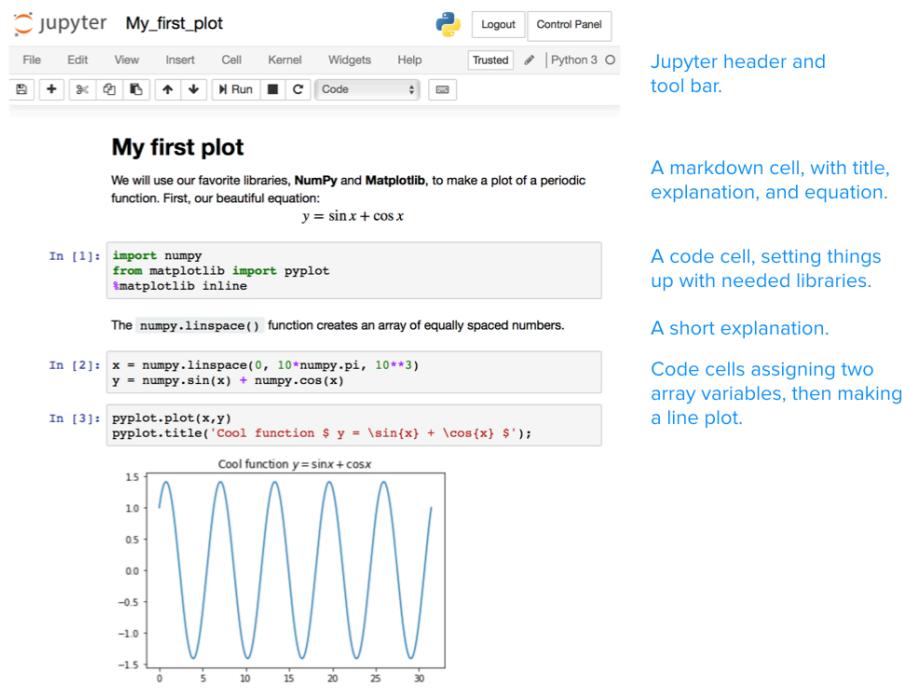


图 2.1: 一个 Jupyter 笔记本，从一个包含标题和解释的 markdown 单元开始（包括一个用 LaTeX 渲染的方程式）。三个代码单元产生最后的内联图。

一个 Jupyter 笔记本，从一个包含标题和解释的 markdown 单元开始（包括一个用 LaTeX 渲染的方程）。三个代码单元产生最后的内联图。

Jupyter 是一个免费的、开源的平台，对学生来说是一个很好的学习环境。对于教师来说，它提高了我们的效率，减少了认知负荷，因此我们可以吸引学生。笔记本可以在众多环境中实现你作为教师的目标，从 STEM 实验室或人文叙事，到讲台上的演讲或翻转课堂。我们在小班和有数百名学生的班级中使用 Jupyter 笔记本。Jupyter 笔记本可以用来教授一个讲座的一部分，也可以用来教授整个课程。Jupyter 笔记本使我们和我们的学生能够就一个问题进行对话，并链接到资源，如音频、视频、图像、可视化—甚至允许学生混合和重新混合这些。然而，学生们除了需要安装一个现代的网络浏览器外，不需要其他任何东西来使用这个免费软件。

Jupyter 笔记本可以用来组织课堂材料和对象，存储并为学生提供阅读材料，展示和分享讲座材料，进行现场编码，探索和与材料互动，支持自定进度的学习，为学生的作业打分，解决作业问题，或使材料可重复使用给他人（见第 3 章和第 4 章）。

继续阅读，了解我们是如何将 Jupyter 笔记本用于教学，使我们的学生和我们自己都受益。Jupyter 笔记本支持广泛的学习目标，包括学习编程、学习领域知识，以及练习交流技能，如讲故事。本书的作者曾使用 Jupyter 笔记本进行教学。

科学物理学和天文学地球科学生物学认知科学计算机科学数据科学统计学社会科学写作写作研讨会写作和技术交流数字人文学科音乐文本分析元数据处理工程学化学工程（动力学和反应器设计）机械工程航空航天工程编程入门高中学院和大学水平的课程（真正的入门到高级课程）我们在教育方面对笔记本的其他使用包括。

建立模型/模拟（有和没有编程）。使用小工具来演示并与模拟进行互动过程和数据的可视化

2.3. 课程的益处 & 趣闻轶事

2.3.1. 公约

作为教师，我们经常为吸引学生而努力，特别是当我们受到课程形式（例如，在线，50分钟的讲座）、现有技术、学生的分心和/或其他因素的限制时。尽管如此，我们还是有责任在这些限制下创造环境和经验，以吸引学生参与我们的课程。这就是笔记本可以为你提供另一个工具，以打破平凡，让学生参与到他们的学习中。

与数据的对话

Jupyter 的创造者将其描述为一套用于互动和探索性计算的开源工具，以及一个用于创建计算叙述的平台。**Jupyter** 允许我们，作为教育者，叙述“学生和数据之间的对话”。考虑一下这个例子，使用多年来许多国家的预期寿命数据。

我用一小段代码做了一个显示时间演变的图形，就是所谓的“意大利面条图”（见图）。看着这个混乱的图形，我指出大多数线条显示出随着时间的推移而增长：全世界的预期寿命都在提高。但是有几条线显示在某一年出现了明显的下滑。我可以问学生：哪个国家出现了这种下降？那里发生了什么？为什么？通过更多的编码，我们发现柬埔寨在 1977 年的预期寿命约为 30 岁，令人震惊，而卢旺达在 1992 年的预期寿命甚至更糟。然后我们就有机会讨论为什么这些国家经历了死亡危机。这些数据带来了有意义的讨论，有许多可能的路径，涉及历史、政治、经济和健康。- 罗蕾娜-巴尔巴

Life expectancy in the years 1952–2007, across 142 countries

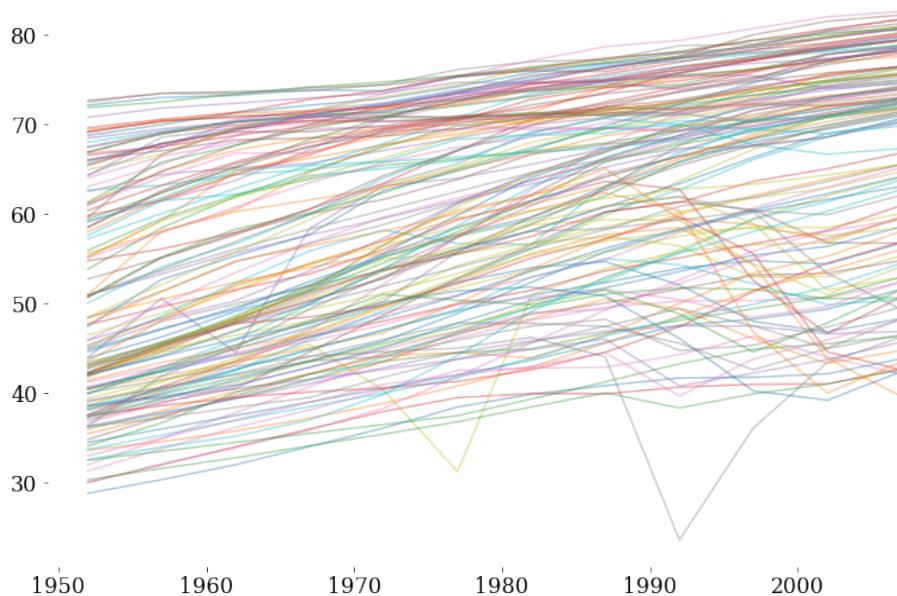


图 2.2: 来自: <http://go.gwu.edu/engcomp2lesson4>

Jupyter 笔记本是连接的基本工具—让学习者参与到他们思维的转换中。将计算融合到叙述中的机会，创造与数据的对话是一种强大而有效的交流形式。有了 **Jupyter**，你现在有了一种新的内容形式来创建并与学习者分享：可计算内容。在一个每个主题都可以有数据支持的处理方式的世界里，计算设备无所不在，无孔不入，自然语言和计算的结合创造了引人注目的交流和学习机会。

2.3.2. 参与

让学生参与你的课程需要他们的参与和与你、他们的同伴和/或内容的互动（Moore, 1989）。当然，如何、何时以及为什么使用学生参与你的课程，将取决于你的目标、课程中教学内容的具体目标、你的学生以及其他因素。然而，使用笔记本可以鼓励学生参与，并为你提供更多的工具来促进参与。笔记本也可以将学生与真实的外部受众联系起来。例如，学生可以消费其他班级的笔记本，并在其他人可以阅读的地方发布笔记本。

真实世界的经验--将概念变为现实

笔记本是活的文件，这意味着它们可以被编辑，以回应学生的问题或输入，并在讲座或演讲中被用作对话的内容。

我们小组使用 **Jupyter** 笔记本作为“应用程序”来展示地球物理学的概念。这些笔记本-应用程序将数值模拟与小工具和相关的图画联系起来。在课堂上，我们要求学生根据他们感兴趣的应用或案例研究来帮助定义输入参数。在显示结果之前，我们要求学生建立一个他们期望的心理图像。如果结果图像符合他们的期望，那么我们就强化了一个概念，如果不符，这就是一个学习的机会。作为教员，我们可以通过更新模拟的输入来与学生的问题进行互动，以便与他们一起探讨概念。学生可以通过像 **Binder** 这样的免费网络平台访问同样的笔记本，所以只需通过一个链接，他们就可以掌握方向盘，自己参与到概念中去。笔记本将概念带入生活，并作为学习者和教育者之间互动的一个对话内容。- 林赛-海吉

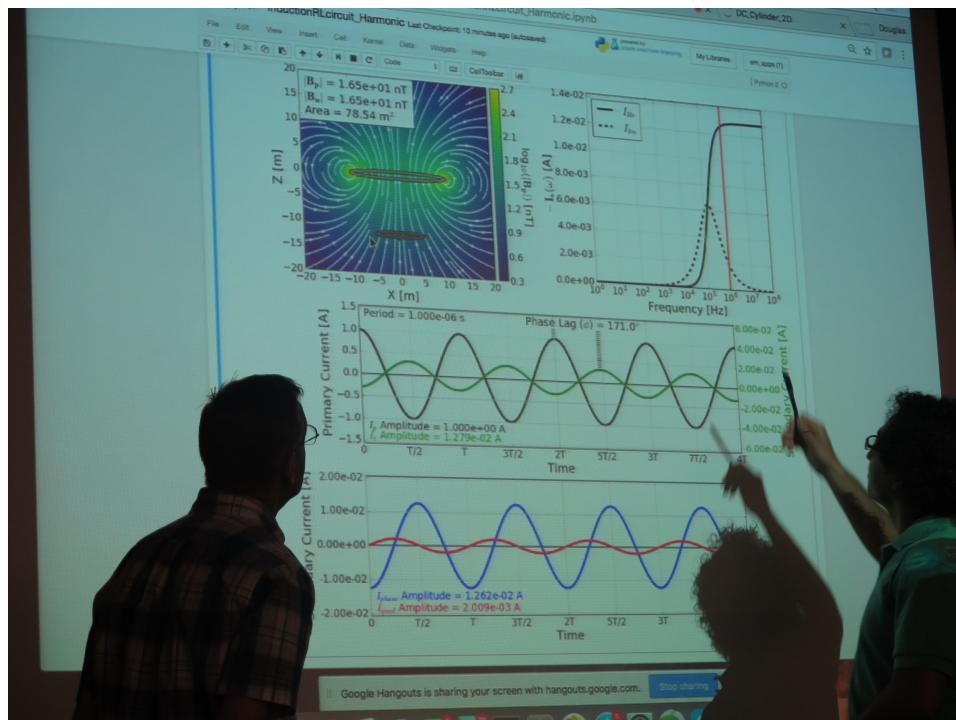


图 2.3: 道格拉斯-奥尔登堡博士（左）在地球物理电磁学的短期课程中与学生进行交流 (<https://geosci.xyz>)。照片来源: Seogi Kang

真实世界的经验--离开的门票

用 **Jupyter** 笔记本在课堂上产生参与的另一个例子是活动魔法，可作为一个扩展。它通过笔记本创建了所谓的“离开的票”（或“退出的票”）。离开票”的想法是结束课堂或实验室的一个很好的方式。简而言之，它只是你给学生的一个调查表（见图）。通常，这些调查是通过个人回应系统（也

被称为“点击器”或 PRS) 或手机进行的。这种调查有几个用途。

给教师一些关于学生整体理解的反馈意见为学生提供时间和机会来回顾和综合今天的材料让学生把最近的知识应用到一个新的问题上。为学习材料提供一个额外的实例

In [1]: %activity "/home/dblank/activities/datastructures01"

Question ID: cs206/p1/1

How are Maps and Sets similar?

1) both have keys and values
2) both have an explicit ordering
3) both are sets of ordered pairs
4) can be implemented with hashes
5) both organize abstract information

1 2 3 4 5

Respond:

Previous Next

图 2.4: 从学生的角度看活动魔法的例子。显示了一个问题，有多个选择的答案，有按钮供他们输入。

这些问题通常不需要太多的时间来回答，而是为了捕捉班级对话的本质。经过一分钟左右的思考，学生选择他们的答案（通过点击其中一个按钮），教师显示格式塔的结果（见图）。

In [4]: %activity "/home/dblank/activities/datastructures01"

Question ID: cs206/p1/1

How are Maps and Sets similar?

1) both have keys and values
2) both have an explicit ordering
3) both are sets of ordered pairs
4) can be implemented with hashes
5) both organize abstract information

1 2 3 4 5

Respond:

Previous Results Next

A bar chart titled "Results" shows the distribution of student responses. The x-axis represents the five options (1, 2, 3, 4, 5), and the y-axis represents the count from 0.0 to 7.0. The bars for options 4 and 5 reach the maximum value of 7.0.

Response	Count
1	~0.5
2	0
3	0
4	7.0
5	7.0

图 2.5: 活动魔法，从教员的角度来看。一旦所有的学生都有机会回答，投影仪上就会显示出图表。

好的“退出票”问题可以是特定领域的问题，也可以是元认知问题（例如关于一个人的学习风格），或者是高层次的组织问题（例如，“今天讨论的目的是什么？”）。我们建议在课堂结束时留出足够的时间（也许是 10 分钟），以便进行充分和完整的总结讨论。讨论结束后，如果你觉得没有足够的学生获得你所期望的洞察力，你可能希望调整接下来的班会。关于“离开的门票”的更多信息，见 <https://www.brown.edu/sheridan/teaching-learning-resources/teaching-resources/course-design/classroom-assessment/entrance-and-exit/sample>。关于 Jupyter 笔记本扩展的更多信息，见使用和安装 Calysto 活动魔法。

2.3.3. 增进了解

在任何课程中，你通常会试图实现一系列不同的目标。本杰明-布鲁姆(https://en.wikipedia.org/wiki/Bloom%27s_taxonomy)为我们想要实现的详细目标提供了一个框架，从基本知识（如，术语、具体事实、趋势和顺序、分类

和类别等)一直到评价和创造能力(如,抽象关系、判断但基于标准、原创作品)。实现前者(即基本知识和理解)比理解(即评价和创造)要容易得多;然而,作为教育工作者,我们大多数时候都在努力提高学生对我们所教主题的复杂理解。好消息是,笔记本提供了一个有价值的教学工具,以实现理解—例如,让学生从被动地观看课程内容转向主动地探索、分析、综合和评价内容。

真实世界的经验--以自己的节奏引导学习者

航空航天工程中使用的计算流体动力学(CFD)的基本理论是基于对纳维-斯托克斯方程的理解。”**CFD Python**”是一个基于实践模块的 Jupyter 笔记本集, **Lorena Barba** 于 2009 年在波士顿大学的计算流体动力学(CFD)课程中开始使用该模块。这个为期 5 周的模块开发了一些工作实例,这些实例相互关联,逐步引导学习者创建一个程序,在 12 个步骤中解决流体动力学的纳维-斯托克斯方程。

2013 年,我被邀请在阿根廷的拉丁美洲高性能计算学校教授为期两天的小型课程。Jupyter 笔记本平台使我能够创建一个引导性的叙述,以支持具有不同背景经验和知识的学习者。在这次活动中,我们根据 CFD 课程模块编写了笔记本,作为迷你课程的教学支架。20 名学生以自学的方式完成了笔记本,而我则从一个桌子到另一个桌子询问和回答问题。大约四名学生在两天内完成了所有的 12 个步骤,其中大部分学生完成了第 8 个步骤,少数学生在课程结束时在第 4 或第 5 个步骤上落后了。对于那些完成整个模块的人来说,他们在 2 天内完成了我在课堂上的普通学生通常需要 5 周才能完成的任务。看到这一幕,我大开眼界:代码中的工作实例的力量,以及允许学习者按照自己的节奏学习的能力,使这些学习者有了显著的变化。- 罗蕾娜-巴尔巴

基于开发”**CFD Python**”学习模块的经验 (Barba & Forsyth, 2018), 我们采用了这种基本的设计模式来创建使用可计算内容的课程。

把它分解成小的步骤把小步骤分成大步骤增加叙述和联系链接到文档穿插简单的练习用挑战问题/任务来调剂在网上公开发表这对学生的理解特别有帮助。

2.3.4. 提高学生的成绩

学习的目标往往是通过学生的表现来实现的。这通常是通过我们在教学过程中和教学结束时试图评估的内容最明显的。使用笔记本,我们可以为学生创造各种表演机会,从而给他们更多的练习和反馈机会,同时也给我们这些教员更多的机会来评估他们的表演能力。

真实世界的经验--工作实例效应

工作实例效应是认知负荷效应中最著名和最广泛的研究 (Sweller, 2006)。它指的是为如何解决一个问题提供充分的指导,从而使学生的表现比没有指导的情况下解决问题的表现更好。对于复杂的任务,没有经验的或初级的学习者从工作实例程序中受益最大。一项研究 (Chen, Kalyuga, & Sweller, 2015) 得出结论:“工作实例效应发生在复杂的、高元素的互动性材料上,这些材料给工作记忆带来了沉重的负担”,“当处理学习者可能难以理解的复杂材料时,高水平的指导可能会比低水平的指导带来更多的表现”。这种基于研究的指导似乎特别适用于教新手程序员在他们的主题(科学、工程或其他)范围内使用计算。

2.3.5. 增加学生的职业准备

在培养学生应用所学知识的过程中,努力使课程中发生的事情与他们在职业生涯中所经历的事情保持一致是很重要的。从使用平行软件到反映工作流程,我们希望我们的学生能够体验并为

工作场所做好准备。当然，我们认识到，工作场所不是一成不变的，职业所需的技能也总是不断涌现的，使用笔记本提供了一个灵活的平台来培养技能，并建立学生能做的事的档案。

真实世界的经验--发表数据叙述作为行业能力的证明

对于数据科学职业来说，公开分享关于数据分析项目的叙述，在面试时可以很好地展示学生的潜力。伊丽莎白-威克斯让她的学生开发一个 **Jupyter** 笔记本，讲述在课堂上完成的数据挖掘和分析项目的故事。然后，学生们将这个笔记本发布到他们的 **Github** 个人资料页面。由于 **Jupyter** 是工业界最流行的交流数据科学成果的方式之一，学生们拥有一个非常有价值的潜在职业关键。

TODO: 添加伊丽莎白的名言

2.4. 学生福利

为学生创造机会，使其发展成为学习者，这超出了你可能使用笔记本的任何具体课程的界限。通过丰富他们在课程中的学习经验，你将帮助他们发展有价值的技能和思维方式，并将其带入其他课程和职业生涯中。

2.4.1. 计算性思维

Jupyter 笔记本支持广泛的学习目标。它的交互性能够建立对领域知识的直观理解，例如在改变参数时对系统的机械响应的理解，或者理解一个算法的行为方式。笔记本还可以帮助教授有效的沟通技巧，将散文与图形结合起来，形成一个强有力的叙述。最后，通过将代码与文字描述和可视化相结合，笔记本可以支持教学或加强编程技能。即使笔记本被设计成被动消费，接触代码也有助于向学生展示如何做某事—他们可以自己做。这也有助于为那些不认为自己是传统“计算机科学”类型的学生揭开编码的神秘面纱。

使用笔记本，你可以创造丰富的学习经验，将计算思维的核心基础联系起来。

分解。将数据、过程或问题分解成较小的、可管理的部分模式识别。观察数据中的模式、趋势和规律性抽象。识别产生这些模式的一般原则算法设计。为解决这个问题和类似的问题制定循序渐进的指令

2.4.2. 开放源代码

将笔记本整合到课堂上，也让学生接触到一个庞大且不断增长的开源工具生态系统。这支持了他们的教育，同时也提供了与数据科学和机器学习等对训练有素的员工需求量大的行业所使用的工具相同的环境中的经验。这些工具的开源性质也确保了所有学生—包括那些在传统大学环境之外的学生—都能获得并负担得起课程内容。

与 **Mathematica** 等专有笔记本技术，或 **Matlab** 或 **C++** 等特定编程语言/环境不同，学生使用 **Jupyter** 笔记本学习的门槛可以非常低。至少，在讲座期间，学生可以简单地观看/阅读使用笔记本的互动演示，以取代幻灯片或讲义。在他们自己身上，使用云服务，如 **Binder** 或 **Jupyter-Hub**，学生可以打开任何现代网络浏览器到一些地址并与笔记本互动（这种技术的例子可以在 <https://jupyter.org/try>），不需要任何安装或配置。在最复杂的情况下，学生可以安装 **Anaconda**，并按照简单的指示安装 **Jupyter** 笔记本，它在所有平台上的工作和外观都是一样的，而且是免费和开源的。

2.4.3. 主动学习

由于其互动性，笔记本能够实现一系列的主动学习方法，这已被证明能够提高科学、工程和数学的成绩（Freeman 等人，2014）。开始时，学生可以通过阅读和运行笔记本来消费笔记本的内容，然后转为编辑或完成笔记本的作业。这使学生能够专注于内容和概念，而不仅仅是记笔记。

在布卢姆分类学的顶端是纯粹的创造，例如，学生可以编写完整的计算性文章。在这两种情况下，笔记本都支持学生具有广泛经验和能力的课程：需要帮助的学生可以依靠散文解释和现有代码的支架，同时也为更有经验的学生提供伸展和探索的空间。伴随着代码的额外注释和散文也有助于支持非传统的学习者和来自代表性不足的群体的学生，他们可能对编程的初始经验/舒适度较低。

通过使用笔记本，灌输主动学习的习惯，也会带来超越课程范围的好处。互动性推动了参与、兴趣和对概念的探索。在你的课程中，参与的学生更有可能在其他课程和其他方面成为参与的学习者。

2.5. 教员福利

笔记本可以在不同的层次和形式上被采用，根据课程的需要和教师的舒适度/兴趣水平提供灵活性：课堂演示、互动实验室、辅助材料（例如，书本替代物、讲义补充物）、作业，或翻转学习环境中的全部课程内容。笔记本为没有经验的教员提供了一条通往主动学习方法的途径，但并不强求一种特定的教学风格。

至少，笔记本可以用来制作可发布的互动式讲义，将叙述性文字、图片、视频与图像和结果融合在一起，以呈现概念。此外，这些课程材料可以逐步开发，从一个低强度的草稿开始，到一个更完善的、可发布的文件，随着时间的推移，可以很容易地扩展，并被其他人采用。围绕软件工具和教育资源的开源社区的发展，为现有资源的再利用和改编创造了更多的机会。

虽然许多笔记本作者确实使用 Python，但 Jupyter 笔记本支持许多语言，所以学生（和导师）不会被一种特定的语言所束缚。事实上，Jupyter 这个名字来自于三种语言。此外，这些（免费）工具的进入门槛极低—使用云基础设施意味着学生和导师不必安装任何东西，而在“最坏”的情况下，安装需要一些命令行的考察，但这些都是免费的、公开的和跨平台的。

2.6. 结论

我们希望本章除能说明，用 Jupyter 笔记本教学对你和你的学生都有价值。我们已经表明，笔记本是一种工具，可以提高学生的参与度、参与度、理解力、表现力，以及为他们的职业生涯做准备。这些都是可以在各种学科和内容领域实现的实质性成就。通过几个现实世界的例子，我们试图说明教师使用笔记本的众多方式。希望这些，结合后面的章节，能够指导你：**1)** 支持学生的学习；**2)** 让你有信心使用笔记本；**3)** 帮助你了解必要的后勤保障；**4)** 帮助你对使用 Jupyter 笔记本所能实现的目标有清晰的预期。

3

Notebooks in teaching and learning

Jupyter 笔记本对教师来说是一个有价值的工具，但只有当你在课程的背景下正确应用它们时，它们的价值才能得到发挥。在本章中，你将了解教师如何初步构建他们的课程设计，然后确定何时以及如何使用笔记本来实现其目标。

3.1. 3.1 Oh the places your notebooks will go!

3.1.1. 3.1.1 Introduction

在第四章中，您将了解到在课程设计中使用笔记本的许多创造性方法。很多时候，笔记本可以被改编到你已经在做的课程活动中，而其他时候，笔记本会给你机会来扩展你过去所做的事情，以提高学生的参与度、参与度、理解度和表现度。**Jupyter** 笔记本可以成为你现有教学工具箱中的一个重要成员，在学习环境中的每一个环节都很有用。**Jupyter** 的新使用者可以从小处着手，将笔记本纳入单一模块、作业或课堂活动中。这是一个很好的方法，可以看到你的学习对象如何与笔记本环境互动，并以低成本/风险的方式探索笔记本托管系统。

教员们在他们的课堂上采用 **Jupyter** 的水平各不相同，每个人都利用了环境的强大功能。将现有的课程过渡到任何新的平台似乎都很艰巨，但 **Jupyter** 笔记本是模块化的，是采用迭代开发方法的理想选择。有些人可能会发现自己继承了已经在 **Jupyter** 中构建的课程，而其他人则会选择完全在 **Jupyter** 中构建新课程。

本节旨在启发你关于 **Jupyter** 在课堂内容和演示设计中的许多用途，并预览 **Jupyter** 生态系统的其他元素如何能被整合到该用途中。其中一些用途可以作为实验快速启动，以试水。

3.1.2. 3.1.2 Jupyter notebooks as textbooks

教员们经常把 **Jupyter** 笔记本写成线性叙述性文件。这些笔记本将被学生和学习者阅读，也许是通过工作，被标记，是一种相对单面的信息消费体验。

大多数情况下，这些笔记本是作为学习者在课前需要做的阅读，作为参考材料（例如，为未来的评估进行复习），或者作为自我学习的一部分，学习者自己完成的东西。**Jupyter** 笔记本可以由完全静态的文本构成，它可以作为将现有材料过渡到笔记本环境的起点。

这种传统的静态教科书章节可以通过将内联代码文本改为支持修改和实验的可执行代码单元，扩展到一个活跃的空间。添加带有询问代码的建议的提示和例子，在不改写原始内容的情况下，进一步扩展主动学习的机会。交互式滑块、用户输入源和可操作的可视化是其他小工具和插件如何开辟更多可能性的例子。

正如你在后面的章节中所看到的，许多作者正在使用 **Jupyter** 笔记本作为他们的主要创作和出版平台。这些材料在纸上和网上发表，意味着学习经验的互动部分是资源中的一流元素。

3.1.3. 3.1.3 Notebooks as workbooks/primer

工作簿通过包括要求学生操作或创造新内容的活跃元素，使学生参与到笔记本环境中。这使学生从一个像书本一样的被动或静态的学习环境进入一个主动的学习体验，他们可以参与并批判性地思考这些内容。

你可以在一本工作簿中包括许多教学模式（在下一章中讨论），精心制作一个完全定制的学习体验。这些工作簿可以作为学生的独立学习（例如，翻转课堂的前期工作），或者作为个人或小组的课堂活动的一部分。

在技术领域的教学中，在概念的背景下进行探索是向学生展示真实经验的一个理想环境。孤立地研究一个概念可能会降低问题的复杂性，使之更容易理解，但这就把它从它存在的背景中剔除了，可能会使学生更难把所有的碎片放在一起，并把它综合到他们更大的问题解决方式中。

对这种背景的保留增加了说明性文字的复杂性，在不相干的文件中的技术参考，或一些模板代码。在一个大的脚本或冗长的方向清单中提供这些内容，可能会让学生感到难以承受。**Jupyter** 笔记本是教授这些复杂工作流程的优秀工具，因为它不是一个带有冗长代码注释的脚本，也不是有断开的文档，而是可以将这种指导性文本准确地嵌入到编码发生的地方。这种指导材料也可以被格式化为 **markdown** 单元，在视觉上与代码不同，使其更容易被视觉扫描，并为添加有用的格式化提供了可能。

各种各样的活动都可以用可执行的代码单元来支持，因此学习者可以在一个互动和迭代的环境中探索这个空间。他们可以看到和检查周围的部分代码，但不需要触摸它，为作业和挑战保持适当的颗粒度。使用 **markdown** 格式来分隔各部分，有助于为更大的问题提供支架，帮助他们真正体验真实世界的工作流程，而不是静态地阅读它。

3.1.4. 3.1.4 Notebooks as worksheets/drill sets

许多编程和领域任务都有特定的语法需要学习和有效记忆，然后才能被内化。类似于数学作业单，学习者专注于高度细化或非语境化的问题集的简短工作单让他们以一种孤立的、高度集中的方式练习复杂的语法和程序。一组有针对性的小任务可以练习这些复杂的问题，而不用担心额外的语法错误或其他解决问题的要求，可以减少学生可能被淹没的认知开销。

Jupyter 笔记本的基于单元格的性质使得基于代码的交互式工作表对学生来说是一种干净的运行体验。每个问题的提示都可以写在 **markdown** 单元格中，也许可以引用初始单元格中建立的一个对象或数据文件。例如，一个列表或其他数据结构将被定义在顶部，下面的练习集中在相关的方法和使用语法上。然后，每个答案将由学生在该标记单元下面的单个代码单元中完成。这意味着输出和错误与产生它们的代码在一起，所以成功或错误很容易被追踪到源头。

在后面的章节中讨论的自动交易工具或单元测试的使用，可以被添加到给学生提供关于他们工作的即时反馈。例子或所需的输出也可以用问题的 **Markdown** 单元来重现，以获得进一步的指导。

3.1.5. 3.1.5 Notebooks as notepaper or course packets

笔记本能够用人类的散文和工作代码来表现线性经验，这意味着这些笔记本可以在课堂上作为学生的记事本使用。他们可以捕捉课程或讲座的线性叙述结构，并实际运行他们所记下的代码。这确保了他们写下的东西确实有效，并为他们在今后的作业中提供了强有力的可重复使用的文件。

鼓励学生使用 **Jupyter** 笔记本做课堂笔记，为在课堂上提供支架和支持提供了进一步的机会。提供带有讲座内容或课堂上涉及的其他材料大纲的笔记本，可以成为一种有用的主动学习和参与策略。你可以提供一个基本空白的笔记本，上面只有主题标题，并要求他们在这些地方以自己的风格做笔记。或者你可以选择在笔记本中嵌入参考说明、例子，甚至是小活动，要求他们既做笔记，又和你一起完成例子。

应注意鼓励学生在笔记本中仔细保持他们的代码的线性顺序。后面一章有更多关于 **Jupyter** 的具体注意事项的信息，学生应该注意。

3.1.6. 3.1.6 Notebooks as an app

笔记本甚至在非编码的课堂内容或活动中也有一席之地。交互式的用户输入，如鼠标或触摸屏控制的滑块、按钮、高亮等，允许笔记本用户操作可视化、工具或模型的输入参数，而无需直接编辑笔记本中的任何代码。这些策略支持交互式的计算探索，或者将笔记本变成一个高级的计算器工具，供学生在家庭作业中使用。然后，这些笔记本被视为应用程序，被分发或提供给学生在课堂上使用或自己探索。

这使得你可以将现有的研究工作流程作为基础课的计算模块的一部分让新手学生使用，或者将静态教科书或阅读中的内容模拟成主动学习活动的内容。**Jupyter** 笔记本的适应性和重用性也为学生创造了机会，使他们能够进一步使用这些代码，并将其用于作业或其他研究。

3.1.7. 3.1.7 Notebooks as lab reports or assignments

编程和技术课程可能需要各种作业交付物。学生可能会被要求制作论文、演示文稿、工作代码、分析报告，甚至是艺术或音乐。这些作业中的许多都可以直接在笔记本环境中得到支持。任何书面工作都可以在笔记本环境中用 **markdown** 完成，这对于由数据驱动的交流内容或结合代码内容来说是理想的。例如，一个学生可以在笔记本内写一篇计算性文章，并使用其中一个演示工具在课堂上将报告展示出来，所有这些都使用同一个笔记本。

编码作业可以提交给 **Jupyter** 支持的学习管理系统，并被自动评分（在后面的章节中讨论），提供即时反馈和自动化的评分过程。这为许多课程开辟了自定进度和高度扩展的选项，特别是开放的 **MOOCs** 或大的部分。同时，内联可视化选项意味着带有图形输出的作业可以自成一体，而不需要在文字处理文档中嵌入图像，也不需要在脚本中附加图像集。

笔记本的多种转换和托管选项意味着它们可以通过多种格式共享或提交。例如，转换为 **HTML** 意味着查看内容的开销为零，而对 **markdown** 和 **PDF** 的支持则开辟了可访问性和其他出版平台。

3.1.8. 3.1.8 Notebooks as interactive multimedia platforms

各种媒体格式都可以嵌入到笔记本中，而其他工具更提供了平台，可以更直接地将笔记本与多媒体内容连接起来。教学内容可能被分割成短视频（通常用于翻转课堂），或者各种静态图像可能对一项作业很重要。**Jupyter** 笔记本中的 **markdown** 单元提供了几种放置超链接和嵌入各种媒体的方法。

一些小工具也可用于在笔记本中直接嵌入可播放的音频和视频内容（包括来自流媒体视频服务）。这为学生创造了一个连贯的平台体验，因此他们不必退出或改变屏幕来处理他们的作业和参考这些内容。

其他工具可用于更直接地将笔记本内容与视频指南联系起来。课程中的视频讲座或内容的运行时间可能会很长，而相关的笔记本往往也是一样长。像 **Oriole** 这样的工具提供了一个平台，你可以将视频的时间戳整合到笔记本中，创造互动的视频体验。例如，你可以在笔记本中加入 **Youtube** 视频，在视频前后加入文字和/或编码机会。使用时间码，你还可以引导学生通过视频与笔记本的文字同步。这进一步扩大了创造一个单一的、有凝聚力的互动体验的能力，而不需要学生在不同的材料之间来回走动。

3.1.9. 3.1.9 Notebooks as a demonstration platform

最终，你将不得不在课堂上展示笔记本。这可能是作为如何使用笔记本的演示，更多的是展示传统风格的讲座，创建和编辑代码，或使用互动功能来探索一个实验。在这些情况下，字体大小、组织和可访问性的正常标准是适用的。

在你的演讲中展示实际的笔记本是一个自然的起点。这些内容可能包括来自 **markdown** 和 **LaTeX** 的文本，代码，以及独立的数字和草图。例如，讲师可以展示笔记本，缓慢地滚动材料，与包含代码的单元格进行互动，同时还可以利用数字或模拟（如黑板）草图设备。自定义风格插件可用于改变笔记本的背景颜色、字体和其他浏览方面，以提高演示质量和可访问性。

有几个幻灯片工具可用，它允许你标记笔记本单元内容，以实现更传统的幻灯片演示模式，而不必从标准笔记本中退出。这些幻灯片可以分散在笔记本内容中的特定单元格中，这些单元格将对应于单个幻灯片，其他内容在演示中被忽略。然后，这可以与通常的笔记本界面一起显示，并可以在其他形式的内容之间翻转。

对学生来说，使用 **Jupyter** 笔记本或 **JupyterLab**，以其标准形式阅读笔记本并与之互动是很自然的，这也可用于演示。**Jupyter Lab** 提供了便利和统一性，能够在同一环境中打开和编辑源文件（**.py** 等），没有操作系统或浏览器特有的杂乱。如果使用这种模式进行演示，建议对浏览器进行全面筛选。演示风格差异很大，从完全准备好的笔记本到空白板的现场编码。

许多人发现加入另一种模式是很有用的，如一个物理或数字的“板”，用于自由绘制图表，通过一个数学推导，或其他书面程序任务。笔记本可以是这些演示的一部分，也可以是完整的环境，这取决于你的个人教学风格和内容需要。

应该注意笔记本的展示和演示方式。做现场演示给你几个选择。你可以选择

- 滚动浏览一个笔记本
- 通过按顺序执行单元格来浏览笔记本
- 在一个基本完整的笔记本中填写细节或数值
- 调整或充实有一些内容的笔记本
- 在完全空白的笔记本上添加内容

这些策略中的每一个都在课堂上有用武之地，它们的使用应该根据听众的需要和学习目标来决定。例如，在笔记本上为每个模块或章节的结尾准备了挑战，但内容是空白的，这就使你有机会在课堂上开发代码，但在一个结构中保持你的工作流程有条不紊，并将你的形成性评估直接编入你的演示文稿。纳入参考信息可以使这些文件对学生来说更加完整，并回答常见的问题。关于你可

以加入的内容，有很多可能性。请记住，学生经常会要求你在课堂教学结束后分享这些笔记本的副本。

你对现场编码的关注程度可能会由课堂的领域内容决定。编程课程显然会优先考虑让学生在编写和输入代码方面有更全面的练习。然而，一个研究计算模型的概念课可能会交互式地调整一个模型的参数，并只讨论该数学模型内发生的事情。在这个过程中，没有编写或直接改变代码。

3.1.10. 3.1.10 Notebooks as a live coding environment

现场编码，顾名思义，就是在教学过程中主动写代码。这可能是录制的屏幕广播的一部分，也可能是亲自上阵的课堂。现场编码的过程对学生和教师有几个好处。

展示建立代码例子的过程，展示了代码制作的自然非线性过程，但通过代码例子的逻辑行走，可以减缓教学速度，突出代码中每个元素的原因。

在代码中引入错误（无论是有目的的还是意外的）有一个额外的好处，那就是给演示者一个机会来完成调试过程，并证明完美的代码永远不会在第一次创建。

现场编码也可以是一个提供主动学习经验的机会，通过提供笔记本上的代码，让学生在做现场编码演示之前尝试填写缺失的行。对学生在这一过程中的反馈是一个有用的方法，可以判断学生保留了什么，以及在哪些方面有困难，从而获得及时的教学机会。

形成性评估和预测提示也可以直接纳入笔记本或作为讲座叙述的一部分。创造现场编码的机会可以在任何时候进行，只要有代码块存在，但挑选出特别能说明问题的例子或关键点，并适当地为例子提供支架是很关键的。例如，如果关键的概念是在一个 `for` 循环里面，那么只对 `for` 循环的内部部分进行编码是有帮助的，而不是用支架来压制演示，如设置。然而，相反的情况也可能发生。如果展示了太多的复杂性或脚手架，学习者可能会努力理解脚手架而不是集中在关键概念上。

许多使用现场编码的教师会选择让学生和他们一起编码。这允许他们练习他们正在学习的东西，在他们环境的自然背景下看到它，犯所有正常的错误和错字，并且所有这些都在他们可以提问的环境中（或暂停视频）。积极地与教师一起编码也包括要求他们积极地听教师讲课，并参与到内容中。

滚动或 `shift+enter` 的演示方式不是现场编码，而是现场演示。虽然这些限制或否定了现场编码环境的好处，但加快演示速度或跑完与学习目标无关的代码的好处可能更重要。取消学生对内容的主动参与，最终可能导致他们对课程的脱离，或错过大块的信息。教员应该平衡他们对现场编码和现场演示的加入，以确保学生积极地参与到课程中最重要的方面。

在现场编码课程中，教室里的信息带宽需要仔细管理，特别是当学生试图跟随时。现场编码的节奏大致有三个阶段：准备、打字和解释。这三个阶段很快就会相继进行，但都是独立的阶段。准备是第一个阶段，你停下来解释你要做什么。打字是下一个阶段，你应该边打边说，但只说你正在打的内容。这可以确保学习者在屏幕上看到的和从你那里听到的是一致的。他们很可能会在自己的屏幕和你的屏幕之间来回看，以至于他们在打字的时候往往无法停下来听你说什么。最后一个阶段是停下来，解释你打的是什么，以及当你运行代码时已经或将要发生什么。你可以选择执行代码并解释结果，或者在运行代码之前加入一个形成性评估或预测问题。停下来解释你刚写的代码并浏览结果，让学生有时间跟上你的打字，有时间考虑发生了什么，还有一个自然的地方来问（你也要问）关于发生了什么的问题。

现场编码确实需要练习来适应，但对于你约束你的学习者的节奏和保持与你的学生的接触来说，是非常有力的。

3.1.11. 3.1.11 Conclusion about places

正如你刚才所看到的，笔记本提供了一个灵活的工具，可以用许多方式来实现你的课程目标。笔记本很灵活，你可以从相对被动的学生学习到非常主动的学生学习中使用它们，你可以在你的讲座中或在翻转课堂环境中使用它们。在你的课程中使用笔记本没有单一的最佳方式，你在探索各种选择时，要开始用下一章中描述的各种教学模式来填充你对笔记本的使用。

3.2. 3.2 Before You Begin…

本章主要介绍将笔记本纳入课堂时的课程考虑。有经验的讲师可以选择略过本章的部分内容，而特别关注 Jupyter 笔记本会如何改变他们目前的教学风格。

在你开始将 Jupyter 笔记本添加到你的课程中之前，花些时间。

- 明确你的教学目标
- 理解你的学生
- 制定你的内容策略
- 考虑学习环境的背景

Jupyter 笔记本是一种工具；它在这种环境下的使用要服从于教学的期望。设定学习的期望值取决于你的目标、你的学生、内容、学习环境背景和你。

3.2.1. 3.2.1 Identify Your Teaching Goals

就像创造一个建筑或机器人或书籍一样，重要的是在开始这个过程时要有一个明确的目标，即你要创造什么和为什么。这个”和为什么”可能是你在整个过程中做出的最重要的决定，它将（或至少应该）指导接下来的所有决定。这里的”为什么”不是指为什么要使用 Jupyter 笔记本，而是指你对学生的目标。

目标是教他们批判性思维技能，还是教他们如何执行一组特定的函数来解决问题？目标是让他们能够将数学概念转化为现实世界的应用，还是教他们如何编码？要具体明确，然后让答案指导你的决定。

3.2.2. 3.2.2 Understand Your Students

你的学生是决定何时和如何在你的教学中使用笔记本和其他工具的核心。如果你把教五年级学生和教研究生作比较，那么这些决定的一个明显的例子就是选择。然而，还有一些更微妙的差别，你也要注意和监督。例如，在一个教室里，你会发现学生的背景和技能水平有差异。根据不同的领域，有些学生可能有丰富的多语言编码经验，而有些学生可能是第一次进行计算探索。让所有不同经验水平的学生都参与进来并感到兴奋，这是一个挑战。

3.2.2.1 Learn About Your Students

在许多情况下，你在上课或研讨会的第一天之前，对你的学生几乎没有背景资料。你必须迅速学习，并准备调整教学，以适应出席会议的学生，而不是你希望或预期的学生。了解学生的一些关键考虑因素。

- 动机：你的学生为什么要参加？他们在研讨会或课堂之外的目标是什么？他们是否很快就会应用他们所学的知识，还是几个月或可能几年都不会？

- **入门技能:** 学生在参加教学时有哪些技能，包括技术（如基本的计算机、编码、计算）和社会心理（合作、展示、提问）？
- **先前的主题知识:** 具体到教学内容，他们来时已经知道什么？对内容的态度。他们是对学习感到兴奋，还是对学习内容感到紧张？他们对自己在课堂上的成功有多大信心？
- **对授课形式的态度:** 他们对授课形式（如讲座、翻转课堂、实验室）和/或技术（如学习管理系统）是否有积极或消极的态度？
- **学习偏好:** 学习者对主动学习的经历感到满意吗？（例如，与团队合作，与教师互动，在学习中使用技术，等等。）
- **群体特征:** 从这些考虑因素来看，这个小组有多大的差异？他们以前是否在一起接受过指导？你有一个小班还是一个非常大的班级？

在课前做一个简单的在线调查，对了解你的听众非常有帮助。

3.2.3. 3.2.3 Cultivate Student Study Skills and Learning Strategies

不要假设你的学生已经掌握了必要的学习技能，能够有效地从你的课程中学习。从记笔记的技能（即，寻找模式，辨别什么值得写下来，与读物的联系，等等）到多任务处理（即，知道什么时候可以在学习时打开他们的电子邮件，什么时候他们真的必须集中精力），这些技能往往需要在学习经历中发展。

对于大多数学生来说，通过笔记本等工具进行学习，与他们以前的学习经历有很大的不同—特别是在传统的高中或大型讲座的本科课程结束后。为了学习的明确目的而积极地与技术互动（即，不只是与朋友交往）是有价值的，但却不常见。

花时间与你的学生一起工作，帮助他们建立基础，使他们能够从你所创造的经验中获得最大利益。在教学的早期，与他们讨论如何使用笔记本，以及他们必须如何调整他们的学习策略（在哪里、何时、如何学习），以便最好地学习内容和实现教学目标。

3.2.4. 3.2.4 Develop Your Content Strategy

教学内容应该支持你为该课程确定的教学目标。有时目标和内容是同义的，但有时却不是。例如，如果你的目标是让学生能够计算联合概率，那么你的内容将是同义的，是关于计算联合概率的。而在其他情况下，你的目标可能是发展与逻辑谬误有关的批判性思维能力，而你用来实现这一目标的内容是对政治话语的分析，这不一定是同义的。

在这两种情况下，内容指导着为什么、何时、以及如何使用笔记本来实现目标。请记住，笔记本是我们可以用来实现目标的解决方案；它们不应该与目标本身相混淆，即使这样它们也是密切相关的。

你的教学内容并不局限于你要教的东西；它涉及到活动、练习、反馈机会和评估。每一个都支持你的教学目标，并从 **Jupyter** 笔记本内外的不同工具的使用中获益。例如，如果你的内容目标表明，学生在准备进入另一个内容主题之前需要一些特定的知识（例如，什么是逻辑谬误），那么你应该在选择哪种工具（例如，笔记本中的填空项，或在课堂上对学生的口头提问，或自我反思）最适合你的教学步骤之前评估这些知识。

对教学的目标和相关内容有一个全面的概述，是对如何、何时、何地以及为何在教学中使用笔记本做出正确决定的关键。

3.2.5. 3.2.5 Consider the Context of the Learning Environment

教学发生在许多有趣的情境中，但学习并不局限于我们所构建的教学文本。学生在教学前、教学中和教学后都在学习，而笔记本可能是学生在所有这些情况下的一个综合组成部分（见下文）。

在我们的教学情境中，也就是我们可能对学生的学习影响最大的时候，我们也有很多选择来提供学习经验。例如，作为教员，你可以把笔记本作为翻转课堂的预习作业；在面授或在线课堂上使用笔记本来演示或为学生提供练习；和/或在教学结束后很长时间内使用笔记本来做家庭作业、评估以及学生可以使用的资源。所有这些都为您使用笔记本创造了不同的背景。

在决定使用笔记本时，你应该考虑的其他方面的背景，包括教学环境和学习者的后期表现环境。

3.2.5.1 Instructional Environment

如今，教学环境可能相当复杂，涉及多个方面。例如，教学可能包括在线视频、教室里的简短讲座，然后是实验室里的小组活动。这些都是独特的环境，其特点可能会影响你如何、何时、何地以及为何使用笔记本。

- **课堂** 学生将如何参与教学内容？考虑到教室的布局（如小桌，讲堂），是否有机会让学生参与其中？
- **实验室**：实验室环境的物理结构如何为同伴学习和分享提供机会？有哪些硬件和软件工具可以提供给学生？
- **翻转课堂**。学生必须从预习活动中获得哪些必要的知识和技能？视频在翻转课堂中的作用是什么？学生是否准备好了在翻转课堂方法中独立学习的学习技巧和策略？所有的学生都能在课前接触到教材（可能是笔记本）吗？学生在教室外是否有机会使用电脑？在课程中如何激励和评估预读/预看？
- **在线课堂**。学生是否准备好了学习技巧和策略，以便在在线课堂上独立学习？

3.2.5.2 Performance Environment

希望在教学结束后，你的学生会应用你所教的知识，而他们应用你的教学的环境可能有很大的不同。你要考虑教学环境和表演环境之间的任何差异会如何影响学生应用所学知识的能力。

- **组织/管理方面的支持**: 学生在使用你所使用的教学和工具时是否会得到支持？
- **社会支持**: 如果他们通过小组项目学习，他们是否也能在自己的工作中应用这些指导？
- **物理方面**: 他们是否可以使用与教学中相同或相似的工具和资源？例如，如果学生只在笔记本上学习，他们是否也能在雇主使用的 IDE 中应用所学知识？

3.2.6. 3.2.6 Crafting an experience: choose the right tool and approach for the task

作为一名教师，你为你的课堂创建了许多类型的学习对象。**Jupyter** 笔记本可以用来展示许多类型的信息，从幻灯片到书籍章节到家庭作业。你可以想象，你在幻灯片、书本章节、家庭作业和课堂活动的工作表内呈现的信息可能有很大的不同。**Jupyter** 笔记本可以用于所有这些活动，但这都是在同一类型的文档平台中。

这种在 Jupyter 笔记本环境中可以自由表达的方式会给设计过程带来很大的压力和决策疲劳。你打算在一个空白的笔记本中制作一个讲座文件，而一个空白的笔记本用于课堂活动，在第一次创建时看起来都是一样的。

内容和设计的决定应该由课程的目的和听众的需要来驱动。有多少次你问了别人一个问题，而他们却以完全没有帮助的方式回答？也许你问的是如何实现某个东西，但他们的回答是概念性的，或者你问的是关于某个东西存在的理由，但他们给你的是一个语法信息。

学习目标和教学实践是复杂的，而且往往是针对具体领域和技能的，但关于它们的决定又回到了一个基本问题：你在这里想做什么？作为教员，你不仅要为学习体验选择合适的技术，还要选择合适的活动来完成当天和课程的学习任务。

学生们有各种各样的经验来帮助学习过程。有些主题是概念性的，学生需要时间去尝试并建立直觉；有些是纯粹的语法，需要参考和练习，以建立他们在使用过程中的内化知识，有些则介于两者之间。

在学习过程的节奏中，学生们独立工作、共同工作、阅读、制作、聆听、解决问题，以及挣扎。每项活动都应该有一个目的，作为一个更大的经验的一部分。就像餐馆或企业主为他们的顾客精心设计体验一样，你也在为你的学生设计一种体验。

这种体验的设计不是一个简单的过程，而是由你对你的听众、领域和你自己的指导能力的专业知识驱动的。在这一切中，你的核心观点应该是将各种元素融合在一起，使之成为一种体验，从而达到你所期望的学习目标。

也许对于上面的部分，可以添加一个主要决策考虑因素的方框图或文字云的视觉。

3.2.7. 3.2.7 Transitioning to and from Jupyter

虽然本书的重点是在教育中使用 Jupyter 笔记本，但我们认识到，学生有不同的背景和对编码的熟悉程度。并非所有的学生都准备好从传统的讲授式课堂直接跳入笔记本；而其他学生可能有大量的编程经验，可能更接近于准备过渡到 IDEs。

根据你对目标和背景的分析，你要考虑引入和退出笔记本作为实现目标的工具的适当指令点。随着你的学生获得编程经验，他们可能会对使用 IDE 更感兴趣。同样地，你可以选择先介绍一个更传统的 IDE，然后再介绍 Jupyter 笔记本。可以支持这种过渡的练习包括布置作业，让他们在 3、4 个不同的 IDE 中完成同样的任务，然后在他们决定哪种环境最能支持他们当时的雄心时，对这些经验进行反思。

3.2.8. 3.2.8 Conclusion about teaching practices

在本章中，我们试图将 Jupyter 笔记本的使用纳入良好的课程设计过程。笔记本是许多教师可以用来提高学生的参与度、参与度、理解力和表现力的工具，但这并不是说每门课程和每节课都应该使用笔记本。让你的课程目标决定何时以及如何最好地使用笔记本来实现这些目标。

4

Chapter 4 A catalogue of pedagogical patterns

4.1. 4.1 Introduction

在这一章中，我们介绍了一系列与 Jupyter 的教学和学习特别一致的模式。每个模式都是针对特定的学习目标、受众和教学形式。考虑到这些，我们描述了每个模式及其支持学习目标的教学特点，介绍了一个实际的例子，并在每个模式的结尾处说明了你想要注意的任何潜在陷阱。

4.2. 4.2 Shift-Enter for the win

Notebook1

(part1.ipynb)

描述。学习者不是阅读关于某一主题的静态章节，而是阅读和执行代码，以及可能与一个小部件互动，以探索概念。从一个完整的笔记本开始，教师或学习者通过键入 **SHIFT + ENTER** 来逐个浏览笔记本的单元。

例子。笔记本（或笔记本的集合）可以作为一个主题的静态教科书的替代品。

学习目标。这种模式可以用来介绍一个主题或促进对一组工具的认识。此外，它还可以作为提供建议编程接口（API）之旅的文档。

受众（s）。根据笔记本的风格，这种模式可用于不同的编程能力。

形式（讲座/实验/...）。这种模式可以作为静态教科书的替代品。在一个教程中，一个完整的笔记本可以用来提供一个软件包的应用编程接口（API）的参观。

特点。这种方法的一个好处是，学习者有一个完整的工作实例，他们可以在此基础上改编或构建。它提供了比静态教科书更丰富的互动机会。

隐患。这种方式不能促使学生参与。让一个班级通过笔记本进行互动，可能会导致一些学生比其他学生完成得更快（例如，通过 **SHIFT+ENTER** 进行竞赛）。将长的笔记本分成许多小的笔记

本，可以帮助掌握讲课的节奏。让一个主笔记本作为目录，可以帮助学生在课堂上导航。笔记本可以在 **markdown** 单元格中被链接为。

[笔记本 1](part1.ipynb)

4.3. 44.3 Fill in the blanks

描述。为了将注意力集中在工作流程的某一方面，可以将工作流程的支架和大部分内容布置出来，并删除一些元素，目的是让学生（或在演示时由教师）填补这些部分。这个练习可能伴随着代码应该通过的一个小测试，或者一个情节，或者代码应该产生的值（如果正确的话）。

例子。计算中的一个基本概念是使用 **for** 循环来累积结果。一个演示累加器的填空练习可以列出初始化，提供 **for** 循环的骨架并包括绘图代码，目的是让学生在 **for** 循环中写出更新步骤。

相关模式。这种模式与目标练习相似；不同的是，目标练习往往侧重于多步骤过程中的一个较大的步骤。填空练习往往规模较小，而且更直接。

学习目标。这种模式将注意力集中在任务的某一环节上，并提供演示该环节如何融入大局或更大的工作流程的好处。对于带领学生参观一个 **API**，要求他们使用软件的文档，或者将注意力集中在一个多步骤计算模型的一个方面，这都是一种有效的方法。

受众 (**s**)。这种方法可用于各种学生，从那些刚开始接触计算概念的学生到那些有丰富经验的学生。

形式（讲座/实验/...）。作业和实验可以采用这种方法（可以用 **nbgrader** 来帮助评分）。它也可以在讲座或辅导中使用，由教师演示如何填空。

陷阱。有些学生认为这种方法并不吸引人。特别是，如果练习对于学生的编程能力水平来说过于简单，就会被认为是“补课”任务。

4.3.1. 4.4 Target Practice

描述。目标练习模式将学习者的注意力集中在一个多步骤工作流程的一个组成部分。教员提供所有的工作流程步骤，只有一个练习的重点；学生将在笔记本上实现“目标”步骤。

例子。在一项气候科学作业中，给学生的笔记本提供了从公共数据库中获取和解析 20 年每小时平均温度数据的代码。学生被要求设计一种算法来计算年平均温度和标准偏差。在这之后，还向学生提供了绘图代码，该代码绘制了带有显示标准偏差的误差条的年度温度。

相关模式。这种模式类似于 [填空] 练习。填空”的练习通常比较小，也比较直接，而“目标练习”往往比较大（例如，在一个多步骤过程中的整个步骤）。

学习目标。目标练习的目的是将注意力集中在工作流程的一个组成部分上，并练习解决这个组成部分的技能。它也可以用来反思在综合工作流程或分析的目标步骤中所作选择的更广泛的后果。

受众。这种方法假定学习者具有一定的编程能力，因为他们通常被要求从头开始练习这个步骤。

形式（讲座/实验/...）。这种方法很容易在作业或实验室中使用。它也可以用在课堂演示中，由教师对缺少的部分进行现场编码。如果前面的步骤已经在前面的讲座中讨论过了，那就很有好处。

陷阱。由于在实施方面有更多的自由，这种方法通常比填空法更有吸引力。然而，从更多的“白板”开始，可能需要更多的教师投入，以使学生开始。单元测试和指向有用的库函数通常是有帮助的，但可能会过度限制解决方案的空间，从而降低期望学生的创造力和解决问题的水平。指导的

数量应该根据班级的情况仔细调整，通过对形成性评估的提示进行调整。在小组中工作可以帮助减轻学生在开始时遇到困难或追求切入点的风险。

4.3.2. 4.5 Tweak, twiddle, and frob

说明。学生们会得到一个带有工作实例的笔记本。他们首先阅读文本，运行代码，并解释结果。然后，他们被要求做一系列的修改，并再次运行代码；这些修改可以是小的（调整），中等规模的（扭动），或更大的（青蛙）。关于这些术语的起源，见《新黑客词典》(Raymond, 1996)。

提供各种规模的操作，使学生能够以适合其背景和风格的方式与笔记本互动。对技术感到不知所措的学生可以从小的、安全的改变开始，享受立即的成功，并逐步提高。有更多经验或更少耐心的学生可以进行更彻底的改变，通过“破坏性测试”来学习。

例子。在机器学习中，实现一个有效的算法有很多步骤。

理解一个问题确定适当的机器学习算法，以创造期望的结果确定数据和特征集优化机器学习算法的配置我们可以编写机器学习笔记本，让学生修改参数并在多个细节层面上进行互动。

拨弄超参数，快速看到结果的微小改进。调整特征集，创建新的模型以产生更大的影响通过用新的算法或新的版本替换机器学习算法来进行 **frob** 这种模式特别适用于有很多参数的例子。

学习目标。这种模式通过看到参数之间的关系以及它们对结果的影响，帮助学生获得领域知识。它还可以帮助学生学习新的笔记本使用模式。

这种模式类似于“作为应用程序的笔记本”；不同的是，在这种模式中，学生可以看到更多的代码，如果他们将来要做更大的改动，这可以帮助他们定位。

受众 (**s**)。这种模式可以适用于没有编程经验的学生；他们只需要能够编辑单元格的内容和运行一个笔记本。它也可以与没有领域背景的学生一起工作；他们可以通过探索参数的效果来了解领域的情况。

形式（讲座/实验/...）。这种模式适合于研讨会的形式，即引导学生在笔记本上进行有时间限制的实验机会。它也适合于结对编程，一个导航员可以提出修改建议，一个驾驶员可以实现这些建议。

特点。可以帮助学生克服对破坏代码的焦虑，并建立自我探索的舒适感。

陷阱。这种模式的一个危险是，学生可能难以开始，所以你可能会建议一些例子；然而，另一个危险是，如果你提供了例子，学生会按照他们被告知的去做，而不能进行探索。第三个危险是，学生所做的改变可能过于无序；在这种情况下，参数的效果可能会在混乱中丢失。

启用技术。理想情况下，学生应该在某种版本控制系统中工作，如果他们破坏了什么（而且不知道如何修复），可以让他们恢复到以前的版本。请注意，撤销和重做，即 **Ctrl-z** 和 **Ctrl-y**，可以用来在每个单元格内深层穿越历史，但不能跨单元格。

4.3.3. 4.6 Notebook as an app

描述。笔记本可以用来快速生成用户界面，学生和教师可以通过滑块、输入框和切换按钮与代码互动。代码可以运行数值模拟或进行简单的计算，其输出通常是一个图形或图像。

例子。在地球物理学中，直流电阻测量涉及到将两个电极连接到地面上，电流通过这些电极被感应。电流在地球上流动，其行为取决于地下结构的电阻率；电流在电阻周围流动，被导入导体。在导体和电阻之间的界面上，电荷积聚，这些电荷产生电动势，我们在地球表面测量。这些步骤中的每一步都可以通过模拟来展示，学生或教师建立一个模型，并查看电流、电荷和电动势的情况。

相关模式。自上而下的顺序

学习目标。这种方法可以有效地关注特定领域的知识，促进对模型或计算的探索。

受众 (**s**)。这种方式对于只有极少编程经验的学生来说是有效的，因为他们不需要读、写、或看代码。

形式（讲座/实验/...）。在讲座中，这种风格的笔记本可以被教师用来有条不紊地一步步讲解一个概念。它对促进课堂参与也很有用，因为学生可以提出不同的参数选择，教师可以根据学生的问题调整输入参数。

在实验室或作业中，笔记本可以作为一个“应用程序”使用，围绕它提出问题和练习。

特点。笔记本作为应用程序可以用来促进学生在课堂上的参与。在实验室、作业或课堂活动中，这种方法降低了学生探索复杂模型的门槛。

陷阱。重要的是要有结构良好的练习和问题，让学生用应用程序来解决。与任何应用程序一样，简单地要求学生玩它并不能促进有效的参与。

在为学生安排练习时，我们建议将说明和问题放在一个单独的文件中，而不是笔记本中。如果学生把笔记本看作是一个应用程序，他们往往想与之互动，而不是阅读它。通过将说明和问题与笔记本放在一起，他们可以在阅读时看到应用程序的内容。

这种方法不打算用于发展学生的编程技能。

使能技术。小工具，特定领域的库，如模拟工具。

4.3.4. 4.7 Win-day-one

描述。

例子。为了解决一个使用有限体积方法的数值模拟，必须设计一个网格，形成微分算子，设置边界条件，生成右手边，然后解决这个系统。当然，每个步骤都有重要的考虑因素。即使是一个中等规模的问题，为了保持内存的使用，稀疏矩阵也是必要的，网格必须适当设计，以满足边界条件，求解器需要与系统矩阵的结构兼容。这些细节对组装数值计算至关重要，但如果在前面介绍，它们可能会使对话不堪重负。

在双赢的方法中，首先向学习者展示一个简明的例子，其中的许多细节被抽象为函数或对象。例如，`get_mesh`、`get_pde` 和 `solve` 等方法就抽象出了网格设计、创建微分算子和求解方程组等细节。在随后的笔记本中，我们将有条不紊地处理工作流程，并讨论每个组件的内部工作原理。

相关模式。自上而下的顺序，通过实例证明，通过反例反驳

学习目标。这可以成为介绍复杂过程的有效方法，为每个组成部分如何配合提供背景，并集中注意力。

受众。这种方式对从有一些编程经验的学生到有丰富经验的学生都很有效。

形式（讲座/实验/...）。这种方式对辅导和研讨会很有效，可以在多个讲座中使用。这在介绍新课题时可以起到帮助吸引学生的作用，因为他们在学习过程的早期就取得了显著的效果。

陷阱。“胜利日”的一个危险是，“胜利”是压倒性的（太多细节）或太神奇的（太少细节）。需要选择一个适当的细节水平，以便工作流程的每一个组成部分都能被展示出来，但要在高层次上。

4.3.5. 4.8 Top-down sequence

说明。特别是在 STEM 领域，默认的演示顺序是自下而上的，这意味着我们在学生学习如何使用它们或它们的用途之前，先教给学生事物的工作原理（有时还要证明它们的工作原理）。

笔记本提供了自上而下展示主题的机会；也就是说，学生在学习工具如何工作之前，先学习工具的用途以及如何使用它。

例子

在数字信号处理中，最重要的思想之一是离散傅里叶变换，它依赖于复数运算；在自下而上的方法中，我们将不得不从教授或复习复数开始，这不是特别吸引人。

与在纸上写数学相比，在笔记本上，学生可以使用一个为他们做离散傅里叶变换的库，因此他们了解它的用途，并在我们要求他们做理解它的工作之前看到学习它的价值。

一些重要的方法在本质上是泄漏的抽象，需要用户的专业知识来有效和可靠地使用。这通常是因为真正可靠的解决方案（如果存在的话）对于常见的情况来说是不成比例的昂贵。数值积分以及离散化和求解微分方程的方法往往属于这一类。除了在深入研究细节之前获得直觉之外，自上而下的模式还可以用来暴露这些漏洞，作为充分理解方法的动力，以解释和纠正这些缺陷。例如，我们可以通过展示一个通过了一些一致性测试的求解器，但在一般情况下不收敛（或次收敛）来激励收敛分析和验证（Roache, 2004）；或者通过展示一个已经被验证为平稳解的求解器在有奇点或不连续的问题上产生错误的结果来激励保守/兼容离散。例如，考虑 Gibbs 和 Runge 现象、Gram-Schmidt 的不稳定性（Trefethen & Bau, 1997）、熵原则（LeVeque, 2002）、涡流粘性（Mishra & Spinolo, 2015），以及 LBB/inf-sup 稳定性和“变异犯罪”（Brenner & Scott, 2008；Chapelle & Bathe, 1993）。

学习目标。这种模式有利于在介绍技术领域内容之前建立直觉、背景和动机，而不是在实施细节经常占据中心位置的环境中建立起来。

受众。这种模式对那些编程能力有限的学生很有效，因为他们可以使用一个库，不用写太多的代码就能看到结果。

形式（讲座/实验/...）。这种模式可以在一节课或家庭作业中使用，也可以在课程中分散使用；例如，学生可以在第一天使用一个工具，在最后一天发现它是如何工作的。

特点。向学生展示价值，并迅速奖励他们的注意力（见 **Win-day-one**）。

陷阱。这种模式的一个潜在危险是，如果学生认为他们已经了解了这个工具的用途和使用方法，那么他们学习这个工具的积极性可能会降低。这种危险可以通过说明了解它的工作原理的额外好处来减轻（假设确实有这样的好处，但仅断言知道它的工作原理就一定更好，是不够的）。学生在尝试解决问题时发现的“有趣的”失败模式（见上面的例子），对于激励学生深入理解是很有帮助的。

4.3.6. 4.9 Two bites at every apple

Pitfalls: It can be difficult to construct questions for each audience that require equal amounts of difficulty.

描述。这种模式涉及到写一个可以同时从不同角度解决多个受众的活动。在面对混合的学生观众时，这可能是很有力的。

例子。假设你有一群学生，其中一些是计算机专业的学生，一些是物理专业的学生，要求他们想出两个计算一个区域中心点的表达式。计算机专业的学生的任务是用 **for** 循环将离散的区域片断加起来，物理专业的学生的任务是使用积分定义。当学生想出他们的表达方式时，他们可以与其他背景的人配对，他们可以尝试解释他们的方法如何与对方匹配，并比较他们的最终答案。

学习目标。有能力将一个领域/语言翻译成另一种语言。向来自不同领域的人解释复杂的话题。

受众。由不同背景的群体组成。

形式。这种形式涉及个人和小组工作，但可以在实验室或讲座中使用。基本的笔记本将包括对问题的概述，然后提出问题，这些问题的答案是相同的，但针对不同的听众进行表述。可以有一个包含两个问题的单一笔记本，以便学生在理解后可以填写他们的同伴的解决方案，或者可以为每个小组提供单独的笔记本，这样他们就不会被另一个问题分散注意力。

特点。实践证明，小组合作和同伴教学不仅能有效地巩固学生的知识，还能向学生介绍新的概念。

陷阱。为每个听众构建需要同等难度的问题可能很困难。

4.3.7. 4.10 Coding as translation

Converting mathematics to code is a critical skill today that many students, especially those without strong programming backgrounds, struggle to do. Explicitly taking an equation and translating it step-by-step to the code can help these students make the transition to attaining this skill.

描述。将数学转换为代码是当今的一项重要技能，许多学生，特别是那些没有强大的编程背景的学生，都在努力做到这一点。明确地将一个方程一步步转化为代码，可以帮助这些学生过渡到达到了这一技能。

例子。假设你想展示矩阵-向量乘法从方程到数字计算的转换。这将涉及到设置和解释数学，并建议用循环来代替和，并正确地初始化和。

学习目标。将数学转化为代码（反之亦然）。

听众。了解理论但在编程方面有困难的学习者。

格式。这种类型的模式通常最好是以笔记本的形式出现，上面有一些说明性的文字，可能还有一些支架式的代码，这样学生就可以专注于关键领域。这可以像实验室练习一样容易完成，也可以在讲课时留出一些时间让学生在前进前自己解决。这种模式的关键是数学符号（如求和）和代码（如 **for** 循环）之间有明确的联系。

特点。这种模式可以降低编程知识较少的学生承担更复杂任务的障碍。

陷阱。如果练习没有适当的脚手架，即太复杂，学生可能会被拒绝。如果代码例子太复杂，步骤太多，就更是如此。例如，在上面的例子中避免复合运算符（`+=`）可以帮助学生保持。

4.3.8. 4.11 Symbolic math over pencil + paper

描述。你的目标是传达对一个受复杂数学系统支配的物理系统的理解。为了揭示系统的基本行为，必须进行代数运算，但如何进行代数运算并不是本课的目标。在这种情况下，你想看到代数结果，然后教给学生这个系统的基本含义。

例子。水动力学的欧拉方程是一个关于质量、动量和能量守恒的偏微分方程系统。它们的数学特性承认波的解决方案，系统的矩阵形式的特征值和特征向量对于理解系统中密度、压力、速度等的物理行为非常重要。用纸笔算出特征值是很乏味的，也不是本课的目的。在这种情况下，我们可以使用像 SymPy (Meurer 等人, 2017) 这样的符号数学库来为我们做数学分析，找到系统的特征值和特征向量，然后我们可以用这个结果来继续我们对系统的理论讨论。

学习目标。学生将看到如何进行理论分析中出现的符号化数学运算。

听众：想集中精力学习的 STEM 学生。希望专注于理解一个数学系统而不担心代数细节的 STEM 学生。

格式。这作为一本笔记本，作为主要讲座的补充，效果很好。因为讲座的目标是理论，所以笔记本可以 [TODO: complete]。

特点。把对当前讨论来说是次要的数学细节抽象成一个单独的单元，让学生自己去探索。

陷阱。只有在代数对主要的学习目标并不重要，而只是为了达到主要目标而必须做的事情的情况下，这种方法才行得通。

4.3.9. 4.12 Replace analysis with numerical methods

描述。一些用数学分析难以理解的观点，用计算机模拟和数值方法就很容易理解。

在通常的介绍中，学生看到并学会在一系列简单的例子上进行数学分析，只有在必要时才诉诸于数值方法。在另一种模式中，学生跳过分析，从模拟和数值方法开始，在获得实践经验后可选择访问分析。

例子。在统计学中，假设检验是一个中心思想，对学生来说是出了名的难以理解。学生们学会了在一系列特殊情况下计算 p 值的方法，但他们中的许多人从来没有理解过这个框架，或者 p 值意味着什么。另一种方法是通过模拟来计算抽样分布和 p 值；据传闻，许多学生表示这种方法使框架更加清晰。这样的模拟也可以被用来解释大多数学生和教师所持有的错误观念 (Haller & Krauss, 2002)。

同样地，在排队理论中，有一些分析结果在狭窄的条件下适用；当这些条件不适用时，就没有分析解决方案。然而，排队系统适合于模拟和可视化，在模拟中很容易探索广泛的条件。

同样，对于微分方程来说，只有少数特殊情况有分析解；绝大多数有趣的、现实的问题都没有。

学习目标。这种模式主要是帮助学生更清楚地看到该领域的大概念，但它也是一个发展他们编程技能的机会。它还为学生提供了在现实世界中遇到类似问题时可能需要的工具，在现实世界中，分析方法往往是不适用的，脆弱的，或复杂的，无法有效使用。

受众 (s)。这种模式要求学生对编程有一定的适应性，尽管他们有可能从看到的例子中得到一些好处，而不去实施它们。非编程人员可以通过准备好的笔记本来使用这种模式；见《第一天的胜利》。

形式（讲座/实验/...）。这个模式可以用于课内活动或家庭作业。

特点。学生可以理解一般的想法，而不会被特殊情况的细节所困扰；而且他们能够探索更多有趣和现实的例子。

陷阱。如果学生不是舒适的程序员，他们可能会被实施细节和调试问题所困扰，而完全错过领域内容。重要的是，实施工作的范围要适合班级中所有的学生。结对编程可以帮助减轻这些问题，特别是如果每对学生中至少有一个有编程技能的学生，而且如果学生被指导有效地结对编程（不要让更有经验的学生占主导地位）。

4.3.10. 4.13 The API is the lesson

描述。当学生使用软件库时，他们会接触到构成应用编程接口 (API) 的函数和对象。学习 API 可以是认知上的开销；也就是说，学生必须学习材料以完成计算上的工作，但这对他们对主题的理解没有帮助。但是，API 也可以是课程；也就是说，通过学习 API，学生隐含地学习了预期的内容。

例子。在数字信号处理中，最重要的想法之一是信号的两种表现形式之间的关系：作为时域的波和作为频域的频谱。假设 API 提供了两个对象，叫做 Wave 和 Spectrum，以及两个函数，一个是接收 Wave 并返回 Spectrum，另一个是接收 Spectrum 并返回 Wave。通过使用这个 API，学生们隐约了解到 Wave 和频谱是相同信息的等价表示；给定其中一个，你就可以计算另一个。

相关的模式。自上而下的序列

学习目标。这种模式有助于将学生的注意力从实施细节转移到领域内容上。

听众。如果学生有一些使用库和探索 API 的经验，这种模式是最有效的。

形式（讲座/实验/...）。这个模式的 TODO：完整

陷阱。这种模式的一个危害是，学生有时会认为学习 API 的成本很高，而没有意识到其好处。可能有必要帮助他们看到，学习 API 是课程的一部分，而不仅仅是开销。

4.3.11. 4.14 Proof by example, disproof by counterexample

Description: In many classes, students see general results derived or proved, and then use those results in programs. Notebooks can help students understand how these results work in practice, when they apply, and how they fail when they do not.

Example: In statistics, the Central Limit Theorem (CLT) gives the conditions when the sum of random variables converge to a Gaussian distribution. Students can generate random variables from a variety of distributions and test whether the sums converge and how quickly.

The classical Gram–Schmidt is unstable while the modified method is stable. Students can find matrices for which this instability produces obviously unusable results. They can also find matrices for which modified Gram-Schmidt produces unusable results due to its lack of backward stability, and this can be used to motivate Householder factorization and discussion of backward stability.

Some numerical methods for PDE converge with an assumption on smoothness of coefficients. Students can show how violating these assumptions leads to erroneous solutions, thus motivating discussion of conservative/compatible methods that can converge in such circumstances.

Learning goals: This pattern is primarily useful for developing mathematical or domain knowledge, but students might also develop programming experience by writing code to run the examples and test the outcomes. This is especially true if the space of (counter-)examples is “small”, such that principled exploration (e.g., by finding an eigenvector, running an optimization algorithm, or searching a dictionary) is beneficial.

Audience(s): This pattern requires students to have some programming experience.

Format (lecture / lab / …): This pattern can be used for in-class activities or homework.

Features: Helps students translate from theoretical results to practical implications, and to remember the assumptions and limitations of theory.

Pitfalls: This pattern requires additional time and student effort on a topic that might not deserve the additional resources.

描述。在许多课程中，学生看到的是推导或证明的一般结果，然后在程序中使用这些结果。笔记本可以帮助学生了解这些结果在实践中是如何工作的，什么时候适用，以及不适用时如何失败。

例子。在统计学中，中心极限定理（CLT）给出了随机变量之和收敛于高斯分布的条件。学生可以从各种分布中产生随机变量，并测试其总和是否收敛以及收敛的速度。

经典的 Gram-Schmidt 方法是不稳定的，而修改后的办法是稳定的。学生们可以找到这种不稳定性产生明显不可用结果的矩阵。他们也可以找到修正的 Gram-Schmidt 由于缺乏后向稳定性而产生不可用结果的矩阵，这可以用来激励 Householder 因式分解和讨论后向稳定性。

一些 PDE 的数值方法收敛时有一个关于系数平滑性的假设。学生可以展示违反这些假设如何导致错误的解决方案，从而激发对在这种情况下可以收敛的保守/兼容方法的讨论。

学习目标。这种模式主要用于发展数学或领域知识，但学生也可以通过编写代码来运行例子和测试结果来发展编程经验。如果（反）例的空间是“小”的，这样的原则性探索（例如，通过寻找特征向量，运行优化算法，或搜索字典）是有益的，那就更是如此。

受众 (s)。本模式要求学生有一定的编程经验。

形式 (讲座/实验/...)。该模式可用于课内活动或家庭作业。

特点。帮助学生从理论结果转化为实际意义，并记住理论的假设和限制。

陷阱。这种模式需要在一个可能不值得增加资源的主题上花费额外的时间和学生精力。

4.3.12. 4.15 The world is your dataset

Description: Notebooks provide several ways to connect students with the world beyond the classroom: one simple way is to collect data from external sources. Data is available in many different formats that require different software tools to collect and parse.

If a dataset is available in a standard format, like CSV, it can be downloaded from inside the notebook, which demonstrates a good practice for data integrity (going to the source rather than working with a copy) and demystifies the source of the data.

For data in tabular form on a web page, it is often possible to use Pandas to parse the HTML and generate a DataFrame. Also, for less structured sources, tools like Scrapy can be used to extract data, “scrape”, from sources that would be hard to collect manually, and to automate cleaning and validation steps.

Examples: Datasets like the National Survey of Family Growth are available in files that can be downloaded directly from their website, but the terms of use forbid redistributing the data. So the best way for an instructor to share this data is to provide students with code to input into a notebook cell, which, when executed, will download the data set the first time the student runs the notebook.

Many Wikipedia pages contain data in HTML tables; most of them can be imported easily into a notebook using Pandas.

Sources of sports-related statistics are often embedded in large networks of linked web pages. Tools like Scrapy can navigate these networks to collect data in forms more amenable to automated analysis.

Audience(s): Students with limited programming experience can work with datasets in standard formats, but scraping data might require more programming experience.

Format (lecture / lab / …): This pattern lends itself to more open-ended project work where students are responsible for identifying data sources, collecting data, cleaning, and validating, but it can be adapted to more scaffolded work (see Target Practice).

Features: Contributes to students' feelings of autonomy and connectedness.

Pitfalls: A hazard of this pattern is that students can spend too much time looking for data that is not available. They might need coaching about how to make do with the data they can get, even if it is not ideal.

Enabling technologies: Pandas, Scrapy, R, ROpenSci packages

描述。笔记本提供了几种方法将学生与教室以外的世界联系起来：一种简单的方法是收集来自外部的数据。数据有许多不同的格式，需要不同的软件工具来收集和解析。

如果一个数据集有标准的格式，如 CSV，它可以从笔记本内部下载，这显示了数据完整性的良好做法（去到源头而不是用副本工作），并使数据的来源变得神秘。

对于网页上的表格形式的数据，通常可以使用 Pandas 来解析 HTML 并生成一个 DataFrame。另外，对于结构化程度较低的数据源，像 Scrapy 这样的工具可以用来提取数据，“搜刮”，从那些难以手工收集的数据源中提取数据，并自动进行清理和验证步骤。

例子。像全国家庭成长调查这样的数据集，可以直接从他们的网站上下载文件，但使用条款禁止重新分发数据。因此，教员分享这些数据的最好方法是向学生提供代码，让他们输入笔记本单元，当执行时，学生第一次运行笔记本时，就会下载数据集。

许多维基百科页面包含 HTML 表格中的数据；其中大部分都可以用 Pandas 轻松导入笔记本中。

与体育有关的统计数据的来源通常被嵌入到大型的链接网页网络中。像 Scrapy 这样的工具可以浏览这些网络，以更适合于自动分析的形式收集数据。

受众。具有有限编程经验的学生可以处理标准格式的数据集，但刮取数据可能需要更多的编程经验。

形式（讲座/实验/...）。这种模式适合于更开放的项目工作，学生负责识别数据源、收集数据、清理和验证，但它也可以适应更多的支架式工作（见目标实践）。

特点。有助于增强学生的自主性和联系性。

陷阱。这种模式的一个危险是，学生可能会花太多时间去寻找那些不存在的数据。他们可能需要指导如何利用他们能得到的数据，即使它并不理想。

启用技术。Pandas, Scrapy, R, ROpenSci 软件包

4.3.13. 4.16 Now you try (with different data or process)

Description: Students start with a complete working example provided by an instructor and then they change the dataset or process to apply the notebook to an area of their own choosing. This method can allow more or less fluctuation depending on the skills of the students. For example we can allow students to select new datasets from a list that ensures the cells of the notebook will all still work or we can give them freedom to try new data structures or add new processes to break the notebooks and learn as they go through the process of fixing the broken cells.

Example: An instructor designs a lesson in exploratory data analysis to scrape the critics' reviews for a specific movie from a particular movie review website and then provide some simple visualizations. The students have a few options:

Green Circle - replace the movie name and pick any movie they want and then step through the new notebook and see the new results. Blue Square - adjust the notebook to scrape users' reviews rather than critics' reviews and then fix any data parsing problems. Black Diamond - add different visualizations tailored to explore the user reviews (as opposed to the initial visualizations that are tailored for the critics' reviews). There are various ways to test the properties of numerical methods. For example, students can use the method of manufactured solutions to test the order of accuracy for a differential equation solver. They can also measure cost as the resolution is increased and present the results in a way that would help an analyst decide which method to use

given external requirements (e.g., using accuracy versus cost tradeoff curves).

Related patterns: Top-down sequence

Learning goals: This pattern allows students to apply their knowledge

Audience(s): This pattern can be tailored for students with more or less experience even in the same course.

Format (lecture / lab / …): This pattern is best in a lab or an interactive tutorial

Pitfalls: A hazard of this pattern is that students may go completely off the rails and choose datasets or new processes that have not been tested and will not work in the timeframe allowed.

说明。学生从教师提供的一个完整的工作实例开始，然后他们改变数据集或过程，将笔记本应用到他们自己选择的领域。这种方法可以根据学生的技能，允许更多或更少的波动。例如，我们可以让学生从一个列表中选择新的数据集，以确保笔记本的单元格仍然有效，或者我们可以让他们自由地尝试新的数据结构或添加新的过程来打破笔记本，并在他们修复破损能单元格的过程中学习。

例子。一位教师设计了一个探索性数据分析的课程，从一个特定的电影评论网站上搜刮评论家对某部电影的评论，然后提供一些简单的可视化。学生们有几个选择。

绿色圆圈—替换电影名称，挑选任何他们想要的电影，然后在新的笔记本上踩点，看到新的结果。蓝方—调整笔记本以搜刮用户的评论而不是影评人的评论，然后修复任何数据解析问题。黑钻石—增加不同的可视化，为探索用户评论而定制（相对于最初的可视化是为评论家的评论定制的）。有各种方法来测试数字方法的特性。例如，学生们可以用制造解的方法来测试微分方程求解器的精度等级。他们还可以随着分辨率的提高而测量成本，并将结果呈现出来，以帮助分析师决定在外部要求下使用哪种方法（例如，使用精度与成本的权衡曲线）。

相关的模式。自上而下的顺序

学习目标。该模式使学生能够应用他们的知识

听众。这种模式可以为有更多或更少经验的学生量身定做，即使在同一课程中也是如此。

形式（讲座/实验/…）。这种模式最好是在实验室或互动辅导中进行。

陷阱。这种模式的一个危险是，学生可能会完全走火入魔，选择未经测试的数据集或新程序，在允许的时间范围内无法工作。

4.3.14. 4.17 Connect to external audiences

Description: This is in some sense the opposite of “the world is your dataset.” Here the goal is to take a workflow or computational exploration and share it with the world so others can see it, learn from it, reuse and remix it.

Examples: Your students are doing an observational astronomy lab where they take data from a telescope of a transiting exoplanet (a planet around a star other than our Sun) and they examine the lightcurve to learn about the planet. The students present the lab as a Jupyter notebook with a reproducible workflow that starts with reading in their data (images), walks through cleaning and reducing the images, and then performs photometry on the host star to produce a lightcurve. The end product is a plot showing the star’s brightness, dimming just slightly as the unseen planet comes between the star and earth. Elated with their result, the students want to share their data and workflow so anyone else can redo the analysis.

Learning goals: Reproducibility is an important part of the scientific process. Having completed the primary scientific analysis that was the goal of the lab (obtaining a lightcurve of a tran-

siting exoplanet), the students now can learn reproducible science practices by hosting their notebook on a webserver (e.g., Github) along with the data. An essential part of making the notebook reproducible will be ensuring that the notebook clearly lists the needed dependencies.

Audience(s): All students—everyone should learn about reproducibility.

Format: A self-contained notebook hosted on a webserver.

Features: Teaches students about reproducible science workflows.

Pitfalls: You need to be clear about the library requirements needed to run the notebook. Also, since the data files are likely separate from the notebook, it is possible for copies of the notebook to get shared without the data. Students may also be shy or fearful of showing their work publicly, so explaining the benefits may be needed to curtail their worries.

描述。这在某种意义上与“世界就是你的数据集”相反。这里的目标是将一个工作流程或计算探索与世界分享，以便其他人可以看到它，从中学习，重新使用和重新混合它。

举个例子。你的学生正在做一个观测性的天文学实验室，他们从望远镜中获取一个凌日系外行星（围绕太阳以外的恒星的行星）的数据，他们检查光曲线以了解这个行星。学生们以 Jupyter 笔记本的形式展示了该实验室的可重复工作流程，从读取他们的数据（图像）开始，到清理和减少图像，然后对主星进行光度测量，产生光曲线。最终的产品是一个显示恒星亮度的图表，当看不见的行星出现在恒星和地球之间时，它就会稍微变暗。学生们对他们的结果感到高兴，他们想分享他们的数据和工作流程，以便其他人可以重新进行分析。

学习目标。可重复性是科学过程的一个重要部分。在完成了作为实验室目标的主要科学分析（获得一颗凌日系外行星的光曲线）之后，学生们现在可以通过将他们的笔记本与数据一起托管在网络服务器（如 Github）上，学习可重复的科学实践。使笔记本可重复使用的一个重要部分是确保笔记本清楚地列出所需的依赖关系。

受众 (s)。所有学生—每个人都应该学习可重复性。

格式。一个独立的笔记本托管在一个网络服务器上。

特点。教会学生关于可重复的科学工作流程。

陷阱。你需要清楚地了解运行笔记本所需的库要求。另外，由于数据文件可能与笔记本分开，笔记本的副本有可能在没有数据的情况下被共享。学生们也可能对公开展示他们的工作感到害羞或恐惧，因此，可能需要解释其好处以减少他们的担忧。

4.3.15. 4.18 There can be only one

Description: This pattern involves creating a competition between individual students or teams of students. Clear goals and metrics need to be defined and then students submit notebooks that are scored and evaluated. Competitions can span months or be completed in a single class.

The Jupyter ecosystem support for reproducibility and data sharing make it a great environment for creating healthy competitions. Kaggle is a site that hosts many machine learning competitions using Jupyter as its underlying infrastructure and is a great place for advanced students to extend their knowledge with a chance of winning cash prizes and solving current world problems.

Examples: Identify a machine learning problem and a labeled dataset your students can use to train their model. Then select an evaluation metric and detail your problem statement and rules. Finally, launch your competition and allow your students to submit their notebooks and post their

results on a public leaderboard.

Learning goals: Creative problem solving is a key aspect of this pattern. In addition, if the competition is team-based then the students will learn how to work in groups and communicate effectively and share responsibilities.

Audience(s): Students can benefit from healthy competition and working in teams, however it is critical that a safe, fun, and engaging environment is created. Advanced students can be pointed toward kaggle or other public competitions that may be in their area of interest and give them a chance to test their skills in the real world.

Format: A competition can be defined with any metrics and rules and can be run in multiple ways. The students can help define the rules or a simple vote can decide the winners. For a more formal competition instructor can host free competitions for their class hosted by Kaggle. <https://www.kaggle.com/about/inclass/overview>

Features: Teaches students about creative problem solving and teamwork.

Pitfalls: Creating a fair competition is not trivial and considerations regarding data, metrics, rules, and scoring may be time consuming. Competition is risky business and effort should be made to ensure both winners and losers enjoy the experience.

说明。这种模式涉及在学生个人或学生团队之间建立一种竞争。需要定义明确的目标和指标，然后学生提交笔记本，并进行打分和评估。比赛可以跨越几个月，也可以在一个班里完成。

Jupyter 生态系统对可重复性和数据共享的支持使其成为创造健康竞赛的绝佳环境。**Kaggle** 是一个使用 **Jupyter** 作为底层基础设施的网站，举办了许多机器学习比赛，是高级学生扩展知识的好地方，有机会赢得现金奖励和解决当前世界的问题。

例子。确定一个机器学习问题和一个标记的数据集，你的学生可以用来训练他们的模型。然后选择一个评估指标，并详细说明你的问题陈述和规则。最后，启动你的比赛，让你的学生提交他们的笔记本并在公共排行榜上公布他们的结果。

学习目标。创造性地解决问题是这个模式的一个关键方面。此外，如果比赛是基于团队的，那么学生将学习如何在小组中工作，并进行有效的沟通和分担责任。

受众 (**s**)。学生可以从健康的竞争和团队工作中受益，然而，创造一个安全、有趣和有吸引力的环境是至关重要的。高年级学生可以被引向 **kaggle** 或其他公共比赛，这些比赛可能是他们感兴趣的领域，让他们有机会在现实世界中测试他们的技能。

格式。一个比赛可以用任何指标和规则来定义，并可以用多种方式进行。学生们可以帮助定义规则，或者通过简单的投票来决定获胜者。对于一个更正式的比赛，教师可以为他们的班级举办由 **Kaggle** 主办的免费比赛。<https://www.kaggle.com/about/inclass/overview>

特点。教会学生如何创造性地解决问题和团队合作。

陷阱。创建一个公平的比赛不是小事，关于数据、指标、规则和评分的考虑可能很耗时。竞争是有风险的，应努力确保赢家和输家都能享受这一经历。使用 www.DeepL.com/Translator 翻译(免费版)

4.3.16. 4.19 Hello, world!

Description: In some situations (such as the first day of class of a very introductory course) you may wish to do no more (and no less) than build confidence in the students' abilities to be able to write a first computer program. Traditionally, the first program written was a "hello, world"

program: a program that did nothing but display the text “hello, world” on the screen. However, these days students can have much more fun, and step fully into the creative world of computing with very little instruction.

Example: Draw a rectangle. Change the numbers, run it again, and see what happens.

Figure: a first sketch using Calysto Processing, a Java-based language designed for creating art.

Learning goals: Reduce stress, build confidence, connect onto their personal lives. Requires that they do learn the basics of Jupyter including: log in, open a new notebook, enter the provided code, and execute it. Often leads to a very animated, active learning classroom activity (“How can I change the colors?”, “How do I draw a circle?”, etc.)

Audience(s): Beginning students.

Format (lecture / lab / …): First day of class, in-class exercise. Build on what students already know from typing and reading (e.g., cut and paste, read top-to-bottom).

Features: Open ended, creative, fun.

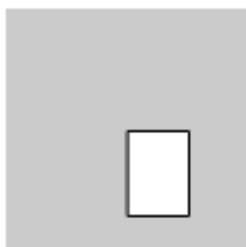
Pitfalls: Works best when used with a pre-installed Jupyter (see the relevant chapter). Rather than telling students that they can do it, just do it. As a first assignment, to cut down on the vast possibilities, we suggest limiting the palette of options. For example, restrict their drawings to use only a single shape, such as rectangle or triangle. We suggest having the students draw something in their life that is important or meaningful to them. We suggest discussing the coordinate grid for the first assignment and sketching an idea on paper first.

说明。在某些情况下（如非常入门的课程的第一天），你可能希望做的事情不多（也不少），只是在学生的能力上建立信心，使他们能够编写第一个计算机程序。传统上，编写的第一个程序是“hello, world”程序：一个除了在屏幕上显示文字“hello, world”之外什么都不做的程序。然而，现在的学生可以有更多的乐趣，只需很少的指导就可以完全步入创造性的计算机世界。

例子：

```
In [1]: rect(50, 50, 25, 35);
```

Sketch #1:



Sketch #1 state: Done.

图 4.1: 画一个长方形。改变数字，再次运行，看看会发生什么。

图：使用 Calysto Processing 的第一幅草图，这是一种基于 Java 的语言，旨在创造艺术。

学习目标。减轻压力，建立自信，连接到他们的个人生活。要求他们确实学习 Jupyter 的基础

知识，包括：登录，打开一个新的笔记本，输入提供的代码，并执行它。通常会导致一个非常生动、活跃的学习课堂活动（“我怎样才能改变颜色？”、“我怎样才能画一个圆？”等等）。

受众 (**s**)。初学的学生。

形式（讲座/实验/...）。上课第一天，课内练习。建立在学生已经知道的打字和阅读的基础上（例如，剪切和粘贴，从上到下阅读）。

特点。开放性、创造性、趣味性。

陷阱。与预装的 Jupyter 一起使用时效果最好（见相关章节）。与其告诉学生他们可以做，不如直接做。作为第一项作业，为了减少大量的可能性，我们建议限制选项的调色板。例如，限制他们的画只使用单一的形状，如矩形或三角形。我们建议让学生画一些他们生活中重要的或有意义的东西。我们建议在第一次作业中讨论坐标网格，并先在纸上画出一个想法。

4.3.17. 4.20 Test driven development

The instructor provides tests written in a unit testing framework like unittest or doctest; students write code to make the tests pass.

Example: TODO: Necessary?

Learning goals: Helps students learn a good software development process.

Audience(s): This pattern requires students to have some programming experience.

Format (lecture / lab / …): This pattern can be used for in-class activities or homework.

Features: Helps students focus on the task at hand and know when they are done (at least to the degree that the tests are complete).

Pitfalls: Some Python unit testing frameworks are not designed to work with notebooks, and can be awkward to use. On the other hand, nbgrader [TODO: add cross reference to nbgrader] supports automated testing of the code students write in notebooks; in that environment, the tests are not visible to students, which may or may not be a bug.

This pattern requires the overhead of teaching students about the unit testing framework. Students working to make tests pass can lose their view of the big picture, and feel like they have been robbed of autonomy. This type of exercise is best used sparingly.

描述：教员提供用单元测试框架（如 unittest 或 doctest）编写的测试；学生编写代码以使测试通过。

例子： TODO: 有必要吗？

学习目标：帮助学生学习一个好的软件开发过程。

受众 (**s**)：这种模式要求学生有一定的编程经验。

形式（讲座/实验/...）：该模式可用于课内活动或家庭作业。

特点：帮助学生专注于手头的任务，并知道他们何时完成（至少在测试完成的程度上）。

陷阱：一些 Python 单元测试框架不是为笔记本设计的，使用起来可能很麻烦。另一方面，nbgrader [TODO: 添加对 nbgrader 的交叉引用] 支持对学生写在笔记本上的代码进行自动测试；在这种环境下，测试对学生来说是不可见的，这可能是也可能不是一个错误。

这种模式需要教给学生关于单元测试框架的开销。学生们为了使测试通过而工作，可能会失去对大局的看法，感觉他们被剥夺了自主权。这种类型的练习最好少用。

4.3.18. 4.21 Code reviews

代码审查涉及学生或教师对别人的代码提供反馈。这种模式涉及到同行的工作，也是向学生提供反馈的一种手段，除了他们代码的正确性之外，还有代码的可读性和风格。

例子：向学生提出一个他们必须写出解决方案的问题，例如计算一个数字的平方根而不使用内置函数，但让他们为他们的函数写一个测试，使用一个内置函数来计算答案。在他们完成后，让学生们结成对子，对彼此的代码进行互评，不仅对他们解决问题的方式进行评论，如列出他们方法的利弊，而且对代码的可读性进行评论。

学习目标：学会阅读和理解别人的代码。学会写可读的代码。

受众 0：任何参与编码的学生群体。

格式：一旦在笔记本（或简单的脚本）中制定了一个合适的问题，那么课堂上的审查，就像上面的例子一样，可以用于同行的审查。另外，学生可以将他们的笔记本/脚本上传到 GitHub 等平台上，然后使用那里的工具进行代码审查。必须提供足够的支架，使学生了解这个过程，如何提出建设性的意见，以及为什么这个过程很重要。如果教师想审查并提供反馈，可以收集笔记本/脚本，并对其进行评论，同时向学生解释如何对其进行评分（如果有的话）。

特点：这种模式不仅导致了对写代码的人的反馈，也导致了对读者的反馈。代码审查也是工业界使用的软件开发过程中一个关键部分，为学生提供了一个对该过程的看法。这也可以通过适当的代码样式、注释和文档来确保学生的代码是可读的。

陷阱：学生们需要被正确地告知代码审查将如何影响他们的成绩，特别是如果使用了同行审查。GitHub 上的笔记本并不像脚本那样容易审查。

4.3.19. 4.22 Bug hunt

说明：教员提供一个笔记本，其中包含故意的错误的代码。学生们被要求找到并修复这些错误。可以提供自动测试来帮助学生了解是否有一些 bug 没有被修复。

例子：TODO

学习目标：这种模式可以帮助学生发展编程技能，尤其是调试（当然）；它还可以让学生练习阅读别人的代码，这可以成为展示良好做法的机会，或警告不良做法。它也可以用来教学生如何使用调试工具。

受众 (s)：本模式要求学生有一定的编程经验。

形式 (讲座/实验/...)：该模式可用于课内活动或家庭作业。

特点：可以吸引人，很有趣；培养重要的元技能。

陷阱：这些错误需要根据学生的能力进行调整：如果太容易，就没有吸引力；如果太难，就可能令人沮丧。

4.3.20. 4.23 Adversarial programming

描述：这种模式涉及参与者写出一个问题的解决方案，以及试图使书面解决方案失败的测试。这种模式可以通过多种方式进行，包括让学生完成任务并结对交换解决方案/测试，或者由教师编写解决方案，然后由学生编写测试。

例子：学生的任务是写一个函数，通过一些适当的输入找到指定的多项式的根。他们还被要求写一组测试，说明他们的函数通过和失败。当学生完成这些任务后，他们会交换笔记本，并将他们写的测试结果用于他们同伴的函数。最后，他们将讨论他们方法上的任何差异，以及他们是否能想出不使对方的测试失败的方法，或者所提供的测试是无效的。

学习目标: 学习编写单元测试。批判性地思考对手可能如何破解他们的解决方案。

5

Jupyter Notebook ecosystem

5.1. 5.1 Language support: kernels

Jupyter 系统支持超过 100 种编程语言（在 Jupyter 生态系统中称为“内核”），包括 Python、Java、R、Julia、Matlab、Octave、Scheme、Processing、Scala 等等。开箱后，Jupyter 只运行 IPython 内核，但也可以安装其他内核。语言支持继续由开源社区添加，最新列表的最佳来源是由项目维护的 wiki 页面：<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>。这些项目是由开源社区开发和维护的，存在不同程度的支持。一些内核可能由大量活跃的（甚至是付费的）开发者支持，而另一些则可能是一个人的宠物项目。当尝试一个新的内核时，我们建议探索一个内核的用户社区和文档，看看它是否对你（和你的学生）的使用有适当的支持水平。

Jupyter 内核的灵活性使导师们能够为特定的环境挑选合适的语言。例如，导师可以用 Python 教编程，而改用 R 教统计，然后可能用 Scala 教大数据处理。无论选择哪种语言，Jupyter 的界面都是一样的。因此，在课程内或课程间使用多种语言时，可以减轻一些认知负担（例如，在学生的数字人文和生物课程之间，用户界面保持不变）。不过，学生们通常会欣赏在一门课程中持续使用同一种语言。

5.2. 5.2 Using Jupyter notebooks

当在教室的数据投影仪或大屏幕显示器上使用 Jupyter 笔记本时，我们建议给学生具体说明笔记本的用户界面的含义。它并不完全是直观的。

笔记本的第一个和最突出的组成部分是单元格。事实上，笔记本的全部内容只由单元格组成。这些单元格可以采取两种形式之一：文本或代码。我们将在下一节中描述笔记本的编写；然而，在这里我们要确定代码单元的一些微妙而重要的组成部分。

代码单元由三个区域组成：输入区、显示区和输出区。输入区是由单元格左边的 In []: 提示来识别的。In 提示的括号之间可以是三个项目之一：一个数字、一个星号或一个空白。数字表示这个单元格已经被执行，数字的值表示执行的顺序。例如，通常情况下，在你打开笔记本后执行第一个单元格后，其提示将是 In [1]:。

专业提示

在使用笔记本教学时，你经常想用我的名字来指代一个单元。你可以用一个单元格的输入提示号来指代它。然而，请记住，如果你再次执行该单元格，这个数字会发生变化，或者如果学生也在执行他们自己的笔记本副本，他们的数字可能不同。一个更好的参考单元格的方法可能是参考单元格上方的文本，因为在你执行单元格时，文本不会改变。关于引用代码行的问题，请参见下面的提示和技巧部分。在执行单元格之前，输入提示数字区将是空白的。因此，你可以一眼看出该单元格还没有被执行。还有一种情况是，如果输入提示符中确实有一个数字，那么该单元格在过去就已经运行过了。然而，该单元可能没有在这个会话中运行过，因此输出可能显示的是旧的结果。我们建议从菜单中运行。单元，所有输出，在演示开始时清除。这将所有单元格输入初始化为空白状态。

在单元格的执行过程中，输入提示将包含一个星号。如果看起来时间过了太多，而你仍然看到 `In []`：你的代码可能处于无限循环中，或者你与内核失去了通信。你可能不得不中断或重新启动内核。这将在下面讨论。

最后，将输入单元下面的显示和输出区域分开是很重要的。这两个区域之间的区别是微妙的、令人困惑的，但在某些情况下是非常重要的。显示区是为代码产生的任何项目保留的，以供查看。这包括简单的文本（即 `print("hello, world")`）或绘图中的数字。输出区是为单元格“返回”的项目保留的。这就是为什么在许多笔记本中，你可能会看到一个变量赋值，后面紧跟着该变量，像这样。

`x = 2434 + 33476 x` 在这个例子中，除非你把它打印到显示区，或者返回这个值，否则你实际上不会看到计算的值。在这里，我们把它作为单元格的最后一个值返回。

请记住，在屏幕或显示器上的笔记本的底部部分可能对房间后面的学生不可见。确保字体足够大，而且在演示学生无法接触到的代码时，不要走得太快。我们还建议你隐藏 Jupyter 工具条和标题，以便为实际的笔记本提供更多空间（在 Jupyter 视图菜单下选择切换标题和切换视图）。

5.3. 5.3 Authoring Jupyter notebooks

在开始为你的课程编写笔记本之前，我们建议你在互联网上寻找相关课程。有一个类似的课程，教员已经为其编写了笔记本，你可以使用或改编为你的课程。笔记本作者通常很乐意合作开发开源教育资源，或者让他们的资源被其他教师使用。下面的章节着重于 Python，因为它是目前拥有最大 Jupyter 功能支持的语言。

5.3.1 5.3.1 Accessing documentation in the notebook

高质量图书馆的一个最好的特点是它们的文档，学生和其他用户可能会经常查阅这些文档。在笔记本单元中，**TAB** 键可以自动完成（或提供完成提示），**SHIFT-TAB** 键可以调出全部文档。同样，在一个方法或函数后面使用问号，会在单元格运行后调出文档，如图 5.1 所示。

在方法或函数后使用问号，会在执行单元格后出现文档。在方法或函数后使用问号，会在执行单元格后显示文档。

在课堂上，在现场编码时或在解释代码如何工作时使用这一功能，有助于使学生对有效地使用库工作感到满意。

5.3.2 5.3.2 Widgets

小工具为学习者和指导者提供了与代码输出互动的机会，如图表和表格。小工具是“迷你”的图形用户界面（GUI），使笔记本用户能够访问滑动条、切换按钮和文本框。它们可以与代码一起

使用，允许改变思维方式，从以编程为主要目标转变为以探索一个模型或计算为主要目标。另外，代码可以被隐藏起来，小工具可以用来创建一个笔记本“应用程序”，将输入参数与模拟和绘图联系起来。

目前，只有一小部分内核具有 **widget** 功能。小组件的参考实现是 **Jupyter-Python** 小组件 (<https://ipywidgets.rtfd.io>)。它包括生成和显示滑块、进度条、文本框、复选框、切换按钮等的 **widget** 组件。许多流行的可视化工具，如 **Matplotlib**, **Plotly**, **leaflet.js**, **three.js**，都有 **Jupyter-Python** 部件的实现。文档中包含了所有 **widget** 及其变化的最新列表。交互方法允许你包装一个函数，它可能是一个简单的计算，也可能是一个复杂的模拟，产生一个图，并为该函数的每个输入提供小工具。图 5.2 显示了一个正弦曲线图的简单例子，它的频率由一个滑动条控制。另一个具有一些部件功能的内核是 **C++** (<https://github.com/QuantStack/xw-widgets>)。

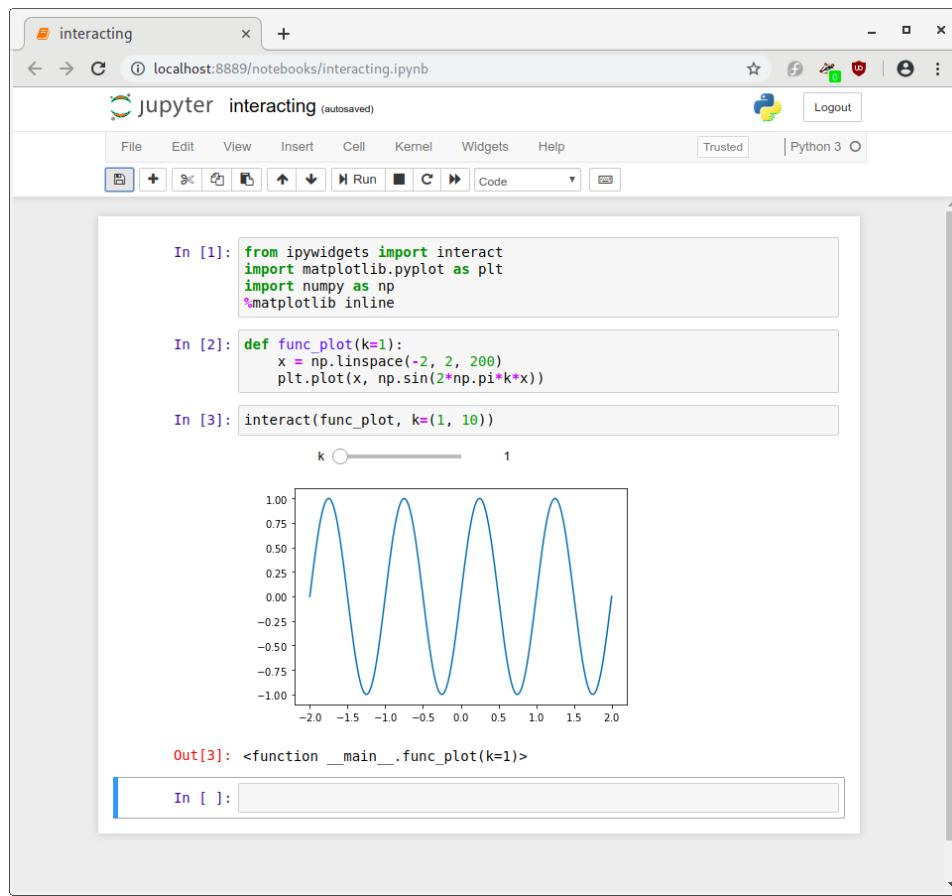


图 5.1：在这里，一个滑块允许用户在绘图时交互地改变我们函数中的变量 **k**。

除了 IPywidgets 库，**ipyleaflet** 库 (<https://ipyleaflet.rtfd.io>) 也可以在笔记本中显示一个交互式地图。

例子

从 **ipyleaflet** 导入地图地图 (`center=[34.6252978589571, -77.34580993652344], zoom=10`)

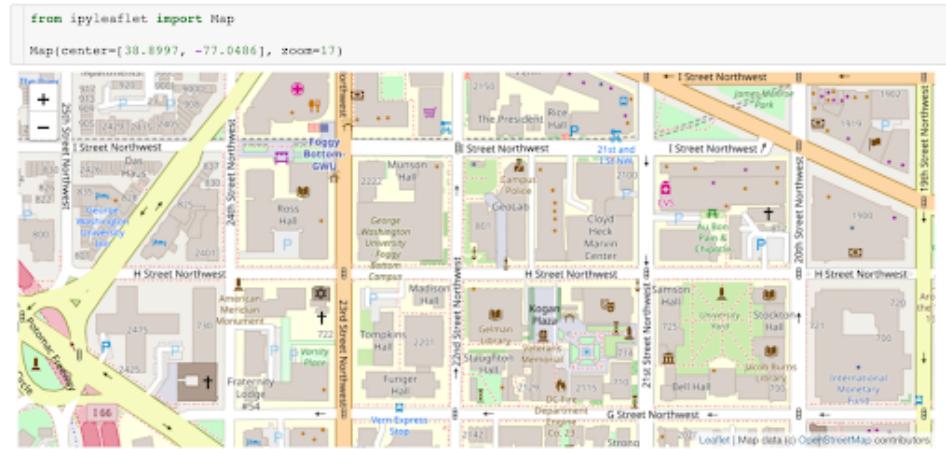


图 5.2: 使用 ipyleafletalt_text 的交互式地图部件。

对于有野心的读者来说，有一些资源可以让你写出你自己的自定义部件。**widget cookie cutter** 项目 (<https://ipywidgets.rtfd.io>) 是一个好的开始。

5.3.3. 5.3.3 Magics

魔法是元命令，只在 **Jupyter** 中发挥作用，并允许用户访问语言/内核的特定功能。例如，**IPython** 内核提供了许多魔法，当使用 **Python** 作为主要语言开发 **Jupyter** 笔记本时，这些魔法是有用的。这些都是有记录的，我们在这里只列举其中的几个。许多其他的魔法可用于不同的内核，但它们是特定于 **Jupyter** 的，所以可能无法在 **Jupyter** 之外的独立脚本中使用该语言。在某些情况下，你可能想少用魔法，以避免将这些元命令与你正在教授的语言中的实际命令混淆起来。对于单行命令，魔法总是以单一的% 开头，或者以%% 开头，将命令应用于整个单元。有些魔法可以使用单倍或双倍的

例子 **Matplotlib** 是可视化的一个常见选择。在 **Jupyter** 中，魔法`%matplotlib` 允许将结果显示在笔记本中。`%matplotlib inline` 产生嵌入笔记本的静态图像，而`%matplotlib notebook` 产生交互式图像（有缩放、平移等功能）。

`%run` 魔法允许运行外部脚本（和其他笔记本），捕获输出并显示在笔记本中，例如，`%run my_script.py` 或 `%run %%`

`%time` 魔法对后面的 **Python** 表达式的执行进行计时，例如，`%time sum(range(1000))`。

`%timeit` 魔法与`%time` 相似，但它多次运行表达式并报告平均执行时间。

`%reset` 魔法删除了所有用户定义的变量以及输入和输出。魔法通常有“标志”，遵循 **Unix** 命令模式。例如，`%reset -s` 是一个软复位，只删除用户定义的变量。这些命令对于避免乱序执行的问题很有用。

`%debug` 法宝是在代码因异常而停止后使用的（即“程序崩溃了”）。在崩溃后立即输入`%debug` 魔法，你将被放入导致问题的环境中。从那里你可以探索变量并找到问题的原因。

对单元格的全部内容进行操作的魔法的一个很好的例子是`%HTML` 魔法，它迫使单元格被解释为 **HTML** 并被适当地渲染。你也可以在运行 **IPython** 内核的同时使用魔法来调用其他语言。例如，你可以通过使用`%%R` 魔法从 **IPython** 笔记本中运行 **R** 代码。

专业提示

在 **IPython** 内核中，你也可以使用`%shell` 魔法。这通常被简写为!，可以从 **shell**/终端运行并

返回结果。在 **IPython** 中，你也可以将魔法与普通的 **Python** 代码混合使用。例如，`files = ! ls` 将在终端中使用 `ls (list files)` 命令，返回列表，并将 **Python** 变量 `files` 设置为该列表。

5.3.4. 5.3.4 Notebooks under version control

将笔记本置于版本控制之下是一种很好的方式，不仅可以跟踪内容的变化，还可以分享内容。在一门课程中，如果有多人为该课程的笔记本开发做出贡献，那么结合 **GitHub** 这样的平台使用版本控制，就可以跟踪作者，并为审查新的贡献或概述新作业、活动等的开发要求提供交流工具。使用版本控制的另一个好处是，有些服务会提供你所公开的笔记本的渲染视图。**GitHub** 显示笔记本的渲染版本，而不是笔记本的 **ASCII** 文本。**LaTeX** 渲染可能会出现一些隐患，因为平台对笔记本的渲染并不总是与它们在活动的 **Jupyter** 界面中出现的一样。

我们应该提到在版本控制下保持笔记本的一些注意事项。代码输出，包括图像，都存储在资源库中，除非你在提交修改前清除输出。这可能会使审查变化变得困难，因为即使内容没有实际变化，输出的变化也会被发现。如果对输出进行跟踪，跟踪的笔记本也会变得很大。即使清除了输出，由于笔记本的格式（笔记本是纯文本文件，文件格式表示为 **JSON**），审查更改也会变得很尴尬。社区正在积极开发工具，使 **Jupyter** 笔记本更容易使用版本控制；其中一个工具是 **nbdime**（见方框）。

nbdime nbdime.readthedocs.io/

nbdime 包括一套用于审查 **Jupyter** 笔记本的变化（"diffs"）和合并变化的工具。你可以使用终端比较笔记本的版本，在浏览器上查看丰富的变化，并以各种方式进行合并。因为 **nbdime** 了解笔记本文档的结构，它可以做出聪明的"差异和合并"决定。另一个改善你的版本控制经验的选择是将 **Jupyter** 笔记本导出为 **markdown** 文档，例如使用 **jupytext** 工具。然后你就可以用通常的方式来审查纯文本文件的差异。

5.3.5. 5.3.5 Testing notebooks

在分发笔记本之前，特别是如果您与多个课程材料的贡献者合作，在向学生分发笔记本或在现场演示中使用之前，测试笔记本可以帮助减轻意外的错误。至少，您可以通过重启内核并从上到下运行所有单元来测试笔记本是否从上到下干净地执行。这可以从菜单中完成（重启 + 运行所有）。

虽然它需要更多的设置，但可以使用持续集成服务自动运行测试，如 **TravisCI** (<https://travis-ci.org>)。这需要通过命令行执行整个笔记本，例如 `jupyter nbconvert --ExecutePreprocessor.enabled=True --to=html my_notebook.ipynb > my_notebook.html`。然后你可以在 **GitHub** 上发布这个文件：<https://github.com/opengeophysics/testipynb>

5.3.6. 5.3.6 Essential Python libraries

本节的目的是介绍 **Python** 生态系统中一些最广泛使用的软件包。如前所述，100 多个内核在 **Jupyter** 中实现了不同的编程语言。但 **Python** 是许多学科的共同选择，这是因为它拥有庞大的开源社区，开发并维护着一个由超过 15 万个软件包组成的生态系统。

核心 **Python** 库 (<https://docs.python.org>) 包含基本数据类型，如列表和字典，以及核心功能，如算术运算符和简单的文件分析器。大多数任务都可以通过核心 **Python** 实现。然而，通过更高级别的库，它们往往变得更容易。这尤其适用于使用 **Python** 的科学计算。在 **Python** 生态系统中的大量软件包中，**NumPy**、**Scipy**、**Matplotlib** 和 **Pandas** 是最常用的。熟悉这些库的一个好资源是 **Scipy** 讲义 <https://scipy-lectures.org/>。

Numpy (<http://www.numpy.org/>) 是一个用 Python 进行数值和科学计算的基本库。它包含了数字数组的数据结构, 线性代数的工具, 随机数的能力, 以及更多。SciPy (<https://docs.scipy.org/>) 为科学计算提供了一套多样的函数, 如优化、插值、统计和信号处理。它还包括许多学科的基本常数, 如光速, 以及稀疏矩阵的数据结构。Matplotlib (<https://matplotlib.org/>) 是 Python 的核心绘图库, 可以通过%matplotlib notebook 或%matplotlib inline cell magics 在笔记本中内联使用。Pandas (<https://pandas.pydata.org/>) 提供了数据分析的资源和灵活的数据结构, 用于标记表格数据。

5.3.7. 5.3.7 Advanced topic: extensions

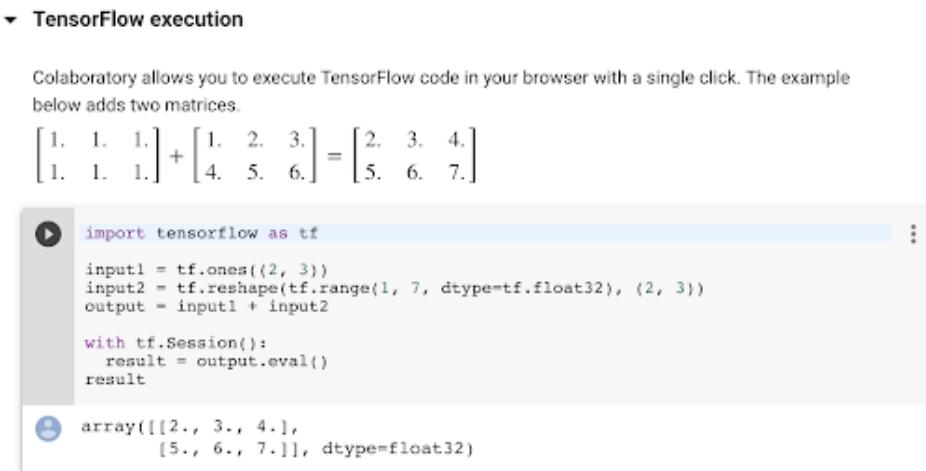
Example of incorporating Jigsaw, a block-based extension, in a Jupyter Notebook. The extension allows the user to assemble code blocks that can then be translated into Python or Java, and executed.

有许多社区贡献的扩展可以为 Jupyter 笔记本增加功能。这些扩展包括显示笔记本的自动目录, 或美化代码, 或隐藏/显示解决方案单元。以下是如何安装和启用扩展功能的链接: <https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/install.html>

这里有一个捆绑在一起的扩展列表: <https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextension>

创建自定义扩展是一种扩展或定制 Jupyter 的方式, 以增加当前扩展或开箱即用的功能。这些扩展可能是针对特定的内核的。以下是关于如何创建和安装自定义扩展的说明: https://jupyter-notebook.readthedocs.io/en/stable/extending/frontend_extensions.html

图十显示了 Google Collaboratory, 许多与 Jupyter 笔记本交互的工具之一, 如何利用 Jupyter 扩展的力量进行自定义交互和展示。



The screenshot shows a Jupyter notebook cell titled "TensorFlow execution". The cell contains the following code:

```
import tensorflow as tf
input1 = tf.ones((2, 3))
input2 = tf.reshape(tf.range(1, 7, dtype=tf.float32), (2, 3))
output = input1 + input2
with tf.Session():
    result = output.eval()
result
```

The output of the cell is:

```
array([[2., 3., 4.],
       [5., 6., 7.]], dtype=float32)
```

图 5.3: Google Collaboratory 使用 Jupyter 扩展来为他们的用户定制 Jupyter。代码单元左边的运行/播放图标是用扩展创建的。这在标准的 Jupyter 软件中是不存在的。TensorFlow 是一个用于在 Python 中创建机器学习实验的库。Google Collaboratory 使用 Jupyter 扩展来为他们的用户定制 Jupyter。代码单元左边的运行/播放图标是使用扩展创建的。这在标准的 Jupyter 软件中是不存在的。TensorFlow 是一个用于在 Python 中创建机器学习实验的库。

Jupyter 的扩展集正在不断发展。教育家们正在探索在教学法中使用笔记本的新的和有趣的方法。虽然目前的扩展列表太长, 无法一一列举, 但你可以通过这个实时的 Binder 笔记本 (Binder 在下一章有详细描述), 互动地体验一些最有用的扩展。这个实时笔记本演示了以下内容。

在代码单元中打开行号 (使之更容易参考某一行代码) 代码折叠扩展 (隐藏代码块以帮助集中注意力) 锁定和冻结单元格的扩展 (防止对单元格的改变) 为错误信息提供更好的用户接口的扩展

一个“乌龟”扩展（在笔记本中的画布上作画）基于块的编程扩展基于块的编程扩展（称为 **Jigsaw**）允许用户使用拖放的代码块进行编程，这些代码块可以与 **Jupyter** 笔记本中的其他单元整合（见图）。基于块的语言的优点（和缺点）是计算机教育研究中活跃的研究课题（例如，见 **Mark Guzdial** 的优秀的计算机教育研究博客，特别是那些关于基于块的语言的帖子）。

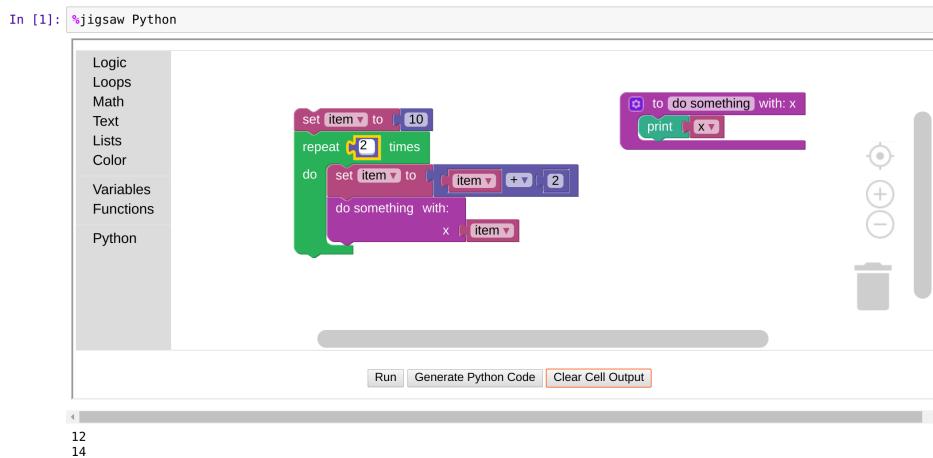


图 5.4：在 **Jupyter** 笔记本中加入 **Jigsaw**（一种基于块的扩展）的例子。该扩展允许用户组装代码块，然后翻译成 **Python** 或 **Java**，并执行。

5.4. 5.4 Tips and tricks

5.4.1. 5.4.1 Reminders

如果你在传统课堂上使用一个笔记本作为独立的练习（即这是你课堂上唯一的计算部分），那么在该笔记本的顶部有几个单元格，回顾如何浏览笔记本和如何插入单元格等，是很有帮助的。

5.4.2. 5.4.2 Feedback

在互动环节中，我们如何获得学生的反馈，看学生是否完成了练习？

一个低技术含量的解决方案是给学生不同颜色的便签，一个表示“完成”，一个表示“需要帮助”，他们可以把这些便签贴在电脑的背面。然后，教师可以迅速抬起头来，对全班学生的状态进行调查，并决定如何进行。

在屏幕上投射 **Slack** 或类似的聊天组，让学生复制粘贴解决方案（只要它们是简短的功能），是让班上每个人看到彼此的解决方案的好方法。投射多个学生的解决方案的一个积极方面是，它可以显示解决一个问题的各种方法。这样就有机会讨论解决方案的可读性和效率。缺点是，在一个大班里，大量的帖子会让人不知所措。相反，投票可以用来汇总学生的答案，并向教师提供某种形式的反馈。**Nbgrader** 或 **travis-CI** 也可以作为选项，要求学生提交完成的代码，并对其进行自动评估。然而，这些都需要更多的设置，并可能需要一些时间来完成。

5.4.3. 5.4.3 Explaining each cell

考虑将代码块的注释移到代码单元的正上方或下方的标记单元中。在 **markdown** 单元中的注释通常读起来更好，并且在讨论或描述代码时给你更大的灵活性。然而，在一个代码块中的简短注释仍然是有用的。

5.4.4. 5.4.4 How to structure code cells

你应该在一个单元格中放多少代码？随着经验的积累，你会形成自己的写法，即 **noteook**。通常情况下，你会希望保持较少的行数，这样就很容易理解，而且你可以在单元格的上方设置有用的注释。然而，我们建议把“作为一个有意义的单元”的代码放到一个单元中。例如，如果你有一些相互高度依赖的代码行，那么你可能想把它们放在一起。举个例子，考虑两行代码：一行打开一个文件，另一行从文件中读取数据。把这些代码放在同一个单元格中可能是个好主意，这样它们总是一起被执行。否则，如果学生在第二次独立执行单元格时可能会遇到错误（例如，没有更多的数据）。

具体来说，搞乱单元格之间的依赖关系是新用户使用笔记本的大部分困惑来自于此。例如，如果你改变了一个变量的名称（没有重新启动笔记本），那么下面的代码单元可能继续使用旧的变量名称（和值）。后来，当再次运行笔记本时，笔记本可能会以意想不到的方式失败，因为旧的变量已不存在。这有时被称为“隐藏状态问题”。这是一个开放的研究问题，研究人员正在探索各种可能的解决方案。例如，尝试在互联网上搜索“**jupyter** 依赖图”或“**jupyter** 数据流笔记本”。

专业提示

你可以使用按键 **CONTROL + SHIFT + -** 在光标处轻松地将一个单元格分成两部分。你也可以用 **SHIFT + m** 合并多个单元格。这两种方法也可以从编辑下的菜单中获得。另一方面，如果你想为学生提供一个交互式添加单元格的地方，并检查该过程中特定点的状态，将代码行分开往往是一个有用的想法。以苏格拉底的方法提出探究性的问题是一种非常有用的技术，可以吸引读者并鼓励他们成为更多的读者。学生们并不自然地知道在笔记本中插入单元格和探索项目。你将需要明确地教授这一技能。事实上，教学生如何有效地将代码编织到他们自己的笔记本故事中是用笔记本教学的一个重要组成部分。使用 www.DeepL.com/Translator 翻译（免费版）

5.4.5. 5.4.5 Custom styling

新的笔记本创建者经常试图集中管理标题、方程式和其他文本项目的格式。例如，创作者可能不使用标准的 **markdown** 标题，而是使用 **HTML** 样式来过度设计标题。这可能会产生两个问题。

笔记本 **markdown** 的渲染可能会改变，而你的格式化的 **HTML** 标题可能不会随着时间的推移保持相同的外观。

使用 **markdown** 创建的标题可以被笔记本工具使用，比如自动创建目录。

我们的建议是抵制自定义样式的欲望，而只是使用默认的表示方法。如果你想做定制（例如你想给某些单元格着色），你可以使用 **CSS**。

5.4.6. 5.4.6 Length of notebooks

笔记本作者有时会把笔记本写得很长，有很多主题和章节。笔记本的章节和单元格目前不容易以复制/粘贴的方式重复使用，以混合笔记本内的内容。在这个功能可用之前，我们建议作者围绕简短的主题制作简短的、独立的笔记本。这使得其他笔记本作者可以混合和匹配笔记本来创建课程。

5.5. 5.5 Gotchas

5.5.1. Programming language ≠ Jupyter

完全用 **Jupyter** 来教课，会让学生觉得这就是所有计算探索的方式。特别是，学生可能会被混淆，以为编程需要笔记本，而不是理解笔记本只是与特定语言互动的一种方式。这一点应该定期予

以明确。强化这一点的一个好方法是展示如何将在笔记本中开发和调试的功能剪切粘贴到一个脚本中（如 Python 中以.py 结尾的文件），然后将其导入笔记本以重新获得该功能。另外，集成开发环境（IDE）Spyder 有一个插件 (<https://github.com/spyder-ide/spyder-notebook>)，允许笔记本与 Python 脚本和 Python 终端一起显示，这对展示这种二分法很有用。

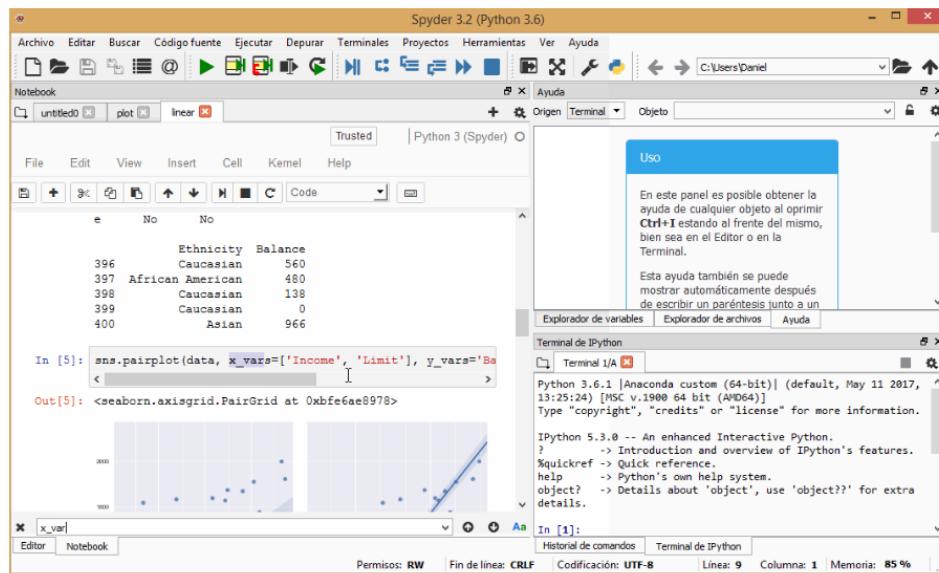


图 5.5: 来自: <http://go.gwu.edu/engcomp2lesson4>

Jupyter 笔记本显示在 Spyder 的一个窗口窗格中。

5.5.2. Restart, restart, restart…

通常，学生可能需要停止计算，这可以通过按下工具栏上的“中断”按钮来实现。但是，也应该让学生知道如何在笔记本中重新启动内核，以及这意味着什么。有几种情况下，学生可能需要这样做。有时学生写的代码会进入无限循环。在这种情况下，笔记本给出的视觉提示是微妙的，学生可能没有意识到这一点，不明白为什么笔记本没有反应。在现场编码的情况下，向学生展示这种情况是很有用的，告诉他们如何重新启动内核并继续进行。

第二个可能需要重启内核的例子是由于笔记本如何存储计算的状态。我们认为，由于笔记本是以线性方式布置的，所以状态将总是反映出如果笔记本从开始运行到那一点会发生什么。然而，在笔记本上不按顺序工作是很常见的，例如，如果学生对以前的某个例子提出问题。如果变量在随后的单元中被改变了，那么当你重新运行笔记本中较早的单元时，它的值可能不会反映你所期望的。重新启动内核有时是唯一的解决办法。

5.5.3. Notebook hygiene

许多问题可以通过开发对增量和非线性执行具有鲁棒性的笔记本来缓解。主要的原则是尽量减少执行单元的副作用，在不同的语言中表现得有些不同；我们在今天的建议与 Python 有关，可能需要为其他语言进行调整。笔记本通常应该能够连续执行，例如通过“重新启动内核并运行所有单元”。(一个例外是当一个笔记本为了现场编码或学生练习而故意不完整时，请看 nbgrader 或练习的 estnations 以获得更优雅的处理方法)。变量突变是最常见的方法，当以非线性方式执行单元时（例如，在回答学生的问题或比较和对比不同的方法时），笔记本可能出现故障。有时，这种突

变是偶然的，是通过虚拟变量，而这些虚拟变量并不打算在它们所使用的单元格范围之外具有意义。它们的范围可以通过把它们放在一个函数中来限制，即使这个函数只被调用一次。重新定义函数通常可以通过对所需功能进行参数化来避免，就像在设计一个库时通常会做的那样（尽管这对新手程序员来说可能是一种分散注意力的软件设计）。函数定义应该很少或没有对其包围范围内的变量的依赖性。当在课堂上为演示和形成性评估而修改单元时，复制单元或修改/执行是很有用的，这样在转到其他可能使用的单元时，符合要求的实现仍然存在。此外，你可以通过在一个单元中分组代码来减少这些问题，这些代码应该总是按顺序执行，因为单元中的代码总是有顺序的。

6

Chapter 6 Getting your class going with Jupyter

对于如何将 **Jupyter** 笔记本带给你的学生，你有几种选择。你可以要求学生在自己的电脑上安装 **Jupyter**，在实验室的电脑上安装 **Jupyter** 供学生使用，或者在学生通过互联网访问的远程服务器上运行 **Jupyter**。

6.1. 6.1 Local installation on students' or lab computers

Y

“本地安装”意味着每台计算机都在运行包括 **Jupyter** 笔记本的软件。通常情况下，这需要安装一个包括 **Jupyter**、**Python** 和可能的其他语言内核的发行版。

一个包括 **Jupyter** 的流行软件发行版是 **Anaconda**，它很容易在 **Windows**、**Mac** 和 **Linux** 上安装。因为它可以用用户级别的权限来安装所有的东西，所以不需要用户对计算机有管理员（或 **root**）权限。**Anaconda** 包括 1500 多个软件包，为学习者提供大部分（如果不是全部）所需软件。**Jupyter** 笔记本可以通过启动 **Jupyter** 或通过 **Spyder IDE** 打开。这些属性使它对个人使用和安装在机构控制的计算机上都有吸引力。

什么是 **Anaconda**？

你会看到许多教育家和课程作者推荐 **Anaconda** 发行版。**Anaconda** 是一个软件包管理器，一个环境管理器，一个 **Python** 发行版，一个超过 1500 个开源软件包的集合，包括 **Jupyter**。它可以免费下载，开放源代码，易于安装，可以在任何计算机系统（**Windows**、**Mac OS X** 或 **Linux**）上使用。它还包括 **conda** 打包工具，用于更新和安装 **Python** 和 **R** 生态系统的新包，并管理计算环境。根据该公司的网页，**Anaconda** 有超过 600 万的用户；见。什么是 **Anaconda**? **Software Carpentry** 项目提供了 **Anaconda** 的安装说明，并配有视频。另外两个可以运行 **Jupyter** 笔记本的易于安装的软件包是 **nteract** 和 **Hydrogen**。**nteract** 的安装方法是在其网站上下载二进制安装程序，然后双击安装文件。一旦安装了 **nteract**，学生本地系统上任何带有图形界面的 **Jupyter** 笔记本都可以被双击，它将在 **nteract** 中打开。氢气是一个非常受欢迎的开源 **Atom** 编辑器的插件；目前有 70 多万

人在使用它。氢可以让用户在 **Atom** 编辑器中编辑、显示和执行一个笔记本。

你可以要求学生在自己的电脑上安装 **Jupyter**, 或者让他们在实验室的电脑上使用它。这些也可以结合起来：给学生提供在他们自己身上安装的说明，但也要告诉他们，如果他们不能在他们的笔记本电脑上工作，在实验室里也可以使用。这样，你就不需要一个足够大的计算机实验室来容纳所有人，也不需要担心不是每个人都能自己搞定它。

6.1.1. 6.1.1 Jupyter on student-owned computers

在学生拥有的电脑上安装的好处包括。

一旦学生在他们的电脑上安装了软件，他们就可以一直使用它；他们可以在任何地方工作，他们可以将其用于实习、工作和其他非学校活动。

他们以后很容易安装其他软件包。

学生们学会了安装和设置 **Jupyter**, 以及一般的软件，这是他们可能需要的技能。

只要每个学生有足够的 **CPU** 能力和内存来支持预定的应用程序，那么班级的总计算能力就会随着学生人数的增加而扩展。

你可以采用 **Jupyter** 而不需要机构的支持或资源。

学生可以在自己喜欢的操作系统上学习使用 **Jupyter**, 例如 **Linux**、**Mac** 或 **Windows**, 这意味着他们已经熟悉了自己操作系统的基本习惯。

缺点包括。

这种方法只有在每个学生都拥有一台容量足够大的电脑时才有可能。

拥有性能较差的计算机的学生可能处于不公平的劣势。

虽然安装通常很容易，但仍然需要时间。对于一学期的课程来说，你在课堂开始时花费的时间可能是值得的，因为它贯穿了 **Jupyter** 的使用，但对于在关于其他东西的课程中使用 **Jupyter** 的单一模块或一次性作业来说，这就是一个障碍。

另外，花在调试深奥问题上的时间会随着学生人数的增加而增加：一个 **25** 人的班级肯定会有的一些人使用 **32** 位处理器、不兼容的库、过时的操作系统、过度热衷的病毒检查器等等，而一个有 **100** 名学生的班级会有 **4** 倍的学生。一个解决方法是让学生结伴工作：减少一半以上的学生不能工作的概率。

安装的库的版本不一致会给学生带来问题，并可能导致学生运行代码时出现不同的行为。

尽管 **Jupyter** 是跨平台的，在 **Windows**、**Mac** 或 **Linux** 上的表现最好是一样的，而且 **Anaconda** 等发行版在所有平台上的表现也非常相似，但安装和启动它的说明在每个操作系统上都略有不同，所以像“双击这里”或“输入这个命令”这样的细粒度说明对 **Linux**、**Mac** 和 **Windows** 用户需要不同的版本，这在介绍材料的教师只有一个平台可用的情况下可能是个挑战。值得制定详细的说明，让学生可以按照自己的节奏进行学习，而不是仅仅依靠课堂上的现场演示，这仅适用于一小部分学生。

6.1.2. 6.1.2 Jupyter on lab computers

使用实验室电脑而不是学生自己的电脑有统一性和提高公平性的好处。每个学生都会有完全相同的设置，而且指令对每个人的作用都是一样的。这减少了所需的个人技术支持，并保证所有学生都能获得足够的计算能力。

然而，这种部署也有一些弊端。

根据你对计算机实验室的控制程度，你可能需要机构的许可和支持。

学生可能被限制在只有当他们在校园里和计算机实验室开放的时候才能做作业，这对非住校学生或有全职工作的学生来说可能是一个不公平的缺点。

可能很难在需要时安装额外的软件包，而且学生可能不被允许安装他们在项目中需要的软件包。

即使在计算机实验室，也很难在不同的机器上保持一致性，并保持所有安装的功能。

6.2. 6.2 Jupyter on remote servers

即使 **Jupyter** 在本地运行，它也是作为一个网络应用程序运行的；也就是说，它在连接到服务器的浏览器中运行。在本地安装中，浏览器和服务器在同一台机器上运行。但也有可能远程运行服务器。

在这种情况下，学生不需要安装任何东西；他们只需要运行浏览器并加载一个 **URL**。

有几种方法可以在远程服务器上运行 **Jupyter**。

你可以在你或你的机构拥有的服务器上运行 **Jupyter**。你可以在云中运行的临时环境中运行 **Jupyter**。你可以在云端运行的持久性环境中运行 **Jupyter**。远程运行 **Jupyter** 有许多在实验室中运行的优势：你可以提供一个一致的环境，并保证所有学生都能获得足够的计算资源。而且，它减轻了实验室安装的一个缺点，因为学生可以从任何地方访问云资源，而不仅仅是在校园里。

在云端工作也意味着学生不必管理他们自己的笔记本电脑硬盘的备份。虽然学生仍然可能无意中覆盖、删除或破坏存储在云端的笔记本内容，但如果笔记本电脑损坏或丢失，他们不会失去整个工作。

对于简单的、一次性的 **Jupyter** 使用（例如，用于单一的作业或课堂活动），云选项是非常有吸引力的，因为它不需要在课堂上讨论安装额外的软件。

6.2.1. 6.2.1 Running in a temporary environment in the cloud

在云中运行 **Jupyter** 的最简单的选择是使用提供临时环境的云服务。其中一些服务是免费的，你可以不安装任何东西就使用它们。

这些环境很适合在不广泛使用 **Jupyter** 的课堂上进行简短的示例。学生可以打开一个笔记本，按一下按钮就可以开始运行。

然而，这些服务也有一些限制。

如果你的笔记本依赖于特定的包，或者包的特定版本，那么就很难满足这些要求。

这些服务在一个临时环境中运行笔记本，如果它被闲置就会消失。所以它们可能不适合管理学生的工作。

这些服务中的一些不能保证服务水平，可能无法满足你在课堂或研讨会上的需要。

Binder mybinder.org

Binder 是一个由 **Project Jupyter** 提供的开源服务。它允许一组驻留在公共资源库中的笔记本的所有者在 **Binder** 服务中预先建立一个图像，并获得一个可共享的链接，任何访问者都可以使用该链接来获得 **JupyterHub** 的工作实例，并预先加载资源库中的笔记本。这个会话是暂时的（当关闭标签或窗口时，用户所做的任何改变都会被删除），但它是完全互动的。**Binder** 是目前最受欢迎的服务之一，用于举办一次性的研讨会或教程。

6.2.2. 6.2.2 Running on servers you control

如果你可以使用具有足够计算能力的服务器或集群来支持你的班级—包括 CPU, 特别是内存—你可以使用 **JupyterHub** 提供 **Jupyter** 作为一种服务。

JupyterHub 是一个开源软件, 为小组中的每个用户提供一个基于云的 **Jupyter** 应用程序。每个用户在服务器上都有自己的账户和主目录。**Hub**, **JupyterHub** 的中央系统, 允许认证用户和启动单个 **Jupyter** 笔记本服务器。启动笔记本服务器的程序可以使用各种技术方案。更多细节, 请参见 <https://github.com/jupyterhub/jupyterhub/wiki/Spawners>

一旦 **Hub** 启动了用户的笔记本服务器, 在云端运行的 **Jupyter** 笔记本的行为就像 **Jupyter** 安装在个人电脑上时一样, 但 **JupyterHub** 将在远程云端电脑上运行笔记本并存储文件。如果学生希望使用本地安装的笔记本, 他们可以将存储在云端的笔记本下载到他们的本地计算机上。此外, 学生可以将笔记本 (和其他文件) 从他们的本地计算机上传到云端。

虽然任何人都可以在自己的 **Linux** 或 **Mac** 电脑上运行 **JupyterHub** 服务器, 但安装和配置 **JupyterHub** 需要跨越 **Linux/Unix** 操作系统、系统管理和网络的复杂知识。欲了解更多信息, 请参见。

<https://github.com/jupyterhub/jupyterhub> (基本的 **JupyterHub** 项目, 可以安装在裸机服务器、虚拟专用服务器 (**VPS**) 或商业云集群上)

<https://github.com/jupyterhub/the-littlest-jupyterhub> (**JupyterHub** 在远程服务器或 **VPS** 上的简化安装)

<https://github.com/jupyterhub/zero-to-jupyterhub-k8s> (在 **Kubernetes** 云系统上安装 **JupyterHub** 的分步指南)

提供 **JupyterHub** 服务有几个好处。首先, 学生可以立即启动并运行—他们不需要花时间安装软件。他们导航到一个网址, 登录到 **JupyterHub**, 然后开始使用 **Jupyter**。这种快速登录并开始计算的能力是让学生参与课程的有力方式, 建立了信心, 并避免了有时在学生的电脑上安装软件的紧张经历。

然而, 在你自己的服务器上运行 **JupyterHub** 是有缺点的。

开始并不容易; 大多数导师需要 (或至少受益于) 机构的支持, 而这些支持可能并不存在。

它可能难以扩展: 如果学生人数增加, 你可能需要更多的计算能力。而且, 学生产生的负载可能是不均匀的; 例如, 如果每个人都同时运行一个计算密集型的例子, 你的服务器可能无法处理它。

这个选项可能很昂贵, 除非你已经有足够功率的服务器。

6.2.3. 6.2.3 Running Jupyter in the cloud

如果你或你的机构不拥有支持你的班级的计算硬件, 你可以在 **AWS** 和 **Microsoft Azure** 等云服务提供的虚拟服务器上运行 **JupyterHub**。在这些环境中, 你可以按照上一节所述安装 **JupyterHub**。

在云中使用 **Jupyter** 的商业产品也是存在的, 其中一些提供免费试用或“免费”定价模式。它们包括。

CoCalc(以前的 **SageMathCloud**)(<https://cocalc.com>)是一个在线的开源计算环境, 对 **Jupyter** 笔记本有一流的支持, 由 **SageMath** 公司支持。它是少数允许多个用户同时编辑 **Jupyter** 笔记本的服务之一。它还允许笔记本用户循环查看笔记本的修订历史, 并默认提供一些流行的内核。该服务包括与项目合作者分享文件的能力。它是免费使用的, 更多的计算资源可以通过支付月费、年费或基于课程的订阅费来获得。教员可以为整个班级的资源付费, 或者要求学生付费并订阅一个学

期的资源。教员可以利用课程管理系统进行作业分配、收集、评分等。免费版本限制了对互联网的访问，以防止滥用，有效地阻止了对标准包管理器的使用。虽然教师可以通过向服务上传文件或要求公司安装软件来解决这一限制，但这对许多用户来说可能是繁琐的。付费版本解除了这一限制，允许使用标准的软件包管理器（例如，`pip`、`conda`、`R`、`Julia` 等）。

Gryd (<https://gryd.us>) 是另一个有免费级别的订阅服务。它包括课程管理功能，比如创建课程、邀请学生、部署自动评分的作业。

HubHero (<https://hubhero.net>) 为教师提供专业配置的 **JupyterHub** 服务器。对于不超过 30 名学生的课程，他们提供社区安装，让你在自己的硬件或你选择的云供应商上拥有自己的 **JupyterHub**。对于更大的课程或完全管理的部署，也可以使用托管解决方案。**Hub Hero** 的所有者是 Tim Head，他是 <https://mybinder.org> 的项目负责人和 **JupyterHub** 的贡献者。

Kaggle Kernels (<https://kaggle.com/kernels>) 是免费托管的 **Python** 和 **R** 笔记本，可以使用大量的计算资源（四核、16GB 内存、GPU、互联网连接，每次运行时间可达 6 小时）。所有的笔记本还可以无缝访问 **Kaggle Datasets** 上的数据集。教员可以使用 **Kaggle Datasets** 上的数千个数据集之一（非常适合实施“世界是你的数据集”教学模式），也可以上传自己的数据（每个数据集最多 10GB）。**Kaggle** 还可以帮助实现“只能有一种模式”，即课内竞赛，该功能允许教师轻松设置带有自动排行榜的竞赛。学生在 **Kaggle** 上做的所有工作都会被保存下来，在课程/研讨会结束后还可以使用。额外的软件依赖可以通过“`pip`”安装在笔记本中，或者通过向 **Kaggle Dockerfile** 发送 **Pull Request** 来包含在公共环境中。

这些服务最大的优点是，它们不需要安装，只需要教员进行最少的设置，其中一些还提供与学习管理系统整合的功能。但是，教员一般要创建学生账户，并设置学生环境。

这些服务具有高度的可扩展性，也就是说，它们可以处理大量的学生和不均衡的负载。然而，它们并不是万无一失的；它们可能需要一些照顾，以确保学生能够获得足够的资源。

这些服务的最大缺点是它们可能很昂贵。有些是按学生收费的，对计算和内存的使用有限制。有些是按实际使用量收费，这可能是不可预测的（可能需要教员对学生活动进行限制）。

其他缺点包括：

可能很难或不可能安装你需要的软件包，或软件包的特定版本。

有些服务对学生的活动有限制，例如，他们访问外部服务的能力可能有限。

许多这些服务都是相对较新的，它们有时会使导师和学生面临粗糙的边缘。

% 学生通常在课程结束后（或在有限的时间内）失去对其账户的访问。

在商业服务器上分享学生信息可能会有隐私问题。一些机构与一个或多个这样的供应商签订了协议，以解决隐私问题。

6.3. 6.3 Distribution and collection of materials

你可能想把课程材料分发给学生并从他们那里收集。有各种各样的选择。一些重要的事情需要考虑。

你想公开分享你的笔记本，还是需要隐私？

学生们创建或编辑的笔记本可以公开吗？还是需要隐私？

你打算如何评估收集的笔记本？

您需要与您的 LMS 集成吗？

您需要与文件共享系统整合吗？

你想用显示单元格输出的方式分发吗？

学生是否需要在他们自己（或实验室）的电脑上不容易得到的软件？

Jupyter 笔记本是纯文本的计算机文件，所以你可以把它们分发给学生，并使用任何处理文本文件的系统收集它们，包括 **GitHub**、**Google Drive** 和（作为最后手段）电子邮件附件。

6.3.1. 6.3.1 Learning management systems

许多教员使用学习管理系统（LMS）与学生沟通。这些工具提供私人文件共享和作业，连接到学生的机构计算机账户，它们可以用来分发和收集笔记本的文本文件。然而，大多数 LMS 工具还没有认识到笔记本，所以它们不能呈现笔记本，也不能让导师轻松地对笔记本进行评论或评分。

一些工具和工作流程正在积极开发中，以利用学习工具互操作性（LTI）标准将 **Jupyter** 生态系统与 LMS 生态系统连接起来。当你读到这篇文章时，你可能会发现，这些选项已经有所改善。

6.3.2. 6.3.2 Web hosting

笔记本可以在任何网站上公开托管，因此学生可以通过点击链接下载文件。大多数网络托管软件对笔记本不敏感，但你可以使用 **nbviewer** 来分享公共笔记本，以静态网页的形式呈现。

nbviewer nbviewer.jupyter.org

nbviewer 是由 Project Jupyter 提供的一项网络服务。你可以输入任何公开托管的笔记本的 URL，然后得到一个完全呈现笔记本内容的网页。一些浏览器扩展和附加组件可以让你通过点击按钮在 **nbviewer** 中打开一个笔记本。见。在 **nbviewer** 中打开。

6.3.3. 6.3.3 GitHub

分发和收集笔记本的流行工具之一是 **GitHub**，一个软件的托管和协作平台。**GitHub** 是基于 **git**，一个版本控制系统。版本控制下的文件通常被托管在 **GitHub**、**GitLab** 或 **Bitbucket** 等服务上，这些服务都能识别笔记本。例如，当你在 **GitHub** 上查看笔记本时，你会看到一个渲染的笔记本，其中包括格式化的文本、排版的数学、代码高亮，以及代码的输出，包括数字。

GitHub Pages（和其他类似的服务）也可以用来托管渲染的笔记本，持续集成服务可以根据笔记本构建网页，然后显示内容。见。 **Jupyter Book** 和使用 **ctr** 来做这件事。

学术机构的教育工作者可以使用 **GitHub Classroom**，它允许导师为一个班级设置作业。学生点击作业的链接，作业库的副本就会被创建，并以作业内容（可以是笔记本）初始化。每个学生的资源库都可以是私有的，只允许学生和教师访问。这可以成为向一个大班分发作业的有效方式。

git 的一个缺点是它很难使用。如果它对学生学习版本控制很有价值的话，也许值得在你的课堂上花时间来教 **git**。但如果这不是你的课的学习目标之一，你可以使用 **GitHub Desktop** 和 **git for Windows** 这样的图形界面，尽量减少学生对 **git** 的接触。

用于比较文件和合并修改的默认 **git** 工具在 **Jupyter** 笔记本上并不好用。然而，一些专门的工具可以帮助完成这些任务（见版本控制下的笔记本）。

6.3.4. 6.3.4 JupyterHub

如果你的学生使用 **JupyterHub**，你可以手动或通过脚本将笔记本和任何相关文件直接放到学生的目录中。如果你的 **JupyterHub** 实例上有 **nbgrader**，你可以用它来收集和分发笔记本（无论你是否选择使用 **nbgrader** 的评估功能）。这样，你就可以开发笔记本，并逐步让学生看到，让他们“取用”。然后，他们可以编辑笔记本或在他们的存储空间创建的目录中创建新的笔记本，然后将他

们的笔记本发布回给你，供你下载、查看或用 **nbgrader** 工具进行评估（关于这个工具的详情，见下一节）。

nbgrader

nbgrader 是一个用于创建、处理和自动评定基于 **Jupyter** 笔记本的作业的工具。它作为一个 **Jupyter** 扩展，由课程创建者安装在他们的计算机上。**nbgrader** 是 **Jupyter** 生态系统中的一个灵活的项目，可以分发和收集材料。顾名思义，它也可以给作业打分；它可以以分布式的方式使用，即每个学生都在自己的电脑上运行 **Jupyter**，也可以以集中式的方式使用，例如，如果学生在 **JupyterHub** 安装上都有一个账户。（更多细节见评估部分。）<https://nbgrader.readthedocs.io>

6.3.5. 6.3.5 Using an LMS and nbgrader together:

nbgrader 与学习管理系统的整合还很原始，但以下是一个能与当前工具配合的策略。

教员使用 **nbgrader** 创建一个作业笔记本，然后通过 **LMS** 将作业分发给学生。

学生完成作业并将解决方案上传到 **LMS**。

教员将完成的作业下载为一个压缩文件，并在 **Jupyter** 环境中提取学生的解决方案。

教员和评分员使用 **nbgrader** 对作业进行评分，并将成绩保存到 **CSV** 文件中。

然后，**CSV** 文件被上传到 **LMS**。

一些使这个工作流程更容易的工具包括 **nbgrader** 中 **ZipCollect** 功能的 **Extractor** 插件。

6.4. 6.4 Assessing student learning with Jupyter notebooks

许多教育工作者将课程评估活动开发为 **Jupyter** 笔记本。这包括考试、课内活动、家庭作业和项目。

处理基于笔记本提交的评估的简单方法是：让学生把它们打印出来，用电子邮件发送，作为标准的电子文档提交（例如，进入 **LMS**），或者把它们放到一个共享文件夹中。这时，教师可以用传统的方式给它们打分，比如在打印稿上写上评语或在 **PDF** 上添加注释。

专业提示

打印出来的笔记本有时会造成页面空间的浪费，特别是对于有许多图片或数字的笔记本。转换为 **PDF** 需要安装大型/复杂的 **LaTeX**。导出到 **HTML**，然后打印，往往能得到更好的结果。**nbgrader** 允许笔记本中的代码单元被标记为自动评分或手动评分。教员可以创建一个可以完全自动评分的作业，在笔记本创建后几乎不需要任何工作。这使得评分变得更加容易，并且在大班授课的情况下也能很好地扩展。然而，在 **nbgrader** 中创建这样一个自动评分的笔记本可能是相当耗时的。此外，在教学上，一个完全自动评分的笔记本可能有严重的缺点。例如，研究表明，当学生能够积极地将一个主题与他们自己的兴趣联系起来时，他们会学得更好 [需要引用]。鼓励这种做法的一个方法是在每次提交的作业中设置一个“反思”问题。这样的反思问题可以鼓励学生以个人方式对材料进行评论，但它不能被自动评分。另一个缺点是，简单地用单元测试来自动评分，不太可能评估许多你可能为作业设定的学习目标，例如，使用特定软件设计模式的能力。为了解决这个问题，你可以为作业的一部分创建手动评分的单元，并向学生提供书面反馈。

注意事项

在写这篇文章的时候，**nbgrader** 有一些限制，需要谨慎使用。例如，在一个多班的环境中使用它（比如在 **JupyterHub** 上），需要指导者协调作业的命名，使它们不发生冲突。**nbgrader** 是一个复杂的工具，可以设置成允许多个评分员、教学助理等。关于使用 **nbgrader** 的更多信息，见

<https://github.com/jupyter/nbgrader>。

一些第三方的基于笔记本的评估方案确实存在。例如，CoCalc、Vocareum 和 Gryd 提供了一个云笔记本平台，也可以对类似于或使用 nbgrader 的评估进行评分。

[TODO] cocalc.com...[Data8 Data9] //www.vocareum.com) TODO

6.5. 6.5 How do you create Jupyter notebooks for reuse and sharing?

当你为你的讲座、计算论文或家庭作业创建笔记本时，你可能希望考虑如何使它们有可能被自己和他人重复使用。

首先，你可能想让这些材料可以公开访问，并可以通过互联网找到。这意味着要避免把笔记本放在“围墙花园”里，如课程管理系统。也就是说，用户可能有机会接触到一些材料，但却无法看到其他材料。你必须决定你是否希望其他人有完全的访问权。例如，许多教师不希望学生能够看到可能有解决方案或解决方案提示的笔记本，因此限制他们的访问。

要与他人分享你的笔记本，你可以把它提交到 <https://www.engage-csedu.org/>。这个精心策划的开放教育计算资源集合由国家信息技术妇女中心（NCWIT）维护。

如果你决定让你的笔记本可以被其他人重新使用，请明确说明这些材料可以在哪个许可下使用。例如，你可以在你的笔记本底部包括一个知识共享的归属和共享声明。添加许可证可以让人们重新使用你的材料，而不需要明确征求许可。

GitHub 可能是托管和分享笔记本的最常见的服务，在那里可以查看（包括渲染）、下载或由他人分叉。（私人仓库也可以用来限制对同事、学生或其他组织的可见性）。然而，请确保注意保持笔记本版本控制的一些隐患（详见版本控制下的笔记本）。

共享的另一个潜在问题涉及到你可能想包含在笔记本中的外部文件。这与可由笔记本的代码直接创建的内容（比如说一幅图）形成对比。可能的内容包括数据、图像和视频，使用代码和嵌入标记的 markdown 或 HTML。其含义是，如果你分享你的笔记本，你必须把外部文件和笔记本一起包括进去。这可以通过多种方式实现，包括使用版本控制库、压缩文件或文件共享服务。共享笔记本的另一个外部依赖性问题涉及软件库。在这种情况下，你分享一个配置文件，用户可以用它来设置相同的环境。这些文件的例子包括 conda env.yml, pip requirements.txt, 或 dockerfile。

由于 Jupyter 笔记本将单元格的输出嵌入到 ipynb 文件本身（例如，图片、视频等），文件可能会变大。为了能够通过 Github、Gitlab 或 nbviewer 上的渲染器显示单元格的输出，在笔记本执行后保存它，然后上传到这些服务。如果你想减少文件大小，并在清除了代码单元输出的情况下向别人提供笔记本，请在笔记本的下拉界面中选择这个选项。然后，用户需要自己执行笔记本才能看到输出。

6.6. 6.6 Jupyter: a 21st Century genre of Open Educational Resources and practices

教育工作者创造了教学材料。随着互联网的出现，一个教育工作者社区开始制作开放的传统教学材料。与此同时，一个软件开发者社区开始创建开放源代码软件。每个社区都发展了他们自己的开发模式。特别是开源软件社区倾向于分布式和协作式工作的集市风格 1。Jupyter 笔记本可能

是这些社区第一次合并。**Jupyter** 笔记本的作者正在将他们所使用的内容创建模式应用于创建教授计算或通过计算进行教学的开放教育资源。

开放教育涵盖了一个庞大的社区，有自己的会议和期刊，有领导者和倡导的实践。最明显努力与开放教育资源（OER）有关：创造和采用开放许可的学习材料。1994 年，Wayne Hodgins 创造了“学习对象”这一术语，并传播了数字材料可以被设计和制作成可重复使用的想法。随后，人们努力开发元数据标准、内容交换等（解决如何找到对象以重新使用它们的问题）。1998 年，David Wiley 创造了“开放内容”一词，并传播了自由和开放源码软件（FOSS）的原则可以应用于万维网上的内容的想法（OpenContent, 1998）。Creative Commons 非营利组织成立于 2001 年，为共享内容提供现成的许可协议，并在开放教育资源的传播中发挥了重要的基础设施作用。知识共享协议现在是开放教育最广泛使用的许可框架。2001 年，麻省理工学院的 OpenCourseWare（OCW）也启动了。麻省理工学院承诺对其课程材料的非商业用途进行免费的公共访问。这是在机构层面上的一个独特的承诺，通过麻省理工学院的品牌得到加强。其他大学也加入了 OCW 运动。莱斯大学有 OpenStax 项目（现在的前身是 Connexion），CMU 有开放学习倡议（OLI），犹他州立大学有开放和可持续学习中心，等等。今天，开放教育联盟有来自世界各地的数百名成员。OER 中反复出现的话题是：降低学生购买教科书的成本，增加使用机会，以及处理版权和许可证问题。

在过去的几年里，使用 **Jupyter** 的教育工作者一直在以笔记本的形式创建和分享各种教育材料，通常是在知识共享署名许可（CC-BY）下。事实上，**Jupyter** 是一种新的开放式教育资源类型。但是，除了创造开放的内容，使用 **Jupyter** 的教育工作者往往积极参与 **Jupyter** 社区，并采用开源软件的文化。这是一种具有强烈道德承诺的文化，与访问自由、透明度和治理有关（Coleman, 2012）。他们创造的内容在一个开放的模式下，具有给予访问的价值（OER 的定义）。但开源文化也促进了合作文化。在这方面，使用 **Jupyter** 进行教学，为教育者参与开放开发并与他人合作制作课程、教程、课程甚至书籍提供了新的可能性。

专业提示

分享教育笔记本，获得反馈，并为你的工作获得荣誉的一种方式是在《开源教育杂志》上发表。这是一份经同行评议的期刊，针对开发使用代码教学的 OER 的教育工作者。除了收到为你的工作做广告的出版物外，同行评议的过程将使软件、代码和教育材料的质量更高。

7

研究使用的案例

7.1. Jupyter 笔记本支持大面积招生的扩展性

本章的贡献者：如果你将以下一些信息整合到你的案例中，你可能会增加新用户的采用。

证明你可以提高学生的能力。吸引材料和参与课堂理解材料并表现良好为自己的职业生涯做准备享受这种学习方式描述。它如何与学生的学习方式相适应它是如何与他们的教学方式相联系的需要的资源（支持、硬件等）。必要的物流（例如，需要多少时间？说实话：时间是一个考虑因素，也是人们不采用新做法的一个重要原因，但不是他们停止使用的原因）**Jupyter** 在促进学习方面的作用，教师的能力

7.1.1. 支持加州大学伯克利分校的大量招生课程

加州大学伯克利分校在 2015 年秋季为大约 100 名新生本科生开设了一门名为“数据科学基础”（也称为 Data-8）的试点课程。Data-8 是伯克利历史上增长最快的课程，完全基于 Jupyter，使得该课程在 2018 年可以扩展到 1400 名学生。这种规模是由 Jupyter 的共享计算环境实现的。特别是，Jupyter 允许“基于浏览器的计算，避免了学生安装软件、传输文件或更新库的需要”（见《未来的课程和它背后的技术》

<https://data.berkeley.edu/news/coursefuture>。

Data-8 由 JupyterHub 驱动，所有的课程材料都是公开发表的

<http://data8.org>。

7.1.2. 在加拿大各地开展的活动大规模采用 Jupyter

认识到数据科学、计算研究和教育资源的重要性，太平洋数学科学研究所（PIMS）与加拿大计算机公司和 Cybера 合作，推出了 JupyterHub 平台（项目名称为 Syzygy），以支持加拿大各地的研究人员和教育工作者。Syzygy (<http://syzygy.ca>) 提供了使用现有机构证书访问云托管的 Jupyter 资源，并鼓励发展计算和数据科学技能。目前，全国有 16 个机构（麦克马斯特大学、皇后大学、SFU、阿尔伯塔大学、UBC、卡尔加里大学、莱斯布里奇大学、新布朗斯维克大学、渥太华大学、Regina 大学、USask、UToronto、UVic、华盛顿大学（美国）、滑铁卢大学、Yorku）可以使用它，

这些机构有超过 11000 人在使用。

Syzygy 被广泛地用于教学，但也被用于研究活动。一个值得注意的例子是英属哥伦比亚大学的科学软件研讨会，研究生和博士后研究人员在那里聚会，与他们的同行分享和学习数据科学的技术。作为 **syzygy** 的一部分，也正在进行一些倡议，通过提供对更大和更多类型的资源（GPU、并行机器、不同的语言内核等）的无缝访问，加深其与研究的相关性。

Callysto (<https://callysto.ca/>) 是一个相关的项目，也是由 **PIMS** 和 **Cybera** 发起的，旨在将 **Jupyter** 带到加拿大的初中和高中(5-12 年级)。**Callysto** 专注于创建和策划开放内容(<https://github.com/callysto>)。这些内容构成了项目研讨会的基础，教师可以通过互动的方式处理这些材料，然后再把它们带回教室。这些内容链接到支持性的 **JupyterHub** 安装（与学区网络的认证系统集成），允许轻松访问这些材料和 **Jupyter** 环境来学习和创作。

- 伊恩-埃里森

7.1.3. 快速转换：将现有课程转移到 Python 和 Jupyter (在最后一分钟)。

多年来，我们的化学工程动力学课程一直使用微分方程和非线性同步方程的软件来模拟反应器和解决设计问题。教科书中推荐和描述的这个软件被安装在学院的计算机实验室中，但学生拥有的计算机的许可证很昂贵，而且只适用于 **Windows**。2015 年春季，我被告知我的班级现在有 52 名学生，但最大的计算机实验室只能容纳 40 人。随着学期的进行，我们接近需要数值模拟的章节，我使用 **Python** 和 **SciPy** 重写了例子，并创建了 **Jupyter** 笔记本，引导学生完成设置和解决问题的步骤。我在网上找到了 **Lorena Barba** 的开源 MOOC 材料，并将其改编为我的“入门”笔记本。我让学生在他们自己的电脑上安装 **Anaconda**，并在没有任何中央基础设施或学院 IT 人员支持的情况下，让每个人都开始运行。我发现 **Jupyter** 笔记本的格式，包括“讲义”式的评论，以及简短的、不令人生畏的代码片段，是非常有效的。几年后，我把这门课程传给了一位新的讲师，他拿着我的课程材料，自学了一些 **Python**，并继续使用 **Jupyter** 笔记本来传递内容和布置作业。

第一年，由于我在学期中很晚才引入 **Jupyter** 笔记本，所以在边缘上有点粗糙。不过，很明显，这种方法得到了学生的共鸣。我 2016 年课程的一位校友写道：“我认为你的课程非常成功，尤其是使用 **Jupyter** 笔记本作为课堂和作业工具。我仍然记得我们在课堂上讨论的具体问题（例如，异质催化的微流控反应器阵列），我觉得在整个课程中使用 **Python** 来解决问题，大大有利于我对基本概念的理解。我后来 [在制药业] 使用 **Python**，在那里我建立了生物信息学数据分析的工具，蛋白质工程实验的突变网络分析，以及从实验数据和分子热力学预测 **RNA** 结构”。

- 理查德-韦斯特使

7.2. CFD Python " 的故事：以自己的节奏引导学习者

“CFD Python ”是一个基于 **Jupyter** 笔记本的集合，我于 2009 年在波士顿大学的计算流体力学 (CFD) 课程中开始使用该实践模块。这个为期 5 周的模块开发了一些工作实例，这些实例相互关联，逐步引导学习者创建一个程序，在 12 个步骤中解决流体力学的纳维-斯托克斯方程。2013 年，我被邀请在阿根廷的拉美高性能计算学校教授一门小型课程。**Jupyter** 笔记本平台使我能够创建一个引导性的叙述，以支持具有不同背景经验和知识的学习者。在这次活动中，我们根据 CFD 课程模块编写了笔记本，在为期 2 天的迷你课程中作为教学支架使用。20 名学生以自学的方式完成了笔记本，而我则挨个询问和回答问题。大约四名学生在两天内完成了所有的课程，其中大部分人达到了第 8 步，少数人在课程结束时落后于第 4 或第 5 步。对于那些完成整个模块的人来说，

他们在 2 天内完成了我的普通学生在课堂上通常需要 5 周才能完成的任务。看到这一幕，我大开眼界：代码中的工作实例的力量，以及允许学习者按照自己的节奏学习的能力，使这些学习者有了显著的变化。

REF - Barba, Lorena A., and Forsyth, Gilbert F. (2018). CFD Python: 纳维尔-斯托克斯方程的 12 个步骤。开源教育杂志, 1 (9), 21,

<https://doi.org/10.21105/jose.00021>

基于开发“CFD Python”学习模块的经验，我们采用了这种基本的设计模式来创建使用可计算内容的课程。

1. 把它分解成小的步骤
2. 把小步骤分成大步骤
3. 增加叙述和联系
4. 链接到文档
5. 穿插简单的练习
6. 用挑战问题/任务来调剂
7. 在网上公开发表

- 罗蕾娜-A-巴尔巴

7.3. 用 music21 分析音乐

2013 年，我在阅读了 Luciano Ramalho 的教程后，开始对学习更多的 Python 感兴趣，因为他正在编写 **Fluent Python**。由于我倾向于寻找与我的外部兴趣（音乐、艺术和自然）相匹配的项目，我在寻找与音乐有关的 Python 项目时，遇到了 Myke Cuthbert 的 music21 项目。Music21 是一个开源的音乐理论和分析库，由麻省理工学院的 Michael Cuthbert 教授维护，提供了一套工具来快速简单地回答关于音乐的问题。用户只需几行代码就可以创建、分析和分享音乐。迈克对笔记本的使用迷住了我。与我以前做过的许多事情不同，笔记本使我很容易上手，并能写出能做实际工作的小代码片段！我越是使用笔记本，就越是觉得它是个好东西。我使用笔记本的次数越多，并把它们展示给我在 Fab Lab San Diego 教授的人看，我就越能看到笔记本的力量，它能吸引用户，使他们有能力去探索和学习。

音乐是一种通用的语言，吸引着各种出身、年龄、教育水平和兴趣的学习者。作为一个具有广泛吸引力的学科，音乐提供了吸引和愉悦学习者的机会。它是一门易懂的学科，对于来自计算机科学和工程以外的学科的学习者来说，门槛很低。

- 卡罗尔-维林

教育的好处

- 课程笔记本可以根据音乐中适合年龄的内容进行定制
- 多感官的
- 在 K12 阶段能够与标准接轨
- 带来多学科学习的可能性

- 写作

- 历史

- 数学
 - 科学
- 通过音频和盲文实现无障碍

杂项引言（也许挑几个?）

”我认为 **music21** 是由两部分组成的。第一部分是基础设施，即阅读、书写和操作乐谱的程序，而第二部分包括一个更高级别的分析工具包—从和弦和调上生成罗马数字，将和弦放入正常形式，检查平行五度，识别包含特定音高或和弦的音阶，等等。普林斯顿大学音乐教授布鲁斯-泰莫兹科 (**Bruce Tymoczko**)。

包容性

”它不是排他性的，而是包容性的，这就是爵士乐的全部精神”。- 赫比-汉考克

教育

”所以，你不能停留在一个地方，无论那个地方多么舒适。一切都是为了成长。”-梅维斯-斯台普斯

普遍性

”灵魂中的音乐可以被宇宙听到。”-老子

沟通

音乐是世界上最伟大的交流。即使人们不理解你所唱的语言，当他们听到好的音乐时，他们仍然知道它。

”在初学者的头脑中，有许多可能性。在专家的头脑中，可能性很少”。-铃木顺龙

7.4. 计算机科学中的互动性（高中和初中）

谁

Cal Poly SLO 的 EPIC 项目的高中和初中学生完成了一个两小时的计算机科学互动性研讨会。讲习班的参与者包括双重语言学习者（英语作为第二语言）和在讲习班之前接触电脑机会有限的学生。

为什么

为那些可能认为自己没有能力学习编码或使用计算的高危群体提供早期机会

说明除了数学和科学之外，还有许多技能是创建软件应用程序所需要的。

两个小时的研讨会，最大限度地“动手”探索，目的是建立对计算机科学的持续兴趣

内容

- 简短的讲座
 - 互动讨论-听
 - 实践-做/应用这一部分是自定进度的，以吸引不同的学习方式和原有知识。
 - 回顾 - 讨论
- 8 个左右的项目，列出成就
- 现代课程包括 **p5.js**、**jupyter**、**binder**、深度学习和机器学习与 **TensorFlow** 和 **Magenta**（艺术和音乐）。
- 目标是让学生了解他们可以用 **CS** 来解决现实世界的问题

教员的方法

- 从高质量的、有吸引力的内容开始
- 自带笔记本
- 使用部件来增加额外的互动性

7.5. 使用 Jupyter 的交互式地球物理学

GeoSci.xyz 项目 (<https://geosci.xyz>) 是围绕地球科学的学习资源和软件，努力发展一个科学家和教育工作者的社区。该项目包括多本开源教科书，每本教科书都有相关的 Jupyter 笔记本“应用程序”，作为探索地球物理学概念的互动模拟引擎。我们在不列颠哥伦比亚大学的应用地球物理学本科课程中使用了这些资源；该课程主要由地质学家和工程师（非地球物理学专业）学习。2017 年，我们为全世界 26 个不同国家的专业人士、研究生和研究人员提供了为期 2 天的短期课程 (<https://disc2017.geosci.xyz>)。在这两门课程中，我们的目标是为学习者提供各种地球物理方法（如磁力、重力、地震、电磁学）和支配物理学的概念的概述；我们不深入研究数学的细节，也不期望学生编程或写任何一行代码。Jupyter 笔记本在这些课程中的作用是作为可视化和探索物理学的工具。

在讲课过程中，笔记本作为一种演示媒介，形成了一种动态的演示风格，我们作为导师可以根据学生的输入选择模型参数。当学生在实验室和作业中使用这些相同的笔记本时，概念得到了强化。我们发现，当学生首先被要求批判性地思考他们期望看到的东西，然后将结果可视化时，笔记本的应用是最有效的。如果结果的图像符合他们的期望，那么他们就理解了这个概念，如果不符，这就是一个学习和进一步探索的机会。

- Lindsey Heagy

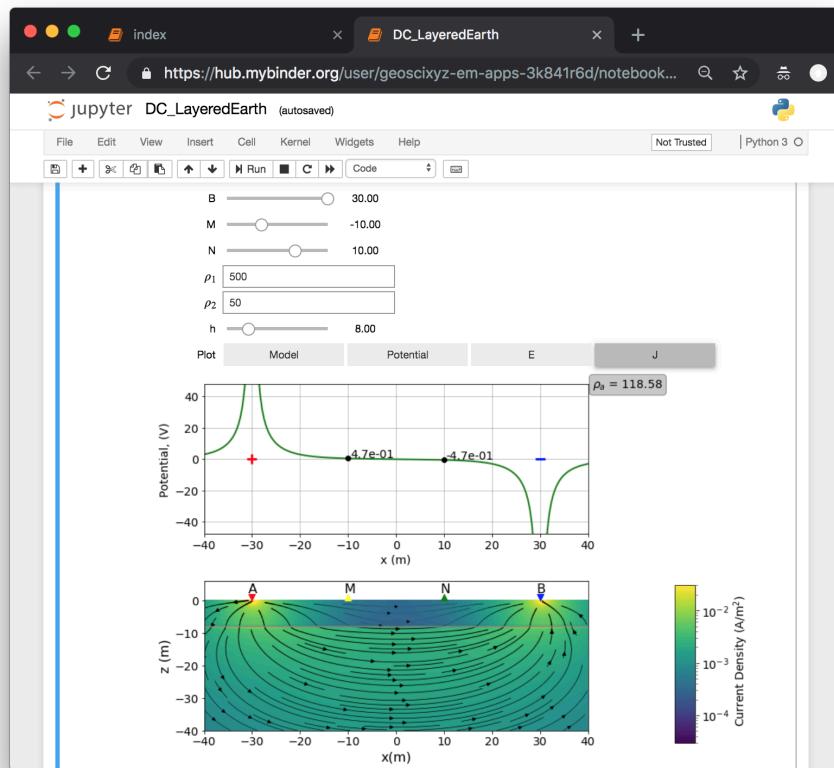


图 7.1: “笔记本”应用程序”用于探索两层地球上的直流电阻率实验 (<https://em.geosci.xyz/apps.html>)。

7.6. 7.6 Investigating hurricanes

谁

参观哥伦比亚大学工程与应用科学学院的初中和高中学生，进行实地考察。

为什么？

学生们经常来参观实验室，体验学校正在进行的一些研究。不幸的是，某些领域，如计算数学和飓风研究，并不适合这些类型的活动。

什么

一个计算机实验室被保留了一个小时，Jupyter 笔记本被用来指导学生完成一些基本的可视化和数据分析，鼓励学生改变显示的代码来回答诸如“桑迪飓风去了哪里？”和“1981 年期间发生了什么风暴？”等问题。这包括飓风轨迹的一些可视化，按风暴的强度着色，以及对每年平均风暴数量的分析。笔记本脚本可在

<https://github.com/applied-math/demos>

查询。

- Kyle T. Mandli

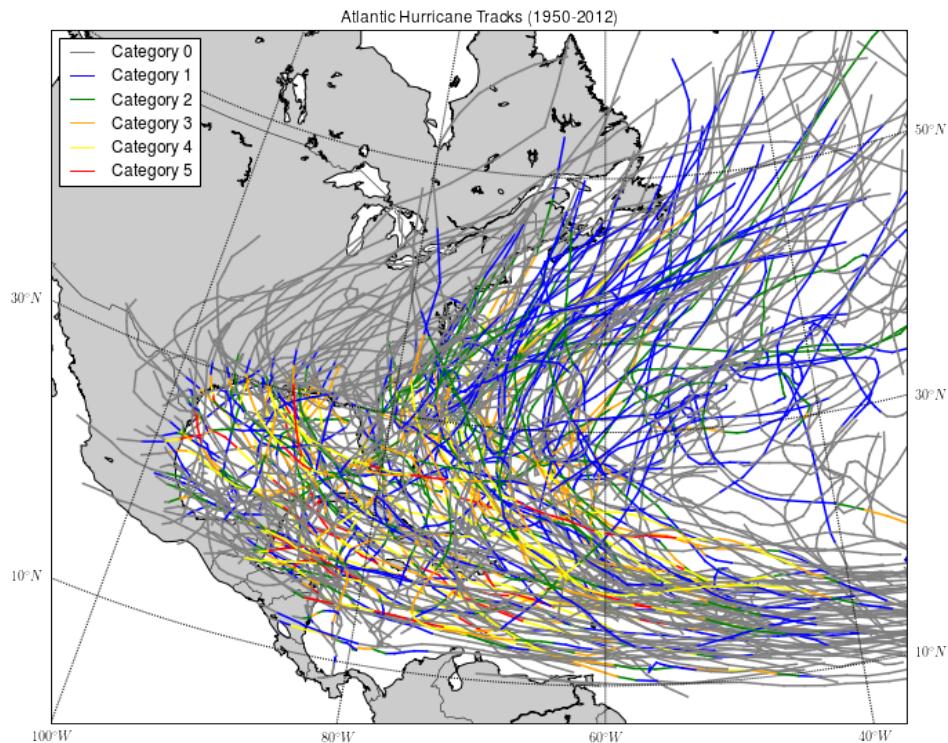


图 7.2: 笔记本上的可视化, <https://github.com/applied-math/demos>, 展示了 1950-2012 年大西洋飓风的轨迹, 并以颜色显示风暴的类别。

7.7. 7.7 Riemann Problems and Jupyter Solutions

我们首先设想将这个项目作为一个教学辅助工具, 为研究生课程中的守恒定律的数值方法进行互动说明, 以说明困难的概念。这些物理定律以一阶双曲偏微分方程的形式出现在波的传播应用中, 如流体力学、交通流、水波 (如海啸) 和电磁波等等。黎曼问题对应于一个具有片状常数初始条件的守恒定律, 这样的问题解决起来相对简单, 同时还能捕捉到守恒定律的特征动态。由于其离散性、简单性和对动力学的编码能力, 黎曼问题是现代守恒定律数值方法的关键因素。

该项目自然而然地演变成了 **Jupyter** 笔记本, 其想法是编纂一本互动的书。每一章都旨在解决一个特定应用的黎曼问题, 如声学、浅水方程和欧拉方程。我们还想用我们的书来鼓励阅读和出版互动式笔记本。因此, 我们决定, 我们的书除了可以下载的笔记本外, 还应该有印刷版和 HTML 版, 以吸引更多的读者, 并鼓励更多的作者和出版商探索这个互动平台。

正如人们所期望的那样, 出现了几个新的问题, 即如何使一本书在所有不同的平台上兼容, 我们想在这些平台上展示这本书, 特别是由于笔记本大量使用了互动小工具和动画。在我们的出版商 (**SIAM**) 和几个从事基于 **Jupyter** 的交互式书籍出版工具的开发者的帮助下, 我们即将完成这个项目。我们很高兴地说, 这个项目促进了其中一些工具的发展, 这也鼓励了我们的出版商更深入地研究互动式图书出版。

这本书应该在未来几个月内完成, 目前的状态可以在
http://www.clawpack.org/riemann_book/index.html
 查询。

- David I. Ketcheson, Randall J. LeVeque, and Mauricio J. del Razo

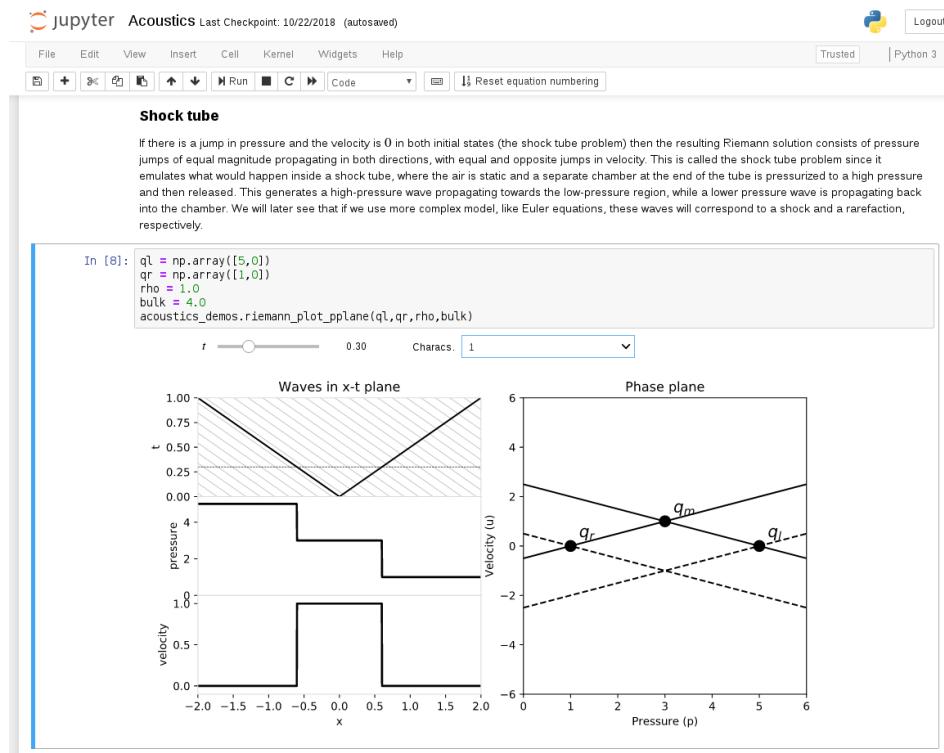
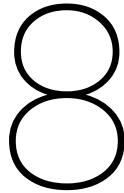


图 7.3: 来自本书声学方程一章的可视化, 网址是: https://github.com/clawpack/riemann_book 它显示了声学方程的黎曼问题解决方案的互动可视化, 其中初始条件模拟了一个冲击管。



关于作者

8.1. 项目负责人

Lorena A. Barba

乔治华盛顿大学 labarba@email.gwu.edu @LorenaABarba Lorena A. Barba 是乔治华盛顿大学机械和航空航天工程系副教授。她在 2013 年采用了 Jupyter，并从那时起在她教授的每门课程中使用它。她的公开课材料广为人知，被成千上万的学习者使用。CFD Python 和 Numerical MOOC 就是最好的例子。

8.2. 定稿阶段的作者

Lecia J. Barker

科罗拉多大学博尔德分校 lecia.barker@colorado.edu @leciab Lecia Barker 是科罗拉多大学博尔德分校信息科学系的副教授和本科研究的副主席。她也是国家妇女与信息技术中心的高级研究科学家。她的研究小组正在研究计算机科学本科教学实践的扩散和采用。Lecia 拥有科罗拉多大学博尔德分校的通信学博士学位和圣地亚哥州立大学的市场营销 MBA 学位。

Douglas Blank

布林莫尔学院 dblank@brynmawr.edu @dougblank 道格拉斯-布兰克是布林莫尔学院计算机科学系的副教授，布林莫尔学院是美国宾夕法尼亚州费城外的一所小型的全女性学院。他拥有印第安纳大学布卢明顿分校认知科学和计算机科学的联合博士学位。20 多年来，道格拉斯教授所有级别的计算机科学课程。在过去的 4 年中，他在课堂上专门使用 Jupyter 笔记本。道格拉斯在计算机科学教育、机器学习、人工智能和深度学习等领域发表过文章。他是 Engage-CS-Edu.org 的顾问委员会成员，这是谷歌和国家妇女与信息技术中心（NCWIT）的联合项目。道格拉斯还在他的网站 douglasblank.com 上写文字和代码。

Jed Brown

科罗拉多大学博尔德分校 jed@jedbrown.org @five9a2 杰德-布朗是科罗拉多大学博尔德分校的计算机科学助理教授。他使用 Jupyter Notebook 和 nbgrader 教授数值和科学计算课程已有三年，并领导一个研究小组，为计算科学开发计算方法和社区软件。

Allen Downey

奥林学院 downey@allendowney.com @AllenDowney Allen Downey 是 Olin 学院的计算机科学教授，也是一系列与软件和数据科学相关的开源教科书的作者，包括由 O'Reilly Media 出版的 Think Python, Think Bayes, 和 Think Complexity。这些书以及基于这些书的课程都广泛使用 Jupyter 笔记本。多尼教授拥有加州大学伯克利分校的计算机科学博士学位，以及麻省理工学院的硕士和学士学位。

Tim George

Jupyter 项目 tgeorgeux@gmail.com Timothy George 是 Project Jupyter 的首席 UI/UX 设计师，主要专注于 JupyterLab。除了他的正式职责外，Tim 还在与 Jupyter 合作设计战略、未来产品、管理、多样性和包容性方面。他在加州大学欧文分校的唐纳德-布伦信息学和计算机科学学院学习人机交互，并获得了硕士学位。

Lindsey Heagy

加州大学伯克利分校 lindseyheagy@gmail.com @lindsey_jh Lindsey Heagy 是加州大学伯克利分校的博士后研究员，从事 Jupyter 项目和 Jupyter 在地球科学领域的研究。她最近在不列颠哥伦比亚大学完成了她的地球物理学博士学位。她是 GeoSci.xyz 的项目负责人，该项目旨在建立合作、互动、基于网络的地球科学教科书，也是 SimPEG 的核心贡献者，这是一个用于地球物理模拟和反演的开源框架。GeoSci.xyz 项目在很大程度上依靠 Jupyter 来使内容变得生动。

Kyle Mandli

哥伦比亚大学 kyle.mandli@columbia.edu @KyleMandli Kyle Mandli 是哥伦比亚大学应用物理和应用数学系的助理教授。他开发了一套以 Jupyter 笔记本为中心的公开课程笔记，并将 Jupyter 与 nbgrader 一起用于家庭作业。他的其他研究兴趣包括开发沿海灾害的计算方法，如风暴潮和海啸。

Jason K. Moore

加州大学戴维斯分校 jkm@ucdavis.edu @moorepants 杰森-K-摩尔是加州大学戴维斯分校机械和航空航天工程的助理教学教授。他目前教授动力学和机械设计相关课程。他利用 Jupyter 笔记本来教授建模和仿真，并正在编写一本关于机械振动的教科书。他负责 LibreTexts 项目中的 Jupyter 相关功能，同时也是 SymPy 和 PyDy 项目的核心开发者，该项目利用 Jupyter 进行培训讲座，例如 PyDy 教程和 SymPy 代码生成教程。杰森拥有加州大学戴维斯分校和老多明尼安大学的机械工程博士、硕士和学士学位。

David Lippert

莱多斯公司 David Lippert 是弗吉尼亚州阿灵顿 Leidos 的一名软件工程师。他主要利用 Jupyter 笔记本进行探索性数据分析以及训练和评估机器学习算法。他曾编写 Jupyter 笔记本来创建新的苏斯博士十四行诗，并评估烂番茄番茄仪是否可信。他拥有米德尔伯里学院的计算机科学学士学位。

Kyle E. Niemeyer

俄勒冈州立大学 kyle.niemeyer@oregonstate.edu @kyleniemeyer Kyle Niemeyer 是俄勒冈州立大学机械、工业和制造工程学院的机械工程助理教授。他教授解微分方程的数值和分析方法以及气体动力学等课程，最近还开发了一门关于工程研究软件开发的研究生课程。他的研究小组开发和应用燃烧和化学反应流体流的建模方法。他也是化学动力学、热力学和传输过程的 Cantera 开源项目的指导委员会成员。

Ryan Watkins

乔治华盛顿大学 rwatkins@gwu.edu @parsingscience 瑞安·沃特金斯是位于华盛顿特区的乔

治华盛顿大学的教育技术教授。他领导人与技术合作 (HTC) 博士项目领域，并教授需求评估、教学设计和研究方法等课程。**Ryan** 的研究重点是人们和组织如何定义和评估需求。他是“解析科学”的联合主持人，这是一个播客，研究人员在这里分享他们科学背后的故事。他还开发了 **We Share Science** 平台，用于分享研究的视频摘要。

Richard H. West

东北大学 R.West@northeastern.edu @Richardhwest 理查德-韦斯特是位于波士顿的东北大学化学工程系副教授。他领导着一个研究小组，研究燃烧或催化等复杂反应系统的计算模型。他是 **Cantera** 开源项目的核心成员。在“化学工程中的计算模型”的选修课上，他将 **Python** 和 **Jupyter** 整合到本科和研究生水平的化学动力学和反应器设计的核心课程中。作为国家科学基金会 CAREER 奖的一部分，他正在开发模块，教学生使用 **Python** 和 **SciPy** 来解决化学工程问题。

Elizabeth Wickes

伊利诺伊大学厄巴纳-香槟分校 wickes1@illinois.edu @ 伊丽莎白-威克斯 Elizabeth Wickes 是伊利诺伊大学厄巴纳-香槟分校信息科学院的一名讲师。她从信息和数据科学的角度讲授基础编程，以及其他关于开放数据和可重复性的课程作业。她的编程课程讲义是用 **Jupyter** 笔记本编写的，课堂上通过现场编码进行教学。

Carol Willing

加州大学圣路易斯奥比斯波分校 willingc@gmail.com @WillingCarol Carol Willing 是 Cal Poly San Luis Obispo 的研究软件工程师，全职从事 **Jupyter** 项目。她是 **Python** 软件基金会研究员和前主任；**Jupyter** 项目指导委员会成员；以及 **CPython** 和 **Jupyter** 的核心开发者。Carol 拥有麻省理工学院的管理学硕士学位和杜克大学的电子工程学士学位。

Michael Zingale

石溪大学 Michael.Zingale@stonybrook.edu @Michael_Zingale Michael Zingale 是石溪大学的副教授和计算天体物理学家。他拥有芝加哥大学的博士学位（2000 年）。他经常讲授科学计算研究生课程的数值方法和 **Python**，大部分内容依靠 **Jupyter** 笔记本和 **Python** 进行演示。他是开放教育资源的倡导者，作为开放天体物理学书架项目的创始人，他在那里主持他的计算天体物理流体力学导论文本。

9

Conclusion

A conclusion...