

2 Using JupyterLab JupyterLab, 用起来!

特别适合探究式学习，代码、绘图、脚本、公式 ...



教育不是为生活做准备；教育就是生活本身。

Education is not a preparation for life; education is life itself.

—— 约翰·杜威 (John Dewey) | 美国著名哲学家、教育家、心理学家 | 1859 ~ 1952



- ◀ `ax.plot_wireframe()` 用于在三维子图 `ax` 上绘制网格曲
- ◀ `fig.add_subplot(projection='3d')` 用于在图形对象 `fig` 上添加一个三维子图
- ◀ `matplotlib.pyplot.figure()` 用于创建一个新的图形窗口或画布，用于绘制各种数据可视化图表
- ◀ `matplotlib.pyplot.grid()` 在当前图表中添加网格线
- ◀ `matplotlib.pyplot.plot()` 绘制折线图
- ◀ `matplotlib.pyplot.scatter()` 绘制散点图
- ◀ `matplotlib.pyplot.subplot()` 用于在一个图表中创建一个子图，并指定子图的位置或排列方式
- ◀ `matplotlib.pyplot.subplots()` 创建一个包含多个子图的图表，返回一个包含图表对象和子图对象的元组
- ◀ `matplotlib.pyplot.subplots()` 创建一个包含多个子图的图表，返回一个包含图表对象和子图对象的元组
- ◀ `matplotlib.pyplot.title()` 设置当前图表的标题，相当于对于特定轴 `ax` 对象 `ax.set_title()`
- ◀ `matplotlib.pyplot.xlabel()` 设置当前图表 `x` 轴的标签，相当于对于特定轴 `ax` 对象 `ax.set_xlabel()`
- ◀ `matplotlib.pyplot.xlim()` 设置当前图表 `x` 轴显示范围，相当于对于特定轴 `ax` 对象 `ax.set_xlim()` 或 `ax.set_xbound()`
- ◀ `matplotlib.pyplot.xticks()` 设置当前图表 `x` 轴刻度位置，相当于对于特定轴 `ax` 对象 `ax.set_xticks()`
- ◀ `matplotlib.pyplot.ylabel()` 设置当前图表 `y` 轴的标签，相当于对于特定轴 `ax` 对象 `ax.set_ylabel()`
- ◀ `matplotlib.pyplot.ylim()` 设置当前图表 `y` 轴显示范围，相当于对于特定轴 `ax` 对象 `ax.set_ylim()` 或 `ax.set_ybound()`
- ◀ `matplotlib.pyplot.yticks()` 设置当前图表 `y` 轴刻度位置，相当于对于特定轴 `ax` 对象 `ax.set_yticks()`
- ◀ `numpy.arange()` 生成一个包含给定范围内等间隔的数值的数组
- ◀ `numpy.linspace()` 生成在指定范围内均匀间隔的数值，并返回一个数组
- ◀ `numpy.meshgrid()` 用于生成多维网格化数据
- ◀ `plotly.express.data.iris()` 从 Plotly 库里加载鸢尾花数据集
- ◀ `plotly.express.scatter()` 绘制可交互的散点图
- ◀ `plotly.graph_objects.Figure()` 用于创建一个新的图形对象，用于绘制各种交互式数据可视化图表
- ◀ `plotly.graph_objects.Surface()` 绘制可交互的网格曲面
- ◀ `seaborn.scatterplot()` 绘制散点图



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

2.1 啥是 JupyterLab?

JupyterLab 集合“浏览器 + 编程 + 文档 + 绘图 + 多媒体 + 发布”众多功能与一身。“鸢尾花书”不同场合反复提过，对于初学者，哪怕是有开发经验的读者来说，JupyterLab 都特别适合探究式学习。目前《数学要素》、《可视之美》中，几乎所有的代码都是用 JupyterLab 写的。如果大家对 JupyterLab 反馈正面，其他分册也考虑提供 Jupyter Notebook 配套文件。

这一话题将和大家聊一聊如何使用 JupyterLab。注意，本节不求“事无巨细”地介绍 JupyterLab，而是要全景地浏览 JupyterLab 的主要功能，保证“够用就好”，以便大家轻装上阵。

对于 JupyterLab 的外观、窗口布局等细节问题，这个话题就不展开了，大家如果有需要可以很容易搜索到结果。当大家对 JupyterLab 熟悉之后，建议大家了解如何用 JupyterLab 的 debug 功能。此外，很多开发者专门针对 JupyterLab 开发各种小插件，很多插件的确能提高工作效率，也建议大家自行了解。

大家 JupyterLab 用熟之后，会发现这一节最重要的内容只有——快捷键。



什么是 JupyterLab?

JupyterLab 是一个交互式开发环境，可以让用户创建和共享 Jupyter 笔记本、代码、数据和文档。它是 Jupyter Notebook 的升级版，提供了更强大的功能和更直观的用户界面。JupyterLab 支持多种语言，包括 Python、R、Julia 和 Scala 等。它还提供了多个面向数据科学的扩展，如 JupyterLab Git、JupyterLab LaTeX 和 JupyterLab Debugger 等，使得数据科学家和开发人员可以更加高效地进行数据分析、机器学习和模型开发等工作。JupyterLab 的主要特点包括：基于 web 的用户界面，可以让用户同时在一个界面中管理多个笔记本和文件。支持多种文件格式，包括 Jupyter 笔记本、Markdown 文档、Python 脚本和 CSV 文件等。可以通过拖放和分栏等方式来组织和管理笔记本和文件。提供了一组内置的编辑器、终端、文件浏览器和输出查看器等工具。可以通过扩展系统来扩展和定制 JupyterLab 的功能。

2.2 使用 JupyterLab: 立刻用起来

新建 Notebook

大家首先通过 Anaconda Navigator (上一节内容) 打开 JupyterLab。

如图 1 所示，不管点击 A 或 B 都会看到 C 这个图标，点击 C 就会生成一个 Notebook。此外，新建 Notebook 前，点击图 1 中 D，我们可以改变文件路径。

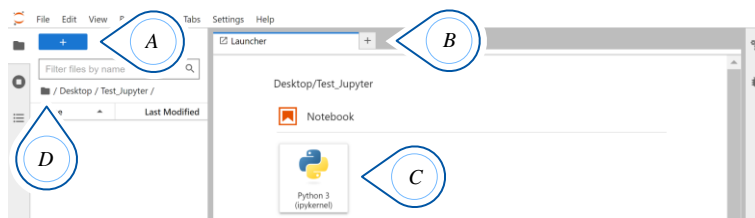


图 1. 新建 Notebook

如图 2 所示，Notebook 界面的有很多板块。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

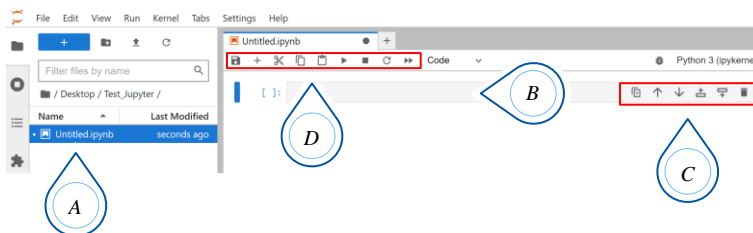


图 2. JupyterLab 中新建 Notebook 界面



JupyterLab 中的 cell 是什么？

在 JupyterLab 中，Cell (单元格) 是指一个包含代码或文本的矩形区域，它是用户编写和执行代码、编写文本和渲染 Markdown 的基本单位。Cell 可以包含多种类型的内容，包括代码、Markdown、LaTeX 公式等。JupyterLab 中的 Cell 可以通过交互式的方式进行编辑和执行。例如，在 Code Cell 中，用户可以编写 Python 代码，并使用 Shift+Enter 快捷键执行代码并显示结果；在 Markdown Cell 中，用户可以使用 Markdown 语法编写文本，并使用 Shift+Enter 快捷键渲染 Markdown 文本。JupyterLab 中的 Cell 还支持多种交互式扩展，例如使用 IPython Magic 命令、使用自动完成、代码补全和代码调试等。Cell 也可以被复制、剪切、粘贴、移动和删除，使得用户可以轻松地组织和管理笔记本中的内容。

对于初学者，大家先注意 4 点：

- ▶ 图 2 中的 A 对应的是 Notebook 默认的名字。右键可以对文件进行各种操作，比如重命名、剪切、复制、粘贴、删除等等。
- ▶ 图 2 中的 B 是 Notebook 中第一个 cell。在 Notebook 里，一个基本的代码块被称作一个 cell。注意，一个 Notebook 可以有若干 cell；而一个 cell 理论上可以有无数行代码。
- ▶ 图 2 中的 C 对应的是 cell 的几个常见操作——复制并向下粘贴、向上、向下、向上加 cell、向下加 cell、删除 cell。
- ▶ 图 2 中的 D 对应的操作——保存文件、向下加 cell、剪切 cell、复制 cell、粘贴 cell、运行当前 cell 后移动 (或创建) 到下一个 cell、停止运行、重启 kernel、重启重跑所有 cell、code/markdown 转换。

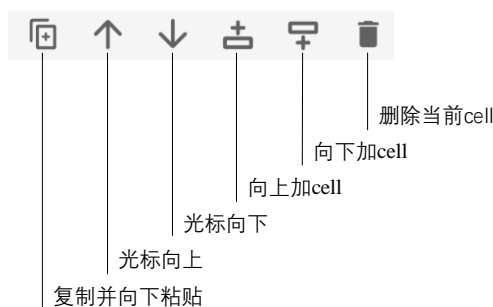


图 3. C 对应的是 cell 的几个常见操作

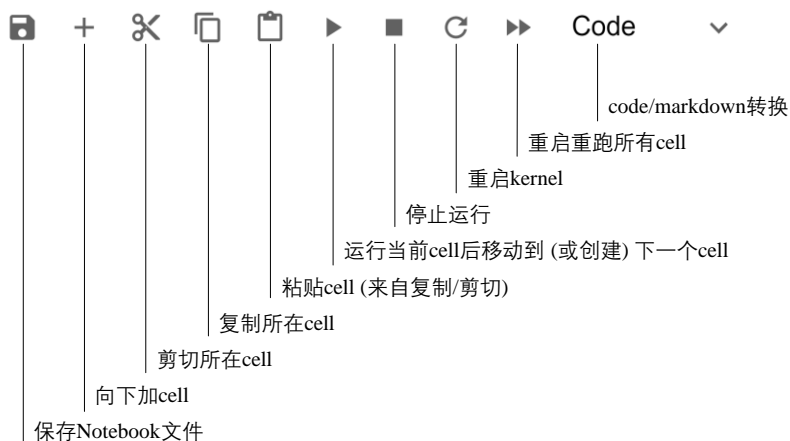


图 4.4 对应的是 cell 的几个常见操作



JupyterLab 中的 kernel 是什么？

JupyterLab 中，内核 (kernel) 是指与特定编程语言交互的后台进程，它负责编译和执行用户在 JupyterLab 中编写的代码，并返回执行结果。内核与 JupyterLab 之间通过一种称为“Jupyter 协议”的通信协议进行交互。打开一个新的 notebook 或 console 时，JupyterLab 会自动启动一个内核，这个内核将与该 notebook 或 console 中编写的代码进行交互。在 notebook 或 console 中编写代码，并使用内核来执行它们。内核还可以保存笔记本中的变量和状态，使得大家可以在多个代码单元格之间共享变量和状态。JupyterLab 支持多种编程语言的内核，可以在启动 notebook 或 console 时选择要使用的内核。例如，如果想使用 Python 内核，可以选择“Python 3”内核。一旦选择了内核，JupyterLab 将与该内核建立连接，并使用它来执行该 notebook 或 console 中编写的代码。如果希望在 notebook 或 console 中使用其他语言的内核，需要先安装并配置这些内核。

代码 vs 文本

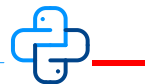
Jupyter 的 cell 常用两种状态——代码、文本。文本也叫 markdown。两种状态之间可以相互转换。

顾名思义，代码状态的 cell 中的内容会被视为“代码”，# 开头的部分会被视作为“注释”

文本 markdown 状态下，整个 cell 的内容可以是文本/Latex 公式/超链接/图片等等，这个 cell 不会被当成代码执行。图 4 中的“code/markdown”选项可以帮助我们在这两种 cell 状态切换。

我们常在 JupyterLab 中敲入各种 Latex 公式，本书后续将会见缝插针地讲解如何用 Latex 写各种公式。

多数时候为了提高切换效率，我们通常使用快捷键。下面介绍 JupyterLab 中常用的快捷键。



本节配套的 Jupyter Notebook 文件 BK_2_Topic_1.02_1.ipynb 向大家展示如何在 Jupyter Notebook 中进行探究式学习。本节配套的微课视频会逐 cell 讲解这个 Notebook 文件。



JupyterLab 中的 markdown 是什么？

在 JupyterLab 中，Markdown 是一种轻量级标记语言，可以用于编写文档、笔记和报告等。通过使用 Markdown 语法，用户可以在 JupyterLab 中轻松地创建格式化文本、插入图片、添加链接、创建列表等。Markdown 语法非常简单，易于学习和使用。例如，使

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

用 Markdown 语法，用户可以使用井号 (#) 来创建标题，使用 “-” 或 “*” 符号加上空格来创建 bullet list，使用双星号(**)来加粗文本，使用单星号(*)来斜体文本等。用户可以在 Markdown 单元格中编写 Markdown 语法，然后使用 Shift+Enter 键来渲染 Markdown 文本。JupyterLab 中的 Markdown 支持 LaTeX 语法，用户可以使用 LaTeX 语法来插入数学公式，从而方便地创建数学笔记和报告。

2.3 快捷键：这一章最有用的内容

建议大家使用快捷键完成常见 cell 操作。JupyterLab 的快捷键分成两种状态：a) 编辑模式；b) 命令模式。

编辑模式，允许大家向 cell 中敲入代码或 markdown 文本。表 1 总结编辑模式下常用快捷键。为了帮助大家识别这些快捷键组合，图 5 给出标准键盘主键盘上各个按键的位置。

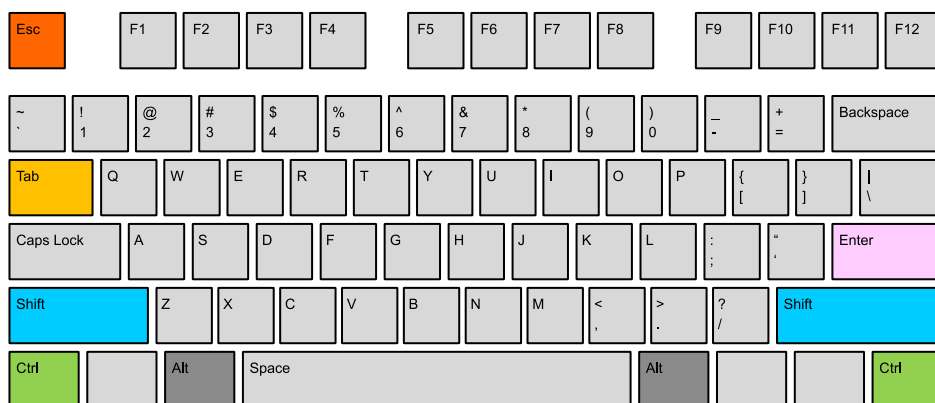


图 5. 标准键盘，Mac 的 command 对应 ctrl

命令模式，单击 **esc** 进入命令模式，这时可以通过键盘键入命令快捷键。表 2 总结命令模式下常用快捷键。

注意，表格中的加号 + 表示“一起按下”，不是让大家按加号键。加号 + 前后的按键没有先后顺序。

此外，本书 GitHub 中还给出 JupyterLab 快捷键的 cheat sheet，建议大家专门将其打印出来，编程的时候放在一边参考。

表 1 和表 2 两个表格中都是常用默认快捷键。如果大家对某个快捷键组合不满意，可以自行修改。特别是需要在多个 IDE 之间转换时，由于不同 IDE 的默认快捷键不同，一般都会将常用快捷键统一设置成自己习惯的组合。JupyterLab 中修改快捷键的路径为 Settings → Advanced Settings Editor (或 esc → ctrl + ,) → 搜索 Keyboard Shortcuts。注意，不建议初学者修改默认快捷键。

表 1. 编辑模式，常用快捷键

快捷键组合	功能
esc	进入“命令”模式；鼠标左键单击任何 cell 返回，或单击 enter 返回编辑模式
ctrl + M	进入“命令”模式

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

ctrl + S	保存；尽管 JupyterLab 会自动保存，建议大家还是要养成边写边存的好习惯
shift + enter	执行 + 跳转；运行当前 cell 中的代码，光标跳转到下一 cell
ctrl + enter	执行；运行当前 cell 中的代码
alt + enter	执行 + 创建 cell；运行当前 cell 中的代码，并在下方创建一个新 cell
ctrl + shift + -	分割；在光标所在位置将代码/文本分割成两个 cells
ctrl + /	注释/撤销注释；对所在行，或选中行进行注释/撤销注释操作
ctrl + [向左缩进；行首减四个空格
ctrl +]	向右缩进；行首加四个空格
ctrl + A	全选；全选当前 cell 内容
ctrl + Z	撤销；撤销上一个键盘操作
ctrl + shift + Z	重做：恢复刚才撤销命令对应操作，相当于“撤销撤销”
ctrl + C	复制；复制选中的代码或文本
ctrl + X	剪切；剪切选中的代码或文本
ctrl + V	粘贴；粘贴复制/剪切的代码或文本
ctrl + F	查询；实际上就是浏览器的搜索
home	跳到某一行开头
end	跳到某一行结尾
ctrl + home	跳到多行 cell 第一行开头
ctrl + end	跳到多行 cell 最后一行结尾
tab	代码补齐；忘记函数拼写时，可以给出前一两个字母，按 tab 键得到提示
shift + tab	对键入的函数提供帮助文档
ctrl + B	展开/关闭左侧 sidebar

表 2. 命令模式，常用快捷键

快捷键组合	功能
esc	编辑模式下，进入“命令”模式；鼠标左键单击任何 cell 返回，或单击 enter 返回编辑模式
esc → M	在按下 esc 进入编辑模式后，将当前 cell 从代码 markdown 转成文本
esc → Y	将当前 cell 从文本 markdown 转成代码
enter	从命令模式进入编辑模式，或者鼠标左键单击任何 cell
esc → A	插入；在当前 cell 上方插入新 cell
esc → B	插入；在当前 cell 下方插入新 cell
esc → D → D	删除；在按下 esc 进入编辑模式后，连续按两下 D，删除当前 cell
esc → 0 → 0	重启 kernel；在按下 esc 进入编辑模式后，连续按两下零 0，重启 kernel
esc → ctrl + B	展开/关闭左侧 sidebar
esc → ctrl + A	选中所有 cells
esc → shift + ▲	选中当前和上方 cell，不断按 shift + ▲ 不断选中更上一层 cell
esc → shift + ▼	选中当前和下方 cell，不断按 shift + ▼ 不断选中更下一层 cell
shift + M	合并；将所有选中的 cells 合并；如果没有多选 cell，则将当前 cell 和下方 cell 合并






本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

 + 	执行 + 跳转；运行当前 cell 中的代码，光标跳转到下一 cell；和编辑模式一致
 + 	执行；运行当前 cell 中的代码；和编辑模式一致
 + 	执行 + 创建 cell；运行当前 cell 中的代码，并在下方创建一个新 cell；和边际模式一致
 → ①	一级标题，等同于 markdown 状态下 #
 → ②	二级标题，等同于 markdown 状态下 ##
 → ③	三级标题，等同于 markdown 状态下 ###，以此类推



这一章的习题很简单，请大家从零开始复刻 Bk1_Ch2_01.ipynb，并在创建 Jupyter Notebook 文档的过程使用快捷键。

* 这道题目很基础，本书不给答案。