



Python 101

聊聊“巨蟒”

有用废话多是 ChatGPT 生成，安装测试 Anaconda 是关键




悟知天地纵横交错，万物相生相联之网。而你我也系其中一环，一念一动皆牵动周身。

There is urgency in coming to see the world as a web of interrelated processes of which we are integral parts, so that all of our choices and actions have consequences for the world around us.

—— 阿尔弗雷德·怀特海 (Alfred Whitehead) | 英国数学家、哲学家 | 1861 ~ 1947



本节很多问题都采用 ChatGPT 答案，作者对回答文本略加编辑。本书中，ChatGPT 的答案用  标识。建议大家在学习时，不管是概念、程序、数学上遇到问题，都可以使用类似 ChatGPT 工具作为助手。值得注意的是，ChatGPT 经常犯错，大家注意甄别。



什么是 ChatGPT?

ChatGPT 是一种基于自然语言处理 (Natural Language Processing, NLP) 技术的人工智能 (AI) 应用程序，它是由 OpenAI 公司开发的一种大规模预训练语言模型。ChatGPT 使用深度神经网络来模拟人类的对话过程，它可以理解和生成人类语言，可以用于实现聊天机器人、智能客服、智能助手等应用。ChatGPT 还可以帮助用户进行文本自动生成、文本摘要、文本分类、情感分析等任务。ChatGPT 使用 Python 编程语言进行搭建。在搭建 ChatGPT 时，OpenAI 使用了 Python 的深度学习框架 TensorFlow 和 PyTorch，以及一些其他的 Python 库和工具来完成。

Python 有啥用?

Python 的用途极为广泛，特别是在机器学习、深度学习领域。Python 免费开源，语言语法友好，而且社区活跃。这就是为什么“鸢尾花书”系列会选择 Python 作为编程语言的原因。

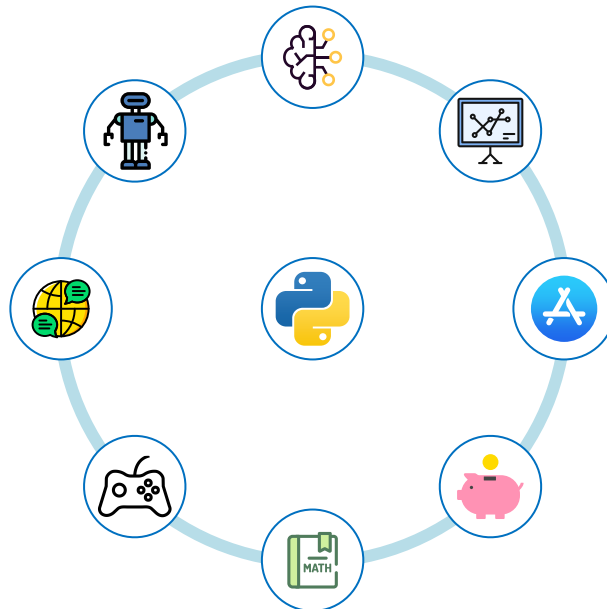


图 1. Python 应用场景

如图 1 所示，Python 具有广泛的用途，比如：

- ▶ 机器学习：Python 在数据科学和机器学习领域非常受欢迎，因为它有很多强大的库和工具，如 NumPy、Pandas、Scikit-learn 等。
- ▶ 深度学习：Python 的深度学习工具，比如 PyTorch、TensorFlow，常用来开发各种人工智能应用，比如智能设备、无人驾驶、自然语言处理工具等。
- ▶ Web 开发：Python 可以用于 Web 开发，有许多流行的 Web 框架，如 Django、Flask 等。
- ▶ 自动化脚本：Python 可以用于自动化任务，例如自动备份、自动化测试、爬虫等。

- ▶ 游戏开发：Python 可以用于游戏开发，如 Pygame 等库和工具。
- ▶ 系统管理和网络编程：Python 可以用于系统管理和网络编程，例如网络爬虫、服务器开发、安全工具等。



什么是 Python?

Python 是一种高级编程语言，由 Guido van Rossum 于 1991 年首次发布。它是一种解释型语言，使用动态类型系统和自动内存管理。Python 具有简单易学、易于阅读和编写、可移植性强等特点，广泛应用于 Web 开发、数据分析、人工智能、科学计算、自动化等领域。Python 语言具有丰富的标准库和第三方库，可支持各种编程任务，例如文件处理、网络编程、GUI 开发、图像处理、数据可视化、机器学习等。Python 的语法简洁清晰，易于阅读和理解，因此也被广泛应用于教育和科研领域。Python 语言的解释器可用于不同的操作系统，例如 Windows、MacOS、Linux 等，因此 Python 具有很好的跨平台性。

图 2 所示为本书涉及到的 9 个重要的 Python 库。



图 2. 《编程不难》涉及到的 Python 库



Python 中，什么是模块、包、库？

在 Python 中，模块、包、库是三个常见的概念。它们的含义如下：

模块 (Module): 是一个 Python 程序文件，包含了一组相关的函数、类、变量和常量等，可供其他程序引用。Python 中的模块是一种可重用的代码组件，可用于将相关的代码组织到一起，以便更好地管理和维护代码。一个模块可以包含多个函数、类、变量和常量等，可以被其他模块或程序引用和调用。

包 (Package): 是一组相关的模块的集合，用于组织 Python 代码的层次结构。一个包是一个文件夹，其中包含其他模块或子包。包是一种通过模块命名空间进行模块组织的方式，可用于更好地组织和管理大型代码库。

库 (Library): 是由一组模块和包组成的软件组件，提供了一系列函数、类、变量和常量等，用于解决特定问题。Python 标准库是 Python 官方提供的一组库，包含了大量的模块和功能，可以直接使用。此外，还有第三方库，如 NumPy、Pandas、Matplotlib 等，用于数据处理、科学计算、可视化等领域。

需要注意的是，模块是最小的可重用代码单元，而包和库是由多个模块组成的更大的结构。在 Python 中，通常使用 `import` 语句来引入需要使用的包、库或模块。

我们为什么要学 Python?

作者认为，面向人工智能时代的教育，特别是数学教育，必须结合编程、可视化、应用。而 Python 既是编程工具，也拥有大量可视化工具，同时可以用来完成各种数据科学、机器学习任务。“编程 + 可视化 + 数学 + 机器学习”正是“鸢尾花书”创作的内核，贯穿整套图册始终。

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

对于初高中生、大学生，学习 Python 有很多好处，比如：

- ▶ 培养编程思维：Python 作为一种编程语言，可以帮助大家培养编程思维能力。大家可以通过编写简单的程序和解决各种问题，锻炼逻辑思维、问题解决和创造力等能力。
- ▶ 高效地学习数学及其他学科：将公式、模型写成 Python 代码的过程，本身就是一种“习题”。而且这类习题比传统课本习题更能激发大家的兴趣。
- ▶ 图形化强化记忆：公式、定理、定义、解题技巧 ... 大家考完试也就忘记了。但是利用 Python 编程，把公式、定理、定义变成一幅幅活生生的图形之后，这些概念将会深深地刻在大家脑中，甚至一辈子不会忘记。
- ▶ 提高学习效率：Python 可以用于自动化各种重复性的任务，如数据处理、文本处理等。大家可以通过编写 Python 程序来自动化这些任务，从而节省时间和精力，提高学习效率。
- ▶ 为未来的学习和职业做准备：Python 是一种非常流行的编程语言，它在数据科学、机器学习、人工智能等领域有很多应用，大家可以通过学习 Python，为未来的学习和职业做准备，提前掌握一些必要的技能和知识。
- ▶ 增强实践能力：学习 Python 可以让大家更容易地将学到的知识应用到实践中，例如编写简单应用程序、游戏、网站等，这有助于大家增强实践能力和探索新领域的能力。
- ▶ 培养团队合作意识：学习 Python 可以让大家更容易地与他人合作，例如在编写程序时可以分工合作、交流想法等，这有助于培养团队合作意识和沟通能力。

未来已来，你来不来？

怎么学 Python？

和汉语、英语一样，Python 也是语言。只不过，Python 是人和机器交互的语言。Python 的语法有绝对的对错，不能模棱两可。错误的命令，要么出 bug，要么得出错误的结果。

大家学 Python 时，一定要吸取英语教育的教训。学了十几年的哑巴英语，原因就是只关注词汇、语法、应试，而不是利用语言进行口头、书面交流。这是学 Python 必须避免。建议大家在学 Python 编程的时候，一定要抱着极强的“功利心”——立刻开始 coding、debugging！



学习 Python 和学习英语有什么相似之处？

学习 Python 和学习英语有一些相似之处，以下是其中一些：

都需要掌握基础知识：学习 Python 和学习英语都需要掌握基础知识，Python 的基础语法、数据类型、流程控制语句、函数等，英语的基本词汇、语法、发音等。

都需要不断练习：学习 Python 和学习英语都需要不断地练习，Python 需要编写程序来实践，英语需要口语练习和写作练习。

都需要实践和应用：学习 Python 和学习英语都需要不断地实践和应用，Python 可以应用到数据处理、人工智能、游戏开发等领域，英语可以应用到国际交流、留学、工作等方面。

都需要耐心和坚持：学习 Python 和学习英语都需要耐心和坚持，需要花费大量时间和精力来学习和练习，才能达到良好的掌握和应用水平。

总之，学习 Python 和学习英语都需要掌握基础知识、不断练习、实践和应用，同时也需要耐心和坚持。虽然二者是不同的领域，但都是对自己未来发展非常有帮助的技能。

鸢尾花书有编程、可视化、数学、机器学习这几个重要的元素，而这几个元素都离不开 Python。下面我们逐一聊一下。

Python 和可视化什么关系？

Python 和可视化有很密切的关系。Python 中有很多强大的可视化库和工具，可以帮助用户对数据进行可视化呈现。

以下是 Python 和可视化的一些关系：

- ▶ 数据可视化：Python 中有许多数据可视化的库，例如 Matplotlib、Seaborn、Plotly 等，可以帮助用户将数据可视化呈现出来，从而更好地理解数据的分布、趋势等信息。本书的绘图部分将蜻蜓点水地讲解 Matplotlib、Seaborn、Plotly 常用绘图命令。“鸢尾花书”的《可视之美》一册将
- ▶ 图像处理：Python 中有许多图像处理的库，例如 OpenCV 等，可以帮助用户进行图像处理和分析，同时也可以将处理后的图像进行可视化呈现。
- ▶ 交互式可视化：Python 中也有许多用于交互式可视化的库，例如 Bokeh、Altair 等，可以帮助用户建立交互式的数据可视化应用程序。
- ▶ 3D 可视化：Python 中也有许多用于 3D 可视化的库，例如 Mayavi、VisPy 等，可以帮助用户对三维数据进行可视化呈现。

Python 和学数学什么关系？

Python 和数学有着密切的关系。Python 是一种非常适合数学建模和数据分析的编程语言，拥有大量的数学计算库和工具。

以下是 Python 和数学的一些关系：

- ▶ 数学计算：Python 中有很多用于数学计算的库和工具，例如 NumPy、SciPy 等，可以帮助用户进行矩阵运算、微积分、最优化、统计分析等数学计算任务。
- ▶ 数据分析：Python 中有很多用于数据分析的库和工具，例如 Pandas、Matplotlib、Seaborn 等，可以帮助用户对数据进行统计分析、可视化呈现等。
- ▶ 数学建模：Python 中还有很多用于数学建模的库和工具，例如 SymPy 等，可以帮助用户进行数学建模和优化任务。
- ▶ 教学和研究：Python 也被广泛应用于数学教学和研究领域，例如用 Python 实现数学实验、数学模型的探索、算法的实现等。

Python 和机器学习什么关系？

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

Python 与机器学习有非常密切的关系。Python 是一种简单易学、可读性强的编程语言，同时也拥有丰富的第三方库和工具，这使得 Python 成为机器学习领域的重要工具之一。

机器学习是一种应用人工智能的技术，通过让计算机从数据中学习并改善性能，来实现对未知数据的预测和决策。

Python 在机器学习领域的应用非常广泛，主要有以下几个方面：

- ▶ 数据处理和分析：Python 中有许多用于数据处理和分析的库，例如 Pandas、NumPy 和 SciPy，这些库能够帮助用户轻松地处理和分析数据。
- ▶ 机器学习框架：Python 中也有许多用于机器学习的框架，例如 TensorFlow、PyTorch 和 Scikit-Learn 等，这些框架可以帮助用户更加高效地进行机器学习建模和预测。
- ▶ 可视化工具：Python 中的 Matplotlib 和 Seaborn 等可视化库，可以帮助用户更加清晰地理解数据和模型，以及呈现结果。
- ▶ 自然语言处理：Python 中的自然语言处理库，例如 NLTK 和 Spacy 等，可以帮助用户进行文本数据的处理、分析和预测。



什么是机器学习？

机器学习是一种人工智能技术，它使计算机系统能够通过数据和经验自主学习和改进，而无需显式地编程指令。简单来说，机器学习是通过训练算法从数据中学习模式和规律，然后利用这些模式和规律来进行预测或决策。在机器学习中，模型是通过训练算法从大量数据中学习而来的，这些数据被称为训练数据集。训练数据集包含已知结果的输入输出对，这些输入输出对用于训练模型来预测未知数据的输出。训练数据集中的数据越多，训练时间越长，模型就越准确。机器学习可以应用于各种领域，例如语音识别、图像识别、自然语言处理、推荐系统和金融分析等。它已成为当今科技领域中最热门和最具前途的领域之一。

Python 集成开发环境

Python 有很多常用的 IDE (集成开发环境)，比如：

- ▶ JupyterLab：基于 Web 的交互式开发环境，支持多种编程语言，包括 Python，可以快速编写、测试和共享代码，非常适合数据科学和机器学习领域。作者认为，JupyterLab 和 Jupyter Notebook 非常适合大家做探究式学习。目前，《编程不难》、《可视之美》两册的配套的代码多是 Jupyter 笔记。这个话题后文将详细介绍如何使用 JupyterLab。
- ▶ Spyder：基于 Qt 开发的 Python IDE，提供了一个集成的开发环境，包括编辑器、调试器和控制台，非常适合科学计算和数据分析。虽然“鸢尾花书”剩余几册的代码都是在 Spyder 中完成，建议初学者还是在 JupyterLab 中分段运行代码。对于 MATLAB 转 Python 的读者来说，Spyder 可能是最容易上手的 IDE。在所有的 Python IDE 中，Spyder 最像 MATLAB。
- ▶ PyCharm：JetBrains 公司开发的跨平台 Python IDE，提供了许多功能，包括代码智能提示、代码自动完成、调试和单元测试等。建议有 Python 开发经验的读者使用 PyCharm 运行本书代码。



什么是集成开发环境？

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

集成开发环境 (Integrated Development Environment, 简称 IDE) 是一种用于软件开发的工具。它通常包括一个代码编辑器、一个调试器和一个构建工具, 以及其他功能, 例如自动补全、语法高亮、代码重构等。IDE 的目的是提供一个集成的工作环境, 使开发人员能够更高效地编写、调试和测试代码。使用 IDE 可以极大地提高开发效率。例如, 它可以帮助开发人员在编写代码时自动补全函数名称、参数等, 减少打错代码的风险; 它可以提供一些调试工具来检测和修复代码中的错误, 使得开发人员更容易发现问题; 它可以通过自动构建工具来编译和构建代码, 减少手动操作的繁琐过程。总之, IDE 是一种开发人员必备的工具, 可以让开发人员更加专注于编写高质量的代码。

表 1. 比较三个常用的 IDE

维度	JupyterLab	Spyder	PyCharm
适用场景	数据科学、机器学习、交互式	科学计算、数据分析	通用编程、开发
编辑器	基于 Web 的文本编辑器	Qt 构建的文本编辑器	IntelliJ IDEA 编辑器
调试器	内置的交互式调试器	内置的调试器	内置的调试器
插件支持	丰富的插件生态系统	插件支持较少	丰富的插件生态系统
社区支持	由 Jupyter 项目支持	由 Spyder 社区支持	由 JetBrains 公司支持
扩展性	支持自定义和扩展	可以自定义外观和行为	支持自定义和扩展
学习曲线	平缓	友好	稍微陡峭
收费?	免费	免费	有免费和付费版本
平台支持	支持 Windows、Mac 和 Linux	支持 Windows、Mac 和 Linux	支持 Windows、Mac 和 Linux

Anaconda

Anaconda 可谓“科学计算全家桶”, 包含科学计算领域可能用到的大部分 Python 工具, 包括 Python 解释器、常用的第三方库、包管理器、IDE 等。前文提到的 JupyterLab、Spyder、PyCharm 这三个 IDE 都在 Anaconda 中。

什么是 Anaconda?

Anaconda 是一个流行的 Python 发行版, 由 Anaconda, Inc. 开发和维护, 旨在为数据科学、机器学习和科学计算提供一个全面的工具包。Anaconda 集成了许多常用的 Python 库和工具, 如 NumPy、SciPy、Pandas、Matplotlib、Scikit-learn、Jupyter Notebook 等。它还包括一个名为 conda 的软件包管理器, 可以帮助用户安装、更新和管理 Python 库和依赖项。Anaconda 还提供了名为 Anaconda Navigator 的图形用户界面, 用户可以通过这个界面轻松地管理他们的 Python 环境、安装和卸载库、启动 Jupyter Notebook 等操作。除了 Python 环境和库之外, Anaconda 还包括许多其他工具 and 应用程序, 如 Spyder、PyCharm、VS Code、R 语言环境等等, 使得它成为数据科学家和研究人员的的首选工具之一。Anaconda 可以安装在多个平台上, 包括 Windows、Linux 和 Mac OS X。

安装 Anaconda

下文手把手教大家如何在 Windows 上安装、测试 Anaconda, 有经验的读者可以跳过。

对于 Mac 用户, 大家可以参考如下链接安装 Anaconda:

<https://docs.anaconda.com/anaconda/install/mac-os/>

要是想特别安装某个版本的 Python, 请参考:

https://pythonhowto.readthedocs.io/zh_CN/latest/install.html

注意, Anaconda 安装后大概占用 5G 空间。有 Python 开发经验的读者, 可以根据需求自行分别安装 JupyterLab、Spyder、PyCharm。

在 Windows 上安装 Anaconda 可以按照以下步骤进行：

a) 下载。在 Anaconda 官网 (<https://www.anaconda.com/>) 下载适合大家操作系统的 Anaconda 版本，选择对应的 Python 版本（一般建议选择最新版 Python3.x），并下载对应的安装程序。注意，Anaconda 不断推出新版本，大家下载的版本号肯定和下图的版本号不同。建议大家从官网下载最新版本安装程序。



图 3. 安装程序图标

b) 运行安装程序：下载完毕后，双击下载文件运行安装程序。在安装程序打开后，点击“Next”进入下一步。

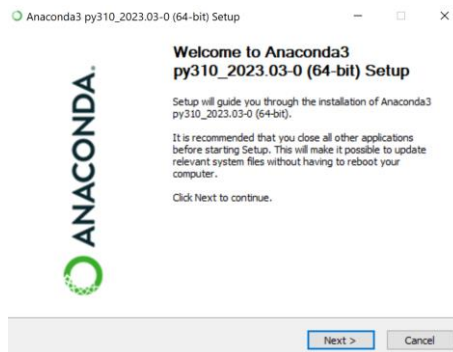


图 4. 运行安装程序

c) 阅读协议：阅读协议并同意“I Agree”，然后点击“Next”。

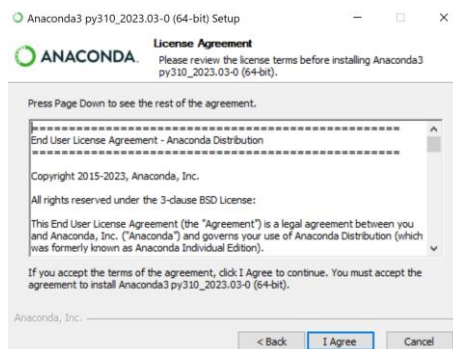


图 5. 阅读协议

d) 安装类型：推荐默认“Just Me”；对于多用户 PC，可以选择“All Users”；然后点击“Next”。

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

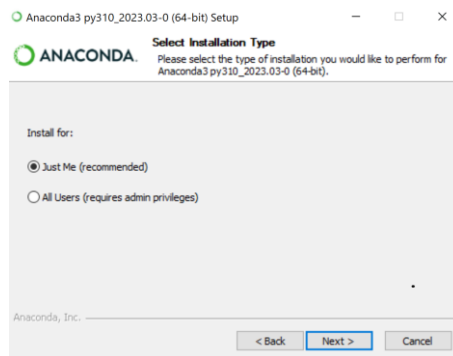


图 6. 安装类型

e) 安装路径：可以指定 Anaconda 的安装路径 (建议零基础读者选择默认路径)，然后点击“Next”。

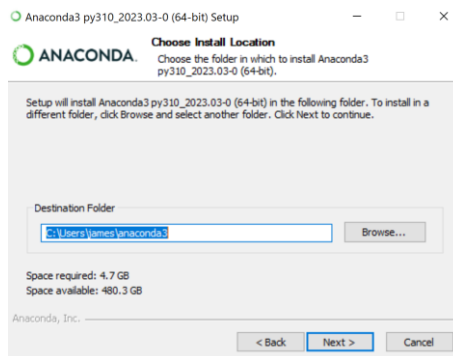


图 7. 安装路径

f) 配置环境变量：选择是否将 Anaconda 添加到系统环境变量中，建议勾选该选项，这样就可以在命令行中使用 Anaconda 的工具了。然后点击“Install”进行安装。

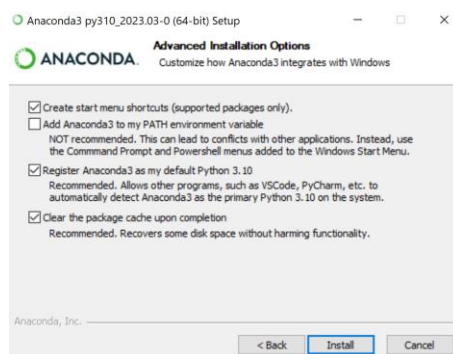


图 8. 安装选择

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

g) 等待安装完成：安装过程可能持续 10 分钟左右。等待安装完成后，会弹出“Installation Complete”对话框，点击“Next”。如果这步持续时间过长（超过一小时），建议强制停止安装，删除安装包。关机再开机，重新下载安装包从头开始再尝试安装。

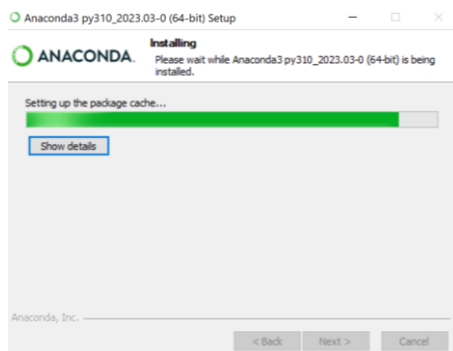


图 9. 等待安装完成

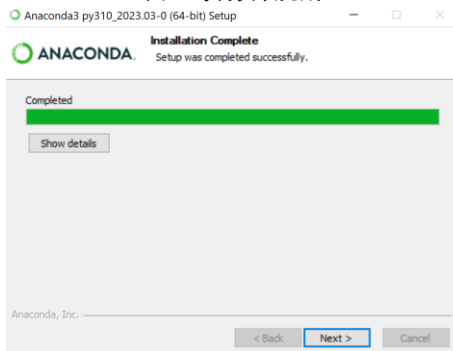


图 10. 安装完成

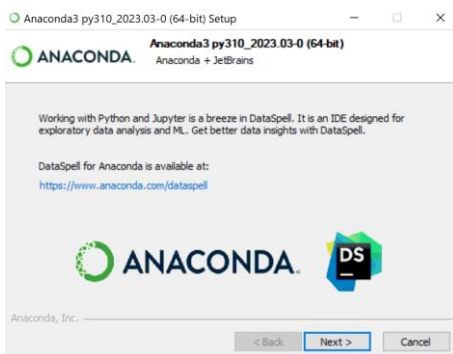


图 11. 广告时间，点 Next

h) 完成安装：点击“Finish”完成 Anaconda 的安装。之后会跳出两个网页，不需要理会，关闭即可。



图 12. 确认完成

安装完成后，可以在“开始菜单”中找到 Anaconda 的安装目录，并启动“Anaconda Navigator”来使用 Anaconda 的工具和功能。同时，也可以在命令行中使用 Anaconda 的工具和命令，例如使用“conda”命令来管理 Python 的虚拟环境和安装依赖包等。

测试 JupyterLab

这是本节最后，也是最关键的一个任务。

要打开并测试 JupyterLab，可以按照以下步骤进行：

a) 找到并打开 Anaconda Navigator (需要 1 分钟左右，稍安勿躁)，点击 JupyterLab 对应的 Launch。马上一个网页将会跳出来，建议大家默认使用 Chrome 浏览器，Firefox 或 Edge 也都可以。

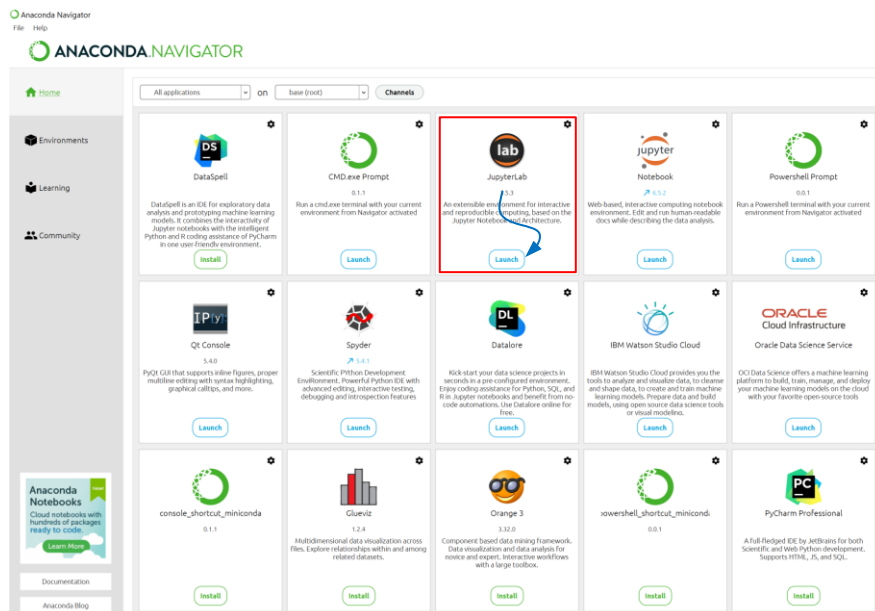


图 13. Anaconda Navigator 界面

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

b) 进入 JupyterLab 界面，点击 Notebook (Python 3)，创建 Jupyter Notebook。

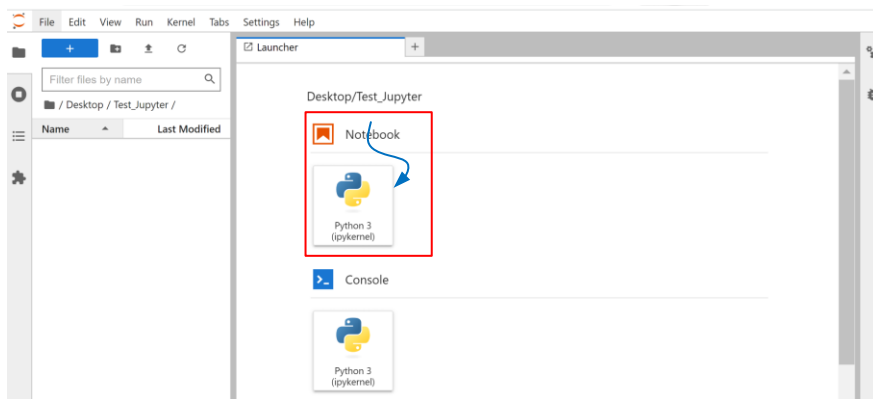


图 14. JupyterLab 界面

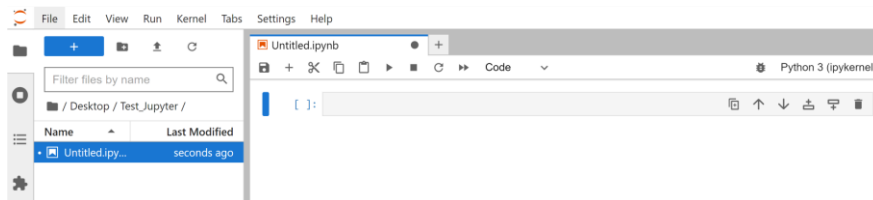


图 15. 创建 Jupyter Notebook

c) 在下面窗口中输入， $1 + 2$ ，然后点击“Ctrl + Enter”快捷键，运行并得到 3 这个结果。大家也可以尝试“Shift + Enter”快捷键，运行代码同时生成新区域，大家自己可以先玩一会。下一节将专门讲解如何使用 JupyterLab。

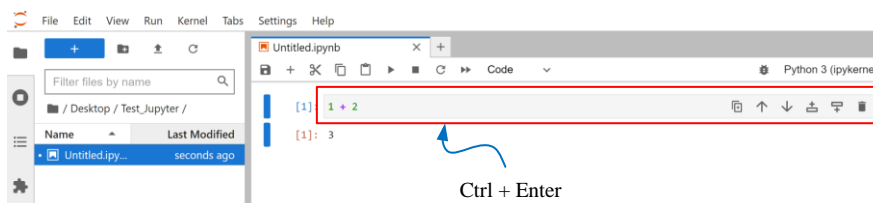


图 16. 运算



这一节的习题只需要大家完成 Anaconda 安装，并测试 JupyterLab。

* 这道题目很基础，本书不给答案。