# Python解释器和笔记，与GitHub代码托管

## 1. Python解释器和笔记

用代码来解决专业的问题时，代码只是解决问题的工具，而本质是发现解决问题的方法。“工欲善其事，必先利其器”。作为工具，除了Python本身内置的各类方法外，各类库所提供的方法更是数不胜数。为了方便库的安装和管理，就数据分析类，本书主要使用[Anaconda](https://www.anaconda.com/)环境管理器（Python包管理器）；当涉及用[Flask](https://flask.palletsprojects.com)①等扩展库构建Web应用时，则推荐使用[PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows)。

为方便交流，本书在阐述“基础实验”时，主要使用anaconda下的[JupyterLab](https://jupyterlab.readthedocs.io/en/stable/index.html)②，基于网页单元式的集成开发环境（Integrated development environment, IDE）。JupyterLab包括Code格式的的代码交互式解释器；[Markdown](https://www.markdownguide.org/)③格式的文本编辑器；和Raw格式保持原始字符的方式。集成有文本编辑器的代码交互式解释器，可以边书写代码，边记录笔记，不仅方便个人注释代码，同时方便代码分享，便于他人学习或查阅。在解决专项问题时，通常会包含大量代码，需要架构合理的文件结构，以包、子包和模块的方式管理代码，方便代码迁移和发布，因此在“专项研究”部分使用[Spyder](https://www.spyder-ide.org/)交互式解释器书写代码，而不再建议使用JupyterLab的模式。

书写代码时有时需要快速的浏览代码和数据，可以辅助使用[Notepad++](https://notepad-plus-plus.org/)文本编辑器。NotePad++支持众多各类编程语言语法，并能够高亮显示查看；同时支持多种编码，多国语言编写功能，及一些拓展了文本编辑能力的有用工具。

另一经常使用到的源码编辑器是[Visual Studio Code(VS Code)](https://code.visualstudio.com/)，支持代码调试、语法高亮、智能代码提示、代码重构和嵌入[Git](https://git-scm.com/download/win)④等功能。本书用[docsify](https://docsify.js.org/)⑤部署网页版时，使用VS Code实现。

每个人都会有自己做笔记的习惯，作代码的笔记，与我们常规使用Microsoft OneNote⑥等工具有所不同，要高亮显示代码格式，最好能运行查看结果，因此需要结合自身情况来选择笔记工具，使得学习代码这件事事半功倍。

| 序号 | 解释器名称 | 免费与否 | 推荐说明 | 官方网址 |
| --- | --- | --- | --- | --- |
| 1 | [python 官方](https://www.python.org/downloads/) | 免费 | **不推荐**，但轻便，可以安装，偶尔用于随手简单代码的验证 | <https://www.python.org> |
| 2 | [Anaconda](https://www.anaconda.com/) | 个人版(Individual Edition)免费；团队版(Team Edition)和企业版(Enterprise Edition)付费 | 集成了众多科学包及其依赖项，**强烈推荐**，因为自行解决库依赖项是件令人十分苦恼的事情。其中包含Spyder和JupyterLab为本书所使用 | <https://www.anaconda.com> |
| 3 | [Visual Studio Code(VS Code)](https://code.visualstudio.com/) | 免费 | **推荐使用**，用于查看代码非常方便，并支持多种语言格式 | <https://code.visualstudio.com> |
| 4 | [PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows) | 付费 | **一般推荐**，通常只在部署网页时使用，本书“实验用网络应用平台的部署部分”用到该平台 | <https://www.jetbrains.com/pycharm> |
| 5 | [Notepad++](https://notepad-plus-plus.org/) | 免费 | 仅用于查看代码、文本编辑，**推荐使用**，轻便的纯文本代码编辑器，可以用于代码查看，尤其兼容众多文件格式，当有些文件乱码时，不妨尝试用此编辑器 | <https://notepad-plus-plus.org> |

## 2. GitHub代码托管

本书书写过程中使用[GitHub](https://github.com/richieBao)托管代码和用VS Code书写Markdown说明文档。在本地和云端同步是直接应用GitHub提供的[GitHub Desktop](https://docs.github.com/en/desktop)⑦来推送（push）和拉取（pull）代码及相关文档。有时也会直接应用[StackEdit](https://stackedit.io/)⑧等工具书写推送Markdown文档到云端。亦可以直接在GitHub上在线编辑。

GitHub是一个使用Git进行软件开发和版本控制的互联网托管服务平台。它为每个项目提供Git的分布式版本控制及访问控制、错误跟踪、软件功能请求、任务管理、持续集成和维基（wikis）⑨。截至2022年6月，GitHub报告有8300万名开发者和超过2亿个代码仓库，包括至少2800万个公共代码仓库。

在学习本书或者建立自己的代码开发项目进行数据分析，开发应用（Application, App），布局网站等任何应用编程语言的代码工程，都强烈推荐使用GitHub代码托管平台，1是，方便不断增加的个人代码仓库的管理；2是，避免本地代码丢失；3是，方便云端多人或者团队协作；4是，可以配置代码仓库为私有，也可以配置为公共开源，易于代码分享传播；5是，可以应用GitHub网页功能，将个人的代码仓库发布为网页形式，通常用于代码仓库的说明等。







注释（Notes）：

① Flask，使用Python编写的轻量级Web应用框架（<https://flask.palletsprojects.com/>），因为不需要特定的工具或依赖库，也称为微框架（microframework）。

② JupyterLab，基于网页单元式的集成开发环境（<https://jupyterlab.readthedocs.io/en/stable/index.html>）。

③ Markdown，轻量级标记语言，支持图片、图表和数学公式等，且易读易写，广泛用于撰写帮助文档（<https://www.markdownguide.org/>）。

④ Git，是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到很大的项目版本管理（<https://git-scm.com/>）。VS Code集成了源码控制管理（Source control management, SCM），包括开箱即用的Git支持（<https://code.visualstudio.com/docs/sourcecontrol/overview>）。

⑤ docsify，通过智能的加载和解析Markdown文件，即时显示生成一个网站（<https://docsify.js.org>）。可以通过配置index.html文件，部署到GitHub页面上，发布网站。

⑥ Microsoft OneNote，为数字记录笔记，支持笔记、绘图、屏幕截图和音频等功能的记事程序。

⑦ GitHub Desktop，可以用图形用户界面（Graphical User Interface，GUI）而不是命令行或网页浏览器与GitHub互动。GitHub Desktop可以完成大多数Git命令，例如推送和拉取，克隆远程仓库，对更改内容以可视化确认等（<https://docs.github.com/en/desktop>）。

⑧ StackEdit，Markdown编辑器，可以将本地工作区的任何文件与云端的Google Drive、Dropbox和GitHub等账户中的文件同步。同步机制每分钟都会在后台进行，实现文件的下载、合并和上传修改文件等功能（<https://stackedit.io/>）。

⑨ wikis（维基），为超文本出版物，使用网络浏览器协作编辑和管理。一个典型的wiki包括项目主题、领域等多个页面，可以向公众开放，也可以仅限于在一个组织内部使用，以维护其知识库。