

第1章 Python 语言的开发环境安装

本章学习目标：了解 Python 语言的特点及编程风格；掌握 Python 开发环境的安装与配置，能根据开发需求完成库的导入与环境设置；掌握使用 Jupyter Notebook 编写和运行 Python 代码的方法；了解 AI 工具在 Python 编程中的辅助作用。

1.1 本章知识点

1. Python 语言特点

Python 语言是由荷兰科学家 Guido van Rossum 设计的解释型高级编程语言。Python 语言的核心特性有跨平台性、接近自然语言的简洁语法、丰富的第三方库以及支持面向对象编程。正是这些特性使得 Python 学习门槛较低，并在数据分析、人工智能等领域获得了极为广泛的应用。

通常 Python 语言开发环境由 3 部分组成：Python 语言解释器、包管理工具与集成开发环境。

2. Python 安装与运行

一个完整的 Python 开发环境通常包含 Python 解释器和一款集成开发环境。安装与运行的步骤如图 1-1 所示。需要先下载并安装 Python，再下载并安装 PyCharm；编写 Python 语言程序前需要配置 PyCharm。

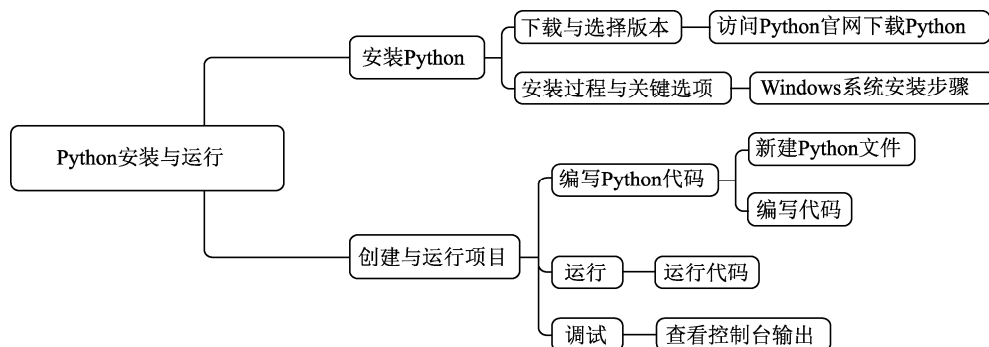


图 1-1 Python 开发环境与配置

从官网下载并保存最新版本的 Python 语言安装包文件，如版本文件 python-3.13.7-amd64.exe，双击安装包文件；如图 1-2 所示，选择默认安装（也可以选择用户设置安装），并等待安装成功。



图 1-2 Python 安装界面

3. Anaconda 的安装与配置

Anaconda 是一种开源的数据科学平台，集成了 Python 语言解释器、科学计算库（NumPy、Pandas、Matplotlib 等）、Jupyter Notebook 开发工具，以及用于包与环境管理的核心工具 Conda 等。Anaconda 安装步骤如图 1-3 所示。

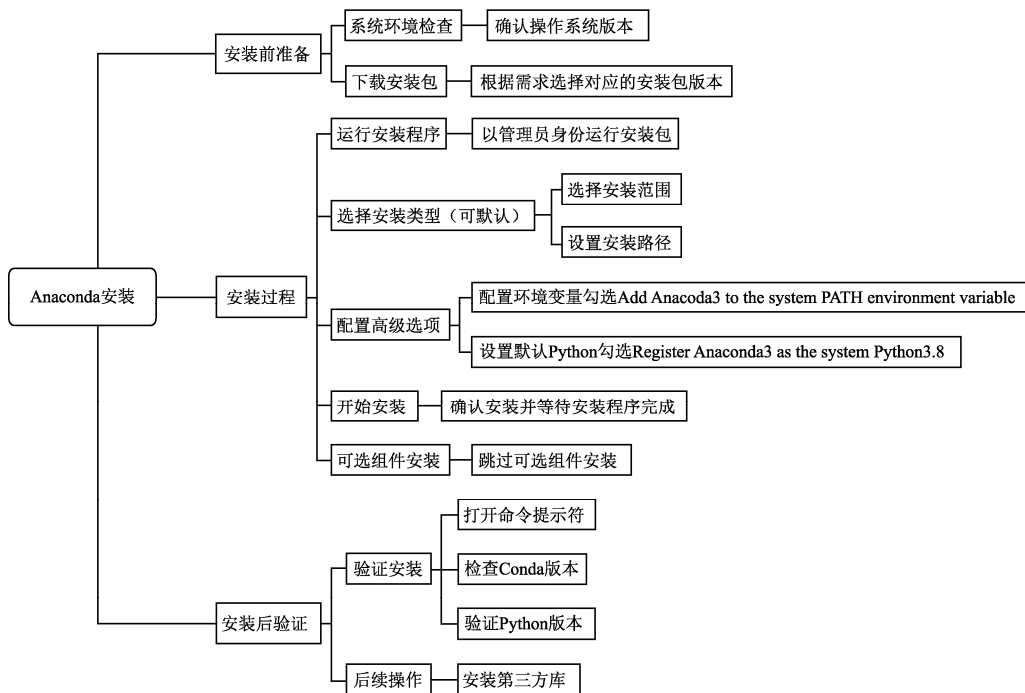


图 1-3 Anaconda 安装步骤

(1) 查看系统并下载安装文件。

如图 1-4 所示，本书对应的系统为 Windows 10 的 64 位系统，采用安装文件为 Anaconda3-202X.XX-Windows-x86_64.exe。读者可通过国内镜像下载并保存对应版本 Anaconda 安装文

件，链接地址为 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>。

Anaconda3-2021.05-Linux-aarch64.sh	412.6 MiB	2025-02-14 03:28
Anaconda3-2021.05-Linux-ppc64le.sh	285.3 MiB	2025-02-14 03:28
Anaconda3-2021.05-Linux-s390x.sh	291.7 MiB	2025-02-14 03:28
Anaconda3-2021.05-Linux-x86_64.sh	544.4 MiB	2025-02-14 03:28
Anaconda3-2021.05-MacOSX-x86_64.pkg	440.3 MiB	2025-02-14 03:28
Anaconda3-2021.05-MacOSX-x86_64.sh	432.7 MiB	2025-02-14 03:28
Anaconda3-2021.05-Windows-x86.exe	408.5 MiB	2025-02-14 03:28
Anaconda3-2021.05-Windows-x86_64.exe	477.2 MiB	2025-02-14 03:28
Anaconda3-2021.11-Linux-aarch64.sh	487.7 MiB	2025-02-14 03:28
Anaconda3-2021.11-Linux-ppc64le.sh	254.9 MiB	2025-02-14 03:28
Anaconda3-2021.11-Linux-s390x.sh	241.7 MiB	2025-02-14 03:28

图 1-4 Anaconda3 安装文件下载

(2) Anaconda 安装。

选中下载好的安装文件，右击后选择“以管理员身份运行”命令。若弹出系统安全提示，单击“运行”按钮。安装程序启动后，按以下步骤完成安装：首先单击 Next 按钮；然后在许可协议界面单击 I Agree 按钮；接着在用户选择界面选择 All Users，再单击 Next 按钮；随后在安装路径设置界面选择默认路径 C:\ProgramData\Anaconda3，单击 Next 按钮；在安装选项界面勾选相关配置项后，单击 Install 按钮开始安装（见图 1-5）；安装完成后，依次单击 Next 按钮，根据需要选择是否添加 PyCharm 工具，最后单击 Finish 按钮完成安装。

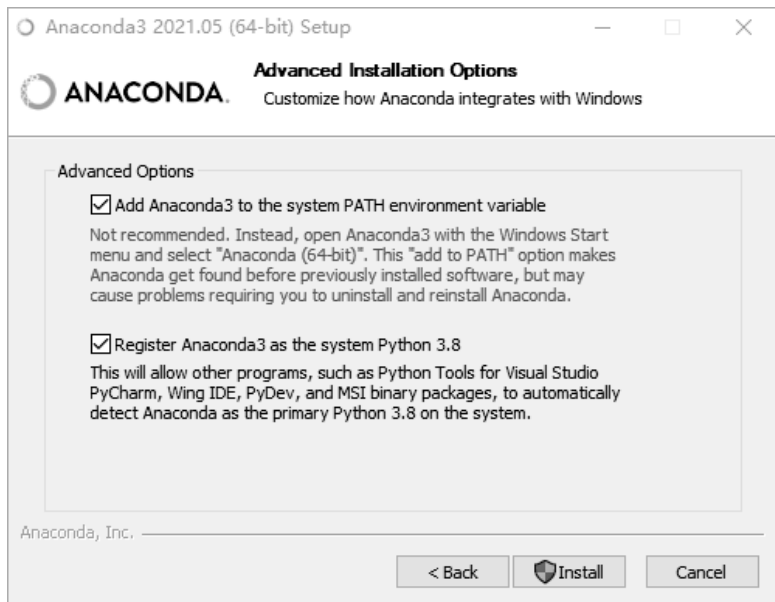


图 1-5 软件安装配置系统选项

(3) Anaconda 安装验证与库安装。

单击系统“窗口”后，找到 Anaconda3，找到子项“Anaconda Prompt（或 Powershell）”，右击，依次选择“更多”→“以管理员身份运行”，弹出命令窗口，如图 1-6 和图 1-7 所示。



图 1-6 启动 Anaconda Prompt 命令窗口

输入命令 Python，可以查看 Anaconda3 集成的 Python 版本并以命令模式进入程序编辑环境，可按 Ctrl+Z 键退回命令窗口。

输入命令 pip list，查看 Anaconda3 已安装库与对应版本，如图 1-8 所示。

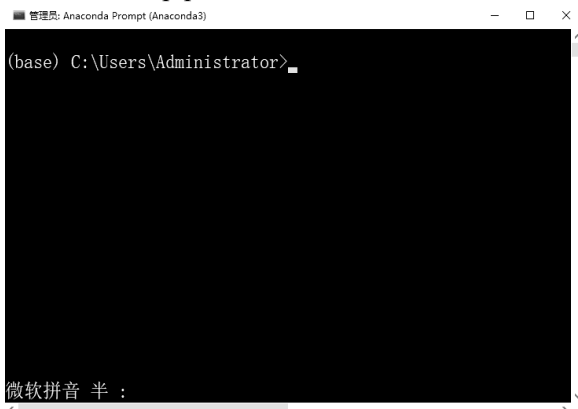


图 1-7 Anaconda Prompt 命令窗口

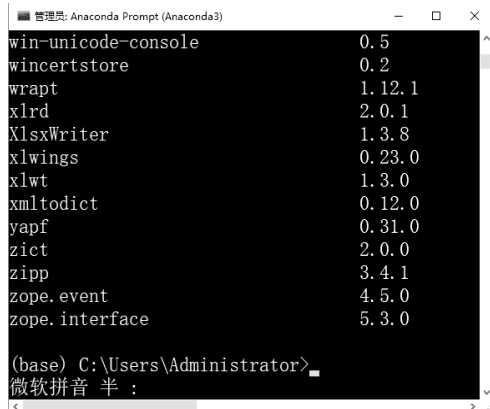


图 1-8 Anaconda 已安装库

输入命令“pip install 库名==版本号”，安装对应版本的库。表 1-1 所示为本书后续章节需要使用到的重要库。

表 1-1 重要库

库名称	安装命令	用途
NumPy	pip install numpy == 1.20.1	科学计算
Pandas	pip install pandas== 1.2.4	数据分析
Matplotlib	pip install matplotlib== 3.3.4	数据可视化
Jieba	pip install jieba== 0.42.1	分词(中文)
Wordcloud	pip install wordcloud== 1.8.1	词云图
Scikit-learn	pip install scikit-learn== 0.24.1	机器学习
TensorFlow	pip install tensorflow== 1.20.1	深度学习

1.2 基础实验

实验 1.1 Python IDLE——“Hello,World!” 程序

实验目的：了解 Python 自带的集成开发环境（IDLE），掌握使用 Python IDLE 环境进行代码编写与运行的基本方法。

实验内容：IDLE 作为 Python 自带的集成开发环境，适合初学者快速测试代码片段；使用 Python IDLE 输出“Hello,World!”程序。

实验步骤：

Step1: 启动 Python IDLE。

启动 Python 的 IDLE，如图 1-9 与图 1-10 所示。



图 1-9 启动 IDLE

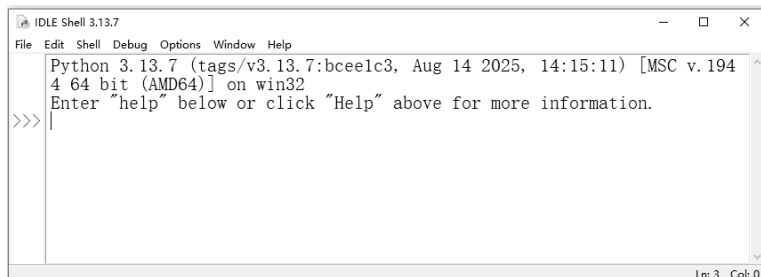


图 1-10 进入 IDLE Shell 模式

Step2: 输入代码，运行并查看输出结果，如图 1-11 所示。

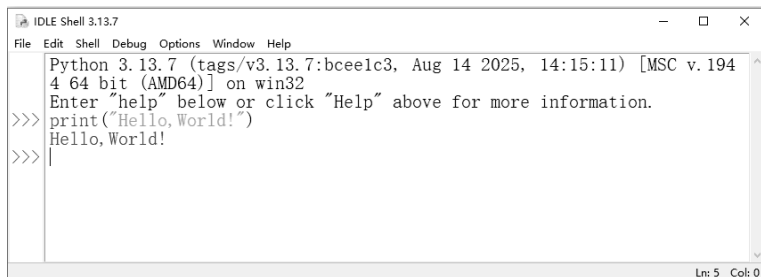


图 1-11 代码输入与运行

参考代码：

```
print("Hello,World!") #输出指定的字符串至控制台
```

思考题：新建 py 程序文件并运行。

实验 1.2 Jupyter Notebook 开发环境——“Hello,World!” 程序

实验目的：了解 Jupyter Notebook 开发环境。

实验内容：使用 Jupyter Notebook 编写“Hello,World!”程序。

实验步骤：

Step1：启动 Jupyter Notebook，如图 1-12 所示；图 1-13 所示为启动本地服务器，不可关闭；图 1-14 所示为启动浏览器并新建 Python3 程序。

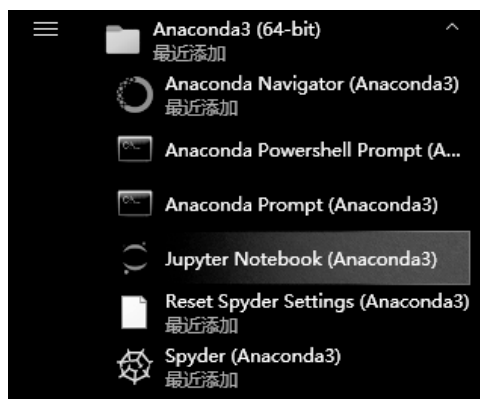


图 1-12 启动 Jupyter Notebook

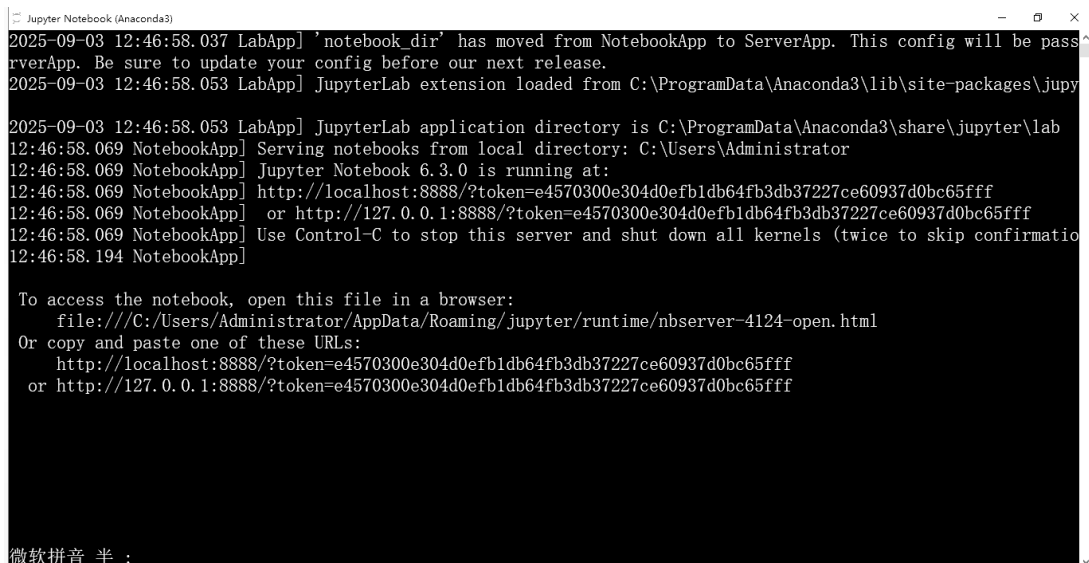


图 1-13 启动本地服务器

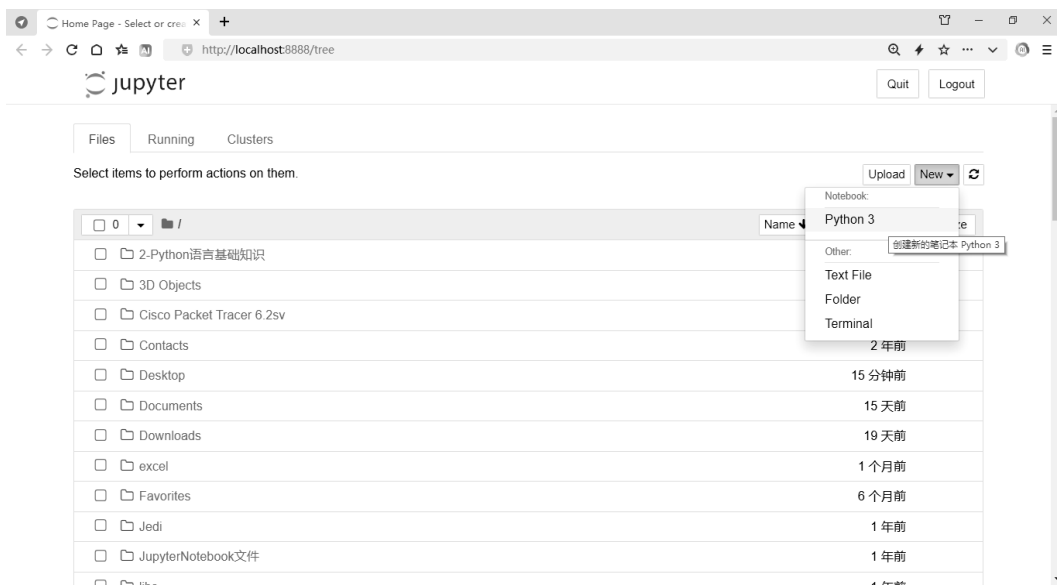


图 1-14 启动浏览器并新建 Python3 程序

Step2: 输入代码，并运行。输出结果如图 1-15 所示。

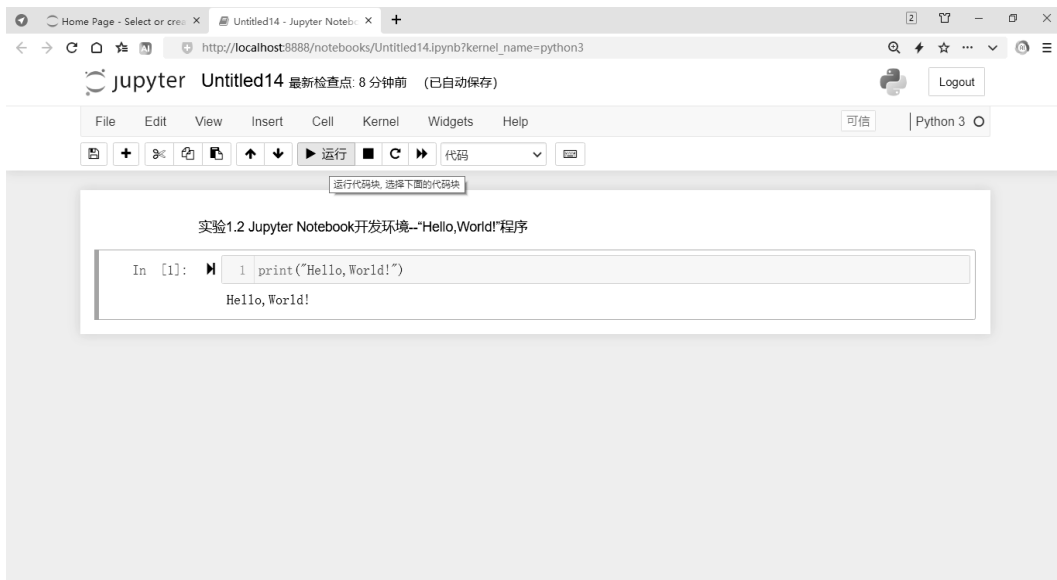


图 1-15 代码与输出结果

参考代码:

```
print("Hello,World!") #输出字符串
```

思考题 使用将 Jupyter Notebook 脚本保存为 Python 程序（py 文件）。

1.3 进阶实验

实验 1.3 AI 生成代码与代码分析

实验目的：探索 AI 工具在编程过程中的辅助作用及应用效果。

实验内容：使用 AI 工具实现代码生成与代码运行。

Step1：启动 AI 大模型，输入提示语。如“我是学习 Python 语言的新手，生成一行代码，实现输出"Hello,World!"。”，如图 1-16 所示。



图 1-16 AI 大模型输入提示语

Step2：运行代码，输出结果如图 1-17 所示。



图 1-17 参考大模型生成代码与运行结果

1.4 本章小结

Python 是跨平台、易学习的解释型高级语言，适合数据分析、AI 等领域；Anaconda 是集成数据科学工具的平台与强大的包管理器；Jupyter Notebook 是一个基于 Web 的交互式编程与入门学习的编程工具。

通过本章学习，学生将掌握 Anaconda 集成开发环境的安装与配置，初步学会使用 Jupyter Notebook 的交互式编程环境编写与运行 Python 程序。

1.5 练习题

1. 尝试至少两种不同的方式来编写输出“Hi, AI!”的 Python 语言程序。
2. 了解 Pycharm、VS Code 等集成开发环境，对比它们与 Jupyter Notebook 开发环境的区别。
3. 启动 Anaconda Navigator，了解虚拟环境以及如何管理。