

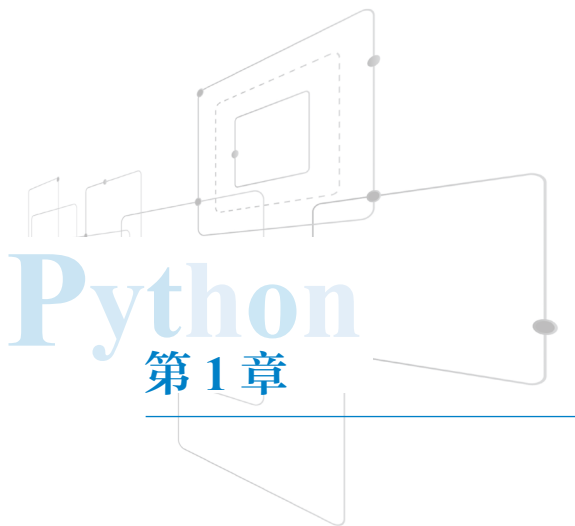
# Python

第一部分

编程基础

---





# Python 语言概述

## 1.1 Python 语言与数据分析应用

### 1.1.1 Python 语言发展及特点

人类要和计算机进行“沟通与对话”，即用指令操作计算机执行一系列动作，必须使用计算机能“听懂和理解”的一种特别语言。就像中国人要和英国人沟通，可以采用英语一样，人类与计算机交流也是使用一种特别设计的有着自身语法规则的语言：计算机程序设计语言。正如全世界的语言种类非常多，与计算机沟通的计算机程序设计语言也非常多，如 C 语言、Java 语言，当然还有本书即将介绍的 Python 语言。

Python 语言比较类似英语，人类完全能够看懂它，而且还能自如地使用 Python 语言形成复杂的程序和软件，对计算机各种操作施行精确指挥。但是计算机是无法直接理解英语的，当然也无法直接理解类似英语的 Python 语言。事实上，计算机都无法直接理解所有的程序设计语言。

一般通过一个翻译来将程序设计语言（包括 Python 语言、C 语言、Java 语言等）自动翻译成计算机能理解的语言，而这个翻译称为编译器或解释器（图 1-1）。



图 1-1 Python 必须有一个解释器才能运行 Python 程序

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。Python 的设计具有很强的可读性，相比其他语言它具有更有特色的语法结构。

Python 是一种解释型语言：这意味着开发过程中没有编译这个环节。类似于 PHP（Hypertext Preprocessor，超文本预处理器）和 Perl 语言。

Python 是交互式语言：这意味着使用者可以在一个 Python 提示符 `>>>` 后直接执行代码。

Python 是面向对象语言：这意味着 Python 支持面向对象的风格或代码用对象封装的编程技术。

Python 是初学者的语言：Python 对初级程序员而言，是一种伟大的语言，它支持广泛的应用程序开发，这些应用程序既可以是简单的文字处理程序，又可以是 WWW 浏览器，还可以是游戏等。

### 1. Python 的发展历程

Python 之父——荷兰人 Guido van Rossum，于 1982 年从阿姆斯特丹大学取得了数学和计算机硕士学位。

20 世纪 80 年代中期，Guido van Rossum 还在 CWI（数学和理论计算机科学领域的研究中心）为 ABC 语言贡献代码。ABC 语言是一个为编程初学者打造的研究项目。ABC 语言给了 Guido 很大的影响，Python 从 ABC 语言中继承了很多东西，如字符串、列表和字节数列都支持索引、切片排序和拼接操作。

在 CWI 工作了一段时间后，Guido 构思了一门致力于解决问题的编程语言，他觉得现有的编程语言对非计算机专业的人十分不友好。于是，1989 年 12 月，为了打发无聊的圣诞节假期，Guido 开始编写 Python 的第一个版本。

1991 年，Python 的第一个解释器诞生了，它是由 C 语言实现的，有很多语法来自 C 语言，又受到了 ABC 语言很多的影响。Python 1.0 版本于 1994 年 1 月发布，这个版本的主要功能是 lambda、map、filter 和 reduce，但是 Guido 并不喜欢这个版本。6 年半之后的 2000 年 10 月，Python 2.0 发布了。这个版本的主要新功能是内存管理和循环检测垃圾收集器以及对 Unicode 的支持。然而，尤为重要的变化是开发流程的改变，Python 此时有了一个更透明的社区。2008 年 12 月，Python 3.0 发布了。Python 3.x 不向后兼容 Python 2.x，这意味着 Python 3.x 可能无法运行 Python 2.x 的代码，并且 Python 2 已于 2020 年停止官方支持。

现在，Python 的社区变得更加庞大和活跃。越来越多的人开始使用 Python，并且贡献了大量的代码和库。Python 的生态系统变得更加丰富和多样化，包括各种框架、库和工具。其中一些最受欢迎的框架包括 TensorFlow、PyTorch 和 Keras。在过去的几十年里，Python 已经成为一种广泛使用的编程语言，它的应用领域不断扩大，社区变得越来越庞大和活跃。Python 的未来充满着无限的可能性，我们可以期待更多的创新和发展。

### 2. Python 的特点

Python 秉承“优雅、明确、简单”的设计理念，具有以下特点。

(1) 简单易学。Python 是一种代表简单主义思想的语言，读者在开发 Python 程序时，可以专注于解决问题本身而不用顾虑语法的细枝末节。Python 极其容易上手，因为 Python 有极其简单的说明文档。

(2) 免费、开源。Python 是 FLOSS（自由、开放源代码软件）之一，使用者可以自由地发布软件的备份，阅读它的源代码，对它的源代码进行改动，或者将其中一部分代码用于新的自由软件中。且用户使用 Python 进行开发或者发布自己的程序，不需要支付任何费用。

(3) 高级语言。Python 封装较深，屏蔽了很多底层细节，如 Python 会自动管理内存（需要时自动分配，不需要时自动释放），因而用 Python 编写程序时无须考虑如何管理程序使用的内存一类的底层细节。

(4) 面向对象。Python 既支持面向过程的编程，也支持面向对象的编程。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。

(5) 可扩展性、可扩充性和可嵌入性。如果需要一段关键代码运行得更快或者希望某些算法不公开，可以部分程序用 C 或 C++ 编写，然后在 Python 中调用它们。同时也可以把 Python 代码嵌入 C 或 C++ 程序，从而向程序用户提供脚本功能。Python 能把其他语言“粘”在一起，所以被称为“胶水语言”。

(6) 解释性。Python 语言写的程序不需要编译成二进制代码，可以直接从源代码运行程序。在计算机内部，Python 解释器把源代码转换成字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行，这使得使用 Python 更加简单，也使得 Python 程序更加易于移植。

(7) 可移植性。由于它的开源特点，Python 能够运行在不同的平台上。绝大多数 Python 程序无须修改便可以在众多平台上运行，这些平台包括 Linux、Windows、QNX、VMS 以及 Google 基于 Linux 开发的 Android 平台等。

(8) 丰富的库。Python 拥有许多功能丰富的库，其标准库非常庞大，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、HTML、WAV 文件、密码系统、GUI（图形用户界面）、Tk 等，这就是 Python 被誉为“功能齐全”的原因。此外，还提供了许多其他高质量的库，如 wxPython、Twisted 和 Python 图像库等。

### 1.1.2 大数据分析及应用

在当今时代，大数据已经成为许多企业决策的重要依据。为了更好地利用这些数据，许多企业的 IT 部门都开始使用各种工具和技术来处理和分析大数据。

大数据分析是指用适当的统计分析方法对收集来的大量数据进行统计分析，提取有用信息并形成结论，进而对数据加以详细研究和概括总结的过程。在实际应用中，数据分析

可帮助人们和企业做出判断，以便采取适当行动。而 Python 语言是一种强大的工具，能够在大数据分析过程中充分发挥优势。Python 语言的可读性较强，且易于学习和使用，它也是数据分析领域中最受欢迎的编程语言之一。

### 1. 数据预处理

数据预处理是数据整理的先前步骤，是指对所收集数据进行分类或分组前所做的审核、筛选、排序等必要的处理。现实世界中的大规模数据往往都是杂乱、不完整、不一致、有噪声、冗余的，无法直接进行数据分析，或分析结果不尽如人意。为了提高数据分析的质量，数据预处理技术应运而生。

数据预处理有多种方法，如数据清理、数据集成、数据变换、数据归约等。这些数据处理技术在数据挖掘之前使用，可大大提高数据分析的质量，降低实际分析所需要的时间。

### 2. 数据分析方法

完整的数据分析主要包括六大步骤，依次为分析设计、数据收集、数据处理、数据分析、数据展现、报告撰写等，所以也叫数据分析六部曲。而对数据进行分析是数据分析的核心过程，下面介绍 4 种常用的统计分析方法。

#### 1) 描述性统计分析

描述性统计是指运用制表和分类、图形以及计算概括性数据来描述数据特征的各项活动。描述性统计分析要对调查总体所有变量的有关数据进行统计性描述，主要包括数据的频数分析、集中趋势分析、离散程度分析、分布以及一些基本的统计图形。

#### 2) 回归分析

回归分析是一种用于研究变量之间关系的统计方法。在金融市场中，回归分析可以用来研究股票价格和其他变量之间的关系，例如，利率通货膨胀率和公司盈利等。通过回归分析，投资者可以预测股票价格的变化，并做出相应的投资决策。

#### 3) 相关分析

相关分析就是对总体中确实具有联系的标志进行分析，其主体是对总体中具有因果关系标志的分析。它是描述客观事物相互间关系的密切程度并用适当的统计指标表示出来的过程。在一段时期内，出生率随经济水平上升而上升，这说明两指标间是正相关关系；而在另一时期，随着经济水平进一步发展，出现出生率下降的现象，两指标间就是负相关关系。

相关分析与回归分析之间的区别：回归分析侧重于研究随机变量间的依赖关系，以便用一个变量去预测另一个变量；相关分析侧重于发现随机变量间的种种相关特性。

#### 4) 时间序列分析

时间序列分析是一种用于研究时间序列数据的统计方法。在金融市场中，时间序列分析可以用来研究股票价格、汇率和利率等变量随时间的变化趋势。通过时间序列分析，投资者可以预测未来的市场趋势，并做出相应的投资决策。

### 1.1.3 Python 程序库简介

#### 1. 基本概念

包、库、模块是 Python 中常用的概念。一般情况下，模块指一个包含若干函数定义、类定义或常量的 Python 源程序文件，库或包指包含若干模块并且其中一个文件名为 `_init_.py` 的文件夹。对于包含完整功能代码的单个模块也可以叫作库，例如，标准库 `re` 和 `re` 模块这两种说法都可以。但一般不把库叫作模块，例如，`tkinter` 库包含若干模块文件，此时一般说标准库 `tkinter` 而不是 `tkinter` 模块。

在 Python 中，有内置模块、标准库和扩展库之分。内置模块和标准库是 Python 官方的标准安装包自带的，内置模块没有对应的文件，可以认为是封装在 Python 解释器主程序中的；标准库对应的程序文件在 Python 安装路径中的 `Lib` 文件夹中。

Python 官方的标准安装包自带了 `math`（数学模块）、`random`（随机模块）、`datetime`（日期时间模块）、`collections`（包含更多扩展版本序列的模块）、`functools`（与函数以及函数式编程有关的模块）、`urllib`（与网页内容读取以及网页地址解析有关的模块）、`itertools`（与序列迭代有关的模块）、`string`（字符串操作）、`re`（正则表达式模块）、`os`（系统编程模块）、`os.path`（与文件、文件夹有关的模块）、`zlib`（数据压缩模块）、`hashlib`（安全散列与报文摘要模块）、`socket`（套接字编程模块）、`tkinter`（GUI 编程模块）、`sqlite3`（操作 SQLite 数据库的模块）、`csv`（读写 CSV 文件的模块）、`json`（读写 JSON 文件的模块）、`pickle`（数据序列化与反序列化的模块）、`statistics`（统计模块）、`time`（与时间操作有关的模块）等大量内置模块和标准库（完整清单可以通过官方在线帮助文档 <https://docs.python.org/3/library/index.html> 进行查看），但没有集成任何扩展库，程序员可以根据实际需要再安装第三方扩展库。

截至 2023 年 6 月，`pypi` 已经收录了超过 46 万个扩展库项目，涉及很多领域的应用，例如，`jieba`（用于中文分词）、`moviepy`（用于编辑视频文件）、`xlrd`（用于读取 Excel 2003 之前版本文件）、`xlwt`（用于写入 Excel 2003 之前版本文件）、`openpyxl`（用于读写 Excel 2007 及更新版本文件）、`python-docx`（用于读写 Word 2007 及更新版本文件）、`python-pptx`（用于读写 PowerPoint 2007 及更新版本文件）、`pymupdf`（用于操作 PDF 文件）、`pymssql`（用于操作 Microsoft SQL Server 数据库）、`pypinyin`（用于处理中文拼音）、`pillow`（用于数字图像处理）、`pyopengl`（用于计算机图形学编程）、`NumPy`（用于数组计算与矩阵计算）、`SciPy`（用于科学计算）、`pandas`（用于数据分析与处理）、`Matplotlib`（用于数据可视化或科学计算可视化）、`requests`（用于实现网络爬虫功能）、`beautifulsoup4`（用于解析网页源代码）、`scrapy`（爬虫框架）、`sklearn`（用于机器学习）、`PyTorch`、`TensorFlow`（用于深度学习）、`flask`、`Django`（用于网站开发）等几乎渗透到所有领域的扩展库或第三方库。

## 2. 库的安装与管理

Python 有两个基本的库管理工具，即 `easy_install` 和 `pip`。目前大部分用户都采用 `pip` 进行对扩展库的查看、安装与卸载。下面介绍 `pip` 命令几个常用的方法。

### 1) 查看扩展库

```
cmd>pip list
```

例如: D:\Python 311\Scripts>pip list

### 2) 查看当前安装的库

```
cmd>pip show Package
```

例如: D:\Python 311\Scripts>pip show numpy

### 3) 安装指定版本的扩展库

```
cmd>pip install Package == 版本号
```

例如: D:\Python 311\Scripts>pip install tensorflow == 1.9.0

### 4) 离线安装扩展库文件 whl

```
cmd>pip install Package.whl
```

例如: D:\Python 311\Scripts>pip install pyshp-2.1.3-py3-none-any.whl

### 5) 卸载扩展库

```
cmd>pip uninstall Package
```

例如: D:\Python 311\Scripts>pip uninstall numpy

### 6) 更新扩展库

```
cmd>pip install-U Package
```

例如: D:\Python 311\Scripts>pip install-U tensorflow

**注:** U 为大写字母。

## 1.1.4 实训任务: 安装 package

### 例 1: 库的安装。

打开命令提示符 `cmd` 窗口 (按 Windows+R 组合键后输入 “`cmd`” 并按 Enter 键, 如图 1-2 所示), 使用 `cd` 命令进入安装 Python 的 `scripts` 文件夹中 (例如, Python 安装的路径为 `D:\python311`), 输入 “`pip install Package`” (比如 `numpy`), 按 Enter 键进行安装即可。如图 1-3 所示, 表明此库安装成功 (也可以在打开命令提示符 `cmd` 窗口后, 直接输入 “`pip install Package`” 进行安装)。程序运行效果可以扫码观看。



由于默认 `pip` 获取的是 Python 官方源, 经常下载较慢甚至不可用, 这时可以使用国内 Python 镜像源。下面列出一些常用的国内镜像源。

清华大学: <https://pypi.tuna.tsinghua.edu.cn/simple>

阿里云: <http://mirrors.aliyun.com/pypi/simple/>

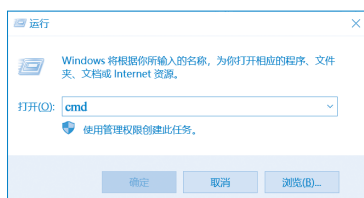


图 1-2 运行 cmd

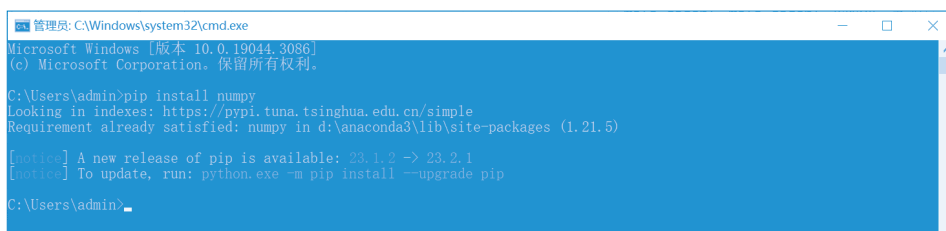


图 1-3 运行安装 numpy

中国科学技术大学：<https://pypi.mirrors.ustc.edu.cn/simple/>

华中理工大学：<http://pypi.hustunique.com/>

山东理工大学：<http://pypi.sdutlinux.org/>

豆瓣：<http://pypi.douban.com/simple/>

使用时只需要在命令提示符后面输入“pip install -i <https://pypi.tuna.tsinghua.edu.cn/> simple numpy”（如使用清华大学镜像源）就可以成功安装扩展库 numpy。

**例 2：**永久更换安装源（以 Windows 为例）。

步骤如下。

(1) 单击“此电脑”，在最上面的文件夹窗口中输入“%APPDATA%”，如图 1-4 所示。

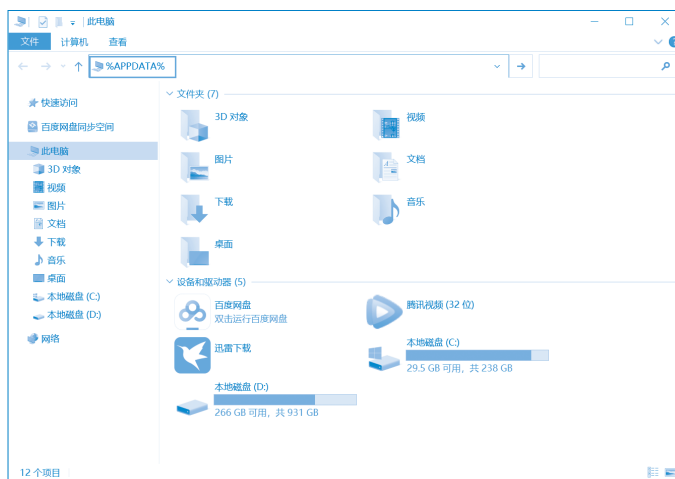


图 1-4 在文件夹窗口输入“%APPDATA%”

(2) 回车跳转到目录 C:\...\admin\AppData\Roaming，新建 pip 文件夹 C:\Users\admin\AppData\Roaming\pip。

(3) 创建 pip.ini 文件（注意扩展名）。

(4) 打开文件输入以下内容，保存并关闭即可，如图 1-5 所示。

```
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
```

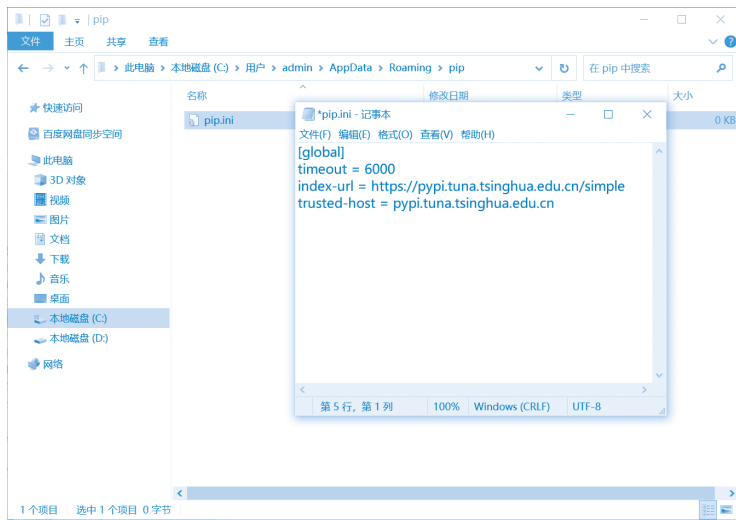


图 1-5 创建 pip.ini 文件

## 1.2 Python 程序开发与开发环境

### 1.2.1 Python 程序开发与运行

Python 是一种解释型的脚本编程语言，这样的编程语言一般支持以下两种代码运行方式。

#### 1. 交互式

在命令提示符 cmd 窗口中输入 Python 命令，看到 >>> 提示符（出现“>>>”即进入 Python 解释器）就可以开始输入代码了，交互式代码如下。程序运行效果可以扫码观看。



```
>>>print('Hello Python!')
Hello Python!
>>>a=2
>>>b=4
>>>a+b
6
```

## 2. 文件式

Python 源文件是一种纯文本文件，内部没有任何特殊格式，可以使用任何文本编辑器打开。例如，Windows 下的记事本程序，Linux 下的 Vim、gedit 等，macOS 下的 TextEdit 工具。

使用编辑器创建一个源文件，并输入下面的代码，保存为 demo.py。

```
print('Hello Python!')
a=2
b=4
print(a+b)
```

进入命令提示符 cmd 窗口，首先切换到 demo.py 所在的目录，然后输入下面的命令就可以运行源文件了。程序运行效果可以扫码观看。



```
cd /d D:\pythonproject # 切换到 demo.py 所在的目录
python demo.py         # 运行代码
```

也可以在命令提示符 cmd 窗口中用另外一种方法运行 Python 程序，输入“Python+ 程序地址 + 程序名 .py”，代码如下。程序运行效果可以扫码观看。



```
python D:\pythonproject\demo.py
```

运行完该命令，可以立即看到输出结果，如下所示。

```
Hello Python!
6
```

### 1.2.2 集成开发环境

解释器只是解决了翻译问题，还不能很方便地编写 Python 程序（图 1-6）；仅有编译器的确已经可以通过“命令行”（command line）来编写 Python 程序了，但是显然非常不方便。

```
管理员: C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 10.0.19044.3086]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\admin>python
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello python!')
hello python!
>>>
```

图 1-6 仅有编译器的 Python 命令行开发

因此还需要有一个非常容易使用的编写程序的开发环境，这种开发环境被称为集成开发环境（Integrated Development Environment, IDE）。IDE 是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具。IDE 是

集成了代码编写功能、分析功能、编译功能、调试功能等一体化的软件开发服务套件（图 1-7）。

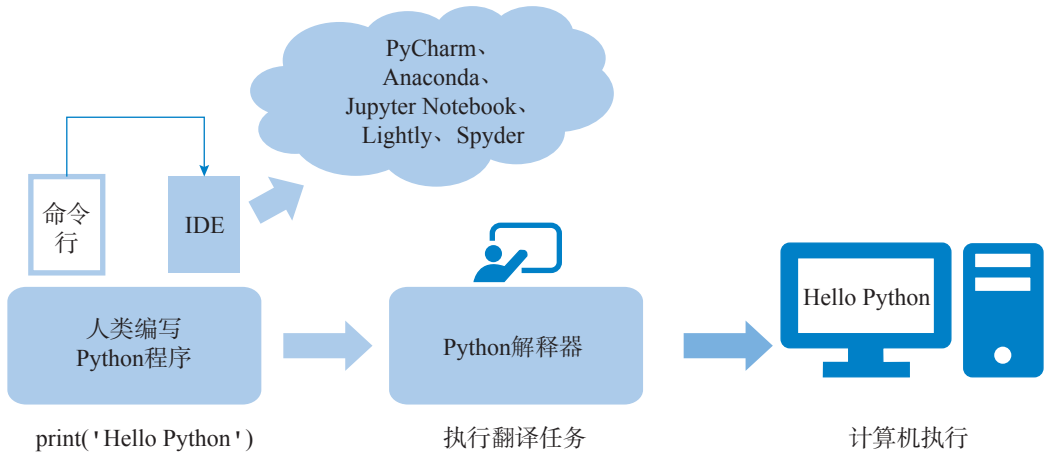


图 1-7 采用 IDE 的 Python 程序开发

Python 是一种开源、免费的脚本语言，它并没有提供一个官方的开发环境（仅自带了一个 IDLE），需要用户自主来选择编辑工具。目前，Python 的开发环境有很多种，可选择的 IDE 类别很多，既可以选用 Python 自带的 IDLE，也可以选择使用 PyCharm、Anaconda、Jupyter Notebook、Lightly、Spyder 等作为 IDE。

其中，IDLE 是 Python 内置的集成开发环境，它由 Python 安装包提供，也就是 Python 自带的文本编辑器。IDLE 为开发人员提供了许多有用的功能，如自动缩进、语法高亮显示、单词自动完成以及命令历史等，在这些功能的帮助下，用户能够有效地提高开发效率。更多的集成开发环境将会在 1.3 节中进行介绍。

## 【小知识】

### IDE 与 IDLE 的区别

IDLE 是 Python 内置的集成开发环境，它由 Python 安装包提供。而 IDE 一般由第三方公司或者组织提供，如 PyCharm、Anaconda、Jupyter Notebook、Lightly、Spyder 等。事实上，Anaconda 不仅提供 IDE，它更是一个开源的 Python 集成发行版本，包含 conda、Python 等 180 多个科学包及其依赖项。

IDLE 全称是“Integrated Development and Learning Environment”，是 Python 的集成开发和学习环境。它被打包为 Python 包装的可选部分，当安装好 Python 以后，IDLE 就自动安装好了，不需要另外去安装。使用 IDLE 运行 Python 程序的方法如下。

- (1) 从 Python 官网 <https://www.python.org/> 下载安装 Python 解释器，如图 1-8 所示。
- (2) 单击 Windows “菜单”按钮，如图 1-9 中数字 1 所示；然后从菜单中找到 IDLE（Python 3.10 64-bit）选项即可。

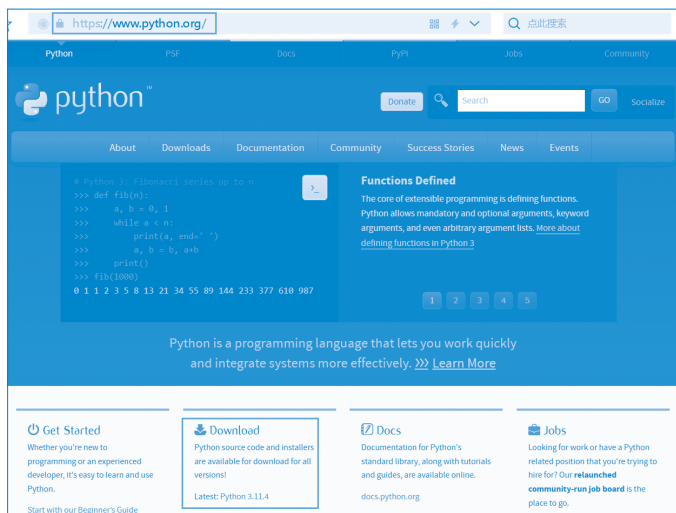


图 1-8 官网下载 Python 解释器

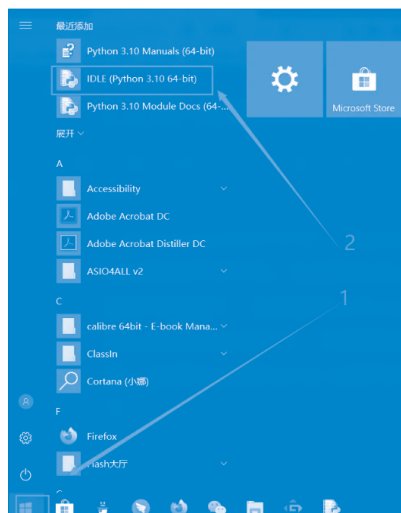


图 1-9 Python IDLE

### 1.2.3 实训任务：编写简单程序

**例 3：**分别用交互式 and 文件式两种方法，编辑并运行 print 语句。程序运行效果可以扫码观看。



交互式代码如下。

```
>>>print('Hello Python!')
Hello Python!
>>>print('Hello World!')
Hello World!
>>>print(1+2)
3
>>>print(2-1)
1
```

文件式代码如下。

```
print('Hello Python!')
print('Hello World!')
print(1+2)           # 运用算术运算符：加号
print(2-1)          # 运用算术运算符：减号
```

**注意：**在 Python 中，以 # 开头的信息称为注释，是用户在代码中加入的说明信息，不被解释器解释与执行。

**例 4：**采用 IDLE 编写上述代码。

- (1) 进入 Python 安装目录，运行 IDLE，或者直接使用命令行运行（图 1-10）。
- (2) 使用 File 菜单中的 New File 选项来创建一个新的 Python 文件（图 1-11）。我们可以在新文件中编写 Python 代码，然后保存文件。保存好文件后再次



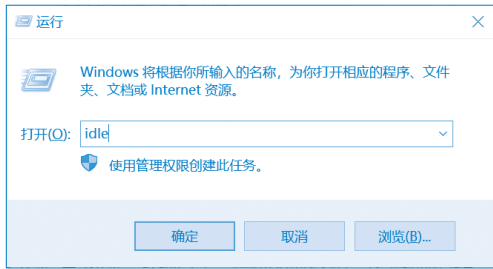


图 1-10 运行 IDLE

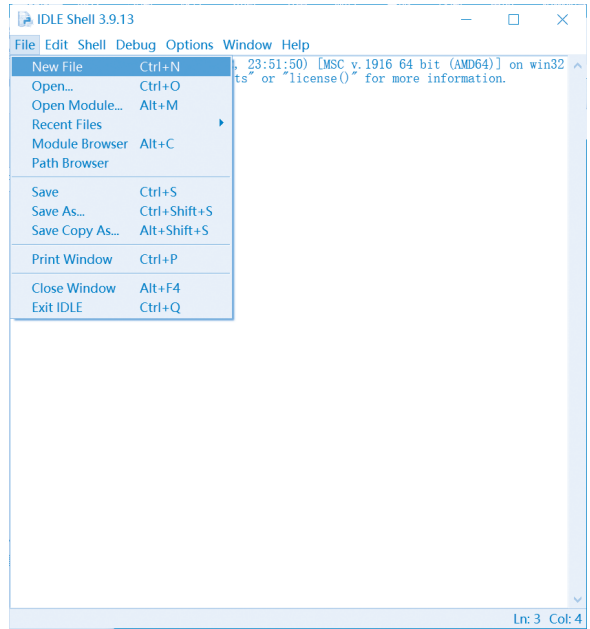


图 1-11 新建一个 Python 文件

使用 Open 命令打开，可以使用 Run 菜单中的 Run Module 选项来运行 Python 程序。这将打开 Python 解释器，并执行编写的代码。如果代码中有任何错误，解释器将会输出错误信息（图 1-12）。

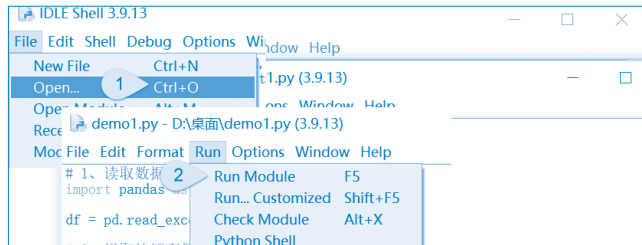


图 1-12 IDLE 打开 .py 文件并运行 Python 程序

## 1.3 Anaconda 与 Python

### 1.3.1 Anaconda

Anaconda 是一个第三方、开源、免费的工具，支持 800 多个第三方库，包含多个主流的 Python 开发调试环境，适合在数据处理和科学计算的领域使用。由于 Anaconda 是一个跨平台工具，因此它可以很好地在 Windows、Linux 和 macOS 上使用。

Anaconda 具有开源、安装过程简单、高性能使用 Python 和 R 语言、免费的社区支持等特点。

Anaconda 的安装过程如下。安装过程可以扫码观看。



(1) 打开 Anaconda 的官方网站 <https://www.anaconda.com>, 如图 1-13 所示, 单击 Download, 选择操作系统类型, 然后选择需要的软件版本下载即可。

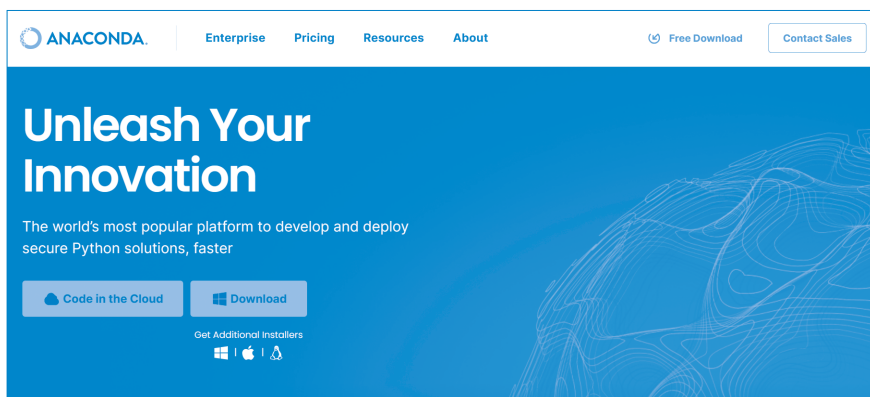


图 1-13 Anaconda 官方网站

(2) 打开下载的程序文件, 例如 Anaconda3-2022.10-Windows-x86\_64.exe, 如图 1-14 所示; 单击 Next 按钮进入安装许可协议界面, 如图 1-15 所示。

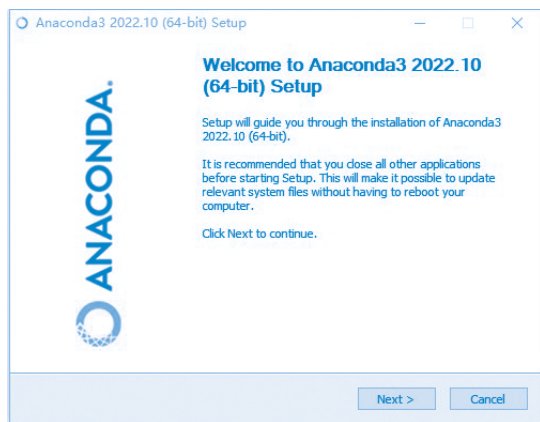


图 1-14 Anaconda3 安装界面

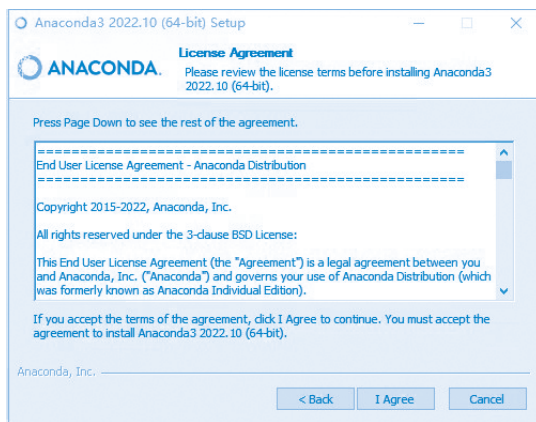


图 1-15 Anaconda3 安装许可协议界面

(3) 单击 I Agree 按钮进入安装类型界面, 选择相应的安装类型选项, 单击 Next 按钮进入安装路径界面, 如图 1-16 所示。

(4) 选择 Anaconda3 的安装路径, 单击 Next 按钮进入高级安装选项界面, 如图 1-17 所示。

(5) 勾选两个复选框, 第一个是添加到环境变量, 第二个是默认使用 Python 3.9, 单击 Install 按钮。安装完成后, 单击 Next 按钮进入安装完成界面, 单击 Finish 按钮结束安装, 如图 1-18 所示。

(6) 安装完成后, 可在“开始”菜单中找到 Anaconda3 文件夹, 查看所包含的内容, 如图 1-19 所示。

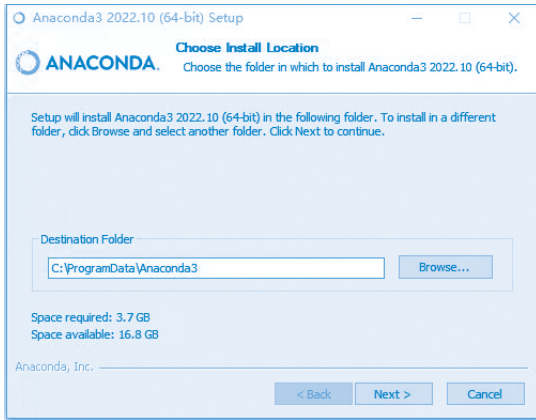


图 1-16 Anaconda3 安装路径界面

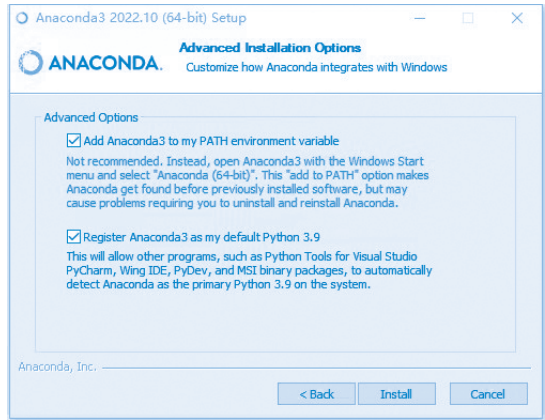


图 1-17 Anaconda3 高级安装选项界面

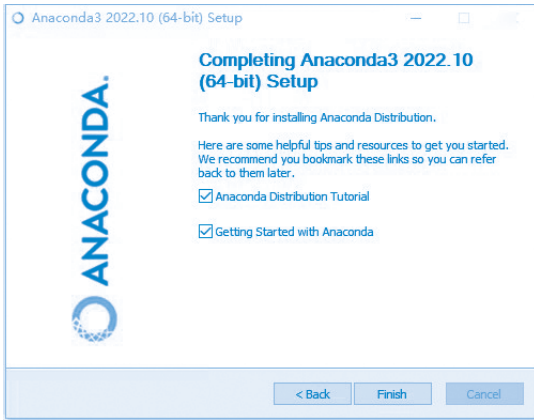


图 1-18 Anaconda3 安装完成

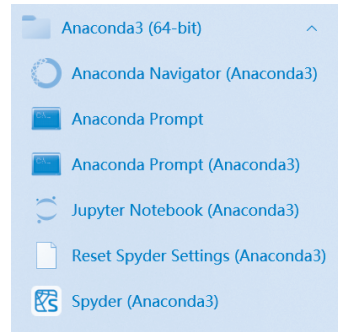


图 1-19 Anaconda3 文件夹

(7) 双击图 1-19 中的 Anaconda Navigator，若成功启动，则说明真正地安装了 Anaconda；如果未成功，请务必仔细检查以上安装步骤。启动页面如图 1-20 所示。

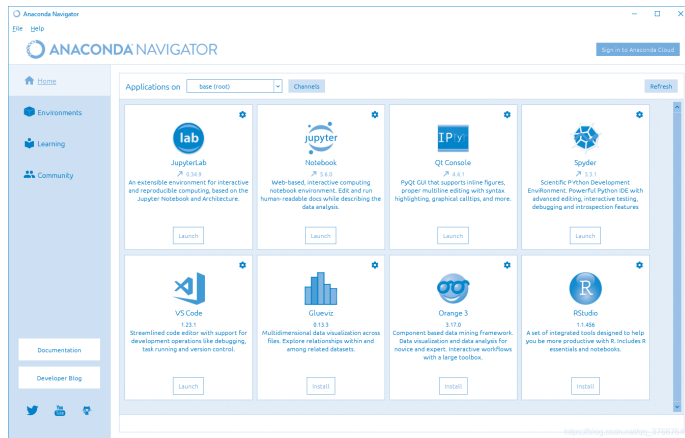


图 1-20 Anaconda Navigator 界面

## 1.3.2 Jupyter Notebook、Lightly

### 1. Anaconda & Jupyter Notebook

Jupyter Notebook 是基于网页的用于交互计算的应用程序，其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。

Jupyter 提供了一个 IDE，用户可以在该环境里写代码、运行代码、查看结果，并在其中可视化数据。鉴于众多优点，Jupyter 成为数据科学家喜欢的一款工具，它能够执行各种端到端的任务，如数据清洗、统计建模、机器学习等。而且它还允许把代码写入独立单元（cell）中，然后单独执行。

在“开始”菜单中找到 Anaconda3 文件夹。单击 Jupyter Notebook 即可启动 Jupyter，打开之后就可以看到如图 1-21 所示的 Jupyter 主界面了。

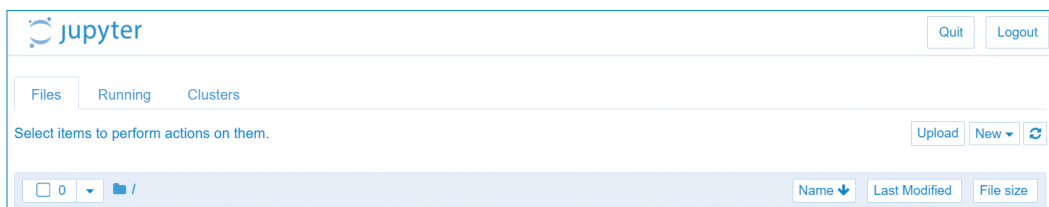


图 1-21 Jupyter 主界面

如果想新建一个 Notebook，单击面板右侧的 New，会出现 4 个选项。这里介绍使用 Notebook 来编写 Python 脚本。选择 Python 3，新建完成的界面如图 1-22 所示。

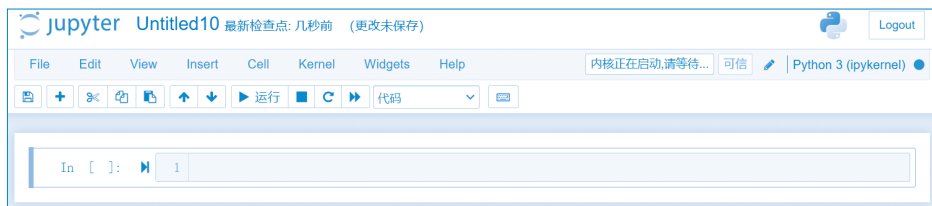


图 1-22 新建完成的界面

在 cell 中输入代码后，可以单击“运行”按钮或者按 Shift + Enter 组合键运行代码，如图 1-23 所示。

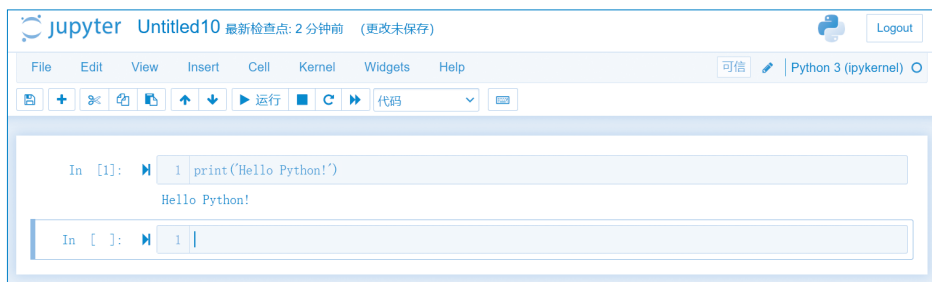


图 1-23 运行代码后的界面

## 2. Lightly

Lightly 是一款轻量且功能强大的集成开发工具。Lightly 支持多种主流的编程语言，用户无须任何操作，Lightly 会自动检测和生成开发环境，用户的任何修改都能实时同步到云端，并支持邀请其他人线上协助调试，共同协助开发。它既可以在网页上直接新建项目编程，也可以下载客户端体验更完整的 IDE 功能，如图 1-24 所示。



图 1-24 Lightly 官方网站

在浏览器地址栏中输入 <https://lightly.teamcode.com>，单击“在线使用”，登录后进入主界面，免费版提供了 500MB 的使用空间。单击右上角的“新建项目”，选择 Python，有 Python 3.7/3.8/3.9/3.10 四个版本可选。新建项目后，就可以输入代码了，还可以生成链接邀请好友、组员等一同协作，如图 1-25 所示。



图 1-25 Lightly 在线使用界面

### 1.3.3 PyCharm

PyCharm 是由 JetBrains 打造的一款 Python IDE，它拥有一整套在用户使用 Python 语言进行开发时使其效率提高的工具，如调试、语法高亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。此外，PyCharm 还提供了一些高级功能，用于支持 Django 框架下的专业 Web 开发。

PyCharm 是唯一一款专门面向 Python 的全功能集成开发环境。PyCharm 的产品分为社区（免费）版和专业（收费）版。我们编写的绝大多数程序，使用社区的免费版就可以完成。PyCharm 的安装与配置过程可以扫码观看。



## 1. PyCharm 安装

(1) 进入 PyCharm 官网 <https://www.jetbrains.com/pycharm/>，根据需要下载相应的版本。PyCharm 在 Windows 环境下有专业版（收费版）和社区版（免费版）两个不同的版本，以下载社区版为例。

(2) 打开下载的程序文件，例如 `pycharm-community-2023.1.3.exe`，如图 1-26 所示，单击 Next 按钮进入 PyCharm 安装路径界面，如图 1-27 所示。

(3) 选择 PyCharm 安装路径，单击 Next 按钮进入 PyCharm 选项界面，如图 1-28 所示。

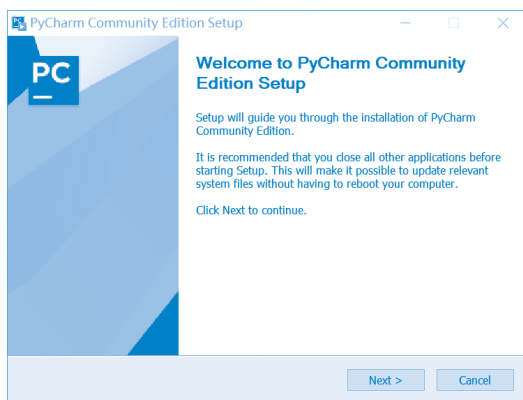


图 1-26 PyCharm 安装界面

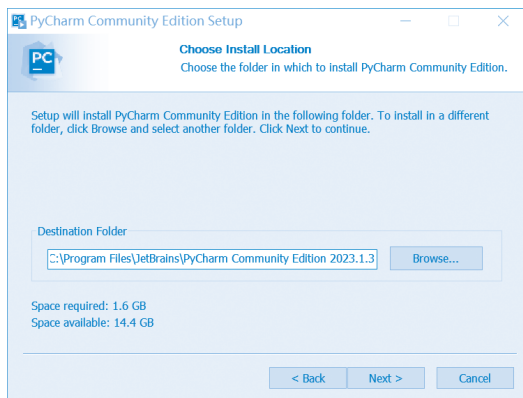


图 1-27 PyCharm 安装路径界面

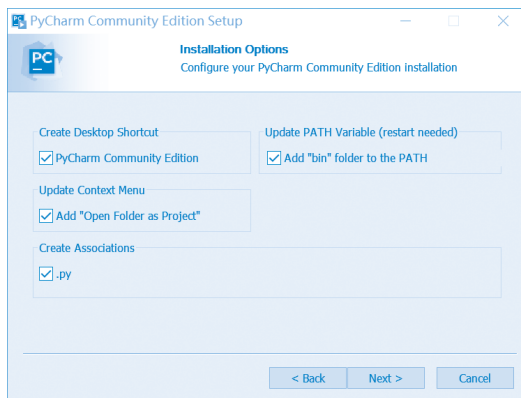


图 1-28 PyCharm 选项界面

(4) 选择相应的选项，单击 Next 按钮进入 PyCharm 菜单文件界面，如图 1-29 所示。使用默认设置，单击 Install 按钮开始安装。

(5) 安装完成后，如图 1-30 所示，单击 Finish 按钮结束安装。

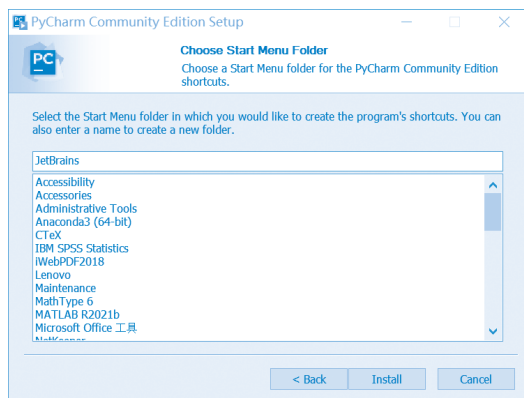


图 1-29 PyCharm 菜单文件界面

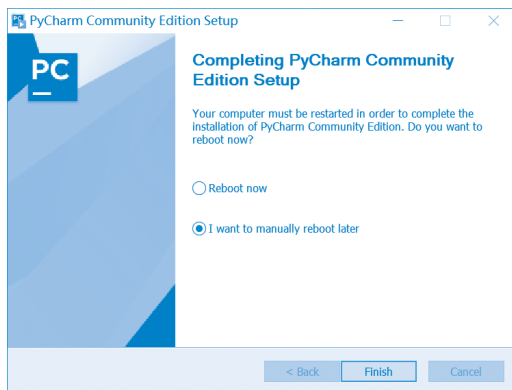


图 1-30 PyCharm 安装完成

## 2. PyCharm 配置

(1) 双击桌面快捷方式，启动 PyCharm，在弹出的协议窗口中勾选“同意协议”，然后单击 Continue 按钮，进入 PyCharm 欢迎界面，如图 1-31 所示。

(2) 单击 New Project 按钮新建项目，进入项目配置界面，如图 1-32 所示。先更改 Location，即新项目文件夹的路径和名称；然后单击左下方的 Previously configured interpreter，继续单击 Add Interpreter 按钮进去找到 Python 位置，前提是之前已经安装好 Python，如图 1-33 所示，单击 OK 按钮，返回项目配置界面；单击 Create 按钮，完成配置 Python 解释器。

(3) 根据个人喜好依次选择 File → Settings → Editor → Color Scheme/Font，确定主题和字体等。

(4) 统一字符编码设置为 UTF-8，如图 1-34 所示。

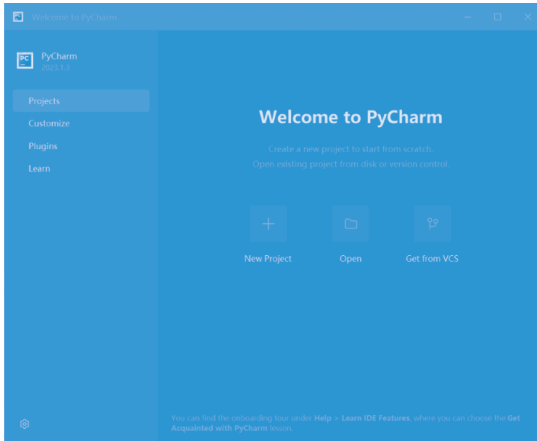


图 1-31 PyCharm 欢迎界面

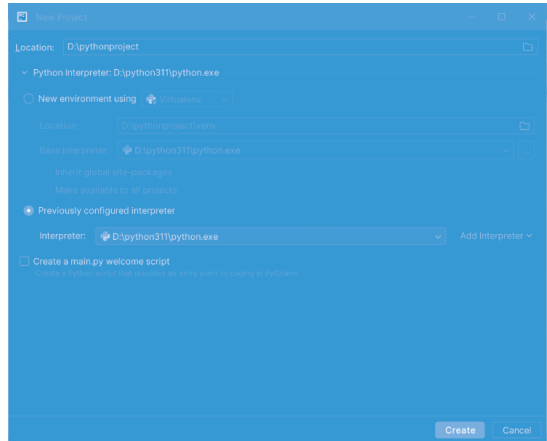


图 1-32 PyCharm 项目配置界面

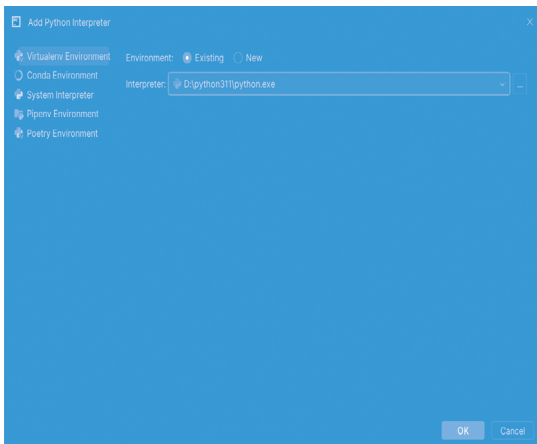


图 1-33 PyCharm 解释器界面

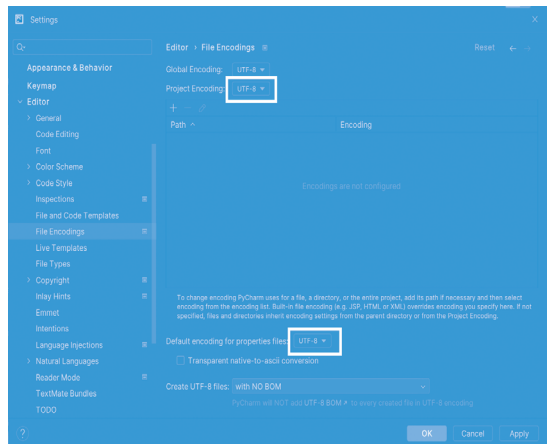


图 1-34 PyCharm 字符编码设置

(5) 依次选择 File → Settings → Editor → File and Code Templates → Python Script 设置自动添加头部注释。头部注释包括 Python 解释器的位置、字符集、作者信息、创建脚本的时间等；还可以依次选择 File → Settings → Keymap → Editor Actions 查看和自定义快捷键等。

### 1.3.4 实训任务：配置开发环境

#### 例 5：配置 Pycharm、安装插件 CSV Editor。

从官网下载 PyCharm 软件并完成安装配置。PyCharm 是一款功能强大的 Python 开发 IDE，拥有强大的功能和插件市场。PyCharm 的插件市场包含各类插件，包括主题、代码辅助工具、静态分析工具等，这些插件为 Python 开发带来了许多便利。因此，安装配置完成后，在 PyCharm 中安装插件，方法如下。

#### 1. 在 PyCharm 中打开插件市场

在 PyCharm 中依次单击 File → Manage IDE Settings → Settings Sync → Plugins，搜索所需要的插件并直接进行安装。安装完成后，PyCharm 会提示重新启动，重新启动后，插件即可启用。程序运行效果可以扫码观看。



#### 2. 手动获取插件市场链接地址

进入 PyCharm 的官方插件市场 <https://plugins.jetbrains.com/>，浏览所需要的插件，在插件详情页中下载插件的 .zip 文件。在 PyCharm 中依次单击 File → Manage IDE Settings → Settings Sync → Plugins，单击右上角设置符号中的 install plugin from disk，手动选择所下载的 .zip 文件，确认即可安装。安装完成后，PyCharm 会提示重新启动，重新启动后，插件即可启用。

## 1.4 课后思考

1. Python 是一门怎样的语言？
2. 列举常用的数据分析方法。
3. 描述 Python 的特点。
4. 描述 Python 支持哪几种代码运行方式。
5. 描述 Python IDE。
6. 简述 Python 安装扩展库常使用的工具及其使用方法。