python 编程基础



类目：计算机类
书名：python 编程基础
主编：陶娅芝　陈令刚　刘旭东
出版社：电子科技大学出版社
开本：大 16 开
书号：978-7-5770-2546-9
使用层次：通用
出版时间：2026 年 8 月
定价：49.80 元
印刷方式：双色
是否有资源：有



Python编程基础

策划编辑：万晓桐
责任编辑：万晓桐
封面设计：旗语书装

电子科技大学出版社
微信公众号

ISBN 978-7-5770-2546-9
9 787577 025469 >
定价：49.80元

计算机类创新融合教材
“互联网+”教育改革新理念教材

Python编程基础

主编 ◎ 陶娅芝 陈令刚 刘旭东

电子科技大学出版社

计算机类创新融合教材
“互联网+”教育改革新理念教材



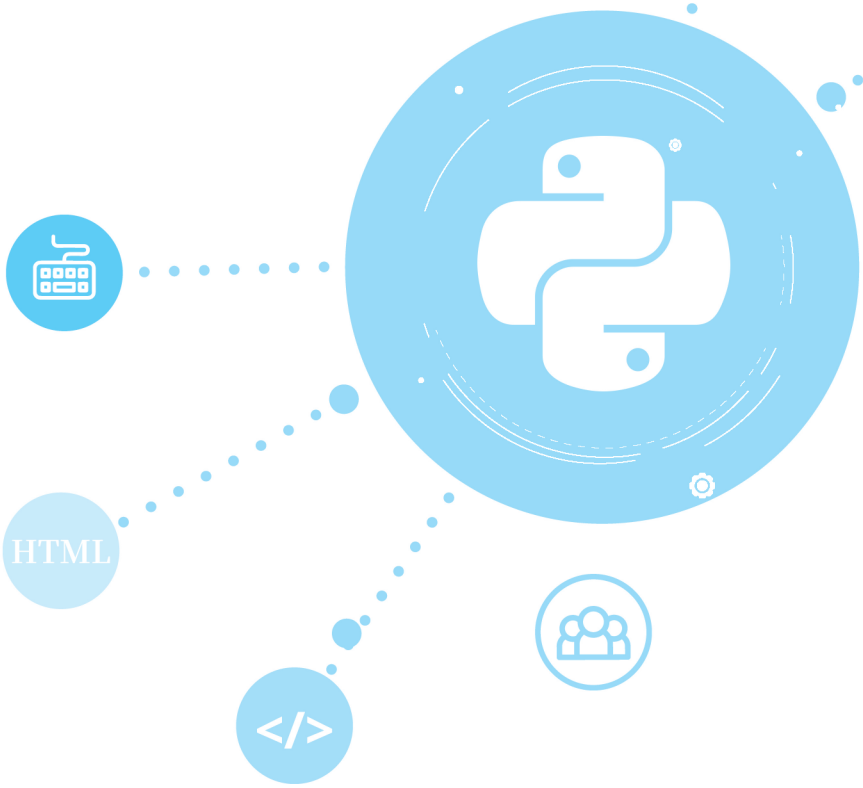
Python
编程基础



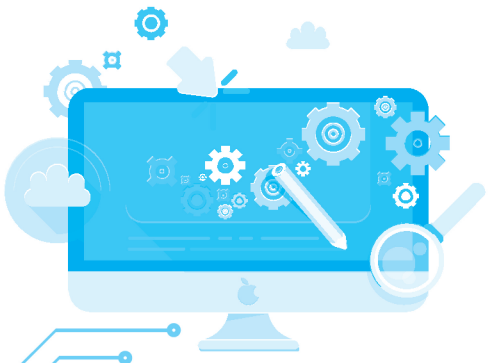
主编 ◎ 陶娅芝 陈令刚 刘旭东

电子科技大学出版社

计 算 机 类 创 新 融 合 教 材
“互联网 +”教育改革新理念教材



Python 编程基础



主　编 ◎ 陶娅芝　陈令刚　刘旭东
副主编 ◎ 张思明　朱佳强　刘　辉
董　毅　黄金花

电子科技大学出版社
University of Electronic Science and Technology of China Press
·成都·

图书在版编目(CIP)数据

Python 编程基础 / 陶娅芝，陈令刚，刘旭东主编. 成都 ：成都电子科大出版社，2026. 8. -- ISBN 978-7-5770-2546-9

Ⅰ. TP312.8

中国国家版本馆 CIP 数据核字第 2026BJ9488 号

Python 编程基础

Python BIANCHENG JICHU

陶娅芝　陈令刚　刘旭东　主编

策划编辑　万晓桐
责任编辑　万晓桐
责任校对　罗　雅
责任印制　梁　硕

出版发行　电子科技大学出版社
　　成都市一环路东一段 159 号电子信息产业大厦九楼　邮编　610051
主　　页　www. uestcp. com. cn
服务电话　028-83203399
邮购电话　028-83201495

印　　刷　三河市祥达印刷包装有限公司
成品尺寸　210 mm×285 mm
印　　张　14
字　　数　371 千字
版　　次　2026 年 8 月第 1 版
印　　次　2026 年 8 月第 1 次印刷
书　　号　ISBN 978-7-5770-2546-9
定　　价　49.80 元

版权所有　侵权必究

PREFACE 前言

Python 是一门简单易学且功能强大的编程语言。它拥有高效的高级数据结构，并且能够用简单而又高效的方式进行面向对象编程。Python 是一种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，它编写的程序具有很强的可读性。

本书以由浅入深、循序渐进的方式展开讲解，以合理的结构和经典的范例对 Python 最基本和实用的功能进行了详细的介绍，具有极高的实用价值。通过对本书的学习，读者不仅可以掌握 Python 的基本知识和应用技巧，而且可以灵活地运用 Python 进行各种编程应用。

一 本书特点

◇ 实例丰富

本书的实例不管是数量还是种类都非常丰富。本书结合大量的 Python 编程实例，详细讲解了 Python 原理与应用知识要点，让读者在学习案例的过程中潜移默化地掌握 Python 应用技巧。

◇ 突出提升技能

本书从全面提升 Python 实际应用能力的角度出发，结合大量的案例来讲解如何使用 Python，使读者了解 Python 基本原理并能够独立地完成各种 Python 应用操作。

本书中有很多实例就是 Python 开发项目案例，经过作者精心提炼和改编，不仅能帮助读者学好知识点，更重要的是能够帮助读者掌握实际的操作技能和开发能力。

◇ 技能与思政教育紧密结合

在讲解 Python 开发专业知识的同时，紧密结合思政教育主旋律，从专业知识角度触类旁通地引导学生提升相关思政品质。

◇ 项目式教学，实操性强

全书采用项目式教学，把 Python 理论知识分解并融入一个个实践操作的训练项目中，增强了本书的实用性。

二 本书的基本内容

全书分为 Python 概述、Python 基础知识、函数运算、标准库函数、程序结构和高级功能等六个项目。

本书实例丰富、内容翔实、操作方法简单易学，不仅适合作为学校计算机与软件工程相关专业教材，也可供从事 Python 编程相关工作的专业人士参考。

编　者

2026 年 4 月

CONTENTS 目录

项目一 Python 概述 …… 1

【思政目标】 …… 1

【技能目标】 …… 1

【项目导读】 …… 1

任务 1 初识 Python …… 2

任务 2 解释器 IDLE Shell 3.11.0 …… 9

任务 3 集成开发环境 PyCharm …… 13

任务 4 集成开发环境 VS Code …… 30

【项目实战】 …… 40

项目二 Python 基础知识 …… 45

【思政目标】 …… 45

【技能目标】 …… 45

【项目导读】 …… 45

任务 1 Python 语法基础 …… 46

任务 2 基本数据类型 …… 58

任务 3 数据类型转换 …… 73

【项目实战】 …… 77

项目三 函数运算 …… 81
【思政目标】 …… 81
【技能目标】 …… 81
【项目导读】 …… 81
任务 1 内置函数 …… 82
任务 2 自定义函数 …… 102
任务 3 面向对象编程 …… 110
【项目实战】 …… 116

项目四 标准库函数 …… 120
【思政目标】 …… 120
【技能目标】 …… 120
【项目导读】 …… 120
任务 1 os 模块 …… 121
任务 2 datetime 模块 …… 135
任务 3 数学运算模块 …… 140
【项目实战】 …… 150

项目五 程序结构 …… 154
【思政目标】 …… 154
【技能目标】 …… 154
【项目导读】 …… 154
任务 1 顺序结构 …… 155
任务 2 选择结构 …… 159
任务 3 循环结构 …… 171
任务 4 程序的流程控制 …… 178
【项目实战】 …… 183

项目六 高级功能 …… 186
【思政目标】 …… 186
【技能目标】 …… 186
【项目导读】 …… 186
任务 1 Python GUI 编程 …… 187
任务 2 Python AI 绘画 …… 204
【项目实战】 …… 209

参考文献 …… 215

项目一

Python 概 述

★ ★ ★

思政目标

- ◇ 提高软件操作和信息处理能力，提升学生的适应性和学习能力。
- ◇ 引导学生自主学习，对比其他编程语言，与时俱进，学习 Python 语言。

技能目标

- ◇ 掌握 Python 的安装方法。
- ◇ 掌握 PyCharm 的安装方法。
- ◇ 熟悉 PyCharm 的编辑环境。
- ◇ 熟悉 VS Code 的编辑环境。

项目导读

Python 是一种代表简洁设计思想的语言。阅读一段规范的 Python 程序就感觉像是在读英语一样。它使你能够专注于解决问题而不是去搞明白语言本身。同时，Python 毫无学习门槛，极容易学会，因为 Python 有极其简单的说明文档。

任务 1 初识 Python

任务引入

小唐是一名大学生，大一下半学期选修了信息技术，老师介绍了当下流行的几种编程语言，推荐大家学习。如 Matlab、Python 。小唐从最简单的 Python 入手。那么，Python 究竟是一款什么样的软件？如何下载安装？软件界面是什么样的呢？

知识准备

自诞生之初，Python 的定位就是优雅、明确、简单，因此提供了高效的高级数据结构，严谨的程序设计语言，使得 Python 程序总是简单易懂，被广泛应用于系统管理任务的处理和 Web 编程，成为最受欢迎的程序设计语言之一。其主要特点有以下几点。

- Python 是一种解释型语言：在开发过程中没有了编译这个环节，类似于 PHP 和 Perl 语言。
- Python 是交互式语言：可以在一个 Python 提示符>>>>>后直接执行代码。
- Python 是面向对象语言：Python 支持面向对象的风格或代码封装在对象的编程技术。
- Python 是初学者的语言：对于初级程序员而言，Python 是一种伟大的语言。它支持广泛的应用程序开发，从简单的文字处理到 WWW 浏览器再到游戏。

一、Python 简介

1989 年，荷兰数学和计算机科学研究学会(CWI)的吉多·范罗苏姆(Guido van Rossum)为了克服 ABC 语言非开放的缺点，受 Modula-3 的影响，结合了 Unix shell 和 C 的习惯，使用 C 语言开发了一个新的脚本解释程序，命名为 Python 。

1991 年，Python 发布了第一个版本。此时，Python 已经具有了类、函数、异常处理，包含列表和字典在内的核心数据类型，以及以模块为基础的拓展系统。

1991—1994 年，Python 增加了 lambda、map、filter and reduce。1995 年，Guido van Rossum 在弗吉尼亚州的国家创新研究公司(CNRI)继续他在 Python 上的工作，发布了该软件的多个版本。

1999 年，Python 的 web 框架之祖——Zope 1 发布。

2000 年 5 月，Guido van Rossum 和 Python 核心开发团队转到 BeOpen. com，并组建了 BeOpen Python Labs 团队。同年 10 月，BeOpen Python Labs 团队转到 Digital Creations(现

为 Zope Corporation)，发布了 Python 2，加入了内存回收机制，构成了现在 Python 语言框架的基础，该系列稳定版本是 Python 2.7。

2001 年，Python 软件基金会(PSF)成立，这是一个专为拥有 Python 相关知识产权而创建的非营利组织。2005 年，Web 框架 Django 诞生，自此以后，Python 的使用率呈线性增长。

2008 年 12 月 3 日，Python 发布了 Python 3，该版本不完全兼容 Python 2。2011 年 1 月，Python 3 被 TIOBE 编程语言排行榜评为 2010 年度最佳语言。

2021 年 10 月 4 日，Python 正式发布了 Python 3.10 版本。2022 年 10 月 24 日发布 Python 3.11 版本。

2023 年 10 月 2 日，Python 发布 Python 3.12.0 版本，2023 年 12 月 8 日发布了 Python 3.12.1 版本，2024 年 2 月 6 日，Python 正式发布了 Python 3.12.2 版本。

二、安装 Python

Python 是一种解释型脚本语言，若要运行编写的代码，需要先安装 Python 解释器。

1. 下载 Python

打开 Python 官方下载页面 https://www.Python .org/downloads/，如图 1-1 所示，在“Active Python Releases”选项组下显示不同版本的 Python（Python 3.8～ Python 3.13 版）。其中，Python 3.13 版是已发布的版本，于 2024 年 10 月 7 日发布。

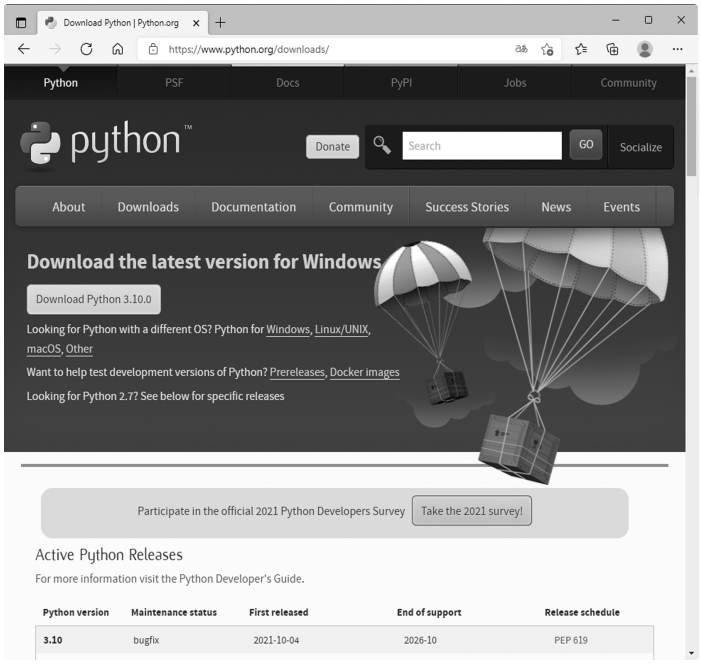


图 1-1　Python 版本列表

若需要下载 Python 3.12.2 版，直接在官网最上端单击“Download Python 3.12.2”按钮，直接下载 Python 3.12.2 版的安装程序 Python -3.12.2-amd64.exe(64 位的完整的离线安装包)即可。

提示

每个版本的 Python 根据不同的计算机操作系统，分为不同的安装软件。计算机操作系统包括 Windows、Linux/UNIX、macOS 等。一般学生使用 Windows 系统的较多，因此这里只介绍 64 位 Windows 环境下 Python 3.12.2 版的下载及安装过程。本书中介绍的程序也是在该系统下进行演示的。

2. 安装 Python

(1)双击安装文件 Python-3.12.2-amd64.exe，弹出 Python 3.12.2(64-bit) Setup 对话框中的安装界面 Install Python 3.12.2(64-bit)，下面介绍该界面中的选项。

- InstallNow：默认安装且默认安装路径不能更改(一般默认安装在 C 盘)。
- Customize installation：自定义安装。
- Use admin privileges when installing py.exe：勾选该复选框，安装 py.exe 时使用管理员权限。
- Add python.exe to PATH：勾选该复选框，将 Python 自动加到环境变量中，如图 1-2 所示。这样可以将 Python 命令工具所在目录添加到系统 Path 环境变量中，以后开发程序或者运行 Python 命令会非常方便。



图 1-2　安装界面

提示

勾选“Add python.exe to PATH”复选框非常重要，若不勾选该选项，软件安装完成，进行检查时，命令行提示符将显示“Python 不是内部或外部命令，也不是可运行的程序或批处理文件”，如图 1-3 所示。若要解决这种问题，还需要手动在计算机的环境变量中添加 Python 安装路径。

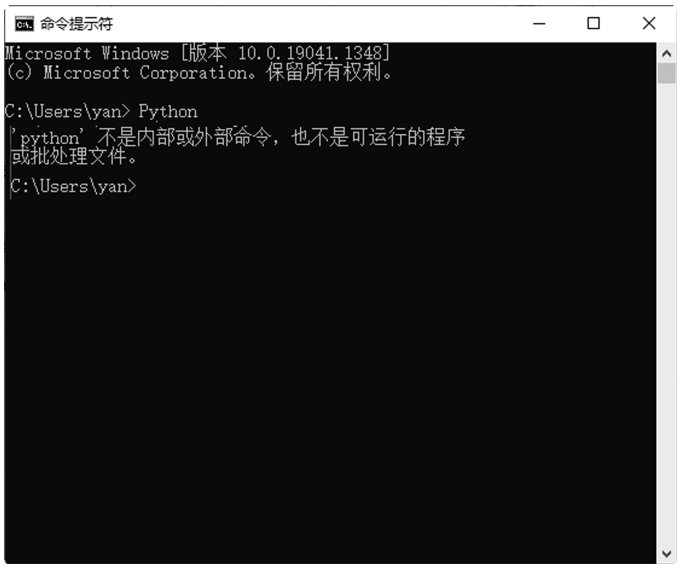


图 1-3　安装错误信息

(2)单击“Customize installation”(自定义安装)按钮，弹出 Python 3.12.2(64-bit) Setup 对话框中的选项设置界面 Optional Features，选择默认参数设置，如图 1-4 所示。

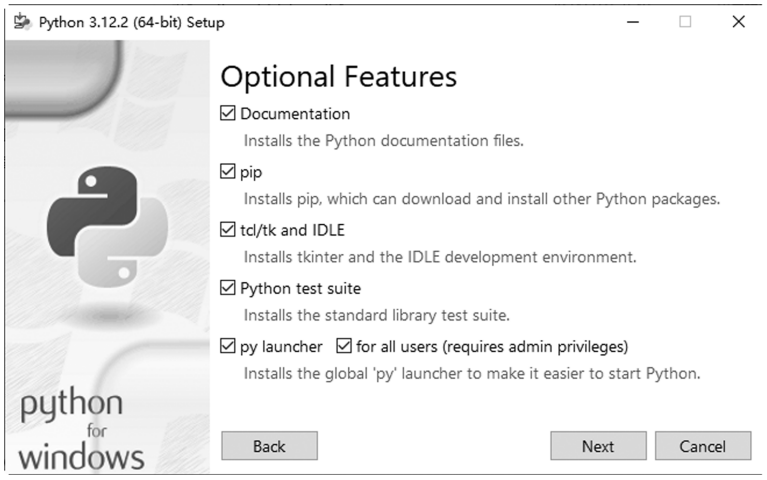


图 1-4　选项设置对话框

• Documentation：勾选该复选框，安装 Python 帮助文档。

• pip：勾选该复选框，安装下载 Python 的工具 pip，pip 是现代通用的 Python 包管理工具，英文全称是 Python install packages。

• td/tk and IDLE：安装开发环境 tkinter 和 IDLE。

• Python test suite：安装标准库测试套件。

• py launcher：安装 Python 的启动器。

• for all users(requires admin privileges)：适用所有用户。

(3)单击“Next”(下一步)按钮，进入下一个高级设置画面 Advanced Options，在“Customize install location”文本框更改安装地址(选择的路径应具有写入权限，不建议安装在 C

盘)，其余选择默认设置，设置完毕后如图 1-5 所示。

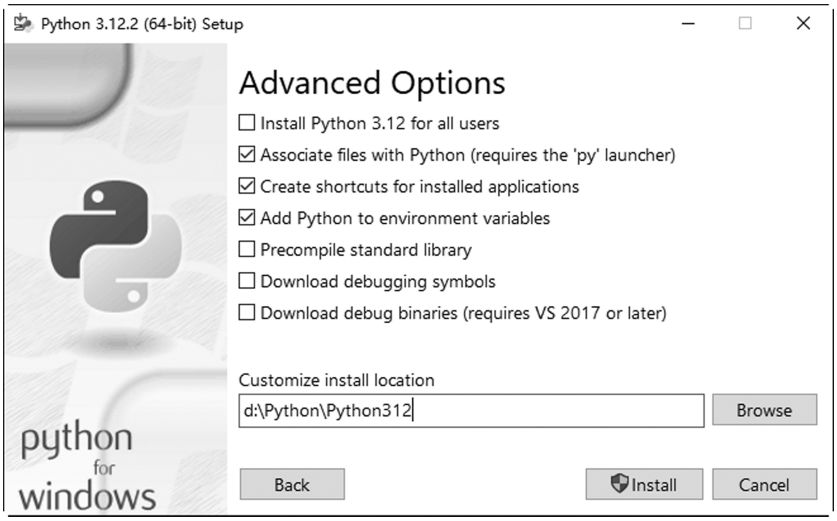


图 1-5　高级设置

(4)确定好安装路径后，单击“Install”(安装)按钮，此时对话框内会显示安装进度，如图 1-6 所示。由于系统需要复制大量文件，所以需要等待几分钟。在安装过程中，可以随时单击 Cancel(取消)按钮来终止安装过程。

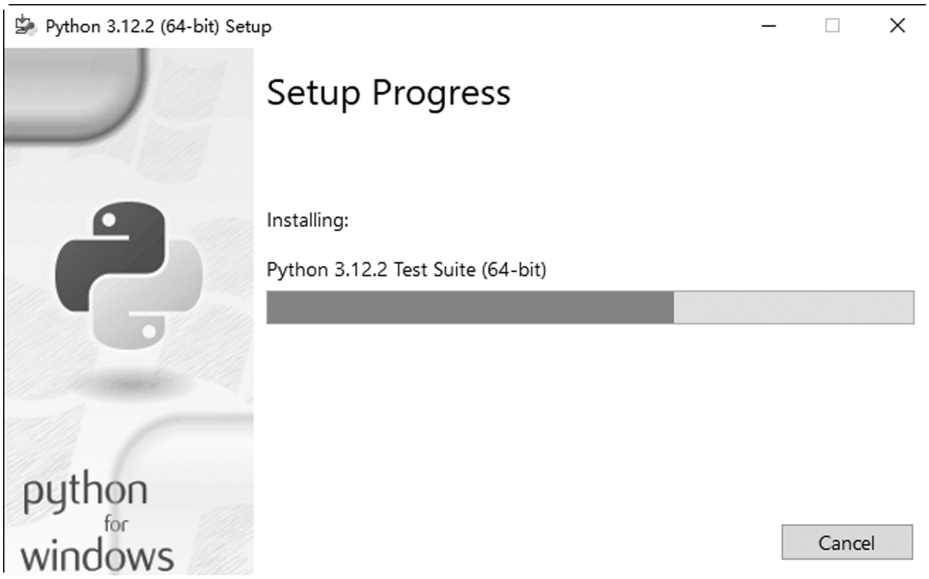


图 1-6　安装进度对话框

(5)安装结束后，会出现一个“Setup was successful”(安装成功)对话框，如图 1-7 所示。单击 Close(关闭)按钮，即可完成 Python 3.12.2 版的安装工作。

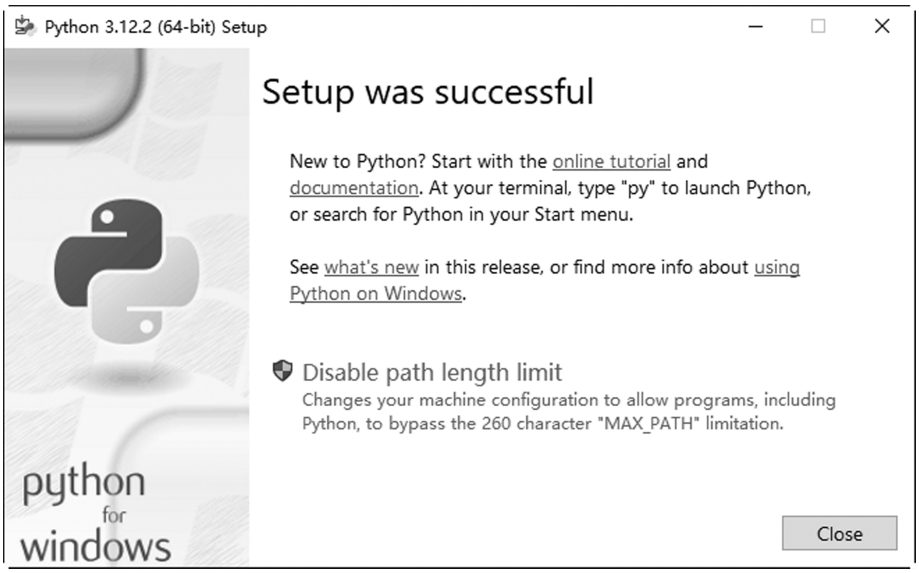


图 1-7 “Setupwas successful”(安装成功)对话框

3. 检查安装是否成功

Python 安装结束后，还需要检查安装是否成功。

(1)安装结束后，单击桌面左下角的“开始”按钮，在弹出的程序列表中选择“Windows 系统”→“运行”命令，打开“运行”对话框。

(2)在“运行”对话框中输入“cmd”命令，如图 1-8 所示，单击“确定”按钮，打开命令提示符窗口。

(3)在命令提示符后输入“Python ”命令，按 Enter 键，出现如图 1-9 所示的运行结果，表示 Python 安装成功。

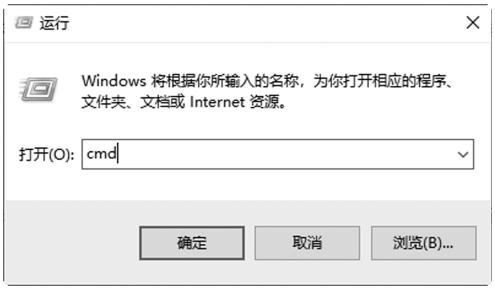


图 1-8 “运行”对话框

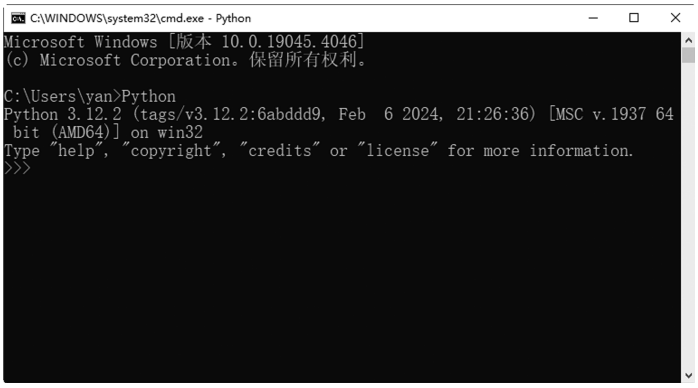


图 1-9 安装检查

案例 1 ——Python 的交互式运行环境

当安装好 Python 后，接下来通过一个简单的加减法运算实例，帮助读者熟悉 Python 的交互式运行环境。

【操作步骤】

(1)按键盘上的“Windows+R”组合键，打开“运行”对话框。

(2)在“打开”文本框中输入“cmd”，单击“确定”按钮或按 Enter 键，打开命令提示符窗口，如图 1-10 所示。

(3)在命令提示符后输入“Python ”命令，按 Enter 键，显示 Python 的版本信息，同时启动交互式运行环境，并显示命令提示符>>>，如图 1-11 所示。

(4)在命令提示符>>>后输入下面的代码：

```
>>> 10+128-12
```

(5)按 Enter 键执行命令，即可在命令行窗口中看到执行结果，并输出一个新的命令提示符>>>，如图 1-12 所示。



图 1-10 命令提示符窗口

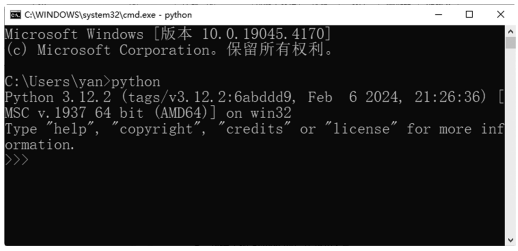


图 1-11 显示 Python 的版本信息

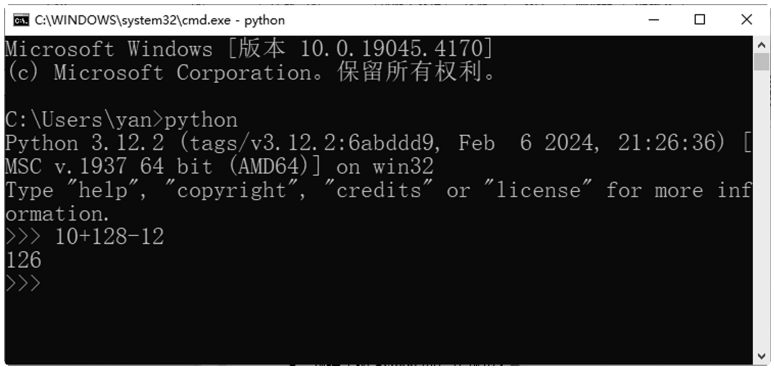


图 1-12 程序运行结果

(6)按“Ctrl+Z”组合键，或者输入 exit()命令，并按 Enter 键，即可退出交互式运行环境，回到 Windows 命令行程序，如图 1-13 所示。



图 1-13 退出交互式运行环境

(7)在命令提示符>>>后输入下面的代码：

```
>>> 10+128-12
```

Windows 命令行程序不识别上面的代码，会显示报错，如图 1-14 所示。



图 1-14 Windows 命令行程序报错

任务 2 解释器 IDLE Shell 3.11.0

任务引入

小唐的课堂作业是演示一个程序代码，这就需要选择 Python 编辑环境。那么，Python 的编辑环境是什么？怎么新建文件呢？做好的 py 文件怎么保存到指定位置？怎么打开已有的 py 文件？

知识准备

Python 的文件类型主要分为源代码(source file)、字节码(byte-code file)、优化的字节码(optimized file)三种。这些代码都可以直接运行，不需要编译或者连接。这正是 Python 语言的特性，Python 的文件通过 Python . exe 和 Python w. exe 解释运行。

- 源代码：Python 的源文件以“py”为扩展名，Python . exe 解释运行，可在控制台下运行。“pyw”是图形开发用户接口(GUI)文件的扩展名，作为桌面应用程序，这种文件用于开发图形界面，由 Python w. exe 解释运行。“py”和“pyw”的文件都可以用文本编辑器打开并编辑。
- 字节代码：Python 的源文件经过编译之后生成扩展名为“pyc”，该文件不能用文本编辑器打开或编辑。“pyc”文件与平台无关，因此 Python 程序可以运行在 Windows、Unix 和 Linux 等系统上。利用运行脚本可以将“py”文件编译成“pyc”文件。
- 优化代码：经过优化的代码生成扩展名为“pyo”的文件，该类文件也不能用文本编辑器打开或者编辑。“pyo”类型的文件需要通过命令行工具生成。通过“Win＋R”组合键启动命令行，进入“py”文件所在目录，输入命令进行编译。

Python 解释器 IDLE Shell 3. 11. 0 是一个功能完备的交互式编程客户端，在安装 Python 时会自动安装。

在开始菜单中单击 Python 3. 12 文件夹中的 IDLE(Python 3. 12 64-bit)菜单项，即可启动 IDLE Shell 3. 12. 2，如图 1-15 所示。在 IDLE Shell 中还可以新建 Python 文件，适合于较复杂的程序，或后续需要重复执行的程序，

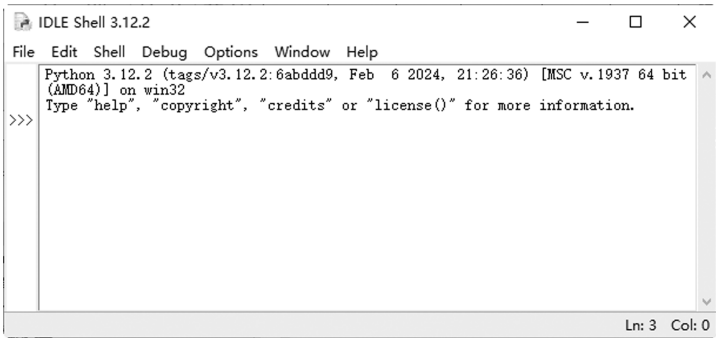


图 1-15　IDLE Shell 3. 12. 2 编辑器

在 Python 环境中，脚本文件的基本操作包括新建文件、打开文件、保存文件及关闭文件。

一、新建文件

在实际应用中，直接在 Python 工作空间的命令窗口中输入简单的命令并不能够满足用户的所有需求，因此 Python 提供了另一种强大的工作方式，即利用脚本文件编程，脚本文件一般以 . py 为后缀。

选择菜单栏中的“File”（文件）→“New File”（新建文件）命令，或直接按“Ctrl＋N”组合键，将打开一个新窗口，如图 1-16 所示。在该窗口中，可以直接编写 Python 代码，标题栏上的“untitled”为新建脚本文件的名称。

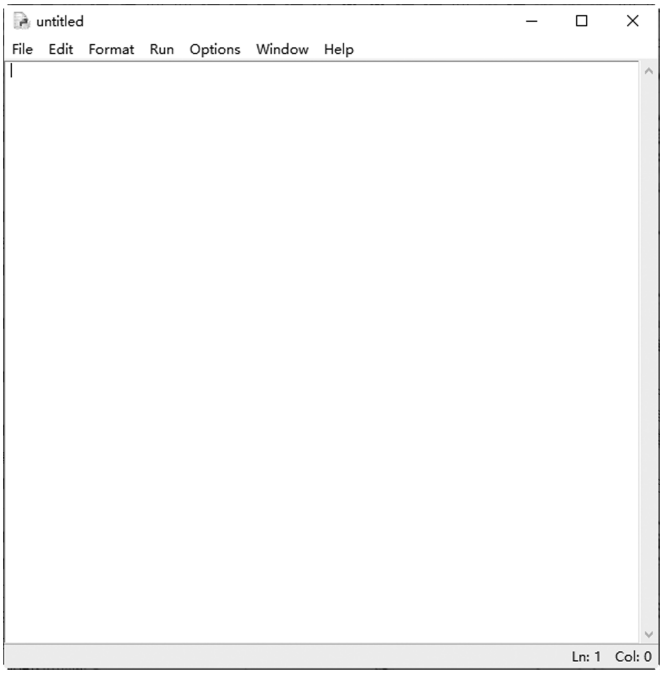


图 1-16　代码编辑文件

在输入一行代码后再按下 Enter 键，代码将自动换到下一行，等待继续输入，如图 1-17 所示，文件名带 * 号表示文件未保存。

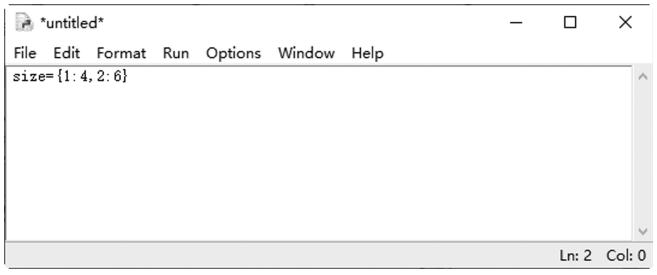


图 1-17　编辑脚本文件

二、保存文件

在处理脚本文件时，应时常保存文件，以防因为意外情况造成数据丢失。Python 提供了三种保存文件的常用方法。

1. 保存文件

选择菜单栏中的“File”（文件）→“Save”（保存文件）命令，或按“Ctrl＋S”组合键，保存文件。

在保存文件时，如果文件已经保存过，Python 将用新的文件内容覆盖原有的内容；如果新建的文件还未命名，则打开如图 1-18 所示的“另存为”对话框，单击要保存的位置，指定文

件的保存路径和名称，单击“保存”按钮，保存文件。

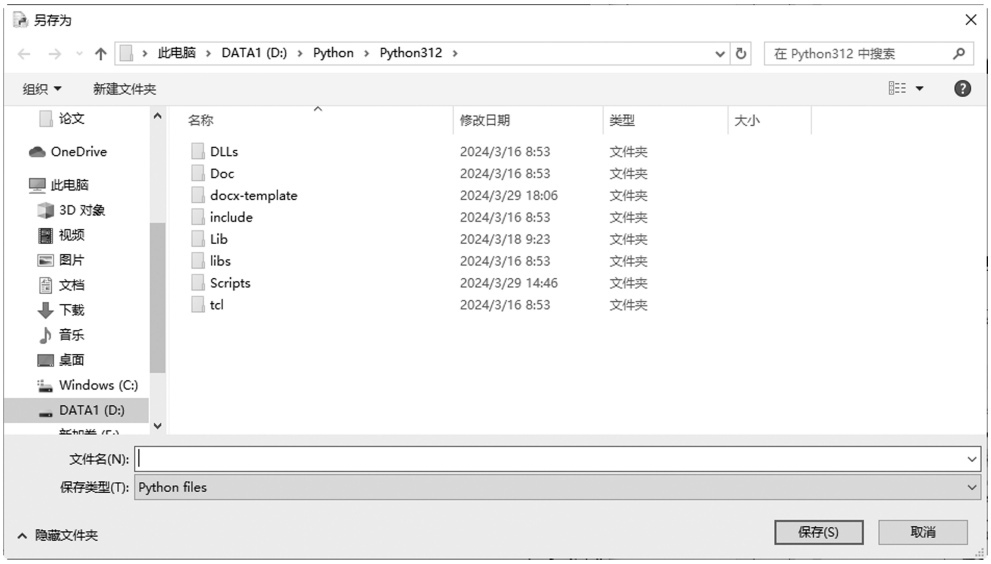


图 1-18 “另存为”对话框

2. 另存文件

选择菜单栏中的“File”(文件)→“Save As”(另存文件)命令，或按“Ctrl＋Shift＋S”组合键，打开 “另存为”对话框，保存文件。

3. 保存备份

选择菜单栏中的“File” (文件)→“Save Copy As”(保存备份文件)命令，或按“Alt＋Shift＋S”组合键，打开“另存为”对话框。对于重要的文件，在保存时可以将原版本保存为备份文件。

三、打开文件

打开一个已经存在的文件的常用步骤如下：

(1)选择菜单栏中的“File” (文件)→“Open”(打开文件)命令，或按“Ctrl＋O”组合键，打开如图 1-19 所示的“打开”对话框。

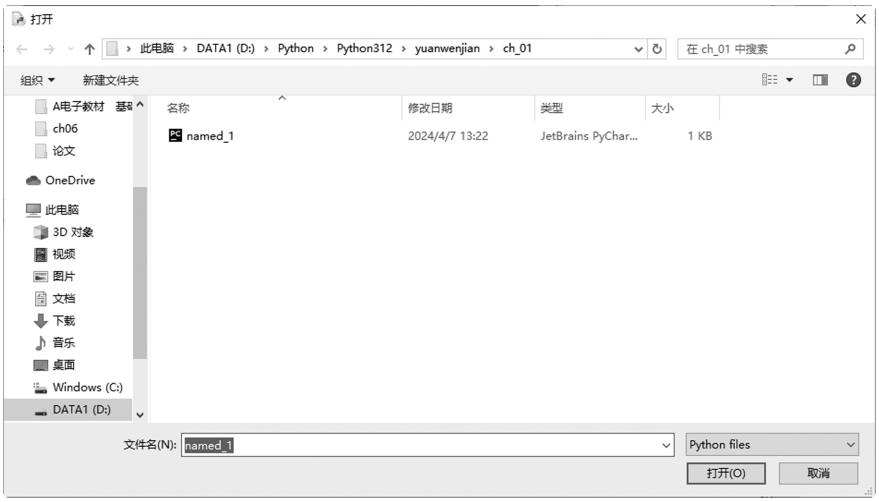


图 1-19 “打开”对话框

(2)浏览到文件所在路径，先单击文件名称，然后单击“打开”按钮，即可在 Python 中打开指定的文件。

四、关闭文件

如果不再需要某个打开的文件，应将其关闭，这样既可以节约一部分内存，也可以防止数据丢失。关闭文件常用的方法有以下两种：

- 单击“File”（文件）→“Close”(关闭文件)命令；
- 按“Alt＋F4”组合键。

五、运行文件

运行文件的常用步骤如下：

选择菜单栏中的 “Run”(运行) →“Run Module”(运行模块)命令，或直接按 F5 键，运行脚本文件中的程序，如图 1-20 所示。

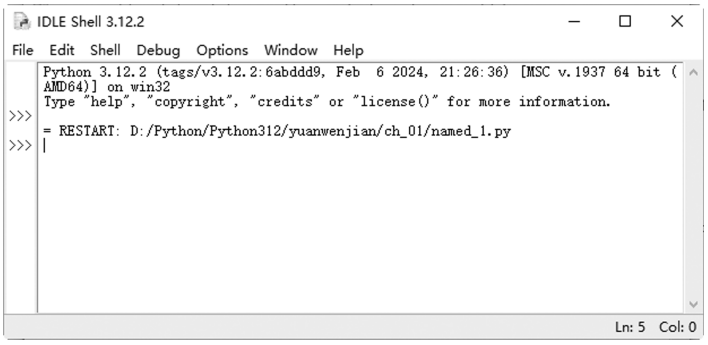


图 1-20　运行结果

任务 3　集成开发环境 PyCharm

任务引入

小唐准备演示一个希望小学的调查报告，其中包含图文的显示，需要多次调试和运行，自带的解释器使用过于烦琐，这就需要选择 Python 编辑环境。那么，Python 编辑环境还有什么？什么是 PyCharm？如何安装、使用 PyCharm？

知识准备

PyCharm 是由捷克的软件开发公司 JetBrains 打造的一款 Python IDE。PyCharm 具备一般 Python IDE 的功能，比如调试、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。PyCharm 还提供了一些很好的功能用于 Django 开发，同时支持 Google App Engine，以及在 NET 和 Mono 上实现的 Python 语言 IronPython 。

一、安装 PyCharm

首先，下载软件。

(1)登录 Pycharm 的官网下载地址：https：//www. jetbrains. com/pycharm/download/#section= Windows，可以看到有两个版本：Professional(专业版，收费)和 Community(社区版，免费)可供选择，如图 1-21 所示。

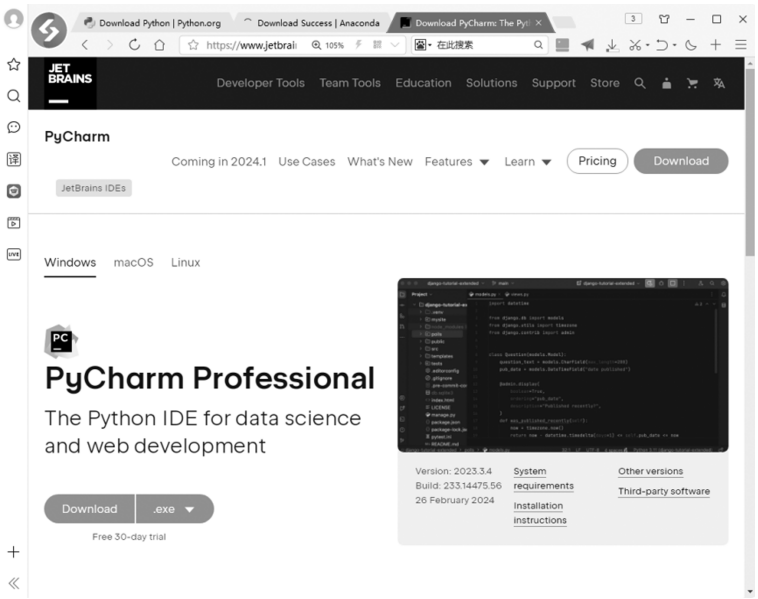


图 1-21　PyCharm 官网下载页面

一般对于初学者或学生来说，下载免费社区版即可。

(2)在 Community 版本下方的“Download(下载)”按钮右侧的下拉列表中，先根据计算机操作系统的 CPU 架构选择对应的下载程序 Windows(. exe)或 Windows ARM64(. exe)；然后单击“Download(下载)”按钮，下载 PyCharm 2023. 3. 4 版本 pycharm-community-2023. 3. 4. exe 文件。

(3)双击下载的 PyCharm-community-2023. 3. 4. exe 文件，弹出 PyCharm 的安装界面，如图 1-22 所示。

(4)单击“Next”(下一步)按钮，在打开的对话框中设置 PyCharm 的安装路径。默认安装在系统盘下，可以单击“Browse. . .(浏览)”按钮，自定义安装路径，如图 1-23 所示。



图 1-22　安装界面

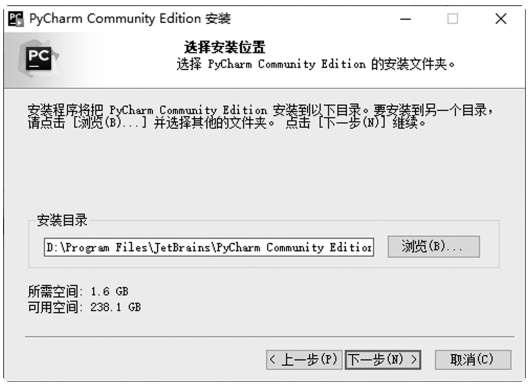


图 1-23　安装路径对话框

(5)单击“Next”(下一步)按钮，在如图 1-24 所示的对话框中设置安装选项。

- PyCharm Community Edition：勾选该复选框，为 PyCharm 创建桌面快捷方式。
- Add “bin” folder to the PATH：勾选该复选框，将 PyCharm 的启动目录添加到环境变量，执行该操作后，需要重启计算机。
- Add “Open Folder as Project”：勾选该复选框，添加鼠标右键菜单，使用打开项目的方式打开选中的文件夹。
- .py：勾选该复选框，将 .py 文件与 PyCharm 关联起来，默认用 PyCharm 打开 .py 文件。在勾选该选项后，PyCharm 每次打开的速度会比较慢。

(6)单击“Next”(下一步)按钮，设置 PyCharm 快捷方式所在的开始菜单项，默认选择 JetBrains，如图 1-25 所示。单击“安装”按钮，此时对话框内会显示安装进度，如图 1-26 所示，然后耐心等待安装完毕即可。



图 1-24　安装选项设置对话框

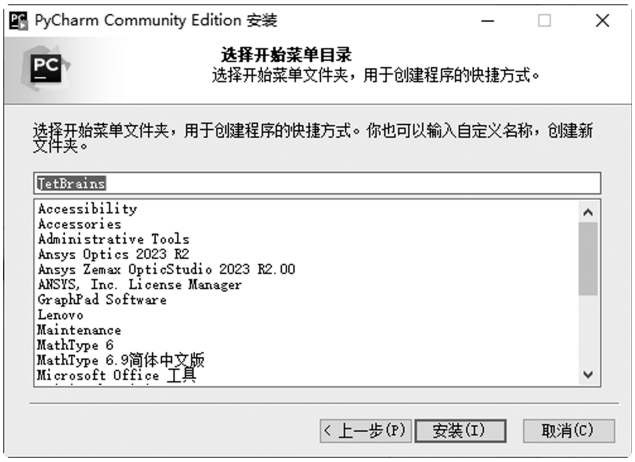


图 1-25　选择安装菜单文件

(7)安装结束后会出现一个对话框，提示用户必须重启计算机以完成 PyCharm 的安装，如图 1-27 所示。用户可以选中 Reboot now(立即重启)按钮即可自动重启计算机，也可以选中 I want to manually reboot later 按钮后，手动重启计算机。



图 1-26　安装过程

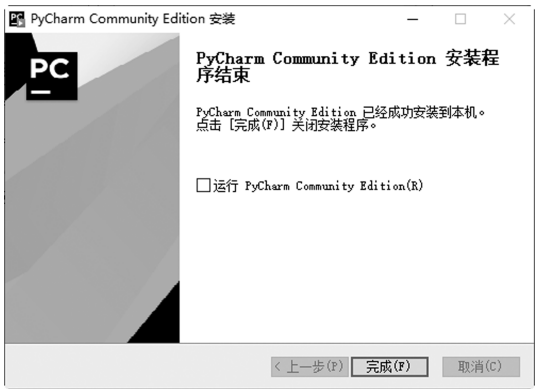


图 1-27　“安装”对话框

二、配置 PyCharm

双击运行桌面上的 PyCharm 图标“PyCharm Community Edition 2023.3.4”，激活 PyCharm 启动界面，如图 1-28 所示。进入用户协议界面，勾选“I confirm that I have read and accept the terms of this User Agreement”复选框，同意用户使用协议，如图 1-29 所示。



图 1-28　启动界面

单击“Continue”(继续)按钮，弹出数据共享界面 DATA SHARING，如图 1-30 所示，确定是否需要进行数据共享，选择“Don't Send”按钮，激活 PyCharm 启动界面。

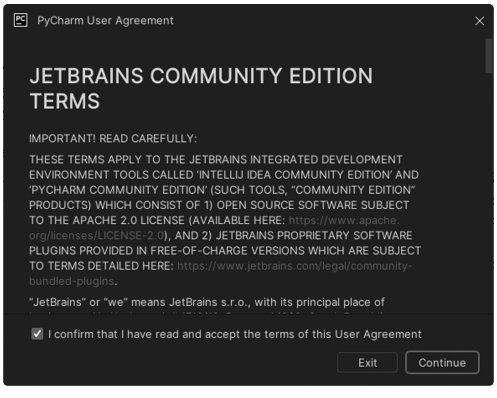


图 1-29　协议对话框

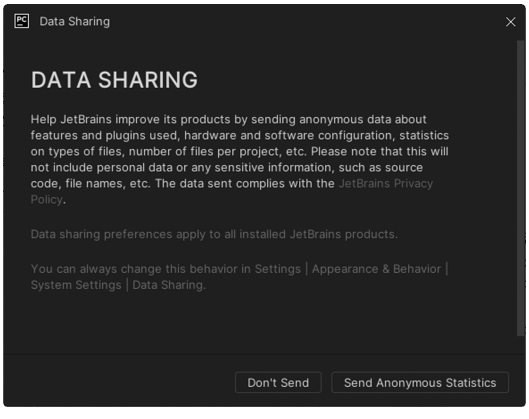


图 1-30　数据共享界面

PyCharm 激活之后会自动跳转到编辑界面，如图 1-31 所示。该界面包括 Projects(项目)、Customize(自定义)、Plugins(插件)、Learn(学习文档)四个选项卡。

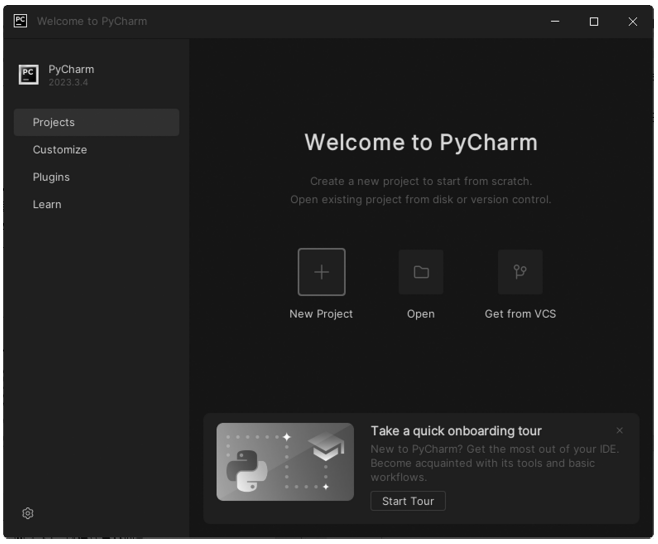


图 1-31　编辑界面

单击“Customize”(自定义)选项卡，在如图 1-32 所示的界面可以设置编辑区参数。

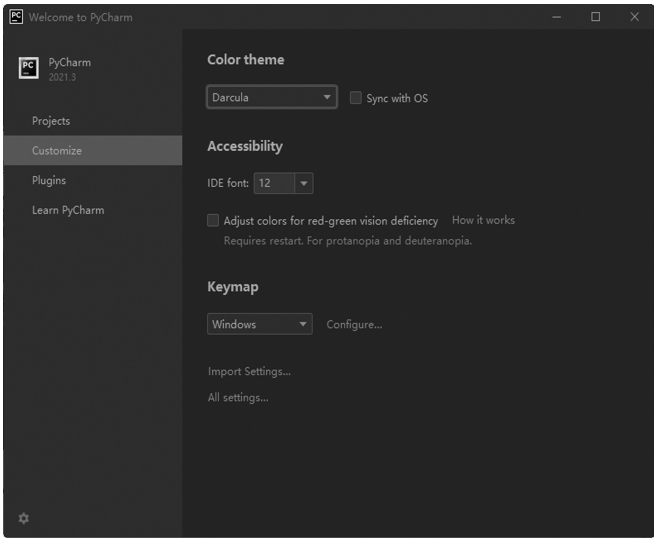


图 1-32　“Customize(自定义)”选项卡

在“Color theme”(颜色主题)选项下拉列表中显示几个主题。

- Dark：提供黑色主题界面，一般默认选择该项。
- Light：白色主题。
- Light with Light Header：白色高亮主题。
- Darcula：黑色经典主题，比 Dark(黑色主题)颜色要浅。
- High contrast：黑色高对比度主题。
- IntelliJ Light：更明亮的白色主题，如图 1-33 所示。本书后续的编辑器界面将采用白

色主题进行讲解。

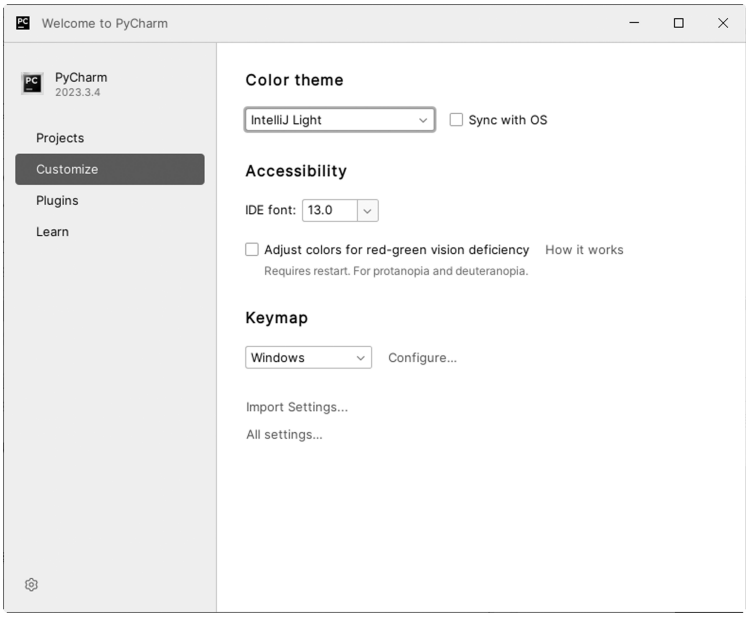


图 1-33　IntelliJ Light 主题

在“Accessibility”(辅助功能)选项组中可以看到“IDE font”(开发环境字体)默认为 12，用户可以根据需要修改字体。本书将字体大小设置为 14.0 磅。

在 Keymap(键位图)选项组，可以根据需要配置键盘快捷键。

三、汉化 Pycharm

Pycharm 默认的是英文版，为了方便操作，可以将 pycharm 改为中文版，也就是所谓的汉化。

选择菜单栏中的“File”(文件)→“All settings”(所有设置)命令，弹出“settings”(设置)对话框，选择“plugins”(插件) 选项卡，在“Plugins”(插件)文本框内输入 Chinese，单击右侧的“Marketplace”(市场)按钮，在左侧列表中选中“Chinese(Simplifiled)Language Pack”(中文语言包)，如图 1-34 所示。

单击“Install”(安装)按钮，当下载完成后，该按钮会变为“Restart IDE”(重启)按钮，如图 1-35 所示，单击该按钮，弹出“询问”对话框，如图 1-36 所示；单击“Restart”(重启)按钮，重启 PyCharm 之后，所有的菜单将变为中文，如图 1-37 所示。

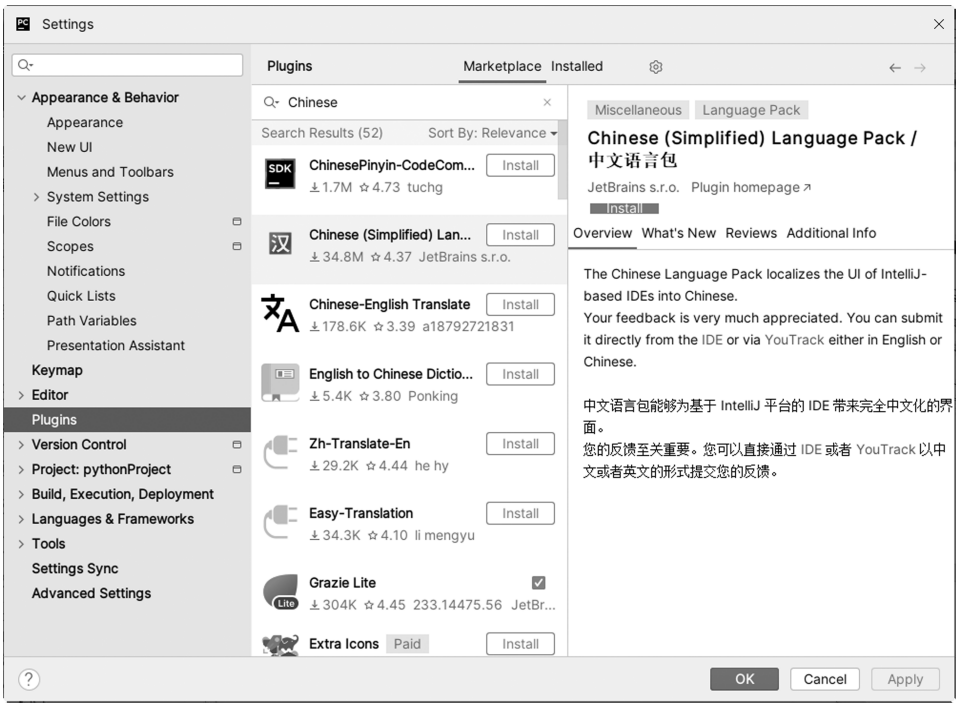


图 1-34 “Settings”(设置)对话框

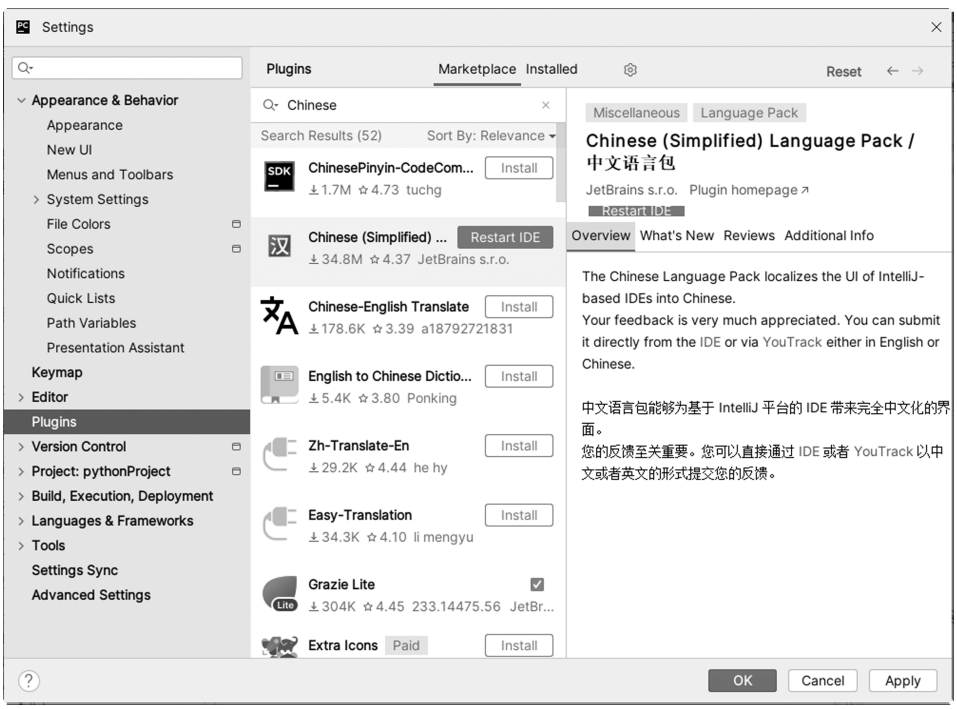


图 1-35 下载中文语言包

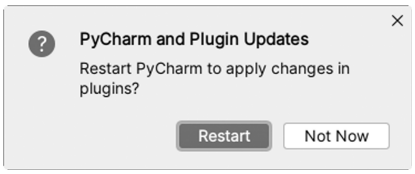


图 1-36　询问对话框



图 1-37　中文版用户界面

四、PyCharm 的编辑环境

在“项目”选项卡中单击“新建项目”按钮，在弹出的“新建项目”对话框中设置项目路径和名称，如图 1-38 所示。

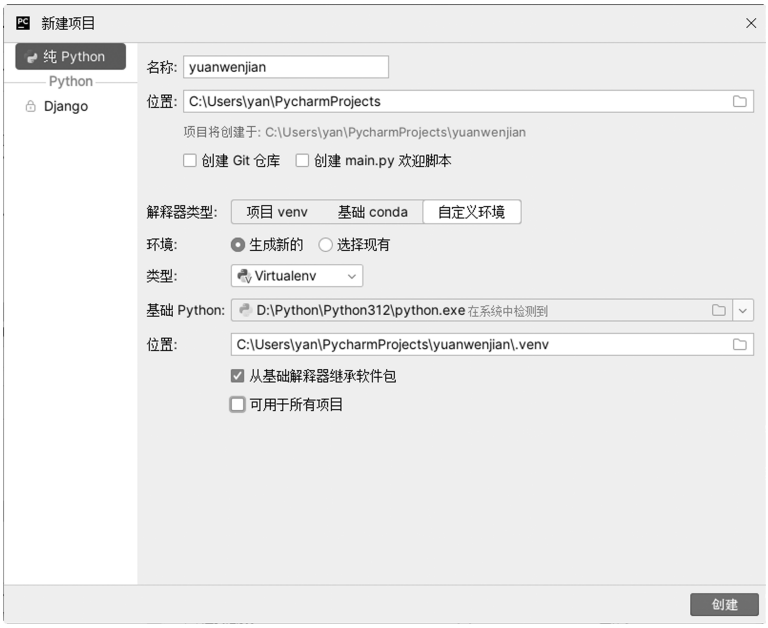


图 1-38　“新建项目”对话框

在“解释器类型”选项下，选择新项目的解释器环境：项目 venv、基础 conda、自定义环境。

(1)“项目 venv”：选择该虚拟环境，选择系统里面安装的 Python 作为解释器，在“Python 版本”下选择版本号，如图 1-39 所示。该虚拟环境是专门针对当前项目创建 venv 环境，用 pip 安装的包都安装到了 venv 环境(文件夹)下，系统 Python 环境不受任何影响。

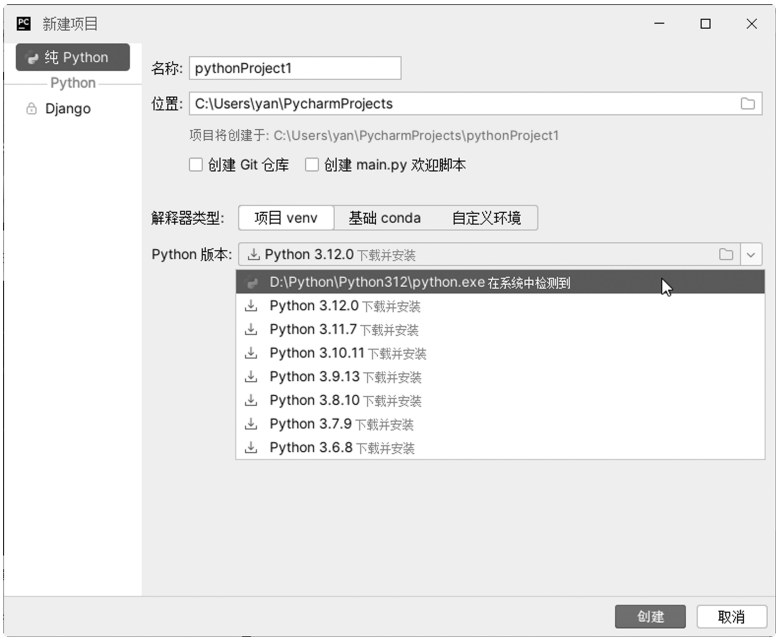


图 1-39 “项目 venv”类型

(2)“基础 conda”：如果安装了 Anaconda 软件，也可以使用 Anaconda 里面附带安装的 Python 解释器，其与使用本地解释器类似，如图 1-40 所示。

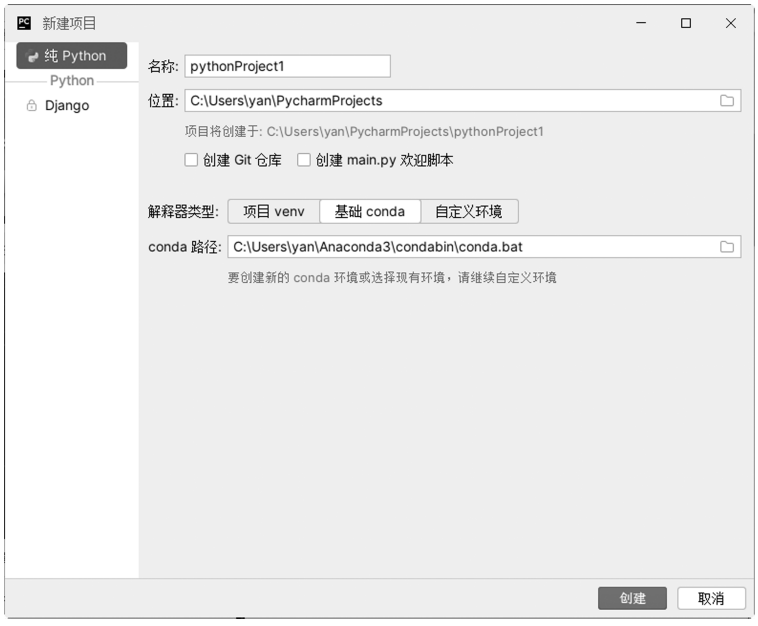


图 1-40 “基础 conda”类型

(3)“自定义环境”：创建虚拟环境，如图 1-41 所示。

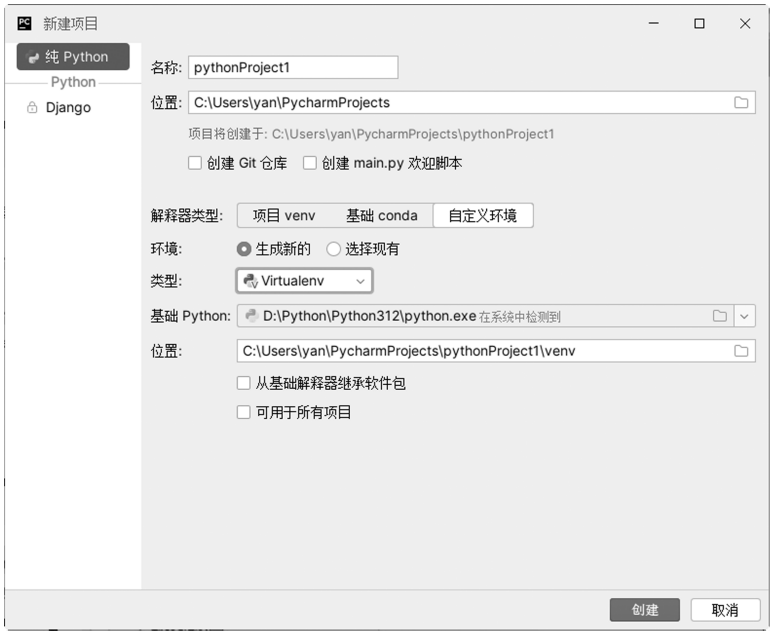


图 1-41 “自定义环境”解释器

1)生成新的：创建新的环境。

①“类型”下拉列表中显示四种解释器类型：

- Virtualenv：默认选择该选项，在项目文件夹下创建一个独立的虚拟环境。
- Conda：选择该选项，在项目文件夹下创建一个使用 Anaconda 的环境。
- Pipenv：是一个 Python 包管理工具，可选择 Python 版本，更适合简单的 Python 项目。
- Poetry：是一个 Python 包管理和打包的工具，提供了打包发布的功能。

②从基础解释器继承软件包：勾选该复选框，在 Virtualenv 中创建的解释器中安装全局 Python 中的所有软件包。否则，在 Virtualenv 中创建的解释器仅具有基本的工具，例如 pip 和 setuptools。

③可用于所有项目：勾选该复选框，将上面的参数设置应用到所有的项目中。

2)选择现有：直接选择已存在的本地解释器，直接采用 Anaconda 下的 Python 解释器，这样可以直接使用本地通过 pip 安装完成的各种安装包。

单击“创建”按钮，不仅新建了一个项目目录，而且配置好了开发环境。打开 PyCharm 编辑器界面，在左侧的“项目”导航器中将显示创建的项目，如图 1-42 所示。



图 1-42 PyCharm 编辑器界面

五、创建 Python 文件

创建 PyCharm 项目文件后，还需要创建 Python 文件，才可以进入 PyCharm 与 Python 交互代码的编译界面。

选择菜单栏中的“文件”→“新建”命令，或在“项目”面板上单击鼠标右键，在弹出的快捷菜单中选择“新建”命令，弹出如图 1-43 所示的子菜单。

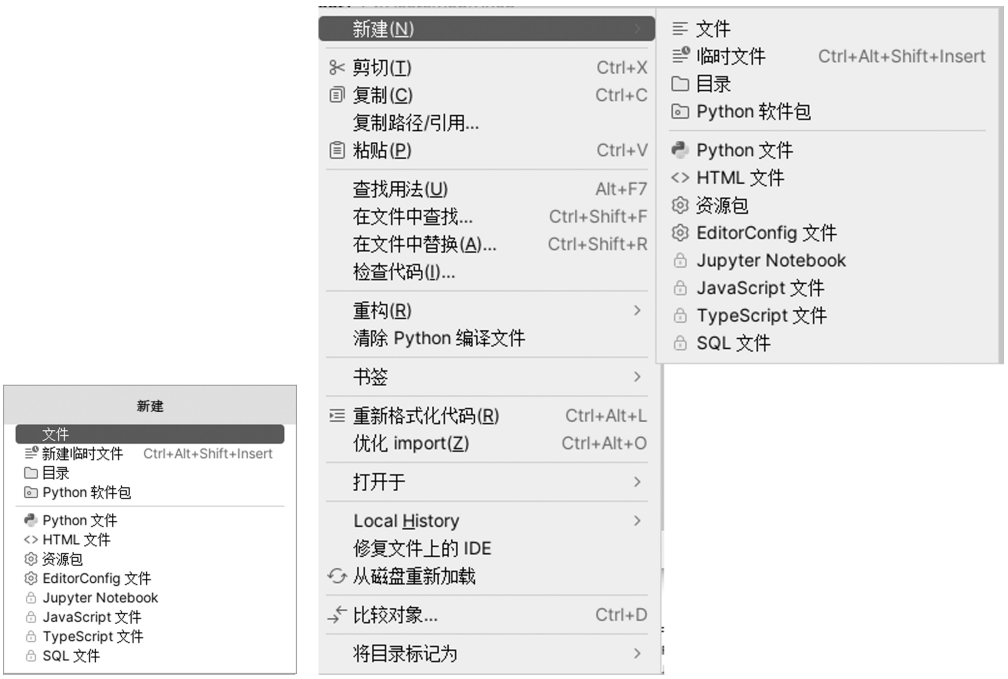


图 1-43 新建文件类型

Python 文件类型主要分为以下几种：

- 新建临时文件：当需要临时存储数据时，可以创建临时文件，这些文件位于单独的目录中。
- 目录：新建一个文件夹，用来存放不同用途的文件，如图 1-44 中的“supply”。Python 包的 Common 下固定会有一个 _ init _ . py 文件，而目录没有，这是包与目录的区别。
- Python 软件包：新建一个文件夹，包含用来归属分类不同功能作用的代码文件，如图 1-44 中的“Common”。
- Python 文件：源代码文件，以“. py”为扩展名，可以直接运行，不需要编译或连接。此外，还有以“. pyw”为扩展名的图形用户界面(GUI)文件，用于开发图形界面的桌面应用程序，由 Python w. exe 解释运行。
- XML 文件：XML 与 JSON 类似，也是一种数据交换格式，但它更加灵活。在 Python 中有内置的 xml 模块，可以用来读写 XML 文件。
- 资源包：创建一个新的模块。
- EditorConfig 文件：新建一个用于定义编码样式相关的配置文件。
- Jupyter Notebook：新建一个 Jupyter Notebook 文件。
- JavaScript 文件：创建一个 Python 脚本文件
- TypeScript 文件：创建一个类文件。
- SQL 文件：创建一个数据库文件。



图 1-44　新建目录和软件包

选择“Python 文件”命令，弹出“新建 Python 文件”对话框，新建的 Python 文件包含 Python 文件、Python 单元测试(文件)、Python 存根(文件)三种类型。

选择“Python 文件”选项，如图 1-45 所示，输入文件名称 Python file 01，按 Enter 键，进入 Python file 01. py 的编辑界面，才可以进入 PyCharm 与 Python 交互代码编译界面，如图 1-46 所示。

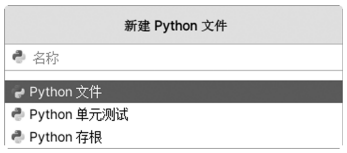


图 1-45 “新建 Python 文件”对话框

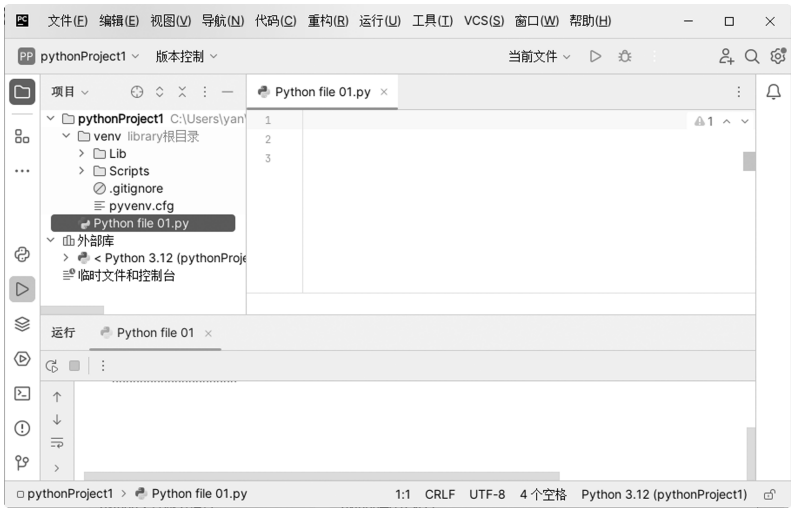


图 1-46 编辑界面

六、运行 Python 文件

在 PyCharm 中运行 Python 文件常用的方法有以下三种：

- 在 Python 文件的命令行窗口中，在需要运行的代码上用鼠标右键单击，从弹出的快捷菜单中选择命令“运行”；
- 单击菜单栏下方工具栏中的“运行”按钮 ▶ ；
- 按“Ctrl＋Shift＋F10”组合键。

如果打开了“运行”面板，单击该面板左侧工具栏中的“运行”按钮 ▶ ，也可以运行 Python 文件。

案例 2 ——PyCharm 项目文件

本实例在 PyCharm 中创建一个简单的 Python 文件并运行，帮助读者进一步熟悉 PyCharm 的工作环境，以及编辑、运行 Python 文件的操作方法。

【操作步骤】

1. 新建项目文件

双击桌面上的“PyCharm Community Edition 2023.3.4”图标，启动 PyCharm，自动打开“欢迎访问 PyCharm”窗口。

在“项目”选项卡中单击“新建项目”按钮＋，在弹出的“新建项目”对话框中选择“名称”文本框，在“名称”文本框内输入“Python Project1”，在“路径”文本框下选择项目路径，勾选“创建 Git 仓库”“创建 main.py 欢迎脚本”复选框，在“解释器类型”中选择“自定义环境”，“环境”为“生成新的”，“类型”为“Virtualenv”，“基础 Python ”默认为检测到的最新版本，并选择默认虚拟环境位置，勾选“从基础解释器集成软件包”复选框，如图 1-47 所示。

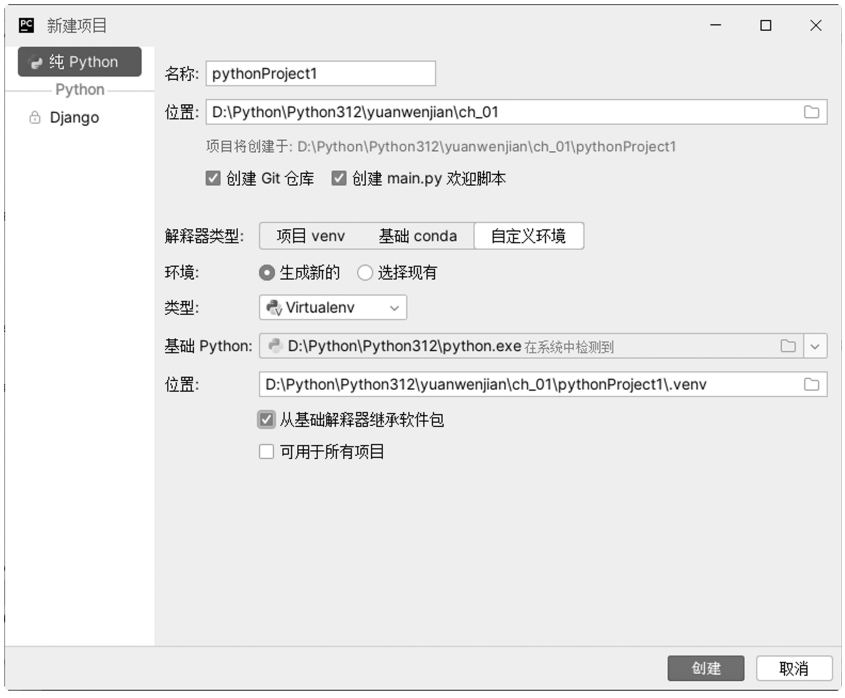


图 1-47 “新建项目”对话框

单击“创建”按钮，创建定义虚拟环境的项目文件“python Project1”，如图 1-48 所示。在右侧窗口中自动打开创建的“main.py”文件。同时，在源文件目录下创建文件夹 Python Project1，其中包含 .idea、.venv、main 这三个文件，如图 1-49 所示。

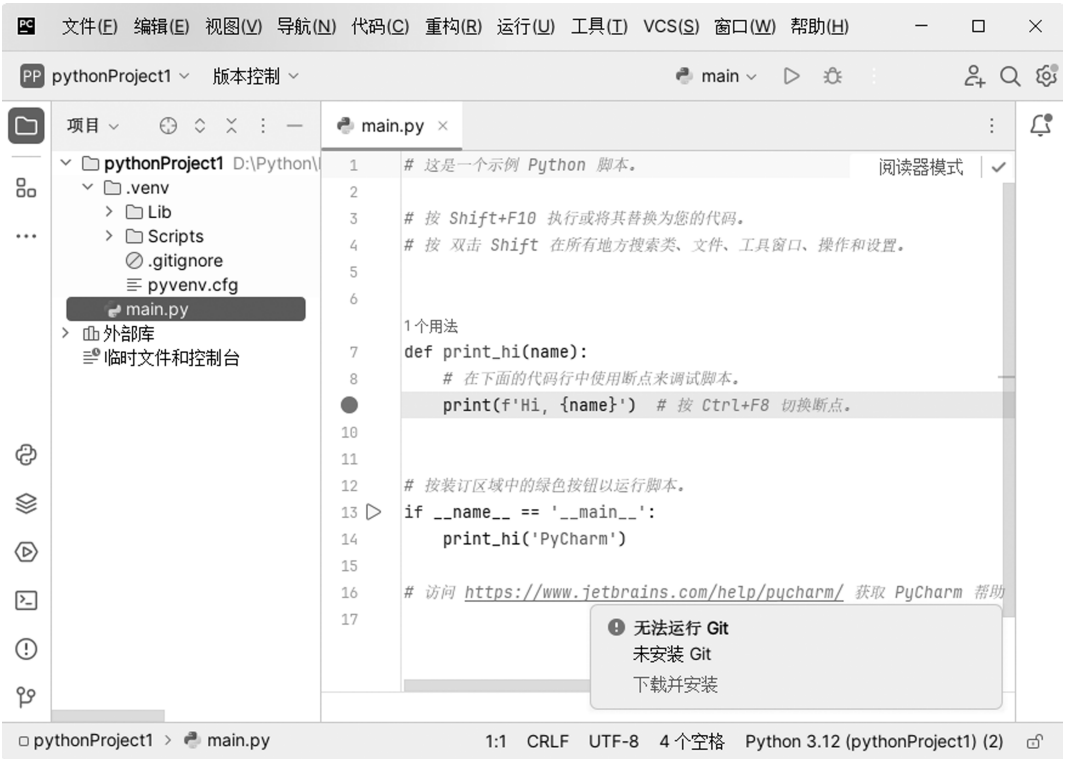


图 1-48 创建项目文件

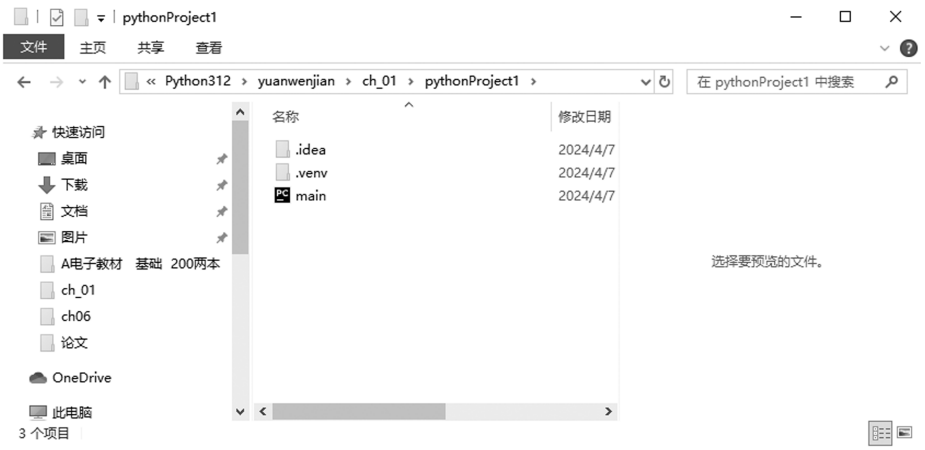


图 1-49 创建项目文件夹 Python Project1

2. 新建目录文件

在左侧“项目”面板中选中项目文件“Python Project1”，选择菜单栏中的“文件”→“新建”→“目录”命令，弹出“新建目录”对话框，输入“examole _ 1”，如图 1-50 所示。按 Enter 键，在当前项目文件下创建一个目录文件“examole _ 1”，如图 1-51 所示。



图 1-50 “新建目录”对话框

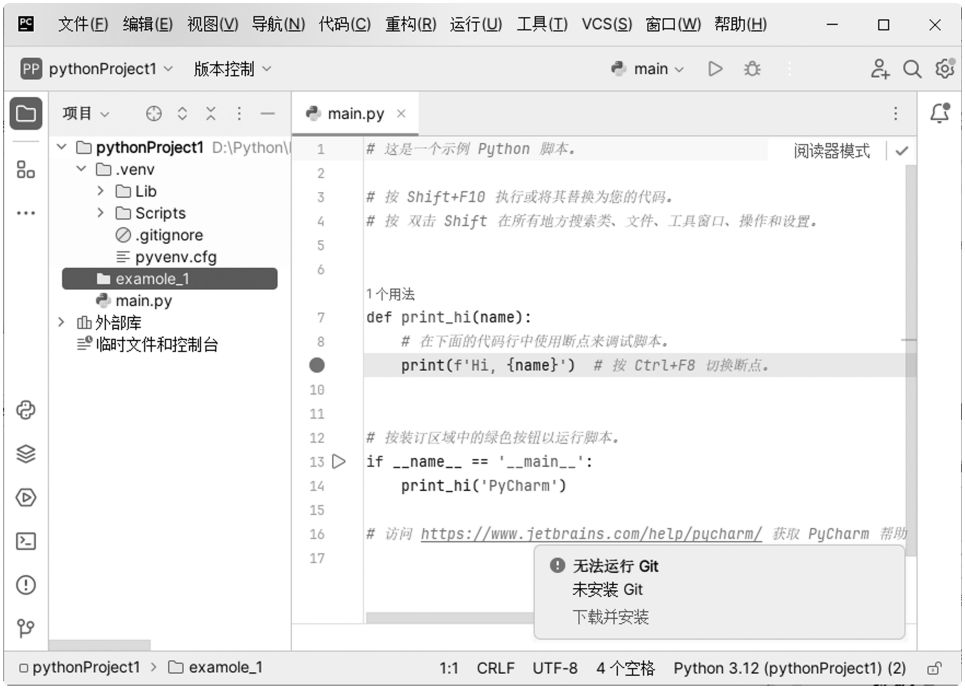


图 1-51　新建目录文件

3. 新建 Python 文件



图 1-52　“Python 文件”对话框

在左侧“项目”面板中选中 examole_1 目录。选择菜单栏中的“文件”→“新建”→“Python 文件”命令，弹出“Python 文件”对话框，在下面的选项中选择“Python 文件”，输入“named_2”，如图 1-52 所示。按下 Enter 键，在当前目录下创建一个名为“named_2.py”的 Python 文件，如图 1-53 所示。

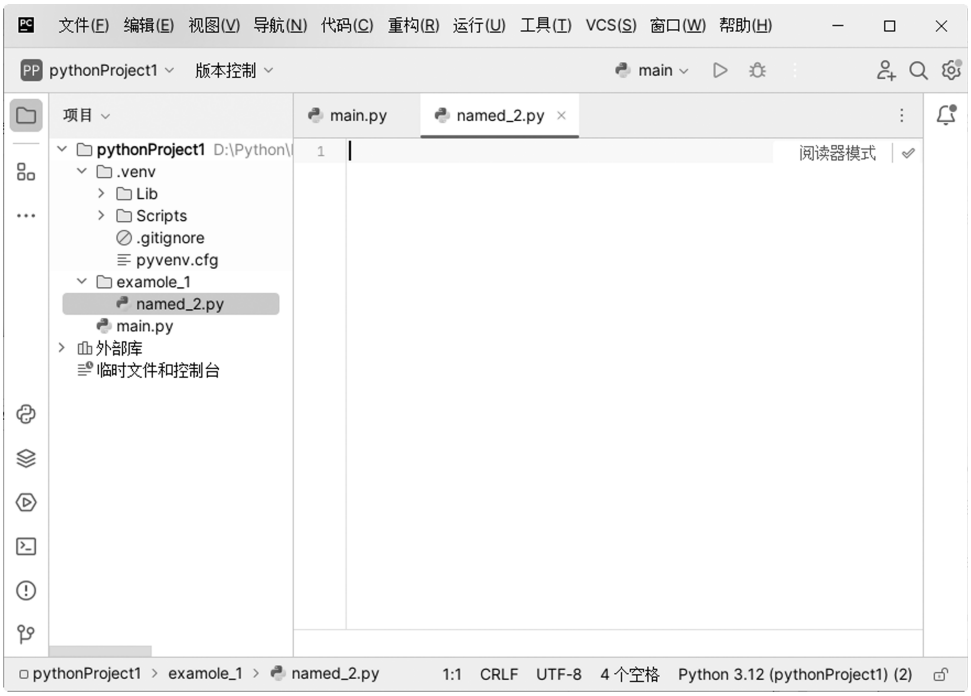


图 1-53　新建 Python 文件

4. 程序编写和运行

在命令行窗口中编写如下程序：

```
# 一个简单的 Python 文件
# 数据计算
a = 1  # 定义数值变量 a，并赋初值为 1
b = 2  # 定义数值变量 b，并赋初值为 2
# 输出变量值
print('a', a)
print('b', b)
print('a + b', a + b)
```

在命令行窗口顶部下拉列表中选择“当前文件”，即可打开“运行”面板，显示运行结果，如图 1-54 所示。

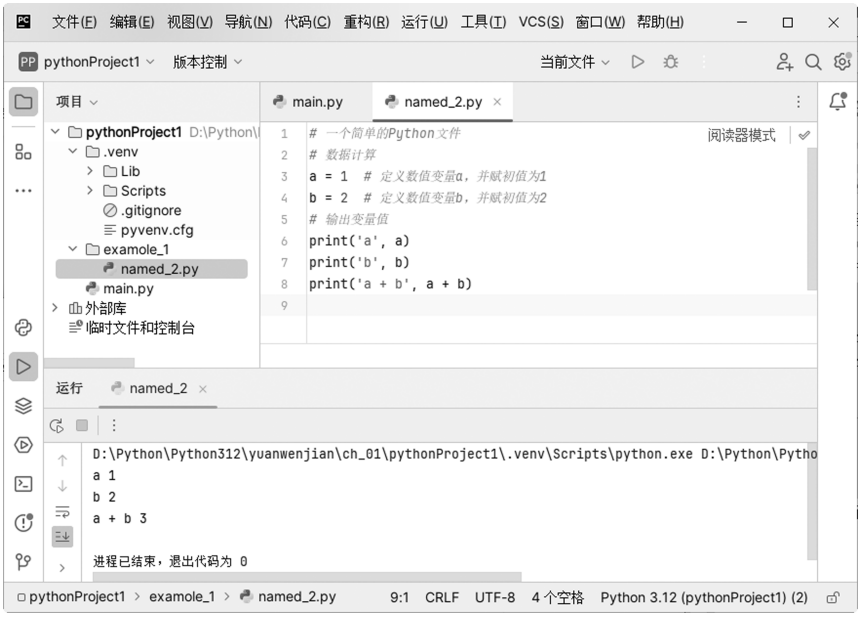


图 1-54　运行结果

任务 4　集成开发环境 VS Code

任务引入

小唐为了增加课堂作业的丰富性，使用多个编辑器来演示收集的调查报告来演示文稿，最终决定挑选 VS Code。那么，VS Code 究竟是一款什么样的编辑器？如何下载安装？界面是什么样的呢？

知识准备

Visual Stdio Code 简称“VS Code”，是一款跨平台的、免费且开源的现代轻量级代码编辑器，支持几乎所有的主流开发语言的语法高亮、智能代码补全、自定义快捷键、括号匹配和颜色区分、代码片段提示、代码对比等特性，也拥有对 git 开箱即用的支持。同时，它还支持插件扩展，通过丰富的插件，用户能获得更多高效的功能。

一、安装 VS Code

安装 VS Code 首先需要下载软件。

(1)登录 VS Code 的官网下载地址：https：//code. visualstudio. com/，如图 1-55 所示。

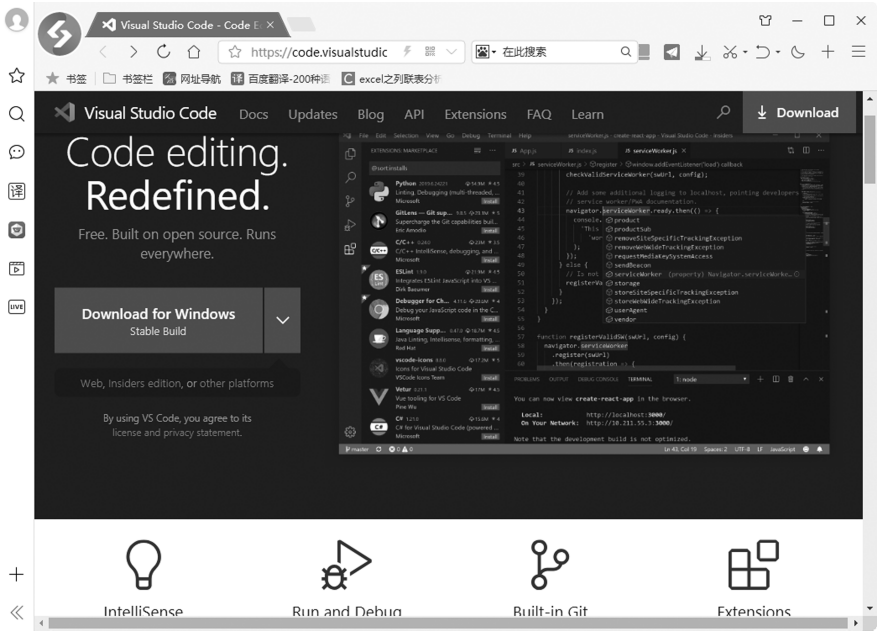


图 1-55　官网下载页面

(2)单击“Download”(下载)按钮，更新页面，根据计算机操作系统选择对应的下载程序，如图 1-56 所示。单击“Windows”下的“.zip”“x64”按钮，下载“VS Code-win32-x64-1.87.2.zip”文件。



图 1-56 选择下载版本

VS Code 安装很简单，下载 zip 解压后，双击解压文件 VS Code-win32-x64-1.87.2 中的“Code.exe”，如图 1-57 所示。直接打开用户界面，如图 1-58 所示。

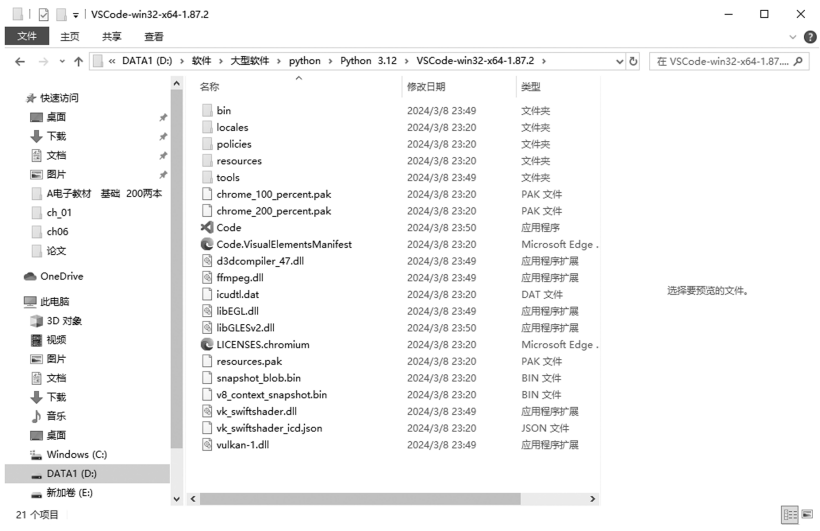


图 1-57 解压文件

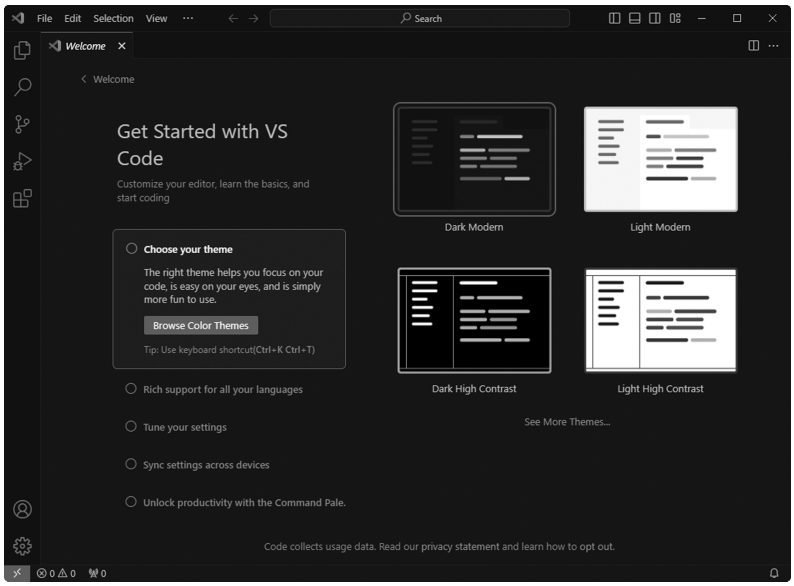


图 1-58　VS Code 用户界面

二、设置用户界面颜色

VS Code 用户界面默认提供黑色界面，为了提高使用舒适度，可以切换界面主题。

单击“Get Started with VS Code”选项卡“Choose your theme”(选择主题)下的“Browse Coler Themes”(查找颜色主题)按钮，在弹出的列表中显示颜色主题，如图 1-59 所示。选择“Light (Visual Studio)Visual Studio Light”(白色主题)选项，自动更换用户界面颜色主题，结果如图 1-60 所示。

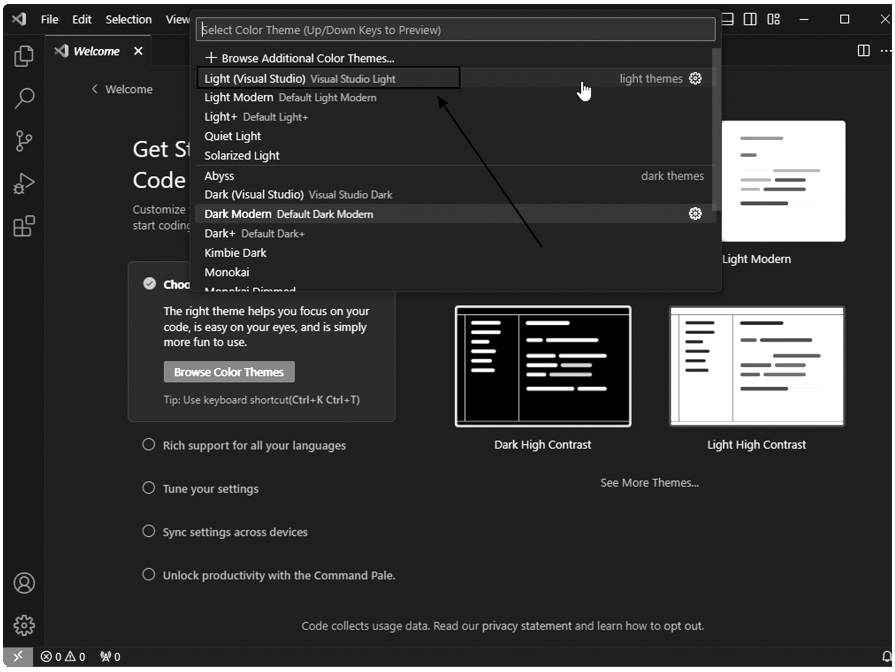


图 1-59　选择颜色主题