



新型工业化·新计算·计算机应用与技术类系列

COMPUTER APPLICATION

Python程序设计 实用教程

余 宏 张良均 / 主 编

谢彩云 曾小芹 汪振东 / 副主编
戴靓婕 李亦欣 白明明

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本零基础、多层次、重应用的 Python 编程教材。本书共 14 章，分为基础部分和进阶部分。基础部分（第 1 章～第 8 章）主要讲解 Python 的基础知识和应用，包括 Python 概述、Python 语言基础、程序的控制结构、函数、组合数据类型、模块、文件和数据格式化、典型 Python 模块的应用。进阶部分（第 9 章～第 14 章）主要讲解 Python 高级应用开发所需的知识和技术，包括面向对象编程、图形用户界面编程、多线程编程、数据库编程、网络编程和网络爬虫。

全书内容以应用为核心，每章内容都与应用案例紧密结合并配有上机实践，有助于读者理解知识、应用知识，提升通过 Python 编程解决实际问题的能力。

本书难度适中，可作为高等学校相关专业 Python 程序设计课程的教材，也可作为广大程序设计爱好者的自学参考书，还可作为参加全国计算机等级考试二级 Python 语言程序设计考试的辅导用书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Python 程序设计实用教程 / 余宏，张良均主编.
北京：电子工业出版社，2025. 1. -- ISBN 978-7-121-49372-0

I . TP312.8

中国国家版本馆 CIP 数据核字第 20240JR959 号

责任编辑：刘 瑀 文字编辑：孟泓辰

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1 092 1/16 印张：22 字数：619.52 千字

版 次：2025 年 1 月第 1 版

印 次：2025 年 1 月第 1 次印刷

定 价：64.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：menghc@phei.com.cn。

前 言

在信息化和数字化迅猛发展的今天，人工智能和大数据等新一代信息技术融合到各个领域，这些新技术和应用的核心就是程序。选择一门高级程序设计语言作为教学内容，介绍程序设计的基本方法，培养学生分析问题、利用计算机求解问题的思维方式和初步应用能力，满足信息社会各领域对计算机应用人才的需求，是高等学校计算机基础教育中程序设计课程的重要任务。

毋庸置疑，Python 是目前最适合编程入门的程序设计语言，它具有简单易学、免费开源、功能强大等特点，加上丰富的第三方库的支持，Python 已经成为人工智能学习的首选编程语言。

本书内容

基础部分包括第 1~8 章，主要讲解 Python 的基础知识和应用，包括 Python 概述、Python 语言基础、程序的控制结构、函数、组合数据类型、模块、文件和数据格式化、典型 Python 模块的应用，是程序设计入门学习的必备内容，以夯实程序设计基础、培养逻辑思维、提升编程能力为目标。

进阶部分包括第 9~14 章，主要讲解 Python 高级应用开发所需的知识和技术，包括面向对象编程、图形用户界面编程、多线程编程、数据库编程、网络编程和网络爬虫，很好地体现了计算思维的本质，并结合实际应用展示了 Python 的“计算生态”。

本书特点

(1) Python 程序设计涉及的范围非常广泛，本书内容编排并不求全、求深，而是考虑零基础读者的接受能力，语法介绍以够用和实用为原则，以实际应用为目的，讲解 Python 的基础知识及高级应用。在明晰基本理论的前提下，注重 Python 实际应用的讲解，而不是对技术进行深度挖掘，让读者对 Python 的体系架构有全面的认识。

(2) 本书在讲解时采用理论与实践应用相结合的方式，针对每章重要的知识点，配有兼具实用性与趣味性的应用案例，方便读者更好地将理论知识应用于实际场景中，加深对知识的理解和掌握。还通过场景化的上机实践，提升读者通过编程解决实际问题的能力。

(3) 本书在编写的过程中，将知识传授与思想政治教育相融合，在应用案例中融入课程思政内容，激发学生建设科技强国的责任担当，提升学生的职业素养，落实德才兼备的高素质卓越工程师和高技能人才的培养要求。

本书适用对象

本书既可作为高等学校相关专业 Python 程序设计课程的教材，也可作为广大程序设计爱好者的自学参考书。本书覆盖全国计算机等级考试二级 Python 语言程序设计考试大纲的大部分内容，还可作为该考试的辅导用书。

代码下载及问题反馈

全书由余宏、张良均任主编并负责统稿，谢彩云、曾小芹、汪振东、戴靓婕、李亦欣、白

明明任副主编,其中第1、2章由白明明编写,第3、4章由曾小芹编写,第5、12章由戴靓婕编写,第6、9章由谢彩云编写,第7、8章由李亦欣编写,第10、11章由余宏编写,第13、14章由汪振东编写;全书的应用案例(除第13章外)由余宏编写,上机实践(除第10、11、13章外)由张良均编写。豫章师范学院为本书的第一编写单位,广东泰迪智能科技有限公司为参编单位。

为了帮助读者更好地使用本书,本书提供原始数据文件、Python 程序代码,以及 PPT 课件、习题和上机实践参考答案等教学资源,读者可以扫描封底二维码下载。

尽管在本书的编写过程中,我们力求严谨,但由于技术的发展日新月异,加之水平有限,时间紧迫,书中难免存在不足之处,敬请广大读者批评指正。如果您有更多的宝贵意见,欢迎在泰迪学社微信公众号(TipDataMining)回复“图书反馈”进行反馈,也可联系邮箱154945907@qq.com。更多本系列图书的信息可以在泰迪云教材网站查阅。

编 者

2024 年 7 月

电子工业出版社版权所有
盗版必究

目 录

第 1 章 Python 概述	001	2.5.4 应用案例 2: 文本 进度条	030
1.1 Python 的特点及应用	002	2.6 运算符	031
1.1.1 Python 的特点	002	2.6.1 算术运算符	031
1.1.2 Python 的应用	002	2.6.2 比较运算符	032
1.2 Python 环境配置	002	2.6.3 赋值运算符	032
1.2.1 安装 Python 解释器	003	2.6.4 逻辑运算符	033
1.2.2 运行 Python 程序	006	2.6.5 位运算符	034
1.3 集成开发环境 PyCharm 的 安装及使用	008	2.6.6 运算符优先级	034
1.3.1 PyCharm 的下载与安装	008	2.6.7 应用案例 3: 汇率转换	035
1.3.2 PyCharm 的使用	011	2.7 上机实践: 过滤停用词	036
小结	015	小结	037
习题	015	习题	037
第 2 章 Python 语言基础	017	第 3 章 程序的控制结构	039
2.1 代码风格	018	3.1 程序的顺序结构	040
2.2 变量	018	3.2 程序的选择结构	040
2.2.1 变量命名的限制	018	3.2.1 单分支结构	040
2.2.2 变量赋值	019	3.2.2 双分支结构	041
2.3 数据类型	019	3.2.3 多分支结构	042
2.4 基本输入/输出	022	3.2.4 应用案例 1: 航空 会员等级划分	043
2.4.1 input() 函数	022	3.3 程序的循环结构	044
2.4.2 eval() 函数	022	3.3.1 遍历循环: for 语句	044
2.4.3 print() 函数	023	3.3.2 无限循环: while 循环	046
2.4.4 转义字符	023	3.3.3 循环保留字: break 语句和 continue 语句	047
2.4.5 应用案例 1: 关注 个人健康管理	024	3.3.4 应用案例 2: 密码 强度检查	050
2.5 字符串	025	3.4 程序的异常处理	053
2.5.1 字符串的格式化输出	025	3.4.1 try...except 语句	053
2.5.2 字符串的常见操作	028	3.4.2 else 子句	056
2.5.3 字符串的索引和切片	029	3.4.3 异常处理 finally 子句	057

3.5 上机实践: DNA 序列分析器 ..	058	5.2.2 列表元素的访问	084
小结	060	5.2.3 列表元素的添加、 修改和删除	086
习题	060	5.2.4 列表的内置函数与 常见方法	089
第 4 章 函数	062	5.2.5 嵌套列表	091
4.1 函数引入的意义	063	5.2.6 元组的创建与元素访问 ..	092
4.2 函数的定义与调用	063	5.3 字典	093
4.2.1 函数的定义	063	5.3.1 字典的创建	093
4.2.2 函数的调用	064	5.3.2 字典的访问与更新	094
4.3 函数的参数传递	065	5.3.3 字典的方法	095
4.3.1 默认参数	066	5.4 集合	097
4.3.2 位置参数	066	5.4.1 集合的创建	097
4.3.3 关键字参数 (命名参数)	067	5.4.2 集合的访问与更新	098
4.3.4 可变参数 (打包参数) ..	067	5.4.3 应用案例: 学生 模拟选课系统	100
4.4 函数的返回值	068	5.5 上机实践: 智慧菜篮	104
4.5 变量的作用域	069	小结	108
4.5.1 局部作用域与局部变量 ..	070	习题	108
4.5.2 全局作用域与全局变量 ..	070	第 6 章 模块	110
4.6 函数的特殊形式	072	6.1 模块概述	111
4.6.1 lambda 表达式及 匿名函数	072	6.1.1 相关概念	111
4.6.2 递归函数	073	6.1.2 模块的导入方式	111
4.6.3 应用案例: 递归函数 应用——失之毫厘, 谬以千里	074	6.2 自定义模块	112
4.7 上机实践: 学生成绩管理系统 ..	075	6.3 模块的导入特性	114
小结	078	6.3.1 __all__ 属性	114
习题	078	6.3.2 __name__ 属性	115
第 5 章 组合数据类型	081	6.4 Python 中的包	116
5.1 组合数据类型概述	082	6.4.1 包的结构	116
5.1.1 序列类型	082	6.4.2 包的导入	116
5.1.2 集合类型	082	6.4.3 常用的标准模块	117
5.1.3 映射类型	082	6.5 第三方库的下载与安装	119
5.2 列表与元组	083	6.6 应用案例	120
5.2.1 列表的创建与删除	083	6.6.1 应用案例 1: 随机生成验证码	120
		6.6.2 应用案例 2: 考试倒计时器	121

6.7 上机实践：猜数字赢奖品	125	8.2.2 应用案例 2：文本 词频统计	157
小结	127	8.3 wordcloud 模块	159
习题	127	8.3.1 wordcloud 模块解析	159
第 7 章 文件和数据格式化	129	8.3.2 应用案例 3：绘制 《三国演义》出场 人物词云图	160
7.1 文件概述	130	8.4 NumPy 库	162
7.2 文件的基本操作	130	8.4.1 NumPy 库解析	162
7.2.1 文件的打开和关闭	130	8.4.2 应用案例 4：绘制 正弦波与余弦波的 对比图形	166
7.2.2 从文件中读取数据	132	8.5 Matplotlib 库	168
7.2.3 向文件写入数据	133	8.5.1 pyplot 模块解析	168
7.2.4 文件的定位读写	134	8.5.2 应用案例 5：绘制 短跑运动员技能 评估雷达图	174
7.2.5 应用案例 1： 文件备份	136	8.6 上机实践：统计《西游记》 出场次数前 10 名的人物	176
7.3 目录操作	137	小结	178
7.3.1 创建目录	137	习题	178
7.3.2 删除目录	138	第 9 章 面向对象编程	180
7.4 二维数据的格式化与处理	139	9.1 面向对象的基本概念	181
7.4.1 csv 文件的读写	139	9.2 类与对象	181
7.4.2 Excel 文件（XLSX） 的读写	140	9.2.1 类的定义	182
7.5 高维数据的格式化与处理	143	9.2.2 对象的创建与使用	183
7.5.1 JSON 模块概述	143	9.2.3 类的属性	184
7.5.2 JSON 模块解析	143	9.2.4 实例方法	184
7.5.3 应用案例 2： 文件保存	145	9.2.5 类成员的访问限制	186
7.6 上机实践：计算 csv 文件 中的数据平均值	147	9.3 构造方法与析构方法	188
小结	149	9.3.1 构造方法	188
习题	149	9.3.2 析构方法	189
第 8 章 典型 Python 模块的应用	151	9.4 类方法和静态方法	190
8.1 turtle 模块	152	9.4.1 类方法	190
8.1.1 turtle 模块解析	152	9.4.2 静态方法	191
8.1.2 应用案例 1：物体的 自由落体运动	153	9.5 应用案例：超市管理系统	192
8.2 jieba 库	155		
8.2.1 jieba 库解析	155		

9.6 类的继承.....	196	10.6 菜单.....	233
9.6.1 单继承.....	196	10.7 消息对话框.....	236
9.6.2 多继承.....	196	10.8 应用案例：制作简易计算器...	237
9.6.3 方法的重写.....	197	10.9 上机实践：求解线性方程组...	238
9.7 多态.....	199	小结.....	241
9.8 应用案例 2：太空探索游戏.....	200	习题.....	242
9.9 上机实践.....	204		
9.9.1 实验 1：学生 成绩更新.....	204	第 11 章 多线程编程	244
9.9.2 实验 2：动物乐园.....	206	11.1 进程与线程.....	245
小结.....	209	11.1.1 进程的概念.....	245
习题.....	209	11.1.2 线程的概念.....	245
第 10 章 图形用户界面编程	211	11.2 Python 与多线程.....	246
10.1 tkinter 概述.....	212	11.2.1 单线程程序与 多线程程序.....	247
10.1.1 初识 tkinter.....	212	11.2.2 线程的生命周期.....	249
10.1.2 构建简单的 GUI 程序.....	212	11.2.3 线程的创建和启动.....	249
10.2 tkinter 组件概述.....	215	11.3 线程同步.....	252
10.2.1 tkinter 的核心组件.....	215	11.3.1 多线程使用不当 造成的数据混乱.....	252
10.2.2 组件的通用属性.....	216	11.3.2 利用 Lock 实现 线程同步.....	254
10.3 常用组件.....	218	11.4 应用案例：幸运抽奖大转盘...	260
10.3.1 标签 Label 和 按钮 Button.....	218	11.5 上机实践：多线程模拟 多窗口航班售票.....	264
10.3.2 文本框 Entry 和 文本域 Text.....	219	小结.....	265
10.3.3 复选框 Checkbutton 和单选按钮 Radiobutton.....	222	习题.....	266
10.4 布局管理.....	225	第 12 章 数据库编程	267
10.4.1 pack 布局管理器.....	225	12.1 数据库基础.....	268
10.4.2 grid 布局管理器.....	226	12.2 SQL 基础.....	269
10.4.3 place 布局管理器.....	227	12.2.1 数据库与表的创建.....	270
10.4.4 使用 Frame 的 复杂布局.....	228	12.2.2 插入、修改和 删除表中的数据.....	271
10.5 事件处理.....	229	12.2.3 数据查询.....	272
10.5.1 command 事件 处理方式.....	229	12.3 MySQL 数据库.....	274
10.5.2 bind 事件处理方式.....	230	12.3.1 MySQL 数据库的 下载与安装.....	275

12.3.2	MySQL 服务器的 启动、关闭和登录	276	13.5.3	多线程并发服务器	310
12.4	Python 的 MySQL 编程	279	13.6	I/O 多路转接服务器	311
12.4.1	安装 pymysql 模块	279	13.6.1	select 并发服务器	311
12.4.2	访问 MySQL 数据库的步骤	279	13.6.2	epoll 并发服务器	312
12.4.3	创建数据库和表	281	13.7	上机实践：一对一聊天程序	313
12.4.4	数据库的插入、 更新、删除和查询	283	小结		315
12.5	应用案例：用户注册登录	287	习题		315
12.6	上机实践：设计学生 信息管理系统	291			
小结		295			
习题		296			
第 13 章	网络编程	297	第 14 章	网络爬虫	317
13.1	网络概述	298	14.1	Web 的基础知识	318
13.1.1	网络体系结构	298	14.1.1	HTTP	318
13.1.2	网络数据传输流程	298	14.1.2	HTML 简介	320
13.1.3	网络应用架构	299	14.1.3	CSS 简介	321
13.1.4	IP 地址和端口号	299	14.1.4	JavaScript 简介	323
13.2	socket 网络编程基础	300	14.1.5	Web 服务器	324
13.2.1	socket() 方法	301	14.2	爬虫概述	325
13.2.2	socket 通信流程	302	14.3	抓取网页数据	326
13.2.3	socket 内置方法	303	14.3.1	requests 库介绍	326
13.2.4	应用案例：扫描 开放端口	304	14.3.2	requests 库抓取网页	327
13.3	基于 UDP 的网络聊天通信	305	14.4	解析网页数据	329
13.4	基于 TCP 的数据转换	306	14.4.1	网页数据结构分析	329
13.5	TCP 并发服务器	308	14.4.2	BeautifulSoup 库简介	331
13.5.1	单进程非阻塞服务器	308	14.4.3	BeautifulSoup 库 解析网页数据	331
13.5.2	多进程并发服务器	309	14.4.4	应用案例：爬取 某平台课程数据	333
			14.5	上机实践：爬取某网站的 优秀作品信息	336
			小结		339
			习题		339
			参考文献		340

第 1 章

Python 概述

Python 是一种功能强大的编程语言，具有诸多优势，包括易于学习和使用、有强大的社区支持和丰富的库等。本章将介绍 Python 的特点及应用、Python 环境配置、PyCharm 的安装及使用等。

学习目标

- (1) 了解 Python 的特点及应用。
- (2) 掌握 Python 的安装及配置。
- (3) 熟练掌握 Python 的程序运行。
- (4) 熟练掌握 PyCharm 的使用。

1.1 Python 的特点及应用

Python 是一种解释型、面向对象的高级程序设计语言。Python 的语法规则相对简单，使得程序代码清晰、简洁。

1.1.1 Python 的特点

Python 的特点如下。

(1) 简单易学：Python 的语法简洁，非常接近自然语言，它仅需要少量关键字就可识别条件、循环、函数等程序结构。与其他编程语言相比，Python 可以使用更少的代码实现相同的功能。

(2) 面向对象：Python 支持面向对象编程，可以使用类和对象来组织和管理代码。面向对象的编程范式使得代码更加模块化、可重用和易于维护。

(3) 丰富的库：Python 拥有丰富的内置标准库和第三方库，可以帮助开发人员快速、高效地处理各种工作。

(4) 可移植性：Python 作为一种解释型语言，可以在任何安装了 Python 解释器的环境中执行，因此使用 Python 开发的程序具有良好的可移植性。

(5) 免费开源：Python 是开源的，并且有一个庞大而活跃的开发者社区，提供了丰富的资源和支持。

1.1.2 Python 的应用

Python 被广泛应用于许多领域，主要的应用领域如下。

(1) Web 开发：Python 具有丰富的 Web 框架，如 Django 和 TurboGears，可以用于开发 Web 应用、网站、API 等。许多大型网站都是使用 Python 开发的，如 Google、豆瓣、YouTube 等。

(2) 数据科学：Python 具有各种数据科学工具和库，如 NumPy、pandas、scikit-learn 等，可以用于数据分析、数据建模、机器学习等。

(3) 人工智能：Python 是人工智能领域的主要编程语言，应用领域涵盖自然语言处理、图像处理、语音识别、深度学习等。

(4) 自动化运维和测试：Python 是编写自动化工具和脚本的主要语言之一，可以用于自动化测试、系统监测、数据采集等。

(5) 游戏开发：Python 在游戏开发领域有着广泛应用，提供 Pygame 和 Panda3D 等库。

1.2 Python 环境配置

Python 环境配置通常指的是安装 Python 解释器和管理第三方库。为了方便开发，还可以安装一些集成开发环境来编写和运行 Python 代码。常见的集成开发环境有 PyCharm、

VS Code 等。

1.2.1 安装 Python 解释器

安装 Python 解释器通常是一个相对简单的过程，以下是针对 Windows 系统的 Python 安装步骤。

1. 下载 Python

在 Python 官网可以下载 Python 解释器以搭建 Python 开发环境。具体操作步骤如下。

- (1) 打开官网，进入页面后单击导航中的“Downloads”。
- (2) 选择“Downloads”菜单下的“Windows”命令，如图 1-1 所示。

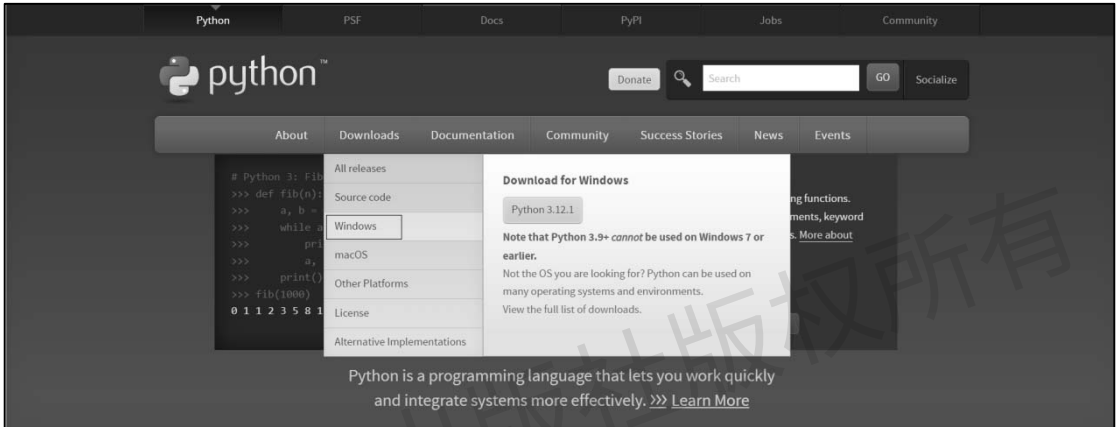


图 1-1 Python 官网

(3) 找到 Python 3.12.1 安装包，如果 Windows 系统版本是 32 位的，那么单击“Windows installer (32-bit)”超链接，然后下载；如果 Windows 系统版本是 64 位的，那么单击“Windows installer (64-bit)”超链接，然后下载，如图 1-2 所示。

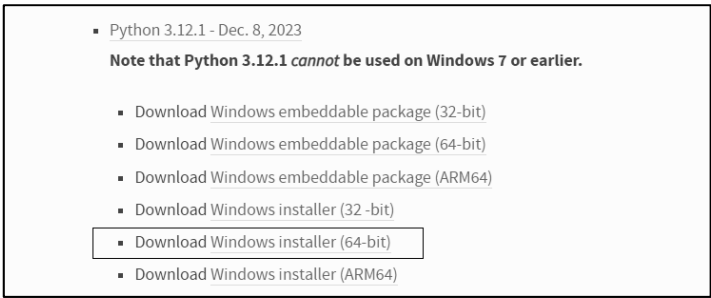


图 1-2 Python 3.12.1 安装包

2. 安装 Python

安装 Python 的步骤如下。

- (1) 下载完成后，双击所下载的文件，打开安装向导窗口。
- 安装界面显示了两种安装方式，分别为“Install Now”和“Customize installation”。其中，

“Install Now”为默认安装方式，“Customize installation”为自定义安装方式。

此外，安装界面下方有一个“Add Python.exe to PATH”选项，若勾选此复选框，则会将 Python 解释器的安装路径自动添加到环境变量中；若不勾选此复选框，则在使用 Python 解释器之前需要手动将 Python 解释器的安装路径添加到环境变量中。

(2) 勾选“Add Python to PATH”复选框，单击“Customize installaion”按钮，如图 1-3 所示。



图 1-3 Python 安装向导窗口

(3) 默认勾选图 1-4 中的所有功能，这些功能的相关介绍如下。

①Documentation: Python 帮助文档，其目的是帮助开发者查看 API 和相关说明。

②pip: Python 包管理工具，提供了查找、下载、安装、卸载 Python 包的功能。

③td/tk and IDLE: tk 是 Python 的标准图形用户界面接口，IDLE 是 Python 自带的简洁的集成开发环境。

④Python test suite: Python 标准库测试套件。

⑤py launcher: 安装 Python Launcher 后可以通过全局命令 py 方便地启动 Python。

⑥for all users: 适用所有用户使用。

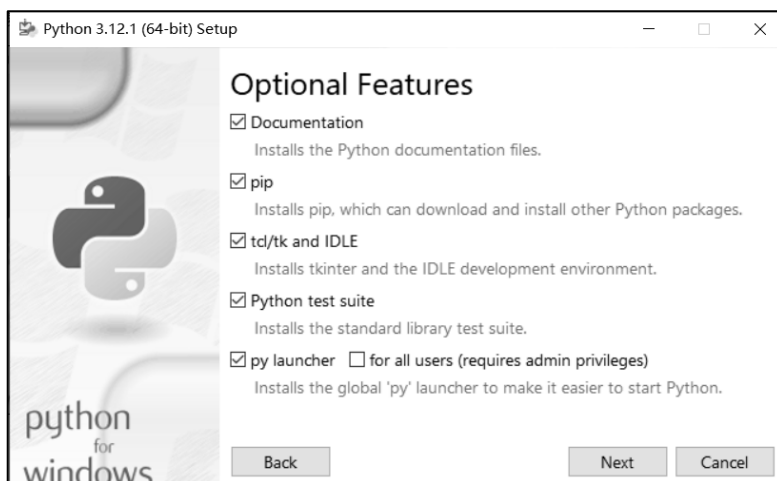


图 1-4 Python 可选功能界面

(4) 保持默认配置，单击“Next”按钮进入设置高级选项的界面，用户在该界面中依然可以根据自身需求勾选功能，并设置 Python 安装路径，具体如图 1-5 所示。

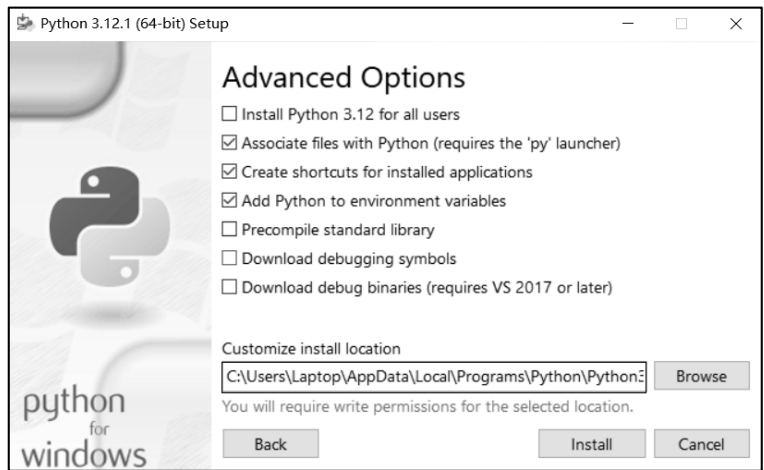


图 1-5 勾选功能，设置安装路径

(5) 选定好 Python 的安装路径后，单击“Install”按钮开始安装，安装成功后如图 1-6 所示。

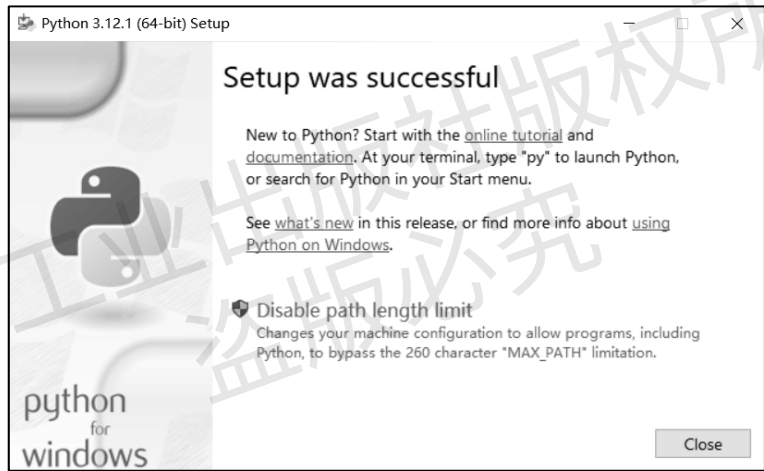


图 1-6 Python 安装成功

至此，Python 安装完成，可以使用 Windows 系统中的命令提示符检测 Python 是否安装成功。在 Windows 系统中打开命令提示符窗口，输入“Python”后按回车键将显示 Python 的版本信息，表明安装成功，如图 1-7 所示。

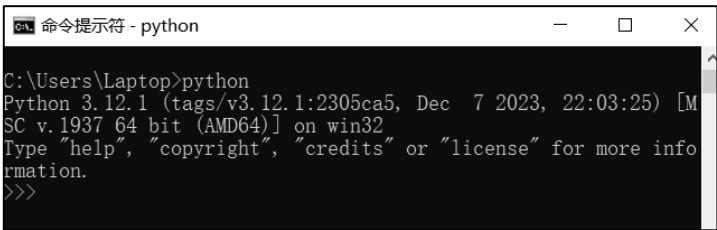


图 1-7 命令提示符窗口

1.2.2 运行 Python 程序

运行 Python 程序的基本步骤取决于操作系统和编辑器或集成开发环境。以下介绍一些通用的方法。

1. 使用 Windows 系统的命令行工具

命令提示符窗口是 Windows 环境下的虚拟 DOS 窗口。打开命令提示符窗口后,输入 Python 命令,如代码 1-1 所示,按回车键,如果出现“>>>”符号,那么说明已经进入 Python 交互式编程环境。Windows 运行结果如图 1-8 所示。

代码 1-1 Python 命令

```
1 print('Hello world') # 输出 Hello world 语句
```

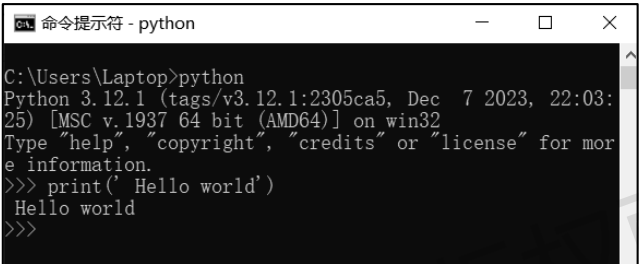


图 1-8 Windows 运行结果

2. 使用带图形界面的 Python Shell——IDLE

IDLE 是开发 Python 程序的基本集成开发环境。IDLE 适合用来测试和演示一些简单代码的执行效果。IDLE 由 Guido 亲自编写,安装好 Python 后,可以在“开始”菜单中选择“IDLE (Python 3.12 64-bit)”,即可打开环境界面,如图 1-9 所示,程序 IDLE 运行结果如图 1-10 所示。

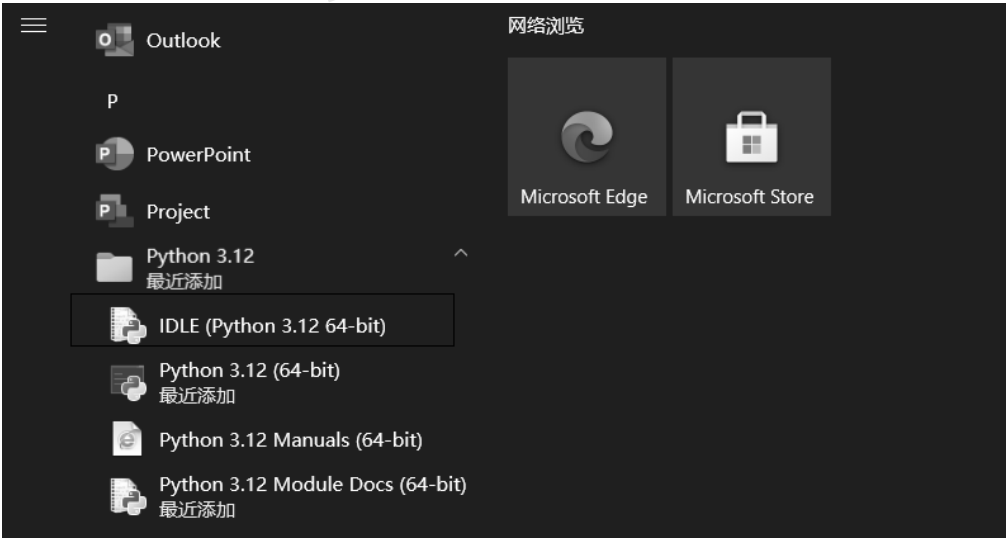


图 1-9 打开环境界面

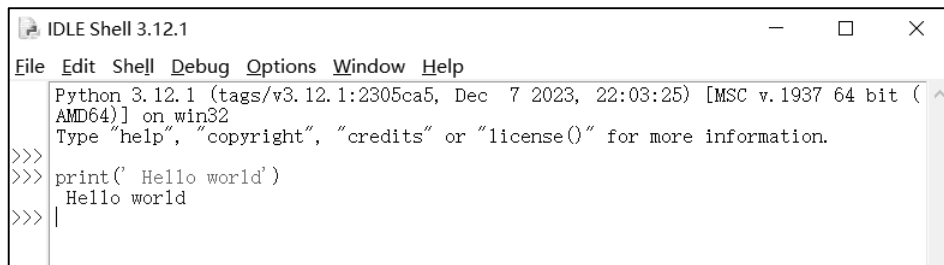


图 1-10 程序 IDLE 运行结果

3. 使用命令行版本的 Python Shell

在命令行输入 “`print('Hello world')`” 后按回车键，输出括号中的语句，如图 1-11 所示。

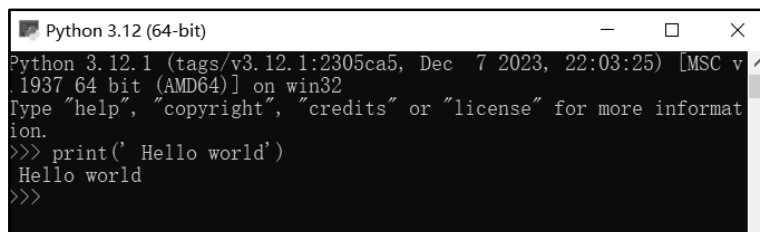


图 1-11 命令行版本的 Python Shell

4. 使用脚本模式

脚本模式指将 Python 代码保存为 .py 文件（Python 脚本文件），通过执行该文件来运行 Python 程序。可以通过以下步骤运行。

- （1）使用任意文本编辑器（如记事本、Sublime Text）创建一个以 .py 结尾的 Python 脚本文件，如 hello.py，将文件保存在命令行默认路径下。
- （2）在脚本文件中编写 Python 代码，如代码 1-2 所示。

代码 1-2 Python 代码 1

```
1 print('Hello world') # 输出 Hello world 语句
```

- （3）打开命令行工具。

- （4）使用 “`python hello.py`” 命令运行该脚本文件，如图 1-12 所示。



图 1-12 运行结果

1.3 集成开发环境 PyCharm 的安装及使用

PyCharm 是 JetBrains 公司开发的 Python 集成开发环境，由于具有智能代码编辑器、智能提示、自动导入等功能，已成为 Python 专业开发人员和初学者广泛使用的一款 Python 开发工具。

1.3.1 PyCharm 的下载与安装

下面以 Windows 系统为例，介绍如何下载并安装 PyCharm。

1. PyCharm 的下载

- (1) 打开浏览器，访问 JetBrains 中文官网。
- (2) 在官网首页，找到 PyCharm 的下载页面。PyCharm 分为专业版和社区版，专业版是收费的，而社区版是免费的。社区版已经足够满足大多数基础的 Python 编程需求。
- (3) 选择 PyCharm 的社区版，如图 1-13 所示，单击“下载”按钮。



图 1-13 PyCharm 社区版下载

2. PyCharm 的安装

- (1) 下载完成后，双击安装程序打开安装界面，如图 1-14 所示，单击“下一步”按钮。
- (2) 在安装界面中自定义安装目录，如图 1-15 所示，单击“下一步”按钮。



图 1-14 PyCharm 安装界面

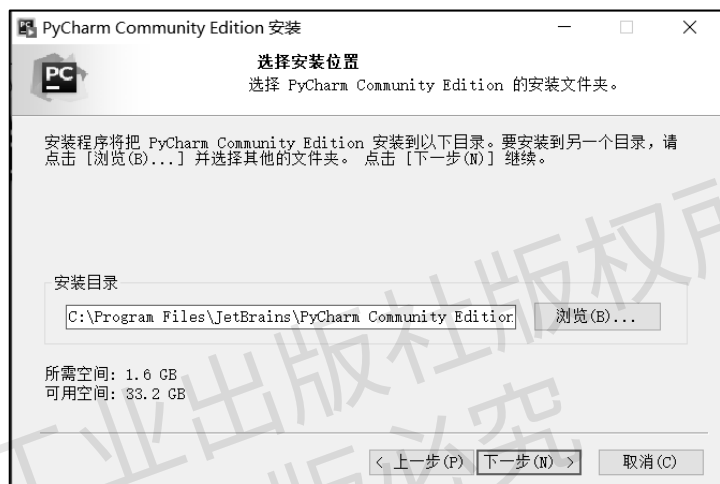


图 1-15 PyCharm 安装目录

(3) 在下一个安装界面中创建桌面快捷方式并关联.py 文件，如图 1-16 所示，单击“下一步”按钮。



图 1-16 PyCharm 安装向导

(4) 安装完成后，单击“完成”按钮，如图 1-17 所示。



图 1-17 PyCharm 安装完成

(5) 双击桌面上的 PyCharm 图标，进入如图 1-18 所示的界面。第一个选项为“Config or installation directory”（配置或安装目录），第二个选项为“Do not import settings”（不导入设置），选择第二个选项即可。

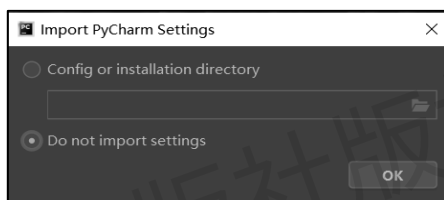


图 1-18 PyCharm 启动

(6) 进入 Welcome to PyCharm 窗口，提示安装成功，如图 1-19 所示。“Welcome to PyCharm”窗口的左侧面板中有 4 个选项，分别是 Projects、Customize、Plugins 和 Learn，这 4 个选项分别表示项目、自定义配置、插件和学习 PyCharm 的帮助文档。右侧面板中有 3 个选项，分别是 New Project、Open 和 Get from VCS，这 3 个选项的功能分别是创建新项目、打开已有项目和从版本控制系统中获取项目。

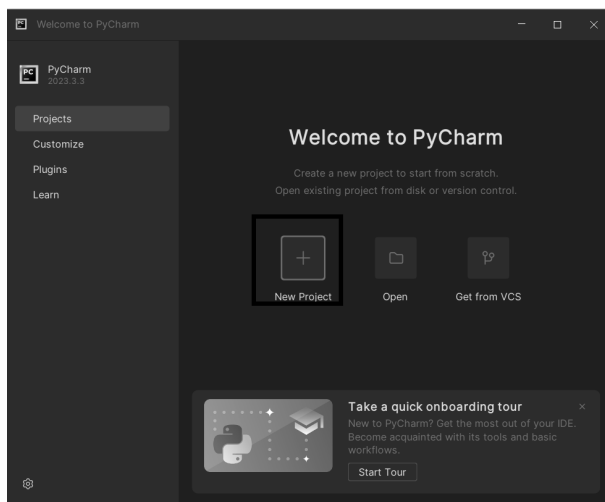


图 1-19 Welcome to PyCharm 窗口

1.3.2 PyCharm 的使用

PyCharm 提供了代码编辑、调试、测试等多种功能，使 Python 开发变得更加高效和便捷。

1. PyCharm 的配置

下面以自定义配置为例，演示如何使用 PyCharm 重置颜色主题，并在项目中运行代码，具体步骤如下。

(1) 在图 1-20 中，单击窗口左侧的“Customize”打开自定义配置面板，在该面板中选择颜色主题为 Light。

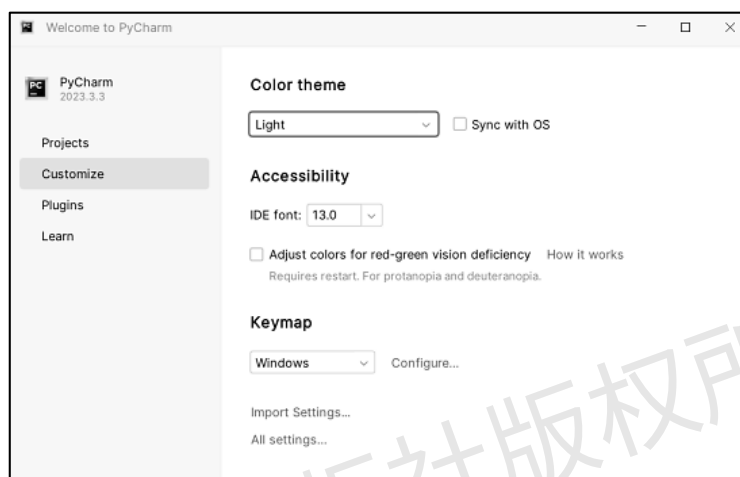


图 1-20 选择颜色主题

(2) 单击窗口左侧的“Projects”选项切换回项目面板，单击右侧面板中的“New Project”选项进入 New Project 窗口，如图 1-21 所示，该窗口的部分选项解释如下。

①Name: 项目名称。

②Location: 项目存储的位置。

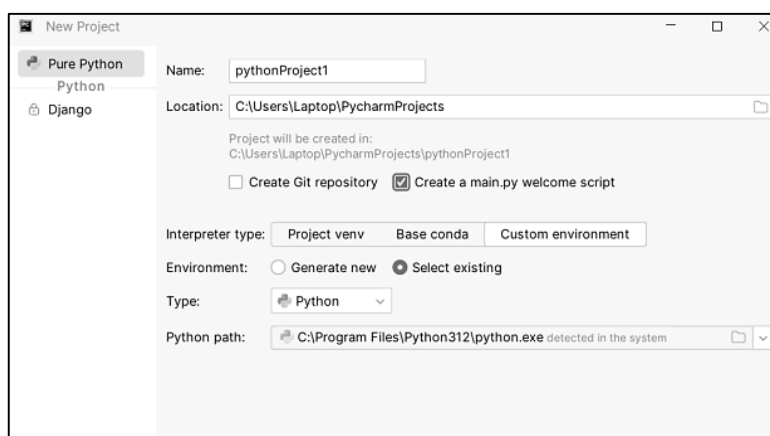


图 1-21 New Project 窗口

③Create a main.py welcome script 复选框: 选择是否将 main.py 文件添加到新创建的项目中。

main.py 文件包含简单的 Python 代码示例，可以作为项目的起始文件。

④Interpreter type: 解释器类型，其中 Project venv 是指 Virtualenv Environment，即 Python 的虚拟环境；Base conda 是指 Conda Environment，即使用 Anaconda 带有的 Python 解释器；Custom environment 是指自定义设置环境，即配置下载到本地的其他解释器。选择“Custom environment”，选择已经安装的本地 Python 解释器。

(3) 上一步完成后，会出现如图 1-22 所示的界面。



图 1-22 打开默认的 main 文件

(4) 查看 main 文件，单击右上角的运行图标即可运行程序，出现图 1-23 所示的结果说明运行成功，此时 PyCharm 环境搭建就完成了。



图 1-23 main 文件运行成功

2. PyCharm 界面介绍

PyCharm 是可用于编写代码的 IDE 工具。为了方便读者编写或修改代码，本书的代码均使用 PyCharm 编写和测试。PyCharm 界面如图 1-24 所示。

由图 1-24 标注可知，PyCharm 界面可分为菜单栏、项目结构区、代码区、信息显示区和工具栏，其工作范围介绍如下。

(1) 菜单栏：包含影响整个项目或部分项目的命令，如打开项目、创建项目、重构代码、运行和调试应用程序、保存文件等。

(2) 项目结构区：已经创建完成的项目或文件的展示区域。

(3) 代码区：编写代码的区域。

(4) 信息显示区：查看程序输出信息的区域。

(5) 工具栏：放置快捷命令，可以实现 Python 交互式等功能。

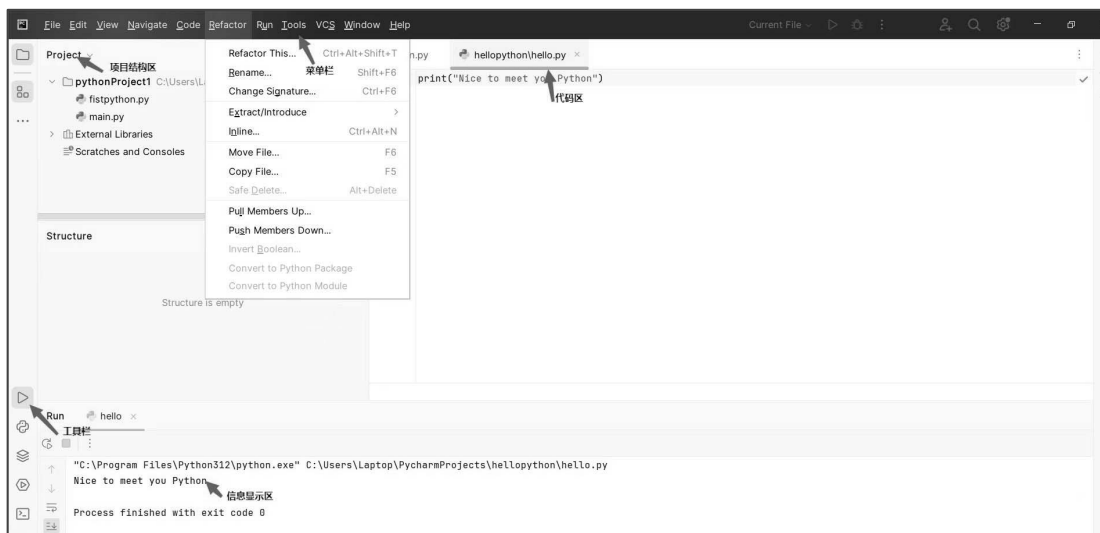


图 1-24 PyCharm 界面

3. Python 项目的创建

下面使用 PyCharm 工具创建 Python 项目和文件，在 Python 文件中编写代码，并展示代码的运行结果，具体步骤如下。

(1) 打开 PyCharm 界面，新建一个名为“hellopython”的项目，如图 1-25 所示。

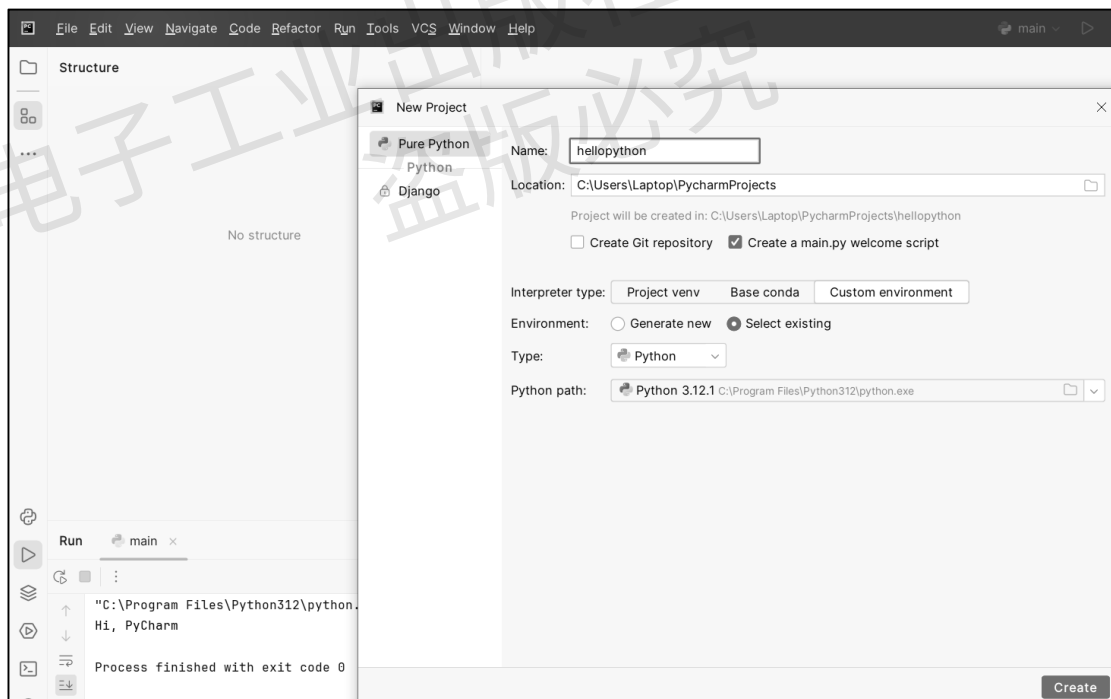


图 1-25 新建项目

(2) 单击“Create”按钮，出现一个提示窗口，单击“This Window”按钮，如图 1-26 所示。

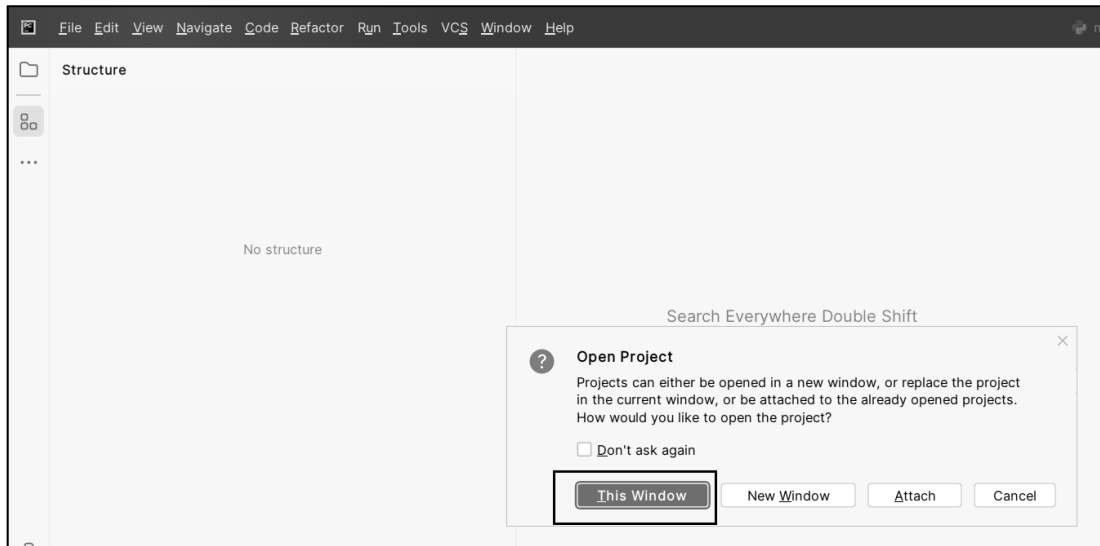


图 1-26 单击 “This Window” 按钮

(3) 鼠标右键单击如图 1-25 所示的项目名称, 在弹出的快捷菜单中选择 “New” → “Python File”, 会弹出 New Python file 窗口。在 New Python file 窗口的文本框中输入 “hello”, 按回车键后即可看到 hellopython 目录下新增的 hello.py 文件。

在 hello.py 文件编辑区输入如代码 1-3 所示的代码, 在 PyCharm 中的界面如图 1-27 所示。

代码 1-3 Python 代码 2

```
1 print("Nice to meet you Python") # 输出基本语句
```

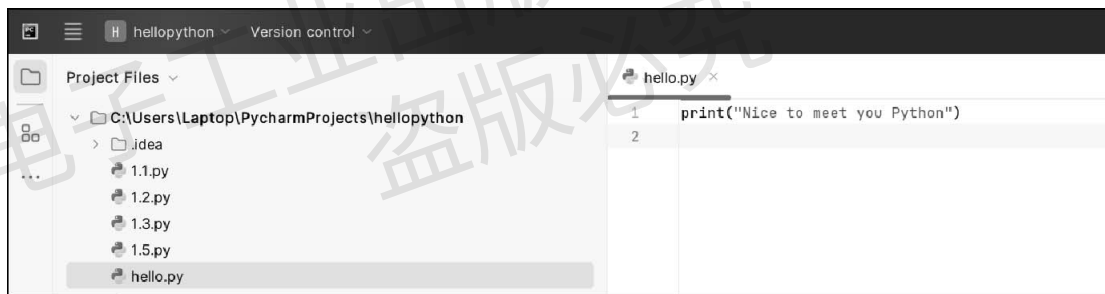


图 1-27 hello.py 代码

(4) 因默认运行的文件为 main.py, 运行 hello.py 需切换到 Current File, 如图 1-28 所示。再单击运行图标, 结果出现在界面下方的信息显示区。



图 1-28 切换到 Current File

(5) 信息显示区输出了 hello.py 文件的运行结果，如图 1-29 所示。



图 1-29 hello.py 的运行结果

小结

本章介绍了 Python 的特点及应用，同时介绍了 Python 环境的搭建过程，其中包括如何在 Windows 系统上安装 Python 解释器。此外，还介绍了常用的 Python 集成开发环境 PyCharm 的安装和基本使用。

习题

1. 选择题

- (1) Python 的打开方式不包括 ()。
- A. 在命令提示符窗口输入 Python 命令
 - B. 使用带图形界面的 Python Shell——IDLE
 - C. 使用命令行版本的 Python Shell
 - D. 使用 Python Module Docs
- (2) 以下关于在 PyCharm 中创建.py 文件的操作正确的是 ()。
- A. File→NewFile
 - B. File→New Project
 - C. File→New→Python File
 - D. File→Open
- (3) 下列选项中，属于 Python 游戏开发领域的是 ()。
- A. NumPy
 - B. SciPy
 - C. Matplotlib
 - D. Pygame
- (4) 下列选项中，不属于 Python 语言特点的是 ()。
- A. 简洁
 - B. 开源
 - C. 不支持中文
 - D. 可移植
- (5) 下列关于 Python 解释器的描述中，错误的是 ()。
- A. Python 程序的执行依赖 Python 解释器
 - B. Python 解释器仅支持 Windows 和 Linux 系统
 - C. 只有在计算机中安装 Python 解释器，才能运行 Python 代码
 - D. 在安装 Python 解释器的过程中，可设置环境变量

2. 简答题

- (1) 简述 Python 的应用领域。
- (2) 简述 Python 程序的运行过程。
- (3) 简述 Python 的特性。

3. 编程题

- (1) 在 PyCharm 中至少使用两种方式输出 “Nice to meet you Python”。
- (2) 在 PyCharm 中输入 “Nice to meet you Python”，并输出 “Nice to meet you Python”。
- (3) 编辑程序，实现从键盘输入一个整数和一个字符，在屏幕上显示输出信息。
- (4) 在 PyCharm 中创建一个名为 “hello_world.py” 的文件，并输入简单的语句来打印 “Hello, World!”。
- (5) 编辑程序，实现从键盘输入一个字符串。

电子工业出版社版权所有
盗版必究