

校企“双元”合作新形态教材



Python程序设计 基础项目化教程

Python CHENGXU SHEJI JICHU XIANGMUHUA JIAOCHENG

主编 李 莉 杨娜娜
主审 劳 飞



北京邮电大学出版社
www.buptpress.com

校企“双元”合作新形态教材

Python程序设计 基础项目化教程

Python CHENGXU SHEJI JICHU XIANGMUHUA JIAOCHENG

主 编 李 莉 杨娜娜
副主编 丁 冉 谷 平 韩霄萌
 张 倩 石宁飞
主 审 劳 飞



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书旨在通过项目化教学帮助学生掌握 Python 语言的基本语法、编程思想、常用库和框架等知识,培养学生的编程能力和解决问题的能力。全书共分 9 个项目,内容包括实现智能交互、设计购物车程序、设计简易大运会数据统计程序、设计登录验证程序、编写数独游戏、开发新能源汽车管理系统、花式处理文件、异常的处理及开发滨滨文创店智能销售系统。

本书既适合作为高等职业院校 Python 程序设计课程的教材,也可作为广大 Python 编程爱好者的自学参考书。

图书在版编目(CIP)数据

Python 程序设计基础项目化教程 / 李莉, 杨娜娜主
编. -- 北京 : 北京邮电大学出版社, 2024. -- ISBN
978-7-5635-7365-3
I. TP312.8
中国国家版本馆 CIP 数据核字第 2024MF8178 号

策划编辑: 高 锐 责任编辑: 柳卫清 封面设计: 刘文东

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 三河市龙大印装有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 13.75

字 数: 284 千字

版 次: 2024 年 10 月第 1 版

印 次: 2024 年 10 月第 1 次印刷

ISBN 978-7-5635-7365-3

定 价: 53.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

服务电话: 400-615-1233

随着信息技术的快速发展,编程已成为现代社会中人们不可或缺的一项技能。Python 作为一种功能强大、简单易学的编程语言,在数据分析、人工智能和 Web 开发等领域都有着广泛的应用。因此,掌握 Python 语言对于大学生未来的职业发展具有重要的意义。

本书以党的二十大精神为指导,将知识技能与素质教育有机融合,潜移默化地培养学生的职业精神和道德素养。本书以教育部制定的《高等职业教育专科信息技术课程标准(2021 年版)》中程序设计模块的相关要求为依据,以提升学生就业实战能力为导向,从现实生活中的实际案例出发,利用“项目+任务”学习法来完成案例实践,使学生在轻松完成案例的体验中掌握 Python 程序设计的操作要点。

程序设计基础是高等职业院校计算机及相关专业学生的一门必修课程,它是学校素质教育的重要组成部分。该课程的特点是具有较强的针对性、实用性、指导性和实践性。本书的设计主要体现了“以能力为本位、以项目课程为主体”的思想,打破了传统的学科体系的模式,将实际生活中用到的知识点融合在课程学习的各个项目中。

项目 1 为实现智能交互。该项目通过认识 Python、运行交互式界面等任务详细介绍 Python 的发展历史与特点、应用领域、使用工具、交互式编写代码、文件式编写代码等内容。

项目 2 为设计购物车程序。该项目通过定义并输出商品信息、统计商品数据、计算商品总价等任务详细介绍变量与常量、基本数据类型及转换、基本运算符和表达式的运用等内容。

项目 3 为设计简易大运会数据统计程序。该项目通过设计国家名称中英文对照表、实现运动员国籍对照、呈现赛项、更新奖牌榜等任务详细介绍元组、列表、集合、字典的定义、格式与用法、常见操作等内容。

项目 4 为设计登录验证程序。该项目通过实现垃圾分类、打印九九乘法表、编写“剪刀石头布”小游戏等任务详细介绍流程控制基本结构、分支语句的分类、for 语句和 while 语句的格式与用法及嵌套等内容。

项目 5 为编写数独游戏。该项目通过编写计算器程序、制订 365 天存钱计划等任务详细介绍函数的定义与特点、格式与用法、常见的函数与操作、参数的使用和变量作用域、递归函数的概念和用法、lambda 表达式的格式与用法、函数的封装和模块等内容。

项目 6 为开发新能源汽车管理系统。该项目通过设计新能源汽车类、将“探寻新能源概念融入”井字棋游戏设计等任务详细介绍 Python 面向对象的概念,类和对象的定义及使用、属性和方法,类的继承、封装和多态等内容。

项目 7 为花式处理文件。该项目通过绘制签名墙、查询身份证的归属地等任务详细介绍文件的打开及关闭、文件的读写、文件的复制与重命名、文件路径操作、文件的定位读取等内容。

项目 8 为异常的处理。该项目通过捕获异常、处理异常的方法和抛出异常等任务详细介绍异常的定义、try...except 语句与 raise 语句的使用、捕获异常的方法、异常传递等内容。

项目 9 为开发滨滨文创店智能销售系统。该项目通过 4 个任务详细介绍软件开发、系统架构设计、软件测试等内容。

本书特色主要体现在以下几个方面。

(1)以实践为导向。本书通过项目引领、任务驱动的方式来组织教学内容。每个项目包括多个具体任务,这些任务基本从属于一个中心主题,旨在帮助学生在解决实际问题的过程中掌握 Python 编程技能。这种项目驱动的学习方式使学生能够将理论知识与实际应用结合起来,提高编程实践能力。

(2)注重语法基础。虽然 Python 的语法相对简单,但也有一些重要的概念需要掌握,如变量、数据类型和控制结构等。本书注重这些基础知识的讲解和练习,确保学生在开始项目实践之前具备扎实的语法基础。

(3)强调团队合作。在本书的每个项目中,学生通常需要分组合作,共同完成任务。这种团队合作的方式不仅可以培养学生的团队协作能力,还可以让他们学会如何与他人有效沟通,共同解决问题。

(4)不断更新和优化。由于 Python 和其生态系统的发展非常迅速,本书后续会不断更新和优化,以反映最新的编程技术和最佳的实践效果,保证学生可以学到最新、最实用的 Python 编程知识。

本书由滨州职业学院、莱芜职业技术学院骨干教师和企业专家联合编写,由滨州职业学院李莉、杨娜娜担任主编,并负责课程大纲把握和统稿等工作;滨州职业学院丁冉,莱芜职业技术学院谷平,滨州职业学院韩霄萌、张倩及国基北盛(南京)科技发展有限公司石宁飞担任副主编,滨州职业学院劳飞担任主审。具体编写分工为:项目1和项目2由谷平编写,项目3由韩霄萌编写,项目4由李莉编写,项目5由石宁飞编写,项目6和项目7由张倩编写,项目8由杨娜娜编写,项目9由丁冉编写。

在此,要特别感谢学校领导和编者家人对本书编写工作的大力支持,他们的关爱与支持是本书编写完成的动力与源泉。

由于编者水平有限,书中难免存在不足之处,欢迎广大读者批评指正,以便修订时改正。

编者

项目 1 / 实现智能交互 1**任务 1.1 认识 Python 2**

1.1.1 了解 Python 的前世今生 2

1.1.2 探究 Python 的特点 3

1.1.3 了解 Python 的应用领域 4

1.1.4 掌握 Python 常用的开发工具 5

任务 1.2 运行交互式界面 14

1.2.1 启用交互式开发界面 14

1.2.2 实现文件式编程界面 14

项目 2 / 设计购物车程序 21**任务 2.1 定义并输出商品信息 22**

2.1.1 区分变量与常量 22

2.1.2 掌握 Python 的规范编程步骤 23

任务 2.2 统计商品数据 27

2.2.1 认识基本数据类型 28

2.2.2 掌握基本数据类型的转换方法 30

任务 2.3 计算商品总价 32

2.3.1 掌握常用操作运算符的使用方法 32

2.3.2 熟知运算符优先级 37

项目 3 / 设计简易大运会数据统计程序 43**任务 3.1 设计国家名称中英文对照表 44**

3.1.1 认识元组 44

3.1.2 掌握实现元组的常见方法和函数 47

任务 3.2 实现运动员国籍对照 50

3.2.1	认识列表	50
3.2.2	掌握实现列表的常见方法和函数	53
任务 3.3	呈现赛项	57
3.3.1	认识集合	57
3.3.2	掌握实现集合的常见方法和函数	59
任务 3.4	更新奖牌榜	61
3.4.1	认识字典	62
3.4.2	掌握实现字典的常见方法和函数	66

项目 4 // **设计登录验证程序** **73**

任务 4.1	实现垃圾分类	74
4.1.1	了解流程控制结构	74
4.1.2	实现 if 相关语句	74
任务 4.2	打印九九乘法表	79
4.2.1	编写 for 循环语句	79
4.2.2	了解 break 和 continue 语句	81
4.2.3	嵌套使用 for 循环语句	83
任务 4.3	编写“剪刀石头布”小游戏	85
4.3.1	编写 while 循环语句	85
4.3.2	嵌套使用 while 循环语句	86

项目 5 // **编写数独游戏** **95**

任务 5.1	编写计算器程序	96
5.1.1	了解函数的定义及作用	96
5.1.2	调用函数	98
5.1.3	认识常见的函数	100
5.1.4	了解封装和模块	102
任务 5.2	制订 365 天存钱计划	105
5.2.1	了解参数的类型及传递方法	105
5.2.2	了解变量的作用域	108

项目 6 / 开发新能源汽车管理系统 119

- 任务 6.1 设计新能源汽车类 120**
 - 6.1.1 了解 Python 面向对象的定义 120
 - 6.1.2 掌握类和对象的定义和使用方法 121
 - 6.1.3 掌握类和对象的属性及方法 123
- 任务 6.2 将“探寻新能源”概念融入井字棋游戏设计 126**
 - 6.2.1 类的继承 127
 - 6.2.2 类的封装 128
 - 6.2.3 类的多态 132

项目 7 / 花式处理文件 141

- 任务 7.1 绘制签名墙 142**
 - 7.1.1 打开和关闭文件 142
 - 7.1.2 读写文件 143
- 任务 7.2 查询身份证的归属地 149**
 - 7.2.1 复制文件 149
 - 7.2.2 重命名文件 150
 - 7.2.3 设置文件的工作路径 151
 - 7.2.4 定位读取文件 153

项目 8 / 异常的处理 163

- 任务 8.1 捕获异常 164**
 - 8.1.1 认识异常 164
 - 8.1.2 运用 try...except 语句 166
 - 8.1.3 捕获异常信息 168
- 任务 8.2 处理异常的方法 169**
 - 8.2.1 运用 raise 语句 169
 - 8.2.2 掌握异常的参数 170

任务 8.3	抛出异常	172
8.3.1	用户自定义异常	172
8.3.2	异常的传递	173

项目 9 / **开发滨滨文创店智能销售系统** **183**

任务 9.1	建立系统架构	184
9.1.1	设计软件开发流程	184
9.1.2	进行需求分析	185
9.1.3	设计软件系统	186
任务 9.2	设计类	189
9.2.1	了解常见的类	189
9.2.2	设计类图	189
任务 9.3	实现模块功能	192
9.3.1	进行详细设计	193
9.3.2	进行数据结构设计	193
任务 9.4	测试软件	206
9.4.1	给软件测试分类	206
9.4.2	明确软件测试的重要性	207
9.4.3	测试用例	207

项目 1

实现智能交互

Python 语言具有简单易学的语法、高效的运行速度、丰富的库、对数据处理广泛的支持、强大的解释器,以及可以与多种编程语言进行交互等优势。其被广泛应用于各个领域。Python 语言的学习难度相对较低,学习周期短,可以快速上手。本项目首先介绍 Python 的历史、特点,然后介绍如何搭建 Python 开发环境及如何进行开发环境的测试等内容。



学习目标

1. 知识目标

- (1)了解 Python 的历史和特点。
- (2)了解 Python 的应用领域。
- (3)掌握交互式编写代码、文件式编写代码的方法。

2. 技能目标

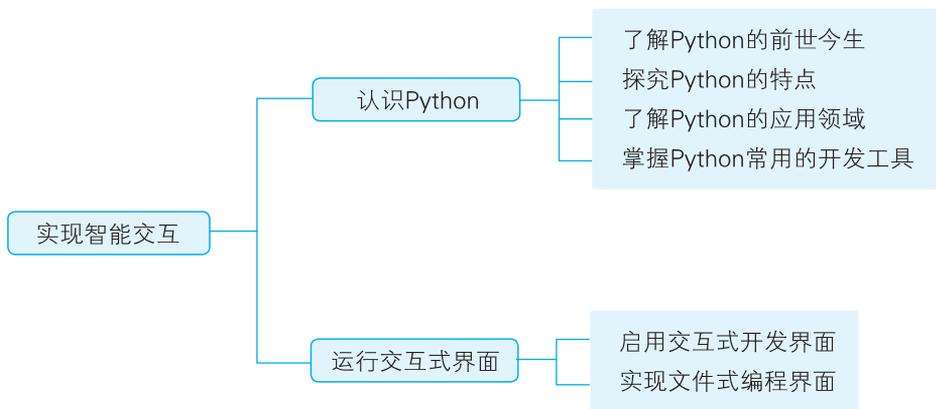
- (1)熟练下载并安装 Python 3 软件。
- (2)熟练下载并安装 PyCharm 软件。
- (3)熟练下载并安装 Anaconda 软件。
- (4)能够实现 Python 的交互式界面编程。

3. 素养目标

- (1)树立团结协作意识。

- (2) 激发对 Python 语言的学习兴趣。
- (3) 根据 Python 语言的就业方向,明确职业规划。

知识图谱



任务 1.1 认识 Python

任务描述

Python 语言是一种既强大又易于学习的编程语言。它的语法简洁明了,可读性强,非常适合初学者入门学习。为了激发对 Python 编程的兴趣,学习者需要不断尝试、实践和探索。了解 Python 语言的历史、特点及应用领域,可以帮助学习者建立起对 Python 语言的学习兴趣。通过学习本任务,学习者可以深入了解 Python 语言的奥秘。在掌握 Python 的基础知识后,将学习如何下载并安装相关软件。

相关知识

1.1.1 了解 Python 的前世今生

Python 的发展历史可以分为 6 个阶段,如图 1-1 所示。



视频

认识 Python

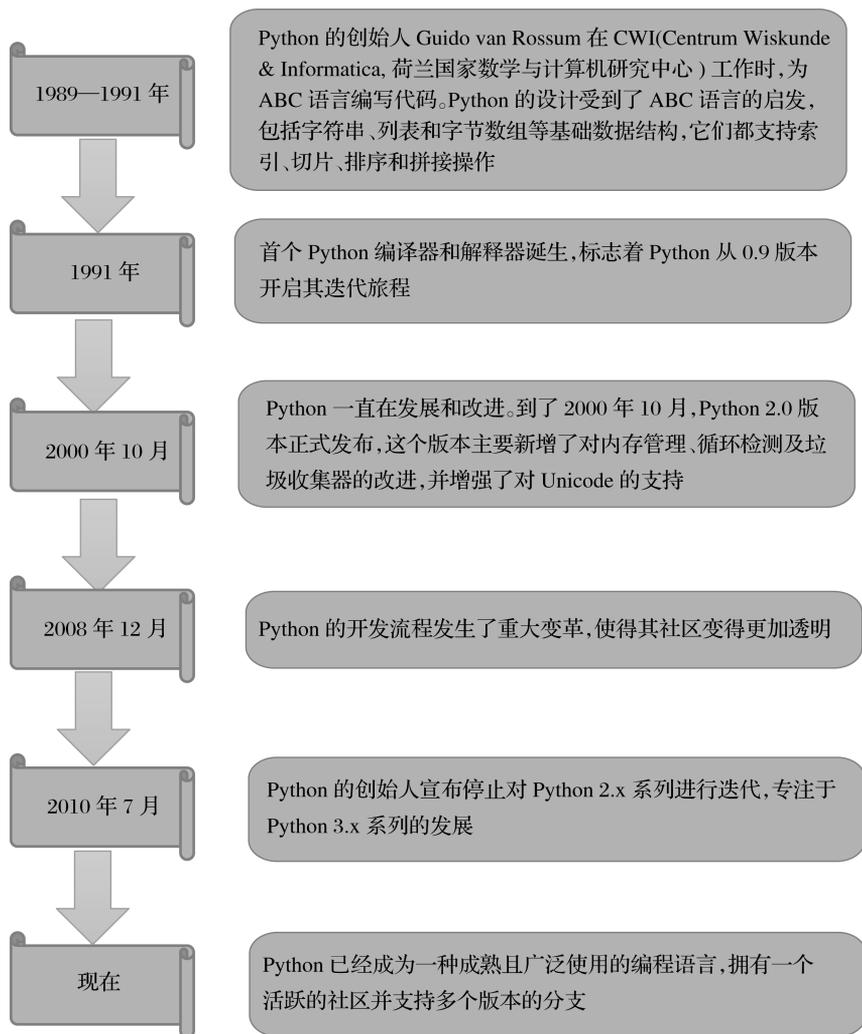


图 1-1 Python 的发展历史

1.1.2 探究 Python 的特点

自诞生以来,Python 凭借其独特的特点和优势,逐渐在编程领域崭露头角。Python 的主要特点大致有以下几点。

1. 语法简洁易懂

Python 的语法清晰明了,没有过多烦琐的语法规则,因此它成为了编程初学者的首选语言。Python 采用缩进来表示代码块,而非使用大括号,这种简洁的语法让代码更加易读易写。此外,Python 还支持多种编程范式,包括面向对象、面向过程和函数式编程,这赋予了使用 Python 编写代码时的极大灵活性。

2. 是一种解释型语言

开发者可以直接运行 Python 代码,并立即获得运行结果。这种即时反馈的特性让 Python 在调试和测试阶段具备更高的效率。此外,作为一种解释型语言,Python 具有跨平

台的特性,这意味着它可以在不同的操作系统上运行,无需担心平台兼容性问题。

3. 具有丰富的标准库和第三方库

Python 的标准库包含了大量用于常见任务的函数和模块,如文件操作、网络编程及数据库访问等。这些库为开发者提供了强大的功能支持,使得他们可以快速构建出功能完备的应用程序。此外,Python 还拥有庞大的第三方库生态系统,这些库覆盖了从数据分析、机器学习到 Web 开发等多个领域。开发者可以借助这些第三方库快速实现复杂的功能,从而提高开发效率。

4. 具有可扩展性

尽管 Python 本质上是一种解释型语言,但它却具备良好的可扩展性,允许开发者通过 C、C++ 等编译型语言对其进行扩展。这意味着开发者可以在 Python 中调用 C、C++ 等语言的代码,从而实现更高的执行效率或实现特定功能。这种可扩展性使得 Python 在性能要求较高的场景下也能展现出卓越的表现。

5. 提供庞大的社区支持

Python 社区非常活跃,汇聚了大量的开发者和爱好者。这个社区不仅为 Python 的发展提供了源源不断的动力,还为开发者提供了丰富的资源和帮助。无论是遇到技术难题,还是寻找学习资料,Python 社区都能为开发者提供强有力的支持。

1.1.3 了解 Python 的应用领域

“我学会了 Python 语言后,可以从事哪些方面的工作?”“学习 Python 语言,对我的求职和就业有哪些帮助?”相信大多数的 Python 初学者在开始学习之前都会有这样的疑问。众所周知,Python 如今是极为热门的编程语言之一,它不仅提供了高薪的就业机会,还拥有广阔的职业前景。

1. Python 自动化运维

像我们平时玩的网游,它们都是运行在服务器上的,有些大型游戏背后可能依赖于成千上万的服务器集群。这些服务器可以简单地理解为功能强大的计算机,它们同样会面临死机、卡顿及需要重启等问题。那么,如何高效的同时维护这么多服务器的状态呢?这时,我们就可以编写一些 Python 脚本来自动化完成这些维护任务。

2. 网络爬虫

简而言之,爬虫就像一个智能探测工具,它的基本操作是通过模拟人的行为来浏览各个网站,单击按钮,检索数据,并将获取到的信息存储起来。掌握了 Python 爬虫技术之后,人们便可以轻松收集网络上的相关数据。而要精通这项技术,则需要学习前端基础知识,掌握 Python 爬虫库及数据库等技术。

3. 人工智能与大数据

随着人工智能和大数据的发展,Python 语言的地位日益凸显,其简洁易懂的代码特性吸引了越来越多的人选择学习它。在人工智能时代,Python 无疑是领先的编程语言之一。无论是机器学习(machine learning),还是深度学习(deep learning)的研究与应用,其最常用的工具和框架如 Numpy、SciPy、Pandas、PyTorch 等,都需要通过 Python 来调用。因此,掌



视频

Python 的应用领域

握 Python 是人工智能工程师的必备技能之一。

4. Python 游戏开发

在游戏开发领域,Python 的招聘主要集中在游戏服务器开发方面,职责涵盖网络游戏的服务端功能开发、性能优化等工作。Python 为游戏开发者提供了多种库和框架,如 Pygame、Panda 3D 和 Arcade 等,这些工具可以帮助开发者更加高效地创建游戏,同时提升游戏的品质。

5. Web 开发 (Python 后端)

Python 拥有众多优秀的开发框架,如 Flask、Django 和 Bootstar 等,能够实现网站的快速搭建。当需要为网站添加新功能时,往往只需要利用 Python 添加几行代码,这一特性深受众多初创型公司的青睐。事实上,一些全球顶级的流量大户,如 YouTube、Quora、Reddit、Instagram 和 Netflix 等,都选择将 Python 作为其后端开发的重要语言。

1.1.4 掌握 Python 常用的开发工具

“工欲善其事,必先利其器。”Python 作为一门广泛应用的编程语言,其开发环境的选择和配置对开发效率和舒适度具有重要影响。为更好地学习和使用 Python,需要安装并配置好 Python 运行所需的相关软件。接下来,将介绍一些 Python 开发中常用的工具。

1. Python 解释器

Python 的官方解释器是 CPython,它是默认的解释器,此外,还有其他实现方式,如 Jython 和 IronPython 等。用户可以从 Python 官方网站下载最新版本的 Python,并根据自己的操作系统选择合适的安装包进行安装。Python 自带了一个集成开发和学习环境(integrated development and learning environment, IDLE),这是基于 Tkinter 编写的纯 Python IDE,支持 Windows、macOS 和 Linux 等多种操作系统平台,适合初学者学习使用。在 Python 语言的发展历程中,主要经历了以下三个关键版本。

(1)Python 1. x:这是 Python 语言的早期版本,1994 年由 Guido van Rossum 发布。Python 1.0 版本相较于现在的版本,其功能有限,缺乏很多现在 Python 版本中常用的特性。

(2)Python 2. x:于 2000 年发布,虽然它拥有强大的库和框架支持,但由于存在一些设计缺陷,如语法不一致和 Unicode 支持不足等问题。因此,Python 2. x 在 2020 年已经停止官方支持。从 2020 年开始,Python 2. x 不再接收安全更新,官方鼓励开发者升级到 Python 3. x 以获取更好的支持和新特性。

(3)Python 3. x:于 2008 年发布,是 Python 的新一代版本。修复了 Python 2. x 中的许多问题,并引入了一些强大的新特性,如更完善的 Unicode 支持、函数注解及异步编程等。Python 3. x 系列不断更新迭代,每个新版本都带来了性能提升、bug 修复以及新的语言特性和库支持。

Python 环境变量指的是 Python 解释器在计算机中的位置。在使用 Python 进行编程时,需要告诉计算机 Python 解释器所在的路径,这是通过设置环境变量来实现的。环境变量是操作系统中的一种特殊变量,用于存储系统常用的路径、配置信息等。在 Python 编程中,环境变量的设置会直接影响到 Python 程序的运行效果。



视频

掌握 Python 常用的
开发工具

2. PyCharm

PyCharm 是一款专业的 Python 集成开发环境(integrated development environment, IDE),由 JetBrains 公司开发。它是一个跨平台的全功能 Python 开发工具,专为 Python 开发打造。PyCharm 提供两个版本:一个是免费的社区版,适合个人开发者和小型团队使用;另一个是付费的专业版,面向企业开发者,提供了更多高级功能和支持。PyCharm 因其简洁的界面、易用性以及灵活的主题设置而广受程序员好评。它集成了多种提高开发效率的功能,包括智能代码补全、调试、语法高亮、项目管理与导航、代码跳转、智能提示、图形化的调试器和运行器、自动完成、单元测试、版本控制、遵循 PEP 8 规范的代码质量检查及智能重构等。此外,PyCharm 支持 1 000 多个插件,程序员还可以编写自己的插件来扩展其功能。

3. Anaconda

Anaconda 是目前国内外高校中 Python 教学最流行的软件平台,它集成了 Python 环境管理、代码编辑器及包管理等功能,并提供了一键安装的便捷方式。Anaconda 集成了 Conda 和 Python 在内的超过 180 个科学包及其依赖项,如 Numpy、Pandas、TensorFlow 和 SciPy 等,这些科学包涵盖了数据分析、数学计算、数据可视化及深度学习等多个领域。Anaconda 3 对应的是 Python 3. x 版本,其默认编码方式为 UTF-8,有助于避免字符编码问题。若计算机上已安装 Python,安装 Anaconda 并不会造成冲突,因为脚本和程序默认使用的是 Anaconda 自带的 Python 版本。此外,作为一个开源的 Python 发行版,Anaconda 支持 Windows、Linux 和 macOS 等多种操作系统。



任务实现

搭建 Python 和 PyCharm、Anaconda 开发环境的操作步骤如下。

1. 下载并安装 Python

在 Python 官网(<https://www.python.org/>)中可以下载 Python 安装包。进入官网后,在“Downloads”下拉列表中选择“Windows”选项,随后跳转到 Python 的下载页面,单击“Python 3. 12. 2”(注意,版本号会随时间更新,实际下载时可能会有所不同)下载安装包,如图 1-2 所示。



视频

Python 开发环境
的搭建

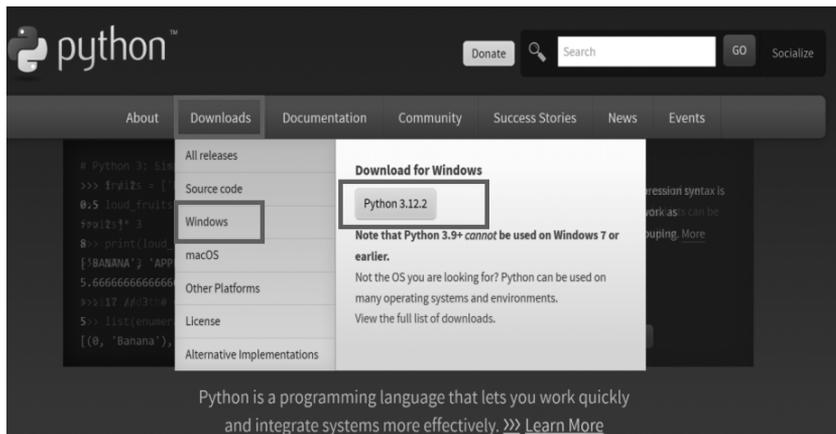


图 1-2 下载 Python 安装包

下载完成后安装 Python, 在安装过程中系统会弹出内容选择对话框, 需要选中“Add python.exe to PATH”等复选框, 可以将 Python 的安装路径添加到环境变量 PATH 中, 这样就可以在任意文件夹下使用 Python 命令。单击“Install Now”按钮就可以开始安装了, 如图 1-3 所示。

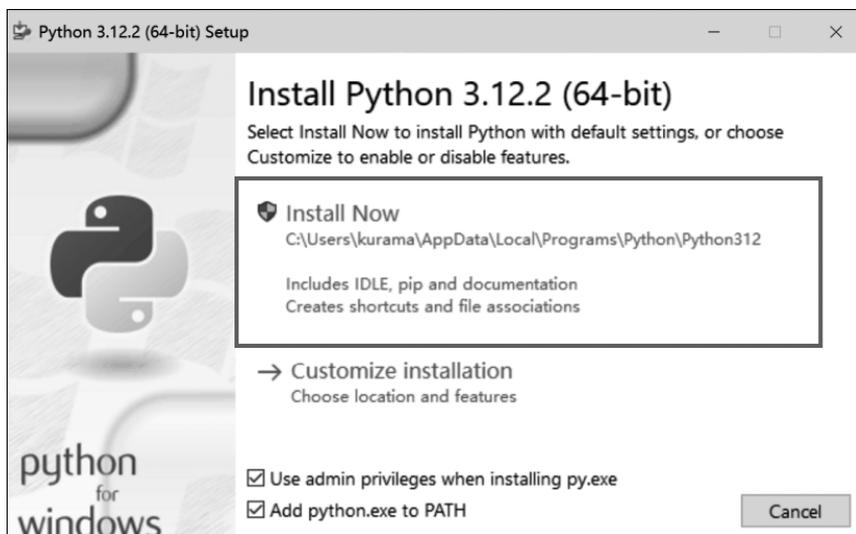


图 1-3 单击“Install Now”安装 Python

系统提示安装成功, 如图 1-4 所示。

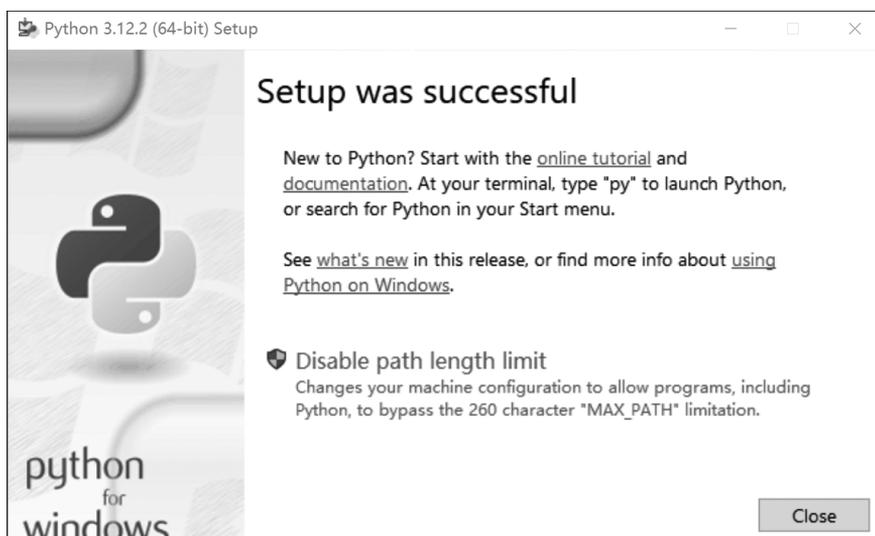


图 1-4 成功安装 Python

2. 下载并安装集成开发环境 PyCharm

访问 PyCharm 官网 (<https://www.jetbrains.com/zh-cn/pycharm/download/?section=windows>) 下载 PyCharm。该官网中包含 PyCharm Professional、PyCharm Community Edition 两个版本, 此处选择免费的 PyCharm Community Edition 版本进行下载。

下载成功后,双击 PyCharm 安装程序,系统会弹出其安装界面,如图 1-5 所示。



图 1-5 PyCharm 安装界面

选择 PyCharm 安装目录,如图 1-6 所示。



图 1-6 选择 PyCharm 安装目录

确定好安装目录后,单击“下一步”按钮进入安装选项界面,在该界面中选中需要的功能复选框,如图 1-7 所示。

打开“选择开始菜单目录”界面,该界面中依然保持默认配置,单击“安装”按钮,如图 1-8 所示。

安装完成后界面提示安装程序完成信息,单击“完成”按钮,结束 PyCharm 程序安装,如图 1-9 所示。



图 1-7 PyCharm 安装选项界面



图 1-8 “选择开始菜单目录”界面



图 1-9 完成 PyCharm 安装

PyCharm 安装完成后默认显示为英文版,用户可以根据需要将其设置为中文版。双击 PyCharm 的快捷图标以打开软件,单击界面左上角的“File”菜单按钮,在其下拉列表中选择“Settings”选项,如图 1-10 所示。

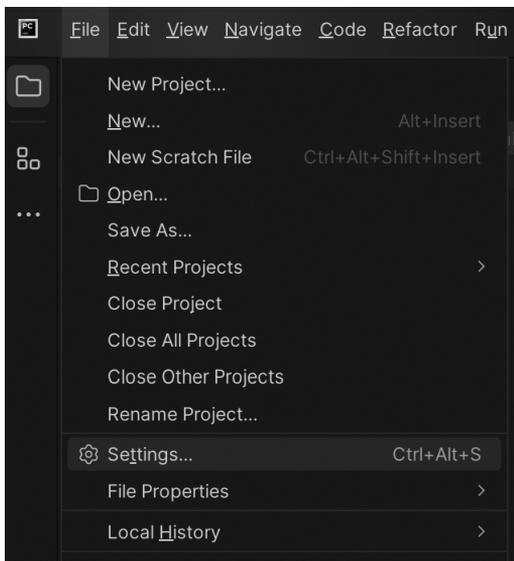


图 1-10 选择“Settings”选项

打开“Settings”界面,选择左侧窗格中的“Plugins”选项,在右侧搜索框中输入并搜索“chinese”,然后选择相应插件进行安装,如图 1-11 所示。

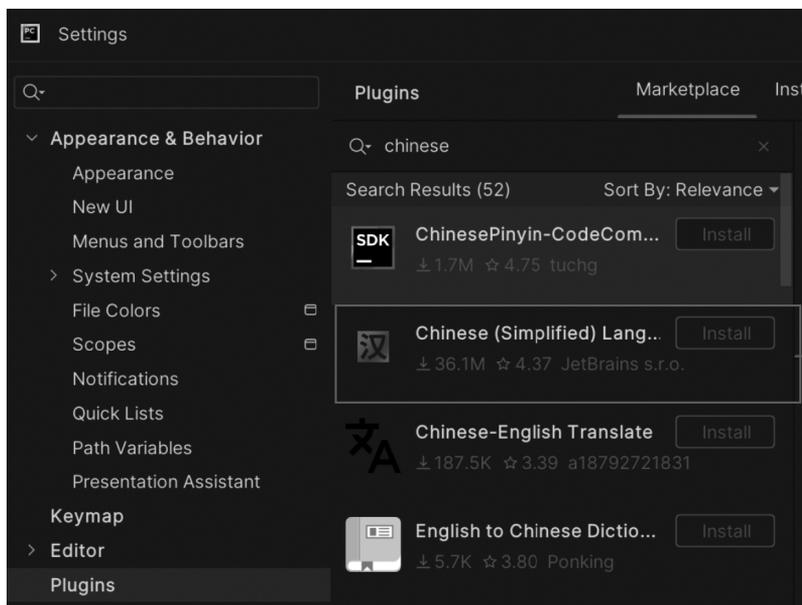


图 1-11 选择中文插件

安装完成后,系统会重启 PyCharm,随后可以看到 PyCharm 页面显示变成了简体中文,如图 1-12 所示。



图 1-12 完成 PyCharm 页面汉化

3. 下载并安装 Anaconda

在 Anaconda 官网 (<https://www.anaconda.com/download>) 下载 Anaconda 程序安装包, 如图 1-13 所示。

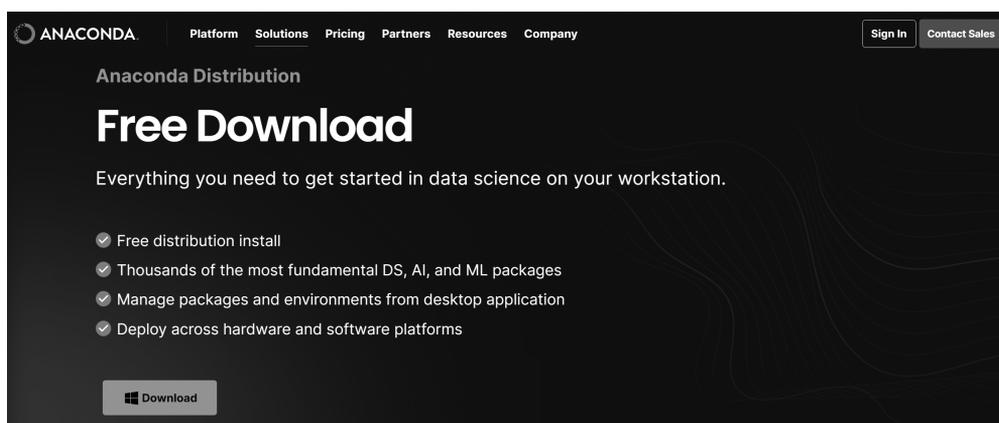


图 1-13 Anaconda 软件下载页面

单击“Download”按钮开始下载。下载完成后, 双击安装程序进行安装, 在打开的安装界面中单击“Next”按钮, 如图 1-14 所示。

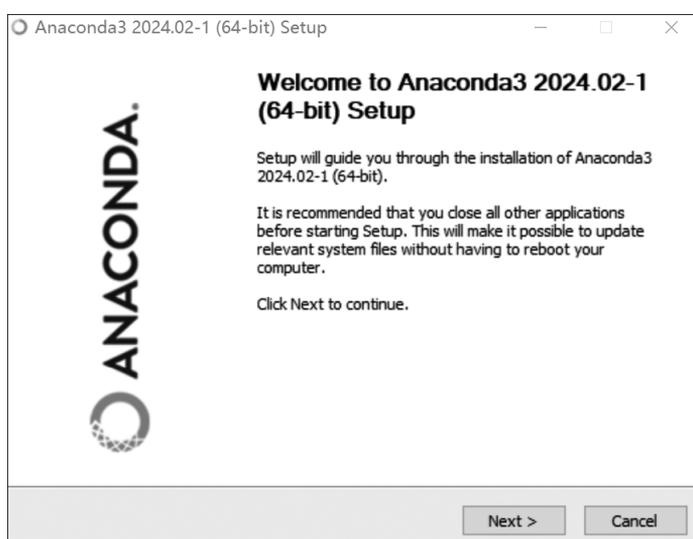


图 1-14 单击“Next”按钮

选中“All Users(requires admin privileges)”单选按钮,设置用户,再单击“Next”按钮,如图 1-15 所示。

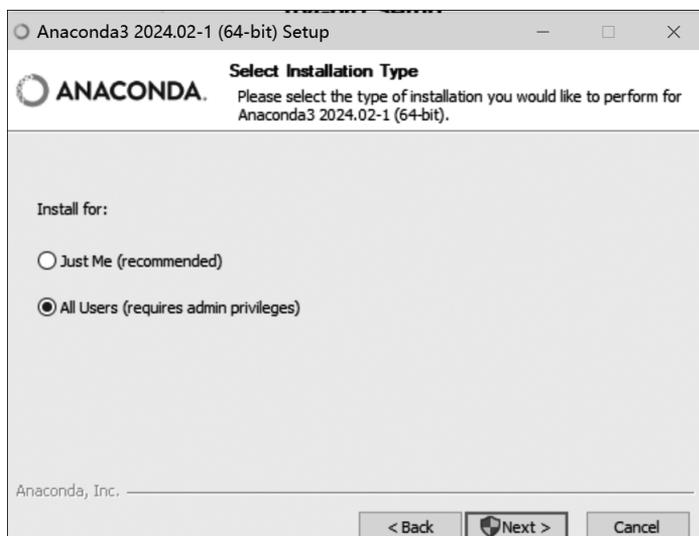


图 1-15 设置用户

在接下来出现的安装界面中都单击“Next”按钮,选择安装路径(这里可以不用安装在 C 盘),如图 1-16 所示。

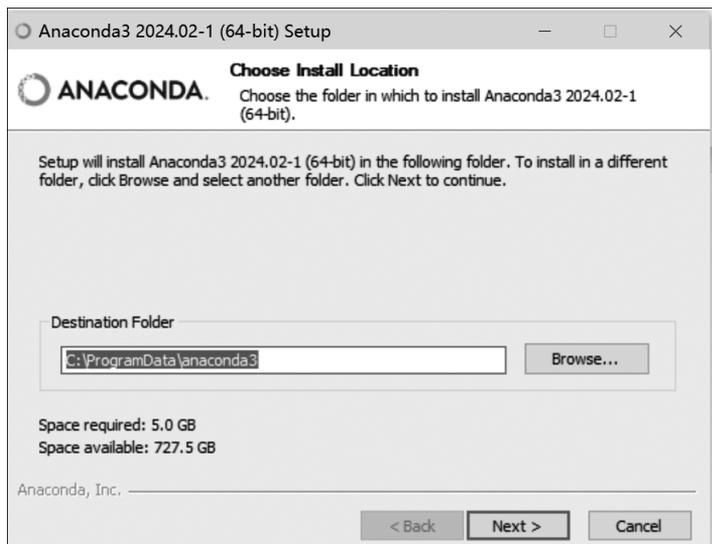


图 1-16 选择安装路径

添加环境变量,选中图 1-17 中的前两个复选框,再单击“Install”按钮,如图 1-17 所示。在接下来出现的界面中继续单击“Next”按钮,直到安装完成,如图 1-18 所示。最后单击“Finish”按钮完成安装,如图 1-19 所示。

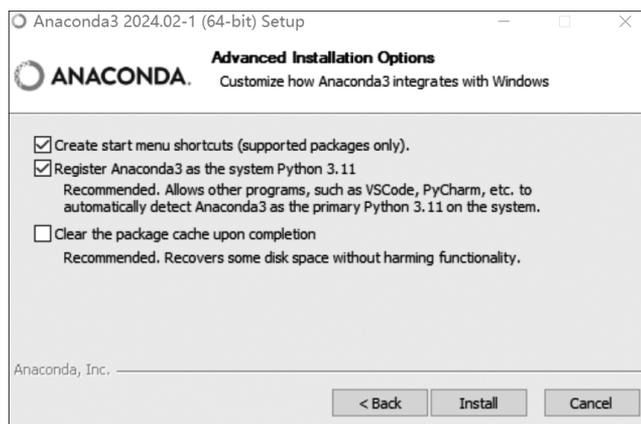


图 1-17 添加环境变量



图 1-18 单击“Next”按钮继续安装

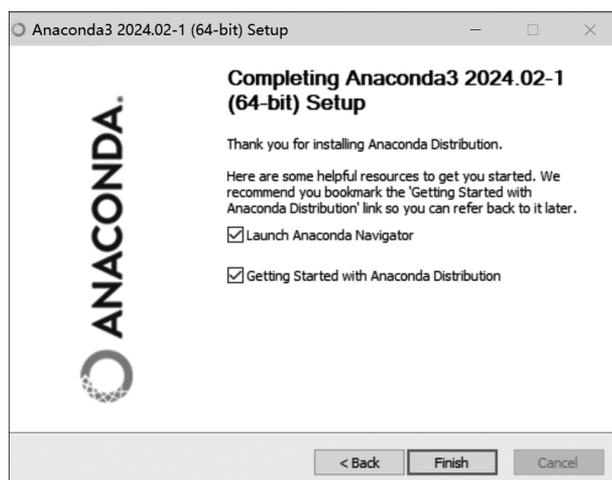


图 1-19 单击“Finish”按钮完成安装

任务 1.2 运行交互式界面

任务描述

用户界面的交互性对于用户体验至关重要。在数字化时代,无论是开发应用程序、设计网站,还是构建设备界面,都需要一个令人印象深刻的用户界面。本任务将学习如何使用 Python 语言来设计并实现交互式界面。

相关知识

Python 语言提供了多种交互式界面来进行编程,两种常用的方法是启用交互式开发界面和实现文件式编程界面。

1.2.1 启用交互式开发界面

首先来学习 Python 自带的交互式开发环境(interactive development environment, IDLE),也被称交互式开发界面。虽然这个环境具有直观易用的特点,但它也存在一定的局限性,如在使用时,它通常只允许每次执行单条语句并即时查看结果。若需逐行调试代码,则用户需要先将代码保存为 .py 文件后再进行运行和调试。

在 Windows 系统的开始菜单的搜索栏中输入“IDLE”或“Python”,然后单击“Python 3.12 (64-bit)”进入 IDLE 界面,如图 1-20 所示。



图 1-20 单击 Python 3.12(64-bit)

用户可以在 Shell 界面中直接编写 Python 代码,并使用简单的 `print()` 函数来编写输出语句。在 Python 语言中,`print()` 是一个内置函数,用于向控制台输出内容。

例如,使用 `print()` 函数输出“Welcome to Binzhou”,按 Enter 键确认。

运行结果如下。

```
>>>print("Welcome to Binzhou")
Welcome to Binzhou
```

1.2.2 实现文件式编程界面

IDLE 除了支持交互式编写代码,还支持文件式编写代码。在交互式窗口中选择“File”→“New File”选项,创建并打开一个新的界面,如图 1-21 所示。

在打开的界面中编写代码:`print("Welcome to Binzhou")`,然后选择“File”→“Save As”

选项,将文件以“first_app”命名并保存,之后在该界面中选择“Run”→“Run Module”选项运行代码,如图 1-22 所示。

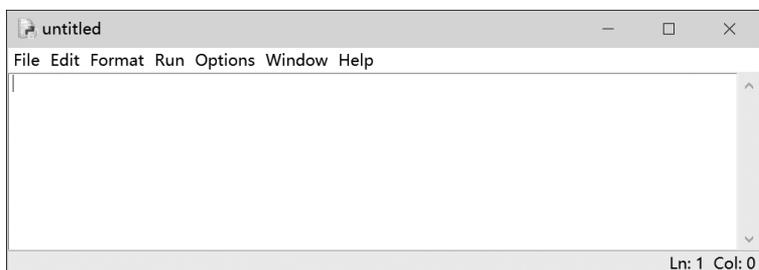


图 1-21 创建并打开一个新的界面

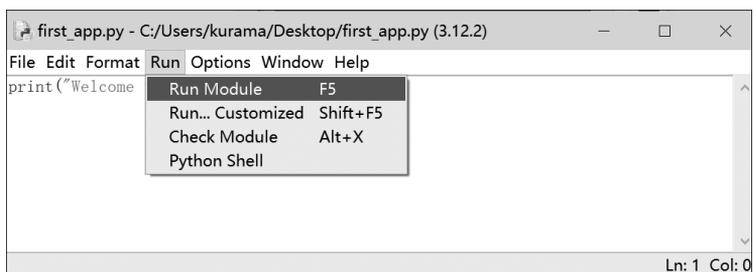


图 1-22 运行代码操作

运行结果如下。

```
Welcome to Binzhou
```



任务实现

使用 Python 编写一个猜随机数小游戏。游戏开始时,程序会随机生成一个 1 到 10 之间的整数,玩家需要输入一个数进行猜测;程序会根据玩家的猜测给出提示,告知他们输入的数字是过大、过小还是正确。如果玩家猜对了,则程序会显示“恭喜!你猜对了。”,并提示玩家猜测的次数,随后游戏结束。如果玩家没有猜对,则游戏继续,直到玩家猜对为止。

猜随机数小游戏的代码如下。

```
import random
print("-----欢迎来到随机数游戏-----")
# 生成随机数
number = random.randint(1, 10)
guess_count = 0
while True:
    # 获取用户输入
    guess = int(input("请输入一个 1 到 10 之间的整数:"))
```

```
if guess == number:
    print("恭喜! 你猜对了。")
break
elif guess < number:
    print("太小了,再试一次。")
else:
    print("太大了,再试一次。")
    guess_count += 1
print("你一共猜测了", guess_count, "次。")
```

运行结果如下。

```
-----欢迎来到随机数游戏-----
请输入一个 1 到 10 之间的整数:4
太大了,再试一次。
请输入一个 1 到 10 之间的整数:1
太小了,再试一次。
请输入一个 1 到 10 之间的整数:2
恭喜! 你猜对了。
你一共猜测了 3 次。
```

综合运用

以 PyCharm 开发环境为例,新建 Python 文件。双击桌面上的 PyCharm 快捷方式图标,进入导入配置文件的界面,如图 1-23 所示。

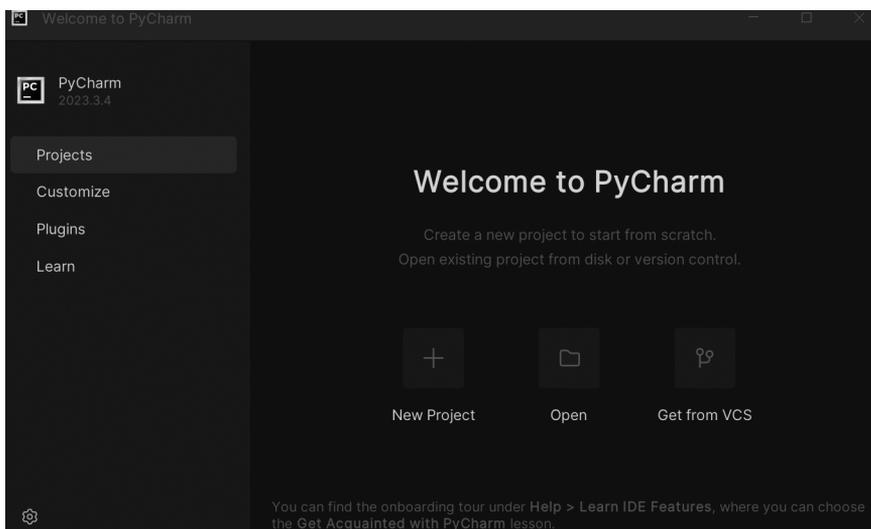


图 1-23 导入配置文件的界面

单击“New Project”按钮,创建一个 Python 项目 chapter01,选中项目名称“chapter01”并右击,然后在弹出的快捷菜单中选择“New”→“Python File”选项,如图 1-24 所示。

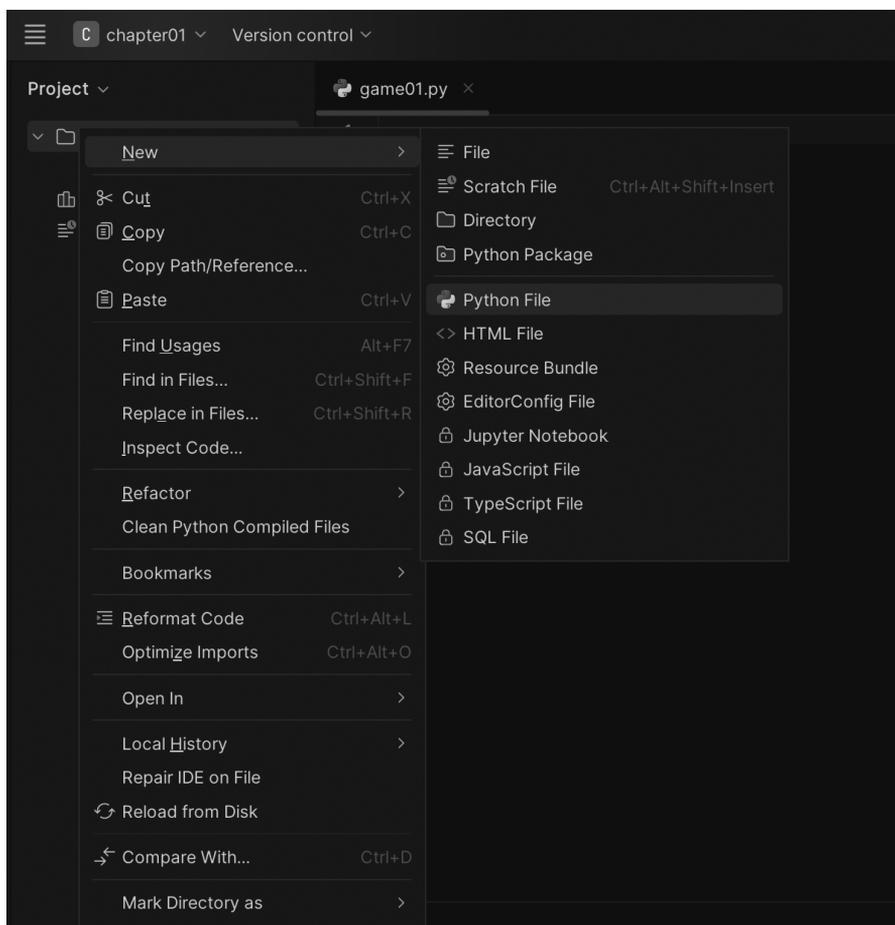


图 1-24 创建项目

在项目中创建一个 py 文件,并将该文件命名为“game01”,如图 1-25 所示。

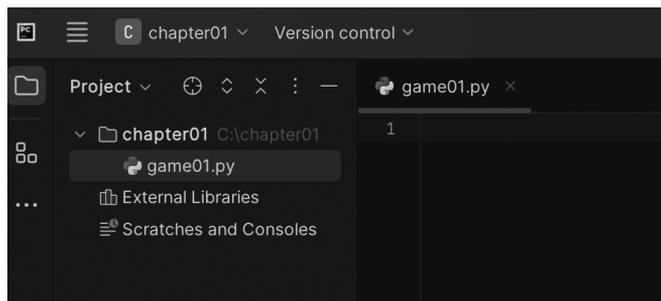


图 1-25 创建 game01.py 文件

在创建好的“game01.py”文件中编写代码,如图 1-26 所示。
运行“game01.py”文件,程序的运行结果会在 PyCharm 结果输出区进行显示。

```
1 import random
2 print("-----欢迎来到随机数游戏-----")
3 # 生成随机数
4 number = random.randint(1, 10)
5 guess_count = 0
6 while True:
7     # 获取用户输入
8     guess = int(input("请输入一个1到10之间的整数: "))
9     if guess == number:
10        print("恭喜! 你猜对了。")
11        break
12    elif guess < number:
13        print("太小了, 再试一次。")
14    else:
15        print("太大了, 再试一次。")
16    guess_count += 1
17 print("你一共猜测了", guess_count, "次。")
```

图 1-26 编写代码

程序的运行结果如下。

```
-----欢迎来到随机数游戏-----
请输入一个1到10之间的整数:3
太小了,再试一次。
请输入一个1到10之间的整数:4
恭喜! 你猜对了。
你一共猜测了2次。
```



课后寄语

通过本项目的学习,我们了解了 Python 语言的发展史、特点及应用领域,并掌握了 Python 开发环境的搭建方法。我们能够独立完成 IDLE 的下载与安装、扩展库的安装及模块导入,并且能够熟练掌握 IDLE 的基本操作。在安装软件的过程中,建议读者增强网络安全意识,尽量选择官方网站下载软件,以确保计算机和个人信息的安全,避免受到潜在威胁。



项目评价

教师结合本项目的学习情况完成项目评价,并将评价结果填入表 1-1 中。

(4)Python 代码可以在()操作系统中运行。

A. Windows

B. Linux

C. MacOS

D. 以上都可以

3. 简答题

简述 Python 环境变量的含义和作用。

4. 编程题

(1)使用 PyCharm 软件编写一段 Python 小程序并运行。

(2)使用 Anaconda 软件编写一段 Python 小程序并运行。