

项目 1 猜价赢大奖



学习目标

知识目标：

1. 了解 Python 的基本功能。
2. 认识 Python 的应用领域。
3. 掌握 Python 编码规范。

技能目标：

1. 能够熟练搭建集成开发环境 PyCharm。
2. 能够在 PyCharm 中实现 Python 程序的编写、运行。

素质目标：

1. 培养程序设计严谨、认真的职业素养。
2. 培养程序设计的基本逻辑思维能力。
3. 尊重知识产权,培养创新意识。

1.1 项目情景

某电子产品公司要举行一个大型庆祝活动,在活动现场设置了一个“猜价赢大奖”的活动。系统给定产品价格,现场展示产品实物,观众通过设计的游戏程序进行猜价,5 次以内能猜中,观众就获得此奖品实物。因活动现场奖品较多,参加活动的嘉宾人数众多,为保障活动顺利无误实施,现需要根据该场景需求开发一个程序。

该程序的核心功能是输入猜价与产品实际价格对比,最多猜 5 次,5 次及 5 次以内猜中即停。基于对该项目的需求分析,项目经理列出需要完成的任务清单,如表 1-1 所示。

表 1-1 项目 1 任务清单

任务序号	任务名称	知识储备
T1-1	猜价赢大奖	• Python 简介 • Python 的应用领域 • Python 开发环境搭建 • Python 程序开发运行流程

1.2 相关知识

1.2.1 Python 简介

Python 是一种高层次、解释型的编程语言,具有解释性、编译性、互动性和面向对象的特点。

Python 由吉多·范罗苏姆在 20 世纪 90 年代初设计,最初是作为 ABC 语言的一种替代品。它以其高效的高级数据结构和对面向对象编程的支持而受到欢迎。

(1) Python 的设计思想与特点。Python 的设计强调代码的可读性和简洁的语法,这使它既适合专业的开发人员,也适合编程初学者。它支持多态、异常处理和多重继承等高级面向对象编程(object oriented programming,OOP)概念。

(2) Python 的应用领域。Python 在多个领域都有广泛的应用,包括网络开发、数据分析、人工智能、科学计算等。它的多样性和灵活性使其成为当今最受欢迎的编程语言之一。

(3) Python 的发展趋势。随着技术的不断进步,Python 也在不断发展。它拥有一个活跃的开发社区,不断有新的库和工具被开发出来,以适应不断变化的技术需求和市场趋势。

Python 以其易学性、强大的功能和广泛的应用,成为当今编程世界中的一颗璀璨之星。无论你是想要进入编程世界,还是希望扩展你的技能集,Python 都值得考虑。

微课 1-1: Python 简介



1.2.2 Python 的应用领域

Python 的应用领域非常广泛,涵盖了从 Web 开发到科学计算的各个方面。

1. Web 开发

Python 在 Web 开发领域非常流行,主要得益于其丰富的框架资源,如 Django、Flask 和 Pyramid 等。这些框架提供了全面的工具集,包括路由、模板引擎、ORM(对象关系映射)以及表单处理等,使得快速开发复杂的 Web 应用成为可能。

2. 数据科学与机器学习

数据科学是 Python 的一大亮点。Pandas、NumPy、SciPy 和 Matplotlib 等库为数据分析和可视化提供了强大支持。Scikit-learn、TensorFlow 和 PyTorch 等库则支持机器学习和深度学习模型的构建和训练。

3. 人工智能

Python 的 AI 库使得它在人工智能领域也非常受欢迎。自然语言处理(NLP)可以使用

NLTK 或 spaCy 等库进行文本分析和处理。计算机视觉可以使用 OpenCV 等库来处理图像和视频数据。

4. 网络爬虫

Python 的 requests 库和 BeautifulSoup 库使得爬取网页数据变得简单,广泛用于数据挖掘和市场分析。

5. 游戏开发

虽然不如 C++ 等语言在 3D 游戏开发中流行,但 Python 的 Pygame 库提供了制作 2D 游戏的所需功能,对于入门级或小型项目而言,Python 是一个不错的选择。

6. 科学计算与教育

Python 在科学计算领域也有着广泛应用。SciPy 提供了数学算法和便利的函数,SymPy 则用于符号计算。Python 也常用于教学,因为它被认为是一种更接近于“自然语言”的编程语言,学生可以更容易地学习和理解。

7. 金融

在金融领域,Python 用于量化交易、风险管理等,其数据分析和计算能力在这一领域尤为重要。Pandas 等库提供了对时间序列数据的高效处理能力,而如 QuantLib 等库则专门用于金融工程。

8. 自动化与脚本编写

Python 因其简单易用的特性,常被用于编写各种自动化脚本,从而提高工作效率。它可以帮助用户自动处理日常任务,如文件管理、系统配置和网络管理等。

9. 云计算与 DevOps

Python 与多个云服务平台兼容,可以用来开发和管理云基础设施,以及在云平台部署应用和服务。在 DevOps 中,Python 脚本常用于持续集成和持续部署(CI/CD)流程中。

10. 物联网

Python 的简单性和跨平台特性使其成为物联网项目的热门选择。它可以运行在树莓派等微型计算机上,用于控制和数据采集。

Python 的多样性和灵活性,加上其庞大的社区支持和丰富的库资源,使其成为当今最受欢迎的编程语言之一。无论是初学者还是专业人士,Python 都是一个值得学习的编程语言。

微课 1-2: Python 的应用领域



1.2.3 Python 开发环境搭建

1. Python 解释器

Python 解释器的下载、安装及环境配置是使用 Python 进行编程不可或缺的基础步骤，它们为 Python 代码的编写与执行搭建起稳固的平台。

(1) 确定开发所用设备操作系统。在下载 Python 解释器之前，需要确认开发使用设备的操作系统（Windows、macOS 或 Linux），以确保下载正确的安装包。本书中程序开发所用设备操作系统是 Windows 操作系统，后续将以此为例。

(2) 下载安装。访问 Python 的官方网站 <https://www.python.org/>，单击 Downloads 菜单，根据操作系统选择相应的安装包进行下载。本书选择的是适合 Windows 系统的 Python 3.11.8 下载，如图 1-1 所示。



图 1-1 版本选择

下载完成后，运行安装程序并按照提示完成安装过程。在安装过程中，需要选择 Python 解释器的安装路径，确保 IDE(integrated development environment,集成开发环境)能够正

确识别和使用 Python 环境,以便顺利进行开发和调试工作。为了方便在命令行中直接运行 Python 命令,建议勾选 Add python.exe to PATH 选项,如图 1-2 所示。这一操作将 Python 的可执行文件路径添加到系统的环境变量中。

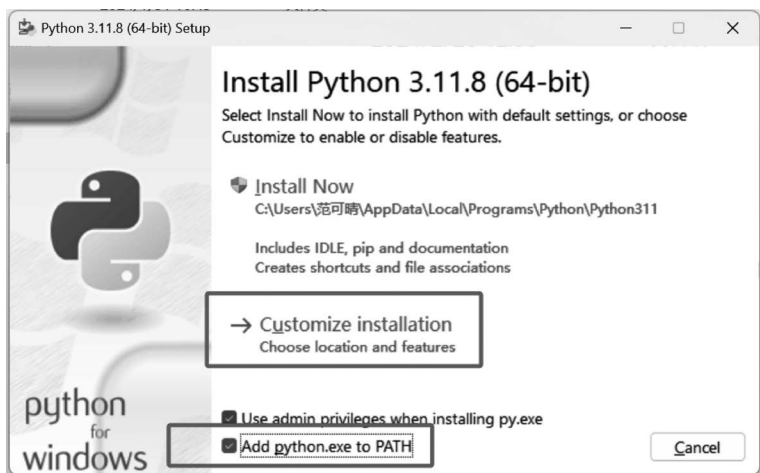


图 1-2 安装配置

配置安装选项,如图 1-3 所示。

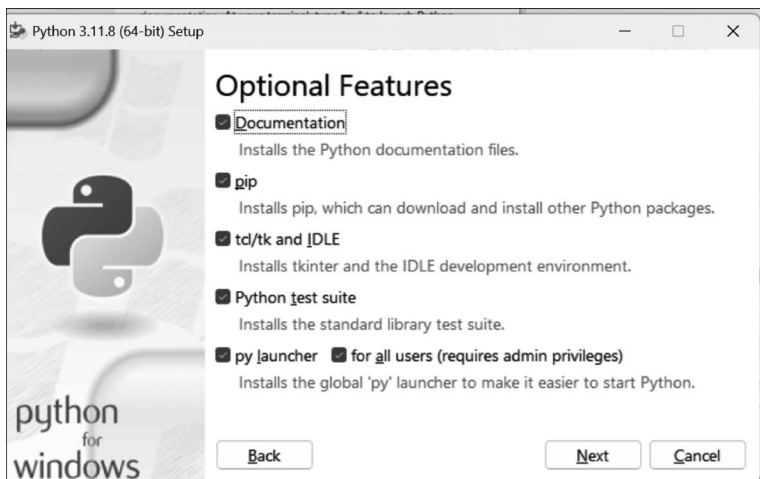


图 1-3 安装选项

选择安装位置,如图 1-4 所示。

单击图 1-4 中的 Install 按钮,执行安装命令后,显示安装进度,如图 1-5 所示。

接着出现 Setup was successful 界面,如图 1-6 所示,表示安装成功。

(3) 配置环境变量。如果系统没有自动配置环境变量,需要手动将 Python 的安装路径添加到系统的 PATH 环境变量中。请按照以下步骤操作:

- ① 找到 Python 的安装路径。本书的案例安装路径是 D:\python311。
- ② 右击“计算机”或“此电脑”,选择“属性”命令。

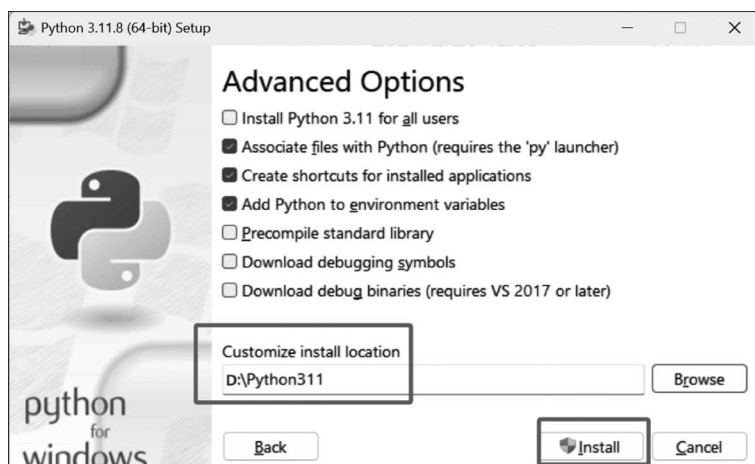


图 1-4 安装位置

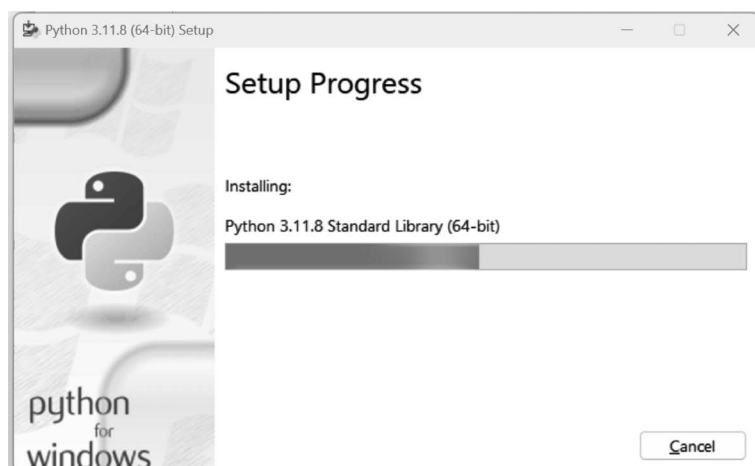


图 1-5 安装中

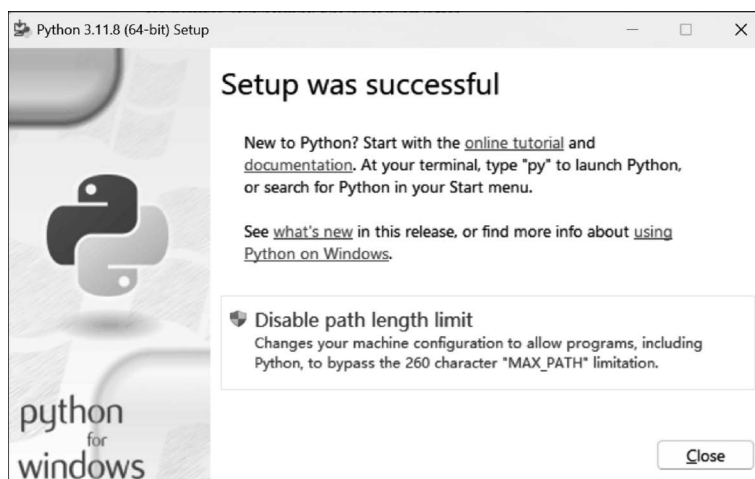


图 1-6 安装成功

③ 在弹出的窗口中选择“高级系统设置”命令,打开“系统属性”对话框,然后在“高级”选项卡中单击“环境变量”按钮。

④ 在弹出的“环境变量”对话框中选中“系统变量”中的变量 Path,然后单击“编辑”按钮,打开“编辑环境变量”对话框。

⑤ 在弹出的对话框中单击“新建”按钮,在光标所在位置输入 Python 的安装路径(如 D:\python311)。

⑥ 单击“确定”按钮,保存并关闭所有打开的对话框。

现在已经成功配置了 Python 环境变量。可以在命令提示符或 PowerShell 中输入 python 命令来启动 Python 解释器。

(4) 试用。在 Windows 命令提示符或 PowerShell 中输入 python 命令来启动 Python 解释器,如图 1-7 所示,说明安装配置成功。

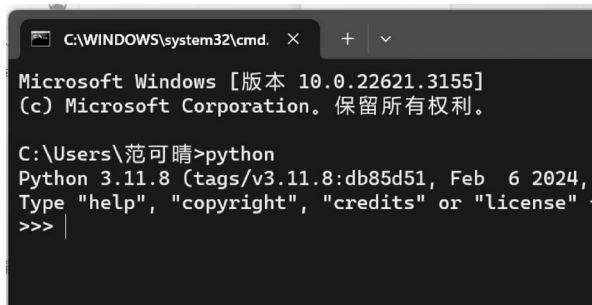


图 1-7 启动 Python 解释器

在“>>>”后面可以输入命令,例如输入:

```
print("Hello World!")
```

按 Enter 键会显示输出“Hello World!”,如图 1-8 所示。

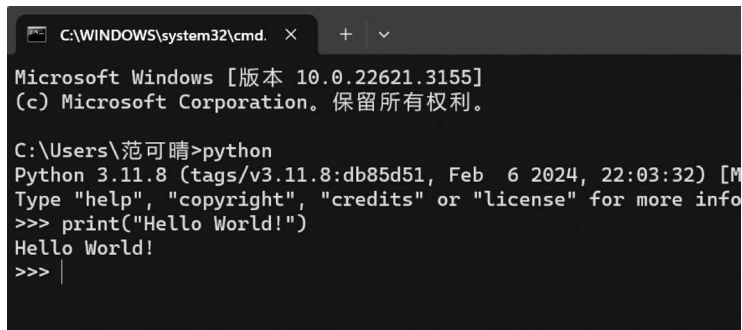


图 1-8 命令窗口

Python 官方提供了一个简单的集成开发和学习环境。IDLE(integrated development and learning environment,集成开发和学习环境)被设计成一个轻量级的工具,特别适合于初学者学习和使用 Python 语言。按键盘上的 Windows 键打开 Windows 菜单,如图 1-9 所示,从菜单中选择 IDLE(Python 3.11 64-bit)命令打开 Python 开发环境。

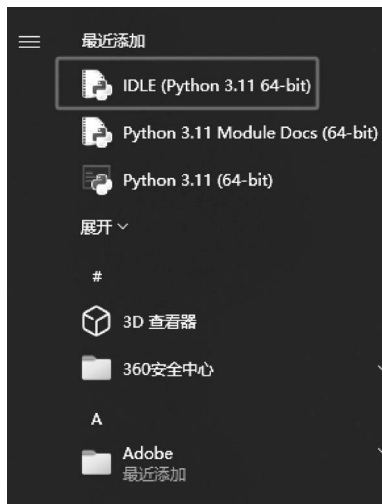


图 1-9 IDLE

在 IDLE 中可以交互式输入 Python 语句并执行,如图 1-10 所示,也可以选择 File→New File 命令打开 Python 程序编辑窗口,编写由多行语句构成的 Python 程序。



图 1-10 IDLE 应用

微课 1-3: Python 解释器安装



2. PyCharm

PyCharm 是一个功能强大且跨平台的 Python IDE,适用于 Windows、macOS 和 Linux。它提供代码补全、语法高亮、调试、性能分析、版本控制等功能,旨在提高 Python 开发效率。PyCharm 是 Python 开发者的理想选择。

(1) 下载。访问 PyCharm 官网 <https://www.jetbrains.com/pycharm> 的下载页面, 会看到两个主要的版本可供选择: 专业版(Professional)和社区版(Community)。专业版是付费版本, 提供更全面的功能; 而社区版则是免费的, 适合学生和新入门的开发者使用。本书选择 Windows 下的 Community 版 2023 下载使用。

(2) 安装。下载完成后, 双击安装包, 启动安装向导, 如图 1-11 所示。

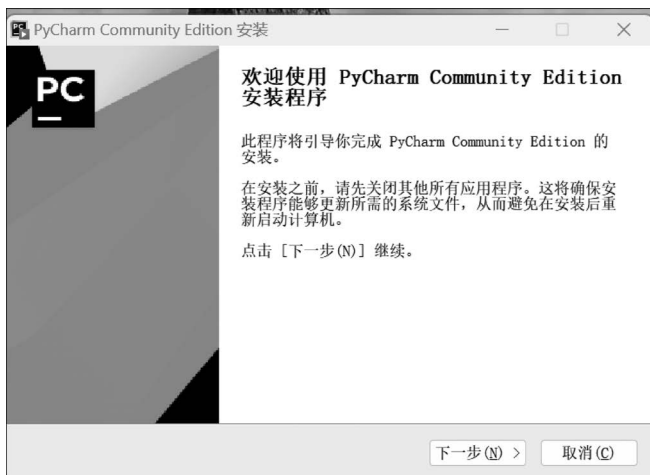


图 1-11 启动 PyCharm 安装

在安装过程中, 可以选择安装目录, 如图 1-12 所示。建议选择非系统盘(如 D 盘或 E 盘)以避免占用系统资源。



图 1-12 选择安装位置

安装过程中的安装选项设置如图 1-13 所示。建议选择 PyCharm Community Edition、“添加 'bin' 文件夹到 PATH”和“.py”3 个选项, 方便开发使用。

选择开始菜单目录, 如图 1-14 所示, 使用默认的 JetBrains。单击“安装”按钮, 等待安装完成。

安装程序结束, 如图 1-15 所示。可以重启系统, 开始使用 PyCharm。



图 1-13 安装选项



图 1-14 选择开始菜单目录



图 1-15 安装完成