高等院校计算机应用系列教材

Python基础教程
（微课版）

林志灿　涂晓彬　林智欣　主编

清華大學出版社
北　京

内 容 简 介

本书专门针对Python新手量身定做，涵盖Python 3实际开发中经常用到的重要知识点，内容主要包括Python 语言的类型和对象、运算符和表达式、编程结构和控制流、函数、序列、多线程编程、正则表达式、面向对象编程、文件和目录操作、数据库编程、网络编程和邮件收发、Django 框架和项目范例。在介绍知识点的过程中，理论和实践相结合。书中还安排了不少实践示例，以帮助读者巩固所学，能够学以致用。

本书内容丰富、结构合理、思路清晰、语言简洁流畅、示例翔实。本书可作为高等院校 Python 程序设计课程的教材，也可作为 Python Web 应用开发人员的参考资料。

本书配套的电子课件、实例源文件、习题答案和思维导图可以到 http://www.tupwk.com.cn/downpage 网站下载，也可以扫描前言中的“配套资源”二维码获取。扫描前言中的“看视频”二维码可以直接观看教学视频。

图书在版编目(CIP)数据

Python基础教程：微课版 / 林志灿，涂晓彬，林智欣主编. —北京：清华大学出版社，2023.9
高等院校计算机应用系列教材
ISBN 978-7-302-64581-8

Ⅰ. ①P… Ⅱ. ①林… ②涂… ③林… Ⅲ. ①软件工具—程序设计—高等学校—教材 Ⅳ. ①TP311.561

中国国家版本馆 CIP 数据核字 (2023) 第 180342 号

责任编辑：胡辰浩
封面设计：高娟妮
版式设计：孔祥峰
责任校对：成凤进
责任印制：宋 林

出版发行：清华大学出版社
网 址：https://www.tup.com.cn，https://www.wqxuetang.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-83470000 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市龙大印装有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张：**21.5 **字 数：**510 千字
版 次：2023 年 11 月第 1 版 **印 次：**2023 年 11 月第 1 次印刷
定 价：86.00 元

产品编号：101493-01

前言

Python是一种解释型的、面向对象的、带有动态语义的高级程序设计语言。在使用Python时，开发人员可以保持自己的代码风格，可以使用更清晰易懂的程序来实现想要的功能。对于一个没有任何编程经历的人来说，既简单又强大的Python就是完美的选择。

随着云计算、大数据、人工智能等技术的迅速崛起，对Python人才的迫切需求和现实中Python人才的匮乏让长期沉默的Python语言瞬间备受青睐，本书作为教材，可以说是应运而生。随着技术的快速发展，Python的版本也在更新迭代，不断推出新版本，目前主流的版本为Python 3系列，每推出一个Python新版本都会增加不少新特性。本书基于Python 3.11编写而成，可以满足想学习和了解Python最新版本及其特性的读者。

本书专门为Python新手量身定做，是编者学习和使用Python过程中的体会和经验总结，涵盖实际开发中的所有重要知识点，内容详尽，代码可读性和可操作性强。

本书主要介绍Python语言的类型和对象、运算符和表达式、编程结构和控制流、函数、序列、多线程编程、正则表达式、面向对象编程、文件和目录操作、数据库编程、网络操作和邮件收发等。在讲解每个知识点时，先讲解理论，后列举实际示例，各章还安排了习题，以帮助读者将所学应用到实际中，做到学以致用。

本书的特色是，使用通俗易懂的描述和丰富的代码示例，提高本书的可读性，将复杂问题简化，使学习Python变得轻松。

本书共分14章，各章内容安排如下。

第1章主要介绍Python的起源、应用场合、前景以及Python 3的新特性。

第2章主要介绍Python的基础知识，为后续章节学习相关内容做铺垫。

第3章重点介绍字符和序列(列表、元组、集合等)。

第4章重点介绍流程控制语句，主要包括分支结构、循环结构。

第5章主要介绍正则表达式。

第6章重点介绍函数。函数是组织好的、可重用的、用来实现单一或相关功能的代码段。

第7章重点介绍面向对象编程技术。Python从设计之初就是一门面向对象的语言，提供了一些语言特性以支持面向对象编程。

第8章重点介绍模块，从import语句开始介绍，然后逐步深入。

第9章介绍如何处理各种异常和错误，以及创建和自定义异常类。

第10章重点介绍如何使用Python在硬盘上创建、读取和保存文件，以及目录的创建、删除、遍历等。

第11章主要介绍Python多线程编程。

第12章重点介绍Python数据库编程，并实现简单的增删改查操作。

第13章重点介绍Python网络编程。

第14章介绍如何使用Django框架创建一个投票管理系统，以及如何打包和发布该系统。

本书分为14章，由闽南理工学院的林志灿、涂晓彬、林智欣合作编写完成。其中林志灿编写了第1、3、5、12、14章，涂晓彬编写了第2、4、6、8、11章，林智欣编写了第7、9、10、13章。由于作者水平有限，书中难免存在不足之处，欢迎广大读者批评指正。我们的信箱是992116@qq.com，电话是010-62796045。

本书配套的电子课件、实例源文件、习题答案和思维导图可以到http://www.tupwk.com.cn/downpage网站下载，也可以扫描下方的“配套资源”二维码获取。扫描下方的“看视频”二维码可以直接观看教学视频。

扫描下载

配套资源

扫一扫

看视频

编　者

2023年9月

目录

第 1 章

初识 Python

Python语言伴随人工智能的兴起而得到快速蓬勃的发展，它是一种解释型的、面向对象的、动态数据类型的高级程序设计语言。像Perl语言一样，Python源代码同样遵循GPL协议。Python优雅的语法和动态类型，再结合它的解释性，使其在大多数平台的许多领域成为编写脚本或开发应用程序的理想语言。从云端、客户端到物联网终端，Python无处不在，它已成为人工智能首选的编程语言。

本章的学习目标：

- 了解Python语言的起源与应用领域；
- 熟悉Python语言的版本，Python的下载、安装和启动；
- 了解Python常用的开发工具；
- 熟悉进行Python编程的几种方式；
- 了解Python初学者可能遇到的常见问题。

1.1 Python概述

Python是一门跨平台、开源、免费的解释型高级动态编程语言。除了解释执行，Python还支持伪编译，通过将源代码转换为字节码来优化程序、提高运行速度以及对源代码进行保密，并且支持使用py2exe、pyinstaller、cx_Freeze或其他类似工具将Python程序及其所有依赖库打包为扩展名为.exe的可执行程序，从而可以脱离Python解释器环境和相关依赖库而在Windows平台上独立运行；Python支持命令式编程、函数式编程，完全支持面向对象程序设计。也有人喜欢把Python语言称为“胶水语言”，因为它可以把多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。

目前Python的主流版本为Python 3.x。在选择Python版本时，一定要先考虑清楚使用Python来做哪方面的开发，有哪些扩展库可以使用，这些扩展库最高支持哪个版本的

Python，等等，确定完之后再做出选择，这样就不会把时间浪费在Python和各种扩展库的反复安装上。

1.1.1 Python起源

1989年圣诞节期间，在阿姆斯特丹，Guido van Rossum为了打发圣诞节的无趣，决定开发一种新的脚本解释程序，作为ABC 语言的一种继承。之所以选用Python作为名称，是因为Guido本人是英国一个名叫Monty Python的喜剧团体的爱好者。

ABC语言是由Guido参与设计的一种教学语言。就Guido本人看来，ABC这种语言非常优美和强大，是专门为非专业程序员设计的。但是ABC语言最终并没有成功，究其原因，Guido认为是其非开放性造成的。Guido决定在Python中避免这一错误。同时，他还想实现在ABC语言中闪现过但未曾实现的特性。

就这样，Python诞生了。可以说，Python是从ABC语言发展而来的，主要受到Modula-3(另一种相当优美且强大的语言，为小型团体而设计)的影响，并且结合了UNIX shell和C语言的使用习惯。

自2004年以后，Python的使用率呈线性增长。2011年1月，Python在TIOBE编程语言排行榜上被评为2010年度最受欢迎的语言。

由于Python语言的简洁性、易读性及可扩展性，在国外用Python做科学计算的研究机构日益增多，一些知名大学已采用Python来教授程序设计课程。例如卡内基-梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用Python语言讲授。众多开源的科学计算软件包都提供了Python的调用接口，例如著名的计算机视觉库OpenCV、三维可视化库VTK、医学图像处理库ITK。而Python专用的科学计算扩展库就更多了，例如NumPy、SciPy和matplotlib三个十分经典的科学计算扩展库，它们分别为Python提供快速数组处理、数值运算及绘图功能。因此，Python语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。

1.1.2 Python版本

Python 作为一种语言，它也是随时间而逐步演进的：

- 早期版本的 Python 被称作是 Python 1；
- 在 2000 年，Python 2 的第一个版本发布，它目前仍在使用中；
- 2008 年 Python 3 的第一个版本发布，它是目前的最高版本。

Python 2 于 2000 年 10 月 16 日发布，其最后一个版本是 2.7。Python 2.7 在 2020 年 1 月 1 日已无法得到 Python 社区的支持，所以其状态类似于 Windows XP 目前的状态。

Python 3 于 2008 年 12 月 3 日发布，目前的版本是 3.11.2。Python 3 是目前最活跃的版本，基本上新开发的 Python 代码都会支持 Python 3。

Python 4 是未来的版本，目前还处于萌芽状态，至今没有相关发布。本教程未涉及 Python 4 的相关内容。

Python 3 和 Python 2 并不是完全兼容的，即在 Python 2 中可以运行的代码并不一定可

以在 Python 3 中运行。Python 社区意识到了这个问题，所以在 Python 3 中也提供了一些工具，如 2to3，这些工具可以帮助用户将 Python 2 编写的代码转换成 Python 3 编写的代码。

现阶段来看，多数 Python 库都完成了向 Python 3 迁移的任务，本书的代码也将以 Python 3 为主。建议读者安装 Python 3.5 及以上版本来练习本书中的代码示例。

1.1.3 Python应用

1. 常规软件开发

Python支持函数式编程和面向对象编程，能够承担任何类型软件的开发工作。因此，常规的软件开发、脚本编写、网络编程等都属于标配能力。

2. 科学计算

随着NumPy、SciPy、matplotlib等众多扩展库的开发，Python越来越适合用于科学计算、绘制高质量的2D和3D图形。与科学计算领域最流行的商业软件MATLAB相比，Python是一门通用的程序设计语言，比MATLAB所采用的脚本语言的应用范围更广，有更多的扩展库提供支持。虽然MATLAB中的许多高级功能和工具箱目前还是无法替代的，不过在日常的科研中仍然有很多的工作可由Python代劳。

3. 自动化运维

该功能几乎是Python应用的自留地。作为运维工程师首选的编程语言，Python在自动化运维方面已深入人心，比如Saltstack和Ansible都是大名鼎鼎的自动化平台。

4. 云计算

开源云计算解决方案OpenStack就是基于Python开发的。

5. Web开发

基于Python的Web开发框架有很多，比如耳熟能详的Django，还有Tornado和Flask。其中，Python+Django架构组合的应用范围非常广，开发速度非常快，学习门槛低，能够帮助开发人员快速搭建可用的Web服务。

6. 网络爬虫

网络爬虫也称网络蜘蛛，是大数据行业中获取数据的核心工具。没有网络爬虫自动地、不分昼夜地、高智能地在互联网上爬取免费的数据，那些大数据相关的公司恐怕要少四分之三。能够编写网络爬虫的编程语言有不少，但Python绝对是其中的主流之一，其Scrapy爬虫框架的应用非常广泛。

7. 数据分析

在大量数据的基础上，结合科学计算、机器学习等技术，对数据进行清洗、去重、规格化和有针对性的分析是大数据行业的基石。Python是进行数据分析的主流语言之一。

8. 人工智能

Python在人工智能大范畴领域内的机器学习、神经网络、深度学习等方面都是主流的编程语言，得到广泛的支持和应用。

1.2 搭建Python开发环境

在Windows、Linux、macOS操作系统上，都可以搭建Python开发环境。

1.2.1 下载Python

要搭建Python开发环境，首先必须下载和安装相应的工具。这些工具可免费下载。

(1) 可以从Python官方网站下载安装包。

(2) 也可以从网上下载ActivePython组件包。ActivePython是对Python核心模块和常用模块的二进制封装，是ActiveState公司发布的Python开发环境。ActivePython使得Python的安装更容易，并且可以应用在各种操作系统上。ActivePython包含一些常用的Python扩展，以及Windows环境下的编程接口。如果是Windows用户，下载msi包安装即可；如果是UNIX用户，下载tar.gz包直接解压即可。

(3) Python的IDE具体包括PythonWin、Eclipse+PyDev插件、Komodo、EditPlus、PyCharm等。

1.2.2 安装Python

1. 在Windows操作系统上安装Python

从官网的Windows发行版本列表中找到需要的安装程序，如图1-1所示。32位操作系统下载32-bit安装包，64位操作系统下载64-bit安装包。

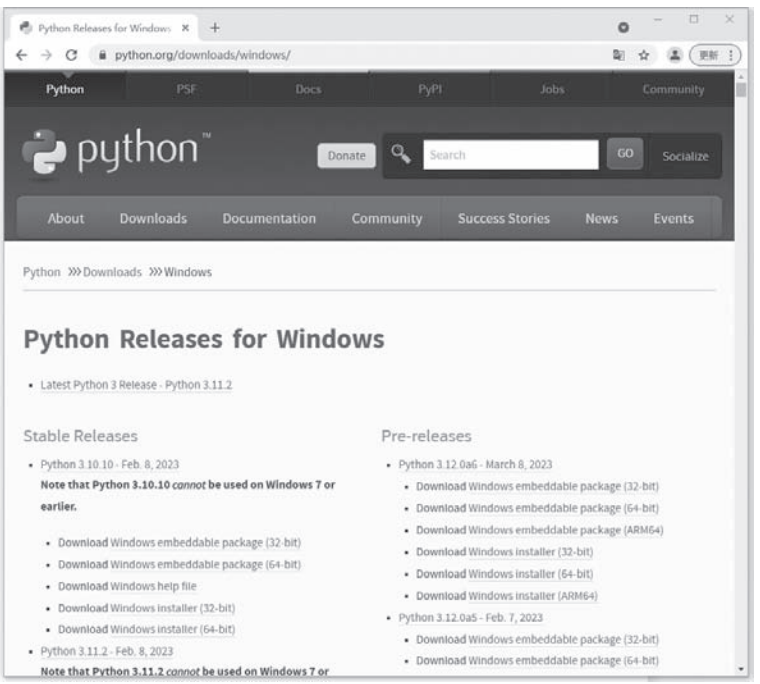

图1-1 Python下载界面

本书使用的是Python 3.11.2。单击Latest Python 3 Release - Python 3.11.2，打开下载界面，下拉界面至Files列表，然后单击Windows installer(64-bit)下载项开始下载，如图1-2所示。

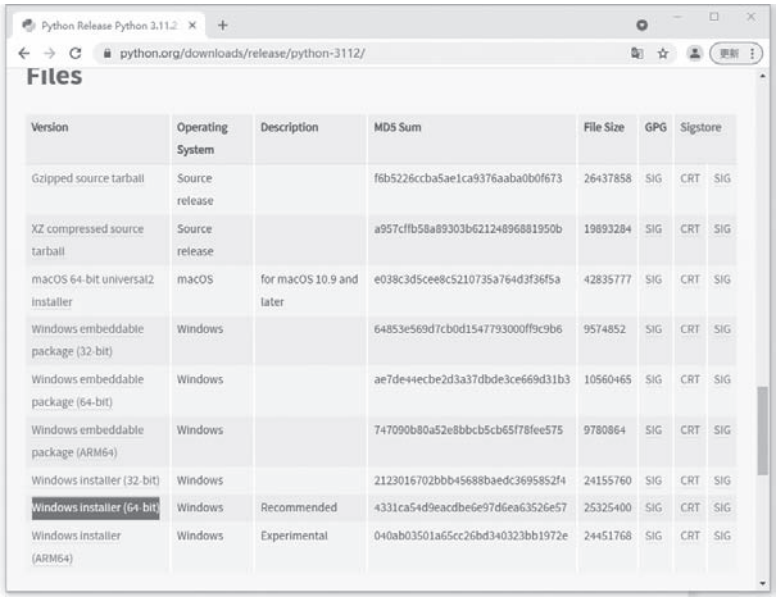

图1-2　下载Python 3.11.2

下载Python安装程序后，双击运行安装程序，初始界面如图1-3所示。选中底部的Add python.exe to PATH复选框，然后单击Install Now按钮。系统可能要求确认对系统所做的更改，单击OK按钮接受这些更改，进入安装界面，如图1-4所示。如果需要修改安装路径，可以单击Customize installation进行修改。

图1-3　Python安装程序的初始界面

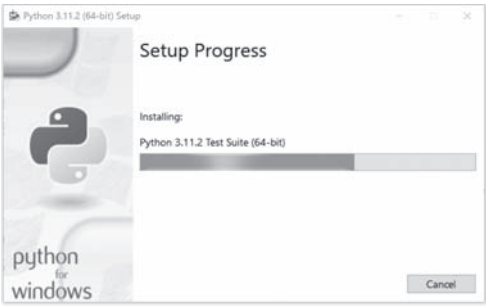

图1-4　Python安装界面

在安装过程的最后，将看到如图1-5所示的界面，单击Close按钮关闭即可。

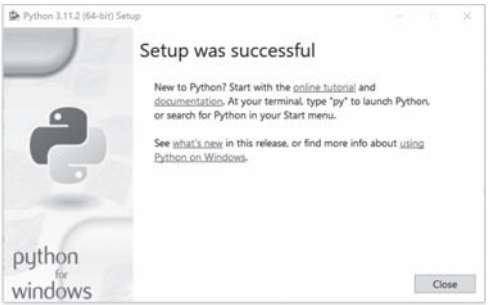

图1-5　安装成功界面

2. 在Linux操作系统上安装Python

在Linux操作系统上安装Python要简单得多，这里以Ubuntu Linux为例。Python在Ubuntu下有两种常用安装方法。

- 通过Ubuntu官方的apt工具包安装。

- 通过编译Python源代码安装。

例如，采用apt安装方式时，输入以下命令：

```
sudo apt-get install python3.11.2
```

apt将Python安装包下载到本地并自动进行安装。Python被默认安装到usr/local/lib/python311目录中。安装完毕后，可以直接输入python命令来查看Python版本号或是否安装成功。

1.2.3 启动Python

安装成功后，打开Windows的命令提示符窗口，输入python命令，即可显示当前Python的版本号，并进入Python交互模式，如图1-6所示。

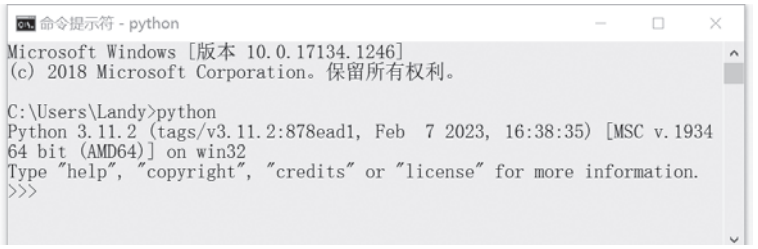

图1-6 在命令提示符窗口中启动Python

在Python交互模式下可以直接输入python命令并执行。例如，输入以下命令：

```
>>> print("hello")
hello
>>> print("hello world")
hello world
>>> x=3
>>> y=4
>>> x+y
7
```

命令提示符窗口如图1-7所示。

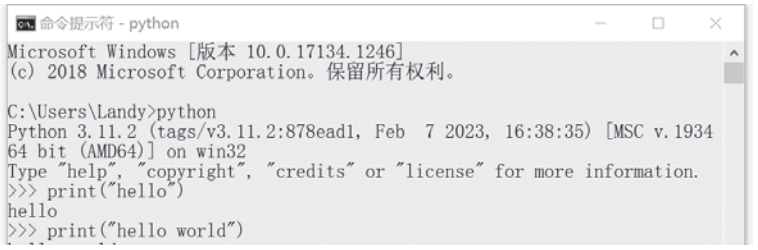

图1-7 直接输入python命令执行

在命令提示符窗口中使用交互模式执行python命令，只适用于测试功能。当关闭窗口时，所有输入的命令和执行结果均无法重现，因此，对于一些需要重复使用的代码，显得无能为力。

因此，本书使用IDLE(Integrated Development Learning Environment，集成开发学习环境)来讲解Python的功能，IDLE又称Eric Idle(Monty Python剧组的一位成员)。

Python安装成功后，同时会安装IDLE。在Windows的【开始】菜单的所有程序中可以找到IDLE的启动项，如图1-8所示。单击IDLE(Python 3.11 64-bit)选项，打开IDLE窗口，如图1-9所示。

图1-8　IDLE启动项

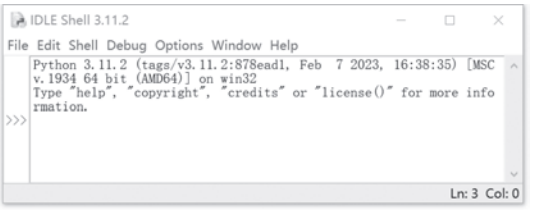

图1-9　IDLE窗口

在IDLE中，可通过两种方式与Python交互：shell(输入的Python命令将立即执行)或文本编辑器(允许创建程序代码文档)。目前几乎所有的操作系统都支持IDLE。

在macOS或Linux操作系统中，打开终端，输入idle，按下Enter键即可。

shell内有引擎，IDLE程序包含Python引擎。Python引擎运行Python程序。与命令提示符窗口一样，IDLE Python Shell也可以执行输入的Python命令，将它们输入Python引擎，然后显示结果。

1.2.4　多版本Python及虚拟环境的安装

实际开发中，有可能需要用到不同的Python版本或版本库，这难免会涉及不同Python版本的安装以及不同扩展库的安装。本节以Windows环境为例，介绍多版本Python及虚拟环境的安装。

1. 多版本Python的安装

前面已经安装了Python 3.11.2，这里介绍一下Python 3.7.9的安装。

(1) 首先从Python官网下载Python 3.7.9的安装包，然后双击运行，打开如图1-10所示的界面。

(2) 单击Customize installation按钮，打开Optional Features界面，如图1-11所示。

图1-10　安装Python 3.7.9的初始界面

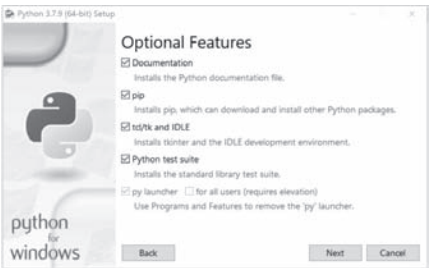

图1-11　Optional Features界面

(3) 单击Next按钮，打开Advanced Options界面，设置安装路径，如图1-12所示。从中可以看出，不能添加路径到系统变量，先单击Install按钮进行安装，后面再专门解决。

(4) 单击Next按钮进入安装过程，如图1-13所示。安装完毕后，单击Finish按钮完成安装。至此，Python 3.7.9已安装完成。

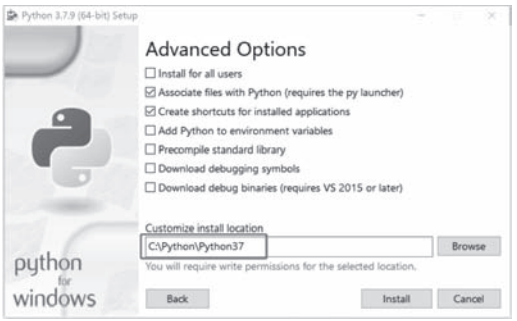

图1-12　Advanced Options界面

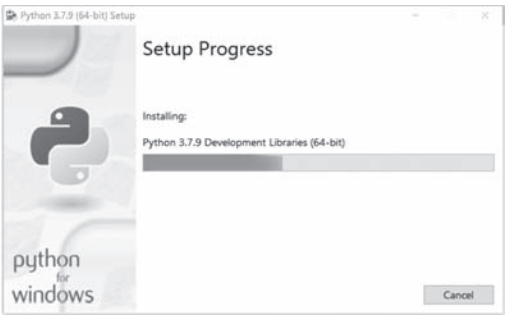

图1-13　安装Python 3.7.9

(5) 此时运行cmd命令进入DOS命令窗口，输入python命令显示Python 3.11.2版本，因为没有把Python 3.7.9版本添加到环境变量中。下面来设置系统环境变量。

打开【控制面板】|【系统和安全】|【系统】，单击【高级系统设置】按钮，如图1-14所示。

图1-14　单击【高级系统设置】按钮

(6) 打开【系统属性】对话框，单击【环境变量】按钮，如图1-15所示。

图1-15　单击【环境变量】按钮

(7) 打开【环境变量】对话框，如图1-16所示，在【系统变量】列表框中选择Path，然后单击【编辑】按钮。

(8) 弹出“编辑环境变量”对话框，单击【新建】按钮，分别添加C:\Python\Python37和

C:\Python\Python37\Scripts到环境变量中，如图1-17所示。

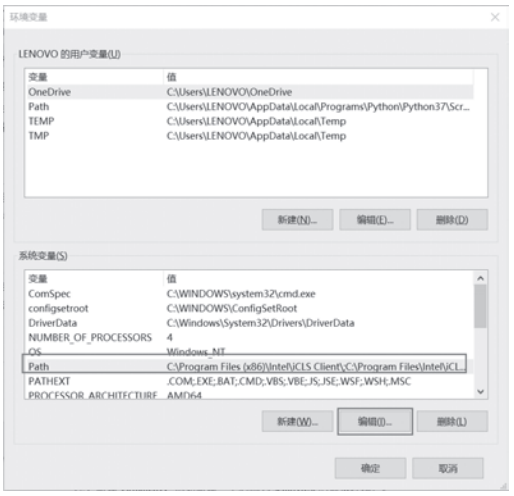

图1-16　【环境变量】对话框

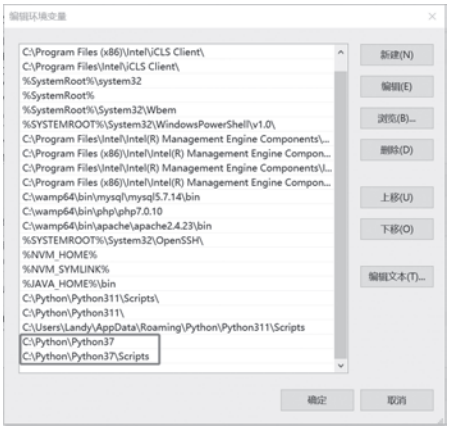

图1-17　添加路径

(9) 找到Python的安装目录，分别将Python37和Python311子目录中python.exe和pythonw.exe的名称修改为python37.exe、pythonw37.exe和python311.exe、pythonw311.exe。

然后运行cmd命令，输入python37即可运行Python 3.7.9版本，输入python311即可运行Python 3.11.2版本，如图1-18所示。

图1-18　执行python37和python311命令

(10) Python安装包需要用到包管理工具pip，但是当同时安装多版本Python时，pip只是其中一个版本，以下将提供一种修改方式，重新安装两个版本的pip，使得两个Python版本的pip能够共存。

在DOS命令窗口中输入以下命令：

```
python311 -m pip install --upgrade pip --force-reinstall
python37 -m pip install --upgrade pip --force-reinstall
```

会显示重新安装成功。

2. Python虚拟环境的安装

在Windows操作系统中安装Python虚拟环境的步骤如下。

(1) 安装virtualenv镜像，执行以下命令(pip3.11为Python311下的pip)：

```
pip3.11 install virtualenv
```

(2) 新建virtualenv，例如，在Python311安装目录下新建一个名为scrapytest的虚拟环境：

```
virtualenv  scrapytest
```

(3) 使用cd命令进入C:\Python\Python311\scrapytest\Scripts目录，直接输入activate命令并执行，进入虚拟环境，如图1-19所示。进入虚拟环境，就可以运行Python进行测试了。

图1-19　进入虚拟环境

(4) 当安装多个Python版本时，可以更改虚拟环境的Python版本，例如，要为虚拟环境更改Python版本到Python 3.7，命令如下：

```
virtualenv -p C:\Python\Python37\python37.exe C:\Python\Python311\scrapytest
```

(5) 当不需要使用虚拟环境时，可以退出虚拟环境，执行以下命令：

```
deactivate.bat
```

(6) 如果虚拟环境过多，管理起来会不太方便。这时，可以使用专门的虚拟环境管理包virtualenvwrapper进行管理，pip安装如下(此处调用的是Python311下的pip311)：

```
pip3.11 install virtualenvwrapper
```

在Windows操作系统中安装如下：

```
pip3.11 install virtualenvwrapper-win
```

(7) 安装完毕后，在C:\Python\Python311下建立workon文件夹，然后设置环境变量WORKON_HOME为C:\Python\Python311\workon。设置完成后，可以使用virtualenvwrapper管理虚拟环境，这时新建虚拟环境的命令格式如下：

```
mkvirtualenv virtual_name
```

例如，要新建一个名为py3scrapy的虚拟环境，命令如图1-20所示。

(8) 要查看已安装的虚拟环境，可以执行workon命令，如图1-21所示。

图1-20　使用virtualenvwrapper新建虚拟环境

图1-21　查看已安装的虚拟环境

1.3　Python开发环境的使用

“工欲善其事，必先利其器。”学习Python语言也一样，熟悉开发环境是学习一门编程语言的第一步。

1.3.1　使用自带的IDLE

IDLE是Python的官方标准开发环境，从官方网站下载并安装合适的Python版本后，也就同时安装了IDLE。相对于其他Python开发环境而言，IDLE虽比较简单，但具备Python应用开发的几乎所有功能，且不需要进行复杂配置，界面如图1-22所示。

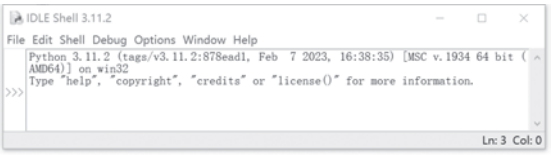

图1-22　Python 3.11.2 IDLE的界面

1.3.2　常用的第三方开发工具

除了默认安装的IDLE，还有大量的其他开发环境，如wingIDE、PyCharm、PythonWin、Eclipse、Spyder、IPython、Komodo等。严格来说，所有这些开发环境都是对Python解释器python.exe的封装，核心是完全一样的，只是加了外挂而已。这些开发环境使用起来方便，减少了出错率，尤其是拼写错误。以PyCharm为例，界面如图1-23所示。

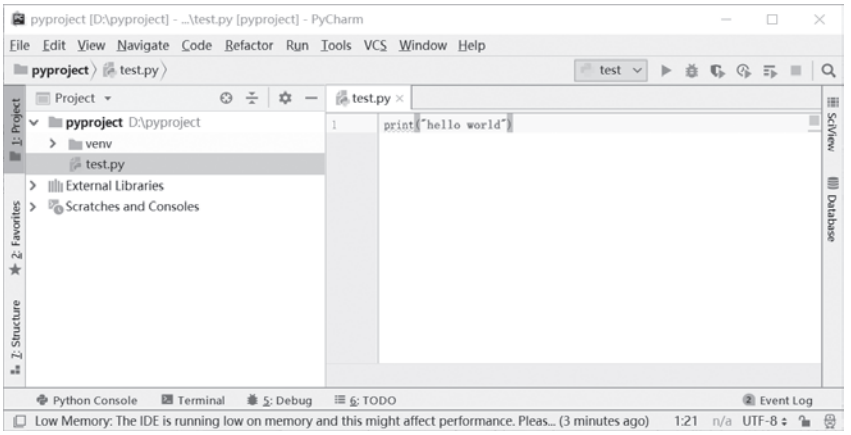

图1-23　PyCharm开发环境

1.3.3　官网交互式环境

如果暂时什么都不想安装，只是简单地想试试Python语言的功能，可以试试Python官方网站提供的Interactive Shell。登录Python官方网站后，单击图1-24所示方框内的右箭头图标，稍等片刻即可进入如图1-25所示的界面。

❖ 注意：

如果想尝试在安卓手机上编写Python程序，可以安装支持Python 3.x的QPython 3。

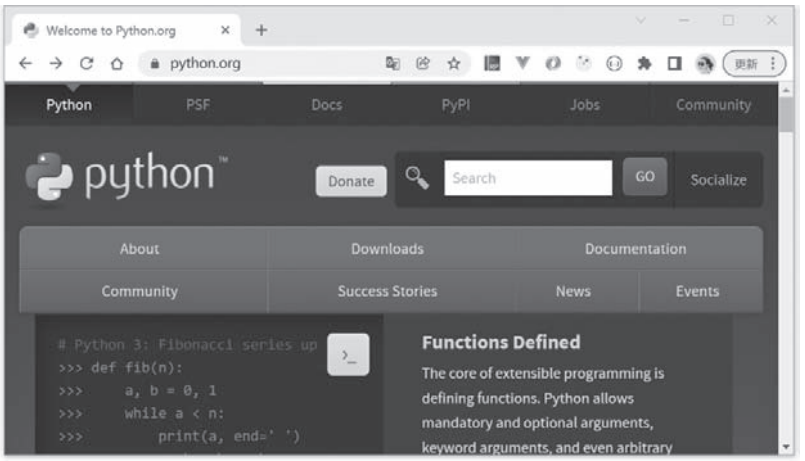

图1-24　Python官方网站

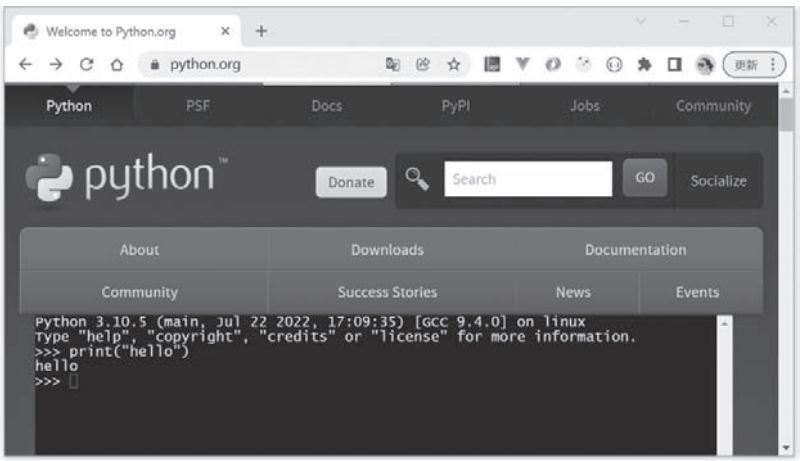

图1-25　Python官方网站提供的Interactive Shell界面

1.4　初学者常见的问题

1.4.1　为什么提示“python不是内部或外部命令……”

已经安装了Python，但是在DOS命令窗口中运行python命令时却提示“python不是内部或外部命令，也不是可运行的程序”。

原因：在环境变量中未给Path添加值。

解决办法：打开环境变量，为系统变量中的Path变量添加Python安装路径，假如Python的安装路径为C:\Python\Python37，就将这个路径添加到系统环境变量中(参照前面1.2.2节的操作方法)，然后再运行python命令。

1.4.2　如何在Python交互模式下运行.py文件

要运行已编写好的.py文件，可以单击【开始】菜单，在【搜索程序和文件】文本框中输入完整的文件名(包括路径)。例如，要运行D:\ceshi.py文件，可以使用下面的命令：

```
python311　D:\ceshi.py
```

在运行.py文件时，如果文件名或路径比较长，可以先在命令窗口中输入python加一个空格，然后直接把文件拖放到空格的位置，这时文件的完整路径将显示在空格的右侧，最后按下Enter键运行即可。

1.5　本章实战

1.5.1　IDLE的简单使用

本节采用标准的IDLE作为开发环境来演示Python的强大功能，几乎所有代码都可以直接在其他开发环境中运行，不需要做任何修改。有时候可能需要同时安装多个不同的版本，例如，同时安装Python 3.11.2和Python 3.7.9，可以根据不同的开发需求在两个版本之间切换。多版本并存并不会影响在IDLE环境中直接运行程序，只需要启动相应版本的IDLE。在命令提示符环境中运行Python程序时，可在调用Python主程序时指定其完整路径，或者修改系统环境变量Path来实现不同版本之间的切换(Path系统环境变量的修改方法参考前面的内容)。

如果能够熟练使用开发环境提供的一些快捷键，可以大幅提升开发效率。在IDLE环境中，除了撤销(Ctrl+Z)、全选(Ctrl+A)、复制(Ctrl+C)、粘贴(Ctrl+V)、剪切(Ctrl+X)等常规快捷键，其他比较常用的快捷键及功能说明如表1-1所示。

表1-1　IDLE中的常用快捷键及功能说明

快捷键	功能说明
Tab	补全单词，列出全部可选单词供选择
Alt+P	浏览历史命令(上一条)
Alt+N	浏览历史命令(下一条)
Ctrl+F6	重启Shell，之前定义的对象和导入的模块全部失效
F1	打开Python帮助文档
Alt+/	自动补全前面曾经出现过的单词，如果之前有多个单词具有相同的前缀，则在多个单词间循环切换
Ctrl+]	缩进代码块
Ctrl+[	取消代码块缩进
Alt+3	注释代码块
Alt+4	取消代码块注释

启动Python后默认处于交互模式，直接在Python提示符“>>>”的后面输入相应命令并按Enter键即可执行这些命令。如果执行顺利，可立即看到执行结果，否则会提示有关错误的信息并抛出异常。例如：

```
>>> 2+1                                  #井号为注释符，其后的内容不会被执行
3
>>> import math                          #导入Python标准库math
>>> math.sqrt(25)                        #使用标准库函数sqrt计算平方根
5.0
>>> 3**2                                 #使用**进行幂运算
9
>>> 3*(1+9)
30
>>> 2/0                                  #除0错误，会抛出异常
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    2/0
ZeroDivisionError: division by zero
>>> x='hello                             #语法错误，字符串的末尾缺少一个单引号
SyntaxError: incomplete input
```

从以上代码可以看出，交互模式一般用来实现一些简单的业务逻辑，或者验证某些功能。复杂的业务逻辑更多的是通过编写Python程序来实现，这样能方便代码的不断完善和重复利用。在IDLE界面中使用菜单命令File | New File创建一个程序文件，输入代码并保存为文件(务必保证扩展名为.py，如果是GUI程序，扩展名为.pyw)。然后，使用菜单命令Run | Run Module运行程序，程序运行的结果将直接显示在IDLE交互界面中。例如，假设程序文件test1.py的内容如下：

```
def main():
    print("this is a test program")
main()
```

在IDLE环境中运行该文件后，显示结果如下：

```
========== RESTART: D:/pyproject/test1.py ==========
this is a test program
```

在命令提示符环境中运行的方法和结果如图1-26所示。

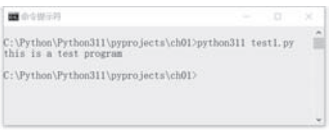

图1-26　在命令提示符环境中运行程序

❖ 技巧：

为提高代码的运行速度，以及对Python源代码进行保密，可以在命令提示符环境中使用python311 –OO –m py_compile file.py命令将Python程序file.py伪编译为.pyc文件，其中选项 –OO表示优化编译。

1.5.2　pip工具的使用

Python语言中有三类库：内置库、标准库和扩展库。其中，内置库和标准库在Python安装成功后即安装。内置库不需要使用import命令导入就能直接使用；标准库和扩展库需要先导入才能使用。扩展库主要通过pip工具来管理。

使用pip工具之前需要查看是否可用，打开命令提示符环境，输入 pip，如图1-27所示。如果pip工具不能使用，检查Python的安装目录，找到安装目录中的pip.exe文件，然后将其添加到系统环境变量Path中，之后重启再试。

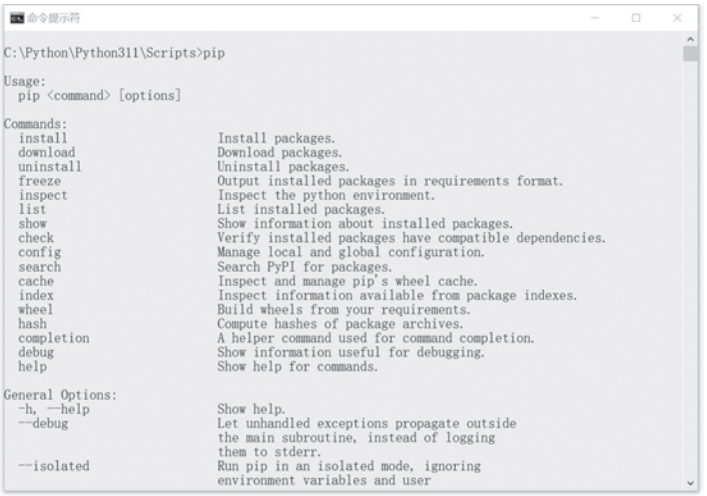

图1-27　pip工具

常用的pip命令如下。

- pip list：查看已安装的扩展库。
- pip install package_name：安装名为package_name的扩展库。
- pip uninstall package_name：卸载名为package_name的扩展库。

例如，使用pip工具安装pandas库，命令如图1-28所示。

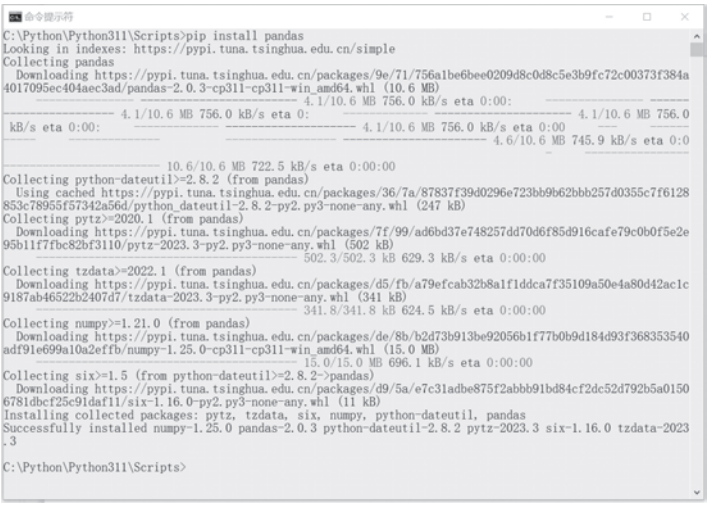

图1-28　安装pandas库

1.5.3 初始化环境

本书后面章节的示例讲解主要使用Python 3.11.2，为了使用方便，这里将Python 3.11.2安装目录下的python311.exe改回python.exe，pythonw311.exe改回pythonw.exe，如图1-29所示。

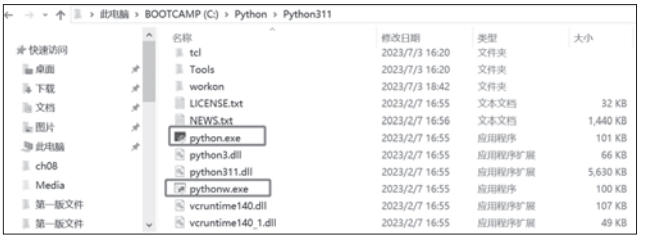

图1-29 修改文件名

打开命令提示符窗口，输入python命令，测试设置是否成功，如图1-30所示。然后重新执行pip命令：

```
python -m pip install --upgrade pip --force-reinstall
```

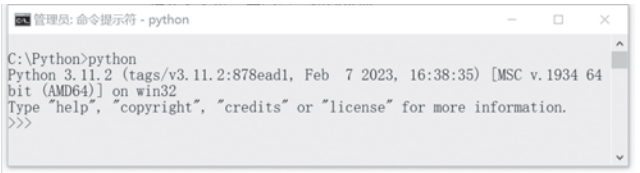

图1-30 测试设置

1.6 本章小结

本章作为开篇，首先介绍了Python语言的起源、Python的主流版本、Python语言的应用领域。Python是一门胶水语言，目前主流版本为Python 3.x。Python语言目前主要用于软件开发、科学计算、自动化运维、云计算、Web开发、网络爬虫、数据分析和人工智能等领域。

在使用Python语言之前，需要先搭建Python开发环境。到Python官方网站下载匹配操作系统的安装包，一键安装即可。可以在同一台计算机上安装多个Python版本或通过Python虚拟环境来使用多版本的Python。可以通过pip工具来安装、卸载、更新Python扩展库。可以在官网的交互式环境、IDLE工具或者专门的IDE环境(如PyCharm软件)中编写、调试Python程序。

1.7 思考与练习

1. Python语言的主流版本及应用主要有哪些？
2. 请实践在Windows操作系统上搭建Python开发环境。
3. 如何在Windows操作系统上安装多版本Python？
4. 如何安装Python虚拟环境？
5. 如何在IDLE中创建Python程序并执行？
6. 如何通过pip工具安装扩展库，以及如何在Python程序中使用扩展库？