高等院校计算机应用系列教材

Python 语言程序设计入门（第二版）（微课版）

焉德军　编著

清華大學出版社
北　京

Python语言程序设计入门(第二版)(微课版) 文前.indd 1　2023-06-26 10:14:18

内 容 简 介

本书以全国计算机等级考试二级Python语言程序设计考试大纲为指导，围绕Python语言的基础语法和数据结构组织内容。全书共分8章，包括Python概述、Python语言基础、Python程序的控制结构、函数、组合数据类型、文件和数据格式化，以及模块、包与库的使用、图形用户界面设计等内容。

本书还结合教学内容，合理地设计了一些课程思政案例，如社会主义核心价值观知识问答程序，为更好地开展课程思政提供了便利条件。

本书实例丰富，注重利用Python解决实际问题能力的培养，与《Python语言程序设计入门实验指导(第二版)》一起构成了一套完整的教学用书，可作为高等院校的教学参考书，也可作为全国计算机等级考试(NCRE)备考人员的参考资料。

本书配套的电子课件、实例源文件和习题答案可以到http://www.tupwk.com.cn/downpage 网站下载，也可以扫描前言中的二维码获取。扫描前言中的“看视频”二维码可以直接观看教学视频。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

Python语言程序设计入门：微课版/焉德军编著. —2版. —北京：清华大学出版社，2023.6
高等院校计算机应用系列教材
ISBN 978-7-302-63727-1

Ⅰ. ①P… Ⅱ. ①焉… 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 099677 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：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.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：**14.75　　**字　　数：**350 千字
版　　次：2021 年 8 月第 1 版　　2023 年 7 月第 2 版　　**印　　次：**2023 年 7 月第 1 次印刷
定　　价：79.00 元

产品编号：101906-01

前言

Python语言诞生于20世纪90年代，是一种跨平台、开源、面向对象、解释型、动态数据类型的高级计算机程序设计语言，在Web开发、科学计算、人工智能、大数据分析和系统运维等领域得到广泛应用，深受人们的青睐。不论你是计算机类专业的学生，还是非计算机类专业的学生，也不论你是否有一定的编程基础，只要你想学习Python语言，我们相信本书是一种比较合适的入门教材。

随着计算机基础教育形式的革新，2018年，大连民族大学计算机基础实验教学中心成立了Python语言课组，课组成员有焉德军、李宏岩、郑江超、隋励丽、杨为明、若曼、郑智强、王铎等多名老师。从课组成立开始，课组成员多次进行集体备课，进行Python语言程序设计集中学习，并多次参加各类Python语言程序设计相关的培训班：2019年4月，Python语言课组的五名教师，参加了在长沙举办的第三届全国高校Python语言与计算生态教学研讨会；2019年7月，Python语言课组全体成员参加了在南开大学举办的Python语言教学培训班；2019年8月，Python语言课组的两名教师，参加了全国高校大数据联盟举办的Python编程及大数据分析教师研修班；2020年1月，Python语言课组的两名教师，参加了北京雷课教育举办的Python人工智能及大数据分析研修班；2020年1月，Python语言课组全体成员参加了由东华大学举办的Python语言与大数据培训。经过一系列的培训和学习以及课组成员间的交流研讨，我们对于有关Python语言课程的教学内容、教学方法、教学手段等方面有了深刻了解，增强了在全校大范围开设Python语言程序设计课程的信心。2019年秋季学期，计算机基础实验教学中心停开了已经开设多年的VB程序设计课程和Access数据库课程，在全校5个学院21个专业开设了Python语言程序设计课程。

经过两年的学习和教学实践，Python语言课组积累了丰富的经验，着手编写适合高校学生学习的教材《Python语言程序设计入门》和实验教程《Python语言程序设计入门实验指导》。《Python语言程序设计入门(第二版)(微课版)》以全国计算机等级考试二级Python语言程序设计考试大纲为指导，围绕Python的基础语法和数据结构组织内容，包含Python概述、Python语言基础、Python程序的控制结构、函数、组合数据类型、文件和数据格式化，以及模块、包与库的使用。此外，还涉及图形用户界面设计等内容。《Python语言程序设计入门实验指导(第二版)》则包含3部分内容：与主教材内容相关的14个实验项目；Python语言的二级等级考试大纲和模拟题；网络爬虫、数据分析、数据可视化等项目实训。

为了更好地开展线上线下混合模式教学，结合教材，我们录制了44个微课视频，总时长630分钟，在中国大学MOOC的SPOC学校专有课程(大连民族大学)上线(http://www.icourse163.org/course/preview/DLNU-1461020176?tid=1461806466)。同时，基于“百科园通用考试平台”，我

们构建了Python语言程序设计题库，为实施过程化考核和形成性评价奠定了扎实基础。

为了更好地开展课程思政，结合教学内容，我们合理地设计了一些课程思政案例，如鸿蒙操作系统、社会主义核心价值观知识问答程序、习近平总书记在庆祝中华人民共和国成立70周年大会上的讲话词频分析、《中共中央关于坚持和完善中国特色社会主义制度、推进国家治理体系和治理能力现代化若干重大问题的决定》词云图等，所有这些课程思政案例，与教学内容紧密结合，不突兀，不牵强，因势利导、顺势而为地自然融入，起到润物无声、潜移默化的效果。在潜移默化中，让学生增长知识，坚定学生的理想信念，激发学生的爱国热情，培养学生具有民族自信心和维护国家利益的责任感，唤醒学生“为中华之崛起而读书”的原动力。

本套教材以程序设计初学者为对象，由浅入深、循序渐进地讲述Python语言的基本概念、基本语法和数据结构等基础知识，包括Python语言开发环境的安装、变量与数据类型、程序控制结构、函数和模块、文件、Python标准库和第三方库应用等。

通过本套教材，可以让程序设计初学者快速掌握程序设计的基本思想和一般方法，达到如下目标。

- 知识传授目标：使学生掌握Python语言的数据类型、基本控制结构、函数设计以及部分标准库和扩展库的使用，理解文件的基本处理方法，了解当下热门领域的Python扩展库的使用方法。
- 能力培养目标：培养学生分析问题、解决问题的能力，培养学生的计算思维和信息素养，使学生掌握程序设计方法，具备利用Python语言编程解决实际问题的能力。
- 价值塑造目标：将科技创新、爱国主义精神等思政元素融入教学，着眼于学生道德素养的熏陶濡染，培养学生一丝不苟、严谨认真、求真务实的工作作风和工匠精神，为学生学习后续课程、参加工作和开展科学研究打下良好基础。

在本套教材的编写过程中，我们参阅了很多有关Python语言方面的图书资料和网络资源，借鉴和吸收了其中的很多宝贵经验，在此向相关作者表示衷心的感谢。

《Python语言程序设计入门》一书2021年8月出版，荣获全国高等院校计算机基础教育研究会2021年学术成果推优活动优秀教材奖。

由于编者水平有限，书中难免有疏漏之处，敬请各位同行和读者批评指正，在此表示感谢。我们的邮箱是992116@qq.com，电话是010-62796045。

本书配套的电子课件、实例源文件和习题答案可以到http://www.tupwk.com.cn/downpage网站下载，也可以扫描下方的二维码获取。扫描下方的“看视频”二维码可以直接观看教学视频。

配套资源

扫描下载

看视频

扫一扫

编　者

2023年4月

Python语言程序设计入门(第二版)(微课版) 文前.indd 2　　2023-06-26 10:14:18

目录

第1章 Python概述 …… 1

1.1 计算机系统简介 …… 1
1.1.1 计算机系统的组成 …… 1
1.1.2 计算机硬件系统 …… 2
1.1.3 计算机软件系统 …… 4
1.2 数制与编码 …… 6
1.2.1 数制的基本概念 …… 6
1.2.2 常用的数制 …… 7
1.2.3 数制间的转换 …… 8
1.2.4 数据在计算机中的表示方式 …… 10
1.2.5 字符编码 …… 11
1.3 Python语言简介 …… 13
1.3.1 Python语言发展简史 …… 13
1.3.2 Python语言的特点 …… 14
1.3.3 Python语言的应用领域 …… 15
1.4 Python语言开发环境 …… 15
1.4.1 下载和安装Python …… 15
1.4.2 内置的IDLE开发环境 …… 19
1.4.3 Python常用的其他一些集成开发环境 …… 21
1.5 初识Python程序 …… 21
1.5.1 把Python解释器用作计算器 …… 21
1.5.2 Python程序示例 …… 23
1.5.3 Python程序编码规范 …… 25
1.5.4 Python的帮助文档 …… 26
1.6 习题 …… 29

第2章 Python语言基础 …… 30

2.1 标识符与关键字 …… 30
2.1.1 标识符 …… 30
2.1.2 关键字 …… 30
2.1.3 Python内置的标准函数 …… 31
2.2 变量与常量 …… 31
2.2.1 变量 …… 31
2.2.2 常量 …… 33
2.3 数据类型 …… 33
2.3.1 数字类型 …… 33
2.3.2 字符串类型 …… 37
2.3.3 列表、元组、字典和集合简介 …… 45
2.4 类型判断和类型间转换 …… 52
2.4.1 类型判断 …… 52
2.4.2 类型间转换 …… 52
2.5 基本输入输出函数 …… 53
2.5.1 input()函数 …… 53
2.5.2 print()函数 …… 54
2.6 运算符 …… 55
2.6.1 算术运算符 …… 55
2.6.2 比较运算符 …… 55
2.6.3 逻辑运算符 …… 56
2.6.4 赋值运算符 …… 56
2.6.5 成员运算符 …… 57
2.6.6 身份运算符 …… 57
2.6.7 位运算符 …… 58
2.6.8 运算符的优先级和结合性 …… 59
2.7 应用问题选讲 …… 60
2.8 习题 …… 63

第3章 Python程序的控制结构 …… 65

3.1 顺序结构 …… 65
3.2 分支结构 …… 66
3.2.1 单分支结构：if语句 …… 66
3.2.2 双分支结构：if-else语句 …… 67

3.2.3 多分支结构：if-elif-else语句 …… 68
3.2.4 分支嵌套 …… 70
3.3 循环结构 …… 71
3.3.1 条件循环：while语句 …… 71
3.3.2 遍历循环：for语句 …… 73
3.3.3 循环的嵌套 …… 75
3.4 break、continue和pass语句 …… 75
3.4.1 break语句 …… 76
3.4.2 continue语句 …… 76
3.4.3 pass语句 …… 77
3.5 循环结构中的else语句 …… 77
3.6 程序的异常处理：try-except …… 78
3.7 应用问题选讲 …… 80
3.8 习题 …… 88

第4章 函数 …… 91
4.1 函数的定义与调用 …… 91
4.1.1 定义函数 …… 91
4.1.2 调用函数 …… 92
4.2 函数的参数与返回值 …… 93
4.2.1 参数传递 …… 93
4.2.2 位置参数 …… 95
4.2.3 关键字参数 …… 96
4.2.4 带默认值的参数 …… 96
4.2.5 可变长参数 …… 97
4.2.6 函数的返回值 …… 98
4.3 匿名函数 …… 99
4.4 函数的嵌套调用与递归调用 …… 100
4.4.1 函数的嵌套调用 …… 100
4.4.2 函数的递归调用 …… 102
4.5 变量的作用域 …… 104
4.5.1 局部变量 …… 104
4.5.2 全局变量 …… 105
4.5.3 global语句 …… 106
4.6 应用问题选讲 …… 107
4.7 习题 …… 110

第5章 组合数据类型 …… 112
5.1 列表 …… 112
5.1.1 列表及其操作方法 …… 112
5.1.2 遍历列表 …… 114
5.1.3 复制列表 …… 116
5.1.4 列表推导式 …… 118
5.1.5 二维列表 …… 119
5.2 元组 …… 121
5.3 字典 …… 121
5.3.1 字典及其操作方法 …… 121
5.3.2 遍历字典 …… 123
5.4 集合 …… 125
5.4.1 集合及其操作方法 …… 125
5.4.2 遍历集合 …… 126
5.4.3 集合中的运算 …… 127
5.5 应用问题选讲 …… 128
5.6 习题 …… 134

第6章 文件和数据格式化 …… 137
6.1 文件概述 …… 137
6.2 文件的基本操作 …… 138
6.2.1 文件的打开与关闭 …… 138
6.2.2 文件的读/写 …… 140
6.2.3 文件的定位读/写 …… 142
6.3 采用CSV格式读/写文件 …… 144
6.3.1 CSV文件概述 …… 144
6.3.2 读/写CSV文件 …… 144
6.4 读/写JSON文件 …… 147
6.5 应用问题选讲 …… 148
6.6 习题 …… 150

第7章 模块、包与库 …… 152
7.1 模块 …… 152
7.1.1 模块的概念 …… 152
7.1.2 模块的导入与使用 …… 153
7.1.3 模块搜索路径 …… 157
7.2 Python中的包 …… 159
7.3 Python中的标准库 …… 159
7.3.1 math库 …… 160
7.3.2 random库 …… 161
7.3.3 time库 …… 162
7.3.4 turtle库 …… 164
7.4 Python中的第三方库 …… 169

Python语言程序设计入门(第二版)(微课版) 文前.indd 4 2023-06-26 10:14:18

7.4.1　第三方库简介 ……169
7.4.2　下载与安装第三方库 ……170
7.4.3　使用PyInstaller打包文件 ……171
7.4.4　jieba库 ……173
7.4.5　wordcloud库 ……175
7.5　应用问题选讲 ……178
7.6　习题 ……181

第8章　图形用户界面设计 ……184
8.1　图形用户界面概述 ……184
8.1.1　图形用户界面概念的引入 ……184
8.1.2　图形用户界面的常用设计模块 ……185
8.1.3　tkinter模块 ……186
8.1.4　tkinter组件常用的标准属性 ……188
8.1.5　tkinter组件的几何布局管理器 ……190
8.2　tkinter的常用组件 ……192
8.2.1　标签组件Label ……192
8.2.2　按钮组件Button ……192
8.2.3　文本框组件Entry ……193
8.2.4　列表框组件Listbox ……195
8.2.5　单选按钮组件Radiobutton ……197
8.2.6　复选框组件Checkbutton ……198
8.2.7　菜单组件Menu ……199
8.2.8　子窗体组件Toplevel ……201
8.2.9　其他一些常用组件 ……202
8.3　tkinter的事件处理 ……203
8.3.1　事件类型 ……203
8.3.2　使用command参数处理事件 ……204
8.3.3　使用bind()方法处理事件 ……204
8.4　应用问题选讲 ……204
8.5　习题 ……207

参考文献 ……208

附录A　字符与ASCII码对照表 ……209

附录B　习题参考解答 ……210

附录C　微课视频目录 ……226

Python语言程序设计入门(第二版)(微课版) 文前.indd 5　2023-06-26 10:14:18

第 1 章

Python 概述

Python诞生于20世纪90年代，是一种跨平台、开源、面向对象、解释型、动态数据类型的高级计算机程序设计语言，在Web开发、科学计算、人工智能、大数据分析和系统运维等领域得到广泛应用，深受人们的青睐。本章将首先介绍计算机系统，让读者初步认识计算机程序和计算机中的信息表示，了解Python程序的开发环境，并理解Python程序的执行过程。

1.1 计算机系统简介

1.1.1 计算机系统的组成

完整的计算机系统包括硬件系统和软件系统两部分，如图1-1所示。组成一台计算机的物理设备的总称就是计算机硬件系统，这是实实在在的物体，是计算机工作的基础。指挥计算机工作的各种程序的集合称为计算机软件系统，这是计算机的灵魂，是控制和操纵计算机工作的核心。计算机通过执行程序而运行，计算机在工作时需要软硬件协同工作，二者缺一不可。

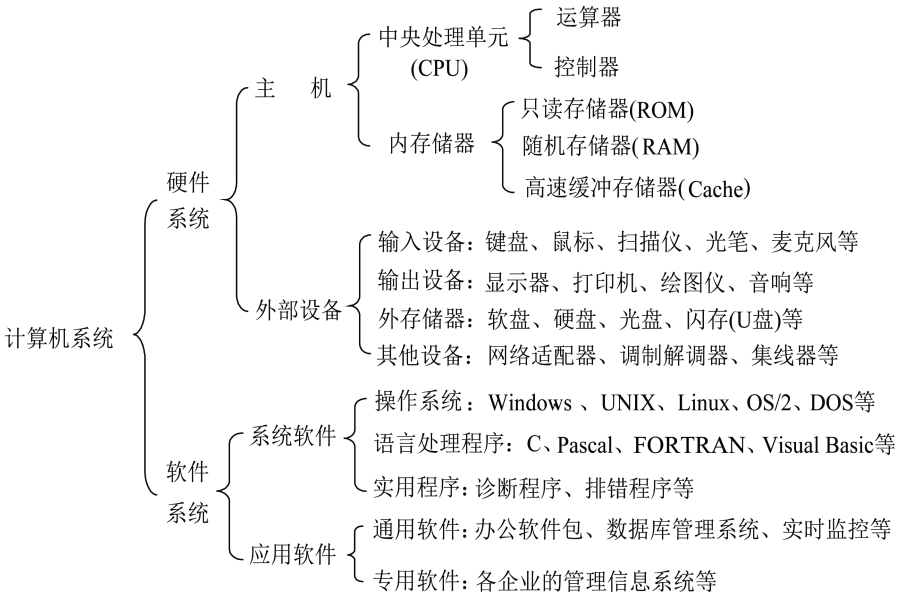


图1-1　计算机系统的组成结构

1.1.2　计算机硬件系统

计算机硬件(Computer Hardware)又称硬件平台，是指计算机系统所包含的各种机械的、电子的、磁性的装置和设备，如运算器、磁盘、键盘、显示器和打印机等。每个功能部件各尽其职、协调工作，缺少其中任何一个就不能成为完整的计算机系统。

计算机能够处理存储的数据。可以说，存储和处理是一个整体：存储是为了处理，处理需要存储。“存储和处理的整体性”的最初表达是美国普林斯顿大学的冯•诺依曼于1945年提出的计算机体系结构思想，一般称为“程序存储思想”。计算机从1946年问世至今都是以这种思想为基本依据的，主要特点可归结为以下3点。

(1) 计算机由5部分组成：运算器、控制器、存储器、输入设备和输出设备。

(2) 程序和数据存放在存储器中并按地址寻访。

(3) 程序和数据用二进制表示，与十进制相比，实现二进制运算的结构简单，容易控制。

如今，半个多世纪过去了，计算机的系统结构发生了很大的变化，但就结构原理来说，仍然是冯•诺依曼型计算机，如图1-2所示，图中的实线为数据流，虚线为控制流。

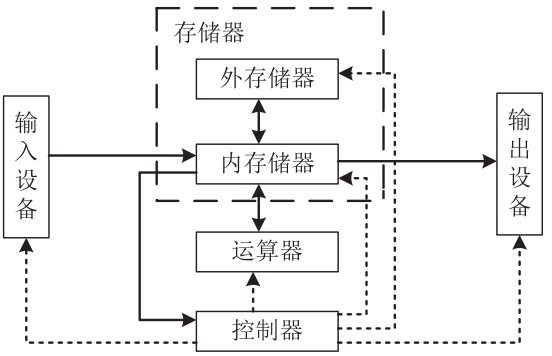


图1-2　冯•诺依曼型计算机的结构

Python语言程序设计入门(第二版)(微课版) 正文.indd 2　2023-06-26 10:24:44

硬件是计算机工作的物质基础，计算机的性能，如运算速度、存储容量、计算精度、可靠性等在很大程度上取决于硬件的配置。下面简单介绍计算机的5个基本组成部分。

1. 运算器

运算器的主要功能是进行算术运算和逻辑运算。计算机最主要的工作就是执行运算，大量的数据运算任务是在运算器中进行的。

运算器又称算术逻辑部件(Arithmetic and Logic Unit，ALU)。

在计算机中，算术运算是指加、减、乘、除等基本运算。逻辑运算是指逻辑判断、关系比较以及其他的基本逻辑运算，如与、或、非等。但不管是算术运算还是逻辑运算，都只是基本运算。也就是说，运算器只能做这些最简单的运算，复杂的运算都要通过基本运算一步步实现。然而，运算器的运算速度却快得惊人，因而计算机具有高速的信息处理功能。

运算器中的数据取自内存，运算的结果则送回内存。运算器对内存的读/写操作是在控制器的控制之下进行的。

2. 控制器

控制器是计算机的神经中枢和指挥中心，只有在它的控制下整个计算机才能有条不紊地工作，自动执行程序。控制器的功能是依次从存储器取出指令、翻译指令、分析指令，向其他部件发出控制信号，指挥计算机各部件协同工作。

运算器和控制器合称中央处理单元(Central Processing Unit，CPU)。

3. 存储器

存储器的主要功能是存放程序和数据。使用时，可以从存储器中取出信息，不破坏原有的内容，这种操作称为存储器的读操作；也可以把信息写入存储器，原来的内容会被抹掉，这种操作称为存储器的写操作。

存储器分为程序存储区、数据存储区和栈。程序存储区存放程序中的指令，数据存储区存放数据。CPU通过地址总线发出相应的地址，找到存储器中与该地址对应的存储单元，然后通过数据总线操作存储单元中的数据。

存储器通常分为内存储器和外存储器。

1) 内存储器

内存储器简称内存(又称主存)，是计算机中信息交流的中心。用户通过输入设备输入的程序和数据最初被送入内存，控制器执行的指令和运算器处理的数据取自内存，运算的中间结果和最终结果保存在内存中，输出设备输出的信息来自内存。内存中的信息若要长期保存，则应送到外存储器中。总之，内存要与计算机的各个部件打交道，进行数据交换。因此，内存的存取速度直接影响计算机的运算速度。

2) 外存储器

外存储器设置在主机外部，简称外存(又称辅存)，主要用来长期存放暂时不用的程序和数据。通常，外存不和计算机的其他部件直接交换数据，只和内存交换数据，而且不是按单个数据进行存取，而是成批地进行数据交换。常用的外存有磁盘、磁带和光盘等。由于

Python语言程序设计入门(第二版)(微课版) 正文.indd 3　2023-06-26 10:24:44

外存储器安装在主机外部，因此也可以归为外部设备。

4. 输入设备

输入设备用来接收用户输入的原始数据和程序，并将它们转换为计算机可以识别的形式(二进制代码)，而后存放到内存中。常用的输入设备有键盘、鼠标、扫描仪、光笔、数字化仪和麦克风等。

5. 输出设备

输出设备用于将存放在内存中的由计算机处理的结果转换为人们所能接受的形式。常用的输出设备有显示器、打印机、绘图仪和音响等。

1.1.3 计算机软件系统

计算机软件(Computer Software)是相对于计算机硬件而言的，包括计算机运行所需的各种程序、数据以及有关的技术文档资料。只有硬件而没有任何软件支持的计算机称为裸机。在裸机上只能运行机器语言程序，使用很不方便，效率也低。硬件是软件赖以运行的物质基础，软件是计算机的灵魂，是发挥计算机功能的关键。

计算机软件通常可分为系统软件和应用软件两大类。用户与计算机系统各层次之间的关系如图1-3所示。

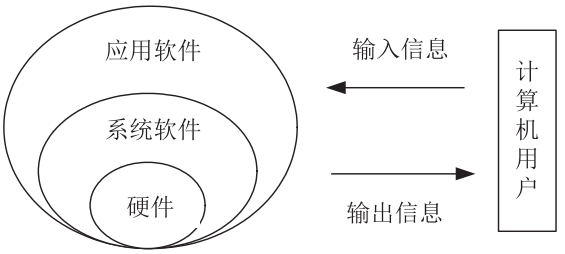


图1-3　用户与计算机系统各层次之间的关系

1. 系统软件

系统软件是管理、监控和维护计算机资源的软件，用来扩展计算机的功能，提高计算机的工作效率，方便用户使用计算机的软件。系统软件包括操作系统、程序设计语言、语言处理程序、数据库管理程序、系统辅助处理程序等。

1) 操作系统

在计算机软件中，最重要且最基本的就是操作系统(Operating System，OS)。操作系统是最底层的软件，控制所有在计算机上运行的程序并管理整个计算机的资源，在裸机与应用程序及用户之间架起了一座沟通的桥梁。没有操作系统，用户就无法自如地应用各种软件或程序。

目前常见的操作系统有Windows、UNIX、Linux、Android、macOS、iOS、Blackberry OS和HarmonyOS(鸿蒙)等。

Python语言程序设计入门(第二版)(微课版) 正文.indd 4　　2023-06-26 10:24:44

2) 程序设计语言

程序设计的基本方法称为IPO：用户编写程序，输入(Input)计算机；然后由计算机将其翻译成机器语言并在计算机上运行(Process)；最后输出(Output)结果。计算机语言大致分为机器语言、汇编语言和高级语言。

机器语言：机器语言是以二进制代码表示的指令集合，是计算机能直接识别和执行的计算机语言。优点是执行效率高、速度快；但直观性差，可读性不强，因而给计算机的推广和使用带来了极大的困难。

汇编语言：汇编语言是符号化的机器语言，这种语言使用助记符来表示指令中的操作码和操作数的指令系统。汇编语言相比机器语言前进了一步，助记符比较容易记忆，可读性也好，但编写程序的效率不高、难度较大、维护较困难，属于低级语言。

高级语言：高级语言是接近人类自然语言和数学语言的计算机语言，是第三代计算机语言。高级语言的特点就是与计算机的指令系统无关，因而从根本上摆脱了对机器的依赖，使之独立于机器，面向过程，进而面向用户。由于易学易记，便于书写和维护，高级语言提高了程序设计的效率和可靠性。目前广泛使用的高级语言有C、C++、Java、JavaScript、PHP、HTML、Perl、Go、LISP和Python等。

3) 语言处理程序

将计算机不能直接执行的使用非机器语言编写的程序翻译成能直接执行的机器语言的翻译程序称为语言处理程序。

使用各种程序设计语言编写的程序称为源程序，计算机不能直接识别和执行源程序。把计算机本身不能直接识别的源程序翻译成机器能够识别的机器指令代码后，计算机才能执行，这种翻译后的程序称为目标程序。

计算机将源程序翻译成机器指令的方式有两种：编译方式和解释方式。编译方式与解释方式的工作过程如图1-4所示。

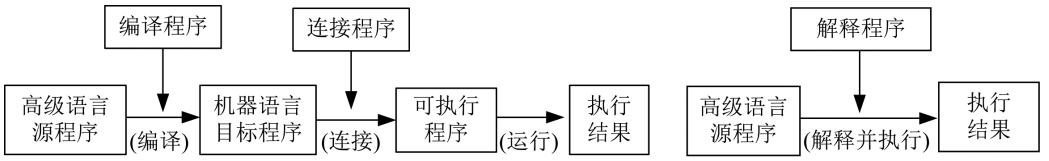


图1-4　将源程序翻译成机器指令的过程

由图1-4可以看出，编译方式会使用相应的编译程序把源程序翻译成机器语言的目标程序，然后连接成可执行程序，运行可执行程序后得到结果。目标程序和可执行程序都以文件的形式存放在磁盘上，再次运行程序时，只需要直接运行可执行程序，而不必重新编译和连接。

解释方式则把源程序输入计算机，然后使用相应语言的解释程序对代码进行逐条解释、逐条执行，执行后只能得到结果，而不能保存解释后的机器代码，下次运行程序时需要重新解释并执行。

4) 数据库管理系统

数据库管理系统是能够对数据进行加工、管理的系统软件。常见的数据库管理系统有

Python语言程序设计入门(第二版)(微课版) 正文.indd 5　2023-06-26 10:24:44

MySQL、DB2、Oracle、Access和SQL Server等。

5) 系统辅助处理程序

系统辅助处理程序又称“软件研制开发工具”“支持软件”或“工具软件”，主要有编辑程序、调试程序、装配和连接程序、测试程序等。

2. 应用软件

应用软件是用户利用计算机及系统软件，为解决实际问题而开发的软件的总称。应用软件一般分为两大类：通用软件和专用软件。

通用软件支持最基本的应用，如文字处理软件(Word)、表格处理软件(Excel)等。

专用软件是专门为某一专业领域开发的软件，如财务管理系统、计算机辅助设计(CAD)软件和仅限于某个部门使用的应用数据库管理系统等。

1.2 数制与编码

在计算机系统中，数字和符号都是使用电子元器件的不同状态来表示的。根据计算机的这一特点，我们提出了这样的问题：数值在计算机中是如何表示和运算的？这就是本节将要讨论的“数制”问题。

1.2.1 数制的基本概念

通过一组固定的数字(数码符号)和一套统一的规则来表示数值的方法称为数制。数制的种类很多，除了我们熟悉的十进制，还有二十四进制(24小时为1天)、六十进制(60秒为1分钟、60分钟为1小时)、二进制(手套、筷子等两只/支为一双)等。

计算机系统采用二进制的主要原因是电路设计简单、运算简单、工作可靠且逻辑性强。

不论是哪一种数制，它们的计数和运算都有一些共同的规律和特点。

1. 逢*R*进一

*R*是数制中所需数字字符的总数，称为基数(Radix)。例如，十进制使用0、1、2、3、4、5、6、7、8、9这10个不同的数制符号来表示数值。在十进制中，基数是10，表示逢十进一。

2. 位权表示法

位权(简称权)是指一个数字在某个位置上所代表的值，处在不同位置的数字所代表的值不同，每个数字的位置决定了这个数字的值或位权。例如，在十进制数586中，5的位权是100(即10^2)。

位权与基数的关系是：位权的值是基数的若干次幂。因此，使用任何一种数制表示的数都可以写成按位权展开的多项式之和。例如，十进制数256.07可以用如下形式表示：

Python语言程序设计入门(第二版)(微课版) 正文.indd 6 2023-06-26 10:24:44

$$(256.07)_{10}=2\times10^{2}+5\times10^{1}+6\times10^{0}+0\times10^{-1}+7\times10^{-2}$$

位权表示法的原则是数字的总数等于基数，每个数字都要乘以基数的幂次，幂次是由每个数字所在的位置决定的。排列方式是以小数点为界，整数自右向左依次为0次方、1次方、2次方，以此类推；小数自左向右依次为负1次方、负2次方，以此类推。

1.2.2　常用的数制

在使用计算机解决实际问题的过程中，我们往往使用的是十进制数，而计算机内部使用的是二进制数。另外，在计算机应用中经常需要使用十六进制数或八进制数，二进制数因其与十六进制数和八进制数正好有倍数关系，如2^3等于8、2^4等于16，所以十分便于在计算机应用中进行表示。

1. 十进制数(Decimal)

按“逢十进一”的原则进行计数，称为十进制数，每一位计满10时向高位进1。对于任意的十进制数，都可以使用小数点分成整数部分和小数部分。

十进制数的特点是：数字的个数等于基数10，逢十进一，借一当十；最大数字是9，最小数字是0，有10个数制符号——0、1、2、3、4、5、6、7、8、9；在具体表示时，每个数字都要乘以基数10的幂次。

2. 二进制数(Binary)

按“逢二进一”的原则进行计数，称为二进制数，每一位计满2时向高位进1。

二进制数的特点是：数字的个数等于基数2；最大数字是1，最小数字是0，只有两个数制符号——0和1；在数值的表示中，每个数字都要乘以2的幂次，这就是每一位的位权。第一位的位权是2^0，第二位的位权是2^1，第三位的位权是2^2，以此类推。

例如，二进制数1101.11可表示为：

$$(1101.11)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}=(13.75)_{10}$$

二进制的算术运算与十进制类似，操作简单、直观，更容易实现。

3. 八进制数(Octal)

八进制数的进位规则是“逢八进一”，基数R=8，采用的数制符号是0、1、2、3、4、5、6、7，每一位的位权是8的幂次。例如，八进制数376.4可表示为：

$$\begin{aligned}(376.4)_8&=3\times8^2+7\times8^1+6\times8^0+4\times8^{-1}\\&=3\times64+7\times8+6+0.5=(254.5)_{10}\end{aligned}$$

4. 十六进制数(Hexadecimal)

十六进制数的进位规则是“逢十六进一”，基数R=16，采用的数制符号为0、1~9、A、B、C、D、E、F，其中A～F分别代表十进制中的10～15，每一位的位权是16的幂次。例如，十六进制数3AB.11可表示为：

Python语言程序设计入门(第二版)(微课版) 正文.indd 7　2023-06-26 10:24:44

$$(3AB.11)_{16}=3\times16^{2}+10\times16^{1}+11\times16^{0}+1\times16^{-1}+1\times16^{-2}\approx(939.0664)_{10}$$

5. 常用数制的对应关系

常用数制的对应关系如表1-1所示。

表1-1　常用数制的对应关系

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

1.2.3　数制间的转换

将数由一种数制转换成另一种数制称为数制间的转换。由于计算机采用二进制，而日常生活中人们习惯使用十进制，因此计算机在进行数据处理时就必须把输入的十进制数转换成计算机所能接收的二进制数，计算机运行结束后，再把二进制数转换成人们习惯的十进制数并输出。这两个转换过程完全由计算机系统自动完成。

1. 二进制数与十进制数之间的转换

1) 将二进制数转换成十进制数

将二进制数转换成十进制数的方法前面已经讲过，只需要将二进制数按位权展开，然后将各项数值按十进制数相加，便可得到等值的十进制数。例如：

$$(10110.11)_{2}=1\times2^{4}+1\times2^{2}+1\times2^{1}+1\times2^{-1}+1\times2^{-2}=(22.75)_{10}$$

同理，为了将任意进制数转换为十进制数，只需要将数$(N)_R$写成按位权展开的多项式，然后按十进制规则进行运算，便可求得相应的十进制数$(N)_{10}$。

Python语言程序设计入门(第二版)(微课版) 正文.indd 8　2023-06-26 10:24:44

2) 将十进制数转换成二进制数

在将十进制数转换成二进制数时，需要对整数部分和小数部分分别进行转换。

(1) 整数转换。整数转换采用除2取余法。例如，将$(57)_{10}$转换为二进制数，用除2取余法得：

$$
\begin{array}{r|r l}
2 & 57 & \text{余数} \\
2 & 28 & \cdots\cdots\cdots 1=a_0 \\
2 & 14 & \cdots\cdots\cdots 0=a_1 \\
2 & 7 & \cdots\cdots\cdots 0=a_2 \\
2 & 3 & \cdots\cdots\cdots 1=a_3 \\
2 & 1 & \cdots\cdots\cdots 1=a_4 \\
 & 0 & \cdots\cdots\cdots 1=a_5
\end{array}
$$

结果：$(57)_{10}=(111001)_2$

(2) 小数转换。小数转换采用乘2取整法。例如，将$(0.834)_{10}$转换成二进制数，用乘2取整法得：

$$
\begin{array}{rl}
0.834 & \\
\times\quad 2 & \text{整数} \\
\hline
1.668 & \cdots\cdots\cdots 1=a_{-1} \\
0.668 & \\
\times\quad 2 & \\
\hline
1.336 & \cdots\cdots\cdots 1=a_{-2} \\
0.336 & \\
\times\quad 2 & \\
\hline
0.672 & \cdots\cdots\cdots 0=a_{-3} \\
\times\quad 2 & \\
\hline
1.344 & \cdots\cdots\cdots 1=a_{-4}
\end{array}
$$

结果：$(0.834)_{10}\approx(0.1101)_2$

在对小数部分乘2取整的过程中，不一定能使最后的乘积为0，因此存在转换误差。通常，当二进制数的精度已经达到预定的要求时，运算便可结束。

在将带有整数和小数的十进制数转换成二进制数时，必须对整数部分和小数部分分别按除2取余法和乘2取整法进行转换，之后再将两者的转换结果合并起来即可。

同理，要将十进制数转换成任意的R进制数$(N)_R$，整数转换可采用除R取余法，小数转换可采用乘R取整法。

2. 二进制数与八进制数、十六进制数之间的转换

八进制数和十六进制数的基数分别为$8=2^3$、$16=2^4$，所以三位的二进制数恰好相当于一位的八进制数，四位的二进制数则相当于一位的十六进制数，它们之间的相互转换是很方便的。

将二进制数转换成八进制数的方法是：从小数点开始，分别向左、向右将二进制数按每三位一组的形式进行分组(不足三位的补0)，然后写出与每一组二进制数等值的八进

Python语言程序设计入门(第二版)(微课版) 正文.indd 9　　2023-06-26 10:24:45

制数。

例如，将二进制数100110110111.00101转换成八进制数，结果是：

$$(100110110111.00101)_2 = (4667.12)_8$$

将八进制数转换成二进制数的方法恰好与将二进制数转换成八进制数相反：从小数点开始，分别向左、向右将八进制数的每一位数字转换成三位的二进制数即可。

将二进制数转换成十六进制数的方法和二进制数与八进制数之间的转换方法相似，从小数点开始，分别向左、向右将二进制数按每四位一组的形式进行分组(不足四位的补0)，然后写出与每一组二进制数等值的十六进制数。

例如，将二进制数1111000001011101.0111101转换成十六进制数，结果是：

$$(1111000001011101.0111101)_2=(F05D.7A)_{16}$$

类似地，在将十六进制数转换成二进制数时，可按相反的过程进行操作。

1.2.4 数据在计算机中的表示方式

计算机中处理的数据可分为数值型数据和非数值型数据两类。数值型数据是指数学中的代数值，具有量的含义，如235、−328.45或3/8等；非数值型数据是指输入计算机中的所有文字信息，没有量的含义，如数字0～9、大写字母A～Z或小写字母a～z、汉字、图形、声音及一切可印刷的符号(如+、−、！、#、%)等。

由于计算机采用二进制，因此这些数据在计算机内部必须以二进制编码的形式表示。也就是说，一切输入计算机中的数据都是由0和1 两个数字进行组合的。数值型数据有正负之分，数学中使用符号+和−来分别表示正数和负数，但在计算机中，数的正负符号也要使用0和1来表示。

1. 有符号数的表示方法

在计算机中，有符号的数通常使用原码、反码和补码3种形式来表示，这么做的主要目的就是解决减法运算的问题。任何正数的原码、反码和补码形式都完全相同，而负数的原码、反码和补码形式则不同。

1) 数的原码表示

正数的符号位用0表示，负数的符号位用1表示，有效值部分用二进制绝对值表示，这种表示法称为原码。原码对0的表示方法不唯一，分为正的0(000…00)和负的0(100…00)。

2) 数的反码表示

正数的反码和原码相同，负数的反码则是对原码的除符号位外的其他各位取反：0变1，1变0。

例如：$(+76)_{原}=(+76)_{反}$ =01001100

$(-76)_{原}$ = 11001100　$(-76)_{反}$ = 10110011

可以验证，任意数的反码的反码即为原码本身。

3) 数的补码表示

正数的补码和原码相同，负数的补码则需要对反码加1。

Python语言程序设计入门(第二版)(微课版) 正文.indd 10　2023-06-26 10:24:45

例如：$(+76)_{原}=(+76)_{反}=(+76)_{补}=01001100$

$(-76)_{原}=11001100$　$(-76)_{反}=10110011$　$(-76)_{补}=10110100$

可以验证，任意数的补码的补码即为原码本身。

2. 定点数与浮点数

数值除了有正负之分，还有带小数点的数值。当所要处理的数值含有小数部分时，计算机就必须解决数值中的小数点的表示问题。在计算机中，通常采用隐含规定小数点的位置这种形式来表示带小数点的数值。

根据小数点的位置是否固定，数的表示方法可以分为定点整数、定点小数和浮点数3种类型。定点整数和定点小数统称为定点数。

1) 定点整数

定点整数是指小数点隐含固定在整个数值的最后，符号位右边的所有位数表示的是整数。如果用8位表示定点整数，那么00000110表示二进制数+0000110，也就是十进制数+6。

2) 定点小数

定点小数是指小数点隐含固定在某个位置的小数。人们通常将小数点固定在最高数据位的左边。如果用8位表示定点小数，那么01100000表示二进制数+0.1100000，也就是十进制数+0.75。

由此可见，定点数可以表示纯小数和整数。定点整数和定点小数在计算机中的表示并没有什么区别，小数点完全靠事先约定而隐含在不同位置。

3) 浮点数

浮点数是指小数点位置不固定的数，浮点数既有整数部分又有小数部分。在计算机中，浮点数通常由阶码(也称为指数)和尾数两部分进行表示。其中阶码用二进制定点整数表示，尾数用二进制定点小数表示，阶码的长度决定数的范围，尾数的长度决定数的精度。为保证不损失有效数字，通常还会对尾数进行规格化处理，从而保证尾数的最高位为1。实际数值可通过阶码进行调整。

浮点数的格式多种多样。例如，某计算机用32位表示浮点数，阶码部分为8位补码的定点整数，尾数部分为24位补码的定点小数。浮点数的最大特点在于比定点数表示的数值范围大。

例如，二进制的+110110等于$2^6\times0.110110$，阶码为6，也就是二进制的+110，尾数为+0.110110。浮点数的表示形式如图1-5所示。

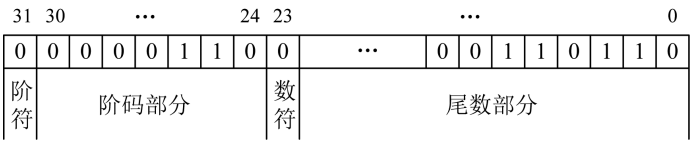


图1-5　浮点数示例

1.2.5　字符编码

字符编码(Character Encoding)是指把字符集中的字符编码为指定集合中的某一对象(如

Python语言程序设计入门(第二版)(微课版) 正文.indd 11　2023-06-26 10:24:45

自然数序列等)，以便文本在计算机中存储和通过通信网络进行传递，常见的例子包括将拉丁字母表编码成摩斯电码和ASCII(American Standard Code for Information Interchange，美国信息交换标准码)。

1. 西文编码

目前使用最广泛的西文字符集及其编码是ASCII字符集和ASCII码，ASCII码同时也被国际标准化组织(International Organization for Standardization，ISO)批准为国际标准。ASCII码将字母、数字和其他符号使用0~127的整数进行编号，并用7个二进制位(bit，比特)表示这种整数。另外，我们通常额外使用一个扩充的二进制位，以便以字节(Byte)的方式进行存储，最高位通常用于奇偶校验。

基本的ASCII字符集共有128个字符，其中有96个可打印字符，包括常用的字母、数字、标点符号等。另外，还有32个控制字符，如附录A所示。

字母和数字的ASCII码值的记忆非常简单，只要记住一个字母或数字的ASCII码值(例如， 0的ASCII码值为48，A的ASCII码值为65，a的ASCII码值为97)，就可以推算出其余大小写字母、数字的ASCII码值。

2. 中文编码

为了扩充ASCII编码，以显示本国的语言，不同的国家和地区制定了不同的标准，由此产生了GB2312、BIG5、JIS等编码标准。这些编码标准使用2字节来代表一个字符的各种汉字延伸编码方式，称为ANSI编码，又称MBCS(Muilti-Bytes Character Set，多字节字符集)。在简体中文操作系统下，ANSI编码代表GB2312编码；在日文操作系统下，ANSI编码代表JIS编码。

GB2312编码通行于我国内地，是于1980年发布的《信息交换用汉字编码字符集 基本集》，标准号为GB 2312—1980，通常被称为国标码。几乎所有的中文系统和国际化的软件都支持GB2312。

GB2312是简体中文字符集，由6763个常用汉字和682个全角的非汉字字符组成。其中汉字根据使用的频率分为两级：一级汉字(常用汉字)3755个，按汉语拼音字母排列；二级汉字(不常用汉字)3008个，按偏旁部首排列。

区位码是指每个汉字的GB2312编码的对应表示，以4位的十进制数字表示，前两位称为“区码”，后两位称为“位码”，共分为94区和94位。例如，汉字“中”的区位码为5448，转换为十六进制后就是$(36)_{16}(30)_{16}$，其中区码为54、位码为48。

为了与ASCII码一致，避开ASCII码中的前32个非图形字符，将区位码的区码和位码分别加上十进制数32[或$(20)_{16}$]，便得到国标码。例如，汉字“中”的国标码为86 80，对应的十六进制表示为$(56)_{16}(50)_{16}$，通常表示为5650H(H在这里代表十六进制)。

汉字机内码采用2字节存储一个汉字，为了避免与ASCII码冲突而出现二义性，在将国标码转换为机内码时对每字节(8个二进制位)“高位加1”，这等同于对汉字国标码的十六进制表示加$(80)_{16}$，也就是加80H。所以，汉字的国标码与其内码之间的关系是：汉字的机内码=汉字的国标码+8080H。

汉字输入码(外码)是指用户从键盘上输入汉字时使用的汉字编码，常用的汉字输入码

Python语言程序设计入门(第二版)(微课版) 正文.indd 12 2023-06-26 10:24:45

有拼音编码(如全拼、双拼、微软拼音输入法、自然码、智能ABC、搜狗等)、字形编码(五笔、表形码、郑码输入法等)。

GB2312的出现，基本满足了汉字的计算机处理需要，但对于人名、古汉语等方面出现的罕用字，GB2312处理不了，为了能够处理这些罕用字，后来诞生了GBK及GB18030汉字字符集。

另外，在我国的香港、澳门特别行政区和台湾地区，普遍使用的是繁体中文字符集。为了统一繁体字符集编码，1984年，制定了繁体中文编码方案BIG5。在互联网中检索繁体中文网站时，所打开的网页大多都是通过BIG5编码产生的文档。

3. Unicode编码

不同ANSI编码之间互不兼容，当信息在国际上交流时，无法将属于两种语言的文字，存储在同一段 ANSI 编码的文本中。ANSI编码最大的缺点是，同一个编码值在不同的编码体系中代表不同的汉字，这就很容易造成混乱。为了消除这种混乱现象，诞生了Unicode编码。

Unicode作为编码方案，是为了解决传统字符编码方案的局限性而产生的，它为每种语言中的每个字符设定了统一且唯一的二进制编码，可以容纳100多万个符号，以满足跨语言、跨平台进行文本转换、处理的要求。

Unicode虽然统一了编码方式，但是效率不高，比如UCS-4(Unicode的标准之一)规定用4字节存储一个符号，那么每个英文字母必然有3字节是0，这对存储和传输来说都十分耗费资源，于是出现了UTF-8编码。

UTF-8(8-bit Unicode Transformation Format)是针对Unicode的一种可变长度字符编码，可用来表示Unicode标准中的任何字符，而且其编码中的第一个字节与ASCII相容。目前，UTF-8逐渐成为电子邮件、网页及其他存储或传送文字应用中优先使用的编码，Python 3.x默认使用UTF-8编码。

1.3　Python语言简介

1.3.1　Python语言发展简史

Python是由荷兰人Guido Van Rossum于1989年圣诞节期间，在阿姆斯特丹，为了打发圣诞节的无趣而开发的一种解释型脚本语言。

Python本身也是由诸多其他语言发展而来的，包括ABC、C、C++、SmallTalk、UNIX shell以及一些其他的脚本语言。

1991年，Python的第一个解释器诞生了。它是用C语言实现的，有很多语法来自C，但又受到了ABC语言的极大影响。

1994年1月，Python 1.0发布。这个版本的主要新功能包括lambda、map、filter和reduce。

Python语言程序设计入门(第二版)(微课版) 正文.indd 13　2023-06-26 10:24:45

2000年10月，Python 2.0发布。这个版本的主要新功能包括内存管理、循环检测垃圾收集器以及对Unicode的支持。

2008年12月，Python 3.0发布。Python 3.x不向后兼容Python 2.x，这意味着Python 3.x可能无法运行使用Python 2.x编写的代码。Python 3.0代表着Python语言的未来。

2019年10月，Python 3.8发布。

自从2004年以后，Python的使用率呈线性增长。

2011年1月，Python被TIOBE编程语言排行榜评为2010年年度编程语言；2019年1月，Python被评为2018年年度编程语言，8年后Python重登王座。

在IEEE Spectrum 2017年年度编程语言排行榜上，Python夺冠；在IEEE Spectrum 2018年年度编程语言排行榜上，Python卫冕；在IEEE Spectrum 2019年年度编程语言排行榜上，Python依然是榜单状元。Python在IEEE Spectrum年度编程语言排行榜上连续三年排名第一，已经成为世界上最受欢迎的编程语言。

1.3.2 Python语言的特点

Python语言具有如下特点。

(1) 入门简单：Python有相对较少的关键字，结构简单，语法定义明确。阅读编写良好的Python程序，感觉就像读英语，非常接近自然语言，学习起来更加简单。Python的这种伪代码风格，也使得人们在利用它解决问题时，能够更多地专注于问题本身而不是语言的细节。

(2) 免费开源：Python是自由/开放源代码软件(Free/Libre and Open Source Software，FLOSS)。使用者可以自由地发布软件的副本、阅读源代码、对代码进行改动并用于新的软件中，不需要支付费用，也不涉及版权问题。

(3) 高层语言：在编写Python程序时，不必考虑诸如如何管理程序使用的内存之类的底层细节。

(4) 良好的可移植性：Python的开源本质决定了它可以被移植到许多平台上，包括Linux、Windows、Macintosh、Solaris及Android等。作为脚本语言，Python程序可以在任何安装了解释器的计算机环境中执行，实现了“一次编写，到处运行”。

(5) 易于维护：Python语言通过强制缩进体现语句间的逻辑关系，提高了程序的可读性，增强了程序的可维护性。

(6) 解释型语言：使用Python语言编写的程序，可以直接从源代码运行。在计算机内部，Python解释器把源代码转换成字节码，再翻译成机器语言并运行。Python的解释型语言特性使用户可以将一些代码行在交互方式下直接测试执行，既提高了开发速度，也易于程序的编写与调试。

(7) 面向对象：Python既支持面向过程编程，也支持面向对象编程。在“面向过程”的编程语言中，程序是用过程或仅仅包含可重用代码的函数构建的。在“面向对象”的编程语言中，程序是用数据和功能组合而成的对象构建的。Python是完全面向对象的编程语言，一切皆对象，函数、模块、数字、字符串等都是对象。Python以一种非常强大而又简

Python语言程序设计入门(第二版)(微课版) 正文.indd 14 2023-06-26 10:24:45

单的方式实现了面向对象编程。

(8) 可扩展性：Python提供了丰富的API和工具，以便程序员能够轻松地使用C/C++语言来编写扩充模块。

(9) 广泛的标准库：Python有非常庞大的标准库，如math、random、time、turtle、json等，可以帮助用户处理各种工作，只要安装了Python，所有这些功能都可以使用。

(10) 丰富的第三方库：除了标准库，Python还有大量的第三方库，如NumPy、SciPy、SymPy、matplotlib、scikit-learn、Requests、Scrapy、wxPython、Pygame、Django、jieba、PyInstaller等，为我们解决各类问题提供了强大的工具。利用合适的第三方库可以节省开发时间，满足各种复杂的需求，更重要的是，合理利用这些资源会使学习与开发变得更加便利与高效。

(11) 运行速度慢：Python是解释型语言，源代码在执行时会逐行翻译成机器代码，翻译过程比较耗时，所以运行速度相对较慢。

1.3.3　Python语言的应用领域

Python语言的应用领域十分广泛，覆盖了Web开发、科学计算、系统运维、游戏开发、GUI编程、数据库编程、大数据分析、人工智能等诸多领域。

(1) Web开发：Python提供丰富的模块以支持Socket编程和多线程编程，能方便、快速地开发网络服务程序。Python还拥有优秀的Django、Tornado、Flask等Web开发框架，并且得到众多开源插件的支持，足以满足各种不同的Web开发需求。

(2) 科学计算：随着NumPy、SciPy、matplotlib和Pandas 等众多程序库的开发，Python越来越适合于进行科学计算以及绘制高质量的2D和3D图像。

(3) 系统运维：Python标准库包含多个用于调用操作系统功能的库。例如，通过第三方软件包pywin32，Python能够访问Windows API；通过IronPython，Python程序能够直接调用.NET Framework。Python已成为运维工程师首选的编程语言，在自动化运维方面深入人心。

(4) 大数据分析：Python是大数据分析的主流语言之一，网络爬虫是大数据行业获取数据的核心工具，Google 等搜索引擎公司大量地使用Python编写网络爬虫。目前，Scrapy爬虫框架的应用非常广泛，该框架就是使用Python实现的。

(5) 人工智能：Python已是人工智能领域主流的编程语言。目前世界上一些十分优秀的人工智能学习框架，如TensorFlow、Theano、Keras等，都是使用Python 实现的。

1.4　Python语言开发环境

1.4.1　下载和安装Python

在学习Python语言之前，首先要搭建编程环境。Python可以安装在Linux、Windows、

Python语言程序设计入门(第二版)(微课版) 正文.indd 15　2023-06-26 10:24:45

macOS、iOS等主流平台上。本节以Windows平台为例，介绍Python的安装过程。

在Windows上安装Python和安装普通软件一样简单，下载安装包以后，基本上单击“下一步”按钮即可。

进入Python官网下载界面，如图1-6所示。

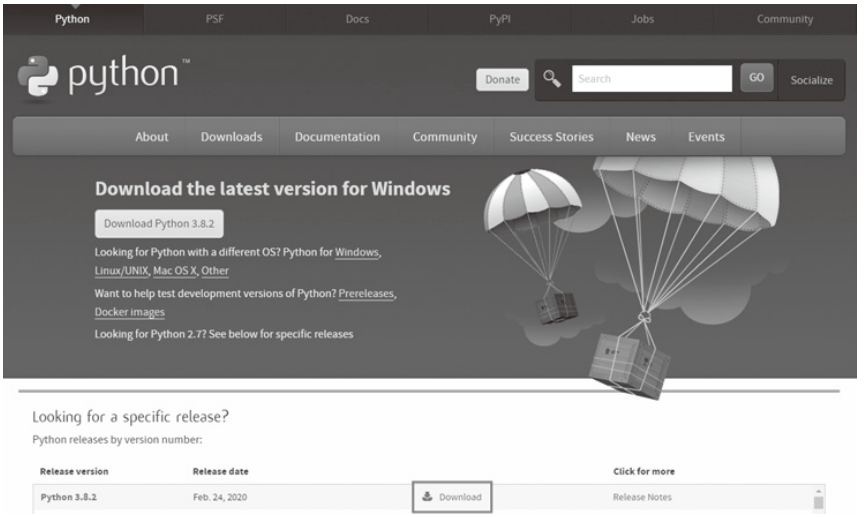
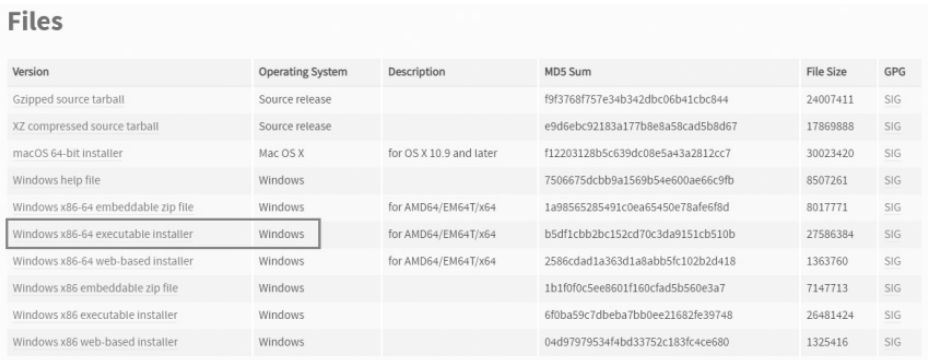


图1-6　Python官网下载界面

目前最新的版本是Python 3.8.2，单击图1-6中的版本号或Downloads选项，进入对应版本的下载界面，滚动到最后即可看到Python安装包，如图1-7所示。



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		f9f3768f757e34b342dbc06b41cbc844	24007411	SIG
XZ compressed source tarball	Source release		e9d6ebc92183a177b8e8a58cad5b8d67	17869888	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	f12203128b5c639dc08e5a43a2812cc7	30023420	SIG
Windows help file	Windows		7506675dcbb9a1569b54e600ae66c9fb	8507261	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	1a98565285491c0ea65450e78afe6f8d	8017771	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	b5df1cbb2bc152cd70c3da9151cb510b	27586384	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	2586cdad1a363d1a8abb5fc102b2d418	1363760	SIG
Windows x86 embeddable zip file	Windows		1b1f0f0c5ee8601f160cfad5b560e3a7	7147713	SIG
Windows x86 executable installer	Windows		6f0ba59c7dbeba7bb0ee21682fe39748	26481424	SIG
Windows x86 web-based installer	Windows		04d97979534f4bd33752c183fc4ce680	1325416	SIG

图1-7　Python安装包界面

关于Python安装包的几点说明：

- 以Windows x86-64开头的是64位的Python安装程序。
- 以Windows x86开头的是32位的Python安装程序。
- embeddable zip file表示.zip格式的绿色免安装版本。
- web-based installer表示通过网络进行安装。
- executable installer表示.exe格式的可执行程序，通常会选择此类安装包。

选择Windows x86-64 executable installer(这是64位的完整的离线安装包)，单击“下载”按钮，将文件保存到指定的文件夹中，如图1-8所示。

Python语言程序设计入门(第二版)(微课版) 正文.indd 16　2023-06-26 10:24:45

图1-8　下载Python安装包

双击所下载的python-3.8.2-amd64.exe文件，这样就可以正式开始安装Python了，如图1-9所示。

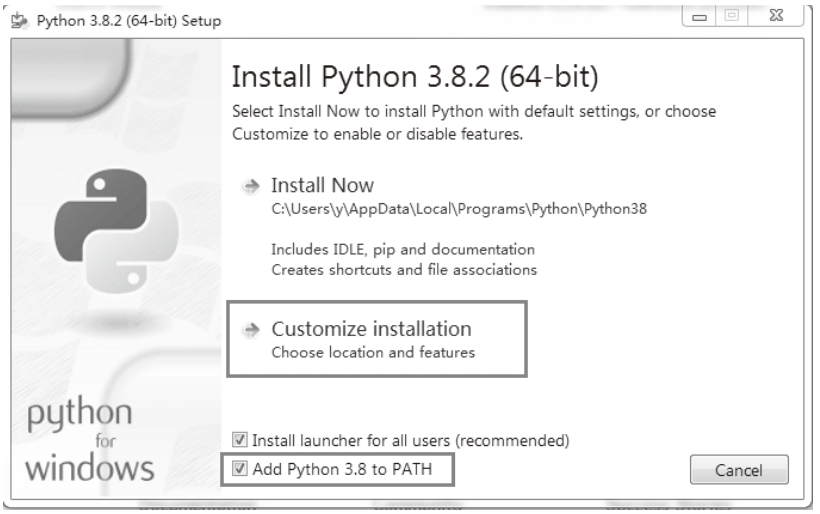


图1-9　Python安装界面

在图1-9中选中Add Python 3.8 to PATH复选框，这样就可以将 Python命令工具所在的目录添加到PATH环境变量中了，以后开发程序或运行Python命令将会非常方便。

Python支持两种安装方式：默认安装和自定义安装。默认安装会选择所有组件，并将它们安装到C盘上；自定义安装则允许手动选择想要安装的组件，并将它们安装到其他磁盘上。

在这里，我们选择自定义安装，将Python安装到常用目录下以免C盘文件过多。单击Customize installation选项，进入下一个界面。

选择想要安装的 Python 组件，若没有特殊要求，保持默认即可，如图1-10所示。

Python语言程序设计入门(第二版)(微课版) 正文.indd 17　2023-06-26 10:24:45

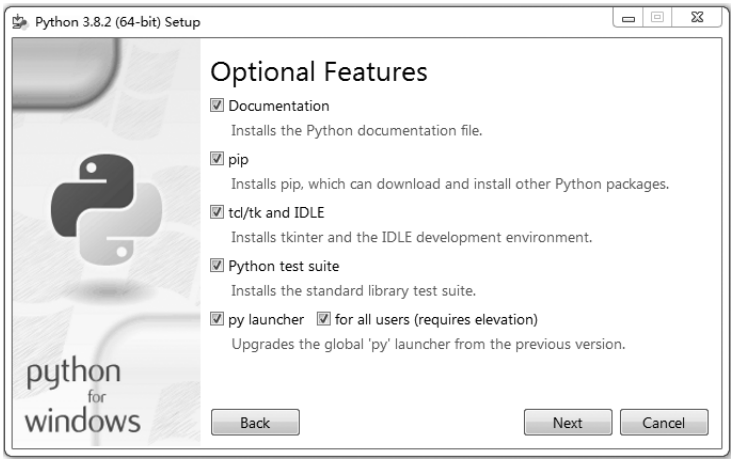


图1-10　Python的可选组件界面

单击Next按钮，进入Python的高级选项界面，进行安装目录的选择等设置，如图1-11所示。

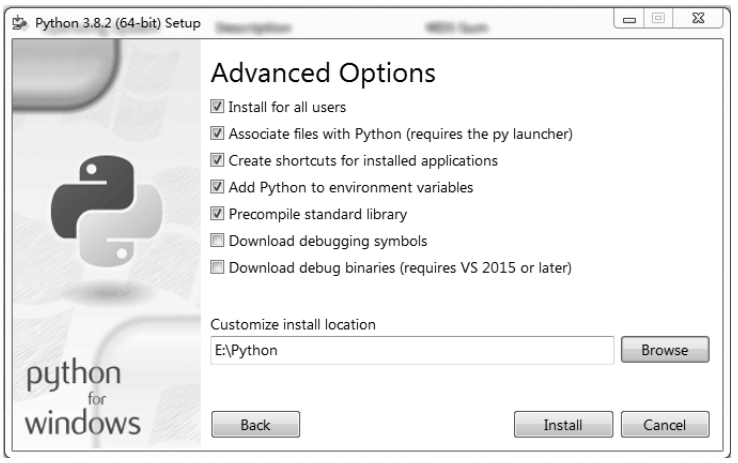


图1-11　Python的高级选项界面

单击Install按钮进行安装，安装完毕后会弹出安装成功界面，如图1-12所示。

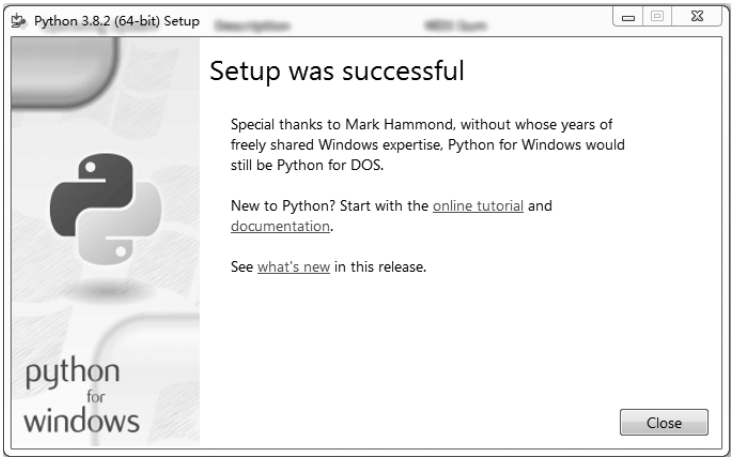


图1-12　Python的安装成功界面

Python语言程序设计入门(第二版)(微课版) 正文.indd 18　2023-06-26 10:24:45

Python安装成功后，在Windows系统的“开始”菜单中，找到Python 3.8文件夹，单击该文件夹，将看到如图1-13所示的Python命令。

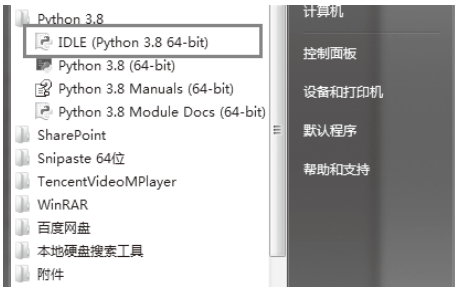


图1-13　Python命令

图1-13所示的Python命令的含义如下。

- IDLE (Python 3.8 64-bit)：启动Python 自带的集成开发环境。
- Python 3.8 (64-bit)：启动Python的命令行交互环境解释器。
- Python 3.8 Manuals (64-bit)：打开Python的帮助文档。
- Python 3.8 Module Docs (64-bit)：以内置服务器的方式打开Python模块的帮助文档。

图1-14展示了Python的命令行交互环境解释器，在Python的提示符>>>后输入Python语句print("Hello World!")，按Enter键即可解释执行该语句，得到运行结果。

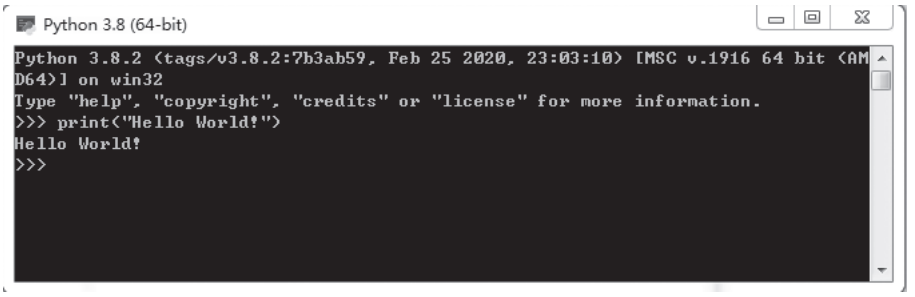


图1-14　Python的命令行交互环境解释器

这是字符界面，使用起来不太方便。通常，我们使用的是Python自带的集成开发环境IDLE，当然也可以使用其他一些集成开发环境，如PyCharm、Spyder、Vim等。

1.4.2　内置的IDLE开发环境

在Windows“开始”菜单的“所有程序”中选择Python 3.8的IDLE(Python 3.8 64-bit)命令，即可打开Python自带的集成开发环境IDLE。运行IDLE后，首先看到的是主窗口——Python 3.8.2 Shell，这是一个解释器，在这个解释器的提示符>>>后输入一条Python语句，如print("Hello,World!")，按Enter键，IDLE会立即解释执行这条语句，并显示运行结果。这样的解释器通常被称为shell。

这是一种交互模式的Python程序，解释器即时响应用户输入的语句，并给出相应的输出结果。这种方式每次只执行一条语句，适用于调试少量代码。

Python语言程序设计入门(第二版)(微课版) 正文.indd 19　2023-06-26 10:24:45

另一种是文件模式的Python程序，可将语句按Python语法格式要求写入文件中，保存成扩展名为.py的文件，然后启动Python解释器，批量执行文件中的语句。具体操作步骤如下。

(1) 在图1-15所示的主窗口中依次选择File→New File命令，如图1-16所示，即可新建Python脚本程序，如图1-17所示——窗口的标题栏显示了程序的名称，初始文件名为untitled，带有符号*，表示程序还没有保存。

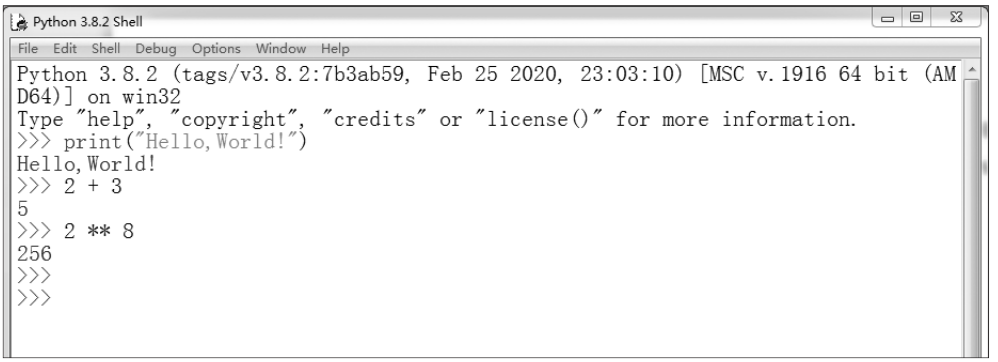


图1-15　Python自带的集成开发环境

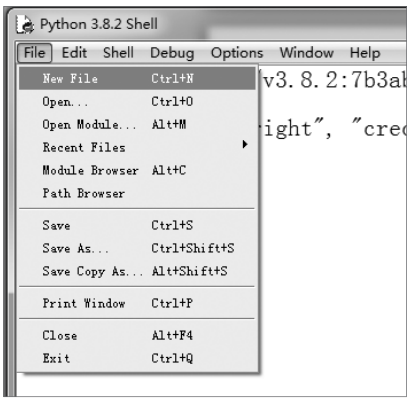


图1-16　启动Python文件模式

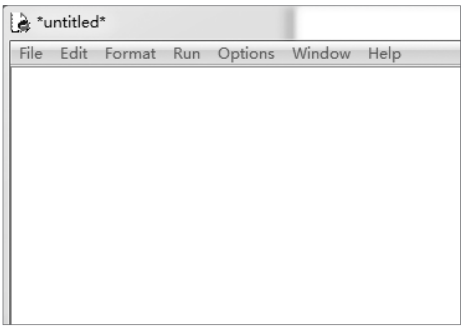


图1-17　新建Python脚本程序

(2) 在图1-17所示的窗口中编写第一个Python程序，并依次选择File→Save命令，将程序保存为hello.py文件，如图1-18所示。

图1-18　Python文件模式的程序

(3) 在图1-18所示的窗口中依次选择Run→Run Module命令，即可运行当前的Python程序，并在shell中显示运行结果，如图1-19所示。

Python语言程序设计入门(第二版)(微课版) 正文.indd 20　2023-06-26 10:24:45

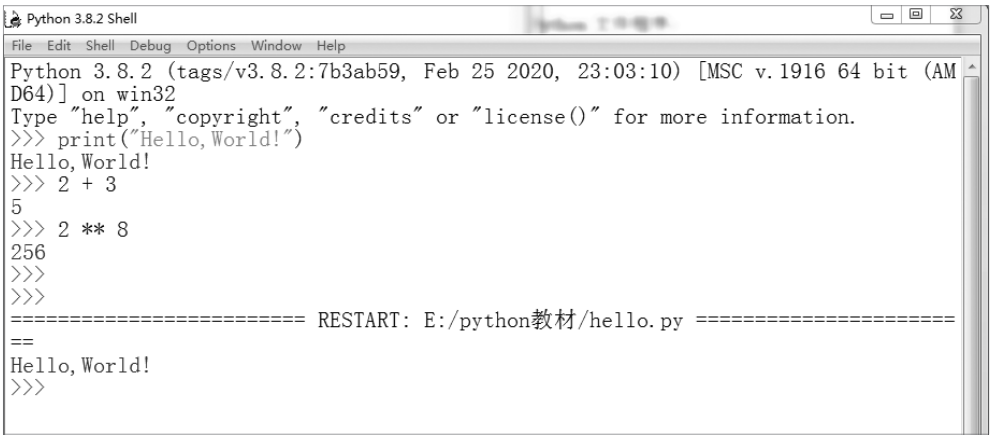


图1-19　Python程序的运行结果

打开已有Python程序的方法是：在IDLE窗口中依次选择File→Open命令，选择想要打开的文件夹和Python程序即可；也可在文件夹中右击想要打开的Python程序，依次选择Edit with IDLE命令和Edit with IDLE 3.8(64-bit)命令。

1.4.3　Python常用的其他一些集成开发环境

除了Python自带的集成开发环境IDLE，还有一些被广泛使用的其他集成开发环境，如PyCharm、Spyder、Vim等。从事数据分析和机器学习工作的用户，也可以直接安装Anaconda，因为Anaconda包含了conda、Python、NumPy、SciPy、matplotlib等数量超过180个的科学包及其依赖项。

1.5　初识Python程序

1.5.1　把Python解释器用作计算器

Python解释器可以作为计算器使用，在IDLE交互模式下，输入一行数学表达式，按Enter键，即可返回运算结果。

例1-1　在IDLE交互模式下，计算数学表达式的值。

```
>>> 99 + 88 - 5*6
157
>>> (18 + 36)*7 / 2
189.0
>>> 2 ** 8
256
>>> 2 ** 0.5
1.4142135623730951
```


Python语言程序设计入门(第二版)(微课版) 正文.indd 21　2023-06-26 10:24:45

```
>>> 17 // 3
5
>>> 17.0 // 3
5.0
>>> 17 / 3
5.666666666666667
>>> 17 % 3
2
>>>
```

说明：

- 以上表达式的语法很直白：运算符+、−、*、/代表的就是通常的加、减、乘、除。
- 整数(如88)的类型是int，带有小数部分的数字(如189.0)的类型是float。
- 除法(/)永远返回一个浮点数(float)。
- 使用//运算符时，结果将丢掉任何小数部分。
- 可使用%运算符计算余数。
- 可使用**运算符计算幂的乘方。

那么，如何计算三角函数、指数函数、对数函数的值呢？如sin(π/6)、sin(2)、ln2等。若直接输入，系统会报错，如图1-20所示。

```
>>> sin(2)
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    sin(2)
NameError: name 'sin' is not defined
>>>
```

图1-20　系统发出的报错信息

导入系统自带的math标准库，即可解决相关问题。

例1-2　导入并使用math标准库。

```
>>> import math
>>> math.sin(2)
0.9092974268256817
>>> math.sin(math.pi/6)
0.49999999999999994
>>> math.log(2)
0.6931471805599453
>>>
```

说明：

- math标准库是内置的Python模块，import语句用来导入模块。
- .是成员运算符，用来调用模块中的函数或方法。

导入math标准库后，利用内置的dir()函数，可以查阅math标准库中的内容：

Python语言程序设计入门(第二版)(微课版) 正文.indd 22　2023-06-26 10:24:45

```
>>> dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2',
'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial',
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'isqrt', 'ldexp',
'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'perm', 'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh',
'sqrt', 'tan', 'tanh', 'tau', 'trunc']
>>>
```

利用内置的help()函数，可以查阅相关函数或方法的使用说明：

```
>>> help(math.gcd)
Help on built-in function gcd in module math:

gcd(x, y, /)
    greatest common divisor of x and y
>>>
>>> math.gcd(24,90)
6
>>>
```

math模块中的内置函数gcd(x,y)能够返回x和y的最大公约数。

```
>>> help(math.factorial)
Help on built-in function factorial in module math:

factorial(x, /)
    Find x!.

    Raise a ValueError if x is negative or non-integral.

>>> math.factorial(4)
24
>>>
```

math模块中的内置函数factorial(x)能够返回x的阶乘。

学习Python语言时，一定要充分利用系统的帮助文件。

1.5.2　Python程序示例

Python语言的特点之一就是入门简单，关键字相对较少，结构简单，语法定义明确，非常接近于自然语言。阅读编写良好的Python程序，就像阅读英语一样轻松。例1-3是一个计算身体质量指数(BMI)的Python程序，试着读一下这个程序，暂时忽略语法含义，看看能否读懂。按照程序设计的IPO方法分析这个程序，思考一下如何利用Python编程来解决实际问题。

Python语言程序设计入门(第二版)(微课版) 正文.indd 23　2023-06-26 10:24:46

例1-3　编程计算BMI，从键盘输入人体的身高和体重。

```
# -*- coding: UTF-8 -*-

'''
BMI计算
作者、时间、版本号等

BMI:Body Mass Index，身体质量指数
BMI = 体重(单位kg)/(身高(单位m))^2
国际上常用的衡量人体肥胖和健康程度的重要标准，主要用于统计分析

国内BMI值(kg/m^2)：
<18.5，偏瘦；
18.5~24，正常；
24~28，偏胖；
>= 28，肥胖。
'''

height = eval(input("请输入您的身高(单位为米)："))
weight = eval(input("请输入您的体重(单位为千克)："))

bmi = weight/(height ** 2)              #计算身体质量指数

print("您的BMI为：",bmi)

# 判断身材是否合理
if bmi < 18.5:
    print("您偏瘦，请加强营养！")
elif bmi >= 18.5 and bmi < 24:
    print("恭喜您，体重正常，请注意保持！")
elif bmi >= 24 and bmi < 28:
    print("您偏胖，请加强锻炼！")
else:
    print("您体重过大，请适度节食、加强锻炼！")
```

运行IDLE，打开该程序，依次选择Run→Run Module，进入shell界面，按照提示“请输入您的身高(单位为米)：”输入1.7并按Enter键，再按照提示“请输入您的体重(单位为千克)：”输入80并按Enter键，得到的计算结果如图1-21所示。

这是一个编码格式较规范的程序，通过分析该程序，希望你能够初步了解Python语言的编码规范，逐步养成良好的编程习惯，以编写出格式规范的代码。

第1行“#-*- coding:UTF-8 -*-”是编码格式声明，在Windows平台上必须位于Python文件的第一行。由于Python 3.x版本默认将程序保存为UTF-8格式，因此一般情况下，第1行代码可以省略。

Python语言程序设计入门(第二版)(微课版) 正文.indd 24　2023-06-26 10:24:46

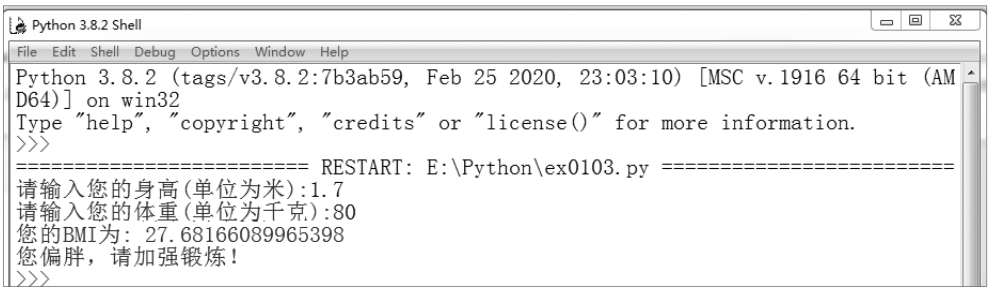


图1-21　计算结果

从第2行的三个单引号'''(或三个双引号""")开始，到第13行的三个单引号'''(或三个双引号""")结束的部分是文档字符串。文档字符串是包、模块、类或函数里的第一条语句，通常包含版本信息、参数说明、功能说明等，用于表明如何使用这个包、模块、类或函数(方法)，甚至包括使用示例和单元测试。

第14行的height = eval(input("请输入您的身高(单位为米):"))是输入语句，作用是利用input()函数接收用户的键盘输入，并利用eval()函数将输入的字符串转换为数值，然后赋值给变量height，以便进行数学运算。

第15行同第14行，作用是将用户从键盘输入的数据赋值给变量weight。

第16行的bmi = weight/(height ** 2)是赋值语句，作用是将等号右边的数学表达式的计算结果赋值给变量bmi。另外，#后面的文本是行注释。

第17行的print("您的BMI为：",bmi)是输出语句，作用是输出计算出的BMI值。

第18行的“#判断身材是否合理”是块注释语句，用于说明下方语句块的功能。

第19~26行是多分支结构的if-elif-else语句，作用是根据计算出的BMI值和国内的BMI指标体系，判断用户的身材是否合理并输出结果。

其实，这个语句块很好理解。我们可以将其看作英文，试着翻译一下。

第19和20行是一条完整的语句，第20行相对于第19行缩进了4个空格以表明逻辑相关性，可翻译为：如果(if)bmi < 18.5 真，输出“您偏瘦，请加强营养！”。

第21和22行是一条完整的语句，可翻译为：否则，如果(elif)bmi >= 18.5 and bmi < 24 真，输出“恭喜您，体重正常，请注意保持！”。

第23和24行是一条完整的语句，可翻译为：否则，如果(elif)bmi >= 24 and bmi < 28 真，输出“您偏胖，请加强锻炼！”。

第25和26行是一条完整的语句，可翻译为：否则(else)，输出“您体重过大，请适度节食、加强锻炼！”。

例1-3中的程序还不够完善，有许多需要改进的地方。请思考该程序还有哪些不足，在哪些方面还可以改进，你希望程序是什么样的，等等。随着今后的学习，希望你利用所学的知识不断地完善这个程序。对于程序设计来说，没有最好，只有更好！

1.5.3　Python程序编码规范

Python采用PEP 8——Style Guide for Python Code作为编码规范，在这里，我们结合例

Python语言程序设计入门(第二版)(微课版) 正文.indd 25　2023-06-26 10:24:46

1-3，对Python编码规范做简要说明。

- 语句：通常一行书写一条语句，每条语句的最大长度为79个字符；语句过长时，可以换行书写，换行时可以使用反斜杠\，但最好在语句的外部加上一对括号，比如( )、[ ]或{ }。
- 注释：对于不是一目了然的代码，应在行尾添加注释。注释通常以#和一个空格开始，和语句在同一行，但至少要用两个空格和语句分开；对于复杂的操作，应在操作开始前写上若干行注释，每行都要以#和一个空格开始，形成块注释。
- 缩进：每个缩进层级使用4个空格，不要使用Tab键缩进代码。
- 空行：使用必要的空行可以增强代码的可读性。通常情况下，在编码格式声明、模块导入、常量和全局变量声明、顶级定义(如函数或类的定义)之间空两行，而在方法和函数定义之间空一行。另外，在函数或方法内部，可以在必要的地方空一行以增强可读性，但应避免连续空行。
- 空格：通常情况下，在二元运算符的两侧各加一个空格；不要在逗号和冒号的前面加空格，但应该在它们的后面加空格；在函数的参数列表中，逗号之后要加一个空格；左括号之后、右括号之前不要加空格。
- 文档字符串：文档字符串是包、模块、类或函数中的第一条语句，可通过对象的doc成员自动提取，并由pydoc使用。通过导入模块，可利用help()函数查看文档字符串的内容，如图1-22所示。

```
>>> import ex0103
>>> help(ex0103)
Help on module ex0103:

NAME
    ex0103

DESCRIPTION
    BMI计算
    作者、时间、版本号等

    BMI:Body Mass Index，身体质量指数
    BMI = 体重(单位kg)/身高(单位m)^2
    国际上常用的衡量人体肥胖和健康程度的重要标准，主要用于统计分析

    国内BMI值（kg/m^2）：
    <18.5，偏瘦；
    18.5~24，正常；
    24~28，偏胖；
    >=28，肥胖。

DATA
    bmi = 27.68166089965398
    height = 1.7
    weight = 80

FILE
    e:\python\ex0103.py


>>>
```

图1-22　查看例1-3中的文档字符串

1.5.4　Python的帮助文档

学习Python语言难免会遇到各种问题，我们必须学会充分利用Python的帮助文档以解决相关问题。

Python语言程序设计入门(第二版)(微课版) 正文.indd 26　2023-06-26 10:24:46

在IDLE环境下，依次选择Help→Python Docs命令，如图1-23所示，即可进入Python文档初始界面，如图1-24所示。

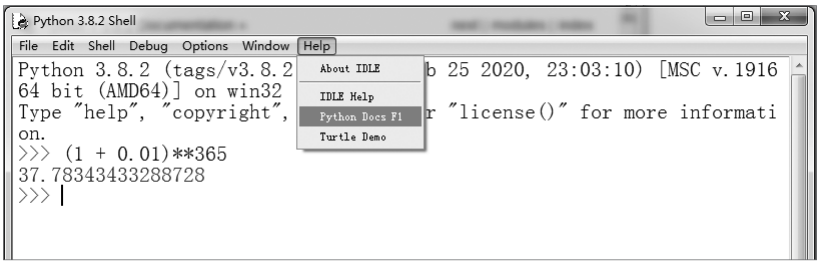


图1-23　依次选择Help→Python Docs命令

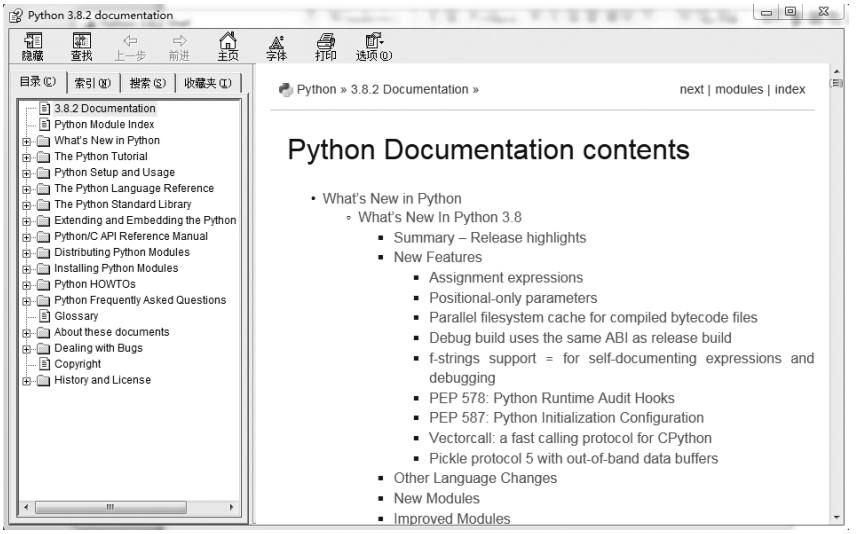


图1-24　Python文档初始界面

在图1-24所示的Python文档初始界面中，单击目录中的Python Module Index条目，即可进入Python Module Index界面，如图1-25所示，可以按索引查找相关模块的帮助信息。

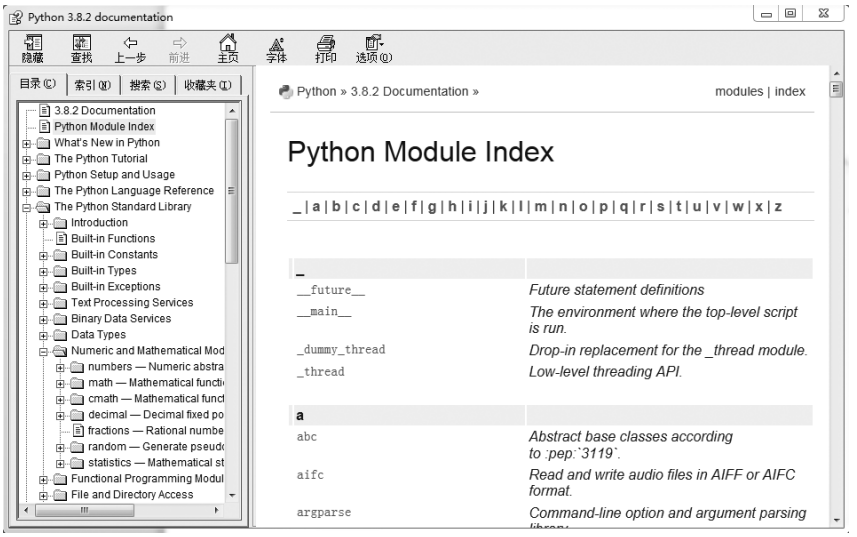


图1-25　Python Module Index界面

Python语言程序设计入门(第二版)(微课版) 正文.indd 27　2023-06-26 10:24:46

在图1-24所示的Python文档初始界面中，依次双击目录中的The Python Standard Library→Built-in Functions条目，即可进入Built-in Functions界面，如图1-26所示。单击想要查看的Python内置函数，即可得到对应的帮助文件。

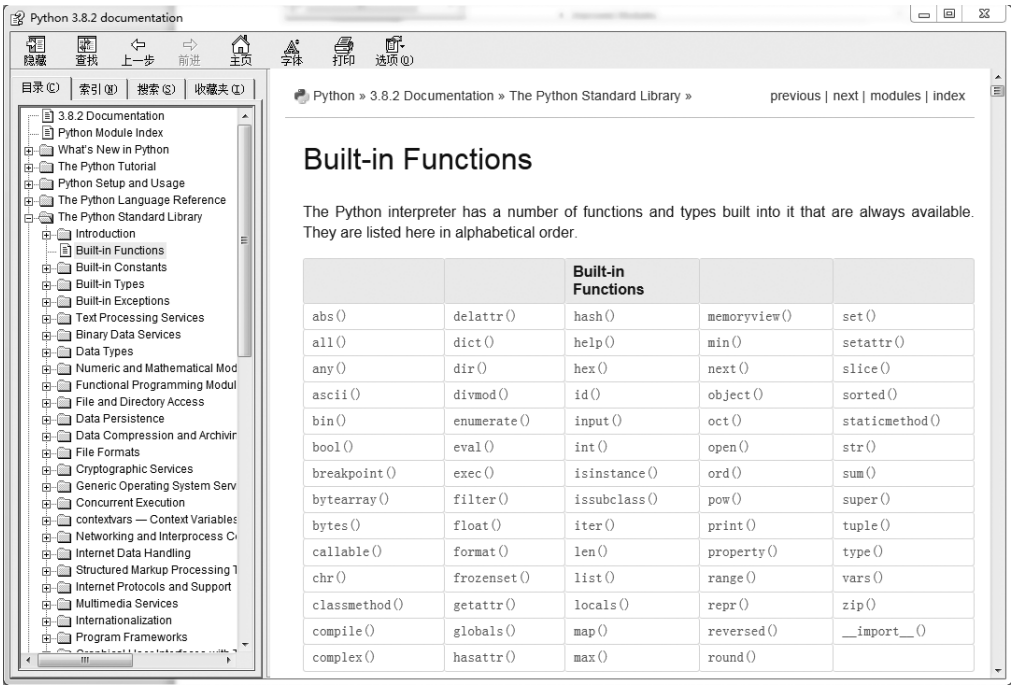


图1-26　Built-in Functions界面

在实践中，更常用的方法是，随时利用Python内置函数help()查找有关主题的帮助文档。例如，在命令提示符<<<后输入help(id)后，按Enter键即可得到有关id()函数的帮助文档，如图1-27所示。

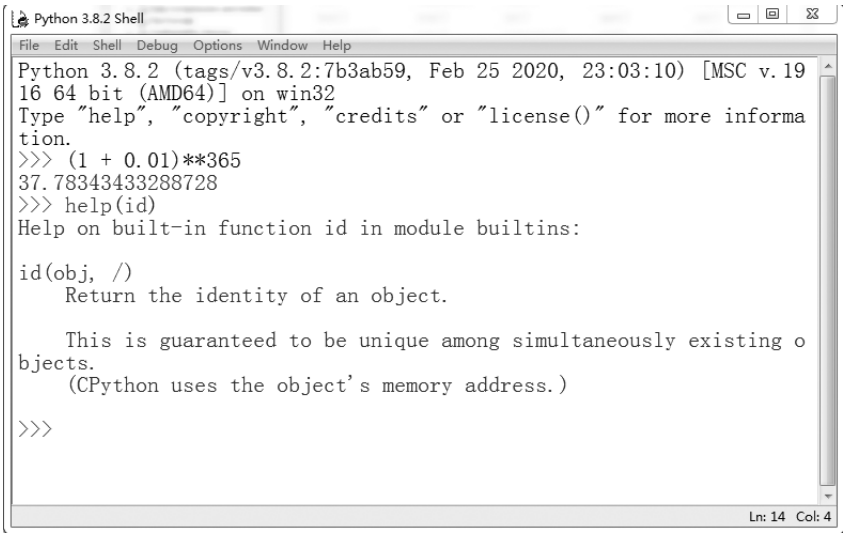


图1-27　help()函数用法演示

另外，互联网上也有各种资源，你可以充分利用。

Python语言程序设计入门(第二版)(微课版) 正文.indd 28　2023-06-26 10:24:46

1.6　习题

一、选择题

1. Python是一种(　　)类型的编程语言。

A. 机器语言　　B. 解释　　C. 编译　　D. 汇编语言

2. Python语言通过(　　)来体现语句之间的逻辑关系。

A. {}　　B. ()　　C. 缩进　　D. 自动识别逻辑

3. 以下不属于Python语言特点的是(　　)。

A. 语法简洁　　B. 依赖平台　　C. 支持中文　　D. 类库丰富

二、编程题

1. 计算$(1+0.01)^{365}$和$(1-0.01)^{365}$的值，请谈一下你的感想。

2. 参照例1-2，计算100 × COS 18° 的值。

3. 温度的刻画有两种不同的体系，分别是摄氏(Celsius)温度和华氏(Fahrenheit)温度，请参照例1-3，编程实现两种体系下温度之间的转换：

(1) 从键盘输入摄氏温度，转换为相应的华氏温度。

(2) 从键盘输入华氏温度，转换为相应的摄氏温度。

其中，转换公式如下：

$$Celsius = (Fahrenheit - 32) / 1.8$$

$$Fahrenheit = Celsius \times 1.8 + 32$$

Python语言程序设计入门(第二版)(微课版) 正文.indd 29　　2023-06-26 10:24:46