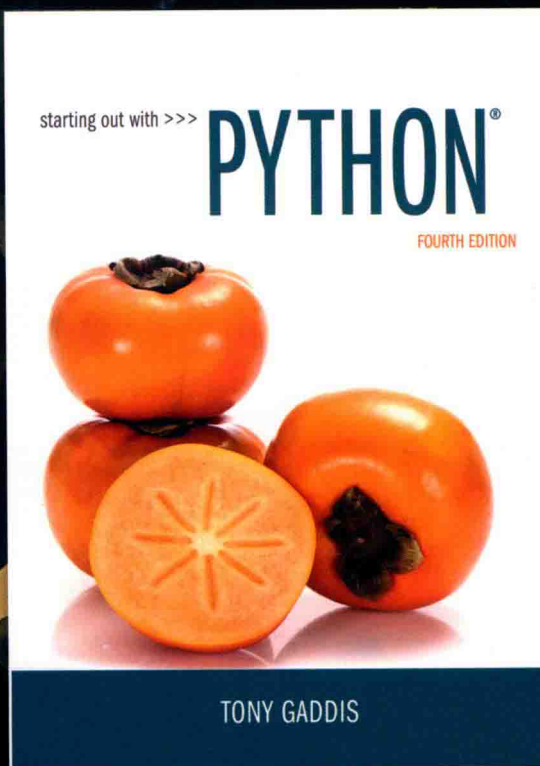


# Python程序设计基础

[美] 托尼·加迪斯 (Tony Gaddis) 著  
苏小红 叶麟 袁永峰 译

Starting Out with Python  
Fourth Edition

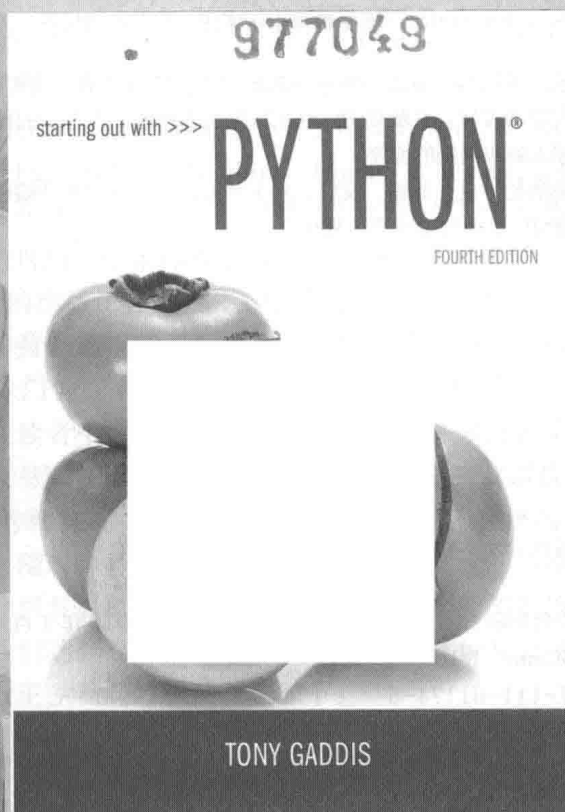


计 算 机 程 序 设 计 丛 书

# Python程序设计基础

[美] 托尼·加迪斯 (Tony Gaddis) 著  
苏小红 叶麟 袁永峰 译

Starting Out with Python  
Fourth Edition



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

Python 程序设计基础 (原书第 4 版)/(美) 托尼·加迪斯 (Tony Gaddis) 著; 苏小红, 叶麟, 袁永峰译. —北京: 机械工业出版社, 2018.11  
(计算机科学丛书)

书名原文: Starting Out with Python, Fourth Edition

ISBN 978-7-111-61174-5

I. P… II. ①托… ②苏… ③叶… ④袁… III. 软件工具—程序设计—教材 IV. TP311.561

中国版本图书馆 CIP 数据核字 (2018) 第 235235 号

本书版权登记号: 图字 01-2017-7503

Authorized translation from the English language edition, entitled *Starting Out with Python, Fourth Edition*, ISBN 978-0-13-444432-1 by Tony Gaddis, published by Pearson Education, Inc, Copyright © 2018, 2015, 2012, 2009 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press, Copyright © 2019.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

本书使用 Python 语言讲授编程概念和解决问题的方法。通过示例、伪代码、流程图和其他工具, 读者可以学习如何设计程序的逻辑, 并使用 Python 实现这些程序。本书中的程序包括突出特定编程主题的简短示例, 以及更深入解决问题的程序。

本书面向编程零基础的初学者, 清晰、友好、易于理解, 是非常理想的编程入门课程教材, 也适合作为以 Python 为语言的编程逻辑与设计课程教材。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 毕彦青

责任校对: 殷虹

印刷: 北京市荣盛彩色印刷有限公司

版次: 2019 年 1 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 33

书号: ISBN 978-7-111-61174-5

定价: 99.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机科学中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson、McGraw-Hill、Elsevier、MIT、John Wiley & Sons、Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：[www.hzbook.com](http://www.hzbook.com)

电子邮件：[hzsj@hzbook.com](mailto:hzsj@hzbook.com)

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

相比于计算机硬件，程序设计语言一直是计算机舞台上最为活跃、最为耀眼的明星。据不完全统计，目前已经诞生了超过 2000 种程序设计语言，可谓百花齐放。然而，大浪淘沙，一部分程序设计语言如泥牛入海般地悄无声影。而 Java、C、C++ 和 Python 却依然傲视群雄。程序设计语言之间此消彼长的纷争局面也恰恰反映了语言并无高低贵贱，无须非我即他。而本书的主角——Python 程序设计语言正是“天下武功唯快不破”的集大成者。Python 的“快”体现在：①起手快，与 C 语言“九层之台起于累土”的厚重相比，Python 语言的易学易用可以让编程新手有“信手拈来”“立竿见影”的轻快；②开发快，Python 语言强大的社区支持和数以万计的第三方库，可以让编程任务有“他山之石”“左右逢源”的畅快；③覆盖快，Python 语言是 TensorFlow、paddlepaddle 等最新人工智能框架的首选，可以让编程达人有“凭高望远”“敢为人先”的痛快。

本书的译者曾在哈尔滨工业大学计算机学院主讲“高级语言程序设计 (Python)”课程，在授课过程中深切地感受到一本合适的教材对学生的重要性。和选择编程语言一样，适合的就是最好的。一本好的教材不仅关注知识的传授，还可以激发学生浓厚的学习兴趣，这也是程序设计语言强调实践的关键所在。虽然关于 Python 程序语言的书籍在市面上已有不少，但大多面向具有一定编程基础的读者，不是专门为初次接触 Python 语言的读者而精心定制的。作为教科书，本书面向零基础编程的读者，对 Python 语言进行了深入浅出而细致的讲解，更难能可贵的是，本书为初学者量身定制了大量的编程习题和实践问题，可以让读者感同身受地体会 Python 之美和程序设计之美。

本书由苏小红、袁永峰和叶麟三位主译，他们都是活跃在程序设计语言课程教学实践中的一线老师。机械工业出版社华章公司编辑张梦玲在本书的整个翻译过程中提供了许多帮助，在此予以衷心感谢。

译文虽经多次修改和校对，但由于译者的水平有限，加之时间仓促，疏漏及缺点、错误在所难免，我们真诚地希望读者不吝赐教，不胜感激。

译者

2018 年 10 月于哈尔滨工业大学

本书面向编程零基础的初学者，使用 Python 语言来讲授编程概念和解决问题的方法。通过易于理解的示例、伪代码、流程图和其他工具，学生可以学习如何设计程序的逻辑，然后使用 Python 实现这些程序。本书是非常理想的编程入门课程的教材，也适合作为以 Python 为语言的编程逻辑与设计课程的教材。

本书的特点是清晰、友好、易于理解的描述。此外，它有着丰富而简明实用的示例程序，包括突出特定编程主题的简短示例，以及更深入解决问题的程序。每章提供一个或多个案例研究，对具体问题进行逐步分析，并向学生展示如何解决它。

## 先控制结构，然后类

Python 是一种完全面向对象的编程语言，但是读者不必在开始编程时就去理解面向对象的概念。本教材首先向读者介绍数据存储、输入和输出、控制结构、函数、序列和列表、文件 I/O 以及从标准类库中创建的对象的基本原理。然后，介绍如何编写类，探索继承和多态性以及学习编写递归函数。最后，介绍如何开发简单的事件驱动的 GUI 应用程序。

## 第 4 版的变化

本书清晰的写作风格和以前版本一样。然而，也做出了许多补充和改进，总结如下：

- 在第 2 章到第 5 章中增加了 Python 龟 (Turtle) 图形库的新节。机器龟图形库是 Python 的标准部分，以有趣的方式向从来没有编写过代码的学生介绍编程概念。这个库允许学生通过编写 Python 语句在画布上移动游标来绘制图形。  
新的机器龟图形库章节设计灵活，可以作为选学内容纳入现有的教学大纲，或者完全跳过。
- 第 2 章增加了关于命名常量的新节 (2.9 节)。虽然 Python 不支持真正的常量，但可以创建变量名，这些名称象征着在程序执行时不应更改。本节教授学生避免使用“幻数”，并创建符号名，使其代码更加自文档化和易于维护。
- 第 7 章增加了使用 matplotlib 包从列表中绘制图表的新节 (7.10 节)。新节介绍了如何安装 matplotlib 包，并使用它绘制折线图、条形图和饼图。
- 第 13 章增加了在 GUI 应用程序中使用 Canvas 控件绘制图形的新节 (13.9 节)。新节介绍了如何使用 Canvas 控件绘制直线、矩形、椭圆、弧、多边形和文本。
- 增加了一些新的、更具挑战性的编程问题。
- 附录 E 是一个讨论各种形式 import 语句的新附录。
- 附录 F 是一个讨论如何使用 pip 实用程序安装第三方模块的新附录。

## 各章概览

### 第 1 章：计算机与编程

本章首先对计算机的工作方式、数据的存储和操作方式以及使用高级语言编写程序的原因进行非常具体和易于理解的解释。本章中还介绍了 Python、交互模式、脚本模式和 IDLE 环境。

## 第 2 章：输入、处理与输出

本章介绍程序的开发周期、变量、数据类型和使用顺序结构编写的简单程序。学生可以学习如何编写简单的程序：从键盘读取输入、执行数学运算并输出到屏幕上。作为设计程序的工具，本章也对伪码和流程图进行了介绍。本章最后介绍了机器龟图形库。

## 第 3 章：选择结构与布尔逻辑

本章介绍关系运算符和布尔表达式，并展示如何使用选择结构控制程序流程，涉及 if、if-else 和 if-elif-else 语句，还讨论了嵌套选择结构和逻辑运算符。本章最后讨论了如何使用选择结构测试机器龟的状态。

## 第 4 章：循环结构

本章展示如何使用 while 循环和 for 循环创建循环结构，讨论了计数器、累加器和标记，以及验证输入的循环技术。本章最后介绍如何使用循环和机器龟图形库进行绘图设计。

## 第 5 章：函数

本章首先介绍如何编写和调用 void 函数，展示了使用函数对程序进行模块化的好处，并讨论了自顶向下的设计方法。然后，介绍如何将参数传递给函数，讨论了常见的库函数，如生成随机数的函数。在学习如何调用库函数并使用其返回值之后，讲解如何定义和调用自己的函数。接下来介绍如何使用模块来组织函数。本章最后讨论了如何使用函数模块化机器龟图形库代码。

## 第 6 章：文件和异常

本章介绍顺序文件的输入和输出，学习如何读取和写入大量数据，并将数据存储为字段和记录。本章最后讨论异常并演示如何编写异常处理的代码。

## 第 7 章：列表和元组

本章介绍 Python 中序列的概念，并探讨两个常见 Python 序列的使用：列表和元组。学生学习使用列表进行类似数组的操作，例如在列表中存储对象、遍历列表、搜索列表中的元素以及计算列表中元素的总和和平均值。本章讨论了切片和许多列表方法，涉及一维和二维列表。本章还包括对 matplotlib 包的讨论，以及如何使用它从列表中绘制图表。

## 第 8 章：深入字符串

本章介绍如何更加细致地处理字符串，讨论遍历字符串中各个字符的字符串切片和算法，并介绍用于字符和文本处理的内置函数和字符串方法。

## 第 9 章：字典和集合

本章介绍字典和集合数据结构，学习将数据存储为字典中的键值对、检索值、更改现有值、添加新键值对以及删除键值对。学生学习如何将值存储为集合中的唯一元素，并执行常见的集合操作，例如并集、交集、差集和对称差集。本章最后对对象序列化进行了讨论，并介绍了 Python pickle 模块。

## 第 10 章：类与面向对象编程

本章比较面向过程和面向对象的编程实践，涵盖了类和对象的基本概念，并讨论了属性、方法、封装和数据隐藏、\_\_init\_\_ 函数（类似于构造函数）、访问器、赋值器，学习如何使用 UML 对类进行建模，以及如何在特定问题中查找类。

## 第 11 章：继承

本章继续学习继承和多态，所涉及的主题包括超类、子类、\_\_init\_\_ 函数在继承、方法覆盖和多态中的工作方式。

## 第 12 章：递归

本章讨论递归及其在问题求解中的应用，给出了递归调用的可视化跟踪和递归应用，展示了许多任务的递归算法，例如查找因子，求最大公约数，对列表中的一系列值求和，并给出了经典汉诺塔的例子。

## 第 13 章：GUI 编程

本章讨论使用 Python 的 tkinter 模块设计 GUI 应用程序的基本方法，包括标签、按钮、输入字段、单选按钮、复选框和对话框等基本控件。学生还可以学习 GUI 应用程序中的工作方式以及如何编写回调函数来处理事件。本章包括对 Canvas 控件的讨论，以及如何使用它来绘制直线、矩形、椭圆、弧、多边形和文本。

## 附录 A：Python 安装

本附录解释如何下载和安装 Python 3 解释器。

## 附录 B：IDLE 简介

本附录概述 Python 附带的 IDLE 集成开发环境。

## 附录 C：ASCII 码表

作为参考，本附录列出 ASCII 字符集。

## 附录 D：预定义颜色

本附录列出可与机器龟图形库、matplotlib 和 tkinter 一起使用的预定义颜色名称。

## 附录 E：import 语句详解

本附录讨论使用 import 语句的各种方法。例如，可以使用 import 语句导入模块、类、函数或为模块分配别名。

## 附录 F：使用 pip 工具安装模块

本附录讨论如何使用 pip 实用工具从 Python 包索引或 PyPI 安装第三方模块。

## 附录 G：检查点参考答案

本附录给出了正文中出现的检查点问题的答案。

## 本书的组织

本教材以循序渐进的方式讲授编程。每一章都包含了一组主题，并随着学生的进展来逐步积累知识。虽然可以很容易地按照章节现在的顺序进行讲授，但可以有一定的灵活性，以按照希望的顺序进行教学安排。图 P-1 显示了章节之间的依赖关系。箭头指明讲授某章节之前必须覆盖的章节。

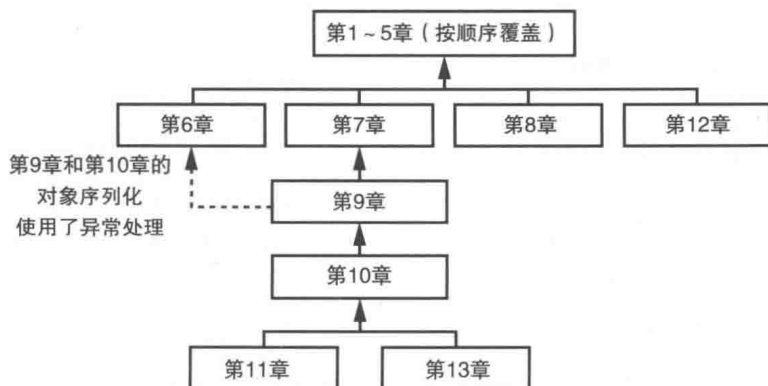







图 P-1 章节依赖关系



## 本书的特点

	<b>概念</b>	每个主要节从概念开始，概括了本节的要点。
	<b>示例程序</b>	每章都有大量完整和部分的示例程序，旨在突出当前主题。
	<b>聚光灯</b>	每章都有一个或多个聚光灯案例研究，提供详细的、循序渐进的问题分析，并告诉学生如何解决它们。
	<b>注</b>	贯穿全书，对有趣的或经常被误解的上下文相关的主题进行简短的解释。
	<b>提示</b>	提供解决不同编程问题的最佳技术。
	<b>警告</b>	提醒学生注意可能导致程序出现故障或数据丢失的编程技术或实践。
	<b>检查点</b>	贯穿全书，在学习新主题后快速检查学生的掌握情况。
	<b>复习题</b>	位于每章末，包括多项选择题、判断题、算法工作室、简答题和编程题。

## 补充材料<sup>⊖</sup>

### 学生在线资源

以下资源可在 [www.pearsonhighered.com/cs-resources](http://www.pearsonhighered.com/cs-resources) 上找到：

- 每个示例程序的源代码
- 本书的配套视频注释

### 教师资源

- 所有复习题的答案
- PPT
- 测试库

## 致谢

感谢下列评审人员的专业建议：

Sonya Dennis  
*Morehouse College*

Diane Innes  
*Sandhills Community College*

John Kinuthia  
*Nazareth College of Rochester*

Frank Liu  
*Sam Houston State University*

Haris Ribic  
*SUNY at Binghamton*

Anita Sutton  
*Germanna Community College*

Christopher Urban  
*SUNY Institute of Technology*

Nanette Veilleux  
*Simmons College*

Brent Wilson  
*George Fox University*

⊖ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请联系机械工业出版社华章公司，电话 13601156823，邮箱 wangguang@hzbook.com。——编辑注

以前版本的审稿人：

Paul Amer  
*University of Delaware*

James Atlas  
*University of Delaware*

James Carrier  
*Guilford Technical Community College*

John Cavazos  
*University of Delaware*

Desmond K. H. Chun  
*Chabot Community College*

Barbara Goldner  
*North Seattle Community College*

Paul Gruhn  
*Manchester Community College*

Bob Husson  
*Craven Community College*

Diane Innes  
*Sandhills Community College*

Daniel Jinguji  
*North Seattle Community College*

Gary Marrer  
*Glendale Community College*

Keith Mehl  
*Chabot College*

Shyamal Mitra  
*University of Texas at Austin*

Vince Offenback  
*North Seattle Community College*

Smiljana Petrovic  
*Iona College*

Raymond Pettit  
*Abilene Christian University*

Janet Renwick  
*University of Arkansas–Fort Smith*

Ken Robol  
*Beaufort Community College*

Eric Shaffer  
*University of Illinois at Urbana-  
Champaign*

Tom Stokke  
*University of North Dakota*

Ann Ford Tyson  
*Florida State University*

Karen Ughetta  
*Virginia Western Community College*

Linda F. Wilson  
*Texas Lutheran University*

还要感谢家人和朋友对我所有项目的支持。感谢 Matt Goldstein 编辑和 Kristy Alaura 助理编辑的指导和鼓励，感谢营销经理 Demetrius Hall 的辛勤工作，感谢 Sandra Rodriguez 领导的生产团队。谢谢大家！

出版者的话

译者序

前言

**第 1 章 计算机与编程** ..... 1

1.1 简介 ..... 1

1.2 硬件和软件 ..... 2

1.2.1 硬件 ..... 2

1.2.2 软件 ..... 4

1.3 计算机如何存储数据 ..... 5

1.3.1 存储数字 ..... 6

1.3.2 存储字符 ..... 7

1.3.3 高级数字存储 ..... 8

1.3.4 其他类型的数据 ..... 8

1.4 程序如何工作 ..... 9

1.4.1 从机器语言到汇编语言 ..... 11

1.4.2 高级语言 ..... 11

1.4.3 关键字、操作符和语法概述 ..... 12

1.4.4 编译器和解释器 ..... 13

1.5 使用 Python ..... 14

1.5.1 安装 Python ..... 14

1.5.2 Python 解释器 ..... 14

1.5.3 交互模式 ..... 15

1.5.4 在脚本模式下编写和运行

Python 程序 ..... 16

1.5.5 IDLE 编程环境 ..... 16

复习题 ..... 17

**第 2 章 输入、处理与输出** ..... 20

2.1 设计一个程序 ..... 20

2.1.1 程序开发周期 ..... 20

2.1.2 设计过程中的更多技术细节 ..... 21

2.1.3 理解程序将要完成的任务 ..... 21

2.1.4 决定为完成任务所需采取的  
步骤 ..... 21

2.1.5 伪码 ..... 22

2.1.6 流程图 ..... 22

2.2 输入、处理与输出 ..... 23

2.3 用 print 函数显示输出 ..... 24

2.4 注释 ..... 26

2.5 变量 ..... 27

2.5.1 用赋值语句创建变量 ..... 27

2.5.2 变量命名规则 ..... 29

2.5.3 用 print 函数显示多项内容 ..... 30

2.5.4 变量再赋值 ..... 30

2.5.5 数值数据类型和数值文本 ..... 31

2.5.6 用数据类型 str 来存储  
字符串 ..... 32

2.6 从键盘读取输入 ..... 34

2.7 执行计算 ..... 37

2.7.1 浮点数除法与整数除法 ..... 39

2.7.2 运算符的优先级 ..... 39

2.7.3 用圆括号将运算分组 ..... 40

2.7.4 指数运算符 ..... 42

2.7.5 求余运算符 ..... 42

2.7.6 将数学公式转换成程序语句 ..... 43

2.7.7 混合数据类型的表达式与数据  
类型转换 ..... 45

2.7.8 长语句拆分 ..... 46

2.8 关于数据输出的更多介绍 ..... 47

2.8.1 抑制 print 函数的换行功能 ..... 47

2.8.2 指定一个输出项分隔符 ..... 47

2.8.3 转义字符 ..... 48

2.8.4 用运算符 + 来显示多个  
输出项 ..... 49

2.8.5 数据格式化 ..... 49

2.8.6 科学记数法形式的格式化 ..... 50

2.8.7 插入逗号分隔符 ..... 51

2.8.8 指定最小域宽 ..... 51

2.8.9 格式化浮点数为百分数形式 ..... 52

2.8.10 格式化整数	52	3.5.5 再次分析判定贷款资格的 程序	96
2.9 有名常量	53	3.5.6 又一个判定贷款资格的程序	97
2.10 机器龟图形库简介	54	3.5.7 用逻辑运算符检查数据范围	98
2.10.1 使用机器龟来画线	55	3.6 布尔变量	99
2.10.2 机器龟的转向	55	3.7 机器龟图形库: 判断机器龟的 状态	100
2.10.3 将机器龟的朝向设置为特定 的角度	57	3.7.1 获取机器龟的位置	100
2.10.4 获取机器龟的当前朝向	57	3.7.2 获取机器龟的朝向	100
2.10.5 画笔的抬起和放下	57	3.7.3 检测画笔是否被放下	101
2.10.6 绘制圆和点	58	3.7.4 判断机器龟是否可见	101
2.10.7 修改画笔的宽度	59	3.7.5 获取当前颜色	101
2.10.8 改变画笔的颜色	59	3.7.6 获取画笔的线宽	102
2.10.9 修改背景的颜色	59	3.7.7 获取机器龟的画线速度	102
2.10.10 重新设置屏幕	59	复习题	107
2.10.11 指定图形窗口的大小	59	<b>第 4 章 循环结构</b>	114
2.10.12 移动机器龟到指定的位置	60	4.1 循环结构简介	114
2.10.13 获取机器龟的当前位置	60	4.2 while 循环: 条件控制的循环	115
2.10.14 控制机器龟的动画速度	61	4.2.1 while 循环是先测试的循环	117
2.10.15 隐藏机器龟	61	4.2.2 无限循环	119
2.10.16 在图形窗口中显示文本	61	4.3 for 循环: 计数控制的循环	120
2.10.17 图形填充	62	4.3.1 在 for 循环中使用 range 函数	122
2.10.18 用命令 turtle.done() 来保持 图形窗口的开放状态	63	4.3.2 在循环内部使用目标变量	123
复习题	70	4.3.3 让用户控制循环迭代	126
<b>第 3 章 选择结构与布尔逻辑</b>	76	4.3.4 生成一个取值范围从高到低 的迭代序列	127
3.1 if 语句	76	4.4 计算累加和	128
3.1.1 布尔表达式与关系运算符	77	4.5 标记	131
3.1.2 综合应用	79	4.6 验证输入的循环	133
3.2 if-else 语句	82	4.7 嵌套循环	137
3.3 字符串比较	84	4.8 机器龟图形库: 用循环语句进行 绘图设计	142
3.4 嵌套的选择结构与 if-elif-else 语句	88	复习题	145
3.4.1 测试一组条件	91	<b>第 5 章 函数</b>	150
3.4.2 if-elif-else 语句	93	5.1 函数简介	150
3.5 逻辑运算符	94	5.1.1 使用函数模块化程序的好处	150
3.5.1 运算符 and	95	5.1.2 void 函数和有返回值函数	151
3.5.2 运算符 or	95	5.2 定义和调用 void 函数	152
3.5.3 短路定值	96		
3.5.4 运算符 not	96		

5.2.1	函数名	152	6.1.2	文件访问方法	210
5.2.2	定义和调用函数	152	6.1.3	文件名和文件对象	210
5.2.3	Python 的缩进	155	6.1.4	打开文件	211
5.3	使用函数设计程序	156	6.1.5	指定文件的位置	211
5.3.1	使用函数流程图化程序	156	6.1.6	将数据写入文件	212
5.3.2	自顶向下的设计	157	6.1.7	从文件读取数据	213
5.3.3	层次图	157	6.1.8	将换行符连接到字符串	215
5.3.4	暂停执行直到用户按 Enter 键	160	6.1.9	读取字符串并删除其中的 换行符	216
5.4	局部变量	160	6.1.10	将数据追加到已有文件	217
5.5	向函数传递参数	162	6.1.11	读写数值数据	218
5.5.1	参数变量的作用域	164	6.2	使用循环处理文件	221
5.5.2	传递多个参数	165	6.2.1	使用循环读取文件并检查文件 的结尾	221
5.5.3	改变参数	166	6.2.2	使用 Python 的 for 循环读取 多行	223
5.5.4	关键字参数	168	6.3	处理记录	227
5.6	全局变量和全局常量	169	6.4	异常	237
5.7	有返回值的函数简介：生成 随机数	172	6.4.1	处理多个异常	242
5.7.1	标准库函数和 import 语句	173	6.4.2	使用 except 语句捕获所有 异常	243
5.7.2	产生随机数	173	6.4.3	显示异常的默认错误信息	244
5.7.3	交互模式下的随机数实验	176	6.4.4	else 语句	245
5.7.4	randrange、random 和 uniform 函数	179	6.4.5	finally 语句	246
5.7.5	随机数种子	179	6.4.6	如果异常没有被处理怎么办	247
5.8	自己编写有返回值的函数	181	复习题	247	
5.8.1	充分利用 return 语句	182	第 7 章 列表和元组	251	
5.8.2	如何使用有返回值的函数	182	7.1	序列	251
5.8.3	使用 IPO 图	184	7.2	列表简介	251
5.8.4	返回字符串	187	7.2.1	重复运算符	252
5.8.5	返回布尔值	188	7.2.2	使用 for 循环在列表上迭代	253
5.8.6	返回多个值	189	7.2.3	索引	253
5.9	math 模块	190	7.2.4	len 函数	254
5.10	在模块中存储函数	192	7.2.5	列表是可变的	254
5.11	机器龟图形库：使用函数模块化 代码	195	7.2.6	连接列表	256
复习题		200	7.3	列表切片	257
第 6 章 文件和异常		208	7.4	使用 in 操作符在列表中查找元素	259
6.1	文件输入和输出简介	208	7.5	列表方法和有用的内置函数	261
6.1.1	文件类型	209	7.5.1	del 语句	265

7.5.2 min 和 max 函数	265	9.1.5 删除元素	329
7.6 复制列表	266	9.1.6 获取字典中元素的数量	329
7.7 处理列表	267	9.1.7 字典中数据类型的混合	330
7.7.1 计算列表中的数值之和	269	9.1.8 创建空字典	331
7.7.2 计算列表中数值的平均值	269	9.1.9 使用 for 循环遍历字典	331
7.7.3 将列表作为参数传递给函数	270	9.1.10 常用字典方法	332
7.7.4 从函数返回一个列表	271	9.2 集合	344
7.7.5 处理列表和文件	274	9.2.1 创建集合	345
7.8 二维列表	277	9.2.2 获取集合中元素的数量	345
7.9 元组	280	9.2.3 添加和删除元素	346
7.9.1 重点是什么	281	9.2.4 使用 for 循环在集合上迭代	347
7.9.2 列表和元组间的转换	282	9.2.5 使用 in 和 not in 操作符判断 集合中的值	348
7.10 使用 matplotlib 包画出列表 数据	282	9.2.6 求集合的并集	348
7.10.1 导入 pyplot 模块	283	9.2.7 求集合的交集	349
7.10.2 绘制折线图	283	9.2.8 求两个集合的差集	349
7.10.3 绘制条形图	290	9.2.9 求集合的对称差集	350
7.10.4 绘制饼图	293	9.2.10 求子集和超集	350
复习题	296	9.3 序列化对象	354
<b>第 8 章 深入字符串</b>	<b>302</b>	复习题	359
8.1 字符串的基本操作	302	<b>第 10 章 类与面向对象编程</b>	<b>365</b>
8.1.1 访问字符串中的单个字符	302	10.1 面向过程和面向对象程序设计	365
8.1.2 字符串连接	305	10.1.1 对象可重用性	366
8.1.3 字符串是不可变的	306	10.1.2 一个常见的对象例子	366
8.2 字符串切片	307	10.2 类	367
8.3 测试、搜索和操作字符串	311	10.2.1 类定义	368
8.3.1 使用 in 和 not in 测试字符串	311	10.2.2 隐藏属性	373
8.3.2 字符串方法	311	10.2.3 在模块中存储类	375
8.3.3 重复操作符	318	10.2.4 BankAccount 类	376
8.3.4 分割字符串	319	10.2.5 __str__ 方法	379
复习题	321	10.3 使用实例	381
<b>第 9 章 字典和集合</b>	<b>326</b>	10.3.1 Accessor 和 Mutator 方法	385
9.1 字典	326	10.3.2 传递对象作为参数	388
9.1.1 创建字典	326	10.4 设计类的技巧	400
9.1.2 从字典中检索值	327	10.4.1 统一建模语言	400
9.1.3 使用 in 和 not 操作符测试字典 中的值	327	10.4.2 使用类解决问题	400
9.1.4 向已有字典中添加元素	328	10.4.3 确定一个类的任务	405
		10.4.4 这仅仅是开始	408
		复习题	408

第 11 章 继承	413
11.1 继承简介	413
11.1.1 泛化和特殊化	413
11.1.2 继承和 is a 关系	413
11.1.3 UML 图中的继承	420
11.2 多态	424
复习题	430
第 12 章 递归	432
12.1 递归简介	432
12.2 递归求解问题	434
12.2.1 使用递归计算阶乘	434
12.2.2 直接递归和间接递归	436
12.3 递归算法示例	437
12.3.1 递归求解列表中元素的和	437
12.3.2 斐波那契数列	438
12.3.3 求最大公约数	439
12.3.4 汉诺塔	440
12.3.5 递归与循环	443
复习题	443
第 13 章 GUI 编程	446
13.1 GUI	446
13.2 tkinter 模块	447
13.3 Label 控件	449
13.4 Frame 控件	451
13.5 Button 控件和信息对话框	453
13.6 使用 Entry 控件获得输入	456
13.7 使用标签显示输出	458
13.8 Radio 按钮和 Check 按钮	464
13.8.1 Radio 按钮	464
13.8.2 Radiobutton 的回调函数	466
13.8.3 Check 按钮	466
13.9 使用 Canvas 组件绘制图形	468
13.9.1 Canvas 组件的屏幕坐标系	469
13.9.2 绘制直线: create_line 方法	470
13.9.3 绘制矩形: create_rectangle 方法	472
13.9.4 绘制椭圆: create_oval 方法	473
13.9.5 绘制弧: create_arc 方法	475
13.9.6 绘制多边形: create_polygon 方法	478
13.9.7 绘制文本: create_text 方法	480
复习题	483
附录 A Python 安装	488
附录 B IDLE 简介	490
附录 C ASCII 码表	494
附录 D 预定义颜色	495
附录 E import 语句详解	499
附录 F 使用 pip 工具安装模块	501
附录 G 检查点参考答案	502

## 计算机与编程

## 1.1 简介

生活中人们越来越多地使用电脑来解决实际问题。在学校，学生使用计算机完成学业，如撰写论文、搜索文章、发送电子邮件和参加在线课程。在工作中，人们使用计算机分析数据、制作演示文稿、进行电子交易、与客户和同事在线交流、控制机器设备制造以及执行其他许多事情。在家里，人们使用电脑来完成账单支付、在线购物、与朋友和家人通信以及玩游戏等。别忘了，手机、平板电脑、智能手机、汽车导航系统以及其他许多设备也都是计算机。计算机已经覆盖我们日常生活的方方面面。

计算机之所以可以做很多事情，是因为它们可以被编程。这意味着计算机不仅仅可完成一项工作，而是可完成程序告诉它们做的任何工作。程序是计算机完成某种任务时执行的一组指令。例如，图 1-1 显示了 Microsoft Word 和 PowerPoint 这两个常用程序的屏幕截图。

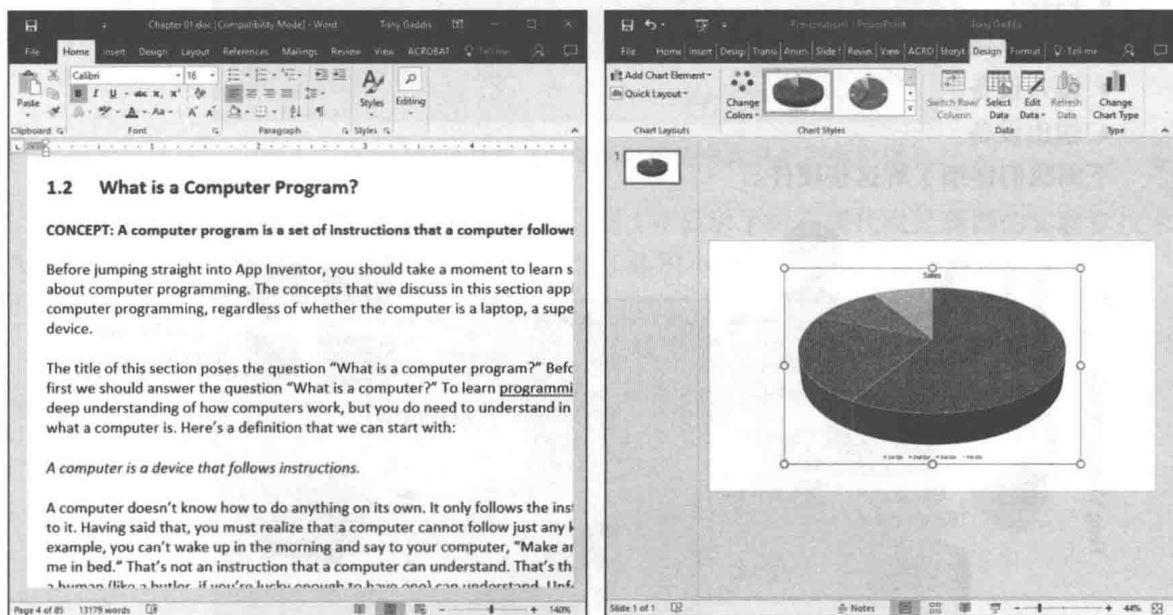


图 1-1 Microsoft Word 和 PowerPoint 应用程序

程序通常被称为软件。软件对计算机至关重要，因为它控制着计算机的一切。计算机的所有软件都是由程序员或软件开发人员创建的。所谓程序员或软件开发人员是指经过培训的具备设计、开发和测试计算机程序技能的人员。计算机程序设计是一个令人兴奋和值得从事的职业。你会发现程序员可以在商业、医药、政府、执法、农业、学术、娱乐等诸多领域择业。

本书以 Python 语言介绍了计算机编程的基本概念。对于初学者来说，Python 语言是一个不错的选择，因为它易于学习并且可以快速上手编写程序。Python 也是一种强大的语言，深受专业软件开发人员的欢迎。据报道，谷歌、美国国家航空航天局、YouTube、游戏公司、



纽约证券交易所和其他许多公司都在使用 Python。

在开始学习编程之前，你需要了解关于计算机及其工作原理的一些基本知识。这将为后续学习计算机科学奠定坚实的基础。本章中，首先，将讨论组成计算机的硬件。其次，将学习计算机如何存储数据并执行程序。最后，将简要介绍编写 Python 程序的编辑软件。

## 1.2 硬件和软件

**概念：**构成计算机的物理设备称为计算机硬件。在计算机上运行的程序称为软件。

### 1.2.1 硬件

术语硬件是指构成计算机的所有物理设备或组件。计算机不是一个单一的设备，而是一组能够协同工作的设备。就像组成交响乐团的不同乐器一样，电脑中的每个设备都有其自身的部分。

如果你曾购买过电脑，你可能会看到导购单上列出了微处理器、内存、磁盘驱动器、视频显示器、图形卡等硬件。除非你已经掌握了很多有关计算机的知识，或者至少有一位朋友了解这些知识，否则了解这些不同的硬件将会是一个挑战。如图 1-2 所示，典型的计算机系统包含以下主要硬件：

- 中央处理器 (CPU)
- 内存
- 辅助存储设备
- 输入设备
- 输出设备

下面我们详细了解这些硬件。

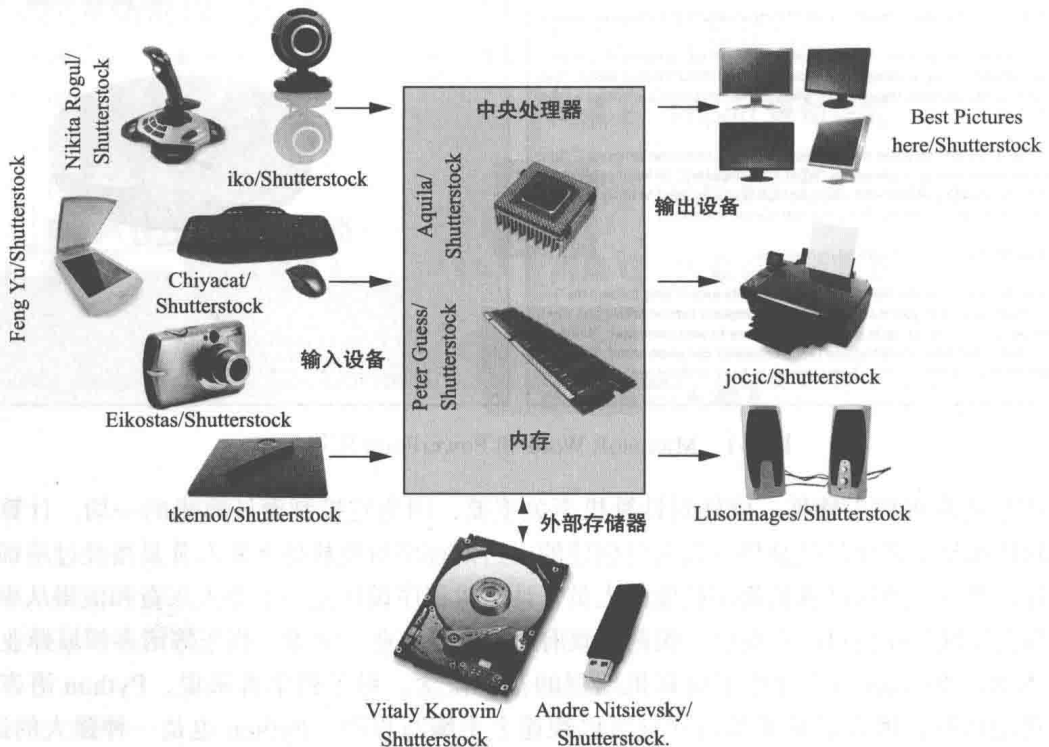


图 1-2 计算机系统的典型硬件