

全国计算机等级 考试二级教程



教育部考试中心

——Python 语言程序设计 (2018年版)

高等教育出版社

非
外
借

全国计算机等级考试二级教程

——Python 语言程序设计 (2018 年版)

Quanguo Jisuanji Dengji Kaoshi Erji Jiaocheng
——Python Yuyan Chengxu Sheji

高 天 著

高等教育出版社·北京

内容提要

本书是根据教育部考试中心制订的《全国计算机等级考试二级 Python 语言程序设计考试大纲(2018 年版)》的要求而编写的。本书共分为 11 章,第 1 章介绍程序设计的基本方法,主要包括 Python 语言概述和开发环境配置;第 2 章和第 3 章主要介绍 Python 语言基本语法元素和基本数据类型;第 4 章介绍程序的控制结构;第 5 章介绍函数和代码复用;第 6 章介绍组合数据类型;第 7 章介绍文件和数据格式化;第 8 章至第 11 章介绍 Python 的计算生态、标准库和第三方库。

本书立足自学,在内容上力求新颖,选材采用一批实用有趣的例子,以增加学习的乐趣。本教程配套自学电子教案 ppt 一套,以及全部行文代码和程序资源,可以扫码知识导图上的二维码访问这些资源,这些资源只对第一个防伪码绑定手机的读者免费使用,其他情况需付费使用。防伪码只可以绑定一次。

本教程不仅可以作为全国计算机等级考试教材,也可以作为普通高等院校及各类学校或机构的教学用书,更是一本程序设计爱好者的自学参考用书。

图书在版编目(CIP)数据

全国计算机等级考试二级教程:2018 年版. Python 语言程序设计 / 教育部考试中心编. -- 北京: 高等教育出版社, 2018.7

ISBN 978-7-04-050223-7

I. ①全… II. ①教… III. ①电子计算机-水平考试-教材②软件工具-程序设计-水平考试-教材 IV.

①TP3

中国版本图书馆 CIP 数据核字(2018)第 161231 号

策划编辑 何新权 责任编辑 何新权 封面设计 杨立新 版式设计 徐艳妮
插图绘制 于博 责任校对 刁丽丽 责任印制 耿轩

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	北京市白帆印务有限公司		http://www.hepmall.com
开 本	787mm×1092mm 1/16		http://www.hepmall.cn
印 张	14.75	版 次	2018 年 7 月第 1 版
字 数	360 千字	印 次	2018 年 7 月第 1 次印刷
购书热线	010-58581118	定 价	42.00 元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究
物料号 50223-00

积极发展全国计算机等级考试 为培养计算机应用专门人才 促进信息 产业发展作出贡献 (序)

中国科协副主席 中国系统仿真学会理事长
第五届全国计算机等级考试委员会主任委员
赵沁平

当今,人类正在步入一个以智力资源的占有和配置,知识生产、分配和使用为最重要因素的知识经济时代,也就是小平同志提出的“科学技术是第一生产力”的时代。世界各国的竞争已成为以经济为基础、以科技(特别是高科技)为先导的综合国力的竞争。在高科技中,信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域,迅速改变着人们的工作、生活和社会的结构,是当代发展知识经济的支柱之一。

在信息科学技术中,计算机硬件及通信设施是载体,计算机软件是核心。软件是人类知识的固化,是知识经济的基本表征,软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代,软件是信息化的核心,国民经济和国防建设、社会发展、人民生活都离不开软件,软件无处不在。软件产业是增长快速的朝阳产业,是具有高附加值、高投入高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程,促进从注重量的增长向注重质的提高方向发展。软件产业是关系到国家经济安全和文化安全,体现国家综合实力,决定 21 世纪国际竞争地位的战略产业。

为了适应知识经济发展的需要,大力促进信息产业的发展,需要在全民中普及计算机的基本知识,培养一批又一批能熟练运用计算机和软件技术的各行各业的应用型人才。

1994 年,国家教委(现教育部)推出了全国计算机等级考试,这是一种专门评价应试人员对计算机软硬件实际掌握能力的考试。它不限制报考人员的学历和年龄,从而为培养各行业计算机应用人才开辟了一条广阔的道路。

1994 年是推出全国计算机等级考试的第一年,当年参加考试的有 1 万余人,2017 年报考人数已达 620 万人。截至 2017 年年底,全国计算机等级考试共开考 50 次,考生人数累计达 7 665 万人,有 2 885 万人获得了各级计算机等级证书。

事实说明,鼓励社会各阶层人士通过各种途径掌握计算机应用技术,并通过等级考试对他们的能力予以科学、公正、权威性的认证,是一种比较好的、有效的计算机应用人才培养途径,符合我国的具体国情。等级考试同时也为用人单位录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看,不论是管理人员还是应试人员,对该项考试的内容和



形式都给予了充分肯定。

计算机技术日新月异。全国计算机等级考试大纲顺应技术发展和需求的变化,从2010年开始对新版考试大纲进行调研和修订,在考试体系、考试内容、考试形式等方面都做了较大调整,希望等级考试更能反映当前计算机技术的应用实际,使培养计算机应用人才的工作更健康地向前发展。

全国计算机等级考试取得了良好的效果,这有赖于各有关单位专家在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中付出的大量心血和辛勤劳动,他们为这项工作的开展作出了重要的贡献。我们在此向他们表示衷心的感谢!

我们相信,在21世纪知识经济和加快发展信息产业的形势下,在教育部考试中心的精心组织领导下,在全国各有关专家的大力配合下,全国计算机等级考试一定会以“激励引导成才,科学评价用才,服务社会选材”为目标,服务考生和社会,为我国培养计算机应用专门人才的事业作出更大的贡献。

前 言

考试不是目的,但却是有益学习的过程!

全国计算机等级考试是教育部考试中心组织的水平性考试,对教学或学习内容具有重要的“指挥棒”作用。很高兴地看到,在2018年,我国最重要的计算机类水平考试增设了Python语言科目。

Python语言是一门简洁但不简单的编程语言。语法简洁、快速入门、灵活好用,任何人都可以拿来写个小程序解解闷;快速封装C代码、十几万生态库、大量高级语法设计,即使最专业的程序员也很难掌握其全部精髓。这样既接地气又高大上的编程语言“前无古人”,在可见的未来也是“后无来者”,更重要的是,Python还是一门通用语言,适合任何类型的程序设计需求。

Python语言无疑是当下最热门的编程语言。从2013年国内的默默无闻到2017年的势压群芳,Python语言的推广和普及经过了“漫长”但“终有结果”的过程。有人说数据科学发展需要Python,有人说人工智能火热才关注Python,都有些道理,但从专业角度来说,Python自有其独特且终能大成的内因,无疑,这门语言必将或已经成为计算机发展历史上具有标志性作用的重要编程语言。

伟大的精神才能产生伟大的作品。如果把Python语言看成作品,那么一定要向Guido van Rossum致敬!致敬他领导并设计了这样简洁好用的工具、致敬他坚定开源开放造福世界的精神、致敬他精深专业并勇于否定的决心。享受Python编程乐趣的同时,更要领会精神,以创造真正价值为导向,这样的人生才能更精彩。

为了满足广大考生学习和应考的需要,教育部考试中心组织专家经过多次研讨和反复推敲,制订并颁布了《全国计算机等级考试二级Python语言程序设计考试大纲(2018年版)》,并根据考试大纲要求组织编写了本教程。

本书在内容上力求新颖,选材尽量采用有趣的例子,每节内有一个小提示,每章开始列明学习要点,并配有知识导图辅助说明重点。教程配套自学电子教案ppt一套,以及全部行文代码和程序资源,可以扫码知识导图上的二维码访问这些资源,这些资源只对第一个防伪码绑定手机的读者免费使用,其他情况需付费使用。防伪码只可以绑定一次。

本教程在编写过程中得到了教育部考试中心和高等教育出版社的大力支持和深切关怀,在此深表谢意。同时特别感谢杨雅婷、李天龙两位研究生的工作,他们负责了全书所有习题答案的整理以及所有知识导图的绘制,聪明又勤奋,是祖国未来的栋梁。

由于时间仓促、水平有限,书中仍可能存在疏漏或错误之处,敬请广大读者批评指正。

编 者

目 录

第 1 章 程序设计基本方法	1
1.1 程序设计语言	2
1.1.1 程序设计语言概述	2
1.1.2 编译和解释	2
1.1.3 计算机编程	3
1.2 Python 语言概述	4
1.2.1 Python 语言的发展	4
1.2.2 Python 最小程序	5
1.3 Python 开发环境配置	5
1.3.1 Python 开发环境安装	5
1.3.2 Python 程序的编辑方式	8
1.3.3 Python 程序的运行方式	8
1.4 程序的基本编写方法	9
1.4.1 理解问题的计算部分	9
1.4.2 IPO 程序编写方法	10
1.5 Python 程序的特点	10
1.6 实例解析——Python 小程序	11
本章小结	14
习题 1	15
第 2 章 Python 语言基本语法元素	18
2.1 程序的格式框架	19
2.1.1 缩进	19
2.1.2 注释	19
2.2 语法元素的名称	20
2.2.1 变量	20
2.2.2 命名	20
2.2.3 保留字	21
2.3 数据类型	21
2.3.1 数据类型概述	21
2.3.2 数字类型	22
2.3.3 字符串类型	22
2.4 程序的语句元素	23
2.4.1 表达式	23
2.4.2 赋值语句	24
2.4.3 引用	24
2.4.4 其他语句	25
2.5 基本输入输出函数	26
2.5.1 input() 函数	26
2.5.2 eval() 函数	27
2.5.3 print() 函数	28
2.6 源程序的书写风格	29
2.7 实例解析——倒背如流	30
本章小结	31
习题 2	31
第 3 章 基本数据类型	33
3.1 数字类型	34
3.1.1 整数类型	34
3.1.2 浮点数类型	34
3.1.3 复数类型	36
3.2 数字类型的运算	37
3.2.1 数值运算操作符	37
3.2.2 数值运算函数	38
3.3 字符串类型及格式化	40
3.3.1 字符串的索引	41
3.3.2 字符串的切片	41
3.3.3 format() 方法的基本使用	42
3.3.4 format() 方法的格式控制	43
3.4 字符串类型的操作	46
3.4.1 字符串操作符	46
3.4.2 字符串处理函数	46
3.4.3 字符串处理方法	47
3.5 类型判断和类型间转换	49
3.6 实例解析——恺撒密码	51
本章小结	54
习题 3	55
第 4 章 程序的控制结构	57
4.1 程序的三种控制结构	58
4.1.1 程序流程图	58

4.1.2 程序控制结构基础	58	6.3.1 列表的操作函数	96
4.1.3 程序控制结构扩展	59	6.3.2 列表的操作方法	98
4.2 程序的分支结构	60	6.4 字典类型	100
4.2.1 单分支结构: if	60	6.4.1 字典的定义	100
4.2.2 二分支结构: if-else	61	6.4.2 字典的索引	101
4.2.3 多分支结构: if-elif-else	61	6.5 字典类型的操作	102
4.2.4 判断条件及组合	63	6.5.1 字典的操作函数	102
4.3 程序的循环结构	64	6.5.2 字典的操作方法	102
4.3.1 遍历循环: for	64	6.6 实例解析——文本词频统计	105
4.3.2 无限循环: while	66	本章小结	107
4.3.3 循环控制: break 和 continue	67	习题 6	108
4.4 程序的异常处理	68	第 7 章 文件和数据格式化	110
4.5 实例解析——猜数字游戏	70	7.1 文件的使用	111
本章小结	74	7.1.1 文件的类型	111
习题 4	74	7.1.2 文件的打开和关闭	112
第 5 章 函数和代码复用	76	7.1.3 文件的读写	113
5.1 函数的基本使用	77	7.2 数据组织的维度	116
5.1.1 函数的定义	77	7.2.1 一维数据	116
5.1.2 函数的使用	77	7.2.2 二维数据	117
5.2 函数的参数传递	79	7.2.3 高维数据	117
5.2.1 可选参数传递	79	7.3 一维数据的处理	118
5.2.2 参数名称传递	79	7.3.1 一维数据的表示	118
5.2.3 函数的返回值	79	7.3.2 一维数据的存储	118
5.3 变量的作用域	80	7.3.3 一维数据的处理	119
5.3.1 局部变量	80	7.4 二维数据的处理	120
5.3.2 全局变量	81	7.4.1 二维数据的表示	120
5.4 代码复用	81	7.4.2 二维数据的存储	120
5.5 实例解析——软文的诗词风	82	7.4.3 二维数据的处理	121
本章小结	87	7.5 实例解析——国家财政数据趋势 演算	122
习题 5	87	本章小结	126
第 6 章 组合数据类型	90	习题 7	126
6.1 组合数据类型的基本概念	91	第 8 章 Python 计算生态	129
6.1.1 集合类型概述	91	8.1 计算思维	130
6.1.2 序列类型概述	93	8.2 程序设计方法论	131
6.1.3 映射类型概述	94	8.2.1 自顶向下设计	131
6.2 列表类型	95	8.2.2 自底向上执行	137
6.2.1 列表的定义	95	8.3 计算生态	138
6.2.2 列表的索引	95	8.3.1 Python 标准库	139
6.2.3 列表的切片	96	8.3.2 Python 第三方库	139
6.3 列表类型的操作	96		

8.4 基本的 Python 内置函数	139	11.1 网络爬虫方向	185
8.5 实例解析——Web 页面元素 提取	141	11.1.1 requests	185
本章小结	145	11.1.2 scrapy	185
习题 8	145	11.2 数据分析方向	186
第 9 章 Python 标准库概览	147	11.2.1 numpy	186
9.1 turtle 库概述	148	11.2.2 scipy	187
9.2 turtle 库与基本绘图	149	11.2.3 pandas	187
9.2.1 窗体函数	149	11.3 文本处理方向	187
9.2.2 画笔状态函数	150	11.3.1 pdfminer	187
9.2.3 画笔运动函数	151	11.3.2 openpyxl	188
9.3 random 库概述	154	11.3.3 python-docx	188
9.4 random 库与随机数运用	155	11.3.4 beautifulsoup4	188
9.5 time 库概述	156	11.4 数据可视化方向	189
9.6 time 库与程序计时	158	11.4.1 matplotlib	189
9.7 实例解析——雪景艺术绘图	160	11.4.2 TVTK	189
本章小结	163	11.4.3 mayavi	189
习题 9	163	11.5 用户图形界面方向	190
第 10 章 Python 第三方库概览	165	11.5.1 PyQt5	190
10.1 Python 第三方库的获取和 安装	166	11.5.2 wxPython	190
10.1.1 pip 工具安装	166	11.5.3 PyGTK	190
10.1.2 自定义安装	166	11.6 机器学习方向	191
10.1.3 文件安装	166	11.6.1 scikit-learn	191
10.1.4 pip 工具使用	167	11.6.2 TensorFlow	191
10.2 PyInstaller 库概述	169	11.6.3 Theano	191
10.3 PyInstaller 库与程序打包	169	11.7 Web 开发方向	192
10.4 jieba 库概述	171	11.7.1 Django	192
10.5 jieba 库与中文分词	171	11.7.2 Pyramid	192
10.6 wordcloud 库概述	173	11.7.3 Flask	192
10.7 wordcloud 库与可视化词云	174	11.8 游戏开发方向	193
10.8 实例解析——《红楼梦》人物 出场词云	176	11.8.1 Pygame	193
10.8.1 《红楼梦》人物出场统计	176	11.8.2 Panda3D	193
10.8.2 《红楼梦》人物出场词云	179	11.8.3 cocos2d	193
本章小结	181	11.9 更多第三方库	194
习题 10	181	11.9.1 PIL	194
第 11 章 Python 第三方库纵览	184	11.9.2 SymPy	194
		11.9.3 NLTK	194
		11.9.4 WeRoBot	194
		11.9.5 MyQR	195
		本章小结	195
		习题 11	195



附录 1	考试指导	197	Python 语言程序设计考试大纲 (2018 年版)	210	
附录 2	Python 保留字表	206	附录 7	全国计算机等级考试二级 Python 语言程序设计样题及 参考答案	212
附录 3	常用 Unicode 编码表	207	附录 8	习题参考答案	218
附录 4	常用 RGB 色彩对应表	208			
附录 5	Python 内置函数全表	209			
附录 6	全国计算机等级考试二级				

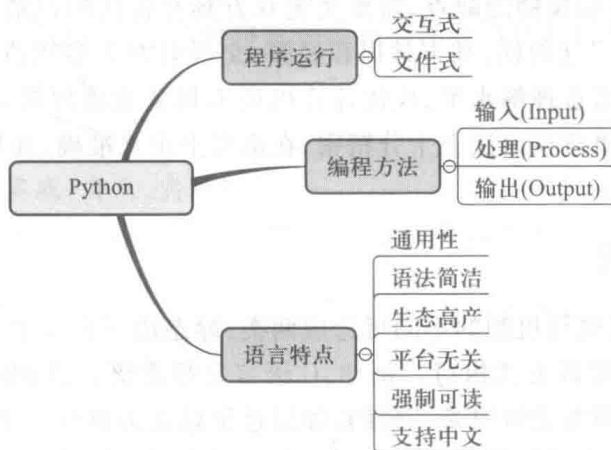
第 1 章

程序设计基本方法

学习要点

- Python 语言的特点

知识导图



1.1 程序设计语言

1.1.1 程序设计语言概述

程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系,它按照特定规则组织计算机指令,使计算机能够自动进行各种运算处理。程序设计语言也叫编程语言,本书两种说法都会出现并交替使用。计算机程序是按照程序设计语言规则组织起来的一组计算机指令。

程序设计语言发展经历了机器语言、汇编语言到高级语言的3个阶段。其中,机器语言和汇编语言都是直接操作计算机硬件的编程语言,只有计算机工程师在编写操作系统与硬件交互的底层程序或对程序进行反编译等情况下使用,这两类语言与具体CPU结构相关,不是当今程序设计主流方式。相比机器语言和汇编语言,高级语言是一种与计算机硬件无关、用于表达语法逻辑、更接近自然语言的一类计算机程序设计语言。

——为什么不能用自然语言,如中文,直接编写程序呢?

——因为自然语言不够精确,存在计算机无法理解的二义性。

自然语言具有不严密和模糊的缺点,需要交流双方具有较高的识别能力。例如,“我看见一个人在公园,带着望远镜。”这句话,基于常识和经验,交谈双方大多数情况下能够理解彼此表达的意思,但这需要较高的语言理解水平,现代计算机还不具备准确理解这种模糊性的完备智能。相比自然语言,程序设计语言在语法上十分精密,在语义上定义准确,在规则上十分严格,进而保证语法含义的唯一性。

1.1.2 编译和解释

高级语言根据计算机执行机制的不同可分成两类:静态语言和脚本语言,静态语言采用编译方式执行,脚本语言采用解释方式执行。例如,C语言是静态语言,Python语言是脚本语言。无论哪种执行方式,用户使用方法可以是一致的,如通过鼠标双击执行一个程序。

编译是将源代码转换成目标代码的过程。通常,源代码是高级语言代码,目标代码是机器语言代码,执行编译的计算机程序称为编译器(compiler)。如图1.1展示了程序的编译和执行过程,其中,虚线表示目标代码被计算机运行。编译器将源代码转换成目标代码,计算机可以立即或稍后运行这个目标代码。

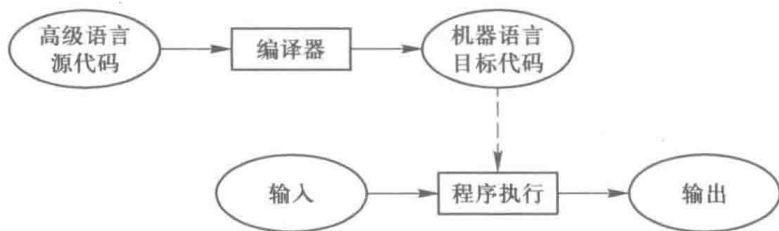


图 1.1 程序的编译和执行过程

解释是将源代码逐条转换成目标代码同时逐条运行目标代码的过程。执行解释的计算机程

序称为解释器(interpreter)。如图 1.2 展示了程序的解释和执行过程。其中,高级语言源代码与数据一同输入给解释器,然后输出运行结果。

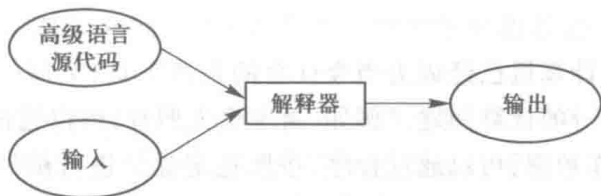


图 1.2 程序的解释和执行过程

编译和解释的区别在于编译是一次性地翻译,一旦程序被编译,不再需要编译程序或者源代码。解释则在每次程序运行时都需要解释器和源代码。这两者的区别类似于外语资料的翻译和同声传译。

简单来说,解释执行方式是逐条运行用户编写的代码,没有纵览全部代码的性能优化过程,因此执行性能略低,但它支持跨硬件或操作系统平台,对升级维护十分有利,适合非性能关键的程序运行场景。

采用编译方式执行的编程语言是静态语言,如 C 语言、Java 语言等;采用解释方式执行的编程语言是脚本语言,如 JavaScript 语言、PHP 语言等。Python 语言是一种被广泛使用的高级通用脚本编程语言,采用解释方式执行,但它的解释器也保留了编译器的部分功能,随着程序运行,解释器也会生成一个完整的目标代码。这种将解释器和编译器结合的新解释器是现代脚本语言为了提升计算性能的一种有益演进。

提示:如何学好编程?

重要的事情说三遍:实践、实践、实践!

具体来说,从小的计算需求入手,做高质量习题,在解决问题中学习,体会编程带来的快乐。

1.1.3 计算机编程

——为什么要学习计算机编程?

——因为“编程”是当今众多行业技术升级和发展的主要手段,非常有用!

——因为“编程”是件很有趣的事儿,能启迪思维,还有诗和远方……

编程能够训练思维。编程体现了一种抽象交互关系、形式化执行的思维模式,称为“计算思维”。计算思维是区别于以数学为代表的逻辑思维和以物理为代表的实证思维的第三种思维模式。编程是一个求解问题的过程,首先需要分析问题、抽象内容之间的交互关系、设计利用计算机求解问题的确定性方法,进而通过编写和调试代码解决问题,这是从抽象问题到解决问题的完整过程。计算思维的训练能够促进人类思考,增进观察力和深化对交互关系的理解。

编程能够增进认识。编写程序不单是求解计算题,它要求程序员不仅要思考解决问题的方法,更要思考如何让程序有更好的用户体验、更高的执行效率和更有趣的展示效果。不同群体、不同时代、不同文化背景对程序使用有不同理解,编程需要对时代大环境和用户群体小环境有更多认识,从细微处给出更好的程序体验,这些思考和实践将帮助程序员加深对用户行为以及社会和文化的认识。

编程能够带来乐趣。计算机编程是展示自身思想和能力的舞台,能够将程序员的所思所想变为现实。编程的开始有各种动机,或者去展示自己的青春风采,或者讽刺不文明的社会现象,



或者向爱慕的对象表达情愫,所有这些想法都可以通过编写程序变成现实,并通过互联网零成本分发获得更大的影响力。这些努力会让世界增加新的颜色、让自己变得更酷,提升心理满足感和安全感。

编程能够提高效率。计算机已经成为当今社会的普通工具,掌握一定的编程技术有助于更好地利用计算机解决所面对的计算问题。例如,对于个人照片,可以通过程序读取照片元属性自动进行归类整理;对于工作数据,可以通过程序,按照特定算法进行批处理,并绘制统计图表;对于朋友圈的好文,可以通过程序实时关注随时点赞。可见,掌握一些编程技术能够提高学习、工作和生活效率。

编程带来就业机会。程序员是信息时代最重要的工作岗位之一,国内外程序员的缺口都在百万级以上规模,就业前景广阔。程序员职业往往并不需要掌握多种编程语言,精通一种就能够获得就业机会。如果读者不喜欢自己的专业或现在的工作,那就认真学习程序设计,换个更有趣的工作吧!

很多读者都有一个误区,认为编程很难学。事实上,“编程不是一件很难的事儿”,因为编写程序有一定的框架和模式,只要理解了并且稍加练习就会有很好的学习效果。学习一门编程语言,首先要对该语言的语法做到既能系统掌握,又能灵活运用。其次要学会结合计算问题设计程序结构,从程序块、功能块角度理解并设计整个程序框架。最后要掌握解决问题的实践能力,即从理解计算问题开始,设计问题的解决方法,并通过编程语言来实现。学习计算机编程的重点在于练习,不仅要多看代码,照着编写,调试运行,还要在参考代码思路的基础上独立编写,学会举一反三。为了帮助读者掌握程序设计方法,本书将大量编程的概念和语法通过有趣的实例组织起来,并展示 Python 语言的魅力和力量。希望这样的设计能够为读者在 Python 语言学习过程中带来快乐和价值。

1.2 Python 语言概述

——“Python 语言有 2.x 和 3.x 版本,有人建议学 2.x 版本,是这样吗?”

——“Python 2.x 已经是过去,Python 3.x 是这个语言的现在和未来。”

1.2.1 Python 语言的发展

Python 语言由 Guido van Rossum 设计并领导开发,最早的可用版本诞生于 1991 年。回顾历史,1989 年 12 月,Guido 考虑启动一个开发项目打发圣诞节假期,决定为当时正在构思的脚本语言写一个解释器,因此诞生了 Python 语言。Python 语言由大牛的“偶然”所思而诞生,但经过广大程序员近 30 年的发展和应用,Python 语言变成了当代计算机技术发展的一件大事。

Python 语言解释器的全部代码都是开源的,可以在 Python 语言的主网站(<https://www.python.org/>)自由下载。

2000 年 10 月,Python 2.0 版本发布,标志着 Python 完成了自身涅槃,开启了 Python 广泛应用的新时代。2010 年,Python 2.x 系列发布了最后一个版本,主版本号为 2.7,用于终结 2.x 系列版本的发展,并且不再进行重大改进。

2008年12月,Python 3.0版本发布,这个版本解释器内部完全采用面向对象方式实现,在语法层面做了很多重大改进。这些重要修改所付出的代价是3.x系列版本代码无法向下兼容2.x系列的既有语法,因此,所有基于Python 2.x系列版本编写的代码都必须经过修改后才能被3.x系列版本解释器运行。

Python语言经历了一个痛苦但令人期待的版本更迭过程,从2008年开始,用Python编写的几万个标准库和第三方库开始了版本升级过程,这个过程前后历时8年。2016年,所有Python重要的标准库和第三方库都已经在Python 3.x版本下进行演进和发展。Python语言版本升级过程宣告结束。

提示:如何直观判断一个Python程序是否为3.x版本?

最直观、最显著、最常用的判断方法是查看print。Python 3.x版本用print()函数替换了Python 2.x版本中的print语句,两者功能一样,格式不同,如下:

```
Python2.x: >>>print "The Zen of Python"
Python3.x: >>>print("The Zen of Python")
```

1.2.2 Python 小程序

学习编程语言有一个惯例,即编写一个小程序:在屏幕上打印输出“Hello World”。这个程序虽小,却是初学者了解编程语言的第一步。

Python语言的最小程序只有一行代码,如下:

```
1 | print("Hello World")
```

上述代码中,print()表示将括号中引号内的信息输出到屏幕上。该代码在Python运行环境中的执行效果如下:

```
>>>print("Hello World")
Hello World
```

其中,第一行的“>>>”是Python语言运行环境的提示符,表示可以在此符号后输入Python语句。第二行是Python语句的执行结果。

Python语言支持中文等非西文字符的直接使用,带中文的Python小程序在运行环境中的执行效果如下:

```
>>>print("世界,你好")
世界,你好
```

1.3 Python 开发环境配置

1.3.1 Python 开发环境安装

运行Python程序的关键是安装Python语言解释器。Python解释器安装程序是一个轻量级

的小尺寸软件,文件大小为 25~30 MB,下载网址如下:

<https://www.python.org/downloads>

Python 安装程序也可以在本书关联的中文网站下载,网址如下:

<https://python123.io/download>

Python 主网站下载页面如图 1.3 所示,本书关联的 Python 中文网站下载页面如图 1.4 所示。

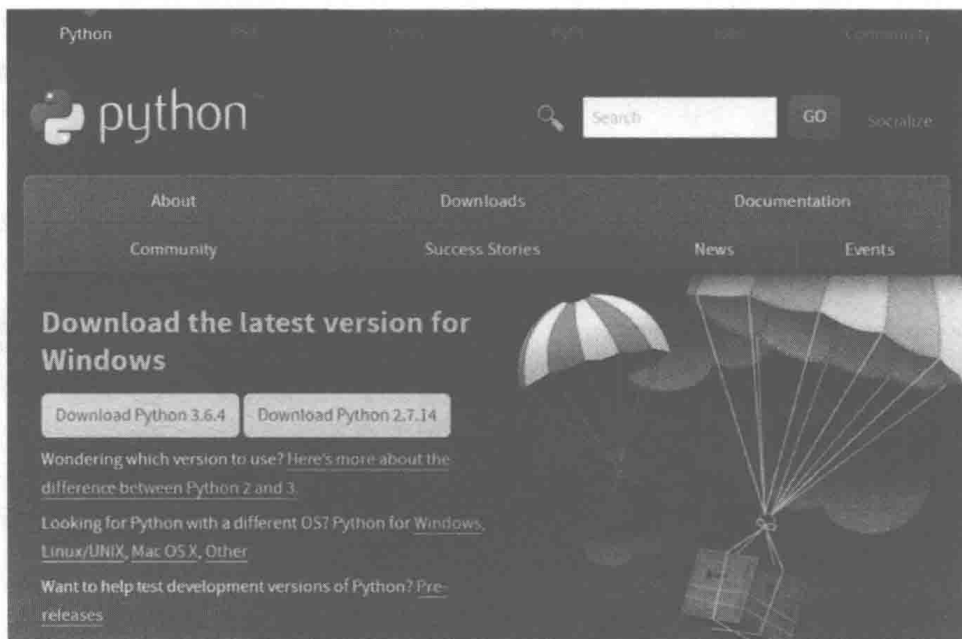


图 1.3 Python 主网站下载页面

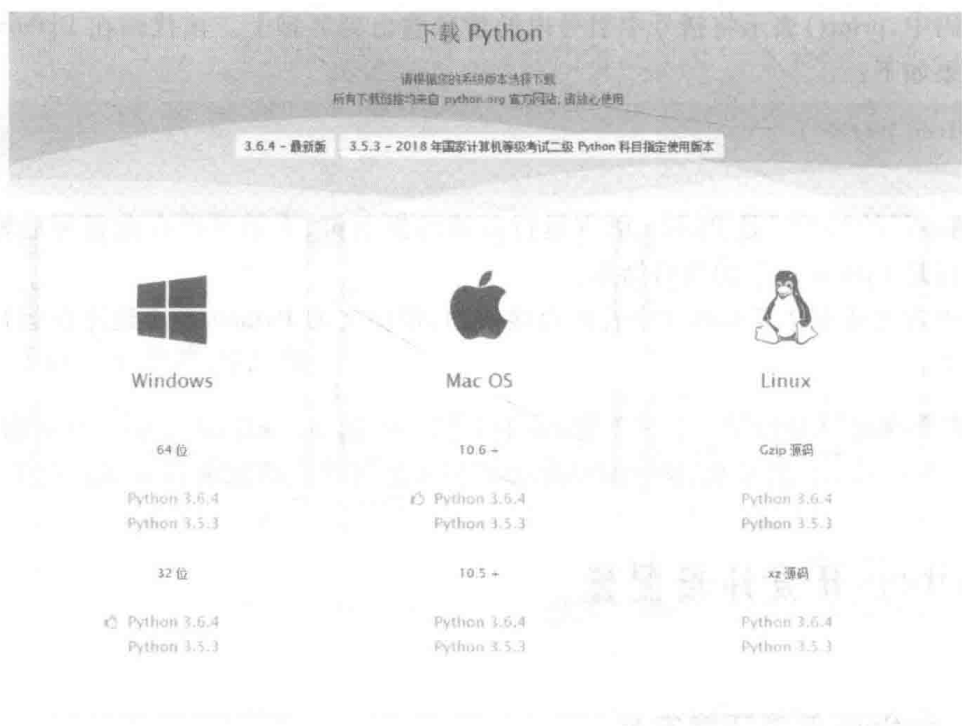


图 1.4 本书关联的 Python 中文网站下载页面

无论哪个网站,下载 Python 解释器安装程序都需要选择所用的操作系统,这里建议暂时选用 32 位的 Python 3.5.3 版本。

随着 Python 语言及全国计算机等级考试的发展,推荐使用的 Python 版本也将会演进。对于初学 Python 语言的读者来说,从稳定性和兼容性角度考虑,不建议选用最新版本,2018 年度建议使用 3.5.3 版本,本书统一使用该版本。对于 Windows 7 及更早的操作系统版本,如果不能安装 3.5.3 版本,请安装使用 3.4.2 版本。

Python 解释器的安装会启动一个引导过程,以 Windows 操作系统为例,该过程如图 1.5 所示。在该页面中,请手工选中图中矩形框内的“Add Python 3.5 to PATH”选项。



图 1.5 安装程序引导过程的启动页

安装成功后将显示如图 1.6 所示界面。

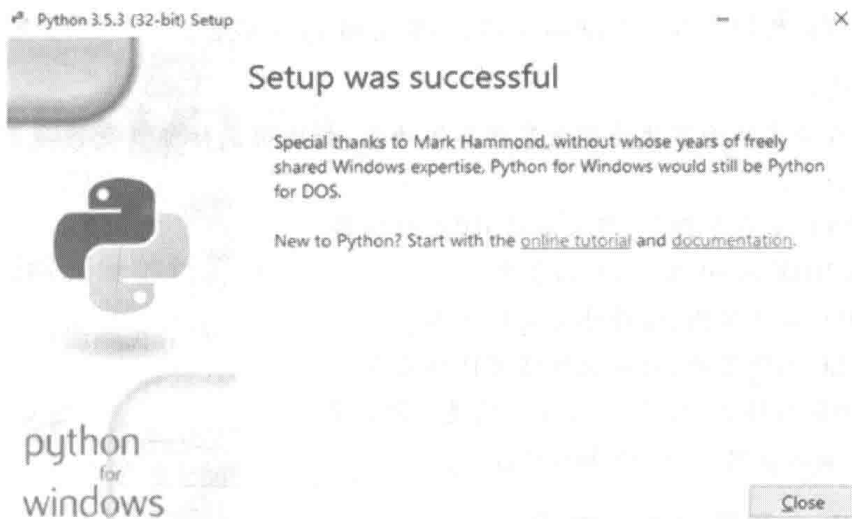


图 1.6 安装程序引导过程的结束页