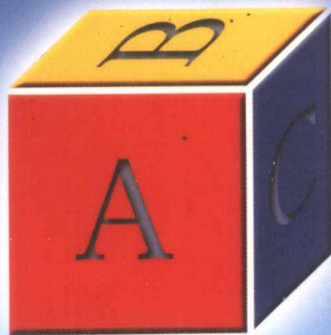


Learn to Program
Using Python

Addison
Wesley



用 Python 学编程

献给自学编程者和所有希望掌握计算机
编程艺术的从业人员



2PY



[美] Alan Gauld 著
刘彩虹 张光庭 译

Pearson
Education

清华大学出版

119

TP312P.J

G24

用 Python 学编程

(Learn to Program Using Python)

[美] Alan Gauld 著

刘彩虹 张光庭 译

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书以语法简单、功能强大且免费的 Python 语言为范例,循序渐进地讲解计算机编程的基本原理与应用,既不涉及深奥的编程理论,也不讨论任何语言细节,而是利用一些关键结构来表示问题,可把这些结构翻译成任何一种适合解决该类问题的语言。

全书分 4 部分,共 23 章。每章讨论一个主题,开头有简单的内容提要,结尾有启发思考的问题和小结。内容涉及编程的概念和理论、数据、运算符、4 种基本编程结构,以及面向对象编程的高级主题,最后教读者利用所学内容一步步地制作出两个综合案例。

这是一本专为计算机编程初学者编写的教学参考书,适合没有任何编程经验的用户学习使用。

用 Python 学编程

Alan Gauld:Learn to Program Using Python

EISBN 0-201-70938-4

Copyright © 2001 by Addison-Wesley

Authorized translation from the English language edition published by Pearson Education.

All rights reserved. For sale in the People's Republic of China Only.

北京市版权局著作权合同登记号 图字 01-2001-5314 号

本书中文简体字版由美国培生教育出版集团授权清华大学出版社在中国境内出版发行。未经出版者书面许可,任何人不得以任何方式复制或抄袭本书的任何部分。

版权所有,盗版必究。

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

书 名 : 用 Python 学编程
作 者 : [美]Alan Gauld
译 者 : 刘彩虹 张光庭
出 版 者 : 清华大学出版社(北京清华大学学研大厦, 邮编 100084)
<http://www.tup.tsinghua.edu.cn>
印 刷 者 : 北京市耀华印刷有限公司
发 行 者 : 新华书店总店北京发行所
开 本 : 异 16 印张: 15.125 字数: 299 千字
版 次 : 2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷
盘 号 : ISBN 7-900643-49-4
印 数 : 0001~4000
定 价 : 32.00 元(1CD)

前 言

为什么写这本书

我开始写这本书是为了回答两位朋友提出的问题。这两位朋友都是熟练的计算机用户，想进一步学习程序设计。由于这两个人都能访问因特网，为了节省时间，我决定从因特网上寻找一本包含他们所需知识的联机教程。但很令我失望，因为实在无法找到一本真正适合初学者使用的教程。许多教程都是讲解如何使用特定语言编程的，而且假设读者具有一定的编程知识。所以我决定自己动手为初学者编写一本真正适合他们的教程（你可以到以下网站查看本书的联机版本：<http://www.crosswinds.net/~agauld/>，在配书盘上这个版本的教程是一个zip文件）。我最初认为10到20页就已经足够了，但编写起来之后却发现文件越来越大，很快就超过了50页，而且还招来不少人访问我的网站，他们中的有些人又问了一些其他问题，有些人则要求我把某些问题讲得更清楚些。为了回答这些问题，我进一步提高了联机教程的质量，补充了一些内容。后来有些读者建议，如果这部教程整理成书出版，一定很受欢迎，于是就有了这本书。

我的专业背景

我是一名电子工程专业出身的专职程序员，20世纪70年代中期就开始投身于计算机和信息技术产业，从内嵌式微控制器到主机billing系统都有所涉及。从那时到现在，先后使用过多种计算机编程语言和操作系统。

目 录

第1部分 引 论

第1章 关于本书	3
1.1 什么人要学编程	3
1.2 本书的内容	4
1.3 为什么选择Python	5
1.4 如何阅读本书	5
1.5 本书内容的组织	6
第2章 学习编程的要求	8
2.1 关于编程者	8
2.2 先决条件	9
2.3 Python	9
第3章 什么是编程	11
3.1 定义	11
3.2 发展史简述	12
3.3 各种程序的共同特点	13
第4章 开始	20
4.1 使用Python	20

4.2 Python的出错消息	21
-----------------------	----

第2部分 程序设计基础

第5章 简单的指令序列	25
--------------------------	-----------

5.1 简单的语句	25
5.2 Python中的计算器	25
5.3 使用格式化字符串	27
5.4 注释	28
5.5 访问模块	29

第6章 原材料——数据	31
--------------------------	-----------

6.1 数据定义	31
6.2 变量定义	32
6.3 基本数据类型	33
6.4 集合	38
6.4.1 Python中的集合	38
6.4.2 其他集合类型	42
6.5 高级数据类型	44
6.5.1 文件	44
6.5.2 日期和时间	44
6.5.3 复杂/用户定义类型	45
6.6 Python的专用运算符	47

第7章 顺序结构	49
-----------------------	-----------

7.1 计算面积	51
----------------	----

7.1.1 求圆的面积	51
7.1.2 求复杂多边形的面积	51
7.2 12倍乘法表	52
第8章 循环——重复的艺术	54
8.1 For循环	54
8.2 While循环	55
8.3 再次返回12倍乘法表	56
8.4 嵌套循环	57
第9章 选择	60
9.1 链式IF语句	61
9.2 使用逻辑运算符合并判断条件	63
第10章 与用户交互	66
10.1 从用户那里获得数据	66
10.2 命令行变量	67
第11章 模块化程序设计	69
11.1 什么是模块	69
11.2 使用函数	70
11.3 使用模块	71
11.3.1 sys模块	71
11.3.2 其他的模块	72
11.4 定义新函数	73
11.4.1 从函数中返回值	74
11.4.2 函数中值的传递	75

11.4.3 默认参数值	75
11.4.4 单词计数	78
11.5 创建自己的模块	79
第12章 文件和文本处理	82
12.1 输入和输出文件	82
12.2 处理大型文件	83
12.3 修改已经存在的文件	85
12.4 往文件中添加数据	86
12.5 单词计数	87
第13章 编程风格	89
13.1 多一些注释	89
13.1.1 描述代码行	89
13.1.2 段落描述	90
13.1.3 描述文件	91
13.2 文档字符串	91
13.3 变量名	92
第14章 错误处理	94
14.1 传统的错误处理方法	94
14.2 特殊的错误处理方法	95
14.3 出现错误	96
14.4 嵌套错误处理程序	98
14.5 日志文件	99

第3部分 高级专题

第15章 递归	103
15.1 什么是递归	103
15.2 列表的递归	105
第16章 名字空间	107
16.1 简单回顾	107
16.2 Python的方法	108
16.2.1 避免函数中的名字冲突	109
16.2.2 变量名的处理	110
第17章 面向对象编程	112
17.1 什么是面向对象编程	112
17.2 数据和函数的结合	113
17.3 定义类	113
17.4 使用类	114
17.5 同样的方法, 不同的结果	115
17.6 继承	116
17.6.1 BankAccount类	116
17.6.2 InterestAccount 类	118
17.6.3 ChargingAccount类	118
17.6.4 测试系统	119
第18章 事件驱动编程	122
18.1 模拟一个事件循环	122

18.2 一个GUI程序	124
第19章 正则表达式	127
19.1 定义	127
19.2 序列	128
19.3 重复	129
19.4 条件	131
19.5 在Python中使用正则表达式	132
第20章 调试	136
20.1 停止运行、查找错误和思考原因	136
20.2 输出	137
20.3 Python调试器	138
20.3.1 启动调试器	139
20.3.2 跟踪代码	140
20.3.3 断点	142
20.4 用IDLE进行调试	144
20.5 注	146
第21章 设计解决方案	148
21.1 理解问题	148
21.1.1 需求说明	149
21.1.2 非功能性需求	149
21.1.3 优先级和范围	149
21.2 考虑选择	150
21.3 静态设计和动态设计	150
21.3.1 数据或对象	150

21.3.2 定时和同步	151
21.4 在环境中工作	151
21.4.1 操作系统的限制	152
21.4.2 与现有系统的集成	152
21.5 测试	152
21.6 配置管理	153

第4部分 案 例 研 究

第22章 文法计数器	157
22.1 计算行数、单词数及字符数	157
22.2 计算句子数目	159
22.3 将伪代码转换为模块	160
22.3.1 getCharGroups()	162
22.3.2 getPunctuation()	163
22.4 最终的文法模块	165
22.5 类和对象	166
22.5.1 文本文档	169
22.5.2 HTML文档	170
22.6 添加图形用户界面(GUI)	171
22.7 设计图形用户界面	173
第23章 猜谜游戏	180
23.1 Hangman游戏	180
23.2 为解决方案创建原型	182
23.2.1 选择一个单词	182

23.2.2 读入已猜测的字母	183
23.2.3 保留得分	183
23.3 对象框架	184
23.3.1 Game	185
23.3.2 Target	185
23.3.3 Guess	185
23.3.4 Outcome	185
23.4 抽象框架	185
23.5 测试框架——一个简单的单词游戏	187
23.6 基于文本的Hangman	189
23.7 GUI Hangman	192
23.8 其他想法	199
23.8.1 Mastermind	199
23.8.2 Rock/Scissors/Paper	199
23.8.3 Minesweeper	199
结尾	201

第5部分 附 录

附录A 安装和调试Python	205
A.1 运行安装程序	205
A.2 测试安装	206
A.3 运行IDLE	206
A.3.1 使用Python提示符	207
A.3.2 从文件中生成、保存和运行程序	207
A.4 最后工作	208

附录B 一些有用的编程语言	209
B.1 Python.....	209
B.1.1 Hello World.....	209
B.1.2 实用性.....	210
B.2 BASIC.....	210
B.2.1 Hello World.....	210
B.2.2 实用性.....	211
B.2.3 BASIC的其他版本	211
B.3 C/C++.....	212
B.3.1 Hello World.....	212
B.3.2 实用性.....	213
B.4 Java.....	213
B.4.1 Hello World.....	214
B.4.2 实用性.....	214
B.5 对象Pascal (Delphi)	215
B.5.1 Hello World.....	215
B.5.2 实用性.....	216
B.6 Perl.....	216
B.6.1 Hello World.....	217
B.6.2 实用性.....	218
B.7 Smalltalk.....	218
B.7.1 Hello World.....	218
B.7.2 实用性.....	219
B.8 其他有意义的语言	219
B.8.1 Awk.....	220
B.8.2 Eiffel.....	220
B.8.3 Lisp.....	220
B.8.4 Prolog.....	221

B.8.5	Ruby.....	221
B.8.6	Tcl/Tk.....	221
附录C	参考资料.....	222
C.1	可读书目.....	222
C.1.1	Python.....	222
C.1.2	其他语言.....	223
C.1.3	基本编程.....	224
C.1.4	面向对象编程.....	225
C.2	可访问的站点.....	227

第1部分 引论

在这一部分中，我们介绍了本书的重点、编程的基本要领和发展史、阅读本书以及成为一个成功程序员的先决条件。还简要地说明了我们将要用到的编程语言和编程环境。

第1章 关于本书

内容提要：

- 本书所面向的读者
- 要讨论的内容
- 如何阅读本书
- 本书内容的组织方式

1.1 什么人要学编程

近几年来，就像在前言中提到过的两位朋友一样，许多人都买了个人计算机（PC），并且着了迷，同时，IT技术已广泛应用到商业领域，这使许多没有专业知识的爱好者扮演起系统管理员和同事导师的角色；与此同时，Internet迅速走向大众化，带来了计算机学习的新浪潮。这三类人最终会面对同样的一个问题：标准的软件包已无法满足自己的需要。但是，市场上的成品软件常常不能满足各类用户的需求，这使得用户要么接受这些他们不太满意的软件，功能上受到限制；要么从中获得动力，去突破这些限制而成为一个程序员。

这些问题可能会以下面三种形式显现出来：

- 你需要一个程序，但是在软件商店里或Internet上找不到合适的。
- 需要自定义一个已有的应用程序，但自定义的方法不能通过简单地录制/回放宏命令来实现。
- 用户希望能创建具有丰富动态内容的Web站点，就像人们在Web上冲浪时看到的那样。不幸的是，所见即所得编辑器编写的基本HTML语句，缺乏做这种开发工作的能力或灵活性。