



Python

程序员指南

杨昆 汪兴东 / 编著

- ◇ 本书主要讲解 Python 语言的基础知识、编程及应用
- ◇ 本书的概念和思路清晰，并对相应的概念提供了大量的编程实例，具有很高的参考和应用价值
- ◇ 本书的内容深入浅出、通俗易懂，从最简单的例子着眼逐步进阶，不仅适合初学者，对使用 Python 编写各种大型、复杂应用程序的程序员来说也是一本实用的参考书

 随书附赠光盘，内含红旗中文 2000 办公平台、Python 工具箱和实例等精彩内容



中国青年出版社

e 时代网络学科推荐教程
E-time Network Discipline Recommend Textbooks



Python

程序员指南

杨昆 汪兴东 / 编著

2012



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由中国青年出版社独家出版。未经出版者书面许可，任何单位和个人不得以任何形式复制或传播本书的部分或全部。

策 划：胡守文

王修文

郭 光

责任编辑：江 颖

朱新媛

责任校对：肖新民

书 名：《Python 程序员指南》

编 著：杨昆 汪兴东

出版发行：中国青年出版社

地址：北京市东四十二条 21 号 邮政编码：100708

电话：(010) 84015588 传真：(010) 64053266

印 刷：北京新丰印刷厂

开 本：16 开

版 次：2001 年 8 月北京第 1 版

印 次：2001 年 8 月第 1 次印刷

印 数：1-5000

定 价：29.00 元 (附 1CD)

作者序

最近举办的第九届 Python 研讨会上, Guido van Rossum 宣布了 Python 软件基金会 (PSF) 正式成立。Python 是一种公共域的面向对象的动态语言。1990 年由 Guido van Rossum 开发, 用 Monty Python 剧团的名字命名, 作为一种描述性语言和快速的开发工具, Python 很快得到普及。Python 是真正的免费软件, 因为关于软件的拷贝或者发布任何用 Python 开发的应用程序没有规则限制。只要得到一份 Python 的拷贝, 就等于得到了全部源代码、个调试程序, 一个代码浏览器和一套常用的 GUI 界面。它可以在包括 Linux 在内的任何操作系统平台上运行。

Python 已经变成目前使用的最流行的语言之一, 它通常作为编译语言如 C 和描述性语言如 Perl 和 tcl/tk 之间的一种中介语言。为什么 Python 如此受欢迎呢? Python 语言本身是用语描述性的, 但是有几个特征使它不仅仅是一种简单的描述工具。比如, Python 是可展开的, 这就使得它可以适应并扩展以满足用户的需要。Python 代码易于阅读和维护, Python 也是面向对象的, 尽管你不必使用面向对象特性进行开发。

虽然在国内 Python 的脚步声刚刚踏响, 但在国外 Python 早已经成为一种成熟的解析语言并在各个方面得到了广泛的应用。Linux 的老大 RedHat 的安装程序就是用 Python 编写的, 在国内比较著名的 Linux 系统中, 红旗 Linux 和北京红旗中文 2000 软件技术有限公司 (该公司成立于 2000 年 12 月, 是香港文化传信集团之中文 2000 科技有限公司与北京中科红旗软件技术有限公司合作经营的企业, 该公司的骨干技术人员来自中科院软件所开放系统与中文信息处理中心。合作双方在打破国外软件尤其是操作系统领域的垄断地位, 振兴民族软件产业方面, 有着共同的立场和抱负——就是要为普通用户提供一个廉价易用的用户操作环境——中文 2000。) 的中文 2000 Linux 的安装程序也是用 Python 语言开发的。

迪斯尼公司的工程师发现 Python 语言在使用的时候非常方便, 更由于整个迪斯尼公司大量采用了开放源码软件, 最后毫不犹豫地用 Python 取代了 Perl 语言。现在, 迪斯尼公司生产的各种新动画片中许多功能都是由 Python 实现的。

如今, 在网络中已出现了许多研究、探讨、普及 Python 的中文站点, 但是这些站点上的内容比较零散, 很不系统, 并不适合 Python 的学习者。

本书比较完整地介绍了 Python 的知识和应用, 力图做到概念和思路清晰, 并对相应的

概念提供了大量的编程实例。读者可以在本书介绍的知识的基础上开发出更加复杂的应用程序。

本书由汪兴刚和杨昆两人编著。在编写过程中,得到了很多 Python 程序员的支持和帮助,在此深表谢意。

本书基本上覆盖了 Python 语言基础知识和应用的内容,但是限于本书的篇幅,还有一些内容不能完全叙述。比如应用和安全方面,对一种语言而言是比较重要的,但由于本书毕竟不是专著,因此不可能做到面面俱到。限于时间和作者水平,不当之处请读者指正。

杨昆 汪兴刚
2001 年 5 月

前言

如果你曾经写过大型的 shell script, 大概能了解那种感觉: 想要添加一个功能, 但是这个 script 已经够大够慢够复杂了, 或者说, 你想要加入的新功能需要调用系统功能或是其他函数, 但是这些功能/函数只有 C 才能调用。你要解决的问题好像并没有严重到要重新用 C 来写整个程序, 或者有些问题因为要用到可变长度的字符串或是特别的数据结构(像是用排序过的文件名称组成序列(list)), 用 C 来写实在比 shell 麻烦得多, 又或者你根本不是对 C 很熟。

另外一个情境是这样的: 也许你要使用好几个 C 的链接库, 但是标准开发 C 程序的过程(写/编译/测试/重新编译)实在太费时间, 你需要能快速开发好软件。又或者你已经写好一个应用程序, 这个程序可以使用一个扩展的语言来控制。你不想创造一种语言, 可是还得写好这个语言的编译器, 还得把这个编译器跟你的程序放在一起。

在这些情况之下, Python 也许正是你所需要的语言。Python 虽然简单, 却是不折不扣的程序语言。对大型的程序来说, 它比起 shell 能提供更多的结构性及支持。另外一方面, 它也提供了比 C 语言更多的错误检查。由于 Python 是一个非常高级的语言, 所以它有许多内建的数据类型, 像是有弹性的数组及字典(dictionary)等等, 如果用 C 来做的话得花上你大半天的时间。正是因为 Python 有较为一般性的数据类型, 所以它的可应用范围比起 awk 甚至是 Perl 要广很多, 最起码, Python 跟这些语言一样容易开发。

Python 的另外一个特点就是可以将程序切成小模块, 然后这些模块还可以应用在其他程序之中。Python 本身也有一个相当大的标准模块库让你使用, 或者当作学习 Python 程序设计的范例。在 Python 中也有内建的模块可以提供许多功能, 诸如: 文件 I/O、系统调用、sockets, 甚至是与 Tk 之类的 GUI 工具互动的接口。

Python 是一个直译式的语言, 可以省掉你在开发程序时不少编译及连接程序的时间。这个 Python 的直译器甚至可以交互式地使用, 让你在写一些小程序来试验 Python 语言的特性、或是测试程序时可以节省不少时间。你还可以把 Python 直译器当作计算器呢。

Python 让你可以写出非常精练且可读性高的程序。用 Python 写出的程序通常比用 C 或 C++写的程序要短得多。为什么这么说呢?

因为其高级的数据类型, 使你可以用很简单的语句(statement)表达复杂的运作过程。

Python 使用缩排来代替 C/C++ 中常见的前后括号 {};

Python 不需要变量或是参数的声明;

Python 是扩展性高的语言。如果你知道如何写 C 语言程序的话,你很容易就能在 Python 的直译器中加入新的内建函数(function)或是模块,这样做的好处是你可以让程序中关键的部分速度调到最快,或者是连接 Python 到 binary 的链接库(例如是厂商做好的图形链接库)去。一但你真的需要,也可以把 Python 直译器加入到你用 C 写的应用程序里面去,然后 Python 就变成你的应用程序的扩展或是商业化的语言了。

关于附赠光盘

为了方便读者使用，本书附带配套光盘一张。光盘包含两大块内容：

一、红旗中文2000办公平台（RedOffice beta 0.8版）

此平台是北京红旗中文 2000 软件技术有限公司开发的中文 LINUX 系统。安装时用户只需将光盘放入光驱。以光盘启动系统，即可安装，用户可选择图形安装或者是文本安装。此系统是为熟悉 LINUX 的用户准备的，请慎重使用，如果在使用中有任何问题，请与北京红旗中文 2000 软件技术有限公司技术支持部联系。电话：010-62148118 转技术支持。也可访问红旗中文 2000 公司网站查询。网址：<http://www.ch2000.com.cn> Email:support@ch2000.com.cn

二、Python工具箱

Python 工具箱内包括最新的 Python 发布版以及丰富的 Python 工具和实例。书里 Python 实例中的例子均可在“\tools\例子”中找到。另外“\tools\例子”这个目录在 Windows 下可能无法访问，需在 Linux 系统下访问。

Python 工具箱所在目录为“\tools”。其内容简要介绍如下：

1. Pthon 2.1

目录：\tools\Python2.1

说明：包括 Linux、Windows 环境下的 Python2.1 版及其他。Linux 版为 rpm 包，直接安装，Windows 版为 exe 的安装文件。

2. Jpython

目录：\tools\Jpython

3. PythonGTK

目录：\tools\pythonGTK

4. PythonXML

目录：\tools\pythonXML

5. Wxpython

目录：\tools\wxpython

6. Zope

目录：\tools\Zope

7. 官方文档

目录：\tools\官方文档

8. 例子

目录：\tools\例子

9. 未归类

目录：\tools\未归类

目 录

第一部分 利用 Python 编程

第 1 章 Python 概述

- 1.1 Python 的起源 3
- 1.2 Python 的优越性 4
- 1.3 了解 Python 语言 7
- 1.4 Python 的发展 9

第 2 章 安装并启动 Python

- 2.1 准备运行 Python 11
 - 2.1.1 安装 Python 11
 - 2.1.2 设置 Python 环境变量 13
- 2.2 使用 Python 的直译器 14
 - 2.2.1 参数的传递 15
 - 2.2.2 互动模式 15
 - 2.2.3 程序错误处理 16
 - 2.2.4 执行 Python 脚本 (script) 16
 - 2.2.5 交互式启动文件 (startup file) 16
- 2.3 在 Windows 下安装 Python 17
- 2.4 在 Apache 下设置 Python 18
 - 2.4.1 准备 18
 - 2.4.2 配置 18
 - 2.4.3 测试 19
 - 2.4.4 后话 20
- 2.5 PyGTK 在 Windows 下的安装 20

- 2.5.1 安装准备 20
- 2.5.2 安装 21
- 2.5.3 测试 “Hello, world!” 程序 21

第 3 章 Python 语法

- 3.1 把 Python 当作计算器来用 24
 - 3.1.1 数字 25
 - 3.1.2 字符串 27
 - 3.1.3 Unicode 字符串 33
 - 3.1.4 列 (List) 34
- 3.2 迈向程序设计的第一步 36

第 4 章 变量、运算符和表达式

- 4.1 Python 语言的基本数据类型 39
- 4.2 标识符和关键字 40
- 4.3 声明变量 42
- 4.4 字符、字符串变量 42
- 4.5 数值类型 45

第 5 章 Python 数据结构

- 5.1 列表 51
 - 函数程序设计工具 52
- 5.2 del 语句 53
- 5.3 序表和序列 54
- 5.4 字典 55
- 5.5 条件的进一步讨论 56
- 5.6 序列与其他类型的比较 57

第6章 控制流

6.1 if 语句	59
6.2 while 循环	61
6.3 for 循环	62
6.4 try 语句	63
6.5 range()函数	65
6.6 break 及 continue 及循环中的 else 子句	66
6.7 pass 语句	66
6.8 定义函数	67
6.8.1 预设内定参数值	69
6.8.2 关键词参数	71
6.8.3 随意的参数串	73
6.8.4 Lambda 形式	73
6.8.5 批注字符串	73

第7章 函数

7.1 定义函数	75
7.2 使用参数	77
7.2.1 预设内定参数值	77
7.2.2 关键词参数	79
7.2.3 随意的参数串	81
7.2.4 Lambda 形式	81
7.2.5 批注字符串	81

第8章 类与对象

8.1 Class (类)	83
8.2 术语的使用说明	83
8.3 Python 的可用范围 (Scopes) 及 命名空间 (Naming Spaces)	84
8.4 Class (类别) 初探	86

8.4.1 定义 Class (类别) 的语法	86
8.4.2 类别对象 (Class Objects)	87
8.4.3 特例对象 (instance objects)	88
8.4.4 Method Objects (方法对象)	89
8.5 一些随意的想法	90
8.6 继承 (Inheritance)	92
多重继承	93
8.7 Private 变量	94
8.8 其他	95
例外 (Exceptions) 也可以是类别	95

第9章 Python 语言调试

9.1 句法错	97
9.2 例外	97
9.3 例外处理	98
9.4 产生例外	100
9.5 用户自定义例外	101
9.6 定义清理动作	101

第10章 Python 的杀手程序 Zope

10.1 Zope 简介	103
10.2 Zope 动态网页发展及管理系 统简介	105
10.2.1 Zope 的内容管理器 (content manager)	105
10.2.2 新增一个对象	105
10.2.3 编辑一个 DTML 文件对象	107
10.2.4 文件的属性	108
10.2.5 Zope Document Template Markup Language	108
10.2.6 特殊 TAG 的格式	109

10.2.7 变量与运算式.....	109
10.2.8 条件式.....	111
10.2.9 循环.....	111
10.2.10 Zope 的安全机制	112
10.2.11 Zope 如何决定用户	113
10.2.12 结语.....	114

10.3 Zope 与 Python 的关系.....	114
-----------------------------	-----

第 11 章 Python 实例

11.1 Hello World 程序	119
11.2 变量和控制流	120
11.3 基本数据类型	121
11.4 基本数据类型 II: 次序和字典...122	
11.5 函数和模块	126
11.6 有用的混合运算	129
11.7 对象休止	131
11.8 定义对象	133
11.9 面向对象的概念	136
11.10 更多的面向对象的概念	140
11.11 特殊类程序	146
11.12 Python GUI 编程简介	151
11.13 TK 小部件	151
11.14 TK 部件2.....	154
11.15 TK 图形	157
11.16 TK 图形2.....	163
11.17 TK 图形3.....	170
11.18 CGI 编程	177

第二部分 wxPython 程序设计

第 12 章 wxPython 在 Win32 下编程

12.1 wxPython 简介.....	185
12.1.1 wxWindows.....	185
12.1.2 wxWdinws + Python = wxPython	186
12.2 初识 wxPython.....	186
12.2.1 哪里可以得到 wxPython.....	186
12.2.2 一个简单的例子	186
12.2.3 在 wxPython 中的事件.....	189
12.3 用 Python 创建一个 Doubletalk 浏览器.....	191
12.3.1 MDI 框架	192
12.3.2 图标	194
12.3.3 时间	194
12.3.4 主菜单	194
12.3.5 wxFileDialog	197
12.3.6 wxListCtrl.....	198
12.4 xPython 窗口布局.....	201
12.4.1 约束	201
12.4.2 布局算法	202
12.5 大小管理器 (sizer)	202
12.5.1 资源	203
12.5.2 强制力	203
12.6 wxDialog and friends	203

第三部分 Python 的高级应用

第 13 章 Python 和 XML

13.1 XML 的发展历史	207
----------------------	-----

13.2	XML 的优点	208
13.3	XML 的技术实现	209
13.4	XML 的相关技术	210
13.4.1	Xlink 与 Xpointer	210
13.4.2	Xpointer	213
13.4.3	DOM (Document Object Model)	213
13.4.4	Namespaces	214
13.4.5	TML	216
13.5	XML DOM	218
13.6	thon 和 XML	224
13.6.1	主要模块和包	225
13.6.2	文档对象模型	230
13.6.3	将 HTML 转换成 XML	231
13.6.4	将 Python 对象转换成 XML	232
13.6.5	将 XML 文档转换成 Python 对象	234
13.6.6	Python 交互式会话	236
13.6.7	重新安排 DOM 树	236
13.7	Python 和 XML 的结合	237
13.7.1	xml pickle	238
13.7.2	xml pickle 设计特点	241
13.7.3	xml objectify	243
13.7.4	xml_objectify 的设计特点	245

13.7.5	xml_objectify 的前景	247
--------	-------------------------	-----

第 14 章 Python 中的 Curses 编程

14.1	Curses 的历史与版本	249
14.2	认识 Curses 编程的思路	250
14.3	Curses 多视窗处理方式	258
14.4	Python: Curses 编程	264

第 15 章 Python 中的 TK 编程

15.1	TK 简要描述	271
15.2	基本知识	272
15.2.1	最小的 [Tkinter] 程序	272
15.2.2	main() 函数	272
15.2.3	应用几何图形管理器	274
15.2.4	菜单	274
15.2.5	接受用户输入	275

第四部分 附录

附录 A 交互式输入编辑及代换过去的内容

A.1	整行编辑	279
A.2	代换过去的内容	279
A.3	键盘连接	280
A.4	评注	281

附录 B Python 资源

第一部分 利用 Python 编程



第 1 章 Python 概述

语言是人们描述现实世界，表达自己思想观念的工具。而计算机语言是人与计算机交流的工具。一方面人类使用各种计算机语言将所关心的现实世界映射到计算机世界；另一方面，人类又可以通过计算机语言创造现实世界中并不存在的虚拟世界。

计算机的数学理论基础是图灵于 1937 年提出的图灵机模型，而现代电子计算机的体系结构及实际计算模型则是来自冯·诺依曼 1946 年提出的“程序放入内存，顺序执行”的思想，因此，现在的计算机通常被称为冯·诺依曼计算机。计算机语言的发展历程从此正式开始，从而使计算机语言的使用人员开始被称为程序员。

1.1 Python的起源

在计算机语言的发展过程中，先后出现的语言至少有几千种，但是真正能普及应用的计算机语言却是屈指可数的。一种计算机语言要能流行普及，除了要有独有的特色以外，还要切合当时的应用需求。

早期程序员们使用机器语言来进行编程运算，直接对以数字表示的机器代码进行操作。后来为了便于阅读，就将机器代码以英文字符串来表示，于是出现了汇编语言。1956 年首先在 IBM 公司的计算机上实现的由美国的计算机科学家巴科斯设计 FORTRAN 语言，标志着高级语言的到来。早期的这些计算机语言都是面向计算机专业人员，为了普及计算机语言，使计算机更为大众化，出现了入门级的 BASIC 语言，至今 BASIC 语言仍然是绝大多数软件开发人员接触到的第一门计算机语言，同时也最流行的计算机语言。

70 年代初，结构化程序设计的思想孵化出两种结构化程序设计语言，一种是 PASCAL 语言，另一种是 C 语言。这两种语言的语法结构基本上是等价的，它们都是通过函数和过程等语言特性来构成结构化程序设计的基础。但是很主要的区别在于 PASCAL 语言强调的是语言的可读性，因此 PASCAL 语言至今成为学习算法和数据结构等软件基础知识的教学语言；而 C 语言强调的是语言的简洁性以及高效性，因此 C 语言成为之后几十年中主流的软件开发语言，高效性使 C 语言的地位已相当于一种“高级汇编语言”。

当计算机技术逐渐在各种应用中得到普及时，使得软件开发效率提到日程上来。原有的

高级语言，如 BASIC、PASCAL 等结合可视化的界面编程技术、面向对象思想、数据库技术，产生了所谓的第四代语言，如 Visual Basic，Delphi 等。Visual Basic 的语言基础是 BASIC 语言，Delphi 的语言基础是 PASCAL，这两种语言都是软件开发人员所熟知的语言。

在 Web 技术的发展过程中，得到极大的普及的是 Java 语言。Java 是面向对象的网络语言，它的独特的网络特性包括：平台独立性、动态代码下载、为多媒体功能而设计的多线程、为通过 Internet 快速传送而设计的紧凑的代码格式。

Python 语言和其他很多语言一样，在一种特定的环境下成长发展起来的，Python 的创始人为 Guido van Rossum。1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。之所以选中 Python（大蟒蛇的意思）作为程序的名字，是因为他是一个 Monty 大蟒蛇飞行马戏团的爱好者。

ABC 是由 Guido 参加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 语言并没有成功，究其原因，Guido 认为是非开放造成的。Guido 决心在 Python 中避免这一错误（的确如此，Python 与其他的语言如 C，C++ 和 Java 结合的非常好）。同时，他还想实现在 ABC 中闪现过但未曾实现的东西。就这样，Python 在 Guido 手中诞生了。实际上，第一个实现是在 Mac 机上。可以说，Python 是从 ABC 发展起来，主要受到了 Modula-3 的影响。并且结合了 Unix Shell 和 C 的习惯。

1.2 Python 的优越性

计算机业经过了 PC 革命，又迎来了网络革命，下一个大的革命也许就是智能革命。但在智能时代到来之前，还是有很多有意义的改进工作可做，近年发展的一些半自动开发工具一定程度上降低了劳动强度，对程序语言的改进一直在进行。

然而对于编程来说，程序语言不是关键性的因素，但对工作效率却有不可低估的影响。经过测试证明一些 Script 语言（如 Python，Perl 等）和传统的语言（如 C，C++）相比，开发速度有 5 倍以上的差距。抽象地讲，Python 是一门解释性的、面向对象的、动态语义特征的高层语言。它的高层次的内置数据结构，以及动态类型和动态绑定，这一切使得它非常适合于快速应用开发，也适合于作为胶水语言连接已有的部件。Python 的简单而易于阅读的语法强调了可读性，因此降低了程序维护的费用。Python 支持模块和包，并鼓励程序模块化和代码重用。Python 的解释器和标准扩展库的源码和二进制格式在各个主要平台上都可以免费得到，而且可以免费分发。

通常，程序员爱上 Python 是因为它能增加生产力。由于没有编译过程，编辑—测试—调

试周期相当快。调试 Python 程序很简单：一个错误永远不会导致一个段错误。当解释器发现错误时，它就引发一个异常。当程序没有捕捉到异常，解释器就打印一个堆栈跟踪。一个源码级调试器允许我们检查局部和全局变量、计算表达式、设置断点、单步跟踪等等。调试器是用 Python 写的，这证明了 Python 的能力。另外，最快的调试程序的方法是增加几条打印语句：快捷的编辑—测试—调试周期使得这个简单的办法十分有效。

就像前面讲到的一样，一门计算机语言如果没有其特殊性和优越性是不可能得到迅速的发展和应用的。就简单而言 Python 语言有很多和其他流行语言一样的优越性。

● 容易学习

对于 Python 来说，能够在短期内迅速地得到普及和发展是在于它的易学性。对于一名掌握 C、Perl 或 shell 语言的程序员而言，花 3 至 5 个小时就能基本了解 Python 语言的基本特性。这和 Python 语言本身的简洁的特性是分不开的。

同时 Python 提供了一个交互式环境，这也是它容易学习的主要原因之一，就像以前的 Basic 和一些数据库语言环境（如 Dbase, SQL 等）一样，正如一个资深程序员所说，软件的复杂是因为千万件事堆在一起，交互式环境恰好可以让我们把程序分解开，我们可以试验每一个不清楚的语言成分，同时这也是一个很好的测试平台。

下面我们来看看同一个函数用 Python 和 Perl 来实现以证明其特性：

```
#—— Python version. ——
def pairwiseSum (list1, list2) :
    result = []
    for i in range (len (list1)) :
        result.append (list1[i] + list2[i])
    return result

#—— Perl version. ——
sub pairwiseSum {
    my ($arg1, $arg2) = @_ ;
    @list1 = @$arg1 ; @list2 = @$arg2 ;
    for ($i=0 ; $i < length (@list1) ; $i++) {
        push (@result, $list1[$i] + $list2[$i]) ;
    }
    return (@result) ;
}
```