



今日電子

高级编程“深入学习”丛书

深入学习：

Python Developer's
Handbook

python

程序开发

[美] André Lessa 著

张晓晖 张晓昕 王艳斌 等译

SAMS



電子工業出版社

Publishing House of Electronics Industry
URL: <http://www.phei.com.cn>

高级编程“深入学习”丛书

深入学习：Python 程序开发

Python Developer's Handbook

[美] André Lessa 著

张晓晖 张晓昕 王艳斌 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · Beijing

内 容 简 介

本书以翔实的内容和详尽信息资源并举的方式系统地介绍了自由、开源、跨平台的面向对象高层编程语言和网络服务器端脚本语言——Python。从Python编程语言的发展、特征、支持源、安装步骤、语言基础、主要模块扩展、异常处理以及面向对象思想的介绍入手,讨论了扩展和嵌入Python、对象接口和分布、数据库编程以及其他高级编程主题。探讨了使用Python程序实现基础网络思想、网络开发、脚本编程以及数据分析和处理等网络编程主题。同时介绍了Python图形界面设计中的GUI选项、主要GUI开发环境和编程工具。最后重点介绍了Python的Java实现——JPython。

本书内容丰富、叙述清晰、循序渐进,是国内第一本专门介绍Python的译著,由于其中包括丰富的资源Web链接,因而也可以作为信息指南使用。本书适用于各级别的计算机用户。

Authorized translation from the English language edition published by Sams Publishing, Copyright ©2001.

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 Publisher.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright ©2001.

本书中文简体版专有翻译出版权由Pearson教育集团所属的Sams Publishing授予电子工业出版社。其原文版权及中文翻译出版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

图书在版编目(CIP)数据

深入学习: Python 程序开发 / (美) 莱萨 (Lessa, A.) 著; 张晓晖等译. - 北京: 电子工业出版社, 2001.8
(高级编程“深入学习”丛书)

书名原文: Python Developer's Handbook

ISBN 7-5053-6942-3

I. 深… II. ①莱… ②张… III. PYTHON 语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第055745号

丛 书 名: 高级编程“深入学习”丛书

书 名: 深入学习: Python 程序开发

原 书 名: Python Developer's Handbook

原 著 者: [美] André Lessa

译 者: 张晓晖 张晓昕 王艳斌 等

责任编辑: 史 平

排版制作: 今日电子公司制作部

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787 × 1092 1/16 印张: 38.25 字数: 955 千字

版 次: 2001年8月第1版 2001年8月第1次印刷

书 号: ISBN 7-5053-6942-3

TP·3958

定 价: 58.00 元

版权贸易合同登记号 图字: 01-2000-3551

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话: 88211980 68279077

译者序

自 Guido van Rossum 创建 Python 以来,该语言历经十余年演变,已经发展成为当前最强大的面向对象高层编程语言和网络服务器端脚本语言之一。计算机业界迎来网络革命之际,面对 Perl(珍珠)语言的夺目之光,Python(巨蟒)更显示出其灵蛇般的上天入地之能——ILM、Yahoo!、Red Hat 以及 NASA 等都是它的用户。

享有诸如 C 等编译式语言和 Perl 等脚本语言之间的桥梁语言美誉的 Python,对我们似乎还很陌生,但在美国它以简洁、优美和容易学习等特性而逐渐走红。高层次的内置数据结构、动态类型和动态绑定使其非常适合于快速应用开发,也适合于作为粘合语言连接已有的部件;天然生就的跨平台运行能力使其可以运行于多种平台之上;支持跨语言开发的能力和开放式源代码使其应用程序可以自动扩展;精小的语言内核和简洁的语法使其容易阅读和学习。这种超强的适应性、超大的功能性、超凡的扩容性、超常的简易性使 Python 成为一种极酷的编程语言。如果要为既要求 GUI 实现、又要求采用复杂例程、同时强调 OOP 技术的项目选择最佳的编程语言,但又无力进行大规模投资,Python 就是解决方案。

本书由具有多年 IT 经验的 Python 专家 André dos Santos Lessa 撰写,内容翔实,将引导读者一步步走入 Python 世界。全书内容由七部分组成。由第 1 章到第 4 章构成的第一部分属于编程基础部分。主要介绍了 Python 编程语言的发展、优点、主要特征、支持源以及安装步骤;概览了语言基础和目前可用的主要模块扩展;用示例说明了如何处理异常情况以及如何避免出错并以直观全面的方式介绍了面向对象的编程方法。第二部分包括第 6 章到第 9 章,主要涵盖一些高级编程主题,其中包括扩展和嵌入 Python、对象接口和分布、数据库编程和其他一些主题。从第 10 章到第 13 章的第三部分论述 Python 的网络编程主题。介绍了如何使用 Python 程序实现基础网络思想、进行网络开发、脚本编程以及数据分析和处理等。第 14 章、第 15 章构成第四部分,引入了可用于设计 Python 图形界面的 CGI 选项并重点介绍了作为标准 Python GUI 的 Tkinter。第五部分即 Python 开发部分包括第 16 章和第 17 章,主要讨论了用于开发 Python 应用程序的主要 GUI 开发环境和编程工具。第 18 章独自构成第六部分,重点介绍了 Python 的 Java 实现——JPython。第七部分为附录,涵盖 Python/C API、在特定平台上运行 Python 等内容。

信息源丰富全面是本书的另一特点,作者编入了他所能获得的所有有关 Python 的信息,涵盖了 Python 语言最重要的方面。但值得注意的是,在原书编写和中文版翻译的几个月时间内,Python 推出了几个升级版本(目前是 Python 2.1),因而有些重要的方面没有收入。

本书主要由张晓晖、张晓昕和王艳斌组织翻译。参与本书翻译及文字录入等工作的人员还有:王新农、刘喜芳、王义天、陈晓东、韩庆军、李莉、兰朝利、刘建军、李铁胜、陈海燕、张君等。限于水平,错误之处在所难免,敬请读者和同行批评指正。

关于作者

我是 André dos Santos Lessa。决定从事 IT 业时我刚满 11 岁，那天也是我第一次看到一台真正的计算机——确切地说只是一台 TK85。此后的下一个生日，我得到了一台 TK90X。接下来是 MSX、386、486 等等。这样长期的计算机应用背景使许多大门向我敞开，我获得了计算机领域的学士和硕士学位。

现在我是具有 8 年多专业 IT 经验（从数据库管理到 Web 设计）的 IT 顾问。目前为 Emplifi 公司工作——倾注我最好的专业技能以支持 Deloitte Consulting 的项目。

由于非常喜欢采用新技术，特别是与 Web 相关技术的缘故，我为 Web 创建和设计了一些有趣的站点。www.lessaworld.com、www.bebemania.com.br 和 www.alugueaqui.com.br 便是拙作。

我最近参与的项目叫 iTraceYou.com，该项目既国际化又具有坚实基础，向我们所习惯的良好但老套的服务引入了新颖的安全性理念。

我出生在巴西的 Rio de Janeiro，1998 年移居美国以寻求具有挑战性的职业。在业余时间（每天只有几秒钟），我与妻子 Renata 共度美好时光。目前我们住在 Pittsburgh，Renata 怀着我们的第一个孩子，我们叫他 João Pedro。

如果需要，可以通过以下地址与我联系：webmaster@lessaworld.com。

请告诉我们你的想法

作为本书读者，你是我们最重要的批评者和评论者。我们非常重视你的意见并且希望知道我们取得的成绩、尚需要改进之处，还希望我们出版哪些领域的图书以及其他有益的见解。

读者可以直接给我们发 E-mail 或信函，告诉我们你喜欢或不喜欢本书的哪些方面，以及提供如何改进本书的建议。

但请注意，我无法对与本书主题相关的技术性问题提供帮助，并且因函件太多的缘故，请恕我不能一一回复。

E-mail: webdev_sams@mcp.com

Mail: Mark Taber

Associate Publisher

Sams Publishing

201 West 103rd Street

Indianapolis, IN 46290 USA

序 言

尚在孩提之时，我就做过这样一个梦：有一天蛇会主宰整个世界（实际上，我猜想企鹅也是这个梦的一部分……不过不要紧）。当时我并没有过分在意，因为我以为这个梦是由于看多了日本电视连续剧所致。后来在十几岁时，我参与了一个科学项目，其中需要花些时间研究蛇以便在一个展览会展示。对红尾蟒蛇和银环蛇进行了分析之后，我发现了一条3岁大、10英尺长、40多磅重的巨兽。我立刻意识到这条蛇是我多年前在梦中见到的那只。它的名字是Python（巨蟒），但那时我还搞不清这只爬行动物与世界主宰的关系。

15年前，我正努力在我那台老电视机上选择频道的时候，一个特别的节目引起了我的注意——一只巨大的动画脚在节目序幕中跳舞。节目开始之后，出现一帮拿鹦鹉和伐木工作取乐的有趣家伙。在看了多集这部情节剧和他们的五部影片后，我决定写一本关于他们的书。我注意到他们也叫Python，可能这就是答案。那个剧团会主宰整个世界，我希望让每个人都知这一点。最初我打算写一本关于这群演员以及他们最著名的短剧的书，但当我意识到编辑不会给我足够的时间让我写一本约25 030页的书时，我放弃了这个想法，尽管它可能是一本畅销读物。

尽管我梦想的事情无一成真，但它们至少具有一个共同之处——即名叫Python。Python也是一种脚本语言，其名称起源更多地与英国剧团有关而与爬行动物没有多少关系。本书将引导读者一步步走入Python世界。Python是一种引人入胜的编程语言，它可以帮助解决我们可能面对的几乎各类IT挑战。几乎所有与IT相关的任务，如数据库系统的处理或Web驱动应用程序的设计都可以采用Python加以解决。可能这就是我所做的梦的答案。

最近几个月以来，我一直致力于组织所能获得的所有有关Python的信息，并把它们编入本书。我无法断言已把所有细枝末节都写入了本书，但我敢肯定我涵盖了Python语言最重要的方面。请注意，在编写本书的5个月时间内，Python推出了几个升级版本，所以对于本书无法收编一些重要的方面我深表歉意。

本书共包括18章和一些附录，每一章论述Python语言的一个特定方面。在各章中，读者会发现许多关于如何使用Python以满足自己需要的提示。读者可能会同意这样的观点：以这种全面且最新的方式阐述Python语言的各个方面是不可能的。这就是我选择提供其他资源——Web链接（其上材料对读者学习有用）的原因所在。

那么，本书讨论的内容是什么呢？

简要的回答是：

本书以对Python语言的极全面评述为开始，接着论述构成Python发布包一部分的模块，之后讨论面向对象编程、网络、Web开发、图形界面和其他重要主题。最后一章论述JPython，即运行在Java系统上的Python版本。

详细回答如下：

- 第1章论述 Python 是什么，必须使用 Python 的原因，如何获得支持以及如何安装。
- 第2章对 Python 编程语言进行全面评述。本章结束之时，读者将了解如何创建 Python 应用程序。
- 第3章讨论目前可用的主要模块扩展以及使用它们的目的所在。本章重点在于帮助读者扩展有关 Python 库的知识，展示 Python 编程语言的已有资源。
- 第4章借助示例说明如何处理异常情况以及如何避免出错消息。
- 第5章以一种非常全面且直接的方式介绍面向对象编程方法。阅读完本章之后，读者就可以很容易地在程序中创建并使用对象和类。
- 第6章讨论扩展和嵌入 Python。读者将了解如何使用其他语言扩展 Python 方法以及如何从其他应用程序内部调用 Python 方法。
- 第7章解释对象接口和分布。该章解释对象分布以及如何从其他系统中使用它们。
- 第8章阐述 Python 中可用的所有数据库选项。本章为那些尚不了解数据库的读者解释数据库的工作原理和如何执行基本的 SQL 语句。
- 第9章提供有关使用和处理一些高级主题的有用信息，这些高级主题包括图像、声音、线程和科学 Python 模块。
- 第10章介绍基础网络思想，以及如何使用 Python 程序实现这些思想。
- 第11章介绍如何使用 Python 进行网络开发，同时也介绍一些著名的 Python 第三方 Web 应用程序。
- 第12章阐述如何使用 Python 进行脚本编程。
- 第13章说明如何使用 Python 进行数据分析和处理，例如 XML 分析和邮件处理。
- 第14章介绍 Python 中用于图形设计的可用 GUI 选项。
- 第15章介绍 Tkinter。对尚不了解它的读者而言，Tkinter 是标准 Python GUI。
- 第16章提供一些性能建议，并引导读者在保持风格的同时编写简洁的代码。
- 第17章介绍一些编程工具。读者将了解如何安心地进行所有的开发步骤，包括如何调试、编译和发布 Python 应用程序。
- 第18章论述使用 JPython 混用 Java 和 Python 多么容易。

既然大家已经了解到有大量有趣的内容有待学习，我建议大家接受如下提示：

阅读本书的最好方式是坐在舒服的沙滩椅上或躺在床上，并保持放松。如果由于某种原因，认为某个专题很乏味，则翻到自己感兴趣的章节即可。读者可以在稍后回到自己原来离开的位置。本书可以从头读起，也可以直接阅读介绍特定功能的章节，选择权利全在读者自己！

那么，还等什么？马上打开书，准备统治全世界吧。

目 录

第一部分 编程基础

第 1 章 导言	2
Python 简介	2
为什么使用 Python	3
主要技术特征	5
Python 发布包	8
安装和配置 Python	9
Python 与其他语言	12
补丁和故障列表	14
PSA 和 Python 社团	14
小结	16
第 2 章 Python 概述	18
语言评述	18
shell 环境	18
程序	21
内置数据类型	24
运算符	30
表达式	32
控制语句	40
数据结构	42
函数和过程	49
模块和包	54
输入和输出	58
文件处理	61
小结	63
代码范例	64
第 3 章 Python 库	70
Python 库	70
Python 服务	71
字符串组	80
杂项	84

一般操作系统	87
可选操作系统	94
调试器	97
剖视器	97
互联网协议和支持	97
互联网数据处理	99
限制执行	102
多媒体	102
加密	103
UNIX 专用模块	104
SGI IRIX 专用模块	107
Sun OS 专用模块	108
MS Windows 专用模块	108
Macintosh 专用模块	109
无说明模块	109
小结	112
第 4 章 异常处理	114
异常处理	114
标准异常（从其他模块获取帮助）.....	117
引发异常	119
捕获异常	123
try/finally	126
创建用户定义异常	126
标准异常层次体系	127
小结	130
代码范例	131
第 5 章 面向对象编程	134
面向对象编程	134
Python 面向对象编程介绍	134
Python 类和实例	136
方法处理	143
专用方法	144
继承	148
多态	151
封装	153
元类	155
小结	157
代码范例	158

第二部分 高级编程

第6章 扩展和嵌入 Python	166
扩展和嵌入 Python	166
Python/C API	167
扩展	167
编译和链接扩展模块	177
简单包装器接口生成器	181
其他包装器	183
嵌入	183
小结	187
代码范例	188
第7章 对象接口和分布	193
对象接口和分布	193
接口对象	193
COM 对象简介	194
在 Python 中实现 COM 对象	197
使用 Python 分布对象	212
小结	220
代码范例	221
第8章 数据库操作	227
操作数据库	227
平面数据库	227
DBM 数据库	229
对象串行化及持久性存储	235
ODBC 模块	240
ActiveX 数据对象	243
使用 SQL	244
Python DB API	250
小结	260
第9章 其他高级主题	261
其他高级主题	261
处理影像	261
操作声音	264
受限执行模式	268
科学计算	270
正则表达式	274

线程	279
小结	283
代码范例	284

第三部分 网络编程

第 10 章 网络基础	290
网络化	290
网络化概念	290
HTTP	299
访问 URL	305
FTP	308
SMTP/POP3/IMAP	309
Telnet 和 Gopher 新闻组	311
小结	313
第 11 章 Web 开发	315
Web 开发	315
为 Python/CGI 脚本配置 Web 服务器	315
第三方互联网应用程序	320
其他应用程序	324
站点管理工具	326
小结	329
第 12 章 脚本编程	332
Web 编程	332
CGI 概述	332
cgi 模块	334
创建、安装并运行脚本	335
Python Active Scripting	354
小结	359
第 13 章 数据处理	360
分析和处理数据	360
XML 处理	360
XML-RPC	374
XDR 数据交换格式	375
处理其他标记语言	379
MIME 分析和处理	387
普通转换函数	398

小结	401
----------	-----

第四部分 图形界面

第 14 章 Python 和 GUI	404
Python GUI 工具箱	404
Tkinter 模块	405
其他 GUI 模块概述	405
设计一个优秀界面	415
小结	416
第 15 章 Tkinter	418
Tcl/Tk 概述	418
Tkinter	419
几何管理	421
处理 Tkinter 事件	425
Tkinter 配件	428
设计应用程序	453
Python 元配件	457
Tkinter 资源	458
小结	459

第五部分 用 Python 开发

第 16 章 开发环境	462
构建 Python 应用程序	462
开发策略	462
集成开发环境	470
IDLE	470
Pythonwin	479
小结	486
第 17 章 开发工具	488
Python 程序的开发过程	488
编译 Python	488
编辑代码	492
Python 脚本	494
生成可执行的 Python 字节码	496
解释器	497

调试应用程序	499
剖视 Python	505
发布 Python 应用程序	513
小结	516

第六部分 Python 和 Java

第 18 章 JPython	520
欢迎使用 JPython	520
Java 集成	523
下载和安装 JPython	524
解释器	527
JPython 注册表	528
创建图形界面	530
嵌入	531
jpythone	532
运行 JPython 小应用程序	533
小结	534

第七部分 附 录

附录 A Python/C API	538
附录 B 在特定平台上运行 Python	589
附录 C 过渡到 Python 2.0	596

第一部分

编程基础

- 第 1 章 导言
- 第 2 章 Python 概述
- 第 3 章 Python 库
- 第 4 章 异常处理
- 第 5 章 面向对象编程

第1章 导 言

本章介绍Python作为一种优秀语言的原因所在、应用Python的理由，其主要特征、支持源以及安装步骤。

Python 简介

Python是日益引起业界及市场广泛关注的开放源代码语言。由于Python语言针对各类主要操作系统加以实现，因而不仅便于使用，而且具有可运行在多种平台之上的能力。自Guido van Rossum创建该语言以来，Python历经11年不断改进，已经发展成为当前可用的最强大的编程语言之一。

Python是一种优秀的原型语言。短短几分钟内，我们就可以开发使用其他语言可能需要花费几小时的原型。Python的核心引擎体现全面的面向对象思想。因此，利用Python编写面向对象的应用程序比利用诸如Java或C++等其他语言容易得多。

正如作者所言，Python是一项开放源工程。因此，它的确是免费的。其许可协定不会涉及版权转让或版权事宜。读者可以修改、改进、舍弃、销售Python，而且甚至可以为了商业用途免费分发。其版权仅向作者提供如下可能发生的法律问题的保护：因采用Python导致错误而引起的对作者的诉讼，或因某人假称拥有该语言所有权而引起的法律问题。

或许读者尚不了解Python，但使用该语言的公司已有许多。问题在于这些公司不想大肆宣扬，因为他们不想引起竞争者的注意。那么，世界上到底有谁在使用Python呢？运行Python应用程序的一些公司如下：ILM、Yahoo!、Red Hat以及NASA等。

注意：可以通过访问网站<http://www.python.org/news.html>了解有关Python的最新消息。

现在，许多开发者都致力于Python支持。这意味着当前有许多人正在为该语言测试并设计模块。如果花点时间访问Python的官方网站，可以得到正努力为Python提供一些新技术（如XML和图像处理）的几个开发组的名单。

Perl和Java早就拥有了一支忠诚的编程队伍；今天，Python也开始起步。

注意，如果拥有编程经验，Python是一种极易编码的语言。Guido声称自己的每次Python应用过程都充满乐趣。通过阅读本书，学习Python的过程也会是一段令人激动的经历。读者不久将体验这种感受。

本章首先对Python语言的主要特征进行简要评述。本书的其余章节对作者稍后提及的主题展开详细讨论。

Python 是什么

我们这样定义 Python:

Python 是一种解释性的、面向对象的高层语言以及强大的网络服务器端脚本语言。与其他脚本语言一样, Python 代码类似于伪代码。Python 的语法规则和精巧设计使得它便于由多位程序员组成的开发队伍阅读。该语言的语法并不丰富, 这的确很有帮助。语法简单的潜在意义是我们可以集中关注应用程序本身涉及的规则, 而不必把时间浪费在考虑应该使用哪一个命令之上。

Guido van Rossum 也曾指出: “丰富的语法带来的负担多于帮助。”

Python 还是一种交互式的、可移植的、易于学习、方便使用的严谨语言。此外, Python 提供动态语义和快速的原型开发能力。

Python 也是适合用做连接已有组件的粘合语言。它可以嵌入到使用其他语言 (C/C++、Java 等等) 编写的应用程序中; 此外, 也可以向 Python 添加新模块, 以扩展其核心词汇。

Python 是一种非常稳定的语言, 这不仅因为它在市场上已存在了 10 年之久, 还因为其解释器和所有标准库同时以源代码和二进制码格式存在。向所有人分发源代码是一项优秀的开发策略, 因为这样一来世界各地的开发者就可以协同工作。任何人都可以向由 Python 的创始人 Guido van Rossum 领导的官方开发队伍提交建议和补丁。

Guido 是脚本语言 ABC 的第二实现版本的合作创建者, 脚本语言 ABC 曾在 20 世纪 80 年代被少数人主要用于教学目的。

1989 年圣诞, 在荷兰阿姆斯特丹的 CWI, 因教育目的需要, Python 应运而生。Guido 当时是 CWI 的一名研究人员。起初, Python 只是 Guido 聊以度假而开发的自娱项目。后来, Python 变成 CWI 的 Amoeba Project 的一部分。其第一个公开版本于 1991 年 2 月发布。

很长时间以来, Python 的开发工作都在位于美国弗吉尼亚州 Reston 的 CNRI 进行。2000 年 6 月, Python 的开发队伍转移到了 PythonLabs (BeOpen Network 的一个成员机构), 该实验室由包括 Guido 在内的 Python 语言的前驱开发者们所维持。

2000 年 10 月 27 日, 由于对 Python 的未来发展存在分歧, 整个 PythonLabs 队伍离开了 BeOpen.com。PythonLabs 开发队伍现在为 Digital Creation 公司工作 (Zope 的制作商, <http://www.digicool.com>), Guido 也刚刚宣布了创建一个称为 Python 软件基金会的非赢利机构以拥有 Python 未来开发所有权的构想。

顺便提及, Python 因英国滑稽剧团 Monty Python 而得名。20 世纪 70 年代该团曾在 BBC 上演了一部名为 Monty Python's Flying Circus 的系列滑稽剧。Guido 是该剧的超级戏迷。

为什么使用 Python

请看下面的有趣场景:

假如读者在没有一支经过专业训练的编程者队伍的前提下, 为既要求 GUI 实现、又要求采用复杂例程、同时强调面向对象编程技术的项目选择最佳的编程语言。而且碰巧不幸的是, 读者无力进行大规模投资, 怎么办呢? 如果我是读者, 我会选择 Python。

但是如果读者是一位只想知道 Python 可以提供哪些强大优点的编程者, 或许会扪心自问:

“既然早已熟悉其他语言，为什么还需要 Python 呢？”

回答相当简单。尽管最初没有打算把 Python 发展成一种通用语言，但如果熟悉了该语言，则几乎任何事情都可很容易地进行。下面几段列举并解释了 Python 作为一种极“酷”的编程语言的原因所在，以及是什么使得 Python 比其他语言更加灵活。

可读性

Python 的语法既简洁又具可读性。Python 语法的组织方式使编程者有序可循。无论是专家还是初学者都很容易理解代码，而且都能极其快速地学会使用 Python。需要特别提及的是，与其他语言（如 Perl）相比，Python 包含极少“方言”。而且由于 Python 代码块结构采用缩进格式定义，因而由不正确缩进引起代码出错的可能性也大大减少。

获得支持的便利性

Python 社团一直向 Python 用户提供支持。众所周知，每个人都可免费获得 Python 代码。因此，世界各地成千上万开发者正致力于寻找代码故障并创建修补这些故障的补丁的工作。此外，许多人正在创建该语言的增强版本并发送它们以寻求认可。

学习速成

该语言非常容易学习，因为其源代码类似于伪代码，没有冗长奇怪的代码行。因此，少许训练就可速成。公司不需要花费很多时间让其程序员学习采用 Python 进行编程。着手学习 Python 之际，几乎可以立即利用它编写有用的代码。而且些许实践之后就可使工作效率迅速增加。

可以在友好的、解释性 Python 环境中设计高层、面向对象的程序代码。该特征尤其适合于小型任务。

编码快速

Python 以几种方式提供快速反馈。首先，编程者可以忽略其他语言要求执行的许多任务。因此，这既降低了程序维护成本，也减少了开发时间。如果需要，Python 可以进行快速的代码改编。因而读者可以在不用改动模块内编码作业规则的情况下，改变应用程序高层。

Python 发布包带有的交互式解释器为开发项目提供快速的开发策略。与要求几个确定阶段（如编译、测试以及运行）的传统编程语言和要求在执行环境之外编辑代码的其他脚本语言不同，Python 是一种即备即运行语言。每次使用 Python 的交互式解释器之时，只需执行拥有的代码即可。与 Perl 相比，该特征可提供的直接好处是可以交互式地测试并操作代码。

Python 提供自底向上的开发风格，这样我们就可以通过在解释器中导入并测试关键函数，然后编写调用这些函数的顶层代码创建应用程序。

解释器易于扩展，这使得我们可以把自己偏爱的 C 代码当作编译扩展模块嵌入。

重用性

Python 通过模块和程序包的实现而鼓励程序重用。目前已经开发了一个大型模块集合并把