大数据创新人才培养系列

Python 程序设计案例教程

PYTHON APPLICATIONS PROGRAMMING

◎ 徐光侠 常光辉 解绍词 黄德玲 主编

简练的语言**讲解理论**，丰富的案例**介绍应用**

采用当前最新的 **Python 3 版本**，准确、及时反映最新**成果**及**趋势**

详解**网络爬虫**、**数据处理**、**Web 开发**三大 Python 热门应用

中国工信出版集团 人民邮电出版社 POSTS & TELECOM PRESS

Python
程序设计案例教程

PYTHON APPLICATIONS
PROGRAMMING

徐光侠 常光辉 解绍词 黄德玲 主编

人民邮电出版社
北京

图书在版编目（CIP）数据

Python程序设计案例教程 / 徐光侠等主编. — 北京：人民邮电出版社，2017.6
（大数据创新人才培养系列）
ISBN 978-7-115-45213-9

Ⅰ. ①P… Ⅱ. ①徐… Ⅲ. ①软件工具－程序设计－教材 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第054757号

内 容 提 要

本书共 12 章，详细介绍了 Python 语言编程的方方面面。本书从 Python 的发展历程引入，介绍了 Python 语言的优点以及利用 Python 可以做些什么，随后引领读者循序渐进地学习了数据类型、组合数据类型、控制语句与函数、类和继承等基础内容。书中还探讨了模块的创建和使用、包的导入、文件的操作、调试及异常。为了进一步提升读者对 Python 程序设计的理解，本书在“程序开发进阶”这一章讲解了面向对象程序设计、函数式编程、多线程、thread 和 threading 模块。第 10 章正则表达式、第 11 章网络编程是对 Python 的两个应用领域的讲解。前 11 章，每一章都配有练习题，知识点讲解与课后练习相结合，方便读者巩固所学的知识和技巧。最后一章详细讲解了 Python 的 3 个热门应用—网络爬虫、数据处理、Web 开发，每个应用都提供了一个具体的小项目，以便读者跟随编者的思维进行实战练习。

本书是一本实用的学习指南，适合对计算机编程语言有一定基础的本科生、研究生以及大数据从业人员阅读。

◆ 主　　编　徐光侠　常光辉　解绍词　黄德玲
责任编辑　刘　博
责任印制　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：22.25　　2017 年 6 月第 1 版
字数：588 千字　　2017 年 6 月北京第 1 次印刷

定价：59.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前　言

Python 语言于 20 世纪 90 年代初由荷兰人 Guido van Rossum （吉多·范罗苏姆）首次公开发布，经过历次版本的修正，不断演化改进，目前已成为最受欢迎的程序设计语言之一。近年来，Python 多次登上诸如 TIOBE、PYP、StackOverFlow、GitHub、Indeed、Glassdoor 等各大编程语言社区排行榜。根据 TIOBE 最新排名，Python 与 Java、C、C++、C#一起成为全球最流行语言的前 5 位。

Python 语言之所以如此受欢迎，其主要原因是它拥有简洁的语法、良好的可读性以及功能的可扩展性。在各高校及行业应用层面，采用 Python 做教学、科研、应用开发的机构日益增多。在高校方面，一些国际知名大学采用 Python 语言来教授课程设计，典型的有麻省理工学院的计算机科学及编程导论、卡耐基梅隆大学的编程基础、美国加州大学伯克利分校的人工智能课程。在行业应用方面，Python 已经渗透到数据分析、互联网开发、工业智能化、游戏开发等重要的工业应用领域。另外，Python 也是一门易用性很强的程序设计语言，开发者利用它可以轻松实现一些较复杂的软件功能。究其原因是众多开源软件包都提供了 Python 的调用接口，例如著名的三维可视化库 VTK、计算机视觉库 OpenCV 等。而在科学计算的扩展库方面，NumPy、SciPy、matplotlib 专门为 Python 提供了强大的快速数组处理、数值运算以及绘图等功能。这些良好的第三方支持，推动了 Python 语言不断发展壮大。

基于 Python 语言的种种优点及其在 Web 开发、智能分析、机器人和游戏开发等领域中的深入应用，Python 教学和技术培训也已在高校和社会软件技术培训机构中广泛开展。因此，学校及行业对 Python 课程教材的需求与日俱增。但是目前理论阐述和工程实践紧密结合的教材为数不多，编者试图将这两方面结合起来，为读者提供一本有益的参考教材。

本书的编写原则是：①适应原则。Python 语言有自己独特的语法以及编程方式，与传统的 Java 语言、C 语言等或多或少有一些不同之处，编者试着从一个软件开发者的角度，在编程语言的大框架下，分析这些编程语言的细节差异，使读者能够很好地适应 Python 的学习。②科学原则。本教材既是知识产品的再生产、再创造，也是编者教学经验的总结和提高。其覆盖范围广、内容新，既有面的铺开，又有点的深化，举例符合题意，使读者学习起来事半功倍。③实用原则。本教材采用的是当前最新的 Python 3 版本，能够准确、及时地反映这门语言发展的最新成果及趋势，使读者能够很好地学到前沿的新技术。

本书从基础和实践两个层面引导读者学习 Python 这门学科，系统、全面地讨论了 Python 编程的思想和方法。第 1 章 ~ 第 3 章主要介绍了 Python 的基本知识以及理论基础。第 4 章 ~ 第 7 章详细介绍了 Python 编程的核心技术，着眼于控制语句与函数、模块和包、类和继承、文件和 I/O 的重点知识、使用场景以及注意事项的描述，每一个章节都搭配了详细的 Python 程序，让读者全面理解 Python 编程。

第 8 章是程序开发的进阶，着重介绍了抽象类、多继承、多线程等知识点，并针对每一个知识点给出了详细的例子。第 9 章具体介绍了软件开发语言中的重点——调试及异常，有编程语言常用的 try…except、finally 语句介绍和实例，也有特殊的 assert 语句和 with 语句介绍和实例。第 10 章重点介绍了正则表达式，并针对每一个知识点给出相关实例。第 11 章首先介绍了编程框架以及常用模块，然后结合实际应用给出实例。第 12 章给出了 3 个完整的例子——网络爬虫、数据处理和 Web 开发。

本书的编写特色在于：①理论+案例的编写风格。首先以简练的语言进行理论知识的讲解，然后配上丰富而实用的案例，在保证教材体系及比例科学的前提下，增加案例教学比重。②充分考虑学生学习之便利。考虑到当今大学生的实际情况，本教材所选的实例都贴近读者的理解水平；术语引入的节奏合理，不会让读者产生晦涩的感觉；其个别难点，都尽量讲解详尽与清晰。③实践性很强。本教材是在编者长期与 IT 企业合作进行软件研发积累的经验，以及企业内部进行专业培训的讲义的基础上，结合笔者多年的教学经验，研究国内外 Python 语言教材的优缺点，收集了相关的互联网资料，最后整理和改编而成，具有很强的实践性。

本书由徐光侠、常光辉、解绍词和黄德玲任主编。参加编写的人员及安排为：徐光侠编写第 1、8 和 12 章，常光辉编写第 2、3 和 4 章，解绍词编写第 9、10 和 11 章，黄德玲编写第 5、6 和 7 章。本书在编写过程中得到实验室主任刘宴兵教授的大力支持，特别感谢研究生团队对本书进行文字编辑和图片处理等付出的辛勤劳动。

由于编者水平有限，加之 Python 语言的发展日新月异，书中难免会有疏漏和不妥之处，敬请广大读者不吝赐教，编者 E-mail：xugx@cqupt.edu.cn。

编　者

2017 年 1 月于重庆邮电大学

目　录

第1章 入门

本章内容提要：

- Python 的发展历程
- 为什么使用 Python
- Python 可以做些什么
- Python 的优点
- Python 和其他语言的比较
- 项目开始

Python 是一种面向对象的、解释性的计算机程序设计语言，也是一种功能强大而完善的通用型语言，它已经有二十多年的发展历史，因此已经非常成熟和稳定。它具有脚本语言中最丰富和强大的类库，同时也借鉴了简单脚本和解释语言的易用性。它拥有非常简洁而清晰的语法特点，几乎可以在所有的操作系统中运行，能够支持绝大多数应用系统的构建。

进入 21 世纪以来，随着计算科学及大数据技术的发展，在行业应用和学术研究中采用 Python 进行科学计算的趋势越来越猛。而在众多解释型语言中，Python 最大的特点是拥有一个广泛而活跃的科学计算社区，从而为解决 Python 的各类问题提供了有力的保障。Python 目前被广泛地应用在 Web 开发、运维自动化、测试自动化、数据挖掘，甚至机器人、电脑动画等多个行业和领域。

1.1 Python 的发展历程

Guido van Rossum 是 Python 编程语言的创始人，1982 年他在阿姆斯特丹大学获得数学和计算机科学专业的硕士学位。Guido 在那个年代就已经学习并且使用过 C、Pascal、Fortran 等高级语言，而以上语言在设计时拥有一个共同的基本原则：使机器运行得更快。Guido 希望有一种语言，既能够全面调用计算机的功能接口，又可以轻松地编程。Guido 从 ABC 语言上看到了希望，并且也成为了 ABC 语言的设计者。但是在设计 ABC 语言时存在着一些问题，比如 ABC 语言可扩展性差、编译器体量很大、不能直接进行 IO、语法晦涩、学习困难等，致使 ABC 语言最终没能获得成功。

1989 年的圣诞节假期，Guido 为了打发时间，决定开发一个新的脚本解释程序，作为 ABC 语言的一种继承，于是开始编写 Python 语言的编译/解释器。他希望这个新的叫作 Python 的语言，能够成为一种功能全面、易学易用、扩展能力强的语言。于是，在 1991 年诞生了第一个 Python

编译器（同时也是解释器）。它用 C 语言来实现，并能够调用 C 语言的库文件。

Python 语法很多来自 C 语言，但又受到 ABC 语言的强烈影响。Python 非常注重可扩展性，它可以在多个层次上进行扩展。在底层上面可以引用 C 语言的库，在高层上面可以直接引入.py 文件。我们可以快速地使用 Python 写.py 文件作为扩展模块。

Python 的设计哲学是“优雅、明确、简单”。Python 开发者的哲学理念是“用一种方法，最好是只有一种方法来做一件事”。在设计 Python 语言时，如果面临多种选择，Python 开发者一般会拒绝花哨的语法，而选择明确的或者很少有歧义的语法。

在 Python 的开发过程中，社区起到了重要的作用。Python 自身的一些功能和大部分的标准库都来自于社区。Python 的开发者来自不同的领域，他们将不同领域的优点带给 Python。

到今天为止，Python 的框架已经确立。Python 语言以对象为核心组织代码，支持多种编程范式，采用动态类型，自动进行内存回收。Python 支持解释运行，并能调用 C 库进行拓展。Python 有强大的标准库。

1.2 为什么使用 Python

Python 作为一种高级程序设计语言，自从 20 世纪 90 年代初诞生以来，它的支持者就一直稳步增加。近年来，Python 逐渐被广泛应用于处理系统管理工作（比如它是很多 Linux 发行版的重要组成部分），它可以用于教授零基础的人们学习编程。2004 年以后，Python 的使用率呈线性增长。由表 1-1 可以看出，Python 已经成为最受欢迎的程序设计语言之一。

表 1-1　　各种编程语言历年的排名榜

编程语言	2016	2011	2006	2001	1996	1991	1986
Java	1	1	1	3	17	–	–
C	2	2	2	1	1	1	1
C++	3	3	3	2	2	2	5
C#	4	5	6	11	–	–	–
Python	5	6	7	25	23	–	–
PHP	6	4	4	8	-	–	–
JavaScript	7	9	8	7	21	–	–
Visual Basic.NET	8	29	–	–	–	–	–
Perl	9		5	4	3	–	–
Ruby	10	10	21	32	–	–	–
Ada	27	16	16	17	7	4	2
Lisp	28	12	12	14	6	7	3
Pascal	62	13	17	15	4	3	7

2011 年 1 月，Python 被 TIOBE 编程语言排行榜评为 2010 年度语言。最近几年 Python 变得越来越流行，在 2016 年 8 月 TIOBE 编程语言排行榜中，Python 已处在第五的位置，如表 1-2 所示。

2014 年斯坦福大学计算机博士 Philip Guo 为了调查 Python 的受欢迎程度，对美国高校计算机系中使用 Python 来教授入门课程的情况进行了研究。这项研究根据 2014 年美国大学 U.S.News 排行榜给出的排名，选取了 Top39 高校中的计算机系作为研究对象。该项研究结果显示 Top39 中

有 24 家大学在入门课程中教授 Python，可见 Python 早已成为美国大学计算机科学系入门课程中最受欢迎的编程语言。

表 1-2　　2016 年 8 月 TIOBE 编程语言排行 TOP 20 榜单

2016 年 8 月	2015 年 8 月	排名变化	编程语言	使用率	变动率
1	1	–	Java	19.010%	–0.26%
2	2	–	C	11.303%	–3.43%
3	3	–	C++	5.800%	–1.94%
4	4	–	C#	4.907%	+0.07%
5	5	–	Python	4.404%	+0.34%
6	7	↑	PHP	3.173%	+0.44%
7	9	↑	JavaScript	2.705%	+0.54%
8	8	–	Visual Basic.NET	2.518%	–0.19%
9	10	↑	Perl	2.511%	+0.39%
10	12	↑	Assembly language	2.364%	+0.60%
11	14	↑	Delphi/Object Pascal	2.278%	+0.87%
12	13	↑	Ruby	2.278%	+0.86%
13	11	↓	Visual Basic	2.046%	+0.26%
14	17	↑	Swift	1.983%	+0.80%
15	6	↓	Objective-C	1.884%	–1.31%
16	37	↑	Groovy	1.637%	+1.27%
17	20	↑	R	1.605%	+0.60%
18	15	↓	MATLAB	1.538%	0.31%
19	19	–	PL/SQL	1.349%	0.21%
20	95	↑	Go	1.270%	+1.19%

Packt Publishing 是世界上关于编程方面最大的出版商，它在 2016 年对 11 000 名访客进行了调查，调查内容包括开发者使用的编程语言、喜欢的框架、薪酬信息等几个方面。调查显示，Python 和 JavaScript 是当今最流行的编程语言，而 Java 紧随其后，排名第三。

Python 语言之所以这么受欢迎，主要有以下 5 个方面。

1. 软件质量

- Python 的语法简洁，注重可读性、一致性，从而保证了代码便于理解和维护。
- 在设计 Python 语言时，如果面临多种选择，Python 开发者一般会选择明确的或者很少有歧义的语法。
- Python 采用模块化设计和 OPP 在内的一些工具来提示程序的可重用性。

2. 开发效率

- Python 代码的大小往往只有 C++或 Java 代码的 1/5 ~ 1/3。
- 不需要传统编译/静态语言所必需的编译及连接等步骤，这样进一步提高了程序员的效率。

3. 程序的可移植性

绝大多数的 Python 程序不做任何改变就可以在所有主流计算机平台上运行。例如，在 Linux 和 Windows 之间移植 Python 代码，只需简单地在机器间复制代码即可。此外，Python 提供了多种可供选择的独立程序，包括用户图形界面、数据库接入、基于 Web 的系统等。甚至包括程序启动和文件夹处理等操作系统接口，Python 都尽可能地考虑了程序的可移植性。

4. 标准库的支持

Python 内置了很多预编译并且可移植的功能模块，这些功能模块叫作标准库（standard library）。标准库支持一系列应用级的编程任务，涵盖了从字符模式到网络脚本编程的匹配等方面。此外，Python 可通过自己开发的库或众多第三方的应用支持软件进行扩展。Python 的第三方支持工具包括网站开发、数值计算、串口读写、游戏开发等各个方面。

5. 组件集成

Python 脚本可通过灵活的集成机制轻松地与应用程序的其他部分进行通信，这种集成使 Python 成为产品定制和扩展的工具。如今，Python 代码可以调用 C 和 C++的库，可以被 C 和 C++的程序调用，可以与 Java 组件集成，可以与 COM 和.NET 等框架进行通信，并且可以通过 SOAP、XML-RPC 和 CORBA 等接口与网络进行交互。Python 绝不仅仅是一个独立的工具。

Python 已经成为最受欢迎的程序设计语言之一，它已经被广泛应于计算机游戏和生物信息等各种领域，我们更应该在学习与工作中使用 Python 语言。

1.3 Python 可以做些什么

Python 是一种面向对象的程序设计语言，它作为一种功能强大且通用的编程语言而广受好评，具有非常清晰的语法特点并且适用于多种操作系统。Python 在软件质量控制、提升开发效率、可移植性、组件集成、丰富库支持等各个方面均处于领先地位。许多大公司都在使用 Python 完成各种各样的任务，例如 YouTube、Instagram、Google、Yahoo 等，甚至美国航空航天局都大量地使用 Python。

1. 系统编程

Python 拥有操作系统服务的内置接口，使其成为可移植的操作系统维护工具（有时也称为 Shell 工具）。Python 程序可以搜索文件和目录树，可以运行其他程序，可以用进程或线程进行并行处理等。

Python 的标准库绑定了 POSIX（可移植操作系统接口）以及其他常规操作系统工具：环境变量、文件、套接字、管道、进程、多线程、正则表达式模式匹配、命令行参数、标准流接口、Shell 命令启动器、文件名扩展等。此外，很多 Python 的系统工具在设计过程中也考虑了其可移植性。例如，复制目录树的脚本不需要做任何修改就可以在几乎所有的 Python 平台上运行。

2. 科学与数字计算

Python 被广泛地应用于科学和数字计算中，例如 NumPy、SciPy、Biopython、SunPy 等 Python 扩展工具，经常被应用于生物信息学、物理、建筑、地理信息系统、图像可视化分析、生命科学等领域。

NumPy 数值编程扩展包括很多高级工具，例如矩阵对象、标准数学库的接口等。它将 Python 变成一个缜密严谨并简单易用的数值计算工具，这个工具通常可以用来替代已有的代码，而这些代码都是用 FORTRAN 或 C++等编译语言编写的。其他一些数值计算工具为 Python 提供了动画、3D 可视化、并行处理等功能的支持。例如，常用的 SciPy 和 ScientificPython 扩展，为使用科学编程工具以及 NumPy 代码提供了额外的库。

随着 NumPy、SciPy、Matplotlib、ETS 等众多程序库的开发，Python 越来越适合于做科学计算。与科学计算领域最流行的商业软件 MATLAB 相比，Python 是一门真正的通用程序设计语言，

比MATLAB所采用的脚本语言的应用范围更广泛，有更多程序库的支持，适用于Windows和Linux等多种平台，完全免费并且开放源码。虽然 MATLAB 中的某些高级功能目前还无法替代，但是对于基础性、前瞻性的科研工作和应用系统的开发，完全可以用 Python 来完成。

3. 数据库编程

Python 提供了所有主流的数据库接口，例如：Sybase、Oracle、Informix、ODBC、MySQL、PostgreSQL、SQLite 等。Python 定义了一种脚本，可以存取 SQL 数据库系统的可移植 API，这个 API 对于各种底层应用的数据库系统都是统一的。例如，因为厂商的接口需要实现为可移植的 API，所以一个自由软件 MySQL 系统的脚本不需做改变就可以工作在其他系统上（例如 Oracle），仅需要将底层的厂商接口替换掉就可以实现。

Python 标准的 pickle 模块提供了一个简单的对象持久化系统，它能够让程序轻松地将整个 Python 对象保存或恢复到文件和文件类的对象中。在网络上，同样可以找到名叫 ZODB 的第三方系统，它为 Python 脚本提供了完整的面向对象数据库系统，系统 SQLObject 可以将关系数据库映射至 Python 的类模块。并且从 Python 2.5 版本开始，SQLite 已经成为 Python 自带标准库的一部分了。

4. 游戏、多媒体、人工智能、XML、机器人等

Python 的应用领域很多，远比这里提到的要多得多。

- 可以利用 pygame 系统使用 Python 对图形和游戏进行编程。
- 利用 PIL、Piddle、ReportLab 等模块，可以处理图像、声音、视频、动画等，从而为你的程序添加亮丽的光彩。动态图表的生成、统计分析图表都可以通过 Python 来完成。
- 用 PyOpenGl 模块，可以非常迅速地编写出三维场景。
- 用 PyRo 工具包进行机器人控制编程。
- 用 XML 库、xmlrpclib 模块和其他一些第三方扩展进行 XML 解析。
- 使用神经网络仿真器和专业的系统 Shell 进行 AI 编程。
- 使用 NLTK 包进行自然语言分析，甚至可以使用 PySol 程序下棋娱乐。

5. 快速原型

对于 Python 程序来说，使用 Python 或 C 编写的组件看起来都是一样的。正因为如此，我们可以在一开始利用 Python 做系统原型，之后再将组件移植到 C 或 C++这样的编译语言上。当原型确定后就不需要重写，这是 Python 和其他的原型工具不同的地方。系统中执行效率不高的部分可以保持不变，从而使使用和维护变得轻松起来。

6. Internet 脚本

Python 提供了标准的 Internet 模块，无论是在服务器端还是在客户端，它都能使 Python 程序广泛地在多种网络任务中发挥作用。脚本可以通过套接字进行通信；从发给服务器端的 CGI 脚本的表单中提取信息；通过 FTP 传输文件；解析、生成和分析 XML 文件；发送、接收、编写和解析 Email；通过 URL 获取网页；从获取的网页中解析 HTML 和 XML 文件；通过 XML-RPC、SOAP 和 Telnet 通信等。Python 的库使这一切变得相当简单。

不仅如此，从网络上还可以获得很多使用 Python 进行 Internet 编程的第三方工具。例如，HTMLGen 可以从 Python 类的描述中生成 HTML 文件；mod_python 包可以使在 Apache 服务器上运行的 Python 程序更具效率，其支持 Python Server Page 这样的服务器端模板；Jython 系统提供了无缝的 Python/Java 集成，而且支持在客户端运行的服务器端 Applet。

此外，近些年出现了许多针对 Python 的 Web 开发工具包，例如 Django、TurboGears、Web2py、

Pylons、Zope 和 WebWare，它们使得 Python 能够快速构建功能完善和高质量的网站。例如国内的豆瓣、果壳网等；国外的 Google、Dropbox 等。很多这样的工具包包含了诸如对象关系映射器、模型/视图/控制器架构、服务器端脚本和模板，以及支持 Ajax 等功能，从而提供了完整的、企业级的 Web 开发解决方案。

7. 用户图形接口

Python 的简洁以及快速的开发周期十分适合开发 GUI 程序。Python 内置了 TKinter 的标准面向对象接口 Tk GUI API，使 Python 程序可以生成可移植的 GUI（图形化界面）。Python/Tkinter GUI 不做任何改变就可以运行在微软 Windows、X Windows（UNIX 和 Linux）以及 Mac OS（Classic 和 OS X 都支持）等平台上。同时，一个免费的扩展包 PMW，为 Tkinter 工具包增加了一些高级部件。此外，基于 C++平台的工具包 wxPython GUI API 可以使用 Python 构建可移植的 GUI。

诸如 PythonCard 和 Dabo 等一些高级工具包是构建在 wxPython 和 Tkinter 的基础 API 之上的。通过适当的库，你可以在 Python 中使用其他的 GUI 工具包。例如，通过 PyQt 使用 Qt、通过 PyGTK 使用 GTK、通过 PyWin32 使用 MFC、通过 IronPython 使用.NET，以及通过 Jython（Java 版本的 Python）使用 Swing 等。对于运行于浏览器中的应用或具有一些简单界面需求的应用，Jython 和 Python Web 框架以及服务器端 CGI 脚本都提供了其他的用户界面的选择。

8. 嵌入和扩展

Python 可以嵌入到其他应用程序中，也可以通过 C/C++编写扩展模块，从而可以提高程序的运行速度，还能完成只有通过 C/C++才能完成的工作。现在 Python 已经可以和 C#相结合，并且结合到 Visual Studio 里边，实现微软的.Net 思想。如果你会 C 语言，再学习 Python，这将是一个非常棒的一种选择。

如果你掌握了 Python，想在 Java 里应用它，你可以采用 Jython。Jython 是采用 Java 语言实现的 Python。这样，你只要按照 Python 的语法，就可以调用 Java 的各种类库，快速地编写出基于 Java 的程序，也就是通过 Jython 编写 Java 程序。这样就可以更为快速地实现 Java 的功能。Python 在面向对象方面和 Java 是相通的。

除了 C/C++和 Java，Python 目前还可以和 Delphi、VB 结合。

9. 组件集成

Python 可以通过 C/C++系统进行扩展，并能够嵌套 C/C++系统的特性，使其能够作为一种灵活的黏合语言，可以脚本化处理其他系统和组件的行为。例如，将一个 C 语言库集成到 Python 中，能够利用 Python 进行测试并调用库中的其他组件；将 Python 嵌入到产品中，在不需要重新编译整个产品或分发源代码的情况下，能够进行产品的单独定制。

为了在脚本中更好地使用，当 Python 连接编译好组件时，SWIG 和 SIP 这样的代码生成工具可以让这部分工作自动完成，并且 CPython 系统允许代码混合到 Python 和类似 C 的代码中。Python 还提供了一些更大的框架，如基于微软 WindowsCOM 的 Python，基于 Java 实现的 Jython，基于.NET 实现的 IronPython 和各种 CORBA 工具包。此外，Python 还提供了多种不同的脚本组件。例如，在 Windows 中，Python 脚本可利用框架对微软 Word 和 Excel 文件进行脚本处理。

10. 企业、政务及教学辅助的应用

目前，Python 已经成功地实现企业级应用，在全球已经有很多公司采用 Python 进行企业级软件的开发和应用。同时，通过 Python 技术，成功地实现了许多政务应用。

另外，Python 可以应用在教学活动中。用 Python 语言设计的教学辅助工具可以完成教学工作中重复性的工作，提高教学活动的工作效率。例如，可以利用 Python 语言编写文件操作题的自动

评卷程序、客观题的评分工具和辅助完成主观题的批量评分。

Python 提供了丰富的 API 和工具，以便程序员能够轻松地使用 C、C++、Cython 语言来编写扩展模块。在 Google 内部的很多项目使用 C++编写性能要求极高的部分，然后用 Python 调用相应的模块。目前使用 Python 的企业如下。

- Google 在其网络搜索系统中广泛应用了 Python，并且聘用了 Python 的创作者。
- YouTube 视频的分享服务大部分是由 Python 编写的。
- Intel、Cisco、Hewlett-Packard、Qualcomm 和 IBM 使用 Python 进行硬件测试。
- Industrial Light & Magic（工业光魔公司，是著名的电影特效制作公司）等公司使用 Python 制作动画电影。
- 在经济市场预测方面，JPMorgan Chase、UBS 等金融机构使用了 Python。
- NASA、Los Alamos（洛斯阿拉莫斯国家实验室）、Fermilab（费米实验室）等使用 Python 实现科学计算任务。
- IRobot 使用 Python 开发了商业机器人真空吸尘器。
- NSA（National Security Agency，美国国家安全局）在加密和智能分析中使用 Python。

1.4　Python 的优点

Python 是一种实际应用较为广泛的计算机语言，它具有很多优点，比如在设计上坚持清晰划一的风格,这使得 Python 成为一门易读、易维护，并且被大量用户所欢迎的语言。Python 语言的优点如下。

1. 免费

Python 的使用和分发是完全免费的。就像其他的开源软件一样，例如 Tcl、Perl、Linux 和 Apache。你可以从 Internet 上免费获得 Python 系统的源代码。复制 Python，将其嵌入你的系统或者随产品一起发布都没有任何限制。

“免费”并不代表“无支持”，恰恰相反，Python 的在线社区对用户需求的响应和商业软件一样快。而且，由于 Python 完全开放源代码，提高了开发者的实力，并产生了一个很大的专家团队。尽管学习研究或改变一个程序语言的实现并不是对每一个人来说都那么有趣，但是当你知道还有源代码和无尽的文档资源作为帮助的时候，这是多么的令人欣慰。你不需要去依赖商业厂商。

Python 的开发是由社区驱动的，是 Internet 大范围的协同合作努力的结果。这个团体包括 Python 的创始者 Guido van Rossum——Python 社区内公认的“终身的慈善独裁者”（Benevolent Dictator for Life，BDFL）。Python 语言的改变必须遵循一套规范的、有约束力的程序（称作 PEP 流程），并且需要经过规范的测试系统进行彻底检查。

2. 高级

每一代编程语言的出现都会将计算机科学提升到崭新的高度。比如 C 语言诞生了更多的像 C++、Java 这样的现代编译语言。我们并没有止步于此，而是有了更强大的、可以进行系统调用的解释型脚本语言，例如 Tcl、Perl 和 Python。

这些语言都有高级的数据结构，这样就减少了以前“框架”开发需要的时间。像 Python 中的列表（大小可变的数组）和字典（哈希表）就是内建于语言本身的。在核心语言中提供这些重要的构建单元，可以鼓励人们使用它们，缩短开发时间与代码量，生产出可读性更好的代码。

在 C 语言中，对于混杂数组（Python 中的列表）和哈希表（Pyhton 中的字典）还没有相应的标准库，所以它们经常被重复实现，并被复制到每个新项目中去。这个过程混乱而且容易产生错误。C++使用标准模板库改进了这种情况，但是其标准模板库是很难与 Python 内建的列表和字典的简洁和易读性相提并论的。

3. 可升级

大家常常将 Pyhton 与批处理或 UNIX 下的 Shell 相提并论。简单的 Shell 脚本可以用来处理简单的任务，就算它们可以在长度上（无限度的）增长，但是功能总会有所穷尽。Shell 脚本代码重用度很低，因此，你只能止步于小项目。实际上，即使一些小项目也可能导致脚本又臭又长。Python 却不是这样，你可以不断地在各个项目中完善你的代码，添加额外的新的或者现存的 Python 元素，也可以随时重用代码。Pyhton 提倡简洁的代码设计、高级的数据结构和模块化的组件，这些特点可以让你在提升项目的范围和规模的同时，确保灵活、一致性并缩短必要的调试时间。

“可升级”这个术语经常用于衡量硬件的负荷，通常指系统添加了新的硬件后带来的性能提升。我们试图用“可升级”来传达一种观念，这就是：Python 提供了基本的开发模块，你可以在它上面开发你的软件，而且当这些需要扩展和增长时，Python 的可插入性和模块化结构能使你的项目生机盎然并易于管理。

4. 易维护

源代码维护是软件开发生命周期的组成成分。只要不被其他软件取代或被放弃使用，你的软件通常会保持继续的再开发。Python 项目的成功很大程度上要归功于源代码的易于维护，当然这也要视代码长度和复杂度而定。Python 另一个激动人心的优势就是，当你在阅读自己六个月之前写的脚本程序的时候，不会把自己搞得一头雾水，也不需要借助参考手册才能读懂自己的软件。

5. 面向对象

从根本上讲，Python 是一种面向对象（OOP）的语言。它的类模块支持多态、操作符重载和多重继承等高级概念，语法简洁，十分易于使用。事实上，即使你不懂这些术语，仍会发现学习 Python 比学习其他面向对象语言要容易得多。

除了作为一种强大的代码构建和重用手段以外，Python 的 OOP 特性使它成为面向对象系统语言（如 C++和 Java）的理想脚本工具。例如，通过适当地粘接代码，Python 程序可以对 C++、Java 和 C#的类进行子类的定制。

OOP 是 Python 的一个选择而已，这一点非常重要。不必强迫自己立马成为一个面向对象高手，你同样可以继续深入学习。就像 C++一样，Python 既支持面向对象编程也支持面向过程编程的模式。

6. 可混合

Python 程序能够以多种方式轻易地与其他语言编写的组件“粘接”在一起。例如，Python 的 C 语言 API 可以帮助 Python 程序灵活地调用 C 程序。这意味着可以根据需要给 Python 程序添加功能，或者在其他环境系统中使用 Python。例如，将 Python 与 C 或者 C++写成的库文件混合起来，使 Python 成为一个前端语言和定制工具。就像之前我们所提到过的那样，这使 Python 成为一个很好的快速原型工具。出于开发速度的考虑，系统可以先使用 Python 实现，之后转移至 C，根据不同时期性能的需要逐步实现系统。

7. 可移植

Python 的标准实现是由可移植的 ANSI C 编写的（ANSI C 是由美国国家标准协会 ANSI 及国际标准化组织 ISO 推出的关于 C 语言的标准），可以在目前所有的主流平台上编译和运行。例如，

如今从 PDA（Personal Digital Assistant，掌上电脑）到超级计算机，到处可以见到 Python 在运行。Python 可以在下列平台上运行（这里只是部分列表）。

- Linux 和 UNIX 系统。
- 微软 Windows 和 DOS（所有版本）。
- Mac OS（包括 OS X 和 Classic）。
- BeOS、OS/2、VMS 和 QNX。
- 实时操作系统，例如 VxWorks。
- Cray 超级计算机和 IBM 大型机。
- 运行 Palm OS、PocketPC 和 Linux 的 PDA。
- 运行 Windows Mobile 和 Symbian OS 的移动电话。
- 游戏终端和 iPod 等。

除了语言解释器本身以外，Python 发行时自带的标准库和模块在实现上也都尽可能地考虑到了跨平台的移植性。此外，Python 程序自动编译成可移植的字节码，这些字节码在已安装兼容版本 Python 的平台上运行的结果都是相同的。

这些意味着 Python 程序的核心语言和标准库可以在 Linux、Windows 和其他带有 Python 解释器的平台无差别地运行。大多数 Python 外围接口都有平台相关的扩展（例如，COM 支持 Windows），但是核心语言和库在任何平台都一样。Python 还包含了一个叫作 Tkinter 的 Tk GUI 工具包，它可以使 Python 程序实现功能完整的，并且无需做任何修改即可在所有主流 GUI 平台运行的用户图形界面。

8. 扩展库

Python 标准库的确很大。它能够帮助你完成许多工作，包括正则表达式生成、文档生成、单元测试、线程、数据库、网页浏览器、CGI（公共网关接口）、FTP（文件传输协议）、电子邮件、XML（可扩展标记语言）、XML-RPC（远程方法调用）、HTML（超文本标记语言）、WAV（音频格式）文件、加密、GUI（图形用户界面）以及生成其他系统相关的代码。只要安装了 Python，所有这些都能做到。

除了标准库，还有各式各样的其他高质量库，你可以在 Python 包索引找到它们。

9. 解释性

一个用编译性语言（比如 C 或 C++）写的程序可以从源文件转换到你的计算机使用的语言（二进制代码，即 0 和 1）。这个过程通过编译器和不同的标记、选项完成。运行程序的时候，连接/转载器软件把你的程序从硬盘复制到内存中并且运行。而 Python 语言写的程序不需要编译成二进制代码，你可以直接从源代码运行程序。在计算机内部，Python 解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。这使得使用 Python 更加简单。也使得 Python 程序更加易于移植。

10. 功能强大

从特性的观点来看，Python 是一个混合体。它丰富的工具集使它介于传统的脚本语言（例如，Tcl、Scheme 和 Perl）和系统语言（例如，C、C++和 Java）之间。Python 具有脚本语言的简单性和易用性，并且具有在编译语言中才能找到的高级软件工程工具。不像其他脚本语言，这种结合使 Python 在长期大型的开发项目中十分有用。下面是一些 Python 工具箱中的工具简介。

（1）动态类型

Python 在运行过程中随时跟踪对象的种类，不需要代码中关于复杂的类型和大小的声明。事