

NITE 国家信息技术紧缺人才培养工程指定教材

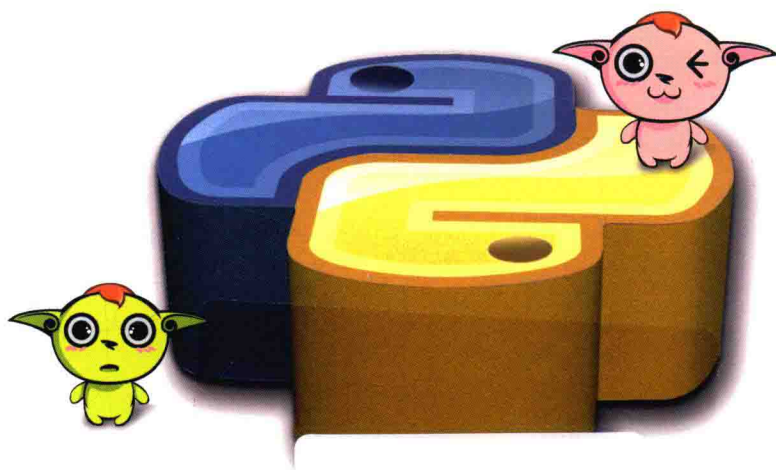


工业和信息化“十三五”人才培养规划教材

Python快速编程入门

黑马程序员 编著

有问题，就找黑马程序员问答精灵！



添加QQ或微信号208695827，获取教学答案、源码，抢“助学金红包”。

本书涵盖了Python开发的基础知识，旨在帮助广大Python读者快速掌握Python编程技能。

本书提供免费教学资源，包括12个精美PPT、1000道测试题、50余个教学案例、长达40小时的教学视频。

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

NITE 国家信息技术紧缺人才培养工程指定教材



工业和信息化“十三五”人才培养规划教材

Python快速编程入门

黑马程序员 编著

有问题，就找黑马程序员问答精灵！



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Python快速编程入门 / 黑马程序员编著. — 北京 :
人民邮电出版社, 2017.9
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-45836-0

I. ①P… II. ①黑… III. ①软件工具—程序设计—
高等学校—教材 IV. ①TP311.561

中国版本图书馆CIP数据核字(2017)第181427号

内 容 提 要

Python 是一种面向对象、解释性的高级程序语言, 它已经被应用在众多领域, 包括 Web 开发、操作系统管理、服务器运维的自动化脚本、科学计算、桌面软件、服务器软件(网络软件)、游戏等方面。

本书以 Windows 为平台, 系统全面地讲解了 Python 3 的基础知识, 其中, 第 1 章主要是带领读者认识 Python; 第 2 章主要针对 Python 的基础语法进行讲解; 第 3 章主要介绍 Python 中的常用语句; 第 4~5 章主要介绍了字符串、列表、元组、字典等类型; 第 6~7 章讲解了函数的基础和高级知识; 第 8 章讲解了 Python 中的文件操作; 第 9 章讲解了 Python 中异常的相关知识; 第 10 章讲解了 Python 中的模块; 第 11~12 章侧重讲解了面向对象编程思想; 第 13 章围绕着面向对象的编程思想, 开发了一个飞机大战的小游戏。

本书附有配套视频、源代码、习题、教学课件等资源, 为了帮助初学者更好地学习本书中的内容, 还提供了在线答疑, 希望得到更多读者的关注。

本书既可作为高等院校本、专科计算机相关专业的程序设计课程教材, 也可作为 Python 编程基础的培训教材, 是一本适合广大编程开发初学者的入门级教材。

-
- ◆ 编 著 黑马程序员
责任编辑 范博涛
责任印制 焦志炜
 - ◆ 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>
三河市中晟雅豪印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17 2017 年 9 月第 1 版
字数: 418 千字 2017 年 9 月河北第 1 次印刷
-

定价: 39.80 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

江苏传智播客教育科技股份有限公司(简称传智播客)是一家致力于培养高素质软件开发人才的科技公司,“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后,想从事IT行业但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格,包括了严格的技术测试、自学能力测试,还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员质量,并降低了企业的用人风险。

自“黑马程序员”成立以来,教学研发团队一直致力于打造精品课程资源,不断在产、学、研3个层面创新自己的执教理念与教学方针,并集中“黑马程序员”的优势力量,有针对性地出版了计算机系列教材50多种,制作了教学视频数十套,发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材,还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务:

1. 专业的辅助学习平台“博学谷”(<http://yx.boxuegu.com>), 专业老师在线为您解答。
2. 针对高校学生在学习过程中存在的压力等问题,我们还面向大学生量身打造了“播妞”。播妞不仅致力推行快乐学习,还会有定期的助学红包雨。同学快来添加播妞微信/QQ: 208695827。
3. 高校学生也可扫描下方二维码,加入播妞粉丝团,获取最新学习资源,与播妞一起快乐学习。



为IT教师提供的配套服务:

针对高校教学,“黑马程序员”为IT系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源,高校老师可关注码大牛老师微信/QQ: 2011168841,获取配套资源,也可以扫描下方二维码,加入专为IT教师打造的师资服务平台——“教学好助手”,了解“黑马程序员”最新教师教学辅助资源相关动态。



Python 是一种面向对象、解释性的高级程序语言，它已经被应用在众多领域，包括 Web 开发、操作系统管理、服务器运维的自动化脚本、科学计算、桌面软件、服务器软件（网络软件）、游戏等方面，未来它将被大规模应用在人工智能方面。

Python 的设计哲学是“优雅”“明确”“简单”，它的语法清楚、干净、易读、易维护，编程简单直接，更适合初学编程者，让初学者专注于编程逻辑，而不是困惑于晦涩的语法细节上。对于想快速就职的读者而言，学习 Python 无疑是一条捷径。

为什么要学习本书

随着人工智能时代的来临，Python 成为了人们学习编程的首选语言。本书站在零基础读者的角度，循序渐进地讲解了学习 Python 必备的基础知识，帮助读者建立编程思维和面向对象思想。

本书在讲解时，多采用生活中的例子来引入，同时利用生活插图，将生活与计算机语法结合起来进行讲解，最大程度地降低读者的阅读难度，同时也提升了读者的阅读兴趣。

作为开发人员，要想真正掌握一门计算机语言，离不开多动手练习，所以本书在绘声绘色讲解知识的同时，不断地增加案例，有针对某个知识点的案例，也有针对某章的案例，还有针对某个模块知识的案例，最大程度地帮助读者真正掌握 Python 这门语言的核心基础。

如何使用本书

本书基于 Python 3，系统全面地讲解了 Python 的基础知识，全书共 13 章，具体章节内容介绍如下所示。

第 1 章主要是带领大家认识 Python，包括 Python 的发展历程、Python 的特点、Python 的应用领域、Python 的版本、Python 的安装、IPython、集成开发环境的使用、Python 程序执行原理等。通过本章的学习，希望大家能够对 Python 有一个初步的认识，能够独立完成 Python 开发工具的安装和基本使用，为后面学习 Python 开发做好准备。

第 2 章主要针对 Python 的基础语法进行讲解，包括变量、数据类型、运算符、类型转换等。这些知识都是最基础的语法，也比较容易理解。希望读者在初学 Python 时，多动手写代码，加深印象，为后期的深入学习打好扎实功底。

第 3 章主要介绍的是 Python 的常用语句，包括判断语句、循环语句以及其他子句。其中，判断语句主要是 if 语句，循环语句主要是 for 语句和 while 语句。在 Python 开发中，这些语句的使用频率非常高，希望读者可以多加以理解，并熟练掌握它们的使用。

第 4 章首先对字符串进行了讲解，介绍了什么是字符串以及对特殊字符的转义；接着讲解了字符串的格式化输入/输出、字符串的访问；最后讲解了字符串的内建函数以及字符串运算符。通过本章的学习，希望读者能结合案例多加练习，熟练掌握字符串操作的相关函数。

第 5 章主要介绍了列表、元组和字典，其中，列表的内容主要讲解了循环遍历、增删改查、排序、嵌套；元组的内容主要讲解了增删查操作，这里需要强调的是，元组是无法进行修改的；

字典的内容主要讲解了元素的获取,包括键和值的获取,另外,还讲解了字典的增删改查、遍历。希望读者通过对本章的学习,能够清楚地知道这三种类型各自的特点,这样在后续开发过程中,可以选择合适的类型对数据进行操作。

第6章主要针对函数进行了讲解,包括函数的定义调用、函数的参数、函数的返回值、函数的嵌套、递归函数、匿名函数、日期时间函数和随机数函数。函数作为关联功能的代码段,可以很好地提高应用的模块性,希望读者能用好这些函数,并学会查询相关的函数手册。

第7章继续探讨函数的高级内容,包括闭包、装饰器,以及与序列相关的3个内置函数。通过对本章内容的学习,读者应该对函数有了更深入的理解,并能够熟练地应用到开发中。

第8章主要针对Python中的文件操作进行讲解,包括文件的打开关闭、文件的读写、文件的重命名、文件的删除等。通过本章的学习,希望读者可以认真掌握文件的相关操作,能够熟练使用相关方法来实现功能。

第9章围绕着Python的异常进行介绍,包括异常类,抛出和捕捉系统内置的异常,抛出和捕捉自定义异常,以及with和as环境安装器。通过对本章的学习,读者应该深入了解异常产生的原理,并知道如何在程序中运行它们。

第10章主要针对Python中的模块进行了讲解,包括模块的制作,包、模块的发布,模块的安装和使用。希望通过本章的学习,大家能够理解使用模块的好处,并在以后的工作中灵活的借助模块完成想要的功能。

第11章首先介绍了面向对象编程的基本知识,包括面向对象概述,类和对象及根据类创建对象,然后介绍了构造方法和析构方法的使用,然后又介绍了self和__str__方法的简单使用,最后开发了一个反恐游戏的案例。通过对本章内容的学习,读者对面向对象应该有了初步的了解,为后面面向对象的深入学习做好铺垫。

第12章继续介绍面向对象编程的知识,包括面向对象的三大特征(封装、继承和多态),然后介绍了类属性和实例属性,最后介绍了类方法和静态方法。通过对本章内容的学习,读者应该对面向对象有了深入的理解,为面向对象编程埋下了伏笔。

第13章围绕着面向对象的编程思想,开发飞机大战游戏的部分功能,包括搭建游戏界面,创建玩家飞机类和敌人飞机类,飞机发射子弹等,并且利用继承的技巧优化代码。通过对本章内容的学习,大家将知道如何在实际应用中,利用面向对象思想进行编程。

在学习过程中,读者一定要亲自实践本书中的案例代码。如果不能完全理解书中所讲知识,读者可以登录博学谷平台,通过平台中的教学视频进行深入学习。学习完一个知识点后,要及时在博学谷平台上进行测试,以巩固学习内容。

另外,如果读者在理解知识点的过程中遇到困难,建议不要纠结于某个地方,可以先往后学习。通常来讲,通过逐渐深入的学习,前面不懂和疑惑的知识点也就能理解了。在学习编程的过程中,一定要多动手实践,如果在实践的过程中遇到问题,建议多思考,理清思路,认真分析问题发生的原因,并在问题解决后总结出经验。

致谢

本书的编写和整理工作由传智播客教育科技有限公司完成,主要参与人员有吕春林、高美云、王晓娟、刘传梅、郑瑶瑶、邢文鹏、王铭东等。全体人员在这近一年的编写过程中付出了很多辛勤的汗水,在此一并表示衷心的感谢。

意见反馈

尽管我们付出了最大的努力，但书中难免会有不妥之处，欢迎各界专家和读者朋友们来信给予宝贵意见，我们将不胜感激。您在阅读本书时，如发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至：itcast_book@vip.sina.com。

黑马程序员

2017年6月5日于北京

目录

CONTENTS

专属于老师及学生的在线教育平台
yx.boxuegu.com

让 IT 教学更简单

教师获取教材配套资源

教案

授课资源

考试系统

在线题库

教学辅助
案例

添加微信/QQ

2011168841

让 IT 学习更有效

学生获取课后作业习题答案及配套源码

添加播妞微信/QQ

208695827

学习问答精灵: ask.boxuegu.com

更多学习视频: dvd.boxuegu.com



专属大学生的圈子

第 1 章 Python 概述 1

1.1 认识 Python 2

1.1.1 Python 的发展历程 2

1.1.2 Python 的特点 2

1.1.3 Python 的应用领域 3

1.1.4 Python 的版本 4

1.1.5 Python 2.x 和 Python 3.x 版本
区别 5

1.2 Python 安装 8

1.2.1 Windows 安装 Python 开发环境 9

1.2.2 Linux 安装 Python 开发环境 12

1.2.3 Mac OS 安装 Python 开发环境 13

1.3 IPython——交互式解释器 14

1.4 集成开发环境——PyCharm 15

1.4.1 PyCharm 的下载安装 15

1.4.2 PyCharm 的使用 18

1.5 Python 程序执行原理 22

1.6 本章小结 22

1.7 本章习题 23

第 2 章 Python 基础语法 25

2.1 基本语法 26

2.1.1 注释 26

2.1.2 行与缩进 26

2.1.3 语句换行 27

2.2 变量和数据类型 27

2.2.1 变量和赋值 27

2.2.2 变量的类型 28

2.3 标识符和关键字 30

2.3.1 标识符 30

2.3.2 关键字 31

2.4 简单数值类型.....	31	3.2.6 while 嵌套案例——九九乘法表.....	59
2.4.1 整型.....	31	3.3 Python 的其他语句.....	60
2.4.2 浮点型.....	32	3.3.1 break 语句.....	60
2.4.3 布尔类型.....	32	3.3.2 continue 语句.....	60
2.4.4 复数类型.....	33	3.3.3 pass 语句.....	61
2.4.5 数字类型转换.....	33	3.3.4 else 语句.....	61
2.5 运算符.....	34	3.4 本章小结.....	62
2.5.1 算术运算符.....	34	3.5 本章习题.....	62
2.5.2 赋值运算符.....	35		
2.5.3 复合赋值运算符.....	36	第 4 章 字符串.....	65
2.5.4 比较运算符.....	37	4.1 字符串介绍.....	66
2.5.5 逻辑运算符.....	39	4.1.1 什么是字符串.....	66
2.5.6 成员运算符.....	40	4.1.2 转义字符.....	66
2.6 位运算.....	41	4.2 字符串的输出和输入.....	68
2.6.1 按位左移.....	42	4.2.1 字符串输出.....	68
2.6.2 按位右移.....	42	4.2.2 字符串输入.....	69
2.6.3 按位与.....	42	4.3 访问字符串中的值.....	70
2.6.4 按位或.....	43	4.3.1 字符串的存储方式.....	70
2.6.5 按位异或.....	43	4.3.2 使用切片截取字符串.....	70
2.6.6 按位取反.....	43	4.4 字符串内建函数.....	70
2.7 运算符优先级.....	44	4.4.1 find.....	71
2.8 本章小结.....	45	4.4.2 index.....	72
2.9 本章习题.....	45	4.4.3 count.....	73
		4.4.4 replace.....	73
第 3 章 Python 常用语句.....	48	4.4.5 split.....	74
3.1 判断语句.....	49	4.4.6 capitalize.....	74
3.1.1 if 语句.....	49	4.4.7 title.....	75
3.1.2 if-else 语句.....	50	4.4.8 startswith.....	75
3.1.3 if-elif 语句.....	52	4.4.9 endswith.....	76
3.1.4 if 嵌套.....	53	4.4.10 upper.....	76
3.1.5 if 案例——猜拳游戏.....	54	4.4.11 ljust.....	77
3.2 循环语句.....	55	4.4.12 rjust.....	77
3.2.1 while 循环.....	56	4.4.13 center.....	78
3.2.2 for 循环.....	56	4.4.14 lstrip.....	78
3.2.3 while 循环案例——计算 1~100 偶数和.....	57	4.4.15 rstrip.....	79
3.2.4 while 嵌套.....	57	4.4.16 strip.....	79
3.2.5 while 嵌套案例——打印图形.....	58	4.5 字符串运算符.....	80
		4.6 本章小结.....	81

4.7 本章习题.....	81	6.5.4 有参数、有返回值的函数.....	113
第5章 列表、元组和字典.....	84	6.6 函数案例——名片管理器....	113
5.1 列表概述.....	85	6.7 函数的嵌套调用.....	116
5.2 列表的循环遍历.....	85	6.8 变量作用域.....	117
5.3 列表的常见操作.....	86	6.8.1 LEGB 原则.....	117
5.3.1 在列表中增加元素.....	86	6.8.2 全局变量和局部变量.....	118
5.3.2 在列表中查找元素.....	88	6.8.3 global 和 nonlocal 关键字.....	119
5.3.3 在列表中修改元素.....	89	6.9 递归函数和匿名函数.....	120
5.3.4 在列表中删除元素.....	89	6.9.1 递归函数.....	120
5.3.5 列表的排序操作.....	91	6.9.2 匿名函数.....	121
5.4 列表的嵌套.....	92	6.10 日期时间函数.....	123
5.4.1 什么是列表的嵌套.....	92	6.10.1 时间函数.....	123
5.4.2 列表嵌套案例——分配老师到 不同的办公室.....	92	6.10.2 日历函数.....	127
5.5 元组.....	93	6.11 随机数函数.....	128
5.6 字典.....	96	6.12 阶段案例——学生管理 系统.....	130
5.6.1 字典介绍.....	96	6.13 本章小结.....	134
5.6.2 字典的常见操作.....	96	6.14 本章习题.....	134
5.6.3 字典的遍历.....	100	第7章 高级函数.....	137
5.7 本章小结.....	101	7.1 闭包.....	138
5.8 本章习题.....	102	7.2 装饰器.....	139
第6章 Python 函数.....	105	7.2.1 什么是装饰器.....	139
6.1 什么是函数.....	106	7.2.2 多个装饰器.....	141
6.2 函数的定义和调用.....	106	7.2.3 装饰器对有参数函数进行装饰.....	143
6.2.1 定义函数.....	106	7.2.4 装饰器对带有返回值的函数进行 装饰.....	145
6.2.2 调用函数.....	107	7.2.5 带有参数的装饰器.....	146
6.3 函数的参数.....	107	7.3 常见 Python 内置函数.....	147
6.3.1 函数参数的传递.....	107	7.3.1 map 函数.....	147
6.3.2 默认参数.....	108	7.3.2 filter 函数.....	148
6.3.3 不定长参数.....	109	7.3.3 reduce 函数.....	149
6.4 函数的返回值.....	111	7.4 本章小结.....	150
6.5 函数的四种类型.....	111	7.5 本章习题.....	150
6.5.1 无参数、无返回值的函数.....	111	第8章 Python 文件操作.....	153
6.5.2 无参数、有返回值的函数.....	112	8.1 文件的打开和关闭.....	154
6.5.3 有参数、无返回值的函数.....	112		

8.1.1	文件的打开	154	9.8	本章习题	187
8.1.2	文件模式	154	第 10 章 Python 模块	189	
8.1.3	文件的关闭	155	10.1	模块的基本使用	190
8.2	文件的读写	155	10.2	模块的制作	192
8.2.1	写文件	155	10.3	Python 中的包	193
8.2.2	读文件	156	10.4	模块的发布	194
8.2.3	文件读写应用——制作文件的备份	158	10.5	模块的安装	196
8.2.4	文件的定位读写	159	10.6	本章小结	196
8.3	文件的重命名和删除	162	10.7	本章习题	197
8.3.1	文件的重命名	162	第 11 章 Python 面向对象编程 (上)	198	
8.3.2	文件的删除	162	11.1	面向对象编程概述	199
8.4	文件夹的相关操作	162	11.2	类和对象	200
8.5	文件操作应用——批量修改文件名	163	11.2.1	类和对象的关系	200
8.6	文件案例——学生管理系统 (文件版)	164	11.2.2	类的定义	201
8.7	本章小结	166	11.2.3	根据类创建对象	201
8.8	本章习题	166	11.3	构造方法和析构方法	202
第 9 章 异常	169		11.3.1	构造方法	203
9.1	异常简介	170	11.3.2	析构方法	204
9.2	异常类	170	11.4	self 的使用	205
9.3	异常处理	172	11.5	运算符重载	206
9.3.1	捕获简单异常	173	11.5.1	加法运算符重载	206
9.3.2	捕获多个异常	174	11.5.2	索引和分片重载	207
9.3.3	捕获异常的描述信息	175	11.5.3	定制对象的字符串形式	210
9.3.4	捕获所有的异常	176	11.6	阶段案例——反恐精英 CS	212
9.3.5	没有捕获到异常	178	11.7	本章小结	219
9.3.6	终止行为	179	11.8	本章习题	219
9.4	抛出异常	181	第 12 章 Python 面向对象编程 (下)	221	
9.4.1	raise 语句	181	12.1	封装 (隐藏数据和保护属性)	222
9.4.2	assert 语句	183	12.2	继承	223
9.5	自定义异常	184	12.2.1	单继承	223
9.6	预定义清理	185			
9.6.1	with 语句	185			
9.6.2	上下文管理器	186			
9.7	本章小结	187			

12.2.2 多继承	225	13.2.2 搭建游戏界面.....	241
12.2.3 重写父类方法与调用父类方法 ...	227	13.2.3 检查键盘的输入.....	242
12.3 多态	229	13.3 显示玩家飞机, 控制移动	
12.4 类属性和实例属性	231	方向	244
12.5 类方法和静态方法	232	13.4 玩家飞机发射子弹.....	246
12.6 本章小结	234	13.4.1 显示子弹	246
12.7 本章习题	234	13.4.2 子弹移动	247
第 13 章 飞机大战.....	236	13.5 敌人飞机.....	249
13.1 游戏介绍.....	237	13.5.1 显示敌人飞机.....	249
13.1.1 飞机大战游戏简介	237	13.5.2 控制敌人飞机移动.....	251
13.1.2 分析游戏显示的过程	238	13.5.3 敌人飞机发射子弹.....	252
13.2 游戏准备工作	238	13.6 优化程序代码	254
13.2.1 添加 pygame 模块.....	238	13.6.1 抽取子弹基类.....	254
		13.6.2 抽取飞机基类.....	255
		13.7 本章小结.....	258

Chapter 1

第 1 章 Python 概述

Python

学习目标

- 了解 Python 的发展历程。
- 了解 Python 的特点和应用领域。
- 可以独立完成 Python 的安装。
- 会简单使用 PyCharm 新建 Python 文件。
- 掌握 Python 程序的执行原理。



欢迎来到 Python 的世界。从本章开始，我们将从 Python 的发展、特点和版本等内容开始，带领大家认识 Python 这门语言，渐渐地，你会发现 Python 是一门非常有趣的语言，如果你学会了 Python，希望可以将它推荐给更多的人。现在，让我们一起开始 Python 的学习。

1.1 认识 Python

1.1.1 Python 的发展历程

Python 的创始人荷兰的 Guido van Rossum (见图 1-1)。1989 年，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 作为该编程语言的名字，是因为他是一个叫 Monty Python 喜剧团体的爱好者。

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 认为是其非开放造成的。Guido 决心在 Python 中避免这一错误，同时，他还想实现在 ABC 中闪过过但未曾实现的东西，就这样，Python 在 Guido 手中诞生了。可以说，Python 是从 ABC 发展起来的，主要是受到了 Modula-3 的影响，并且结合了 Unix shell 和 C 的习惯。

1991 年，Python 第一个公开发行人版本发行，它是一种面向对象的解释型计算机程序设计语言，它是使用 C 语言实现的，并且能够调用 C 语言的库文件。从一出生，Python 已经具有了类、函数、异常处理，包含列表和字典在内的核心数据类型，以及模块为基础的拓展系统。

最初的 Python 完全由 Guido 本人开发。Python 得到 Guido 同事的欢迎，他们迅速地反馈使用意见，并参与到 Python 的改进中。Guido 和一些同事构成 Python 的核心团队，他们将自己大部分的业余时间用于发掘 Python。随后，Python 拓展到研究所之外。Python 将许多机器层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思考。Python 程序员可以花更多的时间用于思考程序的逻辑，而不是具体的实现细节，这一特征吸引了广大的程序员，Python 开始流行。2011 年 1 月，Python 赢得 TIOBE 编程语言排行榜的年度语言。



图 1-1 Python 创始人

1.1.2 Python 的特点

Python 具有以下显著的特点。

(1) 简单易学

Python 是一种代表简单主义思想的语言。阅读一个良好的 Python 程序就感觉像是在读英语段落一样，尽管这个英语段落的语法要求非常严格。Python 最大的优点之一是具有伪代码的本质，它使我们在开发 Python 程序时，专注的是解决问题，而不是搞明白语言本身。

(2) 开源

Python 是 FLOSS (自由/开放源码软件) 之一。简单地说，你可以自由地发布这个软件的拷贝，阅读它的源代码，对它做改动，把它的一部分用于新的自由软件中。FLOSS 是基于一个团

体分享知识的概念，这是为什么 Python 如此优秀的原因之一——它是由一群希望看到一个更加优秀的 Python 的人创造并经常改进着的。

(3) 高级语言

Python 是高级语言。当使用 Python 语言编写程序时，无需再考虑诸如如何管理程序使用的内存一类的底层细节。

(4) 可移植性

由于 Python 的开源本质，它已经被移植在许多平台上。如果小心地避免使用依赖于系统的特性，那么所有 Python 程序无需修改就可以在下述任何平台上运行。这些平台包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acorn RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE，甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。

(5) 解释性

一个用编译性语言如 C 或 C++ 写的程序可以从源文件（即 C 或 C++ 语言）转换到一个计算机使用的语言。这个过程通过编译器和不同的标记、选项完成。当运行程序的时候，连接/转载器软件把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，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。事实上，由于不再担心如何编译程序，如何确保连接转载正确的库等，这一切使得使用 Python 变得更加简单。由于只需要把 Python 程序复制到另外一台计算机上，它就可以工作了，这也使得 Python 程序更加易于移植。

(6) 面向对象

Python 既支持面向过程编程，也支持面向对象编程。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如 C++ 和 Java 相比，Python 以一种非常强大又简单的方式实现面向对象编程。

(7) 可扩展性

如果需要一段关键代码运行的更快或者希望某些算法不公开，就可以把部分程序用 C 或 C++ 语言编写，然后在 Python 程序中使用它们。

(8) 丰富的库

Python 标准库确实很庞大。它可以帮助你处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI（图形用户界面）、Tk 和其他与系统有关的操作。记住，只要安装了 Python，所有这些功能都是可用的。这被称作 Python 的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如 wxPython、Twisted 和 Python 图像库等。

(9) 规范的代码

Python 采用强制缩进的方式使得代码具有极佳的可读性。

1.1.3 Python 的应用领域

Python 应用的领域包括如下场景。

(1) Web 应用开发

Python 经常被用于 Web 开发。例如,通过 `mod_wsgi` 模块,Apache 可以运行用 Python 编写的 Web 程序。Python 定义了 WSGI 标准应用接口来协调 HTTP 服务器与基于 Python 的 Web 程序之间的通信。一些 Web 框架,如 Django、TurboGears、web2py、Zope 等,可以让程序员轻松地开发和管理复杂的 Web 程序。

(2) 操作系统管理、服务器运维的自动化脚本

在很多操作系统里,Python 是标准的系统组件。大多数 Linux 发行版以及 NetBSD、OpenBSD 和 Mac OS X 都集成了 Python,可以在终端下直接运行 Python。有一些 Linux 发行版的安装器使用 Python 语言编写,如 Ubuntu 的 Ubiquity 安装器,Red Hat Linux 和 Fedora 的 Anaconda 安装器。Gentoo Linux 使用 Python 来编写它的 Portage (包管理系统)。Python 标准库包含了多个调用操作系统功能的库。通过 `pywin32` 这个第三方软件包,Python 能够访问 Windows 的 COM 服务及其他 Windows API。使用 IronPython,Python 能够直接调用 .Net Framework。一般说来,Python 编写的系统管理脚本在可读性、性能、代码重用度、扩展性几方面都优于普通的 shell 脚本。

(3) 科学计算

NumPy、SciPy、Matplotlib 可以让 Python 程序员编写科学计算程序。

(4) 桌面软件

PyQt、PySide、wxPython、PyGTK 是 Python 快速开发桌面应用程序的利器。

(5) 服务器软件 (网络软件)

Python 对于各种网络协议的支持很完善,所以经常被用于编写服务器软件、网络爬虫。第三方库 Twisted 支持异步网络编程和多数标准的网络协议 (包含客户端和服务端),并且提供了多种工具,被广泛用于编写高性能的服务器软件。

(6) 游戏

很多游戏使用 C++ 编写图形显示等高性能模块,而使用 Python 或者 Lua 编写游戏的逻辑、服务器。相较于 Python, Lua 的功能更简单,体积更小;而 Python 则支持更多的特性和数据类型。

(7) 构思实现,产品早期原型和迭代

YouTube、Google、Yahoo!、NASA 都在内部大量地使用 Python。

1.1.4 Python 的版本

Python 发展到现在,经历了多个版本,大家可以在 Python 官网查看。网址是 <https://www.python.org/downloads/>。截至目前,仍然保留的版本主要是基于 Python 2.x 和 Python 3.x。表 1-1 列举了 Python 的历史版本。

表 1-1 Python 历史版本

版本号	发布时间	拥有者	GPL 兼容
0.9.0~1.2	1991~1995	CWI	是
1.3~1.5.2	1995~1999	CNRI	是
1.6	2000	CNRI	否

续表

版本号	发布时间	拥有者	GPL 兼容
2.0	2000	BeOpen.com	否
1.6.1	2001	CNRI	否
2.1	2001	PSF	否
2.0.1	2001-06-22	PSF	是
2.2~2.7.11	2001~2015	PSF	是
2.7.12	2016-06	PSF	是
2.7.13	2016-12	PSF	是
3.x	2008~至今	PSF	是

在本书成稿时，Python 2 的最新版本是 Python 2.7.13，Python 3 的最新版本是 Python 3.6。关于 Python 2.x 和 Python 3.x 的区别将在下一个小节进行介绍，这里大家只需知道，Python 3.x 不再兼容现有的 2.x 程序，在实际使用时，大家要根据实际情况选择使用的版本。

1.1.5 Python 2.x 和 Python 3.x 版本区别

和 Python 2.x 版本相比，Python 3.x 版本在语句输出、编码、运算和异常等方面做出了一些调整，具体如下。

1. print 函数替代了 print 语句

在 Python 2.x 中，输出数据使用的是 print 语句。但是在 Python 3.x 中，print 语句没有了，取而代之的是 print 函数。示例代码如下：

Python 2.x

```
>>> print(3,4)
(3, 4)
```

Python 3.x

```
>>> print(3,4)
3 4
```

2. Python 3.x 默认使用 UTF-8 编码

Python 2.x 中有 ASCII Str()类型，unicode()是单独的，不是 byte 类型。在 Python 3.x 版本的源代码中，默认使用的是 UTF-8 编码，从而可以很好地支持中文或其他非英文字符。例如，输出一句中文，使用 Python 2.x 和 Python 3.x 的输出结果如下所示：

Python 2.x

```
>>> str = "我爱北京天安门"
>>> str
'\xe6\x88\x91\xe7\x88\xb1\xe5\x8c\x97\xe4\xba\xac\xe5\xa4\xa9\xe5\xae\x89\xe9\x97\xa8'
```

Python 3.x

```
>>> str = "我爱北京天安门"
>>> str
'我爱北京天安门'
```