

大学计算机基础教育规划教材

Python 程序设计基础

周元哲 编著

1+X

清华大学出版社



大学计算机基础教育规划教材

Python 程序设计基础

周元哲 编著

1+X

清华大学出版社
北京

内 容 简 介

本书共分 14 章,内容包括 Python 编程概述、数据类型和表达式、顺序与选择结构、循环结构、序列与字典、数据结构与算法、函数与模块、面向对象程序设计基础、文件、用户界面设计、绘图、数据库应用、网络编程和异常处理。

本书内容精练、由浅入深,注重学习的连续性和渐进性,章节之间的实例具有关联性。本书适合作为高等院校相关专业 Python 程序设计的教材或教学参考书,可以供计算机应用开发的各类技术人员参考,亦可作为全国计算机等级考试、软件技术资格与水平考试的培训资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Python 程序设计基础/周元哲编著. —北京:清华大学出版社,2015

大学计算机基础教育规划教材

ISBN 978-7-302-40526-9

I. ①P… II. ①周… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2015)第 136798 号

责任编辑:张 民

封面设计:傅瑞学

责任校对:梁 毅

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:保定市 中画美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:12.25 字 数:303 千字

版 次:2015 年 8 月第 1 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~2000

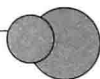
定 价:25.00 元

产品编号:064298-01

大学计算机基础教育规划教材

近 期 书 目

- 大学计算机基础(第4版) (“国家精品课程”、“高等教育国家级教学成果奖”配套教材、普通高等教育“十一五”国家级规划教材)
- 大学计算机基础实验指导书 (“国家精品课程”、“高等教育国家级教学成果奖”配套教材)
- 大学计算机应用基础(第2版) (“国家精品课程”、“高等教育国家级教学成果奖”配套教材、教育部普通高等教育精品教材、普通高等教育“十一五”国家级规划教材)
- 大学计算机应用基础实验指导 (“国家精品课程”、“高等教育国家级教学成果奖”配套教材)
- 大学计算机基础——计算思维初步
- 计算机程序设计基础——精讲多练 C/C++ 语言 (“国家精品课程”、“高等教育国家级教学成果奖”配套教材、教育部普通高等教育精品教材、普通高等教育“十一五”国家级规划教材)
- C/C++ 语言程序设计案例教程 (“国家精品课程”、“高等教育国家级教学成果奖”配套教材)
- C 程序设计 (“高等教育国家级教学成果奖”配套教材、“陕西省精品课程”主讲教材、陕西普通高校优秀教材一等奖)
- C++ 程序设计 (“高等教育国家级教学成果奖”配套教材)
- C# 程序设计 (“高等教育国家级教学成果奖”配套教材)
- Visual Basic 2005 程序设计 (“国家精品课程”、“高等教育国家级教学成果奖”配套教材、普通高等教育“十一五”国家级规划教材)
- Visual Basic 程序设计语言
- Java 语言程序设计基础(第2版) (普通高等教育“十一五”国家级规划教材)
- Java 语言应用开发基础 (普通高等教育“十一五”国家级规划教材)
- 微机原理及接口技术(第2版)
- 单片机及嵌入式系统(第2版)
- 数据库技术及应用——Access
- SQL Server 数据库应用教程(第2版) (普通高等教育“十一五”国家级规划教材)
- Visual FoxPro 8.0 程序设计
- 多媒体技术及应用 (“高等教育国家级教学成果奖”配套教材、普通高等教育“十一五”国家级规划教材)
- 多媒体文化基础 (北京市高等教育精品教材立项项目)
- 网络应用基础 (“高等教育国家级教学成果奖”配套教材)
- 计算机网络技术及应用(第2版)
- 计算机网络基本原理与 Internet 实践
- 可视化计算 (“高等教育国家级教学成果奖”配套教材)
- Web 应用程序设计基础(第2版)
- Web 标准网页设计与 ASP
- MATLAB 基础教程
- Visual Basic .NET 程序设计 (“高等教育国家级教学成果奖”配套教材)
- 微机原理·接口技术及应用
- Python 程序设计基础



Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言,在计算机程序设计语言的历史演变中具有划时代的意义。

Python 具有简单、现代、类型安全、性能优良等特点,是面向对象程序设计教学的主干语言之一。全书从教学实践角度对 Python 进行了全面的阐述,全书共分 14 章,内容包括 Python 编程概述、数据类型和表达式、顺序与选择结构、循环结构、序列与字典、数据结构与算法、函数与模块、面向对象程序设计基础、文件、用户界面设计、绘图、数据库应用、网络编程和异常处理。

在编写本书的整个过程中,作者结合多年从事计算机编程语言的教学经验,在教材内容的选取上力图精简,摒弃陈旧和繁杂的语法规定,只介绍 Python 语言的基本语法规定和面向对象的基本特征,培养学生更快地掌握编程思想和编程方法,提高学生的编程应用开发能力。

在本书的编写过程中,西安邮电大学通信与信息工程学院庞胜利、王江舟,计算机科学与技术学院李晓戈、刘伟、张庆生、孟伟君、王小银阅读部分手稿。华东师范大学江红老师等对本书提出了很多宝贵的意见。本书在写作过程中参阅了大量中英文的专著、教材、论文、报告及网上的资料,由于篇幅所限,未能一一列出,在此,一并表示敬意和衷心的感谢。

本书内容精练、文字简洁、结构合理、实训题目经典实用、综合性强,明确定位面向初、中级读者,由“入门”起步,侧重“提高”,特别适合作为高等院校相关专业 Python 程序设计的教材或教学参考书,也可以供从事计算机应用开发的各类技术人员参考,亦可作为全国计算机等级考试、软件技术资格与水平考试的培训资料。

本书的 Python 版本为 2.7.3,所有程序都在 Python 自带的 IDE 和 notepad++ 编辑器进行调试和运行。由于作者水平有限,时间紧迫,本书难免有疏漏之处,恳请广大读者批评指正。本书作者的电子信箱是 zhouyuanzhe@163.com。

作者
2015 年 3 月



第 1 章 Python 编程概述	1
1.1 计算机基础知识	1
1.1.1 计算机组成	1
1.1.2 软件和程序	2
1.1.3 程序设计语言	2
1.2 Python 的发展历史	3
1.3 Python 的特点	4
1.4 Python 的应用场合	5
1.5 Python 解释器	6
1.5.1 在 Ubuntu 下安装 Python	6
1.5.2 在 Windows 下安装 Python	7
1.6 Python 编辑器	8
1.6.1 IDLE	8
1.6.2 Notepad++	9
1.6.3 Ulipad	10
1.6.4 Eclipse+PyDev	12
1.6.5 Vim 和 emacs	13
1.7 Python 与其他语言关系	14
1.8 习题	15
第 2 章 数据类型和表达式	16
2.1 数据类型	16
2.1.1 数值	16
2.1.2 布尔型	17
2.1.3 字符串	17
2.1.4 空值	17
2.2 变量与常量	17
2.2.1 标识符	17
2.2.2 变量	18

2.2.3	常量	18
2.3	运算符	18
2.3.1	算术运算符	19
2.3.2	关系运算符	19
2.3.3	逻辑运算符	20
2.3.4	身份运算符	20
2.3.5	位运算符	20
2.4	表达式	21
2.4.1	表达式组成	21
2.4.2	优先级	22
2.4.3	结合性	22
2.5	系统函数	22
2.5.1	数学函数	22
2.5.2	转换函数	23
2.5.3	随机数函数	23
2.6	Python 字符	24
2.6.1	保留字符	24
2.6.2	转义字符	24
2.7	习题	25
第 3 章	顺序与选择结构	26
3.1	程序设计过程	26
3.1.1	三种基本逻辑结构	26
3.1.2	程序流程图	26
3.1.3	Python 程序设计流程	27
3.2	代码书写规则	28
3.2.1	缩进	28
3.2.2	逻辑行与物理行	28
3.2.3	空行	29
3.2.4	注释	29
3.3	顺序结构程序设计	30
3.3.1	赋值语句	30
3.3.2	输入与输出	31
3.3.3	顺序结构	32
3.4	选择结构程序设计	33
3.4.1	单分支	33
3.4.2	双分支	35
3.4.3	多分支	35

3.4.4 选择结构嵌套	36
3.5 程序设计方法与风格	39
3.5.1 语句构造方法	39
3.5.2 编程规范	40
3.6 习题	41
第4章 循环结构	42
4.1 循环	42
4.1.1 循环引入	42
4.1.2 循环概述	42
4.2 while 语句	42
4.2.1 确定次数循环	43
4.2.2 不确定次数循环	45
4.2.3 无限循环	49
4.3 for 语句	49
4.4 辅助语句	51
4.4.1 break 语句	51
4.4.2 continue 语句	52
4.4.3 else 语句	53
4.4.4 pass 语句	53
4.5 循环嵌套	54
4.6 习题	58
第5章 序列与字典	59
5.1 序列	59
5.1.1 序列概念	59
5.1.2 序列通用操作	59
5.2 列表	61
5.2.1 列表概念	61
5.2.2 列表操作	62
5.3 元组	65
5.3.1 元组概念	65
5.3.2 元组操作	66
5.4 字符串	67
5.4.1 字符串操作	67
5.4.2 字符串、列表、元组转换	68
5.5 字典	69
5.5.1 字典概念	69

5.5.2	字典操作	70
5.6	习题	73
第 6 章	数据结构与算法	74
6.1	数据结构	74
6.1.1	线性结构	74
6.1.2	非线性结构	75
6.1.3	序列与数据结构	76
6.2	查找和排序	76
6.2.1	查找	76
6.2.2	排序	78
6.3	算法	79
6.3.1	五个特性	79
6.3.2	三个层次	79
6.4	有特点的数	80
6.4.1	最小值和最大值	80
6.4.2	完全数	81
6.4.3	水仙花数	81
6.4.4	与素数有关的数	82
6.5	经典趣味题	84
6.5.1	鸡兔问题	84
6.5.2	百钱买百鸡	85
6.5.3	猴子吃桃	86
6.6	习题	86
第 7 章	函数与模块	88
7.1	函数	88
7.1.1	函数概念	88
7.1.2	函数声明和调用	88
7.1.3	实参和形参	90
7.1.4	引用传参	90
7.1.5	return 语句	91
7.1.6	函数是对象	91
7.2	参数类型	92
7.2.1	必备参数	92
7.2.2	默认参数	92
7.2.3	关键参数	93
7.2.4	可变长参数	93

7.3	两类特殊函数	94
7.3.1	lambda 函数	94
7.3.2	递归函数	95
7.4	变量作用域	96
7.4.1	局部变量	96
7.4.2	全局变量	97
7.5	模块	98
7.5.1	命名空间	98
7.5.2	模块定义与导入	98
7.6	习题	100
第 8 章	面向对象程序设计基础	101
8.1	面向对象概述	101
8.1.1	基本概念	101
8.1.2	与面向过程不同	102
8.1.3	面向对象三大特性	102
8.2	类和对象	103
8.3	类属性与实例属性	104
8.3.1	实例属性	104
8.3.2	类属性	105
8.4	方法	106
8.4.1	对象方法	107
8.4.2	类方法	107
8.4.3	静态方法	108
8.5	构造函数与析构函数	108
8.5.1	构造函数	108
8.5.2	析构函数	109
8.6	继承性	110
8.7	多态性	113
8.7.1	方法重载	113
8.7.2	运算符重载	113
8.8	习题	114
第 9 章	文件	115
9.1	文件概念	115
9.1.1	字符编码	115
9.1.2	文件分类	116
9.2	文件打开和关闭	116

9.3	文件操作	118
9.3.1	写操作	118
9.3.2	读操作	118
9.3.3	文件指针	119
9.4	存储器	120
9.5	与文件相关的模块	121
9.5.1	os 模块	121
9.5.2	os.path 模块	121
9.5.3	shutil 模块	122
9.6	习题	123
第 10 章	用户界面设计	124
10.1	概述	124
10.1.1	界面设计原则	124
10.1.2	常用 GUI 工具	124
10.2	Tkinter 编程	125
10.2.1	Tkinter 简介	125
10.2.2	实例讲解	126
10.3	wxPython 编程	127
10.3.1	wxPython 简介	127
10.3.2	wxPython 开发流程	128
10.3.3	Frame 创建与使用	128
10.4	控件	130
10.4.1	静态文本	130
10.4.2	输入文本	131
10.4.3	命令按钮	132
10.4.4	滑块、微调控制框	133
10.4.5	单选钮和复选框	135
10.4.6	列表框和组合框	136
10.4.7	菜单	138
10.4.8	工具栏和状态栏	140
10.5	对话框	141
10.5.1	警告对话框	142
10.5.2	单行文本对话框	143
10.5.3	列表选择对话框	143
10.6	习题	144

第 11 章 绘图	146
11.1 绘图概念	146
11.1.1 绘图简介	146
11.1.2 坐标系	146
11.2 海龟绘图	147
11.2.1 turtle 绘图方法	147
11.2.2 实例讲解	147
11.3 Canvas 绘图	150
11.3.1 Canvas 绘图方法	150
11.3.2 实例讲解	151
11.4 Numpy 与 Matplotlib	152
11.4.1 Numpy 简介	152
11.4.2 Matplotlib 简介	153
11.5 习题	155
第 12 章 数据库应用	156
12.1 数据库概念	156
12.1.1 数据库管理系统	156
12.1.2 关系型数据库	156
12.1.3 结构化查询语言	157
12.2 Python 数据库访问模块	159
12.2.1 通用数据库访问模块	159
12.2.2 专用数据库访问模块	159
12.3 Python 操作数据库	160
12.3.1 连接对象和游标	160
12.3.2 操作数据库过程	161
12.4 Python 与两个数据库	161
12.4.1 SQLite3	161
12.4.2 MySQL	162
12.5 习题	163
第 13 章 网络编程	164
13.1 网络基础知识	164
13.2 TCP/IP	164
13.2.1 TCP/IP 四层模型	165
13.2.2 IP 地址和端口号	165
13.3 Socket	166

13.3.1	TCP 连接	166
13.3.2	UDP 连接	168
13.4	电子邮件	169
13.4.1	SMTP 发送邮件	169
13.4.2	POP3 收取邮件	170
13.5	习题	171
第 14 章	异常处理	172
14.1	错误类型	172
14.1.1	语法错误	172
14.1.2	运行时错误	172
14.1.3	逻辑错误	173
14.2	捕获和处理异常	173
14.2.1	try...except...else 语句	173
14.2.2	try...finally 语句	175
14.3	两个特殊语句	176
14.3.1	raise 语句	176
14.3.2	with 语句	178
14.4	调试	178
14.4.1	调试策略	178
14.4.2	IDLE 调试器	179
14.5	习题	179
参考文献	180

第1章

Python编程概述



本章首先讲述了计算机相关的基础知识,如计算机组成、软件和程序,以及程序设计语言。然后介绍了 Python 的发展历史、特点以及应用场合,并就 Python 在 Ubuntu 和 Windows 环境中的安装和使用,以及 Python 编辑器作了说明,最后讲解了 Python 与其他语言之间的关系。

1.1 计算机基础知识

本节从计算机组成、软件和程序、程序设计语言等方面进行介绍。

1.1.1 计算机组成

计算机是由集成电路组成的电子设备,它由两部分组成:硬件系统和软件系统。所有可视的设备和外围设备都属于硬件系统。1944年,美籍匈牙利数学家冯·诺依曼提出了计算机基本结构和工作方式的设想,为计算机的诞生和发展提供了理论基础。时至今日,尽管计算机软硬件技术飞速发展,但计算机本身的体系结构并没有明显的突破,当今的计算机的体系结构仍属于冯·诺依曼架构。

冯·诺依曼提出的理论要点有以下两点:

(1) 计算机硬件设备由存储器、运算器、控制器、输入设备和输出设备五部分组成。其中,运算器和控制器组成中央处理器单元(CPU, Center Process Unit)。中央处理单元用于执行指令,如算术操作、从别的设备写入或读出数据。存储器分为内存和外存。CPU 从内存中读取所需要的数据,进行处理。内存中存储的数据是临时的,当程序退出或者计算机关机时,数据将会丢失。如果需要永久存储数据,需要用到外存,如硬盘、闪存等设备。键盘、鼠标等输入设备用于接受用户输入数据和指令,显示器通常作为输出设备。

(2) 存储程序思想——把计算过程描述为由许多命令按一定顺序组成的程序,然后把程序和数据一起输入计算机,计算机对已存入的程序和数据处理后,输出结果。图 1.1 为计算机体系结构图。

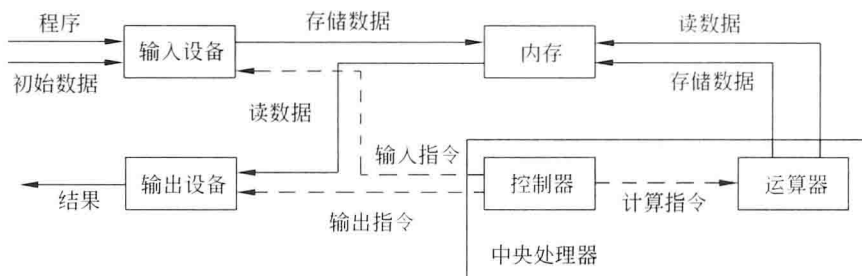


图 1.1 计算机体系结构图

1.1.2 软件和程序

相对于硬件系统而言,软件系统是由一些“不可视”的部分组成的,它是一系列按照特定顺序组织的计算机数据和指令的集合,如程序、数据、音频、视频等。实际上,不论指令还是数据都以二进制编码形式存在计算机中。在二进制系统中只有两个数(0和1),这是因为计算机硬件组成的物理器件具有两种稳定状态,如门电路的导通与截止、电压的高与低,恰好对应表示1和0两个符号。

计算机软件一般分为系统软件和应用软件两大类。系统软件为计算机用户提供最基本的功能,一般是操作系统和通用平台,如 UNIX, Windows, Linux 和 Android 等,帮助用户管理计算机的硬件。应用软件则是为了特定目的而设计的软件,不同的应用软件根据用户和所服务的领域提供不同的功能,如 Office, Photoshop 和游戏软件等。

一般认为,软件包括以下一些内容:

- (1) 运行时,能够提供所要求功能和性能的指令或计算机程序集合。
- (2) 程序能够满意地处理信息的数据结构。
- (3) 描述程序功能需求、程序如何操作和使用所要求的文档。

软件和程序是两个概念,对于初学者往往会混淆。其实,这发生在软件发展历史的第一阶段(20世纪50年代初期至60年代中期),由于软件的生产个体化,规模较小,功能单一,软件只有程序而无文档,形成了“软件等于程序”的错误观念。程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合,通过使用与自然语言具有相似的语法和语义的程序设计语言编写源代码,利用特定的工具将其翻译成CPU所能执行的指令,完成特定的目的。

1.1.3 程序设计语言

程序设计语言的发展经过了以下几个阶段。

1. 第一代程序设计语言

机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合,指令是由0和1组成的一串代码,通过线路变成电信号,让计算机执行各种不同的操作。机器语言具有直接执行特点。编程人员需要熟记所用计算机的全部由若干个0和1组成

的指令代码和代码的含义,机器语言具有难读、难编、难记和易出错的缺点。

2. 第二代程序设计语言

为了克服机器语言的缺点,人们用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码(如用 ADD 表示运算符号“+”的机器代码),采用助记符号编写程序,于是就产生了汇编语言。汇编语言比用机器语言的二进制代码编程要方便些,在一定程度上简化了编程过程。

汇编语言又称为符号语言,不能直接被机器识别,要使用一种程序将汇编语言翻译成机器语言。当汇编语言产生面向硬件操作控制信息的指令时,使用起来依旧烦琐,程序无结构,通用性也差。但是,使用汇编语言来编制系统软件和过程控制软件,其目标程序占用内存空间少,运行速度快,有着高级语言不可替代的用途。

3. 第三代程序设计语言

由于机器语言、汇编语言依赖于硬件体系,要求使用者必须对硬件结构及其工作原理都十分熟悉,因此人们又发明了与人类自然语言相接近且能为计算机所接受的规则明确、通用易学的计算机语言,其语法和结构具有类似文字的表现形式。1954年,第一个面向科学计算的高级计算机语言——FORTRAN 语言被正式推广使用,FORTRAN 语言是 Formula Translation 的缩写,意为“公式翻译”,是数值计算领域使用的主要语言。1972年,作为程序语言的里程碑,C 语言诞生了,它不但具有高级语言的特点,又具有汇编语言的特点,逐渐成为教学科研和软件开发的主要语言。

4. 第四代程序设计语言

面向对象程序设计语言、脚本语言、人工智能语言等通常被认为是第四代程序设计语言。SIMULA67 是第一个面向对象程序设计语言,特别是 1995 年 5 月由 Sun 公司推出的 Java 程序设计语言,可以撰写跨平台应用软件。第四代程序设计语言提供了功能强大的非过程化问题定义手段,用户只须告知系统做什么,而无须说明怎么做,因此可大大提高软件生产效率。Python 就属于第四代程序设计语言。

1.2 Python 的发展历史

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言,被列入 LAMP (Linux, Apache, MySQL 以及 Python/Perl/PHP)。Python 由 Guido van Rossum 于 1989 年年底发明,第一个公开发行人版发行于 1991 年。像 Perl 语言一样,Python 源代码同样遵循 GPL(General Public License,翻译为 GNU 通用公共许可证)协议。

Python 2.0 于 2000 年 10 月 16 日发布,实现垃圾回收,并支持 Unicode。Python 的 3.0 版本于 2008 年 12 月 3 日发布,常被称为 Python 3000,或简称 Py3k,相对于 Python 的早期版本,作了较大的升级。由于未考虑向下相容,导致早期 Python 版本设计的程序无法在 Python 3.0 上正常执行。为此,Python 2.6 和 2.7 作为一个过渡版本,基本使用

了 Python 2.x 的语法和库,同时考虑了向 Python 3.0 的迁移,允许使用部分 Python 3.0 的语法与函数。

本书的所有程序都在 Python 2.7.3 版本的开发环境中进行调试和运行。

1.3 Python 的特点

Python 是一种简单易学、功能强大的编程语言,它有高效率的高层数据结构,可以简单而有效地实现面向对象编程。Python 具有如下一些特点:

1. 简单易学

Python 作为代表简单主义思想的语言,其语法简洁而清晰,结构简单,可以快速上手,易于学习,Python 在学习过程中不用计较程序语言中在形式上的诸多细节和规则,便于专注程序本身的逻辑和算法,探究程序执行的过程。

2. 免费开源

Python 是 FLOSS(自由/开放源码软件)之一,可以自由地发布这个软件的副本,阅读它的源代码,对它做改动,并将它用于新的自由软件中。

3. 解释型语言

计算机并不能直接接收和执行用高级语言编写的源程序,源程序在输入计算机时,通过“翻译程序”翻译成机器语言形式的目标程序,计算机才能识别和执行。这种“翻译”通常有两种方式:一种是编译执行;另一种是解释执行。

编译执行是指源程序代码先由编译器编译成可执行的机器码,然后再执行;解释执行是指源代码程序被解释器直接读取执行。编译执行和解释执行各有优缺点,编译执行可一次性将高级语言源程序编译成二进制的可执行指令,通常执行效率高。而解释执行是由该语言(如 HTML)运行环境(如浏览器)读取一条该语言的源程序,然后转变成二进制指令交给计算机执行,通常可以灵活地跨平台。C、C++ 等采用编译执行方式,Python 作为解释型语言,与 Java 语言类似,不需要编译成二进制代码,通过解释器把源代码转换成称为字节码的中间形式,由虚拟机负责在不同的计算机运行,因此,Python 程序便于移植,可在众多平台运行,如 Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acom RISC OS, VxWorks, PlayStation, Sharp Zaurus 和 Windows CE 等。

4. 面向对象

Python 是完全面向对象的语言。函数、模块、数字、字符串都是对象,并且完全支持继承、重载、派生和多重继承。Python 语言编写程序无须考虑硬件和内存等底层细节。