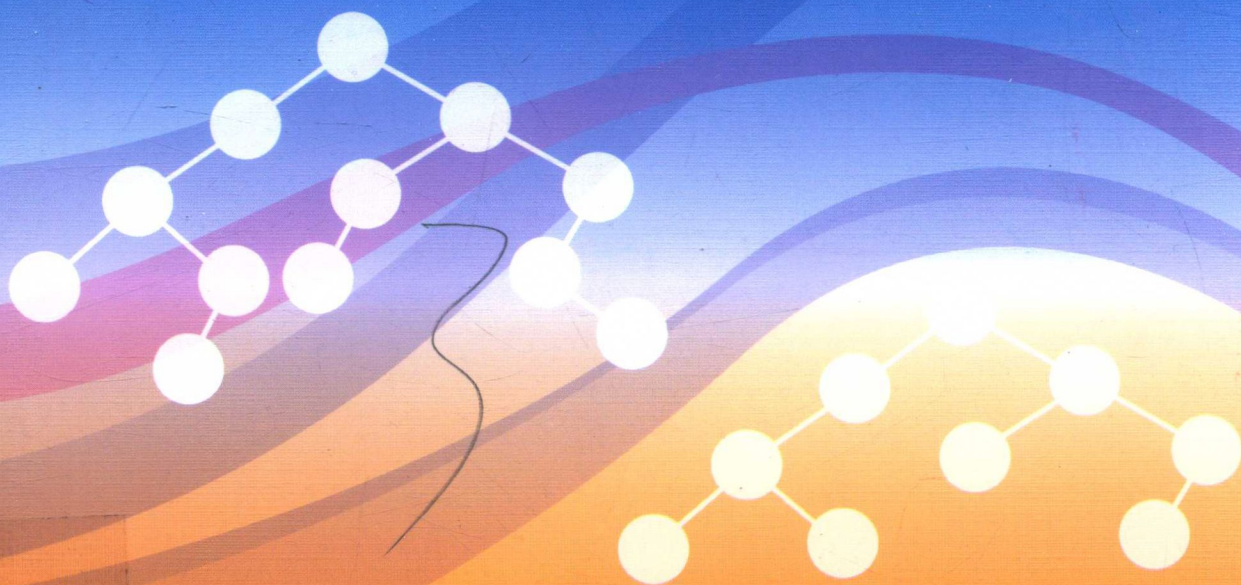


高等学校计算机专业规划教材

# Python程序设计教程



邱仲潘 刘燕文 王水德 编著

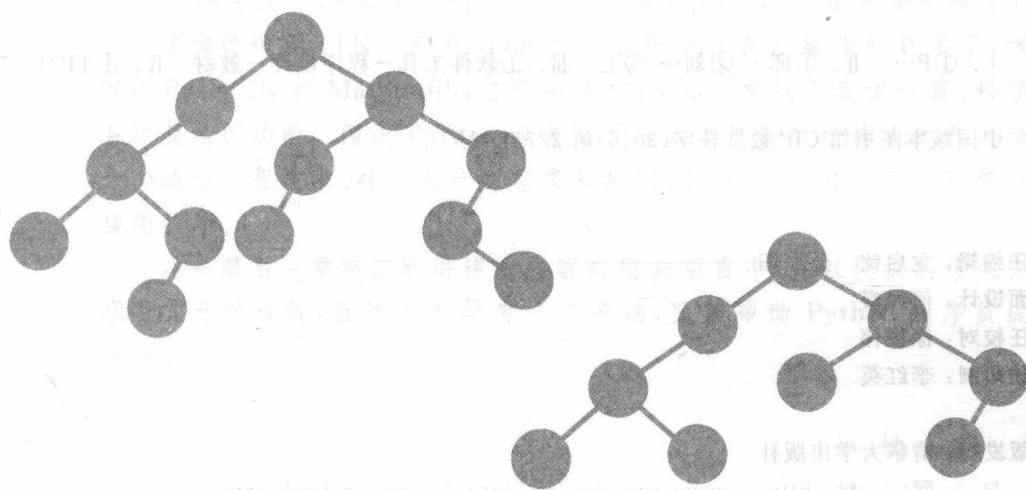
清华大学出版社



高等学校计算机专业规划教材

# Python程序设计教程

邱仲潘 刘燕文 王水德 编著



清华大学出版社  
北京

## 内 容 简 介

本书层次鲜明、结构严谨、内容翔实,由浅入深介绍 Python 程序设计的方方面面。最后一章将前面讲述的内容应用到项目中,并以模板的形式介绍项目的开发过程,理论联系实际项目,既适合初学者夯实基础,又能帮助 Python 程序员提升技能。

本书适合各类大中专学校学生作为教材,也可以作为程序员自学读物。

Python 程序设计教程

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 程序设计教程/邱仲潘,刘燕文,王水德编著. —北京:清华大学出版社,2016  
(高等学校计算机专业规划教材)  
ISBN 978-7-302-45098-6

I. ①P… II. ①邱… ②刘… ③王 III. ①软件工具—程序设计—教材 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2016)第 227117 号

责任编辑:龙启铭 薛 阳

封面设计:何凤霞

责任校对:徐俊伟

责任印制:李红英

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 字 数:462千字

版 次:2016年11月第1版 印 次:2016年11月第1次印刷

印 数:1~2000

定 价:39.50元

产品编号:066022-01

清华大学出版社

北京



Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月,它被 TIOBE 编程语言排行榜评为 2010 年度语言。自从 2004 年以后,Python 的使用率呈线性增长。

由于 Python 语言的简洁、易读以及可扩展性,在国外用 Python 做科学计算的研究机构日益增多,一些知名大学已经采用 Python 教授程序设计课程。例如卡耐基梅隆大学的编程基础和麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。众多开源的科学计算软件包都提供了 Python 的调用接口,例如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了,例如 NumPy、SciPy 和 Matplotlib,它们分别为 Python 提供了数值计算、科学计算以及绘图功能。因此 Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表,甚至开发科学计算应用程序。

本书最后一章将前面讲述的内容应用到项目中,并以模板的形式介绍项目的开发过程,既适合初学者夯实基础,又能帮助 Python 程序员提升技能。

编 者

2016 年 5 月



## 第 1 章 Python 语言简介 /1

1.1	什么是 Python 语言	1
1.2	Python 语言的发展历史	2
1.3	Python 语言的特点	3
1.4	Python 语言的应用	5
1.5	Python 的安装	5
1.6	第一个 Python 程序	12
1.7	本章小结	14
1.8	习题	14

## 第 2 章 数据类型、运算符以及表达式 /15

2.1	数据类型	15
2.1.1	变量	15
2.1.2	整型	17
2.1.3	浮点型	18
2.1.4	布尔型	19
2.2	输入与输出	20
2.2.1	print 语句	20
2.2.2	input 函数与 raw_input 函数	23
2.3	运算符	25
2.3.1	Python 语言运算符简介	25
2.3.2	算术运算符和算术表达式	25
2.3.3	关系运算符和关系表达式	26
2.3.4	逻辑运算符和逻辑表达式	28
2.3.5	位运算符	30
2.3.6	赋值运算符	35
2.3.7	其他运算符	35
2.3.8	运算符的优先级	37
2.4	本章小结	39
2.5	习题	39

**第 3 章 程序流程控制 /41**

3.1	算法概述	41
3.1.1	算法及其要素和特性	41
3.1.2	算法表示方法	42
3.1.3	自上而下求精法	48
3.2	控制结构	50
3.3	选择结构	51
3.3.1	if 选择结构	51
3.3.2	if/else 选择结构	52
3.3.3	if/elif/else 选择结构	55
3.4	循环结构	58
3.4.1	while 循环结构	58
3.4.2	for 循环结构	61
3.5	本章小结	64
3.6	习题	65

**第 4 章 序列: 字符串、列表和元组 /67**

4.1	概述	67
4.1.1	序列	67
4.1.2	序列类型操作符	67
4.1.3	序列类型内建函数	69
4.2	字符串	70
4.2.1	创建字符串	70
4.2.2	访问字符串	72
4.2.3	字符串操作符	73
4.2.4	常用字符串内建函数	75
4.3	列表	77
4.3.1	创建列表	78
4.3.2	访问列表	78
4.3.3	更新列表	80
4.3.4	列表操作符	83
4.3.5	常用列表内建函数	84
4.4	元组	89
4.4.1	创建元组	89
4.4.2	访问元组	90
4.4.3	元组操作符	91
4.4.4	常用元组内建函数	91



4.5	本章小结	93
4.6	习题	94
<b>第5章 映射和集合类型 /96</b>		
5.1	映射类型——字典	96
5.1.1	创建字典	96
5.1.2	访问字典	98
5.1.3	更新字典	99
5.1.4	字典操作符	104
5.1.5	常用字典内建函数	106
5.2	集合类型	109
5.2.1	创建集合	110
5.2.2	访问集合	111
5.2.3	更新集合(可变集合)	111
5.2.4	集合操作符	113
5.2.5	常用集合内建函数	115
5.3	本章小结	117
5.4	习题	118
<b>第6章 函数 /120</b>		
6.1	概述	120
6.2	函数的定义	121
6.2.1	无参函数的定义	121
6.2.2	有参函数的定义	122
6.2.3	空函数	122
6.3	函数参数和函数返回值	123
6.3.1	参数传递	123
6.3.2	位置参数	126
6.3.3	默认参数	127
6.3.4	关键字参数	128
6.3.5	可变长度参数	129
6.3.6	函数返回值	133
6.4	函数属性和内嵌函数	134
6.4.1	函数属性	134
6.4.2	内嵌函数	135
6.5	函数的嵌套调用	136
6.6	函数的递归调用	138
6.7	变量的作用域	143



6.7.1	局部变量	143
6.7.2	全局变量	144
6.8	本章小结	148
6.9	习题	149
<b>第 7 章 面向对象编程 /152</b>		
7.1	概述	152
7.1.1	什么是面向对象的程序设计	152
7.1.2	面向对象程序设计的特点	154
7.2	类的定义和对象的创建	155
7.2.1	类和对象的关系	155
7.2.2	类的定义	156
7.2.3	对象的创建	157
7.3	类、对象的属性和方法	159
7.3.1	属性	159
7.3.2	方法	163
7.4	组合	169
7.5	继承与派生	172
7.5.1	继承与派生的概念	172
7.5.2	派生类的定义	175
7.5.3	派生类的组成	177
7.5.4	多重继承	178
7.6	新式类的高级特性	184
7.6.1	__slots__ 类属性	184
7.6.2	__getattr__ () 特殊方法	185
7.6.3	描述符	186
7.7	本章小结	188
7.8	习题	189
<b>第 8 章 模块和包 /194</b>		
8.1	命名空间	194
8.1.1	命名空间的分类	194
8.1.2	命名空间的规则	195
8.1.3	命名空间的例子	195
8.2	模块	197
8.2.1	什么是模块	197
8.2.2	导入模块	197
8.2.3	模块导入的特性	202





8.2.4	模块内建函数	202
8.3	包	204
8.3.1	包的概述	204
8.3.2	包管理工具——pip	205
8.4	本章小结	207
8.5	习题	207
<b>第 9 章</b>	<b>异常</b>	<b>/210</b>
9.1	异常	210
9.1.1	什么是异常	210
9.1.2	标准异常类	211
9.2	异常处理	215
9.2.1	try...except 语句	215
9.2.2	try...except...else 语句	217
9.2.3	try...except...finally 语句	219
9.3	抛出异常和自定义异常	221
9.3.1	抛出异常	221
9.3.2	自定义异常	222
9.4	调试程序	223
9.4.1	使用 PythonWin 调试程序	224
9.4.2	使用 Eclipse for Python 调试程序	228
9.5	本章小结	236
9.6	习题	236
<b>第 10 章</b>	<b>文件</b>	<b>/240</b>
10.1	文件概述	240
10.2	文件的打开与关闭	241
10.2.1	文件的打开	241
10.2.2	文件的关闭	243
10.3	文件的读写	243
10.3.1	文件的读取	243
10.3.2	文件的写入	247
10.4	文件的定位	249
10.4.1	seek 函数	249
10.4.2	tell 函数	251
10.5	文件的备份和删除	252
10.5.1	文件的备份	252
10.5.2	文件的删除	254



10.6	文件夹的创建和删除	255
10.6.1	文件夹的创建	256
10.6.2	文件夹的删除	256
10.7	本章小结	257
10.8	习题	258
<b>第 11 章 项目开发实例 /262</b>		
11.1	Django 框架简介	262
11.2	MVC 模式	263
11.2.1	MVC 的概念	263
11.2.2	Django 的 MTV 模式	264
11.3	Django 安装	265
11.4	创建 Django 项目	267
11.4.1	创建开发项目	267
11.4.2	运行开发服务器	268
11.5	Django 项目的高级配置	270
11.5.1	创建项目应用	270
11.5.2	配置文件	270
11.5.3	设计数据模型	273
11.5.4	数据迁移	274
11.6	Template 模板	276
11.6.1	什么是模板	276
11.6.2	模板的继承	277
11.6.3	静态文件服务	279
11.7	学生信息管理	281
11.7.1	查询学生	281
11.7.2	添加学生	284
11.7.3	修改学生	290
11.7.4	删除学生	296
11.8	本章小结	298
11.9	习题	299

## 第 1 章

# Python 语言简介

### 本章学习目标

- 了解 Python 语言的发展历史
- 掌握 Python 语言的特点
- 了解 Python 的应用
- 熟练掌握 Python 的安装
- 掌握第一个 Python 程序

本章先向读者介绍一些有关 Python 的背景知识,什么是 Python 以及它的发展历史,然后介绍 Python 语言的特色、应用领域,在读者对 Python 语言有一定的了解之后,紧接着介绍 Python 的安装以及第一个 Python 程序,最后,本章末尾给出的练习题将使读者进一步巩固本章重要的知识点。

## 1.1 什么是 Python 语言

Python 是一种简单易学、面向对象、解释型的计算机程序设计语言,它既具备传统编译型程序设计语言的强大功能,又在某种程度上具备比较简单的脚本和解析型程序设计语言的易用性。其丰富的类库和简单易学的面向对象的编程特点深受初学者的喜爱,成为高等院校开设程序设计课程的主流编程语言之一,同时还因其具备可移植、可扩展等特性成为软件公司进行快速应用程序开发以及科研单位进行科学研究的主流编程语言。

Python 语言的语法简洁而清晰,具有丰富和强大的类库。它常被昵称为胶水语言,因为它能够很轻松地把用其他语言(尤其是 C/C++)编写的各种模块联结在一起。常见的一种应用情形是:使用 Python 快速生成程序的原型(有时甚至是程序的最终界面),然后对其中有特别要求的部分,用更合适的语言改写,比如 3D 游戏中的图形渲染模块,速度要求非常高,就可以用 C++ 重写。Python 是一种高层次的结合了解释性、编译性、互动性和面向对象的脚本语言,具有很强的可读性。

现在,全世界差不多有六百多种编程语言,但流行的编程语言也就二十来种。比如 C、Java、.NET、PHP 等。我们不能说什么语言比较好,这几种编程语言各有千秋。C 语言是可以用来编写操作系统的贴近硬件的语言,所以,C 语言适合开发那些追求运行速度、充分发挥硬件性能的程序,而 Python 是用来编写应用程序的高级编程语言。

当用一种语言开始作真正的软件开发时,我们除了要编写代码外,还需要很多基本的已经写好的现成的东西,来帮助我们加快开发进度。比如说,要编写一个电子邮件客户端,如果先从最底层开始编写网络协议相关的代码,那估计一年半载也开发不出来。高级编程语言通常都会提供一个比较完善的基础代码库,让用户能直接调用,比如,针对电子邮件协议的 SMTP 库,针对桌面环境的 GUI 库,在这些已有的代码库的基础上开发,一个电子邮件客户端几天就能开发出来。

Python 为我们提供了非常完善的基础代码库,覆盖了网络、文件、GUI、数据库、文本等主要内容。用 Python 开发,许多功能不必从零开始编写,可以直接使用现成的模块。

除了内置的基础库外,Python 还有大量的第三方库,也就是别人开发好供我们直接调用的模块。当然,如果我们开发的代码封装得很好,让其他开发者很方便地调用,也可以作为第三方库提供给别人调用。

许多大型网站就是用 Python 开发的,例如国外著名的视频分享平台 YouTube 和 Instagram,还有国内的豆瓣,同时还包括 Google、Yahoo 等大型公司,甚至 NASA (National Aeronautics and Space Administration,美国国家航空航天局)都大量地使用 Python。

## 1.2 Python 语言的发展历史

Python 语言的开发工作由 Guido van Rossum 开始于 1989 年末,接下来转移至荷兰的 CWI(Centrum voor Wiskunde en Informatica,国家数学和计算机科学研究院),并最终于 1991 年初公开发表。是什么促使他开发一门新的语言呢?一种程序设计语言的发明通常归结为两个原因:一是有一个资金充裕的大型研发项目作为支撑;二是因为缺乏某种软件工具而造成的困境,人们需要开发出一个新的工具来完成当时那些枯燥或者耗时的工作,而这些工作大部分又都是能够自动完成的。

Guido van Rossum 是 CWI 的一名研究员,他认识到高级教学语言 ABC(All Basic Code)因其语言不是开源的,不利于改进或扩展的重大缺点,因此,van Rossum 下定决心开发一种可扩展的高级编程语言,为其研究小组的 Amoeba 分布式操作系统执行管理任务。他从 ABC 汲取了大量的语法,并从系统编程语言 Modular-3 借鉴错误处理机制,开发出了一种能够通过类和编程接口进行扩展的高级编程语言,他将这种新语言命名为 Python(愿意为“大蟒蛇”)——来源于 BBC 当时正在热播的喜剧片 *Monty Python*。

自 1991 年初公开发行后,Python 开发者和用户社区逐渐壮大,Python 语言逐渐演变成一种成熟的、并获得良好支持的程序设计语言。Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月,Python 因在所有编程语言中占有最多市场份额,赢得 Tiobe 2010 年度语言大奖。自从 2004 年以后,Python 的使用率是呈线性增长的趋势。

由于 Python 语言的简洁、易读以及可扩展性,在国外用 Python 做科学计算的研究

机构日益增多,一些知名大学已经采用 Python 讲授程序设计课程。例如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。众多开源的科学计算软件包都提供了 Python 的调用接口,例如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了,例如 NumPy、SciPy 和 Matplotlib 这三个十分经典的科学计算扩展库,它们分别为 Python 提供了数值计算、科学计算以及绘图功能。因此 Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表,甚至开发科学计算应用程序。

### 1.3 Python 语言的特点

一种语言之所以能够存在和发展,并具有较强的生命力,总是有其不同于(或优于)其他语言的特点。Python 语言的主要特点如下。

#### 1. 免费开源

像 Java、PHP 等语言都是开放源代码的,这些语言都得到了广大编程人员的认可,并对其进行改进,使其越来越完善。而 Python 也是考虑到长远的发展,采取了向公众开放源代码的策略,这样就能使任何一个 Python 语言的爱好者都能够自由发布这个软件的拷贝、阅读源代码并把它运用到新的开源软件中,这就是为什么 Python 语言如此优秀的原因之一——它一直被一些更加优秀的人不断改进。

#### 2. 高级

程序设计语言每次更新换代都使我们进入更高一级。汇编语言可以直接对硬件进行操作,适合于对机器码很熟悉的开发人员。随后出现了 FORTRAN、C 和 Pascal 等语言,它们把计算任务带到一个更高的水平,并且开创了软件行业。这些语言又演化为如今的解析型系统设计语言 C++ 和 Java。再向上就是 Tcl、Perl 和 Python 等功能强大、能够进行系统调用的解析型脚本程序设计语言。这些语言都具有更高的数据结构,大大减少了项目中不可或缺的“程序框架”的开发时间。Python 语言还建立了更为有效的数据类型,比如列表(list,即可变数组)和字典(hash table,即哈希表)等,减少开发时间的同时也减少了代码长度。

#### 3. 易学

相对于其他编程语言,Python 语言关键字少、结构简单、语法清晰,具有很强的伪代码特性,方便阅读,这样就使得程序设计初学者可以在更短的时间内轻松上手。

#### 4. 易读

Python 与其他语言显著的差异是:它没有其他语言通常用来定义变量、定义代码块和进行模式匹配的命令式符号。通常这些符号包括:美元符号(\$)、分号(;)等。没有这些符号,Python 代码变得更加清晰和易于阅读。

#### 5. 面向对象

像 Java、C# 语言一样,Python 也支持面向对象编程,不同的是它还支持面向过程的

编程。面向对象的程序设计(Object Oriented Programming, OOP)为结构化和过程化程序设计语言增添了新的活力,面向对象编程技术的关键性观念是它将数据及对数据的操作行为组合在一起,作为一个相互依存、不可分割的整体——对象。而在面向过程的编程中,程序是由过程或可重用的函数模块来构建起来的。

## 6. 解释执行

Python 是一种解释型的语言,使用这种语言编写的程序,不需要编译成计算机可执行的二进制代码,而是直接从源代码运行程序。在计算机内部,像使用 C/C++ 等编译型语言编写的程序,必须通过编译器和不同的标记、选项把程序的源代码编译成计算机可执行的二进制语言。当运行程序时,连接/转载器软件再把程序从硬盘复制到内存中并且执行。而 Python 程序是通过 Python 解释器解释并执行的,Python 解释器把程序的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语言并执行,使得程序员无须关心程序如何编译、程序中用到的库如何加载等复杂问题。这样,使用 Python 将会更加简单,也更容易移植。

## 7. 灵活性

人们通常会把 Python 语言与批处理或 UNIX 系统下的 shell 脚本语言相提并论。简单的 shell 脚本可以用来处理简单的任务,shell 脚本的代码重用度很低,因此,它只能局限于小项目。而 Python 可以开发很大型的项目,用户可以不断地在各个项目中完善自己的代码,随时重用已写好的代码。Python 提倡简洁的代码设计、高级的数据结构和模块化的组件,这些特点可以让用户在扩大项目规模的同时,确保灵活性和一致性,并缩短必要的调试时间。

## 8. 可扩展性

Python 的可扩展性使得程序员能够灵活地附加程序,缩短开发周期,因为 Python 是基于 C 语言开发的,所以用 C/C++ 来编写 Python 的扩展功能。发展到现在,Python 也有基于 Java 实现的 Jython,从而使得 Python 可以在更多的语言中使用。

## 9. 嵌入性

Python 的嵌入性是指它可以作为一种成熟的脚本语言,并且以一种很方便的方式嵌入到其他的程序中,比如 C/C++ 中。

## 10. 可移植性

Python 具有强大的可移植性,只需要把 Python 程序拷贝到另一台计算机上就可以很方便地移植到各种主流的系统平台中,这是因为 Python 是用 C 写的,又由于 C 的可移植性,使得 Python 可以运行在任何带有 ANSI C 编译器的平台上。尽管有一些针对不同平台开发的特有模块,但是在任何一个平台上用 Python 开发的通用软件都可以稍作修改或者原封不动地在其他平台上运行。这种可移植性既适用于不同的架构,也适用于不同的操作系统。

## 11. 易维护

源代码维护是软件开发生命周期的组成部分。Python 项目的成功很大程度上要归功于其源代码的易于维护,同时也因为 Python 本身就是易于学习和阅读的,从而使得用 Python 语言开发的项目具有易维护的特点。

## 12. 丰富的类库

Python 是世界上具有标准库最大的编程语言。基于庞大的标准库,我们可以编写程序来处理各种工作,包括正则表达式、文档生成、单元测试、线程等功能。

## 13. 内存管理器

在程序开发过程中,我们会遇到像使用 C/C++ 时要考虑的程序的内存管理问题。即使开发的是很小的程序,应用程序的修改和管理也需要程序员额外负责,这就需要开发者付出更多的精力。而在 Python 的程序开发过程中,Python 解析器承担了程序的内存管理工作,使得程序员从内存事务处理中解脱出来,致力于程序功能的实现,从而减少错误,缩短开发周期。

# 1.4 Python 语言的应用

由于 Python 语言具有简单易学、可扩展、可移植等优点,自 2006 年以来,Python 已成为继 C++、Java 之后的第三种编程语言,更多地被应用到著名的搜索引擎,如 Google,还有应用到曾经称霸智能手机市场的 Nokia 所采用的 Symbian 操作系统上,可见 Python 的应用领域非常广泛。表 1-1 介绍了 Python 语言的应用领域。

表 1-1 Python 语言的应用领域

应用领域	详细描述
系统编程	提供 API 编程接口,能够方便地进行系统维护和管理,是很多系统管理员理想的编程工具,是 Linux 系统下的标志性语言之一
图形处理	含有庞大的对诸如 PIL、Tkinter 等图形类库的支持,能够方便地进行图形处理
数字处理	NumPy 扩展提供了大量与许多标准数学库对应的接口,可以方便地处理数学问题
文本处理	Python 提供了很多模块,如 re 模块能够处理正则表达式,又如 SGML、XML 分析模块可进行文本的编程开发
数据库编程	通过 Python DB-API(数据库应用程序编程接口)规范模块,可以与 Microsoft SQL Server、Oracle、Sybase、DB2、MySQL、SQLite 等数据库通信。Python 自带的 Gadget 模块可提供完整的 SQL 环境
网络编程	提供丰富的模块支持 Socket 编程,能够方便、快速地开发分布式应用程序
Web 编程	支持 HTML、XML 等标记语言
多媒体应用	Python 的 PyGame 模块可用于编写游戏软件,PyOpenGL 模块则封装了 OpenGL 应用程序编程接口,能进行二维和三维图像处理

## 1.5 Python 的安装

Python 语言是跨平台的,它可以运行在 Windows、MAC 和各种 Linux/UNIX 系统上。在 Windows 上编写 Python 程序,可以轻松方便地移植到 MAC 和各种 Linux/

UNIX 系统上。

在了解了 Python 的特点和应用领域之后,我们就可以进行 Python 语言开发的学习了。在学习之前,首先必须要知道如何获取 Python 开发工具、如何安装以及怎样启动 Python 开发工具,用 Python 编写的程序只有在安装了 Python 和配置好开发环境的前提下才能够运行,在这里我们将讲解如何获取、安装和启动 Python。

在告诉读者如何获取 Python 开发工具前,先给读者普及一些 Python 版本的知识。目前,Python 有两个版本系列,一个是 2. x 版,一个是 3. x 版,这两个版本是不兼容的,因为现在 Python 正在朝着 3. x 版本进化,在进化过程中,大量的针对 2. x 版本的代码要修改后才能运行,所以,目前有许多第三方库还暂时无法在 3. x 上使用。

为了保证用户的程序能够正常使用大量的第三方库,我们的教程仍以 2. x 版本为基础,确切地说,是目前官方网站中发布的最新版本 2. 7. 10。同时考虑到使用 Windows 系统的用户占了绝大部分,再者,初学者可能对 Linux 系统不熟悉,所以本节主要讲解基于 Windows 系统的安装。

## 1. Python 的获取

我们可以从 Python 的官方网站下载该软件。打开浏览器,在地址栏输入“https://www.python.org/”打开官方网站,选择 Downloads 菜单项,之后再单击 Python 2. 7. 10 (可能在读者看到本书时,Python 又已经出了新的版本,此时,可以单击 View the full list 查找到本书讲解的安装版本),如图 1-1 所示。

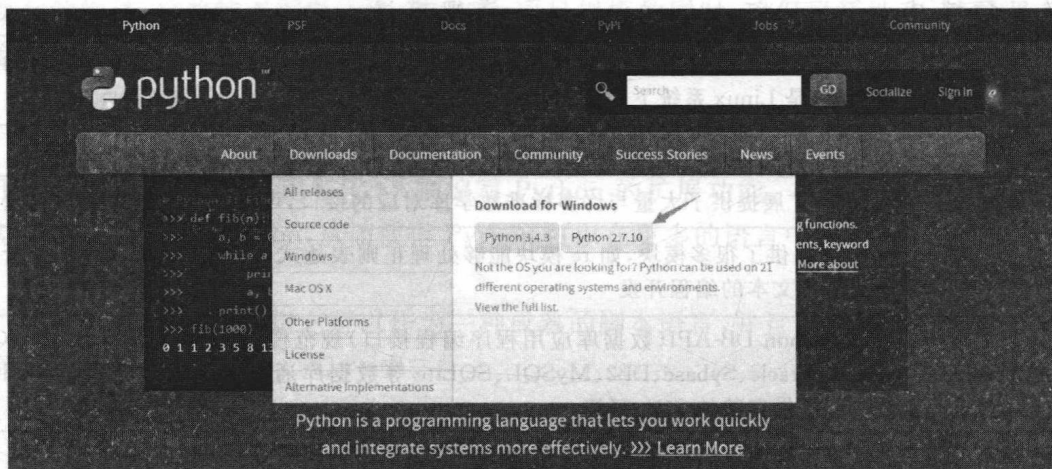


图 1-1 Python 官方网站 Python 2. 7. 10 下载提示图

## 2. Python 的安装

(1) 在下载目录中找到刚刚下载好的 Python 安装文件 python-2. 7. 10. msi, 双击这个文件, 会弹出 Python 安装程序的安装向导对话框, 如图 1-2 所示。

(2) 在这里我们可以看到两个单选按钮, 第一个 Install for all users 是为所有用户安装, 第二个 Install just for me 是为个人用户安装。我们单击第一个单选按钮, 然后单击 Next 按钮进入 Python 的安装路径设置界面, 如图 1-3 所示。

(3) 选择安装路径。可以把路径更改为硬盘的任意路径(建议不要安装在有中文的



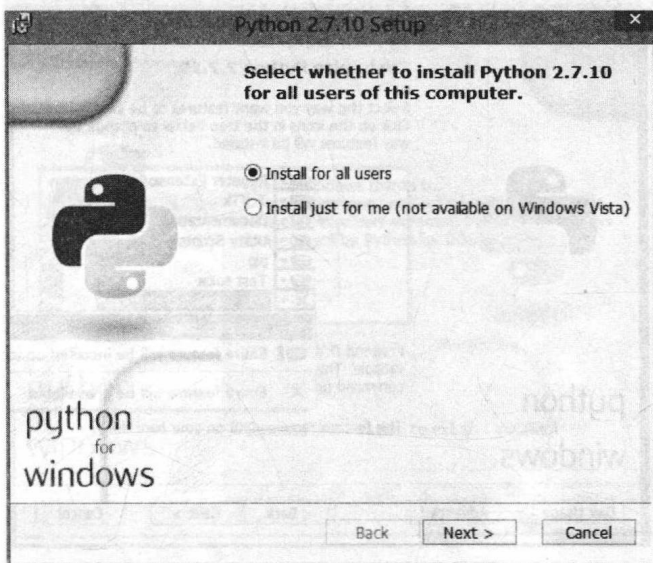


图 1-2 Python 安装程序向导对话框

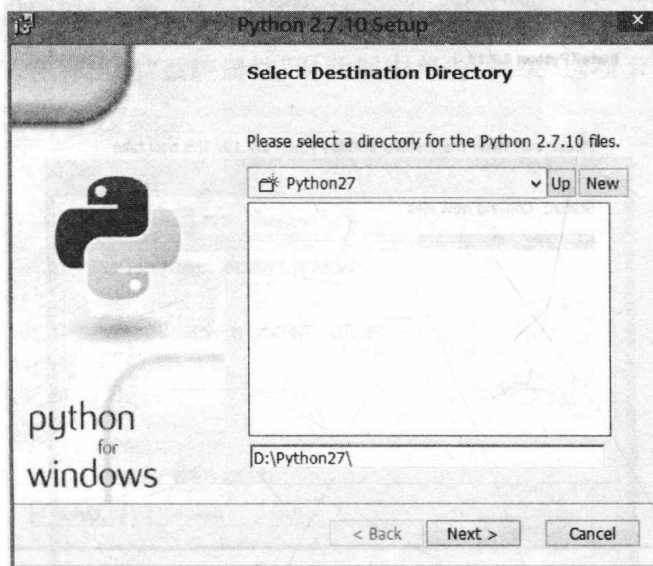


图 1-3 Python 安装路径设置界面

路径下,否则可能会无法启动 Python)。在这我们把 Python 安装在 D:\Python27 目录下,选择好安装路径后,单击 Next 按钮,进入 Python 安装组件选择界面,如果不想自己手动配置环境变量,可以下拉右边的滚动条,单击 Add python.exe to Path 左边的小三角图标,选择 Will be installed on local hard drive,选择这一步,系统在安装 Python 时会自动把其环境变量配置好,我们建议要学会自己手动配置环境变量,如图 1-4 所示。

(4) 单击 Next 按钮,进入 Python 工具包的安装界面,如图 1-5 所示。

(5) 等程序安装完成后,会提示程序安装成功界面,最后单击 Finish 按钮完成 Python 的安装,如图 1-6 所示。