



21世纪高等学校计算机
应用技术规划教材

Python 程序设计教程



© 杨年华 主编 柳青 郑戟明 副主编

清华大学出版社





21世纪高等学校计算机
应用技术规划教材

Python 程序设计教程



◎ 杨年华 主编 柳青 郑戟明 副主编



清华大学出版社
北京

内 容 简 介

本书共分为 16 章,第 1 章主要介绍 Python 的发展历史、特点、下载与安装方法、使用方式、集成开发环境、内置模块、帮助的使用等内容;第 2 章主要介绍 Python 语言的基础知识。第 3 章主要介绍程序控制结构;第 4 章主要介绍 Python 中的常用数据结构,包括序列、字典、集合等数据结构;第 5 章主要介绍函数的定义和调用、基于函数的抽象与求精思想、递归思想等内容。第 6 章主要介绍文件的操作;第 7 章主要介绍 Python 的面向对象编程方法;第 8 章主要介绍类的继承与组合两种重用方式;第 9 章主要介绍 Python 中的异常处理方法;第 10 章主要介绍使用 wxPython 进行用户图形界面设计的方法;第 11 章主要讨论 Python 程序的打包和发布方法;第 12 章主要介绍利用 Python 进行数据库应用开发;第 13 章主要介绍利用 Python 进行网络数据获取的方法;第 14 章主要介绍利用 Python 进行数据分析和绘图的基础知识;第 15 章主要介绍基于 Python 的网站开发方法;第 16 章主要介绍 Python 作为脚本语言在 SPSS 中的使用方法。本书中的代码均在 Python 2.7.11 中测试通过。

本书一方面侧重基础知识的讲解,另一方面侧重利用 Python 进行数据处理的方法和应用。为了方便理解,本书主要挑选经济管理类的案例。本书适合非计算机专业本科生使用,可作为计算机程序设计的入门教材或 Python 爱好者的参考书。

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/杨年华主编. —北京:清华大学出版社,2017
(21 世纪高等学校计算机应用技术规划教材)
ISBN 978-7-302-47722-8

I. ①P… II. ①杨… 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 160436 号

责任编辑:黄 芝 李 晔

封面设计:刘 键

责任校对:徐俊伟

责任印制:沈 露

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:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.25

字 数:459 千字

版 次:2017 年 10 月第 1 版

印 次:2017 年 10 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

Python 是一种简单、易学、免费、开源的跨平台编程语言,支持命令式和函数式编程。它支持完全面向对象的程序设计。一方面,由于其简单的语法,使得使用者不必太多关注语言本身,而将主要精力集中于业务逻辑。因此 Python 语言拥有各行各业的众多使用者,使得其拥有各行业使用者在社区贡献的各种强大的标准库、扩展库等。另一方面,随着大数据时代的到来,Python 的强大数据处理能力备受关注。近年来,Python 程序设计语言受到了企业界、科研单位和教育机构的广泛重视。

大数据时代的学生需要掌握数据处理的基本技术。Python 简单易学,具有强大的数据处理能力,并且是一门通用的程序设计语言。因此,Python 程序设计语言既适合作为程序设计的入门课程,也适合作为非计算机专业学生用来解决数据分析等各种问题的通用工具。国外很多著名高校的计算机或非计算机专业已经将 Python 作为程序设计入门课程。国内的高校也纷纷开设相关课程。尤其是随着计算思维和大数据概念的普及,Python 程序设计在高校中的教学开始全面展开。

本书编著者所在学校从 2014 级开始在经济管理类专业全校公共课中开设了相关课程,该课程主要面向经济管理类的学生开设。现有教材中的大部分案例面向理工科专业,难以贴近经济管理类专业,甚至有部分学生对案例难以理解。为了使上课的案例与学生专业知识更加紧密结合,我们组织编写了此教材。

本书由工作在教学第一线的高校教师编写完成。在编写本书时,编者注重理论与实践相结合,不仅有基础的理论知识,更有详细、通俗易懂的案例。作为一本介绍 Python 基础知识与应用的教材,本书内容简单易懂、层次脉络清晰、难度适中,内容、案例、难点安排恰当,非常适合教学。

本书共 16 章,主要内容及编写分工如下:

第 1 章由杨年华负责编写,主要介绍 Python 的发展历史、特点、下载与安装方法、使用方式、集成开发环境、内置模块介绍、帮助的使用等。

第 2 章由郑戟明负责编写,主要介绍 Python 语言的基础知识,包括控制台的使用、标识符与变量、数据类型、常用内置函数等。

第 3 章由张晓黎负责编写,主要介绍程序控制结构,包括分支控制、循环控制等。

第 4 章由郑戟明负责编写,主要介绍 Python 中的常用数据结构,包括序列、字典、集合等数据结构。

第 5 章由柳青负责编写,主要介绍函数的定义和调用、形参与实参、函数的返回、位置参数、默认参数、关键参数、可变长度参数、序列作为参数、基于函数的抽象与求精思想、递归思想等。

第 6 章由肖宇负责编写,主要介绍文件的打开与关闭、文件读写、文件指针、文件对话框等。

第7章由柳青负责编写,主要介绍Python的对象与方法、类的定义、类的属性、构造函数、类的方法、析构函数、可变对象与不可变对象、运算符的重载等。

第8章由杨年华负责编写,主要介绍类的继承与组合两种重用方式。

第9章由杨年华负责编写,主要介绍Python中的异常、异常类、异常的捕获与处理、自定义异常类、with语句、断言等。

第10章由孙辞海负责编写,主要介绍使用wxPython进行用户图形界面设计的方法。

第11章由孙辞海负责编写,主要讨论Python程序的打包和发布方法。

第12章由肖宇负责编写,主要介绍利用Python进行数据库应用开发。

第13章由肖宇负责编写,主要介绍利用Python进行网络数据获取的方法。

第14章由张晓黎负责编写,主要介绍利用Python进行数据分析和绘图基础知识。

第15章由孙辞海负责编写,主要介绍基于Python的网站开发方法。

第16章由曹玉茹负责编写,主要介绍Python作为脚本语言在SPSS中的使用方法。

本书适合非计算机专业本科生使用,也可作为计算机程序设计的入门教材或Python爱好者的参考书。

本书提供全套教学课件和源代码,配套资源可登录清华大学出版社官方网站下载。

由于时间仓促,作者水平有限,书中难免出现纰漏,不足之处敬请批评指正,并反馈给我们。

本书编写组

2017年4月

第 1 章 Python 概述	1
1.1 Python 语言的发展史	1
1.2 Python 语言的特点	2
1.3 Python 的下载与安装	3
1.3.1 Python 的下载	3
1.3.2 Python 的安装	3
1.4 开始使用 Python	6
1.4.1 交互方式	6
1.4.2 代码文件方式	7
1.4.3 代码风格	8
1.5 Python 的集成开发环境	9
1.5.1 Eclipse 中的 PyDev 插件	9
1.5.2 Eric	11
1.6 模块	11
1.6.1 标准模块	12
1.6.2 第三方模块	13
1.7 使用帮助	14
1.8 本章小结	17
习题 1	17
第 2 章 Python 语言基础知识	18
2.1 输入与输出	18
2.1.1 数据的输入	18
2.1.2 数据的输出	21
2.2 标识符与变量	22
2.2.1 标识符	22
2.2.2 变量	22
2.2.3 赋值语句	23
2.3 数据类型及运算	23
2.3.1 数据类型	23
2.3.2 运算符和表达式	25
2.3.3 运算表达式	25

2.4 常见的 Python 函数	26
2.5 本章小结	29
习题 2	29
第 3 章 控制语句	30
3.1 分支结构控制语句	30
3.1.1 if 语句	30
3.1.2 if/else 语句	31
3.1.3 if/elif/else 语句	33
3.1.4 选择结构嵌套	35
3.2 循环结构控制语句	37
3.2.1 while 语句	37
3.2.2 for 语句	39
3.2.3 循环嵌套	40
3.2.4 break 语句和 continue 语句	43
3.3 应用实例	46
3.3.1 学生成绩统计	46
3.3.2 天气状况分析	47
3.4 本章小结	49
习题 3	49
第 4 章 常用数据结构	51
4.1 序列	51
4.1.1 列表 list	51
4.1.2 元组 tuple	58
4.1.3 字符串	59
4.1.4 列表与元组之间的转换	64
4.2 字典	65
4.2.1 创建字典	66
4.2.2 字典操作	66
4.2.3 字典方法	67
4.2.4 列表、元组与字典之间的转换	68
4.3 集合	71
4.3.1 集合的创建	71
4.3.2 集合的运算	72
4.3.3 集合的方法	73
4.4 本章小结	76
习题 4	76

第 5 章 函数的设计	77
5.1 函数的定义	77
5.2 函数的调用	80
5.3 形参与实参	82
5.4 函数的返回	84
5.5 位置参数	87
5.6 默认参数与关键参数	88
5.7 可变长度参数	90
5.8 序列作为参数	94
5.9 基于函数的抽象与求精	97
5.9.1 自顶向下设计	97
5.9.2 自顶向下的实现	98
5.9.3 自底向上的实现与单元测试	101
5.10 递归	105
5.11 本章小结	108
习题 5	108
第 6 章 文件操作	110
6.1 打开与关闭文件	110
6.2 读写文件	110
6.2.1 从文件读取数据	111
6.2.2 向文件写入数据	113
6.3 文件指针	114
6.4 文件对话框	114
6.4.1 基于 win32ui 构建文件对话框	115
6.4.2 基于 tkinter 构建文件对话框	116
6.5 应用实例：文本文件操作	117
6.6 本章小结	120
习题 6	121
第 7 章 类与对象	122
7.1 认识 Python 中的对象和方法	122
7.2 类的定义	123
7.3 类的属性	125
7.3.1 类属性和实例属性	125
7.3.2 公有属性和私有属性	126
7.4 构造函数	127
7.5 类的方法	129

7.5.1 类的方法调用的过程	129
7.5.2 类的方法分类	129
7.6 析构函数	131
7.7 可变对象与不可变对象	132
7.8 get 和 set 方法	134
7.9 运算符的重载	137
7.10 面向对象和面向过程	143
7.10.1 类的抽象与封装	143
7.10.2 面向过程编程	143
7.10.3 面向对象编程	144
7.11 本章小结	148
习题 7	149
第 8 章 类的重用	151
8.1 类的重用方法	151
8.2 类的继承	151
8.2.1 父类与子类	151
8.2.2 继承的语法	152
8.2.3 子类继承父类的属性	154
8.2.4 子类继承父类的方法	156
8.2.5 继承关系下的构造方法	159
8.2.6 多重继承	162
8.3 类的组合	164
8.3.1 组合的语法	164
8.3.2 继承与组合的结合	166
8.4 本章小结	167
习题 8	167
第 9 章 异常处理	168
9.1 异常	168
9.2 Python 中的异常类	169
9.3 捕获与处理异常	171
9.4 自定义异常类	173
9.5 with 语句	174
9.6 断言	175
9.7 本章小结	176
习题 9	176

第 10 章 图形用户界面程序设计	177
10.1 图形用户界面平台的选择	177
10.2 wxPython 的安装	178
10.3 Hello World 的窗口程序	178
10.4 布局与事件	179
10.4.1 BoxSizer	179
10.4.2 GridSizer	180
10.4.3 事件处理	181
10.5 使用 wxFormBuilder 设计界面	181
10.6 应用实例:条形码图片识别	184
10.6.1 应用需求	184
10.6.2 条形码识别程序	185
10.6.3 界面设计	185
10.6.4 完整代码	186
10.7 本章小结	189
习题 10	189
第 11 章 程序打包发布	190
11.1 setuptools 程序打包发布工具	190
11.1.1 程序为什么要打包	190
11.1.2 推荐使用 setuptools 打包发布	190
11.1.3 setuptools 使用步骤	191
11.2 py2exe 打包	192
11.2.1 py2exe 的安装	192
11.2.2 py2exe 的简易打包	193
11.2.3 py2exe 的高级打包技巧	194
11.3 应用实例	195
11.4 本章小结	197
习题 11	197
第 12 章 数据库应用开发	198
12.1 Python Database API 简介	198
12.1.1 全局变量	198
12.1.2 连接与游标	199
12.2 结构化查询语言	200
12.2.1 数据定义语言	200
12.2.2 数据操作语言	202
12.2.3 数据查询语言	203

12.3	SQLite	203
12.3.1	SQLite 数据类型	204
12.3.2	sqlite3 模块	206
12.4	应用实例：学生管理数据库系统	209
12.4.1	数据表结构	209
12.4.2	学生管理数据库系统实现	210
12.5	本章小结	215
	习题 12	216
第 13 章 网络数据获取		217
13.1	网页数据的组织形式	217
13.1.1	HTML	217
13.1.2	XML	220
13.2	利用 urllib 处理 HTTP 协议	222
13.3	利用 BeautifulSoup4 解析 HTML 文档	226
13.3.1	BeautifulSoup4 中的对象	227
13.3.2	遍历文档树	230
13.4	应用实例	234
13.5	本章小结	240
	习题 13	241
第 14 章 数据分析与绘图基础		242
14.1	numpy 基础与常用函数	242
14.1.1	numpy 的 ndarray 数组类	242
14.1.2	数组的元素级运算与函数	244
14.1.3	数组的基本统计分析函数	245
14.2	matplotlib 基础与常用参数设置	247
14.2.1	折线图	248
14.2.2	散点图	251
14.2.3	直方图	251
14.3	常用分析函数与绘图示例	252
14.3.1	简单移动平均	252
14.3.2	指数移动平均	253
14.4	本章小结	255
	习题 14	255
第 15 章 网站设计		256
15.1	网站应用的发展历史与展望	256
15.2	HTTP 超文本传输协议	257

15.2.1	什么是 HTTP	257
15.2.2	HTTP 的具体内容	257
15.3	HTML 超文本标记语言	259
15.3.1	什么是 HTML	259
15.3.2	HTML、CSS、JavaScript 的简介	259
15.4	使用 WSGI 接口创建动态网页	261
15.5	使用 Python 网络框架来建立网站	264
15.6	应用实例:报名网站	265
15.7	本章小结	268
习题 15	268
第 16 章	在 SPSS 中使用 Python	270
16.1	SPSS Syntax 简介	270
16.1.1	程序编辑窗口界面	271
16.1.2	Paste 按钮	271
16.2	SPSS 中 Python 插件的安装	272
16.2.1	安装工具	272
16.2.2	工具设置	272
16.3	SPSS 中运行 Python	273
16.3.1	SPSS 中运行 Python 方式	273
16.3.2	SPSS 中运行 Python 案例	276
16.4	本章小结	277
习题 16	277
参考文献	278

第 1 章

Python概述

本章学习目标

- 熟练掌握 Python 开发环境的安装方法
- 熟悉 Python 开发环境的使用方法
- 熟悉第三方模块的安装方法
- 熟悉帮助文档的查看方法

本章首先向读者讲述 Python 的发展历史与特点,再介绍 Python 开发环境的安装,并以简单的实例介绍 Python 开发环境的使用方法;然后介绍两种开源的基础开发环境;接着介绍第三方模块的安装方法;最后介绍如何查看帮助信息。

1.1 Python 语言的发展史

Python 的发明者 Guido von Rossum(吉多·范罗苏姆,见图 1.1),荷兰人。1982 年,Guido 从阿姆斯特丹大学(University of Amsterdam)获得了数学和计算机硕士学位,并于同年加入 CWI(Centrum voor Wiskunde en Informatica,国家数学和计算机科学研究院)。

1989 年,Guido 开始设计 Python 语言的编译/解释器,以实现一种易学易用、可拓展的通用程序设计语言。Python 这个名字来自于 Guido 所挚爱的电视剧 Monty Python's Flying Circus。

1991 年,第一个用 C 语言实现的 Python 编译器/解释器诞生。从诞生之时起,Python 就具有类(class)、函数(function)、异常处理(exception)、列表(list)和字典(dictionary)等核心数据类型,允许在多个层次上进行扩展。

最初的 Python 完全由 Guido 开发。随着 Python 得到 Guido 同事们的欢迎,他们迅速地反馈使用意见,并参与了 Python 的改进。随后,Python 拓展到 CWI 之外。

Python 将许多机器层面上的实现细节隐藏,交给

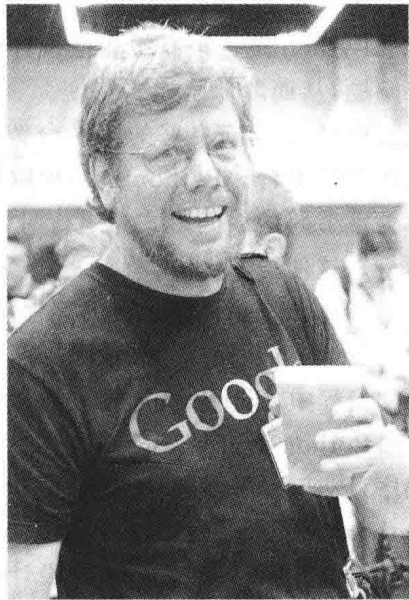


图 1.1 Guido von Rossum

编译器处理。Python 程序员可以花更多的时间用于思考程序的逻辑,而不是具体地实现细节。这一特征使得 Python 开始流行,尤其是在非计算机专业领域得到更加广泛的关注。

Python 是一种面向对象的、解释性的通用计算机程序设计语言。它以对象为核心组织代码(Everything is object),支持多种编程范式(multi-paradigm),采用动态类型(dynamic typing),自动进行内存回收(garbage collection)。它既具有强大的标准库(battery included),也拥有丰富的第三方扩展包。

目前,Python 已经进入到 3. x 的时代。由于 Python 3. x 向后不兼容,许多利用 Python 2. x 开发的第三方包在 3. x 版本中无法使用,从 2. x 到 3. x 的过渡将是一个漫长的过程。由于 Python 2. x 拥有丰富的第三方扩展包,方便应用开发时的直接调用,因此本书采用 Python 2. 7 版本。

现在,Python 已经成为最受欢迎的程序设计语言之一,它在 TIOBE 编程语言排行榜中的名次不断上升,其中 2016 年 1 月的排名升至第 5 位。

1.2 Python 语言的特点

Python 语法简洁、清晰。一个结构良好的 Python 程序就像伪代码,类似于用普通的英语描述一个事情的逻辑。因此 Python 程序设计语言也比较容易学习和掌握。Python 简单、易学的特点使得用户能够专注于解决问题的逻辑,而不为烦琐的语法所困惑。很多非计算机专业人士选择 Python 语言作为其解决问题的编程语言。同样,很多计算机专业也开始选择 Python 语言作为培养学生程序设计能力的入门语言。

Python 是纯粹的自由软件,源代码和解释器 CPython 遵循 GPL(GNU General Public License)协议。用户不但可以自由地下载使用,还可以自由地发布这个软件的副本、阅读它的源代码、改动源代码,把它的一部分用于新的自由软件中。在开源社区中有许多优秀的专业人士来维护、更新、改进 Python 语言。这些都是使得 Python 如此优秀的重要原因。

Python 是一个高级程序设计语言,用户在使用时无须考虑诸如如何管理内存之类的底层问题,从而降低了技术难度。

Python 具有良好的跨平台特性。可以运行于 Windows、UNIX、Linux、安卓等大部分操作系统平台。Python 是一种解释性语言。开发工具首先把 Python 编写的源代码转换成称为字节码的中间形式。运行时,解释器再把字节码翻译成适合于特定环境的机器语言并运行。这使得 Python 程序更加易于移植。

Python 支持面向过程的编程,程序可以由过程或仅仅是由可重用代码的函数构建起来。同时,Python 从设计之初就是一门面向对象的语言,因此也支持面向对象的编程。在面向对象的编程中,Python 程序由表示数据的属性和表示特定功能的方法组合而成的类来构建。

Python 语言具有良好的可扩展性。例如,Python 可以调用使用 C、C++ 等语言编写的程序,Python 可以调用 R 语言中的对象以利用其专业的数据分析能力。这一特性使得 Python 语言适合用来进行系统集成,也可以包含使用者原有的软件资产。同样也可以将 Python 程序嵌入到其他程序设计语言中,或者作为一些软件的二次开发脚本语言。例如,Python 可以作为 SPSS 的脚本语言。另一方面,在 Python 程序中嵌入其他程序设计语言编写的模块可能会在一定程度上影响 Python 程序的可移植性。

Python 标准库非常庞大,可以处理各种工作。而且,由于 Python 开源、免费的特征,不同社区的 Python 爱好者贡献了大量实用且高质量的扩展库,方便在程序设计时直接调用。

在 Python 的 2. x 版本中, long 类型的整数长度只受计算机内存的限制,可以表示一个很大的整数。在 3. x 版本中,将 int 和 long 两种类型合并为 int。因此在 3. x 版本中, int 类型表示的长度不受限制,直到内存耗尽。另一方面,在进行大量数据分析时, Python 的运算速度相对较快。 Python 的这些特点使得其更适合处理大数据相关应用。

Python 采用强制空格缩进的方式使得代码具有较好可读性。但是这种使用强制空格缩进的方式同时也带来了一些隐患,使得一些无意的触碰键盘等行为可能导致空格的增删,从而导致程序的逻辑错误。

1.3 Python 的下载与安装

1.3.1 Python 的下载

用户可以从 <https://www.Python.org/ftp/Python/> 下载相应版本的 Python 源代码、安装程序和帮助文件等。在网页上单击相应版本号(如 2. 7. 11)后,用户根据所使用的操作系统,选择适合不同操作系统的文件。例如,用户要安装到 64 位 Windows 操作系统,可以下载名为 Python-2. 7. 11. amd64. msi 的文件;如果是 32 位 Windows 操作系统,则选择 Python-2. 7. 11. msi。

1.3.2 Python 的安装

下面以在 Windows 7 的 32 位操作系统上安装 Python 2. 7. 11 版本为例,简要介绍 Python 开发环境的安装过程,步骤如下:

(1) 双击安装程序 Python-2. 7. 11. msi,进入如图 1. 2 所示的界面。



图 1.2 在 Windows 下安装 Python—选择是否所有操作系统用户可用

(2) 在如图 1.2 所示的界面中单击 Next 按钮,出现如图 1.3 所示的界面。



图 1.3 在 Windows 下安装 Python—选择安装目录

(3) 在如图 1.3 所示的界面中选择 Python 的安装路径,然后单击 Next 按钮,出现如图 1.4 所示的界面。

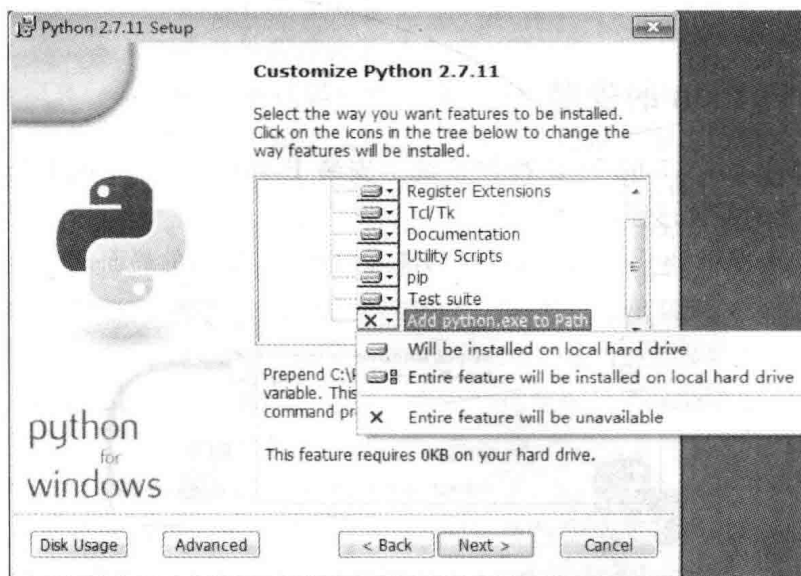


图 1.4 在 Windows 下安装 Python—选择安装组件

(4) 在如图 1.4 所示的界面中,将滚动条拉到最后,单击 Add python.exe to Path 左边的下三角按钮 ▾,并选择 Will be installed on local hard drive。然后单击 Next 按钮,出现如图 1.5 所示的界面。

(5) 稍后,出现如图 1.6 所示的界面,单击 Finish 按钮,结束安装。

大部分的 Linux 操作系统(如 Ubuntu)默认安装就包含了 Python 开发环境。如果要安

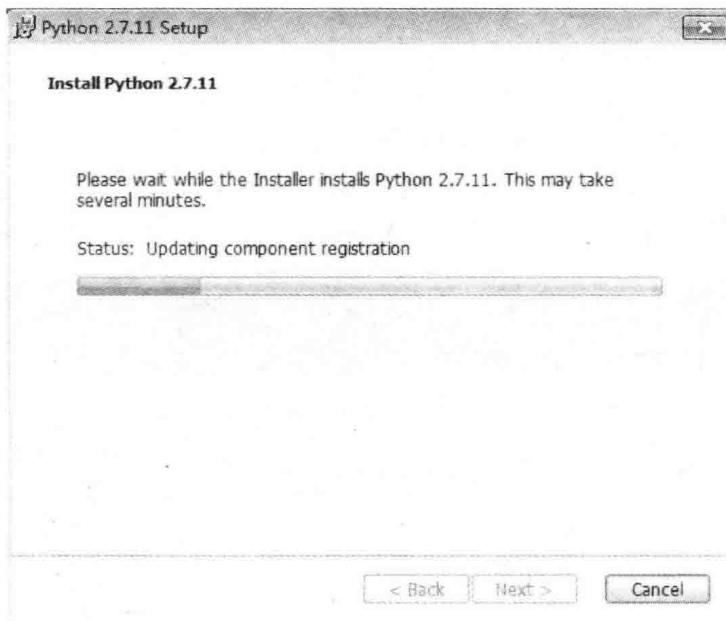


图 1.5 在 Windows 下安装 Python—安装进程状态显示



图 1.6 在 Windows 下安装 Python—结束安装

装特定版本的 Python,需要自己手动安装。这里简单介绍一下在 Ubuntu 操作系统中如何安装 Python 2.7.11 版本。在其他版本的 Linux 操作系统下安装方法相同。具体步骤如下:

(1) 在 Python 官方网站下载与用户 Ubuntu 版本相适应的 Python 版本。本书使用的是 32 位的 Ubuntu 操作系统。与此版本操作系统相适应的 Python 的下载地址为 <https://www.python.org/ftp/python/2.7.11/python-2.7.11.tar.xz>。下载的软件包是 Python-