



普通高等教育

软件工程

“十二五”规划教材

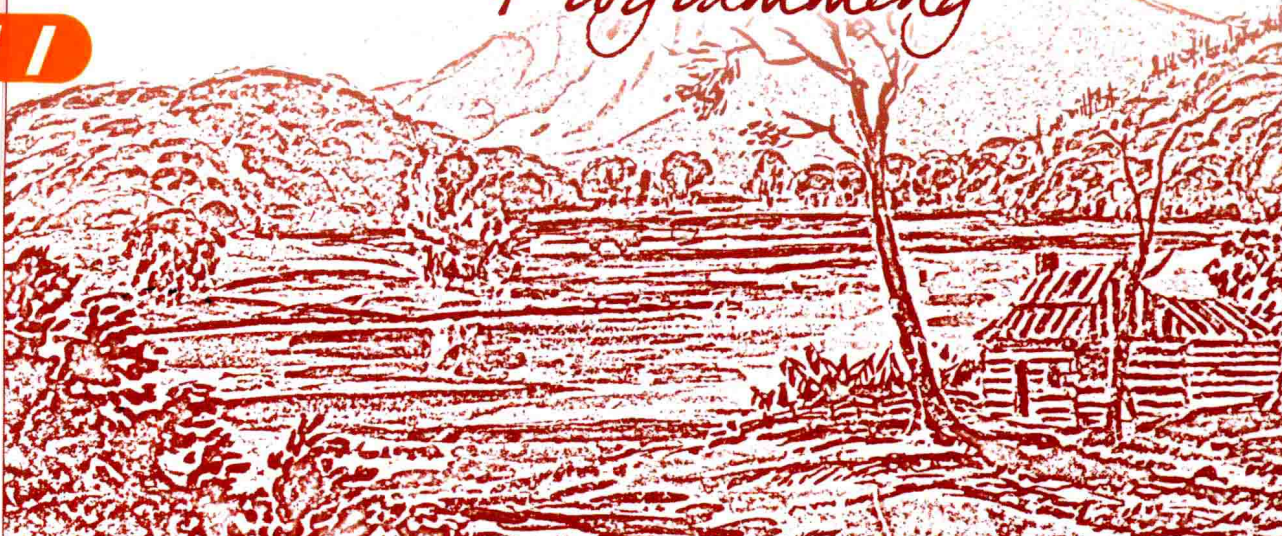
12th Five-Year Plan Textbooks
of Software Engineering

Python基础教程

刘浪 ◎ 主编

郭江涛 于晓强 宋燕红 ◎ 编著

*Python
Programming*



中国通信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



普通高等教育

软件工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

Python基础教程

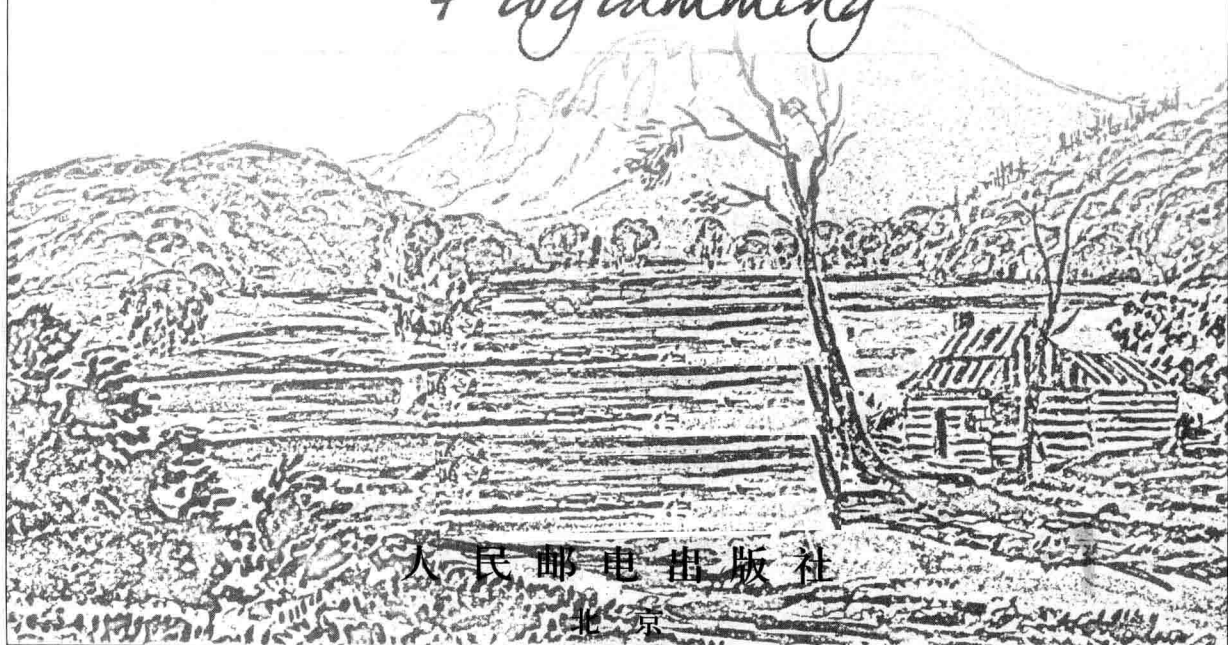
刘浪 ◎ 主编

郭江涛 于晓强 宋燕红 ◎ 编著

*Python
Programming*

人民邮电出版社

北京



图书在版编目(CIP)数据

Python基础教程 / 刘浪主编 ; 郭江涛, 于晓强, 宋燕红编著. — 北京 : 人民邮电出版社, 2015.9
普通高等教育软件工程“十二五”规划教材
ISBN 978-7-115-39868-0

I. ①P… II. ①刘… ②郭… ③于… ④宋… 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165630号

内 容 提 要

Python 诞生于 20 世纪 90 年代初, 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, 是最受欢迎的程序设计语言之一。本书包括基础篇和高级篇, 全面介绍 Python 编程的基础知识和实用技术。读者在阅读本书时可以充分了解和体验 Python 语言的强大功能。

本书既可以作为大学本科“应用程序设计”课程的教材, 也可作为高职高专院校相关专业的教材, 或作为 Web 开发人员的参考用书。

-
- ◆ 主 编 刘 浪
编 著 郭江涛 于晓强 宋燕红
责任编辑 邹文波
责任印制 沈 蓉 彭志环
 - ◆ 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>
北京圣夫亚美印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 18.5 2015 年 9 月第 1 版
字数: 376 千字 2015 年 9 月北京第 1 次印刷

定价: 45.00 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

前 言

Python 诞生于 20 世纪 90 年代初，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，是最受欢迎的程序设计语言之一。高等院校的许多专业都开设了相关的课程。

Python 语言简洁，语法简单，只需要掌握基本的英文单词就可以读懂 Python 程序。Python 兼容很多平台，支持的平台包括 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 和 PocketPC 等。

编者在多年使用 Python 开发应用程序和研究相关课程教学的基础上编写了本书。本书内容分为两篇。第 1 篇介绍基础知识，由第 1~6 章组成，包括 Python 概述、Python 语言基础、Python 函数、Python 面向对象程序设计、Python 模块和函数式编程；第 2 篇介绍 Python 编程的高级技术，由第 7~11 章组成，详尽地讲解了 I/O 编程、图形界面编程、多任务编程、网络编程以及 Python 数据库编程等技术。另外，本书每章都配有相应的习题，帮助读者理解所学习的内容，使读者加深印象、学以致用。

本书提供教学 PPT 课件和源程序文件等，需要者可以登录人民邮电出版社教学服务与资源网 (<http://www.ptpedu.com.cn>) 免费下载。

本书在内容的选择及深度的把握上充分考虑初学者的特点，内容安排上力求做到循序渐进，不仅适合教学，也适合开发 Web 应用程序的各类人员自学使用。

由于编者水平有限，书中难免存在不足之处，敬请广大读者批评指正。

编 者
2015 年 4 月

目 录

第 1 篇 基础篇

第 1 章 Python 概述 2

1.1 初识 Python	2
1.1.1 什么是 Python	2
1.1.2 Python 的特性	3
1.2 开始 Python 编程	5
1.2.1 下载和安装 Python	5
1.2.2 执行 Python 脚本文件	6
1.2.3 Python 语言的基本语法	7
1.2.4 下载和安装 Pywin32	7
1.3 Python 文本编辑器 IDLE	9
1.3.1 打开 IDLE	9
1.3.2 新建 Python 脚本	10
1.3.3 保存 Python 脚本	10
1.3.4 打开 Python 脚本	10
1.3.5 语法高亮	10
1.3.6 自动完成	10
1.3.7 语法提示	11
1.3.8 运行 Python 程序	11
1.3.9 IDLE 的菜单项	12
习 题	13

第 2 章 Python 语言基础 14

2.1 常量和变量	14
2.1.1 常量	14
2.1.2 变量	16
2.1.3 常量与变量的数据类型转换	19
2.2 运算符和表达式	21
2.2.1 运算符	21
2.2.2 表达式	24
2.3 常用语句	25
2.3.1 赋值语句	25
2.3.2 条件分支语句	25

2.3.3 循环语句	28
2.3.4 try-except 异常处理语句	30
2.4 序列数据结构	31
2.4.1 列表的应用与实例	31
2.4.2 元组应用与实例	38
2.4.3 字典的应用与实例	41
2.4.4 集合的应用与实例	45
习 题	51

第 3 章 Python 函数 53

3.1 声明和调用函数	53
3.1.1 自定义函数	53
3.1.2 调用函数	54
3.1.3 变量的作用域	55
3.1.4 在调试窗口中查看变量的值	55
3.2 参数和返回值	57
3.2.1 在函数中传递参数	57
3.2.2 函数的返回值	63
3.3 Python 内置函数的使用	64
3.3.1 数学运算函数	64
3.3.2 字符串处理函数	64
3.3.3 其他常用内置函数	69
习 题	74

第 4 章 Python 面向对象程序设计 76

4.1 面向对象程序设计基础	76
4.1.1 面向对象程序设计思想概述	76
4.1.2 面向对象程序设计中的基本概念	77
4.2 定义和使用类	77
4.2.1 声明类	77
4.2.2 静态变量	80

4.2.3	静态方法的使用	81
4.2.4	类方法的使用	82
4.2.5	使用 instance()函数判断对象类型	82
4.3	类的继承和多态	83
4.3.1	继承	83
4.3.2	抽象类和多态	85
4.4	复制对象	87
4.4.1	通过赋值复制对象	87
4.4.2	通过函数参数复制对象	88
	习题	89

第5章 Python 模块..... 90

5.1	Python 标准库中的常用模块	90
5.1.1	sys 模块	90
5.1.2	platform 模块	93
5.1.3	与数学有关的模块	97
5.1.4	time 模块	102
5.2	自定义和使用模块	105
5.2.1	创建自定义模块	105
5.2.2	导入模块	105

第2篇

第7章 I/O 编程..... 120

7.1	输入和显示数据	120
7.1.1	输入数据	120
7.1.2	输出数据	121
7.2	文件操作	123
7.2.1	打开文件	123
7.2.2	关闭文件	124
7.2.3	读取文件内容	124
7.2.4	写入文件	127
7.2.5	文件指针	129
7.2.6	截断文件	130
7.2.7	文件属性	130
7.2.8	复制文件	132
7.2.9	移动文件	132
7.2.10	删除文件	133

习题	106
----	-----

第6章 函数式编程..... 107

6.1	函数式编程概述	107
6.1.1	什么是函数式编程	107
6.1.2	函数式编程的优点	108
6.2	Python 函数式编程常用的函数	108
6.2.1	lambda 表达式	109
6.2.2	使用 map()函数	110
6.2.3	filter()函数	111
6.2.4	reduce()函数	112
6.2.5	zip()函数	113
6.2.6	普通编程方式与函数式编程的对比	114
6.3	闭包和递归函数	114
6.3.1	闭包	115
6.3.2	递归函数	115
6.4	迭代器和生成器	116
6.4.1	迭代器	116
6.4.2	生成器	117
	习题	118

高级篇

7.2.11	重命名文件	133
7.3	目录编程	133
7.3.1	获取当前目录	133
7.3.2	获取目录内容	133
7.3.3	创建目录	134
7.3.4	删除目录	134
	习题	134

第8章 图形界面编程..... 136

8.1	常用 Tkinter 组件的使用	136
8.1.1	弹出消息框	136
8.1.2	创建 Windows 窗口	139
8.1.3	Label 组件	141
8.1.4	Button 组件	144
8.1.5	Canvas 画布组件	146
8.1.6	Checkbutton 组件	158

8.1.7	Entry 组件	160	10.3.1	SMTP 编程	220
8.1.8	Frame 组件	161	10.3.2	POP 编程	224
8.1.9	Listbox 组件	162	习 题		231
8.1.10	Menu 组件	164	第 11 章 Python 数据库编程	233	
8.1.11	Radiobutton 组件	167	11.1	数据库技术基础	233
8.1.12	Scale 组件	168	11.1.1	数据库的基本概念	233
8.1.13	Text 组件	170	11.1.2	关系数据库	235
8.2	窗体布局	171	11.2	SQLite 数据库	235
8.2.1	pack()方法	171	11.2.1	下载和安装 SQLite 数据库	235
8.2.2	grid()方法	172	11.2.2	创建 SQLite 数据库	236
8.2.3	place()方法	174	11.2.3	数据类型	237
8.3	Tkinter 字体	174	11.2.4	创建表	237
8.3.1	导入 tkFont 模块	175	11.2.5	向表中添加列	239
8.3.2	设置组件的字体	175	11.2.6	向表中插入数据	240
8.4	事件处理	176	11.2.7	修改表中的数据	241
习 题		179	11.2.8	删除数据	241
第 9 章 多任务编程	180		11.2.9	查询数据	242
9.1	多进程编程	180	11.2.10	在 Python 中访问 SQLite 数据库	242
9.1.1	什么是进程	180	11.3	MySQL 数据库	245
9.1.2	进程的状态	181	11.3.1	安装 MySQL 数据库	245
9.2	进程编程	181	11.3.2	MySQL-Front	248
9.2.1	创建进程	182	11.3.3	创建数据库	249
9.2.2	枚举系统进程	185	11.3.4	删除数据库	250
9.2.3	终止进程	188	11.3.5	MySQL 数据类型	250
9.2.4	进程池	189	11.3.6	创建表	252
9.3	多线程编程	191	11.3.7	编辑和查看表	254
9.3.1	线程的概念	191	11.3.8	删除表	255
9.3.2	threading 模块	193	11.3.9	插入数据	255
习 题		207	11.3.10	修改数据	257
第 10 章 网络编程	209		11.3.11	删除数据	258
10.1	网络通信模型和 TCP/IP 协议簇	209	11.3.12	使用 SELECT 语句查询数据	259
10.1.1	OSI 参考模型	209	11.3.13	在 Python 中访问 MySQL 数据库	265
10.1.2	TCP/IP 协议簇体系结构	210	习 题		267
10.2	Socket 编程	212	附录 实验	269	
10.2.1	Socket 的工作原理和基本概念	212	实验 1	开始 Python 编程	269
10.2.2	基于 TCP 的 Socket 编程	214	实验 2	Python 语言基础	271
10.2.3	基于 UDP 的 Socket 编程	218			
10.3	电子邮件编程	220			

实验 3 Python 函数	273	实验 8 图形界面编程	282
实验 4 Python 面向对象程序设计	275	实验 9 多任务编程	284
实验 5 Python 模块	277	实验 10 网络编程	285
实验 6 函数式编程	279	实验 11 Python 数据库编程	287
实验 7 I/O 编程	280		

第 9 章 多线程编程

9.1 多线程编程	180
9.1.1 什么是多线程	180
9.1.2 多线程的优缺点	181
9.2 线程编程	181
9.2.1 创建线程	182
9.2.2 线程同步与互斥	184
9.2.3 停止线程	184
9.2.4 线程池	185
9.3 多线程编程	187
9.3.1 线程的概念	187
9.3.2 threading 模块	188
习 题	189

第 10 章 网络编程

10.1 网络编程模型和 TCP/IP 协议族	209
10.1.1 OSI 七层模型	209
10.1.2 TCP/IP 协议族体系结构	210
10.2 Socket 编程	212
10.2.1 Socket 的工作原理和基本概念	212
10.2.2 基于 TCP 的 Socket 编程	214
10.2.3 基于 UDP 的 Socket 编程	218
10.3 电子邮件编程	220

第一篇 基础篇



第 1 章

Python 概述

Python 诞生于 20 世纪 90 年代初，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，是最受欢迎的程序设计语言之一。本章介绍 Python 语言的基本情况。

1.1 初识 Python

首先来了解一下什么是 Python，它又有哪些特性。

1.1.1 什么是 Python

Python 于 20 世纪 80 年代末由荷兰人 Guido Van Rossum（见图 1-1）设计实现。

1991 年，Van Rossum 公布了 0.9.0 版本的 Python 源代码。此版本已经实现了类、函数以及列表、字典和字符串等基本的数据类型。本书将在第 2 章介绍基本数据类型，第 3 章介绍函数，第 4 章介绍类。

0.9.0 版本还集成了模块系统，Van Rossum 将模块描述为 Python 主要的编程单元。

1994 年，Python 1.0 发布了。1.0 版新增了函数式工具。关于函数式编程将在第 6 章介绍。

Python 2.0 集成了列表推导式（List comprehension），具体情况将在第 2 章介绍。

Python 3.0 也称为 Python 3000 或 Python 3K。相对于 Python 的早期版本，这是一个较大的升级。为了不带入过多的累赘，Python 3.0 在设计的时候没有考虑向下兼容。Python 3.0 的主要设计思想就是通过移除传统的做事方式从而减少特性的重复。很多针对早期 Python 版本设计的程序都无法在 Python 3.0 上正常运行。为了照顾现有程序，Python 2.6 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



图 1-1 Guido Van Rossum

法与函数。基于早期 Python 版本而能正常运行于 Python 2.6 并无警告的程序，可以通过一个 2 到 3 的转换工具无缝迁移到 Python 3.0。本书内容基于 Python 3.0。

经过多年的发展，Python 已经成为非常流行的热门程序开发语言。到底有多流行？让我们看看知名的 TIOBE 开发语言排行榜。TIOBE 排行榜是根据互联网上有经验的程序员、课程和第三方厂商的数量，并使用搜索引擎（如 Google、Bing、Yahoo!、百度）以及 Wikipedia、Amazon、YouTube 统计出排名数据，用于反映编程语言的热门程度；但并不能说明一门编程语言好不好。该排行榜可以用来衡量开发者的编程技术能否跟上趋势，以及指导开发者应该及时掌握哪个编程语言。TIOBE 的官方网址如下：

<http://www.tiobe.com/>

2015 年 2 月的 TIOBE 排行榜显示，Python 排名第 8，如图 1-2 所示。

Feb 2015	Feb 2014	Change	Programming Language	Ratings	Change
1	1		C	16.488%	-1.85%
2	2		Java	15.345%	-1.97%
3	4	▲	C++	6.612%	-0.28%
4	3	▼	Objective-C	6.024%	-5.32%
5	5		C#	5.738%	-0.71%
6	9	▲	JavaScript	3.514%	+1.58%
7	6	▼	PHP	3.170%	-1.05%
8	8		Python	2.882%	+0.72%
9	10	▲	Visual Basic .NET	2.026%	+0.23%
10	-	▲	Visual Basic	1.718%	+1.72%
11	20	▲	Delphi/Object Pascal	1.574%	+1.05%
-	-	-	Perl	1.390%	+0.50%

图 1-2 2015 年 2 月的 TIOBE 排行榜

可以看到，排名前 10 的编程语言依次是 C、Java、C++、Objective-C、C#、JavaScript、PHP、Python、Visual Basic.NET 和 Visual Basic。很多流行的编程语言都没有入围前 10，如 Delphi、Perl、Transact-SQL 等，可见 Python 的流行程度。

1.1.2 Python 的特性

在学习 Python 语言之前，首先简要介绍一下 Python 的基本特性。

(1) 简单易学：Python 语言很简洁，语法也很简单，只需要掌握基本的英文单词就可以读懂 Python 程序。这对于初学者无疑是个好消息。因为简单就意味着易学，可以很轻松的上手。

(2) Python 是开源的、免费的：开源是开放源代码的简称。也就是说，用户可以免费获取 Python 的发布版本，阅读、甚至修改源代码。很多志愿者将自己的源代码添加到 Python 中，从而使其日臻完善。

(3) Python 是高级语言：与 Java 和 C 一样，Python 不依赖任何硬件系统，因此属于高级开发语言。在使用 Python 开发应用程序时，不需要关注低级的硬件问题，如内存管理。

(4) 高可移植性：由于开源的缘故，Python 兼容很多平台。如果在编程时多加留意系统依赖的特性，Python 程序无须进行任何修改，就可以在各种平台上运行。Python 支持的平台包括 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 和 PocketPC 等。

(5) Python 是解释型语言：计算机不能直接理解高级语言，只能直接理解机器语言。使用解释型语言编写的源代码不是直接翻译成机器语言，而是先翻译成中间代码，再由解释器对中间代码进行解释运行。因此使用 Python 编写的程序不需要翻译成二进制的机器语言，而是直接从源代码运行，即运行 Python 程序时，由 Python 解释器将源代码转换为字节码（中间代码），然后再执行这些字节码。过程如图 1-3 所示。

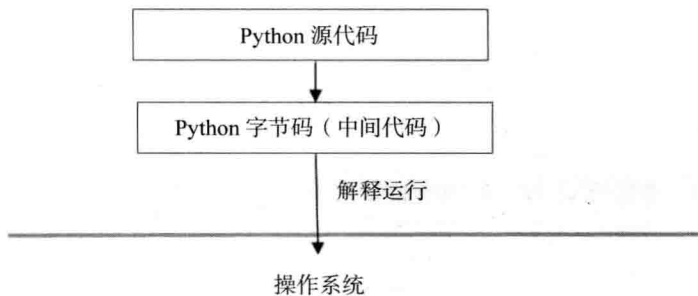


图 1-3 Python 程序的运行过程

(6) Python 全面支持面向对象的程序设计思想：面向对象是目前最流行的程序设计思想。所谓面向对象，就是基于对象的概念，以对象为中心，类和继承为构造机制，认识了解刻画客观世界以及开发出相应的软件系统。关于面向对象的程序设计思想的细节将在第 4 章介绍。

(7) 高可扩展性：如果希望一段代码可以很快的运行，或者不希望公开一个算法，则可以使用 C 或 C++ 编写这段程序，然后在 Python 中调用，从而实现了对 Python 程序的扩展。

(8) 支持嵌入式编程：可以将 Python 程序嵌入到 C/C++ 程序中，从而为 C/C++ 程序提供脚本能力。

(9) 功能强大的开发库：Python 标准库非常庞大，可以实现包括正则表达式、文档生成、单元测试、线程、数据库、浏览器、CGI、FTP、Email、XML、XML-RPC、HTML、加密、

GUI（图形用户界面）等功能。除了标准库外，还有很多其他的功能强大的库，本书后面部分会介绍这些库的具体情况。

1.2 开始 Python 编程

本节将介绍配置 Python 开发的环境，并介绍一个简单的 Python 程序。通过对本节的学习，读者就可以开始 Python 编程。

1.2.1 下载和安装 Python

访问网址 <https://www.python.org/downloads/>，可以下载 Python，如图 1-4 所示。

在编写本书时，Python for Windows 有 2 个最新版本，2.0 系列的最新版本为 Python 2.7.8，3.0 系列的最新版本为 Python 3.4.2。读者看到的情况也许会略有不同。本书内容基于 Python 3.4.2。

单击 Download Python 3.4.2 按钮，下载得到 python-3.4.2.msi。双击 python-3.4.2.msi，即可按照向导安装 Python 3.4.2。



图 1-4 下载 Python

Python 3.4.2 的默认安装目录为 C:\Python34。安装完成后，将 C:\Python34 添加到环境变量 Path 中。不同操作系统下添加环境变量的方法略有不同，这里就不具体介绍了。

在 Windows 7 中安装后，在“开始”菜单的所有程序中会出现一个 Python 3.4 分组。单击其下面的 Python 3.4 (command line - 32 bit) 菜单项，就可以打开 python 命令窗口，如图 1-5 所示。

也可以打开 Windows 命令窗口，然后运行 python 命令，来打开 python 命令窗口。

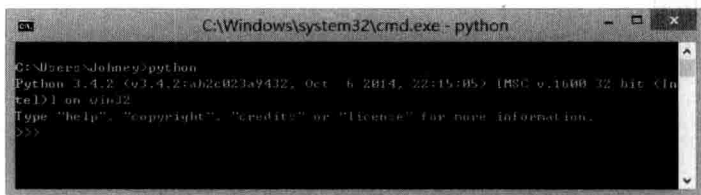


图 1-5 Python 3.4.2 安装成功后打开 python 命令窗口

python 命令实际上就是 Python 的解释器。在>>>后面输入 Python 程序，按回车键后即可被解释执行。例如，输入下面的代码，可以打印“我是 Python”，运行结果如图 1-6 所示。

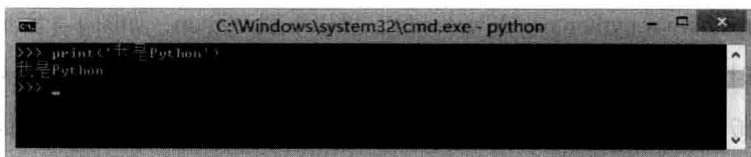


图 1-6 打印“我是 Python”的结果

```
print('我是 Python')
```

Print()函数用于输出数据，关于函数的具体情况将在第 3 章中介绍。按 Ctrl+Z 组合键可以退出 Python 环境。

1.2.2 执行 Python 脚本文件

1.2.1 小节介绍了在命令行里面执行 Python 程序的方法。这正是解释型语言的特点，可以一行一行语句地解释执行，不需要编译生成一个 exe 文件。但这也不是程序员所习惯的编程方式，比较大的应用程序都是存放在一个文件中，然后一起执行的。Python 当然也可以这样，Python 脚本文件的扩展名为.py。

【例 1-1】 创建一个文件 MyfirstPython.py，使用记事本编辑它的内容如下：

```
# My first Python program\nprint('I am Python')
```

保存后，打开命令窗口。切换到 MyfirstPython.py 所在的目录，然后执行下面的命令：

```
python MyfirstPython.py
```

运行结果如下：

```
I am Python
```

#是 Python 的注释符。关于注释将在 1.2.4 小节中介绍。'I am Python'是一个字符串。关于字符串的具体情况将在第 2 章介绍。

1.2.3 Python 语言的基本语法

本节介绍 Python 语言的基本语法，这些都是日后编写 Python 程序需要了解 and 注意的。

1. Python 语句

Python 程序由 Python 语句组成，通常一行编写一个语句。例如：

```
print('Hello,')
print('I am Python')
```

Python 语句可以没有结束符，不像 C 或 C#那样在语句后面必须有分号(;)表示结束。当然，Python 程序中也可以根据习惯在语句后面使用分号(;)。

也可以把多个语句写在一行，此时就要在语句后面加上分号(;)表示结束。

【例 1-2】 把多个语句写在一行的例子。

```
print('Hello,'); print('I am Python');
```

2. 缩进

缩进指在代码行前面添加空格或按 Tab 键，这样做可以使程序更有层次、更有结构感，从而使程序更易读。

在 Python 程序中，缩进不是任意的。平级的语句行（代码块）的缩进必须相同。

【例 1-3】 语句缩进的例子。

```
print('Hello,');
  print('I am Python');
```

运行这段程序的结果如下：

```
File "例 1-3.py", line 2
  print('I am Python');
  ^
IndentationError: unexpected indent
```

从输出的错误信息中可以看到，unexpected indent 表明缩进格式不对。因为第 2 行语句的开始有 1 个空格。可见 Python 的语法是很严谨的。

1.2.4 下载和安装 Pywin32

Python 是跨平台的编程语言，兼容很多平台。本书内容基于 Windows 平台，Pywin32 是 Windows 平台下的 Python 扩展库，提供了很多与 Windows 系统操作相关的模块。本书后面介绍的一些功能和实例就是基于 Pywin32 的。本节介绍下载和安装 Pywin32 的方法。

访问网址 <http://sourceforge.net/projects/pywin32/>，可以下载 Pywin32 安装包。

网站页面如图 1-7 所示。单击 Browse All Files 超链接，可以打开选择产品页面，如图 1-8 所示。

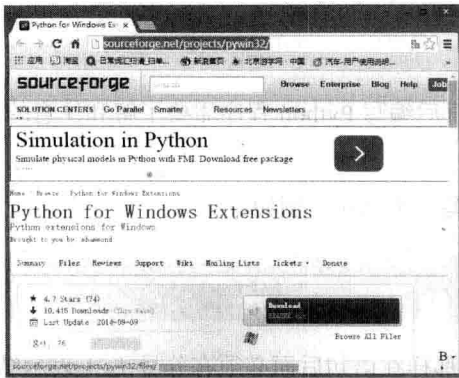


图 1-7 Pywin32 项目主页

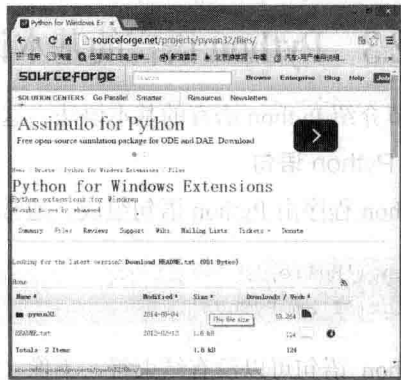


图 1-8 选择产品页面

单击 `pywin32` 目录，可以打开选择 Pywin32 版本的页面，如图 1-9 所示。单击最新的 Pywin32 版本超链接，可以打开下载文件列表页，如图 1-10 所示。

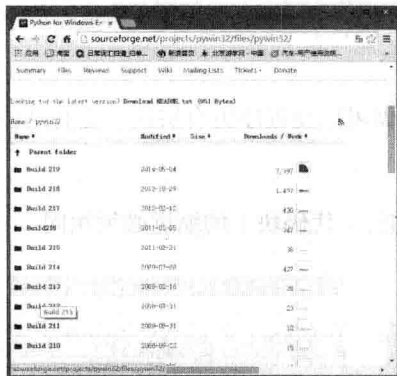


图 1-9 选择 Pywin32 版本

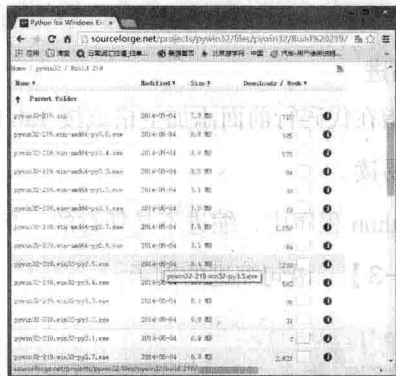


图 1-10 下载文件列表页面

在笔者编写本书时，Pywin32 的最新版本为 219。根据 Python 的版本选择要下载的安装包。例如，本书使用的是 Python3.4，因此单击 `pywin32-219.win32-py3.4.exe` 超链接，可以下载得到 Pywin32 的安装包 `pywin32-219.win32-py3.4.exe`。



当读者阅读本书时，下载页面和 Pywin32 的最新版本可能都会发生变化。读者可以参照上面的内容自行查找，也可以通过搜索引擎搜索下载 Pywin32 的相关页面。本书的源代码包里也提供了 `pywin32-219.zip`，读者可以直接使用。

运行 `pywin32-219.win32-py3.4.exe`，就可以安装 Pywin32。首先打开欢迎窗口，如图 1-11 所示。单击“下一步”按钮，打开选择目录窗口，如图 1-12 所示。

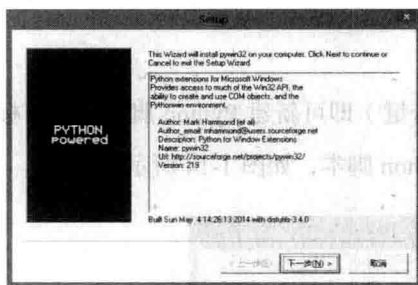


图 1-11 欢迎窗口

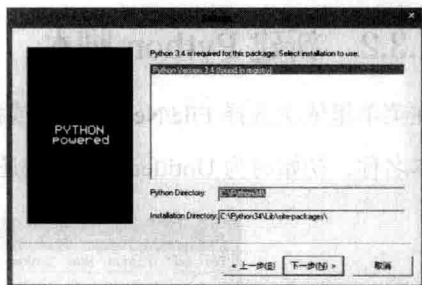


图 1-12 选择目录窗口

安装程序会从注册表中自动获取 Python3.4 的安装目录（如 C:\Python34），默认的 Pywin32 安装目录是 C:\Python34\Lib\site-packages，读者也可以手动设置。设置完成后，单击“下一步”按钮，打开准备安装窗口，再单击“下一步”按钮即可开始安装。安装完成后，会打开完成窗口。

1.3 Python 文本编辑器 IDLE

Python 是一种脚本语言，它并没有提供一个官方的开发环境，需要用户自主选择编辑工具。

1.3.1 打开 IDLE

Python 对文本编辑器没有特殊要求，完全可以使用 Windows 记事本编辑 Python 程序。但是 Windows 记事本的功能毕竟太过简单，而且没有对 Python 的特殊支持，因此不建议使用。

本节介绍 Python 自带的文本编辑器 IDLE。IDLE 的启动文件是 idle.bat，它位于 C:\Python34\Lib\idlelib 目录下。运行 idle.bat，即可打开文本编辑器 IDLE，如图 1-13 所示。也可以在开始菜单的所有程序中，选择 Python 3.4 分组下面的 IDLE（Python 3.4 GUI - 32 bit）菜单项，打开 IDLE 窗口。

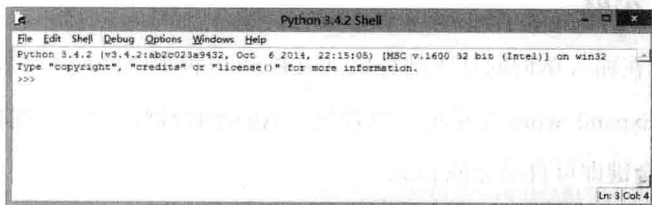


图 1-13 文本编辑器 IDLE

稍微有点遗憾的是，IDLE 没有汉化版。不过对于学习 Python 编程的读者来说，IDLE 菜单里的这点英文很简单。