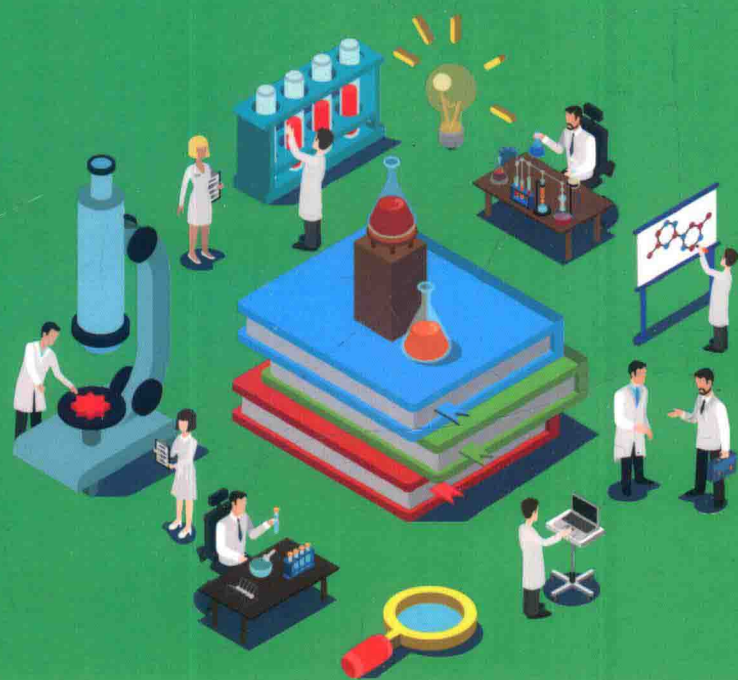


Python

编程基础

Python Programming



极客学院 出品

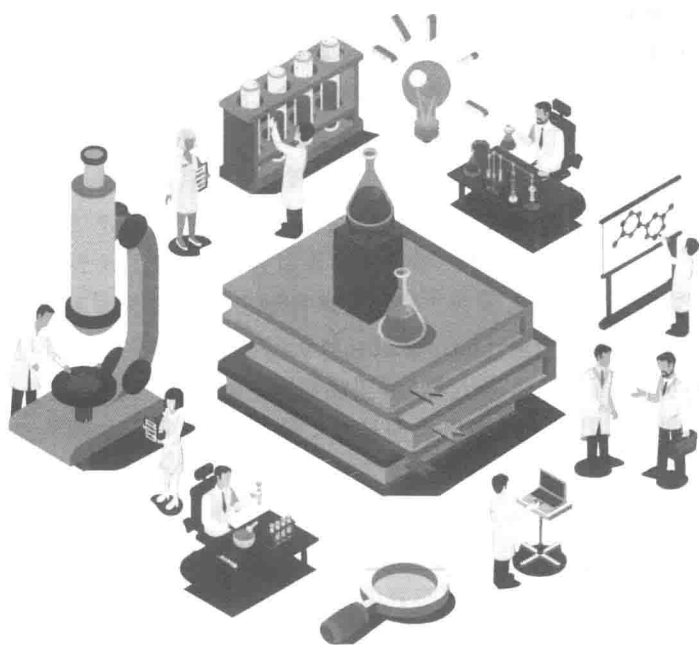
闫俊伢 主编 夏玉萍 陈实 荣宏 王文晶 副主编

- ◆ 45 个在线微课视频配合图书同步讲解，难点不再有，步骤跟着走
- ◆ Python 网络编程 + 数据库编程 + MVC 开发案例
- ◆ 提供全部案例的源代码、PPT 课件

Python

编程基础

Python Programming



极客学院 出品

闫俊伢 主编 夏玉萍 陈实 荣宏 王文晶 副主编

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Python编程基础 / 闫俊伢主编. — 北京 : 人民邮电出版社, 2016. 10
ISBN 978-7-115-43414-2

I. ①P… II. ①闫… III. ①软件工具—程序设计
IV. ①TP311.561

中国版本图书馆CIP数据核字(2016)第196784号

内 容 提 要

本书全面介绍了 Python 编程的基础知识和实用技术, 内容包括: Python 概述、Python 语言基础、常用 Python 语句、Python 函数、Python 模块、I/O 编程、使用 Python 程序控制计算机、Python 数据结构、多任务编程、Python 网络编程、Python 数据库编程、Web 框架开发以及附录实验。

本书既可以作为大学本、专科“应用程序设计”课程的教材, 也可以作为 Web 开发人员的参考用书。

-
- ◆ 主 编 闫俊伢
 - 副 主 编 夏玉萍 陈 实 荣 宏 王文晶
 - 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>
固安县铭成印刷有限公司印刷

 - ◆ 开本: 787×1092 1/16
印张: 18.25 2016 年 10 月第 1 版
字数: 479 千字 2016 年 10 月河北第 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

目 录

第一部分 基础篇

第 1 章 Python 概述	2	2.2.1 简单数据类型转换	26
1.1 初识 Python	2	2.2.2 列表	28
1.1.1 什么是 Python	2	2.2.3 元组	34
1.1.2 Python 的特点	4	2.2.4 字典	36
1.1.3 Python 各版本之间的差异	5	2.2.5 集合	39
1.2 开始 Python 编程	5	2.3 运算符和表达式	45
1.2.1 下载和安装 Python	5	2.3.1 运算符	45
1.2.2 执行 Python 脚本文件	6	2.3.2 表达式	48
1.2.3 Python 语言的基本语法	7	2.4 Python 对象	49
1.2.4 下载和安装 Pywin32	7	2.4.1 面向对象程序设计思想概述	49
1.3 Python 自带文本编辑器 IDLE 的 使用方法	9	2.4.2 定义和使用类	50
1.3.1 打开 IDLE	9	2.4.3 类的静态成员	53
1.3.2 新建 Python 脚本	9	2.4.4 类方法	54
1.3.3 保存 Python 脚本	9	2.4.5 使用 isinstance() 函数判断对象 类型	55
1.3.4 打开 Python 脚本	10	2.4.6 类的继承和多态	56
1.3.5 语法高亮功能	10	2.4.7 对象的序列化	59
1.3.6 自动完成功能	10	2.4.8 对象的赋值	62
1.3.7 语法提示功能	10	本章练习	62
1.3.8 运行 Python 程序	11	第 3 章 常用 Python 语句	64
1.3.9 IDLE 的菜单项	11	3.1 赋值语句	64
1.4 流行的 Python 集成开发环境 PyCharm	13	3.1.1 通过赋值语句实现序列解包	64
1.4.1 下载和安装 PyCharm	13	3.1.2 链式赋值	65
1.4.2 PyCharm 的使用方法	14	3.2 控制语句	65
本章练习	17	3.2.1 条件分支语句	65
第 2 章 Python 语言基础	19	3.2.2 循环语句	68
2.1 常量和变量	19	3.3 异常处理语句	70
2.1.1 常量	19	本章练习	71
2.1.2 Python 中数的类型与字符串	21	第 4 章 Python 函数	73
2.1.3 变量	24	4.1 声明和调用函数	73
2.2 数据类型	26	4.1.1 声明函数	73
		4.1.2 调用函数	74

4.2 参数和返回值.....	75	5.2.2 platform 模块.....	104
4.2.1 在函数中传递参数.....	75	5.2.3 与数学有关的模块.....	107
4.2.2 函数的返回值.....	80	5.2.4 time 模块.....	112
4.3 全局变量和局部变量.....	80	5.3 自定义和使用模块.....	114
4.3.1 变量的作用域.....	80	5.3.1 创建自定义模块.....	114
4.3.2 在 IDLE 的调试窗口中查看变量的值.....	81	5.3.2 使用自定义模块.....	114
4.3.3 在 PyCharm 的调试窗口中查看变量的值.....	83	本章练习.....	115
4.4 常用 Python 内置函数.....	84	第 6 章 I/O 编程.....	116
4.4.1 数学运算函数.....	84	6.1 输入和显示数据.....	116
4.4.2 字符串处理函数.....	85	6.1.1 输入数据.....	116
4.5 函数式编程.....	88	6.1.2 输出数据.....	117
4.5.1 函数式编程概述.....	89	6.2 文件操作.....	119
4.5.2 Python 函数式编程常用的函数.....	90	6.2.1 打开文件.....	119
4.5.3 普通编程方式与函数式编程的对比.....	94	6.2.2 关闭文件.....	120
4.6 闭包和递归函数.....	95	6.2.3 读取文件内容.....	120
4.6.1 闭包.....	95	6.2.4 向文件中写入数据.....	122
4.6.2 递归函数.....	95	6.2.5 文件指针.....	123
4.7 迭代器和生成器.....	96	6.2.6 截断文件.....	124
4.7.1 迭代器.....	96	6.2.7 文件属性.....	125
4.7.2 生成器.....	97	6.2.8 复制文件.....	126
本章练习.....	97	6.2.9 移动文件.....	126
第 5 章 Python 模块.....	100	6.2.10 删除文件.....	127
5.1 模块的基本概念.....	100	6.2.11 重命名文件.....	127
5.1.1 什么是模块.....	100	6.3 目录编程.....	127
5.1.2 如何导入模块.....	101	6.3.1 获取当前目录.....	127
5.2 Python 标准库中的常用模块.....	102	6.3.2 获取目录内容.....	127
5.2.1 sys 模块.....	102	6.3.3 创建目录.....	128
		6.3.4 删除目录.....	128
		本章练习.....	128

第二部分 高级篇

第 7 章 使用 Python 程序控制计算机..... 130

7.1 远程控制实例的需求分析.....	130	命令.....	133
7.2 CMD 命令.....	131	7.3 电子邮件编程.....	135
7.2.1 执行 CMD 命令.....	131	7.3.1 SMTP 编程.....	136
7.2.2 在 Python 程序中执行 CMD		7.3.2 POP 编程.....	139

7.4 Python 远程操控计算机的实例.....	143
7.4.1 发送指令端程序.....	143
7.4.2 接收指令端程序.....	144

本章练习	145	10.1.2 TCP/IP 协议簇体系结构	186
第 8 章 Python 数据结构	146	10.2 Socket 编程	188
8.1 Python 数据结构概述	146	10.2.1 Socket 的工作原理和基本概念	188
8.1.1 什么是数据结构	146	10.2.2 基于 TCP 的 Socket 编程	189
8.1.2 数据结构和算法的关系	147	10.2.3 基于 UDP 的 Socket 编程	193
8.2 栈	147	本章练习	194
8.2.1 栈的工作原理	147	第 11 章 Python 数据库编程	196
8.2.2 利用 Python 列表实现栈的数据 结构	148	11.1 数据库技术基础	196
8.3 队列	150	11.1.1 数据库的基本概念	196
8.3.1 队列的工作原理	150	11.1.2 关系数据库	198
8.3.2 利用 Python 列表实现队列的数据 结构	151	11.2 SQLite 数据库	198
8.4 树	153	11.2.1 下载和安装 SQLite 数据库	198
8.4.1 树的工作原理	153	11.2.2 创建 SQLite 数据库	199
8.4.2 遍历二叉树	154	11.2.3 数据类型	199
8.4.3 在 Python 程序中实现树的数据 结构	154	11.2.4 创建表	200
8.5 链表	157	11.2.5 向表中添加列	202
8.5.1 链表的工作原理	157	11.2.6 向表中插入数据	202
8.5.2 利用 Python 实现单向链表的 数据结构	158	11.2.7 修改表中的数据	203
本章练习	161	11.2.8 删除数据	204
第 9 章 多任务编程	162	11.2.9 查询数据	204
9.1 多进程编程	162	11.2.10 在 Python 中访问 SQLite 数据库	205
9.1.1 进程的概念	162	11.3 MySQL 数据库	207
9.1.2 进程的状态	163	11.3.1 安装 MySQL 数据库	207
9.2 进程编程	163	11.3.2 MySQL-Front	209
9.2.1 创建进程	163	11.3.3 创建数据库	210
9.2.2 枚举系统进程	167	11.3.4 删除数据库	211
9.3 多线程编程	170	11.3.5 MySQL 数据类型	211
9.3.1 线程的概念	170	11.3.6 创建表	212
9.3.2 threading 模块	171	11.3.7 编辑和查看表	215
本章练习	183	11.3.8 删除表	215
第 10 章 Python 网络编程	185	11.3.9 插入数据	215
10.1 网络通信模型和 TCP/IP 协议簇	185	11.3.10 修改数据	217
10.1.1 OSI 参考模型	185	11.3.11 删除数据	218
		11.3.12 使用 SELECT 语句查询数据	218
		11.3.13 在 Python 中访问 MySQL 数据库	223
		本章练习	226

第 12 章 Web 框架开发 227

12.1	Web 应用程序设计与开发概述	227
12.1.1	Web 应用程序设计语言的产生与发展	227
12.1.2	Web 应用程序的工作原理	228
12.2	HTML 概述	229
12.2.1	HTML 网页基本结构	229
12.2.2	设置网页背景和颜色	229
12.2.3	设置字体属性	231
12.2.4	超级链接	232
12.2.5	图像和动画	234
12.2.6	表格	234
12.2.7	使用框架	235
12.2.8	其他常用标签	237
12.3	Web 开发框架介绍	239
12.3.1	什么是框架	240
12.3.2	Web 开发框架	240
12.3.3	Python 中的 Web 框架	241
12.4	Django 框架	242
12.4.1	MVC 编程模式	242

12.4.2	下载和安装 Django 框架	243
12.4.3	创建和管理 Django 项目	244
12.4.4	Django 视图	246
12.4.5	Django 模板语法及使用	249
12.4.6	Django 模型	251
12.4.7	Django 表单	255
	本章练习	261

附录 实验 262

实验 1	开始 Python 编程	262
实验 2	Python 语言基础	264
实验 3	常用 Python 语句	267
实验 4	Python 函数	268
实验 5	Python 模块	270
实验 6	I/O 编程	271
实验 7	使用 Python 程序控制计算机	273
实验 8	Python 数据结构	275
实验 9	多任务编程	276
实验 10	网络编程	277
实验 11	Python 数据库编程	278
实验 12	Web 框架开发	280

第一部分

基础篇

- 第1章 Python 概述
- 第2章 Python 语言基础
- 第3章 常用 Python 语句
- 第4章 Python 函数
- 第5章 Python 模块
- 第6章 I/O 编程

第 1 章

Python 概述

学前提示

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。

知识要点

- 什么是 Python
- 下载和安装 Python
- Python 语言的基本语法
- PyCharm
- Python 的特性
- 执行 Python 脚本文件
- Python 文本编辑器 IDLE 的使用方法

1.1 初识 Python

首先来了解一下什么是 Python，它又有哪些特性。

极客学院

jikexueyuan.com



极客学院在线视频学习网址：

<http://www.jikexueyuan.com/course/776.html>

手机扫描二维码



Python 概述

1.1.1 什么是 Python

Python 于 20 世纪 80 年代末由荷兰人 Guido van Rossum（如图 1-1 所示）设计实现。他后来去了 Google 工作。据说在他给 Google 的简历里面只有简单的 3 个单词：“I wrote Python.”

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和类，第 4 章介绍函数。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。1.0 版本新增了函数式工具。关于函数式编程将在本书第 4 章中介绍。

Python 2.0 集成了列表推导式 (List Comprehension)。

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来减少特性的重复。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 to 3 的转换工具无缝地迁移到 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 开发语言排行榜吧。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(但不能说明一门编程语言好与不好)。该指数可以用来衡量开发者的编程技术能否跟上趋势以及应该及时掌握哪门编程语言。TIOBE 的官方网址如下:

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

2015 年 9 月的 TIOBE 排行榜显示,Python 排名第 5,如图 1-2 所示。



图 1-1 Guido van Rossum

Sep. 2015	Sep 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	19.565%	+5.43%
2	1	▼	C	15.621%	-1.10%
3	4	▲	C++	6.782%	+2.11%
4	5	▲	C#	4.909%	+0.56%
5	8	▲	Python	3.864%	+0.88%
6	7	▲	PHP	2.530%	-0.59%
7	9	▲	JavaScript	2.342%	-0.11%
8	11	▲	Visual Basic .NET	2.062%	+0.53%
9	12	▲	Perl	1.899%	+0.53%
10	3	▼	Objective-C	1.821%	-8.11%
11	29	▲	Assembly language	1.806%	+1.22%
12	13	▲	Ruby	1.783%	+0.50%
13	15	▲	Delphi/Object Pascal	1.745%	+0.59%

图 1-2 2015 年 9 月的 TIOBE 排行榜

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

1.1.2 Python 的特点

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

(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，可以很轻松地上手。

(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，从而使其不断完善。

(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，例如内存管理。

(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 程序无需进行任何修改，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。

(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。Python 程序的运行过程如图 1-3 所示。

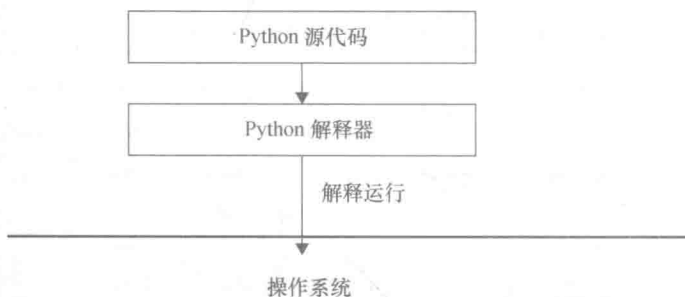


图 1-3 Python 程序的运行过程

(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本书第 2 章中介绍。

(7) 高可扩展性：如果希望一段代码可以很快地运行，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 或 C++ 程序中，从而为 C 或 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，本书后续章节将会介绍这些库的具体情况。

1.1.3 Python 各版本之间的差异

在 Python 的官方网站上，同时提供了 2.0 系列和 3.0 系列 2 个版本。这 2 个版本的差异如表 1-1 所列。

表 1-1 Python 2.0 系列和 Python 3.0 系列的差异

项目	Python 2.0	Python 3.0
使用范围	更广泛	较不广泛
支持的包数量	较多	较少
客人维护性	可能不更新	更面向未来

由于历史的原因，Python 2.0 推出得更早，支持它的库就更多一些。由于一些第三方库已经停止了更新，因此很多第三方库不再支持 Python 3.0。当然，越来越多的第三方库正在向 Python 3.0 迁移，Python 3.0 的应用也比较广泛，只是与 Python 2.0 相比要少一些。

从可维护性的角度讲，Python 基金会已经明确表示：Python 2.0 将不再被更新，很可能只是更新到 2.7 版本。而 Python 3.0 将持续更新下去。因此 Python 3.0 更面向未来。

本书内容基于 Python 2.7.10。

1.2 开始 Python 编程

本节将配置 Python 开发的环境，并介绍一个简单的 Python 程序。通过本节的学习，读者可以开始 Python 编程。

1.2.1 下载和安装 Python

访问如下网址可以下载 Python，如图 1-4 所示。

<https://www.python.org/downloads/>

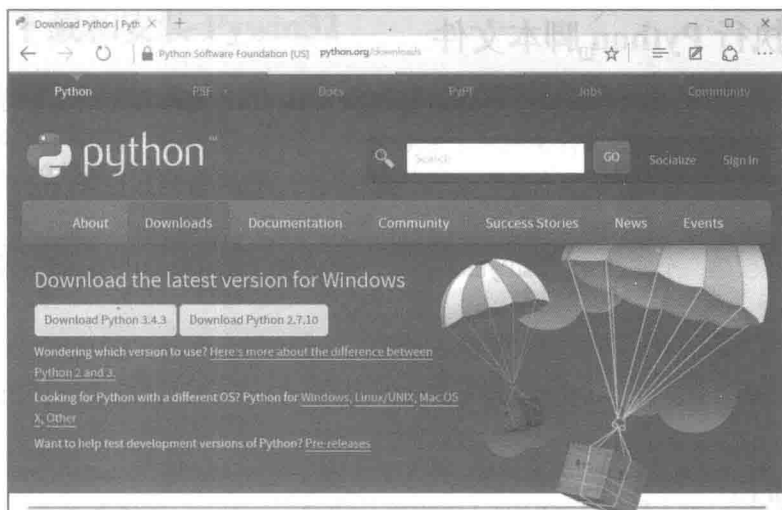


图 1-4 下载 Python

在笔者编写本书时,Python for Windows 有 2 个最新版本,2.0 系列的最新版本为 Python 2.7.10,3.0 系列的最新版本为 Python 3.4.3。读者看到的情况也许会略有不同。

单击 Download Python 2.7.10 按钮,下载得到 python-2.7.10.msi。双击 python-2.7.10.msi,即可按照向导安装 Python 2.7.10。

Python 2.7.10 的默认安装目录为 C:\Python27。安装完成后,将 C:\Python27 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,这里不具体介绍。

在 Windows 7 中安装后,“开始”菜单的“所有程序”中会出现一个 Python 2.7 分组。单击其下的 Python (command line)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。

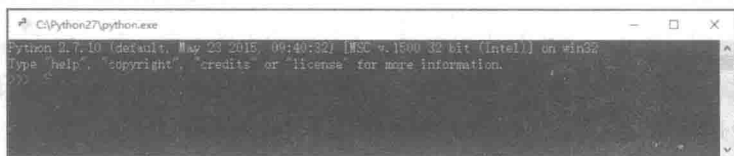


图 1-5 Python 2.7.10 安装成功后打开 Python 命令窗口

Python 命令实际上就是 Python 的解释器。在>>>后面输入 Python 程序,回车后即可被解释执行。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。

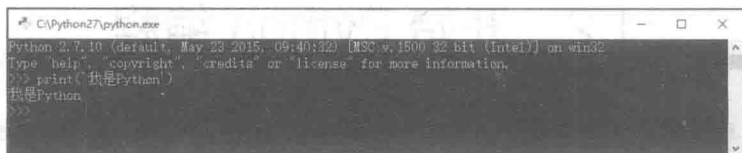


图 1-6 打印“我是 Python”的结果

```
print('我是 Python')
```

print()函数用于输出数据。关于函数的具体情况将在本书第 4 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

第 1.2.1 节介绍了在命令行里执行 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
print('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。'I am Python'是一个字符串，关于字符串的具体情况将在本书第 2 章中介绍。

1.2.3 Python 语言的基本语法

本节介绍 Python 语言的基本语法，这些都是编写 Python 程序需要了解和注意的。

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 程序中也可以根据编写人员的个人习惯在语句后面使用分号 (;)。

也可以把多个语句写在一行，此时则需要在语句后面加上分号 (;) 表示结束。例如：

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

2. 缩进

缩进指在代码行前面添加空格或 Tab，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 1-2】 语句缩进的例子。

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

运行这段程序的结果如下：

```
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，可以兼容很多平台。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 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。

访问下面的网址可以下载 Pywin32 安装包。

```
http://sourceforge.net/projects/pywin32/
```

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。

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面，如图 1-10 所示。

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 Python 2.7.10，因此单击 pywin32-219.win32-py2.7.exe 超链接，可以下载得到 Pywin32 的安装包 pywin32-219.win32-py3.4.exe。



图 1-7 Pywin32 项目主页

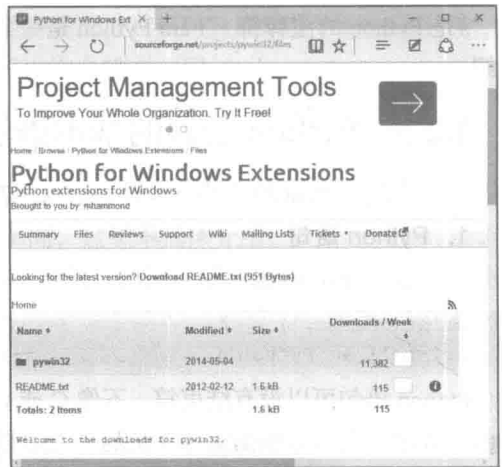


图 1-8 选择产品页面

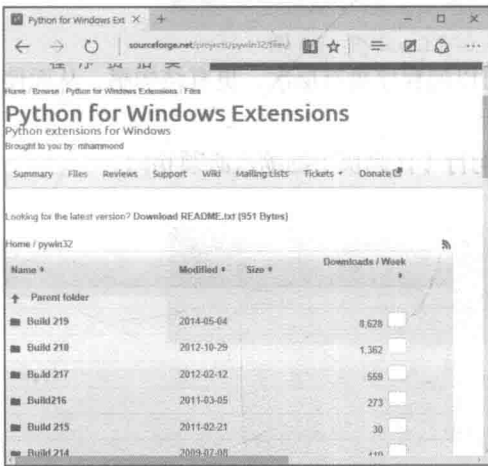


图 1-9 选择 Pywin32 版本

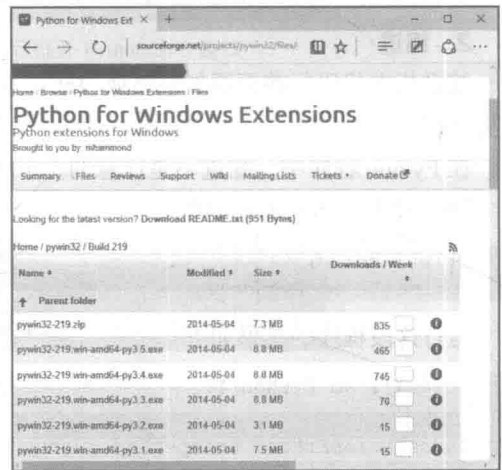


图 1-10 下载文件列表页面



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-py2.7.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。

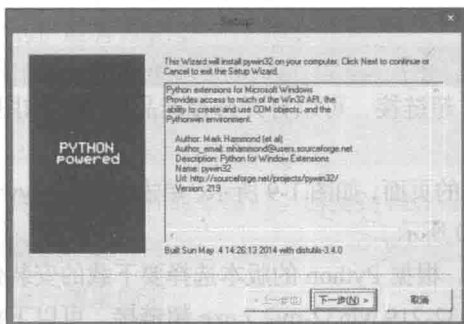


图 1-11 欢迎窗口

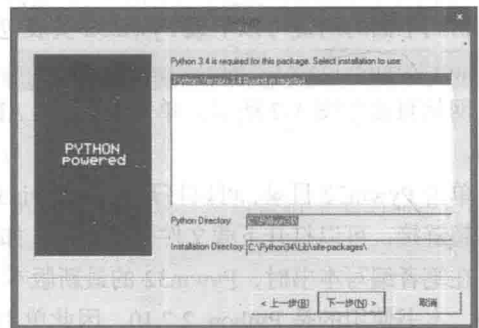


图 1-12 选择目录窗口

安装程序会从注册表中自动获取 Python 2.7 的安装目录（如 C:\Python27），默认的 Pywin32 安装目录是 C:\Python27\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 自带文本编辑器 IDLE 的使用方法

Python 是一种脚本语言，它并没有提供一个官方的开发环境，用户可以使用 Python 自带的编辑器 IDLE，也可以自主选择编辑工具。

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 2.7 分组下面的 IDLE（Python GUI）菜单项，打开 IDLE 窗口。

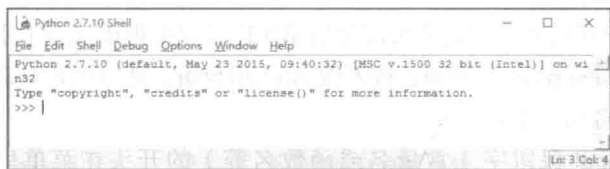


图 1-13 文本编辑器 IDLE

稍微有点遗憾的是，IDLE 没有汉化版本。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。

1.3.2 新建 Python 脚本

在菜单里依次选择 File/New File（或按 Ctrl+N 组合键）即可新建 Python 脚本，窗口标题显示脚本名称，初始时为 Untitled，也就是还没有保存 Python 脚本。如图 1-14 所示。

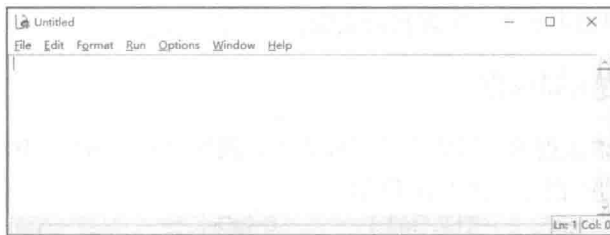


图 1-14 新建 Python 脚本的窗口

1.3.3 保存 Python 脚本

在菜单里依次选择 File/Save File（或按 Ctrl+S 组合键）即可保存 Python 脚本。如果是第一次保存，则会弹出保存文件对话框，要求用户输入保存的文件名。