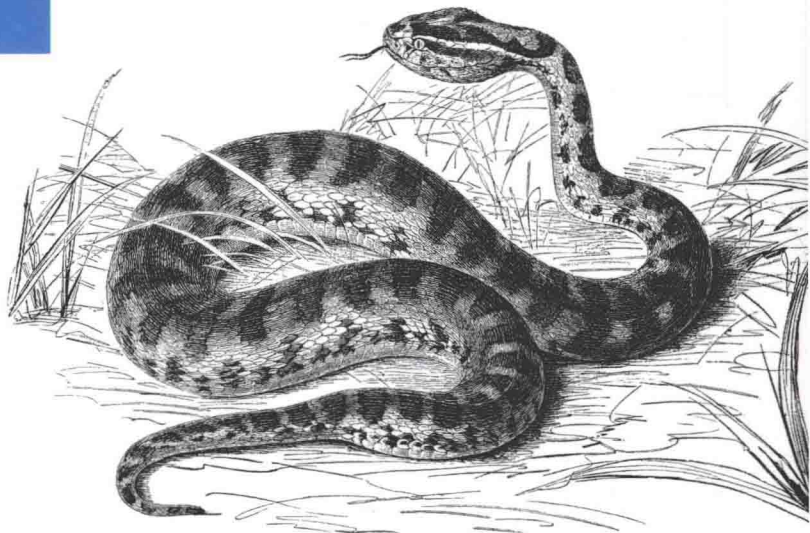




工业和信息化“十三五”
人才培养规划教材



Python 3 基础教程

Python 3 Course

邓英 夏帮贵 ◎ 主编



零基础入门，快速掌握 **Python** 的基本技术和使用方法
应用能力培养为核心，**内容精心编排**，包含大量**编程实践案例**
教学资源丰富，含 **PPT 课件、教学大纲、教案、教学进度表、习题答案**



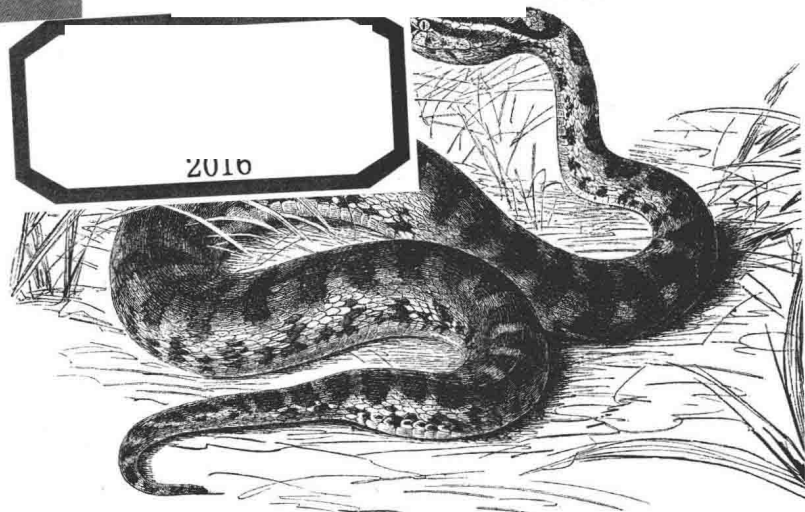
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化“十三五”
人才培养规划教材



Python 3 基础教程

Python 3 Course

邓英 夏帮贵 © 主编

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Python 3基础教程 / 邓英, 夏帮贵主编. -- 北京 :
人民邮电出版社, 2016. 10
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-43487-6

I. ①P… II. ①邓… ②夏… 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(2016)第207987号

内 容 提 要

本书注重基础、循序渐进,系统地讲述了 Python 程序设计开发的相关知识。全书共分 8 章,涵盖了 Python 开发环境及工具、编程基础、程序流程控制、函数与模块、面向对象编程、异常处理、数据库编程和 tkinter GUI 编程等内容。

本书内容丰富,讲解详细,适用于初、中级 Python 用户,可作为各类院校计算机相关专业教材,也可作为 Python 爱好者的参考书。

◆ 主 编 邓 英 夏帮贵

责任编辑 桑 珊

执行编辑 左仲海

责任印制 沈 蓉 焦志炜

◆ 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

2016 年 10 月第 1 版

字数: 439 千字

2016 年 10 月河北第 1 次印刷

定价: 42.00 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

FOREWORD

Python 因其功能强大、简单易学、开发成本低廉，已成为深受广大应用开发人员喜爱的程序设计语言之一。作为一门优秀的程序设计语言，Python 被广泛应用到各种领域，从简单的文字处理，到网站和游戏开发，甚至机器人和航天飞机控制，都可以找到 Python 的身影。

本书特别针对 Python 零基础编程爱好者，进行了内容编排和章节组织，争取让读者在短时间内掌握 Python 的基本技术和方法。

本书具有如下特点。

1. 零基础入门

读者即使没有其他程序设计语言的相关基础，跟随本书也可轻松掌握 Python 的各种基本技术和使用方法。

2. 学习成本低

本书在构建开发环境时，选择了应用最为广泛的 Windows 操作系统、稳定版 Python 3.5，使用 Python 3.5 自带的集成开发工具 IDLE 等进行学习和操作，没有特别的软件和硬件要求。

3. 内容编排精心设计

Python 程序设计涉及的范围非常广泛，本书内容编排并不求全、求深，而是考虑零基础读者的接受能力，选择 Python 中必备、实用的知识进行讲解。知识和配套实例循序渐进、环环相扣。

4. 强调理论与实践相结合

书中每章末尾的编程实践环节都尽量安排一个短小、完整的实例，方便教师教学，也方便学生学习。

5. 完整收集学习必备资源

为了方便读者学习，本书提供所有实例的源代码、数据库文件以及资源。源代码可在学习过程中直接使用，参考相关章节进行配置即可。

Python 3 基础教程

本书主要内容如下所示。

章节	主要内容
第 1 章	认识 Python、Python 程序运行方式、Python 开发环境和工具
第 2 章	Python 程序基本结构、基本输入和输出、数据类型（数字、集合、字符串、列表、元组、字典和文件）
第 3 章	if 分支结构、for 循环、while 循环、迭代和列表解析
第 4 章	函数、变量作用域、模块、模块包
第 5 章	理解 Python 的面向对象、定义和使用类、对象的属性和方法、类的继承、运算符重载、模块中的类
第 6 章	Python 异常处理机制、主动引发异常、自定义异常类
第 7 章	了解关系数据库、访问 SQLite 数据库、访问 MySQL 数据库、访问 Microsoft SQL Server 数据库
第 8 章	tkinter 编程基础、tkinter 组件、对话框

本书由西华大学邓英、夏帮贵主编，夏帮贵编写第 1~3 章，邓英编写第 4~8 章并负责全书统稿。同时赖霞、刘凡新、雷鸣彬、邵军、宋玉忠、陈利娅、陈子春、刘鹏惠、申绪英等人也参与了本书的编写。本书源代码、PPT 等相关资源可登录人民邮电出版社教育社区（www.ryjiaoyu.com）免费下载。

由于编者水平有限，书中难免存在不妥之处，敬请广大读者批评指正。

编者

2016 年 7 月



目 录

CONTENTS

第 1 章 Python 起步

1

- 1.1 认识 Python 1
 - 1.1.1 Python 的发展 1
 - 1.1.2 Python 的应用案例 2
 - 1.1.3 Python 的应用领域 2
 - 1.1.4 Python 的特点 4
 - 1.1.5 Python 的版本 6
 - 1.1.6 Python 3.x 与 2.x 的区别 7
- 1.2 Python 程序的运行方式 9
 - 1.2.1 计算机程序设计语言分类 10
 - 1.2.2 计算机程序的运行方式 10
 - 1.2.3 Python 程序的运行方式 10
 - 1.2.4 Python 的实现 11
 - 1.2.5 Python 程序的可执行文件 11
- 1.3 Python 开发环境及工具 12
 - 1.3.1 Python 的下载和安装 12
 - 1.3.2 Python 编程工具: IDLE 15
 - 1.3.3 Python 编程工具: 记事本 16
 - 1.3.4 Python 编程工具: Eclipse+PyDev 17
 - 1.3.5 Python 编程工具: NetBeans 26
- 1.4 编程实践 31
 - 1.4.1 使用 IDLE 交互模式 31
 - 1.4.2 直接使用 Python 交互模式 32
 - 1.4.3 用 IDLE 编写 Python 程序输出 32
 - 1.4.4 在命令行运行 Python 程序 33
 - 1.4.5 双击文件图标运行 Python 程序 34
- 1.5 本章小结 34
- 1.6 习题 35

第 2 章 Python 编程基础

36

- 2.1 Python 程序基本结构 36
 - 2.1.1 用缩进表示代码块 36
 - 2.1.2 代码注释 37
 - 2.1.3 语句续行 37
 - 2.1.4 语句分隔 37
 - 2.1.5 关键字与大小写 38
- 2.2 基本输入和输出 38
 - 2.2.1 基本输入 38
 - 2.2.2 基本输出 39
- 2.3 数据类型: 数字 40
 - 2.3.1 数字常量 40
 - 2.3.2 数字运算 42

2.3.3 小数	45	2.7.2 列表基本操作	69
2.3.4 分数	46	2.7.3 常用列表方法	72
2.3.5 数学函数	47	2.8 数据类型：元组	74
2.4 变量与动态数据类型	49	2.8.1 元组特点	74
2.4.1 变量与对象	49	2.8.2 元组的基本操作	74
2.4.2 对象的垃圾回收	49	2.8.3 元组方法	76
2.4.3 变量命名规则	50	2.9 数据类型：字典	77
2.4.4 赋值语句	50	2.9.1 字典的特点	77
2.4.5 变量的共享引用	52	2.9.2 字典基本操作	77
2.5 数据类型：集合	53	2.9.3 字典常用方法	79
2.5.1 集合常量	53	2.9.4 字典视图	81
2.5.2 集合运算	54	2.10 数据类型：文件	83
2.5.3 集合基本操作	54	2.10.1 打开和关闭文件	83
2.5.4 冻结集合	55	2.10.2 读写文本文件	83
2.6 数据类型：字符串	56	2.10.3 读写二进制文件	88
2.6.1 字符串常量	56	2.10.4 用文件存储 Python	
2.6.2 字符串基本操作	58	对象	88
2.6.3 字符串方法	61	2.11 编程实践：用户数据的	
2.6.4 字符串格式化表达式	66	序列化	89
2.6.5 bytes 字符串	68	2.12 本章小结	90
2.7 数据类型：列表	69	2.13 习题	90
2.7.1 列表基本特点	69		

第 3 章 程序流程控制

91

3.1 if 分支结构	91	3.3.2 嵌套使用 while 循环	97
3.1.1 if 语句基本结构	91	3.4 迭代和列表解析	98
3.1.2 真值测试	92	3.4.1 迭代	98
3.1.3 if...else 三元表达式	94	3.4.2 列表解析	100
3.2 for 循环	94	3.4.3 zip、map 和 filter	102
3.2.1 for 循环基本格式	94	3.5 编程实践	103
3.2.2 多个变量迭代	95	3.5.1 输出数字金字塔	103
3.2.3 break 和 continue	96	3.5.2 生成 10 个两位的	
3.2.4 嵌套使用 for 循环	96	随机素数	104
3.3 while 循环	96	3.6 本章小结	104
3.3.1 while 循环基本结构	97	3.7 习题	105

第 4 章 函数与模块

106

4.1 函数	106	4.3.6 嵌套导入模块	122
4.1.1 定义函数	106	4.3.7 查看模块对象属性	122
4.1.2 函数调用	107	4.3.8 <code>__name__</code> 属性和 命令行参数	123
4.1.3 函数参数	107	4.3.9 隐藏模块数据	124
4.1.4 函数嵌套定义	110	4.4 模块包	125
4.1.5 lambda 函数	110	4.4.1 包的基本结构	125
4.1.6 递归函数	110	4.4.2 导入包	125
4.1.7 函数列表	111	4.4.3 相对导入	125
4.2 变量作用域	111	4.5 编程实践：函数库	127
4.2.1 有哪些作用域	111	4.5.1 实现杨辉三角函数	127
4.2.2 global 语句	113	4.5.2 实现“汉诺塔” 模拟函数	128
4.2.3 nonlocal 语句	114	4.5.3 交互模式导入函数 测试	130
4.3 模块	114	4.6 本章小结	130
4.3.1 导入模块	115	4.7 习题	131
4.3.2 导入与执行模块	116		
4.3.3 用 import 还是 from	118		
4.3.4 重新载入模块	119		
4.3.5 模块搜索路径	120		

第 5 章 面向对象编程

132

5.1 理解 Python 的面向 对象	132	5.3.4 构造函数和析构函数	139
5.1.1 Python 的类	132	5.4 类的继承	139
5.1.2 Python 中的对象	133	5.4.1 简单继承	139
5.2 定义和使用类	133	5.4.2 定义子类的属性和方法	140
5.2.1 定义类	133	5.4.3 调用超类的构造函数	141
5.2.2 使用类	134	5.4.4 多重继承	141
5.3 对象的属性和方法	135	5.5 运算符重载	142
5.3.1 对象的属性	135	5.5.1 加法运算重载	143
5.3.2 对象的方法	137	5.5.2 索引和分片重载	143
5.3.3 类的“伪私有”属性和 方法	138	5.5.3 自定义迭代器对象	144
		5.5.4 定制对象的字符串 形式	146

5.6 模块中的类	148	5.7.2 系统功能实现	151
5.7 编程实践：用户注册信息 管理系统	148	5.8 本章小结	155
5.7.1 系统功能预览	149	5.9 习题	155

第6章 异常处理 157

6.1 Python 异常处理机制	157	6.2.2 异常链：异常引发异常	164
6.1.1 为什么使用异常处理	157	6.2.3 assert 语句	164
6.1.2 异常处理基本结构	159	6.3 自定义异常类	165
6.1.3 常见异常类型	160	6.3.1 异常类的超类	165
6.1.4 捕捉多个异常	160	6.3.2 创建自定义异常类	166
6.1.5 except...as 与统一 处理	160	6.3.3 用自定义异常类保存 异常日志	166
6.1.6 捕捉所有异常	161	6.4 编程实践：为用户注册信息 管理系统添加异常处理	167
6.1.7 异常处理结构的嵌套	162	6.5 本章小结	170
6.1.8 try...finally 终止行为	162	6.6 习题	170
6.2 主动引发异常	163		
6.2.1 raise 语句	163		

第7章 Python 数据库编程 171

7.1 了解关系数据库	171	7.2.6 使用 Row 对象	182
7.1.1 数据模型	171	7.2.7 修改记录	183
7.1.2 关系数据库的概念和 特点	172	7.2.8 删除记录	183
7.1.3 关系数据库语言 SQL	173	7.2.9 导入文件中的数据	184
7.2 访问 SQLite 数据库	179	7.3 访问 MySQL 数据库	186
7.2.1 了解 Python 的 SQL 接口	179	7.3.1 下载安装 MySQL	187
7.2.2 连接和创建 SQLite 数据库	179	7.3.2 访问 MySQL 数据库 实例	194
7.2.3 创建表	180	7.3.3 连接 MySQL 服务器	195
7.2.4 添加记录	180	7.3.4 MySQL 数据库操作	197
7.2.5 执行查询	181	7.3.5 MySQL 表操作	199
		7.3.6 MySQL 查询参数	201
		7.3.7 使用存储过程	201

7.3.8 使用事务	202	7.4.5 建立 SQL Server 服务器 连接	214
7.4 访问 Microsoft SQL Server 数据库	204	7.4.6 使用 ADO 的记录集对象	217
7.4.1 下载和安装 Python 的 Windows 扩展组件	204	7.4.7 使用 ADO 访问 Excel 文件	222
7.4.2 生成 ADO 的 Python 支持 文件	207	7.5 编程实践：基于数据库的 用户注册信息管理系统	224
7.4.3 安装 SQL Server	207	7.6 本章小结	228
7.4.4 快速生成 SQL Server 连接字符串	211	7.7 习题	229

第 8 章 tkinter GUI 编程

230

8.1 tkinter 编程基础	230	8.2.9 滚动条组件 Scrollbar	254
8.1.1 第一个 tkinter GUI 程序	230	8.2.10 标签框架 LabelFrame	256
8.1.2 组件打包	232	8.2.11 文本框组件 Text	257
8.1.3 添加按钮和事件处理 函数	234	8.2.12 画布组件 Canvas	259
8.1.4 Packer 布局	236	8.2.13 顶层窗口组件 Toplevel	261
8.1.5 Grid 布局	237	8.2.14 菜单组件 Menu	263
8.1.6 Place 布局	238	8.3 对话框	264
8.1.7 使用框架	239	8.3.1 消息对话框	264
8.2 tkinter 组件	239	8.3.2 文件对话框	266
8.2.1 组件通用属性设置	239	8.3.3 颜色对话框	269
8.2.2 标签组件 Label	242	8.4 编程实践：GUI 版的用户 注册信息管理系统	269
8.2.3 按钮组件 Button	244	8.4.1 系统功能预览	270
8.2.4 输入组件 Entry	245	8.4.2 显示全部已注册用户	270
8.2.5 列表框组件 Listbox	249	8.4.3 查找/修改/删除用户信息	271
8.2.6 复选框组件 Checkbutton	250	8.4.4 添加新用户	271
8.2.7 单选按钮组件 Radiobutton	252	8.4.5 系统功能实现	271
8.2.8 旋转框组件 Spinbox	253	8.5 本章小结	277
		8.6 习题	278



第 1 章 Python 起步

Python 是一种面向对象的、解释型的计算机高级程序设计语言，由 Guido van Rossum 于 1989 年开始开发，并于 1991 年发布第一个公开发行人版。Python 是纯粹的开源自由软件，其源代码和解释器均遵循 GPL(GNU General Public License)协议。Python 具有语法简洁、易于学习、功能强大、可扩展性强、跨平台等诸多特点，逐渐成为最受欢迎的程序设计语言之一。



本章要点

- Python 的发展
- Python 的特点
- Python 运行方式
- Python 开发环境配置
- Python 开发工具

1.1 认识 Python

欢迎来到 Python 的世界。本节将从 Python 的发展、应用、特点和版本等内容开始，带领读者了解和认识 Python。

1.1.1 Python 的发展

1989 年圣诞节，Guido 在阿姆斯特丹开始准备开发一种新的程序设计语言。作为 Monty Python 喜剧团体的粉丝，Guido 将这门新的语言命名为 Python。Python 的灵感来自 ABC 语言——Guido 参与开发的一种适用于非专业程序开发人员的教学语言。Python 认为 ABC 语言优美、功能强大，ABC 未获得成功主要是非开放造成的，所以，Guido 一开始将 Python 主要目标之一定位在开放性。同时，Guido 还想在 Python 中实现一些在 ABC 中闪过过但未曾实现的东西。可以说，Python 起源于 ABC，并受到了 Modula-3 语言的影响，同时结合了 UNIX Shell 和 C 语言的习惯。

经过多年的发展，Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月，TIOBE 将 Python 评为编程语言排行榜 2010 年度语言。TIOBE 网站地址为：http://www.tiobe.com/tiobe_index。图 1-1 显示了截至 2016 年 7 月的 TIOBE 程序设计语言排行榜，Python 在众多的程序设计语言中排在 Java、C、C++ 之后，处于第 4 位，其影响力可见一斑。

TIOBE Index for July 2016

July Headline: Assembly Language enters the Top 10 again

It might come as surprise that the lowest level programming language that exists has re-entered the TIOBE index top 10. Why would anyone write code at such a low level, being far less productive if compared to using any other programming language and being vulnerable to all kinds of programming mistakes? The only reasonable explanation for this is that the number of very small devices that are only able to run assembly code is increasing. Even your tooth brush or coffee machine are running assembly code nowadays. Another reason for adoption is performance. If performance is key, nobody can beat assembly code. Other interesting moves this month are: good old Tcl jumps from #65 to #48, CFML (ColdFusion) from #102 to #66 and Maple from #94 to #74.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the best programming language or the language in which most lines of code have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Jul 2016	Jul 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.804%	+2.08%
2	2		C	12.238%	-3.91%
3	3		C++	6.311%	-2.33%
4	5	▲	Python	4.156%	-0.09%
5	4	▼	C#	3.920%	-1.73%
6	7	▲	PHP	3.272%	+0.36%
7	9	▲	JavaScript	2.643%	+0.45%
8	6		Visual Basic .NET	2.517%	+0.09%
9	11	▲	Perl	2.428%	+0.62%
10	12	▲	Assembly language	2.281%	+0.75%

图 1-1 TIOBE 程序设计语言排行榜

详细了解 Python 的现状，请访问 Python 官方网站：<http://www.python.org>。

1.1.2 Python 的应用案例

目前，不少公司使用了 Python 来开发自己的产品。例如：

- 世界上最大的视频分享网站 YouTube 的视频分享服务大部分采用 Python 实现。
- 首个 P2P 文件分享软件 BitTorrent 也是采用 Python 实现。
- 大型多人在线游戏 EVE Online 广泛使用了 Python。
- Autodesk 旗下的著名三维建模和动画软件 Maya，为 Python 提供了 API (Application Programming Interface，应用程序编程接口)。

- Intel、Cisco、HP、Seagate、Qualcomm 和 IBM 等公司使用 Python 进行硬件测试。
- Pixar、Industrial Light & Magic 等电影公司使用 Python 制作动画电影。
- JPMorgan Chase、UBS、GETCO、Citadel 等公司使用 Python 做市场预测。
- NASA、Los Alamos、Fermilab、JPL 等使用 Python 执行科学计算。
- iRobot 使用 Python 开发了机器人真空吸尘器。
- ESRI 使用 Python 为终端用户定制 GIS 地图产品。
- NSA 应用 Python 实现加密和智能分析。
- IronPort 使用 Python 实现其电子邮件服务器产品。
- Google、Yahoo、搜狐邮箱、知乎、果壳等也广泛使用了 Python。

在 Python 官方网站的 Python Success Stories (<http://www.python.org/about/success/>) 中介绍了大量应用 Python 获得成功的公司和组织机构的典型案列。

1.1.3 Python 的应用领域

作为一门优秀的程序设计语言，Python 被广泛应用于各种领域，从简单的文字处理，到网站和游戏开发，甚至于机器人和航天飞机控制，都可以找到 Python 的身影。

1. 系统编程

Python 可以访问操作系统 API，为其编写可移植的系统维护和管理工具。Python 程序可以访问系统目录和文件，可以运行其他程序，也可编程进程和线程执行并行处理等。

Python 标准库集成了 POSIX (Portable Operating System Interface, 可移植操作系统接口) 和其他常见操作系统工具：环境变量、文件、套接字、管道、进程、多线程、正则表达式、命令行参数、标准流接口、Shell 目录启动器、文件扩展名等。大部分系统工具继承了 Python 的可移植特性。

2. GUI 编程

GUI 指 Graphical User Interface (图形用户界面)。使用 Python 可以非常简单、快捷地实现 GUI 程序。Python 内置了 Tkinter 的标准面向对象接口 Tk GUI API。应用 Tk GUI API 实现的 Python GUI 程序，可以不做任何改变运行在 Windows、X Window (UNIX 和 Linux) 和 Mac OS 等多种平台上。

还可以使用其他一些扩展包，如 PMW、wxPython GUI、PythonCard 和 Dabo 等，也可在 Python 中创建 GUI 应用。

3. Web 应用

Python 包含了标准 Internet 模块，可用于实现各种网络任务。Python 脚本可以通过套接字进行网络通信；可编写服务器 CGI 脚本处理客户端表单信息；可通过 FTP 传输文件；可以生成、解析和分析 XML 文件；可处理 Email；可通过 URL 获取网页；可从网页中解析 HTML 和 XML；可通过 XML-PRC、SOAP 和 Telnet 通信。

Python 也可用第三方工具进行 Web 应用开发。例如，HTMLgen 包可用于生成 HTML 文件，mod_python 可用于生成运行于 Apache 服务器上的 Python 应用并支持 Python Server Page，Jython 可以实现与 Java 无缝集成的 Applet。大量的第三方工具，使 Python 可用于开发完整的、企业级的 Web 应用。

4. 组件集成

Python 本身是用 C 实现的，所以可以使用 C/C++ 进行扩展。Python 也被称为“胶水”语言，既可集成各种 C/C++ 库，也可将 Python 嵌入到其他语言中。

例如，PythonCOM 是 Windows 支持的 COM，Jython 是基于 Java 平台的 Python 系统，基于 .NET 平台可实现 IronPython 和各种 CORBA 工具包。利用组件包，可以编写 Python 脚本，在 Windows 中处理微软的 Word 和 Excel 文件。

5. 数据库编程

Python 提供了访问各种主流数据库的 API，包括 Microsoft SQL Server、Oracle、Sybase、DB2、MySQL、SQLite 等。Python 还自带一个 Gadfly 模块，提供了一个完整的 SQL 环境。从 Python 2.5 起，SQLite 成为了 Python 内置标准库的一部分。

6. 数值计算和科学计算

Python 的 NumPy 扩展提供了大量的标准数学库的接口，例如矩阵对象、标准数学库等。SciPy 和 Matplotlib 扩展也为 Python 提供了快速数组处理、数值运算以及绘图功能。众多的

扩展库使 Python 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。

相比于著名的科学计算商业软件 MATLAB, Python 是完全免费的,而且是一门更易学、更严谨的程序设计语言。MATLAB 主要专注于工程和科学计算,而 Python 通过各种扩展库,可以在满足计算的同时,实现文件管理、界面设计、网络通信等各种高级任务。

7. 游戏、图像、人工智能、机器人、XML 等其他领域

Python 的应用领域非常多,其中包括:

- 使用 Pygame 扩展包进行图形和游戏应用开发。
- 使用 PySerial 扩展包在 Windows、Linux 或其他系统上开发串口通信应用。
- 使用 PIL、PyOpenGL、Blender、Maya 和其他扩展包开发图形或 3D 应用。
- 使用 PyRo 扩展包开发机器人控制程序。
- 使用 XML 库、xmlrpclib 模块或第三方扩展包解析 XML。
- 使用 PyBrain 扩展包开发人工智能应用。
- 使用 NLTK 扩展包开发自然语言分析应用。



提示: Python 的应用精彩纷呈,可以从 Python 官网了解更多信息,也可从 IBM 的“developerWorks 中国”网站的“Python 技术专题”(<http://www.ibm.com/developerworks/cn/linux/theme/python/>) 了解 Python 的各种应用。

1.1.4 Python 的特点

Python 具有下列显著特点。

1. Python 是免费的开源自由软件

Python 遵循 GPL 协议,也是免费的,不管是用于个人还是商业用途,开发人员都无须支付任何费用,也不用担心版权问题。

作为开源软件,程序员可以获得 Python 源代码,以研究其内部细节,并可加以修改使其针对目标更加适用。也可以将 Python 嵌入系统或随产品一起发布,甚至于销售 Python 的源代码,都没有任何限制。

2. Python 是面向对象的

面向对象 (Object Oriented, OO) 是现代高级程序设计语言的一个重要特征。多态、运算符重载、继承和多重继承等面向对象编程 (Object Oriented Programming, OOP) 的主要特征也在 Python 的类模块中得到很好的支持。得益于 Python 简洁的语法和数据类型系统,Python 中的 OOP 也变得极为简单,比其他语言容易。

OOP 是 Python 的一个重要特征,初学者也不必为此感到担心。Python 同样支持传统的面向过程的编程模式,完全可以在具有一定基础之后再深入学习 Python 的 OOP。

3. Python 具有良好的跨平台特性

Python 是用 ANSI C 实现的。C 语言因为跨平台和良好的可移植性成为了经典的程序设计语言。这意味着 Python 也具有有良好的跨平台特性,可在目前所有的主流平台上

编译和运行。所以，在 Windows 下编程的 Python 程序，可以轻松地在 Linux 等其他系统中运行。

因为 Python 是开源的，可以轻松将其移植到各种不同平台，包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE、PocketPC、Symbian、Android 和 iOS 等平台。

4. Python 功能强大

Python 既属于脚本语言，也属于高级程序设计语言，所以，Python 具有脚本语言（如 Perl、Tcl 和 Scheme 等）的简单、易用特点，也具有高级程序设计语言（如 C、C++ 和 Java 等）的强大功能。



提示：提到脚本语言容易让人想到 JavaScript 和 VBScript 等网页脚本编程语言，Python 的功能远远不是普通脚本语言能够比拟的。

Python 具有的一些强大功能如下。

- **动态数据类型**：Python 在代码运行过程中跟踪变量的数据类型，不需要在代码中声明变量的类型，也不要求在使用之前对变量进行类型声明。
- **自动内存管理**：良好的内存管理机制意味着程序运行具有更高的性能。Python 程序员无须关心内存的使用和管理，Python 自动分配和回收内存。
- **大型程序支持**：通过子模块、类和异常等工具，允许 Python 应用于大型程序开发。
- **内置数据结构**：Python 提供了常用数据结构支持。例如，列表、字段、字符串等都属于 Python 内置对象。同时，Python 也实现了各种数据结构的的标准操作，如合并、分片、排序和映射等。
- **内置库**：Python 提供丰富的标准库，从正则表达式匹配到网络等，使 Python 可以现实多种应用。
- **第三方工具集成**：Python 很容易集成第三方工具，通过各种扩展包将其应用到各种不同领域。

5. Python 简单易学

Python 的设计理念是“优雅”“明确”“简单”，提倡“用一种方法，最好是只有一种方法来做一件事”，所以，Python 语言语法简洁、代码易读。在国外一些知名大学，开始采用 Python 作为程序设计课程的编程语言。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均使用 Python 语言讲授。

Python 不强调数据类型，变量在使用之前无须定义其数据类型，同一个变量可存储不同类型的数据。



提示：可以在交互方式下执行 `import this` 命令，查看 Python 开发人员的设计理念，如图 1-2 所示。

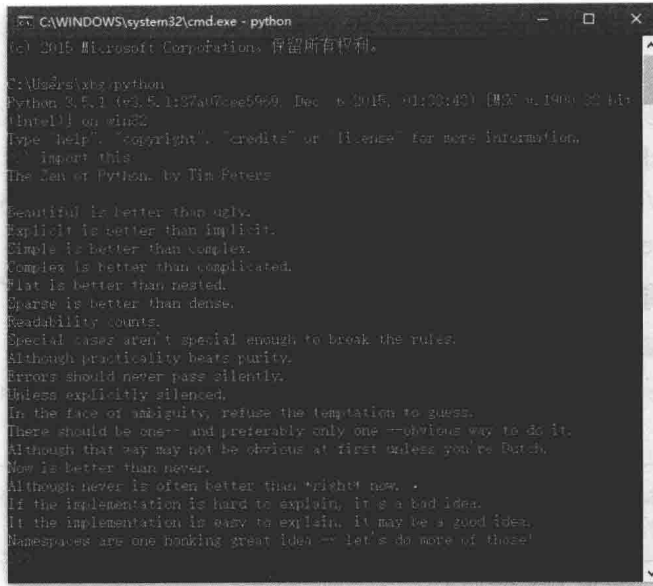


图 1-2 Python 开发人员的设计理念

1.1.5 Python 的版本

Python 发展到现在，经历了多个版本，如表 1-1 所示。

表 1-1 Python 版本历史

版本号	年份	拥有者	GPL 兼容
0.9.0~1.2	1991—1995	CWI	是
1.3~1.5.2	1995—1999	CNRI	是
1.6	2000	CNRI	否
2.0	2000	BeOpen.com	否
1.6.1	2001	CNRI	否
2.1	2001	PSF	否
2.0.1	2001	PSF	是
2.1.1	2001	PSF	是
2.1.2	2002	PSF	是
2.1.3	2002	PSF	是
2.2~2.7	2001—2015	PSF	是
3.x	2008 至今	PSF	是



提示：截至本书成稿时，Python 最新为 3.5.1 版本。需要注意的是 Python 3.x 不再兼容现有的 2.x 程序。可根据实际需要选择使用的版本，当然，选择较新的版本有利于以后软件升级。

作为一个开源软件，Python 拥有一个参与者众多的开发社区，它保持 Python 的不断更新和改进。Python 的开发者通过一个在线的源代码控制系统协同工作，所有对 Python 的修改必须遵循 PEP (Python Enhancement Proposals) 协议，并通过 Python 扩展回归测试系统的测试。目前，由一个非正式的组织 PSF (Python Software Foundation, Python 软件基金) 负责组织会议并处理 Python 的知识产权问题。

1.1.6 Python 3.x 与 2.x 的区别

Python 3.x 与 2.x 的主要区别如下。

1. Python 3.x 默认使用 UTF-8 编码

Python 3.x 源代码中的字符默认使用 UTF-8 编码，可以很好地支持中文或其他非英文字符。例如，在 Python 3.x 中可使用汉字作为变量名。

```
>>>长度=100
>>> print(长度)
100
```

在 Python 2.x 中不能使用汉字作为变量名，否则会出错。例如：

```
>>>速度=100
File "<stdin>", line 1
速度=100
  ^
SyntaxError: invalid syntax
```

2. print()函数代替了 print 语句

在 Python 3.x 中，调用 print()函数来输出数据。例如：

```
>>> x=100
>>> print(10,'abc',x)
10 abc 100
```

在 Python 2.x 中则使用 print 语句。例如：

```
>>> x=100
>>> print 10,'abc',x
10 abc 100
>>> print(10,'abc',x)          #这里 print 语句将(10,'abc',x)作为一个元组输出
(10, 'abc', 100)
```

3. 完全面向对象

在 Python 2.x 中的各种数据类型，在 3.x 中全面升级为类 (class)。例如，在 Python 2.x 中输出数据类型结果如下。

```
>>> int,float,str
(<type 'int'>, <type 'float'>, <type 'str'>)
```

在 Python 3.x 中输出数据类型结果如下。

```
>>> int,float,str
(<class 'int'>, <class 'float'>, <class 'str'>)
```

4. 用视图和迭代器代替了列表

下面的常用方法或函数在 Python 2.x 中返回列表，在 3.x 中有很多改变。

- 在 3.x 中，字典的 keys()、items()和 values()方法用返回视图代替了列表，不再支