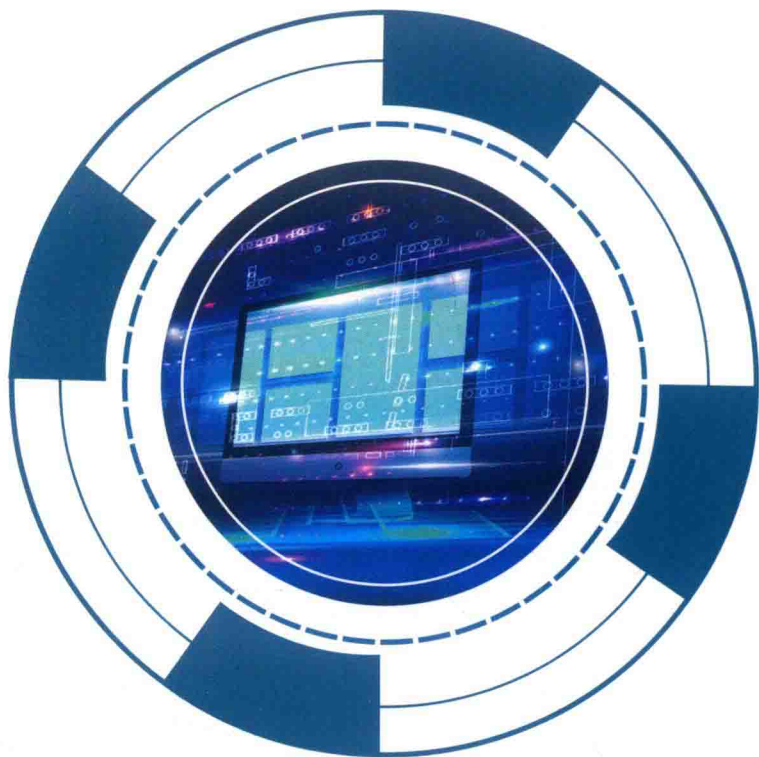




普通高等教育“十三五”规划教材



01011010110101011101010110101101011101
01011010110101011101010110101101011101

01011010101011101010101101011011101
0101101011101011101010111010101101011101

0101101011010101110101011101010110101101
0101101010101110101011010101011101

0101101010101101010101010101010101

Python 3 程序设计基础

刘德山 付彬彬 黄和 主编

 科学出版社

非外借

普通高等教育“十三五”规划教材

Python 3 程序设计基础

刘德山 付彬彬 黄 和 主 编

孙治军 张云龙 刘 钢 副主编

科学出版社

北 京

内 容 简 介

本书以易懂的语言、翔实的示例、全新的内容来诠释 Python 这门“简单”“优雅”“明确”“易学”的计算机语言。全书共 12 章，第 1~8 章是 Python 语言基础，覆盖了全国计算机等级考试二级 Python 语言程序设计的主要内容；第 9~11 章是 Python 语言的应用，包括图形用户界面、面向对象程序设计、数据库编程等内容；第 12 章重点介绍了 Python 第三方库的应用。本书内容以应用为核心展开，力求以知识的最小集来实现最大范围的应用。

本书难度适中，主要面向普通本科院校非计算机专业的学生，适合作为初学者学习 Python 程序设计的教材，也可作为全国计算机等级考试二级 Python 程序设计的参考用书。

图书在版编目(CIP)数据

Python 3 程序设计基础 / 刘德山, 付彬彬, 黄和主编. —北京: 科学出版社, 2018.11

普通高等教育“十三五”规划教材

ISBN 978-7-03-059071-8

I. ①P… II. ①刘… ②付… ③黄… 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 (2018) 第 232899 号

责任编辑: 宋 丽 王 惠 / 责任校对: 马英菊
责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

三河市良远印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2018 年 11 月第一版 开本: 787×1092 1/16

2018 年 11 月第一次印刷 印张: 19

字数: 448 000

定价: 49.00 元

(如有印装质量问题, 我社负责调换(良远))

销售部电话 010-62136230 编辑部电话 010-62135397-2052



版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

近年来，Python 已经成为计算机语言中的热词。

人工智能已经上升为国家战略。Python 丰富的 AI 库、机器学习库、自然语言和文本处理库，使其成为人工智能时代最好的语言。Python 还应用于数据分析、组件集成、图像处理、科学计算等众多领域。

为适应跨界创新的需求，不同层次、不同专业的读者，迫切需要一种可以更多地专注于要解决的问题，而不必更多地考虑细节的计算机语言，让计算机语言回归服务的功能，Python 是最佳选择。

Python 以“简单”“优雅”“明确”“易学”等特点成为学习编程的入门语言。十几万的第三方库形成了 Python 的“计算生态”，推动了 Python 的发展。

Python 在业界广泛使用，如 YouTube、BT、豆瓣、知乎、Google、Yahoo!、Facebook、百度、腾讯、美团等都在使用 Python。

面对 Python 诸多的应用和需求，以及适合于所有专业学生学习的特点，2018 年，教育部将 Python 纳入全国计算机等级考试范围，相信未来几年 Python 将得到更好地普及与发展。

作者曾主讲 C、Java、Python 系列课程，从教学实践中精选了大量的示例，从“实用、易用、有效”的角度组织编写了本书，让读者能以最短的时间，了解和学习这门简单、优雅的语言。本书内容以应用为核心展开，难度适中，力求以知识的最小集来实现最大范围的应用。本书追求语言易懂、示例翔实、结构明晰的风格。

本书主要有以下特色：

1. 重点突出。在保证内容科学、完整的前提下，由浅入深地安排章节次序。考虑到 Python 语言的应用特色，更多强调应用思维。例如，在组合数据类型的应用、文件操作、第三方库等章节，更着重体现应用特性。将 Python 面向对象程序设计部分安排到第 11 章，教师可以根据需要选讲。

2. 案例资源丰富。全书设计了 237 个示例，覆盖 Python 的重要知识点；加强实践环节，精心设计和编写课后习题；教材编写与教学资源建设同步，提供教学课件、程序源码。

3. 综合考虑全国计算机等级考试和 Python 应用需求。书中的知识点基本覆盖了等级考试需要的核心内容，又对部分使用频率低的内容做了删减。

本书结构如下：

第 1~8 章是 Python 语言基础，覆盖了全国计算机等级考试二级 Python 语言程序设计的主要内容，教学设计约 32 学时。

第 9~11 章是 Python 语言的应用，包括图形用户界面、面向对象程序设计、数据库编程等内容。

第 12 章详细介绍了 Python 第三方库的应用。读者应掌握 Python 第三方库文档的查阅方法，了解 Python 库编程的特点，领会 Python 编程的“计算生态”特征。

本书建议教学的组织形式是示例—分析—练习—总结。从应用的角度介绍语言，通过示例来说明编程的方法和过程。建议授课 32~48 学时，第 9~12 章根据需要选讲。

本书适合作为 Python 程序设计课程的教材，以及全国计算机等级考试辅导教材。

本书由刘德山、付彬彬、黄和任主编，孙治军、张云龙、刘钢任副主编，李舒、王婷婷参与编写。书中难免存在疏漏和不足之处，恳请读者指正。

48 学时不长，稍纵即逝；48 学时不短，您可以学习和发挥 Python 的荣光。人生苦短，一起学习 Python。

编者

2018 年 8 月

目 录

第 1 章 Python 概述	1
1.1 计算机语言	1
1.1.1 计算机语言概述	1
1.1.2 编译与解释	2
1.2 Python 简介	3
1.2.1 Python 的历史	3
1.2.2 Python 的特点	3
1.2.3 Python 的应用	4
1.3 Python 的开发环境	5
1.3.1 Python 下载和安装	5
1.3.2 Python 的 IDLE 开发环境	7
1.3.3 PyCharm 集成开发环境	8
1.4 Python 程序执行过程	12
1.4.1 Python 程序执行原理	12
1.4.2 建立和运行 Python 程序	12
1.5 Python 程序方法与应用	14
1.5.1 程序设计方法	14
1.5.2 程序设计示例	15
小结	19
习题	20
第 2 章 Python 基础知识	21
2.1 程序的书写规范	21
2.1.1 Python 的语句	21
2.1.2 代码块与缩进	22
2.1.3 注释	22
2.2 标识符和关键字	23
2.2.1 标识符	23
2.2.2 关键字	23
2.3 Python 的数值类型与变量	24
2.3.1 数值类型	24
2.3.2 变量和常量	26
2.4 Python 的字符串类型	27
2.4.1 字符串的表示	27
2.4.2 字符串输出的格式化	28
2.4.3 字符串的操作符	32
2.4.4 内置的字符串处理函数	33

2.5 Python 的运算符	36
2.5.1 算术运算符	36
2.5.2 比较运算符	37
2.5.3 逻辑运算符	37
2.5.4 赋值运算符	37
2.5.5 位运算符	38
2.6 运算符的优先级	39
小结	40
习题	40
第 3 章 Python 程序的流程控制	42
3.1 输入/输出语句	42
3.1.1 输入语句	42
3.1.2 输出语句	43
3.2 程序设计流程	44
3.2.1 程序流程图	44
3.2.2 结构化程序设计基本流程	44
3.3 分支结构	45
3.4 循环结构	47
3.4.1 遍历循环: for	47
3.4.2 条件循环: while	49
3.4.3 循环的嵌套	50
3.5 流程控制的其他语句	51
3.5.1 跳转语句	51
3.5.2 pass 语句	51
3.5.3 循环结构中的 else 语句	52
3.6 流程控制语句的应用	53
小结	55
习题	55
第 4 章 Python 的组合数据类型	58
4.1 序列类型	58
4.2 列表	59
4.2.1 列表的基本操作	59
4.2.2 列表的方法	60
4.2.3 遍历列表	61
4.3 元组	62
4.3.1 元组的基本操作	62
4.3.2 元组与列表的转换	63
4.4 字典	63
4.4.1 字典的基本操作	64
4.4.2 字典的常用方法	65

4.5 集合	68
4.5.1 集合的常用操作	68
4.5.2 集合运算	70
4.6 组合数据类型的应用	70
小结	72
习题	73
第5章 Python 函数	75
5.1 函数的定义和调用	75
5.1.1 函数的定义	75
5.1.2 函数的调用	76
5.1.3 函数的嵌套	77
5.2 函数的参数和返回值	78
5.2.1 函数的参数	78
5.2.2 默认参数	80
5.2.3 可变参数	81
5.2.4 函数的返回值	83
5.2.5 lambda 函数	83
5.3 闭包和递归函数	84
5.3.1 闭包	84
5.3.2 递归函数	86
5.4 变量的作用域	87
5.4.1 局部变量	87
5.4.2 全局变量	88
5.4.3 global 语句	90
5.5 Python 的内置函数	91
5.5.1 数学运算函数	91
5.5.2 字符串运算函数	91
5.5.3 转换函数	91
5.5.4 序列操作函数	92
5.5.5 Python 操作相关函数	97
小结	98
习题	99
第6章 模块与 Python 的库	101
6.1 模块	101
6.1.1 模块的概念	101
6.1.2 导入模块	101
6.1.3 执行模块	103
6.1.4 模块搜索路径	104
6.1.5 __name__ 属性	105
6.2 包	106

6.3	Python 的标准库	107
6.3.1	math 库	107
6.3.2	random 库	109
6.3.3	datetime 库	110
6.3.4	turtle 库	114
6.4	Python 的第三方库	117
6.4.1	第三方库简介	118
6.4.2	使用 pip 工具安装第三方库	118
6.4.3	使用 pyinstaller 库打包文件	120
6.5	jieba 库的应用	121
6.5.1	jieba 库简介	122
6.5.2	jieba 库的分词函数	122
6.5.3	添加单词和自定义词典	123
6.5.4	基于 TF-IDF 算法的关键词抽取	125
6.5.5	中文文本的词频统计	125
6.5.6	打包词频统计程序	129
	小结	130
	习题	130
第 7 章	Python 的文件操作	132
7.1	文件的相关概念	132
7.2	文件的打开和关闭	133
7.3	文件的读/写操作	135
7.3.1	读取文件数据	135
7.3.2	向文件写数据	137
7.3.3	文件的定位读/写	138
7.3.4	读/写二进制文件	139
7.4	文件和目录操作	141
7.4.1	常用的文件操作函数	141
7.4.2	文件的复制、删除、重命名操作	142
7.4.3	文件的目录操作	143
7.5	使用 CSV 文件格式读/写数据	144
7.5.1	CSV 文件简介	144
7.5.2	读写 CSV 文件	145
7.6	文件操作的应用	148
	小结	151
	习题	151
第 8 章	异常处理	153
8.1	异常处理概述	153
8.1.1	异常的概念	153
8.1.2	异常示例	154

8.2	Python 的异常类	155
8.3	异常处理机制	158
8.3.1	try-except 语句	158
8.3.2	else 语句和 finally 语句	159
8.3.3	捕获所有的异常	161
8.4	抛出异常	162
8.4.1	raise 语句	163
8.4.2	抛出异常示例	164
8.5	断言与上下文管理	165
8.5.1	断言	165
8.5.2	上下文管理	167
8.6	自定义异常	168
	小结	169
	习题	169
第 9 章	tkinter GUI 编程	172
9.1	tkinter 编程概述	172
9.1.1	第一个 tkinter GUI 程序	172
9.1.2	设置窗口和组件的属性	173
9.2	tkinter GUI 的布局管理	174
9.2.1	使用 pack()方法的布局	175
9.2.2	使用 grid()方法的布局	176
9.2.3	使用 place()方法的布局	177
9.2.4	使用框架的复杂布局	178
9.3	tkinter 的常用组件	180
9.3.1	Label 组件	180
9.3.2	Button 组件	181
9.3.3	Entry 组件	182
9.3.4	Listbox 组件	184
9.3.5	Radiobutton 组件	186
9.3.6	Checkbutton 组件	188
9.3.7	Text 组件	189
9.3.8	Spinbox 组件	191
9.4	tkinter 的事件处理	192
9.4.1	使用 command 参数实现事件处理	192
9.4.2	使用组件的 bind()方法实现事件处理	194
9.5	tkinter GUI 的应用	196
	小结	198
	习题	199
第 10 章	Python 的数据库编程	200
10.1	数据库的基础知识	200
10.1.1	数据库的概念	200

10.1.2	关系型数据库	201
10.1.3	Python 的 sqlite3 模块	202
10.2	SQLite 数据库	202
10.2.1	SQLite 数据库简介	202
10.2.2	下载和安装 SQLite 数据库	203
10.2.3	SQLite3 常用命令	203
10.2.4	SQLite3 的数据类型	204
10.2.5	sqlite3 模块中的对象	205
10.2.6	SQLite3 的函数	206
10.2.7	创建 SQLite 数据库	207
10.3	关系数据库语言 SQL	207
10.3.1	数据表的建立和删除	207
10.3.2	向表中添加列	208
10.3.3	向表中插入数据	209
10.3.4	修改表中的数据	209
10.3.5	删除数据	210
10.3.6	查询数据	210
10.4	Python 的 SQLite3 编程	211
10.4.1	访问数据库的步骤	211
10.4.2	创建数据库和表	213
10.4.3	数据库的插入、更新和删除操作	213
10.5	SQLite 编程的应用	214
	小结	219
	习题	219
第 11 章	面向对象程序设计	222
11.1	面向对象程序设计概述	222
11.1.1	面向对象程序设计的概念	222
11.1.2	面向对象程序设计的特点	223
11.2	创建类与对象	224
11.2.1	创建类	224
11.2.2	创建对象	225
11.3	构造方法和析构方法	226
11.3.1	构造方法	226
11.3.2	析构方法	227
11.3.3	self 参数	228
11.3.4	实例属性和类属性	229
11.3.5	类方法和静态方法	230
11.4	类的继承	233
11.4.1	继承的实现	233
11.4.2	方法重写	235
11.4.3	Python 的多继承	236
11.5	类的多态	237

11.6	运算符重载	239
11.7	面向对象程序设计的应用	242
	小结	249
	习题	249
第 12 章	Python 的第三方库	252
12.1	Python 常见的第三方库	252
12.2	科学计算的 numpy 库	253
12.2.1	numpy 库的使用	253
12.2.2	numpy 数组的算术运算	258
12.2.3	numpy 数组的形状操作	260
12.3	图表绘制的 matplotlib 库	262
12.3.1	matplotlib 介绍	262
12.3.2	matplotlib.pyplot 模块	262
12.3.3	绘制直方图、条形图、饼状图	267
12.4	爬取网页的 urllib 和 requests 库	271
12.4.1	爬取网页的基础知识	271
12.4.2	urllib 库	272
12.4.3	requests 库	275
12.5	解析网页的 BeautifulSoup4 库	277
12.5.1	BeautifulSoup4 库概述	277
12.5.2	BeautifulSoup4 库的对象	279
12.5.3	BeautifulSoup4 库操作解析文档树	281
12.6	网页爬取示例	285
	小结	288
	习题	289
	参考文献	290

第 1 章

Python 概述

Python 是一种面向对象的、解释型的高级计算机语言，可应用于 Web 开发、科学计算、游戏开发、图形用户界面设计等方面。那么，什么是计算机语言？解释型语言有什么特点？Python 语言有什么特点？本章将介绍这些内容，并介绍 Python 程序的开发环境和程序的执行过程。

1.1 计算机语言

1.1.1 计算机语言概述

让计算机按照用户的目的完成相应的操作，需要使用计算机语言。计算机语言即程序设计语言，是用于描述计算机所执行的操作的语言。从计算机产生到现在，作为软件开发工具的程序设计语言经历了机器语言、汇编语言、高级语言等几个阶段。

1. 机器语言

机器语言是采用计算机指令格式并以二进制编码表达各种操作的语言。计算机能够直接理解和执行机器语言程序。

机器语言的特点是能够被计算机直接识别，执行速度快，占用存储空间小，但难读、难记，编程难度大，调试、修改烦琐，而且不同型号的计算机具有不同的机器指令系统。

2. 汇编语言

汇编语言是一种符号语言，它用助记符来表达指令功能。

汇编语言程序较机器语言程序易读、易写，并保持了机器语言执行速度快、占用存储空间小的优点。但汇编语言的语句功能比较简单，程序的编写仍然比较复杂，而且程序难以移植，因为汇编语言和机器语言都是面向机器的语言，都是为特定的计算机系统而设计的。汇编语言程序不能被计算机直接识别和执行，需要由一种起翻译作用的程序

(汇编程序)，将其翻译成机器语言程序（目标程序），计算机才能执行，这一翻译过程称为“汇编”。

机器语言和汇编语言统称为低级语言。

3. 高级语言

高级语言是面向问题的语言，比较接近于人类的自然语言。高级语言是与计算机结构无关的程序设计语言，具有更强的表达能力，可以方便地表示数据的运算和程序控制结构，能更有效地描述各种算法，使用户容易掌握。

Python 语言是一种高级语言，例如，计算并显示 $5+11$ 运算结果的 Python 语言程序如下：

```
>>> print(5+11)
16      #运算结果
```

用高级语言编写的程序（源程序）不能被计算机直接识别和执行，需要经过翻译程序翻译成机器语言程序（目标程序）才能执行，高级语言的翻译程序有编译程序和解释程序两种。下面分别介绍编译程序和解释程序。

1.1.2 编译与解释

用不同的高级语言编写的计算机程序，其执行方式是不同的。这里所说的执行方式是指计算机执行一个程序的过程。按照计算机程序的执行方式，将高级语言分成两类：静态语言和脚本语言。静态语言采用编译执行的方式，脚本语言采用解释执行的方式。无论哪种方式，用户执行程序的方法是一致的，如通过鼠标双击执行一个程序。

1. 编译

编译是将源程序代码转换成目标代码的过程。源代码是计算机高级语言代码，而目标代码则是机器语言代码。执行编译的计算机程序称为编译器（compiler）。

2. 解释

解释是将源代码逐条转换成目标代码，同时逐条运行目标代码的过程。执行解释的计算机程序称为解释器（interpreter）。

解释和编译的区别在于，编译是一次性地翻译，程序被编译后，运行时不再需要源代码。解释则是在每次程序运行时都需要解释器和源代码。这两者的区别类似于外语资料的翻译和实时的同声传译。

编译的过程只进行一次，所以编译过程的速度并不是关键，关键是生成的目标代码的执行速度。因此，编译器一般集成尽可能多的优化技术，使生成的目标代码有更高的执行效率。而解释器考虑到执行速度，不会集成太多的优化技术。

1.2 Python 简介

1.2.1 Python 的历史

Python 的作者 Guido van Rossum 是荷兰人。Guido 理想中的计算机语言，应能够方便地调用计算机的各项功能，如打印、绘图、语音等，而且程序可以轻松地编辑与运行，适合所有人学习和使用。1989 年，Guido 开始编写这种理想的计算机语言的脚本解释程序，并命名为 Python。Python 语言的目标是成为功能全面、易学易用、可拓展的语言。

Python 的第一个公开版本于 1991 年发布。它是用 C 语言实现的，能够调用 C 语言的库文件，具有类、函数、异常处理等功能，包含表和词典等核心数据类型及以模块为基础的拓展系统。

之后，在 Python 的发展过程中，存在 Python 2.x 和 Python 3.x 两个不同系列的版本，这两个版本之间不兼容。为了满足用户的需要，目前是 Python 2.x 和 Python 3.x 两个版本并存。Python 2.x 和 Python 3.x 两个不同的版本并存的原因是，Python 3.0 发布时不支持 Python 2.0 版本，导致很多用户无法正常升级使用新版本，所以后来又发布了一个 Python 2.7 的过渡版本，而且 Python 2.7 将被支持到 2020 年。Python 3.x 从 2008 年开始发布，目前的最高版本是 Python 3.6，本书的程序使用 Python 3.6 版本的编译器。

1.2.2 Python 的特点

Python 语言是目前最流行和发展最迅速的计算机语言，它具有如下特点：

(1) 简单，所以易学

Python 以简单易学的特点成为学习编程的入门语言。一个好的 Python 程序像一篇英文文档，非常接近于人的自然语言，用户在应用 Python 过程中，可以更多地专注于要解决的问题，而不必过多地考虑计算机语言的细节，回归语言的服务功能。

(2) 开源，拥有众多的开发群体

用户可以查看 Python 源代码，研究其代码细节或进行二次开发。用户不需要为使用 Python 进行软件开发支付费用，不涉及版权问题。因为开源，越来越多的优秀程序员加入 Python 开发中，Python 功能愈加丰富和完善。

(3) Python 是解释型语言

Python 程序可以直接从源代码运行。在计算机内部，Python 解释器把源代码转换成称为字节码的中间形式，然后把它翻译成计算机使用的机器语言并运行。基于解释型的特点，用户可以将一些代码行在交互方式下直接测试执行，使得 Python 的学习更加容易。

(4) 良好的跨平台性和可移植性

Python 的开源本质，使得它可以被移植到多个平台。如果用户的 Python 程序使用了依赖于系统的特性，Python 程序可能需要修改与平台相关的代码。Python 的应用平台包括 Linux、Windows、Mac OS、Solaris、OS/2、FreeBSD、Amiga、Android、iOS 等。

(5) 面向对象

Python 既支持面向过程的编程，也支持面向对象的编程。在面向过程的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如 C++ 和 Java 相比，Python 以一种非常强大又简单的方式实现面向对象编程，为大型程序的开发提供了方便。

(6) 可扩展性和丰富的第三方库

Python 程序可以使用 C 或 C++ 编写的程序，从而使某段关键代码运行得更快或者实现某些算法不公开；也可以把 Python 程序嵌入 C 或 C++ 程序，提高 C 或 C++ 程序的脚本支持能力，具有良好的可扩展性。

Python 还有功能强大的开发库。Python 标准库可以处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、HTML、WAV 文件、密码系统、GUI（图形用户界面）和其他与系统有关的操作。Python 的这些功能都是可用且完备的。除了标准库，还有大量高质量的第三方库，如 wxPython、Twisted 和 Python 图像库等。

1.2.3 Python 的应用

Python 应用领域覆盖 Web 开发、科学运算、系统运维、GUI 编程、数据库编程等诸多方面。

(1) Web 开发

Python 包含标准的 Internet 模块，可用于实现网络通信及应用。Python 的第三方框架包括 Django、Web2py、Zope 等，可以让程序员方便地开发 Web 应用程序。典型的 Web 应用如 Google 爬虫、Google 广告、世界上最大的视频网站 YouTube、豆瓣、知乎等都使用 Python 开发。

(2) 科学运算

Python 广泛应用于人工智能与深度学习领域，典型的第三方库包括 NumPy、SciPy、Matplotlib 等。随着众多程序库的开发，Python 越来越适合于做科学计算、绘制高质量的 2D 和 3D 图像。例如，美国航空航天局（NASA）大量使用 Python 进行数据分析和运算。

(3) 云计算

Python 是云计算领域应用最广泛的语言，典型应用如开源的云计算管理平台 OpenStack。

(4) 系统运维

Python 是运维人员必备语言。Python 标准库包含多个调用操作系统功能的库。通过第三方软件包 pywin32，Python 能够访问 Windows API。使用 IronPython，Python 能够直接调用 .NET Framework。一般来说，Python 编写的系统管理脚本在可读性、性能、代码重用度、扩展性几方面都优于普通的 Shell 脚本。

(5) GUI 编程

Python 可以非常简单、快捷地实现 GUI (图形用户界面) 程序。Python 内置了 tkinter 的标准面向对象接口 Tk GUI API, 可以非常方便地开发图形界面应用程序。还可以使用其他一些扩展包, 如 wxPython、PyQt、Dabo 等, 也可在 Python 中创建 GUI 应用。

1.3 Python 的开发环境

1.3.1 Python 下载和安装

Python 是一个轻量级的软件, 可以在 Python 官网下载, 网址如下:

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

Python 开发包下载页面如图 1-1 所示, 本书使用的是适合 Windows 操作系统的 Python 3.6.5 版, 也可以下载适合 Linux、iOS、Android 等操作系统的 Python 开发包, 或选择其他 Python 版本。

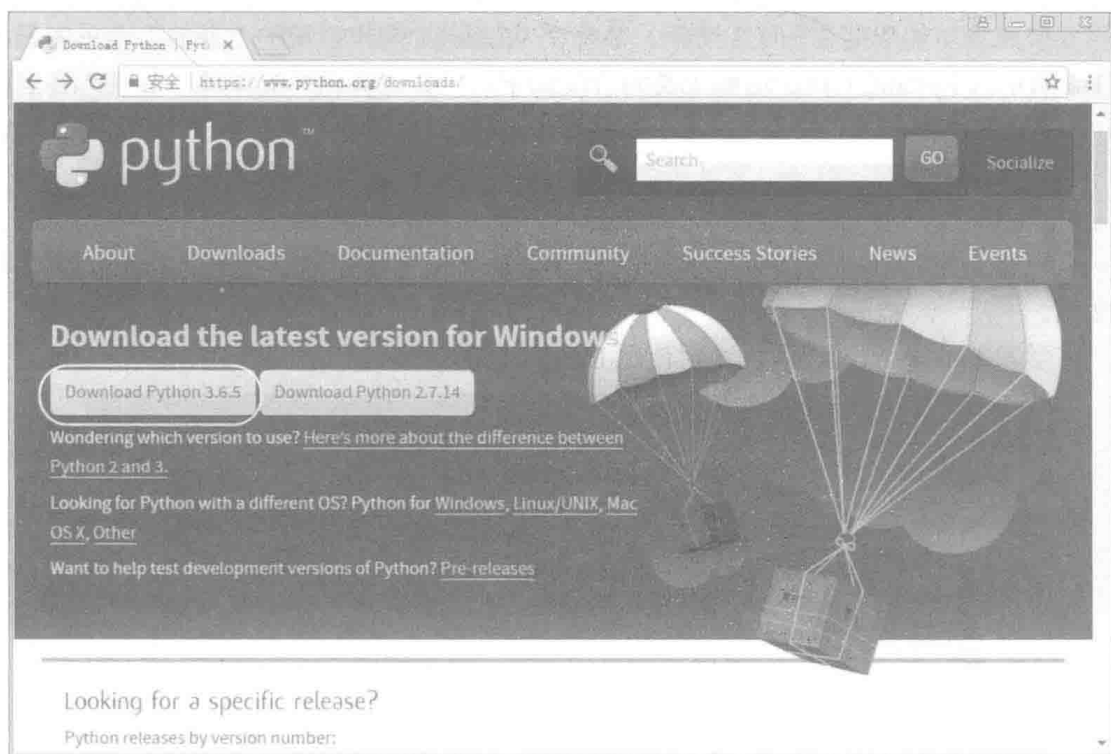


图 1-1 Python 官网下载页面

双击下载的 Python 安装程序 python-3.6.5.exe, 将启动安装向导, 按提示操作即可。在图 1-2 的安装页面中, 选中 Add Python 3.6 to PATH 复选框, 将 Python 的可执行文件路径添加到 Windows 操作系统的环境变量 path 中, 方便在将来的开发中启动各种 Python 工具。