



21世纪高等学校计算机  
基础实用规划教材

# Python 程序设计

◎ 祁瑞华 主编 郑旭红 副主编



清华大学出版社

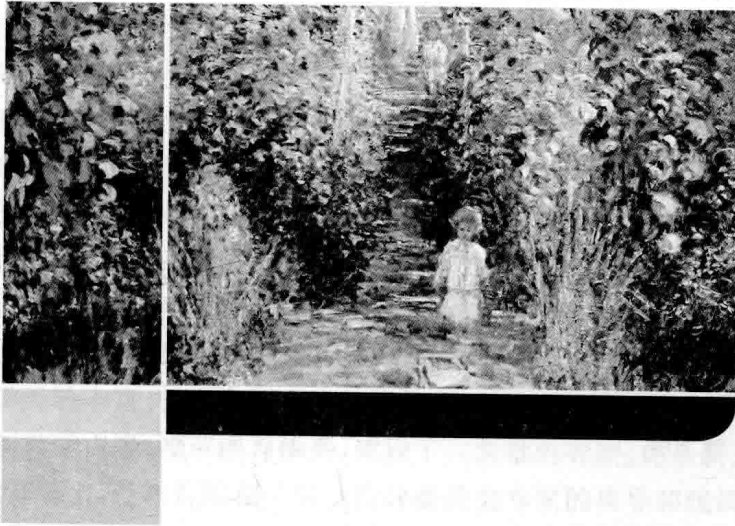




21世纪高等学校计算机  
基础实用规划教材

# Python 程序设计

◎ 祁瑞华 主编 郑旭红 副主编



清华大学出版社  
北京

## 内 容 简 介

本书以零基础为起点介绍 Python 程序设计方法。各章内容由浅入深、相互衔接、前后呼应、循序渐进。全书各章节选用丰富的程序设计语言经典实例来讲解基本概念和程序设计方法,同时配有大量习题供读者练习。

全书共 9 章,第 1 章介绍 Python 语言开发环境;第 2 章介绍 Python 程序设计语言的基础语法;第 3 章介绍列表和元组;第 4 章介绍字符串;第 5 章介绍字典和集合;第 6 章介绍函数与模块;第 7 章介绍 Python 的程序流程控制;第 8 章介绍文件操作;第 9 章介绍异常处理。

本书语言表达简洁、严谨、流畅,内容通俗易懂、重点突出、实例丰富,适合作为高等院校各专业程序设计语言课程的教材,也可以作为非计算机专业公共基础课教材。

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/祁瑞华主编. —北京:清华大学出版社,2018

(21 世纪高等学校计算机基础实用规划教材)

ISBN 978-7-302-49808-7

I. ①P… II. ①祁… 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(2018)第 033242 号

责任编辑:贾 斌 薛 阳

封面设计:刘 键

责任校对:李建庄

责任印制:王静怡

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:13.25

字 数:324 千字

版 次:2018 年 3 月第 1 版

印 次:2018 年 3 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

产品编号:077974-01

# 出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

---

Python 语言作为一种免费、开源语言,已被许多学校引入教学过程。它是面向对象和过程的程序设计语言,具有丰富的数据结构、可移植性强、语言简洁、程序可读性强等特点。本书根据实际教学经验,对内容进行选择,力求面向读者,以程序设计零基础为起点,结合 Python 程序设计的基础知识、Python 的基础语法、列表和元组、字符串、字典和集合、函数与模块、Python 的控制语句、文件操作和异常处理,通过丰富的代码实例和示例,向读者介绍 Python 程序设计的方法及主要思想。

本书编者长期从事计算机课程的教学工作,具有丰富的教学经验和较强的科学研究能力。编者本着加强基础、注重实践、强调思想的教学、突出实践应用能力和创新能力培养的原则,力求使本书有较强的可读性、适用性和先进性。

本教材从零基础起点出发,结构精简,语言流畅,具体特点如下。

(1) 由浅入深、循序渐进地介绍 Python 程序设计语言,让读者能够较为系统全面地掌握程序设计的理论和应用。

(2) 运用丰富的案例解释程序设计方法和思想,易于学习者理解。

(3) 提供大量配套习题供读者深入学习、掌握教材内容,所提供的代码实例和案例均在 Python 2.7 环境下通过调试和运行。

本书由祁瑞华任主编,郑旭红任副主编。提供本书初稿的主要有祁瑞华(第 1 章)、李富宇(第 2 章)、刘彩虹(第 3 章和第 4 章)、郭旭(第 5 章、第 8 章、第 9 章),杨松(第 6 章),郑旭红(第 7 章)。参加书中内容、习题和解答编写的还有刘强、秦兵兵、蔡晓丹、杨岚、徐玲和魏晓聪等。

本书可作为(但不限于):

- (1) 计算机专业本科生程序设计教材;
- (2) 会计、经济、管理、统计以及其他非工科专业本科生程序设计教材;
- (3) 非计算机专业本科生公共基础课程程序设计教材;
- (4) 专科院校或职业技术学院程序设计教材;
- (5) Python 培训用书;
- (6) 编程爱好者自学用书。

本书所提供的程序示例及实例均在 Python 2.7 环境下进行了调试和运行,同时,为了帮助读者更好地学习 Python,编者在每章后编写了大量的习题供读者练习。

在本书的编写过程中,清华大学出版社的魏江江老师和贾斌老师提出了许多宝贵的意

见,在此致以衷心的感谢。

由于 Python 程序设计技术的发展日新月异,加之作者水平有限,书中难免存在不足之处,敬请广大读者批评指正。

编者

2017 年 11 月于大连

# 目 录

---

<b>第 1 章 Python 概述</b> .....	1
1.1 初识 Python .....	1
1.1.1 Python 语言的特点 .....	1
1.1.2 Python 语言的应用领域 .....	1
1.2 Python 版本和开发环境 .....	2
1.2.1 Python 语言的版本 .....	2
1.2.2 Python 的集成开发环境 .....	3
1.3 Python 的安装和设置 .....	4
1.3.1 Python 的下载 .....	4
1.3.2 Python 的安装 .....	4
1.3.3 Python 扩展包的安装和管理 .....	6
1.4 执行 Python 程序 .....	8
1.4.1 Python 解释器执行 Python 程序 .....	8
1.4.2 Python 集成开发环境 IDLE .....	9
1.4.3 Anaconda——Python 科学计算环境 .....	12
1.5 本章小结 .....	22
1.6 上机实验 .....	23
上机实验 1 Python 的安装和使用 .....	23
上机实验 2 Anaconda 的安装和使用 .....	24
习题 1 .....	25
<b>第 2 章 Python 的基础语法</b> .....	26
2.1 Python 的文件类型 .....	26
2.2 Python 的编码规范 .....	26
2.2.1 命名规则 .....	26
2.2.2 代码缩进与冒号 .....	27
2.2.3 使用空行分隔代码 .....	28
2.2.4 正确的注释 .....	29
2.2.5 语句的分隔 .....	29
2.2.6 PEP 8 编码规范 .....	29



2.3	变量和常量	29
2.3.1	变量的命名和赋值	29
2.3.2	局部变量和全局变量	30
2.3.3	常量	32
2.4	数据类型	32
2.5	运算符	35
2.5.1	算术运算符	35
2.5.2	关系运算符	36
2.5.3	逻辑运算符	37
2.6	本章小结	38
2.7	上机实验	38
	上机实验 1 代码缩进	38
	上机实验 2 添加注释	38
	上机实验 3 关系运算符	39
	习题 2	39
<b>第 3 章</b>	<b>列表和元组</b>	<b>42</b>
3.1	序列简介	42
3.1.1	索引	42
3.1.2	切片	43
3.2	序列相加	46
3.2.1	序列重复	46
3.2.2	成员资格	47
3.2.3	序列比较	48
3.2.4	序列排序	49
3.2.5	长度、最小值和最大值	49
3.3	列表	50
3.3.1	列表的创建与删除	51
3.3.2	列表元素的增加	52
3.3.3	列表元素的删除	54
3.3.4	列表元素的访问与计数	55
3.3.5	成员资格判断	56
3.3.6	切片操作	56
3.3.7	列表排序	58
3.3.8	列表推导式	59
3.4	元组	60
3.4.1	元组的创建与删除	60
3.4.2	元组的基本操作	61
3.4.3	元组与列表的区别	62

3.4.4	生成器推导式 .....	62
3.5	本章小结 .....	63
3.6	上机实验 .....	63
	上机实验 1 Python 列表与集合 .....	63
	上机实验 2 序列内置函数 .....	64
	上机实验 3 偶数变换 .....	65
	上机实验 4 六组数据结构 .....	65
	习题 3 .....	66
<b>第 4 章</b>	<b>字符串与正则表达式 .....</b>	<b>71</b>
4.1	字符串 .....	71
	4.1.1 字符串常量 .....	71
	4.1.2 字符串的转义符 .....	72
	4.1.3 字符串的基本操作 .....	72
	4.1.4 字符串方法 .....	73
	4.1.5 字符串的格式化 .....	79
4.2	正则表达式 .....	82
	4.2.1 简单的正则表达式 .....	82
	4.2.2 re 模块主要方法 .....	85
	4.2.3 使用正则表达式对象 .....	86
	4.2.4 子模式与 match 对象 .....	87
4.3	本章小结 .....	88
4.4	上机实验 .....	88
	上机实验 1 字符串的格式化 .....	88
	上机实验 2 字符串的切片操作 .....	89
	上机实验 3 字符串的拆分操作 .....	89
	上机实验 4 字符串的拆分和连接操作 .....	90
	上机实验 5 正则表达式 .....	90
	习题 4 .....	91
<b>第 5 章</b>	<b>字典和集合 .....</b>	<b>95</b>
5.1	字典 .....	95
	5.1.1 字典的创建与删除 .....	95
	5.1.2 字典元素的访问 .....	97
	5.1.3 字典元素的修改与添加 .....	99
	5.1.4 有序字典 .....	100
5.2	集合 .....	102
	5.2.1 集合的创建与删除 .....	102
	5.2.2 集合操作 .....	103

5.2.3 不可变集合	105
5.3 本章小结	106
5.4 上机实验	107
上机实验 1 字典	107
上机实验 2 有序字典	108
上机实验 3 集合	108
上机实验 4 不可变集合	109
习题 5	109
<b>第 6 章 函数与模块</b>	<b>114</b>
6.1 Python 的程序结构	114
6.2 函数	115
6.2.1 函数的定义	115
6.2.2 函数的参数	116
6.2.3 函数的返回值	119
6.2.4 函数的嵌套	120
6.2.5 递归函数	121
6.2.6 Lambda 函数	122
6.2.7 变量作用域	123
6.3 模块	126
6.3.1 模块的创建	126
6.3.2 模块的导入	127
6.3.3 模块的属性	129
6.3.4 模块的内置函数	130
6.3.5 自定义包	131
6.4 本章小结	132
6.5 上机实验	133
上机实验 1 Python 用户自定义函数	133
上机实验 2 Python 递归函数	134
上机实验 3 自定义模块的使用	136
上机实验 4 包的使用	138
习题 6	139
<b>第 7 章 Python 的控制语句</b>	<b>144</b>
7.1 print 语句	144
7.2 赋值语句	146
7.3 条件语句	147
7.3.1 if 语句	148
7.3.2 else 子句	148

7.3.3	elif 子句 .....	149
7.3.4	if 语句的嵌套代码块 .....	151
7.3.5	条件语句程序举例 .....	153
7.4	循环语句 .....	155
7.4.1	while 循环 .....	155
7.4.2	for 循环 .....	157
7.4.3	Python break 和 continue 语句 .....	158
7.4.4	Python 循环嵌套 .....	161
7.5	本章小结 .....	162
7.6	上机实验 .....	162
	上机实验 1 简单的数据处理 .....	162
	上机实验 2 if 语句的应用 .....	164
	上机实验 3 指定次数的循环 .....	165
	习题 7 .....	165
<b>第 8 章</b>	<b>文件操作 .....</b>	<b>169</b>
8.1	显示和输入数据 .....	169
8.1.1	输出数据 .....	169
8.1.2	输入数据 .....	170
8.2	文件操作 .....	171
8.2.1	打开文件 .....	171
8.2.2	关闭文件 .....	172
8.2.3	读取文件内容 .....	172
8.2.4	向文件中写入数据 .....	173
8.2.5	文件属性 .....	174
8.2.6	删除文件 .....	174
8.2.7	重命名文件 .....	175
8.2.8	复制文件 .....	175
8.2.9	移动文件 .....	176
8.3	目录编程 .....	176
8.3.1	获取当前目录 .....	176
8.3.2	获取目录内容 .....	176
8.3.3	创建目录 .....	178
8.3.4	删除目录 .....	178
8.4	本章小结 .....	178
8.5	上机实验 .....	180
	上机实验 1 猜数字(标准输入输出) .....	180
	上机实验 2 file 对象操作 .....	181
	上机实验 3 遍历文件夹 .....	181

上机实验 4 文件/目录操作 .....	182
习题 8 .....	183
<b>第 9 章 异常处理</b> .....	<b>187</b>
9.1 异常 .....	187
9.2 捕获异常 .....	189
9.2.1 try...except 语句 .....	189
9.2.2 捕获多种异常 .....	190
9.2.3 捕获所有异常 .....	191
9.3 finally 语句 .....	191
9.4 本章小结 .....	192
9.5 上机实验 .....	192
上机实验 1 常见异常 .....	192
上机实验 2 带异常处理的文件读写 .....	193
习题 9 .....	193
<b>参考文献</b> .....	<b>197</b>

Python 是一种简单易学、功能强大的编程语言,具有高效率的高级数据结构和简洁有效的面向对象编程方法。Python 凭借其具有简练的语法、动态的编程方法和解释执行的特点,已经成为很多领域和平台上的脚本撰写和快速应用开发的理想语言。

## 1.1 初识 Python

### 1.1.1 Python 语言的特点

Python 语言的特点总结为如下几条:

(1) Python 是一种高层次的结合了解释型、交互型和面向对象的脚本语言。

(2) Python 语言风格简洁。基于“对于一个特定问题,只提供一种最好的解决方法”的思路,Python 语言具有简洁清晰的语法风格,易读懂、易维护。

(3) Python 语言具有强大的处理能力,集成了模块、异常处理和类的概念,内置支持灵活的数组和字典等高级数据结构类型,完全支持继承、重载、派生、多继承,支持重载运算符和动态类型,源代码易于复用。

(4) Python 语言结构清晰,关键字相对较少,容易学习。

(5) Python 语言提供了丰富的 API 和工具,可作为扩展语言为各种应用开发接口,可编程接口方便对接当前主要的系统、函数库和应用程序,可在 C 或 C++ 中扩展。

(6) Python 语言开发的应用具有很好的可移植性和兼容性,可以在 UNIX、Mac、OS/2、MS-DOS、Windows 等各个版本的操作系统上运行。

(7) Python 语言语法限制严格,对代码行的缩进等编程习惯有严格要求。

(8) Python 语言提供了丰富的标准库和扩展库。Python 标准库功能齐全,提供了系统管理、网络通信、文本处理、文件处理、网页浏览器、数据库接口、电子邮件、密码系统、图形用户界面等操作。除了标准库,Python 还提供了操作系统管理、科学计算、自然语言处理、Web 开发、图形用户界面开发和多媒体应用等多个领域的高质量第三方扩展包。

### 1.1.2 Python 语言的应用领域

Python 语言目前已经广泛应用于多个领域,主要包括:

#### 1. 操作系统管理

Python 语言具有易读性好、效率高、代码重用性好、扩展性好等优势,适合用于编写操作系统管理脚本。Python 提供了操作系统管理扩展包 Ansible、Salt、OpenStack 等。

## 2. 科学计算

Python 开发环境很适合用于处理实验数据、制作图表或者开发科学计算应用程序。Python 科学计算扩展库包括了快速数组处理模块 NumPy、数值运算模块 SciPy、数据分析和建模库 Pandas、可视化和交互式并行计算模块 IPython 和绘图模块 matplotlib 等,其他开源科学计算软件包也为 Python 提供了调用接口,例如计算机视觉库 OpenCV、医学图像处理库 ITK、三维可视化库 VTK 等。

## 3. 自然语言处理

Python 语言本身可以完成文本处理任务,同时还拥有功能强大的第三方自然语言处理工具库,包括 NLTK、spaCy、Pattern、TextBlob、Gensim、PyNLPI、Polyglot、MontyLingua、BLLIP Parser、Query 等,提供了分词、词干提取、词性标注、语法分析、情感分析、语义推理、机器翻译等类库,以及机器学习的向量空间模型、分类算法和聚类算法等丰富的功能。

## 4. Web 编程

Python 提供了多种 Web 编程解决方案和模块,可以方便地定制服务器软件,Python 提供的 Web 应用开发框架有 Django 和 Pyramid 等,微型 Python Web 框架有 Flask 和 Bottle 等,提供的高级内容管理系统有 Plone 和 django CMS 等。提供的工具集包括 Socket 编程、CGI、Freeform、Zope、CMF、Plone、Silva、Nuxeo CPS、WebWare、Twisted Python、CherryPy、SkunkWeb、Quixote、4Suite Server、Spyce、Albatross、Cheetah、mod\_python 等,Python 的标准库支持的 Internet 协议包括 HTML、XML、JSON、E-mail processing、FTP、IMAP 等。

## 5. 图形用户界面开发

Python 提供的 GUI 编程模块包括 Tkinter、wxPython、PyGObject、PyQt、PySide、Kivy 等,用户可以根据需要编写出强大的跨平台用户界面程序。

## 6. 多媒体应用

Python 提供了丰富的多媒体应用模块,包括能进行二维和三维图像处理的 PyOpenGL 模块,以及可用于编写游戏软件的 PyGame 模块等。

# 1.2 Python 版本和开发环境

## 1.2.1 Python 语言的版本

Python 语言是由 Guido van Rossum 在 1989 年底于荷兰国家数学和计算机科学研究所设计的,Python 的设计基于多种计算机语言,包括 ABC、Modula-3、C、C++、Algol-68、SmallTalk、UNIX Shell 和其他的脚本语言。

Python 语言的第一个公开发行人版本发行于 1991 年,目前主要使用的版本是 Python 2 和 Python 3。Python 开发团队同时维护 Python 2.x 和 Python 3.x 两个系列,Python 3.x 的发行时间并不一定晚于 Python 2.x。

本书编写完成时最新版本为 Python 2.7.13 和 Python 3.6.2,更多版本的更新可以关注 Python 官方网站 <https://www.python.org/>。

由于 Python 3 在设计时不考虑向下兼容,还有很多 Python 第三方扩展库目前尚未支

持 Python 3,因此作为 Python 的初学者,选择合适的版本是首要问题。选择 Python 版本的参考原则如下:

(1) 如果开发的应用对 Python 的版本有特殊要求,应该按此要求选择 Python 的版本。

(2) 如果在开发中需要使用特定的第三方扩展库,要注意确定是否与 Python 3 兼容,如果不兼容建议选择 Python 2。

如果没有特别说明,本书后续部分的讲解基于 Python 2.7 版本。

## 1.2.2 Python 的集成开发环境

Python 是一门跨平台的语言,集成开发环境可以提供 Python 程序开发环境的各种应用程序,一般包括代码编辑器、编译器、调试器和图形用户界面等工具,同时集成代码编写功能、分析功能、编译功能、调试功能等功能于一体。使用 Python 集成开发环境,可以帮助开发者提高开发的速度和效率,减少失误,也方便管理开发工作和组织资源。

常用的 Python 集成开发环境主要有:

### 1. IDLE

IDLE 是 Python 内置的集成开发环境。当安装好 Python 以后,IDLE 就自动安装好了。基本功能包括语法加亮、段落缩进、基本文本编辑、调试程序等。

### 2. Anaconda

Anaconda 是 Python 的集成安装软件,完全开源和免费。其中默认安装 Python、IPython、集成开发环境 Spyder 和众多流行的科学、数学、工程、数据分析的 Python 包,支持 Linux、Windows、Mac 等操作系统平台,支持 Python 2.x 和 Python 3.x,可在多版本 Python 之间自由切换。

Anaconda 额外的加速、优化是收费的,对于学术用途可以申请免费的 License。

Anaconda 的下载和安装详见 1.4.3 节。

### 3. Spyder

Spyder 是一个强大的开放源代码的交互式跨平台 Python 语言科学运算开发环境,集成了 NumPy、SciPy、Matplotlib 与 IPython 等开源软件。提供高级的代码编辑、交互测试、调试等特性,支持包括 Windows、Linux 和 OS X 操作系统。

Spyder 官方下载网址为 <https://pypi.python.org/pypi/spyder>。

### 4. Eclipse+PyDev

PyDev 是 Eclipse 开发的 Python 集成开发环境,支持 Python、Jython 和 IronPython 的开发。Eclipse+PyDev 插件适合开发 Python Web 应用,功能包括:自动代码完成、语法高亮、代码分析、调试器,以及内置的交互浏览器。

PyDev 官方下载网址为 <http://pydev.org/>。

### 5. PyCharm

PyCharm 是 JetBrains 开发的 Python 集成开发环境,功能包括:调试器、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制等,并支持 Google App Engine 和 IronPython。

PyCharm 专业版是商业软件,提供部分功能受限制的免费简装版本,官方下载地址为 <http://www.jetbrains.com/pycharm/>。



## 6. Wing

Wing 是一个功能强大的 Python 集成开发环境,兼容 Python 2. x 和 Python 3. x,可以结合 Tkinter、mod\_wsgi、Django、matplotlib、Zope、Plone、App Engine、PyQt、PySide、wxPython 等 Python 框架使用,支持测试驱动开发,集成了单元测试和 Django 框架的执行和调试功能,支持 Windows、Linux、OS X 等操作系统。

Wing 的专业版是商用软件,也提供了免费的简装版本,官方下载地址为 <https://wingware.com/>。

## 1.3 Python 的安装和设置

### 1.3.1 Python 的下载

Python 在不同操作系统平台上的安装和配置过程基本一致,本书基于 Windows 10 和 Python 2.7 搭建 Python 开发环境。

首先下载 Python 安装程序,步骤如下:

(1) 打开 Python 官方网站下载页面 <https://www.python.org/downloads/>,显示如图 1-1 所示页面。

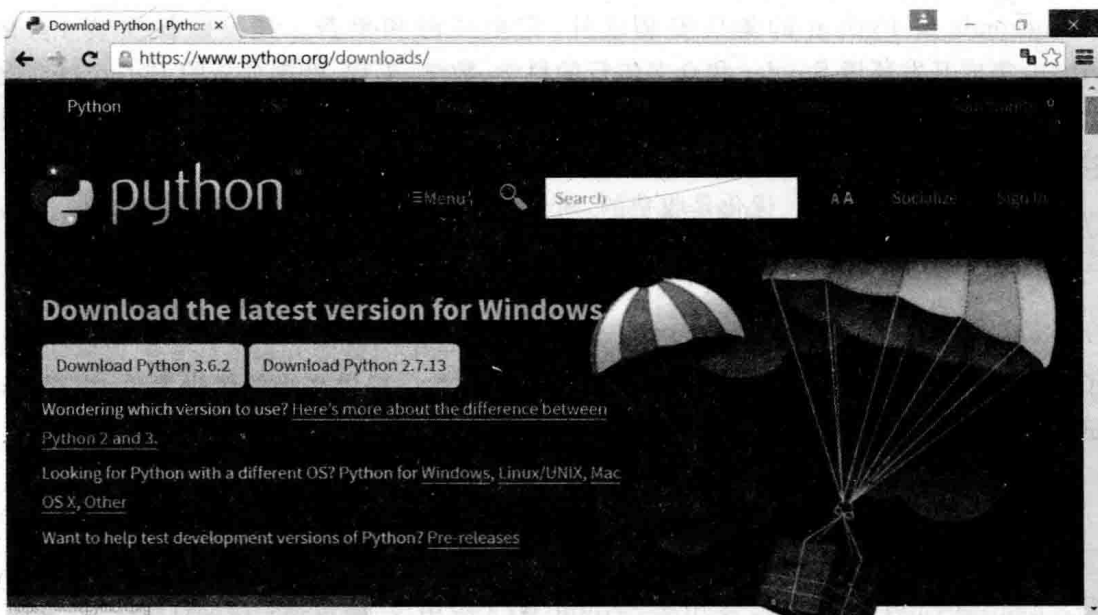


图 1-1 Python 官方网站下载页面

(2) 下载 Python 安装程序。例如单击图 1-1 中的“Download Python 2.7.13”按钮,下载安装文件 Python 2.7.13.msi,共 18.3MB。

### 1.3.2 Python 的安装

Python 的安装过程如下:

(1) 双击下载的安装文件 Python 2.7.13.msi,启动如图 1-2 所示的安装界面。