



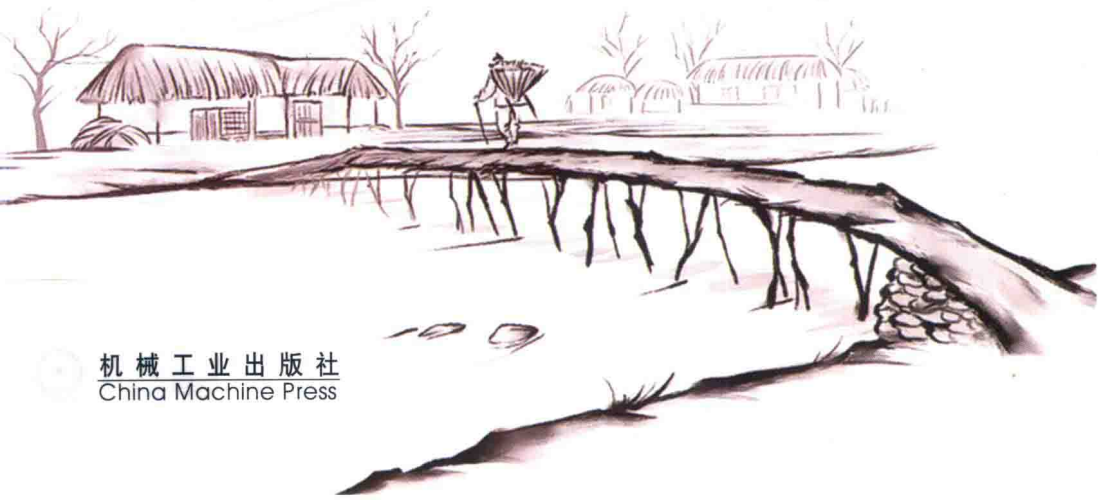
从零开始学 Python 程序设计

适用于Python3.5以上版本

从基本语法入手，兼顾理论与实践应用，轻松掌握Python程序设计语言

吴惠茹 等编著

丰富的范例程序，让初学者快速上手
精要的表达方式，强化编程与排错能力
精心设计的课后习题，检验学习效果



机械工业出版社
China Machine Press

从零开始学

Python

程序设计

吴惠茹 等编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

从零开始学Python程序设计 / 吴惠茹等编著. —北京: 机械工业出版社, 2017.11

ISBN 978-7-111-58381-3

I. ①从… II. ①吴… 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 (2017) 第266213号

Python语言是面向对象的模块化设计语言, 它易于学习、易于维护、可读性强, 适合初学者作为第一门程序设计语言。

本书以 Python 3.5 版本作为教学版, 针对初学者的特点, 分为五篇内容进行全面讲解: 第一篇为基础入门篇 (第1~3章), 介绍语言的基本功能; 第二篇为有序和无序篇 (第4~6章), 介绍控制台应用程序和窗口应用程序; 第三篇为标准函数篇 (第7和8章), 介绍标准函数库的应用; 第四篇为面向对象篇 (第9~11章), 介绍封装、继承和多态; 第五篇为高级篇 (第12~15章), 介绍Python使用各种模块来处理数据流的高级应用。全书辅以丰富的范例程序和精简的表达方式来降低学习难度, 在每个章节的最后提供课后习题, 倡导读者多动手实践。达到举一反三, 快速掌握Python语言的编程方法。

从零开始学Python程序设计

出版发行: 机械工业出版社 (北京市西城区百万庄大街22号 邮政编码: 100037)

责任编辑: 夏非彼 迟振春

责任校对: 王 叶

印 刷: 中国电影出版社印刷厂

版 次: 2018年1月第1版第1次印刷

开 本: 188mm×260mm 1/16

印 张: 28.25

书 号: ISBN 978-7-111-58381-3

定 价: 79.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前言

“大蟒蛇”出没！Python程序设计语言来了！

Python的创始人Guido van Rossum是Monty“大蟒蛇”飞行马戏团的一位爱好者。在1989年圣诞节期间，他决心开发一个新的脚本解释程序，并把这个程序设计语言取名为Python（这个英文单词的意思是“巨蛇”或者“蟒蛇”）。

本书内容共分为5篇：基础入门篇、有序和无序篇、标准函数篇、面向对象篇和高级篇。

基础入门篇（第1~3章）

踏上学习之旅时，首先把重点放在Python语言的基础语法上。Python本身提供了IDEL软件作为Python程序的集成开发环境。本篇从Python内置的类型讲起，与其他程序设计语言不同的是，每个变量都指向引用的对象。Python为处理数值提供了丰富的功能，如整数、浮点数和复数，甚至还可以使用有理数（分数）。对于改变程序语句的流程控制，可根据Python的简明原则，if/else条件语句以及for、while循环语句就可以“行遍天下”。

有序和无序篇（第4~6章）

掌握了Python的基本语法，当然就可以“游历天下”了。在本篇中，我们先来认识序列类型的字符串String、列表List和元组Tuple，映射类型的字典Dictionary，无序类型的集合Set。对于Python来说，即使是单个字符也是字符串，所以处理字符串有相当多的方法，如切片、索引、搜索和结合。而可变数据列表List类型和不可变的元组Tuple类型，与一般程序设计语言中数组的不同之处是，它们可以存放不同类型的元素。Dictionary类型以key和value来形成对应关系（映射关系），而集合Set类型也支持数学的集合运算。

标准函数篇（第7和8章）

Python有强大的标准函数库，内容包罗万象，本书无法逐一介绍。本篇首先介绍的是自定义函数，随后综合了各个章节使用过的内置函数。本篇对于函数中接收数据的参数和进行传递的参数有较多的介绍。在学习导入模块时，会介绍处理与日期、时间有关的模块。

面向对象篇（第9~11章）

以面向对象为基础，探讨面向对象程序设计的3个特性：继承（Inheritance）、封装

(Encapsulation) 和多态 (Polymorphism)。其他程序设计语言会以构造函数来创建、初始化对象。Python则分两个阶段，先以 `__new()` 方法创建对象，再以 `__init__` 初始化对象。所有的类、属性和方法都是公有的，想要封装，可借助属性修饰器 (`@property`) 或者用下划线 “_” 来指明它是私有的。继承采用多重机制，不过未进行更深入的探讨，而是以单一继承来介绍 `is_a` 和 `has_a` 的用法。

在很多情况下，编写程序都会发生异常（或称例外），除了可以采用 `try`、`except`、`finally` 语句之外，还可以搭配 `raise`、`assert` 语句从程序代码中抛出异常。

高级篇（第12~15章）

Python使用IO模块来处理数据流，即以文字和二进制数据配合功能强大的内置函数 `open()`。GUI以tkinter为主，简单地介绍Label、Entry、Text、Radiobutton、Checkbutton和Button组件，以及用于版面布局的 `pack()`、`grid()` 和 `place()` 方法。标准对话框的 `messagebox`、`simpledialog`、`filedialog`、`colorchooser` 可用于提供信息，如输入简单数据、打开文件和产生调色板。本篇还介绍了制作菜单的Menu组件以及有关鼠标、键盘的事件。最后一章以Django软件包来说明MVT架构，大致介绍前端Web、后端数据库的运行原理。

本书由吴惠茹主编，同时参与编写工作的还有施妍然、王国春、郭丹阳、孟宗斌、魏忠波、王翔、关静、卞诚君、周晓娟、闫秀华、刘雪连、孙学南等。如果读者在学习过程中遇到无法解决的问题，或者对本书持有意见或建议，请电子邮件联系 booksaga@126.com。

本书范例程序的下载地址如下：

<http://pan.baidu.com/s/1bo7kMKZ>（注意区分数字与字母的大小写）

由于编者水平有限，疏漏在所难免，恳请广大读者批评指正。

编者
2017年9月

目 录

前言

第1章 Python的世界	1
1.1 轻松搞定Python	1
1.1.1 Python的起源	2
1.1.2 Python的版本	3
1.1.3 安装Python	3
1.1.4 测试Python 3.5软件	8
1.1.5 Python 3.5有什么名堂	10
1.2 Python的IDE-IDLE	12
1.2.1 有哪些IDE软件	12
1.2.2 Python Shell	13
1.2.3 IDLE的环境设置	18
1.3 Python的编写风格	26
1.3.1 第一个Python程序	27
1.3.2 程序的缩排和注释	28
1.3.3 语句的分行和合并	28
1.3.4 程序的输入和输出	29
章节回顾	29
课后习题	30
第2章 Python基本语法	32
2.1 变量	32
2.1.1 标识符的命名规则	32
2.1.2 保留字和关键字	33
2.1.3 给变量赋值	33
2.2 认识Python的内置类型	37
2.3 Python的数据类型	38
2.3.1 整数类型	38
2.3.2 使用浮点数	41
2.3.3 复数类型	44

2.3.4	更精确的Decimal类型	44
2.3.5	例外——分数	47
2.4	使用表达式	48
2.4.1	算术运算符	48
2.4.2	赋值运算符	51
2.4.3	比较运算符	52
2.4.4	逻辑运算符	52
2.4.5	位运算符	53
	章节回顾	55
	课后习题	55
第3章	流程结构	57
3.1	程序设计语言结构	57
3.2	根据条件进行选择	58
3.2.1	if语句采用单向判断	58
3.2.2	if/else语句做双向判断	60
3.2.3	多重选择	62
3.3	循环语句	64
3.3.1	for循环	64
3.3.2	while循环	67
3.3.3	嵌套循环	69
3.3.4	产生随机数的random模块	71
3.4	特殊流程控制	72
3.4.1	break语句	73
3.4.2	continue语句	74
	章节回顾	75
	课后习题	75
第4章	序列类型与字符串	78
4.1	序列类型	78
4.1.1	序列与迭代器	79
4.1.2	创建序列数据	80
4.1.3	序列元素及其操作	80
4.1.4	与序列有关的函数	83
4.2	字符串	84
4.2.1	创建字符串	84
4.2.2	切片的概念	87
4.2.3	转义字符	91

4.2.4 字符串常用函数.....	92
4.3 格式化字符串.....	100
4.3.1 %运算符——格式字符串.....	100
4.3.2 内置函数format().....	102
4.3.3 str.format()方法.....	104
章节回顾.....	107
课后习题.....	108
第5章 元组与列表.....	111
5.1 元组.....	111
5.1.1 创建元组.....	111
5.1.2 读取元组元素.....	115
5.1.3 元组的相关操作.....	116
5.2 列表的基本操作.....	119
5.2.1 创建列表.....	120
5.2.2 与列表有关的方法.....	122
5.2.3 数据排序.....	124
5.2.4 列表推导式.....	128
5.3 二维列表.....	133
5.3.1 产生矩阵.....	133
5.3.2 读取矩阵.....	134
5.3.3 矩阵与列表推导式.....	136
5.3.4 不规则列表.....	139
5.4 列表与复制.....	141
5.4.1 列表与浅复制.....	141
5.4.2 copy模块的copy()方法.....	143
5.4.3 deepcopy()方法复制对象本身.....	144
章节回顾.....	145
课后习题.....	145
第6章 字典和集合.....	148
6.1 字典.....	148
6.1.1 映射类型与字典.....	148
6.1.2 创建字典.....	149
6.1.3 键、值的操作.....	153
6.1.4 与字典有关的方法.....	158
6.1.5 字典推导式.....	161
6.2 默认字典和有序字典.....	163

6.2.1 默认字典.....	163
6.2.2 有序字典.....	165
6.3 集合.....	168
6.3.1 什么是哈希.....	169
6.3.2 创建集合对象.....	170
6.3.3 集合的相关操作.....	171
6.3.4 集合的数学计算.....	173
6.3.5 集合推导式.....	179
章节回顾.....	179
课后习题.....	180
第7章 函数.....	182
7.1 认识函数.....	182
7.1.1 Python的内置函数.....	183
7.1.2 函数基础.....	186
7.1.3 定义函数.....	186
7.1.4 调用函数.....	187
7.1.5 返回值.....	188
7.2 参数的基本机制.....	190
7.2.1 参数如何传递.....	191
7.2.2 位置参数有顺序性.....	193
7.2.3 默认参数值.....	193
7.2.4 关键字参数.....	197
7.3 可长可短的参数行.....	199
7.3.1 形式参数的*表达式.....	199
7.3.2 **表达式与字典配合.....	202
7.3.3 *运算符拆分可迭代对象.....	205
7.3.4 **运算符拆分字典对象.....	207
7.4 更多函数的讨论.....	211
7.4.1 作用域.....	211
7.4.2 函数是第一等“公民”.....	214
7.4.3 局部函数与闭包.....	216
7.4.4 lambda函数.....	218
7.4.5 递归.....	223
章节回顾.....	225
课后习题.....	226

第8章 模块与函数库.....	230
8.1 导入模块.....	230
8.1.1 命令行的参数列表.....	230
8.1.2 import/as语句.....	232
8.1.3 from/import语句.....	232
8.1.4 命名空间与dir()函数.....	233
8.2 使用模块.....	234
8.2.1 模块搜索路径.....	234
8.2.2 自定义模块.....	235
8.2.3 reload()函数.....	238
8.3 认识函数库.....	240
8.3.1 随机数.....	240
8.3.2 pprint模块让打印更有看头.....	243
8.4 处理时间与日期.....	245
8.4.1 获取时间戳的time模块.....	245
8.4.2 提供日期和时间的datetime模块.....	249
8.4.3 calendar模块.....	257
章节回顾.....	260
课后习题.....	260
第9章 面向对象基础.....	263
9.1 认识面向对象.....	263
9.1.1 对象具有属性和方法.....	264
9.1.2 类是对象的蓝图.....	264
9.1.3 抽象化概念.....	265
9.2 类与对象.....	265
9.2.1 定义类.....	266
9.2.2 先构造再初始化对象.....	270
9.2.3 设置、检查对象的属性.....	274
9.2.4 处理对象的特殊方法.....	275
9.3 类与修饰器.....	278
9.3.1 类也有属性.....	278
9.3.2 认识修饰器.....	279
9.3.3 类修饰器.....	284
9.3.4 类方法和静态方法.....	288
9.4 重载运算符.....	290
9.4.1 重载算术运算符.....	290

9.4.2	对重载加号运算符做更多了解.....	291
9.4.3	重载比较大小的运算符.....	295
	章节回顾.....	296
	课后习题.....	297
第10章	浅谈继承机制.....	299
10.1	认识继承.....	299
10.1.1	与继承有关的名词.....	299
10.1.2	继承概念.....	299
10.1.3	特化和泛化.....	300
10.1.4	组合.....	301
10.2	继承机制.....	301
10.2.1	产生继承.....	301
10.2.2	子类覆盖父类的方法.....	306
10.2.3	以特性存取属性.....	309
10.3	抽象类与多态.....	313
10.3.1	定义抽象类.....	313
10.3.2	多态.....	314
10.3.3	组合.....	316
	章节回顾.....	317
	课后习题.....	318
第11章	异常处理机制.....	320
11.1	什么是异常.....	320
11.1.1	程序错误.....	320
11.1.2	引发异常.....	321
11.1.3	内置的Exception类.....	321
11.2	异常处理情况.....	325
11.2.1	设置捕捉器.....	325
11.2.2	try语句究竟是如何工作的.....	328
11.2.3	try/else语句.....	329
11.2.4	try/finally语句.....	330
11.3	以程序抛出异常.....	332
11.3.1	raise语句引发异常.....	332
11.3.2	assert语句.....	335
11.3.3	用户自定义异常处理.....	336
	章节回顾.....	338
	课后习题.....	338

第12章 数据流与文件.....	340
12.1 认识io模块.....	340
12.1.1 io模块提供了什么.....	340
12.1.2 io模块提供的方法.....	341
12.1.3 TextIOBase类.....	343
12.2 文本文件的读和写.....	346
12.2.1 文件和指定模式.....	346
12.2.2 with/as语句.....	350
12.2.3 读取文本文件.....	351
12.2.4 浅谈文字编码.....	353
12.3 二进制数据.....	356
12.3.1 认识byte与bytearray.....	356
12.3.2 struct模块与二进制数据.....	357
12.4 文本文件并非只有文字.....	359
12.4.1 CSV格式.....	360
12.4.2 JSON格式.....	361
章节回顾.....	364
课后习题.....	364
第13章 GUI.....	366
13.1 Python GUI.....	366
13.1.1 GUI相关软件包.....	366
13.1.2 认识tkinter软件包.....	367
13.1.3 编写一个简单的窗口程序.....	369
13.2 管理版面布局.....	370
13.2.1 Frame为容器.....	371
13.2.2 版面布局——pack()方法.....	374
13.2.3 grid()方法用行、列决定位置.....	377
13.2.4 用坐标定位的place()方法.....	378
13.3 处理文字的组件.....	379
13.3.1 Label组件.....	379
13.3.2 Entry接收单行文字.....	381
13.3.3 Text接收多行文字.....	382
13.3.4 Button组件.....	384
13.4 选项组件.....	386
13.4.1 Checkbutton组件.....	386
13.4.2 Radiobutton组件.....	388

章节回顾.....	389
课后习题.....	390
第14章 GUI其他组件	392
14.1 对话框.....	392
14.1.1 messagebox组件.....	392
14.1.2 文件对话框.....	396
14.1.3 提供颜色的colorchooser.....	399
14.1.4 简易对话框.....	401
14.2 菜单与事件处理.....	403
14.2.1 使用Menu组件创建菜单.....	403
14.2.2 事件的触发.....	408
14.2.3 事件的处理.....	410
14.2.4 产生快捷菜单.....	412
14.3 Canvas绘制图形.....	413
14.3.1 认识Canvas组件.....	414
14.3.2 用鼠标绘制线条.....	415
14.3.3 绘制几何图形.....	417
章节回顾.....	418
课后习题.....	419
第15章 Django与Web.....	421
15.1 认识Django软件包.....	421
15.1.1 Web相关名词.....	421
15.1.2 安装Django.....	422
15.1.3 创建Django的Web项目.....	422
15.1.4 配置文件settings.py.....	425
15.2 视图、URL与简单模板.....	426
15.2.1 浅谈正则表达式.....	426
15.2.2 第一个视图程序.....	427
15.2.3 第二个视图程序.....	431
15.2.4 使用模板.....	433
15.3 模型与数据库.....	435
15.3.1 建立模型.....	435
15.3.2 从数据表读取数据.....	437
章节回顾.....	438
课后习题.....	439

第1章 Python的世界

学习导引

- 从 Python 的起源、版本到软件安装来初步认识 Python
- 编写程序代码，所使用的工具从简单的记事本到 Python 官方的 IDLE，虽然功能不是非常强大，但是它还是可以提供不少协助的
- 第一个 Python 程序，熟悉它的语言结构和风格

如何走进Python的世界？就从Python的起源开始探索吧！先来认识Python程序设计语言的发展及其版本，首先安装Python官方的CPython软件到Windows操作系统中，包含编写Python程序时可搭配的IDE软件，并进一步熟悉IDLE的操作界面。然后以一个简易的Python程序来了解Python程序设计语言的编写风格。

1.1 轻松搞定Python

Python程序设计语言有什么特色？相对于其他计算机程序设计语言，Python的魅力何在？Python官方自己的注释为：“简单易学，语法简洁，编译式的计算机语言”，由此可以看出这门语言的端倪。

注意

任何一种程序设计语言都需要编译程序或解释器把源代码转译成计算机能够理解的机器码。

- 编译程序（Compiler）：需要完整的源代码才会进行编译，生成可执行程序，最后链接函数库才能执行。
- 解释器（Interpreter，或称解释程序）：在执行时，动态地将程序代码逐句翻译成机器码。

更通俗的说法，编译程序就像是一个翻译软件，必须有整篇文章才进行翻译；解释器则像是一位可以跟随你到处走的口译人员，随时都可以进行翻译。

关于Python为什么出色的另一个佐证：从TIOBE Software在2017年3月所公布的世界程序设计语言排名可以看出，Python已上升到第5名，如图1-1所示。

Mar 2017	Mar 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.384%	-4.14%
2	2		C	7.742%	-6.86%
3	3		C++	5.184%	-1.54%
4	4		C#	4.409%	+0.14%
5	5		Python	3.919%	-0.34%
6	7	^	Visual Basic .NET	3.171%	+0.61%
7	6	v	PHP	3.009%	+0.24%
8	8		JavaScript	2.667%	+0.33%
9	11	^	Delphi/Object Pascal	2.544%	+0.54%
10	14	^	Swift	2.268%	+0.68%
11	9	v	Perl	2.261%	+0.01%
12	10	v	Ruby	2.254%	+0.02%
13	12	v	Assembly language	2.232%	+0.39%
14	16	^	R	2.015%	+0.73%
15	13	v	Visual Basic	2.008%	+0.33%
16	15	v	Objective-C	1.997%	+0.54%
17	48	^	Go	1.982%	+1.78%
18	18		MATLAB	1.854%	+0.66%
19	19		PL/SQL	1.672%	+0.48%
20	26	^	Scratch	1.472%	+0.70%

图 1-1

现在就一起进入Python的世界，一起了解它的语言特色，感受它的无穷乐趣。

1.1.1 Python的起源

Python程序设计语言诞生于1989年，创始人Guido van Rossum（吉多·范罗苏姆）创立了新的脚本语言（Script Language），发展至今已有20多年的历史。Python是一种高级语言，支持面向对象。语言本身能跨越平台，无论是Linux、Mac还是Windows都能畅行无阻。



吉多·范罗苏姆之所以选中Python作为程序名称，是因为他是蒙提·派森飞行马戏团（Monty Python's Flying Circus, BBC电视剧）的粉丝。他参考了ABC（All Basic Code）程序设计语言、C语言和其他一些程序设计语言来构造Python程序设计语言。

☞ Python 的类型是动态、强类型

Python采用动态类型。它的语法在程序代码执行时才会检查，所以在某些情况下运行时可能会抛出异常。同时，Python也是强类型语言，不同的数据类型采用高标准规范。例如，在Python语言中，数字加字符串这种没有明确定义的运算是非法的。

☞ Python 是脚本语言也是程序设计语言

管理操作系统时，系统管理员会依据常例编写程序，让计算机按表操课。这些程序必须借助操作系统的Shell（或称命令壳、命令解析器）配合脚本语言来编写。Python既能支持脚本

语言，也能像一般的程序设计语言应用于程序实现中。

☞ Python 是胶水语言，资源丰富

也有人将Python视为胶水语言（Glue Language）的一种，这源自于脚本语言（Script Language，或称描述语言）的发展。在Linux系统下，能把相关功能的程序（可能由不同的程序设计语言所编写）如同胶水般“粘”在一起。

Python程序设计语言除了本身拥有功能完备的标准函数库外，也能加入第三方函数库（或称第三方软件包），轻松完成很多常见的任务。

- Web 应用：可以使用 Django、Flask 或 Tornado 等软件包。
- GUI 开发：支持的软件包有 Tkinter、wxPython、PyQt 等。
- 操作系统：除了 Windows 之外，多数操作系统都将 Python 内置为标准组件，可以在“命令提示符”下执行。而 Linux 发行的版本会以 Python 语言编写成安装程序，如 Ubuntu 的 Ubiquity。

1.1.2 Python的版本

有关Python版本的发布情况，可从表1-1简单了解Python较重要的版本。

表1-1 Python较重要的版本

版本	简介
2.0	2000年10月16日发布，支持Unicode和垃圾回收机制
2.7	2010年7月3日发布
3.0	2008年12月3日发布，此版不完全兼容之前的Python源代码
3.5	发布于2015年9月13日，本书采用版本

一般来说，软件语言的版本都是不断更新累进的。但是有趣的是，Python语言却是Python 2.x和Python 3.x同时存在，而彼此之间并非完全兼容。Python 2.7是Python官方于2.x系列所发布的最新版本，由于资源较丰富，因此第三方函数库以它为基础的依然不少。Python 3.x（也称Python 3000或Py3k）有不能向下兼容的不便，提供支持的软件包也较有限。无论如何，它们都属于Python程序设计语言！考虑到越来越多的人会转向Python 3.x，本书会以Python 3.5为主进行介绍，带领读者来了解Python的语法和结构。

1.1.3 安装Python

翻译Python程序代码必须通过Python集成环境所提供的解释器。究竟有哪些解释器（Interpreter）呢？由表1-2进行说明。

表1-2 Python的不同解释器

解释器	简介
CPython	官方的解释器，以C语言编写，本书使用的软件
ZhPy	中文称为周蟒，可使用繁/简中文语句编写程序

解释器	简介
PyPy	使用Python语言编写, 执行速度会比CPython快
IronPython	可调用.NET平台的函数库, 将Python程序编译成.NET程序
Jython	Java语言编写, 可以直接调用Java函数库

现阶段讨论以Python基本语法为主, 因此以Python官方软件CPython为载体, 包含Python 3.5和pip。

- Python 3.5: 由CPython提供的解释器, Python官方团队制作, 其源代码完全开放, 具有标准架构, 让他人能遵循此标准制定Python的执行环境, 在本书中会直接以Python来称呼它。
- pip: 用来管理Python第三方函数库的工具, 内置于CPython软件中, 安装时能通过选项加入(可参考Python软件安装的步骤4)。

本书以Windows操作系统为开发环境进行上述软件的安装。Python官方网站网址为<https://www.python.org/>。

步骤01 进入Python官网, 找到①Downloads, 展开选项之后, ②选择下载Windows环境的版本, 如图1-2所示。

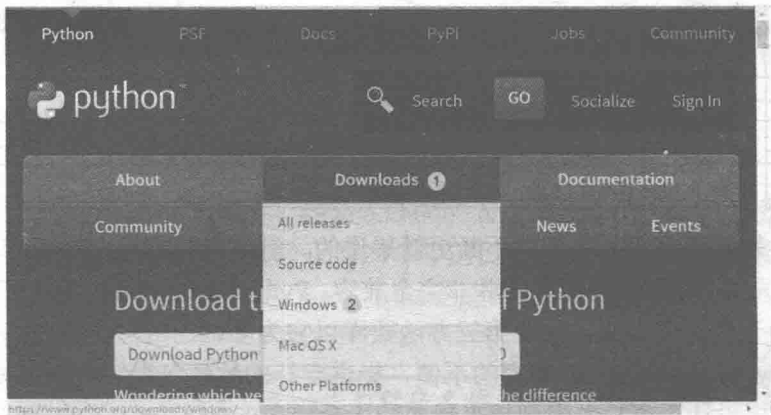


图 1-2

步骤02 完成软件的下载后, 双击即可启动Python安装软件。

步骤03 进入Python安装界面, 单击①Customize installation选项来安装, 在界面下方勾选②Add Python 3.5 to PATH复选框, 如图1-3所示。

注意

Add Python 3.5 to PATH

表示要将Python软件的执行路径加到Windows的环境变量中。如此一来, 在“命令提示符”下就可以执行Python指令了。