

TURING

图灵程序设计丛书



Peachpit
Press

[加] Toby Donaldson 著
袁国忠 译



Python编程入门

(第3版)

Python: Visual QuickStart Guide
Third Edition



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

[加] Toby Donaldson 著
袁国忠 译



Python编程入门

(第3版)

Python: Visual QuickStart Guide
Third Edition

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Python编程入门：第3版 / (加) 唐纳森
(Donaldson, T.) 著；袁国忠译. -- 北京：人民邮电出
版社，2013. 12

(图灵程序设计丛书)

书名原文: Python:visual quickstart guide,
third edition

ISBN 978-7-115-33374-2

I. ①P… II. ①唐… ②袁… III. ①软件工具—程序
设计 IV. ①TP311.56

中国版本图书馆CIP数据核字(2013)第242069号

内 容 提 要

本书是图文并茂的 Python 学习参考书，书中并不包含深奥的理论或者高级应用，而是以大量来自实战的例子、屏幕图和详细的解释，用通俗易懂的语言结合常见任务，对 Python 的各项基础知识进行了介绍，以帮助读者成为一名真正的 Python 程序员。

本书适合各个层次的 Python 开发人员阅读。

-
- ◆ 著 [加] Toby Donaldson
 - 译 袁国忠
 - 责任编辑 李 瑛
 - 责任印制 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>
 - 三河市海波印务有限公司印刷
 - ◆ 开本：800×1000 1/16
 - 印张：13.25
 - 字数：212千字 2013年12月第1版
 - 印数：1-3 000册 2013年12月河北第1次印刷
 - 著作权合同登记号 图字：01-2013-6736号

定价：39.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京崇工商广字第 0021 号

版权声明

Authorized translation from the English language edition, entitled *Python: Visual QuickStart Guide, Third Edition* by Toby Donaldson, published by Pearson Education, Inc., publishing as Peachpit Press. Copyright © 2014 by Toby Donaldson.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Pearson Education Inc.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

致 谢

感谢Clifford Colby和Scout Festa在本书第三版出版过程中贡献的专业知识和耐心；感谢西蒙弗雷泽大学（SFU）的学生们让我明白了学习Python的最佳方式；感谢西蒙弗雷泽大学计算机系的John Edgar和其他教师带给我愉悦的共事体验；感谢Bonnie、Thomas和Emily建议我不要在致谢中喋喋不休。最后要特别感谢Guido van Rossum和Python社区的其他成员，感谢你们打造了一种如此趣味盎然的编程语言。

目 录

第 1 章 编程简介	1	2.7.1 列出模块中的函数	20
1.1 Python 语言	2	2.7.2 打印文档字符串	21
1.2 Python 适合用于做什么	3	2.8 类型转换	22
1.3 程序员如何工作	4	2.8.1 将整数和字符串转换为浮点数	22
1.4 安装 Python	6	2.8.2 将整数和浮点数转换为字符串	22
1.4.1 在 Windows 系统上安装 Python	6	2.8.3 将浮点数转换为整数	23
1.4.2 在 Mac 系统上安装 Python	7	2.8.4 将字符串转换为数字	23
1.4.3 在 Linux 系统上安装 Python	7	2.9 变量和值	24
第 2 章 算术、字符串与变量	9	2.10 赋值语句	26
2.1 交互式命令 shell	10	2.11 变量如何引用值	28
2.1.1 shell 提示符	10	2.11.1 赋值时不复制	28
2.1.2 记录	10	2.11.2 数字和字符串是不可变的	28
2.2 整数算术	11	2.12 多重赋值	29
2.2.1 整除	11	第 3 章 编写程序	31
2.2.2 求值顺序	12	3.1 使用 IDLE 的编辑器	32
2.2.3 长度不受限制	12	3.1.1 在 IDLE 中编写程序	32
2.3 浮点数算术	13	3.1.2 从命令行运行程序	33
2.3.1 浮点数字面量	13	3.1.3 从命令行调用 Python	33
2.3.2 溢出	14	3.2 编译源代码	35
2.3.3 精度有限	14	3.3 从键盘读取字符串	36
2.3.4 复数	15	3.3.1 跟踪程序	36
2.4 其他数学函数	16	3.3.2 从键盘读取数字	38
2.4.1 使用返回值	16	3.4 在屏幕上打印字符串	39
2.4.2 导入模块	16	3.5 源代码注释	41
2.5 字符串	17	3.6 程序的组织	42
2.5.1 标识字符串	17	第 4 章 流程控制	43
2.5.2 字符串的长度	18	4.1 布尔逻辑	44
2.6 字符串拼接	19	4.1.1 逻辑相等	45
2.7 获取帮助	20		

4.1.2	逻辑与	45	6.1.1	负数索引	85
4.1.3	逻辑或	45	6.1.2	使用 for 循环访问字符	86
4.1.4	逻辑非	45	6.2	字符	87
4.1.5	计算较长的布尔表达式	46	6.3	字符串切片	89
4.1.6	计算包含圆括号的布尔表达式	46	6.3.1	获取切片的捷径	90
4.1.7	计算不包含圆括号的布尔表达式	47	6.3.2	使用负数索引的切片	91
4.1.8	短路求值	48	6.4	标准字符串函数	92
4.2	if 语句	49	6.4.1	测试函数	92
4.3	代码块和缩进	51	6.4.2	搜索函数	93
4.3.1	if/elif 语句	52	6.4.3	改变大小写的函数	94
4.3.2	条件表达式	53	6.4.4	设置格式的函数	94
4.4	循环	54	6.4.5	剥除函数	95
4.4.1	for 循环	54	6.4.6	拆分函数	95
4.4.2	while 循环	56	6.4.7	替换函数	96
4.5	比较 for 循环和 while 循环	59	6.4.8	其他函数	97
4.5.1	计算阶乘	59	6.5	正则表达式	98
4.5.2	计算用户输入的数字的总和	61	6.5.1	简单的正则表达式	98
4.5.3	计算未知个数字的总和	62	6.5.2	使用正则表达式匹配字符串	99
4.6	跳出循环和语句块	64	6.5.3	其他正则表达式	100
4.7	循环中的循环	66			
第 5 章	函数	67	第 7 章	数据结构	101
5.1	调用函数	68	7.1	type 命令	102
5.1.1	不返回值的函数	69	7.2	序列	103
5.1.2	给函数名赋值	69	7.3	元组	104
5.2	定义函数	70	7.3.1	元组是不可变的	105
5.3	变量的作用域	73	7.3.2	元组函数	106
5.4	使用 main 函数	75	7.4	列表	108
5.5	函数的参数	76	7.5	列表函数	110
5.5.1	按引用传递	76	7.6	列表排序	113
5.5.2	一个重要示例	77	7.7	列表解析	115
5.5.3	默认值	78	7.7.1	列表解析示例	116
5.5.4	关键字参数	79	7.7.2	使用列表解析进行筛选	117
5.6	模块	80	7.8	字典	118
5.6.1	创建 Python 模块	80	7.8.1	对键的限制	119
5.6.2	名称空间	82	7.8.2	字典函数	120
			7.9	集合	122
第 6 章	字符串	83	第 8 章	输入和输出	123
6.1	字符串索引	84	8.1	设置字符串格式	124

8.1.1	字符串插入	124	10.2	显示对象	156
8.1.2	转换说明符	125	10.3	灵活的初始化	160
8.2	格式字符串	126	10.4	设置函数和获取函数	162
8.3	读写文件	128	10.4.1	特性装饰器	163
8.3.1	文件夹	130	10.4.2	私有变量	166
8.3.2	当前工作目录	130	10.5	继承	168
8.4	检查文件和文件夹	131	10.6	多态	171
8.5	处理文本文件	134	10.6.1	实现 get_move 函数	172
8.5.1	逐行读取文本文件	134	10.6.2	玩游戏 Undercut	173
8.5.2	将整个文本文件作为一个 字符串进行读取	135	10.7	更深入地学习	175
8.5.3	写入文本文件	136	第 11 章	案例研究：文本统计	177
8.5.4	附加到文本文件末尾	136	11.1	问题描述	178
8.5.5	将字符串插入到文件开头	137	11.2	保留想要的字母	180
8.6	处理二进制文件	138	11.3	使用大型数据文件测试代码	182
8.7	读取网页	141	11.4	找出出现次数较多的单词	184
第 9 章	异常处理	143	11.5	将字符串转换为次数字典	187
9.1	异常	144	11.6	组织在一起	188
9.2	捕获异常	146	11.7	练习	190
9.2.1	try/except 块	148	11.8	最终的程序	192
9.2.2	捕获多种异常	149	附录 A	深受欢迎的 Python 包	195
9.2.3	捕获所有异常	149	附录 B	比较 Python 2 和 Python 3	199
9.3	清理操作	150	索引		203
第 10 章	面向对象编程	153			
10.1	编写类	154			

1

编程简介

深入探讨 Python 编程之前，先大致了解一下 Python 是什么及其可用于编写哪些类型的程序，这大有裨益。本章还将概述程序员所做的工作。最后将介绍如何安装 Python 及运行其自带的 IDLE 编辑器。

如果你是编程新手，本章可助你为学习 Python 编程语言做好准备。

如果你已掌握这些基本概念，可跳到有关如何安装 Python 和运行编辑器的章节。

本章内容

- Python 语言
 - Python 适合用于做什么
 - 程序员如何工作
 - 安装 Python
-

1.1 Python 语言

那么 Python 是什么呢? 简单地说, 它是一种计算机编程语言及一组配套的软件工具和库。Python 最初由 Guido van Rossum 于 20 世纪 90 年代初开发, 当前由世界各地的数十位程序员 (包括 van Rossum) 负责维护。

Python 易于理解和学习。相比于用其他大多数编程语言编写的程序, Python 程序更整洁: Python 几乎没有多余的符号, 且使用的是简单易懂的英语名称。

Python 语言的效率极高。精通 Python 后, 与使用其他大多数编程语言相比, 使用 Python 可在更短的时间内完成更多的工作。Python 支持但不强制你使用面向对象编程 (OOP)。

Python 自带了各种现成库, 供你在自己的程序中使用。有些 Python 程序员喜欢这样说: Python “开箱即可使用”。

Python 的一个极其实用的特点是易于维护。鉴于 Python 程序理解和修改起来相对容易, 程序员可轻松地确保它们紧跟潮流。在程序员所做的工作中, 程序维护所占的比例很可能高达甚至超过 50%, 因此在很多专业人士看来, Python 对维护的支持是个亮点。

最后, 说说名称 Python 的由来。据 Python 之父 Guido van Rossum 说, Python 是以喜剧团体 Monty Python (巨蟒小组) 的名字命名的。虽然这种起源充满喜庆色彩, 但 Python 当前使用的标识确乎是两条缠在一起的蛇 (可能是蟒蛇), 其中一条为蓝色, 另一条为黄色。

1.2 Python 适合用于做什么

虽然 Python 是一种通用语言，可用于编写任何类型的程序，但它最常用于编写下述应用程序。

- **脚本。**这些简短的程序自动执行常见的管理任务，如在系统中新增用户、将文件上传到网站、在不使用浏览器的情况下下载网页等。
- **网站开发。**作为快速创建动态网站的工具，Django (www.djangoproject.com)、Bottle (www.bottlepy.org) 和 Zope (www.zope.org) 等众多 Python 项目深受开发人员的欢迎。例如，深受欢迎的新闻网站 www.reddit.com 就是使用 Python 开发的。
- **文本处理。**Python 在字符串和文本文件处理方面提供了强大的支持，包括正则表达式和 Unicode。
- **科学计算。**网上有很多卓越的 Python 科学计算库，提供了用于统计、数学计算和绘图的函数。
- **教育。**鉴于 Python 简洁实用，越来越多的学校将其作为第一门编程教学语言。

当然，Python 并非对任何项目来说都是最佳选择，其速度通常比 Java、C#、C++ 等语言慢，因此开发新操作系统时不会使用 Python。

然而，需要最大限度地减少程序员花在项目上的时间时，Python 通常是最佳选择。

1.3 程序员如何工作

虽然对如何编写程序没有严格的规定，但大多数程序员都采用类似的流程。

该程序开发流程如下。

1. 确定程序要做什么，即搞清楚需求。

2. 编写源代码，这里是使用 Python 集成开发环境 IDLE 或其他文本编辑器编写 Python 代码。这一步通常最有趣也最具挑战性，要求你创造性地解决问题。Python 源代码文件使用扩展名 .py，如 web.py、urlexpand.py、clean.py 等。

3. 使用 Python 解释器将源代码转换为目标代码。Python 将目标代码存储在 .pyc 文件中，例如，如果源代码存储在文件 urlexpand.py 中，目标代码将存储在文件 urlexpand.pyc 中。

4. 运行或执行程序。就 Python 而言，通常紧接着第 2 步自动完成这一步。实际上，Python 程序员很少直接与目标代码（.pyc 文件）交互。

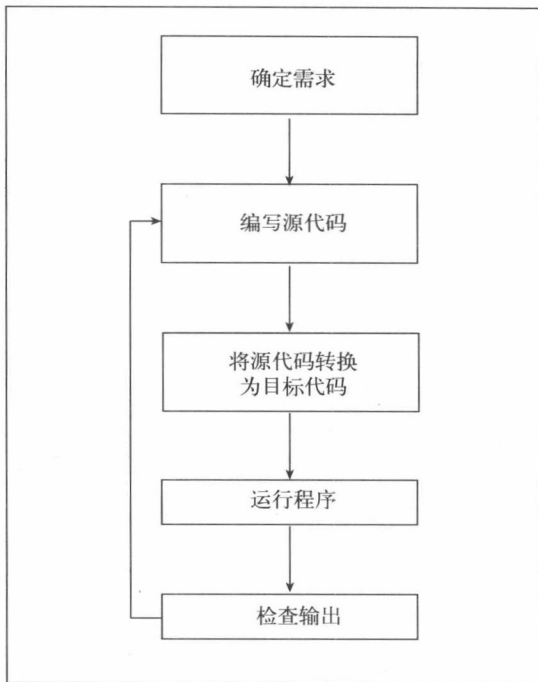


图 1-1 基本计算机程序编写步骤。检查程序输出时，通常会发现错误。为修复错误，必须回到步骤“编写源代码”

术语说明

我们通常将 .py 文件的内容称为程序、源代码或代码。

目标代码有时也称为可执行代码、可执行文件或软件。

5. 最后，检查程序的输出。如果发现错误，回到第 2 步并尽力修复错误。修复错误的过程称为调试。开发庞大或复杂的程序时，可能大部分时间都用在调试上，因此经验丰富的程序员设计程序时，会尽力采用可最大限度地减少调试时间的方式。

如图 1-1 所示，这是个循环往复的过程：编写程序，测试，修复错误，再测试……直到程序正确运行。

1.4 安装 Python

Python 是一种实践性语言，下面来看看如何在计算机上安装它。

1.4.1 在 Windows 系统上安装 Python

步骤如下。

1. 访问 Python 下载页面 www.python.org/download。

2. 选择最新的 Python 3 版本（其名称类似于 Python 3.x，其中 x 是一个较小的数字），这将打开相应的下载页面，其中说明了如何下载用于不同计算机系统的 Python。

3. 根据计算机使用的操作系统，单击相应的安装程序链接。例如，如果是 Windows 操作系统，单击 Windows x86 MSI Installer (3.x)。

4. 下载完毕后，双击安装程序以运行它。

5. 安装完成后（需要几分钟），通过测试看看是否正确安装了 Python。为此，打开“开始”菜单并选择“所有程序”，将看到一个与 Python 3.0 相关的选项（其背景通常为黄色）。选择其中的选项 IDLE (Python GUI)，一段时间后程序 IDLE 将启动，如图 1-2 所示。

6. 输入 $24 * 7$ 并按回车，应出现数字 168。

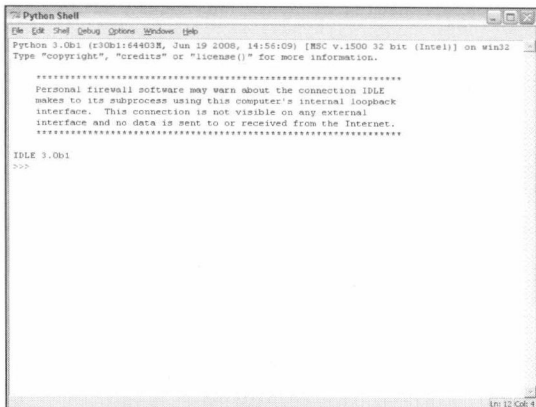


图 1-2 IDLE 编辑器的起始屏幕。第一行指出了当前使用的是哪个 Python 版本（这里为 3.0b1）

1.4.2 在 Mac 系统上安装 Python

OS X 自带并安装了一个 Python 版本，但该版本没有 IDLE 编辑器，通常也不是最新版本。要安装更新的 Python 版本，可按 www.python.org/download/mac/ 给出的说明做，也可从 www.pythonmac.org/packages/ 下载一个安装程序并运行它。下载安装程序时，务必选择正确的 Python 版本（3.0 或更高版本），并确保 Mac OS 版本号与你的操作系统版本号一致。

1.4.3 在 Linux 系统上安装 Python

如果你使用的是 Linux，很可能已经安装了 Python。要确认这一点，可打开命令行窗口并输入 `python`，如果输出与图 1-2 类似，说明 Python 运行正常。

务必检查版本号。本书介绍的是 Python 3，如果当前安装的是 Python 2.x 或更早的版本，就应安装 Python 3。

具体如何安装因 Linux 系统而异。例如，在 Ubuntu Linux 系统上，需要在 Synaptic Package Manager 中搜索 Python。你也可以访问 www.python.org/download，了解如何在 Linux 系统上安装 Python。

2

算术、字符串与变量

要学习编程，首先要了解基本的 Python 数据类型：整型（整数）、浮点数（带小数点的数字）和字符串。所有程序都使用这些（及其他）数据类型，因此牢固掌握它们的用法至关重要。

使用字符串的程序如此之多，所以 Python 提供了强大的字符串支持。本章将介绍字符串的基本知识，而第 6 章还将回过头来更深入地介绍。

我们还将介绍最重要的编程概念——变量。变量用于存储和操作数据，如果不使用几个变量，就很难编写出有用的程序。

与学习弹钢琴或说外语一样，学习编程的最佳方式也是多练。因此，本章将利用交互式命令 shell（IDLE）来介绍上述所有知识，你最好跟着做：在计算机上输入书里介绍的示例。

本章内容

- 交互式命令 shell
 - 整数算术
 - 浮点数算术
 - 其他数学函数
 - 字符串
 - 字符串拼接
 - 获取帮助
 - 类型转换
 - 变量和值
 - 赋值语句
 - 变量如何引用值
 - 多重赋值
-