

小小的 Python 编程故事

在故事中学编程

不知不觉掌握Python

让你的孩子领先一步

毛雪涛 丁毓峰 编著



入门很快

面向零基础孩子，
立刻就能上手编程

实例很多

全书包括195个实例，
涵盖Python的方方面面

经验很足

毛老师带10岁女儿学Python，
孩子能独立编写扫雷程序

代码很全

书中代码累计达到23 448行，
满足孩子的探索欲望



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

小小的 Python 编程故事

毛雪涛 丁毓峰 编著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是写给孩子看的 Python 编程书，共包括 54 章，分为三个部分。基础部分包含了 Python 编程环境的搭建、第一个 Python 程序的展示、程序的调试方法、异常的处理、Python 的基本数据类型和运算符，还包含了基本程序控制结构、类的使用以及常见模块的使用等内容。

实战部分包含了 Python 循环的应用、冒泡排序和选择排序等基础算法、文件操作、系统信息获取，还包含了图形用户界面编程、正则表达式、多线程程序设计，以及网络编程、数据库编程等内容。

提高部分包含了蒙特卡罗算法、欧几里得算法、递归算法、狄杰特斯拉算法、贪心算法、KNN 算法以及并行计算等内容，这些内容可能会有一些难度，但对于那些喜欢计算科学的读者来说起到了启蒙的作用。

本书适合没有任何编程基础的人学习使用，尤其适合孩子、文科生等非计算机专业的人员使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

小小的 Python 编程故事 / 毛雪涛, 丁毓峰编著. —北京: 电子工业出版社, 2019.1

ISBN 978-7-121-35401-4

I. ①小… II. ①毛… ②丁… 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 253126 号

策划编辑: 张月萍

责任编辑: 牛 勇

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16 印张: 18.75

字数: 420 千字

版 次: 2019 年 1 月第 1 版

印 次: 2019 年 1 月第 1 次印刷

印 数: 4000 册 定价: 69.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。

前 言

如今，计算机的应用已经无处不在，而创造这些应用的人将是未来的主宰，因为计算机应用的基础是程序设计。

牛津大学在 2013 年发布了一项报告，预测未来 20 年里将有一半的工作被机器取代。2014 年，英国把图形化编程纳入了 5 岁以上小朋友的必修课。欧洲其他的一些国家也将编程课纳入了初等义务教育中。编程将和目前的英文一样，成为一种基本能力。

编程可以使孩子拥有更严谨的思维，能让孩子努力理解看不见摸不着的数据结构，能锻炼孩子从具体到概括的抽象能力，也能训练孩子独立钻研问题的能力。

另外，学习编程也能建立孩子对于计算机的正确认识——它不是专门用来玩游戏的游戏机，它是一种生产工具，而且这种生产工具还能够生产高级的产品——程序。

2017 年 7 月，国务院印发了关于《新一代人工智能发展规划》的通知，提出了面向 2030 年我国新一代人工智能发展的指导思想。通知指出“实施全民智能教育项目，在中小学阶段设置人工智能相关课程，逐步推广编程教育，鼓励社会力量参与寓教于乐的编程教学软件、游戏的开发和推广。”人工智能建立在计算机科学的基础之上。

细数当今影响人类的科技巨头，IBM、谷歌、微软、苹果、华为、阿里巴巴、腾讯、百度、小米等，全都和计算机科学相关，它们代表了当今社会发展的潮流。要想跟上潮流的步伐，一个比较一致的意见是——学习编程，越早越好！

目前已经出现的程序设计语言估计有上百种了吧！但是我还是推荐 Python。为什么推荐学习 Python 语言不用多说。正如它的官方网站上的简短描述：Python 强大、快速；兼容性好；可移植；友好、易学；开放。总之，Python 是一门越来越流行的程序设计语言。

介绍 Python 的各种书籍层出不穷,然而,学习一门程序设计语言从来都不是一件容易的事,对于广大读者来说,技术书籍总是缺少那么一点点生趣。本书就是专门为了解决这一问题而创作的。它采用生活化的语言讲述 Python 程序设计的知识,从基础开始一直讲到算法。

兴趣是最好的老师,但枯燥的灌输很快就会将兴趣浇灭。为了维持读者的学习兴趣或者说帮助读者们坚持读完本书,作者绞尽脑汁,将 Python 知识与一个个的小故事联系起来,并最终解决问题。

关于本书

这是一本适合少年儿童学习的程序设计语言入门书籍。它像一本故事书一样,利用短小的篇幅,让读者了解到他们正在学习的 Python 知识如何与现实生活联系起来。

全书没有严肃的教条,没有大篇幅的理论,也没有生涩的专业术语,力求让目标读者阅读起来没有压力。

本书共安排了 54 章。虽然 Python 语言是一门系统化的课程,但是 54 个章节基本上都可单独阅读。如果读者已经学习过部分 Python 的内容,完全可以根据自己的程度,从任何一个章节开始阅读本书,从而节省宝贵的时间。

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 扫码直达本书页面。

- **下载资源:** 本书中部分图片的彩色版本可在[下载资源处](#)下载。
- **提交勘误:** 您对书中内容的修改意见可在[提交勘误处](#)提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- **交流互动:** 在页面下方[读者评论处](#)留下您的疑问或观点, 与我们和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/35401>



目 录

第 1 章 启程：Python 之旅	1	第 5 章 动物园里动物多：数据类型	17
1.1 懒散的壳：Python IDLE Shell.....	1	5.1 狮子、老虎和大象：标准数据类型	17
1.2 小小的成绩单：Python 程序演示	2	5.2 牛牛的牛爷爷：数值类型	19
第 2 章 捉虫子：调试程序	5	第 6 章 神秘的 X 生物：变量的命名和赋值	21
2.1 小小的沉思：bug 和 debug	5	6.1 变量的命名规则	21
2.2 错在哪？打印调试信息	6	6.2 X 馆和神秘生物：变量赋值	23
2.3 使用 IDLE 的 Debugger 工具	9	第 7 章 弟弟的作业题：算术运算和算术赋值	24
第 3 章 一个“假程序”：代码注释	11	7.1 算术运算符	24
3.1 诗词填空：单行注释	11	7.2 二进制的魔术：位运算	26
3.2 牛牛的程序	12	7.3 赋值运算符	27
第 4 章 漂亮的展示牌：输入、输出和处理	14	第 8 章 真真假假：比较运算和逻辑运算	29
4.1 漂亮的展示牌：输出	14	8.1 真和假：逻辑	29
4.2 接受你的请求：输入	15		
4.3 程序存在的意义：处理	16		

8.2	能够组成三角形吗	30	第 14 章	老狼老狼几点了：循环结构	63
8.3	逻辑运算	31	14.1	没完没了：while 语句	63
第 9 章	有身份的 MVP：成员、身份和优先级	34	14.2	老狼该休息了：for 语句	65
9.1	篮球梦的开始：成员运算符	34	14.3	小花的脾气：break、continue 和 pass	66
9.2	谁是 MVP：身份运算符	35	第 15 章	同学通讯录：字典	68
9.3	运算符的优先级	37	15.1	制作通讯录：字典和键值对	68
第 10 章	我的世界：字符编码和字符串	39	15.2	通讯录的作用：访问字典元素 ..	70
10.1	从数值到符号：编码	39	15.3	记录了多少同学	72
10.2	小小的 1000 只羊：字符串	41	15.4	一个变两个：字典的复制	72
10.3	没烦恼的诗人：转义字符	43	第 16 章	飞越地平线：基本队列	75
10.4	字符串函数	44	16.1	乐园永恒的主题：创建队列	75
第 11 章	王者的药：条件控制	47	16.2	FIFO：队列的基本性质	76
11.1	健康系统：if 语句	47	第 17 章	小小建筑师：函数与参数传递	79
11.2	健康系统加强版：if 语句的嵌套	48	17.1	墙壁和地板：函数的定义和调用	79
第 12 章	阿波菲斯的剑鞘：列表	51	17.2	参数传递	81
12.1	物品列表	51	第 18 章	幸运大转盘：随机数发生器	85
12.2	了解自己的物品：列表的函数 ..	53	18.1	谁是幸运顾客：choice()	85
12.3	新的物品：列表的操作	54	18.2	免费的蛋糕：sample()	86
第 13 章	小小蛋糕店：元组和区间	57	18.3	洗牌：shuffle()	87
13.1	第一个菜单：建立元组	57	第 19 章	爷爷的怪蛋糕：类和对象	89
13.2	请问第 4 种是什么蛋糕	58	19.1	蛋糕模板：类的定义	89
13.3	各式各样的菜单	60	19.2	制造蛋糕：创建对象	91
13.4	等差数列的创造者：range()	61	19.3	如何制造蛋糕？构造方法	91

第 20 章 蛋糕家族：类的继承	94	25.2 改良的冒泡排序	128
20.1 古怪蛋糕也是蛋糕	94	第 26 章 销量排行榜：选择排序	130
20.2 这是遗传：继承的特性	96	26.1 销量冠军：求最大项	130
第 21 章 特工联盟：模块	99	26.2 选择排序	132
21.1 联盟条约：什么是模块	99	26.3 选择排序和冒泡排序哪个 更快	133
21.2 联盟宣言：模块内的变量 和程序	101	第 27 章 程序员的暴力：穷举法	135
21.3 模块的其他特征	103	27.1 百钱买百鸡	135
第 22 章 妈妈生日快乐：日期和时间 ...	105	27.2 破解通关密码	137
22.1 5 月的日历	105	第 28 章 开心森林：最短路径问题	139
22.2 母亲节是哪一天	108	28.1 乘车路线图	139
22.3 顾客驾到：记录当前时间	109	28.2 图的代码实现	140
22.4 时间元组和时间戳	110	28.3 广度优先搜索	141
第 23 章 警报，警报：异常处理	112	第 29 章 小小日记本：文件基本操作 ...	144
23.1 小小的错误：语法错误	112	29.1 创建日记本	144
23.2 非正常行为：异常	113	29.2 写日记：写入文件	146
23.3 异常捕手：异常处理	115	29.3 翻看旧日记：读取文件	147
23.4 个性化的异常处理	116	29.4 读取指定日记	148
23.5 小小的恶作剧：抛出异常	120	第 30 章 识得庐山真面目：与系统 打交道	150
第 24 章 鸡兔同笼：循环的应用	122	30.1 系统信息：OS 常用方法	150
24.1 雉兔各几何	122	30.2 文件系统信息	151
24.2 更多的笼子	123	30.3 调用系统命令	152
24.3 “鸡兔同笼”游戏	124	第 31 章 高级身份牌：GUI 编程初步 ...	154
第 25 章 步数排行榜：冒泡排序	125	31.1 提拉米苏的身份牌	154
25.1 前后交换：冒泡排序的基本 操作	125		

31.2	舒芙蕾的身份牌: Text.....	156	第 38 章	散文中的动词: 正则表达式...187
31.3	更多的小部件	158	38.1	找到杨柳、燕子和桃花
第 32 章	一触即发: 事件编程	159	38.2	找到“动词”: 正则表达式的模式
32.1	蛋糕列表: Listbox.....	159		189
32.2	程序的感知: 事件响应	160	第 39 章	小小的爬虫: 正则表达式的应用
第 33 章	印象派: Canvas 绘图	163		192
33.1	一条直线: Canvas 初探	163	39.1	切割网页: 为匹配做准备
33.2	标注坐标点: 绘制文字	165	39.2	找出文字中的链接: 正则匹配...193
33.3	方块和椭圆	166	第 40 章	大蛇卡丁车: 多线程
第 34 章	三国名人录: 绘制图像.....	168		195
34.1	神机妙算诸葛亮	168	40.1	赛况直播: 了解多线程
34.2	三国名人录	169	40.2	小小的秘密武器: 线程锁
第 35 章	生命在于运动: Canvas 动画.....	172		198
35.1	Just move.....	172	第 41 章	您有一个包裹: JSON 处理...201
35.2	上下左右: 控制动画	174	41.1	小小的礼物: JSON 编码.....
第 36 章	超强背景音: 播放声音.....	176	41.2	吉森的回信: 解析 JSON.....
36.1	播放 wav 文件	176		203
36.2	pip 和 pygame: 安装外部模块...178		第 42 章	来自蛋糕店的问候: Web 服务器与 CGI 程序
36.3	蛋糕店的主题曲: 播放 mp3 ...179			205
第 37 章	猜数游戏: GUI 应用	181	42.1	网站的基础: Web 服务器
37.1	音乐和音效	181	42.2	蛋糕店的问候: 第一个 CGI 程序
37.2	游戏的交互: 事件处理	183		207
37.3	游戏界面	185	第 43 章	为顾客服务: GET 和 POST
				210
			43.1	填写蛋糕的名字: 客户表单 ...210
			43.2	客户表单处理程序
			43.3	隐藏信息的传递方式: POST...214

第 44 章 小 i 是个机器人： socket 编程	215	第 50 章 汉诺塔问题：递归的应用	253
44.1 给小 i 发送消息：客户端	215	50.1 简化的汉诺塔：三阶刚刚好 ...	253
44.2 小 i 的回答：服务器	217	50.2 汉诺塔问题的步骤数	256
第 45 章 小小伊妹儿：邮件发送程序...	220	第 51 章 别针换摩托：迪杰特斯拉 算法	259
45.1 “吉森，你好!”：文字邮件 ...	220	51.1 交换大会：有向加权图	259
45.2 小小的近照：发送附件	223	51.2 小 D 的办法：最优路径	262
第 46 章 信息大爆炸：初识数据库 ...	226	51.3 “换”梦成真：最优路径算法...	264
46.1 什么是数据库	226	第 52 章 验证哥德巴赫猜想： 并行计算	267
46.2 挠痒痒：连接 MySQL 数据库 ...	229	52.1 什么是哥德巴赫猜想	267
第 47 章 聪明的 BOSS：数据库应用 ..	233	52.2 充分利用 CPU：并行计算	269
47.1 First of All：创建数据库	233	第 53 章 小小旅行家：贪心算法	273
47.2 Drop：删除数据库	235	53.1 旅行商问题	273
47.3 员工与蛋糕：创建数据表	236	53.2 环球旅行：贪心算法	275
47.4 添加第一个员工	240	第 54 章 电影分类和猜蛋糕： KNN 算法	279
第 48 章 大厨的“派”：随机数的应用	246	54.1 你会看电影吗？特征抽取	279
48.1 神秘的厨师：蒙特卡罗	246	54.2 和哪部电影最像？分类	280
48.2 派和 π ：蒙特卡罗法应用	247	54.3 做多少蛋糕才合适？回归	282
第 49 章 欧几里得算法：辗转相除 ...	249	附录 A 如何安装 Python	285
49.1 操场划分：最大公约数	249		
49.2 最小公倍数	251		

第 1 章

启程：Python 之旅

你好，我是一个和你一样的中学生，我的爱好是看书、画画、制作蛋糕和编程。我喜欢用程序来解决学校和生活中的事情。我有许多好朋友，我经常和他们一起讨论功课和蛋糕。我的梦想是开一家小小的蛋糕店。我叫小小。

1.1 懒散的壳：Python IDLE Shell

小小的 Windows 系统下已经安装了非常多的程序，Python 是他最喜欢的一个。Python 是干什么用的呢？先打开看看再说吧！

单击“所有程序”，可以在菜单中找到 Python 3.6 这个文件夹图标，单击展开 Python 3.6 文件夹，可以发现下面有几个 Python 条目，如图 1.1 所示。

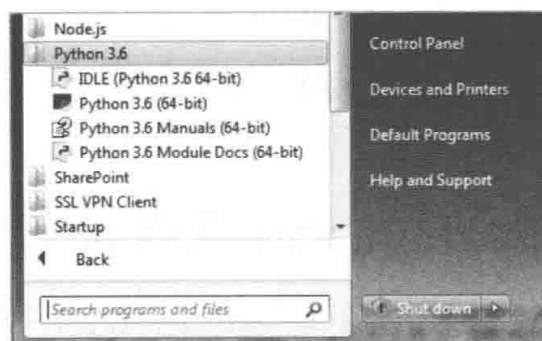


图 1.1 启动 Python IDLE

其中，IDLE(Python 3.6 64-bit)是 Python 的图形界面开发环境；Python 3.6(64-bit)是字符界面开发环境；Python 3.6 Manuals(64-bit)和 Python 3.6 Module Docs(64-bit)分别是 Python 的用户文档和模块文档。

首先，我们打开 IDLE 这个被称为“空闲”的程序。IDLE 默认启动界面为 Python IDLE Shell，如图 1.2 所示。Shell 是外壳的意思，这很形象地说明了这个程序是用来包裹 Python 内含的复杂机制，而给用户提供一个光鲜的图形界面的。用户在 Shell 中可以与 Python 内核进行交互。

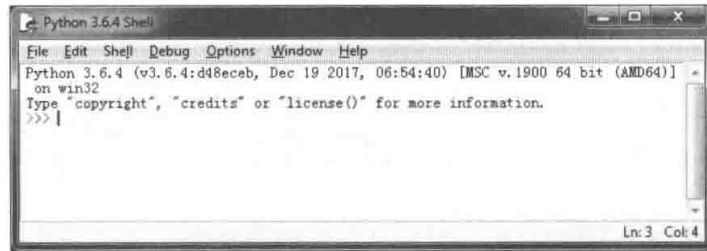


图 1.2 Python IDLE Shell

可以看到，界面中“>>>”符号后面有一个闪烁的光标。“>>>”是 Python 的提示符，光标指示程序等待键入信息。在“>>>”符号后面输入下面代码：

```
print ("我的名字")
```

输入完成后，按回车键，就会运行这行代码，运行结果如图 1.3 所示。

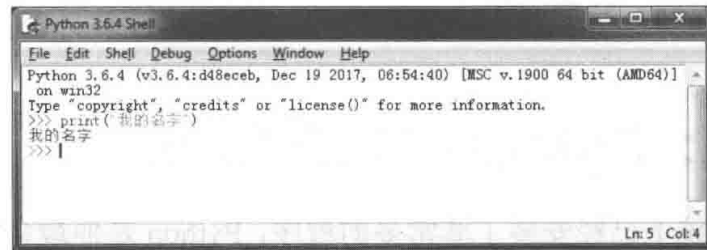


图 1.3 在 Python IDLE Shell 环境下运行程序

你键入的这行文本是一个简单的 Python 语句，它会输出（print）你输入的消息。print 是命令的名称，这是一个用于输出的命令。后面括号中的部分为要输出的内容，内容用引号括起来，是为了表示它只是一串字符。你可以在引号中放任何内容，然后把它输出到屏幕上。

今后，懒散的壳（IDLE Shell）会陪伴我们很长时间。

1.2 小小的成绩单：Python 程序演示

“Python 好简单呀！”正说着，你可能已经输出了各种各样的字符。不过，Python 可不是你想象中的那么简单，据说 Python 是人工智能的首选语言呢！

Python 这么厉害, 小小有点不相信, 他打算让程序做一件事情: 输出自己的成绩单。首先输出学生名字, 然后再输出语文、数学和英语三门主课的成绩, 并且计算总分, 最后输出三门课的最高分、最低分和平均分。Python 会怎么做呢? Python IDLE 有一个妙招!

打开 Python IDLE Shell, 选择 File 菜单下的 New File 菜单项, 如图 1.4 所示。

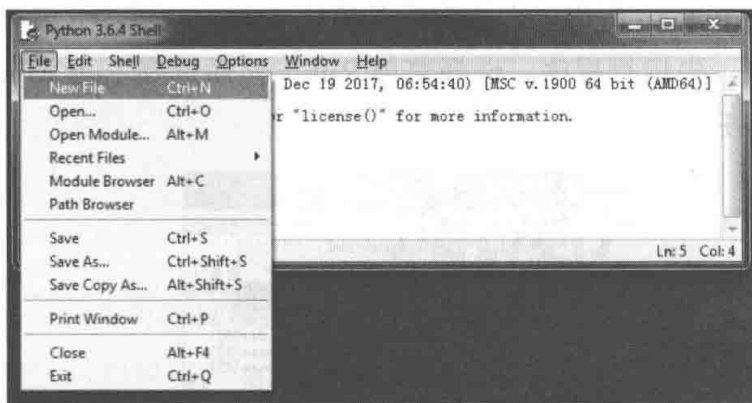


图 1.4 File 菜单

这时会弹出一个新的空白窗口, 如图 1.5 所示。注意, 这个窗口中只有一个光标, 没有“>>>”提示符, 窗口标题栏也没有 Shell 字样。这是一个编辑窗口。

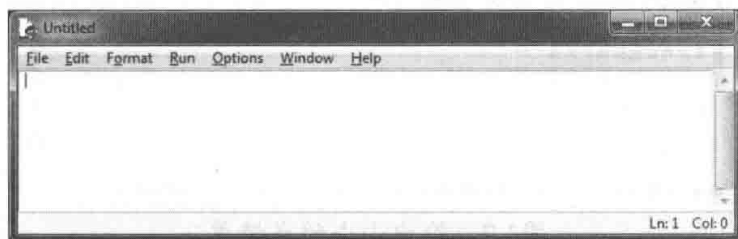


图 1.5 程序编辑器

选择 File 菜单下的 Save 菜单项, 将这个编辑窗口保存为一个 Python 程序文件, 并保存在 C:\Workspace\Chapter1 目录下, 文件名为 hello.py。然后在窗口中输入以下代码:

```
#在屏幕上输出姓名和成绩
score = [80,70,90]           #定义变量 score
min_grade = None           #定义变量 min_grade 最低分
max_grade = None           #定义变量 max_grade 最高分
sum_grade = 0.0            #定义变量 sum_grade 总分
print ("我的名字是: 小小。") #输出相关信息
print ("本次考试我的语文成绩: %d"%score[0])
print ("      数学成绩: %d"%score[1])
print ("      英语成绩: %d"%score[2])
sum_grade = score[0]+ score[1]+score[2]    #求总分
ave_grade = sum_grade / 3.    #求平均分
min_grade=score.index(min(score))          #最低分
```

```
max_grade=score.index(max(score))      #最高分
print ("总 分:", sum_grade)             #输出总分
print ("最高分:", score[max_grade])     #输出最高分
print ("最低分:", score[min_grade])     #输出最低分
print ("平均分:", ave_grade)           #输出平均分
```

这就是一个完整的 Python 程序，它的作用是输出小小的成绩单。具体每条程序语句的含义后面会详细介绍。现在，先运行程序。选择菜单 Run，然后选择 Run Module 菜单项，如图 1.6 所示。也可以按快捷键 F5。

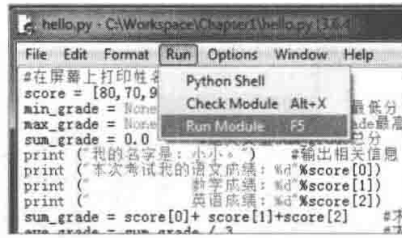


图 1.6 运行Python程序

运行后结果显示在 Shell 窗口中，如图 1.7 所示。

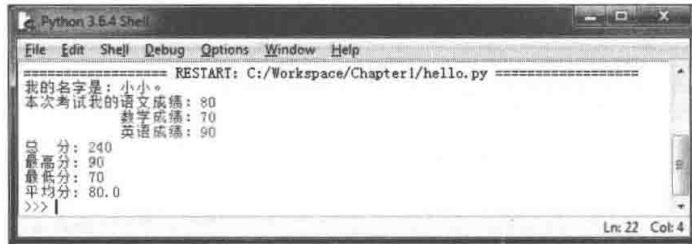


图 1.7 输出小小的成绩单

小小同学这次考试三门课的成绩以及成绩统计信息都清楚地显示在窗口中。程序执行完毕后，熟悉的光标又开始在提示符后面闪烁了。这时，你可以继续在光标处输入 Python 命令并执行，如图 1.8 所示。



图 1.8 继续执行命令

知道了这些，我们就可以开始伟大的 Python 之旅了。

第 2 章

捉虫子：调试程序

阅读了前面的内容后，小小同学觉得自己可以写 Python 程序了，他决定先写一个“hello, 提拉米苏!”程序。代码只有一行：

```
print("hello, 提拉米苏!")
```

可是运行后却出，错，了！

2.1 小小的沉思：bug 和 debug

小小的“hello, 提拉米苏!”程序在 IDLE 中运行时出错了，弹出了一个警告框，如图 2.1 所示。

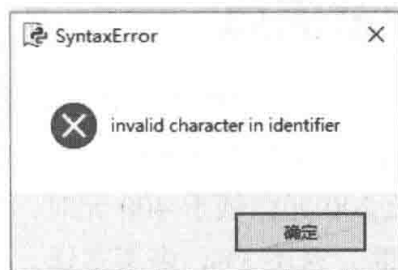


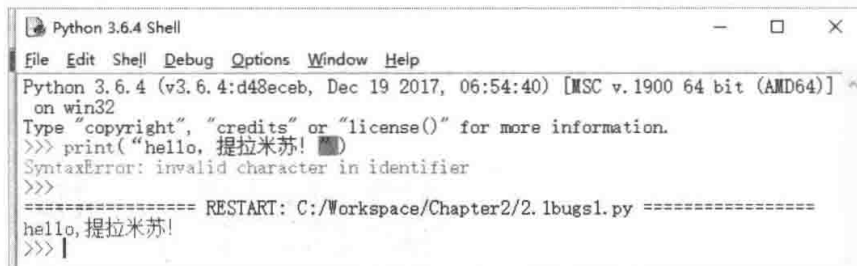
图2.1 警告框

这个警告的意思是有语法错误，就是说，你写的代码，Python 的 IDLE 看不懂！IDLE 在运行 Python 程序时会先检查一遍代码，发现其中存在语法错误会给出错误警告。小小仔细检查了

一遍代码，发现 `print()` 中的引号似乎写得不对。于是他把代码修改了一下：

```
# 引号错误: print("hello, 提拉米苏!")  
print("hello, 提拉米苏!")
```

将引号改为英文的引号，再次运行就成功了，如图 2.2 所示。



The screenshot shows a Python 3.6.4 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area shows the following content:
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello, 提拉米苏!")
SyntaxError: invalid character in identifier
>>>
===== RESTART: C:/Workspace/Chapter2/2.1bugs1.py =====
hello, 提拉米苏!
>>> |

图2.2 “hello, 提拉米苏!” 程序正确运行了

虽然程序正确运行了，可是小小却陷入了沉思：“这个程序只有一行代码，那如果编写更复杂的程序，如何才能保证代码不出错呢？”答案是：“无法保证！”有人统计过，一次就编写成功没有错误的程序大约只有 1%。代码越长，出错的可能性就越大。这真是太糟糕了！不过反过来想，我们心理也平衡了，毕竟自己写的程序出错了也并不是什么稀奇的事。

我们虽然无法保证程序不出错，但是仔细检查代码，还是可以找出其中的错误的，然后改正它们。

在实际编程中，程序中出现的错误可没有这里的错误这么简单，成千上万行的代码中会出现千奇百怪的错误，让人防不胜防。这些错误就像一群藏在程序中的虫子，在英文中被称作 `bug`。有的 `bug` 很简单，看看出错信息就可以知道问题出在哪，有的 `bug` 很复杂，需要知道在出错时，哪些变量的值是正确的，哪些变量的值是错误的。因此，需要通过一定的手段来修复 `bug`，这个过程叫作调（`tiao`）试，在英文中叫作 `debug`。

2.2 错在哪？打印调试信息

小小家门口有家蛋糕店，今天打出了一个促销广告：购买蛋糕总价小于或等于 100 元时，返还 10% 现金；购买总价高于 100 元但低于 200 元时，低于 100 元的部分返还 10%，高于 100 元的部分返还 7.5%；购买总价高于 200 元但低于 400 元时，高于 200 元的部分返还 5%；购买总价超过 400 元时，超过 400 元的部分返还 3%，多买多送。小小一看，这个规则太复杂了，到底能有多少优惠，只能写个程序来算一算了。

程序代码如下：

```
#程序 2.2 printLog.py
```

```

while 1:
    i = int(input('输入购买总价:'))
    if i==0:
        break
    arr = [400,200,100,0]
    rat = [0.03,0.05,0.075,0.1]
    r = 0
    for idx in range(0,3):
        if i>arr[idx]:
            r+=(i-arr[idx])*rat[idx]
            i=arr[idx]
    print('总优惠金额: ',r)

```

选择菜单命令 Run→Run Module 运行程序，输入购买的总价，得到应返现的金额，如图 2.3 所示。



```

===== RESTART: E:\蛋糕店中的Python\Chapter2\2.2 printLog.py =====
===
输入购买总价:50
总优惠金额: 0
输入购买总价:110
总优惠金额: 0.75
输入购买总价:230
总优惠金额: 9.0
输入购买总价:

```

图2.3 程序运行的结果

对这个结果满意吗？购买总价为 50 元时，总优惠金额为 0！显然这个结果是错误的。必须通过调试来找出程序中的错误。

如何调试程序呢？一般可以分三个步骤：

第一步，让程序分段运行。也就是说，一段一段地来排除错误，逐渐缩小抓虫子的范围。这就需要使用一些办法，把程序分成一段一段的，运行一段后就停下来。

第二步，检查程序运行到分段处时，变量的值是否正确。

第三步，确定出错的根源在哪里，并进行修正，然后再回到第一步进行新一轮调试。

首先，小小在程序中添加了一些 `print()` 语句，通过输出一些有用的中间信息来判断错误在哪，修改后的代码如下：

```

#2.2printLog1.py
while 1:
    i = int(input('输入购买总价:'))
    if i==0:
        break
    arr = [400,200,100,0]

```