

简明易懂、有趣味性的

Python

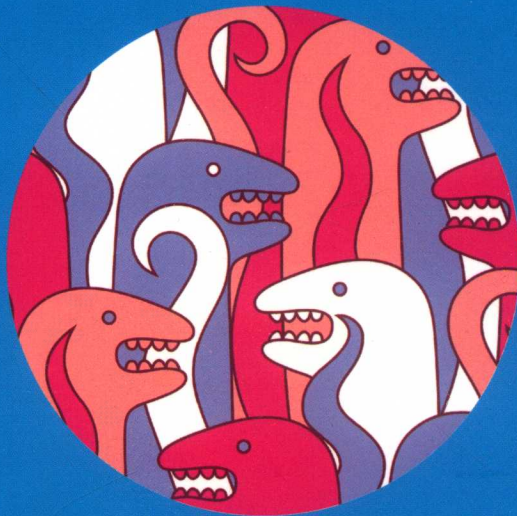
游戏开发教程

PYTHON GAME DESIGN CASE
IN ACTION

Python

游戏设计案例实战

夏敏捷 尚展垒 著



以**游戏开发案例**为导向，
内容涵盖**Python 游戏开发**的各个方面

讲解**19**个不同游戏的**开发思路及具体实现**，并给出**完整代码**，
还介绍了**Pygame**游戏开发模块



中国工信出版集团



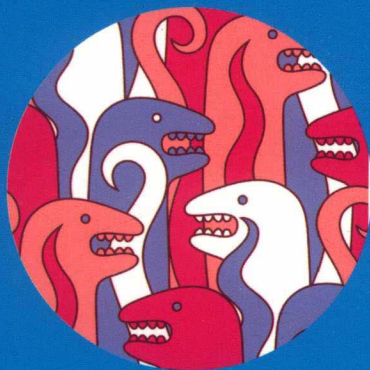
人民邮电出版社
POSTS & TELECOM PRESS

简明易懂、有趣味性的
Python
游戏开发教程

PYTHON GAME DESIGN CASE
IN ACTION

Python

游戏设计案例实战



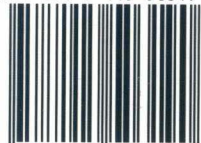
Python 语言的简洁、易读以及可扩展性，使之成为目前十分受欢迎的程序设计语言之一。本书通过编写生动有趣的游戏来介绍 Python 编程，通过分析案例来讲解编程的原理。本书直接展示了游戏案例的源代码，并对源代码进行了详细的剖析。同时，本书为每个游戏案例提供了详细的设计思路、关键技术分析以及具体的解决方案，让读者能真正地了解 Python 程序设计的秘密。最后，本书还介绍了专门用于游戏开发的跨平台的 Python 模块——Pygame。

免/费/提/供
PPT等教学相关资料

 人邮教育
www.rjiaoyu.com

教材服务热线：010-81055256
反馈 / 投稿 / 推荐信箱：315@ptpress.com.cn
人民邮电出版社教育服务与资源下载社区：www.rjiaoyu.com

ISBN 978-7-115-50319-0



9 787115 503190 >

ISBN 978-7-115-50319-0

定价：59.80元

封面设计：董志桢

通俗易懂、有趣味性的

Python

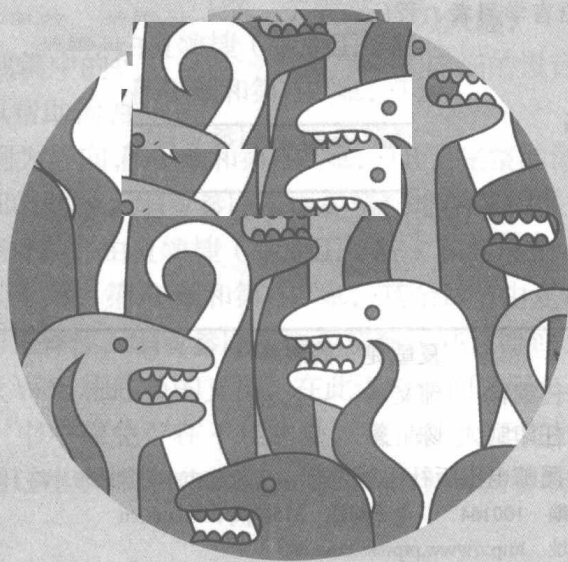
游戏开发教程

PYTHON GAME DESIGN CASE
IN ACTION

Python

游戏设计案例实战

夏敏捷 尚展垒 著



人民邮电出版社

北京

此为试看, 需要完整PDF请访问: www.ezrbook.com

图书在版编目 (CIP) 数据

Python 游戏设计案例实战 / 夏敏捷, 尚展垒著. —
北京: 人民邮电出版社, 2019. 11 (2020. 3重印)
ISBN 978-7-115-50319-0

I. ①P… II. ①夏… ②尚… III. ①软件工具—游戏
程序—程序设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2018)第275759号

内 容 提 要

本书以 Python 3.5 为编程环境, 从基本的程序设计思想入手, 逐步开展 Python 语言教学, 是一本面向广大编程学习者的程序设计类教材。全书分为基础篇、实战篇和提高篇。基础篇包括第 1 章~第 8 章, 主要讲解 Python 的基础语法知识、控制语句、函数、文件、面向对象编程基础、Tkinter 图形界面设计、数据库应用、网络编程和多线程等内容, 并以小游戏案例作为各章的阶段任务; 实战篇包括第 9 章~第 18 章, 综合应用前面各章中介绍的技术, 重现几个经典游戏的开发过程; 提高篇包括第 19 章~第 20 章, 主要介绍跨平台的 Python 模块——Pygame, 并应用 Pygame 模块重现了几个经典游戏的开发过程。

本书以游戏开发案例为导向, 内容通俗易懂, 图文并茂, 适合作为高等院校计算机相关专业的教材, 还可作为 Python 语言学习者、程序设计人员和游戏编程爱好者的参考书。

-
- ◆ 著 夏敏捷 尚展垒
责任编辑 邹文波
责任印制 陈 犇
 - ◆ 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>
涿州市京南印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 21 2019 年 11 月第 1 版
字数: 500 千字 2020 年 3 月河北第 2 次印刷
-

定价: 59.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

前言

Python 是新兴程序设计语言，是一种解释型、面向对象、动态数据类型的高级程序设计语言。自 20 世纪 90 年代初 Python 语言诞生至今，它逐渐被广泛应用于处理系统管理任务和科学计算，成为最受欢迎的程序设计语言之一。由于 Python 语言的简洁、易读以及可扩展性，十分适合用于游戏开发，因此，笔者编写了本书。

本书内容如下：基础篇包括第 1 章~第 8 章，主要讲解 Python 的基础知识和面向对象编程基础、Tkinter 图形界面设计、网络编程和多线程、Python 数据库应用、图像处理等知识，每章最后都有应用本章知识点的游戏案例；实战篇包括第 9 章~第 18 章，综合应用前 8 章介绍的技术，重现如连连看、推箱子、中国象棋、两人麻将、扫雷游戏、华容道等经典游戏的开发过程；提高篇包括第 19 章和第 20 章，讲解了基于 Pygame 的游戏设计基本知识，并应用 Pygame 开发贪吃蛇和飞机大战游戏案例，以及一款有趣的数学休闲益智游戏——2048。

本书主要特点如下：

(1) Python 程序设计涉及的范围非常广泛，本书内容编排不求全、求深，而是考虑零基础读者的接受能力，语言语法介绍以应用为原则，尽量选取 Python 中必备、实用的知识进行讲解，着力强化程序思维能力的培养；

(2) 游戏案例选取贴近生活，有助于提高读者的学习兴趣；

(3) 编者开发实战篇中的每款游戏案例均提供了详细的设计思路、关键技术分析以及具体的解决步骤，以帮助读者提高学习效率。

本书以游戏开发案例为导向，编者不仅在书中列出了完整的源代码，同时还对所有的源代码进行了非常详细的解释，力求引导读者更轻松的学习相关知识。

本书受到中原工学院资助，由夏敏捷（中原工学院）和尚展垒（郑州轻工业大学）编写。其中，尚展垒编写第 4 章、第 9 章和第 19 章，其余章节由夏敏捷编写。在本书的编写过程中，为确保内容的正确性，编者参阅了很多资料，并且得到了中原工学院计算机学院郑秋生教授和众多资深 Web 程序员的支持，在此谨向他们表示衷心的感谢。

由于编者水平有限，书中难免存在不足之处，敬请广大读者批评指正。若读者有问题需要与编者沟通，可通过电子邮件（xmj@zut.edu.cn）与编者联系。

夏敏捷

2019 年 2 月

目录

基础篇

第1章 Python 基础知识 2

1.1 Python 语言概述 2

1.1.1 Python 语言简介 2

1.1.2 安装 Python 3

1.1.3 Python 开发环境 IDLE 的启动 5

1.1.4 利用 IDLE 创建 Python 程序 5

1.1.5 在 IDLE 中运行和调试 Python 程序 6

1.1.6 Python 基本输入 8

1.1.7 Python 基本输出 9

1.1.8 Python 代码规范 9

1.1.9 Python 帮助 11

1.2 Python 语法基础 12

1.2.1 Python 数据类型 12

1.2.2 序列数据结构 13

1.2.3 Python 控制语句 21

1.2.4 Python 函数与模块 26

1.3 Python 文件的使用 30

1.3.1 打开（建立）文件 30

1.3.2 读取文本文件 32

1.3.3 写文本文件 33

1.3.4 文件内移动 35

1.3.5 文件的关闭 36

1.3.6 文件应用案例——游戏地图存储 36

1.4 Python 的第三方库 38

思考题 39

第2章 序列应用——猜单词游戏 40

2.1 游戏介绍 40

2.2 程序设计的思路 40

2.3 random 模块 41

2.4 程序设计的步骤 43

思考题 45

第3章 面向对象设计应用——发牌游戏 46

3.1 游戏介绍 46

3.2 Python 面向对象设计 46

3.2.1 定义和使用类 47

3.2.2 构造函数 `__init__` 48

3.2.3 析构函数 48

3.2.4 实例属性和类属性 49

3.2.5 私有成员与公有成员 50

3.2.6 方法 51

3.2.7 类的继承 52

3.2.8 多态 54

3.3 程序设计的步骤 56

3.3.1 设计类 56

3.3.2 主程序 59

思考题 59

第4章 Python 图形界面设计——猜数字游戏	60	6.4.2 读取试题信息	108
4.1 游戏介绍	60	6.4.3 界面和逻辑设计	109
4.2 Python 图形界面设计	60	思考题	110
4.2.1 创建 Windows 窗口	61	第7章 网络编程和多线程——网络五子棋游戏	111
4.2.2 几何布局管理器	61	7.1 游戏介绍	111
4.2.3 Tkinter 组件	65	7.2 网络编程基础	112
4.2.4 Tkinter 字体	76	7.2.1 互联网 TCP/IP	112
4.2.5 Python 事件处理	78	7.2.2 IP	113
4.3 程序设计的步骤	83	7.2.3 TCP 和 UDP	113
思考题	85	7.2.4 端口	113
第5章 Tkinter 图形绘制——图形版发牌程序	86	7.2.5 socket	114
5.1 程序功能介绍	86	7.3 TCP 编程	117
5.2 程序设计的思路	86	7.3.1 TCP 客户端编程	117
5.3 Canvas 图形绘制技术	87	7.3.2 TCP 服务器端编程	119
5.3.1 Canvas 画布组件	87	7.4 UDP 编程	122
5.3.2 Canvas 上的图形对象	88	7.5 多线程编程	124
5.4 程序设计的步骤	97	7.5.1 进程和线程	124
思考题	99	7.5.2 创建线程	125
第6章 数据库应用——智力问答游戏	100	7.5.3 线程同步	129
6.1 游戏介绍	100	7.5.4 定时器	131
6.2 程序设计的思路	100	7.6 程序设计的步骤	131
6.3 数据库访问技术	101	7.6.1 数据通信协议设计和判断输赢的算法	131
6.3.1 访问数据库的步骤	101	7.6.2 服务器端程序设计	135
6.3.2 创建数据库和表	102	7.6.3 客户端程序设计	140
6.3.3 数据库的插入、更新和删除操作	103	思考题	144
6.3.4 数据库表的查询操作	104	第8章 Python 图像处理——人物拼图游戏	145
6.3.5 数据库使用实例——学生通信录	104	8.1 游戏介绍	145
6.4 程序设计的步骤	107	8.2 程序设计的思路	145
6.4.1 生成试题库	107	8.3 Python 图像处理	146
		8.3.1 Python 图像处理类库	146
		8.3.2 复制和粘贴图像区域	148
		8.3.3 调整尺寸和旋转	149

8.3.4 转换成灰度图像	149	12.3.3 对保存麻将牌的列表排序	203
8.3.5 对像素进行操作	150	12.4 程序设计的步骤	204
8.4 程序设计的步骤	150	12.4.1 麻将牌类设计	204
8.4.1 Python 处理图片切割	150	12.4.2 设计游戏主程序	206
8.4.2 游戏逻辑的实现	152	第 13 章 贪吃蛇游戏	217
思考题	155	13.1 游戏介绍	217
实战篇		13.2 程序设计的思路	217
第 9 章 人机对战井字棋游戏	157	13.3 程序设计的步骤	218
9.1 游戏介绍	157	13.3.1 Grid 类(场地类)	218
9.2 程序设计的思路	158	13.3.2 Food 类(豆类)	219
9.3 程序设计的步骤	158	13.3.3 Snake 类(蛇类)	219
9.4 窗体版游戏	162	13.3.4 SnakeGame(游戏逻辑)类	220
第 10 章 连连看游戏	168	第 14 章 人机对战黑白棋游戏	223
10.1 游戏介绍	168	14.1 游戏介绍	223
10.2 程序设计的思路	169	14.2 程序设计的思路	224
10.3 程序设计的步骤	179	14.3 程序设计的步骤	224
第 11 章 推箱子游戏	184	第 15 章 扫雷游戏	232
11.1 游戏介绍	184	15.1 游戏介绍	232
11.2 程序设计的思路	185	15.2 程序设计的思路	233
11.3 关键技术	187	15.3 关键技术	233
11.4 程序设计的步骤	188	15.4 程序设计的步骤	235
第 12 章 两人麻将游戏	193	第 16 章 中国象棋游戏	241
12.1 游戏介绍	193	16.1 游戏介绍	241
12.2 程序设计的思路	194	16.2 关键技术	242
12.2.1 素材图片	194	16.3 程序设计的思路	244
12.2.2 游戏逻辑的实现	195	16.4 程序设计的步骤	247
12.2.3 碰牌、吃牌判断	195	第 17 章 21 点扑克牌游戏	257
12.2.4 和牌算法	196	17.1 游戏介绍	257
12.2.5 实现电脑机器人智能出牌	200	17.2 关键技术	257
12.3 关键技术	202	17.3 程序设计的步骤	259
12.3.1 声音播放	202		
12.3.2 返回对应位置的组件	202		

第 18 章 华容道游戏	265	19.2.4 Pygame 的字体使用	288
18.1 游戏介绍	265	19.2.5 Pygame 的声音播放	289
18.2 程序设计的思路	265	19.2.6 Pygame 的精灵使用	291
18.3 程序设计的步骤	266	19.3 基于 Pygame 设计贪吃蛇游戏	296
		19.4 基于 Pygame 设计飞机大战 游戏	303
		19.4.1 游戏角色	303
		19.4.2 游戏界面显示	306
		19.4.3 游戏逻辑的实现	308
		19.5 基于 Pygame 设计黑白棋游戏	313
		第 20 章 2048 游戏	320
		20.1 游戏介绍	320
		20.2 程序设计的思路	320
		20.3 程序设计的步骤	322
		参考文献	328
提高篇			
第 19 章 基于 Pygame 游戏 设计	275		
19.1 Pygame 基础知识	275		
19.1.1 安装 Pygame 库	275		
19.1.2 Pygame 的模块	275		
19.2 Pygame 的使用	278		
19.2.1 Pygame 开发游戏的主要 流程	278		
19.2.2 Pygame 的图像图形绘制	280		
19.2.3 Pygame 的键盘和鼠标事件 的处理	284		

Python 基础知识

Python 是一门跨平台、开源、免费的解释型高级动态编程语言。Python 作为动态语言十分适合初学编程者。Python 可以让初学者把精力集中在编程对象和思维方法上，而不用过多地关注语法、类型等外在因素。Python 易于学习，拥有大量的库，可以高效地开发各种应用程序。本章介绍 Python 语言的特点、Python 的安装、Python 开发环境 IDLE 的使用，以及进行 Python 程序设计需要学习的基础内容。

1.1 Python 语言概述

1.1.1 Python 语言简介

Python 于 1989 年年底由吉多范罗·苏姆 (Guido van Rossum) 开发，被广泛应用于处理系统管理任务和科学计算，是最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为 2010 年度语言。自 2004 年以后，Python 的使用率呈线性增长，在 TIOBE 公布的 2017 年编程语言指数排行榜中，Python 的排名处于第四位（前 3 位是 Java、C 和 C++）。2018 年 8 月，根据 IEEE Spectrum 发布的研究报告显示，Python 在 2017 年已经成为世界上最受欢迎的语言。

Python 支持命令式编程、函数式编程，完全支持面向对象程序设计，语法简洁清晰，并且拥有大量的扩展库，几乎支持所有领域的编程开发。

Python 为我们提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容。使用 Python 进行开发，许多功能不必从零编写，直接使用现成的即可。除了内置的库外，Python 还有大量的第三方库，也就是其他开发者开发的，可供开发者直接使用。Python 就像“胶水”一样，可以把多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。因此，Python 程序看上去总是简单易懂，初学者学 Python，不但入门容易，而且将来深入学习下去，也可以编写更加复杂的程序。

Python 同时也支持伪编译将 Python 源程序转换为字节码来优化程序和提高运行速度，还可以在没安装 Python 解释器和相关依赖包的平台上运行。

Python 语言的应用领域主要包括如下几种。

(1) Web 开发。Python 语言支持网站开发，比较流行的开发框架有 web2py、Django 等。许多大型网站就是用 Python 开发的，如 YouTube、Instagram 等。很多大公司，如

Google、Yahoo 等，甚至 NASA（美国航空航天局）都大量地使用 Python 进行开发。

（2）网络编程。Python 语言提供了 socket 模块，对 socket 接口进行了两次封装，支持对 socket 接口的访问；还提供了 urllib、httplib、scrapy 等大量模块，用于对网页内容进行读取和处理，并结合多线程编程以及其他相关模块可以快速开发网页爬虫等应用程序。

（3）科学计算与数据可视化。Python 中用于科学计算与数据可视化的模块很多，如 NumPy、SciPy、Matplotlib、Traits、TVTK、Mayavi、VPython、OpenCV 等，涉及的应用领域包括数值计算、符号计算、二维图表、三维数据可视化、三维动画演示、图像处理以及界面设计等。

（4）数据库应用。Python 数据库模块有很多，例如，可以通过内置的 sqlite3 模块访问 SQLite 数据库，使用 pywin32 模块访问 Access 数据库，使用 pymysql 模块访问 MySQL 数据库，使用 pywin32 和 pymssql 模块访问 SQL Sever 数据库。

（5）多媒体开发。PyMedia 模块可以对 WAV、MP3、AVI 等多媒体格式文件进行编码、解码和播放；PyOpenGL 模块封装了 OpenGL 应用程序编程接口，通过该模块可在 Python 程序中集成二维或三维图形；Python 图形库（Python Imaging Library, PIL）为 Python 提供了强大的图像处理功能，并提供广泛的图像文件格式支持。

（6）电子游戏应用。Pygame 就是用来开发电子游戏软件的 Python 模块。使用 Pygame 模块，可以在 Python 程序中创建功能丰富的游戏和多媒体程序。

Python 有大量的第三方库，很多功能都能通过相应的 Python 库便捷地实现。

目前，Python 有两个版本，一个是 2.x 版，另一个是 3.x 版，这两个版本是不兼容的。由于 3.x 版越来越普及，本书将以最新的 Python 3.5 版本为基础进行讲解。

1.1.2 安装 Python

1. 在 Mac 上安装 Python

如果使用的 Mac 系统是 OS X 10.8~10.10，那么系统自带的 Python 版本是 2.7。要安装最新的 Python 3.5，有如下两个方法。

方法一，从 Python 官网下载 Python 3.5 的安装程序，双击运行并安装；

方法二，如果安装了 Homebrew，直接通过命令 `brew install python3` 安装即可。

2. 在 Linux 上安装 Python

如果使用的是 Linux 系统（需要用户具有 Linux 系统管理经验），那么先下载 Python-3.5.0b4.tgz，再使用解压命令 `tar -zxvf Python-3.5.0b4.tgz`，然后切换到解压的安装目录，执行：

```
[root@www python]#cd Python-3.5.0
[root@www Python-3.5.0]#./configure
[root@www Python-3.5.0]#make
[root@www Python-3.5.0]#makeinstall
```

至此，安装完成。

输入 `python`，如果出现下面的提示：

```
Python 3.5.0 (#1, Aug 06 2015, 14:04:52)
[GCC 4.1.1 20061130 (Red Hat 4.1.1-43)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

出现此提示说明安装成功了。不同 Linux 系统下，显示的第二行有可能不同。

3. 在 Windows 上安装 Python

首先，根据你的 Windows 版本（64 位还是 32 位）从 Python 的官方网站下载 Python 3.5 对应的 64 位安装程序或 32 位安装程序，然后，运行下载的 EXE 安装包，安装界面如图 1-1 所示。

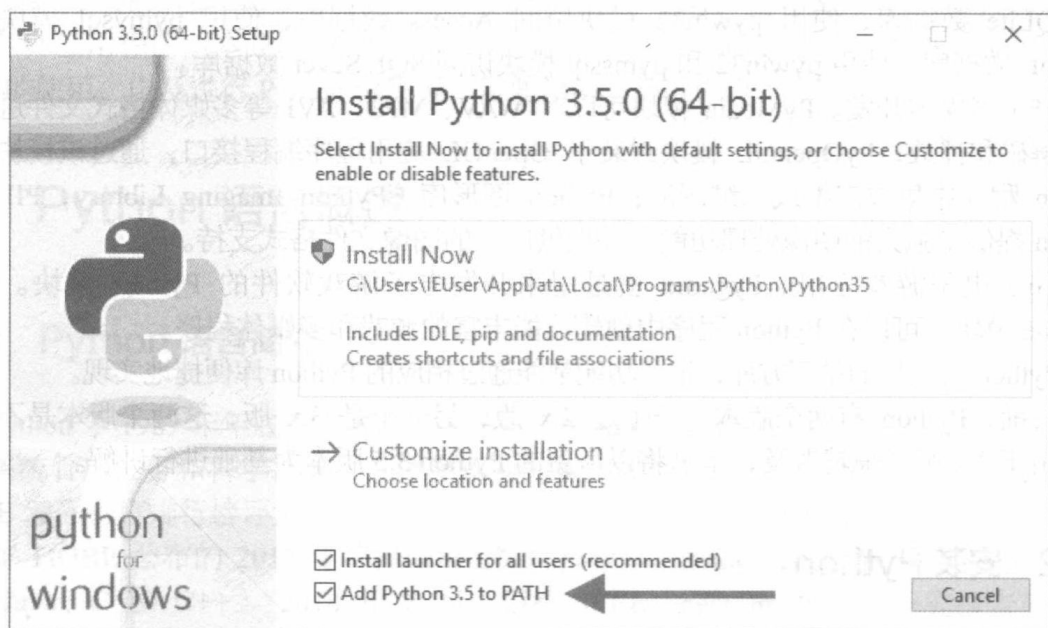


图 1-1 在 Windows 上安装 Python 3.5 的界面

特别要注意在图 1-1 中勾选“Add Python 3.5 to PATH”，然后单击“Install Now”即可完成安装。

安装成功后，使用 `cmd` 打开命令提示符窗口，输入 `python` 后，会出现图 1-2 所示的命令提示符窗口。若在窗口中看到 Python 的版本信息，就说明 Python 安装成功。

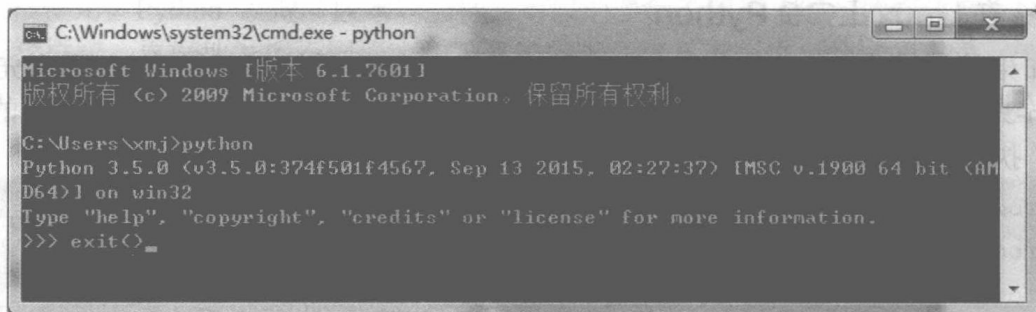


图 1-2 命令提示符窗口

提示符“>>>”表示我们已经在 Python 交互式环境中了，可以输入任何 Python 代码，按回车键后会立刻得到执行结果。现在，输入 `exit()` 并按回车键，就可以退出 Python 交互式环境（或者也可以直接关掉命令行窗口）。

1.1.3 Python 开发环境 IDLE 的启动

安装 Python 后，我们可以从“开始”菜单→“所有程序”→“Python 3.5”→“IDLE (Python 3.5)”来启动 IDLE。IDLE 启动后的初始窗口如图 1-3 所示。

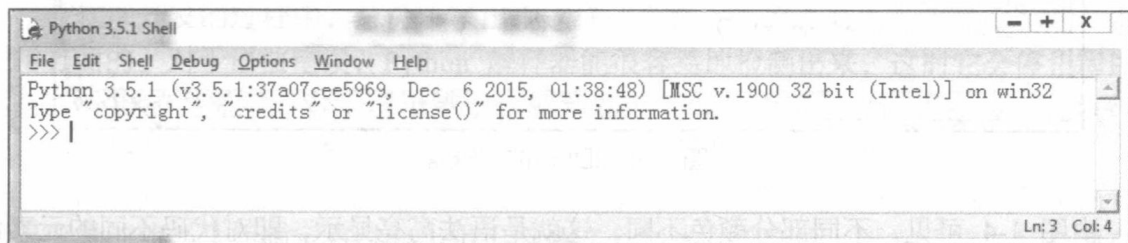


图 1-3 IDLE 的交互式编程模式 (Python shell)

如图 1-3 所示，启动 IDLE 后首先映入我们眼帘的是 Python Shell，我们通过它可以在 IDLE 内部使用交互式编程模式来执行 Python 命令。

如果使用交互式编程模式，那么直接在 IDLE 提示符“>>>”后面输入相应的命令并按回车键执行即可；如果执行成功的话，马上就可以看到执行结果，否则会抛出异常。

例如，查看已安装版本的方法（在所启动的 IDLE 界面标题栏也可以直接看到）：

```
>>> import sys
>>> sys.version
结果: '3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)]'
>>> 3+4
结果: 7
>>> 5/0
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    5/0
ZeroDivisionError: division by zero
```

除此之外，IDLE 含有一个编辑器，可用来编辑 Python 程序（或者脚本）文件；还含有一个调试器来调试 Python 脚本。下面我们从 IDLE 的编辑器开始介绍。

可在 IDLE 界面中使用菜单“File”→“New File”命令启动编辑器（见图 1-4），来创建一个程序文件，输入代码并保存为文件（务必要保证扩展名为“.py”）。

1.1.4 利用 IDLE 创建 Python 程序

IDLE 为开发人员提供了许多有用的特性，如自动缩进、语法高亮显示、单词自动完成以及命令历史等，这些功能能够有效地提高我们的开发效率。下面我们通过一个实例

来对这些特性分别加以介绍。示例程序的源代码如下所示。

```
#示例一
p = input("Please input your password:\n")
if p!="123":
    print("password error! ")
```

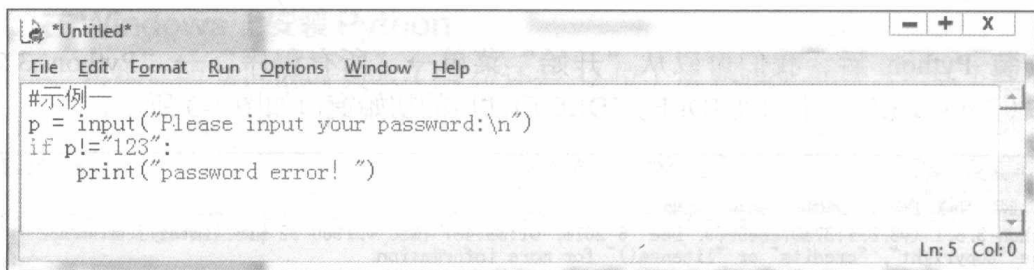


图 1-4 IDLE 的编辑器

由图 1-4 可见，不同部分颜色不同，这就是语法高亮显示，即对代码不同的元素使用不同的颜色进行显示。使用默认设置时，关键字显示为橘红色，注释显示为红色，字符串为绿色，解释器的输出显示为蓝色。在键入代码时，会自动应用这些颜色并突出显示。语法高亮显示的好处是，可以更容易区分不同的语法元素，从而提高可读性；与此同时，语法高亮显示还降低了出错的可能性。比如，如果输入的变量名显示为橘红色，那么，开发人员就需要注意了，这说明该名称与预留的关键字冲突，所以必须对变量更换名称。

单词自动完成是指，当用户输入单词的一部分后，从“Edit”菜单选择“Expand word”项，或者直接按【Alt+/】组合键自动完成该单词。

当用户在 if 关键字所在行的冒号后面按回车键之后，IDLE 将自动进行缩进。在一般情况下，IDLE 将代码缩进一级，即四个空格。如果想改变这个默认的缩进量的话，可以从“Format”菜单选择“New indent width”项来进行修改。初学者需要注意的是，尽管自动缩进功能非常方便，但是不能完全依赖它，因为有时候自动缩进未必完全满足实际需求，所以还需要仔细检查一下。

创建好程序之后，从“File”菜单中选择“Save”保存程序。如果是新文件，则会弹出“Save as”对话框，可以在该对话框中指定文件名和保存位置。保存后，文件名会自动显示在屏幕顶部的蓝色标题栏中。如果文件中存在尚未存盘的内容，则标题栏的文件名前后会有星号出现。

1.1.5 在 IDLE 中运行和调试 Python 程序

1. 运行 Python 程序

要使用 IDLE 执行程序的话，可以从“Run”菜单中选择“Run Module”菜单项（或按【F5】键），该菜单项的功能是执行当前文件。对于示例一的程序，执行情况如图 1-5 所示。

用户输入的密码是“777”，由于密码错误，会出现输出“password error! ”。

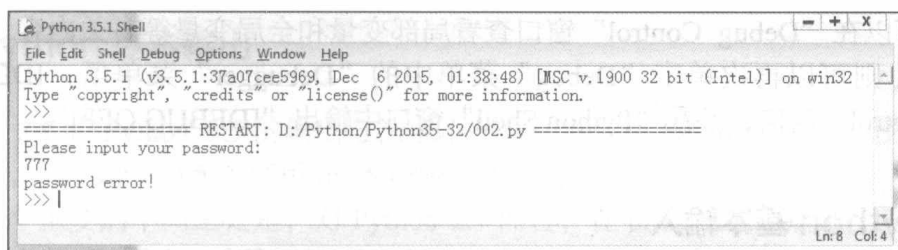


图 1-5 运行界面

2. 使用 IDLE 的调试器

在软件开发的过程中，总免不了出现这样或那样的错误，其中有语法方面的，也有逻辑方面的。对于语法错误，Python 解释器能很容易地检测出来，这时它会停止程序的运行并给出错误提示。对于逻辑错误，解释器就无能为力了，这时程序会一直执行下去，但是得到的运行结果往往却是错误的。因此，我们常常需要对程序进行调试。

最简单的调试方法是直接显示程序数据，例如，可以在某些关键位置用 `print` 语句显示出变量的值，从而确定有没有出错。但是这个办法比较麻烦，因为开发人员必须在所有可疑的地方都插入打印语句。在程序调试完后，还必须将这些打印语句全部清除。

除此之外，用户还可以使用调试器来进行调试。利用调试器，用户可以分析被调试程序的数据，并监视程序的执行流程。调试器的功能包括暂停程序执行、检查和修改变量、调用方法而无须更改程序代码等。IDLE 也提供了一个调试器，帮助开发人员来查找逻辑错误。

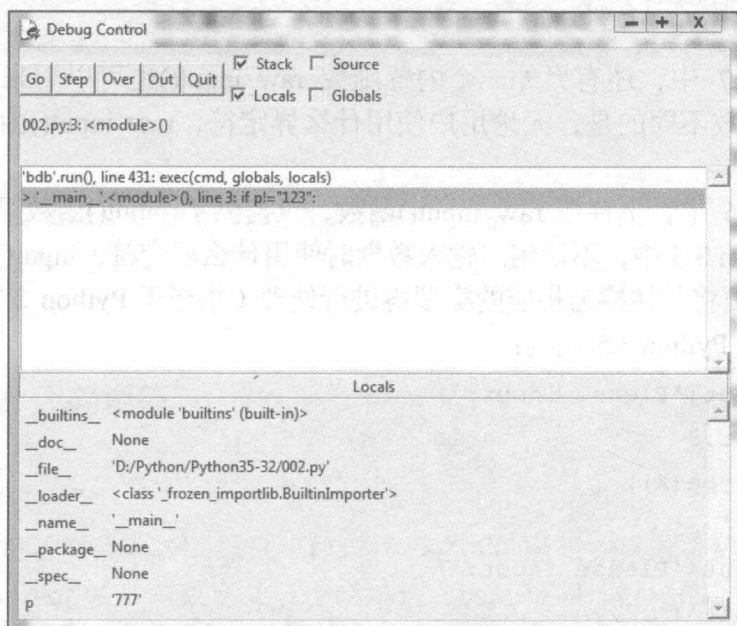


图 1-6 “Debug Control” 调试窗口

下面简单介绍 IDLE 的调试器的使用方法。在“Python Shell”窗口中单击“Debug”菜单中的“Debugger”菜单项，就可以启动 IDLE 的交互式调试器。这时，IDLE 会打开图 1-6 所示的“Debug Control”窗口，并在“Python Shell”窗口中输出“[DEBUG ON]”与“>>>”。

我们可以在“Debug Control”窗口查看局部变量和全局变量等有关内容。如果要退出调试器，则可以再次单击“Debug”菜单中的“Debugger”菜单项，IDLE 会关闭“Debug Control”窗口，并在“Python Shell”窗口中输出 “[DEBUG OFF]”。

1.1.6 Python 基本输入

用 Python 进行程序设计，输入是通过 `input()` 函数来实现的，`input()` 函数的一般格式为：

```
x=input('提示:')
```

该函数返回输入的对象。可输入数字、字符串和其他任意类型对象。

尽管 Python 2.7 和 Python 3.5 形式一样，但 Python 2.x 和 Python 3.x 对该函数的解释略有不同。在 Python 2.7 中，该函数返回结果的类型由输入值时所使用的界定符来决定，如下面的 Python 2.7 代码：

```
>>> x = input("Please input:")
Please input:3                                #没有界定符，整数
>>> print type(x)
<type 'int'>
>>> x = input("Please input:")
Please input:'3'                              #单引号，字符串
>>> print type(x)
<type 'str'>
```

在 Python 2.7 中，还有另外一个内置函数 `raw_input()` 也可以用来接收用户输入的值。与 `input()` 函数不同的是，无论用户使用什么界定符，`raw_input()` 函数返回结果的类型一律为字符串。

在 Python 3.5 中，不存在 `raw_input()` 函数，只提供了 `input()` 函数用来接收用户的键盘输入。在 Python 3.5 中，不论用户输入数据时使用什么界定符，`input()` 函数的返回结果都是字符串，需要将其转换为相应的类型再进行处理（相当于 Python 2.7 中的 `raw_input()` 函数）。如下面的 Python 3.5 代码：

```
>>> x = input('Please input:')
Please input:3
>>> print(type(x))
<class 'str'>
>>> x = input('Please input:')
Please input:'1'
>>> print(type(x))
<class 'str'>
>>> x = input('Please input:')
Please input:[1,2,3]
>>> print(type(x))
<class 'str'>
```