

21世纪高等学校计算机类课程创新规划教材 · 微课版



Python 程序设计案例教程

从入门到机器学习

微课版

◎ 张思民 编著

100个
典型案例

丰富的
教学资源

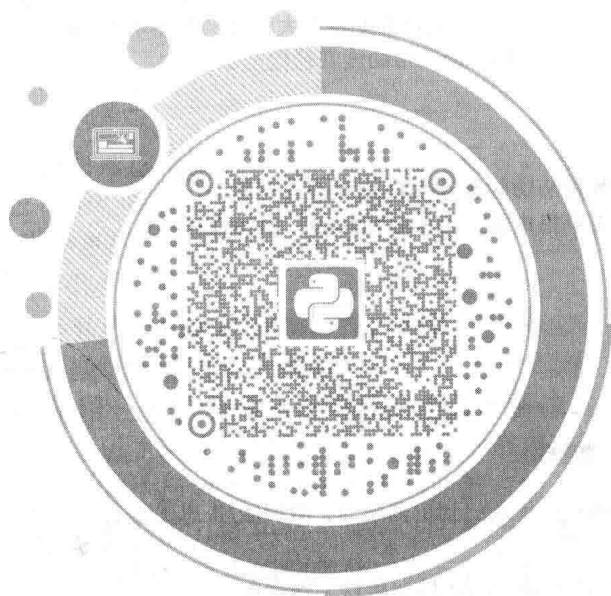
300分钟
视频讲解

- 语法快速入门
- 类与模块
- 多线程
- 绘图及数字图像处理
- 文件与数据库操作
- 图形用户界面设计
- 异常处理及正则表达式
- 网络编程与网络爬虫设计
- 机器学习

清华大学出版社



21世纪高等学校计算机类课程创新规划教材 · 微课版



Python 程序设计案例教程

从入门到机器学习

微课版

◎ 张思民 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍 Python 应用程序设计方法, 主要内容包括 Python 语法快速入门、类与模块、图形用户界面设计、绘图及数字图像处理、文件与数据库操作、多线程、异常处理及正则表达式、网络编程与网络爬虫设计、算法设计与机器学习实战等。本书每章都配有视频教学内容, 以帮助读者学习和理解。

本书讲解详细, 案例丰富, 每一个知识点都配备了大量案例和图示加以说明, 并通过典型案例对 Python 应用程序设计方法进行详细的分析和解释, 帮助读者轻松上手。

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 程序设计案例教程: 从入门到机器学习: 微课版/张思民编著. —北京: 清华大学出版社, 2018 (2018.10重印)

(21 世纪高等学校计算机类课程创新规划教材·微课版)

ISBN 978-7-302-51014-7

I. ①P… II. ①张… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2018) 第 186984 号

责任编辑: 魏江江 赵晓宁

封面设计: 刘 键

责任校对: 李建庄

责任印制: 丛怀宇

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

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

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

印 装 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 16 字 数: 388 千字

版 次: 2018 年 10 月第 1 版 印 次: 2018 年 10 月第 2 次印刷

印 数: 1501~3500

定 价: 49.00 元

产品编号: 080369-01

前 言

Python 是一种面向对象的解释型计算机程序设计语言。这门强大的语言如今在大学和
一些大型软件开发公司中广泛使用，其应用也越来越广。

本书从 Python 初学者的角度进行选材和编写，在编写过程中，注重基础知识和实战应
用相结合，本书有以下几个特点：

(1) 浅显易懂。本书从人们认知规律出发，对每一个概念，用简单的示例或图示来加
以说明，并用短小的典型示例进行分析解释。

(2) 内容新颖而实用。人们学习编程的目的是为了解决人们生活和生产实践中的问
题，本书使用 Python 3.x 以上版本编写代码，大部分章节精选了实用案例，可以帮助解决
读者在学习和实际应用过程中所遇到的一些困难和问题。

(3) 本书在体系结构的安排上将 Python 编程的基础知识和一般编程思想有机结合，对
基础知识重点介绍与其他编程语言不同的部分，而与其他编程语言相同的语法部分则简略
介绍。因此，本书适合具有初步编程语言基础的读者学习。

本书共 9 章，其内容简单介绍如下。

第 1 章主要介绍 Python 的安装与配置、Python 程序编写规范和简单的 Python 程序
示例。

第 2 章简要地介绍数据类型、列表与元组、字典与集合、程序的三大控制结构（顺序
结构、分支结构、循环结构）、函数的基本语法与应用。

第 3 章主要介绍类与模块的基本知识，并介绍了使用 pip 安装和管理扩展模块的方法。

第 4 章主要介绍窗体容器、按钮和文本框等组件、界面布局管理等图形用户界面设计
的方法，还介绍了鼠标与键盘事件及其应用示例。

第 5 章主要介绍绘图与数字图像处理的基本方法。

第 6 章主要介绍数据的存储，包括文件的读写、对 Excel 表格的处理、对 SQLite 数据
库及 MySQL 数据库记录增删改查的操作。

第 7 章主要介绍多线程、异常处理及正则表达式。

第 8 章主要介绍基于 TCP 及 UDP 的套接字编程和网络爬虫程序的设计，并介绍了爬
取网络数据的几个典型案例，还介绍了 Python 在网络程序开发中的方法和技巧，旨在提升
读者的开发技能，达成学以致用之目标。

第 9 章主要介绍了常见数据结构，还介绍了两个 Python 的热门算法设计应用——数据
分析和机器学习的应用案例。

建议教学安排（根据课程设置了两个课时分配方案）：

章 节	方案 1/学时	方案 2/学时
第 1 章 Python 语言快速入门	2	2
第 2 章 Python 语法速览	4	8
第 3 章 类与模块	2	4

续表

章 节	方案 1/学时	方案 2/学时
第 4 章 图形用户界面设计	4	8
第 5 章 绘图与图像处理	4	6
第 6 章 文件与数据库编程（数据存储）	6	12
第 7 章 多线程与异常处理	2	6
第 8 章 网络程序设计	6	14
第 9 章 算法设计及机器学习实战入门	2	4
合计	32	64

学编程必须动手才能见到成效，本书在设计上特别强调讲练结合，注重实践，不仅在讲解的过程中结合大量代码示例，同时适时穿插小项目演练，以锻炼读者的程序设计能力。

有很多人认为 Python 简单易学，但其实 Python 的复杂程度要远高于许多人的想象，诸多概念被隐藏在看似简单的代码背后。这也是 Python 易学难精的主要原因。因此，要强调动手实践，多编写、多练习，熟能生巧，从学习中体验到程序设计的乐趣和成功的喜悦，增强学习信心。

本书由张思民编著。梁维娜参加本书编写及程序测试工作，在此表示感谢。

编 者

2018 年 5 月

目 录

第 1 章 Python 语言快速入门	1
1.1 Python 的安装与配置	1
1.2 运行 Python 程序	2
1.2.1 运行 Python 的方式	2
1.2.2 Python 编写规范	4
1.3 编写简单的 Python 程序	5
习题 1	9
第 2 章 Python 语法速览	10
2.1 Python 的数据类型	10
2.2 列表和元组	12
2.2.1 列表定义与列表元素	12
2.2.2 列表的操作函数	13
2.2.3 元组	15
2.3 字典和集合	16
2.3.1 字典	16
2.3.2 集合	17
2.4 程序控制结构	18
2.4.1 顺序控制语句	18
2.4.2 if 选择语句	21
2.4.3 循环语句	25
2.5 函数	31
2.5.1 函数的定义与调用	31
2.5.2 局部变量与全局变量	32
2.5.3 常用内置函数	33
2.5.4 匿名函数 lambda	35
2.6 案例精选	35
习题 2	41

第 3 章 类与模块	43
3.1 类和对象	43
3.1.1 类的格式与创建对象	43
3.1.2 类的继承	46
3.1.3 运算符重载	48
3.2 模块	48
3.2.1 模块的导入	48
3.2.2 自定义模块	49
3.2.3 常用标准库模块	50
3.2.4 使用 pip 安装和管理扩展模块	54
3.3 案例精选	55
习题 3	57
第 4 章 图形用户界面设计	58
4.1 图形用户界面概述	58
4.1.1 常用设计图形界面的模块	58
4.1.2 tkinter 模块	58
4.2 窗体容器和组件	59
4.2.1 窗体容器和标签组件	59
4.2.2 按钮和事件处理	61
4.3 界面布局管理	63
4.4 文本框组件	65
4.5 其他常用组件	68
4.5.1 单选按钮和复选框	68
4.5.2 标签框架、下拉列表框和滚动文本框	69
4.6 菜单与对话框	71
4.6.1 菜单	71
4.6.2 对话框	73
4.7 鼠标键盘事件	78
4.7.1 鼠标事件	78
4.7.2 键盘事件	80
4.8 案例精选	81
习题 4	83
第 5 章 绘图及图像处理	85
5.1 绘制图形	85

5.1.1	用画布组件绘图	85
5.1.2	用 turtle 模块绘图	88
5.2	数字图像处理基础	90
5.2.1	Python 图像处理类库 PIL	90
5.2.2	图像处理技术	91
5.3	案例精选	94
	习题 5	105
第 6 章	文件与数据库编程（数据存储）	106
6.1	文件目录	106
6.1.1	文件目录函数	106
6.1.2	文件目录操作	106
6.2	文件的读写操作	108
6.2.1	文件操作函数	108
6.2.2	打开和关闭文件	109
6.2.3	读取文件操作	109
6.2.4	写入文件操作	111
6.2.5	二进制文件的读写	114
6.2.6	对 Excel 数据的读写操作	115
6.2.7	处理 JSON 格式数据	119
6.3	Python 数据库编程	124
6.3.1	SQLite 数据库编程	124
6.3.2	操作 MySQL 数据库	129
6.4	案例精选	133
6.4.1	多功能文本编辑器	133
6.4.2	保存结构化数据	137
6.4.3	英汉小词典设计	139
	习题 6	142
第 7 章	多线程与异常处理	144
7.1	多线程编程	144
7.1.1	线程与多线程	144
7.1.2	线程的生命周期	145
7.1.3	创建线程的 threading.Thread 类	146
7.1.4	线程同步	150
7.2	异常处理	153
7.2.1	Python 中的常见标准异常	153

7.2.2	异常的捕捉与处理	154
7.3	正则表达式	156
7.3.1	字符匹配与匹配模式	156
7.3.2	正则表达式的规则	157
7.3.3	正则表达式 re 模块的方法	158
7.4	案例精选	160
	习题 7	163
第 8 章	网络程序设计	164
8.1	套接字 Socket 编程基础	164
8.1.1	套接字 Socket	164
8.1.2	TCP 与 UDP	165
8.2	套接字 Socket 程序设计	166
8.2.1	基于 TCP 的客户机/服务器模式	166
8.2.2	基于 UDP 的网络程序设计	169
8.3	网络应用案例精选	170
8.3.1	文件传输协议 FTP 应用	170
8.3.2	基于 TCP 的端口扫描器	172
8.3.3	远程控制计算机	174
8.3.4	网络域名解析	176
8.4	网络爬虫实战入门	178
8.4.1	抓取网页数据	178
8.4.2	网络爬虫简介	182
8.5	网络爬虫案例精选	188
8.5.1	爬取某网站大学排名榜	188
8.5.2	爬取网络版小说——《红楼梦》	189
8.5.3	爬取天气预报信息	193
8.5.4	网络爬虫利器——Requests	195
8.5.5	爬取购物网站商品信息	198
8.6	Python Web 服务简介	201
	习题 8	203
第 9 章	算法设计及机器学习实战入门	204
9.1	常见的数据结构	204
9.1.1	堆栈	204
9.1.2	队列	205
9.1.3	链表	207

9.1.4	树	213
9.2	迷宫问题算法设计	219
9.3	曲线点抽稀算法	223
9.3.1	道格拉斯-普克算法	224
9.3.2	垂距限值算法	227
9.4	Python 机器学习实战入门	229
9.4.1	机器学习及其算法	229
9.4.2	机器学习应用实例	231
9.5	机器学习案例精选	233

Python 是一种面向对象的解释型计算机编程语言。Python 语言具有通用性、高效性、跨平台移植性和安全性，广泛应用于科学计算、自然语言处理、图形图像处理、游戏开发、Web 应用等方面，在全球范围内拥有众多开发者专业社群。



视频录像

1.1 Python 的安装与配置

1. Python 的下载和安装

学习 Python 需要一个程序开发环境。只有安装并配置了 Python 系统开发环境之后，Python 程序才能运行。经过长期发展，Python 同时流行两个不同的版本，它们分别是 2.7.x 和 3.x 版本。注意，这两个版本是不兼容的，本书是基于 3.x 版本编写的。

可以在 Python 的官方网站 <https://www.python.org/downloads/> 下载 Python 安装包，如图 1.1 所示。

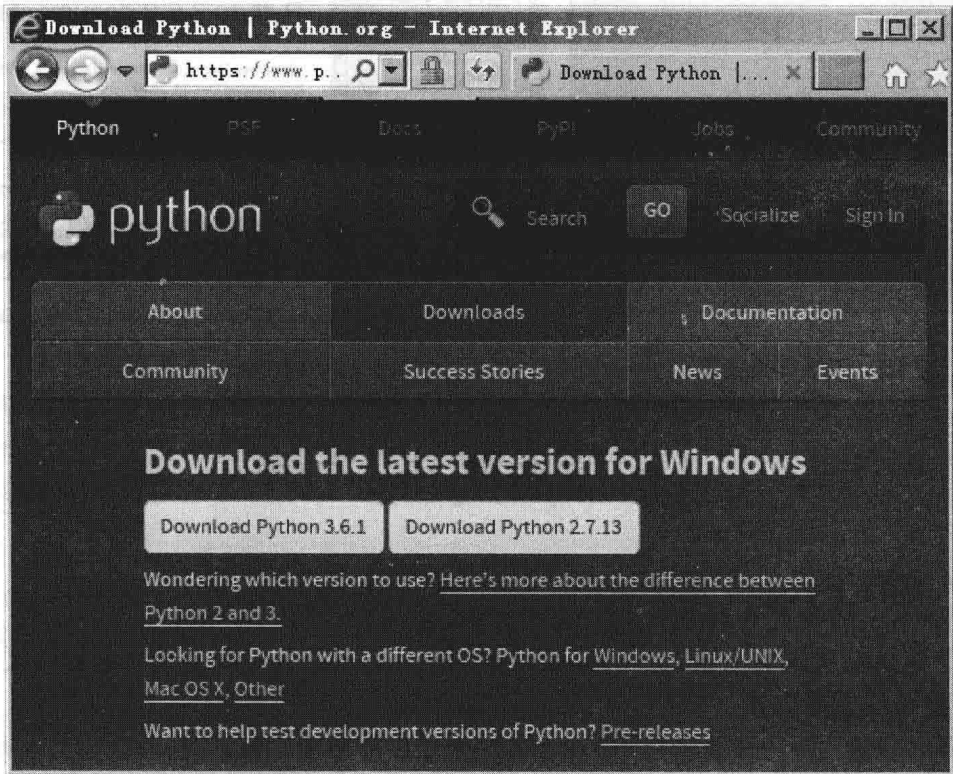


图 1.1 Python 安装包下载

下载 Python 安装包后，就可以运行安装程序，进入 Python 的安装界面，按照提示完成安装。

2. Python 开发环境的配置

安装完成后，还需要配置 Python 的环境变量。这里假设 Python 系统安装在“C:\Python\”目录下。

在 Windows 操作系统下，右击桌面上的“计算机”图标，在弹出的快捷菜单中选择“属性”选项，打开“系统属性”对话框。在“系统属性”对话框的“高级”选项卡中，单击“环境变量”按钮，打开“环境变量”对话框。在“系统变量”列表框中，双击 Path 变量，打开“编辑系统变量”对话框。在“变量值”文本框中输入填写 Python 的安装路径，这里输入“C:\Python\”，如图 1.2 所示。

配置完成后，不论当前目录是在何处，执行 Python 命令时，操作系统都会执行这条命令，从而可以在任何目录下执行 Python 源程序代码。

3. Python 在线帮助文档

Python 还提供非常完善的 Python 帮助文档，这是进行程序设计的工具。Python 帮助文档在 Python 安装目录的 doc 目录下，双击即可打开，如图 1.3 所示。



图 1.2 设置环境变量

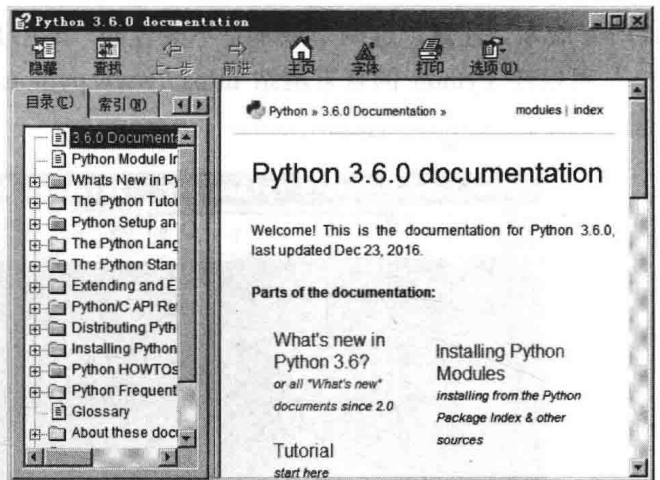


图 1.3 Python 在线帮助文档

1.2 运行 Python 程序

1.2.1 运行 Python 的方式

运行 Python 有两种方式：一种是命令行的交互方式；另一种是使用源程序文件方式。

1. 命令行交互方式

选择“开始”→“所有程序”→Python→IDLE 菜单项，启动 Python 运行环境，进入交互编程方式。

在 IDLE 提示符“>>>”后面输入单条 Python 语句，按 Enter 键执行该语句，马上就可

以看到执行结果，如图 1.4 所示。

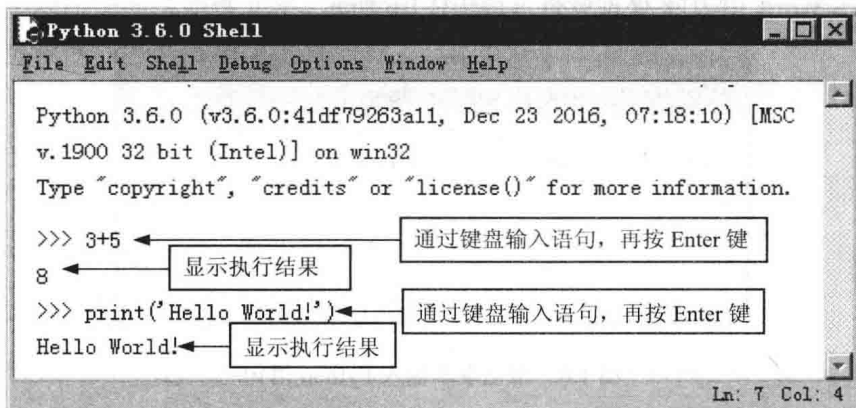


图 1.4 Python 交互方式

2. 源程序的文件方式

Python 应用程序的开发方式：使用文本编辑器，编写 Python 源程序，并保存扩展名为 py 的文件。

Python 应用程序的开发过程如图 1.5 所示。

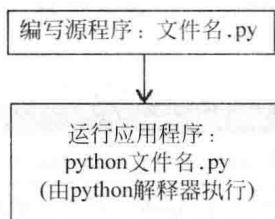


图 1.5 Python 程序的开发过程

(1) 建立 Python 源文件

要建立一个 Python 程序，首先创建 Python 的源代码，即建立一个文本文档，包括有符合 Python 规范的语句。

开发一个 Python 程序必须遵循如下基本原则：

- Python 程序中一行就是一条语句，语句结束不需要使用分号；
- Python 采用缩进格式标记一组语句，缩进量相同的是同一组语句，也称为程序段；
- 一条语句也可以分多行书写，用反斜杠 (\) 表示续行。

例如：

```
a = (3 + 2) * (6 - 4) * (8 + 6) \
* (12 - 5)
```

和

```
a = (3 + 2) * (6 - 4) * (8 + 6) * (12 - 5)
```

是相同的。

下面编写一个最简单的 Python 程序，这里用记事本或其他纯文本编辑器输入下列语句（不能使用 MS Word 等文字处理软件），如图 1.6 所示。

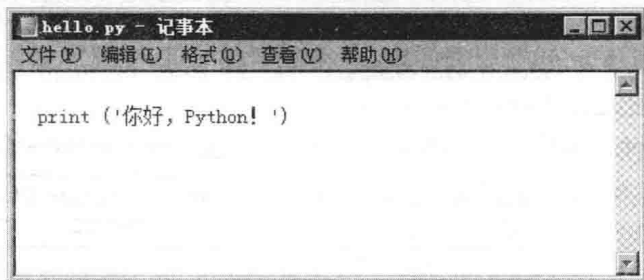


图 1.6 用记事本输入 Python 语句

将上述源代码保存到 D:\pytest 目录下，命名为 hello.py 文件。

(2) 运行程序

下面在命令控制台窗口中运行程序。

在命令控制台窗口中，在提示符“D:\pytest>”后输入运行程序命令：

```
python hello.py
```

注意：如果当前目录不是“D:\pytest”，则应使用 cd 命令，进入到该目录，如图 1.7 所示。

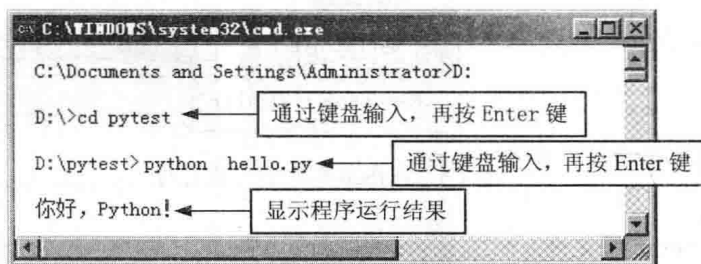


图 1.7 运行 hello.py 程序

1.2.2 Python 编写规范

1. 标识符命名规则

① 文件名、类名、模块名、变量名、函数名等标识符的第一个字符必须是字母表中字母或下划线（_）。

② 标识符的其他部分由字母、数字和下划线组成，且标识符区分大小写字母。

③ 源文件的扩展名为 py。

2. 代码缩进

Python 程序依靠代码块的缩进来体现代码之间的逻辑关系。通常，以 4 个空格或制表符（按 Tab 键）为基本缩进单位。缩进量相同的一组语句，称为一个语句块或程序段。需要注意的是，空格的缩进方式与制表符的缩进方式不能混用。

3. 程序中的注释语句

注释是程序中的说明性文字，是程序的非执行部分。它的作用是为程序添加说明，增加程序的可读性。Python 语言使用两种方式对程序进行注释：

① 单行注释以“#”符号和一个空格开头。如果在语句行内注释（即语句与注释同在一行），注释语句符与语句之间至少要用两个空格分开。

例如：

```
print('Hello') # 输出显示语句
```

② 多行注释用 3 个单引号'''或 3 个双引号"""将注释括起来。

例如：

```
'''  
这是多行注释，用三个单引号  
这是多行注释，用三个单引号  
这是多行注释，用三个单引号  
'''
```

4. 代码过长的折行处理

当一行代码较长，需要折行（换行）时，可以使用反斜杠\延续行。

例如：

```
io3 = can.create_oval(65,70,185,170, outline='yellow', fill='yellow')
```

可以写成：

```
io3 = can.create_oval(65,70,185,170, \  
                    outline='yellow', \  
                    fill='yellow')
```

1.3 编写简单的 Python 程序

【例 1-1】 在命令窗口中显示输出内容的程序。

程序代码如下：

```
str = 'Python 语言入门很简单。\\n明白了吗?'  
print(str)
```

操作步骤如下：

① 在编辑工具中输入上述程序，如图 1.8 所示。

```
str = 'Python 语言入门很简单。\\n明白了吗?'  
print(str)
```

图 1.8 在编辑工具中输入源程序

将输入完成的源程序保存为 `ex1_1.py`。

② 执行程序：

```
python ex1_1.py
```

其运行结果如图 1.9 所示。

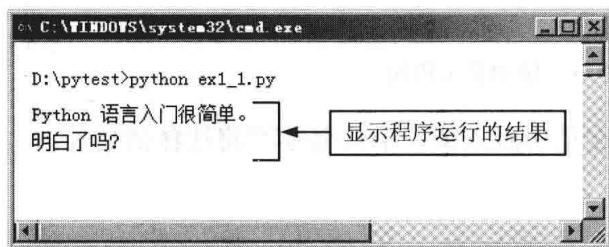


图 1.9 运行结果

【程序说明】

`print()` 为命令窗口输出语句，输出语句中的“`\n`”是换行符，换行符后面的字符将在下一行显示。

【例 1-2】 输出语句 `print()` 有“原样照印”及简单计算功能。

```
print ('5 + 3 = ', 5+3)
```

用单引号括起来的 '5+3=' 将按原样显示，称为“原样照印”。而没有用单引号括起来的 5+3 将进行加法计算

将其保存为 `ex1_2.py`。执行程序：

```
python ex1_2.py
```

其运行结果如图 1.10 所示。

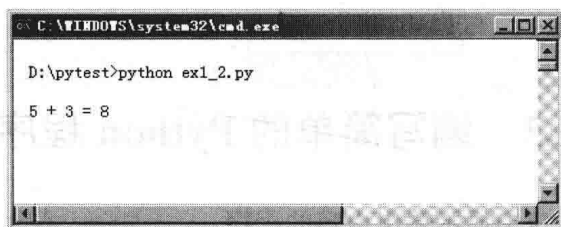


图 1.10 输出语句的“原样照印”及运算功能

【例 1-3】 应用输出语句的“原样照印”功能，输出一个用“*”号组成的三角形。程序代码如下：

```
print('*')
print('* *')
print('* * *')
print('* * * *')
```

将其保存为 `ex1_3.py`，执行程序：


```
python ex1_3.py
```

其运行结果如图 1.11 所示。

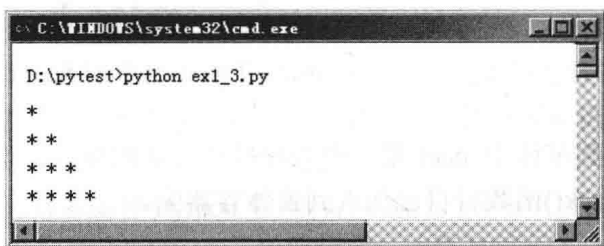


图 1.11 输出用“*”组成的三角形

【例 1-4】 在窗体中显示输出的内容。

程序代码如下：

```
import tkinter
top = tkinter.Tk()
labell = tkinter.Label(top, text = '在窗体中显示输出内容!')
labell.pack()
top.mainloop()
```

将其保存为 ex1_4.py，运行程序：

```
python ex1_4.py
```

其运行结果如图 1.12 所示。

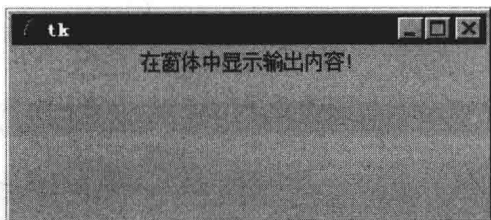


图 1.12 Python 窗体程序的运行结果

【程序说明】

① 程序的第 1 行：

```
import tkinter
```

是一条导入模块的 import 语句。import 语句为编译器找到程序使用的 tkinter 模块。

② 在程序的第 2 行：

```
top = tkinter.Tk()
```

表示创建一个顶层窗体对象。Tk 是模块 tkinter 的类，通过 tkinter.Tk() 创建窗体对象。

③ 程序的第 3 行：