

高等教育质量工程信息技术系列示范教材

Python

开发案例教程

张基温 魏士靖 编著



清华大学出版社

高等教育质量工程信息技术系列示范教材

Python开发案例教程

张基温 魏士靖 编著

清华大学出版社
北京

内 容 简 介

本书是一本 Python 基础教材。全书以 Python 3.0 为蓝本，分 6 章介绍。

第 1 章从模仿计算器进行简单的计算入手，带领读者迈进 Python 门槛；顺势引导读者掌握 Python 模块的用法、变量的用法；再进一步引入选择结构和重复结构，使读者有了程序和算法的基本概念，进入程序设计的殿堂。第 2 章首先介绍 Python 特有的数据对象与变量之间的关系，然后较详细地介绍了 Python 各种内置的数据类型。第 3 章从正常处理和异常处理两个角度介绍 Python 程序过程的两种基本组织形式：函数和异常处理，并介绍与之相关的名字空间和作用域的概念。第 4 章把读者从面向过程带到面向类的程序设计，内容包括类与对象、类与对象的通用属性与操作、类的继承。第 5 章为 Python GUI 开发，使读者具备开发友好界面程序的能力。第 6 章为 Python 应用开发举例，从数据处理和网络应用两个最基本应用领域，培养读者的应用开发能力。

本书力求内容精练、概念准确、代码便于阅读、习题丰富全面、适合教，也容易学；例子分正反两种，以利正本清源。为了便于初学者很快使用 Python 以丰富的模块支撑的环境，书后给出了 Python 内置函数、Python 文件和目录管理、Python 3.0 标准异常类体系和 Python 标准模块库目录。

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 开发案例教程 / 张基温等编著. —北京：清华大学出版社，2019

(高等教育质量工程信息技术系列示范教材)

ISBN 978-7-302-51190-8

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 210613 号

责任编辑：白立军 常建丽

封面设计：常雪影

责任校对：梁 毅

责任印制：宋 林

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：北京密云胶印厂

经 销：全国新华书店

开 本：185mm×260mm

印 张：19 彩 插：2

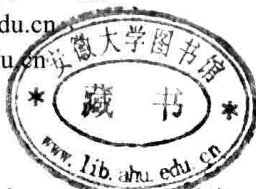
字 数：472 千字

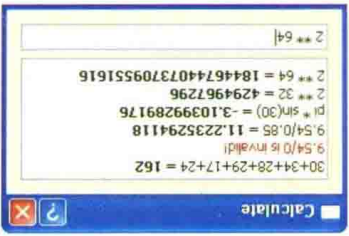
版 次：2019 年 1 月第 1 版

印 次：2019 年 1 月第 1 次印刷

定 价：49.00 元

产品编号：079725-01





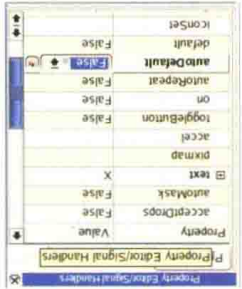
(a) 多行文本框



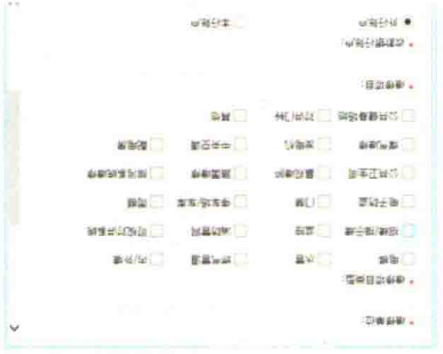
(b) 标签、单行文本框与按钮



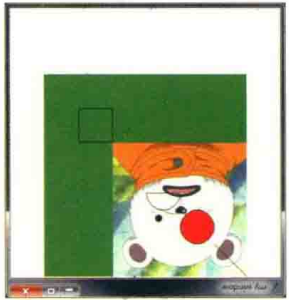
(c) 菜单条与菜单



(d) 列表框与滚动条



(e) 复选框与滚动条



(f) 画布

图 5.1 常用组件形成的界面



图 5.9 标签制作示例



图 5.10 代码 5-10 的运行结果



图 5.11 三春晖图片浏览器运行效果示例

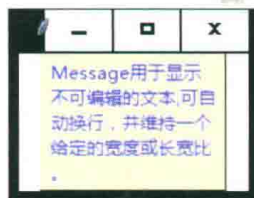
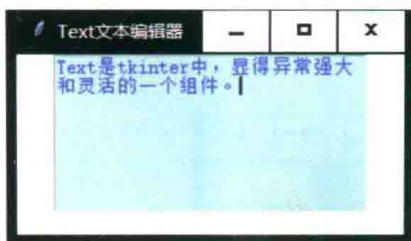


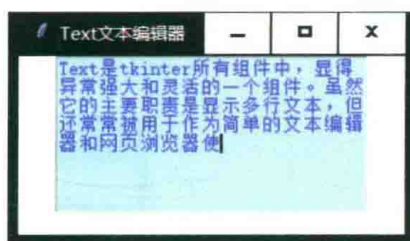
图 5.12 一个简单的消息框



图 5.13 简易四则计算器运行情况

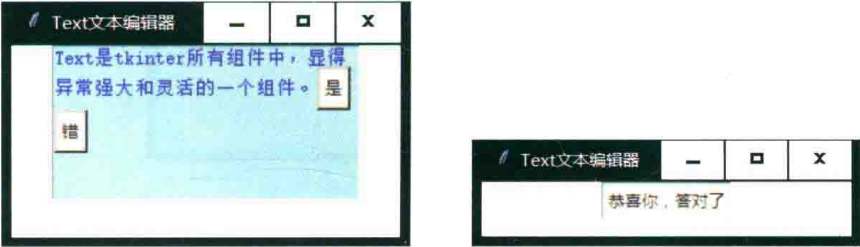


(a) 初始显示



(b) 手动插入文字后的显示

图 5.14 代码 5-14 的执行情况



(a) 初始显示 (b) 按了“是”后的显示

图 5.15 代码 5-15 的执行情况



图 5.16 代码 5-16 的执行情况

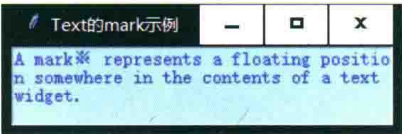


图 5.17 在记号处插入字符※



图 5.18 代码 5-18 的运行情况



图 5.19 代码 5-19 的运行情况

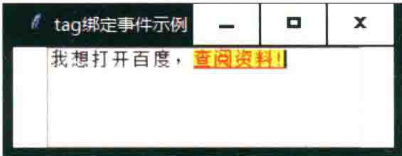


图 5.20 代码 5-20 的客户端

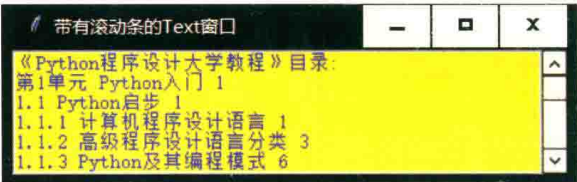


图 5.21 加入了滚动快的 Text



(a) 一种单选框样式

(b) 另一种单选框样式

图 5.22 代码 5-22 的执行情况

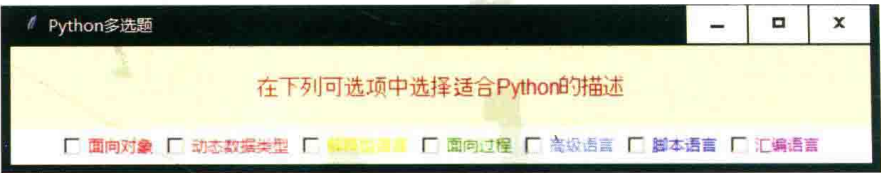


图 5.23 代码 5-23 的执行情况



图 6.20 执行代码 6-16 显示出的页面

前 言

近年来，一种程序设计语言日渐粲然，使许多红极一时的程序设计语言黯然失色，得到人们的空前青睐，使得在学界和业界出现了揭竿而起、应者云集的景象。这种程序设计语言就是 Python。本书也想在此时为熊熊燃起的 Python 烈火再添上一把柴。

(一)

Python 之所以能够冉冉升起，在于其鲜明的特色。

Python 简单、易学。它虽然是用 C 语言写的，但是它摒弃了 C 语言中任性不羁的指针，降低了学习和应用的难度。

Python 代码明确、优雅。其代码描述具有伪代码风格，使人容易理解；其强制缩进的规则使得代码具有极佳的可读性。

Python 自由、开放。Python 是 FLOSS (Free/Libre and Open Source Software, 自由/开放源码软件) 之一。它支持向不同的平台上移植，允许部分程序用应用广泛的 C/C++ 语言编写；它可提供脚本功能，允许把 Python 程序嵌入 C/C++ 程序中。它还鼓励更优秀者的创造、改进与扩张，因此使其在短短的发展历程中形成异常庞大、几乎覆盖一切应用领域的标准库和第三方库，为开发者提供了丰富的可复用资源和便利的开发环境。

(二)

为了彰显优势，Python 博采众长、趋利避害，形成一套独特的语法体系。其中有些语法现象是用别的语言的语法体系解释不清楚的，强行解释反而会误导学者。本书力图正本清源，从基本理论出发，对 Python 的语法给出一个清晰而本原的概念和解释，以此为基础快速而扎实地将学习者带进 Python 应用开发中展现才干。

本书共分 6 章。第 1 章从初中水平的读者就能懂的计算开始，将读者引进 Python 世界。同时，插进一些最基本的语法知识，如输入输出、变量、模块，然后通过选择和循环结构带领读者在简单算法中试水。

第 2~4 章在第 1 章的基础上深入浅出地介绍数据类型、面向过程的结构和面向对象的结构。在此期间让学习者进一步理解对象与变量、各种原子类型和内置容器类型、函数、异常处理、名字空间与作用域。

第 5、6 章是应用开发。第 5 章为 Python UGI 开发，第 6 章为 Python 应用开发举例。这两章的内容突出了 Python 应用开发的两个要素：领域知识的了解和相应模块的应用。

(三)

著名心理学家皮亚杰创建的结构主义对教师的主要职责定义是为学习者创建学习环境。教材是为学习者创建的一种学习环境。除正文的内容选择、顺序安排之外，还有例题、练习题和附录。

本书例题力求代码精干，以便读者理解。练习题是以大节为单位进行组织的，并且题型多样，针对性强，便于学习者学习某一节后，立即可以从不同角度进行检测。

鉴于已经出版的多种教材中存在的对 Python 基本概念解释含混，甚至错误的情况，本书还收集了一些著作中的错误概念作为反例放在相关的习题中，供读者分析、批判，以正本清源，提高读者对 Python 语法的辨别、理解和应用能力。

本书的附录由四部分组成：Python 内置函数、Python 3.0 标准异常类结构、文件与目录管理和 Python 标准模块库目录。这些内容相当于一个常用手册，可以为初学者提供一个继续学习或扩展学习的环境。

(四)

进行编写中，收集并设计了多种类型的习题，并且在每一节后面都给出了相应的练习题。作为 Python 教材，本书把附录和习题作为正文之外的两个重要的学习环境。本书的附录包括操作符、内置函数、模块目录和异常类结构。这些内容相当于一本简明的应用手册，会给想继续深入并提高自己 Python 开发能力的学习者提供一个扩展的环境。

这些附录也表明 Python 开源代码的特点和社区广大热心者的支持，是 Python 生命力的源泉。虽然目前 Python 已经有上千种模块可以被利用，而附录中列出的 Python 3x 的标准模块库仅有 20 多项，但已足以对 Python 的应用范围画出一个轮廓。

(五)

在本书出版之前，魏士靖细心进行了代码校验和文字校对，并制作了 PPT；吴灼伟设计了书中部分插图；刘砚秋、赵忠孝、姚威、张展为、张秋菊、史林娟、张有明等也参与了部分工作。在此谨表谢意。

本书的出版是我在程序设计教学改革工作中跨上的一个新台阶。本人衷心希望得到有关专家和读者的批评和建议，也希望能多结交一些志同道合者，把本书改得更好一些。

张基温

戊戌初夏于穗小海之畔

目 录

第 1 章 一个万能计算器	1
1.1 简单算术计算	1
1.1.1 Python 算术操作符	1
1.1.2 回显与 print()	2
1.1.3 Python 表达式计算规则	3
练习 1.1	3
1.2 使用内置数学函数计算	4
1.2.1 函数与内置函数	4
1.2.2 Python 计算型内置函数对象	4
练习 1.2	5
1.3 利用 math 模块进行计算	6
1.3.1 模块化程序设计与 Python 模块	6
1.3.2 导入模块或对象	6
1.3.3 math 模块中的常量与函数	8
练习 1.3	10
1.4 使用变量计算	11
1.4.1 数据对象、变量与赋值	11
1.4.2 几种不同的赋值方式	12
1.4.3 Python 标识符与保留字	13
1.4.4 input() 函数	14
练习 1.4	15
1.5 选择型计算	16
1.5.1 if-else 型选择的基本结构	16
1.5.2 选择表达式	17
1.5.3 if-else 蜕化结构	18
1.5.4 if-else 嵌套	18
1.5.5 if-elif 选择结构	19
练习 1.5	20
1.6 重复型计算	21
1.6.1 while 语句	22
1.6.2 for 语句	23
1.6.3 循环嵌套	25

1.6.4	在 IDLE 中执行功能完整的代码段	27
1.6.5	循环中断语句与短路控制	28
	练习 1.6	29
1.7	穷举与迭代	30
1.7.1	穷举	30
1.7.2	迭代	32
	练习 1.7	37
第 2 章	Python 数据对象	40
2.1	Python 数据对象的属性	40
2.1.1	Python 数据对象的类型	40
2.1.2	Python 对象的身份码 ID 与判是操作	43
2.1.3	可变对象与不可变对象	45
2.1.4	Python 数据对象生命期与垃圾回收	46
2.1.5	Python 小整数对象池与大整数对象池	49
	练习 2.1	51
2.2	Python 基本数据类型	52
2.2.1	bool 类型	52
2.2.2	int 类型	56
2.2.3	float 类型	58
2.2.4	complex 类型	59
2.2.5	Python 数据类型转换	60
	练习 2.2	62
2.3	序列	64
2.3.1	序列对象的构建	64
2.3.2	序列通用操作	66
2.3.3	列表的个性化操作	72
	练习 2.3	74
2.4	字符串	77
2.4.1	字符串编码与解码	77
2.4.2	字符串的个性化操作	81
2.4.3	字符串格式化与 format() 方法	83
2.4.4	print() 函数的格式控制	86
2.4.5	正则表达式	89
	练习 2.4	95
2.5	字典与集合	96
2.5.1	字典	96
2.5.2	集合	100
	练习 2.5	103

2.6	Python 数据文件	106
2.6.1	数据文件概述	106
2.6.2	open()函数	107
2.6.3	文件属性与方法	109
	练习 2.6	110
第 3 章	Python 过程组织与管理	112
3.1	Python 函数	112
3.1.1	函数及其关键环节	112
3.1.2	Python 函数参数技术	116
3.1.3	Python 函数的第一类对象特性	120
3.1.4	函数标注	121
3.1.5	递归	122
3.1.6	lambda 表达式	125
	练习 3.1	125
3.2	Python 异常处理	129
3.2.1	异常处理的基本思路与异常类型	129
3.2.2	try-except 语句	131
3.2.3	异常类型的层次结构	133
3.2.4	else 子句与 finally 子句	133
3.2.5	异常的人工触发: raise 与 assert	134
	练习 3.2	135
3.3	Python 命名空间与作用域	136
3.3.1	Python 命名空间	136
3.3.2	Python 作用域	139
3.3.3	Python 名字解析的 LEGB 规则	144
	练习 3.3	145
第 4 章	面向类的程序设计	147
4.1	类及其组成	147
4.1.1	类模型与类语法	147
4.1.2	对象的生成与__init__()方法	150
4.1.3	最小特权原则与成员访问限制	152
4.1.4	实例方法、静态方法与类方法	155
	练习 4.1	157
4.2	Python 类的内置属性、方法与函数	158
4.2.1	类的内置属性	158
4.2.2	获取类与对象特征的内置函数	160
4.2.3	操作符重载	163
4.2.4	Python 类属性配置与管理内置方法	166

练习 4.2.....	173
4.3 类的继承.....	175
4.3.1 类的继承及其关系测试.....	176
4.3.2 新式类与 object.....	178
4.3.3 子类访问父类成员的规则.....	180
4.3.4 子类实例的初始化与 super.....	180
练习 4.3.....	185
第 5 章 Python GUI 开发	188
5.1 组件、布局与事件处理.....	188
5.1.1 组件.....	188
5.1.2 布局与布局管理器.....	192
5.1.3 事件绑定与事件处理.....	195
练习 5.1.....	200
5.2 GUI 程序结构.....	201
5.2.1 基于 tkinter 的 GUI 开发环节.....	201
5.2.2 面向对象的 GUI 程序框架.....	204
练习 5.2.....	206
5.3 GUI 制作示例.....	206
5.3.1 Label 与 Button.....	207
5.3.2 Entry 与 Message.....	213
5.3.3 Text 与滚动条.....	217
5.3.4 选择框.....	224
5.3.5 菜单.....	230
练习 5.3.....	233
第 6 章 Python 应用开发举例	234
6.1 Python 数据库操作.....	234
6.1.1 数据库与 SQL.....	234
6.1.2 用 pyodbc 访问数据库.....	235
6.1.3 SQLite3 数据库.....	242
练习 6.1.....	244
6.2 Python Socket 编程.....	245
6.2.1 TCP/IP 与 Socket.....	245
6.2.2 Socket 模块与 Socket 对象.....	249
6.2.3 TCP 的 Python Socket 编程.....	251
6.2.4 UDP 的 Python Socket 编程.....	254
练习 6.2.....	256
6.3 Python WWW 应用开发.....	258
6.3.1 WWW 及其关键技术.....	258

6.3.2	urllib 模块库	264
6.3.3	urllib.parse 模块与 URL 解析	266
6.3.4	urllib.request 模块与网页抓取	267
6.3.5	网页提交表单	270
6.3.6	urllib.error 模块与异常处理	271
6.3.7	webbrowser 模块	272
	练习 6.3	273
附录		275
附录 A	Python 内置函数	275
A.1	数学运算	275
A.2	逻辑判断	275
A.3	容器操作	276
A.4	字符串相关	276
A.5	类型转换	278
A.6	I/O 操作	278
A.7	反射相关	278
A.8	其他	280
附录 B	Python 3.0 标准异常类结构 (PEP 348)	280
附录 C	文件与目录管理	281
C.1	文件访问函数	281
C.2	目录操作	282
C.3	获取或判断文件和路径属性的函数	282
C.4	路径操作	283
C.5	文件压缩 (zipfile 模块)	283
C.6	文件复制 (shutil 模块)	285
附录 D	Python 标准模块库目录	285
D.1	文本	285
D.2	二进制数据	286
D.3	数据类型	286
D.4	数学	286
D.5	函数式编程	286
D.6	文件与目录	286
D.7	持久化	287
D.8	压缩	287
D.9	文件格式化	287
D.10	加密	287
D.11	操作系统工具	287
D.12	并发与并行	288

D.13	进程间通信.....	288
D.14	互联网相关.....	288
D.15	HTML 与 XML.....	289
D.16	互联网协议与支持.....	289
D.17	多媒体.....	290
D.18	国际化.....	290
D.19	编程框架.....	290
D.20	Tk 图形用户接口	290
D.21	开发工具.....	291
D.22	调试.....	291
D.23	运行时.....	291
D.24	解释器.....	291
D.25	导入模块.....	292
D.26	Python 语言	292
D.27	其他.....	292
D.28	Windows 相关.....	292
D.29	UNIX 相关.....	292
参考文献.....		294

第1章 一个万能计算器

Python 提供了两种基本的编程模式：文本编程模式和交互编程模式。文件编程模式是将程序代码以脚本形式写到一个文件里，然后用 Python 执行，可以做到一次编程，多次执行。交互编程模式也称命令编程模式，使用 IDLE（Integrated DeveLopment Environment/Integrated Development and Learning Environment）就会弹出图 1.1 所示的 Python 交互编程环境。在一大堆信息显示之后，出现 Python 命令提示符>>>。在提示符>>>后面可以输入一个 Python 命令，按 Enter 键，便可以由 Python 解释器解释执行，给出结果。这种对代码及时反馈的模式非常适合初学者验证语法规则，对有经验的 Python 人员也可用于尝试新的 API、库以及函数，但只能适合于一次编程，一次执行。对于一般人来说，它可以被当作是一个万能计算器。

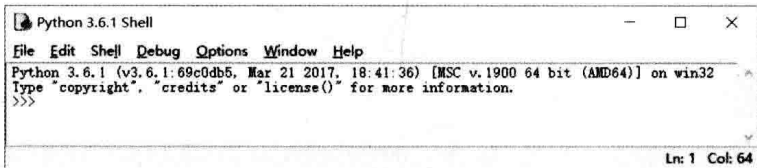


图 1.1 Python 3.6.1 自带的 IDLE

1.1 简单算术计算

加、减、乘、除是最常用的计算。在 Python IDLE 中，也可以方便地进行这些计算。

1.1.1 Python 算术操作符

操作符（operators）是计算操作简洁表示。为了便于用户计算，各种计算机程序设计语言中都会将一些最常用的算术操作符供用户使用。表 1.1 为 Python 内置的算术操作符。

表 1.1 Python 内置的算术操作符（假定 $a = 10$, $b = 3$ ）

优先级	操作符	描 述	实 例	结合方向
高	+, -	正负号	+ 5, - 3	右先
	**	幂	$a ** b$ 为 10 的 3 次方, 返回 1000	
中	//	整除	$a // b$ 返回 3	左先
	%	取模	$a \% b$ 返回 1	
	/	浮点除	a / b 返回 3.3333333333333335	
	*	两个数相乘	$a * b$ 返回 30	
低	+	两对象相加	$a + b$ 返回 13	
	-	两对象相减	$a - b$ 返回 7	

说明:

(1) 注意, Python 中的算术操作符与普通数学中的算术操作符有所不同, 如用 “*” 表示乘, 用 “/” 和 “//” 表示除。

(2) 在 Python 3.0 中有两种除法: 真除 (/) 和 floor 除 (//)。

真除也称浮点除, 即无论是对两个整数相除, 还是对带小数的数相除, 结果都要保持小数部分。

floor 除是整除的一种, 其结果是真除结果向下舍入 (或称向 $-\infty$ 舍入) 得到的整数, 而不是简单地将小数部分截掉。

(3) 模操作符 % 也是执行 floor 整除的余数, 先计算出 floor 整除值, 再按被除数 - (整除值 * 除数) 计算而成。

代码 1-1 Python 3.0 中的 “/” “//” 与 “%” 的用法示例。

```
>>> 1 / 3
0.3333333333333333
>>> -1 / 3
-0.3333333333333333
>>> 1 // 3
0
>>> -1 // 3
-1
>>> 1 % 3
1
>>> -1 % 3
2
```

需要注意的是, 一般来说, 在一行中写的一个 Python 指令也称为一个语句。这个指令写完用回车操作就可以结束, 不需要其他语句结尾符号。但是, 若想在一行中写几个 Python 指令, 就需要用分号在指令之间进行分隔。例如, 代码 1-1 可以改写如下。

```
>>> 1 / 3; -1 / 3
0.3333333333333333
-0.3333333333333333
>>> 1 // 3; -1 // 3; 1 % 3; -1 % 3
0
-1
1
2
```

1.1.2 回显与 print()

在交互模式下输入一个表达式, 就会返回该表达式的值, 是不需要使用输出指令的输出操作的。严格地说, 这种 “输出” 称为 “回显” (echo)。回显使用简便, 但往往会受某些限制。在正式情况下, Python 输出使用的输出指令是内置函数 print()。

代码 1-2 回显与 print() 的用法比较。

```
>>> 1/7
```