

编程经典译丛

Graphical user interfaces for Python programs

Python and Tkinter Programming

# Python与Tkinter 编程



〔美〕John E. Grayson 著

陈文志 高垒 缪瑾 崔广仁 蒋涛 译

国防工业出版社

 MANNING

<http://www.ndip.com.cn>

编程经典译丛

# Python 与 Tkinter 编程

*Python and Tkinter Programming*

John E. Grayson 著

陈文志 高垒 缪瑾 崔广仁 蒋涛 译

国防工业出版社

·北京·

著作权合同登记号 图字:军-2001-011 号

图书在版编目 (CIP) 数据

Python 与 Tkinter 编程 / (美)格雷森 (Grayson, J.E.) 著;  
陈文志等译. —北京: 国防工业出版社, 2002.9

(编程经典译丛)

书名原文: Python and Tkinter Programming

ISBN 7-118-02781-2

I.P... II.①格...②陈... III. ①PYTHON 语言—  
程序设计②图形程序包, Tkinter IV.TP312

中国版本图书馆 CIP 数据核字 (2002) 第 004507 号

©2000 by Manning Publications Co. All rights reserved.

Published by arrangement with Manning Publications Co., USA.

本书中文简体版由 Manning Publications Co. 授予国防工业出版社  
(独家) 出版。版权所有, 侵权必究。

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

\*

开本 787×1092 1/16 印张 35 3/4 821 千字  
2002 年 9 月第 1 版 2002 年 9 月北京第 1 次印刷

印数: 1-3000 册 定价: 66.00 元

---

(本书如有印装错误, 我社负责调换)

# 前 言

---

我是在 1993 年加入位于 Rhode Island 的一个小公司时结识 Python 的。他们的主要产品是为 X/Motif 产生 C、C++、Ada 和 Python 编码的 GUI-builder。我的任务是为 X/Motif 和 Python 扩展面向对象界面。在这之前，我一直怀疑解释性语言的用途，因此，开始这工作时，我兴趣不大。两天之后，我上瘾了。使用 C 语言编码很复杂的界面，能够很容易地完成！不久，我更乐于选择使用 Python 而不是 C 来开发界面。

离开 Rhode Island 的公司之后，我开始使用 Tkinter 开发软件，这成了有名的 Python GUI。我说服了一家公司（我在那里做点合同工）来通过 Python 编码开发一个濒临超时和超出预算的大项目，项目很成功。四年以后，该公司有了很多 Python 程序员，他们的项目中相当的代码是通过 Tkinter 和 Python 来完成的。

正是这些经历，使得我编写这本书。早些时候，Python 可用文档资料很少。Tkinter Life Preserver 是第一本帮助人们把基本信息凑到一起的书。1997 年，Fredrik Lundh 在网上发布了不少优秀的 Python 类文档，这些为近两年 Python 编程起到极大的作用。我所见到的几个问题之一是：虽然有很多例程(伴随 Python 的发布不少)，但他们大部分内容太简洁，不能代表 Python 程序完整框架。当然，要把代码一个个连起来运转也不难，但是，当体系结构依赖于解释语言时，很容易得到一个在执行速度、美感、可维护性和可扩展性意义上差劲的产品。

因此，我写此书时别人问的第一个问题是：“我如何编个 XXX？”我会给他一堆我写过的源代码。像大部分专业程序员一样，他们将详细研究。我相信，从一个完整程序着手是学习某一特定编程语言的好方法，能够达到特定的目标。

我练习“空手道”的时候常去新泽西 Shukokai 世界总部，和 Sensei Shigeru Kimura 练习。Sensei Kimura 经常告诉我们：“我不能告诉你怎么做（某一特定技术）——你必须去‘偷’。”我的学习 Tkinter 的办法也是一样的。如果社区里的人解决了某个问题，我们必须从他们那里“偷”。现在，我不是建议侵犯版权，我的意思是，我们从能得到的任何资料学习知识。我希望你把本书的例子作为程序的起点。在一些地方，我已经使用了其他程序员的代码或想法。如果是这样，我已经向原作者致谢了。如果你使用了这里的一些代码，我希望你能够对原作者致谢。毕竟，我们“偷”来的东西比我们自己的更有

价值，这是从 Sensei 那里得来的。

我对 Douglas A. Young 的 *The X Window System: Programming and Applications with Xt* 一书的风格印象很深刻。它有些老了，但是有一个完整的代码，其中许多可以做新应用程序的模板。《Python 与 Tkinter 编程》一书有些类似的东西。你会发现一些很长的、别的编程书籍中很少见到的例子。我希望许多这里的例子可以作为解决特定问题的程序员的模板或源码。

展示整个完整例子而不是一个程序段将使你们熟悉到我的编程风格。在《Python 与 Tkinter 编程》的审阅过程中，一些审阅者就提出一些别的风格。在可能的地方，我都采纳了他们的意见。因此，书中的例子包含了多人的风格。我希望读者在自己的编程实践中能有所提高。

我希望你发现《Python 与 Tkinter 编程》一书有用。如果它在你编程时能为你省下一两个小时，你用来阅读的几个小时也是值得的。

## 特别致谢

---

《Python 与 Tkinter》一书是多人智慧的结晶。他们中的每人都为此书的出版尽心尽力。是他们的很多金玉良言，使此书更完美。

我要感谢技术审阅队伍：Fred L. Drake, Robin Friedrich, Alan Gauld, Bob Gibson, Lynn Grande Doug Hellmann, Gattrett G. Hodgson, Paul Kendrew, Andrew M.Kuchling, Cameron Laird Gregory A. Landrum, Ivan Van Ladingham, Burt Leavenworth, Ken McDonald, Frank McGeough, Robert Meegan, William Peloquin, Robert J. Roberts 和 Guido van Rossum。他们提供了详细的评论，使此书的内容、重点和准确性得到大大改进。

很多例子源代码是从其他人的代码中得来的。我要感谢这些代码编写者允许此书使用他们的代码。

Doug Hellman 在 Pmw 中写了个优秀的模块，GUIAppD.py，我改写后成为书中的 AppShell.py，书中还使用了很多例子。Doug 允许我使用其代码。如果你发现此代码可以在你的应用中使用的話，请感谢原作者。

Konrad Hinsien 写了 TkPlotCanvas.py，原本是在 Numpy，即数字操作最优的扩展模块下的。我将它改为不需要在 Numpy 下面，还增加了些图形功能。如果你发现能用到你工作中，感谢原作者。

第 8 章的树和点类是从 OpenChem 他们发布的公开码里得来的。你是否想看看他们组织将来发行的版本？因为本书中树控件的例子都来源于他们。

附录 B 把 Tk 手册作为 Tkinter 文档的起点。他们的版权所有，加利福尼亚的 Regents 和 Sun Microsystems 允许从中导出些结果，向他们致谢。

我要感谢 General Dynamics 的 Gordon Smith，他在负责的许多项目中很有信心地使用 Python 和 Tkinter，看到他们的使用，是导致我写此书的重要原因。我曾把其中一些草稿章节给他们的员工和内部学员测试，来解决他们编程任务。

其次，我要感谢 Manning 出版社把我的想法变为此书的诸君。我和发行人 Marjan Bace 进行过长时间的谈话，从而使我完成了这一复杂任务，使得本书对读者有用。Ted Kennedy 协调了整个复审过程，提出了许多建设性意见。Mary Piergics 和技术编辑 Kristen Black，排字员 Dottie Marsico 一道负责此书的出版。Doug Hellman 是一位很好的技术编辑，纠正了不少打印结果中的代码错误。

最后，我要感谢我妻子 Allison，我的孩子 Nina、Chris、Jeff 和 Alana，他们懂得牺牲一个伴侣和父亲也意味着赢得一个作家。

## 致读者

---

《Python 与 Tkinter 编程》旨在为知道或熟悉 Python，而又想在书中加入图形界面的程序员（可能是使用 Manning 的 *Quick Python* 为指导）而写的。因为《Python 与 Tkinter 编程》提供了大量的功能例子和详细代码注解，有其他编程经验但没有 Python 经验的程序员会发现此书为他们立即解决实际问题很有帮助。

此书还可以作为 Tcl/Tk 脚本程序员把 Tcl/Tk 转变为 Python 和 Tkinter 的好指南。然而，并不打算从哲学的角度来讨论是否适合——我存有偏爱。

## 关于作者

---

John Grayson 是一个图形用户界面(GUI)方面的顾问。他为一家美国通讯大公司从事应用设计多年，致力于设计革新性的界面，并引入了 Python 和面向对象编程方法取代传统的开发方式。另外他还为商业用户提供 Python 与 Tkinter 的应用。

他获得应用生物学学士学位和分子生物学博士学位，但此决非一个障碍（尤其因为他 90% 的论文是酶行为的计算机模型）。

在专攻用户界面之前，他是个操作系统专家，后来为 Stratus Computer 公司 Unix 开发支持方法。他在 Pacer 软件公司做了个 F77 编译器和 Unix 便携工具。他还是在英国和英国的 Prime Computer 公司的操作系统专家。

# 约 定

---

《Python 与 Tkinter 编程》一书中例子起到了极大的作用。不少编程书籍中例子的特征是短小简洁，把一两个要点很好地表述出来——但其实不起什么作用。本书中，例子可能适合你的应用或者正是需要的。很多例子的目的是单独运行，而不是交互式的。大部分例子包括标号以及对应的注解。例如：

```
def mouseDown (self, event):  
    self.currentobject = None  
    self.lastx = self.startx = self.canvas.canvasx (event.x)  
    self.lasty = self.starty = self.canvas.canvasy (event.y)  
    if not self.currentFunc:  
        self.selobj = self.canvas.find_closest (self.startx,  
                                                self.starty) [ 0 ]  
        self.canvas.itemconfig (self.selobj, width=2)  
  
    self.canvas.lift (self.selobj)
```

---

## 代码注解

❶ 鼠标按下方法反选任何当前已经选定对象。事件返回在屏幕中鼠标点击点的 x 和 y 坐标。画布控件的 `canvasx` 和 `canvasy` 方法…

❷ 如果绘画功能没有选中，则我们在选择模式下，并且在画布中对最近的对象定位，然后选中。该方法……

偶尔，我把部分代码变为黑体强调其在代码例子中的重要性。

有些地方代码跨越了多页，我已经把源代码散开，这样注释和源代码的距离可以保持比较近。任何例子中的标号都是连续的。



## 关于封面

---

本书封面是从 1805 年版的 Sylvain Maréchal 的四卷地区服饰习俗概述中得来的。这本书第一次于 1788 年在巴黎出版，是在巴黎革命前一年。单单标题就 30 多字。

Costumes Civils actuels de tous les peuples connus dessinés d'après nature gravées et coloriés, accompagnés d'une notice historique sur leurs coutumes, moeurs, religions, etc. , etc., redigés par M. Sylvain Maréchal

四卷包括图片的注解：“gravé à la manière noire par mixelle d'après Desrais et colorié”。很明显，能将雕刻师和插图者姓氏罗列下来，已经不错了，因为毕竟他们只是艺术家而已。而手绘这些东西的人则默默无闻。

这一丰富多彩的艺术品集生动地向我们说明了 200 年前的城市和乡村文化如何远离而去。服饰不断变化，不同地区也差异甚远。当年的多样性渐已消失。现在，很难区分一个大陆和另一个大陆之间的差异。也许，我们已经用文化的多样化换取了人生的复杂——确实，多么丰富和激动的技术环境啊。在无法区分一本计算机书和另一本计算机书的时候，Manning 通过在封面上使用 200 年前的地区生活的多样性来庆祝计算机事业的发明和兴起，带回到 Marechal 图片的生活中。想一想，Marechal 的世界是一个远远不同于我们的世界，在那里，人们愿意花时间去阅读 30 个字的标题。

## 作者在线

---

《Python 与 Tkinter 编程》的购买包括了可以进入一个私人 Internet 论坛，在那里，你可以对此书做评论，提出技术问题以及从作者及其他读者那里获得帮助。你通过你的浏览器浏览 <http://www.manning.com/grayson> 进入论坛。那里，你可以订阅论坛，此站点还提供信息，告诉你一旦注册之后如何加入论坛，可以得到什么样的帮助以及论坛行为准则。

书中所有的例子在 Manning 主页上都有，URL 为 [www.manning.com/grayson](http://www.manning.com/grayson)，包括了一个指向源代码的链接。

## 内 容 简 介

Tkinter 是解释性脚本语言 Python 的图形界面开发工具包，是一个清晰的、面向对象的图形界面开发工具，它可以让你快速地开发出可以运行在 Unix、Macintosh 和 Windows 等系统平台下的 Python 应用程序。

本书详细地介绍了 Python 语言的语法和函数及其图形界面的开发方法，重点介绍了在 Tkinter 可视化平台下的编程方法。

本书非常适合各个层次的 Python 程序员，同时也适合利用 Python 开发各种脚本的设计人员。

# 目 录

## 第1部分 基本概念

<b>第1章 Python</b> .....	3
1.1 Python 编程及特点概述.....	3
1.1.1 为什么是 Python.....	4
1.1.2 Python 能被用于何处.....	4
1.2 关键字数据类型：表、元组和字典.....	4
1.2.1 表.....	5
1.2.2 元组.....	6
1.2.3 字典.....	7
1.3 类.....	8
1.3.1 类是怎样描述对象的.....	8
1.3.2 定义类.....	8
1.3.3 Python 技巧.....	8
1.3.4 一个实例的初始化.....	9
1.3.5 方法.....	9
1.3.6 私有与公有变量和方法.....	9
1.3.7 继承性.....	10
1.3.8 多重继承性.....	10
1.3.9 混合类.....	10
<b>第2章 Tkinter</b> .....	11
2.1 Tkinter 模块.....	11
2.1.1 Tkinter 是什么.....	11
2.1.2 性能如何.....	11
2.1.3 如何运用 Tkinter.....	12
2.1.4 Tkinter 的特点.....	12
2.2 Tcl/Tk 对 Tkinter 的映像.....	13
2.3 Win32 和 Unix 下的 GUI.....	13
2.4 Tkinter 类的层次.....	14
2.5 Tkinter 控件的外观.....	15
<b>第3章 建立一个应用</b> .....	16
3.1 计算器范例：关键特点.....	18

3.2	计算器范例：源代码	19
3.3	检查应用结构	25
3.4	应用的扩展	26

## 第2部分 显示

<b>第4章</b>	<b>Tkinter 控件</b>	<b>29</b>
4.1	Tkinter 控件漫游	29
4.1.1	顶层	29
4.1.2	框架	30
4.1.3	标签	32
4.1.4	按钮	32
4.1.5	输入	34
4.1.6	单选按钮	34
4.1.7	复选按钮	35
4.1.8	主菜单	35
4.1.9	消息	38
4.1.10	文本	39
4.1.11	画布	40
4.1.12	滚动条	41
4.1.13	列表框	41
4.1.14	标尺	42
4.2	字体和颜色	43
4.2.1	字体描述符	43
4.2.2	X 视窗系统字体描述符	43
4.2.3	颜色	44
4.2.4	设置宽应用默认字体和颜色	44
4.3	Pmw 大控件漫游	45
4.3.1	关于框	45
4.3.2	浮动图	46
4.3.3	按钮框	47
4.3.4	组合框	47
4.3.5	组合对话框	49
4.3.6	计数器	49
4.3.7	计数对话框控件	50
4.3.8	对话框	51
4.3.9	输入域	51
4.3.10	组	52
4.3.11	标签控件	53

4.3.12	菜单条 .....	54
4.3.13	消息栏 .....	54
4.3.14	消息对话 .....	55
4.3.15	记事本 R .....	56
4.3.16	记事本 S .....	57
4.3.17	记事本 .....	57
4.3.18	选项菜单 .....	58
4.3.19	窗格控件 .....	59
4.3.20	提示对话框 .....	60
4.3.21	单选选项 .....	60
4.3.22	滚动画布 .....	61
4.3.23	滚动区域 .....	62
4.3.24	滚动框架 .....	62
4.3.25	滚动列表框 .....	63
4.3.26	滚动文本 .....	64
4.3.27	选项对话 .....	64
4.3.28	文本对话 .....	65
4.3.29	时间计数 .....	66
4.4	创建新的人控件 .....	66
4.4.1	大控件的描述 .....	66
4.4.2	选项 .....	67
4.4.3	制作大控件类 .....	67
<b>第 5 章</b>	<b>屏幕版面设计 .....</b>	<b>70</b>
5.1	版面设计的介绍 .....	70
5.2	打包器 .....	71
5.2.1	使用展开选项 .....	74
5.2.2	使用填充选项 .....	74
5.2.3	使用 padx 和 pady 选项 .....	75
5.2.4	使用锚选项 .....	76
5.2.5	使用分层性包装 .....	76
5.3	网格 .....	77
5.4	放置器 .....	82
5.5	小结 .....	85
<b>第 6 章</b>	<b>事件, 捆绑和回叫 .....</b>	<b>87</b>
6.1	事件驱动系统: 评论 .....	87
6.1.1	事件是什么 .....	87
6.1.2	事件的传播 .....	88
6.1.3	事件类型 .....	88
6.2	Tkinter 事件 .....	89

6.3	回叫	93
6.4	λ 表达	94
6.5	捆绑事件和回叫	95
6.5.1	捆绑方法	95
6.5.2	处理多重捆绑	97
6.6	定时器和背景程序	97
6.7	动态回叫管理者	98
6.8	使事件运作	98
6.8.1	捆绑动态数据与控制件	98
6.8.2	数据确认	102
6.8.3	格式化 (smart) 控件	107
6.9	小结	109
<b>第 7 章</b>	<b>使用类、控件和特殊控件</b>	<b>110</b>
7.1	创建发光二极管类	110
7.1.1	再试试	115
7.1.2	什么改变了	119
7.2	构件类库	119
7.2.1	将六边形螺帽加入类库	121
7.2.2	创建开关类	123
7.2.3	创建大控件	126
7.3	小结	129
<b>第 8 章</b>	<b>对话框和窗体</b>	<b>130</b>
8.1	对话框	130
8.1.1	标准对话框	131
8.1.2	数据输入对话框	132
8.1.3	简单表格	135
8.1.4	Tkinter 变量	140
8.2	标准窗体框架	144
8.3	数据字典	154
8.4	活页夹	160
8.5	浏览器	163
8.6	压缩程序	172
8.7	图像映射	180
8.8	小结	187
<b>第 9 章</b>	<b>面板与机器</b>	<b>188</b>
9.1	创建前端面板	188
9.2	模块性	189
9.3	实现前端面板	190
9.4	GIF、BMP 与图层	204

9.5	一个更完整的例子 .....	209
9.6	使用 POV-Ray 创建的虚拟设备 .....	222
9.6.1	现在看点完全不同的东西 .....	222
9.7	小结 .....	225
<b>第 10 章</b>	<b>绘制点和标尺 .....</b>	<b>226</b>
10.1	在画布上绘图 .....	226
10.1.1	移动画布对象 .....	231
10.2	一个更完整的绘图程序 .....	232
10.3	滚动画布 .....	240
10.4	标尺工具 .....	243
10.5	缩放画布对象 .....	246
10.6	一些已完成的小玩意 .....	251
10.7	快速绘制 .....	260
10.8	小结 .....	263
<b>第 11 章</b>	<b>图形和图表 .....</b>	<b>264</b>
11.1	简单图形 .....	264
11.2	图形控件 .....	266
11.2.1	添加条形图 .....	273
11.2.2	饼图 .....	276
11.3	三维图形 .....	279
11.4	带形记录图 .....	284
11.5	小结 .....	286
<b>第 12 章</b>	<b>导航 .....</b>	<b>287</b>
12.1	引言: 导航方法 .....	287
12.2	鼠标导航 .....	287
12.3	键盘导航: “无鼠标导航” .....	288
12.4	建立应用程序的导航 .....	288
12.5	图像映射 .....	291
12.6	小结 .....	291
<b>第 13 章</b>	<b>窗口管理器 .....</b>	<b>292</b>
13.1	什么是窗口管理器 .....	292
13.2	几何方法 .....	293
13.3	可见性方法 .....	293
13.4	图标方法 .....	294
13.5	协议方法 .....	294
13.6	其他 wm 方法 .....	295

## 第3部分 组合起来

<b>第14章 Python 扩展</b> .....	299
14.1 写 Python 扩展.....	299
14.2 建立 Python 扩展.....	301
14.2.1 在 Unix 下静态链接一个扩展.....	302
14.2.2 在 Windows 下静态链接一个扩展.....	302
14.2.3 在 Unix 下构建动态模块.....	303
14.2.4 在 Windows 下构建动态模块.....	303
14.2.5 安装动态模块.....	304
14.2.6 使用动态模块.....	304
14.3 在扩展中使用 Python API.....	304
14.4 在 C++中构建扩展.....	306
14.5 格式化字符串.....	306
14.6 引用计数.....	309
14.7 嵌入式的 Python.....	309
14.8 小结.....	313
<b>第15章 调试程序</b> .....	314
15.1 为什么使用 print 表达式.....	314
15.2 一个简单的例子.....	314
15.3 如何调试.....	317
15.4 Tkinter 开发器.....	318
15.5 pdb.....	319
15.6 IDLE.....	319
15.7 DDD.....	320
<b>第16章 设计有效的图形应用程序</b> .....	321
16.1 友好界面设计的元素.....	321
16.2 人为因素.....	324
16.2.1 选择字体.....	325
16.2.2 在 GUI 中使用颜色.....	326
16.2.3 尺寸的考虑.....	327
16.3 可选择的 GUI.....	327
16.4 小结.....	328
<b>第17章 性能编程</b> .....	329
17.1 每日加速.....	329
17.1.1 程序组织.....	329
17.1.2 使用 Python 优化器.....	330
17.1.3 检查代码.....	330



17.2 Tkinter 性能 .....	330
17.2.1 让它保持简短 .....	331
17.2.2 删减本地变量 .....	331
17.2.3 保持简单 .....	332
17.2.4 快速初始化 .....	332
17.2.5 节省事件 .....	332
17.3 Python 技术 .....	332
17.3.1 引用模块 .....	333
17.3.2 字符串链接 .....	333
17.3.3 正确使用嵌套的循环 .....	334
17.3.4 减少模块引用 .....	334
17.3.5 使用本地变量 .....	335
17.3.6 使用异常处理 .....	335
17.3.7 使用 map, filter 和 reduce .....	336
17.4 程序剖析 .....	337
17.5 Python 扩展 .....	340
17.6 小结 .....	340
<b>第 18 章 线程和异步技术 .....</b>	<b>341</b>
18.1 线程 .....	341
18.1.1 非图形用户界面下的线程 .....	341
18.1.2 图形用户界面下的线程 .....	344
18.2 “after” 处理 .....	348
18.3 小结 .....	351
<b>第 19 章 发布 Tkinter 程序 .....</b>	<b>353</b>
19.1 发布程序的一般问题 .....	353
19.2 发布 Unix 程序 .....	354
19.3 发布 Win32 程序 .....	354
19.4 Python 发布工具 .....	356

## 第 4 部分 附 录

附录 A Tk 到 Tkinter 映射 .....	361
附录 B Tkinter 参考资料 .....	392
附录 C Pmw 参考: Python megawidgets .....	485
附录 D 创建和安装 Python, Tkinter .....	539
附录 E 事件与键盘系统 .....	545
附录 F 光标 .....	549
附录 G 参考文献 .....	551