

Python GUI设计 tkinter菜鸟编程

洪锦魁◎著

tkinter 适用Python 3.x版

270个完整程序实例

跨平台窗口程序设计

tkinter Widget解说

tkinter.ttk Widget解说

Widget共通属性

Widget共通方法

变量类别Variable Classes

事件绑定Events and Binds

小算盘(Calculator)

文书编辑程序(Editor)

动画游戏设计

清华大学出版社

TP311.501
2015

Python GUI设计 tkinter菜鸟编程

洪锦魁◎著



清华大学出版社
北京

内 容 简 介

这是国内首先使用 tkinter 设计 GUI 的中文 Python 书籍之一。

本书主要讲解如何在窗口内使用 Python 的内部模块 tkinter 设计图形用户接口 (GUI) 程序, 使用户可以利用图形接口与计算机沟通。tkinter 模块是一个跨平台的窗口应用程序, 使用它设计的程序未来可以在 Windows、Mac、Linux 系统上执行。

Python 的 tkinter 模块内部有许多 Widget(可以翻译为控件或组件或部件), 有了这些 Widget 就可以设计出所有与图形接口相关的程序应用。本书介绍的 tkinter 模块 Widget 包括 Button(按钮)、Canvas(画布)、Checkbox(选项钮)、Entry(文本框)、Frame(框架)、Label(卷标)、LabelFrame(卷标框架)、Listbox(窗体)、Menu(菜单)、MenuItem(单选按钮)、Message(信息)、OptionMenu(下拉式窗体)、PanedWindow(面板)、RadioButton(选项钮)、Scale(滚动条值控制)、Scrollbar(滚动条)、Spinbox(可微调输入控件)、Text(文字区域)、Toplevel(上层窗口)。

此外, 本书还介绍了与 tkinter 模块设计应用有关的变量类别 (Variable Classes) 与事件绑定 (Events and Binds) 概念。

为了详细讲解 GUI 设计, 本书共使用了约 270 个程序实例, 详细解析各种 Widget 的用法, 同时也将应用扩充到设计文字编辑程序、计算器、动画与相关的游戏设计中。相信读者学完本书可以轻松将 GUI 知识应用到未来职场, 成为一位称职的软件工程师, 并成为 Python 领域的高手。

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 GUI设计: tkinter菜鸟编程 / 洪锦魁著. —北京: 清华大学出版社, 2019
ISBN 978-7-302-53064-0

I. ①P… II. ①洪… 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 (2019) 第 098354 号

责任编辑: 杨迪娜 薛 阳

封面设计: 杨玉兰

责任校对: 胡伟民

责任印制: 李红英

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

印 装 者: 三河市铭诚印务有限公司

经 销: 全国新华书店

开 本: 170mm×240mm 印 张: 20 字 数: 596 千字

版 次: 2019 年 8 月第 1 版 印 次: 2019 年 8 月第 1 次印刷

定 价: 79.00 元

产品编号: 081902-01



作者简介

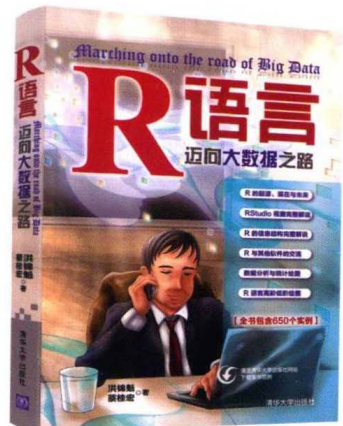
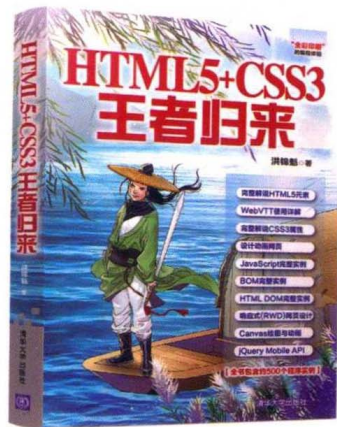
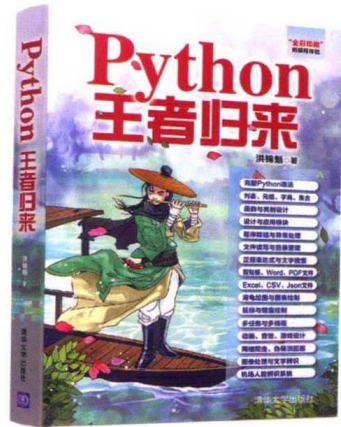
洪锦魁 中国台湾省计算机专家，IT知名作者。

近年来作品：

- ◆ Java王者归来
- ◆ Python王者归来
- ◆ HTML5 + CSS3王者归来
- ◆ R语言迈向大数据之路

其著作特色：

所有程序语法会依特性分类，同时以实用的程序范例进行解说，可以让读者事半功倍地轻松掌握相关知识。



序

本书是一本使用 tkinter 进行 Python GUI 设计的中文图书。

作者在 2017 年 12 月出版了《Python 入门迈向高手之路王者归来》。该书从上市到现在，连续几个月皆是台湾地区 Python 领域最畅销的书籍。该书约 820 页，虽然是目前 Python 图书讲解内容较丰富、应用较广泛的图书，但受限于篇幅，作者深知该书仍无法涵盖所有 Python 的应用，特别是在 GUI 设计部分只是粗浅讲解。

在 Python 应用程序内附有 tkinter 模块。这个模块主要用于设计用户图形接口 (Graphical User Interface, GUI)，也可以用于设计跨平台的窗口应用程序。程序设计人员可以使用此模块的控件 (Widget) 设计图形接口让用户与计算机沟通。tkinter 模块简单好用，但是目前却少有书籍对这个模块做过完整的功能介绍，这也是作者决定撰写本书的动力。

本书基本上不对 Python 语法进行介绍，所以读者需要有一定的 Python 知识基础才适合阅读本书，如果没有 Python 基础，建议先阅读作者所著下列两本书之一，建立起完整的 Python 知识框架。

《Python 零基础最强入门之路王者归来》

《Python 入门迈向高手之路王者归来》

本书将通过约 270 个程序实例讲解下列知识。

- (1) Python tkinter Widget ;
- (2) Python tkinter.ttk Widget ;
- (3) Widget 常用属性 ;
- (4) Widget 常用方法 ;
- (5) 变量类别 ;
- (6) 事件与绑定 ;
- (7) 计算器设计 ;

(8) 文本编辑程序设计；

(9) 动画游戏设计。

作者曾编写过许多计算机书籍。本书沿袭作者以往著作的特色，程序实例丰富，相信读者只要遵循本书的学习路线，必定可以在最短时间内精通窗口程序设计。本书内容虽力求完美，但是书中疏漏与不足之处在所难免，希望读者不吝指正。

目 录

第 1 章 基本概念

- 1-1 认识 GUI 和 tkinter 2
- 1-2 建立窗口 3
- 1-3 窗口属性的设置 3
- 1-4 窗口位置的设置 5
- 1-5 认识 tkinter 的 Widget 7
 - 1-5-1 tkinter 的 Widget 7
 - 1-5-2 加强版的 tkinter 模块 8
- 1-6 Widget 的共同属性 9
- 1-7 Widget 的共同方法 9

第 2 章 标签 Label

- 2-1 标签 Label 的基本应用 12
- 2-2 Widget 共同属性 Color 14
- 2-3 Widget 的共同属性 Dimensions 15
- 2-4 Widget 的共同属性 Anchor 15
- 2-5 Label 文字输出换行位置
wraplength 17
- 2-6 Widget 的共同属性 Font 17
- 2-7 Label 的 justify 参数 18
- 2-8 Widget 的共同属性
Bitmaps 20
- 2-9 compound 参数 20
- 2-10 Widget 的共同属性 relief 22
- 2-11 标签文字与标签区间的间距
padx/pady 22
- 2-12 图像 PhotoImage 23
- 2-13 Widget 的共同方法
config() 27
- 2-14 Widget 的共同属性
Cursors 28
- 2-15 Widget 的共同方法 keys() 29
- 2-16 分隔线 Separator 30

第 3 章 窗口控件配置管理员

- 3-1 Widget Layout Manager 33
- 3-2 pack 方法 33
 - 3-2-1 side 参数 33
 - 3-2-2 padx/pady 参数 37
 - 3-2-3 ipadx/ipady 参数 40
 - 3-2-4 anchor 参数 41
 - 3-2-5 fill 参数 42
 - 3-2-6 expand 参数 45
 - 3-2-7 pack 的方法 47
- 3-3 grid 方法 48
 - 3-3-1 row 和 column 48
 - 3-3-2 colspan 参数 50
 - 3-3-3 rowspan 参数 51
 - 3-3-4 padx 和 pady 参数 52
 - 3-3-5 sticky 参数 53
 - 3-3-6 grid 方法的应用 55
 - 3-3-7 rowconfigure() 和
columnconfigure() 56
- 3-4 place 方法 58
 - 3-4-1 x/y 参数 58
 - 3-4-2 width/height 参数 59
 - 3-4-3 relx/rely 参数与 relwidth/relheight
参数 60
- 3-5 Widget 控件位置总结 62

第 4 章 功能按钮 Button

- 4-1 功能按钮基本概念 64
- 4-2 使用 Lambda 表达式 68
- 4-3 建立含图像的功能按钮 69
- 4-4 简易计算器按钮布局的应用 70
- 4-5 设计鼠标光标在功能
按钮上的形状 72

第 5 章 文本框 Entry

- 5-1 文本框 Entry 的基本概念 74
- 5-2 使用 show 参数隐藏输入的字符... 75
- 5-3 Entry 的 get() 方法 77
- 5-4 Entry 的 insert() 方法 79
- 5-5 Entry 的 delete() 方法 80
- 5-6 计算数学表达式使用 eval() 81

第 6 章 变量类别

- 6-1 变量类别的基本概念 84
- 6-2 get() 与 set() 84
- 6-3 追踪 trace() 使用模式 w 86
- 6-4 追踪 trace() 使用模式 r 88
- 6-5 trace() 方法调用的
callback 方法参数 89
- 6-6 计算器的设计 90

第 7 章 选项按钮与复选框

- 7-1 Radiobutton 选项按钮 94
 - 7-1-1 选项按钮的基本概念 94
 - 7-1-2 将字典应用在选项按钮上 97
 - 7-1-3 盒子选项按钮 98
 - 7-1-4 建立含图像的选项按钮 99
- 7-2 Checkbutton 复选框 101
 - 7-2-1 复选框的基本概念 101
- 7-3 简单编辑程序的应用 105

第 8 章 容器控件

- 8-1 框架 Frame 108
 - 8-1-1 框架的基本概念 108
 - 8-1-2 在框架内创建 Widget
控件 110
 - 8-1-3 活用 relief 属性 110
 - 8-1-4 在含 raised 属性的框架内
创建复选框 111
 - 8-1-5 额外对 relief 属性的支持 112
- 8-2 标签框架 LabelFrame 113
 - 8-2-1 标签框架的基本概念 113
 - 8-2-2 将标签框架应用于复选框 115
- 8-3 顶层窗口 Toplevel 116
 - 8-3-1 Toplevel 窗口的基本概念 116

- 8-3-2 使用 Toplevel 窗口仿真对话框 117

第 9 章 与数字有关的 Widget

- 9-1 Scale 的数值输入控制 120
 - 9-1-1 Scale 的基本概念 120
 - 9-1-2 取得与设置 Scale 的尺度值 122
 - 9-1-3 使用 Scale 设置窗口背景颜色 123
 - 9-1-4 askcolor() 方法 125
 - 9-1-5 容器的应用 126
- 9-2 Spinbox 控件 127
 - 9-2-1 Spinbox 控件基本概念 127
 - 9-2-2 get() 方法的应用 129
 - 9-2-3 以序列存储 Spinbox 的数值
数据 130
 - 9-2-4 非数值数据 131

第 10 章 Message 与 Messagebox

- 10-1 Message 133
 - 10-1-1 Message 的基本概念 133
 - 10-1-2 使用字符串变量处理 text
参数 134
- 10-2 Messagebox 135

第 11 章 事件和绑定

- 11-1 Widget 的 command
参数 141
- 11-2 事件绑定 142
 - 11-2-1 鼠标绑定的基本应用 145
 - 11-2-2 键盘绑定的基本应用 147
 - 11-2-3 键盘与鼠标事件绑定的陷阱 148
- 11-3 取消绑定 149
- 11-4 一个事件绑定多个事件处理
程序 151
- 11-5 Protocols 152

第 12 章 列表框 Listbox 与滚动条
Scrollbar

- 12-1 建立列表框 154
- 12-2 建立列表框项目 insert() 155
- 12-3 Listbox 的基本操作 159
 - 12-3-1 列出列表框的选项数量
size() 159

- 12-3-2 选取特定索引项 selection_
 - set() 160
- 12-3-3 删除特定索引项 delete() 161
- 12-3-4 传回指定的索引项 get() 163
- 12-3-5 传回所选取项目的索引
 - curselection() 164
- 12-3-6 检查指定索引项是否被选取
 - selection_includes() 165
- 12-4 Listbox 与事件绑定 165
 - 12-4-1 虚拟绑定应用于单选 165
 - 12-4-2 虚拟绑定应用于多选 167
- 12-5 活用加入和删除项目 168
- 12-6 Listbox 项目的排序 170
- 12-7 拖曳 Listbox 中的项目 171
- 12-8 滚动条的设计 173
- 第 13 章 OptionMenu 与 Combobox**
 - 13-1 下拉式列表 OptionMenu 177
 - 13-1-1 建立基本的 OptionMenu 177
 - 13-1-2 使用元组建立列表项目 178
 - 13-1-3 建立默认选项 set() 178
 - 13-1-4 获得选项内容 get() 179
 - 13-2 组合框 Combobox 180
 - 13-2-1 建立 Combobox 180
 - 13-2-2 设置默认选项 current() 181
 - 13-2-3 获得目前选项 get() 182
 - 13-2-4 绑定 Combobox 183
- 第 14 章 容器 PanedWindow 和 Notebook**
 - 14-1 PanedWindow 186
 - 14-1-1 PanedWindow 基本概念 186
 - 14-1-2 插入子控件 add() 186
 - 14-1-3 建立 LabelFrame 当作子对象 .. 187
 - 14-1-4 tkinter.ttk 模块的 weight
 - 参数 188
 - 14-1-5 在 PanedWindow 内插入不同
 - 控件 190
 - 14-2 Notebook 191
 - 14-2-1 Notebook 基本概念 191
 - 14-2-2 绑定选项卡与子控件内容 192
- 第 15 章 进度条 Progressbar**
 - 15-1 Progressbar 的基本应用 195
 - 15-2 Progressbar 动画设计 196
 - 15-3 Progressbar 的方法 start() /
 - step() / stop() 198
 - 15-4 indeterminate 模式 200
- 第 16 章 菜单 Menu 和工具栏 Toolbars**
 - 16-1 菜单 Menu 设计的基本概念 202
 - 16-2 tearoff 参数 204
 - 16-3 菜单列表间加上分隔线 205
 - 16-4 建立多个菜单的应用 206
 - 16-5 Alt 快捷键 208
 - 16-6 Ctrl+ 快捷键 210
 - 16-7 创建子菜单 211
 - 16-8 建立弹出式菜单 212
 - 16-9 add_checkbutton() 213
 - 16-10 创建工具栏 Toolbar 215
- 第 17 章 文字区域 Text**
 - 17-1 文字区域 Text 的基本概念 218
 - 17-2 插入文字 insert() 220
 - 17-3 Text 加上滚动条 Scrollbar
 - 设计 221
 - 17-4 字形 224
 - 17-4-1 family 224
 - 17-4-2 weight 225
 - 17-4-3 size 227
 - 17-5 选取文字 228
 - 17-6 认识 Text 的索引 229
 - 17-7 建立书签 232
 - 17-8 标签 233
 - 17-9 Cut/Copy/Paste 功能 236
 - 17-10 复原与重复 239
 - 17-11 查找文字 241
 - 17-12 拼写检查 243

17-13	存储 Text 控件内容	244	19-1-4	绘制圆弧 create_arc()	282
17-14	新建文档	248	19-1-5	绘制圆或椭圆 create_oval()	284
17-15	打开文档	249	19-1-6	绘制多边形 create_polygon()	285
17-16	默认含滚动条的 ScrolledText 控件	251	19-1-7	输出文字 create_text()	286
17-17	插入图像	252	19-1-8	更改画布背景颜色	286
第 18 章 Treeview			19-1-9	插入图像 create_image()	287
18-1	Treeview 的基本概念	254	19-2	鼠标拖曳绘制线条	288
18-2	格式化 Treeview 栏位内容	258	19-3	动画设计	289
18-3	建立不同颜色的行内容	260	19-3-1	基本动画	289
18-4	建立层级式的 Treeview	262	19-3-2	多个球移动的设计	290
18-5	插入图像	263	19-3-3	将随机数应用于多个球体的 移动	291
18-6	Selection 选项发生与事件 触发	264	19-3-4	消息绑定	292
18-7	删除项目	266	19-4	反弹球游戏设计	293
18-8	插入项目	267	19-4-1	设计球往下移动	293
18-9	双击某个项目	270	19-4-2	设计让球上下反弹	295
18-10	Treeview 绑定滚动条	271	19-4-3	设计让球在画布四面反弹	296
18-11	排序	272	19-4-4	建立球拍	297
第 19 章 Canvas			19-4-5	设计球拍移动	298
19-1	绘图功能	277	19-4-6	球拍与球碰撞的处理	299
19-1-1	建立画布	277	19-4-7	完整的游戏	301
19-1-2	绘制线条 create_line()	277	附录 A	RGB 色彩表	304
19-1-3	绘制矩形 create_rectangle()	281	附录 B	函数或方法索引表	310



第 1 章

基本概念

本章摘要

- 1-1 认识 GUI 和 tkinter
- 1-2 建立窗口
- 1-3 窗口属性的设置
- 1-4 窗口位置的设置
- 1-5 认识 tkinter 的 Widget
- 1-6 Widget 的共同属性
- 1-7 Widget 的共同方法

1-1 认识 GUI 和 tkinter

GUI 英文全称是 **Graphical User Interface**，中文为**图形用户接口**。早期人与计算机之间的沟通是文字形式的沟通，例如，早期的 DOS 操作系统、Windows 的命令提示符窗口、Linux 系统，等等。本书主要说明如何设计图形用户接口，以让用户可以与计算机进行沟通，并介绍使用 Python 内附的 tkinter 模块设计相关程序。

tkinter 是一个**开放源码**的图形接口开发工具，原来是用 TCL(Tool Command Language, 工具命令语言)编写的 GUI 函数库，最初发展是从 1991 年开始，具有跨平台的特性，可以在 Linux、Windows、Mac OS 等操作系统上执行。这个 tkinter 工具提供许多图形接口，例如，标签 (Label)、菜单 (Menu)、按钮 (Button) 等。目前，这个 tkinter 工具已经移植到 Python 语言，属于 Python 语言内建的模块，在 Python 2 版本中该模块名称是 tkinter，在 Python 3 版本中该模块被称为 tkinter 模块。

在安装 Python 时，就已经同时安装此模块了，在使用前只需导入此模块即可，如下所示。

```
from tkinter import *
```

之后我们就可以使用此模块的工具设计多样化的 GUI 程序了。软件版本变化很快，在正式进入 Python 的 tkinter 模块前首先介绍如何了解自己的 tkinter 版本。

程序实例 ch1_0.py: 列出 tkinter 版本。

```
1 # ch1_0.py
2 import tkinter
3
4 print(tkinter.TkVersion)
```

执行结果

```
===== RESTART: D:/PythonGUI/ch1_0.py =====
8.6
>>>
```

一般 8.5 以后的版本功能比较健全。

1-2 建立窗口

可以使用下列方法建立窗口。

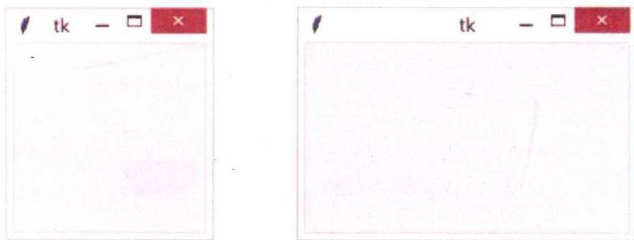
```
root = Tk()           # root 是自定义的 Tk 对象名称，也可以取其他名称
root.mainloop()      # 放在程序最后一行
```

通常将使用 Tk() 方法建立的窗口称为**根窗口**，之后可以在此根窗口中建立许多**控件**，也可以在此根窗口中建立上层窗口。本例中笔者用 root 当作对象名称，读者也可以自行取其他名称。上述 mainloop() 方法可以让程序继续执行，同时进入等待与处理窗口事件，单击窗口右上方的“关闭”按钮，此程序才会结束。

程序实例 ch1_1.py: 建立空白窗口，窗口默认名称是 tk。

```
1 # ch1_1.py
2 from tkinter import *
3
4 root = Tk()
5 root.mainloop()
```

执行结果 下方右图是更改窗口大小后的结果。



上述左边窗口大小是默认大小，当窗口出现后，可以拖曳移动窗口或更改窗口大小。

注 在 GUI 程序设计中，有时候也将上述所建立的窗口 (window) 称为**容器** (container)。

1-3 窗口属性的设置

下列是与窗口相关的方法。

方法	说明
<code>title()</code>	可以设置窗口的标题
<code>geometry("widthxheight+x+y")</code>	设置窗口宽 <code>width</code> 与高 <code>height</code> ，单位是像素 <code>pixel</code> ，设定窗口位置
<code>maxsize(width,height)</code>	拖曳时可以设置窗口最大的宽 (<code>width</code>) 与高 (<code>height</code>)
<code>minsize(width,height)</code>	拖曳时可以设置窗口最小的宽 (<code>width</code>) 与高 (<code>height</code>)
<code>configure(bg="color")</code>	设置窗口的背景颜色
<code>resizable(True,True)</code>	可设置是否更改窗口大小，第一个参数是宽，第二个参数是高，如果要固定窗口宽与高，可以使用 <code>resizable(0,0)</code>
<code>state("zoomed")</code>	最大化窗口
<code>iconify()</code>	最小化窗口
<code>iconbitmap("xx.ico")</code>	更改默认窗口图标

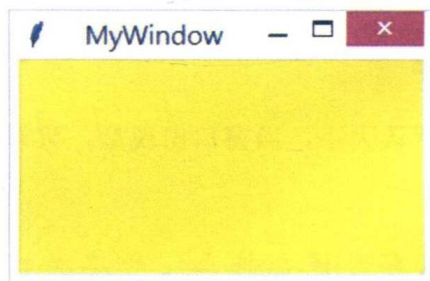
程序实例 `ch1_2.py`: 设置窗口标题为 `MyWindow`，同时设置宽是 300，高是 160。

```

1 # ch1_2.py
2 from tkinter import *
3
4 root = Tk()
5 root.title("MyWindow") # 窗口标题
6 root.geometry("300x160") # 窗口大小
7 root.configure(bg='yellow') # 窗口背景颜色
8 root.mainloop()

```

执行结果



上述第 7 行笔者使用 `bg` 设置了窗口背景颜色，相关颜色名称可以参考附录 A。除了可以使用名称直接设置色彩，还可以使用十六进制方式设置色彩 `RGB`，其中每个色彩用两个十六进制数字表示。从附录 A 的色彩表也可以看到 `RGB` 数值所代表的颜色。

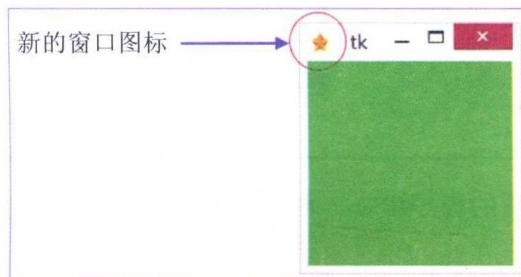
程序实例 `ch1_3.py`: 使用 `mystar.ico` 更改系统默认的图标，同时使用另一种更改背景颜色的方法。

```

1 # ch1_3.py
2 from tkinter import *
3
4 root = Tk()
5 root.configure(bg='#00ff00') # 窗口背景颜色
6 root.iconbitmap("mystar.ico") # 更改图标
7 root.mainloop()

```

执行结果



1-4 窗口位置的设置

`geometry()` 方法除了可以设置窗口的大小，也可以设置窗口的位置，此时它的语法格式如下。

```
geometry(widthxheight+x+y)
```

上述 `widthxheight` 已说明是窗口的宽和高，`width` 与 `height` 用 `x` 分隔。“+`x`”表示 `x` 是窗口左边距离屏幕左边的距离，如果是“-`x`”，则表示 `x` 是窗口右边距离屏幕右边的距离。“+`y`”表示 `y` 是窗口上边距离屏幕上边的距离，如果是“-`y`”则表示 `y` 是窗口下边距离屏幕下边的距离。

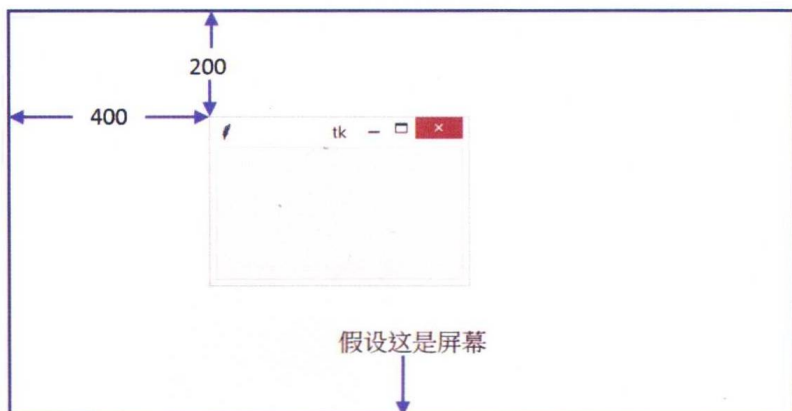
程序实例 `ch1_4.py`: 建立一个 `300×160` 大小的窗口，此窗口左上角坐标是 `(400,200)`。

```

1 # ch1_4.py
2 from tkinter import *
3
4 root = Tk()
5 root.geometry("300x160+400+200") # 距离屏幕左上角(400,200)
6 root.mainloop()

```


执行结果



Python 是一个很灵活的程序语言，可参考下列实例。

程序实例 ch1_5.py: 重新设计 geometry() 方法，读者可以自行判断使用哪一种方式建立窗口与设置窗口位置。

```

1 # ch1_5.py
2 from tkinter import *
3
4 root = Tk()
5 w = 300      # 窗口宽
6 h = 160     # 窗口高
7 x = 400     # 窗口左上角x轴位置
8 y = 200     # 窗口左上角Y轴位置
9 root.geometry("%dx%d+%d+%d" % (w,h,x,y))
10 root.mainloop()

```

执行结果 与 ch1_4.py 相同。

在 tkinter 模块中可以使用下列方法获得屏幕的宽度和高度。

```

wininfo_screenwidth( )      # 屏幕宽度
wininfo_screenheight( )    # 屏幕高度

```

程序实例 ch1_6.py: 设计窗口同时将此窗口放在屏幕中央。

```

1 # ch1_6.py
2 from tkinter import *
3
4 root = Tk()
5 screenWidth = root.wininfo_screenwidth() # 屏幕宽度
6 screenHeight = root.wininfo_screenheight() # 屏幕高度
7 w = 300 # 窗口宽
8 h = 160 # 窗口高
9 x = (screenWidth - w) / 2 # 窗口左上角x轴位置
10 y = (screenHeight - h) / 2 # 窗口左上角Y轴位置
11 root.geometry("%dx%d+%d+%d" % (w,h,x,y))
12 root.mainloop()

```