第16章形用户界面编程

图形用户界面(Graphical User Interface,简称GUI),又称图形用户接口,它是一套工具集, 可以结合各种计算机语言进行界面开发。Python作为一门优秀的语言，它自诞生之日起，已经 结合了很多优秀的GU［工具集编写出了许多应用程序，接下来，本章将针对Pythoii中的图形 用户界面编程进行讲解。

16o 1 Tkanter

Tkinter是Python标准的GUI库,它是基于Tk工具包接口。起初,Tk是基于TCL语言设 计的，后来才被移植到其他脚本语言中。Tk和Tkin翊可以在大多数的UNIX平台下使用，同 样也可以应用于Windows和Mac OS系统。 "

16JJ MilTkooter ..

Tkiil也•基于的Tk技术最初是为TCL语言所设计的。由于Tk技术非常灵活，很快被应用 在很多计算机语言上,包括 Python ( Tkinter)、Perl (Perl/Tk)、Tuby(Ruby/TK)等。

Tkinter是Python默认的GUI库，它相对于其他Python GUI库简单易学，我们可以使用少量 的代码，编写出功能强大的GUI界面，同时，Tkin宜的跨平台特性，使得它深受广大公司的喜爱。

由于Tkin切是内置在Pythoii中的*，*所以我们不需要单独安装Tkinter,只需要在使用的时候, 导入tkinter模块即可。示例如下:

from tkinter import \*

这里需要说明的是，虽然Tkinte『很好使用，但如果要开发一些大型的应用° Tkinte『提供 的功能还是太少,有些功能需要我们自己去实现,为此,后续出现了 wxPython、PyQt这些第 三方的库，大家可以在实际的GUI编程中，选择性使用适合自己的工具包。

16J.2

在深入学习Tkin坨『之前，我们先开发一个简单的Tkintw程序，通过这个程序来认识如何 使用Tkinter进行用户界面开发。

化©A Tkmter模块

有两种方式可以导入Tkimer,具体如下：

方式一：

import tkinter

方式二

from tkinter import \*

更多情况下，使用第2种方式导入Tkinter模块。

2.创建根窗口

图形用户界面程序都需要一个根窗口(或者叫主窗口)，它就好比绘画需要画纸一样，这 个根窗口就是承载图形用户界面各个组件的一个对象，一个应用程序只能有一个根窗口。TK 类提供了创建根窗口的构造函数，示例如下《

root=Tk()

3.进入消息循环

不停地告诉GM (geometry manager)有一个组件产生了，具体代*，矿\*.".壬坤*

码如下： 「 1|

root.mainloop() 11 \

完整的代码如下所示。 I I

[from tkinter import \* \ [](#bookmark1401" \o "Current Document)

root=Tk() j I

root oinainloop () \ ]

程序运行后的结果如图16』所示。

图16」运行结果图 〔沒小提示：

如果我们希望在创建窗口时，对窗口的大小进行设置，那么可以调用该窗口对象的 geometry方法来实现，示例如下:

root。geometry(v200x200 + 100+200

上述代码中，"200x200+100+200"就是设置窗口大小的数据，它对应的格式是“宽x高+ 水平偏移量+竖直偏移量”。

16o2 Tkinter

16-2J

Tkinter提供了许多组件，其中最核心组件有15个，具体如表164所示。

表16-1 Tkinter核心组件

|  |  |
| --- | --- |
| 组 件 | 描 述 |
| Button | 按钮组件，在应用程序中显示按钮 |
| Canvas | 画布组件，用于显示图形元素 |
| Checkbutton | 多选框组件，用于在程序中提供多项选择框 |

续表

|  |  |  |
| --- | --- | --- |
| 组 件 | 描 述 | |
| Entry | 输入组件，用于显示简单的文本内容 | |
| Frame | 框架组件，用于在屏幕上显示一个矩形区域 | |
| Label | 标签组件，可以显示文本和位图 | |
| Listbox | 列表框组件，显示一个字符串列表给用户 | |
| Menubutton | 菜单按钮组件，显示菜单项 | |
| Menu | 菜单组件显示菜单项，下拉菜单和弹出菜单 | |
| Message | 消息组件，用来显示多行文本 | |
| Radiobutton | 单选按钮组件，显示一个单选的按钮状态 | |
| Scale | 范围组件，显示一个数值刻度 | |
| Scrollbar | 滚动条组件，当内容超过可化区域时使用 | |
| Text | 文本组件，显示多行文本 | |
| Toplevel | 容器组件，作用和Frame比较类似 | |
| Spinbox | 输入组件，与Entry类似，但是可以指定输入范围值 | |
| PanedWindow | 窗口布局管理的插件，可以包含一个或多个子组件 | |
| LabelFrame | 一个简单的容器组件，常用于复杂的窗口布局 | |
| tkMessageBox | 用于显示应用程序的消息框 | |
| 下面,我们在第一个Tkinter程序的基础上,添加一个Label组件,具体代码如下:  from tkinter import \*  root=Tk() label^Label (root, text=?学 Python,来黑马程序员')  label.pack() root »mainloop ()  上述代码中,第3行代码创建了一个Label组件,其中,text是组件属性,表示的是Label | | |
| 组件的文本信，亂第4行代码使用pack将label组件进行布局, 也就是将Label组件布局到窗口中。在学习布局管理器之前, 大家只要知道pack是一个布局管理器,用于将组件包容到 | | 「IIIIIIW |
| 窗口中即可。  程序的运行效果如图16・2所示。 | | . 图16.2运行结果 |

16.2.2 ssawti

标准属性也就是所有组件的共同属性，如大小、字体和颜色等等。

1. 缰俸宾那(高虞植蠹廈)

默认情况下，组件的大小由组件内容决定，可以通过Width和height设置组件的宽度和高度。 如果设置的尺寸为整数歹那么该组件的大小是以像素为单位，或者我们也可以通过包含数字单 位的字符串来设置。

1. 缱俸颜笆

有两种方式可以指定组件的颜色：

(1 )可以使用一个字符串，指定红色，绿色和蓝色十六进制数字的比例。例如# FFF” 是白色的，“# 000000"是黑色的，"# OOOffiDOO"是纯绿色，和“# 00FFFF"是纯青色(绿 加蓝)。

(2)还可以使用任何本地定义的标准颜色名称。颜色“白色”"黑色"“红色” “绿色”“蓝 色” “青色” “黄色” “洋红”将始终可用。

可以使用三元组设置属性font的值，从而改变组件的字体，示例如下： label = Label (framer text= , ，黑马程序员! z

font=('italic \* , 12, 1 italic underline'))

上述代码中，使用font设置字体时，第一个元素表示的是字体， 第二个元素指的是字体大小,第三个元素的字体的附加信息。

MW N NE

4-镯点

锚点是用来定义文本的相对位置参考点，anchor属性可以设置 组件的停靠位置，即所在区域的方位，包括东西南北等。常见的锚 属性可以使用的常数及对应的方位如图16-3所示。

W CENTER E：

SVV S SE

鼠

mlief属性用于设置组件的浮雕式，它的常量值如下所示。 图16-3锚点常量对应方位

® FLATo

* RAISED o ，
* SUNKEN o

© GROOVEo

° RIDGEo

* SOLID o

下面是一排按钮的样式，通过设置不同常量值，这些按钮所呈现的样式是不同的，具体 如下：

from tkinter import top =Tk()

Button(topz text =°FLATn 7 relief=FLAT)

button one =

button\_two = button\_three

Button(top, text =nRAISEDnz relief=RAISED)

=Button(top, text -"SUNKEN", relief^SUNKEN)

button\_four = Button(top, text =°GROOVE relief=GROOVE)

button\_five = Button(top, text =nRIDGE", relief=RIDGE)

button\_one.pack()

button\_\_two。pack () button\_three.pack()

button\_four opack()

button\_five。pack () top oinainloop ()

效果如图16・4所示。

&位圏

bitmap属性设置显示的位图，你会使用这个属性显示一个位 图。有以下类型的可用位图：

* error o
* gray?5o

® gray50o

0 gray25o

° grayl2o

° hourglass o

® infbo

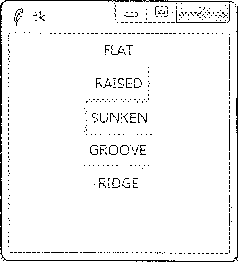
* questhead o

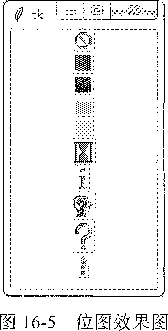
图16.4组件的样式

* question o
* warning o

下面看一个例子，具体如下:

from tkinter

import \*



button\_eight = Button(topz button\_nine = Button(topr

relief=RAISED,\bitmap="warning")

top = Tk()

button\_one = Button(topz text =nerror"f relief=RAISEDz\bitmap=nerror") button\_two = Button(top, text ="gray75"r relief=RAISED,\bitmap="gray7 5n) button\_three = Button(top^ text =ngray75"z relief=RA.ISED7\bitmap=ngray75") button\_four = Button (topf text =ngray25n f relief=RAISED^\bitmap="gray25") button\_five = Button(topf text =ngrayl2° r relief=RAISEDz\bitmap="grayl2") button\_six = Button(top7 text =nhourglass"r relief=RAISEDz\bitmap=nhourglass") button\_seven = Button (topz text ="info", relief=RAISEDz\bitmap=°infon)

text =nquesthead°z relief=RAISEDf\bitmap=nquesthead") text =°question"z relief=RAISEDf\bitmap="question")

button\_ten = Button(top, text =nwarningn7 button\_one。pack() button\_two.pack() button\_three.pack() button\_four.pack() button\_five。pack () button\_six.pack() button\_seven.pack() button\_eight.pack() button\_nine ^pack() button\_ten.pack() top oinainloop ()

运行结果如图16^5所示。

7.雍标

CWSCK属性设置鼠标经过组件时光标的形状。根据您不同的操作系统，鼠标的形状可能会 有所不同，cursor可以取的值如表16-2所示。

表16-2 cursor可以取的值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| "arrow" | "circle" | "clock" | "cross" | "dotbox1' | "exchange" |
| "fleur" | "heart" | "mann | "mouse" | ',pirate" | "plus" |
| 吸huttie" | °sizingn | "spider" | "spraycan" | ''star" | "target0 |
| "tcross" | "trek" | "watch" |  |  |  |

下面以其中的几个值为例，看一下效果，具体如下:

from tkinter import \*

top = Tk()

button\_one = Button(top, text =°circlen F relief=RAISEDz cursor="circle") button\_two = Button (topz text ="plus° f relief=RAISEDZcursor="plus°) button\_three = Button(topz text ="clock"t relief=RAISEDzcursor="clock") button\_one.pack() button\_two.pack() button\_three.pack() top.mainloop()

效果如图16・6所示。

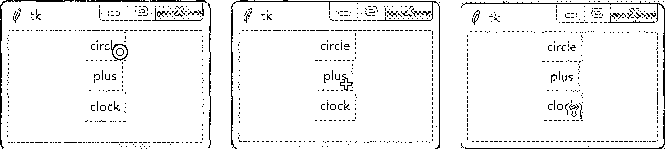


图16-6光标效果图

有三种方式可以设置组件属性:

t在创建组件对象日第使用梅造函数来创建组件弄谖置属性 示例如下：

button= Button (topr text ="c丄ock”)

2.组件对象创建眉，使照寧翼索引的商武谩置属魅 示例如下:

button[° fg°] = "red"

button[nbg°] = "blue"

&使照configO方法更新多个属性对象

示例如下:

button . config (fg=nred" , bg=nblue° )

下面看一个案例，具体如下：

from tkinter import \*

root=Tk()

button=Button(rootz text=1 clock')

|  |  |
| --- | --- |
| button[\* fg \*] ='white \* |  |
| button[\* bg \*] =!blue \* |  |
| button e pack() |  |
| root 8mainloop() |  |
| 运行结果如图16・7所示。 |  |
| 1戒 3 | 图16-7运行结果 |

16.3J ffWB （ Butoo ）

按钮组件使用Button表示，它可以显示文字或图像。创建按钮组件的语法格式如下所示: button = Button （ master^ option=value7 …。）

参数介绍如下：

* master:代表父窗口*。*
* option:组件的属性选项列表。

以下是这个组件最常用的选项列表，具体如表1&3所示。

表16・3 Button组件属性

|  |  |
| --- | --- |
| 属 性 | 说 明 |
| background | 背景颜色 |
| borderwidth | 边框宽度（像素），默认是2个像素 |
| foreground | 正常前景（文字）颜色 |
| height | 高度（用于文本按钮）或像素（用于图像） |
| width | 宽度（用于文本按钮）或像素（用于图像） |
| image | 要显示在标签上的图像 |
| padx | 文字左侧和右侧的附加填充 |
| pady | 文本上方和下方的附加填充。 |
| text | 按钮要显示的内容 |
| command | 点击按钮时触发的动作 |

按钮是可以点击的，如果希望点击按钮后可以执行一定的功能，我们可以使用command 属性来设置回调函数。下面是一个Button组件的使用案例，具体代码如下：

from tkinter import \*

root = Tk（）

def callback（）:

print （'学Python,来黑马程序员1）

button = Button （rootf text= F 人生苦短,我用 Python 1 command=callback）

button.pack（）

root.mainloop（）

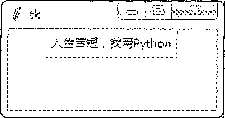
上述代码中，第5行代码创建了一个按钮组件，并使用坨xt 设置按钮的显示文字，使用command属性指定按钮的回调函 数callback,这样一来，我们只要点击一次按钮，就会调用一次 callback 函数。

图16-8运行结果

效果如图16『8所示。

16.3.2 If ( Lab®0) 「

标签组件可以说是最简单的组件了，它不执行任何功能,只用于显示信息。创建标签组件 的语法格式如下所示。

Label = Label ( masterz optionz 。。。)

参数介绍如下：

° master:代表父窗口。

® option:组件的属性选项列表。

以下是Label组件最常用的选项列表，具体如表16.4所示。

表16-4 Label组件常见属性

|  |  |  |
| --- | --- | --- |
| 属 性 | 说 明 | |
| background | 标签背景颜色 | |
| borderwidth | 标签边框宽度(像素)，默认是2 | |
| foreground | 正常前景(文字)颜色 | |
| height | 标签高度 | |
| width | 标签的宽度 | |
| image | 要显示在标签上的图像 | |
| padx | 文字左侧和右侧的附加填充 | |
| pady | 文本上方和下方的附加填充 | |
| state | 标签状态，其值可以是NORMAL、ACTIVE, DISABLED | |
| 下面在一个窗口中使用标签组件显示信息，具体代码如下：  from tkinter import \* root=Tk()  label=Label (root, text=z 学 Python,来黑马程序员') label opack()  root.mainloop()  效果如图16.9所示。 | |  |
| *板KKKKK!機霽錦繭St爛KBti蠢!* |
| [ Mg f来黑马程序悬 |
| 图16-9运行结果 |

16.3.3 黨奉偃(Eotry}

Entry用于接收单行文本字符串的输入，创建文本框组件的语法如下所示。

entry = Entry( master, option, ….)

参数介绍如下：

* master:这代表了父窗口。
* option:组件的属性选项列表。

以下是Entry组件最常用的选项列表，具体如表所示。

表16-5 Entry组件常用属性

|  |  |  |
| --- | --- | --- |
| 属 性 | 说 明 | |
| background | 文本框背景颜色 | |
| borderwidth | 文本框边框宽度 | |
| foreground | 文字颜色，值为颜色或者颜色代码，如"red”、"#ffOOO” | |
| highlightthickness | 文本框高亮边框的宽度 | |
| highlightbackground | 文本框高亮边框颜色，当文本框未获取焦点时显示，只有 设置了 highlightthickness属性，设置该属性才会有效 | |
| highlightcolor | 文本框高亮边框颜色，当文本框获取焦点时显示，只有设 置了 highlightthickness属性，设置该属性才会有效 | |
| selectbackground | 选中文字的背景颜色 | |
| show | 指定文本框内容显示为字符，满足字符即可。如密码可以 将值设为\* | |
| textvariable | 文本框的值，是一个StringVar()对象 | |
| width | 文本框宽度 | |
| xscrollcommand | 文本框口水平滚动。设置这个选项为水平滚动条的方法是 set() | |
| 需要注意的是*，*Entry组件只有width属性，没有heigth属性。 如果想获得文本框的内容，可以调用组件的get方法来实现。 下面看一个文本框的案例,具体代码如下:  from tkinter import 大  top = Tk()  Label = Label (topz text=zzUser Namez,)  Label.pack( side= LEFT) entry = Entry(top, bd=5) entry opack(side = RIGHT) top。mainloop()  运行效果如图16J0所示。  1 割乳盛 | | ;User Name | j  图16-40运行结果 |

Text组件用来显示和处理多行文本。在Tkint如所有组件中9 Text组件显得很灵活，而且 功能也很强大，适用于多种任务。虽然Text组件的主要目的是显示多行文本，但它常常被用于 作为简单的文本编辑器和网页浏览器使用。创建多行文本框的语法格式如下所示。

text = Text ( master, option^ 。… )

参数介绍如下；

* master:代表父窗口。
* option:组件的属性选项列表o

以下是Text组件最常用的选项列表，具体如表16^6所示。

|  |  |  |
| --- | --- | --- |
| 表〔6・6 Text组件常用属性 | | |
| 属 性 | 说 明 | |
| background | 文本框背景颜色 | |
| borderwidth | 多行文本框边框宽度 | |
| foreground | 文字颜色，值为颜色或者颜色代码，如'red，、'#ffOOO' | |
| highlightthickness | 多行文本框高亮边框的宽度 | |
| highlightbackground | 多行文本框高亮边框颜色，当文本框未获取焦点时显示，只有设置了 highlightthickness 属性，设置该属性才会有效 | |
| highlightcolor | 多行文本框高亮边框颜色，当文本框获取焦点时显示，只有设置了 highlightthickness属 性，设置该属性才会有效 | |
| selectbackground | 选中文字的背景颜色 | |
| - state | 指定文本框内容显示为字符，满足字符即可。 如密码可以将值设为 | |
| width | 文本框宽度 | |
| xscrollcommand | 文本框口水平滚动 | |
| yscrollcommand | 文本框口竖直滚动 | |
| 下面是一个Text组件的使用案例9具体如下:  from tkinter import \* root = Tk()  label=Label (rootz text= 1请写下您的意见或者建议, ) label.pack()  text=Text(root,width=30 f height=5 7 bg=!yellow f,hig hlightbackground='red \*)  text.pack()  root oinainloop ()  效果如图1641所示。 | |  |
|  |
| 暴下雜意舞券建發 1  *鹽養辭纜饗备餐餐経縁縁経蓦養薰t\*槌礬\*冬普?藉経赢普年1*  *"...* |
| 图1641运行结果 |

1®o3n5 鶯繼鐵翘(^adiobutoo)

单选按钮使用Radiobutton表示，它用于选择一组选项中的一个。创建Radiobutton组件的 语法如下所示。

radio\_button = Radiobutton ( master, option, …。 )

参数介绍如下:

* master:代表父窗口。
* option:组件的属性列表选项。

以下是Radiobutton组件最常用的选项列表，具体如表16^7所示。

|  |  |
| --- | --- |
| -7 Radiobutton缉件常用属性 | |
| 属 性 | 说 明 |
| background | 按钮背景颜色 |
| foreground | 文字颜色，值为颜色或者颜色代码，如'red\ '#fm00' |
| activebackground | 当鼠标在按钮上的背景颜色 |
| activefbreground | 当鼠标在按钮上的前景颜色 |
| borderwidth | 边框的宽度，默认是2像素 |
| command | 点击该按钮时触发的动作 |
| image | 单选按钮文本图形图像显示 |
| highlightbackground | 文本框高亮背景颜色 |
| highlightcolor | 文本框高亮边框颜色 |
| justify | 如果文本包含多行，此选项将控制文本在CENTER （默认）、LEFT、RIGHT |
| padx | 单选按钮与文本内容左边距或右边距，默认值是1 |
| pady | 单选按钮与文本内容上边距或下边距，默认值是i |
| state | 指定单选按钮的状态 |
| text | 单选按钮旁边的文本。多行文本可以用气T来换行 |
| value | 指定的是RadioButton关联的值 |
| variable | 指定的是RadioButkm选中时设置的变量名，这个必须是全局的变量名，可以使用 get函数获取 |
| width | 字符中的标签宽度（不是像素），如果未设置此选项，则标签将按其内容进行大小调整 |

下面看一个Radiobutton组件的使用案例，具体代码如下:

from tkinter import \*

def sei(): selection = "You selected the option " + str(var.get()) label <, config (text=selection)

root = Tk() var = IntVar() radio\_button\_one = Radiobutton(root, text="Option 1", variable=varf

value=lf command=sel) radio\_button\_one.pack() radio\_button\_two = Radiobutton(root, text="Option 2H, variable=varz

value=2 z command=sel) radio\_button\_two.pack() radio\_button\_three = Radiobutton(root, text=HOption 3°, variable=var,

value=3 f command=sel) radio\_button\_three•pack() label = Label(root) label»pack() root einainloop ()

效果如图1642所示。

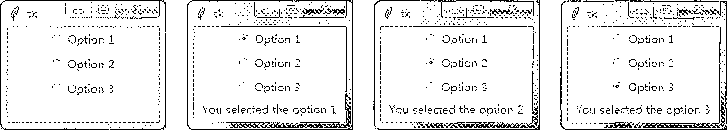


图1642运行结果

1®q3d® 富SB ( Cheekbofttoo )

复选框使用Checkbutton表示,它可以在多个选项中选择多项。创建Checkbutton组件的语 法如下所示。

check = Checkbutton (master z option^ …。)

参数介绍如下：

* master:代表父窗口。
* option:组件的属性列表选项。

以下是Checkbutton组件最常用的选项列表，具体如表16-8所示。

表16-8 Checkbutton组件常用属性

|  |  |
| --- | --- |
| 属 性 | 说 明 |
| background | 复选框背景颜色 |
| foreground | 文字颜色，值为颜色或者颜色代码，如Ted，、'#ffOOO' |
| activebackground | 当鼠标在复选框的背景颜色 |
| aciiveforeground | 当鼠标在复选框上的前景颜色 |
| borderwidth | 边框的宽度，默认是2像素 |
| command | 点击该复选框时触发的动作 |
| image | 复选框文本图形图像显示 |
| highlightcolor | 复选框高亮边框颜色，当复选框获取焦点时显示 |
| justify | 如果文本包含多行，此选项将控制文本在CENTER (默认)、LEFT或RIGHT |
| padx | 复选框与文本内容左边距或右边距，默认值是1 |
| pady | 复选框与文本内容上边距或下边距，默认值是1 |
| state | 指定复选框的状态 |
| text | 单选按钮旁边的文本。多行文本可以用气1”来换行 |
| variable | 指定的是Checkbutton选中时设置的变量名，这个必须是全局的变量名 |
| width | 字符中的标签宽度(不是像素)，如果未设置此选项，则签将按其内容进行大小调整 |
| height | 文本占的行高度，默认是1行的高度 |

下面看一个Checkbutton组件的使用案例。

from tkinter import \*

top = Tk()

label=Label (top, text= 1请选择您爱好的球类运动:7 )

fheight=2 rwidth *= 2*0) check\_one。pack() check\_two。pack() check\_three。pack() top oinainloop () 效果如图16-43所示。

| ° z height=2 fwidth = | 20) |  |  |
| --- | --- | --- | --- |
| che c k\_two = | Checkbutton (topz text = | "篮 | 球 |
| "fheight=2 f width = | 20) |  |  |
| check\_three | =Checkbutton(topf text = | ，，排 | 球 |

label.pack()

check\_one = Checkbutton (top, text =叫足 球

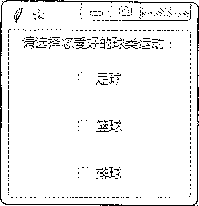
16h3q7 刿譎滙(Lostbon)

图16-13运行结果

列表框部件用于显示一个项目列表。创建Listbox组件的函数如下所示。 List \_box = Listbox (master^ option^ …) 参数介绍如下：

* master:代表父窗口。
* option:组件的属性列表选项。

以下是Listbox组件最常用的选项列表，具体如表16・9所示。

表16-9 Listbox组件常用属性

|  |  |
| --- | --- |
| 属 性 | 说明 … |
| background | 列表框背景颜色 |
| foreground | 文字颜色，值为颜色或者颜色代码，如'gT、!#ffOOO1' |
| height | 列表框中的高度，单位是行的高度，而不是像素 |
| highlightcolor | 当组件突出重点时,重点显示的颜色 |
| selectbackground | 显示选定文本的背景颜色 |
| width | 字符中的组件的宽度。默认值为2。 |
| xscrollcommand | 如果你想允许用户水平滚动列表框，可以把列表框组件链接到一个水平滚动条 |
| yscrollcommand | 如果你想允许用户垂直滚动列表框，可以把列表框组件链接到一个垂直滚动条 |

下面看一个Listbox组件的使用案例。 from tkinter import \*

top = Tk()

List\_box = Listbox(top)

List\_box.insert (1r

List\_box.insert (2 f

List\_box.insert (3r

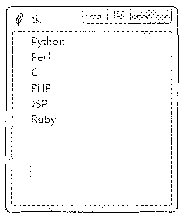
List\_box。insert (4 7

List\_box.insert (5Z

List\_box.insert(6 f "Ruby")

"Python0) nPerln) nC°) nPHPn)

°JSP°)

List\_box.pack() top。mainloop()

i 16o4 »

菜单可以容纳的信息量非常大，因为一个菜单可以有子菜单，每个菜单都包含多个选项。 在Python中,菜单使用Menu类创建,该类提供了 add command方法，该方法的command属 性指定单击该菜单项时执行的函数。

下面，我们按照下列三步创建一个菜单：

(1 )使用Menu类新建一个菜单。

(2 )调用add command方法添加菜单项

(3)指定菜单项的名称及被点击时调用的方法。

实现代码如下:

from tkinter import \*

top = Tk()

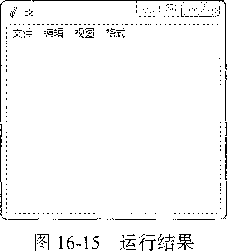
menu = Menu(top)

def callback():

print (\* this is menu8)

for item in卩文件七，编辑七？视图七『格式叮：

menu.add command(label=itemA command=callback)

top [ ?menu\* ] = menu

top.mainloop ()

效果如图16J5所示。

代码介绍如下《

上述代码中，使用Menu类来新建一个菜单,Menu和其他的 组件一样，第一个是parent,这里通常可以为窗口。

然后我们用add\_commmand方法来为它添加菜单项,如果该 菜单是顶层菜单，则添加的菜单项依次向右添加。如果该菜单是顶 层菜单的一个菜单项，则它添加的是下拉菜单的菜单项。

add\_comnwid中的参数常用的有label属性，用来指定的是菜单项的名称，command属性 用来指定被点击的时候调用的方法。

最后使用窗口的menu属性指定它的顶层菜单。

1SD4n2

如果有子菜单，那么我们需要使用菜单对象的add\_cascade方法，该方法可以将另一个菜 单作为本菜单的下拉菜单，睫scrnie可以理解为“级联”，即它的作用只是为了引岀后面的菜单。 add\_cascade方法同样具有menu和label属性。

下面我们新建菜单对象,然后使用add\_command添加菜单项,最后使用add cascade方法 将这些新建的菜单，作为子菜单关联到主菜单上，具体代码如下：

from tkinter import \*

top = Tk()

menu = Menu(top)

fmenu = Menu(menu)

for item in ['新建1 , 1保存\* ,'另存为1 , 1关闭9 ]: fmenu.add\_command(label=item)

emenu = Menu(menu)

for item in [ 1复制1 ,'粘贴','全选1 7 f清除1 ]: emenu <, add\_command (label=item)

vmenu = Menu(menu)

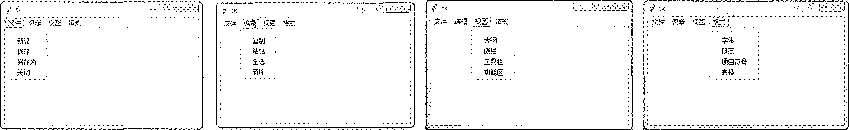
for item in [ 1大纲'’'侧栏1 , 1工具栏\* ，功能区，]： vmenu。add\_command(label=item)

gmenu = Menu(menu)

for item in [ ?字体',1段落',1项目符号1, \*表格\* ]:

gmenu.add\_command(label=item)

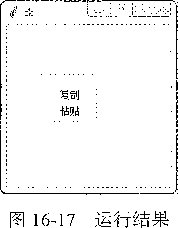
menu« add\_cascade (label= s 文件 ? f menu=fmenu) menu . add\_cascade (labels 1 编辑 \* menu=emenu) menu e add\_cascade (labels1 视图 f r menu=vmenu) menu e add\_cascade (label= 5 格式 1 f menu=gmenu) top[fmenu ?] = menu



top.mainloop()

运行结果如图16』6所示。

图1646运行结果



16.4.3瓣鑑就藜颦

弹出式菜单又称“右键菜单” ?它通常是鼠标右击产生的菜单。鼠标右击，此时可以根据

需要判断鼠标位置来确定是哪个弹出菜单被弹出须然后使用Me®类的pop方法弹出菜单即可。 下面是一个弹出菜单的使用案例。

from tkinter import \* root = Tk()

menu = Menu(root)

for item in [ 1 复制 1 , 9 粘贴 1 ]:

menu. add.\_command (label=item) def pop(event):

menu.post(event.x\_rootA event.y\_root) root o bind(\* <Button-3> ? f pop) root emainloop()

Tkin贤提供了多种类型的对话框，这里我们介绍三种，分别是消息对话框、文件对话框和 颜色选择对话框。

ISnSol ( messagebox)

messagebox是tkinter的一个子模块,它用来显示文本信息、提供警告信息或错误信息。 messagebox 包含的消息框类型有 showinfb, showwaming, showerror, askquestion, askokcancel, askyesno 9 askretrycancel等。这些消息框有相同的语法，具体如下:

tkMessageBox.FunctionName(titlez message [, options])

参数的介绍如下：

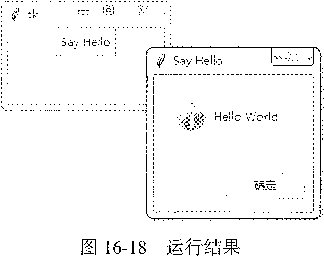
• title:指的是消息框的标题栏显示的文本。

® message:指的是消息框的文本消息。

® options:可以调整外观的选项。

除此之夕卜，这些消息框还有一些默认的常量，它们都在messagebox中定义好了，具体如下:

1. 默认按钮:ABORT, RETRY、IGNORE. OK、CANCEL. YES、NO
2. 图标常量：ERROR,INFOQUESTION,WARNING

下面，我们看一个消息对话框的例子，具体如下：

from tkintersmessagebox import \*

from tkinter import \*

top = Tk()

def hello(): showinfo("Say Hello"f "Hello World") button = Button(topz text="Say Hello"z command=hello) button ..pack () top.mainloop() 效果如图1648所示。

16.5.2 (ffOedofog )

几乎所有实用的桌面应用都有打开文件进行读取或写入的功能。有些程序还需要提供选择 目录功能o tkinter中的filedialog包提供了这个功能。

下面是一个文件对话框的例子。

from tkinter import \*

from tkinter .filedialog import \*

def callback():

name = askopenfilename () print(name)

button = Button(text= , File Open J 7 command^callback)

button.pack()

mainloop()

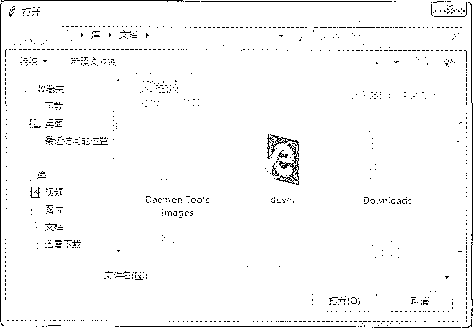
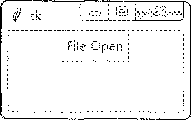


图1649运行结果

1®q5q3 11 鱼鑿鶴调谐滙(c©ferch©©s®B")

tkinter中还有一个模块是colorchooser ,它可以实现颜色的选择。colorchooser获取颜色的 语法格式如下所示。

result = tkColorChooseraskColor (color 7 option=va.luef °。。)

上述代码中,如果用户单击颜色选择对话框的确认按钮? askColor()方法会返回一个二元组, 例如? ((106? 150?98)J#6a9662?)?该二元组的第一个元素是一个代表三原色RGB的三元组，每 种原色的取值范围为0-255；第二个元素是一种十六进制表示颜色的方法。如果用户单击取消 按钮9 askColor()方法同样会返回一个二元组，值为(None, None)。

在askColor()方法中的options参数可以是以下选项:

-color0使用颜色选择对话框设置一个默认选择的颜色，如果未指定color选项，则默认 选择颜色为灰色。

• titleo显示在对话框顶部标题栏的标题文本，默认的标题为“Color” o

° parento设置后，颜色选择对话框会在parent这个窗口之上显示。默认行为是在root窗 口之上显7K o

from tkinter import \*

from tkinter.colorchooser import \*

def callback():

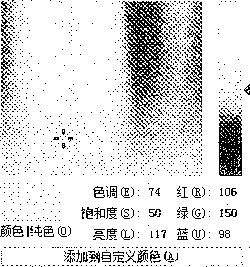
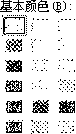
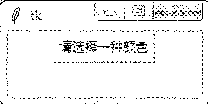
result = askcolor (color="#6A9662nz title ="请选择一种您喜欢的颜色") print(result)

root = Tk()

button = Button (rootf t㊀xt='请选择一种颜色 \* command=callback)

button。pack()

mainloop ()



*讓ggs.囲夢* 厂"繼讒®:

图 16-20

运行结果

« I6o6 「

Canvas (画布)是一个矩形区域，用于绘制图形或者其他复杂的布局。我们可以在画布上 放置图形、文本、组件或者框架。

创建画布的语法格式如下所示:

canvas = Canvas (master7 option=valuez …。)

参数介绍如下：

° master:表7K 父窗口。

* option：最常用的选项列表。这些选项可以用作键值对，由逗号分隔。

使用Canvas对象的方法可以绘制的图形如下所示。

* create\_arc()绘制圆弧,格式如下:

arc = canvas » create\_arc(x0 z y0 xlz ylstarts extentz option)

上述格式中? (x03y0)表示左上角,(xl3yl)表示右下角，start表示起始角度，默认为0。， extent表示终止校对，默认是90。。

* createjine()绘制直线或者折线，格式如下？

line = canvas. create\_line (xO, yO, xl, ylz xn, ynz options)

上述格式中，(x0,y0), (xl9yl),…，(xn,yii)表示的是折线上的坐标点。

* create\_oval()绘制椭圆,格式如下:

oval = canvas.create\_oval(x0, yO, xl, ylr options)

上述格式中,(x0,y0)表示左上角,(xl,yl)表示右下角。

* create\_ploygon()绘制多边形,格式如下:

oval = canvas . create\_polygon (x0, yO, xl, yl7 . . .,xnz ynz options)

上述格式中,(x0,y0), (xl,yl),…，(xn,yn)表示的是多边形各个顶点坐标。

* create\_rectangle()绘制矩形,格式如下:

rect = canvas.create\_rectangle(x0 f yO z xl,ylf option)

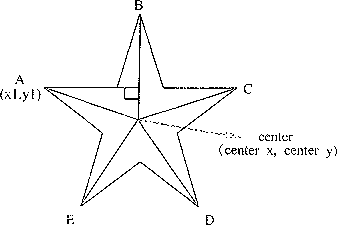
上述格式中，(xO,yO)表示左上角，(xl?yl)表示右下角。

使用Canvas绘制的图形，都支持很多option选项，常见的选项如下所示。 • width:边框线宽。

。fill:填充颜色。

* outline:边框颜色。
* dash:边框线型。
* arrow:绘制的直线是否带箭头，取值可以 为FIRST、LAST、BOTH,分别用于设 置在起点、终点还是两端带箭头。

下面，我们使用Canvas绘制一个五角星， 具体如图16-21所示。

图162所示的五角星是一个多边形，我们 只需要计算出多边形各个顶点的坐标即可完成绘

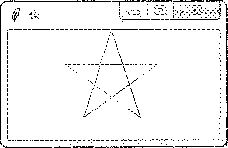
五角星

图 16-21

制。具体代码如下所示。

from tkinter import import math

root = Tk() canvas = Canvas(root, width=200z height=100) canvas。pack() center\_x = 100 center\_y = 50 r = 50 points =[

#左上点

| center\_x - | int(r \* | math.sin(2 \* | math.pi / | 5))' |
| --- | --- | --- | --- | --- |
| center\_y 一 | int(r \* | math.cos (2 \* | math.pi / | 5)), |
| #右上点 |  |  |  |  |
| center\_\_x + | int (r \* | math.sin (2 \* | math.pi / | 5) ) *E* |
| center\_y - | int (r \* | math o cos(2 \* | math.pi / | 5)), |
| #左下熹 |  |  |  |  |
| center\_x - | int(r \* | math.sin(math | / 5)), |  |
| center\_y + | int(r 大 | math.cos(math | •pi / 5)), |  |
| #顶点— |  |  |  |  |
| center\_\_xf |  |  |  |  |
| center\_y - | r, |  |  |  |
| #右下熹 |  |  |  |  |
| center\_x + | int(r \* | math.sin(math | .pi / 5))z |  |
| center\_y + | int(r \* | math•cos(math | •pi / 5)) |  |

]

canvas.create\_polygon(pointsz outline='red1f fi丄丄=宀) 一

root Omainloop()

图16-22运行结果

运行结果如图16.22所示。

*k*多学-■■招：绘制扇形、弓形与圆弧

当我们使用create\_arc()方法绘制圆弧时,可以传入了 style选项,style选项的值可以是 ARC (默认值)、CHORD或PIESLICE,下面先看一段示例代码。

from tkinter import \*

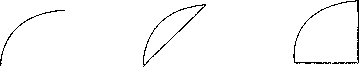
root = Tk()

canvas = Canvas(width=200z height=200)

canvas.pack()

* canvas。create\_arc (100 ? 100 r 200 f 200 f start:=90 f extent=90 r style=ARC)
* canvas.create\_arc(100^100^200^200^ start=90 f extent=90 style=CHORD)
* canvas.create\_arc(100^100^200^200f start=90,extent=90 z style^PIESLICE) root oinainloop ()

上述代码中，使用不同styk值绘制出的效果也是不同的，如图16.23所示。



style = ARC

style = CHORD

style = PIESLLCE

图16-23 style不同值对应的效果

ISof Tlrnter

Tkinter支持三种几何布局管理器，分另U是pack、grid和place*，*它们主要决定组件的排列位置， 注意，不管是哪种几何布局管理器，在同一个父窗口中°他们是不可以混用的。下面对这几种 布局管理器进行介绍。

pgk是一种几何布局管理器，在前面的例子，我们已经用过很多次了，它可以看作一个容 器，将创建的组件添加到父组件中。

pack的常用属性如下:

姻expand：如果设置为Tme,那么组件会扩展填充。

。fill：确定组件是否填充额外空间，其值可以是non©、x、y或both。

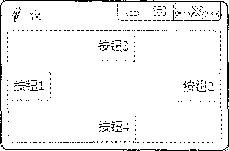
® side:决定组件的排列方式,取值可以为TOP (默认)、BOTTOM. LEFT或RIGHT。 下面看一个例子，具体如下： from tkinter import \* root=Tk() button\_one=Button (text= !按钮 11 ) button\_one.pack(side^LEFT) button\_two=Button (text= ' 按钮 2 1 ) button\_two »pack(side=RIGHT) button\_three=Button (text= 5 按钮 3 ') button\_three.pack(side=TOP) button\_f our=Button (text=1 按钮 4 v) button-four.pack(side=BOTTOM) root.mainloop() 效果如图16.24所示。

图16-24 pack布局管理器

彳盅久2 grid JM風赫贏簷1■器

Grid（网格）几何布局管理器会将组件放置到一个二维的表格里，主组件被分割成一系列的 行和列，表格中的单元格都可以放组件。下面，我们看一个图§如图16-25所示。

|  |  |  |
| --- | --- | --- |
| Label-01 | Entry-01 | Button |
| Label-02 | Entry-02 |

图16-25表格

图1&25共有2行3列，布局中包含了 5个组件，其中: ° Lable-01在第0列,第0行。

° Lable-01在第0列,第1行。

° Entry-01在第1列，第0行。

° Entry-02在第1列，第1行。

° Button在第2列，占两行位置。

下面编写代码，实现上述布局，具体如下：

from tkinter import \*

root = Tk()

Label (rootz text=nFirst" ) . grid (row=0) # 位于第 1 行的标签组件

Label (root, text="Second" ) » grid (row=l) # 位于第 2 行的标签组件

entry\_one = Entry(root) entry\_two = Entry(root)

button = Button (rootz text= 1 计算 1 , height=2) # 按钮的高度占据两行

button . grid (row=0, column=2 f rowspan=2) # 按钮位于第 1 行第 2 歹0,且跨 2 行 entry\_one . grid (row=0 r column=l) # 位于第 1 行,第 2 列的文本框

entry\_two <. grid (row=l r column^l) # 位于第 2 行，第 2 列的文本4匡

root.mainloop()

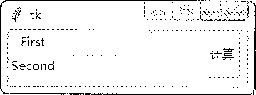
上述代码中,参数row表示的是行,column表示的列, rowspan表示占据的列数，其中，参数row和column都是从0 开始,也就是说,当row=0时表示第1行,当column=0时表示 第1列，依此类推。

图16-26运行结果

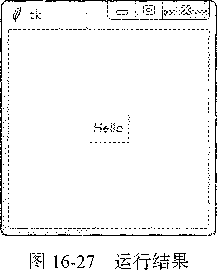
效果如图16-26所示。

1 ⑥冒总

place几何布局管理器可以指定组件放在一个特定的位置，它分为绝对布局和相对布局， place使用的常见属性如下所示。

* anchor :组件其他选项的确切位置。
* relx? rely :相对窗口宽度和高度的位置，取值范围是［0,1.0］。例如，relx=05rely=0位置为 左上角，relx=0.59rely-0.5位置为屏幕中心。
* x9 y :绝对布局的坐标，单位是像素。

下面看一个place几何布局的案例。•

from tkinter import \*

from tkinter.messagebox import \* root = Tk() def hello\_call\_back(): showinfo( "Hello Python"r "Hello World") button = Button(rootr text="Hello"f command=hello call\_back) button opack() button.place(relx=0.5 f rely=0 » 5 z anchor=CENTER) root.mainloop()

效果如图1627所示。

简单来说，事件就是触发的一些动作歹比如，单击菜单，鼠标单击等，根据不同的动作， 我们希望程序可以执行的功能是不同的。这时，我们需要对事件进行处理，也就是调用不同的 函数来完成功能。下面我们来介绍Tkinter的事件。

16.8J TWoter

Tkintew中的事件都使用字符串来描述，基本格式如下：

<modifier-type-detail>

上述格式讲解如下：

。type指的是事件的类型。常见的事件类型如表16J0所示。

表16-10常见的type类型

|  |  |
| --- | --- |
| type | - 含 义 |
| Activate | 组件从“未激活”到“激活”触发的事件 |
| Button | 点击鼠标触发的事件，detail可以指定具体的哪个按键： <Button-1>鼠标左键  <Button-2>鼠标中键  <Button-3>鼠标右键  <Button-4>滚轮上滚  <Button-5>滚轮下滚 |
| ButtonRelease | 用户释放鼠标按键触发的事件 |
| KeyPress | 用户按下键盘按键触发的事件 |
| Configure | 组件尺寸发生变化时触发的事件,detail可指定哪个按键 |
| Enter | 鼠标进入组件触发的事件  注意，这里指的不是用户按下回车键 |
| Motion | 鼠标在组件内移动的整个过程都会触发的事件 |

• modifier或detail可以提供一些附加信息，多数时候可以省略不写，常见的modifier附加 信息如表1641所示。

W 16-11 modifier常见的附加信息

|  |  |
| --- | --- |
| modifier | 含 义 |
| Alt | 当按下Ah键时 |
| Any | 表示任何类型的按钮被按下时 |
| Control | 当按Ctrl键时 |
| Double | 当后续两个事件被连续触发的时候  例如，<Double-Button-l>表示鼠标左键的双击事件 |
| Shift | 当按下Shift按键时 |
| Triple | 跟Double类似，它表示的是三个事件被连续触发 |

下面罗列一些Tkinter事件中常用的组合键：

* <Any-Key-x>:任何一个按键+X。
* <Alt-Key-x>: Alt+X。
* <Control-Key-x> : Control+Xo
* <Shift-Key-x> : Shift+X。
* <Alt-Button-1 >: Alt+ 鼠标左键。
* <Control-Button-1 > : Control+ 鼠标左键。
* <Shift-Button-l> : Shift+ 鼠标左键。

1嵐既宏 事歸蘭篆履屬錢

事件对象是一个标准的Pythoii对象类，拥有大量的属性去描述事件。事件对象的常用属 性如表16-12所示。

>16-12事件的常用属性

|  |  |
| --- | --- |
| 属 性 | *含义* |
| widget | 触发事件的组件 |
| x,y | 当前的鼠标位置，单位：像素 |
| x\_root,y\_root | 当前鼠标位置相对于屏幕左上角的位置，单位\*像素 |
| char | 字符代码（仅键盘事件）字符串的格式 |
| keysym | 按键符合（仅键盘事件） |
| keycode | 按键代码（仅键盘事件） |
| num | 按钮数字（仅鼠标按键事件） |
| width/height | 组件的新形状（仅coniigiire事件） |
| type | 事件类型 |

当事件为vKey>、vKeyPress〉、<KeyRelease> , detail可以通过设定具体的按键名（keysym） 来筛选，例如，<Key.H>表示按下键盘上的大写字母H时触发事件。表16』3罗列了键盘特殊按 键的 keysym 和 keycode o

表16-13键盘的按键

|  |  |  |
| --- | --- | --- |
| 按键名 | 按键码 | 代表的按键 |
| Alt\_L | 64 | 左边的Alt按键 |
| Alt\_R | 113 | 右边的Alt按键 |
| Backspace | 22 | Backspace(退格)按键 |
| Control\_L | 37 | 左边的Ctrl按键 |
| Control\_R | 109 | 右边的Ctrl按键 |
| Delete | 107 | Delete按键 |
| End | 103 | End按键 |
| Cancel | 110 | break按键 |

1®n@n3

将组件和事件关联起来就是事件绑定。在Python中"kin宜允许我们将事件绑定在组件上, 格式如下所示。

widget obind(events handler)

如果相应的Rvmt发生了，就会调用handler处理事件。例如，处理鼠标点击按钮事件，具 体如下：

from tkinter import \*

from tkinter.»messagebox import \*

def handler (event)::

showinfo ("点到了 °,，你好! ' )

root=Tk()

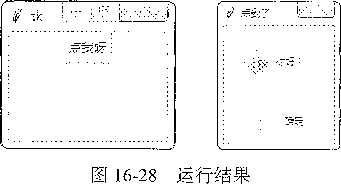
button=Button (rootf text= ' 点我呀 \* )

button.bind('<Button-l> ? z handler)

button o pack()

root »mainloop()

效果如图16^28所示。



除此之外，tkinter还运行将事件绑定在类上，这样这个类的任何一个实例都会触发事件。 格式如下所示。

widget.bind\_class(1 widget17 eventz handler)

如果希望将一个事件绑定在程序的所有组件上，可以使用bind\_疝函数，具体如下：

widget.bind\_all('widget'7 event,handleer)

16o9 »GUI " "

除了 Tkinter, Python还支持很多图形用户界面编程的工具包，这里，我们介绍几种比较流 行的工具包。

16d9J wxPjth©o .

wxPython是Python语言的一套优秀的GUI图形库,允许Python程序员很方便的创建完整 的、功能键全的GUI用户界面。它不仅支持开源，而且具有非常优秀的跨平台能力，能够运行 在Windows.绝大多数的UNIX或类UMX系统、Macintosh OS X ±o根据操作系统的不同， 我们可以进入网站<https://www.wxpython.org> T载不同的安装包，安装完成后，即可使用wxPy- thoxi开发图形用户界面了。 '

下面，我们使用wxPython创建一个Hello Word程序*，*开发流程如下所示。

(1 )导入wx模块。

1. 定义一个表示应用程序类的对象。
2. 创建根窗口对象，设置窗口的尺寸。
3. 在窗口中添加组件对象。

(5 )使用show方法激活窗口。

(6)通知应用程序循环输入应用程序对象。

下面是Hello World程序的代码:

import wx

app = wx.App()

window = wx . Frame (None, title ="黑马程序员"r size = (400 f 300))

panel = wx Panel (window)

label = wx.StaticText(panelz label = "Hello World"r pos = (100z100)) window。Show(True) app.MainLoop()

16.9.2

PyGTK也可以让我们使用Python轻松创建初级具有图形用户界面的程序。底层的GTK+ 提供了各种各样的可视元素和功能。如果需要，我们可以开发在GNOME桌面系统运行的功能 完整的软件。

PyGTK真正具有跨平台性，它能不加修改地、稳定运行于各种操作系统之上，如Linux. Windows. MacOS等。除了简单易用和出色的开发能力外，PyGTK还有一流的处理本地化语 言的独特功能。

由于PyGTK不是Python官方安装程序的一部分，因此要从其官方网站http://www.pygtk„ org/下载PyGTK的安装程序°由于PyGTK的开发过程与wxPython类似,这里我们不再进行介绍, 感兴趣的读者可以在官网自行了解。

328 o Pjthoo Pythoo

凱芻PjQft .

PyQt是Qt库的Python版本，它实现了一个Pythoii的模块集，是创建GUI应用程序的最 *强大库之一,*因为PyQt实现了一个Python模块集。它有超过300个类,将近6000个函数和方 法,可以运行在所有主要操作系统上，包括UNIX, Windows和Mac。

Qt采用双许可证,开发人员可以选择GPL和商业许可,在此之前，GPL版本只能用在 UNIX系统上,从PyQt的版本4开始,GPL许可证可用于所有支持的平台。

1&凱凰 JPjthoo

JPython是一种完整的语言，而不是一个Java翻译器或仅仅是一个Python编译器，它是一 个Python语言在Java中的完全实现。JPythoii也有很多从CPython中继承的模块库。最有趣的 事情是JPythg不像CPython或其他任何高级语言，它提供了对其实现语言的一切存取。所以 JPython不仅给你WT Python的库，同时也提供了所有的Sva类，这使其有一个巨大的资源库。