

CentOS7和Ubuntu安装Tkinter

Tkinter是什么

Tkinter 模块元素简要说明

Label&Button标签、按钮

Entry&Text输入、文本框

Text设置字体颜色

Listbox列表部件

Radiobutton选择按钮

Scale比例

Checkbutton勾选项

Canvas画布

Menubar菜单

frame框架

messagebox弹窗

pack grid place 放置位置

实例-登录窗口

Combobox下拉菜单

右键下拉菜单

OptionMenu选项

Toplevel顶级窗口

LabelFrame组件

Menubutton菜单按钮

Message：轻松自动换行的文本显示

[PanedWindow窗口布局管理器](#)

[scrollbar滚动条](#)

[Text添加滚动条](#)

[spinbox输入控件](#)

[各种控件查用参数参考文档](#)

[Tkinter之pack所有属性详解](#)

[Tkinter之place所有属性详解](#)

[随窗体伸缩、设置最小窗体、关闭窗口](#)

[读取文件和图片、文件上传实现进度](#)

文本: label、entry、listbox、text

触发函数: button、radiobutton(关联一个变量不同赋值)、checkboxbutton(不同变量、onvalue、offvalue, 默认赋值为True、False), button都有state选项。

combobox菜单、OptionMenu、右键菜单, 通过绑定事件触发函数, 前2个绑定的函数要有形参接收(event)

CentOS7和Ubuntu安装Tkinter

CentOS安装, 需要安装软件包:

```
1 yum install python-tools
2 yum install tk-devel # 第一步已安装这个依赖
3 yum install python3-tkinter # 同时重装了python3
4 Installed:
5  python3-tkinter.x86_64 0:3.6.8-18.el7
6 Dependency Installed:
7  libtirpc.x86_64 0:0.2.4-0.16.el7
8  python3.x86_64 0:3.6.8-18.el7
9  python3-libs.x86_64 0:3.6.8-18.el7
10 python3-pip.noarch 0:9.0.3-8.el7
11 python3-setuptools.noarch 0:39.2.0-10.el7
```

安装的同时也安装了python3, 如果需要恢复到之前的python版本, 需要手动再次安装python。或者可以尝试下载安装包, 使用rpm安装: rpm -ivh 软件包 --nodeps --force, 之后再安装相关的依赖包。

Ubuntu16安装：

```
1 首先sudo apt-get update
2 然后sudo apt-get install python3-tk
```

Tkinter是什么

Tkinter 是使用 python 进行窗口视窗设计的模块。Tkinter模块("Tk 接口")是Python的标准Tk GUI工具包的接口。作为 python 特定的GUI界面，是一个图像的窗口，tkinter是python自带的，可以编辑的GUI界面，我们可以用GUI 实现很多直观的功能，比如想开发一个计算器，如果只是一个程序输入，输出窗口的话，是没用用户体验的。所有开发一个图像化的小窗口，就是必要的。

对于稍有GUI编程经验的人来说，Python的Tkinter界面库是非常简单的。python的GUI库非常多，选择Tkinter，一是最为简单，二是自带库，不需下载安装，随时使用，三则是从需求出发，Python作为一种脚本语言，一种胶水语言，一般不会用它来开发复杂的桌面应用，它并不具备这方面的优势，使用Python，可以把它作为一个灵活的工具，而不是作为主要开发语言，那么在工作中，需要制作一个小工具，肯定是需要有界面的，不仅自己用，也能分享别人使用，在这种需求下，Tkinter是足够胜任的！

这篇文章主要做一个简单概述和实践编程，对于从没有接触过GUI的新手，在脑中树立一个基本的界面编程概念，同时自己也能学会如何简单的实现一些小的图形窗口功能。

对于Tkinter编程，可以用两个比喻来理解：

- 第一个，作画。我们都见过美术生写生的情景，先支一个画架，放上画板，蒙上画布，构思内容，用铅笔画草图，组织结构和比例，调色板调色，最后画笔勾勒。相应的，对应到tkinter编程，那么我们的显示屏就是支起来的画架，根窗体就是画板，在tkinter中则是Toplevel，画布就是tkinter中的容器（Frame），画板上可以放很多张画布（Canvas），tkinter中的容器中也可以放很多个容器，绘画中的构图布局则是tkinter中的布局管理器（几何管理器-geometry），绘画的内容就是tkinter中的一个小组件，一幅画由许多元素构成，而我们的GUI界面，就是有一个个组件拼装起来的，它们就是widget。
- 第二个，我们小时候都玩过积木，只要发挥创意，相同的积木可以堆出各种造型。tkinter的组件也可以看做一个个积木，形状或许不同，其本质都是一样的，就是一个积木，不管它长什么样子，它始终就是积木！所以这些小组件都有许多共性，另外，个人认为，学习界面编程，最重要的不是一开始学习每个积木的样子，不是学习每个组件怎

么用，而是这些组件该怎么放。初始学习中，怎么放远远比怎么用重要的多。网上有大量的文章资料，基本全是介绍组件怎么用的，对于怎么放，也就是tkinter中的布局管理器，都是一笔带过，这对初学者有点本末倒置，或许绝大部分是转载的原因吧，极少是自己真正写的。组件怎么用不是最迫切的，用到的时候再去了解也不迟，边用边学反而更好。因此我将专门写一章，详细介绍布局管理器的使用。

Tkinter 模块元素简要说明

tkinter 类	元素	简要说明
Button	按钮	类似标签,但提供额外的功能, 点击时执行一个动作,例如鼠标掠过、按下、释放以及键盘操作/事件
Canvas	画布	提供绘图功能(直线、椭圆、多边形、矩形) 可以包含图形或位图
Checkbutton	复选框	允许用户选择或反选一个选项, 一组方框,可以选择其中的任意个(类似 HTML 中的 checkbox)
Entry	单行文本框	单行文字域, 显示一行文本, 用来收集键盘输入(类似 HTML 中的 text)
Frame	框架	用来承载放置其他GUI元素, 就是一个容器
Label	标签	用于显示不可编辑的文本或图标
LabelFrame	容器控件	是一个简单的容器控件。常用与复杂的窗口布局
Listbox	列表框	一个选项列表,用户可以从中选择
Menu	菜单	点下菜单按钮后弹出的一个选项列表,用户可以从中选择
Menubutton	菜单按钮	用来包含菜单的组件(有下拉式、层叠式等等)
Message	消息框	类似于标签,但可以显示多行文本
OptionMenu	选择菜单	下拉菜单的一个改版, 弥补了Listbox无法下拉列表框的遗憾
PanedWindow	窗口布局管理	是一个窗口布局管理的插件, 可以包含一个或者多个子控件。
Radiobutton	单选框	允许用户从多个选项选取一个, 一组按钮,其中只有一个可被“按下” (类似 HTML 中的 radio)
Scale	进度条	线性“滑块”组件,可设定起始值和结束值,会显示当前位置的精确值
Scrollbar	滚动条	对其支持的组件(文本域、画布、列表框、文本框)提供滚动功能
Spinbox	输入控件	与Entry类似, 但是可以指定输入范围值
Text	多行文本框	多行文字区域, 显示多行文本, 可用来收集(或显示)用户输入的文字(类似 HTML 中的 textarea)
Toplevel	顶层	类似框架,为其他的控件提供单独的容器
messageBox	消息框	用于显示你应用程序的消息框。(Python2中为tkMessageBox)

Label&Button标签、按钮

```
1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x100')
6
7 var = tk.StringVar()
8 l = tk.Label(window, textvariable=var, bg='green', font=('Arial',12),width=15, height=2)
9 l.pack()
10
```

```

11 on_hit = False
12 def hit_me():
13     global on_hit
14     if not on_hit:
15         on_hit = True
16         var.set('you hit me')
17     else:
18         on_hit = False
19         var.set('')
20 # 以上改进
21 # def hit_me():
22 #     if var.get():
23 #         var.set('')
24 #     else:
25 #         var.set('you hit me')
26
27 b = tk.Button(window, text='hit me', width=10, height=2, command=hit_me)
28 b.pack()
29 window.mainloop()
30

```

Entry&Text输入、文本框

```

1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x200')
6
7 # 隐藏 show='*',显示None, 1: 所有输入都是1
8 e = tk.Entry(window, show=None) # 可以设置背景、字体、board等等
9 e.pack()
10
11 def insert_point():
12     var = e.get()
13     t.insert('insert', var)
14     # t.insert('insert', var + '\n') # 插入并换行
15 def insert_end():
16     var = e.get()
17     t.insert('end', var)

```

```

18     # t.insert(1.1, var) # 第一行第一列
19
20 b1 = tk.Button(window, text='insert point', width=10, height=2, command=insert_point)
21 b1.pack()
22 b2 = tk.Button(window, text='insert end', width=10, height=2, command=insert_end)
23 b2.pack()
24
25 t = tk.Text(window, height=4, width=20)
26 t.pack()
27 window.mainloop()
28

```

text可以关闭折行(即同一行达到文本框长度不显示在第二行)、增加水平滚动条和垂直滚动条见Scrollbar介绍

```

1 参数wrap:
2 1. 设置当一行文本的长度超过 width 选项设置的宽度时, 是否自动换行
3 2. 该选项的值可以是: "none" (不自动换行), "char" (默认) (按字符自动换行) 和
   "word" (按单词自动换行)

```

Text设置字体颜色

```

1 import tkinter as tk
2 import tkinter.font as tf
3 window = tk.Tk()
4 window.geometry('300x400')
5 window.title('测试')
6
7 text = tk.Text(window,width=20,height=20)
8 text.pack()
9
10 char = ['你好','你好呀','你今天真好看','谢谢'] #插入到text中
11 #####更改部分#####
12 for i in range(4):
13     if i == 0 :
14         a = tk.INSERT
15         text.insert(a,char[i]+'\\n','tag') #申明使用tag中的设置
16     elif i == 2:
17         a1 = tk.INSERT
18         text.insert(a1,char[i]+'\\n','tag_1') #申明使用tag__1中的设置
19     else:
20         a2 = tk.INSERT

```

```

21         text.insert('end', char[i]+'\\n', 'normal')
22
23     ft = tf.Font(family='微软雅黑',size=10) ###有很多参数
24     text.tag_add('tag',a) #申明一个tag,在a位置使用
25     text.tag_config('tag',foreground='red',font =ft ) #设置tag即插入文字的大小,颜色等
26
27     text.tag_add('tag_1',a1)
28     text.tag_config('tag_1',foreground = 'blue',background='pink',font = ft)
29
30     text.tag_add('normal', a2)
31     text.tag_config('normal', background='white', font=ft)
32     #foreground字体颜色
33     #font字体样式,大小等
34     #background 背景色
35     #####更改部分#####
36     window.mainloop()
37

```

Listbox列表部件

```

1  import tkinter as tk
2
3  window = tk.Tk(className='window')
4  window.title('my window')
5  window.geometry('200x200')
6
7  var1 = tk.StringVar()
8  l = tk.Label(window, bg='yellow', width=4, height=2, textvariable=var1)
9  l.pack()
10
11  def print_selection():
12      value = lb.get(lb.curselection())
13      var1.set(value)
14
15
16  b1 = tk.Button(window, text='print selection', width=10, height=2, command=print_selection)
17  b1.pack()
18
19  var2 = tk.StringVar()
20  var2.set((11,22,33,44))

```

```

21 lb = tk.Listbox(window, listvariable=var2)
22 list_items = [1,2,3,4]
23 for item in list_items:
24     lb.insert('end', item)
25 lb.insert(1, 'first')
26 lb.insert(2, 'second')
27 lb.delete(2)
28 lb.pack()
29
30 window.mainloop()
31

```

Radiobutton选择按钮

可以设置很多相关参数，主要有：前景、背景、字体、颜色、边框、限制radio长度、长度、高度等等

```

1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x200')
6
7 var = tk.StringVar()
8 var.set(0) # 赋值之后，单选框默认不选择，或者 var = tk.IntVar()
9 l = tk.Label(window, bg='yellow', width=20, height=2, text='')
10 l.pack()
11
12 def print_selection():
13     l.config(text='you have selected ' + var.get())
14
15 # state的值normal、active表现一样，disabled则为灰色不可选。
16 r1 = tk.Radiobutton(window, text='OpitonA', variable=var, value='A', command=print_selection, state='normal')
17 r1.pack()
18 r2 = tk.Radiobutton(window, text='OpitonB', variable=var, value='B', command=print_selection)
19 r2.pack()
20 r3 = tk.Radiobutton(window, text='OpitonC', variable=var, value='C', command=print_selection)
21 r3.pack()
22

```



```
23 window.mainloop()
24
```

设置radio的返回值，通过返回值给之后设计做判断，添加frame，把所有的单选框放frame中，通过选择OpinionB，激活OpitonA

```
1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x200')
6
7 var = tk.StringVar()
8 var.set(0)
9 l = tk.Label(window, bg='yellow', width=20, height=2, text='')
10 l.pack()
11
12 radio_value = ''
13 def print_selection():
14     global radio_value
15     l.config(text='you have selected ' + var.get())
16     radio_value = var.get()
17     print(radio_value)
18
19 def chage_stat():
20     r1.config(state='active')
21
22 # relief= flat, groove, raised, ridge, solid, or sunken
23 frame = tk.Frame(window, relief='raised')
24 frame.pack(side='left')
25
26 r1 = tk.Radiobutton(frame, text='OpitonA', variable=var, value='A', command=
rint_selection, state='disabled', relief='sunken')
27 r1.pack()
28 r2 = tk.Radiobutton(frame, text='OpitonB', variable=var, value='B', command=
hage_stat, relief='ridge' )
29 r2.pack()
30 r3 = tk.Radiobutton(frame, text='OpitonC', variable=var, value='C', command=
rint_selection, relief='ridge')
31 r3.pack()
32
33 window.mainloop()
```

Scale比例

```
1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x200')
6
7
8 l = tk.Label(window, bg='yellow', width=20, height=2, text='empty')
9 l.pack()
10
11 def print_selection(v):
12     l.config(text='you have selected ' + v)
13
14 s = tk.Scale(window, label='try me', from_=0, to=10, orient=tk.HORIZONTAL,
15             th=200, showvalue=1, tickinterval=2, resolution=0.1, command=print_selection)
16 s.pack()
17 window.mainloop()
18
```

Checkbutton勾选项

```
1 import tkinter as tk
2
3 window = tk.Tk(className='window')
4 window.title('my window')
5 window.geometry('200x200')
6
7
8 l = tk.Label(window, bg='yellow', width=20, height=2, text='empty')
9 l.pack()
10
11 def print_selection():
12     # l.config(text='you have selected ' + v)
13     if (var1.get() == 1) & (var2.get() == 0):
14         l.config(text='I only love python')
15     elif (var1.get() == 0) & (var2.get() == 1):
16         l.config(text='I love C++')
17     elif (var1.get() == 0) & (var2.get() == 0):
```

```

18         l.config(text='I do not love either')
19     else:
20         l.config(text='I love both')
21
22     var1 = tk.IntVar()
23     var2 = tk.IntVar()
24     # 通过触发函数，激活某个操作，也可以让对象实体从灰色变成active，从而可执行某些操作
25     # 和radiobutton一行，checkboxbutton也有state='disabled'状态选项
26     c1 = tk.Checkbutton(window, text='Python', variable=var1, onvalue=1, offvalue=0, command=print_selection)
27     c2 = tk.Checkbutton(window, text='C++', variable=var2, onvalue=1, offvalue=0, command=print_selection)
28     c1.pack()
29     c2.pack()
30     # c1.pack(side='left')
31     # c2.pack(side='left')
32
33     # c1.place(x=50, y=50, anchor='nw')
34     # c2.place(x=50, y=70, anchor='nw')
35
36     window.mainloop()
37

```

也可以通过以下判断是否选中：

```

1  h1 = tkinter.BooleanVar() # 设置选择框对象,通过if h1.get() == True判断，选中为真，不选为假
2  cb1 = tkinter.Checkbutton(wuya,text='周杰伦',variable=h1,command=func)
3  cb1.pack()

```

Canvas画布

```

1  from os import startfile
2  import tkinter as tk
3
4  window = tk.Tk(className='window')
5  window.title('my window')
6  window.geometry('200x200')
7
8  canvas = tk.Canvas(window, bg='blue', height=100, width=200)
9  image_file = tk.PhotoImage(file='./ins.png')
10 image = canvas.create_image(0,0, anchor='nw', image=image_file)
11 x0,y0,x1,y1 = 50,50,80,80
12 line = canvas.create_line(x0,y0,x1,y1)

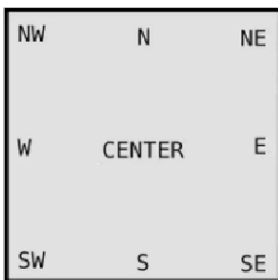
```

```

13 oval = canvas.create_oval(x0,y0,x1,y1, fill='red')
14 arc = canvas.create_arc(x0+30, y0+30, x1+30, y1+30, start=0, extent=180)
15 rect = canvas.create_rectangle(100, 30, 100+20, 30+20)
16 canvas.pack()
17
18 def moveit():
19     canvas.move(rect, 0, 2)
20 b = tk.Button(window, text='move', command=moveit)
21 b.pack()
22
23 window.mainloop()
24

```

图片锚定点位置参数图:



Menubar菜单

```

1 import tkinter as tk
2
3
4 window = tk.Tk(className='window')
5 window.title('my window')
6 window.geometry('300x200+600+300')
7
8 l = tk.Label(window, text='', bg='yellow')
9 l.pack()
10
11 counter = 0
12 def do_job():
13     global counter
14     l.config(text='do' + str(counter))
15     counter += 1
16 menubar = tk.Menu(window)
17 filemenu = tk.Menu(menubar, tearoff=0, borderwidth=5)
18 menubar.add_cascade(label='File', menu=filemenu)
19 filemenu.add_command(label='New', command=do_job)
20 filemenu.add_command(label='Open', command=do_job)

```

```

21 filemenu.add_command(label='Save', command=do_job)
22 filemenu.add_separator()
23 filemenu.add_command(label='Exit', command=window.quit)
24
25 editmenu = tk.Menu(menuubar, tearoff=0)
26 menuubar.add_cascade(label='Edit', menu=editmenu)
27 editmenu.add_command(label='Cut', command=do_job)
28 editmenu.add_command(label='Copy', command=do_job)
29 editmenu.add_command(label='Paste', command=do_job)
30
31 submenu = tk.Menu(filemenu)
32 filemenu.add_cascade(label='Import', menu=submenu, underline=0)
33 submenu.add_command(label='Submenu1', command=do_job)
34
35
36 window.config(menu=menuubar)
37
38 window.mainloop()
39

```

frame框架

```

1 import tkinter as tk
2
3
4 window = tk.Tk(className='window')
5 window.title('my window')
6 window.geometry('200x200')
7
8 l = tk.Label(window, text='on the window', bg='yellow').pack()
9
10 frm = tk.Frame(window)
11 frm.pack()
12 frm_l = tk.Frame(frm,)
13 frm_r = tk.Frame(frm)
14 frm_l.pack(side='left')
15 frm_r.pack(side='right')
16
17 tk.Label(frm_l, text='on the frm_l1').pack()
18 tk.Label(frm_l, text='on the frm_l2').pack()
19 tk.Label(frm_r, text='on the frm_r1').pack()

```

```
20
21 window.mainloop()
22
```

messagebox弹窗

```
1 import tkinter as tk
2 from tkinter import messagebox
3
4
5 window = tk.Tk(className='window')
6 window.title('my window')
7 window.geometry('200x200')
8
9 def hit_me():
10     # messagebox.showinfo(title='Hi', message='hahahaha')
11     # messagebox.showwarning(title='Hi', message='nononono')
12     # messagebox.showerror(title='Hi', message='crash happend')
13     # res = messagebox.askquestion(title='Hi', message='hhahahaha') # return
    es or no
14     # print(res)
15     # res = messagebox.askyesno(title='Hi', message='ask yes no') # return
    True False
16     # print(res)
17     res = messagebox.askretrycancel(title='Hi', message='ask retry cancel')
    # return True False
18     print(res)
19     # res = messagebox.askokcancel(title='Hi', message='ask ok cancel') # r
    eturn True False
20     # print(res)
21
22 tk.Button(window, text='hit me', command=hit_me).pack()
23
24 window.mainloop()
25
```

pack grid place 放置位置

pack使用:

```
1 import tkinter as tk
2 from tkinter import messagebox
3
4
```

```

5 window = tk.Tk(className='window')
6 window.title('my window')
7 window.geometry('200x200')
8
9 tk.Label(window, text=1).pack(side='top')
10 tk.Label(window, text=2).pack(side='bottom')
11 tk.Label(window, text=3).pack(side='left')
12 tk.Label(window, text=4).pack(side='right')
13
14 window.mainloop()
15

```

grid使用:

```

1 import tkinter as tk
2 from tkinter import messagebox
3
4
5 window = tk.Tk(className='window')
6 window.title('my window')
7 window.geometry('200x200')
8
9 for i in range(4):
10     for j in range(3):
11         # tk.Label(window, text=1).grid(row=i, column=j, padx=10 ,pady=10) #
        设置空间周围间距
12         tk.Label(window, text=1).grid(row=i, column=j, ipadx=10 ,ipady=10) #
        设置标签内字体和边距间隔
13
14 window.mainloop()
15

```

place使用:

```

1 import tkinter as tk
2 from tkinter import messagebox
3
4
5 window = tk.Tk(className='window')
6 window.title('my window')
7 window.geometry('200x200')
8
9 tk.Label(window, text=1).place(x=10, y=100, anchor='nw') # north west
10

```

```
11 window.mainloop()
12
```

实例-登录窗口

主窗体:

```
1 import tkinter as tk
2 from tkinter import messagebox
3
4 window = tk.Tk()
5 window.title('Welcom to Mywebsite')
6 window.geometry('450x300')
7
8 canvas = tk.Canvas(window, height=200, width=500)
9 image_file = tk.PhotoImage(file='./bibi.png')
10 image = canvas.create_image(0,0, anchor='nw', image=image_file)
11 canvas.pack(side='top')
12
13 tk.Label(window, text='User Name: ').place(x=50, y=150)
14 tk.Label(window, text='Password: ').place(x=50, y=190)
15 # tk.Label(window, text='User Name: ', font=
16 # ('Arial', 14)).place(relx=0.2, y=160)
17 # tk.Label(window, text='Password: ', font=
18 # ('Arial', 14)).place(relx=0.2, y=190)
19 # tk.Label(window, text='User Name: ', font=('Arial', 12)).grid(row=0, column=0, sticky='w')
20 # tk.Label(window, text='Password: ', font=('Arial', 12)).grid(row=1, column=0, sticky='w')
21
22 var_usr_name = tk.StringVar()
23 var_usr_name.set('exmaple@python.com')
24 entry_usr_name = tk.Entry(window, textvariable=var_usr_name)
25 entry_usr_name.place(x=160,y=150)
26 var_usr_pwd = tk.StringVar()
27 # var_usr_pwd.set('*****')
28 entry_usr_pwd = tk.Entry(window, textvariable=var_usr_pwd, show='*')
29 entry_usr_pwd.place(x=160,y=190)
30
31 def usr_login():
32     pass
33
34 def usr_sign_up():
```



```

32     pass
33
34 btn_login = tk.Button(window, text='Login', command=usr_login)
35 btn_login.place(x=170, y=230)
36 btn_sign_up = tk.Button(window, text='Sign up', command=usr_sign_up)
37 btn_sign_up.place(x=270, y=230)
38 window.mainloop()
39

```

window窗体使用pack布局，添加frame框架，里面使用grid布局，放大缩小窗体用户名密码始终在中间。如果直接在window窗体上使用pack和grid会报错，如果都使用place，窗口放大缩小相对位置会发生变化：

```

1  import tkinter as tk
2  from tkinter import messagebox
3
4  window = tk.Tk(className='Welcome')
5  window.geometry('600x480')
6
7  canvas = tk.Canvas(window, height=200, width=600)
8  image_file = tk.PhotoImage(file='./bibi.png')
9  image = canvas.create_image(0,0, anchor='nw', image=image_file)
10 canvas.pack(side='top')
11
12 frame = tk.Frame(window)
13 frame.pack(side='top')
14
15 tk.Label(frame, text='User Name: ', font=
('Arial', 12)).grid(row=1, column=0, sticky='w')
16 tk.Label(frame, text='Password: ', font=
('Arial', 12)).grid(row=2, column=0, sticky='w')
17
18 var_usr_name = tk.StringVar()
19 var_usr_name.set('example@python.com')
20 entry_usr_name = tk.Entry(frame, textvariable=var_usr_name)
21 entry_usr_name.grid(row=1, column=1, sticky='w')
22 var_usr_pwd = tk.StringVar()
23 var_usr_pwd.set('*****')
24 entry_usr_pwd = tk.Entry(frame, textvariable=var_usr_pwd, show='*')
25 entry_usr_pwd.grid(row=2, column=1, sticky='w')
26
27 window.mainloop()

```

用户登录函数书写：

```

1 import tkinter as tk
2 from tkinter import messagebox
3 import pickle
4
5 window = tk.Tk()
6 window.title('Welcom to Mywebsite')
7 window.geometry('450x300')
8
9 canvas = tk.Canvas(window, height=200, width=500)
10 image_file = tk.PhotoImage(file='./bibi.png')
11 image = canvas.create_image(0,0, anchor='nw', image=image_file)
12 canvas.pack(side='top')
13
14 tk.Label(window, text='User Name: ').place(x=50, y=150)
15 tk.Label(window, text='Password: ').place(x=50, y=190)
16
17 var_usr_name = tk.StringVar()
18 var_usr_name.set('exmaple@python.com')
19 entry_usr_pwd = tk.Entry(window, textvariable=var_usr_name)
20 entry_usr_pwd.place(x=160,y=150)
21 var_usr_pwd = tk.StringVar()
22 # var_usr_pwd.set('*****')
23 entry_usr_pwd = tk.Entry(window, textvariable=var_usr_pwd, show='*')
24 entry_usr_pwd.place(x=160,y=190)
25
26 def usr_login():
27     usr_name = var_usr_name.get()
28     usr_pwd = var_usr_pwd.get()
29     try:
30         with open('usrs_info.pickle', 'rb') as usr_file:
31             usrs_info = pickle.load(usr_file)
32     except FileNotFoundError:
33         with open('usrs_info.pickle', 'wb') as usr_file:
34             usrs_info = {'admin': 'admin'}
35             pickle.dump(usrs_info, usr_file)
36     if usr_name in usrs_info:
37         if usr_pwd == usrs_info[usr_name]:
38             messagebox.showinfo(title='Welcome', message='How are you toda
y ' + usr_name)
39         else:
40             messagebox.showerror(message='Error, your password is wrong,tr
y again.')

```

```

41     else:
42         is_sign_up = messagebox.askyesno('Welcome', 'You have not sign up yet, Sign up today?')
43         if is_sign_up:
44             usr_sign_up()
45
46 def usr_sign_up():
47     pass
48
49 btn_login = tk.Button(window, text='Login', command=usr_login)
50 btn_login.place(x=170, y=230)
51 btn_sign_up = tk.Button(window, text='Sign up', command=usr_sign_up)
52 btn_sign_up.place(x=270, y=230)
53 window.mainloop()
54

```

加上用户注册函数,整体代码如下:

```

1  import tkinter as tk
2  from tkinter import messagebox
3  import pickle
4
5  window = tk.Tk()
6  window.title('Welcom to Mywebsite')
7  window.geometry('450x300')
8
9  canvas = tk.Canvas(window, height=200, width=500)
10 image_file = tk.PhotoImage(file='./bibi.png')
11 image = canvas.create_image(0,0, anchor='nw', image=image_file)
12 canvas.pack(side='top')
13
14 tk.Label(window, text='User Name: ').place(x=50, y=150)
15 tk.Label(window, text='Password: ').place(x=50, y=190)
16
17 var_usr_name = tk.StringVar()
18 var_usr_name.set('exmaple@python.com')
19 entry_usr_pwd = tk.Entry(window, textvariable=var_usr_name)
20 entry_usr_pwd.place(x=160, y=150)
21 var_usr_pwd = tk.StringVar()
22 # var_usr_pwd.set('*****')
23 entry_usr_pwd = tk.Entry(window, textvariable=var_usr_pwd, show='*')
24 entry_usr_pwd.place(x=160, y=190)
25

```

```

26 def usr_login():
27     usr_name = var_usr_name.get()
28     usr_pwd = var_usr_pwd.get()
29     try:
30         with open('usrs_info.pickle', 'rb') as usr_file:
31             usrs_info = pickle.load(usr_file)
32     except FileNotFoundError:
33         with open('usrs_info.pickle', 'wb') as usr_file:
34             usrs_info = {'admin': 'admin'}
35             pickle.dump(usrs_info, usr_file)
36     if usr_name in usrs_info:
37         if usr_pwd == usrs_info[usr_name]:
38             messagebox.showinfo(title='Welcome', message='How are you toda
y ' + usr_name)
39         else:
40             messagebox.showerror(message='Error, your password is wrong, tr
y again.')
41     else:
42         is_sign_up = messagebox.askyesno('Welcome', 'You have not sign up y
et, Sign up today?' )
43         if is_sign_up:
44             usr_sign_up()
45
46 def usr_sign_up():
47     def sign_to_my_python():
48         np = new_pwd.get()
49         npf = new_pwd_confirm.get()
50         nn = new_name.get()
51         with open('usrs_info.pickle', 'rb') as usr_file:
52             exist_usr_info = pickle.load(usr_file)
53         if np != npf:
54             messagebox.showerror('Error', 'Password and confirm password mu
st be the same!')
55         elif nn in exist_usr_info:
56             messagebox.showerror('error', 'The user has been signed up')
57         else:
58             exist_usr_info[nn] = np
59             with open('usrs_info.pickle', 'wb') as usr_file:
60                 pickle.dump(exist_usr_info, usr_file)
61             messagebox.showinfo('Welcom', 'You have signed up successful !')
62             window_sign_up.destroy()
63

```

```

64     window_sign_up = tk.Toplevel(window)
65     window_sign_up.geometry('350x200')
66     window_sign_up.title('Sign up window')
67
68     new_name = tk.StringVar()
69     new_name.set('example@python.com')
70     tk.Label(window_sign_up, text='User name: ').place(x=10, y=10)
71     entry_new_name = tk.Entry(window_sign_up, textvariable=new_name)
72     entry_new_name.place(x=150, y=10)
73
74     new_pwd = tk.StringVar()
75     tk.Label(window_sign_up, text='Password: ').place(x=10, y=50)
76     entry_usr_pwd = tk.Entry(window_sign_up, textvariable=new_pwd, show='*')
77     entry_usr_pwd.place(x=150, y=50)
78
79     new_pwd_confirm = tk.StringVar()
80     tk.Label(window_sign_up, text='Confirm password: ').place(x=10, y=90)
81     entry_usr_pwd_confirm = tk.Entry(window_sign_up, textvariable=new_pwd_confirm, show='*')
82     entry_usr_pwd_confirm.place(x=150, y=90)
83
84     btn_confirm_sign_up = tk.Button(window_sign_up, text='Sign up', command=sign_to_my_python)
85     btn_confirm_sign_up.place(x=150, y=130)
86
87     btn_login = tk.Button(window, text='Login', command=usr_login)
88     btn_login.place(x=170, y=230)
89     btn_sign_up = tk.Button(window, text='Sign up', command=usr_sign_up)
90     btn_sign_up.place(x=270, y=230)
91     window.mainloop()
92

```

Combobox下拉菜单

```

1  import tkinter as tk
2  from tkinter import ttk
3
4
5  window = tk.Tk(className='Combobox test')
6  window.geometry('300x200+600+300')
7
8  var = tk.StringVar()

```

```

9
10 com = ttk.Combobox(window, textvariable=var)
11 com.pack()
12 com['value'] = ('北京', '深圳', '上海')
13 com.current(2)
14
15 def com_select(event):
16     x = com.get()
17     print(var.get(), x)
18
19 com.bind('<<ComboboxSelected>>', com_select)
20
21 window.mainloop()

```

使用grid和combobox, 精确放置位置。设置combobox的宽度即下拉菜单高度:

```

1 import tkinter as tk
2 from tkinter import ttk
3
4 window = tk.Tk(className='Combobox test')
5 window.geometry('600x400+600+300')
6
7 var = tk.StringVar()
8
9 com = ttk.Combobox(window, textvariable=var, width=5, height=3)
10 com.grid(column=0, row=0, ipadx=5, pady=15)
11 com['value'] = ('北京', '深圳', '上海')
12 com.current(2)
13 def com_select(event):
14     x = com.get()
15     print(var.get(), x)
16 com.bind('<<ComboboxSelected>>', com_select)
17
18 def com1_select(event):
19     x = com.get() # com.get()的值也等于var的值, 即var.get()
20     print(var1.get(), x) # var1.get() 设置var1的值
21 var1 = tk.StringVar()
22 com1 = ttk.Combobox(window, textvariable=var1)
23 com1.grid(column=1, row=0, pady=15, padx=10)
24 com1['value'] = ('python', 'java', 'mysql')
25 com1.current(0)
26 com1.bind('<<ComboboxSelected>>', com1_select)
27
28 window.mainloop()

```

右键下拉菜单

```
1 import tkinter as tk
2
3 window = tk.Tk()
4 window.title('Right click test')
5 window.geometry('800x600+600+300')
6
7 def callback():
8     pass
9
10 menubar = tk.Menu(window, tearoff=False)
11 menubar.add_command(label='Undo', command=callback)
12 menubar.add_command(label='Redo', command=callback)
13
14 frame = tk.Frame(window, width=200, height=200) # 定义框架大小
15 frame.pack() # 贴到window窗体的位置, 默认为top
16
17 def popup(event):
18     menubar.post(event.x_root, event.y_root) # 相对menubar的位置弹出
19     # menubar.post(event.x, event.y) # 屏幕的绝对位置弹出
20 frame.bind('<Button-3>', popup) # 只有再窗体内右键才有弹出菜单
21
22 window.mainloop()
```

OptionMenu选项

```
1 import tkinter as tk
2
3 window = tk.Tk(className="Option menu")
4 window.geometry('300x200+600+300')
5
6 lang = ['python', 'PHP', 'Java', 'Go']
7 var = tk.StringVar()
8 var.set('python')
9
10 def printoption(event):
11     print(var.get())
12
13 om = tk.OptionMenu(window, var, *lang)
14 om.bind('<Button-1>', printoption)
```

```
15 om.pack()  
16  
17 window.mainloop()
```

Toplevel顶级窗口

Toplevel 组件是一个独立的顶级窗口，这种窗口通常拥有标题栏、边框等部件。Toplevel 组件和 Tk（根窗口）是一个级别的

```
1 import tkinter as tk  
2  
3 window = tk.Tk()  
4 window.geometry('300x200')  
5  
6 def create_toplevel():  
7     top = tk.Toplevel()  
8     top.title('Toplevel window')  
9     top.geometry('600x300')  
10  
11     msg = tk.Message(top, text='I love python', width=300)  
12     msg.pack()  
13  
14     tk.Label(top, text='User Name: ').pack()  
15     tk.Label(top, text='Password: ').pack()  
16  
17 b = tk.Button(window, text='create window', command=create_toplevel)  
18 b.pack()  
19  
20 window.mainloop()  
21
```

LabelFrame组件

没错，LabelFrame就是Frame的小老弟，它的本质核心也是为了实现一个规划布局。

如果你想将一些相关的组件分为一组，在页面上形成一个区域，与其他组件或区域分隔开来，即可以使用LabelFrame组件。

在默认情况下，LabelFrame 会在其子组件的周围绘制边框和标题。当你想要将一些相关的组件形成一个整体的时候，可以使用 LabelFrame 组件，比如一系列 Radiobutton（单选按钮）组件。

```
1 import tkinter as tk  
2
```



```

3 window = tk.Tk(className='LabelFrame container')
4 window.geometry('300x200+600+300')
5
6 def radiocall():
7     pass
8
9 lb = tk.LabelFrame(window, text='Select Language', padx=10, pady=10, labelan
chor='n')
10 lb.pack()
11
12 v = tk.IntVar()
13 lang = [('python', 1), ('Java', 2), ('shell', 3)]
14 for lang, num in lang:
15     b = tk.Radiobutton(lb, text=lang, variable=v, value=num, command=radioc
all)
16     b.pack(anchor='w')
17
18 window.mainloop()
19

```

Menubutton菜单按钮

该组件适用于你希望菜单按钮出现在其他位置的时候。

创建一个 Menubutton 组件，并创建一个 Menu 组件与之相关联：

```

1 import tkinter as tk
2
3 window = tk.Tk(className='MenuButton')
4 window.geometry('300x200+600+300')
5
6 mb = tk.Menubutton(window, text='菜单', relief='raised')
7 mb.pack()
8
9 def callback():
10     pass
11
12 filemenu = tk.Menu(mb, tearoff=False)
13 filemenu.add_command(label='打开', command=callback)
14 filemenu.add_command(label='保存', command=callback)
15 filemenu.add_separator()
16 filemenu.add_command(label='退出', command=window.quit)
17
18 mb.config(menu=filemenu)
19

```

Message：轻松自动换行的文本显示

Message组件，翻译过来就是消息，通常用于显示多行文本消息。

似乎Message的作用跟Label组件差不多，label只能显示在一行，message可以显示在多行（达到文本消息框宽度自动换行），可以设置长宽比例，对齐方式等等

。在默认情况下，Message组件会按照宽高比150%来进行换行

```

1 import tkinter as tk
2
3 root = tk.Tk()
4
5 # 固定窗口宽度为200，为了将所有文本显示完全高度设置为300
6 root.geometry('200x300')
7
8 word_1='摇摇晃晃进入青春岁月，如此般奢侈的年华，如此般摇曳的时光，\
9 如此般年轻的我们，有着太多美好的时光，有着太多的无奈。'
10
11 word_2=''摇摇晃晃进入青春岁月，
12 如此般奢侈的年华，
13 如此般摇曳的时光，
14 如此般年轻的我们，
15 有着太多美好的时光，
16 有着太多的无奈。''
17
18 l1 = tk.Label(root, text=word_1)
19 l1.pack(padx=10, pady=10)
20
21 l2 = tk.Label(root, text=word_2, justify='left')
22 l2.pack(padx=10, pady=10)
23
24 m1 = tk.Message(root, text=word_1)
25 m1.pack(padx=10, pady=10)
26
27 root.mainloop()
28

```

PanedWindow窗口布局管理器

PanedWindow 组件（Tk8.4 新增）是一个空间管理组件。跟 Frame 组件类似，都是为组件提供一个框架，不过 PanedWindow 允许让用户调整应用程序的空间划分。

何时使用 PanedWindow 组件？

当你需要提供一个可供用户调整的多空间框架的时候，可以使用 PanedWindow 组件。

PanedWindow 组件会为每一个子组件生成一个独立的窗格，用户可以自由调整窗格的大小。

创建一个 2 个窗格的 PanedWindow 组件非常简单：

```
1 import tkinter as tk
2
3 pw = tk.PanedWindow(orient='vertical') # 组件垂直分布
4 pw.pack(fill='both', expand=1)
5
6 top = tk.Label(pw, text='top pane')
7 pw.add(top)
8
9 bottom = tk.Label(pw, text='bottom pane', background='blue')
10 pw.add(bottom)
11
12 tk.mainloop()
```

样式如下：



创建3个窗格：

```
1 import tkinter as tk
2
3 # 默认是左右分布，水平分布；显示手柄和分割线；窗口长宽，子组件add长宽有效，pack无效
4 p1 = tk.PanedWindow(showhandle=True, sashrelief='sunken', height=100, width=100)
5 p1.pack(fill='both', expand=1) # expand:组件分开放平均占用整个空间，等于0则仅挨着放
6
7 left = tk.Label(p1, text='left pane')
8 p1.add(left)
9
10 p2 = tk.PanedWindow(orient='vertical') # 里面的组件垂直分布
11 p1.add(p2)
12
```

```
13 top = tk.Label(p2, text='top pane')
14 p2.add(top)
15
16 bottom = tk.Label(p2, text='bottom pane')
17 p2.add(bottom)
18
19 tk.mainloop()
20
```

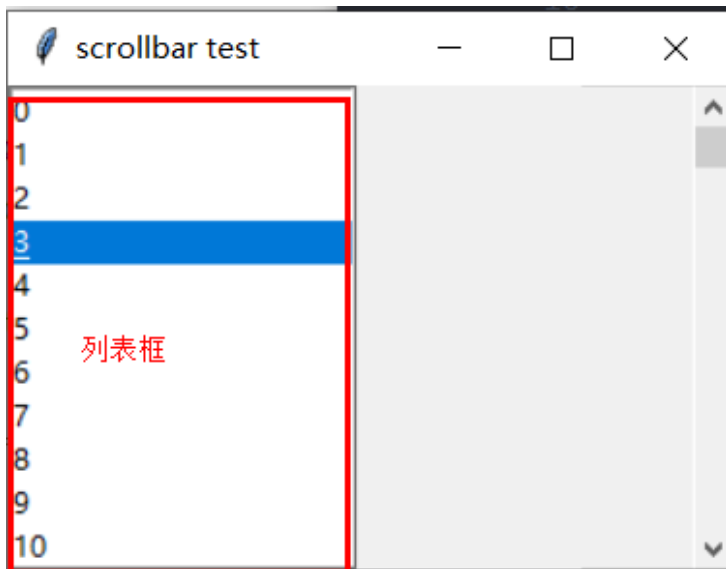
除了以上参数外，还可以设置：颜色、边框、分割是否可调节等等，也可以依附在父窗体上，如tkinter.Tk()

scrollbar滚动条

Scrollbar（滚动条）组件用于滚动一些组件的可见范围，根据方向可分为垂直滚动条和水平滚动条。Scrollbar 组件常常被用于实现文本、画布和列表框的滚动。

列表框上添加垂直滚动条：

```
1 import tkinter as tk
2
3 window = tk.Tk(className='scrollbar test')
4 window.geometry('300x200+600+300')
5
6 sb = tk.Scrollbar(window)
7 sb.pack(side='right', fill='y')
8
9 lb = tk.Listbox(window, yscrollcommand=sb.set)
10
11 for i in range(100):
12     lb.insert('end', str(i))
13
14 lb.pack(side='left', fill='both')
15
16 sb.config(command=lb.yview)
17
18 window.mainloop()
19
```



1. 设置该组件的 `yscrollbarcommand` 选项为 Scrollbar 组件的 `set()` 方法；也就是在列表框中鼠标中键滚动触发 `sb.set()` 方法

2. 设置 Scrollbar 组件的 `command` 选项为该组件的 `yview()` 方法，使用 `sb` 的滚动条会触发 `lb.yview` 方法使得列表浏览窗口发生变化

分析：当 Listbox 组件的可视范围发生改变的时候，Listbox 组件通过调用 `set()` 方法通知 Scrollbar 组件。而当用户操纵滚动条的时候，将自动调用 Listbox 组件的 `yview()` 方法。

添加水平滚动条方法跟上边一样，只是将 `yscrollcommand` 改为 `xscrollcommand`，`yview` 改为 `xview` 即可。

Text添加滚动条

```
1 import tkinter as tk
2
3 window = tk.Tk(className='Text滚动条')
4 window.geometry('300x200+300+200')
5
6 t = tk.Text(window, font=('arial', 10), width=30, height=10)
7 t.pack(side='left')
8
9 sb = tk.Scrollbar(window)
10 sb.pack(side='right', fill='y')
11
12 t.config(yscrollcommand=sb.set)
13 sb.config(command=t.yview)
14
15 str1 = """
16 1. 诸恶莫做, 众善奉行, 莫以善小而不为, 莫以恶小而为之。
17 2. 志不强者智不达。
18 3. 一定要充分利用生活中的闲暇时光, 不要让任何一个发展自我的机会溜走。
```

```

19 4.静坐常思己过,闲谈莫论人非,能受苦乃为志士,肯吃亏不是痴人,敬君子方显有德,怕小人不
   算无能,退一步天高地阔,让三分心平气和,欲...
20 5.人生的道路上多点快乐!多点开心!""
21
22 t.insert('insert', str1)
23
24 window.mainloop()
25

```

spinbox输入控件

Spinbox 组件 (Tk8.4 新增) 是 Entry 组件的变体, 用于从一些固定的值中选取一个。

何时使用 Spinbox 组件?

Spinbox 组件通常用于在限定数字中选取的情况下代替普通的 Entry 组件。

Spinbox 组件跟 Entry 组件用法非常相似, 主要区别是使用 Spinbox 组件, 你可以通过范围或者元组指定允许用户输入的内容。

```

1 import tkinter as tk
2
3 window = tk.Tk(className='spinbox test')
4 window.geometry('300x200+600+300')
5
6 sb = tk.Spinbox(window, from_=0, to=10)
7 sb.pack(side='top')
8
9 window.mainloop()

```

你还可以通过元组指定允许输入的值:

```

1 import tkinter as tk
2
3 window = tk.Tk(className='spinbox test')
4 window.geometry('300x200+600+300')
5
6 sb = tk.Spinbox(window, values=('python', 'java', 'shell', 'mysql'), wrap=True)
7 sb.pack()
8
9 window.mainloop()
10

```

各种控件查用参数参考文档

<https://www.cnblogs.com/aland-1415/p/6849193.html>

Tkinter之pack所有属性详解

<https://blog.csdn.net/zhang07083/article/details/103381397?spm=1001.2014.3001.5501>

Tkinter之place所有属性详解

<https://blog.csdn.net/zhang07083/article/details/103387798?spm=1001.2014.3001.5501>

随窗体伸缩、设置最小窗体、关闭窗口

自由伸缩：

```
1 b = tk.Button(window, text='My Button', height=1, width=10, command=callback)
2 # b.pack()
3 b.place(relx=0.5, rely=0.3, relwidth=0.5, relheight=0.1)
```

设置最小窗体

```
1 window = tk.Tk(className='window')
2 window.title('my window')
3 window.geometry('600x480+300+200')
4 window.minsize(300, 200)
```

关闭窗口

在菜单栏中：window.quit

在函数中：window.destroy()

读取文件和图片、文件上传实现进度

读取文件和图片：

该组件用于保存和打开文件

- askopenfilename() 用于打开文件
- asksaveasfilename() 用于保存文件

以上两个函数的返回值为文件的完整路径

filedialog组件API:

参数 作用

defaulttextension 指定文件后缀，该后缀会自动添加

filetypes 指定筛选文件类型的下拉菜单选项（如：filetypes=[('','PNG' , 'png'),('','GIF' , 'gif')])

initidir 打开/保存文件时的默认路径（默认路径是当前文件夹）

parent 如果想让对话框显示在子窗口w上，可以设置parent=w

title 指定文件对话框的标题栏

```
1 import tkinter as tk
2 from tkinter import filedialog
3 from PIL import Image
4
5 window = tk.Tk(className='Up load file')
6 window.geometry('300x200+600+300')
7 window.withdraw() # 隐藏主窗体
8
9 file_dir = r'D:\VScode_work\test'
10 # filetypes第一个为文件名匹配，第二个元素为后缀匹配
11 file_name = filedialog.askopenfilename(initialdir=file_dir, filetypes=
[('','py txt')], title='Upload File')
12 # filedialog.askopenfilenames 可以选择多个文件
13 with open(file_name, 'rb') as f:
14     print(f.read().decode())
15 # image = Image.open(file_name)
16 # image.show()
17 print(file_name,type(file_name))
18 # D:/VScode_work/test/text.txt <class 'str'>
19
```

文件上传进度条：

```
1 from tkinter import *
2 from tkinter import filedialog, ttk
3 import time
4
5 root = Tk() # 创建一个Tkinter.Tk()实例
6 # root.withdraw() #隐藏窗口
7 root.wm_attributes('-topmost', 1)
8 root.title('消息')
9 root.geometry('600x300+450+200')
10 localName = ''
11
12
13 def up():
14     up.place_forget() # 隐藏部件
15     select.place_forget()
```



```
16     mpb = ttk.Progressbar(root, orient="horizontal", length=200, mode="determinate")
17     mpb.pack()
18     mpb["maximum"] = 100
19     mpb["value"] = 0
20     Label(root, text='正在上传...', fg='red').pack()
21     for i in range(100):
22         mpb["value"] = i + 1
23         root.update()
24         time.sleep(0.1)
25
26
27 def callback():
28     global localName
29     localName = filedialog.askopenfilename(title=u"选择文件")
30     if localName == '':
31         lb.config(text="您没有选择任何文件")
32     else:
33         lb.config(text=localName)
34         select.config(text="重新上传")
35         select.place(anchor=CENTER, x=240, y=60)
36         up.config(text="确认上传", )
37         up.place(anchor=CENTER, x=330, y=60)
38         up.pack_forget()
39
40
41 lb = Label(root, text='')
42 lb.pack()
43 select = Button(root, text='选择文件', command=callback)
44 select.pack()
45 up = Button(root, text='', width=0, height=0, command=up)
46 up.pack_forget() # 如果之前有pack过up, 限制让pack不生效, 让后面的place生效
47 print(localName)
48 root.mainloop()
```