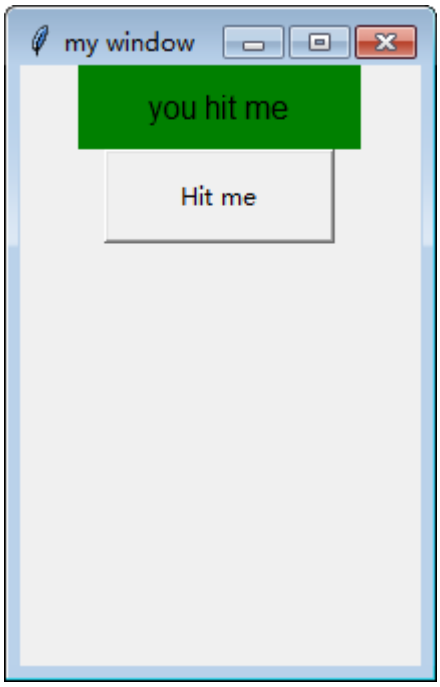
6.使用 TKinter 进行 GUI 程序设计

6.1 Label&Button 标签和按钮

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸
var=tk.StringVar()#字符串变量
var.set('OMG!this is TK!')#变量赋值
l=tk.Label(window,textvariable=var,bg='green',font=('Arial',12),width=15,height=2)#实例化一个标签,父容器是 window
l.pack()#将标签对齐放置
on_hit=False
def hit_me():#点击按钮的事件
    global on_hit#引用全局变量
    if on_hit==False:
        on_hit=True
        var.set('you hit me')
    else:
        on_hit=False
        var.set('')
b=tk.Button(window,text='Hit  me',width=15,height=2,command=hit_me)#实例化一个按钮，点击事件为 hit_me 函数
b.pack()#将标签对齐放置
window.mainloop()#循环显示
```

运行结果：



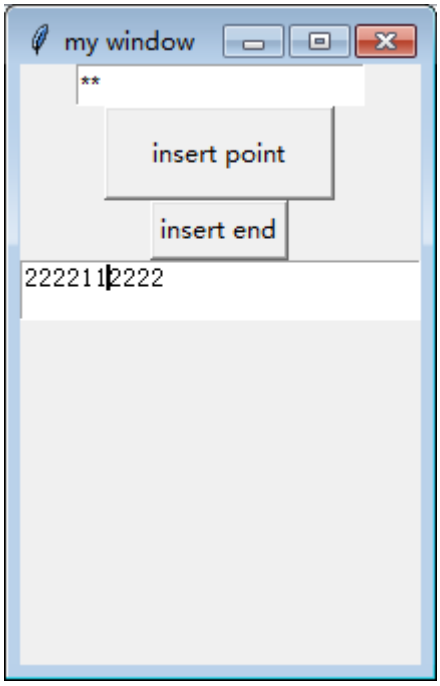
6.2 Entry&Text 输入框和文本框

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸

e=tk.Entry(window,show='*')#实例化一个输入框,显示为*
e.pack()#放置

def insert_point():
    var=e.get()#获取输入框中的内容
    t.insert('insert',var)#在光标位置插入
def insert_end():
    var=e.get()
    t.insert('end',var)#在文本的最后位置插入


b1=tk.Button(window,text='insert point',width=15,height=2,command=insert_point)
b1.pack()#放置


b2=tk.Button(window,text='insert end',command=insert_end)#第二个按钮，默认宽高
b2.pack()#放置
```

```
t=tk.Text(window,height=2)#实例化一个文本框
t.pack()#放置
window.mainloop()#循环显示
```

运行结果：



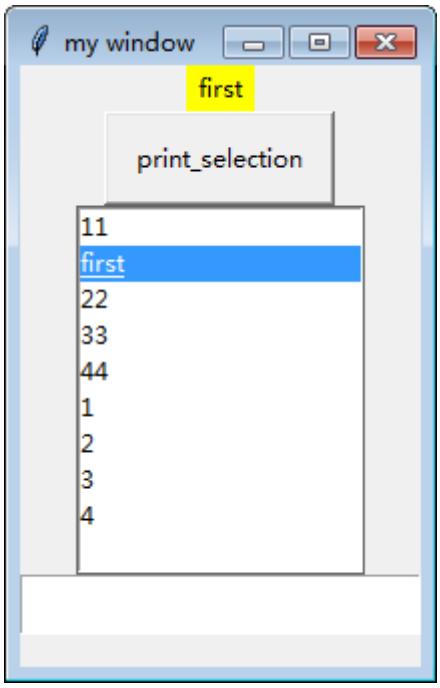
6.3 Listbox 列表

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸

var1 = tk.StringVar()
l=tk.Label(window,bg='yellow',width=4,textvariable=var1)
l.pack()
def print_selection():
    value=lb.get(lb.curselection())#获取列表光标选中的项
    var1.set(value)
#创建按钮
b1=tk.Button(window,text='print_selection',width=15,height=2,command=print_selection)
b1.pack()#放置
#创建列表
var2 = tk.StringVar()
```

```
var2.set((11,22,33,44))#列表初始值
lb=tk.Listbox(window,listvariable=var2)
lb.pack()
#朝列表中添加数据
list_items=[1,2,3,4]
for item in list_items:
        lb.insert('end',item)
lb.insert(1,'first')
lb.insert(2,'second')
lb.delete(2)

t=tk.Text(window,height=2)#实例化一个文本框
t.pack()#放置
window.mainloop()#循环显示
```

运行结果：



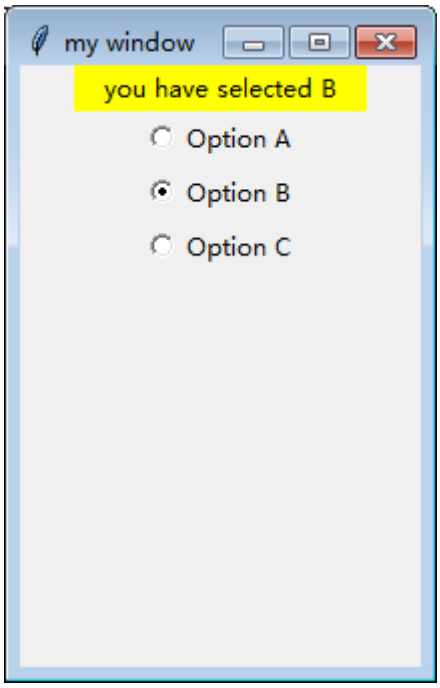
6.4 Radiobutton 单选按钮

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸



l=tk.Label(window,bg='yellow',width=20,text='empty')
```

```
l.pack()


var = tk.StringVar()
def print_selection():
    l.config(text='you have selected '+var.get())
#创建单选按钮
r1=tk.Radiobutton(window,text='Option  A',variable=var,value='A',command=print_selection)# 当按钮选择时，会修改 var 的值，并触发 print_selection 方法
r1.pack()
r2=tk.Radiobutton(window,text='Option B',variable=var,value='B',command=print_selection)
r2.pack()
r3=tk.Radiobutton(window,text='Option C',variable=var,value='C',command=print_selection)
r3.pack()
window.mainloop()#循环显示
```

运行结果：



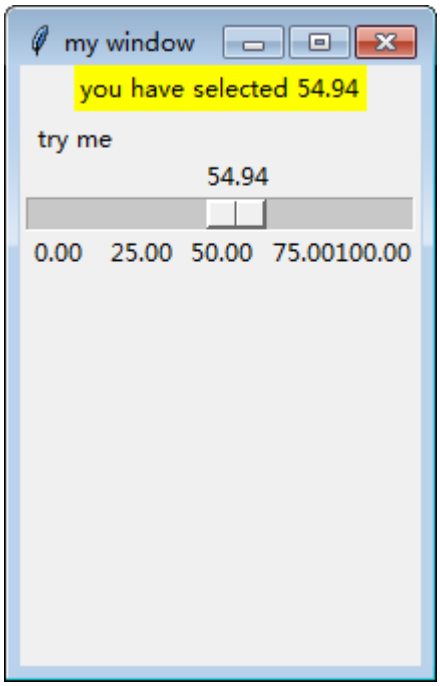
6.5 Scale 滑动条

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸



l=tk.Label(window,bg='yellow',width=20,text='empty')
```

```
l.pack()

def print_selection(v):
    l.config(text='you have selected '+v)
#创建单选按钮
s=tk.Scale(window,label='try
me',from_=0,to=100,orient=tk.HORIZONTAL,length=200,showvalue=1,tickinterval=25,resolution=
0.01,command=print_selection)
s.pack()
window.mainloop()#循环显示
```

运行结果：



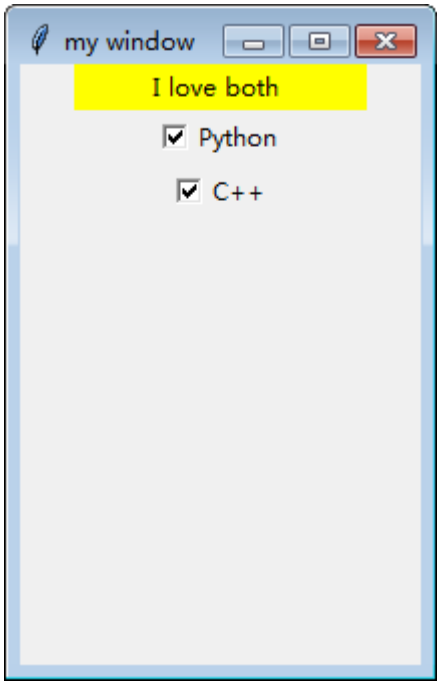
6.6 Checkbutton 多选按钮

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸



l=tk.Label(window,bg='yellow',width=20,text='empty')
l.pack()

def print_selection():
    if(var1.get()==2)&(var2.get()==0):
```

```
            l.config(text='I love only Python ')
        elif(var1.get()==0)&(var2.get()==1):
            l.config(text='I love only C++ ')
        elif(var1.get()==1)&(var2.get()==1):
            l.config(text='I love both ')
        else:
            l.config(text='I do not love either ')

var1=tk.IntVar()
var2=tk.IntVar()
c1=tk.Checkbutton(window,text='Python',variable=var1,onvalue=1,offvalue=0,command=print_s
election)
c2=tk.Checkbutton(window,text='C++',variable=var2,onvalue=1,offvalue=0,command=print_selec
tion)
c1.pack()
c2.pack()

window.mainloop()#循环显示
```

运行结果：



6.7 Canvas 画布

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
```

```
window.geometry('200x300')#窗口的尺寸

#创建画布
canvas=tk.Canvas(window,bg='blue',height=100,width=200)

#绘制图片
image_file=tk.PhotoImage(file='welcome.gif')
image=canvas.create_image(0,0,anchor='nw',image=image_file)



#绘制线、圆、扇形、正方形
x0,y0,x1,y1=50,50,80,80
line=canvas.create_line(x0,y0,x1,y1)
oval=canvas.create_oval(x0,y0,x1,y1,fill='red')
arc=canvas.create_arc(x0+30,y0+30,x1+30,y1+30,start=0,extent=180,fill='yellow')
rect=canvas.create_rectangle(100,30,100+20,30+20,fill='pink')

canvas.pack()
#当按钮点击时，移动正方形
def moveit():
    canvas.move(rect,0,2)
b=tk.Button(window,text='move',command=moveit)
b.pack()
window.mainloop()#循环显示
```

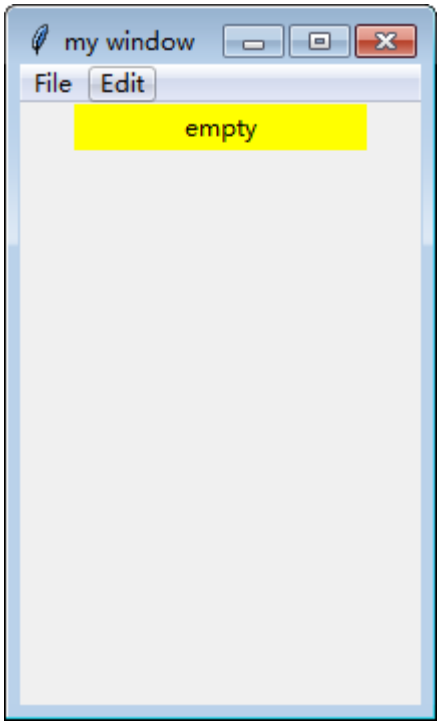
运行结果：

6.8 Menubar 菜单

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸

l=tk.Label(window,bg='yellow',width=20,text='empty')
l.pack()
counter=0
def do_job():
    global counter
    l.config(text='do'+str(counter))
    counter+=1
#添加菜单
menubar=tk.Menu(window)#添加菜单容器
filemenu=tk.Menu(menubar,tearoff=0)#添加一级子菜单
editmenu=tk.Menu(menubar,tearoff=0)#添加一级子菜单
submenu=tk.Menu(filemenu,tearoff=0)#添加二级子菜单
menubar.add_cascade(label='File',menu=filemenu)#添加子菜单的标签
menubar.add_cascade(label='Edit',menu=editmenu)#添加子菜单的标签
#filemenu
filemenu.add_command(label='New',command=do_job)#添加命令
filemenu.add_command(label='Open',command=do_job)#添加命令
filemenu.add_command(label='Save',command=do_job)#添加命令
filemenu.add_separator()#添加分割线
filemenu.add_command(label='Exit',command=window.quit)#添加命令
filemenu.add_cascade(label='Import',menu=submenu,underline=0)#添加子菜单的标签
#editmenu
editmenu.add_command(label='Cut',command=do_job)#添加命令
editmenu.add_command(label='Copy',command=do_job)#添加命令
editmenu.add_command(label='Paste',command=do_job)#添加命令
#submenu
submenu.add_command(label='Submenu1',command=do_job)#添加命令
window.config(menu=menubar)#使用菜单
window.mainloop()#循环显示
```

运行结果：

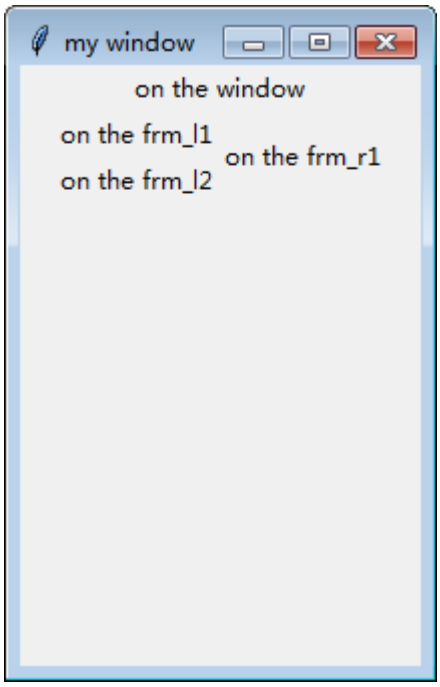


6.9 Frame 框架

使用示例：

```
import tkinter as tk
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸

tk.Label(window,text='on the window').pack()

frm=tk.Frame(window)#新建一个框架
frm.pack()#放置

frm_l=tk.Frame(frm)#新建子框架
frm_r=tk.Frame(frm)
frm_l.pack(side='left')
frm_r.pack(side='right')

tk.Label(frm_l,text='on the frm_l1').pack()
tk.Label(frm_l,text='on the frm_l2').pack()
tk.Label(frm_r,text='on the frm_r1').pack()

window.mainloop()#循环显示
```

运行结果：



6.10 Messagebox 弹窗

使用示例：

```
import tkinter as tk
from tkinter import messagebox
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸
def hit_me():
    #消息型弹窗
    tk.messagebox.showinfo(title='showinfo',message='aaaaaa')#消息弹窗
    tk.messagebox.showwarning(title='showwarning',message='bbbbbb')#警告弹窗
    tk.messagebox.showerror(title='showerror',message='ccccc')#错误弹窗
    #互动型弹出窗
    print(tk.messagebox.askquestion(title='askquestion'))#return 'yes','no'
    print(tk.messagebox.askyesno(title='askyesno'))#return True,False
    print(tk.messagebox.askokcancel(title='askokcancel'))#return True,False
    print(tk.messagebox.askretrycancel(title='askretrycancel'))#return True,False
    print(tk.messagebox.askyesnocancel(title='askyesnocancel'))#return True,False,None
tk.Button(window,text='hit me',command=hit_me).pack()

window.mainloop()#循环显示
```

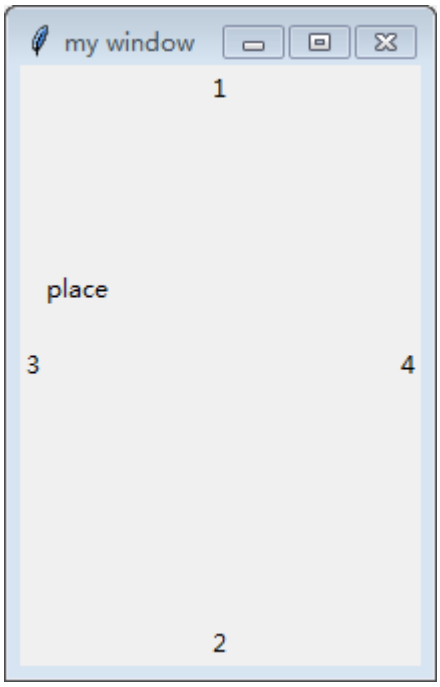
运行结果：

6.11 pack grid place 放置位置

使用示例：

```
import tkinter as tk
from tkinter import messagebox
window=tk.Tk()#实例化窗口
window.title('my window')#窗口的标题
window.geometry('200x300')#窗口的尺寸

#pack
tk.Label(window,text='1').pack(side='top')
tk.Label(window,text='2').pack(side='bottom')
tk.Label(window,text='3').pack(side='left')
tk.Label(window,text='4').pack(side='right')
#grid
'''for i in range(4):
    for j in range(3):
        tk.Label(window,text='grid').grid(row=i,column=j,ipadx=10,ipady=10)'''
#place
tk.Label(window,text='place').place(x=10,y=100,anchor='nw')

window.mainloop()#循环显示
```

运行结果：


my window
1
place
3
4
2