

GUI编程

BBCLOUD 林凡

https://github.com/lvancer/course_python

lin029011@163.com

大纲

GUI

Tkinter

练习



GUI

GUI全称是**图像用户界面**，相对于命令行，GUI使用起来就很方便了。

主流的GUI有：

- 1、**Tkinter**：Python自带，轻量级，一些简单的界面可以快速开发。
- 2、**wxPython**：中级，专业的小工具可以使用。
- 3、**PyQt5**：比较专业的开发工具，借助C++语言开发的Qt5框架，加了一层Python壳。有大量的工具简化开发，如界面可以直接通过拖动完成。

下面主要介绍Tkinter，了解基本的GUI编写方法。

Tkinter

第一个GUI界面

```
import tkinter as tk          # Tk库

root = tk.Tk()                # 创建Tk对象
root.title("My First GUI")    # 界面标题
root.geometry('400x300')      # 界面大小
root.mainloop()              # 启动界面
```

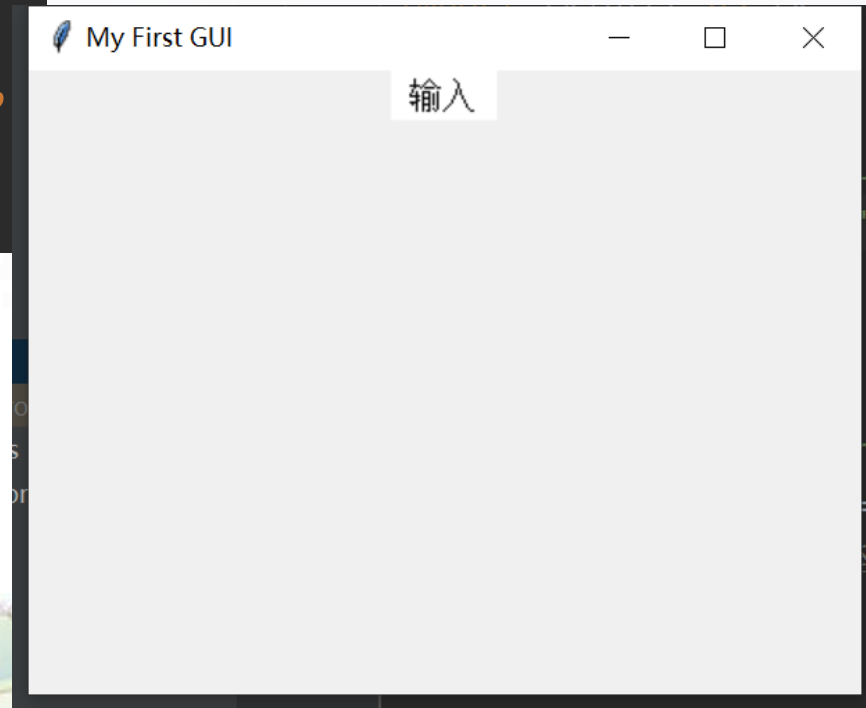
这就是一个空白的界面，下面我们开始添加控件。
所有的界面代码都在`mainloop`方法前。



Tkinter

Label：标签文字。

```
label = tk.Label(root, text="输入",  
                 bg="white", font=("Arial", 12),  
                 width=5, height=1)  
label.pack()      # 添加到界面
```



第一个参数是根节点root，所有控件通用。

text：现实的文本，Label核心参数。

bg：背景颜色【可不填，控件通用】。

font：字体【可不填，控件通用】。

width，**height**：宽度和高度，不是像素为单位【可不填，控件通用】。

Tkinter

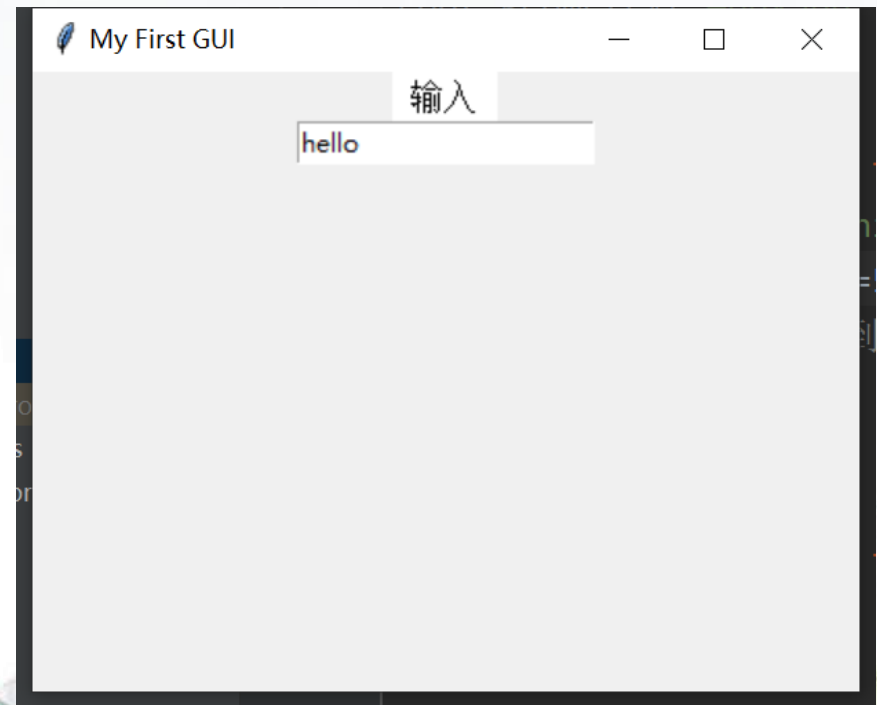
Entry：单行文本输入框。

```
var = tk.StringVar()      # 用于存储内容的变量
entry = tk.Entry(root, textvariable=var)
var.set("hello")          # 设置输入框内容
print(var.get())          # 获取输入框内容
entry.pack()
```

textvariable：要绑定的存储变量。

StringVar是tk的字符串变量，用于给控件进行变量存储。

绑定到控件后，使用**set**，**get**方法进行设置与获取。



Tkinter

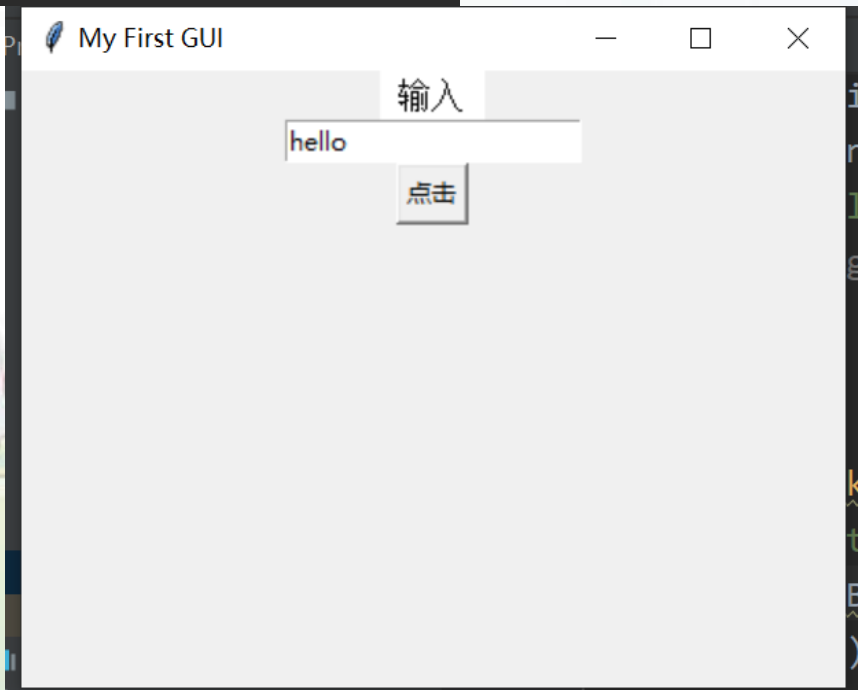
Button：按钮。

```
def btn_click():                                # 按钮点击触发方法
    print('btn_click')
button = tk.Button(root, text='点击', command=btn_click)
button.pack()
```

text：按钮文字。

command：按钮绑定的点击事件。

点击事件就是一个定义好的方法。



Tkinter

tkMessageBox：消息弹窗。

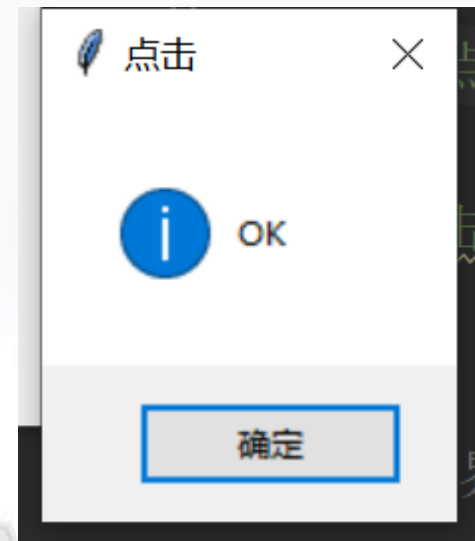
```
import tkinter.messagebox  
tkinter.messagebox.showinfo('点击', 'OK')
```

参数1为消息标题。

参数2为消息内容。

放入按钮点击事件中

```
def btn_click():  
    tkinter.messagebox.showinfo('点击', 'OK')
```



Tkinter

Frame：容器。

可以放入其他控件。

frame以**root**为根节点，
其他几个控件都以**frame**
为根节点。

这样就可以进行封装。

side：排列方式。

```
frame = tk.Frame(root)

label = tk.Label(frame, text="输入")
label.pack(side=tk.LEFT)

var = tk.StringVar()
entry = tk.Entry(frame, textvariable=var)
entry.pack(side=tk.LEFT)

def btn_click():
    tkinter.messagebox.showinfo('点击', 'OK')

button = tk.Button(frame, text='点击', command=btn_click)
button.pack(side=tk.RIGHT)

frame.pack(side=tk.TOP)
```

Tkinter

```
class MyFrame(tk.Frame):  
  
    def __init__(self, master):  
        tk.Frame.__init__(self, master)  
  
        self.label = tk.Label(self, text="输入")  
        self.label.pack(side=tk.LEFT)  
        self.var = tk.StringVar()  
        self.entry = tk.Entry(self, textvariable=self.var)  
        self.entry.pack(side=tk.LEFT)  
        self.button = tk.Button(self, text='点击', command=self.btn_click)  
        self.button.pack(side=tk.RIGHT)  
  
    def btn_click(self):  
        tkinter.messagebox.showinfo('输入内容', self.var.get())
```

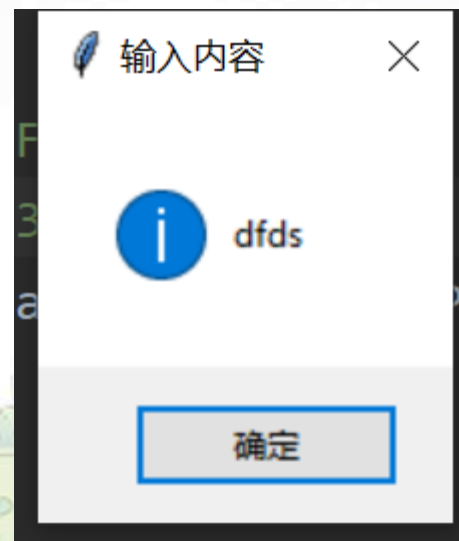
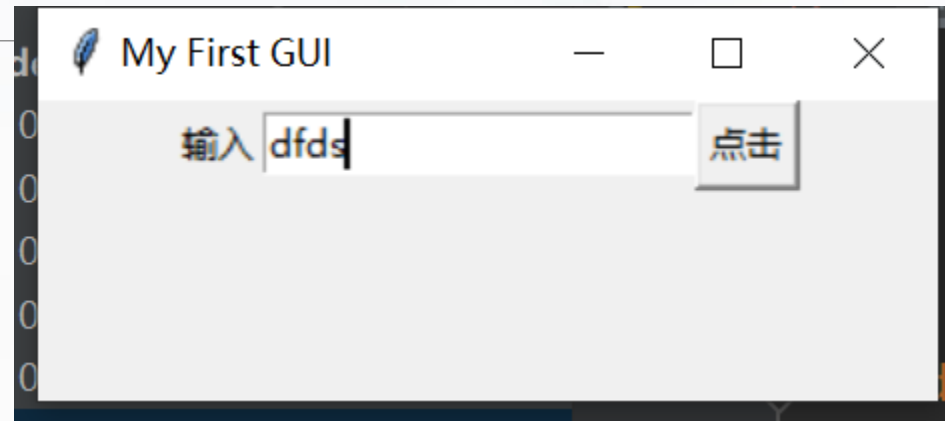
Tkinter

嵌入root中，执行。

```
if __name__ == '__main__':  
    root = tk.Tk()  
    root.title("My First GUI")  
    root.geometry('300x100')  
    MyFrame(root).pack(side=tk.TOP)  
    root.mainloop()
```

推荐封装类进行界面编写，方便扩展。

PyQt也是通过类进行界面编写的。



Tkinter

grid：网格布局，与pack对应。

```
self.label.grid(column=0, row=0, sticky=tk.W)  
self.entry.grid(column=0, row=1, sticky=tk.W)  
self.button.grid(column=1, row=2, sticky=tk.W)
```

label 被放在了第0列，第0行。

entry被放在了第0列，第1行。

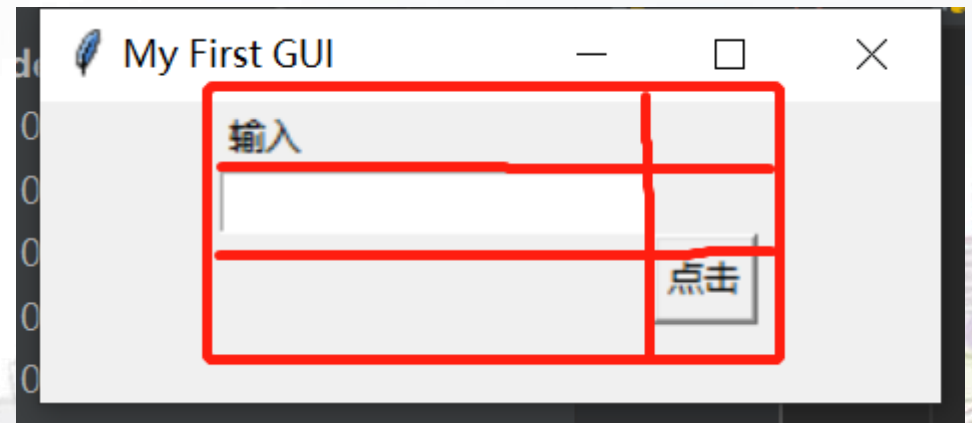
button被放在了第1列，第2行。

column：列

row：行

sticky：内容对齐方式。N【上对齐】，S【下对齐】，W【左对齐】，E【右对齐】

lin029011@163.com



Tkinter

Checkbutton：选择框。

```
def checkbutton_select(x):    #  
    print(x.get())  
check = tk.IntVar()  
checkbutton = tk.Checkbutton(root, text="选择框", variable=check,  
                             command=lambda : checkbutton_select(check))  
checkbutton.pack()
```

text：文字。

variable：绑定的变量。

state：设定为disabled时，不可操作。

command：绑定的命令。这里使用了lambda表达式来传递参数。

将check变量传递到checkbutton_select中去。



Tkinter

制作一个多选框：



```
self.check1 = tk.IntVar()
self.checkbutton1 = tk.Checkbutton(self, text="安卓", variable=self.check1)
self.checkbutton1.grid(column=0, row=0, sticky=tk.W)
self.check2 = tk.IntVar()
self.checkbutton2 = tk.Checkbutton(self, text="苹果", variable=self.check2)
self.checkbutton2.grid(column=1, row=0, sticky=tk.W)
self.check3 = tk.IntVar()
self.checkbutton3 = tk.Checkbutton(self, text="PC", variable=self.check3,
                                   state='disabled')
self.checkbutton3.select() # 选择
self.checkbutton3.grid(column=2, row=0, sticky=tk.W)
```

尝试做一个全选按钮。

Tkinter

Radiobutton：单选框。



```
self.value = tk.IntVar()
options = [('中国', 0), ('美国', 1), ('日本', 2), ('韩国', 3)] # 选项
for t, v in options:
    tk.Radiobutton(self, text=t, value=v, variable=self.value,
                   command=self.radio_select).pack()
```

text：显示文字。

value：值。通过循环options生成所有的选项。

variable：绑定的变量。所有radiobutton都绑定同一个变量。

command：绑定的选择方法。所有radiobutton都绑定同一个事件方法。

事件方法中，通过绑定的变量就能获得当前选择的值。

Tkinter

Listbox：列表。

```
self.listbox = tk.Listbox(self, selectmode=tk.SINGLE)
for item in range(10):
    self.listbox.insert(tk.END, item)
self.listbox.pack(side=tk.LEFT)
```

selectmode：选择模式。**SINGLE**、**BROWSE**【单选】，**MULTIPLE**、**EXTENDED**【多选】

insert方法用于添加项，第一个参数是添加位置。

添加一个删除按钮，点击后删除选中项。

```
tk.Button(self, text='删除', command=self.delete).pack(side=tk.TOP)

def delete(self):
    self.listbox.delete('active') # 删除选中项
```

练习

1、编写一个简易计算器。

