

TK 教程

一、导入 tkinter 模块

```
from tkinter import *
```

二、创建 GUI 根窗体

```
root = Tk() # 创建名为 root 的主窗体

root.title('JupiterRobot 功能演示') # 为窗体命名

root.geometry('580x320') # 这里的乘号不是 *，而是小写英文字母 x

root.mainloop() #主窗体循环，一般写在所有代码的最后一行，保证运行代码时窗口持续存在
```

```
1 from tkinter import *
2
3 root = Tk() # 创建名为root的主窗体
4 root.title('JupiterRobot功能演示') # 为窗体命名
5 root.geometry('580x320') # 这里的乘号不是 *，而是小写英文字母 x
6 root.mainloop() #主窗体循环，一般写在所有代码的最后一行，保证运行代码时窗口持续存在
7
8
```



三、添加控件并编写相应的函数

1. 常用控件

(1) Label 标签

```
lb = Label(root, text='我是标签')
```

lb.pack() # 用于显示控件

```
from tkinter import *

root = Tk() # 创建名为root的主窗体
root.title('JupiterRobot功能演示') # 为窗体命名
root.geometry('580x320') # 这里的乘号不是 * , 而是小写英文字母 x

lb = Label(root, text='我是标签')
lb.pack() # 用于显示控件

root.mainloop() #主窗体循环, 一般写在所有代码最后
```



括号内第一个 root 是将控件放置在 'root' 界面, 第二个 text 就是标签控件要显示的标签内容, 这两个选项是最常用的。还有一些别的选项比如:

bg='#d3fbfb' #背景颜色

fg='blue' #字体颜色

font=('华文新魏',16) # 字体和大小

width=18

height=2

relief=SUNKEN # 属性 relief 为控件呈现出来的 3D 浮雕样式, 有 FLAT(平的)、RAISED(凸起的)、SUNKEN(凹陷的)、GROOVE(沟槽状边缘) 和 RIDGE(脊状边缘) 5 种。

(2) Button 按钮

```
btn = Button(root, text='我是按钮', font=('黑体',14),
command=cmd)

btn.pack()
```


```
from tkinter import *

root = Tk() # 创建名为root的主窗体
root.title('JupiterRobot功能演示') # 为窗体命名
root.geometry('580x320') # 这里的乘号不是 * , 而是小写英文字母 x

def cmd():
    pass

btn = Button(root, text='我是按钮', font=('黑体',14), command=cmd)
btn.pack()

root.mainloop() #主窗体循环
```



按钮可以用 `command=cmd` 的方法绑定函数

(3) Entry 输入框

```
inp = Entry(root)

inp.pack()
```

```
from tkinter import *

root = Tk() # 创建名为root的主窗体
root.title('JupiterRobot功能演示') # 为窗体命名
root.geometry('580x320') # 这里的乘号不是 * , 而是小写英文字母 x

inp = Entry(root)
inp.pack()

root.mainloop() #主窗体循环
```



(4) Notebook 切换界面

使用 Notebook 需要导入 `tkinter.ttk`

```
import tkinter.ttk
```

先创建 Notebook 实例，再用 `frame()` 加入分页

```
note = tkinter.ttk.Notebook(root)
```

```

tb1 = Frame() # 分页 1

note.add(tb1, text='我是分页 1')

tb2 = Frame() # 分页 2

note.add(tb2, text='我是分页 2')

note.pack(padx=0, pady=0, fill=BOTH, expand=True)

```

```

from tkinter import *
import tkinter.ttk
root = Tk() # 创建名为root的主窗体
root.title('JupiterRobot功能演示') # 为窗体命名
root.geometry('580x320') # 这里的乘号不是 * , 而是小写英文字母 x

note = tkinter.ttk.Notebook(root)
tb1 = Frame() # 分页1
note.add(tb1, text='我是分页1')
tb2 = Frame() # 分页2
note.add(tb2, text='我是分页2')
note.pack(padx=0, pady=0, fill=BOTH, expand=True)

```

```

root.mainloop() #主

```



想要在分页中加入控件只需要将第一个参数改成分页名即可，如 tb1

2. .place()方法

如果控件较多，使用.pack()方法无法按照自己的意愿放置控件，这时候就需要将.pack()换成.place()方法

根据控件实例在父容器中的绝对或相对位置参数进行布局。其常用布局参数如下：

x,y: 控件实例在根窗体中水平和垂直方向上的其实位置（单位为像素）。注意，根窗体左上角为 0,0,水平向右，垂直向下为正方向。

relx,rely: 控件实例在根窗体中水平和垂直方向上起始布局的相对位置。即相对于根窗体宽和高的比例位置，取值在 0.0~1.0 之间。

height,width: 控件实例本身的高度和宽度（单位为像素）。

relheight,relwidth: 控件实例相对于根窗体的高度和宽度比例，取值在 0.0~1.0 之间。

利用 place()方法配合 relx,rely 和 relheight,relwidth 参数所得的到的界面可自适应根窗体尺寸的大小。

```
btn.place(relx=0.1, rely=0.1, relheight=0.2 , relwidth=0.3
```

5) # 例如这个就是将按钮放置在窗体横向 10%，纵向 10%的位置，并且按钮款为窗体的 35%，高为窗体的 20%

```
from tkinter import *
import tkinter.ttk
root = Tk() # 创建名为root的主窗体
root.title('JupiterRobot功能演示') # 为窗体命名
root.geometry('580x320') # 这里的乘号不是 * ，而是小写英文字母 x

btn = Button(root, text='我是按钮')
btn.place(relx=0.1, rely=0.1, relheight=0.2 , relwidth=0.35)

root.mainloop() #
```



3. 编写函数

```
def start():

    cmd1 = 'roscore'

    os.system(f"gnome-terminal --window -e 'bash -c
\"{cmd1};exec bash\"")
```

编写好函数后在按钮或其他控件中的 `command` 参数中加入函数名，即可用控件启动函数

```
btn = Button(root, text='roscore', font=('黑体',14),
command=start)
```