

dummies.com

Using Tkinter Widgets in Python

By Brendan Scott

Tkinter in Python comes with a lot of good widgets. *Widgets* are standard graphical user interface (GUI) elements, like different kinds of buttons and menus. Most of the Tkinter widgets are given here.

Label Widget



A Label widget shows text to the user. You can update the widget programmatically to, for example, provide a readout or status bar.

```
import Tkinter
parent_widget = Tkinter.Tk()
label_widget = Tkinter.Label(parent_widget,
text="A Label")
label_widget.pack()
Tkinter.mainloop()
```

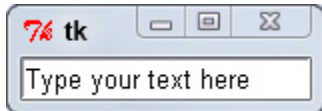
Button Widget



A Button can be on and off. When a user clicks it, the button emits an event. Images can be displayed on buttons.

```
import Tkinter
parent_widget = Tkinter.Tk()
button_widget = Tkinter.Button(parent_widget,
                                text="A Button")
button_widget.pack()
Tkinter.mainloop()
```

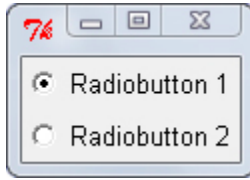
Entry Widget



An Entry widget gets text input from the user.

```
import Tkinter
parent_widget = Tkinter.Tk()
entry_widget = Tkinter.Entry(parent_widget)
entry_widget.insert(0, "Type your text here")
entry_widget.pack()
Tkinter.mainloop()
```

Radiobutton Widget



A Radiobutton lets you put buttons together, so that only one of them can be clicked. If one button is on and the user clicks another, the first is set to off. Use Tkinter variables (mainly Tkinter.IntVar and Tkinter.StringVar) to access its state.

```
import Tkinter
parent_widget = Tkinter.Tk()
v = Tkinter.IntVar()
v.set(1) # need to use v.set and v.get to
# set and get the value of this variable
radiobutton_widget1 =
Tkinter.Radiobutton(parent_widget,

text="Radiobutton 1",

variable=v,

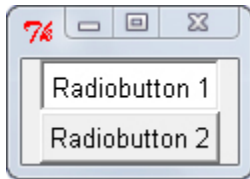
value=1)
radiobutton_widget2 =
Tkinter.Radiobutton(parent_widget,

text="Radiobutton 2",

variable=v,
```

```
value=2)
radiobutton_widget1.pack()
radiobutton_widget2.pack()
Tkinter.mainloop()
```

Radiobutton Widget (Alternate)



You can display a Radiobutton without the dot indicator. In that case it displays its state by being sunken or raised.

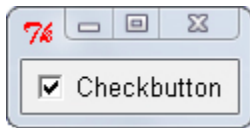
```
import Tkinter
parent_widget = Tkinter.Tk()
v = Tkinter.IntVar()
v.set(1)
radiobutton_widget1 =
Tkinter.Radiobutton(parent_widget,

text="Radiobutton 1",
                                variable=v,
value=1,

indicatoron=False)
radiobutton_widget2 =
```

```
Tkinter.Radiobutton(parent_widget,  
  
text="Radiobutton 2",  
  
variable=v,  
  
value=2,  
  
indicatoron=False)  
radiobutton_widget1.pack()  
radiobutton_widget2.pack()  
Tkinter.mainloop()
```

Checkbutton Widget

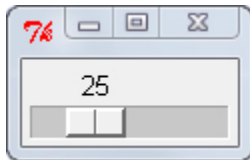


A Checkbutton records on/off or true/false status. Like a Radiobutton, a Checkbutton widget can be displayed without its check mark, and you need to use a Tkinter variable to access its state.

```
import Tkinter  
parent_widget = Tkinter.Tk()  
checkbutton_widget =  
Tkinter.Checkbutton(parent_widget,  
  
text="Checkbutton")
```

```
checkboxbutton_widget.select()  
checkboxbutton_widget.pack()  
Tkinter.mainloop()
```

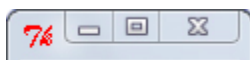
Scale Widget: Horizontal

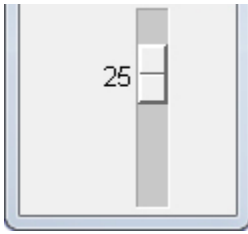


Use a Scale widget when you want a slider that goes from one value to another. You can set the start and end values, as well as the step. For example, you can have a slider that has only the even values between 2 and 100. Access its current value by its `get` method; set its current value by its `set` method.

```
import Tkinter  
parent_widget = Tkinter.Tk()  
scale_widget = Tkinter.Scale(parent_widget,  
    from_=0, to=100,  
  
    orient=Tkinter.HORIZONTAL)  
scale_widget.set(25)  
scale_widget.pack()  
Tkinter.mainloop()
```

Scale Widget: Vertical



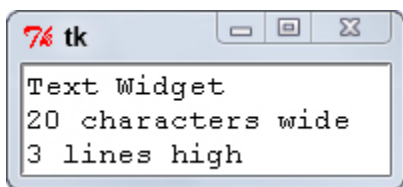


A Scale widget can be *vertical* (up and down).

```
import Tkinter
parent_widget = Tkinter.Tk()
scale_widget = Tkinter.Scale(parent_widget,
                              from_=0, to=100,

                              orient=Tkinter.VERTICAL)
scale_widget.set(25)
scale_widget.pack()
Tkinter.mainloop()
```

Text Widget

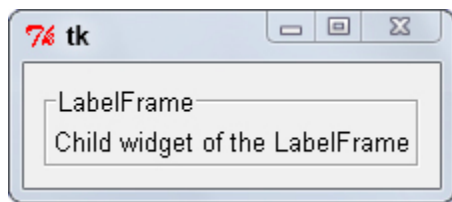


Use a Text widget to show large areas of text. The Text widget lets the user edit and search.

```
import Tkinter
parent_widget = Tkinter.Tk()
```

```
text_widget = Tkinter.Text(parent_widget,  
                             width=20, height=3)  
text_widget.insert(Tkinter.END,  
    "Text Widget\n20 characters wide\n3 lines  
high")  
text_widget.pack()  
Tkinter.mainloop()
```

LabelFrame Widget



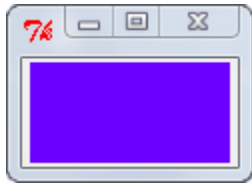
The LabelFrame acts as a parent widget for other widgets, displaying them with a title and an outline. LabelFrame has to have a child widget before you can see it.

```
import Tkinter  
parent_widget = Tkinter.Tk()  
labelframe_widget =  
Tkinter.LabelFrame(parent_widget,  
  
text="LabelFrame")  
label_widget=Tkinter.Label(labelframe_widget,  
    text="Child widget of the LabelFrame")  
labelframe_widget.pack(padx=10, pady=10)
```



```
label_widget.pack()  
Tkinter.mainloop()
```

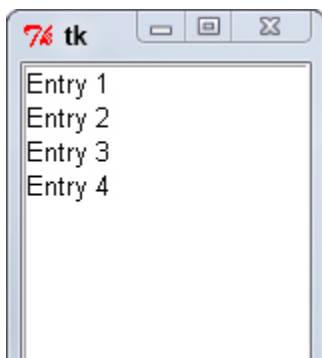
Canvas Widget



You use a Canvas widget to draw on. It supports different drawing methods.

```
import Tkinter  
parent_widget = Tkinter.Tk()  
canvas_widget = Tkinter.Canvas(parent_widget,  
                                bg="blue",  
                                width=100,  
                                height= 50)  
  
canvas_widget.pack()  
Tkinter.mainloop()
```

Listbox Widget

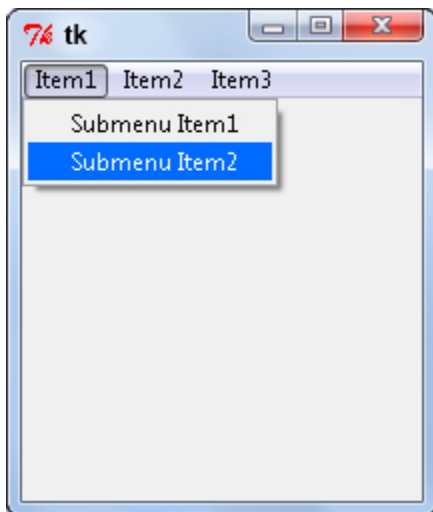




Listbox lets the user choose from one set of options or displays a list of items.

```
import Tkinter
parent_widget = Tkinter.Tk()
listbox_entries = ["Entry 1", "Entry 2",
                  "Entry 3", "Entry 4"]
listbox_widget = Tkinter.Listbox(parent_widget)
for entry in listbox_entries:
    listbox_widget.insert(Tkinter.END, entry)
listbox_widget.pack()
Tkinter.mainloop()
```

Menu Widget



The Menu widget can create a menu bar. Creating menus can be

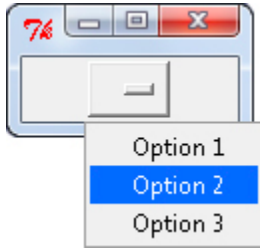
hard, especially if you want drop-down menus. To do that, you use a separate Menu widget for each drop-down menu you're creating.

```
import Tkinter
parent_widget = Tkinter.Tk()
def menu_callback():
    print("I'm in the menu callback!")
def submenu_callback():
    print("I'm in the submenu callback!")
menu_widget = Tkinter.Menu(parent_widget)
submenu_widget = Tkinter.Menu(menu_widget,
tearoff=False)
submenu_widget.add_command(label="Submenu Item1",

command=submenu_callback)
submenu_widget.add_command(label="Submenu Item2",

command=submenu_callback)
menu_widget.add_cascade(label="Item1",
menu=submenu_widget)
menu_widget.add_command(label="Item2",
                        command=menu_callback)
menu_widget.add_command(label="Item3",
                        command=menu_callback)
parent_widget.config(menu=menu_widget)
Tkinter.mainloop()
```

OptionMenu Widget



The OptionMenu widget lets the user choose from a list of options. To use the OptionMenu the right way, you'll probably need to bind it to a separate callback that updates other information based on the user's selection. Get the currently selected value with its `get` method.

```
import Tkinter
parent_widget = Tkinter.Tk()
control_variable =
Tkinter.StringVar(parent_widget)
OPTION_TUPLE = ("Option 1", "Option 2", "Option
3")
optionmenu_widget =
Tkinter.OptionMenu(parent_widget,
                    control_variable,
                    *OPTION_TUPLE)

optionmenu_widget.pack()
Tkinter.mainloop()
```