



## "التجربة الثامنة"

### بناء واجهات رسومية باستخدام لغة python

#### 1-مقدمة:

يمكن الاستفادة من بناء الواجهات الرسومية Graphic User Interface باستخدام لغة python للتحكم بأقطاب GPIO ضمن لوحة Raspberry pi، وهي مدعومة من قبل كل أنظمة تشغيل هذه اللوحة، وتعد من أهم ميزاتها.

#### 2- مكتبة TKinter:

- هي أهم عنصر لبناء الواجهات في python .
- سميت كذلك اختصاراً للعبارة التالية Tool Kit interface.
- تحتوي الأدوات الأساسية لبناء الواجهة وطرق وتوابع وعناصر البناء والتحكم مثل القوائم والأزرار والشريط واللافتات وغيرها. وجميعاً يمكن ربطها مع أقطاب GPIO وفق الخوارزمية المطلوبة لتنفيذها.
- يختلف اسم المكتبة في إصدار python2 ( **TKinter** ) عن python3 ( **tkinter** ) .
- لتحميل المكتبة من سطر الأوامر:  
Sudo apt-get install python-tk
- قائمة بالعناصر التي تحتويها هذه المكتبة:

1. button
2. canvas
3. checkbutton
4. combobox
5. entry
6. frame
7. label
8. labelframe
9. listbox
10. menu
11. menubutton
12. message
13. notebook
14. tk\_optionMenu
15. panedwindow
16. progressbar
17. radiobutton
18. scale
19. scrollbar
20. separator
21. sizegrip
22. spinbox
23. text
24. treeview



### 3- أداة TK:

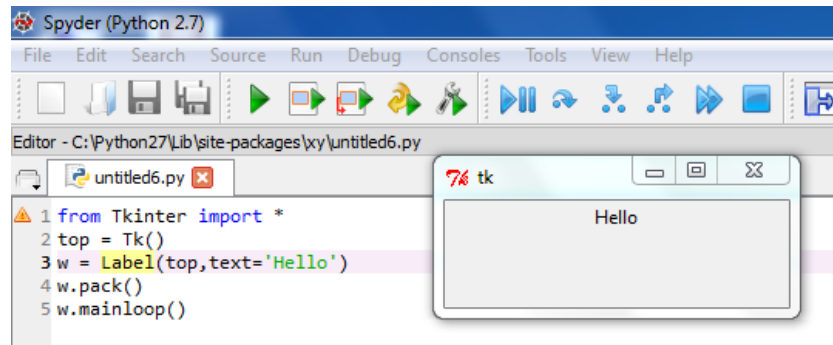
هي الأداة الأساسية لتصميم GUI، وهي عبارة عن class من مكتبة Tool Kit interface، يستخدم لتصميم كائن الواجهة (Top level window object)

```
Import Tkinter  
Top = Tkinter.Tk()
```

### 4- تطبيقات:

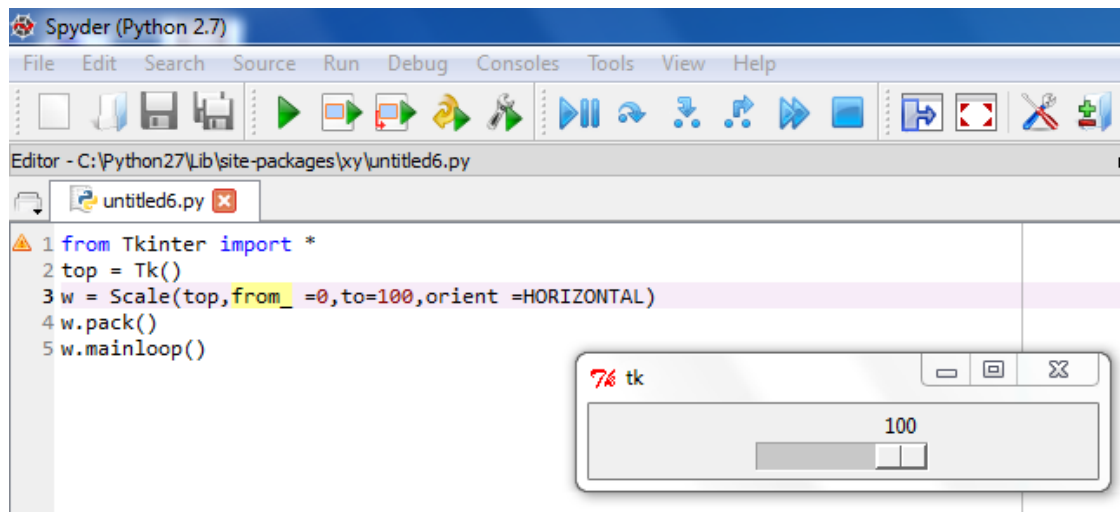
#### تطبيق 1:

```
from Tkinter import *  
top = Tk()  
w = Label(top, text='Hello')  
w.pack()  
w.mainloop()
```



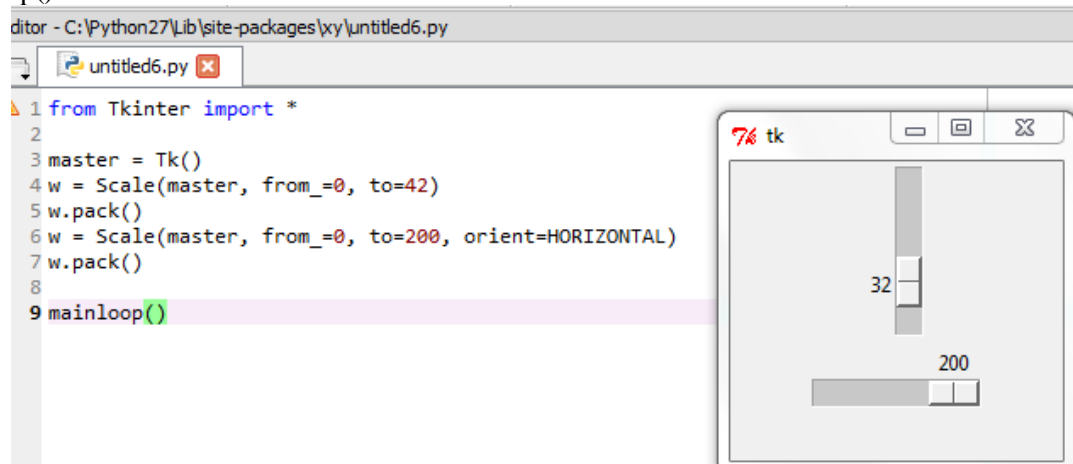
#### تطبيق 2:

```
from Tkinter import*  
top = Tk()  
w = Scale(top,from_ =0,to=100,orient =HORIZONTAL)  
w.pack()  
w.mainloop()
```



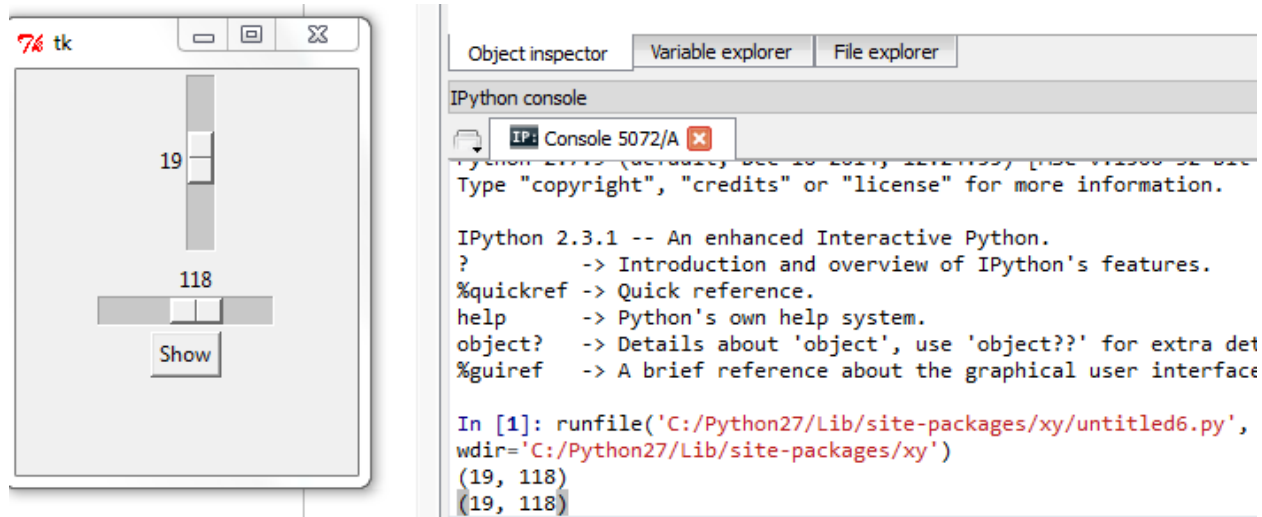
تطبيق 3:

```
from Tkinter import *
master = Tk()
w = Scale(master, from_=0, to=42)
w.pack()
w = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w.pack()
mainloop()
```



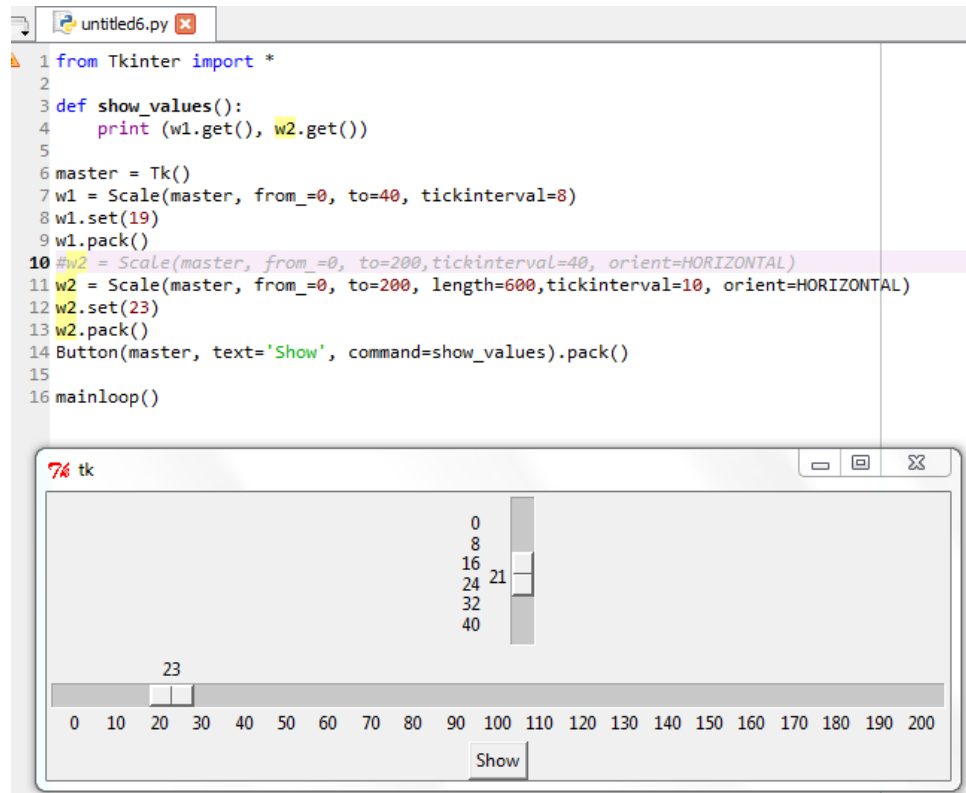
تطبيق 4:

```
from Tkinter import *
def show_values():
    print (w1.get(), w2.get())
master = Tk()
w1 = Scale(master, from_=0, to=42)
w1.set(19)
w1.pack()
w2 = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w2.set(23)
w2.pack()
Button(master, text='Show', command=show_values).pack()
mainloop()
```



## تطبيق 5:

```
from Tkinter import *
def show_values():
    print (w1.get(), w2.get())
master = Tk()
w1 = Scale(master, from_=0, to=40, tickinterval=8)
w1.set(19)
w1.pack()
#w2 = Scale(master, from_=0, to=200, tickinterval=40, orient=HORIZONTAL)
w2 = Scale(master, from_=0, to=200, length=600, tickinterval=10, orient=HORIZONTAL)
w2.set(23)
w2.pack()
Button(master, text='Show', command=show_values).pack()
mainloop()
```



## تطبيق 6:

```
from Tkinter import*
from tkMessageBox import*
def answer():
    showerror("Answer", "Sorry, no answer available")
```

```
def callback():
    if askyesno('Verify', 'Really quit:'):
        showwarning('Yes', 'Not yet implemented')
    else:
        showinfo('No', 'Quit has been cancelled')
```

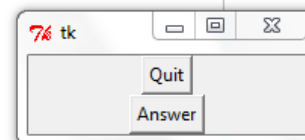
```
Button(text='Quit', command=callback).pack()
Button(text='Answer', command=answer).pack()
mainloop()
```

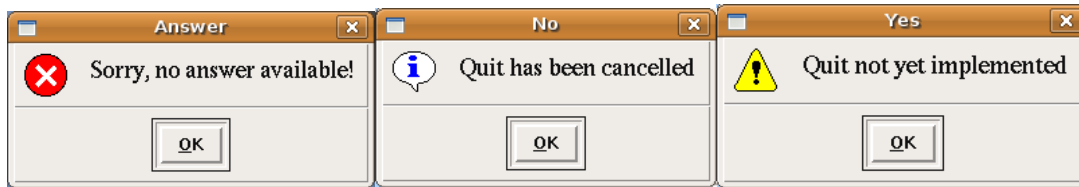
```
from Tkinter import *
from tkMessageBox import *

def answer():
    showerror("Answer", "Sorry, no answer available")

def callback():
    if askyesno('Verify', 'Really quit?'):
        showwarning('Yes', 'Not yet implemented')
    else:
        showinfo('No', 'Quit has been cancelled')

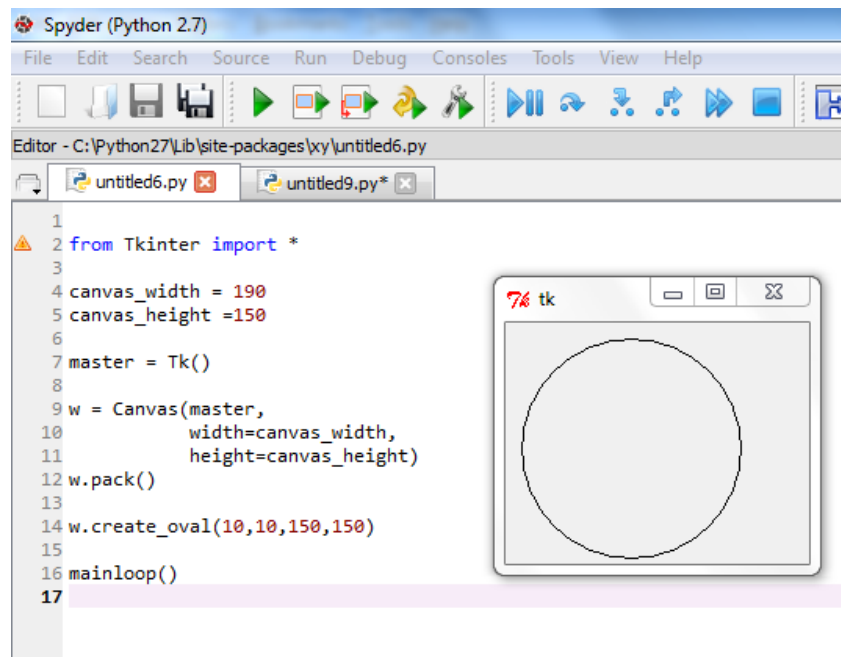
Button(text='Quit', command=callback).pack()
Button(text='Answer', command=answer).pack()
mainloop()
```





### تطبيق 7:

```
from Tkinter import *
canvas_width = 190
canvas_height = 150
master = Tk()
w = Canvas(master,
width=canvas_width,
height=canvas_height)
w.pack()
w.create_oval(10,10,150,150)
mainloop()
```



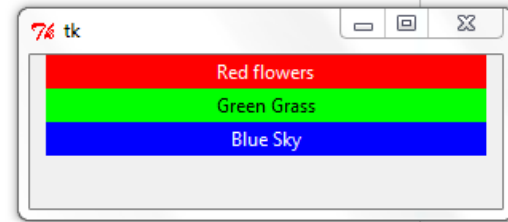
### تطبيق 8:

```
from Tkinter import *
root = Tk()
w = Label(root, text="Red flowers", bg="red", fg="white")
w.pack(fill=X,padx=10)
w = Label(root, text="Green Grass", bg="green", fg="black")
w.pack(fill=X,padx=10)
w = Label(root, text="Blue Sky", bg="blue", fg="white")
w.pack(fill=X,padx=10)
```



mainloop()

```
7
8 from Tkinter import *
9 root = Tk()
10 w = Label(root, text="Red flowers", bg="red", fg="white")
11 w.pack(fill=X, padx=10)
12 w = Label(root, text="Green Grass", bg="green", fg="black")
13 w.pack(fill=X, padx=10)
14 w = Label(root, text="Blue Sky", bg="blue", fg="white")
15 w.pack(fill=X, padx=10)
16 mainloop()
```



وظيفة 1: ما نتائج تنفيذ البرنامج التالي:

```
from Tkinter import *
colours = ['red','green','orange','white','yellow','blue']
r = 0
for c in colours:
    Label(text=c, relief=RIDGE,width=15).grid(row=r,column=0)
    Entry(bg=c, relief=SUNKEN,width=10).grid(row=r,column=1)
    r = r + 1
mainloop()
```

وظيفة 2:

تصميم واجهة تحتوي شريط منزلق من 0 حتى 100 للتحكم بشدة إضاءة LED، وذلك بالاستفادة من التعليمات المذكورة سابقاً، وممايلي:

```
w = Scale(master, from_=0, to=100, orient=HORIZONTAL,command =update)
def update(duty)
p.ChangeDutyCycle(float(duty))
```

ملاحظة: التابع update يأخذ قيمة duty الآنية من الشريط المنزلق ويغير قيمة DutyCycle وفقاً.

المراجع:

<http://www.python-course.eu>