

"التجربة الثامنة"

بناء واجمات رسومية باستخدام لغة python

1-مقدمة:

يمكن الاستفادة من بناء الواجهات الرسومية Graphic User Interface باستخدام لغة python للتحكم بأقطاب ضمن الاستفادة من بناء الواجهات الرسومية Raspberry pi ضمن لوحة GPIO ضمن لوحة المعامن لوحة المعامن لوحة المعامن لوحة المعامن المعامن

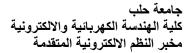
2- مكتبة TKinter:

- هي أهم عنصر لبناء الواجهات في python .
- سميت كذلك اختصارا للعبارة التالية Tool Kit interface.
- تحتوي الأدوات الأساسية لبناء الواجهة وطرق وتوابع وعناصر البناء والتحكم مثل القوائم والأزرار والشريط واللافتات وغيرها. وجميعا يمكن ربطها مع أقطاب GPIO وفق الخوارزمية المطلوب تنفيذها.
 - يختلف اسم المكتبة في إصدار TKinter) python2 عن Tkinter) . (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



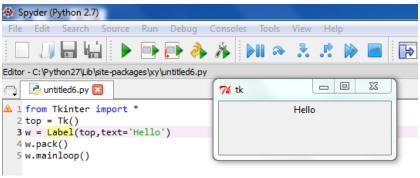


5- أداة TK: هي الأداة الأساسية لتصميم GUI، وهي عبارة عن class من مكتبة Tool Kit interface، يستخدم لتصميم كائن (Top level window object) الواجهة

Import Tkinter Top = Tkinter.Tk()

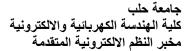
تطبيق 1:

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

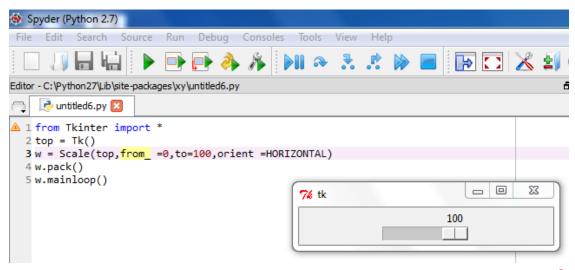


تطبيق 2:

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

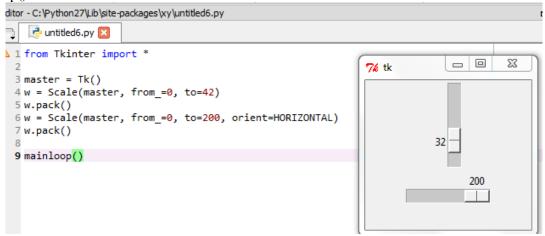






تطبيق 3:

```
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()
```



تطبيق4:

```
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()

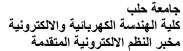
w2 = Scale(master, from_=0, to=200, orient=HORIZONTAL)

w2.set(23)

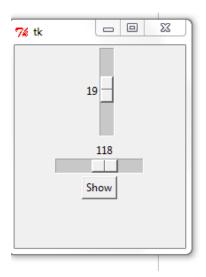
w2.pack()

Button(master, text='Show', command=show_values).pack()

mainloop()
```







```
Variable explorer
                                File explorer
 Object inspector
IPython console
    III Console 5072/A 🔀
Type "copyright", "credits" or "license" for more information.
IPython 2.3.1 -- An enhanced Interactive Python.
           -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
         -> Python's own help system.
help
object? -> Details about 'object', use 'object??' for extra det
%guiref -> A brief reference about the graphical user interface
In [1]: runfile('C:/Python27/Lib/site-packages/xy/untitled6.py',
wdir='C:/Python27/Lib/site-packages/xy')
(19, 118)
```

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()
```

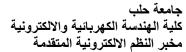


from Tkinter import*
from tkMessageBox import*

```
🔁 untitled6.py 🔀
1 from Tkinter import *
 3 def show_values():
      print (w1.get(), w2.get())
6 master = Tk()
7 w1 = Scale(master, from_=0, to=40, tickinterval=8)
9 w1.pack()
10 #w2 = Scale(master, from_=0, to=200,tickinterval=40, orient=HORIZONTAL)
11 w2 = Scale(master, from_=0, to=200, length=600,tickinterval=10, orient=HORIZONTAL)
12 w2.set(23)
13 w2.pack()
14 Button(master, text='Show', command=show_values).pack()
16 mainloop()
                                                                            _ 0
                                                                                     23
 7% tk
                                           0
                                           8
                                          16
                                          24 21
32
                                          40
             23
       10
           20
              30
                   40
                       50 60 70 80
                                        90 100 110 120 130 140 150 160 170 180 190 200
                                           Show
```

تطبيق 6:

```
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")
                                                                                                   _ 0
                                                                                                               \Sigma S
                                                                                      7% tk
                 def callback():
                     if askyesno('Verify', 'Really quit?'):
    showwarning('Yes', 'Not yet implemented')
                                                                                                   Quit
                                                                                                 Answer
                          showinfo('No', 'Quit has been cancelled')
                Button(text='Quit', command=callback).pack()
Button(text='Answer', command=answer).pack()
                 mainloop()
```

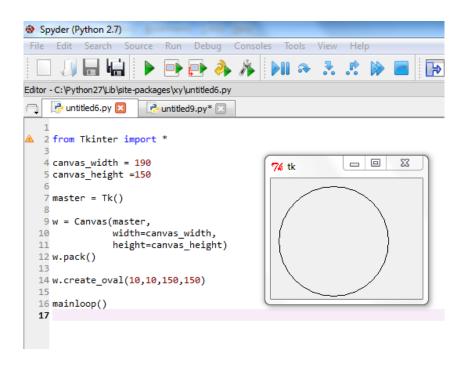






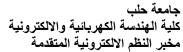
تطبيق 7:

```
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)
```



المراجع:



```
mainloop()
           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()
                                                                                     _ 0
                                                         7% tk
                                                                         Red flowers
                                                                         Green Grass
                                                                           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()
   تصميم واجهة تحتوي شريط منزلق من 0 حتى 100 للتحكم بشدة إضاءة LED، وذلك بالاستفادة من التعليمات المذكورة
                                                                                                سابقا، وممايلي:
w = Scale(master, from_=0, to=100, orient=HORIZONTAL,command =update)
def update(duty)
p.ChangeDutyCycle(float(duty))
                    ملحظة: التابع update يأخذ قيمة duty الآنية من الشريط المنزلق ويغير قيمة DutyCycle وفقها.
```

http://www.python-course.eu