
16. ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΜΕ TKINTER II

16.0.0 Λύσεις προηγούμενων ασκήσεων

1.

'''

Γράψτε ένα προγραμματάκι το οποίο να έχει τρία πεδία εισαγωγής:
Όνομα, User ID, και Password, καθώς κι ένα κουμπί εισαγωγής,
"ΥΠΟΒΟΛΗ".

'''

```
import tkinter as tk
```

```
root = tk.Tk()
```

```
root.geometry("400x250")
```

```
name = tk.Label(root, text = "Όνομα").place(x = 30, y = 50)
```

```
email = tk.Label(root, text = "User ID").place(x = 30, y = 90)
```

```
password = tk.Label(root, text = "Password").place(x = 30, y = 130)
```

```
sbmitbtn = tk.Button(root, text = "ΥΠΟΒΟΛΗ", activebackground =  
"green", activeforeground = "blue").place(x = 120, y = 170)
```

```
entry1 = tk.Entry(root).place(x = 85, y = 50)
```

```
entry2 = tk.Entry(root).place(x = 85, y = 90)
```

```
entry3 = tk.Entry(root).place(x = 90, y = 130)
```

```
root.mainloop()
```

2.

'''

Γράψτε ένα προγραμματάκι χρησιμοποιώντας το tkinter module, το οποίο πρέπει να δημιουργεί ένα checkbox με κείμενο:

"Τσεκάρισμα όταν είναι True". Χρησιμοποιείστε μια μεταβλητή boolean για να διευκολυνθείτε.

'''

```
import tkinter as tk
```

```
from tkinter import ttk
```

```
root = tk.Tk()
```

```
my_boolean_var = tk.BooleanVar()
```

```
my_checkbox = ttk.Checkbutton(  
    text="Τσεκάρισμα όταν είναι True",  
    variable=my_boolean_var  
)
```

```
my_checkbox.pack()
```

```
root.mainloop()
```

16.0.1 Σημείωση:

Το ttk είναι ένα module το οποίο παρουσιάστηκε μετά το tkinter 8.5 στην προσπάθεια να ξεχωρίσει τη λειτουργικότητα των παραθύρων από την εμφάνισή τους. Πρέπει να χρησιμοποιείται με προσοχή, καθώς δεν υπάρχει πλήρης συμβατότητα με τις παραμέτρους και τις μεθόδους του tk. Δείτε [εδώ](#) την αντίστοιχη βιβλιογραφία της Python.

16.0 Tkinter II

Έχουμε δει ότι το Tkinter είναι η τυπική βιβλιοθήκη GUI για την Python. Η Python όταν συνδυάζεται με το Tkinter παρέχει έναν γρήγορο και εύκολο τρόπο δημιουργίας εφαρμογών GUI. Το Tkinter είναι μια ισχυρή αντικειμενοστραφής διεπαφή στην εργαλειοθήκη Tk GUI.

Σε γενικές γραμμές, για να δημιουργήσουμε ένα γραφικό περιβάλλον με το Tkinter, κάνουμε τα παρακάτω βήματα

- Κάνουμε `import tkinter` .
- Δημιουργούμε το κύριο παράθυρο της εφαρμογής GUI.
- Προσθέτουμε ένα ή περισσότερα από γραφικά στοιχεία στην εφαρμογή.
- Εισαγάγουμε τον κύριο βρόχο συμβάντος ώστε πλέον η διαπαφή/εφαρμογή μας να μπορεί να «ακούει» τον χρήστη.

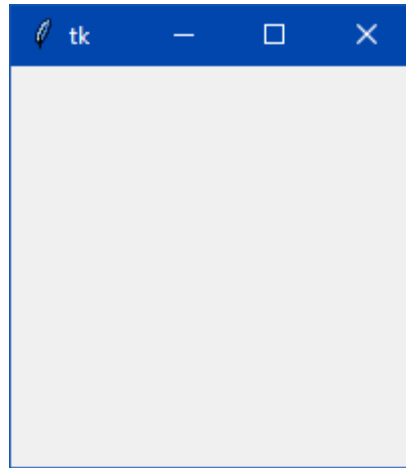
Υπενθυμίζουμε ένα απλό παράδειγμα γραφικής διεπαφής το οποίο μπορούμε να τρέξουμε και γραμμή-γραμμή στον IDLE:

```
>>> import tkinter
```

```
>>> top = tkinter.Tk()
```

```
>>> top.mainloop()
```

Σαν έξοδο έχουμε το παράθυρο:



16.1 Γραφικά στοιχεία – widgets του Tkinter

Το Tkinter μας παρέχει διάφορα στοιχεία ελέγχου, όπως κουμπιά, ετικέτες και πλαίσια κειμένου που χρησιμοποιούνται σε μια εφαρμογή γραφικού περιβάλλοντος. Αυτά τα στοιχεία ελέγχου ονομάζονται συνήθως widgets (γραφικά στοιχεία).

Αυτή τη στιγμή υπάρχουν 15 τύποι γραφικών στοιχείων στο Tkinter.

Ας δούμε αυτά τα widgets καθώς και μια σύντομη περιγραφή στον παρακάτω πίνακα:

Γραφικά στοιχεία περιγραφές και κώδικας

16.1.1 Button - Κουμπί

Το γραφικό στοιχείο Button χρησιμοποιείται για την εμφάνιση των κουμπιών στην εφαρμογή μας.

Παράδειγμα:

Το Button widget χρησιμοποιείται για την προσθήκη κουμπιών σε ένα παράθυρο. Κάθε κουμπί μπορεί να εμφανίζει κείμενο ή εικόνες που μεταφέρει το σκοπό των κουμπιών. Μπορούμε να συνδέσουμε μια μέθοδο σε ένα κουμπί που καλείται όταν κάνουμε κλικ στο κουμπί.

Σύνταξη

w = Button (master, option = value, ...)

Παράμετροι

- **master** – Το γονικό παράθυρο.
- **Επιλογές** – Ακολουθεί η λίστα με τις πιο συχνά χρησιμοποιούμενες επιλογές για αυτό το γραφικό στοιχείο. Αυτές οι επιλογές μπορούν να χρησιμοποιηθούν ως ζεύγη κλειδιού-τιμής που χωρίζονται με κόμμα.

1 *activebackground*

Χρώμα φόντου όταν το κουμπί βρίσκεται κάτω από τον κέρσορα.

2 *activeforeground*

Χρώμα προσκηνίου όταν το κουμπί βρίσκεται κάτω από τον κέρσορα.

3 *bd*

Πλάτος περιγράμματος σε pixel. Η προεπιλογή είναι 2.

4 *bg*

Κανονικό χρώμα φόντου.

5 *command*

Λειτουργία ή μέθοδος που θα κληθεί όταν κάνετε κλικ στο κουμπί.

6 *fg*

Κανονικό χρώμα προσκηνίου (κειμένου).

7 *font*

Γραμματοσειρά κειμένου που θα χρησιμοποιηθεί για την ετικέτα του κουμπιού.

8 *height*

Ύψος του κουμπιού σε γραμμές κειμένου (για κουμπιά κειμένου) ή εικονοστοιχεία (για εικόνες).

Πλήρη σειρά επιλογών, καθώς και συχνά χρησιμοποιούμενων μεθόδων με το χειριστήριο, μπορούμε να βρούμε στην τεκμηρίωση της Python.

Ας δούμε ένα παράδειγμα δημιουργίας κουμπιού:

```
from tkinter import *
```

```
from tkinter import messagebox
```

```
# Δημιουργία αντικειμένου
```

```
top = Tk()
```

```
top.geometry("100x100")
```

```
#Δημιουργία μεθόδου η οποία καλείται με το πάτημα του  
κουμπιού
```

```
def helloCallBack():
```

```
msg = messagebox.showinfo( "Hello Python", "Hello World")
```

Δημιουργία του αντικειμένου κουμπιού

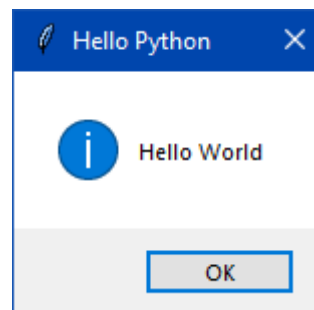
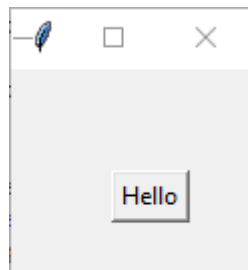
`B = Button(top, text = "Hello", command = helloCallBack)`

Τοποθέτηση του κουμπιού

`B.place(x = 50,y = 50)`

`top.mainloop()`

Το αποτέλεσμα είναι το πρώτο παραθυράκι και πατώντας το κουμπί «Hello», προκύπτει το δεύτερο παράθυρο (δεξιά)



16.1.2 Canvas - Καμβάς

Το widget Canvas χρησιμοποιείται για τη σχεδίαση σχημάτων, όπως γραμμές, οβάλ, πολύγωνα και ορθογώνια.

Σύνταξη

`w = Canvas (master, option = value, ...)`

Παράμετροι

- **master** – Το γονικό παράθυρο.

- **Επιλογές** – Ακολουθεί η λίστα με τις πιο συχνά χρησιμοποιούμενες επιλογές για αυτό το γραφικό στοιχείο. Αυτές οι επιλογές μπορούν να χρησιμοποιηθούν ως ζεύγη κλειδιού-τιμής που χωρίζονται με κόμμα.

1 *bd*

Πλάτος περιγράμματος σε pixel. Η προεπιλογή είναι 2.

2 *bg*

Κανονικό χρώμα φόντου.

5 *confine*

Αν είναι true (default), ο καμβάς δεν μπορεί να βγει έξω από την περιοχή του scrolling.

6 *scrollregion*

Ένα tuple (w,n,e,s) το οποίο καθορίζει πόσο μεγάλη περιοχή του καμβά μπορεί να γίνει scrolled: w = αριστερή πλευρά, n = η πάνω πλευρά, e = η αριστερή πλευρά και s = η κάτω πλευρά.

7 *width*

Το μέγεθος του καμβά στον άξονα των x.

Πλήρη σειρά επιλογών, καθώς και συχνά χρησιμοποιούμενων μεθόδων με το χειριστήριο, μπορούμε να βρούμε στην τεκμηρίωση της Python.

Επίσης υποστηρίζονται τα παρακάτω ενδιαφέροντα αντικείμενα:

arc . Δημιουργεί ένα τόξο το οποίο μπορεί να είναι χορδή ή μια απλή καμπύλη.

```
coord = 10, 50, 240, 210
```

```
arc = canvas.create_arc(coord, start = 0, extent = 150, fill = "blue")
```

image . Δημιουργεί ένα αντικείμενο εικόνας, το οποίο μπορεί να είναι ένα στιγμιότυπο της κλάσης `BitmapImage` ή της `PhotoImage`.

```
filename = PhotoImage(file = "sunshine.gif")
```

```
image = canvas.create_image(50, 50, anchor = NE, image = filename)
```

line . Δημιουργεί μια γραμμή.

```
line = canvas.create_line(x0, y0, x1, y1, ..., xn, yn, options)
```

oval . Δημιουργεί έναν κύκλο ή μια έλλειψη στις δεδομένες συντεταγμένες. Παίρνει δύο ζεύγη συντεταγμένων. Την επάνω αριστερή και την κάτω δεξιά γωνία του τετραγώνου που εμπεριέχει το οβάλ.

```
oval = canvas.create_oval(x0, y0, x1, y1, options)
```

polygon . Creates a polygon item that must have at least three vertices.

```
oval = canvas.create_polygon(x0, y0, x1, y1,...xn, yn, options)
```

Ας δούμε ένα παράδειγμα δημιουργίας καμβά:

```
from tkinter import *
```

```
from tkinter import messagebox
```

```
top = Tk()
```

```
C = Canvas(top, bg = "blue", height = 250, width = 300)
```

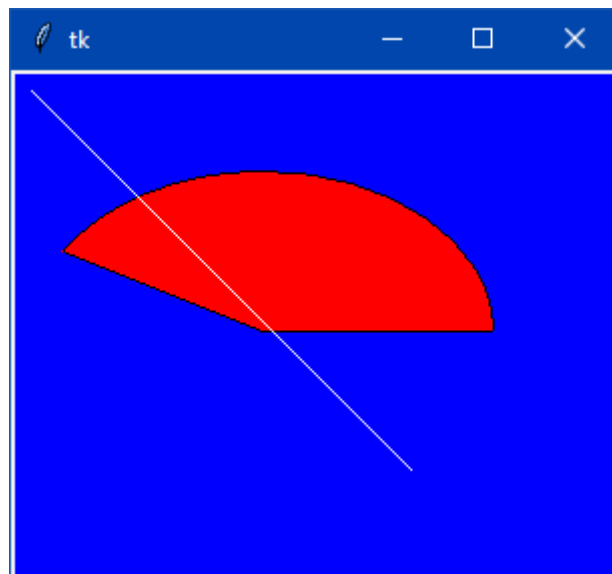
```
coord = 10, 50, 240, 210
```

```
arc = C.create_arc(coord, start = 0, extent = 150, fill = "red")
```

```
line = C.create_line(10,10,200,200,fill = 'white')
```

```
C.pack()
```

```
top.mainloop()
```



16.1.3 Checkbutton - Κουμπί ελέγχου

Το γραφικό στοιχείο Checkbutton χρησιμοποιείται για την εμφάνιση ενός αριθμού επιλογών ως πλαίσια ελέγχου. Ο χρήστης μπορεί να επιλέξει πολλές επιλογές ταυτόχρονα.

Σύνταξη

w = Checkbutton (master, option, ...)

Παράμετροι

- **master** – Το γονικό παράθυρο.
- **Επιλογές** – Ακολουθεί η λίστα με τις πιο συχνά χρησιμοποιούμενες επιλογές για αυτό το γραφικό στοιχείο. Αυτές οι επιλογές μπορούν να χρησιμοποιηθούν ως ζεύγη κλειδιού-τιμής που χωρίζονται με κόμμα.

1 *activebackground*

Χρώμα φόντου όταν το κουμπί βρίσκεται κάτω από τον κέρσορα.

2 *activeforeground*

Χρώμα προσκηνίου όταν το κουμπί βρίσκεται κάτω από τον κέρσορα.

3 *bd*

Πλάτος περιγράμματος σε pixel. Η προεπιλογή είναι 2.

4 *bg*

Κανονικό χρώμα φόντου.

5 *command*

Λειτουργία ή μέθοδος που θα κληθεί όταν κάνετε κλικ στο κουμπί.

6 *fg*

Κανονικό χρώμα προσκηνίου (κειμένου).

7 *font*

Γραμματοσειρά κειμένου που θα χρησιμοποιηθεί για την ετικέτα του κουμπιού.

8 *height*

Ύψος του κουμπιού σε γραμμές κειμένου (για κουμπιά κειμένου) ή εικονοστοιχεία (για εικόνες).

8 *bitmap*

Απεικόνιση μιας μονόχρωμης εικόνας στο κουμπί.

8 *image*

Απεικόνιση μιας γραφικής εικόνας στο κουμπί

Πλήρη σειρά επιλογών, καθώς και συχνά χρησιμοποιούμενων μεθόδων με το χειριστήριο, μπορούμε να βρούμε στην τεκμηρίωση της Python.

Ας δούμε ένα παράδειγμα δημιουργίας checkbox:

from tkinter import *

```
import tkinter
```

```
top = Tk()
```

```
CheckVar1 = IntVar()
```

```
CheckVar2 = IntVar()
```

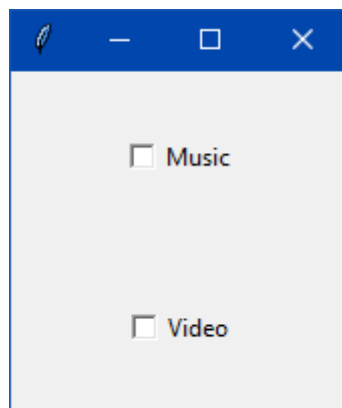
```
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \  
                 onvalue = 1, offvalue = 0, height=5, \  
                 width = 20, )
```

```
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \  
                 onvalue = 1, offvalue = 0, height=5, \  
                 width = 20)
```

```
C1.pack()
```

```
C2.pack()
```

```
top.mainloop()
```



16.1.4 Entry - Είσοδος

Το γραφικό στοιχείο Entry χρησιμοποιείται για την εμφάνιση ενός πεδίου κειμένου μιας γραμμής για την αποδοχή τιμών από έναν χρήστη.

Η σύνταξη και οι παράμετροι είναι παρόμοια και μπορείτε να τα αναζητείτε (σε περίπτωση μη επάρκειας όσων διαθέτουμε) στην τεκμηρίωση της Python.

Πάμε κατευθείαν στο παράδειγμα:

```
from tkinter import *
```

```
top = Tk()
```

```
L1 = Label(top, text = "Όνομα χρήστη")
```

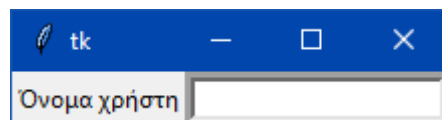
```
L1.pack( side = LEFT)
```

```
E1 = Entry(top, bd = 5)
```

```
E1.pack(side = RIGHT)
```

```
top.mainloop()
```

Έξοδος:



16.1.5 Frame - Πλαίσιο

Το γραφικό στοιχείο Frame χρησιμοποιείται ως γραφικό στοιχείο κοντέινερ για την οργάνωση άλλων γραφικών στοιχείων.

Πάμε κατευθείαν στο παράδειγμα:

```
from tkinter import *
```

```
root = Tk()
```

```
frame = Frame(root)
```

```
frame.pack()
```

```
bottomframe = Frame(root)
```

```
bottomframe.pack( side = BOTTOM )
```

```
redbutton = Button(frame, text = "Κόκκινο", fg = "red")
```

```
redbutton.pack( side = LEFT)
```

```
greenbutton = Button(frame, text = "Καφέ", fg="brown")
```

```
greenbutton.pack( side = LEFT )
```

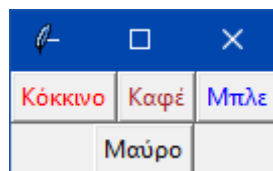
```
bluebutton = Button(frame, text = "Μπλε", fg = "blue")
```

```
bluebutton.pack( side = LEFT )
```

```
blackbutton = Button(bottomframe, text = "Μαύρο", fg = "black")
```

```
blackbutton.pack( side = BOTTOM)
```

```
root.mainloop()
```



16.1.6 Label - Επιγραφή

Το γραφικό στοιχείο Label χρησιμοποιείται για την παροχή λεζάντας μιας γραμμής για άλλα γραφικά στοιχεία. Μπορεί επίσης να περιέχει εικόνες.

Πάμε κατευθείαν στο παράδειγμα:

```
from tkinter import *
```

```
root = Tk()
```

```
var = StringVar()
```

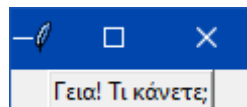
```
label = Label( root, textvariable = var, relief = RAISED )
```

```
var.set("Γεια! Τι κάνετε;")
```

```
label.pack()
```

```
root.mainloop()
```

Η έξοδος είναι:



16.1.7 Listbox - Πλαίσιο λίστας

Το γραφικό στοιχείο Listbox χρησιμοποιείται για την παροχή μιας λίστας επιλογών σε έναν χρήστη.

Παράδειγμα:

```
from tkinter import *
```



```
import tkinter
```

```
top = Tk()
```

```
Lb1 = Listbox(top)
```

```
Lb1.insert(1, "Python")
```

```
Lb1.insert(2, "Perl")
```

```
Lb1.insert(3, "C")
```

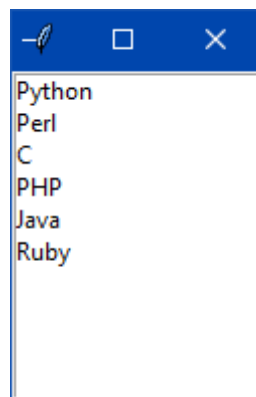
```
Lb1.insert(4, "PHP")
```

```
Lb1.insert(5, "Java")
```

```
Lb1.insert(6, "Ruby")
```

```
Lb1.pack()
```

```
top.mainloop()
```



16.1.8 Menubutton - Κουμπί μενού

Το γραφικό στοιχείο Menubutton χρησιμοποιείται για την εμφάνιση μενού στην εφαρμογή σας.

Παράδειγμα:

```
from tkinter import *
```

```
import tkinter
```

```
top = Tk()
```

```
mb = Menubutton ( top, text = "Σάλτσες", relief = RAISED )
```

```
mb.grid()
```

```
mb.menu = Menu ( mb, tearoff = 0 )
```

```
mb["menu"] = mb.menu
```

```
mayoVar = IntVar()
```

```
ketchVar = IntVar()
```

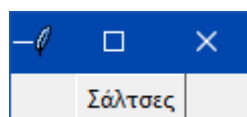
```
mb.menu.add_checkbutton ( label = "Μαγιονέζα",  
                           variable = mayoVar )
```

```
mb.menu.add_checkbutton ( label = "Κέτσαπ",  
                           variable = ketchVar )
```

```
mb.pack()
```

```
top.mainloop()
```

Έξοδος:



16.1.9 Menu - Μενού

Το γραφικό στοιχείο Μενού χρησιμοποιείται για την παροχή διαφόρων εντολών σε έναν χρήστη. Αυτές οι εντολές περιέχονται στο Menubutton.

Πάμε στο παράδειγμα:

```
from tkinter import *  
def donothing():  
    filewin = Toplevel(root)  
    button = Button(filewin, text="Do nothing button")  
    button.pack()  
  
root = Tk()  
menubar = Menu(root)  
filemenu = Menu(menubar, tearoff = 0)  
filemenu.add_command(label = "Νέο", command = donothing)  
filemenu.add_command(label = "Άνοιγμα", command = donothing)  
filemenu.add_command(label = "Αποθήκευση", command =  
donothing)  
filemenu.add_command(label = "Αποθήκευση ως...", command =  
donothing)  
filemenu.add_command(label = "Κλείσιμο", command = donothing)  
  
filemenu.add_separator()  
  
filemenu.add_command(label = "Έξοδος", command = root.quit)  
menubar.add_cascade(label = "Αρχείο", menu = filemenu)  
editmenu = Menu(menubar, tearoff=0)  
editmenu.add_command(label = "Αναιρεση", command = donothing)
```

editmenu.add_separator()

editmenu.add_command(label = "ΑΠΟΚΟΠΗ", command = donothing)

editmenu.add_command(label = "Αντιγραφή", command = donothing)

editmenu.add_command(label = "Επικόλληση", command = donothing)

editmenu.add_command(label = "Διαγραφή", command = donothing)

editmenu.add_command(label = "Επιλογή όλων", command = donothing)

menubar.add_cascade(label = "Επεξεργασία", menu = editmenu)

helpmenu = Menu(menubar, tearoff=0)

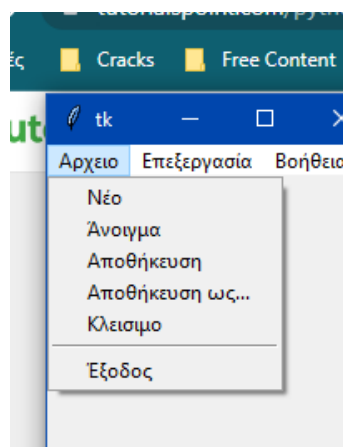
helpmenu.add_command(label = "Αρχείο Οδηγιών", command = donothing)

helpmenu.add_command(label = "Περί...", command = donothing)

menubar.add_cascade(label = "Βοήθεια", menu = helpmenu)

root.config(menu = menubar)

root.mainloop()



16.2 Ασκήσεις

1.

Δημιουργήστε ένα γραφικό περιβάλλον με το tkinter, το οποίο να περιέχει ένα label και να αλλάξετε τη γραμματοσειρά σε Arial Bold, καθώς και το μέγεθός της σε 70 και να την κάνετε με έντονη γραφή. (tkinter_label_change_less16.py).

2.

Δημιουργήστε ένα γραφικό περιβάλλον με ένα παράθυρο στο οποίο να θέσετε το προεπιλεγμένο του μέγεθος σε 600x300 και τον τίτλο του σε «-Προεπιλεγμένο μέγεθος παραθύρου-» με το tkinter. (tkinter_window_16less.py)

3.

Φτιάξτε ένα απλό παράθυρο στο tkinter στο οποίο να απενεργοποιείτε το resizing του παραθύρου. (tkinter_disable_resizing_16less.py)

ΚΑΛΗ ΜΕΛΕΤΗ
