ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Εισαγωγή στους Υπολογιστές

Ενότητα 6: Python Tkinter Παραθυρικές εφαρμογές με Python

Αβούρης Νικόλαος

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Σκοποί ενότητας

- Σκοπός της ενότητας είναι η χρήση της βιβλιοθήκης της Python Tkinter, για την δημιουργία παραθυρικών εφαρμογών.

Περιεχόμενα ενότητας

- Python Tkinter
- Παραθυρικές εφαρμογές με Python
- Παραδείγματα

Python Tkinter Παραθυρικές εφαρμογές με Python

Event based programming / προγραμματισμός με βάση γεγονότα

Το πρόγραμμα είναι ένας βρόχος επανάληψης:

While True:

    σχεδίασε οθόνη

    περίμενε event

    ανάλογα με το event :

        κάλεσε event_handler

root.mainloop()

- Η συνάρτηση **root.mainloop()** ξεκινάει ένα βρόχο επεξεργασίας γεγονότων:

while (True): # Loop forever

περίμενε για πάντα

όταν εμφανιστεί ενδιαφέρον γεγονός

κάλεσε τον eventhandler με το event object

Σημείωση: Πολλά γεγονότα τα διαχειρίζεται η ίδια η Tkinter(πχ ελαχιστοποίηση παράθυρου κλπ.)

Tkinter

- **Tkinter**: βιβλιοθήκη για παραθυρικές εφαρμογές της Python που υλοποιεί το μοντέλο προγραμματισμού γεγονότων (event based)
- Tkinter είναι η διεπαφή της Python με τη γραφική βιβλιοθήκη Tk
- Η Tkinter περιλαμβάνεται στις στάνταρ βιβλιοθήκες της Python δεν χρειάζεται να εγκατασταθεί.
- Εναλλακτικά υπάρχουν γραφικές βιβλιοθήκες όπως η WxPython PyGtk

Σχετικά modules

Import * from Tkinter

tkFont κλάση για τους μορφότυπουςχαρακτήρων

FileDialog(FileDialog, LoadFileDialog, SaveFileDialog)παράθυρο επιλογής αρχείου

tkColor ορίζει την askcolor(initialcolor), για το παράθυρο επιλογής χρώματος

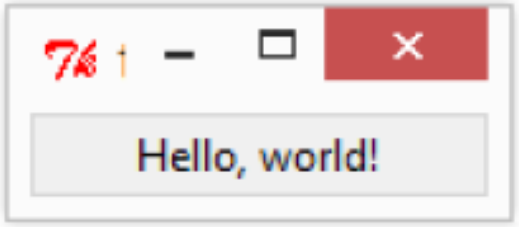
tkSimpleDialog ορίζει την askinteger, askfloatand askstringγια παράθυρα επιλογής τιμών.

PIL για διαχείριση εικόνων τύπου jpg, png

Παραθυρικές εφαρμογές με PythonGUI = graphical user interface

- Με την Tkinter δημιουργούμε παράθυρα που περιέχουν γραφικά στοιχεία **widgets** όπως labels, buttons, entry texts, menu, scroll bars
- Τα widgets δέχονται γεγονότα(**events**)από ενέργειες του χρήστη και ανταποκρίνονται με κλήση συναρτήσεων: event_handlers

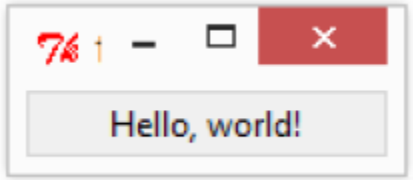
1ο πρόγραμμα



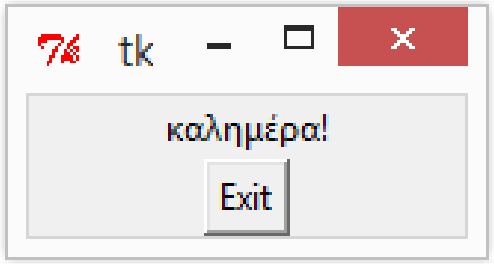

```
from Tkinterimport *
root = Tk() # Δημιουργία παράθυρου root
w = Label(root, text="Hello, world!")
  #δημιουργία label
w.pack() # τοποθέτηση ωστο παράθυρο
root.mainloop() # έναρξη βρόχου γεγονότων
  χρήστη
```

1ο πρόγραμμα με χρήση κλάσεων

```
from Tkinterimport *
class MyApp:
   def __init__(self, myParent):
        self.f1 = Frame(myParent)
        self.f1.pack()
        self.button1 = Button(self.f1, text="Hello world!")
        self.button1.pack()
#------------------------------
root = Tk()
myapp= MyApp(root)
root.mainloop()
```



2ο πρόγραμμα

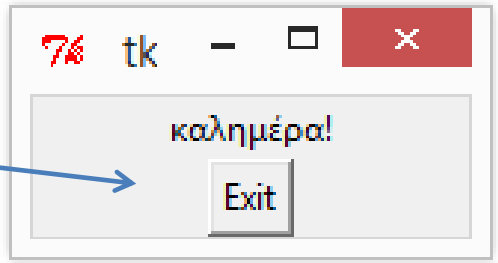



```
# -*-coding: utf-8 -*-
from Tkinterimport *
root = Tk() # Create the root window
w = Label(root, text="καλημέρα!") # Create label
w.pack() # Put the label in the window root
myButton= Button(root, text="Exit")# create button
myButton.pack() # put button in root
root.mainloop() # Start the event loop
```

2ο πρόγραμμα: κλήση handler

- Αν επιθυμούμε το γεγονός επιλογής του πλήκτρου myButton να καλεί τη συνάρτηση ButtonPushed θα πρέπει να τροποποιήσουμε τη δήλωση του:

  myButton= Button(root, text="Exit", command=buttonPushed)



Και να δηλώσουμε τη συνάρτηση

buttonPushed()

Event Handler

Πώς θα ορίσουμε την ButtonPushed ώστε να κλείνει το παράθυρο;

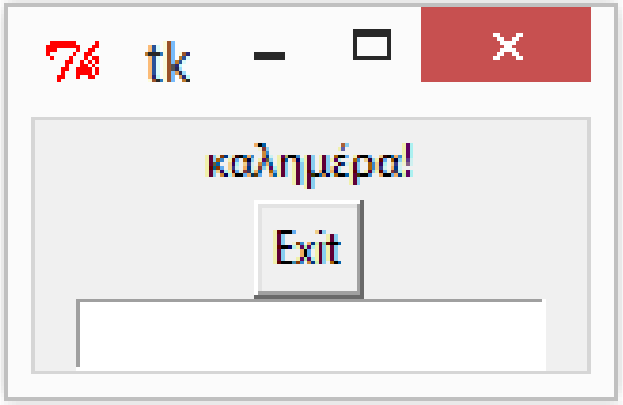
```
def buttonPushed():
        global root
        root.destroy()  #Kill the root window!
```

- Προσοχή: χρήση καθολικής μεταβλητής root, ώστε να χρησιμοποιηθεί εντός της συνάρτησης . Η μεταβλητή αυτή θα πρέπει να δηλωθεί στην αρχή του προγράμματος πριν το κυρίως πρόγραμμα

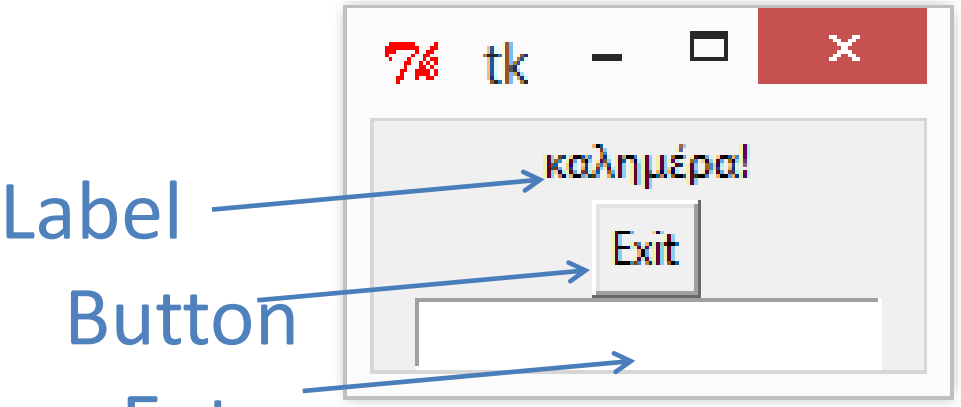
  root=None

Πώς θα προσθέσουμε ένα text entry box

```
tbox=Entry(root)
tbox.pack()
```



Γενικά : δημιουργία widget




```
# Δημιουργία widget
widget = <widgetname>(parent, attributes…)
#βάλε το widget στο παράθυρο
widget.pack()
```

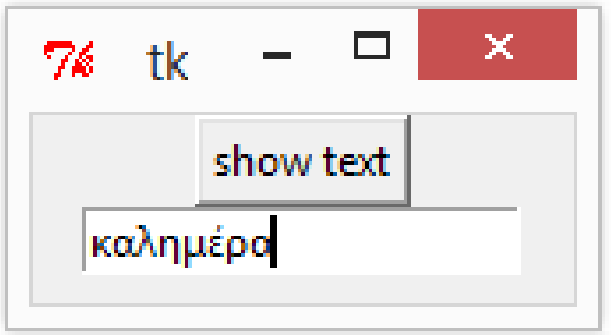
Άσκηση

- Να κατασκευάσετε παράθυρο με πλήκτρο και Entry text box, ώστε όταν επιλέγεται το πλήκτρο να τυπώνει το περιεχόμενο του Entry

Υπόδειξη: στη συνάρτηση handler του Button θα πρέπει να ορίσουμε μεταβλητή η οποία θα παίρνει την τιμή του Entry με τη μέθοδο getκαι θα την τυπώνει(η tboxνα γίνει καθολική μεταβλητή)

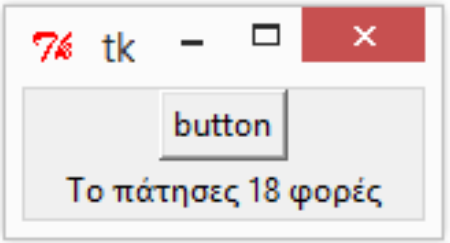
Π.χ.

```
txt = tbox.get()

print txt
```



Μεταβολή κειμένου Label (χρήση StringVar)

- Να δημιουργήσετε ένα παράθυρο με πλήκτρο (Button) και κείμενο (Label) το οποίο όταν πατηθεί το πλήκτρο να εμφανίζει το μήνυμα : «Το πάτησες x φορές»



Βοήθεια: να χρησιμοποιηθεί μεταβλητή

myText= StringVar()

την οποία αλλάζει η μέθοδος: myText.set(“xyz”)

Το Label σχετίζεται με τη μεταβλητή αυτή:

myLabel=Label(root, textvariable=myText)

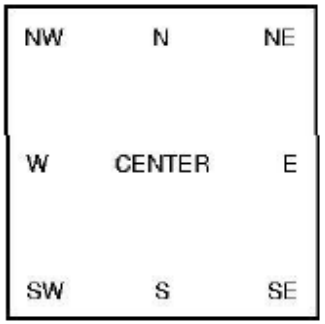
Widgets

- **Γραφικά στοιχεία**: Button, Checkbutton, Entry (εισαγωγή κειμένου), Label (απλό κείμενο), Listbox, Menubutton, Menu, Message (Label πολλών γραμμών), Radiobutton, Scale, Scrollbar, Text (μορφοποιημένο κείμενο πολλών γραμμών), Spinbox (Entry με προ-επιλεγμένες τιμές)
- **Υποδοχείς:** Canvas, Frame, LabelFrame (διαχωριστής Frame), Toplevel, PanedWindow
- **Σύνθετα στοιχεία διαλόγων**: tkMessageBox

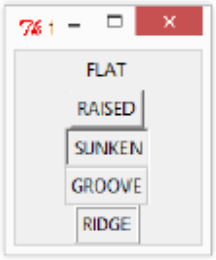
Ιδιότητες

- Χρώματα
- Anchors
- Στυλ Relief
- Cursors

"#ffffff"=white,"#000000"=black, "#000fff000" =pure green, ονόματα χρωμάτων: "white", "black", "red", "green", "blue", "cyan", "yellow", and "magenta" will always be available.



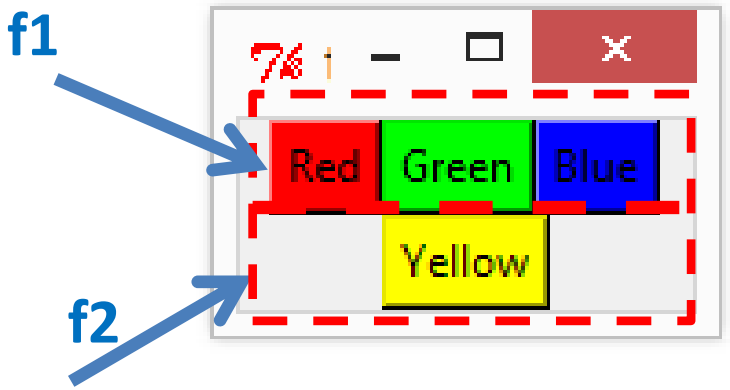
w.pack(side=TOP, anchor='e')



"arrow" "circle" "clock" "cross" "dotbox" "exchange" "fleur" "heart" "man" "mouse" "pirate" "plus" "shuttle" "sizing" "spider" "spraycan" "star" "target" "tcross" "trek" "watch"

Ιεραρχική διάταξη widgets: χρήση Frames

```
from Tkinterimport *
root = Tk()
f1 = Frame(root)
f1.pack()
f2 = Frame(root)
f2.pack( side = BOTTOM )
redb= Button(f1, text="Red", bg="red")
redb.pack( side = LEFT)
greenb= Button(f1, text = "Green",
      bg="green")
greenb.pack( side = LEFT )
blueb= Button(f1, text="Blue", bg="blue")
blueb.pack( side = LEFT )
yellowb= Button(f2, text="Yellow",
      bg="yellow")
yellowb.pack( side = BOTTOM)
root.mainloop()
```




```
root = Tk()
   f1 = Frame(root)
      redb= Button(f1,text="Red", bg="red")
```

Εναλλακτικός τρόπος διάταξης widget: ο διαχειριστής γεωμετρίας grid

Αντί για την μέθοδο pack μπορεί να χρησιμοποιηθεί η grid

w.grid(row = 1, column = 3)

Αυτή ορίζει το χώρο του παράθυρου ή του Frame με λογική στηλών και γραμμών όπως ένας πίνακας

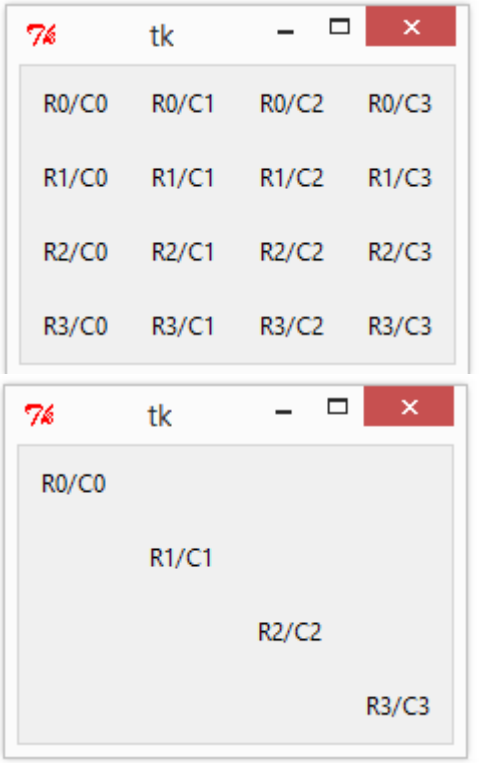
Δεν επιτρέπεται να χρησιμοποιήσουμε τις μεθόδους pack και grid μαζί

Άσκηση: διάταξη widgets με τον διαχειριστή γεωμετρίας grid

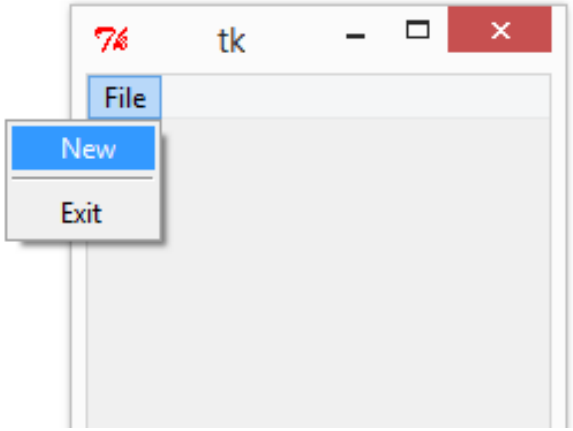
- Να διαταχθούν 16 label σε πίνακα όπως

στο σχήμα, εναλλακτικά μόνο τα διαγώνια στοιχεία .

```
from Tkinterimport *
root = Tk( )
for r in range(4):
    for c in range(4):
        lab=Label(root, text='R%s/C%s'%(r,c),
                  borderwidth=10 )
        lab.grid(row=r, column=c)
root.mainloop()20
```



Δημιουργία μενού



Ένα μενού επιλογών είναι widget με ιεραρχική δομή (υπο-μενού κλπ)

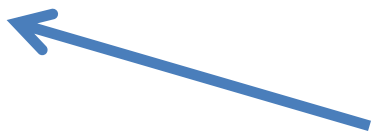
```
Root=Tk()
menubar= Menu(root)
filemenu= Menu(menubar, tearoff=0)
filemenu.add_command(label="New", command=donothing)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=root.quit)
menubar.add_cascade(label="File", menu=filemenu)
root.config(menu=menubar)
root.mainloop()
```

Συνάρτηση που καλείται αν γίνει η επιλογή

Εικόνες

- Εικόνες μόνο τύπου gif στο Tkinter
- Για εικόνες τύπου jpg, pngαπαιτείται η χρήση της βιβλιοθήκης PIL

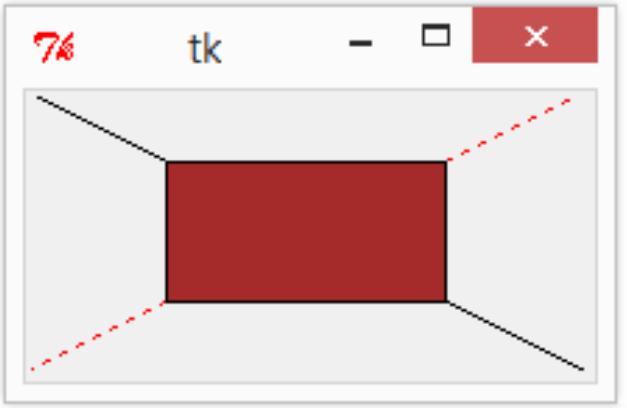
```
image1 = PhotoImage(file=“myimage.gif)
root.geometry("%dx%d+%d+%d" % (400, 300, 0, 0))
l= Label(root, image=image1)
l.image=image1
l.pack(side='top', fill='both', expand='yes')
```

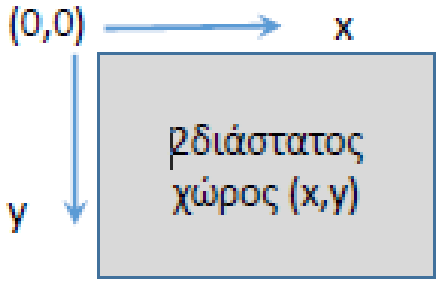


Η εικόνα εισάγεται σε widget τύπου label. Μπορεί επίσης να εισαχθεί σε canvas.

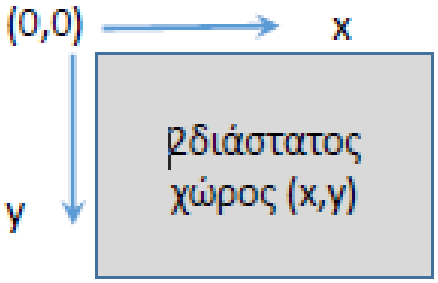
Canvas: υποδοχέας γραφικών αντικειμένων, εικόνων κλπ.

```
from Tkinterimport *

master = Tk()

w = Canvas(master, width=200, height=100)

w.pack()

w.create_line(0, 0, 200, 100)

w.create_line(0, 100, 200, 0, fill="red", dash=(4, 4))

w.create_rectangle(50, 25, 150, 75, fill="brown")

mainloop()
```





Canvas: είδη γραφικών αντικειμένων



- arc (arc, chord, or pieslice)
- bitmap (από αρχείο XBM)
- image (εικόναBitmapImageή PhotoImage)
- line
- oval (κύκλος ή έλλειψη)
- polygon
- rectangle
- text
- window

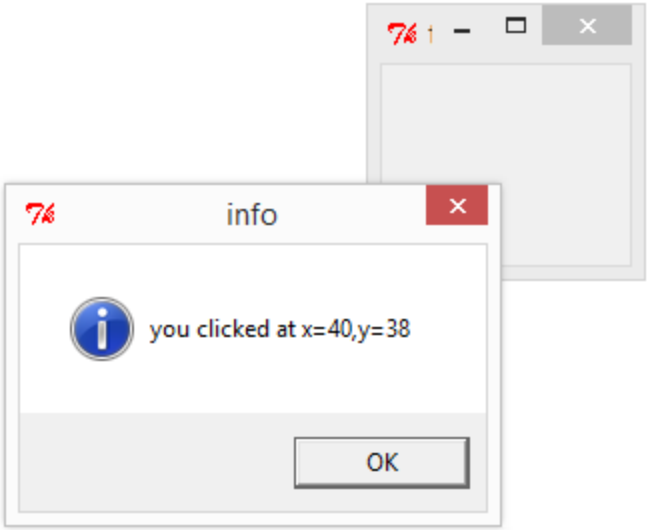
Περισσότερα:
http://effbot.org/tkinterbook/canvas.htm

Dialog box

- Ένα παράθυρο διαλόγουdialog box είναι ένα μονοτροπικό(modal) παράθυροπου θέτει ένα ερώτημα στο χρήστη, δείχνει ένα μήνυμα σφάλματος ζητάει μια πληροφορία κλπ.
- Πολλά τέτοια παράθυρα διαλόγου, όπως το παράθυρο επιλογής αρχείου ή επιλογής χρώματος, κλππροσφέρονται από το γραφικό περιβάλλον και για αυτό είναι ίδια σε πολλές εφαρμογές.

Example: showinfo

```
import tkMessageBox# Another way you can import
from Tkinterimport *
def callback(event):
    tkMessageBox.showinfo("info",
        "you clicked at x=%d,y=%d" % (event.x, event.y))
root = Tk()
frame = Frame(root, width=100, height=100)
frame.bind("<Button-1>", callback)
frame.pack()
root.mainloop()
```



Άλλα παράθυρα διαλόγου: askyesno, askokcancel, askretrycancel, askquestion

from tkMessageBoximport *

ans= askyesno("Continue", "Should I continue?”)

Η μεταβλητή ans θα είναι είτε True (Yes) ή False (No).

Άλλα παράθυρα διαλόγου: askokcancel, askretrycancel,

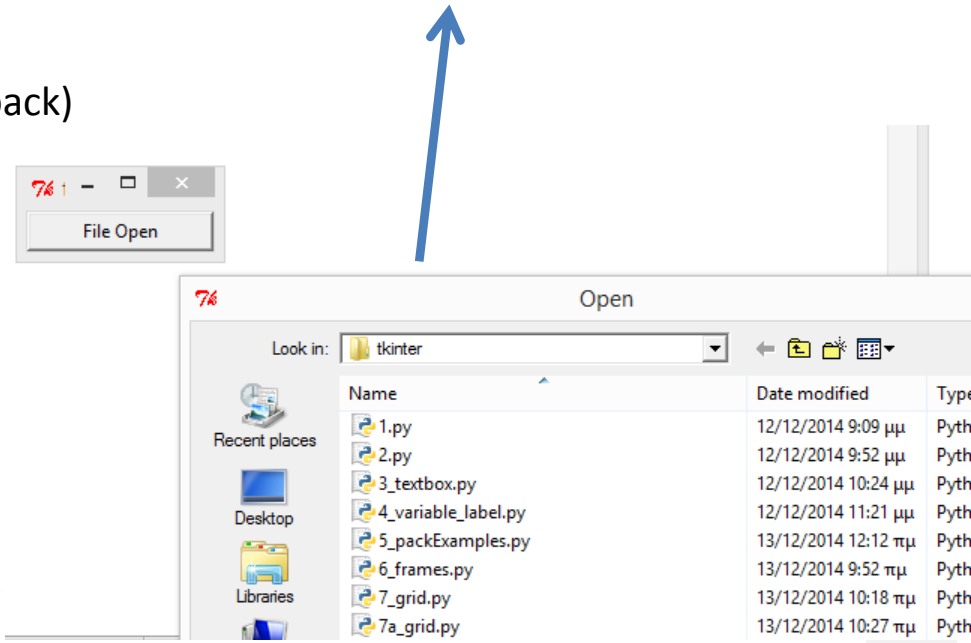
askquestion

Προσοχή: askquestion επιστρέφει “yes” “no” και όχι True / False!

Άνοιγμα αρχείου: askopenfilename

```
from Tkinterimport *
from tkFileDialogimport askopenfilename
defcallback():
name= askopenfilename()
print name
root = Tk()
b=Button(root, text='File Open', command=callback)
b.pack(fill=X)
mainloop()
```

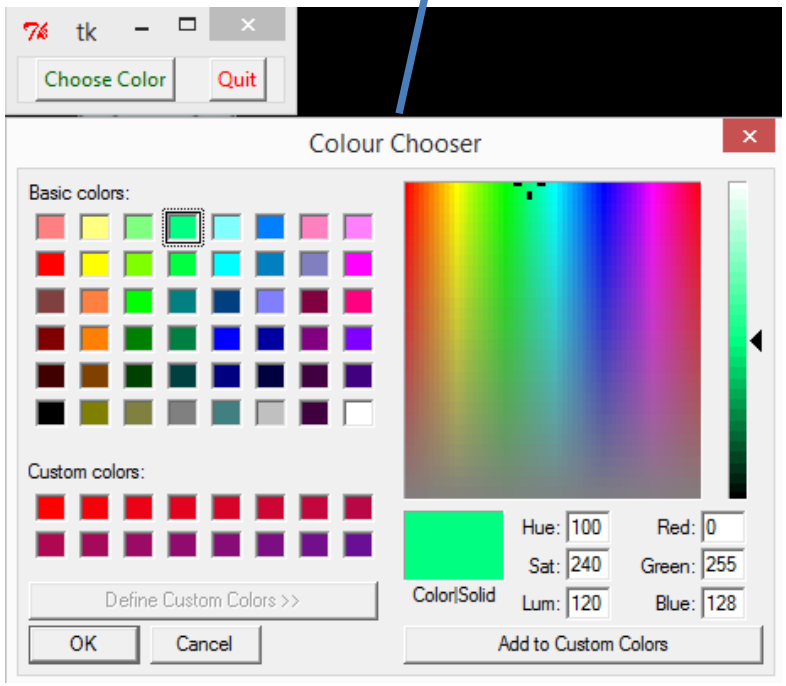
C:/Users/user/PycharmProjects/eclass/tkinter/1.py



Επιλογή χρώματος: askcolor

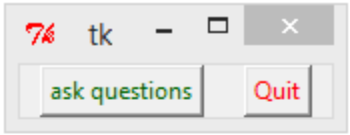
```
from Tkinterimport *
from tkColorChooserimport askcolor
def callback():
    result = askcolor(color="#6A9662", title = "Colour Chooser")
    print result
root = Tk()
b1=Button(root, text='Choose Color',
    fg="darkgreen", command=callback)
B1.pack(side=LEFT, padx=10)
b2=Button(text='Quit', command=root.quit,
    fg="red")
b2.pack(side=LEFT, padx=10)
mainloop()
```

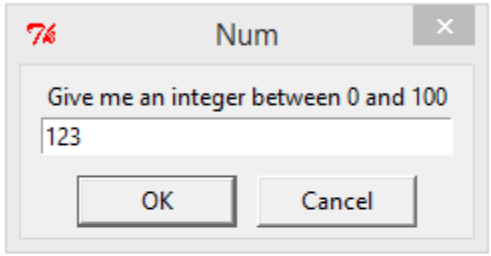
((0, 255, 128), '#00ff80')

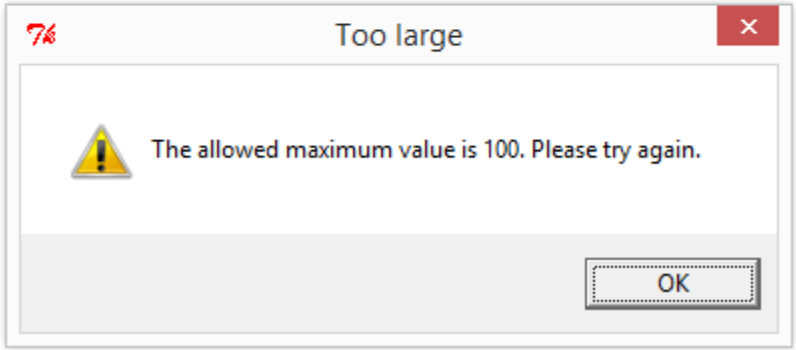


Διάλογος εισαγωγής τιμής από το χρήστη: askstring, askinteger

```
from Tkinterimport *
from tkSimpleDialogimport *
def callback():
        ans= askstring("Title", "Give me your name")
        print ans
        ans= askinteger("Dialog Title", "Give me an integer")
        print ans
        ans= askinteger("Num", "Give me an integer between 0 and 100",
        minvalue=0, maxvalue=100)
        print ans
root = Tk()
b1=Button(root, text='ask questions',fg="darkgreen", command=callback)
b1.pack(side=LEFT, padx=10)
b2=Button(text='Quit', command=root.quit,fg="red")
b2.pack(side=LEFT, padx=10)
mainloop()
```







Γεγονότα από το ποντίκι-πληκτρολόγιο

Χρειάζεται η αντιστοίχιση γεγονότων χρήστη με widget

widget.bind(event, handler)

Κατηγορίες γεγονότων:

<Button-1>(αριστερό πλήκτρο ποντικιού)

<Double-Button-1> (διπλό κλικ)

<Enter> -το ποντίκι εισέρχεται στο widget

<Leave> -το ποντίκι αφήνει το widget

<Return> -πάτημα πλήκτρου enter

<key> (<a> για παράδειγμα) –πάτημα πλήκτρου “a”

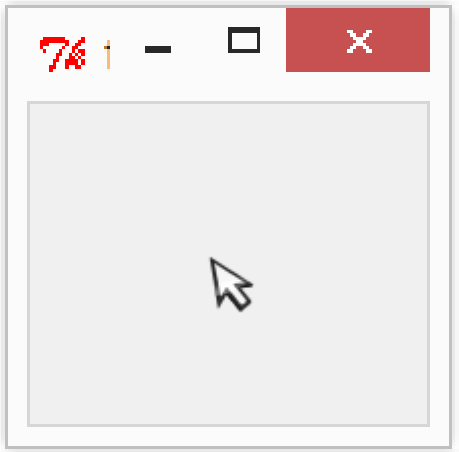
Άσκηση: περιγράψτε τον κώδικα

```
from Tkinterimport *
def main():
    global root
    root = Tk() # Create the root
    (base) window where all widgets
    go
    b = Button(root, text="Logon")
    b.bind("<Enter>",mouseEntered)
    b.bind("<Leave>",mouseExited)
    b.pack()
    root.mainloop() # Start the event
loop
main()
```

```
def mouseEntered(event):
        button = event.widget
        button.config(text = "click!")
def mouseExited(event):
        button = event.widget
        button.config(text =
"Logon")
```

event.x, event.yοι συντεταγμένες x,y

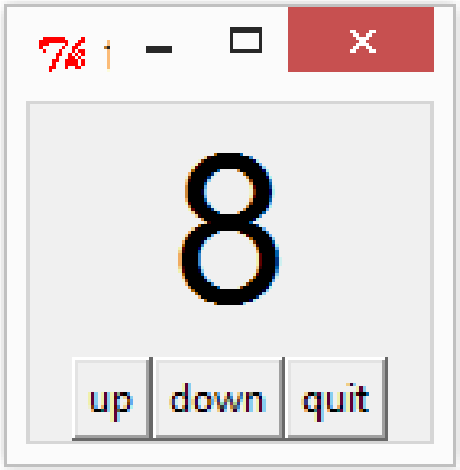
```
from Tkinterimport *
root = Tk()
def callback(event):
    print "clicked at", event.x, event.y
frame = Frame(root, width=100, height=100)
frame.bind("<Button-1>", callback)
frame.pack()
root.mainloop()
```



Clicked at 32 47

Άσκηση

- Να κατασκευάσετε πρόγραμμα Python που δημιουργεί παράθυρο, όπως στο σχήμα, το οποίο εμφανίζει έναν αριθμό (αρχική τιμή 1) και 3 πλήκτρα: up, down, quit
- Όταν πατηθεί το up ο αριθμός αυξάνει κατά 1, down μειώνεται κατά 1 (ως την τιμή 1) και quit κλείνει η εφαρμογή.



Πηγές

- https://wiki.python.org/moin/TkInter (INDEX)
- http://zetcode.com/gui/tkinter/ αντικειμενοστραφήςπροσέγγιση
- http://www.tutorialspoint.com/python/python_gui_programming.htm εισαγωγή στην Tkinter, απλά παραδείγματα
- http://www.ferg.org/thinking_in_tkinter/all_programs.html εισαγωγή στον προγραμματισμό μέσω γεγονότων
- http://effbot.org/tkinterbook/ work in progress

Προσοχή υπάρχουν tutorials για τη βιβλιοθήκη tkinter (Python 3.xx)

Επίσης επεκτάσεις όπως η ttk και η PMW

Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

- Έκδοση 1.0 διαθέσιμη εδώ.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Διαφάνειες 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 23, 24, 26, 29, 31, 32, 33, 36, 37: χρήση του Python Editor IDLE