

week 4

Γραφική διεπαφή χρήστη

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Την εβδομάδα αυτή γίνεται εισαγωγή στη βιβλιοθήκη
tkinter και τα πιο απλά γραφικά της αντικείμενα (widgets)

L4.1

Γραφική διεπαφή χρήστη: εισαγωγή στην tkinter

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Στην διάλεξη 4.1 θα εξετάσουμε το μοντέλο προγραμματισμού με γεγονότα (event programming) και τα βασικά στοιχεία της βιβλιοθήκης tkinter.

week 4

Γραφική διεπαφή χρήστη

Προγραμματίζω
με την python



L4.1 Εισαγωγή στην tkinter- απλά widget

L4.2 Εικόνες και υποδοχείς

L4.3 Μενού και διάλογοι

Μάθημα L4.1

Εισαγωγή στην tkinter

Προγραμματίζω
με την python 

- V4.1.1 Γραφικές διεπαφές χρήστη:
Προγραμματισμός με γεγονότα
- V4.1.2 Το πρώτο γραφικό πρόγραμμα
- V4.1.3 Παράδειγμα: γεννήτρια παραθύρων
- V4.1.4 Απλά widget
- V4.1.5 Εικόνες

Προγραμματίζω
με την python



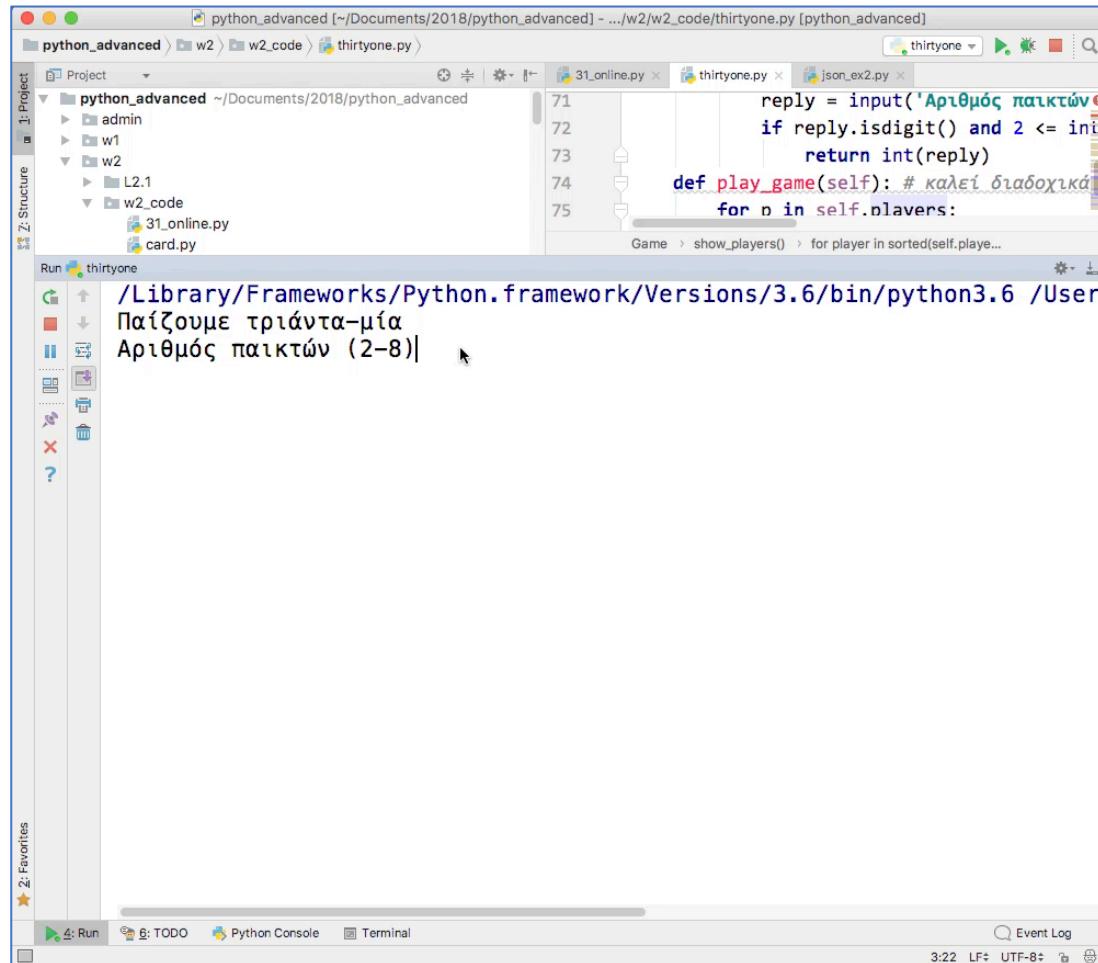
V4.1.1

Γραφικές διεπαφές χρήστη:
Προγραμματισμός με γεγονότα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Γραφική διεπαφή χρήστη GUI: Graphic User Interface

Προγραμματίζω
με την python

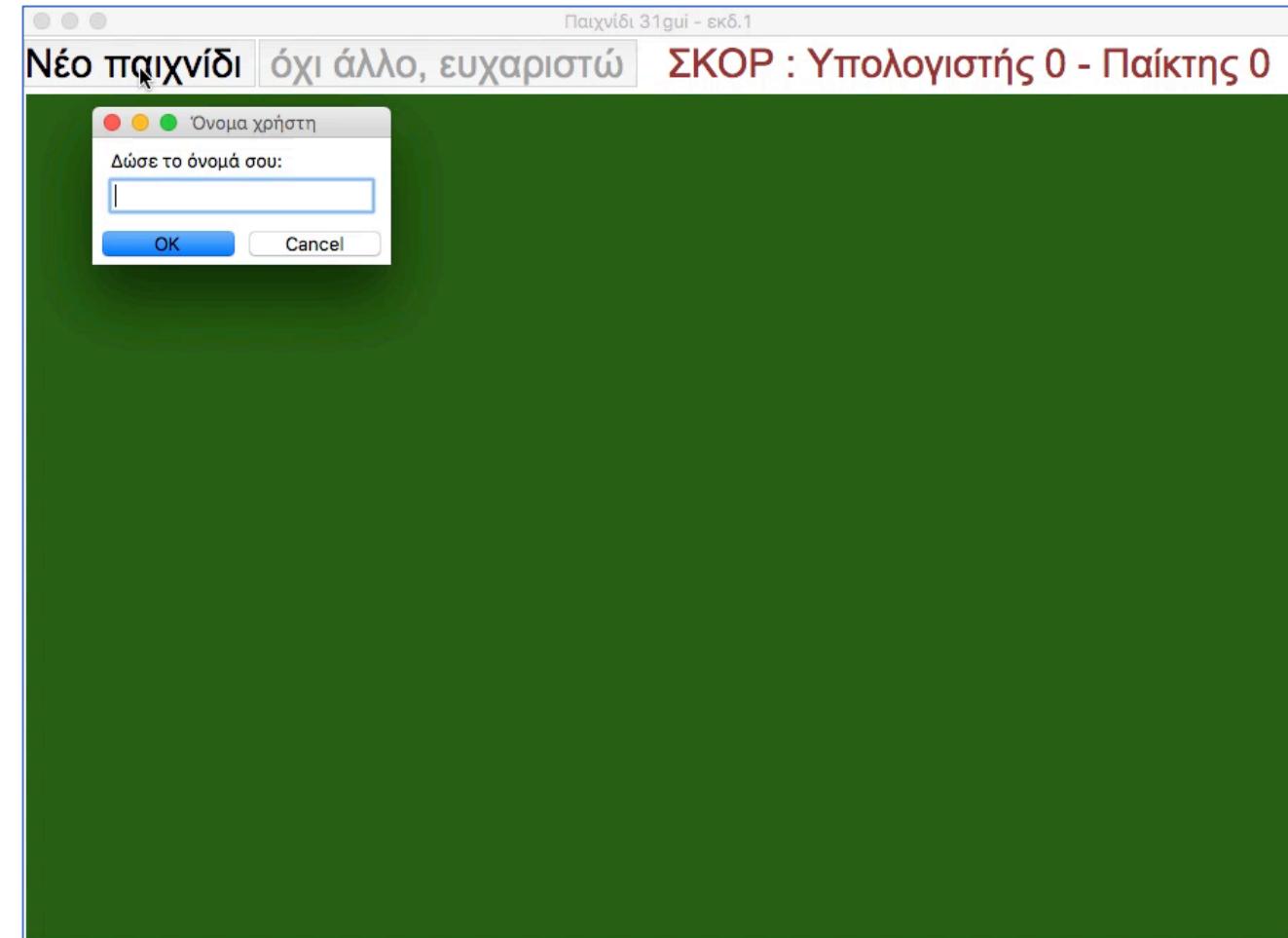


A screenshot of the PyCharm IDE. The code editor shows a file named 'thirtyone.py' with the following code:

```
71     reply = input('Αριθμός παικτών')
72     if reply.isdigit() and 2 <= int(reply) <= 8:
73         return int(reply)
74     def play_game(self): # καλεί διαδοχικά
75         for p in self.players:
```

The terminal window below displays the output of the program:

```
/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6 /Users
Παίζουμε τριάντα-μία
Αριθμός παικτών (2-8)|
```



Event-based programming

προγραμματισμός με γεγονότα

Προγραμματίζω
με την python 

Το πρόγραμμα είναι ένας βρόχος επανάληψης:

`while True:`

σχεδίασε γραφικά αντικείμενα
περίμενε **event** (ενέργεια) από χρήστη
ανάλογα με το event :
κάλεσε συνάρτηση χειριστή (**event_handler**)



tkinter : βιβλιοθήκη για γραφικές εφαρμογές της Python που υλοποιεί προγραμματισμό γεγονότων (event based programming)

- **tkinter** είναι η διεπαφή της Python 3 με τη γραφική βιβλιοθήκη **tk (tcl/tk)**
- Η tkinter περιλαμβάνεται στις βασικές βιβλιοθήκες της Python
- δεν χρειάζεται να εγκατασταθεί.
- Εναλλακτικές γραφικές βιβλιοθήκες είναι οι WxPython PyGtk, PyQt και η kivy

Παραθυρικές εφαρμογές με Python

GUI = graphical user interface

Προγραμματίζω
με την python



- 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**

```
window= tkinter.Tk()  
window.mainloop()
```

Προγραμματίζω
με την python 

Η συνάρτηση `.mainloop()` ξεκινάει ένα βρόχο επεξεργασίας γεγονότων για το αντικείμενο window:

while True: # Loop forever

περίμενε για πάντα

όταν εμφανιστεί ενδιαφέρον γεγονός

κάλεσε τον event-handler με όρισμα το <event object>

Σημείωση: Πολλά γεγονότα τα διαχειρίζεται η ίδια η tkinter όπως πχ κλείσιμο παράθυρου με (x)

Σχετικά modules της tkinter

Προγραμματίζω
με την python 

tkinter.ttk themed tk (περισσότερα widget, χρήση στυλ)

tkinter.scrolledtext δημιουργία παράθυρου με κύληση περιεχομένου

tkinter.colorchooser ορίζει την askcolor(initialcolor), για το παράθυρο επιλογής χρώματος

tkinter.filedialog παράθυρο επιλογής αρχείου

tkinter.font κλάση για τους μορφότυπους χαρακτήρων

tkinter.messagebox κλάση για πλαίσια μηνυμάτων

tkinter.simpledialog ορίζει την askinteger, askfloat and askstring για παράθυρα επιλογής τιμών.

PIL για διαχείριση εικόνων τύπου jpg, png

Προγραμματίζω
με την python



V4.1.2

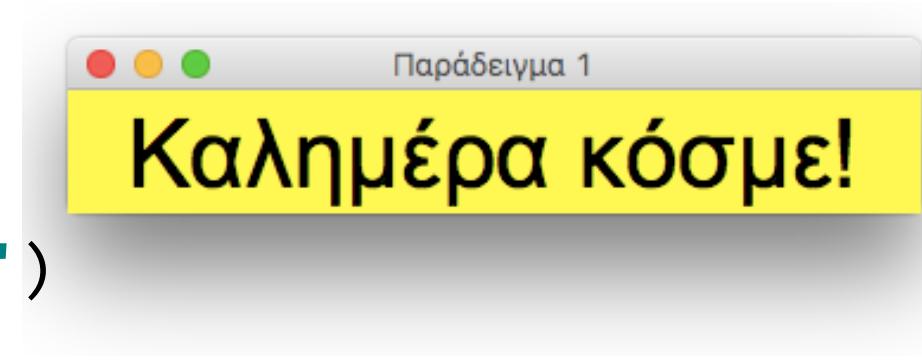
Το πρώτο γραφικό πρόγραμμα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Το πρώτο γραφικό πρόγραμμα το αντικείμενο τύπου Label

```
import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.root = root
        self.root.title('Παράδειγμα 1')
        self.widgets()
    def widgets(self):
        self.w = tk.Label(self.root, text="Καλημέρα κόσμε!", \
                          font = "Arial 36", bg='yellow')
        self.w.pack()
#-----
root = tk.Tk() #δημιουργός παραθύρου
myapp = MyApp(root)
root.mainloop()
```

Προγραμματίζω
με την python



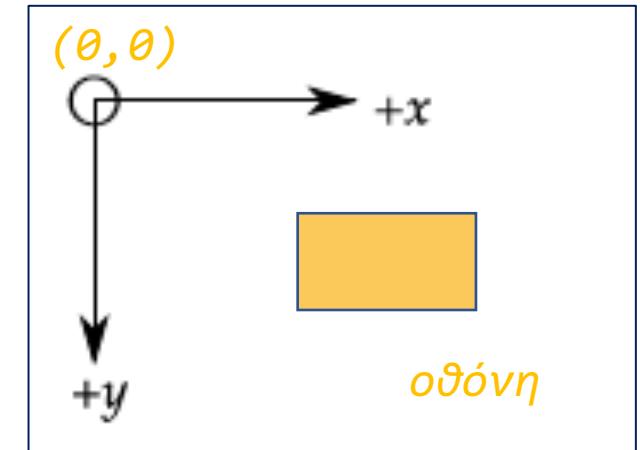


Κλάσεις δημιουργίας παραθύρων

```
root = tk.Tk() # βασικό παράθυρο  
w1 = tk.Toplevel() # άλλα παράθυρα
```

Μέθοδοι των κλάσεων Tk και Toplevel

```
root.title(text) # ο τίτλος του παράθυρου  
root.geometry("{}x{}+{}+{}).format(width, height, x,y, ) # διαστάσεις και θέση  
root.resizable(width = False, height = False) # όχι αλλαγή μεγέθους
```



χρώματα στην tkinter

Προγραμματίζω
με την python 

Τα χρώματα μπορούν να αναφερθούν είτε λεκτικά, είτε ως 3 δεκαεξαδικοί αριθμοί στο πεδίο (0-255) δηλαδή με τιμές που περιγράφονται με 2 bit (μεταξύ 00 –FF).

Μορφή **# rrggbb**. Αυτό επειδή οι 3 αυτοί αριθμοί αντιστοιχούν στις αποχρώσεις των βασικών χρωμάτων RED, GREEN, BLUE και παίρνουν τιμή 00 για πλήρη έλλειψη του χρώματος και FF πλήρη παρουσία.

#FF0000 σημαίνει μέγιστη τιμή για R και μηδέν για G,B άρα το κόκκινο χρώμα.

αντίστοιχα #000000 είναι το μαύρο (μη-χρώμα) και #FFFFFF το λευκό.

Επίσης μπορεί να χρησιμοποιήσει κανείς λεκτικά όπως 'white' , 'black' , 'red' , 'green' , 'blue' , 'cyan' , 'yellow' , 'magenta' , κλπ .

```
r = lambda: random.randint(0, 255)
color = '#{ :02X}{ :02X}{ :02X}'.format(r(), r(), r())
```

τυχαίο χρώμα:

Προγραμματίζω
με την python



V4.1.3

Η μηχανή γεωμετρίας pack

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

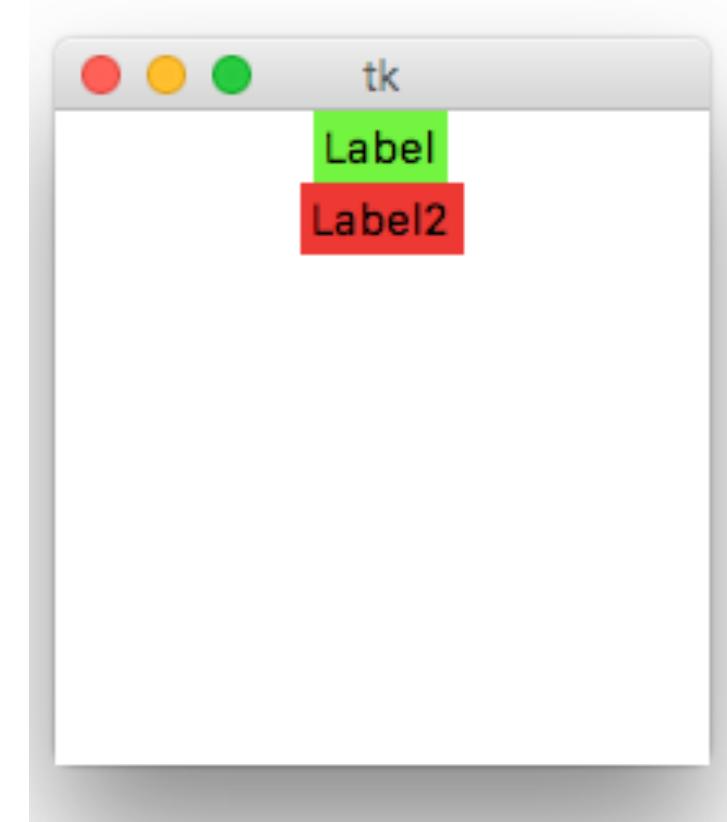
Μηχανή γεωμετρίας pack()

έστω παράθυρο διαστάσεων 200x200 px
στο οποίο τοποθετούμε με την pack() 2 label, ένα
πράσινο και ένα κόκκινο

```
import tkinter as tk
class MyApp():
    def __init__(self, root):
        root.geometry('200x200+200+200')
        l1 = tk.Label(root, text='Label', bg='green')
        l1.pack()
        l2 = tk.Label(root, text='Label2', bg='red')
        l2.pack()
root = tk.Tk()
MyApp(root)
root.mainloop()
```

παράθυρο διαστάσεων
200x200 pixel

Προγραμματίζω
με την python



Μηχανή γεωμετρίας pack()

Προγραμματίζω
με την python



expand = True, False (προκαθορισμένη τιμή)
όταν True καταλαμβάνει όλο το διαθέσιμο χώρο

fill = 'x', 'y', 'both', 'none' (προκαθορισμένη τιμή)
αποφασίζει αν θα μείνει στις ελάχιστες διαστάσεις ή θα
καταλάβει όλον τον διαθέσιμο χώρο, προς την x, ψή φή και τις
δύο διαστάσεις

side = 'left', 'right', 'top' (προκαθορισμένη), 'bottom'
αποφασίζει ποια πλευρά του υποδοχέα αρχίζει, σημείωση: όλα
τα παιδιά ενός υποδοχέα θα πρέπει να έχουν ίδια τιμή και
εξυπηρετούνται με τη σειρά.



Μηχανή γεωμετρίας pack()

Προγραμματίζω
με την python



```
pack()  
pack()
```

```
pack(expand=1, fill = 'y')  
pack(fill = 'both')
```

```
pack(expand=1)  
pack(fill = 'both')
```

```
pack(fill='both', expand=1, side='left')  
pack(fill='both', expand=1, side='right')
```

```
pack(fill = 'both', expand=1)  
pack(fill = 'both', expand=1)
```

Παράμετροι της pack που καθορίζουν τη θέση και συμπεριφορά των γραφικών αντικειμένων στο διαθέσιμο χώρο.

fill = 'both', 'x', 'y'
side = 'left', 'right', 'top', 'bottom'
expand = True, False

Προγραμματίζω
με την python



V4.1.4

Παράδειγμα: γεννήτρια παραθύρων

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Άσκηση

Να γράψετε ένα πρόγραμμα που δημιουργεί ένα αρχικό παράθυρο διαστάσεων 200x200 pixel στη θέση 100,100 και στη συνέχεια παράγει 10 ακόμη παράθυρα σε τυχαίες θέσεις στην οθόνη που το καθένα είναι διαφορετικό χρώμα και περιέχει τον αύξοντα αριθμό δημιουργίας τους.

- πώς μπορούμε να εξασφαλίσουμε ότι δεν θα υπάρχει επικάλυψη;

οι διαστάσεις της οθόνης δίνονται από
`root.winfo_screenheight()`, `root.winfo_screenwidth()`

Προγραμματίζω
με την python



Γεννήτρια παραθύρων

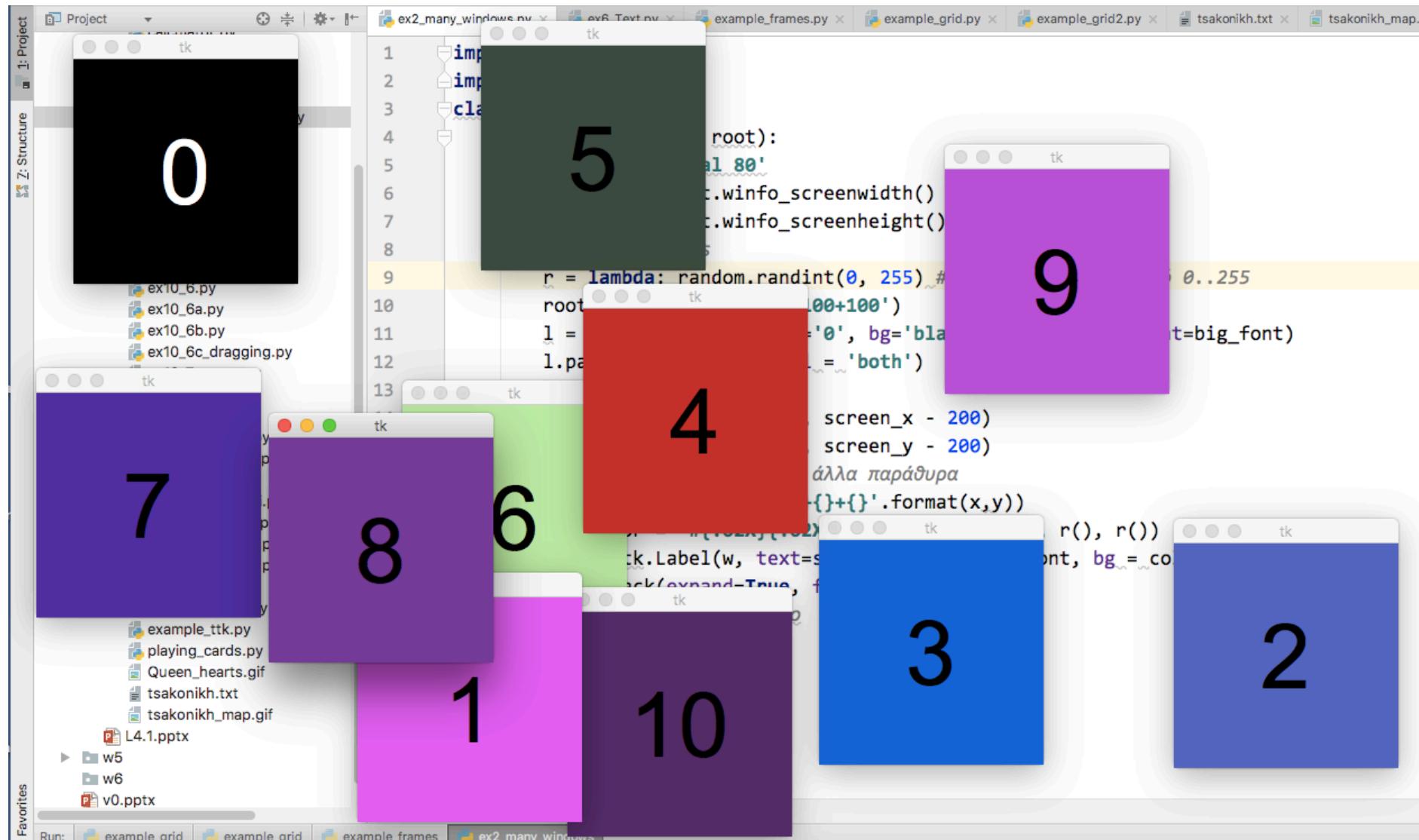
Προγραμματίζω
με την python 

```
for i in range(10):
    x= random.randint(0, screen_x - 200)
    y= random.randint(0, screen_y - 200)
    w = tk.Toplevel() # άλλα παράθυρα
    w.geometry('200x200+{}+{}'.format(x,y))
    color = '#{ :02X}{ :02X}{ :02X}'.format(r(), r(), r())
    l =tk.Label(w, text=str(i+1), font=big_font, bg = color)
    l.pack(expand=True, fill = 'both')
```

Προγραμματίζω
με την python



Γεννήτρια παραθύρων



Προγραμματίζω
με την python



V4.1.5

Απλά widget

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Οι κύριες κλάσεις γραφικών αντικειμένων (widget) της tkinter

Προγραμματίζω με την python 

Label απλή περιοχή για μήνυμα κειμένου

Button Απλό πλήκτρο

Message Ένα label πολλών γραμμών

Entry ένα πεδίο εισαγωγής κειμένου μιας γραμμής

Checkbutton Ένα στοιχείο επιλογής μεταξύ δύο καταστάσεων για πολλαπλή επιλογή onvalue, offvalue

Radiobutton Ένα στοιχείο επιλογής ενός μεταξύ διαφορετικών στοιχείων

Scale κυλιόμενη μπάρα επιλογής

PhotoImage Αντικείμενο για παρουσίαση εικόνων τύπου .gif .ppm .pgm

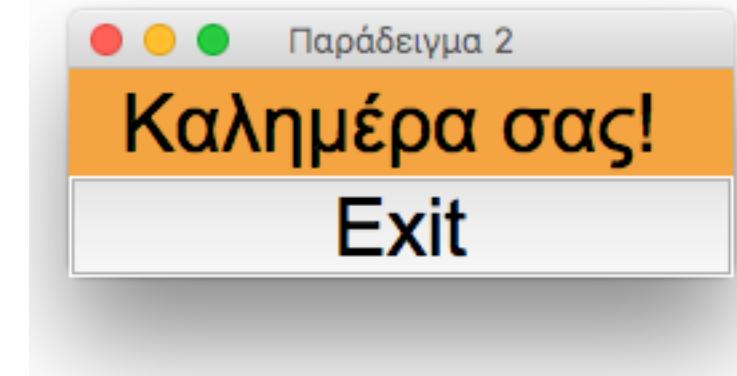
BitmapImage Αντικείμενο για παρουσίαση εικόνων τύπου bitmap .xbm

γραφικό αντικείμενο Button

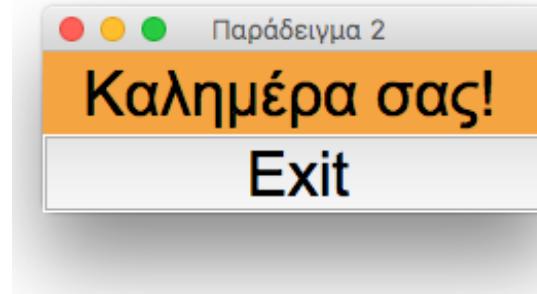
Προγραμματίζω
με την python



```
import tkinter as tk
class MyApp:
    def __init__(self, root):
        self.root = root
        root.title('Παράδειγμα 2')
        self.widgets()
    def widgets(self):
        self.w = tk.Label(self.root, text=" Καλημέρα σας! ", font = "Arial 30",
bg="orange")
        self.w.pack(fill = 'both', expand=1)
        self.b = tk.Button(self.root, text="Exit", font = "Arial 30", command =
self.buttonPushed)
        self.b.pack(fill = 'both', expand=1)
    def buttonPushed(self):
        self.root.destroy() # Kill the root window!
#-----
root = tk.Tk()
myapp = MyApp(root)
root.mainloop()
```



κλήση συνάρτησης χειριστή



Προγραμματίζω
με την python 

Αν επιθυμούμε το γεγονός επιλογής του πλήκτρου **myButton** να καλεί τη συνάρτηση **ButtonPushed** το δηλώνουμε στις παραμέτρους του Button:

```
myButton = Button (root, text= "Τέλος", command = buttonPushed)
```

Στη συνάρτηση **buttonPushed()** που ονομάζεται **event handler** (χειριστής του γεγονότος, πάτημα του πλήκτρου) ορίζουμε τι πρέπει να συμβεί όταν κληθεί.

Πώς θα ορίσουμε τη συνάρτηση χειριστή ButtonPushed ώστε να κλείνει το παράθυρο;

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

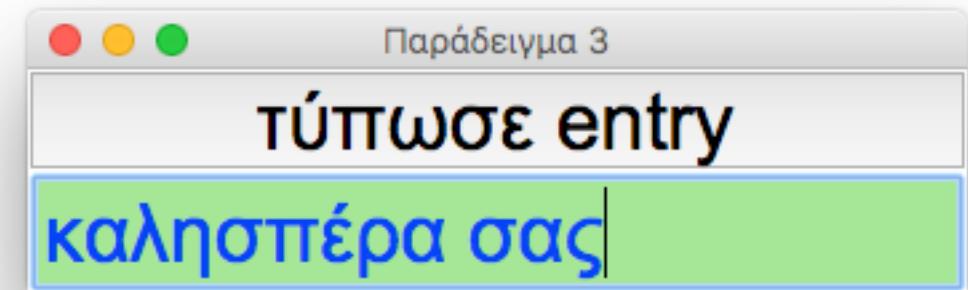
Καλείται η μέθοδος `destroy()` στο αντικείμενο παράθυρο.
Προσοχή: το παράθυρο είναι γνωστό στη μέθοδο μέσω της μεταβλητής `self.p`

Προγραμματίζω
με την python 

το γραφικό αντικείμενο Entry

Προγραμματίζω
με την python 

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



Είναι το γραφικό αντικείμενο που
επιτρέπει την εισαγωγή κειμένου
(πχ σε μία φόρμα)

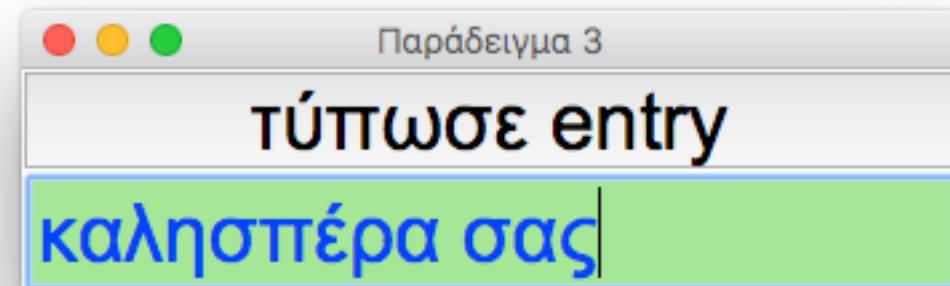
```

import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.fnt = 'Arial 30'
        root.title('Παράδειγμα 3')
        self.root = root
        self.widgets()
    def widgets(self):
        self.button = tk.Button(self.root, text=" τύπωσε entry ", font = self.fnt,
                               command=self.showText)
        self.button.pack(fill='both', expand=1)
        self.entry = tk.Entry(self.root, font= self.fnt, width= 20, bg='lightgreen',
                             fg='blue') # το πλαίσιο εισαγωγής κειμένου
        self.entry.pack(fill='both', expand=1)
    def showText(self): # χειριστής γεγονότος επιλογής πλήκτρου b
        text = self.entry.get() # πάρε το κείμενο που έχει εισαχθεί στο Entry
        print(text) # τύπωσέ το
#-----
root = tk.Tk()
myapp = MyApp(root)

```

Entry

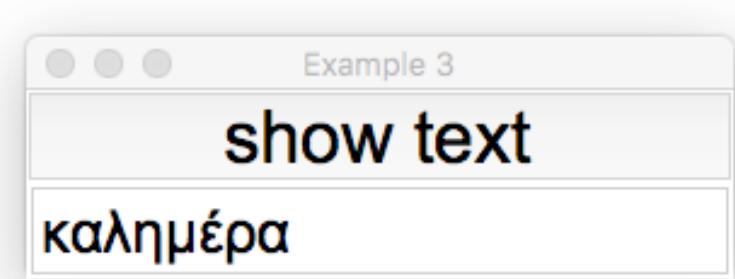
Προγραμματίζω
με την python



Η μέθοδος **get()** του αντικειμένου **Entry**

Προγραμματίζω
με την python 

Στη συνάρτηση χειριστή του Button **showText()** ορίστηκε μεταβλητή **text** η οποία παίρνει την τιμή του Entry με τη μέθοδο **get** και την τυπώνει **mytext = tbox.get()
print (mytext)**



Αντικείμενο **StringVar** για μεταβολή κειμένου Label

Προγραμματίζω
με την python 

```
myText = tkinter.StringVar()
```

To Label σχετίζεται με τη μεταβλητή αυτή:

```
myLabel=tkinter.Label(root, textvariable=myText)
```

Με τη μέθοδο `set()` του αντικειμένου `StringVar`:

```
myText.set("xyz") αλλάζει η τιμή του
```

Προγραμματίζω με την python

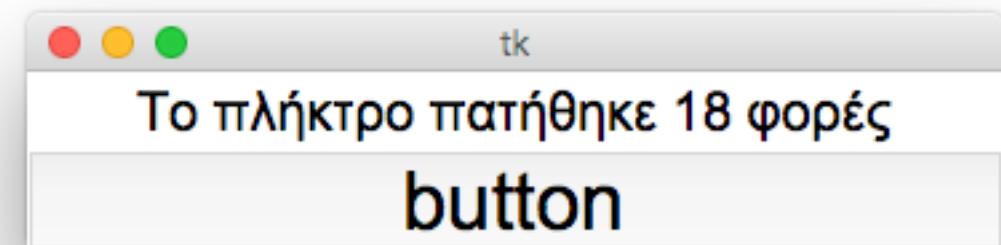


Αντικείμενο **StringVar**

```
import tkinter as tk

class MyApp():
    def __init__(self, root):
        self.r = root
        self.myText = tk.StringVar()
        self.myText.set(30*' ')
        self.mylabel = tk.Label(self.r, textvariable = self.myText, width=30,
                               font="Arial 20")
        self.mylabel.pack(fill='both', expand=1)
        self.b = tk.Button(self.r, text=" button ", font = "Arial 30",
                           command=self.buttonPressed)
        self.b.pack(fill='both', expand=1)
        self.count = 0
    def buttonPressed(self):
        pass
#-----
root = tk.Tk()
myapp = MyApp(root)
root.mainloop()
```

να γράψετε τον χειριστή
buttonPressed

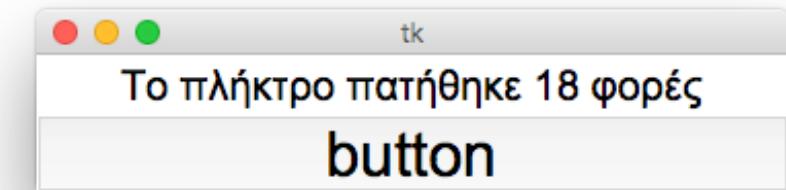


Αντικείμενο StringVar

ο χειριστής γεγονότων
buttonPressed

```
def buttonPressed(self):
    self.count += 1
    if self.count == 1 : end = 'ά '
    else: end = 'ές '
    self.myText.set('Το πλήκτρο πατήθηκε '
                   + str(self.count) + ' φορ' +end)
```

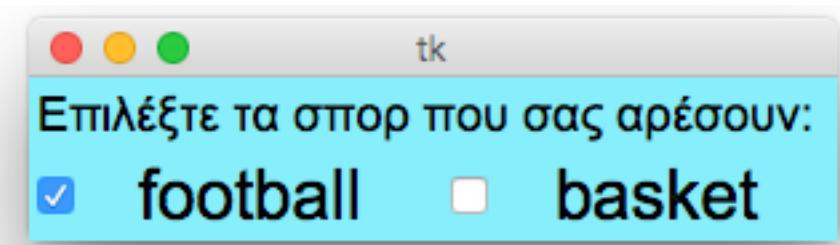
Προγραμματίζω
με την python 



Αντικείμενο Checkbutton

Ζητείται να κατασκευαστεί φόρμα όπως η παρακάτω, η οποία θα τυπώνει τις τιμές που έχουν οι μεταβλητές football και basket

Έχουν επιλεγεί:
Έχουν επιλεγεί: football



```
self.answer1 = tk.StringVar()  
self.check1 = tk.Checkbutton(self.root, text='football ',  
                           command=self.check, font="Arial 26",  
                           bg=self.color, variable=self.answer1,  
                           onvalue='football', offvalue='')
```

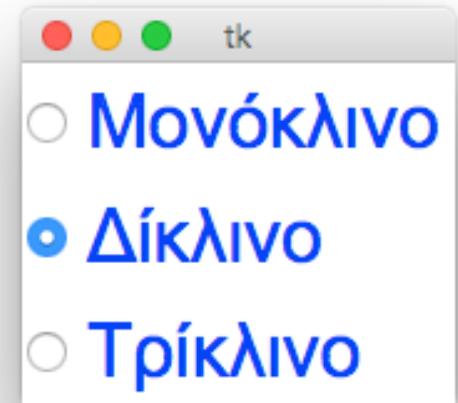
Προγραμματίζω
με την python 

Αντικείμενο Radiobutton

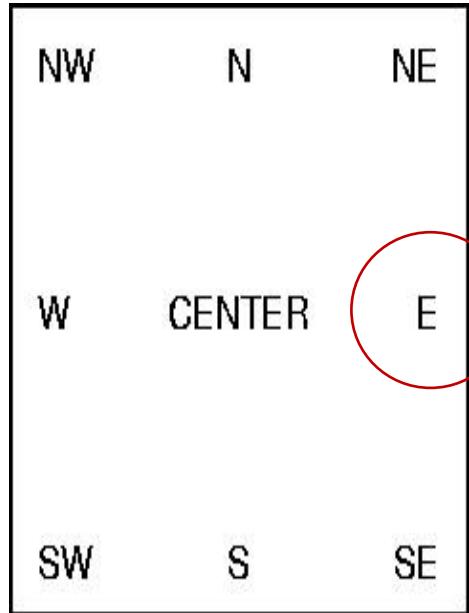
Αμοιβαία αποκλειόμενες επιλογές

```
import tkinter as tk
class App():
    def __init__(self, w):
        self.v = tk.IntVar()
        sel = [('Μονόκλινο',1), ('Δίκλινο',2),('Τρίκλινο',3)]
        for t,val in sel:
            tk.Radiobutton(w, text=t, font=('Arial', 30),
                           variable=self.v, fg= 'blue',
                           value=val, command=self.handle,
                           padx=5, pady=5).pack(anchor='w')
    def handle(self):
        print(self.v.get())
w=tk.Tk()
App(w)
w.mainloop()
```

Προγραμματίζω
με την python



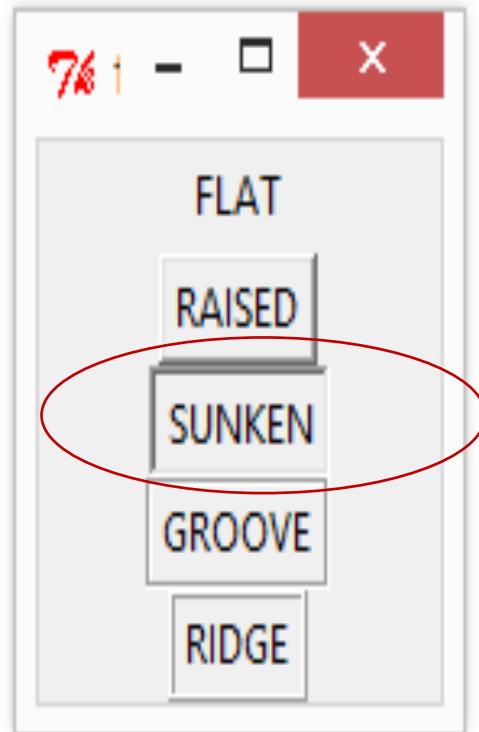
παράμετρος anchor



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

παράμετρος relief

w = widget_constructor (container, .. relief =)



```
B1 = tk.Button(top, text = "FLAT", relief="flat" )  
B2 = tk.Button(top, text = "RAISED", relief= "raised")  
B3 = tk.Button(top, text = "SUNKEN", relief= "sunken")  
B4 = tk.Button(top, text = "GROOVE", relief= "groove")  
B5 = tk.Button(top, text = "RIDGE", relief= "ridge")
```

παράμετρος cursor

Προγραμματίζω
με την python 

```
b = tk.Button(w, text ="circle", relief= "raised", cursor="circle")
```

 arrow	 man
 based_arrow_down	 middlebutton
 based_arrow_up	 mouse
 boat	 pencil
 bogosity	 pirate
 bottom_left_corner	 plus
 bottom_right_corner	 question_arrow
 bottom_side	 right_ptr
 bottom_tee	 right_side
 box_spiral	 right_tee
 center_ptr	 rightbutton
 circle	 rtl_logo

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

Προγραμματίζω
με την python



V4.1.6

Εικόνες

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

PhotoImage : εισαγωγή εικόνων

Προγραμματίζω
με την python 

Επιτρέπονται εικόνες μόνο τύπου gif και bitmap

Για εικόνες τύπου jpg, png απαιτείται η χρήση της πρόσθετης βιβλιοθήκης PIL (python image library)

Παράδειγμα : Εισαγωγή εικόνας σε widget τύπου Label.

```
self.image1 = tkinter.PhotoImage(file= "myimage.gif")
self.l= tkinter.Label(root, image=self.image1)
self.l.pack()
```

PhotoImage

Προγραμματίζω
με την python 

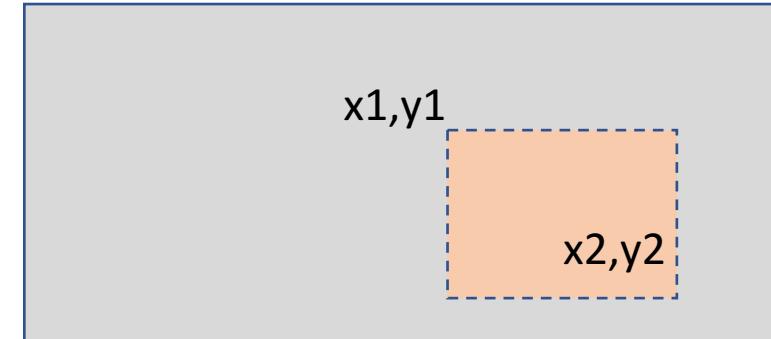
Εικόνες μπορεί να εισαχθούν σε αντικείμενα τύπου **Label**,
Text, **Canvas**, **Button**

Προσοχή: απαιτείται να τηρηθεί αναφορά στο αντικείμενο
PhotoImage για να μην διαγραφεί αυτόματα.

```
self.image = tk.PhotoImage(...)
self.l = tk.Label(image=photo)
self.l.pack()
self.root.image = self.image # keep a reference!
```

<http://effbot.org/pyfaq/why-do-my-tkinter-images-not-appear.htm>

PhotoImage



Προγραμματίζω
με την python 

Πώς μπορούμε να πάρουμε μόνο ένα τμήμα μιας
μεγαλύτερης εικόνας; (πχ από ένα animation sprite)

```
spritesheet = tk.PhotoImage(file= 'myfile.gif')

def subimage(x1, y1, x2, y2, spritesheet):
    dst = tk.PhotoImage()
    dst.tk.call(dst, 'copy', spritesheet, '-from', x1, y1, x2, y2, '-to', 0, 0)
    return dst
```

όπου τα x1,y1, x2,y2 ορίζουν τα όρια του ορθογώνιου τμήματος της αρχικής εικόνας

<https://stackoverflow.com/questions/16579674/using-spritesheets-in-tkinter>

πώς τροποποιούμε τις ιδιότητες ενός γραφικού αντικειμένου;

Προγραμματίζω
με την python



(A) Το αντικείμενο widget έχει ένα λεξικό με τις ιδιότητες του.
Μπορούμε να τροποποιήσουμε αυτό το λεξικό.

```
widget = tk.Label(self.root)  
widget['text'] = 'Καλή σας μέρα'
```

(B) Με κλήση της μεθόδου **config()**

```
widget = tk.Label(self.root)  
widget.config(text='Καλή σας μέρα')
```

Μάθημα L4.2

Υποδοχείς

Προγραμματίζω
με την python 

- V4.2.1 Υποδοχέας Frame
- V4.2.2 Υποδοχέας Notebook
- V4.2.3 Γεωμετρία grid
- V4.2.4 Άλλα γραφικά αντικείμενα

Προγραμματίζω
με την python



V4.2.1

Υποδοχέας Frame

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Υποδοχείς γραφικών αντικειμένων

Προγραμματίζω
με την python 

Tk (βασικό παράθυρο)

Toplevel (άλλα παράθυρα πρώτου επιπέδου)

Frame (ιεραρχικός διαχωρισμός περιοχών ενός παράθυρου)

LabelFrame (ένα Frame με πλαίσιο και τίτλο)

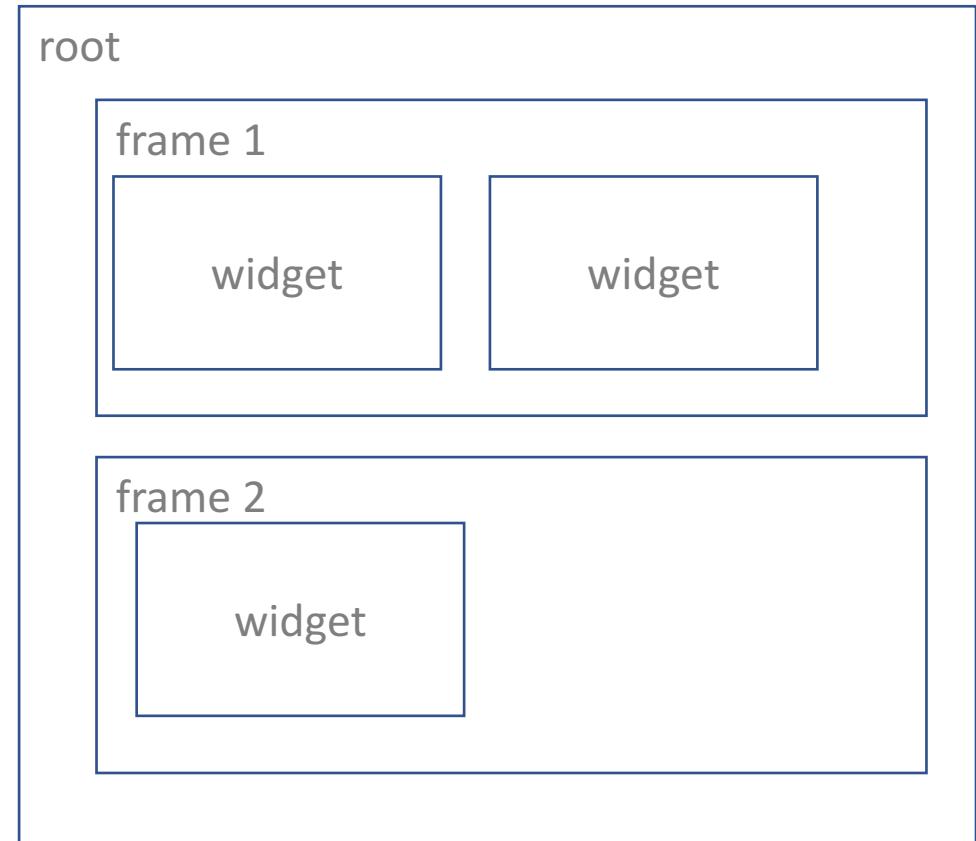
ttk.Notebook επάλληλες καρτέλες

Υποδοχείς Frame /LabelFrame

Προγραμματίζω
με την python 



```
self.f1 = tk.Frame(root)
self.f1.pack()
redb = tk.Label(self.f1, text=' Red')
redb.pack()
```



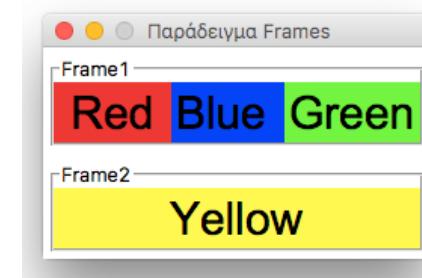
```

import tkinter as tk
class MyApp(tk.Frame):
    def __init__(self, root):
        self.root =root
        root.title('Παράδειγμα Frames')
        root.resizable(False, False)
        myfont = 'Arial 30'
        #Πρώτο πλαίσιο Frame
        self.f1 = tk.LabelFrame(root, text='Frame1')
        self.f1.pack(fill = 'both', expand=True, side='top', padx=5,pady=5)
        red = tk.Label(self.f1, text=' Red ', font=myfont, background="red")
        red.pack(fill = 'both', expand=1, side='left')
        blue = tk.Label(self.f1, text='Blue ', font=myfont, bg='blue')
        blue.pack(fill = 'both', expand=1, side='left')
        green = tk.Label(self.f1, text='Green', font=myfont, bg='green')
        green.pack(fill = 'both', expand=1, side='left')
        # Δεύτερο πλαίσιο Frame
        self.f2 = tk.LabelFrame(root, text='Frame2')
        self.f2.pack(side='bottom', fill='both', expand=1, padx=5, pady=5)
        yellow = tk.Label(self.f2, text='Yellow', font=myfont, bg='yellow')
        yellow.pack(expand=1, fill='both')
root = tk.Tk()
MyApp(root)

```

Frame και LabelFrame

Προγραμματίζω
με την python



Προγραμματίζω
με την python



V4.2.2

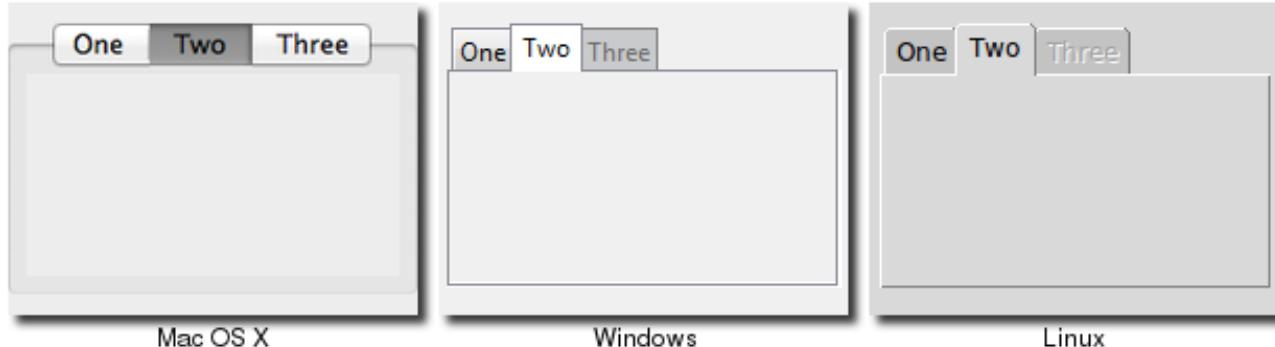
Υποδοχέας: ttk.Notebook

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

ttk.Notebook

επάλληλες καρτέλες

widget της υπο-βιβλιοθήκης ttk (themed tk)



```
from tkinter import ttk
self.note = ttk.Notebook(parent)
f1 = ttk.Frame(self.note)      # πρώτο παράθυρο
f2 = ttk.Frame(self.note)      # δεύτερο παράθυρο
self.note.add(f1, text='πρώτο')
self.note.add(f2, text='δεύτερο')
```

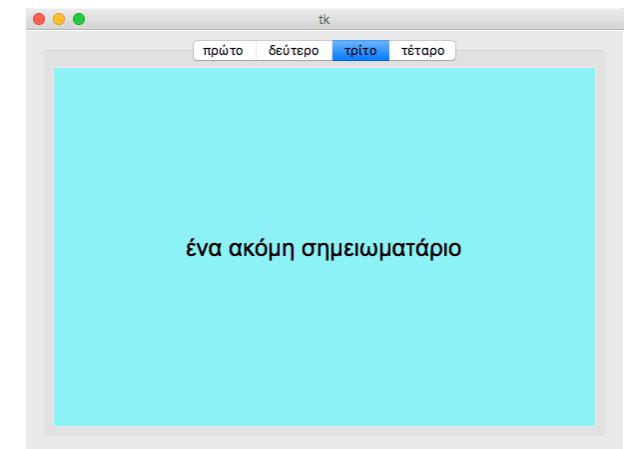
Προγραμματίζω
με την python 

Να δημιουργήσετε 4 επάλληλες καρτέλες με τυχαία χρώματα

```
class MyNotebook():
    def __init__(self, note, name, color):
        self.note = note
        self.f = ttk.Frame(self.note)
        self.f.pack(expand=True, fill='both', padx=1, pady=1)
        self.l = tk.Label(self.f, text='ένα ακόμη σημειωματάριο', bg=color, font='Arial 24')
        self.l.pack(expand=True, fill='both', padx=1, pady=1)
        self.note.add(self.f, text=name)

class App():
    def __init__(self, root):
        self.root = root
        self.root.geometry('600x800+100+100')
        self.note = ttk.Notebook(self.root)
        self.note.pack(expand=True, fill='both', padx=1, pady=1)
        r = lambda : random.randint(0,255)
        mynotebooks = []
        for name in ['πρώτο', 'δεύτερο', 'τρίτο', 'τέταρο']:
            mynotebooks.append(MyNotebook(self.note, name, '#{:02X}{:02X}{:02X}'.format(r(),r(),r())))
```

Προγραμματίζω
με την python



Προγραμματίζω
με την python



V4.2.3

Γεωμετρία grid

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Εναλλακτικός τρόπος διάταξης widget : ο διαχειριστής γεωμετρίας **grid**

Προγραμματίζω
με την python



Αντί για την μέθοδο pack() μπορεί να χρησιμοποιηθεί η grid()

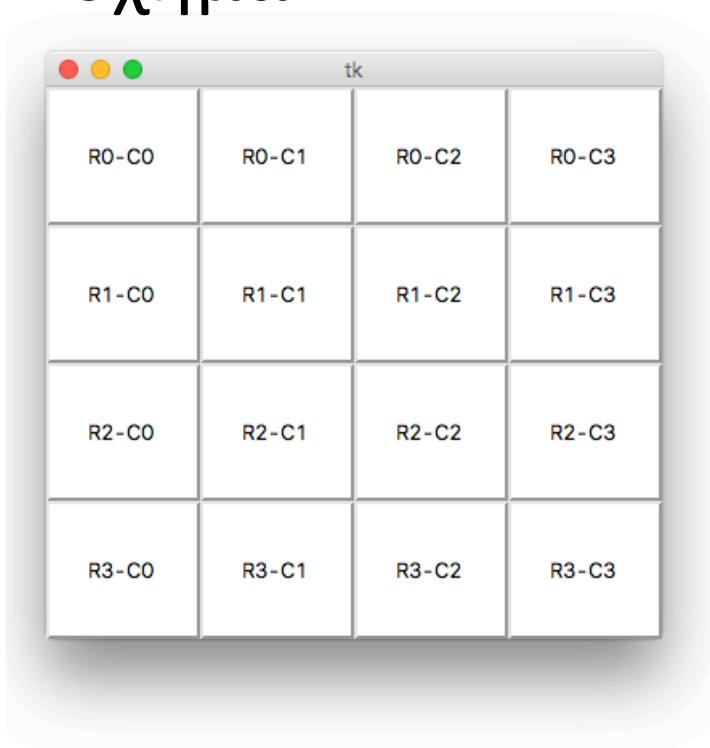
w.grid (row = 0, column = 0, rowspan=3)

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

Παράδειγμα: διάταξη widgets με τον διαχειριστή γεωμετρίας grid

Να διαταχθούν 16 Label σε πίνακα όπως στο σχήμα



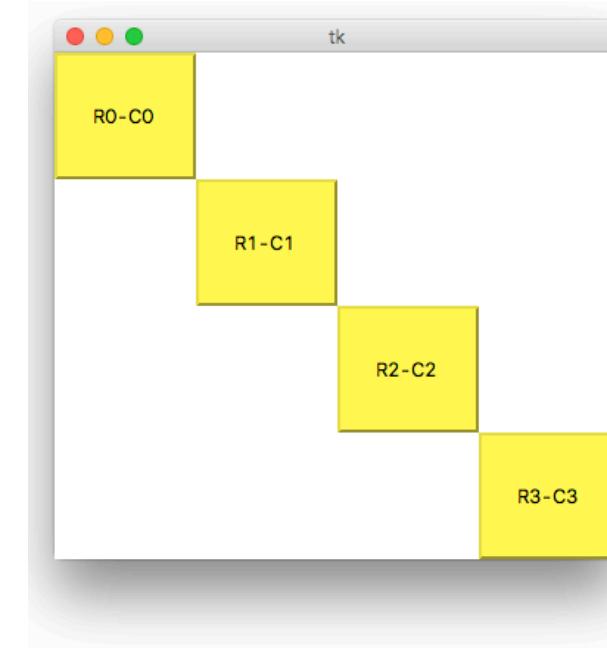
```
import tkinter as tk
class App():
    def __init__(self, root):
        for r in range(4):
            for c in range(4):
                lab = tk.Label(root, width=10, height=5,
                               text='R{}-C{}'.format(r, c),
                               borderwidth=2, relief="raised")
                lab.grid(row=r, column=c)
root = tk.Tk()
App(root)
root.mainloop()
```

Προγραμματίζω
με την python 



Άσκηση:

Τροποποιήστε τον κώδικα ώστε να διαταχθούν 4 Label σε πίνακα μόνο στη διαγώνιο, πειραματιστείτε με παραμέτρους για το πλαίσιο, **borderwidth** και **relief**



Προγραμματίζω
με την python



V4.2.4

Άλλα γραφικά αντικείμενα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Κλάσεις σύνθετων γραφικών αντικειμένων (widget) της tkinter

Προγραμματίζω με την python 

Menubutton Πλήκτρο που ενεργοποιεί ένα Menu για περαιτέρω επιλογές και υπό-μενού

Menu Σύνολο επιλογών σχετισμένων με ένα Menubutton ή παράθυρο

ttk.Combobox Αναδυόμενη λίστα τιμών για επιλογή

Listbox Λίστα για επιλογή μεταξύ αντικειμένων

Scrollbar Αντικείμενο που επιτρέπει την κύλιση άλλων widgets (πχ. listbox, canvas, text)

Text Σύνθετο αντικείμενο για επεξεργασία και παρουσίαση κειμένου πολλών γραμμών

Canvas Αντικείμενο που ορίζει περιοχή γραφικής σχεδίασης που υποστηρίζει γραμμές, κύκλους, εικόνες, κείμενο κλπ.

ttk (themed tk)

Προγραμματίζω
με την python 

ttk.Style ορισμός στυλ

```
ttk.Style().configure('button.TButton', bg= "yellow", font='Arial 30')
root.title('Παράδειγμα 6')
self.root = root
self.button = ttk.Button(self.root, style='button.TButton', text=" show text ",
                        command=self.showText)
self.button.pack(fill='both', expand=1)
self.entry = tk.Entry(self.root, font = "Arial 30", width= 20)
self.entry.pack(fill='both', expand=1)
```

Από την έκδοση 8.5 περιλαμβάνει themes, styles, εμφάνιση στα
widgets όπως αυτή του λειτουργικού, επίσης πρόσθετα στοιχεία
όπως το combobox

ttk widgets

Προγραμματίζω
με την python 

Ttk περιέχει 17 widgets, τα 11 υπήρχαν ήδη στην tkinter:
Button, Checkbutton, Entry, Frame, Label, LabelFrame,
Menubutton, PanedWindow, Radiobutton, Scale and
Scrollbar.

Υπάρχουν 6 νέα widgets
Combobox, Notebook, Progressbar, Separator, Sizegrip
and Treeview. Όλα είναι υπόκλασεις του Widget.

Μάθημα L4.3

Μενού και διάλογοι

Προγραμματίζω
με την python 

- V4.3.1 Δημιουργία μενού
- V4.3.2 Combobox
- V4.3.3 Διάλογοι
- V4.3.4 Διαχείριση γεγονότων bind()

Προγραμματίζω
με την python



V4.3.1 Μενού

Νίκος Αβούρης, Πανεπιστήμιο Πατρών



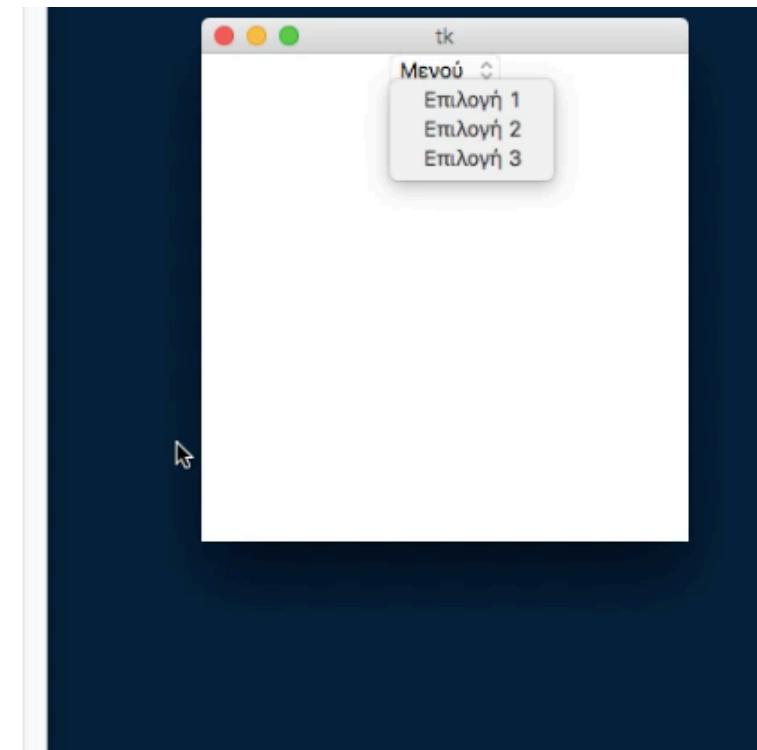
Δημιουργία μενού

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

1. (η πιο συνηθισμένη περίπτωση): μενού σχετιζόμενο
με Menubutton

```
>>> import tkinter as tk
>>> w = tk.Tk()
>>> w.geometry('300x300')
''

>>> mb = tk.Menubutton(w, text = 'Μενού')
>>> mb.pack()
>>> m = tk.Menu(mb)
>>> m.add_command(label='Επιλογή 1')
>>> m.add_command(label='Επιλογή 2')
>>> m.add_command(label='Επιλογή 3')
>>> mb.config(menu=m)
```



Δημιουργία μενού

Προγραμματίζω
με την python 

2. μενού σχετιζόμενο με παράθυρο

```
menubar = tk.Menu(self.root) # σύνδεση menubar-παράθυρου  
self.root.config(menu = menubar) # σύνδεση παράθυρου-menubar
```

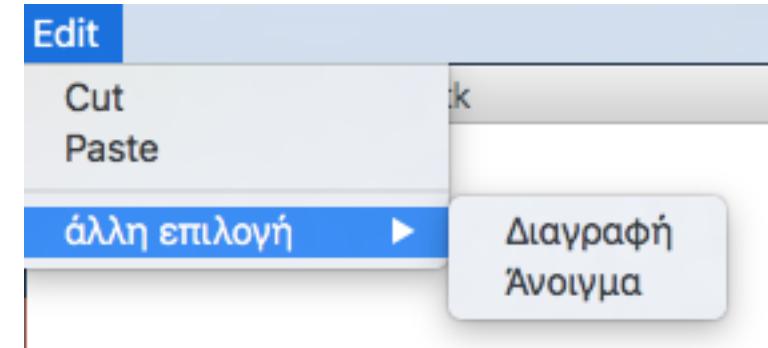
```
# δημιουργία αναδυόμενου μενού και σύνδεση με menubar  
filemenu = tk.Menu(menubar, tearoff=0)  
menubar.add_cascade(label="File", menu=filemenu)  
# επιλογές του μενού filename  
filemenu.add_command(label="Open", command=self.open)  
filemenu.add_command(label="Save", command=self.save)  
filemenu.add_separator()  
filemenu.add_command(label="Exit", command=root.quit)
```

Δημιουργία υπό-μενού

Έαν σε κάποια επιλογή θέλουμε να δημιουργήσουμε ένα υπό-μενού, τότε καλούμε τη μέθοδο `add_cascade()`

```
submenu = tk.Menu(edit, tearoff=True)
edit.add_cascade(label='άλλη επιλογή', menu=submenu, underline=0)
submenu.add_command(label='Διαγραφή', command=root.quit,
underline=0)
submenu.add_command(label='Άνοιγμα', command=self.notdone,
underline=0)
```

Προγραμματίζω
με την python



Επιλογές μενού

```
m = tk.Menu( ...)
```

`m.add_command(option, ...)` πρόσθεσε επιλογή στο μενού με χρήση ορισμάτων label, bitmap, ώστε να ποθετήσεις κείμενο, εικόνες στο μενού - επιλογή command για να συνδέσεις την επιλογή με συνάρτηση

`m.add_cascade(option, ...)` πρόσθεσε ένα μενού επόμενου επιπέδου

`m.add_checkbutton(option, ...)` πρόσθεσε ένα αντικείμενο επιλογής τύπου checkbutton

`m.add_radiobutton(option, ...)` πρόσθεσε ένα αντικείμενο επιλογής τύπου radiobutton

`m.add_separator()` πρόσθεσε μια διαχωριστική γραμμή

Προγραμματίζω
με την python



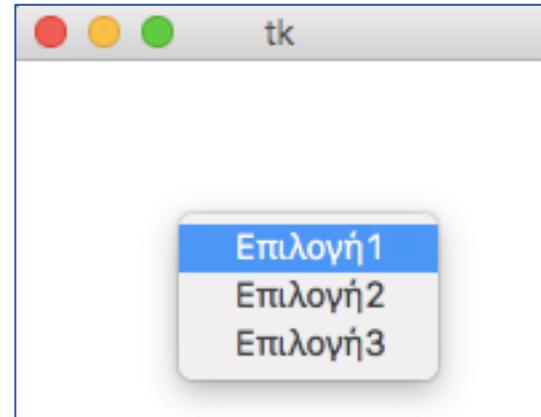
Δημιουργία τοπικού αναδυόμενου μενού (pop up)

Προγραμματίζω με την python

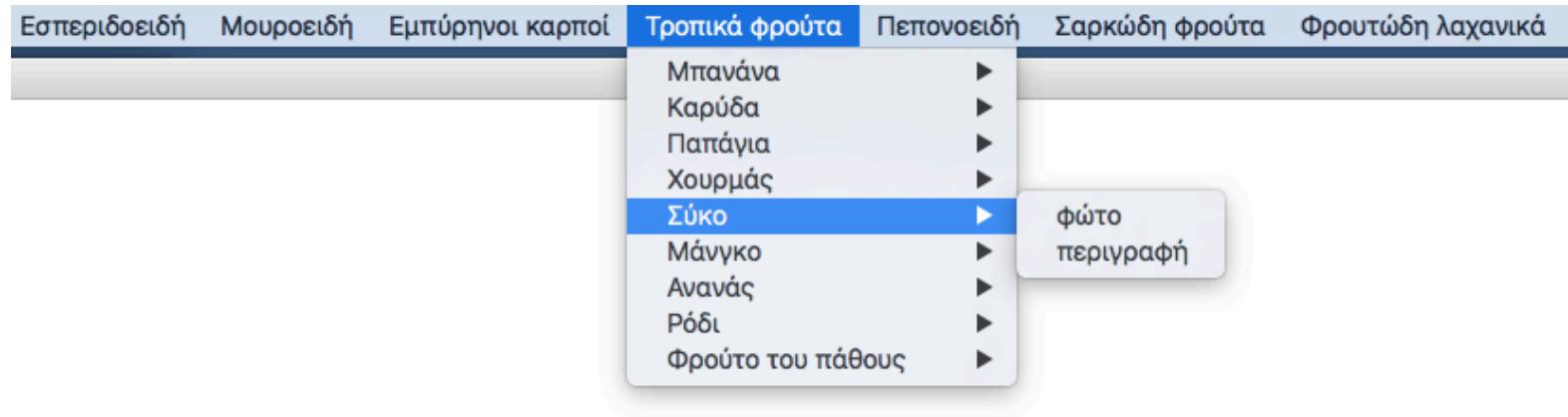


Συνήθως σχετίζεται με δεξί κλειδί ή παρατεταμένη δείξη σε οθόνες αφής. Χρησιμοποιείται η μέθοδος **menu.post(x,y)** και οι συντεταγμένες **event.x_root**, **event.y_root** (τοπικές συντεταγμένες)

```
self.menu = tk.Menu(root)
for i in ('Επιλογή1', 'Επιλογή2', 'Επιλογή3'):
    self.menu.add_command(label=i)
# έλεγχος για τη διαφορά στο δεξί πλήκτρο μεταξύ λειτουργικών συστημάτων
root.bind('<2>', lambda e: self.menu.post(e.x_root, e.y_root))
```



άσκηση



Έστω μια λίστα με φρούτα (αρχείο fruit.txt) ως
ιεραρχία 2 επιπέδων. Για κάθε φρούτο να προσθέσετε
ακόμη 2 επιλογές: (α) περιγραφή και (β) φωτο.
Να κατασκευάσετε μενού αναζήτησης φρούτων 3
επιπέδων.

Προγραμματίζω
με την python



άσκηση

Προγραμματίζω
με την python 

Να κατασκευάσετε ένα μενού μιας επιλογής
που επαναλαμβάνεται ως βάθος η
το κάθε επόμενο επίπεδο i να έχει label =
Επιλογή-i

Προγραμματίζω
με την python



V4.3.2 Combobox

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Combobox

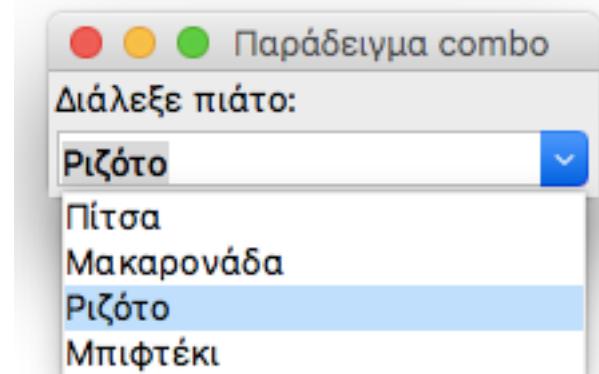
Προγραμματίζω
με την python 

Σύνθετο widget που ανήκει στην υπο-βιβλιοθήκη
ttk

Επιτρέπει την επιλογή από μια λίστα τιμών

```
from tkinter import ttk

def combo(self):
    self.box_value = tk.StringVar()
    self.box = ttk.Combobox(self.root, textvariable=self.box_value, state= 'readonly')
    self.box.bind("<<ComboboxSelected>>", self.newselection)
    self.box[ 'values' ] = ('Πίτσα', 'Μακαρονάδα', 'Ριζότο', "Μπιφτέκι")
    self.box.current(0)
    self.box.pack(expand=1, fill='both')
```



Προγραμματίζω
με την python



V4.3.3 Treeview

Νίκος Αβούρης, Πανεπιστήμιο Πατρών



Treeview

Σύνθετο widget που ανήκει στην υπο-βιβλιοθήκη ttk
Επιτρέπει την παρουσίαση πληροφορίας ιεραρχικής δομής

ορισμός στηλών

```
from tkinter import ttk
```

```
tree = ttk.Treeview(viewer, style = 'Treeview')
tree["columns"]=( "a","b")
tree.column("a", width=100, stretch = True )
tree.column("b", width=50, anchor='e', stretch = False)
tree.heading("a", text="Στήλη 1")
tree.heading("b", text="Στήλη 2")
```

	Size	Modified
widgets	25KB	Yesterday
gallery	2KB	Two weeks ago
resources	220KB	Three weeks ago
tutorial	2.1MB	Ten minutes ago
canvas	18KB	Last week
tree	5KB	Ten minutes ago
text	12KB	Yesterday

Mac OS X

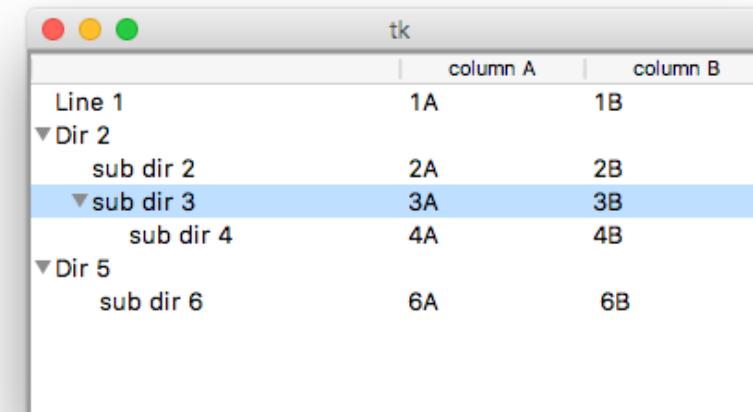
	Size	Modified
widgets	25KB	Yesterday
gallery	2KB	Two weeks ago
resources	220KB	Three weeks ago
tutorial	2.1MB	Ten minutes ago
canvas	18KB	Last week
tree	5KB	Ten minutes ago
text	12KB	Yesterday

Windows

	Size	Modified
widgets	25KB	Yesterday
gallery	2KB	Two weeks ago
resources	220KB	Three weeks ago
tutorial	2.1MB	Ten minutes ago
canvas	18KB	Last week
tree	5KB	Ten minutes ago
text	12KB	Yesterday

Linux

Treeview



Προγραμματίζω
με την python 

εισαγωγή εγγραφών

```
tree.insert("", 0, text="Line 1", values=("1A", "1B")) # εισαγωγή πρώτου επιπέδου
```

```
id2 = tree.insert("", "end", text="Dir 2") # εισαγωγή πρώτου επιπέδου
tree.insert(id2, "end", text="sub dir 2", values=("2A", "2B"))
```

```
id3 = tree.insert(id2, "end", text="sub dir 3", values=("3A", "3B"))
tree.insert(id3, "end", text="sub dir 4", values=("4A", "4B"))
```

```
tree.insert("", 3, iid="dir5", text="Dir 5") # εισαγωγή με ορισμό του iid
tree.insert("dir5", 3, text=" sub dir 6", values=("6A", " 6B"))
```

Προγραμματίζω
με την python



V4.3.4

Τα εγγόνια του Κολοκοτρώνη

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Εφαρμογή της Treeview: Αναζητάμε τα εγγόνια του Κολοκοτρώνη

Προγραμματίζω
με την python



Χωρίς Σύνδεση Συζήτηση για αυτή την IP Συνεισφορές Δημιουργία λογαριασμού Σύνδεση

Λήμμα Συζήτηση Ανάγνωση Επεξεργασία Περισσότερα Αναζήτηση σε Βικιπαίδεια

Οικογένεια Κολοκοτρώνη

Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια

Το λήμμα δεν περιέχει **μηνύές** ή αυτές που περιέχει δεν επαρκούν. Μπορείτε να βοηθήσετε προσθέτοντας την κατάλληλη τεκμηρίωση. Υλικό που είναι απεκμηρωτό μπορεί να αφαιριθεί και να αφαιρεθεί.

Η αιματηση τοποθετήθηκε στις 16/06/2012.

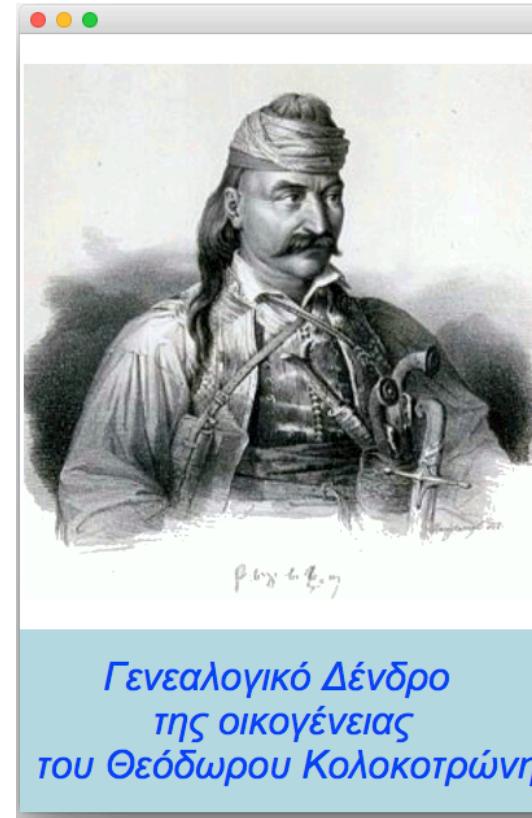
Οικογένεια Κολοκοτρώνη είναι τοπικά ελληνική οικογένεια κλεφτών, η οποία ανέδειξε πολιτικούς και κυρίως στρατιωτικούς. Σημαντικότερος εκπρόσωπός της είναι ο στρατηγός Θεόδωρος Κολοκοτρώνης.

Πίνακας περιεχομένων [Απόκρυψη]

- 1 Οικογενειακή παράδοση
- 2 Μέλη
- 3 Άλλοι
- 4 Παραπομπές

Οικογενειακή παράδοση [Επεξεργασία | επεξεργασία κώδικα]

Ο Θεόδωρος Κολοκοτρώνης στα υπαγεμένα απομνημονεύματά του δίνει^[1] σημαντικά στοιχεία για την ιστορία της οικογένειάς του πριν από αυτόν. Το αρχικό επώνυμό της ήταν **Τσεργίνης**. Στην συνέχεια ο Δήμος Τσεργίνης που έχει την εποχή της Βενετοκρατίας στην Πελοπόννησο(1685-1715), ονομάστηκε Μήποτακας. Ο γιος του



Lithography by Karl Krazeisen (1794-1878) από el.wikipedia.org

Η οικογένεια του Θ.Κολοκοτρώνη		
Η ζωή του	Απόγονοι	
Θεόδωρος Κολοκοτρώνης	Στρατηγός του 1821	5
Πάνος Κολοκοτρώνης	Αγωνιστής του 1821	0
► Ιωάννης (Γενναίος) Κολοκοτρώνη	Στρατιωτικός, πρωθυπουργός της Ελλάδας, σ	4
Κωνσταντίνος Κολοκοτρώνης	Δικηγόρος	0
Γεωργία Κολοκοτρώνη	Σύζυγος του Λεωνίδα Πετιμεζά	0
Θεόδωρος Φαλέζ - Κολοκοτρώνη	Στρατιωτικός, πολιτικός, σύζυγος Αγγελικής Κ	0
► Αικατερίνη Κολοκοτρώνη (1847)	Σύζυγος Ιωάννη Παναγιώτη Ρόδιου	4
Πάνος Κολοκοτρώνης	Συνταγματάρχης, αστυνομικός διευθυντής Αθ	1
► Γεώργιος Κολοκοτρώνης	Αξιωματικός του στρατού	1
► Κωνσταντίνος Θ. Κολοκοτρώνης	Στρατιωτικός, βουλευτής, σύζυγος Ραλλούς Κ	5
► Σμαράγδα Κολοκοτρώνης	Σύζυγος Ανδρέα Ψύλλ	3
Αικατερίνη Κολοκοτρώνη	Κυρία επί των Τιμών της Βασίλισσας Όλγας, σ	0
► Ελένη Κολοκοτρώνη	Σύζυγος Νικολάου Αντωνιάδη	1
Ιωάννης Κολοκοτρώνης	Στρατιωτικός, ήρωας πολέμου	0
► Θεόδωρος Κ. Κολοκοτρώνης	Αρχηγός της αστυνομίας Αθηνών, σύζυγος τη	2
► Ελένη Κολοκοτρώνη	Σύζυγος του Νικήτα Δικαίου	1

Προγραμματίζω
με την python



V4.3.5 Διάλογοι

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Dialog box

Προγραμματίζω
με την python 

Ένα παράθυρο διαλόγου dialog box είναι ένα μονοτροπικό (modal) παράθυρο που 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, προσφέρονται από το γραφικό περιβάλλον και για αυτό είναι ίδια σε πολλές εφαρμογές.

παράθυρα διαλόγου

Προγραμματίζω
με την python 

from tkinter import messagebox

(απλά μηνύματα πληροφόρησης ή ερώτησης ναι/όχι)

from tkinter import simpledialog

Ζητάει ένα αριθμό ή κείμενο

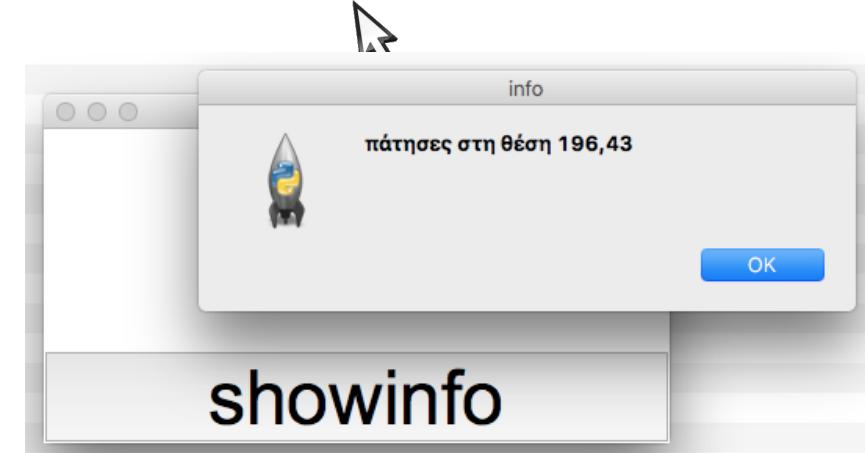
from tkinter import askopenfilename #αρχείο

from tkinter import colorchooser #παλέτα χρωμάτων

messagebox.showinfo

```
import tkinter as tk
from tkinter import messagebox
class App():
    def __init__(self, root):
        self.root = root
        self.root.geometry('400x200+200+200')
        self.widgets()
    def widgets(self):
        self.f = tk.Frame(self.root)
        self.f.pack(expand=True, fill = 'both')
        self.b1 = tk.Button(text='showinfo', font='Arial 48')
        self.b1.pack(side= 'left', expand=True, fill = 'both')
        self.b1.bind('<Button-1>', self.showinfo)
    def showinfo(self, event):
        messagebox.showinfo('info',
                           'πάτησες στη θέση {},{}'.format(event.x, event.y))
root = tk.Tk()
App(root)
root.mainloop()
```

Προγραμματίζω
με την python



Άλλα παράθυρα διαλόγου:
askyesno, askokcancel,
askretrycancel, askquestion

Προγραμματίζω
με την python 

```
from tkinter import messagebox
```

```
ans = messagebox.askyesno("Continue", "Should I continue?")
```

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

Προσοχή: messagebox. askquestion επιστρέφει "yes" "no" και όχι True / False!

Διάλογος εισαγωγής τιμής από το χρήστη: askstring, askinteger

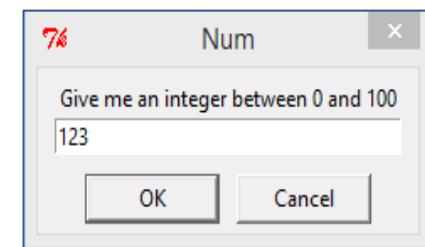
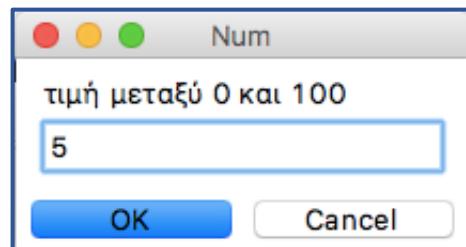
Προγραμματίζω
με την python 

```
from tkinter import simpledialog
```

```
ans = simpledialog.askstring("Title", "Το όνομά σας:")
```

```
ans = simpledialog.askinteger("Dialog Title", "Δώστε ένα ακέραιο")
```

```
ans = simpledialog.askinteger("Num", "τιμή μεταξύ 0 και 100", minvalue=0,  
maxvalue=100)
```



Άνοιγμα αρχείου: askopenfilename

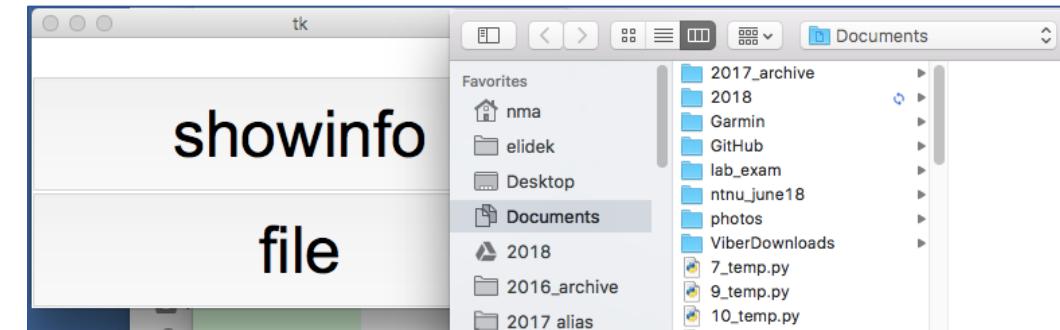
```
from tkinter import filedialog
```

```
from tkinter import filedialog
```

```
...
```

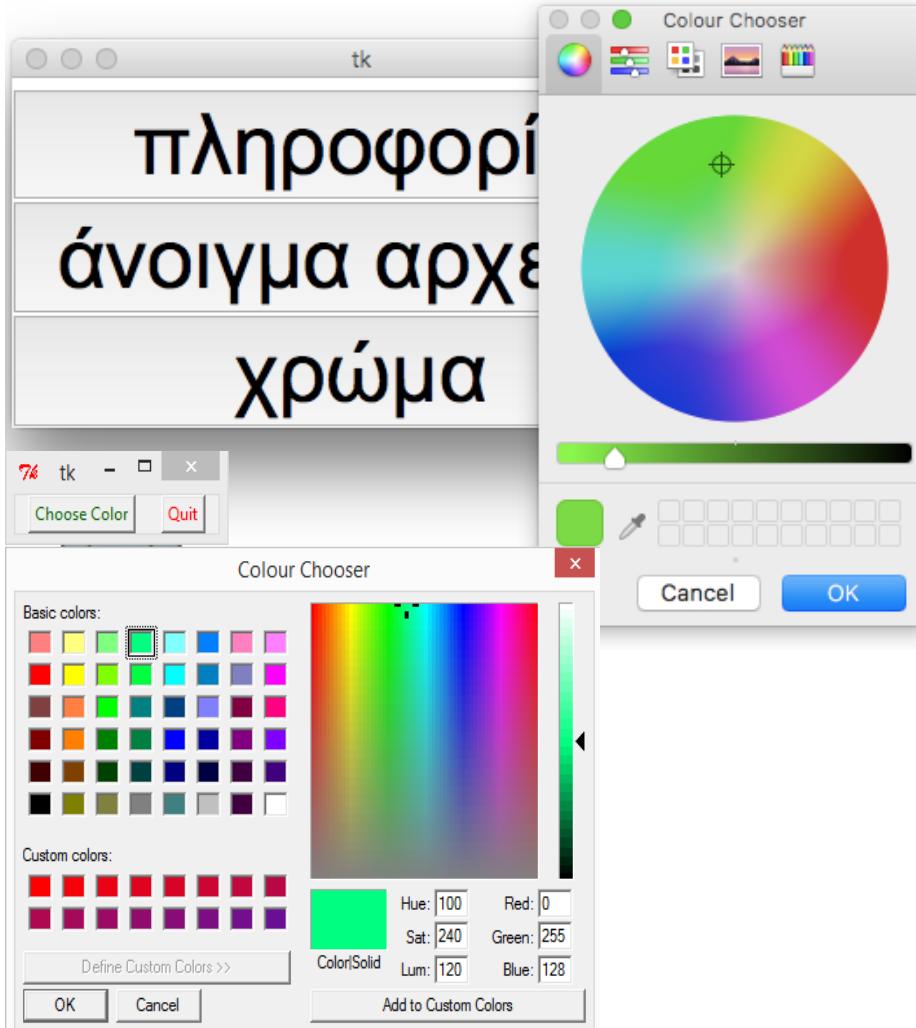
```
    self.b2 = tk.Button(text='file', font='Arial 48')
    self.b2.pack(side='top', expand=True, fill='both')
    self.b2.bind('<Button-1>', self.file)
def file(self, event):
    name = filedialog.askopenfilename()
    print(name)
```

Προγραμματίζω
με την python



Επιλογή χρώματος : askcolor

```
from tkinter import colorchooser
```



Προγραμματίζω
με την python



```
self.b3 = tk.Button(text='χρώμα', font='Arial 48')
self.b3.pack(side='top', expand=True, fill='both')
self.b3.bind('<Button-1>', self.color)
def color(self, event):
    print('color', event)
    result =
        colorchooser.askcolor(color=self.current_color,
title="Colour Chooser")
    self.current_color = result[-1]
```

Προγραμματίζω
με την python



V4.3.6

Διαχείριση γεγονότων bind()

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

`widget.bind(event, handler)`

Όταν συμβεί το γεγονός **event** στο γραφικό αντικείμενο **widget** κάλεσε τη συνάρτηση-χειριστή **handler**.

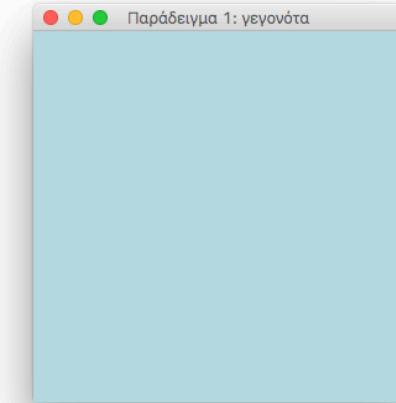
Η συνάρτηση παίρνει ως όρισμα μεταβλητή **event** που αφορά χαρακτηριστικά του γεγονότος, όπως οι συντεταγμένες **event.x**, **event.y**

```
self.f = tk.Frame(self.root, width=300, height=300, borderwidth=10, bg='lightblue')
self.f.pack(expand=True, fill='both')
```

```
self.f.bind("<Button-2>", lambda event: print('2 click at', event.x, event.y))
self.f.bind("<Button-1>", lambda event: print('1 click at', event.x, event.y))
```

Προγραμματίζω
με την python 

```
/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6 ,
1 click at 125 131
1 click at 125 131
2 click at 107 215
2 click at 80 254
1 click at 239 72
2 click at 78 171
1 click at 24 70
```



ΤΥΠΙΚΑ γεγονότα

bind(event, χειριστής)

<Button-1> ή <1>
<ButtonRelease-1>
<Button-3> ή <3>
<Double-Button-1>

πάτημα αριστερού πλήκτρου ποντικιού <B1>
ελευθέρωση αριστερού πλήκτρου
δεξί πλήκτρο ποντικιού/ προσοχή! <2> σε mac
διπλό κλικ

<Enter>
<Leave>
<Return>
<key>
<B1-Motion>
<FocusIn> <FocusOut>
<Up><Down><Left><Right>
<KeyRelease><KeyPress>

το ποντίκι εισέρχεται στο widget
το ποντίκι αφήνει το widget
πάτημα πλήκτρου enter
(α για παράδειγμα) – πάτημα πλήκτρου "a"
μετακίνηση του ποντικιού με πατημένο το Button-1
Έστίαση / απώλεια εστίασης
βέλη κίνησης
Ελευθέρωση-πάτημα πλήκτρου

Προγραμματίζω
με την python 

Άσκηση: περιγράψτε τον κώδικα

```
import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.root = root
        root.title("Παράδειγμα 8: γεγονότα")
        root.geometry("400x300+300+300")
        self.create_widgets()
    def create_widgets(self):
        self.b = tk.Button(self.root, text='Logon', font = 'Arial 40')
        self.b.pack(expand=True, fill='both')
        self.b.bind("<Enter>", self.mouse_entered)
        self.b.bind("<Leave>", self.mouse_exited)
    def mouse_exited(self, event):
        self.b.config(text= 'Logon')
    def mouse_entered(self, event):
        self.b.config(text='Click!')
    def main():
        root = tk.Tk()
        app = MyApp(root)
        root.mainloop()
if __name__ == '__main__': main()
```

Προγραμματίζω
με την python 

Άσκηση

(για συζήτηση)

Προγραμματίζω
με την python 

Να κατασκευάσετε πρόγραμμα Python που υλοποιεί την εφαρμογή μετατροπής θερμοκρασιών από Κελσίου σε Φαρενάιτ, Κέλβιν και αντίθετα με γραφικό τρόπο (να χρησιμοποιήσετε **Combobox** για επιλογή κλίμακας, **Entry** για εισαγωγή τιμής και **Label** για εμφάνιση αποτελέσματος

- $F = C * (9/5) + 32$
- $C = (F-32) * (5/9)$
- $K = (F+459.67) * (5/9),$
- $F = 300 * K * (9/5) - 459.67$

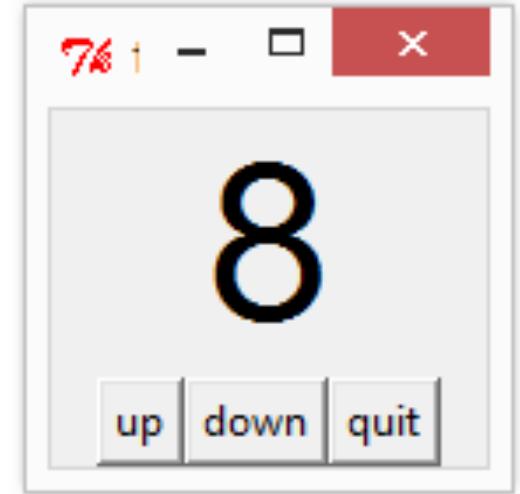
Άσκηση

(για συζήτηση)

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 κατώτερη τιμή 1) και η επιλογή **quit** κλείνει η εφαρμογή.

Προγραμματίζω
με την python 



πηγές

Προγραμματίζω
με την python 

<https://docs.python.org/3/library/tk.html>

<http://effbot.org/tkinterbook/>

<http://www.tkdocs.com/index.html>

https://www.tutorialspoint.com/python3/python_gui_programming.htm

<http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>