

Event-based programming

Προγραμματισμός με γεγονότα

Κάθε πρόγραμμα είναι ένας βρόχος (loop) επανάληψης:

While True:

 Σχεδίασε γραφικά αντικείμενα

 Περίμενε event (ενέργεια) από τον χρήστη

 Ανάλογα με το event :

 Κάλεσε την συνάρτηση χειριστή (event_handler)

tkinter (tk interface) : Βιβλιοθήκη για γραφικές εφαρμογές της python που υλοποιεί τον προγραμματισμό γεγονότων.

- η tkinter είναι η διεπαφή της python 3 με τη γραφική βιβλιοθήκη tk (tcl/tk)
- περιλαμβάνεται στις βασικές βιβλιοθήκες της python και δεν χρειάζεται εγκατάσταση
- Υπάρχουν και άλλες βιβλιοθήκες όπως π.χ. WxPython, PyGtk, PyQt κ.ά

GUI (Graphical User Interface)

Με την tkinter δημιουργούμε παράθυρα που περιέχουν γραφικά στοιχεία **widgets** όπως labels, buttons, entry texts, menu, scroll bars, κá

Τα widgets δέχονται γεγονότα(events) από ενέργειες του χρήστη και ανταποκρίνονται με κλήση συναρτήσεων (event_handlers)

Η βασική λογική ενός προγράμματος

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

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

While True: # περίμενε για πάντα

 Όταν εμφανιστεί ένα γεγονός

 κάλεσε τον event_handler με όρισμα το <event object>

Σημείωση: πολλά γεγονότα τα διαχειρίζεται η ίδια tkinter, όπως το κλείσιμο παραθύρου όταν πατήσουμε στο x

```
import tkinter as tk
w = tk.Tk()
w.mainloop()
```

```
import tkinter as tk
color="#ff0000"
w = tk.Tk()
w.config(bg=color)
w.mainloop()
```

```
import tkinter as tk
w = tk.Tk()
w.config(bg='yellow')
l = tk.Label(w, bg='green', text='Γεια σας, τι κάνετε;', font='Arial 30')
l.pack()
w.mainloop()
```

Πρώτο πρόγραμμα με αντικείμενα (objects)

```
#Παράδειγμα με Label
import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.root = root
        self.root.title('Παράδειγμα')
        self.widgets()
    def widgets(self):
        self.w = tk.Label(self.root, text=" Καλή σας μέρα! ",font = "Arial 36",
bg='yellow')
        self.w.pack()

root = tk.Tk() # αρχικό παράθυρο
myapp = MyApp(root)
root.mainloop()

#Παράδειγμα με Label και Color
import tkinter as tk
class App():
    def __init__(self, root):
        self.root = root
        self.font = 'Arial 40'
        self.root.title('Παράδειγμα 1')
        self.widgets()
    def widgets(self):
        color = 'green'
```

```

        self.l = tk.Label(self.root, text='Καλημέρα σας!', font =self.font,
bg=color)
        print(color)
        self.l.pack()
root = tk.Tk() # αρχικό παράθυρο
App(root)
root.mainloop()

```

```

#Παράδειγμα με random color
import tkinter as tk
import random
class App():
    def __init__(self, root):
        self.root = root
        self.font = 'Arial 40'
        self.root.title('Παράδειγμα 1')
        self.widgets()
    def widgets(self):
        # τυχαίο χρώμα
        c1 = random.randint(0,255)
        c2 = random.randint(0,255)
        c3 = random.randint(0,255)
        color = '#{0:02X}{1:02X}{2:02X}'.format(c1, c2, c3)
        self.l = tk.Label(self.root, text='Καλημέρα σας!', font =self.font,
bg=color)
        print(color)
        self.l.pack()
root = tk.Tk() # αρχικό παράθυρο
App(root)
root.mainloop()

```

Σημείωση : οι εντολές με τα κίτρινα γράμματα θα μπορούσαν να αντικατασταθούν από τις παρακάτω. Ουσιαστικά φτιάχνουμε μια συνάρτηση r() όπου επιστρέφει ένα τυχαίο χρώμα όποτε χρησιμοποιηθεί

```

r =lambda : random.randint(0,255)
color = '#{0:02X}{1:02X}{2:02X}'.format(r(), r(), r())

```

Μηχανή γεωμετρίας pack()

Είναι ο μηχανισμός μέσω του οποίου ορίζεται ο χώρος και η θέση κάθε αντικειμένου μέσα σε έναν υποδοχέα.

```

# Παραδείγματα με την pack
# παράθυρο διαστάσεων 200x200 pixel

```

```

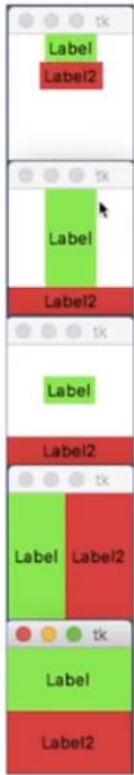
# στο οποίο τοποθετούμε 2 γραφικά αντικείμενα, 2 labels : ένα πράσινο και ένα κόκκινο
import tkinter as tk
class MyApp():
    def __init__(self, root):
        root.geometry('200x200+200+200') # 200x200 διαστάσεις παραθύρου(x,y) 200+200
        συνταγμένες
        l1=tk.Label(root, text='Iasonas', bg='green')
        l1.pack() # ο χώρος που καταλαμβάνει είναι ανάλογος με το περιεχόμενο (λέξη
        Iasonas)
        l2=tk.Label(root, text='Bogiatzidakis', bg='red')
        l2.pack()
root = tk.Tk()
MyApp(root)
root.mainloop()

```

Παράμετροι της μηχανής γεωμετρίας pack()

- **expand** = True, False (προκαθορισμένη τιμή) για τιμή 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,y ή και τις 2 διαστάσεις.
 - **side** = 'left', 'right', 'bottom', 'top' (προκαθορισμένη τιμή) αποφασίζει ποια πλευρά του υποδοχέα αρχίζει.
- Σημείωση : όλα τα παιδιά ενός υποδοχέα που έχουν ίδια τιμή εξυπηρετούνται με τη σειρά, δηλ αν έχουν left θα πάνε με την σειρά

Μηχανή γεωμετρίας pack()



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
```

Η τιμή 'right' δεν επηρεάζει την θέση των αντικειμένων εξαιτίας της προηγούμενης εντολής side='left'

Δεν υπάρχει side επομένως ακολουθείται η προκαθορισμένη τιμή top, από πάνω προς τα κάτω

Τα αντίστοιχα προγράμματα

```
# Παραδείγματα με την pack

import tkinter as tk
class MyApp():
    def __init__(self, root):
        root.geometry('100x100+200+200')
        tk.Label(root, text='Label', bg='green').pack()
        tk.Label(root, text='Label2', bg='red').pack()
        # case 1
        w1 = tk.Toplevel()
        w1.geometry('100x100+300+200')
        tk.Label(w1, text='Label', bg='green').pack(expand=1, fill='y')
        tk.Label(w1, text='Label2', bg='red').pack(fill='both')
        # case 2
        w2 = tk.Toplevel()
        w2.geometry('100x100+400+200')
        tk.Label(w2, text='Label', bg='green').pack(expand=1)
        tk.Label(w2, text='Label2', bg='red').pack(fill='both')
```

```

# case 3
w3 = tk.Toplevel()
w3.geometry('100x100+500+200')
tk.Label(w3, text='Label', bg='green').pack(fill='both', expand=1, side='left')
tk.Label(w3, text='Label2', bg='red').pack(fill='both', expand=1, side='right')
# case 4
w4 = tk.Toplevel()
w4.geometry('100x100+600+200')
tk.Label(w4, text='Label', bg='green').pack(fill = 'both', expand=1)
tk.Label(w4, text='Label2', bg='red').pack(fill = 'both', expand=1)

root = tk.Tk()
MyApp(root)
root.mainloop()

```

Άσκηση

Να γράψεις 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 αριθμού δημιουργίας τους.

Πώς μπορούμε να εξασφαλίσουμε ότι δεν θα υπάρχει επικάλυψη;

Οι διαστάσεις της οθόνης δίνονται από :

`root.winfo_screenheight()`, `root.winfo_screenwidth()`

```

# Παραδείγματα μεθόδων winfo_screenheight / winfo_screenwidth /
import tkinter as tk
w = tk.Tk()
w.geometry('+200+200')
tk.Label(w, text=w.winfo_screenheight()).pack()
tk.Label(w, text=w.winfo_screenwidth()).pack()

# δημιουργία νέου παραθύρου
w1=tk.Toplevel(w)

w.mainloop()

```

```

# Πολλά παράθυρα με την Toplevel

import tkinter as tk
import random
class MyApp():
    def __init__(self, root):
        big_font = 'Arial 80'
        screen_x = root.winfo_screenwidth()
        screen_y = root.winfo_screenheight()
        # random colors
        self.r = lambda: random.randint(0, 255) # τυχαίος αριθμός από 0..255
        root.geometry('200x200+100+100')
        l = tk.Label(root, text='0', bg='black', fg='white', font=big_font)
        l.pack(expand=True, fill = 'both')
        for i in range(5):
            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))
            l = tk.Label(w, text=str(i+1), font=big_font, bg = self.random_colour())
            l.pack(expand=True, fill = 'both')
        def random_colour(self):
            return '#{0:02X}{1:02X}{2:02X}'.format(self.r(), self.r(), self.r())
root = tk.Tk() # βασικό παράθυρο
MyApp(root)
root.mainloop()

```

Οι κύριες κλάσεις γραφικών αντικειμένων (widget) της tkinter

Label : απλή περιοχή για μήνυμα κειμένου

Button: πλήκτρο

Message: ουσιαστικά είναι 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 (on/off) για επιλογή ενός μεταξύ πολλών στοιχείων

Scale: κυλιόμενη μπάρα επιλογής

PhotoImage: αντικείμενο για παρουσίαση εικόνων τύπου .gif .ppm .pgm

BitmapImage: αντικείμενο για παρουσίαση εικόνων bitmap .xbm

Αντικείμενο Button

```
# Button
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.myButton = tk.Button(self.root, text="Exit", font = "Arial 30", command = self.buttonPushed)
        self.myButton.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 (handler) : Όταν πατηθεί το πλήκτρο myButton καλείται η συνάρτηση buttonPushed, η οποία ονομάζεται event handler, χειριστής του γεγονότος «πάτημα του πλήκτρου»

Αντικείμενο Entry

```
# get entry
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 button
        text = self.entry.get() # πάρε το κείμενο που έχει εισαχθεί στο πλαίσιο κειμ
ένου
        print(text) # τύπωσε το
root = tk.Tk()
myapp = MyApp(root)
root.mainloop()
```


Αντικείμενο StringVar – για την τροποποίηση κειμένου Label

```
myText = tkinter.StringVar()
# το Label σχετίζεται με τη μεταβλητή myText
myLabel = tkinter.Label(root, textvariable=myText)
myText.set('XYZ') # αλλάζει η τιμή του
```

Παράδειγμα χρήσης StringVar : Σε ένα παράθυρο τοποθετώ ένα Label και ένα Button. Όταν πατήσω το Button εμφανίζεται στο Label πόσες φορές έχω πατήσει μέχρι τότε το Button.

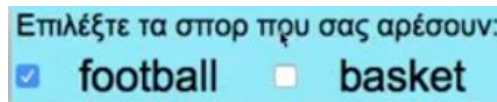
```
# StringVar

import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.r = root
        self.myText = tk.StringVar()
        self.myText.set(30*' ') #για τον καθορισμό του χώρου που θα καταλάβει
        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",
                            bg="yellow", command=self.buttonPressed)
        self.b.pack(fill='both', expand=1)
        self.count = 0
    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)

root = tk.Tk()
myapp = MyApp(root)
root.mainloop()
```

Αντικείμενο Checkbutton

Να φτιαχτεί μια φόρμα όπως η παρακάτω. Κάθε φορά που επιλέγεται κάποιο σπορ να εμφανίζονται όλες οι επιλογές που έχουν γίνει (football / basket)



Έχει επιλεγεί: football

```
# Checkbutton

import tkinter as tk
class MyApp():
    def __init__(self, root):
        self.root = root
        self.color = '#66f0ff'
        self.widgets()

    def widgets(self):
        self.l = tk.Label(self.root, text='Επιλέξτε τα σπορ που σας αρέσουν: ',
                          font="Arial 18", bg=self.color)
        self.l.pack(fill='both', expand =1)
        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='')
        self.check1.pack( side = 'left', fill = 'both', expand = 1)
        self.answer2 = tk.StringVar()
        self.check2 = tk.Checkbutton(self.root, text='basket ', command= self.check,
                                     font="Arial 26",bg=self.color, variable=self.answer2,
                                     onvalue='basket', offvalue='')
        self.check2.pack( side = 'left', fill = 'both', expand = 1)
        self.count = 0

    def check(self):
        print('Έχουν επιλεγεί: ', self.answer1.get(), self.answer2.get())

root = tk.Tk()
myapp = MyApp(root)
root.mainloop()
```

Η μεταβλητή answer1 όταν επιλεγεί το check box check1 παίρνει τιμή 'football' αλλιώς κενό "", αυτή ορίζεται από το onvalue/offvalue του αντικειμένου check1

Αντικείμενο PhotoImage: Εισαγωγή εικόνων

Επιτρέπονται εικόνες μόνο τύπου gif και bitmap. Για εικόνες τύπου jpg, png απαιτείται η χρήση της πρόσθετης βιβλιοθήκης PIL (Python Image Library)

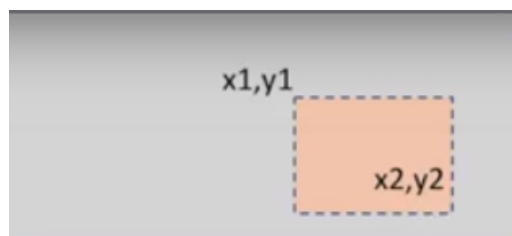
Παράδειγμα : Εισαγωγή εικόνας σε widget τύπου Label

```
# Show image with PhotoImage
import tkinter as tk
class App():
    def __init__(self, root):
        img = tk.PhotoImage(file='python_logo.gif')
        l= tk.Label(root, image = img)
        l.image = img # αυτό είναι απαραίτητο για να κρατήσουμε την αναφορά στο αντικείμενο
        και
        l.pack() # να μην καταστραφεί από τη συλλογή σκουπιδιών!! Πρόκειται για δυσλειτουργία της python
root = tk.Tk()
App(root)
root.mainloop()
```

Πώς μπορώ να πάρω μόνο ένα τμήμα μιας μεγαλύτερης εικόνας; (π.χ. από ένα animation sprite)

```
spritesheet = tk.PhotoImage(file="python_logo.gif")
def subimage(self, 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 ορθογωνίου τμήματος της αρχικής εικόνας



```

# PhotoImage
import tkinter as tk
import random

class App():
    def __init__(self, root):
        self.root = root
        blue = '#3e719a'
        yellow = '#fddb6a'
        self.root.config(bg=random.choice([blue,yellow]))
        self.image1 = tk.PhotoImage(file="python_logo.gif")
        self.x = self.image1.width()
        self.y = self.image1.height()
        print(self.x, self.y)
        puzzle = []
        for i in range(5):
            for j in range(2):
                x1,y1 = int(i*self.x/5), int(j*self.y/2)
                x2,y2 = x1+int(self.x/5), y1+int(self.y/2)
                puzzle.append(self.subimage(x1,y1,x2,y2, self.image1))
        random.shuffle(puzzle) # ανακατεύω τα στοιχεία της λίστας
        for n,im in enumerate(puzzle): # δείνει ζευγάρια, n=0,1,2 im το περιεχόμενο
στη θέση αυτή
            if not n%5 :
                f = tk.Frame(self.root)
                f.pack(expand=True, fill='both')
                l = tk.Label(f, image=im, bg =random.choice([blue,yellow]))
                l.pack(side='left', padx=2, pady=2) # αφήνω 2 pixels περιθώριο
                l.image = im # αναφορά στην αντίστοιχη εικόνα για να μην σβηστεί από τη
συλλογή σκουπιδιών

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

root = tk.Tk()
App(root)
root.mainloop()

```

Επανάληψη : Πώς τροποποιούμε τις ιδιότητες ενός γραφικού αντικειμένου;

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

```

widget = tk.Label(self.root)
widget['text'] = 'Καλημέρα' # όταν ξέρω ποια ιδιότητα θα τροποποιήσω

```

B) Με κλήση της μεθόδου config()

```

widget = tk.Label(self.root)
widget.config(text= 'Καλημέρα')

```

Υποδοχείς Frame / LabelFrame

Υποδοχείς γραφικών αντικειμένων

Tk (βασικό παράθυρο)

Toplevel (κλάση που παράγει «παιδιά» του βασικού παραθύρου, δηλ άλλα παράθυρα

Frame (ιεραρχικός διαχωρισμός περιοχών ενός παραθύρου), είναι ένα γραφικό αντικείμενο που μπορεί να περιέχει άλλα γραφικά αντικείμενα

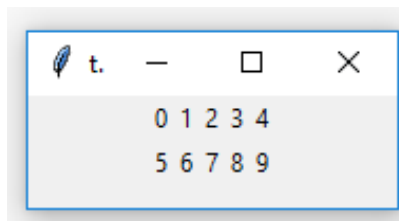
LabelFrame (ένα Frame με πλαίσιο και τίτλο)

ttk.Notebook (καρτέλες-tabs, όπως στους browsers)

Διαχωρισμός χώρου παραθύρου root

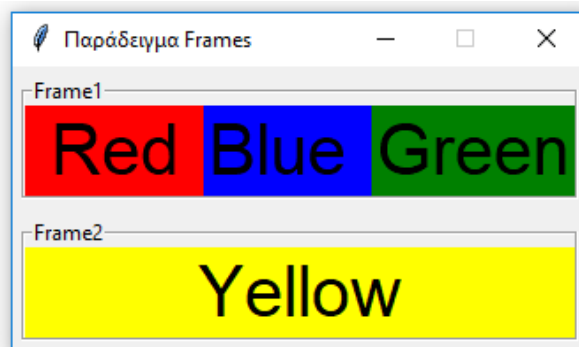


1° Παράδειγμα υποδοχέα Frame



```
import tkinter as tk
root=tk.Tk()
for i in range(10):
    if not i%5:
        f1=tk.Frame(root)
        f1.pack()
        tk.Label(f1, text=str(i)).pack(side='left')
    else:
        tk.Label(f1, text=str(i)).pack(side='left')
root.mainloop()
```

2^ο Παράδειγμα υποδοχέα Frame



```
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, padx=5,pady=5)
        # padx=5, pady=5 από τον υποδοχέα root (μπαμπάς) αφήνω περιθώριο 5px
        red = tk.Label(self.f1, text=' Red ', font=myfont, bg="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)
root.mainloop()
```

Υποδοχέας Notebook

Για δημιουργία καρτελών. Ανήκει στην υποκλάση ttk (σημαίνει themed ttk)

Πριν την χρησιμοποιήσω θα πρέπει να εισάγω την υποκλάση : from tkinter import ttk

Παράδειγμα :

```
from tkinter import ttk
import tkinter as tk
chapters=['1.Chapter','2.Chapter'] # κείμενο των tabs
texts = ["the text of 1st chapter", "the text of 2nd chapter"] # περιεχόμενο
root=tk.Tk()
root.geometry('300x300')
nb = ttk.Notebook(root)
for i in range(len(chapters)):
    l = tk.Label(nb, text=texts[i])
    l.pack(expand=1, fill='both')
    nb.add(l, text=chapters[i]) # προσθήκη tab και περιεχομένου

nb.pack(expand=1, fill='both')
root.mainloop()
```

Άσκηση : Να δημιουργηθούν 4 καρτέλες (tabs) με τυχαία χρώματα, κάθε καρτέλα θα περιέχει ένα label.

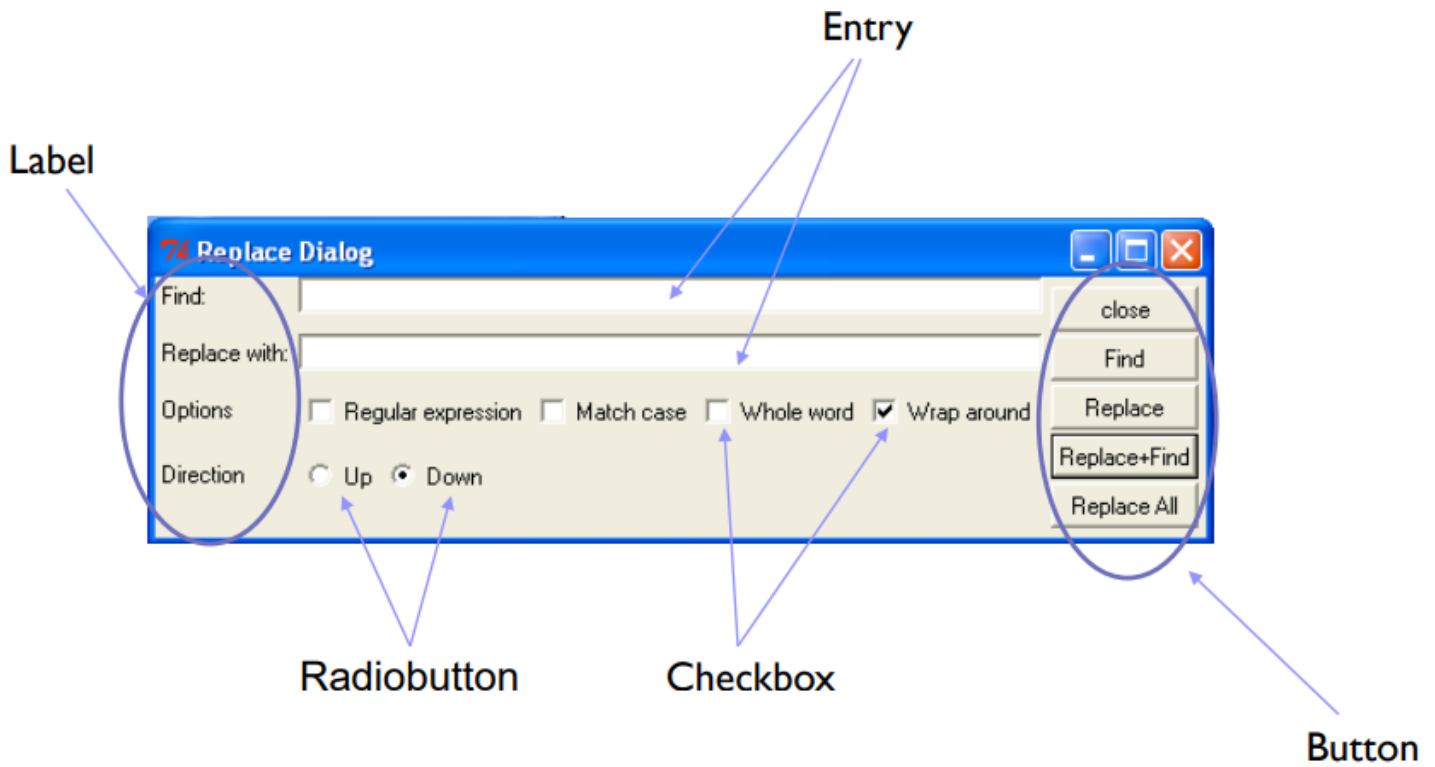
```
# Notebook

import tkinter as tk
from tkinter import ttk
import random

class MyNotebook():
    def __init__(self, note, name, color):
        self.note = note
        self.f = tk.Frame(self.note) # φτιάχνω υποδοχέα Frame
        self.f.pack(expand=True, fill='both', padx=1, pady=1)
        self.l = tk.Label(self.f, text='σημειωματάριο', bg=color, font='Arial 24')
        self.l.pack(expand=True, fill='both', padx=1, pady=1)
        # το frame που δημιουργήσα το επισυνάπτω ως καρτέλα στο note, δημιουργώ δηλ.
        # μια νέα καρτέλα
        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', 'Τέταρτο']:
            mynotebooks.append(MyNotebook(self.note, name, '#{ :02X}{ :02X}{ :02X}'.format
            (r(),r(),r())))

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



```
import tkinter as tk
from tkinter.colorchooser import askcolor

def callback():
    result = askcolor(color="#6A9662",
                      title = "Iasonas's Colour Chooser")
    print(result)

root = tk.Tk()
tk.Button(root,
          text='Choose Color',
          fg="darkgreen",
          command=callback).pack(side=tk.LEFT, padx=10)
tk.Button(text='Quit',
          command=root.quit,
          fg="red").pack(side=tk.LEFT, padx=10)
tk.mainloop()
```


Γεωμετρία GRID

Εναλλακτικός τρόπος διάταξης widgets. Αντί για την μέθοδο pack() μπορεί να χρησιμοποιηθεί η grid(). Π.χ. : w.grid(row=0, column=0, rowspan=3), ορίζεται ο χώρος του παραθύρου ή του Frame με λογική στηλών και γραμμών όπως ένας πίνακας.

Δεν επιτρέπεται να χρησιμοποιήσουμε τις μεθόδους grid και pack μαζί.

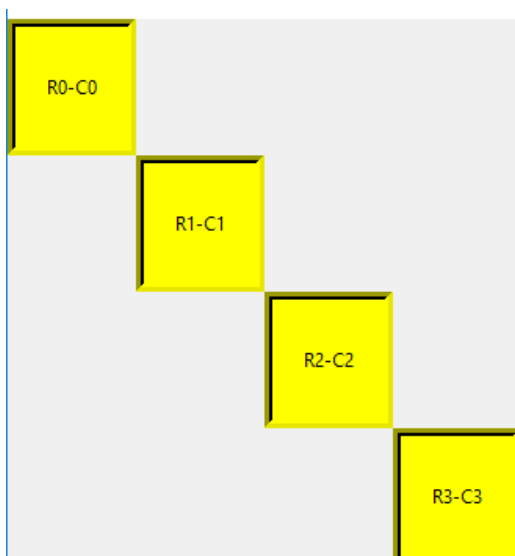
1^ο παράδειγμα: διάταξη widgets με τον διαχειριστή γεωμετρίας grid. Να διαταχθούν 16 label σε πίνακα όπως το παρακάτω σχήμα.

R0-C0	R0-C1	R0-C2	R0-C3
R1-C0	R1-C1	R1-C2	R1-C3
R2-C0	R2-C1	R2-C2	R2-C3
R3-C0	R3-C1	R3-C2	R3-C3

```
# grid παράδειγμα 1 διάταξη 16 αντικειμένων label σε πίνακα 4x4

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") # δημιουργείται το αντικείμενο
                lab.grid(row=r, column=c) # και τοποθετείται στην θέση του
root = tk.Tk()
App(root)
root.mainloop()
```

2^ο παράδειγμα :Τροποποιήστε τον παραπάνω κώδικα ώστε να διαταχθούν 4 label σε πίνακα μόνο στη διαγώνιο , πειραματιστείτε με τις παραμέτρους για το πλαίσιο `borderwidth` και `relief`



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

3^ο παράδειγμα : Δημιουργήστε το παρακάτω :

0	1	2	
0	1	2	5

```

# grid παράδειγμα 3

import tkinter as tk
w = tk.Tk()
f = 'Consolas 30'
for i in range(3):
    l=tk.Label(text=str(i), font=f, width=10, height=5, borderwidth=1, relief='sunken')
    l.grid(row=0,column=i)
for i in range(3):
    l=tk.Label(text=str(i), font=f, width=10, height=5, borderwidth=1, relief='sunken')
    l.grid(row=1,column=i)
l=tk.Label(text='5', bg='yellow', font=f, width=10, height=5, borderwidth=1, relief='sunken')
l.grid(row=0,column=5, rowspan=2)
w.mainloop()

```

Κλάσεις σύνθετων γραφικών αντικειμένων

Menubutton : Πλήκτρο που ενεργοποιεί ένα αντικείμενο 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 (συνήθως λέξεις)

Scrollbar: Αντικείμενο που επιτρέπει την κύλιση άλλων widgets(π.χ. listbox, canvas, text)

Text: Σύνθετο αντικείμενο για επεξεργασία και παρουσίαση κειμένου πολλών γραμμών, μπορείς π.χ. να φτιάξεις έναν στοιχειώδη editor

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

Tk toolkit, ttk (themed tk)

```
# Style

import tkinter as tk
from tkinter import ttk

class MyApp:
    def __init__(self, root):
        #χρήση της κλάσης Style. Όσα buttons έχουν style btn έχουν bg='yellow' και font='Arial 30'
        # δεν χρειάζεται με αυτόν τον τρόπο να περνάω ως παραμέτρους τα bg και font
        s = ttk.Style().configure('btn.TButton', bg='yellow', font='Arial 30')
        root.title('Παράδειγμα 6')
        self.root = root
        self.button = ttk.Button(self.root, style='btn.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) #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 πλαίσιο κειμένου
            print(text) # τύπωσέ το

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

ttk widgets

Η ttk (βασική είναι μια υποβιβλιοθήκη της tkinter) περιέχει 17 widgets, τα 11 υπήρχαν ήδη στην tkinter :

Button, Checkbutton, Entry, Frame, Label, LabelFrame, Menubutton, PanedWidnow, Radiobutton, Scale και Scrollbar.

Υπάρχουν 6 νέα widgets

Combobox, Notebook, Progressbar, Separator, Sizegrip και Treeview. Όλα είναι υποκλάσεις του Widget

Δημιουργία Μενού

Ένα μενού είναι widget με ιεραρχική δομή πολλών επιπέδων (υπό-μενού)

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

```
# Menu

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) # χρειάζεται σύνδεση με το menu γιατί όταν έφτιαχνα το mb δεν είχε
ακόμη οριστεί το m. Βασικά με την config τροποποιώ εκ των υστέρων το mb
w.mainloop()
```

2. Μενού συσχετιζόμενο με παράθυρο

```
# add_cascade

import tkinter as tk
import tkinter.messagebox as ms # get standard dialogs

class App():
    def __init__(self, root):
        # Tk8.0 style top-level window menus
        top = tk.Menu(root) # σύνδεση του μενού top με το παράθυρο root
        root.config(menu=top) # επίσης σύνδεσε το παράθυρο με το μενού

        file = tk.Menu(top) # file : δημιούργησε ένα νέο αντικείμενο τύπου Menu
        top.add_cascade(label='File', menu=file, underline=0)# σύνδεση του μενού file
με το top
        file.add_command(label='New...', command=self.notdone, underline=0)
        file.add_command(label='Open...', command=self.notdone, underline=0)
        file.add_command(label='Quit', command=root.quit, underline=0)

        edit = tk.Menu(top, tearoff=False)
        top.add_cascade(label='Edit', menu=edit, underline=0)
        edit.add_command(label='Cut', command=self.notdone, underline=0)
        edit.add_command(label='Paste', command=self.notdone, underline=0)
        edit.add_separator()
```

```

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)

def notdone(self):
    ms.showerror('Not implemented', 'Not yet available')

if __name__ == "__main__":
    root = tk.Tk()
    App(root)
    root.mainloop()

```

Επιλογές μενού

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

`m.add_command(option, ...)`

πρόσθεσε επιλογή στο μενού με χρήση ορισμάτων `label`, `bitmap`, ώστε να τοποθετήσεις κείμενο, εικόνες στο μενού - επιλογή `command` για να συνδέσεις την επιλογή με συνάρτηση που θα εκτελεστεί.

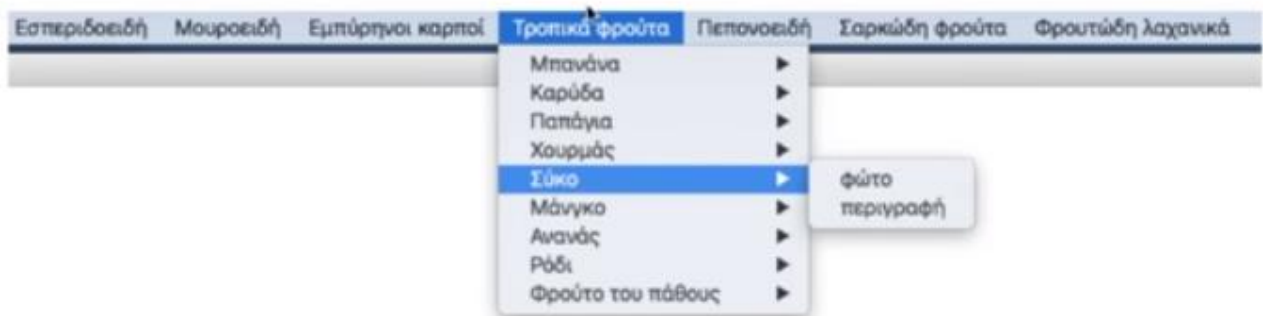
`m.add_cascade(option, ...)` πρόσθεσε ένα μενού επόμενου επιπέδου

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

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

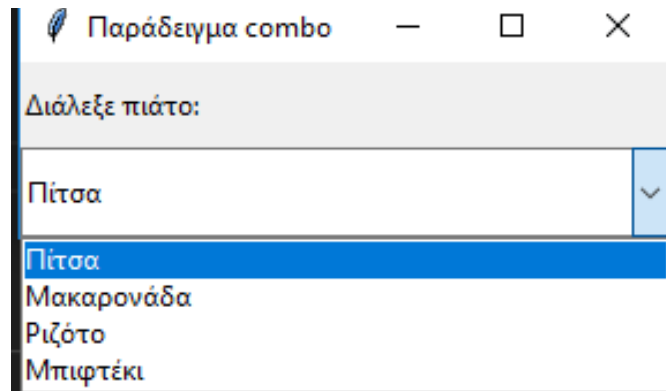
`m.add_separator()` πρόσθεσε μια διαχωριστική γραμμή

Άσκηση : δημιουργήστε το παρακάτω μενού



Combobox

Σύνθετο widget που ανήκει στην υπο-βιβλιοθήκη ttk. Επιτρέπει την επιλογή από μια λίστα τιμών. Είναι ουσιαστικά entry box με πολλές επιλογές από τον χρήστη.



```
import tkinter as tk
from tkinter import ttk
# Παράδειγμα combobox

class MyApp(tk.Frame):
    def __init__(self, root):
        self.root = root
        root.title('Παράδειγμα combo')
        self.label = tk.ttk.Label(self.root, text='Διάλεξε πιάτο:')
        self.label.pack(expand=1, fill='both')
        self.combo()
    def combo(self):
        self.box_value = tk.StringVar() # μεταβλητή που θα αποθηκεύεται η επιλογή του χρήστη
        self.box = ttk.Combobox(self.root, textvariable=self.box_value, state = 'readonly',
                                values = ('Πίτσα', 'Μακαρονάδα', 'Ριζότο', "Μπιφτέκι"))
        self.box.bind("<<ComboboxSelected>>", self.newselection) #γεγονός επιλογής του combo
        box τότε καλείται η newselection
        self.box.current(0) # ποια θα είναι αυτή που θα βλέπει ο χρήστης, συνήθως βάζω την π
        το δημοφιλή
        self.box.pack(expand=1, fill='both')
    def newselection(self, event):
        self.value_of_combo = self.box.get() #"παίρνει" την επιλογή που έκανε ο χρήστης
        print(self.value_of_combo)

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

TreeView

Σύνθετο widget που ανήκει στην υποβιβλιοθήκη 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")
```

stretch: αν θα αλλάζει πλάτος
anchor='e' : στοίχιση αριστερά (east)
heading : επικεφαλίδα στήλης

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"))
```

Tree.insert:

"" : ανήκει στο πρώτο επίπεδο
0 ή end : θέση μέσα στο μενού (πρώτο

Υπάρχουν 2 τρόποι να ορίσεις id : δείτε το id2 και το id5

	column A	column B
Line 1	1A	1B
Dir 2		
sub dir 2	2A	2B
Dir 3		
sub dir 4	4A	4B
Dir 5		
sub dir 6	6A	6B

```

# TreeView
import tkinter as tk
from tkinter import ttk

class App():
    def __init__(self, root):
        tree = ttk.Treeview(root)
        tree["columns"] = ("one", "two")
        tree.column("one", width=100)
        tree.column("two", width=100)
        tree.heading("one", text="column A")
        tree.heading("two", text="column B")
        # εισαγωγή εγγραφών
        tree.insert("", 0, text="Line 1", values=("1A", "1B")) # εισαγωγή πρώτου επιπέδου

        id2 = tree.insert("", "end", text="Dir 2")
        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")
        tree.insert("dir5", 3, text=" sub dir 6", values=("6A", " 6B"))

        tree.pack()

root = tk.Tk()
App(root)
root.mainloop()

```

Dialog box

Ένα παράθυρο διαλόγου dialog box είναι ένα παράθυρο που θέτει ένα ερώτημα στον χρήστη, δείχνει ένα μήνυμα σφάλματος, ζητάει ια 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 :

```
from tkinter import messagebox
για απλά μηνύματα πληροφορόρησης ή ερώτησης ναι/όχι

from tkinter import simpledialog
Ζητάει έναν αριθμό ή κείμενο (π.χ. ζητάω το όνομα ενός παίκτη, για αριθμούς μπορώ να
ορίζω και το διάστημα τιμών που θα πρέπει να δοθούν)

from tkinter import filedialog, colorchooser
```

Άλλα παράθυρα διαλόγου :

askyesno, askokcancel, askretrycancel, askquestion

```
from tkinter import messagebox
```

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

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

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

Διάλογος εισαγωγής τιμής από το χρήστη : askstring, askinteger

```
from tkinter import simpledialog
```

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

```
ans = simpledialog.integer("Dialog Title","Δώστε έναν ακέραιο:")
```

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

```
>>> from tkinter import messagebox
>>> messagebox.showinfo('είσαι στο βόρειο ημισφαίριο')
'ok'
>>> messagebox.showinfo('είσαι στο βόρειο ημισφαίριο')
```

```
>>> from tkinter import messagebox
>>> messagebox.askyesno('Κρίσιμη ερώτηση:', 'Να προχωρήσω;')
```

```
>>> import tkinter as tk
>>> w = tk.Tk()
>>> ans = simpldialog.askinteger("Num", "τιμή μεταξύ 0 και 100", minvalue=0, maxvalue=100)
```

```
# Dialogues
```

```
import tkinter as tk
from tkinter import messagebox, colorchooser, filedialog, simpldialog
class App():
    def __init__(self, root):
        self.root = root
        self.root.geometry('400x400+200+200')
        self.widgets()
    def widgets(self):
        self.current_color = "#6A9662"
        self.f = tk.Frame(self.root)
        self.f.pack(expand=True, fill = 'both')
        self.b1 = tk.Button(text='πληροφορία', font='Arial 48')
        self.b1.pack(side= 'top', expand=True, fill = 'both')
        self.b1.bind('<Button-1>', self.showinfo)
        self.b2 = tk.Button(text='άνοιγμα αρχείου', font='Arial 48')
        self.b2.pack(side='top', expand=True, fill='both')
        self.b2.bind('<Button-1>', self.file)
        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)
        self.b4 = tk.Button(text='δώσε τιμές', font='Arial 48')
        self.b4.pack(side='top', expand=True, fill='both')
        self.b4.bind('<Button-1>', self.ask_value)
    def ask_value(self, event):
        ans = simpldialog.askstring("Title", "Το όνομά σας:")
        print(ans)
        ans = simpldialog.askinteger("Dialog Title", "Δώστε ένα ακέραιο")
        print(ans)
        ans = simpldialog.askinteger("Num", "τιμή μεταξύ 0 και 100",
                                     minvalue=0, maxvalue=100)
        print(ans)
    def color(self, event):
        print('color', event)
        result = colorchooser.askcolor(color=self.current_color, title="Colour Chooser")
        print(result)
        self.current_color = result[-1]
    def file(self, event):
```

```

name = filedialog.askopenfilename()
print(name)
def showinfo(self, event):
    messagebox.showinfo('info',
                        'πάτησες στη θέση {},{}'.format(event.x, event.y))
root = tk.Tk()
App(root)
root.mainloop()

```

Διαχείριση γεγονότων – μέθοδος bind - 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

ΤΥΠΙΚΑ ΓΕΓΝΟΤΑ

bind(event, χειριστής)

<Button-1> ή <1>	πάτημα αριστερού πλήκτρου ποντικιού <B1>
<ButtonRelease-1>	ελευθέρωση αριστερού πλήκτρου
<Button-3> ή <3>	δεξί πλήκτρο ποντικιού/ προσοχή! <2> σε mac
<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"
<B1-Motion>	μετακίνηση του ποντικιού με πατημένο το Button-1
<FocusIn> <FocusOut>	Έστιαση / απώλεια εστίασης
<Up><Down><Left><Right>	βέλη κίνησης
<KeyRelease><KeyPress>	Ελευθέρωση-πάτημα πλήκτρου

```

# Events παράδειγμα 1

import tkinter as tk
class MyApp ():
    def __init__(self, root):
        self.root = root
        root.title("Παράδειγμα 1: γεγονότα")
        self.create_widgets()
    def create_widgets(self):
        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))

```

```

def main():
    root = tk.Tk()
    app = MyApp(root)
    root.mainloop()
if __name__ == '__main__': main()

```

Events παράδειγμα 2

```

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.l = tk.Label(self.root, text='', font = 'Arial 40')
        self.l.pack(expand=True, fill='both')
        self.l.bind("<Enter>", lambda e:self.l.config(text='Έχει μπει'))
        self.l.bind("<Leave>", lambda e: self.l.config(text='Έχει βγει'))
def main():
    root = tk.Tk()
    app = MyApp(root)
    root.mainloop()
if __name__ == '__main__': main()

```

Events παράδειγμα 3

```

import tkinter as tk
class MyApp ():
    def __init__(self, root):
        self.root = root
        root.title("Example 7: detecting events")
        root.geometry("400x300+300+300")
        self.create_widgets()
    def create_widgets(self):
        self.root.bind("<Key>", self.handler)
        self.root.bind("<Button-1>", self.focus)
        self.myText = tk.StringVar()
        self.mylabel = tk.Label(self.root, textvariable=self.myText,
                                font="Arial 30",bg="yellow")
        self.mylabel.pack(fill='both', expand=1)

```

```
def handler(self, event):
    print("πατήθηκε: "+repr(event.char))
    self.myText.set( "πατήθηκε: "+str(event.char)+
                    '\nkωδικός:{}'.format(ord(event.char)))

def focus(self, event):
    self.root.focus_set()
    print ("clicked at", event.x, event.y)
    self.myText.set( "κλικ σε: " + str(event.x) + "," + str(event.y))

def main():
    root = tk.Tk()
    app = MyApp(root)
    root.mainloop()

main()
```