

Python – Εισαγωγή στο Tkinter

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών
Γιάννης Γαροφαλάκης - Σπύρος Σιούτας

Γ. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας



Python - Tkinter

- Παρουσίαση του Tkinter
- Οι βασικές έννοιες λειτουργίας του Tkinter
- Τεχνικές προγραμματισμού σχετικές με τη δημιουργία GUI



GUI - Χαρακτηριστικά

- Windows
- Icons
- Menus
- Pointing
- Graphics



GUI frameworks in JAVA

- AWT – Native widgets, only basic components, dated
- Swing – Java rendering, rich components
- SWT + JFace – Mixture of native widgets and Java rendering; created for Eclipse for faster performance

Others – Apache Pivot, SwingX, JavaFX, ...



Python - Tkinter

- Το μοναδικό GUI embedded από την αρχή στη Python
- Βασισμένο στο Tcl / open source scripting language / GUI widget set που αναπτύχθηκε από τον John Ousterhout (90s)
- Το Tk χρησιμοποιείται σε μεγάλη ποικιλία άλλων γλωσσών (Perl, Ruby, PHP κ.λπ.)
- Cross-platform (Unix / Windows / MacOS)
- Είναι μικρό (έχει περίπου 25 βασικά widgets)

Tkinter - Παράδειγμα

Κώδικας

```
>>> from Tkinter import Label  
>>> x = Label(None, text="Hello World")  
>>> x.pack()  
>>> x.mainloop()
```

Αποτέλεσμα



Tkinter – Παράδειγμα Button

Κώδικας

```
>>> def response():  
...     print "You did it!"  
...  
  
>>> from Tkinter import Button  
  
>>> x = Button(None, text="Do it!", command=response)  
  
>>> x.pack()  
>>> x.mainloop()
```

Αποτέλεσμα



You did it!



Python – Tkinter – Τα βασικά 1/2

- Δημιουργία και διαμόρφωση widget (ετικέτες, κουμπιά, sliders κ.λπ.)
- Σχεδιασμός της τοποθέτησης τους (γεωμετρία)
- Δημιουργία συναρτήσεων που ανταποκρίνονται σε διάφορα συμβάντα GUI (event handling)
- Εκτέλεση ενός event loop

Python – Tkinter – Τα βασικά 2/2

- Για τη δημιουργία ενός GUI χρειάζεται τουλάχιστον ένα αρχικό graphical window
- Το αρχικό graphical window είναι γνωστό ως "root"
- Συνήθως χρησιμοποιείται ένα root graphical window ανά εφαρμογή



Graphical window
(χωρίς widgets)

Tkinter – Root Window

Για να δημιουργήσουμε ένα root window:

```
>>> from Tkinter import *  
>>> root = Tk(className="ApplicationName")  
>>>
```

- Για να ξεκινήσουμε την εκτέλεση του GUI, αρχικοποιούμε το loop

```
>>> root.mainloop()
```

- και εμφανίζεται ένα κενό graphical window (root)



Tkinter – Ρυθμίσεις στα Widgets

- Τα Widgets μπορούν να διαμορφωθούν αρχικά ανάλογα με τις ανάγκες μας

```
>>> b = Button(root, text="A Button", bg="blue", fg="white")
```



- Τα Widgets μπορούν να διαμορφωθούν και στη συνέχεια με άλλες παραμέτρους

```
>>> b.config(bg="red") # Change background
```

- Με την εντολή `cget()` εμφανίζονται οι ρυθμίσεις του widget για την παράμετρο που εισάγουμε

```
>>> b.cget("bg")
```

```
'red'
```

```
>>>
```

Tkinter – Events στα Widgets

Τα περισσότερα widgets ανταποκρίνονται σε διάφορα events

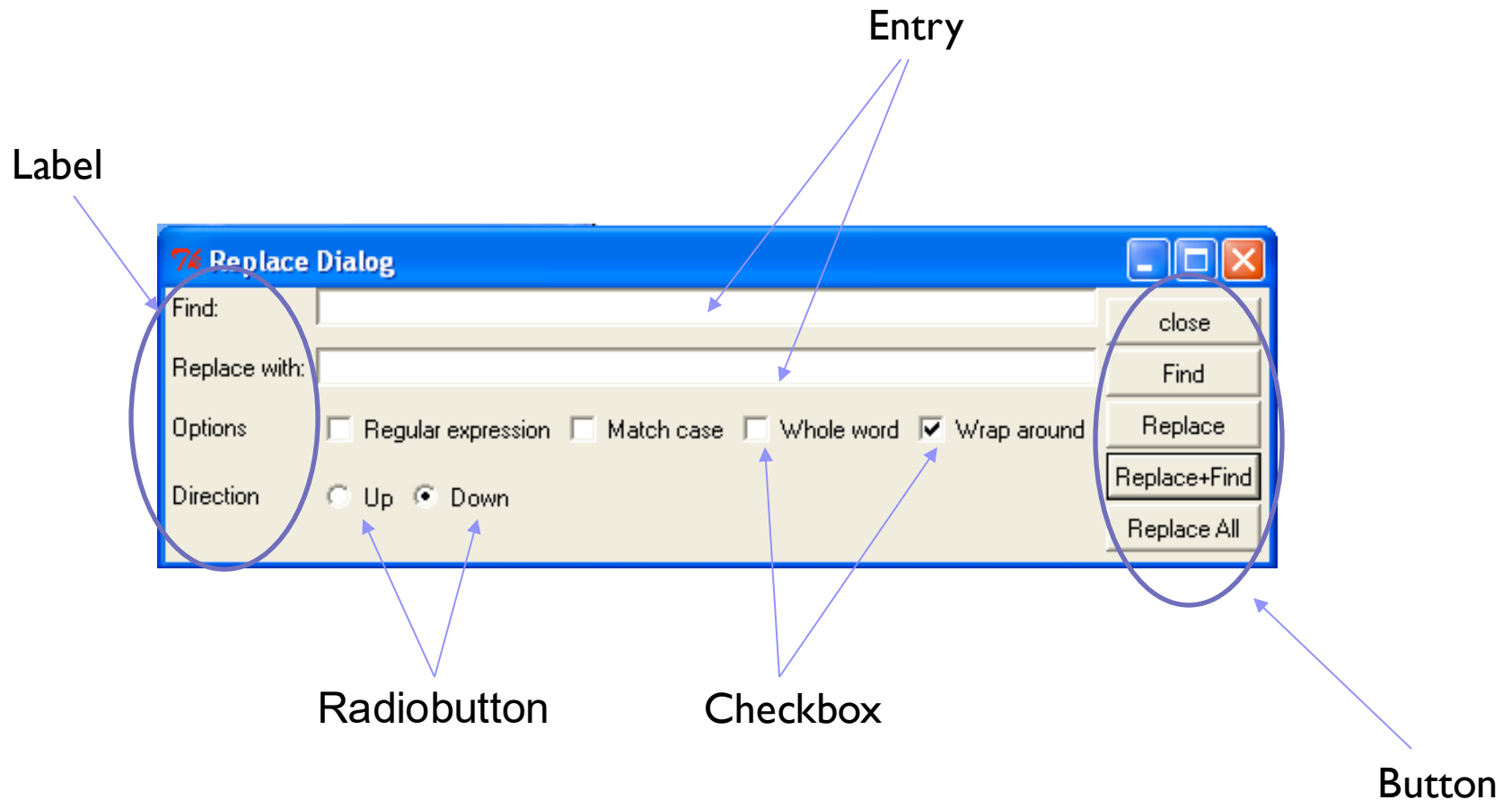
```
>>> def pressed():  
...   print "You pressed it!"  
  
...  
>>> b = Button(root, text="A Button", command=pressed)
```

Event handler



Οι τύποι των event και το protocol handler εξαρτώνται από το widget (π.χ., διαφορετικά για κουμπιά από για scrollbars)

Tkinter – Δημιουργία blocks με Widgets



Tkinter – Widgets: Labels

Δημιουργία ετικετών

```
>>> w = Label(root,text="A label")
```

Συνήθως χρησιμοποιείται για μικρές ετικέτες κειμένου

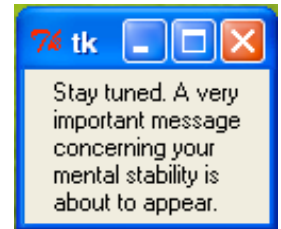


Tkinter – Widgets: Messages

Δημιουργία μηνυμάτων

```
>>> w = Message(root, text="Stay tuned. A very  
important message concerning your mental  
stability is about to appear")
```

Χρησιμοποιείται για ενημερωτικά μηνύματα



Tkinter – Widgets: Buttons

Δημιουργία κουμπιών

```
>>> def when_pressed():  
...     print "Do something"  
...  
>>> w = Button(root,text="Press Me!",command=when_pressed)
```



Tkinter – Widgets: Checkbutton

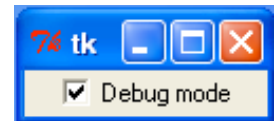
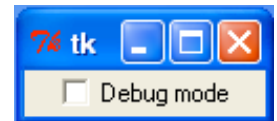
Δημιουργία Checkbuttons

```
>>> debug_mode = IntVar(value=0)
>>> w = Checkbutton(root, text="Debug mode",
...                 variable=debug_mode)
...
>>>
```

```
>>> debug_mode.get()
```

```
1
```

```
>>>
```



Tkinter – Widgets: Radiobutton

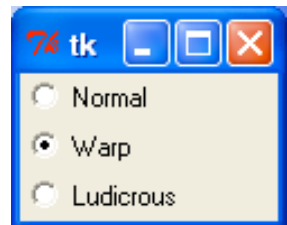
Δημιουργία Radiobuttons

```
>>> speed=StringVar()
>>> r1 = Radiobutton(root,text="Normal",variable=speed,
...                 value="normal")
>>> r2 = Radiobutton(root,text="Warp",variable=speed,
...                 value="warp")
>>> r3 = Radiobutton(root,text="Ludicrous",variable=speed,
...                 value="ludicrous")
```

```
>>> speed.get()
```

```
'warp'
```

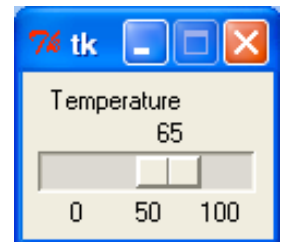
```
>>>
```



Tkinter – Widgets: Scales/Sliders

Δημιουργία Scales/Sliders

```
>>> temp = IntVar()
>>> def on_move(value):
...     print "moved", value
...
>>> w = Scale(root,label="Temperature",variable=temp,
...           from_=0,to=100,tickinterval=50,
...           orient='horizontal',command=on_move)
...
>>>
```



Tkinter – Widgets: Εισαγωγή text

Παράθυρο για εισαγωγή κειμένου

```
>>> value = StringVar(root)
>>> w = Entry(root, textvariable=value)
```

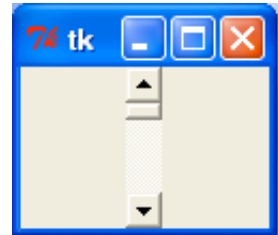
```
>>> value.get()
'This is a test'
>>>
```



Tkinter – Widgets: Scrollbar

Δημιουργία Scrollbar

```
>>> w = Scrollbar(root,orient="vertical")
```

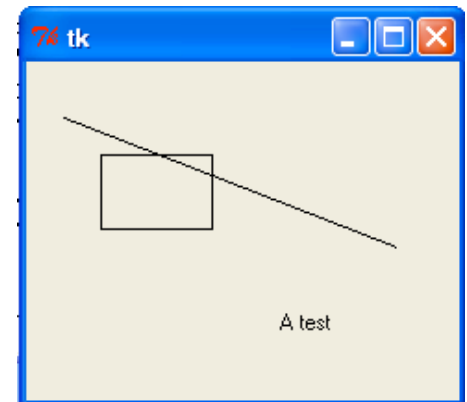


Tkinter – Widgets: Canvas

Δημιουργία Canvas

```
>>> w = Canvas(root,width=250,height=250)
>>> w.create_line(20,30,200,100)
>>> w.create_rectangle(40,50,100,90)
>>> w.create_text(150,140,text="A test")
>>>
```

(x0 , y0 , x1 , y1)



Tkinter – Widgets: Μενού

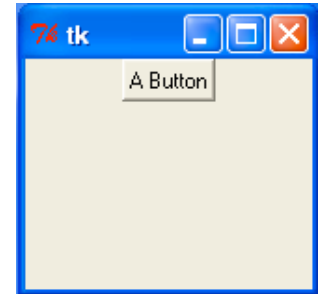
Δημιουργία μενού

```
>>> top = Menu(root)
>>> file = Menu(top)
>>> file.add_command(label='Open',command=open_cmd)
>>> file.add_command(label='Close',command=close_cmd)
>>> top.add_cascade(label="File",menu=file)
>>> edit = Menu(top)
>>> edit.add_command(label="Cut",command=cut_cmd)
>>> edit.add_command(label="Paste",command=paste_cmd)
>>> top.add_cascade(label="Edit",menu=edit)
>>> root.config(menu=top)
>>>
```



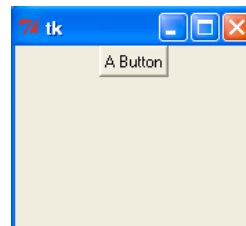
Tkinter – Widgets: Packing

- τα widgets πρέπει να τοποθετηθούν μέσα στο παράθυρο(geometry) κάνοντας χρήση της μεθόδου `pack()`
- Από προεπιλογή η μέθοδος τοποθετεί ένα widget κεντραρισμένο στο πάνω μέρος του παράθυρου.



Choosing Sides

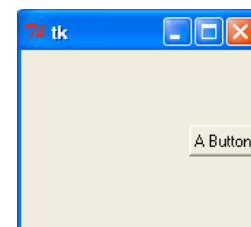
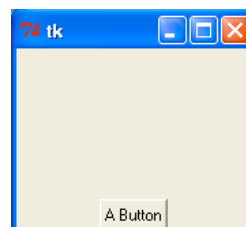
`w.pack(side=TOP)`



`w.pack(side=LEFT)`



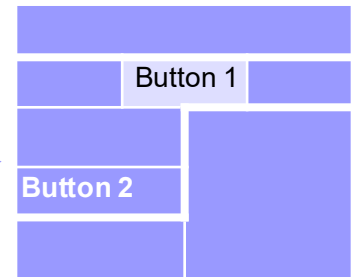
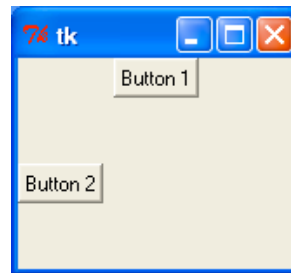
`w.pack(side=BOTTOM)` `w.pack(side=RIGHT)`



Tkinter – Πολλάπλά widgets

More than one widget can be packed

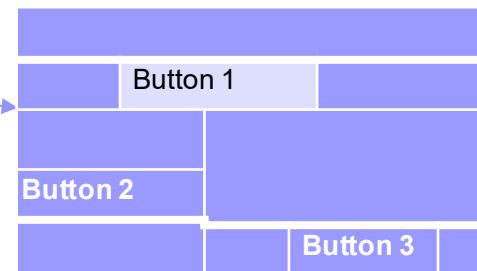
```
>>> root = Tk()
>>> b1 = Button(root, text="Button 1")
>>> b2 = Button(root, text="Button 2")
>>> b1.pack(side=TOP)
>>> b2.pack(side=LEFT)
>>> root.mainloop()
```



Tkinter – Προσθήκη τρίτου κουμπιού

Προσθήκη τρίτου κουμπιού

```
>>> root = Tk()
>>> b1 = Button(root,text="Button 1")
>>> b2 = Button(root,text="Button 2")
>>> b3 = Button(root,text="Button 3")
>>> b1.pack(side=TOP)
>>> b2.pack(side=LEFT)
>>> b3.pack(side=BOTTOM)
>>> root.mainloop()
```



Tkinter – Filling / Expanding

- Filling: Το Widget επεκτείνεται για να χρησιμοποιήσει όλο το χώρο που του έχει εκχωρηθεί
- Expanding: Το Widget επεκτείνεται ώστε να χρησιμοποιεί όλο τον εκχωρημένο χώρο και τον υπόλοιπο ελεύθερο χώρο

```
w.pack(side=SIDE, fill=X)
```

```
w.pack(side=SIDE, fill=Y)
```

```
w.pack(side=SIDE, fill=BOTH)
```

```
w.pack(side=SIDE, fill=FILL, expand=True)
```

Tkinter – Filling

Έχουμε δυο widgets:

```
>>> Button(root, text="tiny").pack()  
>>> Button(root, text="humongous").pack()  
>>>
```



Τα ίδια widgets με χρήση filling

```
>>> Button(root, text="tiny").pack(fill=X)  
>>> Button(root, text="humongous").pack(fill=X)  
>>>
```

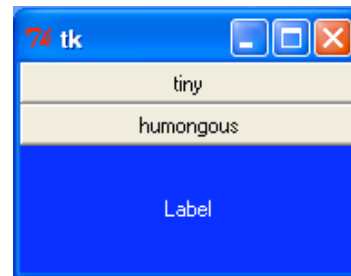
Buttons fill out their horizontal space (X)



Tkinter – Expanding

Expanding - filling

```
>>> Button(root, text="tiny").pack(fill=X)  
>>> Button(root, text="humongous").pack(fill=X)  
>>> w = Label(root, text="Label", bg="blue", fg="white")  
>>> w.pack(fill=BOTH, expand=True)
```



expanded

Tkinter – Frames

- Ένας χώρος για τοποθέτηση widgets
- Χρησιμοποιείται για την ομαδοποίηση widget

```
>>> root = Tk()
>>> f = Frame(root)
>>> Label(f, text="Name :").pack(side=LEFT)
>>> Entry(f).pack(side=RIGHT, fill=X, expand=True)
>>> f.pack()
>>> root.mainloop()
```



Tkinter – Frames

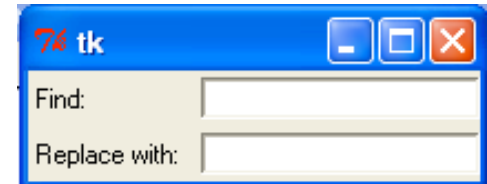
Παράδειγμα:

```
root = Tk()

find = EntryField(root, "Find:")

find.pack(side=TOP, fill=X, pady=1)

replace = EntryField(root, "Replace with:")
replace.pack(side=TOP, fill=X, pady=1)
```



What is PADX and Pady?

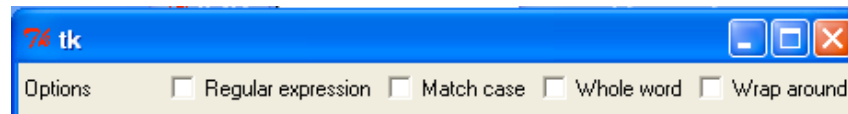
The **padx** puts some space between the button widgets and between the closeButton and the right border of the root window. The **pady** puts some space between the button widgets and the borders of the frame and the borders of the root window

Tkinter – Frames: Option bar

```
class Optionbar(Frame):
    def __init__(self, parent, label, options, labelwidth=12):
        Frame.__init__(self, parent)
        l = Label(self, text=label, width=labelwidth, anchor=W)
        l.pack(side=LEFT)
        for option in options:
            cb = Checkbutton(self, text=option)
            cb.pack(side=LEFT, anchor=W, expand=True)
```

Παράδειγμα:

```
root = Tk()
options = OptionBar(root, "Options",
    ["Regular expression", "Match case", "Whole word", "Wrap around"])
```

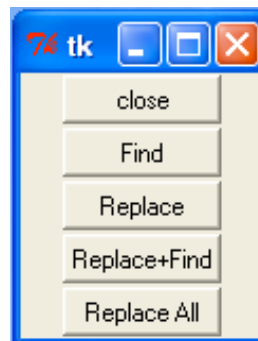


Tkinter – Frames: A series of buttons

```
class ButtonList(Frame):  
    def __init__(self, parent, buttons): Frame.__init__(self, parent)  
        for b in buttons: Button(self, text=b).pack(side=TOP, fill=X, pady=1)
```

Παράδειγμα:

```
root = Tk()  
buttons = ButtonList(root, ["close", "Find", "Replace", "Replace+Find", "Replace All"])  
buttons.pack()
```





Tkinter – Pre-built Widgets

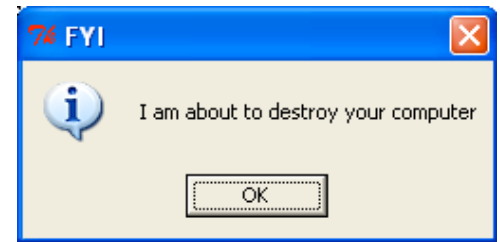
Το Tkinter έχει αρκετά pre-built widgets όπως:

- Standard dialogs
- Απλή εισαγωγή δεδομένων
- Επιλογή αρχείων και χρωμάτων

Tkinter – Pre-built Widgets: Standard Dialogs

Informational dialog

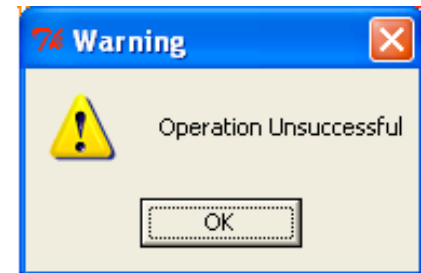
```
>>> from tkMessageBox import *  
>>> showinfo("FYI","I am about to  
destroy your computer")
```



Tkinter – Pre-built Widgets: Standard Dialogs

Warning dialog

```
>>> from tkMessageBox import *  
>>> showwarning("Warning", "Operation  
Unsuccessful")
```



Tkinter – Pre-built Widgets: Standard Dialogs

Error dialog

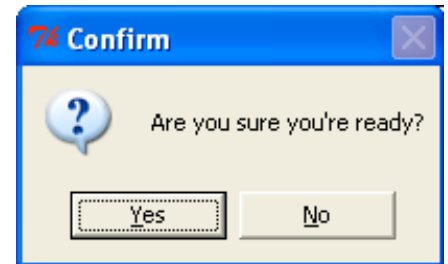
```
>>> from tkMessageBox import *  
>>> showerror("Fatal Error", "Everything  
is hosed!")
```



Tkinter – Pre-built Widgets: Standard Dialogs

Yes/No dialog

```
>>> from tkMessageBox import *  
>>> askyesno("Confirm", "Are you sure  
you're ready?")
```

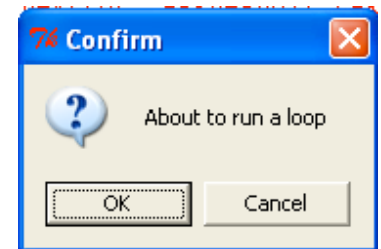


Επιστρέφει True/False

Tkinter – Pre-built Widgets: Standard Dialogs

Ok/Cancel Dialog

```
>>> from tkMessageBox import *  
>>> askokcancel("Confirm", "About to run a loop  
run a loop")
```

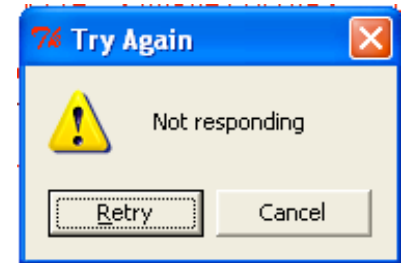


Επιστρέφει True/False

Tkinter – Pre-built Widgets: Standard Dialogs

Retry/Cancel Dialog

```
>>> from tkMessageBox import *  
>>> askretrycandle("Try Again", "Not  
responding")
```

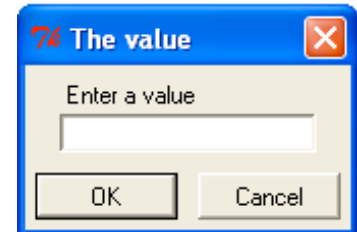


Επιστρέφει True/False

Tkinter – Pre-built Widgets: Entry Dialogs

Εισαγωγή string, integers, floats

```
>>> from tkSimpleDialog import *  
>>> askinteger("The value", "Enter a value")  
  
42  
  
>>>
```



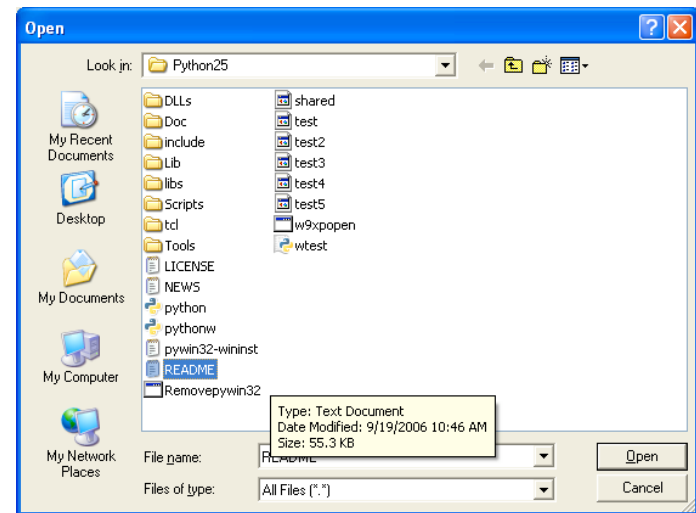
Εναλλακτικά:

```
askinteger()  
askfloat()  
askstring()
```

Tkinter – Pre-built Widgets: Filename Dialog

Επιλογή αρχείου για άνοιγμα

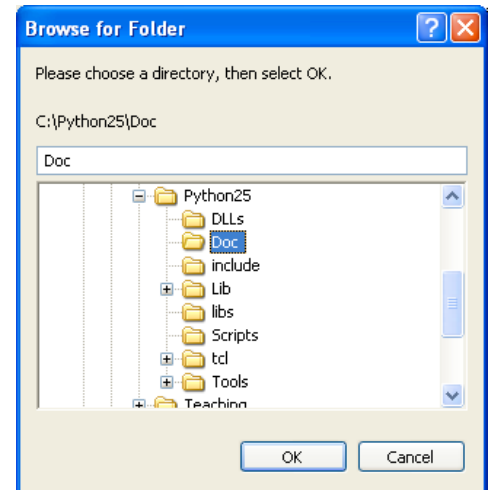
```
>>> from tkFileDialog import *  
>>> askopenfilename()  
'C:/Python25/README.txt'  
>>>
```



Tkinter – Pre-built Widgets: Directory Dialog

Επιλογή φακέλου

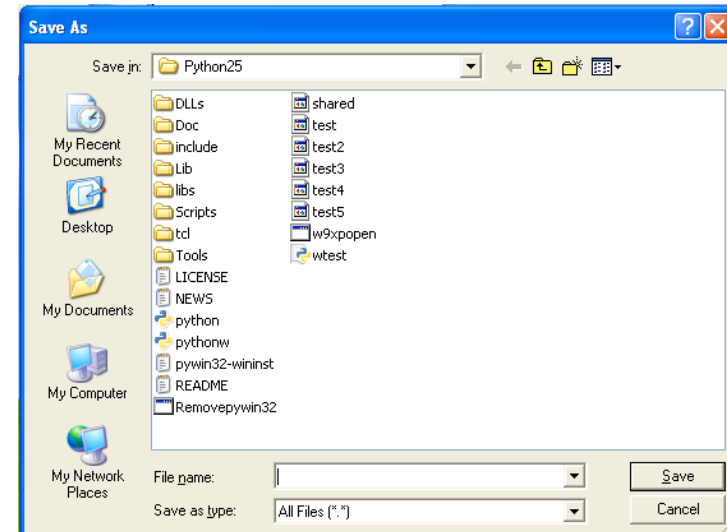
```
>>> from tkFileDialog import *  
>>> askdirectory()  
  
'C:/Python25/Doc'  
>>>
```



Tkinter – Pre-built Widgets: Save as Dialog

Επιλογή ονόματος αρχείου για αποθήκευση

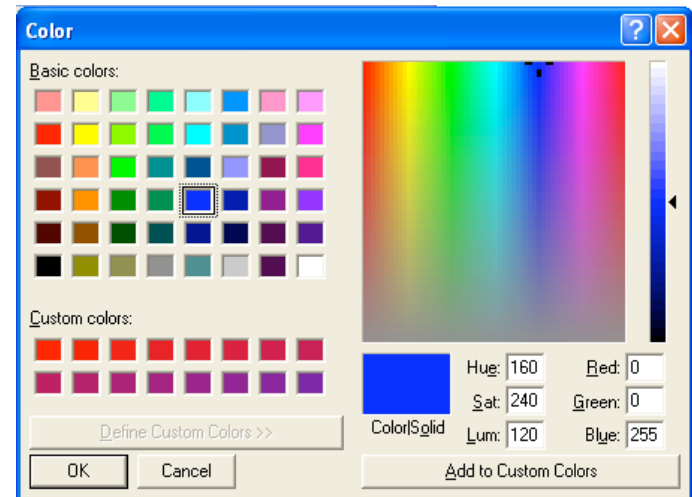
```
>>> from tkinterFileDialog import *  
>>> asksaveasfilename()
```



Tkinter – Pre-built Widgets: Color chooser

Επιλογή χρώματος

```
>>> from tkColorChooser import *  
>>> askcolor()  
  
((0,0,255), '#0000ff')  
>>>
```



Tkinter – Τα widgets σαν αντικείμενα 1/2

Μέθοδος:

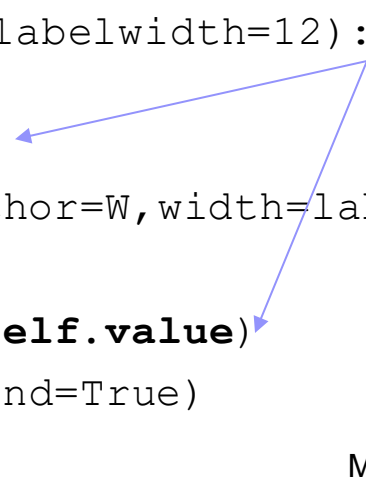
- Μπορούμε να ορίσουμε κάθε widget σαν μια κλάση (συνήθως κληρονομεί την Frame)
- Να αποθηκεύσουμε όλα τα ως attribute των αντικειμένων
- Να δημιουργήσουμε μεθόδους για τη πρόσβαση στα απαραίτητα δεδομένα

Tkinter – Τα widgets σαν αντικείμενα 2/2

```
class EntryField(Frame):  
    def __init__(self, parent, label, labelwidth=12):  
        Frame.__init__(self, parent)  
        self.value = StringVar(self)  
        l = Label(self, text=label, anchor=W, width=labelwidth)  
        l.pack(side=LEFT)  
        e = Entry(self, textvariable=self.value)  
        e.pack(side=RIGHT, fill=X, expand=True)  
  
    def get_value(self):  
        return self.value.get()
```

Attribute is created to hold value of entry field

Method that returns the current value



Tkinter – Τα widgets σαν functions 1/2

Μέθοδος:

- Μπορούμε να γράψουμε μια function που δημιουργεί ένα widget
- Να αποθηκεύσουμε όλα τα state μέσα στη function
- Να χρησιμοποιήσουμε μια function για την επιστροφή

Tkinter – Τα widgets σαν functions 2/2

```
def entryfield(parent, label, labelwidth=12, **packopts):
```

```
    f = Frame(parent)
```

```
    f.pack(**packopts)
```

```
    l = Label(f, text=label, width=labelwidth)
```

```
    l.pack(side=LEFT, anchor=W)
```

```
    value = StringVar(f)
```

```
    e = Entry(f, textvariable=value)
```

```
    e.pack(side=RIGHT, fill=X, expand=True)
```

```
    return lambda: value.get()
```

Δημιουργία Widgets

Μεταβλητή state

Function για επιστροφή
state