

Università degli Studi di Bologna
Polo di Forlì

Fondamenti di Programmazione
MINI CORSO SULLA COSTRUZIONE DI GUI CON
TKINTER TK IN PYTHON

Docente: Dott. Gianluigi Roveda

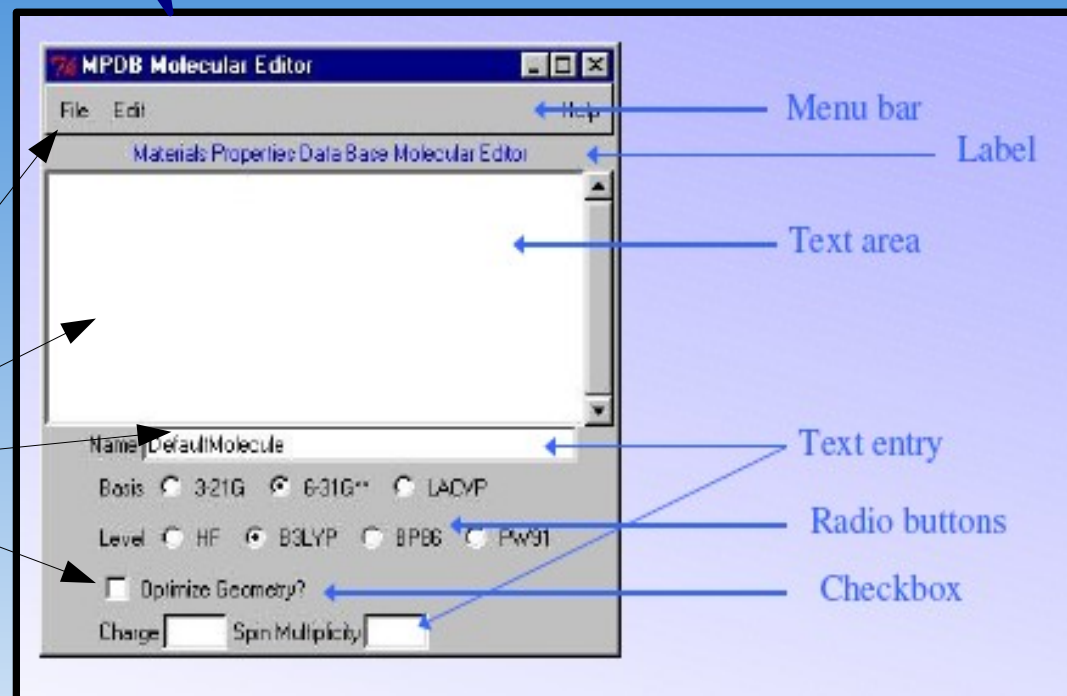
Sommario

- 1. Anatomia di una interfaccia grafica**
- 2. Gestione degli eventi con il modulo Tkinter TK**
- 3. Il primo programma con Tkinter**
- 4. Gerarchia delle classi di Tkinter TK**
- 5. Contenitori e Widget**
- 6. Metodi comuni per widget**
- 7. Impacchettare widget**
- 8. Alcuni attributi del metodo pack**
- 9. Il widget label**
- 10. Il widget button**
- 11. Il widget checkbutton**
- 12. Il widget radiobutton**
- 13. I widget entry e text**

Anatomia di una interfaccia grafica

1 di 2

FINESTRA



WIDGET

Finestra (window): Elemento dei sistemi operativi GUI. Una finestra serve a mostrare il contenuto di un disco o di una directory (finestre di navigazione del sistema operativo), un documento (finestra di documento) in lavorazione, un'applicazione (finestra di applicazione) in funzione, un messaggio per l'utilizzatore (finestra di avviso) o per inserire i parametri per l'esecuzione di un comando (finestra di dialogo).

Widget (derivazione dal gergo settoriale professionale): Elemento dell'interfaccia utente. Un tempo slang dei programmatori, ora termine tecnico di tutto rispetto. Pulsanti, barre di scorrimento, campi di modifica, elenchi ad albero e così via sono tutti widget. I programmatori combinano questi elementi nei programmi con interfaccia grafica. Un widget può scatenare un'evento, ad ogni **evento** associato a un widget e a sua volta associato un metodo che lo gestisce

Gestione degli eventi con il modulo Tkinter TK

Versione del toolkit TK per la costruzione di GUI per

**PASSAGGIO DEL MOUSE
SU UN WIDGET**

**PRESSIONE DI
UN BOTTONE**



**SELEZIONE DI
UN CHECKBOX**

Il primo programma con Tkinter

Importo il modulo Tkinter

```
from Tkinter import *
```

```
r=Tk()
```

```
a=Frame(r)
```

```
a.pack()
```

```
l=Label(a,text="ciao mondo", justify = LEFT)
```

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

```
r.mainloop()
```

esegue il **ciclo d'attesa degli eventi** come metodo dell'oggetto radice

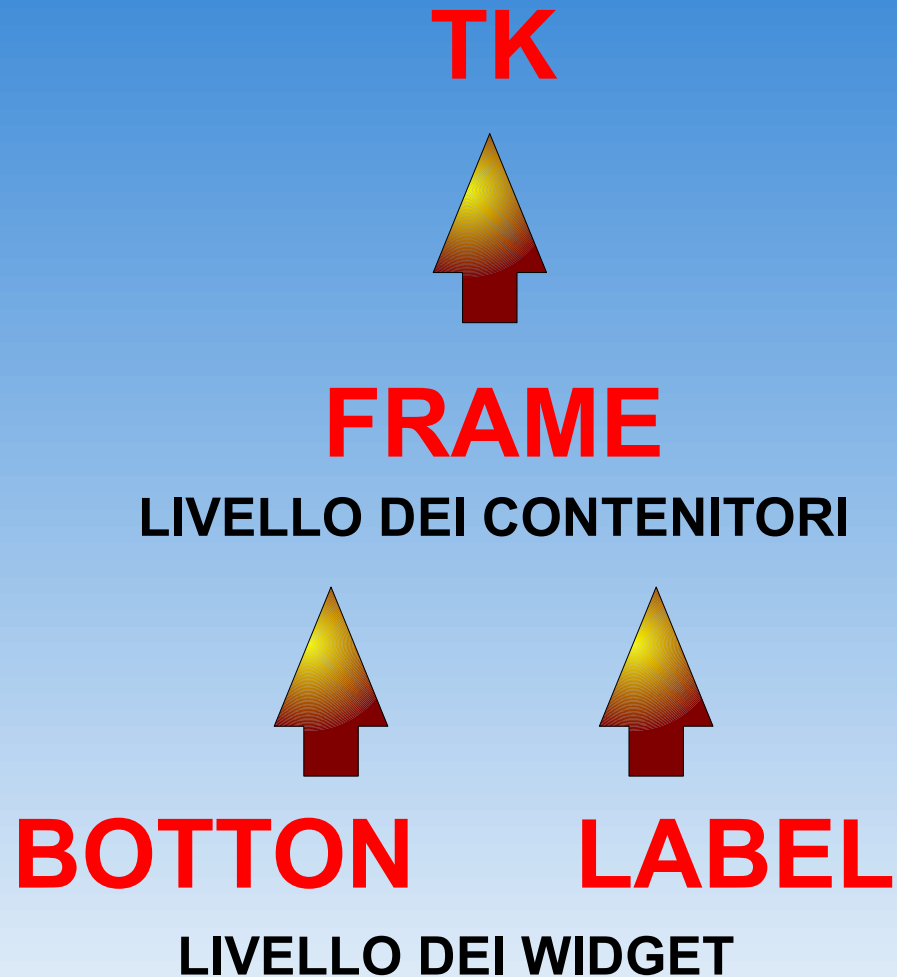
Generazione di un'oggetto di classe Tkinter.Tk comportando la creazione di una finestra principale

Creazione di un oggetto contenitore frame aggiungendolo all'oggetto r

Impacchettamento: processo finalizzato a stabilire una relazione visuale fra una componente GUI e il suo genitore

Aggiungo all'oggetto frame, l'oggetto etichetta che possiede alcune proprietà.

Gerarchia delle classi di Tkinter TK



Contenitori & Widget

3 Contenitori:

- **Frame**: riquadro che contiene altri widget
- **Canvas**: tela, area per visualizzare/modifica immagini
- **TopLevel**: una finestra separata, che può contenere altri elementi

12 Widget:

- **Label**: etichetta, è una scritta o un'immagine
- **Button**: un pulsante
- **Entry**: area per immissione di caratteri
- **Message**: messaggio pop-up
- **Text**: area contenente testo
- **RadioButton**: pulsanti per effettuare una scelta
- **CheckButton**: pulsanti on-off
- **Listbox**: Elenco di possibili scelte alternative
- **Menu/MenuButton**: per costruire menu
- **Scale**: un indicatore di aspetto simile ad una leva che può assumere diversi valori
- **Scrollbar**: barra di scorrimento che può essere associata a canvas, entry, listbox e text widget

Metodi comuni per widget

Configure: modifica gli attributi di un'oggetto, esempio:

```
l = Label(text="Qwi c'è un errore")  
l.pack()  
l.configure(text = 'Qui c'è un errore')
```

cget: metodo che permette di conoscere il valore di un certo attributo, esempio:

```
l = Label(text='ciao mondo')  
l.cget('text')
```

keys: metodo che restituisce una lista di attributi per un dato widget, esempio:

```
a = l.keys()
```

Impacchettare widget

Tutti gli oggetti sono impacchettati nel contenitore che viene organizzato come una tavola bidimensionale suddivisa in righe e colonne.

Pack()
grid()
place()

Tutti gli oggetti impacchettati nello stesso contenitore da pack sono visualizzati nell'ordine d'impacchettamento e l'interfaccia realizzata risulta della grandezza minima necessaria per contenere i widget

Gli oggetti vengono impacchettati nella finestra nella posizione e con le dimensioni richieste esplicitamente in termini assoluti (pixel) o relativi alle caratteristiche dimensionali della finestra.

Impacchettare widget

Alcune opzioni del metodo PACK

`expand = booleano`

espande un widget se il suo contenitore aumenta le dimensioni;

`fill = NONE X, NONE Y, BOTH`

inserisce widget se i contenitori crescono.

`In = "master"`

usa il contenitore principale come contenitore del widget

`side = TOP, BOTTOM, LEFT, RIGHT`

indica dove porre, visivamente, il bottone

Impacchettare widget

Alcune opzioni del metodo PLACE

<http://docs.huihoo.com/tkinter/an-introduction-to-tkinter-1997/place.htm>

`anchor = N, NE, E, SE, SW, W, NW (Default), CENTER`

specifica quale parte del widget deve essere visualizzata alla posizione specificata

`relwidth, relheight = reale (float)`

larghezza e altezza del widget espressa con un valore tra 0 e 1, dove 1 equivale alle dimensioni del loro contenitore.

`relx, rely = reale (float)`

posizione relativa al contenitore

`width, height = intero.`

Per default, non modifica le dimensioni del widget

`x, y = intero`

posizione nel contenitore padre

Esempio: `Button(text=str(i)).place(x=i*30, y=i*30)`

Impacchettare widget

Alcune opzioni del metodo GRID

<http://docs.huihoo.com/tkinter/an-introduction-to-tkinter-1997/grid.htm>

`column` = intero (default 0)

inserisce il widget nella colonna `column`-esima

`columnspan`, `rowspan` = intero (default 0)

per far occupare più di una riga o più di una colonna al widget

`ipadx`, `ipady`, `padx`, `pady` = intero

spazio tra celle e elementi esterni e interni

`row` = intero (default: 0 o la prima riga vuota)

inserisce il widget nella riga `row`-esima

`Sticky` = combinazione di S, N, E, W, NW, NE, SW, SE (separatore: +)

decide come espandere la finestra se necessario

Esempio:

```
Label(text = 'First').grid(row=0,sticky=W)
```

```
Label(text = 'Second').grid(row=1,sticky=W)
```

```
Entry().grid(row=0,column=1)
```

```
Entry().grid(row=1,column=1)
```

```
Button(text='Press me').grid(row=0,column=2,columnspan=2,rowspan=2)
```

Impacchettare widget

Cancellare l'elemento visualizzato

pack_forget()

place_forget()

grid_forget()

**Toglie il widget dalla
finestra visualizzata**



Il widget Label

background	#eae9e8	
foreground	#000000	■
active bg	#eae9e8	
active fg	#eae9e8	
highlight bg	#eae9e8	
highlight color	Black	■
disabled fg	#a3a3a3	■
relief	raised	▼
borderwidth	1	
highlight bd	0	
x pad	1	
y pad	1	
height	0	
width	0	
bitmap		
cursor		
image		...
text	label	
text var		
font	Verdana -13	...
justify	center	▼
wrap length	0	
underline	-1	
anchor	center	▼
state	normal	▼
take focus	0	▼

```
import Tkinter
w=Tkinter.Label(text='Hello, World').pack()
Tkinter.mainloop()
```



Il widget Button

1 di 2

background	#eae9e8
foreground	#000000
active bg	#ececec
active fg	Black
highlight bg	#eae9e8
highlight color	Black
disabled fg	#808080
relief	raised
borderwidth	2
highlight borderwidth	1
x pad	3m
y pad	1m
height	0
width	0
bitmap	
cursor	
image	
command	
text	button
text var	
font	Verdana -13
justify	center
wrap length	0
underline	-1
anchor	center
repeat delay	0
repeat interval	0
state	normal
take focus	

```
from Tkinter import *
```

```
def ciao():  
    print "ciao"
```

```
r=Tk()
```

```
a=Frame(r)
```

```
a.pack()
```

```
l=Label(a,text="ciao mondo", justify = LEFT)
```

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

```
b=Button(a,text="ok",command=ciao)
```

```
b.pack(anchor = E)
```

```
r.mainloop()
```



Button definisce una funzione callback, che viene eseguita quando il pulsante viene premuto, ovvero quando mainloop sospende il suo ciclo.

Il widget Button

2 di 2

```
from Tkinter import *
```

```
r=Tk()
```

```
a=Frame(r)
```

```
a.pack()
```

```
l=Label(a,text="ciao mondo", justify = LEFT)
```

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

```
p1=Button(a)
```

```
p1["text"]="OK"
```

```
p1["background"]="green"
```

```
p1.pack()
```

```
r.mainloop()
```



Il widget checkbutton

background	#eae9e8
foreground	#000000
active bg	#ececec
active fg	Black
highlight bg	#eae9e8
highlight color	Black
select color	yellow
disabled fg	#a3a3a3
relief	flat
borderwidth	2
highlight bd	1
x pad	1
y pad	1
height	0
width	0
bitmap	
cursor	
image	
select image	
command	
text	check
text var	
font	Verdana -13
justify	center
wrap length	0
underline	-1
anchor	center
indicator	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
variable	che32
off value	0
on value	1
state	normal
take focus	

```
from Tkinter import *
```

```
def ciao(a,b):  
    print "Valori:"  
    if (a==""):  
        print "spento\n"  
    print(a)  
    if (b==""):  
        print "spento\n"  
    print(b)
```

```
r=Tk()
```

```
a=Frame(r)
```

```
a.pack()
```

```
l=Label(a,text="ciao mondo", justify = LEFT)
```

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

```
v1 = StringVar(r)
```

```
v2 = StringVar(r)
```

```
c1=Checkbutton(a,text="op 1",variable=v1,onvalue="acceso\n",offvalue="spento\n")
```

```
c1.pack(anchor = W)
```

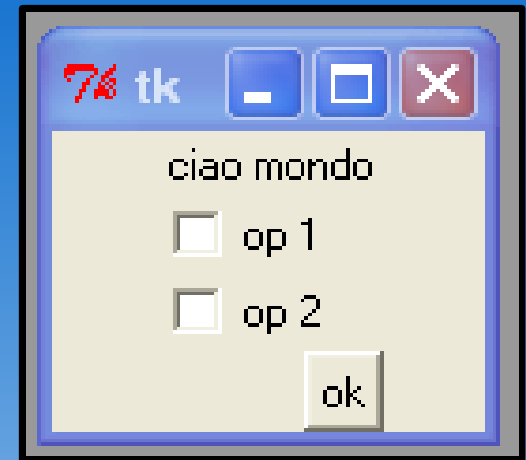
```
c2=Checkbutton(a,text="op 2",variable=v2,onvalue="acceso\n",offvalue="spento\n")
```

```
c2.pack(anchor = W)
```

```
b=Button(a,text="ok",command = lambda:ciao(v1.get(),v2.get()))
```

```
b.pack(anchor = E)
```

```
r.mainloop()
```



Il widget radiobutton

background	#eae9e8
foreground	#000000
active bg	#ececec
active fg	Black
highlight bg	#eae9e8
highlight col	Black
select color	yellow
disabled fg	#a3a3a3
relief	flat
borderwidth	2
highlight bd	1
x pad	1
y pad	1
height	0
width	0
bitmap	
cursor	
image	...
select image	...
command	...
text	radio
text var	
font	Verdana -13
justify	center
wrap length	0
underline	-1
anchor	center
indicator	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
variable	selectedButton
value	
state	normal
take focus	

```
from Tkinter import *
```

```
scelte = ['Poco', 'Abbastanza', 'Molto', 'Non so, devo ancora pensarci']
```

```
t = Tk()
```

```
Label(text='Ti piace TKINTER ?').pack()
```

```
v = StringVar()
```

```
for oggetto in scelte:
```

```
    Radiobutton(text=oggetto, variable=v,value=oggetto).pack(anchor=W)
```

```
t.mainloop()
```



Il widget entry

background	#ffffff	
foreground	Black	■
highlight bg	#eae9e8	■
highlight col	Black	■
select bg	#a9d1ff	■
select fg	#030303	■
disabled fg	#a3a3a3	■
insert bg	Black	■
relief	sunken	▼
borderwidth	2	
insert bd	0	
select bd	1	
highlight bd	1	
width	20	
insert wd	2	
cursor	xterm	
x scroll cm		...
text var		
font	Verdana -13	...
justify	left	▼
insert off	300	
insert on t	600	
export	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	
validate	none	
validate Cm		...
Inv. Cmd		...
show		
state	normal	▼
take focus		▼

Metodi di maggior utilità:

get(): restituisce la stringa visualizzata

insert(index, text): inserisce la stringa text in posizione index

```
from Tkinter import *
```

```
t = Tk()
Label(text='Inserisci testo').pack()
e = Entry()
e.insert(0, "ciao m")
e.pack()
t.mainloop()
```



Il widget text

background	#ffffff
foreground	Black
highlight bg	#eae9e8
highlight col	Black
select bg	#a9d1ff
select fg	#030303
insert bg	Black
relief	sunken
borderwidth	2
insert bd	0
select bd	1
highlight bd	1
x pad	1
y pad	1
height	4
width	15
insert wd	2
cursor	xterm
x scroll cm	...
y scroll cm	...
font	Verdana -13
wrap	char
spacing1	0
spacing2	0
spacing3	0
tabs	
insert off	300
insert on t	600
export	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
set grid	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
state	normal
take focus	

Metodi di maggiore importanza:

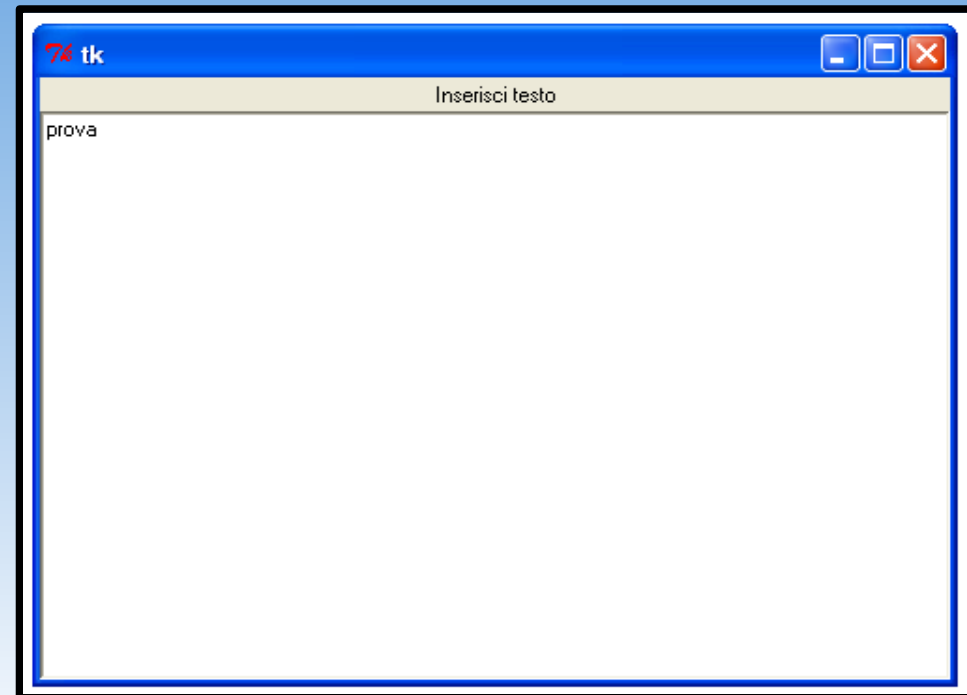
- Get(1.0, GET): restituisce la stringa visualizzata
- insert(1.0, text): inserisce la stringa text in posizione index

```
from Tkinter import *
```

```
t = Tk()  
Label(text='Inserisci testo').pack()
```

```
textbox = Text()  
textbox.insert(1.0, "prova")  
textbox.pack()
```

```
t.mainloop()
```



tkMessageBox

(import tkMessageBox)

Metodi:

- Askokcancel,
- Askquestion,
- Askretrycancel,
- askyesno,
- showerror,
- showinfo,
- showwarning

```
askokcancel('Titolo finestra', Domanda')
```



```
showinfo('Titolo finestra', 'Attenzione!')
```



Sitografia

Fredrik Lundh, *An introduction to Tkinter*

www.pythonware.com/library/tkinter/introduction/index.htm

- John W. Shipman, *Tkinter reference: a GUI for Python*

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

- Alan Gauld, *Imparare a programmare,*

www.freenetpages.co.uk/hp/alan.gauld/italian/

- *Tkinter per sopravvivere*

http://linuxdidattica.org/docs/altre_scuole/msm_p/txs_01.html