

# Text widget tkinter

Il widget text può essere usato sia per salvare l'input di testo dell'utente che per stampare del testo, in entrambi i casi su più linee



# La sintassi di base

`txt1 = Text(Frame)`

```
parola1  
parola2  
parola3  
parola4
```

# Parametri utili

- `height=0` determina l'altezza del widget, è misurato in righe
- `width=0` determina la larghezza del widget, è misurato in caratteri per riga
- `padx-pady=0` determinano rispettivamente il padding a sinistra-destra e sopra-sotto e si misura in pixel

- Wrap= determina il modo in cui si va a capo in caso di riga troppo lunga. CHAR va a capo con un singolo carattere(spezza le parole),WORD va a capo portandosi dietro tutta la parola(non spezza la parola),NONE non va a capo in automatico
- State= determina lo stato del widget:
  - DISABLED: il widget è congelato non può essere usato
  - NORMAL: il widget è normalmente utilizzabile

# Altri parametri

bg	Colore di sfondo
cursor	Per cambiare il cursore quando in mouse è sopra al widget
font	Cambia font e grandezza carattere
fg	Cambia il colore dei caratteri
insertwidth	Larghezza del cursore di inserimento in pixel

# Metodi utili

- `testo=txt1.get("1.0",END)` : questo metodo ritorna il testo del widget presente nel range dato(in questo caso ritorna tutto il testo)
- `self.txtA2.insert("1.0","Testo riga 1 colonna 0")` = inserisco il testo a partire dalla riga e colonna data
- `self.txtA2.delete("1.0",END)` = elimina il testo presente in uno specifico range(in questo caso cancella tutto il testo)

- `txt1.tag_add("tag1", "1.0", "1.6")`= creo un tag chiamato "tag1" delimitandolo in un range
- `txt1.tag_config("tag1", foreground="red", background="yellow")`= configuro il tag "tag1" creato precedentemente dandogli rispettivamente il nome, il colore del carattere e il colore dell'evidenziazione del testo

# Codice di esempio

```
1 from tkinter import *
2
3 class MultiLineText(Tk):
4     #costruttore
5     def __init__(self):
6         super().__init__()
7         self.title("Multi Line Text")
8         self.geometry("475x500")
9         self.resizable(0,0)
10        self.crea_widgets()
11        self.crea_widgets2()
12
13
14
15    def crea_widgets(self):
16        mf=Frame(self)
17        mf.grid()
18        #Multi line Text widget
19        self.lbl1=Label(mf,text="text come multiline input text")
20        self.lbl1.grid(row=0,column=0)
21
22        self.txtA = Text(mf, width=50, height=10, wrap= WORD) # creazione widjed width: numero colonne height: numero righe wrap: WORD=va a capo senza spezzare le parole CHAR= va a capo spezzando le parole
23        self.txtA.grid(row=1,column=0) #inserisco widget nel Frame
24
25        self.btn=Button(mf,text="stampa testo",command=self.stampa)
26        self.btn.grid(row=2,column=0,pady=2,padx=2)
27
28        self.lbl2=Label(mf,text="text come multiline output text")
29        self.lbl2.grid(row=3,column=0)
30
31        self.txtA2 = Text(mf, width=50, height=10, wrap= CHAR) # creazione widjed width: numero colonne height: numero righe wrap: WORD=va a capo senza spezzare le parole CHAR= va a capo spezzando le parole
32        self.txtA2.grid(row=4,column=0) #inserisco widget nel Frame
33        self.txtA2["state"] = DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
34
35
36    def crea_widgets2(self):
37        mf2=Frame(self)
38        mf2.grid()
39
40        self.btn=Button(mf2,text="scrivo all inizio",command=self.SI)
41        self.btn.grid(row=1,column=0,pady=2)
42
43        self.btn=Button(mf2,text="scrivo alla fine",command=self.SF)
44        self.btn.grid(row=1,column=1,pady=2)
45
46
47        self.btn=Button(mf2,text="cancello tutto",command=self.CancAll)
48        self.btn.grid(row=1,column=2,pady=2)
49
50        self.btn=Button(mf2,text="colora primi 6 caratteri",command=self.colora)
51        self.btn.grid(row=1,column=3,pady=2)
```



```

40 self.btn=Button(mf2,text="scrivo all inizio",command=self.SI)
41 self.btn.grid(row=1,column=0,pady=2)
42
43 self.btn=Button(mf2,text="scrivo alla fine",command=self.SF)
44 self.btn.grid(row=1,column=1,pady=2)
45
46
47 self.btn=Button(mf2,text="cancello tutto",command=self.CancAll)
48 self.btn.grid(row=1,column=2,pady=2)
49
50 self.btn=Button(mf2,text="colora primi 6 caratteri",command=self.colora)
51 self.btn.grid(row=1,column=3,pady=2)
52
53 self.btn=Button(mf2,text="elimina colore",command=self.eliminaColore)
54 self.btn.grid(row=1,column=4,pady=2)
55
56
57 def stampa(self):
58     txt=self.txtA.get("1.0",END) # prende tutto il testo in txtA dalla riga 1 colonna 0 fino alla fine "END"
59     print(txt)
60
61 def SI(self):
62     self.txtA2["state"]=NORMAL #abilito il widget in modo da poter scrivere
63     self.txtA2.insert("1.0","Testo inizio") # inserisco il testo all inizio riga 1 colonna 0
64     self.txtA2["state"]=DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
65
66 def SF(self):
67     self.txtA2["state"]=NORMAL #abilito il widget in modo da poter scrivere
68     self.txtA2.insert(END,"Testo fine") # inserisco il testo alla fine END
69     self.txtA2["state"]=DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
70
71 def CancAll(self):
72     self.txtA2["state"]=NORMAL #abilito il widget in modo da poter scrivere
73     self.txtA2.delete("1.0",END) # inserisco il testo alla fine END
74     self.txtA2["state"]=DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
75
76 def colora(self):
77     self.txtA2["state"]=NORMAL #abilito il widget in modo da poter scrivere
78     self.txtA2.tag_add("rosso", "1.0", "1.6") #aggiungo un tag a txtA2 dandogli il nome e il range di caratteri che influisce
79     self.txtA2.tag_config("rosso", foreground="red", background="yellow") #configuro il tag "rosso" in modo da rendere il colore del carattere rosso e evidenziarlo di giallo
80     self.txtA2["state"]=DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
81
82 def eliminaColore(self):
83     self.txtA2["state"]=NORMAL #abilito il widget in modo da poter scrivere
84     self.txtA2.tag_delete("rosso")
85     self.txtA2["state"]=DISABLED # disabilito la casella di testo in modo che l'utente non ci possa scrivere
86
87 def main():
88     f=MultiLineText()
89     f.mainloop()
90
91 main()
92

```

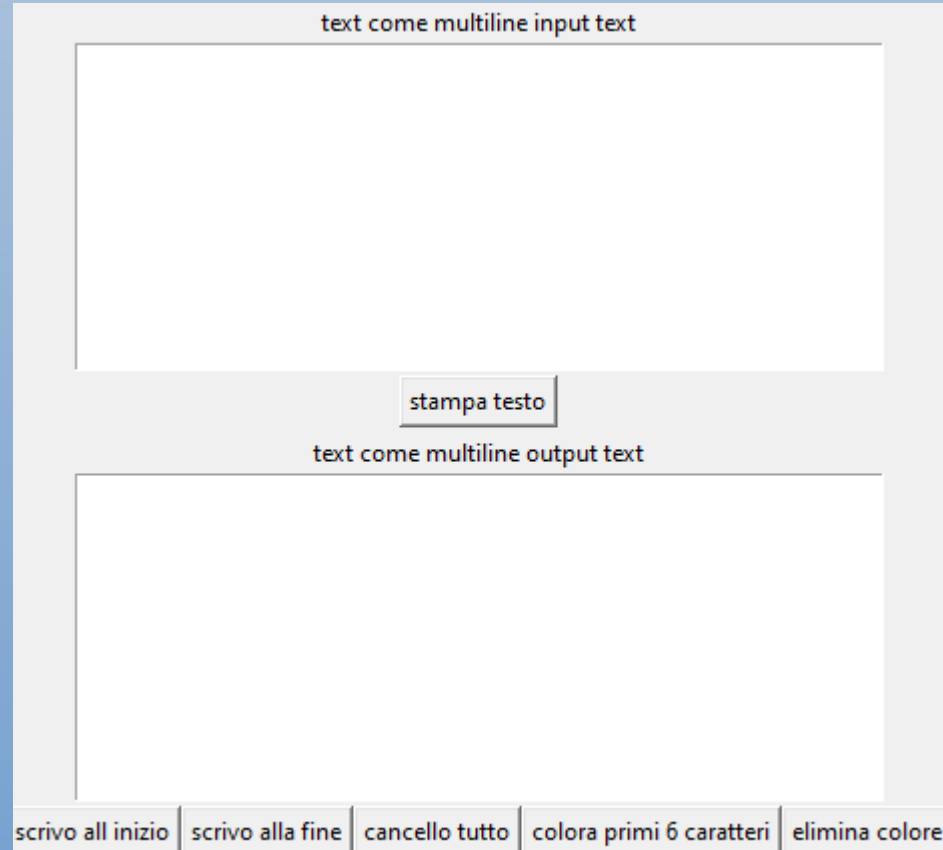
# Esecuzione programma

text come multiline input text

stampa testo

text come multiline output text

scrivo all inizio | scrivo alla fine | cancello tutto | colora primi 6 caratteri | elimina colore



# Text + Scrollbar

La Scrollbar unita alla Text può essere molto comoda perché ci permette di navigare meglio nelle righe e nelle colonne del widget

# Implementazione

Prima di tutto è necessario creare la scrollbar:  
`verticalScroll=Scrollbar(mf, orient=VERTICAL)`  
orient= determina l'orientamento della  
scrollbar(VERTICAL,HORIZONTAL)

Poi creare la text:

txtA =

`Text(Frame, yscrollcommand=verticalScroll.set)`

y/scrollcommand= richiama la set della scrollbar  
quando la scrollbar necessita di essere aggiornata  
(es. quando aggiungiamo o cancelliamo righe)

Poi configurare la scrollbar:

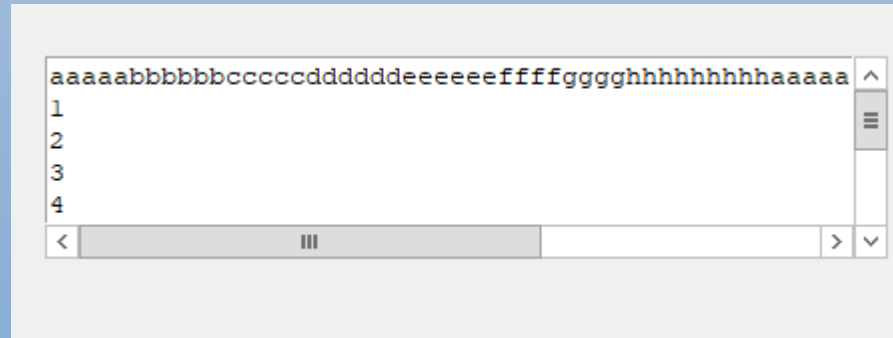
```
verticalScroll.config(command= self.txtA.yview)
```

Command= fa in modo che la y/xview della textA vengano modificati quando agisco sulla scrollbar

# Codice di esempio

```
1 from tkinter import *
2
3 class MultiLineText(Tk):
4     #costruttore
5     def __init__(self):
6         super().__init__()
7         self.title("Multi Line Text Scrollbar")
8         self.geometry("475x300")
9         self.resizable(0,0)
10        self.crea_widgets()
11
12    def crea_widgets(self):
13        mf=Frame(self)
14        mf.grid(pady=30,padx=30)
15        verticalScroll=Scrollbar(mf, orient=VERTICAL) #creo la scrollbar, orient=VERTICAL imposta la scrollbar in verticale
16        horizontalScroll=Scrollbar(mf, orient=HORIZONTAL) #creo la scrollbar, orient=HORIZONTAL imposta la scrollbar in orizzontale
17
18        self.txtA = Text(mf, width=50, height=5, wrap= NONE, yscrollcommand=verticalScroll.set, xscrollcommand=horizontalScroll.set) # creazione widget width: numero colonne height: numero
        righe wrap: WORD=va a capo senza spezzare le parole CHAR= va a capo spezzando le parole y/xscrollcommad= richiama la set della scrollbar quando la scrollbar necessita di essere aggiornata
        (es. quando aggiungiamo o cancelliamo righe)
19        self.txtA.grid(row=1,column=0) #inserisco widget nel Frame
20        self.txtA.insert("1.0", "aaaaabbbbccccddddddeeeeffffgggghhhhhhhhaaaaabbbbccccddddddeeeeffffggggg\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10")
21        # self.txtA["state"]=DISABLED
22
23        verticalScroll.grid(row=1,column=2,sticky=W+N+S, rowspan=2) # inserisco il widget
24        horizontalScroll.grid(row=2,column=0,sticky=W+E) # inserisco il widget
25
26        verticalScroll.config(command= self.txtA.yview) # configuro la scrollbar il metodo yview della txtA viene richiamato quando agisco sulla scrollbar
27        horizontalScroll.config(command= self.txtA.xview) # configuro la scrollbar il metodo xview della txtA viene richiamato quando agisco sulla scrollbar
28
29    def main():
30        f=MultiLineText()
31        f.mainloop()
32
33|
34 main()
```

# Esecuzione programma





# Tkinter

