



# **Tema 21.**

## **Text, ScrollBar, ComboBox și MessageBox în Tkinter**

Ce ne așteaptă?

1. Componenta - Text
2. Componenta - ScrollBar
3. Componenta - ComboBox
4. Ferestre - MessageBox

1. Ce reprezintă componenta Text și cum se creează?
2. Care sunt parametrii de bază ai componentei Text?
3. Cum se înscrie, citește și șterge datele din Text?
4. Ce reprezintă componenta ScrollBar și cum se creează?
5. Care sunt parametrii de bază ai componentei ScrollBar?
6. Cum se asociază ScrollBar unei componente Text
7. Ce reprezintă componenta ComboBox și cum se creează?
8. Care sunt parametrii de bază ai componentei ComboBox?
9. Cum se setează și cum se citesc valorile selectate în ComboBox?
10. Cum se aplica evenimentul <<ComboboxSelected>> ?
11. Ce reprezintă ferestrele MessageBox și cum se creează?
12. Care sunt particularitățile fiecărei ferestre MessageBox

# 1. Componenta - Text

## Crearea și plasarea componentei - Text

- Se utilizează pentru a permite utilizatorului să introducă un mesaj pe mai multe linii
- Se creează utilizând clasa Text()

```
text = Text(root)
```

- Plasarea componentei

```
text.pack()
```

- Exemplu de utilizare

```
from tkinter import *  
root=Tk()  
root.geometry('300x200')  
text = Text(root)  
label = Label(root, text='Introdu o mica descriere a propriei persoane')  
label.pack()  
text.pack()  
root.mainloop()
```

## Parametrii ai componentei Text (1)

- Componenta Text acceptă unii parametri specificați la componenta Label: bg, fg, font, padx, pady
- Exemplu de utilizare

```
from tkinter import *  
root=Tk()  
root.geometry('300x200')
```

```
text = Text(root)  
text.config(bg="red")  
text.config(fg="yellow")  
text.config(padx=50)  
text.config(pady=20)  
text.config(font="Tahoma 20 italic bold")
```

```
label = Label(root, text='Introdu o mica descriere a propriei persoane')  
label.pack()  
text.pack()  
root.mainloop()
```

## Parametrii ai componentei Text (2)

- **height** – înălțimea componentei (în înălțimi a caracterelor), implicit ocupă tot spațiul disponibil  
`text.config(height=3)`
- **width** – lățimea componentei (în caractere ), implicit ocupă tot spațiul disponibil  
`text.config(width=32)`
- **cursor** – tipul cursorului mouse-ului când se află pe componenta:  
"arrow", "circle", "clock", "cross", "fleur", "heart", "man", "plus", etc  
`text.config(cursor="circle")`
- **state** – activarea sau dezactivarea componentei: DISABLED, NORMAL  
`text.config(state=DISABLED)`  
`text.config(state=NORMAL)`
- **yscrollcommand** – permite setarea componentei scrollbar pe verticala după crearea acesteia  
`text.config(yscrollcommand=nume_scrollbar.set)`
- **xscrollcommand** – permite setarea componentei scrollbar pe orizontala după crearea acesteia  
`text.config(xscrollcommand=nume_scrollbar.set)`

## • Înscrierea, citirea și ștergerea conținutului Text

- Metoda `insert()` permite înscrierea programabilă a conținutului în Text:  
*`nume_variabila.insert(index_inceput, continut)`*
- Metoda `get()` - citește conținutul componentei și îl returnează sub formă de string  
*`data= nume_variabila.get()`*
- Metoda `delete()` – permite ștergerea programabilă a conținutului Text  
*`nume_variabila.delete(index_inceput, index_sfarsit)`*
- Pentru componenta Text indexul se va scrie in forma " nr\_linie.nr\_caracter " unde:
  - nr\_linie este numarul liniei și începe de la valoarea 1
  - nr\_caracter numarul caracterului pe linie și începe cu valoarea 0
- La depasirea valorii indexului nu se va genera o eroare ci se va considera ultima valoare a liniei/caracterului

## Exemplu de înscriere, citire și ștergere a conținutului Text

```
from tkinter import *
root=Tk()
root.geometry('500x200')
def inscriere():
    text.insert('1.0',mesaj)
def afisare():
    continut = text.get('2.2',END)
    print(continut)
def stergere():
    text.delete('1.1',END)
text = Text(root, width=60, height=7)
text.pack()
buton_inscriere = Button(root, text="Inscrie", command = inscriere).pack()
buton_afisare = Button(root, text="Afisare", command = afisare).pack()
buton_stergere = Button(root, text="Stergere", command = stergere).pack()
mesaj = '''-Universitatea tehnica a Moldovei
-Facultatea Electronica si Telecomunicatii
-Departamentul Telecomunicatii si Sisteme Electronice'''
root.mainloop()
```



## 2. Componenta - ScrollBar

### Crearea și plasarea componentei - Text

- Se utilizează pentru a permite utilizatorului să gestioneze componentele cu dimensiuni mai mari decât dimensiunile setate
- Se creează utilizând clasa Text()  
`scrollbar = Scrollbar(root)`
- Plasarea componentei este în funcție de componenta pe care o va gestiona

```
text.pack(side = LEFT)
scrollbar.pack(side = RIGHT, fill = Y)
```

```
text.grid(row=0, column=0)
scrollbar.grid(row=0, column=1, sticky=NS)
```

## Parametrii ai componentei ScrollBar

- **orient** – setează orientarea scrollbar-ului: VERTICAL, HORIZONTAL

```
scrollbar_x = Scrollbar(root, orient=HORIZONTAL)
scrollbar_y = Scrollbar(root, orient=VERTICAL)
```

- **command** – permite apelarea parametrului yview sau xview a componentei ce își va deplasa conținutul la modificare poziției scrollbar-ului

```
scrollbar_y.config(command = text.yview)
scrollbar_x.config(command = text.xview)
```

- Pentru ca scrollbar-ul să funcționeze este necesară setarea parametrilor yscrollcommand și xscrollcommand a componentei careia i se atasează

```
text.config(yscrollcommand = scrollbar_y.set)
text.config(xscrollcommand = scrollbar_x.set)
```

## Exemplu se setarea scrollbar pentru componenta Text

```
from tkinter import *  
root=Tk()  
  
scrollbar_y = Scrollbar(root, orient=VERTICAL)  
scrollbar_x = Scrollbar(root, orient=HORIZONTAL)  
  
text = Text(root, yscrollcommand = scrollbar_y.set,  
             xscrollcommand = scrollbar_x.set,  
             height=6, width=32, font="Tahoma 20 italic bold")  
  
text.grid(row=0, column=0)  
scrollbar_y.grid(row=0, column=1, sticky=NS)  
scrollbar_x.grid(row=1, column=0, sticky=EW)  
  
scrollbar_y.config(command = text.yview)  
scrollbar_x.config(command = text.xview)  
  
root.mainloop()
```

### 3. Componenta - ComboBox

#### Crearea și plasarea componentei - ComboBox

- Se utilizează pentru a permite utilizatorului să selecteze o opțiune dintr-o lista de opțiuni
- Pentru crearea componentei Combobox este necesar importat explicit a submodului ttk din modulul tkinter

```
from tkinter import ttk
```

- Se creează utilizând clasa Combobox() a submodulului ttk care trebuie importat explicit

```
combobox = ttk.Combobox(root)
```

- Plasarea componentei

```
combobox.pack()
```

- Pentru a utiliza rezultatul selectiei componentei este necesară salvarea acestuia prin intermediul unei variabile speciale de tip IntVar(), DoubleVar() sau StringVar().

## Parametrii componentei ComboBox

- **values** – permite setarea listei de elemente ce vor fi incluse in ComboBox

```
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']  
combobox = ttk.Combobox(root, values=zilele_saptaminii)
```

- **textvariable** – definește variabila specială (ex. BStringVar()) căreia i se atribuie valoarea selectata în ComboBox

```
var = StringVar()  
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']  
combobox = ttk.Combobox(root, values=zilele_saptaminii, textvariable=var)
```

## Setarea valorii curente în ComboBox

- prin intermediul metodei `set()` – permite orice valoare
  - cu ajutorul variabilei `textvariable`

```
var = StringVar()  
var.set("Alege ziua preferata")  
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']  
combobox = ttk.ComboBox(root, values=zilele_saptaminii, textvariable=var)  
- cu ajutorul variabilei combobox  
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']  
combobox = ttk.ComboBox(root, values=zilele_saptaminii, textvariable=var)  
combobox.set('Alege ziua preferata')
```

- prin intermediul metodei `current()` – permite fixarea indexului valorii din lista
  - doar cu ajutorul variabilei `combobox`

```
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']  
combobox = ttk.ComboBox(root, values=zilele_saptaminii, textvariable=var)  
combobox.set(0)
```

## Evenimentul <<ComboBoxSelected>> și citirea valorii selectate

- Pentru citirea valorii selectate se utilizeaza metoda get()
  - cu ajutorul variabilei textvariable  
`var.get()`
  - cu ajutorul variabilei combobox  
`combobox.get()`
- Evenimentul <<ComboBoxSelected>> permite apelarea unei funcții de manipulare a acestuia când se selectează o valoare din ComboBox

```
def printare(event):  
    print(combobox.get())
```

```
zilele_saptaminii = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri',  
                    'Sambata', 'Duminica']
```

```
combobox = ttk.ComboBox(root, values=zilele_saptaminii)  
combobox.set('Selectati ziua preferata')  
combobox.bind('<<ComboBoxSelected>>', printare)  
combobox.pack()  
root.mainloop()
```

## 4. Ferestre - MessageBox

### Crearea și apelarea ferestrelor- MessageBox

- Se utilizează pentru afișare pe ecran a unor mesaje pentru utilizatorul aplicatiei
- Cele mai utilizate ferestre MessageBox :
  - pentru mesaje de informare: funcția **showinfo()**
  - pentru mesaje de avertizare: funcția **showwarning()**
  - Pentru mesaje de eroare: **showerror()**
  - Pentru o întrebare da/nu: funcția **askyesno()**
- Pentru crearea componentei MessageBox este necesar de importat explicit a submodului messageboxt din modulul tkinter

```
from tkinter import messagebox
```

- Se creează apelând funcția corespunzătoare a submodulului messagebox

```
messagebox.functie(titlul, mesaj)
```



## MessageBox de informare

- Apelarea ferestrei de informare

```
messagebox.showinfo('Titlul ferestrei', 'Mesajul afisat in fereastra')
```

- Exemplu de utilizare

```
from tkinter import *
from tkinter import messagebox
root=Tk()
root.geometry('300x200')
def divizare():
    num1=int(entry1.get())
    num2=int(entry2.get())
    entry3.insert(0, num1/num2)
    messagebox.showinfo('Mesaj de informare', 'Operatie finisata cu succes')
label1=Label(root, text='Introduceti primul numar').pack()
entry1=Entry(root)
entry1.pack()
label2=Label(root, text='Introduceti al doilea numar').pack()
entry2=Entry(root)
entry2.pack()
buton=Button(root, text='Divizare numere', command=divizare).pack()
label3=Label(root, text='Rezultatul difizarii').pack()
entry3=Entry(root)
entry3.pack()
root.mainloop()
```

## MessageBox de avertizare

- Apelarea ferestrei de avertizare

```
messagebox.showwarning('Titlul ferestrei', 'Mesajul afisat in fereastra')
```

- Exemplu de utilizare

```
def divizare():  
    num1 = int(entry1.get())  
    num2 = int(entry2.get())  
    entry3.insert(0, round(num1/num2, 2))  
    messagebox.showwarning('Mesaj de avertizare', 'Rezultatul este rotungit la 2  
cifre dupa virgula')
```

```
label1=Label(root, text='Introduceti primul numar').pack()  
entry1=Entry(root)  
entry1.pack()  
label2=Label(root, text='Introduceti al doilea numar').pack()  
entry2=Entry(root)  
entry2.pack()  
buton=Button(root, text='Divizare numere', command=divizare).pack()  
label3=Label(root, text='Rezultatul difizarii').pack()  
entry3=Entry(root)  
entry3.pack()
```

## MessageBox de eroare

- Apelarea ferestrei de eroare

```
messagebox.showerror('Titlul ferestrei', 'Mesajul afisat in fereastra')
```

- Exemplu de utilizare

```
def divizare():  
    try:  
        num1 = int(entry1.get())  
        num2 = int(entry2.get())  
        entry3.insert(0, num1/num2)  
    except:  
        messagebox.showerror('Mesaj de eroare', 'Ati introdus date gresite')
```

```
label1=Label(root, text='Introduceti primul numar').pack()  
entry1=Entry(root)  
entry1.pack()  
label2=Label(root, text='Introduceti al doilea numar').pack()  
entry2=Entry(root)  
entry2.pack()  
buton=Button(root, text='Divizare numere', command=divizare).pack()  
label3=Label(root, text='Rezultatul difizarii').pack()  
entry3=Entry(root)  
entry3.pack()
```

## MessageBox de întebare da/nu

- Apelarea ferestrei de intrebare da/nu

```
variabila = messagebox.askyesno('Titlul ferestrei', 'Intrebarea din fereastra')
```

- În funcție de raspuns funcția askyesno() returneaza True sau False
- Exemplu de utilizare

```
def divizare():  
    info = messagebox.askyesno("Mesaj intrebare", "Esti sigur ca vrei sa divizezi 2 numere")  
    if info==True:  
        num1 = int(entry1.get())  
        num2 = int(entry2.get())  
        entry3.insert(0, num1/num2)  
    else:  
        return  
label1=Label(root, text='Introduceti primul numar').pack()  
entry1=Entry(root)  
entry1.pack()  
label2=Label(root, text='Introduceti al doilea numar').pack()  
entry2=Entry(root)  
entry2.pack()  
buton=Button(root, text='Divizare numere', command=divizare).pack()  
label3=Label(root, text='Rezultatul difizarii').pack()  
entry3=Entry(root)  
entry3.pack()
```