



# **Tema 18.**

# **Plasarea componentelor în fereastra Tkinter**

Ce ne așteaptă?

1. Metoda `pack()`
2. Metoda `place()`
3. Metoda `grid()`

1. Care sunt etapele de plasarea a componentelor pe fereastră?
2. Care sunt caracteristicile metodei `pack()`?
3. Care sunt rolurile parametrilor `side`, `extend` și `fill` a metodei `pack()`?
4. Ce rol au parametrii `padx` și `pady` a metodei `pack()`?
5. Care sunt caracteristicile metodei `place()`?
6. Cum se realizează setează absolută a poziției și dimensiunilor?
7. Cum se realizează setează relativă a poziției și dimensiunilor?
8. Care sunt caracteristicile metodei `grid()`?
9. Ce rol au parametrilor `row` și `column` a metodei `grid()`?
10. Ce rol are parametrul `sticky` a metodei `grid()`?
11. Ce rol au parametrii `rowspan` și `columnspan` a metodei `grid()`?
12. Ce rol au parametrii `ipadx` și `ipady` a metodei `grid()`?
13. Ce rol au parametrii `padx` și `pady` a metodei `grid()`?

# 1. Metoda *pack()*

## Crearea și plasare componentelor

- **Layout Manager** – controlul aspectului – este un mecanism de poziționare geometrică a componentelor într-un cadru

- **Etapele de plasare a componentelor**

- Crearea și parametrizarea componentei

*nume\_variabila = Clasa\_Componentă(componenta\_parinte, parametri)*

*label = Label(root, text='Salutare tuturor')*

- Plasarea componentei pe fereastra părinte

*nume\_variabila.metodă\_plasare(parametri)*

*label.pack()*

- **Metode de plasare a componentelor în Tkinter**

- Metoda *pack()*
- Metoda *place()*
- Metoda *grid()*

## Caracteristicile metodei *pack()*

- Metoda *pack()* – plasează componentele într-un cadru prin specificarea relație dintre pozițiile acestora și nu prin specificarea locație precise
- Metoda *pack()* este ușor de utilizat însă nu asigură o precizie buna
- Poziția componentelor depinde de ordinea plasării lor în cadru
- Sintaxa metodei *pack()*:

*nume\_componenta.pack(opțiuni)*

- Exemplu

```
buton_rosu = Button(root, text="Rosu", bg="red ", fg="white")
buton_rosu.pack()
buton_verde = Button(root, text="Verde", bg="green ", fg="white")
buton_verde.pack()
buton_albastru = Button(root, text="Albastru", bg="blue ", fg="white")
buton_albastru.pack()
```

## Parametru *side* a metodei *pack()*

- *side* – specifică latura componentei părinte față de care se va poziționa componenta copil  
valori acceptate: LEFT, RIGHT, TOP (implicit), BOTTOM
- Plasarea componentelor una sub alta de sus în jos în ordinea declarării lor (plasare implicită)

```
buton_rosu = Button(root, text="Rosu", bg="red", fg="white")
buton_rosu.pack(side=TOP)
buton_verde = Button(root, text="Verde", bg="green ", fg="white")
buton_verde.pack(side=TOP)
buton_albastru = Button(root, text="Albastru", bg="blue ", fg="white")
buton_albastru.pack(side=TOP)
```

- Plasarea componentelor una după alta de la stânga la dreapta în ordinea declarării lor

```
buton_rosu = Button(root, text="Rosu", bg="red ", fg="white")
buton_rosu.pack(side=LEFT)
buton_verde = Button(root, text="Verde", bg="green ", fg="white")
buton_verde.pack(side=LEFT)
buton_albastru = Button(root, text="Albastru", bg="blue ", fg="white")
buton_albastru.pack(side=LEFT)
```

## Exemplu de aplicare a parametrului side a metodei pack()

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_galben = Button(root, text="Galben", bg="yellow", fg='black')
buton_portocaliu = Button(root, text="Portocaliu", bg="orange", fg='black')
buton_violet = Button(root, text="Violet", bg="purple", fg='white')
buton_marou = Button(root, text="Marou", bg="brown", fg='white')
buton_negru = Button(root, text="Negru", bg="black", fg='white')
```

```
buton_rosu.pack(side=TOP)
buton_verde.pack(side=RIGHT)
buton_albastru.pack(side=BOTTOM)
buton_galben.pack(side=LEFT)
buton_portocaliu.pack(side=TOP)
buton_violet.pack(side=RIGHT)
buton_marou.pack(side=BOTTOM)
buton_negru.pack(side=LEFT)
```

## Parametru fill a metodei pack()

- fill – extinde componenta pe întreg spațiu neoculat de aceasta în funcție de opțiunea side valori acceptate: NONE (implicit), X, Y, BOTH
- Extinderea componentelor când sunt plase una sub alta de sus în jos

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_galben = Button(root, text="Galben", bg="yellow", fg='black')
buton_rosu.pack()
buton_verde.pack(fill = X)
buton_albastru.pack(fill = Y)
buton_galben.pack(fill = BOTH)
```

- Extinderea componentelor când sunt plase una după alta de stânga la dreapta

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_galben = Button(root, text="Galben", bg="yellow", fg='black')
buton_rosu.pack(side=LEFT)
buton_verde.pack(side=LEFT, fill = X)
buton_albastru.pack(side=LEFT, fill = Y)
buton_galben.pack(side=LEFT, fill = BOTH)
```



## Parametru *expand* a metodei *pack()*

- **expand** – extinde componenta pe întreg spațiu care ar putea ocupat de alte eventuale componente  
valori acceptate: False (implicit), True
- Extinderea componentelor pe întreg spațiu când sunt plase una sub alta de sus în jos

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_galben = Button(root, text="Galben", bg="yellow", fg='black')
buton_rosu.pack(expand = True)
buton_verde.pack(fill = X, expand = True)
buton_albastru.pack(fill = Y, expand = True)
buton_galben.pack(fill = BOTH, expand = True)
```

- Extinderea componentelor pe întreg spațiu când sunt plase una după alta de stânga la dreapta

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_galben = Button(root, text="Galben", bg="yellow", fg='black')
buton_rosu.pack(side=LEFT, expand = True)
buton_verde.pack(side=LEFT, fill = X, expand = True)
buton_albastru.pack(side=LEFT, fill = Y, expand = True)
buton_galben.pack(side=LEFT, fill = BOTH, expand = True)
```

## Exemplu combinațiilor opțiunilor fill și expand

- Un buton plasat implicit cu metoda `pack()`

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')  
buton_rosu.pack()
```

- Un buton plasat cu opțiunea `extend=True`

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')  
buton_rosu.pack(extend = True)
```

- Un buton plasat cu opțiunea `extend=True` și opțiune `fill=BOTH`

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')  
buton_rosu.pack(extend = True, fill=BOTH)
```

- Două butoane plasate cu opțiunea `extend=True` și opțiune `fill=BOTH`

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')  
buton_verde = Button(root, text="Verde", bg="green", fg='white')  
buton_rosu.pack(extend = True, fill=BOTH)  
buton_verde.pack(extend = True, fill=BOTH)
```

## Parametrii *padx* și *pady* a metodei *pack()*

- *padx* – stabilește un spațiu în stânga și dreapta hotarului componentei (în pixeli)
- *pady* – stabilește un spațiu deasupra și sub hotarul componentei (în pixeli)
- Exemplu de aplicare

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')

buton_rosu.pack(expand=True, fill=BOTH, pady=10, padx=10)
buton_verde.pack(expand=True, fill=BOTH, pady=10, padx=10)
buton_albastru.pack(expand=True, fill=BOTH, pady=10, padx=10)
```

## 2. Metoda *place()*

### Caracteristicile metodei *place()*

- Metoda *place()* – permite setarea explicită a poziției și a dimensiunilor componentele în termeni absoluți sau relativi față de componenta părinte
- Metoda *place()* asigură o precizie buna însă nu reflectă relația dintre componentele de pe aceeași fereastră
- Neajunsul: modificarea setărilor unei componente implică modificări și la celelalte componente
- Sintaxa metodei *place()*:

*nume\_componenta.place(opțiuni)*

- Exemplu

```
buton_rosu = Button(root, text="Rosu", bg="red", fg="white")
buton_rosu.place(x=0, y=0)
buton_verde = Button(root, text="Verde", bg="green", fg="white")
buton_verde.place(x=300, y=50)
buton_albastru = Button(root, text="Albastru", bg="blue", fg="white")
buton_albastru.place(x=200, y=300)
```

## Setarea absolută a poziției și dimensiunilor

- **Setarea absolută** – specificarea valorii numărului de inch față de colțul stânga sus a componentei părinte (punctul cu coordonatele 0,0)

- **x și y** – coordonatele colțului stânga sus a componentei față de punctul 0,0

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_rosu.place(x=0,y=0)
buton_verde.place(x=50, y=50)
buton_albastru.place(x=200, y=300)
```

- **width și height** – setarea valorilor absolute pentru lățimea și înălțimea componentei

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_rosu.place(x=0,y=0, width=50, height=50)
buton_verde.place(x=50, y=50, width=100, height=100)
buton_albastru.place(x=150, y=150, width=200, height=200)
```

## Setarea relativă a poziției și dimensiunilor

- **Setarea relativă** – specificarea valorii numărului de inch față de colțul stânga sus a componentei părinte (punctul cu coordonatele 0,0)
- **relx și rely** – valorile relative a coordonatelor componentei considerând dimensiunile ferestrei

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_rosu.place(relx=0.1, rely=0.2)
buton_verde.place(relx=0.3, rely=0.5)
buton_albastru.place(relx=0.5, rely=0.7)
```

- **relwidth și relheight** – dimensiunile relative a componentei considerând dimensiunile ferestrei

```
buton_rosu = Button(root, text="Rosu", bg="red", fg='white')
buton_verde = Button(root, text="Verde", bg="green", fg='white')
buton_albastru = Button(root, text="Albastru", bg="blue", fg='white')
buton_rosu.place(relx=0.1, rely=0.2, relwidth=0.2, relheight=0.2)
buton_verde.place(relx=0.3, rely=0.4, relwidth=0.3, relheight=0.3)
buton_albastru.place(relx=0.6, rely=0.7, relwidth=0.4, relheight=0.3)
```

### 3. Metoda *grid()*

#### Caracteristicile metodei *grid()*

- Metoda *grid()* – consideră fereastra părinte drept un tabel și permite plasarea componentelor prin specificarea coordonatei celulei tabelului în care se va afla aceasta
- Sintaxa metodei *grid()*:

*nume\_componenta.grid(opțiuni)*

- Exemplu

```
from tkinter import *
root = Tk()
root.geometry('135x80')
buton_rosu = Button(root, text="Rosu", bg="red", fg="white")
buton_rosu.grid(row=0, column=0)
buton_verde = Button(root, text="Verde", bg="green", fg="white")
buton_verde.grid(row=1, column=1)
buton_albastru = Button(root, text="Albastru", bg="blue", fg="white")
buton_albastru.grid(row=2, column=2)
root.mainloop()
```

## Parametrii row și column a metodei *grid()*

- **row** – numărul liniei pe care va fi amplasată componenta (prima linii are row=0)
- **column** – numărul coloanei pe care va fi amplasată componenta (prima coloana are column=0)
- **Exemplu**

```
label1= Label(root, text="Textul label-ului nr1")
label2= Label(root, text="Textul label-ului nr2")
buton1= Button(root, text="Buton nr1")
buton2= Button(root, text="Buton nr2")
intrare1= Entry(root)
intrare2= Entry(root)
```

```
label1.grid(row=0, column=0)
label2.grid(row=1, column=0)
intrare1.grid(row=0, column=1)
intrare2.grid(row=1, column=1)
buton1.grid(row=0, column=2)
buton2.grid(row=1, column=2)
```



## Parametru *sticky* a metodei *grid()*

- **sticky** – fixează poziția elementului în celulă atunci când celula este mai mare decât elementul
- **sticky** poate lua valorile N, E, S, W, NE, SE, SW, NW, NS, EW, NSEW
- Exemplu

```
label= Label(root, text="Textul labelul-ui", bg="yellow")
buton1= Button(root, text="Buton cu latime mare", bg="red", fg='white')
buton2= Button(root, text="Buton cu inaltime mare", height=3, bg="green",
fg='white')
buton3= Button(root, text="Buton", bg="blue", fg='white')

label.grid(row=0, column=0)
buton1.grid(row=0, column=1)
buton2.grid(row=1, column=0)
buton3.grid(row=1, column=1, sticky=NSEW)
```

## Parametrii *rowspan* și *columnspan* a metodei *grid()*

- **rowspan** – numărul de linii ocupate de componenta (implicit **rowspan=1**)
- **columnspan** – numărul de coloane ocupate de componenta (implicit **columnspan=1**)
- **Exemplu**

```
label1= Label(root, text="Textul label-ului nr1")
label2= Label(root, text="Textul label-ului nr2")
buton1= Button(root, text="Buton nr1")
buton2= Button(root, text="Buton nr2")
intrare1= Entry(root)
intrare2= Entry(root)

label1.grid(row=0, column=0)
label2.grid(row=1, column=0)
intrare1.grid(row=0, column=1)
intrare2.grid(row=1, column=1)
buton1.grid(row=0, column=2, rowspan=2, sticky=NSEW)
buton2.grid(row=2, column=0, columnspan=3, sticky=NSEW)
```

## Parametrii *ipadx* și *ipady* a metodei *grid()*

- *ipadx* – mărește dimensiunea componentei pe orizontală (în inch)
- *ipady* – mărește dimensiunea componentei pe verticală (în inch)
- Exemplu

```
buton1= Button(root, text="Buton nr1")  
buton2= Button(root, text="Buton nr2")  
buton3= Button(root, text="Buton nr3")  
buton4= Button(root, text="Buton nr4")  
buton5= Button(root, text="Buton nr5")  
buton6= Button(root, text="Buton nr6")
```

```
buton1.grid(row=0, column=0)  
buton2.grid(row=0, column=1)  
buton3.grid(row=0, column=2)  
buton4.grid(row=1, column=0)  
buton5.grid(row=1, column=1, ipadx=10, ipady=10)  
buton6.grid(row=1, column=2)
```

## Parametrii *padx* și *pady* a metodei *grid()*

- *padx* – adaugă un spațiu suplimentar exterior componentei pe orizontală (în inch)
- *pady* – adaugă un spațiu suplimentar exterior componentei pe verticală (în inch)
- Exemplu

```
buton1= Button(root, text="Buton nr1")  
buton2= Button(root, text="Buton nr2")  
buton3= Button(root, text="Buton nr3")  
buton4= Button(root, text="Buton nr4")  
buton5= Button(root, text="Buton nr5")  
buton6= Button(root, text="Buton nr6")
```

```
buton1.grid(row=0, column=0)  
buton2.grid(row=0, column=1)  
buton3.grid(row=0, column=2)  
buton4.grid(row=1, column=0)  
buton5.grid(row=1, column=1, padx=10, pady=10)  
buton6.grid(row=1, column=2)
```

## Exemplu de plasarea a componentelor Label, Button și Entry (1)

```
from tkinter import *
root = Tk()
root.geometry('455x220')

def inscriere():
    text_nume=in_nume.get()
    text_prenume=in_prenume.get()
    text_specialitate=in_specialitate.get()
    text_grupa=in_grupa.get()

    if text_nume!="" and text_prenume!="" and text_specialitate!="" and
    text_grupa!="":
        data=f"{text_nume} {text_prenume} {text_specialitate} {text_grupa} \n"
        with open("Registru.txt", "a") as f:
            f.write(data)
        print(data)
        info_confirm.config(text="Inregistrare reusita")
        info_confirm.config(fg="green")
```

## Exemplu de plasarea a componentelor Label, Button și Entry (2)

```
        in_nume.delete(0, END)
        in_prenume.delete(0, END)
        in_specialitate.delete(0, END)
        in_grupa.delete(0, END)
    else:
        info_confirm.config(text="Nu ati introdus toate datele")
        info_confirm.config(fg="red")
    info_confirm.grid(row=6, column=0, columnspan=2, padx=5, pady=5)

titlu = Label(root, text="Aplicatie de inscriere a studentilor", font="Aria 16
bold")
nume=Label(root, text="Numele studentului:")
prenume=Label(root, text="Prenumele studentului:")
specialitate=Label(root, text="Specialitatea:")
grupa=Label(root, text="Grupa:")

in_nume=Entry(root, width=50)
in_prenume=Entry(root, width=50)
in_specialitate=Entry(root, width=50)
in_grupa=Entry(root, width=50)
```

## Exemplu de plasarea a componentelor Label, Button și Entry (3)

```
buton = Button(root, text="Inscriere", command = inscriere)

info_confirm=Label(root)

titlu.grid(row=0, column=0, columnspan=2)
nume.grid(row=1, column=0, padx=5, pady=5, sticky=W)
prenume.grid(row=2, column=0, padx=5, pady=5, sticky=W)
specialitate.grid(row=3, column=0, padx=5, pady=5, sticky=W)
grupa.grid(row=4, column=0, padx=5, pady=5, sticky=W)

in_nume.grid(row=1, column=1, padx=5, pady=5)
in_prenume.grid(row=2, column=1, padx=5, pady=5)
in_specialitate.grid(row=3, column=1, padx=5, pady=5)
in_grupa.grid(row=4, column=1, padx=5, pady=5)
buton.grid(row=5, column=0, columnspan=2, sticky=NSEW, padx=5, pady=5)

root.mainloop()
```