

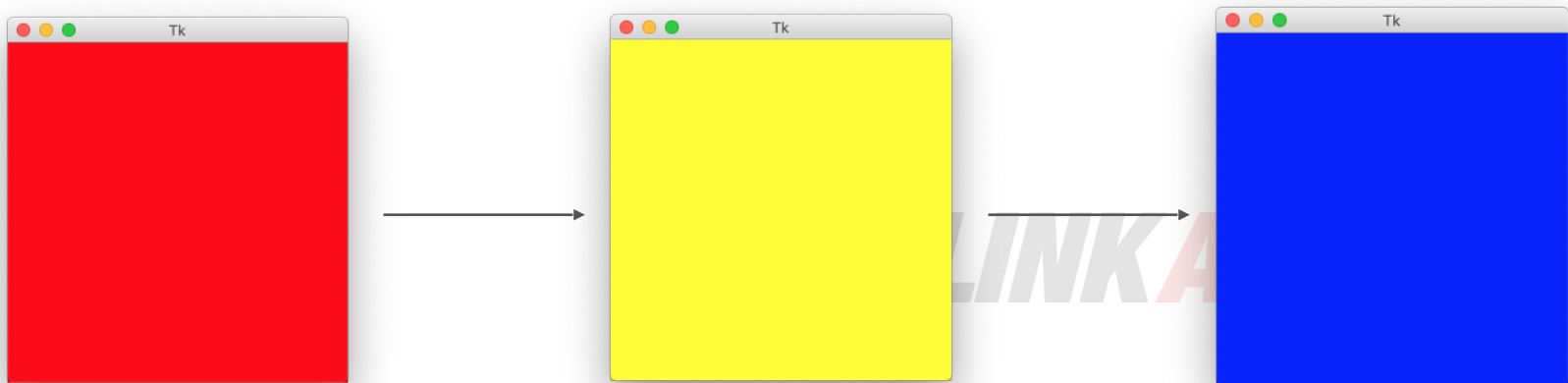
# Graphic Applications Development - 4

Silviu Ojog

***LINKAcademy***

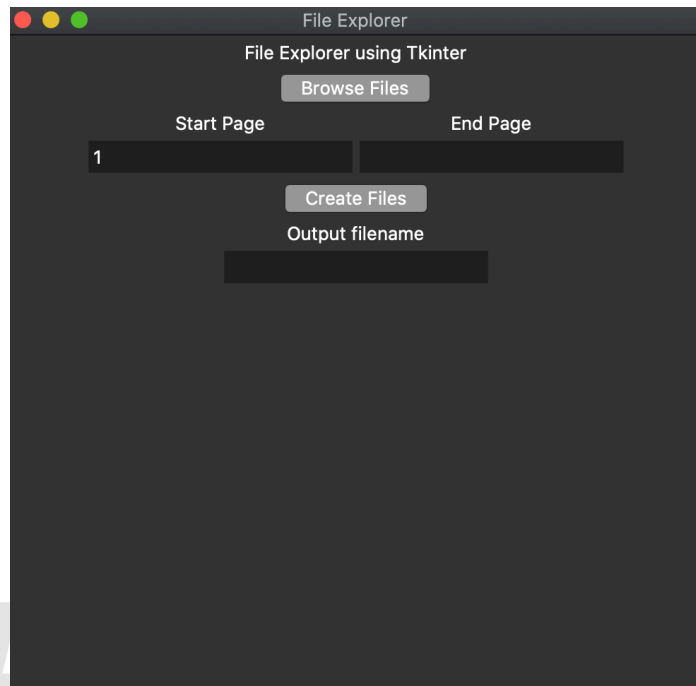
# Exercițiu Switch Frames

- Trebuie să creai trei controale de cadre cu culori de fundal diferite.
- În primul rând, se deschide o fereastră cu un cadru pe întreaga suprafață. Dând clic pe cadru, acesta este înlocuit cu un alt cadru, dând clic pe al doilea, în locul său apare al treilea și dând clic pe al treilea cadru, apare din nou primul.



# Exercitiu

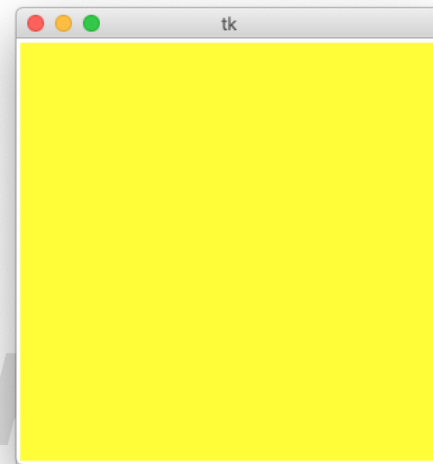
- Creați o aplicație tkinter care va primi un fișier PDF de pe disc și îl va împărți în alte fisiere



# Canvas

- Controlul Canvas asigură desenarea arbitrară.
- `c = tk.Canvas(window, width = 300, height=300, bg="yellow")`
- `c.pack()`

```
c = Canvas(win,  
width=300,  
height=300,  
bg="yellow")  
c.pack()
```

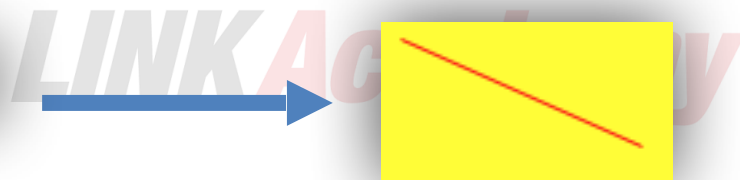


# Desenat in Canvas

- Canvas conține metode pentru desenarea formelor standard (elipsă, dreptunghi, linie...).
- Toate metodele de desenare au prefixul **create**.

```
c.create  
★ create_rectangle  
★ create_bitmap  
★ create_image  
create_arc  
create_line  
create_oval  
create_polygon  
create_text  
create_window
```

```
c.create_line(10,10,100,50,fill="red")
```



# Parametrii metodelor de desenat

- Majoritatea parametrilor metodelor de desenat sunt similare pentru toate metodele.
- Toate formele conțin, de asemenea, attribute specifice.

```
c.create_rectangle(  
    20,20,100,40,  
    fill="red",  
    outline="green",  
    width="3")
```



# Parametrii metodelor de desenat

- Nu toate formele acceptă toate atributele.

```
c.create_line(  
    20,20,100,40,  
    fill="red",  
    outline="green",  
    width="3")
```



```
_tkinter.TclError: unknown option "-outline"
```

```
c.create_line(  
    20,20,100,40,  
    fill="red",  
    width="3")
```

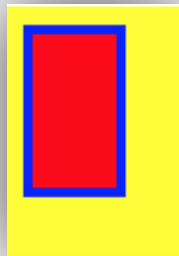


# Exemple de forme

```
c.create_line(  
5,5,30,50,  
fill="red",  
width="3")
```



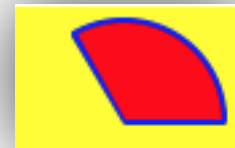
```
c.create_rectangle(  
10,10,50,80,  
fill="red",  
width=4,  
outline="blue"  
)
```



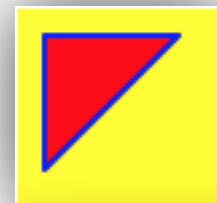
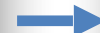
```
c.create_oval(  
10,10,80,60,  
fill="pink",  
width=4,  
outline="red"  
)
```



```
c.create_arc(  
5,5,80,80,  
extent=120,  
width=2,  
fill="red",  
outline="blue",  
style='pieslice')
```



```
c.create_polygon(  
[10,10,60,10,10,60],  
fill="red",  
outline="blue",  
width=2  
)
```



```
img=PhotoImage(file="suv.png")  
c.create_image(  
10,10,  
image=img,  
anchor="nw"  
)
```

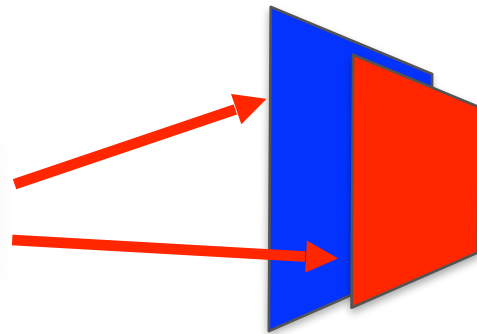




# Handlere

- Tkinter canvas desenează în straturi și, prin urmare, poate efectua operații izolate pe formele desenate.

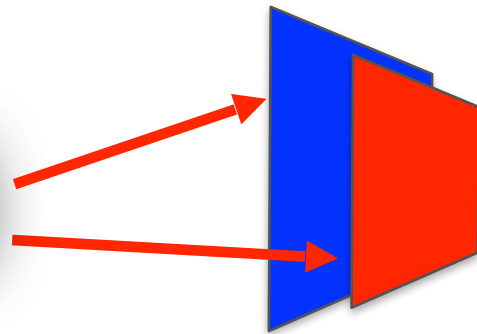
```
blue_rect = c.create_rectangle(5,5,100,50,fill="blue")  
red_rect = c.create_rectangle(50,20,150,80,fill="red")
```



# Taguri

- Tagurile sunt modalitati de selectie a componentelor din canvas.

```
blue_rect = c.create_rectangle(5,5,100,50,fill="blue")  
red_rect = c.create_rectangle(50,20,150,80,fill="red")
```



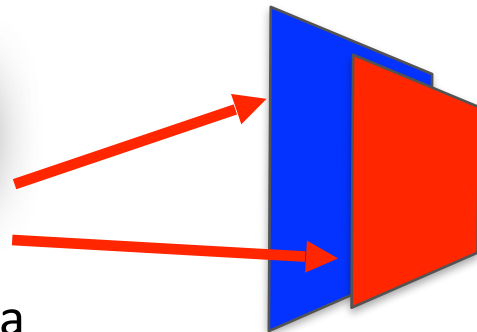
```
blue_rect = c.create_rectangle(  
5,5,100,50,fill="blue",  
tags="br"  
)
```

- Tag explicit

# Operatii forme

- Odată ce avem o formă de tag sau handler, putem să efectuăm diverse operații pe ea.

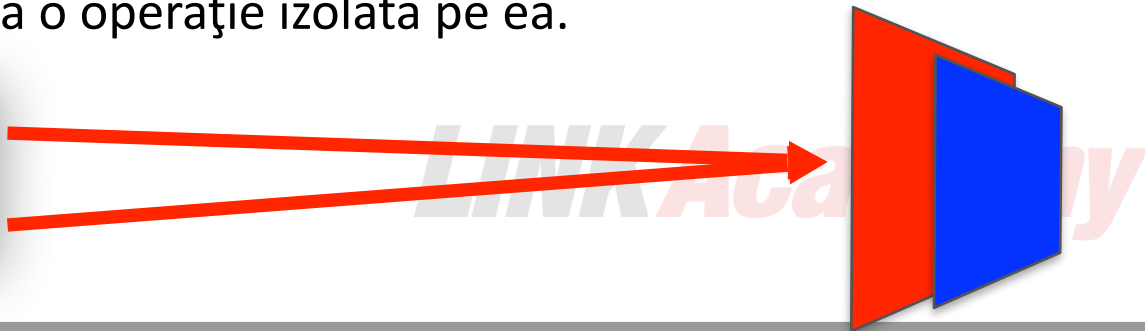
```
blue_rect = c.create_rectangle(5,5,100,50,fill="blue")  
red_rect = c.create_rectangle(50,20,150,80,fill="red")
```



- Folosind un handler sau un tag, putem obține forma dorită și putem efectua o operație izolată pe ea.

```
c.tag_raise(blue_rect)
```

```
c.tag_raise("br")
```



# Operatii forme

- Mutarea pozitiei pe axele X si Y

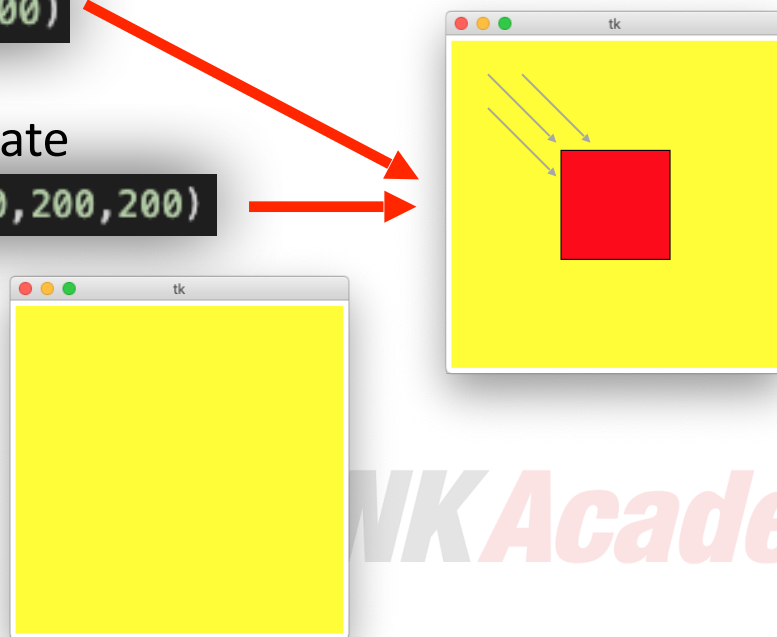
```
c.moveto(red_rect,100,100)
```

- Asignarea unor coordonate

```
c.coords(red_rect,100,100,200,200)
```

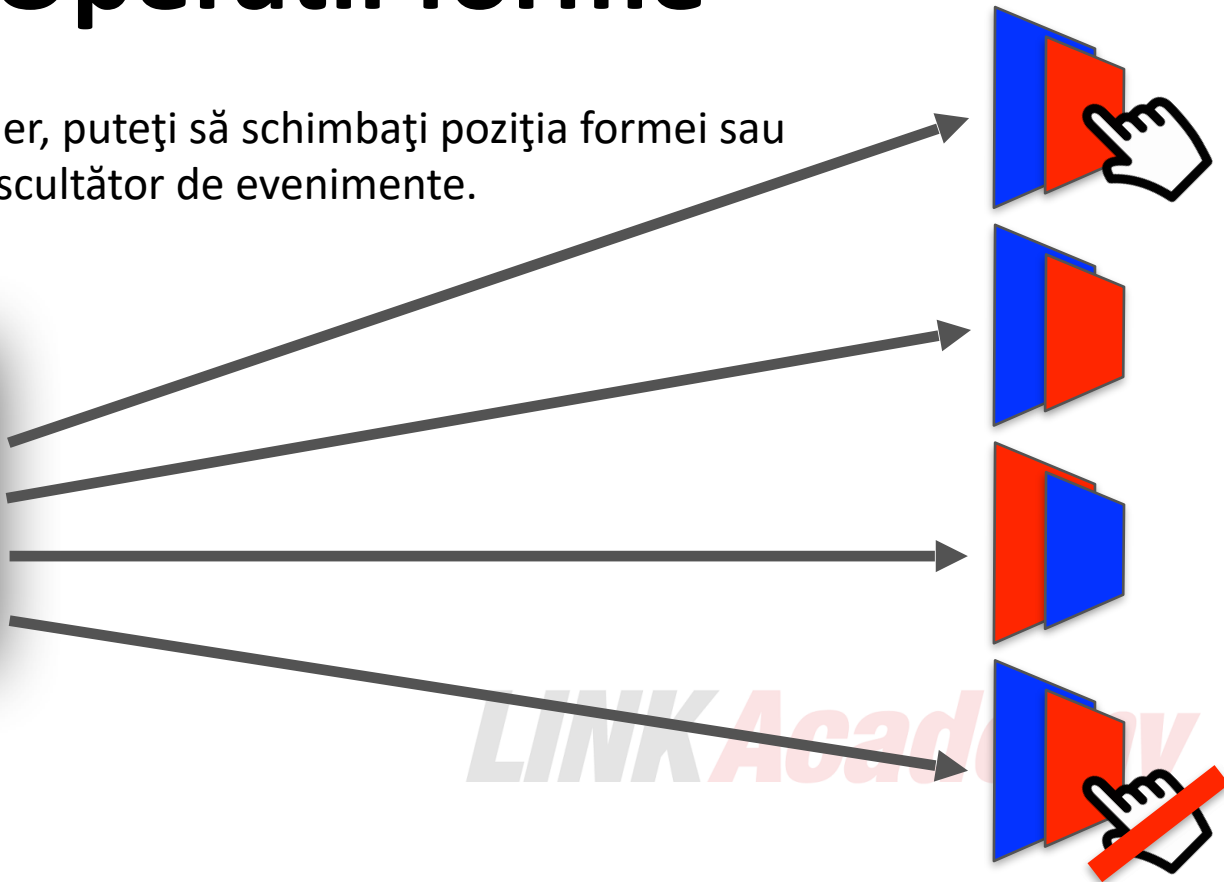
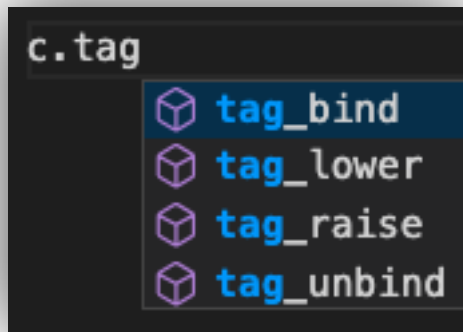
- Stergerea formei din canvas

```
c.delete(red_rect)
```



# Operatii forme

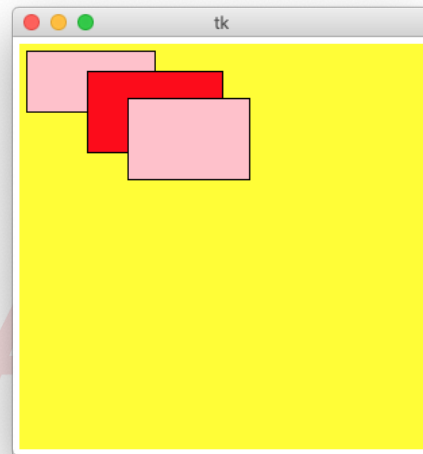
- Cu un tag sau un handler, puteți să schimbați poziția formei sau chiar să-i atribuiți un ascultător de evenimente.



# Gruparea tagurilor

- O formă poate avea mai multe taguri, iar aceste taguri nu sunt unice.
- Acest lucru este potrivit pentru gruparea operațiilor.

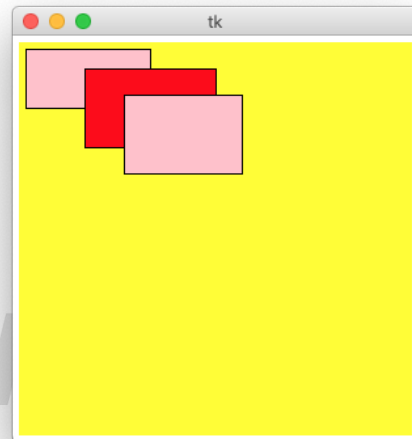
```
blue_rect = c.create_rectangle(5,5,100,50,fill="blue",tags=("b","g"))
red_rect  = c.create_rectangle(50,20,150,80,fill="red")
green_rect = c.create_rectangle(80,40,170,100,fill="green",tags="g")
c.itemconfig("g",fill="pink")
```



# Gruparea tagurilor

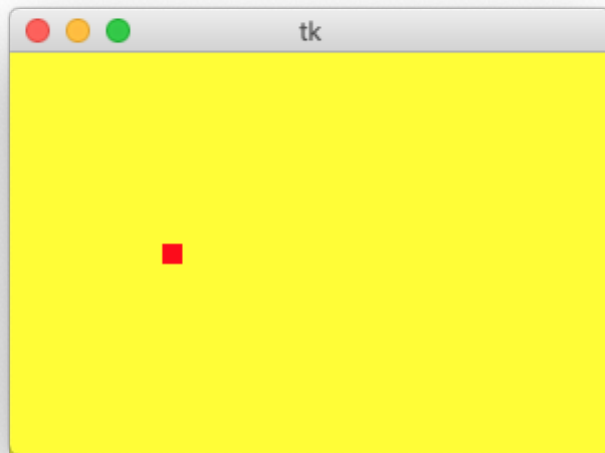
- Pe lângă gruparea manuală, pot fi utilizate și taguri rezervate.
- Tagul **all** reprezintă toate formele de pe pânză/Canvas.
- Tagul **current** reprezintă formularul aflat în prezent sub cursor.

```
def color_element(evt):  
    print(evt)  
    c.itemconfig("all",fill="pink")  
    c.itemconfig("current",fill="red")  
  
blue_rect = c.create_rectangle(5,5,100,50,fill="blue")  
red_rect = c.create_rectangle(50,20,150,80,fill="red")  
green_rect = c.create_rectangle(80,40,170,100,fill="green")  
  
c.bind("<Motion>",color_element)
```



# Animatia

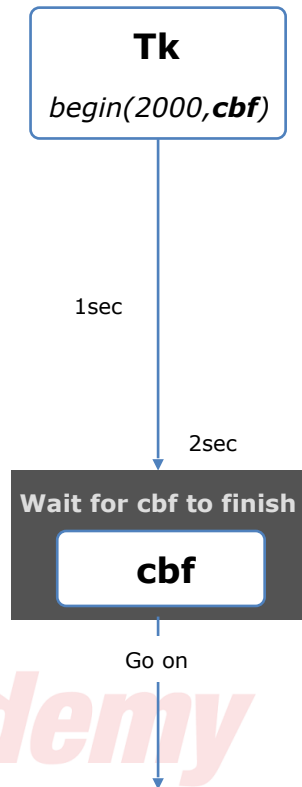
- Animația poate fi realizată în două moduri.
  - Utilizând apelări consecutive ale metodei **after** pe unul sau mai multe elemente;
  - Folosind propria implementare a **buclei principale** a aplicației.





# Animatia cu functia "after"

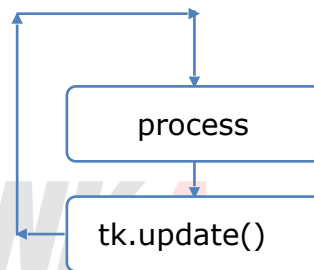
- După pornirea metodei mainloop pe orice element, aplicația tkinter devine blocată.
- Fiecare control tkinter are capacitatea de a accepta o funcție callback care va fi inserată în bucla principală și executată după o anumită perioadă.
- În acest scop, se utilizează metoda **after**.
- Metoda after, acceptă timpul de așteptare în milisecunde și funcția de apelare care trebuie
- `win.after(2,update)`
- Așteptarea începerii funcției callback este asincronă și **nu blochează** execuția, însă funcția propriu-zisă callback o **blochează**.



# Animatia cu functia "after"

- La această tehnică, bucla principală este realizată manual, iar randarea, precum și activarea tuturor celorlalte evenimente de sistem sunt, de asemenea, activate manual, prin metoda **update**.

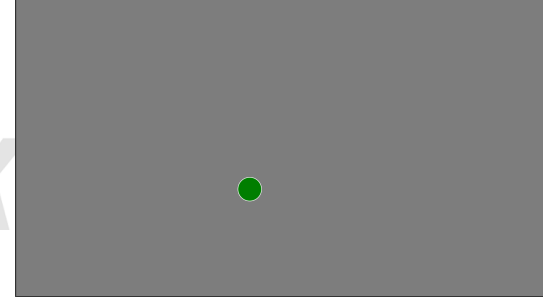
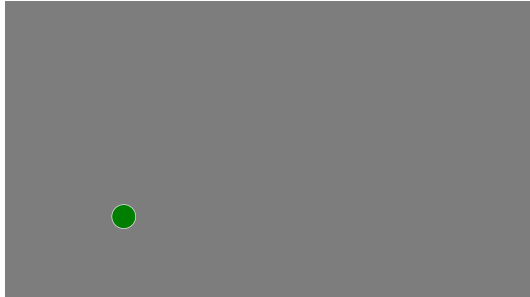
```
while True:  
    win.update()  
    time.sleep(0.002)
```



# Exercitiul 1

- Moving Ball
- Creați un exemplu asemanator celui din HTML
  - [https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D\\_Breakout\\_game\\_pure\\_JavaScript/Bounce\\_off\\_the\\_walls](https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript/Bounce_off_the_walls)

# Exercitiul 1



# Exercitiul 2

- Extindeti exercitiul anterior pentru a adauga o paleta

