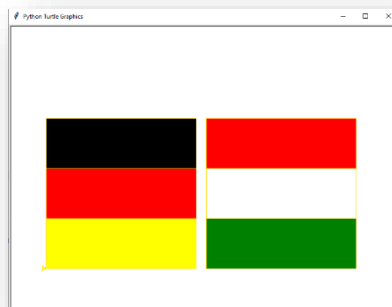


(13f.py)

Rajzoljunk egy más mellé, egy magyar és egy német zászlót!

Alakítsuk át az előző programot:

- Úgy, hogy egy újabb eljárásban – melynek a neve legyen tricolor - a három sávos zászlót is megrajzoljuk.
- A teglalap() eljárásnak előbbre kell lennie, mit a tricolor() eljárásnak, mert a tricolorban felhasználjuk a teglalap eljárást.
- A tricolor nevű eljárásnak legyen három átadandó értéke a f(első), k(középső), és a(Isó)!
- A tricolor() eljárásban megrajzoljuk, pozícionáljuk és színezzük a három téglalapot.
- A főprogramban törlés után megadjuk a rajzolószínt, ami ebben az esetben „arany”.
- Majd meghívjuk a tricolor(f,k,a) eljárást, ahol a változók helyére megadjuk a színek nevét angolul, idézőjelek közé írva.
- Majd felemeljük a „tollat” és úgy pozícionálunk jobbra, hogy a két zászló között 20 képpont legyen.



```

1 from turtle import *
2 def teglalap () :
3     begin_fill()
4     forward(300)
5     left(90)
6     forward(100)
7     left(90)
8     forward(300)
9     left(90)
10    forward(100)
11    end_fill()
12
13 def tricolor(f, k, a):
14    fillcolor(f)
15    teglalap()
16    up()
17    forward(100)
18    left(90)
19    down()
20    fillcolor(k)
21    teglalap()
22    up()
23    forward(100)
24    left(90)
25    down()
26    fillcolor(a)
27    teglalap()
28    left(90)
29
30 reset()
31 color("gold")
32 tricolor("red", "white", "green")
33 up()
34 goto(-320, 0)
35 down()
36 tricolor("black", "red", "yellow")
    
```

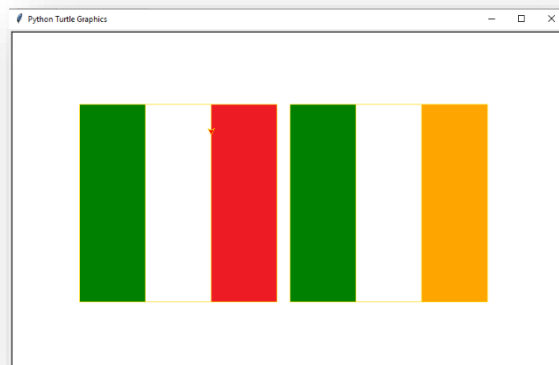
```

1 from turtle import *
2 def teglalap () :
3     begin_fill()
4     forward(100)
5     left(90)
6     forward(300)
7     left(90)
8     forward(100)
9     left(90)
10    forward(300)
11    end_fill()
12
13 def ftricolor(f, k, a):
14    fillcolor(f)
15    teglalap()
16    up()
17    left(90)
18    forward(100)
19    down()
20    fillcolor(k)
21    teglalap()
22    up()
23    left(90)
24    forward(100)
25    down()
26    fillcolor(a)
27    teglalap()
28    left(90)
29
30 reset()
31 color("gold")
32 goto(0, -150)
33 ftricolor("green", "white", "orange")
34 up()
35 goto(-320, -150)
36 down()
37 ftricolor("green", "white", "red")
    
```

(13g.py)

A következő feladatban függőlegesen három csíkos zászlót rajzoltatunk ki egymás mellé.

- A kész zászlók mérete legyen 300*300 képpontos!
- Ennél a feladatnál a logika ugyan az, mint az előzőekben, csak a téglalap méretét és irányát kell újra gondolni, meg a pozícionálásukat.
- Mindig figyeljünk arra, hogy mikor kell felemelni a „tollat”, mikor kell letenni!
- Hova, milyen irányba rajzoljuk meg a téglalapokat!



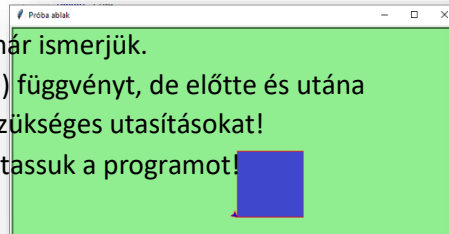
(13h.py)

Ebben a feladatban a grafikus megjelenítésre nézünk egy példát, és megnézzük pár új utasítást!



A feladat egy egyszerű téglalap kirajzoltatása!

- Hozzunk létre egy `negyzet()` nevű eljárást, mely megrajzol egy négyzetet!
- A `bgcolor(„lightgreen”)` parancs **háttérszínt ad az ablaknak!**
- A `title(„Próba ablak”)` utasítással, az **ablak címsorában megjelenő szöveget adhatunk meg!**
- A `color()` és `fillcolor()` utasítást pedig már ismerjük.
- A főprogramban meghívjuk a `negyzet()` függvényt, de előtte és utána megadjuk a zárt alakzat kitöltéséhez szükséges utasításokat!
- A kilépés előtt 3 másodpercet várakoztassuk a programot!



```

14e.py x
1 import time
2 from turtle import *
3 def negyzet(h):
4     for i in range(4):
5         forward(h)
6         left(90)
7
8 bgcolor("lightgreen")
9 title("Próba ablak")
10 color("red")
11 fillcolor("blue")
12
13 begin_fill()
14 negyzet(100)
15 end_fill()
16
17 time.sleep(3.0)
    
```

(13i.py)

Példányok (több teknőc egy programban)

Mint ahogy sok különböző egész változónk is lehet egy programban, úgy sok „teknőcünk” is lehet egyszerre. Mindegyik egy ún. **példány**. Minden egyes példánynak saját tulajdonságai vannak és saját metódusai – így a `tekno_1` rajzolhat egy vékony kék tollal egy bizonyos pozícióban, míg `tekno_2` haladhat a saját útján egy vastag piros tollal, stb... .

- Az ismert módon az első sor megmondja a Pythonnak, hogy töltsse be a „turtle” nevű modult. Amely lehetővé teszi a grafikus felület használatát.
- Majd a 2-4. sorig beállítjuk az ablak tulajdonságát!
- Majd létrehozuk 6-9 sorokban a `teki_1`, 11-14-ig a `teki_2`, és a 16-19-ig a `teki_3`-mat a tulajdonságaival! (rajzolószín, ceruza vastagság)
- Aztán definiálunk először egy `negyzet(a)` függvényt, amelyet pozícionálunk, felemelt ceruzával, majd megrajzolunk egy négyzetet, kitöltőszínnel!
- Így teszünk utána a `háromszög(a)` nevű függvénnyel.
- És a `kör(r)` függvénnyel is.
- A főprogramban csak meghívjuk a függvényeket a megadott értékekkel a minta szerint!
- Végül az utasítás, ami vár az ablak bezárására. (**`win.mainloop()`**)

```

14f.py x
1 import turtle
2 win = turtle.Screen()
3 win.bgcolor("gray")
4 win.title("Négyzet & Háromszög & Kör")
5
6 teki_1 = turtle.Turtle()
7 teki_1.color("blue")
8 teki_1.pensize(4)
9 teki_1.fillcolor("yellow")
10
11 teki_2 = turtle.Turtle()
12 teki_2.color("red")
13 teki_2.pensize(6)
14 teki_2.fillcolor("lightblue")
15
16 teki_3 = turtle.Turtle()
17 teki_3.color("green")
18 teki_3.pensize(8)
19 teki_3.fillcolor("lightpink")
    
```

```

20
21 def negyzet(a):
22     teki_1.up()
23     teki_1.goto(50, 50)
24     teki_1.down()
25     teki_1.begin_fill()
26     for i in range(4):
27         teki_1.forward(a)
28         teki_1.left(90)
29     teki_1.end_fill()
30
31 def háromszög(a):
32     teki_2.up()
33     teki_2.goto(-170, 50)
34     teki_2.down()
35     teki_2.begin_fill()
36     for i in range(3):
37         teki_2.forward(a)
38         teki_2.left(120)
39     teki_2.end_fill()
    
```

```

40
41 def kör(r):
42     teki_3.up()
43     teki_3.goto(0, -50)
44     teki_3.down()
45     teki_3.begin_fill()
46     for i in range(360):
47         teki_3.forward(r)
48         teki_3.left(r)
49     teki_3.end_fill()
50
51 negyzet(100)
52 háromszög(120)
53 kör(1)
54
55 win.mainloop()
    
```

