

----- KVADRATNA SPIRALA -----

```
import turtle as t, random as r
# dovrši program
# boja daje stalno jednu istu, zato kreiram funkciju
# boja = (r.randint(0, 255), r.randint(0, 255), r.randint(0, 255))
t.speed(0)
def randomBoja():
    return [r.randint(0,255), r.randint(0,255), r.randint(0,255)]
# pocetno rotiranje ulevo
t.left(90)
for i in range(5, 200, 5):
    t.color(randomBoja())
    t.forward(i)
    t.left(90)
```

--- KONCENTRICNI KRUGOVI ---

```
# bojimo pozadinu prozora u belo
prozor.fill(pg.Color("white"))

# centar kruga je u centru prozora
centar = (sirina//2, visina//2)

# poluprečnik se menja od 10 do 100, sa korakom 10
for r in range(10, 100, 10):
    # crtamo krug
    pg.draw.circle(prozor, pg.Color("red"), centar, r, 5)
```

--- KONCENTRICNI KRUGOVI ---

```
# bojimo pozadinu prozora u belo
prozor.fill(pg.Color("white"))

# centar kruga je u centru prozora
centar = (150, 150)

# poluprecnik se menja od 10 do 100, sa korakom 10
for r in range(10, 100, 10):
    # crtamo krug
    pg.draw.circle(prozor, pg.Color("red"), centar, r, 5)
```

--- PRAVOUGAONA MREZA ---

```
# bojimo pozadinu prozora u belo
```

```

prozor.fill(pg.Color("white"))

brojPodeoka = 10
dx = sirina / brojPodeoka
dy = visina / brojPodeoka      # izracunaj razmak izmedu podeoka po visini

# crtamo horizontalne linije
for i in range(1, brojPodeoka):
    pg.draw.line(prozor, pg.Color("blue"), (0, i*dy), (sirina, i*dy), 5)

# dodaj kod koji crta vertikalne linije crvenom bojom
for i in range(1, brojPodeoka):
    pg.draw.line(prozor, pg.Color("red"), (i*dx, 0), (i*dx, visina), 5)

```

--- SPARTANJE DOJAGONALAMA ---

```

# bojimo pozadinu prozora u belo
prozor.fill(pg.Color("white"))

# broj podeoka
n = 10
# prirastaj
dx = sirina / n
dy = visina / n

# crtamo n linija iznad sporedne dijagonale (ukljucujuci i nju)
for i in range(n + 1):
    pg.draw.line(prozor, pg.Color("black"), (0, i*dy), (i*dx, 0), 1)
# crtamo n-1 linija ispod sporedne dijagonale (bez nje)
for i in range(1, n):
    pg.draw.line(prozor, pg.Color("black"), (i*dx, visina), (sirina, i*dy), 1)

```

-----U BOJI SPARTANJE-----

```

prozor.fill(pg.Color("white"))

# broj podeoka
n = 10
# prirastaj
dx = sirina / n
dy = visina / n

# crtamo n linija iznad sporedne dijagonale (ukljucujuci i nju)
for i in range(n + 1):
    pg.draw.line(prozor, pg.Color("black"), (0, i*dy), (i*dx, 0), 1)

# crtamo n-1 linija ispod sporedne dijagonale (bez nje)
# dopuni ovde program

```

```
for i in range(1, n):
    pg.draw.line(prozor, pg.Color("black"), (i*dx, visina), (sirina, i*dy), 1)
```

```
# crtamo n linija ispod glavne dijagonale (ukljucujuci i nju)
```

```
# dopuni ovde program
```

```
for i in range(0, n):
    pg.draw.line(prozor, pg.Color("red"), (0, i*dy), ((n-i)*dx, visina), 1)
```

```
# crtamo n-1 linija iznad glavne dijagonale (bez nje)
```

```
# dopuni ovde program
```

```
for i in range(0, n):
    pg.draw.line(prozor, pg.Color("red"), (i*dx, 0), (sirina, (n-i)*dy), 1)
```

```
--- SARANJE ROTIRANIM DUZIMA ---
```

```
prozor.fill(pg.Color("white"))
```

```
# broj podeoka
```

```
n = 10
```

```
# priraštaj
```

```
dx = sirina / n
```

```
dy = visina / n
```

```
# crtamo n linija u donjem levom uglu prozora
```

```
for i in range(1, n):
    pg.draw.line(prozor, pg.Color("black"), (0, i*dy), (i*dx, visina), 1)
```

```
--- RANDOM STEPENIK ---
```

```
# funkcija koja gradi nasumicno odabranu boju
```

```
def nasumicnaBoja():
```

```
    return [random.randint(0,255), random.randint(0,255), random.randint(0,255)]
```

```
prozor.fill(pg.Color("white"))
```

```
brojStepenika = 10
```

```
sirinaStepenika = sirina/brojStepenika
```

```
visinaStepenika = visina/brojStepenika
```

```
for i in range(brojStepenika):
```

```
    pg.draw.rect(prozor, nasumicnaBoja(), (i*sirinaStepenika, visina - i*visinaStepenika, sirinaStepenika, i*visinaStepenika))
```

-----ISPREKIDANE LINIJE-----

```
# funkcija koja gradi nasumicno odabranu boju
def nasumicnaBoja():
    return [random.randint(0,255), random.randint(0,255), random.randint(0,255)]

# bojimo pozadinu u belo
prozor.fill(pg.Color("white"))

# dužina jednog podeoka je 30 piksela
podeok = 30

# sredina prozora
sredina = visina / 2

# dok god levi kraj linije ne ispadne van prozora
x = 0
while x < sirina:
    # crtamo linijicu
    pg.draw.line(prozor, nasumicnaBoja(), (x, sredina), (x + podeok, sredina), 5)
    # pomeramo se za dužinu linije i dužinu razmaka
    x += 2 * podeok
```

-----SAHOVSKA TABLA-----

```
# bojimo pozadinu prozora u belo
prozor.fill("white")

# dimenzije table i polja
brojPolja = 8
sirinaPolja = sirina / brojPolja
visinaPolja = visina / brojPolja

# prolazimo kroz sva polja
for i in range(brojPolja):
    for j in range(brojPolja):
        # bojimo crna polja
        if (i % 2 != 0 and j % 2 == 0) or (j % 2 != 0 and i % 2 == 0):
            pg.draw.rect(prozor, pg.Color("black"), (i*sirinaPolja, j*visinaPolja, sirinaPolja, visinaPolja))
```

-----NEKE JEDNOSTAVNE ANIMACIJE-----

-----pROMENA BOJE POZADINE NA NASUMICAN NACIN-----

```
pg.time.set_timer(pg.USEREVENT, 100)
```

```
kraj = False
```

```
treba_crtati = True
```

```
while not kraj:
```

```
    if treba_crtati:
```

```
        # postavljanje boje pozadine na nasumicnu boju
```

```
        prozor.fill(nasumicna_boja())
```

```
        # osvežavamo prikaz sadržaja ekrana
```

```
        pg.display.update()
```

```
        treba_crtati = False
```

```
    # cekamo na naredni dogadaj
```

```
    dogadjaj = pg.event.wait()
```

```
    # dogadaj iskljucivanja prozora
```

```
    if dogadjaj.type == pg.QUIT:
```

```
        kraj = True
```

```
    # otkucaj tajmera
```

```
    elif dogadjaj.type == pg.USEREVENT:
```

```
        treba_crtati = True
```