

\*\*\*\*\*

# Tema 7

\*\*\*\*\*

## TEME:

- Funkcije, opsti pojmovi
- Funkcije bez returna
- Funkcije sa returnom

## PRIMER 1: Definisanje funkcije za kvadrate

### POSTUPAK:

- Napisati prvo for naredbu i da zelimo da imamo kvadrat kao funkciju kojoj prosledjujemo sve bitne informacije
- Kvadrat funkcija treba da bude zaduzena za crtanje kvadrata

```
import turtle
import random

def kvadrat(x, y, boja, velicina):
    turtle.penup()
    turtle.setpos(x, y)
    turtle.color(boja)
    turtle.pendown()
    turtle.begin_fill()
    for i in range(4):
        turtle.forward(velicina)
        turtle.right(90)
    turtle.end_fill()
    turtle.penup()
```

```
kvadrat(50, -100, "orange", 45)
kvadrat(100, -10, "red", 50)
kvadrat(70, -60, "green", 60)
kvadrat(100, -30, "yellow", 20)
kvadrat(65, -15, "blue", 30)
```



### NAPOMENA:

- Razlikovati ulazne podatke, redosled prosledjivanja i definisanja
- Primetiti da ne moraju da se isto tako zovu
- Primetiti da definisanje funkcija obicno pisemo na vrhu, posle import naredbi
- Primetiti da se razlikuje posle funkcija deo koda koji zovemo glavni deo koda
- Razlikovati da u pozivu funkcije pisemo konkretne stvari koje su informacija toj funkciji sta treba tacno da koristi, u odnosu na samu definiciju funkcije, kada pisemo

apstraktne promenljive

## PRIMER: Crtanje obrnutih kvadrata od veceg ka manjem, upisani i obojeni

### POSTUPAK:

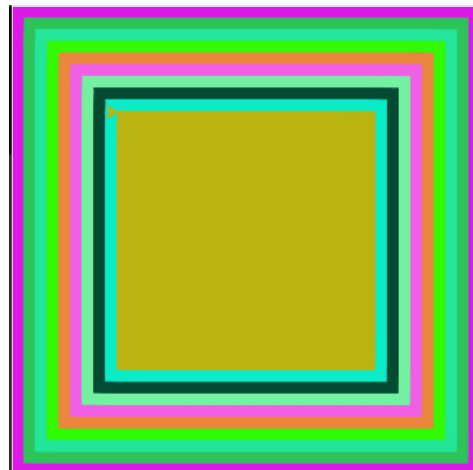
- Koristiti funkciju za kvadrat iz prethodnog primera
- Napraviti funkciju za boju

```
import turtle
import random

def kvadrat(x, y, boja, velicina):
    turtle.penup()
    turtle.setpos(x, y)
    turtle.color(boja)
    turtle.pendown()
    turtle.begin_fill()
    for i in range(4):
        turtle.forward(velicina)
        turtle.right(90)
    turtle.end_fill()
    turtle.penup()

def randomBoja():
    R = random.randint(0, 255)
    G = random.randint(0, 255)
    B = random.randint(0, 255)
    return R,G,B

stranica = 400
X = -200
Y = 200
razlikaKoordinata = 10
for i in range(10):
    noviX = X + i*razlikaKoordinata
    noviY = Y - i*razlikaKoordinata
    novaStranica = stranica - 2*i*razlikaKoordinata
    kvadrat(noviX, noviY, randomBoja(), novaStranica)
```



### NAPOMENA:

- Primiti return naredbu kod funkcija
- Individualno sami da eksperimentisu i prepoznaju kako napraviti jos kvadrata do sredine
- Napraviti vecu razliku izmedju kvadrata

## PRIMER 3 - BONUS : Iscrtavanje kvadrata u gornjem delu a trouglova u donjem delu prostora

### POSTUPAK:

- Sve isto kao ranije samo sto prilikom izbora X i Y moramo da napravimo dva razlicita slucaja. Za to koristimo naredbu grananja.
- Koristiti funkcije za kvadrat i random boju iz prethodnog primera

```
import turtle
import random

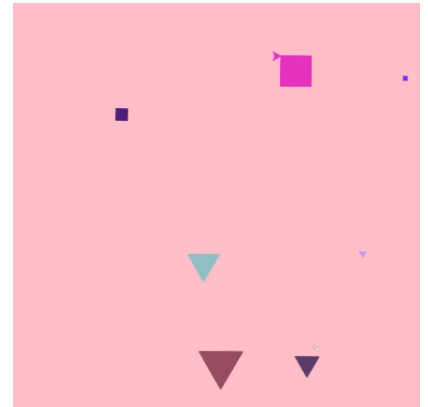
def kvadrat(x, y, boja, velicina):
    turtle.setpos(x, y)
    turtle.color(boja)
    turtle.pendown()
    turtle.begin_fill()
    for i in range(4):
        turtle.forward(velicina)
        turtle.right(90)
    turtle.end_fill()
    turtle.penup()

def randomBoja():
    R = random.randint(0, 255)
    G = random.randint(0, 255)
    B = random.randint(0, 255)
    return R,G,B

def trougao(x, y, boja, velicina):
    turtle.setpos(x, y)
    turtle.color(boja)
    turtle.pendown()
    turtle.begin_fill()
    for i in range(3):
        turtle.forward(velicina)
        turtle.right(120)
    turtle.end_fill()
    turtle.penup()

prostor = turtle.Screen()
prostor.setup(500,500)
prostor.bgcolor("pink")
turtle.penup()

for i in range (10):
    randomboja = (random.randint(0, 255), random.randint(0, 255),
random.randint(0, 255))
```



```
X = random.randint(-250, 250)
Y = random.randint(-250, 250)
stranica = random.randint(0, 50)
if Y >= 0:
    kvadrat(X, Y, randomboja, stranica)
else:
    trougao(X, Y, randomboja, stranica)
```

### **NAPOMENA:**

- Individualno da sami učenici naprave funkciju za trougao