

Python. Funktsiooni parameetrid

Sageli kasutatavad tegevused on otstarbekas vormistada funktsioonidena, mida saab kasutada erinevate algandmetega. Funktsiooni parameetrite hulk, tüübid ja järjekord tuleks valida nii, et funktsiooni oleks mugav kasutada.

Kõikidele parameetritele ei pea funktsiooni poole pöördumisel alati väärtusi andma. Sellised parameetrid asuvad funktsiooni kirjelduses parameetrite loetelu lõpus ja neile on määratud vaikeväärtus (väärtus, mis omistatakse parameetritele siis, kui sellele ei anta väärtust pöördumisel).

Näiteks:

```
from turtle import *
from random import randint

def kiir(x, y, pikkus, nurk, joon=0, värv='black'):
    '''joone pikkus, nurk, algpunkti x ja y, joone värv, joone paksus'''
    penup() #algul pliiats üles (igaks juhuks)
    pencolor(värv) #joone värv (vaikimisi must)
    if joon>0: pensize(joon) #kui joone paksust pole antud, jääb vana
    goto(x,y); pendown() #algasukoht, pliiats alla
    seth(nurk); fd(pikkus) #pööre antud suunas, joone joonistamine
    penup() #pliiats üles
```

Kasutades funktsiooni **kiir** joonistab järgnev skript kollaseid ühest punktist lähtuvaid erineva pikkusega jooni.

```
n = 20; a = 360/n
for i in range(n):
    nurk = i*a
    kiir(100, 100, randint(10,100), nurk, värv='yellow')
```

Veidi keerukam funktsioon **ellipsi** joonistamiseks

```
from turtle import *
from random import randint
from math import *

def ellips(laius, korgus, x=None, y=None, jv=None, tv=None, joon=None, T=True, n=36):
    '''laius, korgus,
    keskpunkti koordinaadid x ja y (kui väärtus andmata, võetakse kilpkonna asukoht)
    jv - joone värv, tv - täitevärv, joon - joone paksus,
    (kui väärtused andmata, jäävad endised väärtused)
    T - kas täita, n - punktide arv ellipsi joonistamisel'''
    penup()
    if x==None: x=xcor() # kui x andmata, võetakse selleks kilpkonna x-asukoht
    if y==None: y=ycor() # kui y andmata, võetakse selleks kilpkonna y-asukoht
    if jv: pencolor(jv) # värvide ja joone paksuse muutmine (kui vaja)
    if tv: fillcolor(tv)
    if joon: pensize(joon)
    a = laius/2; b = korgus/2
    goto(x+a,y); pendown()
    if T: begin_fill() # täitmise algus
    for i in range(n):
        nurk = 2*pi/n*(i+1) # 360/n * 2*pi/360
        x1 = a*cos(nurk)+x; y1 = b*sin(nurk)+y
        goto(x1,y1)
    if filling(): end_fill() # kui kujund täidetakse, tuleb see lõpetada siin
    penup()

ellips(200, 80, tv='dark blue') # ellipsite joonistamine erinevate algandmetega
ellips(150, 50, 0, 0, tv='blue', joon=3)
ellips(100, 30, 0, 0, jv='light blue', joon=3)
ellips(30, 30, randint(-100, 100), -50, joon=10, T=False)
ellips(30, 30, randint(-100, 100), -50, joon=10, T=False)
```

Ülesanne:

Koostada Pythonis funktsioonid kujundite joonistamiseks. Valida sobiv parameetrite hulk ja järjestus.

Kindlasti võiksid olla:

- **ristkülik**, parameetriteks alumise vasakpoolse nurga koordinaadid – x ja y – laius ja kõrgus ning joone laius, joone ja tausta värvid (vaikeväärtustega)
- **täidetud ring**, parameetriteks keskpunkti asukoht (x , y), raadius r , tunnus (kas täita) – vaikimisi väärtus True jm; omaloodud funktsioonis võiks ringi joonistamiseks kasutada mooduli turtle funktsiooni circle.
- **joon**, parameetriteks otspunktide koordinaadid, joone paksus ja värv
- **korrapärane hulknurk**

Ülesanne.

Koostada rakendus, milles kasutatakse loodud funktsioone mingi kompositsiooni moodustamiseks.

Kasutada kordus- ja valikulauseid, küsida andmeid kasutajalt, kasutada juhuarve jm.

Funktsioonid võivad asuda eraldi failis. Oma rakendusele saab need lisada käsuga **import**.