

Funkcije (podprogrami)

Python

Funkcije (podprogrami)

Podprogram je dio programskog koda koji se piše neovisno o glavnom programu. On se izvodi pozivanjem. (upisivanjem imena). Osnovno je obilježje podprograma da ih možemo jednom napisati a više puta pozvati na izvođenje. Potprograme u Pythonu zovemo funkcija.

Pisanje funkcije :

```
def ime_funkcije(popis parametara):  
    blok_naredbi  
    return
```

Funkcije (podprogrami) – načini rada

Četiri su načina rada funkcija u pythonu

1. Funkcija nema ulazne parametre i ne vraća vrijednost
2. Funkcija ima ulazne parametre i ne vraća vrijednost
3. Funkcija nema ulazne parametre i vraća vrijednost
4. Funkcija ima ulazne parametre i vraća vrijednost

*za funkcije koje ne vraćaju vrijednost nije nužno pisati naredbu return

```
def ime_funkcije(popis parametara):  
    blok_naredbi  
    return
```

Napomene:

Funkcije su neovisne od glavnog programa, ne izvršavaju se dok nisu pozvane

Funkciju pozivamo njenim imenom, ukoliko funkcija ima ulazne parametre moramo ih upisat

Funkcije možemo pozivat u glavnom programu proizvoljan broj puta

Ako funkcija vraća vrijednost u glavni program, možemo je spremiti u novu varijablu

Odabir načina pisanja funkcije ovisi o danom problemu

Korištenje funkcija omogućava smislenije strukturiranje programa

1.Funkcija nema ulazne parametre i ne vraća vrijednost u glavni program – zbroj dva unesena broja

#funkcija nema ulaznih parametara i ne vraća vrijednost

```
def zbroj():
```

```
    a=int(input('Upiši prvi broj'))  
    b=int(input('Upiši drugi broj'))  
    z=a+b  
    print('Zbroj je:',z)
```

```
zbroj()
```

Definiranje funkcije: nema ul. Parametre (prazne zagrade)

Unos, brojeva u a i b, obrada i ispis rezultata

Poziv funkcije u glavnom programu: Funkciju pozivamo njenim imenom, a zagrade ostavljamo prazne
Jer nismo definirali ulazne parametre

Kako funkcija nema ulazne parametre i
Ne vraća ništa u glavni program, sve je
Napisano i unutar funkcije (unos, obrada i ispis)
A u glavnom programu smo je samo pozvali

2.Funkcija ima ulazne parametre i ne vraća vrijednost u glavni program – zbroj dva unesena broja

#funkcija ima ulaznih parametara i ne vraća vrijednost

```
def zbroj(a,b):  
    z=a+b  
    print('Zbroj je:',z)  
zbroj(1,2)
```

Definiranje funkcije: Ima ulazne parametre, ovdje smo stavili dvije varijable, a i b, u koje će se spremati ulazne vrijednosti koje će funkcija zbrajati

Ispis rezultata je napisan unutar funkcije, jer ne vraća vrijednost u glavni program. Ako bi funkciju pozvali u glavni program bez vraćanja – naredbe return, u glavnom programu dobijemo ispis **none**

Poziv funkcije sa ulaznim vrijednostima: funkciju pozivamo i u zagradama navodimo one vrijednosti s kojima će funkcija zbrajati. 1 ulazi u parameter ili varijablu a, a 2 u varijablu b.

Napomena:

Ovdje nismo koristili naredbu input za unos podataka jer nam funkcija ima ulazne parametre, pa je input suvišan. One vrijednosti s kojima želimo da funkcija radi navodimo pri pozivu funkcije

3. Funkcija nema ulazne parametre i vraća vrijednost u glavni program – zbroj dva unesena broja

#funkcija nema ulaznih parametara i vraća vrijednost

```
def zbroj():  
    z=a+b  
    return z
```

Definiranje funkcije: nema ul. Parametre (prazne zagrade)

Vraćanje vrijednosti u glavni program: Funkcija rezultat zbroja
Sprema u varijablu *z* i vrijednost varijable *z* vraćamo u glavni program naredbom **return**

```
a=int(input('Upiši prvi broj'))  
b=int(input('Upiši drugi broj'))  
print('Zbroj je', zbroj())
```

Unos vrijednosti smo napisali u glavnom programu (input) i spremali
Smo ih u varijable **a** i **b** Kad funkcija nema ul. Parametre unos je obavezno napisati.

Drugi način:

```
a=int(input('Upiši prvi broj'))  
b=int(input('Upiši prvi broj'))  
c=zbroj()  
print('Zbroj je', c)
```

Pozivanje funkcije umetnuto u naredbu print. Ispisuje se tekst „Zbroj je”
A nakon toga rezultat rada funkcije

U glavnom programu smo mogli spremiti vraćenu vrijednost funkcije u novu varijablu pa nju ispisati

4. Funkcija ima ulazne parametre i vraća vrijednost u glavni program – zbroj dva unesena broja

#funkcija ima ulazne parametre i vraća vrijednost

```
def broj(a,b):  
    zbr=a+b  
    return zbr  
print(broj(5,5))
```

Definiranje funkcije: Ima ulazne parametre, ovdje smo stavili dvije varijable, *a* i *b*, u koje će se spremati ulazne vrijednosti koje će funkcija zbrajati

U glavni program vraćamo vrijednost varijable **zbr**

Ispis i poziv funkcije sa ulaznim vrijednostima **5 i 5**

Funkcije - zadaci

Python

Funkcija – Zadatak 1:

Napravi program koji traži unos dva broja, računa i ispisuje rezultat zbroja, razlike, umnoška i količnika unesenih brojeva. Za svaku računsku operaciju definirati zasebnu funkciju, a unos brojeva i ispis rezultata napisati u glavnom programu. Svaka funkcija neka vrati rezultat u glavni program gdje će se ispisati rezultat

Funkcije – Zadatak 2:

Napisati program koji će uz pomoć funkcije omogućit unos jednog broja. Broj se zatim vraća u glavni program gdje se ispisuje 5 puta

Funkcije – Zadatak 3:

Napiši program koji računa prosjek ocjena iz 5 predmeta po želji. Prosjek se računa sa funkcijom koja ima ulazne parametre i vraća izračun u glavni program

Funkcije – Zadatak 4:

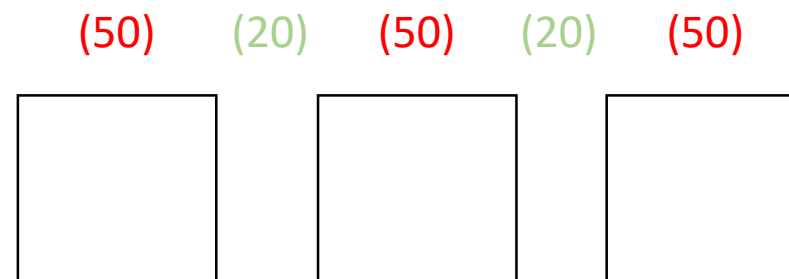
Napraviti program koji omogućava izbor izračuna površine ili opsega pravokutnika. Korisnik izabire radnju iz izbornika koji poziva funkciju. Funkcija Nema ulazne parametre i vraća vrijednost u glavni program

Funkcija – Zadatak 5:

Napravi program koji traži unos duljine stranice a . Program zatim računa i ispisuje površinu i opseg kvadrata. Program također crta kvadrat. Crtanje kvadrata napisati funkcijom.

Funkcije – Zadatak 6:

Napravi program koji crta zadane kvadrate. Definirati dvije funkcije, jednu koja crta kvadrat a drugu koja pomiče kornjaču kako bi stvorila razmak između kvadrata

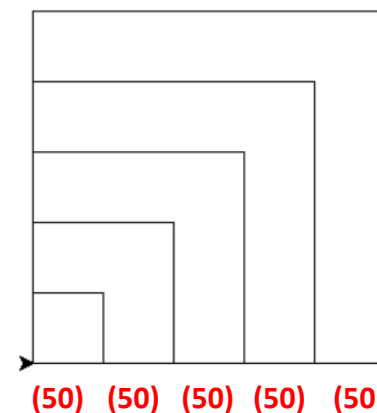


Funkcije – Zadatak 7:

Napiši program koji traži unos broja kuteva i duljinu stranice mnogokuta. Program zatim crta zadani mnogokut.

Funkcija – Zadatak 8:

Napiši program koji traži duljine stranice kvadrata. Program zatim crta 5 kvadrata kako je prikazano na slici. Duljina stranice najmanjeg je 50, a svaki idući kvadrat ima duljinu stranice za 50 veću od prethodnog. Crtanje kvadrata napiši pomoću jedne funkcije sa ulaznim parametrom a



Funkcija – Zadatak 1:

Napravi program koji traži unos dva broja, računa i ispisuje rezultat zbroja, razlike, umnoška i količnika unesenih brojeva. Za svaku računsku operaciju definirati zasebnu funkciju, a unos brojeva i ispis rezultata napisati u glavnom programu. Svaka funkcija neka vrati rezultat u glavni program gdje će se ispisati rezultat

```
def zbroj() :
```

```
    z=a+b
```

```
    return z
```

```
def umnozak() :
```

```
    u=a*b
```

```
    return u
```

```
def razlika() :
```

```
    r=a-b
```

```
    return r
```

```
def kolicnik() :
```

```
    k=a/b
```

```
    return k
```

Definirali smo 4 zasebne funkcije. Nemaju ulazne parametre i vraćaju vrijednosti u glavni program

```
a=int(input('upiši prvi broj:'))
```

```
b=int(input('upiši drugi broj:'))
```

Unos vrijednosti u glavnom programu

```
print('Zbroj je:', zbroj())
```

```
print('Razlika je:', razlika())
```

```
print('Količnik je:', kolicnik())
```

```
print('Umnožak je:', umnozak())
```

Ispis i poziv funkcija – ispisuje se rezultat rada svake pojedine funkcije:

zbroj()

razlika()

kolicnik()

umnozak()

Funkcije – Zadatak 2:

Napisati program koji će uz pomoć funkcije omogućit unos jednog broja. Broj se zatim vraća u glavni program gdje se ispisuje 5 puta

```
#napisat funkciju koja traži unos broja.  
#Broj se vraća u glavni program i ispisuje 5 puta
```

```
def broj():  
    n=int(input('Upiši broj'))  
    return n
```

```
br=broj()
```

```
for i in range (5):  
    print(br)
```

Definiranje funkcije: nema ul. parametre (prazne zagrade)

Unos broja (input) – napisan unutar funkcije

Vraćanje vrijednosti u glavni program

Spremanje **vraćene** vrijednosti u varijablu **br**
Varijabli **br** smo pridružili ono što je napravila funkcija **broj()**

For petljom ispisujemo vrijednost varijable **br**
5 puta.

Funkcija – Zadatak 3:

Napiši program koji računa prosjek ocjena iz 5 predmeta po želji. Prosjek se računa sa funkcijom koja ima ulazne parametre i vraća izračun u glavni program

```
def prosjek(h,m,l,g,k):  
    prosjek=(h+m+l+g+k)/5  
    return prosjek  
pr=prosjek(2,4,4,4,4)  
print(pr)
```

Definiranje funkcije: Ima ulazne parametre, ovdje smo stavili pet varijabli za ocjene iz pet predmeta: **h,m,l,g,k**

Izračun prosjeka: pazimo na točnost matematičkog izraza

Vraćanje **varijable prosjek** u glavni program

Pozivanje funkcije **prosjek** sa ulaznim vrijednostima i spremanje u varijablu **pr**

Ispis vrijednosti varijable **pr**

Primjer sa sata:

```
ivan=prosjek(2,3,5,5,3)  
ana=prosjek(3,4,3,4,5)  
ante=prosjek(4,5,5,5,2)
```

```
print(ivan)  
print(ana)  
print(ante)
```

Na satu smo ovaj zadatak napravili tako da smo funkciju pozvali više puta, za izračun prosjeka više učenika. Svaki put smo unijeli druge parametre i to spremili u varijable sa imenima učenika

Ispisali smo vrijednosti tih varijabli, koje su izračunati prosjeci sa zadanim ulaznim vrijednostima

Funkcija – Zadatak 4:

Napraviti program koji omogućava izbor izračuna površine ili opsega pravokutnika. Korisnik izabire radnju iz izbornika koji poziva funkciju. Funkcija Nema ulazne parametre i vraća vrijednost u glavni program

```
def površina() :  
    a=int(input('Duljina stranice a?'))  
    b=int(input('Duljina stranice b?'))  
    p=a*b  
    return p  
def opseg() :  
    a=int(input('Duljina stranice a?'))  
    b=int(input('Duljina stranice b?'))  
    o=2*a+2*b  
    return o
```

Definiranje funkcije **površina()**: nema ul. parametre (prazne zagrade)
Unos smo napisali naredbama input unutar funkcije

Vraćamo u glavni program vrijednost varijable **p**

Definiranje funkcije **opseg()**: nema ul. parametre (prazne zagrade)
Unos smo napisali naredbama input unutar funkcije

Vraćamo u glavni program vrijednost varijable **o**

```
print('Izaberi:')  
print('Izračun površine---1')  
print('Izračun opsega-----2')  
print('Kraj-----3')  
  
x=int(input('Izbor'))
```

U glavnom programu smo napisali izbornik naredbama print()
Varijabla **x** nam služi da korisnik unese traženi izbor

```
if x==1:  
    print(površina())  
elif x==2:  
    print(opseg())  
else:  
    print('Kraj')
```

Izbornik: Ovisno što korisnik utika (1, 2 ili 3) poziva se određena funkcija.
Koristili smo if elif else strukturu jer imamo tri mogućnosti (površina, opseg ili kraj)

Napomena:

Unutar strukture grananja program izvršava samo jednu radnju, onu koju korisnik izabere!

- Ako je izabran **1** poziva funkciju **površina()**
- Ako je izabran **2** poziva funkciju **opseg()**
- Ako je izabran **3** ispisuje se **kraj**

Funkcija – Zadatak 5:

Napravi program koji traži unos duljine stranice a . Program zatim računa i ispisuje površinu i opseg kvadrata. Program također crta kvadrat. Crtanje kvadrata napisati funkcijom.

```
from turtle import *
```

→ pozivamo modul kornjače za crtanje

```
def kvadrat():
```

→ Definiranje funkcije: **kvadrat()** nema ul. parametre (prazne zagrade)

```
    for i in range (4):
```

Funkcija for petljom crta kvadrat duljine stranice a . Ovu vrijednost unosimo u glavnom programu.

```
        fd(a)
```

Funkcija ne vraća ništa, već samo crta kvadrat.

```
        lt(90)
```

```
a=int(input('Unesi duljinu stranice a:'))
```

```
p=a*a
```

```
o=a*4
```

```
print('Površina kvadrata je:',p,'a opseg:',o)
```

→ Unutar glavnog programa unosimo duljinu stranice a
Računamo površinu i opseg i ispisujemo rezultat.

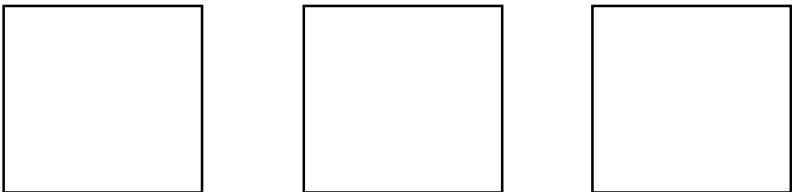
```
kvadrat()
```

→ Zadnjom naredbom pozivamo funkciju **kvadrat()** koja će ga nacrtati sa duljinom stranice a

Funkcija – Zadatak 6:

Napravi program koji crta zadane kvadrate. Definirati dvije funkcije, jednu koja crta kvadrat a drugu koja pomiče kornjaču kako bi stvorila razmak između kvadrata

(50) (20) (50) (20) (50)



***Napomena:** Zadana nam je duljina svakog kvadrata, i duljina razmaka među njima, pa nije potrebno unositi ništa, kao ni pisati funkciju sa ulaznim parametrima*

```
from turtle import*
def kvadrat():
    for i in range(4):
        fd(50)
        lt(90)
def razmak():
    pu();fd(70);pd()
for i in range(3):
    kvadrat()
    razmak()
```

pozivamo modul kornjače za crtanje

Definiranje funkcije: **kvadrat()** nema ul. parametre (prazne zagrade)
Funkcija for petljom crta kvadrat duljine stranice **50**.

Definiranje funkcije: **razmak()** nema ul. parametre (prazne zagrade)
Funkcija podiže kornjaču, pomiče je za 70 i spušta kornjaču.

For petljom 3 puta pozivamo funkcije. Prvo kvadrat(), pa razmak() i tako 3 puta za 3 kvadrata i 3 razmaka
Nama su potrebna samo 2 razmaka, ali jednostavnije je ovako napisati nego ovako:

kvadrat()
razmak()
kvadrat()
razmak()
kvadrat()
razmak()

Funkcija – Zadatak 7:

Napiši program koji traži unos broja kuteva i duljinu stranice mnogokuta. Program zatim crta zadani mnogokut.

```
from turtle import *
```

→ pozivamo modul kornjače za crtanje

```
def mnogokut():  
    for i in range(n):  
        fd(a)  
        lt(360/n)
```

→ Definiranje funkcije **mnogokut()** nema ul. parametre (prazne zagrade)
Funkcija for petljom crta kvadrat duljine stranice **a** sa kutom **360/n**
Petlja ponavlja radnje **n** puta

```
a=int(input('Duljina stranice:'))  
n=int(input('Broj kuteva:'))  
mnogokut()
```

→ U glavnom programu smo napisali unos duljine stranice i spremamo to u varijablu **a** te broj kuteva koji spremamo u varijablu **n**

→ Pozivamo funkciju koja onda radi sa unesenim vrijednostima **a** i **n**

Funkcija – Zadatak 7 – drugi način:

Napiši program koji traži unos broja kuteva i duljinu stranice mnogokuta. Program zatim crta zadani mnogokut.

Ovaj smo zadatak mogli napisati i pomoću funkcije koja ima ulazne parametre i ne vraća vrijednost. Kako zadatak traži unos duljine stranice i broja kuteva mnogokuta Mogli smo to definirati unutar same funkcije.

```
from turtle import*
```

```
def mnogokut(a,n):  
    for i in range(n):  
        fd(a)  
        lt(360/n)
```

```
mnogokut(100,6)
```

Definiranje funkcije **mnogokut(a,n)** Ovdje smo definirali ulazne parametre i to **a** za duljinu stranice **a** n za broj kuteva mnogokuta

Funkcija for petljom crta kvadrat duljine stranice **a** sa kutom **360/n**
Petlja ponavlja radnje **n** puta

Poziv funkcije sa ulaznim parametrima – ovo bi bio **šesterokut (n=6) koji ima duljinu stranice 100**

Ako bi htjeli nacrtati neke druge mnogokute funkciju bi pozvali ovako:

mnogokut(100,3) – trokut duljine stranice 100

mnogokut(50,4) – kvadrat duljine stranice 50

mnogokut(100,5) – peterokut duljine stranice 100

.....itd.

Funkcija – Zadatak 8:

Napiši program koji traži duljine stranice kvadrata. Program zatim crta 5 kvadrata kako je prikazano na slici. Duljina stranice najmanjeg je 50, a svaki idući kvadrat ima duljinu stranice za 50 veću od prethodnog. Crtanje kvadrata napiši pomoću jedne funkcije sa ulaznim parametrom a

Prvi način:

```
from turtle import*  
def kvadrat(a):
```

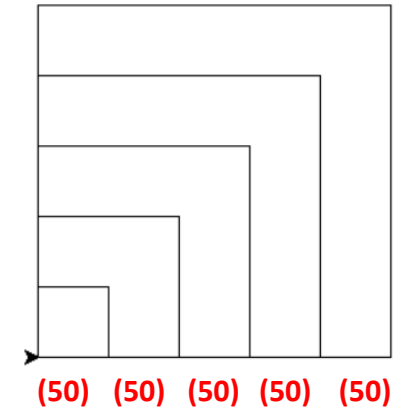
Poziv modula turtle i definiranje funkcije: **kvadrat(a)** sa jednim ulaznim parametrom i

```
    for i in range(4):  
        fd(a)  
        lt(90)
```

Funkcija crta kvadrat duljine stranice a

```
kvadrat(50)  
kvadrat(100)  
kvadrat(150)  
kvadrat(200)  
kvadrat(250)
```

Pozivanje funkcije sa drugačijim ulaznim parametrom



Drugi način korištenjem for petlje:

```
from turtle import*  
def kvadrat(a):  
    for i in range(4):  
        fd(a)  
        lt(90)
```

```
for i in range(6):  
    kvadrat(i*50)
```

Poziv funkcije for petljom 6 puta, sa ulaznim parametrom $i*50$, gdje se varijabla i povećava od 0 – 6 (6 puta se petlja ponavlja).

U prvom prolazu petlje parametar iznosi $0*50$,
u drugom $1*50$
trećem $2*50$,
četvrtom $3*50$,
petom $4*50$
šestom $5*50$