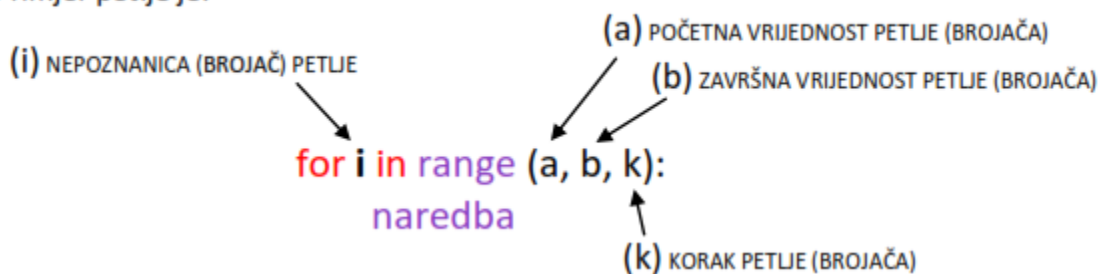


# PROGRAMSKA PETLJA

---

Programske petlje su programske strukture koji omogućavaju višestruko ponavljanje određenog dijela programskog koda. Petlja kao i programski konstrukt grananja ili odluke (if-else) sadržavaju uvjet. Ispunjenjem uvjeta određeni dio koda, izvršava se zadani broj puta. Osim toga petlja sadržava i brojač. Brojač je nešto poput uvjeta, gdje zadajemo broj ponavljanja nekog programskog koda. Ključna riječ za aktivaciju petlje je naredba **for**.

Primjer petlje je:



Uz nju dolazi oznaka **i** koja označava brojač. Brojač petlje zadaje broj ponavljanja određenog koda. Druga ključna riječ u petlji je **range** pomoću koje određujemo uvjet ponavljanja određenog koda. Unutar **range** nalazi se **početna vrijednost brojača** od koje petlja kreće ponavljati određeni kod, **zatim završna vrijednost petlje** te **korak petlje** kojim kažemo brojaču u kojme smjeru treba brojati i ponavljati petlju. Primjerice može se zadati ispis svakog drugog broja kojeg je korisnik unio. Ili se primjerice može zadati ispis negativnog broja. Završetak deklaracije označavamo sa znakom „:“ kao kod programskog konstrukta odluke.

### Primjer programa:

Napišite program kojim ćete koristeći FOR petlju ispisati vrijednost brojača od 1 do 5.

DIJAGRAM TIJEKA	OBJAŠNENJE
<pre>graph TD; A([POČETAK]) --&gt; B[i = 1 DO 6]; B --&gt; C[/ispiši i/]; C --&gt; D(( )); D --&gt; B; D --&gt; E([KRAJ]);</pre>	Određujemo početnu i završnu vrijednost brojača.
	Ispiši svaku trenutačnu vrijednost brojača.
	Ponavljaj radnju dok dođeš do kraja brojača.

Rješenje programa bi izgledalo ovako:

PROGRAM	REZULTAT
<pre>for i in range (1,6): print (i)</pre>	1 2 3 4 5

- Brojač smo u ovom primjeru označili varijablom **i**
- Odredili smo početnu i završnu vrijednost brojača.
- U zadnjem prolazu vrijednost brojača je uvijek za 1 manja.
- Radnja će se ponavljati sve dok brojač ne bude imao vrijednost 6
- U svakom prolazu ispisat će se vrijednost brojača, osim u zadnjem jer se petlja zaustavlja.

---

# ZADACI

---

## Zadatak1:

Ispišite ponavljanje rečenice „Unosimo policu broj“ deset puta.

## Primjer testiranja:

```
====  
(Unosimo policu broj', 1)  
(Unosimo policu broj', 2)  
(Unosimo policu broj', 3)  
(Unosimo policu broj', 4)  
(Unosimo policu broj', 5)  
(Unosimo policu broj', 6)  
(Unosimo policu broj', 7)  
(Unosimo policu broj', 8)  
(Unosimo policu broj', 9)  
(Unosimo policu broj', 10)
```

## Rješenje:

```
for i in range (1,11):  
    print ('Unosimo policu broj', i)
```

Na mjesto uvjeta u for petlji, odredili smo početnu (1) i završnu vrijednost (11).Petlja je ispisala 10 puta rečenicu „Unosimo policu broj“ te broj od 1 do 11 točnije prvih 10 brojeva. Vrijednost brojača unutar petlje mijenja se od 1 do 10 (vrijednost brojača je uvijek za jedan manja od završne vrijednosti brojača).Naredba za ispis unosa svakog pojedinačnog paketa nalazi se unutar petlje Da je primjerice stavljeno od 0 do 10, program bi ispisao brojeve: 0,1,2,3,4,5,6,7,8,9. Odnosno prvih 10 znamenki brojeći od nula.

Primjer:

```
('Unosimo policu broj', 0)
('Unosimo policu broj', 1)
('Unosimo policu broj', 2)
('Unosimo policu broj', 3)
('Unosimo policu broj', 4)
('Unosimo policu broj', 5)
('Unosimo policu broj', 6)
('Unosimo policu broj', 7)
('Unosimo policu broj', 8)
('Unosimo policu broj', 9)
```

### Zadatak2:

Ispisati samo parne brojeve brojevnog niza od prvih 10 brojeva.

### Primjer testiranja:

```
0
2
4
6
8
>>> |
```

### Rješenje:

```
for i in range (0,10,2):
    print (i)
|
```

Kod ovog rješenja, dodano je uz raspon početne i završne vrijednosti i broj koraka. Broj koraka ovdje je 2. To znači da će program ispisivati svaki drugi broj odnosno u ovom slučaju parne brojeve.

### Zadatak3:

Ispiši prvih šest brojeva u obrnutom nizu. Npr. 1,2,3,4,5,6 ispisati kao 6,5,4,3,2,1.

### Primjer testiranja:

```
6
5
4
3
2
1
>>> |
```

### Rješenje:

```
for i in range (6, 0, -1):
    print (i)
|
```

Da bi dobili ispis obrnutog niza brojeva, moralo se najprije postaviti početna vrijednost na broj 6 to jest na broj kojim završava niz. Zatim kao završnu vrijednost postaviti broj 1 ili broj s kojim niz započinje. A broj koraka se zatim postavi na -1 čime se pythonu zapovijeda da treba krenuti u suprotnom smjeru.

### Zadatak4:

Unesite prvih 10 brojeva, ali izuzmite broj 7.

### Primjer testiranja:

```
1
2
3
4
5
6
8
9
10
>>> |
```

### Rješenje:

```
for i in range (1,11):  
    if i != 7:  
        print (i)
```

Da bi omogućili ispisivanje brojeva u nizu, koristili smo for petlju sa zadanom početnom i završnom vrijednosti. Budući da smo ispisali niz brojeva bez broja 7 u njemu, bilo je potrebno kombinirati više programskih struktura poput odluka i petlje. Upravo je to napravljeno u ovom zadatku.

### Zadatak5:

Napišite program koji ispisuje zbroj parnih brojeva od 1 do 20

### Primjer testiranja:

```
('Nakon broja', 0, 'zbroj je', 0)  
( 'Nakon broja', 2, 'zbroj je', 2)  
( 'Nakon broja', 4, 'zbroj je', 6)  
( 'Nakon broja', 6, 'zbroj je', 12)  
( 'Nakon broja', 8, 'zbroj je', 20)  
( 'Nakon broja', 10, 'zbroj je', 30)  
( 'Nakon broja', 12, 'zbroj je', 42)  
( 'Nakon broja', 14, 'zbroj je', 56)  
( 'Nakon broja', 16, 'zbroj je', 72)  
( 'Nakon broja', 18, 'zbroj je', 90)  
( 'Nakon broja', 20, 'zbroj je', 110)
```

### Rješenje:

```
zbroj = 0  
for i in range (0, 21, 2):  
    zbroj = zbroj + i  
    print ('Nakon broja', i , 'zbroj je', zbroj)
```

U for petlji je određen raspon brojeva od 0-21. Za ispis parnih brojeva, određen je korak na broj 2 što znači da se ispisuje svaki parni broj to jest svaki drugi u nizu brojeva. Poslije svake nove vrijednosti brojača, zbroj se povećava za vrijednost brojača (**zbroj = zbroj+i**).

### Zadatak6:

Pomoću petlji istovremeno ispisati dvije riječi „Python“ i „je super“ tri puta.

### Primjer testiranja:

```
python
python
python
je super
je super
je super
```

### Rješenje:

---

```
for i in range (0,3):
    print ('python')
for j in range (0,3):
    print ('je super')
```

Paralelno su pisane dvije petlje s istim uvjetom. Budući da se program ispisuje slijedno od vrha prema dnu, prvo se ispisala riječ „python“, a zatim riječ „je super“.

### Zadatak7:

Unesi neki broj i ispiši sve brojeve do tog broja počevši od 1.

### Primjer testiranja:

```
unesi broj5
1
2
3
4
5
```

### Rješenje:

```
n = int(input('unesi broj'))
for i in range(1, n + 1):
    print(i)
```

U programu je omogućen unos vrijednosti preko tipkovnice. Brojevi se ispisuju od početne vrijednosti 1 do broja koji smo zadali unosom.

### Zadatak8:

Unesi neki broj. Napravi ispis svih parnih brojeva do broja koji ste unijeli.

### Primjer testiranja:

```
unesi broj9  
2  
4  
6  
8
```

### Rješenje:

```
n = int(input('unesi broj'))  
for i in range(2, n+1, 2):  
    print (i)
```

### Zadatak9:

Unesite neki broj i napravite ispis svih brojeva do tog broja u obrnutom redoslijedu.

### Primjer testiranja:

```
unesi broj:8  
8  
7  
6  
5  
4  
3  
2  
1
```

### Rješenje:

```
n = int(input('unesi broj:'))  
for i in range(n, 0, -1):  
    print (i)
```

Kao početna vrijednost u for petlji je stavljeno unos korisnika (n), zatim kao krajnja vrijednost je stavljena nula, a kao korak -1 čime je naznačeno da se petlja kreće u obrnutom redoslijedu.



### Zadatak10:

Omogućite proizvoljan unos brojeva. Zatim unesite te proizvoljne brojeve i ispitajte koliko ih ima parnih, a koliko neparnih.

### Primjer testiranja:

```
Unesi broj brojeva:2
Unesi broj:12
Unesi broj:3
('Zbroj parnih:', 12, 'Zbroj neparnih:', 3)
```

### Rješenje:

```
n = int (input('Unesi broj brojeva:'))
z_p = 0
z_n= 0
for i in range(n):
    t = int(input('Unesi broj:'))
    if t % 2:
        z_n = z_n+t
    else:
        z_p =z_p+t
print ('Zbroj parnih:',z_p,'Zbroj neparnih:', z_n)
```

Najprije je napravljen unos broja brojeva. To jest korisnika se pita koliko brojeva želi unijeti. Nakon toga omogućava mu se unos brojeva. Na primjer ako je napisao da će unijeti dva broja, onda mu je omogućeno da unosi samo dva broja. Nakon toga pomoću if-else strukture, ispituje se koji je broj paran, a koji neparan. Koristi se modularno dijeljenje s ostatkom `%`. Poslije svake nove vrijednosti brojača, zbroj se povećava za vrijednost brojača (`z_n=z_n+t`).