

# Naredba ponavljanja

Naredba **for** omogućuje ponavljanje jedne ili više naredbi unaprijed zadan konačan broj puta.

**For** petlja u Pythonu svakim prolaskom kroz petlju postavi kontrolnu varijablu na sljedeći element iz bilo kojega slijeda podataka (string, lista,...) počevši od prvoga pa sve do zadnjega elementa.

Kada **for** petlja dođe do zadnjega elementa iz slijeda, tada se njezino izvršavanje prekida.

Sintaksa naredbe

```
for i in slijed:  
    Blok naredbi;
```

Ako slijed sadrži izraz, onda se prvo evaluira taj izraz.

## Primjer 1.

Kontrolna varijabla *i* uzastopno se ponavlja (iterira se) po znakovima unutar stringa.

```
>>> for i in "Geodetska":  
        print ('Trenutno slovo je: ', i)  
  
Trenutno slovo je:  G  
Trenutno slovo je:  e  
Trenutno slovo je:  o  
Trenutno slovo je:  d  
Trenutno slovo je:  e  
Trenutno slovo je:  t  
Trenutno slovo je:  s  
Trenutno slovo je:  k  
Trenutno slovo je:  a  
>>> |
```

## Primjer 2.

Kontrolna varijabla **y** uzastopno se ponavlja (iterira se) po znakovima unutar stringa.

```
>>> for y in "GTŠ":
        print ('Trenutno slovo je: ', y)

Trenutno slovo je:  G
Trenutno slovo je:  T
Trenutno slovo je:  Š
>>> |
```

## Primjer 3.

Kontrolna varijabla iterira se po elementima definirane **liste**. Na ekran se ispisuju svi elementi iz liste.

### **Lista**

*Lista je promjenjiva struktura koja se sastoji od poredanog niza članova objekata.*

*Članovi u listi su bilo kakvi objekti različitih tipova.*

*Listu definiramo nabranjem članova odijeljenh zarezima i smještenih unutar uglatih zagrada ([ ]).*

```
>>> ekipa = ['Ivan', 'Marko', 'Jana', 'Lucija']
>>> for p in ekipa:
        print('ekipa su:',p)

ekipa su: Ivan
ekipa su: Marko
ekipa su: Jana
ekipa su: Lucija
>>> |
```

**Primjer 4.**

```

>>> posjednik = ['2158/5', 'Janko Dobrić', 'Gornja Bistra', 'Plavi put 67']
>>> for m in posjednik:
    print('Posjednik:',m)

Posjednik: 2158/5
Posjednik: Janko Dobrić
Posjednik: Gornja Bistra
Posjednik: Plavi put 67
>>> |

```

**Primjer 5.**

```

>>> posjednik = ['2158/5', 'Janko Dobrić', 'Gornja Bistra', 'Plavi put 67']
>>> for m in posjednik:
    print(m , end = ', ')

2158/5, Janko Dobrić, Gornja Bistra, Plavi put 67,
>>> |

```

**Funkcija range()**

Funkcija *range* kreira slijed brojeva.

**Primjer 6.**

```

>>> a = range (5)
>>> a
range(0, 5)
>>> type (a)
<class 'range'>
>>> |

```

**range()** možemo koristiti u tri osnovna oblika:

- **range (kraj)** – kreira slijed brojeva koji počinje *nulom*, a završava s *kraj-1*, s tim da je razlika susjednih elemenata uvijek jedan

**Primjer 7.**

```
>>> for i in range (7):
        print (i, end = ' ')

0 1 2 3 4 5 6
>>> |
```

- **range (pocetak, kraj)** – kreira slijed brojeva koji počinje brojem pocetak, a završava s brojem kraj-1, uz napomenu da je razlika susjednih elemenata u slijedu uvijek jedan

**Primjer 8.**

Ispiši sve brojeve veće od 5, a manje od 15.

```
>>> for i in range (5, 15):
        print(i, end = '')

567891011121314
>>> for i in range (5, 15):
        print(i, end = ' ')

5 6 7 8 9 10 11 12 13 14
>>>
```

- **range (pocetak, kraj, korak)** – kreira slijed brojeva koji počinje brojem pocetak i svaki je sljedeći broj za korak veći od prethodnoga. Slijed završava s najvećim brojem koji je strogo manji od kraj, a može se dobiti na opisani način.

**Primjer 9.**

```

>>> for i in range (0, 50, 5):
    print(i, end = ' ')

0 5 10 15 20 25 30 35 40 45
>>> for i in range (0, 50, 2):
    print(i, end = ' ')

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36
38 40 42 44 46 48
>>>

```

**Naredba break i continue**

Naredba break i continue omogućuju intervencije pri izvršenju naredbe ponavljanja. Naredba break zaustavlja izvršenje naredbe ponavljanja, dok naredba continue zaustavlja izvršenje iteracije.

**Primjer 10.**

Ovaj primjer traži u slijedu broj 99. Kad ga nađe prekida izvršenje naredbe ponavljanja.

```

>>> for i in niz_brojeva:
    if i == 99:
        print ('Pronašao sam broj!')
        break
    print('Ovaj broj', i,'nije traženi broj 99, traži dalje.')

Ovaj broj 1 nije traženi broj 99, traži dalje.
Ovaj broj 11 nije traženi broj 99, traži dalje.
Ovaj broj 24 nije traženi broj 99, traži dalje.
Ovaj broj 36 nije traženi broj 99, traži dalje.
Ovaj broj 89 nije traženi broj 99, traži dalje.
Ovaj broj 26 nije traženi broj 99, traži dalje.
Ovaj broj 35 nije traženi broj 99, traži dalje.
Ovaj broj 24 nije traženi broj 99, traži dalje.
Ovaj broj 56 nije traženi broj 99, traži dalje.
Ovaj broj 37 nije traženi broj 99, traži dalje.
Ovaj broj 19 nije traženi broj 99, traži dalje.
Pronašao sam broj!
>>> |

```

**Primjer 11.**

Ispiši neparne brojeve u zadanom nizu.

```
>>> for i in range (1,15):  
    if i %2 == 0:  
        continue  
    print('Neparan broj: ', i)
```

```
Neparan broj: 1  
Neparan broj: 3  
Neparan broj: 5  
Neparan broj: 7  
Neparan broj: 9  
Neparan broj: 11  
Neparan broj: 13
```