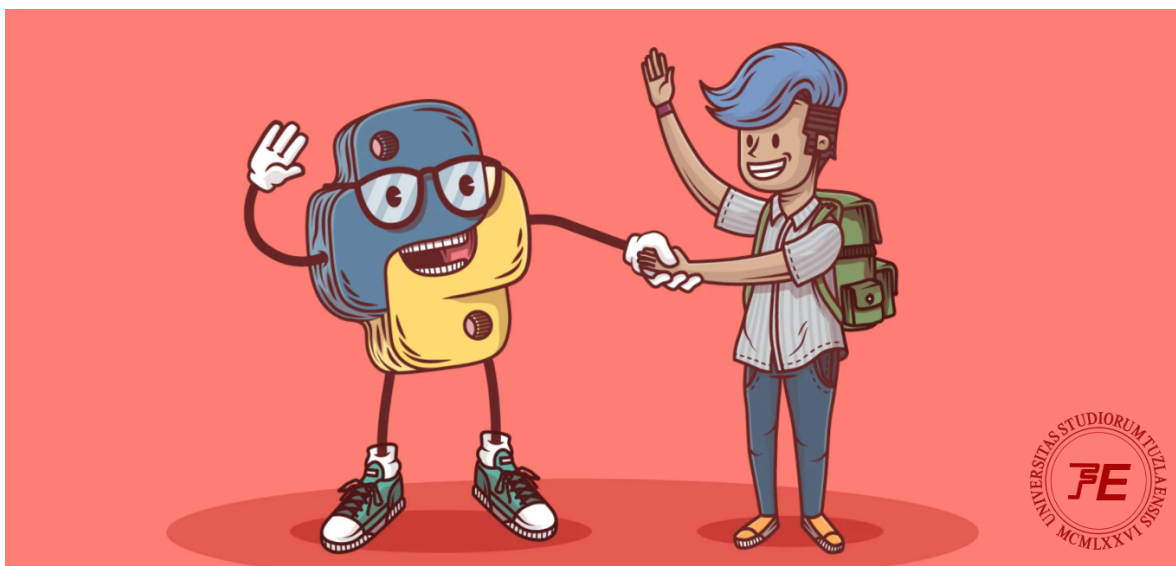


UNIVERZITET U TUZLI  
FAKULTET ELEKTROTEHNIKE

# Python tutoriali

Signali i sistemi

Emina Bajrektarević, Amira Sinanović



Tuzla, Septembar 2020. godine



# Python Tutorijal 3

- Kontrola toka programa
- IF i IF-ELSE naredba
- While petlja
- FOR petlja

Kontrola tokaprograma omogućava da odlučujemo na koji način se program ponaša u zadanim okolnostima, tj. zadacima koji se moraju odraditi zavisno od ulaznih vrijednosti. U Pythonu postoje dva osnovna elementa za kontrolu toka, a to su odluke i petlje. Oduke nam pomažu odlučivati koji dio programskoga koda je potrebno izvršiti, a koji dio programskoga koda je za ulazne vrijednosti nebitan. Petlje olakšaju ponavljanje stvari, npr. ako se žele ispisati svi brojevi od 1 do 1000, to je moguće napraviti na teži i lakši način. Teži način je da se ručno poziva funkcija `print()` te joj se ručno predaju vrijednosti od 1 do 1000, no taj način programerima oduzima puno vremena. Bolji način je da se ispis odradi pomoću petlje čija će početna vrijednost biti 1, završna vrijednost 1000, a faktor uvećanja 1. Na ovaj način se s nekoliko linija programskoga koda može ispisati 1000 brojeva, a prema potrebi količina ispisanih brojeva može se jednostavno povećati ili smanjiti.

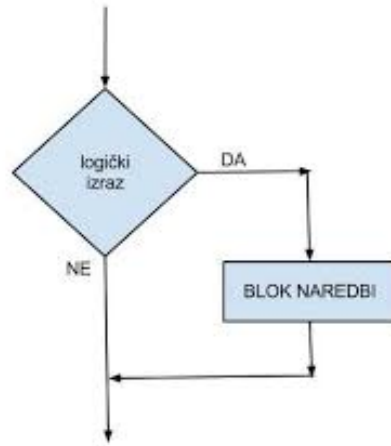
## IF naredba

Sintaksa `if` naredbe zahtjeva da se napiše ključna riječ `if`, nakon toga logički izraz (logički izraz uvijek na kraju završi u `True` ili `False` vrijednosti). Ako je uslov zadovoljen, izvršava se tijelo odluke, no ako izraz nije zadovoljen, tijelo odluke se neće izvršiti. Ključne riječi su riječi koje su unaprijed rezervirane od strane korištenoga programskog jezika, u ovom slučaju od strane Pythona. Iza ključnih riječi običnu se kriju naredbe. Svaki programski jezik ima skup ključnih riječi i te riječi ne mogu se koristiti kao imena varijabli. Princip izvršavanja `IF` naredbe je prikazan na slici ispod.

Nakon logičkog izraza potrebno je staviti dvotačku koja interpreteru označava da slijedi tijelo odluke. Primjer upotrebe `IF` naredbe se naradi u kodu ispod.

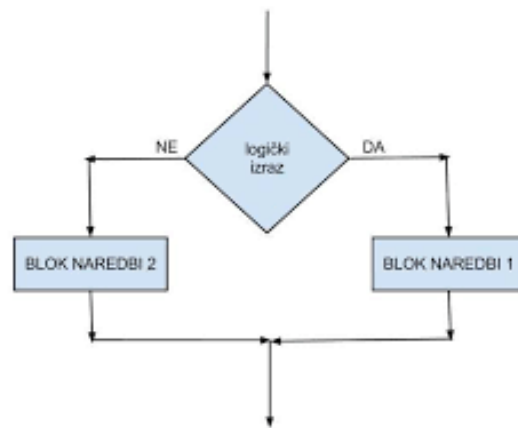
```
1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python skriptu koja nam omogu ava unos jednog cijelog broja i
4 provjerava da li je uneseni broj paran.
5 """
6
7 a=input("Unesite jedan cijeli broj: ");
8 a=eval(a);
9
10 if (a%2==0):
11     print("Uneseni broj je paran.") #ispisano samo ako je uslov zadovoljen
12
13 print(" Tekst nakon If naredbe.") #uvijek ce biti ispisano
```

## IF ELSE naredba



Slika 3.1: IF naredba

Kod IF-ELSE naredbe uvijek će biti izvršen jedan dio. Ako je uslov zadovoljen, izvršit će se tijelo if odluke, u suprotnom će se izvršiti tijelo else odluke. Princip izvršavanja IF-ELSE naredbe je prikazan na slici ispod.



Slika 3.2: IF-ELSE naredba

Ako je tijelo else naredbe samo jedna linija koda može se napisati u istom redu sa ključnom riječju ELSE, ali ako je tijelo iste naredbe složenije i sadrži više od jedne linije koda, sve linije moraju biti jedna ispod druge i jednako uvučene u odnosu na ključnu riješ ELSE, kao što je prikazano u primjeru ispod.

```

1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python skriptu koja nam omogu ava unos jednog cijelog broja i
4 provjerava da li je uneseni broj paran. Ako je broj paran pomno iti
5 ga sa 10, a ako je broj neparan pomno iti ga sa 2.
6 """
7
8 a=input("Unesite jedan cijeli broj: ");
9 a=eval(a);
10
11 if (a%2==0):
12     print("Uneseni broj je paran.") #ispisano samo ako je uslov zadovoljen
  
```

```

13     a*=10;
14 else:
15     print("Uneseni broj je neparan")
16     a*=2;
17     print("Tekst unutar ELSE naredbe")
18     # print("Greska") - tijelo ELSE naredbe mora biti
19     #napisano tako da je svaka linija koda jednako uvu ena
20
21 #nastavak programa
22 print("Tekst van IF-ELSE naredbe")

```

Naredbama if i if-else dodaje se još i if-elif-else uslov. Naredba elif služi za "neograničen" broj uslova, tj. provjera.

```

1 # -*- coding: utf-8 -*-
2 # -*- coding: utf-8 -*-
3 """
4 Napisati Python skriptu koja nam omogu ava unos jednog cijelog broja i
5 provjerava da li je uneseni broj djeljiv sa 10. Ako broj nije djeljiv sa 10
6 provjerava je li broj paran
7 """
8
9 a=input("Unesite jedan cijeli broj: ");
10 a=eval(a);
11
12 if (a%10==0):
13     print("Uneseni broj je djeljiv sa 10") #ispisano samo ako je uslov zadovoljen
14 elif (a%2==0):
15     print("Uneseni broj je paran.") #ispisano samo ako je uslov zadovoljen
16 else:
17     print("Uneseni broj je neparan i nije djeljiv sa 10.")
18
19 #nastavak programa
20 print("Tekst van IF-ELSE naredbe")

```

## WHILE petlja

Ova petlja prije izvršavanja programskog koda koji joj pripada, ispituje da li je uslov istinit ili ne. Ako je uslov istinit tada će se izvršiti tijelo petlje, dok se u suprotnom tijelo petlje preskače i izvršava se programski kod napisan ispod petlje. Sintaksa while petlje sastoji se od ključne riječi while, nakon toga slijedi logički izraz te dvotačka koja interpreteru označava da nakon nje slijedi tijelo petlje.

Primjer upotrebe while petlje se nalazi u primjeru ispod.

```

1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python kod koji e omogu iti korisniku da unese jedan cijeli broj n,
4 a nakon toga koriste i while petlju ispisuje poruku Ja volim Python n puta.
5
6 """
7
8 n= input("Unesite jedan cijeli broj:");
9 n=eval(n);
10
11 while n>0:
12     print("Ja volim Python.")
13     n-=1;
14
15 print("***** Kraj programa. *****")

```

## FOR petlja

```

1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python kod koji e omogu iti korisniku da unese jedan cijeli broj n,
4 a nakon toga koriste i FOR petlju ispisuje poruku Ja volim Python n puta.
5
6 """
7
8 n= input("Unesite jedan cijeli broj:");
9 n=eval(n);
10
11 for x in range(0,n):
12     print("Ja volim Python.")
13
14 print("***** Kraj programa. *****")

```

Za razliku od nekih drugih programskih jezika, kao što su C, C++, Java i tako dalje, u programskom jeziku Python for petlja koristi se na drugačiji način. Kod Pythona for petlja nema početnu ni završnu vrijednost, a ni aritmetički faktor uvećanja. For petlja u Pythonu iterira kroz elemente zadane sekvence. Zadana sekvenca može biti lista ili niz znakova-string. U nastavku slijedi primjer iteracije kroz elemente liste.

U gornjem su primjeru riječi for i in ključne riječi. Ova će se petlja izvršiti 10 puta, što je određeno funkcijom range(). Varijabla i pri svakoj iteracijipoprma drugu vrijednost, tj. sljedeću vrijednost koja se nalazi u listi koju je funkcija range() kreirala. Funkcija range() kreira listu elemenata. Funkcija range() može primiti jednu ili dvije vrijednosti. Ako primi jednu vrijednost, tj. ako je poziv funkcije range() sljedećeg oblika: range(10) tada se kreira lista od 10 elemenata, prvi element poprima vrijednost 0, dok zadnji element poprima vrijednost 9. Ta lista izgleda ovako:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Također, ova funkcija može poprimiti i dvije vrijednosti: range(10, 15) Tada se kreira lista od 5 elemenata, prvi element poprima vrijednost 10, dok zadnji element poprima vrijednost 14. Ta lista izgleda ovako:

10	11	12	13	14
----	----	----	----	----

U slučaju da je druga vrijednost koju prima funkcija range() manja ili jednaka naspram prvoj vrijednosti, rezultat će biti prazna lista.

U nastavku ćemo uraditi primjer Python programa koji određuje je li neki proizvoljni broj spremljen u varijablu n prost ili nije. Prosti brojevi ili prim-brojevi su svi prirodni brojevi strogo veći od broja 1, a koji su djeljivi bez ostatka samo sa brojem 1 i sami sa sobom (na primjer: 2, 3, 5, 7, 11, ...).

```

1 # =====
2 #   Provjera da li je uneseni broj prost.
3 #   =====
4
5
6 n = eval(input("Unesite jedan cijeli broj:"))
7
8 jePrim = True
9
10 for x in range(2, n):
11     if n % x == 0:
12         jePrim = False

```

```

13
14 if jePrim == True:
15     print('Broj', n, 'je prost!')
16 else:
17     print('Broj', n, 'nije prost!')

```

## Break i Continue naredba

Naredba break nam služi da terminiramo for ili while petlju. Kako petlje mogu biti jedna unutar druge, treba obratiti pažnju na to da naredba break izlazi samo iz petlje unutar koje je pozvana. Ova naredba se dosta često koristi kod sprječavanja beskonačnih petlji. Na primjer, uslov while petlje postavi se tako da zauvijek bude tačan, a onda unutar tijela while petlje naredbom if provjeravamo ako je neki uslov zadovoljen. U slučaju da je uslov zadovoljen, naredbom break izlazimo iz petlje.

U nastavku ćemo demonstrirati upotrebu break naredbe na primjeru programskog koda koji određuje da li je broj prost ili ne. Za razliku od prethodnog primjera ovog istog zadatka, u ovom primjeru jednom kada se pronade da je neki broj n djeljiv s nekim drugim brojem, daljnja analiza djeljivosti broja n s ostalim brojevima prestaje.

```

1 # =====
2 #  Provjera da li je uneseni broj prost, koriste i break naredbu
3 # =====
4
5
6 n = eval(input("Unesite jedan cijeli broj:"))
7
8 #for n in range(2, 11):
9 jePrim = True
10 for x in range(2, n):
11     if n % x == 0:
12         jePrim = False
13         break
14 if jePrim == True:
15     print('Broj', n, 'je prost!')
16 else:
17     print('Broj', n, 'nije prost!')

```

Naredba continue preskače na novu iteraciju petlje. Za razliku od naredbe break, naredba continue ne prekida izvođenje programa. Upotrebu naredbe continue ćemo demonstrirati na primjeru Python koda za provjeru parnosti brojeva.

```

1 # -*- coding: utf-8 -*-
2
3 for num in range(2, 6):
4     if num % 2 == 0:
5         print("Parni broj:", num)
6         continue
7     print("Neparni broj:", num)

```

### Zadatak 3.1

Napišite Python program koji će za uneseni broj ispisati da li je broj pozitivan, nula ili negativan koristeći elif naredbe.

### Rješenje

```

1 # -*- coding: utf-8 -*-
2 """
3 Napišite Python program koji će za uneseni broj ispisati da li je broj pozitivan,
4     nula
5     ili negativan koristeći elif naredbe.

```

```

5
6 """
7
8 a=eval(input("Unesite cijeli broj: "));
9
10 if a==0:
11     print("Unijeli ste 0!")
12 elif a>0:
13     print("Broj je pozitivan.")
14 else:
15     print("Broj je negativan.")
16
17 print("***** Kraj programa. *****")

```

### Zadatak 3.2

Napišite Python program koji će za unos dva broja vršiti sljedeće operacije: sabiranje, oduzimanje, množenje i dijeljenje. Na početku je potrebno korisniku omogućiti unos dva broja, zatim odabrati operaciju odabirom znaka te operacije (+, -, \*, /).

### Rješenje

```

1 """
2 Napišite Python program koji će za unos dva broja vršiti sljedeće operacije:
3     sabiranje,
4     oduzimanje, množenje i dijeljenje.
5 Na početku je potrebno korisniku omogućiti unos dva broja, zatim odabrati operaciju
6     odabirom znaka te operacije (+, -, *, /)
7
8 """
9 a=eval(input("Unesite prvi broj: "));
10 b=eval(input("Unesite drugi broj: "));
11 znak=input("Izaberite operaciju: ");
12
13 if znak=='+' :
14     print("Rezultat je:", a+b)
15 elif znak=='-' :
16     print("Rezultat je:", a-b)
17 elif znak=='*' :
18     print("Rezultat je:", a*b)
19 else:
20     print("Rezultat je:", a/b)
21
22 print("***** Kraj programa. *****")

```

### Zadatak 3.3

Napisati Python program koji će ispisati samo parne brojeve iz brojnog niza od prvih 10 pozitivnih cijelih brojeva.

### Rješenje

```

1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python program koji će ispisati samo parne brojeve iz brojnog niza od
4     prvih 10 pozitivnih cijelih brojeva.
5 """
6
7 for x in range(0,10,2):
8     print(x)
9
10 print("***** Kraj programa. *****")

```



### Zadatak 3.4

Ispisati prvih šest pozitivnih cijelih brojeva obrnutim redoslijedom. Npr. 1,2,3,4,5,6 ispisati kao 6,5,4,3,2,1.

### Rješenje

```

1 # -*- coding: utf-8 -*-
2 """
3 Ispisati prvih est pozitivnih cijelih brojeva obrnutim redoslijedom. Npr.
4   1,2,3,4,5,6
5   ispisati kao 6,5,4,3,2,1.
6 """
7 for x in range(6,0,-1):
8     print(x)
9
10 print("***** Kraj programa. *****")

```

### Zadatak 3.5

Napisati Python kod za unos prvih 10 cijelih brojeva, ali izostaviti broj 7.

### Rješenje

```

1 # -*- coding: utf-8 -*-
2 """
3 Napisati Python kod za unos prvih 10 cijelih brojeva, ali izostaviti broj 7.
4 """
5
6 for x in range(0,10):
7     if x!=7:
8         print(x)
9
10 print(" \n ***** Kraj programa. *****")

```

### Zadatak 3.6

U varijable a i b spremite dva broja. Ako je vrijednost varijable a bar za 50 veća od vrijednosti varijable b, a uz to je vrijednost varijable b parna, tada ispišite poruku "Uslovi su zadovoljeni.", u suprotnom ispišite poruku "Uslovi nisu zadovoljeni."

### Zadatak 3.7

Napisati Python kod koji će ispisati sve parne brojeve između 1 i 1000 koju su istovremeno djeljivi i sa 5 i sa 13.

### Zadatak 3.8

Napišite Python program koji ispisuje koliko ima prostih brojeva između dva proizvoljna broja.

