Strukture za kontrolu toka

Kontrola toka u programiranju, koja je dosta slična donošenju odluka u stvarnom životu, veoma je bitna jer nam pomaže da odredimo šta će program sledeće odraditi. Kontrola toka je u Pythonu implementirana pomoću if-else naredbi. Zbog toga jedna if naredba može odlučiti da li će se naredni blok koda izvršiti ili ne.

Strukture koje ćemo obraditi u ovoj nastavnoj jedinici su:

- if naredbe
- if-else naredbe
- ugnežđene if naredbe
- ulančani uslovi (elif naredba)
- jednolinijske uslovljene naredbe

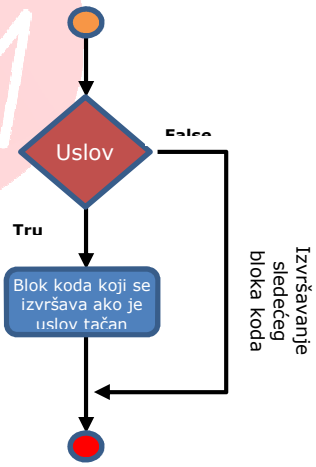
If naredba

Ova naredba u Pythonu se koristi uz dati izraz, pa tako važi pravilo: ako je dati izraz tačan (True), izvršiće se naredni blok koda, a ukoliko je netačan (False) – blok koda će se preskočiti i glavna petlja programa nastavlja daljim tokom. Naredbe koje treba izvršiti u datom bloku ispod if naredbe su uvučene sa četiri prazna mesta. Uvlačenje je jedini način u Pythonu da prevodilac zna da taj deo koda pripada tom bloku. Drugi programski jezici koriste vitičaste zagrade, a samo Python uvlačenje. Sintaksa za if naredbu je sledeća:

```
if <uslov koji se ispituje>:
```

Treba obratiti pažnju i na dve tačke, koje se uvek pišu nakon definisanja izraza u uslovu.

Dijagram logike if naredbe se nalazi na sledećoj ilustraciji.



Slika 9.1. Tok programa kada se koristi if naredba

Copyright © Link group

Prođimo kroz par primera korišćenja if naredbe:

Primer 1 if naredbe

```
x = 1
if x == True:
    print(True)
```

Kako smo u prethodnoj nastavnoj jedinici naučili da su True/False vrednosti potklase klase int(), ovde uviđamo da je vrednost 1 isto što i vrednost True. Važno je napomenuti da se u uslovnim naredbama koristi relacioni operator (==), a ne operator dodele (=).

Primer 2 if naredbe

```
a=5
if a < 0:
    print("Number is negative")
```

Ovaj izraz ne ispisuje ništa iz prostog razloga jer uslov u if naredbi nije tačan. Dakle, glavna programska petlja će samo preskočiti blok koda unutar ove petlje.

```
if a > 0:
    print("Number is positive")
```

Takođe, uslovi u if naredbama se mogu i pisati u zagradama, za čiji ispis sintaksa izgleda ovako:

```
if (condition to test for):
```

If-else naredbe

U jednom od prošlih primera smo imali situaciju kada je uslov u if naredbi netačan. Kako bismo obradili i taj slučaj, koristimo if-else naredbu. Dakle, ova naredba se koristi u slučajevima kada želimo da pokrijemo obe situacije – kada je uslov tačan i kada nije.

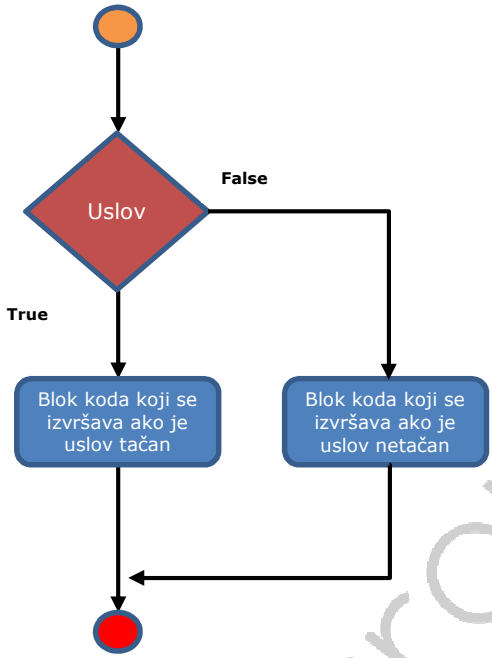
Sintaksa ove naredbe izgleda ovako:

```
if <uslov koji se ispituje>:
    <blok koda koji se izvršava ako je uslov tačan>
else:
    <blok koda koji se izvršava ako uslov nije tačan>
```

Posebnu pažnju treba obratiti na uvlačenje linija koda, jer je to jedini način da Python prevodilac zna da se radi o uslovnim naredbama i blokovima koda unutar njih.

Tok programa kada glavna programska petlja naiđe na if-else naredbu je prikazan na sledećoj ilustraciji:

Copyright © Link group



Slika 9.2. Tok programa kada se koristi if-else naredba

Na sledećem primeru ćemo videti kako izgleda korišćenje ove naredbe.

Primer if/else naredbe:

Radno okruženje

```
year = 2020
if (year == 2020):
    print("Current year: 2020")
else:
    print("Current year is not: 2020")
```

Objašnjenje:

Obratite pažnju na ključnu reč else – gde ona stoji i kako se u tom else bloku mora nalaziti blok koda odn. minimum jedna naredba. Ako u else bloku ne postoji naredba dolazi do SyntaxError greške jer sintaksa Python programskog jezika podrazumeva makar jednu naredbu unutar ovog bloka koda.

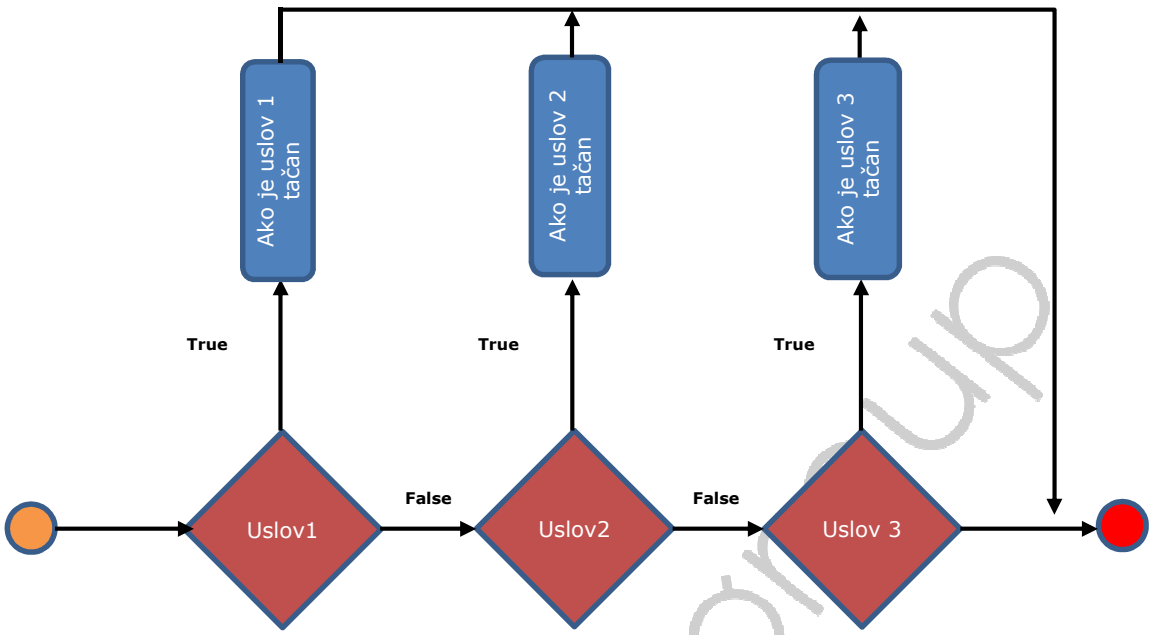
Ako želimo da proverimo više uslova, koristićemo elif naredbu.

Ulančani uslovi (elif naredba)

Za razliku od Jave i C-a, Python sintaksa dozvoljava korišćenje elif naredbe kada je potrebno proveriti više uslova, umesto korišćenja else-if naredbe koje je uobičajena praksa

Copyright © Link group

u drugim programskim jezicima. Tok programa kada se naiđe na if-elif deo je prikazan na sledećoj ilustraciji:



Slika 9.3. If-elif tok programa

Ovde je prikazana logika iz if-elif toka programa. Glavna programska petlja ulazi i proverava uslov 1; ako je tačan, ispuniće se blok koda ispisan u njemu. Ako taj uslov nije tačan, petlja nastavlja dalje do drugog uslova. Opet proverava tačnost datog uslova i odlučuje da li će se izvršiti blok koda u okviru drugog uslova ili će nastaviti do trećeg i na osnovu uslova iz treće naredbe odlučiti kako dalje postupiti.

Važno je napomenuti da se obrati pažnja na pravilno uvlačenje linija koda, pošto su to česte sintaksičke greške.

Primer sintakse if-elif toka programa

Sintaksa if-elif toka programa je sledeća:

```
if <Prvi uslov koji se ispituje>:
    <blok koda koji se izvršava ako je uslov tačan>
elif: <Drug uslov koji se ispituje >
    <blok koda koji se izvršava ako je uslov tačan>
elif: <Treći uslov koji se ispituje >
    <blok koda koji se izvršava ako je uslov tačan>
else:
    <blok koda koji se izvršava ako nijedan od prethodnih uslova nije
tačan>
```

Elif naredbi možemo imati i više od trenutne dve u tom primeru, dok god na kraju iskoristimo naredbu else: . Primer:

Copyright © Link group

Radno okruženje

```
mark = 73

if mark >= 91:
    print("Mark ten")
elif mark >= 81:
    print("Mark nine")
elif mark >= 71:
    print("Mark eight")
elif mark >= 61:
    print("Mark seven")
elif mark >= 51:
    print("Mark six")
else:
    print("Not passed.")
```

Objašnjenje:

Prvi uslov koji je tačan jeste elif mark>=71 i zbog toga će se izvršiti naredba unuar tog bloka koda odn. ispisati 'Mark eight'.

Na ovom primeru možemo videti kako je prvo proveren prvi uslov (`if mark >= 91`), koji nije bio zadovoljen pa je glavna programska petlja nastavila sa proveravanjem sledećih uslova. Nakon što je utvrđeno da je vrednost promenljive mark veća od 71 (`elif mark >= 71`), program je ispisao poruku 'Mark eight' i izašao iz čitavog `if-elif-else` bloka.

Važno je i napomenuti mogućnost ugnežđenih if-else naredbi i naknadne provere uslova.

Primer ugnežđene if-else naredbe

```
if <Prvi uslov koji se ispituje>
    <blok koda koji se izvršava ako je uslov tačan>
    if <ugnežđeni_uslov>:
        <blok koda koji se izvršava ako su oba if uslova bila tačna>
    else:
        <blok koda koji se izvršava ako je prvi if uslov tačan a drugi
      ne>
else:
    <blok koda koji se izvršava ako nijedan od prethodnih uslova nije
tačan>
```

Ovo je samo jedan od mogućih primera kako se ugnežđene if-naredbe mogu koristiti.

Pitanje

Naredni odlomak koda će ispisati:

```
if None:
   print("True")
else:
   print("False")
```

Copyright © Link group

- sintaksnu grešku
- True
- **False**

Objašnjenje:

Ispisaće netačno (`False`), jer je vrednost None interpretirana kao False. U Pythonu, sve vrednosti, bez obzira na tip, interpretirane su kao True, osim sledećih slučajeva:

- *konstante koje su definisane da budu netačne: None i False;*
- *nula ili bilo koji numerički tip podatka koji predstavlja nulu: 0, 0.0, 0j;*
- *prazne sekvence: '', (), [], {}, set().*

Jednolinijske uslovljene naredbe

Ove naredbe se koriste samo u slučajevima kada nakon provere tačnosti datog uslova u if-naredbi imamo da izvršimo samo jednu naredbu.

Sagledajmo to kroz primer sa početka.

Primer

```
x = 1
if x == True:
    print(True)

#Result is True.


# The counterpart to this example is:

x = 1
if x == True: print(True)

# The result will also be True
```

Vežba

Napisati program koji za dati broj n proverava da li je broj pozitivan, negativan ili jednak nuli i poruku ispisuje na ekranu.

Radno okruženje

Rešenje 1:

Zadak se može rešiti na više načina, a mi smo se ovde opredelili za korišćenje ugnežđene if-else konstrukcije.

Copyright © Link group

```
n = 5

if n >= 0:
    if n == 0:
        print("Zero")
    else:
        print("Number is positive")
else:
    print("Number is negative")
```

Rešenje 2:

Takođe, ovo se moglo rešiti i korišćenjem if-elif-else konstrukcije.

```
n = 5

if n == 0:
    print("Zero")
elif n > 0:
    print("Number is positive")
else:
    print("Number is negative")
```

Rezime

- Strukture za kontrolu toka u Pythonu su: if, if-else i if-elif-else.
- If naredba se definiše pomoću ključne reči if koju prati uslov koji treba ispitati. Ako je uslov tačan, kod ispod ove naredbe će se izvršiti, a ako nije, program će nastaviti dalje.
- If-else naredba se definiše iz dva dela. Prvi deo čini ključna reč if koju prati uslov koji treba ispitati, dok drugi deo prati naredba else:. Ako je uslov koji se ispituje u if naredbi tačan, izvršiće se kod ispod ove naredbe; ako nije tačan, izvršiće se kod ispod else naredbe.
- If-elif-else (ulančani uslovi) definišu se iz najmanje tri dela. Prvi deo čini ključna reč if koju prati uslov koji treba ispitati. Drugi deo je minimum jedna (može ih biti i više) elif naredba koja će se izvršiti ako uslov iz if naredbe nije tačan. Ako uslov nije tačan ni u if ni u elif delu koda, izvršiće se deo koda ispod else naredbe.
- U Pythonu je moguće kreiranje ugnežđenih struktura za kontrolu toka.

Copyright © Link group