

Poredak logičkih uvjeta u naredbama odluke

Što smo naučili prošli put

Prošli put smo naučili da postoje tri osnovna logička operatora u naredbama odluke.

To su **not**, **and** i **or**.

Operator **and** koristimo u situacijama kada želimo biti sigurni da su svi uvjeti istovremeno istiniti.

Operator **or** koristimo u situacijama kada želimo biti sigurni da je barem jedan uvjet istinit.

Operator **not** koristimo kada želimo istinitost nekog uvjeta pretvoriti u laž i obrnuto.

Što ćemo naučiti do sljedećeg četvrtka?

Naučit ćemo nešto više o logičkim uvjetima.

Ujedno ćemo naučiti o poretku logičkih operatora, odnosno kojim redoslijedom Python provjerava logičke uvjete i operatore.

Logički uvjet

Promotrite sljedeći logički uvjet:

$n > 5$

Ovaj uvjet može imati **dvije vrijednosti** – **True** ili **False**.

Ako je vrijednost varijable n veća od 5, uvjet će biti **True** (istina).

Ako je vrijednost varijable n manja ili jednaka 5, gornji uvjet će biti **False** (laž).

Stoga se naredba odluke `if n > 5`: zapravo svodi na `if True`: ako je vrijednost varijable n veća od 5, ili na `if False`: ako je vrijednost varijable n manja ili jednaka 5.

Da bi se izvodile naredbe koje ovise o naredbi odluke `if`, uvjet u naredbi `if` mora biti **True**.

Logički uvjeti

```
if n > 5:  
    n = n + 1  
    print(n)
```

Ove dvije naredbe
će se izvršiti samo
ako je uvjet `n > 5`
jednak `True`,
odnosno ako
imamo `if True`:

Poredak operacija

U matematici smo naučili da računске operacije imaju svoj redoslijed izvršavanja.

Prvo se koristi računska operacija **množenja**. Nakon toga operacija **djeljenja**. Potom operacija **zbrajanja** i na kraju operacija **oduzimanja**.

Ako imamo npr. $20 - 3 * 4 : 2$, onda je redoslijed operacija sljedeći:

1. prvo se pomnože brojevi 3 i 4, pa nakon toga imamo $20 - 12 : 2$
2. Nakon toga se podijele brojevi 12 i 2, pa nakon toga imamo $20 - 6$
3. Na kraju se oduzmu brojevi 20 i 6, pa imamo $20 - 6 = 14$

Poredak logičkih operacija

Kao što postoji redoslijed izvođenja računskih operacija, tako postoji i redoslijed provjeravanja logičkih uvjeta i operatora.

Pritom vrijedi sljedeće pravilo:

Prvo se provjerava i izvršava operator **not**. Nakon njega se provjerava i izvršava operator **and**. Zadnji koji se provjerava i izvršava je operator **or**.

Dakle poredak je sljedeći:

1. **not**
2. **and**
3. **or**

Primjer

Promotrite sljedeći primjer:

```
if n == 5 or n == 6 and n == 7:
```

U ovom primjeru imamo tri uvjeta. Prvi i drugi uvjet su povezani operatorom `or`, a drugi i treći uvjet su povezani operatorom `and`.

Koji je redoslijed provjeravanja?

Prvo će se provjeriti `n == 6 and n == 7` jer se operator `and` provjerava prije operatora `or`.

Pretpostavimo da je rezultat tog provjeravanja `False`. Onda ćemo imati sljedeće:

```
if n == 5 or False:
```

Nakon toga će se provjeravati `n == 5 or False`. Ako je vrijednost varijable `n` jednaka 5, onda ćemo imati provjeru `True or False`, a ako nije jednaka 5 onda ćemo imati provjeru `False or False`.

Promjena redoslijeda

Ako u matematici želimo promijeniti redoslijed izvođenja određenih operacija, onda koristimo zagrade.

Npr., ako imamo $20 - 3 * 4$ i ako želimo da se prvo oduzmu brojevi 20 i 3, napisat ćemo sljedeće: $(20 - 3) * 4$. Sada će se prvo oduzeti brojevi 20 i 3, pa ćemo imati $17 * 4$ i konačan rezultat će biti 68.

Na sličan način možemo izmijeniti redoslijed provjeravanja logičkih uvjeta i operatora.

Ako u primjeru s prethodne stranice želimo da se prvo provjeri operator `or`, onda ćemo napisati sljedeće:

```
if (n == 5 or n == 6) and n == 7:
```

Zadaci

Odgovorite na sljedeće zadatke. Odgovore pošaljite u obliku privatne poruke na Yammeru. Odgovore možete poslati do sljedećeg četvrtka.

1. Koji je redoslijed provjeravanja uvjeta u sljedećoj naredbi odluke?

```
if n != 7 and n == 2 or n > 50:
```

2. Koji je redoslijed provjeravanja uvjeta u sljedećoj naredbi odluke?

```
if not n > 5 or n == 7 and n < 100:
```

3. Kako treba izgledati sljedeća naredba odluke ako želimo da se prvo provjeri `or` operator?

```
if n == 7 and n == 9 or n == 11:
```

4. Hoće li se izvršiti sljedeća `print` naredba ako je vrijednost varijable `n` jednaka 5?

```
if n == 5 or n != 7 and n == 9:
```

```
    print(n)
```