



Podmienky

Kryštof Mrózek (445429@mail.muni.cz)

Kristína Tomanková (kristinatomankova@mail.muni.cz)

Radoslav Brunovský (rbrunovsky@mail.muni.cz)

Operátory - připomenutí

- Existuje celá řada operátorů - operací mezi dvěma hodnotami / proměnnými
- Jejich konkrétní význam záleží na datovém typu
- V tuto chvíli se podíváme pouze na operace mezi čísly
- *Cvičení 4: Operace +, -, *, /, **, //, % aplikujte na hodnoty 15 a 4. Zkuste odhadnout význam každé operace*
 - $15 + 4 = 19$ – sčítání
 - $15 - 4 = 11$ – odčítání
 - $15 * 4 = 60$ – násobení
 - $15 / 4 = 3.75$ – dělení
 - $15 ** 4 = 50625$ – umocňování
 - $15 // 4 = 3$ – dělení beze zbytku, tzv. “floor division“ ($15 = 4 * 3 + 3$)
 - $15 \% 4 = 3$ – modulo, zbytek po dělení ($15 = 4 * 3 + 3$)
 - **Všechny tyto operace jsou definované na celých (int) i desetinných (float) číslech**

Boolean výrazy

- Boolean výraz - výraz ktorý je pravdivý alebo nepravdivý - má hodnotu True alebo False
- Používame pri nich racionálne/porovnávacie operátory:
- Operátory vyhodnotia boolean výraz a vrátia hodnotu True alebo False

Table 2-1: Comparison Operators

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Cvičenie

Vyskúšajte vyhodnotiť nasledujúce výrazy

```
1 print(42==42)
2 print(42==99)
3 print(2!=3)
4 print(2!=2)
5 print('hello'=='hello')
6 print('Hello'=='hello')
7 print(True!=True)
8 print(True==True)
9 print(True!=False)
10 print("42"==42)
11 print(42.0==42)
12 print(42<100)
13 print(42>100)
```

Logické operátory

- Porovnávají boolean hodnoty
- Binárne: **and**, **or**
- Unárne: **not**

Operátor **not** vyhodnotí výraz tak, aby mal opačnú pravdivostnú hodnotu

Cvičenie

Vyskúšajte vyhodnotiť nasledujúce výrazy

```
1 print(True and True)
2 print(True and False)
3 print(False and False)
4 print(True or True)
5 print(True or False)
6 print(False or False)
7 print(not True)
```

Podmienky

- Umožňujú spustiť časť kódu iba vtedy, ak je zadaná podmienka splnená
- Veľmi podstatné pri písaní zložitejších programov
- Štruktúrou veľmi podobné ako definovanie funkcie

```
1   a=3
2   if a>=2:
3       print("a je vacsie ako dva")
```

Formát if-else

- Do kódu pod **else** vetvou sa píše časť kódu, ktorý sa spustí keď je podmienka zadaná pri **if** nespĺnená

```
1  a=1
2  if a>=2:
3      print("a je vacsie ako dva")
4  else:
5      print("a je mensie ako dva")
```


Formát if-elif-else

```
1  a=2
2  if a>2:
3      print("a je vacsie ako dva")
4  elif a==2:
5      print("a sa rovna dva")
6  else:
7      print("a je mensie ako dva")
```

Vnorené podmienky

```
1  a=3
2  b=2
3  if a==3:
4      if b!=2:
5          print("a==2 a b!=2")
6      else:
7          print("a==2 a b!=2")
```

Podmienky s logickými operátormi

```
1  a=3
2  b=2
3  if a==3 and b!=2:
4      print("a==2 a b!=2")
5  else:
6      print("a==2 a b==2")
```

Podmienka s not

```
1 som_hladny=False
2
3 if not som_hladny:
4     print("Nemusim jest")
5 elif som_hladny:
6     print("Musim jest")
```

Záverečné cvičenie

Ak je 1 strana trojuholníka väčšia ako súčet ostatných dvoch strán, tak tento trojuholník nenakreslíte. Inak áno. Napíšte funkciu, ktorá bere 3 dĺžky strán, a overí, či je možné zostaviť trojuholník z týchto strán.

Riešenie

```
1  a=1
2  b=2
3  c=3
4
5  if a+b<c or b+c<a or a+c<b:
6  |   print("Trojuholnik sa neda nakreslit")
7  else:
8  |   print("Trojuholnik sa da nakreslit")
```

Záverečné cvičenie

Napíšte funkciu, ktorá bude mať jeden argument - číslo - a rozhodne či je to číslo väčšie, menšie alebo rovné nule a tento výsledok vypíše.