

Oznam

Budúci týždeň vo štvrtok 19.10. je študijné voľno.

Študenti, ktorí majú cvičenia vo štvrtok, sa budúci týždeň môžu zúčastniť cvičenia:

Streda 18.10. od 17:00 do 19:00 v cpu

Pre študentov, ktorí majú cvičenie vo štvrtok, ale nie je účasť na cvičení budúci týždeň povinná.

Ak máte cvičenia vo štvrtok a chceli by ste sa zúčastniť niektorého cvičenia budúci týždeň, ale cvičenie v stredu o 17:00 vám nevyhovuje, napíšte email prednášajúcemu (tomas.fabsic@stuba.sk).

PROG1: Prednáška 4

Podmienené príkazy

Domáca úloha

1. Prečítajte si časti 5.1 - 5.7 v knihe.
2. Vyriešte úlohy z Cvičenia 4.

Vstupy z klávesnice

Program môže používateľa vyzvať aby zadal vstup z klávesnice.

Robí sa to pomocou funkcie **input()**.

S hodnotou, ktorú používateľ zadá, bude Python pracovať ako s **reťazcom**! (Ak chceme, aby Python pracoval s hodnotou ako s číslom, musíme ju **pretypovať**!)

Vstupy z klávesnice

Príklad:

```
def parne(n):  
    for i in range(n):  
        print((i+1)*2)  
  
def parne_interaktivna_verzia():  
    print('Zadajte kolko parnych cisel chcete vypisat.')  
    vstup=input()  
    n=int(vstup)  
    parne(n)  
  
parne_interaktivna_verzia()
```

Operátor % (modulo)

Príkaz

Cislo1 % Cislo2

vráti zvyšok z čísla Cislo1 po delení číslom Cislo2.

Príklad:

```
>>> 7%3  
1
```

Boolovské výrazy

Boolovské výrazy = výrazy, ktoré sú buď pravdivé alebo nepravdivé

Príklad:

```
>>> 5 == 5
```

```
True
```

```
>>> 5 == 6
```

```
False
```

Boolovské výrazy

Príklad:

```
>>> 5 == 5
```

```
True
```

```
>>> 5 == 6
```

```
False
```

Operátor `==` vráti hodnotu **True**, ak sa dve hodnoty rovnajú, inak vráti **False**

Pozor! Častá chyba je písať `=` namiesto `==` !

Ine operatory porovnavania

<code>x != y</code>	<code># x is not equal to y</code>
<code>x > y</code>	<code># x is greater than y</code>
<code>x < y</code>	<code># x is less than y</code>
<code>x >= y</code>	<code># x is greater than or equal to y</code>
<code>x <= y</code>	<code># x is less than or equal to y</code>

Pozor! Operátory `=>` a `=<` **neexistujú!**

Logické operátory

Máme 3 logické operátory: **and**, **or**, **not**

Operátor **and**

Príklad:

```
>>> (2<5) and (2<6)  
True
```

Operátor **and** vráti hodnotu True, iba ak sú obidve podmienky splnené. Inak vráti False.

```
>>> (2<5) and (2<0)  
False
```

Operátor or

Príklad:

```
>>> (2<5) or (2<0)
```

```
True
```

Operátor **or** vráti hodnotu True, ak je aspoň jedna podmienka splnená. Ak ani jedna podmienka nie je splnená, operátor vráti False.

Pozor! Ak sú obidve podmienky splnené operátor **or** vráti True!

```
>>> (2<5) or (2<6)
```

```
True
```

Operátor not

Príklad:

```
>>> not (2<0)  
True
```

Podmienené príkazy

Príklad 1:

```
if x > 0:  
    print('x is positive')
```

Príklad 2:

```
if 0 < x and x < 10:  
    print('x is a positive single-digit number.')
```

Podmienené príkazy - else

Príklad 3:

```
if x % 2 == 0:  
    print('x is even')  
else:  
    print('x is odd')
```

Podmienené príkazy - elif

Príklad 4:

```
if x < y:  
    print('x is less than y')  
elif x > y:  
    print('x is greater than y')  
else:  
    print('x and y are equal')
```

Poznámka: elif je skratka pre else if

Podmienené príkazy - elif

Poznámka: elif je skratka pre else if

Príkazy z príkladu 4 by sme mohli prepísať aj nasledovne:

```
if x<y:
    print('x is less than y')
else:
    if x>y:
        print('x is greater than y')
    else:
        print('x and y are equal')
```

elif vs if

Uvažujte nasledovný skript:

```
print('Zadajte Vas pocet bodov.')
a=input()
x=int(a)
if x>=92:
    print('Vasa znamka je A.')
elif x>=83:
    print('Vasa znamka je B.')
elif x>=74:
    print('Vasa znamka je C.')
elif x>=65:
    print('Vasa znamka je D.')
elif x>=56:
    print('Vasa znamka je E.')
else:
    print('Vasa znamka je FX.')
```

Čo by sa stalo, ak by sme všetky elif-y nahradili if-mi?

Dôležitý príklad z prednášky 2

Pomocou for-cyklu definujte funkciu s parametrom n , ktorá vráti súčet čísel od 1 po n .

Riešenie:

```
def sucet_po_hranicu(n):  
    sucet=0  
    for i in range(n) :  
        sucet=sucet+i+1  
    return sucet
```

Druhý dôležitý príklad

Definujte funkciu s parametrom n , ktorá načíta n čísel z klávesnice a vráti počet načítaných čísel, ktoré boli párne.

Druhý dôležitý príklad

Definujte funkciu s parametrom n , ktorá načíta n čísel z klávesnice a vráti počet načítaných čísel, ktoré boli párne.

Riešenie:

```
def pocet_parnych(n):  
    pocet=0  
    for i in range(n):  
        vstup=input()  
        cislo=int(vstup)  
        if cislo%2==0:  
            pocet=pocet+1  
    return pocet
```

Tretí dôležitý príklad

Definujte funkciu s parametrom n , ktorá načíta n čísel z klávesnice a vráti najväčšie načítané číslo.

Tretí dôležitý príklad

Definujte funkciu s parametrom n, ktorá načíta n čísel z klávesnice a vráti najväčšie načítané číslo.

Riešenie:

```
def max_cislo(n):  
    vstup=input()  
    maximum=int(vstup)  
    for i in range(n-1):  
        vstup=input()  
        cislo=int(vstup)  
        if cislo>maximum:  
            maximum=cislo  
    return maximum
```

Pozor! V riešení bolo dôležité, aby sme premennu maximum inicializovali prvým vstupom! Inicializácia maximum=0 by bola chybná! (Všetky načítané čísla môžu byť záporné!)