

PROG1: Prednáška 4

Podmienenene prikazy

Domaca uloha

1. Precitajte si casti 5.1 - 5.7 v knihe.
2. Vyrieste ulohy z Cvicenia 4.

Vstupy z klavesnice

Program moze uzivatela vyzvat aby zadal vstup z klavesnice

Robi sa to pomocou funkcie **input**.

S hodnotou, ktoru uzivatel zada, bude Python pracovat ako so **stringom**! (Ak chceme aby Python pracoval s hodnotou ako s cislom, musime ju **pretypovat**!)

Vstupy z klavesnice

Priklad:

```
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()
```

Operator %

Prikaz

Cislo1 % Cislo2

vraťi zvyšok z čísla Cislo1 po delení číslom Cislo2.

Priklad:

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

Boolovske vyrazy

Boolovske vyrazy = vyrazy, ktore su bud pravdive alebo nepravdive

Priklad:

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

```
True
```

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

```
False
```

Boolovske vyrazy

Priklad:

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

```
True
```

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

```
False
```

Operator `==` vrati hodnotu `True` ak sa dve hodnoty rovnaju, inak vrati `False`

Pozor! Casta chyba je pisat `=` 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! Operatory `=>` a `=<` neexistuju!

Logicke operator

Mame 3 logicke operator: **and, or, not**

Operator and

Priklad:

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

Operator **and** vrati hodnotu True, iba ak su obidve podmienky splnene. Inak vrati False.

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

Operator **or**

Priklad:

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

```
True
```

Operator **or** vrati hodnotu True, ak je aspon jedna podmienka splnena. Ak ani jedna podmienka nie je splnena, operator vrati False.

Pozor! Ak su obidve podmienky splnene operator **or** vrati True!

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

```
True
```

Operator **not**

Příklad:

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

Podmienenene prikazy

Priklad 1:

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

Priklad 2:

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

Podmienenene prikazy

Priklad 3:

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

Podmienenene prikazy

Priklad 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')
```

Poznamka: elif je skratka pre else if

Podmienenene prikazy

elif-ov mozeme pouzit lubovolne vela

Priklad 5:

```
if choice == 'a':  
    draw_a()  
elif choice == 'b':  
    draw_b()  
elif choice == 'c':  
    draw_c()
```


Vnorene podmienky

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

Dolezity priklad z prednasky 2

Pomocou for-cyklu definujte funkciu s parametrom n, ktora vrati sucet cisel od 1 po n.

Riesenie:

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

Druhy dolezity prikklad

Definujte funkciu s parametrom n , ktora nacita n cisel z klavesnice a vrati pocet nacistanych cisel, ktore boli parne.

Druhý doležitý příklad

Definujte funkci s parametrem n , která načítá n čísel z klávesnice a vrátí počet načítaných čísel, které jsou párne.

Riesenie:

```
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
```

Treti dolezity prikklad

Definujte funkciu s parametrom n , ktora nacita n cisel z klavesnice a vrati najvacsie nacistane cislo.

Treti dolezity priklad

Definujte funkciu s parametrom n, ktora nacita n cisel z klavesnice a vrati najvacsie nacistane cislo.

Riesenie:

```
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 rieseni bolo dolezite, aby sme premennu maximum inicializovali prvym vstupom! Inicializacia maximum=0 by bola chybna! (Vsetky nacistane cisla mozu byt zaporne!)