

Cykly

PLIN048 – Základy programování pro humanitní obory

Richard Holaj
FF MU

12. března 2017

Co nás dnes čeká?

Obecné cykly s podmíněným opakováním

- Podmíněné a nepodmíněné skoky v programu

- Princip cyklu

- Závislost příkazů na kontextu – stejný kód, jiný stav

Cykly s pevně daným opakováním

- Cykly pro procházení kolekcí (foreach)

- Klasická varianta cyklu for

Cvičení

Podmíněné a nepodmíněné skoky v programu

- ▶ elementární instrukce
- ▶ skutečná podstata podmínek a cyklů

Princip cyklu

Zjednodušená interní reprezentace cyklu

```
1 | a = 5
2 | if a > 0:
3 |     print(a)
4 |     a = a - 1
5 |     jmp(2)
```

Zápis cyklu

Python

```
1 |     a = 5
2 |     while a > 0:
3 |         print(a)
4 |         a = a - 1
```

JavaScript

```
1 |     var a = 5;
2 |     while(a > 0) {
3 |         console.log(a);
4 |         a = a - 1;
5 |     }
```

Vyhodnocení cyklu v kontextu I

```
1 a = 5
2 if a > 0:
3     print(a)
4     a = a - 1
5 if a > 0:
6     print(a)
7     a = a - 1
8 if a > 0:
9     print(a)
10    a = a - 1
11 if a > 0:
12    print(a)
13    a = a - 1
14 if a > 0:
15    print(a)
16    a = a - 1
17 if a > 0:
18    print(a)
19    a = a - 1
```

Vyhodnocení cyklu v kontextu II

```
1 | a = 5
2 | print(5)
3 | a = 5 - 1
4 | print(4)
5 | a = 4 - 1
6 | print(3)
7 | a = 3 - 1
8 | print(2)
9 | a = 2 - 1
10 | print(1)
11 | a = 1 - 1
```

Neuspořádané kolekce

Python

```
1 a = {"a":1, "b":2, "c":3}
2 for x in a.values(): #1, 2, 3
3     print(x)
4 for x in a.keys(): #a, b, c
5     print(x)
```

JavaScript

```
1 var a = {"a":1, "b":2, "c":3};
2 for(var x of a) { //1, 2, 3 (experimental)
3     console.log(x);
4 }
5 for(var x in a) { //a, b, c
6     console.log(x);
7 }
```


Neuspořádané kolekce – reprezentace pomocí while

```
1 a = {"a":1, "b":2, "c":3}
2 keys = a.keys() # ["a", "b", "c"]
3 i = 0
4 while i < len(keys):
5     x = keys[i]
6     print(x)
7     i = i + 1
8
9 values = a.values() # [1, 2, 3]
10 i = 0
11 while i < len(values):
12     x = values[i]
13     print(x)
14     i = i + 1
```

Uspořádané kolekce

Python

```
1 a = [7, 8, 9]
2 for x in a: #7, 8, 9
3     print(x)
4 for x in range(len(a)): #0, 1, 2
5     print(x)
```

JavaScript

```
1 var a = [7, 8, 9];
2 for(var x of a) { //7, 8, 9 (experimental)
3     console.log(x);
4 }
5 for(var x in a) { //0, 1, 2
6     console.log(x);
7 }
```

Uspořádané kolekce – reprezentace pomocí while

```
1 b = [7, 8, 9]
2 i = 0
3 while i < len(b):
4     x = b[i]
5     print(x)
6     i = i + 1
7
8 i = 0
9 while i < len(b): # varianta 1
10    x = i
11    print(x)
12    i = i + 1
13 i = 0
14 indices = range(len(b))
15 while i < len(indices): # varianta 2
16    x = indices[i]
17    print(x)
18    i = i + 1
```

Princip cyklu for

- ▶ generuje seznam čísel
- ▶ dolní mez
- ▶ horní mez
- ▶ definice kroku

Ukázka a reprezentace cyklu for

Python

```
1 for i in range(0, 10, 2):
2     print(i)
3 j = 0
4 while(j < 10):
5     print(j)
6     j = j + 2
```

JavaScript

```
1 for(var i = 0; i < 10; i = i + 2) {
2     console.log(i);
3 }
4 var j = 0
5 while(j < 10) {
6     console.log(j);
7     j = j + 2;
8 }
```

Cvičení

Co bude výstupem následujícího kódu?

```
1 | for i in [2, 4, 6, 8]:  
2 |     print(i // 2)
```

Co bude výstupem následujícího kódu?

```
1 | for(var i=1; i<100; i*=2) {  
2 |     console.log(i);  
3 | }
```

Co bude výstupem následujícího kódu?

```
1 | for i in range(1, 3):  
2 |     for j in range(2, 4):  
3 |         print(i + j)
```

Cvičení

Napište kód, který vypíše pomocí jednoho nebo více cyklů následující posloupnost:

1, 4, 9, 16, ...

True, False, False, True, False, False, True, ...

1, 1, 2, 1, 2, 3, 1, 2, 3, 4, ...

Napište kód, který vypíše pomocí jednoho nebo více cyklů prvky kolekce *col* společně s cestami k nim.

```
1 | col = {"A": [1, 3], "B": [2, 7, 5], "C": [8, 2, 9]}
```

Cvičení

Mějme proměnnou *tracks* obsahující název skladeb v playlistu a proměnnou *repeat*, která říká, zda chceme po přehrání poslední skladby pokračovat znovu od začátku. Napište pomocí cyklu program, který bude tento playlist „přehrávat“, kdy přehráním jedné skladby se rozumí výpis jejího názvu.

Předpokládejte, že proměnná *tokens* obsahuje text ve formě seznamu jednotlivých slov, respektive tokenů. Napište program, který vypíše veškeré hapaxy (tokeny, které se v textu vyskytují pouze jednou).