

Цикл for

У багатьох сучасних мовах програмування для задач обходу списків (та інших колекцій даних) використовується особлива семантика - цикл for each (в пер. з англ. "для кожного").

Цикл for можна використати для зміни значення змінної циклу в певних межах.

Простий приклад 1:

```
for x in range(1, 5):  
    print(x)
```

В даному прикладі змінна x буде змінюватися. При першому проході циклу (ще кажуть «при першій ітерації») значення змінної x буде дорівнювати одиниці. І тіло циклу, в якому print(x) виведе значення змінної. Після чого значення змінної зміниться на крок. Якщо крок не вказано, то значення змінної збільшиться на одиницю і стане дорівнювати двом. Далі буде проведена перевірка, чи число два входить в межі від нуля (включно) до п'яти (не включаючи п'ять). Число два входить в дані межі і тому знову виконається тіло циклу, в результаті виконання якого буде виведено на екран число два. І так далі, до чотирьох, тому що якщо записано range(1, 5), то число один входить в межі, а число п'ять – ні. Отже, при виконанні коду прикладу на екран буде виведено:

```
1  
2  
3  
4
```

Простий приклад 2:

```
for x in range(7):  
    print(x)
```

Якщо в параметрах range(), вказано лише одне число, то це кінцеве число діапазону значень. А початкове число – нуль. При виконанні коду прикладу на екран буде виведено:

```
0  
1  
2  
3  
4  
5  
6
```

Простий приклад 3:

```
for x in range(1, 7, 2):  
    print(x)
```

В прикладі вказано перше значення послідовності, останнє число послідовності і крок. При виконанні коду прикладу на екран буде виведено:

```
1  
3  
5
```

Простий приклад 4:

```
for x in range(9, 1, -2):  
    print (x)
```

В прикладі вказано перше значення послідовності, останнє число послідовності і крок. Крок в даному прикладі – від’ємний. При виконанні коду прикладу на екран буде виведено:

```
9  
7  
5  
3
```

Цикл for використовується і для опрацювання списків:

```
data = [5,4,8,23,12,456,74,721]  
for x in data:  
    print (x)
```

Оператор **for** у Python дещо відрізняється від C або Pascal, оператор **for** у мові Python перебирає члени будь-якої послідовності (списку чи рядка) у порядку їхнього в ній розташування. Наприклад:

```
# Довжини рядків:  
a = ['кіт', 'вікно', 'жбурляти']  
for x in a:  
    print x, len(x)
```

Результат:
кіт 3
вікно 5
жбурляти 8

При використанні циклу при роботі зі списками, можна використовувати функції `range()` та `len()` таким чином:

```
data = ['У', 'Марічки', 'є', 'ягнятко']  
for i in range(len(data)):  
    print(i, a[i])
```

Результат:
0 У
1 Марічки
2 є
3 ягнятко

Але більш доречною буде функція `enumerate()` - функція-генератор, що повертає пари (індекс, елемент):

```
data = ['У', 'Марічки', 'є', 'ягнятко']
for i, x in enumerate(data):
    print(i, x)
```

Результат:

```
0 У
1 Марічки
2 є
3 ягнятко
```

Ще один приклад використання функції `enumerate()` -

```
data = [5,4,8,23,12,456,74,721]
for i, x in enumerate(data):
    print(i, x)
```

Результат:

```
0 5
1 4
2 8
3 23
4 12
5 456
6 74
7 721
```

На випадок, якщо потрібно почати нумерацію ні з 0, а з іншого числа, у функції `enumerate()` є опціональний параметр `start`:

```
data = ['У', 'Марічки', 'є', 'ягнятко']
for i, x in enumerate(data, 1):
    print(i, x)
```

Результат:

```
1 У
2 Марічки
3 є
4 ягнятко
```

Інструкція `else`, застосована в циклі `for` (або `while`), перевіряє, чи був проведений вихід з циклу інструкцією `break`, або ж "природним" чином. Блок інструкцій всередині `else` виконається тільки в тому випадку, якщо вихід з циклу стався без допомоги `break`. Приклад, що це ілюструє:

Задача «Мороз на вулиці»:

Умова: В одному вхідному рядку через пропуск записані значення температури повітря кожного ранку протягом місяця (числа можуть бути не цілими). Якщо протягом тижня був хоч один ранок з від'ємною температурою, вивести повідомлення «Був мороз», в іншому випадку вивести повідомлення «Не було морозу»

Варіант коду програми:

```
weather_temp = [float(x) for x in input().split()]
for x in weather_temp:
    if x < 0:
        print('Був мороз')
        break
else:
    print('Не було морозу')
```

