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"). У синтаксисі Python такий цикл записується так:

```
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

Якщо потрібно перебрати послідовність чисел, то тут стане в нагоді стандартна функція **range().** Вона створює список, що містить арифметичну прогресію:

```
>>> range (10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Задане останнє значення не є частиною створеного списку: **range (10)** створює список із десяти елементів з числами у списку від нуля до дев'яти. Можливо також задати початок списку іншим числом та вказати інший крок (навіть від'ємний):

```
>>> range (5, 10)
[5, 6, 7, 8, 9]

>>> range (0, 10, 3)
[0, 3, 6, 9]

>>> range (-10, -100, -30)
[-10, -40, -70]
```

Щоб перебрати індекси послідовності, 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

© 2021, Анатолій Анатолійович. При використанні посилання на pletyvo.in.ua є обов'язковим. for. стор. 1 з 2

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')
```

© 2021, Анатолій Анатолійович. При використанні посилання на pletyvo.in.ua є обов'язковим.