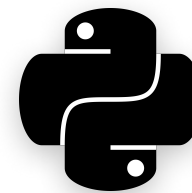


Введение в итераторы

Инструменты Python II



Дарья Козлова

Преподаватель информатики

Перебор с использованием цикла `for`

- Мы можем перебрать список используя цикл `for`.

```
employees = ['Михаил', 'Алексей', 'Дмитрий']  
for employee in employees:  
    print(employee)
```

```
Михаил  
Алексей  
Дмитрий
```

Перебор с использованием цикла for

- Мы можем перебрать строку используя цикл `for`.

```
for letter in 'DeepSkills':  
    print(letter)
```

```
D  
e  
e  
p  
S  
k  
i  
l  
l  
s
```

Перебор с использованием цикла for

- Мы можем перебрать объект range используя цикл for.

```
for i in range(4):  
    print(i)
```

```
0  
1  
2  
3
```

Итераторы и итерируемые объекты

- Итерируемые объекты
 - Списки, строки, словари, файлы.
 - Объект со связанным методом `iter()` .
 - Применение метода `iter()` к объекту создает итератор.
- Итератор
 - Создает следующие значения при помощи `next()` .

Перебор с использованием метода next()

```
word = 'Da'  
it = iter(word)  
next(it)
```

```
'D'
```

```
next(it)
```

```
'a'
```

Перебор с использованием метода `next()`

```
next(it)
```

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
StopIteration
```

Перебор с использованием оператора *

```
word = 'Data'  
it = iter(word)  
print(*it)
```

```
D a t a
```

```
print(*it)
```

- Отсутствуют значения для перебора.

Перебор словарей

```
pythonistas = {'Михаил': 'Омельченко', 'Иван': 'Герц'}  
for key, value in pythonistas.items():  
    print(key, value)
```

Михаил Омельченко

Иван Герц



Перебор файлов

```
file = open('file.txt')  
it = iter(file)  
print(next(it))
```

```
This is the first line.
```

```
print(next(it))
```

```
This is the second line.
```



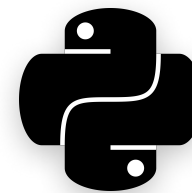
DeepSkills

Попрактикуемся!

Инструменты Python II

Дополнения к итераторам

Инструменты Python II



Дарья Козлова

Преподаватель информатики

Использование функции enumerate()

```
heros = ['Илья', 'Добрыня', 'Алёша', 'Дунай']  
e = enumerate(heros)  
print(type(e))
```

```
<class 'enumerate'>
```

```
e_list = list(e)  
print(e_list)
```

```
[(0, 'Илья'), (1, 'Добрыня'), (2, 'Алёша'), (3, 'Дунай')]
```

Распаковка функцией enumerate()

```
heros = ['Илья', 'Добрыня', 'Алёша', 'Дунай']  
for index, value in enumerate(heros):  
    print(index, value)
```

```
0 Илья  
1 Добрыня  
2 Алёша  
3 Дунай
```

Распаковка функцией enumerate()

```
for index, value in enumerate(heros, start=10):  
    print(index, value)
```

```
10 Илья  
11 Добрыня  
12 Алёша  
13 Дунай
```

Использование функции zip()

```
surnames = ['Муромец', 'Никитич', 'Попович', 'Иванович']  
names = ['Илья', 'Добрыня', 'Алёша', 'Дунай']  
z = zip(surnames, names)  
print(type(z))
```

```
<class 'zip'>
```

```
z_list = list(z)  
print(z_list)
```

```
[('Муромец', 'Илья'), ('Никитич', 'Добрыня'),  
(('Попович', 'Алёша'), ('Иванович', 'Дунай'))]
```


Распаковка функцией zip()

```
surnames = ['Муромец', 'Никитич', 'Попович', 'Иванович']  
names = ['Илья', 'Добрыня', 'Алёша', 'Дунай']  
for z1, z2 in zip(surnames, names):  
    print(z1, z2)
```

```
Муромец Илья  
Никитич Добрыня  
Попович Алёша  
Иванович Дунай
```

Вывод zip() с использованием *

```
surnames = ['Муромец', 'Никитич', 'Попович', 'Иванович']  
names = ['Илья', 'Добрыня', 'Алёша', 'Дунай']  
z = zip(surnames, names)  
print(*z)
```

```
('Муромец', 'Илья') ('Никитич', 'Добрыня')  
('Попович', 'Алёша') ('Иванович', 'Дунай')
```



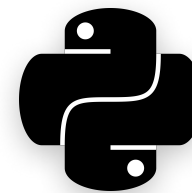
DeepSkills

Попрактикуемся!

Инструменты Python II

Итераторы для чтения больших файлов

Инструменты Python II



Дарья Козлова

Преподаватель информатики

Загрузка данных по частям

- Данные могут иметь очень большой размер для загрузки в память.
- Решение: загрузка данных по частям.
- Функция библиотеки pandas: `read_csv()` .
 - Указать размер загружаемой части: `chunk_size` .

Перебор данных по частям

```
import pandas as pd
result = []
for chunk in pd.read_csv('data.csv', chunksize=1000):
    result.append(sum(chunk['x']))
total = sum(result)
print(total)
```

```
5746925
```

```
for i in range(5):
    print(i)
```

Перебор данных по частям

```
import pandas as pd
total = 0
for chunk in pd.read_csv('data.csv', chunksize=1000):
    total += sum(chunk['x'])
print(total)
```

5746925



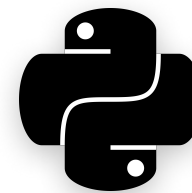
DeepSkills

Попрактикуемся!

Инструменты Python II

Поздравляем!

Инструменты Python II



Дарья Козлова

Преподаватель информатики



Что дальше?

- Генераторы списков и генераторы.
- Генераторы списков
 - Создание списков из списков, столбцов DataFrame и т.д.
 - Код в одну строку.
 - Более эффективные по сравнению с циклом `for`.



DeepSkills

Попрактикуемся!

Инструменты Python II