

# Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  + если все хорошо



ОНЛАЙН-ОБРАЗОВАНИЕ

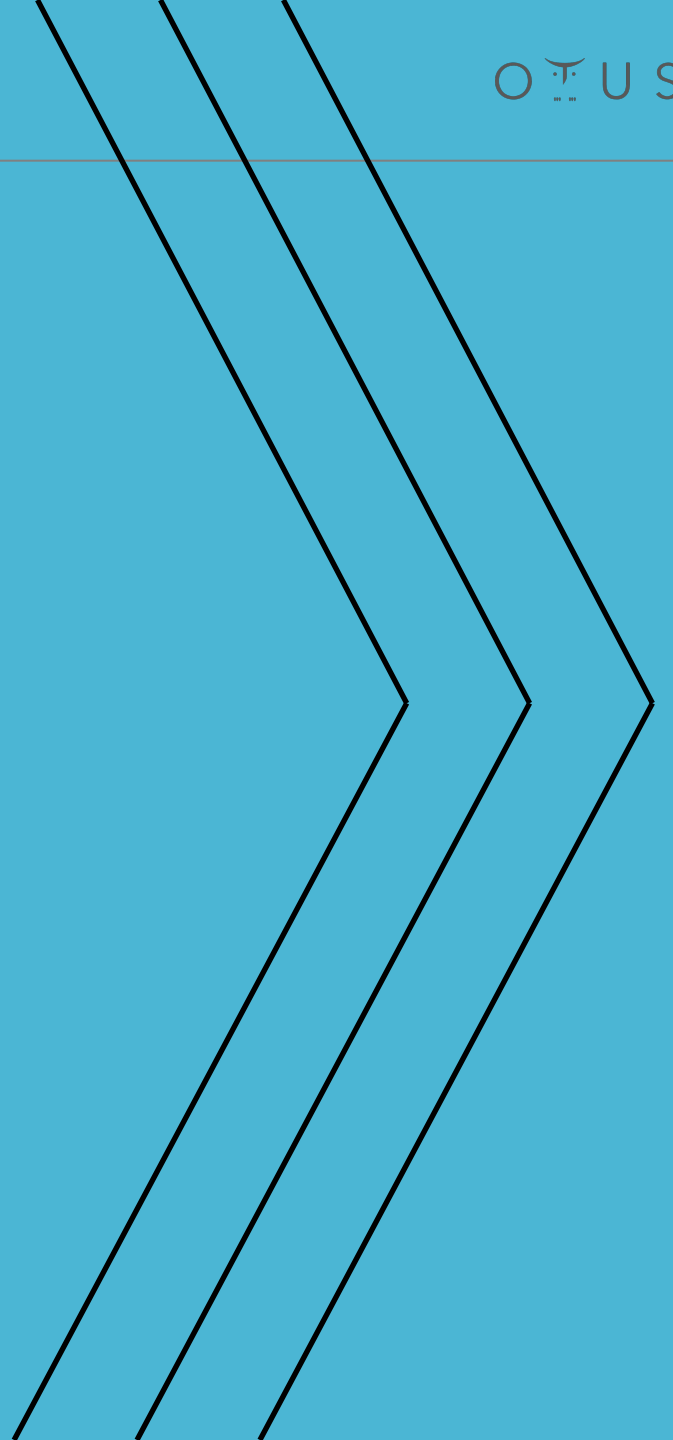
# DDT



- Looping Techniques
- Iterators
- Generators
- Работа с файлами
- Context managers

# 01

## Loops



- Контейнеры – это структуры данных, которые хранят элементы. Данные структуры данных хранятся в памяти и их содержимое тоже. Примеры контейнеров:
  - Списки, очереди
  - Множества
  - Словари
  - Строки
  - Кортежи

```
# Prints out 0,1,2,3,4 and then it prints
"count value reached 5"

count=0
while(count<5):
    print(count)
    count +=1
else:
    print("count value reached %d" %(count))

# Prints out 1,2,3,4
for i in range(1, 10):
    if(i%5==0):
        break
    print(i)
else:
    print("this is not printed because for
loop is terminated because of break but not
due to fail in condition")
```

Ключевые слова:

- else
- break
- continue
- pass

# 02

## Iterators, Generators



Что происходит когда мы итерируемся используя for:

- Вызывается метод `iter()` для получения итератора для “a”
- Вызывается метод `next()` для получения каждого элемента из итератора по очереди
- Выход из цикла, когда метод `next()` выбрасывает исключение `StopIteration`

```
>>> a = ['foo', 'bar', 'baz']
>>> for i in a:
...     print(i)
...
foo
bar
Baz
```

- Iterable – Объект, способный возвращать элементы по одному
  - у него есть `__iter__()` (iterator protocol) или `__getitem__()` (sequence protocol)
  - вызов `iter()` превращает в итератор
  - `for` вызывает `iter()` неявно, сохраняет итератор в безымянной переменной
- Iteration – итерирование, циклический процесс извлечения данных, объектов из коллекции (объекта итерации)
- Iterator – Объект, представляющий последовательность данных
  - Содержит `__iter__()`, который возвращает себя же
  - Iterator так же iterable
  - Завершается после `raise StopIteration`

a generator expression



is

a generator



always is

an iterator



next()

*lazily produce next value*

is



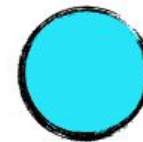
always is

iter()



(an) iterable

typically is



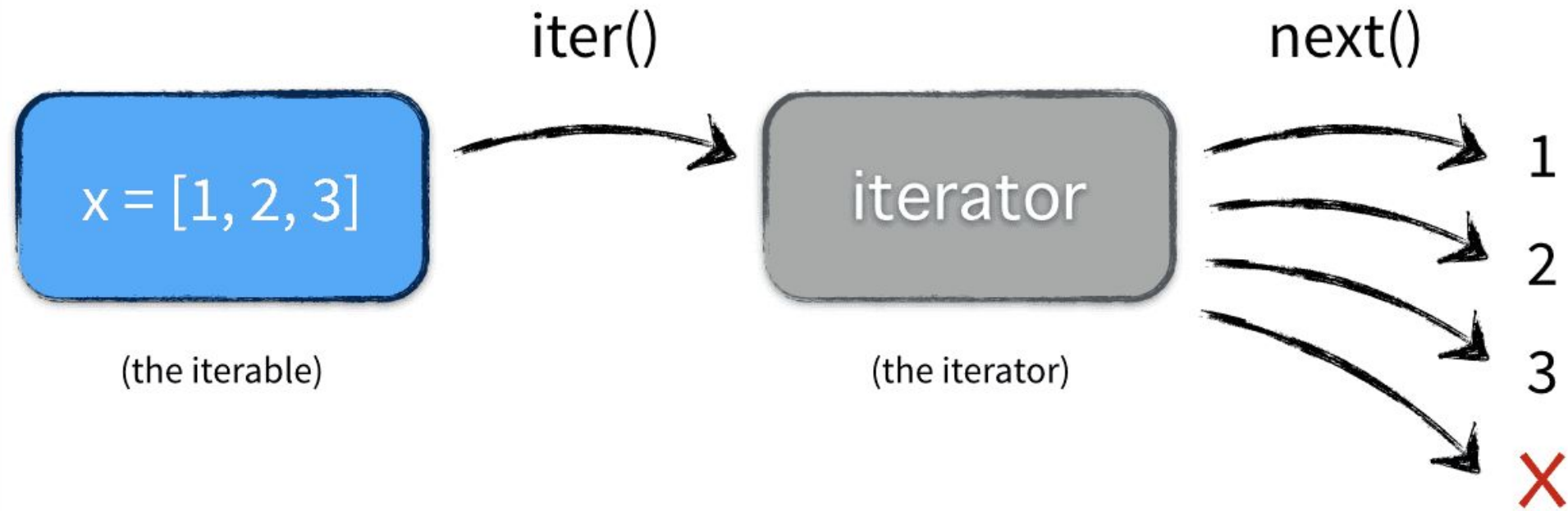
a container

produces



{list, set, dict} comprehension

a generator function



Функция, которая возвращает итератор:

- В ней есть `yield`
- После использования выкидывает `StopIteration` (т.е. использовать можно один раз)
- Может использовать `next()`
- Хранит промежуточные состояния между вызовами `yield`

- Если `yield` внутри функции, то она вернет генератор
- Функция будет вызвана при начале итерирования
- `Yield` возвращает результат как элемент генератора
- `Yield` останавливает функцию до следующей итерации (вызова метода `next()`)
- Функция продолжает работы с предыдущего `yield` (с предшествующего состояния)

# 03

## Работа с файлами

- Бинарные – любой другой файл, кроме текста, который может быть использован другим приложением. Другими словами бинарный файл нужен для других приложений.
- Текстовые – структурированная последовательность линий, каждая линия состоит из последовательности символов
  - Каждая линия заканчивается символом EOL – end of line. (не все редакторы его показывают, но он есть)
- Текстовый файл для людей, бинарный для приложений.



Режим	Обозначение
'r'	открытие на чтение (является значением по умолчанию).
'w'	открытие на запись, содержимое файла удаляется, если файла не существует, создается новый.
'x'	открытие на запись, если файла не существует, иначе исключение.
'a'	открытие на дозапись, информация добавляется в конец файла.
'b'	открытие в двоичном режиме.
't'	открытие в текстовом режиме (является значением по умолчанию).
'+'	открытие на чтение и запись

<https://habr.com/ru/post/132554/>

<https://realpython.com/python-for-loop/>

<https://anandology.com/python-practice-book/iterators.html>

<https://wiki.python.org/moin/Generators>

<https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python>

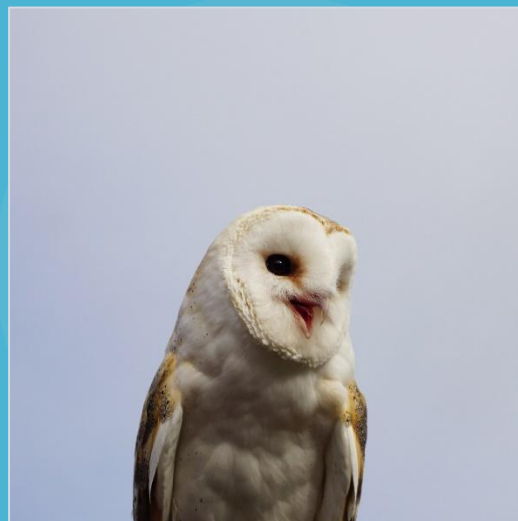
<https://nvie.com/posts/iterators-vs-generators/>

[www.dabeaz.com/generators-uk/](http://www.dabeaz.com/generators-uk/)

<http://www.dabeaz.com/generators/Generators.pdf>

<https://realpython.com/working-with-files-in-python/#traversing-directories-and-processing-files>

[https://www.w3schools.com/python/python\\_json.asp](https://www.w3schools.com/python/python_json.asp)



# Вяземский Семён

[svvyazemsky@gmail.com](mailto:svvyazemsky@gmail.com)

**Спасибо  
за внимание!**

