

# Итераторы и генераторы.

Итерируемые объекты — конечные  
(упорядоченные) коллекции данных.

С точки зрения Python, объект  $A$  итерируемый,  
если существует  $iter(A) \rightarrow A.__iter__()$ ,  
возвращающий итератор.

Итератор  $(it)$  — объект, из которого можно  
получить текущий элемент коллекции через  
вызов  $next(it) \rightarrow it.__next__()$ ,  
который в конце итерации вызывает  
исключение `StopIteration`.

Итерируемый объект может быть своим  
итератором, если `__iter__()` возвращает `self`,  
а внутри объекта существует `__next__()`.

Итераторы дают доступ только к элементу.

Генераторы — объекты, обладающие методом `--next--()`, но не привязанные к коллекции, а вычисляющие свое значение "на лету". Генераторы не обязаны быть конечными!!!

Пример конечного генератора — `range(x, y)`

```
(x**2 for x in range(100))
```

1-й способ — создать генератор

2-й способ — функция-генератор (`range`)

```
def func(...):
```

```
    ...  
    yield ...
```

на выходе `next()`

3-й способ — создавать объект-генератор вручную