

Теоретический материал к занятию Генераторы и итераторы 3 занятие.

В Python итерируемый объект — это объект, над которым производятся так называемые проходы (итерации). Например, как в цикле for.

Большинство типов, хранящих более одного значения являются итерируемыми объектами.

Как пример списки, кортежи, строки.

Итератор - это объект, который позволяет перемещаться (итерироваться) по элементам некоторой последовательности.

Генератор — тоже итератор, но его ключевое свойство — ленивые вычисления.

Г

енератор — это итератор, который работает в режиме обработки по необходимости.

Мы не будем производить вычисления и сохранять значения сразу, а сделаем их “на лету”, когда будут выполняться итерации.

Доступно два способа создания generator: выражение генератора и функция генератора.

Выражение-генератор похож на генератор списка, за исключением ().

```
gen1 = (x*x for x in range(10))
print(type(gen1))
print(next(gen1))
print(next(gen1))
```

Чтобы получить доступ к значениям, нужно сохранить его в переменной, а затем применить к этой переменной функцию next().

Для создания функции генератора в Python внутри функции вместо ключевого слова return используется ключевое слово yield.

```
def counter():
    i=1
    while(i<=10):
        yield i
        i+=1

for i in counter():
    print(i)
```

Мы создали генератор с именем counter() и назначили значение 1 локальной переменной i. Цикл while будет выполняться, пока i меньше или равно 10. Внутри цикла мы возвращаем (yield) значение i и увеличиваем его на единицу.

Генератор может содержать больше одного ключевого слова yield.

```
def my_gen(x):
    while( x> 0):
        if x%2==0:
            yield 'Четное'
        else:
```

```
        yield 'Нечетное'
    x-=1
for i in my_gen(7):
    print(i)
```

Если вы примените функцию `list()` к вызову генератора, она вернет список возвращенных генератором значений, в том порядке, в котором они возвращались. В следующем примере генератор возвращает квадраты чисел, если эти квадраты четные.

```
def even_squares(x):
    for i in range(x):
        if i**2%2==0:
            yield i**2

print(list(even_squares(10)))
```

Чтобы создать список из возвращаемых генератором значений, мы просто применяем функцию `list()` к вызову генератора, а не перебираем его через цикл.

Генератор в каждый отдельный момент удерживает только одно значение — то, которое он возвращает. Поэтому генераторы требуют куда меньше памяти, чем например список.