Generators (генераторы)

Генераторы в Python (если простыми словами) - некая конструкция, с помощью которой можно пройтись по заданному списку и выдать обработанные элементы списка. Вот несколько примеров:

- Список с путями к файлам. Пройтись по нему с помощью генератора и обработать только те пути, которые указывают на существующие файлы
- Пройтись по нодам Maya, и выдать только те, у которых есть определенное имя и тип
- Пройтись по всем костям персонажа, но обработать только те, у которых трансформы установлены в ноль.
- и т.д.

Генератор необязательно только фильтрует данные списка. Он может пройтись по каждому элементы списка, применить к нему какую-то операцию, и выдать результат. Например:

- Список с путями файлов. Пройтись по этому списку, если файл в реальности не существует - создать его. И все это делает генератор.

Как выглядит генератор?

```
def positiveNumbers(lst):
	for i in lst:
		if i > 0:
			yield i

a = [1,2,-4,-5,6,7,-2,9,-1]

for i in positiveNumbers(a):
	print (i)
```

Ключевой фишкой генератора (именно с помощью нее мы узнаем что данная функция - генератор) - это команда yield. Именно она "выплевывает" очередной обработанный элемент в цикл.

В чем его отличие от обычного обхода списка в цикле и проверки числа в теле цикла ?

- Во-первых, генератор эстетически сокращает и улучшает восприятие кода.
- Во-вторых, когда мы проходимся по списку - весь список разом грузится в память. Т.е. если у нас список из миллиона элементов - вся эта штука займет оперативную память, и только потом цикл начнет по ней проходить. Генератор же загружает в память каждый элемент списка отдельно, заменяя его следующим, и затем следующим, выдавая нам только обработанные значения. Генератор называют ленивым списком, т.к. который грузится в память по-элеметно, а не всё разом.

Теперь рассмотрим пример, в котором генератор пройдется по всем элементами обычного списка (которые являются transform объектами Майской сцены), и допустит к дальнейшим процедурам только те объекты, которые являются кривыми и у которых в имени есть слово *ctrl* и вконце стоит порядковый номер. Например кривая body_offset_ctrl_01 должна допуститься к обработке. Мы также воспользуемся регулярными выражениями.

```
import maya.cmds as cmds
import re
CtrlExp = re.compile(".*(ctrl)+.*[0-9]+\Z")

def controlCurves(n):
    for i in n:
        childNode = cmds.listRelatives(i, f=1)
        if not cmds.nodeType(childNode) == "nurbsCurve":
            continue

        if CtrlExp.match(i):
            yield i

listOfObjects = cmds.ls(type="transform")

for i in controlCurves(listOfObjects):
    print i
```

Мы можем вручную пройтись по результату генератора, используя функцию next()

```
def yrange(n):
    i = 1
    while i < n:
        if i%3 == 0:
            yield i
        i += 1

a = yrange(150)

print a.next()
```