Iterators

In [5]:
```
import sys

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # whole list stored in memory
y = map(lambda i: i**2, x)

print(next(y))
print(next(y))
print(next(y))

print("For loop starts:")

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

```
1
4
9
For loop starts:
16
25
36
49
64
81
100
```

In [6]:
```
print(f"Size of the list: {sys.getsizeof(x)}")
print(f"Size of the iterator: {sys.getsizeof(y)}")
```

```
Size of the list: 152
Size of the iterator: 48
```

In [7]:
```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # whole list stored in memory
y = map(lambda i: i**2, x)

while True:
    try:
        value = next(y)
        print(value)
    except StopIteration:
        print('Done')
        break
```

```
1
4
9
16
25
36
49
64
81
100
Done
```

Generators

Basics

In [10]:
```
def gen(n):
    for i in range(n):
        yield i

x = gen(5)
print(next(x))
print(next(x))
print(next(x))

print("For loop starts")
for i in x:
    print(i)
```

```
0
1
2
For loop starts
3
4
```

Use Cases

Let's say you have a file which is gigabtyes or maybe terrabytes in size. And you want to find a particular word in that file. It will be extremely inefficient in memory if you want to read the entire file at once. So, instead you can use a generator to read one line of the file at a time and search for the word in that line only.

In [14]:
```
def reader(file_name):
    for row in open(file_name, 'r'):
        yield row

x = reader("sample.txt") # This file may be a huge file
for row in x:
    if "oop" in row:
        print("Found")
        break
```

```
Found
```

In [ ]:
```

```