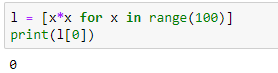
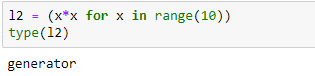
**Generators**

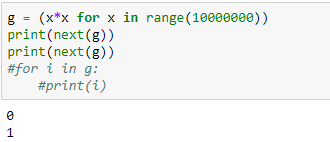
**1.Memory Utilization**





In above cell in the line one will get memory error because when we execute it will create all objects and store in the memory at a time. To overcome it we introduce the generator concept





**diff b/w collections and generators**

* collections the values are always going to stored in the memory .
* if you want to generate big sequence of values then go for generator in case of generator values never going to store in memory

**Generators?**

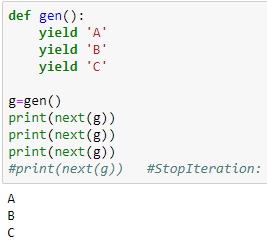
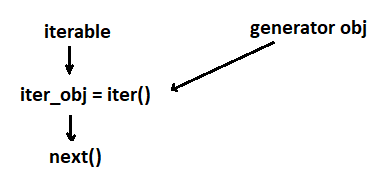
* it is a function which is responsible to generate a sequence of values(it is a function that returns an object which can iterate over as one value at a time)
* we can write generator function just like ordinary functions but it uses yield keyword to return values
* if a function contain at least one yield statement it becomes a generator function

**Why generators**

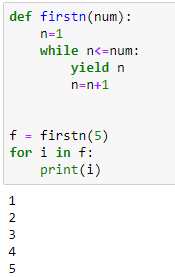
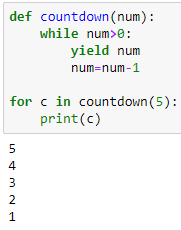
* to handle large volumes of data
* for memory utilization

**diff b/w return and yield**

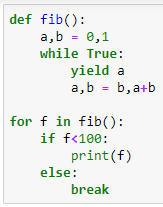
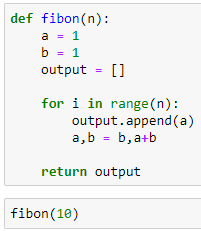
* return statement will terminate the entire function
* yield pauses the function and saving all its states and later continues from there on successive calls



**2.generate first n numbers**



**3.To generate Fibonacci numbers 0,1,1,2,3,5,8,13,21**



**Advantages**

* when compared with class level iterators, generators are very easy to use
* improves memory utilization and performance
* generators are best suitable for reading data from large number of large files
* generators work great for web scraping and crawling