Iterators and Generators

- Iterators allow us to iterate through a sequence of element I.e, visiting each element once in a sequence
- We can pass any sequence to the iterator like list , string , tuple , set and dictionary
- In python there is a built in function called iter for iteration

```
L = [1, 2, 3, 4, 5]

itr = iter(L)

print(next(itr))
print(next(itr))
print(next(itr))
print(next(itr))
print(next(itr))
print(next(itr))
```

#next(itr) give next element in sequence

```
1
2
3
4
5
```

- We can write our own iterators which are called “GENERATORS” they work just like iterators

- Lets see this with an example which displays days of a week and once the sequence is completed and called again it should return the first value and so on…

```
def Days():
    L = ['Sun', 'Mon', 'Tue', 'Wed', 'Thru', 'Fri', 'Sat']
    i = 0

    while True:
        x = L[i]
        i= (i + 1) % 7
        yield x

d =Days()
print(next(d))
print(next(d))
print(next(d))
print(next(d))
print(next(d))
print(next(d))
print(next(d))
print(next(d))
```

For a loop to remain in same state we use **yield** it'll not stop the function but keep function on hold and return value