iterable

The object which implements the __iter__() method is called iterable.

```
lst = [10,20,30,40,50]
print(dir(lst))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',
'__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__',
'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
```

Get Iterator Object

```
lst = [10,20,30,40,50]
print(dir(lst))
result = lst.__iter__()
print(result)
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',
'__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__',
'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
<list_iterator object at 0x000000A7625FB640>
```

Iterator
Iterator is an object that can return data one at a time while iterating over it

If an object to be an iterator, it must implement two methods

Iter()
next()

We can use __iter__() method for iterator object, for next iterations use __next__()

```
lst = [10,20,30,40,50]
result = lst.__iter__()
element1 = result.__next__()
print(element1) # 10
element2 = result.__next__()
print(element2) # 20
element3 = result.__next__()
print(element3) # 30
```

We can use iter() method for iterator object, for next iterations use next(), if there are no iterations we get exception as StopIteration

```
lst = [10,20,30,40,50]
result = iter(lst)
element1 = next(result)
print(element1) # 10
element2 = next(result)
print(element2) # 20
element3 = next(result)
print(element3) # 30
element4 = next(result)
print(element4) # 40
element5 = next(result)
print(element5) # 50
element6 = next(result)
print(element6) # StopIteration
```

```
Traceback (most recent call last):
  File "E:\Python Github\PythonMyWorkSpace\21_Iterators\Eg3.py", line 13, in
<module>
    element6 = next(result)
StopIteration
```

Internally how StopIteration works

```
lst = [10,20,30,40,50]
result = iter(lst)

while True:
   try:
      element = next(result)
      print(element, end=' ')
   except StopIteration:
      break
```

```
10 20 30 40 50
```