**Iterator in python :**

Take a list and print the value using index value 0 and 3(last value)

Graphical user interface, text, application, email

Description automatically generated

Let’s print the value which is out of the range and observe the error

Graphical user interface, text, application

Description automatically generated

Let’s do it with a loop as shown below.

Graphical user interface, text, application

Description automatically generated

* Let’s think about it, do you think that we have something behind the scenes can work as iterator. Consider it variable and iter is a inbuilt function and print(it) which will result in printing the object as shown below.

Graphical user interface, text, application

Description automatically generated

Now if we want to print the value we can say print(it.\_\_next\_\_()) it will print the first value of the list as shown in figure.

Graphical user interface, text, application, email

Description automatically generated

If we repeat the same line again, it will print the second value as shown below. From it class it is calling \_\_next\_\_ method which prints the first value and it remembers the i for the first value and the second next method will print the second value and so on. This is how iterator is used.

Graphical user interface, text, application

Description automatically generated

We can print the values with next function as well as shown below and using for loop

Graphical user interface, text, application

Description automatically generated

Q: What if you want to create your own objects, the objects we are using till now are inbuilt objects. What if we want to create our own iterator.

Let’s create a new class as we need to create an object. Create a class to print ***topten*** values one by one. ***Self.num = 1***# is a counter variable

To create our own iterator we need two methods 1) ***iter*** method #which will give the object or iterator

2)***next***() # which will give the next value or the next object.

If we print the values it will not stop printing the values.

Graphical user interface, text, application

Description automatically generated

Let’s understand what is happening here. Troubleshoot the issue and print the below code we will get the value of 1

Graphical user interface, text, application

Description automatically generated

Similar for 2 values as shown below.

Graphical user interface, text, application

Description automatically generated

The problem is loop will go from start to end and till now we are assuming the end would be ten, nowhere we had mentioned that we want to stop at 10 .

Next method is where we are incrementing so let’s make changes to the next method as shown below. Else condition is where we are raising an exception to stop the loop once we are done with the 10 values and it will stop the loop, which means if we don’t mention the exception it will continue iterating.

Graphical user interface, application

Description automatically generated

class Topten:

    def \_\_init\_\_(self):

        self.num = 1

    def \_\_iter\_\_(self):

        return self

    def \_\_next\_\_(self):

        if self.num <=10:

            val = self.num

            self.num += 1

            return val

        else:

            raise StopIteration

values = Topten()

for i in values:

    print(i)

Conclusion: the only two functions required to create the iterator is next() and iter(). Iter() will give you the object of iterator and next() will give you the next object.

Q: what will happen if we run the below code

Graphical user interface, text, application

Description automatically generated