Iterator and Generator

Mohammad Tasin (Tasin Coder)

Agenda

- What is Iterator
- iter() Function
- next() Function
- StopIteration error
- What is Generator

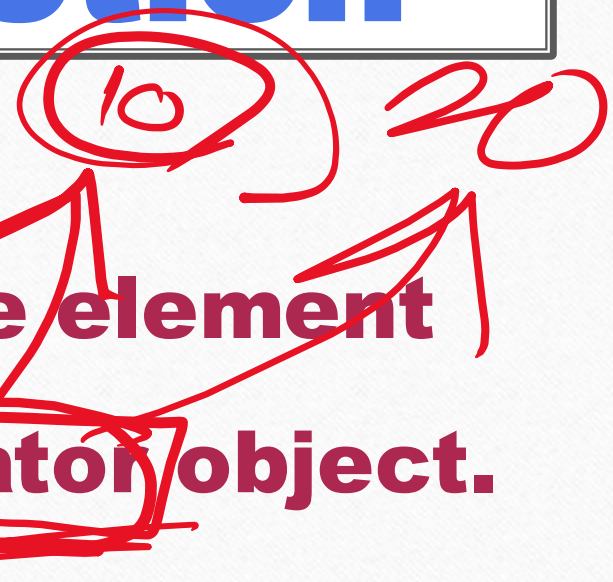
What is Iterator

- An iterator can be seen as a pointer to a container.
- **Iterator** वह है जो किसी भी **container** के **first element** को **point** करता हैं
- The iterator is an abstraction, which enables the programmer to access all the elements of a container, without any deeper knowledge of the data structure of this container object.
- Iterator is implicitly available in the for loop.
- iterator object can only access elements in a sequence from first element to last element.

iter() Function

- iter() is a Function
- iter() में एक argument pass करते है जो की iterable object होना चाहिए और return करता है iterator object जो की point करता है iterable का first element को,
- Syntax :- it = iter(iterable object)

next() Function



- next() is a Function
- next() method returns the element pointed by specified iterator object.
- next() method advances iterator object so that it can point to the next element of the container object.
- Syntax :-- e = next(iterator object)

StopIteration error

- When next() method is called for an iterator object which surpasses the last element of the container object, is produces **StopIteration** exception.

What is Generator

- Generator are special kind of function.
- A generator – Function is defined like a normal function, but whenever it needs to generate a value, it dose so with the yield Keyword rather then return.
- If the body of a def contains yield, the function automatically becomes a generator function.
- Python allows the use of return in generator.
- The return statement in generator is equivalent to raise StopIteration

Generator Code

```
def Fibonacci(n):
    a, b = 0,1
    while n :
        yield a
        a, b = b, a + b
it = Fibonacci(20)
while True:
    try:
        print(next(it),end=' ')
    except(StopIteration):
        print("Print all the elements of Generator")
        break
```