Iterators in Python

What Is Python Iterator?

An iterator is a container object that holds multiple values and provides a mechanism to traverse through them.

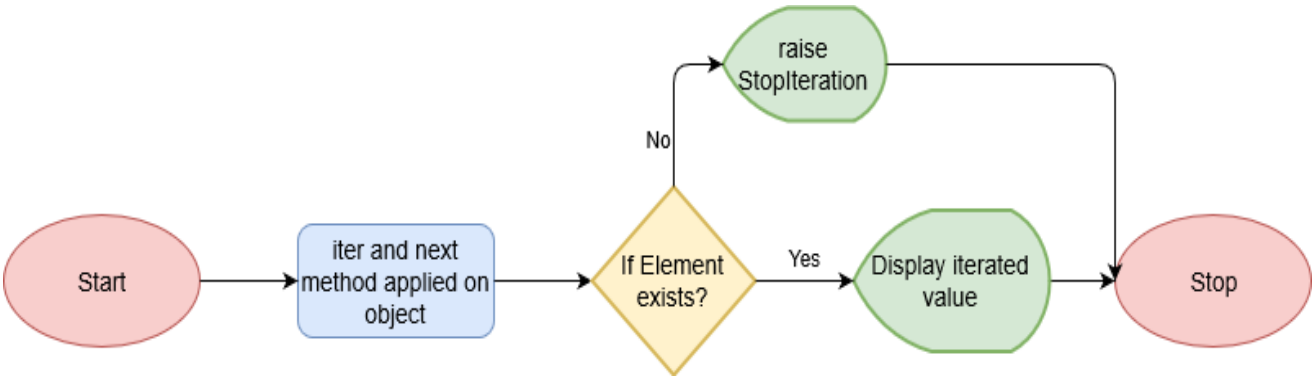
Examples of inbuilt iterators in Python are lists, tuples, etc.

Use them to implement “for” loops, generators, etc.

The Python iterator object implements Iterator Protocol.
It provides two methods, i.e., they are __iter__ and __next__.

__iter__() gets an iterable object whereas with the __next__ method to iterate over the object’s content.

How Does Iterator Work In Python?



Python Iterator Syntax

```
iterable_object = iter(my_object_to_iterate_through)
next(iterable_object)
```

Creating An Iterable Object Of Tuple:

```
Cubes = (1, 8, 27, 64, 125, 216)
cube = iter(Cubes)
print(next(cube))
print(next(cube))
```

#1 Output:

```
1
8
```

Iterating Through An Empty Object:

```
List = []
empty_element = iter(List)
print(next(empty_element))
print(next(empty_element))
```

#3 Output:

```
Traceback (most recent call last):
File "C:\Users\porting-
dev\AppData\Local\Programs\Python\Python35\test11.py", line 3, in
<module>
next(empty_element)
StopIteration
```

Printing A List Of Natural Numbers:

```
class natural_numbers:
    def __init__(self, max = 0):
        self.max = max
    def __iter__(self):
        self.number = 1
        return self

    def __next__(self):
        if self.max == self.number:
            raise StopIteration
        else:
            number = self.number
            self.number += 1
            return number

numbers = natural_numbers(10)
i = iter(numbers)
print("# Calling next() one by one:")
print(next(i))
print(next(i))
print("\n")
```

```
# Call next method in a loop
print("# Calling next() in a loop:")
for i in numbers:
    print(i)
```

To execute the above code use the command python3 /path_to_filename depending upon the default python version used.

#5 Output:

```
# Calling next() one by one:
1
2


# Calling next() in a loop:
1
2
3
4
5
6
7
8
9
```

We wish the above Python Iterator tutorial would have given you a fair idea of using them in real Python programs.