

الدرس دا هنتكلم فيه عن ال **Iterable and Iterator**، ال iterable من كل الدروس اللى فاتت وكلامنا عنه عارفين ان هو ال object اللى بيكون جواه عناصر ينفع يتعمل عليها iteration زي ال list وال string وال tuple وغيرهم.

ال iterator بقا هو ال method اللى بستخدمها علشان اعمل iteration على iterable، يعنى علشان اطبع العناصر اللى جوا ال iterable لازم احوله ل iterator عن طريق فانكشن **iter()**، وعلشان اطبع عنصر بستخدم فانكشن **next()** على ال iterator اللى عملته باستخدام **iter()**، ولما استخدم **next()** على نفس ال iterator تاني هطبعللى العنصر اللى بعده ولما استخدمها تالت هيطبع اللى بعده وهكذا لحد م ميكونش فى عناصر تانيه ويديني **StopIteration**.

طيب انا لما كنت احب اطبع عناصر ال iterable كنت بستخدم for loop وهى بتعدي على العناصر واحد واحد وتطبعهم مع انى مكنتش بحتاج احوله ل iterator، طيب ازاي بيعمل iteration؟ دا لأن ال for loop بيكون built-in فيها عملية تحويل ال iterable ل iterator وعمل ال iteration عليه باستخدام ال **next()**، كل دا بيكون خلف الكواليس فى ال for loop.

```
# -----
# -- Iterable vs Iterator --
# -----
# Iterable
# [1] Object Contains Data That Can Be Iterated Upon
# [2] Examples (String, List, Set, Tuple, Dictionary)
# -----
# Iterator
# [1] Object Used To Iterate Over Iterable Using next() Method Return 1 Element At A Time
# [2] You Can Generate Iterator From Iterable When Using iter() Method
# [3] For Loop Already Calls iter() Method on The Iterable Behind The Scene
# [4] Gives "StopIteration" If Theres No Next Element
# -----

myString = "Muhammad"

myList = [1, 2, 3, 4, 5]

for letter in myString:

    print(letter, end=" ")

print('\n')

for number in myList:

    print(number, end=" ")

print('\n')
```

```
myIterator = iter(myString)

print(next(myIterator))
print(next(myIterator))
print(next(myIterator))
print(next(myIterator))
print(next(myIterator))
print(next(myIterator))

print('\n')

for letter in iter("Scar"):

    print(letter, end=" ")
```

CODE

```
ocuments/Python Course/first.py"
M u h a m m a d

1 2 3 4 5

M
u
h
a
m
m

S c a r
```

OUTPUT