

Iterable & Iterator

```
# =====
# == iterable ==
# =====
# - object contains data that can be iterated upon
# - examples (string, list, tuple, set, dictionary)

# =====
# == iterator ==
# =====
# - object used to iterate over iterable using next() method
# - return 1 element at a time
# - you can generate iterator from iterable when using iter() method
# - for loop already calls iter() method on the iterable behind the scene
# - gives "StopIteration" if there no next element

# string is iterable
myString = "Osama"
for letter in myString:
    print(letter)

# list is iterable
myList = [1, 2, 3, 4]
for number in myList:
    print(number)

# numbers are not iterable
# will make error
# myNumber = 19
# for part in myNumber:
#     print(part)

# Boolean Values are not iterable
# will make error
# myBoolValue = True
# for b in myBoolValue:
#     print(b)

myIterator = iter(myString)
print(next(myIterator))    # O
print(next(myIterator))    # s
print(next(myIterator))    # a
print(next(myIterator))    # m
print(next(myIterator))    # a
```