Whenever we are working with data we need to modify data, filter data so we have in-build functions like **map() filter() reduce()**

map() function

The map function is used when you need to modify all elements with an iterables data

Syntax:

map(function, iterables)

Parameters:

function: The function to be called for each element of the specified iterable.

iterables: One or more iterables

Return Value:

When using map, it returns a map object, which is an iterator

Step 01

```
# Multiply Iterables
l1 = [10,20,30,40,50]
print(l1*2) # [10, 20, 30, 40, 50, 10, 20, 30, 40, 50]
```

```
# Multiply Iterables using for loop and lambda
lst = [10,20,30,40,50]
l = []
r = lambda n : n*2
for i in lst:
    l.append(r(i))
print(l) # [20, 40, 60, 80, 100]
```

```
# Multiply Iterables using Map Function
lst = [10,20,30,40,50]
def d1(n):
    return n*2
result = map(d1, lst)
print(result) # <map object at 0x000000D4731EB400>
print(list(result)) # [20, 40, 60, 80, 100]
```

```
# Multiply Iterables using Map Function with Lambda
lst = [10,20,30,40,50]
result = list(map(lambda x: x*2, lst))
print(result) # [20, 40, 60, 80, 100]
```

Step 02

```
# Multiply Sequence
l1 = [1,2,3,4]
l2 = [1,2,3,4]
print(l1*l2) # TypeError: can't multiply sequence by non-int of type 'list'
```

```
# Multiply Sequence using Map Function
l1 = [1,2,3,4,5]
l2 = [1,2,3,4,5]
def d1(a,b):
    return a*b
result = map(d1, l1,l2) # map(function, iterables)
print(result) # <map object at 0x00000033903CEDC0>
print(list(result)) # [1, 4, 9, 16, 25]
```

```
# Multiply sequence using lambda and map function
l1 = [1,2,3,4]
l2 = [1,2,3,4]
result = map(lambda a, b : a*b, l1, l2) # lambda args : expression, iterables
print(result) # <map object at 0x000000BC037DED60>
print(list(result)) # [1, 4, 9, 16]
```

Step 03

Case Study:

```
employes = [
   {"firstName":"Sai", "lastName":"Kiran", "age":27},
   {"firstName":"Pradeep", "lastName":"Reddy", "age":29},
   {"firstName":"Praneeth", "lastName":"Reddy", "age":35},
   {"firstName":"Ranjith", "lastName":"Yadav", "age":30}
]
print(employes)
print(employes[0]['firstName']+employes[0]['lastName'])
print(employes[1]['firstName']+employes[1]['lastName'])
print(employes[2]['firstName']+employes[2]['lastName'])

m = map(lambda x:x['firstName']+x['lastName'], employes)
print(list(m))
```

```
[{'firstName': 'Sai', 'lastName': 'Kiran', 'age': 27}, {'firstName': 'Pradeep', 'lastName': 'Reddy', 'age': 29}, {'firstName' : 'Praneeth', 'lastName': 'Reddy', 'age': 35}, {'firstName': 'Ranjith', 'lastName': 'Yadav', 'age': 30}]
SaiKiran
PradeepReddy
PraneethReddy
['SaiKiran', 'PradeepReddy', 'PraneethReddy', 'RanjithYadav']
```