Map:

Map function is used when you need to alter all items within an iterable data collection.

It takes two argumnets
map(func, *iterables) --> map object

Note: Map function, it takes a function and iterable datatype and returns filter object

```python
#map without lambda
def d1(a, b):
    return a+b

x = map(d1, (1,2,3,4), (1,2,3,4))
print(list(x))

Output
[2, 4, 6, 8]
```

```python
#map using lambda
lst = [1,2,3,4]
r = map(lambda a: a + a, lst)
print(list(r))

Output
[2, 4, 6, 8]
```

```python
#Find the length of Strings using map()
def d2(n):
    return len(n)

m = map(d2, ("Hari", "Manoj", "Vinod", "Jagadesh"))
print(list(m))
# print(tuple(m))
# print(set(m))

Output
[4, 5, 5, 8]
```

Not to explain

```python
#map()
students = ['1,Hari,Python',
            '2,Mahi,Python',
            '3,Mani,Java',
            '4,Sunny,Python'
]
print(students)

Ouput
['1,Hari,Python', '2,Mahi,Python', '3,Mani,Java',
'4,Sunny,Python']

#split(separator, maxsplit)
#map(func, *iterables) --> map object
x = map(lambda user: (int(user.split(',')[0]), user.split(',')[-1]),students)
y = list(x)
print(y)

Output
[(1, 'Python'), (2, 'Python'), (3, 'Java'), (4,
'Python')]
```