# Map Function

Data: a1, a2, a3 ... an

Function: *f*

# function -*f*- applies to each piece of data.

map(f, data)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

my\_list = [12, 13, 14, 15]

def square(elem):

return elem \*\* 2

res = map(square, my\_list)

print(list(res)) # [144, 169, 196, 225]

# or

res = [square(i) for i in my\_list]

print(res)

# [144, 169, 196, 225]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

country\_temp = [('Berlin', 29), ('PK', 56), ('Can', 19), ('Dubai', 50)]

# def to\_farenheit(data):

# return (data[0], 9/5\*data[1]+32)

to\_farenheit = lambda data: (data[0], 9/5\*data[1]+32)

res = map(to\_farenheit, country\_temp)

print(list(res))

# or

res = [to\_farenheit(i) for i in country\_temp]

print(res)

# [('Berlin', 84.2), ('PK', 132.8), ('Can', 66.2), ('Dubai', 122.0)]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Filter Function

import statistics

my\_list = [12, 20, 44, 77, 56, 88]

average\_val = statistics.mean(my\_list)

# print(average\_val) # 49.5

# Example #01: filter those pieces of list that are above average

res = filter(lambda a: a > average\_val, my\_list)

print(list(res))

# [77, 56, 88]

# or Example#01

res = [i for i in my\_list if i > average\_val]

print(res) # [77, 56, 88]

# Example#02: filter those pieces of list that are less than average

res = filter(lambda a: a < average\_val, my\_list)

print(list(res))

# [12, 20, 44]

# or Example#02

res = [i for i in my\_list if i < average\_val]

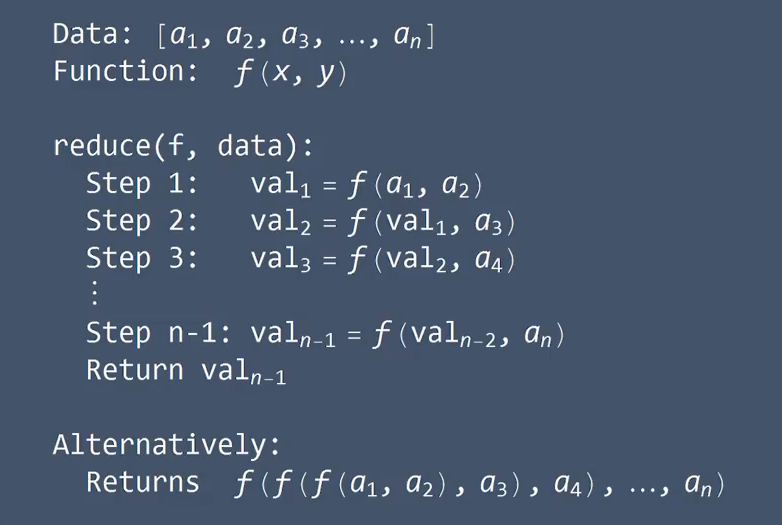
print(res) # [12, 20, 44]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Reduce Function

In python 3+, reduce is not a builtin function. It has been moved to the ‘functools’ module.

**Guido van Rossum**: ‘’Use functools.reduce() if you really need it; however, 99% of the time an explicit for loop is more readable.’’



import functools

my\_list = [12, 20, 44, 77, 56, 88]

# def func(x,y):

# return x \* y

func = lambda x, y: x \* y

res = functools.reduce(func, my\_list)

print(res)

# 4007055360

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

product = 1

for x in my\_list:

product = product \* x

print(product) # 4007055360

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_