* **REDUCE function:-**

**Syntax: reduce(function, iterable, initializer)**

The reduce() function in Python is a powerful function that applies a binary function (a function that takes two arguments) to an iterable in a cumulative way. This means it combines the elements of the iterable in a way that they are reduced to a single output value.

The reduce() function is part of the functools module, so you need to import it before you can use it.

**initializer**: This is optional and it is used as the initial value. If the initializer is present, it is placed before the items of the sequence in the calculation, and serves as a default when the iterable is empty.If the initializer is not provided, the first element of the iterable is used.

from functools import reduce

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

product = reduce((lambda x, y: x \* y), numbers)

print(product) # Output: 120

In this example, reduce() takes two arguments: a function and a list. The function is a lambda function that takes two arguments and returns their product. The reduce() function applies this lambda function to the first two elements of the list, then to the result and the next element, and so on, until it has gone through all the elements in the list. The result is the product of all the elements in the list.

The reduce() function can be very handy when you need to apply a function to an iterable and reduce it to a single cumulative value. Some common uses of reduce() include computing the product or sum of a list of numbers.

1. **Finding the Maximum or Minimum**: You can use reduce() to find the maximum or minimum in a list.

from functools import reduce

numbers = [4, 2, 9, 6, 5, 1, 8, 3, 7]

max\_value = reduce(lambda a, b: a if a > b else b, numbers)

min\_value = reduce(lambda a, b: a if a < b else b, numbers)

print(max\_value) # Output: 9

print(min\_value) # Output: 1

1. **Concatenating Strings**: You can use reduce() to concatenate a list of strings.

from functools import reduce

strings = ['Hello', ' ', 'World', '!']

result = reduce(lambda a, b: a + b, strings)

print(result) # Output: Hello World!

1. **Factorial Calculation**: You can use reduce() to calculate the factorial of a number.

from functools import reduce

n = 5 # you can change this value

factorial = reduce(lambda x, y: x \* y, range(1, n + 1))

print(factorial) # Output: 120

1. **Flattening a Nested List**: You can use reduce() to flatten a nested list.

from functools import reduce

nested\_list = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]

flattened\_list = reduce(lambda x, y: x + y, nested\_list)

print(flattened\_list) # Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]

**The function you pass to reduce() needs to take two arguments: the first is the accumulated value, and the second is the current item in the iteration. The function should return the new accumulated value.**

**5.GCD of a List of Numbers**: You can use reduce() to compute the Greatest Common Divisor (GCD) of a list of numbers.

from functools import reduce

import math

numbers = [48, 96, 36, 60]

gcd = reduce(math.gcd, numbers)

print(gcd) # Output: 12