→ reduce()

It works by calling function we passed for the first two items in the sequence. The result returned by the function is used in another call to function alongside with the next element. This process repeats until we have gone through all the elements in the sequence.

Syntax:

reduce (function, sequence [, initial])

→ zip()

This function is used to combine two or more lists into a single iterable, where elements from corresponding positions are paired together.

The resulting iterable contains tuples, where first element from each list is paired together, and second element from each list is paired together, and so on.

Syntax:

zip (*iterators)

→ id()

The id() function returns a unique id for the specified object. All objects in Python has its own unique id. The id is assigned to the object when it is created.

Syntax:

id(object)

→ Enumerate()

This function adds a counter to an iterable and returns the Enumerate object as the output

Syntax:

Enumerate (iterable, start = 0)

→ map()

The map() function iterates through all items in the given iterable and executes the function we passed as an argument on each of them.

Syntax:

map(function, iterable(s))

→ filter()

. It is similar to map(), it takes a function object and an iterable and creates a new list.

filter() forms a new list that contains only elements that satisfy a certain condition.

Syntax:

filter(function, iterable(s))

jupyter Untitled18 Last Checkpoint: 3 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [1]:
```
'''ENUMERATE FUNCTION'''
languages = ['Python', 'Java', 'JavaScript']
languages = enumerate(languages)
# convert enumerate object to list
print(list(languages))
```

```
[(0, 'Python'), (1, 'Java'), (2, 'JavaScript')]
```

In [2]:
```
'''MAP FUNCTION'''
def starts_with_A(s):
    return s[0] == "A"
fruit = ["Apple", "Banana", "Pear", "Apricot", "Orange"]
objects = map(starts_with_A, fruit)
print(list(objects))
```

```
[True, False, False, True, False]
```

In [3]:
```
'''FILTER FUNCTION'''
def starts_with_A(s):
    return s[0] == "A"
fruit = ["Apple", "Banana", "Pear", "Apricot", "Orange"]
objects = filter(starts_with_A, fruit)
print(list(objects))
```

```
['Apple', 'Apricot']
```

In [4]:
```
'''REDUCE FUNCTION'''
from functools import reduce

def add(x, y):
    return x + y

list = [1, 2, 3, 4]
print(reduce(add, list))
```

```
10
```

In [5]:
```
'''ZIP FUNCTION'''
name = [ "Man", "woman", "Sun", "rain" ]
nos = [ 4, 1, 3, 2 ]

# using zip() to map values
mapped = zip(name, nos)

print(set(mapped))
```

```
{('rain', 2), ('Man', 4), ('Sun', 3), ('woman', 1)}
```

In [6]:
```
'''ID FUNCTION'''
x = ('a', 'b', 'c')
y = id(x)
print(y)
```

```
1788293778496
```