1. What is the relationship between def statements and lambda expressions ?
Ans: def statement is used to define normal functions and lamba expressions are used to create anonymous functions that can be assigned to a variable and be called using the variable. Since lambda is an expression it is a single statement unlike the block of statements in the body of def statement.

2. What is the benefit of lambda?
Ans:

- Reduce the number of lines of code.
- Used when a function is needed temporarily
- Can be used inside another function.
- Can define a function and call it immediately at the end of definition.

3. Compare and contrast map, filter, and reduce.
Ans:
map(): takes a function and list of inputs or a list of functions and returns an output after applying the function on the list.
filter(): creates a list of elements for which a function returns true, resembles a for loop but it is faster.
reduce(): applies a rolling computation to sequential pairs of values in a list

In [6]:
```
import functools
# map function
print('Map:',list(map(lambda x:2*x, [1,2,3,4])))
# fitler function
print('Filter:',list(filter(lambda x:x%2 ==0, [1,2,3,4])))
# reduce function
print('Reduce:',functools.reduce(lambda x,y:x+y, [1,2,3,4,5,6]))
```

```
Map: [2, 4, 6, 8]
Filter: [2, 4]
Reduce: 21
```

4. What are function annotations, and how are they used?
Ans: Function annotations allow you to add arbitrary metadata to function parameters and return value.
They can be used as follows:
In case of simple parameters: def func(x: expression, y: expression = 20)
In case of excess parameters: def func (**args: expression, **kwargs: expression)

5. What are recursive functions, and how are they used?
Ans: A recursive function is a function that calls itself during its execution. The process may repeat several times.

In [9]:
```
def factorial(n):
    if n==1:
        return 1
    else:
        return (n*factorial(n-1))
print(factorial(5))
```

```
120
```

6. What are some general design guidelines for coding functions?
Ans:

- Use a docstring to explain the functionality

- avoid or limit using global variables
- Proper indentation
- Follow a naming convention throughout the code and use self-explanatory names
- Local variables format : localVariable, Global variables format : GlobalVariable
- Constant format: CONSTANT

7. Name three or more ways that functions can communicate results to a caller.
Ans: Function can communicate with the calling function is via print, return or yield.