Principles of FP

① Data should be protected from mutation (state change)

② Declarative style

③ FCC (First Class citizen)

Functions are treated as FCC

HOF and Decorators

HOF → return Funcs

Decorators → Takes f(x) as input and return new F'(x)

lambda Function

lamba input : Output

Map

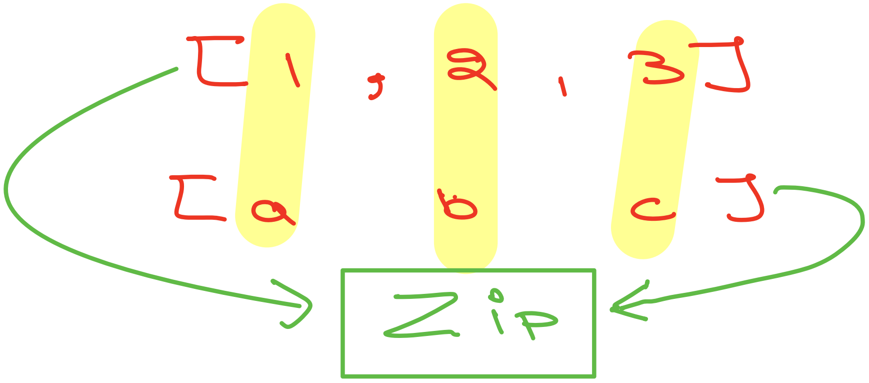
1) It is replacement of For Loops in Functional Programming

map(func , iterable)

Zip

① Takes two or more iterables and get zipped Output

② Output length == min(len-iterable)



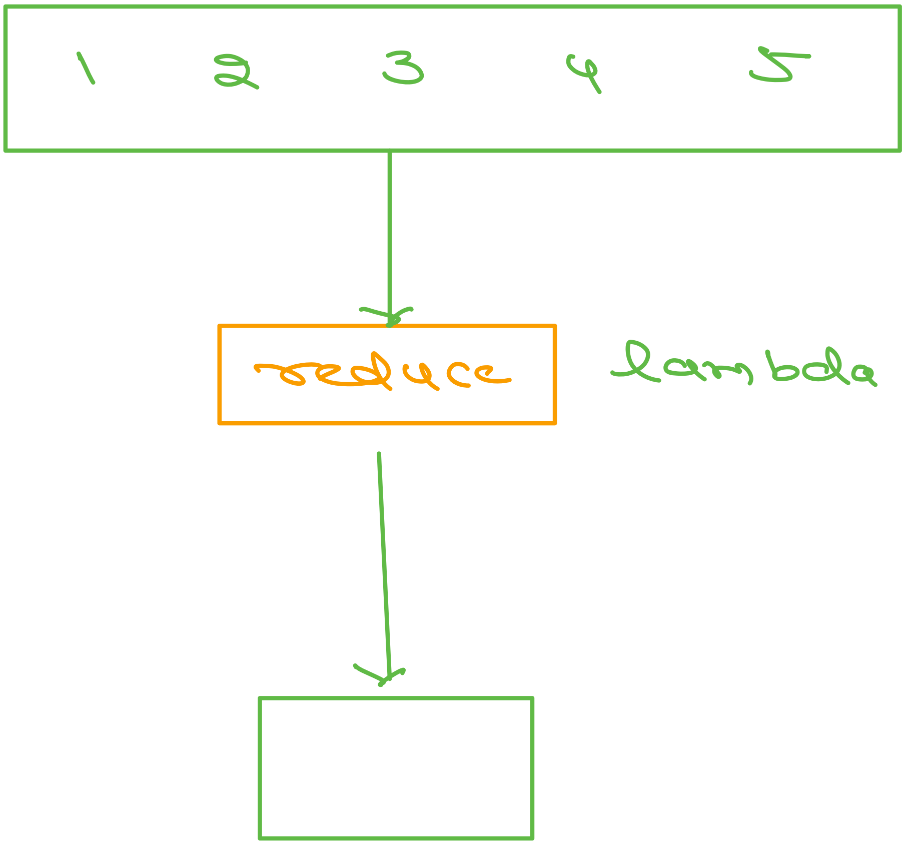
[(1 , a) (2, b) (3 , c)]

Reduce

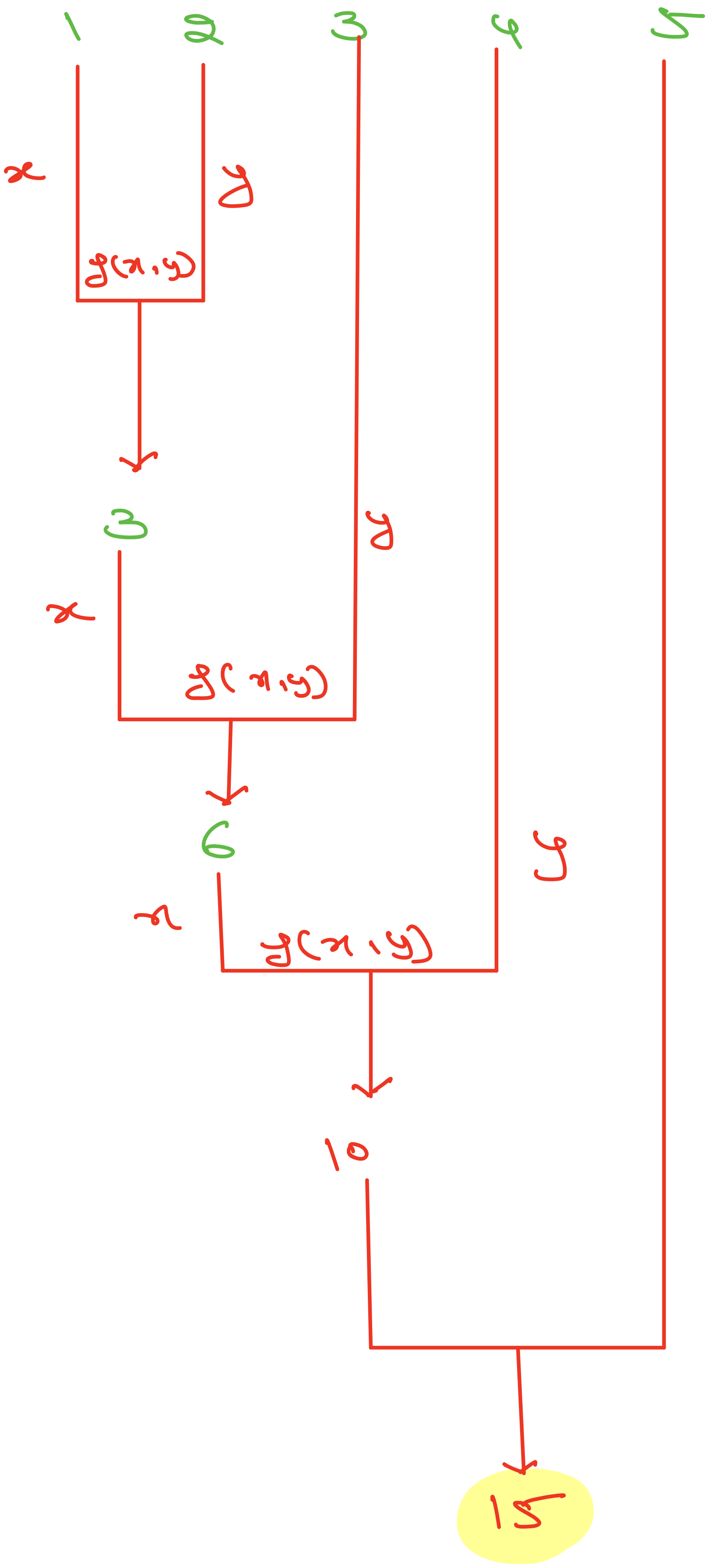
① Takes a Function which accepts atleast two arguments

② Using the Function reduces the iterable to single Value

③ x will be picked from list if there is no 3rd Arg



lambda x, y : x+y



lambda x, y : x+y

Positional, *args, Keword Args, **Kwargs
(default)