

Agenda

① Map

② Filter

③ Reduce

④ Zip

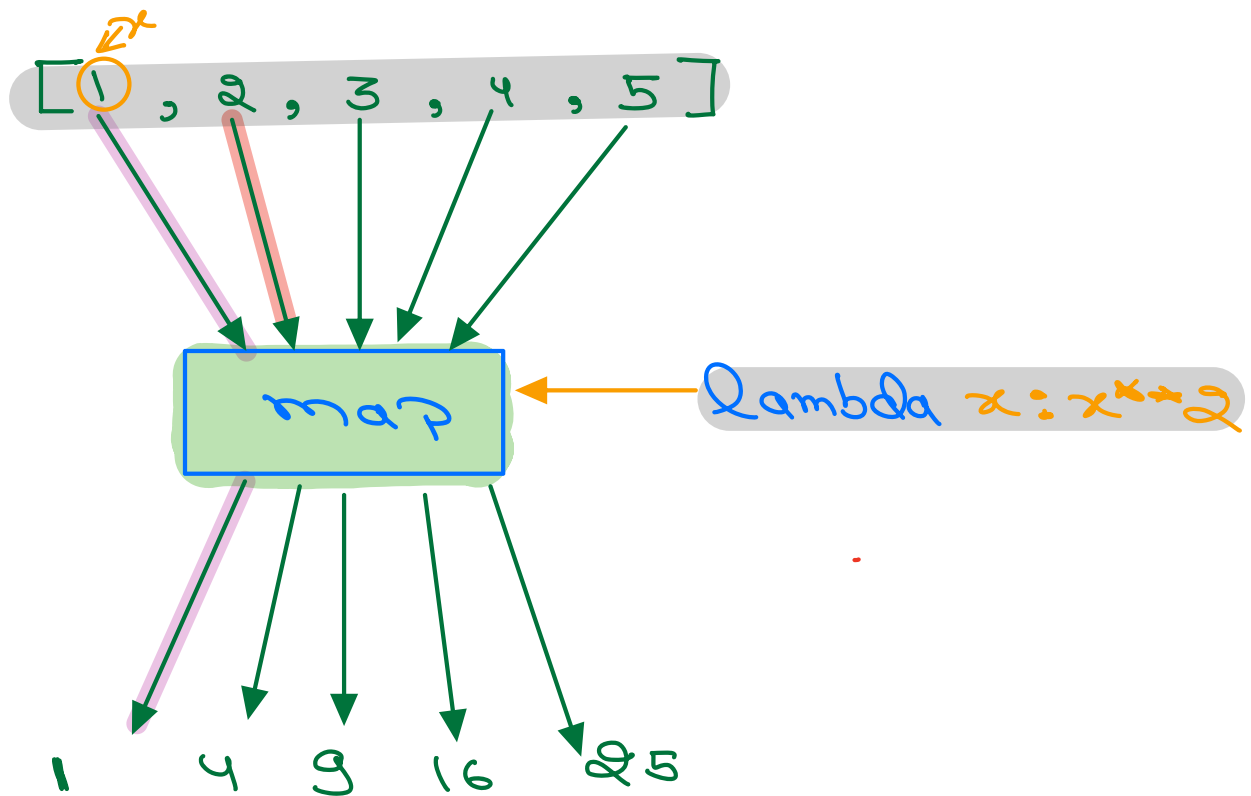
⑤ *args and **kwargs → Functions

MONDAY

Functional Programming

map

- * map is an 'alternative to for loop'
- * `map(function, *iterable)`
- * returns a 'map object of same size' as input iterable



Map applies function to every member of iterable

filter

- filter out element from iterable
- `filter(function, iterable)`
- function is evaluated on each member for boolean and True
- * equivalent values are added into Output
- returns filter object of $\leq \text{len}(it)$

[1, 2, 3, 4, 5]

filter

[2, 4]

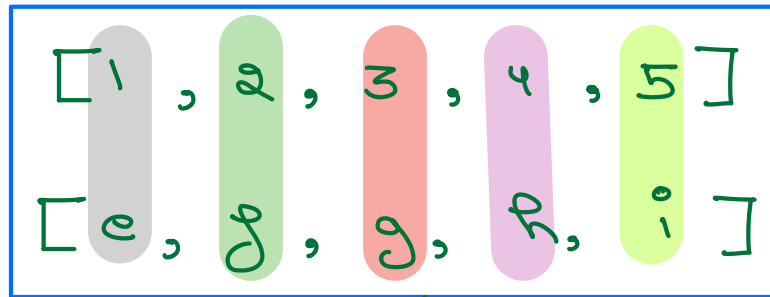
lambda $x: x \% 2 == 0$

lambda $x: x \times 2$ ②

[1, 2, 3, 4, 5]

Zip

- Zip : zips iterable w.r.t index
- Zip (iterable, *iterable)
- returns a zip object of size equal to smallest input iterable



Zip

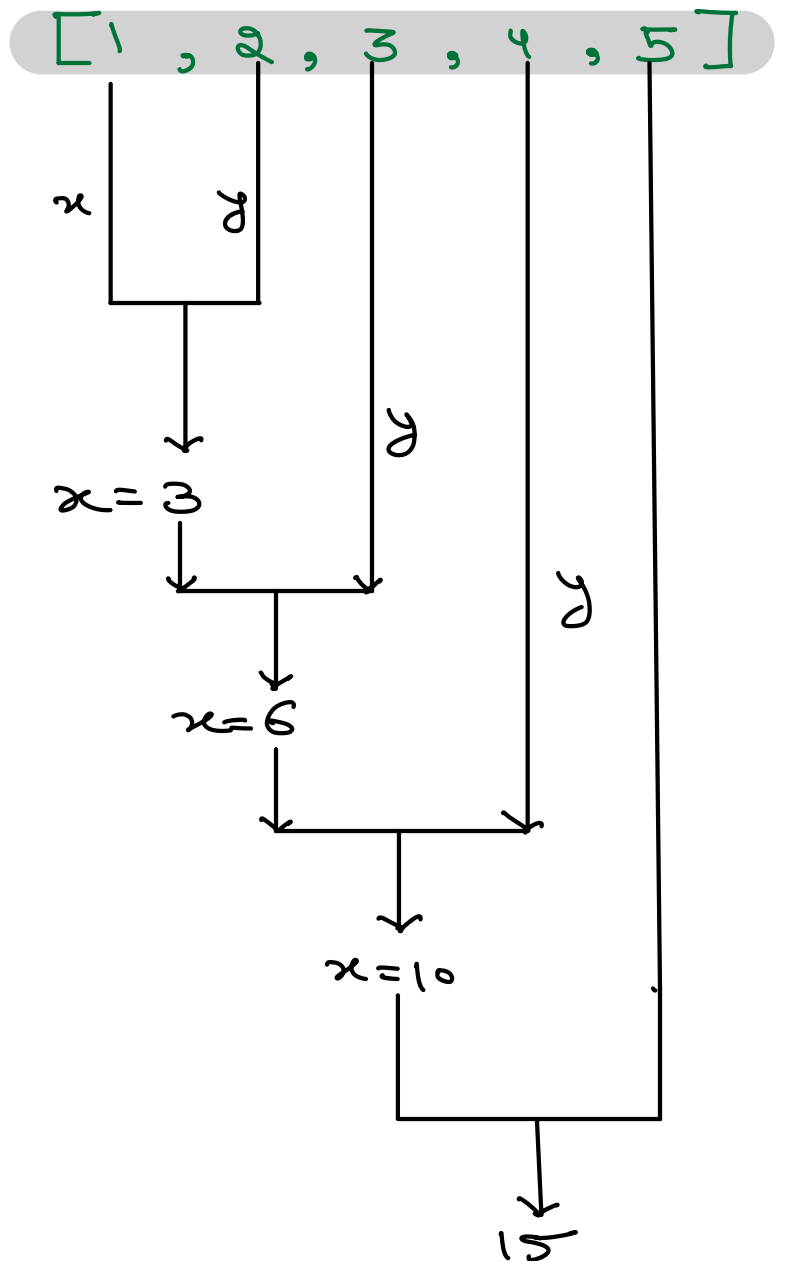
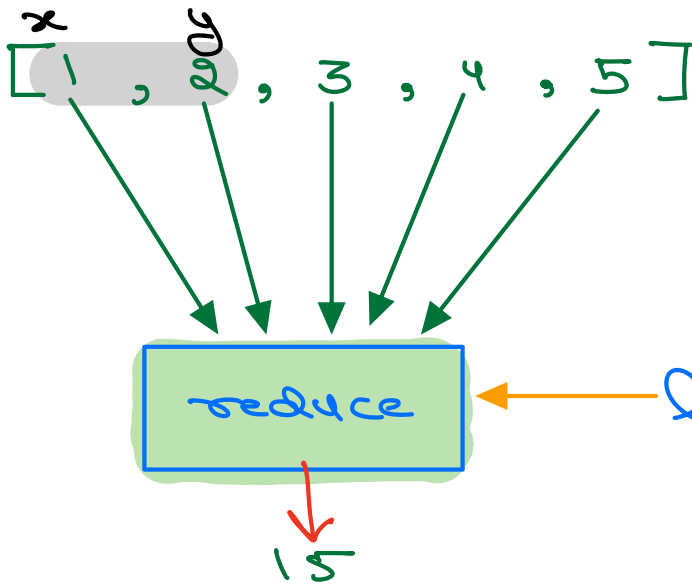
[(1, e), (2, g), (3, g), (4, h), (5, i)]

reduce

- It is used for reducing a iterable to a single value
- reduce (function, *iterable) "

6

Note : function in reduce must take two input arguments



`*args` and `**kwargs`

① Arbitrary Arguments = `*args`

② apart from mandatory arguments, all extra arguments will be stored as tuple inside any variable

③ Arbitrary keyword arguments = `**kwargs`

④ `**kwargs` : Passed as dictionary

Order of passing Arguments

Positional → `*args` → keyworded → `**kwargs`