Functional Programming

1) lambda function
2) Map
3) reduce
4) Filter
5) Zip

— x — x — x — x —— x —— x —— x ——

1) Arbitrary positional and Keyword Args (*args and **kwargs)

What is functional programming?

=> Programming paradigm what to do instead of How to do

1) Declarative programming

Why?

1) Concise and readable

2) Efficient

inspired from

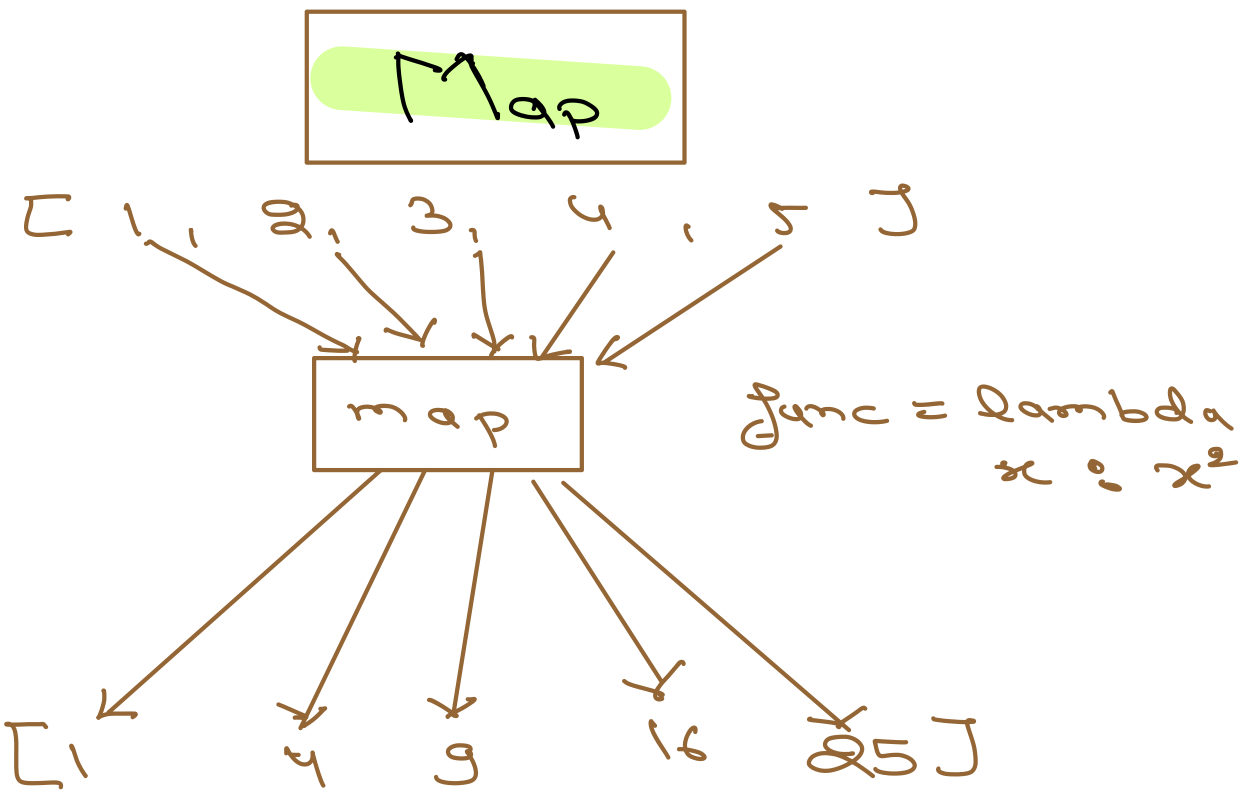
$\boxed{f(x) \longrightarrow y}$

lambda function

```
def    square(x) :
           out = x**2
           return out
```

```
square= lambda x : x**2
```

- also known as Anonymous functions
- It can take 0 or more arguments

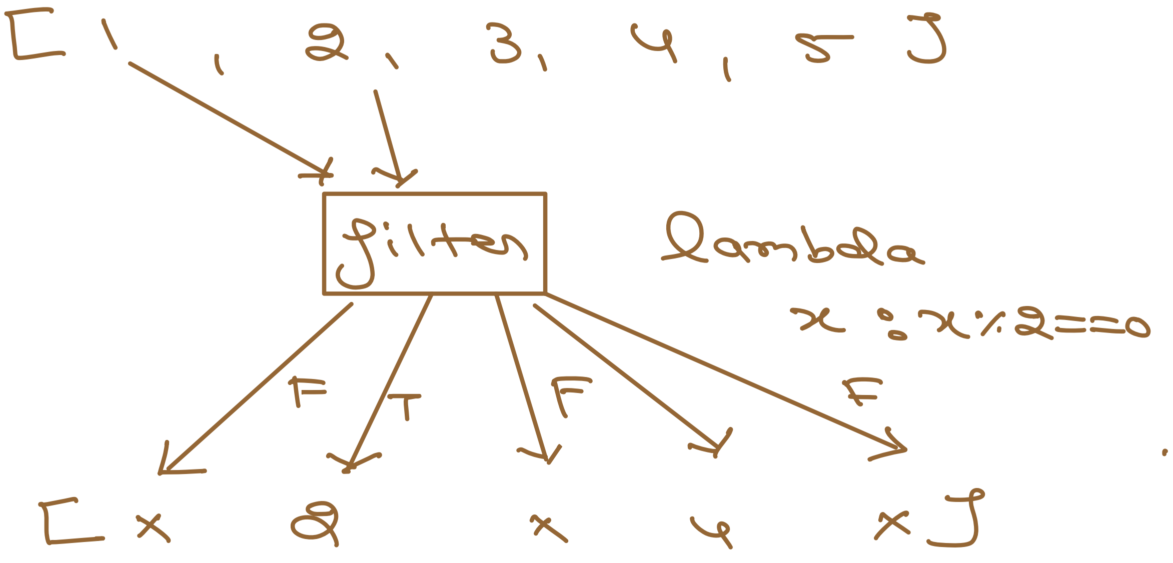


func = lambda x : x^2

- generator → iterable
- len(input) = len(output)

filter

- Filters elements from an iterable
- Takes function which returns True or False
- len(input) ≥ len(output)



size ≤ N

Zip

- Zips two or more iterables
- Output len depends on shortest iter

list1 = (1 , 2 , 3)

list2 = ('a' , 'b' , 'c' , 'd')

zip(list1 , list2)

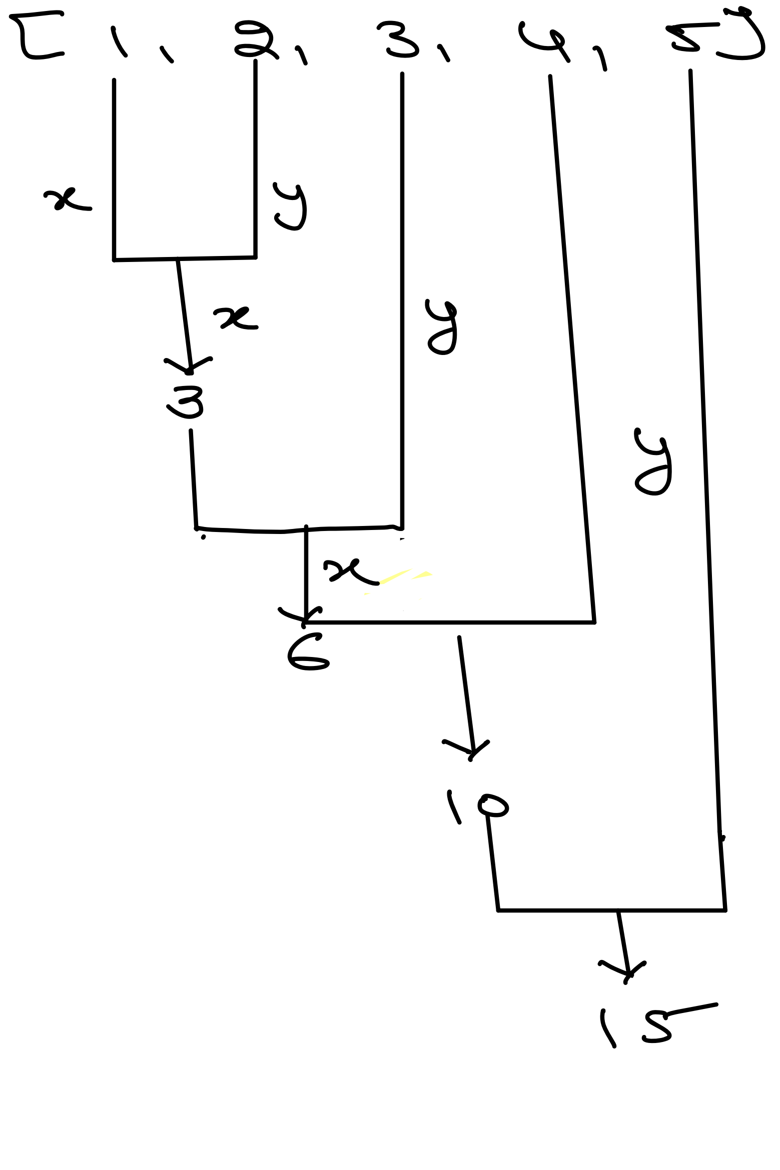
((1 , 'a') , (2 , 'b') , (3 , 'c'))

reduce

- Takes function with two args
- reduces iterable to a single value

[1, 2, 3, 4, 5]

$\boxed{\text{reduce}}$ lambda x, y : x+y



Abitrary Argument

* positional → *args → tuple
* Keyword → **kwargs → Dict