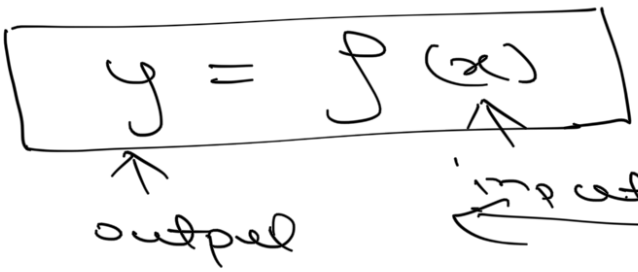
Functional Programming : Part 1

- Some revision Quizzes on OOP
- Functional Programming
  - lambda Function
  - Map
  - Reduce
  - Filter
  - Higher order Function
    - Decorator

- What is Functional Programming



- mathematical Notation
- avoid changing State of data

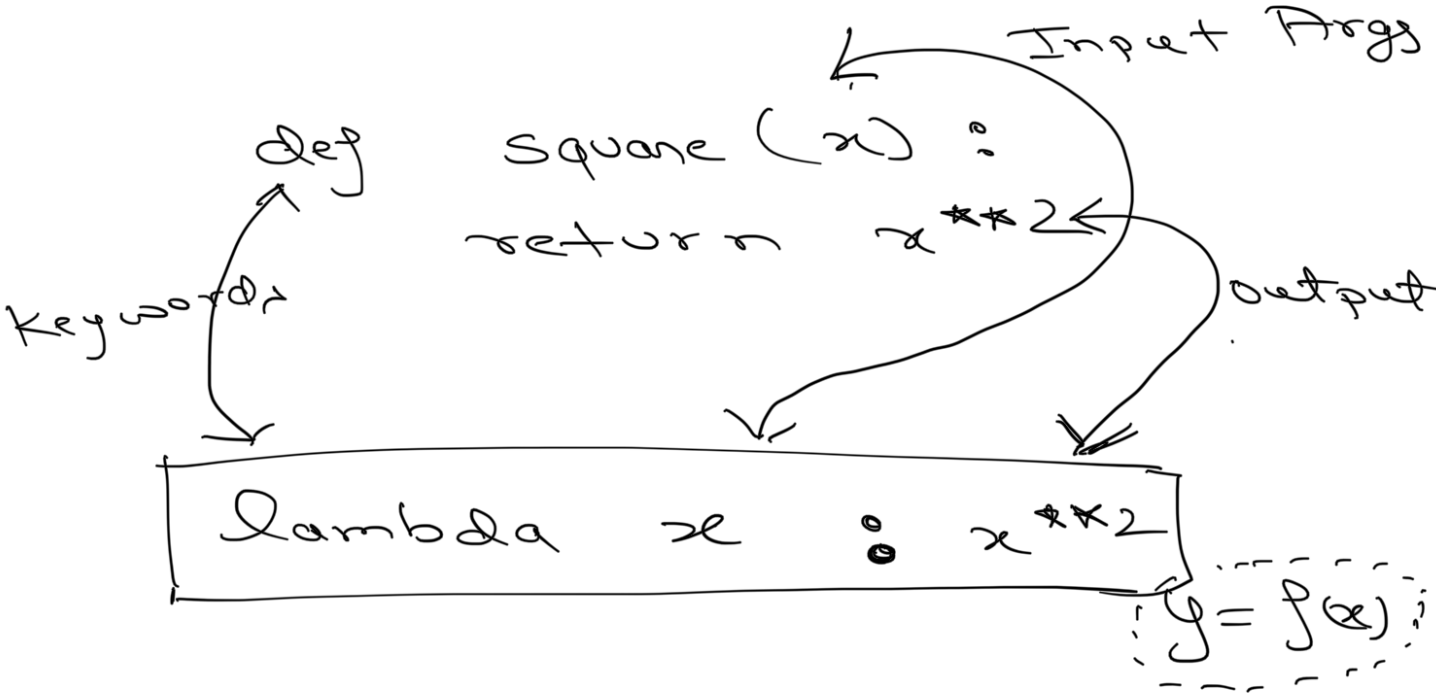
FP is a declarative programming style which focuses

What to do rather than
How to do

Advantages of FP

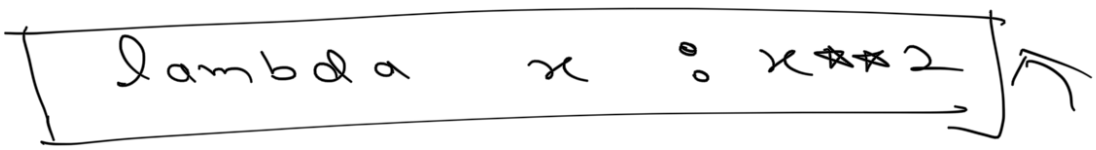
1) Concise and Readable
2) More Efficient and faster
3) Less Errors / bugs

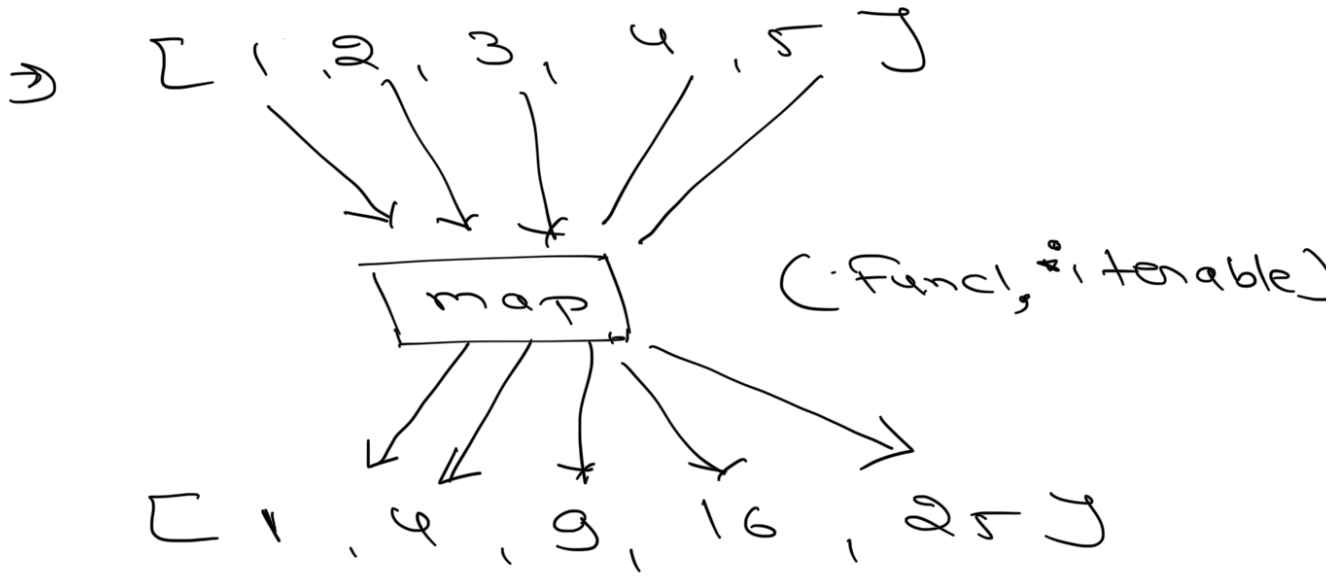
lambda Func^n



2) Anonymous functions

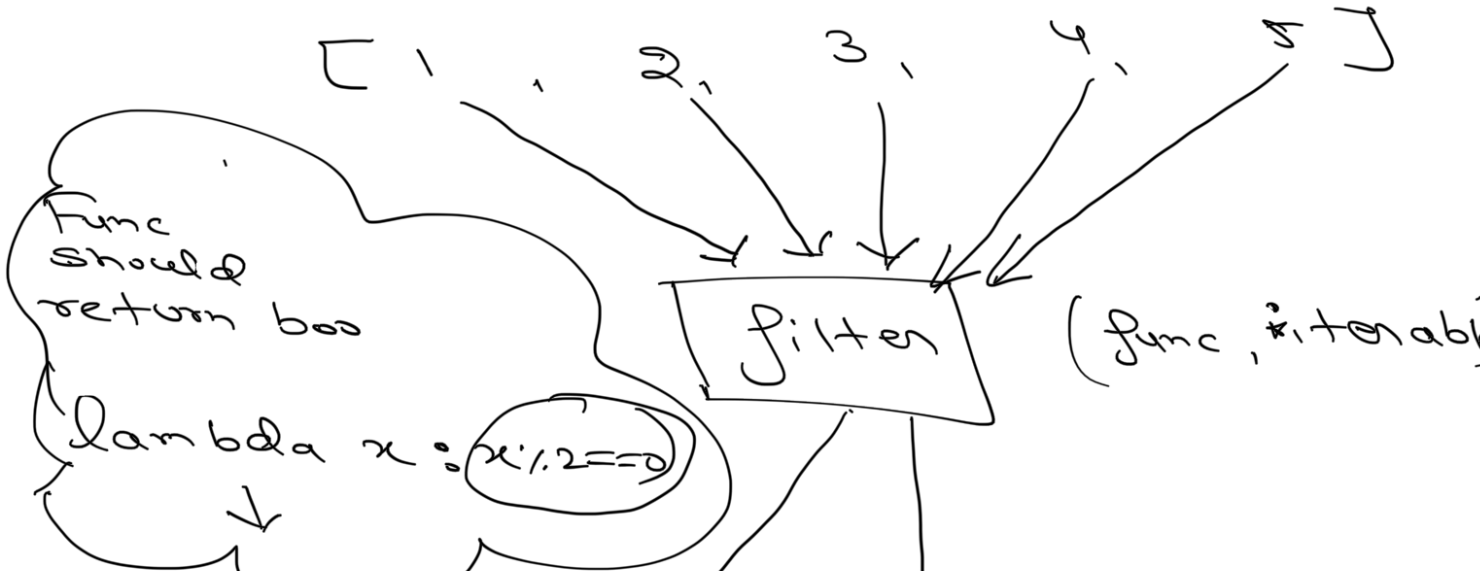
Map





map ∋ N → N

Filter



$\Rightarrow N \rightarrow \leq N$

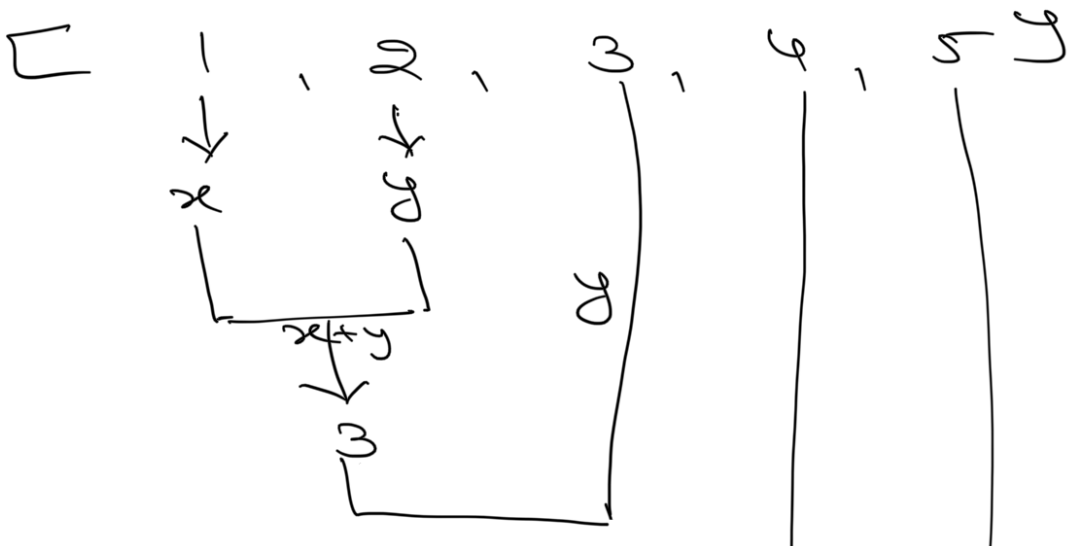
---

reduce

reduces iterable to a single object:

⇒ reduce(func, *iterables)

lambda x, y: x+y



HOF

↳ return a function

```
def get_exp(n):
    ....def exp(x)
        return x**n
    return exp
```

return inner func

→ exp_5 ⇒ get_exp(5)

exp → x**5

exp = 5(10) → 10

→ exp_10 ⇒ gen_exp(10)

→ ex_20 → gen_exp(20)