Functional Programming

Principles of FP

✓ ① Data should be separated from changes (mutations).

↓

then how?

→ via functions.

✓ ② Treat variables as immutable

↳ Don't update original data.

✓ ③ All fn in Python are FCE. /objects.

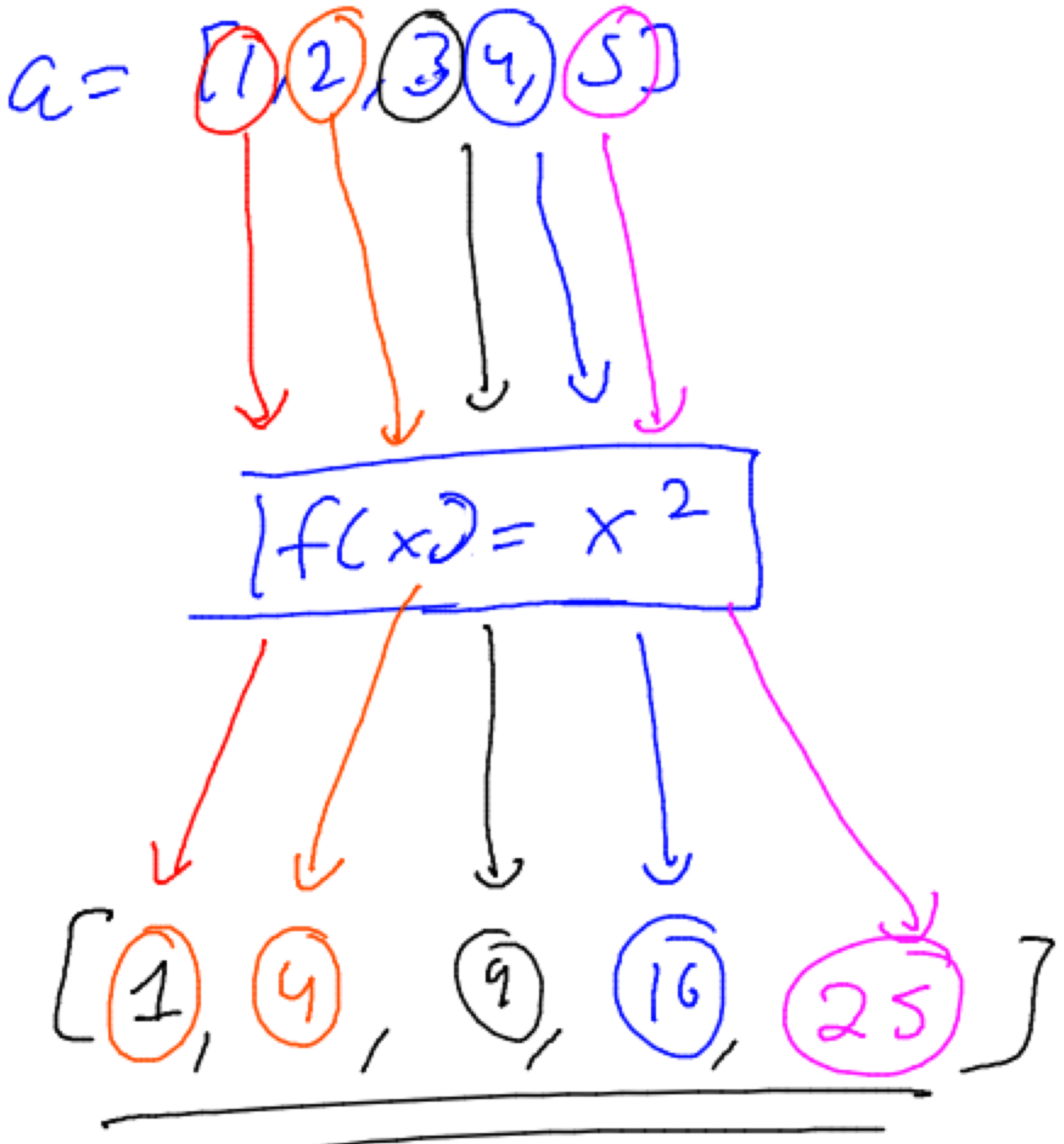
↳ Can store fn in variables.

① Map

__mul__

__add__

map

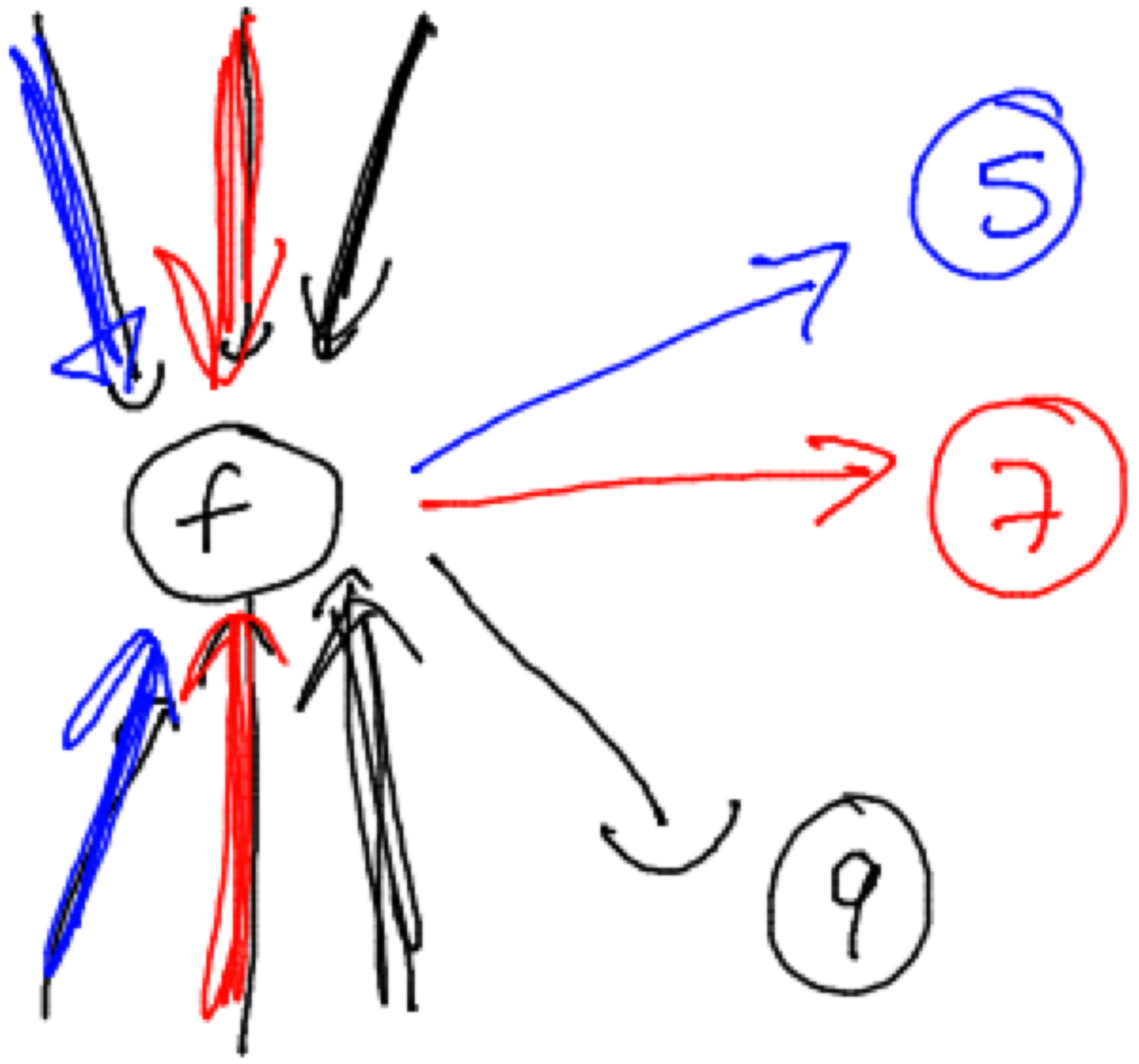


map(function, iterable)

List (map object)

$l_1 = [1, 2, 3]$

map



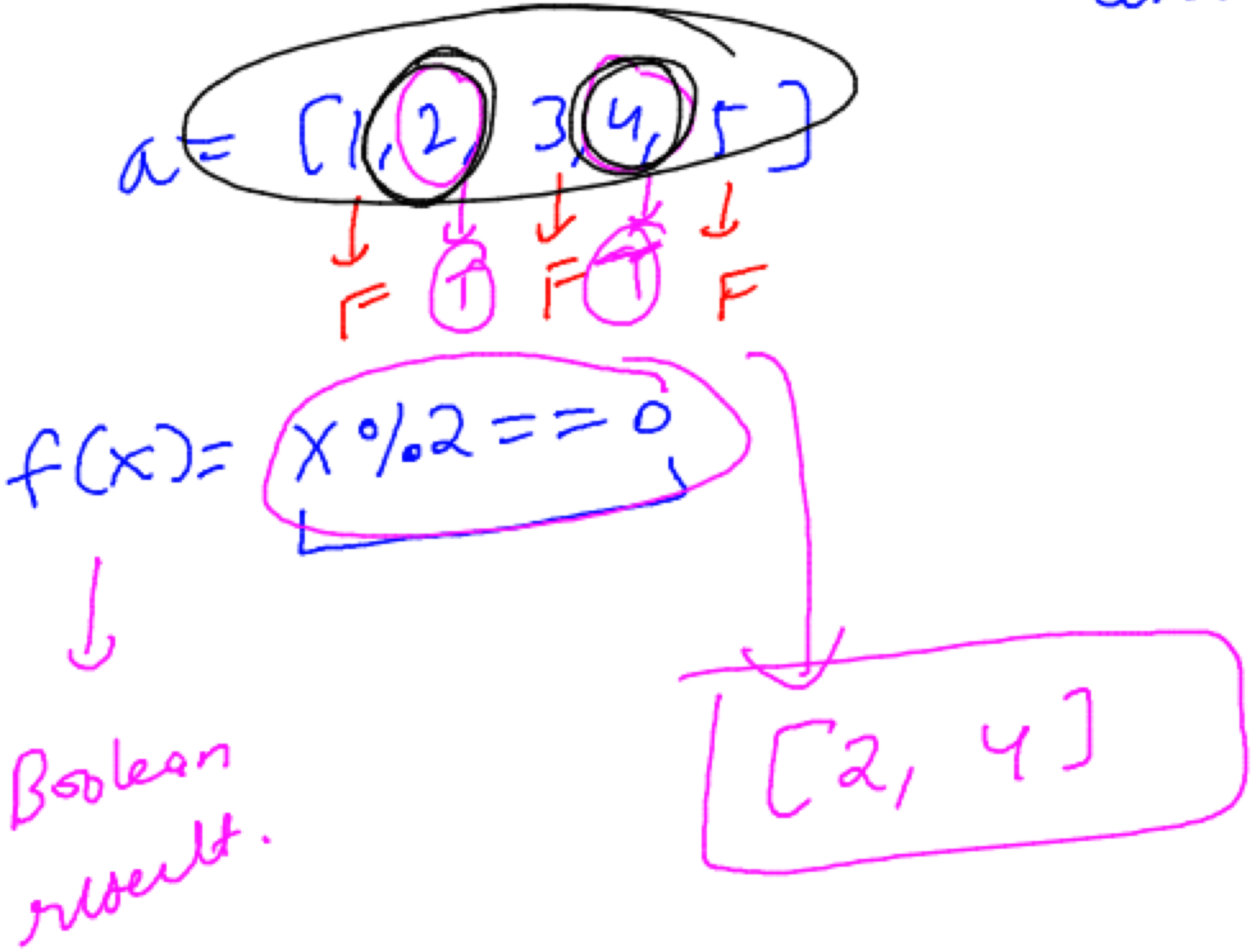
$l_2 = [4, 5, 6]$

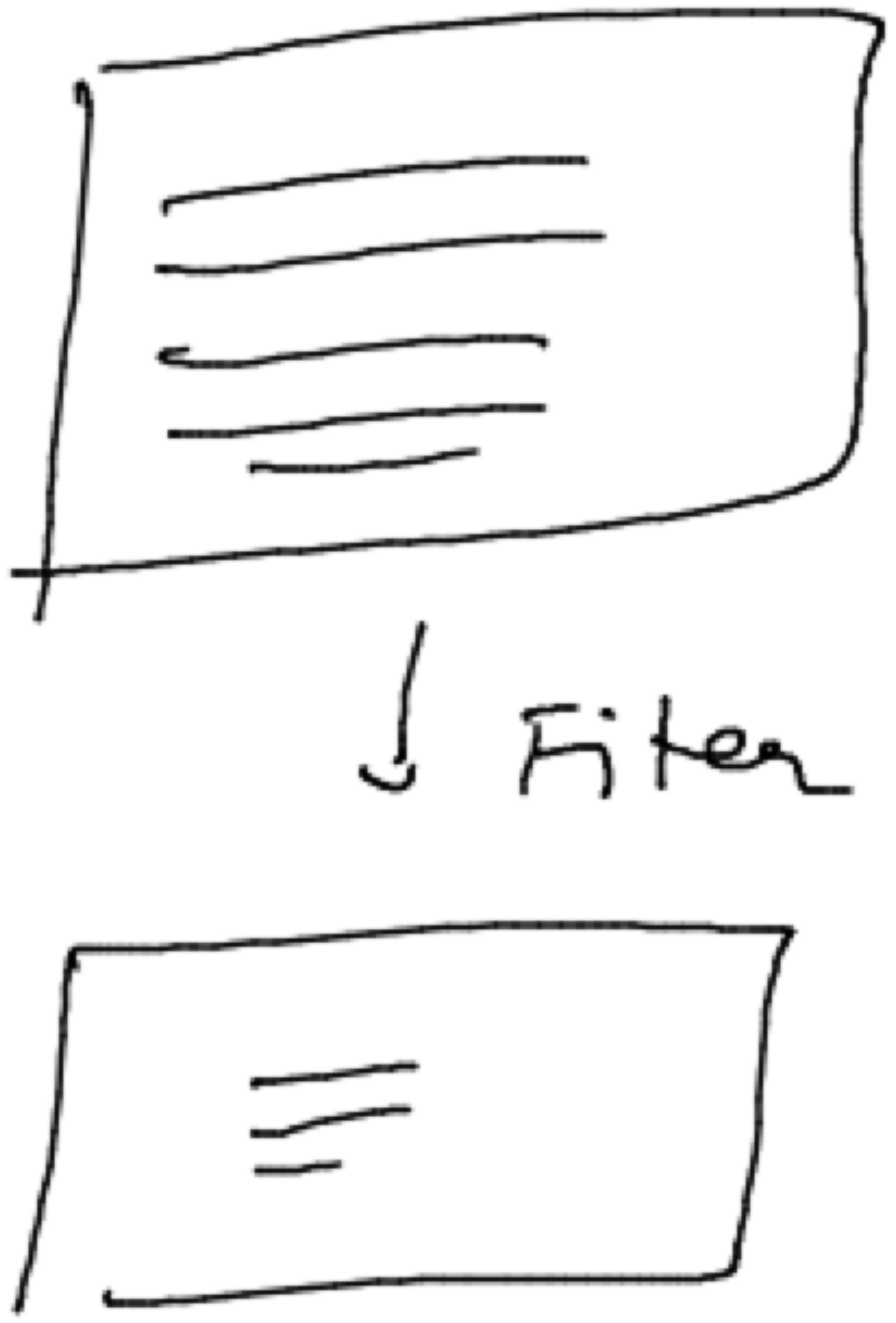
```
def f(x,y):
    return x+y
```

map(f, l_1, l_2)

② Filter

elements based on some condition.





narrow down the results

=> give you elements which have True value when fn is called.

```
res = list(map(lambda x: x%2==0, a))

# [F, T, F, T, F]
```

3 Reduce

→ No loop allowed

→ NO sum fn allowed

Add??

~~map~~

~~Filter~~

cing

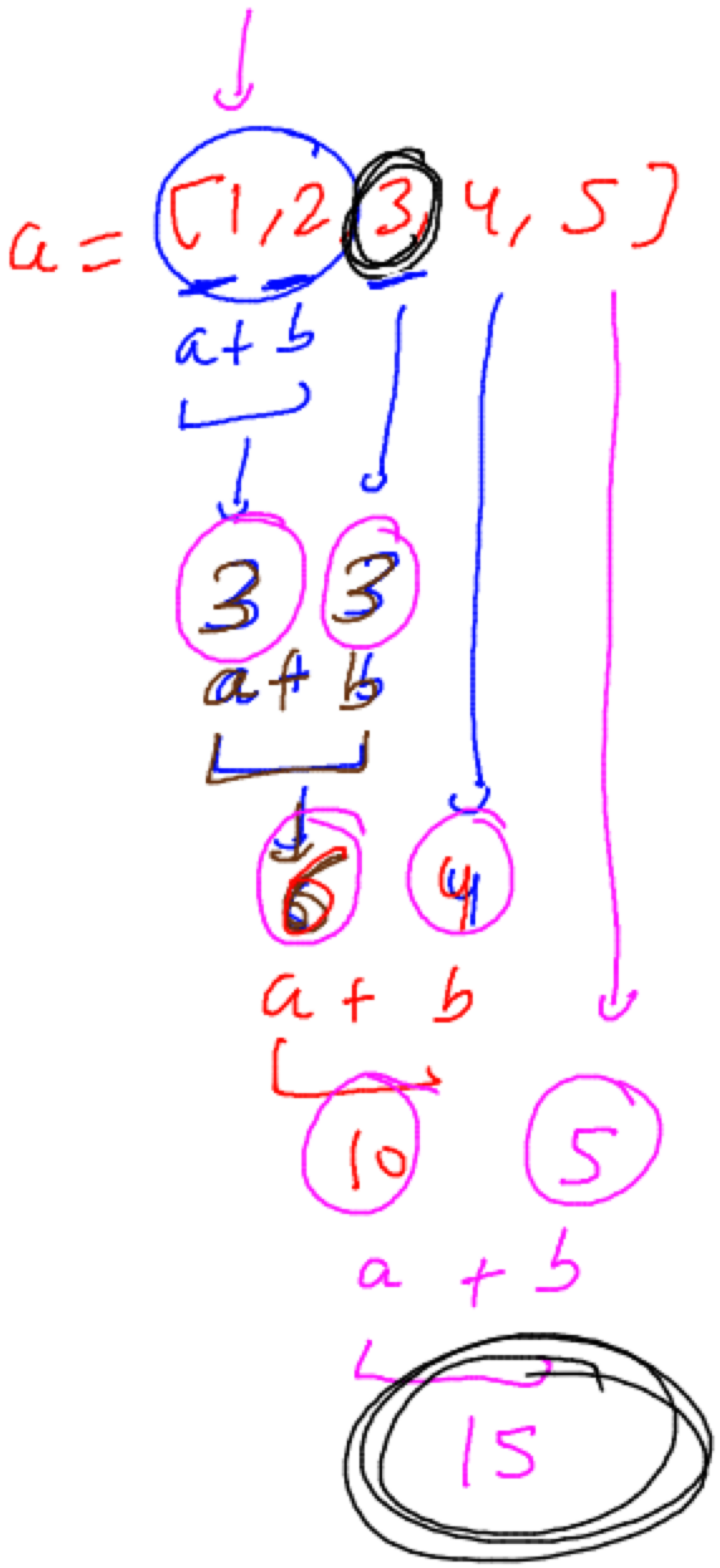
er

able to

single

value

Logical



lambda a,b : a+b

Can only take exactly 2 args.

reduce?