



# Functional & Reactive Programming in Python

MUTHUKUMARAN NAVANEETHAKRISHNAN

[muthuishere@gmail.com](mailto:muthuishere@gmail.com)



# SESSION I

---

- History on Programming Paradigms
- Introduction to Functional Programming
- Functional Programming in Python
  - lambda
  - map
  - filter
  - reduce
  - Partial & higher order functions

---

# HISTORY

---

WHERE IT ALL BEGINS



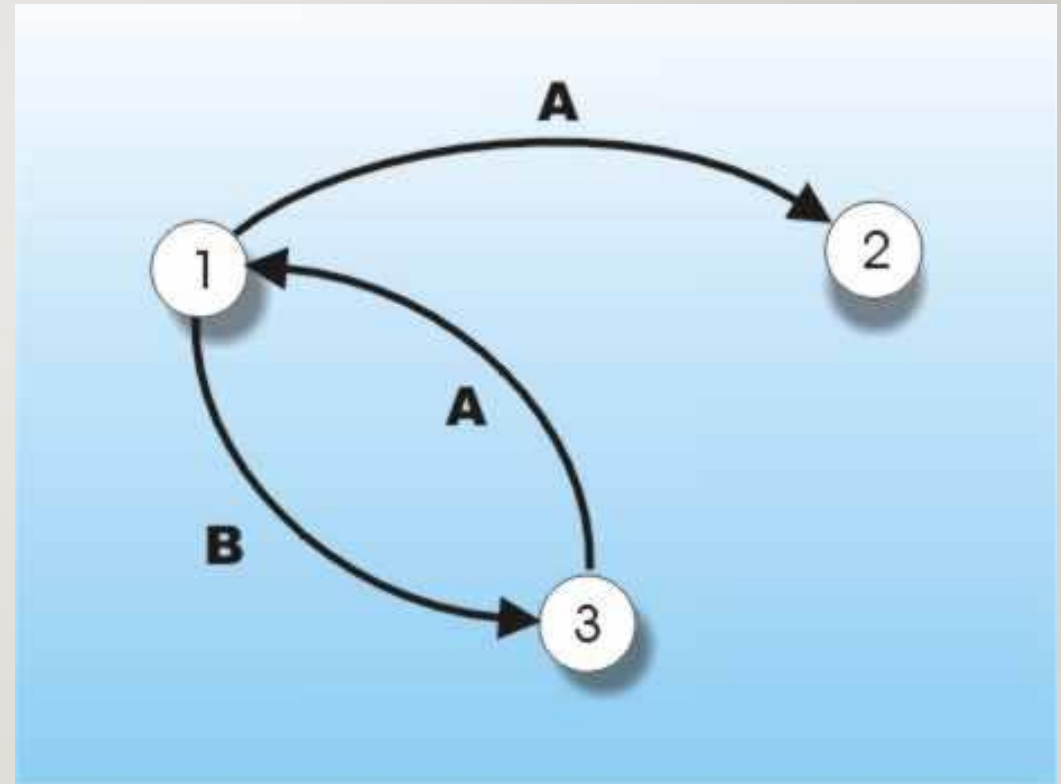
**Alan Turing**



**Alonzo Church**

ALAN TURING DEVELOPS  
TURING MACHINE, -  
FINITE STATE MACHINE

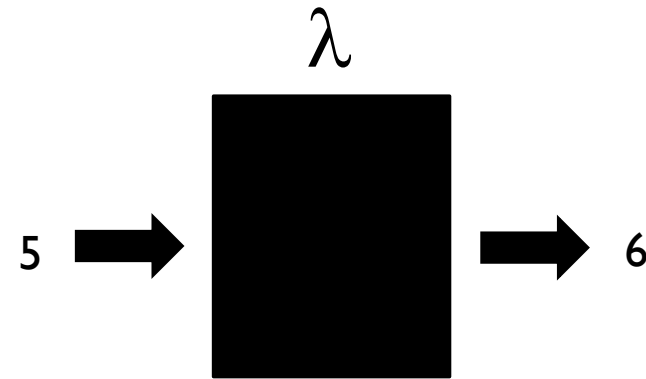
---



# ALONZO CHURCH – LAMDA CALCULUS

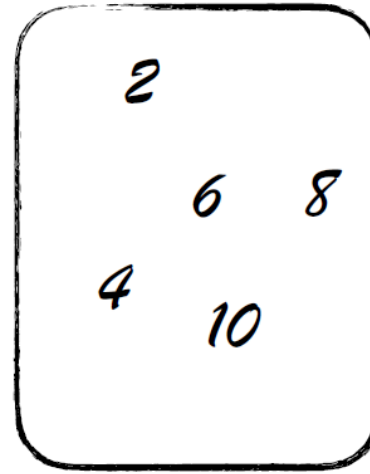
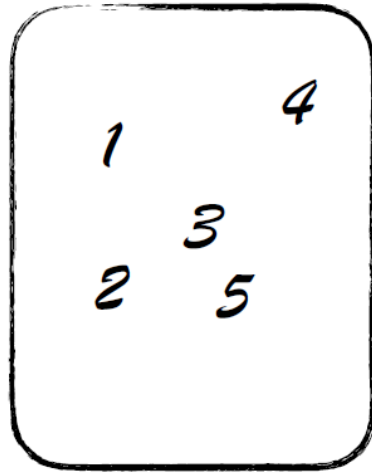
---

$$\lambda x : x + 1$$





$f(x)$



---

# FUNCTIONAL PROGRAMMING

---

INTRODUCTION



# FUNCTIONAL PROGRAMMING

- A style of programming that treats computation as the evaluation of mathematical functions
- Eliminates side effects
- Treats data as being immutable
- Expressions have referential transparency
- Functions can take functions as arguments and return functions as results

# WHAT DOES THE CODE DOES ?

---

```
results = []  
for v in vals:  
    results.append(v * 2)  
return results
```

# WHAT DOES THE CODE DOES ?

---

```
return map(multiply_by_two, vals)
```

```
multiply_by_two = lambda x: x * 2
```

# WHY DO FUNCTIONAL PROGRAMMING?

---



Allows us to write easier-to-understand, more declarative, more concise programs than imperative programming



Allows us to focus on the problem rather than the code



Facilitates parallelism

# Python & functional programming

---

- Applying Functional Programming is hard
- Lambda keyword
- In FP documentation
  - Prefers using function instead of lambda (PEP 8 ,E731)
  - Indirectly imitate users to go with loop than reduce
- Implemented Functional Programming for market case
- Not Much Developers

# LAMBDA

---

# lambda

---

- FP is orchestration of small functions
- Lambda supports creating
  - Single expression anonymous function
  - Can take parameters
  - Method body should be only one line



# LAMBDA - DEMO

---

$$(a + b)^2 = a^2 + b^2 + 2ab$$

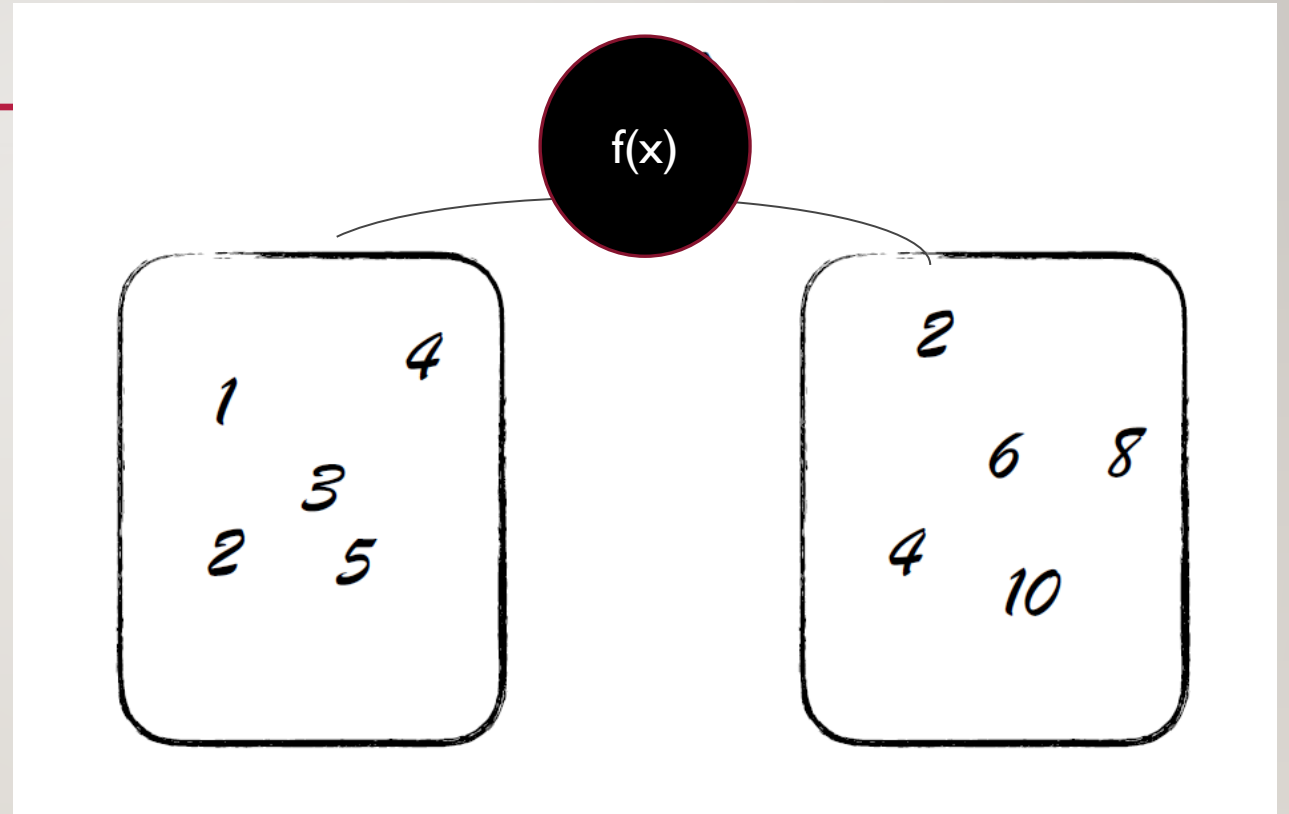
$$(a + b)^2 = a^2 + b^2 - 2ab$$

A grayscale photograph of a map with several black pushpins. A single pushpin is in sharp focus in the lower-left foreground, while others are blurred in the background. A thin red vertical line is positioned to the right of the sharp pushpin. The word "MAP" is centered in white text.

MAP

# map

- Convert from one form to other
- Changing form to a group of values by passing through a function
  - Applied in *numpy* & most of packages



# DATA

---

```
users =[
  {
    "id": 1,
    "first_name": "Mandy",
    "last_name": "Gowan",
    "email": "mgowan0@aol.com",
    "gender": "Female",
    "ip_address": "1.25.197.128",
    "salary": 119885
  },
  {
    "id": 2,
    "first_name": "Janessa",
    "last_name": "Cotterell",
    "email": "jcotterell1@delicious.com",
    "gender": "Female",
    "ip_address": "155.82.134.71",
    "salary": 107629
  },
  {
    "id": 3,
```

# Map demo

---

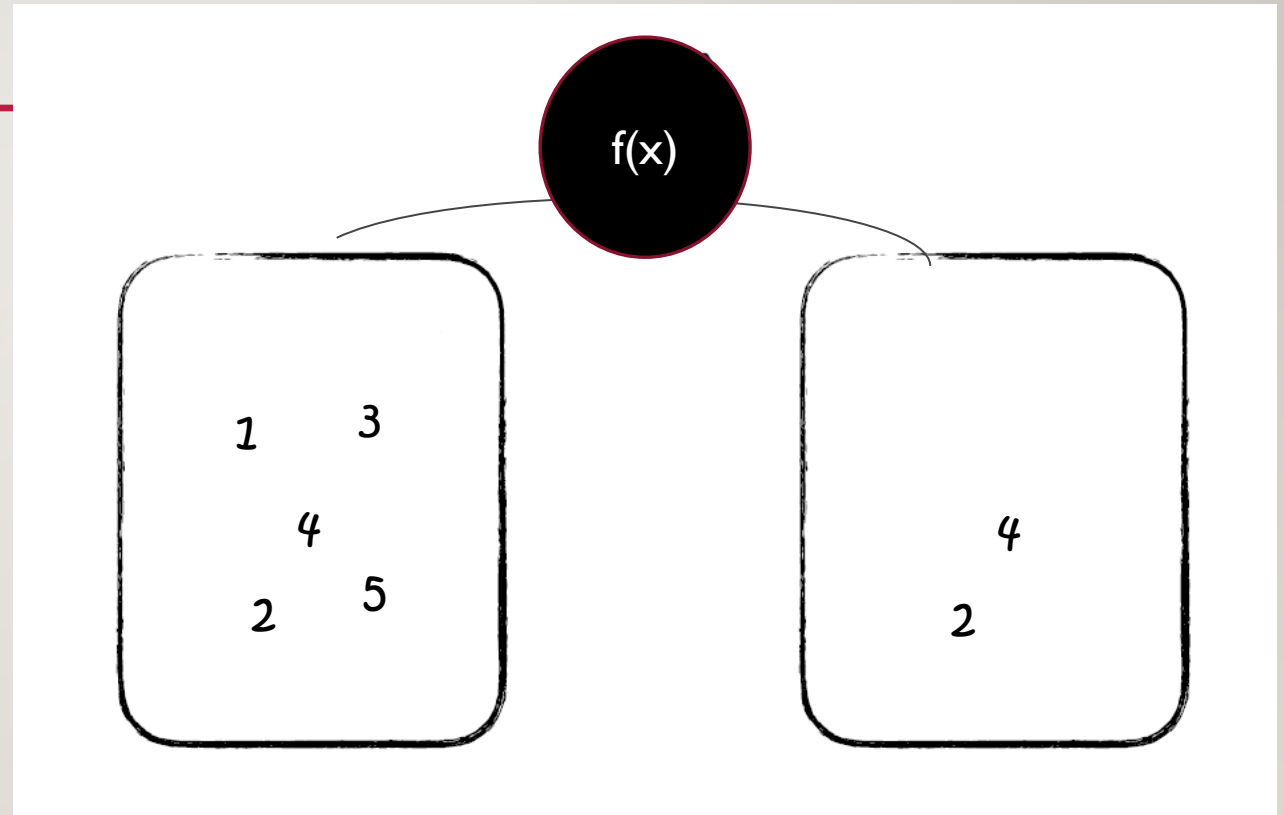
- Get ids of user
- Get names marked with prefix (Mr or Miss)

A grayscale background image of a map with several black pushpins. One pushpin is in sharp focus in the lower-left foreground, while others are blurred in the background. A thin red vertical line is positioned to the left of the word 'FILTER'.

**FILTER**

# filter

- Filter from a set of values by applying through function





# Filter demo

---

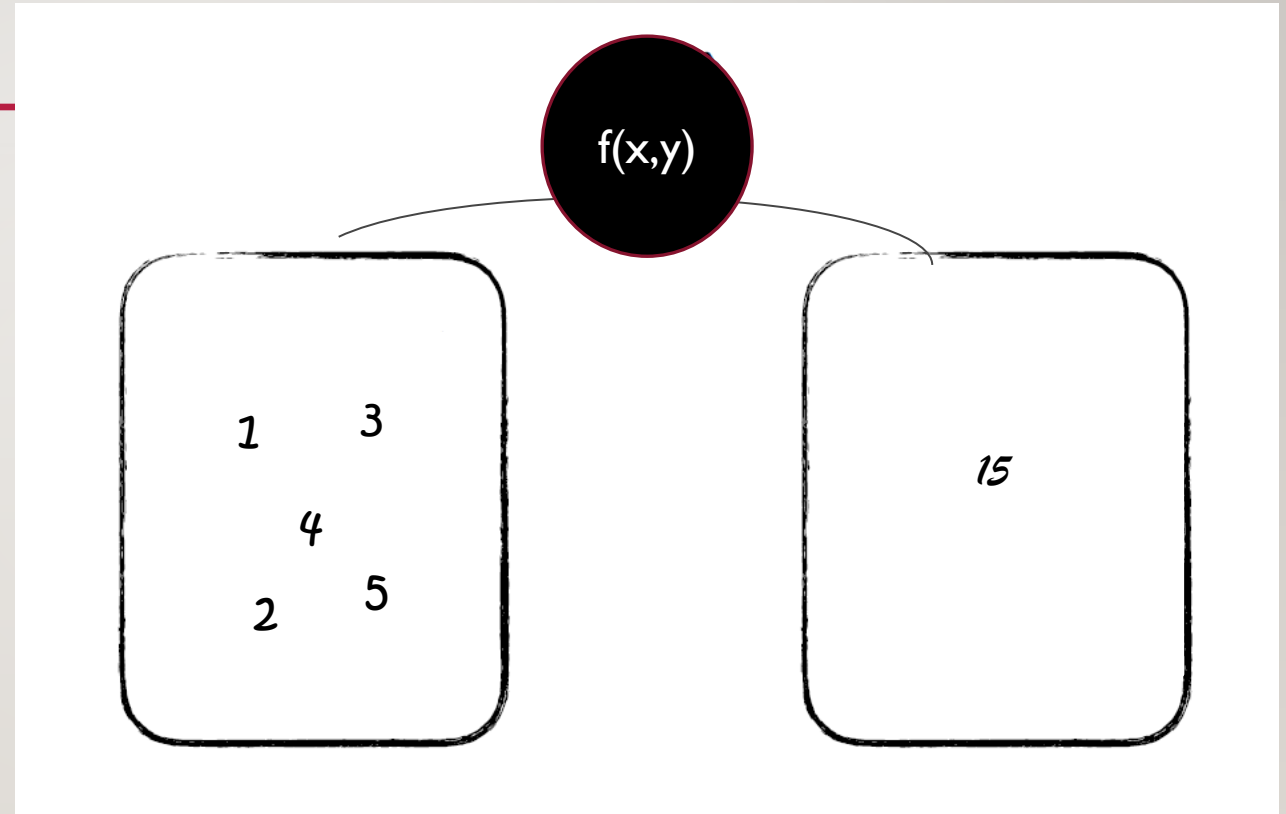
- Query users based on gender

A grayscale image of a map with several black pushpins. One pushpin is in sharp focus in the lower-left foreground, while others are blurred in the background. A thin red vertical line is positioned to the left of the word 'REDUCE'.

REDUCE

# reduce

- Reduce a group of values to a single value by applying a function  $f(x,y)$ 
  - $x$ - accumulator
  - $y$  – current value



# Reduce demo

---

- Find total salaries of users

**END OF SESSION I**