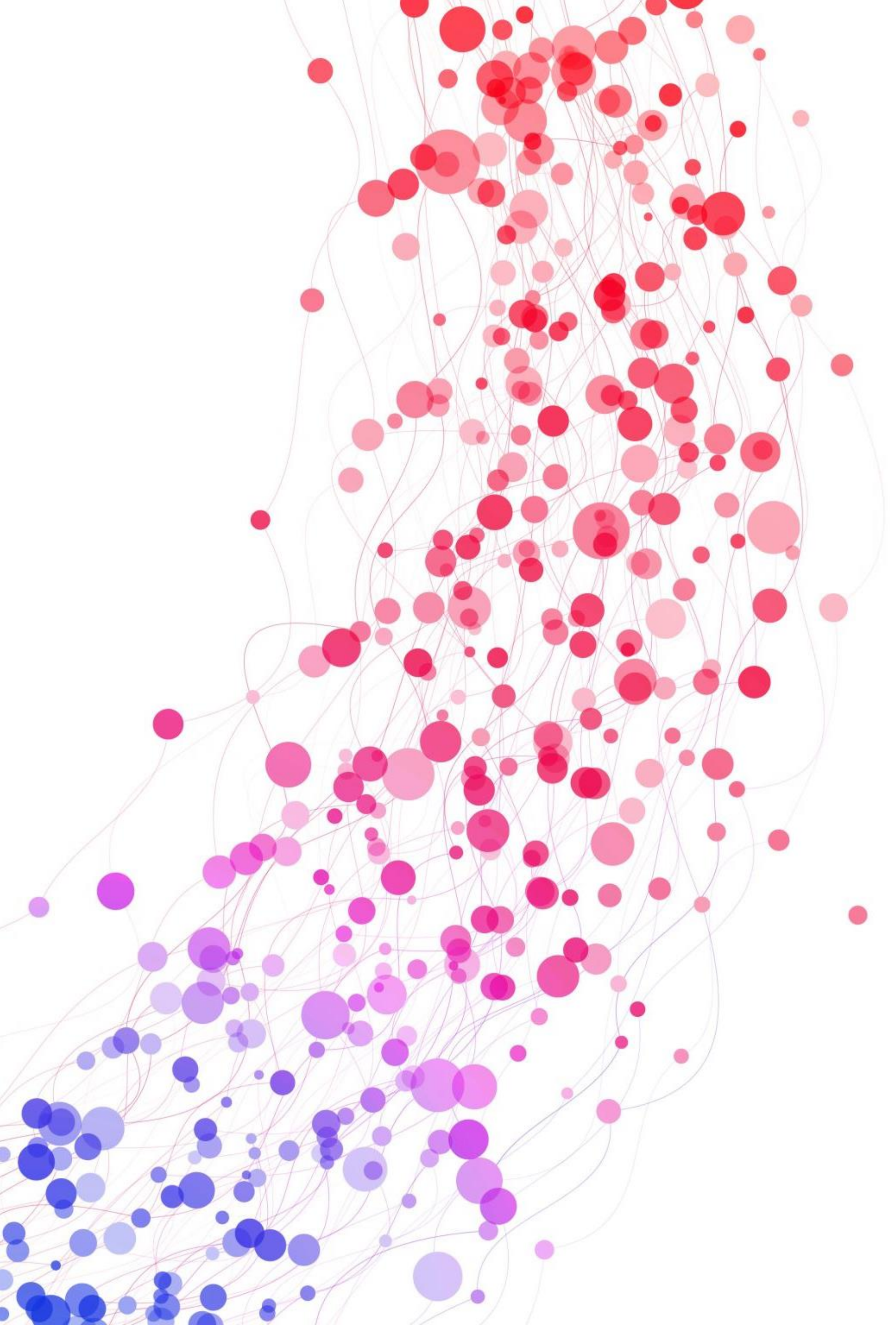


PROGRAMMING PARADIGMS IN PYTHON

(FUNCTIONAL & REACTIVE)

MUTHUKUMARAN NAVANEETHAKRISHNAN

SESSION II

- Generators
- Composing Functionals
- Partials
- Reactive Programming
- Rx Extensions
- Observables

Generators

- Lazy Evaluating Sequences
- Manage state over return type through yield
- Uses __next__ method to see , if there is more data

Generators Demo

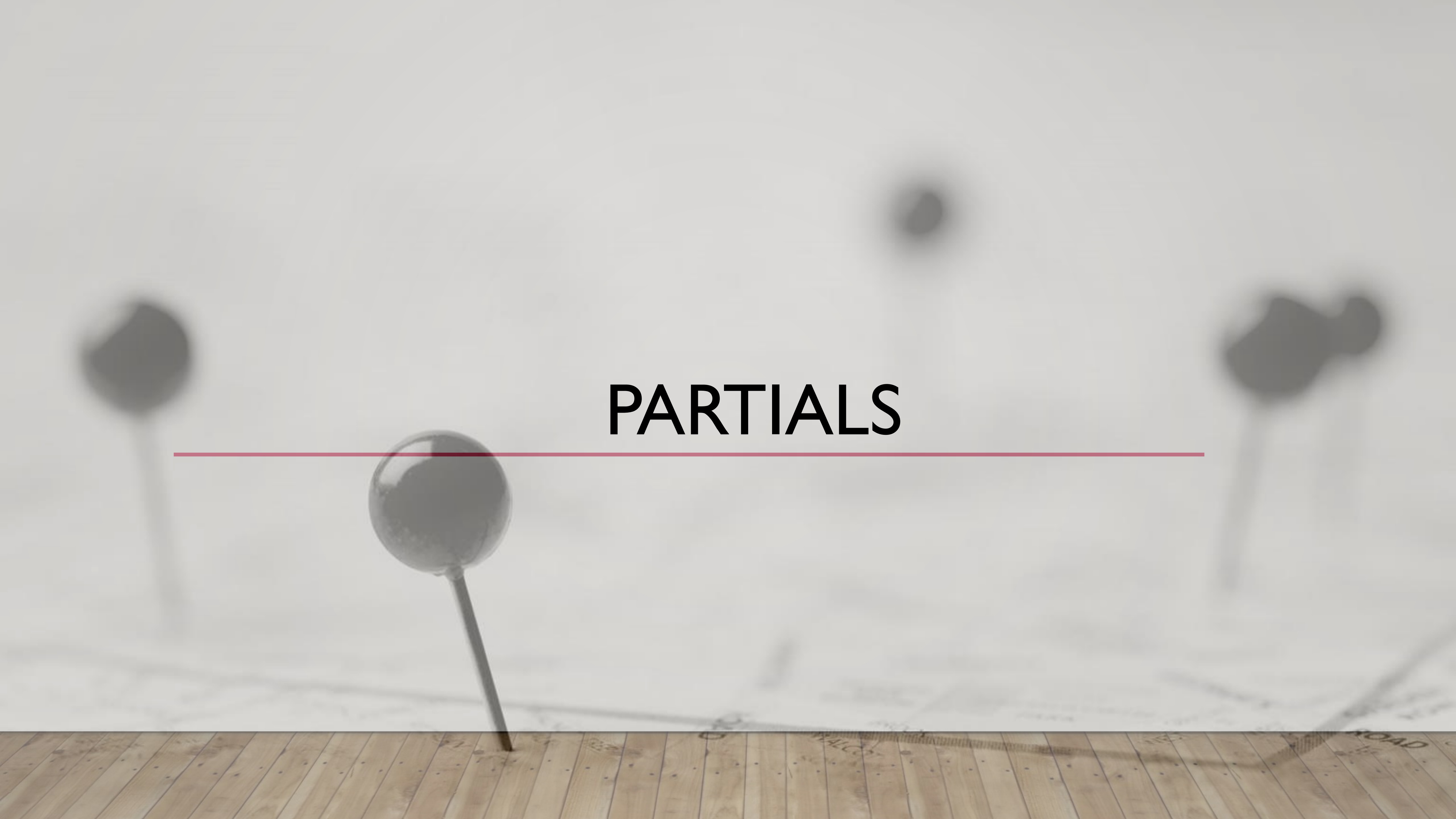
- A program to return multiples of 248 for the given n numbers
- Read a large file line by line

Composing Functionals

- Abstractions
- Orchestration

Compose demo

- Find total salaries of female users
- Find names of user starts with J



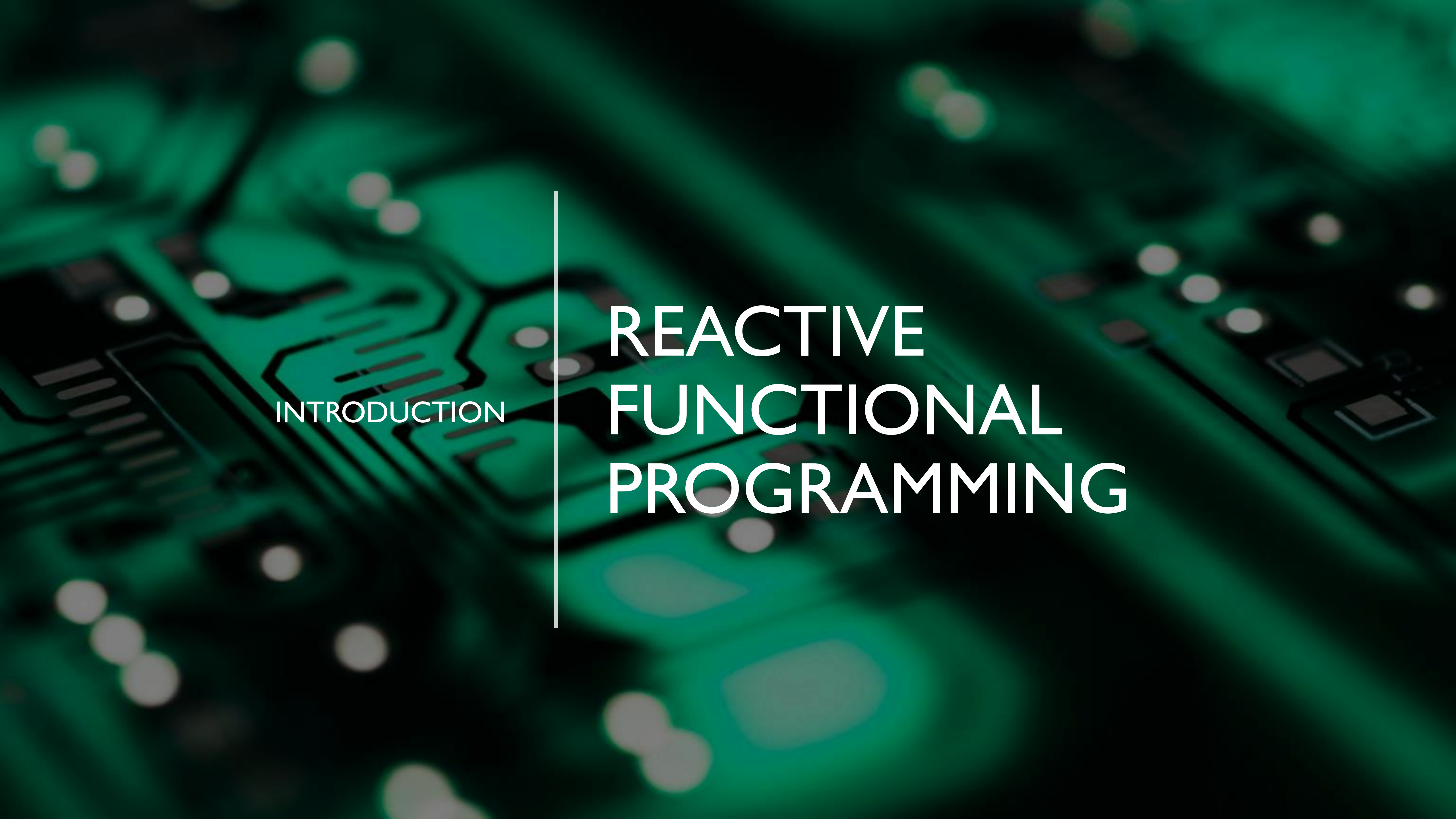
PARTIALS

PARTIALS

- Abstracting or Splitting Function
- For Functions which has more than one parameter
  - Split the function with partial
  - Can execute that function later if required
  - Use it with caution
- Common Use cases
  - Functions which has more than one arguments

Partial demo

- Add Two Numbers
  - Increment One with Partial
- AddTwoNumbersAndMultiplyThird do with partial
  - Add two numbers
  - Multiply two numbers
- Refactor total salaries of female users



INTRODUCTION

REACTIVE FUNCTIONAL PROGRAMMING

REACTIVE PROGRAMMING

```
b : = 4
a  := b + 5
b := 8
```

What
Is the
value of
a ?

REACTIVE PROGRAMMING

```
python_dev_salary = 1000000
python_with_data_science_salary = python_dev_salary + (python_dev_salary * (26/100))
python_dev_salary = 90000
```

What
Is the value of
python_with_data_
science_salary ?

PULL BASED

- Polling from data source
- Uses Iterator pattern
    - **__next__()**

PUSH BASED

- Subscribe to data source
- Handle
  - data arrival
  - completion
  - error

ASSASINATION ATTEMPTS ON JAVASCRIPT

Microsoft Live Labs Volta

GWT

RX EXTENSIONS

Reactive Extensions (Rx)

Microsoft Live Labs Volta

RX EXTENSIONS

- ReactiveX is a combination of the best ideas from
  - Observer pattern
  - Iterator pattern
  - Functional programming
- Invented by Cloud Programmability Team at Microsoft around ~~2011~~2009-2011
- Intended for .Net , moved for Javascript & C++
- Made as open source in 2012

OBSERVABLE

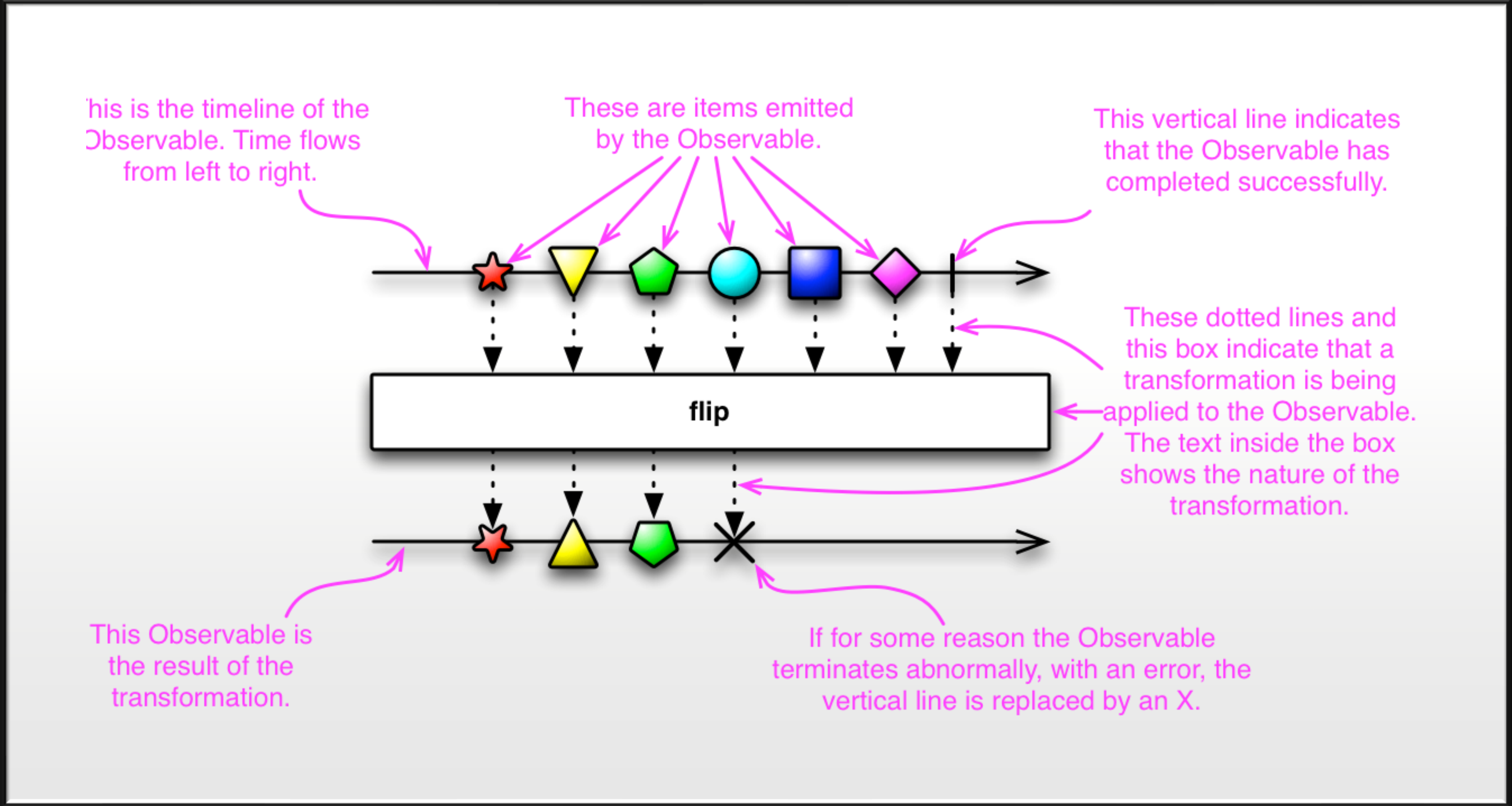
- An *observer subscribes* to an *Observable*.
- observer reacts to whatever item or sequence of items the Observable *emits*.
- The Observable emits three state
  - On Data
  - On Error
  - On Completed

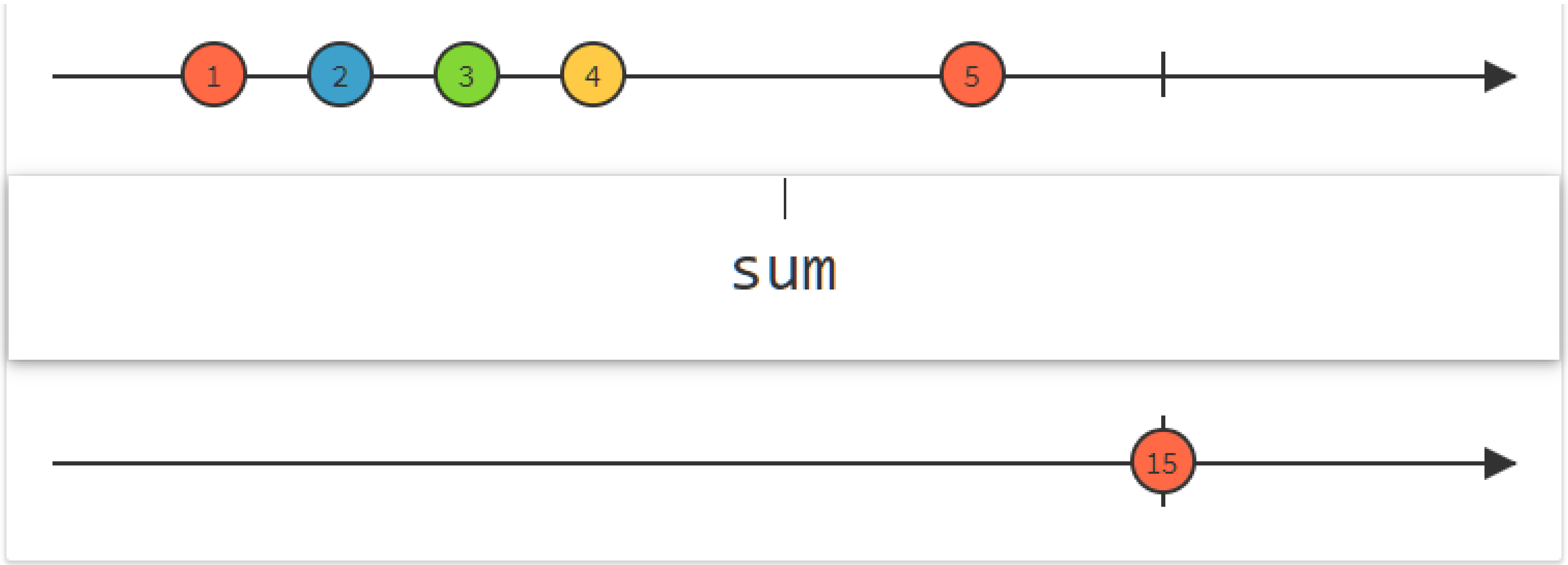
CREATING OBSERVABLE DEMO

- Create Observables to emit
  - Hello
  - World
- Verify onData,OnComplete OnError

REACTIVE PROGRAMMING IN RX

- Streams
- Functional Programming
- Asynchronous Observers


his is the timeline of the
bservable. Time flows
from left to right.
These are items emitted
by the Observable.
This vertical line indicates
that the Observable has
completed successfully.
These dotted lines and
this box indicate that a
transformation is being
applied to the Observable.
The text inside the box
shows the nature of the
transformation.
flip
This Observable is
the result of the
transformation.
If for some reason the Observable
terminates abnormally, with an error, the
vertical line is replaced by an X.


1
2
3
4
5
sum
15

END OF SESSION II