Angular Training
(Intermediate to Advanced)

Narasimha
Sr. Corporate Trainer, Mentor
tnrao.trainer@gmail.com

Schedule for React JS Training

Day1	:	Custom Pipes in Angular
Day2	:	Parent-Child Communication
Day3	:	Custom Directives in Angular
Day4	:	Working with Reactive Forms
Day5	:	Dependency Injection and Services in Angular
Day6	**:**	**RxJS Library – Observables**
Day7	:	Http Client – Server calls in Angular
Day8	:	Routing and Security in Angular

Day6
RxJS Library - Observables

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day6

- What is RxJs?
- Synchronous vs Asynchronous
- Understanding Observables, Subscribing
- Using RxJs operators : Creational, Filtering
- What is Subject in RxJS?
- Summary
- Q & A

What is RxJS?

- RxJS - Reactive Extensions for JavaScript.
- RxJS is a popular library among web developers.
- RxJS ilibrary for reactive programming using observables that makes it easier to compose asynchronous calls.
- It makes easy write asynchronous code using composable Observables instead of callbacks and Promises.

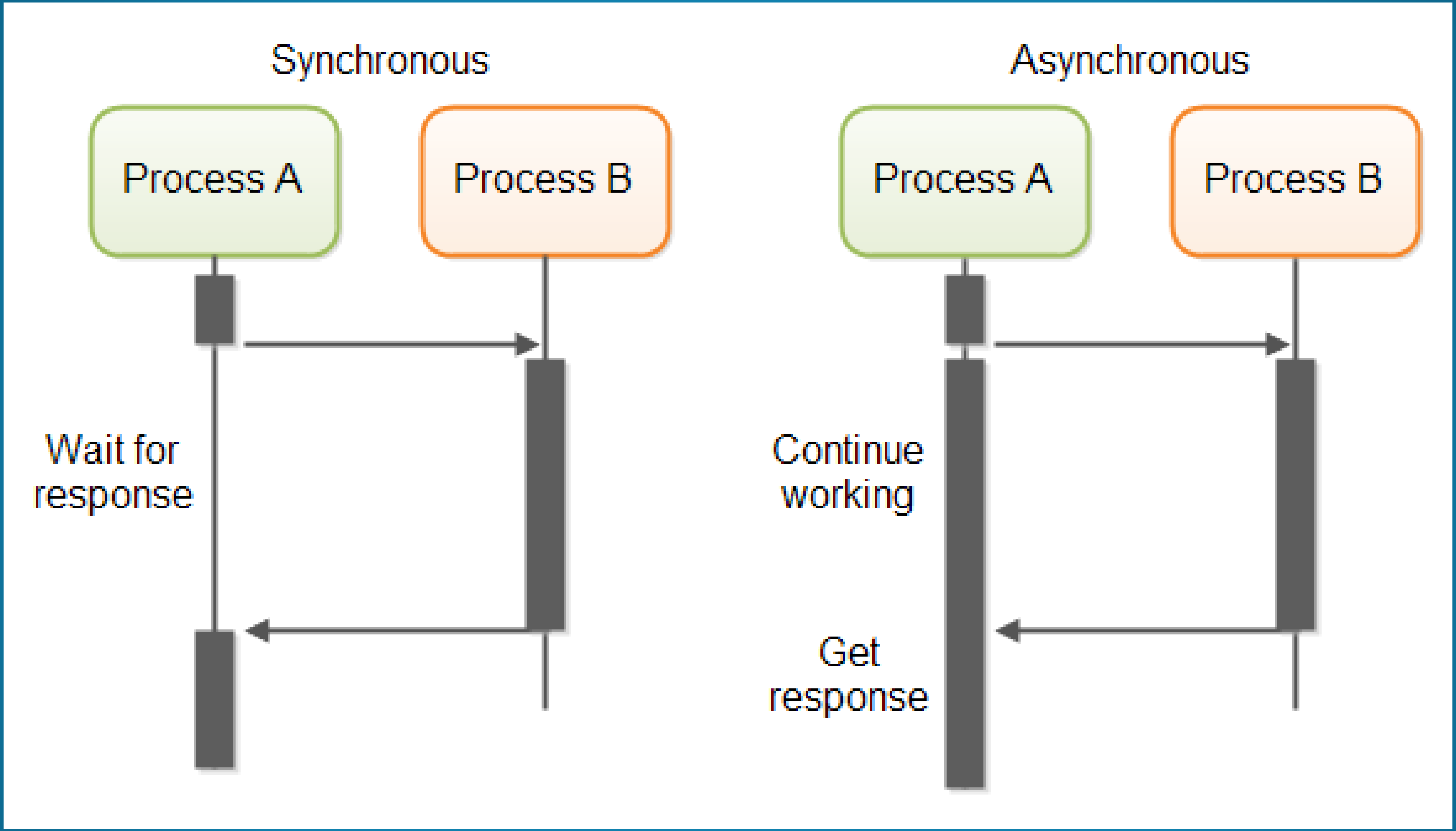
What is RxJS?

- If your project consists of lots of async task handling than RxJS is a good choice.
- RxJS has been integrated in many web development libraries and frameworks such as Angular.
- It is loaded by default with the Angular project.

Synchronous vs Asynchronous

Narasimha

Sr. IT Trainer/Consultant


Synchronous
Process A
Process B
Wait for response
Asynchronous
Process A
Process B
Continue working
Get response

Synchronous vs. Asynchonous

Synchronous: real time

Asynchronous: No immediate response

AJAX Calls - HTTP requests : **asychronous**

Understanding Observables, Subscribing

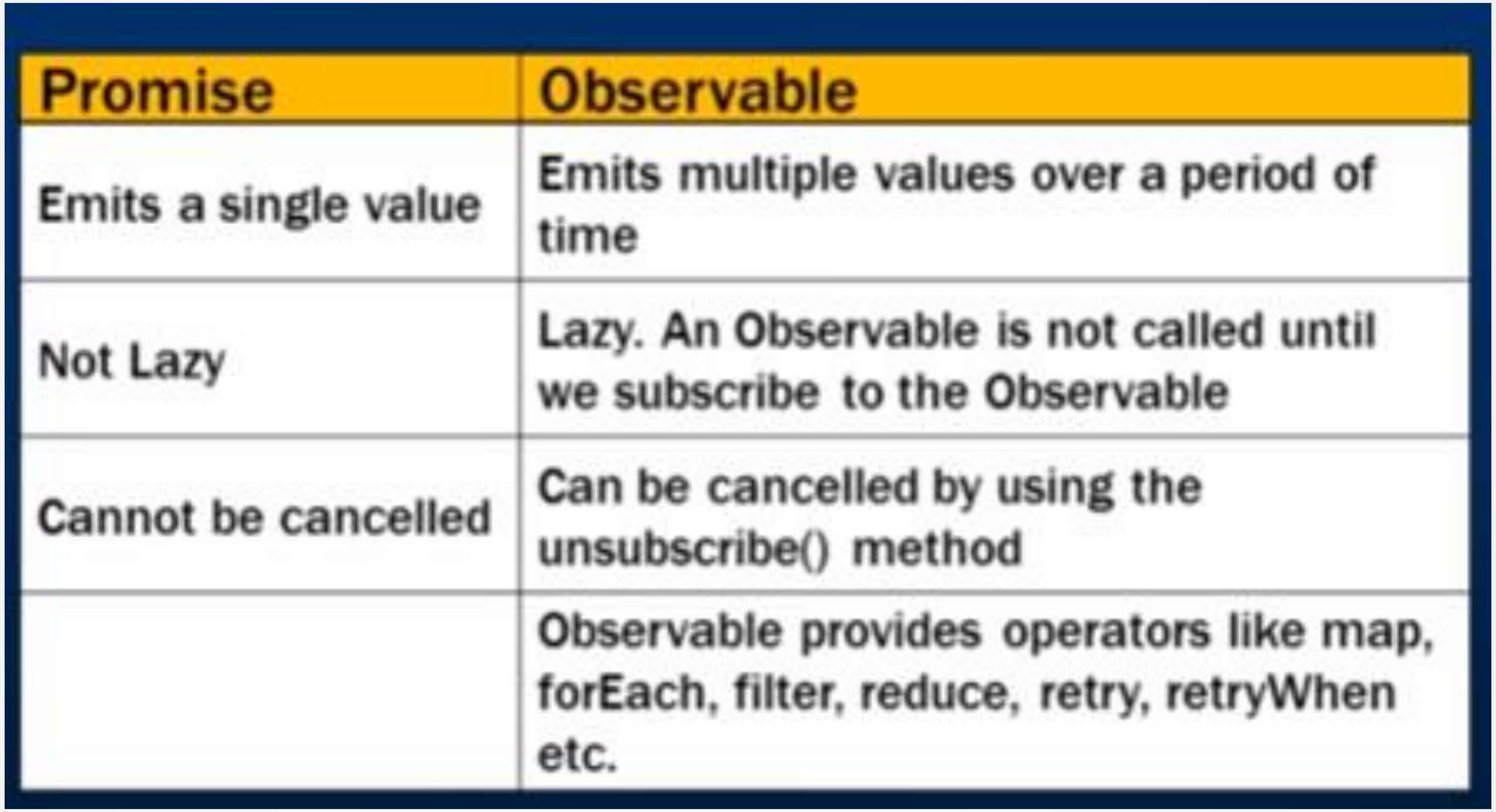
Narasimha

Sr. IT Trainer/Consultant

What is observables in RxJS?

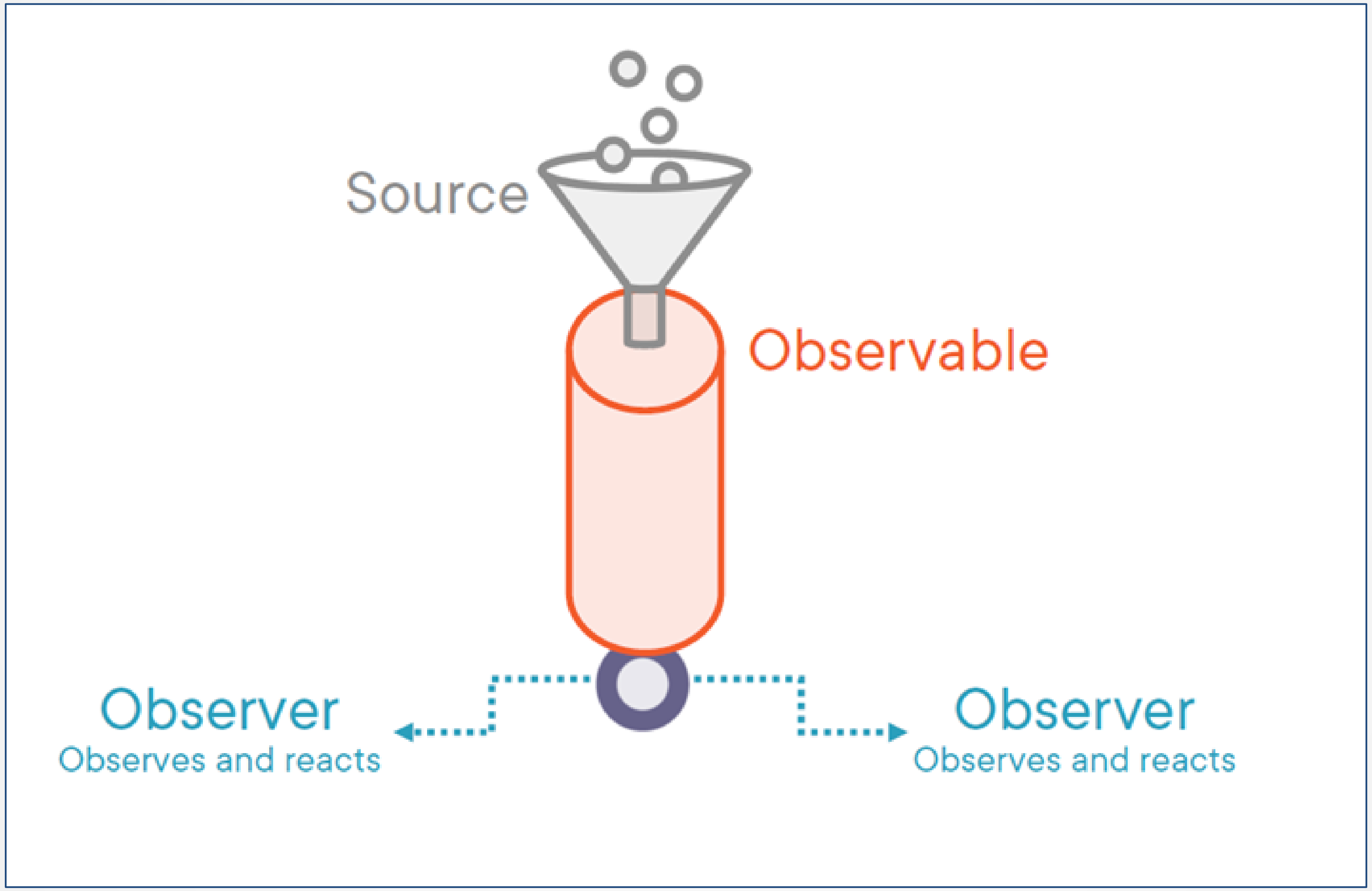
- An observable is a function that creates an observer and attaches it to the source where values are expected from.
  - For example: Http request, etc.
- Observables provide support for passing messages between parts of your application.
- They are used frequently in Angular for asynchronous programming, and handling multiple values.

Observable vs Promises



Promise	Observable
Emits a single value	Emits multiple values over a period of time
Not Lazy	Lazy. An Observable is not called until we subscribe to the Observable
Cannot be cancelled	Can be cancelled by using the unsubscribe() method
	Observable provides operators like map, forEach, filter, reduce, retry, retryWhen etc.

Observables



Anatomy of an Observable

Core Observable concerns:

1. **Creating** Observables
2. **Subscribing** to Observables
3. **Executing** the Observable
4. **Disposing** Observables

Creating Observables

```
const foo = new Observable((subscriber) => {
                        // code to emit the results
});
```

There are three types of values an Observable Execution can deliver:

a. next() : sends a actual success response

b. error() : sends the error response

c. complete() : does not send a value.

Creating Observables

```
let observableObj = new Observable( (subscriber) => {
          subscriber.next("");
          subscriber.error("");
          subscriber.complete();
});
```

Subscrbing Observables

```
observableObj.subscribe(callback);
```

```
observableObj.subscribe(callback, callback, callback);
```

```
observableObj.subscribe({
        next :    callback,
        error :   callback,
        complete :   callback
});
```

Practical HandsOns

RxJS Operators

Narasimha

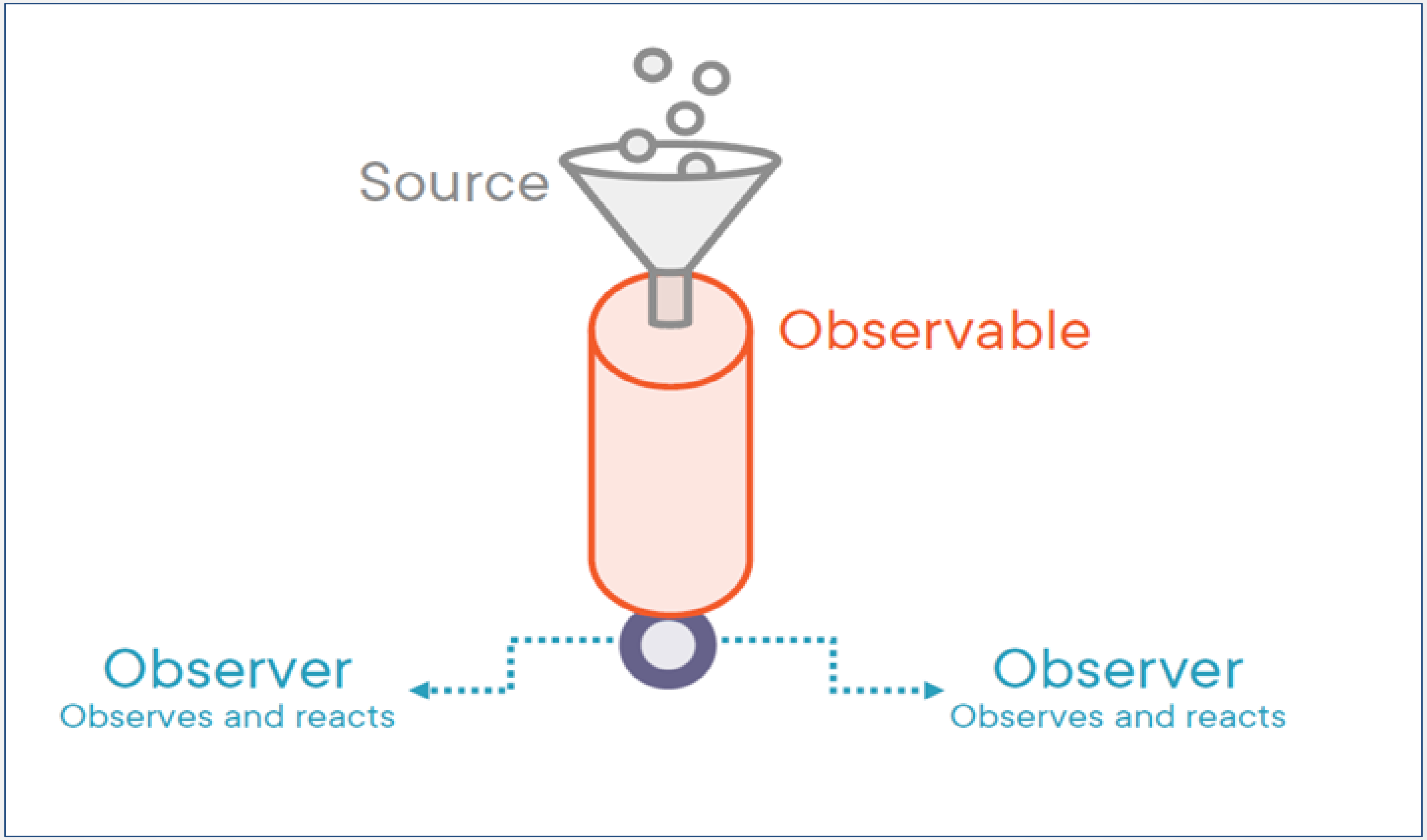
Sr. IT Trainer/Consultant

RxJS Operators

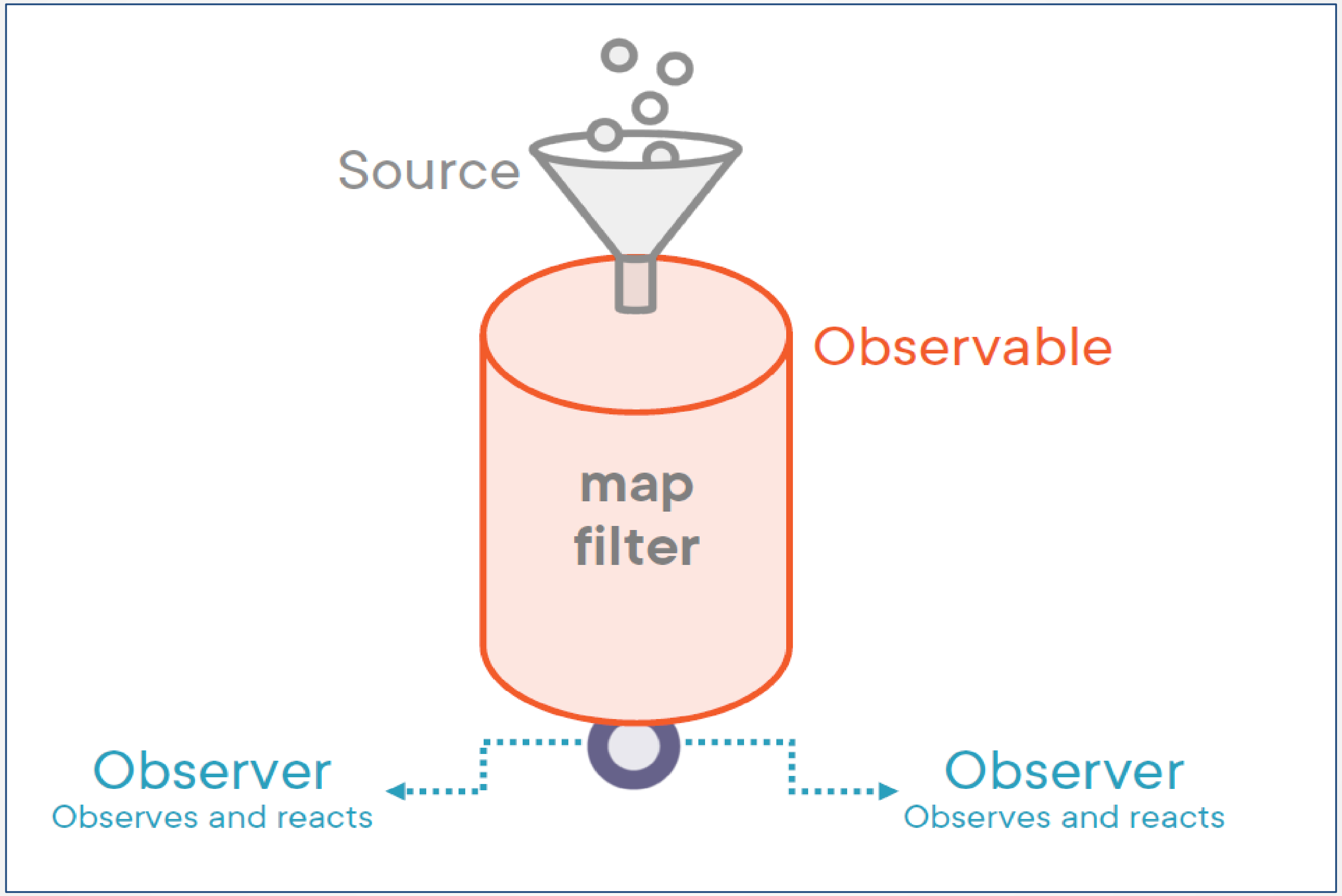
- An operator is a function
- Used to transform and manipulate emitted items
- Apply operators in sequence using the Observable's pipe() method.

```
of(2, 4, 6).pipe( map(item => item * 2))
            .subscribe(item => console.log(item));
```

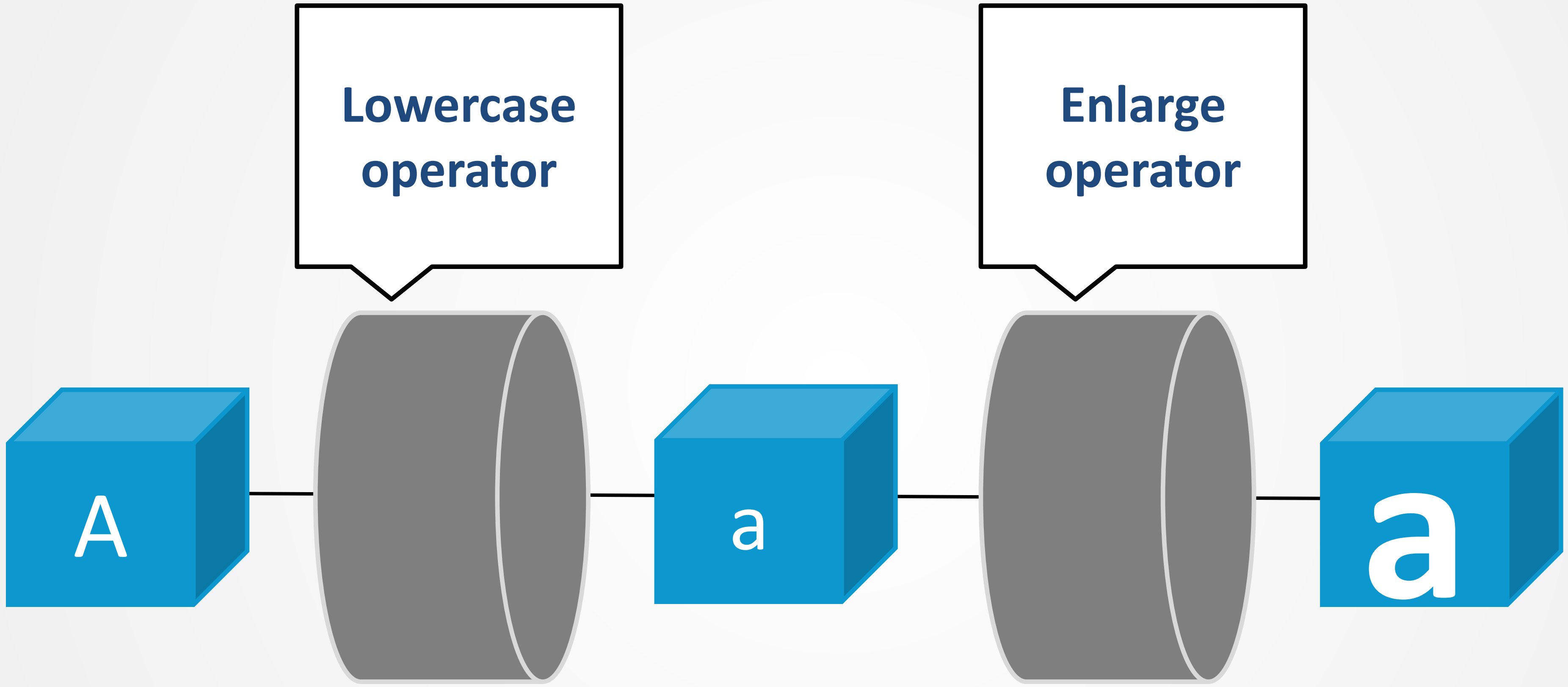
RxJS Operators



RxJS Operators



RxJS Operators



RxJS Operators

Creational	:	of(), from(), interval()
Filtering	:	map(), filter(), pipe()

Practical HandsOns

Q & A