Angular observables

GFT INTERNAL TRAINING

INNOVATE. TRANSFORM. DELIVER.

Agenda

1.intro

2.some useful functions

3.observables

4.learning resources

intro

Observables

- isn't an Angular specific feature, but rather a proposed standard for managing async data that will be included in future javascript versions (hopefully)

- open up a continuous channel of communication in which multiple values of data can be emitted over time

- are a pattern of dealing with data by using array-like operations to parse, modify and maintain data

- Angular uses them extensively - you'll see them in the HTTP service and the event system

RxJS

Being Reactive, my friend

http://reactivex.io

http://reactivex.io/rxjs/

Some useful functions

map

```
a = [1, 2, 3, 4]
▶ (4) [1, 2, 3, 4]

a.map( x => 3*x )
▶ (4) [3, 6, 9, 12]

a
▶ (4) [1, 2, 3, 4]
```

filter

```
a = [1, 2, 3, 4]
▶ (4) [1, 2, 3, 4]

a.filter( x => x>1 )
▶ (3) [2, 3, 4]

a
▶ (4) [1, 2, 3, 4]
```

Observables

Observable === Collection + Time

Multiple values (*)

Array

```
res =
  stocks
  .filter(q => q.symbol == 'FB')
  .map(q => q.quote)

res.forEach(x =>

  ...
```

Observable

```
res =
  stocks
  .filter(q => q.symbol == 'FB')
  .map(q => q.quote)

res.forEach(x =>

  ...
```

Single value (1)

Object

```
var y = f(x);
var z = g(y);
```

Promise

```
fAsync(x).then(...);
gAsync(y).then(...);
```

Synchronous   **Asynchronous**

http://jonalvarezz.github.io/presentation-reactive-programming/#29
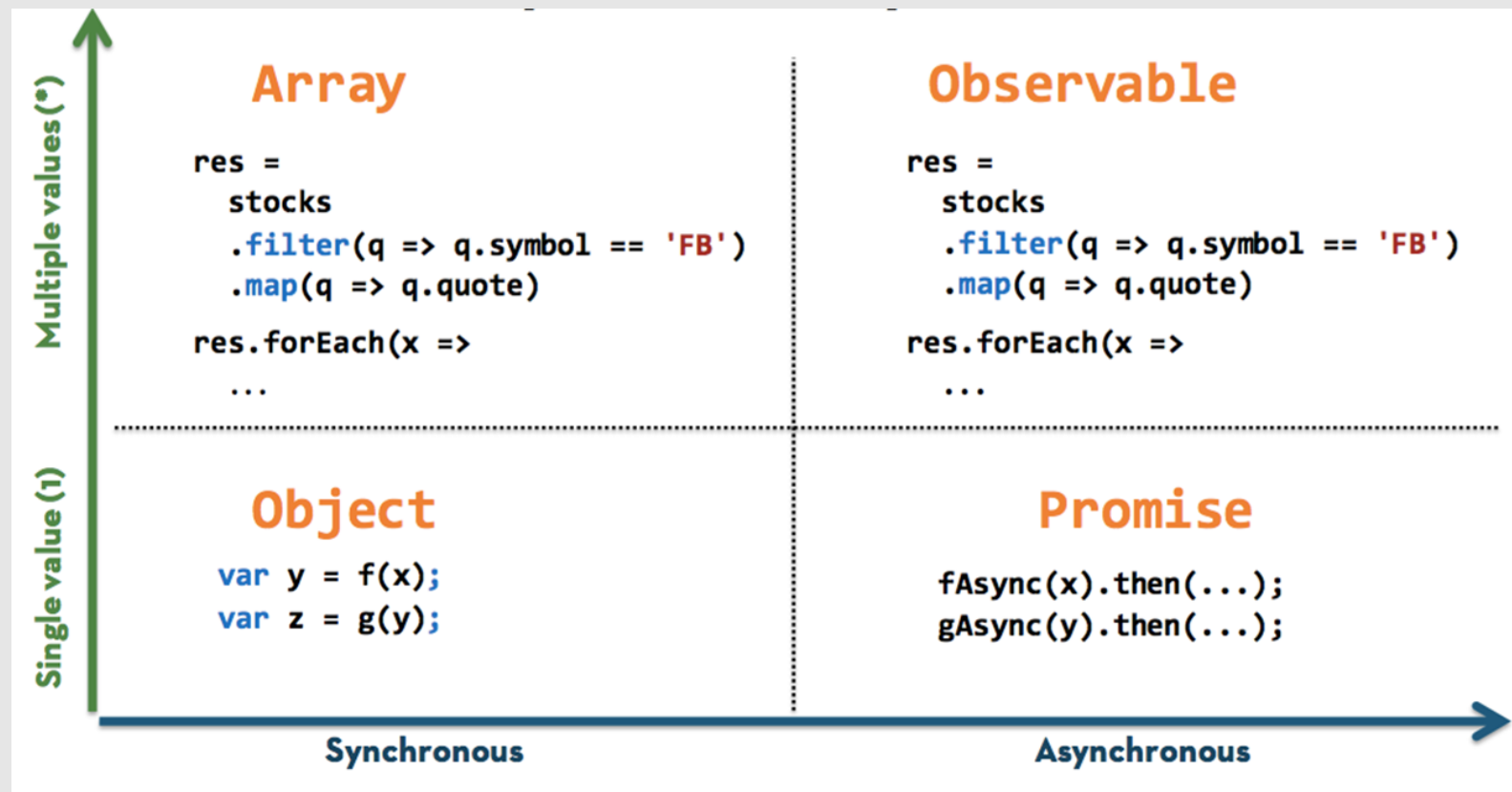
Observables can model

- events

- async server requests

- animations

Adapt Push APIs to Observable

DOM Events

Websockets

Server-sent Events

Node Streams   → Observable

Service Workers

jQuery Events

XMLHttpRequest

setInterval

promises vs observables

Promises	Observables
• Handles a single event when an *async* operation completes or fails • Returns a single value • Supported in ES6	• Handles a stream (zero or more than zero events) • Offers the features provided by promises and more • They are cancellable • They have useful operators like map, reduce, retry...

observer

```
observable.suscribe(

    //new data

    data =>  console.log(data),

    //error

    error => console.log(data),

    //completed

    console.log ("done");

);
```

optional

optional

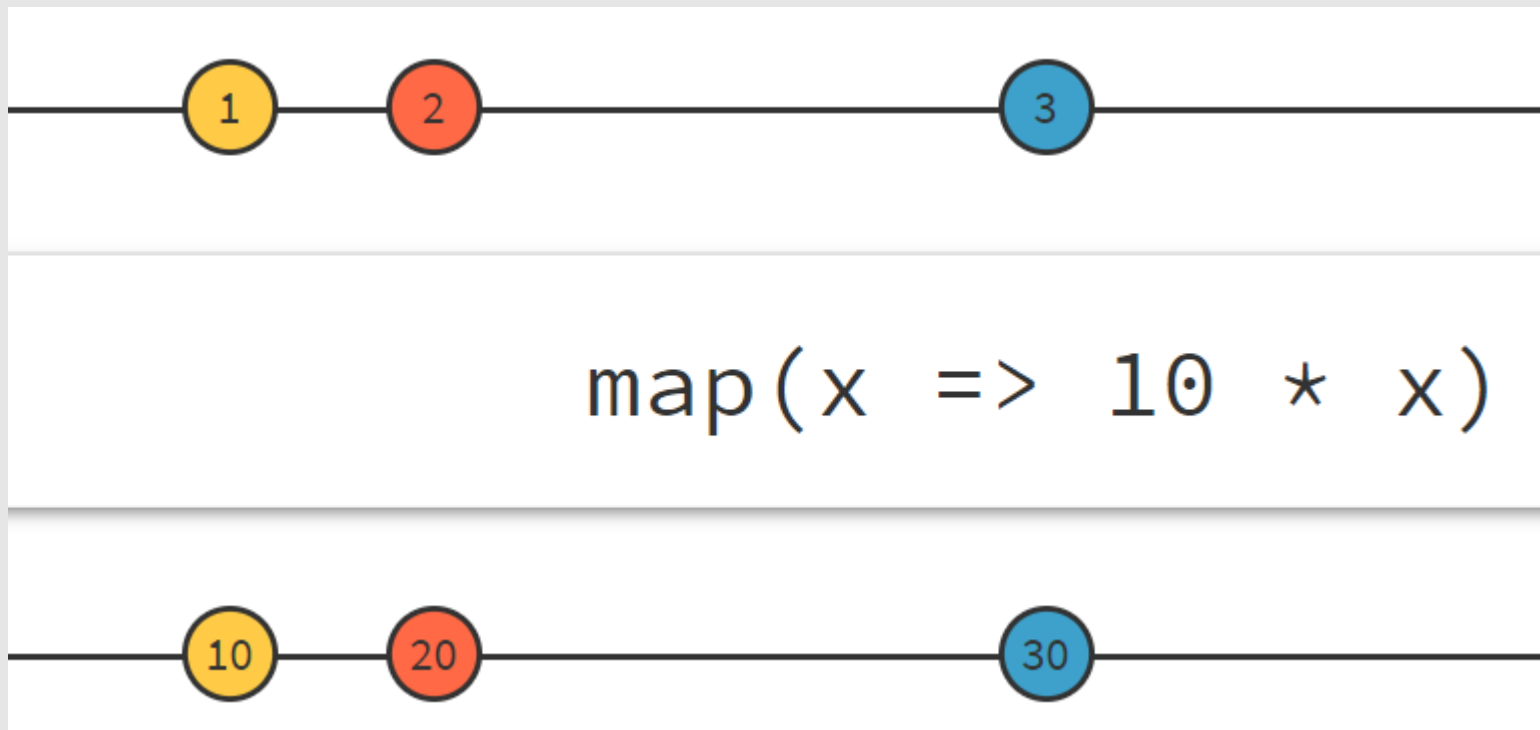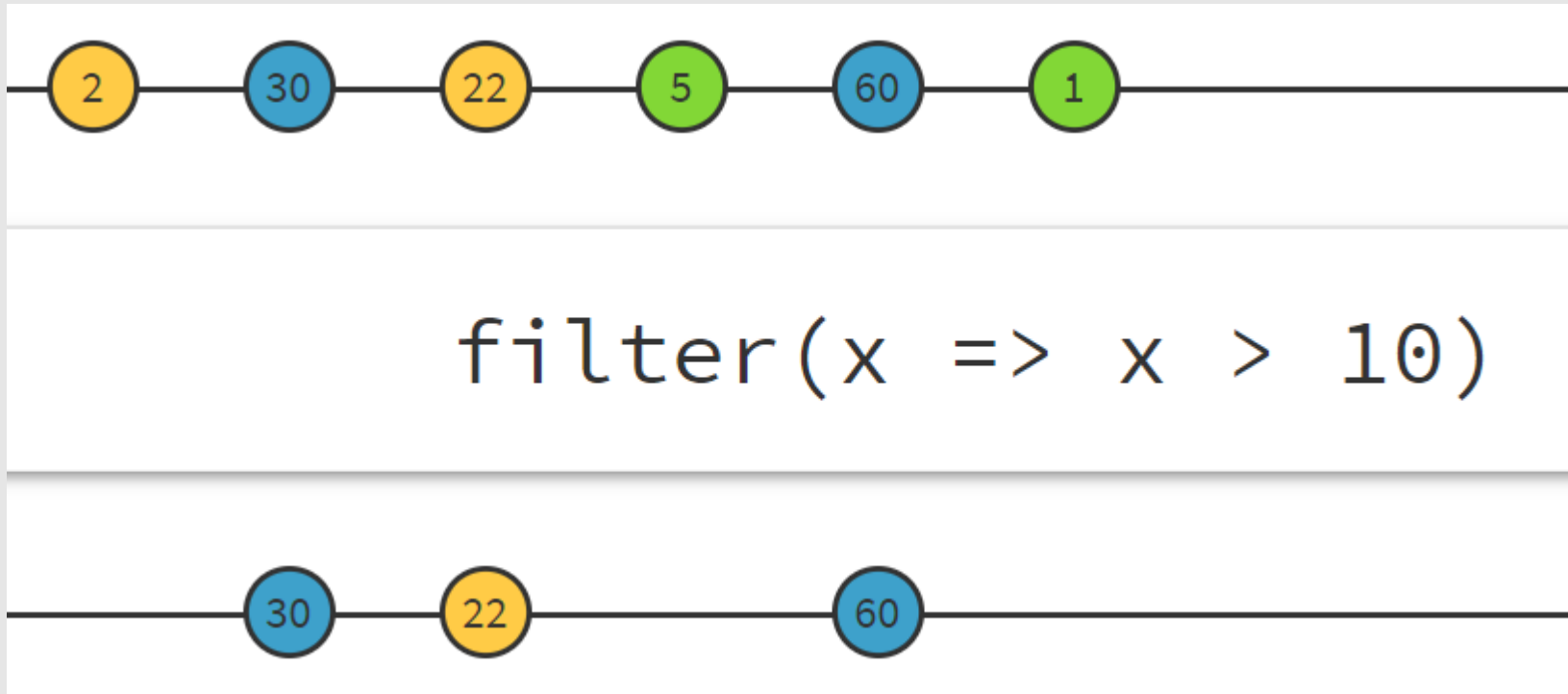http://reactivex.io/rxjs/class/es6/MiscJSDoc.js~ObserverDoc.html

Observables are cancellables

```
let suscription = observable.suscribe(
  //observer
);
```

```
suscription.unsuscribe
```

Observables have cool functions

$map(x => 10 * x)$

Observables have cool functions

Observables have lots of them

http://rxmarbles.com/

http://reactivex.io/rxjs/class/es6/Observable.js~Observable.html

Learning resources

Learning resources

■youtube -> Netflix talk

https://www.youtube.com/watch?v=XRYN2xt11Ek

Jafar Husain
@jhusain

Technical Lead at Netflix, Architect of
@falcorjs, TC-39 Rep.

■course ->

ASYNCHRONOUS PROGRAMMING IN JAVASCRIPT WITH RX.JS AND OBSERVABLES

■exercises online -> http://reactivex.io/learnrx/

Shaping the future of digital business

GFT Internal Technical Training

Eduardo García Ibaseta

eduardo.garcia-ibaseta@gft.com
+34 935 639476

GFT IT Consulting, S.L.
Av. Alcalde Barnils, 69-71
08174 Sant Cugat del Vallès (BARCELONA)