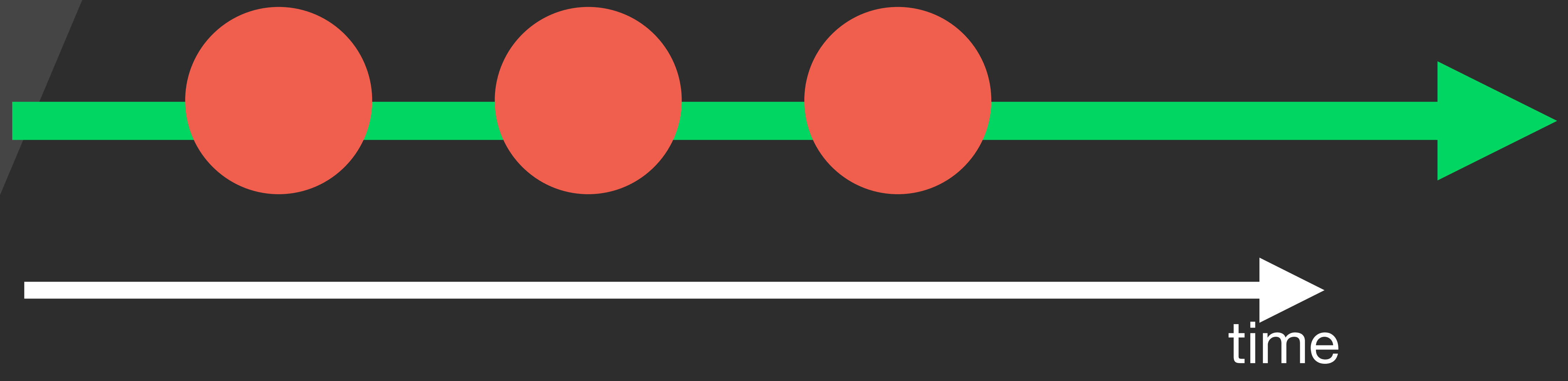
RXJS

RXJS helps us with our async code!

What RXJS is used for

academeez

- RXJS is used to help us with our async code in Javascript
- If you followed our lesson about classifying your async code, you will recall that we represent our async code with marble diagram.



- Unlike promise in RXJS we can send multiple data to the listeners
- RXJS can help us send data to listeners, that data can be sync or async, there can be zero one or infinite amount of data sent.
- We recommend reviewing the chapter about promises before taking this lesson.

Consuming data

- We can consume data in many ways, but those ways can be classified to these types:
    - We are actively asking for data - pull data (For example when we are calling a function)
    - We are passively getting data - data is pushed to us (For example promises)

RXJS pushing data

academeez

- Unlike functions when we call them to ask for data - data pull
- In RXJS data will be pushed to us - data push
- Which means listeners will be attached to a source that push data to them
- Recall our promise lesson, when we push data, we have a source that shouts, and listeners that attach a callback to the shout.
- In this aspect RXJS works the same and we can look at it as a shouter-listener pattern

Promise pushing data

academeez

- Shouter:

```
const helloPromise: Promise<string> = new Promise((resolve) => {
  resolve("Hello listeners! I'm a promise!");
});
```

- Listener:

```
helloPromise.then((msg: string) => {
  console.log(msg); // Hello listeners! I'm a promise!
});
```

Observable pushing data

academeez

- Shouter:

```
import { Observable } from 'rxjs';

const helloObservable: Observable<string> = new Observable((observer) => {
  observer.next("Hello listeners! I'm an RXJS Observable");
  observer.next('Hello again!');
  observer.next('I said hey!');
});
```

Observable pushing data

academeez

- listener:

```
helloObservable.subscribe((msg: string) => {
  console.log(msg);
});

// Hello listeners! I'm an RXJS Observable
// Hello again!
// I said hey!
```

Observable VS Promise 1

- Observable can emit multiple values
- Promise can only resolve once

Observable VS Promise - EX

- Using the previous example, place logs before and after subscribing to promise
- Using the previous example, place logs before and after subscribing to an Observable.
- Did you notice another difference between Observables and Promises?

Observable VS Promise 2

academeez

- Promise - the listeners will always be called async
- RXJS - the listeners will be called sync or async

Summary

academeez

- RXJS can be used to push data to listeners
- Unlike promise we can push multiple data pulses
- That data can by sync or async
- We are not done yet! There is plenty more to learn about this awesome library!

academeez

Thank You

Next Lesson: Observables