

# Observables

## 3. Observable wrapping setInterval

# Observable - setInterval

- In this exercise we will create an Observable that emits an **hello** message to the listeners every second
- Shout hello every second

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

const intervalObservable: Observable<string> = new Observable((observer) => {
  setInterval(() => {
    observer.next('Hello from Observable');
  }, 1000)
});
```

# Listen to the observable

- To listen to the observable we created before

```
intervalObservable.subscribe((msg) => {  
  console.log(msg);  
});
```

# Amount the Observable function called

- The Observable constructor gets the method which created the data to be pushed to the listeners
- In this exercise we will try to figure out how many times that method is called by placing a console message in that function
  - How many times if we have 0 listeners?
  - Hoe many times if we have X listeners?

# Observable is lazy

- Observable async method will not run if there are no listeners
- Unlike Promise which runs the async method right away synchronously
- When you attach a listener it will call the async method right away (sync)

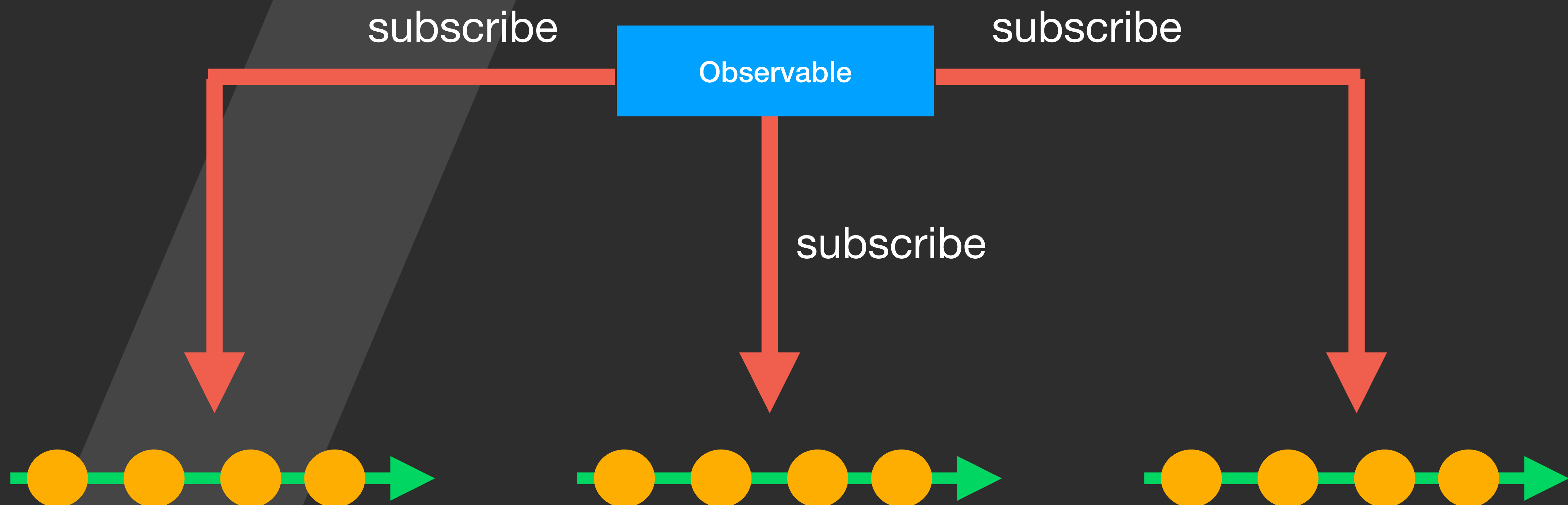
# Observable async method called many times

- The Observable async method will be called for every listener
- Unlike Promise where the async method will be called once regardless of the amount of listeners

# Observable data stream duplicates!

- You can think of the Observable like a function
- subscribe will call that function
- The function will return a data stream for every listener that called the function
- For 100 subscribe calls we will have 100 data streams
- In our example calling 100 subscribe will create 100 setInterval

# Observable data stream duplicates!





# Summary

- We learned that the Observable async method will behave lazy and will not be called unless we attach a listener
- We learned that the number of times the async method is called will be equal to the number of listeners
- We learned that Observable duplicated the data stream for every listener
- When we attach a listeners the async method is called sync

# Thank You

**Next Lesson: 4. Closing Observable**