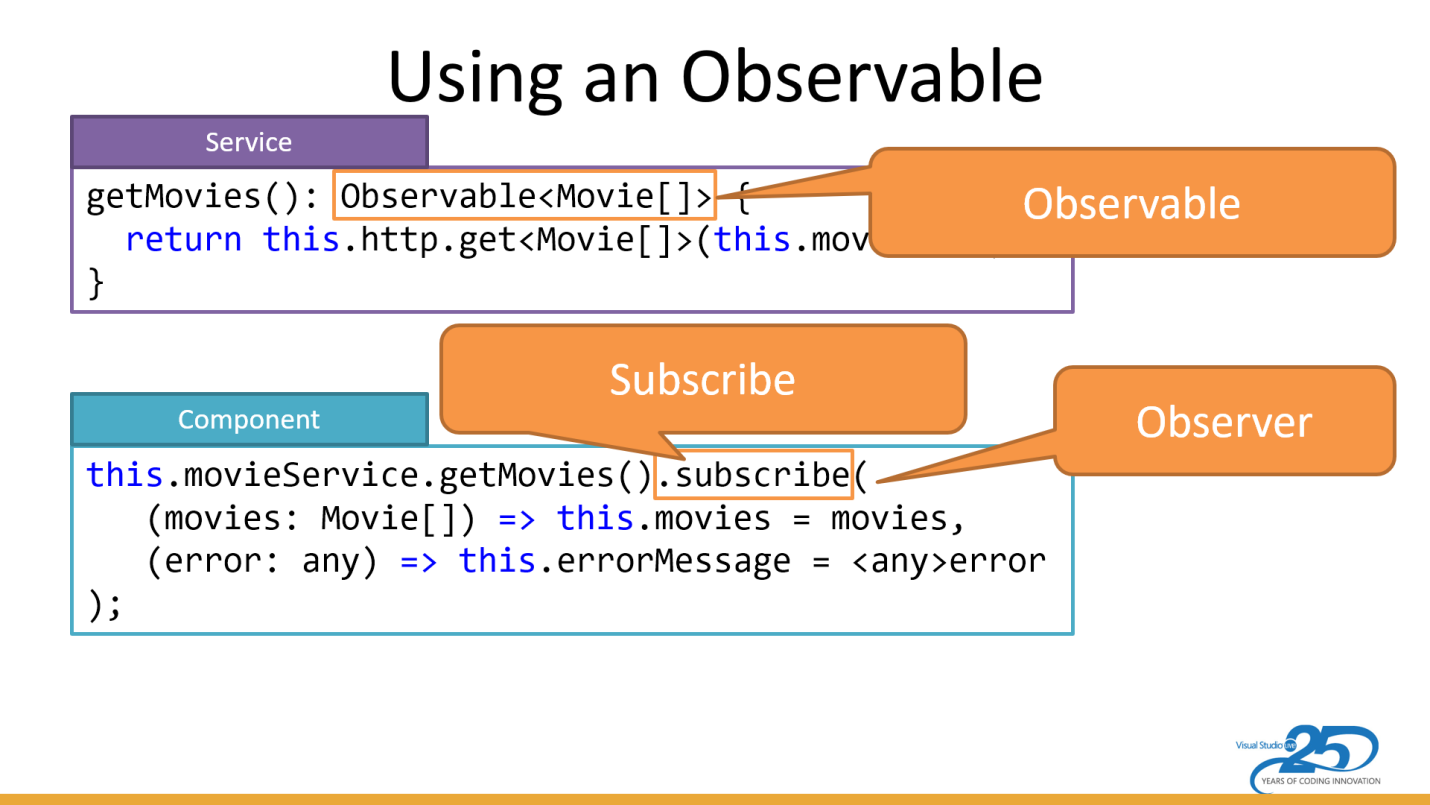
[What is observable, observer and subscribe in angular?](https://stackoverflow.com/questions/51520584/what-is-observable-observer-and-subscribe-in-angular)



As seen above ... an **Observable** is a stream of events or data. They are often returned from Angular methods, such as the http.get and the myinputBox.valueChanges.

**Subscribing** "kicks off" the observable stream. Without a subscribe (or an async pipe) the stream won't start emitting values. It's similar to subscribing to a newspaper or magazine ... you won't start getting them until you subscribe.

The subscribe method takes in an **observer**. An observer has three methods:

* The method to process each time an item is emitted from the observable.
* The method to process any error that occurs.
* The method to clean up anything when the observer completes. This last one is seldom used when working with Angular's observables.

Hope this helps.

Trying to explain with a really simple example:-

1. **Observable** is like a youtube channel of someone else. (( It uploads new videos(data) from time to time, **so it is a data source** for you))
2. Your youtube account is an **Observer**
3. Your youtube account **(Observer)** can only get notifications about whether someone else's youtube channel **(Observable)** has uploaded a new video **(has new data)** or made a livestream **(new event)** only if you have **subscribed** to that channel

**(Observer subscribes Observable to listen for new data/any event)**

where observable is a data source, subscribe is like a method/function , Observer is generally on your side

An **Observable** can be thought of as various data sources(ex: (userInputs)Events, HttpRequests etc).

here creating our custom observable.

var observable = Observable.create((observer: any)=>{

observer.next("Hii")

observer.next("how are you")

observer.complete()

observer.next("This will not send to observer")

});

* next() used to emit values to observer
* complete() indicates that completion of observable is notified.

An **Observer** is basically who subscribes to Observable.

observable.subscribe(

(data: any) => console.log(data), // for handling data

(error: any) => console.log(error), // for handling error

() => console.log('completed'); // for handling completion

);

