**RXJS (Angular)**

**Observable:**

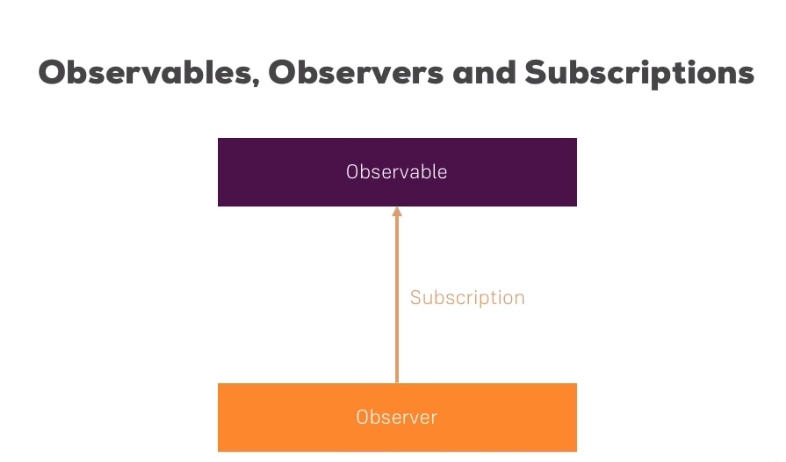
An Observable is an entity that emits (or publishes) multiple data values (stream of data) over time and asynchronously. (wrapper around the stream of data)

**Observers:**

Observers are also called listeners (or consumers) as they can listen or subscribe to get the observed data.

**Subscriptions:**

Subscriptions are objects that are returned when you subscribe to an Observable. They contain many methods such as the unsubscribe() method that you can call to unsubscribe from receving published values from the Observable. (this is a function which will attach observer to the observable)



**A quick recap of what an Observable is:**

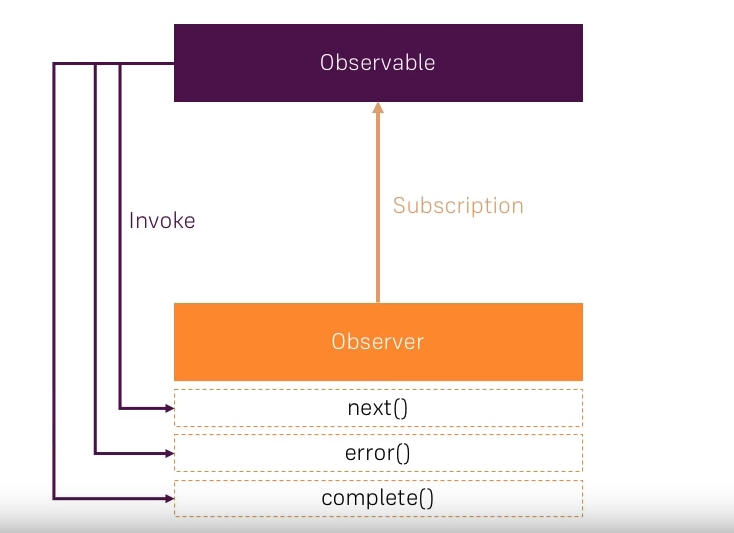
A “Stream” or “Set” of 0 to many values of any type (Numbers, Strings, Array, Objects…).

Functions that ties producer to consumer.

“Lazy” because the logic inside is not executed until someone is subscribing to the stream. Also called a cold observable.

This is the most basic building block of RxJS for programming with asynchronous data streams. Anything can be a stream and wrapped into an Observable: network calls, user inputs, variables, properties, data structures, etc.

**Defining observers**

A handler for receiving observable notifications implements the Observer interface. It is an object that defines callback methods to handle the three types of notifications that an observable can send:

|  |  |
| --- | --- |
| **NOTIFICATION TYPE** | **DESCRIPTION** |
| **next** | Required. A handler for each delivered value. Called zero or more times after execution starts. |
| **error** | Optional. A handler for an error notification. An error halts execution of the observable instance. |
| **complete** | Optional. A handler for the execution-complete notification. Delayed values can continue to be delivered to the next handler after execution is complete. |

An observer object can define any combination of these handlers. If you don't supply a handler for a notification type, the observer ignores notifications of that type.

