Angular questions:

1. **Observable**

* You can subscribe and let the stream of data to come in non-stop until the function complete has been called
* An Observable is an object that over time and asynchronously emits multiple data values

1. **Promise**

#### Promise:

A Promise emits a single event at the completion or failure of an async operation.

* promise emits a single value
* A promise is Not Lazy A Promise cannot be cancelled

Observable:

An observer is like a stream and allows you to pass at least zero or more events where the callback is needed for each event.

* Observable is favored over Promise Emits multiple values over a time.
* The "Observable" is cold. It's not called until we're registered to it.
* You may cancel an Observable with the unsubscribe() method
* Observable provides a lot of efficient operators like map, foreach, filter, reduce, retry, retryWhen etc.

What is the difference between an observable and a promise?

Both Promises and Observables provide us with abstractions that help us deal with the asynchronous nature of our applications. However, there are important differences between the two:

* Observables can define both the setup and teardown aspects of asynchronous behavior.
* Observables are cancellable.
* Moreover, Observables can be retried using one of the retry operators provided by the API, such as retry and retryWhen. On the other hand, Promises require the caller to have access to the original function that returned the promise in order to have a retry capability.

#### Q: **What are RxJS Operators ?**

A stream is a key part of reactive programming. In simple terms,

#### Q: What is **Observers and Subscriptions** ?

Observers and Subscriptions are related to the concept of Observable.

#### Observers -Observer is a set of callbacks that know how to listen to the values of the Observable.

#### Subscription - Subscription is an observable execution

#### 5. What are different between of **Subject, BehaviorSubject and ReplaySubjec**t?

#### Subject :

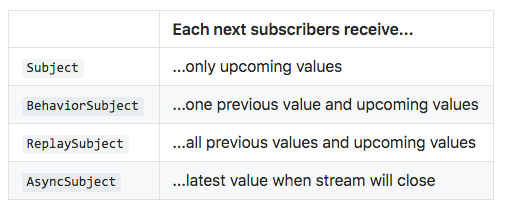
In case of Subject, Observers who are subscribed at a later date will not obtain the data values emitted prior to their subscription.

#### ReplaySubject :

In ReplaySubject, Observers who are subscribed at a later point will receive data values issued prior to their subscription. As it operates by using a buffer that holds the values emitted and re-emits them once new Observers are subscribed.

#### BehaviorSubject :

BehaviorSubject functions similar to ReplaySubject but only re-issues the last emitted value. So you're interested in the last / current value of the observer, if BehaviorSubject is useful.



1. Lifecycle hooks
2. Router params
3. Angular Architecture



1. Dependancy injection
2. Subject and observable

* A Subject is a special type of Observable that allows values to be multicasted to many Observers. Subjects are like EventEmitters.
* Every Subject is an Observable and an Observer. You can subscribe to a Subject, and you can call next to feed values as well as error and complete.

1. Behavioral subject- A variant of Subject that requires an initial value and emits its current value whenever it is subscribed to.
2. Replaysubject- A variant of Subject that "replays" or emits old values to new subscribers. It buffers a set number of values and will emit those values immediately to any new subscribers in addition to emitting new values to existing subscribers.
3. Oberverble and oberver

* <https://stackoverflow.com/questions/47537934/what-is-the-difference-between-a-observable-and-a-subject-in-rxjs>

1. Fucntion =>
2. Inline and block statements
3. == and ===
4. pseudo-class - used to define a special state of an element.

#### Angular 6 updates - New Rendering Engine:

#### Ivy - increases in speed and decreases in application size.