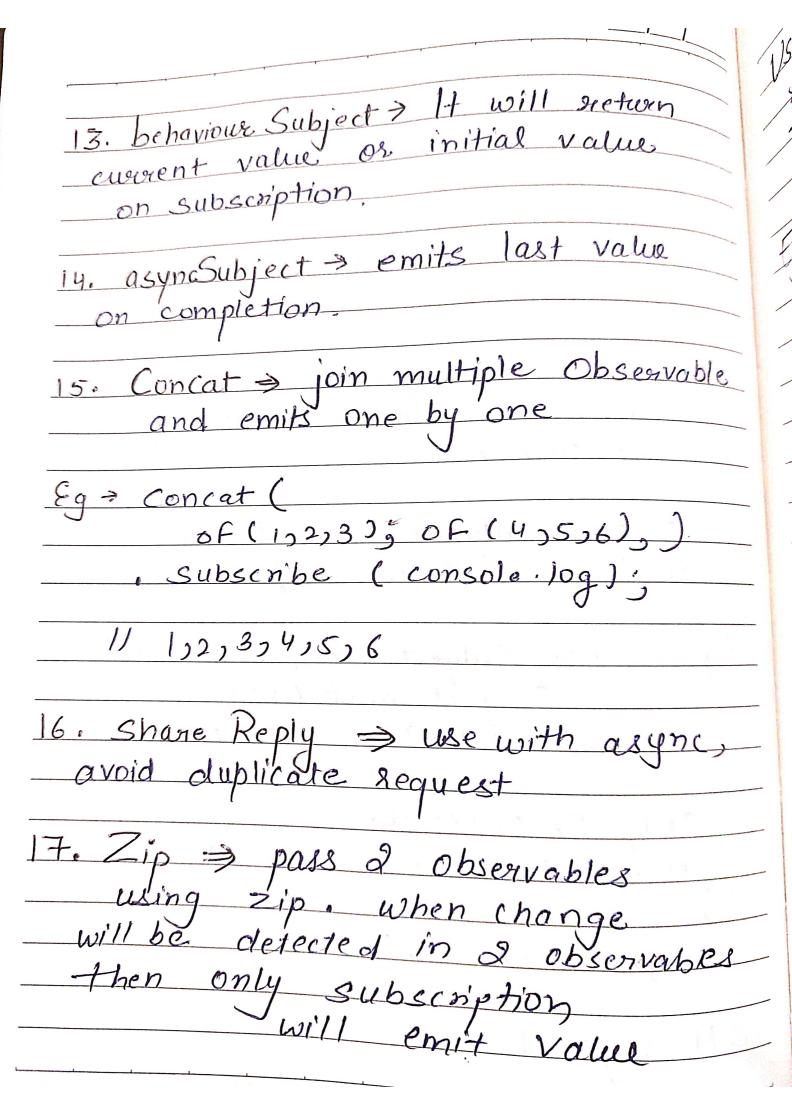
Date:/
RxJS Operators
1. Pipe > Function that takes an observable as its Input and seturn another Observable
2. Filter > Filter Source Observable before Subscribe
before subscribe
3. Tap > perform other side - effects/
modify The au a sie
Eg > obs = new Observable ((observer) => 5 Observer next(1)'
Observer o next (2)
observer. next (3); observer. complete () }). pipe (
tap (data => console.log (1+ap)+ data
filter (data => data > 2) tap (data => console.log ('filter'+date) map ((val => Return val as no x2 g)
map ((val =) seturn val as no x2 g)

Date:/
We use pipe to chain the tap operator
Which just logs value of source observable to console
charvable to console
01)500000
The use Liter to tilter out oberevable
The use of the property of the
ple me filter, to filter out observable according to over requirement.
4. Take > emits only the first
count values by the source
observable mentioned in take
operator.
5. Take Until > use for unsubscribe observable until not getting response
obcervable until not getting response
Can use for complete and destroy
can use for complete cybe exibe.
Can use for complete and destroy observable after subscribe
6. Take Last > emits only last
count values.
the state of the s
Fo Take While > emit value 16
condition is true.
Towns I have been

Date://
Eg > obs = of (1,2,3,4,5,6). pipe (
take while (val > val 23, true)
take Last (3)
take (2)
take Until (Subject)
) #
Output > 1,2 (take while)
4,5,6 (take last)
1,2 (take)
8. retry When > retry with condition. 15 you want to retry after few Seconds, not instantly then use retry When
It you want to retry after hew
seconds, not instantly then
use retry when
Foreg - It getting 404 a exper-
Foreg- If getting 404 gerror, then you should retry
J. J
9. From Event-handling Dom
events using target Element
events using tanget Element and event type

Date://
10. debounce Time > delays the value emitted by source.
12 101 Chanced 3
11. distinct Until Chargea
emits all items that are district
11. distinct Until Changed > emits all items that are distinct from previous items
Subject does not
12: Subject - Subject avec 11
Letwin the Current valle of
12: Subject - Subject does not return the current value of subscription.
It triggers only on next (value)
It triggers only on next (value) call and return output.
Eg > var Subject = new Subject()' Subject. next (1)'s 11 will not output any value
Subject. next (1);
Il will not outbut any value
11 WIII 1101 Carps
subject. subscribe ({
nex+ 1/1/2 -> Conse./pg. (V);
next: $(v) \Rightarrow \text{Conse.log}(v)'$
Subjec next (1);
provide a provid
11 Logs 1



Use Case - When you have to make Several http enequests at same time and need to complete all before emitting value; Eg = const source Dne = of ('Deepa'); const source Two = of ('Jyoti'); const eg = zip (source One Source Two pipe (delny (1000));
Leveral http requests at same time and need to complete all before emitting value; Fg > Const sourceDne = of ('Deepa'); const sourceTwo = of ('Jyoti'); const eg = zip (
time and need to complete all before emitting value; Eg > const sourceDne = of ('Deepa'); const sourceTwo = of ('Jyoti'); const eg = zip (
Eg > Const source Dne = Of ('Deepa'); const source Two = of ('Jyoti'); const eg = zip (
eonst eg = zip (
eonst eg = zip (
eonst eg = zip (
const eg = zip (source One Source Two pipe (delag (1000))
Source Two pipe (delay (1000))
Source Two pipe (delay (1000))
const subscribe = eg. subscribe(val => console.log (val)).
val => console. log (val)
and the second of the second o
18. fork Join > Mait for all observables to complete and then emit an array of last emitted values.
to complete and then emit an array
of last emitted values.
Join multiple Observable to a Single one.
Single one.
and the second of the second o
16 anu Observable came through
example took Toin will not
e atron Junii. Value
It any observable came through error g tork Join will not getween lang value

Flattening Operators
while using multiple observables
while using multiple observables into mested Subscription
1) Merge Map -> Simultaneously executes observables.
executes observables.
* Sequere of execution not quaranteed * If any observable throw error, It will emit other values.
& If any Observable throw Error
_ It will emit other values.
2) Concat Map > does not subscribe to next observable
subscribe to next Observable
until previous completes
* Execute one by one * sequest of execution guaranteed.
* sequest of execution quaranteed
3) exhaust Map > map to inner observable and ignore other until observable complete
Observable and ignore other until
Observable complète
4) Switch Map > try to resubscribe observable by cancelling previous
observable by cancelling previous