

33. ¿Qué son los operadores?

Los operadores son funciones que construyen sobre la fundación de los observables para tener una manipulación más sofisticada de las colecciones.

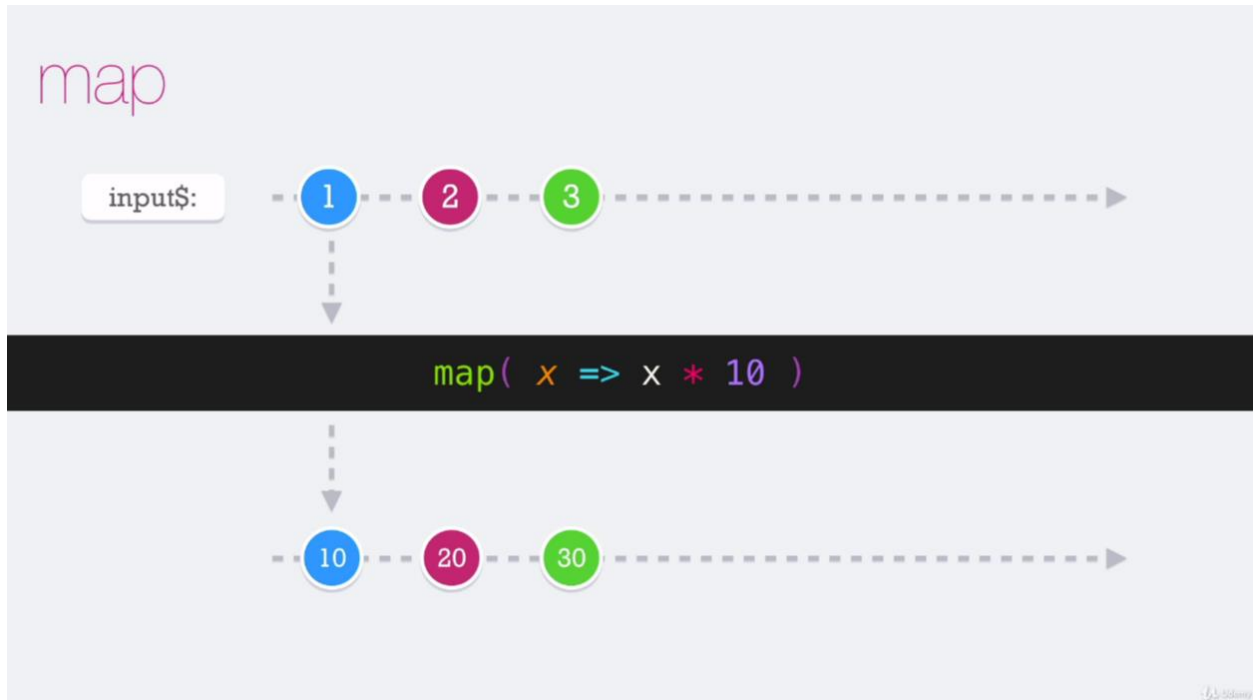
Y tenemos como una analogía por ejemplo que tenemos una manguera (que sería el observable o el stream de información) al cual le conectamos algún adaptador (que sería el operador conectado al stream de información) para transformar, filtrar o hacer algo con el agua (flujo de información).

Otra analogía podría ser una represa a la cual le entra el flujo de agua y del otro lado esta sale de otra manera.

34. map

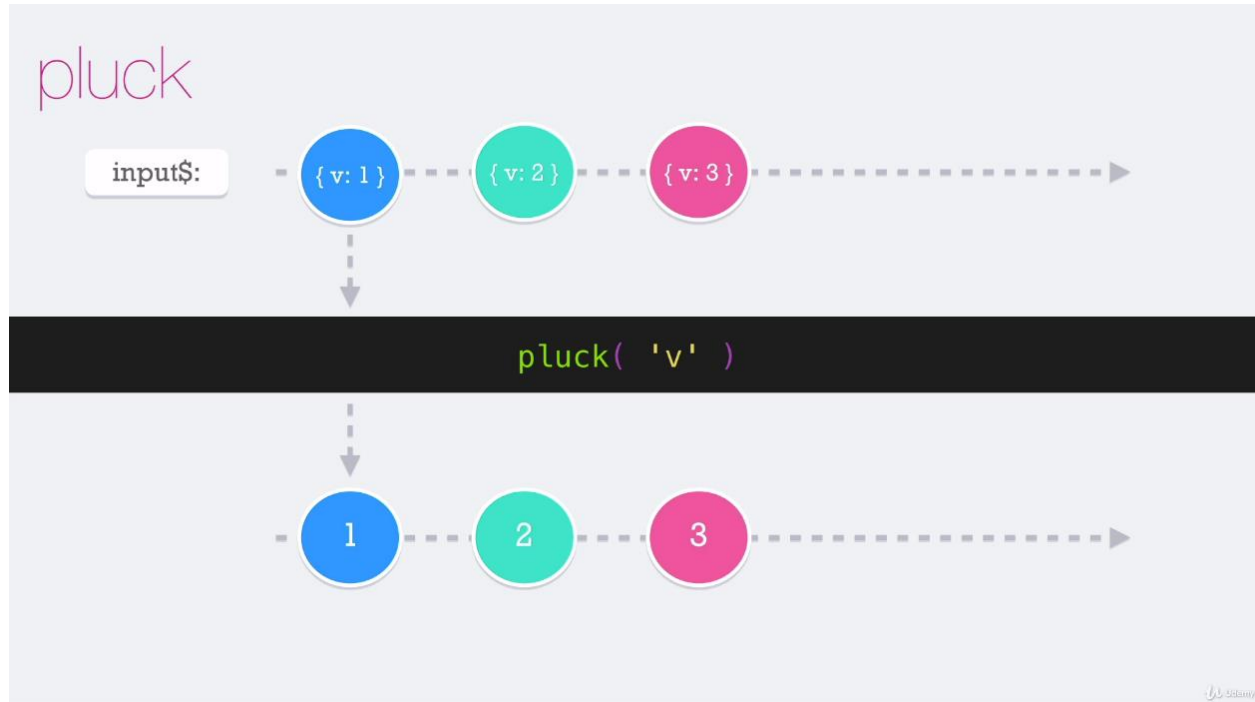
Es uno de los más comunes y más usado ya que nos permite transformar lo que recibimos o lo que emite el observable en algo que necesitamos, nos puede servir para extraer información, o bien para transformarla o si se desea regresar otra cosa totalmente diferente.

El operador map trabaja con cualquier tipo de dato que emita y puede regresar cualquier otro tipo de dato también diferente.



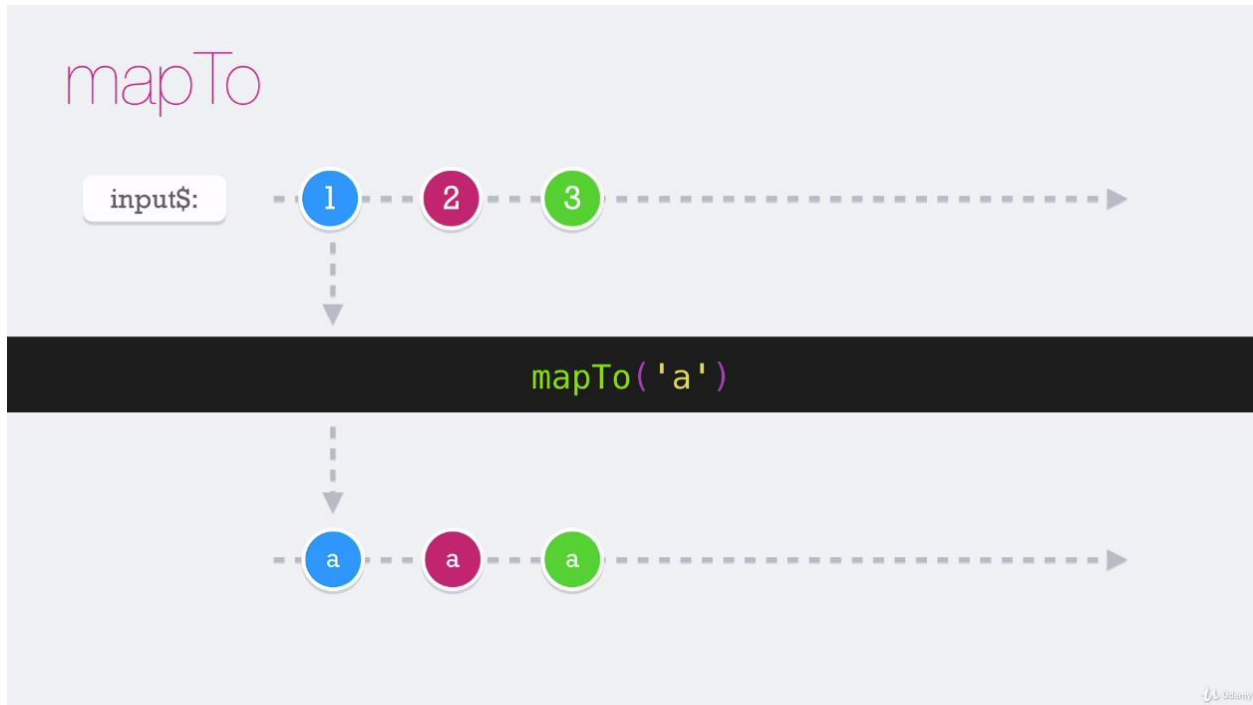
35. pluck

El operador pluck es sumamente útil cuando nosotros ocupamos extraer simplemente un valor del objeto que estamos recibiendo y que ese sea la nueva emisión o salida del observable. El pluck adicionalmente también puede trabajar con objetos anidados.



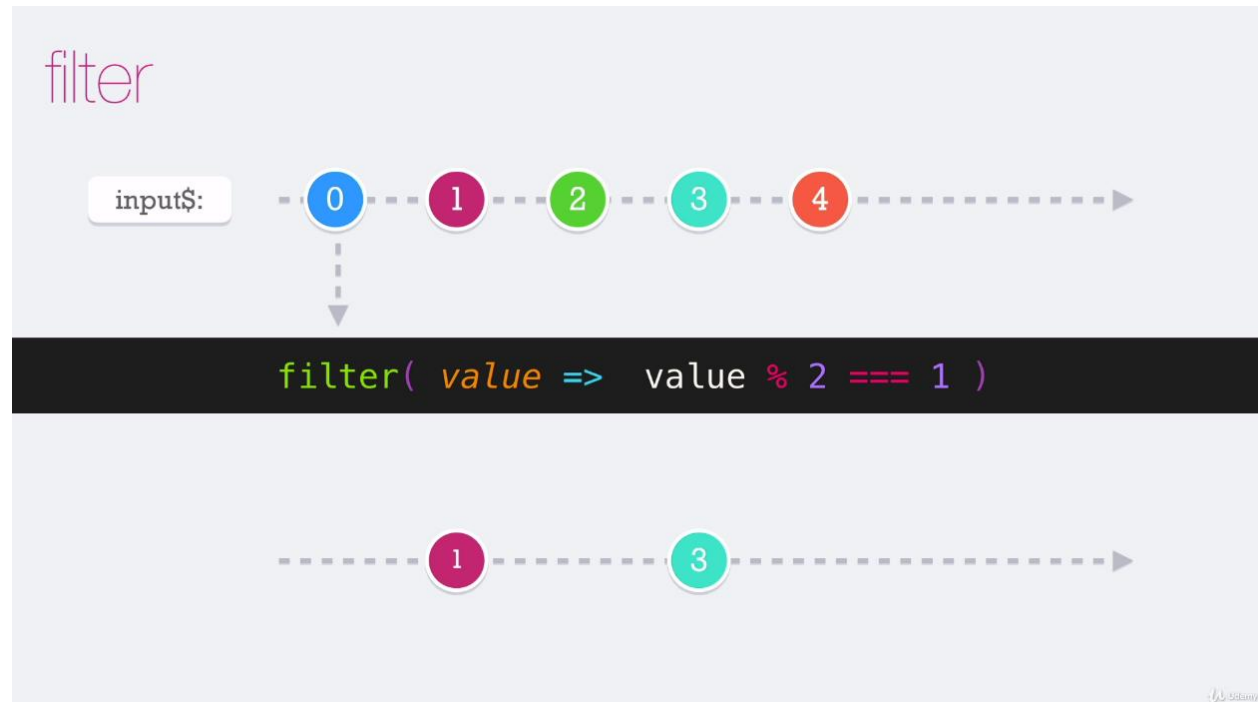
36. mapTo

El operador mapTo es bien interesante y al principio cuesta verle la utilidad ya que este nos permite transformar la entrada en una salida específica la cual pueden ser números, objetos, es decir, lo que sea lo que puede regresar este operador.



37. filter

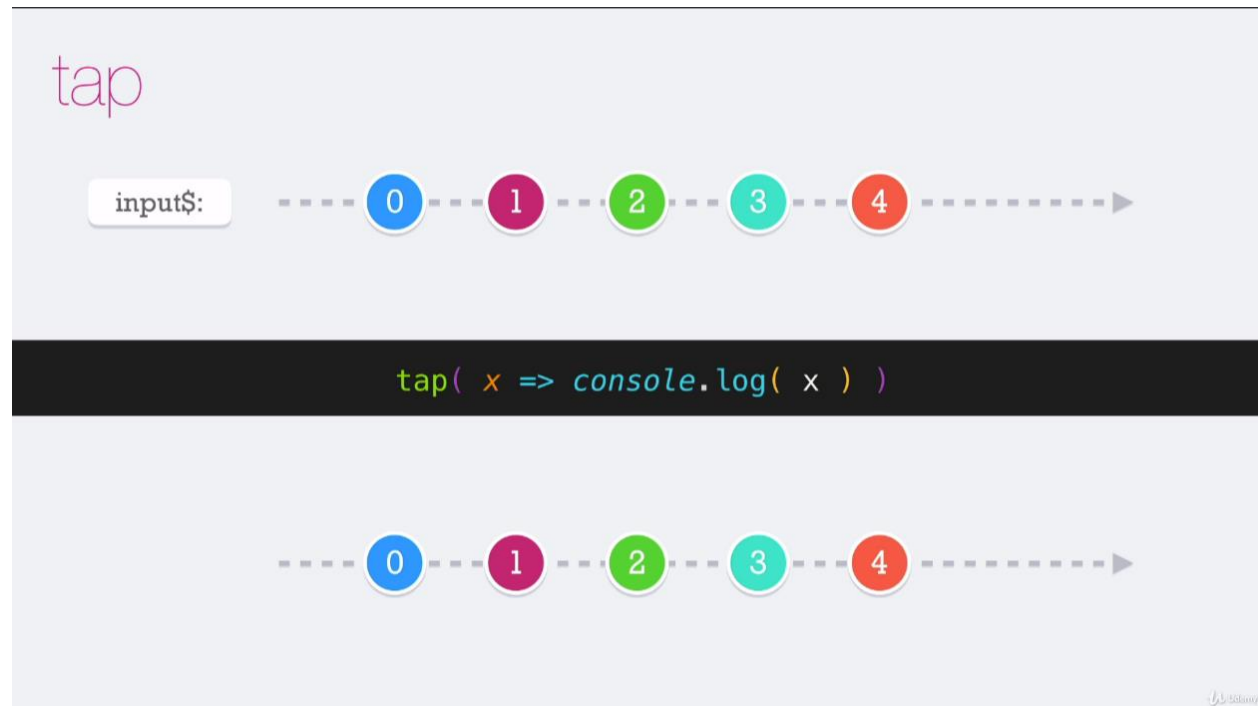
El operador `filter` como su nombre lo indica nos sirve para filtrar las emisiones de los valores que emite el observable. El siguiente ejemplo filtra y saca los valores impares.



39. tap

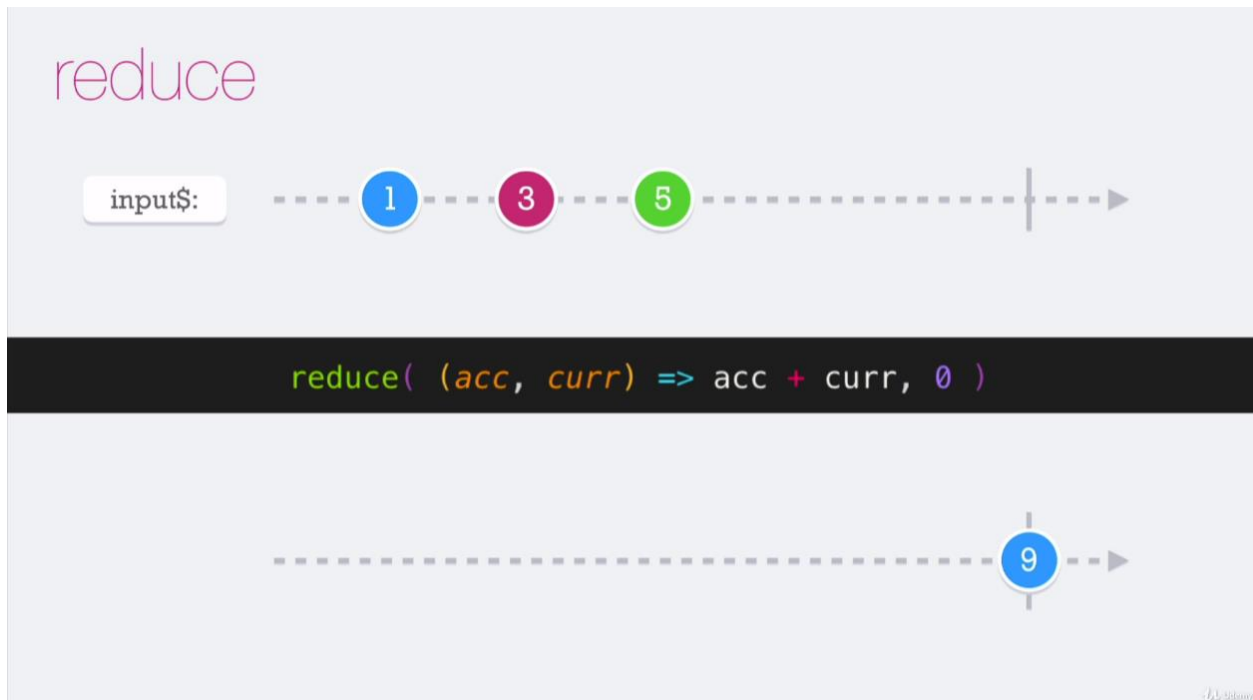
El operador `tap` es útil ya que nos sirve para ver como va fluyendo la información de nuestros observables. Básicamente el operador `tap` nos permite disparar efectos secundarios, pero hay que tener cuidado ya que puede ser que nosotros disparemos acciones accidentalmente. El principal uso del `tap` es disparar efectos secundarios como por ejemplo imprimir en consola el valor que tenemos, o bien que necesitemos disparar alguna acción cuando la información pasa a través o bien que necesitemos disparar alguna acción cuando sea emitido un nuevo valor en el cual está presente el `tap`.

También el `tap` es bien poderoso porque con este podemos deputar todo el proceso de inicio a fin, inclusive simular errores, `next` y `complete`.



42. reduce

El operador reduce básicamente hace lo mismo que el reduce en JavaScript, es decir, este lo que hace es aplicar una función acumuladora a las emisiones producidas por el observable. En el siguiente ejemplo lo que hace es que cada vez que se esta emitiendo un valor tenemos que en el acc es el valor acumulado, el curr es el current value o valor actual, a esto se le aplica la función que dice que va a retornar el valor acumulado más el actual, y el cero que se tiene al final indica básicamente cual es el valor inicial que tiene el acumulador, pero el operador no emite el valor hasta que este se completa.



43. scan

El operador scan es casi lo mismo que el operador reduce solo con una única diferencia que es que cuando los valores son emitidos por el observable inmediatamente van saliendo conforme van ingresando, y regresan su valor acumulado. En el siguiente ejemplo lo que hace es que cada vez que se está emitiendo un valor tenemos que en el acc es el valor acumulado, el curr es el current value o valor actual, a esto se le aplica la función que dice que va a retornar el valor acumulado más el actual, y el cero que se tiene al final indica básicamente cual es el valor inicial que tiene el acumulador.

