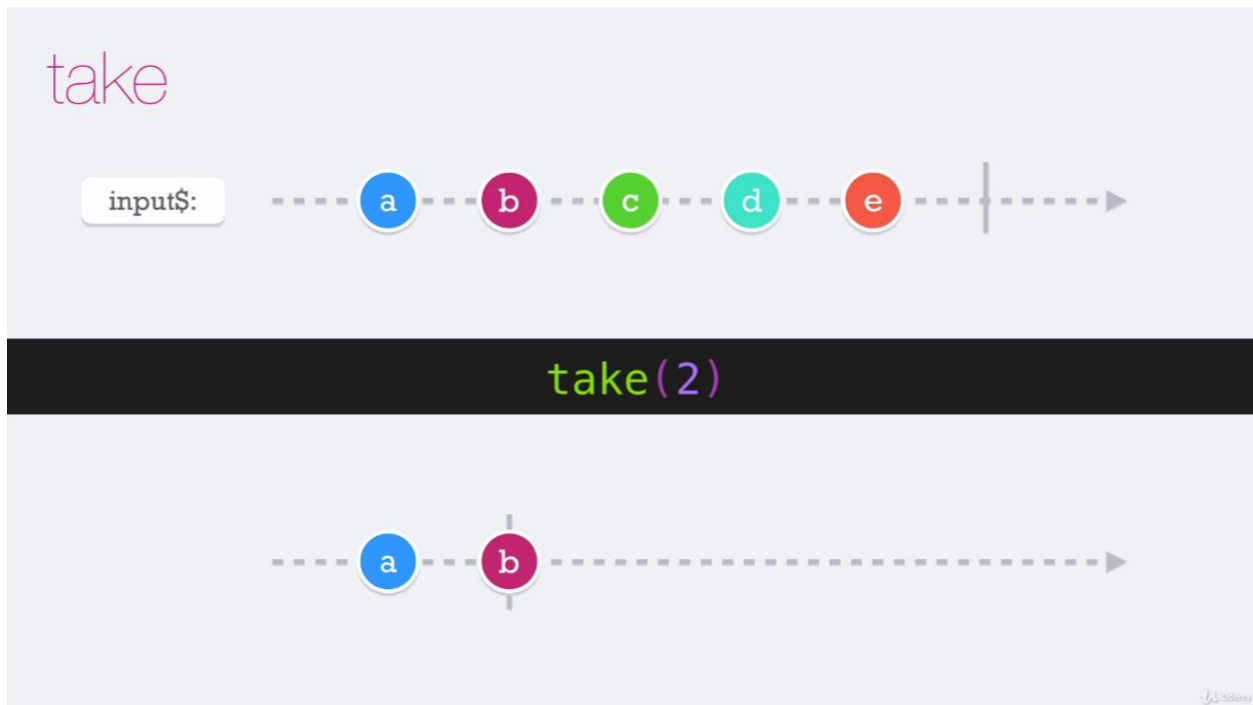


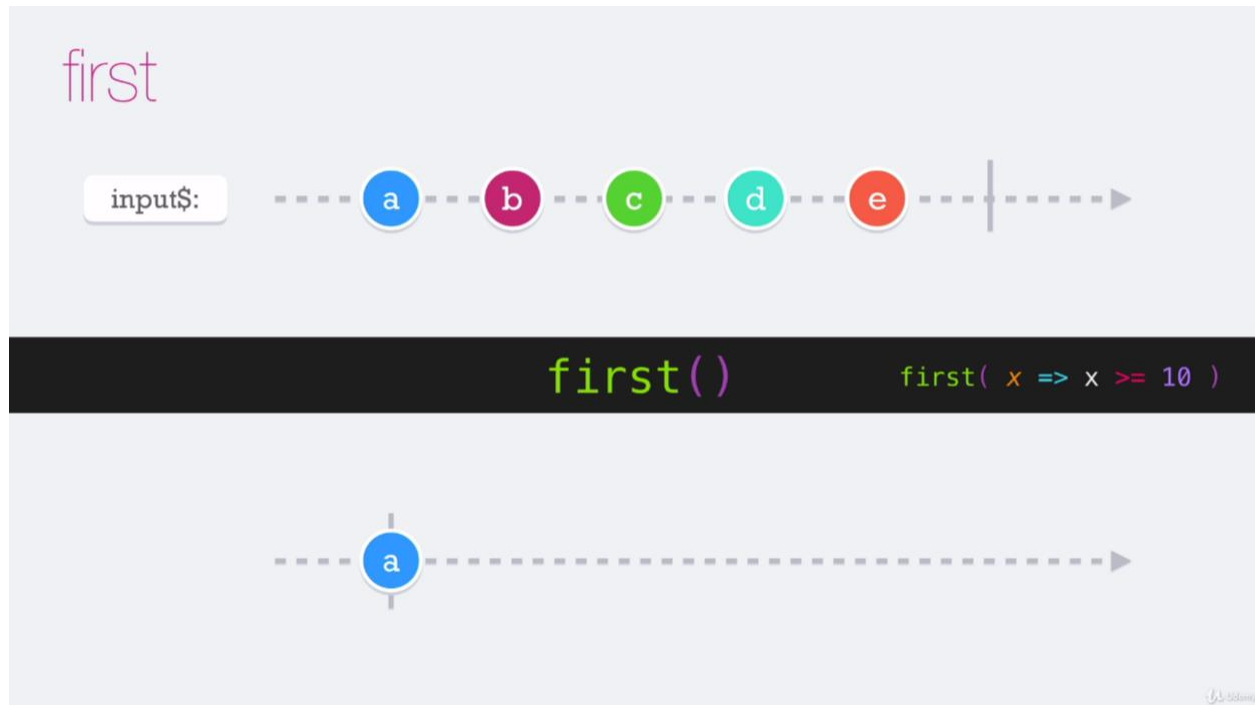
47. take

El operador `take` es sumamente útil cuando queremos limitar la cantidad de emisiones que un observable puede tener. Otra característica del `take` es que cancela la ejecución del observable. Por ejemplo, tenemos el siguiente observable que va a emitir varios valores los cuales con el uso del `take` limitamos a que solo vamos a tener 2 emisiones, por lo tanto cuando llega a esa cantidad deseada este se va a completar inclusive si el operador continua emitiendo valores o en un punto da error, y en caso de que diera error no importaría ya que anteriormente la suscripción se había completado.



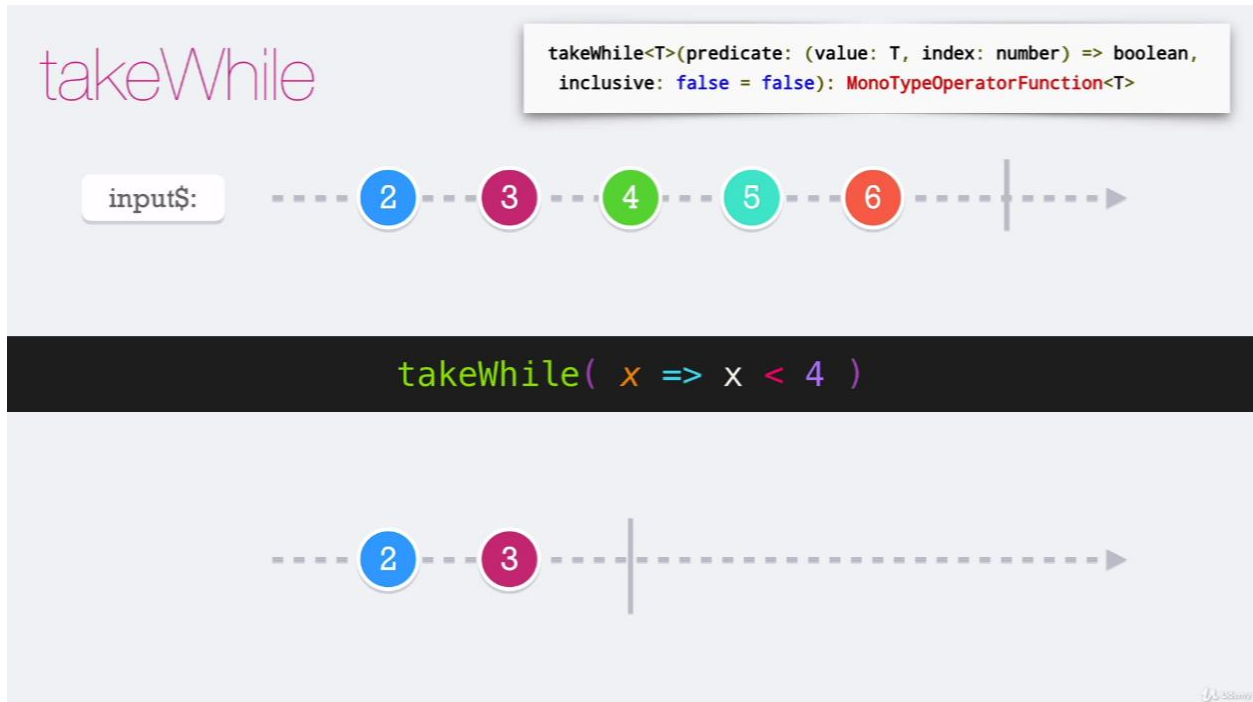
48. first

El operador first es un operador bastante sencillo, pero se le pueden hacer un par de configuraciones muy interesantes.



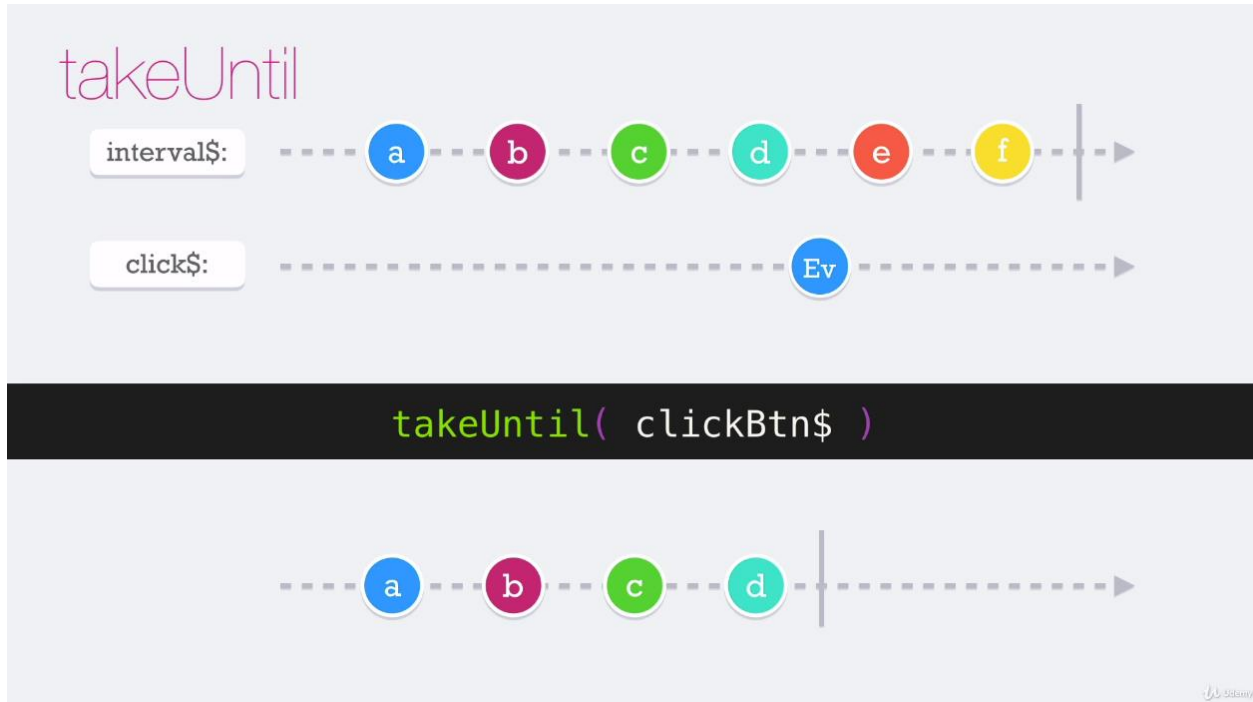
49. takeWhile

El operador `takeWhile` permite recibir valores mientras la condición se cumpla. Cabe anotar que si queremos el ultimo valor que cumplió la condición tenemos otro parámetro el cual es el `inclusive` que por defecto viene en `false`, pero lo podemos alterar para que nos devuelva el último valor que rompió la condición.



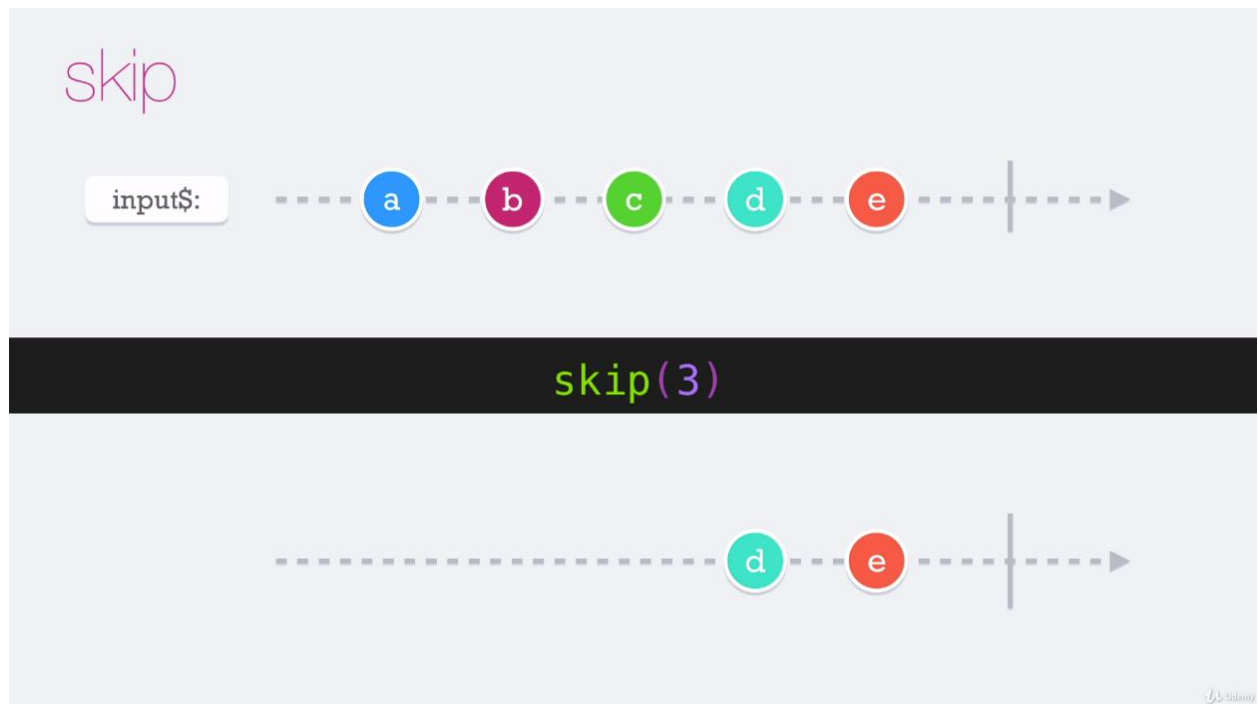
50. takeUntil

El operador `takeUntil` tiene una particularidad ya que este operador recibe como argumento otro observable. Y en sí este operador se traduce a que sigue recibiendo los valores y sigue emitiéndolos hasta que el segundo observable emita su primer valor.



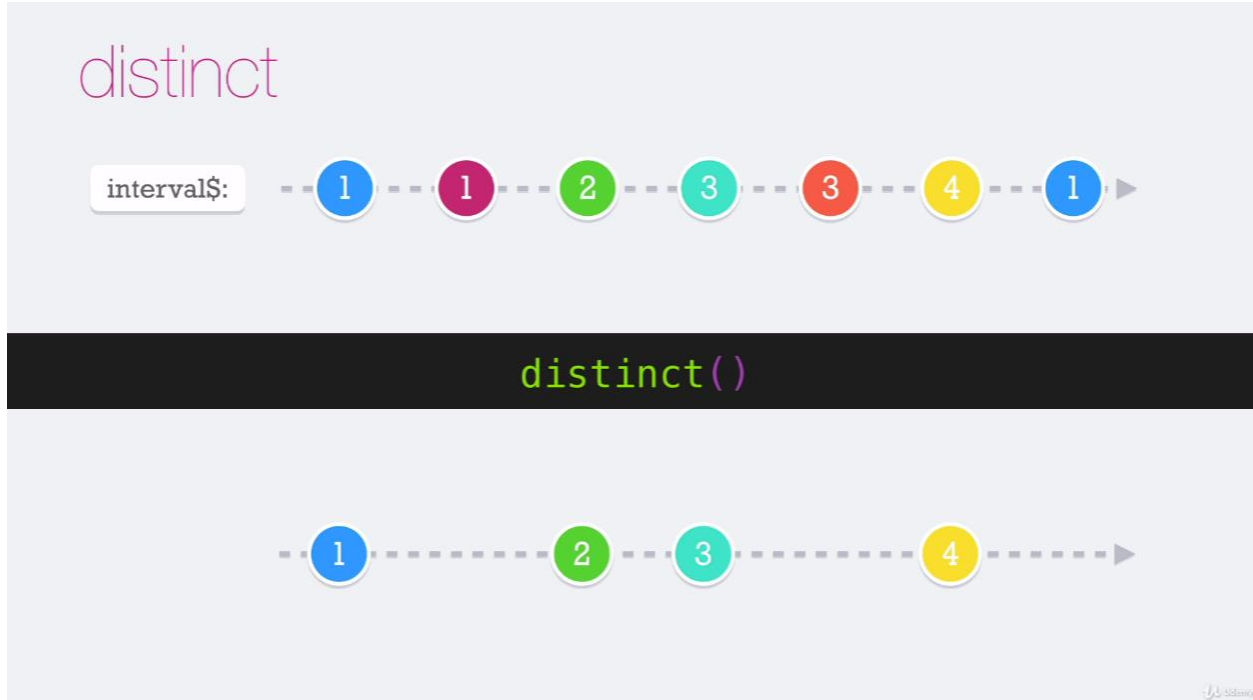
51. skip

El operador skip básicamente sirve para saltar u omitir x cantidad de emisiones iniciales.



52. distinct

El operador `distinct` es un operador el cual deja pasar únicamente los valores que no han sido previamente emitidos por el observable. Cabe resalta que el `distinct` usa el operador o condición estricto de JavaScript (`===`).



53. distinctUntilChanged

El operador `distinctUntilChanged` es muy parecido al operador `distinct` pero tiene una diferencia marcada y es que el `distinctUntilChanged` emite valores siempre y cuando la emisión anterior no sea la misma.

distinctUntilChanged



`distinctUntilChanged()`



54. distinctUntilKeyChanged

El operador `distinctUntilKeyChanged` tiene el mismo concepto que el `distinctUntilChanged` solo que este va a estar pendiente de la propiedad que se le indique, como se muestra en el ejemplo en este caso esta pendiente de la propiedad `k`.

distinctUntilKeyChanged



```
distinctUntilKeyChanged('k')
```

