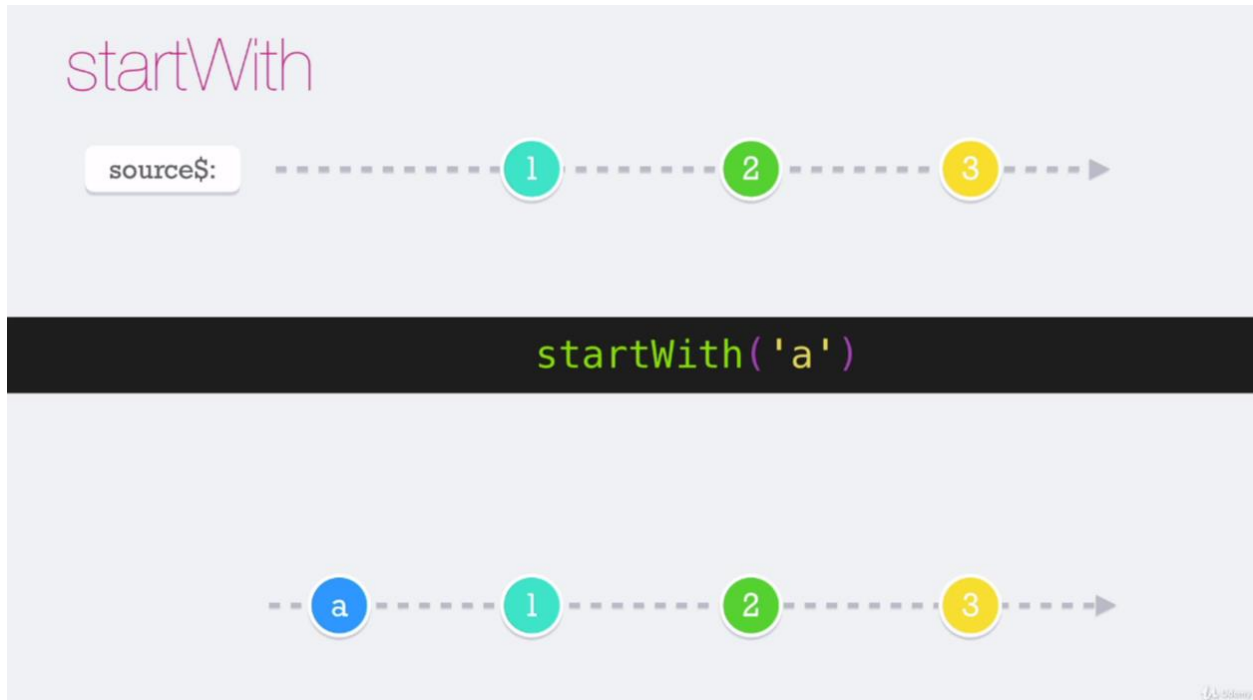


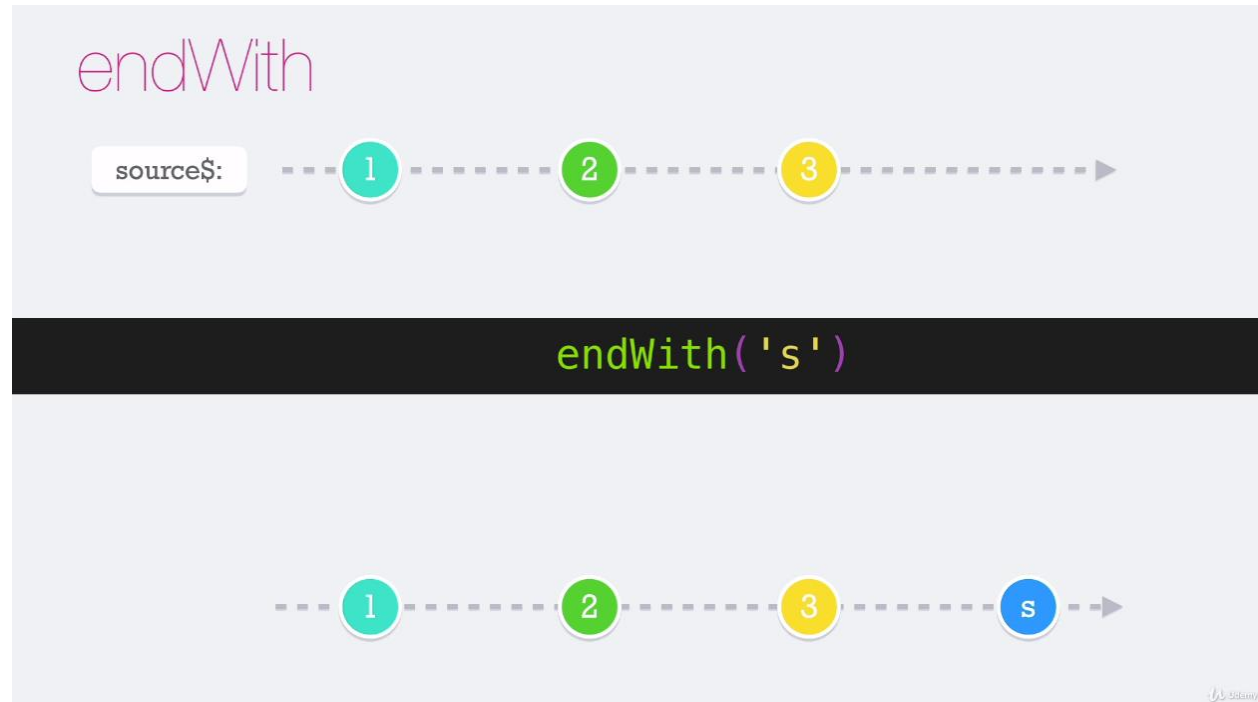
88. startWith

El operador `startWith` nos permite hacer una emisión antes de que el observable inicie a emitir, aunque sea un valor síncrono. En pocas palabras lo que hace el operador es que la primera emisión va a tener el valor que se le especifique los cuales pueden ser strings, pueden ser números, pueden ser objetos, puede ser inclusive otro observable, pero en este caso tendríamos que pasarlo por el operador `mergeAll`.



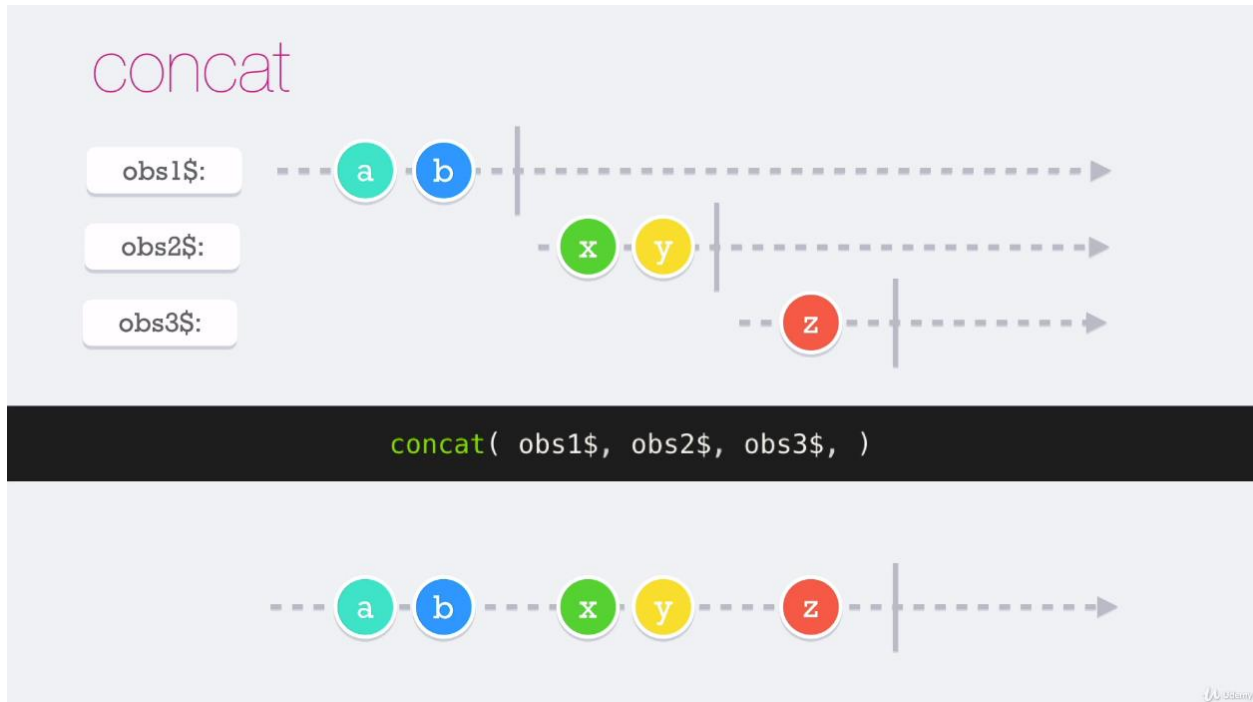
89. endWith

El operador `endWith` es muy similar al `startWith` ya que es un operador que básicamente lo que hace es que antes de que se complete el observable va a emitir el valor que tengamos asignado los cuales pueden ser strings, pueden ser números, pueden ser objetos, puede ser inclusive otro observable, pero en este caso tendríamos que pasarlo por el operador `mergeAll`.



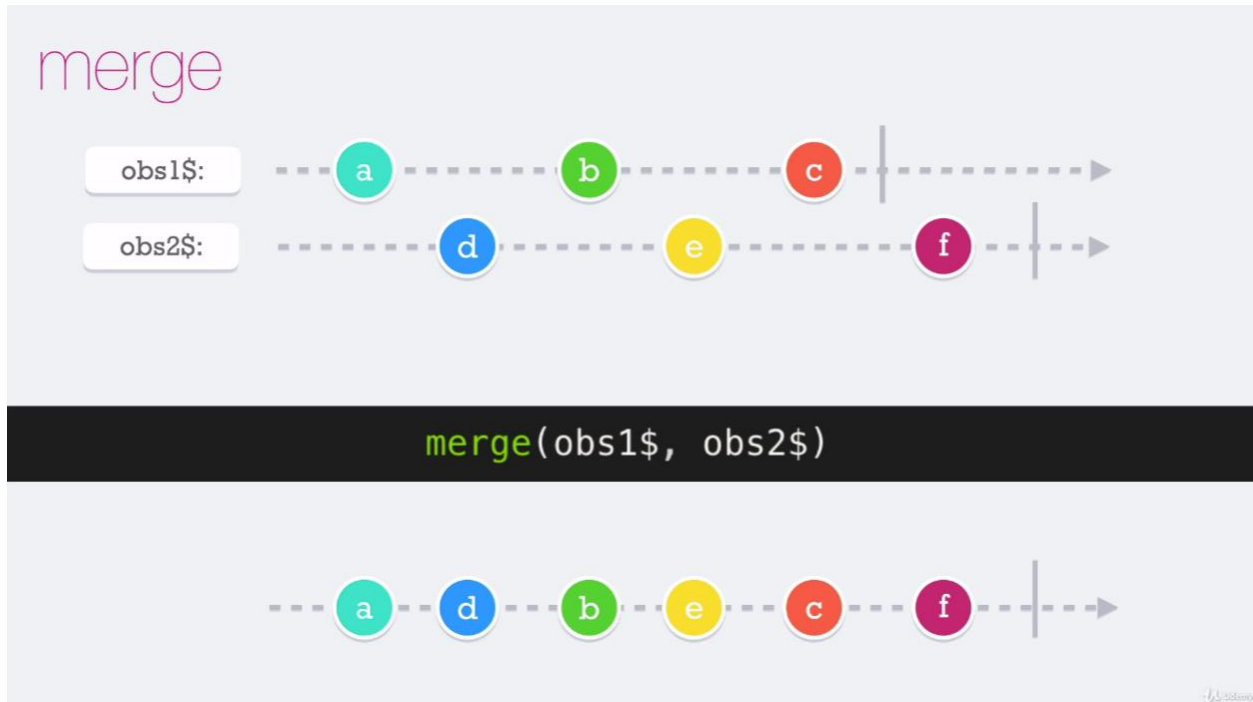
91. concat – Función

Esta es una función no es un operador ya que el operador está obsoleto. Básicamente lo que hace esta función que recibe observables como argumento, también puede recibir un iterable o bien un arreglo, pero usualmente se usa para trabajar con observables, con estos observables el concat va a crear un nuevo observable. Hay que tener en cuenta que hasta que no se complete el activo el siguiente no se van a ejecutar.



92. merge – Método

El método merge recibe uno o más observables y el resultado va a ser el producto de ambos observables combinados simultáneamente. Hay que tener en cuenta que hasta que todos los observables no se completan la suscripción no se va a completar. Adicionalmente el orden de la salida va a ser exactamente el primero que emita un valor.



93. combineLatest

Esta es una función que nos permite enviar observables como argumento, combinarlos y emitir los valores de todos los observables internamente simultáneamente, es importante recalcar que el `combineLatest` regresa un nuevo observable el cual va a emitir valores hasta que todos los observables internos hayan emitido por lo menos un valor, adicionalmente no se va a completar la suscripción hasta que todos los observables se completen.

combineLatest



```
combineLatest( obs1$, obs2$ )
```



94. forkJoin

El forkjoin también puede recibir varios observables como argumentos para trabajar los cuales van a ser retornados como un arreglo, pero con una configuración se pueden retornar como un objeto. Otra cosa que hay que tener en cuenta es que cada uno de los operadores que reciba el forkJoin tienen que ser finitos, de lo contrario este no emitirá ningún valor.

