El concepto de Hot vs Cold Observable esta cada día mas de moda con los temas de programación asíncrona. ¿Qué diferencia existe entre los observables fríos y los observables calientes? . La diferencia fundamental esta en como se comportan en tiempo de ejecución . Veamos un ejemplo sencillo utilizando unos arrays y unos botones en una página html. El primer paso es crear un bloque de código que contenga dos observables.

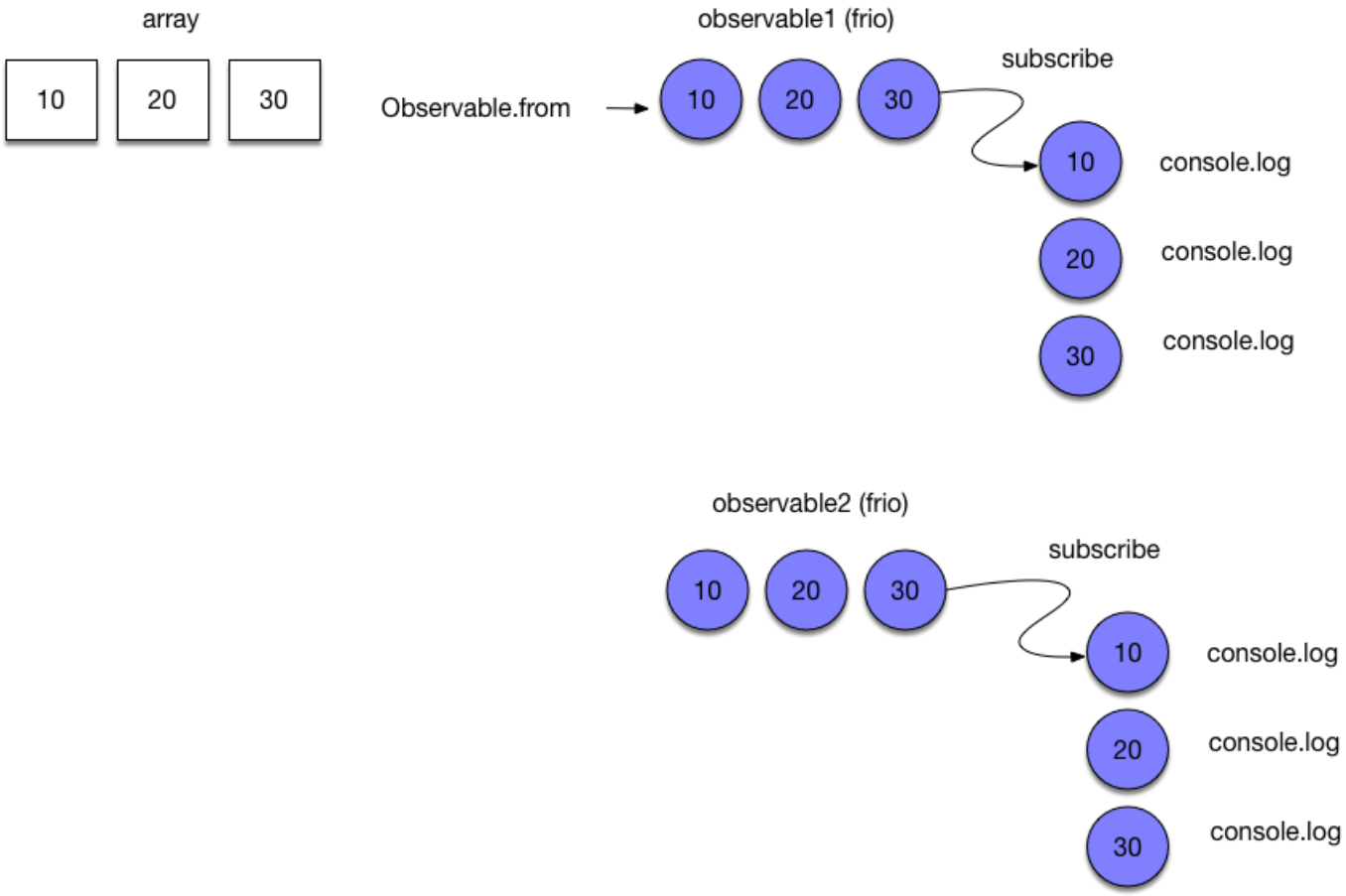
```
var array = [10, 20, 30];

var observable1 = Rx.Observable.from(array);
var observable2 = Rx.Observable.from(array);
observable1.subscribe(function(elemento) {

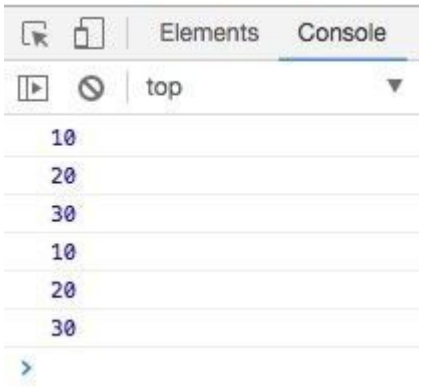
    console.log(elemento);
});
observable2.subscribe(function(elemento) {

    console.log(elemento);
});
```

En este caso hemos construido dos Observables a partir de un array . Ambos observables son de tipo Cold (fríos) ya que cada vez que los invoquemos nos devolverán todos los elementos del array uno a uno



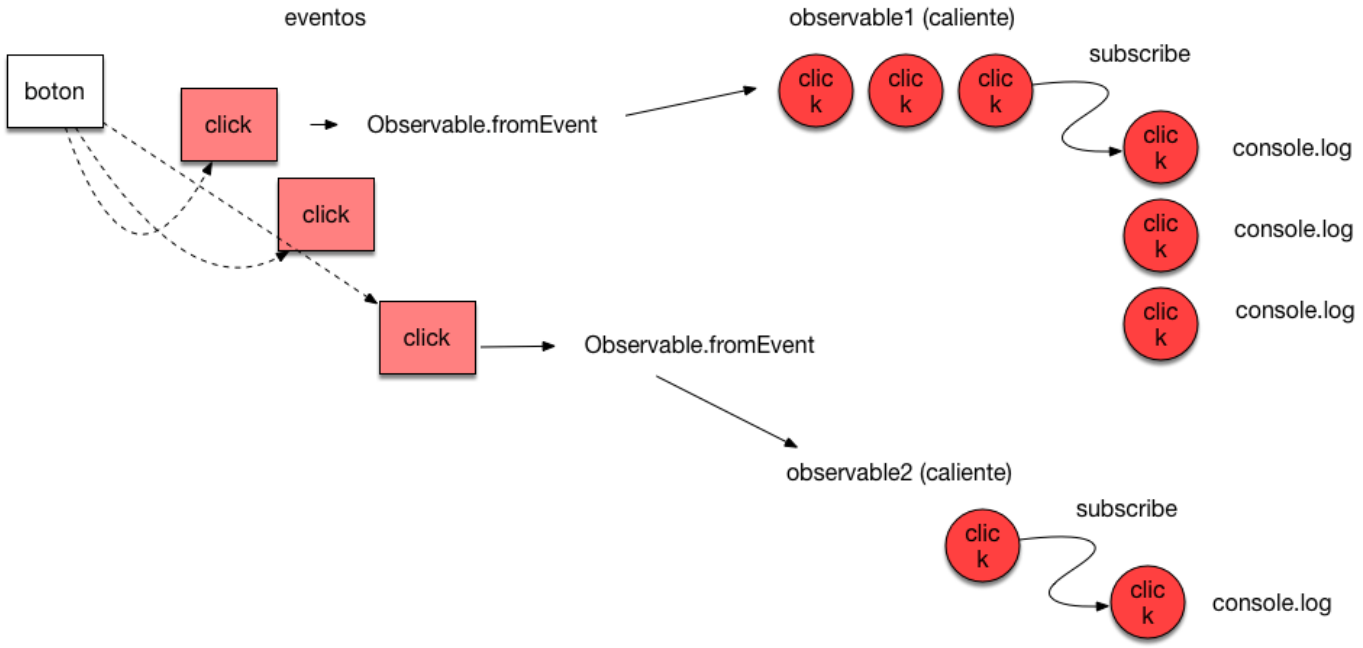
Si ejecutamos y lo mostramos por consola veremos algo así:



Hemos ejecutado los dos Observables y ambos devuelven la lista completa.

Hot vs Cold Observable

El caso de los Hot Observables es diferente se trata de observables a los cuales tu te subscribes en un momento determinado y NO nos devuelven todos los datos emitidos desde el principio sino que nos devuelven los datos a partir del momento en el que nos subscribimos. Esto sucede por ejemplo cuando pulsamos un botón y ligamos un observable a él.



Vamos a verlo en código:

```
<body onload="load()>
<input type="button" id="boton" value="pulsar">
<input type="button" value="aceptar" onClick="registrarObservable()">
</body>
```

El navegador mostrará:



En este caso el código es sencillo tenemos un botón normal y otro botón que registra a futuro un observable. Veamos el código de la función load().

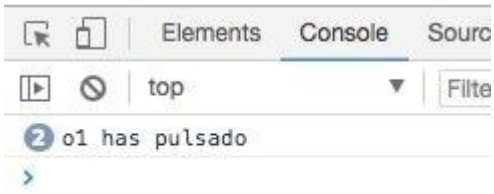
```
function load() {

var boton= document.getElementById("boton")
var observable1= Rx.Observable.fromEvent(boton, 'click')
observable1.subscribe(function(elemento) {

  console.log("o1 has pulsado");

})
}
```

Acabamos de crear un observable ligado al evento de click del botón de pulsar. Cada vez que pulsemos el observador emitirá un mensaje en la consola.



Es momento de pulsar el otro botón y comprobar que los observables relacionados con eventos son de tipo Hot ya que solo reciben los mensajes a partir de que comienza la subscripción. Para ello necesitaremos ver la función de registrarObservable().

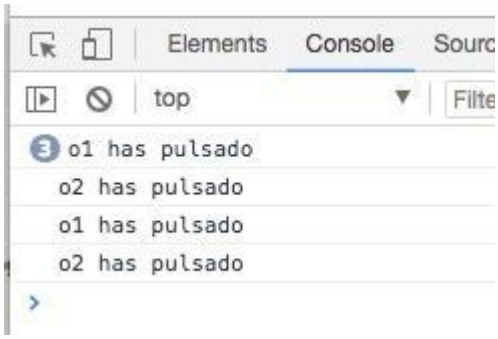
```
function registrarObservable() {

var boton= document.getElementById("boton")
var observable2= Rx.Observable.fromEvent(boton, 'click')
observable1.subscribe(function(elemento) {

  console.log("o2 has pulsado");

})
}
```

Esto nos registra un observable en un momento determinado. El cual solo recibirá los eventos que se produzcan después de su registro:



Hemos realizado un ejemplo de Hot vs Cold Observable utilizando Rx.js y eventos de click.

Otros artículos relacionados

1. Usando Rx Observables en JavaScript
2. RxJS Ajax , fusionando peticiones asíncronas
3. Introducción a RxJava y sus observables
4. Rxjs