**Appendix D. RxJS essentials**

*Synchronous programming* is relatively straightforward in that each line of your code is executed after the previous one. If you invoke a function in line 25 that returns a value, you can use the returned value as an argument for the function invoked in line 26.

*Asynchronous programming* dramatically increases code complexity. In line 37, you can invoke an asynchronous function that will return the value sometime later. Can you invoke a function in line 38 that uses the value returned by the previous function? The short answer is, “It depends.”

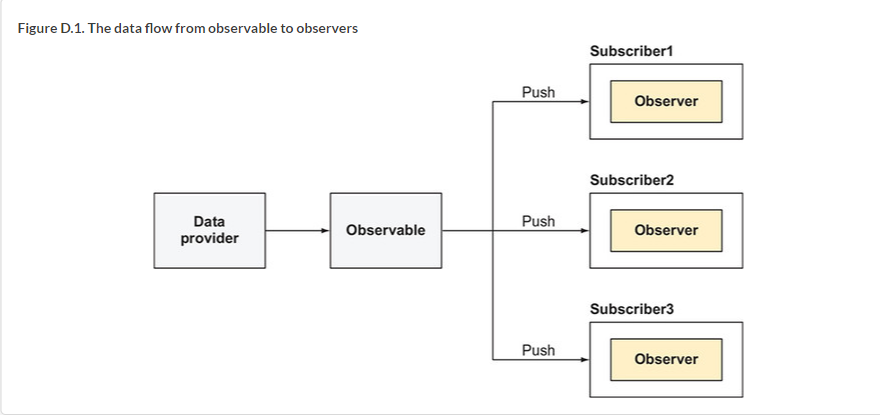
This appendix is an introduction to the RxJS 6 library, which can be used with any JavaScript-based app. It shines when it comes to writing and composing asynchronous code. Because Angular uses the RxJS library internally, we decided to add a primer to this book.

The first library of reactive extensions (Rx) was created by Erik Meijer in 2009. Rx.NET was meant to be used for apps written with Microsoft .Net technology. Then the Rx extensions were ported to multiple languages, and in the JavaScript world, RxJS 6 is the current version of this library.





### D.1. Getting familiar with RxJS terminology

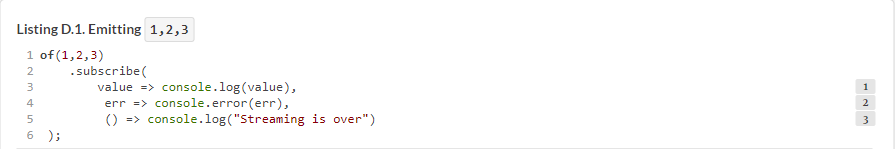


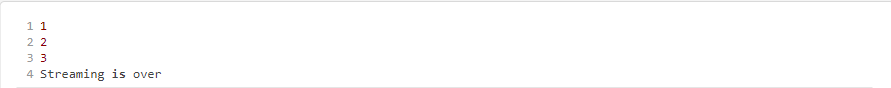
### D.2. Observable, observer, and subscriber





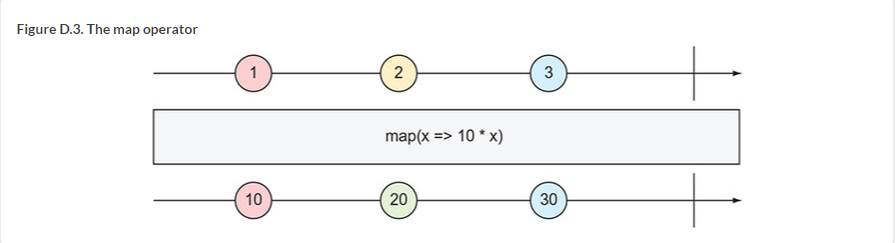
### D.3. Creating observables

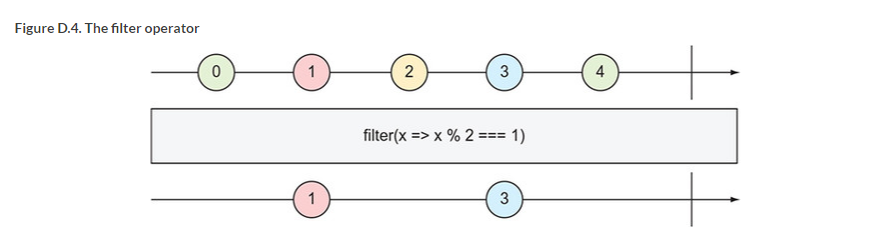


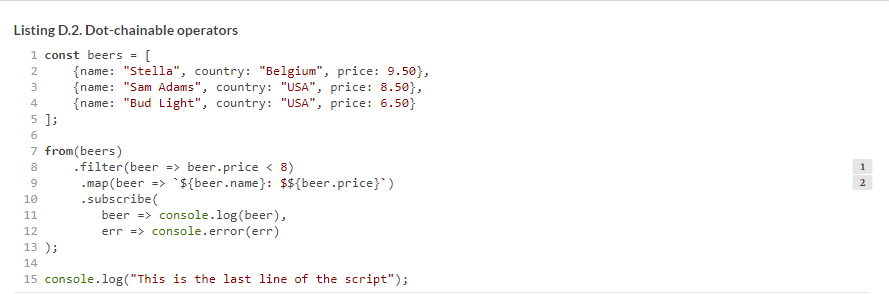


### D.4. Getting familiar with RxJS operators









#### D.4.1. Pipeable operators