

## 异步事件是以时间为索引的常量队列

[⌕ Back to series \(/series/rxswift-101\)](#)

学习reactive programming，如果你之前没有经验，那么最大的障碍，就应该让自己先感性的了解下这究竟是怎么回事儿。一旦你对此有所顿悟，甚至可以说，你已经成功一半儿了。

而当我们理解一个陌生事物的时候，最习惯的办法，就是用一个熟悉的事物去类比它。因此，在这一节里，我们就通过一个也许不太科学的办法，来感性认识下reactive programming。

从过滤一个常量数组说起

假设，我们有一个包含数字1-9的字符串数组：

```
let stringArray = ["1", "2", "3", "4", "5", "6", "7", "8", "9"]
```

现在，要过滤出 stringArray 中的所有偶数，得益于Swift对函数式编程的支持，我们可以这样：

```
let fullFilter = stringArray
    .flatMap {
        Int($0)
    }
    .filter {
        $0 % 2 == 0
    }
// [2, 4, 6, 8]
```

此时，fullFilter 的值就是 [2, 4, 6, 8] 了。接下来，我们不要过滤 stringArray 中所有的值了，只要第5个元素以后的偶数，就可以这样：

```
let partialFilter = stringArray[4 ..< stringArray.count]
    .flatMap {
        Int($0)
    }
    .filter {
        $0 % 2 == 0
    }
// [6, 8]
```

我们把这两次过滤放在一张图里对比一下：

- 第一次我们要过滤全体时，它的状态包含了"1" - "9"；
- 第二次过滤时，它的状态就变成了 5" - "9"，stringArray 的前4个元素对于我们的过滤过程来说，是完全不可见的；

有了这个“不可变队列”（stringArray）、“事件”（产生过滤需求）以及“实际执行的动作”（我们刚才编写的过滤代码）这三个概念之后。我们带着这三个概念，来看一个更实际的例子。这次，我们从用户的输入中，过滤出所有的偶数。

过滤用户输入的数字

在Xcode里新建一个single view application，例如：FilterNumber，在main.storyboard里，拖一个UITextField用于输入数字，然后，把这个UITextField的delegate设置成ViewController：

由于要在用户输入后，过滤偶数，我们给 ViewController 添加一个 extension，并实现下面这个方法：

🔍 字号

● 字号

🖌 默认主题

🖌 金色主题

🖌 暗色主题

```
extension ViewController: UITextFieldDelegate {
    public func textField(
        _ textField: UITextField,
        shouldChangeCharactersIn range: NSRange,
        replacementString string: String) -> Bool {
        // 1. Map input to Int
        if let n = Int(string) {
            // 2. Filter out the even number
            if n % 2 == 0 {
                print("Even: \(n)")
            }
        }

        return true
    }
}
```

这样，我们就可以在用户输入后，字符被显示出来之前得到通知。其中，`string` 参数，就是用户输入的内容，我们几乎是按照和之前同样的逻辑对 `string` 进行了处理，如果是偶数，就在控制台上把它打印出来。最后，统一返回 `true`，让这些输入都显示在 `UITextField` 里。

接下来，按 `Cmd + R` 执行，在 `UITextField` 中输入1-6，就能在控制台看到对应的结果了：

## 对比两个场景之后的启示

为什么要举这个例子呢？因为过滤用户输入这个动作，从某种意义上说，和我们之前过滤常量数组是有着诸多相似之处的：

首先，当我们输入完6之后，之前所有已经输入过的字符就都已经是过去发生的事情了，它们都是不可更改的了，因此，站在时间这个维度上来说，在过滤用户输入的偶数这种事件中，“不可变队列”就是截止到过滤行为结束时，我们输入过的所有字符；

其次，“事件”同样是我们过滤出“不可变队列”中的偶数；

第三，“实际执行的动作”就是写在

`textField(_:shouldChangeCharactersIn:replacementString:)` 这个方法里的代码；

如同我们看到的一样，虽然这两个应用场景在形式上类似，但我们却用了截然不同的实现方式。过滤数组的代码简单直观；过滤用户输入的代码就相对复杂，我们每处理一类事件，就要设置对应的 `delegate`，并引入对应的事件处理方法。以至于，我们一眼看上去刚才编写的 `extension`，都很难把它和过滤输入整数这样的动作联系起来，我们更多的直觉是：喔，这应该是在处理某种用户交互吧。

## What's next?

那么，既然形式上类似，我们为什么不能用过滤数组一样的方式，来过滤用户输入的偶数呢？为了达成这个目的，首先，我们得有一种表达“以时间为索引的常量队列”的方式。而这，就是我们切入 `RxSwift` (<https://github.com/ReactiveX/RxSwift>) 的方式。在下一节中，我们就先来了解下安装 `RxSwift` (<https://github.com/ReactiveX/RxSwift>) 的几种不同方式。

---

◀ 返回视频

(/series/rxswift-101)

安装RxSwift的三种不同方式 ▶

(<https://www.boxueio.com/series/rxswift-101/ebook/205>)

---



职场漂泊的你，每天多学一点。

从开发、测试到运维，让技术不再成为你成长的绊脚石。我们用打磨产品的精神去传播知识，把最新的移动开发技术，通过简单的图表，清晰的视频，简明的文字和切实可行的例子一一向你呈现。让学习不仅是一种需求，也是一种享受。

泊学动态

一个工作十年PM终创业的故事（二） (<https://www.boxueio.com/after-the-full-upgrade-to-swift3>)

Mar 4, 2017

人生中第一次创业的"10有" (<https://www.boxueio.com/founder-chat>)

Jan 9, 2016

猎云网采访报道泊学 (<http://www.lieyunwang.com/archives/144329>)

Dec 31, 2015

What most schools do not teach (<https://www.boxueio.com/what-most-schools-do-not-teach>)

Dec 21, 2015

一个工作十年PM终创业的故事（一） (<https://www.boxueio.com/founder-story>)

May 8, 2015

## 泊学相关

关于泊学

>

加入泊学

>

泊学用户隐私及服务条款 ([HTTPS://WWW.BOXUEIO.COM/TERMS-OF-SERVICE](https://www.boxueio.com/terms-of-service))

版权声明 ([HTTPS://WWW.BOXUEIO.COM/COPYRIGHT-STATEMENT](https://www.boxueio.com/copyright-statement))

## 联系泊学

Email: 10[AT]boxue.io (<mailto:10@boxue.io>)

QQ: 2085489246

2017 © Boxue, All Rights Reserved. 京ICP备15057653号-1 (<http://www.miibeian.gov.cn/>) 京公网安备 11010802020752号 (<http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010802020752>)

友情链接 [SwiftV \(http://www.swiftv.cn/\)](http://www.swiftv.cn/) | [Seay信息安全博客 \(http://www.cnseay.com/\)](http://www.cnseay.com/) | [Swift.gg \(http://swift.gg/\)](http://swift.gg/) | [Laravist \(http://laravist.com/\)](http://laravist.com/) | [SegmentFault \(https://segmentfault.com/\)](https://segmentfault.com/) | [骧青K的博客 \(http://blog.dianqk.org/\)](http://blog.dianqk.org/)