

☰ 它叫Optional, 却必不可少

◀ Optional map和flatMap的应用和实现

两个调试optional的小技巧 ▶

(https://www.boxueio.com/series/optional-is-not-an-option/ebook/144)

(https://www.boxueio.com/series/optional-is-not-an-option/ebook/146)

什么时候需要force unwrapping

☰ Back to series (/series/optional-is-not-an-option)

我们都知道，对于一个optional变量来说，可以用！来强行读取optional包含的值，Swift管它叫作force unwrapping。然而，这种操作并不安全，强制读取值为 nil 的optional会引发运行时错误。于是，每当我们默默在一个optional后面写上！的时候，心里总是会隐隐感到一丝纠结。我们到底什么时候该使用force unwrapping呢？

无论是在Apple的官方文档，还是在Stack overflow上的各种讨论中，你都能找到类似下面的言论：

1. 永远都不要使用这个东西，你会有更好的办法；
2. 当你确定optional一定不为 nil 时；
3. 当你确定你真的必须这样做时；
4. ...

然而，当你没有切身体会的时候，似乎很难理解这些言论的真实含义。其实，就在我上一节内容的最后，就已经遇到了一个非常具体的例子：

```
extension Sequence {
    func myFlatMap<T>(<_ transform:
        (Iterator.Element) -> T?) -> [T] {
        return self.map(transform)
            .filter { $0 != nil }
            .map { $0! } // Safely force unwrapping
    }
}
```

在我们用 filter { \$0 != nil } 过滤掉了 self 中，所有的非 nil 元素之后，在 map 里，我们要获得所有optional元素中包含的值，这时，对 \$0 使用force unwrapping，就满足了之前提到的两个条件：

1. 我们可以确定此时 \$0 一定不为 nil ；
2. 我们也确定真的必须如此；

现在，你对于“绝对安全”和“必须如此”这两个条件，应该有一个更具体的认识了。所以，但凡没有给你如此强烈安全感的场景，不要使用force unwrapping。

而对于第一种“永远都不要使用force unwrapping”的言论，其实也有它的道理，毕竟在我们之前对optional的各种应用方式里，你的确几乎看不到我们使用了force unwrapping。

甚至，即便当你身处在一个相当安全的环境里，的确相比force unwrapping，你会有更好的方法。例如，对下面这个表示视频信息的Dictionary来说：

```
let episodes = [
    "The fail of sentinal values": 100,
    "Common optional operation": 150,
    "Nested optionals": 180,
    "Map and flatMap": 220,
]
```

Key表示视频的标题，Value表示视频的秒数。

如果，我们要对视频时长大于100秒的视频标题排序，形成一个新的 Array，就可以这样：

```
episodes.keys
    .filter { episodes[$0!] > 100 }
    .sorted()
```

在 filter 中，我们筛选大于100秒时长的视频时，这里使用force unwrapping也是绝对安全的。因为 episode 是一个普通的 Dictionary，它一定不为 nil，因此，我们也一定可以使用 keys 读到它的所有键值，即便 episodes 不包含任何内容也没问题。然后，既然读到了键值，用force unwrapping读取它的value，自然也是安全的了。

⊕ 字号

● 字号

🖌 默认主题

🖌 金色主题

🖌 暗色主题

所以，这也算是一个可以使用force unwrapping的场景。但就像我们刚才说的那样，实际上，你仍有语义更好的表达方式，毕竟在 filter 内部再去访问 episodes 看上去并不那么美观。怎么做呢？

```
episodes.filter { (_, duration) in duration > 100 }  
    .map { (title, _) in title }  
    .sorted()
```

我们可以对整个 Dictionary 进行筛选，首先找到所有时长大于100的视频形成新的 Dictionary，然后，把所有的标题，map 成一个普通的 Array，最后，再对它排序。这样，我们就不用任何 force unwrapping 了，而且，就表意来说，要比之前的版本，容易理解的多。

What's next?

以上，就是我们关于force unwrapping的讨论。读到这里，相信你再也不会对这个行为感到纠结了。事实上，需要我们force unwrapping的场景的确非常少，绝大部分时候，我们都有更好的解决方案。然而，事实却并不如我们想象中的完美，当我们接手到使用了force unwrapping的代码，或不得已而为之的情况下，怎么办呢？在下一节中，我们就来介绍几种常用的调试方法，至少，它们可以为各种错误提供更有效的信息，或者，为运行时提供一个更安全的环境。

Optional map和flatMap的应用和实现

(<https://www.boxueio.com/series/optional-is-not-an-option/ebook/144>)

两个调试optional的小技巧

(<https://www.boxueio.com/series/optional-is-not-an-option/ebook/146>)



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