

☰ 它叫Optional, 却必不可少

◀ Chaining and Nil coalescing

Optional map和flatMap的应用和实现 ▶

(<https://www.boxueio.com/series/optional-is-not-an-option/ebook/142>)

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

为什么需要双层嵌套的Optional?

⌕ Back to series ([/series/optional-is-not-an-option](https://www.boxueio.com/series/optional-is-not-an-option))

☰ 字号

● 字号

🖌 默认主题

🖌 金色主题

🖌 暗色主题

如果一个optional封装的类型又是一个optional会怎样呢？在这一节中，我们就来了解这种情况的应用场景以及Swift针对optional双层嵌套进行的优化。

首先，假设我们有一个String类型的Array：

```
let stringOnes: [String] = ["1", "One"]
```

当我们要把stringOnes转变成一个Int数组的时候：

```
let intOnes = stringOnes.map { Int($0) }
```

此时，我们就会得到一个[Optional<Int>]，当我们遍历intOnes的时候，就可以看到这个结果：

```
intOnes.forEach { print($0) }  
// Optional<Int>  
// nil
```

至此，一切都没什么问题。但当你按照我们在之前提到过的while的方式遍历intOnes的时候，你就会发现，Swift悄悄对嵌套的optional进行了处理：

```
var i = intOnes.makeIterator()  
  
while let i = i.next() {  
    print(i)  
}  
// Optional<Int>  
// nil
```

虽然，这会得到和之前for循环同样的结果。但是仔细分析while的执行过程，你会发现，由于next()自身返回一个optional，而intOnes中元素的类型又是Optional<Int>，因此intOnes的迭代器指向的结果就是一个Optional<Optional<Int>>。

1. 当intOnes中的元素不为nil时，通过while let得到的结果，就是我们看到的经过一层unwrapping之后的Optional(1)；

2. 当intOnes中的元素为nil时，我们可以看到while let的到的结果并不是Optional(nil)，而直接是nil；

这说明Swift对嵌套在optional内部的nil进行了识别，当遇到这类情况时，可以直接把nil提取出来，表示结果为nil。

了解了这个特性之后，我们就可以使用for...in来正常遍历intOnes了。例如，使用我们之前提到的for case来读取所有的非nil值：

```
for case let one? in intOnes {  
    print(one) // 1  
}
```

或者统计所有的nil值：

```
for case nil in intOnes {  
    print("got a nil value")  
}
```

如果Swift不能对optional中嵌套的nil进行自动处理，上面的for循环是无法正常工作的。

What's next?

以上，就是双层嵌套optional的应用场景，以及Swift对这种类型进行的一些优化。除了Swift对optional类型的自动处理机制之外，我们也可以通过 `map` 和 `flatMap` 来手动处理optional类型的值，在下一节中，我们就来了解这两个函数的实现和用法。

◀ Chaining and Nil coalescing

(<https://www.boxueio.com/series/optional-is-not-an-option/ebook/142>)

Optional map和flatMap的应用和实现 ▶

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



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 >

泊学用户隐私以及服务条款([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](http://segmentfault.com/) (<http://segmentfault.com/>) | [靛青K的博客](http://blog.dianqk.org/) (<http://blog.dianqk.org/>)