What and Why in Swift 3.1

SE-0141 通过available约束Swift版本

(https://www.boxueio.com/series/what-is-new-in-swift-31/ebook/210)

SR-1446 关于内嵌类型的两种改进

(https://www.boxueio.com/series/what-is-new-in-swift-31/ebook/212)

# SR-1009 使用具象类型约束泛型参数

Back to series (/series/what-is-new-in-swift-31)

字号
字号
默认主题
金色主题
暗色主题

在Swift 3.0中，如果我们要为某个特定类型的optional添加 `extension` ，不是一件太容易的事情。例如，我们先给 `Int` 添加一个判断是否是偶数的 `extension`：

```
extension Int {
    var isEven: Bool {
        return self % 2 == 0
    }
}
```

然后，就可以像下面这样来使用 `isEven`：

```
3.isEven // false
```

但是，如果我们希望 `Int?` 也支持上面的功能，该怎么办呢？之前，我们只能这样，先定义一个读取整数值的 `protocol` ，并给 `Int` 添加一个默认实现，只是返回自身的值就好：

```
protocol IntValue {
    var value: Int { get }
}

extension Int: IntValue {
    var value: Int { return self }
}
```

然后，我们就可以通过对 `Optional` 封装的类型进行约束，达到只为 `Int?` 添加方法的目的了：

```
extension Optional where Wrapped: IntValue {
    var isEven: Bool {
        guard let value = self?.value else {
            return false
        }

        return value % 2 == 0
    }
}
```

这样，所有的 `Int?` 就也支持 `isEven` 操作了：

```
var foo: Int? = 3
foo.isEven // false
```

这虽然可以工作，但是有点儿啰嗦，实际上，我们是通过“所有提供了 `value` computed property”这样的约定来模拟了“只针对 `Int` 扩展 `Optional` ”这样的需求。

在Swift 3.1里，这个缺陷被弥补了，我们不仅可以对泛型参数使用 `protocol` 进行约束，还可以使用具象类型作为类型约束，这有点儿像C++泛型中的模板偏特化技术。我们可以直接把对 `Int?` 的扩展改成这样：

```
extension Optional where Wrapped == Int {
    var isEven: Bool {
        guard let value = self else {
            return false
        }

        return value % 2 == 0
    }
}
```

之前的 `foo.isEven` 仍旧可以正常工作，但 `Wrapped == Int` 看上去则简单直观多了。

SE-0141 通过available约束Swift版本
(https://www.boxueio.com/series/what-is-new-in-swift-31/ebook/210)

SR-1446 关于内嵌类型的两种改进
(https://www.boxueio.com/series/what-is-new-in-swift-31/ebook/212)

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)

版权声明 (HTTPS://WWW.BOXUEIO.COM/COPYRIGHT-STATEMENT)

## 联系泊学

Email: 10[AT]boxue.io (mailto:10@boxue.io)
QQ: 2085489246



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