Swift 3 Collections

和Set相关的基础知识 (https://www.boxueio.com/series/collection-types/ebook/132)

理解Range和Collection的关系 (https://www.boxueio.com/series/collection-types/ebook/134)

常用的Set方法

Back to series (/series/collection-types)

在理解了 Set 最基本的操作之后，这一节中，我们来看一些更实际的 Set 用法，它当然不仅仅是和 Dictionary 存储值的形式不同这么简单。其中第一个要提到的就是，作为表示一组值的无序集合，Set 支持各种常用的代数运算方法。

Set的代数运算

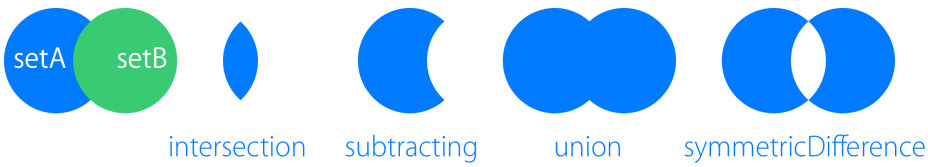
为了介绍各种运算方法，先定义两个 Set：

```
var setA: Set = [1, 2, 3, 4, 5, 6]
var setB: Set = [4, 5, 6, 7, 8, 9]
```

然后，我们就可以对 setA 和 setB 进行下面的运算：

```
// {5, 6, 4}
let interSectAB: Set = setA.intersection(setB)
// {9, 7, 2, 3, 1, 8}
let symmetricDiffAB: Set = setA.symmetricDifference(setB)
// {2, 4, 9, 5, 6, 7, 3, 1, 8}
let unionAB: Set = setA.union(setB)
// {2, 3, 1}
let aSubstractB: Set = setA.subtracting(setB)
```

而上面这些代码的含义，我们可以用下面这张图清楚的表现出来：



除此之外，上面这些API还有一个“可修改Set自身”的版本，而命名方式，就是在这些API的名称前面，加上form，例如：

```
setA.formIntersection(setB) // { 5, 6, 4 }
```

这样， setA 的值，就被修改成了取交集之后的值。关于这些API的用法，大家也可以参考SetAlgebra protocol (https://developer.apple.com/reference/swift/setalgebra)中的说明。

把Set用作内部支持类型

很多时候，除了把 Set 作为一个集合类型返回给用户之外，我们还可以把它作为函数的内部支持类型来使用。例如，借助于 Set 不能包含重复元素的特性，为任意一个序列类型去重。

我们给 Sequence 添加下面的扩展：

```
extension Sequence where Iterator.Element: Hashable {
    func unique() -> [Iterator.Element] {
        var result: Set<Iterator.Element> = []

        return filter {
            if result.contains($0) {
                return false
            }
            else {
                result.insert($0)
                return true
            }
        }
    }
}
```

在这个例子里，我们使用了 Set 不能包含重复元素的特性，用 result 保存了所有已经筛选的元素，如果遇到重复的元素，就跳过，否则，就把它添加到 result 里用于下一次筛选。这样，我们就可以直接使用 unique 来去重了：

```
[1, 1, 2, 2, 3, 3, 4, 4].unique() // [1, 2, 3, 4]
```

IndexSet和CharacterSet

在Swift标准库里， Set 是唯一一个支持 SetAlgebra protocol的类型。但是，在Foundation里，却还有两个额外的类型： IndexSet 和 CharacterSet 。

其中， IndexSet 和 Set<Int> 是非常类似的，例如：

```
let oneToSix: IndexSet = [1, 2, 3, 4, 5, 6]
```

但当我们要表达一连串正整数时，尤其是这个整数范围比较大的时候，使用 IndexSet 要比使用 Set<Int> 更经济一些，因为 Set<Int> 会保存这个范围里的每一个整数，而 IndexSet 则会使用类似 1...6 这样的形式保存一个范围。因此，要表达的范围越大，使用 IndexSet 就会越经济。并且，由于 IndexSet 也完全实现了 SetAlgebra 和 Collection 这两个protocol，因此，它的用法，和 Set 几乎是相同的。

另一个类 Set 类型，就是 CharacterSet ，它主要表示某一类字符的集合，通常，我们用这个类型来判断字符串中是否包含特定类型的字符，例如：

```
// String
let hw = "Hello world!"

// CharacterSet
let numbers = CharacterSet(charactersIn: "123456789")
let letters = CharacterSet.letters

// Actions
hw.rangeOfCharacter(from: numbers) // nil
hw.rangeOfCharacter(from: letters) //
```

在上面的代码中可以看到，我们即可以自定义一个字符集合，也可以使用标准库中预定义好的一些特定的集合。大家可以在CharacterSet官方文档(https://developer.apple.com/reference/foundation/characterset)中，找到完整的类型定义。

定义好集合之后，我们就可以使用 rangeOfCharacter(from:) 来判断 String 对象是否包含特定的字符了。如果包含， rangeOfCharacter 会返回一个 Range 对象，否则，就返回 nil 。

What's next?

以上，就是 Set 类型常用的一些操作和概念。至此，关于Swift标准库中和unordered collection相关的内容，就告一段落了。在下一节中，我们将讨论一个集合类型与range操作符关系的问题。

◀ 和Set相关的基础知识
(https://www.boxueio.com/series/collection-types/ebook/132)

理解Range和Collection的关系 ▶
(https://www.boxueio.com/series/collection-types/ebook/134)

字号
字号
默认主题
金色主题
暗色主题

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/)