

☰ Swift 3 的第一印象

[⏪ 使用Tuple打包数据](#)

[使用“Markdown方言”编写代码注释 ⏩](#)

<https://www.boxueio.com/series/swift-up-and-running/ebook/4>

<https://www.boxueio.com/series/swift-up-and-running/ebook/84>

快速过一遍常用操作符

操作符，是一个编程语言中完成各种运算不可或缺的元素。Swift中的大部分操作符，都符合我们对各种运算的理解。因此，在这篇文档里，我们将带着大家快速过一遍Swift中的各种操作符，并对其中大家可能不太熟悉的部分，做一些特别的说明。

☰ Back to series (/series/swift-up-and-running/)

基本操作符 (Basic operators)

字号

● 字号

🖌 默认主题

🖌 金色主题

🖌 暗色主题

赋值操作符

这个最简单，我们之前已经用过多次，等号右边的值赋值给等号左边的变量：

```
//: ##### Basic assignment
let a = 20
var b = 10
```

基本算术运算操作符

```
let sum = a + b
let sub = a - b
let mul = a * b
let div = a / b
let mod = a % b
```

“Swift 3不再允许浮点数取模。例如：8 % 2.5这样的写法在Swift 3中将会报错。如果要对浮点数取模，只能这样：8.truncatingRemainder(dividingBy: 2.5)。”

复合运算操作符

Swift还支持把赋值和算数运算符组合起来：

```
//: ##### Compound assignment

b += 10 // b = b + 10
b -= 10 // b = b - 10
b *= 10 // b = b * 10
b /= 10 // b = b / 10
b %= 10 // b = b % 10
```

“Swift不会把数字自动转换成 Bool 类型。在需要 Bool 值的地方，你必须明确使用一个 Bool 变量。”

“Swift 3中不再支持自增（++）和自减（--）操作符，使用它们的前缀和后缀版本都会得到一个编译器错误。因此，需要+1/-1的时候，只能使用 b += 1 和 b -= 1 来实现。”

比较操作符

Swift支持以下常用的比较操作：

```
//: ##### Comparison
let isEqual      = sum == 10
let isNotEqual  = sum != 10
let isGreater   = sum > 10
let isLess      = sum < 10
let isGe        = sum >= 10
let isLe        = sum <= 10
```

除此之外，Swift还支持两个用于比较对象引用的操作符：Identity operator，它们用来判断两个操作数是否引用同一个对象，我们在后面讲到面向对象编程的时候，会进一步提到这两个操作符。

```
//: Identity operator

//===
//!=
```

三元操作符

```
/*
 * if condition {
 *     expression1
 * }
 * else {
 *     expression2
 * }
 *
 */

let isSumEqualToTen = isEqual ? "Yes" : "No"
```

在Swift里，一些“日常”的if...else...判断，可以使用下面的三元操作符来替代：cond ? expr1 : expr2。

Nil Coalescing Operator

这是一个Swift特有的操作符，用来处理和Optional有关的判断：

```
// opt != nil ? opt! : b

var userInput: String? = "A user input"
let value = userInput ?? "A default input"
```

如果opt是一个optional，当其不为nil时，就使用optional变量自身的值，否则，就使用??后面的“默认值”。

Range operator

闭区间range operator

我们用下面的方式表达一个包含begin和end的闭区间：begin ... end：

```
//: Closed range operator
// begin...end

for index in 1...5 {
    print(index)
}
```

半开半闭区间range operator

我们用下面的方式表达一个[begin, end)的半开半闭区间：begin ..< end：

```
//: Half-open range operator
// begin..
```

逻辑运算符

Swift支持三种常用的逻辑运算：NOT，AND和OR。它们都返回一个Bool：

```
//: #### Logic operator

let logicalNot = !isEqual
let logicalAnd = isEqual && isLess
let logicalOR  = isGreater || (isLess && isLe)
```

What's next?

了解了常用的操作符之后，下一节，我们来看如何在Swift注释中，使用类似markdown格式的注释。Xcode以及Playground中都可以很好的识别这些注释，从而帮助开发者更好的理解和使用你的代码。

◀ 使用Tuple打包数据

(<https://www.boxueio.com/series/swift-up-and-running/ebook/4>)

使用“Markdown方言”编写代码注释 ▶

(<https://www.boxueio.com/series/swift-up-and-running/ebook/84>)



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@boxue.io (<mailto:10@boxue.io>)

QQ: 2085489246