为代码的执行做个决定

使用简单的样式匹配

(https://www.boxueio.com/series/make-a-decision/ebook/135)

理解样式匹配的实现方式

(https://www.boxueio.com/series/make-a-decision/ebook/137)

# 使用高级样式匹配方式

Back to series (/series/make-a-decision)

字号

字号

默认主题

金色主题

暗色主题

这一节，我们继续了解在分支和循环中，更高级的样式匹配方式。

## 使用where约束条件

除了使用具体的数值对循环或分支条件进行约束外，我们还可以使用 where 进行更复杂的约束。先来看一个简单的例子：

```
for i in 1...10 where i % 2 == 0 {
    print(i)
}
```

这里，我们在 for 循环中使用了 where 限定了进入循环的值必须是偶数，这要比我们在 for 循环内部再写一个 if 判断方便的多。

或者，我们也可以把 where 用在更复杂的value binding语句里。例如，假设我们定义下面这样的 enum 表示手机电量：

```
enum Power {
    case fullyCharged
    case normal(percentage: Double)
    case outOfPower
}
```

然后，定义一个变量 battery 表示手机电池：

```
let battery = Power.normal(percentage: 0.1)
```

这样，我们就可以在绑定 .normal associated value的同时，使用 where 进一步约束它的关联值了：

```
switch battery {
case .normal(let percentage) where percentage <= 0.1:
    print("Almost out of power")
case .normal(let percentage) where percentage >= 0.8:
    print("Almost fully charged")
default:
    print("Normal battery status")
}
```

## 使用逗号串联条件

在之前我们提到过， switch 中的多个 case 默认是不向下贯通的，因此，我们**不能**像C语言一样，通过罗列多个 case 来实现对多个条件统一处理的效果：

```
switch halfPower {
// !!! CANNOT fall through multiple cases !!!
case .fullyCharged:
case .outOfPower:
    print("Fully charged or out of power")
// ...
}
```

要实现这样的效果，我们只能在一条 case 里，用逗号分隔多个情况：

```
switch halfPower {
// ...
case .fullyCharged, .outOfPower
    print("Fully charged or out of power")
// ...
}
```

逗号除了用在 switch...case... 中表示逻辑或的概念，也可以用在 if 中表示逻辑与的概念，例如为了处理之前电量低的情况，我们还可以用 if 这样来实现：

```
if case .normal(let percentage) = battery,
    case 0...0.1 = percentage {
    print("Almost out of power")
}
```

在上面的代码里，第一个 if case 使用value binding读取了 battery 中 .normal 的associated value。接下来，第二个 case 进一步约束了第一个 case 中关联到的值小于10%的情况。

因此，当我们需要对多个条件递进执行判断的时候，不要写下面这样的 if 嵌套：

```
if A {
    if B {
        if C {

        }
    }
}
```

你完全可以使用逗号把这些条件逐步写在一个 if 里：

```
if A, B, C {

}
```

此时，Swift就会执行 if A, then B, then C 这样的语义了。特别是，当ABC之间还有计算关联的时候，这样就会显得额外方便。

## 使用tuple简化多个条件的比较

有时，我们需要在 if 中同时比较多个条件。假设，我们有一对用户名和密码：

```
let username = "11@boxue.io"
let password = 11111111
```

当我们要同时比较这两个值时，最普通的写法，就是在 if 中使用逻辑与（&&）操作符：

```
if username == "11@boxue.io" && password == 11111111 {
    print("correct")
}
```

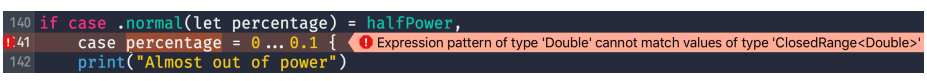
如果你不喜欢在 if 中并列多个比较语句，我们还可以用 case 临时生成一个tuple，并且，让它和 username / password 进行比较：

```
if case ("11@boxue.io", 11111111) = (username, password) {
    print("correct")
}
```

这样，当同时要比较的元素比较多时，这种写法用起来就方便很多。

## What's next?

如果你仔细观察之前我们判断是否电量低的 if 语句就会发现， case 中要比较的值的范围是写在等号左侧的： case 0...0.1 = percentage ，甚至，当你把这个顺序缓过来的时候，编译器会报错：




这是为什么呢？为了理解这个问题，在下一节中，我们就来了解Swift中样式匹配背后的实现规则。

(https://www.boxueio.com/series/make-a-decision/ebook/135) (https://www.boxueio.com/series/make-a-decision/ebook/137)

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/)