

☰ Swift 3 的第一印象

◀ 基于unicode的字符串常用操作

快速过一遍常用操作符 ▶

(<https://www.boxueio.com/series/swift-up-and-running/ebook/122>)

(<https://www.boxueio.com/series/swift-up-and-running/ebook/5>)

使用Tuple打包数据

[⌕ Back to series \(/series/swift-up-and-running\)](#)

很多时候，我们需要把多个不同类型的值，打包成一个单位处理。例如，返回一个HTTP状态信息：

- 状态码: 200; 状态消息: HTTP OK
- 状态码: 404; 状态消息: File not found

或者，返回一条数据库用户信息记录：姓名: Mars；工号: 11；电子邮件: 11@boxue.io 。在Swift里，我们可以使用Tuple 来很方便的处理类似的问题。

🔍 字号

🔍 字号

🖌️ 默认主题

🖌️ 金色主题

🖌️ 暗色主题

定义一个Tuple

我们使用下面的方式，来定义一个Tuple：

```
(value1, value2, value3...)
```

例如，定义我们开始提到的HTTP状态码：

```
//: ##### Define a tuple

let success = (200, "HTTP OK")
let fileNotFound = (404, "File not found")
```

我们还可以给Tuple中的每一个数据成员指定一个名字，例如，定义一个表达用户信息记录的Tuple：

```
(name1: value1, name2: value2, name3: value3...)

//: ##### Define a tuple
let me = (name: "Mars", no: 11, email: "11@boxue.io")
```

访问Tuple中的数据成员

定义好Tuple之后，我们可以使用下面的方式访问Tuple中的数据成员：

```
//: ##### Access tuple content

success.0
success.1

fileNotFound.0
fileNotFound.1
```

如果我们在定义Tuple时，指定了Tuple成员的名字，我们就可以像下面这样访问这些数据成员：

```
//: ##### Access tuple content

me.name
me.no
me.email
```

Tuple Decomposition

我们在定义Tuple的时候，还可以把一个Tuple的值，一一对应的拆分到不同的变量上，这叫做Tuple Decomposition。例如，对于之前定义过的 success ，我们可以这样定义一个新的Tuple：

```
var (successCode, successMessage) = success

print(successCode) // 200
print(successMessage) // HTTP OK
```

之后，就可以直接访问 `successCode` 和 `successMessage` 的值了。这可以提高我们处理Tuple成员时的代码可读性。但要说明的是，我们这里是使用 `success` 的值，构建了一个新的Tuple，因此修改 `successCode` 或 `successMessage` 的值，不会影响到原来的 `success`。

例如，我们修改 `successCode`：

```
successCode = 201

success // (200, "HTTP OK")
```

从结果我们可以看到，之前的 `success` 的值没有被修改。另外，如果我们只是想对应到Tuple中特定的成员，而忽略其它成员，我们可以使用下划线 `_` 来代表那些不需要被对应的成员。例如：

```
let (_, errorMessage) = fileNotFound
print(errorMessage)
```

Tuple type

每一个Tuple的类型，都是由Tuple中所有数据成员一起决定给的。例如，对于一开始我们定义的 `success` 和 `me`，它们的类型就分别是：`(Int, String)` 和 `(String, Int, String)`。当我们需要用type annotation定义一个Tuple的时候，我们可以这样写：

```
var redirect: (Int, String) = (302, "temporary redirect")
```

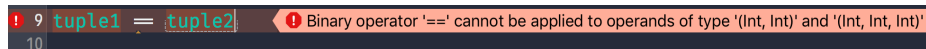
Tuple comparison

当我们比较两个Tuple类型的变量时，要遵循下面的规则：

首先，只有元素个数相同的Tuple变量之间，才能进行比较。例如，下面的代码会引发编译错误：

```
let tuple12 = (1, 2)
let tuple123 = (1, 2, 3)

tuple2 < tuple3
```



```
! 9 tuple1 == tuple2 // Binary operator '==' cannot be applied to operands of type '(Int, Int)' and '(Int, Int, Int)'
```

从上面的结果就能看到，包含两个 `Int` 的Tuple不能和包含三个 `Int` 的Tuple进行比较。

其次，当Tuple中元素个数相同时，比较是按照Tuple中元素的位置从前向后依次进行的：

- 如果Tuple中，相同位置的两个元素相等，则继续比较下一个位置的两个元素，并根据第一个同一位置不相等的两个元素的大小关系，确定两个Tuple变量的关系；
- 如果两个Tuple中所有位置的元素都相等，则两个Tuple变量相等；

因此，对于下面这个例子，`tuple11 < tuple12` 的结果是 `true`：

```
let tuple11 = (1, 1)
let tuple12 = (1, 2)

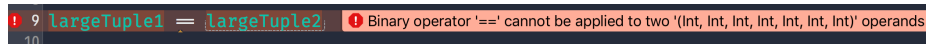
tuple11 < tuple12 // true
```

但是，有一点要说明的是，我们只可以对最多包含6个元素的Tuple变量进行比较，超过这个数量，Swift会报错。例如对于下面这段代码：

```
let largeTuple1 = (1, 2, 3, 4, 5, 6, 7)
let largeTuple2 = (1, 2, 3, 4, 5, 6, 7)

largeTuple1 == largeTuple2 // Error !!!
```

编译器就会提示类似下面这样的错误：



```
! 9 largeTuple1 == largeTuple2 // Binary operator '==' cannot be applied to two '(Int, Int, Int, Int, Int, Int, Int)' operands
```

What's next?

当我们要对数据简单进行打包处理的时候，Tuple是个非常好用的工具。接下来，我们会快速过一遍Swift中的各种操作符。它们之中的绝大部分，都是几乎每个编程语言都具备的功能。如果你了解过其他编程语言，掌握它们自然不在话下。

基于unicode的字符串常用操作

(<https://www.boxueio.com/series/swift-up-and-running/ebook/122>)

快速过一遍常用操作符

(<https://www.boxueio.com/series/swift-up-and-running/ebook/5>)



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>) | 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/>)