Swift 3 的第一印象

整数和浮点数

(https://www.boxueio.com/series/swift-up-and-running/ebook/2)

String和NSString处理Unicode上的差异

(https://www.boxueio.com/series/swift-up-and-running/ebook/107)

忘记旧有的"C风格"字符串吧

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

字号

字号

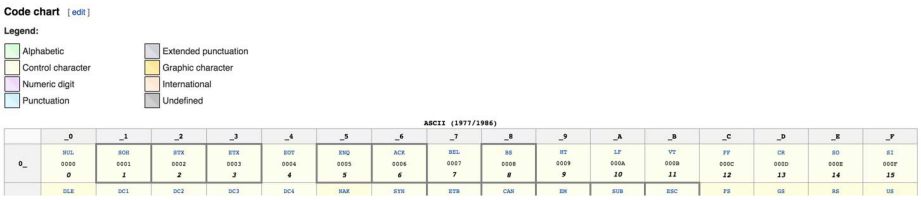
默认主题

金色主题

暗色主题

如果你是个从C语言那个年代一路走来的程序员，无论你在用什么语言编程，*String就是个长度固定的字符数组*这样的想法应该会在你的脑海里根深蒂固。你可能会想，它们终究会是那个样子的：“一个字符串是由若干个字符组成的字符数组，而每个字符则对应着一个8位的ASCII数值。”

而实际上，如果你的产品只针对美国人，这样的想法可能才不会给你带来太大的麻烦。没错，只能是美国人，哪怕是其他英语国家的都不行。因为，这种“定长”的字符数组表示的字符串，只能显示ASCII码表中的128个字符。



当你想为英国的用户显示一个英镑符号（£）的时候，这种“定长数组”都无法满足你的需求。因此，国际标准化组织就希望通过扩展ASCII码表的范围来解决这个问题。

ISO/IEC 8859

如何来扩展呢？最初的方案是仍旧基于8-bit ASCII编码的：

- 一方面，启用了字符编码中的第8个bit，使用了160-255这个范围的数值，并把这个范围的编码定义为 `ISO/IEC 8859` 。而之前的ASCII编码只使用了前7个bit；
- 另一方面，根据不同国家和地区语言符号的特征，对这些扩展的编码表达的意思进行了归类。在Wikipedia (https://en.wikipedia.org/wiki/ISO/IEC_8859)上可以看到，一共定义了16个大类，作为 `ISO/IEC 8859` 的不同部分，被分别定义成 `ISO/IEC 8859-n` ；

我们节选了其中一部分：

从图中可以看到，对于每一个部分，同样的编码值，表示的字符是不同的。

Fix lenght unicode - 1.0

对于这样的方案，很快人们就发现并不能有效的解决问题。从图中可以看到，尽管Part 6 (https://en.wikipedia.org/wiki/ISO/IEC_8859-6)包含了常用的阿拉伯语字符，但是仍旧不支持诸如伊朗等使用阿拉伯语系的文字。何况，在这份列表里，还没有对中文、日文等字符的支持。

于是，为了用一种一致的方式处理不同语系的字符，人们开发了一种叫做Unicode的编码方式。第一个版本的Unicode采用了16-bit等宽字符的编码方案，叫做UCS-2 (http://justsolve.archiveteam.org/wiki/UCS-2)。但很快，这种方案就被证明也不足以有效的表示所有字符。

那么，如果用等宽32-bit表示一个字符呢？所有人都觉得，在绝大多数情况下，这都太不经济了。

Variable length unicode

于是，现如今的unicode采用的是可变长度编码方案。而所谓的“可变长度”包含了两个意思：

- “编码单位（code unit）”的长度是可变的；
- 构成同一个字符的“编码单位”组合也是可变的；

什么是code unit呢？简单来说，code unit和ASCII码的形式是非常类似的，它们是一个个具体的数值。不同的是，它可以由多种长度单位的数字构成：

第一种是用多个连续的8-bit数字表示一个unicode，这就是我们熟知的UTF-8 (https://en.wikipedia.org/wiki/UTF-8)编码。这种编码方式可以很好的对ASCII编码实现兼容。例如，人民币符号 ¥ 的UTF-8编码是： `C2 A5` ；

第二种是用一个16-bit数字表示一个unicode，这种编码方式叫做UTF-16 (https://en.wikipedia.org/wiki/UTF-16)。例如， ¥ 的UTF-16编码是： `00A5` ；

最后，当然就是UTF-32 (https://en.wikipedia.org/wiki/UTF-32)编码， ¥ 的UTF-32编码是： `000000A5` ，你应该不难理解它的含义；

Unicode scalar?

当然，无论我们用什么编码来表示字符，和ASCII编码一样，每一个可变长unicode字符最终也会对应一个数字来表示它的编码值，这个值叫做code point。现如今，这个值的的范围是[0, 0x10FFFF]，而我们只用了还不到12%的空间。因此，还有大量的空间允许我们添加诸如emoji这样的符号。

理解Surrogate pair

看到这里，你可能会想，如果我们使用UTF-16编码，根本无法全部表示上面定义的code point啊。为了解决这个问题，unicode标准保留了UTF-16编码空间中的一些值，它们永远不会被定义成字符，而是和其它UTF-16的编码值组合在一起，表示一个unicode字符。

来看一个例子：

💖Sparkling heart的code point是1F496，显然，他不能用一个UTF-16来表示，因此，整个编码过程是这样的：

1. 用code point减去0x10000，得到0x0F496；
2. 把这个数字变成二进制0000 1111 0100 1001 0110；
3. 取10-MSB：0000 1111 01和10-LSB：00 1001 0110；
4. 用0xD800 + 10-MSB得到0xD83D；
5. 用0xDC00 + 10-LSB得到0xDC96；
6. 这样 D83D DC96 就是这个unicode的UTF-16编码；

而我们用到的0xD800和0xDC00，就叫做surrogate pair。

什么是unicode scalar?

理解了surrogate pair之后，就不难理解unicode scalar了。简单来说，它就是除了surrogate pair之外的code unit。在后面的视频中，我们可以看到，Swift中，我们可以使用 `\u{1F496}` 这样的方式，来表示一个unicode scalar。

忘了String是字符数组这个事情吧

很长时间以来，在其他编程语言里，为了降低复杂性，人们总是试图隐藏一个事实：

"*一个在屏幕上显示的字符可能由多个code unit组合而成。*"

但这却给开发者理解unicode，甚至在处理unicode的代码上留下了很多难以发现和处理的bug。于是，在Swift里， `String` 并没有这样做，开发团队对这个类型最重要的一个设计目标就是尽可能保持这个类型在Unicode上的语义正确。

当然，这样做也是有代价的。在Swift里， `String` 已经彻底不再是一个集合类型。而是一个提供了从多个维度展现一个Unicode视图的类型。你可以得到它的多个 `Characters` ，可以看到它的UTF-8 / UTF-16 / Unicode scalar值等等。

所以，彻底忘了 `String` 是一个字符数组这样的事情吧。哪怕是从概念上，也不要这样去理解Swift中的 `String` 。让自己用一个全新的方式，去理解现如今我们需要处理的字符。

What's next?

以上就是这一节的内容，我们向大家介绍了关于字符编码和Unicode的一些基础概念，了解它们，是在Swift中正确使用 `String` 类型的关键。接下来，我们将和大家讨论和Unicode字符串相等判断有关的话题。

◀ 整数和浮点数

(https://www.boxueio.com/series/swift-up-and-running/ebook/2)

String和NSString处理Unicode上的差异 ▶

(https://www.boxueio.com/series/swift-up-and-running/ebook/107)

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/)