

Swift 字符串

Swift 字符串是一系列字符的集合。例如 "Hello, World!" 这样的有序的字符类型的值的集合，它的数据类型为 **String**。

创建字符串

你可以通过使用字符串字面量或 String 类的实例来创建一个字符串：

```
import Cocoa

// 使用字符串字面量
var stringA = "Hello, World!"
print( stringA )

// String 实例化
var stringB = String("Hello, World!")
print( stringB )
```

以上程序执行输出结果为：

```
Hello, World!
Hello, World!
```

空字符串

你可以使用空的字符串字面量赋值给变量或初始化一个String类的实例来初始值一个空的字符串。 我们可以使用字符串属性 is Empty 来判断字符串是否为空：

```
import Cocoa

// 使用字符串字面量创建空字符串
var stringA = ""

if stringA.isEmpty {
    print( "stringA 是空的" )
} else {
    print( "stringA 不是空的" )
}

// 实例化 String 类来创建空字符串
let stringB = String()

if stringB.isEmpty {
```

```
    print( "stringB 是空的" )
} else {
    print( "stringB 不是空的" )
}
```

以上程序执行输出结果为：

```
stringA 是空的
stringB 是空的
```

字符串常量

你可以将一个字符串赋值给一个变量或常量，变量是可修改的，常量是不可修改的。

```
import Cocoa

// stringA 可被修改
var stringA = "菜鸟教程："
stringA += "http://www.runoob.com"
print( stringA )

// stringB 不能修改
let stringB = String("菜鸟教程：")
stringB += "http://www.runoob.com"
print( stringB )
```

以上程序执行输出结果会报错，因为 stringB 为常量是不能被修改的：

```
error: left side of mutating operator isn't mutable: 'stringB' is a 'let' constant
stringB += "http://www.runoob.com"
```

字符串中插入值

字符串插值是一种构建新字符串的方式，可以在其中包含常量、变量、字面量和表达式。您插入的字符串字面量的每一项都在以反斜线为前缀的圆括号中：

```
import Cocoa

var varA    = 20
let constA  = 100
var varC:Float = 20.0

var stringA = "\(varA) 乘以 \(constA) 等于 \ (varC * 100)"
print( stringA )
```

以上程序执行输出结果为：

```
20 乘以 100 等于 2000.0
```

字符串连接

字符串可以通过 + 号来连接，实例如下：

```
import Cocoa

let constA = "菜鸟教程："
let constB = "http://www.runoob.com"

var stringA = constA + constB

print( stringA )
```

以上程序执行输出结果为：

```
菜鸟教程：http://www.runoob.com
```

字符串长度

字符串长度使用 **String.count** 属性来计算，实例如下：

Swift 3 版本使用的是 String.characters.count

```
import Cocoa

var varA = "www.runoob.com"

print( "\(varA)，长度为 \(varA.count)" )
```

以上程序执行输出结果为：

```
www.runoob.com，长度为 14
```

字符串比较

你可以使用 == 来比较两个字符串是否相等：

```
import Cocoa

var varA    = "Hello, Swift!"
var varB    = "Hello, World!"

if varA == varB {
    print( "\(varA) 与 \(varB) 是相等的" )
} else {
    print( "\(varA) 与 \(varB) 是不相等的" )
}
```

以上程序执行输出结果为：

```
Hello, Swift! 与 Hello, World! 是不相等的
```

Unicode 字符串

Unicode 是一个国际标准，用于文本的编码，Swift 的 String 类型是基于 Unicode 建立的。你可以循环迭代出字符串中 UTF-8 与 UTF-16 的编码，实例如下：

```
import Cocoa

var unicodeString = "菜鸟教程"

print("UTF-8 编码: ")
for code in unicodeString.utf8 {
    print("\(code) ")
}

print("\n")

print("UTF-16 编码: ")
for code in unicodeString.utf16 {
    print("\(code) ")
}
```

以上程序执行输出结果为：

```
UTF-8 编码:
232
143
156
233
184
159
```

230
149
153
231
168
139
UTF-16 编码:
33756
40479
25945
31243

字符串函数及运算符

Swift 支持以下几种字符串函数及运算符：

序号	函数/运算符 & 描述
1	isEmpty 判断字符串是否为空，返回布尔值
2	hasPrefix(prefix: String) 检查字符串是否拥有特定前缀
3	hasSuffix(suffix: String) 检查字符串是否拥有特定后缀。
4	Int(String) 转换字符串数字为整型。 实例: <div><pre>let myString: String = "256" let myInt: Int? = Int(myString)</pre></div>
5	String.count Swift 3 版本使用的是 String.characters.count 计算字符串的长度
6	utf8 您可以通过遍历 String 的 utf8 属性来访问它的 UTF-8 编码
7	utf16 您可以通过遍历 String 的 utf8 属性来访问它的 utf16 编码
8	unicodeScalars

	您可以通过遍历String值的unicodeScalars属性来访问它的 Unicode 标量编码.
9	+ 连接两个字符串，并返回一个新的字符串
10	+= 连接操作符两边的字符串并将新字符串赋值给左边的操作符变量
11	== 判断两个字符串是否相等
12	< 比较两个字符串，对两个字符串的字母逐一比较。
13	!= 比较两个字符串 是否不相等。

← Swift Fallthrough 语句

Swift 字符(Character) →



1 篇笔记

 写笔记



字符串分割数组 -- 基于空格

```
let fullName = "First Last"
let fullNameArr = fullName.characters.split{$0 == " "}.map(String.init)
// or simply:
// let fullNameArr = fullName.characters.split{" "}.map(String.init)

fullNameArr[0] // First
fullNameArr[1] // Last
```

jinxm 1年前 (2017-11-30)