

[← Swift 字符串](#)[Swift 数组 →](#)

Swift 字符(Character)

Swift 的字符是一个单一的字符字符串字面量，数据类型为 Character。

以下实例列出了两个字符实例：

```
import Cocoa

let char1: Character = "A"
let char2: Character = "B"

print("char1 的值为 \(char1)")
print("char2 的值为 \(char2)")
```

以上程序执行输出结果为：

```
char1 的值为 A
char2 的值为 B
```

如果你想在 Character（字符）类型的常量中存储更多的字符，则程序执行会报错，如下所示：

```
import Cocoa

// Swift 中以下赋值会报错
let char: Character = "AB"

print("Value of char \(char)")
```

以上程序执行输出结果为：

```
error: cannot convert value of type 'String' to specified type 'Character'
let char: Character = "AB"
```

空字符变量

Swift 中不能创建空的 Character（字符）类型变量或常量：

```
import Cocoa

// Swift 中以下赋值会报错
let char1: Character = ""
var char2: Character = ""
```

```
print("char1 的值为 \(char1)")
print("char2 的值为 \(char2)")
```

以上程序执行输出结果为：

```
error: cannot convert value of type 'String' to specified type 'Character'
let char1: Character = ""
           ^~
error: cannot convert value of type 'String' to specified type 'Character'
var char2: Character = ""
```

遍历字符串中的字符

Swift 的 String 类型表示特定序列的 Character（字符）类型值的集合。每一个字符值代表一个 Unicode 字符。

您可通过for-in循环来遍历字符串中的characters属性来获取每一个字符的值：

```
import Cocoa

for ch in "Runoob".characters {
    print(ch)
}
```

以上程序执行输出结果为：

```
R
u
n
o
o
b
```

字符串连接字符

以下实例演示了使用 String 的 append() 方法来实现字符串连接字符：

```
import Cocoa

var varA:String = "Hello "
let varB:Character = "G"

varA.append( varB )

print("varC = \(varA)")
```

以上程序执行输出结果为：

```
varC = Hello G
```

[← Swift 字符串](#)

[Swift 数组 →](#)

[✎ 点我分享笔记](#)