

Go 语言递归函数

递归，就是在运行的过程中调用自己。

语法格式如下：

```
func recursion() {  
    recursion() /* 函数调用自身 */  
}  
  
func main() {  
    recursion()  
}
```

Go 语言支持递归。但我们在使用递归时，开发者需要设置退出条件，否则递归将陷入无限循环中。

递归函数对于解决数学上的问题是非常有用的，就像计算阶乘，生成斐波那契数列等。

阶乘

以下实例通过 Go 语言的递归函数实例阶乘：

```
package main  
  
import "fmt"  
  
func Factorial(n uint64)(result uint64) {  
    if (n > 0) {  
        result = n * Factorial(n-1)  
        return result  
    }  
    return 1  
}  
  
func main() {  
    var i int = 15  
    fmt.Printf("%d 的阶乘是 %d\n", i, Factorial(uint64(i)))  
}
```

以上实例执行输出结果为：

```
15 的阶乘是 1307674368000
```

斐波那契数列

以下实例通过 Go 语言的递归函数实现斐波那契数列：

```
package main

import "fmt"

func fibonacci(n int) int {
    if n < 2 {
        return n
    }
    return fibonacci(n-2) + fibonacci(n-1)
}

func main() {
    var i int
    for i = 0; i < 10; i++ {
        fmt.Printf("%d\t", fibonacci(i))
    }
}
```

以上实例执行输出结果为：

```
0    1    1    2    3    5    8    13    21    34
```

[← Go 语言Map\(集合\)](#)

[Go 语言类型转换 →](#)



3 篇笔记

 写笔记