

Go 语言递归函数

递归，就是在运行的过程中调用自己。一个函数调用自己，就叫做递归函数。

构成递归需具备的条件：

- 1.子问题需与原始问题为同样的事，且更为简单。
- 2.不能无限制地调用本身，须有个出口，化简为非递归状况处理。没有退出条件，递归将成为死循环。

数字阶乘

一个正整数的阶乘（factorial）是所有小于及等于该数的正整数的积，并且0的阶乘为1。自然数n的阶乘写作n!。

```
package main

import "fmt"

func factorial(i int) int {
    if i <= 1 {
        return 1
    }
    return i * factorial(i-1)
}

func main() {
    var i int = 7 //7! 7*6*5*4*3*2*1
    fmt.Printf("%d 的阶乘结果是 %d\n", i, factorial(i))
}

#结果
7 的阶乘结果是 5040
```

斐波那契数列(Fibonacci)

这个数列从第3项开始，每一项都等于前两项之和。递归的方法定义： $F(0)=0$, $F(1)=1$, $F(n)=F(n-1)+F(n-2)$ ($n \geq 2$, $n \in \mathbb{N}^*$)。

```
package main

import "fmt"

func fibonacci(i int) int {
    if i == 0 {
        return 0
    }
    if i == 1 {
        return 1
    }
    return fibonacci(i-1) + fibonacci(i-2)
}

func main() {
```

```
var i int
for i = 0; i < 10; i++ {
    fmt.Printf("%d\n", fibonaci(i))
}
```

#结果

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