

Go 语言函数

函数是基本的代码块，用于执行一个任务。

Go 语言最少有个 main() 函数。

你可以通过函数来划分不同功能，逻辑上每个函数执行的是指定的任务。

函数声明告诉了编译器函数的名称，返回类型，和参数。

Go 语言标准库提供了多种可动用的内置的函数。例如，len() 函数可以接受不同类型参数并返回该类型的长度。如果我们传入的是字符串则返回字符串的长度，如果传入的是数组，则返回数组中包含的元素个数。

函数定义

Go 语言函数定义格式如下：

```
func function_name( [parameter list] ) [return_types] {  
    函数体  
}
```

函数定义解析：

- func：函数由 func 开始声明
- function_name：函数名称，函数名和参数列表一起构成了函数签名。
- parameter list：参数列表，参数就像一个占位符，当函数被调用时，你可以将值传递给参数，这个值被称为实际参数。参数列表指定的是参数类型、顺序、及参数个数。参数是可选的，也就是说函数也可以不包含参数。
- return_types：返回类型，函数返回一系列值。return_types 是该列值的数据类型。有些功能不需要返回值，这种情况下 return_types 不是必须的。
- 函数体：函数定义的代码集合。

实例

以下实例为 max() 函数的代码，该函数传入两个整型参数 num1 和 num2，并返回这两个参数的最大值：

实例

```
/* 函数返回两个数的最大值 */  
func max(num1, num2 int) int {  
    /* 声明局部变量 */  
    var result int  
  
    if (num1 > num2) {  
        result = num1  
    } else {  
        result = num2  
    }  
}
```

```
    return result
}
```

函数调用

当创建函数时，你定义了函数需要做什么，通过调用该函数来执行指定任务。

调用函数，向函数传递参数，并返回值，例如：

实例

```
package main

import "fmt"

func main() {
    /* 定义局部变量 */
    var a int = 100
    var b int = 200
    var ret int

    /* 调用函数并返回最大值 */
    ret = max(a, b)

    fmt.Printf( "最大值是 : %d\n", ret )
}

/* 函数返回两个数的最大值 */
func max(num1, num2 int) int {
    /* 定义局部变量 */
    var result int

    if (num1 > num2) {
        result = num1
    } else {
        result = num2
    }
    return result
}
```

以上实例在 main() 函数中调用 max () 函数，执行结果为：

```
最大值是 : 200
```

函数返回多个值

Go 函数可以返回多个值，例如：

实例

```
package main

import "fmt"
```

```
func swap(x, y string) (string, string) {  
    return y, x  
}  
  
func main() {  
    a, b := swap("Mahesh", "Kumar")  
    fmt.Println(a, b)  
}
```

以上实例执行结果为：

```
Kumar Mahesh
```

函数参数

函数如果使用参数，该变量可称为函数的形参。

形参就像定义在函数体内的局部变量。


调用函数，可以通过两种方式来传递参数：


传递类型	描述
值传递	值传递是指在调用函数时将实际参数复制一份传递到函数中，这样在函数中如果对参数进行修改，将不会影响到实际参数。
引用传递	引用传递是指在调用函数时将实际参数的地址传递到函数中，那么在函数中对参数所进行的修改，将影响到实际参数。

默认情况下，Go 语言使用的是值传递，即在调用过程中不会影响到实际参数。

函数用法

函数用法	描述
函数作为值	函数定义后可作为值来使用
闭包	闭包是匿名函数，可在动态编程中使用
方法	方法就是一个包含了接受者的函数

 3 篇笔记

 写笔记

