

# 函数详解1

作者：少林之巔

# 目录

1.函数介绍

2. 多返回值和可变参数

3. defer语句

4. 内置函数介绍

# 函数介绍

1. 定义：有输入、有输出，用来执行一个指定任务的代码块。

```
func functionname([parametername type]) [returntype] {  
    //function body  
}
```

//其中参数列表和返回值列表是可选

# 函数介绍

## 2. 无参数和返回值的函数

```
func functionname() {  
    //function body  
}
```

# 函数介绍

## 3. 练习，实现两个数相加

```
func add(a int, b int) int {  
    Return a + b  
}
```

# 函数介绍

4. 如何连续的一系列参数的类型是一样，前面的类型可以不写，例如：

```
func add(a, b int) int {  
    Return a + b  
}
```

# 函数介绍

## 5. 函数调用

```
func add(a, b int) int {  
    Return a + b  
}  
  
func main() {  
    sum := add(2, 3)  
}
```

# 多返回值和可变参数

## 1. 多返回值

```
func calc(a, b int) (int, int) {  
    sum := a + b  
    sub := a - b  
    return sum, sub  
}  
  
func main() {  
    sum, sub := add(2, 3)  
}
```



# 多返回值和可变参数

## 2. 对返回值进行命名

```
func calc(a, b int) (sum int, sub int) {  
    sum = a + b  
    sub = a - b  
    return  
}  
  
func main() {  
    sum, sub := add(2, 3)  
}
```

# 多返回值和可变参数

## 3. 对返回值进行命名

```
func calc(a, b int) (sum int, sub int) {  
    sum = a + b  
    sub = a - b  
    return  
}  
  
func main() {  
    sum, sub := add(2, 3)  
}
```

# 多返回值和可变参数

## 4. \_标识符

```
func calc(a, b int) (sum int, sub int) {  
    sum = a + b  
    sub = a - b  
    return  
}  
  
func main() {  
    sum, _ := add(2, 3)  
}
```

# 多返回值和可变参数

## 5. 可变参数

```
func calc_v1(b ...int) (sum int, sub int) {  
    return  
}  
  
func calc_v2(a int, b ...int) (sum int, sub int) {  
    return  
}  
  
func calc_v3(a int, b int, c ...int) (sum int, sub int) {  
    return  
}
```

# defer语句

## 1. defer

```
func calc_v1(b ...int) (sum int, sub int) {  
    defer fmt.Println("defer")  
    return  
}
```

# defer语句

2. 多个defer语句，遵循栈的特性：先进后出

```
func calc_v1(b ...int) (sum int, sub int) {  
    defer fmt.Println("defer1")  
    defer fmt.Println("defer2")  
    return  
}
```

# 内置函数

1. close: 主要用来关闭channel
2. len: 用来求长度, 比如string、array、slice、map、channel
3. new: 用来分配内存, 主要用来分配值类型, 比如int、struct。返回的是指针
4. make: 用来分配内存, 主要用来分配引用类型, 比如chan、map、slice
5. append: 用来追加元素到数组、slice中
6. panic和recover: 用来做错误处理

# 练习

1. 求1到100之内的所有质数，并打印到屏幕上
2. 打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如：153是一个“水仙花数”，因为 $153=1^3+5^3+3^3$ 。求100到1000之间的所有水仙花数，
3. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。