



# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

## panic函数

# 目录



PART 01  
**panic函数**



# panic函数

1.panic函数

## panic函数

---

error返回的是一般性的错误，但是**panic函数返回的是让程序崩溃的错误。**

也就是**当遇到不可恢复的错误状态的时候**，如数组访问越界、空指针引用等，这些运行时错误会引起panic异常。

在一般情况下，**我们不应通过调用panic函数来报告普通的错误，而应该只把它作为报告致命错误的一种方式。**当某些不应该发生的场景发生时，我们就应该调用panic。

一般而言，**当panic异常发生时，程序会中断运行，随后程序崩溃并输出日志信息。**日志信息包括panic value和函数调用的堆栈跟踪信息。

当然，如果**直接调用内置的panic函数也会引发panic异常。**

panic函数接受任何值作为参数。



在实际的开发过程中并不会直接调用panic()函数，但是当我们编程的程序遇到致命错误时，系统会自动调用该函数来终止整个程序的运行，也就是系统内置了panic函数。

数组下标越界的问题示例：

```
func TestA() {  
    fmt.Println("func TestA()")  
}  
  
func TestB(x int) {  
    var a [10]int  
    a[x] = 222 //x值为11时，数组越界  
}  
  
func TestC() {  
    fmt.Println("func TestC()")  
}  
  
func main() {  
    TestA()  
    TestB(11) //TestB() 发生异常，中断程序  
    TestC()  
}
```

错误信息如下： (panic异常，导致了整个程序崩溃)

```
func TestA()  
panic: runtime error: index out of range  
  
goroutine 1 [running]:  
main.TestB(...)  
    C:/Users/superman/Desktop/code/go/src/main.go:14  
main.main()  
    C:/Users/superman/Desktop/code/go/src/main.go:23 +0x4c  
错误：进程退出代码 2.
```



## 实例演示





江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看