课程安排，请关注微信公众平台或者官方微博

编程语言： Golang 与 html5

编程工具： Goland 和 HBuilder

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔业。

大家好，我是彬哥,
欢迎大家来到 Golang 语言社区云课堂课程的学习。
社区论坛网址：**www.golang.ltd**
技术交流群 ： 221273219
微信公众号 ：**Golang 语言社区**
微信服务号 ：**Golang 技术社区**

第一季 Go 语言基础、进阶、提高课程

第十九节 Go 语言 错误处理

Go 语言通过内置的错误接口提供了非常简单的错误处理机制。

error 类型是一个接口类型，这是它的定义：

```
type error interface {
    Error() string
}
```

我们可以在编码中通过实现 error 接口类型来生成错误信息。

函数通常在最后的返回值中返回错误信息。使用 errors.New 可返回一个错误信息：

```
func Sqrt(f float64) (float64, error) {
    if f < 0 {
        return 0, errors.New("math: square root of negative number")
    }
    // 实现
}
```

在下面的例子中，我们在调用 Sqrt 的时候传递的一个负数，然后就得到了 non-nil 的 error 对象，将此对象与 nil 比较，结果为 true，所以 fmt.Println(fmt 包在处理 error 时会调用 Error 方法)被调用，以输出错误，请看下面调用的示例代码：

```
result, err:= Sqrt(-1)

if err != nil {
   fmt.Println(err)
}
```

实例

```
package main

import (
     "fmt"
)

// 定义一个 DivideError 结构
type DivideError struct {
     dividee int
     divider int
}

// 实现 `error` 接口
func (de *DivideError) Error() string {
     strFormat := `
    Cannot proceed, the divider is zero.
    dividee: %d
    divider: 0
`
     return fmt.Sprintf(strFormat, de.dividee)
}

// 定义 `int` 类型除法运算的函数
func Divide(varDividee int, varDivider int) (result int, errorMsg str
ing) {
     if varDivider == 0 {
          dData := DivideError{
               dividee: varDividee,
               divider: varDivider,
          }
```

```
        errorMsg = dData.Error()
        return
    } else {
        return varDividee / varDivider, ""
    }

}

func main() {

    // 正常情况
    if result, errorMsg := Divide(100, 10); errorMsg == "" {
        fmt.Println("100/10 = ", result)
    }
    // 当被除数为零的时候会返回错误信息
    if _, errorMsg := Divide(100, 0); errorMsg != "" {
        fmt.Println("errorMsg is: ", errorMsg)
    }

}
```

执行以上程序，输出结果为：

```
100/10 =  10
errorMsg is:
    Cannot proceed, the divider is zero.
    dividee: 100
    divider: 0
```