

Go 语言从入门到项目实战

第4章 流程控制语法

4.1 分支结构if语句

- 由关键字If开头的语句用来判断某个条件是否成立。

- 格式：

```
if condition {  
    //条件成立时要执行的语句  
}else{  
    //条件不成立时要执行的语句  
}
```

4.1 分支结构if语句

- 示例：

```
var numC=500
if numC<0{
    fmt.Println("numC的值为负")
}else if numC==0{
    fmt.Println("numC的值为0")
}else if numC<100{
    fmt.Println("numC的值小于100")
}else if numC<200{
    fmt.Println("numC的值小于200")
}else if numC<300{
    fmt.Println("numC的值小于300")
}else if numC<400{
    fmt.Println("numC的值小于400")
}else if numC<500{
    fmt.Println("numC的值小于500")
}else{
    fmt.Println("numC的值大于或等于1000")
}
```


4.2 Switch...case...分支结构

- 格式：

```
switch variable {  
    case condition1:  
        //condition1成立时要执行的语句  
    case condition2:  
        //condition2成立时要执行的语句  
    default:  
        //无匹配条件是默认执行的语句  
}
```

4.2 Switch...case...分支结构

- 格式：

```
switch {  
    case condition1:  
        //condition1成立时要执行的语句  
    case condition2:  
        //condition2成立时要执行的语句  
    default:  
        //无匹配条件是默认执行的语句  
}
```


4.2 Switch...case...分支结构

- 示例：

```
var str string="hello , 三酷猫"
switch {
case str=="hello , 三酷猫":
    fmt.Println("英语打招呼",str)
    fallthrough
case str!="你好 , 三酷猫":
    fmt.Println("汉语打招呼 你好 , 三酷猫")
default:
    fmt.Println("没有看到三酷猫")
}
```

4.3 循环结构for语句

- 格式：

```
for init; condition; post {  
    //循环体代码块  
}
```


4.3 循环结构for语句

- 示例：

```
sum:= 0
```

```
for i:= 0; i<=10; i++ {
```

```
    sum+=i
```

```
}
```

```
fmt.Println(sum)
```


4.3 循环结构for语句

- 格式：

```
for condition{  
    //循环体代码块  
}
```

4.3 循环结构for语句

- 示例：

```
sum:= 0
```

```
i:=0
```

```
for i<=10{
```

```
    sum+=i
```

```
    i++
```

```
}
```

```
fmt.Println(sum)
```


4.4 流程控制语句

- break语句：

```
var sum int = 0
var i int
for i=1;i<=100;i++){
    sum+=i
    if i==10{
        break
    }
}
fmt.Println(sum)
```

4.4 流程控制语句

- continue语句：

```
var i int
for i=0;i<=5;i++){
    if i>3&& i<5{
        continue
    }
    fmt.Println("i=",i)
}
```


4.4 流程控制语句

- goto语句：

```
func main() {  
    var i int  
    var j int  
    for i = 0; i <= 5; i++ {  
        for j = 0; j <= 5; j++ {  
            if i == 1 {  
                goto stopLoop  
            }  
            fmt.Println("i=", i, ",j=", j)  
        }  
    }  
    stopLoop:  
    fmt.Println("i=1 , 结束循环")  
}
```

4.5 for-range语句

- 格式：

```
for key,value:=range variable{  
    //遍历时要执行的代码块  
}
```


4.5 for-range语句

- 示例：

```
fruit := map[string]float64{
    "apple": 5.4,
    "banana": 6.8,
    "cherry":10.5,
}
for key, value := range fruit {
    fmt.Println(key, value)
}
```