

流程控制语句

if-else

for循环结构（没有其他循环）

for range循环

switch-case语句

if-else

▼ 示例

Go

```
1 func main() {  
2     age := 19  
3     if age > 18 {  
4         fmt.Println("澳门首家线上赌场开业")  
5     } else {  
6         fmt.Println("滚")  
7     }  
8 }
```

for循环结构（没有其他循环）

```
1 for 初始语句 条件表达式 结束语句  
2     循环体语句  
3
```

条件表达式返回 `true` 时循环体不停地进行循环，直到条件表达式返回 `false` 时自动退出循环。

```
1 func forDemo() {  
2     for i := 0; i < 10; i++ {  
3         fmt.Println(i)  
4     }  
5 }
```

```
1 func main() {
2     //基本格式
3     for i := 0; i < 10; i++ {
4         fmt.Println(i)
5     }
6     fmt.Println("\n")
7     //变种1
8     var a = 5
9     for ; a < 10; a++ {
10        fmt.Println(a)
11    }
12    fmt.Println("\n")
13    //变种2
14    var n int = 5
15    for n < 10 {
16        fmt.Println(n)
17        n++
18    }
19 }
```

for range循环

Go语言中可以使用 `for range` 遍历数组、切片、字符串、`map` 及通道 (`channel`)。通过 `for range` 遍历的返回值有以下规律：

1. 数组、切片、字符串返回索引和值。
2. `map` 返回键和值。
3. 通道 (`channel`) 只返回通道内的值。

```
1 s := "Hello World"
2 for i, v := range s {
3     fmt.Println(i, string(v))
4 }
```

打印九九乘法表

```
1 for i := 1; i <= 9; i++ {
2     for j := 1; j <= i; j++ {
3         fmt.Printf("%d*%d=%d\t", i, j, j*i)
4     }
5     fmt.Printf("\n")
6 }
```

switch-case语句

```
1  var finger int
2  fmt.Scanf("%d", &finger)
3  switch finger {
4      case 1:
5          fmt.Println("大拇指")
6      case 2:
7          fmt.Println("食指")
8      case 3:
9          fmt.Println("中指")
10     case 4:
11         fmt.Println("无名指")
12     case 5:
13         fmt.Println("小拇指")
14     default:
15         fmt.Println("无效数字")
16 }
```