Golang 中的运算符

主讲教师：（大地）
合作网站：www.itying.com （IT 营）
我的专栏：https://www.itying.com/category-79-b0.html

1、Golang 内置的运算符

1. 算术运算符
2. 关系运算符
3. 逻辑运算符
4. 位运算符
5. 赋值运算符

2、算数运算符

运算符	描述
+	相加
-	相减
*	相乘
/	相除
%	求余=被除数-（被除数/除数）*除数

注意： ++（自增）和--（自减）在 Go 语言中是单独的语句，并不是运算符。

```
package main
import (
    "fmt"
)

func main() {

    fmt.Println("10+3=", 10+3) // =13
    fmt.Println("10-3=", 10-3) // =7
    fmt.Println("10*3=", 10*3) // =30

    //除法注意：如果运算的数都是整数，那么除后，去掉小数部分，保留整数部分
    fmt.Println("10/3=", 10/3)       //3
    fmt.Println("10.0/3=", 10.0/3) //3.3333333333333335

    // 取余注意  余数=被除数-（被除数/除数）*除数
    fmt.Println("10%3=", 10%3)       // =1
    fmt.Println("-10%3=", -10%3)    // -1
    fmt.Println("10%-3=", 10%-3)    // =1
    fmt.Println("-10%-3=", -10%-3) // =-1

}
```

注意：在 golang 中，++ 和 -- 只能独立使用 **错误写法如下：**

```
var i int = 8
var a int
a = i++ //错误，i++只能独立使用
a = i-- //错误，i--只能独立使用
```

注意：在 golang 中没有前++ **错误写法如下：**

```
var i int = 1
++i // 错误，在 golang 没有 前++
--i // 错误，在 golang 没有 前--
fmt.Println("i=", i)
```

++ --正确写法：

```
var i int = 1
i++
fmt.Println("i=", i)
```

3、关系运算符

运算符	描述
==	检查两个值是否相等，如果相等返回 True 否则返回 False。
!=	检查两个值是否不相等，如果不相等返回 True 否则返回 False。
>	检查左边值是否大于右边值，如果是返回 True 否则返回 False。
>=	检查左边值是否大于等于右边值，如果是返回 True 否则返回 False。
<	检查左边值是否小于右边值，如果是返回 True 否则返回 False。
<=	检查左边值是否小于等于右边值，如果是返回 True 否则返回 False。

```
package main
import (
    "fmt"
)
func main() {
    //演示关系运算符的使用
    var n1 int = 9
    var n2 int = 8
    fmt.Println(n1 == n2) //false
    fmt.Println(n1 != n2) //true
    fmt.Println(n1 > n2)  //true
    fmt.Println(n1 >= n2) //true
    fmt.Println(n1 < n2)  //flase
    fmt.Println(n1 <= n2) //flase
    flag := n1 > n2
    fmt.Println("flag=", flag)

}
```

4、逻辑运算符

运算符	描述
&&	逻辑 AND 运算符。 如果两边的操作数都是 True，则为 True，否则为 False。
||	逻辑 OR 运算符。 如果两边的操作数有一个 True，则为 True，否则为 False。
!	逻辑 NOT 运算符。 如果条件为 True，则为 False，否则为 True。

```
package main
import (
    "fmt"
)
func main() {
    //演示逻辑运算符的使用  &&
    var age int = 40
    if age > 30 && age < 50 {
        fmt.Println("ok1")
    }
    if age > 30 && age < 40 {
        fmt.Println("ok2")
    }
    //演示逻辑运算符的使用  ||

    if age > 30 || age < 50 {
        fmt.Println("ok3")
    }
    if age > 30 || age < 40 {
        fmt.Println("ok4")
    }

    //演示逻辑运算符的使用  !
    if age > 30 {
        fmt.Println("ok5")
    }
    if !(age > 30) {
        fmt.Println("ok6")
    }
}
```

逻辑运算符短路演示

```
package main
import (
    "fmt"
)

func test() bool {
    fmt.Println("test...")
    return true
}

func main() {
    var i int = 10
    if i < 9 && test() {
        fmt.Println("ok...")
    }
    if i > 9 || test() {
        fmt.Println("hello...")
    }
}
```

5、赋值运算符

运算符	描述
=	简单的赋值运算符，将一个表达式的值赋给一个左值
+=	相加后再赋值
-=	相减后再赋值
*=	相乘后再赋值
/=	相除后再赋值
%=	求余后再赋值

```
d := 8 + 2*8 // 赋值运算从右向左
fmt.Println(d)

x := 10
x += 5 //x=x+5
fmt.Println("x += 5 的值:", x)
```

```
x := 10
x -= 5 //x=x-5
fmt.Println("x -= 5 的值:", x)

x := 10
x *= 5 //x=x*5
fmt.Println("x *= 5 的值:", x)

x := 10.0
x /= 5
fmt.Println("x /= 5 的值:", x)

x := 10
x %= 3
fmt.Println("x %= 3 的值:", x)
```

6、运算符练习

练习 1：有两个变量，a 和 b，要求将其进行交换，最终打印结果

```
a := 9
b := 2

t := a
a = b //
b = t //
fmt.Printf("交换后的情况是 a = %v , b=%v \n", a, b)
```

练习 2：假如还有 100 天放假，问：xx 个星期零 xx 天

```
var days int = 100
var week int = days / 7
var day int = days % 7
```

```
fmt.Printf("%d 个星期零%d 天\n", week, day)
```

练习 3：定义一个变量保存华氏温度，华氏温度转换摄氏温度的公式为：5/9*(华氏温度-100)，请求出华氏温度对应的摄氏温度

```
var huashi float32 = 134.2
var sheshi float32 = 5.0 / 9 * (huashi - 100)
fmt.Printf("%v 对应的摄氏温度=%v \n", huashi, sheshi)
```

7、位运算符(了解)

位运算符对整数在内存中的二进制位进行操作。

运算符	描述
&	参与运算的两数各对应的二进位相与。 （两位均为 1 才为 1）
|	参与运算的两数各对应的二进位相或。 （两位有一个为 1 就为 1）
^	参与运算的两数各对应的二进位相异或，当两对应的二进位相异时，结果为 1。（两位不一样则为 1）
<<	左移 n 位就是乘以 2 的 n 次方。 “a<<b”是把 a 的各二进位全部左移 b 位，高位丢弃，低位补 0。
>>	右移 n 位就是除以 2 的 n 次方。 “a>>b”是把 a 的各二进位全部右移 b 位。

```
package main

import "fmt"

func main() {

    /*


        & 两位均为 1 才为 1

        | 两位有一个为 1 就为 1
```

```
        ^ 相异或   两位不一样则为 1

        <<   左移 n 位就是乘以 2 的 n 次方。   “a<<b”是把 a 的各二进位全部左移 b 位，高位丢弃，低位补 0。



        >> 右移 n 位就是除以 2 的 n 次方。

    */

    var a int = 5 //   101

    var b int = 2 //   010

    fmt.Println("a&b=", a&b) // 000   值 0

    fmt.Println("a|b=", a|b) // 111   值 7

    fmt.Println("a^b=", a^b) // 111   值 7

    fmt.Println("5>>2=", a>>b) // 5 右移 2 位   1

    fmt.Println("5<<2=", a<<b) // 5 左移 2 位   10100

    fmt.Println("5<<1=", 5<<1) // 1010

    fmt.Println("5>>1=", 5>>1) // 10

    fmt.Println("7>>2=", 7>>2) // 1

}
```