

运算符	描述	实例
+	相加	A + B 输出结果 30
-	相减	A - B 输出结果 -10
*	相乘	A * B 输出结果 200
/	相除	B / A 输出结果 2
%	求余	B % A 输出结果 0
++	自增	A++ 输出结果 11
--	自减	A-- 输出结果 9

Go 语言没有前置的 ++, --, (~~++~~a)

比较运算符

数组是值 可以比较 但是只有数组的维数和元素个数相同时才能进行比较

```
func TestCompareArray(t *testing.T) {
    a := [...]int{1, 2, 3, 4}
    b := [...]int{1, 2, 3, 4, 5}
    c := [...]int{1, 2, 3, 5}
    t.Log(a == b)
    t.Log(a == c)
}
```

```
PS F:\go\go_learning\src\ch4\operator_test> go test -v .\operator_test.go
# command-line-arguments_test [command-line-arguments.test]
.\operator_test.go:9:13: invalid operation: a == b (mismatched types [4]int and [5]int)
FAIL    command-line-arguments [build failed]
```

```
func TestCompareArray(t *testing.T) {
    a := [...]int{1, 2, 3, 4}
    b := [...]int{1, 2, 3, 4}
    //b := [...]int{1, 2, 3, 4, 5}
    c := [...]int{1, 2, 3, 5}
    t.Log(a == b)
    t.Log(a == c)
}
```

```
PS F:\go\go_learning\src\ch4\operator_test> go test -v .\operator_test.go
=== RUN    TestCompareArray
    operator_test.go:10: true
    operator_test.go:11: false
--- PASS: TestCompareArray (0.00s)
PASS
```

位运算符

与其他语言比较特殊的

^	位反（或位补）	^n等价于m ^ n，其中m和n同类型并且它的二进制表示中所有比特位均为1。比如如果n的类型为int8，则m的值为-1；如果n的类型为uint8，则m的值为255。
---	---------	--

注意：

- 在很多其它流行语言中，位反运算符是用~表示的。

&^ 清位运算符

清位运算符 &^ 是Go中特有的一个运算符。m &^ n 等价于 m & (^n)。

```
func TestBitClear(t *testing.T) {
    a := 7
    a = a &^ Readable
    t.Log(a&Readable == Readable, a&Writable == Writable, a&Executable == Executable)
}
```

```
=== RUN    TestBitClear
    operator_test.go:23: false true true
--- PASS: TestBitClear (0.00s)
```