

- 表达式
 - 算数运算符
 - 二元运算符
 - 一元运算符
 - 关系运算符
 - ==与!=
 - 混淆比较
 - 逻辑运算符
 - 三元运算符操作
 - 连接运算符
 - 优先级

表达式

算数运算符

这些运算操作都是实数

二元运算符

+ --加法
- --减法
* --乘法
/ --除法
^ --指数
% --取余

一元运算符

- --负数

关系运算符

```
<  --小于
>  --大于
<= --小于等于
>= --大于等于
== --全等于比较
~= --不等于比较
```

==与~=

在Lua中的定义是十分独特的，(当变量赋予的值是table表，userdata和function时，是数字或字符串时操作和其他语言相同)

== 仅是自己和自己比才会是true，其余都是false

~= 数值相同并不是等于，也是不等于

例子：

```
a = {}; a.x = 1; a.y = 0;
b = {}; b.x = 1; b.y = 0; --主.副就是C结构体中的表述方法表明主变量中的从属的值
c = a;
a == c;                    --自己和自己比才是true
a ~= b;                    --不相等
```

混淆比较

在字符串和数字混淆比较时报错

```
"2" < "15" --false
```

逻辑运算符

```
and --与
or  --或
not --非
```

注意：0在Lua中表示的仍是真，只有nil和false才会表示为假

三元运算符操作

C语言中操作

```
a ? b : c
```

Lua操作

```
(a and b) or c
```

连接运算符

连接两个字符串的运算符

```
..
```

示例：

```
print("hello" .. "world") -->helloworld
```

优先级

从高到低

优先级	成员
1	^
2	not -(负号)
3	* /
4	+ -
5	..(连接运算符)

优先级	成员
6	< > <= >= ~= ==
7	and
8	or