

Lua 运算符

运算符是一个特殊的符号，用于告诉解释器执行特定的数学或逻辑运算。Lua提供了以下几种运算符类型：

- 算术运算符
- 关系运算符
- 逻辑运算符
- 其他运算符

算术运算符

下表列出了 Lua 语言中的常用算术运算符，设定 A 的值为10，B 的值为 20：

操作符	描述	实例
+	加法	A + B 输出结果 30
-	减法	A - B 输出结果 -10
*	乘法	A * B 输出结果 200
/	除法	B / A w输出结果 2
%	取余	B % A 输出结果 0
^	乘幂	A^2 输出结果 100
-	负号	-A 输出结果v -10

实例

我们可以通过以下实例来更加透彻的理解算术运算符的应用：

```
a = 21
b = 10
c = a + b
print("Line 1 - c 的值为 ", c )
c = a - b
print("Line 2 - c 的值为 ", c )
c = a * b
print("Line 3 - c 的值为 ", c )
c = a / b
print("Line 4 - c 的值为 ", c )
c = a % b
```

```
print("Line 5 - c 的值为 ", c )
c = a^2
print("Line 6 - c 的值为 ", c )
c = -a
print("Line 7 - c 的值为 ", c )
```

以上程序执行结果为：

Line 1 - c 的值为	31
Line 2 - c 的值为	11
Line 3 - c 的值为	210
Line 4 - c 的值为	2.1
Line 5 - c 的值为	1
Line 6 - c 的值为	441
Line 7 - c 的值为	-21

关系运算符

下表列出了 Lua 语言中的常用关系运算符，设定 A 的值为10，B 的值为 20：

操作符	描述	实例
==	等于，检测两个值是否相等，相等返回 true，否则返回 false	(A == B) 为 false。
~=	不等于，检测两个值是否相等，相等返回 false，否则返回 true	(A ~= B) 为 true。
>	大于，如果左边的值大于右边的值，返回 true，否则返回 false	(A > B) 为 false。
<	小于，如果左边的值大于右边的值，返回 false，否则返回 true	(A < B) 为 true。
>=	大于等于，如果左边的值大于等于右边的值，返回 true，否则返回 false	(A >= B) 返回 false。
<=	小于等于，如果左边的值小于等于右边的值，返回 true，否则返回 false	(A <= B) 返回 true。

实例

我们可以通过以下实例来更加透彻的理解关系运算符的应用：

```
a = 21
b = 10

if( a == b )
then
    print("Line 1 - a 等于 b" )
else
```

```
print("Line 1 - a 不等于 b" )
end

if( a ~= b )
then
    print("Line 2 - a 不等于 b" )
else
    print("Line 2 - a 等于 b" )
end

if ( a < b )
then
    print("Line 3 - a 小于 b" )
else
    print("Line 3 - a 大于等于 b" )
end

if ( a > b )
then
    print("Line 4 - a 大于 b" )
else
    print("Line 5 - a 小于等于 b" )
end

-- 修改 a 和 b 的值
a = 5
b = 20
if ( a <= b )
then
    print("Line 5 - a 小于等于 b" )
end

if ( b >= a )
then
    print("Line 6 - b 大于等于 a" )
end
```

以上程序执行结果为：

```
Line 1 - a 不等于 b
Line 2 - a 不等于 b
Line 3 - a 大于等于 b
Line 4 - a 大于 b
Line 5 - a 小于等于 b
Line 6 - b 大于等于 a
```

逻辑运算符

下表列出了 Lua 语言中的常用逻辑运算符，设定 A 的值为 true，B 的值为 false：

操作符	描述	实例
and	逻辑与操作符。若 A 为 false，则返回 A，否则返回 B。	(A and B) 为 false。
or	逻辑或操作符。若 A 为 true，则返回 A，否则返回 B。	(A or B) 为 true。
not	逻辑非操作符。与逻辑运算结果相反，如果条件为 true，逻辑非为 false。	not(A and B) 为 true。

实例

我们可以通过以下实例来更加透彻的理解逻辑运算符的应用：

```
a = true
b = true

if ( a and b )
then
    print("a and b - 条件为 true" )
end

if ( a or b )
then
    print("a or b - 条件为 true" )
end

print("-----分割线-----" )

-- 修改 a 和 b 的值
a = false
b = true

if ( a and b )
then
    print("a and b - 条件为 true" )
else
    print("a and b - 条件为 false" )
end

if ( not( a and b ) )
then
    print("not( a and b) - 条件为 true" )
else
    print("not( a and b) - 条件为 false" )
end
```

以上程序执行结果为：

```
a and b - 条件为 true
a or b - 条件为 true
-----分割线-----
a and b - 条件为 false
not( a and b) - 条件为 true
```

其他运算符

下表列出了 Lua 语言中的连接运算符与计算表或字符串长度的运算符：

操作符	描述	实例
..	连接两个字符串	a..b ，其中 a 为 "Hello " ， b 为 "World", 输出结果为 "Hello World"。
#	一元运算符，返回字符串或表的长度。	#"Hello" 返回 5

实例

我们可以通过以下实例来更加透彻的理解连接运算符与计算表或字符串长度的运算符的应用：

```
a = "Hello "
b = "World"

print("连接字符串 a 和 b ", a..b )

print("b 字符串长度 ",#b )

print("字符串 Test 长度 ",#"Test" )

print("菜鸟教程网址长度 ",#"www.runoob.com" )
```

以上程序执行结果为：

连接字符串 a 和 b	Hello World
b 字符串长度	5
字符串 Test 长度	4
菜鸟教程网址长度	14

运算符优先级

从高到低的顺序：

```

^
not    - (unary)
*      /
+      -
..
<      >      <=      >=      ~=      ==
and
or

```

除了^和..外所有的二元运算符都是左连接的。

a+i < b/2+1	<-->	(a+i) < ((b/2)+1)
5+x^2*8	<-->	5+((x^2)*8)
a < y and y <= z	<-->	(a < y) and (y <= z)
-x^2	<-->	-(x^2)
x^y^z	<-->	x^(y^z)

实例

我们可以通过以下实例来更加透彻的了解 Lua 语言运算符的优先级：

```

a = 20
b = 10
c = 15
d = 5

e = (a + b) * c / d; -- ( 30 * 15 ) / 5
print("(a + b) * c / d 运算值为 :",e )

e = ((a + b) * c) / d; -- (30 * 15 ) / 5
print("((a + b) * c) / d 运算值为 :",e )

e = (a + b) * (c / d); -- (30) * (15/5)
print("(a + b) * (c / d) 运算值为 :",e )

e = a + (b * c) / d; -- 20 + (150/5)
print("a + (b * c) / d 运算值为 :",e )

```

以上程序执行结果为：

```

(a + b) * c / d 运算值为   :    90.0
((a + b) * c) / d 运算值为 :    90.0
(a + b) * (c / d) 运算值为 :    90.0
a + (b * c) / d 运算值为   :    50.0

```

← Lua 函数

Lua 字符串 →

+

4 篇笔记

✎ 写笔记