Operadores

Los operadores son unos signos gráficos que indican al intérprete que debe realizar operaciones lógicas, matemáticas o relacionales. Lua implementa varios tipos de operadores como veremos a continuación.

> En todos los ejemplos a continuación se asume que el valor de la variable `x` es 5 y el de la variable `y` es 10.

Operadores aritméticos

Los operadores aritméticos, como su nombre indica, son aquellos usados para la realización de operaciones matemáticas. Estos operadores permiten todas la operaciones básicas y se describen a continuación:

Operador	Descripción	Ejemplo
+	Suma dos operandos	`x + y = 15`
-	Resta el segundo operando al primero	`x - y = -5`
*	Multiplica ambos operandos	`x - y = -5`
/	Divide Numerador por denominador	`x / y = 0.5`
%	Módulo, devuelve el resto de la división	`y % x = 0`
^	Exponente, eleva un número a otro	`x ^ 3 = 125`
-	Unario, cambia el signo del valor	`-x = -5`

Operadores relacionales

Los operadores relacionales permiten realizar la comparación entre valores. Estos operadores son los usados en las estructuras de control de flujo y bucles para definir las condiciones.

Operador	Descripción	Ejemplo
==	Devuelve true si ambos operandos son iguales	`(x == y) -> false`
~=	Devuelve true si ambos operandos son diferentes	`(x ~= y) -> true`
>	Devuelve true si el operando de la izquierda es mayor que el de la derecha	`(x < y) -> true`
<	Devuelve true si el operando de la izquierda es menor que el de la derecha	`(x < y) -> true`
>=	Devuelve true si el operando de la izquierda es mayor o igual que el de la derecha	`(x >= y) -> false`
<=	Devuelve true si el operando de la izquierda es menor o igual que el de la derecha	`(x <= y) -> true`

Operadores lógicos

Los operadores lógicos permiten la realización de operaciones con expresiones de condición. Estos operadores se usan comúnmente en estructuras de control de flujo y bucles. En este ejemplo se asumen las variables `a = false` y `b = true`.

Operador	Descripción	Ejemplo
and	Operador de conjunción copulativa. Devuelve true si ambas expresiones a izquierda y derecha son true. False en caso contrario	`a and b -> false`
or	Operador de conjunción correlativa. Devuelve true si una de las expresiones a izquierda o derecha son true. False en caso contrario	`a or b -> true`
not	Operador de negación. Invierte el estado lógico de la expresión a la que precede	`not (a == b) -> true`

Otros operadores

Aparte de los operadores anteriores, Lua incluye dos operadores suplementarios, el operador de concatenación y el operador contador.

Operador	Descripción	Ejemplo
..	Concatenación de cadenas	`a = "Hola" .. " " .. "Mundo" -> Hola Mundo`
#	Devuelve la longitud o número de elementos de una cadena o una tabla	`#"Hola Mundo" -> 10`

Precedencia de los operadores

Cuando se escriben expresiones complejas se tornan necesarias las reglas de precedencia, permitiendo así al intérprete evaluar dichas expresiones. Es importante conocer bien estas reglas porque determinan de qué forma se evaluarán las expresiones. Si escribimos expresiones sin tener en cuenta las reglas de precedencia podrían producirse evaluaciones con resultados inesperados.

Las reglas de precedencia definen entonces como se agruparán los elementos de una expresión. Básicamente estas reglas definen el orden de precedencia de los diferentes operadores. Aquellos operadores que tienen mayor precedencia serán evaluados primero. A continuación se muestra la tabla de precedencia de operadores que están ordenados de mayor a menor precedencia:

Orden	Categoría	Operador	Evaluación
1	Unarios	`not # -`	De derecha a izquierda
2	Concatenación	`..`	De derecha a izquierda
3	Multiplicativos	`* / %`	De izquierda a derecha
4	Aditivos	`+ -`	De izquierda a derecha
5	Relacionales	`< > <= >= == ~=`	De izquierda a derecha
6	Igualdad	`== ~=`	De izquierda a derecha
7	Conjuntivos	`and`	De izquierda a derecha
8	Disyuntivos	`or`	De izquierda a derecha

Revisión #4
Creado 23 mayo 2023 07:48:25 por Guillermo
Actualizado 31 julio 2023 12:02:47 por Guillermo