LUA - OPERATORS

An operator is a symbol that tells the interpreter to perform specific mathematical or logical manipulations. Lua language is rich in built-in operators and provides the following type of operators −

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Misc Operators

This tutorial will explain the arithmetic, relational, logical, and other miscellaneous operators one by one.

Arithmetic Operators

Following table shows all the arithmetic operators supported by Lua language. Assume variable **A** holds 10 and variable **B** holds 20 then −

Show Examples

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by de-numerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
^	Exponent Operator takes the exponents	A^2 will give 100
-	Unary - operator acts as negation	-A will give -10

Relational Operators

Following table shows all the relational operators supported by Lua language. Assume variable **A** holds 10 and variable **B** holds 20 then −

Show Examples

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	$A == B$ is not true.
~=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	$A \ = B$ is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	$A > B$ is not true.

<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	$A < B$ is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	$A >= B$ is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	$A <= B$ is true.

Logical Operators

Following table shows all the logical operators supported by Lua language. Assume variable **A** holds true and variable **B** holds false then −

Show Examples

Operator	Description	Example
and	Called Logical AND operator. If both the operands are non zero then condition becomes true.	$AandB$ is false.
or	Called Logical OR Operator. If any of the two operands is non zero then condition becomes true.	$AorB$ is true.
not	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!$AandB$ is true.

Misc Operators

Miscellaneous operators supported by Lua Language include **concatenation** and **length**.

Show Examples

Operator	Description	Example
..	Concatenates two strings.	a..b where a is "Hello " and b is "World", will return "Hello World".
#	An unary operator that return the length of the a string or a table.	#"Hello" will return 5

Operators Precedence in Lua

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator −

For example, x = 7 + 3 * 2; Here x is assigned 13, not 20 because operator * has higher precedence than &plus; so it first get multiplied with 3*2 and then adds into 7.

Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Show Examples

Category	Operator	Associativity
Unary	not # -	Right to left
Concatenation	..	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	< > <= >= == ~=	Left to right
Equality	== ~=	Left to right
Logical AND	and	Left to right
Logical OR	or	Left to right