

# EUPHORIA - OPERATORS

[http://www.tutorialspoint.com/euphoria/euphoria\\_operators.htm](http://www.tutorialspoint.com/euphoria/euphoria_operators.htm)

Copyright © tutorialspoint.com

Euphoria provides a rich set of operators to manipulate variables. We can divide all the Euphoria operators into the following groups –

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

## The Arithmetic Operators

Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. The following table lists the arithmetic operators –

Assume integer variable A holds 10 and variable B holds 20 then –

[Show Examples](#)

Operator	Description	Example
+	Addition - Adds values on either side of the operator	$A + B$ will give 30
-	Subtraction - Subtracts right hand operand from left hand operand	$A - B$ will give -10
*	Multiplication - Multiplies values on either side of the operator	$A * B$ will give 200
/	Division - Divides left hand operand by right hand operand	$B / A$ will give 2
+	Unary plus - This has no impact on the variable value.	+B gives 20
-	Unary minus - This creates a negative value of the given variable.	-B gives -20

## The Relational Operators

There are following relational operators supported by Euphoria 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	$A != B$ is true.

condition becomes 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.

## The Logical Operators

The following table lists the logical operators –

Assume boolean variables A holds 1 and variable B holds 0 then –

[Show Examples](#)

Operator	Description	Example
and	Called Logical AND operator. If both the operands are non zero then then condition becomes true.	$A \text{ and } B$ is false.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	$A \text{ or } B$ is true.
xor	Called Logical XOR Operator. Condition is true if one of them is true, if both operands are true or false then condition becomes false.	$A \text{ xor } B$ is true.
not	Called Logical NOT Operator which negates the result. Using this operator, true becomes false and false becomes true	$\text{not } B$ is true.

You can also apply these operators to numbers other than 1 or 0. The rule is: zero means false and non-zero means true.

## The Assignment Operators:

There are following assignment operators supported by Euphoria language –

[Show Examples](#)

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	$C = A + B$ will assigne value of $A + B$ into C

+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C / A
&=	Concatenation operator	C &= {2} is same as C = {C} & {2}

**Note:** The equals symbol '=' used in an assignment statement is not an operator, it's just part of the syntax of the language.

## Misc Operators

There are few other operators supported by Euphoria Language.

## The '&' Operator

Any two objects may be concatenated using the & operator. The result is a sequence with a length equal to the sum of the lengths of the concatenated objects.

Following is the example –

```
#!/home/euphoria-4.0b2/bin/eui
sequence a, b, c
a = {1, 2, 3}
b = {4}
c = {1, 2, 3} & {4}

printf(1, "Value of c[1] %d\n", c[1] )
printf(1, "Value of c[2] %d\n", c[2] )
printf(1, "Value of c[3] %d\n", c[3] )
printf(1, "Value of c[4] %d\n", c[4] )
```

This would produce following result –

```
Value of c[1] 1
Value of c[2] 2
Value of c[3] 3
Value of c[4] 4
```

## Precedence of Euphoria Operators

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 + 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.

<b>Category</b>	<b>Operator</b>	<b>Associativity</b>
Postfix	function/type calls	
Unary	+ - ! not	Right to left
Multiplicative	* /	Left to right
Additive	+ -	Left to right
Concatenation	&	Left to right
Relational	> >= < <=	Left to right
Equality	= !=	Left to right
Logical AND	and	Left to right
Logical OR	or	Left to right
Logical XOR	xor	Left to right
Comma	,	Left to right

Loading [MathJax]/jax/output/HTML-CSS/jax.js