

# Arithmetic Expressions

## Description

Arithmetic expressions are combinations of variables, constants, or functions that specify a rule to calculate a single numeric value.

## Syntax

***result = expression [operator expression ...]***

## Operators

Priority	Operator	Keyboard Symbol
high 1	unary plus	+
1	unary minus	-
2	exponentiation	**
3	multivalued multiplication	***
3	multivalued division	///
3	multiplication	*
3	division	/
4	multivalued addition	+ + +
4	multivalued subtraction	---
4	addition	+
4	subtraction	-
5	multivalued concatenation	:::
low 5	concatenation	:

## Simple Arithmetic Expressions

The simplest arithmetic expressions consist of a single numeric constant, a variable or an intrinsic function. Following are examples of simple arithmetic expressions:

3.1416	1000	.0005
QTA	INT(2.5)	"144"

An arithmetic operator (+, /, \*, (), etc.) can combine two arithmetic expressions and produce a third arithmetic expression. The third arithmetic expression may in turn be combined with other expressions. For examples

bal\_ford + mo\_end

scoreA + scoreB / 2

amt \* .065

sin(D) / cos(D)

Arithmetic operations follow a strict order of priority. See [Arithmetic Operators](#) for details.

BASIC+ considers character string values that contain only numeric characters to be decimal numbers. For example, the expression 123 + "321" will evaluate to 444.

Character string values that contain non-numeric characters will produce an error message when the program is run. The string value will assume the value of 0 (zero) in this case and the program will terminate.

## Correct Use of Arithmetic Operators

result = - 6 + 3 *The result variable will be -3.*

`result = - (6 + 3)` *The result variable will be - 9.*

`A = 5 result = 3 * 4 ** 2 + A`

The result variable will be 4 raised to the second power multiplied by 3 plus the current value of 5. The result variable will be 53.

## See Also

[Arithmetic operators](#), [Multi-value Arithmetic operators](#)