Cheatography

REXX Scripting Language Cheat Sheet

by br0p0p via cheatography.com/6770/cs/1106/

REXX Language Logo

This cheat sheet was made with information gathered from IBM's VSE/REXX Reference 6.1.

Arithmetic Operators

+	Add
-	Subtract
*	Multiply
/	Divide
%	Integer Divide
//	Remainder (not modulo)
**	Power (raise to whole-number)
Prefix -	Same as: *0 - number*
Prefix +	Same as: *0 + number*

Comparison Operators

=	Equal (numerically or padded)
\=, /=	Not equal
>	Greater than
<	Less than
<>, ><	Greater than or less than (same as not equal)
>=	Greater than or equal to
<=	Less than or equal to
\>	Not greater than
\<	Not less than
==	Strictly equal (identical)
\==, /==	NOT strictly equal

Comparison Operators (cont)

>>	Strictly greater than
<<	Strictly less than
>>=	Strictly greater than or equal to
<<=	Strictly less than or equal to
\>>	Strictly NOT greater than
\<<	Strictly NOT less than

Boolean Operators

&	AND
|	Inclusive OR (either is true evaluates to 1)
&&	Exclusive OR (either but not both is true evaluates to 1)
Prefix \	Logical NOT (Negates: 0->1 and 1->0

Concatenation Operators

(blank)	Concatenate terms with one blank in between
||	Concatenate without an intervening blank
(abuttal)	Concatenate without an intervening blank

You can force concatenation without a blank line by using the || operator.

The **abuttal** operator is assumed between two terms that are not separated by another operator. This can occur when two terms are syntactically distinct, such as a literal string and a symbol, or when they are separated only by a comment.

Concatenation: *A (Z)*
Function Call: *A(Z)*

Other Operators

/*	Begin Comment
*/	End Comment
A-Z, a-z, 0-9, @ # $. ! ? _	Valid Symbol Characters
,	Continuation Character
.	Constant Symbol (at beginning of string)

Valid Symbols:
Fred
Albert.Hall
WHERE?
Valid Numbers:
12
'-17.9'
127.0650
73e+128
' + 7.95E5 '

Constant example: *.12345*

Negative numbers in expressions must use quotes.

The continuation character is used to continue a clause onto the next line.

Common Commands

DROP	unassigns variable: ex. drop *variable name*
EXIT	leaves program unconditionally
INTERPRET	processes instructions that have been built dynamically by evaluating the expression: ex. interpret *expression*
LEAVE	causes an immediate exit from one or more repetitive DO loops (anything other than simple DO)
SAY	writes line to current output stream