A *complete* listing of operators in Nix, and their predence.

> This article was originally published at https://gist.github.com/joepie91/c3c047f3406aea9ec65eebce2ffd449d.

> The information in this article has since been absorbed into the official Nix manual. It is kept here for posterity. It may be outdated by the time you read this.

Lower precedence means a stronger binding; ie. this list is sorted from strongest to weakest binding, and in the case of equal precedence between two operators, the associativity decides the binding.

Prec	Abbreviation	Example	Assoc	Description
1	SELECT	`e . attrpath [or def]`	none	Select attribute denoted by the attribute path `attrpath` from set `e`. (An attribute path is a dot-separated list of attribute names.) If the attribute doesn't exist, return `default` if provided, otherwise abort evaluation.
2	APP	`e1 e2`	left	Call function `e1` with argument `e2`.
3	NEG	`-e`	none	Numeric negation.
4	HAS_ATTR	`e ? attrpath`	none	Test whether set `e` contains the attribute denoted by `attrpath` ; return true or false.
5	CONCAT	`e1 ++ e2`	right	List concatenation.
6	MUL	`e1 * e2`	left	Numeric multiplication.
6	DIV	`e1 / e2`	left	Numeric division.
7	ADD	`e1 + e2`	left	Numeric addition, or string concatenation.
7	SUB	`e1 - e2`	left	Numeric subtraction.
8	NOT	`!e`	left	Boolean negation.
9	UPDATE	`e1 // e2`	right	Return a set consisting of the attributes in `e1` and `e2` (with the latter taking precedence over the former in case of equally named attributes).
10	LT	`e1 < e2`	left	Less than.
10	LTE	`e1 <= e2`	left	Less than or equal.
10	GT	`e1 > e2`	left	Greater than.

Prec	Abbreviation	Example	Assoc	Description
10	GTE	`e1 >= e2`	left	Greater than or equal.
11	EQ	`e1 == e2`	none	Equality.
11	NEQ	`e1 != e2`	none	Inequality.
12	AND	`e1 && e2`	left	Logical AND.
13	OR	`e1 || e2`	left	Logical OR.
14	IMPL	`e1 -> e2`	none	Logical implication (equivalent to `!e1 || e2`).

Revision #1
Created 2024-12-11 02:00:44 UTC by joepie91
Updated 2024-12-11 15:56:33 UTC by joepie91