APL Reference

Operation	*Expression*	*Value*
Data creation	`newint: int -> aplval` make a new integer value	`newint 1`
	`newveci: int list -> aplval` make a new int vector	`newveci [1;2;3]`
Sample data	`A = rho (newveci [2;3]) (indx (newint 6))`	1 2 3 4 5 6
	`V = newveci [2;4;6]`	2 4 6
	`C = newveci [1;0]`	1 0
	`D = newveci [1;0;1]`	1 0 1
Arithmetic	`A *@ A`	1 4 9 16 25 36
	`V -@ (newint 1)`	1 3 5
	`+@`, `/@`, `%@`	
Unary	`~@ V`	-2 -4 -6
Relational	`A >@ (newint 4)`	0 0 0 0 1 1
	`=@`, `<@`, `>=@`, `<=@`, `<>@`	
Logical	`&@`, `|@`	
Reduction	`!+ V`	12
	`maxR A`	3 6
	`!-`, `!*`, `!/`, `!%`, `!|`, `!&`, `minR`	
Compression	`D % V`	2 6
	`C % A`	1 2 3 (a 1,3-matrix)
Shape	`shape A`	2 3
Ravelling	`ravel A`	1 2 3 4 5 6
	`ravel (newint 1)`	1
Restructuring	`rho (shape A) V`	2 4 6 2 4 6
	`rho (shape V) C`	1 0 1
	Note: `rho` requires a vector as the first argument	
Catenation	`A ^@ C`	1 2 3 4 5 6 1 0
Index generation	`indx (newint 5)`	1 2 3 4 5
Transposition	`trans A`	1 4 2 5 3 6
Subscripting	`V @@ (indx (newint 2))`	2 4
	`A @@ (newint 1)`	1 2 3 (a 1,3-matrix)
	`(trans A) @@ (indx (newint 2))`	1 4 2 5
Auxiliary	`show: aplval -> unit` displays the given value	
	`isTrue: aplval -> bool` is the given value True or False?	