Math Operations

Command	Structure	Obs
Addition	[number] **+** [number]	
Subtraction	[number] **-** [number]	
Multiplication	[number] ***** [number]	
Dividion	[number] **/** [number]	
Exponential	[base] ****** [potencia]	
Modulo	[number] **%** [number]	Retorna resto da divisão
Divisão Inteira	[number] **//** [number]	Retorna valor inteiro da divisão

Comparison

Command	Structure
Equal to	**==**
Not Equal to	**!=**
Less than	**<**
Less or Equal to	**<=**
Greater than	**>**
Greater or Equal to	**>=**

Functions

Command	Structure	Example	Obs
Max	**max (** [#1,#2,#3] **)**	**max (**1,2,3**)**	Valor maximo
Min	**min (** [#1,#2,#3] **)**	**min (**1,2,3**)**	Valor minimo
Abs	**abs (** [#1,#2,#3] **)**	**abs (**1,2,3**)**	Valor absoluto
Ascending Sort	**sorted (** [list] **)**	**sorted ([**1,7,5,2,3**])**	Ordem crescente/alfabetica
Sum	**sum (** [list] **)**	**sum ([**1,7,5,2,3**])**	Soma valores da lista
Power	**pow (** [base], [power] **)**		Exponencial

Logic Operations

Command	Structure
And	[expression] **and** [expression]
Or	[expression] **or** [expression]
Not	**not** [expression]

Binary Operations

Command	Structure	Example	Obs
Right Shift	[number] **>>** [number]	5 **>>** 4	**0b**0010100 **>>** 3 **== 0b**000010 **(**20 **>>** 3 **=** 2**)**
Left Shift	[number] **<<** [number]	5 **<<** 3	**0b**000101 **<<** 3 **== 0b**101000 **(**5 **<<** 3 **=** 40**)**
Bitwise AND	[number] **&** [number]	**print bin(**0b1110 **&** 0b101**)**	a**=0b**00101010 **(**42**)** b**=0b**00001111 **(**15**)** a **&** b **== 0b**00001010 **(**10**)**
Bitwise OR	[number] **I** [number]	**print bin(**0b1110 **I** 0b101**)**	a**=0b**00101010 **(**42**)** b**=0b**00001111 **(**15**)** a **I** b **== 0b**00101111 **(**47**)**
Bitwise XOR	[number] **^** [number]	**print bin(**0b1110 **^** 0b101**)**	a**=0b**00101010 **(**42**)** b**=0b**00001111 **(**15**)** a **^** b **== 0b**00100101 **(**37**)**
Bitwise NOT	**~** [number]	**~**88	**add 1 to the number and make it negative**
Print Binary	**print 0b**[number]	**print 0b**101	
Make # in Binary	**bin(** [int number] **)**	**bin(**3**)**	
Make # in Octal	**oct(** [int number] **)**	**oct(**6**)**	
Make # in Hexadecimal	**hex(** [int number] **)**	**hex(**2**)**	