

RedCrab

The Calculator

Neuheiten in der Bedienungsanleitung Version 4.10

copyright © by Redchillicrab, Singapore 2009, 2010, 2011,2012

AddIn

Der Operator **AddIn** liefert als Resultat ein Datenfeld mit einer linearen Zahlenreihe. Die Länge der Zahlenreihe entspricht der des linken Operanden. Die erste Zahl wird vom ersten Elements des linken Operanden übernommen. Die folgenden Werte werden je um den Wert des rechten Operanden erhöht.

Beispiel: `[4..8] addin 0.5 = 4.0 4.5 5.0 5.5 6.0`

Der Wert im ersten Element definiert den Startwert der Reihe, die folgenden Werte haben keine Bedeutung. Sie sind Platzhalter um die Größe des Feldes zu bestimmen. Das folgende Beispiel ergibt daher das gleiche Resultat.

Beispiel: `[4..0] addin 0.5 = 4.0 4.5 5.0 5.5 6.0`

Eine absteigende Zahlenreihe wird durch einen negativen Operanden erzeugt.

Beispiel: `[4..8] addin -0.5 = 4.0 3.5 3.0 2.5 2.0`

AddTo

Der Operator **AddTo** liefert als Resultat ein Datenfeld mit einer linearen Zahlenreihe. Die Zahlenreihe beginnt mit dem erste Wert des linken Operanden und wird schrittweise um den Wert des rechten Operanden erhöht, bis der Wert des nächsten Operanden erreicht ist.

Beispiel: `[4, 7] addto 0.5 = 4.0 4.5 5.0 5.5 6.0 6.5 7.0`

`[4, 1] addto 0.5 = 4.0 3.5 3.0 2.5 2.0 1.5 1.0`

```
[4,6,5] addto 0.5= 4.0 4.5 5.0 5.5 6.0 5.5 5.0
```

AddOn

Die Funktion **AddOn** addiert fortlaufend jeden Wert einer Zahlenreihe zum nächsten Wert.

Beispiel: `addon([10,30,25,-10,15]) = 10 40 65 55 70`

MulIn

Der Operator **MulIn** liefert als Resultat ein Datenfeld mit einer logarithmischen Zahlenreihe. Die Länge der Zahlenreihe entspricht der des linken Operanden. Die erste Zahl wird vom ersten Elements des linken Operanden übernommen. Die folgenden Werte sind je das Produkt aus der fortlaufenden Multiplikation mit dem rechten Operanden.

Beispiel: `[2..8] mulin 2.0= 2 4 8 16 32 64 128`

Der Wert im ersten Element definiert den Startwert der Reihe, die folgenden Werte haben keine Bedeutung. Sie sind Platzhalter um die Größe des Feldes zu bestimmen. Das folgende Beispiel ergibt daher das gleiche Resultat.

Beispiel: `[2..-4] mulin 2.0= 2 4 8 16 32 64 128`

Eine absteigende Zahlenreihe wird durch einen Wert, kleiner als eins, erzeugt.

```
[2..8] mulin 0.5= 2.0 1.0 0.50 0.25 0.13 0.06 0.03
```

MulTo

Der Operator **MulTo** liefert als Resultat ein Datenfeld mit einer logarithmischen Zahlenreihe. Die Zahlenreihe beginnt mit dem erste Wert des linken Operanden und wird schrittweise mit dem Wert des rechten Operanden multipliziert, bis der Wert des nächsten Operanden erreicht ist.

Beispiel: `[1,150] multo 2 = 1 2 4 8 16 32 64 128`

```
[150,3] multo 0.5 = 150 75 37.5 18.8 9.38 4.69
```

Bei aufsteigen Reihen muß der Multiplikator > 1 , bei absteigende < 1 sein. Negative Werte und die Werte 0 und 1 sind nicht erlaubt.

MulAd

Der Operator **MulAdd** liefert als Resultat eine Zahlenreihe in der jeder Wert des linken Operanden zuerst mit dem rechten Operanden multipliziert wird und dann zum nächsten Wert addiert wird.

Beispiel:

```
x = [1..5] fill 100 =100.00 100.00 100.00 100.00 100.00  
y = x mulad 1.1 = 110.00 231.00 364.10 510.51 671.56
```

Der rechte Operant kann auch ein eindimensionales Feld enthalten.

Fill

Das Resultat des Operators ***Fill*** ist ein Datenfeld in der Größe des linken Operanden in dem allen Elemente der Wert des rechten Operanden zugewiesen ist.

Beispiel: $x = [1..5]$ **fill** 8 = 8 8 8 8 8

Patt

Das Resultat des Operators ***Patt*** ist ein Datenfeld in der Größe des linken Operanden in dem allen Elementen fortlaufend die Werte des rechten Feld-Operanden zugewiesen sind.

Beispiel: $x = [1..10]$ **patt** [1,1,2] = 1 1 2 1 1 2 1 1 2 1