

Überblick über die Operationen von **o++o**

(Stand 06.01.2022)

Die meisten bekannten Operationen besitzen eine Stelligkeit (Arität).

Die Quadratwurzel benötigt beispielsweise nur einen Inputwert – das ist in der Regel eine Zahl. Im o++o-Datenmodell kann es auch eine Liste von Zahlen sein. Dann wird die Wurzel aus jeder der Zahlen gezogen. Die Liste wird dann als **ein** Inputwert gezählt, obwohl sie 10 oder 1000 Zahlen umfassen kann. D.h., `sqrt` bleibt auch in diesem Fall einstellig (unär).

`sqrt` muss in o++o dieser Liste folgen. Dadurch benötigt man keine zusätzlichen Klammern.

Anstelle von `sqrt([2 4 7])` kann man in o++o

```
[2 4 7] sqrt
```

oder sogar

```
2 4 7 sqrt
```

schreiben. In beiden Fällen erhält man das gleiche Ergebnis. Man kann `sqrt` sogar auf ein beliebiges Tabment folgen lassen.

Noch bekannter als die Wurzeloperation ist beispielsweise die Addition `+`. Sie benötigt 2 Inputwerte. Ansonsten kann sie nicht angewandt werden und man erhält eine Fehlermeldung (syntaktischer Fehler). Bei

```
3 + 4
```

ist 3 der erste und 4 der zweite Inputwert. Auch hier kann als erster Inputwert eine Liste oder ein anderes Tabment eingesetzt werden.

```
1 3 7 +4
```

ergibt

```
5 7 11
```

Hier und bei vielen anderen Operationen stimmt der Typ des Ergebnisses mit dem Typ des ersten Inputtabments überein. Das obige Ergebnis ist daher ebenfalls eine Liste von Zahlen. Zweistellige Operationen werden in o++o immer zwischen die beiden Inputtabmente geschrieben. Man kann auch sagen, sie werden wie die einstelligen Operationen nach dem ersten Inputwert geschrieben. Das Gleiche trifft in o++o auch für viele dreistellige Operationen zu. „!“ wird als Trennsymbol zwischen dem zweiten und dritten Inputwert benutzt.

```
Hadmersleben subtext 4 ! 5
```

hat beispielsweise das Ergebnis

```
mersl
```

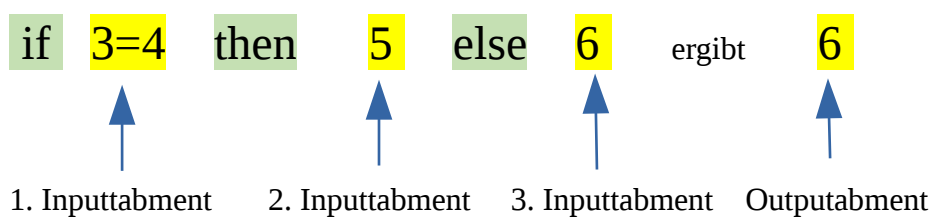
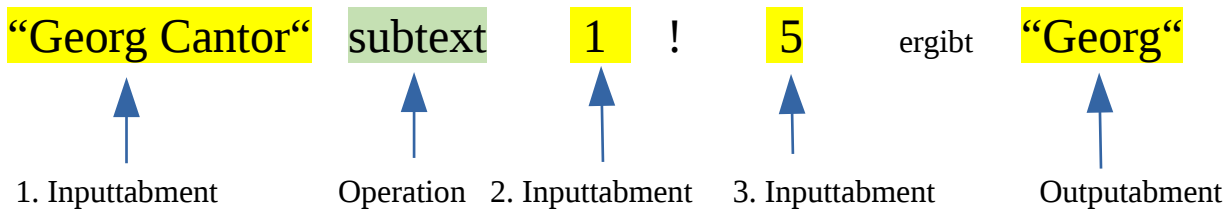
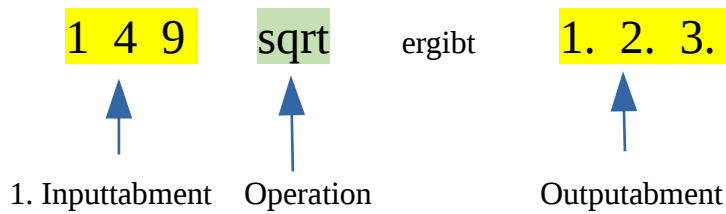
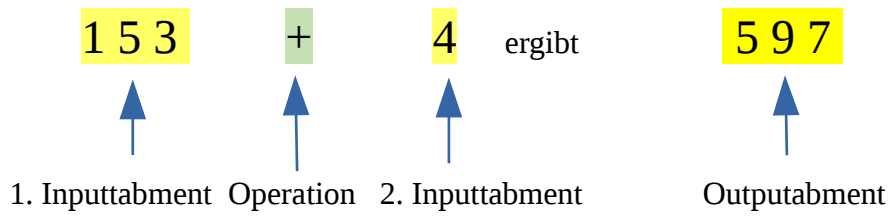
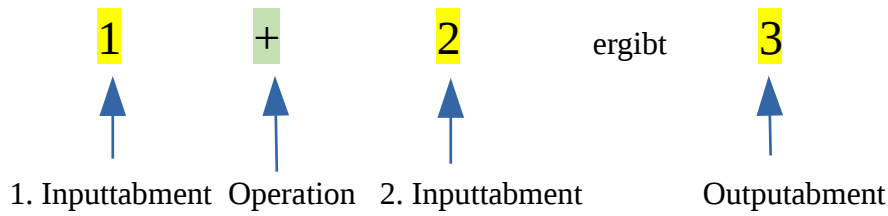
Der erste Inputwert ist „Hadmersleben“. Der zweite Inputwert (4) gibt hierbei die Stelle des Anfangsbuchstaben des Teilworts an und der dritte Inputwert (5) gibt die gewünschte Länge an.

Bei einigen Operationen wird auf gewohnte Schreibweisen zurückgegriffen.

```
if X>3 then 5 else 6
```

benötigt ebenfalls 3 Inputwerte (hier: einen Wahrheitswert, die 5 und die 6). Wenn wir für X 10 einsetzen, ist die Bedingung erfüllt und die if-then-else-Operation liefert 5 zurück.

Im Folgenden werden In- und Outputdaten noch einmal an 5 Beispielen illustriert.



Es sei bereits an dieser Stelle angemerkt, dass in vielen Fällen das Ergebnis der vorangehenden Zeile als erstes Inputtabment einer Operation zählt:

noten.tab

++:

ergibt den Durchschnitt aller Zahlen, die im ersten Inputtabment noten.tab vorkommen. Analog wird durch das Programm

xx.tab

+2

zu jeder Zahl der Tabelle xx.tab 2 addiert. xx.tab ist erstes Inputtabment und 2 das zweite. Bei namen.tab

subtext 3 ! 4

wird in analoger Weise aus jedem Text (TEXT oder WORT)-Wert ein Text der Länge 4 extrahiert. Das Ergebniswort beginnt jeweils mit dem dritten Zeichen des Inputworts. Hier hat die dreistellige Operation subtext die Inputtabmente namen.tab, 3 und 4.

In untenstehender Tabelle benutzen wir folgende Abkürzungen.

TT = Tabment Typ

TT1 = Tabment Typ des ersten Inputtabments

Zahl = ZAHL oder PZAHL oder RATIO

Text = TEXT oder WORT

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
+	2	TT1	Addition von Zahlen und Verbindung von Texten	1 3 + 2.1 = 3.1 5.1 xy ab + de = xyde abde
*	2	TT1	Multiplikation	2 3 5 * 2 = 4 6 10
-	2	TT1	Subtraktion	3 - 2 ergibt 1 1'234 - 345 ergibt 889
:	2	TT1	Division	3:4=0.75
++	1	Zahl	Summe	2 3 6 ++ = 11
**	1	Zahl	Produkt	1 3 5 ** = 15
--	1	Zahl	Mehrfachsubtraktion	20 5 4 -- = 11
::	1	Zahl	Mehrfachdivisionen	64 2 2 :: = 16
%	2	Zahl	Prozentwert	200 % 5 = 10
+%	2	Zahl	Prozentwert hinzufügen	200 +% 5 = 210
-%	2	Zahl	Prozentwert abziehen	200 -% 5 = 190
++:	1	PZAHL	arithmetischer Durchschnitt	1 2 3 2 ++: = 2.0
++1	1	ZAHLEN	Anzahl	3 4 7 9 ++1 = 4
++text	1	TEXT	zu Text verbinden	ab cde fg ++text = abcdefg
++textsep	2	TEXT	zu Text mit Seperator verbinden	ab cde fg ++text “;“ = “ab;cde;fg“
+m	2	X1,..Xn m	Vereinigung zu Menge	{1 2 3} +m {6 3} = {1 2 3 6}
+b	2	X1,..Xn b	Vereinigung zu Multimenge	{1 2 3} +b {6 3} = {{1 2 3 3 6}}

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
+l	2	X1,..Xn l	Vereinigung zu Liste	3 5 5 +l 6 3 = 3 5 5 6 3
-m	2	X1,..Xn m	Differenz zu flachen Mengen	{ 2 4 5 } -m { 6 2 } = { 4 5 }
-b	2	X1,..Xn b	Differenz zu flachen Multimengen	{{1 2 4 4 4}} -b {{2 4}} = {{1 4 4}}
-l	2	X1,..Xn l	Differenz zu flachen Listen	1 3 4 -l 5 3 = 1 4
:m	2	X1,..Xn m	mengentheoretischer Durchschnitt	{1 3 7} :m {3 9} = {3}
:b	2	X1,..Xn b	bag-Durchschnitt	{{2 2 4}} :b {{2 7 7}} = {{2}}
:l	2	X1,..Xn l	Listen-Durchschnitt	2 2 4 :l 3 2 4 = 2 4
*m	2	X1,..Xn m	kartesisches Produkt	{1 2} *m {3 4 6} = {1,3 1,4 1,6 2,3 2,4 2,6}
*b	2	X1,..Xn b	kartesisches Multimengenprodukt	{{1 2}} *b {{2 2}} = {{1,2 1,2 2,2 2,2}}
*l	2	X1,..Xn l	kartesisches Listenprodukt	1 2 *l 2 3 = 1,2 1,2 2,2 2,2
,	2	TT1,TT2	Komma; Paarbildung	1 2, 3 ergibt ZAHL1, ZAHL 1 2 3
, ,	2	TT1,TT2	Das Ergebnis ist ebenfalls ein Paar. , , trennt aber schärfer als , .	2, 3 * 2 ergibt 4, 6 2, , 3 *2 ergibt 2, 6
=	2	BOOL	Gleichheit	1 = 2 ergibt si
:= [at]	1	TT1+ neue Spalte	Zuweisung	X:=1 Y:= X+2 ergibt X, Y 1 3
:= first next at	1	TT1+ neue Spalte	rekursive Zuweisung	BETRAG:=first 100 next BETRAG pred* 1.03 at JAHR
:= firstonr nextonr [at]	1	TT1 +neue Spalte	rekursive Erweiterung mit otto Nummern	ANZ2:=firstonr ANZ nextonr ANZ2 pred *ANZ leftat ANZ
::=	1	TT1	Überschreiben	X::= X+3*Y
=: \$Name	1	TT1	Zuweisung für Variable	2, 3 =: \$X
&	2	TT1	Konjunktion (und)	1=1 & 2=3 =no

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
	2	TT1	Disjunktion (oder)	1=1 2=3 = si
->	2	TT1	logische Implikation	si → si = si
<->	2	TT1	logische Äquivalenz	si <-> no = no
<	2	BOOL	kleiner	3<4 = si
>	2	BOOL	größer	3>4 = no
<=	2	BOOL	kleiner gleich	2 <= 2 ergibt si
>=	2	BOOL	größer gleich	2 >= 4 ergibt no
!=	2	BOOL	ungleich	2 != 1 ergibt no
*mat	2	X1, ..Xn l	Matrizenmultiplikation	(1,2) *mat [2 3] = 8
-1mat	1	TT1	inverse Matrix	<TAB! X1,X2,X3 l 1 0 2 0 2 0 0 0 8 !TAB> -1mat = X1, X2, X3 l 1. -0. -0.25 -0. 0.5 -0. 0. -0. 0.125
&&	1	BOOL	für alle	si,66,si && = si
	1	BOOL	Existenzaggregation	1=2,no = no
..	2	Zahl l	von bis	1 .. 4 = 1 2 3 4
...	3	Zahl l	von bis ! schrittweite	0 ... 6!2 = 0 2 4 6
..x	3	Zahl l	Zufallszahlen von bis ! anzahl	1 ..x 6!3= 5 3 2
1in	2	BOOL	ein Wort der linken Seite ist in der rechten enthalten	[1 2] 1in "1 3 4" = si
'3	1	TT1	große Zahlen in 3er Blöcken formatieren	1234567890 '3 ergibt das WORT ?????????? 1'234'567'890
'4	1	TT1	große Zahlen in 4er Blöcken formatieren	12345.67898 '4 ergibt das WORT ??????? 1'2345.6789'8
abs	1	TT1	absoluter Betrag	-3 abs = 3 7 abs =7
add	2	TT1	füge die zweite Tabelle zur ersten hinzu, wobei	<TAB! X1,X2 l

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
			die einzelnen Spaltennamen übereinstimmen müssen	<pre> 1 0 0 2 !TAB> add <TAB! X2,X1,X3 m 4 5 6 7 8 9 !TAB> ergibt X1, X2 l 1 0 0 2 5 4 8 7 </pre>
at	-		rechts neben	Z:= Y+3 at X
aus	-		(Neu-)Beginn im Programm	aus fluesse.tabh
avec	1	TT1	Selektion (mit)	fluesse.tabh avec LAENGE >800
comp	1+name	Name	Komponente	NAME, VORNAME,ORT Mill Paul Halle comp ORT ergibt Halle; siehe auch nth
cos	1	TT1	Kosinus	pi cos = -1.
det	1	Zahl	Determinante	<pre> <TAB! X1,X2,X3 l 1 0 2 0 2 0 0 0 8 !TAB> det = 16. </pre>
div	2	ZAHL	ganzzahlige Division	11 div 5 =2
divrest	2	ZAHL,ZAHL	ganzzahlige Division mit Rest	11 divrest 5 = 2,1 (nicht 2.1)
e	0	PZAHL	Eulersche Konstante	e hoch 3 ln ergibt 3.
empty_t	0	empty_s	Leere Tabelle mit leerem Kopf	aus empty_t X:=1 ergibt X 1
gib	1+schemen+..	S2	restrukturiere, transformiere ein	aus studenten.tab gib FAK, (ORT,NAME m)m

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
			Tabment in ein Tabment mit gegebenen Schema bzw. gegebener TTD	
<code>giball</code>	1+sche me	S2	alle Werte	<code>giball X Y l</code> Liste aller X- und Y-Elemente (beliebige Tiefe); entspricht dem Doppelslash <code>...//X Y</code> von XPath
<code>gibtop</code>	1+sche me	S2	nur die höchsten Werte	<code>gibtop Xl</code> entspricht dem Slash: <code>t/X</code> : Liste aller X-Subtabmente von t, die in der obersten Ebene von t vorkommen.
<code>hoch</code>	2	TT1	Potenz	<code>4 hoch 1/2 = 2.</code>
<code>if then else</code>	3	TT2=TT3	wenn dann sonst	<code>if 3=4 then 5 else 6</code> ergibt 6
<code>if then</code>	2	TT2	wenn dann	<code>if 3=4 then 5</code> ergibt leer
<code>igib</code>	1+sche me	S2	join und Restrukturierung	<code>studenten.tab, faks.tab</code> <code>igib FAK, DEKAN, NAME m</code>
<code>in</code>	2	BOOL	Enthaltensein von Worten und Zahlen	<code>"1 2 1"</code> in <code>"1 2"</code> = si <code>"1 2 3"</code> in <code>"1 1 2"</code> = no
<code>inmath</code>	2	BOOL	mathematisches Enthaltensein	<code>[1 3]</code> inmath <code>[1 4 3]</code> =si <code>2</code> inmath <code>{6 7 2}</code> =si
<code>keys</code>	2	TT1	effiziente Selektion in Mengen oder Listen	<code>Xl:= 1 ..40</code> <code>Y:=X*X</code> <code>gib X,Y m</code> <code>keys [7 34]</code> ergibt in tab-Format: <code>X, Y m</code> <code>7 49</code> <code>34 1156</code> oder <code>keys <![yy, [y2] zz]!></code>
<code>keyslike</code>	2	TT1	effiziente Selektion in Mengen oder Listen	<code><TAB!</code> <code>NAME, ORT m</code> <code>Clara Oehna</code> <code>Claudia Dallgow</code> <code>Sophia Dallgow</code> <code>!TAB></code> <code>keyslike ["*ia"]</code> ergibt <code>NAME, ORT m</code> <code>Claudia Dallgow</code>

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
				Sophia Dallgow
leftat	-	-	links neben	BRUTTO:=NETTO*1.19 leftat NETTO
letterb	1	WORTb	in Bag von Buchstaben zerlegen	"Heute ist ." letterb ergibt im tabh-Format: WORTb . e e H i s t t u
letterm	1	WORTm	in Menge von Buchstaben zerlegen	"Heute ist Dienstag." letterm ergibt im tabh-Format: WORTm . a D e g H i n s t u
letterl	1	WORTl	in Liste von Buchstaben zerlegen	"Heute ist Dienstag." letterl ergibt im tab-Format: WORTl H e u t e i s t D i e n s t a g .
like	2	BOOL	ähnlich	Hadmersleben like "? admers*" = si '?': repräsentiert ein Zeichen '*': null oder mehrere Zeichen
linreg	1	Y0,ANSTIEG	lineare Regression	<TAB! FLAPREIS,VERKAUFT l 20 0 16 3 15 7 16 4

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
				13 6 10 10 !TAB> linreg = Y0, ANSTIEG 19.73214 -0.98214
lists	1	TT11	Liste von Listen	X1:= 1 2 lists 2 ergibt im tabh-Format X1 1 1 1 1 2 2 1 2 2
ln	1	TT1	natürlicher Logarithmus	e ln = 1.
log	2	TT1	allgemeiner Logarithmus	100 log 10 = 2.
lower	1	TT1	zu Kleinbuchstaben	AsdRRGee34 lower = asdrree34
mal	2	TT11	Vervielfachung	Auto mal 3 = Auto Auto Auto oder xx.tab mal 3
max	1	Zahl	Maximalzahl	12.21,2,Hallo max ergibt 12.21
median	1	Zahl	mittlere Zahl	1 2 4,9.9 median ergibt 3.0
meta	1+Name	Modifikation von TT1	Daten in Metadaten überführen	<TAB! FACH,NOTE m Deu 1 Phy 2 Ma 1 !TAB> meta FACH ergibt DEU, MA, PHY 1 1 2
min	1	Zahl	Minimalzahl	12.21,2,Hallo min ergibt 2
minus	1	TT1	unäres Minus	4+8 minus

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
				ergibt -12
next	-		vor der zweiten Formel	X:=first 100 next X pred *1.03 at Y
nextonr	-		next für onr-Rekursion	X:=firstonr 100 nextonr X pred *1.03 at Y
no	0	BOOL	Boolesche Konstante: Wahrheitswert falsch; entspricht Antwort no (spanisch)	no si = si
not	1	TT1	Negation	si not = no
nth	2	TT1 ohne m	n-te Komponente	1 3 5 nth 2 = 3
onr	1	TT1	Konvertierung	1 3 5.2 "4.5.5" onr ergibt in tabh: 1 3 5.2 4.5.5
onrs	1+Name +1	TT1 um Name erweitert	otto-Nummern in Tabelle generieren	<TAB! X, Ym m k y z y w !TAB> onrs OTTO!k ergibt X, (OTTO, Y m) l k 1 z 2 y 2.1 w
pi	0	PZAHL	Kreiskonstante	KREISFLAECHE:=R*R*pi
poly polynom	2	TT1	Polynom	3 poly [1 2 3] ergibt 18
pos	Name	ZAHL	Position	avec X pos < 10
pos-	Name	ZAHL	Position von hinten	avec X pos- > 5
pred	Name	Name	Vorgänger	X:= first 100 next X pred *1.03
pred_n	Name+1	Name	n-ter Vorgänger	X pred_n 3
pzahl	1	TT1 mit PZAHL	Konvertierung	1/5 6 9.7 pzahl ergibt als tabh-Format 0.2 6. 9.7
pzahl1de	1	TT1 mit PZAHL	Extraktion	"Heute bekomme ich 356,88 Euro und nicht 66,8 ." pzahl1de ergibt 356.88
rat	2	RATIO	Ratios aus ZAHLen	<TAB!

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
			erzeugen	X, Y 1 1 1 2 3 !TAB> Z:= X rat Y ergibt X, (Y, Z 1) 1 1 2 1/2 3 1/3
ratio	1	TT1 mit RATIO	Konvertierung	1/5 6 9.7 ratio ergibt im tabh-Format 1/5 6/1 97/10
rename	1+Name +Name	TT1 bis auf N2	Spaltennamenumbe- nennung	rename X!Y
rest	2	TT1	Rest bei ganzzahligen Division	13 rest 5 ergibt 3
rnd	2	TT1	runden	17.678 3.45 zz 8 rnd 1 ergibt 17.7 3.5 zz 8
rndwort	2	TT1, aber PZAHL wird in WORT überführt; fehlende Nullen werden aufgefüllt	runden	<TAB! X, Y m 0 0.1 1 1.342 0 1. !TAB> rnd2 ergibt X, Y m 0 0.10 1 1.34 0 1.00
route	1	TT1	Streckenfolge füllen	<TAB! X, Y m 0 0 1 1 0 1 !TAB> route zeichnet die 2 Strecken von (0,0) bis (1,1) und (1,1) bis (0,1)
sans	1+Bed	TT1	Selektion (ohne)	sans ORT=Magdeburg sans Magdeburg sans: ohne die spezifizierten (komplexen) Tupel
satzl	1	SATZ! SATZ! TEXT	Liste aller Sätze	"Es ist prima. Toll. Morgen feiern wir." satzl

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
				ergibt im Tabh-Format: SATZ1 Es ist prima. Toll. Morgen feiern wir.
seg	Name	elementares Tupel	Segment	enkel.tabh avec Oehna in NAME seg oder X seg ++: Durchschnitt aller Zahlen des Segments, das X enthält
si	0	BOOL	Wahrheitswert wahr (entspricht der Antwort ja)	si & no = no
sin	1	TT1	Sinusfunktion	3.14159 sin =2.65358979335e-06
split	2	S11	aufsplitten von Text	ORTl:= "Brati,Novi Sad, Belgrad" split "," Ergebnis (ment): <TABM> <ORT>Brati</ORT> <ORT>Novi Sad</ORT> <ORT>Belgrad</ORT> </TABM>
sqrt	1	TT1	Quadratwurzel	4 sqrt ergibt 2.
streu streuung	1	PZAHL	Streuung	[1 2 5 3 5 1] streu ergibt 1.5
subtext	3	TT1	Teiltext	aBCdE subtext 2 ! 3 = BCd
subtext2	3	TT1	Teiltext	aBCdEfgH subtext2 "B" ! fg =CdE
succ	Name		Nachfolger	X succ
succ_n	Name+1		n-ter Nachfolger	X succ_n 3
tag0	1+Name	Name	äußeren Tag setzen	11 13 tag0 XX ergibt als ment-Struktur <XX> 11 13 </XX>
tags	1+ Name	Name m	Jedem Element einer Kollektion einen Namen	{ 1 3 } tags XY ergibt (ment)

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
			geben	<TABM> <XY>1</XY> <XY>3</XY> </TABM>
tagtup	1+ NameTup p	NameTup	Jeder Komponente eines Tupels einen Namen geben	1, 4 tagtup X, Y ergibt (ment) <TABM> <X>1</X> <Y>4</Y> </TABM>
tan	1	TT1	Tangensfunktion	3.14 tan ergibt -0.00159265493641
text	1	TT1	Konvertierung	3.14 ttt 8 text ergibt TEXT1 3.14 ttt 8
textend	2	TT1	Teiltext	asdfgh textend 4 =fgh Rest des Textes ab der spezifizierten Position
textend-	2	TT1	Teiltext	asdfgh textend- 4 =dfgh Rest des Textes ab der spezifizierten Position von hinten gezählt
textindex	2	ZAHL	Position	"Heute ist Dienstag." textindex Di ergibt ZAHL 11
time	0	PZAHL	Systemzeit	time kann ergeben PZAHL 1.557021
trim	1		Leerzeichen hinten und vorne entfernen	" Hi o++o " trim ergibt im MENT-format <TABM>Hi o++o</TABM>
tup	Name	gesamtes Tupel	gesamtes Tupel	enkel.tabh avec Deu in NAME tup
untag0	1		äußeren Tag entfernen	X:=1 untag0 ergibt ZAHL 1
upper	1	TT1	zu Großbuchstaben	1.2, aW upper

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
				ergibt als tab PZAHL, WORT 1.2 AW jeder Kleinbuchstabe wird in einen Großbuchstaben umgewandelt; der Rest verbleibt unverändert.
variance	1	PZAHL	Varianz	[1 2 4 6] variance = 4.91666666667
verti	1+sche me+sch eme	TT1-S2+S1	Daten vertikal anordnen	verti MON,STRAH 1:= JAN .. DEZ verti FACH,NOTE1 1:= PHY1 ..MAL
vlists	1	TT1 l	Liste von Listen	variabel lange Listen; die Operation stimmt mit lists überein, nur dass alle kürzeren Listen noch im Ergebnis eingeschlossen sind.
weg	1+Na- men	TT1 ohne die Namen	Spalten eliminieren	<TABH! X, Ym m 1 2 3 4 5 !TABH> weg Y ergibt im tab-Format Xm 1 4
wort	1	TT1 mit WORT	Konvertierung	"Ich bin gut.Du auch." wort ergibt WORT Ich_bin_gut.Du_auch.
wortb	1	WORTb	Bag aller Worte	"Ich bin. Ich auch." wortb ergibt im tabh-Format WORTb auch bin Ich Ich
wortm	1	WORTm	Menge aller Worte	"We are 6." wortm ={6 are we}
wortl	1	WORTl	Liste aller Worte	"We are 6." wortl ergibt als tabh-Format WORTl We are 6

Operation	Stelligkeit	Output-TT	Bedeutung	Beispiele
zahl	1	TT1 mit ZAHL konvertiert PZAHN und geeignete Text'e in ZAHL	Konvertierung	"12" zahl ergibt 12 3.14 zahl ergibt 3
zahl1	1	TT1 mit ZAHL	erste Zahl im Text	"24:5:33" zahl1 =24
zahl1de	1	TT1 mit ZAHL	erste deutsche Zahl im Text	"Heute bekomme ich 66.356,88 Euro" zahl1de ergibt 66356
zahl2	1	TT1 mit ZAHL	zweite Zahl im Text	"24.05" zahl2 =5
zahl3	1	TT1 mit ZAHL	dritte Zahl im Text	"24:AA:5::087" zahl3 =87

Abschließend sollen noch die Operationen für Schemen angegeben werden. Sie werden beispielsweise in gib und igib-Anweisungen benutzt.

Operation	Stelligkeit	Bedeutung	Beispiele
,	2	Komma; Paarbildung auf Schemaebene	NAME,ORT NAME,HOBBYl
b	1	Schema für Multimengen	NAMEb
m	1	Schema für Mengen	ORTm NAME,ORT m
l	1	Schema für Listen	NOTEl
b-	1	Schema um Multimengen abwärts zu sortieren	NAMEb-
m-	1	Schema um Mengen abwärts zu sortieren	GEHALT,NAME,ORT m-
l-	1	Schema um die Reihenfolge umzukehren	NOTEl-
	2	Alternative	NOTE KLAUSUR l NAME,(NOTE KLAUSUR l) m