

# TI-Nspire™ CX CAS Referenshandbok

## ***Viktigt information***

Texas Instruments lämnar inga garantier, vare sig uttryckliga eller underförstådda, inklusive men inte begränsade till underförstådda garantier i fråga om säljbarhet eller lämplighet för ett speciellt syfte, rörande program eller bokmaterial och gör endast sådant material tillgängligt på en "i befintligt skick"-grund. Under inga omständigheter kommer Texas Instruments att vara skyldigt för speciella, omgivande, oavsiktliga eller följdaktiga skador i samband med eller uppkomna genom köpet eller användandet av dessa material och det enda och exklusiva åtagande som tillfaller Texas Instruments, oavsett formen av åtgärd, skall inte överstiga summan som anges i programmets licens. Texas Instruments är inte heller skadeståndsskyldigt för några krav som uppstått genom användning av denna produkt av annan användare.

© 2026 Texas Instruments Incorporated

De faktiska produkterna kan variera något från de visade bilderna.

# Innehåll

|                                                |            |
|------------------------------------------------|------------|
| <b>Mallar för uttryck</b>                      | <b>1</b>   |
| <b>Alfabetisk lista</b>                        | <b>8</b>   |
| A                                              | 8          |
| B                                              | 17         |
| C                                              | 21         |
| D                                              | 46         |
| E                                              | 59         |
| F                                              | 70         |
| G                                              | 80         |
| I                                              | 90         |
| L                                              | 99         |
| M                                              | 116        |
| N                                              | 124        |
| O                                              | 133        |
| P                                              | 136        |
| Q                                              | 145        |
| R                                              | 148        |
| S                                              | 163        |
| T                                              | 189        |
| U                                              | 205        |
| V                                              | 206        |
| W                                              | 207        |
| X                                              | 209        |
| Z                                              | 210        |
| <b>Symboler</b>                                | <b>219</b> |
| <b>TI-Nspire™ CX II-Kommandon för att rita</b> | <b>246</b> |
| Grafikprogrammering                            | 246        |
| Grafikskärm                                    | 246        |
| Standardvy och inställningar                   | 247        |
| Felmeddelanden på grafikskärmen                | 248        |
| Ogiltiga kommandon i grafikläge                | 248        |
| C                                              | 249        |
| D                                              | 250        |
| F                                              | 253        |
| G                                              | 255        |
| P                                              | 256        |
| S                                              | 258        |
| U                                              | 260        |

|                                                                  |            |
|------------------------------------------------------------------|------------|
| <b>Tomma element</b> .....                                       | <b>261</b> |
| <b>Kortkommandon för att mata in matematiska uttryck</b> .....   | <b>263</b> |
| <b>EOS™-hierarki (Equation Operating System)</b> .....           | <b>265</b> |
| <b>TI-Nspire CX II - TI-Basic programmeringsfunktioner</b> ..... | <b>267</b> |
| Autoindentering i Programeditor .....                            | 267        |
| Förbättrade felmeddelanden för TI-Basic .....                    | 267        |
| <b>Konstanter och värden</b> .....                               | <b>270</b> |
| <b>Felkoder och meddelanden</b> .....                            | <b>271</b> |
| <b>Varningskoder och meddelanden</b> .....                       | <b>280</b> |
| <b>Allmän information</b> .....                                  | <b>282</b> |
| <b>Index</b> .....                                               | <b>283</b> |

# Mallar för uttryck

Mallar för uttryck erbjuder ett enkelt sätt att mata skriva in uttryck med matematiska standardtecken. När du matar in en mall visas den på inmatningsraden med små block i positioner där du kan skriva in element. En markör visar vilket element du kan skriva in.

Använd piltangenterna eller tryck på **tab** för att flytta till varje elements position, och skriv ett värde eller uttryck för det aktuella elementet. Tryck på **enter** eller **ctrl enter** för att utvärdera uttrycket.

## Mall för Bråk

**ctrl** **÷** **tangenter**



**Obs:** Se även / (dela), på sidan 221.

Exempel:

$$\frac{12}{8 \cdot 2} \quad \frac{3}{4}$$

## Mall för Exponent

**^** **tangent**



**Obs:** Skriv in det första värdet, tryck på **^** och skriv sedan in exponenten. För att återföra markören till basraden, tryck på högerpilen (►).

**Obs:** Se även ^ (potens), på sidan 222.

Exempel:

$$2^3 \quad 8$$

## Mall för Kvadratrot

**ctrl** **x<sup>2</sup>** **tangenter**



**Obs:** Se även  $\sqrt{()}$  (kvadratrot), på sidan 232.

Exempel:

$$\sqrt{4} \quad 2$$
$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{\{a\}}, 2\}$$

## Mall för N:te rot

**ctrl** **^** **tangenter**



**Obs:** Se även **root()**, på sidan 159.

Exempel:

## Mall för N:te rot

**ctrl** **^** **tangenter**

$$\sqrt[n]{8} \quad 2$$

$$\sqrt[3]{\{8, 27, b\}} \quad \left\{ 2, 3, b^{\frac{1}{3}} \right\}$$

## e exponent mall

**e<sup>x</sup>** **tangent**

**e** **()**

Basen för den naturliga logaritmen  $e$  upphöjd till

**Obs:** Se även **e<sup>^</sup>()**, på sidan 59.

Exempel:

$$e^1 \quad e$$

$$e^1. \quad 2.71828182846$$

## Mall för Log

**ctrl** **10<sup>x</sup>** **tangenter**

**log** **()**

Beräknar logaritmen till en specificerad bas. För en förinställning av bas 10, utelämna basen.

**Obs:** Se även **log()**, på sidan 111.

Exempel:

$$\log_4(2.) \quad 0.5$$

## Stegvis mall (2 steg)

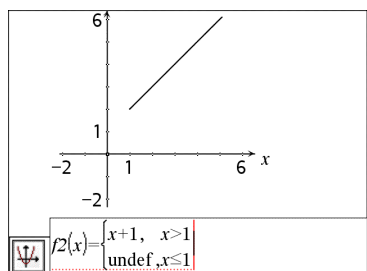
**Katalog >** **tab**

**{** **0.** **0.**  
**{** **0.** **0.**

Låter dig skapa uttryck och villkor för en stegvis funktion med två steg.- För att lägga till ett steg, klicka i mallen och upprepa mallen.

**Obs:** Se även **stegvis()**, på sidan 137.

Exempel:



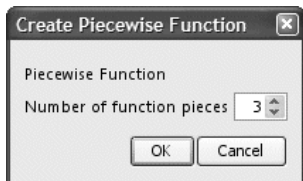
## Stegvis mall (N steg)

Katalog > 

Låter dig skapa uttryck och villkor för en stegvis funktion med N steg.- Promptar för  $N$ .

Exempel:

Se exemplet på Stegvis mall (2 steg).



**Obs:** Se även `stegvis()`, på sidan 137.

## Mall för System med 2 ekvationer

Katalog > 



Skapar ett ekvationssystem med två ekvationer. För att lägga till en rad i ett befintligt system, klicka i mallen och upprepa mallen.

**Obs:** Se även `system()`, på sidan 189.

Exempel:

$$\begin{array}{l} \text{solve} \left( \begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) \quad x=\frac{5}{2} \text{ and } y=\frac{-5}{2} \\ \text{solve} \left( \begin{cases} y=x^2-2 \\ x+2 \cdot y=-1 \end{cases}, x, y \right) \\ x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1 \end{array}$$

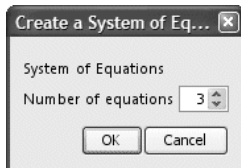
## Mall för System med N ekvationer

Katalog > 

Låter dig skapa ett ekvationssystem med  $N$  ekvationer. Promptar för  $N$ .

Exempel:

Se exemplet på mall för Ekvationssystem (2 ekvationer).



**Obs:** Se även `system()`, på sidan 189.

## Mall för Absolutbelopp

Katalog > 



**Obs:** Se även `abs()`, på sidan 8.

Exempel:

## Mall för Absolutbelopp

Katalog >



$$\left\{ 2, -3, 4, -4^3 \right\}$$

$$\{ 2, 3, 4, 64 \}$$

## Mall för dd°mm'ss.ss''

Katalog >



° ' "

Exempel:

Låter dig skriva in vinklar i formatet **dd°mm'ss.ss''**, där **dd** är antalet decimala grader, **mm** är antalet minuter och **ss.ss** är antalet sekunder.

30°15'10"

$10891 \cdot \pi$

64800

## Matrismall (2 x 2)

Katalog >



$\begin{bmatrix} & \\ & \end{bmatrix}$

Exempel:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot a$$

$$\begin{bmatrix} a & 2 \cdot a \\ 3 \cdot a & 4 \cdot a \end{bmatrix}$$

Skapar en 2 x 2-matris.

## Matrismall (1 x 2)

Katalog >



$\begin{bmatrix} & \end{bmatrix}$

Exempel:

$$\text{crossP}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 4 \end{bmatrix}\right)$$

$$\begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$$

## Matrismall (2 x 1)

Katalog >



$\begin{bmatrix} \\ \\ \end{bmatrix}$

Exempel:

$$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01$$

$$\begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

## Matrismall (m x n)

Katalog >



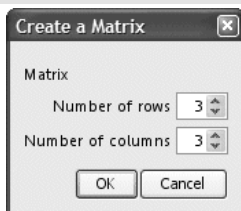
Mallen visas när du har uppmanats att specificera antalet rader och kolumner.

Exempel:

$$\text{diag} \left( \begin{bmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix} \right)$$

$$\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$$





**Obs:** Om du skapar en matris med många rader och kolumner kan det ta några sekunder innan den visas.

Mall för Summa ( $\Sigma$ )Katalog > 

$$\sum_{i=0}^{} (i)$$

Exempel:

$$\sum_{n=3}^7 (n) \quad 25$$

**Obs:** Se även  $\Sigma()$  (**sumSeq**), på sidan 233.

Mall för Produkt ( $\Pi$ )Katalog > 

$$\prod_{i=0}^{} (i)$$

Exempel:

$$\prod_{n=1}^5 \left(\frac{1}{n}\right) \quad \frac{1}{120}$$

**Obs:** Se även  $\Pi()$  (**prodSeq**), på sidan 233.

## Mall för förstaderivata

Katalog > 

$$\frac{d}{dx} (i)$$

Exempel:

$$\frac{d}{dx} (x^3) \quad 3 \cdot x^2$$

$$\frac{d}{dx} (x^3) |_{x=3} \quad 27$$

Mallen för förstaderivata kan också användas för att beräkna förstaderivatan i en punkt.

**Obs:** Se även **d()** (**derivata**), på sidan 230.

## Andraderivata, mall

Katalog > 

$$\frac{d^2}{dx^2}(\square)$$

Mallen för andraderivata kan också användas för att beräkna andraderivatan i en punkt.

**Obs:** Se även **d()** (derivata), på sidan 230.

Exempel:

$$\frac{d^2}{dx^2}(x^3) \quad 6 \cdot x$$

$$\frac{d^2}{dx^2}(x^3)|_{x=3} \quad 18$$

## Mall för N:te derivata

Katalog > 

$$\frac{d^N}{dx^N}(\square)$$

**Obs:** Se även **d()** (derivata), på sidan 230.

Exempel:

$$\frac{d^3}{dx^3}(x^3)|_{x=3} \quad 6$$

## Mall för Bestämd integral

Katalog > 

$$\int_a^b \square dx$$

**Obs:** Se även **f()** **integral()**, på sidan 231.

Exempel:

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

## Mall för obestämd integral

Katalog > 

$$\int \square dx$$

**Obs:** Se även **f()** **integral()**, på sidan 231.

Exempel:

$$\int x^2 dx \quad \frac{x^3}{3}$$

## Mall för Gränsvärde

Katalog > 

$$\lim_{x \rightarrow \square} (\square)$$

Använd - eller (-) för det vänstra gränsvärdet. Använd + för det högra gränsvärdet.

Exempel:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) \quad 13$$

**Obs:** Se även `gränsvärde()`, på sidan 101.

# Alfabetisk lista

Poster som inte är alfabetiska (till exempel, +, ! och >) listas i slutet av detta avsnitt och börjar, här. Om inget annat anges har alla exempel i detta avsnitt utförts i det förinställda återställningsläget och alla variabler betraktas som odefinierade.

## A

abs()

Katalog >

abs(Expr I)⇒uttryck

abs(List I)⇒lista

abs(Matrix I)⇒matris

Ger argumentets absolutbelopp.

Obs: Se även Mall för Absolutbelopp, på sidan 3.

Om argumentet är ett komplext tal erhålls talets modul.

Obs: Alla odefinierade variabler behandlas som reella variabler.

$$\left| \left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\} \right|$$
$$|2-3 \cdot i|$$
$$|z|$$
$$|x+y \cdot i|$$

$$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$$
$$\sqrt{13}$$
$$|z|$$
$$\sqrt{x^2+y^2}$$

amortTbl()

Katalog >

amortTbl(NPmt,N,I,PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [roundValue])⇒matris

Amorteringsfunktion som ger en matris i form av en amorteringstabell för en uppsättning av TVM-argument.

NPmt är antalet inbetalningar som skall inkluderas i tabellen. Tabellen börjar med den första inbetalningen.

N, I, PV, Pmt, FV, PpY, CpY och PmtAt beskrivs i tabellen över TVM-argument, se på sidan 203.

- Om du utelämnar Pmt används förinställningen Pmt=tvmPmt (N,I,PV,FV,PpY,CpY,PmtAt).
- Om du utelämnar FV används förinställningen FV=0.
- Förinställningarna av PpY, CpY och

amortTbl(12,60,10,5000,,,12,12)

|    |        |        |         |
|----|--------|--------|---------|
| 0  | 0.     | 0.     | 5000.   |
| 1  | -41.67 | -64.57 | 4935.43 |
| 2  | -41.13 | -65.11 | 4870.32 |
| 3  | -40.59 | -65.65 | 4804.67 |
| 4  | -40.04 | -66.2  | 4738.47 |
| 5  | -39.49 | -66.75 | 4671.72 |
| 6  | -38.93 | -67.31 | 4604.41 |
| 7  | -38.37 | -67.87 | 4536.54 |
| 8  | -37.8  | -68.44 | 4468.1  |
| 9  | -37.23 | -69.01 | 4399.09 |
| 10 | -36.66 | -69.58 | 4329.51 |
| 11 | -36.08 | -70.16 | 4259.35 |
| 12 | -35.49 | -70.75 | 4188.6  |

8 Alfabetisk lista

*PmtAt* är desamma som för TVM-funktionerna.

*roundValue* anger antalet decimaler för avrundning. Förinställning: 2.

Kolumnerna i resultatmatrisen har följande ordning: Inbetalningsnummer, räntebelopp, kapitalbelopp och balans.

Balansen som visas på rad *n* är balansen efter inbetalning *n*.

Du kan använda resultatmatrisen som indata för de andra amorteringsfunktionerna  $\Sigma\text{Int}()$  och  $\Sigma\text{Prn}()$ , se på sidan 234, och **bal()**, se på sidan 17.

## and (och)

*BooleanExpr1* **and**

*BooleanExpr2*  $\Rightarrow$  Booleskt uttryck

|                                                        |                           |
|--------------------------------------------------------|---------------------------|
| $x \geq 3$ and $x \geq 4$                              | $x \geq 4$                |
| $\{x \geq 3, x \leq 0\}$ and $\{x \geq 4, x \leq -2\}$ | $\{x \geq 4, x \leq -2\}$ |

*BooleanList1* **and**

*BooleanList2*  $\Rightarrow$  Boolesk lista

*BooleanMatrix1* **and**

*BooleanMatrix2*  $\Rightarrow$  Boolesk matris

Ger resultatet sant eller falskt eller en förenklad form av den ursprungliga inmatningen.

*Integer1* **and** *Integer2*  $\Rightarrow$  heltal

I hexadecimalt basläge:

Jämför två reella heltal bit för bit med en **och**-operation. Internt omvandlas båda heltalen till 64-bitars binära tal. När motsvarande bitar jämförs blir resultatet 1 om båda bitarna är 1, annars blir resultatet 0. Det erhållna värdet representerar bitresultatet och visas enligt Bas-läget.

|                    |        |
|--------------------|--------|
| 0h7AC36 and 0h3D5F | 0h2C16 |
|--------------------|--------|

Viktigt: Noll, inte bokstaven O.

I binärt basläge:

|                    |       |
|--------------------|-------|
| 0b100101 and 0b100 | 0b100 |
|--------------------|-------|

Du kan skriva in heltalen i valfri talbas. För en binär eller hexadecimal inmatning måste du använda prefixet 0b respektive 0h. Utan prefix behandlas heltalen som decimala (bas 10).

I decimalt basläge:

|              |   |
|--------------|---|
| 37 and 0b100 | 4 |
|--------------|---|

Om du skriver in ett decimalt heltal som är alltför stort för att anges i 64-bitars binär form används en symmetrisk moduloberäkning för att få ned värdet till lämplig nivå.

**Obs:** En binär inmatning kan ha upp till 64 siffror (exklusive prefixet 0b). En hexadecimal inmatning kan ha upp till 16 siffror.

**angle()**

**angle**(*Expr1*) $\Rightarrow$ *uttryck*

Ger argumentets vinkel med argumentet tolkat som ett komplext tal.

**Obs:** Alla odefinierade variabler behandlas som reella variabler.

I vinkelläget Grader:

$$\text{angle}(0+2\cdot i) \quad 90$$

I vinkelläget Nygrader:

$$\text{angle}(0+3\cdot i) \quad 100$$

I vinkelläget Radianer:

$$\text{angle}(1+i) \quad \frac{\pi}{4}$$

$$\text{angle}(z) \quad \frac{-\pi \cdot (\text{sign}(z)-1)}{2}$$

$$\text{angle}(x+i\cdot y) \quad \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$$

$$\text{angle}\left(\left\{1+2\cdot i, 3+0\cdot i, 0-4\cdot i\right\}\right) \quad \left\{\frac{\pi}{2} - \tan^{-1}\left(\frac{1}{2}\right), 0, -\frac{\pi}{2}\right\}$$

**vinkel**(*List1*) $\Rightarrow$ *lista*

**vinkel**(*Matrix1*) $\Rightarrow$ *matris*

Ger en lista eller matris över vinklarna hos elementen i *List1* eller *Matrix1*, där varje element tolkas som ett komplext tal som representerar en tvådimensionell rektangulär koordinatpunkt.

**ANOVA**

**ANOVA** *List1, List2[, List3, ..., List20][, Flag]*

Utför en 1-vägs variansanalys för att jämföra medelvärdena hos 2 till 20 populationer. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

*Flag*=0 för Data, *Flag*=1 för Statistik

| Resultatvariabel | Beskrivning                                                   |
|------------------|---------------------------------------------------------------|
| stat.F           | Värdet på F-statistiken                                       |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas |
| stat.df          | Frihetsgrader hos grupperna                                   |
| stat.SS          | Kvadratsumma hos grupperna                                    |
| stat.MS          | Kvadratmedelvärde hos grupperna                               |
| stat.dfError     | Frihetsgrader hos felen                                       |
| stat.SSError     | Kvadratsumma hos felen                                        |
| stat.MSError     | Kvadratmedelvärde hos felen                                   |
| stat.sp          | Sammanslagen (pooled) standardavvikelse                       |
| stat.xbarlist    | Medelvärdet på listornas indata                               |
| stat.CLowerList  | 95 % konfidensintervall för medelvärdet hos varje indatalista |
| stat.CUpperList  | 95 % konfidensintervall för medelvärdet hos varje indatalista |

## ANOVA2way

**ANOVA 2-vägs***Lista1,Lista2*  
*[,Lista3,...,Lista10][,LevRow]*

Beräknar en 2-vägs variansanalys för att jämföra medelvärdena hos 2 till 10 populationer. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

*NivRad*=0 för Block

*NivRad*=2,3,...,*Län*-1, för Två Faktorer, där  
*Län*=längd(*Lista1*)=längd(*Lista2*) = ... =  
 längd(*Lista10*) och *Län* / *NivRad* ∈ {2,3,...}

Utdata: Block Design

| Resultatvariabel | Beskrivning                                                   |
|------------------|---------------------------------------------------------------|
| stat.F           | F statistik för kolumnfaktorn                                 |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas |
| stat.df          | Frihetsgrader hos kolumnfaktorn                               |
| stat.SS          | Kvadratsumma hos kolumnfaktorn                                |
| stat.MS          | Kvadratmedelvärde hos kolumnfaktorn                           |
| Statistik.FBlock | F statistik för faktor                                        |
| stat.PValBlock   | Lägsta sannolikhet vid vilken nollhypotesen kan förkastas     |
| stat.dfBlock     | Frihetsgrader hos faktor                                      |
| stat.SSBlock     | Kvadratsumma hos faktor                                       |
| stat.MSBlock     | Kvadratmedelvärde hos faktor                                  |
| stat.dfError     | Frihetsgrader hos felen                                       |
| stat.SSError     | Kvadratsumma hos felen                                        |
| stat.MSError     | Kvadratmedelvärde hos felen                                   |
| stat.s           | Standardavvikelse hos felet                                   |

#### Utdata för KOLUMNFAKTOR

| Resultatvariabel | Beskrivning                         |
|------------------|-------------------------------------|
| stat.Fcol        | F statistik för kolumnfaktorn       |
| stat.PValCol     | Sannolikhetsvärde på kolumnfaktorn  |
| stat.dfCol       | Frihetsgrader hos kolumnfaktorn     |
| stat.SSCol       | Kvadratsumma hos kolumnfaktorn      |
| stat.MSCol       | Kvadratmedelvärde hos kolumnfaktorn |

#### Utdata för RADFAKTOR

| Resultatvariabel | Beskrivning                      |
|------------------|----------------------------------|
| stat.FRow        | F statistik för radfaktorn       |
| stat.PValRow     | Sannolikhetsvärde på radfaktorn  |
| stat.dfRow       | Frihetsgrader hos radfaktorn     |
| stat.SSRow       | Kvadratsumma hos radfaktorn      |
| stat.MSRow       | Kvadratmedelvärde hos radfaktorn |



Utdata för INTERAKTION

| Resultatvariabel  | Beskrivning                         |
|-------------------|-------------------------------------|
| stat.FInteract    | F statistik för interaktionen       |
| stat.PValInteract | Sannolikhetsvärde på interaktionen  |
| stat.dfInteract   | Frihetsgrader hos interaktionen     |
| stat.SSInteract   | Kvadratsumma hos interaktionen      |
| stat.MSInteract   | Kvadratmedelvärde hos interaktionen |

Utdata för FEL

| Resultatvariabel | Beskrivning                 |
|------------------|-----------------------------|
| stat.dfError     | Frihetsgrader hos felen     |
| stat.SSError     | Kvadratsumma hos felen      |
| stat.MSError     | Kvadratmedelvärde hos felen |
| s                | Standardavvikelse hos felet |

| Ans (svar)                                        | <div>ctrl</div> | <div>(-)</div> | tangenter |
|---------------------------------------------------|-----------------|----------------|-----------|
| Ans⇒värde                                         | 56              |                | 56        |
| Ger resultatet på det senast beräknade uttrycket. | 56+4            |                | 60        |
|                                                   | 60+4            |                | 64        |

| approx()                                                                                                                                                                          | Katalog >       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <b>approx(Expr1)⇒uttryck</b>                                                                                                                                                      | <div>approx(<math>\frac{1}{3}</math>)0.333333</div>                                              |
| Visar resultatet av beräkningen av argumentet som ett uttryck med decimala värden, när så är möjligt, oavsett den aktuella inställningen av <b>Auto</b> eller <b>Ungefärlig</b> . | <div>approx(<math>\{\frac{1}{3}, \frac{1}{9}\}</math>)<math>\{0.333333, 0.111111\}</math></div>  |
| Detta motsvarar att skriva in argumentet och trycka på <div>ctrl</div> <div>enter</div> .                                                                                         | <div>approx(<math>\{\sin(\pi), \cos(\pi)\}</math>)<math>\{0, -1\}</math></div>                   |
|                                                                                                                                                                                   | <div>approx(<math>[\sqrt{2}, \sqrt{3}]</math>)<math>[1.41421 \quad 1.73205]</math></div>         |
|                                                                                                                                                                                   | <div>approx(<math>[\frac{1}{3}, \frac{1}{9}]</math>)<math>[0.333333 \quad 0.111111]</math></div> |
| <b>approx(List1)⇒lista</b>                                                                                                                                                        | <div>approx(<math>\{\sin(\pi), \cos(\pi)\}</math>)<math>\{0, -1\}</math></div>                   |
| <b>approx(Matrix1)⇒matris</b>                                                                                                                                                     | <div>approx(<math>[\sqrt{2}, \sqrt{3}]</math>)<math>[1.41421 \quad 1.73205]</math></div>         |

## approx()

Katalog > 

Ger en lista eller *matrix* där varje element har beräknats till ett decimalt värde, när så är möjligt.

## ►approxFraction()

Katalog > 

*Expr* ►**approxFraction**([*Tol*])⇒*uttryck*

*List* ►**approxFraction**([*Tol*])⇒*lista*

*Matrix* ►**approxFraction**([*Tol*])⇒*matrix*

Ger indata som ett bråk med hjälp av toleransen hos *Tol*. Om *Tol* utelämnas används en tolerans på 5.E-14.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva @>**approxFraction**(...).

|                                                                        |                                                         |
|------------------------------------------------------------------------|---------------------------------------------------------|
| $\frac{1}{2} + \frac{1}{3} + \tan(\pi)$                                | 0.833333                                                |
| <hr/>                                                                  |                                                         |
| $0.8333333333333333 \blacktriangleright \text{approxFraction}(5.E-14)$ | $\frac{5}{6}$                                           |
| <hr/>                                                                  |                                                         |
| $\{\pi, 1.5\} \blacktriangleright \text{approxFraction}(5.E-14)$       | $\left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$ |

## approxRational()

Katalog > 

**approxRational**(*Expr*[, *Tol*])⇒*uttryck*

**approxRational**(*List*[, *Tol*])⇒*lista*

**approxRational**(*Matrix*[, *Tol*])⇒*matrix*

Ger argumentet som ett bråk med hjälp av toleransen hos *Tol*. Om *Tol* utelämnas används en tolerans på 5.E-14.

|                                                       |                                                              |
|-------------------------------------------------------|--------------------------------------------------------------|
| $\text{approxRational}(0.333, 5 \cdot 10^{-5})$       | $\frac{333}{1000}$                                           |
| <hr/>                                                 |                                                              |
| $\text{approxRational}(\{0.2, 0.33, 4.125\}, 5.E-14)$ | $\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$ |

## arccos()

Se  $\cos^{-1}()$ , på sidan 32.

## arccosh()

Se  $\cosh^{-1}()$ , på sidan 33.

## arccot()

Se  $\cot^{-1}()$ , på sidan 34.

**arccoth()**Se  $\coth^{-1}()$ , på sidan 35.**arccsc()**Se  $\csc^{-1}()$ , på sidan 38.**arccsch()**Se  $\operatorname{csch}^{-1}()$ , på sidan 38.**arcLen()**Katalog > **arcLen(Expr1,Var,Start,End)**  $\Rightarrow$  uttryckGer båglängden hos *Expr1* från *Start* till *End* med hänsyn till variabeln *Var*.

Båglängden beräknas som en integral baserat på ett definierat funktionsläge.

**arcLen(List1,Var,Start,End)**  $\Rightarrow$  listaGer en lista på båglängden hos varje element i *List1* från *Start* till *End* med hänsyn till *Var*. $\operatorname{arcLen}(\cos(x), x, 0, \pi)$  3.8202

$$\operatorname{arcLen}(f(x), x, a, b) = \int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$$

 $\operatorname{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi)$   
 $\{3.8202, 3.8202\}$ 
**arcsec()**Se  $\sec^{-1}()$ , på sidan 163.**arcsech()**Se  $\operatorname{sech}^{-1}()$ , på sidan 164.**arcsin()**Se  $\sin^{-1}()$ , på sidan 175.**arcsinh()**Se  $\sinh^{-1}()$ , på sidan 176.

## augment()

Katalog > **augment(List1, List2)** ⇒ lista

|                                     |                      |
|-------------------------------------|----------------------|
| augment( $\{1, -3, 2\}, \{5, 4\}$ ) | $\{1, -3, 2, 5, 4\}$ |
|-------------------------------------|----------------------|

Ger en ny lista med *List2* inlagd i slutet på *List1*.

**augment(Matrix1, Matrix2)** ⇒ matris

Ger en ny matris med *Matrix2* fogad till *Matrix1*. När kommatecknet (,) används måste matriserna ha samma raddimensioner och *Matrix2* fogas till *Matrix1* som nya kolumner. Ändrar inte *Matrix1* eller *Matrix2*.

|                                                               |                                                        |
|---------------------------------------------------------------|--------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$         |
| $\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$         | $\begin{bmatrix} 5 \\ 6 \end{bmatrix}$                 |
| augment( <i>m1</i> , <i>m2</i> )                              | $\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$ |

## avgRC()

Katalog > **avgRC(Expr1, Var [=Value] [, Step])** ⇒ uttryck

|                       |                         |
|-----------------------|-------------------------|
| avgRC( $f(x), x, h$ ) | $\frac{f(x+h)-f(x)}{h}$ |
|-----------------------|-------------------------|

**avgRC(Expr1, Var [=Value] [, List1])** ⇒ lista

|                                  |                               |
|----------------------------------|-------------------------------|
| avgRC( $\sin(x), x, h$ )   $x=2$ | $\frac{\sin(h+2)-\sin(2)}{h}$ |
|----------------------------------|-------------------------------|

**avgRC(List1, Var [=Value] [, Step])** ⇒ lista

|                           |                        |
|---------------------------|------------------------|
| avgRC( $x^2 - x + 2, x$ ) | $2 \cdot (x - 0.4995)$ |
|---------------------------|------------------------|

|                                |                      |
|--------------------------------|----------------------|
| avgRC( $x^2 - x + 2, x, 0.1$ ) | $2 \cdot (x - 0.45)$ |
|--------------------------------|----------------------|

**avgRC(Matrix1, Var [=Value] [, Step])** ⇒ matris

|                              |                   |
|------------------------------|-------------------|
| avgRC( $x^2 - x + 2, x, 3$ ) | $2 \cdot (x + 1)$ |
|------------------------------|-------------------|

Ger differenskvoten i positiv riktning.

*Expr1* kan vara ett användardefinierat funktionsnamn (se **Func**).

När *Värde* specificeras överstyr detta värde eventuella tidigare variabeltilldelningar eller aktuella ersättningar av typ "I" för variabeln.

*Step* är stegvärdet. Om *Step* utelämnas används förinställningen 0.001.

Observera att den liknande funktionen **centralDiff()** använder den symmetriska differenskvoten.

## B

### bal()

**bal**(*NPmt*,*N*,*I*,*PV*,[*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*roundValue*])⇒värde

**bal**(*NPmt*,*amortTable*)⇒värde

Amorteringsfunktion som beräknar planerad balans efter en specificerad inbetalning.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* och *PmtAt* beskrivs i tabellen över TVM-argument, se här.

*NPmt* anger numret på den inbetalning efter vilken du vill att data skall beräknas.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* och *PmtAt* beskrivs i tabellen över TVM-argument, se här.

- Om du utelämnar *Pmt* används förinställningen *Pmt=tvmPmt* (*N*,*I*,*PV*,*FV*,*PpY*,*CpY*,*PmtAt*).
- Om du utelämnar *FV* används förinställningen *FV*=0.
- Förinställningarna av *PpY*, *CpY* och *PmtAt* är desamma som för TVM-funktionerna.

*roundValue* anger antalet decimaler för avrundning. Förinställning: 2.

**bal**(*NPmt*,*amortTable*) beräknar lånebalansen efter inbetalning nummer *NPmt*, baserat på amorteringstabell *amortTable*. Argumentet *amortTable* måste vara en matris i den form som beskrivs under **amortTbl()**, här.

**Obs:** Se även **ΣInt()** och **ΣPrn()**, här.

|                           |        |
|---------------------------|--------|
| bal(5,6,5.75,5000,,12,12) | 833.11 |
|---------------------------|--------|

|                                     |  |  |  |  |
|-------------------------------------|--|--|--|--|
| tbl:=amortTbl(6,6,5.75,5000,,12,12) |  |  |  |  |
|-------------------------------------|--|--|--|--|

|   |        |         |         |
|---|--------|---------|---------|
| 0 | 0.     | 0.      | 5000.   |
| 1 | -23.35 | -825.63 | 4174.37 |
| 2 | -19.49 | -829.49 | 3344.88 |
| 3 | -15.62 | -833.36 | 2511.52 |
| 4 | -11.73 | -837.25 | 1674.27 |
| 5 | -7.82  | -841.16 | 833.11  |
| 6 | -3.89  | -845.09 | -11.98  |

|            |         |
|------------|---------|
| bal(4,tbl) | 1674.27 |
|------------|---------|

*Integer1* ►Base2⇒*heltal*

256►Base2

0b100000000

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>Base2.

0h1F►Base2

0b11111

Omvandlar *Integer1* till ett binärt tal. Binära och hexadecimala tal har alltid prefixet 0b respektive 0h. Noll, inte bokstaven O, följt av b eller h.

0b *binärtTal*0h *hexadecimaltTal*

Ett binärt tal kan ha upp till 64 siffror. Ett hexadecimalt tal kan ha upp till 16 siffror.

Utan prefix behandlas *Integer1* som ett decimalt tal (bas 10). Resultatet visas i binär form, oavsett Bas-läget.

Negativa tal visas i "tvåkomplement"-form. Exempel,

-1 visas som 0hFFFFFFFFFFFFFF i Hexadecimalt basläge 0b111...111 (64 1's) i Binärt basläge

-2<sup>63</sup> visas som 0h8000000000000000 i Hexadecimalt basläge 0b100...000 (63 zeros) i Binärt basläge

Om du skriver in ett decimalt heltal som är alltför stort för att anges i 64-bitars binär form används en symmetrisk modulooperation för att få ned värdet till lämplig nivå. Se följande exempel på värden utanför området.

2<sup>63</sup> blir -2<sup>63</sup> och visas som 0h8000000000000000 i Hexadecimalt basläge 0b100...000 (63 nollor) i Binärt basläge

2<sup>64</sup> blir 0 och visas som 0h0 i Hexadecimalt basläge 0b0 i Binärt basläge

$-2^63 - 1$  blir  $2^63 - 1$  och visas som  
 0h7FFFFFFFFFFFFFFF i Hexadecimalt  
 basläge 0b111...111 (64 ettor) i Binärt  
 basläge

*Integer1* ►Base10⇒*heltal*

|                |    |
|----------------|----|
| 0b10011►Base10 | 19 |
|----------------|----|

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>Base10.

|             |    |
|-------------|----|
| 0h1F►Base10 | 31 |
|-------------|----|

Omvandlar *Integer1* till ett decimalt tal (bas 10). En binär eller hexadecimal inmatning måste alltid ha prefixet 0b respektive 0h.

0b *binaryNumber*

0h *hexadecimalNumber*

Noll, inte bokstaven O, följt av b eller h.

Ett binärt tal kan ha upp till 64 siffror. Ett hexadecimalt tal kan ha upp till 16 siffror.

Utan prefix behandlas *Integer1* som ett decimalt tal. Resultatet visas i decimal form, oavsett Bas-läget.

*Integer1* ►Base16⇒*heltal*

|            |       |
|------------|-------|
| 256►Base16 | 0h100 |
|------------|-------|

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>Base16.

|                       |       |
|-----------------------|-------|
| 0b111100001111►Base16 | 0hF0F |
|-----------------------|-------|

Konverterar *Integer1* till ett hexadecimalt tal. Binära och hexadecimala tal har alltid prefixet 0b respektive 0h.

0b *binaryNumber*

0h *hexadecimalNumber*

Noll, inte bokstaven O, följt av b eller h.

Ett binärt tal kan ha upp till 64 siffror. Ett hexadecimalt tal kan ha upp till 16 siffror.

Utan prefix behandlas *Integer1* som ett decimalt tal (bas 10). Resultatet visas i hexadecimal form, oavsett Bas-läget.

Om du skriver in ett decimalt heltal som är alltför stort för att anges i 64-bitars binär form används en symmetrisk modulooperation för att få ned värdet till lämplig nivå. För mer information, se

►Base2, på sidan 18.

## binomCdf()

**binomCdf( $n, p$ )** ⇒ lista

**binomCdf( $n, p, lowBound, upBound$ )** ⇒ tal om *lowBound* och *upBound* är tal, lista om *lowBound* och *upBound* är listor

**binomCdf( $n, p, upBound$ )** för  $P(0 \leq X \leq upBound)$  ⇒ tal om *upBound* är ett tal, lista om *upBound* är en lista

Beräknar en kumulativ sannolikhet för den diskreta binomialfördelningen med  $n$  antal försök och sannolikheten  $p$  för att lyckas vid varje försök.

För  $P(X \leq upBound)$ , sätt *lowBound*=0

## binomPdf()

**binomPdf( $n, p$ )** ⇒ lista

**binomPdf( $n, p, XVal$ )** ⇒ tal om *XVal* är ett tal, lista om *XVal* är en lista

Beräknar en sannolikhet för den diskreta binomialfördelningen med  $n$  antal försök och sannolikheten  $p$  för att lyckas vid varje försök.



**ceiling()**Katalog > **ceiling**(*Expr1*) $\Rightarrow$ *heltal* $\text{ceiling}(.456)$  1.

Ger det närmaste heltal som är  $\geq$  argumentet.

Argumentet kan vara ett reellt eller ett komplext tal.

**Obs:** Se även **floor()**.

**ceiling**(*List1*) $\Rightarrow$ *lista* $\text{ceiling}(\{-3.1, 1.2, 5\})$   $\{-3., 1.3\}$ **ceiling**(*Matrix1*) $\Rightarrow$ *matrix* $\text{ceiling}\left(\begin{bmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{bmatrix}\right)$   $\begin{bmatrix} 0 & -3. \cdot i \\ 2. & 4 \end{bmatrix}$ 

Ger en lista eller matrix över taket för varje element.

**centralDiff()**Katalog > **centralDiff**(*Utr1*, *Var* [=Värde] [,*Steg*]) $\Rightarrow$ *uttryck*
$$\frac{\text{centralDiff}(\cos(x), x, h) - (\cos(x-h) - \cos(x+h))}{2 \cdot h}$$
**centralDiff**(*Utr1*, *Var* [,*Steg*]) | *Var* = *Värde* $\Rightarrow$ *uttryck*
$$\lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h))$$
  $-\sin(x)$ **centralDiff**(*Utr1*, *Var* [=Värde] [,*Lista*]) $\Rightarrow$ *lista*
$$\text{centralDiff}(x^3, x, 0.01)$$
  $3. \cdot (x^2 + 0.000033)$ **centralDiff**(*Lista1*, *Var* [=Värde] [,*Steg*]) $\Rightarrow$ *lista*
$$\text{centralDiff}(\cos(x), x) | x = \frac{\pi}{2}$$
  $-1.$ **centralDiff**(*Matrix1*, *Var* [=Värde] [,*Steg*]) $\Rightarrow$ *matrix*
$$\text{centralDiff}(x^2, x, \{0.01, 0.1\})$$
  $\{2. \cdot x, 2. \cdot x\}$ 

Ger den numeriska derivatan genom att använda formeln för symmetrisk differenskvot.

När *Värde* specificeras överstyr detta värde eventuella tidigare variabeltilldelningar eller aktuella ersättningar av typ " | " för variabeln.

*Steg* är stegvärdet. Om *Steg* utelämnas används förinställningen 0.001.

När du använder *Lista1* eller *Matris1* utförs operationen på värdena i listan eller matriselementen.

**Obs:** Se även **avgRC()** och **d()**.

**cFactor()**

**cFactor**(*Expr1* [, *Var*])  $\Rightarrow$  uttryck

**cFactor**(*List1* [, *Var*])  $\Rightarrow$  lista

**cFactor**(*Matrix1* [, *Var*])  $\Rightarrow$  matris

**cFactor**(*Expr1*) ger en faktorisering av *Expr1* baserad på uttryckets alla variabler med en gemensam nämnare.

*Expr1* faktoriseras så långt det går till linjära, rationella faktorer även om detta inför nya icke-reella tal. Detta alternativ är lämpligt om du vill ha en faktorisering baserad på mer än en variabel.

**cFactor**(*Expr1*, *Var*) ger en faktorisering av *Expr1* baserad på variabeln *Var*.

*Expr1* faktoriseras så långt det går till faktorer som är linjära i *Var*, med kanske icke-reella konstanter, även om detta inför irrationella konstanter eller deluttryck som är irrationella i andra variabler.

Faktorerna och deras termer sorteras med *Var* som huvudvariabel. Liknande potenser av *Var* samlas in i varje faktor. Inkludera *Var* om faktorisering baserad på endast denna variabel behövs och du är villig att acceptera irrationella uttryck i andra variabler för att öka faktoriseringen baserad på *Var*. Viss tillfällig faktorisering kan ske vad gäller andra variabler.

|                                                                            |                                                                   |
|----------------------------------------------------------------------------|-------------------------------------------------------------------|
| $\text{cFactor}\left(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x\right)$ | $a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$                   |
| $\text{cFactor}\left(x^2 + \frac{4}{9}\right)$                             | $\frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}$ |
| $\text{cFactor}\left(x^2 + 3\right)$                                       | $x^2 + 3$                                                         |
| $\text{cFactor}\left(x^2 + a\right)$                                       | $x^2 + a$                                                         |

|                                                                            |                                                       |
|----------------------------------------------------------------------------|-------------------------------------------------------|
| $\text{cFactor}\left(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x\right)$ | $a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$       |
| $\text{cFactor}\left(x^2 + 3 \cdot x\right)$                               | $(x + \sqrt{3} \cdot i) \cdot (x - \sqrt{3} \cdot i)$ |
| $\text{cFactor}\left(x^2 + a \cdot x\right)$                               | $(x + \sqrt{a} \cdot i) \cdot (x + \sqrt{a} \cdot i)$ |

## cFactor()

Katalog > 

Med inställningen Auto i läge **Auto eller Ungefärlig** medger inkludering av *Var* också en uppskattning med koefficienter med flytande decimalkomma när irrationella koefficienter inte explicit kan uttryckas kortfattat med termerna i de inbyggda funktionerna. Även med endast en variabel kan inkludering av *Var* ge en mer fullständig faktorisering.

**Obs:** Se även **faktor()**.

$$\begin{array}{l} \text{cFactor}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3) \\ x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3 \\ \text{cFactor}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3, x) \\ (x-0.964673)\cdot(x+0.611649)\cdot(x+2.12543)\cdot(x+ \end{array}$$

För att se hela resultatet, tryck på  och använd sedan  och  för att flytta markören.

## char()

Katalog > 

**char(Integer)** ⇒ tecken

Ger en teckensträng som innehåller tecknet med numret *Integer* från handenhetens teckenuppsättning. Det giltiga området för *Integer* är 0–65535.

$$\begin{array}{ll} \text{char}(38) & "&" \\ \text{char}(65) & "&" \end{array}$$

## charPoly()

Katalog > 

**charPoly(squareMatrix, Var)** ⇒ polynom

**charPoly**

(*squareMatrix*, *Expr*) ⇒ polynom

**charPoly**

(*squareMatrix1*, *Matrix2*) ⇒ polynom

Ger det karakteristiska polynomet för *squareMatrix*. Det karakteristiska polynomet för  $n \times n$  matris *A*, betecknat  $p_A(\lambda)$ , är polynomet definierat av

$$p_A(\lambda) = \det(\lambda \cdot I - A)$$

där *I* betecknar enhetsmatrisen  $n \times n$ .

*squareMatrix1* och *squareMatrix2* måste ha samma dimensioner.

$$\begin{array}{l} m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix} \\ \text{charPoly}(m, x) & -x^3+5\cdot x^2+7\cdot x-35 \\ \text{charPoly}(m, x^2+1) & -x^6+2\cdot x^4+14\cdot x^2-24 \\ \text{charPoly}(m, m) & 0 \end{array}$$

## χ<sup>2</sup>2way

Katalog > 

**χ<sup>2</sup>2way** *ObsMatrix*

**chi22way** *ObsMatrix*

Beräknar ett  $\chi^2$ -test för association på 2-vägstabellen över antal i den observerade matrisen *ObsMatrix*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en matris, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                                        |
|------------------|------------------------------------------------------------------------------------|
| stat. $\chi^2$   | Chi-kvadratstatistik: $\text{summa (observerad - förväntad)}^2 / \text{förväntad}$ |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas                      |
| stat.df          | Frihetsgrader hos chi-kvadratstatistiken                                           |
| stat.ExpMat      | Matris över förväntad elementräknetabell, baserad på nollhypotesen                 |
| stat.CompMat     | Matris över elementbidrag till chi-kvadratstatistiken                              |

 **$\chi^2$ Cdf()**

**$\chi^2$ Cdf**(*lowBound*,*upBound*,*df*) $\Rightarrow$ tal om *lowBound* och *upBound* är tal, lista om *lowBound* och *upBound* är listor

**chi2Cdf**(*lowBound*,*upBound*,*df*) $\Rightarrow$ tal om *lowBound* och *upBound* är tal, lista om *lowBound* och *upBound* är listor

Beräknar sannolikheten för  $\chi^2$ -fördelning mellan *lowBound* och *upBound* för den specificerade frihetsgraden *df*.

För  $P(X \leq \text{upBound})$ , sätt *lowBound* = 0.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

 **$\chi^2$ GOF**

**$\chi^2$ GOF** *obsList*,*expList*,*df*

**chi2GOF** *obsList*,*expList*,*df*

Utför ett test för att bekräfta att urvalsdata är från en population som följer en specificerad fördelning. *obsList* är en lista med data och måste innehålla heltal. En sammanfattning av resultaten visas i variabeln *stat.results*, på sidan 184.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                     |
|------------------|-----------------------------------------------------------------|
| stat.χ²          | Chi-kvadratstatistik: summa (observerad - förväntad)²/förväntad |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas   |
| stat.df          | Frihetsgrader hos chi-kvadratstatistiken                        |
| stat.CompList    | Elementbidrag till chi-kvadratstatistiken                       |

χ²Pdf()

$\chi^2\text{Pdf}(XVal,df)\Rightarrow tal$  om *XVal* är ett tal, *lista* om *XVal* är en lista

$\text{chi2Pdf}(XVal,df)\Rightarrow tal$  om *XVal* är ett tal, *lista* om *XVal* är en lista

Beräknar värde hos täthetsfunktionen (pdf) för  $\chi^2$ -fördelningen vid ett specificerat *XVal*-värde för den specificerade frihetsgraden *df*.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

ClearAZ

ClearAZ

|         |      |
|---------|------|
| 5 → b   | 5    |
| b       | 5    |
| ClearAZ | Done |
| b       | b    |

Rensar alla variabler som har ett enda tecken i det aktuella problemet.

Om en eller flera variabler är låsta visar detta kommando ett felmeddelande och tar endast bort olåsta variabler. Se

**unLock**, på sidan 206.

**ClrErr**

För ett exempel på **ClrErr**, se exempel 2 under kommandot **Try**, på sidan 199.

Rensar felstatusen och ställer in systemvariabeln *errCode* på noll.

Villkoret **Else** i blocket **Try...Else...EndTry** bör använda **ClrErr** eller **PassErr**. Om felet skall processas eller ignoreras, använd **ClrErr**. Om det är okänt hur felet skall hanteras, använd **PassErr** för att skicka felet vidare till nästa felhanterare. Om det inte finns någon ytterligare felhanterare för **Try...Else...EndTry** visas feldialogrutan som normal.

**Obs:** Se även **PassErr**, på sidan 137 och **Try**, på sidan 199.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

**colAugment()**

**colAugment**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Ger en ny matris med *Matrix2* fogad till *Matrix1*. Matriserna måste ha samma kolumndimensioner och *Matrix2* fogas till *Matrix1* som nya rader. Ändrar inte *Matrix1* eller *Matrix2*.

|                                                               |                                                         |
|---------------------------------------------------------------|---------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$          |
| $\begin{bmatrix} 5 & 6 \end{bmatrix} \rightarrow m2$          | $\begin{bmatrix} 5 & 6 \end{bmatrix}$                   |
| $\text{colAugment}(m1, m2)$                                   | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ |

**colDim()**

**colDim**(*Matrix*) $\Rightarrow$ *uttryck*

Ger antalet kolumner i *Matrix*.

|                                                                                  |   |
|----------------------------------------------------------------------------------|---|
| $\text{colDim}\left(\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}\right)$ | 3 |
|----------------------------------------------------------------------------------|---|

**Obs:** Se även **rowDim()**.

**colNorm()**

**colNorm**(*Matrix*) $\Rightarrow$ *uttryck*

Ger maximum av summorna av absolutbeloppen på elementen i kolumnerna i *Matrix*.

|                                                                          |                                                          |
|--------------------------------------------------------------------------|----------------------------------------------------------|
| $\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$ | $\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$ |
| $\text{colNorm}(mat)$                                                    | 9                                                        |

**Obs:** Odefinierade matriselement är ej tillåtna. Se även **rowNorm()**.

## comDenom()

**comDenom**(*Expr1* [, *Var*])  $\Rightarrow$  *uttryck*

**comDenom**(*List1* [, *Var*])  $\Rightarrow$  *lista*

**comDenom**(*Matrix1* [, *Var*])  $\Rightarrow$  *matrix*

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$$

$$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

**comDenom**(*Expr1*) ger en reducerad kvot mellan en fullt expanderad täljare och en fullt expanderad nämnare.

**comDenom**(*Expr1*, *Var*) ger en reducerad kvot mellan täljare och nämnare som expanderats enligt *Var*. Termerna och deras faktorer sorteras med *Var* som huvudvariabel. Liknande potenser av *Var* samlas in. Viss tillfällig faktorisering kan ske av de insamlade koefficienterna. Jämfört med att utesluta *Var* sparar detta ofta tid, minne och skärmutrymme, vilket gör uttrycket mer begripligt. Det gör också att påföljande operationer på resultatet går snabbare och med mindre risk att minnet tar slut.

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, x\right)$$

$$\frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, y\right)$$

$$\frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

Om *Var* inte förekommer i *Expr1* ger **comDenom**(*Expr1*, *Var*) en reducerad kvot mellan en oexpanderad täljare och en oexpanderad nämnare. Sådana resultat sparar i regel ännu mer tid, minne och skärmutrymme. Sådana delvis faktorerade resultat gör också att påföljande operationer på resultatet går mycket snabbare och med mycket mindre risk att minnet tar slut.

$$\text{Define } \text{comden}(\text{exprn}) = \text{comDenom}(\text{exprn}, \text{abc})$$

$$\text{comden}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right) \quad \frac{(x^2+2 \cdot x+2) \cdot y \cdot (y+1)}{(x+1)^2}$$

Även om det inte finns någon nämnare är funktionen **comden** ofta ett snabbt sätt att erhålla en delvis faktorisering om **factor()** är för långsam eller om den utarmar minnet.

$$\text{comden}(1234 \cdot x^2 \cdot (y^3-y) + 2468 \cdot x \cdot (y^2-1))$$

$$1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2-1)$$

**Tips:** Mata in denna **comden()** funktionsdefinition och prova den rutinmässigt som ett alternativ till **comDenom()** och **factor()**.

## completeSquare ()

**completeSquare(ExprOrEqn, Var)**uttryck eller ekvation

**completeSquare(ExprOrEqn, Var^Power)**⇒uttryck eller ekvation

**completeSquare(ExprOrEqn, Var1, Var2 [...])**⇒uttryck eller ekvation

**completeSquare(ExprOrEqn, {Var1, Var2 [...]})**⇒uttryck eller ekvation

Konverterar ett kvadratisk polynomuttryck på formen  $a \cdot x^2 + b \cdot x + c$  till formen  $a \cdot (x-h)^2 + k$

– eller –

Konverterar en andragradsekvation på formen  $a \cdot x^2 + b \cdot x + c = d$  till formen  $a \cdot (x-h)^2 = k$

Det första argumentet måste vara ett kvadratisk uttryck eller en andragradsekvation i standardform med avseende på det andra argumentet.

Det andra argumentet måste vara en enda envariabelterm eller en enda envariabelterm upphöjd till en rationell potens, till exempel  $x$ ,  $y^2$  eller  $z^{1/3}$ .

Den tredje och fjärde syntaxen försöker att fullborda kvadraten avseende variablerna Var1, Var2 [...]).

$$\text{completeSquare}(x^2+2 \cdot x+3, x) \quad (x+1)^2+2$$

$$\text{completeSquare}(x^2+2 \cdot x=3, x) \quad (x+1)^2=4$$

$$\text{completeSquare}(x^6+2 \cdot x^3+3, x^3) \quad (x^3+1)^2+2$$

$$\text{completeSquare}(x^2+4 \cdot x+y^2+6 \cdot y+3=0, x, y) \quad (x+2)^2+(y+3)^2=10$$

$$\text{completeSquare}(3 \cdot x^2+2 \cdot y+7 \cdot y^2+4 \cdot x=3, \{x, y\}) \quad 3 \cdot \left(x+\frac{2}{3}\right)^2+7 \cdot \left(y+\frac{1}{7}\right)^2=\frac{94}{21}$$

$$\text{completeSquare}(x^2+2 \cdot x \cdot y, x, y) \quad (x+y)^2-y^2$$



**conj()**Katalog > **conj**(*Expr1*) $\Rightarrow$ uttryck

|                           |              |
|---------------------------|--------------|
| $\text{conj}(1+2\cdot i)$ | $1-2\cdot i$ |
|---------------------------|--------------|

**conj**(*List1*) $\Rightarrow$ lista

|                                                                                   |                                                          |
|-----------------------------------------------------------------------------------|----------------------------------------------------------|
| $\text{conj}\left(\begin{bmatrix} 2 & 1-3\cdot i \\ -i & -7 \end{bmatrix}\right)$ | $\begin{bmatrix} 2 & 1+3\cdot i \\ i & -7 \end{bmatrix}$ |
|-----------------------------------------------------------------------------------|----------------------------------------------------------|

**conj**(*Matrix1*) $\Rightarrow$ matrix

|                  |           |
|------------------|-----------|
| $\text{conj}(z)$ | $\bar{z}$ |
|------------------|-----------|

Ger argumentets komplexkonjugat.

|                           |              |
|---------------------------|--------------|
| $\text{conj}(x+i\cdot y)$ | $x-y\cdot i$ |
|---------------------------|--------------|

**Obs:** Alla odefinierade variabler behandlas som reella variabler.**constructMat()**Katalog > **constructMat****(Expr,Var1,Var2,numRows,numCols)**  
 $\Rightarrow$ matrix

|                                                             |                                                                                                                                                                                                         |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right)$ | $\begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$ |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Ger en matrix baserad på argumenten.

*Expr* är ett uttryck i variablerna *Var1* och *Var2*. Element i den resulterande matrisen skapas genom att utvärdera *Expr* för varje ökat värde på *Var1* och *Var2*.*Var1* ökas automatiskt från 1 till och med *numRows*. Inom varje rad ökas *Var2* från 1 till och med *numCols*.**CopyVar**Katalog > **CopyVar** *Var1, Var2*

|                           |      |
|---------------------------|------|
| Define $a(x)=\frac{1}{x}$ | Done |
|---------------------------|------|

**CopyVar** *Var1., Var2.*

|                   |      |
|-------------------|------|
| Define $b(x)=x^2$ | Done |
|-------------------|------|

**CopyVar** *Var1, Var2* kopierar värdet på variabel *Var1* till variabel *Var2*, och skapar *Var2* vid behov. Variabeln *Var1* måste ha ett värde.

|                            |               |
|----------------------------|---------------|
| CopyVar <i>a,c:</i> $c(4)$ | $\frac{1}{4}$ |
|----------------------------|---------------|

|                            |    |
|----------------------------|----|
| CopyVar <i>b,c:</i> $c(4)$ | 16 |
|----------------------------|----|

Om *Var1* är namnet på en befintlig användardefinierad funktion kopieras definitionen på denna funktion till funktionen *Var2*. Funktionen *Var1* måste vara definierad.

*Var1* måste uppfylla kraven för namngivning av variabler eller måste vara ett indirection-uttryck som förenklas till ett variabelnamn som uppfyller kraven.

**CopyVar** *Var1.*, *Var2.* kopierar alla led i variabelgruppen *Var1.* till gruppen *Var2.* och skapar *Var2.* vid behov.

*Var1.* måste vara namnet på en befintlig variabelgrupp, till exempel, den statistiska *stat.nn*-resultat eller variabler skapade med funktionen **LibShortcut()**. Om *Var2.* redan finns ersätter detta kommando alla led som är gemensamma för båda grupperna och lägger till de led som inte redan finns. Om en eller flera medlemmar av *Var2.* är låsta lämnas alla medlemmar av *Var2.* oförändrade.

|                           |                                                                                                                                                                                                                                                      |             |       |       |             |       |        |             |       |       |             |       |       |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------|-------|-------------|-------|--------|-------------|-------|-------|-------------|-------|-------|
| <i>aa.a:=45</i>           | 45                                                                                                                                                                                                                                                   |             |       |       |             |       |        |             |       |       |             |       |       |
| <i>aa.b:=6.78</i>         | 6.78                                                                                                                                                                                                                                                 |             |       |       |             |       |        |             |       |       |             |       |       |
| CopyVar <i>aa.,bb.</i>    | <i>Done</i>                                                                                                                                                                                                                                          |             |       |       |             |       |        |             |       |       |             |       |       |
| getVarInfo <sup>(0)</sup> | <table><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"[" 0</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"[" 0,</td></tr><tr><td><i>bb.a</i></td><td>"NUM"</td><td>"[" 0</td></tr><tr><td><i>bb.b</i></td><td>"NUM"</td><td>"[" 0</td></tr></table> | <i>aa.a</i> | "NUM" | "[" 0 | <i>aa.b</i> | "NUM" | "[" 0, | <i>bb.a</i> | "NUM" | "[" 0 | <i>bb.b</i> | "NUM" | "[" 0 |
| <i>aa.a</i>               | "NUM"                                                                                                                                                                                                                                                | "[" 0       |       |       |             |       |        |             |       |       |             |       |       |
| <i>aa.b</i>               | "NUM"                                                                                                                                                                                                                                                | "[" 0,      |       |       |             |       |        |             |       |       |             |       |       |
| <i>bb.a</i>               | "NUM"                                                                                                                                                                                                                                                | "[" 0       |       |       |             |       |        |             |       |       |             |       |       |
| <i>bb.b</i>               | "NUM"                                                                                                                                                                                                                                                | "[" 0       |       |       |             |       |        |             |       |       |             |       |       |

## corrMat()

**corrMat**(*List1*,*List2*[,...,*List20*])

Beräknar korrelationsmatrisen för den sammanfogade matrisen [*List1*, *List2*, ..., *List20*].

## ►cos

*Expr* ►cos

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>cos.

Representerar *Expr* i termer av cosinus. Detta är en omvandlingsoperator för visning. Den kan endast användas i slutet av inmatningsraden.

►cos reducerar alla potenser av sin(...) modulo  $1 - \cos(\dots)^2$  så att eventuella återstående potenser av cos(...) har exponenter i området (0, 2). Resultatet kommer sålunda att vara fritt från sin(...) om, och endast om, sin(...) finns i det givna uttrycket vid jämna potenser.

|                    |                   |
|--------------------|-------------------|
| $(\sin(x))^2$ ►cos | $1 - (\cos(x))^2$ |
|--------------------|-------------------|

**Obs:** Denna omvandlingsoperator stöds inte i vinkellägena Grader och Nygrader. Innan du använder den, kontrollera att vinkelläget är inställt på Radianer och att *Expr* inte innehåller explicita referenser till vinklar i grader eller nygrader.

## cos()

 tangent

**cos**(*Expr1*) ⇒ *uttryck*

I vinkelläget Grader:

**cos**(*List1*) ⇒ *lista*

|                                  |                                    |
|----------------------------------|------------------------------------|
| $\cos\left(\frac{\pi}{4}\right)$ | $\frac{\sqrt{2}}{2}$               |
| $\cos(45)$                       | $\frac{\sqrt{2}}{2}$               |
| $\cos(\{0,60,90\})$              | $\left\{1, \frac{1}{2}, 0\right\}$ |

**cos**(*Expr1*) ger argumentets cosinus som ett uttryck.

**cos**(*List1*) ger en lista på cosinus för alla element i *List1*.

**Obs:** Argumentet tolkas som en vinkel i grader, nygrader eller radianer enligt det inställda vinkelläget. Du kan använda °, °G eller °R för att tillfälligt överstyra vinkelläget.

I vinkelläget Nygrader:

|                      |                                           |
|----------------------|-------------------------------------------|
| $\cos(\{0,50,100\})$ | $\left\{1, \frac{\sqrt{2}}{2}, 0\right\}$ |
|----------------------|-------------------------------------------|

I vinkelläget Radianer:

|                                  |                      |
|----------------------------------|----------------------|
| $\cos\left(\frac{\pi}{4}\right)$ | $\frac{\sqrt{2}}{2}$ |
| $\cos(45^\circ)$                 | $\frac{\sqrt{2}}{2}$ |

**cos**(*squareMatrix1*) ⇒ *kvadratMatris*

I vinkelläget Radianer:

Ger matrisen med cosinus för *squareMatrix1*. Detta är inte detsamma som att beräkna cosinus för varje element.

|                                                                          |                                                                                                                                     |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| $\cos\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$ | $\begin{bmatrix} 0.212493 & 0.205064 & 0.121389 \\ 0.160871 & 0.259042 & 0.037126 \\ 0.248079 & -0.090153 & 0.218972 \end{bmatrix}$ |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|

När en skalär funktion *f*(A) används på *squareMatrix1* (A) beräknas resultatet med algoritmen:

Beräkna egenvärdena ( $\lambda_i$ ) och egenvektorerna ( $V_i$ ) för A.

*squareMatrix1* måste vara möjlig att diagonalisera. Den får inte heller ha symboliska variabler som inte har tilldelats ett värde.

Forma matriserna:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ and } X = [V_1, V_2, \dots, V_n]$$

Då är  $A = X B X^{-1}$  och  $f(A) = X f(B) X^{-1}$ .  
Exempelvis  $\cos(A) = X \cos(B) X^{-1}$  där:

$\cos(B) =$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Alla beräkningar utförs med flyttalsaritmetik.

$\cos^{-1}(Expr1) \Rightarrow$  uttryck

I vinkelläget Grader:

$\cos^{-1}(List1) \Rightarrow$  lista

$\cos^{-1}(1)$  0

$\cos^{-1}(Expr1)$  ger den vinkel vars cosinus är *Expr1* som ett uttryck.

I vinkelläget Nygrader:

$\cos^{-1}(List1)$  ger en lista på invers cosinus för varje element i *List1*.

$\cos^{-1}(0)$  100

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

I vinkelläget Radianer:

$\cos^{-1}(\{0, 0.2, 0.5\})$   $\left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccos (...)**.

$\cos^{-1}(squareMatrix1) \Rightarrow squareMatrix$

I vinkelläget Radianer och i Rektangulärt komplext format:

## $\cos^{-1}()$

 **tangent**

Ger matrisen med invers cosinus för *squareMatrix1*. Detta är inte detsamma som att beräkna invers cosinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

$$\cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \begin{bmatrix} 1.73485+0.064606\cdot i & -1.49086+2.10514 \\ -0.725533+1.51594\cdot i & 0.623491+0.77836\cdot i \\ -2.08316+2.63205\cdot i & 1.79018-1.27182\cdot i \end{bmatrix}$$

För att se hela resultatet, tryck på  och använd sedan  och  för att flytta markören.

## **cosh()**

**Katalog** > 

**cosh(Expr1)**⇒*uttryck*

I vinkelläget Grader:

**cosh(List1)**⇒*lista*

$$\cosh\left(\left(\frac{\pi}{4}\right)_r\right) \quad \cosh(45)$$

**cosh(Expr1)** ger argumentets hyperboliska cosinus som ett uttryck.

**cosh(List1)** ger en lista på hyperbolisk cosinus för varje element i *List1*.

**cosh(squareMatrix1)**⇒*kvadratMatris*

I vinkelläget Radianer:

Ger matrisen med hyperbolisk cosinus för *squareMatrix1*. Detta är inte detsamma som att beräkna hyperbolisk cosinus för varje element. Se **cos()** för information om beräkningsmetoden.

$$\cosh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

## **cosh<sup>-1</sup>()**

**Katalog** > 

**cosh<sup>-1</sup>(Expr1)**⇒*uttryck*

$$\cosh^{-1}(1) \quad 0$$

**cosh<sup>-1</sup>(List1)**⇒*lista*

$$\cosh^{-1}(\{1,2,1,3\}) \quad \{0,1.37286,\cosh^{-1}(3)\}$$

**cosh<sup>-1</sup>(Expr1)** ger argumentets inversa hyperboliska cosinus som ett uttryck.

**cosh<sup>-1</sup>(List1)** ger en lista på invers hyperbolisk cosinus för varje element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccosh (...)**.

**cosh<sup>-1</sup>(squareMatrix1)⇒kvadratMatris**

Ger matrisen med invers hyperbolisk cosinus för *squareMatrix1*. Detta är inte detsamma som att beräkna invers hyperbolisk cosinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I vinkelläget Radianer och i Rektangulärt komplext format:

$$\cosh^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 2.52503+1.73485\cdot i & -0.009241-1.49086\cdot i \\ 0.486969-0.725533\cdot i & 1.66262+0.623491\cdot i \\ -0.322354-2.08316\cdot i & 1.26707+1.79018\cdot i \end{bmatrix}$$

För att se hela resultatet, tryck på  och använd sedan  och  för att flytta markören.

## cot()

 tangent

**cot(Expr1)⇒uttryck**

**cot(List1)⇒lista**

Ger cotangens för *Expr1* eller en lista på cotangens för alla element i *List1*.

**Obs:** Argumentet tolkas som en vinkel i grader, nygrader eller radianer enligt det inställda vinkelläget. Du kan använda <sup>°</sup>, <sup>G</sup> eller <sup>r</sup> för att tillfälligt överstyra vinkelläget.

I vinkelläget Grader:

$$\cot(45) = 1$$

I vinkelläget Nygrader:

$$\cot(50) = 1$$

I vinkelläget Radianer:

$$\cot(\{1, 2, 1, 3\}) = \left\{ \frac{1}{\tan(1)}, -0.584848, \frac{1}{\tan(3)} \right\}$$

cot<sup>-1</sup>() tangent

**cot<sup>-1</sup>(Expr1)⇒uttryck**

**cot<sup>-1</sup>(List1)⇒lista**

Ger den vinkel vars cotangens är *Expr1* eller en lista på invers cotangens för varje element i *List1*.

I vinkelläget Grader:

$$\cot^{-1}(1) = 45.$$

I vinkelläget Nygrader:

$$\cot^{-1}(1) = 50.$$

**cot<sup>-1</sup>()****trig** tangent

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccot (...)**.

I vinkelläget Radianer:

|                       |                 |
|-----------------------|-----------------|
| cot <sup>-1</sup> (1) | $\frac{\pi}{4}$ |
|-----------------------|-----------------|

**coth()****Katalog >** **coth(Expr1)**⇒uttryck

|           |         |
|-----------|---------|
| coth(1.2) | 1.19954 |
|-----------|---------|

**coth(List1)**⇒lista

|               |                                                |
|---------------|------------------------------------------------|
| coth({1,3,2}) | $\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$ |
|---------------|------------------------------------------------|

Ger hyperbolisk cotangens för *Expr1* eller en lista på hyperbolisk cotangens för alla element i *List1*.

**coth<sup>-1</sup>()****Katalog >** **coth<sup>-1</sup>(Expr1)**⇒uttryck

|                          |          |
|--------------------------|----------|
| coth <sup>-1</sup> (3.5) | 0.293893 |
|--------------------------|----------|

**coth<sup>-1</sup>(List1)**⇒lista

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| coth <sup>-1</sup> ({-2,2,1,6}) | $\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$ |
|---------------------------------|---------------------------------------------------------------------------------------|

Ger den inversa hyperboliska cotangensen för *Expr1* eller en lista på invers hyperbolisk cotangens för alla element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccoth (...)**.

**count()****Katalog >** **count(Value1orList1 [,Value2orList2 [...]])**⇒värde

|              |   |
|--------------|---|
| count(2,4,6) | 3 |
|--------------|---|

|                |   |
|----------------|---|
| count({2,4,6}) | 3 |
|----------------|---|

Ger det totala ackumulerade antalet element i argumenten som utvärderas till numeriska värden.

|                                                                    |   |
|--------------------------------------------------------------------|---|
| count(2,{4,6}, $\begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}$ ) | 7 |
|--------------------------------------------------------------------|---|

Varje argument kan vara ett uttryck, ett värde, en lista eller en matris. Du kan blanda datatyper och använda argument med olika dimensioner.

|                                                                |   |
|----------------------------------------------------------------|---|
| count( $\frac{1}{2}$ ,3+4 <i>i</i> ,undef,"hello",x+5,sign(0)) | 2 |
|----------------------------------------------------------------|---|

För en lista, en matris, eller ett område av celler, utvärderas varje element för att bestämma om det skall inkluderas i räkningen.

I applikationen Listor och kalkylblad kan du använda ett område av celler i stället för ett argument.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

I det sista exemplet räknas endast  $1/2$  och  $3+4*i$ . De återstående argumenten, förutsatt att  $x$  är odefinierad, utvärderas inte till numeriska värden.

## countif()

**countif(List, Criteria) ⇒ värde**

Ger det totala ackumulerade antalet element i *List* som uppfyller specificerade *Criteria*.

*Criteria* kan vara:

- Ett värde, ett uttryck eller en sträng. Som exempel räknar **3** endast de element i *List* som förenklas till värdet 3.
- Ett booleskt uttryck som innehåller symbolen **?** fungerar som platshållare för varje element. Som exempel räknar **?<5** endast de element i *List* som är lägre än 5.

I applikationen Listor och kalkylblad kan du använda ett område av celler i stället för *List*.

Tomma element i listan ignoreras. För mer information om tomma element, se på sidan 261.

**Obs:** Se även **sumif()**, på sidan 188 och **frequency()**, på sidan 78.

|                                               |   |
|-----------------------------------------------|---|
| <code>countif({1,3,"abc",undef,3,1},3)</code> | 2 |
|-----------------------------------------------|---|

Räknar antalet element som är lika med 3.

|                                                   |   |
|---------------------------------------------------|---|
| <code>countif({"abc","def","abc",3},"def")</code> | 1 |
|---------------------------------------------------|---|

Räknar antalet element som är lika med "def."

|                                             |   |
|---------------------------------------------|---|
| <code>countif({x^-2,x^-1,1,x,x^2},x)</code> | 1 |
|---------------------------------------------|---|

Räknar antalet element som är lika med  $x$ . I detta exempel förutsätts att variabeln  $x$  är odefinierad.

|                                          |   |
|------------------------------------------|---|
| <code>countif({1,3,5,7,9},?&lt;5)</code> | 2 |
|------------------------------------------|---|

Räknar 1 och 3.

|                                               |   |
|-----------------------------------------------|---|
| <code>countif({1,3,5,7,9},2&lt;?&lt;8)</code> | 3 |
|-----------------------------------------------|---|

Räknar 3, 5 och 7.

|                                                    |   |
|----------------------------------------------------|---|
| <code>countif({1,3,5,7,9},?&lt;4 or ?&gt;6)</code> | 4 |
|----------------------------------------------------|---|

Räknar 1, 3, 7 och 9.



**cPolyRoots()**Katalog > **cPolyRoots**(*Poly*,*Var*)⇒*lista*

$$\text{polyRoots}(y^3+1,y) \quad \{-1\}$$

**cPolyRoots**(*ListaPåKoeff*)⇒*lista*

$$\text{cPolyRoots}(y^3+1,y) \\ \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$

Den första syntaxen, **cPolyRoots** (*Poly*,*Var*), ger en lista på komplexa rötter i polynomet *Poly* med avseende på *Var*.

$$\text{polyRoots}(x^2+2\cdot x+1,x) \quad \{-1,-1\}$$

*Poly* måste vara ett polynom i en variabel.

$$\text{cPolyRoots}(\{1,2,1\}) \quad \{-1,-1\}$$

Den andra syntaxen, **cPolyRoots** (*ListaPåKoeff*), ger en lista på komplexa rötter för koefficienterna i *ListaPåKoeff*.

**Obs:** Se även **polyRoots()**, på sidan 142.

**crossP()**Katalog > **crossP**(*List1*, *List2*)⇒*lista*

Ger vektorprodukten av *List1* och *List2* som en lista.

$$\text{crossP}(\{a1,b1\},\{a2,b2\}) \\ \{0,0,a1\cdot b2-a2\cdot b1\}$$

*List1* och *List2* måste ha samma dimension och dimensionen måste vara antingen 2 eller 3.

$$\text{crossP}(\{0.1,2.2,-5\},\{1,-0.5,0\}) \\ \{-2.5,-5,-2.25\}$$

**crossP**(*Vector1*, *Vector2*)⇒*vektor*

Ger en rad- eller kolumnvektor (beroende på argumenten) som är vektorprodukten av *Vector1* och *Vector2*.

$$\text{crossP}([1 \ 2 \ 3],[4 \ 5 \ 6]) \quad [-3 \ 6 \ -3]$$

$$\text{crossP}([1 \ 2],[3 \ 4]) \quad [0 \ 0 \ -2]$$

Både *Vector1* och *Vector2* måste vara radvektorer eller båda måste vara kolumnvektorer. Båda vektorerna måste ha samma dimension och dimensionen måste vara antingen 2 eller 3.

**csc()** tangent**csc**(*Expr1*)⇒*uttryck*

I vinkelläget Grader:

**csc**(*List1*)⇒*lista*

$$\text{csc}(45) \quad \sqrt{2}$$

Ger cosekanten för *Expr1* eller en lista på cosekanten för alla element i *List1*.

**csc()** **tangent**

I vinkelläget Nygrader:

$$\text{csc}(50) \quad \sqrt{2}$$

I vinkelläget Radianer:

$$\text{csc}\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) \quad \left\{\frac{1}{\sin(1)}, 1, \frac{2\sqrt{3}}{3}\right\}$$

**csc<sup>-1</sup>()** **tangent****csc<sup>-1</sup>(Expr1)** ⇒ uttryck

I vinkelläget Grader:

**csc<sup>-1</sup>(List1)** ⇒ lista

$$\text{csc}^{-1}(1) \quad 90.$$

Ger den vinkel vars cosekant är *Expr1* eller en lista på den inversa cosekanten för varje element i *List1*.

I vinkelläget Nygrader:

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

$$\text{csc}^{-1}(1) \quad 100.$$

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccsc (...)**.

I vinkelläget Radianer:

$$\text{csc}^{-1}\left(\left\{1, 4, 6\right\}\right) \quad \left\{\frac{\pi}{2}, \sin^{-1}\left(\frac{1}{4}\right), \sin^{-1}\left(\frac{1}{6}\right)\right\}$$

**csch()****Katalog** > **csch(Expr1)** ⇒ uttryck

$$\text{csch}(3) \quad \frac{1}{\sinh(3)}$$

**csch(List1)** ⇒ lista

Ger den hyperboliska cosekanten för *Expr1* eller en lista på den hyperboliska cosekanten för alla element i *List1*.

$$\text{csch}(\{1, 2, 1, 4\}) \quad \left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

**csch<sup>-1</sup>()****Katalog** > **csch<sup>-1</sup>(Expr1)** ⇒ uttryck

$$\text{csch}^{-1}(1) \quad \sinh^{-1}(1)$$

**csch<sup>-1</sup>(List1)** ⇒ lista

$$\text{csch}^{-1}(\{1, 2, 1, 3\}) \quad \left\{\sinh^{-1}(1), 0.459815, \sinh^{-1}\left(\frac{1}{3}\right)\right\}$$

Ger den inversa hyperboliska cosekanten för *Expr1* eller en lista på den inversa hyperboliska cosekanten för alla element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arccsch (...)**.

**cSolve()**

**cSolve**(*Equation, Var*)⇒*Booleskt uttryck*

**cSolve**(*Equation, Var=Guess*)⇒*Booleskt uttryck*

**cSolve**(*Inequality, Var*)⇒*Booleskt uttryck*

Ger möjliga komplexa lösningar på en ekvation eller olikhet för *Var*. Målet är att producera "kandidater" för alla reella och icke-reella lösningar. Även om *Equation* är reell medger **cSolve()** icke-reella resultat i Real Complex Format.

Även om alla odefinierade variabler som inte slutar med ett understrykningstecken (\_) behandlas som om de vore reella kan **cSolve()** lösa polynomekvationer för komplexa lösningar.

**cSolve()** ställer temporärt in området på komplext under lösningen även om det aktuella området är reellt. I det komplexa området använder rationella potenser med udda nämnare principaldelen framför den reella delen. Följaktligen är lösningar från **solve()** på ekvationer som innehåller sådana rationella potenser inte nödvändigtvis deluppsättningar av lösningarna från **cSolve()**.

**cSolve()** börjar med exakta symboliska metoder. **cSolve()** använder om nödvändigt också iterativ ungefärlig komplex polynomfaktoruppdelning.

$$\begin{array}{l} \text{cSolve}(x^3=1,x) \\ x=\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \text{ or } x=-\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \text{ or } x=-1 \\ \hline \text{solve}(x^3=1,x) \qquad \qquad \qquad x=-1 \end{array}$$

$$\begin{array}{l} \text{cSolve}\left(x^{\frac{1}{3}}=1,x\right) \qquad \qquad \qquad \text{false} \\ \hline \text{solve}\left(x^{\frac{1}{3}}=1,x\right) \qquad \qquad \qquad x=-1 \end{array}$$

I läge Display Digits (Visa siffror) för Fix 2:

Obs: Se även **cZeros()**, **solve()** och **zeros()**.

$$\begin{aligned} & \text{exact}\left(\text{cSolve}\left(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3=0,x\right)\right) \\ & \quad x\cdot\left(x^4+4\cdot x^3+5\cdot x^2-6\right)=3 \\ & \text{cSolve}\left(\text{Ans},x\right) \\ & \quad x=-1.11+1.07\cdot i \text{ or } x=-1.11-1.07\cdot i \text{ or } x=2. \end{aligned}$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

**cSolve**(Eqn1 and Eqn2 [and...],  
VarOrGuess1, VarOrGuess2 [, ... ])  
⇒ Booleskt uttryck

**cSolve**(SystemOfEqns, VarOrGuess1,  
VarOrGuess2 [, ...]) ⇒ Booleskt uttryck

Ger möjliga komplexa lösningar på ekvationssystem där varje *VarOrGuess* specificerar en variabel som du vill lösa.

Du kan som alternativ specificera en initial gissning för en variabel. Varje *VarOrGuess* måste ha formen:

*variable*

– eller –

*variable = reellt eller icke-reellt tal*

Som exempel är  $x$  giltigt och likaså  $x=3+i$ .

Om alla ekvationer är polynom och om du INTE specificerar några initiala gissningar använder **cSolve()** eliminationsmetoden Gröbner/Buchberger för att försöka bestämma **alla** komplexa lösningar.

Komplexa lösningar kan innehålla både reella och icke-reella lösningar, som i exemplet till höger.

$$\begin{aligned} & \text{cSolve}\left(u\cdot v-u=v \text{ and } v^2=-u,\{u,v\}\right) \\ & \quad u=\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \text{ and } v=\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \text{ or } u=\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \end{aligned}$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

Ekvationssystem som innehåller polynom kan ha extra variabler som saknar värden, men representerar givna numeriska värden som kan ersättas senare.

Du kan också inkludera Lösningsvariabler som inte visas i ekvationerna. Dessa lösningar visar hur familjer av lösningar kan innehålla godtyckliga konstanter med formen  $ck$  där  $k$  är ett heltalssuffix från 1 till och med 255.

För polynomsystem kan beräkningstiden och användningen av minne i hög grad bero på i vilken ordning du listar Lösningsvariabler. Om ditt första val utarmar minnet, eller tar på ditt tålamod, kan du försöka med att arrangera om variablerna i ekvationerna och/eller i listan *VarOrGuess*.

Om du inte inkluderar några gissningar och om någon ekvation är ett icke-polynom i någon variabel, men alla ekvationer är linjära i alla Lösningsvariabler, använder **cSolve()** Gauss eliminationsmetod för att försöka bestämma alla lösningar.

Om ett system är varken polynomt i alla dess variabler eller linjärt i dess Lösningsvariabler bestämmer **cSolve()** högst en lösning med en ungefärlig iterativ metod. För att göra detta måste antalet Lösningsvariabler vara lika med antalet ekvationer och alla övriga variabler i ekvationerna måste förenklas till tal.

En icke-reell gissning är ofta nödvändig för att bestämma en icke-reell lösning. För konvergens kan en gissning behöva vara ganska nära en lösning.

$$\text{cSolve}\left(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{-(\sqrt{4 \cdot c - 1} \cdot i + 1)^2}{4} \text{ and } v = \frac{\sqrt{4 \cdot c - 1} \cdot i + 1}{2} \text{ or } \rightarrow$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c43 \text{ or } \rightarrow$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

$$\text{cSolve}\left(u + v = e^w \text{ and } u - v = i, \{u, v\}\right) \\ u = \frac{e^w + i}{2} \text{ and } v = \frac{e^w - i}{2}$$

$$\text{cSolve}\left(e^z = w \text{ and } w = z^2, \{w, z\}\right) \\ w = 0.494866 \text{ and } z = 0.703467$$

$$\text{cSolve}\left(e^z = w \text{ and } w = z^2, \{w, z = 1 + i\}\right) \\ w = 0.149606 + 4.8919 \cdot i \text{ and } z = 1.58805 + 1.5402 \cdot i \rightarrow$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

**CubicReg**  $X$ ,  $Y$ , [ $Freq$ ] [,  $Category$ ,  
 $Include$ ]

Utför en tredjegrads regressionsanalys =  $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$  på listorna  $X$  och  $Y$  med frekvensen  $Freq$ . En sammanfattning av resultaten visas i variabeln *stat.results*, på sidan 184.

Alla listor utom *Include* måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

$Freq$  är en frivillig lista på frekvensvärden. Varje element i  $Freq$  specificerar frekvensen för varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

$Category$  är en lista på kategorikoder för motsvarande  $X$ - och  $Y$ -data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel                  | Beskrivning                                                                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn                       | Regressionsekvation: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$                                                                                        |
| stat.a, stat.b, stat.c,<br>stat.d | Regressionskoefficienter                                                                                                                                |
| stat.R <sup>2</sup>               | Determinationskoefficient                                                                                                                               |
| stat.Resid                        | Residualer från regressionsanalysen                                                                                                                     |
| stat.XReg                         | Lista på datapunkter i den modifierade $X$ List som används i regressionen baserat på begränsningar i $Freq$ , $Category$ List och $Include$ Categories |
| stat.YReg                         | Lista på datapunkter i den modifierade $Y$ List som används i regressionen baserat på begränsningar i $Freq$ , $Category$ List och $Include$ Categories |
| stat.FreqReg                      | Lista på frekvenser som motsvarar $stat.XReg$ och $stat.YReg$                                                                                           |

## cumulativeSum()

Katalog > 

**cumulativeSum(List1)⇒lista**

|                          |            |
|--------------------------|------------|
| cumulativeSum({1,2,3,4}) | {1,3,6,10} |
|--------------------------|------------|

Ger en lista på de kumulativa summorna av elementen i *List1* och börjar med element 1.

**cumulativeSum(Matrix1)⇒matrix**

Ger en matrix över de kumulativa summorna av elementen i *Matrix1*.

Varje element är den kumulativa summan i kolumnen räknat uppifrån och ned.

|                                                                                                                    |                                                                                                         |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---|---|---|---|---|--------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|
| <table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr><tr><td>5</td><td>6</td></tr></table> → <i>m1</i> | 1                                                                                                       | 2 | 3 | 4 | 5 | 6 | <table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr><tr><td>5</td><td>6</td></tr></table> | 1 | 2 | 3 | 4 | 5 | 6 |
| 1                                                                                                                  | 2                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 3                                                                                                                  | 4                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 5                                                                                                                  | 6                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 1                                                                                                                  | 2                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 3                                                                                                                  | 4                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 5                                                                                                                  | 6                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| cumulativeSum( <i>m1</i> )                                                                                         | <table><tr><td>1</td><td>2</td></tr><tr><td>4</td><td>6</td></tr><tr><td>9</td><td>12</td></tr></table> | 1 | 2 | 4 | 6 | 9 | 12                                                                                                     |   |   |   |   |   |   |
| 1                                                                                                                  | 2                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 4                                                                                                                  | 6                                                                                                       |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 9                                                                                                                  | 12                                                                                                      |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |

Ett tomt element i *List1* eller *Matrix1* ger ett tomt element i den resulterande listan eller matrisen. För mer information om tomma element, se på sidan 261.

## Cycle

Katalog > 

### Cycle

Funktion som listar heltal från 1 till 100 utom 50.

Överför omedelbart kontroll till nästa iteration i den aktuella slingan (**For**, **While** eller **Loop**).

**Cycle** tillåts inte utanför de tre slingstrukturerna (**For**, **While** eller **Loop**).

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|                                      |      |
|--------------------------------------|------|
| Define <i>g()</i> =Func              | Done |
| Local <i>temp,i</i>                  |      |
| 0→ <i>temp</i>                       |      |
| For <i>i</i> ,1,100,1                |      |
| If <i>i</i> =50                      |      |
| Cycle                                |      |
| <i>temp</i> + <i>i</i> → <i>temp</i> |      |
| EndFor                               |      |
| Return <i>temp</i>                   |      |
| EndFunc                              |      |
| <i>g()</i>                           | 5000 |

## ►Cylind

Katalog > 

*Vector* ►**Cylind**

|                                                           |                                                                      |
|-----------------------------------------------------------|----------------------------------------------------------------------|
| $\begin{bmatrix} 2 & 2 & 3 \end{bmatrix}$ ► <b>Cylind</b> | $\begin{bmatrix} 2\sqrt{2} & \angle \frac{\pi}{4} & 3 \end{bmatrix}$ |
|-----------------------------------------------------------|----------------------------------------------------------------------|

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>**Cylind**.

Visar rad- eller kolumnvektorn i cylindrisk form [*r*,∠*θ*, *z*].

*Vector* måste ha exakt tre element. Den kan vara antingen en rad eller en kolumn.

## cZeros()

**cZeros(Expr, Var)⇒lista**

Ger en lista på möjliga reella och icke-reella värden på *Var* som gör *Expr=0*.

**cZeros()** gör detta genom att beräkna **explist(cSolve(Expr=0,Var),Var)**. Annars är **cZeros()** lika **zeros()**.

**Obs:** Se även **cSolve()**, **solve()** och **zeros()**.

**cZeros({Expr1, Expr2 [, ... ] },  
{VarOrGuess1,VarOrGuess2 [, ... ]  
})⇒matris**

Ger möjliga positioner där uttrycken är noll samtidigt. Varje *VarOrGuess* specificerar en okänd vars värde du söker.

Du kan som alternativ specificera en initial gissning för en variabel. Varje *VarOrGuess* måste ha formen:

*variable*

– eller –

*variable = reellt eller icke-reellt tal*

Som exempel är  $x$  giltigt och likaså  $x=3+i$ .

Om alla ekvationer är polynom och om du INTE specificerar några initiala gissningar använder **cZeros()** eliminationsmetoden Gröbner/Buchberger för att försöka bestämma alla komplexa nollställen.

I läge Display Digits (Visa siffror) för Fix 3:

$$\text{cZeros}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3,x)$$

$$\{-1.114+1.073\cdot i,-1.114-1.073\cdot i,-2.125,-0.612,i$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

$$\text{cZeros}(\text{conj}(z_-)-1-i,z_-) \quad \{1-i\}$$



Komplexa nollställen kan innehålla både reella och icke-reella nollställen, som i exemplet till höger.

Varje rad i den resulterande matrisen representerar ett alternativt nollställe, med komponenterna ordnade på samma sätt som i listan *VarOrGuess*. För att extrahera en rad, indexera matrisen med [row].

$$\text{cZeros}\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

Extrahera rad 2:

$$\text{Ans}[2] \quad \left[ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \quad \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \right]$$

System av polynom kan ha extra variabler som saknar värden, men representerar givna numeriska värden som kan ersättas senare.

Du kan också inkludera okända variabler som inte visas i uttrycken. Dessa nollställen visar hur familjer av nollställen kan innehålla godtyckliga konstanter med formen  $ck$ , där  $k$  är ett heltalssuffix från 1 till och med 255.

För polynomsystem kan beräkningstiden och användningen av minne i hög grad bero på i vilken ordning du listar okända. Om ditt första val utarmar minnet, eller tar på ditt tålamod, kan du försöka med att arrangera om variablerna i uttrycken och/eller i listan *VarOrGuess*.

Om du inte inkluderar några gissningar och om något uttryck är ett icke-polynom i någon variabel, men alla uttryck är linjära i alla okända, använder **cZeros()** Gauss eliminationsmetod för att försöka bestämma alla nollställen.

$$\text{cZeros}\left(\left\{u + v - e^w, u - v - i\right\}, \{u, v\}\right)$$

$$\begin{bmatrix} e^{w+i} & e^{w-i} \\ 2 & 2 \end{bmatrix}$$

Om ett system är varken polynomt i alla dess variabler eller linjärt i dess okända bestämmer **cZeros()** högst ett nollställe med en ungefärlig iterativ metod. För att göra detta måste antalet okända vara lika med antalet uttryck och alla övriga variabler i uttrycken måste förenklas till tal.

$$\text{cZeros}\left(\left\{e^z - w, w - z^2\right\}, \{w, z\}\right)$$

$$\begin{bmatrix} 0.494866 & -0.703467 \end{bmatrix}$$

En icke-reell gissning är ofta nödvändig för att bestämma ett icke-reellt nollställe. För konvergens kan en gissning behöva vara ganska nära ett nollställe.

$$cZeros(\{e^{-z-w}, w-z^2\}, \{w, z=1+i\})$$

$$[0.149606+4.8919\cdot i \quad 1.58805+1.54022\cdot i]$$

## D

**dbd(*date1*,*date2*)**⇒värde

Ger antalet dagar mellan *date1* och *date2* med dagräkningsmetoden.

*date1* och *date2* kan vara tal eller listor på tal inom den normala kalendern. Om både *date1* och *date2* är listor måste de vara lika långa.

*date1* och *date2* måste vara mellan 1950 och 2049.

Du kan mata in datumen i ett av två format. Decimalplaceringen skiljer sig mellan de två datumformaten.

MM.DDYY (format som ofta används i USA)

DDMM.YY (format som ofta används i Europa)

|                     |     |
|---------------------|-----|
| dbd(12.3103,1.0104) | 1   |
| dbd(1.0107,6.0107)  | 151 |
| dbd(3112.03,101.04) | 1   |
| dbd(101.07,106.07)  | 151 |

*Expr1* ►DD⇒värde

*List1* ►DD⇒lista

*Matrix1* ►DD⇒matris

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>DD.

Ger den decimala ekvivalenten till argumentet uttryckt i grader. Argumentet är ett tal, en lista eller en matris som tolkas av vinkelläget i nygrader, radianer eller grader.

I vinkelläget Grader:

|                                |                  |
|--------------------------------|------------------|
| (1.5°)►DD                      | 1.5°             |
| {45°22'14.3"}►DD               | 45.3706°         |
| { { 45°22'14.3",60°0'0" } }►DD | { 45.3706°,60° } |

I vinkelläget Nygrader:

|      |                        |
|------|------------------------|
| 1►DD | $\frac{9}{10}^{\circ}$ |
|------|------------------------|

I vinkelåget Radianer:

|          |          |
|----------|----------|
| (1.5)▶DD | 85.9437° |
|----------|----------|

## Decimal

Katalog > *Expression1* ▶Decimal⇒uttryck*List1* ▶Decimal⇒uttryck*Matrix1* ▶Decimal⇒uttryck

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>Decimal.

Visar argumentet i decimal form.  
Operatören kan endast användas i slutet av inmatningsraden.

|                        |          |
|------------------------|----------|
| $\frac{1}{3}$ ▶Decimal | 0.333333 |
|------------------------|----------|

## Define (Definiera)

Katalog > Define *Var* = *Uttryck*Define *Function*(*Param1*, *Param2*, ...) = *Uttryck*

Definierar variabeln *Var* eller den användardefinierade funktionen *Function*.

Parametrar såsom *Param1* utgör platshållare för att överföra argument till funktionen. När du anropar en användardefinierad funktion måste du ha argument (till exempel, värden eller variabler) som överensstämmer med parametrarna. När funktionen anropas beräknar den *Expression* med de givna argumenten.

*Var* och *Function* får inte vara namnet på en systemvariabel eller inbyggd funktion eller ett kommando.

|                                                           |      |
|-----------------------------------------------------------|------|
| Define $g(x,y)=2 \cdot x-3 \cdot y$                       | Done |
| $g(1,2)$                                                  | -4   |
| $1 \rightarrow a: 2 \rightarrow b: g(a,b)$                | -4   |
| Define $h(x)=\text{when}(x<2, 2 \cdot x-3, -2 \cdot x+3)$ | Done |
| $h(-3)$                                                   | -9   |
| $h(4)$                                                    | -5   |

**Obs:** Denna form av **Define** är ekvivalent med att exekvera uttrycket:

$expression \rightarrow Function$

$(Param1, Param2)$ .

**Define Function**(*Param1*, *Param2*, ...) = **Func**

*Block*

**EndFunc**

**Define Program**(*Param1*, *Param2*, ...) = **Prgm**

*Block*

**EndPrgm**

I denna form kan den användardefinierade funktionen eller programmet exekvera ett block av flera påståenden.

*Block* kan vara antingen ett enstaka påstående eller en serie av påståenden på separata rader. *Block* kan även inkludera uttryck och instruktioner (till exempel, **If**, **Then**, **Else** och **For**).

**Obs för att mata in exemplet:** Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

**Obs:** Se även **Define LibPriv**, på sidan 48 och **Define LibPub**, på sidan 49.

|                       |      |
|-----------------------|------|
| Define $g(x,y)$ =Func | Done |
| If $x > y$ Then       |      |
| Return $x$            |      |
| Else                  |      |
| Return $y$            |      |
| EndIf                 |      |
| EndFunc               |      |

|           |   |
|-----------|---|
| $g(3,-7)$ | 3 |
|-----------|---|

|                                      |      |
|--------------------------------------|------|
| Define $g(x,y)$ =Prgm                |      |
| If $x > y$ Then                      |      |
| Disp $x$ , " greater than ", $y$     |      |
| Else                                 |      |
| Disp $x$ , " not greater than ", $y$ |      |
| EndIf                                |      |
| EndPrgm                              |      |
|                                      | Done |

|           |                   |
|-----------|-------------------|
| $g(3,-7)$ |                   |
|           | 3 greater than -7 |
|           | Done              |

## Define LibPriv

**Define LibPriv** *Var* = *Uttryck*

**Define LibPriv Function**(*Param1*, *Param2*, ...) = *Uttryck*

**Define LibPriv Function**(*Param1*, *Param2*, ...) = **Func**

*Block*

**EndFunc**

**Define LibPriv** *Program*(*Param1*, *Param2*,  
...) = **Prgm**

*Block*

**EndPrgm**

Fungerar på samma sätt som **Define** förutom att en privat biblioteksvariabel, funktion eller program definieras. Privata funktioner och program visas inte i Katalogen.

**Obs:** Se även **Define**, på sidan 47 och **Define LibPub**, på sidan 49.

**Define LibPub**

**Define LibPub** *Var* = *Uttryck*

**Define LibPub** *Function*(*Param1*, *Param2*,  
...) = *Uttryck*

**Define LibPub** *Function*(*Param1*, *Param2*,  
...) = **Func**

*Block*

**EndFunc**

**Define LibPub** *Program*(*Param1*, *Param2*,  
...) = **Prgm**

*Block*

**EndPrgm**

Fungerar på samma sätt som **Define** förutom att en allmän biblioteksvariabel, funktion eller program definieras. Allmänna funktioner och program visas i Katalogen när biblioteket har sparats och uppdaterats.

**Obs:** Se även **Define**, på sidan 47 och **Define LibPriv**, på sidan 48.

**deltaList()**

Se **ΔList()**, på sidan 107.

DelVar

Katalog > 

|                                                                         |                   |           |
|-------------------------------------------------------------------------|-------------------|-----------|
| DelVar <i>Var1</i> [, <i>Var2</i> ] [, <i>Var3</i> ] ...                | $2 \rightarrow a$ | 2         |
| DelVar <i>Var</i> .                                                     | $(a+2)^2$         | 16        |
| Tar bort den specificerade variabeln eller variabelgruppen från minnet. | DelVar <i>a</i>   | Done      |
|                                                                         | $(a+2)^2$         | $(a+2)^2$ |

Om en eller flera variabler är låsta visar detta kommando ett felmeddelande och tar endast bort olåsta variabler. Se **unlock**, på sidan 206.

**DelVar** *Var*. tar bort alla led i variabelgruppen *Var*. variabelgrupp (till exempel, den statistiska *stat.nn*-resultaten eller variabler skapade med funktionen **LibShortcut()**). Punkten (.) i denna form av **DelVar**-kommandot begränsar kommandot till borttagning av en variabelgrupp: den enkla variabeln *Var* påverkas inte.

|                    |                                                                                                                                                                                                                          |             |       |     |     |             |       |     |     |             |       |     |     |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------|-----|-----|-------------|-------|-----|-----|-------------|-------|-----|-----|
| <i>aa.a:=45</i>    | 45                                                                                                                                                                                                                       |             |       |     |     |             |       |     |     |             |       |     |     |
| <i>aa.b:=5.67</i>  | 5.67                                                                                                                                                                                                                     |             |       |     |     |             |       |     |     |             |       |     |     |
| <i>aa.c:=78.9</i>  | 78.9                                                                                                                                                                                                                     |             |       |     |     |             |       |     |     |             |       |     |     |
| getVarInfo()       | <table><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"{"</td><td>"}"</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"{"</td><td>"}"</td></tr><tr><td><i>aa.c</i></td><td>"NUM"</td><td>"{"</td><td>"}"</td></tr></table> | <i>aa.a</i> | "NUM" | "{" | "}" | <i>aa.b</i> | "NUM" | "{" | "}" | <i>aa.c</i> | "NUM" | "{" | "}" |
| <i>aa.a</i>        | "NUM"                                                                                                                                                                                                                    | "{"         | "}"   |     |     |             |       |     |     |             |       |     |     |
| <i>aa.b</i>        | "NUM"                                                                                                                                                                                                                    | "{"         | "}"   |     |     |             |       |     |     |             |       |     |     |
| <i>aa.c</i>        | "NUM"                                                                                                                                                                                                                    | "{"         | "}"   |     |     |             |       |     |     |             |       |     |     |
| DelVar <i>aa</i> . | <i>Done</i>                                                                                                                                                                                                              |             |       |     |     |             |       |     |     |             |       |     |     |
| getVarInfo()       | "NONE"                                                                                                                                                                                                                   |             |       |     |     |             |       |     |     |             |       |     |     |

delVoid()

Katalog > 

|                                       |                       |         |
|---------------------------------------|-----------------------|---------|
| delVoid( <i>List1</i> )⇒ <i>lista</i> | delVoid({ 1,void,3 }) | { 1,3 } |
|---------------------------------------|-----------------------|---------|

Ger en lista med innehållet i *List1* med alla tomma element borttagna.

För mer information om tomma element, se på sidan 261.

derivative()

Se d(), på sidan 230.

**deSolve**(*1stOr2ndOrderODE*, *Var*, *depVar*)  $\Rightarrow$  en allmän lösning

Ger en ekvation som explicit eller implicit specificerar en generell lösning på den ordinära differentialekvationen (ODE) av 1:a eller 2:a ordningen. I ODE:

- Använd en primsymbol (tryck på ) för att beteckna förstaderivat av den beroende variabeln med avseende på den oberoende variabeln.
- Använd två primsymboler för att beteckna motsvarande andraderivata.

Primsymbolen används endast för derivata inom deSolve(). Använd **d()** i övriga fall.

Den generella lösningen på en ekvation av 1:a ordningen innehåller en godtycklig konstant med formen *ck*, där *k* är ett heltalssuffix från 1 till och med 255. Lösningen på en ekvation av 2:a ordningen innehåller två sådana konstanter.

Använd **solve()** på en implicit lösning om du vill försöka konvertera den till en eller flera ekvivalenta explicita lösningar.

När du jämför dina resultat med lösningar i läroböcker eller egna lösningar för hand, tänk på att olika metoder inför godtyckliga konstanter på olika ställen i beräkningen, vilket kan ge olika generella lösningar.

**deSolve**(*1stOrderODEandinitCond*, *Var*, *depVar*)  $\Rightarrow$  en partikulärlösning

Ger en partikulärlösning som uppfyller *1stOrderODE* och *initCond*. Detta är vanligen enklare än att bestämma en allmän lösning, ersätta initiala värden, lösa den godtyckliga konstanten och sedan ersätta värdet i den allmänna lösningen.

*initCond* är en ekvation med formen:

*depVar* (*initialIndependentValue*) = *initialDependentValue*

$$\begin{aligned} \text{deSolve}(y''+2\cdot y'+y &= x^2, x, y) \\ y &= (c3\cdot x + c4) \cdot e^{-x} + x^2 - 4\cdot x + 6 \\ \text{right}(Ans) \rightarrow \text{temp} & \quad (c3\cdot x + c4) \cdot e^{-x} + x^2 - 4\cdot x + 6 \\ \frac{d^2}{dx^2}(\text{temp}) + 2 \cdot \frac{d}{dx}(\text{temp}) + \text{temp} &= x^2 \quad 0 \\ \text{DelVar temp} & \quad Done \end{aligned}$$

$$\begin{aligned} \text{deSolve}(y' &= (\cos(y))^2 \cdot x, x, y) \quad \tan(y) = \frac{x^2}{2} + c4 \\ \text{solve}(Ans, y) & \quad y = \tan^{-1}\left(\frac{x^2 + 2\cdot c4}{2}\right) + n3 \cdot \pi \\ Ans[c4 = c-1 \text{ and } n3 = 0] & \quad y = \tan^{-1}\left(\frac{x^2 + 2\cdot (c-1)}{2}\right) \\ \sin(y) &= (y \cdot e^x + \cos(y)) \cdot y' \rightarrow ode \\ \sin(y) &= (e^x \cdot y + \cos(y)) \cdot y' \\ \text{deSolve}(ode \text{ and } y(0) &= 0, x, y) \rightarrow soln \\ \frac{-(2 \cdot \sin(y) + y^2)}{2} &= (e^x - 1) \cdot e^{-x} \cdot \sin(y) \\ soln|x=0 \text{ and } y=0 & \quad true \\ ode|y' = \text{impDif}(soln, x, y) & \quad true \\ \text{DelVar ode, soln} & \quad Done \end{aligned}$$

*initialIndependentValue* och *initialDependentValue* kan vara variabler såsom *x0* och *y0* som saknar lagrade värden. Implicit derivering kan underlätta verifieringen av implicita lösningar.

**deSolve**

**(2ndOrderODEandinitCond1and  
initCond2, Var, depVar)⇒en  
partikulärlösning**

Ger en partikulärlösning som uppfyller 2nd Order ODE och har ett specificerat värde på den beroende variabeln och dess förstaderivata i en punkt.

För *initCond1*, använd formen:

*depVar* (*initialIndependentValue*) =  
*initialDependentValue*

För *initCond2*, använd formen:

*depVar* (*initialIndependentValue*) =  
*initial1stDerivativeValue*

**deSolve**

**(2ndOrderODEandbndCond1and  
bndCond2, Var, depVar)⇒en  
partikulärlösning**

Ger en partikulärlösning som uppfyller 2ndOrderODE och har specificerade värden i två punkter.

$$\text{deSolve}\left(y''=y^{\frac{-1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0, t, y\right)$$

$$\frac{\frac{3}{2} \cdot y^{\frac{4}{2}}}{3} = t$$

$$\text{deSolve}(y''=x \text{ and } y(0)=1 \text{ and } y'(2)=3, x, y)$$

$$y=\frac{x^3}{6}+x+1$$

$$\text{deSolve}(y''=2 \cdot y' \text{ and } y(3)=1 \text{ and } y'(4)=2, x, y)$$

$$y=e^{2 \cdot x-8}-e^{-2}+1$$

$$\text{deSolve}\left(w''-\frac{2 \cdot w'}{x}+\left(9+\frac{2}{x^2}\right) \cdot w=x \cdot e^x \text{ and } w\left(\frac{\pi}{6}\right)=0 \text{ and } w\left(\frac{\pi}{3}\right)=0, x, w\right)$$

$$w=\frac{x \cdot e^x}{(\ln(e))^2+9}+e^{\frac{3}{x}} \cdot x \cdot \cos(3 \cdot x)-e^{\frac{\pi}{6}} \cdot x \cdot \sin(3 \cdot x)$$



**det()**Katalog > 

**det**(*squareMatrix*[,  
*Tolerance*])⇒uttryck

Ger determinanten för *squareMatrix*.

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tolerance*. Denna tolerans används endast om matrisen har inmatning i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tolerance*.

- Om du använder   eller ställer in **Auto** eller **Ungefärlig** på Approximate utförs beräkningarna med flyttalsaritmetik.
- Om *Tolerance* utelämnas eller inte används beräknas standardtoleransen som:

$5E-14 \cdot \max(\text{dim}(\text{squareMatrix})) \cdot \text{rowNorm}(\text{squareMatrix})$

|                                                                                                                       |                                                            |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| $\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$                                                       | $a \cdot d - b \cdot c$                                    |
| $\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$                                                       | -2                                                         |
| $\det\left(\text{identity}(3) - x \cdot \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{bmatrix}\right)$ | $-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$          |
| $\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1}$                                    | $\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$ |
| $\det(\text{mat1})$                                                                                                   | 0                                                          |
| $\det(\text{mat1}, 1)$                                                                                                | $1. \text{E}20$                                            |

**diag()**Katalog > 

**diag**(*List*)⇒*matrix*

**diag**(*rowMatrix*)⇒*matrix*

**diag**(*columnMatrix*)⇒*matrix*

Ger en matris med värdena i argumentlistan eller matrisen i dess huvuddiagonal.

**diag**(*squareMatrix*)⇒*radMatrix*

Ger en radmatris som innehåller elementen från huvuddiagonalen hos *squareMatrix*.

*squareMatrix* måste vara kvadratisk.

|                                                                     |                                                                     |
|---------------------------------------------------------------------|---------------------------------------------------------------------|
| $\text{diag}([2 \ 4 \ 6])$                                          | $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$ |
| $\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$ | $\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$ |
| $\text{diag}(\text{Ans})$                                           | $\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$                           |

**dim()**Katalog > 

**dim**(*List*)⇒*heltal*

|                           |   |
|---------------------------|---|
| $\text{dim}(\{0, 1, 2\})$ | 3 |
|---------------------------|---|

## dim()

Katalog > 

Ger dimensionen på *List*.

**dim(Matrix)**⇒*lista*

Ger dimensionerna på en matris som en lista med två element {rader, kolumner}.

**dim(String)**⇒*helta*

Ger antalet tecken i teckensträngen *String*.

|                                                                |           |
|----------------------------------------------------------------|-----------|
| $\dim \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$ | $\{3,2\}$ |
| $\dim("Hello")$                                                | 5         |
| $\dim("Hello "&"there")$                                       | 11        |

## Disp

Katalog > 

**Disp exprOrString1 [, exprOrString2] ...**

Visar argumenten i *Calculator*-historiken. Argumenten visas i ordningsföljd med utslutning som separatorer.

Huvudsakligen användbart i program och funktioner för att säkerställa visningen av mellanliggande beräkningar.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|                              |       |
|------------------------------|-------|
| Define chars(start,end)=Prgm |       |
| For i,start,end              |       |
| Disp i," ",char(i)           |       |
| EndFor                       |       |
| EndPrgm                      |       |
|                              | Done  |
| chars(240,243)               |       |
|                              | 240 ð |
|                              | 241 ñ |
|                              | 242 ò |
|                              | 243 ó |
|                              | Done  |

## DispAt

Katalog > 

**DispAt int,expr1 [,expr2 ...] ...**

**DispAt** låter dig specificera den rad där det specifika uttrycket eller strängen kommer att visas på skärmen.

Radnumret kan specificeras som ett uttryck.

Observera att radnumret inte gäller för hela skärmen utan för det område som direkt följer kommandot/programmet.

DispAt

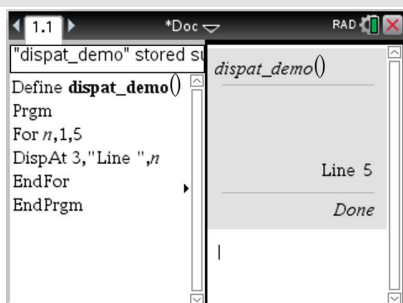
### Exempel

|                       |      |                |
|-----------------------|------|----------------|
| 1.1                   | *Doc | RAD            |
| disp_at_demo 2/3      |      |                |
| Define disp_at_demo() |      | disp_at_demo() |
| Prgm                  |      | Line 1         |
| For n,1,5             |      | Line 2         |
| DispAt n,"Line ",n    |      | Line 3         |
| EndFor                |      | Line 4         |
| EndPrgm               |      | Line 5         |
|                       |      | Done           |

Detta kommando möjliggör kontrollpanelsliknande utdata från program där värdet av ett uttryck eller från en sensoravläsning uppdateras på samma rad.

**DispAtoch Disp** kan användas i samma program.

**Obs:** Maxnummer är inställt på 8 eftersom detta motsvarar en skärm fylld med rader på en skärm hos en handenhet – så länge raderna inte har matematiska uttryck i 2D. Det exakta antalet rader beror på innehållet i den information som visas.



#### Illustrativa exempel:

|                  |              |
|------------------|--------------|
| Define z(=       | Utdata       |
| Prgm             | z()          |
| For n,1,3        | Iteration 1: |
| DispAt 1,"N: ",n | Rad 1: N:1   |
| Disp "Hej"       | Rad 2: Hej   |
| EndFor           | Iteration 2: |
| EndPrgm          | Rad 1: N:2   |
|                  | Rad 2: Hej   |
|                  | Rad 3: Hej   |
|                  | Iteration 3: |
|                  | Rad 1: N:3   |
|                  | Rad 2: Hej   |
|                  | Rad 3: Hej   |
|                  | Rad 4: Hej   |
| Define z1(=      | z1()         |
| Prgm             | Rad 1: N:3   |
| For n,1,3        | Rad 2: Hej   |
| DispAt 1,"N: ",n | Rad 3: Hej   |
| EndFor           | Rad 4: Hej   |
|                  | Rad 5: Hej   |
| For n,1,4        |              |
| Disp "Hej"       |              |
| EndFor           |              |
| EndPrgm          |              |

## Feltillstånd:

| Felmeddelande                                           | Beskrivning                                                                                                                                                                                                  |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Radnummer för DispAt måste vara mellan 1 och 8          | Uttryck utvärderar radnummer utanför intervallet 1-8 (inklusive)                                                                                                                                             |
| Too few arguments (För få argument)                     | Funktionen eller kommandot saknar ett eller flera argument.                                                                                                                                                  |
| Inga argument                                           | Samma som nuvarande dialogruta för "syntaxfel" dialog                                                                                                                                                        |
| Too many arguments (För många argument)                 | Begränsa argument. Samma fel som Disp.                                                                                                                                                                       |
| Invalid data type (Ogiltig datatyp)                     | Första argumentet måste vara ett tal.                                                                                                                                                                        |
| Tom: DispAt tom                                         | Datatypfelet "Hello World" kastas bort (om återanrop definieras)                                                                                                                                             |
| Konverteringsoperator: DispAt 2_ft @> _m, "Hello World" | <b>CAS:</b> Datatypfel kastas bort (om återanrop definieras)<br><b>Numeriska:</b> Konvertering kommer att utvärderas och om resultatet är ett giltigt argument, skriver DispAt ut strängen på resultatraden. |

## ►DMS

Katalog > *Expr* ►DMS

I vinkelläget Grader:

*List* ►DMS $\{45.371\}$  ►DMS  $45^{\circ}22'15.6''$ *Matrix* ►DMS $\{\{45.371,60\}\}$  ►DMS  $\{45^{\circ}22'15.6'',60^{\circ}\}$ 

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>DMS.

Tolkar argumentet som en vinkel och visar motsvarande DMS-värde (DDDDDD°MM'SS.ss"). Se °, ', ", på sidan 238 för DMS-format (grad, minuter, sekunder).

**Obs:** ►DMS konverterar från radianer till grader vid användning i läget radianer. Om inmatningen följs av en gradsymbol ° sker ingen konvertering. Du kan bara använda ►DMS i slutet av en inmatningsrad.

**domain() (område)**Katalog > **domain(Uttr1, Var)⇒uttryck**

Ger området *Uttr1* med avseende på *Var*.

**domain()** kan användas för att undersöka områden hos funktioner. Det är begränsat till reella och ändliga områden.

Denna funktionalitet har begränsningar på grund av brister i förenkling av datoriserad algebra och algoritmlösare.

Vissa funktioner kan inte användas som argument för **domain()**, oavsett om de visas explicit eller inom användardefinierade variabler och funktioner. I följande exempel kan inte uttrycket förenklas eftersom  $\int()$  är en otillåten funktion.

$$\text{domain}\left(\begin{bmatrix} x \\ \frac{1}{t} \, dt, x \\ 1 \end{bmatrix}\right) \rightarrow \text{domain}\left(\begin{bmatrix} x \\ \frac{1}{t} \, dt, x \\ 1 \end{bmatrix}\right)$$

$$\begin{array}{ll} \text{domain}\left(\frac{1}{x+y}, y\right) & -\infty < y < -x \text{ or } -x < y < \infty \\ \text{domain}\left(\frac{x+1}{x^2+2 \cdot x}, x\right) & x \neq -2 \text{ and } x \neq 0 \\ \text{domain}\left(\left(\sqrt{x}\right)^2, x\right) & 0 \leq x < \infty \\ \text{domain}\left(\frac{1}{x+y}, y\right) & -\infty < y < -x \text{ or } -x < y < \infty \end{array}$$

**dominantTerm()**Katalog > **dominantTerm(Expr1, Var [, Point])⇒uttryck**

**dominantTerm(Expr1, Var [, Point]) | Var>Point ⇒uttryck**

**dominantTerm(Expr1, Var [, Point]) | Var<Point ⇒uttryck**

$$\begin{array}{ll} \text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x) & \frac{x^7}{30} \\ \text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right) & \frac{1}{2 \cdot (x-1)} \\ \text{dominantTerm}\left(x^{-2} \cdot \tan\left(x^{\frac{1}{3}}\right), x\right) & \frac{1}{x^3} \\ \text{dominantTerm}(\ln(x^x - 1) \cdot x^{-2}, x) & \frac{\ln(x \cdot \ln(x))}{x^2} \end{array}$$

Ger den dominanta termen i en potensserierepresentation av *Expr1* expanderat kring *Point*. Den dominanta termen är den term som snabbast ökar i storlek nära *Var = Point*. Den resulterande potensen av (*Var - Point*) kan ha en negativ och/eller exponent i bråkform. Koefficienten för denna potens kan inkludera logaritmer av (*Var - Point*) och andra funktioner av *Var* som domineras av alla potenser av (*Var - Point*) som har samma exponenttecken.

*Point* förinställs till 0. *Point* kan vara  $\infty$  eller  $-\infty$ , varvid den dominanta termen kommer att vara den term som har den största exponenten av *Var* snarare än den minsta exponenten av *Var*.

**dominantTerm(...)** ger "dominantTerm(...)" om den inte kan bestämma en sådan representation, till exempel, för essentiella singulariteter såsom  $\sin(1/z)$  vid  $z=0$ ,  $e^{-1/z}$  vid  $z=0$ , eller  $e^z$  vid  $z = \infty$  eller  $-\infty$ .

Om serien eller någon av dess derivator har en språngdiskontinuitet vid *Point* innehåller resultatet troligen deluttryck i formen  $\text{sign}(\dots)$  eller  $\text{abs}(\dots)$  för en reell expansionsvariabel, eller  $(-1)^{\text{floor}(\dots \text{angle}(\dots))}$  för en komplex expansionsvariabel, vilken slutar med "\_". Om du tänker använda den dominanta termen endast för värden på ena sidan av *Point*, bifoga då till **dominantTerm(...)** den lämpliga av " $| \text{Var} > \text{Point}$ ", " $| \text{Var} < \text{Point}$ ", " $| \text{Var} \geq \text{Point}$ " eller " $\text{Var} \leq \text{Point}$ " för att erhålla ett enklare resultat.

**dominantTerm()** fördelas över 1:a-argumentlistor och matriser.

|                                                                               |                                                          |
|-------------------------------------------------------------------------------|----------------------------------------------------------|
| $\text{dominantTerm}\left(e^{\frac{-1}{z}}, z\right)$                         | $\text{dominantTerm}\left(e^{\frac{-1}{z}}, z, 0\right)$ |
| $\text{dominantTerm}\left(\left(1 + \frac{1}{n}\right)^n, n, \infty\right)$   | $e$                                                      |
| $\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 0\right)$     | $\frac{\pi \cdot \text{sign}(x)}{2}$                     |
| $\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, x > 0\right)$ | $\frac{\pi}{2}$                                          |

**dominantTerm()** kan användas när du vill veta det enklaste uttrycket som är asymptotiskt med ett annat uttryck såsom  $Var \rightarrow Point$ . **dominantTerm()** kan också användas när det inte är uppenbart vilken grad den första icke-nolltermen i en serie kommer att ha, och du inte vill gissa varken iterativt eller interaktivt eller använda en programslinga.

**Obs:** Se även **series()**, på sidan 167.

## dotP()

Katalog > 

**dotP(List1, List2)⇒uttryck**

Ger "prick"-produkten av två listor.

|                                    |                                     |
|------------------------------------|-------------------------------------|
| $\text{dotP}(\{a,b,c\},\{d,e,f\})$ | $a \cdot d + b \cdot e + c \cdot f$ |
| $\text{dotP}(\{1,2\},\{5,6\})$     | 17                                  |

**dotP(Vector1, Vector2)⇒uttryck**

Ger "prick"-produkten av två vektorer.

|                                        |                                     |
|----------------------------------------|-------------------------------------|
| $\text{dotP}([a \ b \ c],[d \ e \ f])$ | $a \cdot d + b \cdot e + c \cdot f$ |
| $\text{dotP}([1 \ 2 \ 3],[4 \ 5 \ 6])$ | 32                                  |

Båda måste vara radvektorer eller båda måste vara kolumnvektorer.

## E

## e^()

-tangent

**e^(Expr1)⇒uttryck**

Ger **e** upphöjt till potensen *Expr1*.

|           |         |
|-----------|---------|
| $e^1$     | $e$     |
| $e^{1.}$  | 2.71828 |
| $e^{3^2}$ | $e^9$   |

**Obs:** Se även **e exponentmall**, på sidan 2.

**Obs:** Att trycka på  för att visa  $e^()$  skiljer sig från att trycka på tecknet **E** på tangentbordet.

Du kan skriva in ett komplext tal i den polära formen  $re^{i\theta}$ . Använd dock denna form endast i vinkelläget Radianer: den orsakar ett områdesfel i vinkelläget Grader eller Nygrader.

**e^(List1)⇒lista**

Ger **e** upphöjt till potensen för varje element i *List1*.

|                |                         |
|----------------|-------------------------|
| $\{1,1.,0.5\}$ | $\{e,2.71828,1.64872\}$ |
|----------------|-------------------------|

**e^()****e<sup>x</sup>-tangent****e<sup>(squareMatrix1)</sup>⇒kvadratMatris**

Ger matrisen med exponenten för *squareMatrix1*. Detta är inte detsamma som att beräkna e upphöjt till potensen för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

|                                                                        |                                                                                                                           |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| $e \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$ | $\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$ |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|

**eff()****Katalog >****eff(nominalRate,CpY)⇒värde**

eff(5.75,12)

5.90398

Finansiell funktion som konverterar den nominella räntan *nominalRate* till en årlig effektiv ränta, given av *CpY* som antalet ränteperioder per år.

*nominalRate* måste vara ett reellt tal och *CpY* måste vara ett reellt tal > 0.

**Obs:** Se även **nom()**, på sidan 128.

**eigVc()****Katalog >****eigVc(squareMatrix)⇒matris**

I Rektangulärt komplext format:

Ger en matris som innehåller egenvektorer för en reell eller komplex *squareMatrix* där varje kolumn i resultatet motsvarar ett egenvärde. Observera att en egenvektor inte är unik: den kan vara skalad med vilken konstant faktor som helst. Egenvektorer är normaliserade, vilket innebär att om  $V = [x_1, x_2, \dots, x_n]$ , så är:

$$x_1^2 + x_2^2 + \dots + x_n^2 = 1$$

|                                                                                       |                                                                        |
|---------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| $\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$ |
|---------------------------------------------------------------------------------------|------------------------------------------------------------------------|

eigVc(m1)

|                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\begin{bmatrix} -0.800906 & 0.767947 & ( \\ 0.484029 & 0.573804+0.052258 \cdot i & 0.5738 \cdot \\ 0.352512 & 0.262687+0.096286 \cdot i & 0.2626 \end{bmatrix}$ |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.



*squareMatrix* balanseras först med liknande transformationer tills rad- och kolumnnormerna har så nära samma värde som möjligt. *squareMatrix* reduceras sedan till övre Hessenberg-form och egenvektorer beräknas med en Schur-faktorisering.

**eigVl(*squareMatrix*)** ⇒ lista

Ger en lista på egenvärdena för en reell eller komplex *squareMatrix*.

*squareMatrix* balanseras först med likhetstransformationer tills rad- och kolumnnormerna har så nära samma värde som möjligt. *squareMatrix* reduceras sedan till övre Hessenberg-form och egenvektorer beräknas från den övre Hessenberg-matrisen.

I läget Rektangulärt komplext format:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mI \qquad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

$$\text{eigVl}(mI) \\ \{-4.40941, 2.20471 + 0.763006 \cdot i, 2.20471 - 0.763006 \cdot i\}$$

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

**If *BooleanExpr1* Then**

*Block1*

**ElseIf *BooleanExpr2* Then**

*Block2*

⋮

**ElseIf *BooleanExprN* Then**

*BlockN*

**EndIf**

⋮

Define  $g(x)$  = Func

If  $x \leq -5$  Then

Return 5

ElseIf  $x > -5$  and  $x < 0$  Then

Return  $-x$

ElseIf  $x \geq 0$  and  $x \neq 10$  Then

Return  $x$

ElseIf  $x = 10$  Then

Return 3

EndIf

EndFunc

*Done*

**ElseIf**Katalog > 

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

**EndFor**

Se For, på sidan 75.

**EndFunc**

Se Func, på sidan 79.

**EndIf**

Se If, på sidan 90.

**EndLoop**

Se Loop, på sidan 115.

**EndPrgm**

Se Prgm, på sidan 143.

**EndTry**

Se Try, på sidan 199.

**EndWhile**

Se While, på sidan 209.

**euler ()**Katalog > 

**euler**(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, eulerStep]) ⇒ *matris*

Differentiallikvation:

 $y' = 0,001 * y * (100 - y)$  och  $y(0) = 10$ 

**euler**(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep [,

*eulerStep()*  $\Rightarrow$  *matris*

**euler**(*ListOfExpr*, *Var*, *ListOfDepVars*,  
{*Var0*, *VarMax*}**x** *ListOfDepVars0*,  
*VarStep* [, *eulerStep*])  $\Rightarrow$  *matris*

Använder Eulers metod för att lösa systemet

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

med *depVar*(*Var0*)=*depVar0* på intervallet [*Var0*,*VarMax*]. Ger en matris vars första rad definierar resultatvärdena för *Var* och vars andra rad definierar den första lösningskomponenten vid motsvarande *Var*-värden, och så vidare.

*Expr* är det högra ledet som definierar den ordinära differentialekvationen (ODE).

*SystemOfExpr* är systemet av högerled som definierar systemet av ODE:er (motsvarar ordningen av oberoende variabler i *ListOfDepVars*).

*ListOfExpr* är en lista på högerled som definierar systemet av ODE:er (motsvarar ordningen av oberoende variabler i *ListOfDepVars*).

*Var* är den oberoende variabeln.

*ListOfDepVars* är en lista på oberoende variabler.

{*Var0*, *VarMax*} är en lista med två element som instruerar funktionen att integrera från *Var0* till *VarMax*.

*ListOfDepVars0* är en lista på startvärden för oberoende variabler.

$$\text{euler}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right) \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix}$$

För att se hela resultatet, tryck på  och använd sedan  och  för att flytta markören.

Jämför ovanstående resultat med CAS exakta lösning som erhållits med *deSolve()* och *seqGen()*:

$$\text{deSolve}\left(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\right) \\ y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}.$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right) \\ \{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

Ekvationssystem:

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

med *y1*(0)=2 och *y2*(0)=5

$$\text{euler}\left(\begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right) \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix}$$

*VarStep* är ett tal skilt från noll så att  $\text{sign}(\text{VarStep}) = \text{sign}(\text{VarMax} - \text{Var0})$  och lösningar ges vid  $\text{Var0} + i \cdot \text{VarStep}$  för alla  $i=0,1,2,\dots$  sådana att  $\text{Var0} + i \cdot \text{VarStep}$  är i intervallet  $[\text{var0}, \text{VarMax}]$  (det kanske inte finns ett lösningsvärde vid *VarMax*).

*eulerStep* är ett positivt heltal (förinställs på 1) som definierar antalet euler-steg mellan resultatvärden. Den verkliga stegstorleken som används av euler-metoden är  $\text{VarStep} / \text{eulerStep}$ .

## eval ()

## Hubb-meny

**eval**(*Expr*)  $\Rightarrow$  *string*

**eval()** är giltig endast i TI-Innovator™ Hub Kommandoargument i programmeringskommandona **Get**, **GetStr** och **Send**. Programmet beräknar uttrycket *Expr* och ersätter **eval()**-meddelandet med resultatet som en teckensträng.

Argumentet *Expr* måste generera ett reellt tal.

Ställ in den blå komponenten hos RGB-lysdioden på halv intensitet.

|                                               |     |
|-----------------------------------------------|-----|
| <i>lum</i> :=127                              | 127 |
| Send "SET COLOR.BLUE eval( <i>lum</i> )" Done |     |

Återställ den blå komponenten till OFF.

|                                |
|--------------------------------|
| Send "SET COLOR.BLUE OFF" Done |
|--------------------------------|

Argumentet **eval()** måste generera ett reellt tal.

|                                |
|--------------------------------|
| Send "SET LED eval("4") TO ON" |
| "Error: Invalid data type"     |

Program för att tona in den röda komponenten

|                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Define <b>fadein()</b> =<br>Prgm<br>For <i>i</i> ,0,255,10<br>Send "SET COLOR.RED eval( <i>i</i> )"<br>Wait 0.1<br>EndFor<br>Send "SET COLOR.RED OFF"<br>EndPrgm |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Kör programmet.

|                 |      |
|-----------------|------|
| <i>fadein()</i> | Done |
|-----------------|------|

## eval ()

## Hubb-meny

Även om **eval()** inte visar dess resultat kan du visa den resulterande Hubb-kommandosträngen efter att ha kört kommandot genom att kontrollera någon av följande speciella variabler.

*iostr.SendAns*  
*iostr.GetAns*  
*iostr.GetStrAns*

**Obs:** Se även **Get** (på sidan 81), **GetStr** (på sidan 88), och **Send** (på sidan 164).

|                                         |                            |
|-----------------------------------------|----------------------------|
| $n:=0.25$                               | 0.25                       |
| $m:=8$                                  | 8                          |
| $n \cdot m$                             | 2.                         |
| Send "SET COLOR.BLUE ON TIME eval(n·m)" | Done                       |
| <i>iostr.SendAns</i>                    | "SET COLOR.BLUE ON TIME 2" |

## exact()

## Katalog >

**exact(Expr1 [, Tolerance])** ⇒ uttryck

**exact(List1 [, Tolerance])** ⇒ lista

**exact(Matrix1 [, Tolerance])** ⇒ matris

Använder det aritmetiska läget Exact för att, när så är möjligt, ge argumentet som ett rationellt tal.

*Tolerance* specificerar konverteringens tolerans: förinställningen är 0 (noll).

|                                  |                                                              |
|----------------------------------|--------------------------------------------------------------|
| <b>exact(0.25)</b>               | $\frac{1}{4}$                                                |
| <b>exact(0.333333)</b>           | $\frac{333333}{1000000}$                                     |
| <b>exact(0.333333,0.001)</b>     | $\frac{1}{3}$                                                |
| <b>exact(3.5·x+y)</b>            | $\frac{7 \cdot x}{2} + y$                                    |
| <b>exact({ 0.2,0.33,4.125 })</b> | $\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$ |

## Exit

## Katalog >

### Exit

Avslutar det aktuella blocket **For**, **While** eller **Loop**.

**Exit** tillåts inte utanför de tre slingstrukturerna (**For**, **While** eller **Loop**).

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

### Funktionslista:

|                           |      |
|---------------------------|------|
| Define $g()$ = Func       | Done |
| Local <i>temp,i</i>       |      |
| $0 \rightarrow temp$      |      |
| For <i>i</i> ,1,100,1     |      |
| $temp+i \rightarrow temp$ |      |
| If $temp > 20$ Then       |      |
| Exit                      |      |
| EndIf                     |      |
| EndFor                    |      |
| EndFunc                   |      |
| $g()$                     | 21   |

*Expr* ►exp

Representerar *Expr* i termer av basen för den naturliga logaritmen  $e$ . Detta är en omvandlingsoperator för visning. Den kan endast användas i slutet av inmatningsraden.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>exp.

|                              |                    |
|------------------------------|--------------------|
| $\frac{d}{dx}(e^x + e^{-x})$ | $2 \cdot \sinh(x)$ |
| $2 \cdot \sinh(x)$ ►exp      | $e^x - e^{-x}$     |

exp()

-tangent

exp(*Expr1*) ⇒ uttryck

Ger  $e$  upphöjt till potensen *Expr1*.

**Obs:** Se även  $e$  exponentmall, på sidan 2.

Du kan skriva in ett komplext tal i den polära formen  $re^{i\theta}$ . Använd dock denna form endast i vinkelläget Radianer: den orsakar ett områdesfel i vinkelläget Grader eller Nygrader.

exp(*List1*) ⇒ lista

Ger  $e$  upphöjt till potensen för varje element i *List1*.

exp(*squareMatrix1*) ⇒ kvadratMatris

Ger matrisen med exponenten för *squareMatrix1*. Detta är inte detsamma som att beräkna  $e$  upphöjt till potensen för varje element. Se cos() för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

|           |         |
|-----------|---------|
| $e^1$     | $e$     |
| $e^{1.}$  | 2.71828 |
| $e^{3^2}$ | $e^9$   |

|                    |                           |
|--------------------|---------------------------|
| $e^{\{1,1.,0.5\}}$ | $\{e, 2.71828, 1.64872\}$ |
|--------------------|---------------------------|

|                                                                          |                                                                                                                           |
|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| $e^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}}$ | $\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$ |
|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|

exp►list()

Katalog > 

exp►list(*Expr*,*Var*) ⇒ lista

|                                                                                   |                            |
|-----------------------------------------------------------------------------------|----------------------------|
| $\text{solve}(x^2 - x - 2 = 0, x)$                                                | $x = -1 \text{ or } x = 2$ |
| $\text{exp} \blacktriangleright \text{list}(\text{solve}(x^2 - x - 2 = 0, x), x)$ | $\{-1, 2\}$                |

Undersöker *Expr* för ekvationer som är separerade med ordet "eller," och ger en lista med de högra sidorna av ekvationerna med formen *Var=Expr*. Detta ger dig en enkel metod att extrahera vissa lösningsvärden i resultaten från funktionerna **solve()**, **cSolve()**, **fMin()** och **fMax()**.

**Obs:** **exp►list()** behövs inte med funktionerna **zeros** och **cZeros()** eftersom de direkt ger en lista på lösningsvärden.

Du kan infoga denna funktion med datorns tangentbord genom att skriva **exp@>list(...)**.

**expand()**

**expand(Expr1 [, Var])**⇒uttryck

**expand(List1 [,Var])**⇒lista

**expand(Matrix1 [,Var])**⇒matrix

**expand(Expr1)** ger *Expr1* utvecklat med avseende på alla dess variabler. Utvecklingen är en polynomutveckling för polynom och partiell bråkutveckling för rationella uttryck.

Målsättningen med **expand()** är att transformera *Expr1* till en summa av och/eller skillnad mellan enkla termer. Som kontrast är målsättningen med **factor()** att transformera *Expr1* till en produkt av och/eller kvot mellan enkla faktorer.

$$\begin{array}{l} \text{expand}\left((x+y+1)^2\right) \\ \hline x^2+2\cdot x\cdot y+2\cdot x\cdot y^2+2\cdot y+1 \\ \hline \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \\ \hline \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y-1}-\frac{1}{y} \end{array}$$

**expand(Expr1, Var)** ger *Expr1* utvecklat med avseende på *Var*. Liknande potenser av *Var* samlas in. Termerna och deras faktorer sorteras med *Var* som huvudvariabel. Viss tillfällig faktorisering eller utveckling kan ske av de insamlade koefficienterna. Jämfört med att utesluta *Var* sparar detta ofta tid, minne och skärmutrymme, vilket gör uttrycket mer begripligt.

Även med endast en variabel kan användning av *Var* göra den faktorisering av nämnare som används för partialbråksutveckling mer fullständig.

Tips: För rationella uttryck är **propFrac()** ett snabbare, men mindre extremt, alternativ till **expand()**.

**Obs:** Se även **comDenom()** för en expanderad täljare genom en expanderad nämnare.

**expand(Expr1, [Var])** fördelar också logaritmer och rationella potenser oberoende av *Var*. För ökad fördelning av logaritmer och rationella potenser kan olikhetsbegränsningar vara nödvändiga för att säkerställa att vissa faktorer är naturliga tal.

**expand(Expr1, [Var])** fördelar också absolutvärden, **sign()** och exponentialer oberoende av *Var*.

**Obs:** Se även **tExpand()** för utveckling av trigonometriska uttryck.

$$\begin{array}{l} \text{expand}\left((x+y+1)^2, y\right) \quad y^2+2\cdot y\cdot (x+1)+(x+1)^2 \\ \text{expand}\left((x+y+1)^2, x\right) \quad x^2+2\cdot x\cdot (y+1)+(y+1)^2 \\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}, y\right) \\ \frac{1}{y-1}-\frac{1}{y}+\frac{1}{x\cdot (x-1)} \\ \text{expand}(Ans, x) \quad \frac{1}{x-1}-\frac{1}{x}-\frac{1}{y\cdot (y-1)} \end{array}$$

$$\begin{array}{l} \text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right) \quad \frac{2\cdot x}{x^2-2}+x+1 \\ \text{expand}(Ans, x) \quad \frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1 \end{array}$$

$$\begin{array}{l} \ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y} \quad \ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y} \\ \text{expand}(Ans) \quad \ln(x\cdot y)+\sqrt{2}\cdot \sqrt{x\cdot y}+\ln(2) \\ \text{expand}(Ans)|_{y\geq 0} \quad \ln(x)+\sqrt{2}\cdot \sqrt{x}\cdot \sqrt{y}+\ln(y)+\ln(2) \\ \text{sign}(x\cdot y)+|x\cdot y|+e^{2\cdot x+y} \\ e^{2\cdot x+y+\text{sign}(x\cdot y)+|x\cdot y|} \\ \text{expand}(Ans) \quad \text{sign}(x)\cdot \text{sign}(y)+|x|\cdot |y|+(e^x)^2\cdot e^y \end{array}$$



| expr()                                                                           | Katalog >  |                  |
|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------|
| <b>expr(String)⇒uttryck</b>                                                      |                                                                                           |                  |
| Ger teckensträngen i <i>String</i> som ett uttryck och exekverar det omedelbart. |                                                                                           |                  |
|                                                                                  | expr("1+2+x^2+x")                                                                         | $x^2+x+3$        |
|                                                                                  | expr("expand((1+x)^2)")                                                                   | $x^2+2\cdot x+1$ |
|                                                                                  | "Define cube(x)=x^3" → <i>funcstr</i>                                                     |                  |
|                                                                                  | "Define cube(x)=x^3"                                                                      |                  |
|                                                                                  | expr( <i>funcstr</i> )                                                                    | <i>Done</i>      |
|                                                                                  | cube(2)                                                                                   | 8                |

| ExpReg                                             | Katalog >  |  |
|----------------------------------------------------|---------------------------------------------------------------------------------------------|--|
| <b>ExpReg X, Y [, [Freq][, Category, Include]]</b> |                                                                                             |  |

Beräknar den exponentiella regressionen  $y = a \cdot (b)^x$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                                              |
|---------------------|----------------------------------------------------------|
| stat.RegEqn         | Regressionsekvation: $a \cdot (b)^x$                     |
| stat.a, stat.b      | Regressionskoefficienter                                 |
| stat.r <sup>2</sup> | Koefficient för linjär bestämning av transformerade data |

| Resultatvariabel | Beskrivning                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.r           | Korrelationskoefficient för transformerade data ( $x$ , $\ln(y)$ )                                                                                                          |
| stat.Resid       | Residualer associerade med den exponentiella modellen                                                                                                                       |
| stat.ResidTrans  | Residualer associerade med linjär anpassning av transformerade data                                                                                                         |
| stat.XReg        | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg        | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg     | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## F

### factor()

Katalog > 

**factor**(*Expr1*[, *Var*]) $\Rightarrow$ uttryck

**factor**(*List1*[, *Var*]) $\Rightarrow$ lista

**factor**(*Matrix1*[, *Var*]) $\Rightarrow$ matris

**factor**(*Expr1*) ger en faktorisering av *Expr1* baserad på uttryckets alla variabler med en gemensam nämnare.

*Expr1* faktoriseras så långt det går till linjära, rationella faktorer utan att införa nya icke-reella deluttryck. Detta alternativ är lämpligt om du vill ha en faktorisering baserad på mer än en variabel.

**factor**(*Expr1*, *Var*) ger en faktorisering av *Expr1* baserad på variabeln *Var*.

*Expr1* faktoriseras så långt det går till reella faktorer som är linjära i *Var*, även om detta inför irrationella konstanter eller deluttryck som är irrationella i andra variabler.

$$\begin{array}{r} \text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a) \\ a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1) \\ \hline \text{factor}(x^2+1) \quad x^2+1 \\ \text{factor}(x^2-4) \quad (x-2) \cdot (x+2) \\ \hline \text{factor}(x^2-3) \quad x^2-3 \\ \text{factor}(x^2-a) \quad x^2-a \end{array}$$

$$\begin{array}{r} \text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x) \\ a \cdot (a^2-1) \cdot (x-1) \cdot (x+1) \\ \hline \text{factor}(x^2-3, x) \quad (x+\sqrt{3}) \cdot (x-\sqrt{3}) \\ \text{factor}(x^2-a, x) \quad (x+\sqrt{a}) \cdot (x-\sqrt{a}) \end{array}$$

Faktorerna och deras termer sorteras med *Var* som huvudvariabel. Liknande potenser av *Var* samlas in i varje faktor. Inkludera *Var* om faktorisering baserad på endast denna variabel behövs och du är villig att acceptera irrationella uttryck i andra variabler för att öka faktoriseringen baserad på *Var*. Viss tillfällig faktorisering kan ske vad gäller andra variabler.

Med inställningen Auto i läge **Auto eller Ungefärlig** medger inkludering av *Var* en uppskattning med koefficienter i flyttalsform när irrationella koefficienter inte explicit kan uttryckas kortfattat med termerna i de inbyggda funktionerna. Även med endast en variabel kan inkludering av *Var* ge en mer fullständig faktorisering.

**Obs:** Se även **comDenom()** för ett snabbt sätt att erhålla partiell faktorisering när **factor()** inte är snabb nog eller om den utarmar minnet.

**Obs:** Se även **cFactor()** för faktorisering hela vägen till komplexa koefficienter i sökningen efter linjära faktorer.

**factor(rationalNumber)** ger det rationella talet faktorerat i primtal. För sammansatta tal ökar beräkningstiden exponentiellt med antalet siffror i den näst största faktorn. Som exempel kan faktorisering av ett 30-siffrigt heltal ta mer än en dag och faktorisering av ett 100-siffrigt tal kan ta mer än 100 år.

För att stoppa en beräkning manuellt,

- **Handenhet:** Håll ned  och tryck på  upprepade gånger.
- **Windows®:** Håll ned **F12** och tryck på **Enter** upprepade gånger.
- **Macintosh®:** Håll ned **F5** och tryck på **Enter** upprepade gånger.
- **iPad®:** Appen visar en uppmaning. Du

$$\frac{\text{factor}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3)}{x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3}$$

$$\frac{\text{factor}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3,x)}{(x-0.964673)\cdot(x+0.611649)\cdot(x+2.12543)\cdot(x^2$$

$$\frac{\text{factor}(152417172689)}{123457\cdot 1234577}$$

$$\frac{\text{isPrime}(152417172689)}{\text{false}}$$

kan fortsätta att vänta eller avbryta.

Om du endast vill bestämma om ett tal är ett primtal, använd **isPrime()** i stället. Detta går mycket fortare, särskilt om *rationalNumber* inte är ett primtal och om den näst största faktorn har mer än fem siffror.

## F Cdf()

### F Cdf

*(lowBound,upBound,dfNumer,dfDenom)⇒*  
*number* om *lowBound* och *upBound* är tal,  
*lista* om *lowBound* och *upBound* är listor

### FCdf

*(lowBound,upBound,dfNumer,dfDenom)⇒*  
*tal* om *lowBound* och *upBound* är tal, *lista*  
om *lowBound* och *upBound* är listor

Beräknar sannolikheten för F-fördelningen mellan *undre gräns* och *övre gräns* för det specificerade *dfNämn* (frihetsgrader) och *dfTälj*.

För  $P(X \leq \text{övrGräns})$ , ange *undrGräns* = 0.

## Fill

**Fill** *Expr*, *matrixVar* ⇒ *matrix*

Ersätter varje element i variabeln *matrixVar* med *Expr*.

*matrixVar* måste redan existera.

|                                                |                  |                                                            |
|------------------------------------------------|------------------|------------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ | → <i>amatrix</i> | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$             |
| Fill 1.01, <i>amatrix</i>                      |                  | <i>Done</i>                                                |
| <i>amatrix</i>                                 |                  | $\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$ |

**Fill** *Expr*, *listVar* ⇒ *lista*

Ersätter varje element i variabeln *listVar* med *Expr*.

*listVar* måste redan existera.

|                         |                |                                |
|-------------------------|----------------|--------------------------------|
| $\{1,2,3,4,5\}$         | → <i>alist</i> | $\{1,2,3,4,5\}$                |
| Fill 1.01, <i>alist</i> |                | <i>Done</i>                    |
| <i>alist</i>            |                | $\{1.01,1.01,1.01,1.01,1.01\}$ |

## FiveNumSummary

**FiveNumSummary** *X*[, [*Freq*]  
[, *Category*, *Include*]]

Ger en förkortad version av envariabelstatistiken för listan  $X$ . En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

$X$  representerar en lista på aktuella data.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande  $X$ -värde. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande  $X$ -data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

Ett tomt element i någon av listorna  $X$ , *Freq* eller *Category* resulterar i ett tomrum för motsvarande element i dessa listor. För mer information om tomma element, se på sidan 261.

| Resultatvariabel      | Beskrivning         |
|-----------------------|---------------------|
| stat.MinX             | Minsta x-värde      |
| stat.Q <sub>1</sub> X | Undre kvartil för x |
| stat.MedianX          | Median för x        |
| stat.Q <sub>3</sub> X | Övre kvartil för x  |
| stat.MaxX             | Största x-värde     |

## floor()

**floor(*Expr1*)**  $\Rightarrow$  *heltal*

$\text{floor}(-2.14)$  -3.

Ger det största heltal som är  $\leq$  argumentet. Denna funktion är identisk med **int()**.

Argumentet kan vara ett reellt eller ett komplext tal.

**floor()**Katalog > **floor**(*Listl*) $\Rightarrow$ *lista*

$$\text{floor}\left(\left\{\frac{3}{2}, 0, -5, 3\right\}\right) \quad \{1, 0, -6.\}$$

**floor**(*Matrixl*) $\Rightarrow$ *matris*

$$\text{floor}\left(\begin{pmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{pmatrix}\right) \quad \begin{pmatrix} 1. & 3. \\ 2. & 4. \end{pmatrix}$$

Ger en lista eller matris med golvvärden för varje element.

**Obs:** Se även **ceiling()** och **int()**.

**fMax()**Katalog > **fMax**(*Expr*, *Var*) $\Rightarrow$ *Booleskt uttryck*

$$\text{fMax}\left(1-(x-a)^2-(x-b)^2, x\right) \quad x=\frac{a+b}{2}$$

**fMax**(*Expr*, *Var*, *lowBound*)

$$\text{fMax}\left(.5 \cdot x^3 - x - 2, x\right) \quad x=\infty$$

**fMax**(*Expr*, *Var*, *lowBound*, *upBound*)

**fMax**(*Expr*, *Var*) | *lowBound*  $\leq$  *Var*  $\leq$  *upBound*

Ger ett booleskt uttryck som specificerar möjliga värden på *Var* som maximerar *Expr* eller lokaliserar dess lägsta övre gräns.

Du kan använda ("|")-operatoren begränsning för att begränsa lösningsintervallet och/eller specificera andra begränsningar.

$$\text{fMax}\left(0.5 \cdot x^3 - x - 2, x\right)_{|x \leq 1} \quad x=-0.816497$$

Med inställningen Approximate i läge **Auto** eller **Ungefärlig** söker **fMax()** iterativt efter ett ungefärligt lokalt maximum. Detta går ofta fortare, särskilt om du använder operatoren "|" för att begränsa sökningen till ett relativt litet intervall som innehåller exakt ett lokalt maximum.

**Obs:** Se även **fMin()** och **max()**.

**fMin()**Katalog > **fMin**(*Expr*, *Var*) $\Rightarrow$ *Booleskt uttryck*

$$\text{fMin}\left(1-(x-a)^2-(x-b)^2, x\right) \quad x=-\infty \text{ or } x=\infty$$

**fMin**(*Expr*, *Var*, *lowBound*)

$$\text{fMin}\left(0.5 \cdot x^3 - x - 2, x\right)_{|x \geq 1} \quad x=1.$$

**fMin**(*Expr*, *Var*, *lowBound*, *upBound*)

**fMin**(*Expr*, *Var*) | *lowBound*  $\leq$  *Var*

$\leq_{upBound}$ 

Ger ett booleskt uttryck som specificerar möjliga värden på *Var* som minimerar *Expr* eller lokaliserar dess största nedre gräns.

Du kan använda ("|")-operatoren begränsning för att begränsa lösningsintervallet och/eller specificera andra begränsningar.

Med inställningen Approximate i läge **Auto eller Ungefärlig** söker **fMin()** iterativt efter ett ungefärligt lokalt minimum. Detta går ofta fortare, särskilt om du använder operatoren "|" för att begränsa sökningen till ett relativt litet intervall som innehåller exakt ett lokalt minimum.

**Obs:** Se även **fMax()** och **min()**.

**For****For** *Var, Low, High* [, *Step*]*Block***EndFor**

Exekverar iterativt påståendena i *Block* för varje värde på *Var*, från *Low* till *High*, i steg enligt *Step*.

*Var* får inte vara en systemvariabel.

*Step* kan vara positivt eller negativt. Det förinställda värdet är 1.

*Block* kan vara antingen ett enstaka påstående eller en serie av påståenden separerade med tecknet ":".

**Obs för att mata in exemplet:** Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

Define  $g()$ =Func

Done

Local *tempsum, step, i* $0 \rightarrow tempsum$  $1 \rightarrow step$ For *i*, 1, 100, *step**tempsum* + *i*  $\rightarrow tempsum$ 

EndFor

EndFunc

 $g()$ 

5050

**format(*Expr*[, *formatString*])**⇒*sträng*

Ger *Expr* som en teckensträng baserad på formatmallen.

*Expr* måste förenklas till ett tal.

*formatString* är en sträng och måste ha formen: "F[n]", "S[n]", "E[n]", "G[n][c]", där [ ] indikerar frivilliga delar.

F[n]: Fast format. n är antalet siffror som skall visas efter decimalpunkten.

S[n]: Scientific format (Grundpotensform). n är antalet siffror som skall visas efter decimalpunkten.

E[n]: Engineering format. n är antalet siffror efter den första signifikanta siffran. Exponenten justeras till en multipel av tre och decimalpunkten flyttas åt höger med noll, en eller två siffror.

G[n][c]: Samma som fast format, men separerar också siffror till vänster om basen i grupper om tre. c specificerar det gruppseparerande tecknet och är förinställt på kommatecken. Om c är en punkt visas basen som ett kommatecken.

[Rc]: Samtliga ovanstående specifikationssymboler kan försees med suffix med Rc-basflaggan, där c är ett enstaka tecken som specificerar vad som skall ersättas för baspunkten.

|                                      |                          |
|--------------------------------------|--------------------------|
| <code>format(1.234567,"f3")</code>   | <code>"1.235"</code>     |
| <code>format(1.234567,"s2")</code>   | <code>"1.23E0"</code>    |
| <code>format(1.234567,"e3")</code>   | <code>"1.235E0"</code>   |
| <code>format(1.234567,"g3")</code>   | <code>"1.235"</code>     |
| <code>format(1234.567,"g3")</code>   | <code>"1,234.567"</code> |
| <code>format(1.234567,"g3,r")</code> | <code>"1:235"</code>     |

## fPart()

**fPart(*Expr*)**⇒*uttryck*

**fPart(*List*)**⇒*lista*

**fPart(*Matrix*)**⇒*matris*

Ger argumentets bråkdelen.

Ger, för en lista eller matris, elementens bråkdelen.

|                                    |                             |
|------------------------------------|-----------------------------|
| <code>fPart(-1.234)</code>         | <code>-0.234</code>         |
| <code>fPart({1,-2.3,7.003})</code> | <code>{0,-0.3,0.003}</code> |



Argumentet kan vara ett reellt eller ett komplext tal.

$Fpdf(XVal, dfNumer, dfDenom) \Rightarrow tal$  om  $XVal$  är ett tal, *lista* om  $XVal$  är en lista

Beräknar sannolikheten för F-fördelning vid  $XVal$  för specificerad  $dfNumer$  (frihetsgrader) och  $dfDenom$ .

**freqTable►list**

$(List1, freqIntegerList) \Rightarrow lista$

Ger en lista som innehåller elementen från *List1* expanderad enligt frekvenserna i *freqIntegerList*. Denna funktion kan användas för att skapa en frekvenstabell för applikationen Data & Statistik.

*List1* kan vara vilken giltig lista som helst.

*freqIntegerList* måste ha samma dimensioner som *List1* och får endast innehålla icke-negativa heltalselement. Varje element specificerar antalet gånger motsvarande *List1*-element kommer att upprepas i resultatlistan. Ett värde på noll utesluter motsvarande *List1*-element.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **freqTable@>list(...)**.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

---

```
freqTable►list({ 1,2,3,4 }, { 1,4,3,1 })
               { 1,2,2,2,2,3,3,3,4 }


---


freqTable►list({ 1,2,3,4 }, { 1,4,0,1 })
               { 1,2,2,2,4 }


---


```

**frequency(List1, binsList) ⇒ lista**

Ger en lista med antalet element i *List1*.  
Talen baseras på områden (bins = staplar) som du definierar i *binsList*.

Om *binsList* är {b(1), b(2), ..., b(n)} är de specificerade områdena { $? \leq b(1)$ ,  $b(1) < ? \leq b(2)$ , ...,  $b(n-1) < ? \leq b(n)$ ,  $b(n) > ?$ }. Den resulterande listan är ett element längre än *binsList*.

Varje element i resultatet motsvarar antalet element från *List1* som är i området för denna stapel. Uttryckt enligt funktionen **countIf()** är resultatet {countIf(list,  $? \leq b(1)$ ), countIf(list,  $b(1) < ? \leq b(2)$ ), ..., countIf(list,  $b(n-1) < ? \leq b(n)$ ), countIf(list,  $b(n) > ?$ )}.

Element i *List1* som inte kan "placeras i en stapel" ignoreras. Även tomma element ignoreras. För mer information om tomma element, se på sidan 261.

I applikationen Listor och kalkylblad kan du använda ett område av celler i stället för båda argumenten.

**Obs:** Se även **countIf()**, på sidan 36.

|                                                               |
|---------------------------------------------------------------|
| <i>datalist</i> = { 1, 2, e, 3, $\pi$ , 4, 5, 6, "hello", 7 } |
| { 1, 2, 2.71828, 3, 3.14159, 4, 5, 6, "hello", 7 }            |
| frequency( <i>datalist</i> , { 2.5, 4.5 })      { 2, 4, 3 }   |

Förklaring av resultat:

**2** element från *Datalist* är  $\leq 2.5$

**4** element från *Datalist* är  $> 2.5$  och  $\leq 4.5$

**3** element från *Datalist* är  $> 4.5$

Elementet "hello" är en sträng och kan inte placeras i någon av de definierade staplarna.

## FTest\_2Samp

**FTest\_2Samp** List1, List2[, Freq1[, Freq2[, Hypoth]]]

**FTest\_2Samp** List1, List2[, Freq1[, Freq2[, Hypoth]]]

(Indatalista)

**FTest\_2Samp** sx1, n1, sx2, n2[, Hypoth]

**FTest\_2Samp** sx1, n1, sx2, n2[, Hypoth]

(Summary stats indata)

Utför ett 2-sampel F test. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

eller  $H_a: \sigma_1 > \sigma_2$ , sätt *Hypoth*>0

För  $H_a: \sigma_1 \neq \sigma_2$  (förinställning), sätt *Hypoth*=0

För  $H_a: \sigma_1 < \sigma_2$ , sätt *Hypoth*<0

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel           | Beskrivning                                                                        |
|----------------------------|------------------------------------------------------------------------------------|
| stat.F                     | Beräknad $\hat{U}$ -statistik för datasekvensen                                    |
| stat.PVal                  | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas                      |
| stat.dfNumer               | täljare, frihetsgrader = $n_1 - 1$                                                 |
| stat.dfDenom               | nämnare, frihetsgrader = $n_2 - 1$                                                 |
| stat.sx1, stat.sx2         | Standardavvikelser hos urvalet i datasekvenserna i <i>List 1</i> och <i>List 2</i> |
| stat.x1_bar<br>stat.x2_bar | Medelvärden hos urvalet i datasekvenserna i <i>List 1</i> och <i>List 2</i>        |
| stat.n1, stat.n2           | Storlek på urvalen                                                                 |

## Func

### Func

Definiera en stegvis funktion:

*Block*

```

Define g(x)=Func
  If x<0 Then
    Return 3*cos(x)
  Else
    Return 3-x
  EndIf
EndFunc
  
```

*Done*

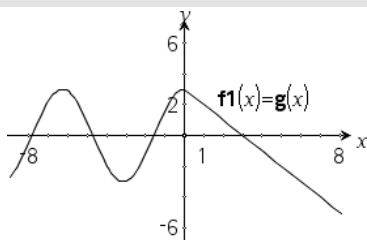
### EndFunc

Mall för att skapa en användardefinierad funktion.

*Block* kan vara ett enstaka påstående, en serie av påståenden separerade med tecknet ":" eller en serie av påståenden på separata rader. Funktionen kan använda instruktionen **Return** för att ge ett specifikt resultat.

Resultat från plottning av  $g(x)$

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.



## G

## gcd()

Katalog > **gcd(Value1, Value2)** ⇒ *uttryck*

gcd(18,33) 3

Ger den största gemensamma delaren för de två argumenten. **gcd** för två bråk är **gcd** för deras täljare dividerat med **lcm** för deras nämnare.

I läge Auto eller Approximate (Ungefärlig) är **gcd** 1.0 för bråkital i flyttalsform.

**gcd(List1, List2)** ⇒ *lista*

gcd({12,14,16},{9,7,5}) {3,7,1}

Ger största gemensamma delare för motsvarande element i *List1* och *List2*.

**gcd(Matrix1, Matrix2)** ⇒ *matrix*gcd( $\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}, \begin{pmatrix} 4 & 8 \\ 12 & 16 \end{pmatrix}$ )  $\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$ 

Ger största gemensamma delare för motsvarande element i *Matrix1* och *Matrix2*.

## geomCdf()

Katalog > 

**geomCdf(p, lowBound, upBound)** ⇒ *tal* om *lowBound* och *upBound* är tal, *lista* om *lowBound* och *upBound* är listor

**geomCdf(p, upBound)** för  $P(1 \leq X \leq upBound)$  ⇒ *tal* om *upBound* är ett tal, *lista* om *upBound* är en lista

Beräknar en kumulativ geometrisk sannolikhet från *lowBound* till *upBound* med den specificerade sannolikheten *p* för att lyckas.

För  $P(X \leq upBound)$ , sätt *lowBound* = 1.

## geomPdf()

**geomPdf(*p*, *XVal*)**  $\Rightarrow$  *tal* om *XVal* är ett tal,  
*lista* om *XVal* är en lista

Beräknar en sannolikhet vid *XVal*, vid vilket försök i försöksomgången som man lyckas första gången, för den diskreta geometriska fördelningen med den specificerade sannolikheten *p* för att lyckas.

## Get

## Hubb-meny

**Get**[*promptString*,] *var*[, *statusVar*]

**Get**[*promptString*,] *func*(*arg1*, ...*argn*)  
[, *statusVar*]

Programmeringskommando: Hämtar ett värde från en ansluten TI-Innovator™ Hub och tilldelar värdet till variabeln *var*.

Värdet måste begäras:

- På förhand genom ett **Send "READ ..."** -kommando.  
— eller —
- Genom att bädda in en **"READ ..."** -begäran som alternativt *promptString*-argument. Med denna metod kan du använda ett enda kommando för att begära värdet och hämta det.

Implicit förenkling äger rum. Till exempel tolkas en mottagen sträng "123" som ett numeriskt värde. För att bevara strängen, använd **GetStr** istället för **Get**.

Exempel: Begär nuvarande värde från hubbens inbyggda ljusnivåsensor. Använd **Get** för att hämta värdet och tilldela det till variabeln *lightval*.

|                        |          |
|------------------------|----------|
| Send "READ BRIGHTNESS" | Done     |
| Get <i>lightval</i>    | Done     |
| <i>lightval</i>        | 0.347922 |

Bädda in READ-begäran i **Get**-kommandot.

|                                         |          |
|-----------------------------------------|----------|
| Get "READ BRIGHTNESS" , <i>lightval</i> | Done     |
| <i>lightval</i>                         | 0.378441 |

Om du inkluderar det valfria argumentet *statusVar*, tilldelas det ett värde baserat på om operationen har lyckats. Värdet noll betyder att inga data mottogs.

I den andra syntaxen kan ett program använda argumentet *func()* för att lagra den mottagna strängen som en funktionsdefinition. Denna syntax fungerar som om programmet exekverade kommandot:

Define *func(arg1, ...argn) = received string*

Programmet kan sedan använda den definierade funktionen *func()*.

**Obs:** Du kan använda kommandot **Get** i ett användardefinierat program, men inte i en funktion.

**Obs:** Se även **GetStr**, på sidan 88 och **Send**, på sidan 164.

## getDenom()

Katalog > 

**getDenom(*Expr1*)** ⇒ *uttryck*

Transformerar argumentet till ett uttryck med reducerad gemensam nämnare och ger sedan dess nämnare.

|                                                               |             |
|---------------------------------------------------------------|-------------|
| $\text{getDenom}\left(\frac{x+2}{y-3}\right)$                 | $y-3$       |
| $\text{getDenom}\left(\frac{2}{7}\right)$                     | 7           |
| $\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$ | $x \cdot y$ |

## getKey()

Katalog > 

**getKey([0|1])** ⇒ **returnString**

**Beskrivning:** **getKey()** – låter ett TI-Basic-program tolka tangentbordsinmatning – på handenhet, dator och emulator på dator.

**getKey()**

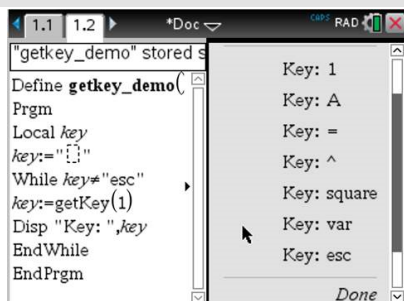
**Exempel:**

**Exempel:**

- keypressed := **getKey()** kommer att returnera en tangent eller tom

sträng om ingen tangent har tryckts ned. Detta anrop returneras omedelbart.

- `keypressed := getKey(1)` väntar till en tangent trycks ned. Detta anrop pausar start av ett program tills en tangent trycks ned.



### Hantering av tangentnedtryckningar:

| Handhållen enhet/emulatortangent | Desktop (dator) | Returvärde   |
|----------------------------------|-----------------|--------------|
| Esc                              | Esc             | "esc"        |
| Pekplatta - toppklick            | Ej tillämpligt  | "up"         |
| På                               | Ej tillämpligt  | "home"       |
| Scratchapps                      | Ej tillämpligt  | "scratchpad" |
| Pekplatta - vänsterklick         | Ej tillämpligt  | "left"       |
| Pekplatta - mittklick            | Ej tillämpligt  | "center"     |
| Pekplatta - högerklick           | Ej tillämpligt  | "right"      |
| Doc                              | Ej tillämpligt  | "doc"        |
| Tab                              | Tab             | "tab"        |
| Pekplatta – nederklick           | Pil ned         | "down"       |
| Meny                             | Ej tillämpligt  | "menu"       |
| Ctrl                             | Ctrl            | ej retur     |
| Skift                            | Skift           | ej retur     |
| Var                              | Ej tillämpligt  | "var"        |
| Radera                           | Ej tillämpligt  | "del"        |
| =                                | =               | "_"          |
| trig                             | Ej tillämpligt  | "trig"       |
| 0 till 9                         | 0–9             | "0" ... "9"  |

| Handhållen<br>enhet/emulator tangent | Desktop (dator) | Returvärde                                          |
|--------------------------------------|-----------------|-----------------------------------------------------|
| Mallar                               | Ej tillämpligt  | "template"                                          |
| Katalog                              | Ej tillämpligt  | "cat"                                               |
|                                      |                 |                                                     |
| ^                                    | ^               | "^"                                                 |
| X^2                                  | Ej tillämpligt  | "square"                                            |
| / (divisionstangent)                 | /               | "/"                                                 |
| * (multiplikationstangent)           | *               | "*"                                                 |
| e^x                                  | Ej tillämpligt  | "exp"                                               |
| 10^x                                 | Ej tillämpligt  | "10power"                                           |
| +                                    | +               | "+"                                                 |
| -                                    | -               | "_"                                                 |
|                                      |                 |                                                     |
| (                                    | (               | "("                                                 |
| )                                    | )               | ")"                                                 |
| .                                    | .               | ". "                                                |
| (-)                                  | Ej tillämpligt  | "-" (negativt tecken)                               |
| Mata in                              | Mata in         | "enter"                                             |
|                                      |                 |                                                     |
| ee                                   | Ej tillämpligt  | "E" (grundpotensform)                               |
| a - z                                | a-z             | alpha = bokstav nedtryckt<br>(gemen)<br>("a" - "z") |
| shift a-z                            | shift a-z       | alpha = bokstav nedtryckt<br>"A" - "Z"              |
|                                      |                 | Obs: ctrl-shift låser versaler<br>(caps)            |
| ?!                                   | Ej tillämpligt  | "?!"                                                |
|                                      |                 |                                                     |
| pi                                   | Ej tillämpligt  | "pi"                                                |
| Flagga                               | Ej tillämpligt  | ej retur                                            |
|                                      |                 |                                                     |
| ,                                    | ,               | " , "                                               |
| Retur                                | Ej tillämpligt  | "return"                                            |



| Handhållen enhet/emulator tangent | Desktop (dator)                                                                                                                   | Returvärde                                                   |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Mellanslag                        | Mellanslag                                                                                                                        | " " (mellanslag)                                             |
| Ej tillgänglig                    |                                                                                                                                   |                                                              |
| Ej tillgänglig                    | Specialtecken såsom @,!,^, etc.                                                                                                   | Tecknet returneras                                           |
| Ej tillämpligt                    | Funktionstangenter                                                                                                                | Inga returnerade tecken                                      |
| Ej tillämpligt                    | Specialkontrolltangenter för dator                                                                                                | Inga returnerade tecken                                      |
| Ej tillgänglig                    | Andra datortangenter som inte finns tillgängliga på handenheten medan getKey() väntar på en tangentnedtryckning. ({, },;, :, ...) | Samma tecken du får i Anteckningar (inte i en matematikruta) |

**Obs:** Det är viktigt att observera att närvaron av **getKey()** i ett program ändrar hur systemet hanterar vissa händelser. Vissa av dessa beskrivs nedan.

**Avsluta program och hantera händelse** – Precis som om användaren vill lämna programmet genom att trycka på tangenten **ON**

**"Support"** nedan innebär – Systemet fungerar som förväntat - programmet fortsätter att köras.

| Händelse                                                                                                    | Enhet                             | Dator – TI-Nspire™ Student Software                                                                  |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------|
| Snabbtest                                                                                                   | Avsluta program, hantera händelse | Samma som handenhet (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-enbart)  |
| Hantering av fjärrfil<br><br>(Inkl. utskick av fil "Exit Press 2 Test" från en annan handenhet eller dator) | Avsluta program, hantera händelse | Samma som handenhet. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-enbart) |
| Avsluta klass                                                                                               | Avsluta program, hantera händelse | Support (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-enbart)              |

| Händelse | Enhet | Dator - TI-Nspire™ Alla versioner |
|----------|-------|-----------------------------------|
|----------|-------|-----------------------------------|

|                                         |                                                                                                                                                                            |                       |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| TI-Innovator™ Hub<br>anslut/koppla från | Support – Kan enkelt<br>utfärda kommandon till TI-<br>Innovator™ Hub. Efter att<br>du lämnat programmet<br>jobbar TI-Innovator™ Hub<br>fortfarande med den<br>handenheten. | Samma som handenheten |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|

## getLangInfo() Katalog >

**getLangInfo()** ⇒ *sträng*

`getLangInfo()`

"en"

Ger en sträng som motsvarar det korta namnet på det aktuella aktiva språket. Du kan exempelvis använda den i ett program eller i en funktion för att bestämma det aktuella språket.

Engelska = "en"

Danska = "da"

Tyska = "de"

Finska = "fi"

Franska = "fr"

Italienska = "it"

Holländska = "nl"

Belgisk holländska = "nl\_BE"

Norska = "no"

Portugisiska = "pt"

Spanska = "es"

Svenska = "sv"

## getLockInfo() Katalog >

**getLockInfo(Var)** ⇒ *värde*

`a:=65`

65

Ger den aktuella låsta/olåsta statusen för variabeln *Var*.

`Lock a`

*Done*

`getLockInfo(a)`

1

*värde* = 0: *Var* är olåst eller finns inte.

`a:=75`

"Error: Variable is locked."

`DelVar a`

"Error: Variable is locked."

*värde* = 1: *Var* är låst och kan inte modifieras eller tas bort.

`Unlock a`

*Done*

`a:=75`

75

`DelVar a`

*Done*

Se **Lock**, på sidan 111 och **unLock**, på sidan 206.

**getMode(ModeNameInteger)**⇒värde

**getMode(0)**⇒lista

**getMode(ModeNameInteger)** ger ett värde som representerar den aktuella lägesinställningen för *ModeNameInteger*.

**getMode(0)** ger en lista på talpar. Varje par består av ett lägesheltal och ett inställningsheltal.

Se nedan för en lista på lägen och deras inställningar.

Om du sparar inställningarna med **getMode(0)** → *var* kan du använda **setMode(var)** i en funktion eller ett program för att temporärt återställa inställningarna endast inom exekveringen av funktionen eller programmet. Se **setMode()**, på sidan 169.

|            |                                     |
|------------|-------------------------------------|
| getMode(0) | { 1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1 } |
| getMode(1) | 7                                   |
| getMode(8) | 1                                   |

| Lägets namn                                        | Lägesheltal | Heltal för inställningar                                                                                                                                                                                                                                       |
|----------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Display Digits<br>(Antal siffror)                  | 1           | 1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12 |
| Angle (Vinkel)                                     | 2           | 1=Radian, 2=Degree, 3=Gradian                                                                                                                                                                                                                                  |
| Exponential<br>Format<br>(Exponentiellt<br>format) | 3           | 1=Normal, 2=Scientific, 3=Engineering                                                                                                                                                                                                                          |
| Real or Complex<br>(Reellt eller<br>Komplext)      | 4           | 1=Real, 2=Rectangular, 3=Polar                                                                                                                                                                                                                                 |
| Auto or Approx.<br>(Auto eller<br>Ungefärlig)      | 5           | 1=Auto, 2=Approximate, 3=Exact                                                                                                                                                                                                                                 |
| Vector Format<br>(Vektorformat)                    | 6           | 1=Rectangular, 2=Cylindrical, 3=Spherical                                                                                                                                                                                                                      |
| Base (Bas)                                         | 7           | 1=Decimal, 2=Hex, 3=Binary                                                                                                                                                                                                                                     |

| Lägets namn                   | Läges-heltal | Heltal för inställningar |
|-------------------------------|--------------|--------------------------|
| Unit System<br>(Enhetssystem) | 8            | 1=SI, 2=Eng/US           |

## getNum() Katalog >

**getNum(*Expr1*)** ⇒ *uttryck*

Transformerar argumentet till ett uttryck med reducerad gemensam nämnare och ger sedan dess täljare.

|                                                       |       |
|-------------------------------------------------------|-------|
| $\text{getNum}\left(\frac{x+2}{y-3}\right)$           | $x+2$ |
| $\text{getNum}\left(\frac{2}{7}\right)$               | 2     |
| $\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$ | $x+y$ |

## GetStr Hubb-meny

**GetStr**[*promptString*,] *var*[, *statusVar*]

Se **Get** för exempel.

**GetStr**[*promptString*,] *func*(*arg1*, ...*argn*)  
[, *statusVar*]

Programmeringskommando: Fungerar precis som kommandot **Get**, förutom att det mottagna värdet alltid tolkas som en sträng. I motsats tolkar kommandot **Get** svaret som ett uttryck såvida det inte är omgivet av citationstecken ("").

**Obs:** Se även **Get**, på sidan 81 och **Send**, på sidan 164.

## getType() Katalog >

**getType(*var*)** ⇒ *sträng*

Ger en sträng som anger datatypen för variabeln *var*.

Om *var* inte har definierats erhålls strängen "NONE".

|                              |             |
|------------------------------|-------------|
| $\{1,2,3\} \rightarrow temp$ | $\{1,2,3\}$ |
| $\text{getType}(temp)$       | "LIST"      |
| $3 \cdot i \rightarrow temp$ | $3 \cdot i$ |
| $\text{getType}(temp)$       | "EXPR"      |
| $\text{DelVar } temp$        | <i>Done</i> |
| $\text{getType}(temp)$       | "NONE"      |

**getVarInfo()** ⇒ *matris* eller *sträng*

**getVarInfo(LibNameString)** ⇒ *matris* eller *sträng*

**getVarInfo()** ger en matris med information (variabelnamn, typ, åtkomlighet till bibliotek och låst/olåst status) för alla variabler och biblioteksobjekt som är definierade i det aktuella problemet.

Om inga variabler är definierade ger **getVarInfo()** strängen "NONE".

**getVarInfo(LibNameString)** ger en matris med information för alla biblioteksobjekt som är definierade i biblioteket *LibNameString*. *LibNameString* måste vara en sträng (text omsluten med citationstecken) eller en strängvariabel.

Om biblioteket *LibNameString* inte finns uppstår ett fel.

Se exemplet till vänster där resultatet av **getVarInfo()** tilldelas variabeln *vs*. Ett försök att visa rad 2 eller rad 3 av *vs* ger ett "Ogiltig lista eller matris"-fel eftersom minst ett av elementen i dessa rader (till exempel, variabel *b*) omvärderas till en matris.

Detta fel kan också inträffa när *Ans* används för att utvärdera ett **getVarInfo()**-resultat på nytt.

Systemet ger ovanstående fel eftersom den aktuella versionen av programvaran inte stöder en generaliserad matrisstruktur där ett element i en matris kan vara antingen en matris eller en lista.

|                                   |                                                                                                                                                                                                                                          |           |        |          |   |     |        |           |   |     |        |          |   |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------|----------|---|-----|--------|-----------|---|-----|--------|----------|---|
| getVarInfo()                      | "NONE"                                                                                                                                                                                                                                   |           |        |          |   |     |        |           |   |     |        |          |   |
| Define $x=5$                      | Done                                                                                                                                                                                                                                     |           |        |          |   |     |        |           |   |     |        |          |   |
| Lock $x$                          | Done                                                                                                                                                                                                                                     |           |        |          |   |     |        |           |   |     |        |          |   |
| Define LibPriv $y=\{1,2,3\}$      | Done                                                                                                                                                                                                                                     |           |        |          |   |     |        |           |   |     |        |          |   |
| Define LibPub $z(x)=3\cdot x^2-x$ | Done                                                                                                                                                                                                                                     |           |        |          |   |     |        |           |   |     |        |          |   |
| getVarInfo()                      | <table><tr><td><math>x</math></td><td>"NUM"</td><td>"{"</td><td>1</td></tr><tr><td><math>y</math></td><td>"LIST"</td><td>"LibPriv"</td><td>0</td></tr><tr><td><math>z</math></td><td>"FUNC"</td><td>"LibPub"</td><td>0</td></tr></table> | $x$       | "NUM"  | "{"      | 1 | $y$ | "LIST" | "LibPriv" | 0 | $z$ | "FUNC" | "LibPub" | 0 |
| $x$                               | "NUM"                                                                                                                                                                                                                                    | "{"       | 1      |          |   |     |        |           |   |     |        |          |   |
| $y$                               | "LIST"                                                                                                                                                                                                                                   | "LibPriv" | 0      |          |   |     |        |           |   |     |        |          |   |
| $z$                               | "FUNC"                                                                                                                                                                                                                                   | "LibPub"  | 0      |          |   |     |        |           |   |     |        |          |   |
| getVarInfo( $tmp3$ )              | "Error: Argument must be a string"                                                                                                                                                                                                       |           |        |          |   |     |        |           |   |     |        |          |   |
| getVarInfo("tmp3")                | <table><tr><td><math>volcy12</math></td><td>"NONE"</td><td>"LibPub"</td><td>0</td></tr></table>                                                                                                                                          | $volcy12$ | "NONE" | "LibPub" | 0 |     |        |           |   |     |        |          |   |
| $volcy12$                         | "NONE"                                                                                                                                                                                                                                   | "LibPub"  | 0      |          |   |     |        |           |   |     |        |          |   |

|                           |                                                                                                                                                                                                                             |     |       |     |   |     |       |     |   |     |       |     |   |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|---|-----|-------|-----|---|-----|-------|-----|---|
| $a:=1$                    | 1                                                                                                                                                                                                                           |     |       |     |   |     |       |     |   |     |       |     |   |
| $b:=[1\ 2]$               | $[1\ 2]$                                                                                                                                                                                                                    |     |       |     |   |     |       |     |   |     |       |     |   |
| $c:=[1\ 3\ 7]$            | $[1\ 3\ 7]$                                                                                                                                                                                                                 |     |       |     |   |     |       |     |   |     |       |     |   |
| $vs:=\text{getVarInfo}()$ | <table><tr><td><math>a</math></td><td>"NUM"</td><td>"{"</td><td>0</td></tr><tr><td><math>b</math></td><td>"MAT"</td><td>"{"</td><td>0</td></tr><tr><td><math>c</math></td><td>"MAT"</td><td>"{"</td><td>0</td></tr></table> | $a$ | "NUM" | "{" | 0 | $b$ | "MAT" | "{" | 0 | $c$ | "MAT" | "{" | 0 |
| $a$                       | "NUM"                                                                                                                                                                                                                       | "{" | 0     |     |   |     |       |     |   |     |       |     |   |
| $b$                       | "MAT"                                                                                                                                                                                                                       | "{" | 0     |     |   |     |       |     |   |     |       |     |   |
| $c$                       | "MAT"                                                                                                                                                                                                                       | "{" | 0     |     |   |     |       |     |   |     |       |     |   |
| $vs[1]$                   | $[1\ \text{"NUM"}\ \text{"{"}\ 0]$                                                                                                                                                                                          |     |       |     |   |     |       |     |   |     |       |     |   |
| $vs[1,1]$                 | 1                                                                                                                                                                                                                           |     |       |     |   |     |       |     |   |     |       |     |   |
| $vs[2]$                   | "Error: Invalid list or matrix"                                                                                                                                                                                             |     |       |     |   |     |       |     |   |     |       |     |   |
| $vs[2,1]$                 | $[1\ 2]$                                                                                                                                                                                                                    |     |       |     |   |     |       |     |   |     |       |     |   |

| Goto                                                                               |                             | Katalog >  |
|------------------------------------------------------------------------------------|-----------------------------|-------------------------------------------------------------------------------------------|
| <b>Goto</b> <i>labelName</i>                                                       | Define $g()$ =Func          | Done                                                                                      |
| Överför kontroll till etiketten <i>labelName</i> .                                 | Local <i>temp,i</i>         |                                                                                           |
| <i>labelName</i> måste definieras i samma funktion med en <b>Lbl</b> -instruktion. | $0 \rightarrow temp$        |                                                                                           |
|                                                                                    | $1 \rightarrow i$           |                                                                                           |
|                                                                                    | Lbl <i>top</i>              |                                                                                           |
|                                                                                    | $temp + i \rightarrow temp$ |                                                                                           |
|                                                                                    | If $i < 10$ Then            |                                                                                           |
|                                                                                    | $i + 1 \rightarrow i$       |                                                                                           |
|                                                                                    | Goto <i>top</i>             |                                                                                           |
|                                                                                    | EndIf                       |                                                                                           |
|                                                                                    | Return <i>temp</i>          |                                                                                           |
|                                                                                    | EndFunc                     |                                                                                           |
|                                                                                    | $g()$                       | 55                                                                                        |

| ►Grad                                                                                              |                                         | Katalog >  |
|----------------------------------------------------------------------------------------------------|-----------------------------------------|---------------------------------------------------------------------------------------------|
| <i>Expr1</i> ► <b>Grad</b> ⇒ <i>uttryck</i>                                                        | I vinkelläget Grader:                   |                                                                                             |
| Konverterar <i>Expr1</i> till en vinkelmätning i nygrader.                                         | $(1.5) \blacktriangleright \text{Grad}$ | $(1.66667)^g$                                                                               |
| <b>Obs:</b> Du kan infoga denna operator med datorns tangentbord genom att skriva @> <b>Grad</b> . | I vinkelläget Radianer:                 |                                                                                             |
|                                                                                                    | $(1.5) \blacktriangleright \text{Grad}$ | $(95.493)^g$                                                                                |

/

| identity()                                              |             | Katalog >       |
|---------------------------------------------------------|-------------|--------------------------------------------------------------------------------------------------|
| <b>identity</b> ( <i>Integer</i> ) ⇒ <i>matrix</i>      | identity(4) | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| Ger identitetsmatrisen med dimensionen <i>Integer</i> . |             |                                                                                                  |
| <i>Integer</i> måste vara positivt heltal.              |             |                                                                                                  |

| If                                                        |                     | Katalog >  |
|-----------------------------------------------------------|---------------------|-----------------------------------------------------------------------------------------------|
| <b>If</b> <i>BooleanExpr</i><br><i>Påståenden</i>         | Define $g(x)$ =Func | Done                                                                                          |
|                                                           | If $x < 0$ Then     |                                                                                               |
|                                                           | Return $x^2$        |                                                                                               |
|                                                           | EndIf               |                                                                                               |
| <b>If</b> <i>BoolesktUttr</i> <b>Then</b><br><i>Block</i> | EndFunc             |                                                                                               |
| <b>EndIf</b>                                              | $g(-2)$             | 4                                                                                             |

Om *BooleanExpr* är sant och exekverar sedan det enstaka påståendet *Statement* eller blocket av påståenden *Block* innan exekveringen fortsätter.

Om *BooleanExpr* är falskt, fortsätter exekveringen utan att exekvera påståendet eller blocket av påståenden.

*Block* kan vara antingen ett enstaka påstående eller en serie av påståenden separerade med tecknet ":" .

**Obs för att mata in exemplet:** Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

**If BoolesktUtrr Then**

*Block1*

**Else**

*Block2*

**EndIf**

Om *BooleanExpr* är sant, exekverar *Block1* och hoppar sedan över *Block2*.

Om *BooleanExpr* är falskt, hoppas *Block1* över och *Block2* exekveras.

*Block1* och *Block2* kan vara enstaka påståenden.

**If BoolesktUtrr1 Then**

*Block1*

**Elseif BoolesktUtrr2 Then**

*Block2*

:

**Elseif BoolesktUtrrN Then**

*BlockN*

**EndIf**

Medger förgrening. If *BooleanExpr1* utvärderar till sant och exekverar *Block1*. If *BooleanExpr1* utvärderar till falskt, utvärderar *BooleanExpr2*, osv.

|                     |      |
|---------------------|------|
| Define $g(x)$ =Func | Done |
| If $x < 0$ Then     |      |
| Return $\neg x$     |      |
| Else                |      |
| Return $x$          |      |
| EndIf               |      |
| EndFunc             |      |
| $g(12)$             | 12   |
| $g(-12)$            | 12   |

|                                        |      |
|----------------------------------------|------|
| Define $g(x)$ =Func                    |      |
| If $x < -5$ Then                       |      |
| Return 5                               |      |
| ElseIf $x > -5$ and $x < 0$ Then       |      |
| Return $\neg x$                        |      |
| ElseIf $x \geq 0$ and $x \neq 10$ Then |      |
| Return $x$                             |      |
| ElseIf $x = 10$ Then                   |      |
| Return 3                               |      |
| EndIf                                  |      |
| EndFunc                                |      |
|                                        | Done |
| $g(-4)$                                | 4    |
| $g(10)$                                | 3    |

**ifFn**(*BooleanExpr*, *Value\_If\_true* [, *Value\_If\_false* [, *Value\_If\_unknown*]])  
 $\Rightarrow$  uttryck, lista eller matris

Utvärderar det booleska uttrycket *BooleanExpr* (eller varje element från *BooleanExpr* ) och producerar ett resultat baserat på följande regler:

- *BooleanExpr* kan testa ett enstaka värde, en lista eller en matris.
- Om ett element i *BooleanExpr* utvärderas som sant erhålls motsvarande element från *Value\_If\_true*.
- Om ett element i *BooleanExpr* utvärderas som falskt erhålls motsvarande element från *Value\_If\_false*. Om du utelämnar *Value\_If\_false* erhålls undef.
- Om ett element i *BooleanExpr* är varken sant eller falskt erhålls motsvarande element från *Value\_If\_unknown*. Om du utelämnar *Value\_If\_unknown* erhålls undef.
- Om det andra, tredje eller fjärde argumentet i funktionen **ifFn()** är ett enstaka uttryck tillämpas det booleska testet på varje position i *BooleanExpr*.

**Obs:** Om det förenklade påståendet *BooleanExpr* inbegriper en lista eller matris måste alla övriga list- eller matrisargument ha samma dimensioner, och resultatet får då samma dimensioner.

---


$$\text{ifFn}(\{1,2,3\} < 2.5, \{5,6,7\}, \{8,9,10\})$$


---


$$\{5,6,10\}$$

Testvärdet på **1** är mindre än 2.5, varför dess motsvarande

*Value\_If\_True* element **5** kopieras till resultatlistan.

Testvärdet på **2** är mindre än 2.5, varför dess motsvarande

*Value\_If\_True* element **6** kopieras till resultatlistan.

Testvärdet på **3** är inte mindre än 2,5, varför dess motsvarande *Value\_If\_False* element **10** kopieras till resultatlistan.

---


$$\text{ifFn}(\{1,2,3\} < 2.5, 4, \{8,9,10\})$$


---


$$\{4,4,10\}$$

*Value\_If\_true* är ett enstaka värde och motsvarar varje vald position.

---


$$\text{ifFn}(\{1,2,3\} < 2.5, \{5,6,7\})$$


---


$$\{5,6,\text{undef}\}$$

*Value\_If\_false* är ej specificerat. Undef används.

---


$$\text{ifFn}(\{2, "a" \} < 2.5, \{6,7\}, \{9,10\}, "err")$$


---


$$\{6, "err" \}$$

Ett valt element från *Value\_If\_true*. Ett valt element från *Value\_If\_unknown*.

**imag**(*ExprI*) uttryck

Ger argumentets imaginärdel.

---


$$\text{imag}(1+2 \cdot i)$$


---


$$2$$


---


$$\text{imag}(z)$$


---


$$0$$


---


$$\text{imag}(x+i \cdot y)$$


---


$$y$$



## imag()

Katalog > 

**Obs:** Alla odefinierade variabler behandlas som reella variabler. Se även `real()`, på sidan 151

**imag(ListI)**  $\Rightarrow$  lista

$$\text{imag}(\{-3, 4-i, i\}) \quad \{0, -1, 1\}$$

Ger en lista på elementens imaginärdelar.

**imag(MatrixI)**  $\Rightarrow$  matris

$$\text{imag}\left(\begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix}\right) \quad \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

Ger en matris med elementens imaginärdelar.

## impDif()

Katalog > 

**impDif(Equation, Var, dependVar [,Ord])**  $\Rightarrow$  uttryck

$$\text{impDif}(x^2 + y^2 = 100, x, y) \quad \begin{array}{l} -x \\ y \end{array}$$

där ordningen *Ord* förinställs på 1.

Beräknar den implicita derivatan för ekvationer i vilka en variabel är implicit definierad med termer från en annan.

## Indirection

Se #(), på sidan 236.

## inString()

Katalog > 

**inString(srcString, subString[, Start])**  $\Rightarrow$  heltal

$$\begin{array}{ll} \text{inString}(\text{"Hello there"}, \text{"the"}) & 7 \\ \text{inString}(\text{"ABCEFG"}, \text{"D"}) & 0 \end{array}$$

Ger teckenpositionen i strängen *srcString* där den första förekomsten av strängen *subString* börjar.

*Start*, om inkluderad, specificerar teckenpositionen inom *srcString* där sökningen börjar. Förinställning = 1 (det första tecknet i strängen *srcString*).

Återgår till noll om *srcString* inte innehåller *subString* eller om *Start* har en större längd än *srcString*.

**int()**Katalog > **int(Expr)** ⇒ *heltal***int(List1)** ⇒ *lista***int(Matrix1)** ⇒ *matris*

|                    |     |
|--------------------|-----|
| $\text{int}(-2.5)$ | -3. |
|--------------------|-----|

|                                                                          |                                              |
|--------------------------------------------------------------------------|----------------------------------------------|
| $\text{int}\left(\begin{bmatrix} -1.234 & 0 & 0.37 \end{bmatrix}\right)$ | $\begin{bmatrix} -2. & 0 & 0. \end{bmatrix}$ |
|--------------------------------------------------------------------------|----------------------------------------------|

Ger det största heltalet som är mindre än eller lika med argumentet. Denna funktion är identisk med **floor()**.

Argumentet kan vara ett reellt eller ett komplext tal.

Ger, för en lista eller matris, det största heltalet för varje element.

**intDiv()**Katalog > **intDiv(Number1, Number2)** ⇒ *heltal***intDiv(List1, List2)** ⇒ *lista***intDiv(Matrix1, Matrix2)** ⇒ *matris*

|                       |    |
|-----------------------|----|
| $\text{intDiv}(-7,2)$ | -3 |
|-----------------------|----|

|                      |   |
|----------------------|---|
| $\text{intDiv}(4,5)$ | 0 |
|----------------------|---|

|                                                 |                |
|-------------------------------------------------|----------------|
| $\text{intDiv}(\{12, -14, -16\}, \{5, 4, -3\})$ | $\{2, -3, 5\}$ |
|-------------------------------------------------|----------------|

Ger heltalsdelen med tecken av ( $\text{Number1} \div \text{Number2}$ ).

Ger, för listor och matriser, heltalsdelen med tecken av ( $\text{argument1} \div \text{argument2}$ ) för varje elementpar.

**integral**Se  $\int()$ , på sidan 231.**Interpolera ()**Katalog > **Interpolera(xValue, xList, yList, yPrimeList)** ⇒ *lista*

Denna funktion gör följande:

Differentiallekvation:

 $y' = -3 \cdot y + 6 \cdot t + 5$  och  $y(0) = 5$ 

|                                                                                                            |
|------------------------------------------------------------------------------------------------------------|
| $rk = rk23(-3 \cdot y + 6 \cdot t + 5, t, y, \{0, 10\}, 5, 1)$                                             |
| $\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 5. & 3.19499 & 5.00394 & 6.99957 & 9.00593 & 10. \end{bmatrix}$ |

För att se hela resultatet, tryck på  och använd sedan  och  för att flytta markören.

## Interpolera ()

Katalog > 

Förutsatt  $xList$ ,  $yList=f(xList)$  och  $yPrimeList=f'(xList)$  används för en okänd funktion  $f$  en kubisk interpolant för att uppskatta funktionen  $f$  vid  $xValue$ . Det förutsätts att  $xList$  är en lista på monotont ökande eller minskande tal, men denna funktion kan ge ett värde även när listan inte uppfyller förutsättningarna. Denna funktion går igenom  $xList$  och söker ett intervall  $[xList[i], xList[i+1]]$  som innehåller  $xValue$ . Om funktionen hittar ett sådant intervall ger den ett interpolerat värde på  $f(xValue)$ , annars ger den **undef**.

$xList$ ,  $yList$  och  $yPrimeList$  måste ha samma dimension  $\geq 2$  och innehålla uttryck som förenklas till tal.

$xValue$  kan vara en odefinierad variabel, ett tal eller en lista av tal.

Använd funktionen `interpolate()` för att beräkna funktionsvärdena för  $xvalueList$ :

```
xvalueList:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7.,7.5,8.,8.5,9.,9.5,10.}

xlist:=mat▶list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}

ylist:=mat▶list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9979}

yprimelist:=-3*y+6*t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.00679}

interpolate(xvalueList,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011,7.00000,8.00000,9.00000,10.0000}
```

## inv $\chi^2$ ()

Katalog > 

**inv $\chi^2$ (Area,df)**

**invChi2(Area,df)**

Beräknar den inversa kumulativa sannolikhetsfunktionen  $\chi^2$  (chi-kvadrat) specificerad av frihetsgraden  $df$  för en given  $Area$  under kurvan.

## invF()

Katalog > 

**invF(Area,dfNumer,dfDenom)**

**invF(Area,dfNumer,dfDenom)**

beräknar den inversa kumulativa fördelningsfunktionen  $F$  specificerad av  $dfNumer$  och  $dfDenom$  för en given  $Area$  under kurvan.

**invBinom**

*(CumulativeProb, NumTrials, Prob, OutputForm)* ⇒ skalär eller matris

Baserat på antalet försök (*NumTrials*) och sannolikheten för önskat utfall av varje försök (*Prob*) ger denna funktion det minimala antalet lyckade utfall, *k*, så att den kumulativa sannolikheten, *k*, är större än eller lika med den givna kumulativa sannolikheten (*CumulativeProb*).

*OutputForm=0*, visar resultatet som en skalär (förvalt).

*OutputForm=1*, visar resultatet som en matris.

Exempel: Marie and Kalle spelar ett tärningsspel. Marie ska gissa hur många gånger som mest tärningen visar en sexa under 30 kast. Om tärningen ger en sexa detta antal gånger eller mindre, vinner Marie. Dessutom får Marie högre poäng ju mindre antal sexor hon gissar. Vilket är det minsta antal gånger Marie kan gissa om hon vill att sannolikheten att vinna ska vara högre än 77 %?

|                                                        |                                                              |
|--------------------------------------------------------|--------------------------------------------------------------|
| $\text{invBinom}\left(0.77, 30, \frac{1}{6}\right)$    | 6                                                            |
| $\text{invBinom}\left(0.77, 30, \frac{1}{6}, 1\right)$ | $\begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix}$ |

**invBinomN()**

*invBinomN(CumulativeProb, Prob, NumSuccess, OutputForm)* ⇒ skalär eller matris

Baserat på sannolikheten för lyckat utfall i varje försök (*Prob*) och antalet lyckade försök (*NumSuccess*) beräknar funktionen det minsta antalet försök, *N*, så att den kumulativa sannolikheten för *x* lyckade utfall är mindre eller lika med den givna kumulativ sannolikheten (*CumulativeProb*).

*OutputForm=0*, visar resultatet som en skalär (förvalt).

*OutputForm=1*, visar resultatet som en matris.

Exempel: Monika övar målskick i basket. Hon vet av erfarenhet att chansen att göra mål är 70 % i varje skott. Hon bestämmer sig för att öva tills hon har gjort 50 mål. Hur många försök måste hon göra för att sannolikheten för att göra minst 50 mål ska vara högre än 0,99?

|                                      |                                                               |
|--------------------------------------|---------------------------------------------------------------|
| $\text{invBinomN}(0.01, 0.7, 49)$    | 86                                                            |
| $\text{invBinomN}(0.01, 0.7, 49, 1)$ | $\begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$ |

**invNorm()**

*invNorm(Area, μ, σ)*

Beräknar den inversa kumulativa normalfördelningen för en given *Area* under normalfördelningskurvan specificerad av  $\mu$  och  $\sigma$ .

**invt**(*Area*,*df*)

Beräknar den inversa kumulativa student-t-fördelningsfunktionen specificerad av frihetsgraden, *df*, för en given *Area* under kurvan.

**iPart**(*Number*) ⇒ *heltal***iPart**(*List1*) ⇒ *lista***iPart**(*Matrix1*) ⇒ *matris*

|                                                                   |                  |
|-------------------------------------------------------------------|------------------|
| $\text{iPart}(-1.234)$                                            | -1.              |
| $\text{iPart}\left(\left\{\frac{3}{2}, 2.3, 7.003\right\}\right)$ | $\{1, -2., 7.\}$ |

Ger argumentets heltalsdel.

Ger, för listor och matriser, heltalsdelen för varje element.

Argumentet kan vara ett reellt eller ett komplext tal.

**irr**(*CF0*,*CFList* [,*CFFreq*]) ⇒ *värde*

Ekonomifunktion som beräknar internräntan på en investering.

*CF0* är det initiala kassaflödet vid tidpunkt 0 och måste vara ett reellt tal.

*CFList* är en lista på kassaflödesbelopp efter det initiala kassaflödet *CF0*.

*CFFreq* är en frivillig lista i vilken varje element specificerar frekvensen för ett grupperat (konsekutivt) kassaflödesbelopp, vilket är det motsvarande elementet i *CFList*. Förinställningen är 1. Om du vill mata in värden måste de vara positiva heltal <10 000.

**Obs:** Se även **mirr()**, på sidan 120.

|                                                |                                |
|------------------------------------------------|--------------------------------|
| $\text{list1} := \{6000, -8000, 2000, -3000\}$ | $\{6000, -8000, 2000, -3000\}$ |
| $\text{list2} := \{2, 2, 2, 1\}$               | $\{2, 2, 2, 1\}$               |
| $\text{irr}(5000, \text{list1}, \text{list2})$ | -4.64484                       |

## isPrime()

Katalog > 

**isPrime(*Number*)** ⇒ *Booleskt konstantuttryck*

Ger sant eller falskt för att indikera om *number* är ett heltal  $\geq 2$  som är jämnt delbart endast med sig självt och 1.

Om *Number* överskrider cirka 306 siffror och saknar faktorer  $\leq 1021$ , visar **isPrime(*Number*)** ett felmeddelande.

Om du endast vill bestämma om *Number* är ett primtal, använd **isPrime()** i stället för **factor()**. Detta går mycket fortare, särskilt om *Number* inte är ett primtal och dess näst största faktor har mer än cirka fem siffror.

**Obs för att mata in exemplet:** Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|            |       |
|------------|-------|
| isPrime(5) | true  |
| isPrime(6) | false |

Funktion för att hitta nästa primtal efter ett specificerat tal:

|                                          |      |
|------------------------------------------|------|
| Define <i>nextprim</i> ( <i>n</i> )=Func | Done |
| Loop                                     |      |
| <i>n</i> + 1 → <i>n</i>                  |      |
| If isPrime( <i>n</i> )                   |      |
| Return <i>n</i>                          |      |
| EndLoop                                  |      |
| EndFunc                                  |      |
| <i>nextprim</i> (7)                      | 11   |

## isVoid()

Katalog > 

**isVoid(*Var*)** ⇒ *Booleskt konstantuttryck*  
**isVoid(*Expr*)** ⇒ *Booleskt konstantuttryck*  
**isVoid(*List*)** ⇒ *lista av Booleska konstantuttryck*

Ger sant eller falskt för att indikera huruvida argumentet är en tom datatyp.

För mer information om tomma element, se på sidan 261.

|                    |                      |
|--------------------|----------------------|
| <i>a</i> := _      | _                    |
| isVoid( <i>a</i> ) | true                 |
| isVoid({ 1,_,3 })  | { false,true,false } |

| Lbl                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Katalog >                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Lbl</b> <i>labelName</i></p> <p>Definierar en etikett med namnet <i>labelName</i> inom en funktion.</p> <p>Du kan använda en <b>Goto</b> <i>labelName</i>-instruktion för att överföra kontroll till instruktionen direkt efter etiketten.</p> <p><i>labelName</i> måste uppfylla samma krav på namngivning som ett variabelnamn.</p> <p><b>Obs för att mata in exemplet:</b> Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.</p> | <pre> Define g()<math>\Rightarrow</math>Func     Local temp,i     0 <math>\rightarrow</math> temp     1 <math>\rightarrow</math> i     Lbl top     temp+i <math>\rightarrow</math> temp     If i&lt;10 Then         i+1 <math>\rightarrow</math> i     Goto top     EndIf     Return temp EndFunc                     </pre> <div> <div>g()</div> <div>55</div> </div> |

| lcm()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Katalog >                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>lcm</b>(<i>Number1</i>, <i>Number2</i>)<math>\Rightarrow</math>uttryck</p> <p><b>lcm</b>(<i>List1</i>, <i>List2</i>)<math>\Rightarrow</math>lista</p> <p><b>lcm</b>(<i>Matrix1</i>, <i>Matrix2</i>)<math>\Rightarrow</math>matrix</p> <p>Ger den minsta gemensamma multipeln för de två argumenten. <b>lcm</b> för två bråk är <b>lcm</b> för deras täljare dividerat med <b>gcd</b> för deras nämnare. <b>lcm</b> för tal i flyttalsform är deras produkt.</p> <p>Ger, för två listor eller matriser, den minsta gemensamma multipeln för de motsvarande elementen.</p> | <pre> lcm(6,9)                     </pre> <div> <div>lcm(<math>\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}</math>)</div> <div>18</div> </div> <div> <div>lcm(<math>\left\{\frac{2}{3}, 14, 80\right\}</math>)</div> <div></div> </div> |

| left()                                                                                                                                                                                                                                                                                                                        | Katalog >                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>left</b>(<i>sourceString</i>[, <i>Num</i>])<math>\Rightarrow</math>sträng</p> <p>Ger <i>Num</i>-tecknen längst till vänster i teckensträngen <i>sourceString</i>.</p> <p>Om du utelämnar <i>Num</i> erhålls alla i <i>sourceString</i>.</p> <p><b>left</b>(<i>List1</i>[, <i>Num</i>])<math>\Rightarrow</math>lista</p> | <pre> left("Hello",2)                     </pre> <div> <div>left(<math>\left\{1, 3, -2, 4\right\}, 3</math>)</div> <div>"He"</div> </div> <div> <div>left(<math>\left\{1, 3, -2, 4\right\}, 3</math>)</div> <div><math>\left\{1, 3, -2\right\}</math></div> </div> |

Ger *Num*-elementen längst till vänster i *List1*.

Om du utelämnar *Num* erhålls alla i *List1*.

**left**(*Comparison*) $\Rightarrow$ uttryck

Ger den vänstra sidan av en ekvation eller olikhet.

---


$$\text{left}(x < 3) \qquad x$$


---

## libShortcut()

**libShortcut**(*LibNameString*,  
*ShortcutNameString*

[, *LibPrivFlag*]) $\Rightarrow$ lista på variabler

Skapar en variabelgrupp i det aktuella problemet som innehåller referenser till alla objekt i det specificerade biblioteksdokumentet *libNameString*. Läger också till gruppmedlemmarna på menyn Variables. Du kan sedan referera till varje objekt med hjälp av dess *ShortcutNameString*.

Ställ *LibPrivFlag*=0 för att utesluta privata biblioteksobjekt (förinställning)

Ställ *LibPrivFlag*=1 för att inkludera privata biblioteksobjekt

För att kopiera en variabelgrupp, se **CopyVar**, på sidan 29.

För att ta bort en variabelgrupp, se **DelVar**, på sidan 50.

Detta exempel förutsätter ett korrekt lagrat och uppdaterat biblioteksdokument med namnet **linalg2** och som innehåller objekt definierade som *clearmat*, *gauss1* och *gauss2*.

---

```
getVarInfo("linalg2")
  [clearmat  "FUNC"  "LibPub "]
  [gauss1    "PRGM"  "LibPriv "]
  [gauss2    "FUNC"  "LibPub "]
libShortcut("linalg2","la")
  {la.clearmat,la.gauss2}
libShortcut("linalg2","la",1)
  {la.clearmat,la.gauss1,la.gauss2}
```

---



**limit**(*Expr1*, *Var*, *Point* [, *Direction*]) $\Rightarrow$ uttryck

**limit**(*List1*, *Var*, *Point* [, *Direction*]) $\Rightarrow$ lista

**limit**(*Matrix1*, *Var*, *Point* [, *Direction*]) $\Rightarrow$ matris

Ger det begärda gränsvärdet.

**Obs:** Se även **Limit template**, på sidan 6.

*Direction* (Riktning): negativ=från vänster, positiv=från höger, annars=båda. (Om *Direction* utelämnas förinställs den till båda.)

Gränsvärden vid positiv  $\infty$  och vid negativ  $\infty$  konverteras alltid till ensidiga gränsvärden från den ändliga sidan.

Beroende på omständigheterna ger **limit** () sig självt eller undef när den inte kan bestämma ett unikt gränsvärde. Detta innebär nödvändigtvis inte att ett unikt gränsvärde inte existerar. "undef" innebär att resultatet antingen är ett okänt tal med ändlig eller oändlig storlek, eller hela uppsättningen av sådana tal.

**limit**() använder metoder såsom L'Hopital's regel, så det finns unika gränsvärden som den inte kan bestämma. Om *Expr1* innehåller odefinierade variabler utöver *Var* kan du behöva begränsa dem för att erhålla ett mer kortfattat resultat.

Gränsvärden kan vara mycket känsliga för avrundningsfel. Undvik om möjligt inställningen Approximate (Ungefärlig) i läge **Auto** eller **Ungefärlig** och ungefärliga tal när du beräknar gränsvärden. Annars har gränsvärden som skulle vara noll eller ha oändlig storlek sannolikt inte dessa egenskaper, och gränsvärden som skulle ha ändlig icke nollstorlek kanske inte har detta.

|                                                                               |           |
|-------------------------------------------------------------------------------|-----------|
| $\lim_{x \rightarrow 5} (2 \cdot x + 3)$                                      | 13        |
| $\lim_{x \rightarrow 0^+} \left( \frac{1}{x} \right)$                         | $\infty$  |
| $\lim_{x \rightarrow 0} \left( \frac{\sin(x)}{x} \right)$                     | 1         |
| $\lim_{h \rightarrow 0} \left( \frac{\sin(x+h) - \sin(x)}{h} \right)$         | $\cos(x)$ |
| $\lim_{n \rightarrow \infty} \left( \left( 1 + \frac{1}{n} \right)^n \right)$ | e         |

|                                                                |          |
|----------------------------------------------------------------|----------|
| $\lim_{x \rightarrow \infty} (a^x)$                            | undef    |
| $\lim_{x \rightarrow \infty} (a^x)   a > 1$                    | $\infty$ |
| $\lim_{x \rightarrow \infty} (a^x)   a > 0 \text{ and } a < 1$ | 0        |

**LinRegBx**  $X, Y, [Freq], [Category, Include]$

Utför den linjära regressionsanalysen  $y = a + b \cdot x$  på listorna  $X$  och  $Y$  med frekvensen  $Freq$ . En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

$Freq$  är en frivillig lista på frekvensvärden. Varje element i  $Freq$  specificerar frekvensen för varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

$Category$  är en lista på kategorikoder för motsvarande  $X$ - och  $Y$ -data.

$Include$  är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                                                                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn         | Regressionsekvation: $a + b \cdot x$                                                                                                                    |
| stat.a, stat.b      | Regressionskoefficienter                                                                                                                                |
| stat.r <sup>2</sup> | Determinationskoefficient                                                                                                                               |
| stat.r              | Korrelationskoefficient                                                                                                                                 |
| stat.Resid          | Residualer från regressionen                                                                                                                            |
| stat.XReg           | Lista på datapunkter i den modifierade $X$ List som används i regressionen baserat på begränsningar i $Freq$ , $Category$ List och $Include$ Categories |
| stat.YReg           | Lista på datapunkter i den modifierade $Y$ List som används i regressionen baserat på begränsningar i $Freq$ , $Category$ List och $Include$ Categories |
| stat.FreqReg        | Lista på frekvenser som motsvarar $stat.XReg$ och $stat.YReg$                                                                                           |

**LinRegMx**  $X, Y, [Freq], [Category, Include]$ 

Beräknar den linjära regressionen  $y = m \cdot x + b$  på listorna  $X$  och  $Y$  med frekvensen  $Freq$ . En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande  $X$ - och  $Y$ -data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn         | Regressionsekvation: $m \cdot x + b$                                                                                                                                        |
| stat.m, stat.b      | Regressionskoefficienter                                                                                                                                                    |
| stat.r <sup>2</sup> | Determinationskoefficient                                                                                                                                                   |
| stat.r              | Korrelationskoefficient                                                                                                                                                     |
| stat.Resid          | Residualer från regressionen                                                                                                                                                |
| stat.XReg           | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg           | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg        | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

**LinRegtIntervals**  $X, Y[, F[, 0[, CLev]]]$

För Slope (Lutning). Beräknar ett nivå-C-konfidensintervall för lutningen.

**LinRegtIntervals**  $X, Y[, F[, 1, Xval[, CLev]]]$

För Response (Svar). Beräknar ett prognostiserat  $y$ -värde, ett nivå-C-prediktionsintervall för en enskilda observation och ett nivå-C-konfidensintervall för medelvärde på svaret.

En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

$F$  är en frivillig lista på frekvensvärden. Varje element i  $F$  specificerar förekomsten av varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                          |
|---------------------|--------------------------------------|
| stat.RegEqn         | Regressionsekvation: $a + b \cdot x$ |
| stat.a, stat.b      | Regressionskoefficienter             |
| stat.df             | Frihetsgrader                        |
| stat.r <sup>2</sup> | Determinationskoefficient            |
| stat.r              | Korrelationskoefficient              |
| stat.Resid          | Residualer från regressionen         |

Endast för typen Slope

| Resultatvariabel           | Beskrivning                      |
|----------------------------|----------------------------------|
| [stat.CLower, stat.CUpper] | Konfidensintervall för lutningen |
| stat.ME                    | Konfidensintervall - felmarginal |

| Resultatvariabel | Beskrivning             |
|------------------|-------------------------|
| stat.SESlope     | Standardfel hos lutning |
| stat.s           | Standardfel hos linjen  |

Endast för typen Response

| Resultatvariabel                 | Beskrivning                                     |
|----------------------------------|-------------------------------------------------|
| [stat.CLower, stat.CUpper]       | Konfidensintervall för medelvärde på svaret     |
| stat.ME                          | Konfidensintervall - felmarginal                |
| stat.SE                          | Standardfel hos medelsvar                       |
| [stat.LowerPred, stat.UpperPred] | Prediktionsintervall för en enskild observation |
| stat.MEPred                      | Prognostiseringsintervall - felmarginal         |
| stat.SEPred                      | Standardfel för prognostisering                 |
| stat. $\hat{y}$                  | $a + b \cdot X_{\text{Val}}$                    |

## LinRegtTest

Katalog > 

**LinRegtTest**  $X, Y[, Freq[, Hypoth]]$

Utför en linjär regressionsanalys på listorna  $X$  och  $Y$  och ett  $t$ -test på lutningens värde  $\beta$  samt korrelationskoefficienten  $\rho$  för ekvationen  $y = \alpha + \beta x$ . Det testar nollhypotesen  $H_0: \beta = 0$  (equivalently,  $\rho = 0$ ) mot en av tre alternativa hypoteser.

Alla listor måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

$Freq$  är en frivillig lista på frekvensvärden. Varje element i  $Freq$  specificerar frekvensen för varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

$Hypoth$  är ett valfritt värde som specificerar en av tre alternativa hypoteser mot vilka nollhypotesen ( $H_0: \beta = \rho = 0$ ) kommer att testas.

För  $H_a: \beta \neq 0$  och  $\rho \neq 0$  (förinställning), ställ  $H_{youth}=0$

För  $H_a: \beta < 0$  och  $\rho < 0$ , ställ  $H_{youth} < 0$

För  $H_a: \beta > 0$  och  $\rho > 0$ , ställ  $H_{youth} > 0$

En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                                                   |
|---------------------|---------------------------------------------------------------|
| stat.RegEqn         | Regressionsekvation: $a + b \cdot x$                          |
| stat.t              | <i>t</i> -Statistik för signifikanstest                       |
| stat.PVal           | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas |
| stat.df             | Frihetsgrader                                                 |
| stat.a, stat.b      | Regressionskoefficienter                                      |
| stat.s              | Standardfel hos linjen                                        |
| stat.SESlope        | Standardfel hos lutning                                       |
| stat.r <sup>2</sup> | Determinationskoefficient                                     |
| stat.r              | Korrelationskoefficient                                       |
| stat.Resid          | Residualer från regressionen                                  |

**linSolve()**Katalog > 

**linSolve**(*SystemAvLinjäraEkv*, *Var1*, *Var2*, ...) ⇒ lista

$$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x + 4 \cdot y = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}\right\}, \{x, y\}\right) \quad \left\{\begin{array}{l} \frac{37}{26}, \frac{1}{26} \end{array}\right\}$$

**linSolve**(*LinjärEkv1* och *LinjärEkv2* och ..., *Var1*, *Var2*, ...) ⇒ lista

$$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}\right\}, \{x, y\}\right) \quad \left\{\begin{array}{l} \frac{3}{2}, \frac{1}{6} \end{array}\right\}$$

**linSolve**(*{LinjärEkv1, LinjärEkv2, ...}*, *Var1*, *Var2*, ...) ⇒ lista

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} + 4 \cdot \text{pear} = 23 \\ 5 \cdot \text{apple} - \text{pear} = 17 \end{array}\right\}, \{\text{apple}, \text{pear}\}\right) \quad \left\{\begin{array}{l} \frac{13}{3}, \frac{14}{3} \end{array}\right\}$$

**linSolve**(*SystemAvLinjäraEkv*, *{Var1, Var2, ...}*) ⇒ lista

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{array}\right\}, \{\text{apple}, \text{pear}\}\right) \quad \left\{\begin{array}{l} \frac{36}{13}, \frac{114}{13} \end{array}\right\}$$

**linSolve**(*LinjärEkv1* och *LinjärEkv2* och ..., *{Var1, Var2, ...}*) ⇒ lista

**linSolve**(*{LinjärEkv1, LinjärEkv2, ...}*, *{Var1, Var2, ...}*) ⇒ lista

Ger en lista på lösningar för variablerna *Var1*, *Var2*, ...

Det första argumentet måste beräknas till ett system av linjära ekvationer eller en enda linjär ekvation. Annars inträffar ett argumentfel.

Som exempel leder beräkningen

**linSolve**(**x=1** och **x=2, x**) till ett "Argumentfel"-resultat.

**ΔList()**Katalog > 

**ΔList**(*List1*) ⇒ lista

$$\Delta\text{List}(\{20, 30, 45, 70\}) \quad \{10, 15, 25\}$$

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **deltaList**(...).

Ger en lista på skillnaderna mellan konsekutiva element i *List1*. Varje element i *List1* subtraheras från nästa element i *List1*. Den resulterande listan är alltid ett element kortare än den ursprungliga *List1*.

## list►mat()

Katalog > 

**list►mat**(*List* [, *elementsPerRow*])⇒*matrix*

Ger en matrix fylld rad efter rad med elementen från *List*.

*elementsPerRow*, om inkluderad, specificerar antalet element per rad. Förinställningen är antalet element i *List* (en rad).

Om *List* inte fyller den resulterande listan läggs nollor till.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **list@>mat(...)**.

|                                                             |                                                         |
|-------------------------------------------------------------|---------------------------------------------------------|
| $\text{list}\blacktriangleright\text{mat}(\{1,2,3\})$       | $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$               |
| $\text{list}\blacktriangleright\text{mat}(\{1,2,3,4,5\},2)$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$ |

## ►ln

Katalog > 

*Expr* ►ln⇒*uttryck*

Medför att inmatningen *Expr* konverteras till ett uttryck som endast innehåller naturliga logaritmer (ln).

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **@>ln**.

|                                                   |                          |
|---------------------------------------------------|--------------------------|
| $\left(\log_{10}(x)\right)\blacktriangleright\ln$ | $\frac{\ln(x)}{\ln(10)}$ |
|---------------------------------------------------|--------------------------|

## ln()

  **tangenter**

**ln**(*ExprI*)⇒*uttryck*

|           |          |
|-----------|----------|
| $\ln(2.)$ | 0.693147 |
|-----------|----------|

**ln**(*ListI*)⇒*lista*

Ger argumentets naturliga logaritm.

Ger, för en lista, elementens naturliga logaritmer.

Om det komplexa formatläget är Real:

|                               |
|-------------------------------|
| $\ln(\{-3,1.2,5\})$           |
| "Error: Non-real calculation" |

Om det komplexa formatläget är Rectangular:

|                     |                                         |
|---------------------|-----------------------------------------|
| $\ln(\{-3,1.2,5\})$ | $\{\ln(3)+\pi\cdot i,0.182322,\ln(5)\}$ |
|---------------------|-----------------------------------------|

**ln**(*squareMatrixI*)⇒*kvadratMatrix*

I vinkelläget Radianer och i Rektangulärt komplext format:



Ger matrisen med naturlig logaritm för *squareMatrix1*. Detta är inte detsamma som att beräkna den naturliga logaritmen för varje element. För information om beräkningsmetoden, se **cos()**.

$$\ln \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

## LnReg

Katalog > 

**LnReg** *X*, *Y*, [*Freq*] [, *Category*, *Include*]

Utför en logaritmisk regressionsanalys  $y = a + b \cdot \ln(x)$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                               |
|------------------|-------------------------------------------|
| stat.RegEqn      | Regressionsekvation: $a + b \cdot \ln(x)$ |
| stat.a, stat.b   | Regressionskoefficienter                  |

| Resultatvariabel    | Beskrivning                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.r <sup>2</sup> | Koefficient för linjär bestämning av transformerade data                                                                                                                    |
| stat.r              | Korrelationskoefficient för transformerade data (ln(x), y)                                                                                                                  |
| stat.Resid          | Residualer associerade med den logaritmiska modellen                                                                                                                        |
| stat.ResidTrans     | Residualer associerade med linjär anpassning av transformerade data                                                                                                         |
| stat.XReg           | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg           | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg        | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

Local

Katalog > 

**Local** *Var1* [, *Var2*] [, *Var3*] ...

Betecknar specificerade *vars* som lokala variabler. Dessa variabler existerar endast under utvärderingen av ett uttryck och tas bort när exekveringen av uttrycket är klar.

**Obs:** Lokala variabler sparar minne eftersom de endast existerar tillfälligt. De stör heller inga befintliga globala variabelvärden. Lokala variabler måste användas för **For**-slingor och för att temporärt spara värden i en flerradig funktion eftersom modifieringar av globala värden inte är tillåtna i en funktion.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|                                 |      |
|---------------------------------|------|
| Define <i>rollcount()</i> =Func |      |
| Local <i>i</i>                  |      |
| 1 → <i>i</i>                    |      |
| Loop                            |      |
| If randInt(1,6)=randInt(1,6)    |      |
| Goto <i>end</i>                 |      |
| <i>i</i> +1 → <i>i</i>          |      |
| EndLoop                         |      |
| Lbl <i>end</i>                  |      |
| Return <i>i</i>                 |      |
| EndFunc                         |      |
|                                 | Done |
| <i>rollcount()</i>              | 16   |
| <i>rollcount()</i>              | 3    |

**Lock***Var1* [, *Var2*] [, *Var3*] ...

**Lock***Var*.

Låser den specificerade variabeln eller variabelgruppen. Låsta variabler kan inte modifieras eller tas bort.

Du kan inte låsa eller låsa upp systemvariabeln *Ans* och du kan inte låsa systemvariabelgrupperna *stat.* och *tvm.*

**Obs:** Kommandot **Lås (Lock)** renser Ångra/Upprepa-historiken när det används på olåsta variabler.

Se **unLock**, på sidan 206 och **getLockInfo()**, på sidan 86.

|                         |                              |
|-------------------------|------------------------------|
| <i>a</i> :=65           | 65                           |
| Lock <i>a</i>           | Done                         |
| getLockInfo( <i>a</i> ) | 1                            |
| <i>a</i> :=75           | "Error: Variable is locked." |
| DelVar <i>a</i>         | "Error: Variable is locked." |
| Unlock <i>a</i>         | Done                         |
| <i>a</i> :=75           | 75                           |
| DelVar <i>a</i>         | Done                         |

## log()

  **tangenter**

**log**(*Expr1* [, *Expr2*]) ⇒ *uttryck*

|                  |         |
|------------------|---------|
| $\log_{10} (2.)$ | 0.30103 |
|------------------|---------|

**log**(*List1* [, *Expr2*]) ⇒ *lista*

|               |     |
|---------------|-----|
| $\log_4 (2.)$ | 0.5 |
|---------------|-----|

Ger bas-*Expr2*-logaritmen för det första argumentet.

|                            |              |
|----------------------------|--------------|
| $\log_3 (10) - \log_3 (5)$ | $\log_3 (2)$ |
|----------------------------|--------------|

**Obs:** Se även **Log template**, på sidan 2.

Ger, för en lista, bas-*Expr2*-logaritmen för elementen.

Om det komplexa formatläget är Real:

|                              |                        |
|------------------------------|------------------------|
| $\log_{10} (\{-3, 1.2, 5\})$ | Error: Non-real result |
|------------------------------|------------------------|

Om det andra argumentet utelämnas används 10 som bas.

Om det komplexa formatläget är Rectangular:

|                              |                                                                             |
|------------------------------|-----------------------------------------------------------------------------|
| $\log_{10} (\{-3, 1.2, 5\})$ | $\left\{ \log_{10} (3) + 1.36438 \cdot i, 0.079181, \log_{10} (5) \right\}$ |
|------------------------------|-----------------------------------------------------------------------------|

**log**(*squareMatrix1* [, *Expr*]) ⇒ *kvadratMatris*

I vinkelläget Radianer och i Rektangulärt komplext format:

## log()

ctrl 10<sup>x</sup> tangenter

Ger matrisen med bas-*Expr*-logaritm för *squareMatrix1*. Detta är inte detsamma som att beräkna bas-*Expr*-logaritmen för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

Om basargumentet utelämnas används 10 som bas.

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.795387+0.753438 \cdot i & 0.003993-0.6474 \cdot i \\ 0.194895-0.315095 \cdot i & 0.462485+0.2707 \cdot i \\ -0.115909-0.904706 \cdot i & 0.488304+0.7774 \cdot i \end{bmatrix}$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

## ►logbase

Katalog >

*Expr* ►**logbase**(*Expr1*)⇒uttryck

Medför att inmatningen Expression förenklas till ett uttryck med basen *Expr1*.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **@>logbase (...)**.

$$\log_3(10) - \log_5(5) \text{ ► } \log_{\text{base}(5)} \left( \frac{\log_5\left(\frac{10}{3}\right)}{\log_5(3)} \right)$$

## Logistic

Katalog >

**Logistic** *X*, *Y*, [*Freq*] [, *Category*, *Include*]

Utför den logistiska regressionsanalysen  $y = (c/(1+a \cdot e^{-bx}))$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel       | Beskrivning                                                                                                                                                                 |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn            | Regressionsekvation: $c/(1+a \cdot e^{-bx})$                                                                                                                                |
| stat.a, stat.b, stat.c | Regressionskoefficienter                                                                                                                                                    |
| stat.Resid             | Residualer från regressionen                                                                                                                                                |
| stat.XReg              | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg              | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg           | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## LogisticD

**LogisticD** *X*, *Y* [, [*Iterations*], [*Freq*] [, *Category*, *Include*] ]

Utför den logistiska regressionsanalysen  $y = (c/(1+a \cdot e^{-bx})+d)$  på listorna *X* och *Y* med frekvensen *Freq*, med ett specificerat antal *Iterationer*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Iterations* är ett valfritt värde som specificerar det maximala antalet gånger en lösning kommer att provas. Om denna utelämnas används 64. Normalt ger större värden bättre noggrannhet, men längre exekveringstider, och vice versa.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel               | Beskrivning                                                                                                                                                                 |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn                    | Regressionsekvation: $c/(1+a \cdot e^{-bx})+d$                                                                                                                              |
| stat.a, stat.b, stat.c, stat.d | Regressionskoefficienter                                                                                                                                                    |
| stat.Resid                     | Residualer från regressionen                                                                                                                                                |
| stat.XReg                      | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg                      | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg                   | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## Loop

*Block*

## EndLoop

Exekverar påståendena i *Block* upprepade gånger. Observera att slingan upprepas i all oändlighet såvida inte en **Goto**- eller **Exit**-instruktion exekveras inom *Block*.

*Block* är en serie av påståenden separerade med tecknet ":".

**Obs för att mata in exemplet:** Se avsnittet Räkna i produktboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|                              |      |
|------------------------------|------|
| Define rollcount()=Func      |      |
| Local i                      |      |
| 1 → i                        |      |
| Loop                         |      |
| If randInt(1,6)=randInt(1,6) |      |
| Goto end                     |      |
| i+1 → i                      |      |
| EndLoop                      |      |
| Lbl end                      |      |
| Return i                     |      |
| EndFunc                      |      |
|                              | Done |
| rollcount()                  | 16   |
| rollcount()                  | 3    |

## LU

**LU** *Matris, lMatris, uMatris, pMatris*  
[,Tol]

Beräknar uppdelningen Doolittle LU (undre-övre) av en reell eller komplex matris. Den undertriangulära matrisen lagras i *lMatris*, den övertriangulära matrisen lagras i *uMatris* och permutationsmatrisen (som beskriver radväxlingarna som har gjorts under beräkningen) lagras i *pMatris*.

$$lMatris \cdot uMatris = pMatris \cdot matris$$

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tol*. Denna tolerans används endast om matrisen har inmatning med tal i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tol*.

- Om du använder   eller ställer in **Auto** eller **Ungefärlig** på Approximate utförs beräkningarna med flyttalsaritmetik.

|                                                                                         |                                                                                                   |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| $\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$                          |
| LU m1,lower,upper,perm                                                                  | Done                                                                                              |
| lower                                                                                   | $\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$ |
| upper                                                                                   | $\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$                            |
| perm                                                                                    | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$                               |

- Om *Tol* utelämnas eller inte används beräknas standardtoleransen som:  
 $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rowNorm}(Matrix)$

Faktoriseringsalgoritmen **LU** använder partiell pivotering med radutbyten.

|                                                               |                                                                      |
|---------------------------------------------------------------|----------------------------------------------------------------------|
| $\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} m & n \\ o & p \end{bmatrix}$                       |
| LU m1,lower,upper,perm                                        | Done                                                                 |
| lower                                                         | $\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$             |
| upper                                                         | $\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$ |
| perm                                                          | $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$                       |

## M

### mat▶list()

**mat▶list(Matrix)**⇒lista

Ger en lista med elementen i *Matrix*. Elementen kopieras från *Matrix* rad för rad.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **mat@>list(...)**.

|                                                                       |                                                        |
|-----------------------------------------------------------------------|--------------------------------------------------------|
| mat▶list( $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ )                 | $\{1, 2, 3\}$                                          |
| $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ |
| mat▶list( <i>m1</i> )                                                 | $\{1, 2, 3, 4, 5, 6\}$                                 |

### max()

**max(Expr1, Expr2)**⇒uttryck

**max(List1, List2)**⇒lista

**max(Matrix1, Matrix2)**⇒matrix

Ger de två argumentens maximum. Ger, om argumenten är två listor eller matriser, en lista eller matris som innehåller maximumvärdet för varje par av motsvarande element.

**max(List)**⇒uttryck

Ger maximelementet i *list*.

**max(Matrix1)**⇒matrix

|                              |            |
|------------------------------|------------|
| max(2.3, 1.4)                | 2.3        |
| max( $\{1, 2\}, \{-4, 3\}$ ) | $\{1, 3\}$ |

|                                 |     |
|---------------------------------|-----|
| max( $\{0, 1, -7, 1.3, 0.5\}$ ) | 1.3 |
|---------------------------------|-----|

|                                                                   |                                           |
|-------------------------------------------------------------------|-------------------------------------------|
| max( $\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}$ ) | $\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$ |
|-------------------------------------------------------------------|-------------------------------------------|



Ger en radvektor som innehåller maximumelementet för varje kolumn i *Matrix1*.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

**Obs:** Se även **fMax()** och **min()**.

## mean()

**mean(List[,freqList])**⇒*uttryck*

Ger medelvärde för elementen i *List*.

Varje *freqList*-element räknar antalet förekomster av motsvarande element i *List*.

**mean(Matrix1[,freqMatrix])**⇒*matrix*

Ger en radvektor med medelvärdena för alla kolumner i *Matrix1*.

Varje *freqMatrix*-element räknar antalet förekomster av motsvarande element i *Matrix1*.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

|                                         |               |
|-----------------------------------------|---------------|
| $\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$  | 0.26          |
| $\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$ | $\frac{5}{3}$ |

I vektorformatet Rectangular:

|                                                                                                                                        |                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| $\text{mean}\left(\begin{bmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{bmatrix}\right)$                                                | $\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$         |
| $\text{mean}\left(\begin{bmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & -\frac{1}{2} \end{bmatrix}\right)$                        | $\begin{bmatrix} -\frac{2}{15} & \frac{5}{6} \end{bmatrix}$  |
| $\text{mean}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \begin{bmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{bmatrix}\right)$ | $\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$ |

## median()

**median(List[,freqList])**⇒*uttryck*

Ger medianen för elementen i *List*.

Varje *freqList*-element räknar antalet förekomster av motsvarande element i *List*.

**median(Matrix1[,freqMatrix])**⇒*matrix*

Ger en radvektor som innehåller medianerna för kolumnerna i *Matrix1*.

|                                          |     |
|------------------------------------------|-----|
| $\text{median}(\{0.2, 0.1, -0.3, 0.4\})$ | 0.2 |
|------------------------------------------|-----|

|                                                                                             |                                            |
|---------------------------------------------------------------------------------------------|--------------------------------------------|
| $\text{median}\left(\begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix}\right)$ | $\begin{bmatrix} 0.4 & -0.3 \end{bmatrix}$ |
|---------------------------------------------------------------------------------------------|--------------------------------------------|

Varje *freqMatrix*-element räknar antalet förekomster av motsvarande element i *Matrix1*.

**Obs:**

- Alla inmatningar i listan eller matrisen måste förenklas till tal.
- Tomma element i listan eller matrisen ignoreras. För mer information om tomma element, se på sidan 261.

## MedMed

**MedMed** *X,Y[, Freq] [, Category, Include]*

Beräknar median-median-linjen  $y = (m \cdot x + b)$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                      |
|------------------|--------------------------------------------------|
| stat.RegEqn      | Ekvation för median-median-linje $m \cdot x + b$ |
| stat.m, stat.b   | Modellkoefficienter                              |

| Resultatvariabel | Beskrivning                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.Resid       | Residualer från median-median-linjen                                                                                                                                        |
| stat.XReg        | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg        | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg     | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## mid()

Katalog > 

**mid(sourceString, Start[, Count])**⇒sträng

Ger *Count*-tecknen från teckensträngen *sourceString* och börjar med teckennumret *Start*.

Ger, om *Count* utelämnas eller är större än dimensionen på *sourceString*, alla tecken från *sourceString* med start från teckennumret *Start*.

*Count* måste vara  $\geq 0$ . Om *Count* = 0 erhålls en tom sträng.

**mid(sourceList, Start [, Count])**⇒lista

Ger *Count*-elementen från *sourceList* och börjar med elementnumret *Start*.

Ger, om *Count* utelämnas eller är större än dimensionen på *sourceList*, alla element från *sourceList* med start från elementnumret *Start*.

*Count* måste vara  $\geq 0$ . Om *Count* = 0 erhålls en tom lista.

**mid(sourceStringList, Start[, Count])**⇒lista

Ger *Count*-strängarna från stränglistan *sourceStringList* och börjar med elementnumret *Start*.

|                        |               |
|------------------------|---------------|
| mid{"Hello there",2}   | "ello there " |
| mid{"Hello there",7,3} | "the "        |
| mid{"Hello there",1,5} | "Hello "      |
| mid{"Hello there",1,0} | "{"}          |

|                    |       |
|--------------------|-------|
| mid{{9,8,7,6},3}   | {7,6} |
| mid{{9,8,7,6},2,2} | {8,7} |
| mid{{9,8,7,6},1,2} | {9,8} |
| mid{{9,8,7,6},1,0} | {"{"} |

|                           |              |
|---------------------------|--------------|
| mid{"A","B","C","D"},2,2} | { "B", "C" } |
|---------------------------|--------------|

**min()**Katalog > **min**(*Expr1*, *Expr2*) $\Rightarrow$ *uttryck* $\min(2.3, 1.4)$  1.4**min**(*List1*, *List2*) $\Rightarrow$ *lista* $\min(\{1, 2\}, \{-4, 3\})$   $\{-4, 2\}$ **min**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Ger de två argumentens minimum. Ger, om argumenten är två listor eller matriser, en lista eller matris som innehåller minimumvärdet för varje par av motsvarande element.

**min**(*List*) $\Rightarrow$ *uttryck* $\min(\{0, 1, -7, 1.3, 0.5\})$  -7Ger minimelementet för *List*.**min**(*Matrix1*) $\Rightarrow$ *matrix* $\min\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$   $\begin{bmatrix} -4 & -3 & 0.3 \end{bmatrix}$ 

Ger en radvektor som innehåller minimelementet för varje kolumn i *Matrix1*.

**Obs:** Se även **fMin()** och **max()**.**mirr()**Katalog > **mirr****(financeRate, reinvestRate, CF0, CFList [CFFreq])** $list1 := \{6000, -8000, 2000, -3000\}$   
 $\{6000, -8000, 2000, -3000\}$ 

Finansiell funktion som beräknar den modifierade internräntan på en investering.

 $list2 := \{2, 2, 2, 1\}$   $\{2, 2, 2, 1\}$  $\text{mirr}(4.65, 12, 5000, list1, list2)$  13.41608607

*financeRate* är den räntesats som du betalar på kassaflödesbeloppen.

*reinvestRate* är den räntesats vid vilken kassaflödena återinvesteras.

*CF0* är det initiala kassaflödet vid tidpunkt 0 och måste vara ett reellt tal.

*CFList* är en lista på kassaflödesbelopp efter det initiala kassaflödet CF0.

*CFFreq* är en frivillig lista i vilken varje element specificerar frekvensen för ett grupperat (konsekutivt) kassaflödesbelopp, vilket är det motsvarande elementet i *CFList*. Förinställningen är 1. Om du vill mata in värden måste de vara positiva heltal < 10.000.

**Obs:** Se även **irr()**, på sidan 97.

**mod**(*Expr1*, *Expr2*) $\Rightarrow$ *uttryck*

|                   |   |
|-------------------|---|
| $\text{mod}(7,0)$ | 7 |
|-------------------|---|

**mod**(*List1*, *List2*) $\Rightarrow$ *lista*

|                   |   |
|-------------------|---|
| $\text{mod}(7,3)$ | 1 |
|-------------------|---|

**mod**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

|                    |   |
|--------------------|---|
| $\text{mod}(-7,3)$ | 2 |
|--------------------|---|

Ger det första argumentet modulo det andra argumentet definierat av identiteterna:

|                    |    |
|--------------------|----|
| $\text{mod}(7,-3)$ | -2 |
|--------------------|----|

|                     |    |
|---------------------|----|
| $\text{mod}(-7,-3)$ | -1 |
|---------------------|----|

|                                        |              |
|----------------------------------------|--------------|
| $\text{mod}(\{12,-14,16\},\{9,7,-5\})$ | $\{3,0,-4\}$ |
|----------------------------------------|--------------|

$\text{mod}(x,0) = x$

$\text{mod}(x,y) = x - y \text{ floor}(x/y)$

När det andra argumentet är skilt från noll är resultatet periodiskt i det argumentet. Resultatet är antingen noll eller har samma tecken som det andra argumentet.

Ger, om argumenten är två listor eller matriser, en lista eller matris som innehåller modulen för varje par av motsvarande element.

**Obs:** Se även **remain()**, på sidan 155

**mRow**(*Expr*, *Matrix1*, *Index*) $\Rightarrow$ *matrix*

Ger en kopia av *Matrix1* med varje element i rad *Index* i *Matrix1* multiplicerat med *Expr*.

|                                                                                         |                                                            |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------|
| $\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right)$ | $\begin{bmatrix} 1 & 2 \\ -1 & \frac{-4}{3} \end{bmatrix}$ |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------|

**mRowAdd()**Katalog > 

**mRowAdd**(*Expr*, *Matrix1*, *Index1*, *Index2*)  $\Rightarrow$  *matrix*

Ger en kopia av *Matrix1* med varje element i rad *Index2* i *Matrix1* ersatt med:

$$Expr \cdot \text{rad } Index1 + \text{rad } Index2$$

$$\begin{array}{l} \text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2 \right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} \\ \text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2 \right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix} \end{array}$$

**MultReg**Katalog > 

**MultReg** *Y*, *X1* [, *X2* [, *X3*, ..., [, *X10*]]]

Beräknar den multipla linjära regressionen i lista *Y* på listorna *X1*, *X2*, ..., *X10*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor måste ha samma dimensioner.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel      | Beskrivning                                                        |
|-----------------------|--------------------------------------------------------------------|
| stat.RegEqn           | Regressionsekvation: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$ |
| stat.b0, stat.b1, ... | Regressionskoefficienter                                           |
| stat.R <sup>2</sup>   | Koefficient för multipel bestämning                                |
| stat.ŷList            | $\hat{y}List = b_0 + b_1 \cdot x_1 + \dots$                        |
| stat.Resid            | Residualer från regressionsanalysen                                |

**MultRegIntervals**Katalog > 

**MultRegIntervals** *Y*, *X1* [, *X2* [, *X3*, ..., [, *X10*]]], *XValList* [, *CLevel*]

Beräknar ett prognostiserat *y*-värde, ett nivå-*C*-prediktionsintervall för en enstaka observation och ett nivå-*C*-konfidensintervall för medelvärdet på svaret.

En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor måste ha samma dimensioner.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel                   | Beskrivning                                                                       |
|------------------------------------|-----------------------------------------------------------------------------------|
| stat.RegEqn                        | Regressionsekvation: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$                |
| stat. $\hat{y}$                    | En punktuppskattning: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ för <i>XValList</i> |
| stat.dfError                       | Fel hos frihetsgrader                                                             |
| stat.CLower, stat.CUpper           | Konfidensintervall för ett medelvärde på svaret                                   |
| stat.ME                            | Konfidensintervall - felmarginal                                                  |
| stat.SE                            | Standardfel hos medelvärdet på svaret                                             |
| stat.LowerPred,<br>stat.UpperrPred | Prediktionsintervall för en enstaka observation                                   |
| stat.MEPred                        | Prediktionsintervall - felmarginal                                                |
| stat.SEPred                        | Standardfel för prognostisering                                                   |
| stat.bList                         | Lista på regressionskoefficienter, $\{b_0, b_1, b_3, \dots\}$                     |
| stat.Resid                         | Residualer från regressionen                                                      |

### MultRegTests *Y, X1[,X2[,X3,...[,X10]]]*

Ett multipelt linjärt regressionstest utför en multipel linjär regressionsanalys på givna data och ger den globala  $F$ -teststatistiken och  $t$ -teststatistiken för koefficienterna.

En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

Utdata

| Resultatvariabel | Beskrivning                                                        |
|------------------|--------------------------------------------------------------------|
| stat.RegEqn      | Regressionsekvation: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$ |
| stat.F           | Global $F$ -teststatistik                                          |

| Resultatvariabel       | Beskrivning                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------|
| stat.PVal              | P-värde associerat med global $F$ -statistik                                                                 |
| stat.R <sup>2</sup>    | Koefficient för multipel bestämning                                                                          |
| stat.AdjR <sup>2</sup> | Justerad koefficient för multipel bestämning                                                                 |
| stat.s                 | Standardavvikelse hos felet                                                                                  |
| stat.DW                | Durbin-Watson-statistik: används för att bestämma om modellen innehåller autokorrelation av första ordningen |
| stat.dfReg             | Frihetsgrader hos regressionen                                                                               |
| stat.SSReg             | Regressionens kvadratsumma                                                                                   |
| stat.MSReg             | Regression medelkvadrat                                                                                      |
| stat.dfError           | Fel hos frihetsgrader                                                                                        |
| stat.SSError           | Felens kvadratsumma                                                                                          |
| stat.MSError           | Felens medelkvadrat                                                                                          |
| stat.bList             | {b <sub>0</sub> ,b <sub>1</sub> ,...} Lista på koefficienter                                                 |
| stat.tList             | Lista på t-statistik för varje koefficient i bList                                                           |
| stat.PList             | Lista på P-värden för varje t-statistik                                                                      |
| stat.SEList            | Lista på standardfel för koefficienter i bList                                                               |
| stat.ŷList             | $\hat{y}List = b_0 + b_1 \cdot x_1 + \dots$                                                                  |
| stat.Resid             | Residualer från regressionen                                                                                 |
| stat.sResid            | Standardiserade residualer: erhållna genom att dividera en residual med dess standardavvikelse               |
| stat.CookDist          | Cooks avstånd: mått på den influens som en observation har, baserat på residual och stigning                 |
| stat.Leverage          | Mått på hur långt värdena i den oberoende variabeln är från sina medelvärden                                 |

## N

| nand                                               | ctrl                       | = | knappar    |
|----------------------------------------------------|----------------------------|---|------------|
| <i>BoolesktUttr1</i> nand <i>BoolesktUttr2</i> ger | $x \geq 3$ and $x \geq 4$  |   | $x \geq 4$ |
| <i>Booleskt uttryck</i>                            | $x \geq 3$ nand $x \geq 4$ |   | $x < 4$    |
| <i>BooleskLista1</i> nand <i>BooleskLista2</i> ger |                            |   |            |
| <i>Boolesk lista</i>                               |                            |   |            |



*BooleskMatris1* **nand** *BooleskMatris2*  
ger *Boolesk matris*

Ger negation av en logisk **and** uppgift på de två argumenten. Ger resultatet sant, falskt eller en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

*Heltal1* **nand** *Heltal2*  $\Rightarrow$  *heltal*

Jämför två reella heltal bit för bit med en **nand**-operation. Internt omvandlas båda heltalen till 64-bitars binära tal. När motsvarande bitar jämförs är resultatet 0 om båda bitarna är 1; annars är resultatet 1. Det returnerade värdet representerar bitresultaten och visas enligt basläget.

Du kan skriva in heltalen i valfri talbas. För en binär eller hexadecimal inmatning måste du använda prefixet 0b respektive 0h. Utan prefix behandlas heltalen som decimala (bas 10).

|                              |                |
|------------------------------|----------------|
| 3 and 4                      | 0              |
| 3 nand 4                     | -1             |
| $\{1,2,3\}$ and $\{3,2,1\}$  | $\{1,2,1\}$    |
| $\{1,2,3\}$ nand $\{3,2,1\}$ | $\{-2,-3,-2\}$ |

## nCr()

Katalog > 

**nCr**(*Expr1*, *Expr2*)  $\Rightarrow$  *uttryck*

För heltal *Expr1* och *Expr2* med *Expr1*  $\geq$  *Expr2*  $\geq 0$  är **nCr()** antalet kombinationer av *Expr1* saker tagna *Expr2* åt gången. (Detta kallas också en binomial koefficient.) Båda argumenten kan vara heltal eller symboliska uttryck.

|                        |                                       |
|------------------------|---------------------------------------|
| $nCr(z,3)$             | $\frac{z \cdot (z-2) \cdot (z-1)}{6}$ |
| $Ans z=5$              | 10                                    |
| $nCr(z,c)$             | $\frac{z!}{c! \cdot (z-c)!}$          |
| $\frac{Ans}{nPr(z,c)}$ | $\frac{1}{c!}$                        |

**nCr**(*Expr*, 0)  $\Rightarrow$  1

**nCr**(*Expr*, *negInteger*)  $\Rightarrow$  0

**nCr**(*Expr*, *posInteger*)  $\Rightarrow$  *Expr* ·  
(*Expr*-1) ... (*Expr*-*posInteger*+1)/  
*posInteger*!

**nCr**(*Expr*, *nonInteger*)  $\Rightarrow$  *expression*!/

**nCr()**Katalog > **((Expr-nonInteger)!·nonInteger!)****nCr(List1, List2)⇒lista**

|                             |              |
|-----------------------------|--------------|
| $nCr(\{5,4,3\}, \{2,4,2\})$ | $\{10,1,3\}$ |
|-----------------------------|--------------|

Ger en lista på kombinationer baserat på motsvarande elementpar i de två listorna. Argumenten måste ha samma liststorlek.

**nCr(Matrix1, Matrix2)⇒matris**

|                                                                                                              |                                                  |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| $nCr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$ | $\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$ |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------|

Ger en matris över kombinationer baserat på motsvarande elementpar i de två matriserna. Argumenten måste ha samma matrisstorlek.

**nDerivative()**Katalog > **nDerivative(Utr1,Var=Värde [,Ordning])⇒värde**

|                        |   |
|------------------------|---|
| $nDerivative( x ,x=1)$ | 1 |
|------------------------|---|

|                             |       |
|-----------------------------|-------|
| $nDerivative( x ,x) _{x=0}$ | undef |
|-----------------------------|-------|

**nDerivative(Utr1,Var[,Ordning]) | Var=Värde⇒värde**

|                                    |       |
|------------------------------------|-------|
| $nDerivative(\sqrt{x-1},x) _{x=1}$ | undef |
|------------------------------------|-------|

Ger den numeriska derivatan beräknad med automatiska deriveringsmetoder.

När *Värde* specificeras överstyr detta värde eventuella tidigare variabeltilldelningar eller aktuella ersättningar av typ "|" för variabeln.

*Ordning* för derivatan måste vara **1** eller **2**.

**newList()**Katalog > **newList(numElements)⇒lista**

|              |               |
|--------------|---------------|
| $newList(4)$ | $\{0,0,0,0\}$ |
|--------------|---------------|

Ger en lista med dimensionen på *numElements*. Varje element är noll.

**newMat()**Katalog > **newMat(numRows, numColumns)⇒matris**

|               |                                                        |
|---------------|--------------------------------------------------------|
| $newMat(2,3)$ | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |
|---------------|--------------------------------------------------------|

Ger en matris med nollor med dimensionen *numRows* gånger *numColumns*.

## nfMax()

**nfMax**(*Expr*, *Var*) $\Rightarrow$ värde

**nfMax**(*Expr*, *Var*, *undrGräns*) $\Rightarrow$ värde

**nfMax**(*Expr*, *Var*, *undrGräns*,  
*övrGräns*) $\Rightarrow$ värde

**nfMax**(*Expr*, *Var*) | *undrGräns* $\leq$ *Var*  
 $\leq$ *övrGräns* $\Rightarrow$ värde

Ger ett möjligt numeriskt värde på variabeln *Var* där lokalt maximum för *Expr* inträffar.

Om du inför *undrGräns* och *övrGräns* söker funktionen i det stängda intervallet [*undrGräns*,*övrGräns*] efter lokalt maximum.

**Obs:** Se även **fMax()** och **d()**.

|                                                 |     |
|-------------------------------------------------|-----|
| $\text{nfMax}(x^2 - 2 \cdot x - 1, x)$          | -1. |
| $\text{nfMax}(0.5 \cdot x^3 - x - 2, x, -5, 5)$ | 5.  |

## nfMin()

**nfMin**(*Expr*, *Var*) $\Rightarrow$ värde

**nfMin**(*Expr*, *Var*, *undrGräns*) $\Rightarrow$ värde

**nfMin**(*Expr*, *Var*, *undrGräns*,  
*övrGräns*) $\Rightarrow$ värde

**nfMin**(*Expr*, *Var*) | *undrGräns* $\leq$ *Var*  
 $\leq$ *övrGräns* $\Rightarrow$ värde

Ger ett möjligt numeriskt värde på variabeln *Var* där lokalt minimum för *Expr* inträffar.

Om du inför *undrGräns* och *övrGräns* söker funktionen i det stängda intervallet [*undrGräns*,*övrGräns*] efter lokalt minimum.

**Obs:** Se även **fMin()** och **d()**.

|                                                 |     |
|-------------------------------------------------|-----|
| $\text{nfMin}(x^2 + 2 \cdot x + 5, x)$          | -1. |
| $\text{nfMin}(0.5 \cdot x^3 - x - 2, x, -5, 5)$ | -5. |

**nInt()**Katalog > 

**nInt**(*Expr1*, *Var*, *Lower*,  
*Upper*) $\Rightarrow$ uttryck

$$\text{nInt}\left(e^{-x^2}, x, -1, 1\right) \quad 1.49365$$

Om integranden *Expr1* inte innehåller någon variabel utöver *Var*, och om *Lower* och *Upper* är konstanter, positiv  $\infty$  eller negativ  $\infty$ , ger **nInt()** en uppskattning av  $\int(\text{Expr1}, \text{Var}, \text{Lower}, \text{Upper})$ . Denna uppskattning är ett vägt genomsnitt av vissa sampelvärden hos integranden i intervallet  $\text{Lower} < \text{Var} < \text{Upper}$ .

Målsättningen är sex signifikanta siffror. Den adaptiva algoritmen bestämmer när det verkar sannolikt att målet har uppnåtts, eller när det verkar osannolikt att ytterligare sampling ger en nämnvärd förbättring.

En varning ("Questionable accuracy") visas när det verkar som om målet inte har uppnåtts.

Man kan kapsla in **nInt()** för att utföra multipel numerisk integrering. Integrationsgränser kan bero på integrationsvariabler utanför gränserna.

**Obs:** Se även **j()**, på sidan 231.

$$\text{nInt}\left(\cos(x), x, -\pi, \pi + 1.E-12\right) \quad -1.04144E-12$$

$$\int_{-\pi}^{\pi+10^{-12}} \cos(x) dx \quad -\sin\left(\frac{1}{1000000000000}\right)$$

$$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$$

**nom()**Katalog > 

**nom**(*effectiveRate*, *CpY*) $\Rightarrow$ värde

$$\text{nom}(5.90398, 12) \quad 5.75$$

Finansiell funktion som konverterar den årliga effektiva räntan *effectiveRate* till en nominell ränta, given av *CpY* som antalet sammansatta ränteperioder per år.

*effectiveRate* måste vara ett reellt tal och *CpY* måste vara ett reellt tal  $> 0$ .

**Obs:** Se även **eff()**, på sidan 60.

*BoolesktUtrr1* **nor** *BoolesktUtrr2* ger  
*Booleskt uttryck*

|                           |            |
|---------------------------|------------|
| $x \geq 3$ or $x \geq 4$  | $x \geq 3$ |
| $x \geq 3$ nor $x \geq 4$ | $x < 3$    |

*BooleskLista1* **nor** *BooleskLista2* ger  
*Boolesk lista*

*BooleskMatris1* **nor** *BooleskMatris2* ger  
*Boolesk matris*

Ger negation av en logisk **or** operation på de två argumenten. Ger resultatet sant, falskt eller en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

*Heltal1* **nor** *Heltal2*  $\Rightarrow$  *heltal*

|                                 |                  |
|---------------------------------|------------------|
| 3 or 4                          | 7                |
| 3 nor 4                         | -8               |
| $\{1, 2, 3\}$ or $\{3, 2, 1\}$  | $\{3, 2, 3\}$    |
| $\{1, 2, 3\}$ nor $\{3, 2, 1\}$ | $\{-4, -3, -4\}$ |

Jämför två reella heltal bit för bit med en **nor**-operation Internt omvandlas båda heltalen till 64-bitars binära tal. När motsvarande bitar jämförs blir resultatet 1 om båda bitarna är 1, annars blir resultatet 0. Det erhållna värdet representerar bitresultatet och visas enligt Bas-läget.

Du kan skriva in heltalen i valfri talbas. För en binär eller hexadecimal inmatning måste du använda prefixet 0b respektive 0h. Utan prefix behandlas heltalen som decimala (bas 10).

## norm()

**norm**(*Matrix*)  $\Rightarrow$  *uttryck*

$$\text{norm} \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) \quad \sqrt{a^2 + b^2 + c^2 + d^2}$$

**norm**(*Vector*)  $\Rightarrow$  *uttryck*

$$\text{norm} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right) \quad \sqrt{30}$$

Ger Frobenius norm.

$$\text{norm} \left( \begin{bmatrix} 1 & 2 \end{bmatrix} \right) \quad \sqrt{5}$$

$$\text{norm} \left( \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) \quad \sqrt{5}$$

**normalLine()**Katalog > **normalLine**(*Expr1*,*Var*,*Point*) $\Rightarrow$ uttryck**normalLine**(*Expr1*,*Var=Point*) $\Rightarrow$ uttryck

Ger normalen för kurvan representerad av *Expr1* vid punkten som specificeras i *Var=Point*.

Kontrollera att den oberoende variabeln inte är definierad. Om exempelvis *f1* (*x*):=5 och *x*:=3 ger **normalLine**(*f1*(*x*),*x*,2) "falskt".

|                                                    |                             |
|----------------------------------------------------|-----------------------------|
| $\text{normalLine}(x^2, x, 1)$                     | $\frac{3}{2} - \frac{x}{2}$ |
| $\text{normalLine}((x-3)^2 - 4, x, 3)$             | $x = 3$                     |
| $\text{normalLine}\left(\frac{1}{x^3}, x=0\right)$ | 0                           |
| $\text{normalLine}(\sqrt{ x }, x=0)$               | undef                       |

**normCdf()**Katalog > 

**normCdf**(*lowBound*,*upBound*[,*μ*[,*σ*]]) $\Rightarrow$ tal  
om *lowBound* och *upBound* är tal, *lista* om  
*lowBound* och *upBound* är listor

Beräknar sannolikheten vid en normalfördelning mellan *lowBound* och *upBound* för specificerad *μ* (förinställning=0) och *σ* (förinställning=1).

För  $P(X \leq \text{upBound})$ , sätt *lowBound* =  $-\infty$ .

**normPdf()**Katalog > 

**normPdf**(*XVal*[,*μ*[,*σ*]]) $\Rightarrow$ tal om *XVal* är ett tal, *lista* om *XVal* är en lista

Beräknar värde hos täthetsfunktionen för normalfördelning vid ett specificerat *XVal*-värde för specificerad *μ* och *σ*.

**not (icke)**Katalog > **not** *BooleanExpr* $\Rightarrow$ Booleskt uttryck

Ger en sann, falsk eller förenklad form av argumentet.

**not** *Integer1* $\Rightarrow$ heltal

|                           |                 |
|---------------------------|-----------------|
| $\text{not}(2 \geq 3)$    | true            |
| $\text{not}(x < 2)$       | $x \geq 2$      |
| $\text{not not innocent}$ | <i>innocent</i> |

I hexadecimalt basläge:

**Viktigt:** Noll, inte bokstaven O.

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| $\text{not } 0\text{h}7\text{AC}36$ | $0\text{hFFFFFFFFFFFF}853\text{C9}$ |
|-------------------------------------|-------------------------------------|

Ger ettkomplementet till ett reellt heltal. Internt omvandlas *Integer1* till ett 64-bitars binärt tal. Värdet på varje bit växlas (0 blir 1 och vice versa) för ettans komplement. Resultaten visas enligt det inställda basläget.

Du kan skriva in heltalet i valfri talbas. För en binär eller hexadecimal inmatning måste du använda prefixet 0b respektive 0h. Utan prefix behandlas heltalet som ett decimalt tal (bas 10).

Om du skriver in ett decimalt heltal som är alltför stort för att anges i 64-bitars binär form används en symmetrisk modulooperation för att få ned värdet till lämplig nivå. För mer information, se ►Base2, på sidan 18.

I binärt basläge:

|                                      |     |
|--------------------------------------|-----|
| 0b100101 ►Base10                     | 37  |
| not 0b100101                         |     |
| 0b11111111111111111111111111111111 ► |     |
| not 0b100101 ►Base10                 | -38 |

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

**Obs:** En binär inmatning kan ha upp till 64 siffror (exklusive prefixet 0b). En hexadecimal inmatning kan ha upp till 16 siffror.

## nPr()

$nPr(Expr1, Expr2) \Rightarrow$  uttryck

För heltal *Expr1* och *Expr2* med  $Expr1 \geq Expr2 \geq 0$  är **nPr()** antalet permutationer av *Expr1* saker tagna *Expr2* åt gången. Båda argumenten kan vara heltal eller symboliska uttryck.

$nPr(Expr, 0) \Rightarrow 1$

$nPr(Expr, negInteger) \Rightarrow 1 / ((Expr+1) \cdot (Expr+2) \dots (expression - negInteger))$

$nPr(Expr, posInteger) \Rightarrow Expr \cdot (Expr-1) \dots (Expr-posInteger+1)$

$nPr(Expr, nonInteger) \Rightarrow Expr! / (Expr-nonInteger)!$

$nPr(List1, List2) \Rightarrow$  lista

Ger en lista på permutationer baserat på motsvarande elementpar i de två listorna. Argumenten måste ha samma liststorlek.

$nPr(Matrix1, Matrix2) \Rightarrow$  matrix

|                          |                                           |
|--------------------------|-------------------------------------------|
| $nPr(z, 3)$              | $z \cdot (z-2) \cdot (z-1)$               |
| $Ans z=5$                | 60                                        |
| $nPr(z, -3)$             | $\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$ |
| $nPr(z, c)$              | $\frac{z!}{(z-c)!}$                       |
| $Ans \cdot nPr(z-c, -c)$ | 1                                         |

|                                 |                 |
|---------------------------------|-----------------|
| $nPr(\{5, 4, 3\}, \{2, 4, 2\})$ | $\{20, 24, 6\}$ |
|---------------------------------|-----------------|

|                                                                                                              |                                                   |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| $nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$ | $\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$ |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------|

Ger en matris över permutationer baserat på motsvarande elementpar i de två matriserna. Argumenten måste ha samma matrisstorlek.

## npv()

**npv**(*InterestRate*,*CFO*,*CFL**ist*  
[,*CFF**req*])

Finansiell funktion som beräknar nettovärdet, dvs. summan av aktuella värden för kassainflöden och kassautflöden. Ett positivt resultat för npv indikerar en vinstgivande investering.

*InterestRate* är räntan med vilken kassaflödena (kapitalanskaffningskostnaderna) diskonteras under en period.

*CFO* är det initiala kassaflödet vid tidpunkt 0 och måste vara ett reellt tal.

*CFL**ist* är en lista på kassaflödesbelopp efter det initiala kassaflödet *CFO*.

*CFF**req* är en lista i vilken varje element specificerar frekvensen för ett grupperat (konsekutivt) kassaflödesbelopp, vilket är det motsvarande elementet i *CFL**ist*. Förinställningen är 1. Om du vill mata in värden måste de vara positiva heltal < 10.000.

|                                         |                                |
|-----------------------------------------|--------------------------------|
| $list1 := \{6000, -8000, 2000, -3000\}$ | $\{6000, -8000, 2000, -3000\}$ |
| $list2 := \{2, 2, 2, 1\}$               | $\{2, 2, 2, 1\}$               |
| $npv(10, 5000, list1, list2)$           | 4769.91                        |

## nSolve()

**nSolve**(*Equation*,*Var*[=*Guess*]) $\Rightarrow$ *tal*  
eller *fel\_sträng*

**nSolve**(*Equation*,*Var*  
[=*Guess*],*lowBound*) $\Rightarrow$ *tal* eller *fel\_*  
*sträng*

**nSolve**(*Equation*,*Var*  
[=*Guess*],*lowBound*,*upBound*) $\Rightarrow$ *tal*  
eller *fel\_sträng*

|                                       |         |
|---------------------------------------|---------|
| $nSolve(x^2 + 5 \cdot x - 25 = 9, x)$ | 3.84429 |
| $nSolve(x^2 = 4, x = -1)$             | -2.     |
| $nSolve(x^2 = 4, x = 1)$              | 2.      |

**Obs:** Om det finns flera lösningar kan du använda en gissning för att lättare hitta en viss lösning.



**nSolve**(*Equation*, *Var*[=*Guess*]) |  
*lowBound* ≤ *Var* ≤ *upBound* ⇒ *tal* eller  
*fel\_sträng*

Söker iterativt efter en ungefärlig reell numerisk lösning på *Equation* för dess variabel. Specificera variabeln som:

*variabel*

– eller –

*variabel* = *reellt tal*

Som exempel är  $x$  giltigt och likaså  $x=3$ .

**nSolve()** är ofta mycket snabbare än **solve()** eller **zeros()**, särskilt om operatoren "|" används för att begränsa sökningen till ett litet intervall som innehåller exakt en enkel lösning.

**nSolve()** försöker att bestämma antingen en punkt där residualen är noll eller två relativt närliggande punkter där residualen har motsatta tecken och inte är överdrivet stor. Om detta inte kan uppnås med ett måttligt antal samplingspunkter erhålls strängen "no solution found".

**Obs:** Se även **cSolve()**, **cZeros()**, **solve()** och **zeros()**.

|                                                                                          |                     |
|------------------------------------------------------------------------------------------|---------------------|
| $\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x)   x < 0$                                     | -8.84429            |
| $\text{nSolve}\left(\frac{(1+r)^{24}-1}{r} = 26, r\right)   r > 0 \text{ and } r < 0.25$ | 0.006886            |
| $\text{nSolve}(x^2 = 1, x)$                                                              | "No solution found" |

## O

### OneVar

**OneVar** [*1*], *X*[, *Freq*][, *Category*, *Include*]]

**OneVar** [*n*], *X1*, *X2*[*X3*[, ..., *X20*]]]

Beräknar 1-variabelstatistik på upp till 20 listor. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X*-argumenten är datalistor.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*-värde. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

Ett tomt element i någon av listorna *X*, *Freq* eller *Category* resulterar i ett tomrum för motsvarande element i dessa listor. Ett tomt element i någon av listorna *X1* till och med *X20* resulterar i ett tomrum för motsvarande element i dessa listor. För mer information om tomma element, se på sidan 261.

| Resultatvariabel             | Beskrivning                                             |
|------------------------------|---------------------------------------------------------|
| stat. $\bar{x}$              | Medelvärde av <i>x</i> -värden                          |
| stat. $\Sigma x$             | Summa av <i>x</i> -värden                               |
| stat. $\Sigma x^2$           | Summa av <i>x</i> <sup>2</sup> -värden                  |
| stat. <i>sx</i>              | Standardavvikelse för <i>x</i> (sampling)               |
| stat. <i>x</i>               | Standardavvikelse för <i>x</i> (population)             |
| stat. <i>n</i>               | Antal datapunkter                                       |
| stat.Min <i>X</i>            | Minsta <i>x</i> -värde                                  |
| stat.Q <sub>1</sub> <i>X</i> | Undre kvartil för <i>x</i>                              |
| stat.Median <i>X</i>         | Median för <i>x</i>                                     |
| stat.Q <sub>3</sub> <i>X</i> | Övre kvartil för <i>x</i>                               |
| stat.Max <i>X</i>            | Största <i>x</i> -värde                                 |
| stat.SSX                     | Kvadratsumma av avvikelser från medelvärdet på <i>x</i> |

*BoolesktUtr1* or *BoolesktUtr2* ger

$x \geq 3$  or  $x \geq 4$

$x \geq 3$

*Booleskt uttryck*

*BooleskLista1* **or** *BooleskLista2* ger  
*Boolesk lista*

*BooleskMatris1* **or** *BooleskMatris2* ger  
*Boolesk matris*

Ger resultatet sant eller falskt eller en  
förenklad form av den ursprungliga  
inmatningen.

Ger sant om ettdera eller båda uttrycken  
förenklas till sant. Ger resultatet falskt  
om båda uttrycken utvärderas som  
falska.

**Obs:** Se **xor**.

**Obs för att mata in exemplet:** Se avsnittet  
Räknare i produkthandboken för  
instruktioner om hur du anger multiline-  
program och funktionsdefinitioner.

*Integer1* **or** *Integer2*  $\Rightarrow$  *heltal*

Jämför två reella heltal bit för bit med en  
or-operation. Internt omvandlas båda  
heltalen till 64-bitars binära tal. När  
motsvarande bitar jämförs blir resultatet  
1 om båda bitarna är 1. Resultatet blir 0  
endast om båda bitarna är 0. Det  
erhållna värdet representerar  
bitresultaten och visas enligt det  
inställda basläget.

Du kan skriva in heltalen i valfri talbas.  
För en binär eller hexadecimal inmatning  
måste du använda prefixet 0b respektive  
0h. Utan prefix behandlas heltalen som  
decimala (bas 10).

Om du skriver in ett decimalt heltal som  
är alltför stort för att anges i 64-bitars  
binär form används en symmetrisk  
modulooperation för att få ned värdet  
till lämplig nivå. För mer information, se  
►**Base2**, på sidan 18.

**Obs:** Se **xor**.

|                             |                                   |
|-----------------------------|-----------------------------------|
| Define $g(x)$ = Func        | Done                              |
| If $x \leq 0$ or $x \geq 5$ |                                   |
| Goto end                    |                                   |
| Return $x \cdot 3$          |                                   |
| Lbl end                     |                                   |
| EndFunc                     |                                   |
| $g(3)$                      | 9                                 |
| $g(0)$                      | A function did not return a value |

I hexadecimalt basläge:

|                   |         |
|-------------------|---------|
| 0h7AC36 or 0h3D5F | 0h7BD7F |
|-------------------|---------|

**Viktigt:** Noll, inte bokstaven O.

I binärt basläge:

|                   |          |
|-------------------|----------|
| 0b100101 or 0b100 | 0b100101 |
|-------------------|----------|

**Obs:** En binär inmatning kan ha upp till 64  
siffror (exklusive prefixet 0b). En hexadecimal  
inmatning kan ha upp till 16 siffror.

**ord()**Katalog > **ord(String)**⇒integer

ord("hello") 104

**ord(List l)**⇒lista

char(104) "h"

ord(char(24)) 24

Ger den numeriska koden för det första tecknet i teckensträngen *String* eller en lista på de första tecknen i varje listelement.

ord({"alpha", "beta"}) {97,98}

**P****P►Rx()**Katalog > **P►Rx(rExpr, θExpr)**⇒uttryck

I vinkelläget Radianer:

**P►Rx(rList, θList)**⇒lista $P►Rx(r, \theta) \cos(\theta) \cdot r$ **P►Rx(rMatrix, θMatrix)**⇒matrix $P►Rx(4, 60^\circ) 2$ 

Ger den ekvivalenta x-koordinaten för paret (r, θ).

$$P►Rx\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right\}$$

$$\left\{\frac{-3}{2}, 5\sqrt{2}, 1.3\right\}$$

**Obs:** Argumentet θ tolkas som en vinkel i antingen grader, nygrader eller i radianer beroende på det aktuella vinkelläget. Om argumentet är ett uttryck kan du använda °, G eller r för att tillfälligt överstyra vinkelläget.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **P@>Rx (...)**.

**P►Ry()**Katalog > **P►Ry(rExpr, θExpr)**⇒uttryck

I vinkelläget Radianer:

**P►Ry(rList, θList)**⇒lista $P►Ry(r, \theta) \sin(\theta) \cdot r$ **P►Ry(rMatrix, θMatrix)**⇒matrix $P►Ry(4, 60^\circ) 2\sqrt{3}$ 

Ger den ekvivalenta y-koordinaten för paret (r, θ).

$$P►Ry\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right\}$$

$$\left\{\frac{-3\sqrt{3}}{2}, -5\sqrt{2}, 0\right\}$$

**Obs:** Argumentet θ tolkas som en vinkel i antingen grader, radianer eller nygrader beroende på det aktuella vinkelläget. Om argumentet är ett uttryck kan du använda °, G eller r för att tillfälligt överstyra vinkelläget.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **P@>Ry (...)**.

## PassErr

### PassErr

För ett exempel på **PassErr**, se exempel 2 under kommandot **Try**, på sidan 199.

Flyttar ett fel till nästa nivå.

Om systemvariabeln *errCode* är noll utför **PassErr** ingenting.

Villkoret **Else** i blocket **Try...Else...EndTry** bör använda **ClrErr** eller **PassErr**. Om felet skall processas eller ignoreras, använd **ClrErr**. Om det är okänt hur felet skall hanteras, använd **PassErr** för att skicka felet vidare till nästa felhanterare. Om det inte finns någon ytterligare felhanterare för **Try...Else...EndTry** visas feldialogrutan som normal.

**Obs:** Se även **ClrErr**, på sidan 26 och **Try**, på sidan 199.

**Kommentarer om inmatningen av exemplet:** applikationen Räkna kan du skriva in flerradiga definitioner genom att trycka på  i stället för  i slutet av varje linje. På datorns tangentbord, håll ned **Alt** och tryck på **Enter**.

## piecewise()

**piecewise**(*Expr1* [, *Condition1* [, *Expr2* [, *Condition2* [, ... ]]]])

Ger definitioner för en stegvis funktion i form av en lista. Du kan också skapa stegvisa definitioner med hjälp av en mall.

**Obs:** Se även **Piecewise template**, på sidan 3.

|                                                                                  |       |
|----------------------------------------------------------------------------------|-------|
| Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$ | Done  |
| $p(1)$                                                                           | 1     |
| $p(-1)$                                                                          | undef |

## poissCdf()

**poissCdf**( $\lambda$ , *lowBound*, *upBound*)  $\Rightarrow$  tal om

**poissCdf()**Katalog > 

*lowBound* och *upBound* är tal, *lista* om  
*lowBound* och *upBound* är listor

**poissCdf( $\lambda$ ,*upBound*)**(för  $P(0 \leq X \leq \textit{upBound}) \Rightarrow \textit{tal}$  om *upBound* är ett tal, lista om *upBound* är en lista

Beräknar en kumulativ sannolikhet för den diskreta Poisson-fördelningen med det specificerade medelvärde  $\lambda$ .

För  $P(X \leq \textit{upBound})$ , sätt *lowBound*=0

**poissPdf()**Katalog > 

**poissPdf( $\lambda$ ,*XVal*)** $\Rightarrow$ *tal* om *XVal* är ett tal, lista om *XVal* är en lista

Beräknar en sannolikhet för den diskreta Poisson-fördelningen med det specificerade medelvärde  $\lambda$ .

**►Polar**Katalog > 

*Vector* ►**Polar**

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>**Polar**.

Visar *vector* i polär form [ $r \angle \theta$ ]. Vektorn måste ha dimensionen 2 och kan vara en rad eller en kolumn.

**Obs:** ►**Polar** är en visa format-instruktion, inte en konverteringsfunktion. Du kan endast använda den i slutet av en inmatningsrad, och den uppdaterar inte *ans*.

**Obs:** Se även ►**Rect**, på sidan 152.

*complexValue* ►**Polar**

Visar *complexVector* i polär form.

- Vinkelläget Grader ger ( $r \angle \theta$ ).
- Vinkelläget Radianer ger  $re^{i\theta}$ .

$$\begin{array}{l} \left[ \begin{array}{cc} 1 & 3 \end{array} \right] \text{►Polar} & \left[ 3.16228 \quad \angle 1.24905 \right] \\ \left[ \begin{array}{cc} x & y \end{array} \right] \text{►Polar} & \left[ \sqrt{x^2+y^2} \quad \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1} \left( \frac{x}{y} \right) \right] \end{array}$$

I vinkelläget Radianer:

$$\begin{array}{l} \left( 3+4 \cdot i \right) \text{►Polar} & e^{i \cdot \left( \frac{\pi}{2} - \tan^{-1} \left( \frac{3}{4} \right) \right)} \cdot 5 \\ \left( \left( 4 \angle \frac{\pi}{3} \right) \right) \text{►Polar} & e^{\frac{i \cdot \pi}{3}} \cdot 4 \end{array}$$

*complexValue* kan ha valfri komplex form. En inmatning av  $re^{i\theta}$  orsakar dock ett fel i vinkelläget Grader.

**Obs:** Du måste använda parenteserna för en  $(r \angle \theta)$  polär inmatning.

I vinkelläget Nygrader:

$$(4 \cdot i) \blacktriangleright \text{Polar} \quad (4 \angle 100.)$$

I vinkelläget Grader:

$$(3+4 \cdot i) \blacktriangleright \text{Polar} \quad \left( 5 \angle 90 - \tan^{-1} \left( \frac{3}{4} \right) \right)$$

## polyCoeffs()

**polyCoeffs(Poly [,Var])**⇒*lista*

Ger en lista på koefficienterna för polynomet *Poly* med avseende på variabeln *Var*.

*Poly* måste vara ett polynomuttryck i *Var*. Vi rekommenderar att du inte utelämnar *Var* såvida inte *Poly* är ett uttryck i bara en variabel.

$$\text{polyCoeffs}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad \{4, -3, 2\}$$

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Expanderar polynomet och väljer *x* för den utelämnade *Var*.

$$\text{polyCoeffs}((x+y+z)^2, x) \quad \{1, 2 \cdot (y+z), (y+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, y) \quad \{1, 2 \cdot (x+z), (x+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, z) \quad \{1, 2 \cdot (x+y), (x+y)^2\}$$

## polyDegree()

**polyDegree(Poly [,Var])**⇒*värde*

Ger graden för polynomuttrycket *Poly* med avseende på variabeln *Var*. Om du utelämnar *Var* väljer funktionen

**polyDegree()** en förinställning från variablerna i polynomet *Poly*.

$$\text{polyDegree}(5) \quad 0$$

$$\text{polyDegree}(\ln(2) + \pi, x) \quad 0$$

Konstanta polynom

**polyDegree()**Katalog > 

*Poly* måste vara ett polynomuttryck i *Var*. Vi rekommenderar att du inte utelämnar *Var* såvida inte *Poly* är ett uttryck i en singular variabel.

|                                                     |       |
|-----------------------------------------------------|-------|
| $\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$ | 2     |
| $\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$          | 5     |
| $\text{polyDegree}((x+y^2+z^3)^2, x)$               | 2     |
| $\text{polyDegree}((x+y^2+z^3)^2, y)$               | 4     |
| $\text{polyDegree}((x-1)^{10000}, x)$               | 10000 |

Graden kan extraheras även om koefficienterna inte kan extrahera den. Detta beror på att graden kan extraheras utan att utveckla polynomet.

**polyEval()**Katalog > 

**polyEval(List1, Expr1) ⇒ uttryck**

**polyEval(List1, List2) ⇒ uttryck**

Tolkar det första argumentet som koefficienten för ett polynom med fallande ordning och ger polynomet utvärderat för det andra argumentets värde.

|                                              |                               |
|----------------------------------------------|-------------------------------|
| $\text{polyEval}(\{a, b, c\}, x)$            | $a \cdot x^2 + b \cdot x + c$ |
| $\text{polyEval}(\{1, 2, 3, 4\}, 2)$         | 26                            |
| $\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$ | $\{26, -262\}$                |

**polyGcd()**Katalog > 

**polyGcd(Expr1, Expr2) ⇒ uttryck**

Ger den största gemensamma delaren för de två argumenten.

*Expr1* och *Expr2* måste vara polynomuttryck.

Listargument, matrisargument och booleska argument är ej tillåtna.

|                                                                           |         |
|---------------------------------------------------------------------------|---------|
| $\text{polyGcd}(100, 30)$                                                 | 10      |
| $\text{polyGcd}(x^2 - 1, x - 1)$                                          | $x - 1$ |
| $\text{polyGcd}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$ | $x - 2$ |



**polyQuotient()**Katalog > **polyQuotient**(*Poly1*,*Poly2*  
[,*Var*]) $\Rightarrow$ uttryck

Ger kvoten på polynomet *Poly1* dividerat med polynomet *Poly2* med avseende på den specificerade variabeln *Var*.

*Poly1* och *Poly2* måste vara polynomuttryck i *Var*. Vi rekommenderar att du inte utelämnar *Var* såvida inte *Poly1* och *Poly2* är uttryck i samma variabel.

|                                                    |                             |
|----------------------------------------------------|-----------------------------|
| $\text{polyQuotient}(x-1, x-3)$                    | 1                           |
| $\text{polyQuotient}(x-1, x^2-1)$                  | 0                           |
| $\text{polyQuotient}(x^2-1, x-1)$                  | $x+1$                       |
| $\text{polyQuotient}(x^3-6x^2+11x-6, x^2-6x+8)$    | $x$                         |
| <hr/>                                              |                             |
| $\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, x)$ | $y-z$                       |
| $\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, y)$ | $2 \cdot x - y + 2 \cdot z$ |
| <hr/>                                              |                             |
| $\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, z)$ | $-(x-y)$                    |

**polyRemainder()**Katalog > **polyRemainder**(*Poly1*,*Poly2*  
[,*Var*]) $\Rightarrow$ uttryck

Ger resten på polynomet *Poly1* dividerat med polynomet *Poly2* med avseende på den specificerade variabeln *Var*.

*Poly1* och *Poly2* måste vara polynomuttryck i *Var*. Vi rekommenderar att du inte utelämnar *Var* såvida inte *Poly1* och *Poly2* är uttryck i samma variabel.

|                                                     |                                                  |
|-----------------------------------------------------|--------------------------------------------------|
| $\text{polyRemainder}(x-1, x-3)$                    | 2                                                |
| $\text{polyRemainder}(x-1, x^2-1)$                  | $x-1$                                            |
| $\text{polyRemainder}(x^2-1, x-1)$                  | 0                                                |
| <hr/>                                               |                                                  |
| $\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)$ | $-(y-z) \cdot (2 \cdot y + z)$                   |
| $\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)$ | $-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2$ |
| $\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)$ | $(x-y) \cdot (x+2 \cdot y)$                      |

**polyRoots**(*Poly*, *Var*)  $\Rightarrow$  lista

$$\text{polyRoots}(y^3+1, y) \quad \{-1\}$$

**polyRoots**(*ListaPåKoeff*)  $\Rightarrow$  lista

$$\text{cPolyRoots}(y^3+1, y) \\ \left\{ -1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i \right\}$$

Den första syntaxen, **polyRoots** (*Poly*, *Var*), ger en lista på reella rötter i polynomet *Poly* med avseende på variabeln *Var*. Om inga reella rötter existerar ger den en tom lista: {}.

$$\text{polyRoots}(x^2+2 \cdot x+1, x) \quad \{-1, -1\}$$

*Poly* måste vara ett polynom i en variabel.

$$\text{polyRoots}(\{1, 2, 1\}) \quad \{-1, -1\}$$

Den andra syntaxen, **polyRoots** (*ListaPåKoeff*) ger en lista på reella rötter för koefficienterna i *ListaPåKoeff*.

**Obs:** Se även **cPolyRoots()**, på sidan 37.

## PowerReg

**PowerReg** *X*, *Y* [, *Freq*] [, *Category*, *Include*]

Utför potensregressionsanalys  $y = (a \cdot (x)^b)$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel    | Beskrivning                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn         | Regressionsekvation: $a \cdot (x)^b$                                                                                                                                        |
| stat.a, stat.b      | Regressionskoefficienter                                                                                                                                                    |
| stat.r <sup>2</sup> | Koefficient för linjär bestämning av transformerade data                                                                                                                    |
| stat.r              | Korrelationskoefficient för transformerade data ( $\ln(x)$ , $\ln(y)$ )                                                                                                     |
| stat.Resid          | Residualer associerade med potensmodellen                                                                                                                                   |
| stat.ResidTrans     | Residualer associerade med linjär anpassning av transformerade data                                                                                                         |
| stat.XReg           | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg           | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg        | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## Prgm

Katalog > 

## Prgm

Beräkna GCD och visa mellanliggande resultat.

*Block*

## EndPrgm

Mall för att skapa ett användardefinierat program. Måste användas med kommandot **Define**, **Define LibPub** eller **Define LibPriv**.

*Block* kan vara ett enstaka påstående, en serie av påståenden separerade med tecknet ":" eller en serie av påståenden på separata rader.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

```

Define proggcd(a,b)=Prgm
    Local d
    While b≠0
        d:=mod(a,b)
        a:=b
        b:=d
    Disp a," ",b
    EndWhile
    Disp "GCD=",a
    EndPrgm

```

*Done*

```

proggcd(4560,450)

```

```

450 60
60 30
30 0
GCD=30

```

*Done*

## product()

Katalog > **product(List[, Start[, end]])** ⇒ *uttryck*

Ger produkten av elementen i *List*. *Start* och *end* är valfria. De specificerar ett område med element.

**product(MatrixI[, Start[, end]])** ⇒ *matrix*

Ger en radvektor som innehåller produkterna av elementen i kolumnerna i *MatrixI*. *Start* och *end* är valfria. De specificerar ett område med rader.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

$$\text{product}(\{1,2,3,4\}) \quad 24$$

$$\text{product}(\{2,x,y\}) \quad 2 \cdot x \cdot y$$

$$\text{product}(\{4,5,8,9\},2,3) \quad 40$$

$$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right) \quad \begin{bmatrix} 28 & 80 & 162 \end{bmatrix}$$

$$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},1,2\right) \quad \begin{bmatrix} 4 & 10 & 18 \end{bmatrix}$$

## propFrac()

Katalog > **propFrac(ExprI[, Var])** ⇒ *uttryck*

**propFrac(rational\_number)** ger *rational\_number* som summan av ett heltal och ett bråk med samma tecken och med större nämnare än täljare.

**propFrac(rational\_expression,Var)** ger summan av egentliga bråk och ett polynom med avseende på *Var*. Graden hos *Var* i nämnaren överskrider graden hos *Var* i täljaren i varje egentligt bråk. Liknande potenser av *Var* samlas in. Termerna och deras faktorer sorteras med *Var* som huvudvariabel.

$$\text{propFrac}\left(\frac{4}{3}\right) \quad 1 + \frac{1}{3}$$

$$\text{propFrac}\left(\frac{-4}{3}\right) \quad -1 - \frac{1}{3}$$

$$\text{propFrac}\left(\frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1}, x\right) \quad \frac{1}{x+1} + x + \frac{y^2+y+1}{y+1}$$

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

Om *Var* utelämnas utförs en utveckling av ett egentligt bråk med avseende på den mest betydande variabeln.

Koefficienterna för polynomdelen görs sedan egentliga, först med avseende på deras mest betydande variabel och så vidare.

För rationella uttryck är **propFrac()** ett snabbare, men mindre extremt alternativ till **expand()**.

Du kan använda funktionen **propFrac()** för att representera blandade bråk och demonstrera addition och subtraktion av blandade bråk.

|                                                                               |                      |
|-------------------------------------------------------------------------------|----------------------|
| $\text{propFrac}\left(\frac{11}{7}\right)$                                    | $1 + \frac{4}{7}$    |
| $\text{propFrac}\left(3 + \frac{1}{11} + 5 + \frac{3}{4}\right)$              | $8 + \frac{37}{44}$  |
| $\text{propFrac}\left(3 + \frac{1}{11} - \left(5 + \frac{3}{4}\right)\right)$ | $-2 - \frac{29}{44}$ |

## Q

### QR

**QR** *Matrix, qMatName, rMatName[, Tol]*

Beräknar faktoriseringen Householder QR av en reell eller komplex matris. De resulterande Q- och R-matriserna lagras i specificerad *MatNames*. Q-matrisen är unitär. R-matrisen är övertriangulär.

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tol*. Denna tolerans används endast om matrisen har inmatning i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tol*.

- Om du använder **ctrl enter** eller ställer in **Auto** eller **Ungefärlig** på Approximate utförs beräkningarna med flyttalsaritmetik.
- Om *Tol* utelämnas eller inte används beräknas standardtoleransen som:

Siffran för flytande komma (9.) i m1 medför att resultat beräknas med flyttalsaritmetik.

|                                                                                     |                                                                                                                                     |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$                                                                |
| QR m1,qm,rm                                                                         | Done                                                                                                                                |
| qm                                                                                  | $\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$ |
| rm                                                                                  | $\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$                              |

5E-14  $\cdot \max(\dim(\text{Matrix})) \cdot \text{rowNorm}(\text{Matrix})$

QR-faktoriseringen beräknas numeriskt med Householder-transformationer. Den symboliska lösningen beräknas med Gram-Schmidt. Kolumnerna i  $q\text{MatName}$  är de ortonormerade basvektorerna som täcker utrymmet som definieras av  $\text{matrix}$ .

|                                                               |                                                                                                                                                                                                                               |                                                |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| $\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow mI$ |                                                                                                                                                                                                                               | $\begin{bmatrix} m & n \\ o & p \end{bmatrix}$ |
| QR $mI, qm, rm$                                               |                                                                                                                                                                                                                               | Done                                           |
| $qm$                                                          | $\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$ |                                                |
| $rm$                                                          | $\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}} \end{bmatrix}$                                                                           |                                                |

## QuadReg

**QuadReg**  $X, Y [, \text{Freq}] [, \text{Category}, \text{Include}]$

Utför den kvadratiske regressionsanalysen  $y = a \cdot x^2 + b \cdot x + c$  på listorna  $X$  och  $Y$  med frekvensen  $\text{Freq}$ . En sammanfattning av resultaten visas i variabeln  $\text{stat.results}$ . (Se på sidan 184.)

Alla listor utom  $\text{Include}$  måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

$\text{Freq}$  är en frivillig lista på frekvensvärden. Varje element i  $\text{Freq}$  specificerar frekvensen för varje motsvarande  $X$ - och  $Y$ -datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

$\text{Category}$  är en lista på kategorikoder för motsvarande  $X$ - och  $Y$ -data.

$\text{Include}$  är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element", på sidan 261.

| Resultatvariabel       | Beskrivning                                                                                                                                                                 |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn            | Regressionsekvation: $a \cdot x^2 + b \cdot x + c$                                                                                                                          |
| stat.a, stat.b, stat.c | Regressionskoefficienter                                                                                                                                                    |
| stat.R <sup>2</sup>    | Determinationskoefficient                                                                                                                                                   |
| stat.Resid             | Residualer från regressionen                                                                                                                                                |
| stat.XReg              | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg              | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg           | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## QuartReg

**QuartReg** *X,Y [, Freq] [, Category, Include]*

Utför en fjärdegrads regressionsanalys  $y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$  på listorna *X* och *Y* med frekvensen *Freq*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

*X* och *Y* är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel                       | Beskrivning                                                                                                                                                                 |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.RegEqn                            | Regressionsekvation: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$                                                                                              |
| stat.a, stat.b, stat.c, stat.d, stat.e | Regressionskoefficienter                                                                                                                                                    |
| stat.R <sup>2</sup>                    | Determinationskoefficient                                                                                                                                                   |
| stat.Resid                             | Residualer från regressionen                                                                                                                                                |
| stat.XReg                              | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg                              | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg                           | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## R

### R ► P0()

**R ► P0** (*xExpr*, *yExpr*) ⇒ *uttryck*

**R ► P0** (*xList*, *yList*) ⇒ *lista*

**R ► P0** (*xMatrix*, *yMatrix*) ⇒ *matrix*

Ger den ekvivalenta  $\theta$ -koordinaten för (*x*,*y*) värden.

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer för respektive inställning av vinkelenhet.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **R@Ptheta (...)**.

Med vinkelmått Grader:

$$\text{R} \blacktriangleright \text{P0}(x, y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Med vinkelenhet Nygrader:

$$\text{R} \blacktriangleright \text{P0}(x, y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Med vinkelenhet Radianer:

$$\begin{aligned} \text{R} \blacktriangleright \text{P0}(3, 2) & \quad \tan^{-1}\left(\frac{2}{3}\right) \\ \text{R} \blacktriangleright \text{P0}\left([3 \quad -4 \quad 2], \left[0 \quad \frac{\pi}{4} \quad 1.5\right]\right) & \quad \left[0 \quad \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \quad 0.643501\right] \end{aligned}$$



**R► Pr()****Katalog >** **R► Pr** (*xExpr*, *yExpr*) ⇒ *uttryck***R► Pr** (*xList*, *yList*) ⇒ *lista***R► Pr** (*xMatrix*, *yMatrix*) ⇒ *matrix*

Ger den ekvivalenta r-koordinaten för argumentparen (*x*,*y*).

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **R@Pr (...)**.

Med vinkelenhet Radianer:

|                                                                                                                                       |                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| $R\blacktriangleright Pr(3,2)$                                                                                                        | $\sqrt{13}$                                                          |
| $R\blacktriangleright Pr(x,y)$                                                                                                        | $\sqrt{x^2+y^2}$                                                     |
| $R\blacktriangleright Pr\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right)$ | $\begin{bmatrix} 3 & \frac{\sqrt{\pi^2+256}}{4} & 2.5 \end{bmatrix}$ |

**► Rad****Katalog >** *Expr1* ► *Rad* ⇒ *uttryck*

Konverterar argumentet till en vinkel i radianer.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **@>Rad**.

Med vinkelmått Grader:

|                                |               |
|--------------------------------|---------------|
| $(1.5)\blacktriangleright Rad$ | $(0.02618)^r$ |
|--------------------------------|---------------|

Med vinkelenhet Nygrader:

|                                |                |
|--------------------------------|----------------|
| $(1.5)\blacktriangleright Rad$ | $(0.023562)^r$ |
|--------------------------------|----------------|

**rand( )****Katalog >** **rand( )** ⇒ *uttryck***rand**(*#Trials*) ⇒ *lista*

**rand( )** ger ett slumpvärde mellan 0 och 1.

**rand**(*#Trials*) ger en lista med *#Trials* slumpvärden mellan 0 och 1.

Bestämmer slumpvalsfröet.

|               |                          |
|---------------|--------------------------|
| RandSeed 1147 | Done                     |
| $rand(2)$     | $\{0.158206, 0.717917\}$ |

**slumpBin( )****Katalog >** **randBin**(*n*, *p*) ⇒ *uttryck***randBin**(*n*, *p*, *#Trials*) ⇒ *lista*

**randBin**(*n*, *p*) ger ett reellt slumptal från en specificerad binomialfördelning.

**randBin**(*n*, *p*, *#Trials*) ger en lista med *#Trials* reella slumptal från en specificerad binomialfördelning.

|                     |                  |
|---------------------|------------------|
| $randBin(80,0.5)$   | 42               |
| $randBin(80,0.5,3)$ | $\{41, 32, 39\}$ |

**slumpBin( )**Katalog > 

**randInt**  
(*lowBound, upBound*)  
⇒ *uttryck*

|                         |             |
|-------------------------|-------------|
| <b>randInt</b> (3,10)   | 5           |
| <b>randInt</b> (3,10,4) | { 9,7,5,8 } |

**randInt**  
(*lowBound, upBound*,  
*#Trials*) ⇒ *lista*

**randInt**  
(*lowBound, upBound*)  
ger ett slumptal  
med heltalsvärde  
inom det område  
som specificeras av  
heltalsgränserna  
*lowBound* och  
*upBound*.

**randInt**  
(*lowBound, upBound*,  
*#Trials*) ger en lista  
med *#Trials*  
slumptal med  
heltalsvärden inom  
det specificerade  
området.

**randMat( )**Katalog > 

**randMat**(*numRows, numColumns*) ⇒  
*matris*

Ger en matris med heltal mellan -9 och 9  
med specificerad dimension.

Båda argumenten måste förenklas till  
heltal.

|               |      |    |    |
|---------------|------|----|----|
| RandSeed 1147 | Done |    |    |
| randMat(3,3)  | 8    | -3 | 6  |
|               | -2   | 3  | -6 |
|               | 0    | 4  | -6 |

**Obs:** Värdena i denna matris ändras varje gång  
du trycker på **enter**.

**slumpNorm( )**Katalog > 

**randNorm**( $\mu, \sigma$ ) ⇒ *uttryck*  
**randNorm**( $\mu, \sigma, \#Trials$ ) ⇒ *lista*

**randNorm**( $\mu, \sigma$ ) ger ett decimalt tal från  
den specificerade normalfördelningen.  
Det kan vara ett reellt tal vilket som  
helst, men det blir kraftigt koncentrerat i  
intervallet  $[\mu - 3 \cdot \sigma, \mu + 3 \cdot \sigma]$ .

|                         |          |
|-------------------------|----------|
| RandSeed 1147           | Done     |
| <b>randNorm</b> (0,1)   | 0.492541 |
| <b>randNorm</b> (3,4.5) | -3.54356 |

**slumpNorm( )****Katalog >** 

**randNorm( $\mu$ ,  $\sigma$ , #Trials)** ger en lista med #Trials decimaltal från den specificerade normalfördelningen.

**randPoly()****Katalog >** 

**randPoly(Var, Order)**  $\Rightarrow$  *uttryck*

Ger ett polynom i *Var* i specificerad *Order*. Koefficienterna är slumpmässiga heltal i intervallet  $-9$  till  $9$ . Den första koefficienten kommer inte att vara noll.

*Order* måste vara  $0-99$ .

|               |                                                            |
|---------------|------------------------------------------------------------|
| RandSeed 1147 | Done                                                       |
| randPoly(x,5) | $-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$ |

**randSamp( )****Katalog >** 

**randSamp(List, #Trials[, noRepl])**  $\Rightarrow$  *lista*

Ger en lista på ett slumpmässigt urval av #Trials försök från *List* med ett alternativ för återläggning (*noRepl*=0) eller ingen återläggning (*noRepl*=1). Förinställningen är med urvalsutbyte.

|                                |               |
|--------------------------------|---------------|
| Define list3={1,2,3,4,5}       | Done          |
| Define list4=randSamp(list3,6) | Done          |
| list4                          | {2,3,4,3,1,2} |

**RandSeed****Katalog >** 

**RandSeed Tal**

Om *Number* = 0 ställs fröna in på fabriksinställningarna för slumpalsgeneratorn. Om *Number*  $\neq 0$ , används det för att generera två frön, vilka lagras i systemvariablerna seed1 och seed2.

|               |          |
|---------------|----------|
| RandSeed 1147 | Done     |
| rand()        | 0.158206 |

**real( )****Katalog >** 

**real(Expr1)** *uttryck*

Ger argumentets reella del.

**Obs:** Alla odefinierade variabler behandlas som reella variabler. Se även **imag( )**, på sidan 92.

**real(List1)**  $\Rightarrow$  *lista*

|                       |     |
|-----------------------|-----|
| real( $2+3 \cdot i$ ) | 2   |
| real( $z$ )           | $z$ |
| real( $x+i \cdot y$ ) | $x$ |

|                               |               |
|-------------------------------|---------------|
| real({ $a+i \cdot b, 3, i$ }) | { $a, 3, 0$ } |
|-------------------------------|---------------|

Ger de reella delarna av alla element.

**real**(*Matrix1*)  $\Rightarrow$  *matrix*

Ger de reella delarna av alla element.

$$\text{real}\left(\begin{bmatrix} a+i\cdot b & 3 \\ c & i \end{bmatrix}\right) \quad \begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$$

## ► Rect

*Vector* ► **Rect**

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **@Rect**.

Visar *Vector* i rektangulär form [x, y, z]. Vektorn måste ha dimensionen 2 eller 3 och kan vara en rad eller en kolumn.

**Obs:** ► **Rect** är en visa format-instruktion, inte en konverteringsfunktion. Du kan endast använda den i slutet av en inmatningsrad, och den uppdaterar inte *ans*.

**Obs:** Se även ► **Polar**, på sidan 138.

*complexValue* ► **Rect**

Visar *complexValue* i rektangulär form  $a+bi$ . *complexValue* kan ha valfri komplex form. En inmatning av  $re^{i\theta}$  orsakar dock ett fel i vinkelläget Grader.

**Obs:** Du måste använda parenteserna för en ( $r\angle \theta$ ) polär inmatning.

$$\left(\begin{bmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{bmatrix}\right) \text{►Rect} \quad \begin{bmatrix} \frac{3\cdot\sqrt{2}}{4} & \frac{3\cdot\sqrt{2}}{4} & \frac{3\cdot\sqrt{3}}{2} \end{bmatrix}$$

$$\begin{bmatrix} a & \angle b & \angle c \end{bmatrix} \quad \begin{bmatrix} a\cdot\cos(b)\cdot\sin(c) & a\cdot\sin(b)\cdot\sin(c) & a\cdot\cos(c) \end{bmatrix}$$

Med vinkelenhet Radianer:

$$\left(4\cdot e^{\frac{\pi}{3}}\right) \text{►Rect} \quad \frac{\pi}{4\cdot e^3}$$

$$\left(\left(4\angle\frac{\pi}{3}\right)\right) \text{►Rect} \quad 2+2\cdot\sqrt{3}\cdot i$$

Med vinkelenhet Nygrader:

$$\left((1\angle 100)\right) \text{►Rect} \quad i$$

Med vinkelmått Grader:

$$\left((4\angle 60)\right) \text{►Rect} \quad 2+2\cdot\sqrt{3}\cdot i$$

**Obs:** För att skriva in tecknet  $\angle$ , välj det från symbollistan i Katalog.

$\text{ref}(\text{Matrix}1, \text{Tol}) \Rightarrow \text{matrix}$

Ger radtrappstegsformen av *Matrix1*.

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tol*. Denna tolerans används endast om matrisen har inmatning i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tol*.

- Om du använder   eller ställer in **Auto or Approximate** på Approximate, utförs beräkningarna med flyttalsaritmetik.
- Om *Tol* utelämnas eller inte används beräknas standardtoleransen som:  
 $5\text{E}-14 \cdot \max(\dim(\text{Matrix}1)) \cdot \text{rowNorm}(\text{Matrix}1)$

Undvik odefinierade element i *Matrix1*. De kan leda till oväntade resultat.

Om till exempel *a* är odefinierat i följande uttryck visas ett varningsmeddelande och resultatet ges som:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Varningen visas på grund av att det generaliserade elementet  $1/a$  inte skulle vara giltigt för  $a=0$ .

Du kan undvika detta genom att i förväg lagra ett värde i *a* eller genom att använda ("|")-operatoren begränsning för att ersätta ett värde såsom visas i följande exempel.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

$$\begin{array}{|l|} \hline \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow m1 \\ \hline \text{ref}(m1) \\ \hline \end{array} \quad \begin{array}{|l|} \hline \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ \hline \begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix} \\ \hline \end{array}$$

$$\text{ref} \left( \begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) | a=0 \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

**Obs:** Se även **rref ( )**, på sidan 162.

## RefreshProbeVars

## RefreshProbeVars

Låter dig visa sensordata från alla anslutna sensorer via TI-grundprogrammet.

| StatusVar-värde     | Status                                                                                                                                                                                       |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>statusVar</i> =0 | Normal (fortsätt med programmet)<br>Vernier DataQuest™-applikationen är i datainsamlingsläge.                                                                                                |
| <i>statusVar</i> =1 | <b>Obs:</b> Vernier DataQuest™-applikationen måste vara i läge "meter" för att detta kommando ska fungera.  |
| <i>statusVar</i> =2 | Vernier DataQuest™-applikationen har inte startats.                                                                                                                                          |
| <i>statusVar</i> =3 | Vernier DataQuest™-applikationen har startats men inga givare har anslutits.                                                                                                                 |

## Exempel

```
Define temp()=
```

```
Prgm
```

```
© Kontrollera om systemet är klart
```

```
RefreshProbeVars status
```

```
If status=0 Then
```

```
Disp "klar"
```

```
För n,1,50
```

```
RefreshProbeVars status
```

```
temperatur:=mätare.temperatur
```

```
Disp "Temperatur: ",temperatur
```

```
If temperatur>30 Then
```

```
Disp "För varm"
```

```
EndIf
```

```
© Vänta 1 sekund mellan mätningarna
```

```
Wait 1
```

```
EndFor
```

```
Else
```

```
Disp "Ej klar. Försök igen senare"
```

```
EndIf
```

```
EndPrgm
```

**Obs:** Detta kan även användas med TI-Innovator™ hubb.

## remain( )

Katalog > 

**remain**(*Expr1*, *Expr2*)  $\Rightarrow$  *uttryck*  
**remain**(*List1*, *List2*)  $\Rightarrow$  *lista*  
**remain**(*Matrix1*, *Matrix2*)  $\Rightarrow$  *matrix*

Ger resten på det första argumentet med avseende på det andra argumentet definierat av identiteterna:

$\text{remain}(x,0) = x$   
 $\text{remain}(x,y) = x - y \cdot \text{iPart}(x/y)$

Som en följd av detta, observera att **remain**(-*x*,*y*) = **remain**(*x*,*y*). Resultatet är antingen noll eller har samma tecken som det första argumentet.

**Obs:** Se även **mod**( ), på sidan 121.

|                                           |             |
|-------------------------------------------|-------------|
| $\text{remain}(7,0)$                      | 7           |
| $\text{remain}(7,3)$                      | 1           |
| $\text{remain}(-7,3)$                     | -1          |
| $\text{remain}(7,-3)$                     | 1           |
| $\text{remain}(-7,-3)$                    | -1          |
| $\text{remain}(\{12,-14,16\},\{9,7,-5\})$ | $\{3,0,1\}$ |

|                                                                                                                          |                                                 |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| $\text{remain}\left(\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix}\right)$ | $\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$ |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|

## Request

Katalog > 

**Begär** *promptString*, *var*[, *DispFlag* [, *statusVar*]]

**Request** *promptString*, *func*(*arg1*, ...*argn*) [, *DispFlag* [, *statusVar*]]

Programmeringskommando: Pausar programmet och visar en dialogruta med meddelandet *promptString* och en svarsruta där användaren ska skriva in svaret.

När användaren skriver in svaret och klickar på **OK** tilldelas variabeln *vardet* värde som står i svarsrutan.

Om användaren klickar på **Cancel** (Avbryt) fortsätter programmet utan att acceptera någon inmatning. Programmet använder det tidigare värdet på *var* om *var* redan var definierad.

Det valfria argumentet *DispFlag* kan vara ett valfritt uttryck.

- Om *DispFlag* utelämnas eller beräknas till **1**, visas promptmeddelandet och användarens svar i Räknares historik.
- Om *DispFlag* beräknas till **0** visas inte prompten och svaret i historiken.

Definiera ett program:

```
Definiera begär_demo()=Prgm
  Begär "Radie: ",r
  Disp "Area = ",pi*r^2
EndPrgm
```

Kör programmet och skriv in ett svar:

`request_demo( )`



Resultat efter tryckning på **OK**:

Radie: 6/2  
Area= 28,2743

Det valfria argumentet *statusVar* ger programmet ett sätt att bestämma hur användaren lämnade dialogrutan. Observera att *statusVar* är beroende av argumentet *DispFlag*.

- Om användaren klickade på **OK** eller tryckte på **Enter** eller **Ctrl+Enter** ges variabeln *statusVar* värdet **1**.
- Annars får variabeln *statusVar* värdet **0**.

Med argumentet *funk( )* kan programmet lagra användarens svar som en funktionsdefinition. Denna syntax fungerar som om användaren exekverade kommandot:

Definiera *funk(arg1, ...argn) = användarens svar*

Programmet kan sedan använda den definierade funktionen *func()*. Strängen *promptString* bör vägleda användaren till att skriva in ett lämpligt *användarsvar* som fullbordar funktionsdefinitionen.

**Obs:** Du kan använda Request - kommandot med ett användardefinierat program, men inte inom en funktion.

För att stoppa ett program som innehåller ett **Request**-kommando inne i en oändlig slinga:

- **Handenhet:** Håll ned  och tryck på  upprepade gånger.
- **Windows®:** Håll ned **F12** och tryck på **Enter** upprepade gånger.
- **Macintosh®:** Håll ned **F5** och tryck på **Enter** upprepade gånger.
- **iPad®:** Appen visar en uppmaning. Du kan fortsätta att vänta eller avbryta.

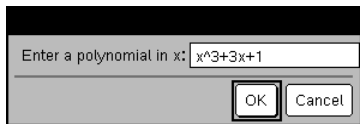
**Obs:** Se även **RequestStr**, på sidan 157.

Definiera ett program:

```
Definiera polynom()=Prgm
  Request "Skriv in ett polynom i
x:",p(x)
  Disp "Reella rötter är:",polyRoots
(p(x),x)
EndPrgm
```

Kör programmet och skriv in ett svar:

polynom()



Resultat efter inmatning av  $x^3+3x+1$  och tryckning på **OK**:

Reella rötter är:  $\{-0,322185\}$



**RequestStr** *promptString*, var[, *DispFlag*]

Programmeringskommando: Fungerar på precis samma sätt som den första syntaxen i kommandot **Request**, förutom att användarens svar alltid tolkas som en sträng. Kommandot **Request** tolkar svaret som ett uttryck såvida inte användaren omger det med citationstecken ("").

Obs: Kommandot **RequestStr** kan användas inom ett användardefinierat program, men inte inom en funktion.

Att stoppa ett program som innehåller ett **RequestStr**-kommando i en oändlig loop:

- **Handenhet:** Håll ned  och tryck på  upprepade gånger.
- **Windows®:** Håll ned **F12** och tryck på **Enter** upprepade gånger.
- **Macintosh®:** Håll ned **F5** och tryck på **Enter** upprepade gånger.
- **iPad®:** Appen visar en uppmaning. Du kan fortsätta att vänta eller avbryta.

**Obs:** Se även **Request**, på sidan 155.

Definiera ett program:

```
Definiera requestStr_demo()=Prgm
  BegärStr "Ditt namn:",namn,0
  Disp "Svaret har ",dim(namn),"
  tecken."
EndPrgm
```

Kör programmet och skriv in ett svar:

begärStr\_demo()



Resultat efter tryckning på **OK** (observera att argumentet *DispFlag* för **0** förbiser prompten och svarar från historiken):

begärStr\_demo()

Svaret har 5 tecken.

## Return

**Return** [*Expr*]

Ger *Expr* som resultatet av funktionen. Använd inom ett **Func...EndFunc**-block.

**Obs:** Använd **Return** utan ett argument inom ett **Prgm...EndPrgm**-block för att gå ur ett program.

**Obs för att mata in exemplet:** Se avsnittet Räkare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer· counter → answer
EndFor
Return answer
EndFunc

factorial (3)
```

6

**right()**Katalog > **right(List1[, Num])** ⇒ lista $\text{right}(\{1, 3, -2, 4\}, 3)$   $\{3, -2, 4\}$ 

Ger *Num*-elementen längst till höger i *List1*.

Om du utelämnar *Num* erhålls alla i *List1*.

**right(sourceString[, Num])** ⇒ sträng $\text{right}(\text{"Hello"}, 2)$  "lo"

Ger *Num*-tecknen längst till höger i teckensträngen *sourceString*.

Om du utelämnar *Num* erhålls alla i *sourceString*.

**right(Comparison)** ⇒ uttryck $\text{right}(x < 3)$  3

Ger den högra sidan av en ekvation eller olikhet.

**rk23 ()**Katalog > 

**rk23(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep[, diftol])** ⇒ *matrix*

**rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])** ⇒ *matrix*

**rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])** ⇒ *matrix*

Använder Runge-Kuttas metod för att lösa systemet

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

med  $\text{depVar}(\text{Var}0) = \text{depVar}0$  i intervallet  $[\text{Var}0, \text{VarMax}]$ . Ger en matris vars första rad definierar resultatvärdena för *Var*, definierade av *VarStep*. Den andra raden definierar värdet på den första lösningskomponenten vid motsvarande värden på *Var*, och så vidare.

Differentialekvation:

$$y' = 0.001 \cdot y \cdot (100 - y) \text{ och } y(0) = 10$$

$$\text{rk23}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1)$$

|  | 0.  | 1.      | 2.      | 3.     | 4.   |
|--|-----|---------|---------|--------|------|
|  | 10. | 10.9367 | 11.9493 | 13.042 | 14.2 |

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

Samma ekvation med *diftol* inställd på  $1.E-6$

$$\text{rk23}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6)$$

|  | 0.  | 1.      | 2.      | 3.      | 4.      |
|--|-----|---------|---------|---------|---------|
|  | 10. | 10.9367 | 11.9495 | 13.0423 | 14.2189 |

Jämför ovanstående resultat med exakt lösning med CAS-verktyg som erhållits med **deSolve()** och **seqGen()**:

$$\text{deSolve}(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9.}$$

*Expr* är det högra ledet som definierar den ordinära differentialekvationen (ODE).

*SystemOfExpr* är ett system av högerled som definierar systemet av ODE:er (motsvarar ordningen av oberoende variabler i *ListOfDepVars*).

*ListOfExpr* är en lista på högerled som definierar systemet av ODE:er (motsvarar ordningen av oberoende variabler i *ListOfDepVars*).

*Var* är den oberoende variabeln.

*ListOfDepVars* är en lista på oberoende variabler.

$\{Var0, VarMax\}$  är en lista med två element som instruerar funktionen att integrera från *Var0* till *VarMax*.

*ListOfDepVars0* är en lista på startvärden för oberoende variabler.

Om *VarStep* utvärderas till ett tal skilt från noll ges  $\text{sign}(VarStep) = \text{sign}(VarMax - Var0)$  och lösningar vid  $Var0 + i \cdot VarStep$  för alla  $i=0,1,2,\dots$  sådana att  $Var0 + i \cdot VarStep$  är i intervallet  $[var0, VarMax]$  (kanske inte ger ett lösningsvärde vid *VarMax*).

Om *VarStep* utvärderas till noll ges lösningar vid "Runge-Kutta"-värdena för *Var*.

*diftol* är feltoleransen (föreställs på 0,001).

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.49\}$$

Ekvationssystem:

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

med  $y1(0)=2$  och  $y2(0)=5$

$$\text{rk23}\left(\begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

|    |         |         |         |         |
|----|---------|---------|---------|---------|
| 0. | 1.      | 2.      | 3.      | 4.      |
| 2. | 1.94103 | 4.78694 | 3.25253 | 1.82848 |
| 5. | 16.8311 | 12.3133 | 3.51112 | 6.27245 |

## Rot()

**root**(*Expr*)  $\Rightarrow$  *root*

**root**(*Expr1*, *Expr2*)  $\Rightarrow$  *root*

**root**(*Expr*) ger kvadratroten ur *Expr*.

|                |                 |
|----------------|-----------------|
| $\sqrt[3]{8}$  | 2               |
| $\sqrt[3]{3}$  | $\frac{1}{3^3}$ |
| $\sqrt[3]{3.}$ | 1.44225         |

**root**(*Expr1*, *Expr2*) ger *Expr2*-roten ur *Expr1*. *Expr1* kan vara en reell eller komplex konstant i flyttalsform, ett heltal eller komplex rationell konstant eller ett allmänt symboliskt uttryck.

**Obs:** Se även **Nth root template**, på sidan 1.

**rotate()**

**rotate**(*Integer1* [, *#ofRotations*])  $\Rightarrow$  *heltal*

Roterar bitarna i ett binärt heltal. *Integer1* kan anges i valfri talbas. Det omvandlas automatiskt till 64 positioners binär form. Om storleken på *Integer1* är alltför stor för denna form, för en symmetrisk modulooperation talet inom området. För mer information, se ► **Base2**, på sidan 18.

Om *#ofRotations* är positiv sker rotationen åt vänster. Om *#ofRotations* är negativ sker rotationen åt höger. Förinställningen är  $-1$  (rotera en position åt höger).

Vid exempelvis rotation åt höger:

Varje bit roteras åt höger.

0b00000000000001111010110000110101

Biten längst till höger roteras till positionen längst till vänster.

ger:

0b10000000000000111101011000011010

Resultatet visas enligt det inställda basläget.

**rotate**(*List1* [, *#ofRotations*])  $\Rightarrow$  *lista*

Ger en kopia av *List1* roterad åt höger eller vänster av *#ofRotations*-elementen. Ändrar inte *List1*.

I binärt basläge:

|                                                         |                           |
|---------------------------------------------------------|---------------------------|
| <code>rotate(0b11111111111111111111111111111111)</code> |                           |
| <code>0b100000000000000000000000000000001</code>        |                           |
| <code>rotate(256,1)</code>                              | <code>0b1000000000</code> |

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

I hexadecimalt basläge:

|                               |                                                    |
|-------------------------------|----------------------------------------------------|
| <code>rotate(0h78E)</code>    | <code>0h3C7</code>                                 |
| <code>rotate(0h78E,-2)</code> | <code>0h800000000000000000000000000000001E3</code> |
| <code>rotate(0h78E,2)</code>  | <code>0h1E38</code>                                |

**Viktigt:** För att skriva in ett binärt eller hexadecimalt tal, använd alltid prefixet 0b eller 0h (noll, inte bokstaven O).

I decimalt basläge:

|                                   |                        |
|-----------------------------------|------------------------|
| <code>rotate({1,2,3,4})</code>    | <code>{4,1,2,3}</code> |
| <code>rotate({1,2,3,4},-2)</code> | <code>{3,4,1,2}</code> |
| <code>rotate({1,2,3,4},1)</code>  | <code>{2,3,4,1}</code> |

**rotate()**Katalog > 

Om *#ofRotations* är positiv sker rotationen åt vänster. Om *#ofRotations* är negativ sker rotationen åt höger. Förinställningen är  $-1$  (rotera ett element åt höger).

**rotate**(*String1* [, *#ofRotations*])  $\Rightarrow$  *sträng*

Ger en kopia av *String1* roterad åt höger eller vänster av *#ofRotations*-tecknen. Ändrar inte *String1*.

|                                 |        |
|---------------------------------|--------|
| <code>rotate("abcd")</code>     | "dabc" |
| <code>rotate("abcd", -2)</code> | "cdab" |
| <code>rotate("abcd", 1)</code>  | "bcda" |

Om *#ofRotations* är positiv sker rotationen åt vänster. Om *#ofRotations* är negativ sker rotationen åt höger. Förinställningen är  $-1$  (rotera ett tecken åt höger).

**round()**Katalog > 

**round**(*Expr1* [, *digits*])  $\Rightarrow$  *uttryck*

Ger argumentet avrundat till det specificerade antalet siffror efter decimalpunkten.

*digits* måste vara ett heltal i intervallet 0–12. Om *digits* inte ingår, ges argumentet avrundat till 12 signifikanta siffror.

**Obs:** Läget Display digits (Visa siffror) kan påverka hur detta visas.

**round**(*List1* [, *digits*])  $\Rightarrow$  *lista*

Ger en lista på elementen avrundade till det specificerade antalet siffror.

**round**(*Matrix1* [, *digits*])  $\Rightarrow$  *matris*

Ger en matris över elementen avrundade till det specificerade antalet siffror.

|                                 |       |
|---------------------------------|-------|
| <code>round(1.234567, 3)</code> | 1.235 |
|---------------------------------|-------|

|                                                                                       |                            |
|---------------------------------------------------------------------------------------|----------------------------|
| <code>round({<math>\pi</math>, <math>\sqrt{2}</math>, <math>\ln(2)</math>}, 4)</code> | { 3.1416, 1.4142, 0.6931 } |
|---------------------------------------------------------------------------------------|----------------------------|

|                                                                                                        |                                                        |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <code>round(<math>\begin{bmatrix} \ln(5) &amp; \ln(3) \\ \pi &amp; e^1 \end{bmatrix}</math>, 1)</code> | $\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$ |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------|

**rowAdd()**Katalog > 

**rowAdd**(*Matrix1*, *rIndex1*, *rIndex2*)  $\Rightarrow$  *matris*

Ger en kopia av *Matrix1* med rad *rIndex2* ersatt av summan av raderna *rIndex1* och *rIndex2*.

|                                                                                                |                                                    |
|------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <code>rowAdd(<math>\begin{bmatrix} 3 &amp; 4 \\ -3 &amp; -2 \end{bmatrix}</math>, 1, 2)</code> | $\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$     |
| <code>rowAdd(<math>\begin{bmatrix} a &amp; b \\ c &amp; d \end{bmatrix}</math>, 1, 2)</code>   | $\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$ |

**rowDim()**Katalog > **rowDim**(*Matrix*)  $\Rightarrow$  uttryckGer antalet rader i *Matrix*.**Obs:** Se även **colDim()**, på sidan 26.

|                                                                        |                                                         |
|------------------------------------------------------------------------|---------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ |
| <b>rowDim</b> ( <i>m1</i> )                                            | 3                                                       |

**rowNorm()**Katalog > **rowNorm**(*Matrix*)  $\Rightarrow$  uttryckGer maximum av summorna av absolutbeloppen på elementen i raderna i *Matrix*.**Obs:** Alla matriselement måste förenklas till tal. Se även **colNorm()**, på sidan 26.

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| <b>rowNorm</b> $\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right)$ | 25 |
|-----------------------------------------------------------------------------------------------------|----|

**rowSwap()**Katalog > **rowSwap**(*Matrix1*, *rIndex1*, *rIndex2*)  
 $\Rightarrow$  *matrix*Ger *Matrix1* med raderna *rIndex1* och *rIndex2* växlade.

|                                                                         |                                                         |
|-------------------------------------------------------------------------|---------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ |
| <b>rowSwap</b> ( <i>mat</i> ,1,3)                                       | $\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$ |

**rref()**Katalog > **rref**(*Matrix1*[, *Tol*])  $\Rightarrow$  *matrix*Ger den reducerade radtrappstegsformen av *Matrix1*.

|                                                                                                                |                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| $\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$ | $\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$ |
|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tol*. Denna tolerans används endast om matrisen har inmatning i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tol*.

|                                                                                                                                                            |                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
|  $\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$ | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|

- Om du använder   eller ställer in **Auto** or **Approximate** på

Approximate, utförs beräkningarna med flyttalsaritmetik.

- Om *Tol* utelämnas eller inte används beräknas standardtoleransen som:  
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

**Obs:** Se även `ref()`, på sidan 153.

## S

## sec()

 tangent

`sec(Expr1)`  $\Rightarrow$  *uttryck*

I vinkelläget Grader:

`sec(List1)`  $\Rightarrow$  *lista*

Ger sekansfunktionen för *Expr1* eller en lista på sekansfunktionerna för alla element i *List1*.

$$\frac{\sec(45)}{\sec(\{1, 2, 3, 4\})} \frac{\sqrt{2}}{\left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\}}$$

**Obs:** Argumentet tolkas som en vinkel i grader, nygrader eller radianer enligt det inställda vinkelläget. Du kan använda  $^{\circ}$ ,  $^{\text{G}}$  eller  $^{\text{r}}$  för att tillfälligt överstyra vinkelläget.

sec<sup>-1</sup>() tangent

`sec-1(Expr1)`  $\Rightarrow$  *uttryck*

I vinkelläget Grader:

`sec-1(List1)`  $\Rightarrow$  *lista*

$$\sec^{-1}(1) \quad 0$$

Ger den vinkel vars sekansfunktion är *Expr1* eller en lista på de inversa sekansfunktionerna för alla element i *List1*.

I vinkelläget Nygrader:

$$\sec^{-1}(\sqrt{2}) \quad 50$$

I vinkelläget Radianer:

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

$$\sec^{-1}(\{1, 2, 5\}) \quad \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\}$$

## sec<sup>-1</sup>()

 tangent

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arcsec (...)**.

## sech()

Katalog > 

**sech(Expr1) ⇒ uttryck**

$$\frac{\operatorname{sech}(3)}{\cosh(3)}$$

**sech(List1) ⇒ lista**

$$\operatorname{sech}(\{1, 2, 3, 4\})$$
$$\left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\}$$

Ger den hyperboliska sekansfunktionen för *Expr1* eller en lista på de hyperboliska sekansfunktionerna för elementen i *List1*.

## sech<sup>-1</sup>()

Katalog > 

**sech<sup>-1</sup>(Expr1) ⇒ uttryck**

I vinkelläget Radianer och i Rektangulärt komplext läge:

**sech<sup>-1</sup>(List1) ⇒ lista**

$$\frac{\operatorname{sech}^{-1}(1)}{\operatorname{sech}^{-1}(\{1, -2, 2, 1\})}$$
$$\left\{ 0, \frac{2 \cdot \pi}{3} \cdot i, 8. \text{E}^{-15} + 1.07448 \cdot i \right\}$$

Ger den inversa hyperboliska sekansfunktionen för *Expr1* eller en lista på den inversa hyperboliska sekansfunktionen för alla element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arcsech (...)**.

## Send

Hubb-meny

**Send** *UtrEllerSträng1* [, *UtrEllerSträng2*] ...

Programmeringskommando: Skickar ett eller flera TI-Innovator™ Hub kommandon till en ansluten hubb.

Exempel: Slå på den blå komponenten i den inbyggda RGB-lysdioden i 0,5 sekunder.

Send "SET COLOR.BLUE ON TIME .5"

*Done*

Exempel: Begär nuvarande värde från hubbens inbyggda ljusnivåsensor. Ett **Get**-kommando hämtar värdet och tilldelar det till variabeln *lightval*.



## Send

## Hubb-menyn

*exprOrString* måste vara ett giltigt TI-Innovator™ Hub Kommando.  
*exprOrString* innehåller vanligen ett "SET ..." -kommando för att styra en enhet eller ett "READ ..." -kommando för att begära data.

Argumenten skickas till hubben i turordning.

**Obs:** Du kan använda kommandot **Send** i ett användardefinierat program, men inte i en funktion.

**Obs:** Se även **Get** (på sidan 81), **GetStr** (på sidan 88) och **eval()** (på sidan 64).

|                        |          |
|------------------------|----------|
| Send "READ BRIGHTNESS" | Done     |
| Get <i>lightval</i>    | Done     |
| <i>lightval</i>        | 0.347922 |

Exempel: Skicka en beräknad frekvens till hubbens inbyggda högtalare. Använd den speciella variabeln *iostr.SendAns* för att visa hubb-kommandot med det beräknade uttrycket.

|                                              |                 |
|----------------------------------------------|-----------------|
| <i>n</i> :=50                                | 50              |
| <i>m</i> :=4                                 | 4               |
| Send "SET SOUND eval( <i>m</i> · <i>n</i> )" | Done            |
| <i>iostr.SendAns</i>                         | "SET SOUND 200" |

## seq()

## Katalog >

**seq**(*Expr*, *Var*, *Low*, *High*, [*Step*]) ⇒ lista

Ökar *Var* från *Low* till *High* i steg om *Step*, utvärderar *Expr* och ger resultaten som en lista. De ursprungliga innehållen i *Var* är fortfarande där när **seq()** har beräknats.

Det förinställda värdet på *Step* = 1.

|                                                                            |                                                                        |
|----------------------------------------------------------------------------|------------------------------------------------------------------------|
| $\text{seq}\left(n^2, n, 1, 6\right)$                                      | $\{1, 4, 9, 16, 25, 36\}$                                              |
| $\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$                          | $\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$ |
| $\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$ | $\frac{1968329}{1270080}$                                              |

**Obs:** För att få ett närmevärde,

**Handenhet:** Tryck på  .

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

|                                                                            |         |
|----------------------------------------------------------------------------|---------|
| $\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$ | 1.54977 |
|----------------------------------------------------------------------------|---------|

## seqGen()

## Katalog >

**seqGen**(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, [*ListOfInitTerms* [, *VarStep* [, *CeilingValue*]]]) ⇒ lista

Genererar de första 5 termerna i talföljden *u* (*n*) = *u*(*n*-1)<sup>2</sup>/2, med *u*(1)=2 och *VarStep*=1.

Genererar en lista på termer för talföljden  $depVar(Var)=Expr$  enligt följande: Ökar den oberoende variabeln  $Var$  från  $Var0$  till  $VarMax$  i steg om  $VarStep$ , utvärderar  $depVar(Var)$  för motsvarande värden på  $Var$  med formeln  $Expr$  och  $ListOfInitTerms$  och ger resultaten som en lista.

**seqGen**(*ListOrSystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*} [, *MatrixOfInitTerms* [, *VarStep* [, *CeilingValue*]]])  $\Rightarrow$  *matrix*

Genererar en matris med termer för ett system (eller en lista) av talföljder *ListOfDepVars* (*Var*)=*ListOrSystemOfExpr* enligt följande: Ökar den oberoende variabeln  $Var$  från  $Var0$  till  $VarMax$  i steg om  $VarStep$ , utvärderar *ListOfDepVars* (*Var*) för motsvarande värden på  $Var$  med formeln *ListOrSystemOfExpr* och *MatrixOfInitTerms* och ger resultaten som en matris.

De ursprungliga innehållen i *Var* är oförändrade när **seqGen()** har beräknats.

Det förinställda värdet på *VarStep* = 1.

$$\text{seqGen}\left(\frac{u(n-1)^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ \left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$$

Exempel i vilket  $Var0=2$ :

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

Exempel i vilket den initiala termen är symbolisk:

$$\text{seqGen}\left(u(n-1)+2, n, u, \{1, 5\}, \{a\}\right) \\ \{a, a+2, a+4, a+6, a+8\}$$

System med två talföljder:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \left[\frac{-}{2}\right]\right) \\ \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Obs: Tomrummet ( ) i den initiala termmatrisen ovan används för att indikera att den initiala termen för  $u_1(n)$  beräknas med den explicita talföljdsformeln  $u_1(n)=1/n$ .

**seqn**(*Expr*{*u*, *n* [, *ListOfInitTerms* [, *nMax* [, *CeilingValue*]]])  $\Rightarrow$  *lista*

Genererar de första 6 termerna i talföljden  $u(n) = u(n-1)/2$ , med  $u(1)=2$ .

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right) \\ \left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

## seqn()

Katalog > 

Genererar en lista på termer för en talföljd,  $u(n)=Expr(u, n)$ , enligt följande: Ökar  $n$  från 1 till  $nMax$  i steg om 1, utvärderar  $u(n)$  för motsvarande värden på  $n$  med formeln  $Expr(u, n)$  och *ListOfInitTerms* och ger resultaten som en lista.

**seqn**(*Expr*( $n$  [,  $nMax$  [, *CeilingValue*]]) $\Rightarrow$ lista

Genererar en lista på termer för en icke-rekursiv talföljd,  $u(n)=Expr(n)$ , enligt följande: Ökar  $n$  från 1 till  $nMax$  i steg om 1, utvärderar  $u(n)$  för motsvarande värden på  $n$  med formeln  $Expr(n)$  och ger resultaten som en lista.

Om  $nMax$  saknas ställs  $nMax$  in på 2500

Om  $nMax=0$  ställs  $nMax$  in på 2500

**Obs:** **seqn()** anropar **seqGen()** med  $n0=1$  och  $nstep=1$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right) \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

## series()

Katalog > 

**series**(*Expr1*, *Var*, *Order* [, *Point*]) $\Rightarrow$ uttryck

**series**(*Expr1*, *Var*, *Order* [, *Point*]) | *Var*>*Point* $\Rightarrow$ uttryck

**series**(*Expr1*, *Var*, *Order* [, *Point*]) | *Var*<*Point* $\Rightarrow$ uttryck

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} - \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-1}}, z, -1\right) \quad z-1$$

$$\text{series}\left(\left(1+\frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} + \frac{29 \cdot x^7}{720}$$

$$\text{series}\left((1+e^x)^2, x, 2, 1\right) \quad (e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

Ger en generaliserad trunkerad potensserierepresentation av *Expr1* utvecklat kring *Point* genom *Order-graden* *Order* kan vara ett rationellt tal vilket som helst. De resulterande potenserna av  $(Var - Point)$  kan inkludera negativa och/eller exponenter i bråkform. Koefficienterna för dessa potenser kan inkludera logaritmer av  $(Var - Point)$  och andra funktioner av *Var* som domineras av alla potenser av  $(Var - Point)$  som har samma exponenttecken.

*Point* förinställs till 0. *Point* kan vara  $\infty$  eller  $-\infty$ , varvid utvecklingen sker genom *Order-graden* i  $1/(Var - Point)$ .

**series(...)** ger "**series(...)**" om den inte kan bestämma en sådan representation, till exempel, för väsentliga singulariteter såsom  $\sin(1/z)$  vid  $z=0$ ,  $e^{-1/z}$  vid  $z=0$ , eller  $e^z$  vid  $z = \infty$  eller  $-\infty$ .

Om serien eller någon av dess derivator har en språngdiskontinuitet vid *Point* innehåller resultatet troligen deluttryck i formen  $\text{sign}(\dots)$  eller  $\text{abs}(\dots)$  för en reell expansionsvariabel, eller  $(-1)^{\text{floor}(\dots \text{angle}(\dots))}$  för en komplex expansionsvariabel, vilken slutar med "\_". Om du tänker använda serien endast för värden på ena sidan av *Point*, lägg då till den lämpliga av " $| Var > Point$ ", " $| Var < Point$ ", " $| Var \geq Point$ " eller " $Var \leq Point$ " för att erhålla ett enklare resultat.

**series()** kan ge symboliska approximationer av obestämda och bestämda integraler, för vilka symboliska lösningar annars inte kan erhållas.

**series()** fördelas över 1:a-argumentlistor och matriser.

**series()** är en generaliserad version av **taylor()**.

Som det sista exemplet till höger illustrerar kan visningsrutinerna nedåt i resultatet som produceras av series(...) arrangera om termer så att den dominanta termen inte är längst till vänster.

**Obs:** Se även `dominantTerm()`, på sidan 57.

## setMode()

**setMode(modeNameInteger, settingInteger)  $\Rightarrow$  heltal**

**setMode(list)  $\Rightarrow$  lista på heltal**

Endast giltigt inom en funktion eller ett program.

**setMode(modeNameInteger, settingInteger)** ställer temporärt in läget *modeNameInteger* på den nya inställningen *settingInteger* och ger ett heltal som motsvarar den ursprungliga inställningen på det läget. Ändringen är begränsad till den tid det tar att exekvera programmet/funktionen.

*modeNameInteger* specificerar vilket läge du vill ställa in. Det måste vara något av de lägesheltal som anges i nedanstående tabell.

*settingInteger* specificerar den nya inställningen för läget. Det måste vara något av de inställningsheltal som anges i nedanstående tabell för det läge som du ställer in.

**setMode(list)** låter dig ändra flera inställningar. *list* innehåller par av lägesheltal och inställningsheltal..

**setMode(list)** ger en liknande lista vars heltalspar representerar de ursprungliga lägena och inställningarna.

Visa ungefärligt värde på  $\pi$  med förinställningen för Display Digits (Visa siffror) och visa sedan  $\pi$  med inställningen Fix2. Kontrollera sedan att förinställningen har återställts efter exekveringen av programmet.

|                              |         |
|------------------------------|---------|
| Define <i>progI()</i> = Prgm | Done    |
| Disp approx( $\pi$ )         |         |
| setMode(1,16)                |         |
| Disp approx( $\pi$ )         |         |
| EndPrgm                      |         |
| <i>progI()</i>               |         |
|                              | 3.14159 |
|                              | 3.14    |
|                              | Done    |

Om du har sparat alla lägesinställningar med **getMode(0)** → *var* kan du använda **setMode(*var*)** för att återställa dessa inställningar tills funktionen eller programmet avslutas. Se **getMode()**, på sidan 87.

**Obs:** De aktuella lägesinställningarna förs vidare till anropade delrutiner. Om någon delrutin ändrar en lägesinställning förloras lägesändringen när kontrollen återgår till den anropande delrutinen.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

| Lägets namn                                        | Läges-heltal | Heltal för inställningar                                                                                                                                                                                                                                       |
|----------------------------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Display Digits<br>(Visa siffror)                   | 1            | 1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12 |
| Angle (Vinkel)                                     | 2            | 1=Radian, 2=Degree, 3=Gradian                                                                                                                                                                                                                                  |
| Exponential<br>Format<br>(Exponentiellt<br>format) | 3            | 1=Normal, 2=Scientific, 3=Engineering                                                                                                                                                                                                                          |
| Real or Complex<br>(Reellt eller<br>Komplext)      | 4            | 1=Real, 2=Rectangular, 3=Polar                                                                                                                                                                                                                                 |
| Auto or Approx.<br>(Auto eller<br>Ungefärlig)      | 5            | 1=Auto, 2=Approximate, 3=Exact                                                                                                                                                                                                                                 |
| Vector Format<br>(Vektorformat)                    | 6            | 1=Rectangular, 2=Cylindrical, 3=Spherical                                                                                                                                                                                                                      |
| Base (Bas)                                         | 7            | 1=Decimal, 2=Hex, 3=Binary                                                                                                                                                                                                                                     |
| Unit System<br>(Enhetssystem)                      | 8            | 1=SI, 2=Eng/US                                                                                                                                                                                                                                                 |

**shift(IntegerI [, #ofShifts])** ⇒ *heltal*

Skiftar bitarna i ett binärt heltal. Du kan skriva in *IntegerI* i valfri talbas. Det omvandlas automatiskt till 64-bitars binär form. Om storleken på *IntegerI* är alltför stor för denna form för en symmetrisk modulooperation talet inom området. För mer information, se ►**Base2**, på sidan 18.

Om *#ofShifts* är positiv sker skiftningen åt vänster. Om *#ofShifts* är negativ sker skiftningen åt höger. Förinställningen är -1 (skifta en bit åt höger).

Vid en skiftning åt höger "droppas" biten längst till höger och 1 infogas som denna bit. Vid skiftning åt vänster "droppas" biten längst till vänster och 0 infogas som denna bit.

Vid exempelvis skiftning åt höger:

Varje bit skiftas åt höger.

0b0000000000000111101011000011010

Infogar 0 om biten längst till vänster är 0 eller 1 om denna bit är 1.

ger:

0b00000000000000111101011000011010

Resultatet visas enligt det inställda basläget. Inledande nollor visas inte.

**shift(ListI [, #ofShifts])** ⇒ *lista*

Ger en kopia av *ListI* skiftad åt höger eller vänster av *#ofShifts*-elementen. Ändrar inte *ListI*.

Om *#ofShifts* är positiv sker skiftningen åt vänster. Om *#ofShifts* är negativ sker skiftningen åt höger. Förinställningen är -1 (skifta ett element åt höger).

Element som infogas i början eller slutet av *list* av skiftningen tilldelas symbolen "undef".

I binärt basläge:

|                              |                      |
|------------------------------|----------------------|
| shift(0b1111010110000110101) | 0b111101011000011010 |
| shift(256,1)                 | 0b1000000000         |

I hexadecimalt basläge:

|                 |        |
|-----------------|--------|
| shift(0h78E)    | 0h3C7  |
| shift(0h78E,-2) | 0h1E3  |
| shift(0h78E,2)  | 0h1E38 |

**Viktigt:** För att skriva in ett binärt eller hexadecimalt tal, använd alltid prefixet 0b eller 0h (noll, inte bokstaven O).

I decimalt basläge:

|                     |                   |
|---------------------|-------------------|
| shift({1,2,3,4})    | {undef,1,2,3}     |
| shift({1,2,3,4},-2) | {undef,undef,1,2} |
| shift({1,2,3,4},2)  | {3,4,undef,undef} |

**shift()**Katalog > **shift(String1 [,#ofShifts])**⇒*sträng*

Ger en kopia av *String1* skiftad åt höger eller vänster av *#ofShifts*-tecknen. Ändrar inte *String1*.

Om *#ofShifts* är positiv sker skiftningen åt vänster. Om *#ofShifts* är negativ sker skiftningen åt höger. Förinställningen är -1 (skifta ett tecken åt höger).

Tecken som infogas i början eller slutet av *string* av skiftningen får ett mellanslag.

|                               |         |
|-------------------------------|---------|
| <code>shift("abcd")</code>    | " abc " |
| <code>shift("abcd",-2)</code> | " ab "  |
| <code>shift("abcd",1)</code>  | "bcd "  |

**sign()**Katalog > **sign(Expr1)**⇒*uttryck***sign(List1)**⇒*lista***sign(Matrix1)**⇒*matrix*

Ger, för ett reellt eller komplext *Expr1*, *Expr1/abs(Expr1)* när *Expr1* ≠ 0.

Ger 1 om *Expr1* är positivt.

Ger -1 om *Expr1* är negativt.

**sign(0)** ger ±1 om det komplexa formatläget är Real, annars ger det sig självt.

**sign(0)** representerar enhetscirkeln i det komplexa området.

Ger, för en lista eller matrix, tecknen för alla element.

|                               |            |
|-------------------------------|------------|
| <code>sign(-3.2)</code>       | -1.        |
| <code>sign({2,3,4,-5})</code> | {1,1,1,-1} |
| <code>sign(1+ x )</code>      | 1          |

Om det komplexa formatläget är Real:

|                             |                                                |
|-----------------------------|------------------------------------------------|
| <code>sign([-3 0 3])</code> | $\begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$ |
|-----------------------------|------------------------------------------------|

**simult()**Katalog > **simult(coeffMatrix, constVector[, Tol])**⇒*matrix*

Ger en kolumnvektor som innehåller lösningarna för ett system av linjära ekvationer.

Obs: Se även **linSolve()**, på sidan 107.

Lös för x och y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

|                                                                                                                               |                                         |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <code>simult(<math>\begin{pmatrix} 1 &amp; 2 \\ 3 &amp; 4 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}</math>)</code> | $\begin{bmatrix} -3 \\ 2 \end{bmatrix}$ |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|

Lösningen är x=-3 och y=2.



*coeffMatrix* måste vara en kvadratisk som innehåller koefficienterna för ekvationerna.

*constVector* måste ha samma antal rader (samma dimension) som *coeffMatrix* och innehålla konstanterna.

Alternativt behandlas varje matriselement som noll om dess absolutvärde är mindre än *Tol*. Denna tolerans används endast om matrisen har inmatning i flyttalsform och inte innehåller några symboliska variabler som inte har tilldelats ett värde. Annars ignoreras *Tol*.

- Om du ställer in läget **Auto** eller **Ungefärlig** på Approximate utförs beräkningarna med flyttalsaritmetik.
- Om *Tol* utelämnas eller inte används beräknas standardtoleransen som:  
 $5E-14 \cdot \max(\dim(\text{coeffMatrix})) \cdot \text{rowNorm}(\text{coeffMatrix})$

**simult(coeffMatrix, constMatrix[, Tol])** ⇒ *matris*

Löser multipla system av linjära ekvationer där varje system har samma koefficienter för variablerna, men olika konstanter.

Varje kolumn i *constMatrix* måste innehålla konstanterna för ett ekvationssystem. Varje kolumn i den resulterande matrisen innehåller lösningen på motsvarande system.

Lös:

$$ax + by = 1$$

$$cx + dy = 2$$

|                                                                                |                                                                                                                     |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matx1}$        | $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$                                                                      |
| $\text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$ | $\begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$ |

Lös:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

|                                                                                                                          |                                                            |
|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| $\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right)$ | $\begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$ |
|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|

För det första systemet är lösningen  $x=-3$  och  $y=2$ . För det andra systemet är lösningen  $x=-7$  och  $y=9/2$ .

*Expr* ▶ **sin**

|                                        |                   |
|----------------------------------------|-------------------|
| $(\cos(x))^2 \blacktriangleright \sin$ | $1 - (\sin(x))^2$ |
|----------------------------------------|-------------------|

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva `@>sin.`

Representerar *Expr* i termer av sinus. Detta är en omvandlingsoperator för visning. Den kan endast användas i slutet av inmatningsraden.

**sin** reducerar alla potenser av  $\cos(\dots)$  modulo  $1 - \sin(\dots)^2$  så att eventuella återstående potenser av  $\sin(\dots)$  har exponenter i området  $(0, 2)$ . Resultatet kommer sålunda att vara fritt från  $\cos(\dots)$  om, och endast om,  $\cos(\dots)$  endast inträffar i det givna uttrycket vid jämna potenser.

**Obs:** Denna omvandlingsoperator stöds inte i vinkellägena Grader och Nygrader. Innan du använder den, kontrollera att vinkelläget är inställt på Radianer och att *Expr* inte innehåller explicita referenser till vinklar i grader eller nygrader.

## sin()

## tangent

**sin(Expr1) ⇒ uttryck**

I vinkelläget Grader:

**sin(List1) ⇒ lista**

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

**sin(Expr1)** Ger sinus för argumentet som ett uttryck.

$$\sin(45) = \frac{\sqrt{2}}{2}$$

**sin(List1)** ger en lista på sinus för alla element i *List1*.

$$\sin(\{0, 60, 90\}) = \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

**Obs:** Argumentet tolkas som en vinkel i antingen grader, nygrader eller radianer beroende på det aktuella vinkelläget. Du kan använda  $^\circ$ ,  $^g$  eller  $r$  för att tillfälligt överstyra vinkelläget.

I vinkelläget Nygrader:

$$\sin(50) = \frac{\sqrt{2}}{2}$$

I vinkelläget Radianer:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

**sin(squareMatrix1)⇒kvadratMatris**

Ger matrisen med sinus för *squareMatrix1*. Detta är inte detsamma som att beräkna sinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I vinkelläget Radianer:

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

**sin<sup>-1</sup>(Expr1)⇒uttryck**

**sin<sup>-1</sup>(List1)⇒lista**

**sin<sup>-1</sup>(Expr1)** ger den vinkel vars sinus är *Expr1* som ett uttryck.

**sin<sup>-1</sup>(List1)** ger en lista på invers sinus för varje element i *List1*.

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arcsin(...)**.

**sin<sup>-1</sup>(squareMatrix1)⇒kvadratMatris**

Ger matrisen med invers sinus för *squareMatrix1*. Detta är inte detsamma som att beräkna invers sinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I vinkelläget Grader:

$$\sin^{-1}(1) = 90$$

I vinkelläget Nygrader:

$$\sin^{-1}(1) = 100$$

I vinkelläget Radianer:

$$\sin^{-1}(\{0,0.2,0.5\}) = \{0,0.201358,0.523599\}$$

I vinkelläget Radianer och i Rektangulärt komplext format:

$$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right) = \begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$$

**sinh()**Katalog > **sinh(*Expr1*)**⇒*uttryck*

sinh(1.2) 1.50946

**sinh(*List1*)**⇒*lista*

sinh({0,1.2,3.}) {0,1.50946,10.0179}

**sinh (*Expr1*)** ger argumentets hyperboliska sinus som ett uttryck.

**sinh (*List1*)** ger en lista på hyperboliska sinus för varje element i *List1*.

**sinh(*squareMatrix1*)**⇒*kvadratMatris*

Ger matrisen med hyperbolisk sinus för *squareMatrix1*. Detta är inte detsamma som att beräkna hyperbolisk sinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I vinkelläget Radianer:

|      |                                                                      |                                                                                                                           |
|------|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| sinh | $\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$ |                                                                                                                           |
|      |                                                                      | $\begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$ |
|      |                                                                      |                                                                                                                           |
|      |                                                                      |                                                                                                                           |

**sinh<sup>-1</sup>()**Katalog > **sinh<sup>-1</sup>(*Expr1*)**⇒*uttryck*sinh<sup>-1</sup>(0) 0**sinh<sup>-1</sup>(*List1*)**⇒*lista*sinh<sup>-1</sup>({0,2.1,3}) {0,1.48748,sinh<sup>-1</sup>(3)}

**sinh<sup>-1</sup>(*Expr1*)** ger argumentets inversa hyperboliska sinus som ett uttryck.

**sinh<sup>-1</sup>(*List1*)** ger en lista på invers hyperbolisk sinus för varje element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arcsinh (...)**.

**sinh<sup>-1</sup>(*squareMatrix1*)**⇒*kvadratMatris*

Ger matrisen med invers hyperbolisk sinus för *squareMatrix1*. Detta är inte detsamma som att beräkna invers hyperbolisk sinus för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet innehåller alltid tal med flytande komma.

I vinkelläget Radianer:

|                    |                                                                      |                                                                                                                             |
|--------------------|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| sinh <sup>-1</sup> | $\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$ |                                                                                                                             |
|                    |                                                                      | $\begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$ |
|                    |                                                                      |                                                                                                                             |
|                    |                                                                      |                                                                                                                             |

**SinReg**  $X, Y$  [, [*Iterations*],[ *Period*] [, *Category*, *Include*] ]

Utför en trigonometrisk regressionsanalys på listorna  $X$  och  $Y$ . En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

*Iterations* är ett värde som specificerar det maximala antalet gånger (1 till och med 16) en lösning kommer att provas. Om denna utelämnas används 8. Normalt ger större värden bättre noggrannhet, men längre exekveringstider, och vice versa.

*Period* specificerar en uppskattad period. Om denna utelämnas bör skillnaden mellan värdena i  $X$  vara lika och i ordningsföljd. Om du specificerar *Period* kan skillnaderna mellan  $x$ -värden vara olika.

*Category* är en lista på kategorikoder för motsvarande  $X$ - och  $Y$ -data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

Resultatet av **SinReg** är alltid i radianer oavsett det inställda vinkelläget.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel               | Beskrivning                                 |
|--------------------------------|---------------------------------------------|
| stat.RegEqn                    | Regressionsekvation: $a \cdot \sin(bx+c)+d$ |
| stat.a, stat.b, stat.c, stat.d | Regressionskoefficienter                    |
| stat.Resid                     | Residualer från regressionen                |

| Resultatvariabel | Beskrivning                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stat.XReg        | Lista på datapunkter i den modifierade <i>X List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.YReg        | Lista på datapunkter i den modifierade <i>Y List</i> som används i regressionen baserat på begränsningar i <i>Freq</i> , <i>Category List</i> och <i>Include Categories</i> |
| stat.FreqReg     | Lista på frekvenser som motsvarar <i>stat.XReg</i> och <i>stat.YReg</i>                                                                                                     |

## solve()

Katalog > 

**solve(Equation, Var)⇒Booleskt uttryck**

**solve(Equation, Var=Guess)⇒Booleskt uttryck**

**solve(Inequality, Var)⇒Booleskt uttryck**

Ger möjliga reella lösningar på en ekvation eller olikhet för *Var*. Målet är att ge "kandidater" för alla lösningar. Det kan dock finnas ekvationer eller lösningar för vilka antalet lösningar är oändligt.

Möjliga lösningar kanske inte är reella ändliga lösningar för vissa kombinationer av värden för odefinierade variabler.

Med inställningen Auto i läge **Auto eller Ungefärlig** är målet att producera exakta lösningar när de är koncisa, och kompletterade med iterativa sökningar med ungefärlig aritmetik när exakta lösningar är opraktiska.

På grund av den förinställda annulleringen av den största gemensamma delaren från täljaren och nämnaren i kvoter kan lösningar vara lösningar endast som gränsvärden från en sida eller båda sidorna.

För olikheter av typ  $\geq$ ,  $\leq$ , eller  $>$  är explicita lösningar osannolika såvida inte olikheten är linjär och endast innehåller *Var*.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

$$\text{Ans}|a=1 \text{ and } b=1 \text{ and } c=1$$

$$x = \frac{-1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

$$\text{solve}((x-a) \cdot e^x = -x \cdot (x-a), x)$$

$$x=a \text{ or } x=0.567143$$

$$(x+1) \cdot \frac{x-1}{x-1} + x-3$$

$$2 \cdot x-2$$

$$\text{solve}(5 \cdot x-2 \geq 2 \cdot x, x)$$

$$x \geq \frac{2}{3}$$

I läget Exact erhålls delar som inte kan lösas som en implicit ekvation eller olikhet.

Använd ("|")-operatoren begränsning för att begränsa lösningsintervallet och/eller andra variabler som ingår i ekvationen eller olikheten När du finner en lösning i ett intervall kan du använda olikhetsoperatorerna för att utesluta det intervallet från efterföljande sökningar.

ger resultatet falskt när inga reella lösningar hittas. ger resultatet sant om **solve()** kan fastställa att varje ändligt reellt värde på *Var* satisfierar ekvationen eller olikheten.

Eftersom **solve()** alltid ger ett booleskt resultat kan du använda "and", "or" och "inte" för att kombinera resultat från **solve()** med varandra eller med andra booleska uttryck.

Lösningar kan innehålla en ny, unik odefinierad konstant med formen *nj* där *j* är ett heltal i intervallet 1–255. Sådana variabler betecknar ett godtyckligt heltal.

I läge Real betecknar bråktalspotenser med udda nämnare endast den rella delen. Annars betecknar flera delade uttryck såsom rationella potenser, logaritmer och inversa trigonometriska funktioner endast huvudintervallet. Följaktligen ger **solve()** endast lösningar som motsvarar den reella delen eller huvudintervallet.

**Obs:** Se även **cSolve()**, **cZeros()**, **nSolve()** och **zeros()**.

**solve(Eqn1 and Eqn2 [and...], VarOrGuess1, VarOrGuess2 [, ...])** ⇒ Booleskt uttryck

**solve(SystemOfEqns, VarOrGuess1, VarOrGuess2 [, ...])** ⇒ Booleskt uttryck

**solve({Eqn1, Eqn2 [,...]} {VarOrGuess1,**

$$\text{exact}\left(\text{solve}\left((x-a) \cdot e^x = x \cdot (x-a), x\right)\right) \\ e^x + x = 0 \text{ or } x = a$$

I vinkelläget Radianer:

$$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) | x > 0 \text{ and } x < 1 \\ x = 0.860334$$

$$\text{solve}(x = x + 1, x) \quad \text{false} \\ \text{solve}(x = x, x) \quad \text{true}$$

$$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x) \quad x \neq -3 \text{ and } x \leq 1$$

I vinkelläget Radianer:

$$\text{solve}(\sin(x) = 0, x) \quad x = n1 \cdot \pi$$

$$\text{solve}\left(x^{\frac{1}{3}} = 1, x\right) \quad x = 1 \\ \text{solve}(\sqrt{x} = 2, x) \quad \text{false} \\ \text{solve}(-\sqrt{x} = 2, x) \quad x = 4$$

$$\text{solve}(y = x^2 - 2 \text{ and } x + 2 \cdot y = 1, \{x, y\}) \\ x = \frac{-3}{2} \text{ and } y = \frac{1}{4} \text{ or } x = 1 \text{ and } y = -1$$

*VarOrGuess2 [ , ... ]})  $\Rightarrow$  Booleskt uttryck*

Ger möjliga reella lösningar på ekvationssystem där varje *VarOrGuess* specificerar en variabel som du vill lösa ut.

Du kan separera ekvationerna med operatoren **and** (och) eller också kan du mata in ett *SystemOfEqns* (Ekvationssystem) med en mall från Katalogen. Antalet *VarOrGuess*-argument måste vara lika med antalet ekvationer. Du kan som alternativ specificera en initial gissning för en variabel. Varje *VarOrGuess* måste ha formen:

*variable*

– eller –

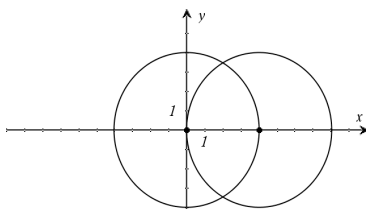
*variable = reellt eller icke-reellt tal*

Som exempel är  $x$  giltigt och likaså  $x=3$ .

Om alla ekvationer är polynom och om du INTE specificerar några initiala gissningar använder **solve()** eliminationsmetoden Gröbner/Buchberger för att försöka bestämma alla reella lösningar.

Lås oss som exempel anta att du har en cirkel med radien  $r$  i origo och en annan cirkel med radien  $r$  där centrum är där den första cirkeln korsar den positiva  $x$ -axeln. Använd **solve()** för att finna skärningspunkterna.

Såsom illustreras med  $r$  i exemplet till höger kan simultana polynomekvationer ha extra variabler som saknar värden, men representerar givna numeriska värden som kan ersättas senare.



$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2}$$



Du kan även (eller i stället) inkludera Lösningsvariabler som inte visas i ekvationerna. Du kan exempelvis inkludera  $z$  som en Lösningsvariabel för att utöka föregående exempel till två parallella överkorsande cylindrar med radien  $r$ .

Cylinderlösningarna illustrerar hur familjer av lösningar kan innehålla godtyckliga konstanter med formen  $ck$  där  $k$  är ett heltalssuffix från 1 till och med 255.

För polynomsystem kan beräkningstiden och användningen av minne i hög grad bero på i vilken ordning du listar Lösningsvariabler. Om ditt första val utarmar minnet, eller tar på ditt tålamod, kan du försöka med att arrangera om variablerna i ekvationerna och/eller i listan *VarOrGuess*.

Om du inte inkluderar några gissningar och om någon ekvation är ett icke-polynom i någon variabel, men alla ekvationer är linjära i Lösningsvariablerna, använder **solve()** Gauss eliminationsmetod för att försöka bestämma alla reella lösningar.

Om ett system är varken polynomt i alla dess variabler eller linjärt i dess Lösningsvariabler bestämmer **solve()** högst en lösning med en ungefärlig iterativ metod. För att göra detta måste antalet Lösningsvariabler vara lika med antalet ekvationer och alla övriga variabler i ekvationerna måste förenklas till tal.

Varje Lösningsvariabel startar vid dess gissade värde om det finns något, annars startar den vid 0.0.

Använd gissningar för att söka ytterligare lösningar en i taget. För konvergens kan en gissning behöva vara ganska nära en lösning.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y\rightarrow$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

$$\text{solve}\left(x+e^z\cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

$$x=\frac{e^z\cdot\sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^z+1}$$

$$\text{solve}\left(e^z\cdot y=1 \text{ and } \neg y=\sin(z), \{y,z\}\right)$$

$$y=2.812\text{E-}10 \text{ and } z=21.9911 \text{ or } y=0.001871\text{P}$$

För att se hela resultatet, tryck på **▲** och använd sedan **◀** och **▶** för att flytta markören.

$$\text{solve}\left(e^z\cdot y=1 \text{ and } \neg y=\sin(z), \{y,z=2\cdot\pi\}\right)$$

$$y=0.001871 \text{ and } z=6.28131$$

SortA

Katalog >

|                                                                                                                                                                          |                                 |               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|---------------|
| <b>SortA</b> <i>List1</i> [, <i>List2</i> ] [, <i>List3</i> ] ...                                                                                                        | $\{2,1,4,3\} \rightarrow list1$ | $\{2,1,4,3\}$ |
| <b>SortA</b> <i>Vector1</i> [, <i>Vector2</i> ] [, <i>Vector3</i> ] ...                                                                                                  | SortA <i>list1</i>              | Done          |
|                                                                                                                                                                          | <i>list1</i>                    | $\{1,2,3,4\}$ |
| Sorterar elementen i det första argumentet i stigande ordning.                                                                                                           | $\{4,3,2,1\} \rightarrow list2$ | $\{4,3,2,1\}$ |
|                                                                                                                                                                          | SortA <i>list2,list1</i>        | Done          |
| Om du inkluderar ytterligare argument sorteras elementen i varje argument så att deras nya positioner matchar de nya positionerna för elementen i det första argumentet. | <i>list2</i>                    | $\{1,2,3,4\}$ |
|                                                                                                                                                                          | <i>list1</i>                    | $\{4,3,2,1\}$ |

Alla argument måste vara namn på listor eller vektorer. Alla argument måste ha samma dimensioner.

Tomma element inom det första argumentet flyttas till botten. För mer information om tomma element, se på sidan 261.

SortD

Katalog >

|                                                                                                    |                                 |               |
|----------------------------------------------------------------------------------------------------|---------------------------------|---------------|
| <b>SortD</b> <i>List1</i> [, <i>List2</i> ] [, <i>List3</i> ] ...                                  | $\{2,1,4,3\} \rightarrow list1$ | $\{2,1,4,3\}$ |
| <b>SortD</b> <i>Vector1</i> [, <i>Vector2</i> ] [, <i>Vector3</i> ] ...                            | $\{1,2,3,4\} \rightarrow list2$ | $\{1,2,3,4\}$ |
|                                                                                                    | SortD <i>list1,list2</i>        | Done          |
| Identisk med <b>SortA</b> med undantag för att <b>SortD</b> sorterar elementen i fallande ordning. | <i>list1</i>                    | $\{4,3,2,1\}$ |
|                                                                                                    | <i>list2</i>                    | $\{3,4,1,2\}$ |

Tomma element inom det första argumentet flyttas till botten. För mer information om tomma element, se på sidan 261.

►Sphere

Katalog >

*Vector* ►Sphere

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @>Sphere.

Visar rad- eller kolumnvektorn i sfärisk form [ρ ∠θ ∠φ].

**Obs:** För att få ett närmevärde,

**Handenhet:** Tryck på .

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

|                                                                            |
|----------------------------------------------------------------------------|
| $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ ►Sphere                          |
| $\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$ |

*Vector* måste ha dimensionen 3 och kan vara antingen en rad- eller en kolumnvektor.

**Obs:** ►Sphere är en visa format-instruktion, inte en konverteringsfunktion. Du kan endast använda den i slutet av en inmatningsrad.

**Obs:** För att få ett närmevärde,

**Handenhet:** Tryck på  .

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

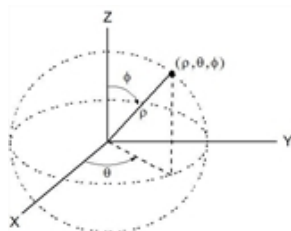
$$\left( \begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \right) \text{►Sphere}$$

$$\left[ 3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$

Tryck på 

$$\left( \begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \right) \text{►Sphere}$$

$$\left[ \sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \sin^{-1} \left( \frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



## sqrt()

**sqrt(Expr1)** ⇒ uttryck

$$\sqrt{4} \quad 2$$

**sqrt(List1)** ⇒ lista

$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\}$$

Ger kvadratroten ur argumentet.

Ger, för en lista, kvadratrötterna ur alla element i *List1*.

**Obs:** Se även **Square root template**, på sidan 1.

stat.results

Visar resultaten av en statistisk beräkning.

Resultaten visas som en uppsättning av namn-värdepar. De specifika namnen som visas beror på den/det senaste utvärderade statistiska funktionen/kommandot.

Du kan kopiera ett namn eller ett värde och klistra in det på andra platser.

**Obs:** Undvik att definiera variabler med samma namn som de variabler vilka används för statistisk analys. I vissa fall kan ett feltillstånd uppstå. Variabelnamn som används för statistisk analys listas i nedanstående tabell.

|                                 |                        |
|---------------------------------|------------------------|
| $xlist := \{1, 2, 3, 4, 5\}$    | $\{1, 2, 3, 4, 5\}$    |
| $ylist := \{4, 8, 11, 14, 17\}$ | $\{4, 8, 11, 14, 17\}$ |

LinRegMx  $xlist, ylist, 1$ : stat.results

|                   |                            |
|-------------------|----------------------------|
| "Title"           | "Linear Regression (mx+b)" |
| "RegEqn"          | "m*x+b"                    |
| "m"               | 3.2                        |
| "b"               | 1.2                        |
| "r <sup>2</sup> " | 0.996109                   |
| "r"               | 0.998053                   |
| "Resid"           | "{...}"                    |

|             |                              |
|-------------|------------------------------|
| stat.values | "Linear Regression (mx+b)"   |
|             | "m*x+b"                      |
|             | 3.2                          |
|             | 1.2                          |
|             | 0.996109                     |
|             | 0.998053                     |
|             | "{-0.4, 0.4, 0.2, 0., -0.2}" |

|                        |                 |                 |                      |                 |
|------------------------|-----------------|-----------------|----------------------|-----------------|
| stat.a                 | stat.dfDenom    | stat.MedianY    | stat.Q3X             | stat.SSBLOCK    |
| stat.AdjR <sup>2</sup> | stat.dfBlock    | stat.MEPred     | stat.Q3Y             | stat.SSCol      |
| stat.b                 | stat.dfCol      | stat.MinX       | stat.r               | stat.SSX        |
| stat.b0                | stat.dfError    | stat.MinY       | stat.r <sup>2</sup>  | stat.SSY        |
| stat.b1                | stat.dfInteract | stat.MS         | stat.RegEqn          | stat.SSError    |
| stat.b2                | stat.dfReg      | stat.MSBlock    | stat.Resid           | stat.SSInteract |
| stat.b3                | stat.dfNumer    | stat.MSCol      | stat.ResidTrans      | stat.SSReg      |
| stat.b4                | stat.dfRow      | stat.MSError    | stat.ox              | stat.SSRow      |
| stat.b5                | stat.DW         | stat.MSInteract | stat.oy              | stat.tList      |
| stat.b6                | stat.e          | stat.MSReg      | stat.ox1             | stat.UpperPred  |
| stat.b7                | stat.ExpMatrix  | stat.MSRow      | stat.ox2             | stat.UpperVal   |
| stat.b8                | stat.F          | stat.n          | stat.Σx              | stat.Σx̄        |
| stat.b9                | stat.FBlock     | stat.ŷ          | stat.Σx <sup>2</sup> | stat.Σx̄1       |
| stat.b10               | stat.Fcol       | stat.ŷ1         | stat.Σxy             | stat.Σx̄2       |
| stat.bList             | stat.FInteract  | stat.ŷ2         | stat.Σy              | stat.Σx̄Diff    |
| stat.χ <sup>2</sup>    | stat.FreqReg    | stat.ŷDiff      | stat.Σy <sup>2</sup> | stat.Σx̄List    |
| stat.c                 | stat.Frow       | stat.PList      | stat.s               | stat.XReg       |

|                 |                |                   |              |                      |
|-----------------|----------------|-------------------|--------------|----------------------|
| stat.CLower     | stat.Leverage  | stat.PVal         | stat.SE      | stat.XVal            |
| stat.CLowerList | stat.LowerPred | stat.PValBlock    | stat.SEList  | stat.XValList        |
| stat.CompList   | stat.LowerVal  | stat.PValCol      | stat.SEPred  | stat. $\bar{y}$      |
| stat.CompMatrix | stat.m         | stat.PValInteract | stat.sResid  | stat. $\hat{y}$      |
| stat.CookDist   | stat.MaxX      | stat.PValRow      | stat.SEslope | stat. $\hat{y}$ List |
| stat.CUpper     | stat.MaxY      | stat.Q1X          | stat.sp      | stat.YReg            |
| stat.CUpperList | stat.ME        | stat.Q1Y          | stat.SS      |                      |
| stat.d          | stat.MedianX   |                   |              |                      |

**Obs:** Varje gång applikationen Listor och kalkylblad beräknar statistiska resultat kopierar den "stat."-gruppvariabler till en "stat#."-grupp där # är ett tal som ökas automatiskt. Detta låter dig bibehålla tidigare resultat medan du utför flera beräkningar.

## stat.values

Katalog > 

stat.values

Se exemplet `stat.results`.

Visar en matris över värdena beräknade för den/det senast utvärderade statistiska funktionen/kommandot.

Till skillnad från `stat.results` utelämnar `stat.values` namnen som är associerade med värdena.

Du kan kopiera ett värde och klistra in det på andra platser.

## stDevPop()

Katalog > 

`stDevPop(List[, freqList])` ⇒ uttryck

I vinkeläget Radianer och i Auto-läge:

Ger populationens standardavvikelse för elementen i *List*.

$$\frac{\text{stDevPop}\left\{\left\{a, b, c\right\}\right\}}{\frac{\sqrt{2 \cdot\left(a^2-a \cdot(b+c)+b^2-b \cdot c+c^2\right)}}{3}} \\ \frac{\text{stDevPop}\left\{\left\{1,2,5,-6,3,-2\right\}\right\}}{\frac{\sqrt{465}}{6}}$$

Varje *freqList*-element räknar antalet förekomster av motsvarande element i *List*.

$$\frac{\text{stDevPop}\left\{\left\{1.3,2.5,-6.4\right\},\left\{3,2,5\right\}\right\}}{4.11107}$$

**Obs:** *List* måste innehålla minst två element. Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

**stDevPop()**Katalog > 

**stDevPop**(*Matrix1* [, *freqMatrix*]) ⇒ *matrix*

Ger en radvektor med populationens standardavvikelser för kolumnerna i *Matrix1*.

Varje *freqMatrix*-element räknar antalet förekomster av motsvarande element i *Matrix1*.

**Obs:** *Matrix1* måste innehålla minst två rader. Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

$$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{4 \cdot \sqrt{6}}{3} & \frac{\sqrt{78}}{3} & \frac{2 \cdot \sqrt{6}}{3} \end{bmatrix}$$

$$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) = \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix}$$

**stDevSamp()**Katalog > 

**stDevSamp**(*List* [, *freqList*]) ⇒ *uttryck*

Ger urvalets standardavvikelse för elementen i *List*.

Varje *freqList*-element räknar antalet förekomster av motsvarande element i *List*.

**Obs:** *List* måste innehålla minst två element. Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

**stDevSamp**(*Matrix1* [, *freqMatrix*]) ⇒ *matrix*

Ger en radvektor med urvalets standardavvikelser för kolumnerna i *Matrix1*.

Varje *freqMatrix*-element räknar antalet förekomster av motsvarande element i *Matrix1*.

**Obs:** *Matrix1* måste innehålla minst två rader. Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

$$\text{stDevSamp}(\{a, b, c\}) = \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3}$$

$$\text{stDevSamp}(\{1, 2, 5, -6, 3, -2\}) = \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}) = 4.33345$$

$$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) = \begin{bmatrix} 4 & \sqrt{13} & 2 \end{bmatrix}$$

$$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) = \begin{bmatrix} 2.7005 & 5.44695 \end{bmatrix}$$

| Stop                                                                                                                                                        | Katalog >  |      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------|
| <b>Stop</b>                                                                                                                                                 | $i:=0$                                                                                    | 0    |
| Programmeringskommando: Avslutar programmet.                                                                                                                | Define $progI()$ =Prgm                                                                    | Done |
|                                                                                                                                                             | For $i,1,10,1$                                                                            |      |
|                                                                                                                                                             | If $i=5$                                                                                  |      |
|                                                                                                                                                             | Stop                                                                                      |      |
|                                                                                                                                                             | EndFor                                                                                    |      |
|                                                                                                                                                             | EndPrgm                                                                                   |      |
| <b>Stop</b> är ej tillåtet i funktioner.                                                                                                                    | $progI()$                                                                                 | Done |
| <b>Obs för att mata in exemplet:</b> Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner. | $i$                                                                                       | 5    |

|              |                                                           |
|--------------|-----------------------------------------------------------|
| <b>Store</b> | <b>Se <math>\rightarrow</math> (store), på sidan 243.</b> |
|--------------|-----------------------------------------------------------|

| string()                                                      | Katalog >  |               |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------|
| <b>string(Expr)<math>\Rightarrow</math>sträng</b>             | $string(1.2345)$                                                                            | "1.2345"      |
| Förenklar <i>Expr</i> och ger resultatet som en teckensträng. | $string(1+2)$                                                                               | "3"           |
|                                                               | $string(\cos(x)+\sqrt{3})$                                                                  | "cos(x)+√(3)" |

| subMat()                                                                                                             | Katalog >  |                                                                     |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <b>subMat(Matrix1[, startRow] [, startCol] [, endRow] [, endCol]) <math>\Rightarrow</math>matris</b>                 | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$          | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ |
| Ger specificerad undermatris av <i>Matrix1</i> .                                                                     | $subMat(m1,2,1,3,2)$                                                                        | $\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$                      |
| Förvalsinställningar: <i>startRow</i> =1, <i>startCol</i> =1, <i>endRow</i> =sista rad, <i>endCol</i> =sista kolumn. | $subMat(m1,2,2)$                                                                            | $\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$                      |

|                    |                                                |
|--------------------|------------------------------------------------|
| <b>Sum (Sigma)</b> | <b>Se <math>\Sigma()</math>, på sidan 233.</b> |
|--------------------|------------------------------------------------|

**sum()**Katalog > **sum(List[, Start[, End]])** ⇒ uttryckGer summan av elementen i *List*.*Start* och *end* är valfria. De specificerar ett område med element.Varje tomt argument ger ett tomt resultat. Tomma element i *List* ignoreras. För mer information om tomma element, se på sidan 261.**sum(Matrix1[, Start[, End]])** ⇒ *matrix*Ger en radvektor som innehåller summorna av elementen i kolumnerna i *Matrix1*.*Start* och *end* är valfria. De specificerar ett område med rader.Varje tomt argument ger ett tomt resultat. Tomma element i *Matrix1* ignoreras. För mer information om tomma element, se på sidan 261.

|                                         |            |
|-----------------------------------------|------------|
| $\text{sum}\{\{1,2,3,4,5\}\}$           | 15         |
| $\text{sum}\{\{a,2\cdot a,3\cdot a\}\}$ | $6\cdot a$ |
| $\text{sum}\{\text{seq}(n,n,1,10)\}$    | 55         |
| $\text{sum}\{\{1,3,5,7,9\},3\}$         | 21         |

|                                                                                                |                                              |
|------------------------------------------------------------------------------------------------|----------------------------------------------|
| $\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$                  | $\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$    |
| $\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$     | $\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$ |
| $\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},2,3\right)$ | $\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$ |

**sumIf()**Katalog > **sumIf(List, Criteria[, SumList])** ⇒ värdeGer den totala kumulerade summan av alla element i *List* som uppfyller specificerade *Criteria*. Du kan också specificera en alternativ lista, *sumList*, för att ge elementen som skall kumuleras.*List* kan vara ett uttryck, en lista eller en *matrix*. *SumList*, om specificerad, måste ha samma dimension(er) som *List*.*Criteria* kan vara:

- Ett värde, ett uttryck eller en sträng. Som exempel ackumulerar **34** endast de element i *List* som förenklas till värdet 34.
- Ett booleskt uttryck som innehåller symbolen ? fungerar som platshållare för varje element. Som exempel ackumulerar **?<10** endast de element i

|                                                   |           |
|---------------------------------------------------|-----------|
| $\text{sumIf}(\{1,2,e,3,\pi,4,5,6\},2.5<?<4.5)$   | $e+\pi+7$ |
| $\text{sumIf}(\{1,2,3,4\},2<?<5,\{10,20,30,40\})$ | 70        |



*List* som är lägre än 10.

När ett *List*-element uppfyller *Criteria* adderas elementet till den ackumulerade summan. Om du inkluderar *sumList* adderas i stället motsvarande element från *sumList* till summan.

I applikationen Listor och kalkylblad kan du använda ett område av celler i stället för *List* och *sumList*.

Tomma element ignoreras. För mer information om tomma element, se på sidan 261.

**Obs:** Se även **countIf()**, på sidan 36.

**system**(*Eqn1* [, *Eqn2* [, *Eqn3* [, ...]]])

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

**system**(*Expr1* [, *Expr2* [, *Expr3* [, ...]]])

Ger ett ekvationssystem formaterat som en lista. Du kan också skapa ett system med en mall.

**Obs:** Se även **System of equations**, på sidan 3.

## T

*Matrix1* **T**  $\Rightarrow$  *matrix*

Ger den komplexkonjugerade transponeringen av *Matrix1*.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **@t**.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T \quad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T \quad \begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$$

**tan()**

**tan(*Expr1*)**⇒*uttryck*

I vinkelläget Grader:

**tan(*List1*)**⇒*lista*

|                                  |                               |
|----------------------------------|-------------------------------|
| $\tan\left(\frac{\pi}{4}\right)$ | 1                             |
| $\tan(45)$                       | 1                             |
| $\tan(\{0,60,90\})$              | $\{0,\sqrt{3},\text{undef}\}$ |

**tan(*Expr1*)** ger tangensen för argumentet som ett uttryck.

**tan(*List1*)** ger en lista på tangensen för alla element i *List1*.

**Obs:** Argumentet tolkas som en vinkel i antingen grader, nygrader eller radianer beroende på det aktuella vinkelläget. Du kan använda °, G eller <sup>r</sup> för att tillfälligt överstyra vinkelläget.

I vinkelläget Nygrader:

|                                  |                        |
|----------------------------------|------------------------|
| $\tan\left(\frac{\pi}{4}\right)$ | 1                      |
| $\tan(50)$                       | 1                      |
| $\tan(\{0,50,100\})$             | $\{0,1,\text{undef}\}$ |

I vinkelläget Radianer:

|                                                                          |                      |
|--------------------------------------------------------------------------|----------------------|
| $\tan\left(\frac{\pi}{4}\right)$                                         | 1                    |
| $\tan(45^\circ)$                                                         | 1                    |
| $\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right)$ | $\{0,\sqrt{3},0,1\}$ |

**tan(*squareMatrix1*)**⇒*kvadratMatrix*

Ger matrisen med tangensen för *squareMatrix1*. Detta är inte detsamma som att beräkna tangensen för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I vinkelläget Radianer:

|                                                                                       |                                                                                                                                |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| $\tan\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$ | $\begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$ |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|

**tan<sup>-1</sup>()**

**tan<sup>-1</sup>(*Expr1*)**⇒*uttryck*

I vinkelläget Grader:

**tan<sup>-1</sup>(*List1*)**⇒*lista*

|                |    |
|----------------|----|
| $\tan^{-1}(1)$ | 45 |
|----------------|----|

**tan<sup>-1</sup>(*Expr1*)** ger den vinkel vars tangens är *Expr1* som ett uttryck.

**tan<sup>-1</sup>(*List1*)** ger en lista på den inversa tangensen för varje element i *List1*.

I vinkelläget Nygrader:

## $\tan^{-1}()$

 **tangent**

**Obs:** Resultatet erhålls som en vinkel i grader, nygrader eller radianer beroende på det aktuella vinkelläget.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arctan (...)**.

**$\tan^{-1}(\text{squareMatrix1}) \Rightarrow \text{kvadratMatrix}$**

Ger matrisen med invers tangens för *squareMatrix1*. Detta är inte detsamma som att beräkna invers tangens för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

$\tan^{-1}(1)$  50

I vinkelläget Radianer:

$\tan^{-1}(\{0,0,2,0,5\})$   $\{0,0.197396,0.463648\}$

I vinkelläget Radianer:

$\tan^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$

|           |          |           |
|-----------|----------|-----------|
| -0.083658 | 1.26629  | 0.62263   |
| 0.748539  | 0.630015 | -0.070012 |
| 1.68608   | -1.18244 | 0.455126  |

## **tangentLine()**

**Katalog >** 

**$\text{tangentLine}(\text{Expr1}, \text{Var}, \text{Point}) \Rightarrow \text{uttryck}$**

**$\text{tangentLine}(\text{Expr1}, \text{Var}=\text{Point}) \Rightarrow \text{uttryck}$**

Ger tangenten för kurvan representerad av *Expr1* vid punkten som specificeras i *Var=Point*.

Kontrollera att den oberoende variabeln inte är definierad. Om exempelvis  $f1(x):=5$  och  $x:=3$  ger **tangentLine(f1(x),x,2)** "falskt".

$\text{tangentLine}(x^2, x, 1)$   $2 \cdot x - 1$

$\text{tangentLine}((x-3)^2 - 4, x=3)$  -4

$\text{tangentLine}\left(x^{\frac{1}{3}}, x=0\right)$   $x=0$

$\text{tangentLine}(\sqrt{x^2 - 4}, x=2)$  undef

$x:=3: \text{tangentLine}(x^2, x, 1)$  5

## **tanh()**

**Katalog >** 

**$\tanh(\text{Expr1}) \Rightarrow \text{uttryck}$**

**$\tanh(\text{List1}) \Rightarrow \text{lista}$**

**$\tanh(\text{Expr1})$**  ger den hyperboliska tangensen för argumentet som ett uttryck.

**$\tanh(\text{List1})$**  ger en lista på den hyperboliska tangensen för varje element i *List1*.

**$\tanh(\text{squareMatrix1}) \Rightarrow \text{kvadratMatrix}$**

$\tanh(1.2)$  0.833655

$\tanh(\{0,1\})$   $\{0, \tanh(1)\}$

I vinkelläget Radianer:

Ger matrisen med hyperbolisk tangens för *squareMatrix1*. Detta är inte detsamma som att beräkna hyperbolisk tangens för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

**tanh<sup>-1</sup>(Expr1)** ⇒ *uttryck*

**tanh<sup>-1</sup>(List1)** ⇒ *lista*

**tanh<sup>-1</sup>(Expr1)** ger argumentets inversa hyperboliska tangens som ett uttryck.

**tanh<sup>-1</sup>(List1)** ger en lista på invers hyperbolisk tangens för varje element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **arctanh (...)**.

**tanh<sup>-1</sup>(squareMatrix1)** ⇒ *kvadratMatris*

Ger matrisen med invers hyperbolisk tangens för *squareMatrix1*. Detta är inte detsamma som att beräkna invers hyperbolisk tangens för varje element. Se **cos()** för information om beräkningsmetoden.

*squareMatrix1* måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

I Rektangulärt komplext format:

$$\begin{aligned} \tanh^{-1}(0) &= 0 \\ \tanh^{-1}(\{1, 2, 3\}) &= \left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i \right\} \end{aligned}$$

I vinkelläget Radianer och i Rektangulärt komplext format:

$$\tanh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.099353 + 0.164058 \cdot i & 0.267834 - 1.4908 \\ -0.087596 - 0.725533 \cdot i & 0.479679 - 0.94730 \\ 0.511463 - 2.08316 \cdot i & -0.878563 + 1.7901 \end{bmatrix}$$

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

**taylor()**Katalog > **taylor**(*Expr1*, *Var*, *Order*[, *Point*]) $\Rightarrow$ uttryck

Ger det begärda Taylorpolynomet. Polynomet inkluderar icke-nolltermer av heltalsgrader från noll till och med *Order* i (*Var* minus *Point*). **taylor()** ger sig själv om det inte finns någon trunkerad potensserie av denna ordning, eller om det skulle kräva negativa exponenter eller exponenter i bråkform. Använd substitution och/eller temporär multiplikation med en potens av (*Var* minus *Point*) för att bestämma mer allmänna potensserier.

*Point* förinställs till noll och utgör expansionspunkten.

|                                                                                                  |                                                               |
|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| $\text{taylor}(e^{\sqrt{x}}, x, 2)$                                                              | $\text{taylor}(e^{\sqrt{x}}, x, 2, 0)$                        |
| $\text{taylor}(e^{t, t, 4}) _{t=\sqrt{x}}$                                                       | $\frac{x^2}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1$ |
| $\text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right)$                                        | $\text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right)$  |
| $\text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right)$ | $-x^3 - x^2 - x - \frac{1}{x} - 1$                            |

**tCdf()**Katalog > 

**tCdf**(*lowBound*, *upBound*, *df*) $\Rightarrow$ tal om *lowBound* och *upBound* är tal, lista om *lowBound* och *upBound* är listor

Beräknar sannolikheten för Student-*t*-fördelning mellan *lowBound* och *upBound* för den specificerade frihetsgraden *df*.

För  $P(X \leq \text{upBound})$ , sätt *lowBound* =  $-\infty$ .

**tCollect()**Katalog > **tCollect**(*Expr1*) $\Rightarrow$ uttryck

Ger ett uttryck i vilket produkter och heltalspotenser av sinus och cosinus konverteras till en linjär kombination av sinus och cosinus för multipla vinklar, vinkelsummor och vinkelskillnader. Transformationen konverterar trigonometriska polynom till en linjär kombination.

|                                                   |                                                         |
|---------------------------------------------------|---------------------------------------------------------|
| $\text{tCollect}(\cos(\alpha)^2)$                 | $\frac{\cos(2 \cdot \alpha) + 1}{2}$                    |
| $\text{tCollect}(\sin(\alpha) \cdot \cos(\beta))$ | $\frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2}$ |

Ibland uppnår **tCollect()** dina mål när den förinställda trigonometriska förenklingen inte gör det. **tCollect()** tenderar att omkasta transformationer utförda av **tExpand()**. Ibland ger två separata steg med **tExpand()** på ett resultat från **tCollect()**, eller vice versa, en förenkling av ett uttryck.

## tExpand()

**tExpand(Expr1)⇒uttryck**

Ger ett uttryck i vilket sinus och cosinus för multipla heltalsvinklar, vinkelsummor och vinkelskillnader utvecklas. På grund av identiteten  $(\sin(x))^2 + (\cos(x))^2 = 1$  finns det många möjliga ekvivalenta resultat. Som en följd härav kan ett resultat skilja sig från ett resultat som visas i andra publikationer.

Ibland uppnår **tExpand()** dina mål när den förinställda trigonometriska förenklingen inte gör det. **tExpand()** tenderar att omkasta transformationer utförda av **tCollect()**. Ibland ger två separata steg med **tCollect()** på ett resultat från **tExpand()**, eller vice versa, en förenkling av ett uttryck.

**Obs:** I läget Grader stör skalning med  $\pi/180$  förmågan hos **tExpand()** att känna igen former som kan utvecklas. För bästa resultat bör **tExpand()** användas i läget Radianer.

$$\begin{aligned} \text{tExpand}(\sin(3 \cdot \phi)) &= 4 \cdot \sin(\phi) \cdot (\cos(\phi))^2 - \sin(\phi) \\ \text{tExpand}(\cos(\alpha - \beta)) &= \cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta) \end{aligned}$$

## Text

**TextfrågeSträng[, DispFlagga]**

Programmeringskommando: Pausar programmet och visar teckensträngen *frågeSträng* i en dialogruta.

När användaren väljer **OK** fortsätter exekveringen av programmet.

Definiera ett program som pausar för att visa vart och ett av fem slumptal i en dialogruta.

Inom mallen Prgm...EndPrgm template, fullborda varje rad genom att trycka på  i stället för **Enter**. På datorns tangentbord, håll ned **Alt** och tryck på **Enter**.

Det valfria argumentet *flagga* kan vara ett valfritt uttryck.

- Om *DispFlagga* utelämnas eller beräknas till **1** sparas textmeddelandet i Räknarens historik.
- Om *DispFlag* beräknas till **0** sparas textmeddelandet inte i historiken.

Om programmet behöver ett svar som skrivs in av användaren, se **Request**, på sidan 155 eller **RequestStr**, på sidan 157.

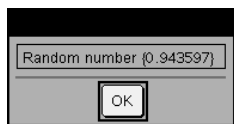
**Obs:** Du kan använda detta kommando inom ett användardefinierat program, men inte inom en funktion.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number " &
string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Kör programmet:

```
text_demo()
```

Exempel på en dialogruta:



**tInterval** *List[,Freq[,CLevel]]*

(Indata lista)

**tInterval**  $\bar{x}, s_x, n[, CLevel]$

(Summary stats indata)

Beräknar ett *t*-konfidensintervall. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel            | Beskrivning                                                                 |
|-----------------------------|-----------------------------------------------------------------------------|
| stat.CLower,<br>stat.CUpper | Konfidensintervall för ett okänt populationsmedelvärde                      |
| stat. $\bar{x}$             | Urvalsmedelvärde för datasekvensen från den slumpmässiga normalfördelningen |
| stat.ME                     | Felmarginal                                                                 |
| stat.df                     | Frihetsgrader                                                               |
| stat. $\sigma_x$            | Standardavvikelse för urvalet                                               |
| stat.n                      | Längden på datasekvenser med urvalsmedelvärde                               |

## tInterval\_2Samp

Katalog > 

**tInterval\_2Samp** *List1, List2[, Freq1[, Freq2  
[, CLevel[, Pooled]]]]*

(Indatalista)

**tInterval\_2Samp**  $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2$   
[, CLevel[, Pooled]]

(Summary stats indata)

Beräknar ett 2-sampel *t*-konfidensintervall.  
En sammanfattning av resultaten visas i  
variabeln *stat.results*. (Se på sidan 184.)

*Pooled=1* slår samman varianser, *Pooled=0*  
slår inte samman varianser.

För information om effekten av tomma  
element i en lista, se "Tomma element" (på  
sidan 261).

| Resultatvariabel                      | Beskrivning                                                                    |
|---------------------------------------|--------------------------------------------------------------------------------|
| stat.CLower,<br>stat.CUpper           | Konfidensintervall innehållande konfidensnivån för fördelningens sannolikhet   |
| stat. $\bar{x}1 - \bar{x}2$           | Urvalsmedelvärden för datasekvenserna från den slumpmässiga normalfördelningen |
| stat.ME                               | Felmarginal                                                                    |
| stat.df                               | Frihetsgrader                                                                  |
| stat. $\bar{x}1$ , stat. $\bar{x}2$   | Urvalsmedelvärden för datasekvenserna från den slumpmässiga normalfördelningen |
| stat. $\sigma x1$ , stat. $\sigma x2$ | Urvalets standardavvikelser för <i>List 1</i> och <i>List 2</i>                |



| Resultatvariabel | Beskrivning                                                            |
|------------------|------------------------------------------------------------------------|
| stat.n1, stat.n2 | Antalet samplingar i datasekvenser                                     |
| stat.sp          | Den sammanslagna standardavvikelsen. Beräknas när <i>Pooled</i> = YES. |

### tmpCnv()

Katalog >

**tmpCnv**(*Expr*  $^{\circ}$ *tempUnit*,  $^{\circ}$ *tempUnit2*)  
 $\Rightarrow$ uttryck  $^{\circ}$ *tempUnit2*

Konverterar ett temperaturvärde specificerat av *Expr* från en enhet till en annan. Giltiga temperaturenheter är:

$^{\circ}$ C Celsius

$^{\circ}$ F Fahrenheit

$^{\circ}$ K Kelvin

$^{\circ}$ R Rankine

För att skriva in symbolen  $^{\circ}$ , välj den på symbollistan i Katalogen.

För att skriva in symbolen  $_{}$ , tryck på

.

Som exempel konverteras 100  $^{\circ}$ C till 212  $^{\circ}$ F.

För att konvertera ett temperaturintervall, använd  $\Delta$ tmpCnv() i stället.

|                                        |                     |
|----------------------------------------|---------------------|
| tmpCnv(100 $^{\circ}$ C, $^{\circ}$ F) | 212 $^{\circ}$ F    |
| tmpCnv(32 $^{\circ}$ F, $^{\circ}$ C)  | 0 $^{\circ}$ C      |
| tmpCnv(0 $^{\circ}$ C, $^{\circ}$ K)   | 273.15 $^{\circ}$ K |
| tmpCnv(0 $^{\circ}$ F, $^{\circ}$ R)   | 459.67 $^{\circ}$ R |

**Obs:** Du kan använda Katalogen för att välja temperaturenheter.

### $\Delta$ tmpCnv()

Katalog >

**$\Delta$ tmpCnv**(*Expr*  $^{\circ}$ *tempUnit*,  $^{\circ}$ *tempUnit2*)  $\Rightarrow$ uttryck  $^{\circ}$ *tempUnit2*

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **deltaTmpCnv (...)**.

Omvandlar ett temperaturintervall (skillnaden mellan två temperaturvärden) specificerat av *Expr* från en enhet till en annan. Giltiga temperaturenheter är:

$^{\circ}$ C Celsius

För att skriva in symbolen  $\Delta$ , välj den på symbollistan i Katalogen.

|                                                 |                  |
|-------------------------------------------------|------------------|
| $\Delta$ tmpCnv(100 $^{\circ}$ C, $^{\circ}$ F) | 180 $^{\circ}$ F |
| $\Delta$ tmpCnv(180 $^{\circ}$ F, $^{\circ}$ C) | 100 $^{\circ}$ C |
| $\Delta$ tmpCnv(100 $^{\circ}$ C, $^{\circ}$ K) | 100 $^{\circ}$ K |
| $\Delta$ tmpCnv(100 $^{\circ}$ F, $^{\circ}$ R) | 100 $^{\circ}$ R |
| $\Delta$ tmpCnv(1 $^{\circ}$ C, $^{\circ}$ F)   | 1.8 $^{\circ}$ F |

**Obs:** Du kan använda Katalogen för att välja temperaturenheter.

## $\Delta$ tmpCnv()

Katalog > 

\_ °F Fahrenheit

\_ °K Kelvin

\_ °R Rankine

För att mata in °, välj den på symbolpaletten eller skriv @d.

För att skriva in symbolen \_, tryck på

 .

1\_ °C och 1\_ °K har samma storlek och likaså 1\_ °F och 1\_ °R. 1\_ °C är dock 9/5 större än 1\_ °F.

Som exempel motsvarar ett 100\_ °C-område (från 0\_ °C till 100\_ °C) ett 180\_ °F-område.

För att konvertera ett visst temperaturvärde i stället för ett intervall, använd **tmpCnv()**.

## tPdf()

Katalog > 

**tPdf(XVal,df)** ⇒ *tal* om *XVal* är ett tal, *lista* om *XVal* är en lista

Beräknar värde hos täthetsfunktionen (pdf) för Student-*t*-fördelningen vid ett specificerat *x*-värde med den specificerade frihetsgraden *df*.

## trace()

Katalog > 

**trace(squareMatrix)** ⇒ *uttryck*

Ger spåret (summan av alla elementen på huvuddiagonalen) av *squareMatrix*.

|       |                                                                     |     |
|-------|---------------------------------------------------------------------|-----|
| trace | $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ | 15  |
| trace | $\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$                      | 2·a |

## Try

*block1*

## Else

*block2*

## EndTry

Exekverar *block1* såvida inte ett fel uppstår. Programexekveringen överförs till *block2* om ett fel uppstår i *block1*. Systemvariabeln *errCode* innehåller felkoden som låter programmet utföra återhämtning efter fel. För en lista på felkoder, se "*Felkoder och meddelanden*" (på sidan 271).

*block1* och *block2* kan vara antingen ett enstaka påstående eller en serie av påståenden separerade med tecknet ":".

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

## Exempel 2

För att se kommandona **Try**, **ClrErr** och **PassErr** i arbete, mata in programmet *eigenvals()* som visas till höger. Kör programmet genom att exekvera vart och ett av följande uttryck.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$\text{eigenvals}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

**Obs:** Se även **ClrErr**, på sidan 26 och **PassErr**, på sidan 137.

```
Define progI()=Prgm
  Try
    z:=z+1
    Disp "z incremented."
  Else
    Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

```
z:=1:progI()
z incremented.
```

Done

```
DelVar z:progI()
Sorry, z undefined.
```

Done

Definiera *eigenvals(a,b)=Prgm*

© Programmet *eigenvals(A,B)* visar egenvärdena för A-B

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Eigenvalues A-B are:",eigVl(a\*b)

Else

If errCode=230 Then

Disp "Error: Product of A-B must be a square matrix"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrmg

**tTest****tTest**  $\mu_0, List[, Freq[, Hypoth]]$ 

(Indatalista)

**tTest**  $\mu_0, \bar{x}, sx, n[, Hypoth]$ 

(Summary stats indata)

Utför ett hypotestest för ett okänt populationsmedelvärde,  $\mu$ , när populationens standardavvikelse,  $\sigma$ , är okänd. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Testa  $H_0: \mu = \mu_0$ , mot en av följande:

För  $H_a: \mu < \mu_0$ , ställ *Hypoth* < 0

För  $H_a: \mu \neq \mu_0$  (förinställning), ställ *Hypoth* = 0

För  $H_a: \mu > \mu_0$ , ställ *Hypoth* > 0

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                   |
|------------------|---------------------------------------------------------------|
| stat.t           | $(\bar{x} - \mu_0) / (\text{stdev} / \text{sqrt}(n))$         |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas |
| stat.df          | Frihetsgrader                                                 |
| stat. $\bar{x}$  | Urvalsmedelvärde för datasekvensen i <i>List</i>              |
| stat.sx          | Urvalets standardavvikelse för datasekvensen                  |
| stat.n           | Urvalets storlek                                              |

**tTest\_2Samp** *List1, List2[, Freq1[, Freq2  
[, Hypoth[, Pooled]]]]*

(Indata lista)

**tTest\_2Samp**  $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2[, Hypoth  
[, Pooled]]$

(Summary stats indata)

Beräknar ett 2-sampel *t*-test. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Testa  $H_0: \mu_1 = \mu_2$ , mot en av följande:

För  $H_a: \mu_1 < \mu_2$ , ställ *Hypoth* < 0

För  $H_a: \mu_1 \neq \mu_2$  (förinställning), ställ *Hypoth* = 0

För  $H_a: \mu_1 > \mu_2$ , ställ *Hypoth* > 0

*Pooled* = 1 slår samman varianser,

*Pooled* = 0 slår inte samman varianser.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel                    | Beskrivning                                                                       |
|-------------------------------------|-----------------------------------------------------------------------------------|
| stat.t                              | Standardnormalvärde beräknat för skillnaden mellan medelvärden                    |
| stat.PVal                           | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas                     |
| stat.df                             | Frihetsgrader hos t-statistiken                                                   |
| stat. $\bar{x}1$ , stat. $\bar{x}2$ | Medelvärden hos urvalet i datasekvenserna i <i>List 1</i> och <i>List 2</i>       |
| stat.sx1, stat.sx2                  | Standardavvikelse hos urvalet i datasekvenserna i <i>List 1</i> och <i>List 2</i> |
| stat.n1, stat.n2                    | Storlek på urvalen                                                                |
| stat.sp                             | Den sammanslagna standardavvikelsen. Beräknad när <i>Pooled</i> = 1.              |

## tvmFV()

**tvmFV** $\{N, I, PV, Pmt, [PpY], [CpY], [PmtAt]\} \Rightarrow \text{värde}$

tvmFV(120, 5, 0, -500, 12, 12)

77641.1

**tvmFV()**Katalog > 

Finansiell funktion som beräknar det framtida värdet på pengar.

**Obs:** Argumenten som används i TVM-funktionerna beskrivs i tabellen över TVM-argumenten, på sidan 203. Se även **amortTbl()**, på sidan 8.

**tvmI()**Katalog > 

**tvmI**(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ *värde*

|                                             |         |
|---------------------------------------------|---------|
| <code>tvmI(240,100000,-1000,0,12,12)</code> | 10.5241 |
|---------------------------------------------|---------|

Finansiell funktion som beräknar räntesatsen per år.

**Obs:** Argumenten som används i TVM-funktionerna beskrivs i tabellen över TVM-argumenten, på sidan 203. Se även **amortTbl()**, på sidan 8.

**tvmN()**Katalog > 

**tvmN**(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ *värde*

|                                         |      |
|-----------------------------------------|------|
| <code>tvmN(5,0,-500,77641,12,12)</code> | 120. |
|-----------------------------------------|------|

Finansiell funktion som beräknar antalet betalningsperioder.

**Obs:** Argumenten som används i TVM-funktionerna beskrivs i tabellen över TVM-argumenten, på sidan 203. Se även **amortTbl()**, på sidan 8.

**tvmPmt()**Katalog > 

**tvmPmt**(*N,I,PV,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ *värde*

|                                         |          |
|-----------------------------------------|----------|
| <code>tvmPmt(60,4,30000,0,12,12)</code> | -552.496 |
|-----------------------------------------|----------|

Finansiell funktion som beräknar beloppet på varje betalning.

**Obs:** Argumenten som används i TVM-funktionerna beskrivs i tabellen över TVM-argumenten, på sidan 203. Se även **amortTbl()**, på sidan 8.

**tvmPV**( $N, I, Pmt, FV, [PpY], [CpY], [PmtAt]$ ) $\Rightarrow$ värde

tvmPV(48, 4, -500, 30000, 12, 12)      -3426.7

Finansiell funktion som beräknar nuvärdet.

**Obs:** Argumenten som används i TVM-funktionerna beskrivs i tabellen över TVM-argumenten, på sidan 203. Se även **amortTbl()**, på sidan 8.

| TVM-argument* | Beskrivning                                                                                                 | Datatyp                         |
|---------------|-------------------------------------------------------------------------------------------------------------|---------------------------------|
| $N$           | Antal betalningsperioder                                                                                    | reellt tal                      |
| $I$           | Årlig räntesats                                                                                             | reellt tal                      |
| $PV$          | Nuvärde                                                                                                     | reellt tal                      |
| $Pmt$         | Betalningsbelopp                                                                                            | reellt tal                      |
| $FV$          | Framtida värde                                                                                              | reellt tal                      |
| $PpY$         | Antal betalningar per år, förinställning = 1                                                                | heltal > 0                      |
| $CpY$         | Ränteperioder per år, förinställning = 1                                                                    | heltal > 0                      |
| $PmtAt$       | Betalning som skall betalas i slutet (end) eller i början (beginning) av varje period, förinställning = end | heltal (0 = end, 1 = beginning) |

\* Dessa argumentnamn avseende tidsjusterat pengavärde påminner om namnen på TVM-variablerna (till exempel, **tvm.pv** och **tvm.pmt**) som används av Finance Solver i applikationen *Calculator*. Finansiella funktioner lagrar dock inte sina argumentvärden eller resultat i TVM-variablerna.

**TwoVar**  $X, Y, [Freq], [Category, Include]$

Utför tvåvariabelstatistik. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Alla listor utom *Include* måste ha samma dimensioner.

$X$  och  $Y$  är listor på oberoende och beroende variabler.

*Freq* är en frivillig lista på frekvensvärden. Varje element i *Freq* specificerar frekvensen för varje motsvarande *X*- och *Y*-datapunkt. Det förinställda värdet är 1. Alla element måste vara heltal  $\geq 0$ .

*Category* är en lista på kategorikoder för motsvarande *X*- och *Y*-data.

*Include* är en lista på en eller flera av kategorikoderna. Endast de dataobjekt vars kategorikod är med på listan tas med i beräkningen.

Ett tomt element i någon av listorna *X*, *Freq* eller *Category* resulterar i ett tomrum för motsvarande element i dessa listor. Ett tomt element i någon av listorna *X1* till och med *X20* resulterar i ett tomrum för motsvarande element i dessa listor. För mer information om tomma element, se på sidan 261.

| Resultatvariabel     | Beskrivning                           |
|----------------------|---------------------------------------|
| stat. $\bar{x}$      | Medelvärde av x-värden                |
| stat. x              | Summa av x-värden                     |
| stat. x2             | Summa av x2-värden                    |
| stat.sx              | Standardavvikelse för x (sampling)    |
| stat. x              | Standardavvikelse för x (population)  |
| stat.n               | Antal datapunkter                     |
| stat. $\bar{y}$      | Medelvärde av y-värden                |
| stat. y              | Summa av y-värden                     |
| stat. y <sup>2</sup> | Summa av y2-värden                    |
| stat.sy              | Standardavvikelse för y (sampling)    |
| stat. y              | Populationens standardavvikelse för y |
| stat. xy             | Summa av x · y-värden                 |
| stat.r               | Korrelationskoefficient               |
| stat.MinX            | Minsta x-värde                        |



| Resultatvariabel         | Beskrivning                                       |
|--------------------------|---------------------------------------------------|
| stat.Q <sub>1</sub> X    | Undre kvartil för x                               |
| stat.MedianX             | Median för x                                      |
| stat.Q <sub>3</sub> X    | Övre kvartil för x                                |
| stat.MaxX                | Största x-värde                                   |
| stat.MinY                | Minsta y-värde                                    |
| stat.Q <sub>1</sub> Y    | Undre kvartil för y                               |
| stat.MedY                | Median för y                                      |
| stat.Q <sub>3</sub> Y    | Övre kvartil för y                                |
| stat.MaxY                | Största y-värde                                   |
| stat. (x- ) <sup>2</sup> | Kvadratsumma för avvikelser från medelvärdet på x |
| stat. (y- ) <sup>2</sup> | Kvadratsumma för avvikelser från medelvärdet på y |

## U

### unitV()

Katalog > 

**unitV(*VectorI*)** ⇒ vektor

Ger en radenhets- eller kolumnenhetsvektor beroende på formen hos *VectorI*.

*VectorI* måste vara antingen en enradsmatris eller en enkolumnsmatris.

$$\begin{aligned} &\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) \\ &\quad \left[ \frac{a}{\sqrt{a^2+b^2+c^2}} \quad \frac{b}{\sqrt{a^2+b^2+c^2}} \quad \frac{c}{\sqrt{a^2+b^2+c^2}} \right] \\ &\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) \quad \left[ \frac{\sqrt{6}}{6} \quad \frac{\sqrt{6}}{3} \quad \frac{\sqrt{6}}{6} \right] \\ &\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) \quad \left[ \begin{array}{c} \frac{\sqrt{14}}{14} \\ \frac{14}{\sqrt{14}} \\ 7 \\ \frac{3 \cdot \sqrt{14}}{14} \end{array} \right] \end{aligned}$$

För att se hela resultatet, tryck på ▲ och använd sedan ◀ och ▶ för att flytta markören.

| unLock                                                                                                           | Katalog >  |                              |
|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------|
| <b>unLock</b> <i>Var1</i> [, <i>Var2</i> ] [, <i>Var3</i> ] ...                                                  | <i>a</i> :=65                                                                             | 65                           |
| <b>unLock</b> <i>Var</i> .                                                                                       | Lock <i>a</i>                                                                             | Done                         |
| Låser upp den specificerade variabeln eller variabelgruppen. Låsta variabler kan inte modifieras eller tas bort. | getLockInfo( <i>a</i> )                                                                   | 1                            |
| Se <b>Lock</b> , på sidan 111 och <b>getLockInfo()</b> , på sidan 86.                                            | <i>a</i> :=75                                                                             | "Error: Variable is locked." |
|                                                                                                                  | DelVar <i>a</i>                                                                           | "Error: Variable is locked." |
|                                                                                                                  | Unlock <i>a</i>                                                                           | Done                         |
|                                                                                                                  | <i>a</i> :=75                                                                             | 75                           |
|                                                                                                                  | DelVar <i>a</i>                                                                           | Done                         |

## V

| varPop()                                                                                         | Katalog >  |         |
|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------|
| <b>varPop</b> ( <i>List</i> [, <i>freqList</i> ])⇒ <i>uttryck</i>                                | varPop({ 5,10,15,20,25,30 })                                                                | 875     |
| Ger populationsvariansen för <i>List</i> .                                                       |                                                                                             | 12      |
| Varje <i>freqList</i> -element räknar antalet förekomster av motsvarande element i <i>List</i> . | Ans: 1.                                                                                     | 72.9167 |

**Obs:** *List* måste innehålla minst två element.

Om ett element i endera listan är tomt ignoreras detta element och även motsvarande element i den andra listan ignoreras. För mer information om tomma element, se på sidan 261.

| varSamp()                                                                                        | Katalog >  |    |
|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----|
| <b>varSamp</b> ( <i>List</i> [, <i>freqList</i> ])⇒ <i>uttryck</i>                               | varSamp({ { <i>a</i> , <i>b</i> , <i>c</i> } })                                              |    |
| Ger sampelvariansen för <i>List</i> .                                                            | $\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$                                      |    |
| Varje <i>freqList</i> -element räknar antalet förekomster av motsvarande element i <i>List</i> . | varSamp({ { 1,2,5, 6,3, 2 } })                                                               | 31 |
|                                                                                                  |                                                                                              | 2  |
|                                                                                                  | varSamp({ { 1,3,5 }, { 4,6,2 } })                                                            | 68 |
|                                                                                                  |                                                                                              | 33 |

**Obs:** *List* måste innehålla minst två element.

Om ett element i endera listan är tomt ignoreras detta element och även motsvarande element i den andra listan ignoreras. För mer information om tomma element, se på sidan 261.

**varSamp**(*Matrix1* [, *freqMatrix*]) ⇒ *matris*

Ger en radvektor som innehåller sampelvariansen för varje kolumn i *Matrix1*.

Varje *freqMatrix*-element räknar antalet konsekutiva förekomster av motsvarande element i *Matrix1*.

**Obs:** *Matrix1* måste innehålla minst två rader.

Om ett element i endera matrisen är tomt ignoreras detta element och även motsvarande element i den andra matrisen ignoreras. För mer information om tomma element, se på sidan 261.

```
varSamp( $\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$ )  $\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$ 
```

---

```
varSamp( $\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}, \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$ )  $\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$ 
```

## W

### Wait

#### Wait *tidISekunder*

Fördröjer exekvering för en period som varar *tidISekunder* sekunder.

**Wait** är speciellt användbart i ett program som behöver en kort fördröjning för att begärda data ska bli tillgängliga.

Argumentet *tidISekunder* måste vara ett uttryck som förenklas till ett decimalvärde i intervallet 0 till 100. Kommandot avrundar detta värde till närmaste 0,1 sekunder.

För att avbryta en **Wait** som pågår,

- **Handenhet:** Håll ned  och tryck på  upprepade gånger.
- **Windows®:** Håll ned **F12** och tryck på **Enter** upprepade gånger.

För att vänta 4 sekunder:

**Wait 4**

För att vänta 1/2 sekund:

**Wait 0.5**

För att vänta 1,3 sekunder med hjälp av variabeln *sekantal*:

**sekantal:=1.3**

**Wait sekantal**

I detta exempel tänds en grön lysdiod i 0,5 sekunder och släcks sedan.

**Send "SET GREEN 1 ON"**

**Wait 0.5**

**Send "SET GREEN 1 OFF"**

- **Macintosh®:** Håll ned **F5** och tryck på **Enter** upprepade gånger.
- **iPad®:** Appen visar en uppmaning. Du kan fortsätta att vänta eller avbryta.

**Obs:** Du kan använda kommandot **Wait** i ett användardefinierat program, men inte inom en funktion.

## warnCodes ()

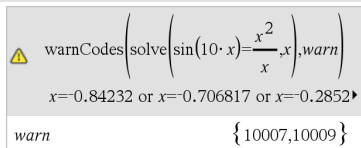
**warnCodes**(*Expr1*, *StatusVar*) $\Rightarrow$ uttryck

Utvärderar uttrycket *Expr1*, ger resultatet och lagrar eventuellt genererade varningskoder i listvariabeln *StatusVar*. Om inga varningar genereras tilldelar denna funktion *StatusVar* en tom lista.

*Expr1* kan vara ett valfritt giltigt matematiskt uttryck i TI-Nspire™ eller TI-Nspire™ CAS. Du kan inte använda ett kommando eller en tilldelning som *Expr1*.

*StatusVar* måste vara ett giltigt variabelnamn.

För en lista på varningskoder och tillhörande meddelanden, se på sidan 280.



För att se hela resultatet, tryck på  $\blacktriangle$  och använd sedan  $\blacktriangleleft$  och  $\blacktriangleright$  för att flytta markören.

## when()

**when**(*Condition*, *trueResult* [, *falseResult*][, *unknownResult*]) $\Rightarrow$ uttryck

Ger *trueResult*, *falseResult* eller *unknownResult* beroende på om *Condition* är sant, falskt eller okänt. Ger indata om det finns alltför få argument för att specificera ett lämpligt resultat.

| when()                                                                                                                                                | Katalog >  |       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------|
| Utelämnna både <i>falseResult</i> och <i>unknownResult</i> för att skapa ett uttryck som är definierat endast i området där <i>Condition</i> är sant. |                                                                                           |       |
| Använd ett <b>undef</b> <i>falseResult</i> för att definiera ett uttryck som plottas endast i ett intervall.                                          |                                                                                           |       |
| <b>when()</b> är användbar för att definiera rekursiva funktioner.                                                                                    |                                                                                           |       |
|                                                                                                                                                       | $\text{when}(x < 0, x + 3)   x = 5$                                                       | undef |
|                                                                                                                                                       | $\text{when}(n > 0, n \cdot \text{factorial}(n - 1), 1) \rightarrow \text{factorial}(n)$  | Done  |
|                                                                                                                                                       | $\text{factorial}(3)$                                                                     | 6     |
|                                                                                                                                                       | $3!$                                                                                      | 6     |

| While                                                                                                                                                     | Katalog >  |                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|----------------|
| <b>While</b> <i>Condition</i>                                                                                                                             | Define $\text{sum\_of\_recip}(n) = \text{Func}$                                             |                |
| <i>Block</i>                                                                                                                                              | Local $i, \text{tempsum}$                                                                   |                |
|                                                                                                                                                           | $1 \rightarrow i$                                                                           |                |
| <b>EndWhile</b>                                                                                                                                           | $0 \rightarrow \text{tempsum}$                                                              |                |
| Exekverar påståendena i <i>Block</i> så länge <i>Condition</i> är sant.                                                                                   | While $i \leq n$                                                                            |                |
| <i>Block</i> kan vara antingen ett enstaka påstående eller en serie av påståenden separerade med tecknet “.”.                                             | $\text{tempsum} + \frac{1}{i} \rightarrow \text{tempsum}$                                   |                |
|                                                                                                                                                           | $i + 1 \rightarrow i$                                                                       |                |
|                                                                                                                                                           | EndWhile                                                                                    |                |
|                                                                                                                                                           | Return $\text{tempsum}$                                                                     |                |
|                                                                                                                                                           | EndFunc                                                                                     |                |
|                                                                                                                                                           | Done                                                                                        |                |
| <b>Obs för att mata in exemplet:</b> Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner. | $\text{sum\_of\_recip}(3)$                                                                  | $\frac{11}{6}$ |

| X                                                                                |                                                                                               |       |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|-------|
| xor                                                                              | Katalog >  |       |
| <i>BoolesktUtr1</i> <b>xor</b> <i>BoolesktUtr2</i> ger <i>Booleskt uttryck</i>   | $\text{true xor true}$                                                                        | false |
|                                                                                  | $5 > 3 \text{ xor } 3 > 5$                                                                    | true  |
| <i>BooleskLista1</i> <b>xor</b> <i>BooleskLista2</i> ger <i>Boolesk lista</i>    |                                                                                               |       |
| <i>BooleskMatris1</i> <b>xor</b> <i>BooleskMatris2</i> ger <i>Boolesk matris</i> |                                                                                               |       |

Ger resultatet sant om *BooleanExpr1* är sant och *BooleanExpr2* är falskt, eller vice versa.

Ger resultatet falskt om båda argumenten är sanna eller falska. Ger ett förenklat booleskt uttryck om något av argumenten inte kan lösas om till sant eller falskt.

**Obs:** Se *eller*, här.

*Integer1 xor Integer2*  $\Rightarrow$  *heltal*

Jämför två reella heltal bit för bit med en **xor**-operation. Internt omvandlas båda heltalen till 64-bitars binära tal. När motsvarande bitar jämförs blir resultatet 1 om en bit (men inte båda) är 1. Resultatet blir 0 om båda bitarna är 0 eller 1. Det erhållna värdet representerar bitresultaten och visas enligt det inställda basläget.

Du kan skriva in heltalen i valfri talbas. För en binär eller hexadecimal inmatning måste du använda prefixet 0b respektive 0h. Utan prefix behandlas heltalen som decimala (bas 10).

Om du skriver in ett decimalt heltal som är alltför stort för att anges i 64-bitars binär form används en symmetrisk modulooperation för att få ned värdet till lämplig nivå. För mer information, se **►Base2**, på sidan 18.

**Obs:** Se *eller*, här.

I hexadecimalt basläge:

**Viktigt:** Noll, inte bokstaven O.

|                    |         |
|--------------------|---------|
| 0h7AC36 xor 0h3D5F | 0h79169 |
|--------------------|---------|

I binärt basläge:

|                    |          |
|--------------------|----------|
| 0b100101 xor 0b100 | 0b100001 |
|--------------------|----------|

**Obs:** En binär inmatning kan ha upp till 64 siffror (exklusive prefixet 0b). En hexadecimal inmatning kan ha upp till 16 siffror.

## Z

### zeros()

**zeros(Expr, Var)**  $\Rightarrow$  *lista*

**zeros(Expr, Var=Guess)**  $\Rightarrow$  *lista*

Ger en lista på möjliga reella värden på *Var* som gör *Expr*=0. **zeros()** gör detta genom att beräkna **exp►list(solve(Expr=0, Var), Var)**.

$$\text{zeros}(a \cdot x^2 + b \cdot x + c, x) = \left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\}$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

För vissa ändamål är resultatformen för **zeros()** mer praktisk än den för **solve()**. Resultatformen för **zeros()** kan dock inte uttrycka implicita lösningar, lösningar som kräver olikheter eller lösningar som inte inbegriper *Var*.

**Obs:** Se även **cSolve()**, **cZeros()** och **solve()**.

**zeros**({*Expr1*, *Expr2*}, {*VarOrGuess1*, *VarOrGuess2* [, ... ]}) ⇒ *matris*

Ger möjliga reella nollställen för de samtidiga algebraiska uttrycken där varje *VarOrGuess* specificerar en okänd vars värde du söker.

Du kan som alternativ specificera en initial gissning för en variabel. Varje *VarOrGuess* måste ha formen:

*variable*

– eller –

*variable = reellt eller icke-reellt tal*

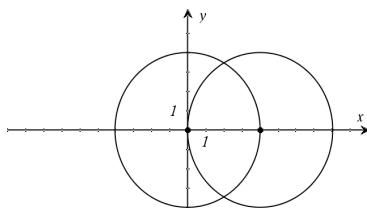
Som exempel är *x* giltigt och likaså *x=3*.

Om alla uttryck är polynom och om du INTE specificerar några initiala gissningar använder **zeros()** eliminationsmetoden Gröbner/Buchberger för att försöka bestämma alla reella nollställen.

Lås oss som exempel anta att du har en cirkel med radien *r* i origo och en annan cirkel med radien *r* där centrum är där den första cirkeln korsar den positiva *x*-axeln. Använd **zeros()** för att finna skärningspunkterna.

Såsom illustreras med *r* i exemplet till höger kan system av polynomuttryck ha extra variabler som saknar värden, men representerar givna numeriska värden som kan ersättas senare.

$$\begin{array}{l} \text{exact}\left(\text{zeros}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1), x\right)\right) \quad \left\{ \begin{array}{c} \boxed{\phantom{x}} \\ \boxed{\phantom{x}} \end{array} \right\} \\ \text{exact}\left(\text{solve}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1) = 0, x\right)\right) \\ e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0 \end{array}$$



$$\text{zeros}\left(\left\{x^2 + y^2 - r^2, (x - r)^2 + y^2 - r^2\right\}, \{x, y\}\right)$$

|               |                               |
|---------------|-------------------------------|
| $\frac{r}{2}$ | $\frac{-\sqrt{3} \cdot r}{2}$ |
| $\frac{r}{2}$ | $\frac{\sqrt{3} \cdot r}{2}$  |

Varje rad i den resulterande matrisen representerar ett alternativt nollställe, med komponenterna ordnade på samma sätt som i listan *VarOrGuess*. För att extrahera en rad, indexera matrisen med [row].

Du kan även (eller i stället) inkludera okända som inte visas i uttrycken. Du kan exempelvis inkludera z som en okänd för att utöka föregående exempel till två parallella överkorsande cylindrar med radien r. Cylinderlösningarna illustrerar hur familjer av nollställten kan innehålla godtyckliga konstanter med formen ck där k är ett heltalssuffix från 1 till och med 255.

För polynomsystem kan beräkningstiden och användningen av minne i hög grad bero på i vilken ordning du listar okända. Om ditt första val utarmar minnet, eller tar på ditt tålamod, kan du försöka med att arrangera om variablerna i uttrycken och/eller i listan *VarOrGuess*.

Om du inte inkluderar några gissningar och om något uttryck är ett icke-polynom i någon variabel, men alla uttryck är linjära i de okända, använder **zeros()** Gauss eliminationsmetod för att försöka bestämma alla reella nollställten.

Om ett system är varken polynomt i alla dess variabler eller linjärt i dess okända bestämmer **zeros()** högst ett nollställe med en ungefärlig iterativ metod. För att göra detta måste antalet okända vara lika med antalet uttryck och alla övriga variabler i uttrycken måste förenklas till tal.

Varje okänd startar vid dess gissade värde om det finns något, annars startar den vid 0.0.

Använd gissningar för att söka ytterligare nollställten i taget. För konvergens kan en gissning behöva vara ganska nära ett nollställe.

Extrahera rad 2:

$$\text{Ans}[2] \quad \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{bmatrix}$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right) \quad \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \end{bmatrix}$$

$$\text{zeros}\left(\left\{x+e^z \cdot y-1, x-y-\sin(z)\right\}, \{x, y\}\right) \quad \begin{bmatrix} \frac{e^z \cdot \sin(z)+1}{e^z+1} & \frac{-\sin(z)-1}{e^z+1} \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z\}\right) \quad \begin{bmatrix} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76\text{E-}11 & 1796.99 \\ 2.\text{E-}13 & 254.469 \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z=2 \cdot \pi\}\right) \quad \begin{bmatrix} 0.001871 & 6.28131 \end{bmatrix}$$



**zInterval**  $\sigma, List[, Freq[, CLevel]]$

(Indata lista)

**zInterval**  $\sigma, \bar{x}, n [, CLevel]$

(Summary stats indata)

Beräknar ett z-konfidensintervall. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel            | Beskrivning                                                                 |
|-----------------------------|-----------------------------------------------------------------------------|
| stat.CLower,<br>stat.CUpper | Konfidensintervall för ett okänt populationsmedelvärde                      |
| stat. $\bar{x}$             | Urvalsmedelvärde för datasekvensen från den slumpmässiga normalfördelningen |
| stat.ME                     | Felmarginal                                                                 |
| stat.sx                     | Standardavvikelse för urvalet                                               |
| stat.n                      | Längden på datasekvenser med urvalsmedelvärde                               |
| stat. $\sigma$              | Känd populationsstandardavvikelse för datasekvensen <i>List</i>             |

## zInterval\_1Prop

**zInterval\_1Prop**  $x, n [, CLevel]$

Beräknar ett 1-proportion z-konfidensintervall. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

$x$  är ett icke negativt heltal.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel            | Beskrivning                                                                  |
|-----------------------------|------------------------------------------------------------------------------|
| stat.CLower,<br>stat.CUpper | Konfidensintervall innehållande konfidensnivån för fördelningens sannolikhet |

| Resultatvariabel | Beskrivning                               |
|------------------|-------------------------------------------|
| stat. $\hat{p}$  | Den beräknade proportionen lyckade försök |
| stat.ME          | Felmarginal                               |
| stat.n           | Antalet samplingar i datasekvens          |

## zInterval\_2Prop

Katalog > 

**zInterval\_2Prop**  $x1, n1, x2, n2[, CLevel]$

Beräknar ett 2-proportion z-konfidensintervall. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

$x1$  och  $x2$  är icke negativa heltal.

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel            | Beskrivning                                                                  |
|-----------------------------|------------------------------------------------------------------------------|
| stat.CLower,<br>stat.CUpper | Konfidensintervall innehållande konfidensnivån för fördelningens sannolikhet |
| stat. $\hat{p}$ Diff        | Den beräknade skillnaden mellan proportioner                                 |
| stat.ME                     | Felmarginal                                                                  |
| stat. $\hat{p}1$            | Proportionsuppskattning för första urvalet                                   |
| stat. $\hat{p}2$            | Proportionsuppskattning för andra urvalet                                    |
| stat.n1                     | Urvalets storlek i datasekvens ett                                           |
| stat.n2                     | Urvalets storlek i datasekvens två                                           |

## zInterval\_2Samp

Katalog > 

**zInterval\_2Samp**  $\sigma_1, \sigma_2, List1, List2[, Freq1[, Freq2[, CLevel]]]$

(Indatalista)

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, CLevel]$

(Summary stats indata)

Beräknar ett 2-sampel  $z$ -konfidensintervall.  
En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel                      | Beskrivning                                                                             |
|---------------------------------------|-----------------------------------------------------------------------------------------|
| stat.CLower, stat.CUpper              | Konfidensintervall innehållande konfidensnivån för fördelningens sannolikhet            |
| stat. $\bar{x}1-\bar{x}2$             | Urvalsmedelvärden för datasekvenserna från den slumpmässiga normalfördelningen          |
| stat.ME                               | Felmarginal                                                                             |
| stat. $\bar{x}1$ , stat. $\bar{x}2$   | Urvalsmedelvärden för datasekvenserna från den slumpmässiga normalfördelningen          |
| stat. $\sigma x1$ , stat. $\sigma x2$ | Urvalets standardavvikelser för <i>List 1</i> och <i>List 2</i>                         |
| stat.n1, stat.n2                      | Antalet samplingar i datasekvenser                                                      |
| stat.r1, stat.r2                      | Kända populationsstandardavvikelser för datasekvenserna <i>List 1</i> och <i>List 2</i> |

## zTest

**zTest**  $\mu0, \sigma, List, [Freq[, Hypoth]]$

(Indatalista)

**zTest**  $\mu0, \sigma, \bar{x}, n[, Hypoth]$

(Summary stats indata)

Utför ett  $z$ -test med frekvensen *freqlist*. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Testa  $H_0: \mu = \mu0$ , mot en av följande:

För  $H_a: \mu < \mu0$ , ställ *Hypoth*<0

För  $H_a: \mu \neq \mu0$  (förinställning), ställ *Hypoth*=0

För  $H_a: \mu > \mu0$ , ställ *Hypoth*>0

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                                            |
|------------------|----------------------------------------------------------------------------------------|
| stat.z           | $(\bar{x} - \mu_0) / (\sigma / \text{sqrt}(n))$                                        |
| stat.P Value     | Lägsta sannolikhet vid vilken nollhypotesen kan förkastas                              |
| stat. $\bar{x}$  | Urvalsmedelvärde för datasekvensen i <i>List</i>                                       |
| stat.sx          | Urvalets standardavvikelse för datasekvensen. Ges endast för <i>Data</i> -inmatningen. |
| stat.n           | Urvalets storlek                                                                       |

## zTest\_1Prop

**zTest\_1Prop**  $p_0, x, n, [Hypothesis]$

Beräknar ett 1-proportion z-test. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

$x$  är ett icke negativt heltal.

Testa  $H_0: p = p_0$  mot en av följande:

För  $H_a: p > p_0$ , ställ *Hypothesis* > 0

För  $H_a: p \neq p_0$  (förinställning), ställ *Hypothesis* = 0

För  $H_a: p < p_0$ , ställ *Hypothesis* < 0

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                   |
|------------------|---------------------------------------------------------------|
| stat.p0          | Hypotetisk populationsproportion                              |
| stat.z           | Standardnormalvärde beräknat för proportionen                 |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas |
| stat. $\hat{p}$  | Uppskattad urvalsproportion                                   |
| stat.n           | Urvalets storlek                                              |

**zTest\_2Prop**  $x1, n1, x2, n2[, Hypoth]$ 

Beräknar ett 2-proportion z-test. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

$x1$  och  $x2$  är icke negativa heltal.

Testa  $H_0: p1 = p2$  mot en av följande:

För  $H_a: p1 > p2$ , ställ *Hypoth*>0

För  $H_a: p1 \neq p2$  (förinställning), ställ *Hypoth*=0

För  $H_a: p < p0$ , ställ *Hypoth*<0

För information om effekten av tomma element i en lista, se "Tomma element" (på sidan 261).

| Resultatvariabel | Beskrivning                                                     |
|------------------|-----------------------------------------------------------------|
| stat.z           | Standardnormalvärde beräknat för skillnaden mellan proportioner |
| stat.PVal        | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas   |
| stat. $\hat{p}1$ | Proportionsuppskattning för första urvalet                      |
| stat. $\hat{p}2$ | Proportionsuppskattning för andra urvalet                       |
| stat. $\hat{p}$  | Uppskattning av sammanslagna urvalsproportioner                 |
| stat.n1, stat.n2 | Antal samplingar i försöksomgång 1 och 2                        |

**zTest\_2Samp****zTest\_2Samp**  $\sigma_1, \sigma_2, List1, List2[, Freq1[, Freq2[, Hypoth]]]$ 

(Indatalista)

**zTest\_2Samp**  $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, Hypoth]$ 

(Summary stats indata)

Beräknar ett 2-sampel z-test. En sammanfattning av resultaten visas i variabeln *stat.results*. (Se på sidan 184.)

Testa  $H_0: \mu1 = \mu2$ , mot en av följande:

För  $H_a: \mu_1 < \mu_2$ , ställ  $Hypoth < 0$

För  $H_a: \mu_1 \neq \mu_2$  (föreställning), ställ  
 $Hypoth = 0$

För  $H_a: \mu_1 > \mu_2$ , ställ  $Hypoth > 0$

För information om effekten av tomma  
element i en lista, se "Tomma element" (på  
sidan 261).

| Resultatvariabel   | Beskrivning                                                                      |
|--------------------|----------------------------------------------------------------------------------|
| stat.z             | Standardnormalvärde beräknat för skillnaden mellan medelvärden                   |
| stat.PVal          | Lägsta signifikansnivå vid vilken nollhypotesen kan förkastas                    |
| stat.x̄1, stat.x̄2 | Medelvärden hos urvalet i datasekvenserna i <i>List1</i> och <i>List2</i>        |
| stat.sx1, stat.sx2 | Standardavvikelser hos urvalet i datasekvenserna i <i>List1</i> och <i>List2</i> |
| stat.n1, stat.n2   | Storlek på urvalen                                                               |

# Symboler

| + (addera)                                                                                                                         |                                                                                               | + tangent                                          |
|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------|
| $Expr1 + Expr2 \Rightarrow \text{uttryck}$                                                                                         | 56                                                                                            | 56                                                 |
| Ger summan av de två argumenten.                                                                                                   | 56+4                                                                                          | 60                                                 |
|                                                                                                                                    | 60+4                                                                                          | 64                                                 |
|                                                                                                                                    | 64+4                                                                                          | 68                                                 |
|                                                                                                                                    | 68+4                                                                                          | 72                                                 |
| $List1 + List2 \Rightarrow \text{lista}$                                                                                           | $\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow l1$                                      | $\left\{ 22, \pi, \frac{\pi}{2} \right\}$          |
| $Matrix1 + Matrix2 \Rightarrow \text{matrix}$                                                                                      | $\left\{ 10, 5, \frac{\pi}{2} \right\} \rightarrow l2$                                        | $\left\{ 10, 5, \frac{\pi}{2} \right\}$            |
| Ger en lista (eller matrix) som innehåller summorna av motsvarande element i $List1$ och $List2$ (eller $Matrix1$ och $Matrix2$ ). | $l1+l2$                                                                                       | $\{ 32, \pi+5, \pi \}$                             |
|                                                                                                                                    | $Ans + \{ \pi, -5, \pi \}$                                                                    | $\{ \pi+32, \pi, 0 \}$                             |
| Argumenten måste ha samma dimensioner.                                                                                             | $\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$ |
| $Expr + List1 \Rightarrow \text{lista}$                                                                                            | $15 + \{ 10, 15, 20 \}$                                                                       | $\{ 25, 30, 35 \}$                                 |
| $List1 + Expr \Rightarrow \text{lista}$                                                                                            | $\{ 10, 15, 20 \} + 15$                                                                       | $\{ 25, 30, 35 \}$                                 |
| Ger en lista på summorna av $Expr$ och varje element i $List1$ .                                                                   |                                                                                               |                                                    |
| $Expr + Matrix1 \Rightarrow \text{matrix}$                                                                                         | $20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$                                           | $\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$   |
| $Matrix1 + Expr \Rightarrow \text{matrix}$                                                                                         |                                                                                               |                                                    |
| Ger en matrix med $Expr$ adderat till varje element i diagonalen av $Matrix1$ .<br>$Matrix1$ måste vara kvadratisk.                |                                                                                               |                                                    |
| <b>Obs:</b> Använd .+ (punkt plus) för att lägga till ett uttryck till varje element.                                              |                                                                                               |                                                    |
| -(subtrahera)                                                                                                                      |                                                                                               | - tangent                                          |
| $Expr1 - Expr2 \Rightarrow \text{uttryck}$                                                                                         | 6-2                                                                                           | 4                                                  |
| Ger $Expr1$ minus $Expr2$ .                                                                                                        | $\pi - \frac{\pi}{6}$                                                                         | $\frac{5 \cdot \pi}{6}$                            |

## -(subtrahera)

 tangent

$List1 - List2 \Rightarrow lista$

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} = \left\{ 12, \pi - 5, 0 \right\}$$

$Matrix1 - Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \end{bmatrix}$$

Subtraherar varje element i  $List2$  (eller  $Matrix2$ ) från motsvarande element i  $List1$  (eller  $Matrix1$ ) och ger resultatet.

Argumenten måste ha samma dimensioner.

$Expr - List1 \Rightarrow lista$

$$15 - \{ 10, 15, 20 \} = \{ 5, 0, -5 \}$$

$List1 - Expr \Rightarrow lista$

$$\{ 10, 15, 20 \} - 15 = \{ -5, 0, 5 \}$$

Subtraherar varje element i  $List1$  från  $Expr$  eller subtraherar  $Expr$  från varje element i  $List1$  och ger en lista på resultaten.

$Expr - Matrix1 \Rightarrow matrix$

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

$Matrix1 - Expr \Rightarrow matrix$

$Expr - Matrix1$  ger en matrix över  $Expr$  gånger enhetsmatrisen minus  $Matrix1$ .  $Matrix1$  måste vara kvadratisk.

$Matrix1 - Expr$  ger en matrix över  $Expr$  gånger enhetsmatrisen subtraherad från  $Matrix1$ .  $Matrix1$  måste vara kvadratisk.

**Obs:** Använd  $-$  (punkt minus) för att subtrahera ett uttryck från varje element.

## ·(multiplicera)

 tangent

$Expr1 \cdot Expr2 \Rightarrow uttryck$

$$2 \cdot 3.45 = 6.9$$

Ger produkten av de två argumenten.

$$x \cdot y \cdot x = x^2 \cdot y$$

$List1 \cdot List2 \Rightarrow lista$

$$\{ 1, 2, 3 \} \cdot \{ 4, 5, 6 \} = \{ 4, 10, 18 \}$$

Ger en lista som innehåller produkterna av motsvarande element i  $List1$  och  $List2$ .

$$\left\{ \frac{2}{a}, \frac{3}{2} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} = \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

Listorna måste ha samma dimensioner.



·(multiplicera)

 tangent

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

Ger en matris över produkten av  $Matrix1$  och  $Matrix2$ .

Antalet kolumner i  $Matrix1$  måste vara lika med antalet rader i  $Matrix2$ .

$Expr \cdot List1 \Rightarrow lista$

$List1 \cdot Expr \Rightarrow lista$

Ger en lista på produkterna av  $Expr$  och varje element i  $List1$ .

$Expr \cdot Matrix1 \Rightarrow matrix$

$Matrix1 \cdot Expr \Rightarrow matrix$

Ger en matris över produkterna av  $Expr$  och varje element i  $Matrix1$ .

**Obs:** Använd ·(punkt multiplicera) för att multiplicera ett uttryck med varje element.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} = \begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

$$\pi \cdot \{4,5,6\} \quad \{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$\lambda \cdot \text{identity}(3) = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

/ (dividera)

 tangent

$Expr1 / Expr2 \Rightarrow uttryck$

Ger kvoten av  $Expr1$  dividerat med  $Expr2$ .

**Obs:** Se även **Fraction template**, på sidan 1.

$List1 / List2 \Rightarrow lista$

Ger en lista på kvoterna av  $List1$  dividerat med  $List2$ .

Listorna måste ha samma dimensioner.

$Expr / List1 \Rightarrow lista$

$List1 / Expr \Rightarrow lista$

Ger en lista på kvoterna av  $Expr$  dividerat med  $List1$  eller  $List1$  dividerat med  $Expr$ .

$Matrix1 / Expr \Rightarrow matrix$

$$\frac{2}{3.45} = 0.57971$$

$$\frac{x^3}{x} = x^2$$

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

$$\frac{a}{\{3,a,\sqrt{a}\}} = \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a,b,c\}}{a \cdot b \cdot c} = \left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} = \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

## / (dividera)



Ger en matris över kvoterna av  
*Matrix1/Expr*.

**Obs:** Använd . / (punkt dividera) för att dividera ett uttryck med varje element.

## ^ (potens)



*Expr1 ^ Expr2*  $\Rightarrow$  *uttryck*

$$p^2 \quad 16$$

*List1 ^ List2*  $\Rightarrow$  *lista*

$$\{a, 2, c\}^{\{1, b, 3\}} \quad \{a, 2^b, c^3\}$$

Ger det första argumentet upphöjt till det andra argumentets potens.

**Obs:** Se även **Exponent template**, på sidan 1.

Ger, för en lista, elementen i *List1* upphöjda till potensen för motsvarande element i *List2*.

I det reella området använder potenser i bråkform som har reducerade exponenter med udda nämnare den reella delen kontra principaldelen för komplext läge.

*Expr ^ List1*  $\Rightarrow$  *lista*

$$p^{\{a, 2, -3\}} \quad \left\{p^a, p^2, \frac{1}{p^3}\right\}$$

Ger *Expr* upphöjt till potensen för elementen i *List1*.

*List1 ^ Expr*  $\Rightarrow$  *lista*

$$\{1, 2, 3, 4\}^{-2} \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Ger elementen i *List1* upphöjda till potensen för *Expr*.

*squareMatrix1 ^ integer*  $\Rightarrow$  *matris*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Ger *squareMatrix1* upphöjd till potensen för *integer* (heltal).

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

*squareMatrix1* måste vara en kvadratisk matris.

Om *integer* = -1 beräknas den inversa matrisen.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Om *integer* < -1 beräknas den inversa matrisen till en lämplig positiv potens.

**x<sup>2</sup> (kvadrat)** **tangent***Expr1*<sup>2</sup> ⇒ *uttryck*

|       |    |
|-------|----|
| $4^2$ | 16 |
|-------|----|

Ger kvadraten på argumentet.

|               |               |
|---------------|---------------|
| $\{2,4,6\}^2$ | $\{4,16,36\}$ |
|---------------|---------------|

*List1*<sup>2</sup> ⇒ *lista*

|                                                                       |                                                                                |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------|
| $\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$ | $\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$ |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------|

Ger en lista med kvadraterna på elementen i *List1*.*squareMatrix1*<sup>2</sup> ⇒ *matris*

|                                                                              |                                                                            |
|------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| $\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \wedge 2$ | $\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$ |
|------------------------------------------------------------------------------|----------------------------------------------------------------------------|

Ger en matris över kvadraten på *squareMatrix1*. Detta är inte detsamma som att beräkna kvadraten på varje element. Använd  $\wedge 2$  för att beräkna kvadraten på varje element.

**+. (punkt addera)**  **tangenter***Matrix1* .+ *Matrix2* ⇒ *matris*

|                                                                                                         |                                                      |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| $\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \text{ .+ } \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$ | $\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$ |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------|

*Expr* .+ *Matrix1* ⇒ *matris*

|                                                              |                                                        |
|--------------------------------------------------------------|--------------------------------------------------------|
| $x \text{ .+ } \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$ | $\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$ |
|--------------------------------------------------------------|--------------------------------------------------------|

*Matrix1* .+ *Matrix2* ger en matris som är summan av varje par av motsvarande element i *Matrix1* och *Matrix2*.

*Expr* .+ *Matrix1* ger en matris som är summan av *Expr* och varje element i *Matrix1*.

**.- (punkt subtrahera)**  **tangenter***Matrix1* .- *Matrix2* ⇒ *matrixs*

|                                                                                                         |                                                      |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| $\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \text{ .- } \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$ | $\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$ |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------|

*Expr* .- *Matrix1* ⇒ *matris*

|                                                              |                                                        |
|--------------------------------------------------------------|--------------------------------------------------------|
| $x \text{ .- } \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$ | $\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$ |
|--------------------------------------------------------------|--------------------------------------------------------|

*Matrix1* .- *Matrix2* ger en matris som är skillnaden mellan varje par av motsvarande element i *Matrix1* och *Matrix2*.

*Expr* .- *Matrix1* ger en matris som är skillnaden mellan *Expr* och varje element i *Matrix1*.

## · (punkt multiplicera)

☐ ☒ tangenter

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

$Expr \cdot Matrix1 \Rightarrow matrix$

$$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

$Matrix1 \cdot Matrix2$  ger en matris som är produkten av varje par av motsvarande element i  $Matrix1$  och  $Matrix2$ .

$Expr \cdot Matrix1$  ger en matris som innehåller produkterna av  $Expr$  och varje element i  $Matrix1$ .

## ./ (punkt dividera)

☐ ☒ tangenter

$Matrix1 ./ Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} ./ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

$Expr ./ Matrix1 \Rightarrow matrix$

$$\begin{bmatrix} x & x \\ c & 4 \\ 5 & d \end{bmatrix} ./ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

$Matrix1 ./ Matrix2$  ger en matris som är kvoten av varje par av motsvarande element i  $Matrix1$  och  $Matrix2$ .

$Expr ./ Matrix1$  ger en matris som är kvoten av  $Expr$  och varje element i  $Matrix1$ .

## .^ (punkt potens)

☐ ☒ tangenter

$Matrix1 .^ Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .^ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

$Expr .^ Matrix1 \Rightarrow matrix$

$$x .^ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

$Matrix1 .^ Matrix2$  ger en matris där varje element i  $Matrix2$  är exponenten för motsvarande element i  $Matrix1$ .

$Expr .^ Matrix1$  ger en matris där varje element i  $Matrix1$  är exponenten för  $Expr$ .

## -(negation)

☒ tangenter

$-Expr1 \Rightarrow uttryck$

$$-2.43 \quad -2.43$$

$-List1 \Rightarrow lista$

$$\{-1, 0.4, 1.2 \in 19\} \quad \{1, -0.4, -1.2 \in 19\}$$

$-Matrix1 \Rightarrow matrix$

$$-a \cdot -b \quad a \cdot b$$

**(-) tangent**

Ob11111111111111111111111111111111111111▶

**ctrl**  **tangenter**

|     |      |
|-----|------|
| 13% | 0.13 |
|-----|------|

---

|                      |                   |
|----------------------|-------------------|
| $\{\{1,10,100\}\}\%$ | $\{0.01,0.1,1.\}$ |
|----------------------|-------------------|

**= tangent**

EndFunc

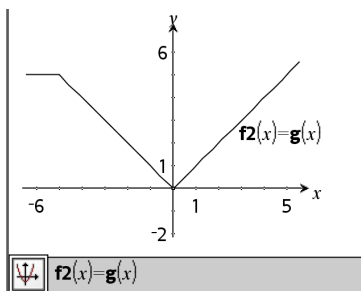
Symboler 225

## = (lika med)

 tangent

**Obs för att mata in exemplet:** Se avsnittet Räknare i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

Resultat från plottning av  $g(x)$



## ≠ (inte lika med)

  tangenter

$Expr1 \neq Expr2 \Rightarrow$  Booleskt uttryck

Se exemplet "=" (lika med).

$List1 \neq List2 \Rightarrow$  Boolesk lista

$Matrix1 \neq Matrix2 \Rightarrow$  Boolesk matris

Ger resultatet sant om  $Expr1$  bestäms inte vara lika med  $Expr2$ .

Ger resultatet falskt om  $Expr1$  bestäms vara lika med  $Expr2$ .

Allt annat ger en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva  $\neq$

## < (mindre än)

  tangenter

$Expr1 < Expr2 \Rightarrow$  Booleskt uttryck

Se exemplet "=" (lika med).

$List1 < List2 \Rightarrow$  Boolesk lista

$Matrix1 < Matrix2 \Rightarrow$  Boolesk matris

Ger resultatet sant om  $Expr1$  bestäms vara mindre än  $Expr2$ .

## < (mindre än)

  tangent

Ger resultatet falskt om *Expr1* bestäms vara större än eller lika med *Expr2*.

Allt annat ger en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

## ≤ (mindre än eller lika med)

  tangent

*Expr1* ≤ *Expr2* ⇒ Booleskt uttryck

Se exemplet “=” (lika med).

*List1* ≤ *List2* ⇒ Boolesk lista

*Matrix1* ≤ *Matrix2* ⇒ Boolesk matris

Ger resultatet sant om *Expr1* bestäms vara mindre än eller lika med *Expr2*.

Ger resultatet falskt om *Expr1* bestäms vara större än *Expr2*.

Allt annat ger en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva <=

## > (större än)

  tangent

*Expr1* > *Expr2* ⇒ Booleskt uttryck

Se exemplet “=” (lika med).

*List1* > *List2* ⇒ Boolesk lista

*Matrix1* > *Matrix2* ⇒ Boolesk matris

Ger resultatet sant om *Expr1* bestäms vara större än *Expr2*.

Ger resultatet falskt om *Expr1* bestäms vara mindre än eller lika med *Expr2*.

Allt annat ger en förenklad form av ekvationen.

## > (större än)

**ctrl** **=** **tangenten**

Ger, för listor och matriser, jämförelser  
element för element.

## ≥ (större än eller lika med)

**ctrl** **=** **tangenten**

$Expr1 \geq Expr2 \Rightarrow$  Booleskt uttryck

Se exemplet “=” (lika med).

$List1 \geq List2 \Rightarrow$  Boolesk lista

$Matrix1 \geq Matrix2 \Rightarrow$  Boolesk matris

Ger resultatet falskt om  $Expr1$  bestäms vara  
större än eller lika med  $Expr2$ .

Ger resultatet falskt om  $Expr1$  bestäms vara  
mindre än  $Expr2$ .

Allt annat ger en förenklad form av  
ekvationen.

Ger, för listor och matriser, jämförelser  
element för element.

**Obs:** Du kan infoga denna operator med  
datorns tangentbord genom att skriva  $\geq$

## $\Rightarrow$ (logisk implikation)

**ctrl** **=** **knappar**

$BoolesktUtr1 \Rightarrow BoolesktUtr2$  ger  
 $Booleskt$  uttryck

$5 > 3$  or  $3 > 5$  true

$5 > 3 \Rightarrow 3 > 5$  false

$BooleskList1 \Rightarrow BooleskList2$  ger  
 $Boolesk$  lista

$3$  or  $4$  7

$3 \Rightarrow 4$  -4

$BooleskMatris1 \Rightarrow BooleskMatris2$  ger  
 $Boolesk$  matris

$\{1,2,3\}$  or  $\{3,2,1\}$   $\{3,2,3\}$

$\{1,2,3\} \Rightarrow \{3,2,1\}$   $\{-1,-1,-3\}$

$Heltal1 \Rightarrow Heltal2$  ger  $Heltal$

Beräknar uttrycket **not**  $\langle argument1 \rangle$  or  
 $\langle argument2 \rangle$  och ger resultatet sant,  
falskt eller en förenklad form av  
ekvationen.

Ger, för listor och matriser, jämförelser  
element för element.



⇒ (logisk implikation)

ctrl = knappar

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva =>

⇔ (logisk dubbel implikation, XNOR)

ctrl = knappar

*BooleskUttr1* ⇔ *BooleskUttr2* ger  
*Boolesk uttryck*

|                            |      |
|----------------------------|------|
| $5 > 3 \text{ xor } 3 > 5$ | true |
|----------------------------|------|

|                               |       |
|-------------------------------|-------|
| $5 > 3 \Leftrightarrow 3 > 5$ | false |
|-------------------------------|-------|

*BooleskLista1* ⇔ *BooleskLista2* ger  
*Boolesk lista*

|                    |   |
|--------------------|---|
| $3 \text{ xor } 4$ | 7 |
|--------------------|---|

|                       |    |
|-----------------------|----|
| $3 \Leftrightarrow 4$ | -8 |
|-----------------------|----|

*BooleskMatris1* ⇔ *BooleskMatris2* ger  
*Boolesk matris*

|                                        |               |
|----------------------------------------|---------------|
| $\{1, 2, 3\} \text{ xor } \{3, 2, 1\}$ | $\{2, 0, 2\}$ |
|----------------------------------------|---------------|

|                                           |                  |
|-------------------------------------------|------------------|
| $\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$ | $\{-3, -1, -3\}$ |
|-------------------------------------------|------------------|

*Heltal1* ⇔ *Heltal2* ger *Heltal*

Ger negation av en **XOR** Boolesk operation på de två argumenten. Ger resultatet sant, falskt eller en förenklad form av ekvationen.

Ger, för listor och matriser, jämförelser element för element.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva <=>

! (fakultet)

?!> tangent

*Expr1!* ⇒ *uttryck*

|      |     |
|------|-----|
| $5!$ | 120 |
|------|-----|

*List1!* ⇒ *lista*

|                    |                  |
|--------------------|------------------|
| $\{\{5, 4, 3\}\}!$ | $\{120, 24, 6\}$ |
|--------------------|------------------|

*Matrix1!* ⇒ *matris*

|                                                 |                                                 |
|-------------------------------------------------|-------------------------------------------------|
| $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ | $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$ |
|-------------------------------------------------|-------------------------------------------------|

Ger argumentets fakultet.

Ger, för en lista eller matris, en lista eller matris med elementens fakulteter.

*String1* & *String2*  $\Rightarrow$  *sträng*

"Hello "&"Nick"

"Hello Nick"

Ger en textsträng som är *String2* bifogad till *String1*.

## d() (derivata)

Katalog >

**d**(*Uttr1*, *Var*[, *Ordning*])  $\Rightarrow$  *uttryck*

**d**(*Lista1*, *Var*[, *Ordning*])  $\Rightarrow$  *lista*

**d**(*Matris1*, *Var*[, *Ordning*])  $\Rightarrow$  *matris*

Ger förstaderivatan av det första argumentet med avseende på variabeln *Var*.

$$\begin{array}{l} \frac{d}{dx}(f(x) \cdot g(x)) \quad \frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x) \\ \frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right) \quad 6 \cdot y^2 \cdot x \\ \frac{d}{dx}\left(\left\{x^2, x^3, x^4\right\}\right) \quad \left\{2 \cdot x, 3 \cdot x^2, 4 \cdot x^3\right\} \end{array}$$

*Ordning*, om inkluderad, måste vara ett heltal. Om ordningen är mindre än noll blir resultatet en antiderivata.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **derivative (...)**.

**d()** följer inte den normala utvärderingsmekanismen för att fullt förenkla dess argument och sedan tillämpa funktionsdefinitionen på dessa fullt förenklade argument. I stället utför **d()** följande steg:

1. Förenklar det andra argumentet endast i sådan utsträckning att det inte leder till en icke-variabel.
2. Förenklar det första argumentet endast i sådan utsträckning att det inte hämtar något lagrat värde för den variabel som bestäms av steg 1.
3. Bestämmer den symboliska derivatan av resultatet från steg 2 med avseende på variabeln från steg 1.

Om variabeln från steg 1 har ett lagrat värde eller ett värde specificerat med ("|")-operatoren begränsning, substituera det värdet i resultatet från steg 3.

**Obs:** Se även **Förstaderivata**, på sidan 5,  
**Andraderivata**, på sidan 6 eller  
**N:te derivata**, på sidan 6.

**Obs:** Se även **Förstaderivata**, på sidan 5  
 eller **Andraderivata**, på sidan 6.

## f() (integrera)

f(Uttr1, Var[, Undre, Övre]) ⇒ uttryck

f(Uttr1, Var[, Konstant]) ⇒ uttryck

Ger integralen av *Uttr1* med avseende  
 på *Var* från *Undre* till *Övre*.

**Obs:** Se även **Mall för bestämd integral**  
 eller **Mall för obestämd integral**, på sidan  
 6.

**Obs:** Du kan infoga denna funktion med  
 datorns tangentbord genom att skriva  
**integral (...)**.

Om *Lower* och *Upper* utelämnas erhålls  
 en primitiv funktion. En symbolisk  
 integrationskonstant utelämnas såvida  
 du inte anger argumentet *Constant*.

Jämbördigt giltiga antiderivator kan skilja  
 med en numerisk konstant. En sådan  
 konstant kan vara maskerad, särskilt när  
 en antiderivata innehåller logaritmer  
 eller inversa trigonometriska funktioner.  
 Dessutom läggs ibland stegvisa  
 konstanta uttryck till för att göra en  
 antiderivata giltig över ett större  
 intervall än med den vanliga formeln.

f() returnerar sig själv för steg av *Expr1*  
 som inte kan bestämmas som en explicit  
 ändlig kombination av dess inbyggda  
 funktioner och operatorer.

När du anger *Lower* och *Upper* görs ett  
 försök att hitta eventuella  
 diskontinuiteter eller diskontinuerliga  
 derivator i intervallet *Lower* < *Var* <  
*Upper* för att dela upp intervallet vid  
 dessa ställen.

$$\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int x^2 dx = \frac{x^3}{3}$$

$$\int (a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

$$\int b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} dx = b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Med inställningen Auto i läge **Auto** eller **Ungefärlig** används numerisk integrering där så är tillämpligt när en antiderivata eller ett gränsvärde inte kan bestämmas.

Med inställningen Approximate görs först ett försök med numerisk integrering om detta är tillämpligt. Antiderivata söks endast där sådan numerisk integrering inte är tillämplig eller misslyckas.

**Obs:** För att få ett närmevärde,

**Handenhet:** Tryck på **ctrl** **enter**.

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

**∫()** kan kapslas in för att göra multipelintegraler. Integrationsgränser kan bero på integrationsvariabler utanför gränserna.

**Obs:** Se även **nInt()**, på sidan 128.

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

$$\int_0^a \int_0^x \ln(x+y) dy dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

**√()** (kvadratroten)

**ctrl** **x²** **tangenter**

**√ (Expr1)⇒uttryck**

**√ (List1)⇒lista**

Ger kvadratroten ur argumentet.

Ger, för en lista, kvadratrötterna ur alla element i *List1*.

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **sqrt (...)**

**Obs:** Se även **Square root template**, på sidan 1.

$$\sqrt{4} \quad 2$$

$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\}$$

## $\Pi()$ (prodSeq)

Katalog > 

$\Pi(\text{Expr1}, \text{Var}, \text{Low}, \text{High}) \Rightarrow \text{uttryck}$

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **prodSeq (...)**.

Utvärderar *Expr1* för varje värde på *Var* från *Low* till *High* och ger produkten av resultaten.

**Obs:** Se även **Product template** ( $\Pi$ ), på sidan 5.

$\Pi(\text{Expr1}, \text{Var}, \text{Low}, \text{Low}-1) \Rightarrow 1$

$\Pi(\text{Expr1}, \text{Var}, \text{Low}, \text{High}) \Rightarrow 1/\Pi(\text{Expr1}, \text{Var}, \text{High}+1, \text{Low}-1)$  om  $\text{High} < \text{Low}-1$

Produktformlerna som används har härletts från följande referens:

Ronald L. Graham, Donald E. Knuth och Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

|                                                                                       |                                           |
|---------------------------------------------------------------------------------------|-------------------------------------------|
| $\prod_{n=1}^5 \left(\frac{1}{n}\right)$                                              | $\frac{1}{120}$                           |
| $\prod_{k=1}^n (k^2)$                                                                 | $(n!)^2$                                  |
| $\prod_{n=1}^5 \left\{ \left\{ \frac{1}{n}, n, 2 \right\} \right\}$                   | $\left\{ \frac{1}{120}, 120, 32 \right\}$ |
| $\prod_{k=4}^3 (k)$                                                                   | 1                                         |
| $\prod_{k=4}^1 \left(\frac{1}{k}\right)$                                              | 6                                         |
| $\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right)$ | $\frac{1}{4}$                             |

## $\Sigma()$ (sumSeq)

Katalog > 

$\Sigma(\text{Expr1}, \text{Var}, \text{Low}, \text{High}) \Rightarrow \text{uttryck}$

**Obs:** Du kan infoga denna funktion med datorns tangentbord genom att skriva **sumSeq (...)**.

Utvärderar *Uttr1* för varje värde på *Var* från *Låg* till *Hög* och ger summan av resultaten.

**Obs:** Se även **Sum template**, på sidan 5.

$\Sigma(\text{Expr1}, \text{Var}, \text{Low}, \text{Low}-1) \Rightarrow 0$

$\Sigma(\text{Expr1}, \text{Var}, \text{Low}, \text{High}) \Rightarrow -\Sigma(\text{Expr1}, \text{Var}, \text{High}+1, \text{Low}-1)$  om  $\text{High} <$

|                                                  |                                               |
|--------------------------------------------------|-----------------------------------------------|
| $\sum_{n=1}^5 \left(\frac{1}{n}\right)$          | $\frac{137}{60}$                              |
| $\sum_{k=1}^n (k^2)$                             | $\frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$ |
| $\sum_{n=1}^{\infty} \left(\frac{1}{n^2}\right)$ | $\frac{\pi^2}{6}$                             |
| $\sum_{k=4}^3 (k)$                               | 0                                             |

Low-1

Summaformlerna som används har härletts från följande referens:

Ronald L. Graham, Donald E. Knuth och Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) = -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

## ΣInt()

ΣInt(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [roundValue])⇒värde

ΣInt(1,3,12,4,75,20000,,,12,12) -218.11

## ΣInt

(NPmt1, NPmt2, amortTable)⇒värde

Amorteringsfunktion som beräknar räntesumman under ett specificerat område av betalningar.

NPmt1 och NPmt2 definierar start och slut på betalningsområdet.

N, I, PV, Pmt, FV, PpY, CpY och PmtAt beskrivs i tabellen över TVM-argument, se på sidan 203.

- Om du utelämnar Pmt används förinställningen Pmt=tvmPmt(N,I,PV,FV,PpY,CpY,PmtAt).
- Om du utelämnar FV används förinställningen FV=0.
- Förinställningarna av PpY, CpY och PmtAt är desamma som för TVM-funktionerna.

roundValue anger antalet decimaler för avrundning. Förinställning: 2.

| tbl:=amortTbl(12,12,4,75,20000,,,12,12) |        |          |         |         |
|-----------------------------------------|--------|----------|---------|---------|
|                                         | 0      | 0.       | 0.      | 20000.  |
| 1                                       | -79.17 | -1630.69 | 18369.3 |         |
| 2                                       | -72.71 | -1637.15 | 16732.2 |         |
| 3                                       | -66.23 | -1643.63 | 15088.5 |         |
| 4                                       | -59.73 | -1650.13 | 13438.4 |         |
| 5                                       | -53.19 | -1656.67 | 11781.7 |         |
| 6                                       | -46.64 | -1663.22 | 10118.5 |         |
| 7                                       | -40.05 | -1669.81 | 8448.7  |         |
| 8                                       | -33.44 | -1676.42 | 6772.28 |         |
| 9                                       | -26.81 | -1683.05 | 5089.23 |         |
| 10                                      | -20.14 | -1689.72 | 3399.51 |         |
| 11                                      | -13.46 | -1696.4  | 1703.11 |         |
| 12                                      | -6.74  | -1703.12 | -0.01   |         |
| ΣInt(1,3,tbl)                           |        |          |         |         |
|                                         |        |          |         | -218.11 |

ΣInt(*NPmt1*,*NPmt2*,*amortTable*)

beräknar räntesumman baserat på amorteringstabellen *amortTable*.

Argumentet *amortTable* måste vara en matris i den form som beskrivs under **amortTbl()**, på sidan 8.

**Obs:** Se även ΣPrn() nedan och **Bal()**, på sidan 17.

## ΣPrn()

ΣPrn(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*roundValue*])⇒värde

ΣPrn(1,3,12,4.75,20000,,,12,12) -4911.47

## ΣPrn

(*NPmt1*,*NPmt2*,*amortTable*)⇒värde

Amorteringsfunktion som beräknar kapitalsumman under ett specificerat område av betalningar.

*NPmt1* och *NPmt2* definierar start och slut på betalningsområdet.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* och *PmtAt* beskrivs i tabellen över TVM-argument, se på sidan 203.

- Om du utelämnar *Pmt* används förinställningen *Pmt=tvmpmt* (*N*,*I*,*PV*,*FV*,*PpY*,*CpY*,*PmtAt*).
- Om du utelämnar *FV* används förinställningen *FV*=0.
- Förinställningarna av *PpY*, *CpY* och *PmtAt* är desamma som för TVM-funktionerna.

*roundValue* anger antalet decimaler för avrundning. Förinställning: 2.

ΣPrn(*NPmt1*,*NPmt2*,*amortTable*)

beräknar summan av betalt kapital baserat på amorteringstabellen *amortTable*. Argumentet *amortTable* måste vara en matris i den form som beskrivs under **amortTbl()**, på sidan 8.

*tbl:=amortTbl*(12,12,4.75,20000,,,12,12)

|    |        |          |         |
|----|--------|----------|---------|
| 0  | 0.     | 0.       | 20000.  |
| 1  | -79.17 | -1630.69 | 18369.3 |
| 2  | -72.71 | -1637.15 | 16732.2 |
| 3  | -66.23 | -1643.63 | 15088.5 |
| 4  | -59.73 | -1650.13 | 13438.4 |
| 5  | -53.19 | -1656.67 | 11781.7 |
| 6  | -46.64 | -1663.22 | 10118.5 |
| 7  | -40.05 | -1669.81 | 8448.7  |
| 8  | -33.44 | -1676.42 | 6772.28 |
| 9  | -26.81 | -1683.05 | 5089.23 |
| 10 | -20.14 | -1689.72 | 3399.51 |
| 11 | -13.46 | -1696.4  | 1703.11 |
| 12 | -6.74  | -1703.12 | -0.01   |

ΣPrn(1,3,*tbl*) -4911.47

**Obs:** Se även ΣInt() ovan och Bal(), på sidan 17.

## # (indirection)

ctrl  tangenter

# varNameString

#{"x"&amp;"y"&amp;"z"} xyz

Avser variabeln vars namn är *varNameString*. Detta låter dig använda strängar för att skapa variabelnamn inom en funktion.

Skapar eller avser variabeln xyz.

10 → r 10

"r" → s1 "r"

#s1 10

Ger värdet på variabeln (r) vars namn är lagrat i variabeln s1.

## E (grundpotensform)

 tangent

mantissaEexponent

23000. 23000.

Skriver in ett tal i grundpotensform. Talet tolkas som *mantissa* × 10<sup>exponent</sup>.

2300000000.+4.1E15 4.1E15

3·10<sup>4</sup> 30000

Tips: Om du vill skriva in en potens av 10 utan att orsaka ett decimalt resultat, använd 10<sup>integer</sup>.

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva @E. Skriv exempelvis 2.3@E4 för att mata in 2.3E4.

## g (nygrad)

 tangent

Expr1g⇒uttryck

I vinkelläge Grader, Nygrader eller Radianer:

List1g⇒lista

cos(50°)  $\frac{\sqrt{2}}{2}$ 

Matrix1g⇒matrix

cos({0,100°,200°}) {1,0,-1}

Denna funktion ger dig ett sätt att specificera en vinkel i nygrader när du är i läge Grader eller Radianer.



**g (nygrad)****1 tangent**

Multipliserar i vinkelläget Radianer  
*Expr1* med  $\pi/200$ .

Multipliserar i vinkelläget Grader *Expr1*  
 med  $g/100$ .

Ger i läget Nygrader *Expr1* oförändrat.

**Obs:** Du kan infoga denna symbol med datorns tangentbord genom att skriva @g.

**r (radian)****1 tangent**

*Expr1*<sup>r</sup> ⇒ *uttryck*

*List1*<sup>r</sup> ⇒ *lista*

*Matrix1*<sup>r</sup> ⇒ *matrix*

Denna funktion ger dig ett sätt att specificera en vinkel i radianer när du är i läge Grader eller Nygrader.

Multipliserar i vinkelläget Grader argumentet med  $180/\pi$ .

Ger i vinkelläget Radianer argumentet oförändrat.

Multipliserar i vinkelläget Nygrader argumentet med  $200/\pi$ .

Tips: Använd <sup>r</sup> om du vill framtvinga radianer i en funktionsdefinition oavsett det inställda läget när funktionen används.

**Obs:** Du kan infoga denna symbol med datorns tangentbord genom att skriva @r.

I vinkelläge Grader, Nygrader eller Radianer:

$$\cos\left(\frac{\pi}{4^r}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos\left(\left\{0^r, \frac{\pi}{12}, \frac{\pi}{12}, \left(\frac{\pi}{12}\right)^r\right\}\right) \quad \left\{1, \frac{(\sqrt{3}+1)\sqrt{2}}{4}, -1\right\}$$

**° (grader)****1 tangent**

*Expr1*<sup>°</sup> ⇒ *uttryck*

*List1*<sup>°</sup> ⇒ *lista*

*Matrix1*<sup>°</sup> ⇒ *matrix*

I vinkelläge Grader, Nygrader eller Radianer:

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

## ° (grader)

## 1 tangent

Denna funktion ger dig ett sätt att specificera en vinkel i grader när du är i läge Nygrader eller Radianer.

Multipliserar i vinkelläget Radianer argumentet med  $\pi/180$ .

Ger i vinkelläget Grader argumentet oförändrat.

Multipliserar i vinkelläget Nygrader argumentet med 10/9.

**Obs:** Du kan infoga denna symbol med datorns tangentbord genom att skriva @d.

I vinkelläget Radianer:

**Obs:** För att få ett närmvärde,

**Handenhet:** Tryck på .

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

$$\cos\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}$$

$$\{1, 0.707107, 0, 0.864976\}$$

## °, ', " (grad/minut/sekund)

## ctrl tangenter

*dd°mm'ss.ss"⇒uttryck*

*dd*Ett positivt eller negativt tal

*mm*Ett icke negativt tal

*ss.ss*Ett icke negativt tal

Ger  $dd+(mm/60)+(ss.ss/3600)$ .

Med detta bas-60 inmatningsformat kan du:

- Skriva in en vinkel i grader/minuter/sekunder oavsett det inställda vinkelläget.
- Skriva in tid som timmar/minuter/sekunder.

**Obs:** Avsluta ss.ss med två apostrofer ("), inte med citationstecken (').

I vinkelläget Grader:

|             |                |
|-------------|----------------|
| 25°13'17.5" | 25.2215        |
| 25°30'      | $\frac{51}{2}$ |

## ∠ (vinkel)

## ctrl tangenter

*[Radius, ∠θ\_Angle]⇒vektor*

(polär indata)

*[Radius, ∠θ\_Angle, Z\_Coordinate]⇒vektor*

I vinkelläget Radianer och med vektorformatet inställt på:

rektangulär

## ∠ (vinkel)

  **tangenter**

(cylindrisk indata)

$$\left[ 5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[ \frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

[Radius, ∠θ\_Angle, ∠θ\_Angle] ⇒ vektor

(sfärisk indata)

Ger koordinater som en vektor beroende på det inställda vektorformatläget: rektangulär, cylindrisk eller sfärisk.

cylindrisk

$$\left[ 5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[ \frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

**Obs:** Du kan infoga denna symbol med datorns tangentbord genom att skriva @<.

sfärisk

$$\left[ 5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[ 5 \quad \angle \frac{\pi}{3} \quad \angle \frac{\pi}{4} \right]$$

(Magnitude ∠ Angle) ⇒ complexValue

(polär indata)

Matar in ett komplext värde i (r∠θ) polär form. Angle tolkas enligt det inställda vinkelläget.

I vinkelläget Radianer och i Rektangulärt komplext format:

$$5+3 \cdot i - \left( 10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

**Obs:** För att få ett närmevärde,

**Handenhet:** Tryck på  .

**Windows®:** Tryck på **Ctrl+Enter**.

**Macintosh®:** Tryck på **⌘+Enter**.

**iPad®:** Håll ned **enter** och välj .

$$5+3 \cdot i - \left( 10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107 \cdot i$$

## ' (prim)

 **tangent**

variable '

variable ''

Skriver in en primsymbol i en differentialekvation. En enda primsymbol betecknar en differentialekvation av första ordningen, två primsymboler betecknar en ekvation av andra ordningen, och så vidare.

$$\text{deSolve} \left( y'' = y^{\frac{-1}{2}} \text{ and } y(0) = 0 \text{ and } y'(0) = 0, t, y \right)$$
$$\frac{2 \cdot y^{\frac{4}{3}}}{3} = t$$

**\_ (understrykningstecken som ett tomt element)**

Se "Tomma element" (på sidan 261).

**\_ (understrykningstecken som enhetsbenämnnare)**

  **tangenten**

*Expr\_Unit*

3·\_m►\_ft

9.84252·\_ft

Anger enheterna för ett *Expr*. Alla enhetsnamn måste börja med en understrykning.

Du kan använda fördefinierade enheter eller skapa dina egna enheter. För en lista på fördefinierade enheter, öppna Katalogen och ta fram fliken Omvandling av enheter. Du kan välja enhetsnamn från Katalogen eller skriva in enhetsnamnen direkt.

*Variable\_*

Förutsatt att  $z$  är odefinierad:

När *Variable* inte har något värde behandlas den som om den representerar ett komplext tal. Som förinställning behandlas variabeln som reell utan \_.

|                    |                    |
|--------------------|--------------------|
| $\text{real}(z)$   | $z$                |
| $\text{real}(z\_)$ | $\text{real}(z\_)$ |
| $\text{imag}(z)$   | 0                  |
| $\text{imag}(z\_)$ | $\text{imag}(z\_)$ |

Om *Variable* har ett värde ignoreras \_ och *Variable* behåller dess ursprungliga datatyp.

**Obs:** Du kan lagra ett komplext tal i en variabel utan att använda \_. För bästa resultat i beräkningar såsom **cSolve()** och **cZeros()** rekommenderas dock \_.

**► (konvertera)**

  **tangenten**

*Expr\_Unit1 ►\_Unit2⇒Expr\_Unit2*

3·\_m►\_ft

9.84252·\_ft

Omvandlar ett uttryck från en enhet till en annan.

Understrykningstecknet \_ betecknar enheterna. Enheterna måste tillhöra samma kategori, till exempel, Längd eller Area.

## ► (konvertera)

  **tangent**

För en lista på fördefinierade enheter, öppna Katalogen och ta fram fliken Omvandling av enheter:

- Du kan välja ett enhetsnamn på listan.
- Du kan välja omvandlingsoperator, ►, längst upp i listan.

Du kan också skriva in enhetsnamnen manuellt. För att skriva “\_” när du skriver enhetsnamn på handenheten, tryck på

 .

**Obs:** För att konvertera temperaturenheter, använd **tmpCnv()** och **ΔtmpCnv()**. Konverteringsoperatoren ► hanterar inte temperaturenheter.

## 10^()

**Katalog >** 

**10^ (Expr1) ⇒ uttryck**

$10^{1.5}$  31.6228

**10^ (List1) ⇒ lista**

$10^{\{0, -2.2, a\}}$   $\left\{1, \frac{1}{100}, 100, 10^a\right\}$

Ger 10 upphöjd till argumentets potens.

Ger, för en lista, 10 upphöjd till potensen för elementen i List1.

**10^ (squareMatrix1) ⇒ kvadratMatris**

Ger 10 upphöjd till potensen för squareMatrix1. Detta är inte detsamma som att beräkna 10 upphöjd till potensen för varje element. Se **cos()** för information om beräkningsmetoden.

$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}}$

|           |           |           |
|-----------|-----------|-----------|
| 1.14336E7 | 8.17155E6 | 6.67589E6 |
| 9.95651E6 | 7.11587E6 | 5.81342E6 |
| 7.65298E6 | 5.46952E6 | 4.46845E6 |

squareMatrix1 måste vara möjlig att diagonalisera. Resultatet visas alltid i flyttalsform.

## ^-1 (inverterat värde)

**Katalog >** 

**Expr1 ^-1 ⇒ uttryck**

$(3.1)^{-1}$  0.322581

**List1 ^-1 ⇒ lista**

$\{a, 4, -0.1, x, -2\}^{-1}$   $\left\{\frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, -\frac{1}{2}\right\}$

Ger argumentets inverterade värde.

Ger, för en lista, de inverterade värdena på elementen i *List1*.

*squareMatrix1*  $\wedge^{-1} \Rightarrow$  *kvadratMatrix*

Ger inversen av *squareMatrix1*.

*squareMatrix1* måste vara en icke singulär kvadratisk matris.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

## | (operatör begränsning)

**tangenter**

*Uttr* | *BoolesktUttr1*  
[*andBoolesktUttr2*]...

$$\begin{array}{ll} x+1|x=3 & 4 \\ x+y|x=\sin(y) & \sin(y)+y \\ x+y|\sin(y)=x & x+y \end{array}$$

*Uttr* | *BoolesktUttr1*  
[*orBoolesktUttr2*]...

("|")-symbolen begränsning fungerar som en binär operator. Operanden till vänster om | är ett uttryck. Operanden till höger om | specificerar ett eller flera relationer som är avsedda att påverka förenklingen av uttrycket. Flera relationer efter | måste förbindas med ett logiskt "and" eller "or"-operatorer.

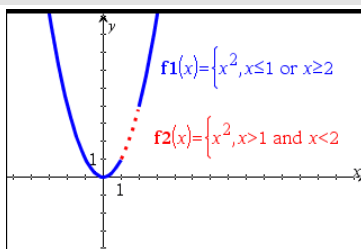
Operatör begränsning ger tre bastyper av funktionalitet:

- Substitutioner
- Intervallbegränsningar
- Uteslutningar

Substitutioner är i form av en likhet såsom  $x=3$  eller  $y=\sin(x)$ . För bästa effektivitet bör den vänstra sidan vara en enkel variabel. *Uttr* | *Variabel* = *värde* ersätter *värde* vid varje förekomst av *Variabel* i *Uttr*.

Intervallbegränsningar tar formen av en eller flera olikheter som förbinds med logiska "and" eller "or"-operatorer. Intervallbegränsningar medger också förenklingar som annars kan vara ogiltiga eller ej beräkningsbara.

$$\begin{array}{ll} x^3-2 \cdot x+7 \rightarrow f(x) & \text{Done} \\ f(x)|x=\sqrt{3} & \sqrt{3}+7 \\ (\sin(x))^2+2 \cdot \sin(x)-6|\sin(x)=d & d^2+2 \cdot d-6 \\ \text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2 & x=1 \\ \sqrt{x} \cdot \frac{1}{x}|x>0 & 1 \\ \sqrt{x} \cdot \frac{1}{x} & \sqrt{\frac{1}{x}} \cdot \sqrt{x} \end{array}$$



Uteslutningar använder jämförelseoperatörn "inte lika med" ( $\neq$  eller  $\neq$ ) för att utesluta ett specifikt värde från övervägning. De används främst för att utesluta en exakt lösning när t.ex. **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()** **zeros()** osv. används.

$$\text{solve}(x^2 - 1 = 0, x) | x \neq 1 \quad x = -1$$

## → (lagra)

ctrl **var** tangent

*Expr* → *Var*

*List* → *Var*

*Matrix* → *Var*

*Expr* → *Function*(*Param1*,...)

*List* → *Function*(*Param1*,...)

*Matrix* → *Function*(*Param1*,...)

Om variabeln *Var* inte finns så skapas den och initialiseras till *Expr*, *List* eller *Matrix*.

Om variabeln *Var* redan finns, och inte är låst eller skyddad, ersätts dess innehåll med *Expr*, *List* eller *Matrix*.

Tips: Om du tänker göra symboliska beräkningar med odefinierade variabler, undvik att lagra någonting i enbokstaviga variabler som ofta används, till exempel, a, b, c, x, y, z, och så vidare.

|                                                                                |                                                        |
|--------------------------------------------------------------------------------|--------------------------------------------------------|
| $\frac{\pi}{4} \rightarrow \text{myvar}$                                       | $\frac{\pi}{4}$                                        |
| $2 \cdot \cos(x) \rightarrow y1(x)$                                            | Done                                                   |
| $\{1, 2, 3, 4\} \rightarrow \text{lst5}$                                       | $\{1, 2, 3, 4\}$                                       |
| $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg}$ | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ |
| "Hello" → <i>str1</i>                                                          | "Hello"                                                |

## → (lagra)

**ctrl** **var** tangent

**Obs:** Du kan infoga denna operator med datorns tangentbord genom att skriva **=:** som kortkommando. Skriv exempelvis **pi/4 =: minvar**.

## := (tilldela)

**ctrl** **={}** tangenter

*Var := Expr*

*Var := List*

*Var := Matrix*

*Function(Param1,...) := Expr*

*Function(Param1,...) := List*

*Function(Param1,...) := Matrix*

Om variabeln *Var* inte finns så skapas *Var* och initialiseras till *Expr*, *List* eller *Matrix*.

Om *Var* redan finns, och inte är låst eller skyddad, ersätts dess innehåll med *Expr*, *List* eller *Matrix*.

Tips: Om du tänker göra symboliska beräkningar med odefinierade variabler, undvik att lagra någonting i enbokstaviga variabler som ofta används, till exempel, a, b, c, x, y, z, och så vidare.

|                                                                      |                                                        |
|----------------------------------------------------------------------|--------------------------------------------------------|
| <i>myvar:=</i> $\frac{\pi}{4}$                                       | $\frac{\pi}{4}$                                        |
| <i>y1(x):=</i> $2 \cdot \cos(x)$                                     | <i>Done</i>                                            |
| <i>lst5:=</i> {1,2,3,4}                                              | {1,2,3,4}                                              |
| <i>matg:=</i> $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ | $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ |
| <i>str1:=</i> "Hello"                                                | "Hello"                                                |

## © (kommentar)

**ctrl** **Ⓢ** tangenter

© [*text*]

© hanterar *text* som en kommentarsrad så att du kan kommentera funktioner och program som du skapar.

© kan vara i början eller var som helst på raden. Allting till höger om ©, till slutet av raden, utgör kommentaren.

**Obs för att mata in exemplet:** Se avsnittet Räkna i produkthandboken för instruktioner om hur du anger multiline-program och funktionsdefinitioner.

|                                                   |             |
|---------------------------------------------------|-------------|
| Define $g(n)$ =Func                               |             |
| © Declare variables                               |             |
| Local <i>i,result</i>                             |             |
| <i>result:=0</i>                                  |             |
| For <i>i</i> ,1, <i>n</i> ,1 ©Loop <i>n</i> times |             |
| <i>result:=result+i</i> <sup>2</sup>              |             |
| EndFor                                            |             |
| Return <i>result</i>                              |             |
| EndFunc                                           |             |
| <i>g(3)</i>                                       | <i>Done</i> |
|                                                   | 14          |



**0b** *binaryNumber*

I decimalt basläge:

**0h** *hexadecimalNumber*

|             |    |
|-------------|----|
| 0b10+0hF+10 | 27 |
|-------------|----|

Betecknar ett binärt respektive ett hexadecimalt tal. För att skriva in ett binärt eller hexadecimalt tal måste du använda prefixet 0b eller 0h oavsett det inställda basläget. Utan prefix behandlas ett tal som ett decimalt tal (bas 10).

I binärt basläge:

|             |         |
|-------------|---------|
| 0b10+0hF+10 | 0b11011 |
|-------------|---------|

Resultaten visas enligt det inställda basläget.

I hexadecimalt basläge:

|             |      |
|-------------|------|
| 0b10+0hF+10 | 0h1B |
|-------------|------|

# TI-Nspire™ CX II-Kommandon för att rita

Detta är ett kompletterande dokument för TI-Nspire™-referensguide och TI-Nspire™ CAS-referensguide. Alla TI-Nspire™ CX II-kommandon kommer att inkorporeras och publiceras i version 5.1 av TI-Nspire™-referensguide och TI-Nspire™ CAS-referensguide.

## Grafikprogrammering

Nya kommandon för grafikprogrammering har lagts till för TI-Nspire™ CX II-handenheter och TI-Nspire™-datorprogramvara.

TI-Nspire™ CX II-handenheterna kommer att byta till detta grafikläge samtidigt som de kör grafikkommandon och byter tillbaka till kontexten i vilken programmet körs efter slutförande av programmet.

"Kör..." kommer att visas i skärmens övre del medan programmet körs. "Avslutad" kommer att visas när programmet slutförs. Valfri knapptryckning kommer att få systemet att gå ut ur grafikläget.

- Övergången till grafikläge triggas automatiskt när en av kommandona för Rita (grafik) påträffas under körning av TI-Basicprogrammet.
- Denna övergång kommer endast att ske när ett program körs från Räknare-appen i ett dokument eller från Beräkna i Scratchpad.
- Övergången ut ur grafikläge sker vid avslutande av programmet.
- Grafikläget är endast tillgängligt på TI-Nspire™ CX II-handenheter och handenhetsvyn på TI-Nspire™ CX II CAS-programvara. Detta innebär att det inte är tillgängligt i datordokumentsvyn på datorn eller på iOS.
  - Om ett grafikkommando påträffas medan ett TI-Basic-program körs från inkorrekt kontext så kommer ett felmeddelande att visas och TI-Basic-programmet avslutas.

## Grafikskärm

Grafikskärmen kommer att innehålla en rubrik på skärmens övre del som inte kan åstadkommas av grafikkommandon.

Grafikskärmens ritområde kommer att rensas (färg = 255,255,255) när grafikskärmen initialiseras.

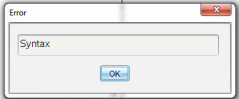
| Grafikskärm | Förval           |
|-------------|------------------|
| Höjd        | 212              |
| Bredd       | 318              |
| Färg        | vit: 255,255,255 |

## Standardvy och inställningar

- Statusikonerna i skärmens övre del (batteristatus, tryck-för-test-status, nätverksindikator osv.) kommer inte att synas när ett grafikprogram körs.
- Standardfärg för att rita: Svart (0,0,0)
- Standardstil på penna - normal, mjuk
  - Tjocklek: 1 (tunn), 2 (normal), 3 (tjockast)
  - Stil 1 (mjuk), 2 (prickad), 3 (streckad)
- Alla kommandon för att rita kommer att använda nuvarande färg och penninställningar; antingen standardvärden eller de som ställts in via TI-Basic-kommandon.
- Typsnitt är bestämt och kan inte ändras.
- Varje utmatning till grafikskärmen kommer att ritas inom ett klippfönster som är storleken av grafikskärmens ritområde. Varje ritad utmatning som sträcker sig utanför detta klippta ritområde för grafikskärmen kommer inte att ritas. Inget felmeddelande kommer att visas.
- Alla x- och y-koordinater specificerade för kommandon för att rita är definierade på så sätt att 0,0 är det vänstra övre hörnet på ritområdet för grafikskärmen.
  - **Undantag:**
    - **DrawText** använder koordinaterna i nedre vänstra hörnet av begränsningsrutan för texten.
    - **SetWindow** använder koordinaterna i skärmens nedre vänstra hörn
- Alla parametrar för kommandona kan ges som uttryck som värderas till ett värde som sedan avrundas till närmsta heltal.

# Felmeddelanden på grafikskärmen

Om valideringen misslyckas så kommer ett felmeddelande att visas.

| Felmeddelande                               | Beskrivning                                                                                                                                       | Visa                                                                                                                                                            |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fel Syntax                                  | Om syntaxkontrollen hittar syntaxfel visar den ett felmeddelande och försöker placera markören nära det första felet så att du kan korrigera det. |                                                                                |
| Fel Too few arguments (För få argument)     | Funktionen eller kommandot saknar ett eller flera argument                                                                                        | <div><b>Error</b><br/>Too few arguments<br/>The function or command is missing one or more arguments.<br/><div>OK</div></div>                                   |
| Fel Too many arguments (För många argument) | Funktionen eller kommandot innehåller för många argument och kan inte utvärderas.                                                                 | <div><b>Error</b><br/>Too many arguments<br/>The function or command contains an excessive number of arguments and cannot be evaluated.<br/><div>OK</div></div> |
| Fel Invalid data type (Ogiltig datatyp)     | Ett argument är av fel datatyp.                                                                                                                   | <div><b>Error</b><br/>Invalid data type<br/>An argument is of the wrong data type.<br/><div>OK</div></div>                                                      |

## Ogiltiga kommandon i grafikläge

Vissa kommandon är inte tillåtna då programmet växlat till grafikläge. Om dessa kommandon påträffas i grafikläge så kommer ett fel visas och programmet kommer att avslutas.

| Otillåtet kommando | Felmeddelande                            |
|--------------------|------------------------------------------|
| Begär              | Förfrågan kan inte utföras i grafikläge  |
| BegärStr           | RequestStr kan inte utföras i grafikläge |
| Text               | Text kan inte utföras i grafikläge       |

Kommandona som skriver ut text till Räknare-appen - **disp** och **dispAt** - kommer att stödja kommandon i grafikkontexten. Texten från dessa kommandon kommer att skickas till Räknare-skärmen (inte på Grafik) och kommer att synas efter att programmet avslutas och systemet byter tillbaka till Räknare-appen

## Rensa

Katalog >   
CXII**Rensa  $x$ ,  $y$ , bredd, höjd**

Rensar hela skärmen om inga parametrar är specificerade.

Om  $x$ ,  $y$ , *bredd* och *höjd* är specificerade så kommer rektangeln definierad av parametrarna att rensas.

Rensa

Rensar hela skärmen

Rensa 10,10,100,50

Rensar ett rektangelområde med övre vänster hörn (10, 10) och med bredd 100, höjd 50

## DrawArc

Katalog >   
CXII**DrawArc** *x, y, bredd, höjd, startAngle, arcAngle*

Rita en båge inom den definierade avgränsande rektangeln med de tillhandahållna start- och bågvinklarna.

*x, y*: övre vänstra koordinaten i avgränsande rektangel

*bredd, höjd*: dimensioner i avgränsande rektangel

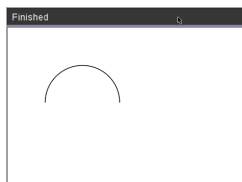
"Bågvinkeln" definierar bågens kurva.

Dessa parametrar kan ges som uttryck som värderas till ett värde som sedan avrundas till närmsta heltal.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se även: [FillArc](#)

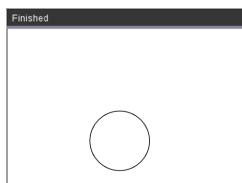
## DrawCircle

Katalog >   
CXII**DrawCircle** *x, y, radie*

*x, y*: koordinat i mittpunkt

*radie*: cirkelns radie

DrawCircle 150,150,40



Se även: [FillCircle](#)

## DrawLine

Katalog >   
CXII

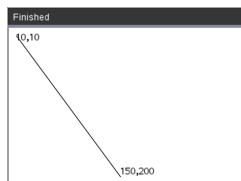
**DrawLine**  $x1, y1, x2, y2$

Rita en linje från  $x1, y1, x2, y2$ .

Uttryck som värderas till ett värde som sedan avrundas till närmsta heltal.

**Skärmgränser:** Om de specificerade koordinaterna orsakar att någon del av linjen ritas utanför grafikskärmen så kommer den delen av linjen att klippas av och inget felmeddelande kommer att visas.

DrawLine 10,10,150,200



## DrawPoly

Katalog >   
CXII

Kommandona har två varianter:

**DrawPoly**  $xlist, ylist$

eller

**DrawPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$

**Obs:** DrawPoly  $xlist, ylist$

Form kommer att binda samman  $x1, y1$  to  $x2, y2, x2, y2$  till  $x3, y3$  och så vidare.

**Obs:** DrawPoly  $x1, y1, x2, y2, x3, y3...xn, yn$  kommer **INTE** automatiskt att bindas samman till  $x1, y1$ .

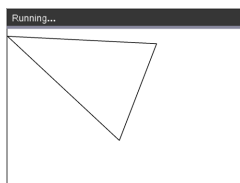
Uttryck som utvärderas till en lista med reella flyttal  
 $xlist, ylist$

Uttryck som utvärderas till ett reellt flyttal  
 $x1, y1...xn, yn$  = koordinater för polygonens hörn

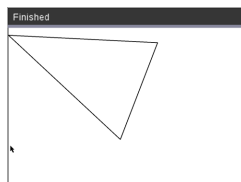
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly  $xlist, ylist$



DrawPoly 0,10,200,20,150,150,0,10



**Obs: DrawPoly:** Inmatningens storleksdimensioner (bredd/höjd) i förhållande till ritade linjer. Dessa linjer är ritade i en begränsningsruta runt de specificerade koordinaterna och dimensionerna på så sätt att den faktiska storleken på den ritade polygonen kommer att vara större än bredden och höjden.

Se även: [FillPoly](#)

## DrawRect

**DrawRect** *x, y, bredd, höjd*

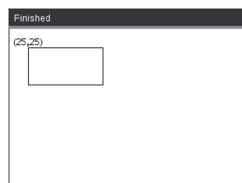
*x, y*: övre vänstra koordinaten i rektangeln

*bredd, höjd*: rektangelns bredd och höjd (rektangel ritad nedåt och höger från startkoordinaten).

**Obs:** Dessa linjer är ritade i en begränsningsruta runt de specificerade koordinaterna och dimensionerna på så sätt att den faktiska storleken på den ritade rektangeln kommer att vara större än vad bredden och höjden indikerar.

Se även: [FillRect](#)

DrawRect 25,25,100,50



## DrawText

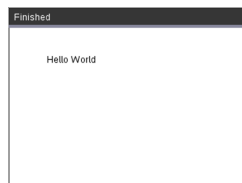
**DrawText** *x, y, exprOrString1*  
*[,exprOrString2]...*

*x, y*: koordinat för textutmatning

Ritar texten i *exprOrString* vid specificerad *x, y*-koordinatplats.

Reglerna för *exprOrString* är samma som för **Disp** - **DrawText** kan ta flera argument.

DrawText 50,50,"Hej världen"





## FillArc

Katalog >   
CXII**FillArc** *x, y, bredd, höjd, startVinkel, bågeVinkel*

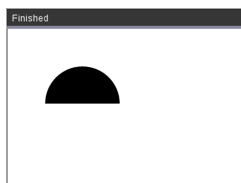
*x, y*: övre vänstra koordinaten i avgränsande rektangel

Rita och fyll en båge inom den definierade avgränsande rektangeln med de tillhandahållna start- och bågvinklarna.

Standardfyllningsfärgen är svart.  
Fyllningsfärgen kan ställas in med [SetColor](#)-kommandot

"Bågvinkeln" definierar bågens kurva

FillArc 50,50,100,100,0,180



## FillCircle

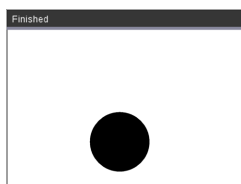
Katalog >   
CXII**FillCircle** *x, y, radie*

*x, y*: koordinat i mittpunkt

Rita och fyll en cirkel vid specificerad mittpunkt med den specificerade radien.

Standardfyllningsfärgen är svart.  
Fyllningsfärgen kan ställas in med [SetColor](#)-kommandot.

FillCircle150,150,40



Här!

## FillPoly

Katalog >   
CXII**FillPoly** *xlist, ylist*

eller

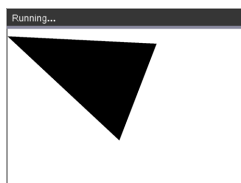
**FillPoly** *x1, y1, x2, y2, x3, y3...xn, yn*

**Obs:** Linjen och färgen specificeras av [StällInFärg](#) och [StällInPenna](#)

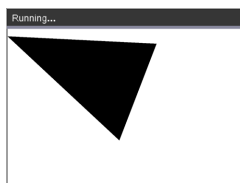
xlist:={0,200,150,0}

ylist:={10,20,150,10}

FillPoly xlist, ylist



FillPoly 0,10,200,20,150,150,0,10



## FillRect

FillRect  $x, y$ , bredd, höjd

FillRect 25,25,100,50

 $x, y$ : övre vänstra koordinaten i rektangeln*bredd, höjd*: rektangelns bredd och höjdRita och fyll en rektangel med övre vänstra hörnet vid koordinaten specificerad av  $(x,y)$ 

Standardfyllningsfärgen är svart.

Fyllningsfärgen kan ställas in med [SetColor](#)-kommandot**Obs:** Linjen och färgen specificeras av [StällInFärg](#) och [StällInPenna](#)

**getPlatform()**Katalog >   
**CXII****getPlatform()**

getPlatform()

"dt"

Ger:

"dt" på applikationer för  
datorprogramvara

"hh" på TI-Nspire™ CX-handenheter

"ios" på TI-Nspire™ CX iPad®-app

## PaintBuffer

Katalog >   
CXII

## PaintBuffer

Färggrafikbuffert till skärm

Detta kommando används i samband med UseBuffer för att öka hastigheten på skärmens display när programmet genererar flera grafiska objekt.

## UseBuffer

För n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radie:=randInt(10,50)

Wait 0,5

XDrawCircle x,y,radie

EndFor

## PaintBuffer

Detta program kommer att visa alla 10 cirklar på en gång.

Om "UseBuffer"-kommandot tas bort så kommer varje cirkel att visas såsom den ritas.

Se även: [UseBuffet](#)

**PlotXY***x, y, form*

*x, y*: koordinater för att plotta form

*form* : ett tal mellan 1 och 13 som specificerar form

1 - Fylld cirkel

2 - Tom cirkel

3 - Fylld kvadrat

4 - Tom kvadrat

5 - Kors

6 - Plus

7 - Tunn

8 - mediumpunkt, solid

9 - mediumpunkt, tom

10 - större punkt, solid

11 - större punkt, tom

12 - största punkt, solid

13 - största punkt, tom

PlotXY 100,100,1

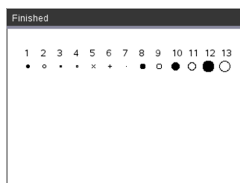


For n,1,13

DrawText 1+22\*n,40,n

PlotXY 5+22\*n,50,n

EndFor



**SetColor**
 Katalog >   
**CXII**
**SetColor**

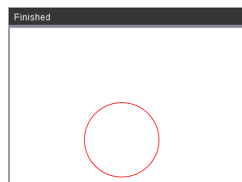
Röd-värde, grön-värde, blå-värde

Värdena för röd, grön och blå måste vara mellan 0 och 255

Ställ in färgen för efterföljande kommandon för Rita

```
SetColor 255,0,0
```

```
DrawCircle 150,150,100
```

**SetPen**
 Katalog >   
**CXII**
**SetPen**

tjocklek, stil

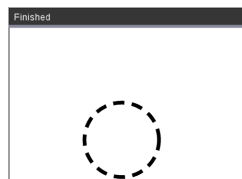
tjocklek: 1 <= tjocklek <= 3 | 1 är tunnast, 3 är tjockast

stil: 1 = Mjuk, 2 = Prickad, 3 = Streckad

Ställer in pennstilen för efterföljande kommandon för Rita

```
SetPen 3,3
```

```
DrawCircle 150,150,50
```

**SertWindow**
 Katalog >   
**CXII**
**SetWindow**

xMin, xMax, yMin, yMax

Etablerar ett logiskt fönster som mappas till grafikens ritområde. Alla parametrar krävs.

Om en del av det ritade objektet är utanför fönstret så kommer utmatningen att klippas (visas ej) och inget felmeddelande kommer att synas.

```
SetWindow 0,160,0,120
```

kommer att ställa in så att utmatningsfönstret har 0,0 i nedre vänstra hörnet och en bredd på 160 och en höjd på 120

```
DrawLine 0,0,100,100
```

```
SetWindow 0,160,0,120
```

```
SetPen 3,3
```

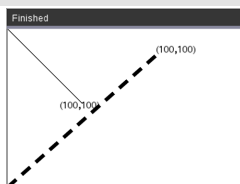
```
DrawLine 0,0,100,100
```

Om  $x_{min}$  är större eller lika med  $x_{max}$  eller om  $y_{min}$  är större eller lika med  $y_{max}$  så kommer ett felmeddelande att visas.

Alla objekt som ritats innan kommandot SetWindow kommer inte att återritas i den nya konfigurationen.

För att återställa fönsterparametrarna till standard, använd:

SetWindow 0,0,0,0



## UseBuffer

Katalog >   
CXII

## UseBuffer

Rita till grafikbuffert istället för skärm (för att öka prestanda)

Detta kommando används i samband med PaintBuffer för att öka hastigheten på skärmens display när programmet genererar flera grafiska objekt.

Med UseBuffer så kommer all grafik att visas endast efter att nästa PaintBuffer-kommando körs.

Kommandot UseBuffer behöver bara användas en gång i programmet, dvs varje användning av PaintBuffer behöver ingen motsvarande UseBuffer

## UseBuffer

För n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radie:=randInt(10,50)

Wait 0,5

XDrawCircle x,y,radie

EndFor

PaintBuffer

Detta program kommer att visa alla 10 cirklar på en gång.

Om "UseBuffer"-kommandot tas bort så kommer varje cirkel att visas såsom den ritas.

Se även: [PaintBuffer](#)



# Tomma element

Vid analys av verkliga data kanske du inte alltid har en komplett datauppsättning. TI-Nspire™ CAS tillåter tomma dataelement så att du kan fortsätta med de nästan kompletta uppgifterna i stället för att behöva börja om från början eller kassera ofullständiga fall.

Du finner ett exempel på data som inbegriper tomma element i kapitlet Listor och kalkylblad under *“Plotta kalkylbladsdata.”*

Med funktionen `delVoid()` kan du ta bort tomma element från en lista. Med funktionen `isVoid()` kan du testa förekomst av tomma element. För mer information, se `delVoid()`, på sidan 50 och `isVoid()`, på sidan 98.

**Obs:** För att manuellt mata in ett tomt element i ett matematiskt uttryck, skriv in “ ” eller nyckelordet `tomrum`. Nyckelordet `tomrum` omvandlas automatiskt till symbolen “ ” när uttrycket utvärderas. För att skriva in “ ” på handenheten, tryck på `ctrl` `⏏`.

## Beräkningar som innehåller tomma element

|                                                                                                           |                                                                                                                                                                                                                                       |                 |   |                         |   |                  |   |                               |                      |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---|-------------------------|---|------------------|---|-------------------------------|----------------------|
| Flertalet beräkningar som inbegriper en tom inmatning producerar ett tomt resultat. Se specialfall nedan. | <table><tr><td><code> _</code></td><td>–</td></tr><tr><td><code>gcd(100,_)</code></td><td>–</td></tr><tr><td><code>3+_</code></td><td>–</td></tr><tr><td><code>{5,_,10}–{3,6,9}</code></td><td><code>{2,_,1}</code></td></tr></table> | <code> _</code> | – | <code>gcd(100,_)</code> | – | <code>3+_</code> | – | <code>{5,_,10}–{3,6,9}</code> | <code>{2,_,1}</code> |
| <code> _</code>                                                                                           | –                                                                                                                                                                                                                                     |                 |   |                         |   |                  |   |                               |                      |
| <code>gcd(100,_)</code>                                                                                   | –                                                                                                                                                                                                                                     |                 |   |                         |   |                  |   |                               |                      |
| <code>3+_</code>                                                                                          | –                                                                                                                                                                                                                                     |                 |   |                         |   |                  |   |                               |                      |
| <code>{5,_,10}–{3,6,9}</code>                                                                             | <code>{2,_,1}</code>                                                                                                                                                                                                                  |                 |   |                         |   |                  |   |                               |                      |

## Lista argument som innehåller tomma element

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|------|----------------------------------|---|-----------------------------------------|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|--------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|
| Följande funktioner och kommandon ignorerar (hoppa över) tomma element som påträffas i listargument:                                                                                                                                                                                                                                                                                                                                                                                                                                    | <table><tr><td><code>sum({2,_,3,5,6,6})</code></td><td>16.6</td></tr><tr><td><code>median({1,2,_,_,3})</code></td><td>2</td></tr><tr><td><code>cumulativeSum({1,2,_,4,5})</code></td><td><code>{1,3,_,7,12}</code></td></tr><tr><td><code>cumulativeSum(<table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>–</td></tr><tr><td>5</td><td>6</td></tr></table>)</code></td><td><table><tr><td>1</td><td>2</td></tr><tr><td>4</td><td>–</td></tr><tr><td>9</td><td>8</td></tr></table></td></tr></table> | <code>sum({2,_,3,5,6,6})</code> | 16.6 | <code>median({1,2,_,_,3})</code> | 2 | <code>cumulativeSum({1,2,_,4,5})</code> | <code>{1,3,_,7,12}</code>                                                                              | <code>cumulativeSum(<table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>–</td></tr><tr><td>5</td><td>6</td></tr></table>)</code> | 1 | 2 | 3 | – | 5 | 6 | <table><tr><td>1</td><td>2</td></tr><tr><td>4</td><td>–</td></tr><tr><td>9</td><td>8</td></tr></table> | 1 | 2 | 4 | – | 9 | 8 |
| <code>sum({2,_,3,5,6,6})</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 16.6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| <code>median({1,2,_,_,3})</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| <code>cumulativeSum({1,2,_,4,5})</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <code>{1,3,_,7,12}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| <code>cumulativeSum(<table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>–</td></tr><tr><td>5</td><td>6</td></tr></table>)</code>                                                                                                                                                                                                                                                                                                                                                                                                      | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 2                               | 3    | –                                | 5 | 6                                       | <table><tr><td>1</td><td>2</td></tr><tr><td>4</td><td>–</td></tr><tr><td>9</td><td>8</td></tr></table> | 1                                                                                                                                  | 2 | 4 | – | 9 | 8 |   |                                                                                                        |   |   |   |   |   |   |
| 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |
| <code>count</code> , <code>countif</code> , <code>cumulativeSum</code> , <code>freqTable</code> → <code>lista</code> , <code>frekvens</code> , <code>max</code> , <code>medelvärde</code> , <code>median</code> , <code>produkt</code> , <code>stDevPop</code> , <code>stDevSamp</code> , <code>summa</code> , <code>sumlf</code> , <code>varPop</code> och <code>varSamp</code> samt regressionsberäkningar, <code>OneVar</code> , <code>TwoVar</code> och <code>FiveNumSummary</code> statistik, konfidensintervaller och stat-tester |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                 |      |                                  |   |                                         |                                                                                                        |                                                                                                                                    |   |   |   |   |   |   |                                                                                                        |   |   |   |   |   |   |

## Lista argument som innehåller tomma element

**SortA** och **SortD** flyttar alla tomma element inom det första argumentet till botten.

|                                   |                    |
|-----------------------------------|--------------------|
| $\{5,4,3,_,1\} \rightarrow list1$ | $\{5,4,3,_,1\}$    |
| $\{5,4,3,2,1\} \rightarrow list2$ | $\{5,4,3,2,1\}$    |
| SortA list1,list2                 | Done               |
| list1                             | $\{1,3,4,5,_{-}\}$ |
| list2                             | $\{1,3,4,5,2\}$    |

|                                   |                    |
|-----------------------------------|--------------------|
| $\{1,2,3,_,5\} \rightarrow list1$ | $\{1,2,3,_,5\}$    |
| $\{1,2,3,4,5\} \rightarrow list2$ | $\{1,2,3,4,5\}$    |
| SortD list1,list2                 | Done               |
| list1                             | $\{5,3,2,1,_{-}\}$ |
| list2                             | $\{5,3,2,1,4\}$    |

I regressionsberäkningar introducerar ett tomrum i en X- eller Y-lista ett tomrum i motsvarande element för residualen.

|                                          |                                              |
|------------------------------------------|----------------------------------------------|
| $l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$ | $\{2,_,3,5,6,6\}$                            |
| LinRegMx l1,l2                           | Done                                         |
| stat.Resid                               | $\{0.434286,_, -0.862857, -0.011429, 0.44\}$ |
| stat.XReg                                | $\{1,_,3,4,5\}$                              |
| stat.YReg                                | $\{2,_,3,5,6,6\}$                            |
| stat.FreqReg                             | $\{1,_,1,1,1,1\}$                            |

En utelämnad kategori i regressionsberäkningar introducerar ett tomrum i motsvarande element för residualen.

|                                           |                 |
|-------------------------------------------|-----------------|
| $l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$      | $\{2,3,5,6,6\}$ |
| $cat:=\{"M","M","F","F"\}; incl:=\{"F"\}$ | $\{"F"\}$       |
| LinRegMx l1,l2,1,cat,incl                 | Done            |
| stat.Resid                                | $\{_,_,0,0\}$   |
| stat.XReg                                 | $\{_,_,4,5\}$   |
| stat.YReg                                 | $\{_,_,5,6,6\}$ |
| stat.FreqReg                              | $\{_,_,1,1,1\}$ |

En frekvens på 0 i regressionsberäkningar introducerar ett tomrum i motsvarande element för residualen.

|                                      |                                       |
|--------------------------------------|---------------------------------------|
| $l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$ | $\{2,3,5,6,6\}$                       |
| LinRegMx l1,l2,{1,0,1,1}             | Done                                  |
| stat.Resid                           | $\{0.069231,_, -0.276923, 0.207692\}$ |
| stat.XReg                            | $\{1,_,4,5\}$                         |
| stat.YReg                            | $\{2,_,5,6,6\}$                       |
| stat.FreqReg                         | $\{1,_,1,1,1\}$                       |

# Kortkommandon för att mata in matematiska uttryck

Med kortkommandon kan du mata in element i matematiska uttryck genom att skriva i stället för att använda Katalogen eller Symbolpaletten. För att exempelvis mata in uttrycket  $\sqrt{6}$  kan du skriva `sqr t (6)` på inmatningsraden. När du trycker på `enter` ändras uttrycket `sqr t (6)` till  $\sqrt{6}$ . Vissa kortkommandon kan användas från både handenheten och datorns tangentbord. Andra är i första hand användbara från datorns tangentbord.

## Från handenheten eller datorns tangentbord

| För att mata in detta:                              | Skriv detta kortkommando:       |
|-----------------------------------------------------|---------------------------------|
| $\pi$                                               | pi                              |
| $\theta$                                            | theta                           |
| $\infty$                                            | infinity                        |
| $\leq$                                              | <=                              |
| $\geq$                                              | >=                              |
| $\neq$                                              | /=                              |
| $\Rightarrow$ (logisk implikation)                  | =>                              |
| $\Leftrightarrow$ (logisk dubbel implikation, XNOR) | <=>                             |
| $\rightarrow$ (lagra operator)                      | =:                              |
| $   $ (absolutbelopp)                               | abs (...)                       |
| $\sqrt{()}$                                         | sqr t (...)                     |
| $d()$                                               | derivative (...)                |
| $\int()$                                            | integral (...)                  |
| $\Sigma()$ (mallen Summa)                           | sumSeq (...)                    |
| $\Pi()$ (mallen Produkt)                            | prodSeq (...)                   |
| $\sin^{-1}()$ , $\cos^{-1}()$ , ...                 | arcsin (...), arccos (...), ... |
| $\Delta\text{List}()$                               | deltaList (...)                 |
| $\Delta\text{tmpCnv}()$                             | deltaTmpCnv (...)               |

## Från datorns tangentbord

| För att mata in detta:   | Skriv detta kortkommando: |
|--------------------------|---------------------------|
| c1, c2, ... (konstanter) | @c1, @c2, ...             |
| n1, n2, ...              | @n1, @n2, ...             |

| För att mata in detta:                           | Skriv detta kortkommando:                     |
|--------------------------------------------------|-----------------------------------------------|
| (heltalskonstanter)                              |                                               |
| $i$ (imaginära enheten)                          | @i                                            |
| e (naturlig logaritmbas e)                       | @e                                            |
| E (grundpotensform)                              | @E                                            |
| T (transponera)                                  | @t                                            |
| r (radianer)                                     | @r                                            |
| ° (grader)                                       | @d                                            |
| g (nygrader)                                     | @g                                            |
| ∠ (vinkel)                                       | @<                                            |
| ► (omvandling)                                   | @>                                            |
| ►Decimal, ►approxFraction<br>( ), och så vidare. | @>Decimal, @>approxFraction(), och så vidare. |

# EOS™-hierarki (Equation Operating System)

Detta avsnitt beskriver det ekvationsoperativsystem (EOS™) som används av TI-Nspire™ CAS Inlärningsteknologi för matematik och naturvetenskap. Tal, variabler och funktioner matas in i en enkel och direkt sekvens. Programvaran EOS™ beräknar uttryck och ekvationer genom parentesgruppering och enligt de prioriteringsregler som beskrivs nedan.

## Ordning vid utvärdering

| Nivå | Operator                                                                                                                                                                  |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Parenteser ( ), hakparenteser [ ], klammerparenteser { }                                                                                                                  |
| 2    | Indirection (#)                                                                                                                                                           |
| 3    | Funktionsanrop                                                                                                                                                            |
| 4    | Postoperatorer: grader-minuter-sekunder ( <sup>°</sup> ,',"), fakultet (!), procent (%), radian ( <sup>r</sup> ), index ([ ]), transponering ( <sup>T</sup> )             |
| 5    | Exponentberäkning, potensoperator (^)                                                                                                                                     |
| 6    | Negation (-)                                                                                                                                                              |
| 7    | Strängsammanlänkning (&)                                                                                                                                                  |
| 8    | Multiplikation (•), division (/)                                                                                                                                          |
| 9    | Addition (+), subtraktion (-)                                                                                                                                             |
| 10   | Likhetsförhållanden: lika med (=), inte lika med (≠ eller /=), mindre än (<), mindre än eller lika med (≤ eller <=), större än (>), större än eller lika med (≥ eller >=) |
| 11   | Logiskt <b>not</b>                                                                                                                                                        |
| 12   | Logiskt <b>and</b>                                                                                                                                                        |
| 13   | Logisk <b>or</b>                                                                                                                                                          |
| 14   | <b>xor, nor, nand</b>                                                                                                                                                     |
| 15   | Logisk implikation (⇒)                                                                                                                                                    |
| 16   | Logisk dubbel implikation, XNOR⇔                                                                                                                                          |
| 17   | (" ")-operatorn begränsning                                                                                                                                               |
| 18   | Spara (→)                                                                                                                                                                 |

## Parenteser, hakparenteser och klammerparenteser

Alla beräkningar inom ett par av parenteser, hakparenteser eller klammerparenteser utvärderas först. I exempelvis uttrycket 4(1+2) beräknar EOS™ först den del av uttrycket som är inom parentesen, 1+2, och multiplicerar sedan resultatet, 3, med 4.

Antalet inledande respektive avslutande parenteser, hakparenteser och klammerparenteser måste vara lika inom ett uttryck eller en ekvation. Annars visas ett felmeddelande som anger det element som saknas. Till exempel visar  $(1+2)/(3+4)$  felmeddelandet "Missing".

**Obs:** Eftersom TI-Nspire™ CAS programvara ger dig möjlighet att definiera dina egna funktioner betraktas ett variabelnamn, följt av ett uttryck inom parentes, som ett "funktionsanrop" i stället för en indirekt multiplikation. Till exempel är  $a(b+c)$  funktionen  $a$  utvärderad med  $b+c$ . För att multiplicera uttrycket  $b+c$  med variabeln  $a$ , använd explicit multiplikation:  $a*(b+c)$ .

## Indirection

Den indirekta operatoren (#) konverterar en sträng till ett variabel- eller funktionsnamn. Till exempel skapar #("x"&"y"&"z") variabelnamnet xyz. "Indirection" ger också möjlighet att skapa och modifiera variabler inne i ett program. Om exempelvis  $10 \rightarrow r$  och  $r \rightarrow s1$  erhålls  $s1=10$ .

## Postoperatorer

Postoperatorer är operatore som kommer direkt efter ett argument, t.ex.  $5!$ ,  $25\%$  eller  $60^\circ 15' 45''$ . Argument som följs av en postoperator utvärderas på den fjärde prioritetsnivån. I exempelvis uttrycket  $4^4 3!$  utvärderas  $3!$  först. Resultatet, 6, blir sedan exponenten för 4, vilket ger 4096.

## Exponentberäkning

Exponentberäkning (^) och exponentberäkning element för element (.^) utvärderas från höger till vänster. Till exempel utvärderas uttrycket  $2^3^2$  på samma sätt som  $2^{(3^2)}$ . Man får resultatet 512. Detta är inte detsamma som  $(2^3)^2$ , vilket ger resultatet 64.

## Negation

För att skriva in ett negativt tal, tryck på  $\boxed{-}$  följt av talet. Postoperationer och exponentberäkningar utförs före negationer. Till exempel är resultatet av  $-x^2$  ett negativt tal och  $-9^2 = -81$ . Använd parenteser för att beräkna kvadraten på ett negativt tal. Till exempel ger  $(-9)^2$  resultatet 81.

## Begränsning ("|")

Argumentet som följer efter ("|")-operatoren begränsning ger en uppsättning av begränsningar som påverkar utvärderingen av argumentet som föregår operatoren.

# TI-Nspire CX II - TI-Basic programmeringsfunktioner

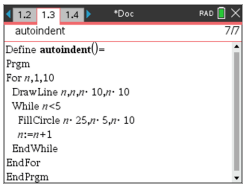
## Autoindentering i Programeditor

Programeditorn för TI-Inspire™ autoindenterar nu satser inuti ett blockkommando.

Blockkommandon är If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Editorn förbereder automatiskt utrymmen för programkommandon inom ett blockkommando. Blockets stängningskommando kommer att justeras tillsammans med öppningskommandot.

Exemplet nedan visar autoindentering i nästlade blockkommandon.



Kodfragment som har kopierats och klistrats in kommer att behålla ursprungsindenteringen.

Öppning av program som skapats i tidigare version av programvaran kommer att behålla ursprungsindenteringen.

## Förbättrade felmeddelanden för TI-Basic

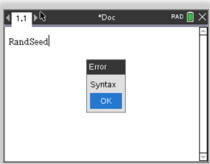
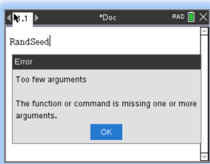
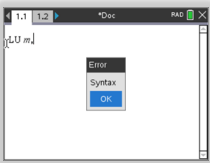
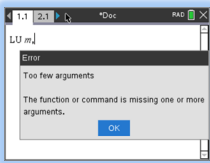
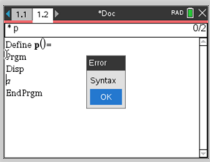
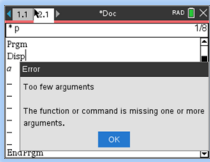
### Fel

| Feltillstånd                  | Nytt meddelande                                                                                                                                                                                                                |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fel i villkorssats (If/While) | En villkorssats löstes inte till <b>SANT</b> eller <b>FALSKT</b><br><b>OBS:</b> Med ändringen att flytta markören på raden med felet så behöver vi inte längre specificera om felet är i en "If"-sats eller i en "While"-sats. |
| Saknar EndIf                  | Förväntade <b>EndIf</b> men hittade en annan endsats                                                                                                                                                                           |
| Saknar EndFor                 | Förväntade <b>EndFor</b> men hittade en annan endsats                                                                                                                                                                          |
| Saknar EndWhile               | Förväntade <b>EndWhile</b> men hittade en annan endsats                                                                                                                                                                        |
| Saknar EndLoop                | Förväntade <b>EndLoop</b> men hittade en annan endsats                                                                                                                                                                         |

| Feltillstånd                                                      | Nytt meddelande                                                                     |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Saknar <b>EndTry</b>                                              | Förväntade <b>EndTry</b> men hittade en annan end-sats                              |
| "Then" utelämnad efter <b>If</b> <condition>                      | Saknar <b>If...Then</b>                                                             |
| "Then" utelämnad efter <b>Elseif</b> <condition>                  | <b>Then</b> saknas i block: <b>Elseif</b> .                                         |
| När "Then", "Else" och "Elseif" påträffades utanför kontrollblock | <b>Else</b> ogiltigt utanför block: <b>If..Then..EndIf</b> eller <b>Try..EndTry</b> |
| "Elseif" syns utanför "If..Then..EndIf"-block                     | <b>Elseif</b> ogiltigt utanför block: <b>If...Then...EndIf</b>                      |
| "Then" syns utanför "If....EndIf"-block                           | <b>Then</b> ogiltigt utanför block: <b>If..EndIf</b>                                |

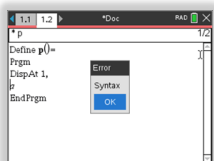
## Syntaxfel

Om kommandon som förväntar sig ett eller flera argument anropas med en ofullständig lista över argument så kommer ett **"För få argument-fel"** att utfärdas istället för ett "syntax"-fel

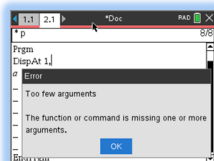
| Aktuellt uppträdande                                                                                                                                                                                                                                                                            | Nytt CX II-uppträdande                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered in the input line. An error dialog box is displayed with the message 'Error Syntax' and an 'OK' button.</p> |  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered in the input line. An error dialog box is displayed with the message 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.</p> |
|  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'L1 m1' is entered in the input line. An error dialog box is displayed with the message 'Error Syntax' and an 'OK' button.</p>   |  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'L1 m1' is entered in the input line. An error dialog box is displayed with the message 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.</p>   |
|  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'p' is entered in the input line. An error dialog box is displayed with the message 'Error Syntax' and an 'OK' button.</p>      |  <p>The screenshot shows the TI-84 Plus CE calculator interface. The command 'p' is entered in the input line. An error dialog box is displayed with the message 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.</p>      |



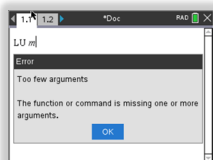
## Aktuellt uppträdande



## Nytt CX II-uppträdande



**Obs:** När en ofullständig lista över argument inte följs av ett kommatecken så är felmeddelandet: "för få argument". Detta är samma som för tidigare versioner.



# Konstanter och värden

Följande tabell listar konstanterna och deras värden som finns tillgängliga när enhetsomvandling utförs (NIST 2018). De kan skrivas in manuellt eller väljas från listan **Konstanter i Verktyg > Enhetsomvandlingar** (Handenhet: Tryck  3).

| Konstant | Namn                                        | Värde                                            |
|----------|---------------------------------------------|--------------------------------------------------|
| _c       | Ljusets hastighet                           | 299792458 _m/_s                                  |
| _Cc      | Coulombs konstant                           | 8987551792.261 _m/_F                             |
| _Fc      | Faradays konstant                           | 96485.33212 _coul/_mol                           |
| _g       | Tyngdkraftens acceleration                  | 9.80665 _m/_s <sup>2</sup>                       |
| _Gc      | Gravitationskonstanten                      | 6.6743E-11 _m <sup>3</sup> /_kg/_s <sup>2</sup>  |
| _h       | Plancks konstant                            | 6.62607015E-34 _J _s                             |
| _k       | Boltzmanns konstant                         | 1.380649E-23 _J/_°K                              |
| _μ0      | Permeabilitet vid vakuum                    | 1.25663706127E-6 _N/_A <sup>2</sup>              |
| _μb      | Bohrmagneton                                | 9.274009994E-24 _J _m <sup>2</sup> /_Wb          |
| _Me      | Elektronens vilomassa                       | 9.10938371390E-31 _kg                            |
| _Mμ      | Myonmassa                                   | 1.883531627E-28 _kg                              |
| _Mn      | Neutronens vilomassa                        | 1.67492750058E-27 _kg                            |
| _Mp      | Protonens vilomassa                         | 1.67262192595E-27 _kg                            |
| _Na      | Avogadros tal                               | 6.02214076E23 /_mol                              |
| _q       | Elektronens laddning                        | 1.602176634E-19 _coul                            |
| _R       | Konstant tryck för vattenpotential          | 0.0831 L bars/mole K                             |
| _Rb      | Bohrradie                                   | 5.29177210903E-11 _m                             |
| _Rc      | Molar gas constant (Allmänna gaskonstanten) | 8.314462618 _J/_mol/_°K                          |
| _Rdb     | Rydbergs konstant                           | 10973731.568160/_m                               |
| _Re      | Elektronradie                               | 2.8179403262E-15 _m                              |
| _u       | Atommassa                                   | 1.66053906893E-27 _kg                            |
| _Vm      | Molvolyt                                    | 2.241396954E-2 _m <sup>3</sup> /_mol             |
| _ε0      | Permittivitet vid vakuum                    | 8.8541878188E-12 _F/_m                           |
| _σ       | Stefan-Boltzmanns konstant                  | 5.670367E-8 _W/_m <sup>2</sup> /_°K <sup>4</sup> |
| _φ0      | Magnetiskt flödeskvantum                    | 2.067838331E-15 _Wb                              |

## Felkoder och meddelanden

När ett fel inträffar placeras dess felkod i variabeln *errorCode*. Användardefinierade program och funktioner kan undersöka *errorCode* för att bestämma orsaken till ett fel. För ett exempel på användningen av *errorCode*, se exempel 2 under kommandot **Try**, här.

**Obs:** Vissa feltillstånd avser endast TI-Nspire™ CAS-produkter medan andra endast avser TI-Nspire™-produkter.

| Felkod | Beskrivning                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10     | A function did not return a value (En funktion gav inget värde)                                                                                                                                                                                                                                                                                                                                                                            |
| 20     | A test did not resolve to TRUE or FALSE. (Ett test gav varken TRUE eller FALSE.)<br><br>I regel kan odefinierade variabler inte jämföras. Som exempel orsakar testet "Om $a < b$ " detta fel om a eller b är odefinierade när Om-påståendet exekveras.                                                                                                                                                                                     |
| 30     | Argument cannot be a folder name. (Argumentet får inte vara ett mappnamn.)                                                                                                                                                                                                                                                                                                                                                                 |
| 40     | Argument error (Argumentfel)                                                                                                                                                                                                                                                                                                                                                                                                               |
| 50     | Argument mismatch (Fel typ av argument)<br><br>Två eller flera argument måste vara av samma typ.                                                                                                                                                                                                                                                                                                                                           |
| 60     | Argument must be a Boolean expression or integer (Argumentet måste vara ett booleskt uttryck eller ett heltal)                                                                                                                                                                                                                                                                                                                             |
| 70     | Argument must be a decimal number (Argumentet måste vara ett decimaltal)                                                                                                                                                                                                                                                                                                                                                                   |
| 90     | Argument must be a list (Argumentet måste vara en lista)                                                                                                                                                                                                                                                                                                                                                                                   |
| 100    | Argument must be a matrix (Argumentet måste vara en matris)                                                                                                                                                                                                                                                                                                                                                                                |
| 130    | Argument must be a string (Argumentet måste vara en sträng)                                                                                                                                                                                                                                                                                                                                                                                |
| 140    | Argument must be a variable name (Argumentet måste vara ett variabelnamn).<br><br>Kontrollera att namnet: <ul style="list-style-type: none"><li>• inte börjar med en siffra</li><li>• inte innehåller mellanslag eller specialtecken</li><li>• inte innehåller understrykningstecken eller punkt på ogiltigt sätt</li><li>• inte överskrider max. längd</li></ul> Se avsnittet Calculator (Räknare) i dokumentationen för mer information. |
| 160    | Argument must be an expression (Argumentet måste vara ett uttryck)                                                                                                                                                                                                                                                                                                                                                                         |
| 165    | Batteries too low for sending or receiving (Batterierna är för svaga för sändning eller mottagning)<br><br>Sätt i nya batterier före sändning eller mottagning.                                                                                                                                                                                                                                                                            |
| 170    | Bound (Gräns)                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Felkod | Beskrivning                                                                                                                                                                                                                                                         |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | Den nedre gränsen måste vara mindre än den övre gränsen för att definiera sökintervallet.                                                                                                                                                                           |
| 180    | Avsluta<br>Tangenten  eller  trycktes ned under en lång beräkning eller under programexekvering.  |
| 190    | Circular definition (Cirkulär definition)<br>Detta meddelande visas för att inte minnet skall ta slut under ändlös ersättning av variabelvärden i samband med förenkling. Till exempel orsakar $a+1 \rightarrow a$ , där $a$ är en odefinierad variabel, detta fel. |
| 200    | Invalid constraint (Ogiltig begränsning)<br>Som exempel skulle $\text{solve}(3x^2-4=0,x) \mid x<0 \text{ eller } x>5$ ge detta felmeddelande eftersom begränsningen är separerad av "eller" i stället för "och".                                                    |
| 210    | Invalid data type (Ogiltig datatyp)<br>Ett argument är av fel datatyp.                                                                                                                                                                                              |
| 220    | Dependent limit (Beroende gräns)                                                                                                                                                                                                                                    |
| 230    | Dimension<br>Ett index för en lista eller för en matris är inte giltigt. Om exempelvis listan {1,2,3,4} lagras i L1 är L1[5] ett dimensionsfel eftersom L1 endast innehåller fyra element.                                                                          |
| 235    | Dimension mismatch. (Dimensioner överensstämmer inte.) Inte tillräckligt med element i listorna.                                                                                                                                                                    |
| 240    | Dimension mismatch (Dimensioner överensstämmer inte)<br>Två eller flera argument måste ha samma dimension. Till exempel är $[1,2]+[1,2,3]$ ett dimensionsfel eftersom matriserna innehåller olika antal element.                                                    |
| 250    | Divide by zero (Division med noll)                                                                                                                                                                                                                                  |
| 260    | Domain error (Områdesfel)<br>Ett argument måste vara inom ett specificerat område. Exempelvis är $\text{rand}(0)$ inte giltigt.                                                                                                                                     |
| 270    | Duplicate variable name (Dubblade variabelnamn)                                                                                                                                                                                                                     |
| 280    | Else and Elseif invalid outside of If...EndIf block (Else och Elseif är ogiltiga utanför If...EndIf-block)                                                                                                                                                          |
| 290    | EndTry is missing the matching Else statement (EndTry saknas i det matchande Else-påståendet)                                                                                                                                                                       |

| Felkod | Beskrivning                                                                                                                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 295    | Excessive iteration (För många iterationer)                                                                                                                                                                                                                 |
| 300    | Expected 2 or 3-element list or matrix (En 2- eller 3-elements lista eller matris förväntades)                                                                                                                                                              |
| 310    | Första argumentet till nSolve måste vara en ekvation i en variabel. Det får inte innehålla någon variabel utan värde förutom den variabel som används i beräkningen.                                                                                        |
| 320    | First argument of solve or cSolve must be an equation or inequality (Första argumentet till solve eller cSolve måste vara en ekvation eller en olikhet)<br><br>Till exempel är solve( $3x^2-4$ ,x) ogiltigt eftersom första argumentet inte är en ekvation. |
| 345    | Inconsistent units (Enheterna är oförenliga)                                                                                                                                                                                                                |
| 350    | Index out of range (Index ligger utanför giltigt område)                                                                                                                                                                                                    |
| 360    | Indirection string is not a valid variable name (Indirection-strängen är inte ett giltigt variabelnamn)                                                                                                                                                     |
| 380    | Undefined Ans (Odefinierad Ans)<br><br>Antingen skapades inte Ans av den föregående beräkningen eller också har ingen tidigare beräkning matats in.                                                                                                         |
| 390    | Invalid assignment (Ogiltig tilldelning)                                                                                                                                                                                                                    |
| 400    | Invalid assignment value (Ogiltigt tilldelningsvärde)                                                                                                                                                                                                       |
| 410    | Invalid command (Ogiltigt kommando)                                                                                                                                                                                                                         |
| 430    | Invalid for the current mode settings (Ogiltigt för de aktuella inställningarna)                                                                                                                                                                            |
| 435    | Invalid guess (Ogiltig gissning)                                                                                                                                                                                                                            |
| 440    | Invalid implied multiply (Ogiltig indirekt multiplikation)<br><br>Till exempel är $x(x+1)$ ogiltig medan $x*(x+1)$ har korrekt syntax. Detta är för att undvika förväxling mellan indirekt multiplikation och funktionsanrop.                               |
| 450    | Invalid in a function or current expression (Ogiltigt i en funktion eller i det aktuella uttrycket)<br><br>Endast vissa kommandon är giltiga i en användardefinierad funktion.                                                                              |
| 490    | Invalid in Try..EndTry block (Ogiltigt i Try..EndTry-block)                                                                                                                                                                                                 |
| 510    | Invalid list or matrix (Ogiltig lista eller matris)                                                                                                                                                                                                         |
| 550    | Invalid outside function or program (Ogiltigt utanför funktion eller program)<br><br>Ett antal kommandon är inte giltiga utanför en funktion eller ett program. Exempelvis kan Local bara användas inne i en funktion eller ett program.                    |

| Felkod | Beskrivning                                                                                                                                                                                                                 |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 560    | Invalid outside Loop..EndLoop, For..EndFor, or While..EndWhile blocks (Ogiltigt utanför Loop..EndLoop, For..EndFor, eller While..EndWhile-block)<br><br>Exempelvis kan kommandot Exit bara användas innanför dessa slingor. |
| 565    | Invalid outside program (Ogiltigt utanför program)                                                                                                                                                                          |
| 570    | Invalid pathname (Ogiltigt sökvägsnamn)<br><br>Till exempel är \var ogiltigt.                                                                                                                                               |
| 575    | Invalid polar complex (Ogiltigt polärt komplext tal)                                                                                                                                                                        |
| 580    | Invalid program reference (Ogiltig programreferens)<br><br>Man får inte referera till Program inom funktioner eller uttryck såsom 1+p(x) där p är ett program.                                                              |
| 600    | Invalid table (Ogiltig tabell)                                                                                                                                                                                              |
| 605    | Invalid use of units (Ogiltig användning av enheter)                                                                                                                                                                        |
| 610    | Invalid variable name in a Local statement (Ogiltigt variabelnamn i ett Local-påstående)                                                                                                                                    |
| 620    | Invalid variable or function name (Ogiltigt variabel- eller funktionsnamn)                                                                                                                                                  |
| 630    | Invalid variable reference (Ogiltig variabelreferens)                                                                                                                                                                       |
| 640    | Invalid vector syntax (Ogiltig syntax för vektor)                                                                                                                                                                           |
| 650    | Link transmission (Länköverföring)<br><br>En överföring mellan två enheter slutfördes inte. Kontrollera att anslutningskabeln är ordentligt ansluten i båda ändarna.                                                        |
| 665    | Matrix not diagonalizable (Matrisen är inte diagonaliserbar)                                                                                                                                                                |
| 670    | Low Memory (Ont om minne)<br><br>1. Ta bort lite data från detta dokument<br>2. Spara och stäng detta dokument<br><br>Om dessa steg inte löser problemet, plocka ur batterierna och sätt i dem igen                         |
| 672    | Resource exhaustion (Resursutmattnig)                                                                                                                                                                                       |
| 673    | Resource exhaustion (Resursutmattnig)                                                                                                                                                                                       |
| 680    | Missing ( ( Saknar " ( "                                                                                                                                                                                                    |
| 690    | Missing ) ( Saknar " ) "                                                                                                                                                                                                    |
| 700    | Missing " ( Saknar " " "                                                                                                                                                                                                    |

| Felkod | Beskrivning                                                                                                                                                                                                                                               |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 710    | Missing ] (Saknar " ] ")                                                                                                                                                                                                                                  |
| 720    | Missing } (Saknar " } ")                                                                                                                                                                                                                                  |
| 730    | Missing start or end of block syntax (Saknar start eller slut i blockets syntax)                                                                                                                                                                          |
| 740    | Missing Then in the If..Endif block (Saknar "Then" i If..Endif-blocket)                                                                                                                                                                                   |
| 750    | Name is not a function or program (Namnet är inte en funktion eller ett program)                                                                                                                                                                          |
| 765    | No functions selected (Inga funktioner är valda)                                                                                                                                                                                                          |
| 780    | No solution found (Ingen lösning funnen)                                                                                                                                                                                                                  |
| 800    | Non-real result (Resultatet är inte ett reellt tal)<br><br>Om exempelvis programvaran har inställningen Real så är $\sqrt{-1}$ ogiltigt.<br><br>För att medge komplexa resultat, ändra lägesinställningen "Real or Complex" till RECTANGULAR eller POLAR. |
| 830    | Overflow (För stort värde)                                                                                                                                                                                                                                |
| 850    | Program not found (Program hittades inte)<br><br>En programreferens inne i ett annat program kunde ej hittas via angiven sökväg under exekveringen.                                                                                                       |
| 855    | Rand type functions not allowed in graphing (Funktioner av slumptyp tillåts ej vid grafitrning)                                                                                                                                                           |
| 860    | Recursion too deep (Rekursjonen för djup)                                                                                                                                                                                                                 |
| 870    | Reserved name or system variable (Reserverat namn eller systemvariabel)                                                                                                                                                                                   |
| 900    | Argument error (Argumentfel)<br><br>Median-median-modellen kunde inte användas på datauppsättningen.                                                                                                                                                      |
| 910    | Syntaxfel                                                                                                                                                                                                                                                 |
| 920    | Text not found (Text hittades inte)                                                                                                                                                                                                                       |
| 930    | Too few arguments (För få argument)<br><br>Funktionen eller kommandot saknar ett eller flera argument.                                                                                                                                                    |
| 940    | Too many arguments (För många argument)<br><br>Uttrycket eller ekvationen innehåller för många argument och kan inte utvärderas.                                                                                                                          |
| 950    | Too many subscripts (För många nedsänkta tecken)                                                                                                                                                                                                          |
| 955    | Too many undefined variables (För många odefinierade variabler)                                                                                                                                                                                           |
| 960    | Variable is not defined (Variabel ej definierad)                                                                                                                                                                                                          |

| Felkod | Beskrivning                                                                                                                                                                                                                                      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <p>Inget värde är tillskrivet variabeln. Använd något av följande kommandon:</p> <ul style="list-style-type: none"> <li>• sto →</li> <li>• :=</li> <li>• <b>Define (Definiera)</b></li> </ul> <p>för att tilldela variabler värden.</p>          |
| 965    | Unlicensed OS (Olicensierat OS)                                                                                                                                                                                                                  |
| 970    | Variable in use so references or changes are not allowed (Variabeln används varför referenser eller ändringar ej tillåts)                                                                                                                        |
| 980    | Variable is protected (Variabeln är skyddad)                                                                                                                                                                                                     |
| 990    | <p>Invalid variable name (Ogiltigt variabelnamn)</p> <p>Kontrollera att namnet inte överskrider max. längd.</p>                                                                                                                                  |
| 1000   | Window variables domain (Fönstervariabeldomän)                                                                                                                                                                                                   |
| 1010   | Zoom                                                                                                                                                                                                                                             |
| 1020   | Internal error (Internt fel)                                                                                                                                                                                                                     |
| 1030   | Protected memory violation (Minnesskydd)                                                                                                                                                                                                         |
| 1040   | Unsupported function. (Funktionen stöds ej.) Denna funktion kräver Computer Algebra System. Prova TI-Nspire™ CAS.                                                                                                                                |
| 1045   | Unsupported operator. (Operatören stöds ej.) Denna operator kräver Computer Algebra System. Prova TI-Nspire™ CAS.                                                                                                                                |
| 1050   | Unsupported feature. (Funktionen stöds ej.) Denna operator kräver Computer Algebra System. Prova TI-Nspire™ CAS.                                                                                                                                 |
| 1060   | <p>Input argument must be numeric. (Inmatade argument måste vara numeriska.)</p> <p>Endast inmatningar som innehåller numeriska värden är tillåtna.</p>                                                                                          |
| 1070   | Trig function argument too big for accurate reduction (Trig-funktionens argument är för stort för en noggrann reduktion)                                                                                                                         |
| 1080   | Unsupported use of Ans.This application does not support Ans. (Användning av Ans stöds inte. Denna applikation stöder inte Ans.)                                                                                                                 |
| 1090   | <p>Function is not defined. (Funktion ej definierad.) Använd något av följande kommandon:</p> <ul style="list-style-type: none"> <li>• <b>Define (Definiera)</b></li> <li>• :=</li> <li>• sto →</li> </ul> <p>för att definiera en funktion.</p> |
| 1100   | Non-real calculation (Icke-reell beräkning)                                                                                                                                                                                                      |



| Felkod | Beskrivning                                                                                                                                                                                                                                                                                                                                                                    |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | Om exempelvis programvaran har inställningen Real så är $\sqrt{-1}$ ogiltigt.<br>För att medge komplexa resultat, ändra lägesinställningen "Real or Complex" till RECTANGULAR eller POLAR.                                                                                                                                                                                     |
| 1110   | Invalid bounds (Ogiltiga gränser)                                                                                                                                                                                                                                                                                                                                              |
| 1120   | No sign change (Ingen teckenändring)                                                                                                                                                                                                                                                                                                                                           |
| 1130   | Argument cannot be a list or matrix (Argumentet får inte vara en lista eller en matris)                                                                                                                                                                                                                                                                                        |
| 1140   | Argument error (Argumentfel)<br><br>Det första argumentet måste vara ett polynomuttryck i det andra argumentet. Om det andra argumentet utelämnas försöker programvaran välja ett förinställt argument.                                                                                                                                                                        |
| 1150   | Argument error (Argumentfel)<br><br>De första två argumenten måste vara polynomuttryck i det tredje argumentet. Om det tredje argumentet utelämnas försöker programvaran välja ett förinställt argument.                                                                                                                                                                       |
| 1160   | Invalid library pathname (Ogiltigt sökvägsnamn för bibliotek)<br><br>Ett sökvägsnamn måste vara i formen <code>xxx\yyy</code> , där: <ul style="list-style-type: none"> <li>Delen <code>xxx</code> kan ha 1 till 16 tecken.</li> <li>Delen <code>yyy</code> kan ha 1 till 15 tecken.</li> </ul> Se avsnittet Bibliotek i dokumentationen för mer information.                  |
| 1170   | Invalid use of library pathname (Ogiltig användning av sökvägsnamn för bibliotek) <ul style="list-style-type: none"> <li>Ett sökvägsnamn får inte ges ett värde med <b>Define</b>, <code>:=</code> eller <code>sto</code> →.</li> <li>Ett sökvägsnamn får inte anges som en Local-variabel eller användas som en parameter i en funktions- eller programdefinition.</li> </ul> |
| 1180   | Invalid library variable name. (Ogiltigt namn på biblioteksvariabel.)<br><br>Kontrollera att namnet: <ul style="list-style-type: none"> <li>inte innehåller en punkt</li> <li>inte börjar med ett understrykningstecken</li> <li>inte överskrider 15 tecken</li> </ul> Se avsnittet Bibliotek i dokumentationen för mer information.                                           |
| 1190   | Biblioteksdokument kunde ej hittas: <ul style="list-style-type: none"> <li>Kontrollera att biblioteket är i MyLib-mappen.</li> <li>Uppdatera bibliotek.</li> </ul> Se avsnittet Bibliotek i dokumentationen för mer information.                                                                                                                                               |

| Felkod | Beskrivning                                                                                                                                                                                                                                                                                                                                                            |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1200   | <p>Biblioteksvariabel kunde ej hittas:</p> <ul style="list-style-type: none"> <li>Kontrollera att biblioteksvariabeln finns i det första problemet i biblioteket.</li> <li>Kontrollera att biblioteksvariabeln har definierats som LibPub eller LibPriv.</li> <li>Uppdatera bibliotek.</li> </ul> <p>Se avsnittet Bibliotek i dokumentationen för mer information.</p> |
| 1210   | <p>Ogiltigt genvägsnamn till bibliotek.</p> <p>Kontrollera att namnet inte:</p> <ul style="list-style-type: none"> <li>innehåller en punkt</li> <li>börjar med ett understrykningstecken</li> <li>överskrider 16 tecken</li> <li>är ett reserverat namn</li> </ul> <p>Se avsnittet Bibliotek i dokumentationen för mer information.</p>                                |
| 1220   | <p>Områdesfel:</p> <p>Funktionerna tangentLine och normalLine stöder endast funktioner med reella värden.</p>                                                                                                                                                                                                                                                          |
| 1230   | <p>Områdesfel.</p> <p>Trigonometriska omvandlingsoperatorer stöds inte i vinkellägena Grader och Nygrader.</p>                                                                                                                                                                                                                                                         |
| 1250   | <p>Argumentfel</p> <p>Använd ett linjärt ekvationssystem.</p> <p>Exempel på ett linjärt ekvationssystem med variablerna x och y:</p> $3x + 7y = 5$ $2y - 5x = -1$                                                                                                                                                                                                      |
| 1260   | <p>Argumentfel:</p> <p>Det första argumentet till nfMin eller nfMax måste vara ett uttryck i en variabel. Det får inte innehålla någon variabel utan värde förutom den variabel som används i beräkningen.</p>                                                                                                                                                         |
| 1270   | <p>Argumentfel</p> <p>Derivatans ordning måste vara lika med 1 eller 2.</p>                                                                                                                                                                                                                                                                                            |
| 1280   | <p>Argumentfel</p> <p>Använd ett polynom i expanderad form i en variabel.</p>                                                                                                                                                                                                                                                                                          |
| 1290   | Argumentfel                                                                                                                                                                                                                                                                                                                                                            |

| Felkod | Beskrivning                                                                             |
|--------|-----------------------------------------------------------------------------------------|
|        | Använd ett polynom i en variabel.                                                       |
| 1300   | Argumentfel<br>Koefficienterna i polynomet måste beräknas till numeriska värden.        |
| 1310   | Argumentfel:<br>En funktion kunde inte utvärderas för ett eller flera av dess argument. |
| 1380   | Argumentfel:<br>Nästlade anrop till funktionen <code>domän()</code> är inte tillåtna.   |

## Varningskoder och meddelanden

Du kan använda funktionen **warnCodes()** för att lagra de varningskoder som genereras genom att utvärdera ett uttryck. Denna tabell listar varje numerisk varningskod och tillhörande meddelande. För ett exempel på lagring av varningskoder, se **warnCodes()**, här.

| Varningskod | Meddelande                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10000       | Operationen kan ge falska lösningar.<br><br>Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt.                                                                                                                                                                                                                                                                                                                  |
| 10001       | Derivering av en ekvation kan ge en falsk ekvation.                                                                                                                                                                                                                                                                                                                                                                                            |
| 10002       | Tvivelaktig lösning<br><br>Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt.                                                                                                                                                                                                                                                                                                                                   |
| 10003       | Tvivelaktig noggrannhet<br><br>Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt.                                                                                                                                                                                                                                                                                                                               |
| 10004       | Operationen kan förlora lösningar.<br><br>Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt.                                                                                                                                                                                                                                                                                                                    |
| 10005       | cSolve kan ange fler nollställen.                                                                                                                                                                                                                                                                                                                                                                                                              |
| 10006       | Solve kan ange fler nollställen.<br><br>Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt.                                                                                                                                                                                                                                                                                                                      |
| 10007       | Flera lösningar kan finnas. Försök att ange lämpliga nedre och övre gränser och/eller en gissning.<br><br>Exempel som använder solve(): <ul style="list-style-type: none"><li>• solve(Ekvation, Var=Gissning)   undrGräns&lt;Var&lt;övrGräns</li><li>• solve(Ekvation, Var)   undrGräns&lt;Var&lt;övrGräns</li><li>• solve(Ekvation, Var=Gissning)</li></ul> Försök att använda grafiska metoder för att verifiera resultaten, om tillämpligt. |
| 10008       | Resultatets definitionsområde kan vara mindre än inmatningens definitionsområde.                                                                                                                                                                                                                                                                                                                                                               |

| Varningskod | Meddelande                                                                                                                                                      |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10009       | Resultatets definitionsområde kan vara större än inmatningens definitionsområde.                                                                                |
| 10012       | Icke reell beräkning                                                                                                                                            |
| 10013       | $\infty^0$ eller $\text{undef}^0$ ersätts med 1                                                                                                                 |
| 10014       | $\text{undef}^0$ ersätts med 1                                                                                                                                  |
| 10015       | $1^\infty$ eller $1^\text{undef}$ ersätts med 1                                                                                                                 |
| 10016       | $1^\text{undef}$ ersätts med 1                                                                                                                                  |
| 10017       | För stort värde ersätts med $\infty$ eller $-\infty$                                                                                                            |
| 10018       | Operationen kräver och ger ett 64-bitars värde.                                                                                                                 |
| 10019       | Resursutmattnings, förenklingen kan vara ofullständig.                                                                                                          |
| 10020       | Trig-funktionens argument är för stort för en noggrann reduktion.                                                                                               |
| 10021       | Inmatningen innehåller en odefinierad parameter.<br>Resultatet kanske inte är giltigt för samtliga möjliga parametervärden.                                     |
| 10022       | En lösning kan vara att ange lämpliga övre och undre gränser.                                                                                                   |
| 10023       | Skalar har multiplicerats med enhetsmatris.                                                                                                                     |
| 10024       | Resultat uppnått med approximerad aritmetik.                                                                                                                    |
| 10025       | Ekvivalens kan inte verifieras i EXAKT läge.                                                                                                                    |
| 10026       | Begränsning kan ignoreras. Ange begränsning i formen "Variable MathTestSymbol Constant" eller en sammanslagning av dessa former, t.ex. " $x < 3$ and $x > 12$ " |

## Allmän information

### ***Hjälp-funktion online***

[education.ti.com/eguide](http://education.ti.com/eguide)

Välj ditt land för ytterligare produktinformation.

### ***Kontakta TI support***

[education.ti.com/ti-cares](http://education.ti.com/ti-cares)

Välj ditt land för teknisk och andra supportresurser.

### ***Service- och garanti-information***

[education.ti.com/warranty](http://education.ti.com/warranty)

Välj ditt land för information om garantins längd och villkor eller om produkttjänsten.

Begränsad garanti. Denna garanti påverkar inte dina lagstadgade rättigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

# Index

|                                |     |                                   |     |
|--------------------------------|-----|-----------------------------------|-----|
| <b>!</b>                       |     | <b>—</b>                          |     |
| !, fakultet .....              | 229 | —, enhetsbeteckning .....         | 240 |
| <b>"</b>                       |     | <b> </b>                          |     |
| ", sekundnotation .....        | 238 | , operatörn begränsning .....     | 242 |
| <b>#</b>                       |     | <b>+</b>                          |     |
| #, indirection .....           | 236 | +, addera .....                   | 219 |
| #, indirection, operator ..... | 266 | <b>/</b>                          |     |
| <b>%</b>                       |     | /, dividera[*] .....              | 221 |
| %, procent .....               | 225 | <b>=</b>                          |     |
| <b>&amp;</b>                   |     | ≠, inte lika med[*] .....         | 226 |
| &, lägg till .....             | 230 | =, lika med .....                 | 225 |
| <b>*</b>                       |     | <b>&gt;</b>                       |     |
| *, multiplicera .....          | 220 | >, större än .....                | 227 |
| <b>,</b>                       |     | <b>Π</b>                          |     |
| ', minutnotation .....         | 238 | Π, produkt[*] .....               | 233 |
| ', prim .....                  | 239 | <b>Σ</b>                          |     |
| <b>—</b>                       |     | Σ( ), summa[*] .....              | 233 |
| —, subtrahera[*] .....         | 219 | ΣInt( ) .....                     | 234 |
| <b>.</b>                       |     | ΣPrn( ) .....                     | 235 |
| ., punkt multiplikation .....  | 224 | <b>√</b>                          |     |
| —, punkt subtraktion .....     | 223 | √, kvadratrot[*] .....            | 232 |
| ./, punkt division .....       | 224 | <b>∫</b>                          |     |
| ^, punkt potens .....          | 224 | ∫, integrera[*] .....             | 231 |
| +, punkt addition .....        | 223 | <b>≤</b>                          |     |
| <b>:</b>                       |     | ≤, mindre än eller lika med ..... | 227 |
| :=, tilldela .....             | 244 | <b>≥</b>                          |     |
| <b>^</b>                       |     | ≥, större än eller lika med ..... | 228 |
| ^, potens .....                | 222 |                                   |     |
| ^-1, inverterat värde .....    | 241 |                                   |     |

|                                                  |          |                                    |       |
|--------------------------------------------------|----------|------------------------------------|-------|
|                                                  |          | #                                  |       |
| ►, konvertera enheter[*]                         | 240      | 0h, hexadecimal indikator          | 245   |
| ►, konvertera till nygradvinkel[Grad]            | 90       | #                                  |       |
| ►approxFraction()                                | 14       | #                                  |       |
| ►Base2, visa som binär[Bas 2]                    | 18       | 2-sampel F Test                    | 78    |
| ►Base10, visa som decimalt heltal<br>[Bas 10]    | 19       | #                                  |       |
| ►Base16, visa som hexadecimal[Bas<br>16]         | 19       | 10^( ), tiopotens                  | 241   |
| ►cos, visning i termer av cosinus[cos]           | 30       | A                                  |       |
| ►Cylind, visa som cylindrisk vektor<br>[Cylind]  | 43       | abs( ), absolutbelopp              | 8     |
| ►DD, visa som decimal vinkel[DD]                 | 46       | absolutbelopp<br>mall              | 3-4   |
| ►Decimal, visa resultat som decimal<br>[Decimal] | 47       | addera, +                          | 219   |
| ►DMS, visa som grad/minut/sekund<br>[DMS]        | 56       | amorteringstabell, amortTbl()      | 8, 17 |
| ►exp, visning i termer av e[exp]                 | 66       | amortTbl(), amorteringstabell      | 8, 17 |
| ►Polar, visa som polär vektor[Polär]             | 138      | and, Booleska operatörer           | 9     |
| ►Rad, omvandla till radianer                     | 149      | andrerivata<br>mall för            | 6     |
| ►Rect, visa som ortogonal vektor                 | 152      | angle( ), vinkel                   | 10    |
| ►sin, visning i termer av sinus[sin]             | 173      | annars, Else                       | 90    |
| ►Sphere, visa som sfärisk vektor<br>[sfär]       | 182      | ANOVA 2-vägs, 2-vägs variansanalys | 11    |
| →                                                |          | ANOVA, 1-vägs variansanalys        | 10    |
| →, lagra                                         | 243      | Ans, sista svar                    | 13    |
| ⇒                                                |          | antal dagar mellan datum, dbd()    | 46    |
| ⇒, logisk implikation[*]                         | 228, 263 | approx(), ungefärlig               | 13    |
| ⇔                                                |          | approxRational()                   | 14    |
| ⇔, logisk dubbel implikation[*]                  | 229      | arccos()                           | 14    |
| ©                                                |          | arccosh()                          | 14    |
| ©, kommentar                                     | 244      | arccosinus, cos <sup>-1</sup> ()   | 32    |
| •                                                |          | arccot()                           | 14    |
| °, grader/minuter/sekunder[*]                    | 238      | arccoth()                          | 15    |
| °, gradnotation[*]                               | 237      | arccsc()                           | 15    |
| #                                                |          | arccsch()                          | 15    |
| Ob, binär indikator                              | 245      | arcLen(), båglängd                 | 15    |
|                                                  |          | arcsec()                           | 15    |
|                                                  |          | arcsech()                          | 15    |
|                                                  |          | arcsin()                           | 15    |
|                                                  |          | arcsinh()                          | 15    |
|                                                  |          | arcsinus, sin <sup>-1</sup> ()     | 175   |
|                                                  |          | arctan()                           | 16    |
|                                                  |          | arctangens, tan <sup>-1</sup> ()   | 190   |
|                                                  |          | artanh()                           | 16    |
|                                                  |          | argument i TVM-funktioner          | 203   |
|                                                  |          | augment(), sammanfoga/slå ihop     | 16    |
|                                                  |          | avgRC(), genomsnittlig             | 16    |



|                                         |          |                                                           |     |
|-----------------------------------------|----------|-----------------------------------------------------------|-----|
| ändringsfrekvens .....                  |          | completeSquare( ), complete square .....                  | 28  |
| avrunda, round( ) .....                 | 161      | conj( ), komplexkonjugat .....                            | 29  |
| avsluta                                 |          | constructMat( ), konstruera matris .....                  | 29  |
| om, EndIf .....                         | 90       | corrMat( ), korrelationsmatris .....                      | 30  |
| avsluta om, EndIf .....                 | 90       | cos( ), cosinus .....                                     | 31  |
| avsluta, Exit .....                     | 65       | cos <sup>-1</sup> ( ), arccosinus .....                   | 32  |
|                                         |          | cosh( ), hyperbolisk cosinus .....                        | 33  |
| <b>B</b>                                |          | cosh <sup>-1</sup> ( ), hyperbolisk arccosinus .....      | 33  |
| båglängd, arcLen( ) .....               | 15       | cosine                                                    |     |
| Begär .....                             | 155      | visning i termer av .....                                 | 30  |
| beräkning, ordning vid .....            | 265      | cosinus, cos( ) .....                                     | 31  |
| bestämd integral                        |          | cot( ), cotangens .....                                   | 34  |
| mall .....                              | 6        | cot <sup>-1</sup> ( ), arccotangens .....                 | 34  |
| bibliotek                               |          | cotangens, cot( ) .....                                   | 34  |
| skapa genvägar till objekt .....        | 100      | coth( ), hyperbolisk cotangens .....                      | 35  |
| binär                                   |          | coth <sup>-1</sup> ( ), hyperbolisk arccotangens .....    | 35  |
| indikator, Ob .....                     | 245      | count( ), räkna poster i en lista .....                   | 35  |
| visa, ►Base2 .....                      | 18       | countif( ), villkorligt räkna poster i en                 |     |
| binomCdf( ) .....                       | 20, 96   | lista .....                                               | 36  |
| binomPdf( ) .....                       | 20       | cPolyRoots() .....                                        | 37  |
| blandade bråk, använda propFrac( )      |          | crossP( ), vektorprodukt .....                            | 37  |
| med .....                               | 144      | csc( ), cosekant .....                                    | 37  |
| Booleska operatorer                     |          | csc <sup>-1</sup> ( ), invers cosekant .....              | 38  |
| ⇒ .....                                 | 228, 263 | csch( ), hyperbolisk cosekant .....                       | 38  |
| ⇔ .....                                 | 229      | csch <sup>-1</sup> ( ), invers hyperbolisk cosekant ..... | 38  |
| and .....                               | 9        | cSolve( ), lösa komplext .....                            | 39  |
| eller .....                             | 134      | CubicReg, kubisk regression .....                         | 42  |
| icke .....                              | 130      | cumulativeSum( ), kumulativ summa .....                   | 43  |
| negerad .....                           | 124      | Cycle, cykel .....                                        | 43  |
| nor .....                               | 129      | cykel, Cycle .....                                        | 43  |
| xor .....                               | 209      | cZeros( ), komplexa nollor .....                          | 44  |
| bråk                                    |          |                                                           |     |
| mall .....                              | 1        | <b>D</b>                                                  |     |
| propFrac .....                          | 144      | d( ), förstaderivata .....                                | 230 |
| <b>C</b>                                |          | days between dates (dagar mellan                          |     |
| Cdf( ) .....                            | 72       | datum), dbd( ) .....                                      | 46  |
| ceiling( ), tak .....                   | 21       | dbd( ), days between dates (dagar                         |     |
| centralDiff( ) .....                    | 21       | mellan datum) .....                                       | 46  |
| cFactor( ), komplex faktor .....        | 22       | decimal                                                   |     |
| char( ), teckensträng .....             | 23       | heltalsvisning, ►Base10 .....                             | 19  |
| charPoly( ) .....                       | 23       | vinkelvisning, ►DD .....                                  | 46  |
| ClearAZ .....                           | 25       | Define .....                                              | 47  |
| ClrErr, rensa fel .....                 | 26       | Define LibPriv .....                                      | 48  |
| colAugment .....                        | 26       | Define LibPub .....                                       | 49  |
| colDim( ), matriskolumnndimension ..... | 26       | Define, define .....                                      | 47  |
| colNorm( ), matrixskolumnnorm .....     | 26       | define, Define .....                                      | 47  |
| comDenom( ), gemensam nämnare .....     | 27       | defining                                                  |     |
|                                         |          | private function or program .....                         | 48  |

|                                                |         |                                         |         |
|------------------------------------------------|---------|-----------------------------------------|---------|
| public function or program .....               | 49      | eller, Boolesk operator .....           | 134     |
| dela heltal, intDiv() .....                    | 94      | else if, Elseif .....                   | 61      |
| deltmatris, subMat( ) .....                    | 189     | Elseif, else if .....                   | 61      |
| deltalist() .....                              | 49      | end .....                               |         |
| deltaTmpCnv() .....                            | 50      | for, EndFor .....                       | 75      |
| DelVar, ta bort variabel .....                 | 50      | funktion, EndFunc .....                 | 79      |
| delVoid( ), ta bort tomma element .....        | 50      | end function, EndFunc .....             | 79      |
| derivata .....                                 |         | EndTry, slut försök .....               | 199     |
| förstaderivata, d( ) .....                     | 230     | EndWhile, slut medan .....              | 209     |
| numerisk derivata, nDeriv( ) .....             | 127     | enheter .....                           |         |
| numerisk derivata, nDerivative( ) .....        | 126     | konvertera .....                        | 240     |
| derivata eller n:te derivata .....             |         | enhetsvektor, unitV( ) .....            | 205     |
| mall .....                                     | 6       | envariabelstatistik, OneVar .....       | 133     |
| derivative() .....                             | 50      | EOS (Ekvationsoperativsystem) .....     | 265     |
| deSolve( ), lösning .....                      | 51      | euler( ), Euler function .....          | 62      |
| det( ), matrisdeterminant .....                | 53      | exact( ), exakt .....                   | 65      |
| diag( ), diagonal matris .....                 | 53      | exakt, exact( ) .....                   | 65      |
| dim( ), dimension .....                        | 53      | Exit, avsluta .....                     | 65      |
| dimension, dim( ) .....                        | 53      | exp( ), e till en potens .....          | 66      |
| Disp, visa data .....                          | 54, 164 | expList( ), uttryck till lista .....    | 66      |
| DispAt .....                                   | 54      | expand( ), expandera .....              | 67      |
| dividera, / .....                              | 221     | expandera, expand( ) .....              | 67      |
| dominant term, dominantTerm( ) .....           | 57      | exponent, E .....                       | 236     |
| dominantTerm( ), dominant term .....           | 57      | exponenter .....                        |         |
| dotP( ), skalärprodukt .....                   | 59      | mall .....                              | 1       |
| <b>E</b>                                       |         | exponentiell regression, ExpReg .....   | 69      |
| e exponent .....                               |         | expr( ), sträng till uttryck .....      | 69, 112 |
| mall .....                                     | 2       | ExpReg, exponentiell regression .....   | 69      |
| e till en potens, e^( ) .....                  | 59, 66  | <b>F</b>                                |         |
| E, exponent .....                              | 236     | factor( ), faktor .....                 | 70      |
| e, visning av uttryck i termer av .....        | 66      | faktor, factor( ) .....                 | 70      |
| e^( ), e till en potens .....                  | 59      | fakultet, ! .....                       | 229     |
| eff ), konvertera nominell till effektiv ..... |         | fel och felsökning .....                |         |
| ränta .....                                    | 60      | flytta fel, PassErr .....               | 137     |
| effektiv ränta, eff( ) .....                   | 60      | rensa fel, ClrErr .....                 | 26      |
| egentligt bråk, propFrac .....                 | 144     | Felkoder och meddelanden .....          | 280     |
| egenvärde, eigVl( ) .....                      | 61      | Fill, fylla matris .....                | 72      |
| egenvektor, eigVc( ) .....                     | 60      | finansiella funktioner, tvnFV( ) .....  | 201     |
| eigVc( ), egenvektor .....                     | 60      | finansiella funktioner, tvnI( ) .....   | 202     |
| eigVl( ), egenvärde .....                      | 61      | finansiella funktioner, tvnN( ) .....   | 202     |
| Ekvationsoperativsystem (EOS) .....            | 265     | finansiella funktioner, tvnPmt( ) ..... | 202     |
| ekvationssystem (2 ekvationer) .....           |         | finansiella funktioner, tvnPv( ) .....  | 203     |
| mall .....                                     | 3       | FiveNumSummary .....                    | 72      |
| ekvationssystem (N ekvationer) .....           |         | floor( ), golv .....                    | 73      |
| mall .....                                     | 3       | flytta fel, PassErr .....               | 137     |
| eller (boolesk), eller .....                   | 134     | fMax( ), funktionsmaximum .....         | 74      |
|                                                |         | fMin( ), funktionsminimum .....         | 74      |
|                                                |         | For .....                               | 75      |



|                                    |     |                                      |          |
|------------------------------------|-----|--------------------------------------|----------|
| <b>I</b>                           |     | lösa, cSolve( )                      | 39       |
| icke, Booleska operatorer          | 130 | konstant                             |          |
| identitetsmatris, identity( )      | 90  | i solve( )                           | 179      |
| identity( ), identitetsmatris      | 90  | konstanter                           |          |
| lf, om                             | 90  | genvägar för                         | 263      |
| iffn( )                            | 92  | i cSolve( )                          | 41       |
| imag( ), imaginärdel               | 92  | i cZeros( )                          | 45       |
| imaginärdel, imag()                | 92  | i deSolve( )                         | 51       |
| ImpDiff( ), implicit derivata      | 93  | i solve( )                           | 181      |
| implicit derivata, Impdif( )       | 93  | konstruera matris, constructMat( )   | 29       |
| indata, Input                      | 93  | konvertera                           |          |
| indirection, #                     | 236 | 4Grad                                | 90       |
| indirection, operator (#)          | 266 | enheter                              | 240      |
| inom sträng, inString()            | 93  | kopiera variabel eller funktion,     |          |
| Input, indata                      | 93  | CopyVar                              | 29       |
| inställningar, hämta aktuella      | 87  | korrelationsmatris, corrMat( )       | 30       |
| inString( ), inom sträng           | 93  | kortkommandon                        | 263      |
| int( ), heltal                     | 94  | kortkommandon, tangentbord           | 263      |
| intDiv( ), dela heltal             | 94  | kubisk regression, CubicReg          | 42       |
| inte lika med, ≠                   | 226 | kumulativ summa, cumulativeSum( )    | 43       |
| integrera, %∞                      | 231 | kvadratisk regression, QuadReg       | 146      |
| interpolate( ), interpolera        | 94  | kvadratrot                           |          |
| invers kumulativ normalfördelning, |     | mall                                 | 1        |
| invNorm( )                         | 96  | kvadratrot, v( )                     | 183, 232 |
| invers, $\wedge^{-1}$              | 241 | kvartär regression, QuartReg         | 147      |
| inverterat värde, $\wedge^{-1}$    | 241 | <b>L</b>                             |          |
| invF( )                            | 95  | lägen                                |          |
| invNorm( ), invers kumulativ       |     | inställning, setMode( )              | 169      |
| normalfördelning                   | 96  | lägesinställningar, getMode( )       | 87       |
| invf( )                            | 97  | lägg till, &                         | 230      |
| Inv $\chi^2$ ( )                   | 95  | lagra                                |          |
| iPart( ), heltalsdel               | 97  | symbol, &                            | 243-244  |
| irr( ), internränta                |     | längd på sträng                      | 53       |
| internal rate of return, irr( )    | 97  | läsa upp variabler eller             |          |
| isPrime( ), primtest               | 98  | variabelgrupper                      | 206      |
| isVoid( ), testa om sant           | 98  | läsa variabler eller variabelgrupper | 111      |
| <b>K</b>                           |     | Låsa, låsa variabel eller            |          |
| kombinationer, nCr( )              | 125 | variabelgrupp                        | 111      |
| Kommandot Wait                     | 207 | Lbl, märka                           | 99       |
| kommentar, ©                       | 244 | lcm, minsta gemensamma multipel      | 99       |
| komplex                            |     | left( ), vänster                     | 99       |
| faktor, cFactor( )                 | 22  | LibPriv                              | 48       |
| komplexa                           |     | LibPub                               | 49       |
| nollor, cZeros( )                  | 44  | libShortcut( ), skapa genvägar till  |          |
| komplext                           |     | biblioteksobjekt                     | 100      |
| konjugat, conj( )                  | 29  | lika med, =                          | 225      |
|                                    |     | limit( ) eller lim( ), gränsvärde    | 101      |

|                                         |          |                                      |     |
|-----------------------------------------|----------|--------------------------------------|-----|
| linjär regression, LinRegAx .....       | 103      | lösning, deSolve( ) .....            | 51  |
| linjär regression, LinRegBx .....       | 102, 104 | LU, matris undre-övre uppdelning ..  | 115 |
| LinRegBx, linjär regression .....       | 102      |                                      |     |
| LinRegMx, linjär regression .....       | 103      | <b>M</b>                             |     |
| LinRegtIntervals, linjär regression ... | 104      | mallar                               |     |
| LinRegtTest .....                       | 105      | absolutbelopp .....                  | 3-4 |
| linSolve() .....                        | 107      | andradderivata .....                 | 6   |
| list►mat( ), lista till matris .....    | 108      | bestämd integral .....               | 6   |
| lista till matris, list►mat( ) .....    | 108      | bråk .....                           | 1   |
| lista, räkna poster .....               | 35       | derivata eller n:te derivata .....   | 6   |
| lista, villkorligt räkna poster .....   | 36       | e exponent .....                     | 2   |
| listor                                  |          | ekvationssystem (2 ekvationer) ..... | 3   |
| kumulativ summa,                        |          | ekvationssystem (N ekvationer) ..... | 3   |
| cumulativeSum( ) .....                  | 43       | exponent .....                       | 1   |
| lista till matris, list►mat( ) .....    | 108      | förstaderivata .....                 | 5   |
| matris till lista, mat►list( ) .....    | 116      | gränsvärde .....                     | 6   |
| maximum, max( ) .....                   | 116      | kvadratrot .....                     | 1   |
| med tomma element .....                 | 261      | Log .....                            | 2   |
| minimum, min( ) .....                   | 120      | matris (1 × 2) .....                 | 4   |
| mittsträng, mid( ) .....                | 119      | matris (2 × 1) .....                 | 4   |
| ny, newList( ) .....                    | 126      | matris (2 × 2) .....                 | 4   |
| produkt, product( ) .....               | 144      | matris (m × n) .....                 | 4   |
| sammanfoga/slå ihop, augment(           |          | n:te rot .....                       | 1   |
| ) .....                                 | 16       | obestämd integral .....              | 6   |
| skalärprodukt, dotP( ) .....            | 59       | produkt (P) .....                    | 5   |
| skillnad, @list( ) .....                | 107      | stegvis funktion (2 steg) .....      | 2   |
| skillnader i en lista, @list( ) .....   | 107      | stegvis funktion (N steg) .....      | 3   |
| sortera fallande, SortD .....           | 182      | summa (G) .....                      | 5   |
| sortera stigande, SortA .....           | 182      | märka, Lbl .....                     | 99  |
| summering, sum( ) .....                 | 188      | mat►list( ), matris till lista ..... | 116 |
| uttryck till lista, exp►list( ) .....   | 66       | matris (1 × 2)                       |     |
| vektorprodukt, crossP( ) .....          | 37       | mall .....                           | 4   |
| ln( ), naturlig logaritm .....          | 108      | matris (2 × 1)                       |     |
| LnReg, logaritmisk regression .....     | 109      | mall .....                           | 4   |
| Local, lokal variabel .....             | 110      | matris (2 × 2)                       |     |
| Log                                     |          | mall .....                           | 4   |
| mall .....                              | 2        | matris (m × n)                       |     |
| logaritmer .....                        | 108      | mall .....                           | 4   |
| logaritmisk regression, LnReg .....     | 109      | matris till lista, mat►list( ) ..... | 116 |
| logisk dubbel implikation, ⇔ .....      | 229      | matriser                             |     |
| logisk implikation, ⇒ .....             | 228, 263 | delmatris, subMat( ) .....           | 189 |
| Logistic, logistisk regression .....    | 112      | determinant, det( ) .....            | 53  |
| LogisticD, logistisk regression .....   | 113      | diagonal, diag( ) .....              | 53  |
| logistisk regression, Logistic .....    | 112      | dimension, dim( ) .....              | 53  |
| logistisk regression, LogisticD .....   | 113      | egenvärde, eigVl( ) .....            | 61  |
| lokal variabel, Local .....             | 110      | egenvektor, eigVc( ) .....           | 60  |
| lokal, Local .....                      | 110      | fylla, Fill .....                    | 72  |
| Loop, slinga .....                      | 115      | identitet, identity( ) .....         | 90  |
| lös, solve( ) .....                     | 178      | kolumndimension, colDim( ) ...       | 26  |

|                                               |     |                                                        |     |
|-----------------------------------------------|-----|--------------------------------------------------------|-----|
| kolumnnorm, colNorm( )                        | 26  | minutnotation, .....                                   | 238 |
| kumulativ summa,<br>cumulativeSum( )          | 43  | mirr( ), modifierad internränta                        | 120 |
| lista till matris, listMat( )                 | 108 | mittsträng, mid( )                                     | 119 |
| matris till lista, matList( )                 | 116 | mod( ), modul                                          | 121 |
| maximum, max( )                               | 116 | modifierad internränta, mirr( )                        | 120 |
| minimum, min( )                               | 120 | modul, mod( )                                          | 121 |
| ny, newMat( )                                 | 126 | mRow( ), matrisradoperation                            | 121 |
| produkt, product( )                           | 144 | mRowAdd( ),<br>matrisradmultiplikation och<br>addition | 122 |
| punkt addition, .+                            | 223 | Multipelt linjärt regression-t-test                    | 123 |
| punkt division, .P                            | 224 | multiplitera, *                                        | 220 |
| punkt multiplikation, .*                      | 224 | MultReg                                                | 122 |
| punkt potens, .^                              | 224 | MultRegIntervals( )                                    | 122 |
| punkt subtraktion, .N                         | 223 | MultRegTests( )                                        | 123 |
| QR-faktorisering, QR                          | 145 |                                                        |     |
| radaddition, rowAdd( )                        | 161 |                                                        |     |
| raddimension, rowDim( )                       | 162 | <b>N</b>                                               |     |
| radmultiplikation och addition,<br>mRowAdd( ) | 122 | n:te rot                                               |     |
| radnorm rowNorm( )                            | 162 | mall                                                   | 1   |
| radoperation, mRow( )                         | 121 | nämnare                                                | 27  |
| reducerad radtrappstegsform,<br>rref( )       | 162 | nand, Boolesk operator                                 | 124 |
| sammanfoga/slå ihop, augment( )               | 16  | när, when( )                                           | 208 |
| slump, randMat( )                             | 150 | naturlig logaritm, ln( )                               | 108 |
| summering, sum( )                             | 188 | nCr( ), kombinationer                                  | 125 |
| transponera, T                                | 189 | nDerivative( ), numerisk derivata                      | 126 |
| trappstegsform, ref( )                        | 153 | negation, skriva in negativa tal                       | 266 |
| undermatris, subMat( )                        | 187 | nettovärde, npv( )                                     | 132 |
| undre-övre uppdelning, LU                     | 115 | newList( ), ny lista                                   | 126 |
| växla matrisrad, rowSwap( )                   | 162 | newMat( ), ny matris                                   | 126 |
| max( ), maximum                               | 116 | nfMax( ), numeriskt<br>funktionsmaximum                | 127 |
| maximum, max( )                               | 116 | nfMin( ), numeriskt<br>funktionsminimum                | 127 |
| mean( ), medelvärde                           | 117 | nInt( ), numerisk integral                             | 128 |
| med,                                          | 242 | nollställen, zeroes( )                                 | 210 |
| medan, While                                  | 209 | nom ), konvertera effektiv till<br>nominell ränta      | 128 |
| medelvärde, mean( )                           | 117 | nominell ränta, nom( )                                 | 128 |
| median( ), median                             | 117 | nor, Booleska operatorer                               | 129 |
| median, median( )                             | 117 | norm( ), Frobenius norm                                | 129 |
| medium-medium-linjeregession,<br>MedMed       | 118 | normal, normalLine( )                                  | 130 |
| MedMed, medium-medium-<br>linjeregession      | 118 | normalLine( )                                          | 130 |
| mid( ), mittsträng                            | 119 | normCdf( )                                             | 130 |
| min( ), minimum                               | 120 | normPdf( )                                             | 130 |
| mindre än eller lika med, {                   | 227 | nPr( ), permutationer                                  | 131 |
| minimum, min( )                               | 120 | npv( ), nettovärde                                     | 132 |
| minsta gemensamma multipel, lcm               | 99  | nSolve( ), numerisk lösning                            | 132 |

|                                    |         |                                |                    |
|------------------------------------|---------|--------------------------------|--------------------|
| numeric                            |         | polynom                        |                    |
| derivata, nDeriv( )                | 127     | slump, randPoly( )             | 151                |
| numerisk                           |         | utvärdera, polyEval( )         | 140                |
| derivata, nDeriv( )                | 127     | PolyRoots()                    | 142                |
| derivata, nDerivative( )           | 126     | potens, ^                      | 222                |
| integral, nInt( )                  | 128     | potensregression,              |                    |
| lösning, nSolve( )                 | 132     | PowerReg                       | 142, 155, 157, 194 |
| ny                                 |         | PowerReg, potensregression     | 142                |
| lista, newList( )                  | 126     | Prgm, definiera program        | 143                |
| matris, newMat( )                  | 126     | prim,                          | 239                |
| nygradsnotation, g                 | 236     | primtalstest, isPrime( )       | 98                 |
|                                    |         | procent, %                     | 225                |
| <b>O</b>                           |         | prodSeq()                      | 144                |
| obestämd integral                  |         | product( ), produkt            | 144                |
| mall                               | 6       | produkt (P)                    |                    |
| objekt                             |         | mall                           | 5                  |
| skapa genvägar till bibliotek      | 100     | produkt, product( )            | 144                |
| om, If                             | 90      | produkt, Π( )                  | 233                |
| område( ), områdesfunktion         | 57      | program och programmering      |                    |
| områdesfunktion, område( )         | 57      | försök, Try                    | 199                |
| omvandla                           |         | rensa fel, ClrErr              | 26                 |
| ►Rad                               | 149     | slut försök, EndTry            | 199                |
| OneVar, envariabelstatistik        | 133     | slut program, EndPrgm          | 143                |
| operatorer                         |         | visa I/O- skärm, Disp          | 164                |
| ordning vid beräkning              | 265     | visa I/O-fönster, Disp         | 54                 |
| operatorn begränsning "   "        | 242     | programmera                    |                    |
| operatorn begränsning, ordning vid |         | definiera program, Prgm        | 143                |
| utvärdering                        | 265     | flytta fel, PassErr            | 137                |
| ord( ), numerisk teckenkod         | 136     | visa data, Disp                | 54                 |
| ortogonalvektorvisning, ►Rect      | 152     | programmering                  |                    |
|                                    |         | visa data, Disp                | 164                |
| <b>P</b>                           |         | programs                       |                    |
| P►Rx( ), rektangulär x-koordinat   | 136     | defining private library       | 48                 |
| P►Ry( ), rektangulär y-koordinat   | 136     | defining public library        | 49                 |
| PassErr, flytta fel                | 137     | propFrac, egentligt bråk       | 144                |
| Pdf( )                             | 77      | punkt                          |                    |
| permutationer, nPr( )              | 131     | addition, .+                   | 223                |
| piecewise( )                       | 137     | division, .P                   | 224                |
| poissCdf( )                        | 137     | multiplikation, *              | 224                |
| poissPdf( )                        | 138     | potens, .^                     | 224                |
| polär                              |         | subtraktion, .N                | 223                |
| visa vektor, ►Polar                | 138     |                                |                    |
| polära                             |         | <b>Q</b>                       |                    |
| koordinater, R►Pθ( )               | 148-149 | QR, QR-faktorisering           | 145                |
| polyCoef( )                        | 139     | QR-faktorisering, QR           | 145                |
| polyDegree( )                      | 139     | QuadReg, kvadratisk regression | 146                |
| polyEval( ), utvärdera polynom     | 140     | QuartReg, kvartär regression   | 147                |
| polyGcd( )                         | 140-141 |                                |                    |

| R                                                            |                    |  |
|--------------------------------------------------------------|--------------------|--|
| R, radian                                                    | 237                |  |
| R►Pr( ), polära koordinater                                  | 149                |  |
| R►Pθ( ), polära koordinater                                  | 148                |  |
| radian, R                                                    | 237                |  |
| räkna poster i en lista villkorligt ,<br>countif( )          | 36                 |  |
| räkna poster i en lista, count( )                            | 35                 |  |
| rand( ), slumpstal                                           | 149                |  |
| randBin, slumpstal                                           | 149                |  |
| randInt( ), slumpheftal                                      | 150                |  |
| randMat( ), slumpmatris                                      | 150                |  |
| randNorm( ), slumpnorm                                       | 150                |  |
| randPoly( ), slumpppolynom                                   | 151                |  |
| randSamp( )                                                  | 151                |  |
| RandSeed, slumptalsfrö                                       | 151                |  |
| real( ), reell                                               | 151                |  |
| reducerad radtrappstegsform, rref()                          | 162                |  |
| reell, real()                                                | 151                |  |
| ref( ), trappstegsform                                       | 153                |  |
| RefreshProbeVars                                             | 154                |  |
| regression                                                   |                    |  |
| exponentiell, ExpReg                                         | 69                 |  |
| kubisk, CubicReg                                             | 42                 |  |
| regressioner                                                 |                    |  |
| kvadratisk, QuadReg                                          | 146                |  |
| kvartär, QuartReg                                            | 147                |  |
| linjär regression, LinRegAx                                  | 103                |  |
| linjär regression, LinRegBx                                  | 102, 104           |  |
| logaritmisk, LnReg                                           | 109                |  |
| Logistic                                                     | 112                |  |
| logistisk, Logistic                                          | 113                |  |
| medium-medium-linje,<br>MedMed                               | 118                |  |
| MultReg                                                      | 122                |  |
| potensregression,<br>PowerReg                                | 142, 155, 157, 194 |  |
| sinus, SinReg                                                | 177                |  |
| rektangulär x-koordinat, P►Rx( )                             | 136                |  |
| rektangulär y-koordinat, P►Ry( )                             | 136                |  |
| remain( ), rest                                              | 155                |  |
| rensa                                                        |                    |  |
| fel, ClrErr                                                  | 26                 |  |
| Rensa                                                        | 249                |  |
| RequestStr                                                   | 157                |  |
| rest, remain( )                                              | 155                |  |
| result                                                       |                    |  |
| visning i termer av cosinus                                  | 30                 |  |
| visning i termer av e                                        | 66                 |  |
| visning i termer av sinus                                    | 173                |  |
| resultat, statistik                                          | 184                |  |
| resultvärden, statistik                                      | 185                |  |
| retur, Return                                                | 157                |  |
| Return, retur                                                | 157                |  |
| right( ), höger                                              | 158                |  |
| right, right( )                                              | 28, 62, 208        |  |
| rita                                                         | 250-252            |  |
| rk23( ), Runge Kutta-funktion                                | 158                |  |
| rotate( ), rotera                                            | 159-160            |  |
| rotera, rotate( )                                            | 159-160            |  |
| round( ), avrunda                                            | 161                |  |
| rowAdd( ), matrisradaddition                                 | 161                |  |
| rowDim( ), matrisraddimension                                | 162                |  |
| rowNorm( ), matrisradnorm                                    | 162                |  |
| rowSwap( ), växla matrisrad                                  | 162                |  |
| rref( ), reducerad trappstegsform                            | 162                |  |
| S                                                            |                    |  |
| sammanfoga/slå ihop, augment( )                              | 16                 |  |
| samtidiga ekvationer, simult( )                              | 172                |  |
| sannolik tæthet, normPdf( )                                  | 130                |  |
| sannolikhet för normalfördelning,<br>normCdf( )              | 130                |  |
| sannolikhet för student-t-fördelning,<br>tCdf( )             | 193                |  |
| sec( ), sekansfunktion                                       | 163                |  |
| sec <sup>-1</sup> ( ), invers sekansfunktion                 | 163                |  |
| sech( ), hyperbolisk sekansfunktion                          | 164                |  |
| sech <sup>-1</sup> ( ), invers hyperbolisk<br>sekansfunktion | 164                |  |
| sekundnotation, "                                            | 238                |  |
| seq( ), talföljd                                             | 165                |  |
| seqGen( )                                                    | 165                |  |
| seqn( )                                                      | 166                |  |
| sequence, seq( )                                             | 165-166            |  |
| series( ), series                                            | 167                |  |
| series, series( )                                            | 167                |  |
| setMode( ), ställ in läge                                    | 169                |  |
| sfärisk vektorvisning, ►Sphere                               | 182                |  |
| shift( ), skifta                                             | 171                |  |
| sign( ), tecken                                              | 172                |  |
| simult( ), samtidiga ekvationer                              | 172                |  |
| sin( ), sinus                                                | 174                |  |
| sin <sup>-1</sup> ( ), arcsinus                              | 175                |  |
| sine                                                         |                    |  |
| visning av uttryck i termer av                               | 173                |  |



|                                                          |              |                                                          |             |
|----------------------------------------------------------|--------------|----------------------------------------------------------|-------------|
| <code>sinh( )</code> , hyperbolisk sinus                 | 176          | <code>population</code>                                  |             |
| <code>sinh<sup>-1</sup>( )</code> , hyperbolisk arcsinus | 176          | <code>stdDevSamp( )</code> , standardavvikelse för urval | 186         |
| <code>SinReg</code> , sinusregression                    | 177          | <code>stegvis funktion (2 steg)</code>                   |             |
| <code>sinus</code> , <code>sin( )</code>                 | 174          | mall                                                     | 2           |
| <code>sinusregression</code> , <code>SinReg</code>       | 177          | <code>stegvis funktion (N steg)</code>                   |             |
| skalär                                                   |              | mall                                                     | 3           |
| produkt, <code>dotP( )</code>                            | 59           | Stoppkommando                                            | 187         |
| skifta, <code>shift( )</code>                            | 171          | större än eller lika med, <code> </code>                 | 228         |
| slinga, <code>Loop</code>                                | 115          | större än, <code>&gt;</code>                             | 227         |
| slump                                                    |              | största gemensamma delare, <code>gcd( )</code>           | 80          |
| matris, <code>randMat( )</code>                          | 150          | sträng                                                   |             |
| norm, <code>randNorm( )</code>                           | 150          | dimension, <code>dim( )</code>                           | 53          |
| polynom, <code>randPoly( )</code>                        | 151          | längd                                                    | 53          |
| slumpfrö, <code>RandSeed</code>                          | 151          | strängar                                                 |             |
| slump sampel                                             | 151          | använda för att skapa                                    |             |
| slut                                                     |              | variabelnamn                                             | 266         |
| försök, <code>EndTry</code>                              | 199          | format, <code>format( )</code>                           | 76          |
| medan, <code>EndWhile</code>                             | 209          | formatera                                                | 76          |
| program, <code>EndPrgrm</code>                           | 143          | höger, <code>right( )</code>                             | 94, 158     |
| slinga, <code>EndLoop</code>                             | 115          | indirection, <code>#</code>                              | 236         |
| slut medan, <code>EndWhile</code>                        | 209          | inom, <code>inString</code>                              | 93          |
| slut slinga, <code>EndLoop</code>                        | 115          | lägg till, <code>&amp;</code>                            | 230         |
| <code>solve( )</code> , lös                              | 178          | mittsträng, <code>mid( )</code>                          | 119         |
| <code>SortA</code> , sortera stigande                    | 182          | numerisk teckenkod, <code>ord( )</code>                  | 136         |
| <code>SortD</code> , sortera fallande                    | 182          | rotera, <code>rotate( )</code>                           | 159-160     |
| sortera                                                  |              | skifta, <code>shift( )</code>                            | 171         |
| fallande, <code>SortD</code>                             | 182          | sträng till uttryck, <code>expr( )</code>                | 69, 112     |
| stigande, <code>SortA</code>                             | 182          | teckensträng, <code>char( )</code>                       | 23          |
| språk                                                    |              | uttryck till sträng, <code>string( )</code>              | 187         |
| hämta språkinformation                                   | 86           | vänster, <code>left( )</code>                            | 99          |
| <code>sqrt( )</code> , kvadratroten                      | 183          | <code>string( )</code> , uttryck till sträng             | 187         |
| ställ in                                                 |              | strings                                                  |             |
| läge, <code>setMode( )</code>                            | 169          | right, <code>right( )</code>                             | 28, 62, 208 |
| standardavvikelse, <code>stdDev( )</code>                | 185-186, 206 | <code>subMat( )</code> , delmatris                       | 189         |
| <code>stat.results</code>                                | 184          | <code>subMat( )</code> , undermatris                     | 187         |
| <code>stat.values</code>                                 | 185          | substitution med <code>" "</code> -operatören            | 242         |
| statistik                                                |              | subtrahera, <code>-</code>                               | 219         |
| envariabelstatistik, <code>OneVar</code>                 | 133          | <code>sum( )</code> , summering                          | 188         |
| fakultet, <code>!</code>                                 | 229          | <code>sumlf( )</code>                                    | 188         |
| kombinationer, <code>nCr( )</code>                       | 125          | summa (G)                                                |             |
| medelvärde, <code>mean( )</code>                         | 117          | mall                                                     | 5           |
| median, <code>median( )</code>                           | 117          | summa av kapitalbetalningar                              | 235         |
| permutationer, <code>nPr( )</code>                       | 131          | summa av räntebetalningar                                | 234         |
| slump norm, <code>randNorm( )</code>                     | 150          | summa, $\Sigma( )$                                       | 233         |
| slumptionsfrö, <code>RandSeed</code>                     | 151          | summering, <code>sum( )</code>                           | 188         |
| standardavvikelse, <code>stdDev( )</code>                | 185-186, 206 | <code>sumSeq()</code>                                    | 189         |
| tvåvariabelresultat, <code>TwoVar</code>                 | 203          | svar (sista), <code>Ans</code>                           | 13          |
| varians, <code>variance( )</code>                        | 206          |                                                          |             |
| <code>stdDevPop( )</code> , standardavvikelse för        | 185          |                                                          |             |



|                                   |         |  |  |
|-----------------------------------|---------|--|--|
| vektorer                          |         |  |  |
| enhet, unitV( )                   | 205     |  |  |
| skalärprodukt, dotP( )            | 59      |  |  |
| vektorprodukt, crossP( )          | 37      |  |  |
| visa cylindrisk vektor, ►Cylind   | 43      |  |  |
| vektorprodukt, crossP( )          | 37      |  |  |
| vinkel, angle( )                  | 10      |  |  |
| visa cylindrisk vektor, ►Cylind   | 43      |  |  |
| visa data, Disp                   | 54, 164 |  |  |
| visa som                          |         |  |  |
| binär, ►Base2                     | 18      |  |  |
| cylindrisk vektor, ►Cylind        | 43      |  |  |
| decimal vinkel, ►DD               | 46      |  |  |
| decimalt heltal, ►Base10          | 19      |  |  |
| grad/minut/sekund, ►DMS           | 56      |  |  |
| hexadecimal, ►Base16              | 19      |  |  |
| ortogonal vektor, ►Rect           | 152     |  |  |
| polär vektor, ►Polar              | 138     |  |  |
| sfärisk vektor, ►Sphere           | 182     |  |  |
| visning i grad/minut/sekund, ►DMS | 56      |  |  |
| void, testa om                    | 98      |  |  |
| <b>W</b>                          |         |  |  |
| warnCodes( ), Warning codes       | 208     |  |  |
| when( ), när                      | 208     |  |  |
| While, medan                      | 209     |  |  |
| <b>X</b>                          |         |  |  |
| x <sup>2</sup> , kvadrat          | 223     |  |  |
| XNOR                              | 229     |  |  |
| xor, Booleskt exklusivt eller     | 209     |  |  |
| <b>Z</b>                          |         |  |  |
| zeroes( ), nollställen            | 210     |  |  |
| zInterval, z konfidensintervall   | 213     |  |  |
| zInterval_1Prop, 1-proportion z-  |         |  |  |
| konfidensintervall                | 213     |  |  |
| zInterval_2Prop, 2-proportion z-  |         |  |  |
| konfidensintervall                | 214     |  |  |
| zInterval_2Samp, 2-sampel z-      |         |  |  |
| konfidensintervall                | 214     |  |  |
| zTest                             | 215     |  |  |
| zTest_1Prop, 1-proportion z-test  | 216     |  |  |
| zTest_2Prop, 2-proportion z-test  | 217     |  |  |
| zTest_2Samp, 2-sampel z-test      | 217     |  |  |
| <b>Δ</b>                          |         |  |  |
| Δlist( ), listskillnad            | 107     |  |  |
| ΔtmpCnv( ) [tmpCnv]               | 197     |  |  |
| <b>χ</b>                          |         |  |  |
| χ <sup>2</sup> 2way               | 23      |  |  |
| χ <sup>2</sup> Cdf( )             | 24      |  |  |
| χ <sup>2</sup> GOF                | 24      |  |  |
| χ <sup>2</sup> Pdf( )             | 25      |  |  |