

# TI-Nspire™ CX Referanseguide

## **Viktig Informasjon**

Dersom ikke annet er uttrykkelig nevnt i Lisensen som finnes vedlagt programmet, gir ikke Texas Instruments noen garanti, verken uttrykt eller underforstått, herunder, men ikke begrenset til noen impliserte garantier for salgbarhet og egnethet for et bestemt formål, med hensyn til noen som helst programmer eller bokmaterialer som kun er tilgjengelig på et "som det er"-grunnlag. Ikke i noen tilfeller kan Texas Instruments bli holdt ansvarlig overfor noen for spesielle, indirekte, tilfeldige eller følgeskader i forbindelse med eller som et resultat av anskaffelsen eller bruken av disse materialene. Texas Instruments' eneste og eksklusive ansvar, uten hensyn til aksjonsformen, kan ikke overstige den summen som er blitt fremsatt i lisensen for programmet. I tillegg kan ikke Texas Instruments bli holdt ansvarlig for noen krav av noe slag mot bruken av disse materialene av en annen part.

© 2023 Texas Instruments Incorporated

Faktiske produkter kan være litt annerledes enn på bilder.

## **Innhold**

<b>Uttrykkssjabloner</b> .....	<b>1</b>
<b>Alfabetisk oversikt</b> .....	<b>7</b>
A .....	7
B .....	16
C .....	20
D .....	37
E .....	46
F .....	54
G .....	61
I .....	72
L .....	79
M .....	95
N .....	103
O .....	112
P .....	115
Q .....	121
R .....	124
S .....	140
T .....	160
U .....	172
V .....	173
W .....	174
X .....	176
Z .....	177
<b>Symboler</b> .....	<b>184</b>
<b>TI-Nspire™ CX II – Tegnekommandoer</b> .....	<b>208</b>
Grafikkprogrammering .....	208
Grafikkskjerm .....	208
Standardvisning og innstillinger .....	209
Feilmeldinger på grafikkskjerm .....	210
Ugyldige kommandoer i grafikkmodus .....	210
C .....	211
D .....	212
F .....	215
G .....	217
P .....	218
S .....	220
U .....	222

<b>Tomme (åpne) elementer .....</b>	<b>223</b>
<b>Snarveier/hurtigtaster for å legge inn matematiske uttrykk .....</b>	<b>225</b>
<b>EOS™ (Ligningsoperativsystem)-hierarkiet .....</b>	<b>227</b>
<b>TI-Nspire CX II – TI-Basic programmeringsfunksjoner .....</b>	<b>229</b>
Auto-innrykk i Programmeringseditor .....	229
Forbedrede feilmeldinger for TI-Basic .....	229
<b>Konstanter og verdier .....</b>	<b>232</b>
<b>Feilkoder og feilmeldinger .....</b>	<b>233</b>
<b>Advarselskoder og -meldinger .....</b>	<b>241</b>
<b>Generell informasjon .....</b>	<b>243</b>
<b>Stikkordregister .....</b>	<b>244</b>

# Uttrykkssjabloner

Med uttrykkssjablonene er det enkelt å skrive inn uttrykk i standardisert, matematisk fremstilling. Når du setter inn en sjablon, kommer den til syne på kommandolinjen med små blokker i posisjoner der du kan legge inn elementer. En markør viser hvilke elementer du kan sette inn.

Bruk pilknappene eller trykk på **tab** for å bevege markøren til hvert elements posisjon, og skriv inn en verdi eller et uttrykk for elementet. Trykk på **enter** eller **ctrl enter** for å behandle uttrykket.

## Brøk-sjablon

**ctrl** **÷** taster



**Merk:** Se også / (divider), side 186.

Eksempel:

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

## Eksponent-sjablon

**^** tast



**Merk:** Skriv inn den første verdien, trykk på **^** og skriv så inn eksponenten. For å flytte markøren tilbake til grunnlinjen, trykk på høyre pil (**►**).

**Merk:** Se også ^ (potens), side 187.

Eksempel:

$$2^3 = 8$$

## Kvadratrot-sjablon

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



**Merk:** Se også  $\sqrt{}$  (kvadratrot), side 196.

Eksempel:

$$\sqrt{4} = 2$$
$$\sqrt{\{9,16,4\}} = \{3,4,2\}$$

## N-te rot-sjablon

**ctrl** **^** taster



**Merk:** Se også rot(), side 136.

Eksempel:

## N-te rot-sjablon

ctrl ^ taster

$\sqrt[3]{8}$	2
$\sqrt[3]{\{8,27,15\}}$	$\{2,3,2.46621\}$

## e eksponent-sjablon

e<sup>x</sup> tast

e<sup>□</sup>

e <sup>1</sup>	2.71828182846
----------------	---------------

Naturlig grunntall  $e$  opphøyd i en eksponent

**Merk:** Se også  $e^{\wedge}()$ , side 46.

## Logaritme-sjablon

ctrl 10<sup>x</sup> taster

log<sub>□</sub>(□)

Eksempel:

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

Beregner logaritme til et spesifisert grunntall. Hvis grunntallet er forhåndsinnstilt på 10, utelates grunntallet.

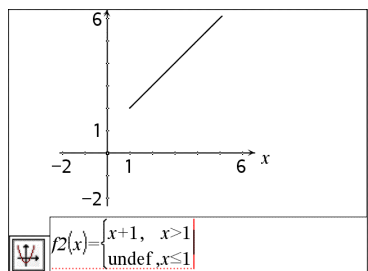
**Merk:** Se også  $\log()$ , side 90.

## Stykkevis sjablon (2-delers)

Katalog > 

{□,□  
□,□}

Eksempel:



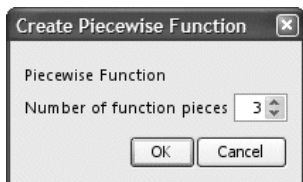
## Stykkevis sjablon (N-delers)

Katalog > 

Lar deg opprette uttrykk og betingelser for en N-delers stykkevis definert funksjon. Ber om  $N$ .

Eksempel:

Se eksemplet for Stykkevis sjablon (2-delers).



**Merk:** Se også `stykkevis()`, side 116.

## Sjablon for ligningssystemer med 2 ukjente

Katalog > 



Oppretter et system av to lineære ligninger. For å legge en rad til et eksisterende system, klikk inn sjablonen og gjenta sjablonen.

**Merk:** Se også `system()`, side 159.

Eksempel:

$$\text{solve} \left( \begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) \quad x = \frac{5}{2} \text{ and } y = \frac{-5}{2}$$

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

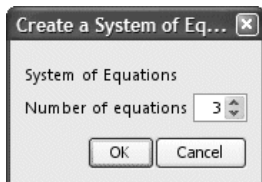
## Sjablon for ligningssystemer med N ukjente

Katalog > 

Lar deg opprette et system av  $N$  lineære ligninger. Ber om  $N$ .

Eksempel:


Se eksemplet for Sjabloner for ligningssystemer (2 ligninger).



**Merk:** Se også `system()`, side 159.

## Sjablon for absoluttverdi

Katalog > 

 Merk: Se også **abs()**, side 7.

Eksempel:

$$\left\{ 2, -3, 4, -4^3 \right\} \quad \left\{ 2, 3, 4, 64 \right\}$$

## gg°mm'ss.ss'' sjablon

Katalog > 



Lar deg sette inn vinkler i **gg° mm' ss.ss''** - format, der **gg** er antallet desimale grader, **mm** er antallet minutter og **ss.ss** er antallet sekunder.

Eksempel:

$$30^{\circ}15'10'' \quad 0.528011$$

## Matrise-sjablon (2 x 2)

Katalog > 



Oppretter en 2 x 2-matrise.

Eksempel:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 5 \quad \begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$$

## Matrise-sjablon (1 x 2)

Katalog > 



Eksempel:

$$\text{crossP}(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 4 \end{bmatrix}) \quad \begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$$

## Matrise-sjablon (2 x 1)

Katalog > 



Eksempel:

$$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

## Matrise-sjablon (m x n)

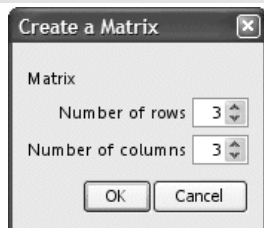
Katalog > 

Sjablonen kommer til syne etter at du er blitt bedt om å spesifisere antallet rader og kolonner.

Eksempel:

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





**Merk:** Hvis du oppretter en matrise med et stort antall rader og kolonner, må du muligens vente en liten stund før den vises på skjermen.

Sum-sjablon ( $\Sigma$ )

$$\sum_{i=1}^n ( )$$

Eksempel:

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

**Merk:** Se også  $\Sigma()$  (sumSeq), side 197.

Produkt-sjablon ( $\Pi$ )

$$\prod_{i=1}^n ( )$$

Eksempel:

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

**Merk:** Se også  $\Pi()$  (prodSeq), side 197.

## Første derivert-sjablon

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

Eksempel:

$$\frac{d}{dx} (|x|)_{x=0} \quad \text{undef}$$

Den første deriverte sjablonen kan brukes for å beregne førstederiverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.

**Merk:** Se også `d()` (derivert), side 195.

## Andre derivert-sjablon

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

Den andre deriverte sjablonen kan brukes for å beregne andrederiverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.

**Merk:** Se også `d()` (derivert), side 195.

Eksempel:

---


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


---

## Bestemt integral-sjablon

$$\int_a^b \square dx$$

Den bestemte integral-sjablonen kan brukes for å beregne den bestemte integralet numerisk ved hjelp av den samme metoden som `nInt()`.

**Merk:** Se også `nInt()`, side 107.

Eksempel:

---


$$\int_0^{10} x^2 dx \quad 333.333$$


---

## Alfabetisk oversikt

Elementer med navn som ikke er alfabetiske (som f.eks. +, !, og >) er opplistet på slutten av dette avsnittet (side 184). Hvis ikke annet er spesifisert, er alle eksemplene i dette avsnittet utført i grunninnstilling-modus, og det antas at ingen av variablene er definert.

### A

#### abs()

Katalog > 

**abs**(Verdi I) ⇒ verdi

$$\left\{ \left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\} \right\} \quad \{1.5708, 1.0472\}$$

**abs**(Liste I) ⇒ liste

$$|2-3 \cdot i| \quad 3.60555$$

**abs**(Matrise I) ⇒ matrise

Returnerer argumentets absoluttverdi.

**Merk:** Se også **Absoluttverdi-sjablon**, side 4.

Hvis argumentet er et komplekst tall, returneres absoluttverdien (modulus).

**Merk:** Alle ubestemte variabler behandles som reelle variabler.

#### amortTbl()

Katalog > 

**amortTbl**(NPmt, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [avrundVerdi]) ⇒ matrise

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

Amortiseringsfunksjon som returnerer en matrise som en amortiseringstabell for et sett med TVM-argumenter.

NPmt er antallet betalinger som skal inkluderes i tabellen. Tabellen starter med den første betalingen.

N, I, PV, Pmt, FV, PpY, CpY og PmtAt er beskrevet i tabellen med TVM-argumenter, side 170.

- Hvis du utelater Pmt, grunninnstilles den til  $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du utelater FV, grunninnstilles den til  $FV = 0$ .

- Grunninnstillingene for  $PpY$ ,  $CpY$  og  $PmtAt$  er de samme som for TVM-funksjonene.

*avrundVerdi* spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

Kolonnene i resultatmatrisen er i denne rekkefølgen: Betalingsnummer, betalt rentebeløp, betalt hovedbeløp og balanse.

Balansen som vises i rad  $n$  er balansen etter betaling  $n$ .

Du kan bruke resultatmatrisen som inndata for de andre amortiseringsfunksjonene  $\Sigma Int()$  og  $\Sigma Prn()$ , side 198, og **bal()**, side 16.

## and

*BoolskUttr1* and *BoolskUttr2*  $\Rightarrow$  *Boolsk uttrykk*

*Boolsk liste1* and *Boolsk liste2*  $\Rightarrow$  *Boolsk liste*

*Boolsk matrise1* and *Boolsk matrise2*  $\Rightarrow$  *Boolsk matrise*

Returnerer sann eller usann eller en forenklet form av opprinnelig uttrykk.

*Heltall1* and *Heltall2*  $\Rightarrow$  *heltall*

Sammenlikner to reelle heltall bit-for-bit med en and-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis en av bitene er 1; ellers er resultatet 0. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

I heksades grunntall-modus:

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

**Viktig:** Null, ikke bokstaven O.

I binær grunntall-modus:

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

I desimalt grunntall-modus:

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

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

Hvis du skriver inn et desimalt heltall som er for stort for en 64-biters binær form med fortegn, brukes en symmetrisk modul-handling for å sette verdien inn i gyldig område.

**Merk:** Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

## angle() vinkel

**angle(Verdi)** ⇒ verdi

Returnerer vinkelen til argumentet, tolker argumentet som et komplekst tall.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

$$\text{angle}(1+i) \quad 0.785398$$

$$\text{angle}(\{1+2\cdot i, 3+0\cdot i, 0-4\cdot i\}) \\ \{1.10715, 0., -1.5708\}$$

**angle(Liste1)** ⇒ liste

**angle(Matrise1)** ⇒ matrise

Returnerer en liste eller vinkelmatrise av elementene i *Liste1* eller *Matrise1*, tolker hvert element som et komplekst tall som representerer et to-dimensjonalt, rektangulært koordinatpunkt.

## ANOVA

**ANOVA** *Liste1*, *Liste2* [, *Liste3*, ..., *Liste20*] [, *Merke*]

Utfører en enveis analyse av varians for å sammenlikne gjennomsnitt for mellom 2 og 20 populasjoner. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

*Merke*=0 for Data, *Merke*=1 for Stats

Utdata-variabel	Beskrivelse
stat.F	Verdi av F-statistikken
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader frihet for gruppene
stat.SS	Sum av kvadrater for gruppene
stat.MS	Gjennomsnitt av kvadrater for gruppene
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
stat.sp	Felles standardavvik
stat.xbarliste	Gjennomsnitt av listenes inndata
stat.CLowerList	95% konfidensintervaller for gjennomsnittet av hver inndata-liste
stat.UpperList	95% konfidensintervaller for gjennomsnittet av hver inndata-liste

## ANOVA2way

**ANOVA2way** *Liste1, Liste2*[,...[*Liste10*]]  
[,*LevRad*]

Beregner en toveis analyse av varians for å sammenlikne gjennomsnitt for mellom 2 og 10 populasjoner. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

*LevRad*=0 for Blokk

*LevRad*=2,3,...,*Len*-1, for To Faktor, hvor  
*Len*=lengde(*Liste1*)=lengde(*Liste2*) = ... =  
lengde(*Liste10*) og *Len* / *LevRad* ∈ {2,3,...}

Utdata: Blokk-oppsett

Utdata-variabel	Beskrivelse
stat.F	F-statistikk over kolonnefaktoren
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader frihet for kolonnefaktoren
stat.SS	Sum av kvadrat for kolonnefaktoren
stat.MS	Gjennomsnitt av kvadrater for kolonnefaktor
stat.FBlock	F-statistikk for faktor
stat.PValBlock	Minste sannsynlighet som null-hypotesen kan forkastes ved
stat.dfBlockstat.dfBlock	Grader frihet for faktor
stat.SSBlock	Sum av kvadrater for faktor
stat.MSBlock	Gjennomsnitt av kvadrater for faktor
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
stat.s	Standardavvik for feilen

#### KOLONNEFAKTOR Utdata

Utdata-variabel	Beskrivelse
stat.Fcol	F-statistikk over kolonnefaktoren
stat.PValCol	Kolonnefaktorens sannsynlighetsverdi
stat.dfCol	Grader frihet for kolonnefaktoren
stat.SSCol	Sum av kvadrater av kolonnefaktoren
stat.MSCol	Gjennomsnitt av kvadrater for kolonnefaktor

#### RADFAKTOR Utdata



Utdata-variabel	Beskrivelse
stat.Frow	F-statistikk over radfaktoren
stat.PValRow	Kolonnefaktorens sannsynlighetsverdi
stat.dfRow	Grader frihet for radfaktoren
stat.SSRow	Sum av kvadrater for radfaktoren
stat.MSRow	Gjennomsnitt av kvadrater for radfaktor




## INTERAKSJON Utdata

Utdata-variabel	Beskrivelse
stat.FInteract	F-statistikk over interaksjonen
stat.PVallInteract	Interaksjonens sannsynlighetsverdi
stat.dfInteract	Grader av frihet for interaksjonen
stat.SSInteract	Sum av kvadrater for interaksjonen
stat.MSInteract	Gjennomsnitt av kvadrater for interaksjon

## FEIL Utdata

Utdata-variabel	Beskrivelse
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
s	Standardavvik for feilen

Ans (svar)	  taster
Ans⇒verdi	56 56
Returnerer resultatet av det sist behandlede uttrykket.	56+4 60
	60+4 64

approx() (tilnærm)	Katalog > 										
<p><b>approx(Verdi1)⇒verdi</b></p> <p>Returnerer behandlingen av argumentet som et uttrykk med desimalverdier, hvis mulig, uavhengig av om modus er <b>Auto</b> eller <b>Tilnærmet</b>.</p> <p>Dette er det samme som å skrive inn argumentet og trykke på  .</p>	<table border="1"> <tbody> <tr> <td><math>\text{approx}\left(\frac{1}{3}\right)</math></td> <td>0.333333</td> </tr> <tr> <td><math>\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)</math></td> <td>{0.333333,0.111111}</td> </tr> <tr> <td><math>\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}</math></td> <td>{0,-1}</td> </tr> <tr> <td><math>\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)</math></td> <td>[1.41421 1.73205]</td> </tr> <tr> <td><math>\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)</math></td> <td>[0.333333 0.111111]</td> </tr> </tbody> </table>	$\text{approx}\left(\frac{1}{3}\right)$	0.333333	$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	{0.333333,0.111111}	$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$	{0,-1}	$\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)$	[1.41421 1.73205]	$\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)$	[0.333333 0.111111]
$\text{approx}\left(\frac{1}{3}\right)$	0.333333										
$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	{0.333333,0.111111}										
$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$	{0,-1}										
$\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)$	[1.41421 1.73205]										
$\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)$	[0.333333 0.111111]										
<b>approx(Liste1)⇒liste</b>	$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$ {0,-1}										
<b>approx(Matrise1)⇒matrise</b>	$\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)$ [1.41421 1.73205]										



## approx() (tilnærm)

Katalog > 

Returnerer en liste eller *matrise* hvor hvert element er blitt behandlet til en desimalverdi, hvis mulig.

## ▶approxFraction()

Katalog > 

*Verdi* ▶`approxFraction([Tol])⇒verdi`

$$\frac{1}{2} + \frac{1}{3} + \tan(\pi) \quad 0.833333$$

*Liste* ▶`approxFraction([Tol])⇒liste`

$$0.8333333333333333 \text{ ▶ } \text{approxFraction}(5 \cdot 10^{-14})$$

*Matrise* ▶`approxFraction([Tol])⇒matrise`

$$\frac{5}{6}$$

Returnerer argumentet som en brøk med en toleranse på *Tol*. Hvis *tol* utelates, brukes en toleranse på 5.E-14.

$$\{\pi, 1.5\} \text{ ▶ } \text{approxFraction}(5 \cdot 10^{-14}) \left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$$

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `@>approxFraction(...)`.

## approxRational()

Katalog > 

`approxRational(Verdi[, Tol])⇒verdi`

$$\text{approxRational}(0.333, 5 \cdot 10^{-5}) \quad \frac{333}{1000}$$

`approxRational(Liste[, Tol])⇒liste`

$$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5 \cdot 10^{-14}) \left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$$

`approxRational(Matrise[, Tol])⇒matrise`

Returnerer argumentet som en brøk med en toleranse på *Tol*. Hvis *Tol* utelates, brukes en toleranse på 5.E-14.

## arccos()

Se  $\cos^{-1}()$ , side 28.

## arccosh()

Se  $\cosh^{-1}()$ , side 29.

## arccot()

Se  $\cot^{-1}()$ , side 30.

**arccoth()** Se  $\text{coth}^{-1}()$ , side 31.

**arccsc()** Se  $\text{csc}^{-1}()$ , side 33.

**arccsch()** Se  $\text{csch}^{-1}()$ , side 34.

**arcsec()** Se  $\text{sec}^{-1}()$ , side 140.

**arcsech()** Se  $\text{sech}^{-1}()$ , side 141.

**arcsin()** Se  $\text{sin}^{-1}()$ , side 149.

**arcsinh()** Se  $\text{sinh}^{-1}()$ , side 150.

**arctan()** Se  $\text{tan}^{-1}()$ , side 161.

**arctanh()** Se  $\text{tanh}^{-1}()$ , side 162.

**() (utvid/sett sammen)**

Katalog > 

**augment(Liste1, Liste2)** ⇒ *liste*

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

Returterer en ny liste som er *Liste2* lagt til på slutten av *Liste1*.

**() (utvid/sett sammen)**Katalog > **augment**(*Matrise1*, *Matrise2*) $\Rightarrow$ *matrise*

Returnerer en ny matrise som er *Matrise2* lagt til på *Matrise1*. Når tegnet “,” brukes, må matrisen ha like raddimensjoner, og *Matrise2* er lagt til på *Matrise1* som nye kolonner. Endrer ikke *Matrise1* eller *Matrise2*.

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

**avgRC() (gjsnEH)**Katalog > **avgRC**(*Uttr1*, *Var* [=Verdi] [, *Trinn*]) $\Rightarrow$ *uttrykk***avgRC**(*Uttr1*, *Var* [=Verdi] [, *Liste1*]) $\Rightarrow$ *liste***avgRC**(*Liste1*, *Var* [=Verdi] [, *Trinn*]) $\Rightarrow$ *liste***avgRC**(*Matrise1*, *Var* [=Verdi] [, *Trinn*]) $\Rightarrow$ *matrise*

Returnerer differenskvotienten tatt i positiv retning (gjennomsnittlig endringshastighet).

*Uttr1* kan være et brukerdefinert funksjonsnavn (se **Func**).

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsstilt verdi eller aktuell “|” erstatning for variabelen.

*Trinn* er trinnverdien. Hvis *Trinn* utelates, brukes grunninnstilling 0,001.

Merk at den liknende funksjonen **centralDiff()** bruker derivasjonskvotienten.

$x:=2$	2
$\text{avgRC}(x^2-x+2,x)$	3.001
$\text{avgRC}(x^2-x+2,x,1)$	3.1
$\text{avgRC}(x^2-x+2,x,3)$	6

**bal()**Katalog > 

**bal**(*NPmt*,*N*,*I*,*PV*,*Pmt*, [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*avrundVerdi*]) $\Rightarrow$ *verdi*

**bal**(*NPmt*,*amortTabell*) $\Rightarrow$ *verdi*

Amortiseringsfunksjon som beregner planlagt balanse etter en spesifisert betaling.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 170.

*NPmt* spesifiserer det betalingsnummeret som du vil at dataene skal beregnes etter.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 170.

- Hvis du utelater *Pmt*, grunninnstilles den til  $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du utelater *FV*, grunninnstilles den til  $FV = 0$ .
- Grunninnstillingene for *PpY*, *CpY* og *PmtAt* er de samme som for TVM-funksjonene.

*avrundVerdi* spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

**bal**(*NPmt*,*amortTabell*) beregner balansen etter betalingsnummer *NPmt*, basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

**Merk:** Se også  $\Sigma\text{Int}()$  og  $\Sigma\text{Prn}()$ , side 198.

<b>bal</b> (5,6,5.75,5000,,12,12)	833.11
-----------------------------------	--------

<i>tbl</i> := <b>amortTbl</b> (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

<b>bal</b> (4, <i>tbl</i> )	1674.27
-----------------------------	---------

*Heltall1* ►Base2⇒*heltall*

256►Base2

0b100000000

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►Base2.

0h1F►Base2

0b111111

Regner om *Heltall1* til et binært tall. Binære eller heksadesimale tall har alltid et prefiks, hhv. 0b eller 0h. Null, ikke bokstaven O, fulgt av b eller h.

*0b binærTall**0h heksadesimalTall*

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks blir *Heltall1* behandlet som et desimalt tall (grunntall 10). Resultatet vises binært, uavhengig av grunntallets modus.

Negative tall vises på "toerkomplement"-form. Eksempel:

-1 vises som

0hFFFFFFFFFFFFFFFF i heksadesimal modus

0b111...111 (64 1-ere) i binær modus

-2<sup>63</sup> vises som

0h8000000000000000 i heksadesimal modus

0b100...000 (63 nuller) i binær modus

Hvis du oppgir et desimalt heltall som ligger utenfor verdiområdet for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. Se følgende eksempler på verdier utenfor verdiområdet.

$2^{63}$  blir  $-2^{63}$  og vises som

0h8000000000000000 i heksadesimal modus

0b100...000 (63 nuller) i binær modus

$2^{64}$  blir 0 og vises som

0h0 i heksadesimal modus

0b0 i binær modus

$-2^{63} - 1$  blir  $2^{63} - 1$  og vises som

0h7FFFFFFFFFFFFFFF i heksadesimal modus

0b111...111 (64 1-ere) i binær modus

## ►Base10 (Grunntall10)

*Heltall1* ►Base10⇒*heltall*

0b10011►Base10	19
0h1F►Base10	31

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►Base10.

Omregner *Heltall1* til et desimaltall (grunntall 10). Binært eller heksadesimalt inndata må alltid ha et prefiks, hhv. 0b eller 0h.

0b binærTall

0h *heksadesimalTall*

Null, ikke bokstaven O, fulgt av b eller h.

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks behandles *Heltall1* som desimaltall. Resultatet vises i desimaltall, uavhengig av grunntall-modus.

*Heltall1 ►Base16⇒heltall*

256►Base16

0h100

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Base16.

0b111100001111►Base16

0hFOF

Omregner *Heltall1* til et heksadesimaltall. Binære eller heksadesimale tall har alltid et prefiks, hhv. 0b eller 0h.

0b binærTall

0h heksadesimalTall

Null, ikke bokstaven O, fulgt av b eller h.

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks blir *Heltall1* behandlet som et desimaltall (grunntall 10). Resultatet vises i heksadesimal, uavhengig av grunntallets modus.

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se ►Base2, side 17.

**binomCdf()***binomCdf(n,p)⇒liste***binomCdf**

*(n,p,nedreGrense,øvreGrense)⇒tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

**binomCdf(n,p,øvreGrense)**for  $P(0 \leq X \leq \text{øvreGrense}) \Rightarrow \text{tall}$  hvis *øvreGrense* er et tall, *liste* hvis *øvreGrense* er en liste

Beregner en kumulativ sannsynlighet for diskret binomisk fordeling med *n* antall forsøk og sannsynlighet *p* for å finne treff ved hvert forsøk.

**binomCdf()**katalog > For  $P(X \leq \text{\textit{\textit{\textit{ovreGrense}}}})$ , sett  $\text{\textit{\textit{\textit{nedreGrense}}}}=0$ **binomPdf()**Katalog > **binomPdf( $n,p$ )** $\Rightarrow$ liste**binomPdf( $n,p,XVerd$ )** $\Rightarrow$ tall hvis  $XVerd$  er et tall, liste hvis  $XVerd$  er en listeBeregner en sannsynlighet ved  $XVerd$  for diskret binomisk fordeling med  $n$  antall forsøk og sannsynlighet  $p$  for å finne treff ved hvert forsøk.**C****ceiling() (øvre)**Katalog > **ceiling( $Verdi$ )** $\Rightarrow$ verdi

$\text{ceiling}(.456)$	1.
------------------------	----

Returnerer det nærmeste heltallet som er  $\geq$  argumentet.

Argumentet kan være et reelt eller et komplekst tall.

**Merk:** Se også **floor()** (nedre).**ceiling( $Liste$ )** $\Rightarrow$ liste

$\text{ceiling}(\{-3.1,1,2.5\})$	$\{-3.,1,3.\}$
----------------------------------	----------------

**ceiling( $Matrice$ )** $\Rightarrow$ matrice

$\text{ceiling}\left(\begin{pmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{pmatrix}\right)$	$\begin{pmatrix} 0 & -3. \cdot i \\ 2. & 4 \end{pmatrix}$
--	---

Returnerer en liste eller matrise med den øvre i hvert element.

**centralDiff()**Katalog > **centralDiff( $Uttr1,Var$  [=Verdi] [,Trinn])** $\Rightarrow$ uttrykk

$\text{centralDiff}(\cos(x),x) _{x=\frac{\pi}{2}}$	-1.
--	-----

**centralDiff( $Uttr1,Var$  [,Trinn])** |  $Var=Verdi$  $\Rightarrow$ uttrykk**centralDiff( $Uttr1,Var$  [=Verdi] [,Liste])** $\Rightarrow$ liste**centralDiff( $Liste1,Var$  [=Verdi] [,Trinn])** $\Rightarrow$ liste



**centralDiff**(*Matrise1*, *Var* [=Verdi]  
[, *Trinn*]) ⇒ *matrise*

Returnerer den numeriske deriverte ved hjelp av derivasjonskvotient-formelen.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsstilt verdi eller aktuell "1" erstatning for variabelen.

*Trinn* er trinnverdien. Hvis *Trinn* utelates, brukes grunninnstilling 0,001.

Hvis du bruker *Liste1* eller *Matrise1*, blir handlingen avbildet gjennom verdiene i listen eller gjennom matriseelementene.

**Merk:** Se også.

## char()

**char**(*Heltall*) ⇒ *tegn*

Returnerer en tegnstring som inneholder det tegnet som er nummerert med *Heltall* fra tegnsettet på grafregneren. Gyldig område for *Heltall* er 0–65535.

char(38)	"&"
char(65)	"A"

## $\chi^2$ 2way

$\chi^2$ 2way *ObsMatrise*

**chi22way** *ObsMatrise*

Beregner en  $\chi^2$  test for samling av "telling" på toveis-tabellen i den observerte matrisen *ObsMatrise*. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

For informasjon om effekten av tomme elementer i en matrise, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat. $\chi^2$	Chi-kvadratstat: $\text{sum}(\text{observert} - \text{forventet})^2 / \text{forventet}$

Utdata-variabel	Beskrivelse
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for chi-kvadratstatistikk
stat.UttrMat	Matrise av forventet element-telletebll ved antatt nullhypotese
stat.KompMat	Matrise av elementbidrag til chi kvadratstatistikk

## $\chi^2$ Cdf()

Katalog > 

$\chi^2$ Cdf(*nedreGrense*,*øvreGrense*,*df*) $\Rightarrow$ tall  
hvis *nedreGrense* og *øvreGrense* er tall,  
*liste* hvis *nedreGrense* og *øvreGrense* er  
lister

chi2Cdf(*nedreGrense*,*øvreGrense*,*df*) $\Rightarrow$ tall  
hvis *nedreGrense* og *øvreGrense* er tall,  
*liste* hvis *nedreGrense* og *øvreGrense* er  
lister

Beregner  $\chi^2$ -fordelingens sannsynlighet mellom *nedreGrense* og *øvreGrense* for det angitte antall frihetsgrader *df*.

For  $P\{X \leq \textit{øvreGrense}\}$ , sett *nedreGrense* = 0.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

## $\chi^2$ GOF

Katalog > 

$\chi^2$ GOF *obsListe*,*uttrListe*,*df*

chi2GOF *obsListe*,*uttrListe*,*df*

Utfører en test for å bekrefte at utvalgsdata er fra en populasjon som er i overensstemmelse med en angitt fordeling. *obsListe* er en liste over antall, og må inneholde heltall. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat. $\chi^2$	Chi-kvadratstat: $\text{sum}((\text{observert} - \text{forventet})^2/\text{forventet})$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for chi-kvadratstatistikk
stat.CompList	Elementbidrag til chi kvadratstatistikk

## $\chi^2$ Pdf()

Katalog > 

$\chi^2\text{Pdf}(X\text{Verd},df) \Rightarrow$  tall hvis  $X\text{Verd}$  er et tall,  
liste hvis  $X\text{Verd}$  er en liste

$\text{chi2Pdf}(X\text{Verd},df) \Rightarrow$  tall hvis  $X\text{Verd}$  er et  
tall, liste hvis  $X\text{Verd}$  er en liste

Beregner sannsynlighetstettheten (pdf) for  $\chi^2$ -fordelingen ved en bestemt  $X\text{Verd}$ -verdi for det angitte antallet frihetsgrader  $df$ .

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

## ClearAZ (slettAZ)

Katalog > 

### ClearAZ

$5 \rightarrow b$	5
-------------------	---

Sletter alle enkelttegn-variabler i det aktuelle oppgaveområdet.

$b$	5
-----	---

ClearAZ	Done
---------	------

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 173.

$b$	"Error: Variable is not defined"
-----	----------------------------------

## ClrErr (SlettFeil)

Katalog > 

### ClrErr

For et eksempel på **ClrErr**, se eksempel 2 under **Try**-kommandoen, side 166.

Tømmer feilstatus og stiller systemvariabelen *feilKode* til null.

**Else** -leddet i **Try...Else...EndTry**-blokken bør bruke **ClrErr** eller **PassErr**. Hvis feilen skal bearbejdes eller ignorerer, bruk **ClrErr**. Hvis det ikke er kjent hva som skal gjøres med feilen, bruk **PassErr** for å sende den til den neste feilbehandleren. Hvis det ikke er flere ventende **Try...Else...EndTry** feilbehandlere, vises feil-dialogboksen som normalt.

**Merk:** Se også **PassErr**, side 116, og **Try**, side 166.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkhåndboken.

## colAugment() (kolUtvid)

**colAugment**(*Matrise1*,  
*Matrise2*) $\Rightarrow$ *matrise*

Returnerer en ny matrise som er *Matrise2* lagt til på *Matrise1*. Matrisene må ha like kolonnedimensjoner, og *Matrise2* er lagt til *Matrise1* som nye rader. Endrer ikke *Matrise1* eller *Matrise2*.

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

## colDim()

**colDim**(*Matrise*) $\Rightarrow$ *uttrykk*

Returnerer antallet kolonner som ligger i *Matrise*.

**Merk:** Se også **radDim()**.

<b>colDim</b> $\left(\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}\right)$	3
---	---

## colNorm()

**colNorm**(*Matrise*) $\Rightarrow$ *uttrykk*

Returnerer den største summene av absoluttverdiene for elementene i kolonnene i *Matrise*.

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$
<b>colNorm</b> ( <i>mat</i> )	9

**colNorm()**

Katalog &gt;

**Merk:** Udefinerte matriseelementer er ikke tillatt. Se også **radNorm()**.

**conj()**

Katalog &gt;

**conj**(*Verdi1*) $\Rightarrow$ *verdi*

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

**conj**(*Liste1*) $\Rightarrow$ *liste*

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

**conj**(*Matrise1*) $\Rightarrow$ *matrise*

Returnerer den komplekse konjugerte av argumentet.

**Merk:** Alle ubestemte variabler behandles som reelle variabler.

**constructMat()**

katalog &gt;

**constructMat**

(  
*Uttr*  
*,Var1,Var2,antRad,antKol*) $\Rightarrow$ *matrise*

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

Returnerer en matrise basert på argumentene.

*Uttr* er et uttrykk i variablene *Var1* og *Var2*. Elementene i resultatmatrisen dannes ved å beregne *Uttr* for hver økte verdi av *Var1* og *Var2*.

*Var1* økes automatisk fra 1 og opp til *antRad*. I hver rad øker *Var2* fra 1 og opp til *antKol*.

**CopyVar (kopiVar)**

katalog &gt;

**CopyVar** *Var1, Var2*

$$\text{Define } a(x) = \frac{1}{x} \qquad \text{Done}$$

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

$$\text{Define } b(x) = x^2 \qquad \text{Done}$$

**CopyVar** *Var1, Var2* kopierer verdien av variabelen *Var1* til variabelen *Var2*, og oppretter *Var2* om nødvendig. Variabel *Var1* må ha en verdi.

$$\text{CopyVar } a, c: c(4) \qquad \frac{1}{4}$$

$$\text{CopyVar } b, c: c(4) \qquad 16$$

Hvis *Var1* er navnet på en eksisterende brukerdefinert funksjon, kopieres definisjonen av denne funksjonen til funksjon *Var2*. Funksjon *Var1* må være definert.

*Var1* må følge reglene for variabelnavn eller være et indirekte uttrykk som kan forenkles til et variabelnavn som oppfyller reglene.

*Var1*. må være navnet på en eksisterende variabelgruppe, for eksempel statistikk *stat.nn*-resultater eller variabler som er opprettet med **LibShortcut()**-funksjonen). Hvis *Var2*. allerede eksisterer, vil denne kommandoen erstatte alle medlemmer som er felles for begge grupper og legge til de medlemmene som ikke allerede eksisterer. Hvis ett eller flere medlemmer av *Var2*. er låst, blir alle medlemmer av *Var2*. værende uendret.

**CopyVar** *Var1*., *Var2*. kopierer alle medlemmene av *Var1*. variabelgruppe til *Var2*. gruppe, og oppretter *Var2*. om nødvendig.

*Var1*. må være navnet på en eksisterende variabelgruppe, for eksempel statistikk *stat.nn*-resultater, eller variabler som er opprettet med **LibShortcut()**-funksjonen. Hvis *Var2*. allerede finnes, vil denne kommandoen erstatte alle medlemmer som er felles for begge grupper, og legge til de medlemmene som ikke allerede finnes. Hvis en enkel (ikke i gruppe) variabel med navnet *Var2* finnes, oppstår det en feil.

<i>aa.a:=45</i>	45																
<i>aa.b:=6.78</i>	6.78																
CopyVar <i>aa</i> ., <i>bb</i> ..	Done																
getVarInfo()	<table border="1"> <tbody> <tr> <td><i>aa.a</i></td> <td>"NUM"</td> <td>"☐"</td> <td>0</td> </tr> <tr> <td><i>aa.b</i></td> <td>"NUM"</td> <td>"☐"</td> <td>0</td> </tr> <tr> <td><i>bb.a</i></td> <td>"NUM"</td> <td>"☐"</td> <td>0</td> </tr> <tr> <td><i>bb.b</i></td> <td>"NUM"</td> <td>"☐"</td> <td>0</td> </tr> </tbody> </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**(*Liste1*,*Liste2*[,...[,*Liste20*]])

Beregner korrelasjonsmatrisen for den utvidede matrisen [ *Liste1*, *Liste2*, . . . , *Liste20* ].

**cos**(Verdi1)⇒verdi

**cos**(Liste1)⇒liste

**cos**(Verdi1) returnerer cosinus til argumentet som en verdi.

**cos**(Liste1) returnerer en liste av cosinus til alle elementer i *Liste1*.

**Merk:** Argumentet tolkes som grader, gradian eller radian av en vinkel, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °, G eller r for å hoppe over vinkelmodusen midlertidig.

**cos**(kvadratMatrise1)⇒kvadratMatrise

Returnerer matrisens cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne cosinus til hvert element.

Når en skalarfunksjon  $f(A)$  virker på *kvadratMatrise1* (A), beregnes resultatet av algoritmen:

Beregner egenverdiene ( $\lambda_i$ ) og egenvektorene ( $V_i$ ) av A.

*kvadratMatrise1* må kunne diagonaliseres. Den kan heller ikke ha symbolske variabler som ikke er tildelt noen verdi.

Utform matrisene:

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

I Grader-vinkelmodus:

$$\cos\left(\left(\frac{\pi}{4}\right)^r\right) \quad 0.707107$$

$$\cos(45) \quad 0.707107$$

$$\cos(\{0,60,90\}) \quad \{1,0.5,0\}$$

I Gradian-vinkelmodus:

$$\cos(\{0,50,100\}) \quad \{1,0.707107,0\}$$

I Radian-vinkelmodus:

$$\cos\left(\frac{\pi}{4}\right) \quad 0.707107$$

$$\cos(45^\circ) \quad 0.707107$$

I Radian-vinkelmodus:

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

## cos()



Da er  $A = X B X^{-1}$  og  $f(A) = X f(B) X^{-1}$ . For eksempel,  $\cos(A) = X \cos(B) X^{-1}$  hvor:

$\cos(B) =$

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

Alle beregningene utføres med flytende desimalpunkt-aritmetikk.

## cos<sup>-1</sup>()



$\cos^{-1}(\text{Verdi}) \Rightarrow \text{verdi}$

I Grader-vinkelmodus:

$\cos^{-1}(\text{ListeI}) \Rightarrow \text{liste}$

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

$\cos^{-1}(\text{VerdiI})$  returnerer vinkelen som har cosinus lik *VerdiI*.

I Gradian-vinkelmodus:

$\cos^{-1}(\text{ListeI})$  returnerer en liste over invers cosinus for hvert element i *ListeI*.

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

**Merk:** Resultatet returneres som en vinkel i enten grader, gradian eller radian, avhengig av aktuell vinkelmodus-innstilling.

I Radian-vinkelmodus:

$\cos^{-1}(\{0,0,2,0,5\})$   
 $\{1.5708, 1.36944, 1.0472\}$

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccos (...)**.

$\cos^{-1}(\text{kvadratMatriseI}) \Rightarrow \text{kvadratMatrise}$

I radian-vinkelmodus og rektangulært, kompleks format:

Returnerer matrisens inverse cosinus til *kvadratMatriseI*. Dette er ikke det samme som å beregne invers cosinus til hvert element. For mer informasjon om beregningmetode, se under **cos()**.

$$\cos^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.778369 \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \cdot i \end{bmatrix}$$

*kvadratMatriseI* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

For å se hele resultatet, trykk på  $\blacktriangle$  og bruk så  $\blacktriangleleft$  og  $\blacktriangleright$  for å bevege markøren.



**cosh**(VerdiI)⇒verdi

I Grader-vinkelmodus:

**cosh**(ListeI)⇒liste

$$\cosh\left(\left(\frac{\pi}{4}\right)r\right) \quad 1.74671\text{E}19$$

**cosh**(VerdiI) returnerer hyperbolsk cosinus til argumentet.**cosh**(ListeI) returnerer en liste over hyperbolsk cosinus til hvert element i ListeI.**cosh**  
(kvadratMatriseI)⇒kvadratMatrise

I Radian-vinkelmodus:

Returnerer matrisens hyperbolske cosinus til kvadratMatriseI. Dette er ikke det samme som å beregne hyperbolsk cosinus til hvert element. For mer informasjon om beregningsmetode, se under **cos**().

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

*kvadratMatriseI* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.**cosh**<sup>-1</sup>(VerdiI)⇒verdi

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

$$\cosh^{-1}(\{1,2,1,3\}) \quad \{0,1.37286,1.76275\}$$

**cosh**<sup>-1</sup>(ListeI)⇒liste**cosh**<sup>-1</sup>(VerdiI) returnerer invers hyperbolsk cosinus til argumentet.**cosh**<sup>-1</sup>(ListeI) returnerer en liste over invers hyperbolsk cosinus til hvert element i ListeI.**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccosh** (...).**cosh**<sup>-1</sup>  
(kvadratMatriseI)⇒kvadratMatrise

I radian-vinkelmodus og rektangulært, kompleks format:

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

Katalog &gt;

Returnerer matrisens inverse hyperbolsk cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers hyperbolsk cosinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\cosh^{-1}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 2.52503+1.73485\cdot i & -0.009241-1.49086\cdot i \\ 0.486969-0.725533\cdot i & 1.66262+0.623491\cdot i \\ -0.322354-2.08316\cdot i & 1.26707+1.79018\cdot i \end{bmatrix}$$

For å se hele resultatet, trykk på og bruk så og for å bevege markøren.

## cot()

**tast**

**cot(Verdi1)** ⇒ verdi

I Grader-vinkelmodus:

**cot(Liste1)** ⇒ liste

$$\text{cot}(45) \quad 1.$$

Returnerer cotangens av *Verdi1* eller returnerer en liste med cotangens til alle elementene i *Liste1*.

I Gradian-vinkelmodus:

**Merk:** Argumentet tolkes som grader, gradianer eller radianer av en vinkel, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °, G eller r for å hoppe over vinkelmodusen midlertidig.

$$\text{cot}(50) \quad 1.$$

I Radian-vinkelmodus:

$$\text{cot}(\{1,2,1,3\}) \\ \{0.642093, -0.584848, -7.01525\}$$

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

**tast**

**cot<sup>-1</sup>(Verdi1)** ⇒ verdi

I Grader-vinkelmodus:

**cot<sup>-1</sup>(Liste1)** ⇒ liste

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

Returnerer vinkelen som har cotangens lik *Verdi1* eller returnerer en liste som inneholder invers cotangens til hvert element i *Liste1*.

I Gradian-vinkelmodus:

**Merk:** Resultatet returneres som en vinkel i enten grader, gradian eller radian, avhengig av aktuell vinkelmodus-innstilling.

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

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccot (...)**.

I Radian-vinkelmodus:

$$\text{cot}^{-1}(1) \quad 0.785398$$

**coth()**Katalog > **coth**(Verdi) ⇒ verdi

coth(1.2) 1.19954

**coth**(Liste1) ⇒ liste

coth({1,3,2}) {1.31304,1.00333}

Returnerer hyperbolsk cotangens til *uttrykk1*, eller returnerer en liste med hyperbolsk cotangens til alle elementene i *liste1*.

**coth<sup>-1</sup>()**Katalog > **coth<sup>-1</sup>**(Verdi) ⇒ verdicoth<sup>-1</sup>(3.5) 0.293893**coth<sup>-1</sup>**(Liste1) ⇒ listecoth<sup>-1</sup>({-2,2,1,6})  
{-0.549306,0.518046,0.168236}

Returnerer invers hyperbolsk cotangens til *Verdi1* eller returnerer en liste som inneholder invers hyperbolsk cotangens til hvert element i *Liste1*.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccoth (...)**.

**count() (antall)**Katalog > **count**(Verdi1 eller Liste1 [,Verdi2 eller Liste2 [...]]) ⇒ verdi

count(2,4,6) 3

count({2,4,6}) 3

Returnerer samlet antall av alle elementer i argumentene som behandles til numeriske verdier.

count(2,{4,6},

8	10
12	14

) 7

Hvert argument kan være et uttrykk, en verdi, liste eller matrise. Du kan blande datatyper og bruke argumenter med forskjellige dimensjoner.

For en liste, matrise eller et celleområde blir hver element behandlet for å bestemme om det bør inkluderes i antallet.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor et argument.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

**countIf() (tellIf)**
**countIf(Liste, Kriterium) ⇒ verdi**

Returnerer samlet antall av alle argumenter i *Liste* som møter de spesifiserte *kriterier*.

*Kriterium* kan være:

- En verdi, et uttrykk eller en streng. For eksempel, **3** teller kun de elementene i *Liste* som forenkles til verdien 3.
- Et boolsk uttrykk som inneholder symbolet **?** som plassholder for hvert element. For eksempel, **? < 5** teller kun de elementene i *Liste* som er mindre enn 5.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor *Liste*.

Tomme (åpne) elementer i listen ignoreres. For mer informasjon om tomme elementer, se side 223.

**Merk:** Se også **sumIf()**, side 159, og **frequency()**, side 59.

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

Teller alle elementer som er lik 3.

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

Teller alle elementer som er lik "def."

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

Teller 1 og 3.

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

Teller 3, 5 og 7.

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

Teller 1, 3, 7 og 9.

**cPolyRoots()**
**cPolyRoots(Poly, Var) ⇒ liste**
**cPolyRoots(KoeffListe) ⇒ liste**

Den første syntaksen, **cPolyRoots(Poly, Var)**, returnerer en liste over komplekse røtter av polynom *Poly* med hensyn på variabel *Var*.

*Poly* må være et polynom i utvidet form i én variabel. Ikke bruk utvidede former, som f.eks.  $y^2 \cdot y + 1$  or  $x \cdot x + 2 \cdot x + 1$

Den andre syntaksen, **cPolyRoots(KoeffListe)**, returnerer en liste over komplekse røtter for koeffisienter i *KoeffListe*.

**Merk:** Se også **polyRoots()**, side 118.

<code>polyRoots(y^3+1,y)</code>	{-1}
---------------------------------	------

<code>cPolyRoots(y^3+1,y)</code>	{-1, 0.5-0.8660254i, 0.5+0.8660254i}
----------------------------------	--------------------------------------

<code>polyRoots(x^2+2*x+1,x)</code>	{-1, -1}
-------------------------------------	----------

<code>cPolyRoots({1,2,1})</code>	{-1, -1}
----------------------------------	----------

**crossP() (kryssprodukt)**Katalog > **crossP(Liste1, Liste2) ⇒ liste**Returnerer kryssproduktet av *Liste1* og *Liste2* som en liste.*Liste1* og *Liste2* må ha lik dimensjon, og dimensjonen må være enten 2 eller 3.**crossP(Vektor1, Vektor2) ⇒ vektor**Returnerer en rad- eller kolonnevektor (avhengig av argumentene) som er kryssproduktet av *Vektor1* og *Vektor2*.Både *Vektor1* og *Vektor2* må være radvektorer, eller begge må være kolonnevektorer. Begge vektorene må ha lik dimensjon, og dimensjonen må være enten 2 eller 3.

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


---


$$\{-2.5, -5., -2.25\}$$

$$\text{crossP}(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \begin{bmatrix} -3 & 6 & -3 \end{bmatrix})$$


---


$$\text{crossP}(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -2 \end{bmatrix})$$

**csc()** **tast****csc(Verdi1) ⇒ verdi**

I Grader-vinkelmodus:

**csc(Liste1) ⇒ liste**

$$\text{csc}(45) \quad 1.41421$$

Returnerer cosekans til *Verdi1* eller returnerer en liste som inneholder cosekans til alle elementene i *Liste1*.

I Gradian-vinkelmodus:

$$\text{csc}(50) \quad 1.41421$$

I Radian-vinkelmodus:

$$\text{csc}\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) \quad \{1.1884, 1., 1.1547\}$$

**csc<sup>-1</sup>()** **tast****csc<sup>-1</sup>(Verdi1) ⇒ verdi**

I Grader-vinkelmodus:

**csc<sup>-1</sup>(Liste1) ⇒ liste**

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

Returnerer vinkelen som har cosekans lik *Verdi1* eller returnerer en liste som inneholder invers cosekans til hvert element i *Liste1*.

I Gradian-vinkelmodus:

## $\text{csc}^{-1}()$

trig 

**Merk:** Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

 $\text{csc}^{-1}(1)$ 

100.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `arccsc (...)`.

I Radian-vinkelmodus:

 $\text{csc}^{-1}(\{1,4,6\}) \quad \{1.5708,0.25268,0.167448\}$ 

## $\text{csch}()$

Katalog >  $\text{csch}(\textit{Verdi}) \Rightarrow \textit{verdi}$  $\text{csch}(3)$ 

0.099822

 $\text{csch}(\textit{Liste1}) \Rightarrow \textit{liste}$  $\text{csch}(\{1,2,1,4\})$  $\{0.850918,0.248641,0.036644\}$ 

Returnerer hyperbolsk cosekans til *Verdi1* eller returnerer en liste med hyperbolsk cosekans til alle elementene i *Liste1*.

## $\text{csch}^{-1}()$

Katalog >  $\text{csch}^{-1}(\textit{Verdi}) \Rightarrow \textit{verdi}$  $\text{csch}^{-1}(1)$ 

0.881374

 $\text{csch}^{-1}(\textit{Liste1}) \Rightarrow \textit{liste}$  $\text{csch}^{-1}(\{1,2,1,3\})$  $\{0.881374,0.459815,0.32745\}$ 

Returnerer invers hyperbolsk cosekans til *Verdi1* eller returnerer en liste som inneholder invers hyperbolsk cosekans til hvert element i *Liste1*.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `arccsch (...)`.

## CubicReg

Katalog > 

**CubicReg** *X*, *Y*, [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den kubiske polynomiske regresjonen  $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$  for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall  $\geq 0$ .

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.R <sup>2</sup>	Koeffisientbestemmelse
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> , og <i>Inkludert kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>inkludert kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## cumulativeSum()

**cumulativeSum(Liste1) ⇒ liste**

$\text{cumulativeSum}(\{1,2,3,4\}) \quad \{1,3,6,10\}$

Returnerer en liste over de kumulative summene av elementene i *Liste1*, og starter med element 1.

## cumulativeSum()

Katalog > 

### cumulativeSum(Matrise1)⇒matrise

Returnerer en matrise av de kumulative summene av elementene i *Matrise1*. Hvert element er den kumulative summen av kolonnen fra topp til bunn.

Et tomt (åpent) element i *Liste1* eller *Matrise1* produserer et åpent element i den resulterende listen eller matrisen. For mer informasjon om tomme elementer, se side 223.

1 2	→ m1	1 2
3 4		3 4
5 6		5 6
cumulativeSum(m1)		1 2
		4 6
		9 12

## Cycle (Løkke)

Katalog > 

### Cycle (Løkke)

Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (**For**, **While**, eller **Loop**).

**Cycle** er ikke tillatt utenfor de tre løkkestrukturene (**For**, **While**, eller **Loop**).

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

Funksjonsliste som summerer heltallene fra 1 til 100 og hopper over 50.

```
Define g()=Func                                     Done
    Local temp,i
    0→temp
    For i,1,100,1
    If i=50
    Cycle
    temp+i→temp
    EndFor
    Return temp
    EndFunc
```

g() 5000

## ►Cylind

Katalog > 

### Vektor►Cylind

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>**Cylind**.

Viser rad- eller kolonnevektor i sylindrisk form [r, ∠θ, z].

*Vektor* må ha nøyaktig tre elementer. Det kan være enten en rad eller en kolonne.

```
[2 2 3]►Cylind
[2.82843 ∠0.785398 3.]
```



## D

### dbd()

Katalog > 

**dbd**(dato1,dato2)⇒verdi

Returnerer antallet dager mellom *dato1* og *dato2* ved hjelp av aktuelt-antall-dager-metoden.

*dato1* og *dato2* kan være tall eller lister av tall innenfor datoområdet på en vanlig kalender. Hvis både *dato1* og *dato2* er lister, må de være like lange.

*dato1* og *dato2* må ligge mellom årene 1950 og 2049.

Du kan legge inn datoene i ett av to formater. Hvor du setter desimalkommaet bestemmer hvilket datoformat du bruker.

MM.DDÅÅ (format som vanligvis brukes i USA)

DDMM.ÅÅ (format som vanligvis brukes 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

### ►DD

Katalog > 

Verdi ►DD⇒verdi

Liste1 ►DD⇒liste

Matrise1 ►DD⇒matrise

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►DD.

Returnerer desimalekvivalenten til argumentet uttrykt i grader. Argumentet er et tall, en liste eller matrise som tolkes av vinkelmodus-innstillingen i gradianer, radianer eller grader.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

1►DD	$\frac{9}{10}$ °
------	------------------

I Radian-vinkelmodus:

(1.5)►DD	85.9437°
----------	----------

**Verdi1** ►Decimal⇒verdi

$\frac{1}{3}$  ►Decimal

0.333333

Listel ►Decimal⇒verdi

Matrise1 ►Decimal⇒verdi

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Decimal.

Viser argumentet i desimalform. Denne operatoren kan kun brukes på slutten av kommandolinjen.

## Define (Definer)

**Define** *Var* = *Uttrykk*

**Define** *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

Definerer variabelen *Var* eller den egendefinerte funksjonen *Funksjon*.

Parametere, som f.eks. *Param1*, er plassholdere for å sette argumenter til funksjonen. Når du kaller opp en egendefinert funksjon, må du legge til argumenter (for eksempel verdier eller variabler) som samsvarer med parametere. Når funksjonen er kalt opp, behandler den *Uttrykk* ved hjelp av de argumentene som er lagt til.

*Var* og *Funksjon* kan ikke være navnet på systemvariabel eller innebygget funksjon eller kommando.

**Merk:** Denne type **Define** er ekvivalent til å utføre uttrykket: *uttrykk* → *Funksjon* (*Param1*,*Param2*).

**Define** *Funksjon*(*Param1*, *Param2*, ...) = **Funk**  
*Blokk*  
**EndFunk**

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

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

Define $g(x,y)=\text{Func}$	Done
If $x>y$ Then	
Return $x$	
Else	
Return $y$	
EndIf	
EndFunc	
$g(3,-7)$	3

I denne formen kan egendefinert funksjon eller program utføre en blokk med flere utsagn.

*Blokk* kan enten være et enkelt utsagn eller en rekke med utsagn på separate linjer. *Blokk* kan også inkludere uttrykk og instruksjoner (som **If**, **Then**, **Else** og **For**).

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

**Merk:** Se også **Define BiblPriv**, side 39 og **Define BiblOff**, side 39.

---

```
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 (Definer BiblPriv)

**Define LibPriv** *Var* = *Uttrykk*

**Define LibPriv** *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

**Define LibPriv** *Funksjon*(*Param1*, *Param2*, ...) = **Funk**  
*Blokk*  
**EndFunk**

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

Opererer på samme måte som **Define**, men definerer en privat biblioteksvariabel, -funksjon eller et -program. Private funksjoner og programmer forekommer ikke i Katalogen.

**Merk:** Se også **Define**, side 38 og **Define LibPub**, side 39.

## Define LibPub (Definer BiblOff)

**Define LibPub** *Var* = *Uttrykk*

**Define LibPub** *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

**Define LibPub** *Funksjon*(*Param1*, *Param2*, ...) = **Funk**  
*Blokk*  
**EndFunk**

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

Opererer på samme måte som **Define**, men definerer en felles (offentlig) biblioteksvariabel, -funksjon eller et -program. Felles (offentlige) funksjoner og programmer forekommer i Katalogen etter at biblioteket er blitt lagret og oppdatert.

**Merk:** Se også **Define**, side 38 og **Define LibPriv**, side 39.

**deltaList()**Se  $\Delta$ List(), side 87.**DelVar**katalog > 

**DelVar** *Var1*[, *Var2*] [, *Var3*] ...

$2 \rightarrow a$	2
-------------------	---

**DelVar** *Var*.

$(a+2)^2$	16
-----------	----

Sletter den angitte variabelen eller variabelgruppen fra minnet.

DelVar <i>a</i>	<i>Done</i>
-----------------	-------------

$(a+2)^2$	"Error: Variable is not defined"
-----------	----------------------------------

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 173.

## DelVar

katalog > 

**DelVar** *Var*. sletter alle medlemmer av *Var*. variabelgruppen (for eksempel statistikk *stat.nn*-resultater eller variabler som er opprettet med **LibShortcut()**-funksjonen). Prikken (.) i denne formen av **DelVar**-kommandoen begrenser den til å slette en variabelgruppe. Enkeltvariabelen *Var* påvirkes ikke.

<i>aa.a</i> :=45	45									
<i>aa.b</i> :=5.67	5.67									
<i>aa.c</i> :=78.9	78.9									
getVarInfo()	<table border="1"><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"[ ]"</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"[ ]"</td></tr><tr><td><i>aa.c</i></td><td>"NUM"</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> .	Done									
getVarInfo()	"NONE"									

## delVoid()

Katalog > 

**delVoid**(*Liste1*)⇒*liste*

delVoid({1,void,3})	{1,3}
---------------------	-------

Returnerer en liste som har innholdet til *Liste1*, der alle tomme (åpne) elementer er fjernet.

For mer informasjon om tomme elementer, se side 223.

## det()

Katalog > 

**det**(*kvadratMatrise*[,  
*Toleranse*])⇒*uttrykk*

$\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	-2
--	----



Returnerer determinanten til *kvadratMatrise*.

$\begin{bmatrix} 1.€20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow mat1$	$\begin{bmatrix} 1.€20 & 1 \\ 0 & 1 \end{bmatrix}$
---	--

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Toleranse*. Denne toleransen brukes bare hvis matrisen har elementer med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Toleranse*.

det( <i>mat1</i> )	0
--------------------	---

det( <i>mat1</i> ,.1)	1.€20
-----------------------	-------

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flyttallsaritmetikk.
- Hvis *Toleranse* utelates eller ikke blir brukt, beregnes standardtoleransen som:

5E-14 · maks(dim(kvadratMatrise)) ·  
radNorm(kvadratMatrise)

## diag()

diag(Liste) ⇒ matrise

diag([2 4 6])	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
---------------	---

diag(radMatrise) ⇒ matrise

diag(kolonneMatrise) ⇒ matrise

Returnerer en matrise med verdiene i argumentlisten eller matrise i hoveddiagonalen.

diag(kvadratMatrise) ⇒ radMatrise

Returnerer en radmatrise som inneholder elementene fra hoveddiagonalen til kvadratMatrise.

$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$	$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$
diag(Ans)	$\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$

kvadratMatrise må være kvadrat.

## dim()

dim(Liste) ⇒ heltall

dim({0,1,2})	3
--------------	---

Returnerer dimensjonen av Liste.

dim(Matrise) ⇒ liste

Returnerer matrisens dimensjoner som en to-elements liste {rader, kolonner}.

dim( $\begin{bmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{bmatrix}$ )	{3,2}
--	-------

dim(Streng) ⇒ heltall

Returnerer antallet tegn som er inneholdt i tegnstrengen Streng.

dim("Hello")	5
dim("Hello "&"there")	11

**Disp** *utrElStreng1* [, *utrElStreng2*] ...

Viser argumentene i *Calculator*-loggen. Argumentene vises suksessivt, med korte avstander som skille.

Hovedsakelig nyttig i programmer og funksjoner for å sikre visning av mellomregninger.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

```
Define chars(start,end)=Prgm
  For i,start,end
  Disp i," ",char(i)
  EndFor
EndPrgm
```

---

*Done*

---

```
chars(240,243)
```

---

240 ø

241 ñ

242 ò

243 ó

---

*Done*

---

## DispAt

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

**DispAt** lar deg angi linjen der det angitte uttrykket eller den angitte strengen skal vises på skjermen.

Linjenummeret kan angis som et uttrykk.

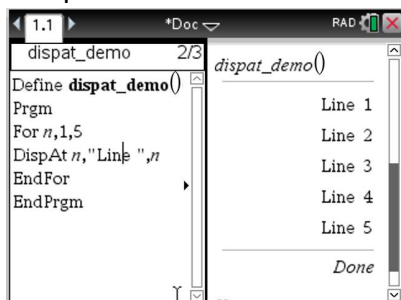
Merk: Linjenummeret gjelder ikke hele skjerm bildet, men kun området som følger umiddelbart etter kommandoen/programmet.

Denne kommandoen lar deg opprette en instrumentbordlignende visning av data fra programmer der verdien til et uttrykk eller en sensoravlesning oppdateres på samme linje.

**DispAt** og **Disp** kan brukes i samme program.

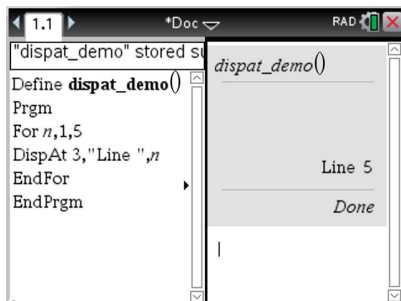
DispAt

### Eksempel



```
1.1 | *Doc | RAD |
dispat_demo 2/3
Define dispat_demo()
Prgm
For n,1,5
DispAt n,"Line ",n
EndFor
EndPrgm
```

dispat\_demo()
Line 1
Line 2
Line 3
Line 4
Line 5
Done



```
1.1 | *Doc | RAD |
"dispat_demo" stored st
Define dispat_demo()
Prgm
For n,1,5
DispAt 3,"Line ",n
EndFor
EndPrgm
```

dispat\_demo()
Line 5
Done

Illustrerende eksempler:

**Merk:** Maksimumsantall er definert som 8 ettersom dette tilsvarer et fullt display av linjer på grafregnerens skjermbilde, såfremt linjene ikke inneholder matematiske uttrykk i 2D. Det nøyaktige antallet linjer avhenger av innholdet til de viste dataene.

<pre>Define z(= Prgm For n,1,3 DispAt 1,"N: ",n Disp "Hallo" EndFor EndPrgm</pre>	<p>Utdata</p> <p>z()</p> <p>Iterasjon 1:</p> <p>Linje 1: N:1 Linje 2: Hallo</p> <p>Iterasjon 2:</p> <p>Linje 1: N:2 Linje 2: Hallo Linje 3: Hallo</p> <p>Iterasjon 3:</p> <p>Linje 1: N:3 Linje 2: Hallo Linje 3: Hallo Linje 4: Hallo</p>
<pre>Define z1(= Prgm For n,1,3 DispAt 1,"N: ",n EndFor  For n,1,4 Disp "Hallo" EndFor EndPrgm</pre>	<p>z1()</p> <p>Linje 1: N:3 Linje 2: Hallo Linje 3: Hallo Linje 4: Hallo Linje 5: Hallo</p>

#### Feilmeldinger:

Feilmelding	Beskrivelse
DispAt-linjenummeret må være mellom 1 og 8	Uttrykk evaluerer linjenummeret utenfor området 1–8 (til og med)
For få argumenter	Funksjonen eller kommandoen mangler et eller flere argumenter.
Ingen argumenter	Det samme som gjeldende Syntaksfeil-dialogboks
For mange argumenter	Begrens argument. Samme feil som Disp.
Ugyldig datatype	Det første argumentet må være et tall.
Åpen: DispAt åpen	Datatypefeilen "Hei alle sammen"



<b>Feilmelding</b>	<b>Beskrivelse</b> iverksettes for den tomme verdien (hvis oppkall er definert)
--------------------	--

## ►DMS (GMS)

Katalog > 

Verdi ►DMS

I Grader-vinkelmodus:

Liste ►DMS

$\{45.371\}$ ►DMS	45°22'15.6"
-------------------	-------------

Matrise ►DMS

$\{\{45.371,60\}\}$ ►DMS	$\{45^\circ 22' 15.6'', 60^\circ\}$
--------------------------	-------------------------------------

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive  $e \rightarrow \text{DMS}$ .

Tolker argumentet som en vinkel og viser ekvivalenten DMS (GGGGG°MM 'SS.ss")-tallet. Se °, ', " (side 202) for DMS-format (grader, minutter, sekunder).

**Merk:** ►DMS vil omregne fra radianer til grader når det brukes i radian-modus. Hvis inndata blir fulgt av et grader-symbol °, finner det ikke sted noe omregning. Du kan bare bruke ►DMS på slutten av en kommandolinje.

## dotP() (prikkP)

Katalog > 

**dotP(Liste1, Liste2) ⇒ uttrykk**

$\text{dotP}(\{1,2\}, \{5,6\})$	17
---------------------------------	----

Returnerer "prikk"produktet av to lister.

**dotP(Vektor1, Vektor2) ⇒ uttrykk**

$\text{dotP}([1 \ 2 \ 3], [4 \ 5 \ 6])$	32
---	----

Returnerer "prikk"produktet av to vektorer.

Begge må være radvektorer, eller begge må være kolonnevektorer.

**e<sup>^</sup>()****e<sup>x</sup>** tast**e<sup>^</sup>(Verdi)** ⇒ verdiReturnerer e opphøyd i *Verdi*-potens.**Merk:** Se også **e eksponent-sjablon**, side 2.**Merk:** Å trykke på **e<sup>x</sup>** for å vise e<sup>^</sup>(er forskjellig fra å trykke på tegnet **E** på tastaturet.

Du kan legge inn et komplekst tall i rei θ polar form. Men bruk denne formen bare i radian-vinkelmodus; den forårsaker grunnmengdefeil i grader- eller gradian-vinkelmodus.

**e<sup>^</sup>(Liste)** ⇒ listeReturnerer tallet e opphøyd i potens av hvert element i *Liste*.**e<sup>^</sup>(kvadratMatrise)** ⇒ kvadratMatriseReturnerer kvadratMatrise som er e opphøyd i *kvadratMatrise*. Dette er ikke det samme som å beregne e opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.*kvadratMatrise* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

e <sup>1</sup>	2.71828
e <sup>3<sup>2</sup></sup>	8103.08

e <sup>{1,1,.0.5}</sup>	{2.71828,2.71828,1.64872}
-------------------------	---------------------------

e <sup>1 5 3</sup>	782.209	559.617	456.509
e <sup>4 2 1</sup>	680.546	488.795	396.521
e <sup>6 -2 1</sup>	524.929	371.222	307.879

**eff()****Katalog** > **eff(nominellRente, CpY)** ⇒ verdiFinansiell funksjon som omregner den nominelle renten *nominellRente* til en årlig effektiv rente, gitt *CpY* som antall renteperioder per år.*nominellRente* må være et reelt tall, og *CpY* må være et reelt tall > 0.**Merk:** Se også **nom()**, side 108.

eff(5.75,12)	5.90398
--------------	---------

## eigVc() (egenvektor)

Katalog > 

**eigVc(kvadratMatrise)**⇒matrise

Returnerer en matrise som inneholder egenvektorer for en reell eller kompleks *kvadratMatrise*, der hver kolonne i resultatet samsvarer med en egenverdi. Merk at en egenvektor ikke er entydig; den kan skaleres av enhver konstant faktor. Egenvektorene er normalisert, dvs. at if  $V = [x_1, x_2, \dots, x_n]$ , then:

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

*kvadratMatrise* blir først balansert med likhetstransformasjoner til normene for rad og kolonne er så nær den samme verdien som mulig. *KvadratMatrisen* blir så redusert til øvre Hessenberg-form og egenvektorene beregnes via en Schur-faktorisering.

I rektangulært, kompleks format:

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

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

## eigVl() (egenverdi)

Katalog > 

**eigVl(kvadratMatrise)**⇒liste

Returnerer en liste over egenverdiene av en reell eller kompleks *kvadratMatrise*.

*kvadratMatrise* blir først balansert med likhetstransformasjoner til normene for rad og kolonne er så nær den samme verdien som mulig. *KvadratMatrisen* blir så redusert til øvre Hessenberg-form og egenverdiene beregnes fra den øvre Hessenberg-matrisen.

I rektangulær, kompleks format-modus:

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

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

**Else**

**Se if, side 72.**

**IfBoolskUttr1 Then***Blokk1***Elseif BoolskUttr2 Then***Blokk2*

⋮

**Elseif Boolsk UttrN Then***BlokkN***Endif**

⋮

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Define  $g(x)$ =FuncIf  $x \leq -5$  Then

Return 5

ElseIf  $x > -5$  and  $x < 0$  ThenReturn  $-x$ ElseIf  $x \geq 0$  and  $x \neq 10$  ThenReturn  $x$ ElseIf  $x = 10$  Then

Return 3

EndIf

EndFunc

*Done***EndFor**

Se For, side 56.

**EndFunc**

Se Func, side 60.

**Endif**

Se If, side 72.

**EndLoop**

Se Loop, side 93.

**EndPrgm**

Se Prgm, side 120.

**EndTry**

Se Try, side 166.

## euler ()

Katalog > 

**euler**(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}, *avhVar0*, *VarTall* [, *eulersIntervall*]) ⇒ *matrise*

**euler**(*SystemAvUttr*, *Var*, *ListeMedAvhVarer*, {*Var0*, *VarMaks*}, *ListeMedAvhVarer0*, *VarIntervall* [, *eulersIntervall*]) ⇒ *matrise*

**euler**(*ListeMedUttr*, *Var*, *ListeMedAvhVarer*, {*Var0*, *VarMaks*}, *ListeMedAvhVarer0*, *VarIntervall* [, *eulersIntervall*]) ⇒ *matrise*

Bruker Eulers metode for å løse systemet

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

med *avhVar*(*Var0*)=*avhVar0* på intervallet [*Var0*,*VarMaks*]. Returnerer en matrise, hvor den første raden definerer verdiene i *Var* -resultatet og hvor den andre raden definerer verdien av den første løsningskomponenten ved de tilsvarende *Var* -verdiene, og så videre.

*Uttr* er høyre side, som definerer den ordinære differensialligningen (ODE).

*SystemAvUttr* er systemet på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i *ListeMedAvhVarer*).

*ListeMedUttr* er en liste på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i *ListeMedAvhVarer*).

*Var* er den uavhengige variabelen.

*ListeMedAvhVarer* er en liste over avhengige variabler.

Differensialligning:

$$y' = 0,001 \cdot y \cdot (100 - y) \text{ og } y(0) = 10$$

---


$$\text{euler}(0,001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1)$$

0.	1.	2.	3.	4.
10.	10.9	11.8712	12.9174	14.042

---

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

System av ligninger:

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

med  $y1(0) = 2$  og  $y2(0) = 5$

---


$$\text{euler}\left(\begin{cases} -y1+0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

0.	1.	2.	3.	4.	5.
2.	1.	1.	3.	27.	243.
5.	10.	30.	90.	90.	-2070.

---

$\{Var0, VarMaks\}$  er en liste med to elementer som forteller funksjonen at den skal integrere fra  $Var0$  til  $VarMaks$ .

$ListeMedAvhVarer$  er en liste over startverdier for avhengige variabler.

$VarIntervall$  er et tall som ikke er null, slik at  $sign(VarIntervall) = sign(VarMaks-Var0)$  og løsninger returneres ved  $Var0+i \cdot VarIntervall$  for alle  $i=0,1,2,\dots$  slik at  $Var0+i \cdot VarIntervall$  er i  $[var0, VarMaks]$  (det kan hende at det ikke er noen løsningsverdi ved  $VarMaks$ ).

$eulersIntervall$  er et positivt heltall (grunninnstilt på 1) som definerer antallet euler-intervaller mellom resultatverdiene. Den faktiske tallstørrelsen som brukes ved eulers metode, er  $VarIntervall/eulersIntervall$ .

## eval ()

## Hub-meny

$eval(Uttr) \Rightarrow streng$

$eval()$  er bare gyldig i TI-Innovator™ Hub kommandoargumentet til programmeringskommandoer **Get**, **GetStr** og **Send**. Programvaren vurderer uttrykk  $Uttr$  og erstatter formuleringen  $eval()$  med resultatet som en tegnstreng.

Argumentet  $Uttr$  må forenkles til et reelt tall.

Sett det blå elementet på RGB LED-skjermen til halv intensitet.

$lum:=127$	127
Send "SET COLOR.BLUE eval(lum)"	Done

Tilbakestill det blå elementet til AV.

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

$eval()$ -argumentet må forenkles til et reelt tall.

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

Programmer for å fade inn det røde elementet

```

Define fadein()=
Prgm
For i,0,255,10
  Send "SET COLOR.RED eval(i)"
  Wait 0.1
EndFor
Send "SET COLOR.RED OFF"
EndPrgm

```

Utfør programmet.

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

Selv om resultatet av **eval()** ikke vises, kan du se den resulterende hub-kommandostrengen etter at du har utført kommandoen ved å inspisere hvilken som helst av følgende spesielle variabler.

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

**Merk:** Les også **Get**(side 62), **GetStr** (side 69) og **Send** (side 141).

## Exit (Avslutt)

Katalog > 

### Exit

Avslutter aktuell **For**, **While**, eller **Loop**-blokk.

**Exit** er ikke tillatt utenfor de tre løkkestrukturene (**For**, **While**, eller **Loop**).

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

### Program:

```

Define g()=Func
Local temp,i
0→temp
For i,1,100,1
temp+i→temp
If temp>20 Then
Exit
EndIf
EndFor
EndFunc

```

<i>g()</i>	21
------------	----

## exp()

 **tast**

**exp(Verdi1)⇒verdi**

$e^1$	2.71828
$e^{3^2}$	8103.08

Returnerer **e** opphøyd i *Uttr1*-potens.

Returnerer **e** opphøyd i *Verdi1*-potens.

**Merk:** Se også **e** eksponent-sjablon, side 2.

Du kan legge inn et komplekst tall i rei  $\theta$  polar form. Men bruk denne formen bare i radian-vinkelmodus; den forårsaker grunnmengdefeil i grader- eller gradian-vinkelmodus.

**exp(Liste1)⇒liste**

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

Returnerer tallet **e** opphøyd i potens av hvert element i *Liste1*.

**exp(kvadratMatrise1)⇒kvadratMatrise**

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

Returnerer kvadratMatrise som er **e** opphøyd i *kvadratMatrise1*. Dette er ikke det samme som å beregne **e** opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

## expr() (uttrykk)

**Katalog** > 

**expr (String)⇒Uttrykk**

"Define cube(x)=x^3" →funcstr	
"Define cube(x)=x^3"	
expr(funcstr)	Done
cube(2)	8

Returnerer tegnstrengen som ligger i *Streng* som et uttrykk og utfører den straks.

## ExpReg

**Katalog** > 

**ExpReg** *X*, *Y* [, [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den eksponensielle regresjonen  $y = a \cdot (b)^x$  for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).



Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall  $\geq 0$ .

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).



Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot (b)^x$
stat.a, stat.b	Regresjonskoeffisienter
stat.r <sup>2</sup>	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data ( $x, \ln(y)$ )
stat.Resid	Residualene for den eksponensielle modellen
stat.ResidTrans	Rester tilordnet ved lineær tilpasning av transformerte data
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> , og <i>Inkludert kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>inkludert kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

**factor() (faktor)**Katalog > 

**factor(*rasjonaltTall*)** returnerer det rasjonale tallet faktorisert i primtall. For sammensatte tall øker behandlingstiden eksponensielt med antallet siffer i den nest største faktoren. For eksempel kan det ta mer enn en hel dag å faktorisere et heltall med 30 siffer, og å faktorisere et tall med 100 siffer kan ta mer enn et århundre.

<code>factor(152417172689)</code>	123457·1234577
<code>isPrime(152417172689)</code>	false

Slik stopper du en beregning manuelt,

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

Hvis du bare vil bestemme om et tall er et primtall, bruk **isPrime()** istedenfor. Det er mye raskere, særlig hvis *rasjonaltTall* ikke er et primtall og hvis den nest største faktoren består av mer enn fem siffer.

**F Cdf()**Katalog > **F Cdf**

(  
*nedGrense*  
,*øvGrense*,*dfTeller*,*dfNevner*)⇒*tall* hvis  
*nedGrens* og *øvGrens* er tall, *liste* hvis  
*nedGrens* og *øvGrens* er lister

**FCdf**

(  
*nedGrense*  
,*øvGrense*,*dfTeller*,*dfNevner*)⇒*tall* hvis  
*nedGrens* og *øvGrens* er tall, *liste* hvis  
*nedGrens* og *øvGrens* er lister

Beregner  $F$  fordelingssannsynligheten mellom *nedGrense* og *ovGrense* for spesifisert *dfTeller* (frihetsgrader) og *dfNevner*.

For  $P(X \leq \text{ovGrens})$ , set *nedGrens* = 0.

## Fill (Fyll)

Fill *Verdi*, *matriseVar*  $\Rightarrow$  *matrise*

Erstatter hvert element i variabel *matriseVar* med *Verdi*.

*matriseVar* må eksistere allerede.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\rightarrow$ <i>amatrix</i>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01, <i>amatrix</i>		Done
<i>amatrix</i>		$\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$

Fill *Verdi*, *listeVar*  $\Rightarrow$  *liste*

Erstatter hvert element i variabel *listeVar* med *Verdi*.

*listeVar* må eksistere allerede.

$\{1,2,3,4,5\}$	$\rightarrow$ <i>alist</i>	$\{1,2,3,4,5\}$
Fill 1.01, <i>alist</i>		Done
<i>alist</i>		$\{1.01,1.01,1.01,1.01,1.01\}$

## FiveNumSummary

FiveNumSummary *X*[,*Frekv*]  
[,*Kategori*,*Inkluder*]]

Gir en forkortet versjon av den 1-variabels statistiske observatoren på listen *X*.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154.)

*X* representerer en liste med dataene.

*Frekv* er en valgfri liste med frekvensverdier.

Hvert element i *Frekv* angir hvor ofte hver korresponderende *X*-verdi forekommer.

Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X*-dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene  $X$ ,  $Frekv$  eller  $Kategori$  resulterer i et åpent element for det tilsvarende elementet til alle disse listene. For mer informasjon om tomme elementer, se side 223.

Utdata-variabel	Beskrivelse
stat.MinX	Minimum av x-verdiene
stat.Q <sub>1</sub> X	Første kvartil av x
stat.MedianX	Medianen av x
stat.Q <sub>3</sub> X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdiene

**floor() (nedre)**

**floor(Verdi)** ⇒ heltall

$\text{floor}(-2.14)$  -3.

Returnerer det største heltallet som er  $\leq$  argumentet. Denne funksjonen er identisk med **int()**.

Argumentet kan være et reelt eller et komplekst tall.

**floor(Liste1)** ⇒ liste

$\text{floor}\left(\left\{\frac{3}{2}, 0, -5.3\right\}\right)$  {1, 0, -6.}

**floor(Matrise1)** ⇒ matrise

$\text{floor}\left(\begin{pmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{pmatrix}\right)$  
 $\begin{bmatrix} 1. & 3. \\ 2. & 4. \end{bmatrix}$

Returnerer en liste eller matrise med nedre verdi for hvert element.

**Merk:** Se også **ceiling()** og **int()**.

**For**

**For** *Var*, *Lav*, *Høy* [, *Intervall*]

Define  $g()$  = Func Done

*Blokk*

Local *tempsum*, *step*, *i*

0 → *tempsum*

1 → *step*

For  $i, 1, 100, \text{step}$

$\text{tempsum} + i \rightarrow \text{tempsum}$

EndFor

EndFunc

**EndFor**

Utfører utsagnene i *Blokk* iterativt for hver verdi av *Var*, fra *Lav* til *Høy*, i trinn på *Intervall*.

*Var* må ikke være en systemvariabel.

$g()$  5050

*Intervall* kan være positiv eller negativ. Grunnverdien er 1.

*Blokk* kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet “.”.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

## format()

**format(Verdi[, formatStreng])**⇒streng

Returnerer *Verdi* som en tegnstring basert på formatsjablonen.

*formatStreng* er en streng og må være av formen: “F[n]”, “S[n]”, “E[n]”, “G[n][c]”, hvor [ ] viser alternative muligheter.

F[n]: Fast format. n er antallet siffer som vises etter desimalpunktet.

V[n]: Vitenskapelig format. n er antallet siffer som vises etter desimalpunktet.

T[n]: Teknisk format. n er antallet siffer etter det første signifikante sifferet. Eksponenten er tilpasset til et multiplum av tre, og desimalpunktet er flyttet til høyre med sifrene null, ett eller to.

G[n][c]: Samme som fast format, men skiller også sifrene til venstre for basen i grupper på tre. c spesifiserer gruppens og basens skilletegn som et komma. Hvis c er en periode, vises basen som et komma.

[Rk]: Som etterledd bak noen av spesifikantene over kan basemerket Rc tilføyes, der hvor c er et enkelt tegn som spesifiserer hva som erstatter komma.

format(1.234567, "f3")	"1.235"
format(1.234567, "s2")	"1.23E0"
format(1.234567, "e3")	"1.235E0"
format(1.234567, "g3")	"1.235"
format(1234.567, "g3")	"1,234.567"
format(1.234567, "g3,r:")	"1:235"

**fPart() (funksjonsdel)**Katalog > **fPart**(*Uttr1*) $\Rightarrow$ uttrykk $fPart(-1.234)$  -0.234**fPart**(*Liste1*) $\Rightarrow$ liste $fPart(\{1,-2,3,7.003\})$   $\{0,-0.3,0.003\}$ **fPart**(*Matrise1*) $\Rightarrow$ matrise

Returnerer brøk-delen i argumentet.

For en liste eller matrise, returneres brøk-delene i elementene.

Argumentet kan være et reelt eller et komplekst tall.

**FPdf()**Katalog > **FPdf**(*XVerdi*,*dfTeller*,*dfNevner*) $\Rightarrow$ tall hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste**FPdf**(*XVerdi*,*dfTeller*,*dfNevner*) $\Rightarrow$ tall hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en listeBeregner F fordelingssannsynligheten mellom *XVerdi* for den spesifiserte *dfTeller* (grader av frihet) og *dfNevner*.**freqTable►liste()**katalog > **freqTable►liste****(Liste1, frekvHeltallListe)** $\Rightarrow$ liste $freqTable►list(\{1,2,3,4\},\{1,4,3,1\})$   
 $\{1,2,2,2,2,3,3,3,4\}$ Returnerer en liste som inneholder elementene fra *Liste1* utvidet i henhold til frekvensene i *frekvHeltallListe*. Denne funksjonen kan brukes til å generere en frekvenstabell for applikasjonen Data og statistikk. $freqTable►list(\{1,2,3,4\},\{1,4,0,1\})$   
 $\{1,2,2,2,2,4\}$ *Liste1* kan være enhver gyldig liste.*frekvHeltallListe* må ha samme dimensjon som *Liste1* og kun inneholde ikke-negative heltallselementer. Hvert element angir hvor mange ganger det korresponderende *Liste1*-elementet skal gjentas i resultatlisten. En verdi lik null utelater det korresponderende *Liste1*-elementet.

**Merk:** Du kan sette inn denne funksjonen fra datamaskinstaturet ved å skrive `freqTable@>list(...)`.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

## frequency() (frekvens)

`frequency(Liste1, stolperListe) ⇒ liste`

Returnerer en liste som inneholder antallet elementer i *Liste1*. Antallet er basert på områder (stolper) som du definerer i *stolperListe*.

Hvis *stolperListe* er  $\{b(1), b(2), \dots, b(n)\}$ , er de spesifiserte områdene  $\{? \leq b(1), b(1) < ? \leq b(2), \dots, b(n-1) < ? \leq b(n), b(n) > ?\}$ . Den resulterende listen er ett element lenger enn *stolperListe*.

Hvert element av resultatet samsvarer med antallet elementer fra *Liste1* som er i området for den stolpen. Uttrykt med begrep fra **countIf()**-funksjonen er resultatet  $\{\text{countIf}(\text{liste}, ? \leq b(1)), \text{countIf}(\text{liste}, b(1) < ? \leq b(2)), \dots, \text{countIf}(\text{liste}, b(n-1) < ? \leq b(n)), \text{countIf}(\text{liste}, b(n) > ?)\}$ .

Elementer fra *Liste1* som ikke kan "plasseres i en stolpe" ignoreres. Tomme (åpne) elementer ignoreres også. For mer informasjon om tomme elementer, se side 223.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor begge argumentene.

**Merk:** Se også **countIf()**, side 32.

<code>dataliste = { 1, 2, e, 3, pi, 4, 5, 6, "hello", 7 }</code>	
<code>{ 1, 2, 2.71828, 3, 3.14159, 4, 5, 6, "hello", 7 }</code>	
<code>frequency(dataliste, { 2.5, 4.5 })</code>	<code>{ 2, 4, 3 }</code>

Forklaring til resultat:

**2** elementer fra *Dataliste* er  $\leq 2,5$

**4** elementer fra *Dataliste* er  $> 2,5$  og  $\leq 4,5$

**3** elementer fra *Dataliste* er  $> 4,5$

Elementet "hallo" er en streng og kan ikke plasseres i nopen av de definerte stolpene.

## F Test\_2Samp (2\_utvalg F test)

`FTest_2Samp(Liste1, Liste2[, Frekv1[, Frekv2[, Hypot]])`

`FTest_2Samp(Liste1, Liste2[, Frekv1[, Frekv2`

[,Hypot]]

(Dataliste inndata)

FTest\_2Samp *sx1,n1,sx2,n2[,Hypot]*FTest\_2Samp *sx1,n1,sx2,n2[,Hypot]*

(Summering statistikk inndata)

Utfører en to-utvalgs F test. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

eller  $H_a: \sigma_1 > \sigma_2$ , sett *Hypoth*>0

For  $H_a: \sigma_1 \neq \sigma_2$  (standard), sett *Hypoth*=0

For  $H_a: \sigma_1 < \sigma_2$ , sett *Hypoth*<0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" på side 223.

Utdata-variabel	Beskrivelse
stat.F	Beregnet $\hat{U}$ -statistikk for datasekvensen
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.dfNumer	frihetsgrad for teller = $n_1 - 1$
stat.dfDenom	frihetsgrad for nevner = $n_2 - 1$
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste 2</i>
stat.x1_bar stat.x2_bar	Utvalgets gjennomsnitt av datasekvenser i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Utvalgenes størrelse

**Func (Funk)****Func***Blokk***EndFunc**

Definere en sammensatt funksjon:

Sjablon for oppretting av brukerdefinert funksjon.

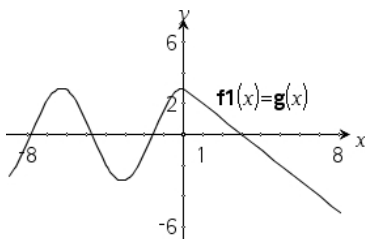


*Blokk* kan være ett enkelt utsagn, en rekke utsagn adskilt med “:”-tegnet, eller en rekke med utsagn på separate linjer. Funksjonen kan bruke **Returner**-kommandoen for å returnere et spesifikt resultat.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

```
Define g(x)=Func
  If x<0 Then
    Return 3*cos(x)
  Else
    Return 3-x
  EndIf
EndFunc
```

Resultat av grafisk fremstilling g(x)



## G

### gcd() (største felles divisor)

**gcd(Tall1, Tall2)** ⇒ uttrykk

$\text{gcd}(18,33)$  3

Returnerer største felles divisor for de to argumentene. **gcd** av to brøker er **gcd** av tellerne dividert med **lcm** av nevnerne.

I modusen **Auto** eller **Tilnærmet** er **gcd** av brøkens flytende desimalpunktall 1,0.

**gcd(Liste1, Liste2)** ⇒ liste

$\text{gcd}(\{12,14,16\},\{9,7,5\})$  {3,7,1}

Returnerer største felles divisorer av samsvarende deler i *Liste1* og *Liste2*.

**gcd(Matrise1, Matrise2)** ⇒ matrise

$\text{gcd}\left(\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}, \begin{pmatrix} 4 & 8 \\ 12 & 16 \end{pmatrix}\right)$   $\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$

Returnerer største felles divisorer av samsvarende deler i *Matrise1* og *Matrise2*.

### geomCdf()

**geomCdf(p, nedreGrense, øvreGrense)** ⇒ tall  
hvis *nedreGrense* og *øvreGrense* er tall,  
*liste* hvis *nedreGrense* og *øvreGrense* er

lister

**geomCdf**( $p, \text{\textit{\textit{øvreGrense}}}$ ) for  $P(1 \leq X \leq \text{\textit{\textit{øvreGrense}}}) \Rightarrow$  tall hvis  $\text{\textit{\textit{øvreGrense}}}$  er et tall, liste hvis  $\text{\textit{\textit{øvreGrense}}}$  er en liste

Beregner en kumulativ geometrisk sannsynlighet fra  $\text{\textit{\textit{nedreGrense}}}$  til  $\text{\textit{\textit{øvreGrense}}}$  med den spesifiserte sannsynligheten for suksess  $p$ .

For  $P(X \leq \text{\textit{\textit{øvreGrense}}})$ , sett  $\text{\textit{\textit{nedreGrense}}} = 1$ .

## geomPdf()

**geomPdf**( $p, X\text{Verdi}$ )  $\Rightarrow$  tall hvis  $X\text{Verdi}$  er et tall, liste hvis  $X\text{Verdi}$  er en liste

Beregner en sannsynlighet ved  $X\text{Verdi}$ , antall forsøk før første suksess inntreffer, for diskret geometrisk fordeling med spesifisert suksess-sannsynligheten  $p$ .

## Get

**Get**[ $\text{\textit{\textit{ledetekstStreng}}}$ ,]  $\text{var}$ [,  $\text{\textit{\textit{statusVar}}}$ ]

**Get**[ $\text{\textit{\textit{ledetekstStreng}}}$ ,]  $\text{funk}$ ( $\text{\textit{\textit{arg1}}}$ , ... $\text{\textit{\textit{argn}}}$ )[,  $\text{\textit{\textit{statusVar}}}$ ]

Programmeringskommando: Henter en verdi fra en tilkoblet TI-Innovator™ Hub og tildeler verdien til variabel  $\text{\textit{\textit{var}}}$ .

Verdien må etterspørres:

- På forhånd, gjennom en **Send «LES ...»**-kommando.  
–eller–
- Ved å innlemme en «LES ...»-forespørsel som det alternative  $\text{\textit{\textit{ledetekstStreng}}}$ -argumentet. Denne metoden lar deg bruke en enkel kommando for å etterspørre verdien og hente den.

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. Bruk **Get** for å hente verdien og tildele den til variabelen  $\text{\textit{\textit{lysver}}}$ .

Send "READ BRIGHTNESS"	Done
Get $\text{\textit{\textit{lightval}}}$	Done
$\text{\textit{\textit{lightval}}}$	0.347922

Innlem LES-forespørslen i **Hent**-kommandoen.

Get "READ BRIGHTNESS", $\text{\textit{\textit{lightval}}}$	Done
$\text{\textit{\textit{lightval}}}$	0.378441

Implisitt forenkling finner sted. For eksempel tolkes en mottatt streng som «123» som en numerisk verdi. For å bevare strengen, bruker du **GetStr** i stedet for **Get**.

Hvis du inkluderer det valgfrie argumentet *statusVar*, tilordnes det en verdi basert på om operasjonen lyktes eller ikke. En verdi på null betyr at ingen data ble mottatt.

I den andre syntaksen lar argumentet *funk()* et program lagre den mottatte strengen som en funksjonsdefinisjon. Denne syntaksen arbeider som om programmet utførte kommandoen:

Definer *funk(arg1, ...argn) = mottatt streng*

Programmet kan så bruke den definerte funksjonen *funk()*.

**Merk:** Du kan bruke **Get**-kommandoen i et brukerdefinert program, men ikke i en funksjon.

**Merk:** Se også **GetStr**, side 69 og **Send**, side 141.

## getDenom() (lesNevner)

Katalog > 

**getDenom(Brøkl) ⇒ verdi**

Omformer argumentet inn til et uttrykk som har en redusert felles nevner og returnerer så uttrykkets nevner.

$x:=5; y:=6$	6
$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	3
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	30

## getKey()

Katalog > 

**getKey([0|1]) ⇒ returnString**

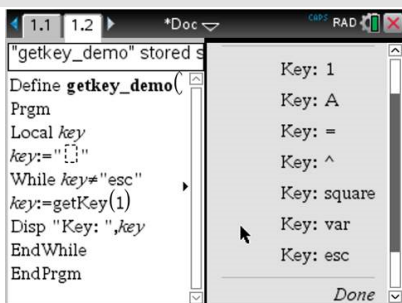
`getKey()`

Eksempel:

**Beskrivelse:** `getKey()` – tillater at et TI-Basic-program henter tastaturinndata – grafregner, stasjonær PC og emulator på skrivebordet.

#### Eksempel:

- `keypressed := getKey()` returnerer en tast eller en tom streng hvis ingen tast er trykket ned. Dette oppkallet returneres umiddelbart.
- `keypressed := getKey(1)` venter til en tast trykkes ned. Dette oppkallet setter utførelsen av programmet på pause til en tast trykkes ned.



#### Håndtering av tastetrykk:

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
Esc	Esc	"esc"
Styreplate – klikk oppe	Ikke relevant	"opp"
På	Ikke relevant	"start"
Kladdeapp	Ikke relevant	"kladdeblokk"
Styreplate – venstreklikk	Ikke relevant	"venstre"
Styreplate – midtklikk	Ikke relevant	"midtre"
Styreplate – høyreklikk	Ikke relevant	"høyre"
Dok	Ikke relevant	"dok"
Kategori	Kategori	"kategori"
Styreplate – klikk nede	Piltast ned	"ned"
Meny	Ikke relevant	"meny"
Ctrl	Ctrl	ingen retur
Skift	Skift	ingen retur
Variabel	Ikke relevant	"var"
Del	Ikke relevant	"del"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
=	=	"="
trigonometri	Ikke relevant	"trigonometri"
0 til og med 9	0-9	"0" ... "9"
Sjabloner	Ikke relevant	"sjablon"
Katalog	Ikke relevant	"kat"
^	^	"^"
X^2	Ikke relevant	"kvadrat"
/ (divisjonstast)	/	"/"
* (multiplikasjonstast)	*	"*"
e^x	Ikke relevant	"eksp"
10^x	Ikke relevant	"10potens"
+	+	"+"
-	-	"_"
(	(	"{"
)	)	"}"
.	.	". "
(-)	Ikke relevant	"_" (negativ-tegn)
Enter	Enter	"enter"
ee	Ikke relevant	"E" (vitenskapelig notasjon E)
a-z	a-z	alpha = bokstav trykket ned (liten bokstav) ("a"- "z")
skift a-z	skift a-z	alpha = bokstav trykket ned "A"- "Z"
		Merk: ctrl-skift brukes som Caps Lock
?!	Ikke relevant	"?!"
pi	Ikke relevant	"pi"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
Flagg	Ikke relevant	ingen retur
,	,	" , "
Returner	Ikke relevant	"returner"
mellomrom	mellomrom	" " (mellomrom)
Utilgjengelig	Spesialtegn som f.eks. @,!,^ osv.	Tegnet er returnert
Ikke relevant	Funksjonstaster	Ingen returnerte tegn
Ikke relevant	Spesielle skrivebordskontrolltaster	Ingen returnerte tegn
Utilgjengelig	Andre skrivebordstaster som ikke er tilgjengelige på kalkulatoren mens getKey() venter på et tastetrykk. ({, }, ;, ...)	Det samme tegnet du får i Notat (ikke i en matematikkboks)

**Merk:** Legg merke til at **getKey()** i et program endrer hvordan hendelser behandles av systemet. Noen av disse beskrives nedenfor.

**Avslutte et program og behandle en hendelse** – På samme måte som om brukeren skulle avslutte et program ved å trykke på **PÅ**-tasten

"**Støtte**" nedenfor betyr – Systemet fungerer som forventet – programmet fortsetter å kjøre.

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Student Software
Hurtigspørring	Avslutte et program, behandle en hendelse	Samme som for grafregner (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Ekstern filbehandling (Inkl. sende filen 'Exit Press 2 Test' fra en annen grafregner eller skrivebords-grafregner)	Avslutte et program, behandle en hendelse	Samme som for en grafregner. (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Avslutt klasse	Avslutte et program, behandle en hendelse	Brukerstøtte (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Alle versjoner
TI-Innovator™ Hub koble til / koble fra	Støtte – kan sende kommandoer til TI-Innovator™ Hub. Etter at du har avsluttet programmet, fungerer TI-Innovator™ Hub fremdeles med grafregneren.	Samme som for en grafregner

## getLangInfo()

katalog > 

**getLangInfo()** ⇒ *streng*

`getLangInfo()`

"en"

Returnerer en streng som svarer til kortnavnet på det aktive språket. Du kan for eksempel bruke den i et program eller en funksjon for å finne aktivt språk.

Engelsk = "en"

Dansk = "da"

Tysk = "de"

Finsk = "fi"

Fransk = "fr"

Italiensk = "it"

Nederlandsk = "nl"

Belgisk nederlandsk = "nl\_BE"

Norsk = "no"

Portugisisk = "pt"

Spansk = "es"

Svensk = "sv"

## getLockInfo()

Katalog > 

**getLockInfo**(*Var*)⇒*verdi*

Returnerer aktuell låst/opplåst status for variabel *Var*.

*verdi* =0: *Var* er låst opp eller eksisterer ikke.

*verdi* =1: *Var* er låst opp og kan ikke modifiseres eller slettes.

Se **Lock**, side 90, og **unLock**, side 173.

<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

## GetMode() (lesModus)

Katalog > 

**GetMode**(*ModusNavnHeltall*)⇒*verdi*

**GetMode**(0)⇒*liste*

**GetMode**(*ModusNavnHeltall*) returnerer en verdi som representerer aktuell innstilling av *ModusNavnHeltall*-modus.

**GetMode**(0) returnerer en liste som inneholder tallpar. Hvert par består av et modusheltall og et innstillingsheltall.

For en opplisting av modusene og deres innstillinger, referer til tabellen under.

Hvis du lagrer innstillingene med **GetMode**(0) → *var*, kan du bruke **GetMode**(*var*) i en funksjon eller et program for midlertidig å gjenopprette innstillingene kun innenfor utføringen av funksjonen eller programmet. Se **GetMode**(), side 144.

getMode(0)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1 }
getMode(1)	7
getMode(7)	1

Modus Navn	Modus Heltall	Innstille heltall
Vis sifre	1	1=Flytende, 2=Flytende1, 3=Flytende2, 4=Flytende3, 5=Flytende4, 6=Flytende5, 7=Flytende6, 8=Flytende7, 9=Flytende8, 10=Flytende9, 11=Flytende10, 12=Flytende11, 13=Flytende12, 14=Fast0, 15=Fast1, 16=Fast2, 17=Fast3, 18=Fast4, 19=Fast5, 20=Fast6, 21=Fast7, 22=Fast8, 23=Fast9, 24=Fast10, 25=Fast11, 26=Fast12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian



Modus Navn	Modus Heltall	Innstillinger heltall
Eksponensielt format	3	1=Normal, 2=Vitenskapelig, 3=Teknisk
Reell eller kompleks	4	1=Reell, 2=Rektangulær, 3=Polar
Auto eller tilnærm.	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær

## getNum() (lesTeller)

Katalog > 

`getNum(Brøkl)` ⇒ verdi

$x:=5; y:=6$  6

Omformert argumentet til et uttrykk som har en redusert felles nevner og returnerer så uttrykkets teller.

$\text{getNum}\left(\frac{x+2}{y-3}\right)$  7

$\text{getNum}\left(\frac{2}{7}\right)$  2

$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$  11

## GetStr

Hub-meny

`GetStr[ledetekstStreng,] var[, statusVar]`

For eksempler, se **Get**.

`GetStr[ledetekstStreng,] funk(arg1, ...argn)`  
[, statusVar]

Programmeringskommando: Virker på nøyaktig samme måte som kommandoen **Get**, bortsett fra at mottatt verdi alltid tolkes som en streng. I motsetning tolker kommandoen **Get** svaret som et uttrykk, med mindre det er satt i anførselstegn ("").

**Merk:** Se også **Get**, side 62 og **Send**, side 141.

**getType**(*var*)⇒*streng*

Returnerer en streng som angir dataens typevariabel *var*.

Hvis *var* ikke er definert, returnerer strengen "INGEN".

{1,2,3}→ <i>temp</i>	{1,2,3}
getType( <i>temp</i> )	"LIST"
3· <i>i</i> → <i>temp</i>	3· <i>i</i>
getType( <i>temp</i> )	"EXPR"
DelVar <i>temp</i>	Done
getType( <i>temp</i> )	"NONE"

getVarInfo()

**getVarInfo**()⇒*matrise* eller *streng*

**getVarInfo**

(*BibliotekNavnStreng*)⇒*matrise* eller *streng*

**getVarInfo**() returnerer en matrise med informasjon (variabelnavn, type, bibliotektilgjengelighet og låst/opplåst status) for alle variabler og biblioteksobjekter som er definert i den aktuelle oppgaven.

Hvis ingen variabler er definert, returnerer **getVarInfo**() strengen "INGEN".

**getVarInfo**

(*BibliotekNavnStreng*)returnerer en matrise med informasjon for alle bibliotekobjektene som er definert i biblioteket *BibliotekNavnStreng*. *BibliotekNavnStreng* må være en streng (tekst omsluttet av anførselstegn) eller en strengvariabel.

Hvis biblioteket *BibliotekNavnStreng* ikke finnes, oppstår det en feil.

getVarInfo()	"NONE"												
Define <i>x</i> =5	Done												
Lock <i>x</i>	Done												
Define LibPriv <i>y</i> ={1,2,3}	Done												
Define LibPub <i>z</i> ( <i>x</i> )=3· <i>x</i> <sup>2</sup> - <i>x</i>	Done												
getVarInfo()	<table border="1"> <tr> <td><i>x</i></td> <td>"NUM"</td> <td>"{ }"</td> <td>1</td> </tr> <tr> <td><i>y</i></td> <td>"LIST"</td> <td>"LibPriv "</td> <td>0</td> </tr> <tr> <td><i>z</i></td> <td>"FUNC"</td> <td>"LibPub "</td> <td>0</td> </tr> </table>	<i>x</i>	"NUM"	"{ }"	1	<i>y</i>	"LIST"	"LibPriv "	0	<i>z</i>	"FUNC"	"LibPub "	0
<i>x</i>	"NUM"	"{ }"	1										
<i>y</i>	"LIST"	"LibPriv "	0										
<i>z</i>	"FUNC"	"LibPub "	0										
getVarInfo( <i>tmp3</i> )	"Error: Argument must be a string"												
getVarInfo("tmp3")	[ <i>volcy12</i> "NONE" "LibPub " 0]												

Se for eksempel til venstre, der resultatet av **getVarInfo()** tilordnes variabelen *vs*. Hvis du forsøker å vise rad 2 eller 3 av *vs*, returneres en "Ugyldig liste eller matrise"-feil, siden minst ett av elementene i de radene (for eksempel variabel *b*) reevalueres til en matrise.

Denne feilen kan også oppstå når du bruker *Ans* til å reevaluere et **getVarInfo()**-resultat.

Systemet viser ovenstående feil fordi den gjeldende versjonen av programvaren ikke støtter en generalisert matrisestruktur der et element kan være enten en matrise eller en liste.

$a:=1$	1												
$b:=[1\ 2]$	$[1\ 2]$												
$c:=[1\ 3\ 7]$	$[1\ 3\ 7]$												
$vs:=getVarInfo()$	<table border="1"> <tr> <td><i>a</i></td> <td>"NUM"</td> <td>"{}"</td> <td>0</td> </tr> <tr> <td><i>b</i></td> <td>"MAT"</td> <td>"{}"</td> <td>0</td> </tr> <tr> <td><i>c</i></td> <td>"MAT"</td> <td>"{}"</td> <td>0</td> </tr> </table>	<i>a</i>	"NUM"	"{}"	0	<i>b</i>	"MAT"	"{}"	0	<i>c</i>	"MAT"	"{}"	0
<i>a</i>	"NUM"	"{}"	0										
<i>b</i>	"MAT"	"{}"	0										
<i>c</i>	"MAT"	"{}"	0										
$vs[1]$	$[1\ "NUM"\ "{}"\ 0]$												
$vs[1,1]$	1												
$vs[2]$	"Error: Invalid list or matrix"												
$vs[2,1]$	$[1\ 2]$												

## Goto (Gåtil)

### Goto etikettNavn

Overfører kontroll til navnet *etikettNavn*.

*etikettNavn* må være definert i samme funksjon med en **Lbl**-instruksjon.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

Define $g()$ =Func	Done
Local <i>temp,i</i>	
0 $\rightarrow$ <i>temp</i>	
1 $\rightarrow$ <i>i</i>	
Lbl <i>top</i>	
<i>temp</i> + <i>i</i> $\rightarrow$ <i>temp</i>	
If <i>i</i> <10 Then	
<i>i</i> +1 $\rightarrow$ <i>i</i>	
Goto <i>top</i>	
EndIf	
Return <i>temp</i>	
EndFunc	
$g()$	55

## ►Grad

### Uttr1 ► Grad $\Rightarrow$ Uttrykk

Omregner *Uttr1* til gradian vinkelmåling.

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive **@>Grad**.

I Grader-vinkelmodus:

$(1.5)$ ►Grad	$(1.66667)^{\circ}$
---------------	---------------------

I Radian-vinkelmodus:

$(1.5)$ ►Grad	$(95.493)^{\circ}$
---------------	--------------------

**identity()**Katalog > **identity(Heltall)** ⇒ *matrise*

Returnerer identitetsmatrisen med dimensjonen *Heltall*.

*Heltallet* må være et positivt heltall.

identity(4)	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1

**Hvis**Katalog > 

**Hvis** *BooleanExpr*  
*Utsagn*

**Hvis** *BooleanExpr*, så  
*Blokk*

**OgHvis**

Hvis *BooleanExpr* behandles som sann, utføres det enkle utsagnet *Utsagn* eller blokken av utsagn *Blokk* før utførelsen fortsetter.

Hvis *BooleanExpr* behandles som usann, fortsettes utførelsen uten å utføre utsagnet eller blokken av utsagn.

*Blokk* kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet «:».

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

**Hvis** *BooleanExpr*, så  
*Blokk1*

**hvis ikke**  
*Blokk2*

**OgHvis**

Hvis *BooleanExpr* behandles som sann, utføres *Blokk1* og utelater så *Blokk2*.

Hvis *BooleanExpr* behandles som usann, utelates *Blokk1*, men *Blokk2* utføres.

Define $g(x)=$ Func	<i>Done</i>
If $x<0$ Then	
Return $x^2$	
EndIf	
EndFunc	
$g(-2)$	4

Define $g(x)=$ Func	<i>Done</i>
If $x<0$ Then	
Return $-x$	
Else	
Return $x$	
EndIf	
EndFunc	
$g(12)$	12
$g(-12)$	12

*Blokk1* og *Blokk2* kan være et enkelt utsagn.

**Hvis** *BooleanExpr1*, så  
*Blokk1*

**EllersHvis** *BooleanExpr2*, så  
*Blokk2*

:

**EllersHvis** *BooleanExprN*, så  
*BlokkN*

**OgHvis**

Tillater forgreining. Hvis *BooleanExpr1* behandles som sann, utføres *Blokk1*.

Hvis *BooleanExpr1* behandles som usann, behandles *BooleanExpr2*, og så videre.

Define $g(x)$ =Func	
If $x < 5$ 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	
	<i>Done</i>
$g(-4)$	4
$g(10)$	3

## ifFn()

**ifFn**(*BooleanExpr*, *Value* *If true*  
[, *Value* *If false* [, *Value* *If unknown*]])  
 $\Rightarrow$  uttrykk, liste eller matrise

Behandler det boolske uttrykket *BooleanExpr* (eller hvert element fra *BooleanExpr* ) og produserer et resultat basert på følgende regler:

- *BooleanExpr* kan teste en enkel verdi, en liste eller en matrise.
- Hvis et element av *BooleanExpr* behandles som sann, returneres det samsvarende elementet fra *Value\_If\_true*.
- Hvis et element av *BooleanExpr* behandles som usann, returneres det samsvarende elementet fra *Value\_If\_false*. Hvis du utelater *Value\_If\_false*, returneres undef.
- Hvis et element av *BooleanExpr* hverken er sant eller usant, returneres det samsvarende elementet *Value\_If\_unknown*. Hvis du utelater *Value\_If\_unknown*, returneres undef.
- Hvis det andre, tredje eller fjerde argumentet i **ifFn()**-funksjonen et

ifFn({1,2,3}<2.5,{5,6,7},{8,9,10})	{5,6,10}
------------------------------------	----------

Testverdi på **1** er mindre enn 2,5, så det er samsvarende

*Value\_If\_True*-element av **5** kopieres til resultatlisten.

Testverdi på **2** er mindre enn 2,5, så det er samsvarende

*Value\_If\_True*-element av **6** kopieres til resultatlisten.

Testverdi på **3** er ikke mindre enn 2,5, så det samsvarende *Value\_If\_False*-elementet på **10** kopieres til resultatlisten.

ifFn({1,2,3}<2.5,4,{8,9,10})	{4,4,10}
------------------------------	----------

*Value\_If\_true* er en enkel verdi og samsvarer med hvilken som helst valgt posisjon.

## ifFn()

Katalog > 

enkelt uttrykk, brukes det boolske uttrykket i hver posisjon i *BooleanExpr*.

**Merk:** Hvis det forenklede utsagnet *BooleanExpr* involverer en liste eller matrise, må alle andre liste- eller matriseargumenter ha samme dimensjoner, og resultatet ha samme dimensjoner.

---

$$\text{ifFn}(\{1,2,3\} < 2.5, \{5,6,7\}) \quad \{5,6,\text{undef}\}$$

---

*Value\_If\_false* er ikke spesifisert. Udef brukes.

---

$$\text{ifFn}(\{2, "a" \} < 2.5, \{6,7\}, \{9,10\}, "err") \quad \{6, "err" \}$$

---

Et element velges fra *Value\_If\_true*. Et element velges fra *Value\_If\_unknown*.

## imag()

Katalog > 

**imag(ValueI)** ⇒ *verdi*

Returnerer den imaginære delen av argumentet.

---

$$\text{imag}(1+2 \cdot i) \quad 2$$

---

**imag(ListI)** ⇒ *liste*

Returnerer en liste over de imaginære delene av elementene.

---

$$\text{imag}(\{-3,4-i,i\}) \quad \{0,-1,1\}$$

---

**imag(MatrixI)** ⇒ *matrise*

Returnerer en matrise over de imaginære delene av elementene.

---

$$\text{imag}\left(\begin{bmatrix} 1 & 2 \\ i \cdot 3 & i \cdot 4 \end{bmatrix}\right) \quad \begin{bmatrix} 0 & 0 \\ 3 & 4 \end{bmatrix}$$

---

## Indireksjon

Se #(), side 200.

## inString()

Katalog > 

**inString(srcString, subString[, Start])** ⇒ *heltall*

Returnerer tegnposisjonen i strengen *srcString* der første forekomst av strengen *subString* begynner.

*Start*, hvis det er inkludert, spesifiserer tegnposisjonen innenfor *srcString* der søket starter. Standard = 1 (det første tegnet i *srcString*).

---

$$\text{inString}(\text{"Hello there"}, \text{"the"}) \quad 7$$

---

---

$$\text{inString}(\text{"ABCEFG"}, \text{"D"}) \quad 0$$

---

## inString()

Katalog > 

Hvis *srcString* ikke inneholder *subString* eller *Start* er > lengden av *srcString*, returneres null.

## int()

Katalog > 

**int(Verdi)** ⇒ *heltall*

**int(List1)** ⇒ *liste*

**int(Matrix1)** ⇒ *matrise*

Returnerer det største heltallet som er mindre enn eller lik argumentet. Denne funksjonen er identisk med **floor()**.

Argumentet kan være et reelt eller et komplekst tall.

For en liste eller matrise, returneres det største heltallet for hvert element.

$\text{int}(-2.5)$	-3.
$\text{int}([-1.234 \ 0 \ 0.37])$	$[-2. \ 0 \ 0.]$

## intDiv()

Katalog > 

**intDiv(Number1, Number2)** ⇒ *heltall*

**intDiv(List1, List2)** ⇒ *liste*

**intDiv(Matrix1, Matrix2)** ⇒ *matrise*

Returnerer heltallsdelen med fortegn av (*Number1* ÷ *Number2*).

For lister og matriser, returneres heltallsdelen med fortegn av (argument 1 ÷ argument 2) for hvert elementpar.

$\text{intDiv}(-7,2)$	-3
$\text{intDiv}(4,5)$	0
$\text{intDiv}(\{12, -14, -16\}, \{5, 4, -3\})$	$\{2, -3, 5\}$

## interpoler ()

Katalog > 

**interpoler(xValue, xList, yList, yPrimeList)** ⇒ *liste*

Denne funksjonen gjør følgende:

Differensialligning:

$y' = -3 \cdot y + 6 \cdot t + 5$  og  $y(0) = 5$

$rK = rk23(-3 \cdot y + 6 \cdot t + 5, t, y, \{0, 10\}, 5, 1)$					
0.	1.	2.	3.	4.	
5.	3.19499	5.00394	6.99957	9.00593	10

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

## interpoler ()

Katalog > 

Gitt  $xList$ ,  $yList=f(xList)$  og  $yPrimeList=f'(xList)$  for en ukjent funksjon  $f$ , brukes en kubisk interpolant for å tilnærme funksjonen  $f$  ved  $xValue$ . Det antas at  $xList$  er en liste over monotont stigende eller synkende tall, men denne funksjonen kan returnere en verdi selv om den ikke er det. Denne funksjonen går gjennom  $xList$  og ser etter et intervall  $[xList[i], xList[i+1]]$  som inneholder  $xValue$ . Hvis den finner et slikt intervall, returnerer den en interpolert verdi for  $f(xValue)$ , ellers returnerer den **undef**.

$xList$ ,  $yList$  og  $yPrimeList$  må være av lik dimensjon  $\geq 2$  og inneholde uttrykk som forenkles til tall.

$xValue$  kan være et tall eller en liste med tall.

Bruk den interpolerte() funksjonen for å beregne funksjonens verdier for x-verdilisten:

```
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.}
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.,1.41503,1.98819,2.00129,1.98221,2.006}
interpolate(xvalueList,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011,7.00011}
```

## invχ<sup>2</sup>()

Katalog > 

**invχ<sup>2</sup>(Area,df)**

**invChi2(Area,df)**

Beregner invers kumulativ  $\chi^2$  (chi-kvadrat) sannsynlighetsfunksjon spesifisert av frihetsgrad,  $df$  for et gitt *Område* under kurven.

## invF()

Katalog > 

**invF(Area,dfNumer,dfDenom)**

**invF(Area,dfNumer,dfDenom)**

Beregner invers kumulativ F-fordelingsfunksjon spesifisert av  $dfNumer$  og  $dfDenom$  for et gitt *Område* under kurven



## invBinom()

Katalog > 

### invBinom

(*CumulativeProb*, *NumTrials*, *Prob*, *OutputForm*) ⇒ skalar eller matrise

Invers binomial. Gitt antall forsøk (*NumTrials*) og sannsynligheten for å lykkes for hvert forsøk (*Prob*). Denne funksjonen returnerer minimum antall suksesser, *k*, slik at verdien, *k*, er større eller lik den oppgitte kumulative sannsynligheten (*CumulativeProb*).

*OutputForm*=0 viser resultat som en skalar (standard).

*OutputForm*=1 viser resultat som en matrise.

Eksempel: Mary og Kevin spiller med terninger. Mary skal gjette maksimalt antall ganger 6 vises på 30 kast. Hvis 6 vises så mange eller færre ganger, vinner Mary. I tillegg vinner hun mer jo mindre tallet hun gjetter er. Hva er det minste tallet Mary kan gjette hvis hun ønsker en sannsynlighet for å vinne som er større enn 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()

Katalog > 

invBinomN(*CumulativeProb*, *Prob*, *NumSuccess*, *OutputForm*) ⇒ skalar eller matrise

Invers binomial med hensyn på *N*. Gitt sannsynligheten for å lykkes med hvert forsøk (*Prob*), og antall suksesser (*NumSuccess*), returnerer denne funksjonen minimum antall forsøk, *N*, slik at verdien, *N*, er mindre eller lik den kumulative sannsynligheten (*CumulativeProb*).

*OutputForm*=0 viser resultat som en skalar (standard).

*OutputForm*=1 viser resultat som en matrise.

Eksempel: Monique øver på målskudd for nettbull. Fra erfaring vet hun at det er 70 % sjanse for at hun treffer med hvilket som helst skudd. Hun har tenkt å holde på til hun skårer 50 mål. Hvor mange skudd må hun forsøke for å sikre at sannsynligheten for å treffe med minst 50 skudd er mer enn 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()

Katalog > 

invNorm(*Area*,  $\mu$ ,  $\sigma$ )

Beregner den inverse, kumulative normale fordelingsfunksjonen for et gitt *område* under den normale fordelingskurven som er spesifisert av  $\mu$  og  $\sigma$ .

**invnt**(*Area,df*)

Beregner invers kumulativ sannsynlighetsfunksjon for student-t spesifisert av frihetsgrad, *df* for et gitt *Område* under kurven.

**iPart**()**iPart**(*Number*) ⇒ *heltall***iPart**(*List1*) ⇒ *liste***iPart**(*Matrix1*) ⇒ *matrise*

$iPart(-1.234)$	-1.
$iPart\left(\left\{\frac{3}{2}, -2.3, 7.003\right\}\right)$	{1, -2, 7.}

Returnerer heltallsdelen av argumentet.

For lister og matriser, returnerer heltallsdelen for hvert element.

Argumentet kan være et reelt eller et komplekst tall.

**irr()****irr**(*CF0,CFList* [,*CFFreq*]) ⇒ *verdi*

Finansiell funksjon som beregner internrente av retur av en investering.

*CF0* er kontantstrømmen ved start kl. 0. Den må være et reelt tall.

*CFList* er en liste over kontantstrømbeløpene etter den innledende kontantstrømmen *CF0*.

*CFFreq* er en valgfri liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (etterfølgende) kontantstrømbeløp, som er det tilsvarende elementet til *CFList*. Standarden er 1. Hvis du legger inn verdier, må dette være positive heltall < 10 000.

**Merk:** Se også **mirr()**, side 99.

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$irr(5000, list1, list2)$	-4.64484

## isPrime()

Katalog > 

**isPrime**(*Number*) ⇒ *Boolsk konstant uttrykk*

Returnerer sann eller usann for å vise om *tall* er et helt tall  $\geq 2$  som bare er delelig med seg selv og 1.

Hvis *Tall* består av mer enn 306 siffer og ikke inneholder noen faktorer, viser  $\leq 1021$ , **isPrime**(*Number*) en feilmelding

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

isPrime(5)	true
isPrime(6)	false

Funksjon for å finne det neste primtallet etter et spesifisert tall:

Define <i>nextprim</i> ( <i>n</i> )=Func	<i>Done</i>
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*) ⇒ *Boolsk konstant uttrykk*  
**isVoid**(*Expr*) ⇒ *Boolsk konstant uttrykk*  
**isVoid**(*List*) ⇒ *liste over boolske konstante uttrykk*

Returnerer sann eller usann for å vise om utsagnet er en åpen (tom) datatype.

For mer informasjon om tomme elementer, se side 223.

<i>a</i> :=_	_
isVoid( <i>a</i> )	true
isVoid({ 1,_,3 })	{ false,true,false }

## L

## Lbl (Nvn)

Katalog > 

### Lbl *etikettNavn*

Definerer en etikett med navnet *etikettNavn* innenfor en funksjon.

Du kan bruke en **Goto** *etikettNavn* - instruksjon for å overføre kontroll til den instruksjonen som umiddelbart følger etter etiketten.

*etikettNavn* må følge de samme reglene for navn som gjelder for variabelnavn.

Define <i>g</i> ()=Func	<i>Done</i>
Local <i>temp</i> , <i>i</i>	
0 → <i>temp</i>	
1 → <i>i</i>	
Lbl <i>top</i>	
<i>temp</i> + <i>i</i> → <i>temp</i>	
If <i>i</i> <10 Then	
<i>i</i> +1 → <i>i</i>	
Goto <i>top</i>	
EndIf	
Return <i>temp</i>	
EndFunc	
<i>g</i> ()	55

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

**lcm()** (mfm)

**lcm**(Tall1, Tall2)⇒uttrykk

lcm(6,9)	18
----------	----

**lcm**(Liste1, Liste2)⇒liste

lcm( $\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}$ )	$\left\{\frac{2}{3}, 14, 80\right\}$
---	--------------------------------------

**lcm**(Matrise1, Matrise2)⇒matrise

Returnerer minste felles multiplum av de to argumentene. **lcm** av to brøker er **lcm** av tellerne dividert med **gcd** av nevnerne. **lcm** av brøk består av flytende desimalpunktall er produktet av teller og nevner.

For to lister eller matriser, returnerer minste felles multiplum for samsvarende elementer.

**left()** (venstre)

**left**(kildeStreng[, Num])⇒streng

left("Hello", 2)	"He"
------------------	------

Returnerer de *Num*-tegnene som ligger lengst til venstre i tegnstrengen *kildeStreng*.

Hvis du utelater *Num*, returneres alle i *kildeStreng*.

**left**(Liste1[, Num])⇒liste

left({1,3,-2,4}, 3)	{1,3,-2}
---------------------	----------

Returnerer de *Num*-elementene som ligger lengst til venstre i *Liste1*.

Hvis du utelater *Num*, returneres alle elementer i *Liste1*.

**left**(Sammenlikning)⇒Uttrykk

Returnerer venstre side av en ligning eller ulikhet.

**libShortcut**(*BibliotekNavnStreng*,  
*HurtigtastNavnStreng*,  
*BiblPrivMerke*)⇒*liste av variabler*

Oppretter en variabelgruppe i den gjeldende oppgaven som inneholder referanser til alle objektene i det angitte bibliotekdokumentet *bibliotekNavnStreng*. Legger også gruppelemmene til i Variablermenyen. Deretter kan du referere til hvert objekt ved å bruke dets *HurtigtastNavnStreng*.

Sett *BiblPrivMerke*=0 hvis du skal ekskludere private bibliotekobjekter (standard)

Sett *BiblPrivMerke*=1 hvis du skal inkludere private bibliotekobjekter

Hvis du skal kopiere en variabelgruppe, se **CopyVar** (side 25).

Hvis du skal slette en variabelgruppe, se **DelVar** (side 40).

Dette eksemplet forutsetter et riktig lagret og oppdatert bibliotekdokument med navnet **linalg2** som inneholder objekter definert som *clearmat*, *gauss1*, og *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 │           │
└──────────┴──────────┴──────────┴──────────┘
```

## LinRegBx (lineær regresjon)

**LinRegBx** *X*, *Y*, [*Frekv*], [*Kategori*, *Inkluder*]

Finner den lineære regresjonen  $y = a + b \cdot x$  for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X* og *Y*-dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot x$
stat.a, stat.b	Regresjonskoeffisienter
stat.r <sup>2</sup>	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

**LinRegMx**  $X, Y[, Frekv[, Kategori, Inkluder]]$

Finner den lineære regresjonen  $y = m \cdot x + b$  for listene  $X$  og  $Y$  med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $m \cdot x + b$
stat.m, stat.b	Regresjonskoeffisienter
stat.r <sup>2</sup>	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

**LinRegtIntervals**  $X, Y[, F[, 0[, CNivå]]]$

For stigningstall. Beregner et konfidensintervall med konfidensnivå  $C$  for stigningstallet.

**LinRegtIntervals**  $X, Y[, F[, 1, Xverd[, CNivå]]]$

For respons. Beregner en predikert  $y$ -verdi, et prediksjonsintervall med nivå  $C$  for én enkelt observasjon, og et konfidensintervall med nivå  $C$  for den gjennomsnittlige responsen.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154.)

Alle listene må ha samme dimensjon.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

$F$  er en valgfri liste over frekvensverdier. Hvert element i  $F$  spesifiserer frekvensen av forekomst for hvert tilsvarende  $X$  og  $Y$  datapunkt. Standardverdien er 1. Alle elementene må være heltall  $\geq 0$ .

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot x$
stat.a, stat.b	Regresjonskoeffisienter
stat.gf	Frihetsgrader
stat.r <sup>2</sup>	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Resid	Residualene fra regresjonen

Gjelder kun stigningstall

Utdata-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensintervall for stigningstallet
stat.ME	Konfidensintervallets feilmargin
stat.SESlope	Standardfeil for stigningstallet
stat.s	Standardfeil rundt linjen

Gjelder kun responstype

Utdata-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensintervall for gjennomsnittlig respons
stat.ME	Konfidensintervallets feilmargin
stat.SE	Standardfeil for gjennomsnittlig respons



Utdata-variabel	Beskrivelse
[stat.LowerPred, stat.UpperPred]	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat.ŷ	$a + b \cdot X$ Verdi

## LinRegtTest

katalog > 

### LinRegtTest $X, Y[, Frekv[, Hypot]]$

Beregner en lineær regresjon på  $X$ - og  $Y$ -listene og en  $t$  test på verdien av stigningstallet  $\beta$  og korrelasjonskoeffisienten  $\rho$  for ligningen  $y = \alpha + \beta x$ . Den tester nullhypotesen  $H_0: \beta = 0$  (tilsvarende,  $\rho = 0$ ) mot én av tre alternative hypoteser.

Alle listene må ha samme dimensjon.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

$Frekv$  er en valgfri liste med frekvensverdier. Hvert element  $i$   $Frekv$  angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

$Hypot$  er en valgfri verdi som angir en av tre alternative hypoteser som nullhypotesen ( $H_0: \beta = \rho = 0$ ) skal testes mot.

For  $H_a: \beta \neq 0$  og  $\rho \neq 0$  (standard), sett  $Hypot = 0$

For  $H_a: \beta < 0$  og  $\rho < 0$ , sett  $Hypot < 0$

For  $H_a: \beta > 0$  og  $\rho > 0$ , sett  $Hypot > 0$

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a + b \cdot x$
stat.t	$t$ -observator for signifikanstest
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Frihetsgrader
stat.a, stat.b	Regresjonskoeffisienter
stat.s	Standardfeil rundt linjen
stat.SESlope	Standardfeil for stigningstallet
stat.r <sup>2</sup>	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Resid	Residualene fra regresjonen

## linSolve()

Katalog > 

**linSolve**(SystemAvLineæreLign, Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\begin{cases} 2 \cdot x + 4 \cdot y = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{cases}, \{x, y\}\right) \quad \begin{cases} 37 \\ 26 \end{cases}, \begin{cases} 1 \\ 26 \end{cases}$$

**linSolve**(LineærLign1 and LineærLign2 and ..., Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\begin{cases} 2 \cdot x = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{cases}, \{x, y\}\right) \quad \begin{cases} 3 \\ 2 \end{cases}, \begin{cases} 1 \\ 6 \end{cases}$$

**linSolve**({LineærLign1, LineærLign2, ...}, Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\begin{cases} \text{apple} + 4 \cdot \text{pear} = 23 \\ 5 \cdot \text{apple} - \text{pear} = 17 \end{cases}, \{\text{apple}, \text{pear}\}\right) \quad \begin{cases} 13 \\ 3 \end{cases}, \begin{cases} 14 \\ 3 \end{cases}$$

**linSolve**(SystemAvLineæreLign, {Var1, Var2, ...}) ⇒ liste

$$\text{linSolve}\left(\begin{cases} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{cases}, \{\text{apple}, \text{pear}\}\right) \quad \begin{cases} 36 \\ 13 \end{cases}, \begin{cases} 114 \\ 13 \end{cases}$$

**linSolve**(LineærLign1 and LineærLign2 and ..., {Var1, Var2, ...}) ⇒ liste

**linSolve**({LineærLign1, LineærLign2, ...}, {Var1, Var2, ...}) ⇒ liste

Returnerer en liste over løsninger for variablene Var1, Var2, ...

Det første argumentet må behandles som et system av lineære ligninger eller som en lineær ligning. Ellers oppstår det en argumentfeil.

Ved for eksempel å behandle linSolve(x=1 og x=2,x) produserer et "Argumentfeil" -resultat.

## $\Delta$ list() (liste)

Katalog >  $\Delta$ List(Liste1)  $\Rightarrow$  liste $\Delta$ List({20,30,45,70})

{10,15,25}

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive `deltaList (...)`.

Returnerer en liste som inneholder differensene mellom nabo-elementer i *Liste1*. Hvert element i *Liste1* subtraheres fra det neste elementet i *Liste1*. Resultatlisten er alltid ett element kortere enn opprinnelige *Liste1*.

## list $\blacktriangleright$ mat()

Katalog > list $\blacktriangleright$ mat(Liste [, elementerRerRad])  $\Rightarrow$  matrise

list $\blacktriangleright$ mat({1,2,3})	[1 2 3]						
list $\blacktriangleright$ mat({1,2,3,4,5},2)	<table border="1"><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr><tr><td>5</td><td>0</td></tr></table>	1	2	3	4	5	0
1	2						
3	4						
5	0						


Returnerer en matrise fylt rad-for-rad med elementene fra *Liste*.

*elementerRerRad*, hvis inkludert, spesifiserer antallet elementer per rad. Grunninnstilling er antallet elementer i *Liste* (en rad).

Hvis *Liste* ikke fyller resultatmatrisen, legges det til nuller.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `list@>mat (...)`.

## ln()

  **taster**ln(Verdi1)  $\Rightarrow$  verdi

ln(2.)

0.693147

ln(Liste1)  $\Rightarrow$  liste

Returnerer argumentets naturlige logaritme.

Hvis kompleks formatmodus er reell:

For en liste, returneres elementenes naturlige logaritmer.

ln({-3,1.2,5})
"Error: Non-real calculation"

Hvis kompleks formatmodus er rektangulær:

$$\ln(\{-3, 1.2, 5\})$$

$$\{1.09861+3.14159 \cdot i, 0.182322, 1.60944\}$$

**ln(kvadratMatrise)⇒kvadratMatrise**

Returnerer matrisens naturlige logaritme av *kvadratMatrise1*. Dette er ikke det samme som å beregne naturlig logaritme av hvert element. For mer informasjon om beregningsmetode, se etter under **cos()**

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I radian-vinkelmodus og rektangulært, kompleks format:

$$\ln \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

**LnReg****LnReg X, Y[, [Frekv] [, Kategori, Inkludert]]**

Flner den logaritmiske regresjonen  $y = a+b \cdot \ln(x)$  for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X* og *Y*-dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot \ln(x)$
stat.a, stat.b	Regresjonskoeffisienter
stat.r <sup>2</sup>	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data ( $\ln(x)$ , $y$ )
stat.Resid	Residualene for den logaritmiske modellen
stat.ResidTrans	Residualene for den lineære tilpasningen av de transformerte dataene
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## Local

Katalog > 

**Local** *Var1* [, *Var2*] [, *Var3*] ...

Deklarerer spesifiserte *vars* som lokale variabler. Disse variablene eksisterer kun mens en funksjon utføres og slettes når funksjonen er ferdig utført.

**Merk:** Lokale variabler sparer plass i minnet, fordi de bare eksisterer midlertidig. Dessuten forstyrrer de ingen eksisterende globale variabelverdier. Bruk lokale variabler for **For** -stigningstall og for midlertidige lagringsverdier i en flerlinjet funksjon, siden endringer på globale variabler ikke er tillatt i en funksjon.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

```

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

**Lock***Var1* [, *Var2*] [, *Var3*] ...

**Lock***Var*.

Låser spesifiserte variabler eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes.



Du kan ikke låse eller låse opp systemvariabelen *\$var*, og du kan ikke låse systemvariabelgruppene *stat*. eller *tvm*.

**Merk:** Kommandoen **Lås (Lock)** tømmer angre/gjør om-loggen når den brukes på ulåste variabler.

Se **unlock**, side 173 og **getLockInfo()**, side 68.

<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()

  **taster**

**log**(*Verdi1* [, *Verdi2*])⇒*verdi*

$$\log_{10} (2.) = 0.30103$$

**log**(*Liste1* [, *Verdi2*])⇒*liste*

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

Returnerer grunntallet -*Verdi2* argumentets logaritme.

$$\log_3 (10) - \log_3 (5) = 0.63093$$

**Merk:** Se også **Log-sjablon**, side 2.

For en liste, returneres grunntall -*Verdi2* for elementenes logaritme.

Hvis kompleks formatmodus er reell:

$$\log_{10} (\{-3,1.2,5\})$$

"Error: Non-real calculation"

Hvis *Uttr2* utelates, brukes 10 som grunntall.

Hvis kompleks formatmodus er rektangulær:

$$\log_{10} (\{-3,1.2,5\})$$

$$\{0.477121+1.36438 \cdot i, 0.079181, 0.69897\}$$

**log**(*kvadratMatrise1* [, *Verdi2*])⇒*kvadratMatrise*

I radian-vinkelmodus og rektangulært, kompleks format:

## log()

ctrl 10<sup>x</sup> taster

Returnerer matrisens grunn- *Verdi2* logaritme av *kvadratMatrise1*. Dette er ikke det samme som å beregne grunntallet- *Verdi2* logaritme av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

Hvis grunntall-argumentet utelates, brukes 10 som grunntall.

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.795387+0.753438 \cdot i & 0.003993-0.6474 \cdot i \\ 0.194895-0.315095 \cdot i & 0.462485+0.2707 \cdot i \\ -0.115909-0.904706 \cdot i & 0.488304+0.7774 \cdot i \end{bmatrix}$$

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

## Logistic

katalog > 

**Logistic** *X*, *Y*, [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den logistiske regresjonen  $y = (c / (1 + a \cdot e^{-bx}))$  for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X* og *Y*-dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regresjonskoeffisienter
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## LogisticD

katalog > 

**LogisticD** *X*, *Y* [, [*Iterasjoner*], [*Frekv*] [, *Kategori*, *Inkluder*] ]

Finner den logistiske regresjonen  $y = (c/(1+a \cdot e^{-bx})+d)$  for listene *X* og *Y* med frekvensen *Frekv*, ved å bruke et angitt antall *Iterasjoner*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Iterasjoner* er en valgfri verdi som angir maksimalt antall ganger det skal gjøres forsøk på å finne en løsning. Hvis utelatt, brukes 64. Vanligvis resulterer større verdier i bedre nøyaktighet men lengre kjøretid, og omvendt.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.



*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X* og *Y*-dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $c/(1+a \cdot e^{-bx})+d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluderte kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluderte kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## Loop (Stigningstall)

### Loop

*Blokk*

### EndLoop

Utfører utsagnene gjentatte ganger i *Blokk*. Merk at stigningstallet utføres uendelig, hvis ikke en **Goto** eller **Exit** instruksjon utføres innenfor *Blokk*.

*Blokk* er en sekvens av utsagn som er adskilt med tegnet.

```

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
    
```

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.



**LU (= nedre/øvre)**

**LU** *Matrise*, *lMatrise*, *uMatrise*, *pMatrise*, *Tol*

Beregner Doolittle LU (lower-upper=nedre-øvre) dekomposisjon av en reell eller kompleks matrise. Den nedre trekantede matrisen lagres i *lMatrise*, den øvre trekantede matrisen i *uMatrise* og permutasjonsmatrisen (som beskriver radskiftene som gjøres i løpet av beregningen) i *pMatrise*.

$$lMatrise \cdot uMatrise = pMatrise \cdot matrise$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \max(\dim(Matrise)) \cdot \text{radNorm}(Matrise)$

Faktorerende algoritme **LU** bruker delvis pivotering med radutvekslinger.

6 12 18	→ m1	6 12 18
5 14 31		5 14 31
3 8 18		3 8 18

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	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

**mat►list() (matrise til liste)**Katalog > **mat►list(Matrise)**⇒*liste*

Returnerer en liste fylt med elementene i *Matrise*. Elementene kopieres fra *Matrise* rad for rad.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **mat@>list(...)**.

$\text{mat}\blacktriangleright\text{list}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}\right)$	$\{1, 2, 3\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$\text{mat}\blacktriangleright\text{list}(mI)$	$\{1, 2, 3, 4, 5, 6\}$

**max()**Katalog > **max(Verdi1, Verdi2)**⇒*Uttrykk***max(Liste1, Liste2)**⇒*liste***max(Matrise1, Matrise2)**⇒*matrise*

Returnerer maksimum (det største) av de to argumentene. Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder maksimum verdi i hvert par av samsvarende elementer.

**max(Liste)**⇒*Uttrykk*Returnerer maksimumelementet i *liste*.**max(Matrise1)**⇒*matrise*

Returnerer en radvektor som inneholder maksimumselementet av hver kolonne i *Matrise1*.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

**Merk:** Se også **min()**.

$\text{max}(2.3, 1.4)$	2.3
$\text{max}(\{1, 2\}, \{-4, 3\})$	$\{1, 3\}$

$\text{max}(\{0, 1, -7, 1.3, 0.5\})$	1.3
--------------------------------------	-----

$\text{max}\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$	$[1 \ 0 \ 7]$
---	---------------

**mean() (gjennomsnitt)**Katalog > **mean(Liste[, frekvListe])**⇒*uttrykk*

Returnerer gjennomsnittet av elementene i *Liste*.

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$	$\frac{5}{3}$

## mean() (gjennomsnitt)

Katalog > 

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**mean(MatriseI[, FrekvMatrise])**  
⇒ *matrise*

Returnerer en radvektor av gjennomsnittet for alle kolonnene i *MatriseI*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *MatriseI*.

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 223.

I rektangulært vektorformat:

$$\begin{array}{l} \text{mean} \left( \begin{array}{cc} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{array} \right) \quad \left[ -0.133333 \quad 0.833333 \right] \\ \text{mean} \left( \begin{array}{cc} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & -\frac{1}{2} \end{array} \right) \quad \left[ \frac{-2}{15} \quad \frac{5}{6} \right] \\ \text{mean} \left( \begin{array}{cc|cc} 1 & 2 & 5 & 3 \\ 3 & 4 & 4 & 1 \\ 5 & 6 & 6 & 2 \end{array} \right) \quad \left[ \frac{47}{15} \quad \frac{11}{3} \right] \end{array}$$

## median()

Katalog > 

**median(Liste[, frekvListe])** ⇒ *uttrykk*

Returnerer medianen av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**median(MatriseI[, frekvMatrise])** ⇒ *matrise*

Returnerer en radvektor som inneholder medianer av kolonnene i *MatriseI*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *MatriseI*.

$$\text{median}(\{0.2, 0, 1, -0.3, 0.4\}) \quad 0.2$$

$$\text{median} \left( \begin{array}{cc} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{array} \right) \quad \left[ 0.4 \quad -0.3 \right]$$

### Merknader:

- Alle inndata i listen eller matrisen må forenkles til tall.
- Tomme (åpne) elementer i listen eller matrisen ignorerer. For mer informasjon om tomme elementer, se side 223.

**MedMed**  $X, Y$  [, *Frekv*] [, *Kategori*,  
*Inkluder*]]

Beregner median-median-linjens  $= (m \cdot x + b)$  for listene  $X$  og  $Y$  med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Ligning for median-median-linjen: $m \cdot x + b$
stat.m, stat.b	Modellkoeffisienter
stat.Resid	Residualene fra median-median-linjen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## mid() (midtstreng)

Katalog > 

**mid(kildeStreng, Start[, Antall])** ⇒ streng

Returnerer *Antall* tegn fra tegnstreng *kildeStreng*, begynnende med tegnnummer *Start*.

Hvis *Antall* utelates eller er større enn dimensjonen på *kildeStreng*, returneres alle tegnene fra *kildeStreng*, begynnende med tegnnummer *Start*.

*Antall* må være  $\geq 0$ . Hvis *Antall* = 0, returneres en tom streng.

**mid(kildeListe, Start [, Antall])** ⇒ liste

Returnerer *Antall* elementer fra *kildeListe*, begynnende med elementnummer *Start*.

Hvis *Antall* utelates eller er større enn dimensjonen på *kildeListe*, returneres alle elementer fra *kildeListe*, begynnende med elementnummer *Start*.

*Antall* må være  $\geq 0$ . Hvis *Antall* = 0, returneres en tom liste.

**mid(kildeStrengListe, Start[, Antall])** ⇒ liste

Returnerer *Antall* strenger fra listen over strenger *kildeStrengListe*, begynnende med elementnummer *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() (minimum)

Katalog > 

**min(Verdi1, Verdi2)** ⇒ Uttrykk

**min(Liste1, Liste2)** ⇒ liste

**min(Matrise1, Matrise2)** ⇒ matrise

Returnerer minimum (det minste) av de to argumentene. Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder minimumsverdien i hvert par av samsvarende elementer.

min(2,3,1,4)	1,4
min({1,2},{-4,3})	{-4,2}

**min()** (minimum)Katalog > **min(Liste)**⇒Uttrykk $\min(\{0,1,-7,1,3,0,5\})$  -7

Returnerer minimumselementet av *Liste*.

**min(Matrise!)**⇒matrise
$$\min\left(\begin{array}{ccc} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{array}\right) \quad [-4 \quad -3 \quad 0.3]$$

Returnerer en radvektor som inneholder minimumselementet av hver kolonne i *Matrise1*.

**Merk:** Se også **max()**.

**mirr()**Katalog > **mirr**

(*finansRente, reinvestRente, CF0, CFListe* [, *CFFrekv*])

$$\begin{array}{l} list1 := \{6000, -8000, 2000, -3000\} \\ \{6000, -8000, 2000, -3000\} \\ list2 := \{2, 2, 2, 1\} \\ \{2, 2, 2, 1\} \\ mirr(4.65, 12, 5000, list1, list2) \quad 13.41608607 \end{array}$$

Finansiell funksjon som returnerer modifisert rente av en investering.

*finansRente* er den renten som du betaler på kontantstrømbeløpene.

*reinvestRente* er den renten som kontantstrømmen reinvesteres til.

*CF0* er kontantstrømmen ved start kl. 0; den må være et reelt tall.

*CFListe* er en liste over kontantstrømbeløpene etter den innledende kontantstrømmen *CF0*.

*CFFrekv* er en valgfri liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (konsekutivt) kontantstrømbeløp, som er det tilsvarende elementet til *CFListe*. Grunninnstilling er 1; hvis du legger inn verdier, må dette være positive heltall < 10.000.

**Merk:** Se også **irr()**, side 78.

**mod()**Katalog > **mod**(Verdi1, Verdi2) ⇒ Uttrykk $\text{mod}(7,0)$  7**mod**(Liste1, Liste2) ⇒ liste $\text{mod}(7,3)$  1**mod**(Matrise1, Matrise2) ⇒ matrise $\text{mod}(-7,3)$  2

Returnerer det første argumentet modulo det andre argumentet slik som definert ved identitetene:

 $\text{mod}(7,-3)$  -2 $\text{mod}(-7,-3)$  -1 $\text{mod}(\{12,-14,16\},\{9,7,-5\})$  {3,0,-4} $\text{mod}(x,0) = x$  $\text{mod}(x,y) = x - y \text{ floor}(x/y)$ 

Hvis det andre argumentet ikke er null, er resultatet periodisk i dette argumentet. Resultatet er enten null eller det har samme fortegn som det andre argumentet.

Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder modulen av hvert par av samsvarende elementer.

**Merk:** Se også **rest()**, side 131**mRow() (mRad)**Katalog > **mRow**(Verdi, Matrise1, Indeks) ⇒ matrise
$$\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & \frac{-4}{3} \\ 3 & \end{bmatrix}$$

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks* av *Matrise1* multiplisert med *Verdi*.

**mRowAdd() (mRadAdd)**Katalog > **mRowAdd**(Verdi, Matrise1, Indeks1, Indeks2) ⇒ matrise
$$\text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$$

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks2* i *Matrise1* erstattet med:

 $Verdi \times \text{rad Indeks1} + \text{rad Indeks2}$



**MultReg**  $Y, XI[,X2[,X3,...[,X10]]]$ 

Beregner multiple lineære regresjoner av liste  $Y$  for listene  $X2, X2, \dots, X10$ . En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.b0, stat.b1, ...	Regresjonskoeffisienter
stat.R <sup>2</sup>	Multipel determinasjonskoeffisient
stat.ŷList	$\hat{y}List = b_0+b_1 \cdot x_1+ \dots$
stat.Resid	Residualene fra regresjonen

**MultRegIntervals****MultRegIntervals**  $Y, XI[,X2[,X3,...[,X10]]], XVerListe[,CNivå]$ 

Beregner en predikert  $y$ -verdi, et prediksjonsintervall med nivå  $C$  for én enkelt observasjon, og et konfidensintervall med nivå  $C$  for den gjennomsnittlige responsen.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.ŷ	Et punktestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XVerListe</i>
stat.dfError	Feilens frihetsgrader

Utdata-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensintervall for gjennomsnittlig respons
stat.ME	Konfidensintervall feilmargin
stat.SE	Standardfeil for gjennomsnittlig respons
stat.LowerPred, stat.UpperPred	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat.bList	Liste over regresjonskoeffisienter, {b0,b1,b2,...}
stat.Resid	Residualene fra regresjonen

## MultRegTests

katalog > 

### MultRegTests $Y, X1[,X2[,X3,...[,X10]]]$

Multipel lineær regresjonstest beregner en multipel lineær regresjon på de angitte dataene og beregner den globale statistiske  $F$ - og  $t$ -testobservatoren for koeffisientene.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).



### Utdata

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.F	Global statistisk $F$ -testobservator
stat.PVal	P-verdi knyttet til global $F$ -observator
stat.R <sup>2</sup>	Multipel determinasjonskoeffisient
stat.AdjR <sup>2</sup>	Justert multipel determinasjonskoeffisient
stat.s	Standardavvik for feilen
stat.DW	Durbin-Watson-observator. Brukes for å bestemme om første ordens autokorrelasjon er til stede i modellen

Utdata-variabel	Beskrivelse
stat.dfReg	Regresjonens frihetsgrader
stat.SSReg	Regresjonens kvadratsum
stat.MSReg	Regresjonens kvadratgjennomsnitt
stat.dfError	Feilens frihetsgrader
stat.SKvFeil	Feilens kvadratsum
stat.MSError	Feilens kvadratgjennomsnitt (gjennomsnittlig kvadratavvik)
stat.bList	{b0,b1,...} Liste over koeffisienter
stat.tList	Liste over statistiske t-observatorer, én for hver koeffisient i bList
stat.PList	Liste over P-verdier for hver t-observator
stat.SEList	List over standardfeil for koeffisientene i bList
stat.yList	$\hat{y}$ List = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualene fra regresjonen
stat.sResid	Standardiserte residualer. Beregnes ved å dividere en restverdi (residual) med dens standardavvik
stat.CookDist	Cooks distanse. Mål for innflytelsen av en observasjon basert på residual og stigning
stat.Leverage	Mål for hvor langt verdiene for den uavhengige variabelen er fra gjennomsnittsverdiene

## N

### nand (ikke både...og)

  -taster

*BoolskUttr1* nand *BoolskUttr2*  
returnerer *Boolsk uttrykk*

*BoolskListe1* nand *BoolskListe2*  
returnerer *Boolsk liste*

*BoolskMatrise1* nand *BoolskMatrise2*  
returnerer *Boolsk matrise*

## nand (ikke både...og)

ctrl [=]-taster

Returnerer negasjon av en logisk and-handling på de to argumentene.  
Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

*Heltall1 nand Heltall2*⇒*heltall*

Sammenlikner to reelle heltall bit-for-bit med en nand-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 0 hvis begge bitene er 1; ellers er resultatet 1. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 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() (antKomb)

Katalog > 

*nCr(Verdi1, Verdi2)*⇒*Uttrykk*

For heltall *Verdi1* og *Verdi2* der  $Verdi1 \geq Verdi2 \geq 0$ , er **nCr()** antall kombinasjoner av *Verdi1* som tar *Verdi2* om gangen. (Dette er også kjent som en binomisk koeffisient.)

*nCr(Verdi, 0)*⇒**1**

*nCr(Verdi, negHeltall)*⇒**0**

*nCr(Verdi, posHeltall)*⇒ *Verdi · (Verdi-1) · ... · (Verdi-posHeltall+1) / posHeltall!*

*nCr(Verdi, ikkeHeltall)*⇒*Uttrykk!*  
*((Verdi-ikkeHeltall)! · ikkeHeltall!)*

*nCr(Liste1, Liste2)*⇒*liste*

<i>nCr(z,3)</i> <sub>z=5</sub>	10
<i>nCr(z,3)</i> <sub>z=6</sub>	20

<i>nCr({5,4,3}, {2,4,2})</i>	{10,1,3}
------------------------------	----------

Returnerer en liste over kombinasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

**nCr(Matrise1, Matrise2)**⇒matrise

Returnerer en matrise av kombinasjoner basert på samsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

$$\text{nCr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) \quad \begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

## nDerivative()

**nDerivative(Utr1, Var=Verdi [, Orden])**⇒verdi

**nDerivative(Utr1, Var[, Orden]) | Var=Verdi**⇒verdi

Returnerer den numeriske deriverte som er beregnet ved hjelp av automatiske derivasjonsmetoder.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsstilt verdi eller aktuell “|” erstatning for variabelen.

Hvis variabelen *Var* ikke inneholder en numerisk verdi, må du oppgi en *Verdi*.

Den deriverte må være av orden **1** eller **2**.

**Merk:** Den nDerivative() algoritmen har en begrensning: den arbeider rekursivt gjennom det ikke-forenklede uttrykket og beregner den numeriske verdien av den første deriverte (og eventuelt den andre) og behandlingen av hvert deluttrykk, som kan føre til et uventet resultat.

Studer eksemplet til høyre. Den første deriverte av  $x \cdot (x^2+x)^{1/3}$  i  $x=0$  er lik 0. Men siden den første deriverte av deluttrykket  $(x^2+x)^{1/3}$  er udefinert i  $x=0$  og denne verdien blir brukt til å beregne den deriverte av hele uttrykket, rapporterer **nDerivative()** resultatet som udefinert og viser en varselmelding.

nDerivative( x ,x=1)	1
nDerivative( x ,x) x=0	undef
nDerivative(√(x-1),x) x=1	undef

nDerivative(x·(x <sup>2</sup> +x) <sup>1/3</sup> ,x,1) x=0	undef
centralDiff(x·(x <sup>2</sup> +x) <sup>1/3</sup> ,x) x=0	0.000033

## nDerivative()

Katalog > 

Hvis du støter på denne begrensningen, verifiserer du løsningen grafisk. Du kan også prøve å bruke **centralDiff()**.

## newList() (nyListe)

Katalog > 

**newList(numElementer)** ⇒ liste

<code>newList(4)</code>	$\{0,0,0,0\}$
-------------------------	---------------

Returnerer en liste med en dimensjon lik *numElementer*. Hvert element er null.

## newMat() (nyMat)

Katalog > 

**newMat(numRader, numKolonner)** ⇒ matrise

<code>newMat(2,3)</code>	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
--------------------------	--

Returnerer en matrise med bare nuller med dimensjonen *numRader* og *numKolonner*.

## nfMax()

Katalog > 

**nfMax(Uttr1, Var)** ⇒ verdi

<code>nfMax(-x<sup>2</sup>-2·x-1,x)</code>	-1.
--	-----

**nfMax(Uttr1, Var, nedGrense)** ⇒ verdi

<code>nfMax(0.5·x<sup>3</sup>-x-2,x,-5,5)</code>	5.
--	----

**nfMax(Uttr1, Var, nedGrense, øvGrense)** ⇒ verdi

**nfMax(Uttr1, Var) | nedGrense ≤ Var ≤ øvGrense** ⇒ verdi

Returnerer et forslag til numerisk verdi av variabel *Var*, der lokalt maksimum av *Uttr1* forekommer.

Hvis du setter *nedGrens* og *øvGrens*, ser funksjonen i det lukkede intervallet [*nedGrens*,*øvGrens*] etter lokalt maksimum.

**nfMin**(*Uttr1*, *Var*)⇒*verdi*

**nfMin**(*Uttr1*, *Var*, *nedGrense*)⇒*verdi*

**nfMin**(*Uttr1*, *Var*, *nedGrense*,  
*øvGrense*)⇒*verdi*

**nfMin**(*Uttr1*, *Var*) | *nedGrense*≤*Var*  
≤*øvGrense*⇒*verdi*

Returnerer et forslag til numerisk verdi av variabel *Var*, der lokalt minimum av *Uttr1* forekommer.

Hvis du setter *nedGrens* og *øvGrens*, ser funksjonen i det lukkede intervallet [*nedGrens*,*øvGrens*] etter lokalt minimum.

$\text{nfMin}(x^2+2\cdot x+5,x)$	-1.
$\text{nfMin}(0.5\cdot x^3-x-2,x,-5,5)$	-5.

**nInt()**

**nInt**(*Uttr1*, *Var*, *Nedre*, *Øvre*)⇒*uttrykk*

Hvis integranden *Uttr1* ikke inneholder andre verdier enn *Var*, og hvis *Nedre* og *Øvre* er konstanter, positiv ∞, eller negativ ∞, så returnerer **nInt**(*en*) tilnærmet av  $\int(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre})$ . Denne tilnærmede er et veiet gjennomsnitt av noen utvalgsverdier av integranden i intervallen *Nedre*<*Var*<*Øvre*.

Målet er seks signifikante sifre. Den adaptive algoritmen slutter når det er sannsynlig at målet er nådd, eller når det er usannsynlig at ekstra utvalg vil gi nevneverdig forbedring.

Det kommer til syne et varsel ("Tvilsom nøyaktighet") når det ser ut til at målet ikke er nådd.

Nest **nInt**() å utføre multiplenumerisk integrasjon. Integrasjonsgrensene kan avhenge av integrasjonsvariabler utenfor dem.

$\text{nInt}(\cos(x),x,\pi,\pi+1.E-12)$	-1.04144E-12
---	--------------

$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x\cdot y}}{\sqrt{x^2-y^2}},y,-x,x\right),x,0,1\right)$	3.30423
---	---------

**nom**(*effektivRente*, *CpY*) $\Rightarrow$ *verdi*



nom(5.90398,12)

5.75

Finansiell funksjon som omregner den årlige effektive renten *effektivRente* til en nominell rente, gitt *CpY* som antall renteperioder per år.

*effektivRente* må være et reelt tall, og *CpY* må være et reelt tall > 0.

**Merk:** Se også **eff()**, side 46.

**nor** (verken ... eller)  -taster

*BoolskUttr1* **nor** *BoolskUttr2* returnerer *Boolsk uttrykk*

*BoolskListe1* **nor** *BoolskListe2* returnerer *Boolsk liste*

*BoolskMatrise1* **nor** *BoolskMatrise2* returnerer *Boolsk matrise*

Returnerer negasjon av en logisk or-handling på de to argumentene. Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

*Heltall1* **nor** *Heltall2* $\Rightarrow$ *heltall*

Sammenlikner to reelle heltall bit-for-bit med en nor-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis begge bitene er 1; ellers er resultatet 0. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

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}



**norm()****katalog** > **norm**(*Matrise*) $\Rightarrow$ *uttrykk*

$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	5.47723
--	---------

**norm**(*Vektor*) $\Rightarrow$ *uttrykk*

$\text{norm}([1 \ 2])$	2.23607
------------------------	---------

Returnerer Frobenius-normen.

$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	2.23607
--	---------

**normCdf()****Katalog** > 

**normCdf**(*nedreGrense*,*øvreGrense* [,  $\mu$  [,  $\sigma$ ]]) $\Rightarrow$ *tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

Beregner sannsynligheten i normalfordelingen mellom *nedreGrense* og *øvreGrense* for den angitte  $\mu$  (standard=0) og  $\sigma$  (standard=1).

For  $P(X \leq \textit{øvreGrense})$ , sett *nedreGrense* = -9E999.

**normPdf()****Katalog** > 

**normPdf**(*XVerdi* [,  $\mu$  [,  $\sigma$ ]]) $\Rightarrow$ *tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner sannsynlighetstetthet for normalfordelingen ved en spesifisert *XVerdi*-verdi for spesifisert  $\mu$  og  $\sigma$ .

**not (ikke)****Katalog** > **not** *BoolskUttr1* $\Rightarrow$ *Boolsk uttrykk*

$\text{not } (2 \geq 3)$	true
--------------------------	------

Returnerer sann, usann eller en forenklet form av argumentet.

$\text{not } 0\text{hB0} \gg \text{Base16}$	0hFFFFFFFFFFFFFFFF4F
---	----------------------

$\text{not not } 2$	2
---------------------	---

**not** *Heltall1* $\Rightarrow$ *heltall*

I heksades grunntall-modus:

Returnerer tallets komplement av et reelt heltall. Internt er *Heltall1* omregnet til et 64-biters binært tall med fortegn. Verdien av hver bit er forskjøvet (0 blir til 1 og motsatt) for tallets komplement. Resultatene vises i forhold til grunntall-modusen.

**Viktig:** Null, ikke bokstaven O.

$\text{not } 0\text{h7AC36}$	0hFFFFFFFFFFFF853C9
------------------------------	---------------------

I binær grunntall-modus:



Returnerer en matrise av permutasjoner basert på tilsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

## npv()

**npv**(*Rentefot*, *CFO*, *CFListe*[, *CFFrekv*])

Finansiell funksjon som beregner netto nåverdi; summen av nåverdier for kontantstrøm inn og ut. Et positivt resultat for npv forteller at en investering er lønnsom.

*Rentefot* er den renten som trekkes fra kontantstrømmene (pengekostnadene) over en periode.

*CFO* er kontantstrømmen ved start kl. 0; den må være et reelt tall.

*CFListe* er en liste med kontantstrømbeløp etter den innledende kontantstrømmen *CFO*.

*CFFrekv* er en liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (konsekutivt) kontantstrømbeløp, som er det tilsvarende elementet til *CFListe*. Grunninnstilling er 1; hvis du legger inn verdier, må dette være positive heltall < 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() (nLøs)

**nSolve**(*Ligning*, *Var*[=*Forslag*]) ⇒ *tall* eller *feil\_streng*

**nSolve**(*Ligning*, *Var* [=*Forslag*], *nedGrense*) ⇒ *tall* eller *feil\_streng*

**nSolve**(*Ligning*, *Var* [=*Forslag*], *nedGrense*, *øvgrense*) ⇒ *tall* eller *feil\_streng*

**nSolve**(*Ligning*, *Var*[=*Forslag*]) |

$nSolve(x^2 + 5 \cdot x - 25 = 9, x)$	3.84429
$nSolve(x^2 = 4, x = -1)$	-2.
$nSolve(x^2 = 4, x = 1)$	2.

**Merk:** Hvis det foreligger flere løsninger, kan du bruke et forslag som hjelp for å finne en spesiell løsning.

$nedGrense \leq Var \leq \o{v}Grense \Rightarrow tall \text{ eller } feil\_streng$

Søker iterativt etter en tilnærmet reell numerisk løsning for *Ligning* i variabelen. Spesifiser variabelen som:

*variabel*

– eller –

*variabel = reelt tall*

For eksempel er  $x$  gyldig, og det er  $x=3$  også.

**nSolve()** prøver å bestemme enten ett punkt der rest er null, eller to relativt nære punkter, der rest har motsatte fortegn og størrelsen på resten ikke er for stor. Hvis den ikke kan oppnå dette med et lite antall utvalgspunkter, returnerer den strengen "fant ingen løsning."

$nSolve(x^2+5 \cdot x-25=9,x) x<0$	-8.84429
$nSolve\left(\frac{(1+r)^{24}-1}{r}=26,r\right) r>0 \text{ and } r<0.25$	0.006886
$nSolve(x^2=-1,x)$	"No solution found"

## O

### OneVar (EnVar)

**OneVar** [1,]*X*[*Frekv*][*Kategori*,*Inkludert*]]

**OneVar**[*n*,]*X1*,*X2*[*X3*[...[,*X20*]]]

Beregner en-variabel-statistikker med opptil 20 lister. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X*-argumentene er datalister.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hver korresponderende *X*-verdi forekommer. Standardverdien er 1. Alle elementene må være heltall  $\geq 0$ .

*Kategori* er en liste med numeriske kategorikoder for de korresponderende *X*-verdiene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene *X*, *Frekv* eller *Kategori* resulterer i et åpent (tomt) element for det tilsvarende elementet til alle disse listene. Et tomt element i enhver av listene fra *X1* til *X20* resulterer i et åpent (tomt) element for det tilsvarende elementet i alle disse listene. For mer informasjon om tomme elementer, se side 223.

Utdata-variabel	Beskrivelse
stat. $\bar{x}$	Gjennomsnitt av x-verdier
stat. $\Sigma x$	Sum av x-verdier
stat. $\Sigma x^2$	Sum av $x^2$ -verdier
stat.sx	Utvalgets standardavvik av x
stat. x	Populasjonens standardavvik av x
stat.n	Antall datapunkter
stat.MinX	Minimum av x-verdier
stat.Q $_1$ X	Første kvartil av x
stat.MedianX	Median av x
stat.Q $_3$ X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.SSX	Sum av kvadratavvik fra gjennomsnittet av x

**or (eller)**

*BoolskUttr1* **or** *BoolskUttr2* returnerer  
*Boolsk uttrykk*

*BoolskListe1* **or** *BoolskListe2* returnerer  
*Boolsk liste*

*BoolskMatrise1* **or** *BoolskMatrise2*  
returnerer *Boolsk matrise*

Returnerer sann eller usann eller en forenklet form av opprinnelig uttrykk.

Returnerer sann hvis ett eller begge uttrykkene er sanne. Returnerer usann kun hvis begge uttrykkene behandles som usanne.

**Merk:** Se `xor`.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkhåndboken.

### *Heltall1* or *Heltall2* **heltall**

Sammenlikner to reelle heltall bit-for-bit med en or-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis minst en av bitene er 1; resultatet er 0 bare hvis begge bitene er 0. Den returnerte verdien representerer bit-resultatene og vises i grunntall-modus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **Base2**, side 17.

**Merk:** Se `xor`.

```
Define g(x)=Func                                     Done
    If x<0 or x>5
        Goto end
    Return x*3
    Lbl end
EndFunc
```

---

```
g(3)                                                9
g(0)        A function did not return a value
```

I heksades grunntall-modus:

```
0h7AC36 or 0h3D5F                                0h7BD7F
```

**Viktig:** Null, ikke bokstaven O.

I binær grunntall-modus:

```
0b100101 or 0b100                                0b100101
```

**Merk:** Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

**ord()** (num. tegnkode)

Katalog &gt;

**ord**(*Streng*) $\Rightarrow$ *heltall*

ord("hello") 104

**ord**(*Liste l*) $\Rightarrow$ *liste*

char(104) "h"

ord(char(24)) 24

Returnerer numerisk kode for de første tegnene i tegnstreng *Streng*, eller en liste over de første tegnene i hvert listeelement.

ord({"alpha", "beta"}) {97,98}

**P****P>R<sub>x</sub>()**

Katalog &gt;

**P>R<sub>x</sub>**(*rUttr*,  $\theta$ *Uttr*) $\Rightarrow$ *uttrykk*

I Radian-vinkelmodus:

**P>R<sub>x</sub>**(*rListe*,  $\theta$ *Liste*) $\Rightarrow$ *liste***P>R<sub>x</sub>**(4,60°) 2.**P>R<sub>x</sub>**(*rMatrise*,  $\theta$ *Matrise*) $\Rightarrow$ *matrise*

$$\mathbf{P>R_x}\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}$$

$$\{-1.5, 7.07107, 1.3\}$$

Returnerer ekvivalent x-koordinat av (r,  $\theta$ ) paret.

**Merk:**  $\theta$ -argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus. Hvis argumentet er et uttrykk, kan du bruke °, G eller r for å hoppe over vinkelmodus-innstillingen midlertidig.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **P@>R<sub>x</sub>** (...).

**P>R<sub>y</sub>()**

Katalog &gt;

**P>R<sub>y</sub>**(*rVerdi*,  $\theta$ *Verdi*) $\Rightarrow$ *verdi*

I Radian-vinkelmodus:

**P>R<sub>y</sub>**(*rListe*,  $\theta$ *Liste*) $\Rightarrow$ *liste***P>R<sub>y</sub>**(4,60°) 3.4641**P>R<sub>y</sub>**(*rMatrise*,  $\theta$ *Matrise*) $\Rightarrow$ *matrise*

$$\mathbf{P>R_y}\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}$$

$$\{-2.59808, -7.07107, 0\}$$

Returnerer ekvivalent y-koordinat av (r,  $\theta$ )-paret.

**Merk:**  $\theta$ -argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **P@>Ry (...)** .

## PassErr (SendFeil)

### PassErr

For et eksempel på **PassErr**, se eksempel 2 under **Try** -kommandoen, side 166.

Sender en feil til neste nivå.

Hvis systemvariabelen *feilKode* er null, gjør ikke **PassErr** noenting.

**Else** -leddet i **Try...Else...EndTry**-blokken bør bruke **ClrErr** eller **PassErr**. Hvis feilen skal bearbeides eller ignoreres, bruk **ClrErr**. Hvis det ikke er kjent hva som skal gjøres med feilen, bruk **PassErr** for å sende den til den neste feilbehandleren. Hvis det ikke er flere ventende **Try...Else...EndTry** feilbehandlere, vises feil-dialogboksen som normalt.

**Merk:** Se også **ClrErr (SlettFeil)**, side 23, og **Try**, side 166.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

## piecewise() (stykkevis)

**piecewise**(*Uttr1* [, *Betingelse1* [, *Uttr2* [, *Betingelse2* [, ... ]]])

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
--	------

Returnerer definisjoner for en stykkevis definert funksjon i form av en liste. Du kan også opprette stykkevis definisjoner med en sjablon.

$p(1)$	1
$p(-1)$	undef

**Merk:** Se også **stykkevis-sjablon**, side 3.

## poissCdf()

**poissCdf**( $\lambda$ , *nedreGrense*, *øvreGrense*)  $\Rightarrow$  tall  
hvis *nedreGrense* og *øvreGrense* er tall,  
*liste* hvis *nedreGrense* og *øvreGrense* er



lister

**poissCdf**( $\lambda$ , *øvreGrense*)(for  $P(0 \leq X \leq \text{øvreGrense}) \Rightarrow$  tall hvis *øvreGrense* er et tall, liste hvis *øvreGrense* er en liste

Beregner en kumulativ sannsynlighet for den diskrete Poisson-fordeling med spesifisert gjennomsnitt  $\lambda$ .

For  $P(X \leq \text{øvreGrense})$ , sett *nedreGrense*=0

**poissPdf**( $\lambda$ , *XVerd*) $\Rightarrow$  tall hvis *XVerd* er et tall, liste hvis *XVerd* er en liste

Beregner en sannsynlighet for diskret Poisson-fordeling med spesifisert gjennomsnitt  $\lambda$ .

*Vektor* ► **Polar**

$[1 \ 3.]$ ►Polar  $[3.16228 \ \angle 71.5651]$

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive `@>Polar`.

Viser *vektor* i polar form [ $r \angle \theta$ ].  
Vektoren må være av dimensjon 2 og kan være en rad eller en kolonne.

**Merk:** ►Polar er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje, og den oppdaterer ikke svar.

**Merk:** Se også ►Rekt, side 128.

*kompleksVerdi* ► **Polar**

Viser *kompleksVektor* i polar form.

- Grader-vinkelmodus returnerer ( $r \angle \theta$ ).
- Radian-vinkelmodus returnerer  $re^{i\theta}$ .

I Radian-vinkelmodus:

$(3+4i)$ ►Polar  $e^{0.927295 \cdot i \cdot 5}$

---

$(4 \angle \frac{\pi}{3})$ ►Polar  $e^{1.0472 \cdot i \cdot 4}$

I Gradian-vinkelmodus:

**► Polar**

Katalog &gt;

*kompleks Verdi* kan ha en hvilken som helst kompleks form. Men hvis du legger inn  $re^{i\theta}$ , forårsaker dette feil når vinkelmodus er grader.

 $(4-i)$ ►Polar $(4\angle 100.)$ 

**Merk:** Du må bruke parentes for å legge inn polar ( $r\angle \theta$ ).

I Grader-vinkelmodus:

 $(3+4i)$ ►Polar $(5\angle 53.1301)$ **polyEval()**

Katalog &gt;

**polyEval(Liste1, Uttr1)**⇒uttrykk $\text{polyEval}\{ \{1,2,3,4\}, 2 \}$  26**polyEval(Liste1, Liste2)**⇒uttrykk $\text{polyEval}\{ \{1,2,3,4\}, \{2,-7\} \}$  {26, -262}

Tolker det første argumentet som koeffisienter for et polynom med fallende eksponenter, og returnerer en utregnet verdi av polynomet, innsatt verdien av det andre argumentet.

**polyRoots()**

Katalog &gt;

**polyRoots(Poly, Var)** ⇒liste $\text{polyRoots}(y^3+1, y)$  {-1}**polyRoots(KoeffListe)** ⇒liste $\text{cPolyRoots}(y^3+1, y)$   
{-1, 0.5-0.866025i, 0.5+0.866025i}

Den første syntaksen, **polyRoots** (*Poly, Var*), returnerer en liste over sanne røtter av polynom *Poly* med hensyn på variabel *Var*. Hvis det ikke eksisterer noen sanne røtter, returneres en tom liste: {}.

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

Poly må være et polynom i utvidet form i én variabel. Ikke bruk utvidede former, som f.eks.  $y^2\cdot y+1$  or  $x\cdot x+2\cdot x+1$

Den andre syntaksen, **polyRoots** (*KoeffListe*), returnerer en liste over sanne røtter for koeffisientene i *KoeffListe*.

**Merk:** Se også **cPolyRoots()**, side 32.

**PowerReg (PotensReg)**

katalog &gt;

**PowerReg X, Y [, Frekv] [, Kategori,**

*Inkluder*]]

Finner potensregresjonen  $y = (a \cdot (x)^b)$  for listene  $X$  og  $Y$  med frekvensen  $Frekv$ . En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 154).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

$Frekv$  er en valgfri liste med frekvensverdier. Hvert element i  $Frekv$  angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot (x)^b$
stat.a, stat.b	Regresjonskoeffisienter
stat.r <sup>2</sup>	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data ( $\ln(x)$ , $\ln(y)$ )
stat.Resid	Residualene for potensmodellen
stat.ResidTrans	Residualene for den lineære tilpasningen av de transformerte dataene
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>

Utdata-variabel	Beskrivelse
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## Prgm

Katalog > 

### Prgm

#### Blokk

#### EndPrgm

Sjablon for å opprette et egendefinert program. Må brukes med kommandoene **Define**, **Define LibPub** eller **Define LibPriv**.

*Blokk* kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet eller en rekke med utsagn på separate linjer.

*Blokk* kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet, eller en rekke med utsagn på separate linjer.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

Beregn GCD og vis mellomresultater.

---

```
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*

---

## prodSeq()

Se  $\Pi()$ , side 197.

## Produkt (PI)

Se  $\Pi()$ , side 197.

## Product()

Katalog > 

**product**(Liste[, Start[, slutt]])⇒uttrykk

Returnerer produktet av elementene i *Liste*. *Start* og *Slutt* er valgfrie. De spesifiserer et elementområde.

---

```
product({1,2,3,4})
```

---

```
product({4,5,8,9},2,3)
```

---

24

40

**Product()**Katalog > 

**product**(*Matrise*1[, *Start*[, *slutt*]]) $\Rightarrow$ *matrise*

Returnerer en radvektor som inneholder produktet av elementene i kolonnene i *Matrise*1. *Start* og *slutt* er alternativer. De spesifiserer et radområde.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

product	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$[28 \ 80 \ 162]$
product	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 1, 2$	$[4 \ 10 \ 18]$

**propFrac() (ekteBrøk)**Katalog > 

**propFrac**(*Verdi*1[, *Var*]) $\Rightarrow$ *verdi*

**propFrac**(*rasjonal\_tall*) returnerer *rasjonalt\_tall* som summen av et heltall og en brøk som har samme fortegn og større nevner enn teller.

**propFrac**(*rasjonalt\_uttrykk*, *Var*) returnerer summen av ekte brøk og et polynom med hensyn på *Var*. Gradene til *Var* i nevneren er større enn gradene til *Var* i telleren i hver ekte brøk. Liknende potenser av *Var* er samlet sammen. Leddene og faktorene deres er sortert med *Var* som hovedvariabel.

Hvis *Var* utelates, utvides den ekte brøken med hensyn på de fleste hovedvariablene. Koeffisientene til den polynomiske delen omgjøres så til "ekte" med hensyn på de fleste hovedvariablene og så videre.

propFrac	$\left(\frac{4}{3}\right)$	$1 + \frac{1}{3}$
propFrac	$\left(\frac{-4}{3}\right)$	$-1 - \frac{1}{3}$

**Q****QR**Katalog > 

**QR** *Matrise*, *qMatNavn*, *rMatNavn*(, *To*1]



Beregner den faste QR faktoriseringen av en reell eller en kompleks matrise. De resulterende matrisene Q og R lagres til det spesifiserte *MatNavn*. Q-matrisen er enhetlig. R-matrisen er øvre trekantet.

Det flytende desimalpunktallet (9.) i m1 gjør at resultatene må beregnes i flytende desimalpunkt-form.

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

QR	<i>m1,qm,rm</i>	<i>Done</i>
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$	
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$	

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \text{maks}(\text{dim}(\text{Matrise})) \cdot \text{radNorm}(\text{Matrise})$

Faktoriseringen QR beregnes numerisk med faste transformasjoner. Den symbolske løsningen beregnes med Gram-Schmidt. Kolonnene i *qMatNavn* er ortonormale grunnvektorer som utspenner rommet som defineres av *matrise*.

## QuadReg (KvadReg)

**QuadReg** *X,Y* [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finner den kvadratiske polynomiske regresjonen  $y = a \cdot x^2 + b \cdot x + c$  for listene *X* og *Y* med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene.

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regresjonskoeffisienter
stat.R <sup>2</sup>	Determinasjonskoeffisient
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## QuartReg

**QuartReg**  $X, Y$  [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finner den fjerdegrads polynomiske regresjonen

$y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$  for listene  $X$  og  $Y$  med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

$X$  og  $Y$  er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt  $X$  og  $Y$  forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regresjonskoeffisienter
stat.R <sup>2</sup>	Determinasjonskoeffisient
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## R

### R ► Pθ()

I Grader-vinkelmodus:

R ► Pθ ( $xValue$ ,  $yValue$ ) ⇒ verdi



**R ▶ Pθ()**

Katalog &gt;

**R ▶ Pθ** (*xList*, *yList*) ⇒ *liste***R ▶ Pθ** (*xMatrix*, *yMatrix*) ⇒ *matrise*Returnerer tilsvarende θ-koordinat for (*x*, *y*) par-argumentene.**Merk:** Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av gjeldende vinkelmodus-innstilling.**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn **R@>Ptheta** (...).

---

**R ▶ Pθ(2,2)** 45.

---

I Gradian-vinkelmodus:

---

**R ▶ Pθ(2,2)** 50.

---

I Radian-vinkelmodus:

---

**R ▶ Pθ(3,2)** 0.588003

---

---

**R ▶ Pθ**  $\left( \begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix} \right)$ 

---

 $\begin{bmatrix} 0. & 2.94771 & 0.643501 \end{bmatrix}$ 

---

**R ▶ Pr()**

Katalog &gt;

**R ▶ Pr** (*xValue*, *yValue*) ⇒ *verdi***R ▶ Pr** (*xList*, *yList*) ⇒ *liste***R ▶ Pr** (*xMatrix*, *yMatrix*) ⇒ *matrise*Returnerer tilsvarende r-koordinat for (*x*, *y*) parargumentene.**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn **R@>Pr** (...).

I Radian-vinkelmodus:

---

**R ▶ Pr(3,2)** 3.60555

---

---

**R ▶ Pr**  $\left( \begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix} \right)$ 

---

 $\begin{bmatrix} 3 & 4.07638 & \frac{5}{2} \end{bmatrix}$ 

---

**▶ Rad**

Katalog &gt;

*Value1* ▶ **Rad** ⇒ *verdi*

Omformer argumentet til radian vinkelmåling.

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn **R@>Ptheta** (...).

I Grader-vinkelmodus:

---

**(1.5) ▶ Rad** (0.02618)<sup>r</sup>

---

I Gradian-vinkelmodus:

---

**(1.5) ▶ Rad** (0.023562)<sup>r</sup>

---

**tilf()**

Katalog &gt;

**rand()** ⇒ *uttrykk***rand(#Trials)** ⇒ *liste*

Angir det tilfeldige tallet for start.

**tilf()**Katalog > 

**rand()** returnerer en tilfeldig verdi mellom 0 og 1.

RandSeed 1147	Done
rand(2)	{0.158206,0.717917}

**rand(#Trials)** returnerer en liste som inneholder #Trials tilfeldige verdier mellom 0 og 1.

**tilfBin()**Katalog > 

**randBin(*n*, *p*)** ⇒ *uttrykk*  
**randBin(*n*, *p*, #Trials)** ⇒ *liste*

randBin(80,0,5)	46.
randBin(80,0,5,3)	{43.,39.,41.}

**randBin(*n*, *p*)** returnerer et tilfeldig reelt tall fra en spesifisert binomisk fordeling.

**randBin(*n*, *p*, #Trials)** returnerer en liste som inneholder #Trials tilfeldige reelle tall fra en spesifisert binomisk fordeling.

**tilfInt()**Katalog > 

**randInt**  
 (*lowBound*,*upBound*)  
 ⇒ *uttrykk*  
**randInt**  
 (*lowBound*,*upBound*,  
 #Trials) ⇒ *liste*

randInt(3,10)	3.
randInt(3,10,4)	{9.,3.,4.,7.}

**randInt**  
 (*lowBound*,*upBound*)  
 returnerer et tilfeldig heltall innen området som er spesifisert av heltallsgrensene *lowBound* og *upBound*.

**randInt**  
 (*lowBound*,*upBound*,  
 #Trials) returnerer en liste som inneholder #Trials tilfeldige heltall innen det spesifiserte området.

## randMat()

Katalog > 

**randMat**(numRows, numColumns) ⇒  
matrise

Returnerer en matrise av heltall mellom -9 og 9 av spesifisert dimensjon.

Begge argumentene må forenkles til heltall.

RandSeed 1147	Done
randMat(3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

**Merk:** Verdiene i denne matrisen endres hver gang du trykker på **enter**.

## randNorm()

Katalog > 

**randNorm**( $\mu$ ,  $\sigma$ ) ⇒ uttrykk  
**randNorm**( $\mu$ ,  $\sigma$ , #Trials) ⇒ liste

**randNorm**( $\mu$ ,  $\sigma$ ) returnerer et desimaltall fra den spesifiserte normalfordelingen. Det kan være et hvilket som helst tall, men vil være sterkt konsentrert i intervallet [ $\mu-3\sigma$ ,  $\mu+3\sigma$ ].

**randNorm**( $\mu$ ,  $\sigma$ , #Trials) returnerer en liste som inneholder #Trials desimaltall fra den spesifiserte normalfordelingen.

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4.5)	-3.54356

## randPoly()

Katalog > 

**randPoly**(Var, Order) ⇒ uttrykk

Returnerer et polynom i Var av den spesifiserte Order. Koeffisientene er tilfeldige heltall i området -9 til 9. Den ledende koeffisienten blir ikke null.

Order må være 0-99.

RandSeed 1147	Done
randPoly(x,5)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

## tilfUtv()

Katalog > 

**randSamp**(List, #Trials[,noRepl]) ⇒  
liste

Returnerer en liste som inneholder et tilfeldig utvalg av #Trials forsøk fra List med mulighet for bytte av utvalg (noRepl=0), eller ingen bytte av utvalg (noRepl=1). Grunninnstillingen er med bytte av utvalg.

Define list3={1,2,3,4,5}	Done
Define list4=randSamp(list3,6)	Done
list4	{1.,3.,3.,1.,3.,1.}

**RandSeed**Katalog > **RandSeed Tall**

Hvis *Tall* = 0, settes startverdien for tilfeldig tall-generatoren til fabrikkinnstilling. Hvis *Tall* ≠ 0, brukes det for å opprette to startverdier, som lagres i systemvariablene *seed1* og *seed2*.

RandSeed 1147	Done
rand()	0.158206

**real()**Katalog > **real(Value1) ⇒ verdi**

Returnerer den reelle delen av argumentet.

real(2+3·i)	2
-------------	---

**real(List1) ⇒ liste**

Returnerer den reelle delen av alle elementer.

real({1+3·i,3,i})	{1,3,0}
-------------------	---------

**real(Matrix1) ⇒ matrise**

Returnerer den reelle delen av alle elementer.

real( $\begin{pmatrix} 1+3 \cdot i & 3 \\ 2 & i \end{pmatrix}$ )	$\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix}$
--	--

**► Rect**Katalog > *Vector* ► **Rect**

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn @>**Rect**.

$\left( \begin{matrix} 3 & \angle & \frac{\pi}{4} & \angle & \frac{\pi}{6} \end{matrix} \right)$ ► Rect	[1.06066 1.06066 2.59808]
---	---------------------------

Viser *Vector* i rektangulær form [x, y, z]. Vektoren må være av dimensjon 2 eller 3 og kan være en rad eller en kolonne.

**Merk:** ► **Rect** er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje, og den oppdaterer ikke *ans*.

**Merk:** Se også ► **Polar**, side 117.

*complexValue* ► **Rect**

I Radian-vinkelmodus:

Viser *complexValue* i rektangulær form  $a+bi$ . *complexValue* kan ha hvilken som helst kompleks form. Men hvis du legger inn  $re^{i\theta}$ , forårsaker dette feil når vinkelmodus er grader.

**Merk:** Du må bruke parentes for å legge inn polar ( $r \angle \theta$ ).

$$\left(4 \cdot e^{\frac{\pi}{3}}\right) \blacktriangleright \text{Rect} \quad 11.3986$$

$$\left(\left(4 \angle \frac{\pi}{3}\right)\right) \blacktriangleright \text{Rect} \quad 2.+3.4641 \cdot i$$

I Gradian-vinkelmodus:

$$\left(\left(1 \angle 100\right)\right) \blacktriangleright \text{Rect} \quad i$$

I Grader-vinkelmodus:

$$\left(\left(4 \angle 60\right)\right) \blacktriangleright \text{Rect} \quad 2.+3.4641 \cdot i$$



**Merk:** For å skrive  $\angle$ , velg den fra symbollisten i Katalogen.

## ref()

**ref(Matrix1[, Tol])**  $\Rightarrow$  matrise

Returnerer eliminasjonsformen av *Matrix1*.

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Unngå udefinerte elementer i *Matrix1*. De kan føre til uventede resultater.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

Hvis for eksempel  $a$  er udefinert i følgende uttrykk, vises en varselmelding, og resultatet vises som:

$$\text{ref} \left( \begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Varselet vises fordi det generaliserte elementet  $1/a$  ikke ville være gyldig for  $a=0$ .

Dette kan du unngå ved å lagre en verdi til  $a$  på forhånd eller ved å bruke begrensningen (" $|$ "), som vist i følgende eksempel.

$$\text{ref} \left( \begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mid a=0 \right) \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

**Merk:** Se også `rref()`, page 139.

## RefreshProbeVars

### RefreshProbeVars

Gir deg tilgang til sensordata fra alle tilkoblede sensorer gjennom TI-Basic-programmet.

#### StatusVar-verdi

`statusVar`  
=0

**Status**  
Normal (fortsett med programmet)  
Vernier DataQuest™-applikasjonen er i datainnsamlingsmodus.

`statusVar`  
=1

**Merk:** Vernier DataQuest™-applikasjonen må være i målermodus for at denne kommandoen skal fungere.



### Eksempel

```
Definer temp()=
Prgm
© Sjekk om systemet er klart
RefreshProbeVars-status
Hvis status=0, så
Disp «klar»
For n,1,50
RefreshProbeVars-status
temperatur:=meter.temperatur
Disp «temperatur:
»,temperature
Hvis temperatur>30 så
```

StatusVar-verdi	Status
<i>statusVar</i> =2	Vernier DataQuest™-applikasjonen er ikke startet.
<i>statusVar</i> =3	Vernier DataQuest™-applikasjonen er startet, men du har ikke koblet til noen sensorer.

```
Disp «for varm»
EndIf
© Vent i 1 sekund mellom
utvalgene
Vent 1
EndFor
Else
Disp «ikke klar. Prøv igjen
senere»
EndIf
EndPrgm
```

Merk: Dette kan også brukes med TI-Innovator™-senter.

## remain()

**remain**(*Value1*, *Value2*) ⇒ *verdi*  
**remain**(*List1*, *List2*) ⇒ *liste*  
**remain**(*Matrix1*, *Matrix2*) ⇒ *matrise*

Returnerer resten av det første argumentet med hensyn på det andre argumentet som definert av identitetene:

$\text{remain}(x,0) = x$   
 $\text{remain}(x,y) = x - y \cdot \text{iPart}(x/y)$

Som en konsekvens, merk at **remain**( $-x,y$ ) = **remain**( $x,y$ ). Resultatet er enten null eller det har samme fortegn som det første argumentet.

**Merk:** Se også **mod()**, side 100.

$\text{remain}(7,0)$	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 }

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

## Forespør

**Forespør** *promptString*, *var*[, *DispFlag* [, *statusVar*]]

Definere et program:

**Forespør** *promptString, func(arg1, ...argn) [, DispFlag [, statusVar]]*

Programmeringskommando: Stopper programmet og viser en dialogboks med meldingen *promptString* og en inndata-boks for brukerens svar.

Når brukeren skriver inn et svar og klikker på **OK**, blir innholdet i inndata-boksen tildelt til variabel *var*.

Hvis brukeren klikker på **Avbryt**, forsetter programmet uten å godta inndata. Programmet bruker forrige verdi for *var* hvis *var* allerede var definert.

Det valgfrie argumentet *DispFlag* kan være et hvilket som helst uttrykk.

- Hvis *DispFlag* utelates eller behandles til **1**, blir prompt-meldingen og brukerens svar vist i Kalkulator-loggen.
- Hvis *DispFlag* behandles til **0**, vises ikke prompt-meldingen eller svaret i loggen.

Det valgfrie *statusVar*-argumentet gir programmet en måte å bestemme hvordan brukeren avviste dialogboksen. Merk at *statusVar* krever *DispFlag*-argumentet.

- Hvis brukeren klikket på **OK** eller trykket **Enter** eller **Ctrl+Enter**, settes variabelen *statusVar* til en verdi på **1**.
- Ellers innstilles variabelen *statusVar* til en verdi på **0**.

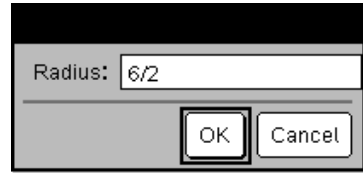
Argumentet *func()* lar et program lagre brukerens svar som en funksjonsdefinisjon. Denne syntaksen arbeider som om brukeren utførte kommandoen:

Definer *func(arg1, ...argn) = brukerens svar*

```
Definer forespør_demo()=Prgm
  Forespør "Radius: ",r
  Vis "Area = ",pi*r^2
EndPrgm
```

Kjør programmet og skriv inn et svar:

forespør\_demo()



Resultat etter trykk på **OK**:

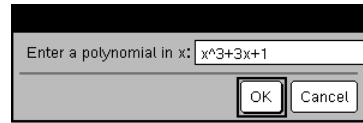
```
Stråle: 6/2
Area= 28,2743
```

Definere et program:

```
Definer polynom()=Prgm
  Forespør "Legg inn et polynom i
  x: ",p(x)
  Vis "Reelle røtter er:",polyRøtter
  (p(x),x)
EndPrgm
```

Kjør programmet og skriv inn et svar:

polynom()



Resultat etter å ha skrevet inn  $x^3+3x+1$  og valgt **OK**:



```
Reelle røtter er: {-0,322185}
```



Programmet kan så bruke den definerte funksjonen *funk()*. *promptString* skal veilede brukeren i å legge inn et passende *bruker-svar* som fullfører funksjonsdefinisjonen.

**Merk:** Kalkulatoren *Forespør* - kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Slik stopper du et program som inneholder en **Request**-kommando inne i en infinitt løkke:

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

**Merk:** Se også **ForespørStr**, page 133.

## ForespørStr

**ForespørStr** *promptString*, *var*[, *DispFlag*]

Programmeringskommando: Arbeider identisk med den første syntaksen i **Request**-kommandoen, bortsett fra at brukerens svar alltid tolkes som en streng. Som kontrast tolker **Request**-kommandoen svaret som et uttrykk hvis ikke brukeren setter det i anførselstegn ("").

**Merk:** Du kan bruke **RequestStr** - kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Slik stopper du et program som inneholder en **ForespørStr**-kommando i en infinitt løkke:



Definere et program:

```
Definer forespørStr_demo()=Prgm
  ForespørStr "Navnet ditt:",navn,0
  Vis "Forespør har ",dim(navn),"
  tegn."
EndPrgm
```

Kjør programmet og skriv inn et svar:

```
forespørStr_demo()
```



- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

Resultat etter klikk på **OK** (Merk at argumentet *DispFlag* på **0** utelater prompten og svaret fra loggen):

```
forespørStr_demo()
```

Svaret har 5 tegn.

**Merk:** Se også **Forespør**, page 131.

## Return

### Return [*Expr*]

Returnerer *Expr* som resultatet av funksjonen. Bruk i en **Func...EndFunc**-blokk.

**Merk:** Bruk **Return** uten et argument i en **Prgm...EndPrgm**-blokk for å avslutte et program.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer·counter → answer
EndFor
Return answer|
EndFunc
```

```
factorial (3)
```

6

## right()

### right(*List1* [, *Num*]) ⇒ *liste*

Returnerer *Num*-elementene som ligger lengst til høyre i *List1*.

Hvis du utelater *Num*, returneres alle elementer i *List1*.

### right(*sourceString* [, *Num*]) ⇒ *streng*

Returnerer *Num*-tegnene som ligger lengst til høyre i tegnstring *sourceString*.

Hvis du utelater *Num*, returneres alle elementer i *sourceString*.

```
right({ 1,3,-2,4 },3)           { 3,-2,4 }
```

```
right("Hello",2)              "lo"
```

**right(Comparison)** ⇒ uttrykk

Returnerer høyre side av en ligning eller ulikhet.

**rk23 ()**

**rk23(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, diftol])** ⇒ matrise

**rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])** ⇒ matrise

**rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])** ⇒ matrise

Bruker Runge-Kutta-metoden for å løse 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  $[\text{Var}0, \text{VarMax}]$ . Returnerer en matrise, hvis første rad definerer resultatverdiene av  $\text{Var}$  som definert av  $\text{VarStep}$ . Den andre raden definerer verdien av den første løsningskomponenten for de tilsvarende  $\text{Var}$ -verdiene, og så videre.

$\text{Expr}$  er høyre side, som definerer den ordinære differensialligningen (ODE).

$\text{SystemOfExpr}$  er et system på høyre side som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i  $\text{ListOfDepVars}$ ).

$\text{ListOfExpr}$  er en liste over høyresider som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i  $\text{ListOfDepVars}$ ).

$\text{Var}$  er den uavhengige variabelen.

$\text{ListOfDepVars}$  er en liste over avhengige variabler.

Differensialligning:

$$y' = 0,001 * y * (100 - y) \text{ og } y(0) = 10$$

$$\text{rk23}\left(0,001 \cdot y \cdot (100 - y), t, y, \left\{0, 100\right\}, 10, 1\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

Samme ligning med  $\text{diftol}$  satt til  $1.E-6$

$$\text{rk23}\left(0,001 \cdot y \cdot (100 - y), t, y, \left\{0, 100\right\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

System av ligninger:

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

med  $y1(0) = 2$  og  $y2(0) = 5$

$$\text{rk23}\left(\begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245



Rotere bitene i et binært heltall. Du kan angi *Integer1* i enhver basis. Den konverteres automatisk til en 64-biters binærform med fortegn. Hvis *Integer1* er for stort for denne formen, brukes en symmetrisk modulusoperasjon til å konvertere tallet inn i gyldig verdiområde. (For mer informasjon, se ► **Base2**, side 17.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er  $-1$  (roteres én bit mot høyre).

For eksempel i en høyre-rotasjon:

Hver bit roterer mot høyre.

0b0000000000000111101011000110101

Bit helt til høyre roterer mot venstre.

produserer:

0b100000000000011110101100011010

Resultatene vises i forhold til grunntall-modusen.

**rotate**(*List1*[,*#ofRotations*]) ⇒ *liste*

Returnerer en kopi av *List1* som er rotert mot høyre eller mot venstre av *#ofRotations*-elementer. Endrer ikke *List1*.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er  $-1$  (roterer ett element mot høyre).

**rotate**(*String1*[,*#ofRotations*]) ⇒ *streng*

Returnerer en kopi av *String1* som er rotert mot høyre eller mot venstre av *#ofRotations*-tegnene. Endrer ikke *String1*.

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

I heksadesimal grunntall-modus:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h8000000000001E3
rotate(0h78E,2)	0h1E38

**Viktig:** Hvis du vil skrive inn et binært eller heksadesimalt tall, må du alltid bruke prefikset 0b eller 0h (null, ikke bokstaven O).

I desimalgrunntall-modus:

rotate({1,2,3,4})	{4,1,2,3}
rotate({1,2,3,4},-2)	{3,4,1,2}
rotate({1,2,3,4},1)	{2,3,4,1}

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

## rotate()

Katalog > 

Hvis  $\#ofRotations$  er positiv, skjer rotasjonen til venstre. Hvis  $\#ofRotations$  er negativ, skjer rotasjonen til høyre. Grunninnstilling er  $-1$  (roterer ett tegn mot høyre).

## round()

Katalog > 

**round**(*Value1* [, *digits*])  $\Rightarrow$  verdi

$\text{round}(1.234567,3)$	1.235
----------------------------	-------

Returnerer argumentet avrundet til spesifisert antall sifre etter desimalpunktet.

*digits* må være et heltall i området 1 – 12. Hvis *digits* ikke er inkludert, returneres argumentet avrundet til 12 signifikante sifre.

**Merk:** Visning av siffermodus kan ha innvirkning på hvordan dette vises.

**round**(*List1* [, *digits*])  $\Rightarrow$  liste

$\text{round}(\{\{\pi, \sqrt{2}, \ln(2)\}, 4\})$	$\{3.1416, 1.4142, 0.6931\}$
--	------------------------------

Returnerer en liste over elementer avrundet til spesifisert antall sifre.

**round**(*Matrix1* [, *digits*])  $\Rightarrow$  matrise

$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1\right)$	$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$
--	--

Returnerer en matrise over elementer avrundet til spesifisert antall sifre.

## rowAdd()

Katalog > 

**rowAdd**(*Matrix1*, *rIndex1*, *rIndex2*)  $\Rightarrow$  matrise

$\text{rowAdd}\left(\begin{bmatrix} 3 & 4 \\ -3 & -2 \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$
--	--

Returnerer en kopi av *Matrix1* med rad *rIndex2* erstattet med summen av rader *rIndex1* og *rIndex2*.

## rowDim()

Katalog > 

**rowDim**(*Matrix*)  $\Rightarrow$  uttrykk

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
--	---

Returnerer antallet rader i *Matrix*.

**Merk:** Se også **colDim()**, side 24.

$\text{rowDim}(m1)$	3
---------------------	---

## rowNorm()

Katalog > 

**rowNorm**(*Matrix*) ⇒ *uttrykk*

Returnerer den største summen av absoluttverdiene for elementene i radene i *Matrix*.

**Merk:** Alle matriseelementene må forenkles til tall. Se også **colNorm()**, side 24.

$$\text{rowNorm} \left( \begin{array}{ccc} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{array} \right) = 25$$

## rowSwap()

Katalog > 

**rowSwap**(*Matrix1*, *rIndex1*, *rIndex2*)  
⇒ *matrise*

Returnerer *Matrix1* med rader *rIndex1* og *rIndex2* ombyttet.

$$\begin{array}{ccc} \begin{array}{cc} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{array} & \rightarrow \text{mat} & \begin{array}{cc} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{array} \\ \text{rowSwap}(\text{mat}, 1, 3) & & \begin{array}{cc} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{array} \end{array}$$

## rref()



Katalog > 

**rref**(*Matrix1*[, *Tol*]) ⇒ *matrise*

Returnerer den reduserte eliminasjonsformen av *Matrix1*.

$$\text{rref} \left( \begin{array}{cccc} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{array} \right) = \begin{array}{ccc} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{array}$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

**Merk:** Se også [ref\(\)](#), page 129.

## S

## sec()

 tast

$\text{sec}(\text{Verdi1}) \Rightarrow \text{verdi}$

I Grader-vinkelmodus:

$\text{sec}(\text{Liste1}) \Rightarrow \text{liste}$

$\text{sec}(45)$	1.41421
------------------	---------

Returnerer sekans til *Verdi1* eller returnerer en liste med sekans til hvert element i *Liste1*.

$\text{sec}(\{1,2,3,4\})$	$\{1.00015, 1.00081, 1.00244\}$
---------------------------	---------------------------------

**Merk:** Argumentet tolkes som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °,  $\text{G}$  eller  $\text{r}$  for å hoppe over vinkelmodusen midlertidig.

 $\text{sec}^{-1}()$  tast

$\text{sec}^{-1}(\text{Verdi1}) \Rightarrow \text{verdi}$

I Grader-vinkelmodus:

$\text{sec}^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

$\text{sec}^{-1}(1)$	0
----------------------	---

Returnerer vinkelen som har sekans lik *Verdi1* eller returnerer en liste med inverse sekanser til hvert element i *Liste1*.

$\text{sec}^{-1}(1)$	0.
----------------------	----

**Merk:** Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

I Gradian-vinkelmodus:

$\text{sec}^{-1}(\sqrt{2})$	50
-----------------------------	----

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive  $\text{arcsec}(\dots)$ .

I Radian-vinkelmodus:

$\text{sec}^{-1}(\{1,2,5\})$	$\{0, 1.0472, 1.36944\}$
------------------------------	--------------------------

## sech()

Katalog > 

$\text{sech}(\text{Verdi1}) \Rightarrow \text{verdi}$

$\text{sech}(3)$	0.099328
------------------	----------

$\text{sech}(\text{Liste1}) \Rightarrow \text{liste}$

$\text{sech}(\{1,2,3,4\})$	$\{0.648054, 0.198522, 0.036619\}$
----------------------------	------------------------------------



Returnerer hyperbolsk sekans av *Verdi1* eller returnerer en liste med hyperbolske sekanser av *Liste1*-elementene.

sech<sup>-1</sup>()

sech<sup>-1</sup>(*Verdi1*) ⇒ *verdi*

sech<sup>-1</sup>(*Liste1*) ⇒ *liste*

Returnerer invers hyperbolsk sekans til *Verdi1* eller returnerer en liste med inverse hyperbolske sekanser til hvert element i *Liste1*.

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arcsech** (...).

I Radian-vinkelmodus og Rectangular-kompleksmodus:

sech <sup>-1</sup> (1)	0
sech <sup>-1</sup> ({1, -2, 2.1})	{0, 2.0944·i, 8.ε-15+1.07448·i}

## Send

Send *uttrEllerStreng1* [, *uttrEllerStreng2*] ...

Programmeringskommando: Sender én eller flere TI-Innovator™ Hub kommandoer til en tilkoblet hub.

*uttrEllerStreng* må være en gyldig TI-Innovator™ Hub kommando. Vanligvis inneholder *uttrEllerStreng* en "INNSTILL ..." -kommando for å kontrollere en enhet eller en «LES ...»-kommando for å etterspørre data.

Argumentene sendes suksessivt til hubben.

**Merk:** Du kan bruke **Send**-kommandoen i et brukerdefinert program, men ikke i en funksjon.

**Merk:** Se også **Get** (side 62), **GetStr** (side 69), og **eval()** (side 50).

Eksempel: Slå på det blå elementet i den innebygde RGB LED-skjermen i 0,5 sekunder.

Send "SET COLOR.BLUE ON TIME .5"	Done
----------------------------------	------

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. En **Get**-kommando henter verdien og tildeler den til variabelen *lysver*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Eksempel: Send en kalkulert frekvens til hubbens innebygde høyttaler. Bruk den spesielle variabelen *iostr.SendAns* for å vise hubkkommandoen med uttrykket som er evaluert.

$n:=50$	50
$m:=4$	4
Send "SET SOUND eval(m · n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

**seq()** (sekv)

Katalog &gt;

**seq**(*Uttr*, *Var*, *Lav*, *Høy*, (*Trinn*)) ⇒ *liste*

Øker *Var* fra *Lav* til *Høy* med trinn på *Intervall*, behandler *Uttr* og returnerer resultatene som en liste. Det opprinnelige innholdet i *Var* er fremdeles der etter at **seq()** er fullført.

Grunnverdien for *Intervall* = 1.

$\text{seq}(n^2, n, 1, 6)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

**Merk:** For å tvinge frem et tilnærmet desimalresultat,

**Grafregner:** Trykk på .

**Windows®:** Trykk på **Ctrl+Enter**.

**Macintosh®:** Trykk på **⌘+Enter**.

**iPad®:** Hold på **enter**, og velg .

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

**seqGen()**

Katalog &gt;

**seqGen**(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}, {*ListeAvInnlLedd* [, *VarIntervall* [, *Loftverdi*]]]) ⇒ *liste*

Genererer en liste over ledd for tallfølge *avhVar*(*Var*)=*Uttr* som følger: Øker uavhengig variabel *Var* fra *Var0* til *VarMaks* med *VarIntervaller*, beregner *avhVar*(*Var*) for tilsvarende verdier av *Var* ved hjelp av *Uttr*-formel og *ListeAvInnlLedd*, og returnerer resultatene som en liste.

**seqGen**(*ListeEllerSystemAvUttr*, *Var*, {*ListeMedAvhVarer*, {*Var0*, *VarMaks*} [, *MatriseAvInnlLedd* [, *VarIntervall* [, *Loftverdi*]]]) ⇒ *matrise*

Genererer de første 5 leddene i tallfølgen  $u(n) = u(n-1)^2/2$ , med  $u(1)=2$  og *VarIntervall*=1.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$	$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$
---	---

Eksempel der *Var0*=2:

Genererer en matrise av ledd for et system (eller en liste) av tallfølger

*ListeMedAvhVarer*

(*Var*)=*ListeEllerSystemAvUttr* som følger: Øker uavhengig variabel *Var* fra *Var0* til *VarMaks* med *VarIntervall*, behandler *ListeMedAvhVarer(Var)* for tilsvarende verdier av *Var* ved hjelp av *ListeEllerSystemAvUttr*-formel og *MatriseAvInnlLedd*, og returnerer resultatene som en matrise.

Opprinnelig innhold i *Var* er uendret etter at **seqGen()** er fullført.

Grunnverdien for *VarIntervall* = 1.

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2,5\}, \{3\}\right)$$

$$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

System av to tallfølger:

$$\text{seqGen}\left(\left[\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right], n, \{u1, u2\}, \{1,5\}, \left[\begin{array}{c} - \\ 2 \end{array}\right]\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Merk: Tomrommet ( ) i matrisen med innledende ledd over brukes for å angi at det innledende leddet for  $u1(n)$  er beregnet ved hjelp av den eksplisitte tallfølgeformelen  $u1(n)=1/n$ .

## seqn()

**seqn**(*Uttr(u, n [, ListeMedInnlLedd[, nMaks [, Loftverdi]]])*) ⇒ *liste*

Genererer en liste over ledd for tallfølge  $u(n)=\text{Uttr}(u, n)$  som følger: Øker  $n$  fra 1 til  $nMaks$  med 1, beregner  $u(n)$  for tilsvarende verdier av  $n$  ved hjelp av formel  $\text{Uttr}(u, n)$  og *ListeMedInnlLedd*, og returnerer resultatene som en liste.

**seqn**(*uttr(n [, nMaks [, Loftverdi]]*) ⇒ *liste*

Genererer en liste over ledd for en ikke-rekursiv tallfølge  $u(n)=\text{Uttr}(n)$  som følger: Øker  $n$  fra 1 til  $nMaks$  med 1, beregner  $u(n)$  for tilsvarende verdier av  $n$  ved hjelp av formelen  $\text{Uttr}(n)$ , og returnerer resultatene som en liste.

Hvis  $nMaks$  mangler, innstilles  $nMaks$  på 2500

Hvis  $nMax=0$ , innstilles  $nMaks$  på 2500

Genererer de første 6 leddene i tallfølgen  $u(n) = u(n-1)/2$ , med  $u(1)=2$ .

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

**Merk:** seqn() kaller seqGen( ) med  $n0=1$  og  $nintervall=1$

## SetMode() (lesModus)

**SetMode(modusNavnHeltall, innstillingHeltall) ⇒ heltall**

**SetMode(liste) ⇒ heltallsliste**

Kun gyldig innenfor en funksjon eller et program.

**SetMode(modusNavnHeltall, innstillingHeltall)** setter foreløpig modus *modusNavnHeltall* til den nye innstillingen *innstillingHeltall*, og returnerer et heltall som samsvarer med den opprinnelige innstillingen av den modusen. Endringen er begrenset til hvor lenge det varer å utføre programmet/funksjonen.

*modusNavnHeltall* spesifiserer hvilken modus du vil stille inn. Det må være en av modus-heltallene fra tabellen nedenfor.

*innstilleHeltall* spesifiserer den nye innstillingen for modusen. Det må være en av innstillingsheltallene fra listen nedenfor for den spesifikke modusen som du stiller inn.

**SetMode(liste)** lar deg endre flere innstillinger. *liste* inneholder tallpar med modusheltall og innstillingeheltall.

**SetMode(liste)** returnerer en liknende liste med heltallpar som representerer de opprinnelige modusene og innstillingene.

Hvis du har lagret alle modusinnstillinger med **SetMode(0) → var**, kan du bruke **SetMode(var)** for å gjenopprette disse innstillingene til funksjonen eller programmet lukkes. Se **SetMode()**, side 68.

Vis tilnærmet verdi av  $\pi$  ved hjelp av grunninnstillingen for Vis sifre, og vis så  $\pi$  med en innstilling på Fast2. Kontroller for å se at grunninnstillingen gjenoprettes etter at programmet utføres.

Define <i>prog1()</i> =Prgm	Done
Disp $\pi$	
setMode(1,16)	
Disp $\pi$	
EndPrgm	
<hr/>	
<i>prog1()</i>	
	3.14159
	3.14
	Done

**Merk:** De aktuelle modusinnstillingene sendes til påkalte underrutiner. Hvis en underrutine endrer en modusinnstilling, går modusinnstillingen tapt når kontrollen går tilbake til påkallingsrutinen.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkhåndboken.

Modus Navn	Modus Heltall	Innstille heltall
Vis sifre	1	1=Flytende, 2=Flytende1, 3=Flytende2, 4=Flytende3, 5=Flytende4, 6=Flytende5, 7=Flytende6, 8=Flytende7, 9=Flytende8, 10=Flytende9, 11=Flytende10, 12=Flytende11, 13=Flytende12, 14=Fast0, 15=Fast1, 16=Fast2, 17=Fast3, 18=Fast4, 19=Fast5, 20=Fast6, 21=Fast7, 22=Fast8, 23=Fast9, 24=Fast10, 25=Fast11, 26=Fast12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponensielt format	3	1=Normal, 2=Vitenskapelig, 3=Teknisk
Reell eller kompleks	4	1=Reell, 2=Rektangulær, 3=Polar
Auto eller tilnærm.	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær

**shift() (skift)**

**shift(Heltall1[,#avSkift])⇒heltall**

I binær grunntall-modus:

```

shift(0b1111010110000110101)
                                0b111101011000011010
shift(256,1)                    0b1000000000

```

I heksades grunntall-modus:

Forskyver (skifter) bitene i et binært heltall. Du kan legge inn *Heltall1* med hvilket som helst grunntall. Det konverteres automatisk til 64-bit binær form med fortegn. Hvis *Heltall1* er for stort for denne formen, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **Base2**, side 17.

Hvis *#avSkift* er positiv, er skift til venstre. Hvis *#avSkift* er negativ, er skift til høyre. Grunninnstilling er -1 (skiftes èn bit mot høyre).

I et høyre-skift er biten helt til høyre droppet og 0 eller 1 lagt inn for å stemme overens med den venstre biten. I et venstre-skift er biten helt til venstre droppet og 0 er lagt inn som høyre-bit.

For eksempel i et høyre-skift:

Hver bit skifter mot høyre.

0b0000000000000111101011000011010

Setter inn 0 hvis biten helt til venstre er 0, eller 1 hvis biten helt til venstre er 1.

produserer:

0b000000000000000111101011000011010

Resultatene vises i forhold til grunntallmodusen. Ledende nuller vises ikke.

**shift**(*Liste1* [,*#avSkift*])⇒*liste*

Returnerer en kopi av *Liste1* skiftet til høyre eller til venstre av *#avSkift*-elementer. Endrer ikke *Liste1*.

Hvis *#avSkift* er positiv, er skift til venstre. Hvis *#avSkift* er negativ, er skift til høyre. Grunninnstillingen er -1 (skiftes et element til høyre).

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

**Viktig:** Hvis du vil skrive inn et binært eller heksadesimalt tall, må du alltid bruke prefikset 0b eller 0h (null, ikke bokstaven O).

I desimalt grunntall-modus:

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() (skift)

Katalog > 

Elementer som introduseres ved begynnelsen eller slutten av *liste* ved skiftet er satt til symbolet "udef".

**shift**(*Streng* *l*, #*avSkift*) $\Rightarrow$ *streng*

Returnerer en kopi av *Streng* *l* skiftet mot høyre eller mot venstre av #*ofShifts*-tegn. Endrer ikke *Streng* *l*.

Hvis #*avSkift* er positiv, er skift til venstre. Hvis #*avSkift* er negativ, er skift til høyre. Grunninnstillingen er -1 (skiftes et tegn mot høyre).

Tegn som introduseres ved begynnelsen eller slutten av *streng* ved skiftet er satt til et mellomrom.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

## sign() (fortegn)

Katalog > 

**sign**(*Verdi*) $\Rightarrow$ *verdi*

**sign**(*Liste* *l*) $\Rightarrow$ *liste* *e*

**sign**(*Matrise* *l*) $\Rightarrow$ *matrise*

For reell og komplekst *Verdi* *l*, returnerer *Verdi* *l* / **abs**(*Verdi* *l*) når *Verdi* *l*  $\neq$  0.

Returnerer 1 hvis *Verdi* *l* er positiv.

Returnerer -1 hvis *Verdi* *l* er negativ.

**sign**(0) returnerer  $\pm 1$  hvis kompleks formatmodus er Reell; ellers returnerer den seg selv.

**sign**(0) representerer enhets sirkelen i den komplekse grunnmengden.

For en liste eller matrise returneres fortegnene for alle elementene.

sign(-3.2)	-1
sign({2,3,4,-5})	{1,1,1,-1}

Hvis kompleks formatmodus er reell:

sign([-3 0 3])	[-1 undef 1]
----------------	--------------

## simult()

Katalog > 

**simult**(*koeff* *Matrise*, *konst* *Vektor* (*Tol*)) $\Rightarrow$ *matrise*

Løs mhp.  $x$  og  $y$ :

$$x + 2y = 1$$

Returnerer en kolonnevektor som inneholder løøsningene til et system av lineære ligninger.

Merk: Se også **linSolve()**, side 86.

*coeffMatrise* må være en kvadratmatrise som inneholder ligningskoeffisientene.

*konstVektor* må ha samme antall rader (samme dimension) som *coeffMatrise* og inneholde konstantene.

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \max(\dim(\text{coeffMatrise})) \cdot \text{radNavncoeffMatrise}$

**simult(coeffMatrise, konstMatrise, Tol)** ⇒ *matrise*

Løser multiple systemer av lineære ligninger, hvor hvert system har samme ligningskoeffisienter men forskjellige konstanter.

Hver kolonne i *konstMatrise* må inneholde konstantene for et ligningssystem. Hver kolonne i resultatmatrisen inneholder løøsningen for det tilsvarende systemet.

$$3x + 4y = -1$$

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

Løøsningen er  $x = -3$  og  $y = 2$ .

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{l} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \text{matx1} \\ \text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \end{array} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 0 \\ \frac{1}{2} \\ 2 \end{bmatrix}$$

Løs:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

For det første systemet er  $x = -3$  og  $y = 2$ . For det andre systemet er  $x = -7$  og  $y = 9/2$ .



## sin()



sin(*Verdi*) ⇒ *verdi*

sin(*Liste*) ⇒ *liste*

sin(*Verdi*) returnerer sinus til argumentet.

sin(*Liste*) returnerer en liste over sinus til alle elementer i *Liste*.

**Merk:** Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke °, G, eller r for å hoppe over vinkelmodusen midlertidig.

sin(*kvadratMatrise*) ⇒ *kvadratMatrise*

Returnerer matrisens sinus til *kvadratMatrise*. Dette er ikke det samme som å beregne sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

$$\sin\left(\frac{\pi}{4} r\right) \quad 0.707107$$

$$\sin(45) \quad 0.707107$$

$$\sin(\{0,60,90\}) \quad \{0,0.866025,1.\}$$

I Gradian-vinkelmodus:

$$\sin(50) \quad 0.707107$$

I Radian-vinkelmodus:

$$\sin\left(\frac{\pi}{4}\right) \quad 0.707107$$

$$\sin(45^\circ) \quad 0.707107$$

I Radian-vinkelmodus:

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

## sin<sup>-1</sup>()



sin<sup>-1</sup>(*Liste*) ⇒ *liste*

sin<sup>-1</sup>(*Verdi*) returnerer vinkelen med sinus lik *Verdi*.

sin<sup>-1</sup>(*Liste*) returnerer en liste over invers sinus til hvert element i *Liste*.

**Merk:** Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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



**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arcsin (...)**.

$\sin^{-1}(\text{kvadratMatrise}) \Rightarrow \text{kvadratMatrise}$

Returnerer matrisens inverse sinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\sin^{-1}(\{0,0,2,0,5\}) \quad \{0,.,0.201358,0.523599\}$$

I radian-vinkelmodus og rektangulær, kompleks modus:

$$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right) \quad \begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$$

## $\sinh()$

Katalog >

$\sinh(\text{Verdi}) \Rightarrow \text{verdi}$

$\sinh(\text{Liste}) \Rightarrow \text{liste}$

$\sinh(\text{Verdi})$  returnerer hyperbolsk sinus til argumentet.

$\sinh(\text{Liste})$  returnerer en liste over hyperbolsk sinus til hvert element i *Liste1*.

### $\sinh$

$\sinh(\text{kvadratMatrise}) \Rightarrow \text{kvadratMatrise}$

Returnerer matrisens hyperbolske sinus for *kvadratMatrise1*. Dette er ikke det samme som å beregne hyperbolsk sinus for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\begin{array}{r} \sinh(1.2) \quad 1.50946 \\ \sinh(\{0,1,2,3\}) \quad \{0,1.50946,10.0179\} \end{array}$$

I Radian-vinkelmodus:

$$\sinh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

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

Katalog >

$\sinh^{-1}(\text{Verdi}) \Rightarrow \text{verdi}$

$\sinh^{-1}(\text{Liste}) \Rightarrow \text{liste}$

$\sinh^{-1}(\text{Verdi})$  returnerer invers hyperbolsk sinus til argumentet.

$$\begin{array}{r} \sinh^{-1}(0) \quad 0 \\ \sinh^{-1}(\{0,2,1,3\}) \quad \{0,1.48748,1.81845\} \end{array}$$

sinh<sup>-1</sup>(*Liste1*) returnerer en liste over invers hyperbolsk sinus til hvert element i *Liste1*.

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arcsinh** (...).

sinh<sup>-1</sup>  
(*kvadratMatrise1*) ⇒ *kvadratMatrise*

Returnerer matrisens inverse hyperbolske sinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers hyperbolsk sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

## SinReg

SinReg *X*, *Y* [, [*Iterasjoner*],[*Periode*] [, *Kategori*, *Inkluder* ]

Finner sinusregresjonen for listene *X* og *Y*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Iterasjoner* er en verdi som angir maksimalt antall ganger (1 til 16) det skal gjøres forsøk på å finne en løsning. Hvis utelatt, brukes 8. Vanligvis resulterer større verdier i bedre nøyaktighet men lengre kjøretid, og omvendt.

*Periode* spesifiserer en estimert periode. Hvis utelatt, bør forskjellen mellom verdiene i *X* være like og i sekvensiell rekkefølge. Hvis du spesifiserer *Periode*, kan forskjellene mellom x-verdiene være ulike.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende  $X$  og  $Y$ -dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Resultatet av **SinReg** er alltid i radianer, uavhengig av innstilling for vinkelmodus.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede $X$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede $Y$ -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## SortA (SorterSt)

**SortA**Liste1[, Liste2] [, Liste3]...

{2,1,4,3} → list1

{2,1,4,3}

**SortA**Vektor1[, Vektor2] [, Vektor3] ...

SortA list1

Done

Sorterer elementene i det første argument i stigende rekkefølge.

list1

{1,2,3,4}

{4,3,2,1} → list2

{4,3,2,1}

Hvis du inkluderer andre argumenter, sorteres elementene av hvert slik at den nye posisjonen deres stemmer overens med den nye posisjonen til elementene i det første argumentet.

SortA list2,list1

Done

list2

{1,2,3,4}

list1

{4,3,2,1}

Alle argumentene må være navn på lister eller vektorer. Alle argumentene må ha like dimensjoner.

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 223.

## SortD (SorterSy)

**SortD** *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

**SortD** *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
---------------------------------	---------------

Identisk med **SortA**, bortsett fra at **SortD** sorterer elementene i fallende rekkefølge.

SortD <i>list1</i> , <i>list2</i>	Done
<i>list1</i>	$\{4,3,2,1\}$
<i>list2</i>	$\{3,4,1,2\}$

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 223.

## ►Sphere (sfærisk)

*Vektor* ►Sphere

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Sphere.

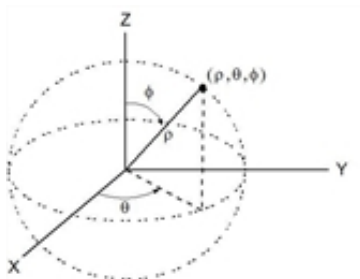
Viser rad- eller kolonnevektor i sfærisk form [ $\rho \angle \theta \angle \phi$ ].

*Vektor* må være av dimensjon 3 og kan enten være en rad- eller en kolonnevektor.

**Merk:** ►Sphere er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje.

$[1 \ 2 \ 3] \blacktriangleright \text{Sphere}$
$[3.74166 \ \angle 1.10715 \ \angle 0.640522]$

$\left( \begin{matrix} 2 & \angle \frac{\pi}{4} & 3 \end{matrix} \right) \blacktriangleright \text{Sphere}$
$[3.60555 \ \angle 0.785398 \ \angle 0.588003]$



sqrt(*Verdi*)⇒*verdi* $\sqrt{4}$  2sqrt(*Listel*)⇒*liste* $\sqrt{\{9,2,4\}}$   $\{3,1.41421,2\}$ 

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Listel*.**Merk:** Se også **Kvadratrot-sjablon**, side 1.

## stat.results (stat.resultats)

stat.results

 $xlist:=\{1,2,3,4,5\}$   $\{1,2,3,4,5\}$ 

Viser resultater fra en statistisk beregning.

 $ylist:=\{4,8,11,14,17\}$   $\{4,8,11,14,17\}$ 

Resultatene vises som et sett av navn-verdi-par. De spesifikke navnene som vises er avhengige av den aller siste brukte statistikkfunksjonen eller kommando.

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"	"{...}"

Du kan kopiere et navn eller en verdi og lime den inn i andre posisjoner.

<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0.,-0.2}"

**Merk:** Unngå å definere variabler som bruker de samme navnene som de som brukes for statistisk analyse. I noen tilfeller kan det oppstå feilbetingelse. I tabellen nedenfor finner du en liste over variabelnavn som brukes for statistisk analyse.

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.̄x
stat.b9	stat.FBlock	stat.β̂	stat.Σx <sup>2</sup>	stat.̄x1
stat.b10	stat.Fcol	stat.β̂1	stat.Σxy	stat.̄x2
stat.bList	stat.FInteract	stat.β̂2	stat.Σy	stat.̄xDiff
stat.χ <sup>2</sup>	stat.FreqReg	stat.β̂Diff	stat.Σy <sup>2</sup>	stat.̄xList
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.ȳ
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.ŷ
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.ŷList
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

**Merk:** Hver gang applikasjonen Lister og regneark beregner statistiske resultater, kopierer den "stat."-gruppevariablene til en "stat#."-gruppe, der # er et tall som økes automatisk. På den måten kan du bevare tidligere resultater mens du utfører flere beregninger.

## stat.values (stat.verdier)

Katalog > 

stat.values

Se [stat.results](#) -eksemplet.

Viser en matrise av verdiene som er beregnet for siste behandlede statistikkfunksjon eller kommando.

I motsetning til [stat.results](#) utelater [stat.values](#) navnene som assosieres med verdiene.

Du kan kopiere en verdi og lime dette inn i andre posisjoner.

**stDevPop**(*Liste*[, *frekvListe*]) $\Rightarrow$ uttrykk

Returnerer populasjonens standardavvik for elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**Merk:** *Liste* må ha minst to elementer. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223

**stDevPop**(*Matrise1*[, *FrekvMatrise*]) $\Rightarrow$ matrise

Returnerer en radvektor av populasjonens standardavvik i kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

**Merk:** *Matrise1* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

I radian-vinkelmodus og automatisk modus:

$\text{stDevPop}(\{1,2,5, -6,3, -2\})$	3.59398
$\text{stDevPop}(\{1.3,2.5, -6.4\}, \{3,2,5\})$	4.11107

$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$	$[3.26599 \quad 2.94392 \quad 1.63299]$
$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$	$[2.52608 \quad 5.21506]$

## stDevSamp() (UtvstdAvv)

**stDevSamp**(*Liste*[, *frekvListe*]) $\Rightarrow$ uttrykk

Returnerer utvalgets standardavvik av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**Merk:** *Liste* må ha minst to elementer. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223

$\text{stDevSamp}(\{1,2,5, -6,3, -2\})$	3.937
$\text{stDevSamp}(\{1.3,2.5, -6.4\}, \{3,2,5\})$	4.33345



## stDevSamp() (UtvstdAvv)

Katalog > 

**stDevSamp**(*Matrise1*,  
*frekvMatrise*) ⇒ *matrise*

Returnerer en radvektor av utvalgets standardavvik av kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

**Merk:** *Matrise1* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 223.

$\text{stDevSamp} \left( \begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix} \right)$
$[4. \quad 3.60555 \quad 2.]$
$\text{stDevSamp} \left( \begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix}, \begin{pmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{pmatrix} \right)$
$[2.7005 \quad 5.44695]$

## Stop (Stopp)

Katalog > 

### Stop

Programmeringskommando: Avslutter programmet.

**Stop** er ikke tillatt i funksjoner.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	<i>Done</i>
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	<i>Done</i>
<i>i</i>	5

## Lagre

Se → (lagre), side 205.

## String() (Streng)

Katalog > 

**string**(*Utr*) ⇒ *string*

Forenkler *Utr* og returnerer resultatet som en tegnstring.

<i>string</i> (1.2345)	"1.2345"
<i>string</i> (1+2)	"3"

**subMat() (undermatrise)**Katalog > 

**subMat**(*MatriseI* [, *startRad*] [, *startKol*] [, *endRad*] [, *endKol*])  
 $\Rightarrow$  *matrise*

Returnerer den spesifiserte undermatrisen til *MatriseI*.

Grunninnstillinger: *startRad*=1, *startKol*=1, *endRad*=siste rad, *endKol*=siste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat( <i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat( <i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

**Sum (Sigma)**

Se G(), side 197.

**sum()**Katalog > 

**sum**(*Liste* [, *Start*] [, *Slutt* ]) $\Rightarrow$  *uttrykk*

Returnerer summen av elementene i *Liste*.

*Start* og *Slutt* er alternativer. De spesifiserer et elementområde.

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *Liste* ignoreres. For mer informasjon om tomme elementer, se side 223.

**sum**(*MatriseI* [, *Start*] [, *Slutt* ]) $\Rightarrow$  *matrise*

Returnerer en radvektor som inneholder summene av elementene i kolonnene i *MatriseI*.

*Start* og *Slutt* er alternativer. De spesifiserer et radområde.

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *MatriseI* ignoreres. For mer informasjon om tomme elementer, se side 223.

sum({ 1,2,3,4,5 })	15
sum({ a,2·a,3·a })	"Error: Variable is not defined"
sum(seq(n,n,1,10))	55
sum({ 1,3,5,7,9 },3)	21

sum( $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ )	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
sum( $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ )	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
sum( $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2,3$ )	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

## sumIf()

Katalog > 

`sumIf(Liste, Kriterium[, SumListe]) ⇒ verdi`

```
sumIf({1,2,e,3,π,4,5,6}, 2.5 < ? < 4.5)  
12.859874482
```

```
sumIf({1,2,3,4}, 2 < ? < 5, {10,20,30,40})  
70
```

Returnerer samlet sum av alle elementene i *Liste* som møter de spesifiserte *Kriterier*. Eventuelt kan du spesifisere en endringsliste, *sumListe*, for å hente de elementene som skal samles (akkumuleres).

*Liste* kan være et uttrykk, en liste eller en matrise. *SumListe*, hvis spesifisert, må ha samme dimensjon(er) som *Liste*.

*Kriterium* kan være:

- En verdi, et uttrykk eller en streng. For eksempel, **34** samler kun de elementene i *Liste* som forenkles til verdien 34.
- Et boolsk uttrykk som inneholder symbolet **?** som plassholder for hvert element. For eksempel, **? < 10** samler kun de elementene i *Liste* som er mindre enn 10.

Hvis et *Liste*-element møter *Kriteriene*, legges dette elementet til den samlende summen. Hvis du inkluderer *sumListe*, legges tilsvarende element fra *sumListe* til summen istedenfor.

I applikasjonen *lister* og *regneark* kan du bruke et celleområde istedenfor *Liste* og *sumListe*.

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 223.

**Merk:** Se også `countIf()`, side 32.

## sumSeq()

Se `Σ()`, side 197.

## system()

Katalog > 

`system(Verdi1 [, Verdi2 [, Verdi3 [, ...]])`

Returnerer et ligningssystem, formatert som en liste. Du kan også opprette et system med en sjablon.

## T

## T(transponert)

*Matrise*<sup>T</sup> ⇒ *matrise*

Returnerer den komplekse konjugerte transponerte av *Matrise*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @t.

## tan()

*tan(Verdi)* ⇒ *verdi*

I Grader-vinkelmodus:

*tan(Liste)* ⇒ *liste*

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) = 1.$$

*tan(Verdi)* returnerer tangens til argumentet.

$$\tan(45) = 1.$$

*tan(Liste)* returnerer en liste over tangensene til alle elementene i *Liste*.

$$\tan(\{0,60,90\}) = \{0.,1.73205,\text{undef}\}$$

**Merk:** Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke °, G, eller r for å hoppe over vinkelmodusen midlertidig.

I Gradian-vinkelmodus:

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) = 1.$$

$$\tan(50) = 1.$$

$$\tan(\{0,50,100\}) = \{0.,1.,\text{undef}\}$$

I Radian-vinkelmodus:

$$\tan\left(\frac{\pi}{4}\right) = 1.$$

$$\tan(45^\circ) = 1.$$

$$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right) = \{0.,1.73205,0.,1.\}$$

*tan(kvadratMatrise)* ⇒ *kvadratMatrise*

I Radian-vinkelmodus:

## tan()

trig 

Returnerer matrisetangensen av *kvadratMatrise1*. Dette er ikke det samme som å beregne tangens for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

## tan<sup>-1</sup>()

trig 

tan<sup>-1</sup>(Verdi) ⇒ verdi

I Grader-vinkelmodus:

tan<sup>-1</sup>(Liste) ⇒ liste

$$\tan^{-1}(1) \quad 45$$

tan<sup>-1</sup>(Verdi) returnerer vinkelen med tangens til *Verdi*.

I Gradian-vinkelmodus:

tan<sup>-1</sup>(Liste) returnerer en liste over de inverse tangenser til hvert element i *Liste*.

$$\tan^{-1}(1) \quad 50$$

**Merk:** Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

I Radian-vinkelmodus:

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arctan(...)**.

### tan<sup>-1</sup>

(*kvadratMatrise1*) ⇒ *kvadratMatrise*

I Radian-vinkelmodus:

Returnerer matrisens inverse tangens til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

## tanh()

Katalog > 

tanh(Verdi) ⇒ verdi

$$\tanh(1.2) \quad 0.833655$$

tanh(Liste) ⇒ liste

$$\tanh(\{0,1\}) \quad \{0, .761594\}$$

**tanh(Verdi)** returnerer hyperbolsk tangens til argumentet.

**tanh(ListeI)** returnerer en liste av hyperbolske tangenser til hvert element i *ListeI*.

**tanh**  
(*kvadratMatriseI*) $\Rightarrow$ *kvadratMatrise*

Returnerer matrisens hyperbolske tangens til *kvadratMatriseI*. Dette er ikke det samme som å beregne hyperbolsk tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatriseI* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$$\text{tanh} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

## tanh<sup>-1</sup>()

**tanh<sup>-1</sup>(Verdi)** $\Rightarrow$ *verdi*

**tanh<sup>-1</sup>(ListeI)** $\Rightarrow$ *liste*

**tanh<sup>-1</sup>(Verdi)** returnerer invers hyperbolsk tangens til argumentet.

**tanh<sup>-1</sup>(ListeI)** returnerer en liste over invers hyperbolsk tangens til hvert element i *ListeI*.




**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arctanh** (...).

**tanh<sup>-1</sup>**  
(*kvadratMatriseI*) $\Rightarrow$ *kvadratMatrise*

Returnerer matrisens inverse hyperbolske tangens til *kvadratMatriseI*. Dette er ikke det samme som å beregne invers hyperbolsk tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.




I rektangulært, kompleks format:

$$\begin{array}{l} \text{tanh}^{-1}(0) \quad 0. \\ \text{tanh}^{-1}(\{1,2,1,3\}) \\ \{ \text{undef}, 0.518046-1.5708 \cdot i, 0.346574-1.5708 \cdot i \} \end{array}$$

For å se hele resultatet, trykk på  og bruk så  og  for å bevege markøren.

I radian-vinkelmodus og rektangulært, kompleks format:

$$\text{tanh}^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.94730 \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

For å se hele resultatet, trykk på  og bruk så  og  for å bevege markøren.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

**tCdf**(*nedGrense*,*øvreGrense*,*df*)⇒*tall* hvis *nedGrens* og *øvreGrens* er *tall*, *liste* hvis *nedGrens* og *øvreGrens* er *lister*

Beregner student-*t*-fordelingens sannsynlighet mellom *nedGrense* og *øvreGrense* for spesifisert grader av frihet *df*.

For  $P(X \leq \text{øvreGrense})$ , sett *nedreGrense* = -9E999.

**Text***promptStreng*[, *VisFlagg*]

Programmeringskommando: Stopper programmet og viser tegnstrengen *promptStreng* i en dialogboks.

Når brukeren klikker på **OK**, fortsetter programmet å utføre.


Det valgfrie *flagg* -argumentet kan være et hvilket som helst uttrykk.

- Hvis *VisFlagg* utelates eller behandles til **1**, blir tekstmeldingen lagt til i Kalkulator-loggen.
- Hvis *VisFlagg* behandles til **0**, blir tekstmeldingen ikke lagt til i loggen.

Hvis programmet trenger et skrevet svar fra brukeren, kan du se etter på **Request**, side 131 eller **RequestStr**, side 133.

**Merk:** Du kan bruke denne kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Definer et program som stopper for å vise hver av de fem tilfeldige tallene i en dialogboks.

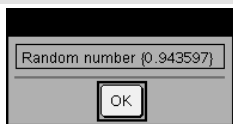
Innenfor malen `Prgm...EndPrgm` fullfører du hver linje ved å trykke på  istedenfor på `enter`. På tastaturet på datamaskinen, hold nede **Alt** og trykk på **Enter**.

```
Define tekst_demo()=Prgm
  For i,1,5
    strinfo:="Random number " &
string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Kjør program:

```
tekst_demo()
```

Eksempel på dialogboks:



## Then (Så)

Se If, side 72.

## tInterval

katalog > **tInterval** *Liste*[,*Frekv*[,*CNivå*]]

(Dataliste-inndata)

**tInterval**  $\bar{x}$ ,*sx*,*n*[,*CNivå*]

(Oppsummerende statistikk-inndata)

Beregner et *t*-konfidensintervall. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjennomsnitt
stat. $\bar{x}$	Utvalgets gjennomsnitt av datasekvensen fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat. $\sigma_x$	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt

## tInterval\_2Samp

Katalog > **tInterval\_2Samp** *Liste1*,*Liste2*[,*Frekv1* [,*Frekv2* [,*CNivå* [,*Felles*]]]]]

(Dataliste inndata)

**tInterval\_2Samp**  $\bar{x}1$ ,*sx1*,*n1*, $\bar{x}2$ ,*sx2*,*n2*



[,CNivå,Felles]

(Summering statistikk inndata)

Beregner et to-utvalgs  $t$  konfidensintervall.En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).*Felles=1* fellesvarianser; *Felles=0* gjør ikke fellesvarianser.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\bar{x}1-\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat. $\bar{x}1$ , stat. $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma x1$ , stat. $\sigma x2$	Utvalgets standardavvik for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.sp	Det felles standardavviket. Beregnet når <i>Felles = JA</i> .

**tPdf()****tPdf(*XVerd*,*df*)**  $\Rightarrow$  tall hvis *XVerd* er et tall,  
*liste* hvis *XVerd* er en listeBeregner sannsynlighetstetthetsfunksjonen (pdf) for Student- $t$ -fordelingen ved en spesifisert  $x$ -verdi med spesifiserte grader av frihet *df*.

**trace()**katalog > **trace**(*kvadratMatrise*)⇒*verdi*

Returnerer diagonalsummen (summen av alle elementene på hoveddiagonalen) til *kvadratMatrise*.

$\text{trace}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	15
$a:=12$	12
$\text{trace}\left(\begin{bmatrix} a & 0 \\ 1 & a \end{bmatrix}\right)$	24

**Try**Katalog > **Try***blokk1***Else***blokk2***EndTry**

Utfører *blokk1* med mindre det oppstår en feil. Programmet overfører til *blokk2* hvis en feil oppstår i *blokk1*.

Systemvariabelen *feilKode* inneholder feilkoden, dermed kan programmet utføre retting av feil. For en liste over feilkoder, se "Feilkoder og feilmeldinger," side 233.

*blokk1* og *blokk2* kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet "·".

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

## Eksempel 2

For å se kommandoene **Try**, **ClrErr** og **PassErr** i drift, legg inn *egenverdier()* - programmet som vist til høyre. Kjør programmet ved å utføre hver av følgende uttrykk.

---


$$\text{egenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$


---

**Merk:** Se også **ClrErr**, side 23, og **PassErr**, side 116.

Define *prog1()*=Prgm

Try

*z:=z+1*

Disp "z incremented."

Else

Disp "Sorry, z undefined."

EndTry

EndPrgm

Done

*z:=1:prog1()*

z incremented.

Done

DelVar *z:prog1()*

Sorry, z undefined.

Done

Define *egenverdier(a,b)*=Prgm© Programmet *egenverdier(A,B)* viser *egenverdier* av *A·B*

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Egenverdier av A·B er: ",egVd(a\*b)

Else

```

If feilKode=230 Then
    Disp "Feil: Produkt av A·B må være en
    kvadratmatrise"
    ClrErr
Else
    PassErr
EndIf
EndTry
EndPrgm

```

**tTest**

**tTest**  $\mu_0, \text{Liste[, Frekv[, Hypot]]}$

(Dataliste inndata)

**tTest**  $\mu_0, \bar{x}, sx, n, [\text{Hypot}]$

(Summering statistikk inndata)

Utfører en hypotesetest for ett enkelt ukjent populasjonsgjennomsnitt  $\mu$  når populasjonens standardavvik  $\sigma$  er ukjent. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

Test  $H_0: \mu = \mu_0$ , mot ett av følgende:

For  $H_a: \mu < \mu_0$ , sett *Hypot*<0

For  $H_a: \mu \neq \mu_0$  (standard), sett *Hypot*=0

For  $H_a: \mu > \mu_0$ , set *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \text{sqrt}(n))$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved

Utdata-variabel	Beskrivelse
stat.df	Grader-av-frihet
stat. $\bar{x}$	Utvalgsgjennomsnitt av datasekvensen i <i>Liste</i>
stat.sx	Utvalgets standardavvik av datasekvensen
stat.n	Utvalgenes størrelse

## tTest\_2Samp

Katalog > 

**tTest\_2Samp** *Liste1, Liste2[, Frekv1[, Frekv2  
[, Hypot[, Felles]]]]*

(Dataliste inndata)

**tTest\_2Samp**  $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2[, Hypot  
[, Felles]]$

(Summering statistikk inndata)

Beregner en to-utvalgs  $t$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

Test  $H_0$ :  $\mu_1 = \mu_2$ , mot ett av følgende:

For  $H_a$ :  $\mu_1 < \mu_2$ , sett *Hypot*<0

For  $H_a$ :  $\mu_1 \neq \mu_2$  (standard), sett *Hypot*=0

For  $H_a$ :  $\mu_1 > \mu_2$ , sett *Hypot*>0

*Felles*=1 fellesvarianser

*Felles* =0 gir ikke fellesvarianser

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.t	Standard normalverdi beregnet for forskjellen i gjennomsnitt
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for t-statistikken
stat. $\bar{x}1$ , stat. $\bar{x}2$	Utvalgets gjennomsnitt av datasekvenser i <i>Liste1</i> og <i>Liste2</i>
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste2</i>

Utdata-variabel	Beskrivelse
stat.n1, stat.n2	Utvalgenes størrelse
stat.sp	Det felles standardavviket. Beregnet når <i>Felles</i> =1.

### tvmFV()

Katalog > 

**tvmFV**(*N,I,PV,Pmt,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ verdi

tvmFV(120,5,0,-500,12,12) 77641.1

Finansiell funksjon som beregner fremtidig verdi for penger.

**Merk:** Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 170. Se også **amortTbl()**, side 7.

### tvmI()

Katalog > 

**tvmI**(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ verdi

tvmI(240,100000,-1000,0,12,12) 10.5241

Finansiell funksjon som beregner rentefoten per år.

**Merk:** Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 170. Se også **amortTbl()**, side 7.

### tvmN()

Katalog > 

**tvmN**(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ verdi

tvmN(5,0,-500,77641,12,12) 120.

Finansiell funksjon som beregner antallet betalingsperioder.

**Merk:** Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 170. Se også **amortTbl()**, side 7.

### tvmPmt()

Katalog > 

**tvmPmt**(*N,I,PV,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ verdi

tvmPmt(60,4,30000,0,12,12) -552.496

**tvmPmt()**Katalog > 

Finansiell funksjon som beregner beløpet for hver betaling.

**Merk:** Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 170. Se også **amortTbl()**, side 7.

**tvmPV()**Katalog > 

**tvmPV**(*N,I,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒verdi

tvmPV(48,4, 500,30000,12,12)      -3426.7

Finansiell funksjon som beregner nåverdien.

**Merk:** Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 170. Se også **amortTbl()**, side 7.

TVM-argument*	Beskrivelse	Datatype
N	Antall betalingsperioder	reelt tall
I	Årlig rente (rentefot)	reelt tall
PV	Nåverdi	reelt tall
Pmt	Betalingsbeløp	reelt tall
FV	Fremtidig verdi	reelt tall
PpY	Antall betalinger pr. år, grunninnstilling=1	heltall > 0
CpY	Antall renteperioder pr. år, grunninnstilling=1	heltall > 0
PmtAt	Betaling som forfaller ved slutten eller begynnelsen av hver periode, grunninnstilling=avslutt	heltall (0=avslutte, 1=begynne)

\* Disse tidsverdi-for-penger-argumentnavnene likner TVM-variabelnavnene som f.eks. **tvm.pv** og **tvm.pmt**) som brukes av *Calculator* applikasjonens finansielløser. Men finansielle funksjoner lagrer ikke argumentverdiene eller resultatene til TVM-variablene.

**TwoVar**Katalog > 

**TwoVar** *X, Y*, [*Frekv*] [, *Kategori*, *Inkluder*]]

Beregner 2-variabels statistiske observatorer. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 154.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

*X* og *Y* er lister av uavhengige og avhengige variabler.

*Frekv* er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

*Kategori* er en liste over numerisk eller streng kategorikoder for de tilsvarende *X* og *Y*-dataene..

*Inkluder* er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene *X*, *Frekv* eller *Kategori* resulterer i et tomt (åpent) element for det tilsvarende elementet til alle disse listene. Et tomt element i enhver av listene fra *X1* til *X20* resulterer i et tomt (åpent) element for det tilsvarende elementet i alle disse listene. For mer informasjon om tomme elementer, se side 223.

Utdata-variabel	Beskrivelse
stat. $\bar{x}$	Gjennomsnitt av x-verdier
stat. x	Sum av x-verdier
stat. x2	Sum av x2 verdier
stat.sx	Utvalgets standardavvik til x
stat. x	Populasjonens standardavvik til x
stat.n	Antall datapunkter
stat. $\bar{y}$	Gjennomsnitt av y-verdier

Utdata-variabel	Beskrivelse
stat. y	Sum av y-verdier
stat. y <sup>2</sup>	Sum av y <sup>2</sup> -verdier
stat.sy	Utvalgets standardavvik til y
stat. y	Populasjonens standardavvik til y
stat. xy	Sum av x · y -verdier
stat.r	Korrelasjonskoeffisient
stat.MinX	Minimum av x-verdier
stat.Q <sub>1</sub> X	Første kvartil av x
stat.MedianX	Median av x
stat.Q <sub>3</sub> X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.MinY	Minimum av y-verdier
stat.Q <sub>1</sub> Y	Første kvartil av y
stat.MedY	Median av y
stat.Q <sub>3</sub> Y	Tredje kvartil av y
stat.MaxY	Maksimum av y-verdier
stat. (x- ) <sup>2</sup>	Sum av kvadratavvik fra gjennomsnittet av x
stat. (y- ) <sup>2</sup>	Sum av kvadrat for avvik fra gjennomsnittet av y

## U

### unitV() (enhetsV)

Katalog > 

**unitV(Vektor1) ⇒ vektor**

Returnerer enten en rad- eller kolonne-  
enhetsvektor, avhengig av formen på  
Vektor1.

Vektor1 må være enten en enkel-rad-  
matrise eller en enkel-kolonne-matrise.

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0.408248 & 0.816497 & 0.408248 \end{bmatrix}$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} 0.267261 \\ 0.534522 \\ 0.801784 \end{bmatrix}$$



**unLock**Katalog > **unLock***Var1* [, *Var2*] [, *Var3*] ...

<i>a</i> :=65	65
---------------	----

**unLock***Var*.

Lock <i>a</i>	Done
---------------	------

Låser opp spesifisert variabel eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes.

getLockInfo( <i>a</i> )	1
-------------------------	---

<i>a</i> :=75	"Error: Variable is locked."
---------------	------------------------------

DelVar <i>a</i>	"Error: Variable is locked."
-----------------	------------------------------

Se **Lock**, side 90, og **getLockInfo()**, side 68.

Unlock <i>a</i>	Done
-----------------	------

<i>a</i> :=75	75
---------------	----

DelVar <i>a</i>	Done
-----------------	------

**V****varPop()**Katalog > **varPop**(*Liste* [, *frekvListe*]) ⇒ *uttrykk*

varPop({5,10,15,20,25,30})	72.9167
----------------------------	---------

Returnerer populasjonsvariansen for *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**Merk:** *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 223.

**varSamp() (utvalgets varians)**Katalog > **varSamp**(*Liste* [, *frekvListe*]) ⇒ *uttrykk*

varSamp({1,2,5,-6,3,-2})	31
--------------------------	----

Returnerer utvalgets varians for *Liste*.

	2
--	---

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

varSamp({1,3,5},{4,6,2})	68
--------------------------	----

	33
--	----

**Merk:** *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 223.

**varSamp**(*Matrise1*,  
*frekvMatrise*)⇒*matrise*

Returnerer en radvektor som inneholder utvalgets varians for hver kolonne i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

Hvis et element i en av matrisene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre matrisen ignoreres også. For mer informasjon om tomme elementer, se side 223.

**Merk:** *Matrise1* må inneholde minst to rader.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$[4.75 \ 1.03 \ 4]$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}, \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$	$[3.91731 \ 2.08411]$

## W

### Wait



#### Wait *tidenISekunder*

Utsetter utførelsen i en periode på *tidenISekunder* sekunder.

**Wait** er spesielt nyttig i et program som krever en kort forsinkelse for at anmodede data skal bli tilgjengelige.

Argumentet *tidenISekunder* må være et uttrykk som forenkler til en desimalverdi i området 0 til og med 100. Kommandoen avrunder denne verdien opp til nærmeste 0,1 sekunder.

For å avbryte en **Wait** som pågår,

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.

For å vente 4 sekunder:

**Wait 4**

For å vente 1/2 sekund:

**Wait 0.5**

For å vente 1,3 sekunder med bruk av variabelen *seccount*:

**seccount:=1.3**  
**Wait seccount**

Dette slår på en grønn indikatorlampe i 0,5 sekunder og slår den deretter av.

**Send "SET GREEN 1 ON"**  
**Wait 0.5**  
**Send "SET GREEN 1 OFF"**

- **Windows®**: Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®**: Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®**: Applikasjonen viser en ledetekst. Du kan fortsette å vente, eller avbryte.

**Merk:** Du kan bruke kommandoen **Wait** innenfor et brukerdefinert program, men ikke innenfor en funksjon.

## warnCodes ()


**warnCodes**(*Uttr1*, *StatusVar*) $\Rightarrow$ uttrykk

Behandler uttrykk *Uttr1*, returnerer resultatet, og lagrer kodene for alle genererte varsler i listevariabelen *StatusVar*. Hvis ingen varsler er generert, tildeler denne funksjonen *StatusVar* en tom liste.

*Uttr1* kan være et hvilket som helst gyldig matematisk uttrykk i TI-Nspire™ eller TI-Nspire™ CAS. Du kan ikke bruke en kommando eller tildeling som *Uttr1*.

*StatusVar* må være et gyldig variabelnavn.

Se side 241 for en liste over varselkoder og assosierte meldinger.

 warnCodes(det([1.23456E-999]),warn)	1.23456E-999
warn	{ 10029 }

## when() (når)

**when**(*Betingelse*, *santResultat* [, *usantResultat*][, *ukjentResultat*])  
 $\Rightarrow$ uttrykk

Returnerer *santResultat*, *usantResultat*, eller *ukjentResultat*, avhengig av om *Betingelse* er sann, usann eller ukjent. Returnerer inndata hvis det er for få argumenter til å spesifisere korrekt resultat.

## when() (når)

Katalog > 

Utelat både *usantResultat* og *ukjentResultat* for å definere et uttrykk bare i det området der *Betingelse* er sann.

Bruk et **undef** *usantResultat* for å definere et uttrykk som bare plottes grafen på et intervall.

**when()** er nyttig for å definere rekursive funksjoner.

$\text{when}(x < 0, x + 3), x = 5$	undef
------------------------------------	-------

$\text{when}(n > 0, n \cdot \text{factoral}(n - 1), 1) \rightarrow \text{factoral}(n)$	Done
$\text{factoral}(3)$	6
$3!$	6

## While

Katalog > 

### While *Betingelse*

*Blokk*

### EndWhile

Utfører utsagnene i *Blokk* så lenge som *Betingelse* er sann.

*Blokk* kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

Define $\text{sum\_of\_recip}(n) = \text{Func}$	
Local $i, \text{tempsum}$	
$1 \rightarrow i$	
$0 \rightarrow \text{tempsum}$	
While $i \leq n$	
$\text{tempsum} + \frac{1}{i} \rightarrow \text{tempsum}$	
$i + 1 \rightarrow i$	
EndWhile	
Return $\text{tempsum}$	
EndFunc	
	Done
$\text{sum\_of\_recip}(3)$	$\frac{11}{6}$

## X

## xor (enten ...eller ...)

Katalog > 

*BoolskUttr1* xor *BoolskUttr2* returnerer *Boolsk uttrykk*

$\text{true xor true}$	false
$5 > 3 \text{ xor } 3 > 5$	true

*BoolskListe1* xor *BoolskListe2* returnerer *Boolsk liste*

*BoolskMatrise1* xor *BoolskMatrise2* returnerer *Boolsk matrise*

Returnerer sann hvis *BoolskUttr1* er sant og *BoolskUttr2* er usant eller omvendt.

Returnerer usann hvis begge argumentene er sanne eller hvis begge er usanne. Returnerer et forenklet Boolsk uttrykk hvis ikke noen av argumentene kan avgjøres som sanne eller usanne.

**Merk:** Se or, side 113.

*Heltall1 xor Heltall2*  $\Rightarrow$  *heltall*

Sammenlikner to reelle heltall bit-for-bit med en **xor**-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis en av bitene (men ikke begge) er 1; resultatet er 0 hvis begge bitene er 0 eller begge biter er 1. Returnert verdi representerer bit-resultatene og vises i grunntall-modus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **►Base2**, side 17.

**Merk:** Se or, side 113.

## Z

### zInterval

**zInterval**  $\sigma, \text{Liste}[, \text{Frekv}[, \text{CNivå}]]$

(Dataliste inndata)

**zInterval**  $\sigma, \bar{x}, n [, \text{CNivå}]$

(Summering statistikk inndata)

I heksades grunntall-modus:

**Viktig:** Null, ikke bokstaven O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

I binær grunntall-modus:

0b100101 xor 0b100	0b100001
--------------------	----------

**Merk:** Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

Beregner et  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen (side 154).

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjennomsnitt
stat. $\bar{x}$	Utvalgets gjennomsnitt av datasekvensen fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.sx	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt
stat. $\sigma$	Kjent populasjons standardavvik for datasekvens <i>Liste</i>

## zInterval\_1Prop

**zInterval\_1Prop**  $x, n$  [ $CNivå$ ]

Beregner et en-proporsjons  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

$x$  er et ikke-negativt heltall.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\hat{p}$	Beregnet andel (brøkdel) av suksesser
stat.ME	Feilmargin
stat.n	Antall utvalg i datasekvens

## zInterval\_2Prop

**zInterval\_2Prop**  $x1, n1, x2, n2$  [ $CNivå$ ]

Beregner et to-proporsjons  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

$x1$  og  $x2$  er ikke-negative heltall.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\hat{p}$ Diff	Beregnet differanse mellom andeler (brøkdeler)
stat.ME	Feilmargin
stat. $\hat{p}1$	Beregnet andel av suksesser i utvalg 1
stat. $\hat{p}2$	Beregnet andel av suksesser i utvalg 2
stat.n1	Utvalgsstørrelse i datasekvens 1
stat.n2	Utvalgsstørrelse i datasekvens 2

**zInterval\_2Samp**  $\sigma_1, \sigma_2, List1, List2$   
 $[, Frekv1[, Frekv2[, CNivå]]]$

(Dataliste inndata)

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}, n2[CNivå]$

(Summering statistikk inndata)

Beregner et to-utvalgs  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling

Utdata-variabel	Beskrivelse
stat. $\bar{x}1$ - $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat. $\bar{x}1$ , stat. $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma1$ , stat. $\sigma2$	Utvalgets standardavvik for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.r1, stat.r2	Kjent populasjons standardavvik for datasekvens <i>Liste 1</i> og <i>Liste 2</i>

## zTest

Katalog > 

**zTest**  $\mu0, \sigma, \text{Liste}, [\text{Frekv}, \text{Hypot}]$

(Dataliste inndata)

**zTest**  $\mu0, \sigma, \bar{x}, n, [\text{Hypot}]$

(Summering statistikk inndata)

Utfører en  $z$ -test med frekvens *frekvlste*. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

Test  $H_0: \mu = \mu0$ , mot ett av følgende:

For  $H_a: \mu < \mu0$ , sett *Hypot*<0

For  $H_a: \mu \neq \mu0$  (standard), sett *Hypot*=0

For  $H_a: \mu > \mu0$ , sett *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu0) / (\sigma / \text{sqrt}(n))$
stat.P-Verdi	Minste sannsynlighet som null-hypotesen kan forkastes ved
stat. $\bar{x}$	Utvalgsgjennomsnitt av datasekvensen i <i>Liste</i>
stat.sx	Utvalgets standardavvik av datasekvensen. Returneres kun for inndata <i>Data</i> .
stat.n	Utvalgenes størrelse



**zTest\_1Prop**  $p0, x, n[, Hypot]$ 

Beregner en en-proporsjons  $z$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

$x$  er et ikke-negativt heltall.

Test  $H_0: p = p0$  mot ett av følgende:

For  $H_a: p > p0$ , sett *Hypot*>0

For  $H_a: p \neq p0$  (standard), sett *Hypot*=0

For  $H_a: p < p0$ , sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 223).

Utdata-variabel	Beskrivelse
stat.p0	Hypotesisk populasjonsandel
stat.z	Standard normalverdi beregnet for andelen
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat. $\hat{p}$	Beregnet andel av suksesser
stat.n	Utvalgenes størrelse

**zTest\_2Prop**  $x1, n1, x2, n2[, Hypot]$ 

Beregner en to-proporsjons  $z$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

$x1$  og  $x2$  er ikke-negative heltall.

Test  $H_0: p1 = p2$  mot ett av følgende:

For  $H_a: p1 > p2$ , sett *Hypot*>0

For  $H_a: p1 \neq p2$  (standard), sett *Hypot*=0

For  $H_a: p < p0$ , sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 223).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for differansen av andelene
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat. $\hat{p}$ 1	Beregnet andel av suksesser i utvalg 1
stat. $\hat{p}$ 2	Beregnet andel av suksesser i utvalg 2
stat. $\hat{p}$	Beregnet samlet andel av suksesser
stat.n1, stat.n2	Antall utvalg som er tatt i forsøk 1 og 2

**zTest\_2Samp**  $\sigma_1, \sigma_2, Liste1, Liste2[, Frekv1 [, Frekv2[, Hypot]]]$

(Dataliste inndata)

**zTest\_2Samp**  $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, Hypot]$

(Summering statistikk inndata)

Beregner en to-utvalgs  $z$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 154).

Test  $H_0: \mu_1 = \mu_2$ , mot ett av følgende:

For  $H_a: \mu_1 < \mu_2$ , sett *Hypot*<0

For  $H_a: \mu_1 \neq \mu_2$  (standard), sett *Hypot*=0

For  $H_a: \mu_1 > \mu_2$ , *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 223).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for forskjellen i gjennomsnitt
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved

Utdata-variabel	Beskrivelse
stat. $\bar{x}$ 1, stat. $\bar{x}$ 2	Utvalgets gjennomsnitt av datasekvenser i <i>Liste1</i> og <i>Liste2</i>
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste2</i>
stat.n1, stat.n2	Utvalgenes størrelse

# Symboler

## + (addere)

**+** tast

$Verdi1 + Verdi2 \Rightarrow verdi$

56	56
----	----

Returnerer summen av de to argumentene.

$56+4$	60
--------	----

$60+4$	64
--------	----

$64+4$	68
--------	----

$68+4$	72
--------	----

$Liste1 + Liste2 \Rightarrow liste$

$\left\{22, \pi, \frac{\pi}{2}\right\} \rightarrow l1$	$\{22, 3.14159, 1.5708\}$
--	---------------------------

$Matrise1 + Matrise2 \Rightarrow matrise$

$\left\{10, 5, \frac{\pi}{2}\right\} \rightarrow l2$	$\{10, 5, 1.5708\}$
--	---------------------

Returnerer en liste (eller matrise) som inneholder summene av tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrise1* og *Matrise2*).

$l1+l2$	$\{32, 8.14159, 3.14159\}$
---------	----------------------------

Dimensjonene i argumentene må være like.

$Verdi + Liste1 \Rightarrow liste$

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
-----------------------	------------------

$Liste1 + Verdi \Rightarrow liste$

$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$
-----------------------	------------------

Returnerer en liste med summene av *Verdi* og hvert element i *Liste1*.

$Verdi + Matrise1 \Rightarrow matrise$

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

$Matrise1 + Verdi \Rightarrow matrise$

Returnerer en matrise med *Verdi* addert til hvert element på diagonalen til *Matrise1*. *Matrise1* må være kvadratisk.

**Merk:** Bruk .+ (prikk pluss) for å addere et uttrykk til hvert element.

## -(subtrahere)

**-** tast

$Verdi1 - Verdi2 \Rightarrow verdi$

$6-2$	4
-------	---

Returnerer *Verdi1* minus *Verdi2*.

$\pi - \frac{\pi}{6}$	2.61799
-----------------------	---------

**-(subtrahere)** $Liste1 - Liste2 \Rightarrow liste$ 

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} \quad \left\{ 12, -1.85841, 0 \right\}$$

 $Matrise1 - Matrise2 \Rightarrow matrise$ 

$$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 2 & 2 \end{bmatrix}$$

Subtraherer hvert element i *Liste2* (eller *Matrise2*) fra tilsvarende element i *Liste1* (eller *Matrise1*), og returnerer resultatene.

Dimensjonene i argumentene må være like.

 $Verdi - Liste1 \Rightarrow liste$ 

$$15 - \{10, 15, 20\} \quad \{5, 0, -5\}$$

 $Liste1 - Verdi \Rightarrow liste$ 

$$\{10, 15, 20\} - 15 \quad \{-5, 0, 5\}$$

Subtraherer hvert *Liste1* element fra *Verdi* eller subtraherer *Verdi* fra hvert *Liste1* element og returnerer en liste over resultatene.

 $Verdi - Matrise1 \Rightarrow matrise$ 

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

 $Matrise1 - Verdi \Rightarrow matrise$ 

*Verdi - Matrise1* returnerer en matrise av *Verdi* hver gang identitetsmatrisen trekkes fra *Matrise1*. *Matrise1* må være kvadratisk.

*Matrise1 - Verdi* returnerer en matrise av *Verdi* hver gang identitetsmatrisen subtraheres fra *Matrise1*. *Matrise1* må være kvadratisk.

**Merk:** Bruk .- (prikk minus) for å subtrahere et uttrykk fra hvert element.

**•(multiplisere)** $Verdi1 \cdot Verdi2 \Rightarrow verdi$ 

$$2 \cdot 3.45 \quad 6.9$$

Returnerer produktet av de to argumentene.

 $Liste1 \cdot Liste2 \Rightarrow liste$ 

$$\{1, 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

Returnerer en liste som inneholder produktene av samsvarende elementer i *Liste1* og *Liste2*.

Dimensjonene i listene må være like.

**•(multiplisere)****x** tast*Matrise1*•*Matrise2*⇒*matrise*Returnerer matriseproduktet av *Matrise1* og *Matrise2*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 7 & 8 \\ 7 & 8 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 42 & 48 \\ 105 & 120 \end{bmatrix}$$

Antallet kolonner i *Matrise1* må være likt antallet rader i *Matrise2*.*Verdi* •*Liste1*⇒*liste*

$$\pi \cdot \{4,5,6\} = \{12.5664, 15.708, 18.8496\}$$

*Liste1* •*Verdi*⇒*liste*Returnerer en liste med produktene av *Verdi* og hvert element i *Liste1*.*Verdi* •*Matrise1*⇒*matrise**Matrise1*•*Verdi*⇒*matrise*Returnerer en matrise med produktene av *Verdi* og hvert element i *Matrise1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$6 \cdot \text{identity}(3) = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

**Merk:** Bruk •(prikk multipliser) for å multiplisere et uttrykk med hvert element.**/ (divider)****÷** tast*Verdi1* / *Verdi2*⇒*verdi*Returnerer kvotienten av *Verdi1* dividert med *Verdi2*.

$$\frac{2}{3.45} = 0.57971$$

**Merk:** Se også **Brøk-sjablon**, side 1.*Liste1* / *Liste2*⇒*liste*Returnerer en liste som inneholder kvotientene av *Liste1* dividert med *Liste2*.

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

Dimensjonene i listene må være like.

*Verdi* / *Liste1* ⇒ *liste**Liste1* / *Verdi* ⇒ *liste*Returnerer en liste med kvotientene av *Verdi* dividert med *Liste1* eller *Liste1* dividert med *Verdi*.

$$\frac{6}{\{3,6,\sqrt{6}\}} = \{2, 1, 2.44949\}$$

$$\frac{\{7,9,2\}}{7 \cdot 9 \cdot 2} = \left\{\frac{1}{18}, \frac{1}{14}, \frac{1}{63}\right\}$$

*Verdi* / *Matrise1*⇒ *matrise**Matrise1* / *Verdi* ⇒ *matrise*

$$\frac{\{7 \ 9 \ 2\}}{7 \cdot 9 \cdot 2} = \begin{bmatrix} \frac{1}{18} & \frac{1}{14} & \frac{1}{63} \end{bmatrix}$$

## / (divider)



Returnerer en matrise med kvotientene av *Matrise1* / *Verdi*.

**Merk:** Bruk . / (prikk divider) for å dividere et uttrykk med hvert element.

## ^ (potens)



*Verdi1* ^ *Verdi2* ⇒ *verdi*

$$4^2 \qquad 16$$

*Liste1* ^ *Liste2* ⇒ *liste*

$$\{2,4,6\}^{\{1,2,3\}} \qquad \{2,16,216\}$$

Returnerer det første argument opphøyd i det andre argumentet.

**Merk:** Se også **Eksponent-sjablon**, side 1.

For en liste, returneres elementene i *Liste1* opphøyd i tilsvarende elementer i *Liste2*.

I reell grunnmengde bruker brøkpotens som har forkortet eksponent med oddetall i nevner en rell forgreining i motsetning til hovedforgreining for kompleks modus.

*Verdi* ^ *Liste1* ⇒ *liste*

$$\pi^{\{1,2,-3\}} \qquad \{3.14159,9.8696,0.032252\}$$

Returnerer *Verdi* opphøyd i elementene i *Liste1*.

*Liste1* ^ *Verdi* ⇒ *liste*

$$\{1,2,3,4\}^{-2} \qquad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Returnerer elementene i *Liste1* opphøyd i *Verdi*.

*kvadratMatrise1* ^ *heltall* ⇒ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \qquad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Returnerer *kvadratMatrise1* opphøyd i *heltall* -potens.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \qquad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

*kvadratMatrise1* må være en kvadratmatrise.

Hvis *heltall* = -1, beregnes invers matrise.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \qquad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Hvis *heltall* < -1, beregnes invers matrise opphøyd i en korrekt positiv potens.

**x<sup>2</sup> (kvadrat)** **tast***Verdi* <sup>2</sup> ⇒ *verdi*

$4^2$	16
-------	----

Returnerer kvadratet av argumentet.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

*Liste* <sup>2</sup> ⇒ *liste*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

Returnerer en liste med kvadrater av elementene i *Liste1*.*kvadratMatrise* <sup>2</sup> ⇒ *matrise*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} .^2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
---	--

Returnerer matrisens kvadrat av *kvadratMatrise1*. Dette er ikke det samme som å beregne kvadratet av hvert element. Bruk  $.^2$  for å beregne kvadratet av hvert element.**.+ (prikk adder)** **taster***Matrise1* .+ *Matrise2* ⇒ *matrise*

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 11 & 32 \\ 23 & 44 \end{bmatrix}$
--	--

*Verdi* .+ *Matrise1* ⇒ *matrise*

$5 .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 15 & 35 \\ 25 & 45 \end{bmatrix}$
---	--

*Matrise1* .+ *Matrise2* returnerer en matrise som er summen av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.*Verdi* .+ *Matrise1* returnerer en matrise som er summen av *Verdi* og hvert element i *Matrise1*.**.- (prikk subtr.)** **taster***Matrise1* .- *Matrise2* ⇒ *matrise*

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$	$\begin{bmatrix} -9 & -18 \\ -27 & -36 \end{bmatrix}$
--	---

*Verdi* .- *Matrise1* ⇒ *matrise*

$5 .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$	$\begin{bmatrix} -5 & -15 \\ -25 & -35 \end{bmatrix}$
---	---

*Matrise1* .- *Matrise2* returnerer en matrise som er differansen mellom hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.*Verdi* .- *Matrise1* returnerer en matrise som er differansen av *Verdi* og hvert element i *Matrise1*.



**.• (prikk mult.)**

.	×	taster
---	---	--------

Matrise1 .• Matrise2 ⇒ matrise

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 10 & 40 \\ 90 & 160 \end{bmatrix}$$

Verdi .• Matrise1 ⇒ matrise

$$5 \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

Matrise1 .• Matrise2 returnerer en matrise som er produktet av hvert par av samsvarende elementer i Matrise1 og Matrise2.

Verdi .• Matrise1 returnerer en matrise med produktene av Verdi og hvert element i Matrise1.

**. / (prikk divider)**

.	÷	taster
---	---	--------

Matrise1 ./ Matrise2 ⇒ matrise

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} ./ \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

Verdi ./ Matrise1 ⇒ matrise

$$5 ./ \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{8} \end{bmatrix}$$

Matrise1 ./ Matrise2 returnerer en matrise som er kvotient av hvert par av samsvarende elementer i Matrise1 og Matrise2.

Verdi ./ Matrise1 returnerer en matrise som er kvotient av Verdi og hvert element i Matrise1.

**.^ (prikk potens)**

.	^	taster
---	---	--------

Matrise1 .^ Matrise2 ⇒ matrise

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .^ \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 27 & \frac{1}{4} \end{bmatrix}$$

Verdi .^ Matrise1 ⇒ matrise

$$5 .^ \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 25 \\ 125 & \frac{1}{5} \end{bmatrix}$$

Matrise1 .^ Matrise2 returnerer en matrise, der hvert element i Matrise2 er eksponenten for samsvarende element i Matrise1.

Verdi .^ Matrise1 returnerer en matrise, der hvert element i Matrise1 er eksponenten for Verdi.



## = (er lik)

 tast

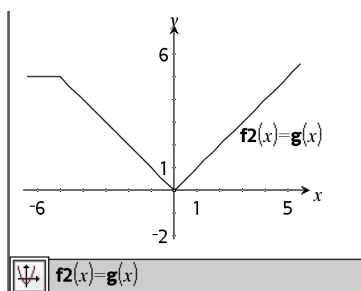
For lister og matriser, returneres sammenlikninger element for element.

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

```
Define g(x)=Func
  If x<=5 Then
    Return 5
  ElseIf x>5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

Done

Resultat av grafisk fremstilling g(x)



## ≠ (ulik)

  taster

$Uttr1 \neq Uttr2 \Rightarrow$  Boolsk uttrykk

Se “=” (er lik) eksempel.

$Liste1 \neq Liste2 \Rightarrow$  Boolsk liste

$Matrise1 \neq Matrise2 \Rightarrow$  Boolsk matrise



Returnerer sann hvis  $Uttr1$  er bestemt å være ulik  $Uttr2$ .

Returnerer usann hvis  $Uttr1$  er bestemt å være lik  $Uttr2$ .

Alt annet returnerer en forenklet form av ligningen.



For lister og matriser, returneres sammenlikninger element for element.

**≠ (ulik)**

  **taster**

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive /=

**< (mindre enn)**

  **taster**

$Uttr1 < Uttr2 \Rightarrow$  *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 < Liste2 \Rightarrow$  *Boolsk liste*

$Matrise1 < Matrise2 \Rightarrow$  *Boolsk matrise*

Returnerer sann hvis  $Uttr1$  er bestemt å være mindre enn  $Uttr2$ .

Returnerer usann hvis  $Uttr1$  er bestemt å være større enn eller lik  $Uttr2$ .

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

**≤ (mindre enn eller lik)**

  **taster**

$Uttr1 \leq Uttr2 \Rightarrow$  *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 \leq Liste2 \Rightarrow$  *Boolsk liste*

$Matrise1 \leq Matrise2 \Rightarrow$  *Boolsk matrise*

Returnerer sann hvis  $Uttr1$  er bestemt å være mindre enn eller lik  $Uttr2$ .


Returnerer usann hvis  $Uttr1$  er bestemt å være større enn  $Uttr2$ .

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive <=

## > (større enn)

ctrl  taster

$Uttr1 > Uttr2 \Rightarrow$  *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 > Liste2 \Rightarrow$  *Boolsk liste*

$Matrise1 > Matrise2 \Rightarrow$  *Boolsk matrise*

Returnerer sann hvis  $Uttr1$  er bestemt å være større enn  $Uttr2$ .

Returnerer usann hvis  $Uttr1$  er bestemt å være mindre enn eller lik  $Uttr2$ .

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

## ≥ (større enn eller lik med)

ctrl  taster

$Uttr1 \geq Uttr2 \Rightarrow$  *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 \geq Liste2 \Rightarrow$  *Boolsk liste*

$Matrise1 \geq Matrise2 \Rightarrow$  *Boolsk matrise*

Returnerer sann hvis  $Uttr1$  er bestemt å være større enn eller lik  $Uttr2$ .

Returnerer usann hvis  $Uttr1$  er bestemt å være mindre enn eller lik  $Uttr2$ .

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive  $>=$

## ⇒ (logisk implikasjon)

ctrl [=]-taster

*BoolskUttr1* ⇒ *BoolskUttr2* returnerer  
*Boolsk uttrykk*

5>3 or 3>5 true

*BoolskListe1* ⇒ *BoolskListe2*  
returnerer *Boolsk liste*

5>3 ⇒ 3>5 false

3 or 4 7

*BoolskMatrise1* ⇒ *BoolskMatrise2*  
returnerer *Boolsk matrise*

3 ⇒ 4 -4

{1,2,3} or {3,2,1} {3,2,3}

{1,2,3} ⇒ {3,2,1} {-1,-1,-3}

*Heltall1* ⇒ *Heltall2* returnerer *Heltall*

Behandler uttrykket **not** <argument1> or <argument2> og returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive ⇒

## ↔ (logisk dobbel implikasjon, XNOR)

ctrl [=]-taster

*BoolskUttr1* ↔ *BoolskUttr2* returnerer  
*Boolsk uttrykk*

5>3 xor 3>5 true

*BoolskListe1* ↔ *BoolskListe2*  
returnerer *Boolsk liste*

5>3 ↔ 3>5 false

3 xor 4 7

*BoolskMatrise1* ↔ *BoolskMatrise2*  
returnerer *Boolsk matrise*

3 ↔ 4 -8

{1,2,3} xor {3,2,1} {2,0,2}

{1,2,3} ↔ {3,2,1} {-3,-1,-3}

*Heltall1* ↔ *Heltall2* returnerer *Heltall*

Returnerer negasjon av en **XOR** Boolsk handling på de to argumentene. Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive <=>

## ! (fakultet)

 **tast**

**Verdi1!** ⇒ verdi

5! 120

**Liste1!** ⇒ liste

$\{\{5,4,3\}\}!$   $\{120,24,6\}$

**Matrise1!** ⇒ matrise

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$   $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Returnerer argumentets fakultet.

For en liste eller matrise, returneres en liste eller matrise av elementenes fakulteter.

## & (legg til)

  **taster**

**Streng1 & Streng2** ⇒ streng

"Hello "&"Nick" "Hello Nick"

Returnerer en tekststreng som er *Streng2* lagt til *Streng1*.

## d ( ) (derivert)

**Katalog** > 

**d(Uttr1, Var[, Orden])** |  
*Var=Verdi*⇒verdi

$\frac{d}{dx}(|x|)|_{x=0}$  undef

**d(Uttr1, Var[, Orden])**⇒verdi

$x:=0: \frac{d}{dx}(|x|)$  undef

**d(Liste1, Var[, Orden])**⇒liste

$x:=3: \frac{d}{dx}(\{x^2, x^3, x^4\})$   $\{6, 27, 108\}$

**d(Matrise1, Var[, Orden])**⇒matrise

Unntatt mens du bruker den første syntaksen, må du lagre en numerisk verdi i variabel *Var* før du behandler **d()**. Se eksemplene.

**d()** kan brukes for å beregne første og andre deriverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.

*Orden*, hvis inkludert, må være=1 eller 2. Standard er 1.

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **derivative** (...).

**Merk:** Se også Første deriverte, side 5 eller Andre deriverte, side 6.

## d () (derivert)

Katalog > 

**Merk:** Algoritmen d() har en begrensning: den arbeider rekursivt gjennom det ikke-forenklede uttrykket og beregner den numeriske verdien av den første deriverte (og eventuelt den andre) og behandlingen av hvert deluttrykk, som kan føre til et uventet resultat.

Studer eksemplet til høyre. Den første deriverte av  $x \cdot (x^2+x)^{1/3}$  i  $x=0$  er lik 0. Men siden den første deriverte av deluttrykket  $(x^2+x)^{1/3}$  er udefinert i  $x=0$  og denne verdien blir brukt til å beregne den deriverte av hele uttrykket, rapporterer **d()** resultatet som udefinert og viser en varselmelding.

Hvis du støter på denne begrensningen, verifiserer du løsningen grafisk. Du kan også prøve å bruke **centralDiff()**.

$$\frac{d}{dx} \left( x \cdot (x^2+x)^{\frac{1}{3}} \right) \Big|_{x=0} \quad \text{undef}$$

$$\text{centralDiff} \left( x \cdot (x^2+x)^{\frac{1}{3}}, x \right) \Big|_{x=0} \quad 0.000033$$

## ∫ () (integral)

Katalog > 

∫(Utt1, Var, Nedre, Øvre) ⇒ verdi



Returnerer integralet av *Utt1* med hensyn på variabelen *Var* fra *Nedre* til *Øvre*. Kan brukes for å beregne den bestemte integralet numerisk ved hjelp av samme metode som nInt().

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **integral** (...).

**Merk:** Se også **nInt()**, side 107 og **Bestemt integral-sjablon**, side 6.

$$\int_0^1 x^2 dx \quad 0.333333$$

## √ () (kvadratroten)

  **taster**

√(Verdi1) ⇒ verdi

√(Liste1) ⇒ liste

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Liste1*.

$$\sqrt{4} \quad 2$$
$$\sqrt{\{9,2,4\}} \quad \{3,1.41421,2\}$$



## $\sqrt{()}$ (kvadratrot)

ctrl x<sup>2</sup> taster

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **sqrt (...)**

**Merk:** Se også **Kvadratrot-sjablon**, side 1.

## $\Pi()$ (prodSeq)

Katalog &gt;

$\Pi(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow$  uttrykk

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **prodSeq (...)**.

Finner *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer produktet av resultatene.

**Merk:** Se også **Produkt-sjablon ( $\Pi$ )**, side 5.

$\Pi(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Nedre}-1) \Rightarrow 1$

$\Pi(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow 1/\Pi(\text{Uttr1}, \text{Var}, \text{Øvre}+1, \text{Nedre}-1)$  hvis  $\text{Øvre} < \text{Nedre} - 1$

Produktformlene som er brukt er hentet fra følgende referanse:

Ronald L. Graham, Donald E. Knuth, og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

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

$$\prod_{n=1}^5 \left( \left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

$$\prod_{k=4}^3 (k) \quad 1$$

$$\prod_{k=4}^1 \left( \frac{1}{k} \right) \quad 6$$

$$\prod_{k=4}^1 \left( \frac{1}{k} \right) \cdot \prod_{k=2}^4 \left( \frac{1}{k} \right) \quad \frac{1}{4}$$

## $\Sigma()$ (sumSeq)

Katalog &gt;

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow$  uttrykk

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **sumSeq (...)**.

Behandler *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer summen av resultatene.

**Merk:** Se også **Sum-sjablon**, side 5.

$$\sum_{n=1}^5 \left( \frac{1}{n} \right) \quad \frac{137}{60}$$

$\Sigma(\text{Uttr}1, \text{Var}, \text{Nedre}, \text{Nedre}-1) \Rightarrow 0$

$$\sum_{k=4}^3 (k) = 0$$

$\Sigma(\text{Uttr}1, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow -\Sigma(\text{Uttr}1, \text{Var}, \text{Øvre}+1, \text{Nedre}-1)$  hvis  $\text{Øvre} < \text{Nedre}-1$

$$\sum_{k=4}^1 (k) = -5$$

Summeringsformlene som er brukt er hentet fra følgende referanse:

Ronald L. Graham, Donald E. Knuth, og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

$\Sigma\text{Int}()$

$\Sigma\text{Int}(\text{NPmt}1, \text{NPmt}2, N, I, \text{PV}, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{avrundVerdi}]) \Rightarrow \text{verdi}$

$\Sigma\text{Int}(1, 3, 12, 4, 75, 20000, , 12, 12) = -213.48$

$\Sigma\text{Int}(\text{NPmt}1, \text{NPmt}2, \text{amortTabell}) \Rightarrow \text{verdi}$

$\text{tbl} := \text{amortTbl}(12, 12, 4, 75, 20000, , 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	6.55	-1703.37	-12.02

$\Sigma\text{Int}(1, 3, \text{tbl}) = -213.48$

Amortiseringsfunksjon som beregner rentesummen i løpet av en spesifisert rekke av betalinger.

$\text{NPmt}1$  og  $\text{NPmt}2$  definerer start- og sluttgrensene for betalingsrekken.

$N, I, \text{PV}, \text{Pmt}, \text{FV}, \text{PpY}, \text{CpY}$  og  $\text{PmtAt}$  er beskrevet i tabellen med TVM-argumenter, side 170.

- Hvis du utelater  $\text{Pmt}$ , grunninnstilles den til  $\text{Pmt} = \text{tvmPmt}(N, I, \text{PV}, \text{FV}, \text{PpY}, \text{CpY}, \text{PmtAt})$ .
- Hvis du utelater  $\text{FV}$ , grunninnstilles den til  $\text{FV} = 0$ .
- Grunninnstillingene for  $\text{PpY}, \text{CpY}$  og  $\text{PmtAt}$  er de samme som for TVM-funksjonene.

$\text{avrundVerdi}$  spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

$\Sigma\text{Int}(NPmt1, NPmt2, amortTabell)$

beregner rentesummen basert på amortiseringstabell *amortTabell*.

Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

**Merk:** Se også  $\Sigma\text{Prn}()$ , nedenfor, og **Bal()**, side 16.

 $\Sigma\text{Prn}()$ 

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [avrundVerdi]) \Rightarrow verdi$

$\Sigma\text{Prn}(1,3,12,4,75,20000,,12,12)$	-4916.28
--	----------

 $\Sigma\text{Prn}$ 

$(NPmt1, NPmt2, amortTable) \Rightarrow verdi$

Amortiseringsfunksjon som beregner summen av hovedbetalinger i løpet av en spesifisert rekke av betalinger.

*NPmt1* og *NPmt2* definerer start- og sluttgrensene for betalingsrekken.

*N, I, PV, Pmt, FV, PpY, CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 170.

- Hvis du utelater *Pmt*, grunninnstilles den til  **$Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$** .
- Hvis du utelater *FV*, grunninnstilles den til  $FV = 0$ .
- Grunninnstillingene for *PpY, CpY* og *PmtAt* er de samme som for TVM-funksjonene.

*avrundVerdi* spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

$\Sigma\text{Prn}(NPmt1, NPmt2, amortTabell)$

beregner summen av hovedbetalinger basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

$tbl := \text{amortTbl}(12, 12, 4, 75, 20000, , 12, 12)$			
0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Prn}(1,3,12)$	-4916.28
----------------------------	----------

**Merk:** Se også ΣInt(), over, og Bal(), side 16.

## # (Indir.ref)

ctrl  taster

## # varNavnStreng

Refererer til variabelen med navnet *varNavnStreng*. På denne måten kan du bruke strenger for å opprette variabelnavn "innenfra" en funksjon.

$xyz:=12$	12
$\#("x"&"y"&"z")$	12

Oppretter eller refererer til variabelen xyz.

$10 \rightarrow r$	10
$"r" \rightarrow s1$	"r"
$\#s1$	10

Returnerer verdien av variabelen (r) som har et navn som er lagret i variabel s1.

## E (vitenskapelig tallnotasjon)

 tast*mantissa*Eksponent

Legger inn et tall i vitenskapelig fremstilling. Tallet blir tolket som en *mantissa*  $\times 10^{\text{eksponent}}$ .

23000.	23000.
2300000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

Tips: Hvis du vil legge inn en potens av 10 uten å forårsake desimale verdier i resultatet, bruk  $10^{\text{heltall}}$ .

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @E. Eksempel: Skriv 2.3@E4 for å legge inn 2.3E4.

## g (gradian)

1 tast

*Utr1g* ⇒ *Uttrykk*

I grader, gradian eller radian modus:

*Utr1g* ⇒ *Uttrykk*

$\cos(50^g)$	0.707107
$\cos(\{0,100^g,200^g\})$	{1,0,-1}

*Liste1g* ⇒ *liste*

*Matrise1g* ⇒ *matrise*

## g (gadian)

1 tast

Denne funksjonen gir deg en mulighet til å spesifisere en vinkel i gradianer mens du er i grader- eller gradian-modus.

I radian-vinkelmodus, multipliseres  $Uttr1$  med  $\pi/200$ .

I grader-vinkelmodus, multipliseres  $Uttr1$  med  $g/100$ .

I gradian modus, returneres  $Uttr1$  uendret.

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @g.

## r (radian)

1 tast

$Verdi1^r \Rightarrow verdi$

I grader, gradian eller radian modus:

$Liste1^r \Rightarrow liste$

$Matrise1^r \Rightarrow matrise$

Denne funksjonen gir deg en mulighet til å spesifisere en gradian vinkel mens du er i grader- eller radian-modus.

$\cos\left(\frac{\pi}{4^r}\right)$	0.707107
$\cos\left(\left\{0^r, \left(\frac{\pi}{12}\right)^r, -(\pi)^r\right\}\right)$	{1, 0.965926, -1.}

I grader-vinkelmodus, multipliseres argumentet med  $180/\pi$ .

I radian-vinkelmodus, returneres argumentet uendret.

I gradian modus, multipliseres argumentet med  $200/\pi$ .

Tips: Bruk  $r$  hvis du vil tvinge radianer inn en funksjonsdefinisjon uavhengig av hvilken modus som er i bruk når du bruker funksjonen.

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @r.

## ° (grader)

1 tast

$Verdi1^\circ \Rightarrow verdi$

I grader, gradian eller radian modus:

## ° (grader)

 tast

Liste1°⇒liste

$\cos(45^\circ)$  0.707107

Matrise1°⇒matrise

I radian modus:

Denne funksjonen gir deg en mulighet til å spesifisere en grader-vinkel mens du er i gradian eller radian modus.

$\cos\left\{\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right\}$   
 $\{1, 0.707107, 0., 0.864976\}$

I radian-vinkelmodus, multipliseres argumentet med  $\pi/180$ .

I grader-vinkelmodus, returneres argumentet uendret.

I gradian-vinkelmodus, multipliseres argumentet med 10/9.

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @d.

## °, ', " (grader/minutter/sekunder)

  taster

gg°mm'ss.ss"⇒Uttrykk

I Grader-vinkelmodus:

gg Et positivt eller negativt tall

$25^\circ 13' 17.5''$  25.2215

mm Et ikke-negativt tall

$25^\circ 30'$   $\frac{51}{2}$

ss.ss Et ikke-negativt tall

Returnerer  $gg+(mm/60)+(ss.ss/3600)$ .

Dette grunntall -60-formatet lar deg:

- Sette inn en vinkel i grader/minutter/sekunder uten hensyn til aktuell vinkelmodus.
- Sette inn tid, som timer/minutter/sekunder.

**Merk:** Sett to apostrofer (") etter ss.ss, ikke et anførselstegn (').

## ∠ (vinkel)

  taster

[Radius,∠θ\_Vinkel]⇒vektor

I radian modus og vektorformat innstilt på rektangulær

(polar inndata)

## ∠ (vinkel)

ctrl  taster

[Radius, ∠θ\_Vinkel, Z\_Koordinat] ⇒ vektor

$$\begin{bmatrix} 5 & \angle 60^\circ & \angle 45^\circ \\ 1.76777 & 3.06186 & 3.53553 \end{bmatrix}$$

(sylindrisk inndata)

sylindrisk

[Radius, ∠θ\_Vinkel, ∠θ\_Vinkel] ⇒ vektor  
(sfærisk inndata)

$$\begin{bmatrix} 5 & \angle 60^\circ & \angle 45^\circ \\ 3.53553 & \angle 1.0472 & 3.53553 \end{bmatrix}$$

Returnerer koordinater som en vektor, avhengig av vektorformatets modusinnstilling: rektangulær, sylindrisk eller sfærisk.

sfærisk

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @<.

$$\begin{bmatrix} 5 & \angle 60^\circ & \angle 45^\circ \\ 5. & \angle 1.0472 & \angle 0.785398 \end{bmatrix}$$

(Størrelse ∠ Vinkel) ⇒ kompleks Verdi

I radian-vinkelmodus og rektangulært, komplekst format:

(polar inndata)

$$5+3 \cdot i - \left( 10 \angle \frac{\pi}{4} \right) \quad -2.07107 - 4.07107 \cdot i$$

Setter inn en kompleks verdi i (r∠θ) polar form. *Vinkelen* tolkes avhengig av aktuell vinkelmodus-innstilling.

\_ (senket strek som et tomt element)

Se "Tomme (åpne) elementer" |, side 223.

## 10^()

Katalog > 

10^ (Verdi I) ⇒ verdi

$$10^{1.5} \quad 31.6228$$

10^ (Liste I) ⇒ liste

Returnerer 10 opphøyd i argumentets potens.

For en liste, returneres 10 opphøyd i elementenes potens i *Liste I*.

10^ (kvadratMatrise) ⇒ kvadratMatrise

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} \quad \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

## 10^()

Katalog > 

Returnerer 10 opphøyd i potensen av *kvadratMatrise1*. Dette er ikke det samme som å beregne 10 opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

*kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

## ^-1(resiprok)

Katalog > 

Verdi  $\wedge^{-1} \Rightarrow$  verdi

$$(3.1)^{-1}$$

0.322581

Liste  $\wedge^{-1} \Rightarrow$  liste

Returnerer den inverse verdien av et argument.

For en liste, returneres den inverse verdien av elementene i *Liste1*.



*kvadratMatrise1*  $\wedge^{-1} \Rightarrow$  *kvadratMatrise*

Returnerer den inverse verdien av *kvadratMatrise1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

*kvadratMatrise1* må være en ikke-singulær kvadratisk matrise.

## | (begrensningsoperator)

  -taster

Uttr | *BoolskUttr1* [and *BoolskUttr2*]...

$$x+1|x=3$$

4

Uttr | *BoolskUttr1* [or *BoolskUttr2*]...

$$x+55|x=\sin(55)$$

54.0002


Begrensningssymbolet ("|") fungerer som en binær operator. Operanden til venstre for | er et uttrykk. Operanden til høyre for | spesifiserer en eller flere relasjoner som kan ha innvirkning på forenklingen av uttrykket. Flere forbindelser etter | må sammenføres av logiske "and" eller "or" operatører.

Med begrensningsoperatoren har du tre utgangstyper av funksjonalitet:

- Erstatninger



## | (begrenningsoperator)

ctrl  -taster

- Intervallbegrensninger
- Eksklusjoner

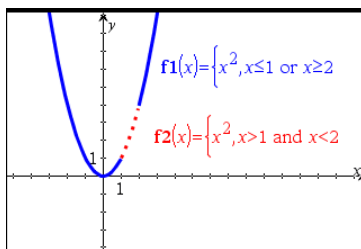
En erstatning har form som en ligning, som  $x=3$  eller  $y=\sin(x)$ . For at den skal være mest effektiv, bør den venstre siden være en enkel variabel. *Uttr* | *Variabel* = *verdi* vil erstatte *verdi* for hver forekomst av *Variabel* i *Uttr*.

Intervallbegrensninger tar form som en eller flere ulikheter som er føyd sammen av logiske "and" eller "or" operatore. En intervallbegrensning tillater også forenkling som ellers kan være ugyldig eller ikke mulig å beregne.

Eksklusjoner bruker relasjons-operatoren "ulik" ( $\neq$  eller  $\neq$ ) for å ekskludere en spesifikk verdi fra å komme i betraktning.

$x^3-2\cdot x+7 \rightarrow f(x)$	Done
$f(x) x=\sqrt{3}$	8.73205

$nSolve(x^3+2\cdot x^2-15\cdot x=0,x)$	0.
$nSolve(x^3+2\cdot x^2-15\cdot x=0,x) x>0$ and $x<5$	3.



## → (lagre)

ctrl var tast

*Verdi* → *Var*

*Liste* → *Var*

*Matrise* → *Var*

*Uttr* → *Funksjon*(*Param1*,...)

*Liste* → *Funksjon*(*Param1*,...)

*Matrise* → *Funksjon*(*Param1*,...)

Hvis variabel *Var* ikke eksisterer, opprettes *Var* og initialiserer den til *Verdi*, *Liste*, eller *Matrise*.

$\frac{\pi}{4} \rightarrow myvar$	0.785398
$2 \cdot \cos(x) \rightarrow yI(x)$	Done
$\{1,2,3,4\} \rightarrow lst5$	$\{1,2,3,4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → <i>str1</i>	"Hello"

**→ (lagre)**

ctrl var

tast

Hvis *Var* allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med *Verdi*, *Liste*, eller *Matrise*.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive =: som en snarvei. Eksempel: Skriv  $\pi/4$   
=: **minvar**.

**:= (tildele)**

ctrl |shift|

taster

*Var* := *Verdi*

*Var* := *Liste*

*Var* := *Matrise*

*Funksjon*(*Param1*,...) := *Uttr*

*Funksjon*(*Param1*,...) := *Liste*

*Funksjon*(*Param1*,...) := *Matrise*

Hvis variabel *Var* ikke eksisterer, opprettes *Var* og initialiserer den til *Verdi*, *Liste*, eller *Matrise*.

Hvis *Var* allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med *Verdi*, *Liste*, eller *Matrise*.

$myvar := \frac{\pi}{4}$	.785398
$y1(x) := 2 \cdot \cos(x)$	Done
$lst5 := \{1, 2, 3, 4\}$	{ 1, 2, 3, 4 }
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := "Hello"$	"Hello"

**© (kommentar)**

ctrl

|shift|

taster

© [*tekst*]

© fremstiller *tekst* som en kommentarlinje som lar deg kommentere funksjoner og programmer som du oppretter.

© kan plasseres ved begynnelsen eller hvor som helst på linjen. Alt som er til høyre for ©, til slutten av linjen, er kommentaren.

Define  $g(n) = \text{Func}$

© Declare variables

Local *i*, *result*

*result* := 0

For *i*, 1, *n*, 1 © Loop *n* times

*result* := *result* +  $i^2$

EndFor

Return *result*

EndFunc

Done

$g(3)$  14

**Merk for å legge inn eksemplet:** For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

**Ob, Oh****OB taster, OH taster****Ob binærTall**

I desimalt grunntall-modus:

0b10+0hF+10	27
-------------	----

**Oh heksadesimalTall**

Markerer hhv. et binært eller heksadesimalt tall. For å sette inn et binært eller heksadesimalt tall må du sette inn prefikset Ob eller Oh uavhengig av grunninnstillingsmodus. Uten prefiks blir et tall behandlet som et desimaltall (grunntall 10).

I binær grunntall-modus:

0b10+0hF+10	0b11011
-------------	---------

Resultatene vises i forhold til grunntall-modusen.

I heksades grunntall-modus:

0b10+0hF+10	0h1B
-------------	------

# TI-Nspire™ CX II – Tegnekommandoer

Dette er et tilleggsdokument for TI-Nspire™-referanseguiden og TI-Nspire™ CAS-referanseguiden. Alle TI-Nspire™ CX II-kommandoene vil bli innlemmet og publisert i versjon 5.1 av TI-Nspire™-referanseguiden og TI-Nspire™ CAS-referanseguiden.

## Grafikkprogrammering

Nye kommandoer er lagt til på TI-Nspire™ CX II-grafregnere og TI-Nspire™-skrivebordsprogrammer for grafikkprogrammering.

TI-Nspire™ CX II-grafregnere vil bytte til denne grafikkmodusen mens den utfører grafikkkommandoer og bytte tilbake til konteksten der programmet ble utført etter at programmet er ferdig.

Skjermen viser «Kjører ...» i topplinjen mens programmet utføres. Det viser «Ferdig» når programmet er fullført. Ethvert tastetrykk tar systemet ut av grafikkmodus.

- Overgangen til grafikkmodus utløses automatisk når en av Draw (grafikk)-kommando oppstår under utførelsen av TI-Basic-programmet.
- Denne overgangen vil bare skje når du kjører et program fra kalkulator; i et dokument eller kalkulator i kladdemark.
- Overgangen ut av grafikkmodus skjer ved avslutning av programmet.
- Grafikkmodusen er bare tilgjengelig på TI-Nspire™ CX II-grafregnerne og skrivebordsvisningen for TI-Nspire™ CX II CAS-grafregnerenviseren. Dette betyr at det ikke er tilgjengelig i datamaskindokumentvisningen på skrivebordet eller i iOS.
  - Hvis det oppdages en grafikkkommando mens du kjører et TI-Basic-program fra feil kontekst, vises en feilmelding og TI-Basic-programmet avsluttes.

## Grafikkskjerm

Grafikkskjerm bildet vil inneholde en overskrift øverst på skjermen som det ikke kan skrives i av grafikkkommandoer.

Tegningsområdet for grafikkskjerm bildet vil bli slettet (farge = 255, 255, 255) når grafikkskjerm bildet initialiseres.

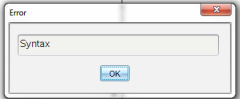
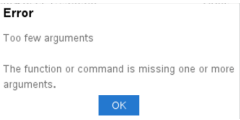
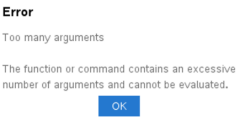
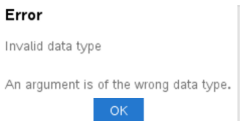
Grafikkskjerm	Standard
Høyde	212
Bredde	318
Farge	hvit: 255,255,255

## Standardvisning og innstillinger

- Statusikonene i topplinjen (batteristatus, trykk-for-å-teste-status, nettverksindikator etc.) vil ikke være synlige når et grafikkprogram kjører.
- Standard tegnefarge: Black (0,0,0)
- Standard pennestil – normal, glatt
  - Tykkelse: 1 (tynn), 2 (normal), 3 (tykkest)
  - Stil 1 (jevn), 2 (stiplet), 3 (punkt)
- Alle tegnekommandoer vil bruke de nåværende innstillingene for farge og penn; enten standardverdier eller de som ble angitt via TI-Basic-kommandoer.
- Skrivestilen er fastsatt og kan ikke endres.
- Eventuelle utdata til grafikkskjerm bildet vil bli tegnet i et klippevindu som er på størrelse med tegneområdet på grafikkskjermen. Tegnete utdata som strekker seg utenfor dette klippede grafikkskjermområdet, tegnes ikke. Ingen feilmelding vises.
- Alle x,y-kordinater angitt for tegnekommandoer er definert slik at 0,0 er øverst til venstre i grafikkskjermens tegneområde.
  - **Unntak:**
    - **DrawText** bruker koordinatene som nedre venstre hjørne av markeringsrammen for teksten.
    - **SetWindow** bruker nedre venstre hjørne av skjermen
- Alle parametere for kommandoene kan gis som uttrykk, som evaluerer til et tall som deretter rundes av til nærmeste heltall.

## Feilmeldinger på grafikkskjerm

Hvis valideringen mislykkes, vises en feilmelding.

Feilmelding	Beskrivelse	Vise
Feil Syntaks	Dersom syntakskontrollen oppdager syntaksfeil, vises en feilmelding, og kontrollen prøver å plassere markøren nær den første feilen så du kan korrigere den.	
Feil For få argumenter	Funksjonen eller kommandoen mangler et eller flere argumenter	
Feil For mange argumenter	Funksjonen eller kommandoen inneholder for mange argumenter og kan ikke behandles.	
Feil Ugyldig datatype	Et argument er av feil datatype.	

### Ugyldige kommandoer i grafikkmodus

Noen kommandoer er ikke tillatt når programmet bytter til grafikkmodus. Hvis disse kommandoene oppstår i grafikkmodus, vises feil og programmet avsluttes.

Forbudt kommando	Feilmelding
Forespør	Forespørselen kan ikke utføres i grafikkmodus
ForespørStr	RequestStr kan ikke utføres i grafikkmodus
Tekst	Tekst kan ikke utføres i grafikkmodus

Kommandoene som skriver ut tekst til kalkulatoren – **disp** og **dispAt** – vil være støttede kommandoer i grafikkonteksten. Teksten fra disse kommandoene vil bli sendt til Kalkulator-skjermbildet (ikke på Grafikk) og vil være synlig etter at programmet er avsluttet og systemet skifter tilbake til kalkulator-appen

## Slett

**Tøm** *x, y, bredde, høyde*

Slett

Tømmer hele skjermen hvis ingen parametere er spesifisert.

Tømmer hele skjerm

Hvis *x, y, bredde* og *høyde* er spesifisert, blir rektangelet som er definert av parametrene slettet.

Clear 10,10,100,50

Tømmer et rektangelområde med øverste venstre hjørne på (10, 10), med bredde 100 og høyde 50

---

## DrawArc

 Katalog >   
 CXII

**DrawArc**  $x, y, \text{bredde}, \text{høyde}, \text{startAngle}, \text{arcAngle}$

Tegn en bue i det definerte avgrensende rektangelet med de angitte start- og buevinklene.

$x, y$ : øvre venstre koordinat for avgrensende rektangel

*bredde, høyde*: dimensjoner av avgrensende rektangel

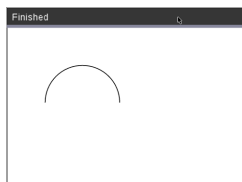
«arc angle» definerer vinkelåpningen for buen.

Disse parametere kan gis som uttrykk, som evaluerer til et tall som deretter rundes av til nærmeste heltall.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se også: [FillArc](#)

## DrawCircle

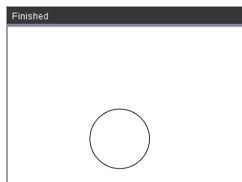
 Katalog >   
 CXII

**DrawCircle**  $x, y, \text{radius}$

$x, y$ : koordinat for sentrum

*radius*: radiusen på sirkelen

DrawCircle 150,150,40



Se også: [FillCircle](#)

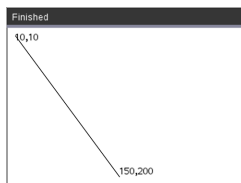


**DrawLine**  $x1, y1, x2, y2$ Tegn en linje fra  $x1, y1, x2, y2$ .

Uttrykk som evaluerer til et tall som deretter rundes av til nærmeste heltall.

**Skjermgrenser:** Hvis de angitte koordinatene fører til at en del av linjen tegnes utenfor grafikkskjermen, blir den delen av linjen klippet og ingen feilmelding vises.

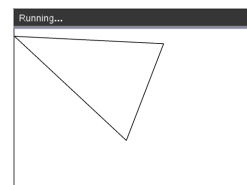
DrawLine 10,10,150,200

**DrawPoly**

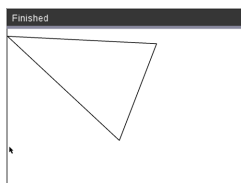
Kommandoene har også to varianter:

**DrawPoly**  $xlist, ylist$ 

eller

**DrawPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$ **Merk:** DrawPoly  $xlist, ylist$ Form vil koble  $x1, y1$  til  $x2, y2$ ,  $x2, y2$  til  $x3, y3$  og så videre.**Merk:** DrawPoly  $x1, y1, x2, y2, x3, y3...xn, yn$  $xn, yn$  vil **IKKE** kobles til  $x1, y1$  automatisk.Uttrykk som evaluerer til en liste over reelle flyttall  
 $xlist, ylist$ Uttrykk som evaluerer til ett enkelt reelt flyttall  
 $x1, y1...xn, yn$  = koordinater for polygohjørnene $xlist:={0,200,150,0}$  $ylist:={10,20,150,10}$ DrawPoly  $xlist,ylist$ 

DrawPoly 0,10,200,20,150,150,0,10



**Merk: DrawPoly:** Inndata-størrelsesdimensjoner (bredde/høyde) i forhold til tegnede linjer.

Linjene tegnes i en avgrensingsboks rundt den angitte koordinaten og dimensjoneres slik at den faktiske størrelsen på den tegnede polygon vil være større enn det bredden og høyden indikerer.

Se også: [FillPoly](#)

## DrawRect

**DrawRect** *x, y, bredde, høyde*

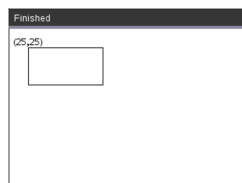
*x, y*: øvre venstre koordinat for rektangel

*bredde, høyde*: bredde og høyde for rektangel (rektangel tegn rett ned og til høyre fra startkoordinat).

**Merk:** Linjene tegnes i en avgrensingsboks rundt den angitte koordinaten og dimensjoneres slik at den faktiske størrelsen på det tegnede rektangelet vil være større enn det bredden og høyden indikerer.

Se også: [FillRect](#)

DrawRect 25,25,100,50



## DrawText

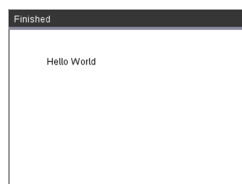
**DrawText** *x, y, exprOrString1*  
*[,exprOrString2]...*

*x, y*: koordinat for tekstutdata


Tegner teksten i *exprOrString* på den angitte *x, y*-koordinatposisjonen.

Reglene for *exprOrString* er de samme som for **Disp** – **DrawText** kan ta flere argumenter.

DrawText 50,50,«Hello World»



## FillArc

 Katalog >  CXII

**FillArc**  $x, y$ , bredde, høyde, startAngle, arcAngle

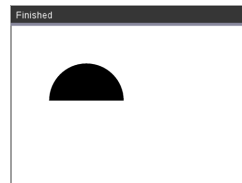
$x, y$ : øvre venstre koordinat for avgrensede rektangel

Tegn og fyll en bue i det definerte avgrensede rektangelet med de angitte start- og buevinklene.

Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommandoen

«arc angle» definerer vinkelåpningen for buen

FillArc 50,50,100,100,0,180



## FillCircle

 Katalog >  CXII

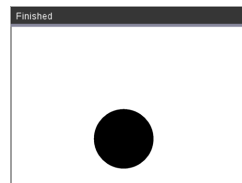
**FillCircle**  $x, y$ , radius

$x, y$ : koordinat for sentrum

Tegn og fyll en sirkel på det angitte senteret med den angitte radiusen.

Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommandoen.

FillCircle 150,150,40



Her!

## FillPoly

 Katalog >  CXII

**FillPoly**  $xlist, ylist$

eller

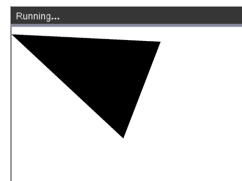
**FillPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$

**Merk:** Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

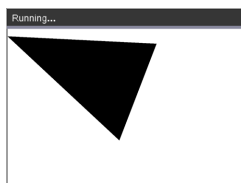
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

FillPoly  $xlist,ylist$



FillPoly 0,10,200,20,150,150,0,10

**FillRect****FillRect** *x, y, bredde, høyde**x, y*: øvre venstre koordinat for rektangel*bredde, høyde*: bredde og høyde for rektangelTegn og fyll et rektangel med øverste venstre hjørne ved koordinaten spesifisert av  $(x,y)$ Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommandoen**Merk:** Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

FillRect 25,25,100,50



## G

### getPlatform()

Katalog >   
CXII

### getPlatform()

getPlatform()

"dt"

Returnerer:

«dt» på skrivebordsprogrammer

«hh» på TI-Nspire™ CX-grafregnere

«ios» på TI-Nspire™ CX iPad®-app

**PaintBuffer**

Tegn grafikkbuffer på skjerm

Denne kommandoen brukes sammen med UseBuffer for å øke visningshastigheten på skjermen når programmet genererer flere grafiske objekter.

UseBuffer

For n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radius:=randInt(10,50)

Wait 0,5

DrawCircle x,y,radius

EndFor

PaintBuffer

Dette programmet viser alle de 10 sirklene samtidig.

Hvis kommandoen «UseBuffer» fjernes, vises hver sirkel slik den er tegnet.

Se også: [UseBuffer](#)

**PlotXY**  $x, y, form$ 

$x, y$  koordinat for plottform

*form*: skriv inn et tall mellom 1 og 13 for å spesifisere formen

- 1 – Fylt sirkel
- 2 – Tom sirkel
- 3 – Fylt firkant
- 4 – Tom firkant
- 5 – Kryss
- 6 – Pluss
- 7 – Tynn
- 8 – middels punkt, fast
- 9 – middels punkt, tomt
- 10 – større punkt, fast
- 11 – større punkt, tomt
- 12 – største punkt, fast
- 13 – største punkt, tomt

PlotXY 100,100,1

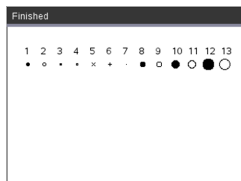


For n,1,13

DrawText 1+22\*n,40,n

PlotXY 5+22\*n,50,n

EndFor



**SetColor**
 Katalog >   
**CXII**
**SetColor**

Red-value, Green-value, Blue-value

Gyldige verdier for rød, grønn og blå er mellom 0 og 255

Angir fargen for påfølgende Tegnekommandoer

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
 Katalog >   
**CXII**
**SetPen**

tykkelse, stil

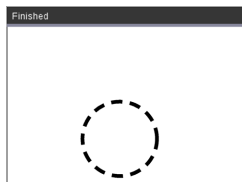
tykkelse: 1 &lt;= tykkelse &lt;= 3 | 1 er tynneste, 3 er tykkeste

stil: 1 = Jevn, 2 = Stiplet, 3 = Punkt

Angir pennestilen for påfølgende Tegnekommandoer

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
 Katalog >   
**CXII**
**SetWindow**

xMin, xMax, yMin, yMax

Etablerer et logisk vindu som settes inn på grafikktegneområdet. Alle parametere er nødvendige.

Hvis deler av et tegnet objekt er utenfor vinduet, blir utdataene klippet (ikke vist) og ingen feilmelding vises.

SetWindow 0,160,0,120

vil sette utskriftsvinduet til 0,0 i nederste venstre hjørne med en bredde på 160 og en høyde på 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

DrawLine 0,0,100,100

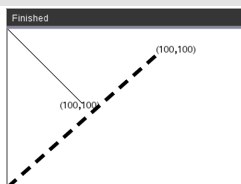


Hvis  $x_{\min}$  er større enn eller lik  $x_{\max}$  eller  $y_{\min}$  er større enn eller lik  $y_{\max}$ , vises en feilmelding.

Eventuelle objekter som er tegnet før en SetWindow-kommando, blir ikke tegnet på nytt i den nye konfigurasjonen.

For å tilbakestille vindusparametrene til standardinnstillingene, bruk:

SetWindow 0,0,0,0



**UseBuffer**

Tegn til grafikkbuffer utenfor skjermen i stedet for til skjerm (for å øke ytelsen)

Denne kommandoen brukes sammen med PaintBuffer for å øke visningshastigheten på skjermen når programmet genererer flere grafiske objekter.

Med UseBuffer vises all grafikk først etter at neste PaintBuffer-kommando er utført.

UseBuffer trenger bare å bli anropt én gang i programmet, dvs. hver bruk av PaintBuffer trenger ikke en tilsvarende UseBuffer

Se også: [PaintBuffer](#)

UseBuffer

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
radius:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,radius
```

```
EndFor
```

```
PaintBuffer
```

Dette programmet viser alle de 10 sirklene samtidig.



Hvis kommandoen «UseBuffer» fjernes, vises hver sirkel slik den er tegnet.

# Tomme (åpne) elementer

Når du analyserer reelle data, kan det hende at du ikke alltid har et komplett datasett. TI-Nspire™ tillater tomme eller åpne dataelementer, slik at du kan fortsette med data som nesten er komplette istedenfor å måtte starte på nytt eller forkaste oppgaver som ikke er fullført.

Under “*Plotte graf fra regnearkdata*” i kapitlet Lister og regneark finner du et eksempel på data som involverer tomme elementer.

Med funksjonen **delVoid()** kan du fjerne tomme elementer fra en liste. Med funksjonen **isVoid()** kan du teste for et tomt element. For detaljer, se **delVoid()**, side 41, og **isVoid()**, side 79.

**Merk:** For å legge inn et tomt element manuelt i et matematisk uttrykk, skriv “\_” eller nøkkelordet **void**. Nøkkelordet **void** konverteres automatisk til et “\_”-symbol når uttrykket blir behandlet. For å skrive inn “\_” på grafregneren, trykk på  .

## Beregninger som involverer åpne elementer

De fleste beregninger som involverer et åpent (tomt) innlegg, vil produsere et åpent (tomt) resultat. Se spesialtilfeller nedenfor.

$\_$	-
$\gcd(100, \_)$	-
$3 + \_$	-
$\{5, \_, 10\} - \{3, 6, 9\}$	$\{2, \_, 1\}$

## Listeutsagn som inneholder åpne (tomme) elementer

Følgende funksjoner og kommandoer ignorerer (hopper over) åpne (tomme) elementer som blir funnet i listeutsagn.

**count**, **countif**, **cumulativeSum**, **freqTable**→**list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumlf**, **varPop**, og **varSamp**, samt regresjonsberegninger, **OneVar**, **TwoVar** og **FiveNumSummary** statistikk, konfidensintervaller og statistikktester

$\text{sum}(\{2, \_, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, \_, \_, 3\})$	2
$\text{cumulativeSum}(\{1, 2, \_, 4, 5\})$	$\{1, 3, \_, 7, 12\}$
$\text{cumulativeSum}\left(\begin{pmatrix} 1 & 2 \\ 3 & \_ \\ 5 & 6 \end{pmatrix}\right)$	$\begin{pmatrix} 1 & 2 \\ 4 & \_ \\ 9 & 8 \end{pmatrix}$

**SortA** og **SortD** flytter alle åpne (tomme) elementer innenfor det første utsagnet til bunnen.

$\{5, 4, 3, \_, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, \_, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
$\text{SortA list1, list2}$	<i>Done</i>
$\text{list1}$	$\{1, 3, 4, 5, \_ \}$
$\text{list2}$	$\{1, 3, 4, 5, 2\}$

## Listeutsagn som inneholder åpne (tomme) elementer

I regresjoner introduserer en åpning i en X- eller Y-liste en åpning for det tilsvarende elementet i en rest.

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD list1,list2	Done
list1	$\{5,3,2,1,_\}$
list2	$\{5,3,2,1,4\}$
<hr/>	
$I1:=\{1,2,3,4,5\}; I2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx I1,I2	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 utelatt kategori i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{"M","M","F","F"\}; incl:=\{"F"\}$	$\{"F"\}$
LinRegMx I1,I2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

En frekvens på 0 i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx I1,I2,\{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	$\{1,_,1,1,1\}$

# Snarveier/hurtigtaster for å legge inn matematiske uttrykk

Hurtigtaster lar deg legge inn matematiske uttrykk ved å skrive i stedet for å bruke katalogen eller symbolpaletten. Eksempel: Når du skal legge inn uttrykket  $\sqrt{6}$ , kan du skrive `sqrt (6)` på kommandolinjen. Når du trykker på `[enter]`, endres uttrykket `sqrt (6)` til  $\sqrt{6}$ . Noen hurtigtaster kan brukes både fra kalkulatoren og fra tastaturet på datamaskinen. Andre er først og fremst nyttige fra tastaturet på datamaskinen.

## Fra kalkulatoren eller datamaskintastaturet

Hvis du skal legge inn dette:	Skriv dette:
$\pi$	<code>pi</code>
$\theta$	<code>theta</code>
$\infty$	<code>infinity</code>
$\leq$	<code>&lt;=</code>
$\geq$	<code>&gt;=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (logisk implikasjon)	<code>=&gt;</code>
$\Leftrightarrow$ (logisk dobbel implikasjon, XNOR)	<code>&lt;=&gt;</code>
$\rightarrow$ (lagre-operator)	<code>=:</code>
$   $ (absoluttverdi)	<code>abs (...)</code>
$\sqrt{()}$	<code>sqrt (...)</code>
$\Sigma()$ (Sum-sjablon)	<code>sumSeq (...)</code>
$\Pi()$ (Produkt-sjablon)	<code>prodSeq (...)</code>
$\sin^{-1}()$ , $\cos^{-1}()$ , ...	<code>arcsin (...)</code> , <code>arccos (...)</code> , ...
$\Delta\text{List}()$	<code>deltaList (...)</code>

## Fra tastaturet på datamaskinen

Hvis du skal legge inn dette:	Skriv dette:
$i$ (imaginær konstant)	<code>@i</code>
$e$ (naturlig log-grunntall $e$ )	<code>@e</code>
$E$ (vitenskapelig notasjon)	<code>@E</code>
$T$ (transponert)	<code>@t</code>

Hvis du skal legge inn dette:	Skriv dette:
$^{\circ}$ (radianer)	@r
$^{\circ}$ (grader)	@d
$\text{g}$ (gadianer)	@g
$\angle$ (vinkel)	@<
► (konvertering)	@>
►Decimal, ►approxFraction ( ), osv.	@>Decimal, @>approxFraction(), osv.

# EOS™ (Ligningsoperativsystem)-hierarkiet

Dette avsnitt beskriver Equation Operating System (ligningsoperativsystem) (EOS™) som brukes av TI-Nspire™ -teknologien for undervisning i matematikk og realfag. Tall, variabler og funksjoner legges inn i en enkel, ukomplisert sekvens. EOS™ -programvaren behandler uttrykk og ligninger ved hjelp av parentetisk gruppering og i samsvar med de prioriteringene som beskrives over.

## Rekkefølge av beregning

Nivå	Operator
1	Parentes ( ), hakeparentes [ ], buet parentes { }
2	Omregning (#)
3	Oppkalling av funksjon
4	Postoperatorer: grader-minutter-sekunder ( <sup>°</sup> , ' , " ), faktultet (!), prosent (%), radianer ( <sup>r</sup> ), senket skrift ([ ] ), transponert ( <sup>T</sup> )
5	Eksponensiering, potens-operator (^)
6	Negasjon (L)
7	Sett sammen streng (&)
8	Multiplikasjon (•), divisjon (/)
9	Addisjon (+), subtraksjon (-)
10	Likhetsrelasjoner: lik (=), ulik (≠ eller /≠), mindre enn (<), mindre enn eller lik (≤ eller <=), større enn (>), større enn eller lik (≥ eller >=)
11	Logisk <b>not</b>
12	Logisk <b>and</b>
13	Logiske <b>or</b>
14	<b>enten...eller, verken ...eller, ikke både ...og</b>
15	Logisk implikasjon (⇒)
16	Logisk dobbel implikasjon, XNOR (↔)
17	Begrensningsoperator (" ")
18	Lagre (→)

## Parenteser, hakeparenteser, buede parenteser

Først behandles alle beregninger som står i parentes, hakeparentes eller buet parentes. I uttrykket  $4(1+2)$  behandler EOS™ -programvaren for eksempel først den delen av uttrykket som står i parenteser,  $1+2$ , og multipliserer deretter resultatet, 3, med 4.

Antallet åpne- og lukkeparenteser, åpne- og lukke-hakeparenteser og buede åpne- og lukkeparenteser må være det samme innenfor ett uttrykk eller én ligning. Hvis ikke,

vises en feilmelding, som angir det manglende elementet. For eksempel vil  $(1+2)/(3+4)$  vise feilmeldingen "Mangler )."

**Merk:** Siden TI-Nspire™ -programvaren gjør at du kan definere dine egne funksjoner, blir et variabelnavn fulgt av et uttrykk i parentes betraktet som en "oppkalling av funksjon" istedenfor halvveis skjult multiplikasjon. For eksempel i  $a(b+c)$  blir funksjonen a beregnet for verdien  $b+c$  (av den variable). For å multiplisere uttrykket  $b+c$  med variabelen a, må du bruke eksplisitt multiplikasjon:  $a*(b+c)$ .

## Omregning

Omregnings-operatoren (#) omregner en streng til et variabel- eller funksjonsnavn. For eksempel oppretter #("x"&"y"&"z") variabelnavnet xyz. Omregning lar deg også opprette og modifisere en variabel mens du er inne i et program. Hvis for eksempel  $10 \rightarrow r$  og " $r$ "  $\rightarrow s1$ , så er  $\#s1=10$ .

## Postoperatorer

En postoperator er en operator som kommer direkte etter et argument, som f.eks. 5!, 25%, eller  $60^\circ 15' 45''$ . Et argument som er fulgt av en postoperator blir behandlet ved fjerde prioritetsnivå. I uttrykket  $4^3!$  blir for eksempel 3! behandlet først. Resultatet, 6, blir så eksponenten av 4 for å oppnå 4096.

## Eksponensiering

Eksponensiering (^) og element-for-element-eksponensiering (.^ ) blir behandlet fra høyre til venstre. Uttrykket  $2^3^2$  blir for eksempel behandlet som det samme som  $2^4$  ( $3^2$ ) for å produsere 512. Dette er forskjellig fra  $(2^3)^2$ , som er 64.

## Negasjon

For å legge inn et negativt tall, trykk på  $(-)$  fulgt av tallet. Postoperasjoner og eksponensiering utføres før negasjon. Resultatet av  $-x^2$  er for eksempel et negativt tall, og  $-9^2 = -81$ . Bruk parenteser for å opphøye et negativt tall i annen potens, som f.eks.  $(-9)^2$  for å produsere 81.

## Begrensning ("|")

Argumentet som følger etter ("|")-operator gir et sett av begrensninger som påvirker hvordan argumentet som står foran operatoren blir behandlet.



# TI-Nspire CX II – TI-Basic programmeringsfunksjoner

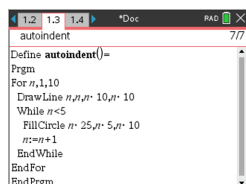
## Auto-innrykk i Programmeringseditor

TI-Nspire™ progeditor gir automatisk innrykking av utsagn i en blokkkommando.

Blokkkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Programeditor vil automatisk legge til mellomrom foran programkommandoer i en blokkkommando. Den avsluttende kommandoen til blokken blir innrettet likt med åpningskommandoer.

Eksemplet nedenfor viser automatisk innrykk i nestede blokkkommandoer.



```
autoindent      77
Define autoindent()=
Prgm
For n,1,10
  DrawLine n,n,n-10,n-10
  While n<5
    FillCircle n-25,n-5,n-10
    n:=n+1
  EndWhile
EndFor
EndPrgm
```

Kodefragmenter som kopieres og limes inn, beholder originalinnrykket.

Åpning av et program som er opprettet i en tidligere versjon av programvaren, beholder originalinnrykket.

---

## Forbedrede feilmeldinger for TI-Basic

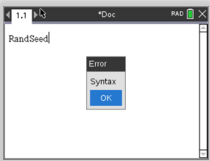
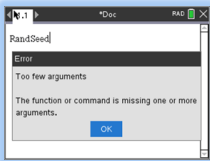
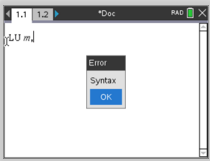
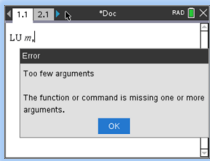
### Feil

Feiltilstand	Ny melding
Feil i betinget utsagn (If/While)	Et betinget utsagn ble ikke løst til <b>SANN</b> eller <b>USANN</b> <b>MERK:</b> Etter endringen for å plassere markøren på linjen med feilen, trenger vi ikke lenger å angi om feilen er i et «If»-utsagn eller i et «While»-utsagn.
Mangler EndIf	Forventet <b>EndIf</b> , men fant en annen end-setning
Mangler EndFor	Forventet <b>EndFor</b> , men fant en annen end-setning
Mangler EndWhile	Forventet <b>EndWhile</b> , men fant en annen end-setning
Mangler EndLoop	Forventet <b>EndLoop</b> , men fant en annen end-setning

Feiltilstand	Ny melding
Mangler <b>EndTry</b>	Forventet <b>EndTry</b> , men fant en annen end-setning
« <b>Then</b> » utelatt etter <b>If</b> <condition>	Mangler <b>If..Then</b>
« <b>Then</b> » utelatt etter <b>Elseif</b> <condition>	<b>Then</b> mangler i blokken: <b>Elseif</b> .
Når « <b>Then</b> », « <b>Else</b> » og « <b>Elseif</b> » ble støtt på utenfor kontrollblokkene	<b>Else</b> er ugyldig utenfor blokkene: <b>If..Then..EndIf</b> eller <b>Try..EndTry</b>
« <b>Elseif</b> » vises utenfor « <b>If..Then..EndIf</b> »-blokken	<b>Elseif</b> er ugyldig utenfor blokk: <b>If..Then..EndIf</b>
« <b>Then</b> » vises utenfor « <b>If....EndIf</b> »-blokken	<b>Then</b> er ugyldig utenfor blokkene: <b>If..EndIf</b>

## Syntaksfeil

Hvis kommandoer som forventer ett eller flere argumenter blir oppkalt med en ufullstendig liste over argumenter, vil en «**For få argumenter-feil**» vises istedenfor for «**syntaks**»-feil

Gjeldende atferd	Ny CX II-atferd
 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'RasdSeed'. A small error dialog box is displayed with the text 'Error', 'Syntax', and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'RasdSeed'. A larger error dialog box is displayed with the text 'Error', 'Too few arguments', 'The function or command is missing one or more arguments.', and an 'OK' button.</p>
 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'LU m'. A small error dialog box is displayed with the text 'Error', 'Syntax', and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'LU m'. A larger error dialog box is displayed with the text 'Error', 'Too few arguments', 'The function or command is missing one or more arguments.', and an 'OK' button.</p>



## Konstanter og verdier

Den følgende tabellen inneholder konstanter og deres verdier, som er tilgjengelige når du utfører enhetsomregninger. De kan skrives inn manuelt eller velges fra listen

**Konstanter i Verktøy > Enhetsomregninger** (Håndholdt enhet: trykk på  3).

Konstant	Navn	Verdi
_c	Lysets hastighet	299792458 _m/_s
_Cc	Coulomb-konstant	8987551787.3682 _m/_F
_Fc	Faraday-konstant	96485,33289 _coul/_mol
_g	Gravitasjonens akselerasjon	9,80665 _m/_s <sup>2</sup>
_Gc	Gravitasjonskonstant	6,67408E-11 _m <sup>3</sup> /_kg/_s <sup>2</sup>
_h	Plancks konstant	6,626070040E-34 _J _s
_k	Boltzmanns konstant	1,38064852E-23 _J/_°K
_μ0	Permeabilitet i vakuum	1,2566370614359E-6 _N/_A <sup>2</sup>
_μb	Bohr-magneton	9,274009994E-24 _J _m <sup>2</sup> /_Wb
_Me	Elektronets hvilemasse	9,10938356E-31 _kg
_Mμ	Myon-masse	1,883531594E-28 _kg
_Mn	Nøytronets hvilemasse	1,674927471E-27 _kg
_Mp	Protonets hvilemasse	1,672621898E-27 _kg
_Na	Avogadros tall	6,022140857E23 /_mol
_q	Elektronladning	1,6021766208E-19 _coul
_Rb	Bohr-radius	5,2917721067E-11 _m
_Rc	Molar gasskonstant	8,3144598 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731,568508/_m
_Re	Elektron-radius	2,8179403227E-15 _m
_u	Atommasse	1,660539040E-27 _kg
_Vm	Molarvolum	2,2413962E-2 _m <sup>3</sup> /_mol
_ε0	Permittivitet i vakuum	8,8541878176204E-12 _F/_m
_σ	Stefan-Boltzmann-konstant	5,670367E-8 _W/_m <sup>2</sup> /_°K <sup>4</sup>
_φ0	Magnetisk flukskvantum	2,067833831E-15 _Wb

## Feilkoder og feilmeldinger

Hvis det oppstår en feil, er koden knyttet til variabel *feilKode*. Egendefinerte programmer og funksjoner kan undersøke *feilKode* for å bestemme årsaken til feilen. For et eksempel på bruk av *feilKode*, se eksempel 2 under kommandoen **Prøv**, side 166.

**Merk:** Noen feilforhold gjelder kun for TI-Nspire™ CAS-produktene, og noen gjelder kun for TI-Nspire™-produktene.

Feilkode	Beskrivelse
10	En funksjon returnerte ingen verdi
20	En test ga ikke resultatet SANN eller USANN.  Vanligvis kan udefinerte variabler ikke sammenliknes. Testen <code>If a&lt;b</code> vil for eksempel forårsake enten at <code>a</code> eller at <code>b</code> ikke er definert, dersom utsagnet <code>If</code> blir utført.
30	Argumentet kan ikke være et mappenavn.
40	Argumentfeil
50	Uoverensstemmelse i argument  To eller flere argumenter må være av samme type.
60	Argumentet må være et Boolsk uttrykk eller et heltall
70	Argumentet må være et desimaltall
90	Argumentet må være en liste
100	Argumentet må være en matrise
130	Argumentet må være en streng
140	Argumentet må være et variabelnavn.  Pass på at navnet: <ul style="list-style-type: none"><li>• ikke begynner med et tall</li><li>• ikke inneholder mellomrom eller spesialtegn</li><li>• ikke bruker senket strek eller punktum på ugyldig måte</li><li>• ikke overgår tillatt lengde</li></ul> Les mer om dette i Kalkulator-avsnittet dokumentasjonen.
160	Argumentet må være et uttrykk
165	For lite strøm i batteriene til å sende/motta  Legg i nye batterier før sending eller mottak.
170	Grense  Den nedre grensen må være mindre enn den øvre grensen for å definere søkeintervallet.

Feilkode	Beskrivelse
180	<p>Avbryt</p> <p>Det ble trykket på tasten <code>esc</code> eller <code>on</code> under en lang beregning eller mens et program ble utført.</p>
190	<p>Sirkulær definisjon</p> <p>Denne meldingen komme til syne for å unngå at du slipper opp for minne under uendelig erstating av variable verdier. For eksempel vil <math>a+1 \rightarrow a</math>, der a er en udefinert variabel, forårsake denne feilen.</p>
200	<p>Ugyldig begrensingsuttrykk</p> <p>For eksempel vil <math>\text{løs}(3x^2-4=0,x) \mid x &lt; 0 \text{ eller } x &gt; 5</math> produsere denne feilmeldingen, fordi begrensningen er skilt med "eller" istedenfor "og".</p>
210	<p>Ugyldig datatype</p> <p>Et argument er av feil datatype.</p>
220	Avhengig grense
230	<p>Dimensjon</p> <p>En liste eller matriseindeks er ikke gyldig. Hvis for eksempel listen <math>\{1,2,3,4\}</math> er lagret i L1, så er L1[5] en dimensjonsfeil, fordi L1 kun inneholder fire elementer.</p>
235	Dimensjonsfeil. Ikke nok elementer i listene.
240	<p>Dimensjonsfeil</p> <p>To eller flere argumenter må være av samme dimensjon. For eksempel er <math>[1,2]+[1,2,3]</math> en dimensjonsfeil, fordi matrisene inneholder ulikt antall elementer.</p>
250	Divisjon med null
260	<p>Grunnmengdefeil</p> <p>Et argument må være i en spesifisert grunnmengde. For eksempel er <math>\text{tilf}(0)</math> ikke gyldig.</p>
270	Duplikatnavn på variabel
280	Else og Elseif ugyldig utenfor If...EndIf-blokk
290	EndTry uten tilhørende Else-uttrykk
295	For mange iterasjoner
300	Forventet 2- eller 3-elements liste eller matrise
310	Det første argumentet av nSolve må være en ligning i én variabel. Det kan ikke inneholde noen annen variabel enn den variabelen som vi er interessert i.
320	<p>Det første argumentet til løs eller kLøs må være en ligning eller ulikhet</p> <p>For eksempel er <math>\text{løs}(3x-4,x)</math> ugyldig fordi det første argumentet ikke er en ligning.</p>

<b>Feilkode</b>	<b>Beskrivelse</b>
345	Inkonsistente enheter
350	Indeks utenfor gyldig område
360	Indireksjonsstrengen er ikke et gyldig variabelnavn
380	Udefinert Svar  Enten opprettet ikke den forrige beregningen noe Svar, eller det ble ikke lagt inn noe forrige beregning.
390	Ugyldig tildeling
400	Ugyldig tildelingsverdi
410	Ugyldig kommando
430	Ugyldig for de gjeldende modusinnstillingene
435	Ugyldig gjetning (startverdi)
440	Ugyldig "skjult" multiplikasjon  For eksempel er $x(x+1)$ ugyldig; derimot er $x*(x+1)$ korrekt syntaks. Dette skal forhindre forvirring mellom halvveis skjult multiplikasjon og oppkalling av funksjon.
450	Ugyldig i en funksjon eller gjeldende uttrykk  Det er kun visse kommandoer som er gyldige i en egendefinert funksjon.
490	Ugyldig i Try..EndTry-blokk
510	Ugyldig liste eller matrise
550	Ugyldig utenfor funksjon eller program  Et antall kommandoer er ikke gyldige utenfor en funksjon eller et program. For eksempel kan ikke Lokal brukes hvis den ikke er inne i en funksjon eller et program.
560	Ugyldig utenfor Loop..EndLoop-, For..EndFor- eller While..EndWhile-blokk  For eksempel er Avslutt-kommandoen kun gyldig inne i disse loop-blokkene.
565	Ugyldig utenfor program
570	Ugyldig banenavn  For eksempel er \var ugyldig.
575	Ugyldig polar kompleks verdi
580	Ugyldig programreferanse  Det kan ikke refereres til programmer inne i funksjoner eller uttrykk, som f.eks. $1+p(x)$ , der p er et program.

Feilkode	Beskrivelse
600	Ugyldig tabell
605	Ugyldig bruk av enheter
610	Ugyldig variabelnavn i Lokalt utsagn
620	Ugyldig variabel- eller funksjonsnavn
630	Ugyldig variabelreferanse
640	Ugyldig vektorsyntaks
650	Kommunikasjons-forbindelse  En kommunikasjon mellom to enheter er ikke fullført. Kontroller at forbindelseskabelen er koplet godt til i begge ender.
665	Matrisen kan ikke diagonaliseres
670	Lite minne  1. Slett noen data i dette dokumentet  2. Lagre og lukk dette dokumentet  Dersom 1 og 2 mislykkes, ta ut batteriene og sett dem inn igjen
672	Ressursbegrensning
673	Ressursbegrensning
680	Manglende (
690	Manglende )
700	Manglende “
710	Manglende ]
720	Manglende }
730	Manglende start eller slutt på bloksyntaks
740	Manglende Then i If..EndIf-blokken
750	Navnet er ikke en funksjon eller et program
765	Ingen funksjoner er valgt
780	Fant ingen løsning
800	Ikke-reelt resultat  Hvis for eksempel programvaren er i Reell innstilling, er $\sqrt{-1}$ ugyldig.  For å tillate komplekse resultater, endre “Reell eller Kompleks” modusinnstilling til REKTANGULÆR eller POLAR.



Feilkode	Beskrivelse
830	Overflyt
850	Fant ikke programmet Det ble ikke funnet en programreferanse i et annet program i oppgitt bane under utføring.
855	Rand-funksjonstyper ikke tillatt i grafer
860	Rekursjonen for dyp
870	Reservert navn eller systemvariabel
900	Argumentfeil Median-median-modell kunne ikke brukes på datasettet.
910	Syntaksfeil
920	Fant ikke teksten
930	For få argumenter Funksjonen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Uttrykket eller ligningen inneholder for mange argumenter og kan ikke behandles.
950	For mange indekser
955	For mange udefinerte variabler
960	Variabelen er ikke definert Variabelen er ikke tildelt noen verdi. Bruk en av følgende kommandoer: <ul style="list-style-type: none"> <li>• <code>sto →</code></li> <li>• <code>:=</code></li> <li>• <b>Define</b></li> </ul> for å tildele variablene verdi.
965	Ulisensiert OS
970	Variabel er i bruk, så referanser eller endringer er ikke tillatt
980	Variabel er beskyttet
990	Ugyldig variabelnavn Pass på at navnet ikke overgår tillatt lengde
1000	Grunnmengde for vindusvariabel
1010	Zoom

Feilkode	Beskrivelse
1020	Intern feil
1030	Overtredelse av beskyttet minne
1040	Ustøttet funksjon. Denne funksjonen krever Computer Algebra System. Prøv TI-Nspire CAS.
1045	Ustøttet operator. Denne operatoren krever Computer Algebra System. Prøv TI-Nspire CAS.
1050	Ustøttet egenskap. Denne operatoren krever Computer Algebra System. Prøv TI-Nspire CAS.
1060	Innlagt argument må være numerisk. Bare innlegg som inneholder numeriske verdier er tillatt.
1070	Trig-funksjonsargument for stort for nøyaktig reduksjon
1080	Ustøttet bruk av Svar. Denne applikasjonen støtter ikke Svar.
1090	Funksjonen er ikke definert. Bruk en av følgende kommandoer: <ul style="list-style-type: none"> <li>• <b>Define</b></li> <li>• :=</li> <li>• sto →</li> </ul> for å definere en funksjon.
1100	Ikke-reell beregning  Hvis for eksempel programvaren er i Reell innstilling, er $\sqrt{-1}$ ugyldig.  For å tillate komplekse resultater, endre "Reell eller Kompleks" modusinnstilling til REKTANGULÆR eller POLAR.
1110	Ugyldige grenser
1120	Tegn ikke endret
1130	Argumentet kan ikke være en liste eller matrise
1140	Argumentfeil  Det første argumentet må være et polynomisk uttrykk i det andre argumentet. Dersom det andre argumentet utelates, prøver programvaren å velge en grunninnstilling.
1150	Argumentfeil  De første to argumentene må være polynomiske uttrykk i det tredje argumentet. Dersom det tredje argumentet utelates, prøver programvaren å velge en grunninnstilling.
1160	Ugyldig banenavn for bibliotek  Et banenavn må være av formen xxx\yyy, der: <ul style="list-style-type: none"> <li>• Delen xxx kan bestå av mellom 1 og 16 tegn.</li> </ul>

Feilkode	Beskrivelse
	<ul style="list-style-type: none"> <li>Delen <code>yyy</code> kan ha 1 til 15 tegn.</li> </ul> <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1170	<p>Ugyldig bruk av banenavn for bibliotek</p> <ul style="list-style-type: none"> <li>En verdi kan ikke tildeles et banenavn som bruker <b>Define</b>, <code>:=</code> eller <code>sto</code> →.</li> <li>Et banenavn kan ikke erklæres som en lokal variabel eller brukes som parameter i en funksjonsdefinisjon eller programdefinisjon.</li> </ul>
1180	<p>Ugyldig variabelnavn på bibliotek</p> <p>Pass på at navnet:</p> <ul style="list-style-type: none"> <li>Ikke inneholder punktum</li> <li>Ikke begynner med senket strek</li> <li>Ikke består av mer enn 15 tegn</li> </ul> <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1190	<p>Bibliotek-dokumentet ble ikke funnet:</p> <ul style="list-style-type: none"> <li>Kontroller om biblioteket er i mappen <code>MittBibl</code>.</li> <li>Oppdater biblioteker.</li> </ul> <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1200	<p>Bibliotek-variabler ble ikke funnet:</p> <ul style="list-style-type: none"> <li>Kontroller om bibliotek-variablene eksisterer i den første oppgaven i biblioteket.</li> <li>Forsikre deg om at bibliotek-variabelen er blitt definert som <code>BiblOff</code> eller <code>BiblPriv</code>.</li> <li>Oppdater biblioteker.</li> </ul> <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1210	<p>Ugyldig navn på snarvei til bibliotek.</p> <p>Pass på at navnet:</p> <ul style="list-style-type: none"> <li>Ikke inneholder punktum</li> <li>Ikke begynner med senket strek</li> <li>Ikke består av mer enn 16 tegn</li> <li>Ikke er et reservert navn</li> </ul> <p>Les mer om dette i Bibliotek-avsnittet i dokumentasjonen.</p>
1220	<p>Grunnmengdefeil:</p> <p>Funksjonene <code>tangentLinje</code> og <code>normalLinje</code> støtter kun funksjoner med reelle verdier.</p>
1230	<p>Grunnmengdefeil.</p>

Feilkode	Beskrivelse
	Trigonometriske omregningsoperatorer støttes ikke i Grader- eller Gradianvinkelmodus.
1250	Argumentfeil  Bruk et system av lineære ligninger.  Eksempel på et system av to lineære ligninger med variablene x og y:  $3x+7y=5$  $2y-5x=-1$
1260	Argumentfeil:  Det første argumentet til nfMin eller nfMax må være et uttrykk i én variabel. Det kan ikke inneholde noen annen variabel enn den variabelen som vi er interessert i.
1270	Argumentfeil  Den deriverte må være av orden 1 eller 2.
1280	Argumentfeil  Bruk et polynom på utvidet (ekspandert) form i én variabel.
1290	Argumentfeil  Bruk et polynom i én variabel.
1300	Argumentfeil  Koeffisientene i polynomet må være numeriske verdier.
1310	Argumentfeil:  En funksjon kan ikke behandles for ett eller flere av dens argumenter.
1380	Argumentfeil:  Nestede oppringinger til område() funksjon er ikke tillatt.

## Advarselskoder og -meldinger

Du kan bruke funksjonen **warnCodes()** for å lagre advarselskodene som ble generert da et uttrykk ble behandlet. Denne tabellen opplyster hver numeriske varselkode og dens assosierte melding. Se **warnCodes()** for et eksempel på lagring av advarselskoder, side 175.

VARSELKODE	Melding
10000	Kommandoen kan gi falske løsninger. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10001	Derivasjon av en ligning kan gi en ugyldig ligning.
10002	Tvilsom løsning Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10003	Tvilsom nøyaktighet Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10004	Kommandoen kan utelate løsninger. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10005	KLøs kan spesifisere flere nullpunkter.
10006	Løs kan spesifisere flere nullpunkter. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10007	Flere løsninger kan eksistere. Prøv å angi passende øvre og nedre grenser, og/eller en gjetning. Eksempler ved bruk av solve(): <ul style="list-style-type: none"><li>• solve(Ligning, Var=Forslag)   nedGrens&lt;Var&lt;øvreGrens</li><li>• solve(Ligning, Var)   nedGrens&lt;Var&lt;øvreGrens</li><li>• solve(Ligning,Var=Forslag)</li></ul> Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10008	Grunnmengden til resultatet kan være mindre enn grunnmengden til innlegget (inndata).
10009	Grunnmengden til resultatet kan være større enn grunnmengden til innlegget (inndata).
10012	Ikke-reell beregning
10013	$\infty^0$ eller $\text{undef}^0$ erstattet av 1
10014	$\text{undef}^0$ erstattet av 1

<b>VARSELKODE</b>	<b>Melding</b>
10015	$1^\infty$ eller $1^{\text{undef}}$ erstattet av 1
10016	$1^{\text{undef}}$ erstattet av 1
10017	Overløp erstattet av $\infty$ eller $-\infty$
10018	Kommando krever og returnerer 64-biters verdi.
10019	Ressursbegrensning, forenkling kanskje ufullstendig.
10020	Trig-funksjonsargument for stort for nøyaktig reduksjon.
10021	Inndataene inneholder en udefinert parameter. Resultatet kanskje ikke gyldig for alle mulige parameterverdier.
10022	Å spesifisere riktig nedre og øvre grense kan gi en løsning.
10023	Skalar har blitt multiplisert med identitetsmatrisen.
10024	Resultat oppnådd med tilnærmet aritmetikk.
10025	Ekvivalens kan ikke verifiseres i EXACT-modus.
10026	Begrensningen kan bli ignorert. Spesifiser begrensning i skjemaet «Variable MathTestSymbol Constant» eller et konjunkt (en kombinasjon) av disse formene, f.eks. 'x<3 and x>-12'

## Generell informasjon

### **Hjelp på nettet (online)**

[education.ti.com/eguide](http://education.ti.com/eguide)

Velg ditt land for mer produktinformasjon.

### **Kontakt TIs brukerstøtte**

[education.ti.com/ti-cares](http://education.ti.com/ti-cares)

Velg ditt land for tekniske og andre støtteressurser.

### **Service og garantiinformasjoner**

[education.ti.com/warranty](http://education.ti.com/warranty)

Velg landet ditt for informasjon om lengden og vilkårene for garantien eller om produkttjenester.

Begrenset garanti. Denne garantien påvirker ikke dine lovmessige rettigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

# Stikkordregister

		, begrensingsoperator .....	204
	'		
'	, fremstilling minutter .....	+	
		+ , addere .....	184
	-	/	
-	, subtrahere[*] .....	/, dividere[*] .....	186
	!	=	
!	, faktultet .....	≠ , ulik[*] .....	191
	"	=, er lik .....	190
"	, fremstilling sekunder .....	>	
	#	>, større enn .....	193
#	, Indir.ref .....	∏	
#	, omregnings-operator .....	∏, produkt, sjablon for .....	5
	%	∏, produkt[*] .....	197
%	, prosent .....	∑	
	&	∑( ), sum[*] .....	197
&	, legg til .....	∑Int( ) .....	198
	*	∑Prn( ) .....	199
*	, multiplisere .....	√	
	.	√, kvadratrott[*] .....	196
.-	, prikk subtraksjon .....	∫	
.*	, prikk multiplikasjon .....	∫, integral[*] .....	196
./	, prikk divisjon .....	≤	
.^	, prikk potens .....	≤ , mindre enn eller lik .....	192
.+	, prikk addisjon .....	≥	
	:	≥ , større enn eller lik med .....	193
:=	, tildele .....	►	
	^	►Cylind, vise som sylindrisk vektor [Sy]ind] .....	36
^	, potens .....	►Polar, vise som polar vektor[Pol]ar] .....	117





<b>B</b>			
begrensningsoperator " "	204	cosinus, cos( )	27
begrensningsoperator, rekkefølge av beregning	227	cot <sup>-1</sup> ( ), invers cotangens	30
behandle polynom, polyEval( )	118	cot( ), cotangens	30
behandling, rekkefølge av	227	cotangens, cot( )	30
bestemt integral		coth <sup>-1</sup> ( ), hyperbolsk invers cotangens	31
sjablon for	6	cPolyRoots()	32
bibliotek		crossP( ), kryssprodukt	33
lage snarveier til objekter	81	csc <sup>-1</sup> ( ), invers cosekans	33
BiblOff	39	csc( ), cosekans	33
BiblPriv	39	csch <sup>-1</sup> ( ), invers hyperbolsk cosekans	34
binær		csch( ), hyperbolsk cosekans	34
indikator, Ob	207	CubicReg, kubisk regresjon	34
vise, ►Grunntall2	17	cumulativeSum( ), kumulativ sum	35
binomCdf( )	19, 77	cycle, Løkke	36
binomPdf( )	20	Cycle, løkke	36
Boolske operatører		<b>D</b>	
⇒	194, 225	d( ), første deriverte	195
⇔	194	dager mellom datoer, dbd( )	37
and	8	dbd( ), dager mellom datoer	37
eller	113	Define, definer	38
enten ... eller ...	176	Definer	38
ikke	109	Definer BiblOff	39
ikke både...og	103	Definer BiblPriv	39
verken ... eller	108	definer, Define	38
brøker		definere	
ekteBrøk	121	felles (offentlig) funksjon eller program	39
sjablon for	1	privat funksjon eller program	39
<b>C</b>		deltaList()	40
c 2 2-veis	21	delVar, slett variabel	40
c 2 Pdf( )	23	delVoid( ), fjern gamle elementer	41
Cdf( )	54	deriverte	
ceiling( ), øvre	20	første deriverte, d( )	195
centralDiff( )	20	numerisk derivert, nDeriv( )	106-107
char( ), tegnstring	21	numerisk derivert, nDerivative( )	105
ClearAZ	23	desimal	
colDim( ), matrisens kolonnedimensjon	24	heltall vise, 4Grunntall10	18
colNorm( ), matrisens kolonnenorm	24	vinkel-visning, ►DD	37
conj( ), kompleks konjugert	25	det( ), matrisedeterminant	41
constructMat( ), konstruer matrise	25	diag( ), matrisediagonal	42
cos <sup>-1</sup> , invers cosinus	28	dim( ), dimensjon	42
cos( ), cosinus	27	dimensjon, dim( )	42
cosh <sup>-1</sup> ( ), hyperbolsk, invers cosinus	29	DispAt	43
cosh( ), hyperbolsk cosinus	29	dividere heltall, intDiv( )	75
		dividere, /	186

dotP( ), prikk produkt ..... 45

## E

e eksponent  
sjablon for ..... 2

e i en potens,  $e^{\wedge}()$  ..... 46, 52

E, eksponent ..... 200

$e^{\wedge}()$ , e i en potens ..... 46

eff ), omregn nominell til effektiv  
rente ..... 46

effektiv rente, eff( ) ..... 46

egendefinerte funksjoner ..... 38

egendefinerte funksjoner og  
programmer ..... 39

egenvektor, eigVc( ) ..... 47

egenverdi, eigVl( ) ..... 47

eigVc( ), egenvektor ..... 47

eigVl( ), egenverdi ..... 47

ekskludering med "|" operator ..... 204

eksponensiell regresjon, ExpReg ..... 52

eksponent, E ..... 200

eksponenter  
sjablon for ..... 1

ekte brøk, propFrac ..... 121

eliminasjonsform, ref( ) ..... 129

eller (Boolsk), eller ..... 113

eller, Boolsk operator ..... 113

else if, ElseIf ..... 48

else, Else ..... 72

Elseif, else if ..... 48

en-variabel-statistikk, OneVar ..... 112

end  
For...EndFor ..... 56

if, EndIf ..... 72

stigningstall, EndLoop ..... 93

while, EndWhile ..... 176

end if, EndIf ..... 72

end stigningstall, EndLoop ..... 93

end while, EndWhile ..... 176

endfunksjon, EndFunc ..... 60

Endret internrente av retur, mirr( ), ..... 99

EndTry, avslutt prøv ..... 166

EndWhile, end while ..... 176

enhetsvektor, unitV( ) ..... 172

enten ... eller ..., Boolsk eksklusiv  
eller ..... 176

EOS (Equation Operating System) .. 227

Equation Operating System ..... 227

(Ligningsoperativsystem)

(EOS) ..... 190

er lik, = ..... 190

erstatning med "|" operator ..... 204

etikettNavn, Lbl ..... 79

euler( ), Euler function ..... 49

Exit, avslutt ..... 51

exp( ), e i en potens ..... 52

expr( ), streng til uttrykk ..... 52

ExpReg, eksponensiell regresjon .... 52

## F

factor( ), faktor ..... 54

faktor, factor( ) ..... 54

fakultet, ! ..... 195

feil og problemløsning  
send feil, SendFeil ..... 116

slett feil, SlettFeil ..... 23

feilkoder og meldinger ..... 241

Fill, matrise fyller ..... 55

finansielle funksjoner, tvnMV( ) ..... 169

finansielle funksjoner, tvml( ) ..... 169

finansielle funksjoner, tvnN( ) ..... 169

finansielle funksjoner, tvnPmt( ) ..... 169

finansielle funksjoner, tvnPV( ) ..... 170

FiveNumSummary ..... 55

fjerdegrads regresjon, QuartReg ..... 123

fjern  
åpne elementer fra liste ..... 41

floor( ), nedre ..... 56

For ..... 56

for, For ..... 56

For, for ..... 56

fordelingsfunksjoner

binomCdf( ) ..... 19, 77

binomPdf( ) ..... 20

c 2 Pdf( ) ..... 23

c22-veis( ) ..... 21

invNorm( ) ..... 77

invt( ) ..... 78

normCdf( ) ..... 109

normPdf( ) ..... 109

poissCdf( ) ..... 116

poissPdf( ) ..... 117

tCdf( ) ..... 163

tPdf( ) ..... 165

$\chi^2$ Cdf( ) ..... 22

$\chi^2$ GOF( ) ..... 22



interpolere( ), interpolere .....	75	left( ), venstre .....	80
invers cosinus, $\cos^{-1}()$ .....	28	legg til, & .....	195
invers kumulativ normalfordeling (invNorm( ) .....	77	lengde på streng .....	42
invers sinus, $\sin()$ .....	149	les modus, lesModus( ) .....	144
invers tangens, $\tan^{-1}()$ .....	161	les/returner lesNevner, getDenom( ) .....	63
invers, $x^{-1}$ .....	204	lesTeller, getNum( ) .....	69
invF( ) .....	76	lesModus( ), les modus .....	144
invNorm( ), invers kumulativ normalfordeling) .....	77	lesModus( ), les modus-innstillinger	68
invt( ) .....	78	libShortcut( ), lage snarveier til bibliotekobjekter .....	81
$\text{Inv}\chi^2()$ .....	76	ligningssystemer (2-ligning) sjablon for .....	3
iPart( ), heltallsdel .....	78	ligningssystemer (N-ligning) sjablon for .....	3
irr( ), internrente internrente, irr( ) .....	78	lineær regresjon, LinRegAx .....	82
isPrime( ), primtest .....	79	lineær regresjon, LinRegBx .....	81, 83
isVoid( ), test for tomrom .....	79	LinRegBx, lineær regresjon .....	81
<b>K</b>			
kolUtvid .....	24	LinRegMx, lineær regresjon .....	82
kombinasjoner, nCr( ) .....	104	LinRegtIntervals, lineær regresjon ..	83
Kommandoen Wait .....	174	LinRegtTest .....	85
kommentar, © .....	206	linSolve() .....	86
kompleks konjugert, conj( ) .....	25	list►mat( ), liste til matrise .....	87
konstruer matrise, constructMat( ) ..	25	liste til matrise, list►mat( ) .....	87
kopiere variabel eller funksjoner, CopyVar .....	25	liste, antall betingede elementer i ..	32
korrelasjonsmatrise, corrMat( ) .....	26	liste, antall elementer i .....	31
korrMat( ), korrelasjonsmatrise .....	26	lister differens, @liste( ) .....	87
kryssprodukt, crossP( ) .....	33	differenser i en liste, @list( ) ...	87
kubisk regresjon, CubicReg .....	34	kryssprodukt, crossP( ) .....	33
kumulativ sum, cumulativeSum( ) ..	35	kumulativ sum, cumulativeSum( ) .....	35
kvadratisk regresjon, QuadReg .....	122	liste til matrise, list►mat( ) .....	87
kvadratrot sjablon for .....	1	maksimum, max( ) .....	95
kvadratrot, $\sqrt{}$ .....	196	matrise til liste, mat $\Delta$ liste( ) ...	95
kvadratrot, $\sqrt{\phantom{x}}$ .....	154	midtstreng, mid( ) .....	98
<b>L</b>			
lagre symbol, & .....	205-206	minimum, min( ) .....	98
Lås, lås variabel eller variabelgruppe	90	nye, newList( ) .....	106
låse opp variabler og variabelgrupper .....	173	prikk produkt, dotP( ) .....	45
låse variabler og variabelgrupper ...	90	produkt, product( ) .....	120
Lbl, etikettNavn .....	79	sorter fallende, SortD .....	153
lcm, minste felles multiplum .....	80	sorter stigende, SortA .....	152
		summering, sum( ) .....	158-159
		tomme elementer i .....	223
		utvid/sett sammen, utvid( ) ....	14
		ln( ), naturlig logaritme .....	87
		LnReg, logaritmisk regresjon .....	88
		Local, lokal variabel .....	89

Logaritme		(lower-upper) .....	
sjablon for .....	2	nye, newMat( ) .....	106
logaritmer .....	87	prikk addisjon, .+ .....	188
logaritmisk regresjon, LnReg .....	88	prikk divisjon, .P .....	189
logisk dobbel implikasjon, $\Leftrightarrow$ .....	194	prikk multiplikasjon, .* .....	189
logisk implikasjon, $\Rightarrow$ .....	194, 225	prikk potens, .^ .....	189
Logistic, logistisk regresjon .....	91	prikk subtraksjon, .N .....	188
LogisticD, logistisk regresjon .....	92	produkt, product( ) .....	120
logistisk regresjon, Logistic .....	91	QR faktorisering, QR .....	121
logistisk regresjon, LogisticD .....	92	radaddisjon, rowAdd( ) .....	138
lokal variabel, Local .....	89	raddimensjon, rowDim( ) .....	138
lokal, Local .....	89	radhandling, mRow( ) .....	100
Loop, stigningstall .....	93	radmultiplikasjon og addisjon, mRowAdd( ) .....	100
LU (lower-upper), matrisens nedre- øvre dekomposisjon .....	94	radnorm, rowNorm( ) .....	139
		radskift, rowSwap( ) .....	139
		redusert eliminasjonsform, rref( ) .....	139
		summering, sum( ) .....	158-159
		tilfeldig, tilfMat( ) .....	127
		transponert, T .....	160
		undermatrise, subMat( ) .....	158
		utvid/sett sammen, augment( ) .....	14
		max( ), maksimum .....	95
		mean( ), gjennomsnitt .....	95
		med (gitt at),   .....	204
		median-median linjeregresjon, MedMed .....	97
		median( ), median .....	96
		median, median( ) .....	96
		MedMed, median-median linjeregresjon .....	97
		mid( ), midtstreng .....	98
		midtstreng, mid( ) .....	98
		min( ), minimum .....	98
		mindre enn eller lik, { .....	192
		minimum, min( ) .....	98
		minste felles multiplum, lcm .....	80
		mirr( ), endret internrente av retur .....	99
		mod( ), modul .....	100
		modul, mod( ) .....	100
		modus-innstillinger, lesModus( ) .....	68
		moduser lesing, lesModus( ) .....	144
		mRow( ), matrise radhandling .....	100
		mRowAdd( ), matrise radmultiplikasjon og addisjon .....	100
		multiplisere, * .....	185
<b>M</b>			
maksimum, max( ) .....	95		
mat $\blacktriangleright$ hist( ), matrise til liste .....	95		
matrices (matriser) undermatrise, subMat( ) .....	159		
matrise (1 $\times$ 2) sjablon for .....	4		
matrise (2 $\times$ 1) sjablon for .....	4		
matrise (2 $\times$ 2) sjablon for .....	4		
matrise (m $\times$ n) sjablon for .....	4		
matrise til liste, mat $\Delta$ liste( ) .....	95		
matriser determinant, det( ) .....	41		
diagonal, diag( ) .....	42		
dimensjon, dim( ) .....	42		
egenvektor, eigVc( ) .....	47		
egenverdi, eigVl( ) .....	47		
eliminasjonsform, ref( ) .....	129		
fill, Fylle .....	55		
identitet, identitet( ) .....	72		
kolonnedimensjoner, colDim( ) .....	24		
kolonnenorm, colNorm( ) .....	24		
kumulativ sum, cumulativeSum( ) .....	35		
liste til matrise, list $\blacktriangleright$ mat( ) .....	87		
maksimum, max( ) .....	95		
matrise til liste, mat $\Delta$ liste( ) ...	95		
minimum, min( ) .....	98		
nedre-øvre dekomposisjon, LU .....	94		







$\sinh()$ , hyperbolsk sinus .....	150	gjennomsnitt, mean( ) .....	95
SinReg, sinusregresjon .....	151	kombinasjoner, nCr( ) .....	104
sinus, sin( ) .....	149	median, median( ) .....	96
sinusregresjon, SinReg .....	151	permutasjoner, nPr( ) .....	110
sjabloner		standardavvik, stdDev( ) .....	156, 173
(T), produkt .....	5	tilfeldig startverdi, RandSeed ..	128
2-delers stykkevis funksjon .....	2	tilfeldig norm, tilfNorm( ) .....	127
Absoluttverdi .....	4	to-variable resultater, TwoVar ..	170
andrederivert .....	6	varians, variance( ) .....	173
bestemt integral .....	6	stdDevPop( ), populasjonens	
brøk .....	1	standardavvik .....	156
e eksponent .....	2	stdDevSamp( ), utvalgets	
eksponent .....	1	standardavvik .....	156
første deriverte .....	5	stigningstall, Loop .....	93
kvadratrot .....	1	Stoppkommando .....	157
ligningssystemer (2-ligning) ....	3	større enn eller lik med,   .....	193
ligningssystemer (N-ligning) ....	3	større enn, > .....	193
Logaritme .....	2	største felles divisor, gcd( ) .....	61
matrise (1 × 2) .....	4	streng	
matrise (2 × 1) .....	4	dimensjon, dim( ) .....	42
matrise (2 × 2) .....	4	lengde .....	42
matrise (m × n) .....	4	strenger	
n-te rot .....	1	brukes for å opprette	
stykkevis funksjon (N-delers) ...	3	variabelnavn .....	228
sum (G) .....	5	format, format( ) .....	57
skift, shift( ) .....	145	formatering .....	57
slett		høyre, høyre( ) .....	75, 134-135
åpne elementer fra liste .....	41	Indir.ref, # .....	200
feil, SlettFeil .....	23	innenfor, inString .....	74
Slett .....	211	legg til, & .....	195
slettAZ .....	23	midtstreng, mid( ) .....	98
slette		rottere, rotate( ) .....	136
variabel, DelVar .....	40	skift, shift( ) .....	145
SlettFeil, slett feil .....	23	streng til uttrykk, expr( ) .....	52
snarveier, tastatur .....	225	tegnkode, ord( ) .....	115
SortA, sorter stigende .....	152	tegnstreng, char( ) .....	21
SortD, sorter fallende .....	153	uttrykk til streng, string( ) .....	157
sorterer		venstre, left( ) .....	80
fallende, SortD .....	153	string( ), uttrykk til streng .....	157
stigende, SortA .....	152	strings	
språk		right, right( ) .....	49, 175
hente språkinformasjon .....	67	stykkevis funksjon (N-delers)	
sqrt( ), kvadratrot .....	154	sjablon for .....	3
standardavvik, stdDev( ) .....	156, 173	stykkevis( ) .....	116
stat.resultater .....	154	subMat( ), undermatrise .....	158-159
stat.verdier .....	155	subtrahere, - .....	184
statistikk		sum (G)	
en-variabel-statistikk, OneVar ..	112	sjablon for .....	5
fakultet, ! .....	195	sum av hovedbetalinger .....	199

sum av rentebetalinger .....	198	tilnærmet, approx( ) .....	12
sum( ), summering .....	158	tInterval, t konfidensintervall .....	164
sum, $\Sigma$ ( ) .....	197	tInterval_2Samp, toutvalg t konfidensintervall .....	164
sumlf( ) .....	159	to-variable resultater, TwoVar .....	170
summering, sum( ) .....	158	tomme (åpne) elementer .....	223
sumSeq( ) .....	159	tomrom, test for .....	79
svar (siste), Ans .....	12	trace( ) .....	166
<b>T</b>			
T, transponert .....	160	transponert, T .....	160
t test, tTest .....	167	Try, feil håndteringskommando .....	166
$\tan^{-1}$ ( ), invers tangens .....	161	try, Try .....	166
tan( ), tangens .....	160	tTest, t test .....	167
tangens, tan( ) .....	160	tTest_2Samp, to-utvalgs t-test .....	168
$\tanh^{-1}$ ( ), hyperbolsk invers tangens .....	162	TVM-argumenter .....	170
$\tanh$ ( ), hyperbolsk tangens .....	161	tvmFV( ) .....	169
tCdf(), sannsynlig student t-fordeling .....	163	tvml( ) .....	169
tegn .....		tvmN( ) .....	169
streng, char( ) .....	21	tvmPmt( ) .....	169
tegnkode, ord( ) .....	115	tvmPV( ) .....	170
tegne .....	212-214	TwoVar, to-variable resultater .....	170
tegnstreng, char( ) .....	21	<b>U</b>	
Tekstkommando .....	163	ulik, $\neq$ .....	191
teller dager mellom datoer, dbd( ) ..	37	undermatrise, subMat( ) .....	158-159
telllf( ), antall betingede elementer i en liste .....	32	unitV( ), enhetsvektor .....	172
test for tomrom, isVoid( ) .....	79	unLock, lås opp variabel eller variabelgruppe .....	173
Test_2S, 2_ utvalg F test .....	59	uttrykk .....	
tidsverdi for penger, antall betalinger .....	169	streng til uttrykk, expr( ) .....	52
tidsverdi for penger, betalingsbeløp .....	169	utvid/sett sammen, augment( ) .....	14
tidsverdi for penger, Fremtidig verdi .....	169	<b>V</b>	
tidsverdi for penger, nåverdi .....	170	variabel .....	
tidsverdi for penger, Rente .....	169	opprette navn fra en tegnstreng .....	228
tier-potens, $10^{\wedge}$ ( ) .....	203	variabler .....	
tilf( ), tilfeldig nummer .....	125	lokal, Local .....	89
tilfBin, tilfeldig tall .....	126	slette alle enkelttegn .....	23
tilfeldig .....		slette, DelVar .....	40
matrise, tilfMat( ) .....	127	variabler og funksjoner .....	
norm, tilfNorm( ) .....	127	kopiere .....	25
polynom, tilfPoly( ) .....	127	variabler, låse og låse opp .....	68, 90, 173
startverdi, RandSeed .....	128	varians, variance( ) .....	173
tilfeldig utvalg .....	127	varPop( ) .....	173
tilfInt( ), tilfeldig heltall .....	126	varSamp( ), utvalgets varians .....	173
tilfNorm(), tilfeldig norm .....	127	varselkoder og meldinger .....	241
tilfPoly(), tilfeldig polynom .....	127	vektorer .....	
tilfSamp( ) .....	127	enhet, unitV( ) .....	172
tilMat( ), tilfeldig matrise .....	127		

kryssprodukt, crossP( ) .....	33		
Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (For, While, eller Loop). .....	36		<b>Δ</b>
prikk produkt, dotP( ) .....	45	Δlist( ), differensliste .....	87
venstre, left( ) .....	80		<b>X</b>
verken ... eller, Boolsk operator .....	108	χ <sup>2</sup> Cdf( ) .....	22
vinkel, angle( ) .....	9	χ <sup>2</sup> GOF .....	22
vis data, Vis .....	43, 141		
Vis, vis data .....	43, 141		
vis som			
binær, 4Grunntall2 .....	17		
desimalt heltall, 4Grunntall10 ..	18		
desimalvinkel, ►DD .....	37		
grader/minutter/sekunder, ►DMS .....	45		
heksadesimal, 4Grunntall16 .....	19		
polar vektor, ►Polar .....	117		
rektangulær vektor, ►Rect .....	128		
sfærisk vektor, ►Sphere .....	153		
sylindrisk vektor, ►Cylind .....	36		
			<b>W</b>
warnCodes( ), Warning codes .....	175		
when( ), når .....	175		
while, While .....	176		
While, while .....	176		
			<b>X</b>
x <sup>-1</sup> , resiprok .....	204		
x <sup>2</sup> , kvadrat .....	188		
XNOR .....	194		
			<b>Z</b>
zInterval, z konfidensintervall .....	177		
zInterval_1Prop, en-proporsjons z konfidensintervall .....	178		
zInterval_2Prop, to-proporsjons z konfidensintervall .....	178		
zInterval_2Samp, to-utvalgs z konfidensintervall .....	179		
zTest .....	180		
zTest_1Prop, en-proporsjons z-test ..	181		
zTest_2Prop, to-proporsjons z-test ..	181		
zTest_2Samp, to-utvalgs z-test .....	182		