

# TI-Nspire™ CX CAS 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>8</b>
A .....	8
B .....	17
C .....	21
D .....	47
E .....	61
F .....	72
G .....	81
I .....	92
L .....	100
M .....	117
N .....	126
O .....	134
P .....	137
Q .....	146
R .....	150
S .....	165
T .....	192
U .....	207
V .....	208
W .....	209
X .....	211
Z .....	213
<b>Symboler</b> .....	<b>221</b>
<b>TI-Nspire™ CX II – Tegnekommandoer</b> .....	<b>248</b>
Grafikkprogrammering .....	248
Grafikkskjerm .....	248
Standardvisning og innstillinger .....	249
Feilmeldinger på grafikkskjerm .....	250
Ugyldige kommandoer i grafikkmodus .....	250
C .....	251
D .....	252
F .....	255
G .....	257
P .....	258
S .....	260
U .....	262

<b>Tomme (åpne) elementer</b> .....	<b>263</b>
<b>Snarveier/hurtigtaster for å legge inn matematiske uttrykk</b> .....	<b>265</b>
<b>EOS™ (Ligningsoperativsystem)-hierarkiet</b> .....	<b>267</b>
<b>TI-Nspire CX II – TI-Basic programmeringsfunksjoner</b> .....	<b>269</b>
Auto-innrykk i Programmeringseditor .....	269
Forbedrede feilmeldinger for TI-Basic .....	269
<b>Konstanter og verdier</b> .....	<b>272</b>
<b>Feilkoder og feilmeldinger</b> .....	<b>273</b>
<b>Advarselskoder og -meldinger</b> .....	<b>281</b>
<b>Generell informasjon</b> .....	<b>283</b>
<b>Stikkordregister</b> .....	<b>284</b>

# Uttrykksjabloner

Med uttrykksjablonene 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 223.

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 224.

Eksempel:

$$2^3 = 8$$

## Kvadratrot-sjablon

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



**Merk:** Se også  $\sqrt{}$  (kvadratrot), side 235.

Eksempel:

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

## N-te rot-sjablon

**ctrl** **^** taster



**Merk:** Se også rot(), side 162.

Eksempel:

## N-te rot-sjablon

ctrl ^ taster

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

## e eksponent-sjablon

e<sup>x</sup> tast

e<sup>□</sup>

Naturlig grunntall  $e$  opphøyd i en eksponent

**Merk:** Se også  $e^{\wedge}()$ , side 61.

$$e^1 \quad e$$
$$e^1. \quad 2.71828182846$$

## Logaritme-sjablon

ctrl 10<sup>x</sup> taster

log<sub>□</sub>(□)

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

**Merk:** Se også  $\log()$ , side 112.

Eksempel:

$$\log_{10}(2.) \quad 0.5$$

## Stykkevis sjablon (2-delers)

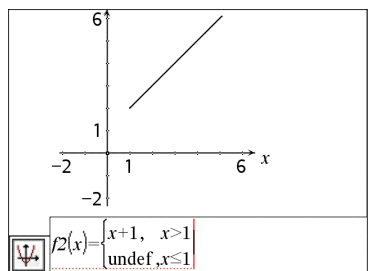
Katalog >  $\left\{ \begin{array}{l} \square \\ \square \end{array} \right.$

$\left\{ \begin{array}{l} \square \\ \square \end{array} \right.$

Lar deg opprette uttrykk og betingelser for en to-delers stykkevis definert funksjon. For å legge til en del, klikk på sjablonen og gjenta sjablonen.

**Merk:** Se også  $\text{stykkevis}()$ , side 139.

Eksempel:



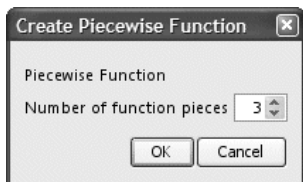
## Stykkevis sjablon (N-delers)

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 139.

## Sjablon for ligningssystemer med 2 ukjente

Katalog > 



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

**Merk:** Se også `system()`, side 192.

Eksempel:

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

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

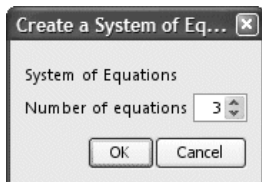
## Sjablon for ligningssystemer med N ukjente

Katalog > 

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

Eksempel:


Se eksemplet for Sjabloner for ligningssystemer (2 ligninger).



**Merk:** Se også `system()`, side 192.

## Sjablon for absoluttverdi

Katalog > 

 Merk: Se også **abs()**, side 8.

Eksempel:

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

## gg°mm'ss.ss'' sjablon

Katalog > 



Eksempel:

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.

$$30^{\circ}15'10'' \quad \frac{10891 \cdot \pi}{64800}$$

## Matrise-sjablon (2 x 2)

Katalog > 



Eksempel:

Oppretter en 2 x 2-matrise.

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

## 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)

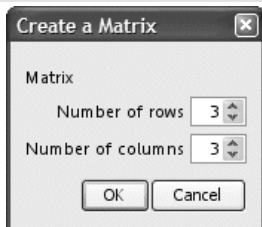
Katalog > 

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 236.

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 235.

## Første derivert-sjablon

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

Eksempel:

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

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

Den første deriverte sjablonen kan også brukes for å beregne førstederiverte i et punkt.

## Første derivert-sjablon

Katalog > 

Merk: Se også **d()** (derivert), side 232.

## Andre derivert-sjablon

Katalog > 

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

Den andre deriverte sjablonen kan også brukes for å beregne andrederiverte i et punkt.

Merk: Se også **d()** (derivert), side 232.

Eksempel:

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

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

## N-te derivert-sjablon

Katalog > 

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

Merk: Se også **d()** (derivert), side 232.

Eksempel:

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

## Bestemt integral-sjablon

Katalog > 

$$\int_a^b f(x) dx$$

Merk: Se også **f()** integral(), side 221.

Eksempel:

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

## ubestemt integral-sjablon

Katalog > 

$$\int f(x) dx$$

Merk: Se også **f()** integral(), side 221.

Eksempel:

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

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

Bruk - eller (-) for venstre grense. Bruk + for høyre grense.

**Merk:** Se også **grense()**, side 102.

Eksempel:

---

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

---

# Alfabetisk oversikt

Elementer med navn som ikke er alfabetiske (som f.eks. +, !, og >) er opplistet på slutten av dette avsnittet (side 221). 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(Uttr1)** ⇒ uttrykk

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

**abs(Liste1)** ⇒ liste

$$|2-3 \cdot i| \quad \sqrt{13}$$

**abs(Matrise1)** ⇒ matrise

$$|z| \quad |z|$$

Returnerer argumentets absoluttverdi.

$$|x+y \cdot i| \quad \sqrt{x^2+y^2}$$

**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 205.

- 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 236, og *bal()*, side 17.

## and

*BoolskUtt1* and *BoolskUtt2*  $\Rightarrow$  *Boolsk uttrykk*

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

*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(Uttr1) ⇒ uttrykk**

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

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

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:

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

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

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

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

**angle(Liste1) ⇒ liste**

**angle(Matrise1) ⇒ matrise**

## angle() vinkel

Katalog > 

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

## ANOVA

Katalog > 

**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 186).

*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

Katalog > 

**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 186).

*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



Utdata-variabel	Beskrivelse
stat.MSCol	Gjennomsnitt av kvadrater for kolonnefaktor

#### RADFAKTOR Utdata

Utdata-variabel	Beskrivelse
stat.Frow	F-statistikk over kolonnefaktoren
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.PValInteract	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)

**ctrl** **(-)** **taster**

Ans⇒verdi

56 56

Returnerer resultatet av det sist behandlede uttrykket.



56+4 60

60+4 64

## approx() (tilnærm)

Katalog > **approx(Uttr1)⇒uttrykk**

Returnerer behandlingen av argumentet som et uttrykk med desimalverdier, hvis mulig, uavhengig av om modus er **Auto** eller **Tilnærmet**.

Dette er det samme som å skrive inn argumentet og trykke på  .

**approx(Liste1)⇒liste****approx(Matrise1)⇒matrise**

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

$\text{approx}\left(\frac{1}{3}\right)$	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}
$\text{approx}(\sqrt{2}, \sqrt{3})$	[1.41421 1.73205]
$\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)$	[0.333333 0.111111]
$\text{approx}\{\sin(\pi), \cos(\pi)\}$	{0,-1}
$\text{approx}(\sqrt{2}, \sqrt{3})$	[1.41421 1.73205]

## approxFraction()

Katalog > **Uttr ▶approxFraction([Tol])⇒uttrykk****Liste ▶approxFraction([Tol])⇒liste****Matrise ▶approxFraction([Tol])⇒matrise**

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

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

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
0.833333333333333 ▶approxFraction(5.E-14)	$\frac{5}{6}$
{ $\pi, 1.5$ } ▶approxFraction(5.E-14)	$\left\{\frac{5419351}{1725033}, \frac{3}{2}\right\}$

## approxRational()

Katalog > **approxRational(Uttr[, Tol])⇒uttrykk****approxRational(Liste[, Tol])⇒liste****approxRational(Matrise[, Tol])⇒matrise**

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

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

**arccos()** Se  $\cos^{-1}()$ , side 33.

**arccosh()** Se  $\cosh^{-1}()$ , side 35.

**arccot()** Se  $\cot^{-1}()$ , side 36.

**arccoth()** Se  $\coth^{-1}()$ , side 37.

**arccsc()** Se  $\csc^{-1}()$ , side 39.

**arccsch()** Se  $\operatorname{csch}^{-1}()$ , side 40.

**arcLen() (bueLen)** Katalog > 

**arcLen**(*Uttr1*, *Var*, *Start*, *Slutt*)  $\Rightarrow$  uttrykk

Returnerer buelengden for *Uttr1* fra *Start* til *Slutt* med hensyn på variabel *Var*.

Buelengden beregnes som et integral av et uttrykk definert i funksjonsmodus.

**arcLen**(*Liste1*, *Var*, *Start*, *End*)  $\Rightarrow$  liste

Returnerer en liste over buelengdene til hvert element i *Liste1* fra *Start* til *Slutt* med hensyn til *Var*.

---

$$\operatorname{arcLen}(\cos(x), x, 0, \pi) \quad 3.8202$$

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

---

---

$$\operatorname{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi) \quad \{3.8202, 3.8202\}$$

---

**arcsec()** Se  $\sec^{-1}()$ , side 166.

**arcsech()**

Se  $\text{sech}^{-1}()$ , side 166.

**arcsin()**

Se  $\text{sin}^{-1}()$ , side 177.

**arcsinh()**

Se  $\text{sinh}^{-1}()$ , side 179.

**arctan()**

Se  $\text{tan}^{-1}()$ , side 193.

**arctanh()**

Se  $\text{tanh}^{-1}()$ , side 195.

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

Katalog > 

**augment(Liste1, Liste2)⇒liste**

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

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

**augment(Matrise1, Matrise2)⇒matrise**

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

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*.

**avgRC() (gjsnEH)**

Katalog &gt;

**avgRC**(*Uttr1*, *Var* [=Verdi] [, *Trinn*])⇒uttrykk

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

**avgRC**(*Uttr1*, *Var* [=Verdi] [, *Liste1*])⇒liste

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

**avgRC**(*Liste1*, *Var* [=Verdi] [, *Trinn*])⇒liste

$$\text{avgRC}(x^2 - x + 2, x) \quad 2 \cdot (x - 0.4995)$$

$$\text{avgRC}(x^2 - x + 2, x, 0.1) \quad 2 \cdot (x - 0.45)$$

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

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

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.

**B****bal()**

Katalog &gt;

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

$$\text{bal}(5, 6, 5.75, 5000, , 12, 12) \quad 833.11$$

**bal**(*NPmt*, *amortTabell*)⇒verdi

$$tbl := \text{amortTbl}(6, 6, 5.75, 5000, , 12, 12)$$

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 205.

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

$$\text{bal}(tbl) \quad 1674.27$$

*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 205.

- Hvis du utelater  $Pmt$ , grunninnstilles den til  $Pmt=tvmpmt(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- 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 8.

**Merk:** Se også  $\Sigma Int()$  og  $\Sigma Prn()$ , side 236.

## ►Base2 (Grunntall2)

*Heltall1* ►Base2⇒*heltall*

256►Base2

0b10000000

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

0h1F►Base2

0b11111

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

0hFFFFFFFFFFFFFFF i heksadesimal  
modus

0b111...111 (64 1-ere) i binær modus

$-2^{63}$  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)

Katalog > 

*Heltall1* ►Base10⇒*heltall*

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

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

0h1F►Base10	31
-------------	----

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.

## ►Base16 (Grunntall16)

Katalog > 

*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 18.

**binomCdf()**

**binomCdf**( $n,p$ )⇒*liste*

**binomCdf**

**binomCdf**( $n,p, \text{nedreGrense}, \text{øvreGrense}$ )⇒*tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

**binomCdf**( $n,p, \text{øvreGrense}$ ) for  $P(0 \leq X \leq \text{øvreGrense})$ ⇒*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.

For  $P(X \leq \text{øvreGrense})$ , sett *nedreGrense*=0

**binomPdf()**

**binomPdf**( $n,p$ )⇒*liste*

**binomPdf**( $n,p, XVerd$ )⇒*tall* hvis *XVerd* er et tall, *liste* hvis *XVerd* er en liste

Beregner 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)**

**ceiling**(*Uttr1*)⇒*heltall*

`ceiling(.456)`

1.

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

**ceiling() (øvre)**Katalog > 

Argumentet kan være et reelt eller et komplekst tall.

**Merk:** Se også **floor()** (nedre).

**ceiling(Liste1)** ⇒ *liste*

**ceiling(Matrise1)** ⇒ *matrise*

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

$\text{ceiling}\{-3.1, 1.2, 5\}$	$\{-3., 1.3\}$
$\text{ceiling}\begin{pmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{pmatrix}$	$\begin{pmatrix} 0 & -3. \cdot i \\ 2. & 4 \end{pmatrix}$

**centralDiff()**Katalog > 

**centralDiff(Uttr1, Var [=Verdi], Trinn)** ⇒ *uttrykk*

**centralDiff(Uttr1, Var, Trinn)** | *Var* = *Verdi* ⇒ *uttrykk*

**centralDiff(Uttr1, Var [=Verdi], Liste)** ⇒ *liste*

**centralDiff(Liste1, Var [=Verdi], Trinn)** ⇒ *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 “|” 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 **d()**.

$\text{centralDiff}(\cos(x), x, h)$	$\frac{-\cos(x-h) - \cos(x+h)}{2 \cdot h}$
$\lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h))$	$-\sin(x)$
$\text{centralDiff}(x^3, x, 0.01)$	$3 \cdot (x^2 + 0.000033)$
$\text{centralDiff}(\cos(x), x)   x = \frac{\pi}{2}$	$-1.$
$\text{centralDiff}(x^2, x, \{0.01, 0.1\})$	$\{2 \cdot x, 2 \cdot x\}$

**cFactor() (kFaktor)**Katalog > 

**cFactor(Uttr1[, Var])** ⇒ *uttrykk*

**cFactor(Liste1[, Var])** ⇒ *liste*

cFactor(MatriseI[,Var])⇒matrise

cFactor(Uttr1) returnerer Uttr1, faktorisert med hensyn på alle dets variabler over en felles nevner.

Uttr1 er faktorisert så mye som mulig i lineære rasjonale faktorer, selv om det innfører nye, ikke-reelle tall. Med dette alternativet kan du faktorisere med hensyn på mer enn en variabel.

cFactor(Uttr1,Var) returnerer Uttr1, faktorisert med hensyn på variabel Var.

Uttr1 er faktorisert så mye som mulig i faktorer som er lineære i Var, muligens med ikke-reelle konstanter, selv om det innfører irrasjonale konstanter eller deluttrykk som er irrasjonale i andre variabler.

Faktorene og leddene deres er sortert med Var som hovedvariabel. Liknende potenser av Var er samlet sammen i hver faktor. Inkluder Var hvis du må faktorisere med hensyn på bare den ene variabelen og du er villig til å akseptere irrasjonale uttrykk i en annen tilfeldig variabel for å øke faktoriseringen med hensyn på Var. Det kan hende at faktor bestemmes tilfeldig med hensyn på andre variabler.




Ved Auto-innstilling av modusen **Auto eller Tilnærmet** vil en inkludering av Var også gjøre det mulig å tilnærme med flytende desimalpunkt-koeffisienter der hvor irrasjonale koeffisienter ikke kan uttrykkes eksplisitt utfra innebygde funksjoner. Selv dersom det bare er én variabel, vil man kunne oppnå en mer komplett faktorisering ved å inkludere Var.

**Merk:** Se også faktor().

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

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

cFactor( $x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$ )	
	$x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$
cFactor( $x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$ )	
	$(x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x + 0.964673) \cdot (x + 0.611649)$

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

**char()**Katalog > **char(Heltall)**⇒tegn

char(38) " &amp; "

char(65) " A "

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

**charPoly()**katalog > **charPoly****(kvadratMatrise, Var)**⇒polynomuttrykk
$$m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix} \quad \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$$
**charPoly****(kvadratMatrise, Uttr)**⇒polynomuttrykkcharPoly(*m*, *x*)  $-x^3 + 5 \cdot x^2 + 7 \cdot x - 35$ **charPoly**
**(**  
*kvadratMatrise1, Matrise2*  
**)**⇒polynomuttrykk
charPoly(*m*, *x*<sup>2</sup>+1)  $-x^6 + 2 \cdot x^4 + 14 \cdot x^2 - 24$ charPoly(*m*, *m*) 0

Returnerer det karakteristiske polynomet til *kvadratMatrise*. Det karakteristiske polynomet til  $n \times n$ -matrisen *A*, angitt som  $p_A(\lambda)$ , er polynomet som er definert ved

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

der *I* er  $n \times n$  identitetsmatrisen.

*kvadratMatrise1* og *kvadratMatrise2* må ha samme dimensjon.

**χ<sup>2</sup>way**Katalog > **χ<sup>2</sup>way** *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 186).

For informasjon om effekten av tomme elementer i en matrise, se "Tomme (åpne) elementer" (side 263).

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.UttrMat	Matrise av forventet element-talletabell ved antatt nullhypotese
stat.KompMat	Matrise av elementbidrag til chi kvadratstatistikk

## $\chi^2\text{Cdf}()$

Katalog > 

$\chi^2\text{Cdf}(\text{nedreGrense}, \text{\o vreGrense}, \text{df}) \Rightarrow \text{tall}$   
hvis *nedreGrense* og *\o vreGrense* er tall,  
*liste* hvis *nedreGrense* og *\o vreGrense* er  
lister

$\text{chi2Cdf}(\text{nedreGrense}, \text{\o vreGrense}, \text{df}) \Rightarrow \text{tall}$   
hvis *nedreGrense* og *\o vreGrense* er tall,  
*liste* hvis *nedreGrense* og *\o vreGrense* er  
lister

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

For  $P\{X \leq \text{\o vreGrense}\}$ , sett *nedreGrense* = 0.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

## $\chi^2\text{GOF}$

Katalog > 

$\chi^2\text{GOF} \text{ obsListe}, \text{uttrListe}, \text{df}$

$\text{chi2GOF} \text{ obsListe}, \text{uttrListe}, \text{df}$

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 186).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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()

$\chi^2$ Pdf(*XVerd*,*df*) $\Rightarrow$ tall hvis *XVerd* er et tall,  
liste hvis *XVerd* er en liste

chi2Pdf(*XVerd*,*df*) $\Rightarrow$ tall hvis *XVerd* er et  
tall, liste hvis *XVerd* er en liste

Beregner sannsynlighetstettheten (pdf) for  $c^2$ -fordelingen ved en bestemt *XVerd*-verdi for det angitte antallet frihetsgrader *df*.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

## ClearAZ (slettAZ)

## ClearAZ

$5 \rightarrow b$	5
<i>b</i>	5
ClearAZ	Done
<i>b</i>	<i>b</i>

Sletter alle enkelttegn-variabler i det aktuelle oppgaveområdet.

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 208.

## ClrErr (SlettFeil)

## ClrErr

Tømmer feilstatus og stiller systemvariabelen *feilKode* til null.

For et eksempel på **ClrErr**, se eksempel 2 under **Try**-kommandoen, side 201.

Else -leddet i Try...Else...EndTry-blokken bør bruke ClrErr eller PassErr. Hvis feilen skal bearbejdes 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 PassErr, side 138, og Try, side 201.

**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) ⇒ 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.

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

## colDim()

colDim(Matrise) ⇒ uttrykk

Returnerer antallet kolonner som ligger i Matrise.

**Merk:** Se også radDim().

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

## colNorm()

colNorm(Matrise) ⇒ uttrykk

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

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

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

## comDenom()

**comDenom**(*Uttr1*[,*Var*]) $\Rightarrow$ uttrykk

**comDenomListe1**[,*Var*]) $\Rightarrow$ liste

**comDenom**(*Matrise1*[,*Var*]) $\Rightarrow$ matrise

**comDenom**(*Uttr1*) returnerer en redusert brøk av en fullt utvidet teller over en fullt utvidet nevner.

**comDenom**(*Uttr1*,*Var*) returnerer en redusert brøk av teller og nevner som er utvidet med hensyn på *Var*. Leddene og faktorene deres er sortert med *Var* som hovedvariabel. Liknende potenser av *Var* er samlet sammen. Det kan hende at faktor bestemmes tilfeldig av de innsamlede koeffisienter. Sammenliknet med å utelate *Var* sparer dette ofte tid samt plass både i minnet og på skjermen, samtidig som uttrykket blir mer forståelig. Det gjør også at etterfølgende handlinger på resultatet går raskere og at minnet ikke belastes så mye.

Hvis *Var* ikke opptrer i *Uttrykk1*, vil **comDenom**(*Uttr1*,*Var*) returnere en redusert brøk av en ikke utvidet teller over en ikke utvidet nevner. Slike resultater sparer vanligvis både tid og plass i både minnet og på skjermen. Slike delvis faktoriserte resultater gjør også at etterfølgende handlinger på resultatet går mye raskere og at minnet ikke belastes så mye.

Selv om det ikke foreligger noen nevner, er **comDen**-funksjonen ofte en rask måte å oppnå delvis faktorisering på, hvis **factor()** er for langsam eller hvis den tar for stor plass i minnet.

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

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

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

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

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

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

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

*Done*

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

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

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



**Tips:** Legg inn denne **comden()**-funksjonsdefinisjonen og prøv den rutinemessig som et alternativ til **comDenom()** og **factor()**.

**completeSquare ()**

**completeSquare**(UtrEllerLign, Var) $\Rightarrow$ uttrykk eller ligning

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

**completeSquare**(UtrEllerLign, Var^Potens) $\Rightarrow$ uttrykk eller ligning

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

**completeSquare**(UtrEllerLign, Var1, Var2 [...]) $\Rightarrow$ uttrykk eller ligning

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

**completeSquare**(UtrEllerLign, {Var1, Var2 [...]} $\Rightarrow$ uttrykk eller ligning

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

Omregner et kvadratisk polynomuttrykk av formen  $a\cdot x^2+b\cdot x+c$  til formen  $a\cdot (x-h)^2+k$

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

- eller -

Omregner en kvadratisk ligning av formen  $a\cdot x^2+b\cdot x+c=d$  til formen  $a\cdot (x-h)^2=k$

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

Det første argumentet må være et kvadratisk uttrykk eller en kvadratisk ligning i standard form med hensyn på det andre argumentet.

Det andre argumentet må være et enkelt ledd i én variabel eller et enkelt ledd i én variabel opphøyd i en rasjonal eksponent, for eksempel  $x$ ,  $y^2$  eller  $z^{1/3}$ .

Den tredje eller fjerde syntaksen forsøker å fullføre kvadratet med hensyn på variabler *Var1*, *Var2* [... ]).

**conj()**

Katalog &gt;

**conj**(*Uttr1*) $\Rightarrow$ uttrykk

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

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

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

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

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

Returnerer den komplekse konjugerte av argumentet.

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

**Merk:** Alle ubestemte variabler behandles som reelle variabler.**constructMat()**

katalog &gt;

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

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

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*

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

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

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

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

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

CopyVar <i>b,c: c(4)</i>	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</i> :=45	45																
<i>aa.b</i> :=6.78	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* ].

## bcos

Uttr **bcos**

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

$$\frac{(\sin(x))^2}{\cos}$$

$$\frac{1 - (\cos(x))^2}{}$$

Representerer *Uttr* med cosinus. Dette er en konverteringsoperator. Den kan bare brukes på slutten av kommandolinjen.

►cos reduserer alle potenser av sin(...) modulus  $1 - \cos(\dots)^2$  slik at alle gjenværende potenser av cos(...) har eksponenter i området (0, 2). Dermed vil resultatet være uten sin(...) hvis og bare hvis sin(...) inntreffer i det gitte uttrykket bare med partallsekspionter.

**Merk:** Denne konverteringsoperatoren støttes ikke i vinkelmodusen grader eller gradianer. Før du bruker den, må du kontrollere at vinkelmodus er satt til radianer, og at *Uttr* ikke inneholder eksplisitte referanser til vinkler i grader eller gradianer.

cos()

 **tast**

cos(*Uttr1*) ⇒ *uttrykk*

I Grader-vinkelmodus:

cos(*Liste1*) ⇒ *liste*

$$\frac{\cos\left(\frac{\pi_r}{4}\right)}{\frac{\sqrt{2}}{2}}$$

cos(*Uttr1*) returnerer cosinus til argumentet som et uttrykk.

$$\frac{\cos(45)}{\frac{\sqrt{2}}{2}}$$

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

$$\frac{\cos(\{0,60,90\})}{\left\{1, \frac{1}{2}\right\}}$$

**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.

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

**cos(kvadratMatriseI) ⇒ kvadratMatrise**

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

Når en skalarfunksjon f(A) virker på kvadratMatriseI (A), beregnes resultatet av algoritmen:

Beregner egenverdiene (li) og egenvektorene (V i) av A.

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

Utform matrisene:

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

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.

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<sup>-1</sup>(Utt) ⇒ uttrykk**

I Grader-vinkelmodus:

**cos<sup>-1</sup>()** **tast****cos<sup>-1</sup>(Liste1)**⇒*liste* $\cos^{-1}(1)$  0**cos<sup>-1</sup>(Utr1)** returnerer vinkelen som har cosinus lik *Utr1* som et uttrykk.

I Gradian-vinkelmodus:

**cos<sup>-1</sup>(Liste1)** returnerer en liste over invers cosinus for hvert element i *Liste1*. $\cos^{-1}(0)$  100**Merk:** Resultatet returneres som en vinkel i enten grader, gradian eller radian, avhengig av aktuell vinkelmodus-innstilling.

I Radian-vinkelmodus:

**Merk:** Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccos (...)**. $\cos^{-1}(\{0,0,2,0,5\})$   $\left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$ **cos<sup>-1</sup>**  
**(kvadratMatrise1)**⇒*kvadratMatrise*

I radian-vinkelmodus og rektangulært, kompleks format:

Returnerer matrisens inverse cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers cosinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**. $\cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$   
 $\begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.77836 \cdot i \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \cdot i \end{bmatrix}$ *kvadratMatrise1* må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

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

**cosh()****Katalog** > **cosh(Utr1)**⇒*uttrykk*

I Grader-vinkelmodus:

**cosh(Liste1)**⇒*liste* $\cosh\left(\left(\frac{\pi}{4}\right)^r\right)$   $\cosh(45)$ **cosh(Utr1)** returnerer hyperbolsk cosinus til argumentet som et uttrykk.**cosh(Liste1)** returnerer en liste over hyperbolsk cosinus til hvert element i *Liste1*.

I Radian-vinkelmodus:

**cosh**  
**(kvadratMatrise1)**⇒*kvadratMatrise*

## cosh()

Katalog > 

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

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

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

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

Katalog > 

**cosh<sup>-1</sup>(Utr1)** ⇒ *uttrykk*

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

**cosh<sup>-1</sup>(Liste1)** ⇒ *liste*

$$\cosh^{-1}(\{1, 2, 1, 3\}) = \{0, 1, 37286, \cosh^{-1}(3)\}$$

**cosh<sup>-1</sup>(Utr1)** returnerer invers hyperbolsk cosinus for argumentet som et uttrykk.

**cosh<sup>-1</sup>(Liste1)** returnerer en liste over invers hyperbolsk cosinus til hvert element i *Liste1*.

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

**cosh<sup>-1</sup>**  
(*kvadratMatrise1*) ⇒ *kvadratMatrise*

Returnerer matrisens inverse hyperbolske 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()**.

I radian-vinkelmodus og rektangulært, kompleks format:

$$\cosh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.4908i \\ 0.486969-0.725533 \cdot i & 1.66262+0.623491i \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018i \end{bmatrix}$$

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

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

## cot()

 **tast**

**cot(Utr1)** ⇒ *uttrykk*

I Grader-vinkelmodus:

## cot()



**cot**(*Liste1*) ⇒ *liste*

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

**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.

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

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



**cot<sup>-1</sup>**(*Utr1*) ⇒ *uttrykk*

**cot<sup>-1</sup>**(*Liste1*) ⇒ *liste*

Returnerer vinkelen som har cotangens lik *Utr1* eller returnerer en liste som inneholder invers cotangens til hvert element i *Liste1*.

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

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

$$\text{cot}^{-1}(1) \quad \frac{\pi}{4}$$

## coth()



**coth**(*Utr1*) ⇒ *uttrykk*

**coth**(*Liste1*) ⇒ *liste*

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

$$\text{coth}(1.2) \quad 1.19954$$

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



**coth<sup>-1</sup>()**

Katalog &gt;

**coth<sup>-1</sup>(Uttr1)**⇒uttrykkcoth<sup>-1</sup>(3,5) 0.293893**coth<sup>-1</sup>(Liste1)**⇒liste
$$\text{coth}^{-1}\left(\left\{-2, 2, 1, 6\right\}\right)$$

$$\left\{\frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2}\right\}$$

Returnerer invers hyperbolsk cotangens til *Uttr1*, eller returnerer en liste med invers hyperbolsk cotangens til hvert element i *Liste1*.

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

**count() (antall)**

Katalog &gt;

**count(Verdi1ellerListe1 [,Verdi2ellerListe2 [...]])**⇒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}, $\begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}$ ) 7

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

$$\text{count}\left(\frac{1}{2}, 3+4\cdot i, \text{undef}, \text{"hello"}, x+5, \text{sign}(0)\right)$$
 2

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

I det siste eksemplet ble bare 1/2 og 3+4\*i talt. De resterende argumentene, dersom x er udefinert, behandler ikke til numeriske verdier.

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 263.

**countIf() (tellIf)**

Katalog &gt;

**countIf(Liste,Kriterium)**⇒verdi

countIf({1,3,"abc",undef,3,1},3) 2

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

Teller alle elementer som er lik 3.

**countIf() (tellIf)**

*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 263.

**Merk:** Se også **sumIf()**, side 191, og **frequency()**, side 79.

$$\text{countIf}\{\{"abc", "def", "abc", 3\}, "def"\} \quad 1$$

Teller alle elementer som er lik "def."

$$\text{countIf}\{\{x^{-2}, x^{-1}, 1, x, x^2\}, x\} \quad 1$$

Teller alle elementer som er lik  $x$ ; dette eksemplet antar at variabelen  $x$  er udefinert.

$$\text{countIf}\{\{1, 3, 5, 7, 9\}, ? < 5\} \quad 2$$

Teller 1 og 3.

$$\text{countIf}\{\{1, 3, 5, 7, 9\}, 2 < ? < 8\} \quad 3$$

Teller 3, 5 og 7.

$$\text{countIf}\{\{1, 3, 5, 7, 9\}, ? < 4 \text{ or } ? > 6\} \quad 4$$

Teller 1, 3, 7 og 9.

**cPolyRoots()**

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

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

**cPolyRoots**(*KoeffListe*) ⇒ *liste*

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

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

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

*Poly* må være et polynom i én variabel.

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

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

**Merk:** Se også **polyRoots()**, side 143.

**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.

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

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

$$\text{crossP}(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \begin{bmatrix} -3 & 6 & -3 \\ 0 & 0 & -2 \end{bmatrix}) = \begin{bmatrix} -3 & 6 & -3 \\ 0 & 0 & -2 \end{bmatrix}$$

**csc()** **tast****csc(Uttr1) ⇒ uttrykk**

I Grader-vinkelmodus:

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

**csc(Liste1) ⇒ liste**Returnerer cosekans til *Uttr1*, eller returnerer en liste med cosekans til hvert element i *Liste1*.

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

**csc<sup>-1</sup>()** **tast****csc<sup>-1</sup>(Uttr1) ⇒ uttrykk**

I Grader-vinkelmodus:

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

**csc<sup>-1</sup>(Liste1) ⇒ liste**Returnerer vinkelen som har cosekans lik *Uttr1*, eller returnerer en liste med invers cosekans til hvert element i *Liste1*.

I Gradian-vinkelmodus:

**csc<sup>-1</sup>()**trig  **tast**

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

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

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

100.

I Radian-vinkelmodus:

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

**csch()**Katalog > **csch**(*Uttr1*) ⇒ *uttrykk*

$$\text{csch}(3)$$

$$\frac{1}{\sinh(3)}$$

**csch**(*Liste1*) ⇒ *liste*

Returnerer hyperbolsk cosekans til *Uttr1* eller returnerer en liste med hyperbolsk cosekans til alle elementene i *Liste1*.

$$\text{csch}\left(\left\{1, 2, 1, 4\right\}\right)$$

$$\left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

**csch<sup>-1</sup>()**Katalog > **csch<sup>-1</sup>**(*Uttr1*) ⇒ *uttrykk*

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

$$\sinh^{-1}(1)$$

**csch<sup>-1</sup>**(*Liste1*) ⇒ *liste*

Returnerer invers hyperbolsk cosekans til *Uttr1*, eller returnerer en liste med invers hyperbolsk cosekans til hvert element i *Liste1*.

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

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

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

**cSolve() (kløp)**Katalog > **cSolve**(*Ligning*, *Var*) ⇒ *Boolsk uttrykk*

$$\text{cSolve}(x^3 = -1, x)$$

**cSolve**(*Ligning*, *Var*=*Forslag*) ⇒ *Boolsk uttrykk*

$$x = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } x = -1$$

**cSolve**(*Ulikhet*, *Var*) ⇒ *Boolsk uttrykk*

$$\text{solve}(x^3 = -1, x)$$

x = -1

Returnerer komplekse løsningsalternativer av en ligning eller ulikhet i *Var*. Målet er å produsere alternativer for alle reelle og ikke-reelle løsninger. Selv om *Ligning* er reell, kan **cSolve()** returnere ikke-reelle resultater i Reelt resultat Komplekst format.

**cSolve()** setter midlertidig grunnmengde til kompleks i løpet av løsningsprosessen selv om den aktuelle grunnmengden er reell. I kompleks grunnmengde bruker brøkpotens med oddetall i nevneren hovedforgreining heller enn reell forgreining. Følgelig er løsninger fra **solve()** på ligninger som omfatter slike brøkpotenser ikke nødvendigvis en delmengde av løsningene fra **cSolve()**.

**cSolve()** starter med eksakte, symbolske metoder. Unntatt i **Eksakt** modus bruke **cSolve()** også iterativ, tilnærmet kompleks polynomisk faktorisering, om nødvendig.

**Merk:** Se også **cZeros()**, **solve()**, og **zeros()**.

**cSolve**[*Lign1* and *Lign2* [**and**...],  
*VarElForslag1*, *VarElForslag2* [, ... ]]  
⇒*Boolsk uttrykk*

**cSolve**[*LignSystem*, *VarElForslag1*,  
*VarElForslag2* [, ...]] ⇒*Boolsk uttrykk*

Returnerer komplekse løsningsalternativer til simultane, algebraiske ligninger, der hvert *varElForslag* spesifiserer en variabel som du vil finne løsningen til.

Alternativt kan du spesifisere et startforslag for en variabel. Hvert *varElForslag* må ha formen:

*variabel*

$\text{cSolve}\left(x^{\frac{1}{3}}=-1,x\right)$	false
$\text{solve}\left(x^{\frac{1}{3}}=-1,x\right)$	$x=-1$

I Vis siffer-modus av Fast 2:

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

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

– eller –

variabel = reelt eller ikke-reelt tall

For eksempel er  $x$  gyldig, og det er også  $x=3+i$ .

Hvis alle ligningene er polynomiske og hvis du IKKE spesifiserer noe startforslag, bruker **cSolve()** Gröbner/Buchbergers leksikale eliminasjonsmetode for å prøve å bestemme alle komplekse løsninger.

Komplekse løsninger kan inkludere både reelle og ikke-reelle løsninger, som i eksemplet til høyre.

$$\text{cSolve}(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v\})$$

$$u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

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

Simultane, polynomiske ligninger kan ha ekstra variabler uten verdiparametre, men som representerer gitte tallverdier som kan settes inn senere.

$$\text{cSolve}(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\})$$

$$u = \frac{-(\sqrt{4 \cdot c - 1} \cdot i + 1)^2}{4} \text{ and } v = \frac{\sqrt{4 \cdot c - 1} \cdot i + 1}{2} \circ$$

Du kan også inkludere løsningsvariabler som ikke forekommer i ligningene. Disse løsningene viser hvordan løsningsfamilier kan inneholde vilkårlige konstanter i form av  $ck$ , hvor  $k$  er et heltall mellom 1 og 255.

$$\text{cSolve}(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\})$$

$$u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c \mathbf{d3} \text{ or}$$

For polynomiske systemer kan beregningstiden eller plassen i minnet sterkt avhenge av hvilken rekkefølge du setter løsningsvariabler i. Hvis startforslaget bruker opp minneplassen eller tålmodigheten din, kan du prøve å flytte om på variablene i ligningene og/eller *varEIForslag* -listen.

Hvis du ikke inkluderer noen forslag og hvis en ligning er ikke-polynomisk i en vilkårlig variabel men alle ligningene er lineære i alle løsningsvariabler, bruker **cSolve()** gaussisk eliminasjon for å prøve å bestemme alle løsninger.

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

$$u = \frac{e^w + i}{2} \text{ and } v = \frac{e^w - i}{2}$$

Hvis et system er verken polynomisk i alle variablene eller lineært i løsningsvariablene, bestemmer **cSolve()** som regel en løsning med en tilnærmet iterativ metode. I så fall må antallet løsningsvariabler være lik antallet ligninger, og alle andre variabler (parametre) i ligningene må forenkles til tall.

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

$w=0.494866 \text{ and } z=0.703467$

Et ikke-reelt forslag er ofte nødvendig for å bestemme en ikke-reell løsning. For konvergens kan det hende at et forslag må være ganske nært en løsning.

$$\text{cSolve}(e^z=w \text{ and } w=z^2, \{w,z=1+i\})$$

$w=0.149606+4.8919 \cdot i \text{ and } z=1.58805+1.5402i$

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

## CubicReg

**CubicReg**  $X, Y, [Frequ]$  [, *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 *Frequ*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 186).

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

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

*Frequ* er en valgfri liste med frekvensverdier. Hvert element i *Frequ* 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 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 263).

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 <i>X-listen</i> 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 <i>Y-listen</i> 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()

Katalog > 

**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(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.

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

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 263.

## Cycle (Løkke)

Katalog > 

**Cycle (Løkke)**

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

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.

Define $g()$ =Func	Done
Local $temp,i$	
$0 \rightarrow temp$	
For $i,1,100,1$	
If $i=50$	
Cycle	
$temp+i \rightarrow temp$	
EndFor	
Return $temp$	
EndFunc	
$g()$	5000

## ►Cylind

Vektor ►Cylind

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

Viser rad- eller kolonnevektor i sylindrisk form  $[r, \angle\theta, z]$ .

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

$[2 \ 2 \ 3]$ ►Cylind	$\left[ 2\sqrt{2} \ \angle \frac{\pi}{4} \ 3 \right]$
-----------------------	---

## cZeros() (kNullp)

**cZeros** (*Utt*, *Var*)  $\Rightarrow$ liste

Returnerer en liste over alternative reelle eller ikke-reelle verdier av *Var* som gir  $Utt=0$ . **cZeros()** gjør dette ved å beregne **uttr**►liste(**cSolve** ( $Utt=0,Var,Var$ )). Ellers er **cZeros()** lik **zeros()**.

**Merk:** Se også **cSolve()**, **solve()** og **zeros()**.

**cZeros**({*Utt1,Utt2* [, ... ] },  
{*VarElForslag1,VarElForslag2* [, ... ]  
}) $\Rightarrow$ matrise

Returnerer alternative posisjoner der alle uttrykkene er null samtidig. Hvert *VarElForslag* spesifiserer en ukjent som du vil finne verdien til.

I Vis siffer-modus av Fast 3:

$cZeros(x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3,x)$ $\{-1.114+1.073 \cdot i,-1.114-1.073 \cdot i,-2.125,-0.612,0\}$
--

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

Alternativt kan du spesifisere et startforslag for en variabel. Hvert *VarEIForslag* må ha formen:

*variabel*

– eller –

*variabel* = reelt eller ikke-reelt tall

For eksempel er  $x$  gyldig, og det er også  $x=3+i$ .

Hvis alle uttrykkene er polynomiske og hvis du IKKE spesifiserer noe startforslag, bruker **cZeros()** Gröbner/Buchbergers leksikale eliminasjonsmetode for å prøve å bestemme **alle** komplekse nullpunkter.

Komplekse nullpunkter kan inkludere både reelle og ikke-reelle nullpunkter, som i eksemplet til høyre.

Hver rad i resultatmatrisen presenterer et alternativt nullpunkt, med komponentene plassert som i *VarEIForslag* -listen. For å trekke ut en rad, pek på matrisen med *[rad]*.

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

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

Trekk ut rad 2:

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

Simultane polynomer kan ha ekstra variabler uten verdi (parametre), men som representerer gitte tallverdier som kan settes inn senere.

Du kan også inkludere ukjente variabler som ikke forekommer i uttrykkene. Disse nullpunktene viser hvordan nullpunkt-familier kan inneholde vilkårlige konstanter i form av  $ck$ , hvor  $k$  er et heltall mellom 1 og 255.

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

$$\begin{bmatrix} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{bmatrix}$$

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

$$\text{cZero}\left(\left\{u \cdot (v-1) - v, u + v^2\right\}, \left\{u, v, w\right\}\right)$$

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

For polynomiske systemer kan beregningstiden eller plassen i minnet sterkt avhenge av hvilken rekkefølge du setter de ukjente i. Hvis startforslaget bruker opp minneplassen eller tålmodigheten din, kan du prøve å flytte om på variablene i uttrykkene og/eller *VarElForslag*-listen.

Hvis du ikke inkluderer noen forslag og hvis et uttrykk er ikke-polynomisk i en vilkårlig variabel men alle uttrykkene er lineære i alle ukjente, bruker **cZeros()** gaussisk eliminasjon for å prøve å bestemme alle nullpunktene.

$$\left| \begin{array}{c} \text{cZeros}(\{u+v-e^w, u-v-i\}, \{u, v\}) \\ \left[ \frac{e^w+i}{2} \quad \frac{e^w-i}{2} \right] \end{array} \right.$$

Hvis et system er verken polynomisk i alle variablene eller lineært i de ukjente, bestemmer **cZeros()** som regel ett nullpunkt med en tilnærmet iterativ metode. I så fall må antallet ukjente være lik antallet uttrykk, og alle andre variabler i uttrykkene må forenkles til tall.

$$\left| \begin{array}{c} \text{cZeros}(\{e^z-w, w-z^2\}, \{w, z\}) \\ [0.494866 \quad -0.703467] \end{array} \right.$$

Et ikke-reelt forslag er ofte nødvendig for å bestemme et ikke-reelt nullpunkt. For konvergens kan det hende at et forslag må være ganske nært et nullpunkt.

$$\left| \begin{array}{c} \text{cZeros}(\{e^{-z}-w, w-z^2\}, \{w, z=1+i\}) \\ [0.149606+4.8919 \cdot i \quad 1.58805+1.54022 \cdot i] \end{array} \right.$$

## D

### dbd()

**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.

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

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)

## ►DD

Verdi ►DD⇒verdi

I Grader-vinkelmodus:

Liste1 ►DD⇒liste

$(1.5^\circ)$ ►DD	1.5°
-------------------	------

Matrise1 ►DD⇒matrise

$(45^\circ 22' 14.3'')$ ►DD	45.3706°
-----------------------------	----------

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

$(\{45^\circ 22' 14.3'', 60^\circ 0' 0''\})$ ►DD	{45.3706°, 60°}
--	-----------------

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 Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

## ►Decimal

Uttr1 ►Decimal⇒Uttrykk

$\frac{1}{3}$ ►Decimal	0.333333
------------------------	----------

Liste1 ►Decimal⇒Uttrykk

Matrise1 ►Decimal⇒Uttrykk

**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** *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 parameterne. 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**

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**).

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

## Define (Definer)

Katalog > 

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

**Merk:** Se også **Define BibPriv**, side 50 og **Define BibOff**, side 50.

```
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 BibPriv)

Katalog > 

**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 49 og **Define LibPub**, side 50.

## Define LibPub (Definer BibOff)

Katalog > 

**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 49 og **Define LibPriv**, side 50.

**deltaList()**

Se  $\Delta$ List(), side 108.

**deltaTmpCnv()**

Se  $\Delta$ tmpCnv(), side 200.

**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 a	Done
----------	------

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

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 208.

**DelVar**

katalog &gt;

**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 &gt;

**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 263.

**derivative()**Se *d()*, side 232.**deSolve()**

Katalog &gt;

**deSolve**(*1.el2.ordensODE, Var, avhVar*)⇒*en generell løsning*

deSolve( $y''+2\cdot y'+y=x^2, x, y$ )	
	$y=(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$
right( <i>Ans</i> )→ <i>temp</i>	$(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$
$\frac{d^2}{dx^2}(temp)+2\cdot\frac{d}{dx}(temp)+temp-x^2$	0
DelVar <i>temp</i>	Done

Returnerer en ligning som eksplisitt eller implisitt spesifiserer en generell løsning til 1.- eller 2.-ordens ordinær differensialligning (ODE). I ODE:

- Bruk et apostrofsymbol (trykk på ) for å markere den første deriverte av den avhengige variabelen med hensyn på den uavhengige variabelen.
- Bruk to apostrofsymboler for å markere den tilsvarende andre deriverte.

Symbolet ' brukes bare for deriverte innenfor deSolve(). I andre tilfeller, bruk **d()**.



Den generelle løsningen av en førsteordens ligning inneholder en vilkårlig konstant av formen  $ck$ , hvor  $k$  er et heltall mellom 1 og 255. Løsningen av en andreordens ligning inneholder to slike konstanter.

Bruk **solve()** på en implisitt løsning hvis du vil prøve å omregne den til en eller flere ekvivalente, eksplisitte løsninger.

Når du sammenlikner resultatene dine med løsningene i et oppgavehefte eller i en håndbok, bør du være klar over at ulike metoder introduserer vilkårlige konstanter ved forskjellige trinn i beregningen, og dette kan frembringe ulike, generelle løsninger.

**deSolve(1.ordensODE and startBet, Var, avhVar)**  $\Rightarrow$  en bestemt løsning

Returnerer en bestemt løsning som tilfredsstiller 1.ordensODE og startBet. Dette er vanligvis enklere enn å bestemme en generell løsning, bytte ut startverdier, finne løsning for den vilkårlige konstanten og deretter sette denne verdien inn i den generelle løsningen.

startBet er en ligning på formen:

avhVar (startUavhengigVerdi) = startAvhengigVerdi

startUavhengigVerdi og startAvhengigVerdi kan være variabler, f.eks.  $x_0$  og  $y_0$ , som ikke har noen lagret verdi. Implisitt derivasjon kan være en hjelp til å verifisere implisitte løsninger.

$$\text{deSolve}(y'=(\cos(y))^2 \cdot x, x, y) \quad \tan(y) = \frac{x^2}{2} + c4$$

$$\text{solve}(Ans, y) \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot c4}{2}\right) + n3 \cdot \pi$$

$$\text{Ans} \{c4=c-1 \text{ and } n3=0\} \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot (c-1)}{2}\right)$$

$$\sin(y) = (y \cdot e^x + \cos(y)) \cdot y' \rightarrow \text{ode} \quad \sin(y) = (e^x \cdot y + \cos(y)) \cdot y'$$

$$\text{deSolve}(\text{ode and } y(0)=0, x, y) \rightarrow \text{soln} \quad \frac{-(2 \cdot \sin(y) + y^2)}{2} = (e^x - 1) \cdot e^{-x} \cdot \sin(y)$$

$$\text{soln} | x=0 \text{ and } y=0 \quad \text{true}$$

$$\text{ode} | y' = \text{impDif}(\text{soln}, x, y) \quad \text{true}$$

$$\text{DelVar } \text{ode}, \text{soln} \quad \text{Done}$$

**deSolve**

**(2.ordensODEandstartBet1andstartBet2, Var, avhVar)** ⇒ en bestemt løsning

Returnerer en bestemt løsning som tilfredsstiller 2. ordens ODE og har en spesifisert verdi av den avhengige variabelen og dens første deriverte i ett punkt.

For *startBet1* bruker du formen:

*avhVar* (*startUavhengigVerdi*) = *startAvhengigVerdi*

For *startBet2* bruker du formen:

*avhVar* (*startUavhengigVerdi*) = *start1.DerivertVerdi*

**deSolve(2.ordensODEand grenseBet1andgrenseBet2, Var, avhVar)** ⇒ en bestemt løsning

Returnerer en bestemt løsning som tilfredsstiller 2.ordensODE og har spesifiserte verdier ved to ulike punkter.

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

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

$$\text{solve}\left(\frac{2 \cdot y^{\frac{4}{3}}}{3} = t, y\right)$$

$$y = \frac{3 \cdot 3^{\frac{1}{3}} \cdot 2^{\frac{2}{3}} \cdot t^{\frac{3}{4}}}{4} \text{ and } t \geq 0$$

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

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

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

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

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

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

**det()**

Katalog &gt;

**det**(*kvadratMatrise*[,  
*Toleranse*])⇒*uttrykk*Returnerer determinanten til  
*kvadratMatrise*.

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 ignorerer *Toleranse*.

- 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:

$$5E-14 \cdot \text{maks}(\text{dim}(\text{kvadratMatrise})) \cdot \text{radNorm}(\text{kvadratMatrise})$$

$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$a \cdot d - b \cdot c$
$\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	-2
$\det\left(\text{identity}(3) - x \cdot \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{bmatrix}\right)$	$-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$
$\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{matI}$	$\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix}$
$\det(\text{matI})$	0
$\det(\text{matI}, 1)$	1.E20

**diag()**

Katalog &gt;

**diag**(*Liste*)⇒*matrise***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*.*kvadratMatrise* må være kvadrat.

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

## dim()

Katalog > **dim(Liste)**⇒heltall

dim({0,1,2}) 3

Returnerer dimensjonen av *Liste*.**dim(Matrise)**⇒listedim( $\begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$ ) {3,2}

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

**dim(Streng)**⇒heltall

dim("Hello") 5

Returnerer antallet tegn som er inneholdt i tegnstrengen *Streng*.

dim("Hello "&amp;"there") 11

## Disp (Vis)

Katalog > **Disp uttrElStreng1 [, uttrElStreng2] ...**

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

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 produktboken.

```
chars(240,243)
-----
240 ø
241 ñ
242 ò
243 ó
-----
Done
```

## DispAt

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

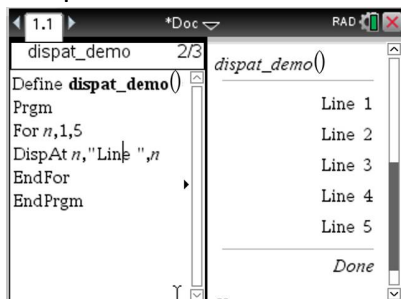
DispAt

**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 skjermbildet, men kun området som følger umiddelbart etter kommandoen/programmet.

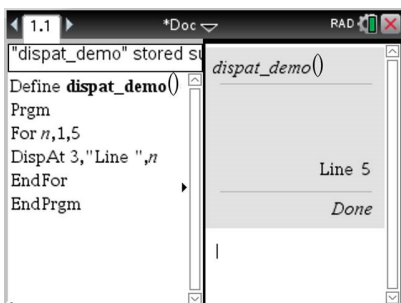
### Eksempel



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.

**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.



**Illustrerende eksempler:**

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

EndPrgm

## Feilmeldinger:

Feilmelding	Beskrivelse
DispAt-linjennummeret må være mellom 1 og 8	Uttrykk evaluerer linjennummeret 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" iverksettes for den tomme verdien (hvis oppkall er definert)
Omregningsoperator: DispAt 2_ft @>_m, "Hei alle sammen"	<b>CAS:</b> Datatypefeilen iverksettes (hvis oppkall er definert) <b>Numerisk:</b> Konverteringen evalueres og hvis resultatet er et gyldig argument, skriver DispAt strengen på resultatlinjen.

## ►DMS (GMS)

Uttr ►DMS

I Grader-vinkelmodus:

Liste ►DMS

(45.371)►DMS	45°22'15.6"
{{(45.371,60)}►DMS	{45°22'15.6",60°}

Matrise ►DMS

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

Tolker argumentet som en vinkel og viser ekvivalenten DMS (GGGGG°MM' SS.ss")-tallet. Se °, ', " (side 240) 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.

**domain() (område)**

Katalog &gt;

**domain(Uttr1, Var) ⇒ uttrykk**Returnerer definisjonsområdet for *Uttr1* med hensyn på *Var*.**domain()** kan brukes til å undersøke definisjonsområder for funksjoner. Det er begrenset til ekte og endelig område.

Denne funksjonaliteten har begrensninger grunnet for svake algebraiske forenklings- og løsningsalgoritmer på datamaskinen.

Enkelte funksjoner kan ikke brukes som argumenter for **domain()**, uavhengig av om de vises eksplisitt eller i brukerdefinerte variabler og funksjoner. Uttrykket kan ikke forenkles i det følgende eksemplet, fordi *f()* er en funksjon som ikke er tillatt.

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

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

**dominantTerm() (dominerende ledd)**

Katalog &gt;

**dominantTerm(Uttr1, Var [, Punkt]) ⇒ uttrykk****dominantTerm(Uttr1, Var [, Punkt]) | Var > Punkt ⇒ uttrykk****dominantTerm(Uttr1, Var [, Punkt]) | Var < Punkt ⇒ uttrykk**

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

Returnerer `dominantTerm` i en potensrekkepresentasjon av *Utr1* som er utvidet rundt *Punkt*. `DominantTerm` er det hvis størrelse vokser raskest i nærheten av *Var = Punkt*. Resulterende potens av (*Var - Punkt*) kan ha en negativ eksponent og/eller en brøkeksponent. Koeffisienten foran denne potensen kan inkludere logaritmer av (*Var - Punkt*) og andre funksjoner av *Var* som er dominert av alle potensene til (*Var - Punkt*) som har samme eksponenttegn (-sign).

*Punkt* grunninnstilles til 0. *Punkt* kan være  $\infty$  eller  $-\infty$ , i så fall vil det dominerende leddet være det leddet som har den største eksponenten av *Var* istedenfor den minste eksponenten av *Var*.

`dominantTerm(...)` returnerer "`dominantTerm(...)`" hvis det ikke er i stand til å bestemme en slik representasjon, som for vesentlige singularpunkt, f.eks.  $\sin(1/z)$  ved  $z=0$ ,  $e^{-1/z}$  ved  $z=0$ , eller  $e^z$  ved  $z = \infty$  eller  $-\infty$ .

Dersom rekken eller en av dens deriverte har en "hopp-diskontinuitet" ved *Punkt*, er det sannsynlig at resultatet inneholder deluttrykk av formen `tegn(...)` eller `abs(...)` for en reell utvidelsesvariabel eller `(-1)^nedre(...vinkel(...))` for en sammensatt utvidelsesvariabel, som er en som ender med "\_". Dersom du vil bruke det dominerende leddet kun for verdier på en side av *Punkt*, så utvider du `dominantTerm(...)` med det passende "`| Var > Punkt`", "`| Var < Punkt`", "`| Var ≥ Punkt`" eller "`Var ≤ Punkt`" for å oppnå et enklere resultat.

`dominantTerm()` fordeler over 1. argument-liste og matriser.

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



## dominantTerm() (dominerende ledd)

Katalog > 

**dominantTerm()** er nyttig når du vil vite det enklest mulige uttrykket som er asymptotisk til et annet uttrykk som  $Var \rightarrow Punkt$ . **dominantTerm()** er også nyttig når det ikke er opplagt hva graden av det første leddet som ikke er null i en rekke vil bli og du ikke iterativt vil gjetten enten interaktivt eller med en programmert loop.

**Merk:** Se også **rekke()**, side 170.

## dotP() (prikkP)

Katalog > 

**dotP(Liste1, Liste2)**  $\Rightarrow$  uttrykk

Returnerer "prikk"produktet av to lister.

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

**dotP(Vektor1, Vektor2)**  $\Rightarrow$  uttrykk

Returnerer "prikk"produktet av to vektorer.

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

Begge må være radvektorer, eller begge må være kolonnevektorer.

## E

## e^()


 **tast**

**e^(Uttr1)**  $\Rightarrow$  uttrykk

Returnerer e opphøyd i *Uttr1*-potens.

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

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

**Merk:** Å trykke på  for å vise  $e^()$  er forskjellig fra å trykke på tegnet **E** på tastaturet.

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.

## $e^{\wedge}()$

$e^x$  tast

$e^{\wedge}(Liste1) \Rightarrow liste$

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

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

$e^{\wedge}(kvadratMatrise1) \Rightarrow kvadratMatrise$

1	5	3	782.209	559.617	456.509
4	2	1	680.546	488.795	396.521
6	-2	1	524.929	371.222	307.879

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.

## eff()

Katalog >

$eff(nominellRente, CpY) \Rightarrow verdi$

$eff(5.75,12)$  5.90398

Finansiell 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 130.

## eigVc() (egenvektor)

Katalog >

$eigVc(kvadratMatrise) \Rightarrow matrise$

I rektangulært, kompleks format:

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:

$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$
$eigVc(mI)$	$\begin{bmatrix} -0.800906 & 0.767947 & ( \\ 0.484029 & 0.573804+0.052258 \cdot i & 0.5738 \\ 0.352512 & 0.262687+0.096286 \cdot i & 0.2626 \end{bmatrix}$

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

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

*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.

## eigVl() (egenverdi)

**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:

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

$$\text{eigVl}(m1) \\ \{-4.40941, 2.20471 + 0.763006 \cdot i, 2.20471 - 0.763006 \cdot i\}$$

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

## Else

## Elseif

**If** *BoolskUttr1* **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  $g(x)$  = Func

If  $x \leq 5$  Then

Return 5

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

Return  $-x$

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

Return  $x$

ElseIf  $x = 10$  Then

Return 3

EndIf

EndFunc

*Done*

**EndFor**

Se For, side 77.

**EndFunc**

Se Func, side 81.

**EndIf**

Se If, side 92.

**EndLoop**

Se Loop, side 116.

**EndPrgm**

Se Prgm, side 144.

**EndTry**

Se Try, side 201.

**EndWhile**

Se While, side 211.

**euler ()**Katalog > 

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

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

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

Differensialligning:

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

euler(0.001·y·(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.

Bruker Eulers metode for å løse systemet

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

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

$\text{Uttr}$  er høyre side, som definerer den ordinære differensialligningen (ODE).

$\text{SystemAvUttr}$  er systemet på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i  $\text{ListeMedAvhVarer}$ ).

$\text{ListeMedUttr}$  er en liste på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i  $\text{ListeMedAvhVarer}$ ).

$\text{Var}$  er den uavhengige variabelen.

$\text{ListeMedAvhVarer}$  er en liste over avhengige variabler.

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

$\text{ListeMedAvhVarer}$  er en liste over startverdier for avhengige variabler.

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

Sammenlign resultatet over med eksakt løsning i CAS som ble funnet ved hjelp av  $\text{deLøs}()$  og  $\text{sekvGen}()$ :

$$\text{deSolve}(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

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.

*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*)** ⇒ *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.

```
fadein()                                Done
```

## eval ()

Hub-meny

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 82), **GetStr** (side 89) og **Send** (side 167).

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

## exact()

Katalog >

**exact**(*Uttr1* [, *Toleranse*])⇒*uttrykk*

**exact**(*Liste1* [, *Toleranse*])⇒*liste*

**exact**(*Matrise1* [, *Toleranse*])⇒*matrise*

Bruker aritmetisk eksakt-modus til om mulig å returnere argumentet uttrykt som et rasjonalt tall.

*Toleranse* spesifiserer toleransen for omregningen. Standard er 0 (null).

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

## 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 produkhåndboken.

### 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
```

*g()* 21

**Uttr** ►exp

Viser *Uttr* uttrykt ved det naturlige grunntallet *e*. Dette er en konverteringsoperator. Den kan bare brukes på slutten av kommandolinjen.

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

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

**exp()**

 **tast**

**exp(Uttr1)⇒Uttrykk**

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

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

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**

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

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

**exp(kvadratMatrise1)⇒kvadratMatrise**

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()**.

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

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



**exp►list**(*Uttr*,*Var*)⇒*liste*

Undersøker *Uttr* for ligninger som er adskilt med ordet "or," og returnerer en liste som inneholder de høyre sidene i ligningen med formen  $Var=Uttr$ . Dette gir deg en enkel måte å trekke ut noen løsningsverdier som er implementert i resultatene av funksjonene **solve()**, **cSolve()**, **fMin()** og **fMax()**.

**Merk:** **exp►list()** er ikke nødvendig ved **zeros-** og **cZeros()**-funksjonene, fordi de returnerer en liste av løsningsverdier direkte.

Du kan sette inn denne funksjonen fra tastaturet ved å skrive **exp@>list (...)**.

$$\frac{\text{solve}(x^2-x-2=0,x)}{\text{exp►list}(\text{solve}(x^2-x-2=0,x),x)} \quad \begin{matrix} x=1 \text{ or } x=2 \\ \{-1,2\} \end{matrix}$$

## expand() (utvid)

**expand**(*Uttr1* [, *Var*])⇒*uttrykk*

**expand**(*Liste1* [,*Var*])⇒*liste*

**expand**(*Matrise1* [,*Var*])⇒*matrise*

**expand**(*Uttr1*) returnerer *Uttr1* utvidet med hensyn på alle variablene.

Utvidelsen er polynomisk utvidelse for polynomer og delbrøkkopp spalting for rasjonale uttrykk.

Hensikten med **expand()** er å omforme *Uttr1* til en sum og/eller differanse av enkle ledd. Derimot er hensikten med **factor()** å omforme *Uttr1* til et produkt og/eller koeffisient av enkle faktorer.

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

**expand(Uttr1,Var)** returnerer *Uttr1* utvidet med hensyn på *Var*. Liknende potenser av *Var* er samlet sammen. Leddene og faktorene deres er sortert med *Var* som hovedvariabel. Det kan forekomme faktoriseringer eller utvidelser av innsamlede koeffisienter. Sammenliknet med å utelate *Var* sparer dette ofte tid samt plass både i minnet og på skjermen, samtidig som uttrykket blir mer forståelig.

Hvis du bruker *Var*, kan dette gjøre faktoriseringen av nevneren som brukes for å spalte en delbrøk mer fullstendig, selv om det bare er én variabel.

Tips: For rasjonale uttrykk er **propFrac()** et raskere, men mindre ekstremt alternativ til **expand()**.

**Merk:** Se også **comDenom()** for en utvidet teller over en utvidet nevner.

**expand(Uttr1,[Var])** oppløser også logaritmer og brøkpotenser uavhengig av *Var*. For økt oppløsning i logaritmer og brøkpotenser kan ulikhetsbegrensninger være nødvendige for å garantere at noen av faktorene er ikke-negative.

**expand(Uttr1, [Var])** oppløser også absoluttverdier, **sign()** og eksponenter, uavhengig av *Var*.

**Merk:** Se også **tExpand()** for trigonometrisk vinkelsum og flervinklet utvidelse.

$\text{expand}\left(\frac{(x+y+1)^2}{y}, y\right)$	$y^2+2\cdot y\cdot(x+1)+(x+1)^2$
$\text{expand}\left(\frac{(x+y+1)^2}{x}, x\right)$	$x^2+2\cdot x\cdot(y+1)+(y+1)^2$
$\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}, y\right)$	$\frac{1}{y-1} + \frac{1}{y} + \frac{1}{x\cdot(x-1)}$
$\text{expand}(Ans,x)$	$\frac{1}{x-1} + \frac{1}{x} + \frac{1}{y\cdot(y-1)}$
$\text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right)$	$\frac{2\cdot x}{x^2-2} + x+1$
$\text{expand}(Ans,x)$	$\frac{1}{x-\sqrt{2}} + \frac{1}{x+\sqrt{2}} + x+1$

$\ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y}$	$\ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y}$
$\text{expand}(Ans)$	$\ln(x\cdot y)+\sqrt{2}\cdot\sqrt{x\cdot y}+\ln(2)$
$\text{expand}(Ans),y\geq 0$	$\ln(x)+\sqrt{2}\cdot\sqrt{x}\cdot\sqrt{y}+\ln(y)+\ln(2)$
$\text{sign}(x\cdot y)+ x\cdot y +e^{2\cdot x+y}$	$e^{2\cdot x+y}+\text{sign}(x\cdot y)+ x\cdot y $
$\text{expand}(Ans)$	$\text{sign}(x)\cdot\text{sign}(y)+ x \cdot y +(e^x)^2\cdot e^y$

**expr (String)⇒Uttrykk**

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

$\text{expr}("1+2+x^2+x")$	$x^2+x+3$
$\text{expr}("expand((1+x)^2)")$	$x^2+2\cdot x+1$
"Define cube(x)=x^3" → <i>funcstr</i>	"Define cube(x)=x^3"
$\text{expr}(funcstr)$	Done
$\text{cube}(2)$	8

**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 186).

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 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 263).

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 <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> , og <i>Inkludert 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>inkludert kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

## F

### factor() (faktor)

Katalog > 

**factor**(*Uttr1*[, *Var*]) ⇒ *Uttrykk*

**factor**(*Liste1*[, *Var*]) ⇒ *liste*

**factor**(*Matrise1*[, *Var*]) ⇒ *matrise*

**factor**(*Uttr1*) returnerer *Uttr1* faktorisert med hensyn på alle dens variabler over en felles nevner.

*Uttr1* er faktorisert så mye som mulig i lineære rasjonale faktorer uten å innføre nye, ikke-reelle deluttrykk. Med dette alternativet kan du faktorisere med hensyn på mer enn en variabel.

**factor**(*Uttr1*, *Var*) returnerer *Uttr1* faktorisert med hensyn på variabel *Var*.

*Uttr1* er faktorisert så mye som mulig mot reelle faktorer som er lineære i *Var*, selv om det innfører irrasjonale konstanter eller deluttrykk som er irrasjonelle i andre variabler.

Faktorene og leddene deres er sortert med *Var* som hovedvariabel. Liknende potenser av *Var* er samlet sammen i hver faktor. Inkluder *Var* hvis du må faktorisere med hensyn på bare den ene variabelen og du er villig til å akseptere irrasjonale uttrykk i en annen tilfeldig variabel for å øke faktoriseringen med hensyn på *Var*. Det kan hende at faktor bestemmes tilfeldig med hensyn på andre variabler.

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

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



For automatisk innstilling av modusen **Auto** eller **Tilnærmet**, vil en inkludering av *Var* også gjøre det mulig å tilnærme med flytende desimalpunkt-koeffisienter der hvor irrasjonelle koeffisienter ikke kan uttrykkes eksplisitt utfra innebygde funksjoner. Selv dersom det bare er én variabel, vil man kunne oppnå en mer komplett faktorisering ved å inkludere *Var*.

**Merk:** Se også **comDenom()** for en rask måte å oppnå delvis faktorisering på, hvis **factor()** er for langsam eller hvis den tar for stor plass i minnet.

**Merk:** Se også **cFactor()** for å faktorisere overalt med komplekse koeffisienter i letingen etter lineære faktorer.

**factor(rasjonalTall)** 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.

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 *rasjonalTall* ikke er et primtall og hvis den nest største faktoren består av mer enn fem siffer.

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

$$\frac{\text{factor}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3)}{(x-0.964673)\cdot(x+0.611649)\cdot(x+2.12543)\cdot(x^2)}$$

factor(152417172689)	123457·1234577
isPrime(152417172689)	false

## 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 *øvGrense* for spesifisert *dfTeller* (frihetsgrader) og *dfNevner*.

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

## Fill (Fyll)

Fill *Uttr*, *matriseVar* ⇒ *matrise*

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

*matriseVar* må eksistere allerede.

1 2	→ <i>amatrix</i>	1 2
3 4		3 4

Fill 1.01, <i>amatrix</i>	<i>Done</i>
---------------------------	-------------

<i>amatrix</i>	1.01 1.01
	1.01 1.01

Fill *Uttr*, *listeVar* ⇒ *liste*

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

*listeVar* må eksistere allerede.

{1,2,3,4,5}	→ <i>alist</i>	{1,2,3,4,5}
-------------	----------------	-------------

Fill 1.01, <i>alist</i>	<i>Done</i>
-------------------------	-------------

<i>alist</i>	{1.01,1.01,1.01,1.01,1.01}
--------------	----------------------------

## FiveNumSummary

FiveNumSummary *X*[,*[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 186.)

*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 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 263.

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

## floor() (nedre)

**floor(Utt1) ⇒ 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**

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

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

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

**fMax()**Katalog > **fMax**(*Uttr*, *Var*) $\Rightarrow$ Boolsk uttrykk

$fMax(1-(x-a)^2-(x-b)^2, x)$	$x = \frac{a+b}{2}$
------------------------------	---------------------

**fMax**(*Uttr*, *Var*, *nedGrense*)

$fMax(.5 \cdot x^3 - x - 2, x)$	$x = \infty$
---------------------------------	--------------

**fMax**(*Uttr*, *Var*, *nedGrense*, *øvgrense*)**fMax**(*Uttr*, *Var*) | *nedGrense*  $\leq$  *Var*  $\leq$  *øvgrense*

Returnerer et Boolsk uttrykk som spesifiserer alternativverdier av *Var* som maksimerer *Uttr* eller lokaliserer den minste øvre grensen.

Du kan bruke ("|")-operatoren for å begrense løsningsintervallene og/eller spesifisere begrensninger.

$fMax(0.5 \cdot x^3 - x - 2, x)   x \leq 1$	$x = 0.816497$
---	----------------

For Tilnærmet innstilling av modusen **Auto** eller **Tilnærmet** søker **fMax()** iterativt etter tilnærmet lokalt maksimum. Dette er ofte raskere, særlig hvis du bruker "|" operatoren for å begrense søket til et relativt lite intervall som inneholder eksakt ett lokalt maksimum.

**Merk:** Se også **fMin()** og **max()**.

**fMin()**Katalog > **fMin**(*Uttr*, *Var*) $\Rightarrow$ Boolsk uttrykk

$fMin(1-(x-a)^2-(x-b)^2, x)$	$x = -\infty$ or $x = \infty$
------------------------------	-------------------------------

**fMin**(*Uttr*, *Var*, *nedGrense*)

$fMin(0.5 \cdot x^3 - x - 2, x)   x \geq 1$	$x = 1.$
---	----------

**fMin**(*Uttr*, *Var*, *nedGrense*, *øvgrense*)**fMin**(*Uttr*, *Var*) | *nedGrense*  $\leq$  *Var*  $\leq$  *øvgrense*

Returnerer et Boolsk uttrykk som spesifiserer alternativverdier av *Var* som minimerer *Uttr* eller lokaliserer den største nedre grensen.

Du kan bruke ("|")-operatoren for å begrense løsningsintervallene og/eller spesifisere begrensninger.



For Tilnærmet innstilling av modusen **Auto eller Tilnærmet**, søker **fMin()** iterativt etter tilnærmet lokalt minimum. Dette er ofte raskere, særlig hvis du bruker “|” operatoren for å begrense søket til et relativt lite intervall som inneholder eksakt ett lokalt minimum.

**Merk:** Se også **fMax()** og **min()**.

## For

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

*Blokk*

## 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.

*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 produktåndboken.

```
Define g() $\Rightarrow$ Func                               Done
Local tempsum,step,i
0  $\rightarrow$  tempsum
1  $\rightarrow$  step
For i,1,100,step
tempsum+i  $\rightarrow$  tempsum
EndFor
EndFunc
```

*g()* 5050

## format()

**format**(*Uttr* [, *formatStreng*]) $\Rightarrow$ *streng*

Returnerer *uttrykk* som en tegnstring basert på formatsjablonen.

*Uttrykket* må forenkles til et tall.

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

```
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"
```

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.

### fPart() (funksjonsdel)

fPart(Uttr1)⇒uttrykk

fPart(-1.234)	-0.234
fPart({1,-2.3,7.003})	{0,-0.3,0.003}

fPart(Liste1)⇒liste

fPart(Matrise1)⇒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()

FPdf(XVerdi,dfTeller,dfNevner)⇒tall hvis XVerdi er et tall, liste hvis XVerdi er en liste

FPdf(XVerdi,dfTeller,dfNevner)⇒tall hvis XVerdi er et tall, liste hvis XVerdi er en liste

Beregner  $F$  fordelingssannsynligheten mellom  $X$  verdi for den spesifiserte  $df$  Teller (grader av frihet) og  $df$  Nevner.

## freqTable▶liste()

## freqTable▶liste

(Liste1, frekvHeltallListe)⇒liste

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.

*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 datamaskintastaturet ved å skrive **freqTable@>list (...)**.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 263.

freqTable▶list({1,2,3,4},{1,4,3,1})	
	{1,2,2,2,2,3,3,3,4}
freqTable▶list({1,2,3,4},{1,4,0,1})	
	{1,2,2,2,2,4}

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

datalist={1,2,e,3, $\pi$ ,4,5,6,"hello",7}	
	{1,2,2.71828,3,3.14159,4,5,6,"hello",7}
frequency(datalist,{2.5,4.5})	{2,4,3}

Forklaring til resultat:

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

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

Hvert element av resultatet samsvarer med antallet elementer fra *Liste1* som er i området for den stolpen. Uttrykt med begrep fra **countlf()**-funksjonen er resultatet { countlf(liste, ?≤b(1)), countlf(liste, b(1)<?≤b(2)), ..., countlf(liste, b(n-1)<?≤b(n)), countlf(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 263.

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

**Merk:** Se også **countlf()**, side 37.

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 186).

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 263.

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 = $n1-1$
stat.dfDenom	frihetsgrad for nevner = $n2-1$
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)

Katalog >

### Func

*Blokk*

### EndFunc

Sjablon for oppretting av brukerdefinert funksjon.

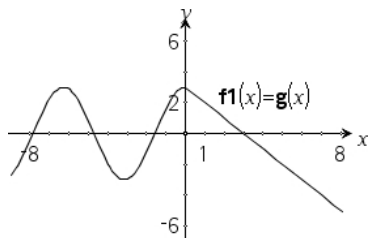
*Blokk* kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet, eller en rekke med utsagn på separate linjer. Funksjonen kan bruke **Returner**-kommandoen for å returnere et spesifikt resultat.

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

Definere en sammensatt funksjon:

```
Define g(x)=Func Done
  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)

Katalog >

**gcd**{Tall1, Tall2} $\Rightarrow$ uttrykk

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.

## gcd() (største felles divisor)

Katalog > 

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

**gcd**(*Liste1*, *Liste2*) $\Rightarrow$ *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*) $\Rightarrow$ *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()

Katalog > 

**geomCdf**(*p*,*nedreGrense*,*øvreGrense*) $\Rightarrow$ *tall*  
hvis *nedreGrense* og *øvreGrense* er tall,  
*liste* hvis *nedreGrense* og *øvreGrense* er lister

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

Beregner en kumulativ geometrisk sannsynlighet fra *nedreGrense* til *øvreGrense* med den spesifiserte sannsynligheten for suksess *p*.

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

## geomPdf()

Katalog > 

**geomPdf**(*p*,*XVerdi*) $\Rightarrow$ *tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner en sannsynlighet ved *XVerdi*, antall forsøk før første suksess inntreffer, for diskret geometrisk fordeling med spesifisert suksess-sannsynligheten *p*.

## Get

Hub-meny

**Get**[*ledetekstStreng*,]*var*[, *statusVar*]

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. Bruk **Get** for å hente verdien og tildele den til variabelen *lysver*.

**Get**[*ledetekstStreng*,] *funk*(*arg1*,  
...*argn*)[, *statusVar*]

Programmeringskommando: Henter en verdi fra en tilkoblet TI-Innovator™ Hub og tildeler verdien til variabel *var*.

Verdien må etterspørres:

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

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 89 og **Send**, side 167.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

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

Get "READ BRIGHTNESS", <i>lightval</i>	Done
<i>lightval</i>	0.378441

getDenom(Uttr1) ⇒ uttrykk

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

$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	$y-3$
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	$x \cdot y$

## getKey()

getKey([0|1]) ⇒ returnString

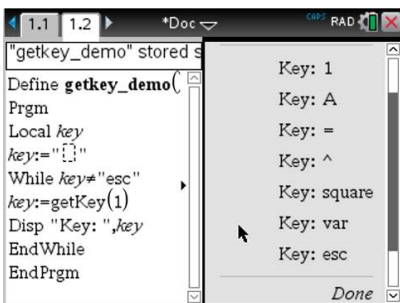
**Beskrivelse:** getKey() – tillater at et TI-Basic-program henter tastaturinnndata – grafregner, stasjonær PC og emulator på skrivebordet.

## 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.

getKey()

Eksempel 1:



## 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"



Håndholdt enhet / emulator tast	Stasjonær PC	Returverdi
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"
=	=	"_="
trigonometri	Ikke relevant	"trigonometri"
0 til og med 9	0-9	"0" ... "9"
Sjabloner	Ikke relevant	"sjablon"
Katalog	Ikke relevant	"kat"
^	^	"^"
X <sup>2</sup>	Ikke relevant	"kvadrat"
/ (divisjonstast)	/	"/"
* (multiplikasjonstast)	*	"*"
e <sup>x</sup>	Ikke relevant	"eksp"
10 <sup>x</sup>	Ikke relevant	"10potens"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	". "
(-)	Ikke relevant	"_" (negativ-tegn)
Enter	Enter	"enter"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
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"
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 skrivebordsstaster 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™)

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Student Software
		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"

## getLangInfo()

katalog > 

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 112, og **unlock**, side 208.

<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.

getMode(0)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1 }
getMode(1)	7
getMode(8)	1

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 171.

Modus Navn	Modus Heltall	Innstillinger 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, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær
Måleenheter	8	1=SI, 2=Eng/USA

## getNum() (lesTeller)

**getNum(*Uttr1*)** ⇒ *uttrykk*

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

$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	$x+y$

## GetStr

**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 82 og **Send**, side 167.

## getType()

Katalog > 

**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()

katalog > 

**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()	"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"> <tbody> <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> </tbody> </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")	<table border="1"> <tbody> <tr> <td>[<i>volcy12</i></td> <td>"NONE"</td> <td>"LibPub"</td> <td>0]</td> </tr> </tbody> </table>	[ <i>volcy12</i>	"NONE"	"LibPub"	0]								
[ <i>volcy12</i>	"NONE"	"LibPub"	0]										

**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.

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.

<i>a</i> :=1	1												
<i>b</i> :=[1 2]	[1 2]												
<i>c</i> :=[1 3 7]	[1 3 7]												
<i>vs</i> :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										
<i>vs</i> [1]	[1 "NUM" "[ ]" 0]												
<i>vs</i> [1,1]	1												
<i>vs</i> [2]	"Error: Invalid list or matrix"												
<i>vs</i> [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 produkt håndboken.

```

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

```

*g()* 55

*Uttr1* ► Grad ⇒ *Uttrykk*

I Grader-vinkelmodus:

Omregner *Uttr1* til gradian vinkelmåling.

(1.5) ► Grad (1.66667)<sup>g</sup>

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

I Radian-vinkelmodus:

(1.5) ► Grad (95.493)<sup>g</sup>

/

## identity()

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

**Hvis** *BooleanExpr*  
*Utsagn*

Define  $g(x) = \text{Func}$  Done  
If  $x < 0$  Then  
Return  $x^2$   
EndIf  
EndFunc

**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 flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

$g(-2)$  4



**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.

*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	Done
If $x < 0$ Then	
Return $-x$	
Else	
Return $x$	
EndIf	
EndFunc	

$g(12)$	12
$g(-12)$	12

Define $g(x)$ =Func	
If $x < 5$ Then	
Return 5	
ElseIf $x > 5$ and $x < 0$ Then	
Return $-x$	
ElseIf $x \geq 0$ and $x \neq 10$ Then	
Return $x$	
ElseIf $x = 10$ Then	
Return 3	
EndIf	
EndFunc	

	Done
$g(-4)$	4
$g(10)$	3

## ifFn()

**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*

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

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.

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 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.

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

$$\overline{\text{iffn}(\{1,2,3\}<2.5,4\{8,9,10\})} \quad \{4,4,10\}$$

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

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

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

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

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

## imag()

**imag**(*Expr1*) ⇒ *uttrykk*

Returnerer den imaginære delen av argumentet.

**Merk:** Alle ubestemte variabler behandles som reelle variabler. Se også *real()*, page 153

**imag**(*List1*) ⇒ *liste*

Returnerer en liste over de imaginære delene av elementene.

**imag**(*Matrix1*) ⇒ *matrise*

Returnerer en matrise over de imaginære delene av elementene.

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

$$\overline{\text{imag}(z)} \quad 0$$

$$\overline{\text{imag}(x+i\cdot y)} \quad y$$

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

$$\overline{\text{imag}\left(\begin{bmatrix} a & b \\ i\cdot c & i\cdot d \end{bmatrix}\right)} \quad \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

## impDif()

Katalog > 

**impDif**(*Equation*, *Var*, *dependVar*  
[, *Ord*]) ⇒ *uttrykk*

$\text{impDif}(x^2+y^2=100,x,y)$	$\frac{-x}{y}$
----------------------------------	----------------

der rekkefølgen *Ord* automatisk går til 1.

Beregner den implisitte deriverte for ligninger som inneholder en variabel som er definert implisitt med hensyn på en annen.

## Indireksjon

Se #(), side 238.

## inString()

Katalog > 

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

$\text{inString}(\text{"Hello there"}, \text{"the"})$	7
$\text{inString}(\text{"ABCEFG"}, \text{"D"})$	0

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*).

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

## int()

Katalog > 

**int**(*Expr*) ⇒ *heltall*

$\text{int}(-2.5)$	-3.
--------------------	-----

**int**(*List1*) ⇒ *liste*

**int**(*Matrix1*) ⇒ *matrise*

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

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.

## intDiv()

Katalog &gt;

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

**intDiv**(*List1*, *List2*) ⇒ liste

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

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

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

<b>intDiv</b> (-7,2)	-3
<b>intDiv</b> (4,5)	0
<b>intDiv</b> ({12,-14,-16},{5,4,-3})	{2,-3,5}

## integral

Se  $\int()$ , side 233.

## interpoler ()

Katalog &gt;

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

Denne funksjonen gjør følgende:

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 ≥ 2 og inneholde uttrykk som forenkles til tall.

*xValue* kan være en udefinert variabel, et tall eller en liste med tall.

Differensialligning:

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

$rk := rk23(-3 \cdot y + 6 \cdot t + 5, t, y, \{0, 10\}, 5, 1)$
$\left\{ \begin{array}{cccccc} 0. & 1. & 2. & 3. & 4. & \\ 5. & 3.19499 & 5.00394 & 6.99957 & 9.00593 & 10. \end{array} \right\}$

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

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

$xvaluelist := seq(i, i, 0, 10, 0.5)$
$\{0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6., 6.5\}$
$xlist := mat \blacktriangleright list(rk[1])$
$\{0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.\}$
$ylist := mat \blacktriangleright list(rk[2])$
$\{5., 3.19499, 5.00394, 6.99957, 9.00593, 10.9978\}$
$yprimelist := -3 \cdot y + 6 \cdot t + 5   y = ylist \text{ and } t = xlist$
$\{-10., 1.41503, 1.98819, 2.00129, 1.98221, 2.006\}$
$interpolate(xvaluelist, xlist, ylist, yprimelist)$
$\{5., 2.67062, 3.19499, 4.02782, 5.00394, 6.0001\}$

## invχ<sup>2</sup>()

Katalog &gt;

**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()

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()

invBinom  
(*CumulativeProb,NumTrials,Prob,OutputForm*) $\Rightarrow$  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 %?

invBinom $\left(0.77,30,\frac{1}{6}\right)$	6
invBinom $\left(0.77,30,\frac{1}{6},1\right)$	$\begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix}$

## invBinomN()

invBinomN(*CumulativeProb,Prob,NumSuccess,OutputForm*) $\Rightarrow$  skalar eller matrise

Eksempel: Monique øver på målskudd for nettbull. Fra erfaring vet hun at det er 70 % sjansje 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?

## invBinomN()

Katalog > 

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.

<code>invBinomN(0.01,0.7,49)</code>	86
<code>invBinomN(0.01,0.7,49,1)</code>	$\begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$

## invNorm()

Katalog > 

`invNorm(Area[,μ[,σ]])`

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

## invt()

Katalog > 

`invt(Area,df)`

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

## iPart()

Katalog > 

`iPart(Number)`  $\Rightarrow$  heltall

`iPart(List1)`  $\Rightarrow$  liste

`iPart(Matrix1)`  $\Rightarrow$  matrise

<code>iPart(-1.234)</code>	-1.
<code>iPart(<math>\left\{\frac{3}{2}, -2.3, 7.003\right\}</math>)</code>	{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()**Katalog > **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 121.

<i>list1</i> := { 6000, -8000, 2000, -3000 }	{ 6000, -8000, 2000, -3000 }
<i>list2</i> := { 2, 2, 2, 1 }	{ 2, 2, 2, 1 }
<b>irr</b> (5000, <i>list1</i> , <i>list2</i> )	-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

Hvis du bare vil fastslå hvorvidt *Number* er et primtall, bruk **isPrime()** i stedet for **factor()**. Det er mye raskere, særlig hvis *Tall* ikke er et primtall og hvis den nest største faktoren består av mer enn fem siffer.

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

<b>isPrime</b> (5)	true
<b>isPrime</b> (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 <b>isPrime</b> ( <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*

$a := \_$	$\_$
$\text{isVoid}(a)$	true
$\text{isVoid}(\{1, \_, 3\})$	{ false, true, false }

Returnerer sann eller usann for å vise om utsagnet er en åpen (tom) datatype.

For mer informasjon om tomme elementer, se side 263.

## 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.

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

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

## lcm() (mfm)

Katalog > 

**lcm(Tall1, Tall2)** ⇒ *uttrykk*

$\text{lcm}(6, 9)$	18
--------------------	----

**lcm(Liste1, Liste2)** ⇒ *liste*

$\text{lcm}\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$	$\left\{\frac{2}{3}, 14, 80\right\}$
---	--------------------------------------

**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 som 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*

`left(x<3)` *x*

Returnerer venstre side av en ligning eller ulikhet.

**libShortcut()**

**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

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}
```

Hvis du skal kopiere en variabelgruppe, se **CopyVar** (side 30).

Hvis du skal slette en variabelgruppe, se **DelVar** (side 51).

**limit() eller lim() (grense)**Katalog > 

**limit(Uttr1, Var, Punkt [,Retning])**⇒uttrykk

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

**limit(Liste1, Var, Punkt[, Retning])**⇒liste

$$\lim_{x \rightarrow 0^+} \left( \frac{1}{x} \right) \quad \infty$$

**limit(Matrise1, Var, Punkt [, Retning])**⇒matrise

$$\lim_{x \rightarrow 0} \left( \frac{\sin(x)}{x} \right) \quad 1$$

Returnerer etterspurt grense.

$$\lim_{h \rightarrow 0} \left( \frac{\sin(x+h) - \sin(x)}{h} \right) \quad \cos(x)$$

**Merk:** Se også **Grense-sjablon**, side 7.

$$\lim_{n \rightarrow \infty} \left( \left( 1 + \frac{1}{n} \right)^n \right) \quad e$$

*Retning:* negativ=fra venstre, positiv=fra høyre, ellers=begge. (Hvis utelatt, normeres *Retning* til begge.)

Grenser ved positiv  $\infty$  og ved negativ  $\infty$  omregnes alltid til ensidige grenser fra den endelige siden.

Avhengig av omstendighetene, returnerer **limit()** seg selv eller udef hvis den ikke kan definere en endelig grense. Dette trenger ikke å bety at det ikke eksisterer noen endelig grense. udef innebærer at resultatet enten er et ukjent tall med endelig eller uendelig størrelse, eller det er et helt sett av slike tall.

**limit()** bruker metoder, som L'Hopital's regel, så det finnes endelige grenser som den ikke kan bestemme. Hvis *Uttr1* inneholder udefinerte variabler utenom *Var*, kan det hende at du må begrense dem for å oppnå et mer nøyaktig resultat.

$$\lim_{x \rightarrow \infty} (a^x) \quad \text{undef}$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 1 \quad \infty$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 0 \text{ and } a < 1 \quad 0$$

Grenser kan være svært utsatte for avrundingsfeil. Hvis mulig, unngå Tilnærm-innstillingen i modusen **Auto** eller **Tilnærmet** og tilnærmede tall når du beregner grenser. Ellers kan det hende at grenser som skulle være null eller ha uendelig størrelse kanskje ikke har dette, og grenser som skulle ha endelig størrelse forskjellig fra null ikke har det.

## 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 186.)

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 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 263).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a + b \cdot x$
stat.a, stat.b	Regresjonskoeffisienter

Utdata-variabel	Beskrivelse
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 (lineær regresjon)

katalog > 

**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 186.)

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 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 263).

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 <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>

## LinRegtIntervals (lineær regresjon)

katalog > 

### 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 186.)

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 263).

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 feilmargen
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 feilmargen
stat.SE	Standardfeil for gjennomsnittlig respons
[stat.LowerPred, stat.UpperPred]	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargen
stat.SEPred	Standardfeil for prediksjonen
stat.ŷ	$a + b \cdot X$ Verdi

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  $i$  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 186).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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

Utdata-variabel	Beskrivelse
stat.r	Korrelasjonskoeffisient
stat.Resid	Residualene fra regresjonen

## linSolve()

Katalog > 

**linSolve**(SystemAvLineæreLign, Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x + 4 \cdot y = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}, \{x, y\}\right\}\right) \quad \left\{\begin{array}{l} 37 \\ 26 \end{array}, \begin{array}{l} 1 \\ 26 \end{array}\right\}$$

**linSolve**(LineærLign1 and LineærLign2 and ..., Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}, \{x, y\}\right\}\right) \quad \left\{\begin{array}{l} 3 \\ 2 \end{array}, \begin{array}{l} 1 \\ 6 \end{array}\right\}$$

**linSolve**({LineærLign1, LineærLign2, ...}, Var1, Var2, ...) ⇒ liste

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} + 4 \cdot \text{pear} = 23 \\ 5 \cdot \text{apple} - \text{pear} = 17 \end{array}, \{\text{apple}, \text{pear}\}\right\}\right) \quad \left\{\begin{array}{l} 13 \\ 3 \end{array}, \begin{array}{l} 14 \\ 3 \end{array}\right\}$$

**linSolve**(SystemAvLineæreLign, {Var1, Var2, ...}) ⇒ liste

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{array}, \{\text{apple}, \text{pear}\}\right\}\right) \quad \left\{\begin{array}{l} 36 \\ 13 \end{array}, \begin{array}{l} 114 \\ 13 \end{array}\right\}$$

**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.

## Δlist() (liste)

Katalog > 

**ΔList**(Liste1) ⇒ liste

$$\Delta\text{List}(\{20, 30, 45, 70\}) \quad \{10, 15, 25\}$$

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



## $\Delta$ list() (liste)

Katalog > 

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▶mat()

Katalog > 

**list▶mat**(*Liste* [, *elementerRerRad*]) ⇒ *matrise*

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**(...).

<code>list▶mat({1,2,3})</code>	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
<code>list▶mat({1,2,3,4,5},2)</code>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

## ▶ln

Katalog > 



*Uttr1* ▶ln ⇒ *Uttrykk*

Fører til at inndata *Uttr1* omregnes til et uttrykk som bare inneholder naturlige logaritmer (ln).

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

$\left(\log_{10}(x)\right) \blacktriangleright \ln$	$\frac{\ln(x)}{\ln(10)}$
---	--------------------------

## ln()

  **taster**

**ln**(*Uttr1*) ⇒ *Uttrykk*

**ln**(*Liste1*) ⇒ *liste*

Returnerer argumentets naturlige logaritme.

Hvis kompleks formatmodus er reell:

<code>ln(2.)</code>	0.693147
---------------------	----------

For en liste, returneres elementenes naturlige logaritmer.

$$\ln\{-3, 1.2, 5\}$$

"Error: Non-real calculation"

### 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.

Hvis kompleks formatmodus er rektangulær:

$$\ln\{-3, 1.2, 5\} \quad \{\ln(3)+\pi\cdot i, 0.182322, \ln(5)\}$$

I radian-vinkelmodus og rektangulært, kompleks format:

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

1.83145+1.73485 <i>i</i>	0.009193-1.49086
0.448761-0.725533 <i>i</i>	1.06491+0.623491 <i>i</i>
-0.266891-2.08316 <i>i</i>	1.12436+1.79018

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

### 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 186).

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 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 263).

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

**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

```

	<i>Done</i>
<i>rollcount</i> ()	16
<i>rollcount</i> ()	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 208 og **getLockInfo()**, side 88.

<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**(*Uttr1* [, *Uttr2*]) ⇒ *Uttrykk*

**log**(*Liste1* [, *Uttr2*]) ⇒ *liste*

Returnerer grunntallet -*Uttr2* argumentets logaritme.

**Merk:** Se også **Log-sjablon**, side 2.

For en liste, returneres grunntall -*Uttr2* for elementenes logaritme.

Hvis *Uttr2* utelates, brukes 10 som grunntall.

**log**(*kvadratMatrise1* [, *Uttr2*]) ⇒ *kvadratMatrise*

$\log_{10}(2.)$	0.30103
$\log_4(2.)$	0.5
$\log_3(10) - \log_3(5)$	$\log_3(2)$

Hvis kompleks formatmodus er reell:

$\log_{10}(\{-3, 1.2, 5\})$	Error: Non-real result
-----------------------------	------------------------

Hvis kompleks formatmodus er rektangulær:

$\log_{10}(\{-3, 1.2, 5\})$	$\left\{ \log_{10}(3) + 1.36438 \cdot i, 0.079181, \log_{10}(5) \right\}$
-----------------------------	---

I radian-vinkelmodus og rektangulært, kompleks format:

## log()

ctrl 10<sup>x</sup> taster

Returnerer matrisens grunntall- *Utr2* logaritme av *kvadratMatrise1*. Dette er ikke det samme som å beregne grunntallet- *Utr2* 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.

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{pmatrix} 0.795387+0.753438 \cdot i & 0.003993-0.6474 \cdot i \\ 0.194895-0.315095 \cdot i & 0.462485+0.2707 \cdot i \\ -0.115909-0.904706 \cdot i & 0.488304+0.7774 \cdot i \end{pmatrix}$$

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

## logbase

Katalog >

*Utr1*  $\blacktriangleright$  **logbase**(*Utr1*)  $\Rightarrow$  uttrykk

Fører til inndata Uttrykk som skal forenkles til et uttrykk som bruker grunntall *Utr1*.

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

$$\log_3(10) - \log_5(5) \blacktriangleright \log_{\text{base}(5)} \left( \frac{\log_5\left(\frac{10}{3}\right)}{\log_5(3)} \right)$$

## 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 186).

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 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 263).

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 $X$ -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>

## LogisticD

**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 186).

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 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 263).

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 $X$ -listen 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 $Y$ -listen 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**

*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.

**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
```

	<i>Done</i>
rollcount()	16
rollcount()	3

**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.

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$	$\rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
LU <i>m1</i> , <i>lower</i> , <i>upper</i> , <i>perm</i>		<i>Done</i>
<i>lower</i>		$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
<i>upper</i>		$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
<i>perm</i>		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$



**LU (= nedre/øvre)**

Katalog &gt;

- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:  
 $5E-14 \cdot \max(\dim(\text{Matrise})) \cdot \text{radNorm}(\text{Matrise})$

Faktorerende algoritme **LU** bruker delvis pivotering med radutvekslinger.

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
LU m1, lower, upper, perm	Done
lower	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$
upper	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$
perm	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

**M****mat▶list() (matrise til liste)**

Katalog &gt;

**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(...)**.

mat▶list( $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ )	{1,2,3}
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
mat▶list(m1)	{1,2,3,4,5,6}

**max()**

Katalog &gt;

**max(Uttr1, Uttr2)⇒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**

max(2.3,1.4)	2.3
max({1,2},{-4,3})	{1,3}
max({0,1,-7,1.3,0.5})	1.3
max( $\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}$ )	$\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$

Returnerer en radvektor som inneholder maksimumselementet av hver kolonne i *Matrise1*.

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 263.

**Merk:** Se også **fMax()** og **min()**.

## mean() (gjennomsnitt)

**mean(Liste[, frekvListe])** ⇒ uttrykk

Returnerer gjennomsnittet av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

**mean(Matrise1[, FrekvMatrise])** ⇒ *matrise*

Returnerer en radvektor av gjennomsnittet for alle kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 263.

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$	$\frac{5}{3}$

I rektangulært vektorformat:

$\text{mean}\left(\begin{bmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{bmatrix}\right)$	$[-0.133333 \quad 0.833333]$
$\text{mean}\left(\begin{bmatrix} 1 & 0 \\ 5 & 0 \\ -1 & 3 \\ 2 & -1 \\ 5 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} -2 & 5 \\ 15 & 6 \end{bmatrix}$
$\text{mean}\left(\begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 4 & 4 & 1 \\ 5 & 6 & 6 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 47 & 11 \\ 15 & 3 \end{bmatrix}$

## median()

**median(Liste[, frekvListe])** ⇒ uttrykk

Returnerer medianen av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

$\text{median}(\{0.2, 0.1, -0.3, 0.4\})$	0.2
--	-----

**median**(*Matrise* [, *frekvMatrise*]) ⇒ *matrise*

$$\text{median} \left( \begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix} \right) = [0.4 \quad -0.3]$$

Returnerer en radvektor som inneholder medianer av kolonnene i *Matrise*1.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise*1.

#### Merknader:

- Alle inndata i listen eller matrisen må forenkles til tall.
- Tomme (åpne) elementer i listen eller matrisen ignoreres. For mer informasjon om tomme elementer, se side 263.

## MedMed

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

Beregner median-median-linjeny =  $(m \cdot x + b)$  for listene *X* og *Y* med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 186.)

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 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 263).

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 <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>

### 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*.

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() (midtstreng)**

Katalog &gt;

*Antall* må være  $\geq 0$ . Hvis *Antall* = 0, returneres en tom liste.

**mid**(*kildeStrengListe*, *Start*[, *Antall*]) $\Rightarrow$ *liste*

Returnerer *Antall* strenger fra listen over strenger *kildeStrengListe*, begynnende med elementnummer *Start*.

mid({"A","B","C","D"},2,2)	
	{"B","C"}

**min() (minimum)**

Katalog &gt;

**min**(*Uttr1*, *Uttr2*) $\Rightarrow$ *Uttrykk*

min(2,3,1.4)	1.4
--------------	-----

**min**(*Liste1*, *Liste2*) $\Rightarrow$ *liste*

min({1,2},{-4,3})	{-4,2}
-------------------	--------

**min**(*Matrise1*, *Matrise2*) $\Rightarrow$ *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**(*Liste*) $\Rightarrow$ *Uttrykk*

min({0,1,-7,1.3,0.5})	-7
-----------------------	----

Returnerer minimumselementet av *Liste*.

**min**(*Matrise!*) $\Rightarrow$ *matrise*

min( $\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}$ )	$\begin{bmatrix} -4 & -3 & 0.3 \end{bmatrix}$
---	---

Returnerer en radvektor som inneholder minimumselementet av hver kolonne i *Matrise!*.

**Merk:** Se også **fMin()** og **max()**.

**mirr()**

Katalog &gt;

**mirr**

(*finansRente*, *reinvestRente*, *CF0*, *CFListe* [, *CFFrekv*])

Finansiell funksjon som returnerer modifisert rente av en investering.

*finansRente* er den renten som du betaler på kontantstrømbeløpene.

<i>list1</i> := {6000, 8000, 2000, -3000}	
	{6000, 8000, 2000, -3000}
<i>list2</i> := {2, 2, 2, 1}	
	{2, 2, 2, 1}
mirr(4.65, 12, 5000, <i>list1</i> , <i>list2</i> )	13.41608607

*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 99.

**mod**(*Uttr1*, *Uttr2*) ⇒ *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 157

**mRow() (mRad)**

Katalog &gt;

**mRow(Utr, Matrise1, Indeks)** ⇒ *matrise*

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks* i *Matrise1* multiplisert med *Utr*.

$$\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & \frac{-4}{3} \end{bmatrix}$$

**mRowAdd() (mRadAdd)**

Katalog &gt;

**mRowAdd(Utr, Matrise1, Indeks1, Indeks2)** ⇒ *matrise*

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks2* i *Matrise1* erstattet med:

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

$$\text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix}$$

$$Utr \times \text{rad Indeks1} + \text{rad Indeks2}$$
**MultReg**

katalog &gt;

**MultReg Y, X1[,X2[,X3,...[,X10]]]**

Beregner multiple lineære regresjoner av liste *Y* for listene *X2, X2, ..., X10*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 186).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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	$\hat{y}\text{List} = b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualene fra regresjonen

**MultRegIntervals**

katalog &gt;

**MultRegIntervals Y, X1[,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 186).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat. $\hat{y}$	Et punkttestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XVerListe</i>
stat.dfError	Feilens frihetsgrader
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, $\{b_0, b_1, b_2, \dots\}$
stat.Resid	Residualene fra regresjonen

## MultRegTests

**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 186).



For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

## 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
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	{ $b_0, b_1, \dots$ } 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.ŷList	$\hat{y}$ 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

**nand (ikke både...og)**

ctrl = -taster

*BoolskUttr1* **nand** *BoolskUttr2*  
returnerer *Boolsk uttrykk*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$x \geq 3$ nand $x \geq 4$	$x < 4$

*BoolskListe1* **nand** *BoolskListe2*  
returnerer *Boolsk liste*

*BoolskMatrise1* **nand** *BoolskMatrise2*  
returnerer *Boolsk matrise*

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.

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}

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).

**nCr() (antKomb)**Katalog > **nCr(Uttr1, Uttr2) ⇒ uttrykk**

For heltall  $Uttr1$  og  $Uttr2$  der  $Uttr1 \geq Uttr2 \geq 0$ , er **nCr()** et antall kombinasjoner av  $Uttr1$ , som tar  $Uttr2$  om gangen. (Dette er også kjent som en binomisk koeffisient.) Begge argumentene kan være heltall eller symbolske uttrykk.

nCr(z,3)	$\frac{z \cdot (z-2) \cdot (z-1)}{6}$
Ans z=5	10
nCr(z,c)	$\frac{z!}{c! \cdot (z-c)!}$
Ans	$\frac{1}{c!}$
nPr(z,c)	$\frac{1}{c!}$

**nCr(Uttr1, 0) ⇒ 1****nCr(Uttr1, negHeltall) ⇒ 0****nCr(Uttr1, posHeltall) ⇒ Uttr1 · (Uttr1-1) · ... · (Uttr1-posHeltall+1) / posHeltall!****nCr(Uttr1, ikke-heltall) ⇒ uttrykk! / ((Uttr1-ikkeHeltall)! · ikkeHeltall!)****nCr(Liste1, Liste2) ⇒ liste**

Returnerer en liste over kombinasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

nCr({5,4,3}, {2,4,2})	{10,1,3}
-----------------------	----------

**nCr(Matrise1, Matrise2) ⇒ matrise**

Returnerer en matrise av kombinasjoner basert på samsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

nCr( $\begin{pmatrix} 6 & 5 \\ 4 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ )	$\begin{pmatrix} 15 & 10 \\ 6 & 3 \end{pmatrix}$
---	--

**nDerivative()**Katalog > **nDerivative(Uttr1, Var=Verdi [,Orden]) ⇒ verdi****nDerivative(Uttr1, 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.

nDerivative( x ,x=1)	1
nDerivative( x ,x) x=0	undef
nDerivative(√(x-1),x) x=1	undef

## nDerivative()

Katalog > 

Den deriverte må være av orden 1 eller 2.

## newList() (nyListe)

Katalog > 

**newList(numElementer)** ⇒ liste

newList(4)	{0,0,0,0}
------------	-----------

Returnerer en liste med en dimensjon lik *numElementer*. Hvert element er null.

## newMat() (nyMat)

Katalog > 

**newMat(numRader, numKolonner)** ⇒ matrise

newMat(2,3)	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
-------------	--

Returnerer en matrise med bare nuller med dimensjonen *numRader* og *numKolonner*.

## nfMax()

Katalog > 

**nfMax(Uttr1, Var)** ⇒ verdi

$\text{nfMax}(x^2 - 2 \cdot x - 1, x)$	-1.
--	-----

**nfMax(Uttr1, Var, nedGrense)** ⇒ verdi

$\text{nfMax}(0.5 \cdot x^3 - x - 2, x, -5, 5)$	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.

**Merk:** Se også **fMax()** og **d()**.

**nfMin()**

Katalog &gt;

**nfMin**(*Uttr1*, *Var*) $\Rightarrow$ verdi**nfMin**(*Uttr1*, *Var*, *nedGrense*) $\Rightarrow$ verdi**nfMin**(*Uttr1*, *Var*, *nedGrense*,  
*øvGrense*) $\Rightarrow$ verdi**nfMin**(*Uttr1*, *Var*) | *nedGrense* $\leq$ *Var*  
 $\leq$ *øvGrense* $\Rightarrow$ 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.

**Merk:** Se også **fMin()** og **d()**.

$\text{nfMin}(x^2+2\cdot x+5,x)$	-1.
$\text{nfMin}(0.5\cdot x^3-x-2,x,-5,5)$	-5.

**nInt()**

Katalog &gt;

**nInt**(*Uttr1*, *Var*, *Nedre*, *Øvre*) $\Rightarrow$ uttrykk

Hvis integranden *Uttr1* ikke inneholder andre verdier enn *Var*, og hvis *Nedre* og *Øvre* er konstanter, positiv  $\infty$ , eller negativ  $\infty$ , så returnerer **nInt()**en tilnærmet av  $\int$ (*Uttr1*, *Var*, *Nedre*, *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.

**Merk:** Se også  $\int$ (**!**), side 221.

$\text{nInt}(e^{-x^2},x,-1,1)$	1.49365
--------------------------------	---------

$\text{nInt}(\cos(x),x,\pi,\pi+1.E-12)$	-1.04144E-12
$\int_{-\pi}^{\pi+10^{-12}} \cos(x) dx$	$-\sin\left(\frac{1}{1000000000000}\right)$

$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x\cdot y}}{\sqrt{x^2-y^2}},y,-x,x\right),x,0,1\right)$	3.30423
---	---------

**nom**(*effektivRente*, *CpY*) ⇒ 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 62.

**nor** (verken ... eller)  -taster

*BoolskUttr1* **nor** *BoolskUttr2* returnerer *Boolsk uttrykk*

 $x \geq 3$  or  $x \geq 4$  $x \geq 3$  $x \geq 3$  nor  $x \geq 4$  $x < 3$ 

*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* ⇒ *heltall*

3 or 4

7

3 nor 4

-8

 $\{1,2,3\}$  or  $\{3,2,1\}$  $\{3,2,3\}$  $\{1,2,3\}$  nor  $\{3,2,1\}$  $\{-4,-3,-4\}$ 

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).

**norm()**katalog > **norm(Matrise)**⇒uttrykk

$$\text{norm}\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \sqrt{a^2+b^2+c^2+d^2}$$

**norm(Vektor)**⇒uttrykk

$$\text{norm}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \sqrt{30}$$

Returnerer Frobenius-normen.

$$\text{norm}\begin{pmatrix} 1 & 2 \end{pmatrix} \quad \sqrt{5}$$

$$\text{norm}\begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \sqrt{5}$$

**normalLine()**katalog > **normalLine(Uttr1,Var,Punkt)**⇒uttrykk

$$\text{normalLine}(x^2,x,1) \quad \frac{3}{2} \frac{x}{2}$$

**normalLine(Uttr1,Var=Punkt)**⇒uttrykk

$$\text{normalLine}((x-3)^2-4,x,3) \quad x=3$$

Returnerer normallinjen til kurven som er representert av *Uttr1* i punktet angitt av *Var=Punkt*.

$$\text{normalLine}\left(\frac{1}{x^3},x=0\right) \quad 0$$

Sørg for at den uavhengige variabelen ikke er definert. Hvis for eksempel  $f_1(x):=5$  og  $x:=3$ , vil **normalLine(f1(x),x,2)** returnere "false" ("usann").

$$\text{normalLine}(\sqrt{|x|},x=0) \quad \text{undef}$$

**normCdf()**Katalog > **normCdf(nedreGrense,øvreGrense[,μ [,σ]])**⇒tall hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er listerBeregner sannsynligheten i normalfordelingen mellom *nedreGrense* og *øvreGrense* for den angitte  $\mu$  (standard=0) og  $\sigma$  (standard=1).For  $P(X \leq \text{øvreGrense})$ , sett *nedreGrense* =  $-\infty$ .**normPdf()**Katalog > **normPdf(XVerdi[ [,μ [,σ]])**⇒tall hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en listeBeregner sannsynlighetstetthet for normalfordelingen ved en spesifisert *XVerdi*-verdi for spesifisert  $\mu$  og  $\sigma$ .

**not** *BoolsUttr1* ⇒ *Boolsk uttrykk*

Returnerer sann, usann eller en forenklet form av argumentet.

**not** *Heltall1* ⇒ *heltall*

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.

Du kan skrive inn heltallet med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten prefiks blir heltallet behandlet som et desimalt tall (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 18.

<code>not(2≥3)</code>	true
<code>not(x&lt;2)</code>	$x \geq 2$
<code>not not innocent</code>	<i>innocent</i>

I heksades grunntall-modus:

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

<code>not 0h7AC36</code>	<code>0hFFFFFFFF853C9</code>
--------------------------	------------------------------

I binær grunntall-modus:

<code>0b100101 ▶ Base10</code>	37
<code>not 0b100101</code>	
<code>0b11111111111111111111111111111111 ▶</code>	
<code>not 0b100101 ▶ Base10</code>	-38

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

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

## nPr() (antPerm)

**nPr**(*Uttr1*, *Uttr2*) ⇒ *uttrykk*

For heltall *Uttr1* og *Uttr2* der  $Uttr1 \geq Uttr2 \geq 0$ , er **nPr**() et antall permutasjoner av *Uttr1* som tar *Uttr2* om gangen. Begge argumentene kan være heltall eller symbolske uttrykk.

**nPr**(*Uttr1*, 0) ⇒ 1

**nPr**(*Uttr1*, *negHeltall*) ⇒  $1 / ((Uttr1 + 1) \cdot (Uttr1 + 2) \cdot \dots \cdot (Uttr1 - negHeltall))$

**nPr**(*Uttr1*, *posHeltall*) ⇒  $Uttr1 \cdot (Uttr1 - 1) \cdot \dots$

$(Uttr1 \text{ posHeltall} + 1)$

<code>nPr(z,3)</code>	$z \cdot (z-2) \cdot (z-1)$
<code>Ans z=5</code>	60
<code>nPr(z,-3)</code>	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
<code>nPr(z,c)</code>	$\frac{z!}{(z-c)!}$
<code>Ans·nPr(z-c,-c)</code>	1



**nPr**(*Uttr1*, *ikkeHeltall*) $\Rightarrow$ *Uttr1!* /  
(*Uttr1-ikkeHeltall*)!

**nPr**(*Liste1*, *Liste2*) $\Rightarrow$ *liste*

nPr({5,4,3},{2,4,2})	{20,24,6}
----------------------	-----------

Returnerer en liste over permutasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

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

nPr( $\begin{pmatrix} 6 & 5 \\ 4 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ )	$\begin{pmatrix} 30 & 20 \\ 12 & 6 \end{pmatrix}$
---	---

Returnerer en matrise av permutasjoner basert på tilsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

## npv()

**npv**(*Rentefot*, *CFO*, *CFListe*, *CFFrekv*)

<i>list1</i> := {6000, -8000, 2000, -3000}	{6000, -8000, 2000, -3000}
--	----------------------------

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.

<i>list2</i> := {2,2,2,1}	{2,2,2,1}
---------------------------	-----------

npv(10,5000, <i>list1</i> , <i>list2</i> )	4769.91
--	---------

*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.

**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*]) | *nedGrense* ≤ *Var* ≤ *øvgrense* ⇒ *tall* 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()** er ofte mye raskere enn **solve()** eller **zeros()**, særlig hvis du bruker “|” operatoren for å begrense søket til et relativt lite intervall som inneholder en eksakt eller enkel løsning.

**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.”

**Merk:** Se også **cSolve()**, **cZeros()**, **solve()** og **zeros()**.

## O

**OneVar** [*1*], *X*[, [*Frek*ν]],[*Kategori*, *Inkludert*]]

$\text{nSolve}(x^2+5\cdot x-25=9, x)$	3.84429
$\text{nSolve}(x^2=4, x=1)$	-2.
$\text{nSolve}(x^2=4, x=1)$	2.

**Merk:** Hvis det foreligger flere løsninger, kan du bruke et forslag som hjelp for å finne en spesiell løsning.

$\text{nSolve}(x^2+5\cdot x-25=9, x)   x < 0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26, r\right)   r > 0 \text{ and } r < 0.25$	0.006886
$\text{nSolve}(x^2=-1, x)$	"No solution found"

**OneVar**[ $n$ , $X1$ , $X2$ [ $X3$ [...[ $X20$ ]]]]

Beregner en-variabel-statistikker med opptil 20 lister. En oversikt over resultatene lagres i *stat.results*-variabelen (side 186).

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 263.

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

Utdata-variabel	Beskrivelse
stat.Q 1X	Første kvartil av x
stat.MedianX	Median av x
stat.Q 3X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.SSX	Sum av kvadratavvik fra gjennomsnittet av x

## or (eller)

Katalog > 

*BoolskUttr1* **or** *BoolskUttr2* returnerer  
*Boolsk uttrykk*

$x \geq 3$  or  $x \geq 4$

$x \geq 3$

*BoolskListe1* **or** *BoolskListe2* returnerer  
*Boolsk liste*

Define  $g(x)$  = Func

Done

If  $x \leq 0$  or  $x \geq 5$

Goto end

Return  $x \cdot 3$

Lbl end

EndFunc

*BoolskMatrise1* **or** *BoolskMatrise2*  
returnerer *Boolsk matrise*

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

$g(3)$

9

Returnerer sann hvis ett eller begge  
uttrykkene er sanne. Returnerer usann  
kun hvis begge uttrykkene behandles  
som usanne.

$g(0)$

A function did not return a value

**Merk:** Se **xor**.

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

*Heltall1* **or** *Heltall2* **pheltall**

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.

I heksades grunntall-modus:

0h7AC36 or 0h3D5F

0h7BD7F

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

I binær grunntall-modus:

0b100101 or 0b100

0b100101

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 18.

**Merk:** Se **xor**.

**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)

**ord**(*Streng*) $\Rightarrow$ *heltall*

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

Returnerer numerisk kode for de første tegnene i tegnstreng *Streng*, eller en liste over de første tegnene i hvert listeelement.

<code>ord("hello")</code>	104
<code>char(104)</code>	"h"
<code>ord(char(24))</code>	24
<code>ord({"alpha", "beta"})</code>	{97,98}

**P****P►Rx()**

**P►Rx**(*rUtr*,  $\theta$ *Utr*) $\Rightarrow$ *uttrykk*

**P►Rx**(*rListe*,  $\theta$ *Liste*) $\Rightarrow$ *liste*

**P►Rx**(*rMatrise*,  $\theta$ *Matrise*) $\Rightarrow$ *matrise*

Returnerer ekvivalent x-koordinat av (*r*,  $\theta$ ) par.

**Merk:**  $\theta$ -argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus. Hvis argumentet er et uttrykk, kan du bruke °,  $\text{G}$  eller  $\text{r}$  for å hoppe over vinkelmodus-innstillingen midlertidig.

I Radian-vinkelmodus:

<code>P►Rx(<i>r</i>, <math>\theta</math>)</code>	$\cos(\theta) \cdot r$
<code>P►Rx(4,60°)</code>	2
<code>P►Rx(<math>\{-3,10,1.3\}, \{\frac{\pi}{3}, \frac{\pi}{4}, 0\})</math></code>	$\{-\frac{3}{2}, 5\sqrt{2}, 1.3\}$

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

**P>Ry**(*rUtr*, *θUtr*) ⇒ *Utr*

**P>Ry**(*rListe*, *θListe*) ⇒ *liste*

**P>Ry**(*rMatrise*, *θMatrise*) ⇒ *matrise*

Returnerer ekvivalent y-koordinat av (r, θ)-paret.

**Merk:** θ-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@>Ry (...)**.

I Radian-vinkelmodus:

$P>Ry(r, \theta)$	$\sin(\theta) \cdot r$
$P>Ry(4, 60^\circ)$	$2 \cdot \sqrt{3}$
$P>Ry\left(\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right)$	$\left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\}$

### PassErr

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

**Merk:** Se også **ClrErr (SlettFeil)**, side 26, og **Try**, side 201.

For et eksempel på **PassErr**, se eksempel 2 under **Try**-kommandoen, side 201.

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

### piecewise() (stykkevis)

**piecewise**(*Uttr1* [, *Betingelse1* [, *Uttr2* [, *Betingelse2* [, ... ]]])

Returnerer definisjoner for en stykkevis definert funksjon i form av en liste. Du kan også opprette stykkevis definisjoner med en sjablon.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
$p(1)$	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  $P(0 \leq X \leq \textit{øvreGrense}) \Rightarrow$  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  $P(X \leq \textit{øvreGrense})$ , sett *nedreGrense*=0

### poissPdf()

**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

**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 154.

*kompleksVerdi* ► Polar

Viser *kompleksVektor* i polar form.

- Grader-vinkelmodus returnerer  $(r \angle \theta)$ .
- Radian-vinkelmodus returnerer  $re^{i\theta}$ .

*kompleksVerdi* kan ha en 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  $(r \angle \theta)$ .

$$\begin{array}{l} [1 \ 3.] \text{►Polar} \quad [3.16228 \ \angle \ 1.24905] \\ [x \ y] \text{►Polar} \\ \left[ \sqrt{x^2+y^2} \ \angle \ \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \right] \end{array}$$

I Radian-vinkelmodus:

$$\begin{array}{l} (3+4 \cdot i) \text{►Polar} \quad e^{i \cdot \left( \frac{\pi}{2} - \tan^{-1}\left(\frac{3}{4}\right) \right)} \cdot 5 \\ \left( 4 \angle \frac{\pi}{3} \right) \text{►Polar} \quad e^{i \cdot \frac{\pi}{3}} \cdot 4 \end{array}$$

I Gradian-vinkelmodus:

$$(4 \cdot i) \text{►Polar} \quad (4 \angle 100.)$$

I Grader-vinkelmodus:

$$(3+4 \cdot i) \text{►Polar} \quad \left( 5 \angle 90 - \tan^{-1}\left(\frac{3}{4}\right) \right)$$

**polyCoeffs()** (**polyKoeffs**)**polyCoeffs**(*Poly* [, *Var*]) ⇒ *liste*

Returnerer en liste over koeffisienter av polynom *Poly* med hensyn på variabel *Var*.

*Poly* må være et polynomisk uttrykk i *Var*. Vi anbefaler at du ikke utelater *Var* med mindre *Poly* er et uttrykk i en enkel variabel.

$$\text{polyCoeffs}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad \{4, -3, 2\}$$

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Utvider polynomet og velger *x* for den utelatte *Var*.



$\text{polyCoeffs}\left((x+y+z)^2, x\right)$	$\{1, 2 \cdot (y+z), (y+z)^2\}$
$\text{polyCoeffs}\left((x+y+z)^2, y\right)$	$\{1, 2 \cdot (x+z), (x+z)^2\}$
$\text{polyCoeffs}\left((x+y+z)^2, z\right)$	$\{1, 2 \cdot (x+y), (x+y)^2\}$

**polyDegree() (polyGrader)**

**polyDegree**(Poly [, Var])  $\Rightarrow$  verdi

Returnerer grader av polynomisk uttrykk *Poly* med hensyn på variabel *Var*. Hvis du utelukker *Var*, velger **polyDegree()** funksjonen en grunninnstilling fra de variablene som ligger i polynom *Poly*.

*Poly* må være et polynomisk uttrykk i *Var*. Vi anbefaler at du ikke utelater *Var* med mindre *Poly* er et uttrykk i en enkel variabel.

$\text{polyDegree}(5)$	0
$\text{polyDegree}(\ln(2) + \pi, x)$	0
Konstante polynomer	
$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$	2
$\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$	5
$\text{polyDegree}\left((x+y^2+z^3)^2, x\right)$	2
$\text{polyDegree}\left((x+y^2+z^3)^2, y\right)$	4
$\text{polyDegree}\left((x-1)^{10000}, x\right)$	10000

Graden kan trekkes ut, selv om koeffisientene ikke kan det. Dette er fordi graden kan trekkes ut uten å utvide polynomet.

**polyEval()**

**polyEval**(Liste1, Uttr1)  $\Rightarrow$  uttrykk

**polyEval**(Liste1, Liste2)  $\Rightarrow$  uttrykk

$\text{polyEval}\{a, b, c, x\}$	$a \cdot x^2 + b \cdot x + c$
$\text{polyEval}\{1, 2, 3, 4, 2\}$	26
$\text{polyEval}\{1, 2, 3, 4, \{2, -7\}\}$	$\{26, -262\}$

**polyEval()**

Katalog &gt;

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.

**polyGcd()**

Katalog &gt;

**polyGcd**(*Uttr1*,*Uttr2*) $\Rightarrow$ uttrykk

Returnerer største felles divisor for de to argumentene.

*Uttr1* og *Uttr2* må være polynomiske uttrykk.

Liste, matrise og boolske argumenter er ikke tillatt.

$\text{polyGcd}(100,30)$	10
$\text{polyGcd}(x^2-1,x-1)$	$x-1$
$\text{polyGcd}(x^3-6x^2+11x-6,x^2-6x+8)$	$x-2$

**polyQuotient() (polyKvotient)**

Katalog &gt;

**polyQuotient**(*Poly1*,*Poly2*  
[,*Var*]) $\Rightarrow$ uttrykk

Returnerer kvotienten av polynom *Poly1* dividert med polynom *Poly2* med hensyn på den spesifiserte variabelen *Var*.

*Poly1* og *Poly2* må være polynomiske uttrykk i *Var*. Vi anbefaler at du ikke utelater *Var* med mindre *Poly1* og *Poly2* er uttrykk i den samme enkeltvariabelen.

$\text{polyQuotient}(x-1,x-3)$	1
$\text{polyQuotient}(x-1,x^2-1)$	0
$\text{polyQuotient}(x^2-1,x-1)$	$x+1$
$\text{polyQuotient}(x^3-6x^2+11x-6,x^2-6x+8)$	$x$
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,x)$	$y-z$
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,y)$	$2\cdot x-y+2\cdot z$
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,z)$	$-(x-y)$

**polyRemainder() (polyRest)**

Katalog &gt;

**polyRemainder**(*Poly1*,*Poly2*  
[,*Var*]) $\Rightarrow$ uttrykk

$\text{polyRemainder}(x-1,x-3)$	2
$\text{polyRemainder}(x-1,x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1,x-1)$	0

## polyRemainder() (polyRest)

Katalog > 

Returnerer rest av polynom *Poly1* dividert med polynom *Poly2* med hensyn på den spesifiserte variabelen *Var*.

*Poly1* og *Poly2* må være polynomiske uttrykk i *Var*. Vi anbefaler at du ikke utelater *Var* med mindre *Poly1* og *Poly2* er uttrykk i den samme enkeltvariabelen.

$$\frac{\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)}{(y-z) \cdot (2 \cdot y+z)}$$

$$\frac{\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)}{-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2}$$

$$\frac{\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)}{(x-y) \cdot (x+2 \cdot y)}$$

## polyRoots()

Katalog > 

**polyRoots(*Poly*, *Var*)** ⇒ *liste*

**polyRoots(*KoeffListe*)** ⇒ *liste*

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: {}.

*Poly* må være et polynom i én variabel.

Den andre syntaksen, **polyRoots(*KoeffListe*)**, returnerer en liste over sanne røtter for koeffisientene i *KoeffListe*.

**Merk:** Se også **cPolyRoots()**, side 38.

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

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

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

$$\frac{\text{polyRoots}(\{1, 2, 1\})}{\{-1, -1\}}$$

## PowerReg (PotensReg)

katalog > 

**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 186).

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 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 263).

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 <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>

## Prgm

**Prgm**  
*Blokk*

Beregne GCD og vis mellomresultater.

**EndPrgm**

Sjablon for å opprette et egendefinert program. Må brukes med kommandoen **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 produkthåndboken.

```
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 235.**Produkt (PI)**Se  $\Pi()$ , side 235.**Product()**Katalog > 

**product**(Liste[, Start[, slutt]]) $\Rightarrow$ uttrykk

Returnerer produktet av elementene i Liste. Start og Slutt er valgfrie. De spesifiserer et elementområde.

**product**(MatriseI[, Start[, slutt]]) $\Rightarrow$ matrise

Returnerer en radvektor som inneholder produktet av elementene i kolonnene i MatriseI. Start og slutt er alternativer. De spesifiserer et radområde.

product({{1,2,3,4}})	24
----------------------	----

product({{2,x,y}})	$2 \cdot x \cdot y$
--------------------	---------------------

product({{4,5,8,9}},2,3)	40
--------------------------	----

product $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$[28 \ 80 \ 162]$
---	-------------------

product $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 1, 2$	$[4 \ 10 \ 18]$
---	-----------------

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 263.

## propFrac() (ekteBrøk)

**propFrac**(*Utt1*[,*Var*]) $\Rightarrow$ uttrykk

**propFrac**(*rasjonal\_tall*) returnerer *rasjonal\_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.

For rasjonale uttrykk er **propFrac()** et raskere, men mindre ekstremt alternativ til **expand()**.

$$\text{propFrac}\left(\frac{4}{3}\right) \quad 1 + \frac{1}{3}$$

$$\text{propFrac}\left(\frac{-4}{3}\right) \quad -1 - \frac{1}{3}$$

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

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

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

## Q



## QR

**QR** *Matrise*, *qMatNavn*, *rMatNavn*(*ToI*)

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*.

- 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:  
 $5E-14 \cdot \max(\dim(\text{Matrise})) \cdot \text{radNorm}(\text{Matrise})$

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*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

QR m1,qm,rm Done

qm	0.123091	0.904534	0.408248
	0.492366	0.301511	-0.816497
	0.86164	-0.301511	0.408248

rm	8.12404	9.60114	11.0782
	0.	0.904534	1.80907
	0.	0.	0.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR m1,qm,rm Done

qm	$\frac{m}{\sqrt{m^2+o^2}}$	$\frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}}$
	$\frac{o}{\sqrt{m^2+o^2}}$	$\frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}}$

rm	$\sqrt{m^2+o^2}$	$\frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}}$
	0	$\frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}}$

## 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 186.)

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 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 263).

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 186.)

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 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 263).

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 ▶ Pθ()

Katalog &gt;

R ▶ Pθ (*xExpr*, *yExpr*) ⇒ *uttrykk*

I Grader-vinkelmodus:

$$R \blacktriangleright P\theta(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

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 (...)**.

I Gradian-vinkelmodus:

$$R \blacktriangleright P\theta(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

I Radian-vinkelmodus:

$$R \blacktriangleright P\theta(3,2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$R \blacktriangleright P\theta\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) \quad \left[0 \ \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \ 0.643501\right]$$

## R ▶ Pr()

Katalog &gt;

R ▶ Pr (*xExpr*, *yExpr*) ⇒ *uttrykk*

I Radian-vinkelmodus:

$$R \blacktriangleright Pr(3,2) \quad \sqrt{13}$$

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 (...)**.

$$R \blacktriangleright Pr(x,y) \quad \sqrt{x^2+y^2}$$

$$R \blacktriangleright Pr\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) \quad \left[3 \ \frac{\sqrt{\pi^2+256}}{4} \ 2.5\right]$$

## ▶ Rad

Katalog &gt;

*Expr1* ▶ Rad ⇒ *uttrykk*

I Grader-vinkelmodus:

Omformer argumentet til radian vinkelmåling.

$$(1.5) \blacktriangleright \text{Rad} \quad (0.02618)^r$$

I Gradian-vinkelmodus:

**► Rad**Katalog > 

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

(1.5)►Rad	(0.023562) <sup>r</sup>
-----------	-------------------------

**tilf()**Katalog > 

**rand()** ⇒ *uttrykk*  
**rand(#Trials)** ⇒ *liste*

Angir det tilfeldige tallet for start.

**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(*n*, *p*)** returnerer et tilfeldig reelt tall fra en spesifisert binomisk fordeling.

randBin(80,0.5)	42
randBin(80,0.5,3)	{41,32,39}

**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)	5
randInt(3,10,4)	{9,7,5,8}

**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()**

**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)	<table border="1"> <tr><td>8</td><td>-3</td><td>6</td></tr> <tr><td>-2</td><td>3</td><td>-6</td></tr> <tr><td>0</td><td>4</td><td>-6</td></tr> </table>	8	-3	6	-2	3	-6	0	4	-6
8	-3	6								
-2	3	-6								
0	4	-6								

**Merk:** Verdene i denne matrisen endres hver  
 gang du trykker på .

**randNorm()**

**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  $[\mu-3\cdot\sigma, \mu+3\cdot\sigma]$ .

**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( <i>x</i> ,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 <i>list3</i> ={1,2,3,4,5}	Done
Define <i>list4</i> =randSamp( <i>list3</i> ,6)	Done
<i>list4</i>	{2,3,4,3,1,2}

## 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**(*Expr1*) ⇒ *uttrykk*

Returnerer den reelle delen av argumentet.

**Merk:** Alle ubestemte variabler behandles som reelle variabler. Se også **imag()**, page 94.

**real**(*List1*) ⇒ *liste*

Returnerer den reelle delen av alle elementer.

real( $2+3 \cdot i$ )	2
real( <i>z</i> )	<i>z</i>
real( $x+i \cdot y$ )	<i>x</i>

real({ <i>a+i·b</i> ,3, <i>i</i> })	{ <i>a</i> ,3,0}
-------------------------------------	------------------

$\text{real}(\text{MatrixI}) \Rightarrow \text{matrise}$

Returnerer den reelle delen av alle elementer.

$$\text{real}\left(\begin{pmatrix} a+i \cdot b & 3 \\ c & i \end{pmatrix}\right) \quad \begin{matrix} a & 3 \\ c & 0 \end{matrix}$$

## ► Rect

*Vector* ► **Rect**

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

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 140.

*complexValue* ► **Rect**

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 ( $r\angle \theta$ ).

$$\left(\begin{pmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{pmatrix}\right) \text{►Rect} \quad \begin{matrix} 3 \cdot \sqrt{2} & 3 \cdot \sqrt{2} & 3 \cdot \sqrt{3} \\ 4 & 4 & 2 \end{matrix}$$

$$\begin{matrix} [a & \angle b & \angle c] \\ [a \cdot \cos(b) \cdot \sin(c) & a \cdot \sin(b) \cdot \sin(c) & a \cdot \cos(c)] \end{matrix}$$

I Radian-vinkelmodus:

$$\left(\begin{pmatrix} 4 \cdot e^{\frac{\pi}{3}} \end{pmatrix}\right) \text{►Rect} \quad \begin{matrix} \frac{\pi}{4} \\ 4 \cdot e^{\frac{\pi}{3}} \end{matrix}$$

$$\left(\left(4 \angle \frac{\pi}{3}\right)\right) \text{►Rect} \quad 2+2 \cdot \sqrt{3} \cdot i$$

I Gradian-vinkelmodus:

$$\left(\left(1 \angle 100\right)\right) \text{►Rect} \quad i$$

I Grader-vinkelmodus:



$$\left(\left(4 \angle 60\right)\right) \text{►Rect} \quad 2+2 \cdot \sqrt{3} \cdot i$$

**Merk:** For å skrive  $\angle$ , velg den fra symbollisten i Katalogen.

$\text{ref}(\text{Matrix1}, \text{Tol}) \Rightarrow \text{matrise}$

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.

Hvis for eksempel *a* er udefinert i følgende uttrykk, vises en varselmelding, og resultatet vises som:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ a & & \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Varslet 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.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow m1$$

$$\text{ref}(m1) \quad \begin{bmatrix} a & b \\ c & d \\ 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$$

$$\text{ref} \left( \begin{array}{ccc|c} a & 1 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \end{array} \right) | a=0 \quad \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}$$

**Merk:** Se også `rref()`, page 165.


## RefreshProbeVars

## RefreshProbeVars

Gir deg tilgang til sensordata fra alle tilkoblede sensorer gjennom TI-Basic-programmet.

StatusVar-  
verdi

## Status

- statusVar* =0 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.
- 
- statusVar* =2 Vernier DataQuest™-applikasjonen er ikke startet.
- statusVar* =3 Vernier DataQuest™-applikasjonen er startet, men du har ikke koblet til noen sensorer.

## 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
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()**Katalog > 

**remain**(*Expr1*, *Expr2*) ⇒ *uttrykk*

**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 122.

$\text{remain}(7,0)$	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}

$\text{remain}\left(\begin{pmatrix} 9 & -7 \\ 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 4 & -3 \end{pmatrix}\right)$	$\begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}$
--	---

**Forespør**Katalog > 

**Forespør** *promptString*, *var*[, *DispFlag* [, *statusVar*]]

**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.

Definere et program:

```
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**:

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*

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

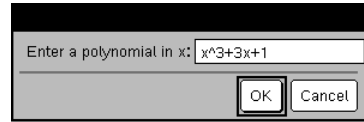
Stråle: 6/2  
Area=- 28,2743

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:  $\{-0,322185\}$

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 159.



## 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:

- **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 **Forespør**, page 157.

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()
```



Resultat etter klikk på **OK** (Merk at argumentet *DispFlag* på **0** utelater prompten og svaret fra loggen):

```
forespørStr_demo()
```

Svaret har 5 tegn.

**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**(*Comparison*) ⇒ *uttrykk*

Returnerer høyre side av en ligning eller ulikhet.

**right**{ { 1,3,-2,4 },3}

{ 3,-2,4 }

**right**("Hello",2)

"lo"

**right**( $x < 3$ )

3

**rk23 ()**

**rk23**(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *difto1*]) ⇒ *matrise*

**rk23**(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *difto1*]) ⇒ *matrise*

Differensialligning:

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

**rk23**{  $0,001 \cdot y \cdot (100 - y)$ ,  $t, y$ , { 0,100 }, 10, 1 }

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

**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 [*Var0*, *VarMax*]. Returnerer en matrise, hvis første rad definerer resultatverdiene av *Var* som definert av *VarStep*. Den andre raden definerer verdien av den første løsningskomponenten for de tilsvarende *Var*-verdiene, og så videre.

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

*SystemOfExpr* er et system på høyre side som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i *ListOfDepVars*).

*ListOfExpr* er en liste over høyresider som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i *ListOfDepVars*).

*Var* er den uavhengige variabelen.

*ListOfDepVars* er en liste over avhengige variabler.

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

*ListOfDepVars0* er en liste over opprinnelige verdier for avhengige variabler.

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

Samme ligning med *diftol* satt til  $1.E-6$

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Sammenlign resultatet ovenfor med eksakt løsning i CAS som ble funnet ved hjelp av `deLøs()` og `sekvGen()`:

$$\text{deSolve}\left(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\right)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9.}$$

$$\text{sekvGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9.}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.4\}$$

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

## rk23 ()

Katalog > 

Hvis  $VarStep$  beregnes til et tall som ikke er null:  $tegn(VarStep) = tegn(VarMax - Var0)$  og løsninger returneres ved  $Var0 + i * VarStep$  for alle  $i=0,1,2,\dots$  slik at  $Var0 + i * VarStep$  er i  $[var0, VarMax]$  (får kanskje ikke en løsningsverdi ved  $VarMax$ ).

Hvis  $VarStep$  beregnes til null, returneres løsningene ved "Runge-Kutta"  $Var$ -verdiene.

$diftol$  er feiltoleransen (grunnverdi på 0,001).

## root()

Katalog > 

$root(Expr) \Rightarrow rot$

$root(Expr1, Expr2) \Rightarrow rot$

$root(Expr)$  returnerer kvadratrotten av  $Expr$ .

$root(Expr1, Expr2)$  returnerer  $Expr2$ -rotten av  $Expr1$ .  $Expr1$  kan være en reell eller kompleks flytende desimalpunkt-konstant, et heltall eller en kompleks rasjonal konstant eller et generelt symbolsk uttrykk.

**Merk:** Se også **mal for N.-rot**, side 1.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	$\frac{1}{3^3}$
$\sqrt[3]{3.}$	1.44225

## rotate()

Katalog > 

$rotate(Integer1[, #ofRotations]) \Rightarrow heltall$

I binær grunntall-modus:

Roterer bitene i et binært heltall. Du kan angi  $Integer1$  i enhver basis. Den konverteres automatisk til en 64-biters binærfom 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 18.

$rotate(0b11111111111111111111111111111111)$	
$0b1000000000000000000000000000000001*$	
$rotate(256,1)$	$0b1000000000$

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

## rotate()

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.

0b00000000000001111010110000110101

Bit helt til høyre roterer mot venstre.

produserer:

0b1000000000000111101011000011010

Resultatene vises i forhold til grunntall-modusen.

**rotate**(*ListI*[,*#ofRotations*])  $\Rightarrow$  *liste*

Returnerer en kopi av *ListI* som er rotert mot høyre eller mot venstre av *#ofRotations*-elementer. Endrer ikke *ListI*.

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**(*StringI*[,*#ofRotations*])  $\Rightarrow$  *streng*

Returnerer en kopi av *StringI* som er rotert mot høyre eller mot venstre av *#ofRotations*-tegnene. Endrer ikke *StringI*.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er  $-1$  (roterer ett tegn mot høyre).

I heksadesimal grunntall-modus:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h80000000000001E3
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"

## round()

**round**(*ExprI*[,*digits*])  $\Rightarrow$  *uttrykk*

Returnerer argumentet avrundet til spesifisert antall sifre etter desimalpunktet.

round(1.234567,3)	1.235
-------------------	-------

## round()

Katalog > 

*digits* må være et heltall i området 1 – 12. Hvis *digits* ikke er inkludert, returneres argumentet avrundet til 12 signifikante siffer.

**Merk:** Visning av siffermodus kan ha innvirkning på hvordan dette vises.

**round(List1[, digits])** ⇒ *liste*

Returnerer en liste over elementer avrundet til spesifisert antall sifre.

$$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4)$$

---

$$\{3.1416, 1.4142, 0.6931\}$$

**round(Matrix1[, digits])** ⇒ *matrise*

Returnerer en matrise over elementer avrundet til spesifisert antall sifre.

$$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1\right)$$

---

$$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

## rowAdd()

Katalog > 

**rowAdd(Matrix1, rIndex1, rIndex2)** ⇒ *matrise*

Returnerer en kopi av *Matrix1* med rad *rIndex2* erstattet med summen av rader *rIndex1* og *rIndex2*.

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

---

$$\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$$
  
$$\text{rowAdd}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$$

---

$$\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$$

## rowDim()

Katalog > 

**rowDim(Matrix)** ⇒ *uttrykk*

Returnerer antallet rader i *Matrix*.

**Merk:** Se også **colDim()**, side 27.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$$

---

$$\text{rowDim}(m1)$$

---

$$3$$

## 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 27.

$$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right)$$

---

$$25$$



## rowSwap()

Katalog > 

**rowSwap**(*Matrix1*, *rIndex1*, *rIndex2*)  
⇒ *matrise*

Returnerer *Matrix1* med rader *rIndex1* og *rIndex2* ombyttet.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
rowSwap(mat,1,3)	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

## rref()


Katalog > 



**rref**(*Matrix1* [, *Tol*]) ⇒ *matrise*

Returnerer den reduserte eliminasjonsformen av *Matrix1*.

$rref\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
---	---

Alternativt 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*.

 $rref\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
---	--

- 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 155.

## S

### sec()

 **tast**

**sec**(*Uttrykk*) ⇒ *Uttrykk*

I Grader-vinkelmodus:

**sec**(*Liste1*) ⇒ *liste*

## sec()

 **tast**

Returnerer sekans til *uttrykk1*, eller returnerer en liste med secant til hvert element i *liste1*.

**Merk:** Argumentet tolkes som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °, G eller  $\frac{\pi}{180}$  for å hoppe over vinkelmodusen midlertidig.

$$\frac{\sec(45)}{\sec(\{1,2,3,4\})} = \frac{\sqrt{2}}{\left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\}}$$

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

 **tast**

sec<sup>-1</sup>(*Uttr1*) ⇒ *Uttrykk*

sec<sup>-1</sup>(*Liste1*) ⇒ *liste*

Returnerer vinkelen som har sekans lik *uttrykk1*, eller returnerer en liste med invers sekans til hvert element i *liste1*.

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

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

I Grader-vinkelmodus:

$$\sec^{-1}(1) = 0$$

I Gradian-vinkelmodus:

$$\sec^{-1}(\sqrt{2}) = 50$$

I Radian-vinkelmodus:

$$\sec^{-1}(\{1,2,5\}) = \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\}$$

## sech()

**Katalog** > 

sech(*Uttr1*) ⇒ *Uttrykk*

sech(*Liste1*) ⇒ *liste*

Returnerer hyperbolsk secant av *uttrykk1*, eller returnerer en liste med hyperbolsk sekans av hvert element i *liste1*.

$$\frac{\operatorname{sech}(3)}{\operatorname{cosh}(3)} = \frac{1}{\cosh(3)}$$

$$\frac{\operatorname{sech}(\{1,2,3,4\})}{\left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\}}$$

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

**Katalog** > 

sech<sup>-1</sup>(*Uttr1*) ⇒ *Uttrykk*

sech<sup>-1</sup>(*Liste1*) ⇒ *liste*

I Radian-vinkelmodus og Rectangular-kompleksmodus:

Returnerer invers hyperbolsk sekans til *uttrykk1*, eller returnerer en liste med invers hyperbolsk sekans til hvert element i *Liste1*.

sech <sup>-1</sup> (1)	0
sech <sup>-1</sup> ({1, -2, 2.1})	$\left\{ 0, \frac{2 \cdot \pi}{3} \cdot i, 8 \cdot \epsilon^{-15} + 1.07448 \cdot i \right\}$

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

## Send

## Hub-meny

Send *utrEllerStreng1*,  
*utrEllerStreng2* ...

Programmeringskommando: Sender én eller flere TI-Innovator™ Hub kommandoer til en tilkoblet hub.

*utrEllerStreng* må være en gyldig TI-Innovator™ Hub kommando. Vanligvis inneholder *utrEllerStreng* 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 82), **GetStr** (side 89), og **eval()** (side 66).

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.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(m·n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

## seq() (sekv)

Katalog >

**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.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

**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 >

**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:

$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$	$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$
---	--

**Eksempel der det innledende leddet er symbolsk:**

$\text{seqGen}\left(u(n-1)+2, n, u, \{1, 5\}, \{a\}\right)$	$\{a, a+2, a+4, a+6, a+8\}$
---	-----------------------------

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.

System av to tallfølger:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}, \left[\begin{array}{c} - \\ 2 \end{array}\right]\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{2} & \frac{19}{24} \end{bmatrix}$$

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* [, *Lofverdi*]]]}])⇒*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 *Uttr*(*u*, *n*) og *ListeMedInnlLedd*, og returnerer resultatene som en liste.

**seqn**(*uttr*{*n* [, *nMaks* [, *Lofverdi*]]})⇒*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 *Uttr*(*n*), og returnerer resultatene som en liste.

Hvis *nMaks* mangler, innstilles *nMaks* på 2500

Hvis *nMax*=0, innstilles *nMaks* på 2500

**Merk:** **seqn()** kaller **seqGen()** med  $n0=1$  og  $n\text{intervall}=1$

Genererer de første 6 leddene i tallfølgen  $u(n) = u(n-1)/2$ , med  $u(1)=2$ .

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$


---


$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

**series(Uttr1, Var, Orden [, Punkt])** ⇒ uttrykk

**series(Uttr1, Var, Orden [, Punkt] | Var > Punkt)** ⇒ uttrykk

**series(Uttr1, Var, Orden [, Punkt] | Var < Punkt)** ⇒ uttrykk

Returnerer en generalisert kuttet potensrekkepresentasjon av *Uttr1* som er utvidet rundt *Punkt* av grad *Orden*. *Orden* kan være et vilkårlig, rasjonelt tall. Resulterende potenser av (*Var* – *Punkt*) kan inkludere negative eksponenter og/eller brøk-eksponenter. Koeffisientene foran disse potensene kan inkludere logaritmer av (*Var* – *Punkt*) og andre funksjoner av *Var* som er dominert av alle potensene til (*Var* – *Punkt*) som har samme eksponenttegn (-sign).

*Punkt* grunninnstilles til 0. *Punkt* kan være  $\infty$  eller  $-\infty$ , i så fall er utvidelsen gjennom grader *Orden* i  $1/(Var - Punkt)$ .

**series(...)** returnerer “**series(...)**” hvis den ikke er i stand til å bestemme en slik representasjon, som for vesentlige singularpunkter, f.eks.  $\sin(1/z)$  ved  $z=0$ ,  $e^{-1/z}$  ved  $z=0$  eller  $e^z$  ved  $z = \infty$  eller  $-\infty$ .

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-}}, z, 1\right) \quad z - 1$$

$$\text{series}\left(\left(1 + \frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} + \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1 + e^x\right)^2, x, 2, 1\right) \quad (e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

Dersom rekken eller en av dens deriverte har en "hopp-diskontinuitet" ved *Punkt*, er det sannsynlig at resultatet inneholder deluttrykk av formen  $\text{tegn}(\dots)$  eller  $\text{abs}(\dots)$  for en reell utvidelsesvariabel eller  $(-1)^{\text{nedre}(\dots\text{vinkel}(\dots)\dots)}$  for en sammensatt utvidelsesvariabel, som er en som ender med "\_". Dersom du vil bruke rekken kun for verdier på den ene siden av *Punkt*, så utvider du med det som passer av " $| \text{Var} > \text{Punkt}$ ", " $| \text{Var} < \text{Punkt}$ ", " $| \text{Var} \geq \text{Punkt}$ " eller " $\text{Var} \leq \text{Punkt}$ " for å oppnå et enklere resultat.

**series()** kan gi symbolsk tilnærming til ubestemte integraler og bestemte integraler som det ellers ikke kan oppnås symbolske løsninger for.

**series()** fordeler over 1. argument-liste og matriser.

**series()** er en generalisert versjon av **taylor()**.

Som vist i det siste eksemplet til høyre kan visningsrutinene nedover fra resultatet som er produsert av **series(...)** arrangere leddene på nytt, slik at **dominantTerm** ikke er det som er helt til venstre.

**Merk:** Se også **dominantTerm()**, side 59.

## SetMode() (lesModus)

**SetMode(modusNavnHeltall, innstillingHeltall)  $\Rightarrow$  heltall**

**SetMode(liste)  $\Rightarrow$  heltallsliste**

Kun gyldig innenfor en funksjon eller et program.

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 gjenopprettes etter at programmet utføres.

**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 88.

**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 flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Define <i>prog1()</i> =Prgm	Done
Disp approx( $\pi$ )	
setMode(1,16)	
Disp approx( $\pi$ )	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done



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, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær
Måleenheter	8	1=SI, 2=Eng/USA

## shift() (skift)

Katalog > 

**shift(Heltall[, #avSkift])** ⇒ *heltall*

Forskyver (skifter) bitene i et binært heltall. Du kan legge inn *Heltall* med hvilket som helst grunntall. Det konverteres automatisk til 64-bit binær form med fortegn. Hvis *Heltall* 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 18.

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:

I binær grunntall-modus:

shift(0b1111010110000110101)	
	0b111101011000011010
shift(256,1)	0b1000000000

I heksades grunntall-modus:

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).

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:

0b00000000000000111101011000011010

Resultatene vises i forhold til grunntallmodusen. Ledende nuller vises ikke.

**shift(Listel [, #avSkift])** ⇒ *liste*

Returnerer en kopi av *Listel* skiftet til høyre eller til venstre av *#avSkift*-elementer. Endrer ikke *Listel*.

Hvis *#avSkift* er positiv, er skift til venstre. Hvis *#avSkift* er negativ, er skift til høyre. Grunninnstillingen er -1 (skiftes et element til høyre).

Elementer som introduseres ved begynnelsen eller slutten av *liste* ved skiftet er satt til symbolet "undef".

**shift(Strengl [, #avSkift])** ⇒ *streng*

Returnerer en kopi av *Strengl* skiftet mot høyre eller mot venstre av *#ofShifts*-tegn. Endrer ikke *Strengl*.

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.

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("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

**sign() (fortegn)**

**sign(Uttryk)** ⇒ *Uttrykk*

**sign(Listel)** ⇒ *list e*

**sign(MatriseI)** ⇒ *matrise*

sign(-3.2)	-1.
sign({2,3,4,-5})	{1,1,1,-1}
sign(1+ x )	1

For reell og komplekst Uttrykk1, returneres  $\text{Uttrykk1}/\text{abs}(\text{Uttrykk1})$  når  $\text{Uttrykk1} \neq 0$ .

Returnerer 1 hvis Uttrykk1 er positiv.

Returnerer -1 hvis  $\text{Uttrykk1}$  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.

Hvis kompleks formatmodus er reell:

$$\text{sign}\left(\begin{bmatrix} -3 & 0 & 3 \end{bmatrix}\right) \quad \begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$$

## simult()

**simult(koeffMatrise, konstVektor, Tol)**  $\Rightarrow$  matrise

Returnerer en kolonnevektor som inneholder løsningene til et system av lineære ligninger.

Merk: Se også **linSolve()**, side 108.

*koeffMatrise* må være en kvadratmatrise som inneholder ligningskoeffisientene.

*konstVektor* må ha samme antall rader (samme dimension) som *koeffMatrise* 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 ignorerer *Tol*.

- Hvis du bruker modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen

Løs mhp. x og y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

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

Løsningen er  $x = -3$  og  $y = 2$ .

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\text{simult}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} a & b \\ c & d \\ -2 \cdot b - d \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$$

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 løsningen for det tilsvarende systemet.

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  $y=9/2$ .

*Uttr* sin

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

Representerer *Uttr* ved sinus. Dette er en konverteringsoperator. Den kan bare brukes på slutten av kommandolinjen.

**sin** reduserer alle potenser av  $\cos(\dots)$  modulus  $1 - \sin(\dots)^2$  slik at alle gjenværende potenser av  $\sin(\dots)$  har eksponenter i området  $(0, 2)$ . Dermed vil resultatet være uten  $\cos(\dots)$  hvis og bare hvis  $\cos(\dots)$  inntreffer i det gitte uttrykket bare med partallsekspontener.

**Merk:** Denne konverteringsoperatoren støttes ikke i vinkelmodusen grader eller gradianer. Før du bruker den, må du kontrollere at vinkelmodus er satt til radianer, og at *Uttr* ikke inneholder eksplisitte referanser til vinkler i grader eller gradianer.

$$\frac{(\cos(x))^2 \text{sin}}{1 - (\sin(x))^2}$$

## sin()



$\sin(\text{Utr1}) \Rightarrow \text{Uttrykk}$

$\sin(\text{Liste1}) \Rightarrow \text{liste}$

$\sin(\text{Utr1})$  returnerer sinus til argumentet som et uttrykk.

$\sin(\text{Liste1})$  returnerer en liste over sinus til alle elementer i *Liste1*.

**Merk:** Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke  $^\circ$ , G, eller  $^r$  for å hoppe over vinkelmodusen midlertidig.

I Grader-vinkelmodus:

$$\begin{array}{r} \sin\left(\frac{\pi}{4}\right) \\ \sin(45) \\ \sin(\{0,60,90\}) \end{array} \begin{array}{r} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \left\{0, \frac{\sqrt{3}}{2}, 1\right\} \end{array}$$

I Gradian-vinkelmodus:

$$\sin(50) \quad \frac{\sqrt{2}}{2}$$

I Radian-vinkelmodus:

$$\begin{array}{r} \sin\left(\frac{\pi}{4}\right) \\ \sin(45^\circ) \end{array} \begin{array}{r} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{array}$$

$\sin(\text{kvadratMatrise1}) \Rightarrow \text{kvadratMatrise}$

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

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

I Radian-vinkelmodus:

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

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



$\sin^{-1}(\text{Utr1}) \Rightarrow \text{Uttrykk}$

verdi

$\sin^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

$\sin^{-1}(\text{Utr1})$  returnerer vinkelen med sinus lik *Utr1* som et uttrykk.

I Grader-vinkelmodus:

$$\sin^{-1}(1) \quad 90$$

I Gradian-vinkelmodus:

$$\sin^{-1}(1) \quad 100$$

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



$\sin^{-1}(\text{Liste1})$  returnerer en liste over invers sinus til hvert element i *Liste1*.

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

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

$\sin^{-1}(\text{kvadratMatrise1}) \Rightarrow \text{kvadratMatrise}$

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.

I Radian-vinkelmodus:

$$\sin^{-1}(\{0,0,2,0,5\}) \quad \{0,0,201358,0,523599\}$$

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 >

$\sinh(\text{Utt1}) \Rightarrow \text{Uttrykk}$

$\sinh(\text{Liste1}) \Rightarrow \text{liste}$

$\sinh(\text{Utt1})$  returnerer hyperbolsk sinus til argumentet som et uttrykk.

$\sinh(\text{Liste1})$  returnerer en liste over hyperbolsk sinus til hvert element i *Liste1*.

$\sinh(\text{kvadratMatrise1}) \Rightarrow \text{kvadratMatrise}$

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.

$$\sinh(1.2) \quad 1.50946$$
$$\sinh(\{0,1,2,3\}) \quad \{0,1.50946,10.0179\}$$

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<sup>-1</sup>(Utt<sub>r</sub>l) ⇒ Uttrykksinh<sup>-1</sup>(0) 0sinh<sup>-1</sup>(Liste l) ⇒ listesinh<sup>-1</sup>({0,2,1,3}) {0,1.48748,sinh<sup>-1</sup>(3)}

sinh<sup>-1</sup>(Utt<sub>r</sub>l) returnerer invers hyperbolsk sinus til argumentet som et uttrykk.

sinh<sup>-1</sup>(Liste l) returnerer en liste over invers hyperbolsk sinus til hvert element i Liste l.

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

sinh<sup>-1</sup>  
(kvadratMatrise l) ⇒ kvadratMatrise

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

kvadratMatrise l 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 186.)

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 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 263).

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>

## solve() (løs)

**solve(Ligning, Var)** ⇒ *Boolsk uttrykk*

**solve(Ligning, Var=Forslag)** ⇒ *Boolsk uttrykk*

**solve(Ulikhet, Var)** ⇒ *Boolsk uttrykk*

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \quad \text{or} \quad x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$



Returnerer reelle løsningskandidater til en ligning eller ulikhet for  $Var$ . Målet er å returnere kandidater for alle løsninger. Det kan imidlertid forekomme ligninger eller ulikheter der antall løsninger er uendelig.

Det kan hende at løsningsalternativene ikke er reelle, endelige tall ved enkelte kombinasjoner av verdier for udefinerte variabler.

For autoinnstilling av modusen **Auto/Tilnærmet** er målet å produsere eksakte løsninger når de er konsise, samt supplere med iterative søk med tilnærmet aritmetikk når eksakte løsninger er upraktiske.

På grunn av forkorting av største felles divisor i teller og nevner kan det hende at en "løsning" bare er løsning i den forenkede ligningen eller ulikheten, og ikke i den opprinnelige ligningen/ulikheten.

For ulikheter av typene  $\geq$ ,  $\leq$ ,  $<$ , eller  $>$ , er eksplisitte løsninger ikke sannsynlige hvis ikke ulikheten er lineær og inneholder kun  $Var$ .

Hvis det er valgt Eksakt innstilling i modusen **Auto/Tilnærmet**, blir deler som ikke kan løses, returnert som en implisitt ligning eller ulikhet.

Bruk begrensingsoperatoren (" $|$ ") for å begrense løsningsintervallet og/eller andre variabler som forekommer i ligningen eller ulikheten. Når du finner en løsning i ett intervall, kan du bruke ulikhet-operatorene for å utelate dette intervallet fra senere søk.

usann returneres hvis ingen reelle løsninger blir funnet. sann returneres hvis **solve()** kan bestemme at enhver endelig reell verdi av  $Var$  tilfredsstiller ligningen eller ulikheten.

---

Ans| $a=1$  and  $b=1$  and  $c=1$

$$x = \frac{-1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1}{2} - \frac{\sqrt{3}}{2} \cdot i$$


---

---

solve( $(x-a) \cdot e^x = x \cdot (x-a), x$ )

$$x=a \text{ or } x=0.567143$$


---

---

$(x+1) \cdot \frac{x-1}{x-1} + x-3$

$$2 \cdot x-2$$


---

---

solve( $5 \cdot x-2 \geq 2 \cdot x, x$ )

$$x \geq \frac{2}{3}$$


---

---

exact(solve( $(x-a) \cdot e^x = x \cdot (x-a), x$ ))

$$e^x + x = 0 \text{ or } x = a$$


---

I Radian-vinkelmodus:

---

solve( $(\tan(x) = \frac{1}{x}), x | x > 0 \text{ and } x < 1$ )

$$x = 0.860334$$


---

---

solve( $x = x + 1, x$ )

false

---

solve( $x = x, x$ )

true

Siden **solve()** alltid returnerer et Boolsk resultat, kan du bruke "and," "or," og "not" for å kombinere resultatene fra **solve()** med hverandre eller med andre Boolske uttrykk.

Løsninger kan inneholde en unik, ny udefinert konstant av formen  $nj$ , der  $j$  er et heltall i intervallet 1–255. Slike variabler betegner et vilkårlig heltall.

I reell modus markerer brøkpotenser med oddetallsnevner bare reell forgreining. Ellers markerer multiple forgreinede uttrykk, som brøkpotens, logaritmer og inverse trigonometriske funksjoner, bare hovedforgreiningen. Dermed produserer **solve()** kun løsninger som samsvarer med den ene reelle forgreiningen eller med hovedforgreiningen.

**Merk:** Se også **cSolve()**, **cZeros()**, **nSolve()** og **zeros()**.

**solve(Lign1 and Lign2 [and... ], VarElForslag1, VarElForslag2 [, ... ])** ⇒ Boolsk uttrykk

**solve(LignSystem, VarElForslag1, VarElForslag2 [, ... ])** ⇒ Boolsk uttrykk

**solve({Lign1, Lign2 [,...]} {VarElForslag1, VarElForslag2 [, ... ]})** ⇒ Boolsk uttrykk

Returnerer reelle løsningsalternativer til de simultane, algebraiske ligningene, der hver *VarElForslag* spesifiserer en variabel som du vil løse med hensyn på.

Du kan skille ligningene med and-operatoren, eller du kan legge inn et *LignSystem* ved å bruke en sjablon fra katalogen. Antall argumenter i *VarElForslag* må passe til antallet ligninger. Alternativt kan du spesifisere et startforslag for en variabel. Hver *VarElForslag* må være på formen:

$$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x) \quad x \neq -3 \text{ and } x \leq 1$$

I Radian-vinkelmodus:

$$\text{solve}(\sin(x) = 0, x) \quad x = n \cdot \pi$$

$$\text{solve}\left(x^{\frac{1}{3}} = 1, x\right) \quad x = -1$$

$$\text{solve}(\sqrt{x-2}, x) \quad \text{false}$$

$$\text{solve}(-\sqrt{x-2}, x) \quad x = 4$$

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

variabel

– eller –

variabel = reelt eller ikke-reelt tall

For eksempel er  $x$  gyldig, og det er  $x=3$  også.

Hvis alle ligningene er polynomer og hvis du IKKE spesifiserer noe startforslag, bruker **solve()** Gröbner/Buchbergers leksikale eliminasjonsmetode for å prøve å bestemme alle reelle løsninger.

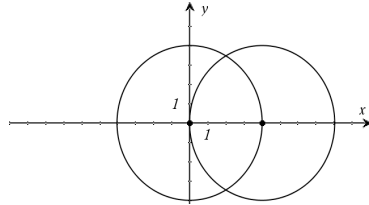
La oss for eksempel anta at du har en sirkel med radius  $r$  om origo og en annen sirkel med radius  $r$  midtstilt der hvor den første sirkelen krysser den positive  $x$ -aksen. Bruk **solve()** for å finne skjæringspunktene.

Som vist med  $r$  i eksemplet til høyre, kan simultane polynomiske ligninger ha ekstra variabler som ikke har noen verdi, men som representerer gitte numeriske verdier som kan settes inn senere.

Du kan også (eller istedenfor) inkludere løsningsvariabler som ikke forekommer i ligningene. For eksempel kan du inkludere  $z$  som en løsningsvariabel for å utvide det forrige eksemplet til to parallelle gjennomskjærende sylindre med radius  $r$ .

Sylinderløsningene viser hvordan løsningsfamilier kan inneholde vilkårlige konstanter i form av  $ck$ , hvor  $k$  er et heltall fra 1 til 255.

For polynomiske systemer kan beregningstiden eller plassen i minnet sterkt avhenge av hvilken rekkefølge du setter løsningsvariabler i. Hvis startforslaget bruker opp minneplassen eller tålmodigheten din, kan du prøve å flytte om på variablene i ligningene og/eller *varElForslag*-listen.






$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1$$

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

Hvis du ikke inkluderer noen forslag og hvis en ligning er ikke-polynomisk i en vilkårlig variabel men alle ligningene er lineære i løsningsvariablene, bruker **solve()** gaussisk eliminasjon for å prøve å bestemme alle reelle løsninger.

Hvis et system verken er polynomisk i alle variablene eller lineær i løsningsvariablene, bestemmer **solve()** som regel en løsning med en tilnærmet iterativ metode. I så fall må antallet løsningsvariabler være lik antallet ligninger, og alle andre variabler (parametre) i ligningene må forenkles til tall.

Hver løsningsvariabel starter ved foreslått verdi hvis den er spesifisert; ellers starter den ved 0.0.




Bruk forslagene til å finne andre løsninger en etter en. For konvergens kan det hende at et forslag må være ganske nært en løsning.

$$\text{solve}\left(x+e^z \cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

$$x=\frac{e^z \cdot \sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^z+1}$$

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y,z\}\right)$$

$$y=2.812\text{E}+10 \text{ and } z=21.9911 \text{ or } y=0.001871\text{I}$$

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

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y,z=2 \cdot \pi\}\right)$$

$$y=0.001871 \text{ and } z=6.28131$$

## SortA (SorterSt)

Katalog > 

**SortA**Liste1[, Liste2] [, Liste3]...

**SortA**Vektor1[, Vektor2] [, Vektor3] ...

Sorterer elementene i det første argument i stigende rekkefølge.

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.

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 263.

$\{2,1,4,3\} \rightarrow \text{list1}$	$\{2,1,4,3\}$
SortA list1	Done
list1	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow \text{list2}$	$\{4,3,2,1\}$
SortA list2,list1	Done
list2	$\{1,2,3,4\}$
list1	$\{4,3,2,1\}$

## SortD (SorterSy)

Katalog > 

**SortD** *Liste1* [, *Liste2*] [, *Liste3*] ...

{2,1,4,3} → *list1* {2,1,4,3}

**SortD** *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

{1,2,3,4} → *list2* {1,2,3,4}

Identisk med **SortA**, bortsett fra at **SortD** sorterer elementene i fallende rekkefølge.

SortD *list1*, *list2* Done

*list1* {4,3,2,1}

*list2* {3,4,1,2}

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 263.

## ►Sphere (sfærisk)

Katalog > 

*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.

**Merk:** For å tvinge frem et tilnærmet desimalresultat,

**Grafregner:** Trykk på  .


**Windows®:** Trykk på **Ctrl+Enter**.

**Macintosh®:** Trykk på +**Enter**.

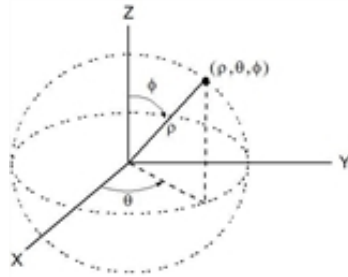
**iPad®:** Hold på **enter**, og velg .

```
[ 1 2 3 ] ► Sphere  
[ 3.74166 1.10715 0.640522 ]
```

```
( ( 2 1.5708 3 ) ) ► Sphere  
[ 3.60555 0.785398 0.588003 ]
```

Trykk på .

```
( ( 2 1.5708 3 ) ) ► Sphere  
[ sqrt(13) 0.785398 sin^-1(2*sqrt(13)/13) ]
```



**sqrt() (kvdrt)**

`sqrt(Uttr1)` ⇒ uttrykk

$$\sqrt{4} \qquad 2$$

`sqrt(Liste1)` ⇒ liste

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

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Liste1*.

**Merk:** Se også **Kvadratrot-sjablon**, side 1.

**stat.results (stat.resultats)**

`stat.results`

$$xlist:=\{1,2,3,4,5\} \qquad \{1,2,3,4,5\}$$

Viser resultater fra en statistisk beregning.

$$ylist:=\{4,8,11,14,17\} \qquad \{4,8,11,14,17\}$$

Resultatene vises som et sett av navn-verdi-par. De spesifikke navnene som vises er avhengige av den aller siste brukte statistikkfunksjonen eller kommandoen.

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.x̄
stat.b9	stat.FBlock	stat. $\hat{p}$	stat.Σx <sup>2</sup>	stat.x̄1
stat.b10	stat.Fcol	stat. $\hat{p}1$	stat.Σxy	stat.x̄2
stat.bList	stat.FInteract	stat. $\hat{p}2$	stat.Σy	stat.x̄Diff
stat.χ <sup>2</sup>	stat.FreqReg	stat. $\hat{p}$ Diff	stat.Σy <sup>2</sup>	stat.x̄List
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() (stAvvPop)

Katalog > 

**stDevPop**(*Liste* [, *frekvListe*]) ⇒ *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 263

**stDevPop**(*Matrise1* [, *FrekvMatrise*]) ⇒ *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 263.

I radian-vinkelmodus og automatisk modus:

$$\text{stDevPop}\{a, b, c\} = \frac{\sqrt{2 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3}$$

$$\text{stDevPop}\{1, 2, 5, -6, 3, -2\} = \frac{\sqrt{465}}{6}$$

$$\text{stDevPop}\{1.3, 2.5, -6.4\}, \{3, 2, 5\} = 4.11107$$

$$\text{stDevPop} \begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix} \begin{bmatrix} 4 \cdot \sqrt{6} & \sqrt{78} & 2 \cdot \sqrt{6} \\ 3 & 3 & 3 \end{bmatrix}$$

$$\text{stDevPop} \begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix} \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix} = \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix}$$



**stDevSamp**(Liste[, frekvListe]) ⇒ 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 ignorerer. For mer informasjon om tomme elementer, se side 263

**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 ignorerer. For mer informasjon om tomme elementer, se side 263.

$$\text{stDevSamp}\{\{a,b,c\}\}$$

$$\frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3}$$

$$\text{stDevSamp}\{\{1,2,5,-6,3,-2\}\} \quad \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}\{\{1.3,2.5,-6.4\},\{3,2,5\}\} \quad 4.33345$$

$$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) \quad \left[4 \quad \sqrt{13} \quad 2\right]$$

$$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) \quad \left[2.7005 \quad 5.44695\right]$$

## Stop (Stopp)

### Stop

Programmeringskommando: Avslutter programmet.

**Stop** er ikke tillatt i funksjoner.

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

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

## Lagre

Se → (lagre), side 245.

**String() (Streng)**Katalog > **string**(*Utr*) ⇒ *string*Forenkler *Utr* og returnerer resultatet som en tegnstring.

<code>string(1.2345)</code>	"1.2345"
<code>string(1+2)</code>	"3"
<code>string(cos(x)+√3)</code>	"cos(x)+√(3)"

**subMat() (undermatrise)**Katalog > **subMat**(*MatriseI* [, *startRad*] [, *startKol*] [, *endRad*] [, *endKol*]) ⇒ *matrise*Returnerer den spesifiserte undermatrisen til *MatriseI*.Grunninnstillinger: *startRad*=1, *startKol*=1, *endRad*=siste rad, *endKol*=siste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>subMat(m1,2,1,3,2)</code>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
<code>subMat(m1,2,2)</code>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

**Sum (Sigma)**

Se G(), side 236.

**sum()**Katalog > **sum**(*Liste* [, *Start*] [, *Slutt* ]) ⇒ *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 263.**sum**(*MatriseI* [, *Start*] [, *Slutt* ]) ⇒ *matrise*Returnerer en radvektor som inneholder summene av elementene i kolonnene i *MatriseI*.*Start* og *Slutt* er alternativer. De spesifiserer et radområde.

<code>sum({1,2,3,4,5})</code>	15
<code>sum({a,2·a,3·a})</code>	6·a
<code>sum(seq(n,n,1,10))</code>	55
<code>sum({1,3,5,7,9},3)</code>	21

<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2,3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *Matrise1* ignorerer. For mer informasjon om tomme elementer, se side 263.

## sumIf()

**sumIf**(*Liste*,*Kriterium*[,  
*SumListe*]) $\Rightarrow$ verdi

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 samlede 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 263.

**Merk:** Se også **countIf()**, side 37.

$\text{sumIf}(\{1,2,e,3,\pi,4,5,6\}, 2.5 < ? < 4.5)$	
	$e + \pi + 7$
$\text{sumIf}(\{1,2,3,4\}, 2 < ? < 5, \{10,20,30,40\})$	
	70

## system()

Katalog > 

system(Uttr1 [, Uttr2 [, Uttr3 [, ...]])

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

system(Ekv1 [, Ekv2 [, Ekv3 [, ...]])

Returnerer et ligningssystem, formatert som en liste. Du kan også opprette et system med en sjablon.

**Merk:** Se også **Ligningssystemer**, side 3.

## T

## T(transponert)

katalog > Matrise I  $\Rightarrow$  matrise

Returnerer den komplekse konjugerte transponerte av *Matrise1*.

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

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T$	$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T$	$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$
$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T$	$\begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$

## tan()

 tasttan(Uttr1)  $\Rightarrow$  Uttrykk

I Grader-vinkelmodus:

tan(Liste1)  $\Rightarrow$  liste

tan(Uttr1) returnerer tangens til argumentet som et uttrykk.

tan(Liste1) returnerer en liste over tangensene til alle elementene i *Liste1*.

**Merk:** Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke °, G, eller  $\text{r}$  for å hoppe over vinkelmodusen midlertidig.

I Gradian-vinkelmodus:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45)$	1
$\tan\{0,60,90\}$	$\{0,\sqrt{3},\text{undef}\}$
$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(50)$	1
$\tan\{0,50,100\}$	$\{0,1,\text{undef}\}$

## tan()

 **tast**

**tan(kvadratMatrise1)** ⇒ kvadratMatrise

Returterer 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.

I Radian-vinkelmodus:

$$\begin{array}{r} \tan\left(\frac{\pi}{4}\right) \qquad \qquad \qquad 1 \\ \hline \tan(45^\circ) \qquad \qquad \qquad 1 \\ \hline \tan\left(\left\{\pi, \frac{\pi}{3}, -\pi, \frac{\pi}{4}\right\}\right) \qquad \qquad \qquad \{0, \sqrt{3}, 0, 1\} \end{array}$$

I Radian-vinkelmodus:

$$\begin{array}{r} \tan\left(\begin{array}{ccc} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{array}\right) \\ \hline \begin{array}{ccc} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{array} \end{array}$$

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

 **tast**

**tan<sup>-1</sup>(Uttr1)** ⇒ Uttrykk

**tan<sup>-1</sup>(Liste1)** ⇒ liste

**tan<sup>-1</sup>(Uttrykk1)** returnerer vinkelen med tangens lik Uttr1, som et uttrykk.

**tan<sup>-1</sup>(Liste1)** returnerer en liste over de inverse tangenser til hvert element i Liste1.

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

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

**tan<sup>-1</sup>**  
(kvadratMatrise1) ⇒ kvadratMatrise

Returterer 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()**.

I Grader-vinkelmodus:

$$\tan^{-1}(1) \qquad \qquad \qquad 45$$

I Gradian-vinkelmodus:

$$\tan^{-1}(1) \qquad \qquad \qquad 50$$

I Radian-vinkelmodus:

$$\tan^{-1}(\{0,0,2,0,5\}) \qquad \{0,0.197396,0.463648\}$$

I Radian-vinkelmodus:

$$\begin{array}{r} \tan^{-1}\left(\begin{array}{ccc} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{array}\right) \\ \hline \begin{array}{ccc} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{array} \end{array}$$

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

trig 

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

## tangentLine()

katalog > 

**tangentLine(Uttr1,Var,Punkt)**⇒uttrykk

$\text{tangentLine}(x^2,x,1)$	$2 \cdot x - 1$
-------------------------------	-----------------

**tangentLine(Uttr1,Var=Punkt)**⇒uttrykk

$\text{tangentLine}((x-3)^2-4,x=3)$	$-4$
-------------------------------------	------

Returnerer tangentlinjen til kurven som er representert av *Uttr1* i punktet som er spesifisert i *Var=Punkt*.

$\text{tangentLine}\left(x^{\frac{1}{3}},x=0\right)$	$x=0$
--	-------

Sørg for at den uavhengige variabelen ikke er definert. Hvis for eksempel  $f1(x):=5$  og  $x:=3$ , vil **tangentLine(f1(x),x,2)** returnere "false" ("usant").

$\text{tangentLine}(\sqrt{x^2-4},x=2)$	undef
--	-------

$x:=3: \text{tangentLine}(x^2,x,1)$	5
-------------------------------------	---

## tanh()

Katalog > 

**tanh(Uttr1)**⇒Uttrykk

$\text{tanh}(1.2)$	0.833655
--------------------	----------

**tanh(Liste1)**⇒liste

$\text{tanh}(\{0,1\})$	$\{0,\text{tanh}(1)\}$
------------------------	------------------------

**tanh(Uttr1)** returnerer hyperbolsk tangens til argumentet som et uttrykk.

**tanh(Liste1)** returnerer en liste av hyperbolske tangenser til hvert element i *Liste1*.

### tanh

(*kvadratMatrise1*)⇒*kvadratMatrise*

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

I Radian-vinkelmodus:

$\text{tanh}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$
---	---

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

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

Katalog >

$\tanh^{-1}(\text{Uttr1}) \Rightarrow \text{Uttrykk}$

I rektangulært, kompleks format:

$\tanh^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

$\tanh^{-1}(\text{Uttrykk1})$  returnerer invers hyperbolsk tangens til argumentet som et uttrykk.

$\tanh^{-1}(\text{Liste1})$  returnerer en liste over invers hyperbolsk tangens til hvert element i *Liste1*.

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

$\tanh^{-1}$   
(*kvadratMatrise1*)  $\Rightarrow$  *kvadratMatrise*

I radian-vinkelmodus og rektangulært, kompleks format:

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

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

$$\tanh^{-1}\left(\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}\right)$$
$$\begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.94730 \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

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

## **taylor()**

Katalog >

$\text{taylor}(\text{Uttr1}, \text{Var}, \text{Orden}, \text{Punkt1}) \Rightarrow \text{uttrykk}$

Returnerer etterspurt Taylor polynom. Polynomet inkluderer ikke-null-ledd med grader i heltall fra null til *Orden* i (*Var* minus *Punkt1*). **taylor()** returnerer seg selv hvis det ikke er noen kuttet potensrekke av denne orden, eller hvis den krever negative eksponenter eller brøkeksponenter. Bruk substitusjon og/eller midlertidig multiplikasjon med en potens av (*Var* minus *Punkt1*) for å bestemme mer generell potensrekke.

*Punkt1* grunninnstilles til null og er "utvidelsespunktet".

$$\text{taylor}(e^{\sqrt{x}}, x, 2) \quad \text{taylor}(e^{\sqrt{x}}, x, 2, 0)$$
$$\text{taylor}(e^{t, 4})|_{t=\sqrt{x}} \quad \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1$$
$$\text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \quad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right)$$
$$\text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right)$$
$$-x^3 - x^2 - x - \frac{1}{x} - 1$$

**tCdf**(*nedGrense*,*øvreGrense*,*df*) $\Rightarrow$  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* =  $-\infty$ .

## tCollect() (tSlåSmm)

**tCollect**(*Uttr1*) $\Rightarrow$ uttrykk

Returnerer et uttrykk der produkter og heltallspotenser av sinus og cosinus er omregnet til en lineær kombinasjon av sinus og cosinus med flersifrede vinkler, vinkelsummer og vinkelforskjeller. Transformasjonene omregner trigonometriske polynomer til en lineær kombinasjon av deres harmoniske ekvivalenter.

Noen ganger vil **tCollect()** oppnå en ønsket forenkling når den grunninnstilte trigonometriske forenklingen ikke gjør det. **tCollect()** kan omgjøre endringer som ble gjort med **tExpand()**. Noen ganger kan et uttrykk forenkles hvis du bruker **tExpand()** på et resultat fra **tCollect()**, eller omvendt, i to separate omganger.

$$\frac{\text{tCollect}(\cos(\alpha)^2)}{\text{tCollect}(\sin(\alpha) \cdot \cos(\beta))} = \frac{\cos(2 \cdot \alpha) + 1}{2} \cdot \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2}$$

## tExpand() (tUtvid)

**tExpand**(*Uttr1*) $\Rightarrow$ uttrykk

Returnerer et uttrykk der sinus og cosinus av flersifrede heltallsvinkler, vinkelsummer og vinkelforskjeller er utvidet. På grunn av identiteten  $(\sin(x))^2 + (\cos(x))^2 = 1$ , er det mange mulige ekvivalente resultater. Derfor kan et resultat være forskjellig fra et resultat som vises i andre publikasjoner.

$$\frac{\text{tExpand}(\sin(3 \cdot \phi))}{\text{tExpand}(\cos(\alpha - \beta))} = \frac{4 \cdot \sin(\phi) \cdot (\cos(\phi))^2 - \sin(\phi)}{\cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta)}$$



Noen ganger vil **tExpand()** oppnå målene dine når den grunninnstilte trigonometriske forenklingen ikke gjør det. **tExpand()** kan omgjøre endringer som ble gjort med **tCollect()**. Noen ganger kan et uttrykk forenkles hvis du bruker **tCollect()** på et resultat fra **tExpand()**, eller omvendt, i to separate omganger.

**Merk:** Grader-modus-skalering med  $\pi/180$  kommer i konflikt med **tExpand()** sin evne til å gjenkjenne uttrykk som kan utvides. For beste resultater bør **tExpand()** brukes i radian modus.

## Text

**TextpromptStreng[, VisFlagg]**

Programmeringskommando: Stopper programmet og viser tegnstringen *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 157 eller **RequestStr**, side 159.

**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.

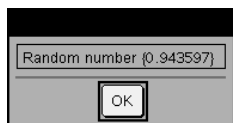
Innenfor malen Prgm...EndPrgm fullfører du hver linje ved å trykke på  istedenfor på **enter**. På tastaturet på datamaskinen, hold nede **Alt** og trykk på **Enter**.

```
Define tekst_demo()=Prgm
  For i,1,5
    stringo:="Random number " &
string(rand(i))
    Text stringo
  EndFor
EndPrgm
```

Kjør program:

tekst\_demo()

Eksempel på dialogboks:



**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 186.)

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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 186).

*Felles*=1 fellesvarianser; *Felles*=0 gjør ikke fellesvarianser.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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</i> = JA.

## tmpCnv()

**tmpCnv**(Utrr °tempEnhet1, \_ °tempEnhet2) ⇒ uttrykk \_ °tempEnhet2

Omregner en temperaturverdi spesifisert av *Utrr* fra en enhet til en annen. Gyldige temperaturenheter er:

\_ °C Celsius

\_ °F Fahrenheit

\_ °K Kelvin

\_ °R Rankine

Velg fra symboler i katalogen for å skrive

°.

for å skrive \_, trykk på  .

For eksempel, 100\_°C omregnes til 212\_°F.

For å omregne et temperaturområde, bruk **ΔtmpCnv()** istedenfor.

tmpCnv(100_°C, _°F)	212_°F
tmpCnv(32_°F, _°C)	0_°C
tmpCnv(0_°C, _°K)	273.15_°K
tmpCnv(0_°F, _°R)	459.67_°R

**Merk:** Du kan bruke katalogen for å velge temperaturenheter.

## $\Delta$ tmpCnv()

Katalog > 

$\Delta$ tmpCnv(Uttrykk  $_{\text{tempEnhet}}$ ,  $_{\text{tempEnhet2}}$ )  $\Rightarrow$  uttrykk  $_{\text{tempEnhet2}}$

**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive `del taTmpCnv (...)`.

Konverterer et temperaturområde (forskjellen mellom to temperaturverdier) som er angitt ved *Uttrykk* fra en måleenhet til en annen. Gyldige temperaturenheter er:


$_{\text{CCelsius}}$

$_{\text{FFahrenheit}}$

$_{\text{KKelvin}}$

$_{\text{RRankine}}$

Hvis du skal sette inn  $^{\circ}$ , kan du velge det fra symbolpaletten eller skrive `ød`.

For å skrive  $_{\text{C}}$  trykk på `ctrl` .

$1_{\text{C}}$  og  $1_{\text{K}}$  har samme størrelse, og det har også  $1_{\text{F}}$  og  $1_{\text{R}}$ . Men  $1_{\text{C}}$  er 9/5 så stor som  $1_{\text{F}}$ .

For eksempel, et  $100_{\text{C}}$ -område (fra  $0_{\text{C}}$  til  $100_{\text{C}}$ ) er ekvivalent til et  $180_{\text{F}}$ -område.

For å omregne en spesiell temperaturverdi istedenfor et -område, bruk `tmpCnv()`.

Velg fra symboler i katalogen for å skrive  $\Delta$ .

$\Delta$ tmpCnv( $100_{\text{C}}$ , $_{\text{F}}$ )	$180_{\text{F}}$
$\Delta$ tmpCnv( $180_{\text{F}}$ , $_{\text{C}}$ )	$100_{\text{C}}$
$\Delta$ tmpCnv( $100_{\text{C}}$ , $_{\text{K}}$ )	$100_{\text{K}}$
$\Delta$ tmpCnv( $100_{\text{F}}$ , $_{\text{R}}$ )	$100_{\text{R}}$
$\Delta$ tmpCnv( $1_{\text{C}}$ , $_{\text{F}}$ )	$1.8_{\text{F}}$

**Merk:** Du kan bruke katalogen for å velge temperaturenheter.

## tPdf()

Katalog > 

$tPdf(XVerd, df) \Rightarrow$  tall hvis *XVerd* er et tall, liste hvis *XVerd* er en liste

Beregner sannsynlighetstetthetsfunksjonen (pdf) for Student-*t*-fordelingen ved en spesifisert *x*-verdi med spesifiserte grader av frihet *df*.

**trace**(*kvadratMatrise*)⇒uttrykk

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
$\text{trace}\left(\begin{bmatrix} a & 0 \\ 1 & a \end{bmatrix}\right)$	$2 \cdot a$

## Try

### 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 273.

*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 flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

### 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.

---


$$\text{egenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$


---

---


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


---

Define `progI()`=Prgm

Try

`z:=z+1`

Disp "z incremented."

Else

Disp "Sorry, z undefined."

EndTry

EndPrgm

Done

`z:=1:progI()`

z incremented.

Done

DelVar `z:progI()`

Sorry, z undefined.

Done

Define `egenverdier(a,b)`=Prgm

© Programmet `egenverdier(A,B)` viser egenverdier av  $A \cdot B$

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Egenverdier av A·B er:",egVd(a\*b)

Else

If `feilKode=230` Then

**Merk:** Se også **ClrErr**, side 26, og **PassErr**, side 138.

Disp "Feil: Produkt av A·B må være en kvadratmatrise"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrgm

## tTest

**tTest**  $\mu_0$ ,*Liste*[,*Frekv*[,*Hypot*]]

(Dataliste inndata)

**tTest**  $\mu_0$ , $\bar{x}$ ,*sx*,*n*, [*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 186).

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 263).

Utdata-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \text{sqrt}(n))$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader-av-frihet

Utdata-variabel	Beskrivelse
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 186).

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 263).

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>
stat.n1, stat.n2	Utvalgenes størrelse

Utdata-variabel	Beskrivelse
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 205. Se også **amortTbl()**, side 8.

### 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 205. Se også **amortTbl()**, side 8.

### 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 205. Se også **amortTbl()**, side 8.

### 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 205. Se også **amortTbl()**, side 8.

**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 205. Se også **amortTbl()**, side 8.

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 186.)

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 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 263.

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
stat. y	Sum av y-verdier

Utdata-variabel	Beskrivelse
stat. $y^2$	Sum av $y^2$ -verdier
stat.sy	Utvalgets standardavvik til y
stat. y	Populasjonens standardavvik til y
stat. xy	Sum av $x \cdot y$ -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. $(x-)^2$	Sum av kvadratavvik fra gjennomsnittet av x
stat. $(y-)^2$	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.

$$\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) = \left[ \frac{a}{\sqrt{a^2+b^2+c^2}} \quad \frac{b}{\sqrt{a^2+b^2+c^2}} \quad \frac{c}{\sqrt{a^2+b^2+c^2}} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) = \left[ \frac{\sqrt{6}}{6} \quad \frac{\sqrt{6}}{3} \quad \frac{\sqrt{6}}{6} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{\sqrt{14}}{7} \\ \frac{3 \cdot \sqrt{14}}{14} \end{bmatrix}$$

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

## unLock

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

**unLock***Var*.

Låser opp spesifisert variabel eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes.

Se **Lock**, side 112, og **getLockInfo()**, side 88.

<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

## V

## varPop()

**varPop**(*Liste* [, *frekvListe*]) ⇒ *uttrykk*

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 263.

varPop({5,10,15,20,25,30})	875
	12
Ans·1.	72.9167

**varSamp**(*Liste* [, *frekvListe*]) ⇒ uttrykk

Returnerer utvalgets varians 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 263.

**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 263.

**Merk:** *Matrise1* må inneholde minst to rader.

$\text{varSamp}(\{a, b, c\})$

$$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$$

$\text{varSamp}(\{1, 2, 5, 6, 3, 2\})$

$\frac{31}{2}$

$\text{varSamp}(\{1, 3, 5\}, \{4, 6, 2\})$

$\frac{68}{33}$

$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right)$  [4.75 1.03 4]

$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right)$  [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.

For å vente 4 sekunder:



**Wait 4**

For å vente 1/2 sekund:

**Wait 0.5**

Argumentet *tiden**Sekunder* 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.
- **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:** Du kan bruke kommandoen **Wait** innenfor et brukerdefinert program, men ikke innenfor en funksjon.

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"**

## warnCodes ()

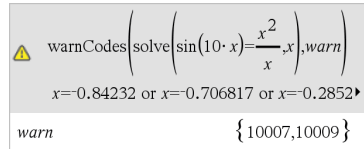
**warnCodes**(*Uttr1*, *StatusVar*) $\Rightarrow$ uttrykk

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 281 for en liste over varselkoder og assosierte meldinger.



The screenshot shows the TI-Nspire CAS interface. At the top, a yellow warning triangle is visible. The input is `warnCodes(solve(sin(10*x)=x^2/x),warn)`. Below the input, the output is displayed as `x=-0.84232 or x=-0.706817 or x=-0.2852`. At the bottom, the variable `warn` is assigned the value `{10007,10009}`.

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

## when() (når)

**when**(*Betingelse*, *santResultat* [, *usantResultat*][, *ukjentResultat*])  
 $\Rightarrow$ uttrykk

## when() (når)

Katalog > 

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.

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
-------------------------------------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n - 1), 1) \rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$	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
$\text{sum\_of\_recip}(3)$	$\frac{11}{6}$
	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 136.

*Heltall1* xor *Heltall2* ⇒ *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 18.

**Merk:** Se or, side 136.

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.



**zeros() (nullpkt)****zeros(Uttr, Var)⇒liste****zeros(Uttr, Var=Forslag)⇒liste**Returnerer en liste med kandidater til reelle verdier av *Var* som gjør  $Uttr=0$ .**zeros()** gjør dette ved å beregne **expliste** (**solve(Uttr=0,Var),Var**).

Noen ganger kan resultatformen for **zeros()** være mer praktisk enn den for **solve()**. Men resultatformen for **zeros()** kan ikke uttrykke implisitte løsninger, løsninger som krever ulikheter, eller løsninger som ikke involverer *Var*.

**Merk:** Se også **cSolve()**, **cZeros()** og **solve()**.**zeros({Uttr1, Uttr2}, {VarElForslag1, VarElForslag2 [, ... ]})⇒matrise**Returnerer alternative reelle nullpunkter for simultane algebraiske uttrykk, der hvert *VarElForslag* spesifiserer en ukjent som du vil finne verdien til.Alternativt kan du spesifisere et startforslag for en variabel. Hvert *VarElForslag* må ha formen:*variabel*

– eller –

*variabel = reelt eller ikke-reelt tall*For eksempel er  $x$  gyldig, og det er  $x=3$  også.

Hvis alle uttrykkene er polynomiske og hvis du IKKE spesifiserer noe startforslag, bruker **zeros()** Gröbner/Buchbergers leksikale eliminasjonsmetode for å prøve å bestemme alle reelle nullpunkter.

$$\text{zeros}\left(\frac{a \cdot x^2 + b \cdot x + c, x}{\left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} + b}{2 \cdot a} \right\}}\right)$$


---


$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

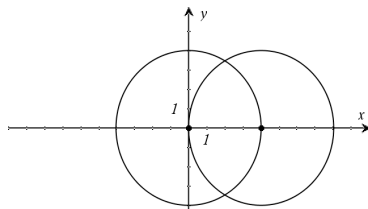
$$\text{exact}\left(\text{zeros}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right), x\right)\right) \quad \{\emptyset\}$$


---


$$\text{exact}\left(\text{solve}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right) = 0, x\right)\right)$$


---


$$e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0$$



La oss for eksempel anta at du har en sirkel med radius  $r$  om origo og en annen sirkel med radius  $r$  midtstilt der hvor den første sirkelen krysser den positive  $x$ -aksen. Bruk **zeros()** for å finne skjæringspunktene.

Som vist med  $r$  i eksemplet til høyre, kan simultane polynomiske uttrykk ha ekstra variabler som ikke har noen verdi, men som representerer gitte numeriske verdier som kan legges til senere.

Hver rad i resultatmatrisen representerer et alternativt (annet) nullpunkt, med komponentene plassert som i *VarEIForslag*-listen. For å trekke ut en rad, pek på matrisen med [*rad*].

Du kan også (eller istedenfor) inkludere ukjente som ikke forekommer i uttrykkene. For eksempel kan du inkludere  $z$  som en ukjent for å utvide det forrige eksemplet til to parallelle gjennomskjærende sylindere med radius  $r$ . Sylindernullpunktene viser hvordan løsningsfamilier av nullpunkter kan inneholde vilkårlige konstanter i form av  $ck$ , hvor  $k$  er et heltall fra 1 til 255.

For polynomiske systemer kan beregningstiden eller plassen i minnet sterkt avhenge av hvilken rekkefølge du setter de ukjente i. Hvis startforslaget bruker opp minneplassen eller tålmodigheten din, kan du prøve å flytte om på variablene i uttrykkene og/eller *VarEIForslag*-listen.

Hvis du ikke inkluderer noen forslag og hvis et uttrykk er ikke-polynomisk i en vilkårlig variabel men alle uttrykkene er lineære i alle ukjente, bruker **zeros()** gaussisk eliminering for å prøve å bestemme alle reelle nullpunkter.

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Trekk ut rad 2:

$$\text{Ans}[2] \quad \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} & c1 \end{bmatrix}$$

$$\text{zeros}\left(\left\{x+e^z\cdot y-1, x-y-\sin(z)\right\}, \{x, y\}\right)$$

$$\begin{bmatrix} \frac{e^z\cdot \sin(z)+1}{e^z+1} & \frac{-\sin(z)-1}{e^z+1} \end{bmatrix}$$

Hvis et system er verken polynomisk i alle variablene eller lineært i de ukjente, bestemmer **zeros()** som regel ett nullpunkt med en tilnærmet iterativ metode. I så fall må antallet ukjente være lik antallet uttrykk, og alle andre variabler i uttrykkene må forenkles til tall.

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z\}\right)$$

0.041458	3.18306
0.001871	6.28131
4.76E-11	1796.99
2.E-13	254.469

Hver ukjente starter ved foreslått verdi hvis den er spesifisert; ellers starter den ved 0.0.

Bruk forslagene til å finne andre løsninger en etter en. For konvergens kan det hende at et forslag må være ganske nært et nullpunkt.

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z = 2 \cdot \pi\}\right)$$

0.001871	6.28131
----------	---------

## zInterval

**zInterval**  $\sigma, \text{Liste}, [\text{Frekv}], [\text{CNivå}]$

(Dataliste inndata)

**zInterval**  $\sigma, \bar{x}, n, [\text{CNivå}]$

(Summering statistikk inndata)

Beregner et  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen (side 186).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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**  $x, n$  [,CNivå]

Beregner et en-proporsjons  $z$  konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

$x$  er et ikke-negativt heltall.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 263).

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 186).

$x1$  og  $x2$  er ikke-negative heltall.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 263).

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

Utdata-variabel	Beskrivelse
stat.n1	Utvalgsstørrelse i datasekvens 1
stat.n2	Utvalgsstørrelse i datasekvens 2

## zInterval\_2Samp

Katalog > 

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \text{Liste1}, \text{Liste2}$   
 $[, \text{Frekv1}, \text{Frekv2}, [\text{CNivå}]]$

(Dataliste inndata)

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \bar{x}_1, n1, \bar{x}_2, n2 [\text{CNivå}]$

(Summering statistikk inndata)

Beregner et to-utvalgs  $z$  konfidensintervall.

En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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. $\bar{x}_1$ , stat. $\bar{x}_2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma_1$ , stat. $\sigma_2$	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**  $\mu, \sigma, \text{Liste}, [\text{Frekv}, \text{Hypot}]$

(Dataliste inndata)

**zTest**  $\mu, \sigma, \bar{x}, n, [\text{Hypot}]$

(Summering statistikk inndata)

Utfører en  $z$ -test med frekvens *frekvlste*. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

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  $H_a: \mu > \mu_0$ , sett *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

Utdata-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
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

**zTest\_1Prop**  $p_0, x, n[,Hypot]$

Beregner en en-proporsjons  $z$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

$x$  er et ikke-negativt heltall.

Test  $H_0: p = p_0$  mot ett av følgende:

For  $H_a: p > p_0$ , sett *Hypot*>0

For  $H_a: p \neq p_0$  (standard), sett *Hypot*=0

For  $H_a: p < p_0$ , sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 263).

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

Katalog > 

### zTest\_2Prop $x1, n1, x2, n2[, Hypot]$

Beregner en to-proporsjons z-test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

$x1$  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 263).

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

Katalog > 

### zTest\_2Samp $\sigma_1, \sigma_2, Liste1, Liste2[, Frekv1$ $[, Frekv2[, Hypot]]]$

(Dataliste inndata)

**zTest\_2Samp**  $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2, Hypot$ 

(Summering statistikk inndata)

Beregner en to-utvalgs  $z$ -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 186).

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 263).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for forskjellen i gjennomsnitt
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
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)



$Uttr1 + Uttr2 \Rightarrow uttrykk$

56	56
----	----

Returnerer summen av de to argumentene.

$56+4$	60
--------	----

$60+4$	64
--------	----

$64+4$	68
--------	----

$68+4$	72
--------	----

$Liste1 + Liste2 \Rightarrow liste$

$\left\{22, \pi, \frac{\pi}{2}\right\} \rightarrow l1$	$\left\{22, \pi, \frac{\pi}{2}\right\}$
--	---

$Matrise1 + Matrise2 \Rightarrow matrise$

$\left\{10, 5, \frac{\pi}{2}\right\} \rightarrow l2$	$\left\{10, 5, \frac{\pi}{2}\right\}$
--	---------------------------------------

Returnerer en liste (eller matrise) som inneholder summene av tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrise1* og *Matrise2*).

$l1+l2$	$\{32, \pi+5, \pi\}$
---------	----------------------

$Ans + \{\pi, 5, \pi\}$	$\{\pi+32, \pi, 0\}$
-------------------------	----------------------

Dimensjonene i argumentene må være like.

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
---	--

$Uttr + Liste1 \Rightarrow liste$

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
-----------------------	------------------

$Liste2 + Uttr \Rightarrow liste$

$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$
-----------------------	------------------

Returnerer en liste som inneholder summene av *Uttr* og hvert element i *Liste1*.

$Uttr + Matrise1 \Rightarrow matrise$

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

$Matrise1 + Uttr \Rightarrow matrise$

Returnerer en matrise med *Uttr* 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)



$Uttr1 - Uttr2 \Rightarrow uttrykk$

$6-2$	4
-------	---

Returnerer *Uttr1* minus *Uttr2*.

$\pi - \frac{\pi}{6}$	$\frac{5 \cdot \pi}{6}$
-----------------------	-------------------------

**-(subtrahere)** $Liste1 - Liste2 \Rightarrow liste$ 

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} = \left\{ 12, \pi - 5, 0 \right\}$$

 $Matrise1 - Matrise2 \Rightarrow matrise$ 

$$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \end{bmatrix}$$

Subtraherer hvert element i *Liste2* (eller *Matrise2*) fra tilsvarende element i *Liste1* (eller *Matrise1*), og returnerer resultatene.

Dimensjonene i argumentene må være like.

 $Uttr - Liste1 \Rightarrow liste$ 

$$20 - \{10, 15, 20\} = \{5, 0, -5\}$$

 $Liste1 - Uttr \Rightarrow liste$ 

$$\{10, 15, 20\} - 15 = \{-5, 0, 5\}$$

Subtraherer hvert *Liste1* element fra *Uttr* eller subtraherer *Uttr* fra hvert *Liste1* element og returnerer en liste over resultatene.

 $Uttr - Matrise1 \Rightarrow matrise$ 

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

 $Matrise1 - Uttr \Rightarrow matrise$ 

*Uttr - Matrise1* returnerer en matrise av *Uttr* hver gang identitetsmatrisen trekkes fra *Matrise1*. *Matrise1* må være kvadratisk.

*Matrise1 - Uttr* returnerer en matrise av *Uttr* hver gang identitetsmatrisen subtraheres fra *Matrise1*. *Matrise1* må være kvadratisk.

**Merk:** Bruk .- (prikk minus) for å subtrahere et uttrykk fra hvert element.

**•(multiplisere)** $Uttr1 \cdot Uttr2 \Rightarrow uttrykk$ 

$$2 \cdot 3.45 = 6.9$$

Returnerer produktet av de to argumentene.

$$x \cdot y \cdot x = x^2 \cdot y$$

 $Liste1 \cdot Liste2 \Rightarrow liste$ 

$$\{1, 2, 3\} \cdot \{4, 5, 6\} = \{4, 10, 18\}$$

Returnerer en liste som inneholder produktene av samsvarende elementer i *Liste1* og *Liste2*.

$$\left\{ \frac{2}{a}, \frac{3}{2} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} = \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

Dimensjonene i listene må være like.

**•(multiplisere)****x** tast*Matrise1*•*Matrise2*⇒*matrise*Returnerer matriseproduktet av *Matrise1* og *Matrise2*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} = \begin{bmatrix} a+2\cdot b+3\cdot c & d+2\cdot e+3\cdot f \\ 4\cdot a+5\cdot b+6\cdot c & 4\cdot d+5\cdot e+6\cdot f \end{bmatrix}$$

Antallet kolonner i *Matrise1* må være likt antallet rader i *Matrise2*.*Uttr*•*Liste1*⇒*liste*

$$\pi \cdot \{4,5,6\} = \{4\cdot\pi, 5\cdot\pi, 6\cdot\pi\}$$

*Liste1*•*Uttr*⇒*liste*Returnerer en liste som inneholder produktene av *Uttr* og hvert element i *Liste1*.*Uttr*•*Matrise1*⇒*matrise**Matrise1*•*Uttr*⇒*matrise*Returnerer en matrise som inneholder produktene av *Uttr* og hvert element i *Matrise1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$1 \cdot \text{identity}(3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Merk:** Bruk **•**(prikk multipliser) for å multiplisere et uttrykk med hvert element.**/ (divider)****÷** tast*Uttr1* / *Uttr2*⇒*uttrykk*Returnerer kvotienten av *Uttr1* dividert med *Uttr2*.

$$\frac{2}{3.45} = 0.57971$$

$$\frac{x^3}{x} = x^2$$

**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.

*Uttr* / *Liste1* ⇒ *liste**Liste1* / *Uttr* ⇒ *liste*Returnerer en liste som inneholder kvotientene av *Uttr* dividert med *Liste1* eller *Liste1* dividert med *Uttr*.

$$\frac{a}{\{3,a,\sqrt{a}\}} = \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a,b,c\}}{a\cdot b\cdot c} = \left\{\frac{1}{b\cdot c}, \frac{1}{a\cdot c}, \frac{1}{a\cdot b}\right\}$$

## / (divider)



*Matrise1 / Uttr* ⇒ *matrise*

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} \qquad \frac{\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}}{b \cdot c \quad a \cdot c \quad a \cdot b}$$

Returnerer en matrise som inneholder kvotientene av *Matrise1 / Uttr*.

**Merk:** Bruk . / (prikk divider) for å dividere et uttrykk med hvert element.

## ^ (potens)



*Uttr1* ^ *Uttr2* ⇒ *Uttrykk*

$$4^2 \qquad 16$$

*Liste1* ^ *Liste2* ⇒ *liste*

$$\{a, 2, c\}^{\{1, b, 3\}} \qquad \{a, 2^b, c^3\}$$

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.

*Uttr* ^ *Liste1* ⇒ *liste*

$$p^{\{a, 2, -3\}} \qquad \left\{ p^a, p^2, \frac{1}{p^3} \right\}$$

Returnerer *Uttr* opphøyd i elementene i *Liste1*.

*Liste1* ^ *Uttr* ⇒ *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 *Uttr*.

*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***Uttr1*<sup>2</sup> ⇒ *Uttrykk*

$4^2$	16
-------	----

Returnerer kvadratet av argumentet.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

*Liste1*<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*.*kvadratMatrise1*<sup>2</sup> ⇒ *matrise*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \wedge 2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
--	--

Returnerer matrisens kvadrat av *kvadratMatrise1*. Dette er ikke det samme som å beregne kvadratet av hvert element. Bruk  $\wedge 2$  for å beregne kvadratet av hvert element.**+. (prikk adder)** **taster***Matrise1* .+ *Matrise2* ⇒ *matrise*

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .+ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
--	--

*Uttr* .+ *Matrise1* ⇒ *matrise*

$x .+ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$
---	--

*Matrise1* .+ *Matrise2* returnerer en matrise som er summen av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.*Uttr* .+ *Matrise1* returnerer en matrise som er summen av *Uttr* og hvert element i *Matrise1*.**.- (prikk subtr.)** **taster***Matrise1* .- *Matrise2* ⇒ *matrise*

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .- \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
--	--

*Uttr* .- *Matrise1* ⇒ *matrise*

$x .- \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$
---	--

*Matrise1* .- *Matrise2* returnerer en matrise som er differansen mellom hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.*Uttr* .- *Matrise1* returnerer en matrise som er differansen av *Uttr* og hvert element i *Matrise1*.

**.• (prikk mult.)**

.	x	taster
---	---	--------

Matrise1 .• Matrise2 ⇒ matrise

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix}$	.	$\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$
--	---	--	--

Uttr .• Matrise1 ⇒ matrise

$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$\begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$
--	--

Matrise1 .• Matrise2 returnerer en matrise som er produktet av hvert par av samsvarende elementer i Matrise1 og Matrise2.

Uttr .• Matrise1 returnerer en matrise med produkter av Uttr og hvert element i Matrise1.

**. / (prikk divider)**

.	÷	taster
---	---	--------

Matrise1 ./ Matrise2 ⇒ matrise

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix}$	./	$\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$
--	----	--	--

Uttr ./ Matrise1 ⇒ matrise

$x \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$
--	--

Matrise1 ./ Matrise2 returnerer en matrise som er kvotient av hvert par av samsvarende elementer i Matrise1 og Matrise2.

Uttr ./ Matrise1 returnerer en matrise som er kvotienten av Uttr og hvert element i Matrise1.

**.^ (prikk potens)**

.	^	taster
---	---	--------

Matrise1 .^ Matrise2 ⇒ matrise

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix}$	.^	$\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$
--	----	--	---

Uttr .^ Matrise1 ⇒ matrise

$x \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$
--	--

Matrise1 .^ Matrise2 returnerer en matrise, der hvert element i Matrise2 er eksponenten for samsvarende element i Matrise1.

Uttr .^ Matrise1 returnerer en matrise, der hvert element i Matrise1 er eksponenten for Uttr.



## = (er lik)

 **tast**

Alt annet returnerer en forenklet form av ligningen.

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  $x \leq -5$  Then

Return 5

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

Return  $-x$

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

Return  $x$

ElseIf  $x = 10$  Then

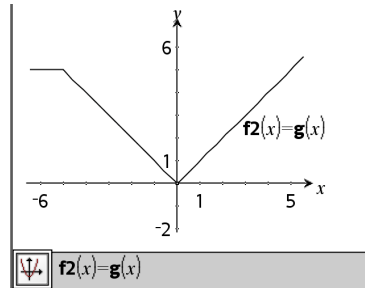
Return 3

EndIf

EndFunc

Done

Resultat av grafisk fremstilling  $g(x)$



## ≠ (ulik)

**ctrl**  **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**

*Uttr1!* ⇒ *Uttrykk*

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*]) ⇒ *uttrykk*

$\frac{d}{dx}(f(x) \cdot g(x))$   $\frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$

**d**(*Liste1*, *Var*[, *Orden*]) ⇒ *liste*

$\frac{d}{dy} \left( \frac{d}{dx}(x^2 \cdot y^3) \right)$   $6 \cdot y^2 \cdot x$

**d**(*Matrise1*, *Var*[, *Orden*]) ⇒ *matrise*

$\frac{d}{dx} \left( \begin{bmatrix} x^2 & x^3 & x^4 \end{bmatrix} \right)$   $\begin{bmatrix} 2 \cdot x & 3 \cdot x^2 & 4 \cdot x^3 \end{bmatrix}$

Returnerer den første deriverte av det første argumentet med hensyn på variabel *Var*.

*Orden*, hvis inkludert, må være et heltall. Hvis orden er mindre enn null, vil resultatet være en anti-derivert.

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

**d()** følger ikke normal behandlingsmekanisme når det gjelder å forenkle argumentene fullstendig og deretter bruke funksjonsdefinisjonen til disse fullstendig forenklede argumentene. I stedet utfører **d()** følgende trinn:

1. Forenkler det andre argumentet kun slik at det ikke fører til en ikke-

variabel.

- Forenkler det første argumentet kun slik at det ikke henter frem noen lagret verdi for variabelen som ble bestemt ved trinn 1.
- Bestemmer den symbolske deriverte av resultatet av trinn 2 med hensyn på variabelen fra trinn 1.

Hvis variabelen fra trinn 1 har en lagret verdi eller en verdi som er spesifisert av begrensingsoperator ("|"), settes denne verdien inn i resultatet fra trinn 3.

**Merk:** Se også Første deriverte, side 5; Andre deriverte, side 6; eller N-te deriverte, side 6.

## ∫() (integral)

∫(Uttr1, Var[, Nedre, Øvre]) ⇒ uttrykk

∫(Uttr1, Var[, Konstant]) ⇒ uttrykk

Returnerer integralet av *Uttr1* med hensyn på variabelen *Var* fra *Nedre* til *Øvre*.

**Merk:** Se også **Bestemt** eller **ubestemt integral-sjablon**, side 6.

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

Returnerer en antiderivert hvis *Nedre* og *Øvre* utelates. En symbolsk integrasjonskonstant utelates hvis ikke du oppgir argumentet *Konstant*.

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

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

$$\int (a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

Like, gyldige anti-deriverte kan variere med en numerisk konstant. Særlig kan en slik konstant være skjult, dersom en anti-derivert inneholder logaritmer eller inverse trigonometriske funksjoner. Dessuten blir noen ganger stykkevis konstante uttrykk lagt til for å gjøre en anti-derivert gyldig over et lengre intervall enn den vanlige formelen.

**∫()** returnerer seg selv for stykker av *Uttr1* som ikke kan bestemmes som en eksplisitt endelig kombinasjon av innebygde funksjoner og operatører.

Hvis du oppgir både *Nedre* og *Øvre*, gjøres det et forsøk på å finne eventuelle diskontinuiteter eller diskontinuerte deriverte i intervallet  $Nedre < Var < Øvre$ , og dele opp intervallet på de stedene.

For autoinnstilling av modusen **Auto eller Tilnærmet**, brukes numerisk integrasjon hvis mulig, dersom en anti-derivert eller en grense ikke kan bestemmes.

For innstilling i **Tilnærmet**, prøves numerisk integrasjon først, hvis mulig. Anti-derivert søkes bare der hvor slik numerisk integrasjon ikke er mulig eller ikke lykkes.

**∫()** kan nestes for å danne flere integraler. Integrasjonsgrensene kan avhenge av integrasjonsvariabler utenfor dem.

**Merk:** Se også **nInt()**, side 129.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2+a^2} dx \quad b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

**Merk:** For å tvinge frem et tilnærmet desimalresultat,

**Grafregner:** Trykk på **ctrl** **enter**.

**Windows®:** Trykk på **Ctrl+Enter**.

**Macintosh®:** Trykk på **⌘+Enter**.

**iPad®:** Hold på **enter**, og velg **≈**.

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

$$\int_0^a \int_0^x \ln(x+y) dy dx \quad \frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

$\sqrt{()}$  (kvadratrot)ctrl x<sup>2</sup> taster $\sqrt{(Uttr1)} \Rightarrow Uttrykk$ 

$$\sqrt{4} \qquad 2$$

 $\sqrt{(Liste1)} \Rightarrow liste$ 

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

Returnerer kvadratrotten til argumentet.

For en liste, returneres kvadratrotten til alle elementene i *Liste1*.**Merk:** Du kan sette inn denne funksjonen fra tastaturet ved å skrive **sqrt** (...)**Merk:** Se også **Kvadratrot-sjablon**, side 1. $\Pi()$  (prodSeq)

Katalog &gt;

 $\Pi(Uttr1, Var, Nedre, Øvre) \Rightarrow uttrykk$ 

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

**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.

$$\prod_{k=1}^n (k^2) \qquad (n!)^2$$

**Merk:** Se også **Produkt-sjablon** ( $\Pi$ ), side 5.

$$\prod_{n=1}^5 \left\{ \left\{ \frac{1}{n}, n, 2 \right\} \right\} \qquad \left\{ \frac{1}{120}, 120, 32 \right\}$$

 $\Pi(Uttr1, Var, Nedre, Nedre-1) \Rightarrow 1$ 

$$\prod_{k=4}^3 (k) \qquad 1$$

 $\Pi(Uttr1, Var, Nedre, Øvre) \Rightarrow 1/\Pi(Uttr1, Var, Øvre+1, Nedre-1)$  hvis  $Øvre < Nedre - 1$ 

Produktformlene som er brukt er hentet fra følgende referanse:

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \qquad 6$$

Ronald L. Graham, Donald E. Knuth, og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) \qquad \frac{1}{4}$$

**$\Sigma()$  (sumSeq)**Katalog >  $\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow \text{uttrykk}$ 

**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.

 $\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Nedre}-1) \Rightarrow 0$ 

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow -\Sigma(\text{Uttr1}, \text{Var}, \text{Øvre}+1, \text{Nedre}-1)$  hvis  $\text{Øvre} < \text{Nedre}-1$

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.

$$\sum_{n=1}^5 \left( \frac{1}{n} \right) \quad \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) \quad \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left( \frac{1}{n^2} \right) \quad \frac{\pi^2}{6}$$

$$\sum_{k=4}^3 (k) \quad 0$$

$$\sum_{k=4}^1 (k) \quad -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) \quad 4$$

 **$\Sigma\text{Int}()$** Katalog > 

$\Sigma\text{Int}(\text{NPmt1}, \text{NPmt2}, \text{N}, \text{I}, \text{PV}, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{avrundVerdi}]) \Rightarrow \text{verdi}$

 **$\Sigma\text{Int}$** 

$(\text{NPmt1}, \text{NPmt2}, \text{amortTabell}) \Rightarrow \text{verdi}$

Amortiseringsfunksjon som beregner rentesummen 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 205.

$$\Sigma\text{Int}(1, 3, 12, 4.75, 20000, 12, 12) \quad -213.48$$



- 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.

$\Sigma\text{Int}(NPmt1, NPmt2, amortTabell)$  beregner rentesummen basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under  $\text{amortTbl}()$ , side 8.

**Merk:** Se også  $\Sigma\text{Prn}()$ , nedenfor, og  $\text{Bal}()$ , side 17.

$tbl:=\text{amortTbl}(12,12,4.75,20000,,12,12)$			
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
$\Sigma\text{Int}(1,3,tbl)$			-213.48

 $\Sigma\text{Prn}()$ 

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [avrundVerdi]) \Rightarrow \text{verdi}$

$\Sigma\text{Prn}(1,3,12,4.75,20000,,12,12)$	-4916.28
--	----------

$\Sigma\text{Prn}(NPmt1, NPmt2, amortTable) \Rightarrow \text{verdi}$

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 205.

- 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

$tbl:=\text{amortTbl}(12,12,4.75,20000,,12,12)$			
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
$\Sigma\text{Prn}(1,3,tbl)$			-4916.28

*PmtAt* er de samme som for TVM-funksjonene.

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

ΣPrn(*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 8.

**Merk:** Se også ΣInt(), over, og Bal(), side 17.

## # (Indir.ref)

  **taster**

# *varNavnStreng*

#{"x"&"y"&"z"} xyz

Refererer til variabelen med navnet *varNavnStreng*. På denne måten kan du bruke strenger for å opprette variabelnavn "innenfra" en funksjon.

Oppretter eller refererer til variabelen xyz.

10 → *r* 10

"r" → *sI* "r"

#*sI* 10

Returnerer verdien av variabelen (*r*) som har et navn som er lagret i variabel *sI*.

## E (vitenskapelig tallnotasjon)

 **tast**

*mantissaEksponent*

23000. 23000.

Legger inn et tall i vitenskapelig fremstilling. Tallet blir tolket som en *mantissa* × 10<sup>eksponent</sup>.

2300000000.+4.1E15 4.1E15

3·10<sup>4</sup> 30000

Tips: Hvis du vil legge inn en potens av 10 uten å forårsake desimale verdier i resultatet, bruk 10<sup>heltall</sup>.

**Merk:** Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive **EE**. Eksempel: Skriv **2.3EE4** for å legge inn 2.3E4.

**g (gradian)****1** tast*UttrI*g ⇒ *Uttrykk*

I grader, gradian eller radian modus:

*UttrI*g ⇒ *Uttrykk*

$$\cos(50^g) \quad \frac{\sqrt{2}}{2}$$

*ListeI*g ⇒ *liste*

$$\cos(\{0,100^g,200^g\}) \quad \{1,0,-1\}$$

*MatriseI*g ⇒ *matrise*

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 *UttrI* med  $\pi/200$ .

I grader-vinkelmodus, multipliseres *UttrI* med  $g/100$ .

I gradian modus, returneres *UttrI* uendret.

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @g.

**r (radian)****1** tast*UttrI*r ⇒ *Uttrykk*

I grader, gradian eller radian modus:

*ListeI*r ⇒ *liste*

$$\cos\left(\frac{\pi}{4^r}\right) \quad \frac{\sqrt{2}}{2}$$

*MatriseI*r ⇒ *matrise*

$$\cos\left(\left\{0^r, \frac{\pi}{12}, \pi^r, -(\pi)^r\right\}\right) \quad \left\{1, \frac{(\sqrt{3+1})\sqrt{2}}{4}, -1\right\}$$

Denne funksjonen gir deg en mulighet til å spesifisere en gradian vinkel mens du er i grader- eller radian-modus.

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.

## r (radian)

1 tast

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @r.

## ° (grader)

1 tast

*Uttr*1°⇒*Uttrykk*

*Liste*1°⇒*liste*

*Matrise*1°⇒*matrise*

Denne funksjonen gir deg en mulighet til å spesifisere en grader-vinkel mens du er i gradian eller radian modus.

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.

I grader, gradian eller radian modus:

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

I radian modus:

**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 .

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \\ \{1., 0.707107, 0., 0.864976\}$$

## °, ', " (grader/minutter/sekunder)

taster

*gg*°*mm*'*ss*.*ss*"⇒*Uttrykk*

*gg* Et positivt eller negativt tall

*mm* Et ikke-negativt tall

*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 (").

I Grader-vinkelmodus:

$$25^\circ 13' 17.5'' \quad 25.2215 \\ 25^\circ 30' \quad \frac{51}{2}$$

## ∠ (vinkel)

  **taster**

[Radius,∠θ\_Vinkel]⇒vektor

(polar inndata)

[Radius,∠θ\_Vinkel, Z\_  
Koordinat]⇒vektor

(sylindrisk inndata)

[Radius,∠θ\_Vinkel,∠θ\_Vinkel]⇒vektor

(sfærisk inndata)

Returnerer koordinater som en vektor, avhengig av vektorformatets modusinnstilling: rektangulær, sylindrisk eller sfærisk.

**Merk:** Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @<.

(Størrelse ∠ Vinkel)⇒kompleksVerdi

(polar inndata)

Setter inn en kompleks verdi i ( $r∠\theta$ ) polar form. *Vinkelen* tolkes avhengig av aktuell vinkelmodus-innstilling.

I radian modus og vektorformat innstilt på: rektangulær

$$\left[ 5 \angle 60^\circ \angle 45^\circ \right] \left[ \frac{5\sqrt{2}}{4} \quad \frac{5\sqrt{6}}{4} \quad \frac{5\sqrt{2}}{2} \right]$$

sylindrisk

$$\left[ 5 \angle 60^\circ \angle 45^\circ \right] \left[ \frac{5\sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5\sqrt{2}}{2} \right]$$

sfærisk

$$\left[ 5 \angle 60^\circ \angle 45^\circ \right] \left[ 5 \angle \frac{\pi}{3} \angle \frac{\pi}{4} \right]$$

I radian-vinkelmodus og rektangulært, komplekst format:

$$5+3\cdot i \left( 10 \angle \frac{\pi}{4} \right) \quad 5-5\sqrt{2}+(3-5\sqrt{2})\cdot i$$

**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 .

$$5+3\cdot i \left( 10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107\cdot i$$

## ' (merke)

 **tast**

variabel '

variabel ''

$$\text{deSolve}\left(y''=y \frac{-1}{2} \text{ and } y(0)=0 \text{ and } y'(0)=0,t,y\right)$$
$$\frac{3}{2\cdot y^4}=t$$
$$\frac{3}{3}$$

## ' (merke)

 **tast**

Setter inn et derivertsymbol i en differensialligning. Et enkelt derivertsymbol markerer en førsteordens differensialligning, to derivertsymboler markerer en annenordens, osv.

## \_ (senket strek som et tomt element)

Se "Tomme (åpne) elementer" |, side 263.

## \_ (senket strek som enhetsbetegnelse)

  **taster**

### *Utrr\_Enhet*

Markerer enhetene for et *Utrr*. Alle enhetsnavnene må begynne med en senket strek.

Du kan bruke forhåndsdefinerte enheter eller opprette dine egne enheter. For en liste over forhåndsdefinerte enheter, åpne Katalogen og vis Enhetsomregnerfanen. Du kan velge enhetsnavn fra katalogen eller skrive inn enhetsnavnet direkte.


### *Variabel\_*

Hvis *Variabel* ikke har noen verdi, blir den behandlet som om den representerer et komplekst tall. Ved grunninnstilling, uten *\_*, behandles variabelen som reell.

Hvis *Variabel* har en verdi, ignoreres *\_* og *Variabel* gjenopptar opprinnelig datatype.

**Merk:** Du kan lagre et komplekst tall til en variabel uten å bruke *\_*. Men for best resultat i beregninger, som **cSolve()** og **cZeros()**, anbefales *\_*.

3·\_m▶\_ft 9.84252·\_ft

**Merk:** Du kan finne omregningssymbolet, ▶, i katalogen. Klikk på , og klikk så på **Matematiske operatører**.

Assuming z is undefined:

$\text{real}(z)$	$z$
$\text{real}(z\_)$	$\text{real}(z\_)$
$\text{imag}(z)$	0
$\text{imag}(z\_)$	$\text{imag}(z\_)$

## ► (omregne)

  **taster**

*Uttr\_Enhet1* ► *\_Enhet2* ⇒ *Uttr\_Enhet2*

3·\_m►\_ft

9.84252·\_ft

Omregner et uttrykk fra en enhet til en annen.

Senket strek-tegnet \_ markerer enheten. Enhetene må være i samme kategori, som f.eks. Lengde eller Areal.

For en liste over forhåndsdefinerte enheter, åpne Katalogen og vis Enhetsomregner-fanen:

- Du kan velge et enhetsnavn fra listen.
- Du kan velge omregningsoperatoren, ►, fra øverst på listen.

Du kan også skrive inn et navn manuelt. For å skrive inn “\_” mens du skriver inn enhetsnavn på grafregneren, trykk på

 .

**Merk:** For å omregne temperaturenheter, bruk **tmpCnv()** og **ΔtmpCnv()**. Omregningsoperatoren ► behandler ikke temperaturenheter.

## 10<sup>^</sup>( )

Katalog > 

**10<sup>^</sup>(Uttr1)** ⇒ *Uttrykk*

$10^{1.5}$  31.6228

**10<sup>^</sup>(Liste1)** ⇒ *liste*

$10^{\{0, -2.2, a\}}$   $\left\{1, \frac{1}{100}, 100, 10^a\right\}$

Returnerer 10 opphøyd i argumentets potens.

For en liste, returneres 10 opphøyd i elementenes potens i *Liste1*.

**10<sup>^</sup>(kvadratMatrise)** ⇒ *kvadratMatrise*

$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}}$

1.14336E7	8.17155E6	6.67589E6
9.95651E6	7.11587E6	5.81342E6
7.65298E6	5.46952E6	4.46845E6

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.

*Uttr1*  $\wedge^{-1} \Rightarrow$  Uttrykk

$$(3.1)^{-1} \quad 0.322581$$

*Liste1*  $\wedge^{-1} \Rightarrow$  liste

$$\{a, 4, 0.1, x, 2\}^{-1} \quad \left\{ \frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, \frac{-1}{2} \right\}$$

Returnerer den inverse verdien av et argument.

For en liste, returneres den inverse verdien av elementene i *Liste1*.*kvadratMatrise1*  $\wedge^{-1} \Rightarrow$  kvadratMatriseReturnerer den inverse verdien av *kvadratMatrise1*.*kvadratMatrise1* må være en ikke-singulær kvadratisk matrise.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ a-2 & a-2 \\ a & -1 \\ 2 \cdot (a-2) & 2 \cdot (a-2) \end{bmatrix}$$

**| (begrensningsoperator)**

-taster

*Uttr* | *BoolskUttr1* [and *BoolskUttr2*]...

$$x+1|x=3 \quad 4$$

*Uttr* | *BoolskUttr1* [or *BoolskUttr2*]...

$$x+y|x=\sin(y) \quad \sin(y)+y$$

$$x+y|\sin(y)=x \quad x+y$$

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 begrensingsoperatoren har du tre utgangstyper av funksjonalitet:

- Erstatninger
- 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*.

$$x^3-2 \cdot x+7 \rightarrow f(x) \quad Done$$

$$f(x)|x=\sqrt{3} \quad \sqrt{3}+7$$

$$(\sin(x))^2+2 \cdot \sin(x)-6|\sin(x)=d \quad d^2+2 \cdot d-6$$

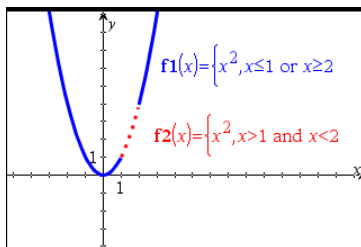


## | (begrenningsoperator)

ctrl  -taster

Intervallbegrensninger tar form som en eller flere ulikheter som er føyd sammen av logiske "and" eller "or" operatorer. En intervallbegrensning tillater også forenkling som ellers kan være ugyldig eller ikke mulig å beregne.

$\text{solve}(x^2-1=0,x) x>0 \text{ and } x<2$	$x=1$
$\sqrt{x} \cdot \sqrt{\frac{1}{x}} x>0$	1
$\sqrt{x} \cdot \sqrt{\frac{1}{x}}$	$\sqrt{\frac{1}{x}} \cdot \sqrt{x}$



Eksklusjoner bruker relasjons-operatoren "ulik" ( $\neq$  or  $\neq$ ) for å ekskludere en spesifikk verdi fra å komme i betraktning. Den brukes først og fremst for å ekskludere en eksakt løsning når du bruker `cSolve()`, `cZeros()`, `fMax()`, `fMin()`, `solve()`, `zeros()`, osv.

$\text{solve}(x^2-1=0,x) x \neq 1$	$x=-1$
------------------------------------	--------

## → (lagre)

ctrl var tast

Uttr → 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 Uttr, Liste, eller Matrise.

Hvis Var allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med Uttr, Liste, eller Matrise.

$\frac{\pi}{4} \rightarrow \text{myvar}$	$\frac{\pi}{4}$
$2 \cdot \cos(x) \rightarrow y1(x)$	Done
$\{1,2,3,4\} \rightarrow \text{lst5}$	$\{1,2,3,4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → str1	"Hello"

## → (lagre)

ctrl var

tast

Tips: Hvis du skal til å gjøre symbolske beregninger med udefinerte variabler, bør du unngå å lagre dem under de mest vanlig brukte en-bokstavs variabler, som a, b, c, x, y, z og så videre.

**Merk:** Du kan sette inn denne operatoren fra tastaturet ved å skrive =: som en snarvei. Eksempel: Skriv  $\pi/4$  =: `minvar`.

## := (tildele)

ctrl |⇧|⇧

taster

*Var* := *Uttr*

*Var* := *Liste*

*Var* := *Matrise*

*Funksjon*(*Param1*,...) := *Uttr*

*Funksjon*(*Param1*,...) := *Liste*

*Funksjon*(*Param1*,...) := *Matrise*

Hvis variabel *Var* ikke eksisterer, opprettes *Var* og initialiserer den til *Uttr*, *Liste*, eller *Matrise*.

Hvis *Var* allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med *Uttr*, *Liste*, eller *Matrise*.

Tips: Hvis du skal til å gjøre symbolske beregninger med udefinerte variabler, bør du unngå å lagre dem under de mest vanlig brukte en-bokstavs variabler, som a, b, c, x, y, z og så videre.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$y1(x) := 2 \cdot \cos(x)$	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)

  **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.

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

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

## Ob, Oh

  **taster, Ob, Oh taster**

### Ob *binærTall*

I desimalt grunntall-modus:

Ob10+OhF+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:

Ob10+OhF+10

Ob11011

Resultatene vises i forhold til grunntall-modusen.

I heksades grunntall-modus:

Ob10+OhF+10

Oh1B

# 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 kladdeark.
- 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

Grafikkskjermbildet vil inneholde en overskrift øverst på skjermen som det ikke kan skrives i av grafikkkommandoer.

Tegningsområdet for grafikkskjermbildet vil bli slettet (farge = 255, 255, 255) når grafikkskjermbildet initialiseres.

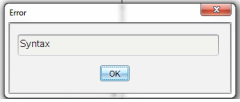
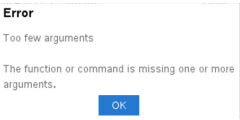
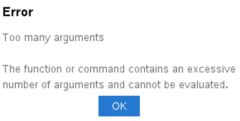
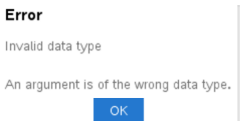
Grafikkskjerm	Standard
Høyde	212
Bredde	318
Farge	hvit: 255,255,255

## 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

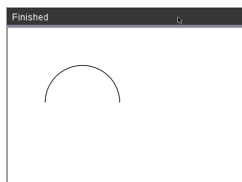
«arc angle» definerer vinkelåpningen for buen.

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

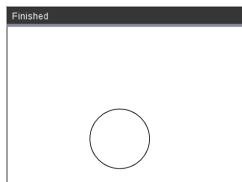
 Katalog >   
 CXII

**DrawCircle**  $x, y, \text{radius}$

$x, y$ : koordinat for sentrum

*radius*: radiusen på sirkelen

DrawCircle 150,150,40



Se også: [FillCircle](#)



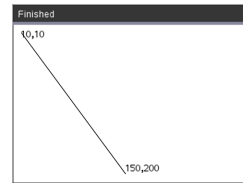
**DrawLine**  $x1, y1, x2, y2$

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

$xlist, ylist$

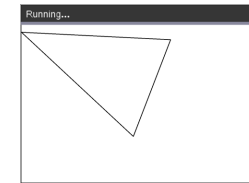
Uttrykk som evaluerer til ett enkelt reelt flyttall

$x1, y1...xn, yn$  = koordinater for polygohjørnene

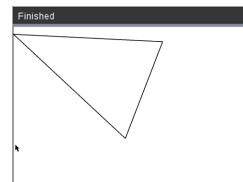
$xlist:={0,200,150,0}$

$ylist:={10,20,150,10}$

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*

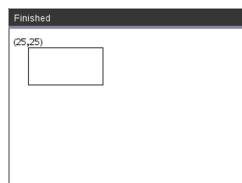
*x, y*: øvre venstre koordinat for rektangel

*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

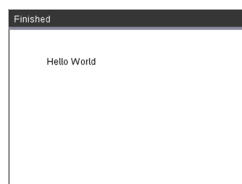
**DrawText** *x, y, exprOrString1*  
*[,exprOrString2]...*

*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

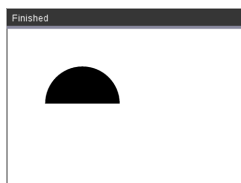
$x, y$ : øvre venstre koordinat for avgrensede rektangel

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](#)-kommando

«arc angle» definerer vinkelåpningen for buen

FillArc 50,50,100,100,0,180



## FillCircle

 Katalog >  CXII

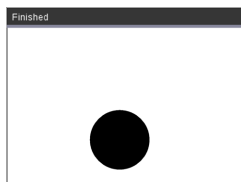
**FillCircle**  $x, y$ , radius

$x, y$ : 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](#)-kommando.

FillCircle 150,150,40



Her!

## FillPoly

 Katalog >  CXII

**FillPoly**  $xlist, ylist$

eller

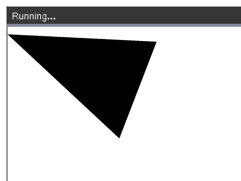
**FillPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$

**Merk:** Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

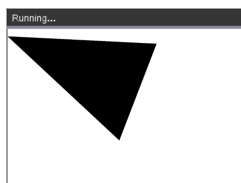
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

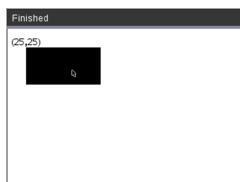
FillPoly  $xlist,ylist$



FillPoly 0,10,200,20,150,150,0,10

**FillRect****FillRect**  $x, y, \text{bredde}, \text{høyde}$  $x, y$ : øvre venstre koordinat for rektangel $\text{bredde}, \text{høyde}$ : 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

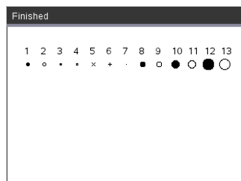
PlotXY 100,100,1

For  $n, 1, 13$ 

DrawText 1+22\*n,40,n

PlotXY 5+22\*n,50,n

EndFor



**SetColor**
 Katalog >   
**CXII**
**SetColor**

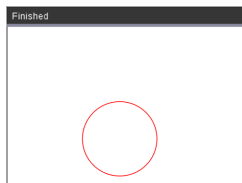
Red-value, Green-value, Blue-value

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

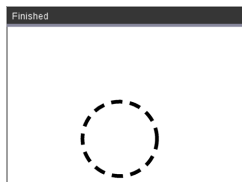
tykkelse: 1 &lt;= tykkelse &lt;= 3 | 1 er tynneste, 3 er tykkeste

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

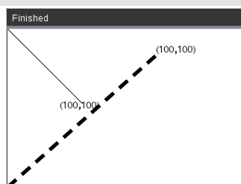


Hvis  $x_{\min}$  er større enn eller lik  $x_{\max}$  eller  $y_{\min}$  er større enn eller lik  $y_{\max}$ , vises en feilmelding.

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™ CAS 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 52, og **isVoid()**, side 100.

**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**, **sumIf**, **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>
$d()$	<code>derivative (...)</code>
$\int()$	<code>integral (...)</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>
$\Delta\text{tmpCnv}()$	<code>deltaTmpCnv (...)</code>

## Fra tastaturet på datamaskinen

Hvis du skal legge inn dette:	Skriv dette:
$c1, c2, \dots$ (konstanter)	<code>@c1, @c2, ...</code>
$n1, n2, \dots$	<code>@n1, @n2, ...</code>

Hvis du skal legge inn dette:	Skriv dette:
(heltallskonstanter)	
$i$ (imaginær konstant)	@i
$e$ (naturlig log-grunntall $e$ )	@e
$E$ (vitenskapelig notasjon)	@E
$\top$ (transponert)	@t
$^{\circ}$ (radianer)	@r
$^{\circ}$ (grader)	@d
$^{\circ}$ (gadianer)	@g
$\sphericalangle$ (vinkel)	@<
► (konvertering)	@>
►Decimal, ►approxFraction ( ), osv.	@>Decimal, @>approxFraction(), osv.

# EOS™ (Ligningsoperativsystem)-hierarkiet

Dette avsnitt beskriver Equation Operating System (ligningsoperativsystem) (EOS™) som brukes av TI-Nspire™ CAS -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> , ' , " ), fakultet (!), 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™ CAS -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 halveis 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  $4096$ .

## Ekspensiering

Ekspensiering (^) og element-for-element-ekspensiering (.^ ) blir behandlet fra høyre til venstre. Uttrykket  $2^3^2$  blir for eksempel behandlet som det samme som  $2^{(3^2)}$  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 ekspensiering 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

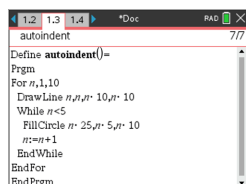
## Auto-innrykk i Programmeringseditor

TI-Nspire™ programeditor 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 åpningskommandoen.

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

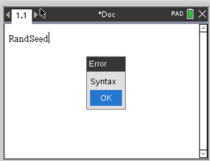
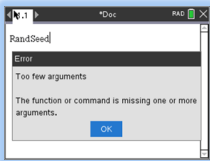
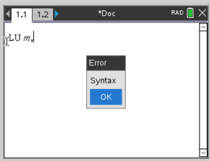
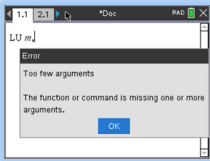
### Feil

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

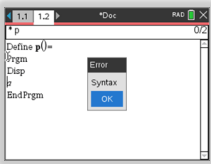
Feiltilstand	Ny melding
Mangler <b>EndTry</b>	Forventet <b>EndTry</b> , men fant en annen end-setning
« <b>Then</b> » utelatt etter <b>If</b> <condition>	Mangler <b>If..Then</b>
« <b>Then</b> » utelatt etter <b>Elseif</b> <condition>	<b>Then</b> mangler i blokken: <b>Elseif</b> .
Når « <b>Then</b> », « <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

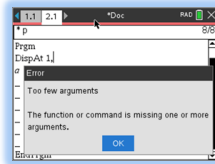
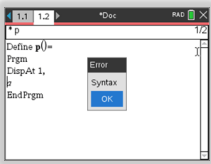
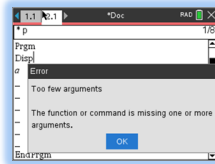
Hvis kommandoer som forventer ett eller flere argumenter blir oppkalt med en ufullstendig liste over argumenter, vil en «**For få argumenter-feil**» vises istedenfor for «**syntaks**»-feil

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>

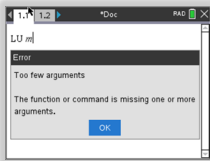
## Gjeldende atferd



## Ny CX II-atferd



**Merk:** Når en ufullstendig liste over argumenter ikke etterfølges av et komma, er feilmeldingen: «for få argumenter». Dette er det samme som i tidligere versjoner.



## 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 201.

**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 $If\ a < b$ vil for eksempel forårsake enten at $a$ eller at $b$ ikke er definert, dersom utsagnet $If$ 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.

<b>Feilkode</b>	<b>Beskrivelse</b>
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>• sto →</li> <li>• :=</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>BibOff</code> eller <code>BibPriv</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 opplyser hver numeriske varselkode og dens assosierte melding. Se **warnCodes()** for et eksempel på lagring av advarselskoder, side 210.

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

		–	
		_, enhetsmarkering .....	242
' , fremstilling minutter .....	240	, begrensingsoperator .....	244
' , prime .....	241		
		+	
– , subtrahere[*] .....	221	+ , addere .....	221
		/	
! , faktultet .....	232	/ , dividere[*] .....	223
		=	
" , fremstilling sekunder .....	240	≠ , ulik[*] .....	228
		= , er lik .....	227
#		>	
# , Indir.ref .....	238	> , større enn .....	230
# , omregnings-operator .....	268		
%		∏	
% , prosent .....	227	∏ , produkt, sjablon for .....	5
		∏ , produkt[*] .....	235
&		∑	
& , legg til .....	232	∑ ( ) , sum[*] .....	236
		∑Int ( ) .....	236
*		∑Prn ( ) .....	237
* , multiplisere .....	222	√	
.		√ , kvadratrots[*] .....	235
– , prikk subtraksjon .....	225	∫	
· , prikk multiplikasjon .....	226	∫ , integral[*] .....	233
./ , prikk divisjon .....	226		
^ , prikk potens .....	226	≤	
·+ , prikk addisjon .....	225	≤ , mindre enn eller lik .....	229
:		≥	
:= , tildele .....	246	≥ , større enn eller lik med .....	230
^			
^ , potens .....	224		



►	
►Cylind, vise som sylindrisk vektor [Sylind] .....	45
►Polar, vise som polar vektor[Polar]	140
►, omregne enheter[*] .....	243
►, omregner til gradian vinkel[Grad]	92
►approxFraction( ) .....	14
►cos, vise uttrykt ved cosinus[cos] ..	31
►DD, vises som desimalvinkel[DD] ..	48
►Desimal, vise resultat som desimal [Decimal] .....	48
►DMS, vise som grader/minutter/sekunder [DMS (GMS)] .....	58
►exp, vis uttrykt ved e[exp] .....	68
►Grunntall10, vise som desimalt heltall[Grunntall10] .....	20
►Grunntall16, vise som heksadesimal[Grunntall16]	20
►Grunntall2, vise som binær [Grunntall2] .....	18
►Rad, omregne til radian vinkel .....	150
►Rect, vise som rektangulær vektor	154
►sin, vise uttrykt ved sinus[sin] .....	176
►Sphere, vise som sfærisk vektor [Sfære (kule)] .....	185
→	
→, lagre .....	245
⇒	
⇒, logisk implikasjon[*] .....	231, 265
↔	
↔, logisk dobbel implikasjon[*] .....	231
©	
©, kommentar .....	247
°	
°, grader fremstilling[*] .....	240
°, grader/minutter/sekunder[*] .....	240

	<b>0</b>	
Ob, binær indikator .....		247
Oh, heksadesimal indikator .....		247
	<b>1</b>	
10^( ), tier-potens .....		243
	<b>2</b>	
2-delers stykkevis funksjon sjablon for .....		2
2-utvalg F test .....		80
	<b>A</b>	
abs( ), absoluttverdi .....		8
Absoluttverdi sjablon for .....		4
addere, + .....		221
amortiseringsstabell, amortTbl( ) .....		8, 17
amortTbl( ), amortiseringstabell .....		8, 17
and, Boolsk operator .....		9
andrederivert sjablon for .....		6
angle( ), vinkel .....		10
ANOVA, enveis varians-analyse .....		11
ANOVA2-veis, toveis varians-analyse		11
Ans, siste svar .....		13
antall betingede elementer i en liste , tellIf( ) .....		37
antall elementer i en liste, antall( ) ..		37
antall( ), antall elementer i en liste ..		37
åpne (tomme) elementer .....		263
åpne elementer, fjern .....		52
approx( ), tilnærmet .....		14
approxRational( ) .....		14
arccos() .....		15
arccosh() .....		15
arccot() .....		15
arccoth() .....		15
arccsc() .....		15
arccsch() .....		15
arcLen( ), buelengde .....		15
arcsec() .....		15
arcsech() .....		16
arcsin() .....		16
arcsinh() .....		16
arctan() .....		16

arctanh()	16	char(), tegnstring	24
argumenter i TVM-funksjoner	205	charPoly()	24
augment(), utvid/sett sammen	16	ClearAZ	26
avgRC(), gjennomsnittlig		colDim(), matrisens	
endringshastighet	17	kolonnedimensjon	27
avrund, round()	163	colNorm(), matrisens kolonnenorm	27
avslutt		comDenom(), fellesnevner	28
funksjon, EndFunc	81	completeSquare(), complete square	29
avslutt, Exit	67	conj(), kompleks konjugert	30
<b>B</b>			
begrensningsoperator " "	244	constructMat(), konstruer matrise	30
begrensningsoperator, rekkefølge av		cos <sup>-1</sup> , invers cosinus	33
beregning	267	cos(), cosinus	32
behandle polynom, polyEval()	141	cosh <sup>-1</sup> (), hyperbolsk, invers cosinus	35
behandling, rekkefølge av	267	cosh(), hyperbolsk cosinus	34
bestemt integral		cosinus	
sjablon for	6	vise uttrykk med hensyn på	31
bibliotek		cosinus, cos()	32
lage snarveier til objekter	101	cot <sup>-1</sup> (), invers cotangens	36
Bibloff	50	cot(), cotangens	35
BibPriv	50	cotangens, cot()	35
binær		coth <sup>-1</sup> (), hyperbolsk invers	
indikator, Ob	247	cotangens	37
vise, ►Grunntall2	18	coth(), hyperbolsk cotangens	36
binomCdf()	21, 97	cPolyRoots()	38
binomPdf()	21	crossP(), kryssprodukt	39
Boolske operatører		csc <sup>-1</sup> (), invers cosekans	39
⇒	231, 265	csc(), cosekans	39
⇔	231	csch <sup>-1</sup> (), invers hyperbolsk cosekans	40
and	9	csch(), hyperbolsk cosekans	40
eller	136	cSolve(), komplekse løs	40
enten ... eller ...	211	CubicReg, kubisk regresjon	43
ikke	132	cumulativeSum(), kumulativ sum	44
ikke både...og	126	cycle, Løkke	44
verken ... eller	130	Cycle, løkke	44
brøker		cZeros(), komplekse nullpunkt	45
ekteBrøk	146	<b>D</b>	
sjablon for	1	d(), første deriverte	232
buelengde, arcLen()	15	dager mellom datoer, dbd()	47
<b>C</b>			
c 2 2-veis	24	dbd(), dager mellom datoer	47
c 2 Pdf()	26	Define, definer	49
Cdf()	74	Definer	49
ceiling(), øvre	21	Definer Bibloff	50
centralDiff()	22	Definer BibPriv	50
cFactor(), kompleks faktor	22	definer, Define	49
		definere	
		felles (offentlig) funksjon eller	
		program	50

privat funksjon eller program ..	50	eksponensiell regresjon, ExpReg .....	71
deltaList() .....	51	eksponent, E .....	238
deltaTmpCnv() .....	51	eksponenter	
DelVar, slett variabel .....	51	sjablon for .....	1
delVoid( ), fjern gamle elementer ...	52	ekte brøk, propFrac .....	146
derivative() .....	52	eliminasjonsform, ref( ) .....	155
deriverte		eller (Boolsk), eller .....	136
første deriverte, d( ) .....	232	eller, Boolsk operator .....	136
numerisk derivert, nDeriv( ) .....	128-129	else if, ElseIf .....	63
numerisk derivert, nDerivative( ) .....	127	else, Else .....	92
deriverte eller n-te deriverte		Elseif, else if .....	63
sjablon for .....	6	en-variabel-statistikk, OneVar .....	134
desimal		end	
heltall vise, 4Grunntall10 .....	20	For...EndFor .....	77
vinkel-visning, ►DD .....	48	if, EndIf .....	92
deSolve( ), løsning .....	52	stigningstall, EndLoop .....	116
det( ), matrisedeterminant .....	55	while, EndWhile .....	211
diag( ), matrisediagonal .....	55	end if, EndIf .....	92
dim( ), dimensjon .....	56	end stigningstall, EndLoop .....	116
dimensjon, dim( ) .....	56	end while, EndWhile .....	211
DispAt .....	56	endfunksjon, EndFunc .....	81
dividere heltall, intDiv( ) .....	96	Endret internrente av retur, mirr( ),	121
dividere, / .....	223	EndTry, avslutt prøv .....	201
dominerende ledd( ), dominerende		EndWhile, end while .....	211
ledd .....	59	enhetsvektor, unitV( ) .....	207
dominerende ledd, dominerende		enten ... eller ..., Boolsk eksklusiv	
ledd( ) .....	59	eller .....	211
dotP( ), prikk produkt .....	61	EOS (Equation Operating System) ..	267
<b>E</b>			
e eksponent		Equation Operating System	
sjablon for .....	2	(Ligningsoperativsystem)	
e i en potens, e^( ) .....	61, 68	(EOS) .....	267
E, eksponent .....	238	er lik, = .....	227
e, vise uttrykk uttrykt ved .....	68	erstatning med " " operator .....	244
e^( ), e i en potens .....	61	etikettNavn, Lbl .....	100
eff( ), omregn nominell til effektiv		euler( ), Euler function .....	64
rente .....	62	exact( ), eksakt .....	67
effektiv rente, eff( ) .....	62	Exit, avslutt .....	67
egendefinerte funksjoner .....	49	exp( ), e i en potens .....	68
egendefinerte funksjoner og		exp►liste( ), uttrykk til liste .....	69
programmer .....	50	expand( ), utvid .....	69
eigenvektor, eigVc( ) .....	62	expr( ), streng til uttrykk .....	113
eigenverdi, eigVl( ) .....	63	expr( ), streng til uttrykk .....	70
eigVc( ), eigenvektor .....	62	ExpReg, eksponensiell regresjon ...	71
eigVl( ), eigenverdi .....	63	<b>F</b>	
eksakt, exact( ) .....	67	factor( ), faktor .....	72
ekskluderer med " " operator .....	244	faktor, factor( ) .....	72
		fakultet, ! .....	232

feil og problemløsning		grader/minutter/sekunder	
send feil, SendFeil	138	fremstilling minutter,	240
slett feil, SlettFeil	26	fremstilling sekunder, "	240
feilkoder og meldinger	281	freqTable( )	79
fellesnevner, comDenom( )	28	Frobenius-norm, norm( )	131
Fill, matrise fyll	74	Func, funksjon	81
finansielle funksjoner, tvnFV( )	204	Func, programfunksjon	81
finansielle funksjoner, tvnI( )	204	funksjoner	
finansielle funksjoner, tvnN( )	204	del, fpart( )	78
finansielle funksjoner, tvnPmt( )	204	egendefinere	49
finansielle funksjoner, tvnPV( )	205	maksimalpunkter for, fMax( )	76
FiveNumSummary	74	minimalpunkter for, fMin( )	76
fjerdegrads regresjon, QuartReg	148	programfunksjon, Func	81
fjern		funksjoner og variabler	
åpne elementer fra liste	52	kopiere	30
floor( ), nedre	75	fyll	255-256
fMax( ), maksimalpunkt for			
funksjonen	76	<b>G</b>	
fMin( ), minimalpunkt for funksjonen	76	G, gradianer	239
For	77	gå til, Goto	91
for, For	77	gcd( ), største felles divisor	81
For, for	77	geomCdf( )	82
fordelingsfunksjoner		geomPdf( )	82
binomCdf( )	21, 97	getDenom( ), lesNevner	84
binomPdf( )	21	getKey()	84
c 2 Pdf( )	26	getLangInfo( ), hent/returner	
c22-veis( )	24	språkinformasjon	87
invNorm( )	98	getLockInfo( ), tester låsestatus av	
invt( )	98	variabel eller	
normCdf( )	131	variabelgruppe	88
normPdf( )	131	getNum( ), les/returner teller	89
poissCdf( )	139	GetStr	89
poissPdf( )	139	getType( ), get type of variable	90
tCdf( )	196	getVarInfo( ), les/returner	
tPdf( )	200	variabelinformasjon	90
$\chi^2$ Cdf( )	25	gjennomsnitt, mean( )	118
$\chi^2$ GOF( )	25	gjennomsnittlig endringshastighet,	
fordelingssfunksjoner		avgRC( )	17
Inv $\chi^2$ ( )	96	Goto, gå til	91
Forespør	157	grader fremstilling, °	240
ForespørStr	159	grader/minutter/sekunder-visning,	
format( ), formatstreng	77	►DMS	58
formatstreng, format( )	77	gadian fremstilling, G	239
første deriverte		grafregnere	
sjablon for	5	omregne	243
fortegn, sign( )	174	grupper, låse og låse opp	112, 208
fpart( ), funksjonsdel	78	grupper, teste låsestatus	88
frekvens( )	79		
fremstilling i	240		

<b>H</b>		
heksadesimal		
indikator, Oh	247	
vise, 4Grunntall16	20	
heltall, int()	95	
heltallsdel, iPart()	98	
Hent	82, 257	
hent/returner		
variabelinformasjon, getVarInfo()	87, 90	
høyre(), høyre	160	
høyre, høyre()	160	
høyre, right()	96	
hurtigtaster	265	
hvis, Hvis	92	
hyperbolic (hyperbolsk)		
invers tangens, tanh <sup>-1</sup> ()	195	
hyperbolsk		
cosinus, cosh()	34	
invers cosinus, cosh <sup>-1</sup> ()	35	
invers sinus, sinh <sup>-1</sup> ()	179	
sinus, sinh()	178	
tangens, tanh()	194	
<b>I</b>		
identitet(), identitetsmatrise	92	
identitetsmatrise, identitet()	92	
If, if	92	
iffn()	93	
ikke både...og, Boolsk operator	126	
ikke, Boolsk operator	132	
imag(), imaginær del	94	
imaginær del, imag()	94	
ImpDif(), implisitt derivert	95	
implisitt derivert, Impdif()	95	
Indir.ref, #	238	
innenfor streng, inString()	95	
innstillinger, les aktuell	88	
Input, inndata	95	
inString(), innenfor streng	95	
int(), heltall	95	
intDiv(), dividere heltall	96	
integral, ∫	233	
interpolere(), interpolere	96	
invers cosinus, cos <sup>-1</sup> ()	33	
invers kumulativ normalfordeling		
(invNorm())	98	
invers sinus, sin / ()	177	
invers tangens, tan <sup>-1</sup> ()	193	
invers, x <sup>-1</sup>	244	
invF()	97	
invNorm(), invers kumulativ normalfordeling	98	
invf()	98	
Invχ <sup>2</sup> ()	96	
iPart(), heltallsdel	98	
irr(), internrente		
internrente, irr()	99	
isPrime(), primtest	99	
isVoid(), test for tomrom	100	
<b>K</b>		
kolUtvid	27	
kombinasjoner, nCr()	127	
Kommandoen Wait	209	
kommentar, ©	247	
kompleks		
faktor, cFactor()	22	
konjugert, conj()	30	
løs, cSolve()	40	
nullpunkt, cZeros()	45	
konstant		
i løs()	182	
konstanter		
hurtigtaster for	265	
i dLLøs()	53	
i kLøs()	42	
i kNullp()	46	
i løs()	183	
konstruer matrise, constructMat()	30	
kopiere variabel eller funksjoner, CopyVar	30	
korrelasjonsmatrise, corrMat()	31	
korrMat(), korrelasjonsmatrise	31	
kryssprodukt, crossP()	39	
kubisk regresjon, CubicReg	43	
kumulativ sum, cumulativeSum()	44	
kvadratisk regresjon, QuadReg	147	
kvadratrot		
sjablon for	1	
kvadratrot, √()	235	
kvadratrot, ∓()	186	



matriser		linjeregresjon	120
determinant, det( )	55	mid( ), midtstreng	120
diagonal, diag( )	55	midtstreng, mid( )	121
dimensjon, dim( )	56	min( ), minimum	229
egenvektor, eigVc( )	62	mindre enn eller lik, {	121
egenverdi, eigVl( )	63	minimum, min( )	100
eliminasjonsform, ref( )	155	minste felles multiplum, lcm	121
fill, Fylle	74	mirr( ), endret internrente av retur	122
identitet, identitet( )	92	mod( ), modul	122
kolonnedimensjoner, colDim( )	27	modul, mod( )	88
kolonnenorm, colNorm( )	27	modus-innstillinger, lesModus( )	171
kumulativ sum, cumulativeSum( )	44	moduser	123
liste til matrise, listMat( )	109	lesing, lesModus( )	123
maksimum, max( )	117	mRow( ), matrise radhandling	123
matrise til liste, matAListe( )	117	mRowAdd( ), matrise	
minimum, min( )	121	radmultiplikasjon og	
nedre-øvre dekomposisjon, LU (lower-upper)	116	addisjon	123
nye, newMat( )	128	multiplisere, *	222
prikk addisjon, .+	225	Multipel lineær regresjon ttest	124
prikk divisjon, .P	226	MultReg	123
prikk multiplikasjon, .*	226	MultRegIntervals( )	123
prikk potens, .^	226	(MultRegIntervaller)	123
prikk subtraksjon, .N	225	MultRegTests( ) (MultRegTester)	124
produkt, product( )	145		
QR faktorisering, QR	146	<b>N</b>	
radaddisjon, rowAdd( )	164	n-te rot	
raddimensjon, rowDim( )	164	sjablon for	1
radhandling, mRow( )	123	når, when( )	210
radmultiplikasjon og addisjon, mRowAdd( )	123	naturlig logaritme, ln( )	109
radnorm, rowNorm( )	164	nCr( ), kombinasjoner	127
radskift, rowSwap( )	165	nDerivative( ), numerisk derivert	127
redusert eliminasjonsform, rref( )	165	nedre, floor( )	75
summering, sum( )	190-191	negasjon, legge inn negative tall	268
tilfeldig, tilfMat( )	152	netto nåverdi, npv( )	133
transponert, T	192	nevner	28
undermatrise, subMat( )	190	newList( ), ny liste	128
utvid/sett sammen, augment( )	16	newMat( ), ny matrise	128
max( ), maksimum	117	nfMax( ), numerisk	
mean( ), gjennomsnitt	118	funksjonsmaksimum	128
med (gitt at),	244	nfMin( ), numerisk	
median-median linjeregresjon, MedMed	119	funksjonsminimum	129
median( ), median	118	nInt( ), numerisk integral	129
median, median( )	118	nom( ), omregn effektiv til nominell rente	130
MedMed, median-median	119	nominell rente, nom( )	130
		norm( ), Frobenius-norm	131
		normal-linje, normalLine( )	131
		normalLine( )	131

normCdf ( )	131	polyGrader ( )	141
normPdf ( )	131	polyKoeff ( )	140
nPr ( ), permutasjoner	132	polynomer	
npv ( ), netto nåverdi	133	behandle, polyEval ( )	141
nSolve ( ), numerisk løsning	134	tilfeldig, tilfPoly ( )	153
nullpunkt, zeroes ( )	213	PolyRoots ( )	143
numerisk		potens, ^	224
derivert, nDeriv ( )	128-129	potensregresjon,	
derivert, nDerivative ( )	127	PowerReg	143, 157, 159, 197
integral, nInt ( )	129	PowerReg, potensregresjon	143
løsning, nSolve ( )	134	Prgm, definer program	144
ny		prikkP	
liste, newList ( )	128	addere, +	225
matrise, newMat ( )	128	divisjon, /	226
		multiplikasjon, *	226
<b>O</b>		potens, ^	226
objekter		produkt, dotP ( )	61
lage snarveier til bibliotek	101	subtraksjon, -	225
område ( ), områdefunksjon	59	prime, .....	241
områdefunksjon, område ( )	59	primtallstest, isPrime ( )	99
omregne		prodSeq ( )	145
►Rad	150	product ( ), produkt	145
4Grad	92	produkt, P ( )	235
grafregnere	243	produkt, product ( )	145
omregnings-operator (#)	268	programmer	
OneVar, en-variabel-statistikk	134	definere felles (offentlig)	
operatører		bibliotek	50
rekkefølge av behandling	267	definere privat bibliotek	50
ord ( ), numerisk tegnkode	137	programmer og programmering	
Overfører øyeblikkelig kontroll til		slett feil, SlettFeil	26
den neste iterasjonen i		try, Try	201
aktuell løkke (For, While,		vis I/O-skjerm, Vis	167
eller Loop).	45	vis I/O skjerm, Vis	56
øvre, ceiling ( )	21-22, 38	programmering	
		definer program, Prgm	144
<b>P</b>		send feil, SendFeil	138
P►Rx ( ), rektangulær x-koordinat	137	vis data, Vis	56, 167
P►Ry ( ), rektangulær y-koordinat	138	propFrac, ekte brøk	146
Pdf ( )	78	prosent, %	227
permutasjoner, nPr ( )	132	<b>Q</b>	
poissCdf ( )	139	QR faktorisering, QR	146
poissPdf ( )	139	QR, QR faktorisering	146
polar		QuadReg, kvadratisk regresjon	147
koordinat, R►Pr ( )	150	QuartReg, fjerdegrads regresjon	148
koordinat, R►Pθ ( )	150	<b>R</b>	
vektor-visning, ►Polar	140	R, radian	239
polyEval ( ), behandle polynom	141		
polyGcd ( )	142		



R►Pr( ), polarkoordinat .....	150	rowNorm(), radnorm i matrise .....	164
R►Pθ( ), polarkoordinat .....	150	rowSwap(), radskift i matrise .....	165
radian, R .....	239	rref( ), redusert eliminasjonsform ..	165
RandSeed, tilfeldig startverdi .....	153		
real( ), reell .....	153	<b>S</b>	
redusert eliminasjonsform, rref( ) ..	165	sannsynlig normalfordeling,	
reell, real( ) .....	153	normCdf( ) .....	131
ref( ), eliminasjonsform .....	155	sannsynlig student- t -fordeling, tCdf	
RefreshProbeVars .....	156	( ) .....	196
regresjon		sannsynlighetstetthet for student- t,	
fjerdegrads, QuartReg .....	148	tPdf() .....	200
kvadratisk, QuadReg .....	147	sannsynlighetstetthet for student-t,	
lineær regresjon, LinRegAx .....	104	tPdf( ) .....	200
lineær regresjon, LinRegBx .....	103	sannsynlighetstetthet, normPdf( ) ..	131
logaritmisk, LnReg .....	110	sec <sup>-1</sup> ( ), invers sekans .....	166
Logistisk .....	113	sec( ), sekans .....	165
logistisk, Logistic .....	114	sech <sup>-1</sup> ( ), invers hyperbolsk sekans ..	166
median-median-linje, MedMed .....	119	sech( ), hyperbolsk secant .....	166
MultReg .....	123	sekvens, seq( ) .....	168
potensregresjon, PowerReg .....	143	send feil, SendFeil .....	138
sinus, SinReg .....	179	SendFeil, send feil .....	138
Regresjon		senket strek, _ .....	242
lineær regresjon, LinRegBx .....	105	seq( ), sekvens .....	168
regresjoner		seqGen( ) .....	168
eksponensiell, ExpReg .....	71	seqn( ) .....	169
kubisk, CubicReg .....	43	sequence, seq( ) .....	168-169
potensregresjon,		sfærisk vektor-visning, ►Sphere .....	185
PowerReg ... 143, 157, 159, 197		shift( ), skift .....	173
rekke( ), rekke .....	170	sign( ), fortegn .....	174
rekke, rekke( ) .....	170	simult( ), simultane ligninger .....	175
rektangulær vektor-visning, ►Rect ..	154	simultane ligninger, simult( ) .....	175
rektangulær x-koordinat, P►Rx( ) ...	137	sin <sup>-1</sup> ( ), invers sinus .....	177
remain( ), rest .....	157	sin( ), sinus .....	177
resiprok, x <sup>-1</sup> .....	244	sinh <sup>-1</sup> ( ), hyperbolsk invers sinus .....	179
rest, remain( ) .....	157	sinh( ), hyperbolsk sinus .....	178
resultat		SinReg, sinusregresjon .....	179
vise uttrykt ved cosinus .....	31	sinus	
vise uttrykt ved e .....	68	vise uttrykk uttrykt ved .....	176
vise uttrykt ved sinus .....	176	sinus, sin( ) .....	177
resultat verdier, statistikker .....	188	sinusregresjon, SinReg .....	179
resultater, statistikk .....	186	sjabloner	
Return, retur .....	160	(Π), produkt .....	5
right, right( ) .....	29, 64, 210	2-delers stykkevis funksjon .....	2
rk23( ), Runge Kutta-funksjon .....	160	Absoluttverdi .....	4
rotate( ), rotere .....	162	andrederivert .....	6
rotere, rotate( ) .....	162	bestemt integral .....	6
round( ), avrund .....	163	brøk .....	1
rowAdd( ), radaddisjon i matrise .....	164	deriverte eller n-te deriverte ...	6
rowDim(), raddimensjon i matrise ..	164		

e eksponent .....	2	tilfeldig norm, tilfNorm( ) .....	152
eksponent .....	1	to-variable resultater, TwoVar ..	205
første deriverte .....	5	varians, variance( ) .....	209
kvadratrot .....	1	stdDevPop( ), populasjonen	
ligningssystemer (2-ligning) ....	3	standardavvik .....	188
ligningssystemer (N-ligning) ....	3	stdDevSamp( ), utvalgets	
lim .....	7	standardavvik .....	189
Logaritme .....	2	stigningstall, Loop .....	116
matrise (1 × 2) .....	4	Stoppkommando .....	189
matrise (2 × 1) .....	4	større enn eller lik med,   .....	230
matrise (2 × 2) .....	4	større enn, > .....	230
matrise (m × n) .....	4	største felles divisor, gcd( ) .....	81
n-te rot .....	1	streng	
stykkevis funksjon (N-delers) ...	3	dimensjon, dim( ) .....	56
sum (G) .....	5	lengde .....	56
ubestemt integral .....	6	strenger	
skift, shift( ) .....	173	brukes for å opprette	
slå sammen trigonometrisk, tCollect( ) .....	196	variabelnavn .....	268
slett		format, format( ) .....	77
åpne elementer fra liste .....	52	formatering .....	77
feil, SlettFeil .....	26	høyre, høyre( ) .....	96, 160
Slett .....	251	Indir.ref, # .....	238
slettAZ .....	26	innenfor, inString .....	95
slette		legg til, & .....	232
variabel, DelVar .....	51	midtstreng, mid( ) .....	120
SlettFeil, slett feil .....	26	rotene, rotate( ) .....	162
snarveier, tastatur .....	265	skift, shift( ) .....	173
solve( ), løs .....	180	streng til uttrykk, expr( ) .....	70, 113
SortA, sorter stigende .....	184	tegnkode, ord( ) .....	137
SortD, sorter fallende .....	185	tegnstreng, char( ) .....	24
sorterer		uttrykk til streng, string( ) .....	190
fallende, SortD .....	185	venstre, left( ) .....	101
stigende, SortA .....	184	string( ), uttrykk til streng .....	190
språk		strings	
hente språkinformasjon .....	87	right, right( ) .....	29, 64, 210
sqrt( ), kvadratrot .....	186	stykkevis funksjon (N-delers)	
standardavvik, stdDev( ) .....	188-189, 208	sjablon for .....	3
stat.resultater .....	186	stykkevis( ) .....	139
stat.verdier .....	188	subMat( ), undermatrise .....	190, 192
statistikk		subtrahere, - .....	221
en-variabel-statistikk, OneVar ..	134	sum (G)	
fakultet, ! .....	232	sjablon for .....	5
gjennomsnitt, mean( ) .....	118	sum av hovedbetalinger .....	237
kombinasjoner, nCr( ) .....	127	sum av rentebetalinger .....	236
median, median( ) .....	118	sum( ), summering .....	190
permutasjoner, nPr( ) .....	132	sum, Σ( ) .....	236
standardavvik, stdDev( ) ...	188-189, 208	sumlf( ) .....	191
tilfeldig startverdi, RandSeed ..	153	summering, sum( ) .....	190
		sumSeq( ) .....	192

svar (siste), Ans .....	13	tilMat(), tilfeldig matrise .....	152
<b>T</b>		tilnærmet, approx() .....	14
T, transponert .....	192	tInterval, t konfidensintervall .....	198
t test, tTest .....	202	tInterval_2Samp, toutvalg t konfidensintervall .....	198
tan <sup>-1</sup> ( ), invers tangens .....	193	tmpCnv() .....	199-200
tan(), tangens .....	192	to-variable resultater, TwoVar .....	205
tangens, tan() .....	192	tomme (åpne) elementer .....	263
tangentLine() .....	194	tomrom, test for .....	100
tangentlinje, tangentLine() .....	194	trace() .....	201
tanh <sup>-1</sup> ( ), hyperbolsk invers tangens .....	195	transponert, T .....	192
tanh(), hyperbolsk tangens .....	194	Try, feil håndteringskommando .....	201
Taylor polynom, taylor() .....	195	try, Try .....	201
taylor(), Taylor polynom .....	195	Try, try .....	201
tCdf(), sannsynlig student t-fordeling .....	196	tTest, t test .....	202
tCollect(), slå sammen trigonometrisk .....	196	tTest_2Samp, to-utvalgs t-test .....	203
tegn		TVM-argumenter .....	205
streng, char() .....	24	tvmFV() .....	204
tegnkode, ord() .....	137	tvmI() .....	204
tegne .....	252-254	tvmN() .....	204
tegnstreng, char() .....	24	tvmPmt() .....	204
Tekstkommando .....	197	tvmPV() .....	205
teller dager mellom datoer, dbd() ..	47	TwoVar, to-variable resultater .....	205
telllf(), antall betingede elementer i en liste .....	37	<b>U</b>	
test for tomrom, isVoid() .....	100	ubestemt integral sjablon for .....	6
Test_2S, 2_ utvalg F test .....	80	ulik, ≠ .....	228
tExpand(), utvide trigonometrisk ...	196	undermatrise, subMat() .....	190, 192
tidsverdi for penger, antall betalinger .....	204	unitV(), enhetsvektor .....	207
tidsverdi for penger, betalingsbeløp .....	204	unLock, lås opp variabel eller variabelgruppe .....	208
tidsverdi for penger, Fremtidig verdi .....	205	uttrykk streng til uttrykk, expr() .....	70, 113
tidsverdi for penger, Nåverdi .....	204	uttrykk til liste, expListe() .....	69
tier-potens, 10 <sup>^</sup> () .....	243	utvid, expand() .....	69
tilf(), tilfeldig nummer .....	151	utvid/sett sammen, augment() .....	16
tilfBin, tilfeldig tall .....	151	utvide trigonometrisk, tExpand() ...	196
tilfeldig		<b>V</b>	
matrise, tilfMat() .....	152	variabel opprette navn fra en tegnstreng .....	268
norm, tilfNorm() .....	152	variabler lokal, Local .....	111
polynom, tilfPoly() .....	153	slette alle enkelttegn .....	26
startverdi, RandSeed .....	153	slette, DelVar .....	51
tilfeldig utvalg .....	153	variabler og funksjoner kopiere .....	30
tilflnt(), tilfeldig heltall .....	151		
tilfNorm(), tilfeldig norm .....	152		
tilfPoly(), tilfeldig polynom .....	153		
tilfSamp() .....	153		

variabler, låse og låse opp	88, 112, 208	zInterval_2Prop, to-proporsjons z	
varians, variance( )	209	konfidensintervall	216
varPop( )	208	zInterval_2Samp, to-utvalgs z	
varSamp( ), utvalgets varians	209	konfidensintervall	217
varelskoder og meldinger	281	zTest	217
vektorer		zTest_1Prop, en-proporsjons z-test	218
enhet, unitV( )	207	zTest_2Prop, to-proporsjons z-test	219
kryssprodukt, crossP( )	39	zTest_2Samp, to-utvalgs z-test	219
Overfører øyeblikkelig kontroll			
til den neste		<b>Δ</b>	
iterasjonen i aktuell		Δlist( ), differensliste	108
løkke (For, While, eller		ΔtmpCnv() [tmpKnv]	200
Loop).	45		
prikk produkt, dotP( )	61	<b>X</b>	
venstre, left( )	101	χ <sup>2</sup> Cdf( )	25
verken ... eller, Boolsk operator	130	χ <sup>2</sup> GOF	25
vinkel, angle( )	10		
vis data, Vis	56, 167		
Vis, vis data	56, 167		
vise som			
binær, 4Grunntall2	18		
desimalt heltall, 4Grunntall10	20		
desimalvinkel, ►DD	48		
grader/minutter/sekunder,			
►DMS	58		
heksadesimal, 4Grunntall16	20		
polar vektor, ►Polar	140		
rektangulær vektor, ►Rect	154		
sfærisk vektor, ►Sphere	185		
sylindrisk vektor, ►Cylind	45		
<b>W</b>			
warnCodes( ), Warning codes	210		
when( ), når	210		
while, While	211		
While, while	211		
<b>X</b>			
x <sup>-1</sup> , resiprok	244		
x <sup>2</sup> , kvadrat	225		
XNOR	231		
<b>Z</b>			
zeroes( ), nullpunkt	213		
zInterval, z konfidensintervall	215		
zInterval_1Prop, en-proporsjons z			
konfidensintervall	216		