



# **Referenzhandbuch für den TI-84 Plus CE-T-Familie Grafik-Taschenrechner**

**Katalog, Befehle und Funktionen, Fehlermeldungen**

**Arithmetischen Operationen, Test Relations und  
Symbole**

Weitere Informationen zu TI Technology finden Sie in der Online-Hilfe unter [education.ti.com/eguide](http://education.ti.com/eguide).

## **Wichtige Informationen**

Sofern in der jeweiligen Programmlizenz nicht ausdrücklich anders aufgeführt, übernimmt Texas Instruments für die Programme oder das Handbuchmaterial keinerlei Garantie, weder direkt noch indirekt. Dies umfasst auch jegliche indirekte Gewährleistung hinsichtlich der Marktgängigkeit oder der Eignung für einen bestimmten Zweck, ist jedoch nicht hierauf beschränkt und dieses Produkt wird lediglich „so wie es ist“ zur Verfügung gestellt. In keinem Fall kann Texas Instruments für Schäden haftbar gemacht werden, die sich entweder in Verbindung mit dem Kauf bzw. Gebrauch dieses Produkts ergeben oder dadurch verursacht werden, dies gilt für spezielle, begleitende und versehentliche Schäden sowie für Folgeschäden. Texas Instruments haftet maximal und ausschließlich in der Höhe des Kaufpreises des Produkts, unabhängig vom jeweiligen Fall. Weiterhin haftet Texas Instruments nicht für Forderungen einer anderen Partei, die sich aus dem Gebrauch dieses Produkts ergeben, welcher Art diese Forderungen auch immer sein mögen.

© 2006 - 2020 Texas Instruments Incorporated

# Contents

<b>Einführung</b> .....	<b>1</b>
<b>KATALOG, Strings, Hyperbelfunktionen</b> .....	<b>2</b>
Was ist der KATALOG? .....	2
Durchsuchen der Kataloghilfe des TI-84 Plus CE-T .....	3
Verwenden der Kataloghilfe .....	5
Eingeben und Verwenden von Strings .....	7
Speichern von Strings in String-Variablen .....	8
Stringfunktionen und -anweisungen im KATALOG .....	10
Hyperbelfunktionen im KATALOG .....	15
<b>Auflistung der Befehle und Funktionen</b> .....	<b>17</b>
Alphabetische KATALOG-Liste .....	19
A .....	19
B .....	21
C .....	22
D .....	25
E .....	28
F .....	31
G .....	34
H .....	37
I .....	39
K .....	44
L .....	45
M .....	49
N .....	52
O .....	55
P .....	56
Q .....	64
R .....	64
S .....	67
T .....	77
U .....	81
V .....	82
W .....	83
X .....	84
Z .....	84

<b>Arithmetische Operationen, Test Relations und Sonderzeichen</b> .....	<b>92</b>
<b>Fehlermeldungen:</b> .....	<b>101</b>
<b>Allgemeine Informationen</b> .....	<b>109</b>
Online-Hilfe .....	109
Kontakt mit TI Support aufnehmen .....	109
Service- und Garantiefinformationen .....	109

# Einführung

In diesem Referenzhandbuch finden Sie die folgenden Informationen:

- **KATALOG, Strings, Hyperbelfunktionen** - Enthält Anweisungen zum Suchen, Verwenden und Eingeben von Zeichenfolgen und anderen Funktionen im KATALOG.
- **Auflistung der Befehle und Funktionen** - Enthält eine alphabetische Liste aller KATALOG-Elemente, verweisend auf:
  - Funktion oder Befehl/Argumente
  - Ergebnisse
  - Schlüssel oder Schlüssel/Menü oder Bildschirm/Element
- **Arithmetische Operationen, Test Relations und Sonderzeichen** - Elemente ohne alphabetischen Namen (z. B. +, ! oder >).
- **Fehlermeldungen** - Enthält eine Liste der Fehlertypen mit möglichen Ursachen und Empfehlungen zur Fehlerbehebung.

# KATALOG, Strings, Hyperbelfunktionen

## *Was ist der KATALOG?*

Der KATALOG ist eine alphabetische Liste aller im TI-84 Plus CE-T verwendeten Funktionen und Anweisungen. Mit Ausnahme der folgenden Elemente lassen sich die Katalogelemente auch über ein Menü oder die Tastatur aufrufen:

- die sechs String-Funktionen
- die sechs Hyperbelfunktionen
- die Anweisung **Löse{** ohne den Gleichungslöser-Editor
- die Inferenzstatistik-Funktionen ohne die Inferenzstatistik-Editoren

**Hinweis:** Die einzigen KATALOG-Programmierbefehle, die Sie vom Hauptbildschirm aus ausführen können, sind **GetCalc{**, **Get{** und **Send{**.

## Durchsuchen der Kataloghilfe des TI-84 Plus CE-T

### Auswählen eines Elements aus dem KATALOG

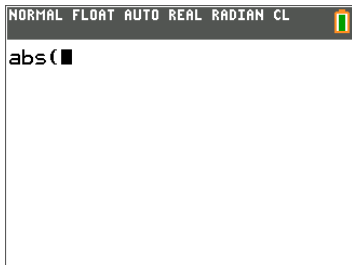
Gehen Sie wie folgt vor, um den **KATALOG** zu durchsuchen und ein Katalogelement auszuwählen.

1. Drücken Sie **[2nd]** [catalog], um den **KATALOG** anzuzeigen.



Das ► in der ersten Spalte ist der Auswahlcursor.

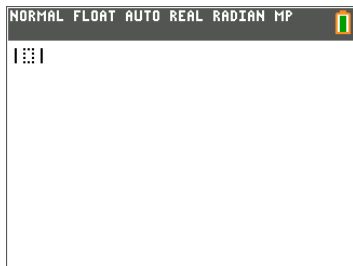
2. Drücken Sie **[↓]** oder **[↑]**, um durch den **KATALOG** zu scrollen, bis der Cursor auf das gewünschte Element zeigt.
  - Um zum ersten Element zu springen, das mit einem bestimmten Buchstaben beginnt, drücken Sie einfach die entsprechende Buchstabentaste – die Alpha-Sperre ist bereits eingeschaltet.
  - Elemente, die mit einer Zahl beginnen, sind in der alphabetischen Reihenfolge entsprechend dem ersten Buchstaben hinter der Zahl einsortiert. **2-PropZTest(** beispielsweise befindet sich bei den Elementen, die mit dem Buchstaben **P** beginnen.
  - Funktionen, die als Sonderzeichen dargestellt werden, wie. z. B.  $+$ ,  $^{-1}$ ,  $<$  und  $\sqrt{}$ , sind hinter dem letzten Element aufgeführt, das mit dem Buchstaben **Z** beginnt. Um zum ersten Sonderzeichen (!) zu springen, drücken Sie **[0]**.
3. Drücken Sie **[enter]**, um das Element in den aktuellen Bildschirm einzufügen.



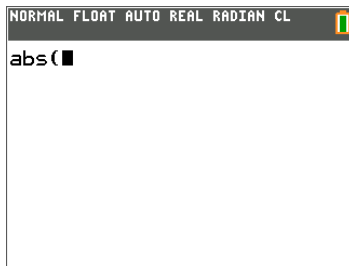
#### Hinweis:

- Um vom Anfang des **KATALOG**-Menüs zum Ende zu gelangen, drücken Sie **[↑]**. Drücken Sie **[↓]**, um vom Ende zum Anfang zu gelangen.

- Wenn sich Ihr TI-84 Plus CE-T im MathPrint™-Modus befindet, fügen viele Funktionen die MathPrint™-Vorlage im Hauptbildschirm ein. **abs(** fügt zum Beispiel die Absolutwert-Vorlage anstelle von **abs(** in den Hauptbildschirm ein.



MathPrint™



Klassisch



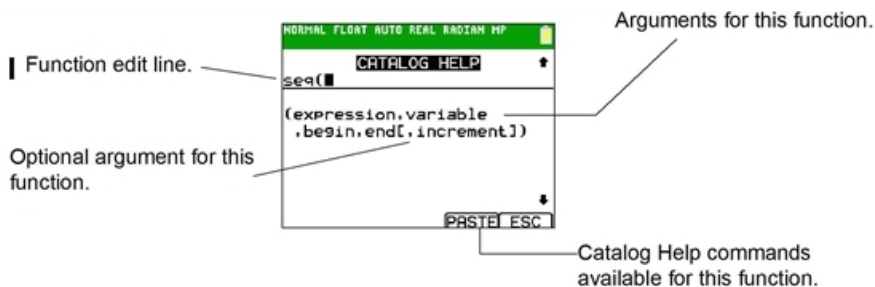
## Verwenden der Kataloghilfe

### Anzeigen der Kataloghilfe

Sie haben zwei Möglichkeiten, die Argumente für Funktionen in der Kataloghilfe anzuzeigen:

- Über eine alpha-/numerische Auflistung der Funktionen im Katalog (z. B.  $\boxed{2nd}$  [catalog]).
- Über die in bestimmten Menüs aufgeführten Funktionen (z. B.  $\boxed{math}$ ).

Die Kataloghilfe listet alle gültigen Argumente für die Funktion unter der Eingabezeile auf. Argumente in Klammern sind optional.



1. Rufen Sie das Menü auf, das die Funktion enthält.
2. Bewegen Sie den Cursor mit  $\boxed{\uparrow}$  und/oder  $\boxed{\downarrow}$  zur Funktion.
3. Drücken Sie  $\boxed{+}$ , um Argumente für die Funktion anzuzeigen. Der Cursor befindet sich auf der Eingabezeile.

### Hinweis:

- Der Katalog ( $\boxed{2nd}$  [catalog]) wird in alphabetischer Reihenfolge angezeigt. Beim Anzeigen des Katalogs ist die Alpha-Sperre eingeschaltet. Drücken Sie den ersten Buchstaben des Funktionsnamens, um alphabetisch davor liegende Funktionsnamen zu überspringen. Bewegen Sie den Cursor mit  $\boxed{\uparrow}$  und/oder  $\boxed{\downarrow}$  zur Funktion.
- Nicht allen Funktionen im Katalog sind Argumente zugeordnet. Wenn die Funktion kein Argument benötigt, zeigt die Kataloghilfe die Meldung „Für dieses Element sind keine Argumente erforderlich“ an.

### Kataloghilfe-Befehle

- Wählen Sie **MEHR** (soweit verfügbar), um mehr Argumente für die Funktion auszuwählen.

```

NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
dim(
(listname)
(matrixname)
↓
[MORE]

```

```

NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
Disp
[valueA,valueB,valueC,...,
value n]
no arguments
↓
[PASTE] ESC

```

- Sofern verfügbar, können Sie Schnellstastmenüs  $\alpha$  [f1] bis [f4] für Argumente verwenden.

```

NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
LinReg(a+bx) L1,L2,
↓
[X]listname,Ylistname
,freqlist,regEqu
1:Y1 6:Y6
2:Y2 7:Y7
3:Y3 8:Y8
4:Y4 9:Y9
5:Y5 0:Y0
↓
[FRAC] [FUNC]

```

- Geben Sie Ihre Argumentwerte in der Eingabezeile der Funktion ein und wählen Sie anschließend **EINFÜGEN**, um die Funktion und die eingegebenen Argumentwerte einzufügen.

**Hinweis:** Das Einfügen ist an den meisten Cursorpositionen möglich.

```

NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
LinReg(a+bx) L1,L2,Y3
↓
[X]listname,Ylistname
,freqlist,regEqu
↓
[PASTE] ESC

```

- Wählen Sie **ESC**, um den Kataloghilfe-Bildschirm zu verlassen.

# Eingeben und Verwenden von Strings

## Was ist ein String?

Ein String ist eine Abfolge von Zeichen, die von Anführungszeichen eingefasst ist. Beim TI-84 Plus EC-T gibt es zwei Hauptanwendungen für Strings.

- Sie definieren Text, der in einem Programm angezeigt wird.
- Sie nehmen Tastatureingaben in ein Programm an.

Zeichen sind die Einheiten, die zu einem String (oder einer Zeichenkette) kombiniert werden.

- Alle Ziffern, Buchstaben und Leerzeichen zählen jeweils als ein Zeichen.
- Jeder Anweisungs- oder Funktionsname, wie z. B. `sin(` oder `cos(`, zählt als ein Zeichen; der TI-84 Plus EC-T interpretiert jeden Anweisungs- oder Funktionsnamen als ein Zeichen.

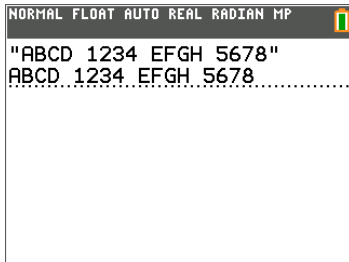
## Eingeben eines Strings

Gehen Sie wie folgt vor, um einen String in eine leere Zeile des Hauptbildschirms einzugeben.

1. Drücken Sie `[alpha]` `["]`, um den Anfang des Strings zu kennzeichnen.
2. Geben Sie die Zeichen ein, aus denen der String besteht.
  - Zum Erstellen der Zeichenkette können Sie jede beliebige Kombination aus Zahlen, Buchstaben, Funktionsnamen oder Anweisungsnamen verwenden.
  - Um ein Leerzeichen einzugeben, drücken Sie `[alpha]` `[ ]`.
  - Um mehrere alphabetische Zeichen hintereinander einzugeben, drücken Sie `[alpha]` `[A-lock]`, um die Alpha-Sperre zu aktivieren.
3. Drücken Sie `[alpha]` `["]`, um das Ende des Strings zu kennzeichnen.

**"String"**

4. Drücken Sie `[enter]`. Auf dem Hauptbildschirm wird die Zeichenkette in der nächsten Zeile ohne Anführungszeichen angezeigt. Auslassungspunkte (...) zeigen an, dass die Zeichenkette über den Bildschirm hinaus reicht. Um zum Anzeigen der gesamten Zeichenkette zu scrollen, drücken Sie `[right]` und `[left]`.



**Hinweis:** Eine Zeichenkette muss von Anführungszeichen eingeschlossen sein. Die Anführungszeichen zählen nicht als Zeichen der Zeichenkette.

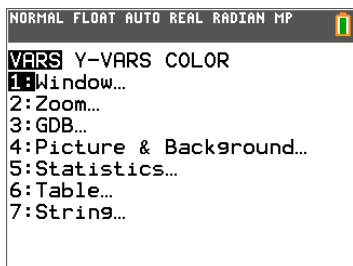
## Speichern von Strings in String-Variablen

### String-Variablen

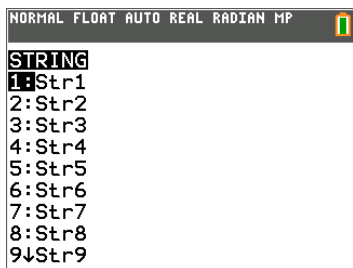
Beim TI-84 Plus EC-T können Sie Strings in 10 Variablen speichern. Sie können String-Variablen mit Stringfunktionen und -anweisungen verwenden.

Gehen Sie wie folgt vor, um das Menü **VARIABLEN STRING** anzuzeigen:

1. Drücken Sie `[vars]`, um das Menü **VARIABLEN** anzuzeigen. Bewegen Sie den Cursor auf **7:String**.



2. Drücken Sie `[enter]`, um das Untermenü **STRING** anzuzeigen.



### Speichern einer Zeichenkette in einer String-Variablen

Gehen Sie wie folgt vor, um einen String in einer String-Variablen zu speichern.

1. Drücken Sie `[alpha]` `["]`, geben Sie den String ein und drücken Sie dann `[alpha]` `["]`.
2. Drücken Sie `[sto->]`.
3. Drücken Sie `[vars]` **7**, um das Menü **VARIABLEN STRING** anzuzeigen.
4. Wählen Sie die String-Variablen (**Str1** bis **Str9** oder **Str0**), in der Sie die Zeichenkette speichern möchten.

```
NORMAL FLOAT AUTO REAL RADIAN MP
STRING
1:Str1
2:Str2
3:Str3
4:Str4
5:Str5
6:Str6
7:Str7
8:Str8
9↓Str9
```

Die String-Variablen werden an der aktuellen Cursorposition neben dem Speichersymbol (→) eingefügt.

5. Drücken Sie `enter`, um die Zeichenkette in der String-Variablen zu speichern. Auf dem Hauptbildschirm wird die gespeicherte Zeichenkette in der nächsten Zeile ohne Anführungszeichen angezeigt.

```
NORMAL FLOAT AUTO REAL RADIAN MP
"HELLO"→Str2
HELLO
█
```

### Den Inhalt einer Variablen anzeigen

Um den Inhalt einer String-Variablen im Hauptbildschirm anzuzeigen, wählen Sie die String-Variablen aus dem Menü **VARIABLEN STRING** aus und drücken dann `enter`. Die Zeichenkette wird angezeigt.

```
NORMAL FLOAT AUTO REAL RADIAN MP
Str2
HELLO
█
```

## Stringfunktionen und -anweisungen im KATALOG

### Anzeigen der Stringfunktionen und -anweisungen im KATALOG

Stringfunktionen und -anweisungen stehen nur im KATALOG zur Verfügung. In der nachstehenden Tabelle sind die Stringfunktionen und -anweisungen in der Reihenfolge aufgeführt, in der sie zwischen den anderen **KATALOG**-Menüpunkten angezeigt werden. Die Auslassungspunkte in der Tabelle zeigen weitere KATALOG-Elemente an.

---

#### KATALOG

...	
Equ►String(	Konvertiert eine Gleichung in eine Zeichenkette.
...	
Ausdr(	Wandelt einen String in einen Ausdruck um.
...	
inString(	Liefert die Nummer des Platzes eines Zeichens.
...	
Länge(	Liefert die Anzahl der Zeichen einer Zeichenkette.
...	
String►Equ(	Wandelt einen String in eine Gleichung um.
sub(	Liefert eine Teilmenge eines Strings als String.
...	

---

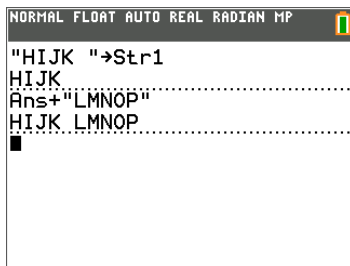
#### Verkettung

Gehen Sie wie folgt vor, um zwei oder mehr Strings zu verketteten.

1. Geben Sie *string1* ein (dabei kann es sich um eine Zeichenkette oder einen String-Namen handeln).
2. Drücken Sie  $\boxed{+}$ .
3. Geben Sie *string2* ein (dabei kann es sich um eine Zeichenkette oder einen String-Namen handeln). Drücken Sie bei Bedarf  $\boxed{+}$  und geben Sie *string3* ein (usw.).

*string1+string2+string3...*

4. Drücken Sie  $\boxed{\text{enter}}$ , um die Strings als eine einzige Zeichenkette anzuzeigen.



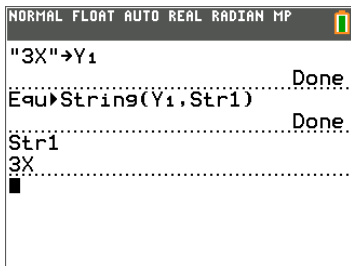
## Auswahl einer String-Funktion aus dem KATALOG

Um eine String-Funktion oder -anweisung auszuwählen und im aktuellen Bildschirm einzufügen, führen Sie die nachstehenden Schritte zur Auswahl eines Elements aus dem KATALOG aus.

### EquString(

**EquString(** konvertiert eine Gleichung in eine Zeichenkette. Die Gleichung muss in einer VARS Y-VARS Variablen gespeichert werden. *Yn* enthält die Gleichung. *Strn* (von **Str1** bis **Str9** oder **Str0**) ist die Zeichenkettenvariable, unter der die Gleichung gespeichert werden soll.

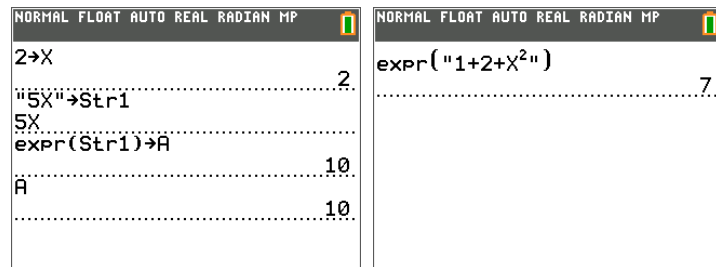
### EquString(Yn,Strn)



### Ausdr(

**Ausdr(** gibt die in *String* enthaltene Zeichenkette als Ausdruck zurück und führt diesen aus. *String* kann ein String oder eine String-Variable sein.

### Ausdr(String)



### inString(

**inString(** gibt an, an welcher Position im *String* das erste Zeichen des *Teilstrings* steht. *String* kann ein String oder eine String-Variable sein. *Start* ist eine optionale Zeichenposition, an der die Suche gestartet werden soll; standardmäßig ist dies 1.

### inString(String,subString[,Start])

```

NORMAL FLOAT AUTO REAL RADIAN MP
inString("PQRSTUVWXYZ", "STU")
.....4
inString("ABCABC", "ABC", 4)
.....4

```

**Hinweis:** Enthält *String* die Zeichenkette *Teilstring* nicht oder ist *Start* größer als die Länge von *String*, gibt **inString()** **Null** zurück.

### Länge()

**Länge()** liefert die Anzahl der Zeichen in *String*. *String* kann ein String oder eine String-Variable sein.

**Hinweis:** Anweisungen und Funktionsnamen wie z. B. **sin()** oder **cos()** zählen jeweils als ein Zeichen.

### Länge(*String*)

```

NORMAL FLOAT AUTO REAL RADIAN MP
"WXYZ"→Str1
WXYZ
length(Str1)
.....4

```

### String→Equ()

**String→Equ()** wandelt *String* in eine Gleichung um und speichert die Gleichung in *Yn*. *String* kann ein String oder eine String-Variable sein. **String→Equ()** ist der Kehrwert von **Equ→String()**.

### String→Equ(*String*, *Yn*)

```

NORMAL FLOAT AUTO REAL RADIAN MP
"2X"→Str2
2X
String→Equ(Str2, Y2)
.....Done

```

```

NORMAL FLOAT AUTO REAL RADIAN MP
Plot1 Plot2 Plot3
Y1=
Y2=2X
Y3=
Y4=
Y5=
Y6=
Y7=
Y8=
Y9=

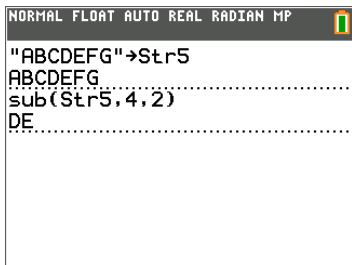
```



## sub(

**sub(** ergibt eine Zeichenkette, die eine Teilmenge eines vorhandenen *Strings* ist. *String* kann ein String oder eine String-Variable sein. *Anfang* ist die Positionsnummer des ersten Zeichens der Teilmenge. *Länge* ist die Anzahl der Zeichen in der Teilmenge.

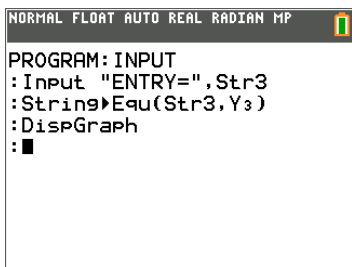
### sub(*String,Anfang,Länge*)



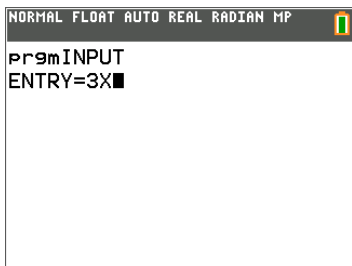
```
NORMAL FLOAT AUTO REAL RADIAN MP
"ABCDEFGH"→Str5
ABCDEFGH
sub(Str5,4,2)
DE
```

### Eingabe einer zu zeichnenden Funktion während der Ausführung eines Programms

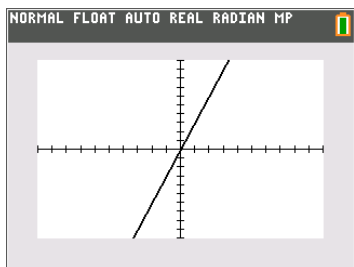
Mit diesen Befehlen können Sie in einem Programm eine Funktion eingeben, die während der Programmausführung gezeichnet wird.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: INPUT
:Input "ENTRY=",Str3
:String→Equ(Str3,Y3)
:DispGraph
:■
```



```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmINPUT
ENTRY=3X■
```



**Hinweis:** Geben Sie bei Ausführung dieses Programms eine Funktion zum Speichern in **Y3** in die Eingabeaufforderung **EINGABE=** ein.

# Hyperbelfunktionen im KATALOG

## Hyperbelfunktionen

Der Zugriff auf die Hyperbelfunktionen ist nur aus dem KATALOG heraus möglich. In der nachstehenden Tabelle sind die Hyperbelfunktionen in der Reihenfolge aufgeführt, in der sie zwischen den anderen **KATALOG**-Menüpunkten angezeigt werden. Die Auslassungspunkte in der Tabelle zeigen weitere KATALOG-Elemente an.

---

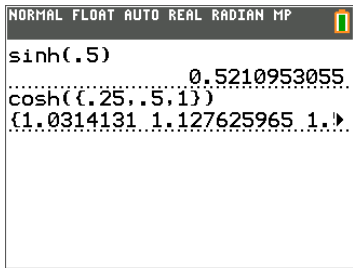
<b>KATALOG</b>	
...	
$\cosh()$	Cosinus hyperbolicus
$\cosh^{-1}()$	Arkuskosinus hyperbolicus
...	
$\sinh()$	Sinus hyperbolicus
$\sinh^{-1}()$	Arkussinus hyperbolicus
...	
$\tanh()$	Tangens hyperbolicus
$\tanh^{-1}()$	Arkustangens hyperbolicus
...	

---

### $\sinh()$ , $\cosh()$ , $\tanh()$

Die Hyperbelfunktionen sind  $\sinh()$ ,  $\cosh()$  und  $\tanh()$ . Sie sind auf für reelle Zahlen, Ausdrücke und Listen anwendbar.

- $\sinh(\text{Wert})$
- $\cosh(\text{Wert})$
- $\tanh(\text{Wert})$



### $\sinh^{-1}()$ , $\cosh^{-1}()$ , $\tanh^{-1}()$

$\sinh^{-1}()$  ist die hyperbolische Arkussinus-Funktion.  $\cosh^{-1}()$  ist die hyperbolische Arkuskosinus-Funktion.  $\tanh^{-1}()$  ist die hyperbolische Arkustangens-Funktion. Sie sind auf für reelle Zahlen, Ausdrücke und Listen anwendbar.

- $\sinh^{-1}(\text{Wert})$
- $\cosh^{-1}(\text{Wert})$
- $\tanh^{-1}(\text{Wert})$

```
NORMAL FLOAT AUTO REAL RADIAN MP
sinh⁻¹({0,1})
{0.881373587}
tanh⁻¹(-.5)
-0.5493061443
█
```

# Auflistung der Befehle und Funktionen

Diese Informationstabelle enthält eine kurze Beschreibung und gegebenenfalls die Syntax der Befehlsargumente sowie die Menü-Positionen der einzelnen Befehle und Funktionen in der Katalog-Liste des Taschenrechners.

Diese Tabelle ist hilfreich bei der Ausführung von Befehlen mit dem Taschenrechner und beim Erstellen von TI-Basic-Programmen.

Elemente, deren Namen nicht alphabetisch sind (wie +, ! und >) finden Sie im Abschnitt *Arithmetische Operationen, Test Relations und Symbole*. Wenn nicht anders angegeben, wurden sämtliche Beispiele im standardmäßigen Reset-Modus ausgeführt, wobei alle Variablen als Standardwert 0 angenommen wurden.

Vom **KATALOG** aus können Sie jede Funktion bzw. jeden Befehl in den Hauptbildschirm oder in eine Befehlszeile des Programm-Editors einfügen.

Die nachstehenden Syntax-Informationen zu Funktions- und Befehlsargumenten finden Sie auch im Taschenrechner und im Programm-Editor von TI Connect™ CE.

- Beim Taschenrechner können Sie den Syntax-Editor der Kataloghilfe aufrufen, indem Sie [+ ] drücken, wenn eine Funktion oder ein Befehl in der Menüliste markiert ist.
- Beim TI Connect™ CE Programm-Editor zeigt die Katalog-Liste auch die Syntax der Argumente für Funktionen und Befehle an.

Beachten Sie bitte, dass einige Funktionen und Befehle nur gültig sind, wenn sie in einem TI-Basic-Programm und nicht vom Hauptbildschirm aus ausgeführt werden.

Die Elemente in dieser Tabelle werden in derselben Reihenfolge aufgeführt wie im **KATALOG** (2nd [Katalog].)

In der nachstehenden Tabelle steht das Zeichen † entweder für einen Tastendruck oder einen bestimmten Befehl, der nur im Modus Programm-Editor des Taschenrechners verfügbar ist. Um den Taschenrechner in den Programmbearbeitungsmodus zu schalten, drücken Sie prgm und wählen Sie **BEARBEITEN**, um ein bestehendes Programm zu bearbeiten, oder **NEU**, um ein neues Programm zu starten.

Einige Argumente sind optional. Optionale Argumente stehen in der Syntaxhilfe der nachstehenden Tabelle in [ ]. [ ] sind keine Sonderzeichen im Taschenrechner und dürfen nicht eingegeben werden. Sie werden nur hier verwendet, um optionale Argumente zu kennzeichnen.

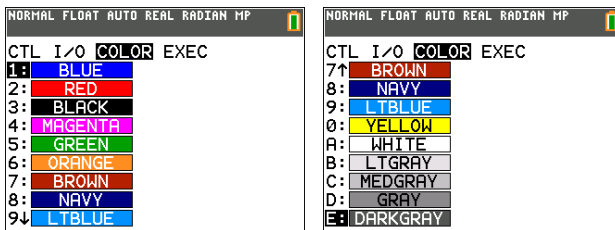
Im Taschenrechner werden Funktionen und Befehle als „Token“ eingefügt. Das bedeutet, dass sie als ein Zeichen eingefügt werden und nicht als einzelne Buchstaben, Sonderzeichen und Leerzeichen. Versuchen Sie nicht, auf dem Taschenrechner eine Funktion oder einen Befehl einzutippen. Fügen Sie sie ausschließlich über das Menü ein. Achten Sie beim Bearbeiten darauf, wie der Cursor über die Tokens hinwegspringt. Dies gibt Ihnen ein besseres Verständnis für Tokens.

Im Programm-Editor von TI Connect™ CE werden Tokens bei Verwendung der Katalog-Struktur ebenso eingefügt. Hier können Sie die Funktionen und Befehle auch von Hand eingeben, wenn Sie Format und Syntax korrekt verwenden. TI Connect™ CE wandelt die

Funktionen und Befehle in Tokens um, wenn Sie das Programm an den Taschenrechner senden. Sie müssen jedoch bei der Eingabe der Funktionen und Befehle die exakte Schreibweise der Tokens einhalten. Beachten Sie in diesem Zusammenhang auch, dass einige Befehle Leerzeichen als Teil des Tokens enthalten, die Sie möglicherweise nicht sehen. Beim Befehl „Pause“ beispielsweise befindet sich am Ende des Tokens ein Leerzeichen. Wenn Sie das Programm an den Taschenrechner senden, können Sie das Programm ausführen. Vorhandene Syntaxfehler können sowohl auf dem Taschenrechner als auch im Programm-Editor von TI Connect™ CE behoben werden.

SATRG	E/A	FARBE	AUSFÜHRG
		Farbnummern	Namen
		10	BLAU
		11	ROT
		12	SCHWARZ
		13	MAGENTA
		14	GRÜN
		15	ORANGE
		16	BRAUN
		17	DKLBLAU
		18	HELLBLAU
		19	GELB
		20	WEIß
		21	HELLGRAU
		22	MITTGRAU
		23	GRAU
		24	DKLGRAU

Sie können auch einen Namen im Menü `vars` (Untermenü **FARBE**) auswählen.



**GraphColor**(Funktion#,Farbe#)

Z. B. **GraphColor**(2,4) oder **GraphColor**(2,MAGENTA).

# Alphabetische KATALOG-Liste

## A

### AbhAbfr

#### AbhAbfr

+ [2nd] [TBLSET]

Stellt eine Tabelle zur Abfrage abhängiger Variablenwerte ein.

Depend: Ask

### AbhAuto

#### AbhAuto

+ [2nd]

[TBLSET]

Stellt eine Tabelle zur automatischen Erzeugung abhängiger Variablenwerte ein.

Depend:

Auto

### abs()

#### abs(*Wert*)

[MATH]

Liefert den Absolutwert einer reellen Zahl, eines Ausdrucks, einer Liste oder einer Matrix.

NUM

1:abs(

### abs()

#### abs(*komplexer Wert*)

[MATH]

Ergibt den Betrag einer komplexen Zahl oder Liste.

KMPLX

5:abs(

### AchsenAus

#### AchsenAus

+ [2nd]

[FORMAT]

Schaltet die Achsen des Diagramms aus.

AchsenAus

### AchsenEin

#### AchsenEin[Farbe#]

+ [2nd]

[FORMAT]

Schaltet die Achsen des Diagramms mit Farbe ein. Die Option *Farbe* ermöglicht die Angabe der Farbe für die Achsen.

AchsenEin

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

### AlleListenLöschen

#### AlleListenLöschen

[2nd] [MEM]

## AlleListenLöschen

Setzt die Dimension aller Listen im Speicher auf **0**.

**SPEICHER**  
**4:AlleListenLöschen**

## ANOVA()

**ANOVA(Liste1,Liste2[,Liste3,...,Liste20])**

**[STAT]**

Führt eine einfache Varianzanalyse durch, um die Mittelwerte von zwei bis maximal 20 Grundgesamtheiten zu vergleichen.

**TESTS**  
**H:ANOVA()**

## Antw

**Antw**

**[2nd] [ANS]**

Liefert die letzte Antwort.

## Archivieren

*Variablenarchivieren*

**[2nd] [MEM]**

Verschiebt die spezifizierte *Variable* aus dem RAM in den Benutzerspeicher.

**5:Archivieren**

## AsymDetktAus

**AsymDetktAus**

+ **[2nd] [FORMAT]**  
**AsymDetktAus**

Deaktiviert Überprüfungen auf Asymptoten rationaler Funktionen während des Zeichnens. Wirkt sich auf die Geschwindigkeit der Kurvenerstellung aus. Führt während des Zeichnens keine zusätzlichen Berechnungen zur pixelweisen Ermittlung von Asymptoten aus. Pixel werden über den Bildschirm auch über eine Asymptote hinweg verbunden.

## AsymDetktEin

**AsymDetktEin**

+ **[2nd] [FORMAT]**  
**AsymDetktEin**

Aktiviert Überprüfungen auf Asymptoten rationaler Funktionen während des Zeichnens. Wirkt sich auf die Geschwindigkeit der Kurvenerstellung aus. Führt mehr Berechnungen aus und verbindet Pixel nicht über eine Asymptote auf einem Graphen hinweg.

## Ausdr(

**Ausdr(String)**

+ **[PRGM]**  
**I/O**  
**Ausdr(**

Gibt die in *String* enthaltene Zeichenkette als Ausdruck zurück und führt diesen Ausdruck aus. *String* kann ein String oder eine String-Variablen sein.



## AusdrAus

### AusdrAus

Schaltet die Anzeige des Ausdrucks während **SPUR** aus.

+ [2nd]  
[FORMAT]  
AusdrAus

## AusdrEin

### AusdrEin

Schaltet die Anzeige des Ausdrucks während **SPUR** ein.

+ [2nd]  
[FORMAT]  
AusdrEin

## Auswählen(

### Auswählen(*XListenname*,*YListenname*)

Wählt einen oder mehrere spezifische Datenpunkte aus einem Streudiagramm oder einem xyLinien-Plot aus und speichert die ausgewählten Datenpunkte in zwei neuen Listen, *XListenname* und *YListenname*.

[2nd] [LIST]  
OPS  
8:Auswählen(

## AUTO Antwort

### AUTO

Zeigt Antworten in einem ähnlichen Format an wie die Eingabe.

[MODE]  
Antworten:  
AUTO

## a+bi

### a+bi

Stellt den Modus auf das Format für kartesische komplexe Zahlen (a+bi).

+ [MODE]  
a+b i

## B

## bal(

### bal(*npmt*,*WertRunden*)

Berechnet den Saldo bei *npmt* für einen Amortisationsplan, der gespeicherte Werte für **PV**, **I%** und **PMT** verwendet, und rundet die Berechnung auf *WertRunden*.

[APPS]  
1:Finanzen  
BERECHNEN  
9:bal(

## binomcdf(

### binomcdf(*AnzahlVersuche*,*p*,*x*)

Berechnet die kumulative Wahrscheinlichkeit an *x* für die diskrete Binomialverteilung mit der festgelegten *AnzahlVersuche* und der

[2nd] [DISTR]  
VERTLG  
B:binomcdf(

## binomcdf(

Wahrscheinlichkeit  $p$  für den Erfolg in jedem Einzelversuch.

## binompdf(

**binompdf**(AnzahlVersuche, $p$ , $x$ )

Berechnet die Wahrscheinlichkeit an  $x$  für die diskrete Binomialverteilung mit der festgelegten AnzahlVersuche und der Wahrscheinlichkeit  $p$  für den Erfolg in jedem Einzelversuch.

**2nd** **[DISTR]**  
**VERTLG**  
**A:binompdf(**

## Bogenmaß

**Bogenmaß**

Stellt den Winkelmodus Bogenmaß ein.

**+** **[MODE]**  
**Bogenmaß**

## ► Bruch

*Wert* ► **Bruch**

Zeigt eine(n) reelle(n) oder komplexe(n) Zahl, Ausdruck, Liste oder Matrix als einen auf seine einfachsten Terme vereinfachten Bruch an.

**[MATH]**  
**MATHE**  
**1: ► Bruch**

## C

## $\chi^2$ GOF

**$\chi^2$ GOF-Test**(beobachteteListe,erwarteteListe,df  
[,ZeichneFlag,Farbe#])

Berechnet eine Testgröße, um zu überprüfen, ob die Stichprobendaten aus einer Grundgesamtheit stammen, die einer bestimmten Verteilung genügt.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**+** **[STAT]**  
**TESTS**  
**D:  $\chi^2$  GOF -**  
**Test(**

## $\chi^2$ -Test(

**$\chi^2$ -Test**(beobachteteMatrix,erwarteteMatrix  
[,ZeichneFlag,Farbe#])

Führt einen Chi-Quadrat-Test durch. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**+** **[STAT]**  
**TESTS**  
**C:  $\chi^2$  - Test(**

## $\chi^2$ cdf(

$\chi^2$ cdf(*untereGrenze,obereGrenze,df*)

**2nd** **[DISTR]**  
**VERTLG**  
**8:  $\chi^2$  cdf(**

Berechnet die  $\chi^2$  Verteilungswahrscheinlichkeit zwischen *untereGrenze* und *untereGrenze* für die angegebenen Freiheitsgrade *df*.

## $\chi^2$ pdf(

$\chi^2$ pdf(*x,df*)

**2nd** **[DISTR]**  
**VERTLG**  
**7:  $\chi^2$  pdf(**

Berechnet die Wahrscheinlichkeitsdichtefunktion (Pdf) einer  $\chi^2$ -Verteilung an einem bestimmten Wert *x* für die vorgegebenen Freiheitsgrade *df*.

## $\chi^2$ -Test(

$\chi^2$ -Test(*beobachteteMatrix,erwarteteMatrix*  
[,*ZeichneFlag,Farbe#*])

**+ [STAT]**  
**TESTS**  
**C:  $\chi^2$  - Test(**

Führt einen Chi-Quadrat-Test durch. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

*Farbe#*: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## ClrHome

ClrHome

**+ [PRGM]**  
**I/O**  
**8:ClrHome**

Löscht den Hauptbildschirm.

## ClrTable

ClrTable

**+ [PRGM]**  
**I/O**  
**9:ClrTable**

Löscht alle Werte einer Tabelle.

## cos(

cos(*Wert*)

**[COS]**

Ergibt den Kosinus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

$\cos^{-1}(\$

$\cos^{-1}(\text{Wert})$

**[2nd]** **[cos<sup>-1</sup>]**

Ergibt den Arkuskosinus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

**cosh(**

**cosh(Wert)**

**[2nd]** **[CATALOG]**

**cosh(**

Liefert den Cosinus hyperbolicus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

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

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

**[2nd]** **[CATALOG]**

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

Liefert den Arkuskosinus hyperbolicus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

**CubicReg**

**CubicReg**

**[XListenname,YListenname,Häufigkeitsliste,regequ]**

**[STAT]**

**BERECHNEN**

**6:CubicReg**

Passt ein Modell für kubische Regression an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

## D

### DatumEinst(

**DatumEinst**(*Jahr, Monat, Tag*)

**2nd** [CATALOG]

Stellt das Datum im Format Jahr, Monat, Tag ein. Das *Jahr* muss vierstellig eingegeben werden, *Monat* und *Tag* können aus einer oder zwei Ziffern bestehen.

**DatumEinst**(

### dbd(

**dbd**(*Datum1, Datum2*)

[APPS]

Zählt die tatsächlichen Tage und gibt die Anzahl der Tage zwischen *Datum1* und *Datum2* zurück.

**1:Finanzen  
BERECHNEN  
D:dbd(**

### DelVar

**DelVar** *Variable*

+ [PRGM]

Löst den Inhalt von *Variable* aus dem Speicher.

**SATRG  
G:DelVar**

### det(

**det**(*Matrix*)

**2nd**

Ergibt die Determinante von *Matrix*.

[MATRIX]  
**MATH  
1:det(**

### DEZ Antwort

**DEZ**

[MODE]

Zeigt Antworten als ganze Zahlen oder als Dezimalzahlen an.

**Antworten:  
DEZ**

### ►Dez

**Wert►Dez**

[MATH]

Zeigt eine(n) reelle(n) oder komplexe(n) Zahl, Ausdruck, Liste oder Matrix im Dezimalformat an.

**MATH  
2: ► Dez**

### DiagnoseAus

**DiagnoseAus**

**2nd** [CATALOG]

## DiagnoseAus

Schaltet den Modus für ausgeschaltete Diagnose ein;  $r$ ,  $r^2$  und  $R^2$  werden nicht als Regressionsmodell-Ergebnisse angezeigt.

**DiagnoseAus**

## DiagnoseAus

**DiagnoseAus**

**2nd [CATALOG]**

Schaltet den Modus für ausgeschaltete Diagnose ein;  $r$ ,  $r^2$  und  $R^2$  werden nicht als Regressionsmodell-Ergebnisse angezeigt.

**DiagnoseAus**

## DiagnoseEin

**DiagnoseEin**

**2nd [CATALOG]**

Schaltet den Modus für eingeschaltete Diagnose ein;  $r$ ,  $r^2$  und  $R^2$  werden als Regressionsmodell-Ergebnisse angezeigt.

**DiagnoseEin**

## Dick

**Dick**

+ **MODE**

Setzt alle Linienart-Einstellungen im Y=Editor auf Dick.

**Dick**

## dim(

**dim(Listenname)**

**2nd [LIST]**

Gibt die Dimension von *Listenname* zurück.

**OPS**

**3:dim(**

## dim(

**dim(Matrixname)**

**2nd [MATRIX]**

Gibt die Dimension von *Matrixname* als Liste zurück.

**MATH**

**3:dim(**

## dim(

*Length*→**dim(Listenname)**

**2nd [LIST]**

Weist einem neuen oder vorhandenen *Listennamen* eine neue Dimension (*Length*) zu.

**OPS**

**3:dim(**

## dim(

{*Zeilen,Spalten*}→**dim(Matrixname)**

**2nd [MATRIX]**

## dim(

Weist einem neuen oder vorhandenen *Matrixnamen* neue Dimensionen zu.

**MATH**  
3:dim(

## Disp

### Disp

Zeigt den Hauptbildschirm an.

+ **PRGM**  
I/O  
3:Disp

## Disp

### Disp [*WertA*,*WertB*,*WertC*,...,*Wert n*]

Zeigt jeden Wert an.

+ **PRGM**  
I/O  
3:Disp

## DispGraph

### DispGraph

Zeigt den Graphen an.

+ **PRGM**  
I/O  
4:DispGraph

## DispTable

### DispTable

Zeigt die Tabelle an.

+ **PRGM**  
I/O  
5:DispTable

## ►DMS

### *Wert*►DMS

Zeigt einen *Wert* im Format DMS an.

**2nd**  
**[ANGLE]**  
WINKEL  
4: ► DMS

## DS<(

### DS<(*Variable*,*Wert*):*BefehlA*:*Befehle*

Verringert *Variable* um 1; überspringt *BefehlA*, wenn *Variable*<*Wert*.

+ **PRGM**  
SATRG  
B:DS<(

## DtFmtEinst(

### DtFmtEinst(*Ganzzahl*)

**2nd**

## DtFmtEinst(

Stellt das Datumsformat ein.

1 = M/T/J  
2 = T/M/J  
3 = J/M/T

[CATALOG]  
DtFmtEinst  
(

## Dünn

Dünn

+ [MODE]  
Dünn

Setzt alle Linienart-Einstellungen im Y=Editor auf Dünn.

## E

e

e

[2nd] [e]

Ergibt die dezimale Annäherung der Konstante  $e$ .

e^(

e^(Potenz)

[2nd] [e<sup>x</sup>]

Ergibt die  $x$ -te Potenz von  $e$ .

e^(

e^(Liste)

[2nd] [e<sup>x</sup>]

Ergibt eine Liste von  $e$  hoch Potenzliste.

E

Exponent:

WertEExponent

[2nd] [EE]

Liefert den *Wert* mal 10 hoch *Exponent*.

E

Exponent:

ListeEExponent

[2nd] [EE]

Liefert die *Listenelemente* mal 10 hoch *Exponent*.



## E

### Exponent:

*Matrix***E***Exponent*

**[2nd] [EE]**

Liefert die *Matrix*elemente mal 10 hoch *Exponent*.

### ►Eff(

►Eff(*Nennrate*,  
*Perioden zusammengesetzt*)

Berechnet die jährliche Effektivverzinsung.

**[APPS]**

**1:Finanzen  
BERECHNEN  
C: ► Eff(**

### Eingaben löschen

#### Eingaben löschen

Löscht den Inhalt des Speicherbereichs Letzte Eingabe.

**[2nd] [MEM]  
SPEICHER  
3:Eingaben  
löschen**

### Eingeben

**Prompt***VariableA*[,*VariableB*,...,*Variable n*]

Fordert die Eingabe eines Werts für *VariableA*, dann *VariableB* usw.

**+ [PRGM]  
I/O**

**2:Eingeben**

### Einheitsm(

**Einheitsm**(*Dimension*)

Ergibt die Einheitsmatrix aus *Dimension* Zeilen x *Dimension* Spalten.

**[2nd] [MATRIX]  
MATH  
5:Einheitsm  
(**

### Else

#### Else

*Siehe* If:Then:Else

### Ende

#### Ende

Kennzeichnet das Ende von **For** (, **If-Then-Else**, **Repeat** oder **While**-Schleife.

**+ [PRGM]  
SATRG  
7:Ende**

## Eng

### Eng

+ [MODE]

Stellt den technischen Anzeigemodus ein.

Eng

## EntArchivieren

### EntArchivierenVariable

[2nd] [MEM]

Verschiebt die angegebenen Variablen aus dem Benutzerspeicher in den RAM.

6:EntArchivieren

Um Variablen zu archivieren, benutzen Sie **Archiv**.

## EquString(

### EquString(Y= var,Strn)

[2nd]

[CATALOG]

Wandelt den Inhalt einer Y= var in eine Zeichenkette um und speichert diese in Strn

Equ ▶ String(

## erweitern()

### erweitern(MatrixA,MatrixB)

[2nd] [MATRIX]

MATH

Ergibt eine Matrix, die durch Anfügen von MatrixB an MatrixA als neue Spalten erzeugt wurde.

7:erweitern(

## erweitern()

### erweitern(ListeA,ListeB)

[2nd] [LIST]

OPS

Gibt eine Liste zurück, die durch Anfügen von ListeB ans Ende von ListeA erzeugt wurde

9:erweitern(

## eval(

### eval(Ausdruck)

+ [PRGM]

I/O

Gibt einen ausgewerteten Ausdruck als Zeichenkette mit 8 signifikanten Ziffern aus. Der Ausdruck muss sich auf einen reellen Ausdruck vereinfachen lassen.

C:eval(

## eval(

TI-Innovator™

Hub

### eval(Ausdruck)

+ [PRGM]

HUB

Gibt einen ausgewerteten Ausdruck als Zeichenkette mit 8 signifikanten Ziffern aus. Der Ausdruck muss sich auf einen reellen Ausdruck vereinfachen lassen.

6:eval(

## ExecLib

### ExecLib

Erweitert TI-Basic (nicht verfügbar)

+ **PRGM**  
SATRG  
K:ExecLib

## ExpReg

### ExpReg

[*XListenname*,*YListenname*,*Häufigkeitsliste*,*regequ*]

Passt ein Modell für exponentielle Regression an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

**STAT**  
BERECHNEN  
0:ExpReg

## F

## Fix

### Fix #

Stellt die Anzahl der Dezimalstellen im Modus „Fixierte Dezimalstellen“ ein.

+ **MODE**  
**0123456789**  
(eine wählen)

## FktAus

**FktAus** [*Funktion#*,*Funktion#*,...,*Funktion n*]

Hebt die Auswahl aller **Y=** Funktionen oder nur der angegebenen **Y=** Funktionen auf.

**VAR**  
Y-VARS  
4:Ein/Aus  
2:FktAus

## FktEin

**FktEin** [*Funktion#*,*Funktion#*,...,*Funktion n*]

Wählt alle **Y=** Funktionen oder nur die angegebenen **Y=** Funktionen aus.

**VAR**  
Y-VARS  
4:Ein/Aus  
1:FktEin

## Fließ

### Fließ

Stellt den Fließkomma-Dezimalmodus ein.

+ **MODE**  
Fließ

## fMax(

**fMax**(*Ausdruck*,*Variable*,*untere*,*obere*[,*Toleranz*])

**MATH**  
MATH

## fMax(

Liefert den Wert von *Variable*, an dem das lokale Maximum von *Ausdruck* zwischen *untere* und *obere* mit der angegebenen *Toleranz* auftritt.

7:fMax(

## fMin(

**fMin(Ausdruck,Variable,untere,obere[,Toleranz])**

**MATH**

Liefert den Wert von *Variable*, an dem das lokale Minimum von *Ausdruck* zwischen *untere* und *obere* mit der angegebenen *Toleranz* auftritt.

**MATH**

6:fMin(

## fnInt(

**fnInt(Ausdruck,Variable,untere,obere[,Toleranz])**

**MATH**

Ergibt das Funktionsintegral von *Ausdruck* in Bezug auf *Variable* zwischen *untere* und *obere* mit der angegebenen *Toleranz*.

**MATH**

9:fnInt(

## Folge(

**Folge(Ausdruck,Variable,Anfang,Ende[,Steigerung])**

**2nd** **[LIST]**

Ergibt eine Liste durch Auswertung des *Ausdrucks* in Bezug auf die *Variable*, vom *Anfang* bis zum *Ende* nach *Steigerung*.

**OPS**

5:Folge(

## Folge(n)

**Folge(n)**

**+ [MODE]**

Im Folge-Modus wird mit **Folge(n)** der Folge-Editortyp zur Eingabe von Folge-Funktionen, u, v, oder w, als Funktion der unabhängigen Variablen *n* eingestellt. Eine Einstellung ist auch über den Y= Editor im **Folge-Modus** möglich.

**Folge(n)**

## Folge(n+1)

**Folge(n+1)**

**+ [MODE]**

Im Folge-Modus wird mit **Folge(n+1)** der Folge-Editortyp zur Eingabe von Folge-Funktionen, u, v, oder w, als Funktion der unabhängigen Variablen *n+1* eingestellt. Eine Einstellung ist auch über den Y= Editor im **Folge-Modus** möglich.

**Folge(n+1)**

## Folge(n+2)

**Folge(n+2)**

**+ [MODE]**

## Folge( $n+2$ )

Im Folge-Modus wird mit **Folge( $n+2$ )** der Folge-Editortyp zur Eingabe von Folge-Funktionen, u, v, oder w, als Funktion der unabhängigen Variablen  $n+2$  eingestellt. Eine Einstellung ist auch über den Y= Editor im **Folge-Modus** möglich.

**Folge( $n+2$ )**

```
NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG
Send(
seq(
Seq
▶SEQ(n) Type
SEQ(n+1) Type
SEQ(n+2) Type
Sequential
setDate(
setDtFmt(
```

**Hinweis:** „Eing“ ist NICHT in der TIC CE PE Syntax enthalten

Auf dem Gerät wird „Eing“ nicht eingefügt und verhält sich so wie z. B. „DEZ Antwort“ auf dem Gerät, wobei „Antworten“ im [Katalog] angezeigt, jedoch nicht eingefügt wird.

## Folge

**Folge**

+ **[MODE]**

Stellt den Folge-Grafikmodus ein.

**Folge**

## For(

**:For(Variable,Anfang,Ende  
[,Steigerung]):Befehle:Ende:Befehle**

+ **[PRGM]**

**SATR**

**4:For(**

Führt *Befehle* bis **Ende** aus und steigert dabei *Variable* von *Anfang* um *Steigerung*, bis *Variable*>*Ende*.

▶ F ◀▶ D

▶ F ◀▶ D

**[ALPHA]** [F-1]

**4: ▶ F ◀▶ D**

oder

**[MATH]**

**NUM**

**B: ▶ F ◀▶ D**

**[MATH]**

**BRUCH**

**3: ▶ F ◀▶ D**

## Fcdf(

**Fcdf**(*untereGrenze, obereGrenze, Zähler df, Nenner df*)

**2nd** **[DISTR]**

Berechnet die  $F$ -Verteilungswahrscheinlichkeit zwischen *untereGrenze* und *obereGrenze* für die angegebenen *Zähler-df* (Freiheitsgrade) und *Nenner-df*.

**VERTLG**  
**0: Fcdf(**

## Fpdf(

**Fpdf**(*x, Zähler df, Nenner df*)

**2nd** **[DISTR]**

Berechnet die  $F$ -Verteilungswahrscheinlichkeit zwischen *untereGrenze* und *obereGrenze* für die angegebenen *Zähler-df* (Freiheitsgrade) und *Nenner-df*.

**VERTLG**  
**9: Fpdf(**

## fTeil(

**fTeil**(*Wert*)

**[MATH]**

Liefert den Bruchanteil bzw. die Bruchanteile einer reellen oder komplexen Zahl, eines Ausdrucks, einer Liste oder einer Matrix.

**NUM**  
**4:fTeil(**

## Fülle(

**Fülle**(*Wert, Matrixname*)

**2nd** **[MATRIX]**

Speichert *Wert* zu jedem Element in *Matrixname*.

**MATH**  
**4:Fülle(**

## Fülle(

**Fülle**(*Wert, Listenname*)

**2nd** **[LIST]**

Speichert *Wert* zu jedem Element in *Listenname*.

**OPS**  
**4:Fülle(**

## Funk

**Funk**

**+ [MODE]**

Stellt den Funktionsgraphen-Modus ein.

**Funk**

## G

### ganzzTeil(

**ganzzTeil**(*Wert*)

**[MATH]**

Liefert den ganzzahligen Teil einer reellen oder komplexen Zahl, eines Ausdrucks, einer Liste oder einer Matrix.

**NUM**  
**3:ganzzTeil(**

## GDBLaden

### GDBLaden $\overline{n}$

**[2nd] [DRAW]**

Stellt alle Einstellungen wieder her, die in der Grafikdatenbank-Variablen **GDB $\overline{n}$**  gespeichert sind.

**STO**

**4:GDBLaden**

## GDBSpeichern

### GDBSpeichern $\overline{n}$

**[2nd] [DRAW]**

Speichert den aktuellen Graphen in Datenbank **GDB $\overline{n}$** .

**STO**

**3:GDBSpeichern**

## geometcdf(

### geometcdf( $p,x$ )

**[2nd] [DISTR]**

Berechnet die kumulative Wahrscheinlichkeit an  $x$ , die Anzahl der Einzelversuche, bis der erste Erfolg eingetreten ist, für die diskrete geometrische Verteilung mit der vorgegebenen Erfolgswahrscheinlichkeit  $p$ .

**VERTLG**

**F:geometcdf(**

## geometpdf(

### geometpdf( $p,x$ )

**[2nd] [DISTR]**

Berechnet die Wahrscheinlichkeit an  $x$ , die Anzahl der Einzelversuche, bis der erste Erfolg eingetreten ist, für die diskrete geometrische Verteilung mit der vorgegebenen Erfolgswahrscheinlichkeit  $p$ .

**VERTLG**

**E:geometpdf(**

## Get(

### Get(*Variable*)

**+ [PRGM]**

Ruft einen Wert von einem angeschlossenen TI-Innovator™ Hub ab und speichert die Daten auf dem empfangenden CE-Taschenrechner in einer Variablen.

**I/O**

**Hinweis:** Siehe auch **Sende(** und **eval(**

**A:Get**

## Get(

### Get(*Variable*)

**TI-Innovator™  
Hub**

**+ [PRGM]**

Ruft einen Wert von einem angeschlossenen TI-Innovator™ Hub ab und speichert die Daten auf dem empfangenden CE-Taschenrechner in einer Variablen.

**HUB**

**Hinweis:** Siehe auch **Sende(** und **eval(**

**5:Get**

## GetCalc(

**GetCalc**(*Variable*[,*portflag*])

+ [PRGM]  
I/O  
0:GetCalc(

Holt Inhalte von *Variable* eines anderen TI-84 Plus CE und speichert sie in *Variable* des empfangenden TI-84 Plus CE. Standardmäßig verwendet der TI-84 Plus CE den USB-Port, sofern dort ein Gerät angeschlossen ist. Ist das USB-Kabel nicht angeschlossen, wird der I/O-Port verwendet.

*portflag*=0 USB-Port verwenden, wenn angeschlossen;

*portflag*=1 USB-Port verwenden;

*portflag*=2 I/O-Port verwenden.(Wird ignoriert, wenn das Programm auf dem TI-84 Plus CE ausgeführt wird.)

## getKey

**getKey**

+ [PRGM]  
I/O  
7:getKey

Liefert den Tastencode für den aktuellen Tastendruck bzw. 0, wenn keine Tasten gedrückt wird.

## ggT(

**ggT**(*WertA*,*WertB*)

[MATH]  
NUM  
9:ggT(

Liefert den größten gemeinsamen Teiler von *WertA* und *WertB*, dabei kann es sich um reelle Zahlen oder Listen handeln.

## GitterAus

**GitterAus**

+ [2nd] [FORMAT]  
GitterAus

Schaltet das Gitterformat aus.

## GitterLinie

**GitterLinie** [*Farbe#*]

+ [2nd]  
[FORMAT]  
GitterLinie

Schaltet Gitterlinien im Graphenbereich in der angegebenen Farbe ein.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## GitterPkt

**GitterPkt** [*Farbe#*]

+ [2nd]  
[FORMAT]  
GitterPkt

Schaltet Gitterpunkte im Graphenbereich in der angegebenen Farbe ein.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.



## Goto

### Goto*Bezeichnung*

Überträgt die Steuerung an *Bezeichnung*.

+ **[PRGM]**  
SATRG  
0:Goto

## Grad

### Grad

Stellt den Winkelmodus Grad ein.

+ **[MODE]**  
Grad

## GraphColor(

### GraphColor(*Funktion#*,*Farbe#*)

Stellt die Farbe für *Funktion#* ein.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

+ **[PRGM]**  
SATRG  
H:GraphColor(

## GraphStyle(

### GraphStyle(*Funktion#*,*Diagrammstil#*)

Stellt einen *Diagrammstil* für *Funktion#* ein.

+ **[PRGM]**  
SATRG  
H:GraphStyle(

## G-T

### G-T

Stellt den vertikal geteilten Graph-Tabelle-Bildschirmmodus ein.

+ **[MODE]**  
GRAPH-  
TABELLE

## H

## HintergrundAus

### HintergrundAus

Schaltet das Hintergrundbild im Graphenbereich aus.

+ **[2nd] [DRAW]**  
HINTERGRUND  
2:HintergrundAus:

## HintergrundEin

### HintergrundEin n

Zeigt ein Menü für die Hintergrundbild-Variable n (Bild#n) im Graphenbereich an.

+ **[2nd] [DRAW]**  
HINTERGRUND  
1:HintergrundEin

## Histogramm

**Histogramm** Plot#(*Typ,XListe*,[,*Häufigkeitsliste, Farbe*#])

Wird als „Typ“-Argument im Befehl verwendet

# ergibt Plot1, Plot2 oder Plot3.

+ [2nd]  
[stat plot]  
TYP

## holDatum

**holDatum**

[2nd] [CATALOG]

holDatum

Gibt eine dem aktuellen Datumswert der Uhr entsprechende Liste zurück.  
Die Liste weist das Format *{Jahr, Monat, Tag}* auf.

## holDtFmt

**holDtFmt**

[2nd]

[CATALOG]

holDtFmt

Gibt eine Ganzzahl zurück, die das im Gerät aktivierte Datumsformat darstellt.

1 = M/T/J  
2 = T/M/J  
3 = J/M/T

## holDtStr(

**holDtStr**(*Ganzzahl*)

[2nd]

[CATALOG]

holDtStr(

Liefert das aktuelle Datum in dem mit *Ganzzahl* vorgegebenen Format, wobei gilt:

1 = M/T/J  
2 = T/M/J  
3 = J/M/T

## holeZeit

**holeZeit**

[2nd] [CATALOG]

holeZeit

Gibt eine dem aktuellen Uhrzeitwert der Uhr entsprechende Liste zurück.  
Die Liste weist das Format *{Stunde, Minute, Sekunde}* auf. Die Uhrzeit wird im 24-Stunden-Format zurückgegeben.

## holZtFmt

**holZtFmt**

[2nd]

[CATALOG]

holZtFmt

Gibt eine Ganzzahl zurück, die das im Gerät aktivierte Uhrzeitformat darstellt.

12 = 12-Stunden-Format  
24 = 24-Stunden-Format

## holZtStr(

**holZtStr**(*Ganzzahl*)

[2nd]

[CATALOG]

holZtStr(

Liefert die aktuelle Uhrzeit in dem mit *Ganzzahl* vorgegebenen Format, wobei gilt:

12 = 12-Stunden-Format

24 = 24-Stunden-Format

## Horiz

**Horiz**

+ [MODE]

Horiz

Stellt die Anzeige auf horizontal geteilten Bildschirm.

## Horizontal

**Horizontal** *y* [,*Farbe*#,*Linestyle*#]

[2nd] [DRAW]

ZEICHNEN

Zeichnet eine horizontale Linie an *y* in einer angegebenen

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

3:Horizontal

Linestyle #: 1-4.

/

## *i*

*i*

[2nd] [*i*]

Gibt die komplexe Zahl *i* zurück.

## If

**If** *Bedingung*:*BefehlA*:*Befehle*

+ [PRGM]

if *Bedingung* = 0 (falsch), überspringt *BefehlA*.

SATRG

1:if

## If

**Then**

**Ende**

**If**:*Bedingung***Then**:*Befehle***Ende**:*Befehle*

+ [PRGM]

Führt *Befehle* von **Then** bis **Ende** aus, wenn *Bedingung* = 1 (wahr).

SATRG

2:Then

If  
Then  
Else  
Ende

If:*Bedingung*Then:*Befehle*Else:*Befehle*Ende:*Befehle*

+ [PRGM]  
SATRG  
3:Else

Führt *Befehle* von **Then** bis **Else** aus, wenn *Bedingung* = 1 (wahr), und von **Else** bis **Ende**, wenn *Bedingung* = 0 (falsch).

imag(

imag(*Wert*)

[MATH]  
KMPLX  
3:imag(

Liefert den imaginären (nicht-reellen) Teil einer komplexen Zahl oder einer Liste mit komplexen Zahlen.

IndpntAbfr

IndpntAbfr

+ [2nd]  
[TBLSET]  
Indpnt:  
Abfrage

Stellt eine Tabelle zur Abfrage unabhängiger Variablenwerte ein.

Input

Input

+ [PRGM]  
I/O  
2:Input

Zeigt einen Graphen an.

Input

Input [*Variable*]

+ [PRGM]  
I/O  
2:Input

Input ["*Text*",*Variable*]

Fragt einen Wert zum Speichern in *Variable* ab.

Input

Input [*Strn*,*Variable*]

+ [PRGM]  
I/O  
2:Input

Zeigt *Strn* an und speichert den eingegebenen Wert in *Variable*.

inString(

inString(*String*,*subString*[,*Start*])

[2nd] [CATALOG]  
inString(

Gibt an, an welcher Position im *String* das erste Zeichen des *Teilstrings*

## inString(

steht, beginnend bei *Start*.

## int(

int(*Wert*)

Liefert die größte Ganzzahl einer reellen oder komplexen Zahl, eines Ausdrucks, einer Liste oder einer Matrix.

**MATH**

**NUM**

**5:int(**

## $\Sigma$ Int(

$\Sigma$ Int(*pmt1*,*pmt2*[,*WertRunden*])

Berechnet die Summe, gerundet auf *WertRunden*, des Zinsbetrags zwischen *pmt1* und *pmt2* für einen Amortisationszeitplan.

**APPS**

**1:FINANZEN  
BERECHNEN**

**A:  $\Sigma$ Int(**

## invBinom(

invBinom(*Fläche*,*Versuch*,*p*)

Die inverse kumulierte Binomialverteilungsfunktion ergibt die Mindestanzahl an Erfolgen, sodass gilt: kumulative Wahrscheinlichkeit für diese Mindestanzahl Erfolge  $\geq$  gegebene kumulative Wahrscheinlichkeit (Fläche). Sollten weitere Informationen benötigt werden, ermitteln Sie für eine vollständige Analyse zusätzlich binomcdf für das Ergebnis aus invBinom(, wie nachstehend gezeigt.

**2nd** **DISTR**

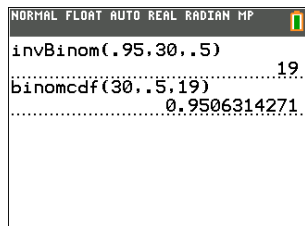
**VERTLG**

**C:invBinom(**

### Einzelheiten:

Eine Münze wird 30 Mal geworfen. Wie oft muss die Münze mit dem Kopf nach oben landen, damit die kumulative Wahrscheinlichkeit für die Häufigkeit für "Kopf" mindestens 0,95 beträgt?

Die Ergebnisse auf dem Bildschirm zeigen zunächst, dass die Mindestanzahl an Erfolgen bei 19 liegt, um die vorgegebene kumulative Wahrscheinlichkeit von mindestens 0,95 zu erreichen. Als nächstes wird die kumulative Wahrscheinlichkeit für bis zu 19 Erfolge mit binomcdf( berechnet; sie liegt bei ca. 0,9506314271 und erfüllt somit die Kriterien, da  $0,9506314271 \geq 0,95$ .



### Alternative Methode:

Geben Sie  $Y1=binomcdf(30,0.5,X)$  ein und verwenden Sie die Wertetabelle (beginnend bei 0, Steigerung um 1), um herauszufinden, wann die kumulative Wahrscheinlichkeit bei oder knapp über der

## inBinom(

vorgegebenen kumulativen Wahrscheinlichkeit liegt. Es werden alle Werte gezeigt, um Entscheidungen treffen zu können. Für dieses Beispiel müssen Sie in der Tabelle die kumulative Wahrscheinlichkeit finden, die knapp über 0,95 liegt. Auch hier ist das Ergebnis, dass der Kopf 19 Mal zu sehen sein muss.

The calculator screen shows the function editor for `inBinomcdf(30, 0.5, X)`. Below the editor is a table with the following data:

X	Y1			
13	0.2923			
14	0.4278			
15	0.5722			
16	0.7077			
17	0.8192			
18	0.8998			
19	0.9506			
20	0.9786			
21	0.9919			
22	0.9974			
23	0.9993			

Below the table, the value `Y1=0.9506314270685` is displayed.

## invNorm(

`invNorm(Fläche[, $\mu$ , $\sigma$ ,Schiefe])`

**2nd** [DISTR]

Schiefe [Katalog]: **LINKS**, **MITTELPUNKT**, **RECHTS**

VERTLG

Berechnet die inverse kumulative Normalverteilungsfunktion für eine bestimmte Fläche unter der Normalverteilungskurve, die durch  $\mu$  und  $\sigma$  festgelegt ist. Das optionale Argument „Schiefe“ für reell  $a$  kann **LINKS** ( $-\infty, a$ ), **MITTELPUNKT** [ $-a, a$ ] oder **RECHTS** ( $a, \infty$ ) sein.

**3:invNorm(**

Die Token (Kurzzeichen) **LINKS**, **MITTELPUNKT** und **RECHTS** finden Sie im [Katalog].

## LINKS

### LINKS

**2nd** [CATALOG]  
LINKS

**LINKS** ist ein Schiefe-Argument für den Befehl `invNorm(`, wobei das optionale Argument „Schiefe“ für reell  $a$  **LINKS** ( $-\infty, a$ ), **MITTELPUNKT** [ $-a, a$ ] oder **RECHTS** ( $a, \infty$ ) sein kann.

Siehe auch `invNorm(`.

## RECHTS

### RECHTS

**2nd** [CATALOG]

## RECHTS

**RECHTS** ist ein Schiefe-Argument für den Befehl **invNorm(**, wobei das optionale Argument „Schiefe“ für reell a **LINKS** ( $-\infty, -a$ ), **MITTELPUNKT**  $[-a, a]$  oder **RECHTS** ( $a, \infty$ ) sein kann.

**RECHTS**

Siehe auch **invNorm(**.

## MITTELPUNKT

### MITTELPUNKT

**2nd** [CATALOG]  
**MITTELPUNKT**

**MITTELPUNKT** ist ein Schiefe-Argument für den Befehl **invNorm(**, wobei das optionale Argument „Schiefe“ für reell a **LINKS** ( $-\infty, -a$ ), **MITTELPUNKT**  $[-a, a]$  oder **RECHTS** ( $a, \infty$ ) sein kann.

Siehe auch **invNorm(**.

## LINKS

## RECHTS

## MITTELPUNKT

```
NORMAL FLOAT AUTO REAL RADIAN MP  a  |
CATALOG
LabelOff
LabelOn
Lbl1
lcm(
LEFT
length(
Line(
LinReg(a+bx)
LinReg(ax+b)
```

```
NORMAL FLOAT AUTO REAL RADIAN MP  a  |
CATALOG
ref(
remainder(
Repeat
Return
RIGHT
round(
row(
row+(
```

```
NORMAL FLOAT AUTO REAL RADIAN MP  a  |
CATALOG
binomcdf(
binompdf(
BorderColor
BoxPlot
CENTER
checkTmr(
χ²cdf(
χ²pdf(
χ²-Test(
```

## invT(

### invT(Fläche, df)

Berechnet die inverse kumulative Student-t-Wahrscheinlichkeitsfunktion, die durch Freiheitsgrade, df, für eine bestimmte Fläche unter der Kurve festgelegt ist.

**2nd** [DISTR]  
**VERTLG**  
**4:invT(**

## irr(

### irr(CF0, CFListe[, CFFreq])

Ergibt den Zinssatz, bei dem der Nettobarwert der Cashflows null ergibt.

**APPS**  
**1:Finanzen**  
**BERECHNEN**  
**8:irr(**

## IS>(

### :IS>(Variable, Wert)

### :BefehlA

### :Befehle

**PRGM**  
**SATRG**  
**A:IS>(**

Steigert *Variable* um 1; überspringt *BefehlA*, wenn *Variable*>*Wert*.

## istUhrEin

### istUhrEin

**2nd** [CATALOG]

istUhrEin

Zeigt an, ob die Uhr EIN oder AUS ist. Liefert den Wert 1, wenn sich die Uhr im Modus EIN befindet. Liefert den Wert 0, wenn sich die Uhr im Modus AUS befindet.

## K

### ►Kartes

#### komplexer Wert ►Kartes

[MATH]

KMPLX

Zeigt einen *komplexen Wert* oder eine Liste im kartesischen Format an.

6: ► Kartes

### Kästchengrafik

#### Kästchengrafik#(Typ,XListe,[,Häufigkeitsliste, Farbe#])

+ **2nd**

[stat plot]

Definiert Plot# (1, 2 oder 3) vom Typ

TYP

### kgV(

#### kgV(WertA,WertB)

[MATH]

NUM

Liefert das kleinste gemeinsame Vielfache von *WertA* und *WertB*, dabei kann es sich um reelle Zahlen oder Listen handeln.

8:kgV(

### KLASSISCH

#### KLASSISCH

[MODE]

Zeigt Eingaben und Ausgaben in einer Zeile an, z. B.  $1/2 + 3/4$ .

KLASSISCH

### konj(

#### konj(Wert)

[MATH]

KMPLX

Ergibt die komplexe Konjugierte einer komplexen Zahl oder einer Liste komplexer Zahlen.

1:konj(

### KoordAus

#### KoordAus

+ **2nd**

[FORMAT]

Schaltet die Anzeige des Cursor-Koordinatenwerts aus.

KoordAus



## KoordEin

### KoordEin

Schaltet die Anzeige des Cursor-Koordinatenwerts ein.

†  
[2nd]  
[FORMAT]  
KoordEin

## Kreis(

### Kreis(*X,Y,Radius[,Farbe#,Linienart#]*)

Zeichnet einen Kreis mit Mittelpunkt (*X,Y*) und *Radius* mit der angegebenen

Farbe#: 10 - 24 oder dem aus [Variablen] FARBE eingefügten Namen der Farbe.

Linestyle#: 1-2.

[2nd] [DRAW]  
ZEICHNEN  
9:Kreis(

## kumSum(

### kumSum(*Liste*)

Gibt eine Liste der kumulierten Summen der Elemente aus *Liste* zurück, beginnend beim ersten Element.

[2nd] [LIST]  
OPS  
6:kumSum(

## kumSum(

### kumSum(*Matrix*)

Gibt eine Matrix der kumulierten Summen der Elemente aus *Matrix* zurück. Jedes Element in der zurückgegebenen Matrix ist die kumulierte Summe einer *Matrix*-Spalte von oben nach unten.

[2nd] [MATRIX]  
MATH  
0:kumSum(

## L

## L

### L*Listenname*

Kennzeichnet die nächsten 1 bis 5 Zeichen als einen vom Benutzer erstellten Listennamen.

[2nd] [LIST]  
OPS  
B: L

## Länge(

### Länge(*String*)

Liefert die Anzahl der Zeichen in *String*.

[2nd]  
[CATALOG]  
Länge(

## Lbl

### Lbl *Bezeichnung*

Erstellt eine *Bezeichnung* mit einem oder zwei Zeichen.

+ **PRGM**  
**SATRG**  
**9:Lbl**

## Linie(

### Linie(*X1,Y1,X2,Y2*[*Löschen#*,*Farbe#*,*Linestyle#*])

Zeichnet eine Linie von (*X1,Y1*) nach (*X2,Y2*) mit folgenden Optionen:  
Löschen #: 1,0, Farbe #: 10-24 und Linestyle #: 1-4.

**2nd** **[DRAW]**  
**ZEICHNEN**  
**2:Linie(**

## Linie(

### Linie(*X1,Y1,X2,Y2,0*[*linie#*])

Löscht eine Linie (Löschen #: 1,0) von (*X1,Y1*) nach (*X2,Y2*).

**2nd** **[DRAW]**  
**ZEICHNEN**  
**2:Linie(**

## LinReg(a+bx)

### LinReg

#### [*XListenname*,*YListenname*,*Häufigkeitsliste*,*regequ*]

Passt ein lineares Regressionsmodell an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

**[STAT]**  
**BERECHNEN**  
**8:LinReg**  
**(a+bx)**

## LinReg(ax+b)

### LinReg(ax+b)

#### [*XListenname*,*YListenname*,*Häufigkeitsliste*,*regequ*]

Passt ein lineares Regressionsmodell an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

**[STAT]**  
**BERECHNEN**  
**4:LinReg**  
**(ax+b)**

## LinRegTInt

### LinRegTInt [

#### *Xlistenname*,*YListenname*,*Häufigkeitsliste*,*Konfidenznive* *au*, *regequ*]

Führt eine lineare Regression aus und berechnet das t-Konfidenzintervall für den Steigungskoeffizienten b.

+ **[STAT]**  
**TESTS**  
**G:LinRegTIn**  
**t**

## LinRegTTest

### LinRegTTest

+ **[STAT]**

## LinRegTTest

[  
*X*Listenname  
,*Y*Listenname,*Häufigkeitsliste*,*Alternative*,*regequ*]

TESTS  
F:LinRegTTest

Führt eine lineare Regression und einen *t*-Test aus. *Alternative*=-1 ist <; *Alternative*=0 ist ; *Alternative*=1 ist >.

## ΔListe(

ΔListe(*Liste*)

[2nd] [LIST]

Ergibt eine Liste mit den Differenzen der aufeinander folgenden Elemente in *Liste*.

OPS  
7: Δ Liste(

## Liste▶matr(

Liste▶matr(*Listenname*<sub>1</sub>,...,*Listenname*<sub>*n*</sub>,*Matrixname*)

[2nd] [LIST]

Füllt *Matrixname* Spalte für Spalte mit den Elementen jedes angegebenen *Listennamens*.

OPS  
0:Liste ▶  
matr(

## In(

In(*Wert*)

[LN]

Liefert den natürlichen Logarithmus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## LnReg

LnReg  
[*X*Listenname,*Y*Listenname,*Häufigkeitsliste*,*regequ*]

[STAT]  
BERECHNEN  
9:LnReg

Passt ein Modell für logarithmische Regression an *X*Listenname und *Y*Listenname mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

## LöBild

LöBild

[2nd] [DRAW]

Löscht alle gezeichneten Elemente von einem Graphen oder Diagramm.

ZEICHNEN  
1:LöBild

## log(

log(*Wert*)

[LOG]

Liefert den Logarithmus einer reellen oder komplexen Zahl, eines

## log(

Ausdrucks oder einer Liste.

## logBASE(

logBASE(*Wert*, *Basis*)

**MATH**

Gibt den Logarithmus eines bestimmten Wertes bezüglich einer angegebenen Basis zurück: logBASE(*Wert*, *Basis*).

**A: logBASE**

## Logistisch

### Logistisch

[*XListenname*, *YListenname*, *Häufigkeitsliste*, *regequ*]

**BERECHNEN**

**B:Logistisch**

Passt ein logistisches Regressionsmodell an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

## LöListe

LöListe*Listenname1* [, *Listenname2*, ..., *Listenname n*]

**STAT**

Stellt die Dimension eines oder mehrerer Listenamen auf 0.

**BEARBEITEN**

**4:LöListe**

## Löse(

Löse(*Ausdruck*, *Variable*, *Schätzung*, {*untere*, *obere*})

† **MATH**

**MATH**

**0:Löse(**

Löst einen *Ausdruck* mit einer *Variablen*, wobei der erste *Lösungsvorschlag* sowie die *untere* und *obere* Grenze, innerhalb derer die Lösung gesucht wird, gegeben sind.

# M

## Manuell anp

**Manuell anp**[*Eqname, Farbe#, Linestyle#*]

Passt eine lineare Gleichung mit der angegebenen Farbe und Linienart in ein Streudiagramm ein.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

Linestyle #: 1-4.

**[STAT]**

**BERECHNEN**  
**D:Manuell**  
**anp**

## MarkeAus

**MarkeAus**

Schaltet die Achsenbezeichnungen aus.

+ **[2nd]** **[FORMAT]**

**MarkeAus**

## MarkeEin

**MarkeEin**

Schaltet die Achsenbezeichnungen ein.

+ **[2nd]**

**[FORMAT]**

**MarkeEin**

## MATHPRINT

**MATHPRINT**

Zeigt die meisten Eingaben und Antworten so an, wie sie in Lehrbüchern dargestellt werden, z. B.  $\frac{1}{2} + \frac{3}{4}$ .

**[MODE]**

**MATHPRINT**

## Matr>Liste(

**Matr>Liste**(*Matrix, Listenname A, ..., Listenname n*)

Füllt jeden *Listennamen* mit Elementen aus jeder Spalte in *Matrix*.

**[2nd]** **[LIST]**

**OPS**

**A:Matr ▶**

**Liste(**

## Matr>Liste(

**Matr>Liste**(*Matrix, Spalte#, Listenname*)

Füllt einen *Listennamen* mit Elementen aus einer bestimmten *Spalte#* in *Matrix*.

**[2nd]** **[LIST]**

**OPS**

**A:Matr ▶**

**Liste(**

**max(****max(WertA,WertB)**Liefert den größeren Wert von *WertA* und *WertB*.**(MATH)**  
NUM  
7:max(**max(****max(Liste)**Liefert den größeren Wert von *WertA* und *WertB*.**(MATH)**  
NUM  
7:max(**max(****max(Liste)**Liefert den größten reellen oder komplexen Wert in *Liste*.**(2nd) (LIST)**  
MATH  
2:max(**max(****max(ListeA,ListeB)**Liefert eine reelle oder komplexe Liste des jeweils größeren Elementepaars in *ListeA* und *ListeB*.**(2nd) (LIST)**  
MATH  
2:max(**max(****max(Wert,Liste)**Liefert eine reelle oder komplexe Liste des jeweils größeren *Werts* oder *Listenelements*.**(2nd) (LIST)**  
MATH  
2:max(**Median(****Median(Liste[,Häufigkeitsliste])**Ergibt den Median der *Liste* mit der Häufigkeit *Häufigkeitsliste*.**(2nd) (LIST)**  
MATH  
4:Median(**Med-Med****Med-Med****[XListenname,YListenname,Häufigkeitsliste,regequ]**Passt ein Median-Median-Modell an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.**(STAT)**  
BERECHNEN  
3:Med-Med

## Menu(

**Menu**("Titel","Text1",*Bezeichnung1*  
[,...,"Text7",*Bezeichnung7*])

Erzeugt bei der Ausführung des Programms ein Menü mit bis zu sieben Elementen.

+ **PRGM**  
**SATRG**  
**C:Menu(**

## min(

**min**(*WertA*,*WertB*)

Liefert den kleineren Wert von *WertA* und *WertB*.

**MATH**  
**NUM**  
**6:min(**

## min(

**min**(*Liste*)

Liefert den kleinsten reellen oder komplexen Wert in *Liste*.

**2nd** **[LIST]**  
**MATH**  
**1:min(**

## min(

**min**(*ListeA*,*ListeB*)

Liefert eine reelle oder komplexe Liste des jeweils kleineren Elemente-Paars in *ListeA* und *ListeB*.

**2nd** **[LIST]**  
**MATH**  
**1:min(**

## min(

**min**(*Wert*,*Liste*)

Liefert eine reelle oder komplexe Liste des jeweils kleineren *Werts* oder *Listenelements*.

**2nd** **[LIST]**  
**MATH**  
**1:min(**

## Mittelwert(

**Mittelwert**(*Liste*[,*Häufigkeitsliste*])

Ergibt den Mittelwert der *Liste* mit der Häufigkeit *Häufigkeitsliste*.

**2nd** **[LIST]**  
**MATH**  
**3:Mittelwert(**

## ModKästchengrafik

**ModKästchengrafik** Plot#(*Typ*,*XListe*,  
[,*Häufigkeitsliste*,*Farbe*#])

Wird als „Typ“-Argument im Befehl verwendet.

# ergibt Plot1, Plot2 oder Plot3.

+ **2nd**  
**[stat plot]**  
**TYP**

## MüllSammeln

### MüllSammeln

**2nd** [CATALOG]

MüllSammeln

Zeigt das Menü „Müll sammeln“ an, mit dem Sie ungenutzten Archivspeicher freigeben können.

## N

$n$

$n$

**MATH**

PRB

3://

**nCr**

*WertA* **nCr** *WertB*

**MATH**

WAHRS

3:nCr

Liefert die Anzahl der Kombinationen von *WertA* eines bestimmten *WertB* zu einem Zeitpunkt.

**nCr**

*Wert* **nCr** *Liste*

**MATH**

WAHRS

3:nCr

Liefert eine Liste der Kombinationen von *Wert* mit jedem Element in *Liste*.

**nCr**

*Liste* **nCr** *Wert*

**MATH**

WAHRS

3:nCr

Liefert eine Liste der Kombinationen von jedem Element in *Liste* mit *Wert*.

**nCr**

*ListeA* **nCr** *ListeB*

**MATH**

WAHRS

3:nCr

Liefert eine Liste der Kombinationen von jedem Element in *ListeA* mit jedem Element in *ListeB*.

**n/d**

**n/d**

**ALPHA** [F1]

Zeigt Ergebnisse als einfachen Bruch an.

1: n/d

oder



n/d

**MATH**

NUM  
D: n/d  
oder

**MATH**

BRUCH  
1:n/d

nDeriv(

nDeriv(*Ausdruck*, *Variable*, *Wert*[, $\epsilon$ ])

**MATH**

MATH  
8:nDeriv(

Wenn der Befehl im klassischen Modus verwendet wird, liefert er eine approximierete numerische Ableitung von *Ausdruck* in Bezug auf *Variable* bei *Wert* mit der spezifischen Toleranz  $\epsilon$ .

Im MathPrint-Modus verwendet die Vorlage der numerischen Ableitung die Standardtoleranz  $\epsilon$ .

► n/d ◄► Un/d

► n/d ◄► Un/d

**ALPHA** [F-1]

3: ► n/d ◄►  
Un/d

Konvertiert die Ergebnisse von einem Bruch in eine gemischte Zahl bzw. von einer gemischten Zahl in einen Bruch, sofern möglich.

oder

**MATH**

NUM

A: ► n/d ◄◄  
Un/d

oder

**MATH**

BRUCH

4: ► n/d ◄  
►Un/d

Netz

Netz

+ **2nd**  
**FORMAT**

Stellt die Verfolgung von Sequenz-Graphen als Netz ein.

## Netz

Netz

### nicht(

nicht(*Wert*)

Liefert den Wert **0**, wenn *Wert* **0** ist. *Wert* kann eine reelle Zahl, ein Ausdruck oder eine Liste sein.

**[2nd][TEST]**  
LOGIK  
4:nicht(

### ►Nom(

►Nom(*Effektivzins*,  
*Verzinsungsperioden*)

Berechnet den Nominalzinssatz.

**[APPS]**  
1:Finanzen  
BERECHNEN  
B: ► Nom(

### Normal

Normal

Stellt den normalen Anzeigemodus ein.

+ **[MODE]**  
Normal

### normalcdf(

normalcdf(*untereGrenze*,*obereGrenze* [,  $\mu$ ,  $\sigma$ ])

Berechnet die Normalverteilungswahrscheinlichkeit zwischen *untereGrenze* und *obereGrenze* für die angegebenen  $\mu$  und  $\sigma$ .

**[2nd][DISTR]**  
VERTLG  
2:normalcdf(

### normalpdf(

normalpdf( $x$  [,  $\mu$ ,  $\sigma$ ])

Berechnet die Wahrscheinlichkeitsdichtefunktion für die Normalverteilung an einem bestimmten Wert  $x$  für die vorgegebenen  $\mu$  und  $\sigma$ .

**[2nd][DISTR]**  
VERTLG  
1:normalpdf(

### NormWahrschdiagr

NormWahrschdiagr Plot#(*Typ*,*XListe*,  
[,*Häufigkeitsliste*,*Farbe*#])

Wird als „Typ“-Argument im Befehl verwendet

# ergibt Plot1, Plot2 oder Plot3.

+ **[2nd]**  
**[stat plot]**  
TYP

**nPr***WertA nPr WertB***MATH**

Liefert die Anzahl der Permutationen von *WertA* eines bestimmten *WertB* zu einem Zeitpunkt.

**WAHRS**  
**2:nPr**
**nPr***Wert nPr Liste***MATH**

Liefert eine Liste der Permutationen von *Wert* mit jedem Element in *Liste*.

**WAHRS**  
**2:nPr**
**nPr***Liste nPr Wert***MATH**

Liefert eine Liste der Permutationen von jedem Element in *Liste* mit *Wert*.

**WAHRS**  
**2:nPr**
**nPr***ListeA nPr ListeB***MATH**

Liefert eine Liste der Permutationen von jedem Element in *ListeA* mit jedem Element in *ListeB*.

**WAHRS**  
**2:nPr**
**npv(****npv(Zinssatz,CF0,CFListe[,CFFreq])****APPS**

Berechnet die Summe der Barwerte für die Bar-Zuflüsse und -Abflüsse.

**1:Finanzen**  
**BERECHNEN**  
**7:npv(**
**O****oder***WertAoderWertB***2nd** **[TEST]**

Liefert den Wert 1, wenn *WertA* oder *WertB* 0 ist. *WertA* und *WertB* können reelle Zahlen, Ausdrücke oder Listen sein.

**LOGIK**  
**2:oder**
**OpenLib(****OpenLib(****+** **PRGM**

Erweitert TI-Basic. (Nicht verfügbar.)

**SATRG**  
**J:OpenLib(**

## Output(

**Output**(*Zeile*,*Spalte*,"*Text*")

+ [PRGM]

I/O

Zeigt *Text* beginnend in einer festgelegten *Zeile* und *Spalte* des Hauptbildschirms an.

6:Output(

## Output(

**Output**(*Zeile*,*Spalte*,"*Wert*")

+ [PRGM]

I/O

Zeigt *Wert* beginnend in einer festgelegten *Zeile* und *Spalte* des Hauptbildschirms an.

6:Output(

## P

### Param

**Param**

+ [MODE]

Par

Stellt den Modus für die parametrische Graphdarstellung ein.

### Pause

**Pause**

+ [PRGM]

SATRG

Hält die Ausführung eines Programms an, bis Sie [ENTER] drücken.

8:Pause

### Pause

**Pause** [*Wert*]

+ [PRGM]

SATRG

Zeigt *Wert* an; hält die Ausführung eines Programms an, bis Sie [ENTER] drücken.

8:Pause

### Pause

**Pause** [*Wert*, *Zeit*]

+ [PRGM]

CTL

Zeigt den Wert im aktuellen Hauptbildschirm an, die Ausführung des Programms wird nach Ablauf der angegebenen Zeit fortgesetzt. Um nur die Zeit zu verwenden, geben Sie Pause „“, *Zeit* ein, wobei der Wert ein Leerstring ist. Die Zeit wird in Sekunden eingegeben.

8:Pause

**Pause** *Wert*, *Zeit*.

### PicLaden

**PicLaden***n*

[2nd] [DRAW]

STO

Zeigt den Graphen an und fügt das in **Pic***n* gespeicherte Bild hinzu.

2:PicLaden

## PicSpeichern

**PicSpeichern***n*

**[2nd] [DRAW]**

**STO**

Speichert das aktuelle Bild in Bild **Pic***n*.

**1:PicSpeichern**

## piecewise

**piecewise**(

**[math]**.

Neue Piecewise-Funktion zur Unterstützung der Eingabe von Funktionen, wie sie im Lehrbuch stehen. Dieser Befehl befindet sich in **[math] MATH B:piecewise**.

**▲** oder **▼** um  
zu **B:piecewise**(  
zu scrollen

## Plot1( Plot2( Plot3(

**Plot#**(*Typ,XListe,YListe[,Zeichen,Farbe#]*)

† **[2nd] [STAT PLOT]**

Definiert **Plot#** (**1, 2** oder **3**) vom *Typ* **Streudiagr** oder **xyLinie** für *XListe* und *YListe* unter Verwendung von *Zeichen* und *Farbe*.

**STATISTISCHE**

**PLOTS**

**1:Plot1**

**2:Plot2**

**3:Plot3**

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**Hinweis:** *XListe* und *YListe* stehen für die XListen- und YListennamen.

## Plot1( Plot2( Plot3(

**Plot#**(*Typ,XListe[,Häufigkeitsliste,Farbe#]*)

† **[2nd] [STAT PLOT]**

Definiert **Plot#** (**1, 2** oder **3**) vom *Typ* **Histogramm** oder **Kästchengrafik** für *XListe* mit der Häufigkeit *Häufigkeitsliste* und Farbe #.

**STATISTISCHE**

**PLOTS**

**1:Plot1**

**2:Plot2**

**3:Plot3**

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**Hinweis:** *XListe* steht für den XListennamen.

## Plot1( Plot2( Plot3(

**Plot#**(*Typ,XListe*,[,*Häufigkeitsliste,Zeichen,Farbe#*])

+ [2nd]  
[STAT PLOT]

Definiert **Plot#** (**1, 2** oder **3**) vom *Typ* **ModKästchengrafik** für *XListe* mit der Häufigkeit *Häufigkeitsliste* unter Verwendung von *Zeichen* und *Farbe #*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**Hinweis:** *XListe* steht für den XListennamen.

STATISTISCHE  
PLOTS  
1:Plot1  
2:Plot2  
3:Plot3

## Plot1( Plot2( Plot3(

**Plot#**(*Typ,Datenliste*,[,*Datenachse,Zeichen,Farbe#*])

+ [2nd]  
[STAT PLOT]

Definiert **Plot#** (**1, 2** oder **3**) vom *Typ* **NormWahrschdiagr** für die *Datenliste* auf der *Datenachse* unter Verwendung von *Zeichen* und *Farbe #* *data axis* can be **X** or **Y**.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**Hinweis:** *Datenliste* steht für den Datenlistennamen.

STATISTISCHE  
PLOTS  
1:Plot1  
2:Plot2  
3:Plot3

## PlotsAus

**PlotsAus** [1,2,3]

[2nd] [STAT PLOT]

Wählt alle statistischen Plots bzw. einen oder mehrere statistische Plots (**1, 2** oder **3**) ab.

STATISTISCHE  
PLOTS  
4:PlotsAus

## PlotsEin

**PlotsEin** [1,2,3]

[2nd] [STAT PLOT]

Wählt alle statistischen Plots bzw. einen oder mehrere statistische Plots (**1, 2** oder **3**).

STATISTISCHE  
PLOTS  
5:PlotsEin

## Pmt\_Anf

**Pmt\_Anf**

[APPS]

Legt vorschüssige Annuitäten fest, bei denen die Zahlung am Anfang der jeweiligen Zahlungsperiode erfolgt.

1:Finanzen  
BERECHNEN  
F:Pmt\_Anf

## Pmt\_Ende

### Pmt\_Ende

Legt nachschüssige Annuitäten fest, bei denen die Zahlung am Ende der jeweiligen Zahlungsperiode erfolgt.

**[APPS]**

**1:Finanzen**  
**BERECHNEN**  
**E:Pmt\_Ende**

## poissoncdf(

### poissoncdf( $\mu, x$ )

Berechnet die kumulative Wahrscheinlichkeit an  $x$  für die diskrete Poisson-Verteilung mit dem vorgegebenen Mittelwert  $\mu$ .

**[2nd] [DISTR]**

**VERTLG**  
**D:poissoncdf**  
**(**

## poissonpdf(

### poissonpdf( $\mu, x$ )

Berechnet die Wahrscheinlichkeit an  $x$  für die diskrete Poisson-Verteilung mit dem vorgegebenen Mittelwert  $\mu$ .

**[2nd] [DISTR]**

**VERTLG**  
**D:poissonpdf**  
**(**

## Polar

### Polar

Stellt den Grafikmodus Polar ein.

**+ [MODE]**

**Polar**

## ►Polar

### *komplexer Wert* ►Polar

Zeigt einen *komplexen Wert* im polaren Format an.

**[MATH]**

**KMPXL**  
**7: ► Polar**

## PolarGC

### PolarGC

Stellt das Format für polare Grafik-Koordinaten ein.

**+ [2nd]**

**[FORMAT]**

**PolarGC**

## prgm

### prgmName

Führt das Programm *Name* aus.

**+ [PRGM]**

**SATRG**  
**D:prgm**

## $\Sigma$ Prn(

$\Sigma$ Prn(*pmt1*,*pmt2*,*WertRunden*)

Berechnet die Summe, gerundet auf *WertRunden*, des Kapitalbetrags zwischen *pmt1* und *pmt2* für einen Amortisationszeitplan.

**APPS**

**1:Finanzen  
BERECHNEN  
0:  $\Sigma$  Prn(**

## prod(

prod(*Liste*[,*Start*,*Ende*])

Ergibt das Produkt aus den Elementen der *Liste* zwischen *Start* und *Ende*

**2nd** **[LIST]**  
**MATH**  
**6:prod(**

## Prompt

Prompt*variableA*[,*variableB*,...,*variable n*]

Prompts for value for *variableA*, then *variableB*, and so on.

**†** **[PRGM]**  
**I/O**  
**2:Prompt**

## 1-PropZInt(

1-PropZInt(*x*,*n*[,*Konfidenzniveau*])

Berechnet ein *z*-Konfidenzintervall für eine Proportion.

**†** **[STAT]**  
**TESTS**  
**A:1-PropZInt(**

## 2-PropZInt(

2-PropZInt(*x1*,*n1*,*x2*,*n2*[,*Konfidenzniveau*])

Berechnet das *z*-Konfidenzintervall für zwei Proportionen.

**†** **[STAT]**  
**TESTS**  
**B:2-PropZInt(**

## 1-PropZTest(

1-PropZTest(*p0*,*x*,*n*[,*Alternative*,*ZeichneFlag*, *Farbe*#])

Berechnet einen *z*-Test für eine Proportion *Alternative*=-1 ist <; *Alternative*=0 ist ; *Alternative*=1 ist >. *ZeichneFlag*=1 zeichnet Ergebnisse; *ZeichneFlag*=0 berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**†** **[STAT]**  
**TESTS**  
**5:1-PropZTest  
(**



## 2-PropZTest{

**2-PropZTest**( $x1, n1, x2, n2$  [, *Alternative*, *ZeichneFlag*, *Farbe#*])

Berechnet einen z-Test für zwei Proportionen. *Alternative=-1* ist <; *Alternative=0* ist =; *Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

+ [STAT]  
TESTS  
6:2-PropZTest  
(

## prüfeTmr{

**prüfeTmr**(*Startzeit*)

Liefert die seit dem Starten des Timers mit **startTmr** abgelaufene Zeit in Sekunden. Die *Startzeit* ist der von **startTmr** angezeigte Wert

[2nd] [CATALOG]  
prüfeTmr(

## P►Rx{

**P►Rx**( $r, \theta$ )

Ergibt **X**, die gegebenen Polarkoordinaten  $r$  und  $\theta$  oder eine Liste mit Polarkoordinaten.

[2nd] [ANGLE]  
WINKEL  
7:P ► Rx(

## P►Ry{

**P►Ry**( $r, \theta$ )

Ergibt **Y**, die gegebenen Polarkoordinaten  $r$  und  $\theta$  oder eine Liste mit Polarkoordinaten.

[2nd] [ANGLE]  
WINKEL  
8:P ► Ry(

## Punkt-Dick

**Punkt-Dick**

Stellt den Punkt-Zeichenmodus ein; setzt alle Diagrammstil-Einstellungen in Y=editor auf Punkt-Dick zurück.

+ [MODE]  
Punkt-Dick

## Punkt-Dünn

**Punkt-Dünn**

Stellt den Punkt-Zeichenmodus ein; setzt alle Diagrammstil-Einstellungen in Y=editor auf Punkt-Dünn zurück.

+ [MODE]  
Punkt-Dünn

## Punkt-Aus{

**Punkt-Aus**( $x, y$  [, *Zeichen*])

[2nd] [DRAW]

### Punkt-Aus(

Löscht einen Punkt an  $(x,y)$  im Graphenbereich unter Verwendung von *Zeichen*. Der Aus-Zustand kann in der durch die Einstellung *ImageVar* oder *Farbe* festgelegten Hintergrundfarbe erfolgen.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**PUNKTE**  
**2:Punkt-Aus(**

### Punkt-Ein(

**Punkt-Ein( $x,y[,Zeichen,Farbe\#]$ )**

Zeichnet einen Punkt an  $(x,y)$  im Graphenbereich unter Verwendung von *Zeichen* und der angegebenen *Farbe#*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**[2nd] [DRAW]**  
**PUNKTE**  
**1:Punkt-Ein(**

### Punkt-umkehren(

**Punkt-umkehren( $x,y[,Farbe\#]$ )**

Schaltet einen Punkt an  $(x,y)$  im Graphenbereich ein oder aus. Ein ausgeschalteter Punkt wird in Hintergrundfarbe dargestellt, ein eingeschalteter in der spezifizierten Farbe

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

**[2nd] [DRAW]**  
**PUNKTE**  
**3:Punkt-umkehren(**

### PwrReg

**PwrReg**  
**[ $XListenname,YListenname,Häufigkeitsliste,regequ$ ]**

Passt ein Potenzregressionsmodell an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

**[STAT]**  
**BERECHNEN**  
**A:PwrReg**

## Pxl-Ändern(

**Pxl-Ändern**(*Zeile,Spalte[,Farbe#]*)

**2nd** [DRAW]

Schaltet im Graphenbereich Aus auf Ein: mit festgelegter *Farbe#*  
Schaltet im Graphenbereich Ein auf Aus: Aus wird mit der eingestellten  
Hintergrundbildvariablen oder -farbe dargestellt.

**PUNKTE**  
**6:Pxl-Ändern**  
(

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## Pxl-Aus(

**Pxl-Aus**(*Zeile,Spalte*)

**2nd** [DRAW]

Der Zustand Aus wird mit der eingestellten Hintergrundbildvariablen oder  
-farbe dargestellt.

**PUNKTE**  
**5:Pxl-Aus(**

## Pxl-Ein(

**Pxl-Ein**(*Zeile,Spalte[,Farbe#]*)

**2nd** [DRAW]

Zeichnet Pixel im Graphenbereich an (*Zeile,Spalte*) in der angegebenen  
Farbe.

**PUNKTE**  
**4:Pxl-Ein(**

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## pxl-Test(

**pxl-Test**(*Zeile,Spalte*)

**2nd** [DRAW]

Liefert den Wert 1, wenn das Pixel (*Zeile,Spalte*) eingeschaltet ist, und  
0, wenn es ausgeschaltet ist.

**PUNKTE**  
**7:pxl-Test(**

## Q

### QuadReg

#### QuadReg

[*XListenname*,*YListenname*,*Häufigkeitsliste*,*regequ*]

[STAT]

BERECHNEN  
5:QuadReg

Passt ein Modell einer quadratischen Regression an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

### QuartReg

#### QuartReg

[*XListenname*,*YListenname*,*Häufigkeitsliste*,*regequ*]

[STAT]

BERECHNEN  
7:QuartReg

Passt ein Modell einer Regression vierter Ordnung an *XListenname* und *YListenname* mit der Häufigkeit *Häufigkeitsliste* an und speichert die Regressionsgleichung in *regequ*.

## R

### RahmenFarbe

RahmenFarbe[Farbe#]

+ [2nd] [FORMAT]

RahmenFarbe

Schaltet eine Rahmenfarbe in der angegebenen Farbe ein, die den Graphenbereich umgibt. Farbe#:1-4.

### RectGC

RectGC

+ [2nd]

[FORMAT]

RectGC

Stellt das Format für kartesische Grafik-Koordinaten ein.

$re^{\theta i}$

$re^{\theta i}$

+ [MODE]

$r e^{\theta i}$

Stellt den Modus auf Polardarstellung komplexer Zahlen ein ( $re^{\theta i}$ ).

### Reell

Reell

+ [MODE]

Reell

Stellt den Modus so ein, dass komplexe Ergebnisse nur bei Eingabe komplexer Zahlen angezeigt werden.

## reell(

### reell(*Wert*)

Ergibt den Realteil einer komplexen Zahl oder einer Liste komplexer Zahlen.

**[MATH]**  
**KMPLX**  
**2:reell(**

## ref(

### ref(*Matrix*)

Ergibt die Diagonalform von *Matrix*.

**[2nd]**  
**[MATRIX]**  
**MATH**  
**A:ref(**

## Rest(

### Rest(*Dividend, Teiler*)

Gibt den Rest der Division zweier ganzer Zahlen, bei welcher der Teiler nicht Null ist, als ganze Zahl aus.

**[MATH]**  
**NUM**  
**0:Rest(**

## Rest(

### Rest(*Liste, Teiler*)

Gibt den Rest der Division zweier Listen, bei welcher der Teiler nicht Null ist, als ganze Zahl aus.

**[MATH]**  
**NUM**  
**0:Rest(**

## Rest(

### Rest(*Dividend, Liste*)

Gibt den Rest der Division zweier ganzer Zahlen, bei welcher der Teiler eine Liste ist, als ganze Zahl aus.

**[MATH]**  
**NUM**  
**0:Rest(**

## Rest(

### Rest(*Liste, Liste*)

Gibt den Rest der Division zweier Listen als ganze Zahl aus.

**[MATH]**  
**NUM**  
**0:Rest(**

## Return

### Return

Kehrt zum aufrufenden Programm zurück.

**+ [PRGM]**  
**SATRG**  
**E:Return**

**R►Pr(****R►Pr( $x,y$ )****2nd** [ANGLE]

WINKEL

5:R ► Pr(

Ergibt R, die gegebenen kartesischen Koordinaten  $x$  und  $y$  oder eine Liste kartesischer Koordinaten.

**R►Pθ(****R►Pθ( $x,y$ )****2nd** [ANGLE]

WINKEL

6:R ► Pθ(

Ergibt  $\theta$ , die gegebenen kartesischen Koordinaten  $x$  und  $y$  oder eine Liste kartesischer Koordinaten.

**rref(****rref(*Matrix*)****2nd** [MATRIX]

MATH

B:rref(

Gibt die reduzierte Diagonalform von *Matrix* zurück.

**runde(****runde(*Wert*[,*#Dezimalstellen*])****MATH**

NUM

2:runde(

Liefert eine Zahl, einen Ausdruck, eine Liste oder Matrix, die auf *#Dezimalzahlen* ( 9) gerundet ist.

**2-SampFTest****2-SampFTest**[  
*Listenname1*,  
*Listenname2*,  
*Häufigkeitsliste1*,*Häufigkeitsliste2,Alternative,ZeichneFlag,Farbe#*]

Führt einen F-Test mit zwei Stichproben durch *Alternative=-1* ist <; *Alternative=0* ist ; *Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

† **STAT****TESTS****E:2-Samp F Test****2-SampFTest****2-SampFTest***Sx1,n1,Sx2,n2*[,*Alternative,ZeichneFlag,Farbe#*]

Führt einen F-Test mit zwei Stichproben durch *Alternative=-1* ist <; *Alternative=0* ist ; *Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

† **STAT****TESTS****E:2-Samp F Test****2-SampTInt****2-SampTInt**[  
*Listenname1*,  
*Listenname2*,  
*Häufigkeitsliste1*,*Häufigkeitsliste2,Konfidenzniveau,verteilt*]**(Datenlisteneingabe)**

Berechnet ein t-Konfidenzintervall für zwei Stichproben. *Verteilt=1* verteilt Varianzen; *Verteilt=0* verteilt keine Varianzen.

† **STAT****TESTS****O:2-SampTInt****2-SampTInt****2-SampTInt** *$\bar{x}1, Sx1, n1, \bar{x}2, Sx2, n2$* [,*Konfidenzniveau,verteilt*]**(Zusammenfassung der statistischen Eingaben)**† **STAT****TESTS****O:2-SampTInt**

## 2-SampTInt

Berechnet ein  $t$ -Konfidenzintervall für zwei Stichproben. *Verteilt=1* verteilt Varianzen; *Verteilt=0* verteilt keine Varianzen.



## 2-SampTTest

### 2-SampTTest

```
[  
  Listenname1  
,  
  Listenname2  
,  
  Häufigkeitsliste1  
,  
  Häufigkeitsliste2  
,Alternative,verteilt,ZeichneFlag,Farbe#])
```

Berechnet einen *t*-Test für zwei Stichproben. *Alternative*= -1 ist <; *Alternative*=0 ist =; *Alternative*=1 ist >. *Verteilt*=1 verteilt Varianzen; *Verteilt*=0 verteilt keine Varianzen. *ZeichneFlag*=1 zeichnet Ergebnisse; *ZeichneFlag*=0 berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

+ **STAT**  
TESTS 4:2-  
SampTTest

## 2-SampTTest

```
2-SampTTest( $\bar{x}1, Sx1, n1, v2, Sx2, n2$   
[,Alternative,verteilt,ZeichneFlag,Farbe#])
```

Berechnet einen *t*-Test für zwei Stichproben. *Alternative*= -1 ist <; *Alternative*=0 ist =; *Alternative*=1 ist >. *Verteilt*=1 verteilt Varianzen; *Verteilt*=0 verteilt keine Varianzen. *ZeichneFlag*=1 zeichnet Ergebnisse; *ZeichneFlag*=0 berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

+ **STAT**  
TESTS 4:2-  
SampTTest

## 2-SampZInt(

```
2-SampZInt( $\sigma_1, \sigma_2$   
[,  
  Listenname1  
,  
  Listenname2  
,  
  Häufigkeitsliste1  
,Häufigkeitsliste2,Konfidenzniveau])  
(Datenlisteneingabe)
```

Berechnet ein *z*-Konfidenzintervall für zwei-Stichproben.

+ **STAT**  
TESTS 9:2-  
SampZInt(

## 2-SampZInt(

**2-SampZInt( $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2$  [, *Konfidenzniveau*])**  
(Zusammenfassung der statistischen Eingaben)

Berechnet ein z-Konfidenzintervall für zwei-Stichproben.

† **STAT**  
**TESTS 9:2-**  
**SampZInt(**

## 2-SampZTest(

2-SampZTest( $\sigma_1, \sigma_2$

[, *Listenname1*

, *Listenname2*

, *Häufigkeitsliste1*

, *Häufigkeitsliste2, Alternative, ZeichneFlag, Farbe#*)

Berechnet einen z-Test für zwei Stichproben. *Alternative=-1* ist <; *Alternative=0* ist =; *Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

† [STAT]  
TESTS 3:2-  
SampZTest(

## 2-SampZTest(

2-SampZTest( $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2$

[, *Alternative, ZeichneFlag, Farbe#*)

Berechnet einen z-Test für zwei Stichproben. *Alternative=-1* ist <; *Alternative=0* ist =; *Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

† [STAT]  
TESTS 3:2-  
SampZTest(

## Schraff $\chi^2$ (

Schraff $\chi^2$ (*untereGrenze, obereGrenze, df, Farbe#*)

Zeichnet die Dichtefunktion für die Verteilung  $\chi^2$ , die durch Freiheitsgrade *df* spezifiziert ist, und schraffiert und färbt die Fläche zwischen *untereGrenze* und *obereGrenze*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

[2nd] [DISTR]  
ZEICHNEN  
3:Schraff  $\chi^2$ (

## SchraffF(

SchraffF

(  
*untereGrenze*  
*, obereGrenze, numerator df, denominator df*  
[, *Farbe#*])

Zeichnet die Dichtefunktion für die Verteilung F, die durch

[2nd] [DISTR]  
ZEICHNEN  
4:Schraff F (

## SchraffF(

*numerator df* und *denominator df* spezifiziert ist, und schraffiert und färbt die Fläche zwischen *untereGrenze* und *obereGrenze*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## Schraff(

**Schraff(*untereFunk,obereFunk*  
[,*Xlinks,Xrechts,Muster,Musteraufl,Farbe#*])**

**2nd [DRAW]  
ZEICHNEN  
7:Schraff(**

Zeichnet *untereFunk* und *obereFunk* in Bezug auf **X** auf den aktuellen Graphen und verwendet *Muster* und *Musteraufl*, um die von *untereFunk*, *obereFunk*, *Xlinks* und *Xrechts* eingeschlossene Fläche zu schraffieren und zu färben. *untereFunk* und *obereFunk* sind in derselben angegebenen Farbe schraffiert.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## SchraffNorm(

**SchraffNorm(*untereGrenze,obereGrenze*  
[, $\mu$ , $\sigma$ ,*Farbe#*])**

**2nd [DISTR]  
ZEICHNEN  
1:SchraffNorm(**

Zeichnet die durch  $\mu$  und  $\sigma$  spezifizierte normale Dichtefunktion und schraffiert und färbt die Fläche zwischen *untereGrenze* und *obereGrenze*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## Schraff\_t(

**Schraff\_t(*untereGrenze,obereGrenze,df*[,*Farbe#*])**

**2nd [DISTR]  
ZEICHNEN  
2:Schraff\_t(**

Zeichnet die Dichtefunktion für die Student-t-Verteilung, die durch Freiheitsgrade *df* spezifiziert ist, und schraffiert oder färbt die Fläche zwischen *untereGrenze* und *obereGrenze*.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## Sci

**Sci**

**+ [MODE]  
Sci**

Stellt den wissenschaftlichen Anzeigemodus ein.

## Send(

### Send(*string*)

Sendet einen oder mehrere TI-Innovator™ Hub Befehle an einen angeschlossenen Hub.

#### Hinweise:

- Siehe auch die mit dem Befehl Send( verbundenen Befehle eval( und Get(.
- TI-Innovator™ Hub-Befehle werden im HUB-Untermenü des Programm-Editors von CE OS v.5.2 unterstützt.

+ **PRGM**  
I/O  
B:Send(

## Send(

### Send(*string*)

Sendet einen oder mehrere TI-Innovator™ Hub Befehle an einen angeschlossenen Hub.

#### Hinweise:

- Siehe auch die mit dem Befehl Send( verbundenen Befehle eval( und Get(.
- TI-Innovator™ Hub-Befehle werden im HUB-Untermenü des Programm-Editors von CE OS v.5.2 unterstützt.

TI-  
Innovator™  
Hub  
+ **PRGM**  
HUB  
Menü-  
Speicherort  
abhängig  
von den TI-  
Innovator  
Hub  
Sensoren

## Sequenziell

### Sequenziell

Stellt den Modus zum sequenziellen Zeichnen von Funktionen ein.

+ **MODE**  
Sequenziell

## SetUpEditor

### SetUpEditor

Löscht alle Listennamen aus dem Stat-Listeneditor und stellt die Listennamen L1 bis L6 für Spalte 1 bis 6 wieder her.

**STAT**  
BEARBEITEN  
5:SetUpEditor

## SetUpEditor

### SetUpEditor *Listenname* [,*Listenname*2,...,*Listenname*20]

Löscht alle Listennamen im Stat-Listeneditor und stellt ihn so ein, dass ein oder mehrere *Listennamen* in der festgelegten Reihenfolge angezeigt werden, beginnend mit Spalte 1.

**STAT**  
BEARBEITEN  
5:SetUpEditor

## Simul

### Simul

+ [MODE]

Simul

Stellt den Modus zum gleichzeitigen Zeichnen von Funktionen ein.

## sin(

### sin(*Wert*)

[SIN]

Ergibt den Sinus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## sin<sup>-1</sup>(

### sin<sup>-1</sup>(*Wert*)

[2nd] [SIN<sup>-1</sup>]

Ergibt den Arkussinus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## sinh(

### sinh(*Wert*)

[2nd] [CATALOG]

sinh(

Ergibt den Sinus hyperbolicus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## sinh<sup>-1</sup> (

### sinh<sup>-1</sup> (*Wert*)

[2nd] [CATALOG]

sinh<sup>-1</sup> (

Ergibt den Arkussinus hyperbolicus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## SinReg

### SinReg

[STAT]

[

BERECHNEN

*Iterationen*

C:SinReg

*,XListenname,YListenname,Periode,regequ]*

Versucht zu *Iterationen* der Zeit ein Sinus-Regressionsmodell an *XListenname* und *YListenname* mittels einer *Periode* anzupassen und speichert die Regressionsgleichung in *regequ*.

## SortAb(

**SortAb**(*Listenname*)

**[2nd] [LIST]**

Sortiert Elemente aus *Listenname* in absteigender Reihenfolge.

**OPS**

**2:SortAb(**

## SortAb(

**SortAb**(*Schlüssellistenname*,*Abhäng-Liste 1*  
[,*Abhäng-Liste 2*,...,*Abhäng-Liste n*])

**[2nd] [LIST]**

Sortiert Elemente aus *Schlüssellistenname* in absteigender Reihenfolge und sortiert anschließend jede *Abhäng-Liste* als eine abhängige Liste.

**OPS**

**2:SortAb(**

## SortAuf(

**SortAuf**(*Listenname*)

**[2nd] [LIST]**

Sortiert Elemente aus *Listenname* in aufsteigender Reihenfolge.

**OPS**

**1:SortAuf(**

## SortAuf(

**SortAuf**(*Schlüssellistenname*,*Abhäng-Liste 1*  
[,*Abhäng-Liste 2*,...,*Abhäng-Liste n*])

**[2nd] [LIST]**

Sortiert Elemente aus *Schlüssellistenname* in aufsteigender Reihenfolge und sortiert anschließend jede *Abhäng-Liste* als eine abhängige Liste.

**OPS**

**1:SortAuf(**

## Speichern →

**Speichern:** *Wert* → *Variable*

**[STO▶]**

Speichert *Wert* in *Variable*

## Spur

**Spur**

**[TRACE]**

Zeigt den Graphen an und wechselt in den **SPUR**-Modus.

## startTmr

**startTmr**

**[2nd] [CATALOG]**

**startTmr**

Startet die Timer-Funktion der Uhr. Speichern oder notieren Sie sich den angezeigten Wert und setzen Sie ihn als Argument für **prüfeTmr()** zur Ermittlung der abgelaufenen Zeit ein.

## StdAbw(

**StdAbw**(*Liste*[,*Häufigkeitsliste*])

**[2nd] [LIST]**

Ergibt die Standardabweichung der Elemente in der *Liste* mit der Häufigkeit *Häufigkeitsliste*.

**MATH**  
**7:StdAbw(**

## STATASSIST AUS

**STATASSIST AUS**

**[2nd] [CATALOG]**

Deaktiviert die Assistenten-Syntaxhilfe für statistische Befehle, Verteilungen und Folge(.

**STATASSIST**  
**AUS**

## STATASSIST EIN

**STATASSIST EIN**

**[2nd] [CATALOG]**

Aktiviert die Assistenten-Syntaxhilfe für statistische Befehle, Verteilungen und Folge(.

**STATASSIST**  
**EIN**

## Stopp

**Stopp**

**+ [PRGM]**

Beendet die Ausführung eines Programms; kehrt zum Hauptbildschirm zurück.

**SATRG**  
**F:Stopp**

## Streudiagr

**Streudiagr** Plot#(*Typ*,*XListe*[,*Häufigkeitsliste*,*Farbe*#])

**+ [2nd] [stat plot]**

**TYP**

Wird als „Typ“-Argument im Befehl verwendet

# ergibt Plot1, Plot2 oder Plot3.

## String→Equ(

**String→Equ**(*String*,*Y= Var*)

**+ [PRGM]**

Wandelt *String* in eine Gleichung um und speichert diese in *Y= var*.

*String* kann ein String oder eine String-Variable sein.

**I/O**  
**F:String→Equ**  
**(**

**String→Equ**( ist der Kehrwert von **Equ→String**(.

## sub(

**sub**(*String*,*Anfang*,*Länge*)

**[2nd] [CATALOG]**

Ergibt einen String, der eine Teilmenge eines anderen *Strings* ist, von *Anfang* bis *Länge*.

**sub(**



## sum(

**sum**(*Liste*[,*Start*,*Ende*])

**2nd** **[LIST]**

Ergibt die Summe der Elemente der *Liste* von *Start* bis *Ende*.

**MATH**

**5:Sum(**

## Summierung $\Sigma$ (

**$\Sigma$** (*Ausdruck*[,*Start*,*Ende*])

**[MATH]**

Klassischer Befehl wie abgebildet.

**NUM**

In MathPrint™ zeigt und ergibt die Vorlage zur Eingabe einer Summierung die Summe der Elemente der *Liste* von *Start* bis *Ende*, wobei *Start* <= *Ende*.

**0: Summierung $\Sigma$ (**

## T

### tan(

**tan**(*Wert*)

**[TAN]**

Liefert den Tangens einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

### tan<sup>-1</sup>(

**tan<sup>-1</sup>**(*Wert*)

**2nd** **[TAN<sup>-1</sup>]**

Liefert den Arkustangens einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

### Tangente(

**Tangente**(*Ausdruck*,*Wert*[,*Farbe*#,*Linestyle*#])

**2nd** **[DRAW]**

Zeichnet eine Linientangente zum *Ausdruck* bei *X=Wert* mit der festgelegten *Farbe* #: 10-24 und der Linienart *Linestyle*#: 1-2.

**ZEICHNEN**

**5:Tangente(**

*Farbe*#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

### tanh(

**tanh**(*Wert*)

**2nd** **[CATALOG]**

Liefert den Tangens hyperbolicus einer reellen oder komplexen Zahl, eines

**tanh(**

## **tanh(**

Ausdrucks oder einer Liste.

## **tanh<sup>-1</sup>(**

**tanh<sup>-1</sup>(Wert)**

**[2nd] [CATALOG]**

**tanh<sup>-1</sup>(**

Liefert den Arkustangens hyperbolicus einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

## **tcdf(**

**tcdf(untereGrenze,obereGrenze,df)**

**[2nd] [DISTR]**

**VERTLG**

**6:tcdf(**

Berechnet für eine Student-*t*-Verteilung mit vorgegebenen Freiheitsgraden *df* die Verteilungswahrscheinlichkeit zwischen *UntereGrenze* und *ObereGrenze*.

## **Text(**

**Text(Zeile,Spalte,Text1,Text2,...,Text n)**

**[2nd] [DRAW]**

**ZEICHNEN**

**0:Text(**

Schreibt *Text* auf den Graphen, beginnend bei Pixel (*Zeile,Spalte*), wobei 0 *Zeile* 164 und 0 *Spalte* 264 ist.

Im Vollmodus muss *Zeile* <=148 sein; *Spalte* muss <=256 sein

Im Modus Horiz muss *Zeile* <=66 sein; *Spalte* muss <=256 sein

Im Modus G-T muss *Zeile* <=126 sein; *Spalte* muss <=176 sein

## **TextFarbe(**

**TextFarbe([Farbe#])**

**+ [2nd] [DRAW]**

**ZEICHNEN**

**A:TextFarbe**

Stellen Sie die Textfarbe ein, bevor Sie den Befehl **Text(** verwenden.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## **Then**

**Then**

*Siehe If:Then*

## TIntervall

**TIntervall** [*Listenname,Häufigkeitsliste,Konfidenzniveau*] + **[STAT]**  
(**Datenlisteneingabe**) TESTS  
8:TIntervall

Berechnet das Konfidenzintervall  $t$ .

## TIntervall

**TIntervall**  $\bar{x},Sx,n$  [,*Konfidenzniveau*] + **[STAT]**  
(**Zusammenfassung der statistischen Eingaben**) TESTS  
8:TIntervall

Berechnet das Konfidenzintervall  $t$ .

## toString(

**toString**( (*Wert* [,*Format*] ) + **[PRGM]**  
I/O  
E:toString(

Wandelt einen Wert in eine Zeichenkette um, wobei *Wert* reell, komplex, ein ausgewerteter Ausdruck, eine Liste oder eine Matrix sein kann. Der String *Wert* wird entsprechend der Moduseinstellung AUTO/DEZ im klassischen *Format* (0) oder im *Dezimalformat* (1) angezeigt.

## tpdf(

**tpdf**( $x,df$ ) **[2nd] [DISTR]**  
VERTLG  
5:tpdf(

Berechnet die Wahrscheinlichkeitsdichtefunktion (Pdf) einer Student- $t$ -Verteilung an einem bestimmten  $x$ -Wert für die vorgegebenen Freiheitsgrade  $df$ .

## T-T-Test

**T-Test**  $\mu 0$  + **[STAT]**  
[ TESTS  
, 2:T-Test(  
,*Listenname*  
,*Häufigkeitsliste,Alternative,ZeichneFlag,Farbe*#)]  
(**Datenlisteneingabe**)

Führt einen  $t$ -Test mit der Häufigkeit *Häufigkeitsliste* durch.  
*Alternative*= -1 ist <; *Alternative*=0 ist ; *Alternative*=1 ist >.  
*ZeichneFlag*=1 zeichnet Ergebnisse; *ZeichneFlag*=0 berechnet Ergebnisse.

*Farbe*#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## T-T-Test

**T-Test**  $\mu 0,\bar{x},Sx,n$  [,*Alternative,ZeichneFlag,Farbe*#)] + **[STAT]**

## T-T-Test

Führt einen  $t$ -Test mit der Häufigkeit *Häufigkeitsliste* durch.  
*Alternative=-1* ist < ; *Alternative=0* ist = ; *Alternative=1* ist > .  
*ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

TESTS  
2:T-Test()

## tvm\_FV

**tvm\_FV**[(N,I%,PV,PMT,P/Y,C/Y)]

Berechnet den Endwert.

APPS

1:Finanzen  
BERECHNEN  
6:tvm\_FV

## tvm\_I%

**tvm\_I%**[(N,PV,PMT,FV,P/Y,C/Y)]

Berechnet den Jahreszinssatz.

APPS

1:Finanzen  
BERECHNEN  
3:tvm\_I%

## tvm\_N

**tvm\_N**[(I%,PV,PMT,FV,P/Y,C/Y)]

Berechnet die Anzahl der Zahlungsperioden.

APPS

1:Finanzen  
BERECHNEN  
5:tvm\_N

## tvm\_Pmt

**tvm\_Pmt**[(N,I%,PV,FV,P/Y,C/Y)]

Berechnet den Betrag jeder Zahlung.

APPS

1:Finanzen  
BERECHNEN  
2:tvm\_Pmt

## tvm\_PV

**tvm\_PV**[(N,I%,PMT,FV,P/Y,C/Y)]

Berechnet den Barwert.

APPS

1:Finanzen  
BERECHNEN  
4:tvm\_PV

# U

## UhrEin

### UhrEin

Schaltet die Anzeige der Uhr im Modusbildschirm ein.

**2nd**  
[CATALOG]  
UhrEin

## UhrAus

### UhrAus

Schaltet die Anzeige der Uhr im Modusbildschirm aus.

**2nd**  
[CATALOG]  
UhrAus

## Un/d

### Un/d

Zeigt Ergebnisse gegebenenfalls als gemischte Zahl an.

**MATH**  
NUM  
C: Un/d

oder

**MATH**  
BRUCH  
2:Un/d

## UnabhAuto

### UnabhAuto

Stellt eine Tabelle zur automatischen Erzeugung unabhängiger Variablenwerte ein.

+ **2nd**  
[TBLSET]  
Indpnt:  
Auto

## und

### *WertA*und*WertB*

Liefert den Wert 1 (wahr), wenn sowohl *WertA* als auch *WertB* wahr ist. Anderenfalls wird 0 (falsch) zurückgegeben.

*WertA* und *WertB* können reelle Zahlen, Ausdrücke oder Listen sein.

### Tipp zum TI Connect™ Programm-Editor:

Das Token ist „\_und\_“, wobei „\_“ ein Leerzeichen ist.

**2nd** [TEST]  
LOGIK  
1:und

## uvAxes

### uvAxes

+ [2nd]  
[FORMAT]

Stellt Sequenz-Graphen so ein, dass  $u(n)$  auf der x-Achse und  $v(n)$  auf der y-Achse dargestellt wird.

uv

## uwAxes

### uwAxes

+ [2nd]  
[FORMAT]

Stellt Sequenz-Graphen so ein, dass  $u(n)$  auf der x-Achse und  $w(n)$  auf der y-Achse dargestellt wird.

uw

## V

## Varianz(

### Varianz(*liste*,*Häufigkeitsliste*)

[2nd] [LIST]

Ergibt die Varianz der Elemente in der *Liste* mit der Häufigkeit *Häufigkeitsliste*.

MATH

8:Varianz(

## 1-VarStats

### 1-VarStats [*Xlistname*,*Häufigkeitsliste*]

[STAT]

Führt eine Analyse mit einer Variablen der Daten in *Xlistname* mit der Häufigkeit *Häufigkeitsliste* durch.

BERECHNEN

1:1-Var Stats

## 2-VarStats

### 2-VarStats [*Xlistname*,*Ylistname*,*Häufigkeitsliste*]

[STAT]

Führt eine Analyse mit zwei Variablen der Daten in *Xlistname* und *Ylistname* mit der Häufigkeit *Häufigkeitsliste* durch.

BERECHNEN

2:2-Var Stats

## Vergrößern

### Vergrößern

+ [ZOOM]

Vergrößert den um die Cursorposition herumliegenden Teil des Graphen.

ZOOM

2:Vergrößern

## Verkleinern

### Verkleinern

+ [ZOOM]

Zeigt einen größeren Teil des Graphen an, wobei sich die Cursorposition in der Mitte befindet.

ZOOM

3:Verkleinern

## Vertikale

**Vertikale**  $x$ [*Farbe#*,*Linestyle#*]

[2nd] [DRAW]

Zeichnet eine vertikale Linie an  $x$  mit der angegebenen Farbe und Linienart.

ZEICHNEN  
4:Vertikale

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

Linienart #: 1-4.

## Voll

**Voll**

+ [MODE]

Schaltet auf Vollbildmodus.

Voll

## vwAxes

**vwAxes**

+ [2nd]

Stellt Sequenz-Graphen so ein, dass  $v(n)$  auf der x-Achse und  $w(n)$  auf der y-Achse dargestellt wird.

[FORMAT]

vw

## W

### Wait

**WaitZeit**

+ [PRGM]

Hält die Ausführung eines Programms für eine gegebene Zeit an. Die maximale Zeit beträgt 100 Sekunden.

CTL

A:Wait

### Wait

TI-  
Innovator™  
Hub

**WaitZeit**

+ [PRGM]

Hält die Ausführung eines Programms für eine gegebene Zeit an. Die maximale Zeit beträgt 100 Sekunden.

HUB

4:Wait

## Wiederholen

**Wiederholen-Bedingung:Befehle:Ende:Befehle**

+ [PRGM]

Führt *Befehle* aus, bis die *Bedingung* wahr ist.

SATRG

6:Wiederholen

## :While

**:WhileBedingung:Befehle**

+ [PRGM]

**:Ende:Befehl**

SATRG

## :While

Führt *Befehle* aus, solange die *Bedingung* wahr ist.

5:While

## Winkel()

**Winkel**(*Wert*)

Ergibt den Polarwinkel einer komplexen Zahl oder einer Liste komplexer Zahlen.

[MATH]  
KIMPLX  
4:Winkel()

## wochTag()

**wochTag**(*Jahr, Monat, Tag*)

Gibt eine Ganzzahl zwischen 1 und 7 zurück, die für einen bestimmten Wochentag steht. Mit **wochTag**() lässt sich ermitteln, auf welchen Wochentag ein bestimmtes Datum fällt. Das *Jahr* muss vierstellig eingegeben werden, *Monat* und *Tag* können aus einer oder zwei Ziffern bestehen.

[2nd] [CATALOG]  
wochTag()  
1:Sonntag  
2:Montag  
3:Dienstag...

## X

### xoder

*WertA* **xoder** *WertB*

Liefert den Wert 1 nur wenn *WertA* oder *WertB*=0. *WertA* und *WertB* können reelle Zahlen, Ausdrücke oder Listen sein.

[2nd] [TEST]  
LOGIK  
3:xoder

### xyLinie

**xyLinie** Plot#(*Typ, XListe*, [, *Häufigkeitsliste, Farbe*#])

Wird als „Typ“-Argument im Befehl verwendet

# ergibt Plot1, Plot2 oder Plot3.

+ [2nd]  
[stat plot]  
TYP

## Z

### ZBox

**ZBox**

Zeigt einen neuen Graphen an, ermöglicht das Zeichnen eines Kästchens, das ein neues Anzeigefenster definiert, und aktualisiert das Fenster.

+ [ZOOM]  
ZOOM  
1:ZBox



## ZBruch1/2

### ZBruch1/2

**ZOOM**

**ZOOM**

**B:ZBruch1/2**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{2}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{2}$  und  $\Delta X$  und  $\Delta Y$  auf  $\frac{1}{4}$ .

## ZBruch1/3

### ZBruch1/3

**ZOOM**

**ZOOM**

**C:ZBruch1/3**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{3}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{3}$  und  $\Delta X$  und  $\Delta Y$  auf  $\frac{1}{6}$ .

## ZBruch1/4

### ZBruch1/4

**ZOOM**

**ZOOM**

**D:ZBruch1/4**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{4}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{4}$  und  $\Delta X$  und  $\Delta Y$  auf  $\frac{1}{8}$ .

## ZBruch1/5

### ZBruch1/5

**ZOOM**

**ZOOM**

**E:ZBruch1/5**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{5}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{5}$  und  $\Delta X$  und  $\Delta Y$  auf  $\frac{1}{10}$ .

## ZBruch1/8

### ZBruch1/8

**ZOOM**

**ZOOM**

**F:ZBruch1/8**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{8}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{8}$  und  $\Delta X$  und  $\Delta Y$  auf  $\frac{1}{16}$ .

## ZBruch1/10

### ZBruch1/10

**ZOOM**

**ZOOM**

**G:ZBruch1/10**

Stellt die Fenstervariablen so ein, dass Sie in Schritten von  $\frac{1}{10}$  von Punkt zu Punkt wechseln können, sofern möglich. Setzt **SpurSchritt** auf  $\frac{1}{10}$  und  $\Delta X$

## ZBruch1/10

und  $\Delta Y$  auf  $\frac{1}{20}$ .

## ZchInv

**ZchInv***Ausdruck*[,Farber#]

**[2nd] [DRAW]**  
**ZEICHNEN**  
**8:ZchInv**

Zeichnet den Kehrwert von *Ausdruck* durch Zeichnen der **X**-Werte auf der Y-Achse und Zeichnen der **Y**-Werte auf der X-Achse mit der angegebenen

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## ZDezimal

**ZDezimal**

+ **[ZOOM]**  
**ZOOM**  
**4:ZDezimal**

Passt das Anzeigefenster so an, dass **SpurSchritt=0,1**,  $\Delta X=0,5$  und  $\Delta Y=0,5$  ist und zeigt den Grafik-Bildschirm mit dem Ursprung in der Bildschirmmitte an.

## ZeichneF

**ZeichneF***Ausdruck*[,Farber#]

**[2nd] [DRAW]**  
**ZEICHNEN**  
**6:ZeichneF**

Zeichnet *Ausdruck* (in Bezug auf **X**) auf den Graphen mit der angegebenen

Farbe#:10 - 24 oder dem aus [Variablen] FARBE eingefügten Namen der Farbe.

## \*Zeile(

**\*Zeile**(*Wert*,*Matrix*,*Zeile*)

**[2nd] [MATRIX]**  
**MATH**  
**E: \* Zeile(**

Liefert eine Matrix mit *Zeile* der *Matrix* multipliziert mit *Wert*, gespeichert in *Zeile*.

## Zeile+(

**Zeile+**(*Matrix*,*ZeileA*,*ZeileB*)

**[2nd] [MATRIX]**  
**MATH**  
**D:Zeile+(**

Liefert eine Matrix mit *ZeileA* der *Matrix* addiert mit *ZeileB*, gespeichert in *ZeileB*.

## \*Zeile+(

**\*Zeile+**(*Wert*,*Matrix*,*ZeileA*,*ZeileB*)

**[2nd] [MATRIX]**  
**MATH**

## \*Zeile+(

Liefert eine Matrix mit *ZeileA* der *Matrix* multipliziert mit *Wert*, addiert mit *ZeileB*, gespeichert in *ZeileB*.

F: \* Zeile+(

## Zeilentausch(

**Zeilentausch**(*Matrix*,*ZeileA*,*ZeileB*)

Liefert eine Matrix, in der *ZeileA* der *Matrix* und *ZeileB* getauscht sind.

**[2nd] [MATRIX]**  
**MATH**  
**C:Zeilentausch(**

## Zeit

**Zeit**

Stellt Sequenz-Diagramme so ein, dass die Zeit beim Zeichnen berücksichtigt wird.

+ **[2nd]**  
**[FORMAT]**  
**Zeit**

## ZeitEinst(

**ZeitEinst**(*Stunde*,*Minute*,*Sekunde*)

Stellt die Zeit im Format Stunde, Minute, Sekunde ein. Die *Stunde* muss im 24-Stunden-Format angegeben werden.

**[2nd] [CATALOG]**  
**ZeitEinst(**

## zeitUmw(

**zeitUmw**(*Sekunden*)

Konvertiert Sekunden in Zeiteinheiten, die sich leichter verstehen und auswerten lassen. Die Liste weist das Format *{Tage,Stunden,Minuten,Sekunden}* auf.

**[2nd] [CATALOG]**  
**zeitUmw**

## ZGanzZ

**ZGanzZ**

Definiert das Anzeigefenster unter Verwendung der folgenden Dimensionen neu: **SpurSchritt=1**,  **$\Delta X=0,5$** ,  **$Xscl=10$** ,  **$\Delta Y=1$** ,  **$Yscl=10$** .

+ **[ZOOM]**  
**ZOOM**  
**8:ZGanzZ**

## ZIntervall

**ZIntervall** $\sigma$

[*Listenname*,*Häufigkeitsliste*,*Konfidenzniveau*]  
(*Datenlisteneingabe*)

Berechnet ein  $z$ -Konfidenzintervall.

+ **[STAT]**  
**TESTS**  
**7:ZIntervall**

## ZIntervall

**ZIntervall** $\sigma, \bar{x}, n$ , [Konfidenzniveau]  
(Zusammenfassung der statistischen Eingaben)

+ [STAT]

TESTS

Berechnet ein  $z$ -Konfidenzintervall.

7:ZIntervall

## ZoomFit

**ZoomFit**

+ [ZOOM]

ZOOM

Berechnet **Ymin** und **Ymax** neu, um die Minimum- und Maximumwerte für **Y** zwischen **Xmin** und **Xmax** der ausgewählten Funktionen einzuschließen, und zeichnet die Funktionen neu.

0:ZoomFit

## ZoomLad

**ZoomLad**

+ [ZOOM]

ZOOM

Zeichnet die ausgewählten Funktionen in einem benutzerdefinierten Anzeigefenster.

3:ZoomLad

## ZoomSpch

**ZoomSpch**

+ [ZOOM]

SPEICHER

Speichert das aktuelle Anzeigefenster sofort.

2:ZoomSpch

## ZoomStat

**ZoomStat**

+ [ZOOM]

ZOOM

Definiert das Anzeigefenster so, dass alle statistischen Datenpunkte angezeigt werden.

9:ZoomStat

## ZQuadr

**ZQuadr**

+ [ZOOM]

ZOOM

Passt die **X**- oder **Y**-Fenstereinstellungen so an, dass Höhe und Breite jedes Pixels im Koordinatensystem gleich groß sind, und aktualisiert das Anzeigefenster.

5:ZQuadr

## ZQuadrant1

**ZQuadrant1**

[ZOOM]

ZOOM

Zeigt den Teil des Graphen an, der sich in Quadrant 1 befindet.

A:ZQuadrant1

## ZStandard

### ZStandard

+ **ZOOM****ZOOM**

Zeichnet die Funktionen sofort neu und setzt dabei die Fenstervariablen auf die Standardwerte.

**6:ZStandard**

## Z-Test(

### Z-Test( $\mu$ , $\sigma$

+ **STAT**

[

**TESTS**

,

**1:Z-Test(***Listenname**,Häufigkeitsliste,Alternative,ZeichneFlag,Farbe#)***(Datenlisteneingabe)**

Führt einen z-Test mit der Häufigkeit *Häufigkeitsliste* durch.

*Alternative=-1* ist <; *Alternative=0* ist ; *Alternative=1* ist >.

*ZeichneFlag=1* zeichnet Ergebnisse; *ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## Z-Test(

### Z-Test( $\mu$ , $\sigma$ , $\bar{x}$ , $n$ [,*Alternative,ZeichneFlag,Farbe#*])

+ **STAT****(Zusammenfassung der statistischen Eingaben)****TESTS****1:Z-Test(**

Führt einen z-Test durch. *Alternative=-1* ist <; *Alternative=0* ist ;

*Alternative=1* ist >. *ZeichneFlag=1* zeichnet Ergebnisse;

*ZeichneFlag=0* berechnet Ergebnisse.

Farbe#: 10 - 24 oder aus [Variablen] FARBE eingefügter Name der Farbe.

## ZtFmtEinst(

### ZtFmtEinst(*Ganzzahl*)

**2nd** **CATALOG****ZtFmtEinst(**

Stellt das Zeitformat ein.

12 = 12-Stunden-Format

24 = 24-Stunden-Format

## ZTrig

### ZTrig

+ **ZOOM****ZOOM 7:ZTrig**

Zeichnet die Funktionen sofort neu und setzt dabei die Fenstervariablen auf die voreingestellten Werte für die Darstellung trigonometrischer Funktionen.

## ZufallM(

**ZufallM**(*Zeilen,Spalten*)

Ergibt eine Zufallsmatrix von *Zeilen* × *Spalten*.

Max. Zeilen x Spalten = 400 Matrixelemente.

**2nd**

**[MATRIX]**

**MATH**

**6:ZufallM(**

## Zufallz

**Zufallz**(*AnzahlVersuche*)

Ergibt eine Zufallszahl zwischen 0 und 1 für eine festgelegte Anzahl von Versuchen *AnzahlVersuche*.

**[MATH]**

**WAHRS**

**1:Zufallz**

## ZufBin(

**ZufBin**(*AnzahlVersuche,prob[,numsimulations]*)

Erzeugt und zeigt eine reelle Zufallszahl aus einer angegebenen Binomialverteilung.

**[MATH]**

**WAHRS**

**7:ZufBin(**

## zufInt(

**zufInt( untere, obere [, AnzahlVersuche])**

Erzeugt und zeigt eine ganzzahlige Zufallszahl innerhalb eines durch *untere* und *obere* Integergrenzen festgelegten Bereichs für eine angegebene Anzahl von Versuchen *AnzahlVersuche*.

**[MATH]**

**WAHRS**

**5:zufInt(**

## zufIntKWhlg(

**zufIntKWhlg(untere, obere [, AnzElemente])**

Gibt eine zufällig geordnete Liste von Ganzzahlen zwischen einer unteren und einer oberen Ganzzahl zurück, wobei die Liste die untere und/oder die obere Ganzzahl enthalten kann. Wenn das optionale Argument *AnzElemente* angegeben ist, werden die ersten *AnzElemente* aufgeführt. Der erste *AnzElemente*-Term in der Liste der ganzzahligen Zufallszahlen wird angezeigt.

**[MATH]**

**WAHRS**

**8:zufIntKWhlg(**

## zufNorm(

**zufNorm( $\mu, \sigma$  [, AnzahlVersuche])**

Erzeugt und zeigt eine reelle Zufallszahl aus einer durch  $\mu$  und  $\sigma$  spezifizierten Normalverteilung für eine festgelegte Anzahl von Versuchen *AnzahlVersuche*.

**[MATH]**

**WAHRS**

**6:zufNorm(**

## ZVorherige

**ZVorherige**

Zeichnet den Graphen mit den Fenstervariablen des vor der letzten Ausführung der **ZOOM**-Anweisung angezeigten Graphen neu.

**+ [ZOOM]**

**SPEICHER**

**1:ZVorherige**

# Arithmetische Operationen, Test Relations und Sonderzeichen

## ! (Fakultät)

Fakultät: *Wert!*

Liefert die Fakultät eines *Werts*.

**MATH**  
WAHRS  
4:!

## ! (Fakultät)

Fakultät: *Liste!*

Liefert die Fakultät von *Listenelementen*.

**MATH**  
WAHRS  
4:!

## ° (Schreibweise in Grad)

Schreibweise in Grad: *Wert*°

Interpretiert *Wert* als Grad; weist Gradangaben im Format DMS aus.

**2nd** **[ANGLE]**  
WINKEL  
1: °

## ⋠ (Bogenmaß)

Bogenmaß: *Winkel*⋠

Interpretiert *Winkel* als Bogenmaß.

**2nd** **[ANGLE]**  
WINKEL  
3:⋠

## T (Transponieren)

Transponieren: *Matrix*<sup>T</sup>

Liefert eine Matrix, in der jedes Element (Zeile, Spalte) mit dem entsprechenden Element (Spalte, Zeile) der *Matrix* getauscht ist.

**2nd** **[MATRIX]**  
MATH  
2: T

## x√

*x. Wurzel*x√*Wert*

Liefert die *x. Wurzel* des *Werts*.

**MATH**  
MATH  
5: x √



$x\sqrt{\phantom{x}}$  $x$ . Wurzel  $x\sqrt{\phantom{x}}$ ListeLiefert die  $x$ . Wurzel der Listenelemente.**MATH**  
**MATH**  
5:  $x\sqrt{\phantom{x}}$  $x\sqrt{\phantom{x}}$ Liste  $x\sqrt{\phantom{x}}$ Wert

Liefert die Listen-Wurzeln des Werts.

**MATH**  
**MATH**  
5:  $x\sqrt{\phantom{x}}$  $x\sqrt{\phantom{x}}$ Liste  $A x\sqrt{\phantom{x}}$ Liste  $B$ Liefert die Liste  $A$ -Wurzeln von Liste  $B$ .**MATH**  
**MATH**  
5:  $x\sqrt{\phantom{x}}$ **3 (dritte Potenz)**Dritte Potenz: Wert<sup>3</sup>

Liefert die dritte Potenz einer reellen oder komplexen Zahl, eines Ausdrucks, einer Liste oder einer Quadratmatrix.

**MATH**  
**MATH**  
3: 3 $\sqrt[3]{\phantom{x}}$  (Kubikwurzel)Kubikwurzel:  $\sqrt[3]{\phantom{x}}$ (Wert)

Liefert die Kubikwurzel einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

**MATH**  
**MATH**  
4:  $\sqrt[3]{\phantom{x}}$ **= (gleich)**Gleich:  
Wert  $A =$  Wert  $B$ Liefert den Wert 1, wenn  $\text{Wert } A = \text{Wert } B$ . Liefert den Wert 0, wenn  $\text{Wert } A \neq \text{Wert } B$ . Wert  $A$  und Wert  $B$  können reelle oder komplexe Zahlen, Ausdrücke, Listen oder Matrizen sein.**2nd** **[TEST]**  
**TEST**  
1:=

**≠ (ungleich)****Ungleich:**

$$\text{Wert}A \neq \text{Wert}B$$

**2nd** **[TEST]**

TEST

2: ≠

Liefert den Wert 1, wenn  $\text{Wert}A \neq \text{Wert}B$ . Liefert den Wert 0, wenn  $\text{Wert}A = \text{Wert}B$ .  $\text{Wert}A$  und  $\text{Wert}B$  können reelle oder komplexe Zahlen, Ausdrücke, Listen oder Matrizen sein.

**< (kleiner als)****Kleiner als:**

$$\text{Wert}A < \text{Wert}B$$

**2nd** **[TEST]**

TEST

5:&lt;

Liefert den Wert 1, wenn  $\text{Wert}A < \text{Wert}B$ . Liefert den Wert 0, wenn  $\text{Wert}A \geq \text{Wert}B$ .  $\text{Wert}A$  und  $\text{Wert}B$  können reelle oder komplexe Zahlen, Ausdrücke oder Listen sein.

**> (größer als)****Größer als:**

$$\text{Wert}A > \text{Wert}B$$

**2nd** **[TEST]**

TEST

3:&gt;

Liefert den Wert 1, wenn  $\text{Wert}A > \text{Wert}B$ . Liefert den Wert 0, wenn  $\text{Wert}A \leq \text{Wert}B$ .  $\text{Wert}A$  und  $\text{Wert}B$  können reelle oder komplexe Zahlen, Ausdrücke oder Listen sein.

**≤ (kleiner oder gleich)****Kleiner oder gleich:**

$$\text{Wert}A \leq \text{Wert}B$$

**2nd** **[TEST]**

TEST

6: ≤

Liefert den Wert 1, wenn  $\text{Wert}A \leq \text{Wert}B$ . Liefert den Wert 0, wenn  $\text{Wert}A > \text{Wert}B$ .  $\text{Wert}A$  und  $\text{Wert}B$  können reelle oder komplexe Zahlen, Ausdrücke oder Listen sein.

**≥ (größer oder gleich)****Größer oder gleich:**

$$\text{Wert}A \geq \text{Wert}B$$

**2nd** **[TEST]**

TEST

4: ≥

Liefert den Wert 1, wenn  $\text{Wert}A \geq \text{Wert}B$ . Liefert den Wert 0, wenn  $\text{Wert}A < \text{Wert}B$ .  $\text{Wert}A$  und  $\text{Wert}B$  können reelle oder komplexe Zahlen, Ausdrücke oder Listen sein.

**<sup>-1</sup> (invers)**Invers:  $\text{Wert}^{-1}$ **x<sup>-1</sup>**

## **$-1$ (invers)**

Ergibt 1 geteilt durch eine reelle oder komplexe Zahl oder einen Ausdruck.

## **$-1$ (invers)**

Invers:  $Liste^{-1}$   $x^{-1}$

Ergibt 1 geteilt durch die *Listenelemente*.

## **$-1$ (invers)**

Invers:  $Matrix^{-1}$   $x^{-1}$

Liefert die inverse *Matrix*.

## **2 (Quadrat)**

Quadrat:  $Wert^2$   $x^2$

Ergibt den *Wert* multipliziert mit sich selbst. *Wert* kann eine reelle oder komplexe Zahl oder ein Ausdruck sein.

## **2 (Quadrat)**

Quadrat:  $Liste^2$   $x^2$

Liefert die quadrierten Elemente der *Liste*.

## **2 (Quadrat)**

Quadrat:  $Matrix^2$   $x^2$

Ergibt die *Matrix* multipliziert mit sich selbst.

## **^ (Potenz)**

Potenzen:  $Wert^{Potenz}$   $\wedge$

Ergibt die *x*-te Potenz des *Werts*. *Wert* kann eine reelle oder komplexe Zahl oder ein Ausdruck sein.

### **^ (Potenz)**

**Potenzen:**  $Liste^{\wedge}Potenz$



Ergibt die  $x$ -te Potenz der Elemente der *Liste*.

### **^ (Potenz)**

**Potenzen:**  $Wert^{\wedge}Liste$



Ergibt den *Wert* hoch die *Listenelemente*.

### **^ (Potenz)**

**Potenzen:**  $Matrix^{\wedge}Potenz$



Ergibt die  $x$ -te Potenz der Elemente der *Matrix*.

### **- (Negation)**

**Negation:**  $-Wert$



Liefert den negativen Wert einer reellen oder komplexen Zahl, eines Ausdrucks, einer Liste oder einer Matrix.

### **10^( (Potenz von Zehn)**

**Potenz von Zehn:**  $10^{\wedge}(Wert)$



Ergibt die  $x$ -te Potenz von 10. *Wert* kann eine reelle oder komplexe Zahl oder ein Ausdruck sein.

### **10^( (Potenz von Zehn)**

**Potenz von Zehn:**  $10^{\wedge}(Liste)$



Ergibt eine Liste von 10 hoch *Liste*

### **√( (Quadratwurzel)**

**Quadratwurzel:**  $\sqrt{\wedge}(Wert)$



Liefert die Quadratwurzel einer reellen oder komplexen Zahl, eines Ausdrucks oder einer Liste.

### **\* (multiplizieren)**

**Multiplikation:**  
 $WertA * WertB$



**\* (multiplizieren)**

Ergibt *WertA* mal *WertB*.

**\* (multiplizieren)**

Multiplikation:

*Wert*\**Liste*



Ergibt den *Wert* mal jedes *Listenelement*.

**\* (multiplizieren)**

Multiplikation:

*Liste*\**Wert*



Ergibt jedes *Listenelement* mal *Wert*.

**\* (multiplizieren)**

Multiplikation:

*ListeA*\**ListeB*



Ergibt die Elemente der *ListeA* mal die Elemente der *ListeB*.

**\* (multiplizieren)**

Multiplikation:

*Wert*\**Matrix*



Ergibt den *Wert* mal die Elemente der *Matrix*.

**\* (multiplizieren)**

Multiplikation:

*MatrixA*\**MatrixB*



Ergibt *MatrixA* mal *MatrixB*.

**/ (dividieren)**

Division: *WertA*/*WertB*



Ergibt *WertA* geteilt durch *WertB*.

### / (dividieren)

**Division:**  $Liste/Wert$



Ergibt die *Listenelemente* geteilt durch den Wert.

### / (dividieren)

**Division:**  $Wert/Liste$



Ergibt den Wert geteilt durch die *Listenelemente*.

### / (dividieren)

**Division:**  $ListeA/ListeB$



Ergibt die Elemente der *ListeA* geteilt durch die Elemente der *ListeB*.

### + (addieren)

**Addition:**  $WertA+WertB$



Ergibt *WertA* plus *WertB*.

### + (addieren)

**Addition:**  $Liste+Wert$



Ergibt eine Liste, in der der *Wert* zu jedem Element der *Liste* hinzuaddiert ist.

### + (addieren)

**Addition:**  $ListeA+ListeB$



Ergibt die Elemente der *ListeA* plus die Elemente der *ListeB*.

### + (addieren)

**Addition:**  
 $MatrixA+MatrixB$



Ergibt die Elemente der *MatrixA* plus die Elemente der *MatrixB*.

### + (Verkettung)

**Verkettung:**

*String1+String2*



Schreibt zwei oder mehrere Zeichenfolgen hintereinander.

### - (subtrahieren)

**Subtraktion:**

*WertA-WertB*



Subtrahiert *WertB* von *WertA*.

### - (subtrahieren)

**Subtraktion:**

*Wert-Liste*



Subtrahiert die Elemente der *Liste* vom *Wert*

### - (subtrahieren)

**Subtraktion:**

*Liste-Wert*



Subtrahiert den *Wert* von den Elementen der *Liste*.

### - (subtrahieren)

**Subtraktion:**

*ListeA-ListeB*



Subtrahiert die Elemente der *ListeB* von den Elementen der *ListeA*.

### - (subtrahieren)

**Subtraktion:**

*MatrixA-MatrixB*



Subtrahiert die Elemente der *MatrixB* von den Elementen der *MatrixA*.

### ' (Minuten-Schreibweise)

**Minuten-Schreibweise:** *Grad° Minuten'*

*Sekunden"*

[ANGLE]

WINKEL

2:'

Interpretiert das Winkelmaß *Minuten* als Minuten.

## " (Sekunden-Schreibweise)

Sekunden-Schreibweise:  $\text{Grad}^\circ \text{Minuten}'$   
 $\text{Sekunden}''$

ALPHA [""]

Interpretiert das Winkelmaß *Sekunden* als Sekunden.



## Fehlermeldungen:

Wenn der TI-84 Plus CE-T einen Fehler erkennt, gibt er eine Fehlermeldung in der Form eines Menütitels aus, wie z. B. **ERR:SYNTAX** oder **ERR:DOMÄNE**. Diese Tabelle enthält alle Fehlertypen, mögliche Ursachen und Vorschläge für die Beseitigung der Fehler. Den in dieser Tabelle aufgeführten Fehlern ist auf dem Bildschirm Ihres Grafikaschenrechners jeweils ein **ERR:** vorangestellt. Beispielsweise zeigt Ihr Grafikaschenrechner den Menü-Titel **ERR:ARCHIVIERT** an, wenn er einen Fehler vom Typ **ARCHIVIERT** erkennt.

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
<b>ARCHIVIERT</b>	Sie haben versucht, eine archivierte Variable zu verwenden, zu bearbeiten oder zu löschen. So erzeugt beispielsweise der Ausdruck <code>dim(L1)</code> einen Fehler, wenn L1 archiviert ist.
<b>ARCHIV VOLL</b>	Sie haben versucht, eine Variable zu archivieren. Es ist jedoch nicht genug Speicherplatz im Archiv frei.
<b>ARGUMENT</b>	Für eine Funktion oder eine Anweisung wurde nicht die richtige Anzahl an Argumenten angegeben. Die Argumente sind kursiv dargestellt. Die Argumente in Klammern sind optional und müssen nicht eingegeben werden. Mehrere Argumente müssen durch ein Komma (,) voneinander getrennt sein. So kann beispielsweise <b>StdAbw</b> ( <i>Liste</i> [, <i>Häufigkeitsliste</i> ]) als <b>StdAbw</b> (L1) oder als <b>StdAbw</b> (L1,L2) eingegeben werden, da die Häufigkeitsliste oder <i>Häufigkeitsliste</i> optional ist.
<b>UNGÜLTIGE ADRESSE</b>	Sie haben versucht, eine Applikation zu senden oder zu empfangen, und bei der Übertragung ist ein Fehler aufgetreten (z. B. eine elektrische Störung).
<b>SCHLECHTE SCHÄTZUNG</b>	Sie haben bei einer <b>BERECHNEN</b> -Operation eine <b>Schätzung</b> abgegeben, die nicht zwischen <b>Linke Grenze</b> und <b>Rechte Grenze</b> liegt. Sie haben für die Funktion <b>Löse</b> ( oder den Gleichungslöser eine <i>Schätzung</i> abgegeben, die nicht zwischen <i>untere</i> und <i>obere</i> liegt. Ihre Schätzung und mehrere Punkte um die Schätzung herum sind nicht definiert. Untersuchen Sie einen Graphen der Funktion. Wenn die Gleichung eine Lösung hat, ändern Sie die Grenzen und/oder Ihre erste Schätzung.
<b>GRENZE</b>	Sie haben in einer <b>BERECHNEN</b> -Operation oder über <b>Auswählen</b> ( <b>Linke Grenze</b> > <b>Rechte Grenze</b> definiert. In <b>fMin</b> (, <b>fMax</b> (, <b>Löse</b> ( oder im Gleichungslöser haben Sie <i>untere</i> <i>obere</i> eingegeben.

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
ABBRUCH	<p>Sie haben die Taste <b>[ON]</b> gedrückt, um die Ausführung eines Programms abzubrechen, eine <b>ZEICHNEN</b>-Anweisung zu unterbrechen oder die Auswertung eines Ausdrucks zu stoppen.</p>
DATENTYP	<p>Sie haben einen Wert oder eine Variable vom falschen Datentyp eingegeben.</p> <p>Sie haben für eine Funktion (einschließlich implizierte Multiplikation) oder eine Anweisung ein Argument eingegeben, bei dem es sich um einen ungültigen Datentyp handelt, wie z. B. eine komplexe Zahl, wenn eine reelle Zahl erforderlich ist.</p> <p>Sie haben in einem Editor einen nicht zulässigen Typ eingegeben, wie z. B. eine Matrix als Element im Stat-Listeneditor.</p> <p>Sie haben versucht, einen falschen Datentyp (wie z. B. eine Matrix) in einer Liste zu speichern.</p> <p>Sie haben versucht, komplexe Zahlen in die MathPrint™-Vorlage n/d einzugeben.</p>
DIMENSIONSFEHLER	<p>Ihr Taschenrechner zeigt den Fehler <b>ERR:DIMENSIONSFEHLER</b> an, wenn Sie versuchen, eine Operation durchzuführen, die sich auf eine oder mehrere Listen oder Matrizen bezieht, deren Dimensionen nicht übereinstimmen. So ergibt beispielsweise die Multiplikation <math>L1 * L2</math> (mit <math>L1 = \{1,2,3,4,5\}</math> und <math>L2 = \{1,2\}</math>) einen Fehler <b>ERR:DIMENSIONSFEHLER</b>, da die Anzahl der Elemente in <math>L1</math> und <math>L2</math> nicht übereinstimmt.</p> <p>Deaktivieren Sie ggf. Plots, um fortzufahren.</p>
DURCH 0 DIVIDIEREN	<p>Sie haben versucht, durch Null zu teilen. Dieser Fehler wird während des Zeichnens nicht ausgegeben. Der TI-84 Plus EC-T erlaubt nicht definierte Werte in einem Graphen.</p> <ul style="list-style-type: none"> <li>Sie haben versucht, eine lineare Regression mit einer vertikalen Linie durchzuführen.</li> </ul>
Domäne	<p>Sie haben für eine Funktion oder Anweisung ein Argument außerhalb des gültigen Bereichs angegeben. Der TI-84 Plus CE-T erlaubt nicht definierte Werte in einem Graphen.</p> <p>Sie haben versucht, eine logarithmische Regression oder eine Potenzregression mit <math>-X</math> oder eine exponentielle Regression oder Potenzregression mit <math>-Y</math> durchzuführen.</p>

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
	Sie haben versucht, $\Sigma Prn($ oder $\Sigma Int($ mit $pmt2 < pmt1$ zu berechnen.
DUPLIZIEREN	Sie haben versucht, einen doppelten Gruppenname zu erstellen.
Doppelter Name	Eine Variable, die Sie übertragen wollten, kann nicht übertragen werden, weil sich im empfangenden Gerät bereits eine Variable mit diesem Namen befindet.
ABGELAUFEN	Sie haben versucht, eine Applikation mit begrenzter Probezeit auszuführen, deren Probezeitraum bereits abgelaufen ist.
Fehler bei Übertragung	Der TI-84 Plus EC-T konnte ein Element nicht übertragen. Überprüfen Sie, dass das Kabel an beiden Geräten fest angeschlossen ist und dass sich das empfangende Gerät im Empfangsmodus befindet.
	Sie haben <b>[ON]</b> gedrückt, um die laufende Übertragung abzubrechen. Richten Sie beim Senden von Dateien ( <b>[LINK]</b> ) zwischen zwei Grafiktaschenrechnern erst das empfangende und dann das sendende Gerät ein.
ID NICHT GEFUNDEN	Dieser Fehler tritt auf, wenn der Befehl <b>SendID</b> ausgeführt wird, die richtige Grafiktaschenrechner-ID jedoch nicht gefunden wird.
UNZUL. VERSCHACHELUNG	Sie haben versucht, eine ungültige Funktion in einem Argument einer Funktion zu verwenden, wie z. B. <b>Folge(</b> innerhalb eines <i>Ausdrucks</i> für <b>Folge(</b> .
INKREMENT	Das Inkrement (Schritt) in <b>Folge(</b> ist 0 oder hat das falsche Zeichen. Der TI-84 Plus EC-T erlaubt nicht definierte Werte in einem Graphen. Das Inkrement in einer <b>For(</b> -Schleife ist 0.
UNGÜLTIG	Sie haben versucht, auf eine Variable zu verweisen oder eine Funktion zu nutzen, wo dies nicht gültig ist. Beispielsweise kann <b>Yn</b> nicht auf <b>Y</b> , <b>Xmin</b> , <b><math>\Delta X</math></b> oder <b>TblStart</b> verweisen. Sie haben im Modus <b>Folge</b> versucht, ein Phasendiagramm zu erstellen, ohne beide Gleichungen des Phasendiagramms zu definieren. Sie haben im Modus <b>Folge</b> versucht, eine rekursive Folge darzustellen, ohne die korrekte Anzahl von Anfangsbedingungen einzugeben. Sie haben im Modus <b>Folge</b> versucht, auf andere

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
<b>DIMENSION UNGÜLTIG</b>	<p>Terme zu verweisen als <math>(n-1)</math> oder <math>(n-2)</math>.</p> <p>Sie haben versucht, einen Grafikstil zuzuweisen, der für den aktuellen Grafikmodus ungültig ist.</p> <p>Sie haben versucht, <b>Auswählen</b>( zu verwenden, ohne mindestens ein xyLinien- oder Streudiagramm ausgewählt (aktiviert) zu haben.</p> <p>Die Fehlermeldung <b>ERR:DIMENSION UNGÜLTIG</b> kann angezeigt werden, wenn Sie versuchen, eine Funktion zu zeichnen, die keine Statistikplot-Funktionen beinhaltet. Der Fehler lässt sich durch Ausschalten der statistischen Plots beheben. Um die statistischen Plots auszuschalten, drücken Sie <b>2nd</b> und wählen Sie anschließend <b>4:PlotsAus</b>.</p> <p>Sie haben für eine Listendimension einen anderen Wert als ein Ganzzahl zwischen 1 und 999 angegeben.</p> <p>Sie haben für eine Matrixdimension einen anderen Wert als ein Ganzzahl zwischen 1 und 99 angegeben.</p> <p>Sie haben versucht, eine nicht quadratische Matrix zu invertieren.</p>
<b>ITERATIONEN</b>	<p>Für die Funktion <b>Löse</b>( oder den Gleichungslöser ist die maximale Anzahl zulässiger Iterationen überschritten. Untersuchen Sie einen Graphen der Funktion. Wenn die Gleichung eine Lösung hat, ändern Sie die Grenzen oder Ihre erste Schätzung oder beides.</p> <p>Die maximale Anzahl zulässiger Iterationen für <b>irr</b>( wurde überschritten.</p> <p>Bei der Berechnung von I% wurde die maximale Anzahl von Iterationen überschritten.</p>
<b>BEZEICHNUNG</b>	<p>Die Bezeichnung in der Anweisung <b>Goto</b> ist nicht mit einer <b>Lbl</b>-Anweisung im Programm definiert.</p>
<b>LINK L1 (oder eine beliebige andere Datei) to Restore</b>	<p>Der Taschenrechner wurde zu Testzwecken deaktiviert. Um die volle Funktionalität wiederherzustellen, nutzen Sie die TI Connect™ CE Software, um eine Datei von Ihrem Computer auf Ihren Taschenrechner herunterzuladen oder übertragen Sie eine Datei von einem anderen TI-84 Plus CE-T auf Ihren Taschenrechner.</p>
<b>SPEICHER</b>	<p>Der Speicher ist für die Ausführung der Anweisung oder Funktion nicht ausreichend. Sie müssen Elemente aus dem Speicher löschen, bevor Sie die</p>

**FEHLERTYP****Mögliche Ursachen und Lösungsvorschläge**

Anweisung oder Funktion ausführen können.

Dieser Fehler wird durch rekursive Fehler ausgelöst, beispielsweise durch Zeichnen der Gleichung  $Y1=Y1$ .

Der Fehler kann auch durch eine Verzweigung einer **If/Then-**, **For(-)**, **While-** oder **Repeat-**Schleife mit **Goto** verursacht werden, da die **Ende-**Anweisung, die die Schleife beendet, nie erreicht wird.

Versuch, eine Matrix mit mehr als 400 Zellen zu erstellen.

**SpeicherVoll**

Sie können ein Element nicht übertragen, weil der Speicherplatz im empfangenden Gerät nicht ausreicht. Sie können das Element überspringen oder den Empfangsmodus beenden.

Bei einer Sicherung des Speichers reicht der Speicher des empfangenden Geräts nicht aus, um alle Elemente aus dem Speicher des sendenden Geräts zu empfangen. Eine Meldung zeigt an, wie viel Bytes im sendenden Gerät gelöscht werden müssen, um den Speicher zu sichern. Löschen Sie einzelne Elemente und versuchen Sie es erneut.

**MODUS**

Sie haben versucht, eine Fenstervariable in einem anderen Grafikmodus zu speichern oder eine Anweisung auszuführen, während sich das Gerät im falschen Modus befunden hat (z. B. **ZchInv** in einem anderen Grafikmodus als **Funk**).

**KEINE ZEICHENÄND.**

Die Funktion **Löse(** oder der Gleichungslöser konnte keine Änderung des Vorzeichens erkennen.

Sie haben versucht, **I%** zu berechnen, wenn **FV**, (**N PMT**) und **PV** 0 sind oder wenn **FV**, (**N PMT**) und **PV** 0 sind.

Sie haben versucht, **irr(** zu berechnen, wenn weder **CFListe** noch **CFO** > 0 ist oder wenn weder **CFListe** noch **CFO** < 0 ist.

**NICHTREELLE ANTW.**

Das Ergebnis einer Berechnung im Modus **Reell** ist ein komplexes Ergebnis. Der TI-84 Plus EC-T erlaubt nicht definierte Werte in einem Graphen.

**ÜBERLAUF**

Sie haben eine Zahl außerhalb des Grafiktaschenrechnerbereichs einzugeben versucht oder berechnet. Der TI-84 Plus EC-T erlaubt nicht definierte Werte in einem Graphen.

**RESERVIERT**

Sie haben versucht, eine Systemvariable falsch zu verwenden.

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
SINGULÄRE MATRIX	<p>Eine singuläre Matrix (Determinante = 0) ist als Argument für <math>-1</math> nicht gültig.</p> <p>Die Anweisung <b>SinReg</b> oder eine Polynomregression hat eine singuläre Matrix (Determinante = 0) ergeben, da der Algorithmus keine Lösung gefunden hat oder weil es keine Lösung gibt.</p> <p>Der TI-84 Plus EC-T erlaubt nicht definierte Werte in einem Graphen.</p>
SINGULARITÄT	<p><i>Ausdruck</i> in der Funktion <b>Löse(</b> oder im Gleichungslöser enthält eine Singularität (einen Punkt, an dem die Funktion nicht definiert ist). Untersuchen Sie einen Graphen der Funktion. Wenn die Gleichung eine Lösung hat, ändern Sie die Grenzen oder Ihre erste Schätzung oder beides.</p>
STAT	<p>Sie haben versucht, eine statistische Berechnung mit dafür ungeeigneten Listen durchzuführen.</p> <p>Für statistische Analysen müssen mindestens zwei Datenpunkte vorliegen.</p> <p><b>Mittel-Mittel</b> muss in jeder Partition mindestens drei Punkte haben.</p> <p>Wenn Sie eine Häufigkeitsliste verwenden, müssen deren Elemente 0 sein.</p> <p>Bei einem Histogramm muss <math>(X_{\max} - X_{\min}) / X_{\text{sc1}}</math> zwischen 0 und 131 liegen.</p>
STAT PLOT	<p>Sie haben versucht, einen Graphen anzuzeigen, während ein statistischer Plot, der eine nicht definierte Liste verwendet, eingeschaltet ist.</p>
SYNTAX	<p>Der Befehl enthält einen Syntaxfehler. Suchen Sie nach falsch gesetzten Funktionen, Argumenten, Klammern oder Kommata.</p> <p><b>StdAbw</b>(<i>Liste</i>[,<i>Häufigkeitsliste</i>]) beispielsweise ist eine Funktion des TI-84 Plus CE-T. Die Argumente sind kursiv dargestellt. Die Argumente in Klammern sind optional und müssen nicht eingegeben werden. Mehrere Argumente müssen durch ein Komma (,) voneinander getrennt sein. So kann beispielsweise <b>StdAbw</b>(<i>Liste</i>[,<i>Häufigkeitsliste</i>]) als StdAbw(L1) oder als StdAbw(L1,L2) eingegeben werden, da die Häufigkeitsliste oder <i>Häufigkeitsliste</i> optional ist.</p>
TOLERANZ N. ERFÜLLT	<p>Sie haben eine Toleranz festgelegt, zu der der Algorithmus kein exaktes Ergebnis liefern kann.</p>
UNDEFINIERT	<p>Sie haben auf eine Variable verwiesen, die aktuell</p>

FEHLERTYP	Mögliche Ursachen und Lösungsvorschläge
ÜBERPRÜFUNG	<p>nicht definiert ist. Beispielsweise haben Sie auf eine statistische Variable verwiesen, für die es keine aktuelle Berechnung gibt, weil beispielsweise eine Liste bearbeitet wurde, oder Sie haben auf eine Variable verweisen, die für die aktuelle Berechnung ungültig ist, wie z. B. <b>a</b> nach <b>Mittel-Mittel</b>.</p> <p>Elektrische Störungen haben zum Abbruch der Verbindung geführt, oder dieser Grafiktaschenrechner ist nicht berechtigt, die Anwendung auszuführen.</p>
VARIABLE	<p>Sie haben versucht, eine Variable zu archivieren, die nicht archiviert werden kann, oder Sie haben versucht, eine Applikation oder Gruppe zu dearchivieren.</p> <p>Zu den Variablen, die nicht archiviert werden können, zählen:</p> <p>Reelle Zahlen <b>LRESID, R, T, X, Y, Theta</b>, statistische Variablen unter <b>Variablen</b>, das Menü<b>STATISTIK</b>, <b>Yvars</b> und die <b>AppldListe</b>.</p>
VERSION	<p>Sie haben versucht, eine nicht kompatible Variablenversion von einem anderen Grafiktaschenrechner zu empfangen.</p> <p>Ein Programm enthält möglicherweise Befehle, die in der BS-Version Ihres Grafiktaschenrechners nicht unterstützt werden. Verwenden Sie immer das aktuellste Betriebssystem. Programme können zwischen dem TI-84 Plus CE und dem TI-84 Plus ausgetauscht werden, jedoch kann ein Versionsfehler angezeigt werden, wenn ein neues TI-84 Plus CE-Programm an einen hochauflösenden Graphenbereich angepasst werden muss.</p>
FENSTERBEREICH	<p>Es liegt ein Problem mit den Fenstervariablen vor. Sie haben <b>Xmax Xmin</b> oder <b>Ymax Ymin</b> definiert. Sie haben <b>θmax θmin</b> und <b>θSchritt &gt; 0</b> (oder umgekehrt) definiert. Sie haben versucht, <b>TSchritt=0</b> zu definieren. Sie haben <b>Tmax Tmin</b> und <b>Tschritt &gt; 0</b> (oder umgekehrt) definiert.</p> <p>Fenstervariablen sind zu klein oder zu groß, um den Graphen korrekt darzustellen. Möglicherweise haben Sie versucht, an einen Punkt zu zoomen, der außerhalb des numerischen Bereichs des TI-84 Plus CE-T liegt.</p>

**FEHLERTYP****Mögliche Ursachen und Lösungsvorschläge****ZOOM**

In **ZBox** ist anstelle einer Box ein Punkt oder eine Gerade definiert.

Eine **ZOOM**-Operation hat einen mathematischen Fehler zurückgegeben.



# Allgemeine Informationen

## **Online-Hilfe**

[education.ti.com/eguide](http://education.ti.com/eguide)

Wählen Sie Ihr Land aus, um weitere Produktinformationen zu erhalten.

## **Kontakt mit TI Support aufnehmen**

[education.ti.com/ti-cares](http://education.ti.com/ti-cares)

Wählen Sie Ihr Land aus, um auf technische und sonstige Support-Ressourcen zuzugreifen.

## **Service- und Garantieinformationen**

[education.ti.com/warranty](http://education.ti.com/warranty)

Wählen Sie für Informationen zur Dauer und den Bedingungen der Garantie bzw. zum Produktservice Ihr Land aus.

Eingeschränkte Garantie. Diese Garantie hat keine Auswirkungen auf Ihre gesetzlichen Rechte.