

# Die wichtigsten GeoGebra-Befehle

Quelle: <https://wiki.geogebra.org/de/Befehle>

## Operatoren und vordefinierte Funktionen

### Sonderzeichen

- Du erhältst  $\pi$  mit der Tastenkombination ALT + p.
- Die Eulersche Zahl  $e$  kann durch die Tastenkombination ALT + e eingegeben werden.
- Das Gradzeichen  $^\circ$  erhältst du mit ALT + o.
- Die imaginäre Einheit  $i$  erhältst du mit der Tastenkombination ALT + i.

### Operatoren

Addition	+	Subtraktion	-
Multiplikation	* oder Leerzeichen	Division	/
Potenzieren	^ oder Hochstellung	Fakultät	!
Skalarprodukt	* oder Leerzeichen	Kreuzprodukt	$\otimes$

### Koordinaten und Achsen

x-Koordinate eines Punktes	x( )	Name der x-Achse	xAchse
y-Koordinate eines Punktes	y( )	Name der y-Achse	yAchse
z-Koordinate eines Punktes	z( )	Name der z-Achse	zAchse

### Vordefinierte Funktionen

Absolutbetrag	abs( )
Quadratwurzel	sqrt( )
Kubikwurzel	cbrt( )
Exponentialfunktion	exp( ) oder $e^x$
Natürlicher Logarithmus zur Basis e	ln( ) oder log( )
Logarithmus zur Basis 10	lg( )
Logarithmus von x zur Basis b	log(b, x)
Sinus	sin( )
Kosinus	cos( )
Tangens	tan( )
Arcus Sinus (Umkehrfunktion des Sinus) - Antwort in Radiant	arcsin( )
Arcus Sinus (Umkehrfunktion des Sinus) - Antwort in Grad	arcsind( )
Arcus Kosinus (Umkehrfunktion des Kosinus) - Antwort in Radiant	arccos( )
Arcus Kosinus (Umkehrfunktion des Kosinus) - Antwort in Grad	arccosd( )
Arcus Tangens (Umkehrfunktion des Tangens) - Antwort in Radiant	arctan( )
Arcus Tangens (Umkehrfunktion des Tangens) - Antwort in Grad	arctand( )

### Punkte und Vielecke

---

#### **Punkt**[ <Objekt> ]

Erzeugt einen neuen Punkt auf dem Objekt. Der entstandene Punkt kann entlang des Pfades bewegt werden.

#### **Punkt**[ <Punkt>, <Vektor> ]

Erzeugt einen neuen Punkt durch Abtragen des Vektors vom gegebenen Punkt.

#### **Mittelpunkt**[ <Punkt>, <Punkt> ]

Bestimmt den Mittelpunkt zwischen zwei Punkten.

**Beispiel:** Mittelpunkt[(1, 1), (5, 1)] liefert (3, 1).

#### **Mittelpunkt**[ <Intervall> ]

Bestimmt den Mittelpunkt des angegebenen Intervalls (als Zahl).

**Beispiel:** Mittelpunkt[2 < x < 4] liefert 3.

#### **Mittelpunkt**[ <Strecke> ]

Bestimmt den Mittelpunkt zur gegebenen Strecke.

#### **Vieleck**[ <Punkt>, ..., <Punkt> ]

Erzeugt ein Vieleck mit den gegebenen Eckpunkten.

**Beispiel:** Vieleck[(1, 1), (3, 0), (3, 2), (0, 4)] erzeugt ein Viereck.

#### **Vieleck**[ <Punkt>, <Punkt>, <Anzahl der Ecken> ]

Erzeugt ein regelmäßiges Vieleck mit  $n$  Ecken.

**Beispiel:** Vieleck[(1, 1), (4, 1), 6] erzeugt ein Sechseck.

#### **Vieleck**[ <Liste von Punkten> ]

Erzeugt ein Vieleck mit den Punkten in der Liste als Eckpunkte.

**Beispiel:** Vieleck[{(0, 0), (2, 1), (1, 3)}] erzeugt ein Dreieck.

### Geraden und Strecken

---

#### **Gerade**[ <Punkt>, <Punkt> ]

Erzeugt eine Gerade durch die beide Punkte  $A$  und  $B$ .

#### **Gerade**[ <Punkt>, <Parallele Gerade> ]

Erzeugt eine Gerade durch den Punkt, die parallel zur gegebenen Gerade ist.

#### **Gerade**[ <Punkt>, <Richtungsvektor> ]

Erzeugt eine Gerade durch den Punkt mit dem angegebenen Richtungsvektor.

#### **Steigung**[ <Gerade> ]

Berechnet die Steigung der Gerade.

**Anmerkung:** Zusätzlich wird das Steigungsdreieck eingezeichnet, die Größe kann im Eigenschaften-Dialog (Registerkarte: Darstellung) geändert werden.

#### **Strecke**[ <Punkt>, <Punkt> ]

Erzeugt eine Strecke zwischen zwei Punkten.

#### **Strahl**[ <Anfangspunkt>, <Punkt> ]

Erzeugt einen Strahl mit dem gegebenen Anfangspunkt durch den gegebenen Punkt.

#### **Strahl**[ <Anfangspunkt>, <Richtungsvektor> ]

Erzeugt einen Strahl mit dem gegebenen Anfangspunkt und dem gegebenen Richtungsvektor.

## Senkrechte

---

**Senkrechte** [ <Punkt>, <Senkrechte Gerade> ]

Erzeugt eine Gerade durch den Punkt und senkrecht zur gegebenen Geraden.

**Senkrechte** [ <Punkt>, <Senkrechte Strecke> ]

Erzeugt eine Gerade durch den Punkt und senkrecht zur gegebenen Strecke.

**Senkrechte** [ <Punkt>, <Vektor> ]

Erzeugt eine Gerade durch den Punkt und senkrecht zum gegebenen Vektor.

**Senkrechte** [ <Punkt>, <Ebene> ]

Erzeugt eine Gerade durch den Punkt, die normal auf die Ebene steht.

**Senkrechte** [ <Punkt>, <Richtung>, <Richtung> ]

Erzeugt eine Gerade durch den Punkt, die normal auf die beiden Richtungen (das können Geraden oder Vektoren sein) steht.

**Mittelsenkrechte** [ <Strecke> ]

Erzeugt die Mittelsenkrechte der Strecke.

**Mittelsenkrechte** [ <Punkt>, <Punkt> ]

Erzeugt die Mittelsenkrechte einer Strecke zwischen zwei Punkten.

## Abstand und Länge

---

**Anmerkung:** Siehe auch das Werkzeug  Abstand oder Länge.

**Abstand** [ <Punkt>, <Objekt> ]

Berechnet den Abstand zwischen dem Punkt und dem Objekt.

**Anmerkung:** Der Befehl funktioniert für Punkte, Strecken, Geraden, Kegelschnitte, Funktionen und implizite Kurven. Bei Funktionen wird ein numerischer Algorithmus verwendet, welcher am besten bei Polynomen funktioniert.

**Abstand** [ <Gerade> , <Gerade> ]

Berechnet den Abstand zwischen zwei parallelen Geraden.

**Anmerkung:** Der Abstand zwischen zwei schneidenden Geraden ist 0. Daher ist dieser Befehl nur für parallele Geraden sinnvoll.

**Abstand** [ <Punkt>, <Punkt> ]

Berechnet den Abstand zwischen zwei Punkten.

**Länge** [ <Objekt> ]

Berechnet die Länge des Objekts.

**Beispiele:**

- Länge [ <Vektor> ] berechnet die Länge des Vektors.
- Länge [ <Punkt> ] berechnet die Länge des Ortsvektors (Verbindungsvektor Punkt-Koordinatenursprung) des Punktes.
- Länge [ <Liste> ] berechnet die Länge der Liste, also die Anzahl der Listenelemente.
- Länge [ <Text> ] berechnet die Länge des Textes, also die Anzahl der Buchstaben.

**Länge** [ <Funktion>, <Startwert>, <Endwert> ]

Berechnet die Länge des Funktionsgraphen im angegebenen Intervall.

**Beispiel:** Länge [  $2x$ , 0, 1 ] berechnet 2.24, was ungefähr  $\sqrt{5}$  entspricht.

## Vektor und Einheitsvektor (Befehle in allen Ansichten)

---

**Anmerkung:** Siehe auch das Vektor-Werkzeug.

**Vektor**[ <Punkt> ]

Erzeugt den Ortsvektor des Punktes.

**Beispiel:** Vektor[(3, 2)] erzeugt  $u = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ .

**Vektor**[ <Anfangspunkt>, <Endpunkt> ]

Erzeugt einen Vektor mit Anfangspunkt und Endpunkt.

**Beispiel:** Vektor[(1, 1), (3, 4)] erzeugt  $u = u = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$

**Einheitsvektor**[ <Vektor> ]

Erzeugt einen Vektor mit Länge 1, der dieselbe Richtung und Orientierung wie der angegebene Vektor hat. Der Vektor muss zuvor definiert werden.

**Beispiel:** Sei  $v = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$ , **Einheitsvektor**[v] erzeugt  $\begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$ .

**Einheitsvektor**[ <Gerade> ]

Erzeugt den Richtungsvektor der Gerade mit Länge 1.

**Einheitsvektor**[ <Strecke> ]

Erzeugt den Richtungsvektor der Strecke mit Länge 1.

## Ebenen

---

**Ebene**[ <Vieleck> ]

Erzeugt eine Ebene durch das Vieleck.

**Ebene**[ <Kegelschnitt> ]

Erzeugt eine Ebene durch den Kegelschnitt.

**Ebene**[ <Punkt>, <Ebene> ]

Erzeugt eine Ebene durch den Punkt und parallel zur gegebenen Ebene.

**Ebene**[ <Punkt>, <Gerade> ]

Erzeugt eine Ebene durch den Punkt und durch die Gerade.

**Ebene**[ <Gerade>, <Gerade> ]

Erzeugt eine Ebene durch die gegebenen Geraden, wenn die beiden Geraden in derselben Ebene liegen.

**Ebene**[ <Punkt>, <Punkt>, <Punkt> ]

Erzeugt eine Ebene durch die drei Punkte.

## Skalar- und Kreuzprodukt

---

**Skalarprodukt**[ <Vektor>, <Vektor> ]

Berechnet das Skalarprodukt zweier Vektoren.

**Beispiel:** Die Eingabe Skalarprodukt[{1, 3, 2},{0, 3, -2}] ergibt 5.

**Kreuzprodukt**[ <Vektor u>, <Vektor v> ]

Berechnet das Kreuzprodukt der Vektoren  $u$  und  $v$ .

Beispiel: Kreuzprodukt[{1, 3, 2}, {0, 3, -2}] liefert  $\{-12, 2, 3\}$ .

**Anmerkung:** In der Eingabezeile können Sie stattdessen auch  $u \otimes v$  verwenden.

## Normalvektor, Einheitsnormalvektor und Normalebenen

---

**Normalvektor**[ <Gerade> ]

(2D) Liefert den Normalvektor der Geraden.

**Normalvektor**[ <Strecke> ]

(2D) Liefert den Normalvektor der Strecke. Dieser besitzt die selbe Länge wie das Segment.

**Normalvektor**[ <Vektor> ]

(2D) Liefert den Normalvektor des gegebenen Vektors.

**Normalvektor**[ <Ebene> ]

(3D) Liefert den Normalvektor der gegebenen Ebene.

**Beispiel:** Normalvektor[ xyEbene ] erzeugt den Normalvektor  $u = (0, 0, 1)$  der xy-Ebene.

**Einheitsnormalvektor**[ < Vektor > ]

Erzeugt den Normalvektor des angegebenen Vektors mit Länge 1. Der Vektor muss zuvor definiert werden.

**Weitere Formen:**

Einheitsnormalvektor[ < Gerade > ], Einheitsnormalvektor[ < Strecke > ]

**Einheitsnormalvektor**[ < Ebene > ]

Berechnet den Einheitsnormalvektor der gegebenen Ebene.

**Normalebene**[ <Punkt>, <Gerade> ]

Erzeugt eine Ebene durch den gegebenen Punkt und normal zur gegebenen Geraden.

**Normalebene**[ <Punkt>, <Vektor> ]

Erzeugt eine Ebene durch den gegebenen Punkt und normal zu dem gegebenen Vektor.

## Winkel

---

**Anmerkung:** Siehe auch Winkel-Werkzeuge

**Winkel**[ <Objekt> ]

**Vektor als Objekt:** Berechnet den Winkel zwischen dem Vektor und der *x-Achse*.

**Punkt als Objekt:** Berechnet den Winkel zwischen dem Ortsvektor des Punkts und der *x-Achse*.

**Zahl:** Konvertiert die Zahl zu einem Winkel, wobei das Ergebnis zwischen  $0^\circ$  und  $360^\circ$  bzw. 0 und  $2\pi$  liegt.

**Winkel**[ <Vektor>, <Vektor> ]

Erzeugt den Winkel zwischen den beiden Vektoren, wobei das Ergebnis zwischen  $0^\circ$  und  $360^\circ$  bzw. 0 und  $2\pi$  liegt.

**Beispiel:** Winkel[Vektor[(1, 1)], Vektor[(2, 5)]] ergibt  $23.2^\circ$  oder den entsprechenden Wert in *Radian*.

**Winkel**[ <Gerade>, <Gerade> ]

Erzeugt den Winkel zwischen den Richtungsvektoren der beiden Geraden, wobei das Ergebnis zwischen  $0^\circ$  und  $360^\circ$  bzw. 0 und  $2\pi$  liegt.

**Beispiel:** Winkel[ $y = x + 2$ ,  $y = 2x + 3$ ] ergibt  $18.43^\circ$  oder den entsprechenden Wert in *Radian*.

**Winkel**[ <Gerade>, <Ebene> ]

Erzeugt den Winkel zwischen der Geraden und der Ebene.

**Beispiel:** Winkel[Gerade[(1, 2, 3), (-2, -2, 0)],  $z = 0$ ] liefert  $30.96^\circ$  oder den entsprechenden Wert in *Radian*.

**Winkel**[ <Ebene>, <Ebene> ]

Erzeugt den Winkel zwischen den zwei gegebenen Ebenen.

**Beispiel:** Winkel[ $2x - y + z = 0$ ,  $z = 0$ ] liefert  $114.09^\circ$  oder den entsprechenden Wert in *Radian*.

## Bedingte Funktionen

Du kannst den booleschen Befehl **Wenn** verwenden, um eine bedingte Funktion zu definieren. Bedingte Funktionen können auch als Argumente in Befehlen wie Ableitung, Integral und Schneide verwendet werden.

### Beispiele:

- Die Eingabe  $f(x) = \text{Wenn}[x < 3, \sin(x), x^2]$  erzeugt eine Funktion, welche für  $x < 3$  die Werte von  $\sin(x)$  und für  $x \geq 3$  die Werte von  $x^2$  annimmt.
- Die Eingabe  $f(x) = \text{Wenn}[0 \leq x \leq 3, x^3]$  erzeugt eine Funktion, welche für  $x$  zwischen 0 und 3 die Werte von  $x^3$  annimmt und für  $x \geq 3$  oder  $x \leq 0$  undefiniert ist.

## Polynomfunktionen (Ganzrationale Funktionen)

### Polynom[ <Funktion> ]

Berechnet die erweiterte Form der Polynomfunktion.

**Beispiel:**  $\text{Polynom}[(x - 3)^2]$  liefert  $x^2 - 6x + 9$ .

### Polynom[ <Liste von Punkten> ]

Erzeugt ein Interpolationspolynom vom Grad  $n - 1$  für die gegebenen  $n$  Punkte.

**Beispiel:**  $\text{Polynom}[\{(1, 1), (2, 3), (3, 6)\}]$  liefert  $0.5x^2 + 0.5x$ .

### Faktoren[ <Polynom> ]

Liefert eine Liste von Listen  $\{Faktor, Exponent\}$ , welche die einzelnen Faktoren des Polynoms enthält. Das Produkt dieser Faktoren potenziert mit dem entsprechenden *Exponent* entspricht dem gegebenen Polynom. Die Faktoren werden aufsteigend nach dem Grad sortiert.

### Koeffizienten[ <Polynom> ]

Erzeugt für ein Polynom  $a_k \cdot x^k + a_{k-1} \cdot x^{k-1} + \dots + a_1 \cdot x + a_0$  die Liste aller Koeffizienten  $a_k, a_{k-1}, \dots, a_1, a_0$

### Grad[ <Polynom> ]

Gibt den Grad des Polynoms (bzgl. der Hauptvariablen) an.

## Trigonometrische Funktionen

### TrigVereinfache[ <Ausdruck> ]

Vereinfacht den gegebenen trigonometrischen Ausdruck.

## Tangente

**Anmerkung:** Siehe auch das Tangenten-Werkzeug .

### Tangente[ <Punkt>, <Funktion> ]

Erzeugt die Tangente an die Funktion in  $x = x(A)$ .

**Anmerkung:**  $x(A)$  ist die  $x$ -Koordinate von dem gegebenen Punkt  $A$ .

**Beispiel:**  $\text{Tangente}[(1, 0), x^2]$  erzeugt  $y = 2x - 1$ .

### Tangente[ <Punkt auf Kurve>, <Kurve> ]

Erzeugt die Tangente durch den Punkt an die Kurve.

### Tangente[ <x-Wert>, <Funktion> ]

Erzeugt die Tangente an die Funktion an der Stelle  $x = x\text{-Wert}$ .

**Beispiel:**  $\text{Tangente}[1, x^2]$  erzeugt  $y = 2x - 1$ .

## Kurvenuntersuchung

---

### Schnittpunkte

**Schneide**[ <Objekt>, <Objekt> ]

Berechnet den Schnittpunkt zweier Objekte.

**Schneide**[ <Objekt>, <Objekt>, <Anfangspunkt> ]

Berechnet den Schnittpunkt zweier Objekte mit dem newtonschen Näherungsverfahren und einem Anfangspunkt.

**Schneide**[ <Funktion>, <Funktion>, <Anfangswert>, <Endwert> ]

Berechnet die Schnittpunkte der beiden Funktionen im angegebenen Intervall.

**Schneide**[ <Funktion>, <Funktion> ]

**(CAS)** Liefert eine Liste, welche die Schnittpunkte von zwei Objekten enthält.

### Nullstellen

**Nullstelle**[ <Polynom> ]

Erzeugt alle Nullstellen eines Polynoms als Schnittpunkte des Funktionsgraphen und der  $x$ -Achse.

**Nullstelle**[ <Funktion>, <Startwert> ]

Berechnet eine Nullstelle der Funktion nach der Newton Methode mit dem Startwert  $a$ .

**Nullstelle**[ <Funktion>, <Startwert>, <Endwert> ]

Sei  $a$  der Startwert und  $b$  der Endwert. Dann berechnet dieser Befehl die Nullstelle einer Funktion im Intervall  $[a, b]$  (regula falsi).

**Nullstellen**[ <Funktion>, <Startwert>, <Endwert> ]

Berechnet die Nullstellen der Funktion im angegebenen Intervall. Dabei muss die Funktion in diesem Intervall stetig sein. Manchmal kann es sein, dass nicht alle Nullstellen gefunden werden, da es sich um einen numerischen Algorithmus handelt.

### Maxima und Minima

**Extremum**[ <Polynomfunktion> ]

Erzeugt alle lokalen Extrema des Polynoms als Punkte auf dem Funktionsgraphen.

**Extremum**[ <Funktion>, <Startwert>, <Endwert> ]

Erzeugt (numerisch) das Extremum der Funktion im offenen Intervall ( <Startwert>, <Endwert> ).

**Max**[ <Funktion>, <Startwert>, <Endwert> ]

Sei  $a$  der Startwert und  $b$  der Endwert. Dann berechnet dieser Befehl den höchsten Punkt einer Funktion im Intervall  $[a, b]$ . Dabei sollte die Funktion stetig sein und nur einen Hochpunkt in diesem Intervall haben.

**Min**[ <Funktion>, <Startwert>, <Endwert> ]

Sei  $a$  der Startwert und  $b$  der Endwert. Dann berechnet dieser Befehl den tiefsten Punkt der Funktion im Intervall  $[a, b]$ . Dabei soll die Funktion stetig sein und nur einen Tiefpunkt in diesem Intervall haben.

### Wendepunkt

**Wendepunkt**[ <Polynom> ]

Berechnet alle Wendepunkte des Polynoms als Punkte auf dem Funktionsgraphen.

## Grenzwerte

---

**Anmerkung:** Nicht alle Grenzwerte können von GeoGebra berechnet werden. In diesen Fällen wird "?" zurückgegeben (und auch falls das korrekte Resultat "undefiniert" lautet).

**Grenzwert**[ <Funktion>, <Wert > ]

Berechnet den Grenzwert der Funktion beim gegebenen Wert. (Dieser kann auch unendlich sein.)

**LinksseitigerGrenzwert**[ <Funktion>, <Wert> ]

Berechnet den linksseitigen Grenzwert der Funktion beim gegebenen Wert.

**RechtsseitigerGrenzwert**[ <Funktion>, <Wert> ]

Berechnet den rechtsseitigen Grenzwert der Funktion beim gegebenen Wert.

**Asymptote**[ <Funktion> ]

GeoGebra wird versuchen alle Asymptoten der Funktion zu finden und als Liste auszugeben. Möglicherweise werden aber nicht alle gefunden, zum Beispiel die vertikalen Asymptoten von nicht-rationalen Funktionen wie  $\ln(x)$ .

## Ableitung (CAS- und Algebra-Ansicht)

---

**Ableitung**[ <Funktion> ]

Liefert die Ableitung der Funktion.

**Beispiel:** **Ableitung**[ $x^3 + x^2 + x$ ] liefert  $3x^2 + 2x + 1$ .

**Ableitung**[ <Funktion>, <Grad der Ableitung> ]

Liefert die  $n$ -te Ableitung der Funktion, wobei  $n$  gleich <Grad der Ableitung> ist.

**Beispiel:** **Ableitung**[ $x^3 + x^2 + x$ , 2] liefert  $6x + 2$ .

**Ableitung**[ <Funktion>, <Variable> ]

Liefert die partielle Ableitung der Funktion nach der gegebenen Variable.

**Beispiel:** **Ableitung**[ $x^3 y^2 + y^2 + x y$ ,  $y$ ] liefert  $2x^3y + x + 2y$ .

**Anmerkungen :**

- Du kannst  $f'(x)$  statt **Ableitung**[ $f$ ] schreiben, oder  $f''(x)$  statt **Ableitung**[ $f$ , 2] und so weiter.
- Den Ausdruck  $f'(x)$  kann man auch verwenden, um die Ableitung an einer bestimmten Stelle zu berechnen. So gibt  $f'(2)$  den Ableitungswert an der Stelle 2 als Zahl aus.

## Integralrechnung

---

### Zerlegungsummen

**Obersumme** [ <Funktion>, <Startwert>, <Endwert>, <Anzahl der Rechtecke> ]

Berechnet die Obersumme der Funktion im Intervall [*Startwert*, *Endwert*] mit *n* Rechtecken.

**Untersumme** [ <Funktion>, <Startwert>, <Endwert>, <Anzahl der Rechtecke> ]

Berechnet die Untersumme der Funktion im Intervall [*Startwert*, *Endwert*] mit *n* Rechtecken.

**Trapezsumme** [ <Funktion>, <Startwert>, <Endwert>, <Anzahl der Trapeze> ]

Berechnet die Trapezsumme der Funktion im Intervall [*Startwert*, *Endwert*] mit *n* Trapezen.

**LinkeSumme** [ <Funktion>, <Startwert>, <Endwert>, <Anzahl der Rechtecke> ]

Berechnet die linke Summe der Funktion im Intervall mit *n* Rechtecken.

**Rechtecksumme** [ <Funktion>, <Startwert>, <Endwert>, <Anzahl der Rechtecke>, <Position Rechtecksanfang> ]

Berechnet zwischen dem *Startwert* und dem *Endwert* die Summe der *n* Rechtecke der Funktion, wobei deren linke Höhe für jedes Intervall bei einem Bruch *d* ( $0 \leq d \leq 1$ ) beginnt. Falls  $d = 0$  ist dieser Befehl äquivalent zum Befehl *LinkeSumme* und falls  $d = 1$  wird die rechte Rechteckssumme der gegebenen Funktion berechnet.

### Integral

**Integral** [<Funktion>]

Berechnet das unbestimmte Integral der Funktion nach der Hauptvariable.

**Integral** [<Funktion>, <Variable>]

Berechnet die partielle Integration der Funktion nach der angegebenen Variable.

**Integral** [<Funktion>, <Startwert>, <Endwert>]

Berechnet das bestimmte Integral der Funktion nach der Hauptvariable im Intervall [*Startwert*, *Endwert*].

**Anmerkung:** Dieser Befehl schattiert auch die Fläche zwischen dem Funktionsgraphen und der *x*-Achse.

**IntegralZwischen** [ <Funktion>, <Funktion>, <Startwert>, <Endwert> ]

Gibt das bestimmte Integral der Differenz  $f(x) - g(x)$  zweier Funktionen *f* und *g* im Intervall [*a*, *b*] an, wobei *a* der Startwert und *b* der Endwert ist.

**Anmerkung:** Dieser Befehl schattiert auch den Bereich zwischen dem Funktionsgraphen von *f* und *g*.

**NIntegral** [ <Funktion>, <Startwert für x>, <Endwert für x> ]

Sei *a* der *Startwert für x*, *b* der *Endwert für x* und *f* die *Funktion*. Der *NIntegral*-Befehl

berechnet das bestimmte Integral  $\int_a^b f(x)dx$  numerisch.

### **Punktliste**

Für eine Regressionskurve benötigst du eine Liste von Punkten. Du erhältst sie am einfachsten mit Hilfe einer Tabelle, wenn du die x- und y-Werte markierst und dann mit der rechten Maustaste "Erzeugen – Liste von Punkten" wählst. Du kannst sie aber aus den Punkten A, B, C, ... auch mit {A,B,C,...} selbst erzeugen.

### **Trendlinie[ <Liste von Punkten> ]**

Berechnet die Trendlinie der gegebenen Punkte. Dabei wird die Summe der Quadratabstände in y-Richtung minimiert.

### **TrendExp [ <Liste von Punkten> ]**

Berechnet als Regressionskurve der Punkte eine Funktion der Form  $a \cdot e^{k \cdot x}$

### **TrendExp2[ <Liste von Punkten> ]**

Berechnet als Regressionskurve der Punkte eine Funktion der Form  $a \cdot b^x$  (wie TrendExp[ <Liste von Punkten> ], für jene Benutzer, die die Bedeutung von e nicht kennen).

### **TrendPoly[ <Liste von Punkten>, <Grad des Polynoms> ]**

Berechnet das Regressionspolynom  $n$ -ten Grades.

Für ein Polynom  $n$ -ten Grades muss die Liste mindestens  $n + 1$  Punkte haben.

### **TrendPot[ <Liste von Punkten> ]**

Berechnet die Regressionskurve in Form einer Potenzfunktion  $a \cdot x^b$ .

Alle Punkte müssen im ersten Quadranten des Koordinatensystems liegen.

### **TrendSin[ <Liste von Punkten> ]**

Berechnet die Regressionskurve in Form einer Sinusfunktion  $a + b \sin(c x + d)$ .

Die Liste sollte aus mindestens 4 Punkten bestehen und zumindest zwei Extrempunkte umfassen. Die ersten beiden lokalen Extrempunkte sollten nicht allzu verschieden von den absoluten Extrempunkten der Kurve sein.

### **Trend[ <Liste von Punkten>, <Liste von Funktionen> ]**

Berechnet die Regressionskurve der angegebenen Punkte als eine Linearkombination der Funktionen aus der Liste.

## Befehle für das CAS-System

### Algebra

**Faktorisiere**[ <Funktion> ]

Faktorisiert die Funktion.

**Beispiel:** Faktorisiere[ $x^2 + x - 6$ ] liefert  $(x + 3)(x - 2)$ .

**Multipliziere**[ <Ausdruck> ]

Multipliziert den Ausdruck.

**Beispiel:** Multipliziere[ $(2x - 1)^2 + 2x + 3$ ] erzeugt den Ausdruck  $4x^2 - 2x + 4$ .

**Vereinfache** [ <Funktion> ]

Vereinfacht, falls möglich, die Terme der angegebenen Funktion. Es dürfen undefinierte Variablen in den Termen enthalten sein.

**Beispiel:** Vereinfache[ $3*x + 4*x + a*x$ ] liefert  $7x + ax$ .

**Ersetze**[ <Ausdruck>, <von>, <durch> ]

Ersetzt *von* im *Ausdruck* mit *durch*.

**Beispiel:** Ersetze[ $(3m - 3)^2 - (m + 3)^2$ ,  $m$ ,  $a$ ] liefert  $8a^2 - 24a$ .

**Ersetze**[ <Ausdruck>, <Substitutionsliste> ]

Ersetzt in *Ausdruck* jede Variable der Substitutionsliste mit der dafür gewählten Variablen oder Zahl.

**Beispiel:** Ersetze[ $2x+3y-z$ , { $x=a, y=2, z=b$ }] liefert  $2a - b + 6$ .

**Lösche**[ <Objekt> ]

Löscht das Objekt und alle davon abhängigen Objekte.

**Beispiel:** Lösche[ $a$ ] löscht  $a$ .

### Matrix

Eine Matrix ist eine rechteckige Anordnung von Zahlen (oder anderen Objekten). Meistens werden sie (wie auch bei der Ausgabe in GeoGebra) mit runden Klammern geschrieben. Eine wichtige Matrix ist die erweiterte Koeffizientenmatrix eines linearen Gleichungssystems.

Zum Gleichungssystem:

$$\begin{aligned} 3x + 6y - 2z &= -15 \\ 3x + 2y + z &= 2 \\ 2x + 5y - 5z &= -23 \end{aligned}$$

gehört die Matrix:

$$\left( \begin{array}{ccc|c} 3 & 6 & -2 & -15 \\ 3 & 2 & 1 & 2 \\ 2 & 5 & -5 & -23 \end{array} \right)$$

Eine Matrix kannst du im CAS mit geschweiften Klammern eingeben:

$$M := \{ \{3, 6, -2, -15\}, \{3, 2, 1, 2\}, \{2, 5, -5, -23\} \}$$

Wenn du alles richtig gemacht hast, gibt GeoGebra dir eine "schöne" Matrix aus. Der Befehl **Treppennormalform** wirkt wie der Befehl `rref` bei manchen CAS-Taschenrechnern.

**Treppennormalform**[M]

$$\left( \begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

Die drei Zeilen geben die Lösungen an:  $x = 1, y = -2, z = 3$ .

## Gleichungen

---

### **LinkeSeite[ <Gleichung> ]**

Gibt die linke Seite der Gleichung an.

**Beispiel:** `LinkeSeite[x + 2 = 3 x + 1]` ergibt  $x + 2$ .

### **LinkeSeite[ <Liste von Gleichungen> ]**

Gibt eine Liste der linken Seiten der Gleichungen an.

**Beispiel:** `LinkeSeite[{a^2 + b^2 = c^2, x + 2 = 3 x + 1}]` ergibt  $\{a^2 + b^2, x+2\}$ .

### **RechteSeite[ <Gleichung> ]**

Gibt die rechte Seite der Gleichung an.

**Beispiel:** `RechteSeite[x + 3 = 3 x + 1]` ergibt  $3 x + 1$ .

### **RechteSeite[ <Liste von Gleichungen> ]**

Gibt eine Liste der rechten Seiten der Gleichungen an.

**Beispiel:** `RechteSeite[{a^2 + b^2 = c^2, x + 2 = 3 x + 1}]` ergibt  $\{c^2, 3x + 1\}$ .

## Lösen von Gleichungen

---

### **Löse[<Gleichung in x>]**

Löst die angegebene Gleichung für die Variable  $x$  und erzeugt eine Liste mit allen Lösungen.

**Beispiel:** `Löse[x^2 = 4x]` berechnet  $\{x = 4, x = 0\}$ .

### **Löse[<Gleichung>, <Variable >]**

Löst die angegebene Gleichung für die angegebene, unbekannte Variable und erzeugt eine Liste mit allen Lösungen.

### **Löse[<Liste von Gleichungen>, <Liste von Variablen>]**

Löst das gegebene Gleichungssystem für die angegebenen Variablen und erzeugt eine Liste mit allen Lösungen.

### **Lösungen[<Gleichung>]**

Löst die angegebene Gleichung für die Variable  $x$  und erzeugt eine Liste mit allen Lösungen.

**Beispiel:** `Lösungen[x^2 = 4x]` liefert  $\{0, 4\}$ .

**Anmerkungen:** Der Unterschied zum Befehl `Löse` besteht darin, dass bei `Löse` die Variablen in der Ausgabe erscheinen. Es sind die gleichen Syntax-Formen wie bei `Löse` möglich.

## NLöse und NLösungen

Die beiden Befehle unterscheiden sich (wie `Löse` und `Lösungen`) nur in der Art der Ausgabe. Es gibt auch hier bei beiden Befehlen die gleichen Syntax-Formen. Für Funktionen, die keine Polynome sind, sollte immer ein Startwert angegeben werden.

### **NLöse[ <Gleichung>, <Variable = Startwert> ]**

Berechnet numerisch die Lösungen der angegebenen Gleichung für die unbekannte Variable mit angegebenen Startwert.

**Beispiel:** `NLöse[cos(x) = x, x = 0]` berechnet  $\{0.74\}$

`NLöse[a^4 + 34a^3 = 34, a = 3]` berechnet  $\{a = 0.99\}$ .

Andere Formen:

`NLöse[ <Gleichung> ]`, `NLöse[ <Gleichung>, <Variable> ]`,

`NLöse[<Liste von Gleichungen>, <Liste von Variablen>]`

### **NLösungen[ <Gleichung>, <Variable = Startwert> ]**

Berechnet numerisch die Lösungen der angegebenen Gleichung für die unbekannte Variable mit angegebenen Startwert.

Andere Formen:

`NLösungen[ <Gleichung> ]`, `NLösungen[ <Gleichung>, <Variable> ]`,

`NLösungen[<Liste von Gleichungen>, <Liste von Variablen>]`