

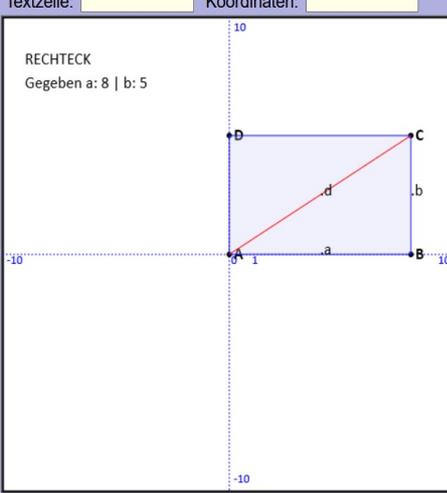
# GEOMATH.HTML

Ein universeller Mathematik-Editor © Herbert Paukert  
(Eine kurze Bedienungsanleitung)

## Demo-Script [01] OHNE Ergebnisausgabe

Textzeile:  Koordinaten:

RECHTECK  
Gegeben a: 8 | b: 5



Gesucht sind: Umfang u Fläche f Diagonale d

Alternative Funktionseingabe (stop mit Leer-Eingabe)  
bei den Scripts [44] bis [54], z.B.  $f(x) = 0.7 \cdot x^2 \cdot \exp(-0.5 \cdot x)$   
Nur den Funktionsterm in das Eingabefeld schreiben:

geomath.html © Herbert Paukert, Version 35.0

```
// Auswahl von 85 verschiedenen Demo-Scripts.
// Die Maus auf [begin] platzieren und dann [Run]
// oder fortlaufend [Parse] oder [F2] anklicken.

// Scripts mit Zufallsgenerator-Eingaben UND
// mit manuellen Eingaben (welche dann den
// Zufall überschreiben) sind mit (&) markiert.
// Scripts mit wahlweisen Ergebnis-Ausgaben
// sind mit (*) markiert. Sie sind ideal zum Üben.

// [01] (&*) Rechteck
// Gegeben ist ein Rechteck mit den zwei Seiten.
// Gesucht sind Umfang und Fläche und Diagonale.
// Zusätzlich wird das Rechteck auch gezeichnet.
// Die zwei Seiten a und b werden eingegeben.
begin
clr()
txt(RECHTECK;-9;8)
inf(Weiter mit Eingabe der Seiten a und b;)
a = round(random(6) + 3)
b = round(random(6) + 3)
apo(_ab;)
A(0/0)
B(a/0)
C(a/b)
D(0/b)
lf3()
```

Taschenrechner — Mozilla Firefox

file:///C:/Users/Susanne Holboeck/De

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	l/z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

**Run** **&** **Goto** **Koord** **ClrGraf** **ClrRes** **Calc** **Help**

**Text save** Browse... No file selected.

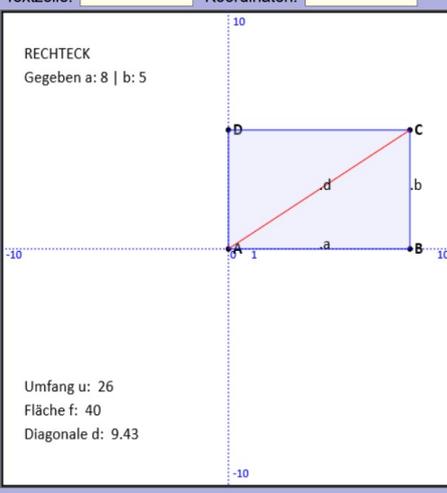
**Graph save** Browse... No file selected.

**Parse** **Prog** **+** **-** **Tprm** **Gprm** **Train** **Del** **Res**

## Demo-Script [01] MIT Ergebnisausgabe

Textzeile:  Koordinaten:

RECHTECK  
Gegeben a: 8 | b: 5



Umfang u: 26  
Fläche f: 40  
Diagonale d: 9.43

Alternative Funktionseingabe (stop mit Leer-Eingabe)  
bei den Scripts [44] bis [54], z.B.  $f(x) = 0.7 \cdot x^2 \cdot \exp(-0.5 \cdot x)$   
Nur den Funktionsterm in das Eingabefeld schreiben:

geomath.html © Herbert Paukert, Version 35.0

```
// Auswahl von 85 verschiedenen Demo-Scripts.
// Die Maus auf [begin] platzieren und dann [Run]
// oder fortlaufend [Parse] oder [F2] anklicken.

// Scripts mit Zufallsgenerator-Eingaben UND
// mit manuellen Eingaben (welche dann den
// Zufall überschreiben) sind mit (&) markiert.
// Scripts mit wahlweisen Ergebnis-Ausgaben
// sind mit (*) markiert. Sie sind ideal zum Üben.

// [01] (&*) Rechteck
// Gegeben ist ein Rechteck mit den zwei Seiten.
// Gesucht sind Umfang und Fläche und Diagonale.
// Zusätzlich wird das Rechteck auch gezeichnet.
// Die zwei Seiten a und b werden eingegeben.
begin
clr()
txt(RECHTECK;-9;8)
inf(Weiter mit Eingabe der Seiten a und b;)
a = round(random(6) + 3)
b = round(random(6) + 3)
apo(_ab;)
A(0/0)
B(a/0)
C(a/b)
D(0/b)
lf3()
```

Taschenrechner — Mozilla Firefox

file:///C:/Users/Susanne Holboeck/De

9.43			
1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	l/z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

**Run** **&** **Goto** **Koord** **ClrGraf** **ClrRes** **Calc** **Help**

**Text save** Browse... No file selected.

**Graph save** Browse... No file selected.

**Parse** **Prog** **+** **-** **Tprm** **Gprm** **Train** **Del** **Res**

Zeichenfläche  
mit 2 Ausgabezeilen  
und 1 Eingabezeile

Befehlseditor  
mit Schalterfeld

Taschenrechner  
(verschiebbar)

---



---

<b>Inhaltsverzeichnis:</b>	Befehle und Liste aller Demo-Scripts	... Seite 2
	<b>Direktes Arbeiten im Mathematik-Editor</b>	... Seite 5 und 10
	Ausdrucke von einigen Scriptbeispielen	... Seite 11
	Sechs Übungs-Scripts (Befehle und Grafiken)	... Seite 17
	Screenshots vom Taschenrechner	... Seite 23
	Screenshots von besonderen Funktionen	... Seite 26

---



---

Der hier vorliegende **Mathematik-Editor** erlaubt die manuelle Eingabe von 26 **kleingeschriebenen Zahlenvariablen** (**a,b,c,...,z**) und von mathematischen Formeln in den Zeilen. Wird der Cursor auf diese platziert, dann werden die Textzeilen mit **[parse]** oder **[F2]** ausgewertet, und die entsprechenden Rechen-ergebnisse werden in den Zeilen rechts vom Gleichheitszeichen (=) angezeigt. Fehler-Arten: [1] "Infinity" (unendlich), [2] "NaN" = Not a Number (undefiniert). **Hinweis:** Die Formelzeilen müssen immer zwischen "begin" und "end" stehen. Die Kommentarzeilen ("/") dürfen nicht zwischen "begin" und "end" stehen, und sie dürfen auch die Spezialzeichen ",", " oder ";" oder "=" nicht enthalten!

---



---

***Folgende Operatoren stehen zur Verfügung***

---



---

(, ), +, -, \*, /, ^, **sqrt**, % (Divisionsrest), **fac** (Fakultät), **round,rd2,rd4** (Runden), **floor abs, exp, log** (e-Log.), **lg** (10-Log.), **deg, rad, sin, cos, tan, asin, acos, atan**, **random(x) = Zufallszahlen in [0,x)**, **PI** (Ludolfsche Zahl) und **E** (Eulersche Zahl). Beispiele: (a % b), sin(rad(x)), deg(asin(x)), sqrt(x), x^(1/2), exp(x), E^x, log(x), lg(x), 10^(lg(x)) liefert x, random(1) liefert Zufallszahlen aus dem halboffenen Intervall [0,1). Grundsätzlich werden hier immer alle Winkelfunktionen im Radianten-Maß berechnet. "deg(x)" wandelt Radianten in Grade um. "rad(x)" wandelt Grade in Radianten um.

---



---

***Folgende Script-Befehle stehen zur Verfügung***

---



---

**Arithmetik:** Variable = Zahl, Variable = Formel

**Geometrie:** P(x/y) ..... 26 **großgeschriebene Punktvariable** (**A,B,C,...,Z**)

**Hinweis:** Jeder arithmetische oder geometrische Befehl darf immer nur in einer Textzeile stehen und alle Befehle eines Scripts müssen immer zwischen "begin" und "end" stehen. Enthält ein Script-Befehl zusätzliche Parameter, dann muss genau dahinter ein ";" stehen. Nur bei mehreren Parametern darf hinter dem letzten Parameter kein Strichpunkt stehen. Die meisten Befehle belegen bestimmte Zahlenvariable mit bestimmten Werten, die von nachfolgenden Befehlen verwendet werden können, z.B. speichert **lng(A;B)** die Länge auf z. Der Spezialbefehl "**apo( \_abc... ; )**" wandelt die Variablenwerte a,b,c... in ihre Beträge um. Der Spezialbefehl "**ank( \_abc... ; )**" löscht in Variablenwerten a,b,c... die Nachkommastellen.

**clr()** Grafik löschen, **clr(g;)** Grafik löschen mit **g** als Halbbreite des Koordinatensystems, Der Standardwert für **g** ist 10. **clr(0;)** Grafik löschen ohne Zeichnung des Koordinatensystems. **gru(a;b;... )** bestimmt die Halbbreite des Koordinatensystems als Maximum von a, b, c, ... **ldx()** Liniendicke mit x = 1, 2 und 3, **lfx()** Linienfarbe mit x = 1 (schwarz), 2 (rot) und 3 (blau), **ren(A;B)** Punkt A zu Punkt B umbenennen, **ren(A;A)** Punktkoordinaten von A anzeigen, **lin(A;B)** Strecke, **li3(A;B;C)** Dreieck, **li4(A;B;C;D)** Viereck, **win(A;S;B)** Winkel mit Scheitel S, **lng(A;B)** Streckenlänge, **umf(A;B;C)** Umfang von Dreiecken, **fla(A;B;C)** Fläche von Dreiecken, **um4(A;B;C;D)** Umfang von konvexen Vierecken, **fl4(A;B;C;D)** Fläche von konvexen Vierecken, Die benachbarten Eckpunkte der Vierecke müssen immer fortlaufend beschriftet werden. **lib(.s;A;B)** Linienbeschriftung von AB mit dem Text s. Vor s muss immer ein Punkt stehen!

**txt(s;x;y)** Textausgabe des Textes s an den Koordinaten x und y der Grafikfläche (Canvas).  
 Kommen innerhalb des Textes auch Zahlenvariable vor und sollen sie durch ihre Zahlenwerte ersetzt werden, dann muss ihnen ein Underline "\_" vorangestellt werden, z.B. "Länge a: \_a".  
 Der Text darf - so wie die Kommentarzeilen - die Zeichen ",", ";" und "=" nicht enthalten.  
 Anstelle von "," wird "|" verwendet, anstelle von "=" werden die Zeichen ":" oder "->" verwendet.  
**inf(s;)** Ausgabe des Textes s in einer Alert-Box, wenn der Text "Weiter", "Gesucht" oder "." enthält.  
 Es gelten die gleichen Regeln wie für den Befehl **txt(s;x;y)**. Steht im so angezeigten Text das Zeichen #, dann erfolgt dort ein einfacher Zeilensprung. Die beiden Befehle sind besondere Script-Befehle mit denen Texte, Variable und deren Zahlenwerte angezeigt werden können.  
 Mit dem Befehl **out(s;)** werden ebenfalls Texte und Variable in einer Kommentarzeile angezeigt.

**Hinweis:** Wenn der Text des "inf"-Befehls das Wort "Gesucht" enthält, kann der Anwender entscheiden, ob die gesuchten Ergebnisse angezeigt werden oder nicht. Die auf Seiten 5,6,7 mit einem (\*) markierten Grundaufgaben enthalten diese Möglichkeit der Aufgabenlösung.  
 Bei Script-Wiederholung mit Ergebnis-Anzeige sind die gleichen Werte einzugeben wie vorher.  
 In Aufgaben, die mit einem (&) markiert sind, erfolgen die Zahleneingaben mit einem Zufalls-generator oder manuell. Bei manueller Eingabe werden die Zufallswerte überschrieben.

**krs(M;r)** Kreis, **arc(M;r;a;e)** Kreisbogen, **kru(A;B;C)** Umkreis, **kri(A;B;C)** Inkreis von Dreiecken,  
**sgk(A;B;M;r)** Schnittpunkt von Gerade und Kreis, **skk(M1;r1;M2;r2)** Schnittpunkt zweier Kreise.  
**hap(A;B)** Streckenmittelpunkt, **swp(A;B;C)** Dreiecks-Schwerpunkt, **ssm(A;B)** Seitensymmetrale,  
**wsm(A;S;B)** Winkelsymmetrale, **ger(A;B)** Gerade durch A und B, **lot(A;B;C)** Lot von A auf BC,  
**gkd(k;d)** Gerade g mit  $y = k*x + d$ , liefert zusätzlich noch zwei Punkte X und Y auf der Geraden g.  
**gkp(k;A)** Gerade g mit Anstieg k durch Punkt A, liefert zusätzlich noch einen Punkt Y auf g.  
 Die zusätzlichen Punkte werden u.a. für den Schnittpunkt von Gerade und Kreis verwendet.  
**sgg(A;B;C;D)** Schnittpunkt zweier Geraden g1 und g2, mit A und B auf g1, mit C und D auf g2,  
**syy(k1,d1,k2,d2)** Schnittpunkt zweier Geraden, mit g1 ( $y = k1*x + d1$ ) und g2 ( $y = k2*x + d2$ ).

**fun(f(x))** Zeichnet die Funktion f(x), **fun(stp)** Unterbricht das Script, **fun()** Beendet das Script,  
**sfg(f(x);g(x);l;r)** Schnitt von f(x) und g(x) im Intervall [l;r], **tg2(f(x);A;l;r)** Tangente von A an f(x),  
**dif(f(x);a;n)** n-te Ableitung von f(x) an der Stelle a mit  $n = 0,1,2$ , **tng(f(x);a)** Kurventangente in a,  
**nul(f(x);a;b;n)** Nullstellen der n-ten Ableitung (0,1,2) von der Funktion f(x) zwischen a und b,  
**int(f(x);a;b)** Integral, **fsp(f(x);a;b)** Flächenschwerpunkt, **fbo(f(x);a;b)** Bogenlänge zwischen a und b,  
**vol(f(x);a;b)** Volumen von Rotationskörpern, **ofl(f(x);a;b)** Oberfläche (Mantel) von Rotationskörpern,  
**uwvt()** Ungebremstes exponentielles Wachstum, **gwt()** Gebremstes logistisches Wachstum,  
**ff1(a;b;f(x))** Gleitende Tangenten, **ff2(f(t);g(t))** Parameter-Funktionen, **ff3(a;b;c)** Hypozykloide.  
**evo(a;n;f(x))** Evolute von f(x) mit Krümmungskreis im Punkt P(a/f(a)) und mit  $n = 1$  oder  $n = 2$ .

**Hinweis:** Bei den neun Scriptbefehlen **fun**, **dif**, **tng**, **nul**, **int**, **fsp**, **fbo**, **vol**, **ofl** sind in dem Funktionsterm f(x) auch Variable möglich, z.B.  $\text{fun}(k*x+d)$  oder  $\text{fun}(\exp(a*x))$ . Zusätzlich kann ein Minuszeichen (-) vor den Funktionsterm f(x) und auch vor die Punktkoordinaten gesetzt werden. Diese Funktionen dürfen keine zusammengesetzte Funktionen sein, wie z.B.  $a*\sin(b*x)+c*\sin(d*x)$ .  
 Der Befehl **fuo(...f(x)...)** ermöglicht die **direkte, manuelle Eingabe** von beliebigen Funktionen, z.B.  $f(x) = 5*\sin(3*x)+2*\sin(4*x)$ . Er darf jedoch keine zusätzlichen Variablennamen enthalten.

Bei den zwölf Befehlen **fuo**, **fun**, **sfg**, **tg2**, **dif**, **tng**, **nul**, **int**, **fsp**, **fbo**, **vol**, **ofl**, die mathematische Funktionen enthalten, kann der Funktionsterm f(x) alternativ eingegeben werden. Dazu muss der Funktionsterm explizit in das Eingabefeld links unten am Bildschirm geschrieben werden. In diesem alternativen Funktionsterm dürfen keine zusätzlichen Variablennamen vorkommen. Ist das Eingabefeld leer, dann wird die im Script direkt eingegebene Funktion verwendet.  
 Der Spezialbefehl **nof()** schaltet den alternativen Funktionsterm aus (bzw. löscht ihn).

**Hinweis:** Wird bei "fuo" ein "w" eingegeben, dann wird der letzte Funktionsterm genommen.

**Arithmetische Grundbefehle:** Rechnen mit Natürlichen Zahlen **naz(a;b)**,  
 Rechnen mit Ganzen Zahlen **gaz(a;b)**, Bruchzahlen **bru(a;b;c;d)**, Dezimalzahlen **dez(a;b)**,  
**tel(a;)** Alle Teiler von a, **pfz(a;)** Primfaktorenzerlegung, **ggt(a;b)** GGT und KGV von a und b,  
**sr1(g;v;t)** Einfache Schlussrechnung ( $x=t*v/g$ ); **sr2(g;v;w)** Rechnung mit Umkehrschluss ( $x=w*g/v$ ),  
**pr1(g;p)** Prozentrechnung ( $a=?$ ), **pr2(a;p)** Prozentrechnung ( $g=?$ ), **pr3(g;a)** Prozentrechnung ( $p=?$ ).  
**lg2(a;b;...;f)** Lineares Gleichungssystem in 2 Variablen mit 6 Koeffizienten a,b,c, d,e,f.  
**lg3(a;b;...;l)** Lineares Gleichungssystem in 3 Variablen mit 12 Koeffizienten a,b,c,d,e,f,g,h,i,j,k,l.  
**gl2(a;b;c)** Quadratische Gleichungen ( $a*x^2 + b*x + c = 0$ ).

**st1()** Eindimensionale Statistik, **st2()** Zweidimensionale Statistik mit maximal 100 Daten.  
**nvt(a;b)** Standardisierte Normalverteilung (Mittelwert 0, Streuung 1) zwischen a und b.  
**vlk(j;A;k;B)** Linearkombination von Vektoren  $C = j*A + k*B$ , **vsp(A;B)** Skalarprodukt.  
 Drei klassische Grundaufgaben der Landvermessung: **hvm(h;v;w)** Höhenvermessung,  
**vve(a;t;u;v;w)** Vorwärtseinschneiden und **rwe(a;b;u;v;w)** Rückwärtseinschneiden.  
**Dreidimensionale Körper:** **qua(a;b;c)** Quader, **pri(a;h)** Prisma, **pyr(a;b;h)** Pyramide,  
**okt(a;)** Oktaeder, **tet(a;)** Tetraeder, **zyl(r;h)** Zylinder, **keg(r;h)** Kegel, **kug(r;)** Kugel,  
**tor(a;b)** Torus, **wuk(a;)** Würfel mit Kugel, **rdd(a;)** Rhombendodekaeder mit Kante a.

Alle Script-Befehle zwischen "**begin**" und "**end**" werden **automatisch** ausgeführt,  
 wenn der Cursor auf "**begin**" platziert und dann der Schalter [**Run**] angeklickt wird.  
 Einige Befehle liefern Ergebnisse, die als Text und als Grafik angezeigt werden.  
 Für die **Punktkoordinaten** können auch **Variablenbezeichner** eingesetzt werden.

Bei Befehlen, die Punkte P(x/y) anzeigen, können deren Koordinaten als Variable  
 x und y weiter verwendet werden. Einige geometrische Befehle belegen **reservierte**  
**Zahlenvariable** mit Werten, die oft in nachfolgenden Befehlen verwendet werden:  
 Länge (z), Winkel (w), Umfang (u), Fläche (f), Volumen (v), Oberfläche (o), Radius (r),  
 Argumente (x) und Funktionswerte (y), Steigung (k) und Abschnitt (d) von Geraden.  
 Die String-Variable s ist bei Funktionen oftmals für die Funktionsgleichung reserviert.  
 Z.B. Wendepunkt und Wendetangente: Zuerst liefert  $nul(0.5*x^3-3*x^2+4*x;a;b;2)$   
 die Wendestelle x, dann  $tng(0.5*x^3-3*x^2+4*x;x)$  mit Wendepunkt-Argument x.

===== *Folgende Schalter stehen zur Verfügung* =====

#### [Run] ... Automatik-Modus ein

[ & ] ... Parser-Ergebnisse zusätzlich im Editor anzeigen (oder nicht)  
 [Goto] ... Sprung zu einem Script-Beginn [mit Nummer]  
 [Koord] ... Größe des Koordinatensystems ändern  
 [ClrGraf] ... Löschung der Grafik  
 [ClrRes] ... Löschung der Ergebniswerte  
 [Calc] ... Öffnen des universellen Taschenrechners "calc.html"  
 [Help] ... Bedienungsanleitung ein- und ausblenden

#### [Parse] oder [F2] ... Einzelschritt-Modus ein

#### [Prog] ... Programmier-Modus ein/aus

[ + ] ... Schrift vergrößern, [ - ] ... Schrift verkleinern  
 [Tprn] ... Markierten Text drucken, [Gprn] ... Grafikbild drucken  
 [Text save] ... Download des Scripts in eine Textdatei (script\*.txt)  
 [Durchsuchen] ... Suchen und Laden einer Script-Datei  
 [Graph save] ... Download der Grafik in eine Grafikdatei (pict\*.jpg)  
 [Durchsuchen] ... Suchen und Laden einer Grafikdatei  
 [Train] ... Training mit den Übungs-Scripts, [F4] ... Zeitverzögerung  
 [Del] ... Löschung von allen Scripts im Editor  
 [Res] ... Die originalen Demo-Scripts wieder in den Editor laden

- » Im **Automatik-Modus [Run]** werden alle Befehlszeilen eines Scripts im Editor, welche zwischen "begin" und "end" stehen - mit oder ohne Zeitverzögerung - automatisch durchlaufen. Der Editor ist für Schreibzugriffe gesperrt. Mit dem Befehl "Variable = Wert" öffnet sich ein Eingabefenster, wo für die mit "Wert" vorbelegte Variable ein beliebiger, neuer Zahlenwert eingegeben werden kann. Rekursive Wertzuweisungen wie "Variable = Variable + 1" sind nicht möglich. Die Ausgabe erfolgt in der Grafikfläche (Canvas) durch spezifische Befehle.
- » Im **Einzelschritt-Modus [Parse]** muss der Cursor auf die Befehlszeile "begin" platziert und dann fortlaufend **[Parse]** angeklickt oder **[F2]** eingetastet werden. Dabei werden alle Parser-Ergebnisse synchron zum entsprechenden Befehl im Editor angezeigt. Auch hier ist der Mathematik-Editor für Schreibzugriffe gesperrt.
- » Im **Programmier-Modus [Prog]** ist der Editor offen und in die einzelnen Zeilen können Wertzuweisungen zu Variablen und zulässige Befehle geschrieben werden. Zur Ausführung der Befehle muss entweder in den Einzelschritt-Modus [Parse] oder in den Automatik-Modus [Run] gewechselt werden.

**Hinweis:** Alle Texte und Scripts im Editor können mit **[Del]** gelöscht werden. Mit Hilfe des Schalters **[Res]** können die originalen Demo-Scripts wieder in den Editor geladen werden. Dabei ist jedoch zu beachten, dass Scriptänderungen oder neue Scripts des Anwenders gelöscht werden. Eigenständige Scripts sollten mit Schalter **[Text save]** in den Download-Ordner des Anwenders gespeichert werden. Von dort können sie dann jederzeit wieder in den Editor geladen werden.

In eigenständigen Scripts sollten die 7 Punktnamen H, P, Q, S, T, X, Y möglichst nicht verwendet werden, weil sie für mehrere geomtrische Befehle reserviert sind. Die restlichen Punkte A, B, C, . . . . ., Z sind aber immer frei verfügbar. Auf [Seite 2] im letzten Absatz sind alle 12 **reservierten Zahlenvariablen** angeführt.

- (!) **Wichtige Hinweise** für Anwender, welche eigene Scripts erzeugen wollen:*
- (!) Mit dem Anklicken von **[Train]** werden **Übungs-Scripts** in den Editor kopiert.*
  - (!) Mit **[Prog]** wird in den Programmier-Modus des Mathematik-Editors gewechselt.*
  - (!) Es können die Scripts abgeändert und Variable und Befehle ausprobiert werden.*
  - (!) Mit **[Parse]** werden dann die Befehle in den einzelnen Editorzeilen ausgeführt.*
  - (!) Mit **[Text save]** wird ein Script in eine Textdatei im Download-Ordner gespeichert.*
  - (!) Mit **[Durchsuchen]** wird ein gespeichertes Script wieder in den Editor geladen.*
  - (!) Will man die Übungs-Scripts beenden, dann muss mit **[Res]** neu gestartet werden.*
  - (!) Die Übungs-Scripts sind in dieser Anleitung auf den Seiten [7,8,9,10] gelistet.*

===== **Folgende 85 Demo-Scripts stehen zur Verfügung** =====  
 (Die Aufgaben mit (\*) sind OHNE oder MIT Ergebnisausgaben durchführbar.)

----- Dreiecke, Vierecke, Gerade, Kreise

- [01] Rechteck (\*)
- [02] Einfache Grundstücke (\*)
- [03] Rechteck und Umkreis (\*)
- [04] Kreis und Kreissektor (\*)
- [05] Kreise im Quadrat (\*)
- [06] Parallelogramm (\*)
- [07] Trapez (\*)
- [08] Deltoid (\*)
- [09] Dreieck (SSS) (\*)
- [10] Dreieck (SWS) (\*)
- [11] Dreieck (SSW) (\*)
- [12] Dreieck (WSW) (\*)

- [13] Inkreis
- [14] Umkreis
- [15] Schwerpunkt
- [16] Lotrechte Gerade
- [17] Höhenschnittpunkt
- [18] Lineare Funktionen
- [19] Schnitt zweier Geraden
- [20] Schnitt von Gerade und Kreis
- [21] Schnitt zweier Kreise

----- Arithmetische Rechenaufgaben

- [22] Rechnen mit natürlichen Zahlen (\*)
- [23] Rechnen mit ganzen Zahlen (\*)
- [24] Rechnen mit Bruchzahlen (\*)
- [25] Rechnen mit Dezimalzahlen (\*)
- [26] Teiler und Primfaktoren (\*)
- [27] GGT und KGV von ganzen Zahlen (\*)
- [28] Schlussrechnen (1) (\*)
- [29] Schlussrechnen (2) (\*)
- [30] Prozentrechnen (1) (\*)
- [31] Prozentrechnen (2) (\*)
- [32] Prozentrechnen (3) (\*)
- [33] Lineares Gleichungssystem (2) (\*)
- [34] Lineares Gleichungssystem (3)
- [35] Quadratische Gleichungen

-----Statistik, Vektoren, Landvermessung

- [36] Eindimensionale Statistik
- [37] Zweidimensionale Statistik
- [38] Normalverteilung
- [39] Linearkombination von Vektoren
- [40] Skalarprodukt von Vektoren
- [41] Höhenvermessung
- [42] Vorwärtseinschneiden
- [43] Rückwärtseinschneiden

----- Funktionen, Differenzial-/Integralrechnung

- [44] Funktionen ohne Zusatz-Variable
- [45] Schnittpunkt von zwei Funktionen
- [46] Tangenten von Punkt an Funktion
- [47] Funktionen mit Zusatz-Variablen
- [48] Funktionswert
- [49] Erster Ableitungswert
- [50] Zweiter Ableitungswert
- [51] Nullstellen
- [52] Extremstellen
- [53] Wendestellen
- [54] Tangenten
- [55] Integral
- [56] Schwerpunkt von Flächen
- [57] Bogenlänge von Kurven
- [58] Volumen von Drehkörpern
- [59] Oberfläche von Drehkörpern

## ----- Kegelschnitte, Schwingungen, Wachstumskurven

- [60] Ellipse (Definition)
- [61] Ellipse (Tangenten)
- [62] Hyperbel (Definition)
- [63] Hyperbel (Tangenten)
- [64] Parabel (Definition)
- [65] Parabel (Tangenten)
- [66] Einfache Schwingungen
- [67] Gedämpfte Schwingungen
- [68] Interferenzen
- [69] Ungebremstes Wachstum
- [70] Gebremstes Wachstum

## ----- Dreidimensionale Körper

- [71] Quader (\*)
- [72] Prisma (\*)
- [73] Pyramide (\*)
- [74] Oktaeder (\*)
- [75] Tetraeder (\*)
- [76] Zylinder (\*)
- [77] Kegel (\*)
- [78] Kugel (\*)
- [79] Würfel mit Kugel (\*)
- [80] Rhombendodekaeder (\*)

## ----- Besondere Funktionen

- [81] Torus
- [82] Gleitende Tangenten
- [83] Evoluten
- [84] Parameter-Funktionen
- [85] Hypozykloid-Funktionen

===== *Folgende 6 Übungs-Scripts stehen zur Verfügung* =====  
 ===== *Erstes Übungs-Script* =====

## // Seiten und Winkel eines Dreiecks ABC

```
begin
clr(10;)
txt(Dreieck ABC; -9; 8) ..... // Der Befehl zeigt bei (-9/8) den angegeben Text („Dreieck ABC“) an
inf(Weiter mit der Eingabe der Eckpunkte ABC;)
a = 0
b = -3.6325
A(a/b)
c = 4
d = 3
B(c/d)
e = -6
f = 0
C(e/f)
li3(A;B;C) ..... // Der Befehl zeichnet das Dreieck ABC.
txt(A( a/ b) | B( c/ d) | C( e/ f); -9; 7) ..... // Der Befehl zeigt bei (-9/7) drei Punkte mit ihren Koordinaten an
inf(Weiter;)
lng(A;B) ..... // Der Befehl ermittelt die Länge der Strecke AB und speichert sie auf z.
c = z
lng(A;C)
b = z
lng(B;C)
a = z
u = rd2(a + b + c)
```

```

win(B;A;C) .... // Der Befehl ermittelt die Größe des Winkels w(A) und speichert sie auf w.
p = w
win(A;B;C)
q = w
win(B;C;A)
r = w
s = round(p + q + r)
fla(A;B;C) ..... // Der Befehl ermittelt die Fläche des Dreiecks ABC und speichert sie auf.
t = rd2(f)
txt(Seiten a: _a | b: _b | c: _c; -9; -7)
txt(Winkel bei A: _p | bei B: _q | bei C: _r; -9; -8)
txt(Umfang: _u | Fläche: _t | Winkelsumme: _s; -9; -9)
end

```

### ===== Zweites Übungs-Script =====

```

// Gerade g durch A lotrecht zu BC
// Gerade h durch A parallel zu BC

```

```

begin
clr()
inf(Weiter mit der Eingabe von drei Punkten;)
a = 2
b = 3
A(a/b)
c = 0
d = -4
B(c/d)
e = 6
f = 0
C(e/f)
ger(B;C) ..... // Der Befehl erzeugt eine Gerade durch B und C, und liefert k und d.
lot(A;B;C) ..... // Der Befehl erzeugt eine lotrechte Gerade durch A auf BC, und liefert
inf(Weiter;) ... // den Lotfußpunkt F, den Abstand AF = z, und k und d.
ger(B;C) ..... // Der Befehl erzeugt eine Gerade durch B und C, und liefert k und d.
gkp(k;A) ..... // Der Befehl übernimmt das k für eine neue Gerade g durch den Punkt A
ren(Y;"" ) ..... // und erzeugt den Punkt Y auf g. ren(Y."") löscht nur die Anzeige von Y.
ger(A;Y) ..... // Der Befehl zeichnet die Gerade g durch die Punkte A und Y noch einmal.
end

```

### ===== Drittes Übungs-Script =====

```

// Schnitt von Gerade  $y = k \cdot x + d$  und
// Kreis mit Mittelpunkt M und Radius r

```

```

begin
clr()
inf(Weiter mit Mittelpunkt M(x/y) und Radius r;)
x = 3
y = 0
M(x/y)
r = 5
krs(M;r) ..... // Der Befehl erzeugt einen Kreis mit Mittelpunkt M und Radius r.
inf(Weiter;)
k = 0.5
d = 2
gkd(k;d) ..... // Der Befehl erzeugt eine Gerade g ( $y = k \cdot x + d$ ) und zwei Punkte X und Y auf g.
ren(X;"" ) ..... // Löscht nur die Anzeige des Punktes X, nicht aber die Punktvariable selbst.
ren(Y;"" ) ..... // Löscht nur die Anzeige des Punktes Y, nicht aber die Punktvariable selbst.
inf(Weiter;)
sgk(X;Y;M;r) ... // Der Befehl schneidet die Gerade durch X und Y mit dem Kreis und liefert die
end ..... // beiden Schnittpunkte S und T.

```

---



---

### Viertes Übungs-Script

---



---

```
// Schnitt von zwei Kurven f(x) und g(x). Die beiden Funktionen werden
// zuerst mit dem Befehl "fuo" direkt eingegeben und dann versucht der
// Befehl "sfg" auf einem Intervall den Schnittpunkt der Kurven zu finden.
// Alternative Funktionseingaben werden hier mittels "nof" verhindert.

begin
clr()
inf(Funktionen f(x) eingeben. ;)
nof()
fuo(0.5*exp(x)) ..... // Funktionsterm eingeben, der automatisch auf Variable s gespeichert wird.
txt(f(x): y -> _s; -9; -8) ..... // Funktionsterm s anzeigen bei (9/-8)
fuo(-1*x^2/5+8)
txt(g(x): y -> _s; -9; -9)
inf(Suchintervall eingeben. ;)
l = 0
r = 5
sfg(f(x);g(x);l;r) ..... // Schnittpunkt suchen im Intervall [l;r]
end
```

---



---

### Fünftes Übungs-Script

---



---

```
// Tangente von Punkt an eine Kurve. Die Funktion f(x) wird zuerst mit
// dem Befehl "fuo" direkt eingegeben und zusätzlich auch ein Punkt A.
// Dann versucht der Befehl "tg2" auf einem Intervall die Tangente von
// Punkt A an die Kurve von f(x) zu finden. Alternative Funktionseingaben
// im Eingabefeld sind möglich. Hinweis: Wird bei "fuo" ein "w" eingegeben,
// dann wird der letzte Funktionsterm genommen (Wiederholungsfunktion).
```

```
begin
clr()
inf(Funktion f(x) eingeben. ;)
fuo(0.1*exp(x)) ..... // Funktionsterm eingeben, der automatisch auf Variable s gespeichert wird.
txt(f(x): y -> _s; -9; -9) ..... // Funktionsterm s anzeigen bei (9/-9)
inf(Weiter mit Punkt A;)
x = 0
y = -3
A(x/y)
inf(Suchintervall eingeben. ;)
l = 0
r = 5
tg2(f(x);A;l;r) ..... // Tangente suchen im Intervall [l;r]
end
```

---



---

### Sechstes Übungs-Script

---



---

```
// Tangenten von Punkt A an den Kreis k(M/r)
// Konstruktion mit dem Lehrsatz von Thales
```

```
begin
clr()
inf(Kreis eingeben. ;)
x = 3
y = 1
M(x/y)
r = 4
s = r
krs(M;r) ..... // Kreis k(M/r) zeichnen
```

```

inf(Punkt eingeben. ;)
a = -6
b = 7
A(a/b) ..... // Punkt A außerhalb des Kreises zeichnen
lin(A;M)
inf(Strecke AM halbieren. ;)
hap(A;M) ..... // Strecke AM halbieren im automatisch erzeugten Punkt H
lng(A;H)
krs(H;z) ..... // Hilfskreis (Thaleskreis) mit Mittelpunkt H und Radius z = HA zeichnen
inf(Hilfskreis zeichnen. ;)
skk(M;s;H;z) ..... // Hilfskreis und gegebenen Kreis in den Punkten S und T schneiden
lin(A;S) ..... // Das Dreieck ASM im Hilfskreis ist rechwinkelig (Thalesatz)
lin(M;S)
lng(M;S)
win(M;S;A)
inf(Hilfskreis mit Kreis schneiden. ;)
ger(A;T) ..... // Tangenten zeichnen
ger(A;S)
end

```

===== **Zusatz-Script zur Algebra im Mathematik-Editor** =====  
===== **erreichbar, so wie die Übungs-Scripts, mittels [Train]** =====

```

// Algebra mit 26 Variablen (a ..... z) und Operatoren:
// ( ) + - * / ^ sqrt % (Rest) fac round rd2 rd4 (Runden)
// floor abs exp log lg deg rad sin cos tan asin acos atan
// random(x) -> Zufallszahlen in [0|x) PI (Ludolf) E (Euler)

// (1) Im Mathematik-Editor Variable und Befehle schreiben.
// Dazu zuerst mit [Prog] in den Programmier-Modus wechseln.
// Hinweis: Nach "begin" immer "out(Algebra;)" schreiben.

// (2) Die einzelnen Mathematik-Zeilen auswerten mittels
// [Parse] oder [Run]. Dazu Starten mit der "begin"-Zeile.
// Optional dazu mit [&] alle Parser-Resultate anzeigen.
// Hinweis: [ClrRes] löscht alle Parser-Resultate.

begin
out(Algebra;)
a = 3
b = 2
c = 7.43
d = 10
e = -a*(a*b - a^b) = 9
e = log(exp(c)) = 7.43
e = rd2(lg(d^a)) = 3
e = rd2(deg(asin(b/4))) = 30
e = rd2(sin(rad(a*d))) = 0.5
e = rd4(random(1)) = 0.5365 // reelle Zufallszahl aus [0;1)
e = round(random(d)) = 6 // ganze Zufallszahl aus [0;10)
e = d^a % a = 1 // Divisionrest von (100/3)
e = fac(d) = 3 628 800 // Faktorielle von 100
e = fac(d)/(fac(b)*fac(d-b)) = 25 // Kombinationen von 100 zur Klasse 2
e = E = 2.718281828459045
e = PI = 3.141592653589793
f = a/0 = Infinity
f = sqrt(-a) = NaN
end

```

---

Es folgen Ausdrücke von verschiedenen Demo-Scripts.

Demo-Script [24] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

Rechnen mit zwei Bruchzahlen  $a/b$  und  $c/d$   
Rechnen mit  $(3/4)$  und  $(5/6)$   
Kürzen soweit wie möglich!

Gesucht: Rechnen mit  $(3/4)$  und  $(5/6)$

Demo-Script [24] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

Rechnen mit zwei Bruchzahlen  $a/b$  und  $c/d$   
Rechnen mit  $(3/4)$  und  $(5/6)$   
Kürzen soweit wie möglich!

-----

Summe =  $19/12 = 1 + 7/12$   
Differenz =  $-1/12 = 0 + -1/12$   
Produkt =  $5/8 = 0 + 5/8$   
Quotient =  $9/10 = 0 + 9/10$

-----

Demo-Script [30] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

PROZENTRECHNEN Teil 1  
 g: Grundmenge | a: Anteil | p: Prozente

Die Grundmenge g ist 120 €  
 Welchen Anteil a erhält man um 35% ?

Gesucht: 120 € ... 100 % ? € ... 35 %

Demo-Script [30] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

-----  
 Prozentrechnung  
 -----  
 Eine Grundmenge g beträgt 120 Euro.  
 Davon beträgt der Prozentsatz p = 35%.  
 Wie groß ist der entsprechende Anteil a ?  
 -----  
 Die Grundmenge g ist immer 100%  
 Mit Hilfe einer einfachen Schlussrechnung  
 erhält man die Basisformel:  $a = (g/100) \cdot p$   
 $a = (120 / 100) \cdot 35 = 42$   
 -----  
 Der gesuchte Anteil a sind 42 Euro !  
 -----

Prozentrechnung (auf 2 Dezimalen gerundet)

Der gesuchte Anteil a sind 42 Euro !

Demo-Script [31] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

PROZENTRECHNEN Teil 2  
g: Grundmenge | a: Anteil | p: Prozente

Der Anteil 42 € beträgt 35%.  
Wieviel ist die Grundmenge g ?

Gesucht: 42 € ... 35 % ? € ... 100 %

Demo-Script [31] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

-----  
Prozentrechnung (1. Umkehrung)  
-----

Der Anteil a beträgt 42 Euro.  
Sein Prozentsatz p ist 35%.  
Wieviel beträgt die Grundmenge g ?  
-----

Die Grundmenge g ist immer 100%  
Es gilt immer die Basisformel:  $a = (g/100) \cdot p$   
Dann folgt durch Umformung:  $g = (a/p) \cdot 100$   
 $g = (42 / 35) \cdot 100 = 120$   
-----

Die Grundmenge g beträgt 120 Euro !  
-----

Prozentrechnung (1. Umkehrung)

Die Grundmenge g beträgt 120 Euro !

Demo-Script [32] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

PROZENTRECHNEN Teil 3  
 g: Grundmenge | a: Anteil | p: Prozente

Die Grundmenge g ist 120 €  
 Der Anteil 42 € ist dann wieviele p% ?

Gesucht: 120 € ... 100 % 42 € ... ? %

Demo-Script [32] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

-----  
 Prozentrechnung (2. Umkehrung)  
 -----

Eine Grundmenge g beträgt 120 Euro.  
 Der Anteil a davon beträgt 42 Euro.  
 Wieviele Prozente p sind das ?

-----

Die Grundmenge g ist immer 100%  
 Es gilt immer die Basisformel:  $a = (g/100) \cdot p$   
 Dann folgt durch Umformung:  $p = (a/g) \cdot 100$   
 $p = (42 / 120) \cdot 100 = 35$

-----

Der gesuchte Prozentsatz p ist 35% !  
 -----

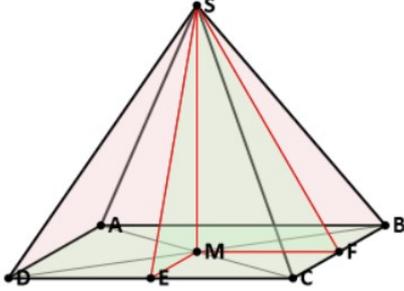
Prozentrechnung (2. Umkehrung)

Der gesuchte Prozentsatz p ist 35% !

Demo-Script [73] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

Pyramide: Kante a = 8, Kante b = 6, Höhe h = 7



Die Grundfläche ist ein Rechteck mit den Seiten a und b.  
 In ihrem Mittelpunkt M ist die Körperhöhe  $h = MS$  errichtet.  
 Die vier Seitenflächen sind gleichschenkelige Dreiecke  
 mit den Seitenhöhen  $ha = ES$  und  $hb = FS$ .  
 Es gilt:  $ha^2 = h^2 + (b/2)^2$  und  $hb^2 = h^2 + (a/2)^2$   
 Zwei gegenüberliegenden Seitenflächen sind kongruent.

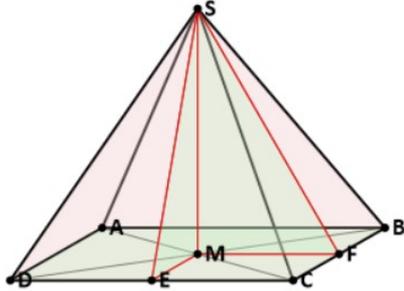
Pyramide: a = 8, b = 6, h = 7

$V = a*b*h/3$ ,  $O = a*b + a*(ha) + b*(hb)$

Demo-Script [73] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

Pyramide: Kante a = 8, Kante b = 6, Höhe h = 7



Die Grundfläche ist ein Rechteck mit den Seiten a und b.  
 In ihrem Mittelpunkt M ist die Körperhöhe  $h = MS$  errichtet.  
 Die vier Seitenflächen sind gleichschenkelige Dreiecke  
 mit den Seitenhöhen  $ha = ES$  und  $hb = FS$ .  
 Es gilt:  $ha^2 = h^2 + (b/2)^2$  und  $hb^2 = h^2 + (a/2)^2$   
 Zwei gegenüberliegenden Seitenflächen sind kongruent.

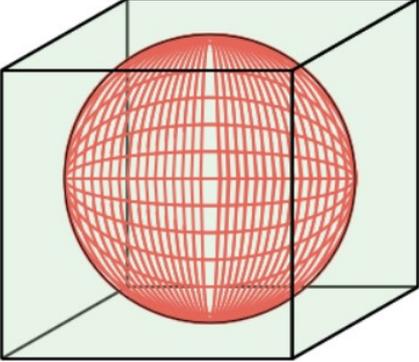
Pyramide: a = 8, b = 6, h = 7

Volumen  $V = 112$ , Oberfläche  $O = 157.3$

Demo-Script [79] OHNE Ergebnisausgabe:

Textzeile:  Koordinaten:

Würfel: Kante  $a = 8$  (Kugeldurchmesser =  $a$ )  
 Aus dem Würfel wird eine Kugel herausgebohrt.  
 Ermittle Volumen und Oberfläche des Restkörpers.



Volumen  $V = \text{Volumen(Würfel)} - \text{Volumen(Kugel)}$   
 Oberfläche  $O = \text{Oberfläche(Würfel)} + \text{Oberfläche(Kugel)}$

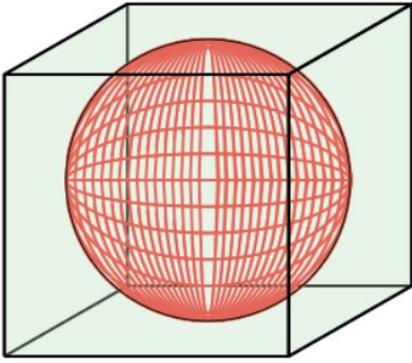
Kante  $a = 8$

$V = a^3 - 4 \cdot (a/2)^3 \cdot \pi / 3, O = 6 \cdot a^2 + 4 \cdot (a/2)^2 \cdot \pi$

Demo-Script [79] MIT Ergebnisausgabe:

Textzeile:  Koordinaten:

Würfel: Kante  $a = 8$  (Kugeldurchmesser =  $a$ )  
 Aus dem Würfel wird eine Kugel herausgebohrt.  
 Ermittle Volumen und Oberfläche des Restkörpers.



Volumen  $V = \text{Volumen(Würfel)} - \text{Volumen(Kugel)}$   
 Oberfläche  $O = \text{Oberfläche(Würfel)} + \text{Oberfläche(Kugel)}$

Kante  $a = 8$

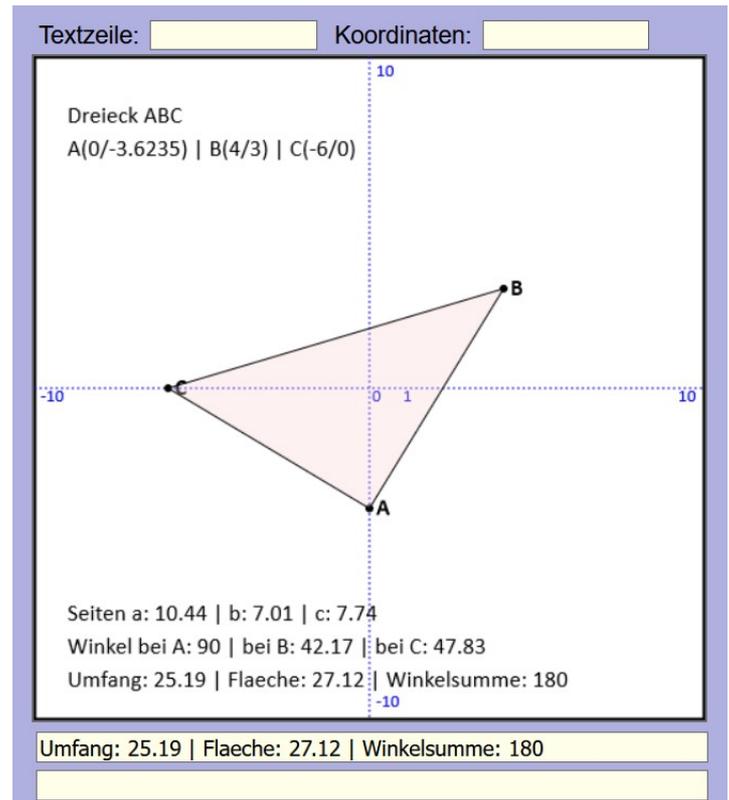
Volumen  $V = 243.92$ , Oberfläche  $O = 585.06$

```

// Erstes Übungs-Script
// Von dem Dreieck ABC werden
// Seiten und Winkel ermittelt

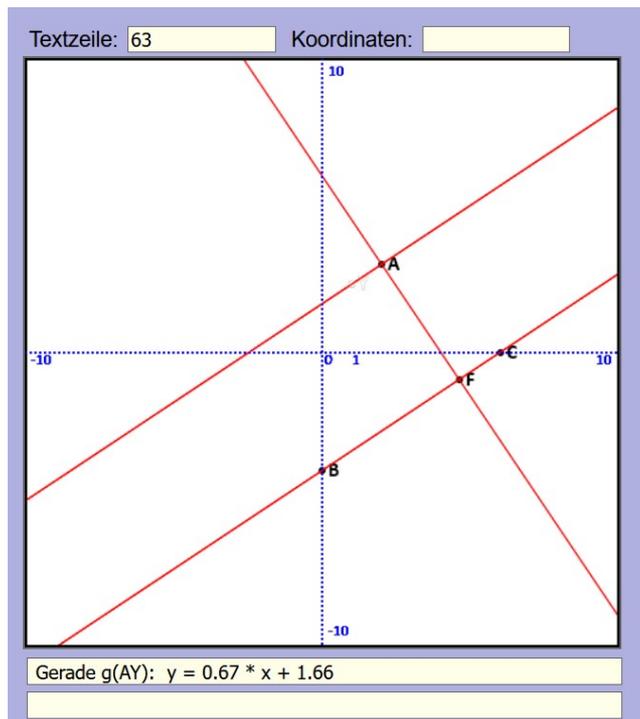
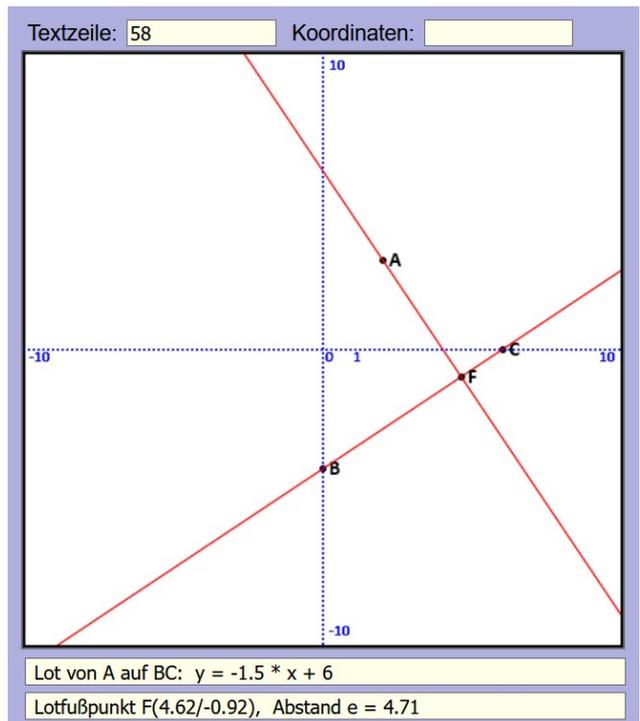
begin
clr(10;)
txt(Dreieck ABC; -9; 8)
inf(Weiter mit der Eingabe der Eckpunkte ABC;)
a = 0
b = -3.6235
A(a/b)
c = 4
d = 3
B(c/d)
e = -6
f = 0
C(e/f)
li3(A;B;C)
txt(A(_a/_b) | B(_c/_d) | C(_e/_f); -9; 7)
inf(Weiter;)
lng(A;B)
c = z
lng(A;C)
b = z
lng(B;C)
a = z
u = rd2(a + b + c)
win(B;A;C)
p = w
win(A;B;C)
q = w
win(B;C;A)
r = w
s = round(p + q + r)
fla(A;B;C)
t = rd2(f)
txt(Seiten a: _a | b: _b | c: _c; -9; -7)
txt(Winkel bei A: _p | bei B: _q | bei C: _r; -9; -8)
txt(Umfang: _u | Flaechе: _t | Winkelsumme: _s; -9; -9)
end

```



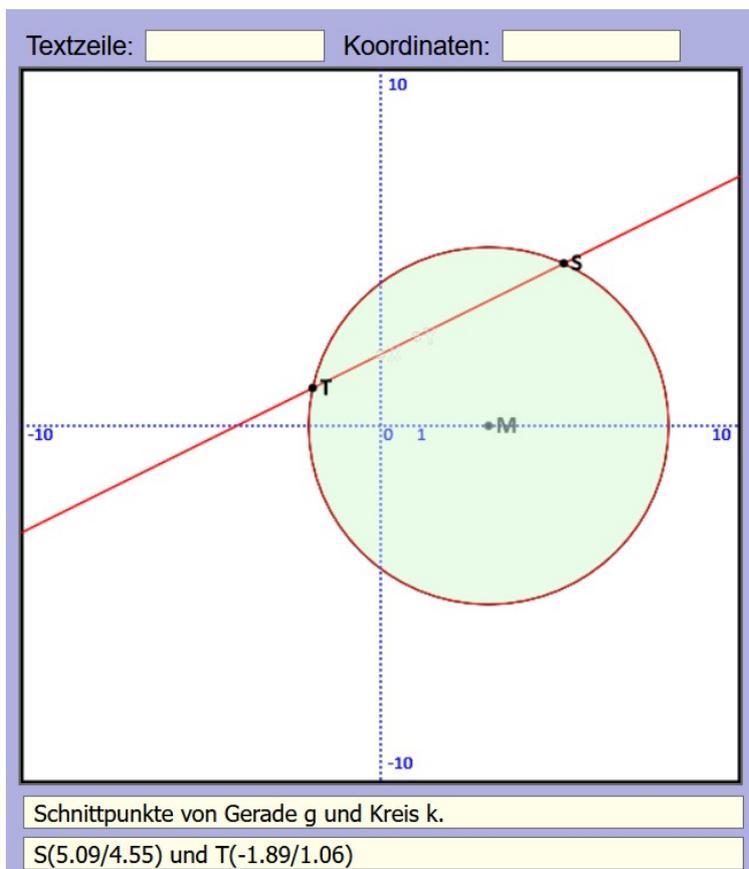
```
// Zweites Übungs-Script
// Gerade g durch A lotrecht zu BC
// Gerade h durch A parallel zu BC
```

```
begin
clr()
inf(Weiter mit der Eingabe von drei Punkten;)
a = 2
b = 3
A(a/b)
c = 0
d = -4
B(c/d)
e = 6
f = 0
C(e/f)
ger(B;C)
lot(A;B;C)
inf(Lotrechte Gerade zu BC. ;)
ger(B;C)
gkp(k;A)
ren(Y;""")
ger(A;Y)
end
```



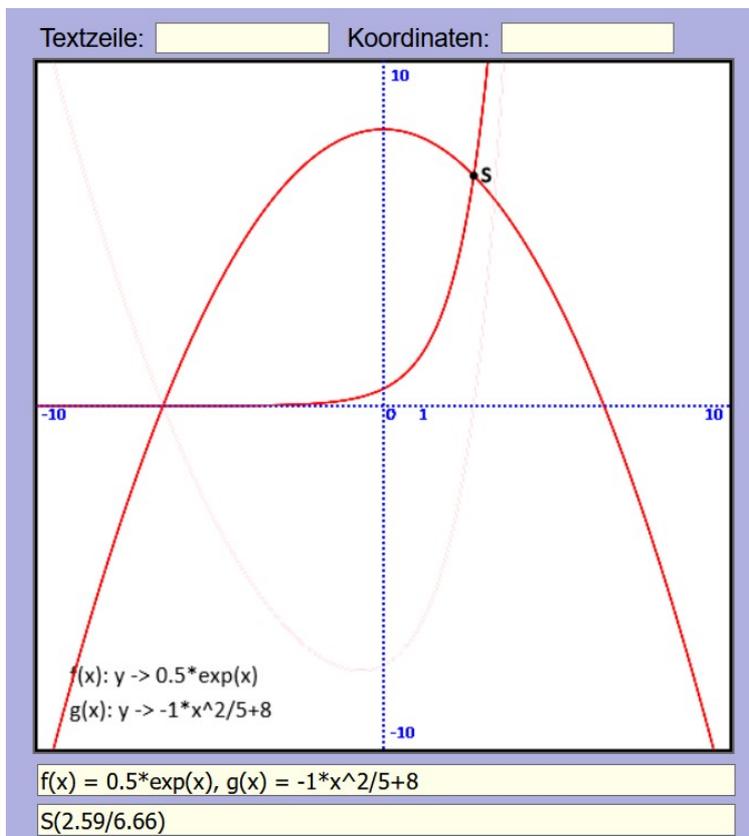
```
// Drittes Übungs-Script
// Schnitt von Gerade  $y = k*x + d$  und
// Kreis mit Mittelpunkt  $M$  und Radius  $r$ 
```

```
begin
clr()
inf(Weiter mit Mittelpunkt  $M(x/y)$  und Radius  $r$ ;)
 $x = 3$ 
 $y = 0$ 
 $M(x/y)$ 
 $r = 5$ 
 $krs(M;r)$ 
inf(Kreis  $k(M/r)$ .;)
 $k = 0.5$ 
 $d = 2$ 
 $gkd(k;d)$ 
inf(Gerade  $y \rightarrow k*x + d$ .;)
ren(X;""")
ren(Y;""")
 $sgk(X;Y;M;r)$ 
end
```



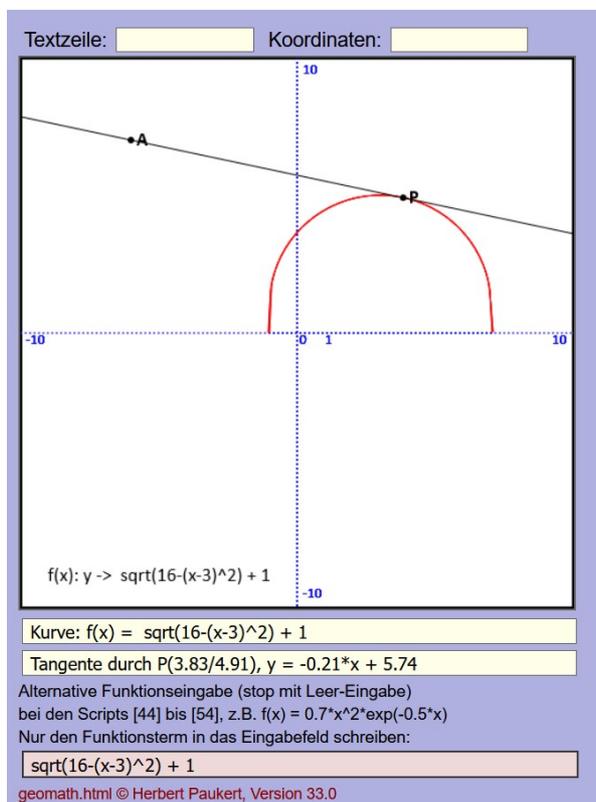
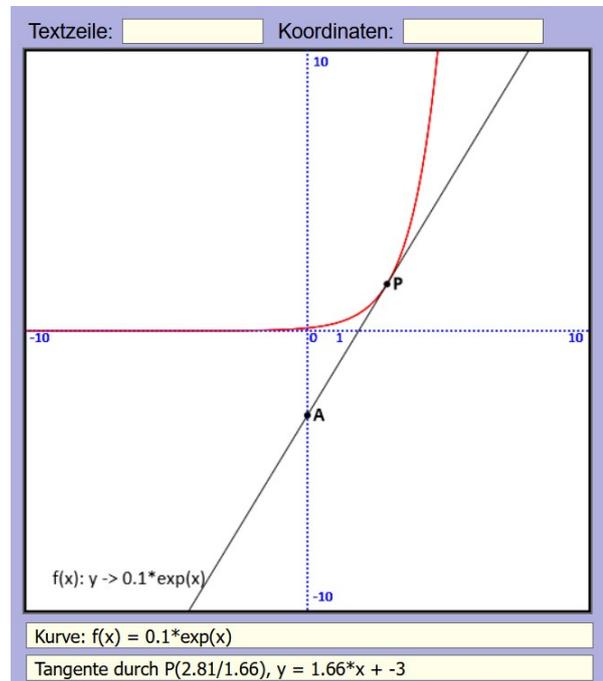
```
// Viertes Übungs-Script
// Schnitt von zwei Kurven f(x) und g(x).
// Die beiden Funktionen werden zuerst mit
// dem Befehl "fuo" direkt eingegeben und
// dann versucht der Befehl "sfg" auf einem
// Intervall den Kurven-Schnittpunkt zu finden.
// Eine alternative Funktionseingabe wird
// durch den Befehl "nof" verhindert.
```

```
begin
clr()
inf(Funktionen f(x) eingeben. ;)
nof()
fuo(0.5*exp(x))
txt(f(x): y -> _s; -9; -8)
fuo(-1*x^2/5+8)
txt(g(x): y -> _s; -9; -9)
inf(Suchintervall eingeben. ;)
l = 0
r = 5
sfg(f(x);g(x);l;r)
end
```

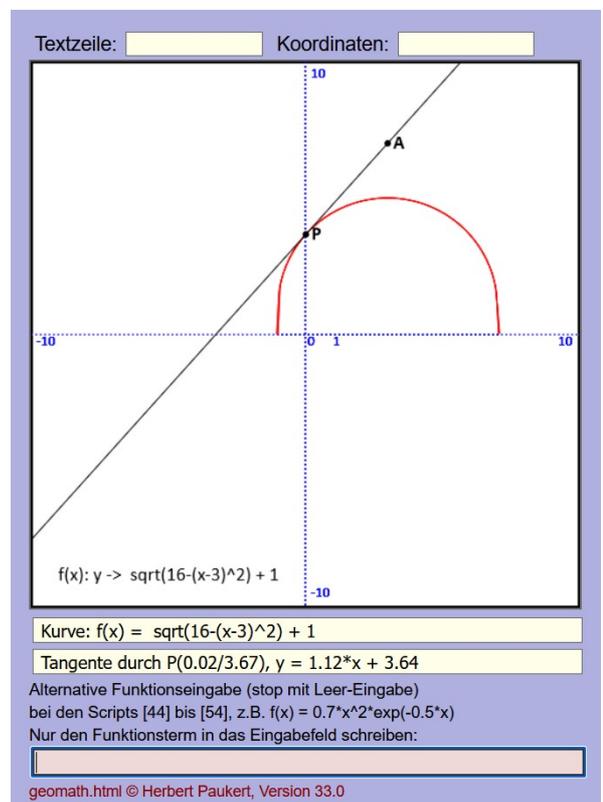


// Fünftes Übungs-Script  
 // Tangente von Punkt an eine Kurve.  
 // Die Funktion f(x) wird zuerst mit dem  
 // Befehl "fuo" direkt eingegeben und  
 // auch ein Punkt A. Dann versucht  
 // der Befehl "tg2" auf einem Intervall  
 // die Tangente von A an f(x) zu finden.  
 // Eine alternative Funktionseingabe ist  
 // im Eingabefeld möglich. Zusätzlich  
 // kann mit der Eingabe von „w“ der  
 // letzte Funktionsterm wiederholt werden.

```
begin
clr()
inf(Funktion f(x) eingeben.;)
fuo(0.1*exp(x))
txt(f(x): y -> _s;-9;-9)
inf(Weiter mit Punkt A;)
x = 0
y = -3
A(x/y)
inf(Suchintervall eingeben.;)
l = 0
r = 5
tg2(f(x);A;l;r)
end
```



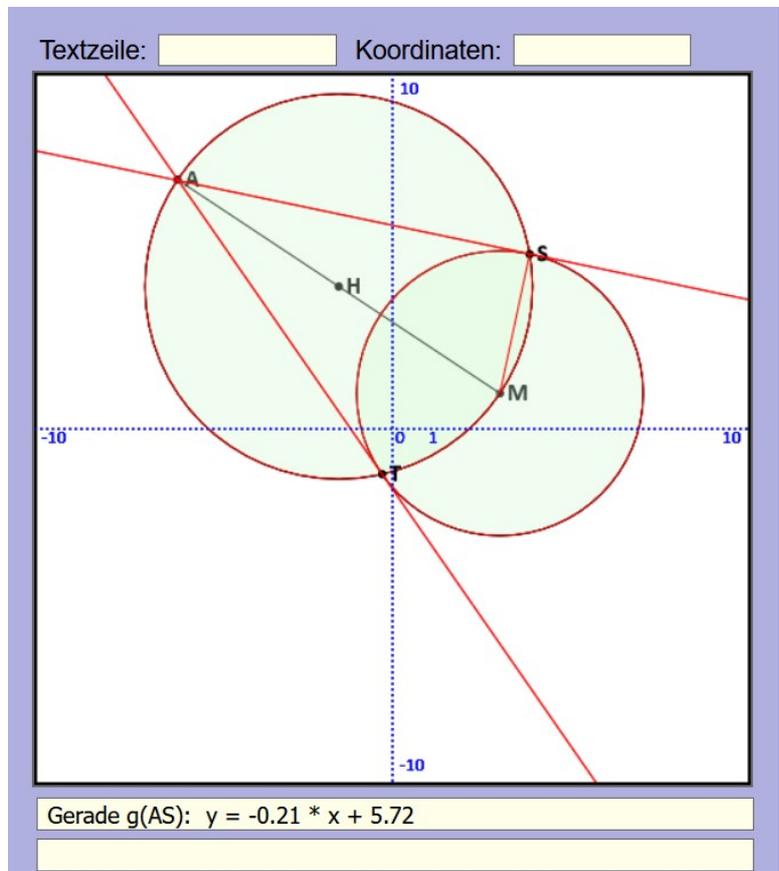
Alternative Funktionseingabe  
 mit Funktion im Eingabefeld



Funktionswiederholung mit „w“  
 mit leerem Eingabefeld

// Sechstes Übungs-Script  
 // Tangenten von Punkt A an den Kreis k(M/r)  
 // Konstruktion mit dem Lehrsatz von Thales

```
begin
clr()
inf(Kreis eingeben. ;)
x = 3
y = 1
M(x/y)
r = 4
s = r
krs(M;r)
inf(Punkt eingeben. ;)
a = -6
b = 7
A(a/b)
lin(A;M)
inf(Strecke AM halbieren. ;)
hap(A;M)
lng(A;H)
krs(H;z)
inf(Hilfskreis zeichnen. ;)
skk(M;s;H;z)
inf(Hilfskreis mit Kreis schneiden. ;)
lin(A;S)
lin(M;S)
lng(M;S)
inf(Weiter. ;)
win(M;S;A)
inf(Weiter. ;)
ger(A;T)
ger(A;S)
end
```



## Screenshots des integrierten Taschenrechners (calc.html)

\*\*\* Hilfsinformation, Teil 1 \*\*\*

-----  
 Rechner (29.3) von H. Paukert.  
 Teil 2 mit Doppelklick auf [del].  
 -----

Eingabezeile löschen: [clear].  
 Letzte Eingabe löschen: [del].  
 2 Dezimalen ein/aus: [dez2].  
 4 Speicher: [stoN-rclN], N=1..4  
 2 fixe Konstante: [pi] und [e].  
 -----

Operationen: -----  
 (,),+,-,\*,/,^,p%,r%,-z,1/z,z!,-=  
 z.B.: Prozente  $100*2$  [p%] (=2)  
 z.B.: Rest von  $100/3$  [r%] (=1)  
 z.B.:  $4^3$  (=64),  $64^{(1/3)}$  (=4)  
 -----

Funktionen: -----  
 lg,sqr,sqrt,sin,cos,tan,  
 log,exp, asin,acos,atan,round.  
 z.B.:  $\text{sqr}(4)$  (=16),  $\text{sqrt}(16)$  (=4)  
 z.B.:  $\text{sin}(90)$  (=1),  $\text{asin}(1)$  (=90)  
 [fpos] wechselt die Position  
 der Funktion in der Eingabezeile  
 (entweder links oder rechts).  
 Manuelle Zusatzfunktionen: floor,  
 rad, deg, atan2(y,x), pow(z,n)...  
 -----

[OK]

\*\*\* Hilfsinformation, Teil 2 \*\*\*

-----  
 Rechner (29.3) von H. Paukert.  
 Teil 1 mit Mausklick auf [help].  
 -----

[var] und (name,value) belegt die  
 Variable "name" (a,b,...z) mit dem  
 Wert "value", um damit zu rechnen,  
 z.B.  $\text{sqr}(\text{sqr}(a)+\text{sqr}(b))$  und [=].  
 Mit [var] und (all,0) werden alle  
 Variable und Speicher gelöscht.  
 Dieses Verfahren ermöglicht eine  
 einfache Algebra mit 26 Variablen,  
 welche verwendet werden, um bequem  
 mit langen Zahlenwerten zu rechnen.  
 -----

Alle Rechenresultate [=] werden  
 in die Zwischenablage kopiert.  
 Einfügen dann mittels [Strg V]  
 oder markieren und verschieben.  
 -----

Hinweis: Wird der Rechner als  
 PopUp in einem Programm mit [X]  
 geschlossen, kann er dort nicht  
 mehr aufgerufen werden!  
 -----

[OK]

name,value \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

a,3| \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

b,4| \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

sqrt(a^2+b^2) \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

(sqr(a) + sqr(b))^(1/2) \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

5 \*

1	2	3	+
4	5	6	-
7	8	9	*
(	0	.	/
)	p%	r%	^
-z	1/z	!z	=
clear	del'	dec2	fpos
sto1	rcl1	sto2	rcl2
sto3	rcl3	sto4	rcl4
sin	cos	tan	sqr
asin	acos	atan	sqrt
e	exp	log	lg
pi	round	var	help

## Screenshots von besonderen Funktionen

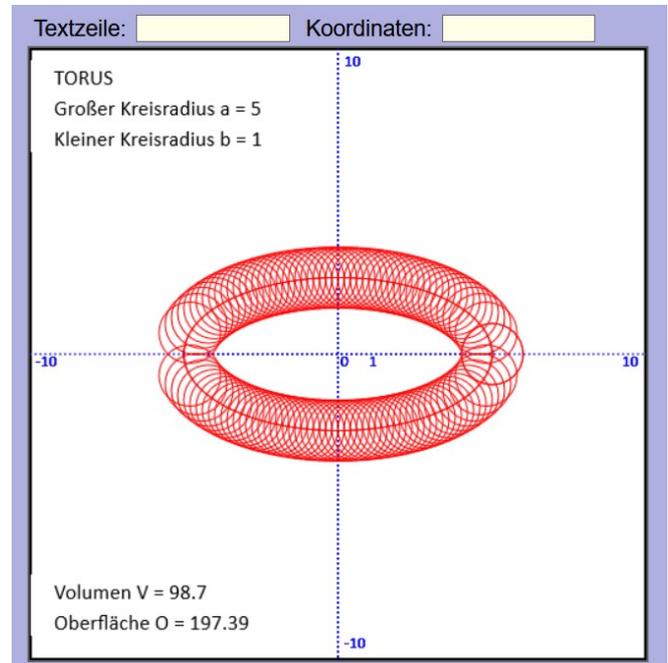
### Demo-Script [81] „Torus“

Textzeile:  Koordinaten:

**Torus:**  
 Jeder Punkt P auf einem großen Kreis  $k[M/a]$  ist der Mittelpunkt eines kleinen Kreises  $k[P/b]$  mit  $b < a$ .  
 Die beiden Kreisebenen sind aufeinander normal. Damit wird ein Torus gebildet. Für Volumen und Oberfläche gilt:  
 $V \rightarrow (b^2 \cdot \pi) \cdot (2 \cdot \pi \cdot a)$  und  $O \rightarrow (2 \cdot \pi \cdot b) \cdot (2 \cdot \pi \cdot a)$   
 Viele Gegenstände des Alltags enthalten Torusteile:  
 Räder | Rollen | Rohre | Ketten | Töpfe . . .

Für die äußere (a) und die innere (i) Torushälfte gilt:  
 $V(a) \rightarrow (b^2 \cdot \pi / 3) \cdot (3 \cdot \pi \cdot a + 4 \cdot b)$   
 $V(i) \rightarrow (b^2 \cdot \pi / 3) \cdot (3 \cdot \pi \cdot a - 4 \cdot b)$   
 Für das Volumen eines Topfes mit abgerundetem Boden müssen die Torushälften beachtet werden!

Viel Vergnügen beim Experimentieren!

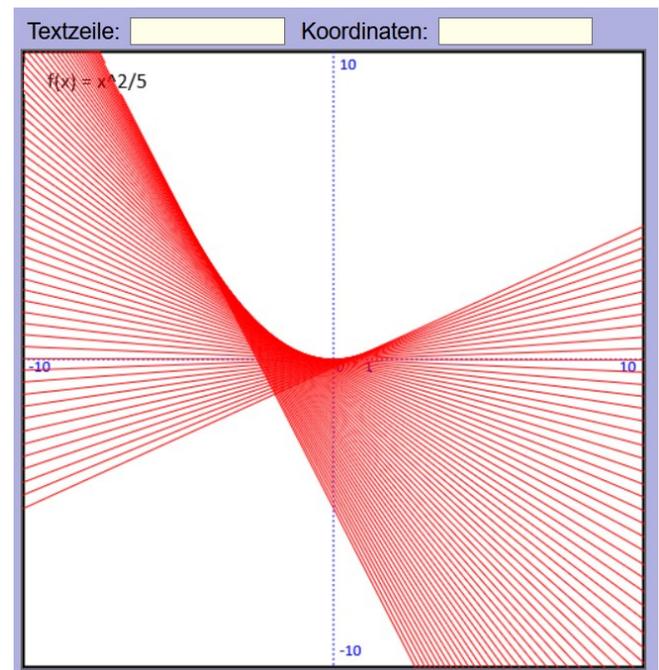


### Demo-Script [82] „Gleitende Tangenten“

Textzeile:  Koordinaten:

**Gleitende Tangenten:**  
 In jedem Punkt P einer Kurve  $y \rightarrow f(x)$  auf dem Intervall  $a < x < b$  wird die Tangente bestimmt und gezeichnet. Dadurch entsteht eine Schar von Tangenten.

Viel Vergnügen beim Experimentieren!



## Screenshots von besonderen Funktionen

### Demo-Script [83] „Evoluten“

Textzeile:  Koordinaten:

Evoluten:  
 Ein Krümmungskreis schmiegt sich optimal in einem Kurvenpunkt  $P(a/f(a))$  an eine Kurve  $f(x)$ .  
 Eine Funktion  $g(x)$  durch die Mittelpunkte von allen Krümmungskreisen heißt Evolute der Kurve.  
 Es gilt:  $f(a) \rightarrow g(a) \mid f'(a) \rightarrow g'(a) \mid f''(a) \rightarrow g''(a)$ .  
 Mit dem Parameter  $n \rightarrow 2$  wird zusätzlich zu  $f(x)$  noch das Spiegelbild  $-f(x)$  gezeichnet. Beispiele:

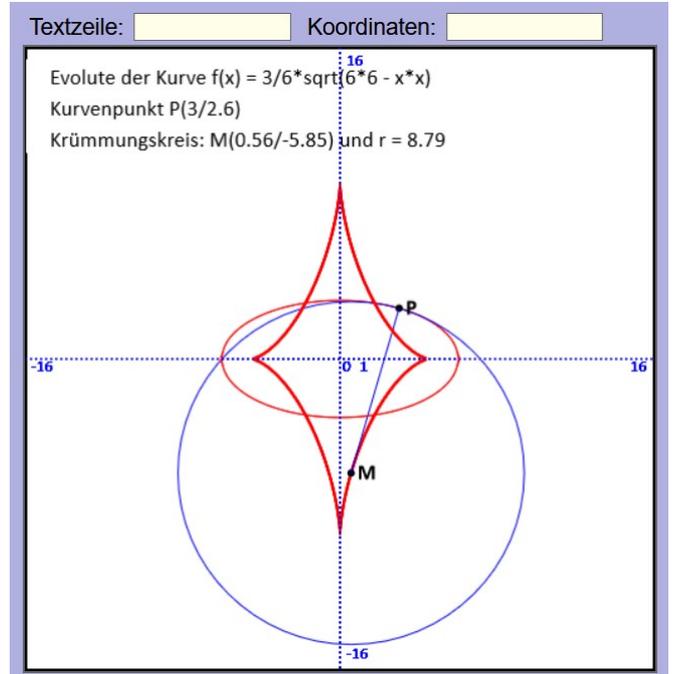
-16 0 1 16

Ellipse  $f(x) \rightarrow 3/6 \cdot \sqrt{6^2 - x^2}$   
 Hyperbel  $f(x) \rightarrow 3/2 \cdot \sqrt{x^2 - 2^2}$   
 Parabel  $f(x) \rightarrow 2 \cdot \sqrt{x}$

Achtung auf Randstellen der Funktion - z.B. bei der Ellipse bei  $x \rightarrow 6$ . Um Fehler bei Krümmungskreisen zu vermeiden sollte dort  $x \rightarrow 5.999$  eingegeben werden!

Viel Vergnügen beim Experimentieren!

-16



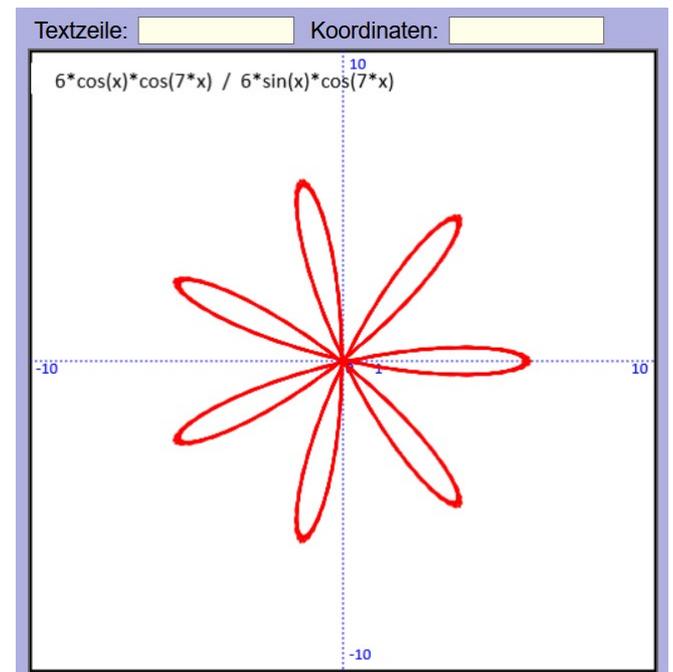
### Demo-Script [84] „Parameter-Funktionen“

Textzeile:  Koordinaten:

Eine Funktion  $y \rightarrow f(x)$  wird dadurch bestimmt dass  $x$  und  $y$  Funktionen von einem Parameter  $t$  sind:  $x \rightarrow g(t)$  und  $y \rightarrow h(t)$ . In den folgenden Beispielen müssen die Funktionen durch einen Schrägstrich getrennt eingegeben werden und für den Parameter  $t$  wird das Argument  $x$  verwendet:

Ellipse:  $( 5 \cdot \cos(x) / 3 \cdot \sin(x) )$   
 Zykloide:  $( x - \sin(x) / 1 - \cos(x) )$   
 Asteroide:  $( 5 \cdot \cos(x)^3 / 5 \cdot \sin(x)^3 )$   
 Lissajous:  $( 5 \cdot \sin(5 \cdot x) / 5 \cdot \cos(3 \cdot x) )$   
 Rosetten:  $( 6 \cdot \cos(x) \cdot \cos(7 \cdot x) / 6 \cdot \sin(x) \cdot \cos(7 \cdot x) )$

Viel Vergnügen beim Experimentieren!



## Screenshots von besonderen Funktionen

### Demo-Script [85] „Hypozykloide“

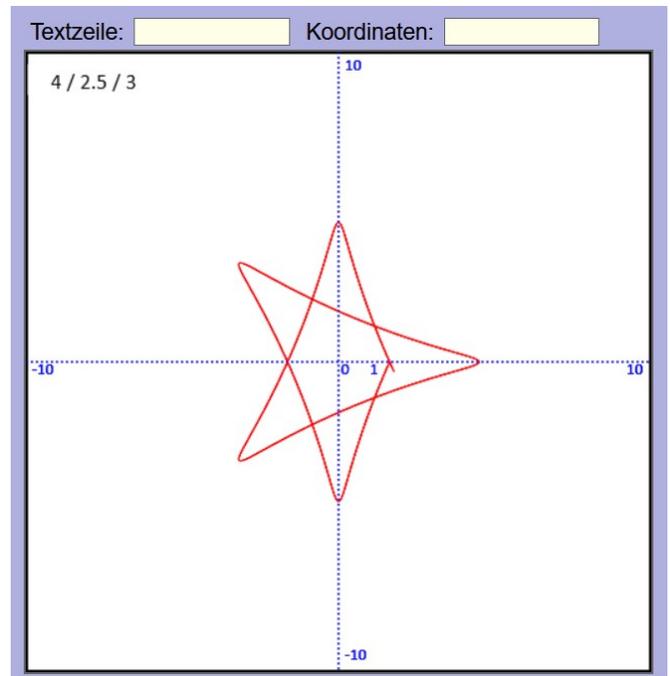
Textzeile:  Koordinaten:

Ein kleiner Kreis ( $r \rightarrow b$ ) rollt innen entlang eines großen Kreises ( $r \rightarrow a$ ). Ein Punkt P ist mit dem Mittelpunkt M des kleinen Kreises starr verbunden mit dem Abstand  $MP \rightarrow c$ . Die Bahnkurve von P heißt Hypozykloide.  
Fallunterscheidung  $c : 0 \mid c < b \mid c : b \mid c > b$

Beispiele für  $(a/b/c)$ :  
 $(4/1/0) - (4/1/1) - (4/1/2) - (4/1/3)$   
 $(4/3/2) - (4/3/3) - (4/3/4) - (4/2.5/3)$   
 $(6/3.5/2.5) - (7/3/5) - (7/4/4) - (7/1.5/2.5)$

Rollt ein Kreis hingegen auf einer Geraden dann beschreibt ein fester Punkt P auf dem Kreisumfang als Bahnkurve eine Zykloide.

Viel Vergnügen beim Experimentieren!




---



---

**ENDE**

---



---